

# Architecture des Ordinateurs

## Introduction

L'INFORMATIQUE couvre aujourd'hui à peu près toutes les branches de l'activité humaine..

Elle a pour objet le traitement d'informations

codées dans les ordinateurs sous une forme binaire à l'aide de *bits* (0 ou 1).

*hardware*

*L'ensemble des organes physiques qui servent à ce traitement est appelé matériel*

*software*

*ensemble des programmes, des langages et des systèmes d'exploitation*

L'organe principal d'un ordinateur

Unité centrale

Exécuter les instructions

Périphériques

Organes d'entrée-sortie

### 1. L'historique

L'IDÉE de faire exécuter des calculs par une machine date du XVII<sup>e</sup> siècle.



Blaise PASCAL



Permettre d'additionner et même soustraire des nombres de 6 chiffres



LEIBNIZ  
(1650 à 1670)



Effectuer les quatre opérations de base.

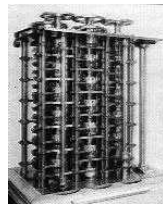
Mécaniques

Charle BABBAGE  
(1820)

Machine à calculer universelle:  
- Programmable  
- Mécanique et numérique.



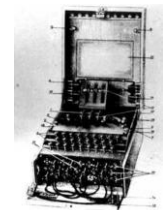
Difficile à réaliser



Enigma

machine  
Electromécanique

Décryptage des messages codés



## 2. Architecture VON NEUMANN

Synthèse des idées  
VON NEUMANN



Objectifs

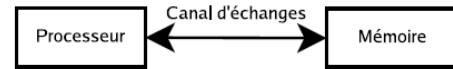
- Méthode souple pour programmer la mémoire
- Les concepts de mémoire tampon, vive et morte



Machine puisse prendre seule des décisions logiques (principe d'autonomie).

### La machine de VON NEUMANN

- L'unité de traitement de processus.
- L'unité de stockage ou mémoire.



ENIAC



## 3. Qu'entend-t-on par architecture ?

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
- Efficacité et facilité de construction
- Performances d'ensemble et facilité de programmation
- etc ...

## 4. Qu'est ce qu'un microprocesseur ?

Un microprocesseur est un circuit intégré complexe.

Intégration sur une puce de fonctions logiques combinatoires et séquentielle

résulte

capable

Interpréter et d'exécuter les instructions d'un programme

## 5. 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
- etc...

## Architecture des Ordinateurs Représentation des données

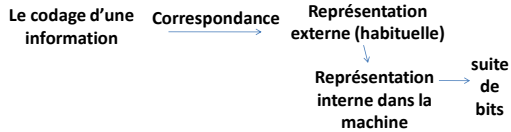
ENSIAS  
2013-2014

## 1. Introduction

Les informations traitées par un ordinateur peuvent être de différents types (texte, nombres, etc.)



Représentées et manipulées par l'ordinateur sous forme binaire.



## Représentation décimale

### Représentation binaire

simple, facile à réaliser

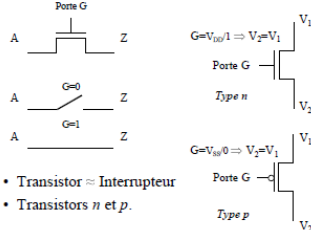
Opérations arithmétiques de base  
(addition, multiplication etc.)  
Faciles à exprimer en base 2

**N.B.** Lorsqu'un processeur comporte des opérateurs qui ne peuvent effectuer des calculs que sur des nombres d'au plus N bits

## 2. Représentation des Portes Élémentaires

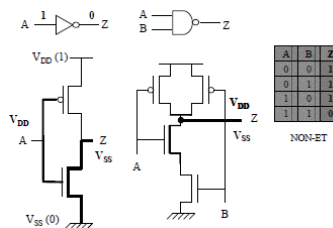
- NON (NOT)
- ET (AND)
- OU (OR)
- NON-ET (NAND)
- NON-OU (NOR)

### Le Transistor



- Transistor  $\approx$  Interrupteur
- Transistors n et p.

### Fonctions Simples avec des Transistors



### Réalisation d'un Circuit

Table de vérité  
Table de karnaugh

## 3. Représentation des nombres entiers

Deux niveaux de valeur:  $V_{DD}$  et  $V_{SS}$  1/0.

- Représentation des nombres en base 2.

### 3.1 ENTIERS NATURELS

Base décimale  
Base binaire 41: 101001

### 3.2. Les nombres négatifs (signés)

- ✓ Le complément vrai  
exemple : 0110 devient 1010 et 0000 devient bien 0000

Admettons que nous ayons des nombres sur 4 bits.

- 0100 représente soit 4, et son complément à 2.
- 1100 représente soit -4.

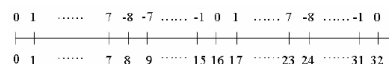
Le codage sur 8 bits ne permet plus d'obtenir des nombres compris entre 0 et 255 mais entre -128 et +127.

Complément à 2  $\rightarrow$  Processeur fonctionnant avec des mots de n bits  $\rightarrow$  Représente l'intervalle de valeurs  $[-2^{n-1}; 2^{n-1}-1]$  en utilisant l'intervalle  $[0; 2^n-1]$

- ✓ Les valeurs 0 à  $2^{n-1}-1$  sont représentées directement par les entiers 0 à  $2^{n-1}-1$
- ✓ Les valeurs  $-2^{n-1}$  à  $-1$  sont représentées par les entiers  $2^{n-1}$  à  $2^n-1$

Avec ce codage, on peut remarquer que  $X$ ,  $-2^{n-1} \leq X \leq -1$  est codé par  $2^n+X$ .

Exemple : n=4 bits.



### 3.4 Représentation biaisée

Appelée aussi « représentation par excès».

Considérer tout nombre binaire codé comme un entier non signé auquel on soustrait une constante, ou biais (bias)

$$\begin{aligned} -97 \text{ biaisé } 127 &\rightarrow -97 + 127 = 30 \\ 8 \text{ biaisé } 127 &\rightarrow 8 + 127 = 135 \end{aligned}$$

### 3.5 Représentation des Nombres Flottants

- On a vu la limitation à  $\pm 32\,767$  avec 16 bits
- Et si on veut coder des nombres nettement plus grands ?
- On passe aux nombres en virgule flottante
- C'est la notation dite des ingénieurs sur les calculatrices

$$\begin{aligned} 300\,000 &= 0,3 \cdot 10^6 \\ 0,3 &\text{ est la mantisse} \\ 6 &\text{ est l'exposant} \end{aligned}$$

- Pour des raisons pratiques, la mantisse est normalisée pour être comprise entre 0 et 1

Exemples

$$\begin{aligned} 65,1234 &= 0,6511 \cdot 10^2 &= 0,6511 \text{ E}2 \\ 0,00034 &= 0,340 \cdot 10^{-3} &= 0,340 \text{ E}-3 \\ -512 &= -0,512 \cdot 10^3 &= -0,512 \text{ E}3 \end{aligned}$$

Bien entendu, la chose reste tout à fait possible en base 2.

L'écriture devient alors:

$$(-1)^{\text{signe}} \cdot \text{Mantisse} \cdot 2^{\pm \text{Exposant}}$$

Codage

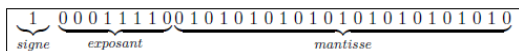
**Exemple:**  $11001 = 11001 \cdot 2^0 = 1,1001 \cdot 2^4 = 1100100000 \cdot 2^{-5}$

La norme IEEE 754

Sur 32 bits

31	30	23 22	0
Signe	Exposant	Mantisse	
1 bit	8 bits	23 bits	

Exemple 1:



{ signe : bit a 1 donc le nombre est négatif.

{ exposant binaire vaut 00011110 = 30 donc l'exposant est =  $30 - 127 = -97$

{ mantisse : (ne pas oublier le premier 1) vaut :

$$1 \cdot 2^{-2} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-10} + 2^{-12} + 2^{-14} + 2^{-16} + 2^{-18} + 2^{-20} + 2^{-22} = 4/3$$

Le nombre vaut donc  $-4/3 \cdot 2^{-97} = 4/3 \cdot (2^{10})^{-10} \cdot 2^3 = 32/3 \cdot 10^{-30}$ .

Exemple 2:

Opération inverse: trouver la représentation sur 32 bits du réel 278.

Le nombre est positif d'où 0 comme bit de signe.

On doit trouver l'exposant  $e$  tel que:

$$X = m \cdot 2^e \text{ avec } 1 \leq m < 2 \text{ d'où } e = 8 \text{ (car } 2^8 < X < 2^9)$$

et donc  $m = 278/256$ .

L'exposant est biaisé à 127 d'où  $e = 127 + 8 = 135$  représentée comme 1000111

Une manière simple de trouver la représentation de  $m$  est de voir que  $X = 2^8 + 2^4 + 2^2 + 2 = d'où m = 1.0001011$  en base 2

ce qui donne la représentation du nombre (ne pas oublier que la partie entière de la mantisse est omise) :

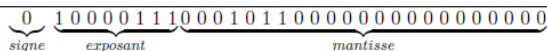


Tableau Récapitulatif

e (exposant)	f (mantisse)	représente
0	0	$\pm 0$
0	$\neq 0$	$\pm 0,fx2^{-127}$ simple Ou $\pm 0,fx2^{-1023}$ double
$0 < e < e_{\max}$	$\forall f$	$\pm 1,fx2^{e-127}$ simple Ou $\pm 1,fx2^{e-1023}$ double
$e_{\max}$ (255 ou 2047)	0	$\pm \infty$
$e_{\max}$ (255 ou 2047)	$\neq 0$	NaN (Not a Number)

## Architecture des Ordinateurs Systèmes à Microprocesseur

### 1. Architecture interne d'un système à microprocesseur

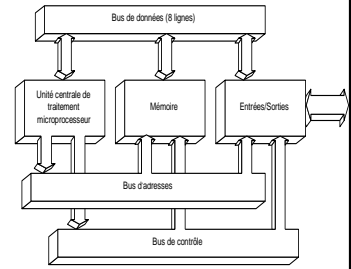
Trois blocs:

- ✓ le microprocesseur,
- ✓ la mémoire
- ✓ les entrées/sorties

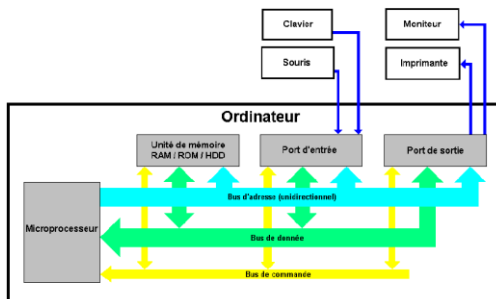
Reliées entre elles  
par trois bus

Communication:

- ✓ bus de données,
- ✓ bus d'adresses,
- ✓ bus de commande.



Architecture interne d'un ordinateur



### 2. Le microprocesseur

Un microprocesseur est  
l'implantation en un seul boîtier,  
en technologie LSI ou VLSI

Unité centrale de  
traitement de données CPU

Unité arithmétique et  
logique ALU

Son principal travail est de fournir aux autres éléments, qui lui sont  
raccordés, la synchronisation nécessaire et fonctionnelle du  
traitement des informations qui lui sont acheminées.

