



Support du cours

Architecture des ordinateurs

Encadré par : A. ERRAMI

Année 2010 - 2011

PLAN DU COURS

Introduction : description générale des différents éléments d'un ordinateur

Chapitre1 : Les nombres et les systèmes de numération

- Le système binaire, le système décimal, le système hexadécimal
Les nombres signés et les nombres non signés
Représentation des caractères
Représentation des réel sous format flottant

Chapitre2 : Architecture interne et principe de fonctionnement du microprocesseur

Les éléments de base d'un ordinateur
Principe de fonctionnement du microprocesseur

Chapitre 3 : Mécanisme d'augmentation des performances dans les nouvelles architectures

- Mécanisme du PIPELINE
- Parallélisme et architecture super scalaire
Mécanisme du cache mémoire
Mécanisme du DMA

Chapitre4 : Les BUS d'ordinateurs

- Les différents niveaux de bus
- Les signaux et les paramètres d'un BUS
 - Exemples de BUS
Mécanismes de partage de BUS

Chapitre5 : Le système d'interruption

- Les différents types d'interruptions Reconnaissance de la source d'interruption
- Traitement d'une demande d'interruption
 - Principe d'un contrôleur d'interruption

Chapitre6 : La mémoire centrale

Structure interne et principe de fonctionnement d'un boîtier mémoire
Classification des boîtiers mémoire
Les barrettes mémoire
Gestion de la mémoire centrale et notion de mémoire virtuelle

Chapitre7 : Les entrées sorties d'un système à microprocesseur

- Classification des périphériques et des types d'interfaces
- Principes d'échange de données en entrée et en sortie
- Exemples de périphériques

Introduction

Qu'entend-t-on par architecture ?

L'architecture d'un système à microprocesseur représente l'organisation de ses différentes unités et de leurs interconnexions. Le choix d'une architecture est toujours le résultat d'un compromis :

- entre performances et coûts
- entre efficacité et facilité de construction
- entre performances d'ensemble et facilité de programmation

Qu'est ce qu'un ordinateur ?

Un ordinateur est une machine de traitement de l'information.

Il est capable de réaliser les opérations suivantes :

- * acquisition de l'information
- * stockage de l'information
- * transformation de l'information par son traitement
- * restitution de l'information

Le mot informatique vient de la contraction des deux mots : information et automatique.

Nous distinguons différents types d'informations :

- * données sous forme de texte, de nombres, de son ou d'images
- * les instructions composant un programme.

Les principales composantes d'un ordinateur

- * une unité centrale (boîtier)
- * des unités de saisies (clavier, souris, ...) ;
- * des unités de sorties (écrans, imprimantes, ...).

Les formes de boîtiers d'un ordinateur



5- L'ordinateur portable



L'écran d'ordinateur

L'écran (ou moniteur informatique) est le tout premier périphérique de sortie de l'ordinateur. Il est connecté à une carte d'affichage (aussi appelée carte graphique) qui est elle même insérée dans un connecteur d'extension de l'unité centrale.

Les moniteurs se différencient par :

- leur taille : elle est définie par la taille de la diagonale exprimée en pouces : 14, 15, 17, 20 et 21
- la résolution : elle présente la finesse avec laquelle l'image



est affichée, elle se mesure en nombre de points appelés

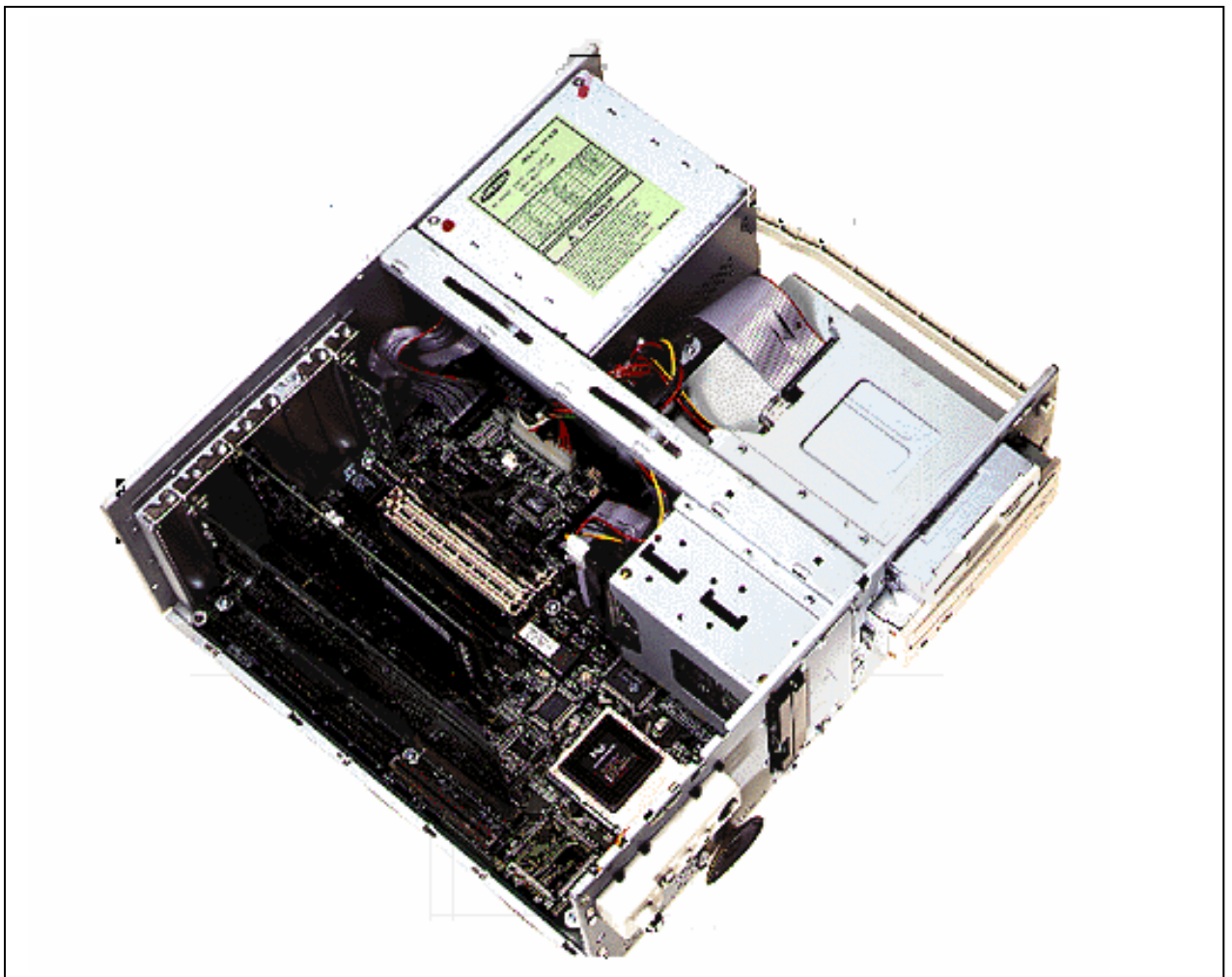
‘ pixels ‘ , qui peuvent être affichés horizontalement et verticalement.

- La norme VGA impose un minimum de 640 x 480 pixels avec 16 couleurs
- La norme SVGA peut atteindre 1280 x 1024 pixels avec 16 millions de couleurs.
- la fréquence : l 'image affichée par un moniteur est en permanence redessinée (rafraîchie) à une fréquence exprimée en hertz (Hz) . Cette fréquence doit être au minimum 75 Hz.

Les éléments de l'unité centrale

L'unité centrale est le cœur de la machine et c'est au sein de cette unité que s'effectue pratiquement toutes les opérations possibles (arithmétiques, logiques, transfert de données, les commandes, les signalisations, etc...). Cette unité comporte, en général, les éléments suivants :

- Bloc alimentation
- Carte mère ;
- Lecteur de disquette
- Lecteur de CD-ROM ou DVD-ROM
- Disque(s) dur(s) ;
- Carte graphique ;
- Carte son ;
- Carte contrôleur (de disque dur, de lecteur de disquette, etc)



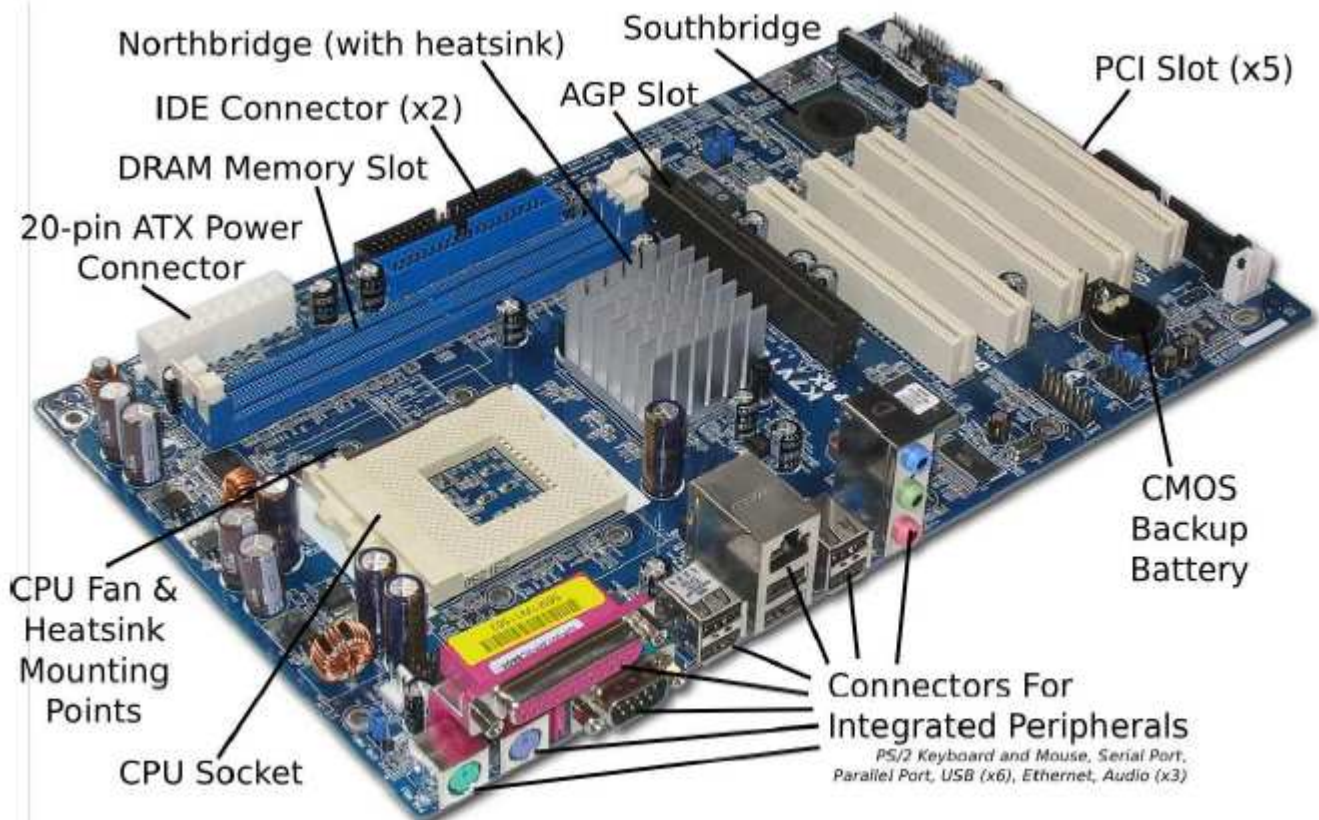
La carte mère et ses composants

La carte mère est l'entité principale de l'UC. Parmi ses principaux éléments :

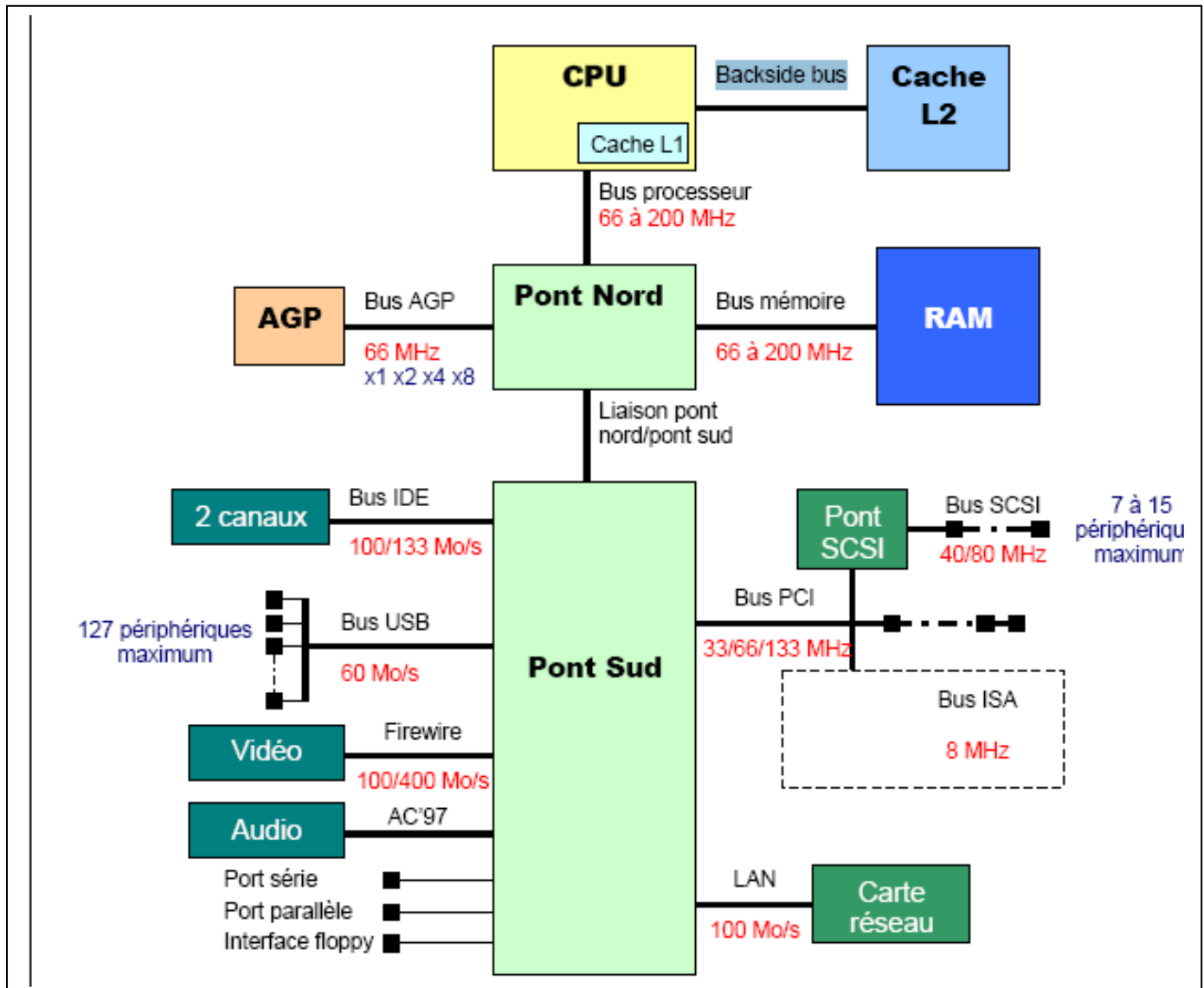
- le microprocesseur
- la mémoire principale ou central
- les bus

Les caractéristiques principales d'une carte mère sont :

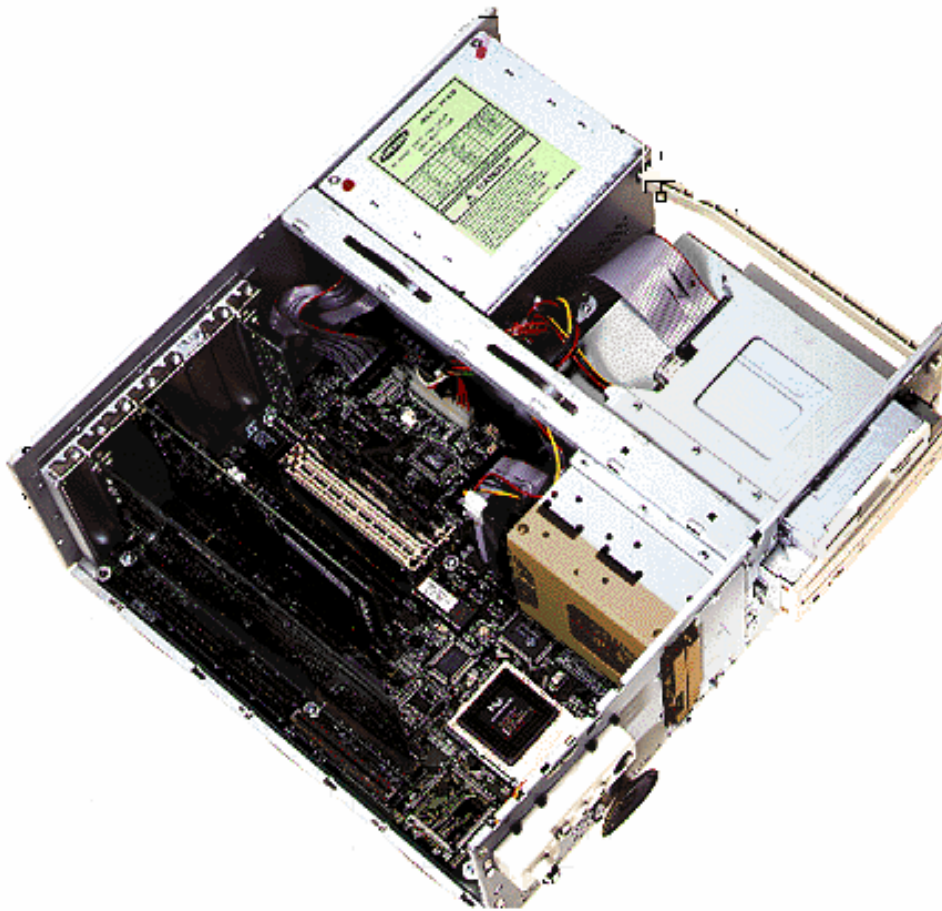
- Type de microprocesseurs et la fréquence de travail ;
- Taille de la mémoire RAM ;
- Taille de la mémoire cache ;
- Vitesse des bus ;
- Type de bios utilisé.



Architecture d'une carte mère



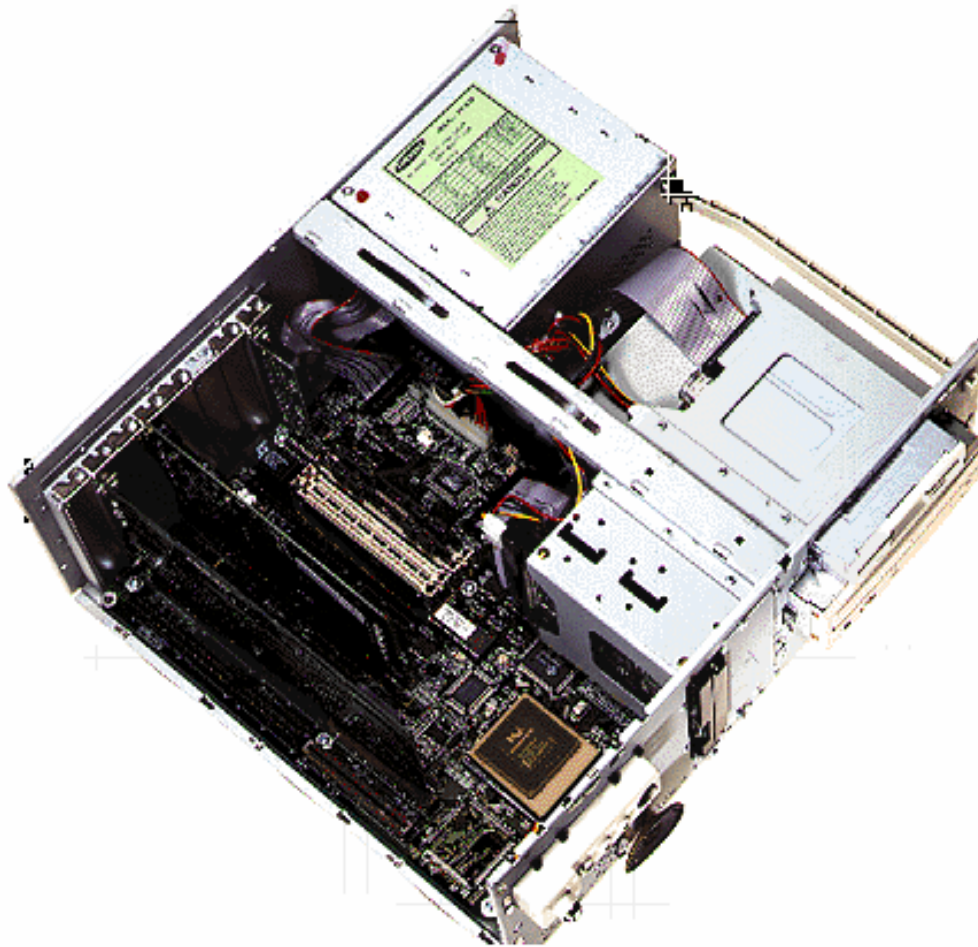
Le disque dur



Le disque dur, dont il existe deux **types**, est un périphérique de **stockage** à enregistrement magnétique, sur lequel sont installés le système, les applications et les documents. Sa capacité est mesurée en millions (méga-octets ou **Mo**) ou en milliards (giga-octets ou **Go**) d'octets.

La mémoire vive de l'ordinateur se vide à l'extinction de celui-ci. C'est pourquoi il est nécessaire d'enregistrer (on dit sauvegarder) sur le disque dur les données sur lesquelles on travaille. Les deux principaux types de disques durs sont appelés IDE, SATA et SCSI. Le premier est le meilleur choix pour un usage courant, alors que le disque SCSI montre certains avantages dans un environnement réseau.

Le microprocesseur



Le **choix** du microprocesseur a une grande importance pour le type de performances que vous pouvez espérer. Il est le véritable **cerveau** de votre système. La famille de processeurs la plus répandue aujourd'hui est le **Pentium** d'Intel, dont il existe plusieurs **versions**.

La famille Pentium se subdivise en plusieurs générations de processeurs : du pentium original à 60 MHz jusqu'au récent Pentium IV à plusieurs GHz.

Deux autres fabricants de processeurs : AMD et CYRIX proposent des alternatives à certains modèles de pentium.

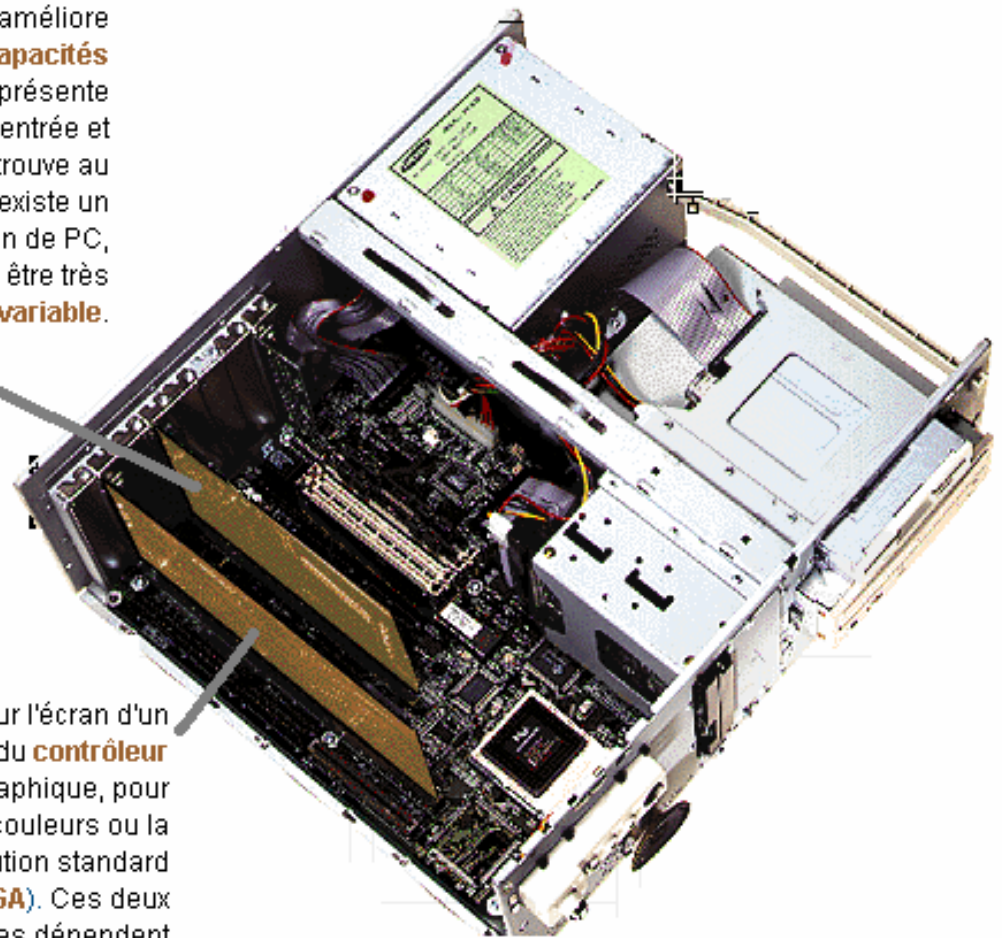


**Système de ventilation
d'un microprocesseur**

La carte graphique

La carte son améliore considérablement les **capacités** musicales du PC. Elle présente plusieurs **connecteurs** d'entrée et de sortie, que l'on trouve au dos de l'ordinateur. S'il existe un **standard** pour les cartes son de PC, leur qualité sonore peut être très **variable**.

L'affichage sur l'écran d'un ordinateur dépend du **contrôleur** graphique, pour le nombre de couleurs ou la **résolution** (la résolution standard est appelée **VGA**). Ces deux critères dépendent de la quantité de **mémoire** installée sur la carte graphique.



Les imprimantes



Pour pouvoir conserver une trace "écrite" de votre travail sur ordinateur, vous avez besoin d'une imprimante. Choisir la bonne imprimante n'est pas qu'une question d'argent. Cela dépend fondamentalement de ce que vous en attendez, d'un usage bureautique intensif à une utilisation domestique de loisirs. Si l'on excepte quelques technologies très coûteuses réservées à des applications très particulières, il existe trois types d'imprimantes : **matricielles**, **jet d'encre** et **laser**.

Programmation

Un programme, souvent appelé logiciel (*software*), est une séquence d'instructions qui indique au matériel informatique (*hardware*) les opérations qu'il doit effectuer sur les données.

Les programmes peuvent être intégrés au hardware, comme sur les micro-ordinateurs que l'on trouve dans les calculatrices, les montres, les moteurs automobiles ou les fours à micro-ondes.

Sur un ordinateur classique coexistent deux types de logiciel, selon le type de tâche qu'ils exécutent.

- * Les logiciels d'application prennent en charge la multitude des tâches pour lesquelles sont utilisés les ordinateurs : traitement de texte, gestion de bases de données, etc.

- * Les logiciels système, qui sont souvent invisibles à l'utilisateur, contrôlent le fonctionnement de l'ordinateur : on les appelle systèmes d'exploitation.

Système d'exploitation

Les systèmes d'exploitation gèrent l'allocation et l'utilisation des ressources matérielles de l'ordinateur, telles que la mémoire, l'unité centrale de traitement, l'espace du disque dur et les périphériques (imprimante, manette de jeu, modem, etc).

