

Cours

Electronique et architecture microprocesseur

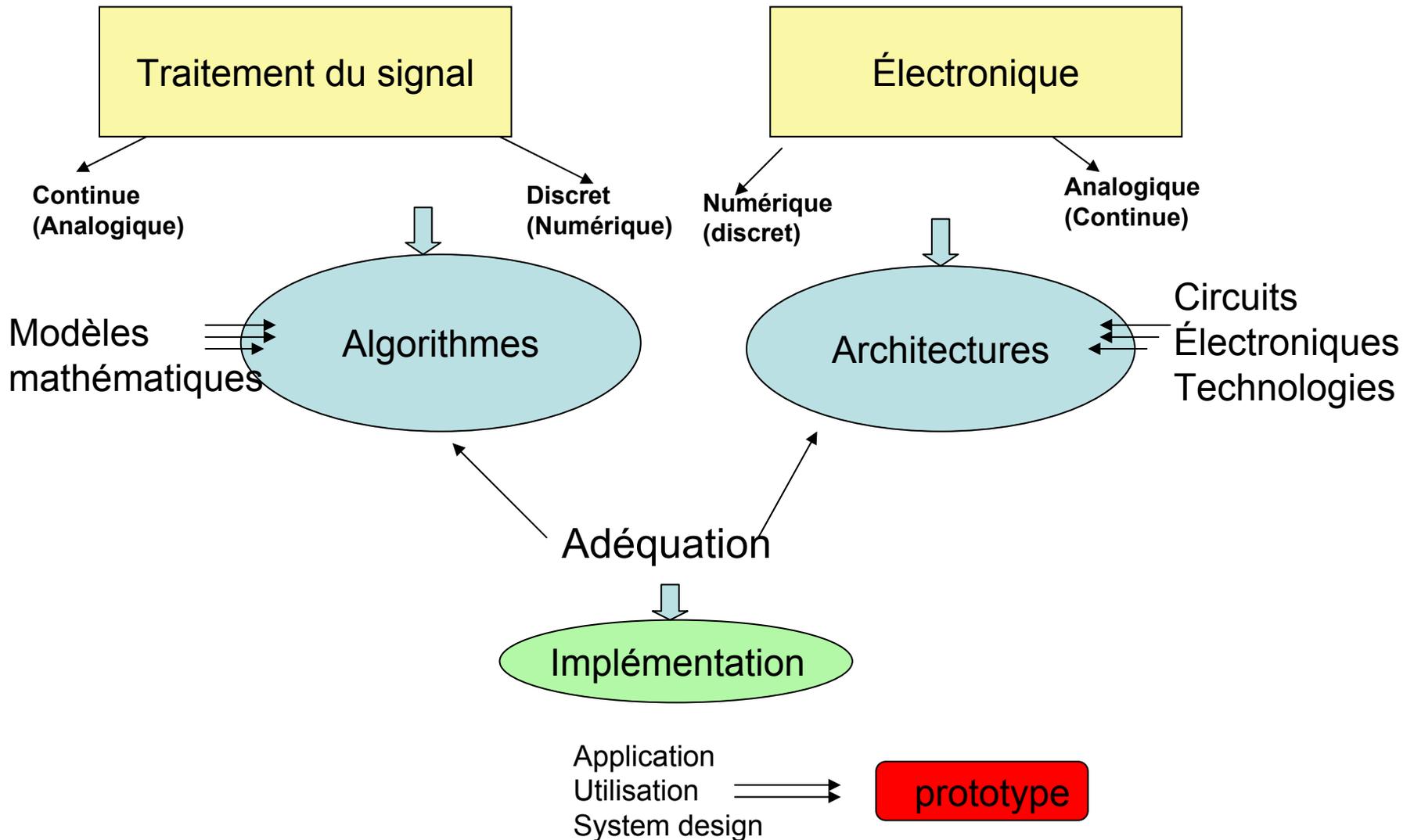
Prof. OUADOU M.