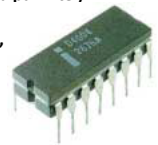
### 2. Unité Arithmétique et Logique (UAL)

- ✓ Opérations arithmétiques :
  - Addition,
  - Soustraction,
  - Multiplication par 2,
  - Division par 2,
  - Changement de signe...
- ✓ Opérations logiques :
  - Décalages,
  - Rotations,
  - Complémentation, "ou" bit à bit, "et" bit à bit, "ou" exclusif...

### Historique des microprocesseurs (Evolution de ce composant)

#### ✓ 4004 (1971): Le premier microprocesseur conçu par Intel)

- Bus de donnée de 4 bits,
- Il fonctionnait à la fréquence de 108 KHz,
- Equipé de 2300 transistors.



#### ✓ 8086: (développé en 1978):

- Premier microprocesseur de type x86
- Microprocesseurs actuels sont en réalité des améliorations du 8086,
- Le 8086 était équipé d'un bus de données de 16 bits,
- Fonctionnait à des fréquences diverses selon plusieurs variantes: 5, 8 ou 10 MHz.

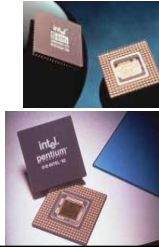
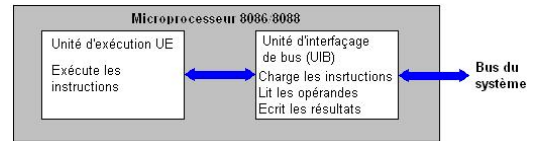


✓ **80486 (1989):**

- Intègre une mémoire cache de 8 Ko,
- Permettant d'accroître sa vitesse de traitement,
- Le 486 est le premier microprocesseur à être équipé de plus d'un million de transistors,
- Le 486 présentait des fréquences 33, 40 et 66 MHz.

✓ **Pentium (1993) :**

- Deux mémoires cache de 8 Ko, la première pour les instructions, la seconde pour les données,
- L'architecture développée permettant à ce microprocesseur d'exécuter deux instructions simultanément,
- Le pentium original fonctionnait à la fréquence de 66 MHz.

**Traitement des instructions**

Deux unités internes distinctes:

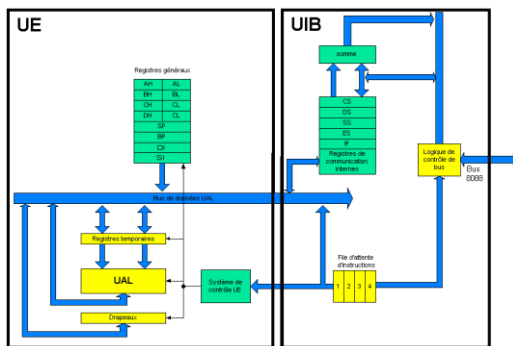
✓ **Unité d'Interfaçage de Bus (UIB):**

- Récupérer et stocker les informations à traiter,
- Etablir les transmissions avec les bus du système.

✓ **Unité d'Exécution (UE):**

- Exécuter les instructions qui lui sont transmises par l'UIB.

**N.B.** Le microprocesseur pris comme exemple est le 8086/8088.

**Schéma détaillé du système d'instructions:****3. La mémoire**

C'est la partie où l'on retrouve les programmes exécutables et les données à traiter.

**Mémoire morte**

L'avantage de la ROM (Read Only Memory), c'est d'être permanente; son contenu ne disparaît pas quand le système n'est plus alimenté

**N.B.** La ROM contient toujours un programme de chargement initial ou un moniteur pour permettre le fonctionnement du système dès la mise sous tension.

**Mémoire vive**

C'est la mémoire de lecture et d'écriture du système ou RAM (Random Access Memory).

**N.B.** Le contenu de ce type de mémoire est perdu lorsque l'on coupe l'alimentation.

**4. Les entrées/sorties**

Les entrées/sorties sont nécessaires pour que le microprocesseur communique avec le monde extérieur

**5. Les bus de communication**

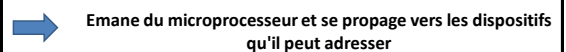
Cette communication est réalisée à l'aide de trois bus qui regroupent les signaux, selon leur fonction.

**5.1. Le bus de données**

C'est un bus bidirectionnel (de 8 bits pour un 6809) sur lequel transitent les données échangées par les éléments du système.

**5.2. Le bus d'adresses**

Il s'agit d'un bus unidirectionnel (de 16 bits pour un 6809)



**Rq** L'adresse qu'il porte permet d'atteindre une case mémoire ou un registre spécifique avec laquelle une opération est désirée.

### 5.3. Le bus de contrôle

C'est un groupe de lignes issues de ou allant vers le microprocesseur et reliant ce dernier à d'autres dispositifs.

➡ Son rôle est de véhiculer les signaux destinés à assurer la synchronisation et la commande de l'ensemble du système.

**Exemple** la ligne Reset et R/W (lecture/écriture), les lignes d'interruptions et l'horloge appartiennent à ce bus.

## Architecture des Ordinateurs Circuits Pour l'Arithmétique Binaire

ENSIAS  
2013-2014

### 1. Addition entière binaire

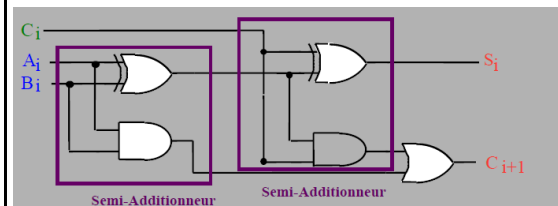
Opération très courante dans un microprocesseur

- ✓ Outre dans l'unité arithmétique
- ✓ Incrémenter le compteur de programme
- ✓ Calculs d'adresses.

➡ Optimisée pour être rapide

Multiplis façons de construire des additionneurs efficaces en temps et en nombre de portes logiques utilisées.

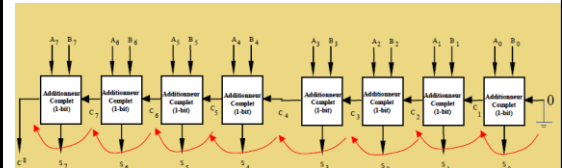
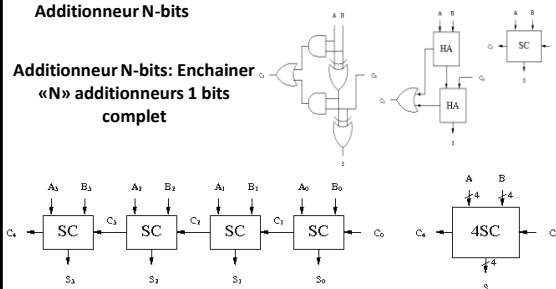
#### Additionneur 1-bit.



N.B. Une répétition du circuit simplifié énormément la conception du circuit

#### Additionneur N-bits

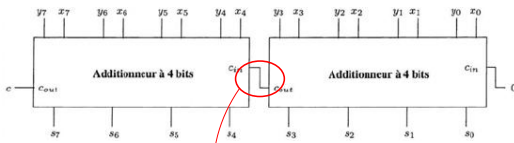
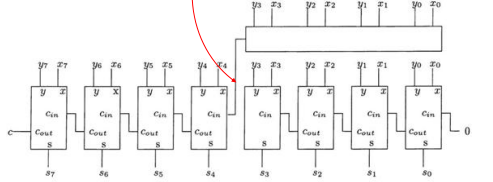
Additionneur N-bits: Enchaîner  
«N» additionneurs 1 bits  
complet



Conséquence: Le temps de propagation (ou le retard) est proportionnel à la taille (n) de l'Additionneur Complet.

Par exemple: Un Additionneur Complet 8-bits est plus lent par rapport à un Additionneur Complet 4-bits, car le temps de la propagation de la Retenue  $C_i$  est deux fois plus pour obtenir le résultat final.

Ce type d'Additionneur s'appelle Additionneur à Propagation de Retenue. Il existe d'autres types d'additionneur plus rapide...

**Amélioration 1:** au lieu d'utiliser un add. 1 bit, on utilise un add. 4 bits.**Amélioration 2:** Accélérateur de la retenue.**Amélioration 3:** Additionneur par anticipation de retenue

Calculer les retenues par un circuit extérieur

 $C_i$  la retenue de l'addition des  $i$  bits de poids faible de A et B.Pour accélérer le calcul des  $C_i$ :

$$C_i = A_i B_i + C_{i-1} \quad (A_i \ominus B_i)$$

$$G_i = A_i B_i \text{ et } P_i = A_i \ominus B_i$$

G (pour Generate en anglais)

P (pour Propagate en anglais).

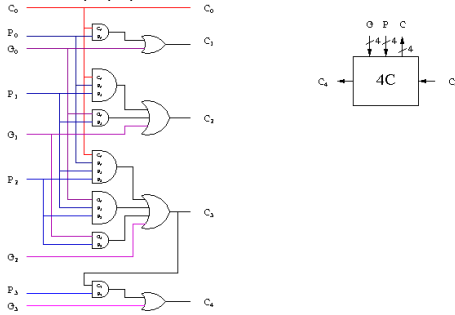
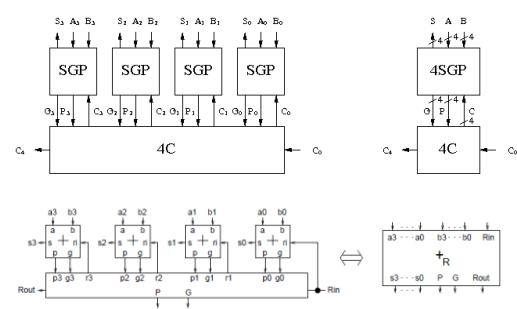
La valeur  $G_i$  est la retenue engendrée par l'addition des deux bits  $A_i$  et  $B_i$ , et la valeur de  $P_i$  détermine si la retenue de  $C_i$  se propage.

 $C_{i+1}$  provient soit :➤ Directement de l'addition des bits  $A_i$  et  $B_i$ ➤ Propagation de la retenue  $C_i$ .

$$C_{i+1} = G_i + P_i C_i$$

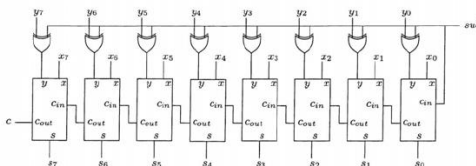
$$C_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} C_{i-1}$$

$$C_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + P_i P_{i-1} P_{i-2} C_{i-2}$$

**Circuit de calcul de  $S_i, G_i, P_i$** **Additionneur 4 bits avec calcul anticipé des retenues****2. Soustraction Binaire**

L'additionneur est capable de traiter des nombres négatifs représentés en complément à deux comme indiqué précédemment.

$x - y$  équivaut au calcul de l'expression  $x + (-y)$ .

**3. Multiplication et division binaire**

La multiplication et la division entière sont encore plus difficiles à implémenter que l'addition.

Il n'existe pas de bon circuit combinatoire itératif.

La solution consiste à utiliser un circuit séquentiel qui effectue une addition par front d'horloge.



### Calcul des indicateurs

Dans tout micro-processeur, il existe des indicateurs ou flags qui sont des registres 1 bit pouvant prendre les valeurs 0 ou 1.

- ✓ Ils sont mis à jour dès qu'un chargement ou une opération logique ou arithmétique est effectuée.
- ✓ Ces indicateurs peuvent ensuite être testés par les branchement conditionnels.