Ce sont des logiciels stockés de façon permanente dans la mémoire, qui interprètent les commandes de l'utilisateur suivant le service qu'il désire : afficher, imprimer ou copier des fichiers de données, lister tous les fichiers d'un répertoire, ou encore exécuter un programme particulier.

Monotâches et multitâches

Il existe deux types de systèmes d'exploitation : les systèmes monotâches, qui ne peuvent exécuter qu'une tâche à la fois, et les systèmes multitâches, qui permettent d'effectuer plusieurs tâches simultanément.

MSDOS : système d'exploitation mono tâche

WINDOWS : système d'exploitation multi tâches

Ebauches historiques :

Les ORDINOSAURES

Machine de PASCAL – 1642



Ebauches historiques :

Les ORDINOSAURES

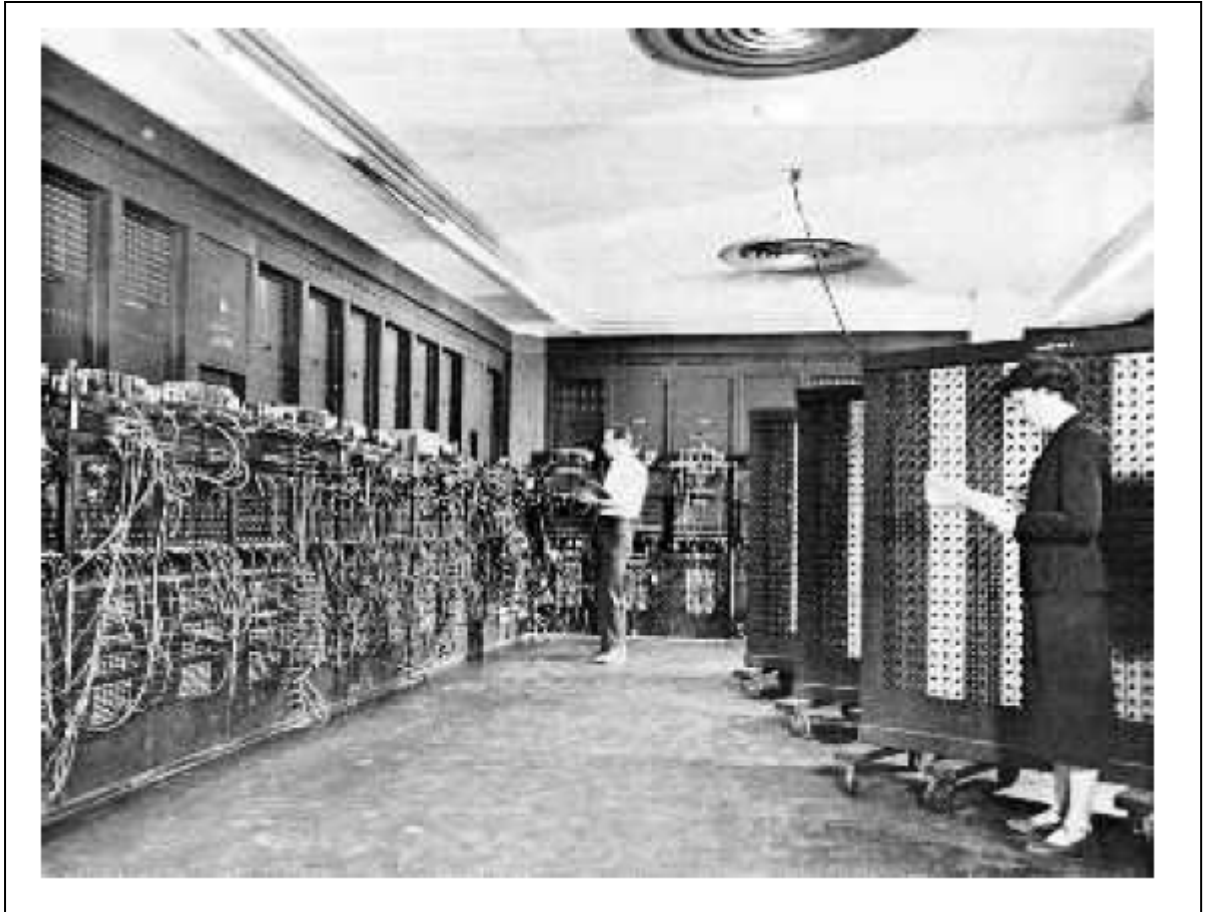
Charles Babbage (1791 – 1871)



Il imagine la *machine à différence*

Ebauches historiques :

ENIAC : Ordinosaure à Tubes (1946)



Ebauches historiques :

Composant élémentaire :

Le TRANSISTOR à tube



John Bardeen, William Shocley, Walter Brattain (1947)

Ebauches historiques :

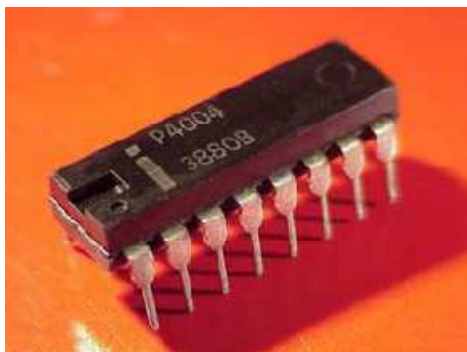
Composant élémentaire :

Le TRANSISTOR à semi-conducteur

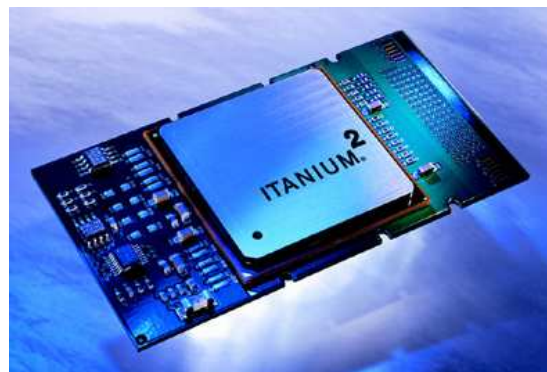


Evolution de la technologie des microprocesseurs

Processeur	Nb de transistors	Année d'apparition
1ère CPU Intel	2300	1971 (cf. ENIAC)
8086	29000	1980
80286	130000	1982
80386DX4	280000	1985
80386SX	275000	1987
486DX4	1,2 millions	1989
486SX	1,185 millions	1991
486DX2	1,2 millions	1992
Pentium	3,2 millions	1993
486DX4	1,6 millions	1994
Pentium Pro	5 millions	1995
Pentium MMX	4,5 millions	1996
Pentium II	7,5 millions	1998
AMD K6	8,8 millions	1997
Celeron	7,5 millions	1998
AMD Athlon (K7)	22 millions	1999
Pentium 4	42 millions	2001
Pentium Extreme	169 millions	2004
Core 2 Duo	291 millions	2006
Core 2 Quad	582 millions	2006
Dual Core Itanium 2	1,7 milliard	2007



1971



2007

Ebauches historiques :

Les premiers ordinateurs de ' l'âge moderne'



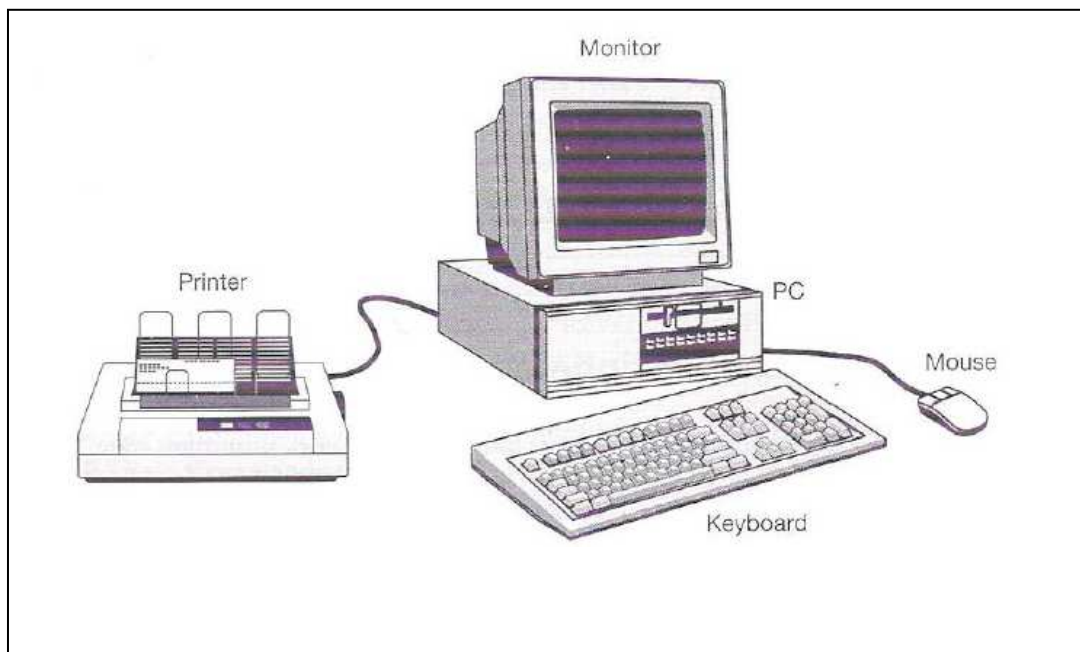
l'apple 1



l'apple II

Ebauches historiques :

Les ordinateurs d'aujourd'hui



Où trouve-t-on des systèmes à microprocesseur ?

Les applications des systèmes à microprocesseurs sont multiples et variées :

- Ordinateur, PDA
- console de jeux
- calculatrice
- télévision
- téléphone portable
- distributeur automatique d'argent
- robotique
- lecteur carte à puce, code barre
- automobile
- instrumentation



1971



Chapitre 1

Les nombres et les systèmes de numération

Soit un nombre N et C_i un ensemble de caractères, on dit que le nombre N est représenté dans la base B s'il existe la relation suivante entre N , C_i et B :

$$N = C_{n-1} \times B^{n-1} + C_{n-2} \times B^{n-2} + \dots + C_1 \times B^1 + C_0 \times B^0$$

B^{n-1} , B^{n-2} , ..., B^1 , B^0 étant les puissances de la base B , ils désignent les poids du nombre N .

Exemples de système de numération :

- système décimal : $B=10$, $C_i \in \{0,1,2,\dots,9\}$
- système hexadécimal : $B=16$, $C_i \in \{0,1,2,\dots,9,A,B,C,D,E,F\}$
- système binaire : $B=2$, $C_i \in \{0,1\}$

Représentation des nombres dans le système Binaire Codé Décimal BCD :

Chaque chiffre décimal composant le nombre est représenté en binaire sur 4 bits :

$$(1\ 3\ 8)_{10} = 0001\ 0011\ 1000$$

Passage d'un système de numération à un autre :

- **du décimal vers le binaire** : technique des divisions successives par 2
- **du binaire vers le décimal** : décomposition en somme des puissances successives de deux
- **du décimal vers l'hexadécimal** : technique des divisions successives par 16.
- **de l'hexadécimal vers le binaire** : décomposition en somme des puissances successives de 16
- **du binaire vers l'hexadécimal et vice versa** : regroupement par bloc de quatre bits

Représentation en binaire des nombres négatifs

Pour représenter les nombres négatifs, on utilise le codage du signe et la technique du complément à deux.

Le complément à deux d'un nombre est obtenu en rajoutant une unité au complément à un.

Le complément à un est obtenu par substitution des '0' par des '1' et des '1' par des '0'.
La valeur du signe est indiqué par le bit de poids fort (le bits MSB) :

- MSB = 0 : signe positif
- MSB = 1 : signe négatif

→ Si le nombre est positif, son MSB = 0 et les autres bits donnent la valeur en binaire du nombre

→ Si le nombre est négatif, son MSB = 1 et les autres bits donnent la valeur en complément à deux du nombre.

Représentation des caractères

Les caractères sont des données non numériques : il n'y a pas de sens à additionner ou multiplier deux caractères. Par contre, il est souvent utile de comparer deux caractères, par exemple pour les trier dans l'ordre alphabétique.

Les caractères, appelés *symboles alphanumériques*, incluent les lettres majuscules et minuscules, les symboles de ponctuation (& ~ , . ; # " - etc...), et les chiffres.

Un texte, ou *chaîne de caractères*, sera représenté comme une suite de caractères.

Le codage des caractères est fait par une table de correspondance indiquant la configuration binaire représentant chaque caractère.

Le code ASCII (*American Standard Code for Information Interchange*) étant le système de codage le plus utilisé à cet effet par les systèmes de traitement de l'information.

Le Code ASCII

BITS					b ₇	0	0	0	0	1	1	1	1
					b ₆	0	0	1	1	0	0	1	1
					b ₅	0	1	0	1	0	1	0	1
					b ₄	0	1	2	3	4	5	6	7
Caractères nationaux	b ₄	b ₃	b ₂	b ₁									
	0	0	0	0	0	NUL	DLE	SP	0	à	P		p
Jeu de commandes	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	£	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	'	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	(	7	G	W	g	w
	1	0	0	0	8	BS	CAN	)	8	H	X	h	x
	1	0	0	1	9	HT	EM	.	9	I	Y	i	y
	1	0	1	0	A	LF	SUB		:	J	Z	j	z
	1	0	1	1	B	VT	ESC	,	;	K		k	é
	1	1	0	0	C	FF	ES		<	L	ç	l	ù
	1	1	0	1	D	CR	GS		"	M	§	m	è
	1	1	1	0	E	SO	RS		>	N	^	n	"
	1	1	1	1	F	SI	US	/	?	O	-	o	DEL

Les commandes du code ASCII

Symbole	Signification	
ACK	Acknowledge	Accusé de réception
BEL	Bell	Sonnerie
BS	Backspace	Retour arrière
CAN	Cancel	Annulation
CR	Carriage Return	Retour chariot
DC	Device control	Commande d'appareil auxiliaire
DEL	Delete	Oblitération
DLE	Data Link Escape	Caractère d'échappement
EM	End Medium	Fin de support
ENQ	Enquiry	Demande
EOT	End Of Transmission	Fin de communication
ESC	Escape	Echappement
ETB	End of Transmission Block	Fin de bloc de transmission
ETX	End Of Text	Fin de texte
FE	Format Effector	Commande de mise en page
FF	Form Feed	Présentation de formule
FS	File Separator	Séparateur de fichiers
GS	Group Separator	Séparateur de groupes
HT	Horizontal Tabulation	Tabulation horizontale
LF	Line Feed	Interligne
NAK	Negative Acknowledge	Accusé de réception négatif
NUL	Null	Nul
RS	Record Separator	Séparateur d'articles
SI	Shift IN	En code
SO	Shift Out	Hors code
SOH	Start Of Heading	Début d'en-tête
SP	Space	Espace
STX	Start Of Text	Début d'en-tête
SYN	Synchronous idle	Synchronisation
TC	Transmission Control	Commande de transmission
US	Unit Separator	Séparateur de sous-article
VT	Vertical Tabulation	Tabulation verticale

Figure 2.5 Le code ASCII.

Représentation des nombres réels : Technique de la virgule Flottante

Les nombres à virgule flottante sont utilisés pour représenter des quantités qui ne peuvent être formulées par des entiers, soit parce qu'elles contiennent des valeurs fractionnelles, soit parce qu'elles tombent en dehors de l'intervalle qui peut être représenté par la largeur de bit du système. Pratiquement tous les ordinateurs modernes utilisent la représentation à virgule flottante spécifiée dans le standard IEEE 754, qui spécifie que les nombres sont représentés par une mantisse et un exposant. À la manière des notations scientifiques, la valeur d'un nombre à virgule flottante est $\text{mantisse} \times 2^{\text{exposant}}$.

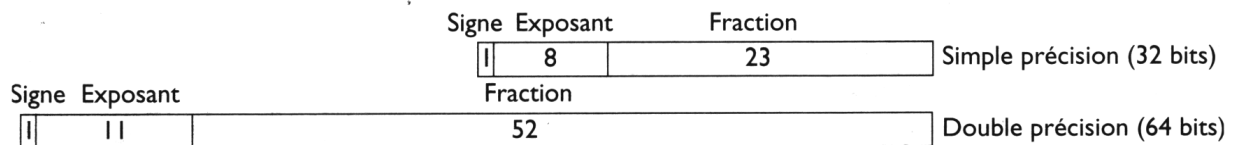
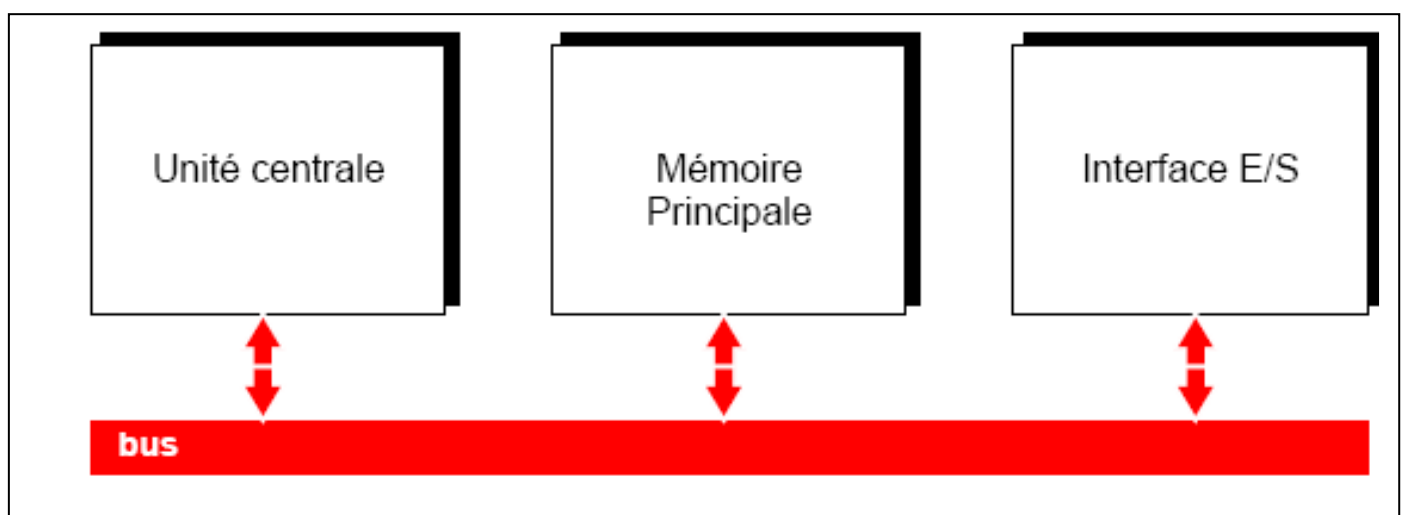


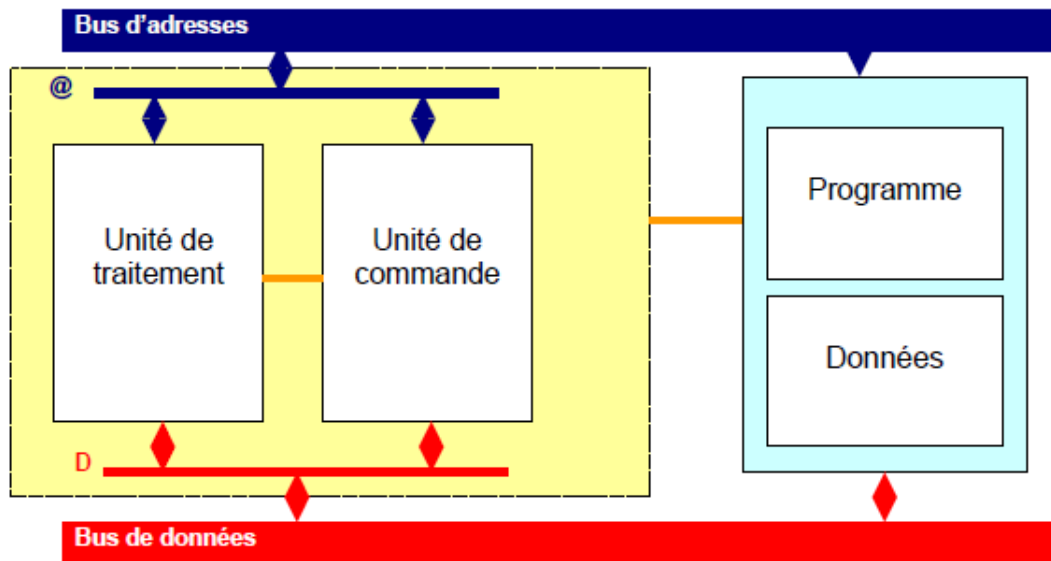
Figure 2.7 Formats à virgule flottante IEEE 754.

Chapitre 2

Architecture interne et principe de fonctionnement du microprocesseur

Architecture de base d'un ordinateur :
Modèle de von Neumann





L'unité centrale

Elle est composée par le microprocesseur qui est chargé d'interpréter et d'exécuter les instructions d'un programme, de lire ou de sauvegarder les résultats dans la mémoire et de communiquer avec les unités d'échange. Toutes les activités du microprocesseur sont cadencées par une horloge.

La mémoire principale

Elle contient les instructions du ou des programmes en cours d'exécution et les données associées à ce programme.

Les interfaces d'entrées/sorties

Elles permettent d'assurer la communication entre le microprocesseur et les périphériques. (capteur, clavier, moniteur ou afficheur, imprimante, modem, etc...).

Les bus

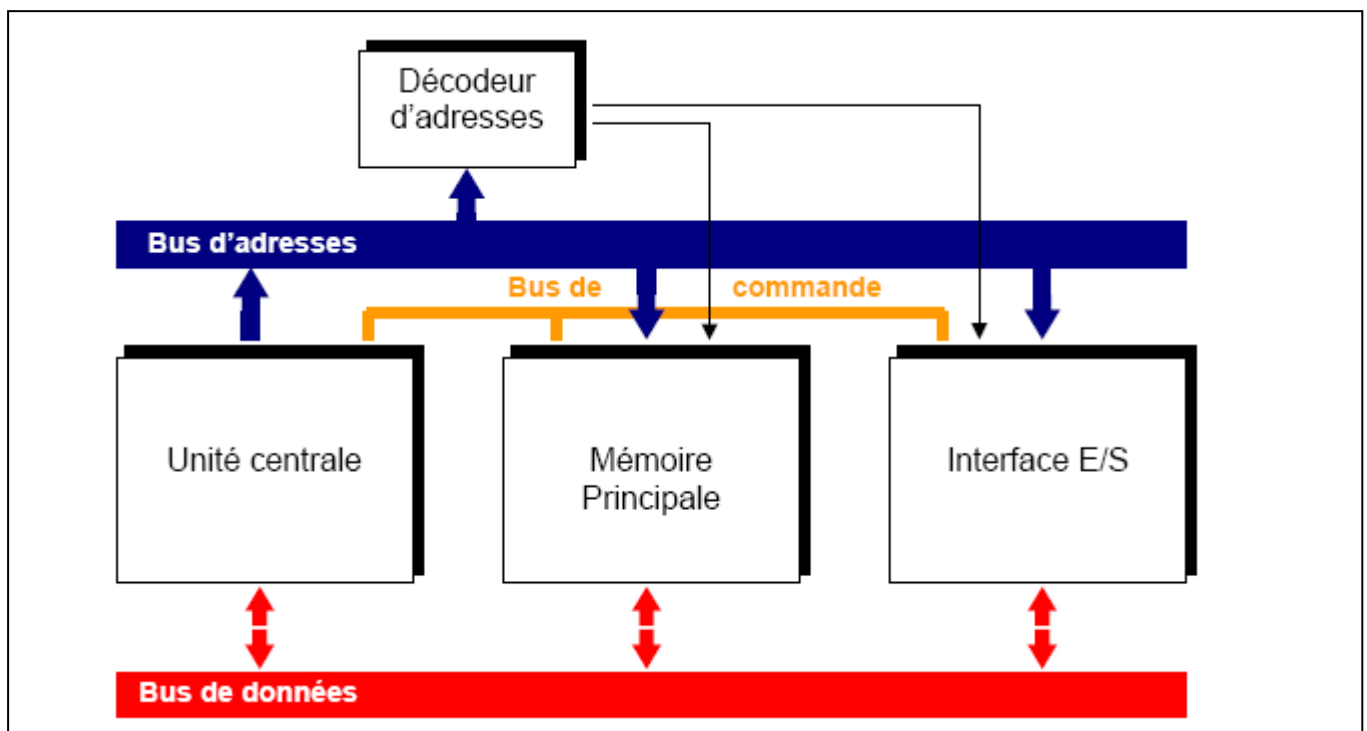
Un bus est un ensemble de fils qui assure la transmission du même type d'information

Décodage d'adresses

La multiplication des périphériques autour du microprocesseur oblige la présence d'un **décodeur d'adresse** chargé d'aiguiller les données présentes sur le bus de données.

En effet, le microprocesseur peut communiquer avec les différentes mémoires et les différents boîtiers d'interface. Ceux-ci sont tous reliés sur le même bus de données et afin d'éviter des conflits, un seul composant doit être sélectionné à la fois.

Lorsqu'on réalise un système microprogrammé, on attribue donc à chaque périphérique une zone d'adresse et une fonction « décodage d'adresse » est donc nécessaire afin de fournir les signaux de sélection de chacun des composants.



Principe de fonctionnement du microprocesseur

Définition du microprocesseur

Un microprocesseur est un circuit intégré numérique, programmable, capable d'exécuter automatiquement un programme comportant un ensemble d'opérations élémentaires.

Le microprocesseur doit assurer, au minimum, les fonctions suivantes:

- opérations arithmétiques et logiques
- décodage et séquençement des instructions
- transfert de données
- possibilité de tests logiques et branchements

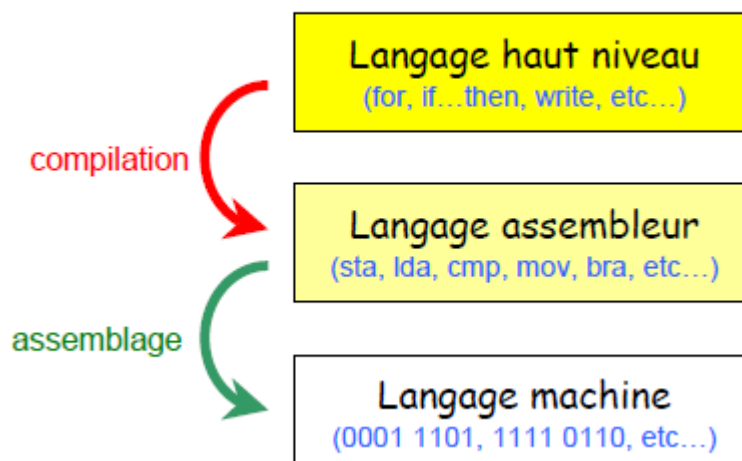
Avantage des microprocesseurs sur la logique câblée

La grande innovation des microprocesseurs est la possibilité de les programmer. Il est ainsi possible de développer un système d'usage relativement général, et de spécifier par la suite, lors de l'adjonction du programme, les différentes et nombreuses applications couvertes par ce même système.

Ceci a pour effet non négligeable de faire décroître dans un facteur important le prix de revient d'un système.

Langage de programmation

Le langage machine est le langage compris par le microprocesseur. Ce langage est difficile à maîtriser puisque chaque instruction est codée par une séquence propre de bits. Afin de faciliter la tâche du programmeur, on a créé différents langages plus ou moins évolués.



Le langage assembleur est le langage le plus « proche » du langage machine. Il est composé par des instructions en général assez rudimentaires que l'on appelle des mnémoniques. Ce sont essentiellement des opérations de transfert de données entre les registres et l'extérieur du microprocesseur (mémoire ou périphérique), ou des opérations arithmétiques ou logiques. Chaque instruction représente un code machine différent. Chaque microprocesseur peut posséder un assembleur différent.