Département de Physique
Faculté des Sciences de Rabat

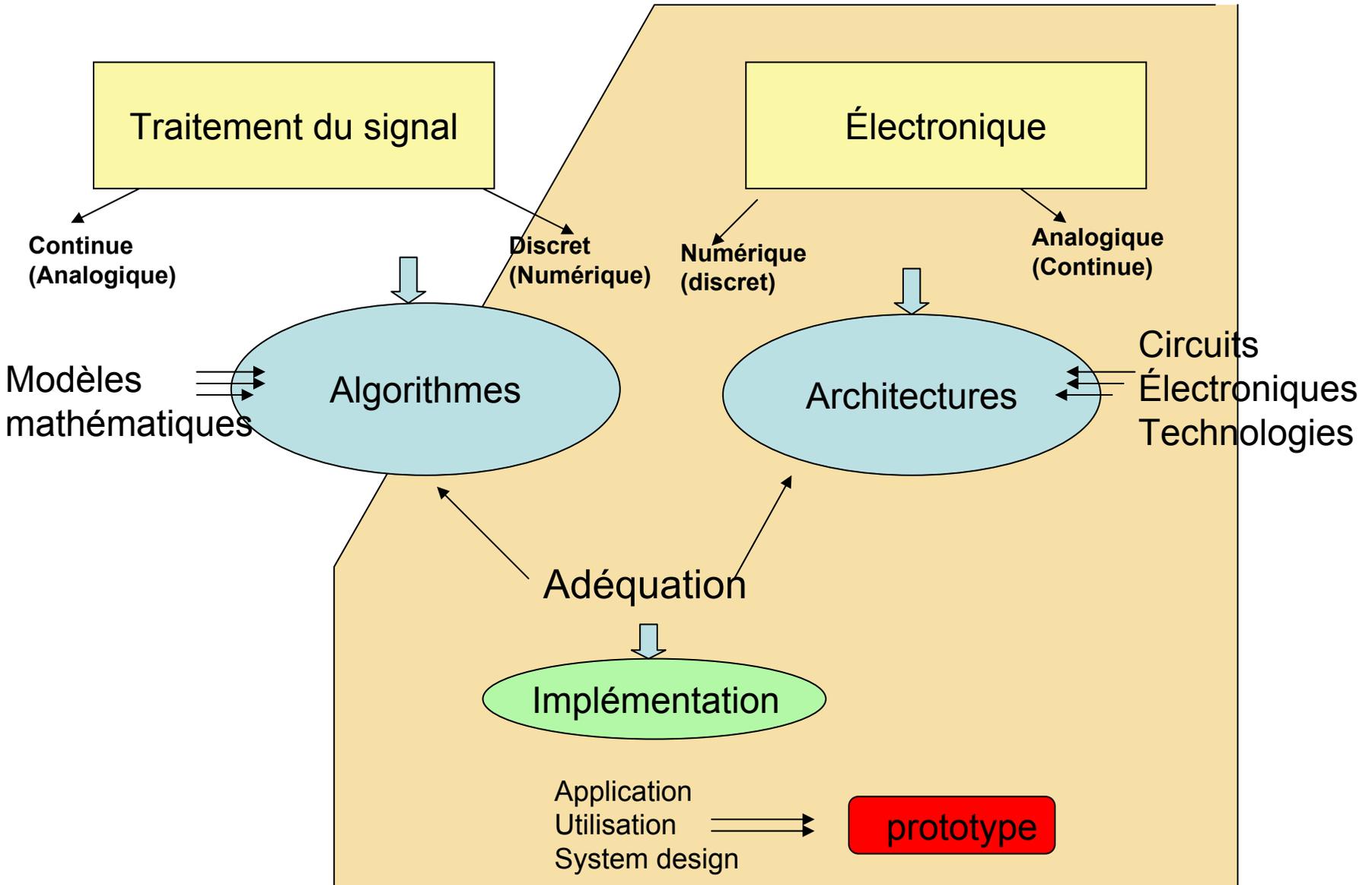
Sommaire

- **Rappels**
 - Électricité
 - Électronique Analogique
- **Electronique numérique**
- **Architecture des processeurs**
 - Les microprocesseurs
 - L'architecture du 6800 de Motorola
 - La programmation Assembleur
 - Les interfaces entrées sorties
 - Les outils de développements
 - La méthodologie d'implémentation
 - Les applications