Les principaux indicateurs N, Z, C et O

#### ✓ Indicateur N (pour Négatif)

- Il indique si le résultat est négatif.
- Il est égal au bit de poids fort du résultat (Les entiers sont représentés en compléments à 2).

#### ✓ Indicateur Z (pour Zéro)

- Il indique si le résultat est égal 0.

#### ✓ Indicateur C (pour Carry)

- Il indique si l'opération a provoqué une retenue.
- Il est mis à 1 lorsqu'il y a une retenue.
- Ceci correspond à un débordement pour une addition de nombres non signés.

#### ✓ Indicateur O (pour Overflow)

Lorsque le nombre de bits sur lesquels les nombres sont codés n'est pas suffisant et que le résultat d'une opération n'est pas codable avec le nombre de bits spécifiés

il y a débordement si

les opérandes X et Y sont de même signe, et

$$|X| + |Y| \geq 2^{n-1}$$

	13	01101	-13	10011
	9	01001	-9	10111
<hr/>				
sur n bits	-10	10110	10	01010
sur n+1 bits	22	010110	-22	101010

- comparer le signe des opérandes et le signe du résultat
  - s'ils sont différents, il y a débordement
- comparer la retenue entrante dans le bit de poids fort avec la retenue sortante
  - si elles sont différentes, il y a débordement

## Architecture des Ordinateurs

### Circuits séquentiels

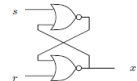
### 1. Bascules et bistables

#### 1.1 Bascules:

SR

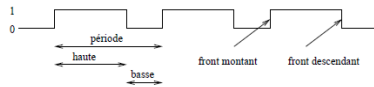
Problème de synchronisation

JK, D



## 1.2 Horloge

Une horloge est un signal périodique. Sa fréquence est l'inverse de sa période (ou temps de cycle).

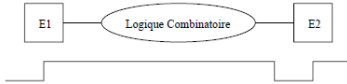


Types de synchronisation:

➤ Synchronisation par niveau

➤ Synchronisation par front d'impulsion

Temps de se stabilisation des blocs de circuits combinatoires



Les "flip-flops" et les "latches" mémoire

L'état et la sortie dépendent de l'histoire de leurs entrées :

- donnée à mémoriser
- ordre de mémorisation

Cet ordre est :

- de type niveau pour le latch,
- de type front pour le flip-flop.

• Latch : un "latch" (= verrou) est une bascule,

- sans horloge pour le latch SR (ou RS),
- ou avec une horloge sensible aux niveaux : latch D ;
- Quelques bascules JK de type "ancien" sont actives sur des niveaux et sont en conséquence des latches.

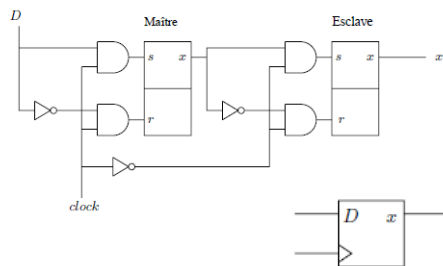
• Flip-Flop : un flip-flop est une bascule avec horloge sensible aux fronts

- la sortie Q du flip-flop reste constante entre deux fronts d'horloge.

## 1.3 BISTABLES

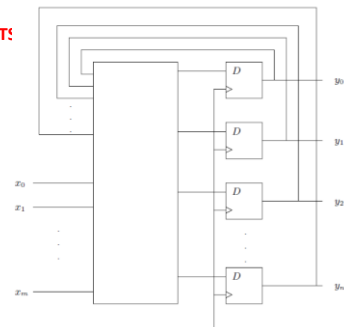
Un bistable est la version synchrone d'une bascule.

Bistable D (en anglais : D flip-flop).



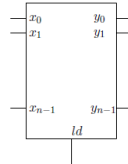
## 2. Circuits séquentiels

### 2.1 TABLE D'ÉTAT



## 2.2. Registres

Un registre est un circuit séquentiel à  $n + 1$  entrées et  $n$  sorties.



## 2.3. Registre à décalage

### 2.3. Compteurs

Diviseur de fréquence,  
Un compteur est un circuit séquentiel (voir chapitre 9) avec 0 entrée et  $n$  sorties.

Ex: Program Counter (PC)

## 2.4 Multiplication Binaire

Additions successives du multiplicande avec lui même décalé

$$\begin{array}{r} 11 \\ \times 101 \\ \hline 11 \\ 00 \\ 11 \\ \hline 1111 \end{array}$$

Résultat codé sur  $2n$  bits pour 2 opérandes sur  $n$  bits

Principe:

- génération de produits partiels
- addition des produits partiels

L'amélioration d'un multiplieur passe par:

- la réduction du nombre de produits partiels
- l'accélération de l'addition des produits partiels

La multiplication consiste à faire une suite d'additions avec le multiplicande décalé vers la gauche. Cette opération est répétée autant de fois qu'il y a d'éléments binaires dans le multiplicateur.

1101	multiplicande
x 1011	multiplicateur
0001101	
0011010	décalage 1 pas
1101000	décalage 3 pas
10001111	résultat

- ✓ 1 registre à décalage pour le multiplicande
- ✓ 1 registre à décalage pour le multiplicateur
- ✓ 1 registre pour le résultat
- ✓ 1 additionneur
- ✓ 1 compteur pour le nombre de pas de décalage

#### 2.4.1 Multiplication séquentielle

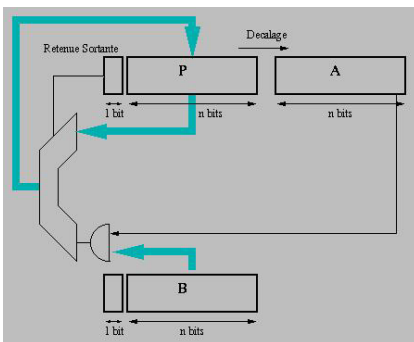
$A \times B$  sur  $n$  bits non signée

- un registre A contient  $A = a_{n-1} \dots a_1 a_0$
- un registre B contient  $B = b_{n-1} \dots b_1 b_0$
- un registre P contient les produits partiels

le résultat

- P contiendra les bits de poids fort
- A contiendra les bits de poids faibles

Schéma synoptique



algorithme

1.  $P \leftarrow 0$

2. faire  $n$  fois

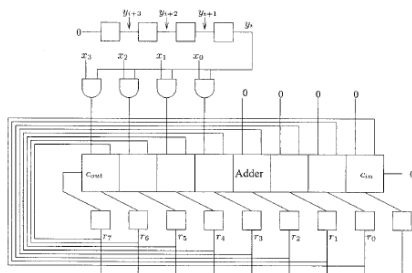
(a) si le chiffre de poids faible de A est 1 alors

$P \leftarrow P + B$

(b) décalage des registres P et A vers la droite

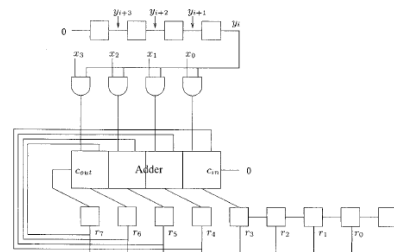
- le bit de poids fort de P reçoit la retenue sortante
- le bit de poids faible de P est transféré dans le bit de poids fort de A
- l'ancien bit de poids faible de A est perdu

Multiplication: circuit 0

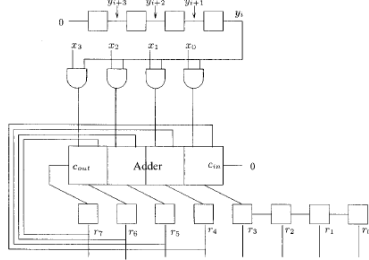


Multiplication: circuit 1

la partie la moins significative de l'additionneur additionne toujours r3, r2, r1 et r0 avec 0

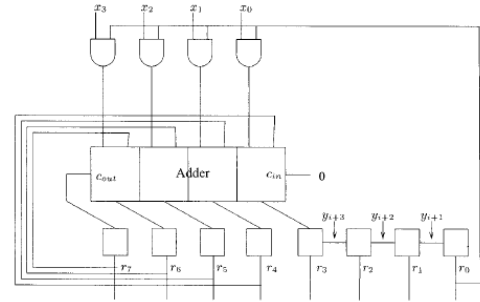


## Multiplication: circuit 2



- les bistables contenant  $r_{n-1}$  à  $r_0$  et les  $n$  bistables supérieurs ne contiennent jamais simultanément des informations.
- Initialement, tous les bistables supérieurs contiennent des informations utiles, et aucun des  $n$  bistables inférieurs le plus à droite n'en contient.
- Après la première étape,  $r_{n-1}$  contient une information utile, mais le bistable supérieur le plus à gauche n'en contient plus.

## Multiplication: circuit final



## 2.4.2 Exercice

Donner le schéma réalisant la multiplication par des circuits combinatoires

Utiliser un circuit pour chaque élément  $x_i y_j$

				$x_3$	$x_2$	$x_1$	$x_0$
				$y_3$	$y_2$	$y_1$	$y_0$
				$x_3y_0$	$x_2y_0$	$x_1y_0$	$x_0y_0$
		$x_3y_1$		$x_2y_1$	$x_1y_1$	$x_0y_1$	
	$x_3y_2$	$x_2y_2$		$x_1y_2$	$x_0y_2$		
$x_3y_3$	$x_2y_3$	$x_1y_3$	$x_0y_3$				
$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$	$p_0$

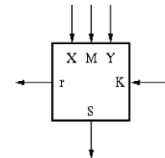
Chaque élément  $x_i y_j$  :

- multiplie  $x_i$  par  $y_j$  ( $x_i \cdot y_j$ )
- additionne la somme partielle provenant de  $x_{i-1} y_{j-1}$
- tient compte de la retenue provenant de  $x_{i-1} y_j$
- envoie la retenue à  $x_{i+1} y_j$
- envoie la somme à  $x_{i-1} y_{j+1}$

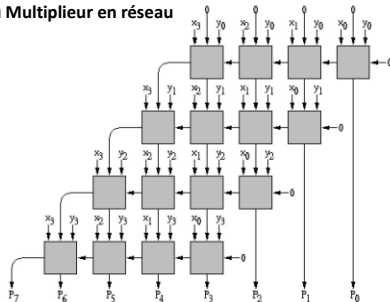
$$(r, S) = X \times Y + K + M$$

S est la somme

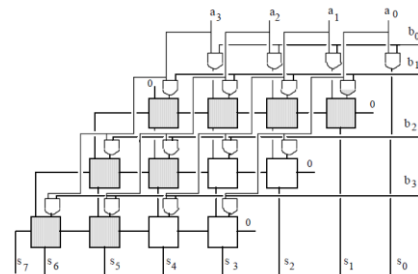
r est la retenue sortante



## Schéma du Multiplier en réseau



## Schéma2 du Multiplier en réseau

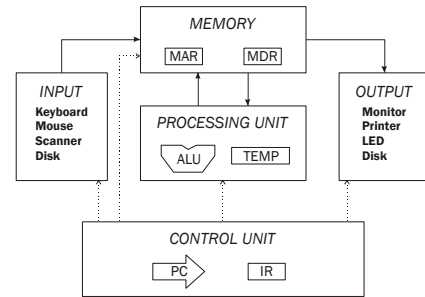


## Architecture des Ordinateurs

### Logique à trois états

73

## Von Neumann Model

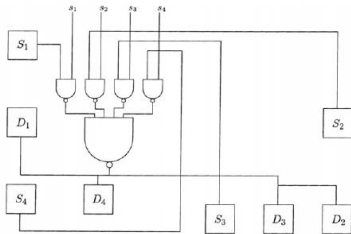


Ce transparent est tiré du cours de Pr. Jacques Lonchamp IUT Nancy Charlemagne

74

### 1. Introduction

Ils manipulent des signaux pouvant être dans l'un des trois états possibles, et non plus seulement 0 et 1.