La difficulté de mise en œuvre de ce type de langage, et leur forte dépendance avec la machine a nécessité la conception de langages de haut niveau, plus adaptés à l'homme, et aux applications qu'il cherchait à développer. Faisant abstraction de toute architecture de machine, ces langages permettent l'expression d'algorithmes sous une forme plus facile à apprendre, et à dominer (C, Pascal, Java, etc...). Chaque instruction en langage de haut niveau correspondra à une succession d'instructions en langage assembleur. Une fois développé, le programme en langage de haut niveau n'est donc pas compréhensible par le microprocesseur. Il faut le compiler pour le traduire en assembleur puis l'assembler pour le convertir en code machine compréhensible par le microprocesseur. Ces opérations sont réalisées à partir de logiciels spécialisés appelés compilateur et assembleur.

Exemple de programme :

Code machine (68HC11)	Assembleur (68HC11)	Langage C
@00 C6 64	LDAB #100	A=0 ;
@01 B6 00	LDAA #0	for (i=1 ; i<101 ; i++) A=A+i ;
@03 1B	ret ABA	
@04 5A	DECB	
@05 26 03	BNE ret	

Fonctionnement d'un microprocesseur

La fonction essentielle d'un microprocesseur est d'exécuter une suite d'instructions : le programme.

Structure d'une instruction

Une instruction comporte toujours un code opératoire (*op-code*) qui définit la nature de l'opération à effectuer et très souvent des données qui portent le nom d'opérandes (*operands, data*).

Exemples d'instructions :

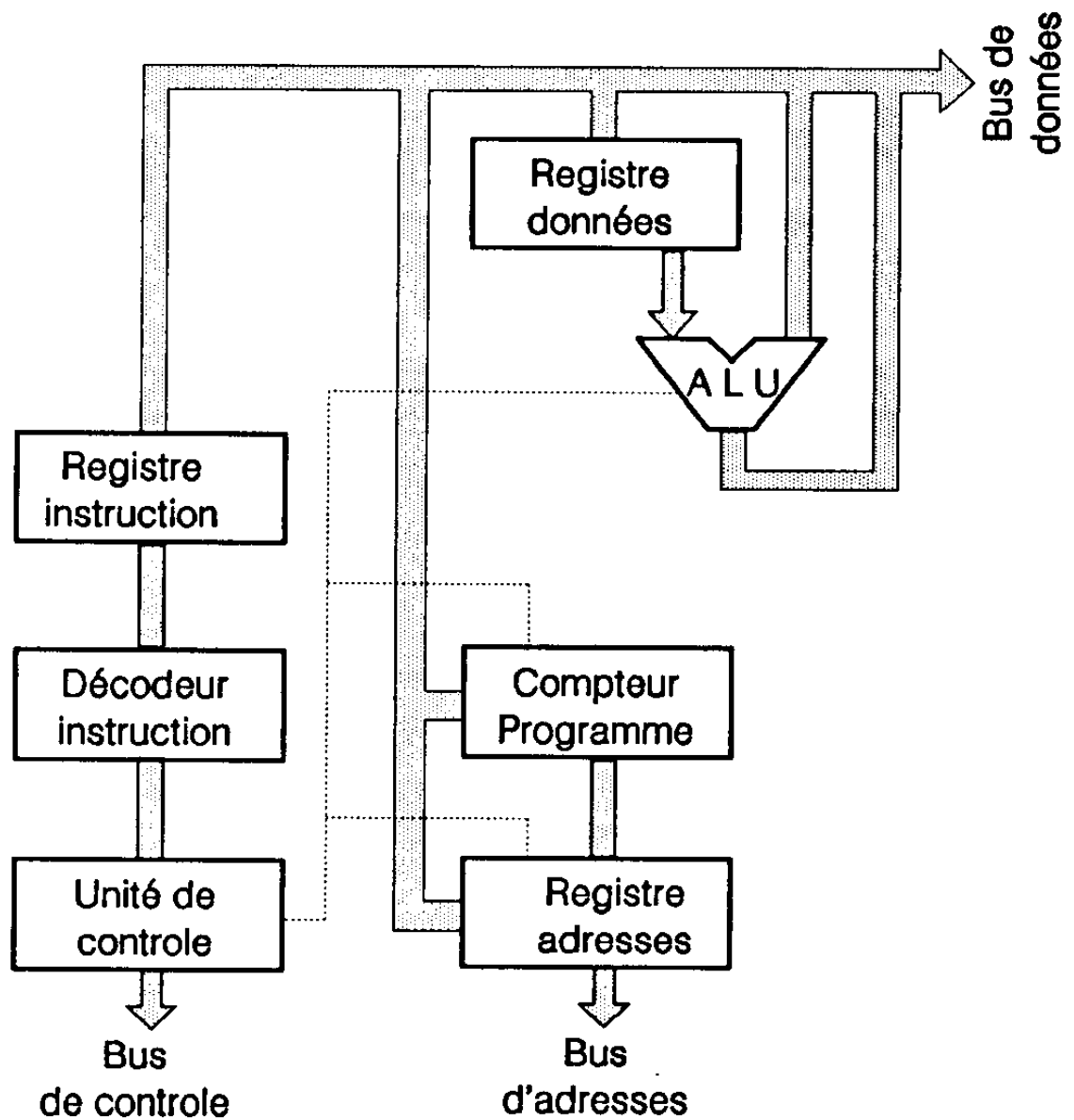
ADD A,B	ADD : code opératoire	A, B : 2 opérandes
NEG C	NEG : code opératoire	C : 1 opérande
NOP	NOP : code opératoire	pas d'opérande

Les données sont traitées par l'Unité Arithmétique et Logique (*ALU : Arithmetic and Logic Unit*) alors que les instructions sont analysées par l'unité de commande (*CU: Control Unit*).

Données et codes opératoires enregistrés dans la mémoire arrivent vers le microprocesseur qui doit, avant traitement, déterminer s'il est en présence d'un code opératoire ou d'une opérande. Pour résoudre ce problème on utilise la convention suivante:

La première information d'un programme est toujours une instruction.

Les informations suivantes sont des données ou des instructions.



Le microprocesseur est constitué d'une unité de commande (UC) qui gère les codes opératoires et d'une ALU qui traite les données. L'UC contient plusieurs organes qui sont :

- le registre d'instructions
- le décodeur d'instructions
- l'unité de contrôle ou séquenceur

Pour exécuter une instruction, l'UC génère une suite d'opérations élémentaires appelées micro-instructions.

Ces micro-instructions sont rangées dans une mémoire interne au microprocesseur et que l'on nomme : mémoire de microprogramme.

Traitement d'une instruction

La toute première chose à effectuer est d'aller récupérer en mémoire le code opératoire de l'instruction, c'est la phase recherche (Fetch).

Ce code opératoire est alors acheminé vers un organe interne au microprocesseur qu'est le registre d'instruction (*Instruction Register*) où il va être stocké temporairement. Le microprocesseur doit alors analyser le code opératoire pour savoir ce qu'il doit faire. C'est la deuxième phase ou phase décodage (decode) , fonction qui est prise en charge par le décodeur d'instruction.

Le décodeur va donc actionner certains circuits afin d'exécuter par leur intermédiaire l'instruction demandée. La troisième phase est donc la phase d'exécution (execute).

Souvent, l'exécution d'une instruction nécessite des opérandes qui doivent être recherchés en mémoire avant l'exécution de l'instruction. C'est l'*unité de commande* (*Sequencer*) qui s'en charge et on a alors les séquences suivantes:

Cherche Décode Cherche Exécute Cherche Décode

1^{ère} instruction

2^{ème} instruction

Alors que les codes opératoires sont chargés dans le registre d'instructions, les opérandes sont stockées dans un autre *tampon* (*Buffer*) que l'on appelle le registre de données (*Data Register*), avant d'être traitées par l'ALU.

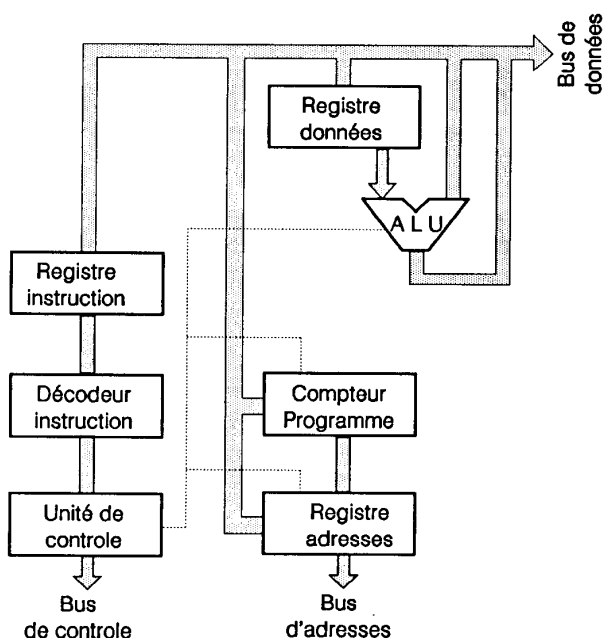
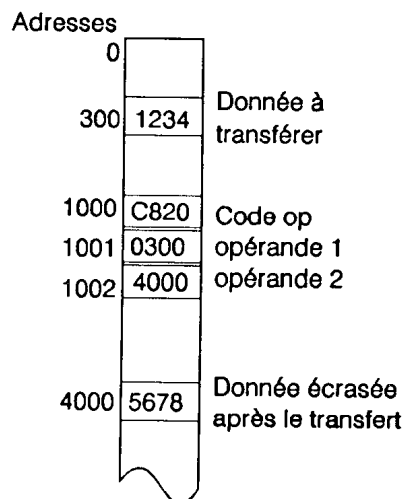
Exemple d'exécution d'une instruction

Soit à exécuter l'instruction :

MOVE @300,@4000

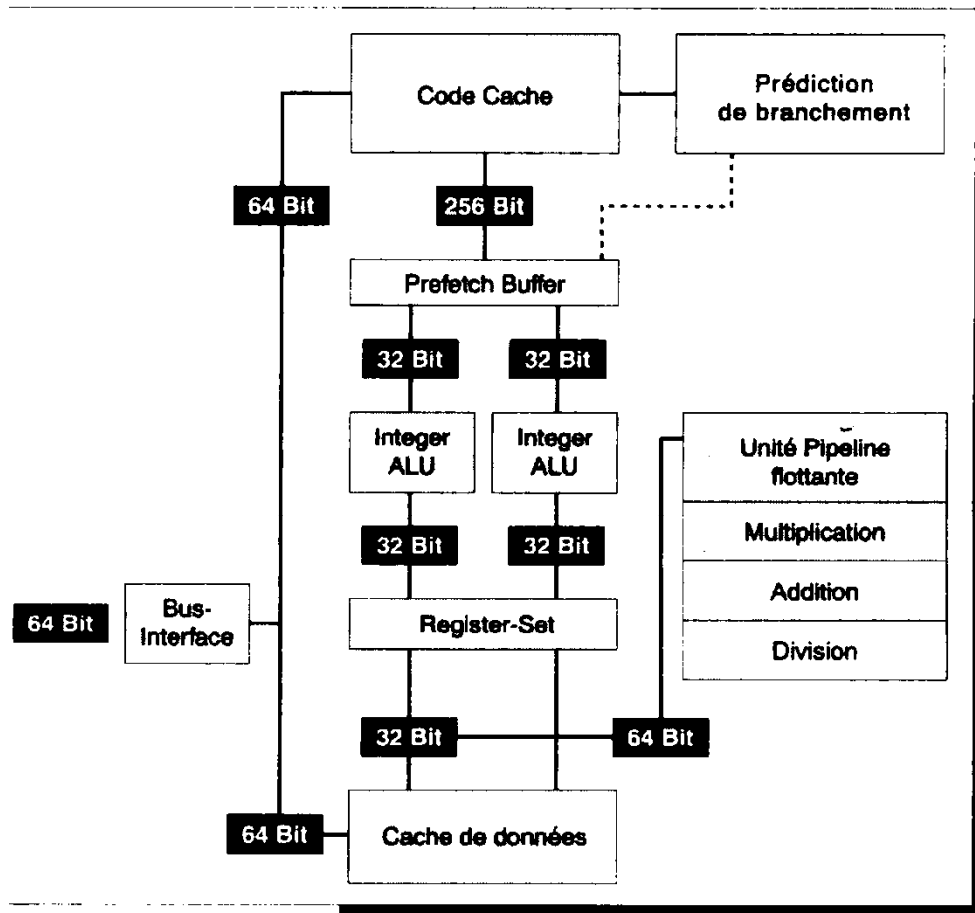
qui consiste à prendre le contenu de l'adresse 300 et à le transférer à l'adresse 4000.

Le microprogramme de cette instruction pourrait être :

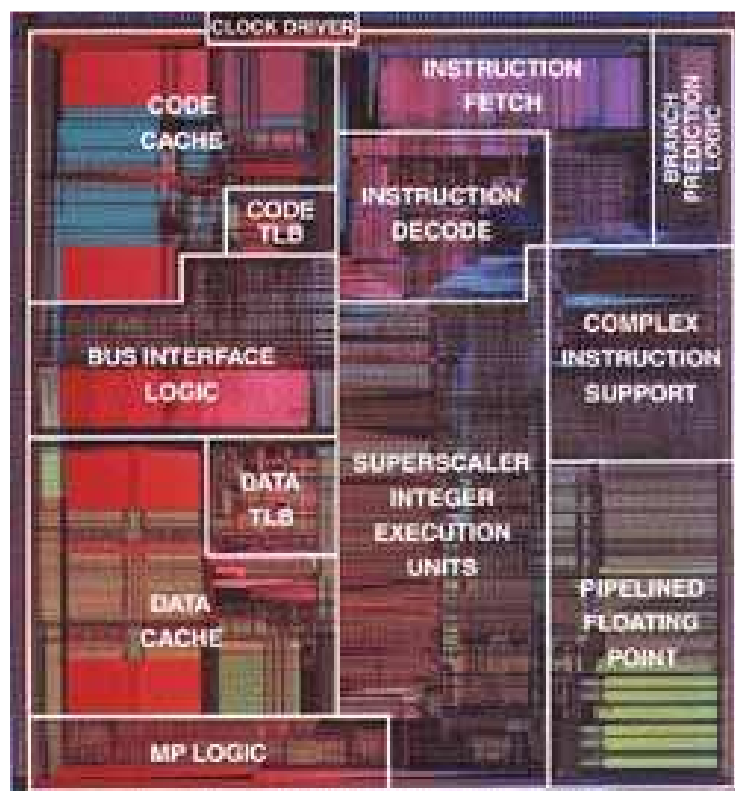


1- placer le PC dans Registre adresse et valider la mémoire	Lecture code OP
2- incrémenter le PC	
3- charger registre instruction avec le code OP présent sur le bus de données	
4- décoder l'instruction (il y a 2 opérandes qui sont des adresses)	décodage
5- placer le PC dans le Registre Adresse et valider la mémoire	Lecture première opérande
6- incrémenter le PC	
7- charger le registre temporaire 1 avec l'opérande 1 (0300)	
8- placer le contenu du registre temporaire 1 dans le registre d'adresse et valider la mémoire	Lecture du contenu de l'adresse 0300
9- charger le registre de données avec la valeur présente sur le bus de données	
10- placer le PC dans le Registre Adresse et valider la mémoire	
11- incrémenter le PC	Lecture deuxième opérande
12 - charger le registre temporaire 1 avec l'opérande 2 (4000)	
13- placer le contenu du registre temporaire 1 dans le registre d'adresse	
14- placer le contenu du registre de données sur le bus de données (contenu de l'adresse 300)	Ecriture à l'adresse 4000
15- valider l'écriture de la mémoire à l'adresse 4000	
16 – instruction suivante	

Le cadencement de ces opérations est réalisé avec l'horloge du microprocesseur à une micro-instruction par top d'horloge : (cycle machine)



Processeur Pentium



Chapitre 3

Mécanismes pour augmenter les performances dans les nouvelles architectures

Le microprocesseur est l'un des éléments les plus déterminant dans le fonctionnement d'une machine.

Du choix du microprocesseur va dépendre le niveau de performance de l'ensemble de la machine.

Performances d'un microprocesseur

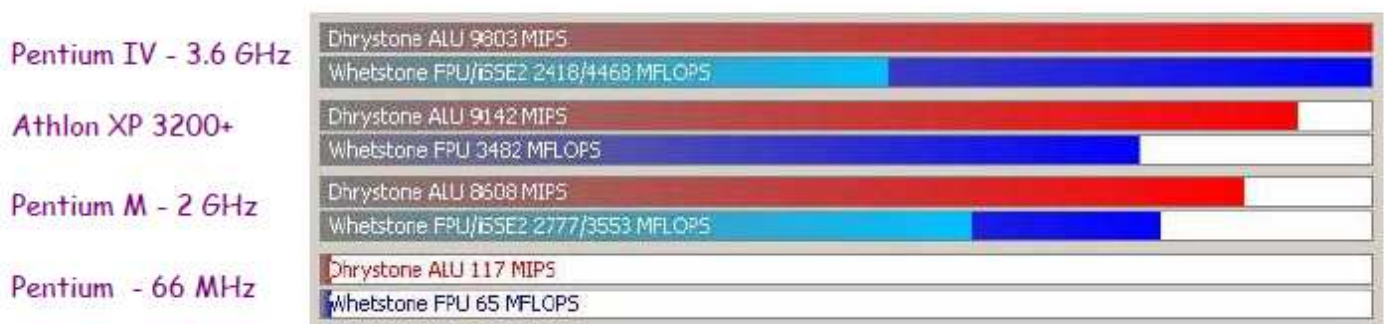
On peut caractériser la puissance d'un microprocesseur par le nombre d'instructions qu'il est capable de traiter par seconde. Pour cela, on définit :

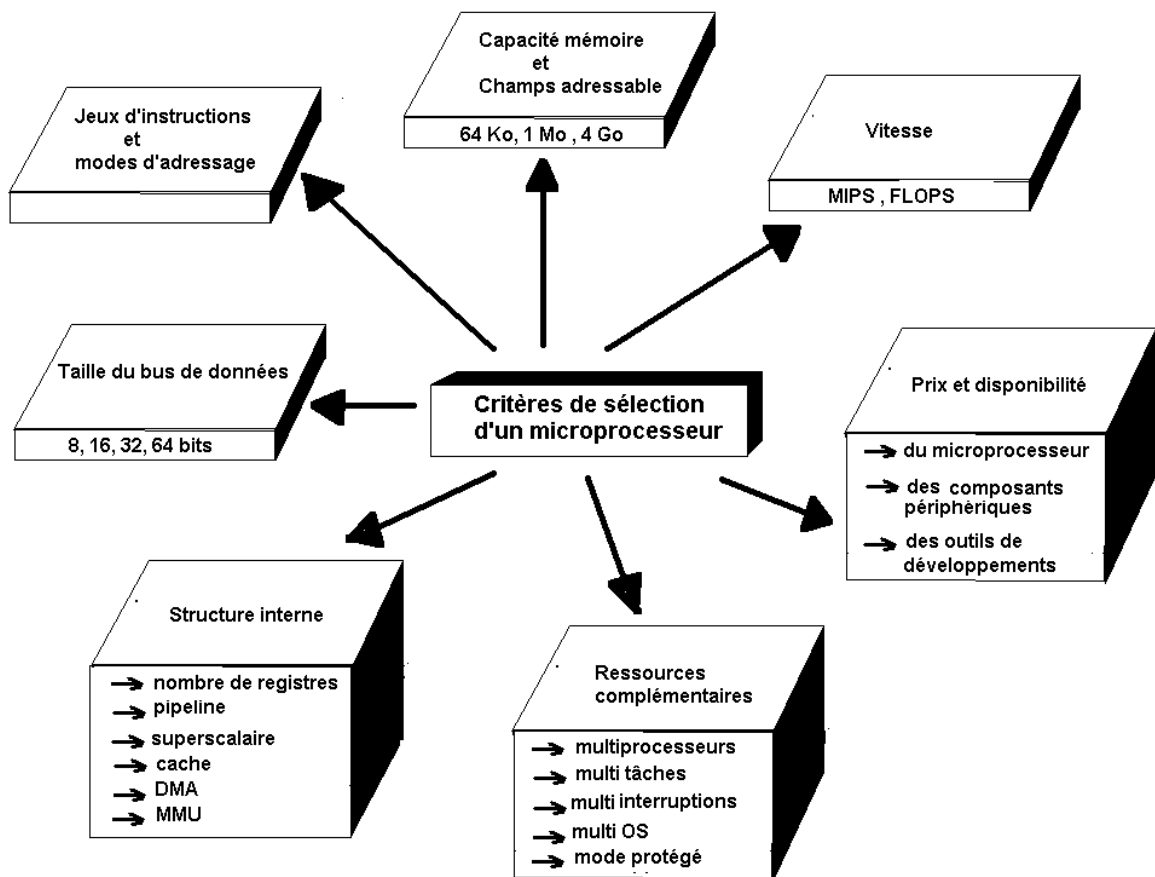
- le CPI (Cycle Par Instruction) qui représente le nombre moyen de cycles d'horloge nécessaire pour l'exécution d'une instruction pour un microprocesseur donné.
- le MIPS (Millions d'Instructions Par Seconde) qui représente la puissance de traitement du microprocesseur.

$$\text{MIPS} = \frac{F_H}{\text{CPI}} \quad \text{avec } F_H \text{ en MHz}$$

- le MFLOPS (Nombre de Millions d'opérations flottantes Par Seconde) qui représente la puissance de calcul du microprocesseur.

Exemple : Performance en MIPS (traitement entier) et MFLOPS (traitement réel) de différents microprocesseurs.



Critères de sélection d'un microprocesseur

Notion d'architecture RISC et CISC

Actuellement l'architecture des microprocesseurs se composent de deux grandes familles :

- L'architecture CISC (Complex Instruction Set Computer)

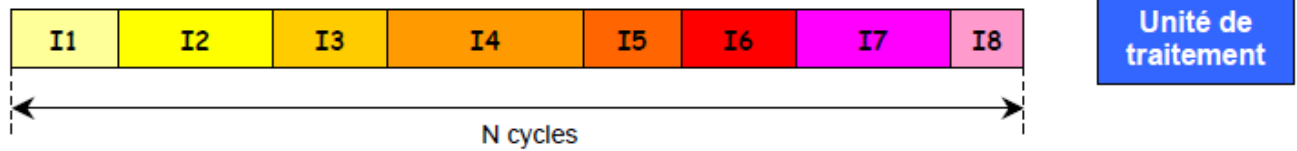
L'architecture RISC (Reduced Instruction Set Computer)

Architecture RISC	Architecture CISC
- instructions simples ne prenant qu'un seul cycle - instructions au format fixe - décodeur simple (câblé) - beaucoup de registres - seules les instructions LOAD et STORE ont accès à la mémoire - peu de modes d'adressage - compilateur complexe	- instructions complexes prenant plusieurs cycles - instructions au format variable - décodeur complexe (microcode) - peu de registres - toutes les instructions sont susceptibles d'accéder à la mémoire - beaucoup de modes d'adressage - compilateur simple

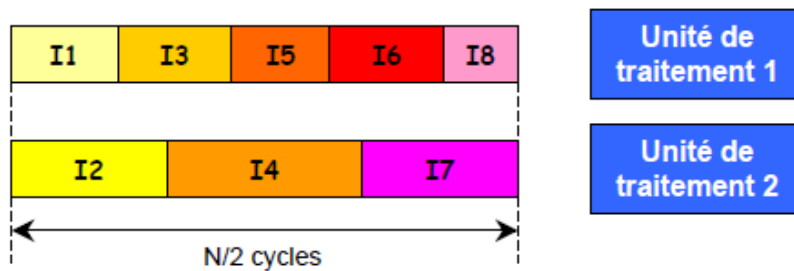
Technologie SUPERSCALAIRE

Cette technologie consiste à faire travailler plusieurs unités d'exécution en parallèle dans le processeur.

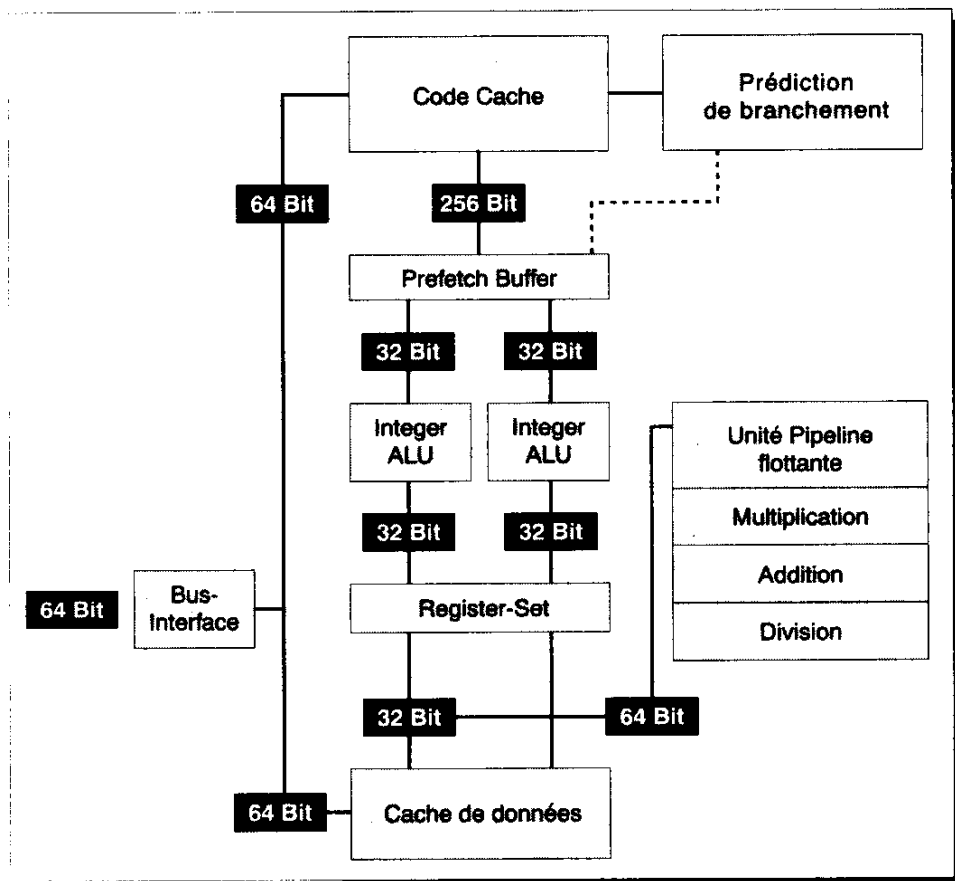
Architecture scalaire :



Architecture superscalaire :



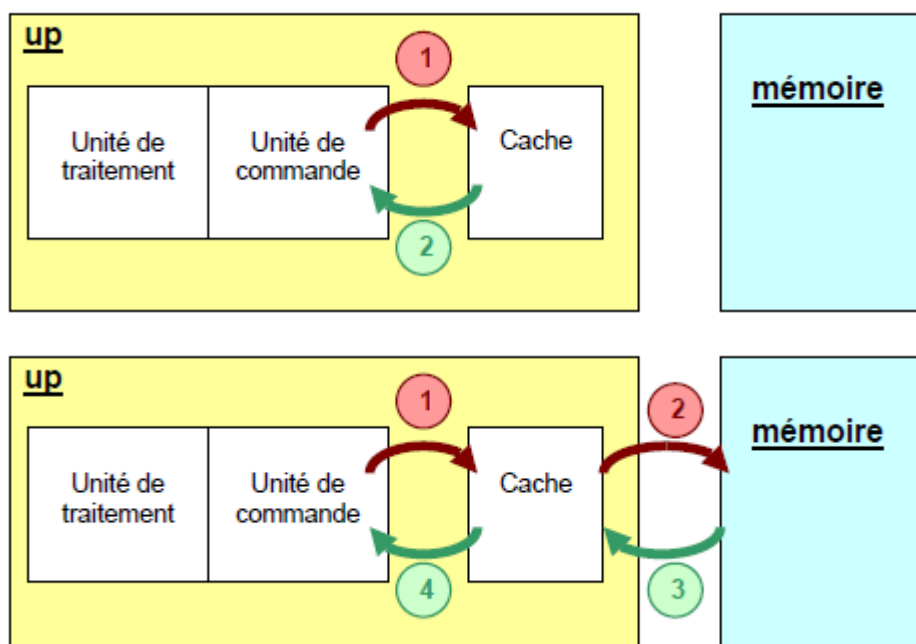
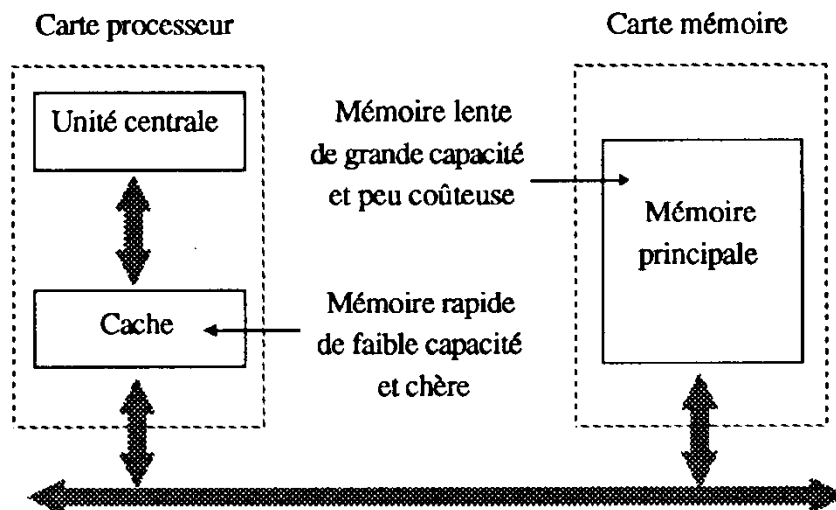
Architecture interne du processeur Pentium



Mécanisme du CACHE

La mémoire cache est une mémoire ultra rapide destinée à éviter les états d'attente (de 6 à 15 ns de temps d'accès contre 60 pour la mémoire vive). Contrairement à la mémoire vive, la mémoire cache (SRAM pour Static RAM) conserve les données qui y sont inscrites. Elle n'a donc pas besoin de rafraîchissement, ce qui évite les états d'attente du processeur.