Cette solution nécessite que toutes les sorties soit routées vers un endroit central.

Connecter les sorties de deux ou plusieurs circuits ensemble peut détruire ceux-ci.

75

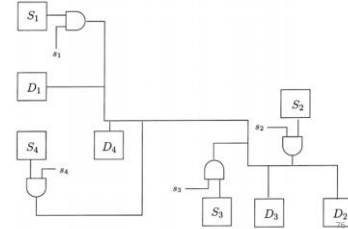
Pour résoudre le problème, il faut utiliser des circuits à *trois états*.

**Entrée Enable:**

*Enable = 1, le circuit se comporte comme le circuit ordinaire correspondant.*  
*Enable = 0, les sorties sont complètement déconnectées du reste du circuit.*

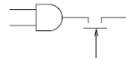
### 2. Retour vers les Transistors

Puisque au plus une source à la fois n'est active: une solution similaire



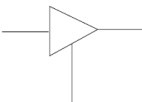
### 2.1 Interrupteur

N'importe quel circuit peut exister en version à trois états



### 2.2 Pilote de bus unidirectionnel

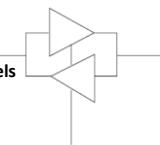
Copie simplement ses entrées sur ses sorties, mais qui a une entrée *enable*



Pilote de bus (en anglais : bus driver).

### 2.3 Pilote de bus bidirectionnel

Version pour des signaux bidirectionnels



77

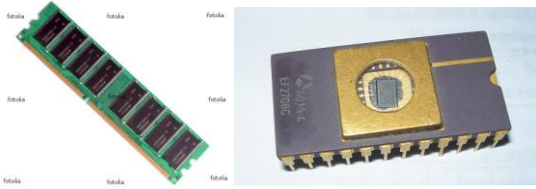
## Architecture des Ordinateurs

### Mémoires

ENSIAS  
2013-2014

78

## Les Mémoires



- Assemblage de Transistors et condensateur
- Circuit décodeur d'adresse
- Exploiter pour construire les:
  - ✓ Registre de processeur,
  - ✓ La mémoire centrale
  - ✓ Les ports d'entrées/sorties (Unités d'échanges)

79

## Mémoriser

Enregistrer	Séquentiel	Liste ou Pile bande magnétique Mémoire de masse
Suivant quel processus ?	Aléatoire	RAM
	Electronique	RAM ROM
Conserver	Magnétique	Disque dur
Sur quel support physique ?	Optique	CD, DVD
	Autres...(quantique)	
Restituer	Rarement	Mémoire de masse ( HD)
A quelle fréquence ?	Souvent	RAM dynamique ( SDRAM sur CM)
	Rapidement	RAM statique (cache 2ème niveau)
(Suivant quel processus ? Voir « Enregistrer »)	Très rapidement	RAM statique (cache 1er niveau)

## Architecture des Ordinateurs Mémoires Vives

81

### 1. Introduction:

- ✓ Mémoires vives
- ✓ Mémoires à lecture et écriture
- ✓ Accès aléatoire (RAM : Random Acces Memory)
- ✓ Temps d'accès à l'information est indépendant de sa place en mémoire.

L'information élémentaire, ou bit (binary digit), est mémorisée dans une cellule ou point mémoire.

Ces cellules sont groupées en mots de  $n$  bits, c'est-à-dire que les  $n$  bits sont traités (écrits ou lus) simultanément.

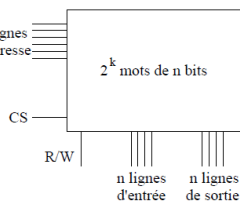
On ne peut pas modifier un seul bit!!!

82

Pour pouvoir identifier individuellement chaque mot on utilise  $k$  lignes d'adresse.

Les lignes d'entrées/sorties  
→ bidirectionnel

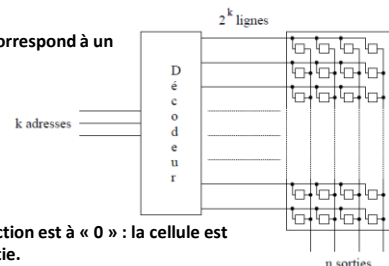
Sélection du bloc (CS)



83

### 2. Adressage bidimensionnel ou matriciel

Chaque ligne correspond à un mot de  $n$  bits



Si la ligne de sélection est à « 0 » : la cellule est "isolée" de la sortie.

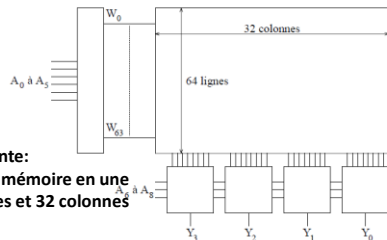
Si la ligne de sélection est à « 1 » : l'information mémorisée se retrouve sur la ligne de sortie

Architecture très simple n'est pas la plus économique en terme de nombre de portes

84

**Ex:** 512 mots de 4 bits soient 2048 bits.  
 $k = 9$  et  $n = 4 \rightarrow$  512 portes ET pour réaliser le décodeur

➤ Architecture adressage X-Y ou bidimensionnel



Economie importante:

Organisation de la mémoire en une matrice de 64 lignes et 32 colonnes (2048 = 64 x 32).

- ✓ 64 portes ET sont nécessaires pour le décodeur
- ✓ 9 portes (8 ET et 1 OU) pour chacun des multiplexeurs
- Total 64 + (9 x 4) = 100 portes

85

Le minimum se situe pour une organisation "carrée", pour laquelle les nombres de lignes et de colonnes sont égaux ou différent d'un facteur 2.

nb lignes	nb colonnes	Décodeur	Multiplexeur	Total
512	4	512	0	512
256	8	256	3	268
128	16	128	5	148
64	32	64	9	100
32	64	32	17	100
16	128	16	33	148
8	256	8	65	268

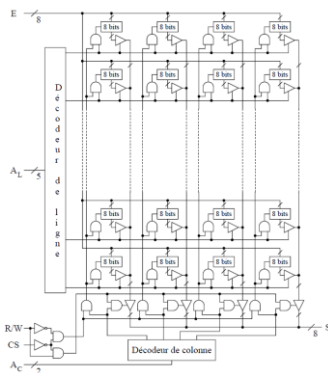
Caractéristiques mémoire: capacité et le format.

Symbole	Préfixe	Capacité
1 k	(kilo)	$2^{10} = 1024$
1 M	(méga)	$2^{20} = 1048576$
1 G	(giga)	$2^{30} = 1073741824$
1 T	(tera)	$2^{40} = 1099511627776$

Capacité =  $2^k$  mots =  $2^k \times n$  bits

86

Autre organisation logique possible pour une mémoire de 128 mots de 8 bits.



87

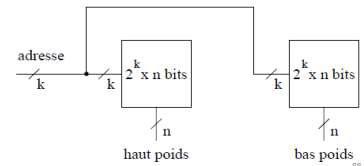
### 3. Assemblage de blocs mémoires

#### 3.1 Augmentation de la longueur de mots

Les techniques d'intégration ne permettent pas d'obtenir des boîtiers ayant des capacités ou des formats suffisants pour toutes les applications.

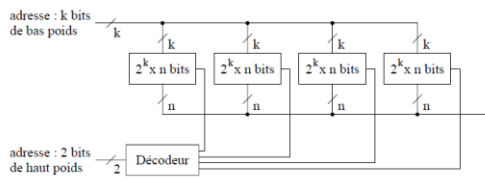
Nécessaire d'associer plusieurs boîtiers:

- ✓ Augmenter la longueur des mots ou le nombre de mots.
- ✓ Améliorer les performances temporelles de la mémoire en faisant fonctionner plusieurs blocs en parallèle.



88

#### 3.2 Augmentation du nombre de mots



4 x  $2^k$  mots de  $n$  bits à l'aide de 4 boîtiers de  $2^k \times n$  bits

Les deux bits de haut poids attaquent un décodeur à quatre sorties

Ces quatre lignes permet de sélectionner un boîtier (entrée de validation du boîtier : CS)

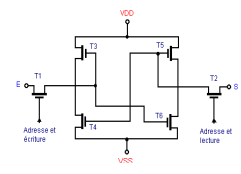
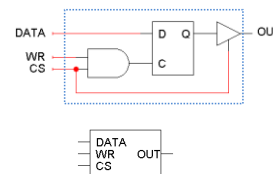
89

### 3.6 Les deux familles de mémoires vives

#### 3.6.1 RAM statiques (SRAM)

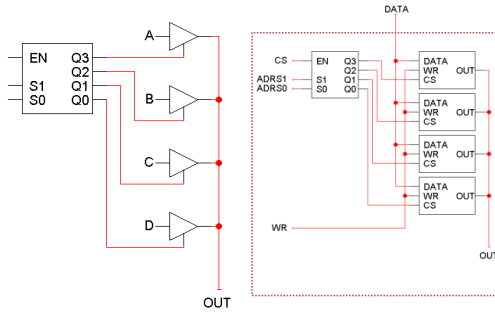
➤ Bascules du type D ou R-S

➤ Bascules garantissent la mémorisation de l'information aussi longtemps que l'alimentation électrique est maintenue sur la mémoire.



90

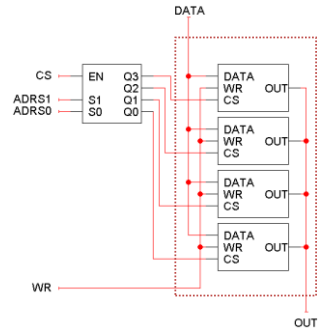
## Première mémoire



91

## Amélioration?

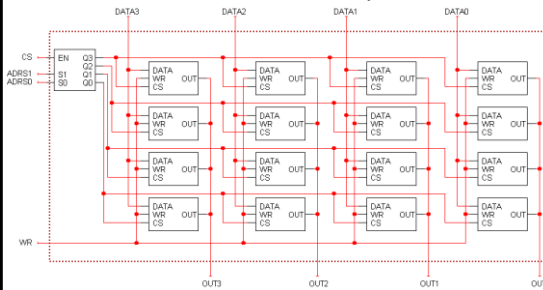
- Voici 4 x 1 RAM de nouveau.
- Comment pouvons-nous faire une mémoire (un souvenir) "plus large" avec plus de morceaux (bits) par mot,
- comme peut-être 4 x 4 RAM ? Dupliquez du truc (de la substance) dans la boîte bleue!



92

## RAM 4 x 4

- Les DATA et les OUT sont maintenant sur quatre blocs
- Possibilité de lire et écrire des mots de quatre bit.



93

## Grande RAMs à partir de petite RAMs

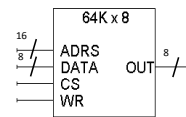
On utilise de petites RAMs comme des composants pour faire de plus grandes mémoires..

Supposez à titre d'exemple, que nous avons environ

64Ko x 8 RAMs :

- $64K_o = 64 \times 1024 = 2^6 \times 2^{10} = 2^{16}$ , ainsi il y a 16 lignes d'adresse.

- Il y a 8 lignes de données.



94

## Fabrication d'une plus grande mémoire

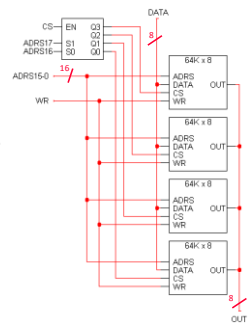
- On peut utiliser quatre blocs 64K x 8 chips Ensemble pour faire une mémoire 256K x 8.
- Pour 256K mots, nous avons besoin de ?? Lignes d'adresses.



95

## Fabrication d'une plus grande mémoire

- Pour 256K mots, nous avons besoin de 18 lignes d'adresses.
  - Les deux lignes d'adresses du poids fort seront utilisées par le décodeur, pour sélectionner un parmi quatre des blocs 64K x 8 RAM.
  - Les autres 16 lignes d'adresses sont utilisées pour l'adressage des blocs 64K x 8.

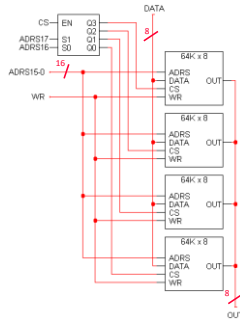


96



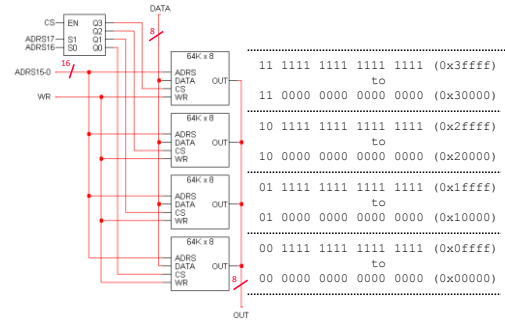
### Analyse de la RAM 256K x 8

- Une mémoire de 256Ko, réalisée à partir de 64Ko
- Quand les deux bits d'adressage de poids fort les plus significatifs de l'adresse sont 00, la RAM est choisie. Il tient des données pour les premières adresses de 64Ko.
- On passe à la RAM suivante en haut quand l'adresse commence par 01. Il tient des données pour les deuxièmes adresses de 64Ko.
- La troisième RAM est choisie par 10.
- La dernière RAM contient les données des adresses de 64Ko finales.



97

### Zones d'adresses

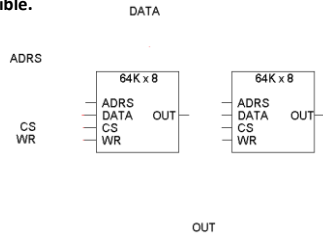


98

### Réaliser une large mémoire

Une mémoire 64K x 16 RAM, réalisée à partir de deux mémoire 64K x 8.

- Le circuit à gauche contient les 8 bits de données du poids fort.
- Le circuit à droite contient les 8 bits de données de poids faible.

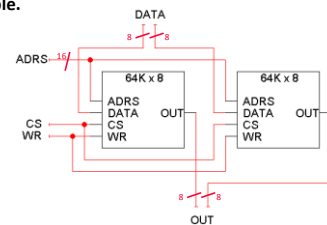


99

### Réaliser une large mémoire

Une mémoire 64K x 16 RAM, réalisée à partir de deux mémoire 64K x 8.

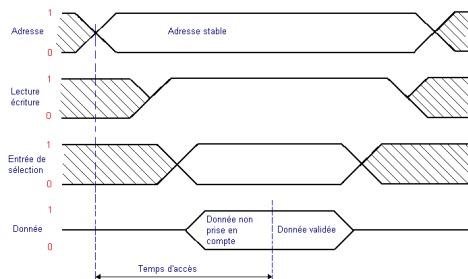
- Le circuit à gauche contient les 8 bits de données du poids fort.
- Le circuit à droite contient les 8 bits de données de poids faible.



100

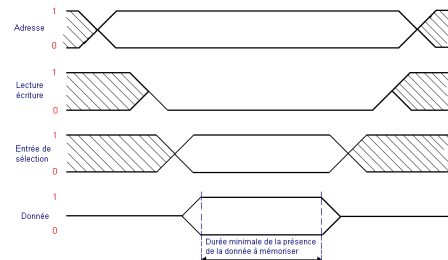
### 3.5 Cycles de fonctionnement d'une mémoire

#### 3.5.1 Chronogramme de lecture d'une mémoire:



101

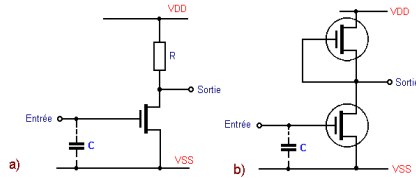
#### 3.5.2 Chronogramme d'écriture d'une mémoire:



102

### 3.6.2 RAM dynamiques (DRAM)

- Élément de mémorisation est constitué par un condensateur et un transistor à effet de champ
- Le transistor joue le rôle d'un interrupteur commandé.
- L'information est mémorisée sous la forme d'une charge électrique stockée dans le condensateur



La technique DRAM permet :

- une plus grande densité d'intégration (environ deux à quatre fois moins de place que dans une mémoire statique).

103

Par contre :

- Courants de fuite le condensateur a tendance à se décharger.
- RAM dynamiques doivent être rafraîchies régulièrement pour entretenir la mémorisation
- Lire l'information avant qu'elle n'ait totalement disparu et de la recharger
- la lecture étant destructive (nécessaire de restaurer la charge électrique à la fin de l'opération)

104

Conséquences du rafraîchissement :

- Tout d'abord il complique la gestion des mémoires dynamiques la durée de ces actions augmente le temps d'accès aux informations.
- Le temps d'attente des données est variable selon que la lecture est interrompue ou non par des opérations de rafraîchissement et la quantité de cellules à restaurer.
- Placer dans le cas le plus défavorable pour déterminer le temps d'accès à utiliser en pratique.

105

### Récapitulatifs:

Les mémoires dynamiques:

- Grande densité d'information,
- Coût par bit plus faible,
- Utilisées pour la mémoire centrale

Les mémoires statiques:

- Plus rapides, sont utilisées pour les caches et les registres.

106

### 3.7 Le rafraîchissement des mémoires dynamiques

- Capacité se décharge naturellement
  - Se décharge lors d'un cycle de lecture
- ➡ Constamment les recharger (rafraîchir)

### 3.8 Rapidité des mémoires

- ✓ Temps d'accès
- ✓ Temps de cycle de lecture.

!!!

Vitesse de fonctionnement des processeurs  
Vitesse d'accès des mémoires

107

### 3.9 Mémoire asynchrone

Le processeur ne peut pas engager un nouvel accès tant que l'accès précédent n'est pas achevé.

- ✓ L'intervalle de temps entre deux accès mémoire consécutif n'est pas régulier.
- ✓ Le processeur ne sait donc pas quand l'information qu'il attend est disponible et doit attendre (wait-state) que la mémoire lui transmette les données.

Exemple mémoire asynchrone

- **FPM** (Fast Page Mode) : a été utilisée principalement sur les machines équipées d'un processeur < à 486 Mhz.
- **EDO** (Extended Data Out) : Elle a été une amélioration de la FPM. Elle a été utilisée pour des fréquences de bus à 66 Mhz.

108

### 3.10 Mémoire synchrone

Le processeur peut engager des accès consécutifs si l'accès précédent n'est pas achevé.

la cadence de sortie des informations est régulière, on évite ainsi les états d'attente (wait state) du processeur.

Fonctionnement synchrone c-à-d un signal d'horloge

- Mémoires dynamiques synchrones (SDRAM)
- Mémoires statiques synchrones (SSRAM)

Parmi les bornes de la mémoire, nous retrouvons un signal d'horloge sur lequel les différents accès à la mémoire seront synchronisés.

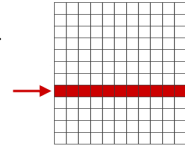
109

Plus on cherche à obtenir une grande quantité d'informations rapidement et plus on risque de rencontrer une erreur.

**Les Timings:** Les timings sont des "délais de sécurité" destinés à éviter les erreurs au cours d'une opération de lecture ou d'écriture...

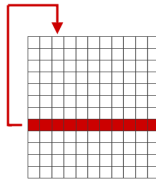
- Accès au mémoire se fait suivant plusieurs étapes
- Chacune de ces étapes nécessitant un temps
- Il y a plusieurs étapes et donc plusieurs temps
- ✓ Cycle time : délai pour un nouveau cycle d'accès.

✓ Le RAS (Row Address Strobe) : c'est le temps nécessaire pour sélectionner une ligne.

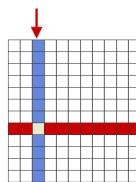


110

✓ Le RAS to CAS (Column Address Strobe) : c'est le temps nécessaire pour passer du mode de sélection de lignes au mode de sélection de colonnes.



✓ Le CAS : c'est le temps nécessaire pour sélectionner une colonne



Timings mémoire: les timings dans l'ordre suivant :

- ❖ CAS
- ❖ RAS to CAS
- ❖ RAS
- ❖ Cycle time

111

### 3.11 Performance

Le temps passé à attendre une réponse de la mémoire (**attente mémoire**) a un impact fondamental sur le temps d'exécution d'un programme:

Temps d'exécution = (Nbre cycles d'exécution + Nbre cycles d'attente mémoire) \* Temps de cycle

La **pénalité d'accès** est le temps (nombre des cycles) nécessaire pour transférer une donnée de la mémoire au processeur.

Le **cache** est un moyen de réduire la pénalité d'accès.

112

### 3.12 Type de DRAM:

#### DR-SDRAM (Direct Rambus DRAM)

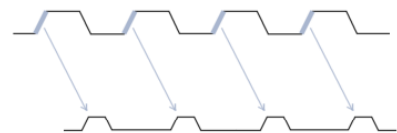
Permet de transférer les données sur un bus de 16 bits de largeur à une cadence de 800Mhz, ce qui lui confère une bande passante de 1,6 Go/s.

#### DDR-SDRAM (Double Data Rate SDRAM)

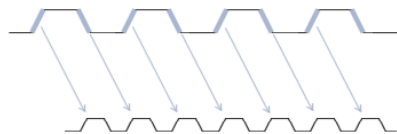
- Une mémoire basée sur la technologie SDRAM
- Permet de doubler le taux de transfert de la SDRAM à fréquence égale.
- Les mémoires DRAM standard utilisent une méthode appelé SDR (*Single Data Rate*) consistant à lire ou à écrire une donnée à chaque front montant.
- La DDR permet de doubler la fréquence des lectures/écritures, avec une horloge cadencée à la même fréquence, en envoyant les données à chaque front montant, ainsi qu'à chaque front descendant.

113

SDR RAM



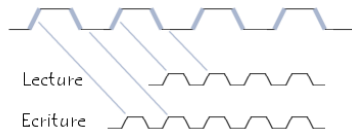
DDR RAM



114

**DDR2-SDRAM**

- Permet d'atteindre des débits deux fois plus élevés que la DDR à fréquence externe égale.
- On parle de QDR (*Quadruple Data Rate* ou *quad-pumped*)
- La mémoire DDR2 utilise deux canaux séparés pour la lecture et pour l'écriture.



La DDR2 possède également un plus grand nombre de connecteurs que la DDR classique (240 pour la DDR2 contre 184 pour la DDR).