Les données qui transitent entre la mémoire vive et le processeur sont stockées dans la mémoire cache qui fait en quelque sorte office de tampon. Si le processeur a de nouveau besoin de ces données et cela arrive souvent qu'il fasse deux fois appel aux mêmes, il les obtiendra beaucoup plus rapidement.



Structure du cache des microprocesseurs

Fonctionnement du cache

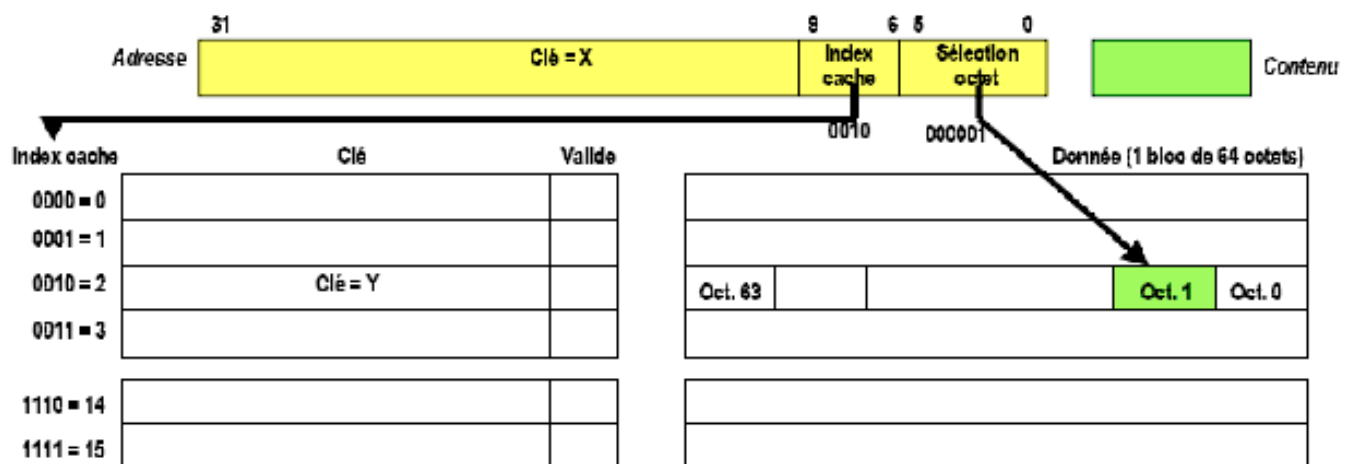
Le cache est un mécanisme purement matériel (**HARDWARE**) qui sera donc totalement transparent pour le logiciel.

Le principe du cache consiste à faire associer à chaque emplacement dans le cache une adresse dans la mémoire centrale. Il existe trois types d'associations possibles :

Association directe (cache direct):

A toute adresse physique peut être associée une entrée unique dans le cache.

- chaque bloc est rangé selon ses bits d'adresse de poids faible
- Ex: cache de 1 Ko = 16 blocs de 64 octets

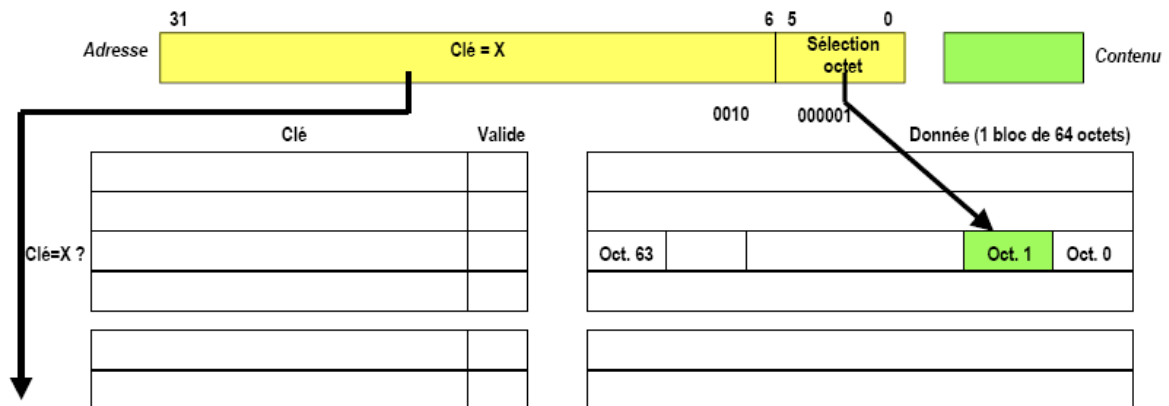


- Lecture du cache: si $X=Y$ et $Valide=1$: **Succès !** (Sinon Echec)
- **Avantage:** décision rapide (on cherche directement à la ligne donnée par l'Index)
- **Inconvénient:** *ping-pong* (impossible d'avoir dans le cache 2 blocs de même index)

Association totale (cache associatif) :

Elle consiste à permettre à n'importe quelle adresse physique d'occuper n'importe quelle entrée dans le cache.

- chaque bloc est rangé n'importe où
- Ex: cache de 1 Ko = 16 blocs de 64 octets

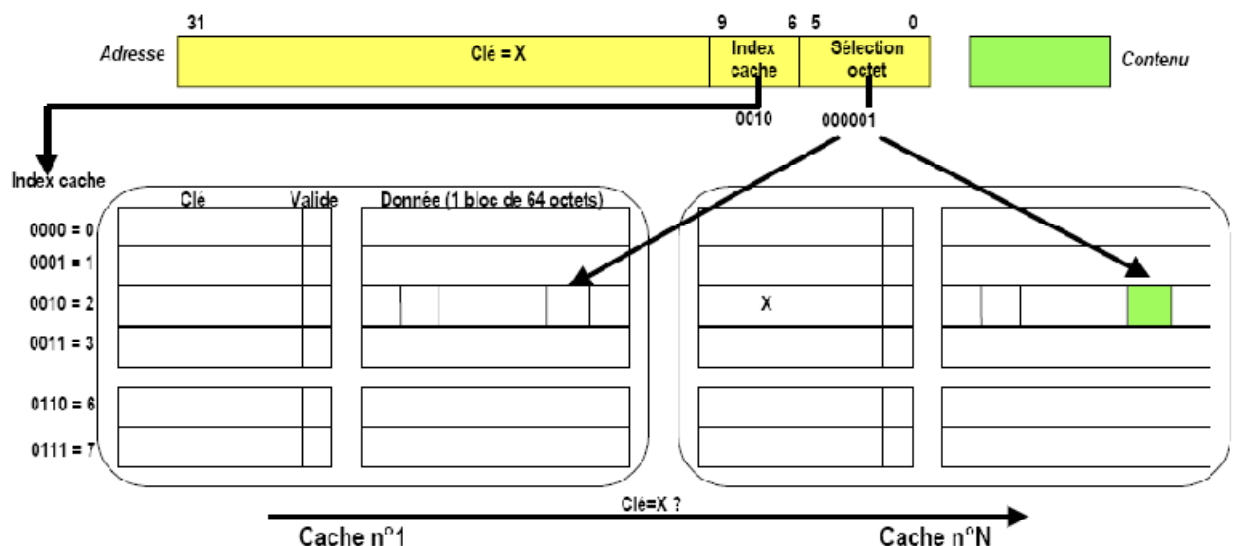


- Lecture du cache: si X est dans le cache et Valide=1 : Succès ! (Sinon Echec)
- Avantage: plus de *ping-pong*
- Inconvénient: décision plus lente (il faut parcourir toutes les clés du cache)

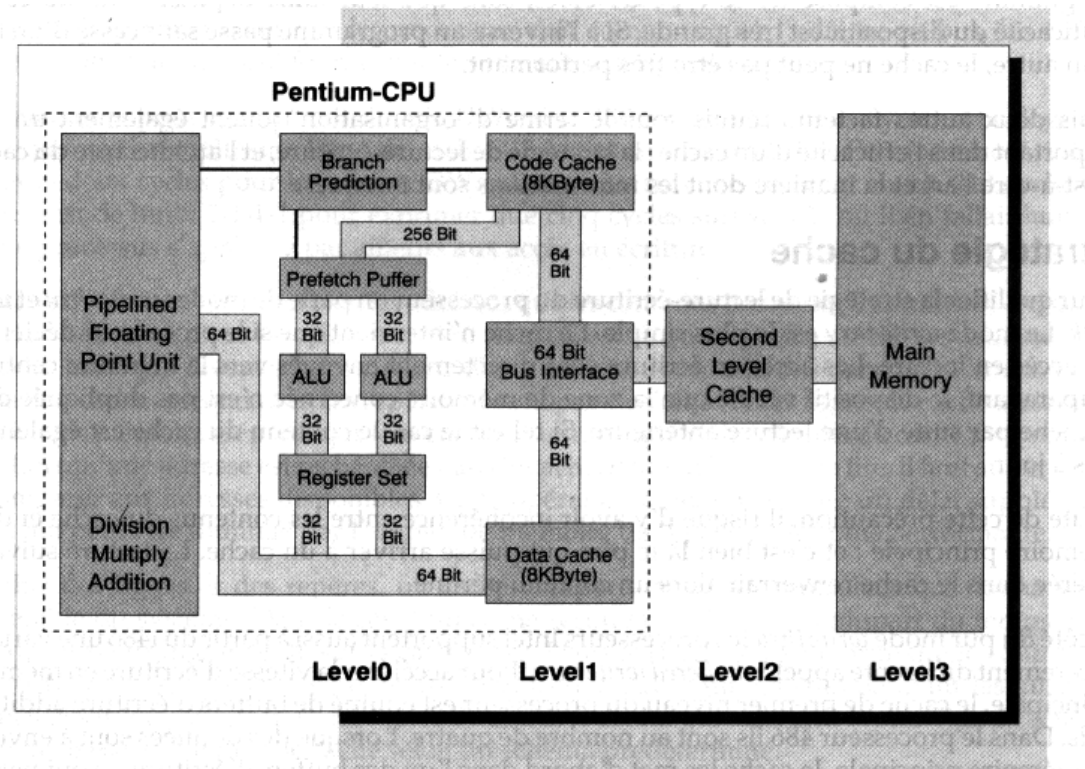
Association partielle (cache associatif à N voies)

Elle consiste à diviser le cache en plusieurs groupes opérant chacun de façon directe en parallèle. A chaque adresse physique, pourront correspondre plusieurs entrées dans le cache.

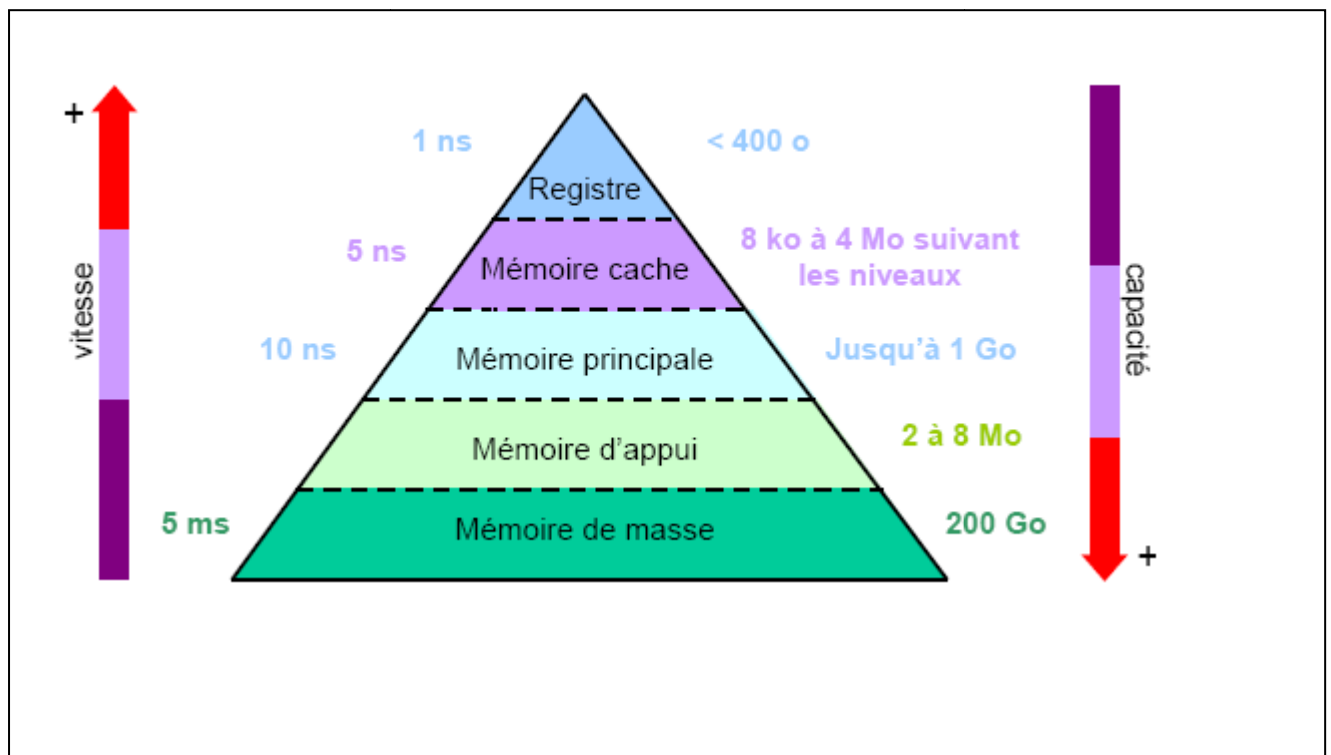
- = N caches directs fonctionnant en parallèle
- Ex: cache de 1 Ko = 16 blocs de 64 octets



Structure du cache des microprocesseurs PENTIUM



Caches de premier, deuxième et troisième niveau.



Mécanisme du DMA

Avec un contrôleur d'entrée sortie qui peut accéder directement à la mémoire principale, le passage des données par le processeur est supprimé.

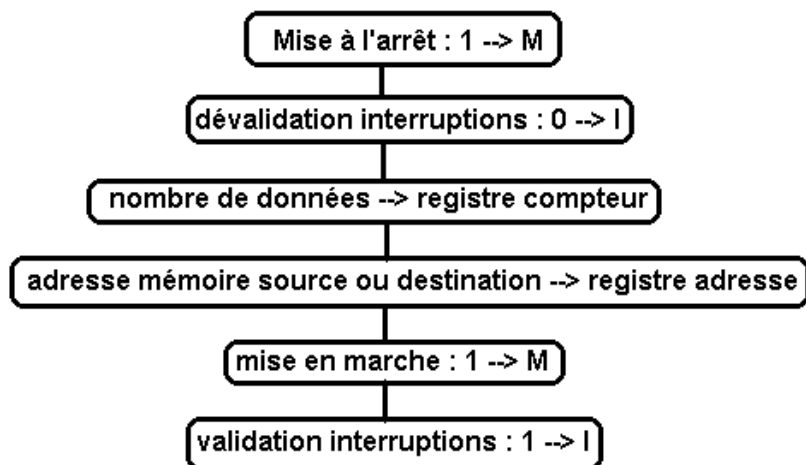
On parle alors de la technique d'accès direct à la mémoire (DMA : Direct Memory Access)

La gestion de ce processus est assurée par un contrôleur spécialisé : Contrôleur DMA.

Architecture interne d'un contrôleur DMA : un contrôleur DMA offre aux circuits d'entrées sorties plusieurs canaux de communication (canaux DMA). Chaque canal étant composé de quatre registres :



Programme de commande :



Canaux DMA du système IBM PC

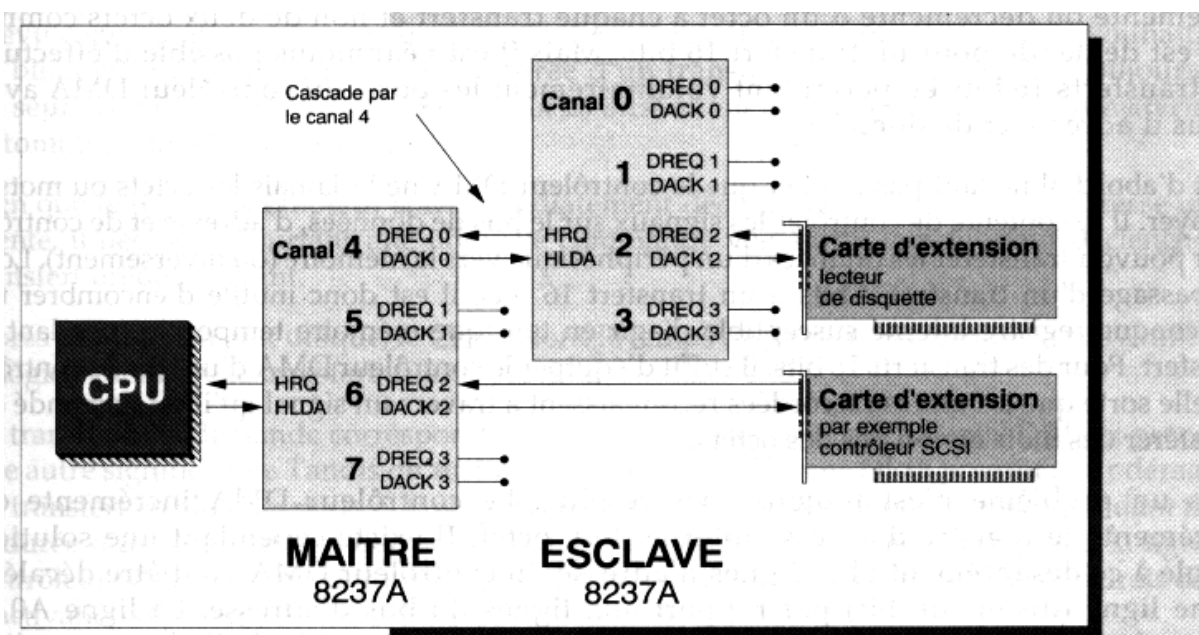
Les canaux DMA sont gérés par un contrôleur intégré à la carte mère dans le Chipset.

Le tableau ci-dessous donne les valeurs classiques des canaux DMA :

	État	Usage habituellement constaté
Canal 0	Libre	Carte réseau, carte graphique, scanner à main...
Canal 1	Libre	Cartes son - son 8 bits
Canal 2	Réservé	Lecteur de disquettes
Canal 3	Libre	Port imprimante
Canal 4	Libre	Contrôleur mémoire, scanner à main...
Canal 5	Libre	CD-ROM, scanner à main, Cartes son - son 16 bits...
Canal 6	Libre	CD-ROM, carte son...
Canal 7	Libre	CD-ROM, carte son...
Canal 8	Libre	

Les canaux DMA

On peut multiplier le nombre de canaux DMA par la mise en cascade de plusieurs contrôleurs DMA :



Mise en cascade de deux contrôleurs DMA dans un AT

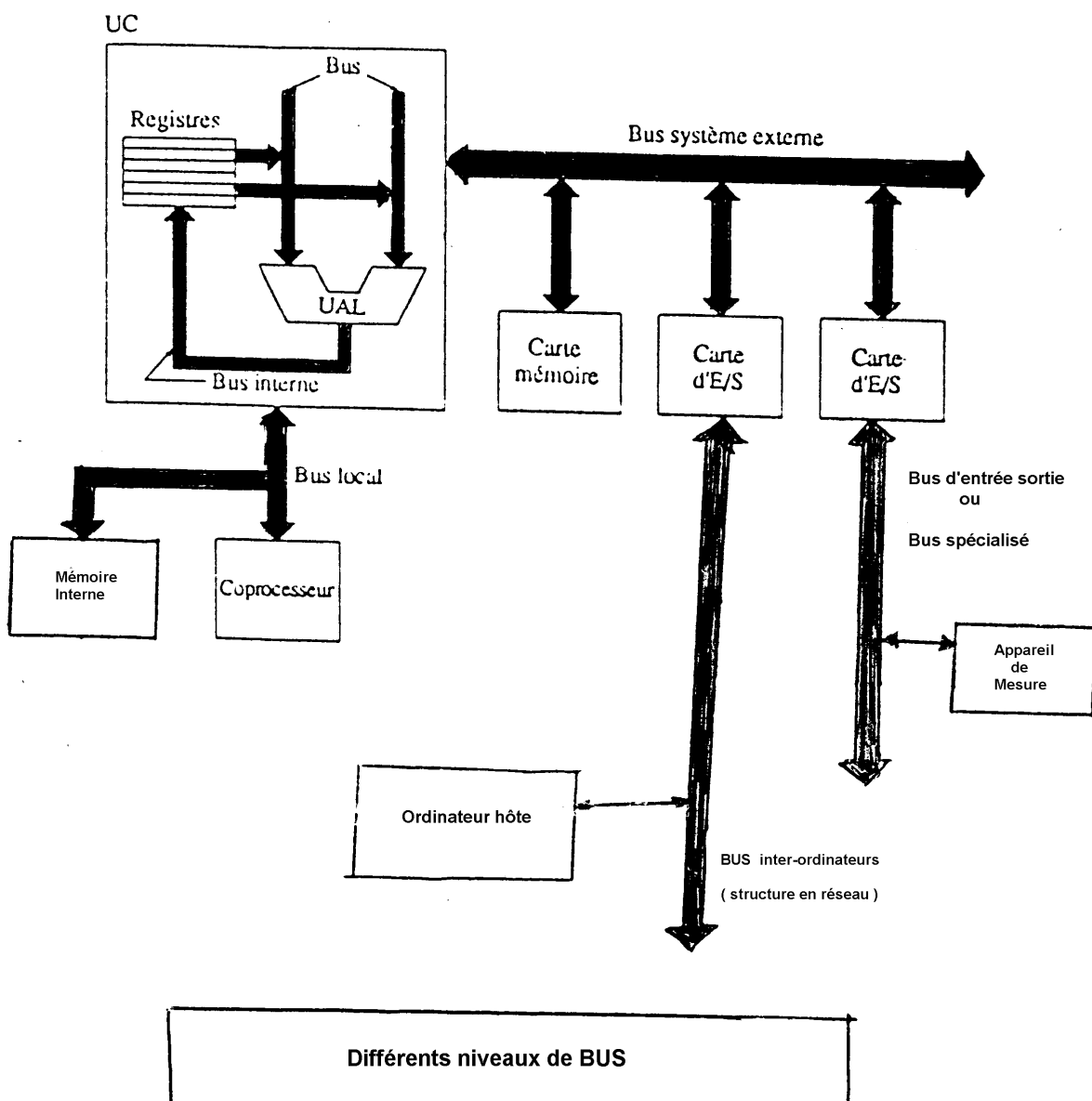
Chapitre 4

Les BUS d'ordinateurs

Un bus est une structure d'interconnexion raccordant plusieurs circuits ou unités d'un ordinateur.

Dans une machine, on trouve plusieurs niveaux de bus :

- Bus interne (Bus processeur)
- Bus local
- Bus global ou Bus système
- Bus spécialisés (Bus d'entrée sortie)
- Bus inter-ordinateur (ou réseau)



Signaux d'un BUS :

On peut classer les signaux véhiculés sur un bus suivant la nature de ces signaux :

- signaux de contrôle :
 - pour l'arbitrage de l'accès au bus
 - pour les interruptions
- signaux de transfert
 - pour le protocole
 - pour les adresses
 - pour les données

Paramètres d'un BUS :

Les paramètres essentiels de conception d'un BUS sont :

- nombre de ligne d'adresses
- nombre de lignes de données
- cadencement des échanges
- rapidité des transferts
- gestion de l'accès au bus
- gestion des interruptions
- contrôle des erreurs

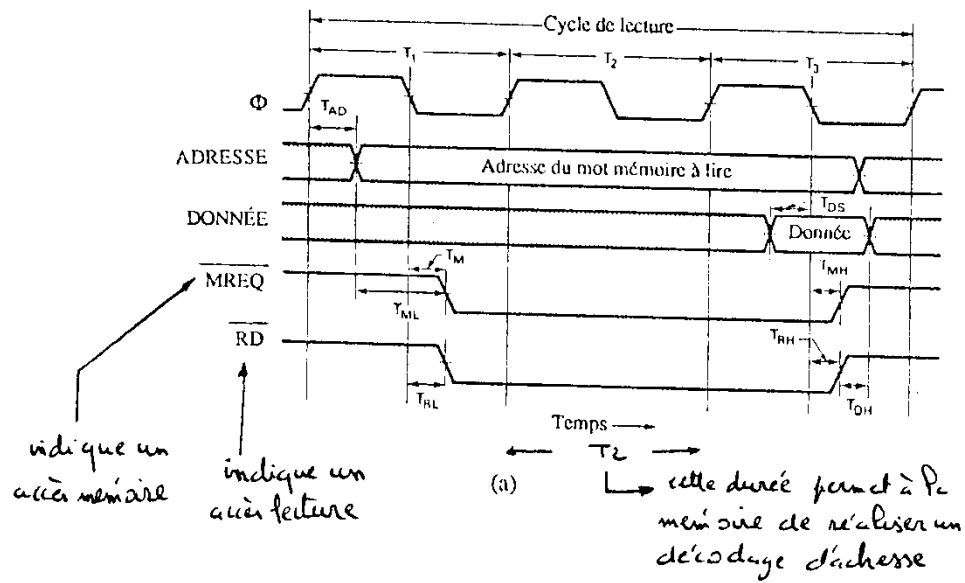
Types de BUS :

Type de cadencement :

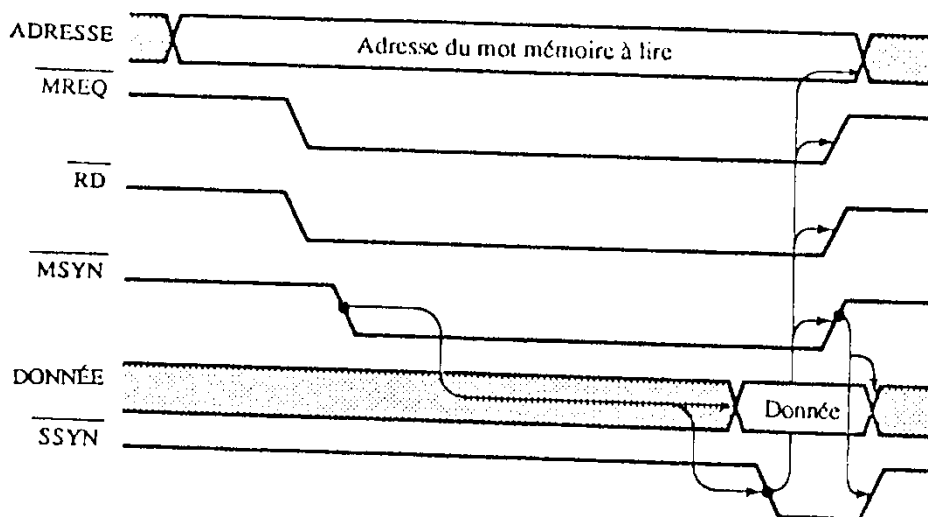
- Bus synchrone
- Bus asynchrone

Format des données :

- Bus série
- Bus parallèle



Cycle de lecture sur un BUS synchrone



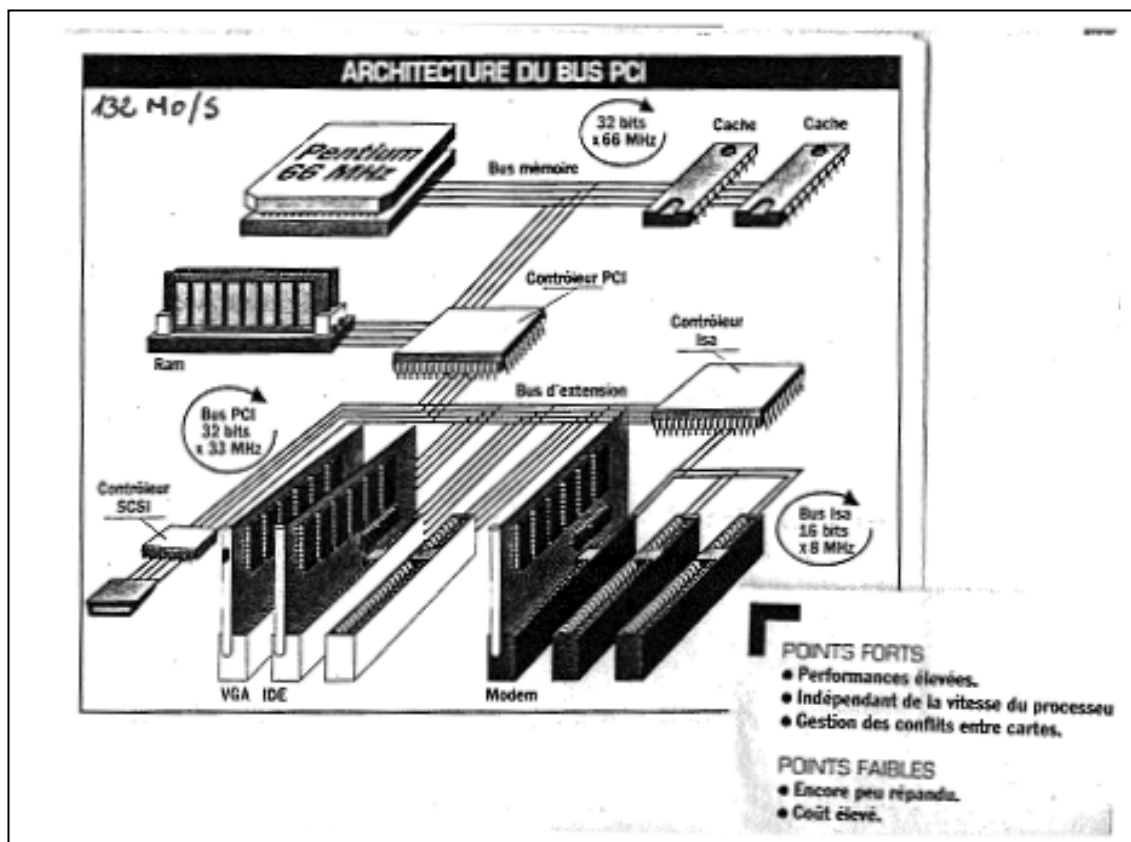
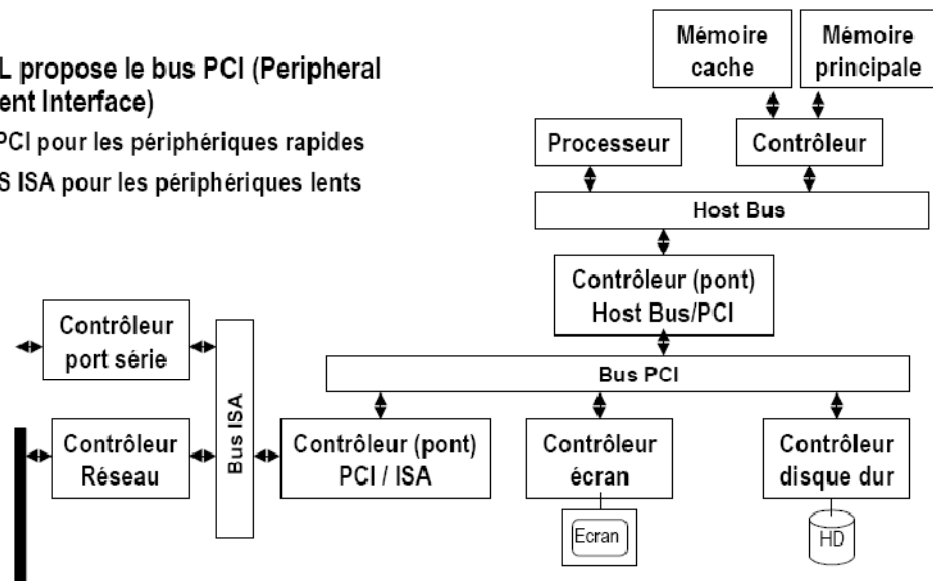
Cycle de lecture sur un BUS asynchrone

Exemples de Bus

Bus : PCI

1993 : INTEL propose le bus PCI (Peripheral Component Interface)

- Bus PCI pour les périphériques rapides
- + BUS ISA pour les périphériques lents



Le bus PCI (Peripheral Component Interconnect) a été développé par Intel en 1993, concurremment à la norme VESA. Il offre, dans sa version 1.0, un bus de 32 bits fonctionnant à 33 MHz ce qui permet d'atteindre un taux de transfert de 132 Mo/s comme

avec un bus VESA. PCI offre l'avantage d'être autoconfigurable, les cartes connectées étant automatiquement détectées et exploitées au mieux. C'est le Plug and Play qui évite donc d'avoir à déplacer des cavaliers sur la carte ou encore d'avoir à configurer les numéros d'interruptions IRQ ou les canaux DMA utilisés par la carte. Dans sa spécification 2.0, PCI présente de nouveaux connecteurs courts et surtout autorise l'accès 64 bits nécessaire à l'exploitation des Pentium.

Type	Largeur	Fréquence	Nombre max. de connecteurs	Taux de transfert
ISA	16 bits	8-12.5 MHz	8	8 Mo/s
EISA	32 bits	8-12.5 MHz	18	50 Mo/s
MCA	32 bits	8-12.5 MHz	8	50 Mo/s
VESA	32 bits	33 MHz	3	132 Mo/s
PCI	32 bits	33 MHz	10	132 Mo/s

Caractéristiques des bus

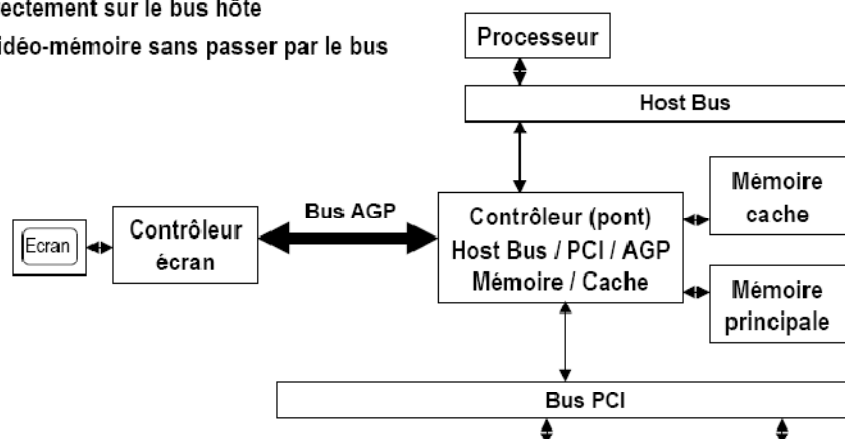
Le BUS AGP

Bus : AGP

1998 : INTEL propose le bus AGP (Accelerated Graphical Port)

Avec l'arrivée du multimédia, l'écran devient un périphérique «couteux» en opérations et en mémoire (films, animations, ...)

- vidéo directement sur le bus hôte
- liaison vidéo-mémoire sans passer par le bus



Le bus AGP (Accelerated Graphics Port) est un bus récent - 1997 - spécialisé dans l'affichage. Il relie directement - au travers du chipset - le processeur de la carte graphique avec le processeur de l'UC et avec la mémoire vive. Il offre un bus de 32 bits, un fonctionnement en mode pipeline ce qui autorise des lectures et écritures simultanées en

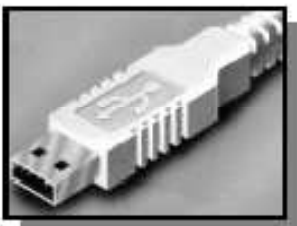
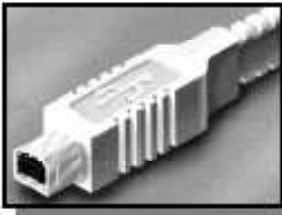
mémoire, des débits atteignant 528 Mo/s (32 bits à 66 MHz) dans la version AGP 2x et la possibilité d'accéder également à la mémoire centrale en sus de la mémoire de la carte graphique. On peut ainsi manipuler des images « lourdes » (affichage tridimensionnel 3D), sans saturer la mémoire de la carte graphique, puisqu'on peut placer une partie de l'image en mémoire centrale.

La version de base AGP offre un débit de 264 Mo/s, deux fois supérieur à celui du bus PCI (132 Mo/s). AGP 2x offre donc un débit de 528 Mo/s et AGP 4x -destiné à tirer partie au maximum des instructions des processeurs

Le BUS USB

- ✓ USB signifie Universal Serial Bus
- ✓ Le USB a été conçu afin de remplacer le port série
- ✓ Il existe deux normes USB principales: USB1.1 (1998) et USB2.0 (2000, révisé en 2002).
- ✓ Le USB supporte 3 vitesses de communication:
 - Basse vitesse à 1.5 Mb/s pour les appareils comme les claviers, souris, joystick
 - Pleine vitesse 12 Mb/s pour des appareils à gros débits comme les modems et téléphones.
 - Haute vitesse 480 Mb/s pour des appareils multimédia à très haut débit comme les caméoscopes (une caméra numérique).
- ✓ Les transactions USB se font à l'aide de jetons, c'est-à-dire à l'aide de bytes transmis ayant des valeurs et significations précises.
- ✓ Le port USB est totalement contrôlé par un contrôleur unique appelé hôte (host). L'hôte, souvent le PC, initie toutes les communications (même les interruptions). Il est le maître absolu du bus.
- ✓ Les jetons et messages transitent dans des trames de 1ms ou de 125us (haute vitesse) (voir plus loin).
- ✓ Le bus USB peut fournir l'alimentation des appareils qui y sont connectés.
- ✓ 4 modes de transfert sont supportés: Contrôle, Interruptions, Bloc et Isochrone. Ils sont tous détaillés plus loin.
- ✓ Les appareils nouvellement branchés sur le port USB sont détectés automatiquement.

Bus USB : description matérielle

Series "A" Connectors	Series "B" Connectors
♦ Series "A" plugs are always oriented upstream towards the <i>Host System</i>  "A" Plugs (From the <i>USB Device</i>) "A" Receptacles (Downstream Output from the <i>USB Host or Hub</i>) 	♦ Series "B" plugs are always oriented downstream towards the <i>USB Device</i>  "B" Plugs (From the <i>Host System</i>) "B" Receptacles (Upstream Input to the <i>USB Device or Hub</i>) 

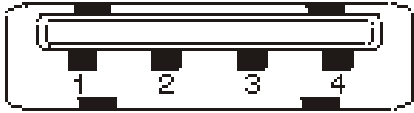
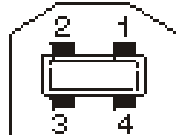



Table 6-1. USB Connector Termination Assignment

Contact Number	Signal Name	Typical Wiring Assignment
1	VBUS	Red
2	D-	White
3	D+	Green
4	GND	Black
Shell	Shield	Drain Wire

- ✓ Le câble USB est constitué de 4 fils: Vbus, GND, D+ et D-. Vbus et GND sont l'alimentation 5Vdc (entre 4.75V et 5.25V) et la référence électrique. D+ et D- servent au transport des données.
- ✓ Il y a deux types de connecteurs USB tel que montré à droite.
- ✓ Plus la fréquence est élevée, plus les câbles doivent être courts. En haute vitesse la longueur max. d'un câble USB est 5m.
- ✓ La puissance maximum pouvant être tirée d'une connection USB est $5 \times (5V \times 100mA)$, soit 2.5W. L'unité de courant de base est 100mA.
- ✓ Les lignes de données D+ et D- fonctionnent en mode différentiel :

On définit sur les lignes de données deux états : J et K.

En basse vitesse :

Etat J : D- > D+ d'au moins 200 mV

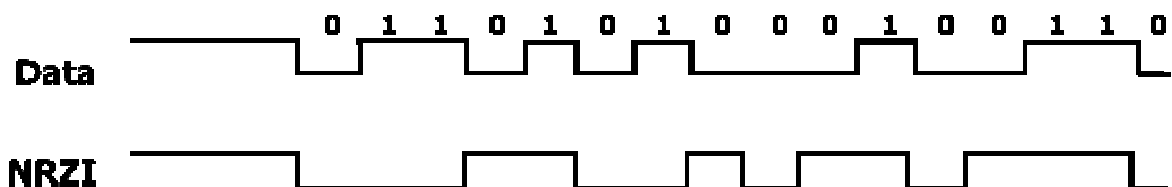
Etat K : D+ > D- d'au moins 200 mV

En pleine vitesse c'est le contraire.

Un état J avec certaines valeurs minimales et maximales de tension indique un état "innocué".

De plus il existe une état où D+ et D- sont au potentiel bas appelé SE0. L'état SE0 pendant plus de 10 ms indique un reset, il est émis vers un appareil 100 ms après sa connexion. Dès qu'un appareil voit le signal de reset pendant plus de 2,5 microseconde, il doit faire son reset et le terminer avant la fin du signal(10 ms).

L'encodage des données se fait selon la méthode NRZI. Un "1" est représenté par l'absence de changement d'état et un "0" par un changement d'état.



Chapitre 4

Système d'interruptions

Généralités

Une interruption permet d'arrêter un programme, en cours d'exécution sur le processeur, pour que celui-ci traite une tâche considérée comme plus urgente. Quand cette tâche est terminée, le processus interrompu doit alors être repris en l'état où il avait été laissé.

Les interruptions permettent donc à des événements, en général externes au microprocesseur, tels que coupures d'alimentation, alarmes, périphériques prêts à reémettre ou à recevoir des données..., d'attirer immédiatement l'attention de l'unité centrale.

Dans la mesure où elle est acceptée par le processeur, l'interruption permet ainsi au circuit périphérique ou au logiciel de suspendre le fonctionnement de ce microprocesseur d'une manière rationnelle et de lui demander l'exécution d'un sousprogramme de service, dit également sous-programme d'interruption.

Types d'interruptions

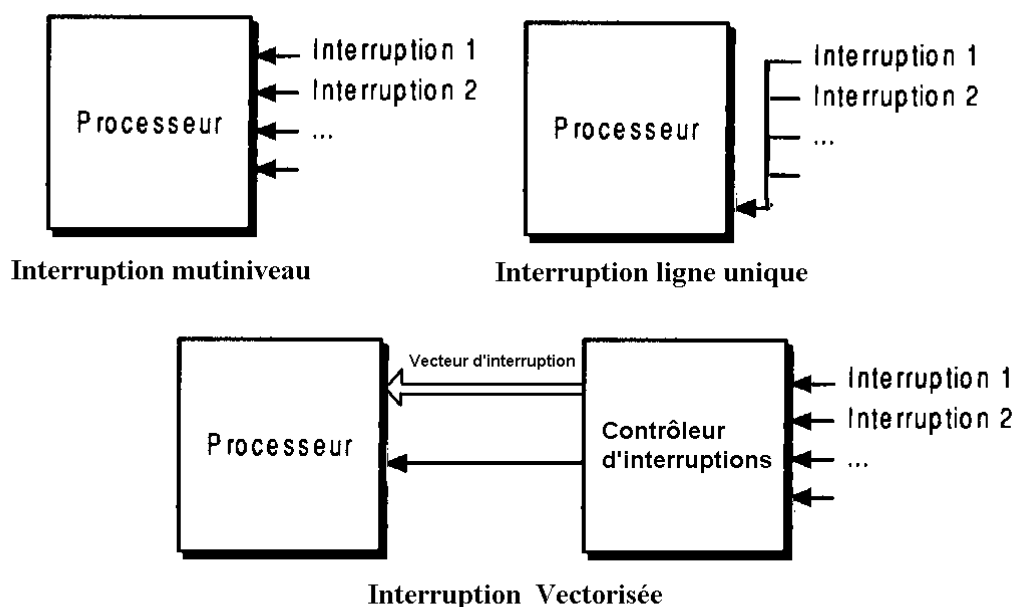
Une interruption peut être provoquée de diverses manières

- par un périphérique, l'interruption est alors dite externe et matérielle,
- par un programme, l'interruption est alors externe et logicielle,
- par le processeur lui-même lors de certains événements exceptionnels, l'interruption est alors dite interne et appelée exception (Trap).

Reconnaissance des interruptions

Il existe divers moyens physiques pour déterminer la source d'une interruption - aussi notée IRQ (*Interruptio* *ReQuest*) - et donc y répondre de manière appropriée.

- Interruption multiniveau
- Interruption ligne unique
- Interruption vectorisée



Traitement des interruptions

Actions réalisées par des circuits logiques

1. Le contrôleur de gestion du terminal active un signal de demande d'interruption sur le bus du système afin d'entreprendre une séquence d'interruption.
2. Dès que l'UC est prête à répondre à la demande, elle active à son tour un signal du bus signifiant l'acquittement de la demande d'interruption.

3. Lorsque le contrôleur du terminal reçoit l'acquittement de sa demande d'interruption, il place sur le bus données un nombre particulier destiné à permettre son identification. Ce nombre est appelé vecteur d'interruption.
4. L'UC enregistre le vecteur d'interruption dans un tampon interne.
5. L'UC range dans la pile le compteur ordinal et le registre d'état courant.
6. L'UC charge une nouvelle valeur dans le compteur ordinal à partir du vecteur d'interruption fourni par le terminal (en mode d'adressage indexé). Le vecteur d'interruption agit comme un index dans une table située en mémoire à une adresse fixe et connue. Supposons que le compteur ordinal contienne par exemple quatre octets, alors le vecteur d'interruption n correspond à l'adresse $4n$. Cette fois le compteur ordinal pointe vers la routine de traitement de l'interruption associée au terminal ayant généré la demande.

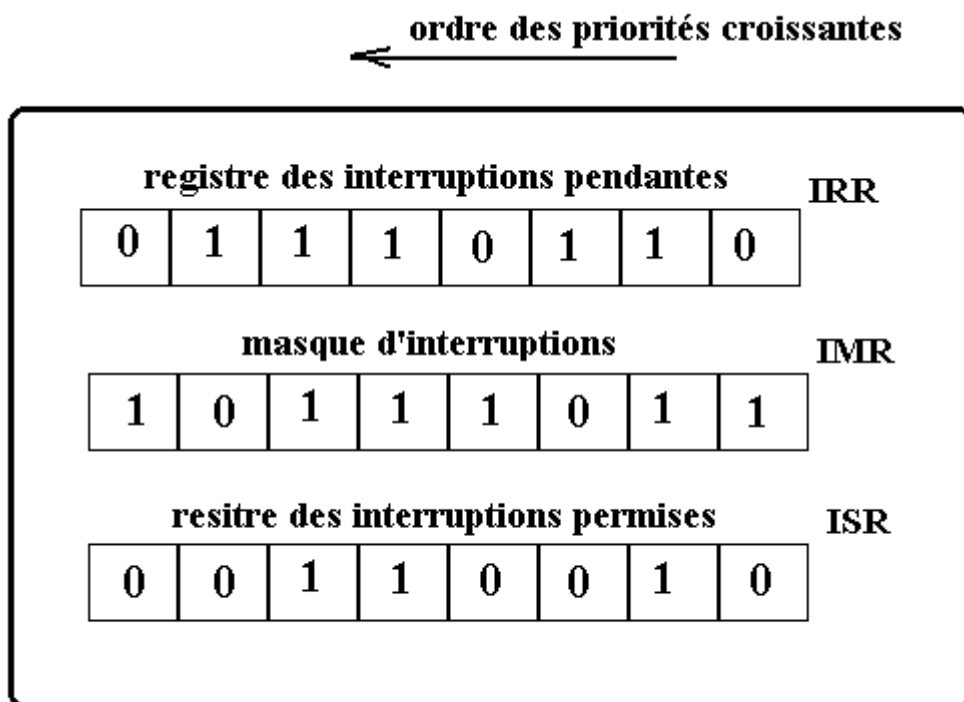
Actions réalisées par programme

7. La première phase de la routine d'interruption consiste à sauvegarder l'état courant des registres de l'UC dans la pile pour les restituer à la fin de la routine d'interruption.
8. Chaque vecteur d'interruption étant la plupart du temps partagé par un ensemble de terminaux de même type, le processeur ne connaît toujours pas le terminal qui a causé la demande. Pour le savoir il doit lire les registres d'état des terminaux partageant le même vecteur.
9. Des informations complémentaires relatives à la demande d'interruption sont obtenues en examinant les bits du registre d'état du terminal.
10. S'il s'agit d'une erreur ou d'une anomalie de fonctionnement dans la phase de transfert de données, par exemple, elle peut être traitée à cet endroit.
11. Si nécessaire, une information particulière est fournie au contrôleur ou au terminal pour lui signifier que la routine d'interruption est terminée. C'est le cas par exemple avec le circuit d'E/S 8259A.

12. La routine d'interruption étant terminée, il y a restauration des registres précédemment sauvegardés.

13. L'exécution de l'instruction de fin de routine d'interruption, par exemple RTI, a pour effet de restituer les valeurs du compteur ordinal et du registre d'état telles qu'elles étaient lorsque la demande d'interruption est apparue. Puis le programme interrompu reprend son exécution comme si rien ne s'était passé.

Principe D'un Contrôleur D'interruptions



Cas du contrôleur d'interruptions 82830

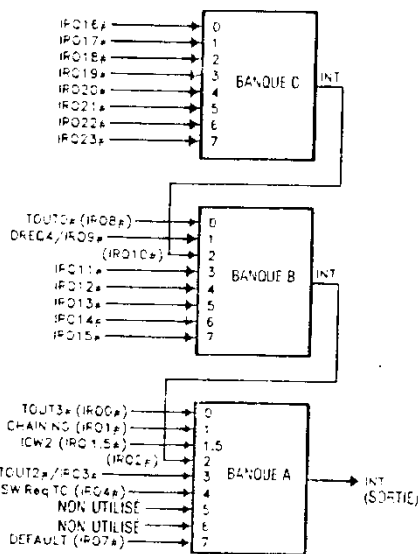


Fig. 4-8 Bloc diagramme du contrôleur d'interruption.

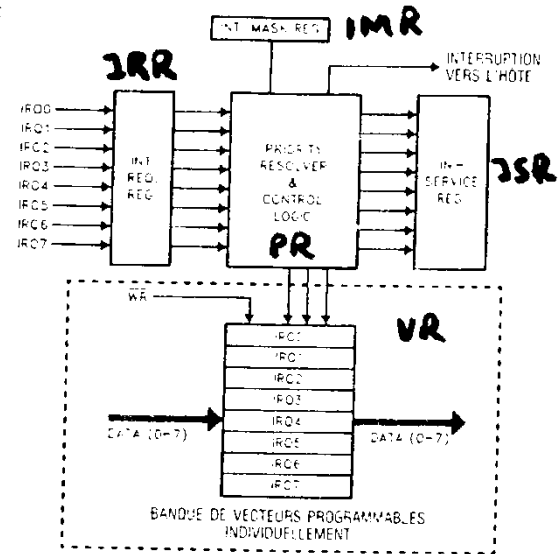
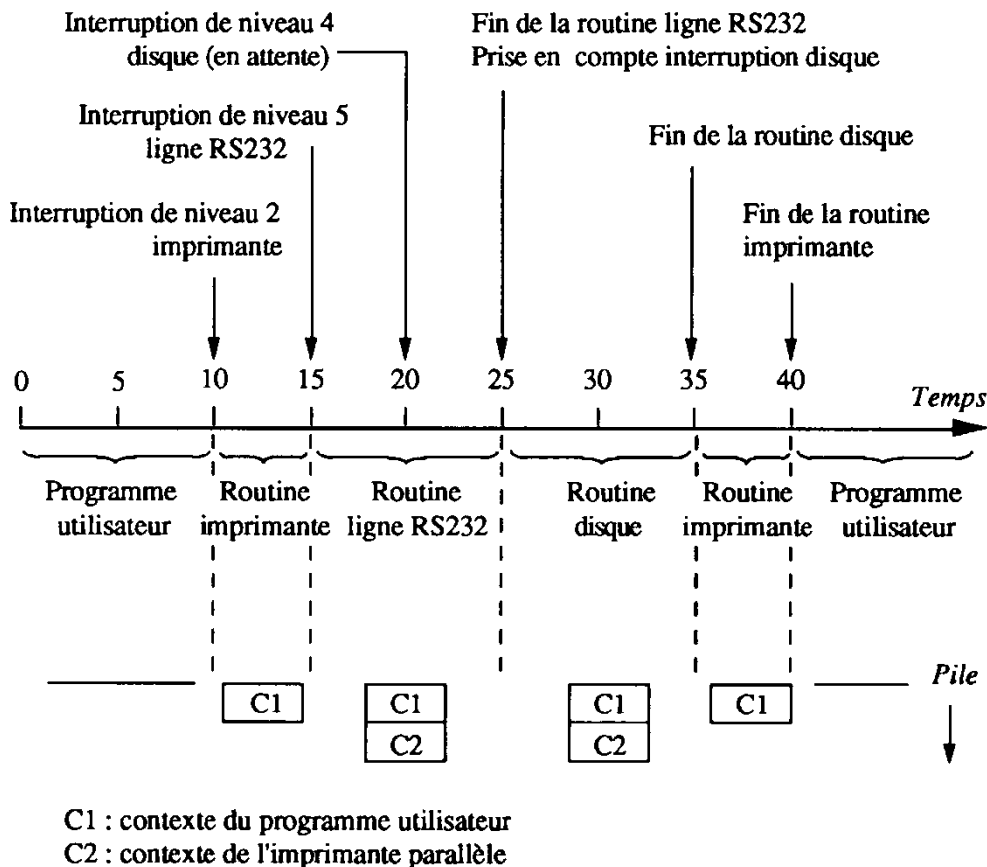


Fig. 4-9 Les 6 unités d'une banque.