115

**DDR3-SDRAM**

- Le DDR3 SDRAM améliore les performances par rapport au DDR2, mais surtout diminue la consommation électrique
- En effet, celle-ci est de 40 % inférieure, en particulier grâce à une baisse du voltage utilisé, une finesse de gravure

116

**3.12 RAM à usage spécifique:****Mémoire graphique**

- Utiliser sur les cartes graphiques.
- Elle ne se présente pas sous la forme de barrettes à insérer, mais en circuits intégrés directement soudés à la carte graphique.
- La mémoire graphique doit obligatoirement permettre la lecture et l'écriture simultanées
- La mémoire graphique reçoit les informations à stocker du processeur graphiques.

117

**➤ La mémoire VRAM (video RAM)**

- ✓ Conçue pour être utilisée sur les cartes graphiques.
- ✓ Plus rapide que la mémoire DRAM classique, sa fréquence atteint 80 MHz et son temps d'accès est de 20 à 25 ns.

**➤ La mémoire WRAM**

- ✓ Amélioration de la VRAM.
- ✓ La bande passante est améliorée de 25%.

**➤ La mémoire SGRAM ("Synchronous Graphic RAM")**

- ✓ Adaptation de la SDRAM à un usage graphique.
- ✓ Elle ne permet pas la lecture et l'écriture simultanées. En revanche, elle permet la récupération et modification de données par blocs entiers (mode "rafales").

118

**3.13 Correction d'erreurs de la mémoire vive**

- 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.

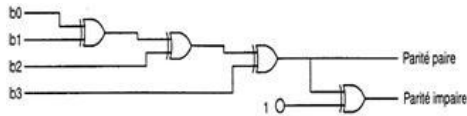
119

**Bit de parité**

- 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.

120

Circuit de génération ou de vérification de parité d'un mot binaire de 4 bits



121

### Barrettes ECC

- 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.

122

### Barrettes avec registre ou tampon (registered ou buffered)

- Ces barrettes ont un registre entre les puces de DRAM et le contrôleur mémoire du système (dans le chipset ou dans le processeur).
- Ce registre retient les données pendant un cycle d'horloge avant qu'elles ne soient envoyées vers le contrôleur mémoire.
- Ce processus augmente la fiabilité du transfert de données, au détriment du temps de traitement, en retard d'un cycle d'horloge par rapport à de la mémoire sans registre.
- Ces modules de mémoire avec registre ne sont généralement utilisés que dans les serveurs.

123

### NV-RAM → Non Volatile RAM.

SRAMs et DRAMs sont des RAM volatiles. NV-RAMs ne le sont pas (comme ROM) et la CPU peut y écrire et lire (comme S/D RAM).

NV-RAMs consomment moins de puissance puisqu'elles utilisent les mêmes CMOS-cellules des SRAMs.

NV-RAMs utilisent des batteries internes (Lithium) comme source d'énergie.

NV-RAMs utilisent un circuit de contrôle intelligent qui permet de connecter VCC aux batteries en cas  
Ou l'alimentation extérieure cesse pour la remplacer par celle des batteries internes.

→ Prix/Cellule des NV-RAMs est extrêmement plus cher comparé aux autres RAMs.

A noter que le maintien de l'énergie peut durer 10 ans après l'extinction de l'alimentation extérieure.

124

## Architecture des Ordinateurs

### Mémoires Mortes

125

### 1. Différents types de PROM

#### 1.1 ROM :

- Ces mémoires ont été écrites une fois par le fabricant.
- On peut lire les informations contenues mais on ne peut les modifier.

#### 1.2 PROM (Programmable ROM) :

##### FROM (Fuse PROM)

- Ces mémoires sont livrées non enregistrées par le fabricant.
- Une fois programmées, on ne peut plus modifier leur contenu.

##### OTP (One Time Prom)

- On ne peut les programmer qu'une seule fois.

126

**EPROM (Erasable PROM)**

- Ces mémoires possèdent les avantages de la PROM avec un plus qui est l'effacement des données par
- L'utilisateur et la possibilité de reprogrammer.

**2. Différents types d'EPROM****UVEPROM (ou EPROM)**

- Effaçable aux UV (10 à 20 minutes d'exposition).
- Effacement total de la mémoire.
- Programmation par tension de 25V.

127

**EEPROM (Electrically EPROM)**

- Effaçable et programmable électriquement.
- Effacement adresse par adresse.
- Coût de fabrication élevé.

**EPROM FLASH**

- Effaçable électriquement.
- Effacement total de la mémoire.
- Plus rapide et moins cher que l'EEPROM.

128

**3. Application des mémoires**

- Les applications sont nombreuses.

Citons les cartes à puce, les calculatrices...

- Les mémoires sont utilisées en association avec les circuits de traitement numérique :

microprocesseur ou microcontrôleur.

129

## Architecture des Ordinateurs

### Le premier ordinateur

130

**1. Éléments de base**

Nous présentons ici les éléments de base nécessaires à la construction de tout ordinateur.

**1.1 Compteur**

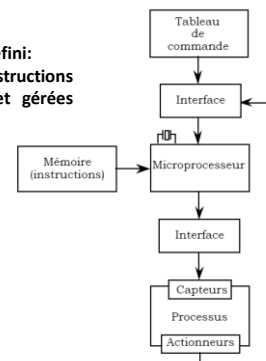
- Compteur avec mise à zéro
- Registre compteur
- Registre compteur avec mise à zéro
- Registre compteur avec mise à zéro et incrémentation

**1.2 Micro-Mémoire****1.3 Décodeur d'instructions**

131

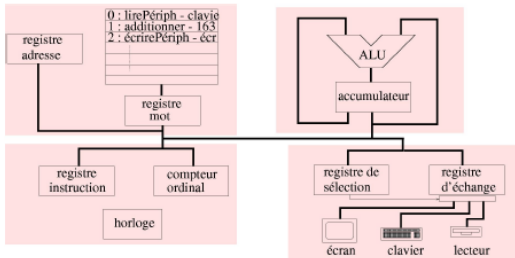
**Schéma synoptique:**

Le fonctionnement est défini:  
Une suite ordonnée d'instructions stockées en mémoire et gérées par cet élément.



132

### La machine complète

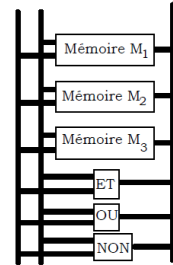


133

### Principe de la logique programmée

Illustration à partir d'un circuit simple constitué de 3 cases mémoires et 3 portes logique ET, OU et NON.

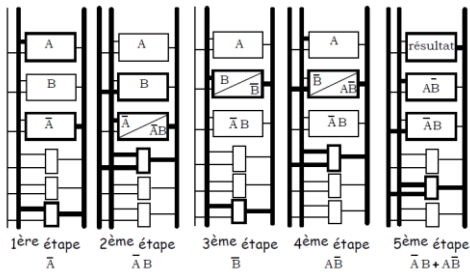
On se propose de réaliser la fonction : A exclusif B



134

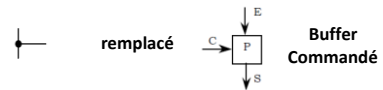
Initialisation [M1] = A et [M2] = B

Déroulement :



135

### Réalisation

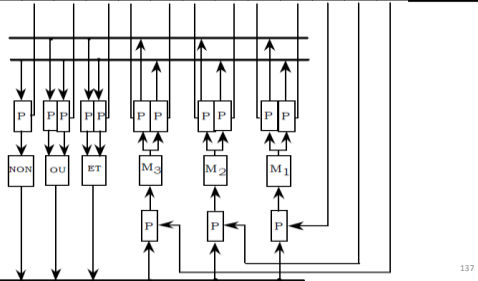


C = 0 porte bloquée

C = 1 porte transparente donc S = E

136

1	1	0	0	0	0	0	0	0	1	0	0	1	$\bar{A}$
2	0	0	1	0	1	1	0	0	0	0	0	1	$\bar{A}B$
3	1	0	0	0	0	0	0	1	0	0	0	1	$\bar{B}$
4	0	0	1	0	0	0	0	1	1	0	0	1	$A\bar{B}$
5	0	1	0	0	1	1	0	0	0	0	0	1	$\bar{A}B + A\bar{B}$



137

### La notion de programme

Un programme qui réalise une fonction particulière comprend :

- Une suite d'instructions.
- Chaque instruction est constituée de plusieurs microinstructions.
- Chaque micro-instruction génère plusieurs micro-commandes destinées à aiguiller correctement les informations.

138

Dans l'exemple du ou exclusif, le programme comprend une instruction constituée de 5 micro-instructions ou phases. Chaque phase génère des micro-commandes qui, au travers des 12 fils, aiguillent correctement les informations.

#### Remarques :

- les phases sont commandées par une logique séquentielle synchrone.
- Les micro-instructions sont stockées sous forme de mots dans une mémoire.

139

## 1.4 Le microprocesseur

le microprocesseur:

- Noté aussi M.P.U. (Microprocessor unit) ou encore C.P.U. (Central Processing Unit)
- Circuit intégré complexe appartenant à la famille des VLSI (Very large scale intégration)
- Capable d'effectuer séquentiellement et automatiquement des suites d'opérations élémentaires.

140

Ce circuit remplit deux fonctions essentielles:

le traitement des données:

- Cette fonction est dédiée à l'U.A.L.
- Elle concerne la manipulation des données sous formes de transfert, opérations arithmétiques, opérations logiques....

le contrôle du système:

- Cette fonction se traduit par des opérations de décodage et d'exécution des ordres exprimés sous forme d'instruction.

141

## Puissance d'un microprocesseur

La notion de puissance est la capacité de traiter un grand nombre d'opérations par seconde sur de grands nombres et en grande quantité.

La puissance se joue donc sur les trois critères suivants:

- La longueur des mots : données et instructions (on parle de largeur du bus des données).
- Le nombre d'octets que le microprocesseur peut adresser (on parle de largeur du bus des adresses).
- La vitesse d'exécution des instructions liée à la fréquence de fonctionnement de l'horloge de synchronisation exprimée en MHz.
- l'aspect dimensionnel renseigne assez bien de la puissance du composant.

142

## Éléments d'un microprocesseur

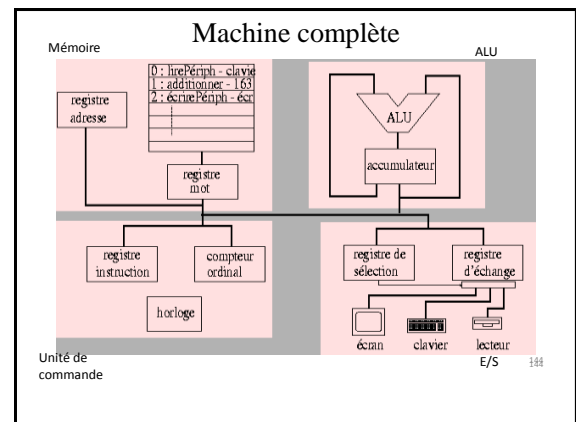
On distingue 3 éléments logiques principaux :

- Une Unité Arithmétique et Logique (U.A.L.)
- Un Accumulateur.