Exemple de déroulement d'une suite d'interruptions à plusieurs niveaux de priorité :



Affectation des vecteurs d'interruptions du système IBM PC

	Assignation fixe	Assignation habituellement constatée par ordre de préférence
IRQ 0	Horloge système (Timer)	
IRQ 1	Clavier	
IRQ 2		Carte écran EGA, VGA, contrôleur d'interruptions, scanner, carte son...
IRQ 3		Port série COM2, COM4, modem-fax, CD-ROM...
IRQ 4		Port série COM1, COM3, modem-fax, souris-bus, CD-ROM...
IRQ 5		Port parallèle LPT2, souris-bus, carte réseau, CD-ROM, carte son, scanner à main...
IRQ 6	Contrôleur disquette	
IRQ 7	Port parallèle LPT1 mais "partageable"	Carte réseau, carte son...
IRQ 8	Horloge temps réel	
IRQ 9		IRQ2 redirigée, CD-ROM, carte son, carte graphique
IRQ 10		Carte réseau, CD-ROM, scanner à main, carte son, contrôleur SCSI...
IRQ 11		Carte réseau, CD-ROM, scanner à main, carte son, contrôleur SCSI...
IRQ 12		Carte réseau, CD-ROM, port souris...
IRQ 13	Coprocasseur	
IRQ 14	1° Contrôleur disque dur	
IRQ 15		Carte réseau, CD-ROM, 2° contrôleur disque dur...

Affectation habituelle des IRQ

Chapitre 5 : La mémoire

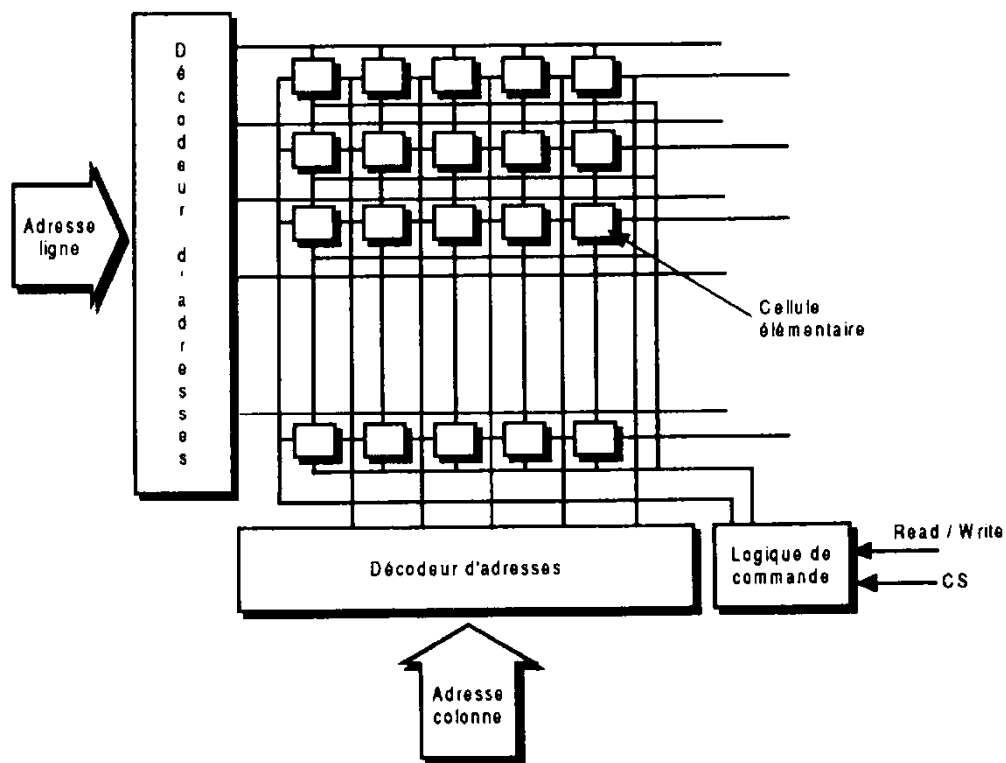
Partie 1 : description de la mémoire

Organisation interne et brochage des boîtiers mémoires vives

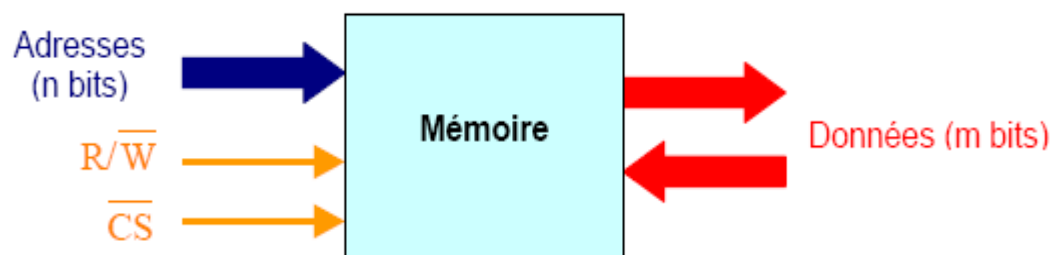
La mémoire est organisée en blocs de un ou plusieurs tableaux de bits de L lignes et C colonnes dont l'accès est piloté par un contrôleur qui sert d'interface entre le microprocesseur et la mémoire.

Chaque location de mémoire est identifiée à l'aide d'une adresse.

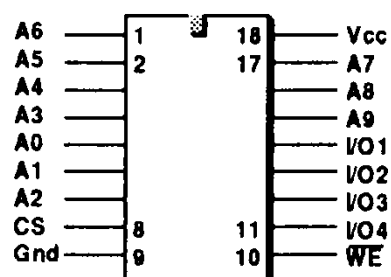
MÉMOIRE							
Adresse	Contenu						
	7	6	5	4	3	2	1 0
000000	0	0	1	1	0	0	1 0
00000 1	1	1	0	0	0	1	1 0
00000 2	1	0	0	1	1	1	0 0
00000 3	1	1	1	0	0	0	0 1
00000 4	0	1	1	1	1	0	0 1
00000 5	1	0	0	0	0	1	1 1
00000 6	1	1	1	0	0	0	1 1
00000 7	1	1	1	0	0	0	0 1
00000 8	1	0	0	0	1	1	1 1
00000 9	1	1	0	0	1	1	1 0
00000 A	0	0	0	0	1	1	1 0
00000 B	1	0	0	1	1	1	0 0
.	.						
.							
.							
0FFFFC	0	0	1	1	0	0	0 1
0FFFFD	1	0	0	0	0	1	0 1
0FFFFE	1	1	0	0	0	0	1 1
0FFFFF	0	0	1	1	0	0	0 0



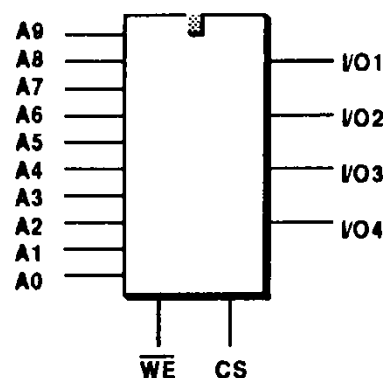
Organisation de la mémoire FPM



- ▶ les entrées d'adresses
- ▶ les entrées de données
- ▶ les sorties de données
- ▶ les entrées de commandes :
 - une entrée de sélection de lecture ou d'écriture. ($\overline{R/W}$)
 - une entrée de sélection du circuit. ($\overline{CS}$)



Brochage Physique



Brochage Logique

Principes de fonctionnement des mémoires vives

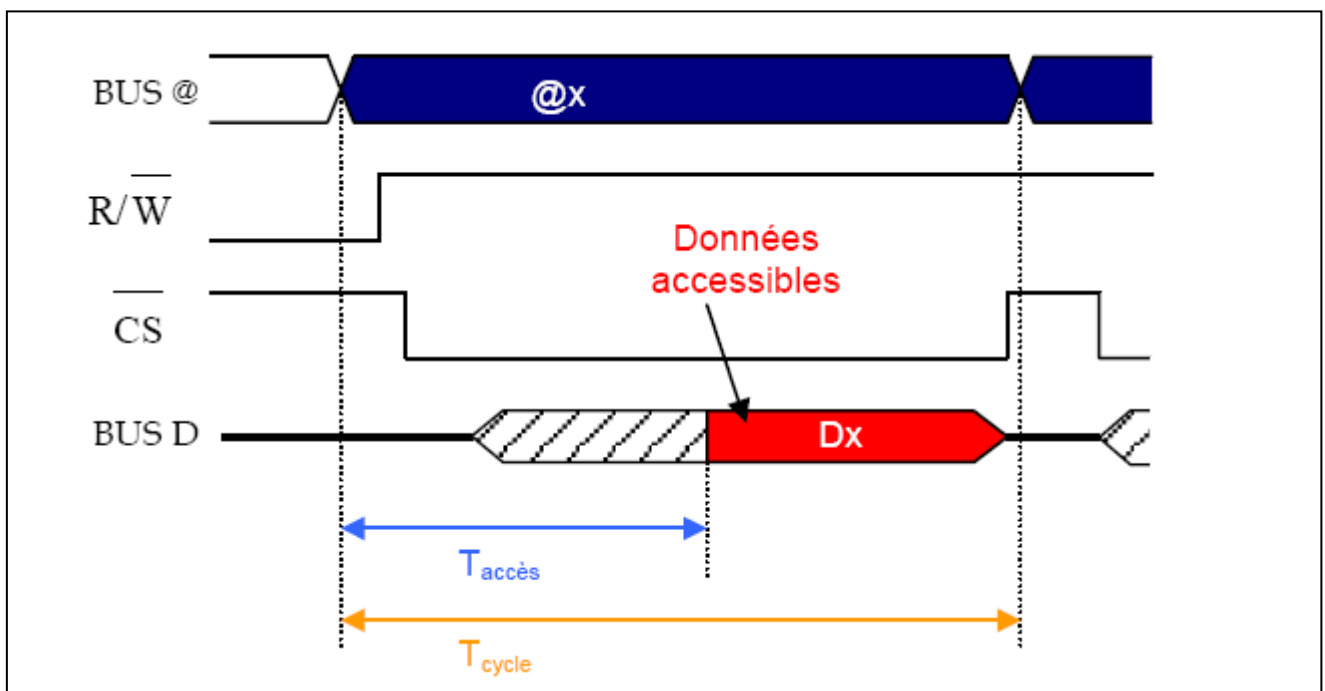
- Ecriture d'une donnée

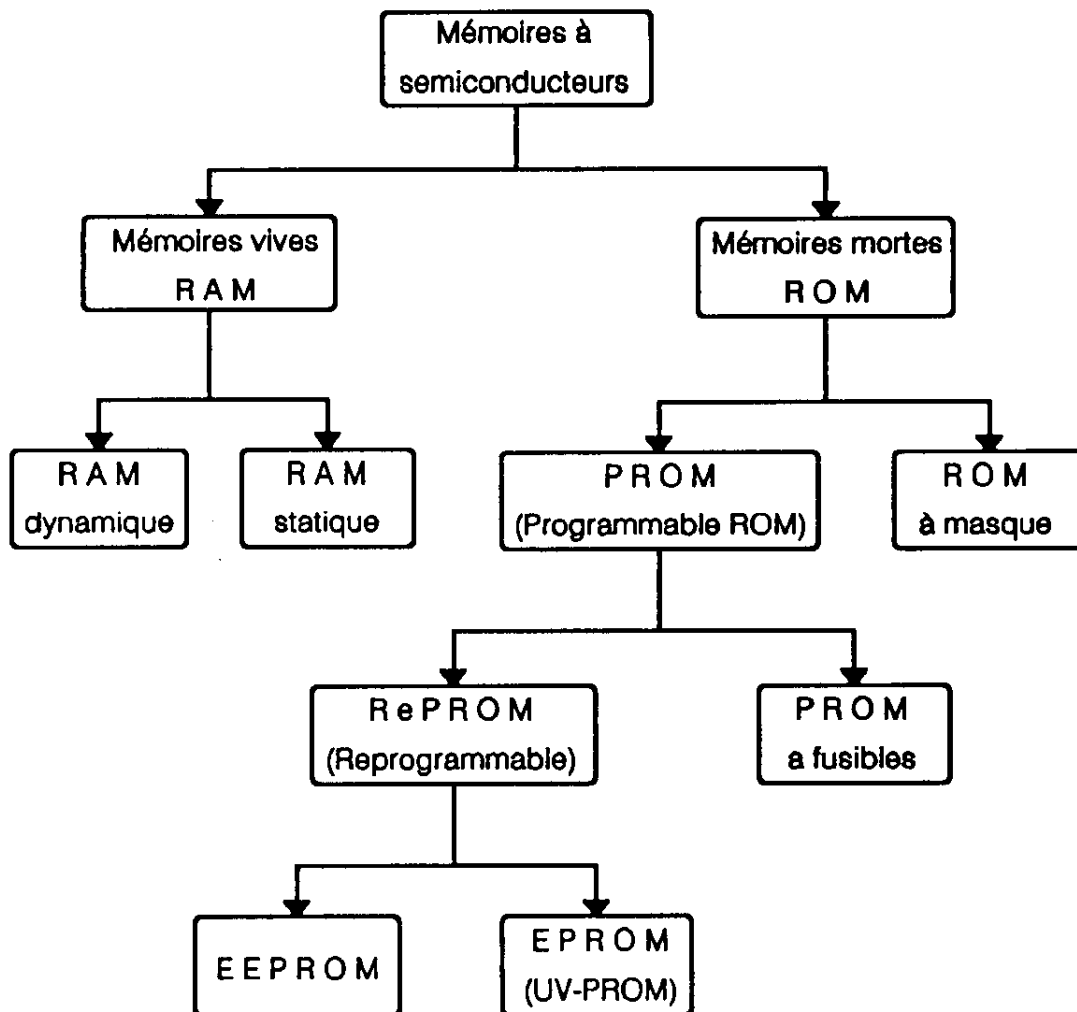
- Sélection de l'adresse mémoire par application de la combinaison binaire désirée sur le bus d'adresses.
- Mise en place de la donnée sur le bus de données.
- Sélection du circuit par activation de la broche CS (*Chip Select*).
- Activation de la commande d'écriture (*Write* ou *R/W*)

- Lecture d'une donnée

- Sélection de l'adresse mémoire par application de la combinaison binaire désirée sur le bus d'adresses.
- Activation de la commande de sélection du boîtier (CS).
- Activation de la commande de lecture (*Read* ou *R/W*).
- Lecture de l'information sur le bus de données.

On peut représenter le fonctionnement de la mémoire sous la forme de chronogramme tel que celui ci-après. Dans cet exemple le chronogramme est celui de la lecture d'une donnée



Classification des boîtiers mémoire

La *mémoire vive*, généralement appelée *RAM* (*Random Access Memory*, traduisez *mémoire à accès aléatoire*), est la mémoire principale du système, c'est-à-dire qu'il s'agit d'un espace permettant de stocker de manière temporaire des données lors de l'exécution d'un programme.

En effet le stockage de données dans la mémoire vive est temporaire, contrairement au stockage de données sur une mémoire de masse telle que le disque dur (mémoire avec laquelle les novices la confondent généralement), car elle permet uniquement de stocker des données tant qu'elle est alimentée électriquement. Ainsi, à chaque fois que l'ordinateur est éteint, toutes les données présentes en mémoire sont irrémédiablement effacées.

La *mémoire morte*, appelée *ROM* pour *Read Only Memory* (traduisez *mémoire en lecture seule*) est un type de mémoire permettant de conserver les informations qui y sont contenues même lorsque la mémoire n'est plus alimentée électriquement. A la base ce type de mémoire ne peut être accédée qu'en lecture. Toutefois il est désormais possible d'enregistrer des informations dans certaines mémoires de type *ROM*.

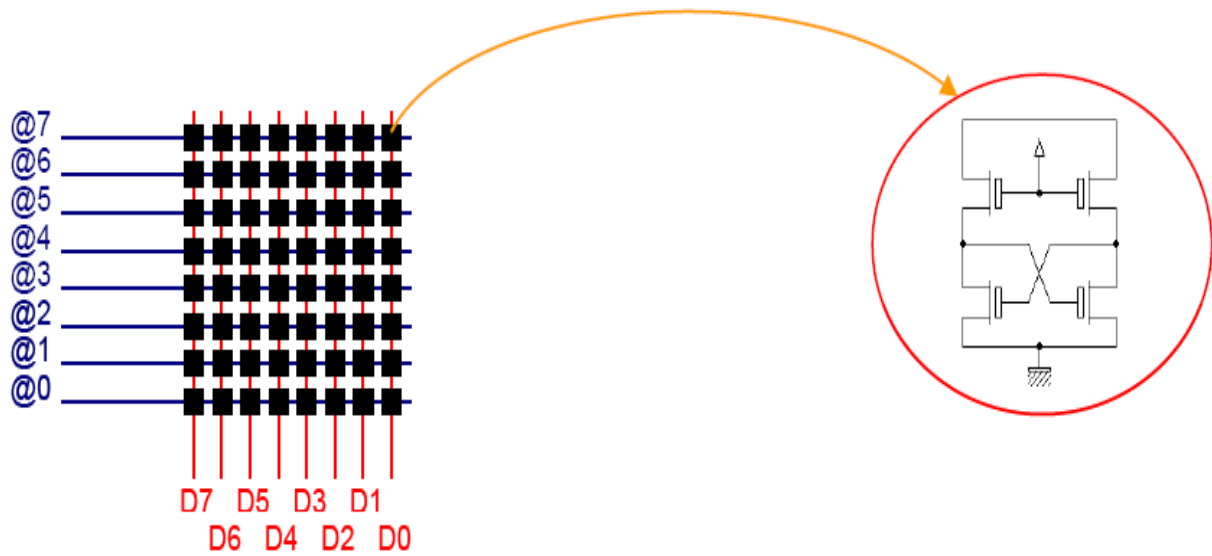
Fonctionnement des mémoires vives

La mémoire vive est constituée de centaines de milliers de petits condensateurs emmagasinant des charges. Lorsqu'il est chargé, l'état logique du condensateur est égal à 1, dans le cas contraire il est à 0, ce qui signifie que chaque condensateur représente un [bit](#) de la mémoire.

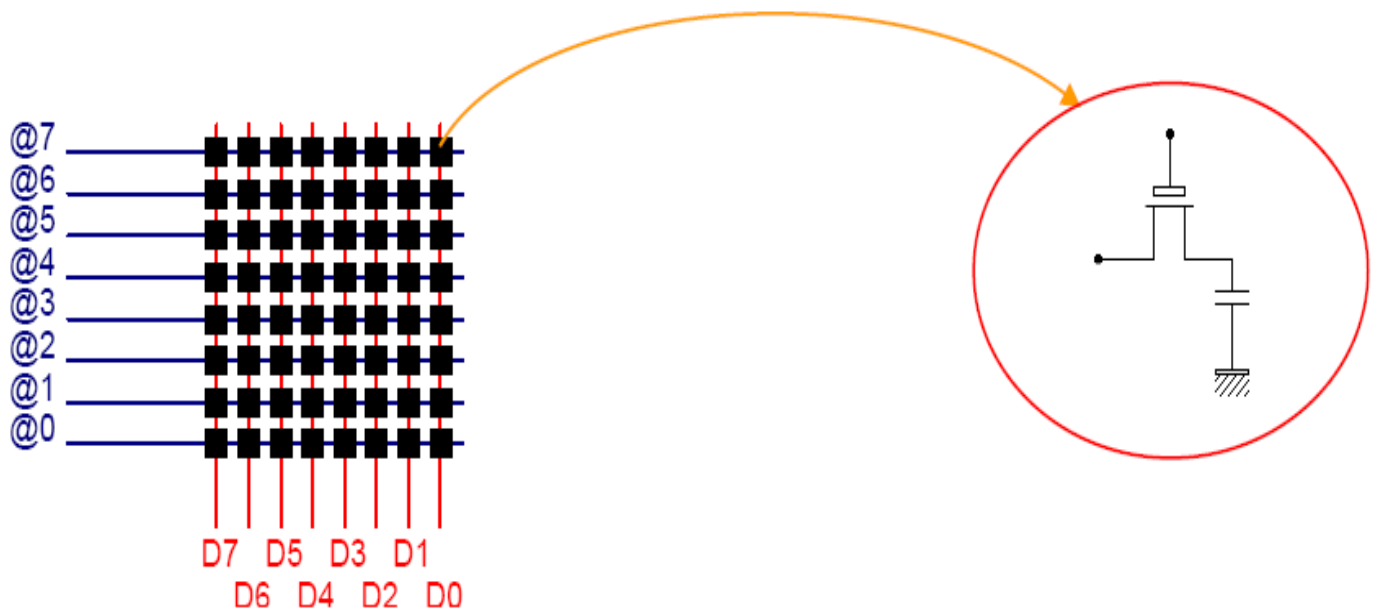
Etant donné que les condensateurs se déchargent, il faut constamment les recharger (le terme exact est *rafraîchir*) à un intervalle de temps régulier appelé cycle de rafraîchissement (d'une durée d'environ 15 nanosecondes (ns) pour une mémoire DRAM).

Chaque condensateur est couplé à un transistor (de type *MOS*) permettant de "récupérer" ou de modifier l'état du condensateur. Ces transistors sont rangés sous forme de tableau (matrice), c'est-à-dire que l'on accède à une "case mémoire" (aussi appelée *point mémoire*) par une ligne et une colonne.

Mémoire Statique



Mémoire Dynamique



Temps d'accès d'une mémoire vive

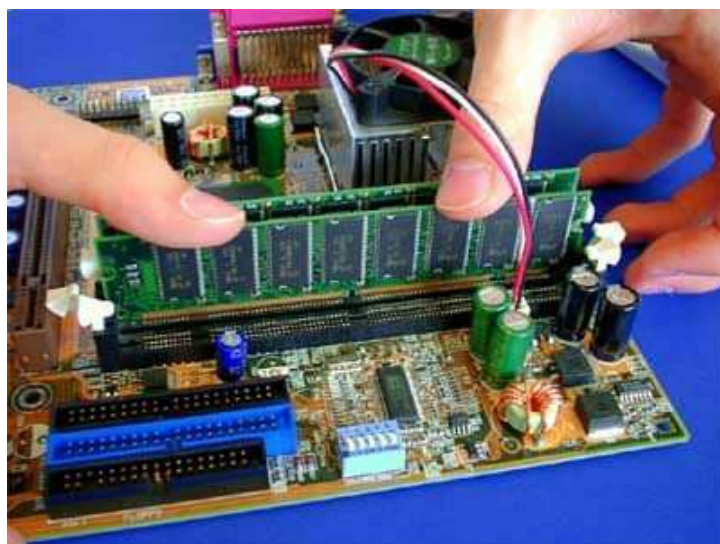
Chaque point mémoire est donc caractérisé par une adresse, correspondant à un numéro de ligne et un numéro de colonne. Or cet accès n'est pas instantané et s'effectue pendant un délai appelé temps de latence. Par conséquent l'accès à une donnée en mémoire dure un temps égal au temps de cycle auquel il faut ajouter le temps de latence.

Ainsi, pour une mémoire de type DRAM, le temps d'accès est de 60 nanosecondes (35ns de délai de cycle et 25ns de temps de latence). Sur un ordinateur, le temps de cycle correspond à l'inverse de la fréquence de l'horloge, par exemple pour un ordinateur cadencé à 200Mhz, le temps de cycle est de 5ns ($1/(200 \cdot 10^6)$).

Par conséquent un ordinateur ayant une fréquence élevée et utilisant des mémoires dont le temps d'accès est beaucoup plus long que le temps de cycle du processeur doit effectuer des cycles d'attente (en anglais *wait state*) pour accéder à la mémoire. Dans le cas d'un ordinateur cadencé à 200Mhz utilisant des mémoires de types DRAM (dont le temps d'accès est de 60ns), il y a 11 cycles d'attente pour un cycle de transfert. Les performances de l'ordinateur sont d'autant diminuées qu'il y a de cycles d'attentes, il est donc conseillé d'utiliser des mémoires plus rapides.

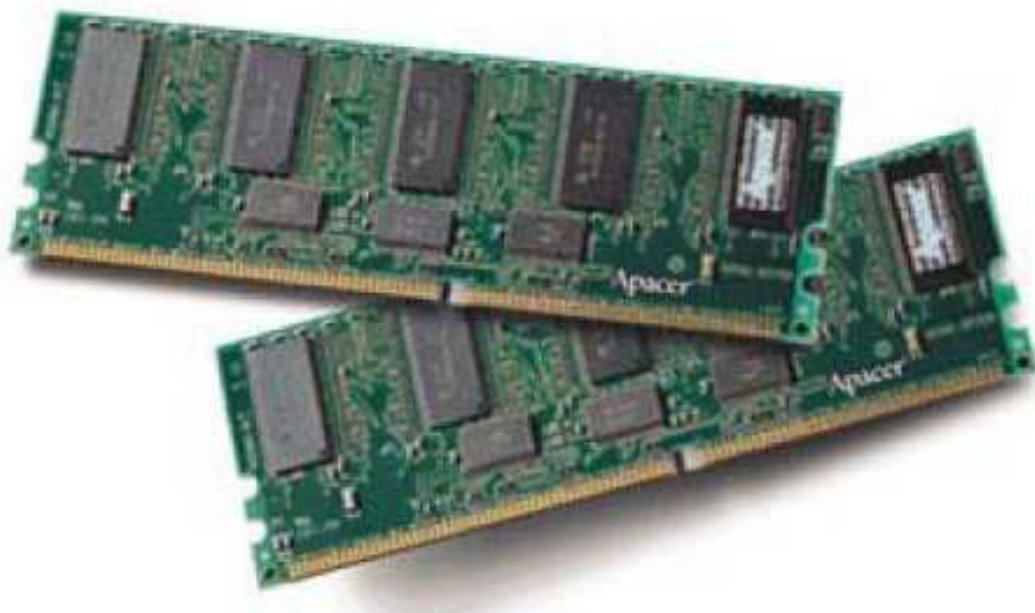
Les barrettes de mémoire vive

La mémoire centrale d'un ordinateur est constituée de barrettes mémoire, chacune est composée de plusieurs boîtiers DRAM



Exemples de barettes mémoire :**Exemples de barrettes mémoire SDRAM :**

Désignation	Type	FSB	Vitesse	B.P.
PC 100	SDR	100 MHz	100 MHz	0,8 Go/s
PC-1600	DDR-I	100 MHz	200 MHz	1,6 Go/s
PC-2100	DDR-I	133 MHz	266 MHz	2.13 Go/s
PC-2700	DDR-I	166 MHz	333 MHz	2.66 Go/s
PC-3200	DDR-I	200 MHz	400 MHz	3,2 Go/s
PC-3500	DDR-I	216 MHz	432 MHz	3.5 Go/s
PC-3700	DDR-I	233 MHz	466 MHz	3.7 Go/s
PC-4000	DDR-I	250 MHz	500 MHz	4.0 Go/s
PC2-3200	DDR-II	100 MHz	400 MHz	3,2 Go/s
PC2-4300	DDR-II	133 MHz	533 MHz	4,3 Go/s
PC2-5300	DDR-II	166 MHz	667 MHz	5,3 Go/s



Barette mémoire avec traitement des erreurs

Certaines mémoires possèdent des mécanismes permettant de pallier les erreurs afin de garantir l'intégrité des données qu'elles contiennent. Ce type de mémoire est généralement utilisé sur des systèmes travaillant sur des données critiques, c'est la raison pour laquelle on trouve ce type de mémoire dans les serveurs.

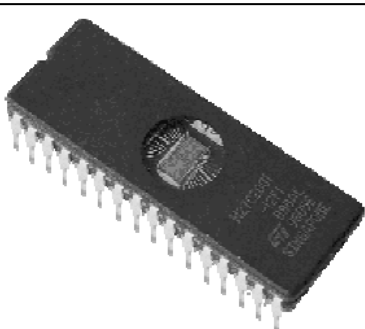
Les barrettes avec bit de parité permettent de s'assurer que les données contenues dans la mémoire sont bien celles que l'on désire. Pour ce faire, un des bits de chaque octet stocké en mémoire sert à conserver la somme des bits de données.

Le bit de parité vaut 0 lorsque la somme des bits de données est impaire et 1 dans le cas contraire.

De cette façon les barrettes avec bit de parité permettent de vérifier l'intégrité des données mais ne permettent pas de corriger les erreurs. De plus pour 8 Mo de mémoire, seulement 7 serviront à stocker des données, dans la mesure où le dernier mégaoctet conservera les bits de parité.

Les barrettes de mémoire ECC (*Error Correction Coding*) sont des mémoires possédant plusieurs bits dédiés à la correction d'erreur (on les appelle ainsi *bits de contrôle*). Ces barrettes, utilisées principalement dans les serveurs, permettent de détecter les erreurs et de les corriger.

Utilisation de la mémoire morte



Mémoire UV PROM

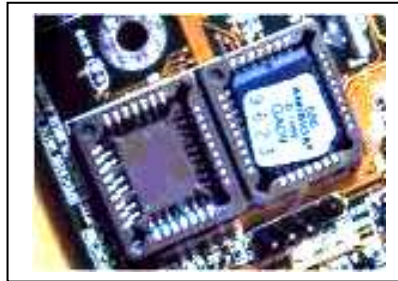


FLASH EPROM

Ce type de mémoire permet notamment de conserver les données nécessaires au démarrage de l'ordinateur. En effet, ces informations ne peuvent être stockées sur le disque.

dur étant donné que les paramètres du disque (essentiels à son initialisation) font partie de ces données vitales à l'amorçage.

Différentes mémoires de type *ROM* contiennent des données essentielles au démarrage, c'est-à-dire:



- Le [BIOS](#) est un programme permettant de piloter les interfaces d'entrée-sortie principales du système, d'où le nom de *BIOS ROM* donné parfois à la puce de mémoire morte de la carte-mère qui l'héberge.
- Le chargeur d'amorce: un programme permettant de charger le système d'exploitation en mémoire (vive) et de le lancer. Celui-ci cherche généralement le système d'exploitation sur le lecteur de disquette, puis sur le disque dur, ce qui permet de pouvoir lancer le système d'exploitation à partir d'une [disquette système](#) en cas de dysfonctionnement du système installé sur le disque dur
- Le Setup CMOS, c'est l'écran disponible à l'allumage de l'ordinateur permettant de modifier les paramètres du système (souvent appelé *BIOS* à tort...)
- Le Power-On Self Test (*POST*), programme exécuté automatiquement à l'amorçage du système permettant de faire un test du système (c'est pour cela par exemple que vous voyez le système "compter" la RAM au démarrage)

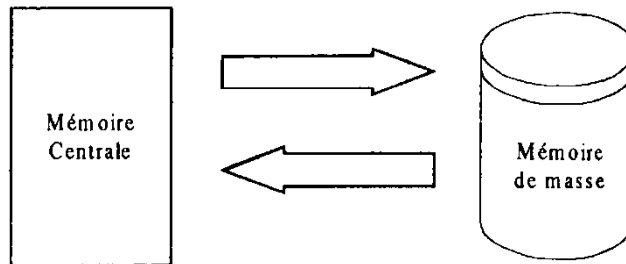
Etant donné que les ROM sont beaucoup plus lentes que les mémoires de types RAM (une ROM a un temps d'accès de l'ordre de 150 ns tandis qu'une mémoire de type SDRAM a un temps d'accès d'environ 10 ns), les instructions contenues dans la ROM sont parfois copiées en RAM au démarrage, on parle alors de *shadowing* (en français cela pourrait se traduire par *ombrage*, mais on parle généralement de *mémoire fantôme*).

Partie 2 : Gestion de la mémoire centrale

Objectifs :

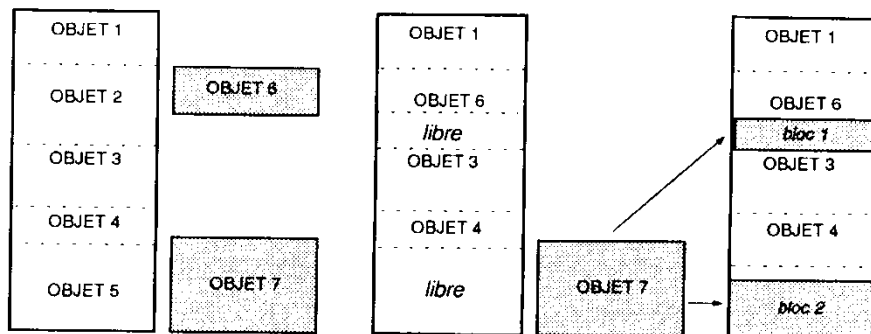
- support de la mémoire virtuelle
- allocation dynamique de la mémoire
- traduction des adresses logiques en adresses physiques
- implantation de mécanismes de sécurité et de protection

Mémoire Virtuelle



Le swapping

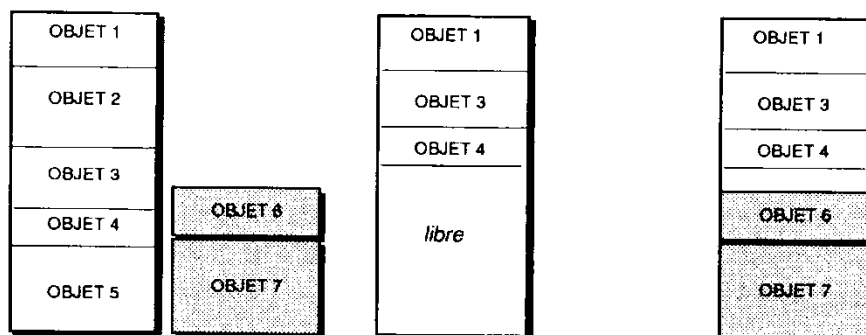
Allocation de la mémoire



Abandon des objets 2 et 5.
Transfert des objets 6 et 7.

Ecriture de l'objet 7 en 2 blocs

Allocation statique de la mémoire



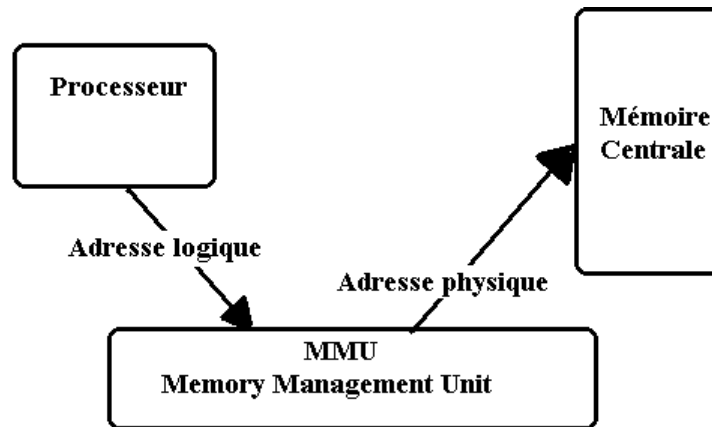
Abandon des objets 2 et 5.
Transfert des objets 6 et 7.

Tassement

Allocation dynamique de la mémoire

Adresse logique et adresse physique :

Un emplacement de l'espace adressable du microprocesseur est repéré par une adresse logique (ou adresse virtuelle), alors qu'on accède à un emplacement de la mémoire physique en le désignant par une adresse physique ou adresse réelle.



Le mécanisme de traduction des adresses logiques en adresses physiques peut faire appel soit :

- à la technique de pagination (cas d'un espace adressable linéaire)
- à la technique de segmentation (cas d'un espace adressable segmenté)

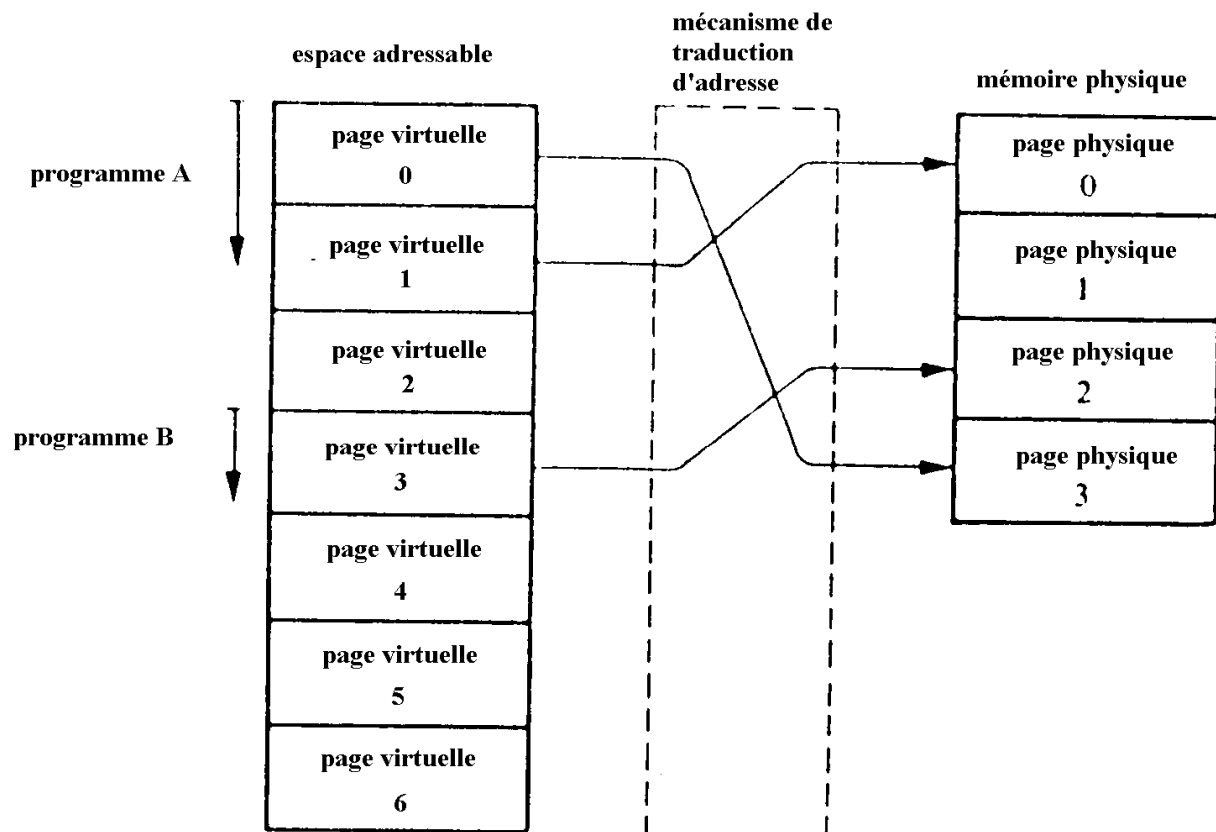


Fig. Allocation de la mémoire pour la pagination

Mécanismes de pagination

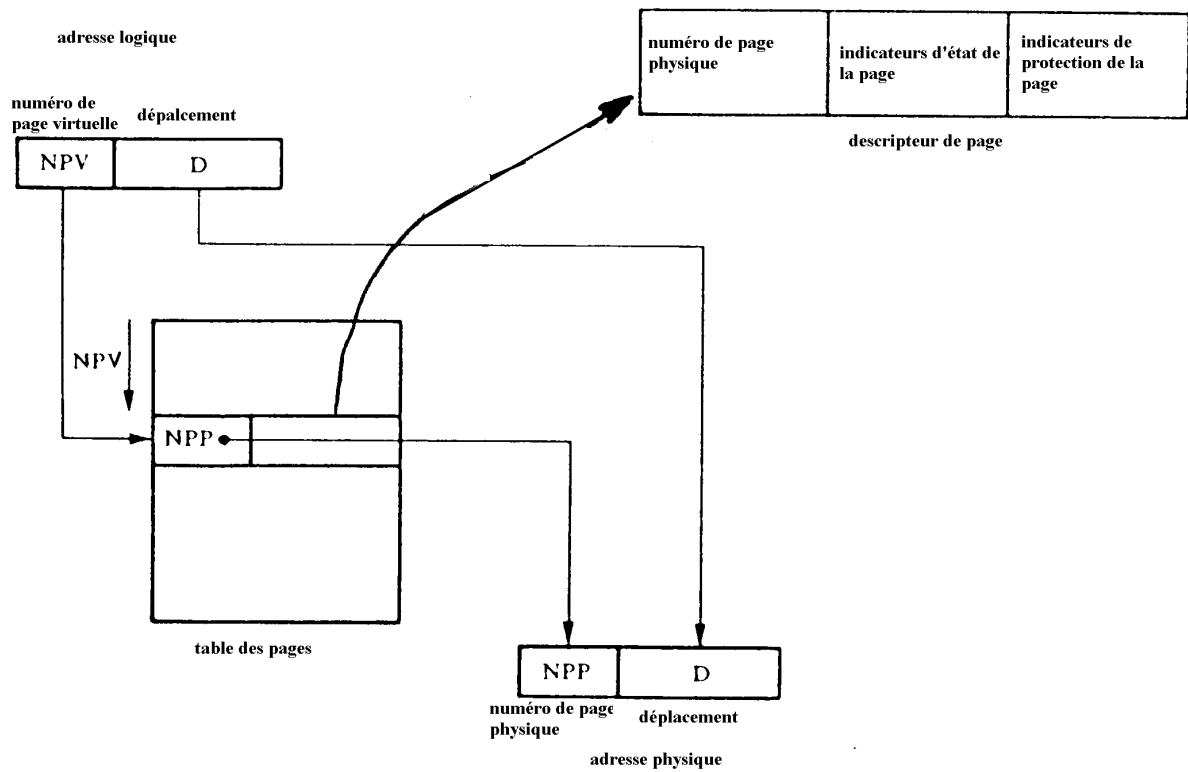


Fig. Traduction d'adresse pour la pagination

Mécanismes de segmentation

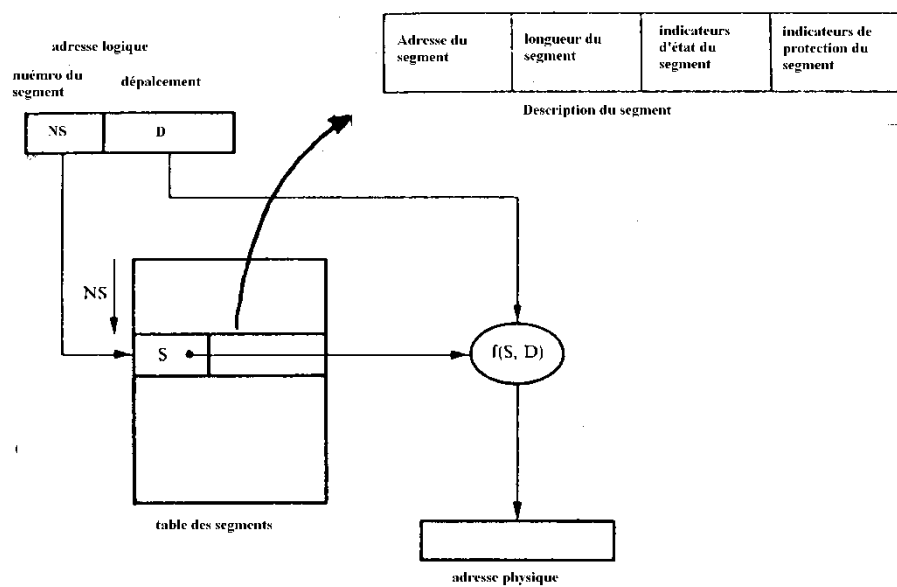


Fig. Traduction d'adresses pour la segmentation

Mécanismes de segmentation paginée

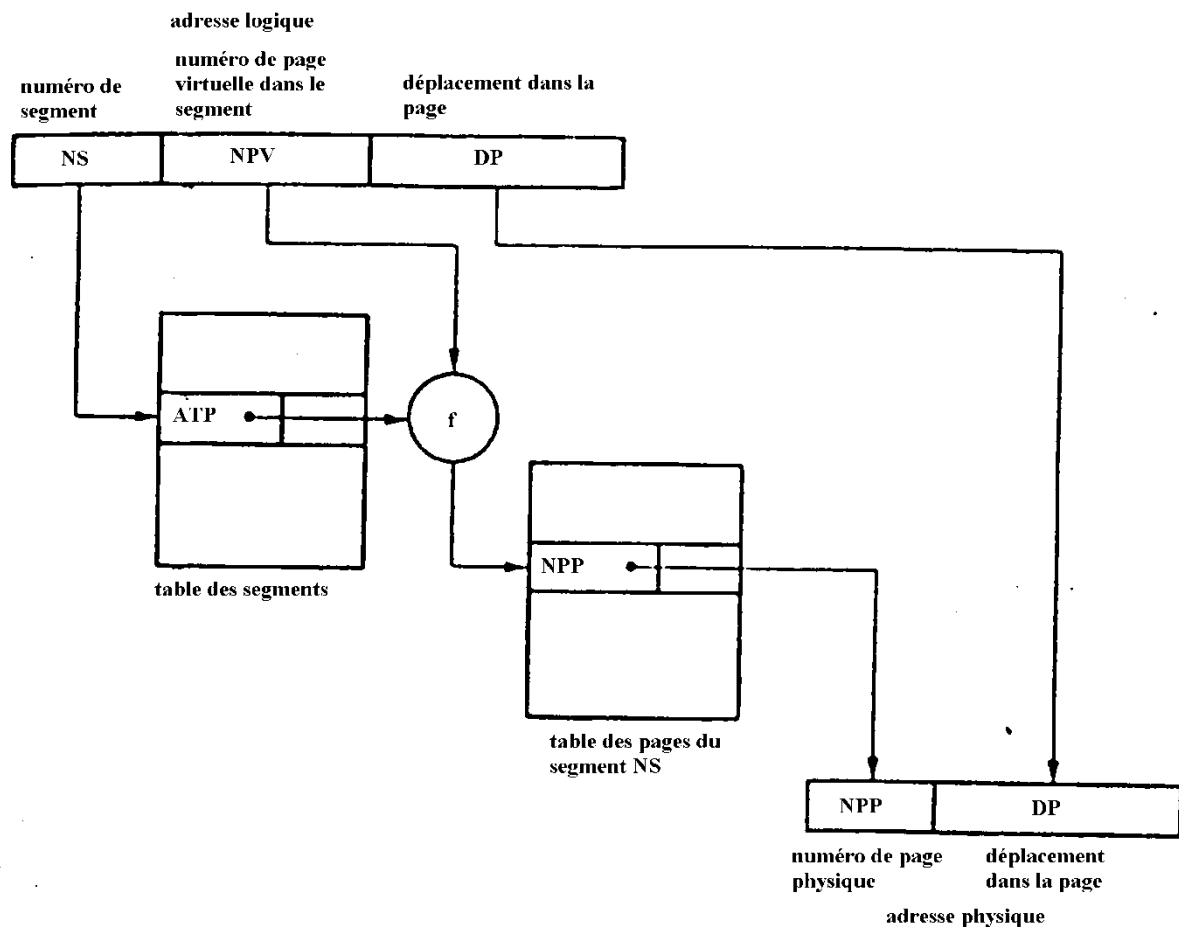


Fig. Traduction d'adresses pour la segmentation paginée

Indicateurs d'état et de protection du segment

Les indicateurs d'état fournissent les informations suivantes :

- la validité ou la présence du segment en mémoire physique
- la modification qui signale si l'on a écrit dans le segment
- l'utilisation qui indique si l'on a accédé au segment

Les indicateurs de protection définissent les droits suivants :

- tout accès est interdit
- seule l'exécution du code est autorisée
- seule la lecture est admise
- l'exécution et l'écriture sont possibles
- l'exécution, la lecture et l'écriture sont admises

Cas des microprocesseurs à partir du 80386

La mémoire virtuelle du microprocesseur 80386 est basée sur un mécanisme de segmentation paginée.

Le 80386 dispose de 16K segments indépendants. Chaque segment peut atteindre 1 Giga mots de 32 bits.

La taille des pages du 80386 étant de 4 Ko, un segment peut donc contenir jusqu'à 1 Méga pages.

Utilisation de la notion de mémoire virtuelle

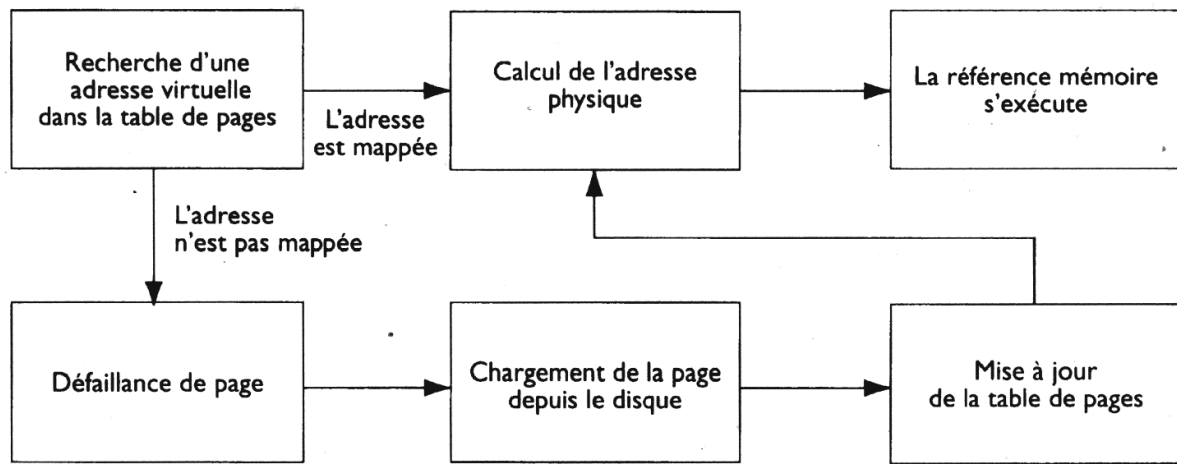


Figure 10.2 Traduction d'adresse.

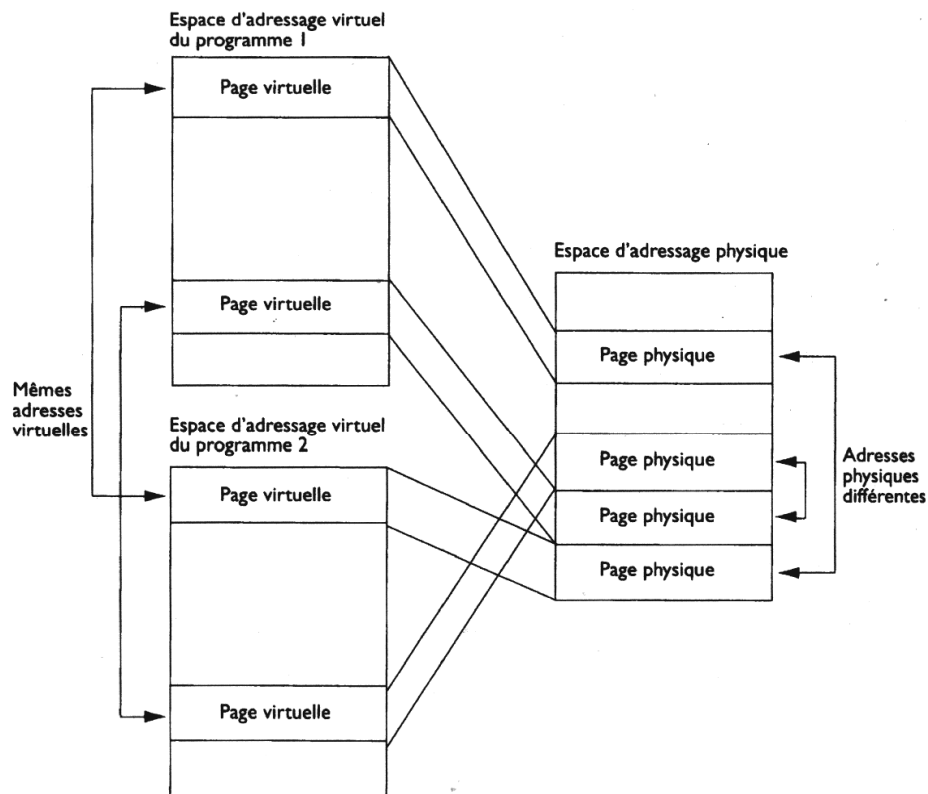


Figure 10.11 Protection offerte par la mémoire virtuelle.