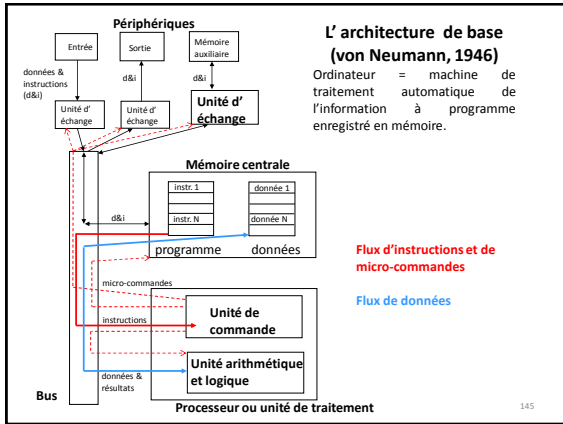
et

- Des registres que l'on nomme couramment :
  - ✓ Le Compteur d'Instructions (C.I.)
  - ✓ Le Registre d'état
  - ✓ Le Registre d'Instructions (R.I.)
  - ✓ Le Registre d'Adresses (R.A.)
  - ✓ Le Registre temporaire des données de base

143



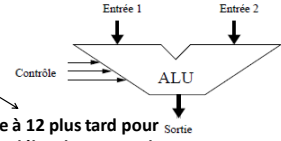




### 1.4 Unité Arithmétique et Logique (ALU)

Une **unité arithmétique et logique (ALU)** d'un processeur est un **circuit** capable d'effectuer les opérations de base comme l'addition, la soustraction, le **ou** et le **et** **bit-à-bit**, etc.

- Deux entrées de 8 bits
- Sortie de 8 bits
- Trois fils de contrôle.



Sera étendue à 12 plus tard pour la gestion des débordement et de la retenue

146

### Contrôle de l'ALU

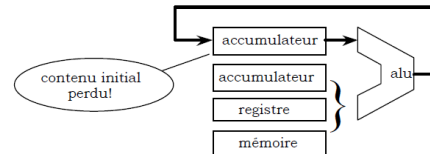
Combinaison	Nom	Valeur de la sortie
000	copy	la première entrée
001	shl	la 2 <sup>e</sup> entrée décalée à gauche d'une position
010	shr	la 2 <sup>e</sup> entrée décalée à droite d'une position
011	add	la somme des deux entrées
100	sub	la différence des deux entrées
101	and	le <b>et</b> logique entre les deux entrées
110	or	le <b>ou</b> logique entre les deux entrées
111	not	le <b>non</b> logique de la deuxième entrée

147

### L'accumulateur

C'est le registre le plus important du microprocesseur, il sert systématiquement lorsque le  $\mu p$  a besoin de "manipuler" des données.

La plupart des opérations logiques et arithmétiques sur les données font appel au couple "UAL - accumulateur" selon la procédure suivante:



148

Il en est de même pour les déplacements et transferts des données d'un endroit à un autre comme :

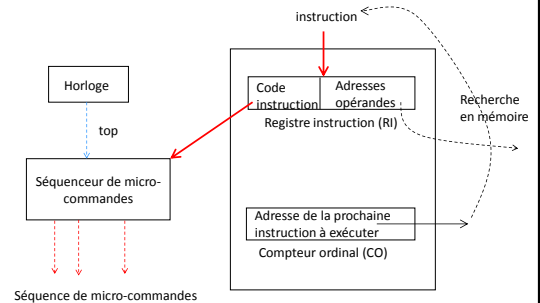
- de mémoire à mémoire.
- de mémoire à unités d'entrée-sortie (I/O).

Cette action se fait en deux temps :

Source vers Accumulateur et ensuite Accumulateur vers destination.

149

### Unité de commande (UC)



Ce diapositive est tiré du cours de Pr. Jacques Lonchamp IUT Nancy Charlemagne

150



**Micro-programme pour l'instruction NOP**

Utilisée pour des raisons d'alignement, ou pour une modification temporaire d'un programme.

En fait, le micro-programme de cette instruction sera terminé par une valeur de 1 pour MOP1 afin de garantir le bon chargement de l'instruction suivante.

Adresse	Contenu	Opération
000000	0110101000000000	fetch
000001	1000000000000000	NOP

157

**Micro-programme pour l'instruction LDIMM (load immediate)**

- ✓ Pour adresser la mémoire à partir de PC, nous utilisons MOP5.
- ✓ De plus, il nous faut MOP3 pour la lecture de la mémoire centrale,
- ✓ ainsi que MOP11 pour stocker la valeur lue dans R0.
- ✓ Par convention, nous allons aussi incrémenter PC pour qu'il contienne le code de l'instruction suivante.
- ✓ Afin de pouvoir charger l'instruction suivante, nous mettons MOP1 à 1 comme dans le cas de l'instruction NOP.

Voici donc le contenu de la micro-mémoire :

Adresse	Contenu	Opération
000000	0110101000000000	fetch
000001	1000000000000000	NOP
000010	101010100010000	LDIMM

158

**Micro-programme pour l'instruction LDm: Sur 2 cycles****➤ 1<sup>er</sup> cycle**

- ✓ Utiliser l'adresse contenue dans PC pour adresser la mémoire (MOP 5 et 3)
- ✓ Charger la valeur se trouvant à cette adresse de la mémoire dans le registre d'adresse à l'aide de MOP9.
- ✓ Simultanément, incrémenter PC (MOP7).

**➤ 2<sup>ème</sup> cycle**

- ✓ Utiliser le contenu du registre d'adresse pour adresser la mémoire centrale (MOP 10 et 3)
- ✓ Charger le contenu dans R0 (MOP11)
- ✓ Afin de pouvoir charger l'instruction suivante, nous mettons MOP1 à 1.

159

Adresse	Contenu	Opération
000000	0110101000000000	fetch
000001	1000000000000000	NOP
000010	101010100010000	LDIMM
000011	001010101000000	LD
000100	101000000110000	LD

160

**2.2 Contenu de la micro-Mémoire**

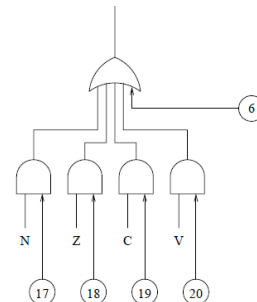
- Chargement et décodage d'une instruction
- Micro-programme pour les instructions
- Micro-programmes pour les instructions arithmétiques

**2.3 Récapitulatif**

Exécuter un programme simple:

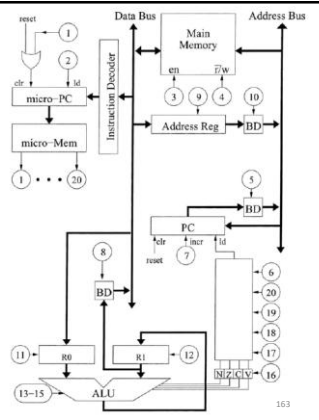
- Opérations de transfert de données entre la mémoire et les registres
- Opérations arithmétiques
- Opération pour modifier le contenu du compteur ordinal permettant ainsi des boucles.

161

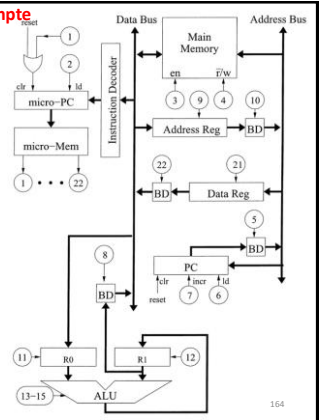
**3. Extensions du premier ordinateur****3.1 Sauts conditionnels**

162

### 3.2 Version avec prise en compte des sauts conditionnels

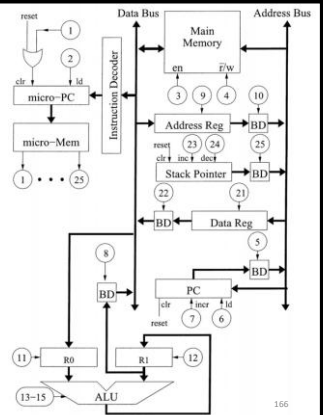


### 3.3 Version avec prise en compte de l'instruction JSR



Ce transparent est tiré du cours de Pr.  
Jacques Lonchamp IUT Nancy Charlemagne

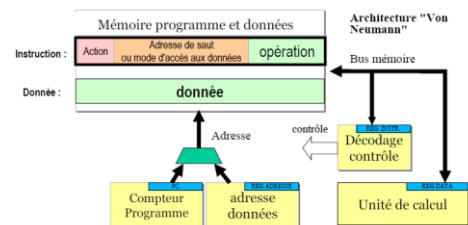
### 3.4 Version avec pile

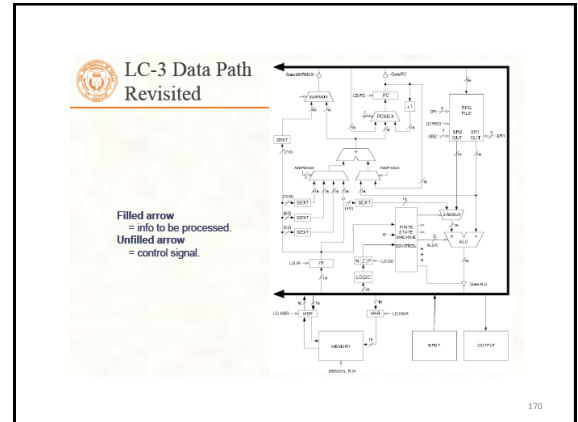
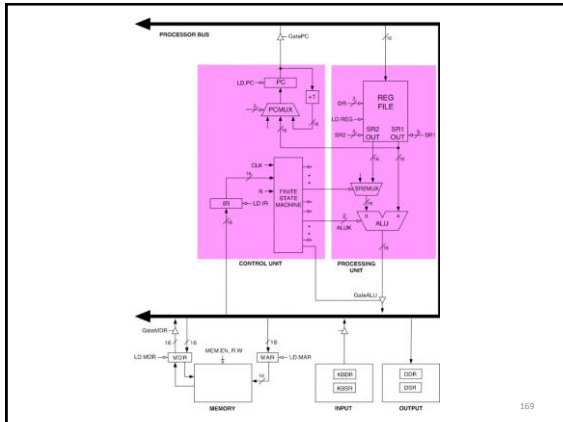


### 3.5 Récapitulatif

Afin d'être réellement utilisable, notre ordinateur doit être doté d'instructions supplémentaires, en particulier pour réaliser des sauts conditionnels, l'appel à des sous-programmes, etc.

La structure de base de l'ordinateur permet d'ajouter facilement les circuits nécessaires pour ces extensions.





## Architecture des Ordinateurs Périphériques, bus, et entrées- sorties

### Les périphériques

Dispositifs matériels permettant d'assurer les échanges d'informations en entrée et en sortie entre l'ordinateur et l'extérieur ou de stocker de manière permanente des informations

- ✓ Clavier
- ✓ Souris
- ✓ Imprimantes
- ✓ Écrans, ...