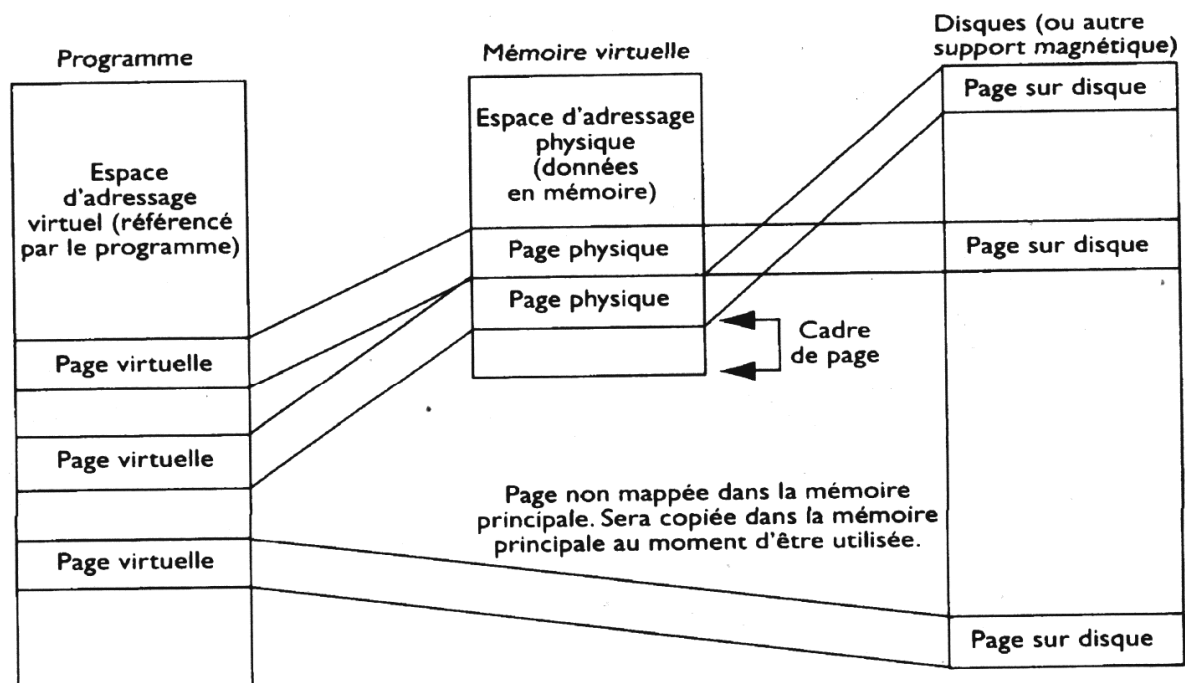
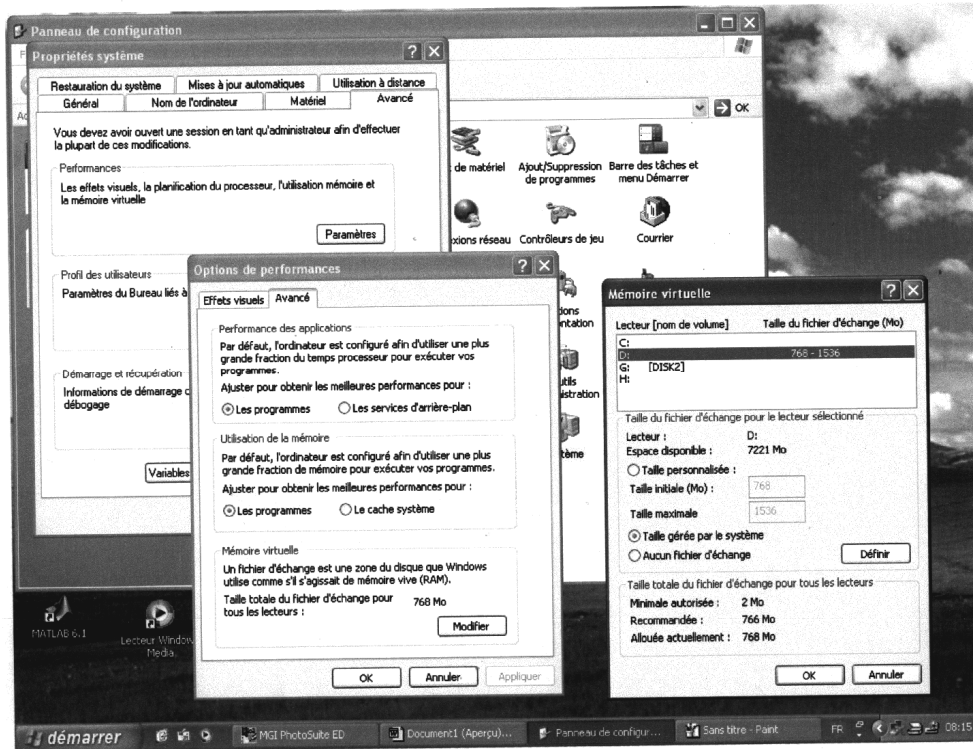


Figure 10.1 Mémoire virtuelle.

Paramétrage du fichier d'échange



Chapitre 6

Les périphériques et les systèmes d'entrée sortie

Les circuits d'entrée sortie

Les entrées sorties d'un ordinateur sont considérées comme des périphériques qui permettent la communication avec ' le monde extérieur '. (écran, clavier, souris, mémoire de masse, imprimante, etc). Ces périphériques sont commandés par le microprocesseur via des circuits appelés *contrôleurs de périphériques*. Cette commande est réalisée en programmant un ensemble de registre internes au contrôleur de périphériques.

On peut classer les registres interne au contrôleur de périphériques en trois familles :

- les registres de données par lesquels transitent les données qui circulent entre le microprocesseurs et le périphérique.
- les registres de commande qui permettent au microprocesseur de configurer le mode de fonctionnement du contrôleur
- les registres d'état à travers lesquels le microprocesseur dispose de l'état actuel du périphérique (libre, occupée, en attente d'information, disposant d'une information, etc)

Classification des périphériques

Classification en fonction de la vitesse de transmission de données

On peut classer les périphériques en trois familles :

- les périphériques lents : moins d'un changement d'état par seconde
 - affichage lumineux

- commutateurs
- relais
- les périphériques moyenne vitesse : de 1 à 10.000 bits de données par seconde
 - clavier
 - souris
 - imprimante
- les périphériques rapides : plus de 10.000 bits de données par seconde
 - mémoire de masse
 - écran vidéo
 - audio

Classification en fonction du format des données échangées

On peut aussi classer les périphériques en fonction du format des données échangées entre le périphérique et le contrôleur associé à ce périphérique.

Il existe deux types de format de données :

- format de données sous forme série : la communication est réalisée bit par bit (communication avec un réseau local, avec une souris, avec un périphérique USB, etc)
- format de données sous forme parallèle : la communication est réalisée en transmettant plusieurs bits simultanément (communication avec un disque dur, un lecteur de CDROM, etc....).

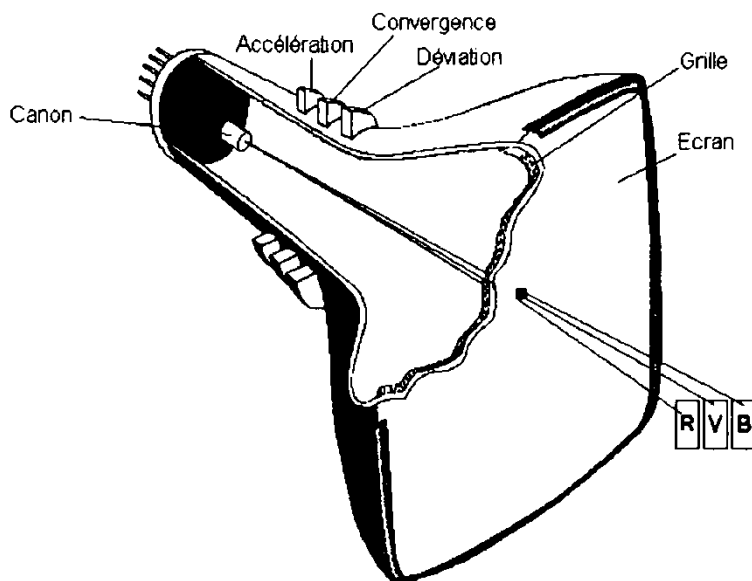
Exemples d'entrées sorties

• Clavier	(entrée)	0.01 Ko/s
• Souris	(entrée)	0.02 Ko/s
• Disquette	(sauvegarde)	50 Ko/s
• Imprimante Laser	(sortie)	100 Ko/s
• CD-Rom	(sauvegarde)	500 Ko/s
• Disque Dur	(sauvegarde)	5000 Ko/s
• Ecran graphique	(sortie)	30000 Ko/s

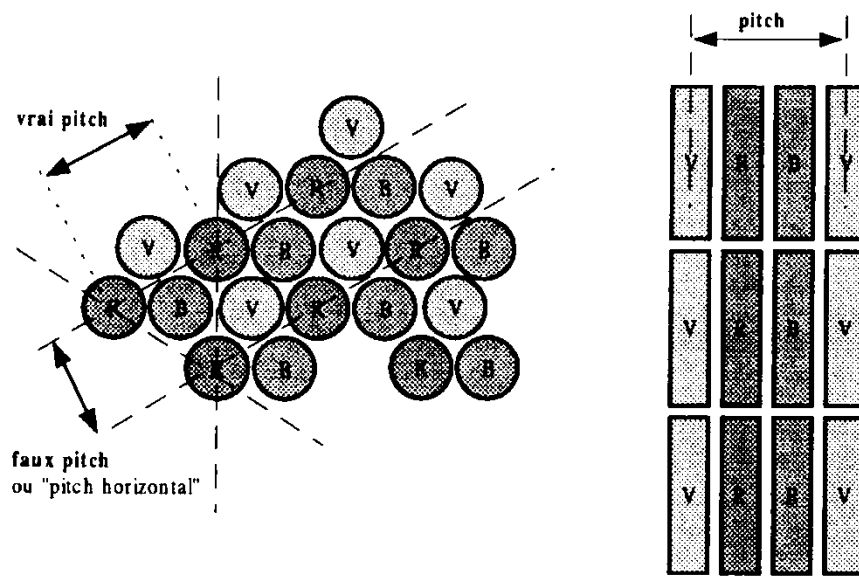
Périphériques d'affichage

L'affichage est assuré par un moniteur vidéo qui reçoit un signal vidéo de la carte vidéo.

Principe d'un moniteur vidéo



Le moniteur couleur - éclaté



Tube classique

Tube Trinitron

Caractéristiques des moniteurs vidéo

- Moniteur TTL ou analogique
- Fréquence verticale, ligne, multi-fréquences...

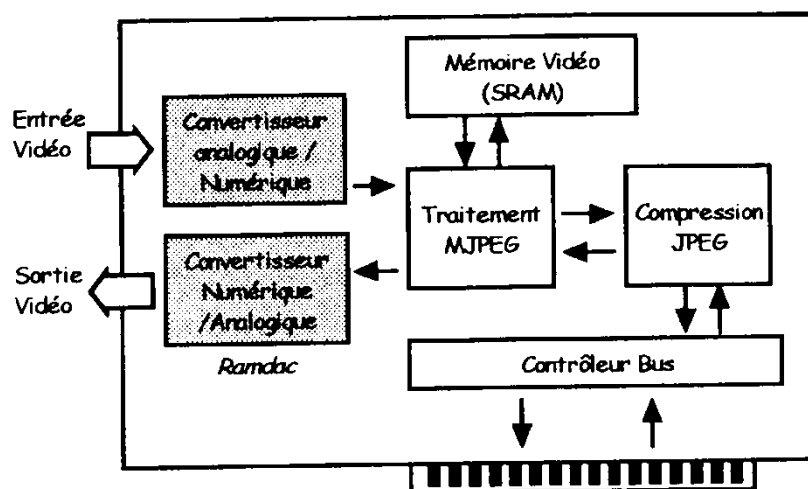
Largeurs de bande et fréquences de balayage des différents systèmes vidéo				
Système vidéo	Résolution	Largeur de bande	Fréquence de balayage vertical	Fréquence de balayage horizontal
MDA	720 * 350	16,257 MHz	50 Hz	18,43 KHz
CGA	640 * 200	14,318 MHz	60 Hz	15,75 KHz
HGC	720 * 350	16,257 MHz	50 Hz	18,43 KHz
EGA	640 * 350	16,257 MHz	60 Hz	21,85 KHz
	640 * 200	14,318 MHz	60 Hz	15,75 KHz
	720 * 350	16,257 MHz	50 Hz	18,43 KHz
VGA	640 * 400	25,175 MHz	70 Hz	31,50 KHz
	720 * 400	28,322 MHz	70 Hz	31,50 KHz
	640 * 480	25,175 MHz	60 Hz	31,50 KHz
	640 * 350	25,175 MHz	70 Hz	31,50 KHz
Super VGA	800 * 600	28,322 MHz	56 Hz	35,20 KHz
	800 * 600	30,425 MHz	60 Hz	37,88 KHz
	1024 * 768	50,350 MHz	60 Hz	48,00 KHz

(Unités : Hz=hertz, KHz=kilohertz, MHz=mégahertz)

- Résolution et bande passante

La carte Vidéo et les modes d'affichage

Le contrôle du moniteur s'effectue grâce à une carte contrôleur électronique (carte écran, carte vidéo...).



La carte VIDEO

De nombreux « standards » d'affichage se sont succédés.

- Les anciens modes MDA, CGA, HGA, EGA, PGA...
- Les modes VGA et Super VGA

En 1987, IBM définit le mode VGA (*Video Graphic Array*) qui émule entièrement les modes CGA et EGA et offre des définitions supplémentaires, dont 640 x 480 pixels en mode graphique et 720 x 400 en mode caractère, soit 30 lignes de 80 caractères (matrice de 9 x 16), en 16 couleurs parmi 262 144.

Les modes S-VGA (Super-VGA), VGA+, Double VGA, VGA étendu... sont autant d'extensions du mode VGA, apparues vers 1989, dont la résolution dépend des constructeurs. On y trouve ainsi des résolutions 800 x 600, 1024 x 768 et 1280 x 1024 (soit 1 310 720 points ou pixels) offrant un choix évoluant de 16 à 256 couleurs à choisir parmi 262 144 couleurs ou même 16 millions. S-VGA est le mode le plus employé actuellement.

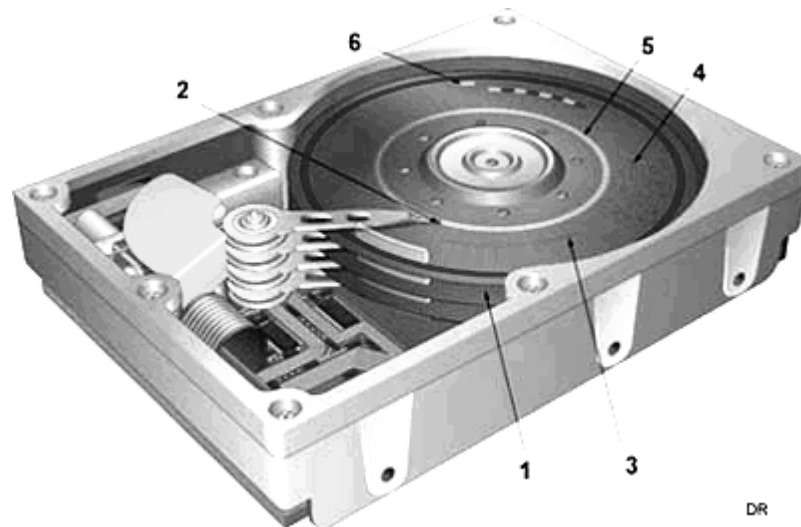
Nombre de couleurs	Nombre de bits
2 (noir et blanc)	1 bit
16	4 bits
256	8 bits
32 768	15 bits
65 536	16 bits
16, 7 millions	24 bits
16,7 millions + 256 niveaux de transparence	32 bits

Relation nombre de couleurs – occupation mémoire du pixel

Résolution	256 Ko	512 Ko	1 Mo	2 Mo	3 Mo	4 Mo
640x480	16	256	16.7 M	16.7 M	16.7 M	16.7 M
800x600	16	256	65 536	16.7 M	16.7 M	16.7 M
1024x768		16	256	65 536	16.7 M	16.7 M
1280x1024			16	256	65 536	16.7 M
1600x1200			16	256	256	65 536

Rapport du nombre de couleur à la résolution et occupation mémoire

La mémoire de masse (disque dur)



1- Plateau : c'est le support de l'information. Il conserve physiquement les données. Sur les disques durs de grande capacité, on trouve, en général, plusieurs plateaux.

2- Tête de lecture : a pour mission de lire et d'écrire les données au fur et à mesure que le disque tourne.

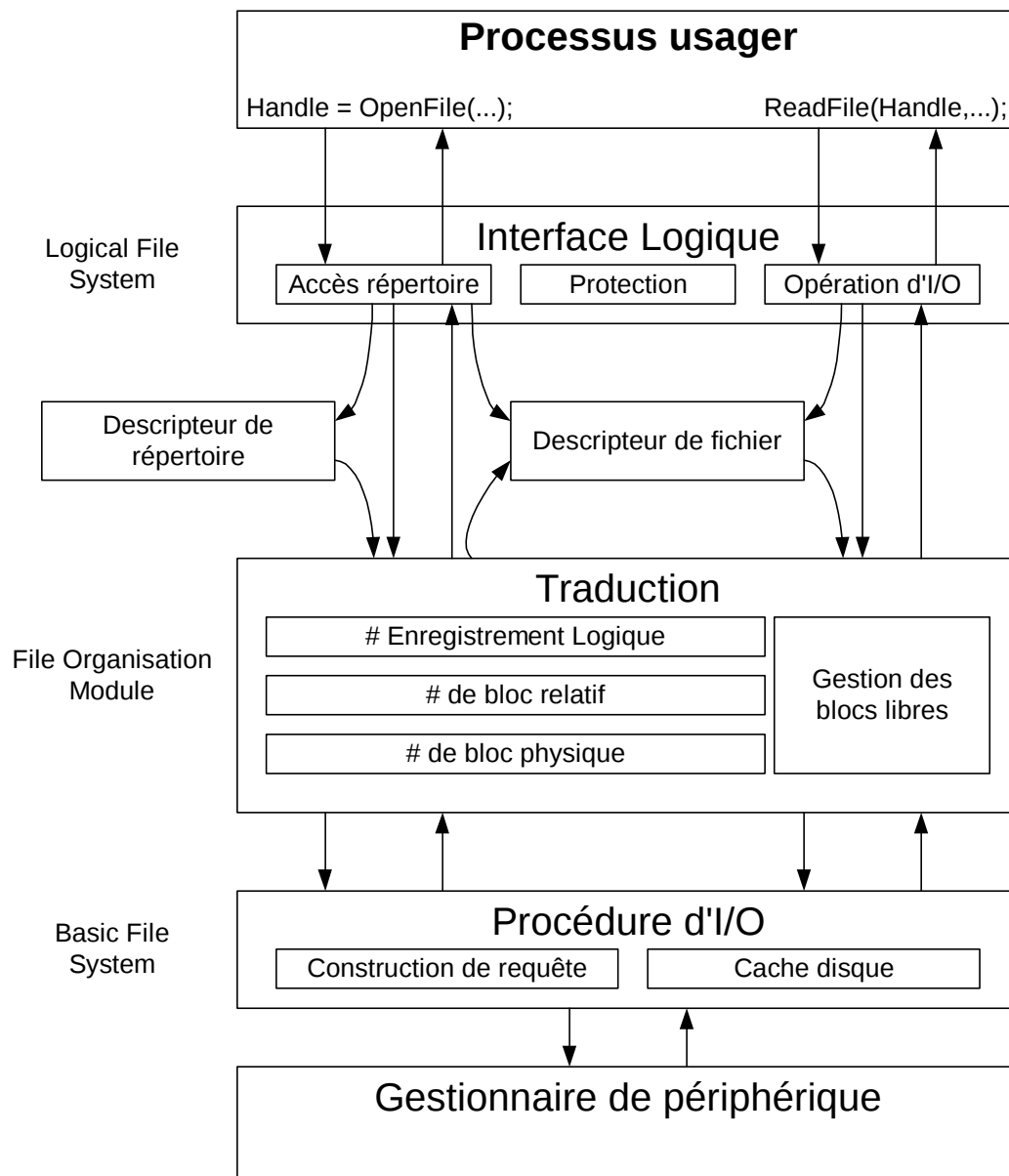
3- Fichier non fragmenté : les données sont inscrites sur des secteurs continus. La lecture de ces fichiers est alors plus rapide.

4- Fichier fragmenté : là, au contraire, les informations se placent dans des blocs non continus. La tête de lecture passe donc plus de temps à "recoller les morceaux" pour les récupérer, d'où un ralentissement global.

5- Piste : zone concentrique sur laquelle figurent les données d'un fichier. Chaque piste est divisée en secteurs.

6- Secteur : division logique d'une piste. Chaque secteur est lui-même divisé en blocs. Ces derniers représentent l'espace minimal attribué à un fichier.

La structure du système de fichier



- L'interface Logique (Logical File System) vérifie si la requête de l'utilisateur est valide (path et permission valides).
- La traduction (File Organisation Module) convertit le fichier logique en fichier physique.
- Le Basic File System est un ensemble de routines permettant d'aller chercher des informations sur un disque dur ou autre support de masse.
- Un processus usager peut ouvrir, lire, ou écrire un fichier identifié à l'aide d'un numéro (handle).

- Lors d'une requête d'ouverture de fichier, l'interface logique vérifie le chemin d'accès au fichier dans une structure contenue en mémoire. Elle vérifie également les droits d'accès au fichier dans un descripteur de fichier. Si l'information sur le répertoire ou le fichier n'est pas en mémoire, il faut aller la chercher sur le disque dur. Après ouverture du fichier, un numéro est attribué au fichier. Ce numéro sert de référence dans une table de fichiers ouverts afin d'éviter d'avoir à re-vérifier le chemin d'accès et les permissions sur le fichier à chaque lecture/écriture.
- Lors d'une opération d'écriture ou de lecture d'un fichier, il faut traduire l'emplacement requis dans le fichier en emplacement sur le disque. C'est le rôle du File Organisation Module. Par exemple, le File Organisation Module déterminera que les bytes 10 à 210 du fichiers C:\ABC.txt correspondront aux blocs de données 3 et 5 du cylindre 3, tête 2, piste 31 sur le disque dur primaire.
- Une fois que les emplacements désirés sur le disque sont identifiés, il faut accéder au disque. Des routines du Basic File System sont utilisées à cette fin. Il s'agit des routines du gestionnaire d'I/O qui crée une requête pour le gestionnaire de périphérique approprié. On remarquera que le Basic File System ira souvent chercher plus de données qu'il n'en faut sur le disque afin d'éviter d'accéder souvent à ce dernier. Une cache est utilisée.

Quel que soit le système de fichier, les types d'information suivants se retrouvent sur un disque dur:

- Données sur le format du disque, sur ces partitions et procédures de démarrage du système d'exploitation (Bootstrap).
- Données sur les répertoires et la structure de répertoires du disque.
- Données décrivant l'emplacement des fichiers sur le disque.
- Contenus des fichiers eux-mêmes.

Allocation chaînée et FAT

Répertoire des fichiers		Contenu
nom fichier	1er cluster	
Venus	1	
Mars	5	
Terre	8	

Index	FAT ptrCluster	Cluster	Disque Données
0	0	0	
1	2	1	fe
2	12	2	mm
3	~	3	s
4	0	4	
5	6	5	ho
6	7	6	mm
7	~	7	es
8	9	8	en
9	10	9	fa
10	3	10	nt
11	0	11	
12	~	12	es
13	0	13	

- FAT signifie File Allocation Table.
- Pour le FAT, une table contient le nom et le chemin de tous les fichiers (Table de répertoire). Pour chaque fichier de la table, le premier cluster du fichier est identifié.
- Une autre table, la FAT, décrit l'utilisation de chaque cluster comme le montre la figure à droite.
- Dans la FAT, un fichier est identifié sous forme d'une chaîne de cluster: chaque cluster pointe sur le cluster suivant du fichier.
- Un caractère spécial (lire nombre spécial: un caractère est un nombre!!!) indique qu'un cluster n'a jamais été utilisé ('0' dans l'exemple). Un autre caractère spécial indique la fin d'un fichier ('~' dans l'exemple).

- Dans l'exemple, les clusters ont 2 bytes. Le fichier Venus, qui contient "femmes", est réparti sur la chaîne de clusters 1-2-12. Le fichier Mars, sur les clusters 5-6-7. Finalement, le fichier Terre est contenu dans les clusters 8-9-10-3.

FAT 16 et FAT 32

- Le FAT16 peut adresser jusqu'à 2^{16} clusters, soit 65536. Cette quantité de clusters est très petite par rapport à un disque dur moderne. Par exemple, un disque dur de 16Go devra avoir des clusters de $16\text{Go}/(2^{16}) = 2^{18} = 256\text{Ko}$. Avec des clusters de cette taille, énormément de mémoire disque est gaspillée.
- Le FAT32 a été créé dans le but d'étendre les capacités du FAT 16. Le FAT 32 peut adresser 2^{32} clusters, soit 4Go clusters. Les clusters du FAT 32 peuvent être plus petits que ceux du FAT16 pour une taille de disque dur donnée. En contrepartie, la FAT d'un système FAT32 sera plus grosse: il faut 4 bytes pour pointer sur le cluster suivant plutôt que 2.

NTFS

- Voyant les limitations de ses systèmes de fichiers, Microsoft a créé le NTFS.
- NTFS signifie NT File system (Certains auteurs donnent une signification aux lettres N et T).
- Le NTFS utilise 64 bits pour adresser les clusters.
- Le NTFS a plusieurs caractéristiques non supportées par le FAT: l'impact des erreurs disque est minimal, le système de fichier contient de l'information pour la sécurité des fichiers, les noms sont en Unicode, etc.
- Le NTFS fonctionne par Volume. Un volume est une fraction d'un disque dur de taille déterminée par l'utilisateur (Un disque dur peut être un unique volume). Chaque volume du NTFS est décrit par des fichiers invisibles à l'utilisateur contenus à l'intérieur du volume. Ces fichiers, des *métafichiers*, décrivent la structure et le contenu du volume. Voici quelques exemples de métafichiers:
 - Le fichier \$BADCLUS décrit les clusters défectueux sur le disque
 - Le fichier \$BITMAP décrit les clusters libres du disque.
 - Le fichier \$LOGFILE décrit toutes les transactions effectuées sur le disque (permet de récupérer d'une faute de disque)
 - ...

Bibliographie

Cours Web :

A brief history of Intel and AMD microprocessors (cours DEUG Université Angers) – *Jean-Michel Richer*
Architecture Avancée des ordinateurs (cours Supelec Rennes) – *Jacques Weiss*
Architecture des ordinateurs (cours IUT GTR Montbéliard) – *Eric Garcia*
Architecture des ordinateurs (cours IUT SRC Marne la Vallée) – *Dominique Présent*
Architecture des ordinateurs (cours Université Franche Compté) – *Didier Teifreto*
Architecture des ordinateurs (cours IUP STRI Toulouse)
Architecture des ordinateurs (cours Université de Sherbrooke) – *Frédéric Mailhot*
Architecture des ordinateurs (cours Polytechnique) – *Olivier Temam*
Architecture des ordinateurs (cours IUT GTR Villetaneuse) – *Emmanuel Viennet*
Architecture des ordinateurs (cours DEUG MIAS) – *Frédéric Vivien*
Architecture des Ordinateurs (cours Licence Informatique USTL) – *David Simplot*
Architecture des machines et systèmes Informatiques – *Joëlle Delacroix*
Architectures des processeurs (cours DEUST Nancy) – *Yannick Chevalier*
Architecture des systèmes à microprocesseurs – *Maryam Siadat et Camille Diou*
Architecture des systèmes à microprocesseurs (cours IUT Mesures Physiques) – *Sébastien Pillement*
Architecture Systèmes et Réseaux (cours DEUG 2^{ème} année) – *Fabrice Bouquet*
Carte graphique (ENIC) – *Julien Lenoir*
Cours de réseau (cours EISTI) – *Bruno Péant*
Cours de réseaux (cours Maîtrise Informatique Université Angers) – *Pascal Nicolas*
Du processeur au système d'exploitation (cours DEUST Nancy) – *Yannick Chevalier*
Introduction to computer architecture (cours DEUG Université Angers) – *Jean-Michel Richer*
Les réseaux : introduction (DESS DCISS) – *Emmanuel .Cecchet*
Les systèmes informatiques (cours CNAM) – *Christian Carrez*

Sites web :

Fonctionnement des composants du PC
<http://www.vulgarisation-informatique.com/composants.php>
Cours d'initiation aux microprocesseurs et aux microcontrôleurs
http://www.polytech-lille.fr/~rlitwak/Cours_MuP/sc00a.htm
Architecture des ordinateurs – Université Angers
<http://www.info.univ-angers.fr/pub/richer/ens/deug2/ud44/>
Les docs de Heissler Frédéric
<http://worldserver.oleane.com/heissler/>
X-86 secret
<http://www.x86-secret.com/>
Le cours hardware d'YBET informatique
<http://www.ybet.be/hardware/hardware1.htm>
Informa Tech
<http://informatech.online.fr/articles/index.php>

Livres :

Architecture et technologie des ordinateurs (Dunod) – *Paolo Zanella et Yves Ligier*
Technologie des ordinateurs et des réseaux (Dunod) – *Pierre-Alain Goupille*
Les microprocesseurs, comment ça marche ? (Dunod) – *T. Hammerstrom et G. Wyant*