### Gestion des entrées-sorties

Trois méthodes de gestion des entrées/sorties

- ✓ La scrutation ou polling
- ✓ Les entrées-sorties pilotées par les interruptions
- ✓ L'utilisation d'un dispositif permettant des accès directs à la mémoire, DMA

Voir Chapitre « Méthodes de transfert d'informations »

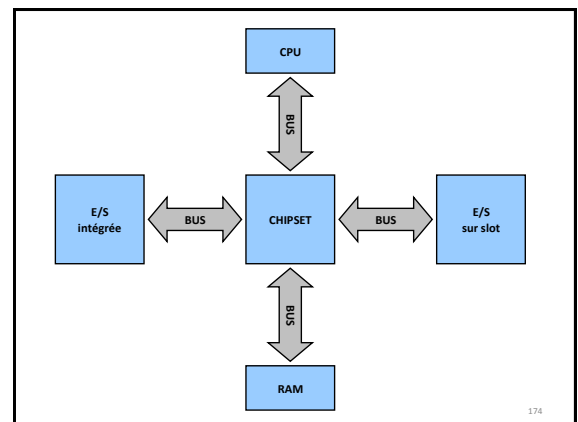
### Définition

Les bus sont des systèmes de câblage qui permettent la transmission d'information entre les composants d'un ordinateur (processeur, mémoire, carte graphique, ...).

Définition : chemin partagé entre plusieurs unités

- Utilisation de la logique à trois états
- Gagner beaucoup d'espace sur la puce
- Diminuer la distance parcourue par un signal.
- Un fil, ou une collection de fils pour un signal.

Un seul équipement transmet à un instant donné



### Le CHIPSET

L'élément chargé d'aiguiller les informations entre les différents bus de l'ordinateur afin de permettre à tous les éléments constitutifs de l'ordinateur de communiquer entre eux.

Dans les systèmes microprocesseurs de base:

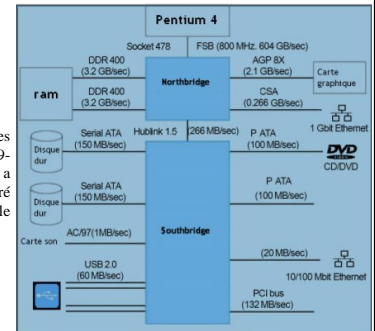
- le processeur est directement relié à la mémoire RAM et aux périphériques d'entrée sortie.
- Les vitesses externes des processeurs sont nettement plus rapides que ces périphériques.
- En plus, les vitesses de FSB (Front Side Bus) augmentent régulièrement: impossible de synchroniser la Ram et les périphériques qui ont des vitesses plus faible.

**Ces deux bus étaient reliés par un composant nommé le chipset, chargé de faire la liaison et de transmettre les données d'un bus à l'autre.**

175

### De nos jours

Les ordinateurs contiennent bien plus que deux bus, et presque chaque composant ou contrôleur de périphérique est connecté sur le *chipset* par un bus.



Sur les ordinateurs ayant des processeurs récents (2009-2010), le northbridge a disparu : celui-ci est intégré directement dans le processeur.

### Front side bus

Le bus qui relie le processeur au chipset. Plus celui-ci est rapide, moins le processeur a de risque d'être ralenti par la mémoire et les opérations de communications avec le chipset.

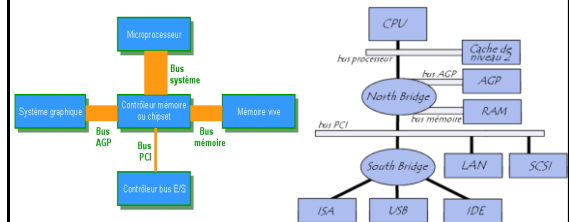
FSB, est cadencé à une fréquence qui est considéré comme l'horloge de base, de laquelle toutes les autres fréquences découlent. Ainsi, la fréquence du processeur est un multiple de la fréquence du FSB.

Sur certaines cartes mères, il est possible de modifier la fréquence du FSB et/ou les coefficients multiplicateurs : on peut ainsi faire varier la fréquence de notre processeur ou de notre mémoire.

Certains préfèrent l'augmenter pour avoir un processeur ou une mémoire plus rapide, et font ce qu'on appelle de l'overclocking (surfréquence en français).

Récemment, le FSB a subi quelques évolutions. Sur les cartes mères récentes, le northbridge est en effet intégré au processeur.

### Les différents types de bus



## 3. Les différents types de bus

Le **bus système** (ou bus processeur):

- canal de communication entre le microprocesseur et les autres composants.
- bus le plus rapide du système.

Le **bus mémoire**:

- il assure le transfert des données entre le microprocesseur et la mémoire vive (RAM).

179

### 3.1 Le bus d'adresses et le bus de commandes:

Il sont des sous-ensembles des bus système et bus mémoire.

- ✓ le bus d'adresses indique l'adresse mémoire ou l'adresse d'un autre bus utilisée lors d'un transfert de données.
- ✓ le bus de commandes véhicule tous les signaux utilisés pour synchroniser les différentes activités qui se déroulent dans les unités fonctionnelles de l'ordinateur.

### 3.2 Les bus E/S

Il assurent la transmission des données à traiter entre les différents éléments de l'ordinateur.

- ✓ les bus E/S de périphériques, à travers lequel les périphériques se parlent entre eux et avec l'ordinateur.
- ✓ les bus E/S d'expansion, permettent l'ajout de cartes d'extensions dans l'ordinateur pour obtenir de nouvelles fonctionnalités.

180

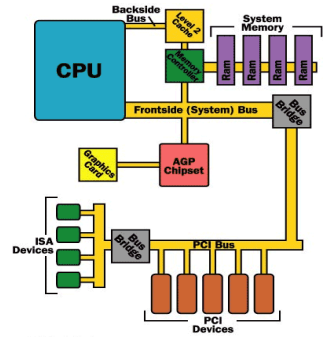
### 3.3 Caractéristique d'un bus

- **Largeur:** le bus comporte un nombre de lignes parallèles, chacune d'entre-elles laissant passer un seul bit à la fois. Un bus comportant 32 lignes est un bus 32 bits
- **Fréquence:** exprimée en mégahertz (MHz), correspond au nombre de fois que le bus est accédé par seconde.
- **Bande passante:** on détermine la bande passante maximale d'un bus à partir de la valeur des deux caractéristiques précédentes. Si un bus 16 bits évolue à une fréquence de 8 MHz (1 Hertz = 1 impulsions/s), on peut estimer sa bande passante théorique maximale à :  $8 \times 16 = 128 \text{ Mbits/s} = 16 \text{ Mo/s}$  (1 octet équivaut à 8 bits).

181

### 4. Etude détaillée des normes de bus E/S

ISA  
Micro Channel (MCA)  
EISA  
VESA Local bus (VL-bus)  
PCI Local bus  
AGP  
PC-Card (PCMCIA)  
USB  
Fire Wire (IEEE-1394)



©2001 Hewlett-Packard

182

### 4.1 Les bus E/S d'expansions

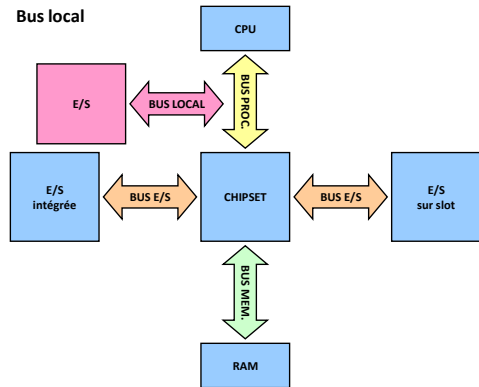
- Le bus ISA (Industry Standard Architecture)
- Le bus MCA (Micro Channel Architecture)
- Le bus EISA (Extended Industry Standard Architecture)

### 4.2 Le bus local

- Le bus VLB (VESA Local Bus)
- Le bus PCI (Peripheral Component Interconnect)
- Bus PCI-X: PCI eXtension
- Le bus AGP: (Accelerated Graphics Port)

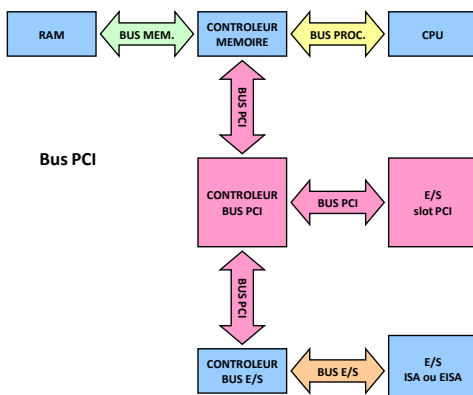
183

#### Bus local



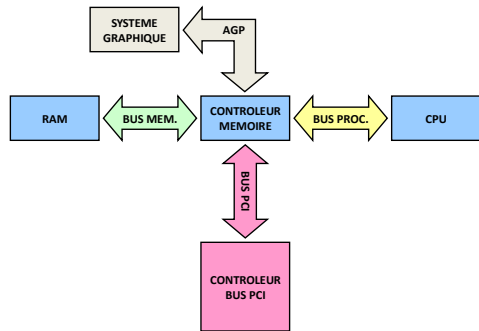
184

#### Bus PCI



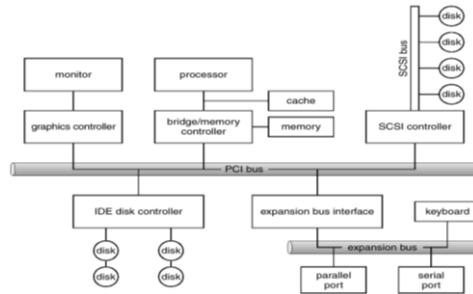
185

#### Le bus AGP



186

### Structure typique de bus PC



## Architecture des Ordinateurs Les bus E/S Périphériques

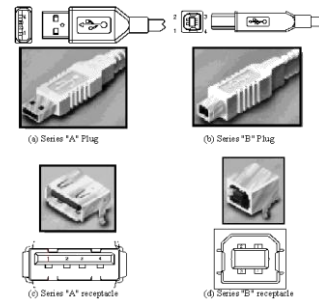
188

### 1. Les bus E/S Périphériques

- Le bus USB (Universal Serial Bus)
- Le bus SCSI (Small Computer System Interface)
- Le bus IEEE-1394 (FireWire)
- Le bus IEEE-1394b
- Le bus PC-Card (PCMCIA)

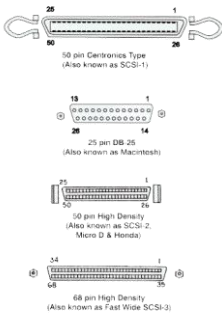
189

#### 1.1 Connecteurs USB



190

#### 1.2 Connecteurs SCSI

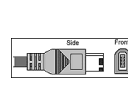


SCSI Table Legend		
Connector	Connector Type	SCSI Supported
	DB25 Pitch = 2.77mm	SCSI-1
	Centronics Pitch = 2.54mm	SCSI-1
	HPDB50 Pitch = 1.27mm	SCSI-2
	IDC50 Female Pitch = 2.54mm	SCSI-1 SCSI-2
	HPDB68 Pitch = 1.27mm	SCSI-3 Ultra SCSI Ultra2
	VHDCI Pitch = 0.8mm	Ultra2

191

#### 1.3 Connecteurs IEEE-1394

IEEE-1394 6 contacts



IEEE-1394 4 contacts



192



#### 1.4 Cartes au formats PC-Card

Types	Épaisseur (mm)	Usage commun
I	3,3	Mémoires (RAM ou Flash RAM)
II	5,0	Adaptateurs LAN, modems
III	10,5	Unités de stockage (disques durs)



193

#### 2. Gestion des Périphériques

##### Bus périphériques

- Inconvénients des bus parallèles
  - Câbles plus gros et plus chers
  - Connecteurs plus gros et plus chers
  - Dispersions de propagation des signaux
  - Limitation en fréquence
- Diminution de taille des équipements

194