LE DOMAINE



LE DOMAINE



Domaines d'application

- **Calcul général**
 - Ordinateurs et micro-ordinateurs
 - Calculatrices

- **Traitement du signal**
 - Télécommunications (réseaux et GSM)
 - Traitement des signaux sonores, de la parole et la musique
 - Signaux sismiques
 - Radars (poursuite de cible)
 - Navigation

 - TV , vidéo et photo numérique
 - Traitement et filtrage de l'image
 - Compression de l'image (pour le stockage et la transmission)
 - Détection d'objets contenus dans une image
 - Reconnaissance de forme et du visage et de l'individu
 - Reconnaissance des caractères

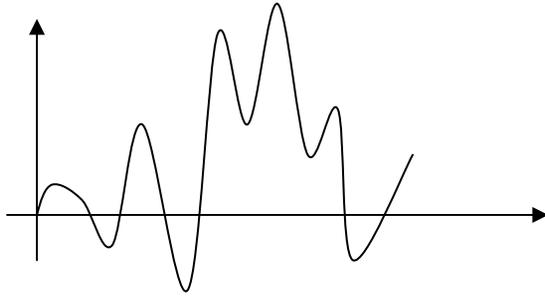
Domaines d'application

- **Automatique**
 - Identification
 - Contrôle adaptatif
 - Surveillance des processus industriels
- **Instrumentation**
 - Oscilloscopes
 - Microscope électronique
 - Appareils de mesure
- **Electronique domestique**
- **Systemes de surveillance**
- **Avions, navires, auto**
 - ... etc.

Rappels

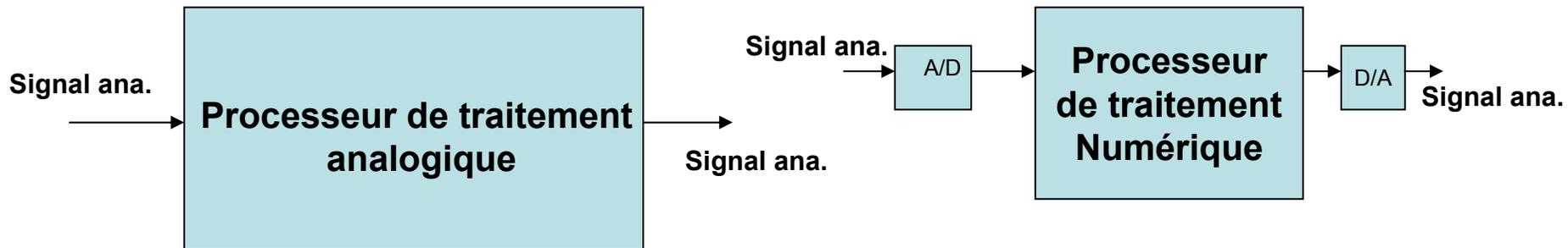
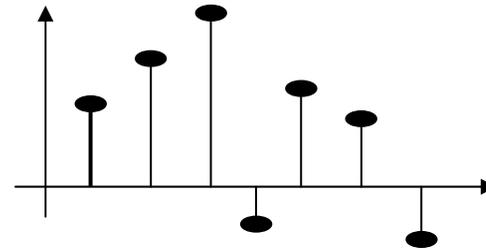
Signal analogique

↓
continue

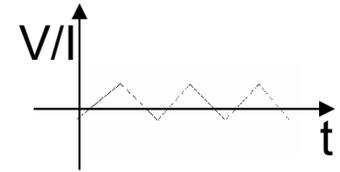
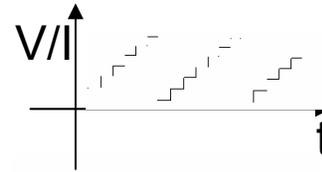
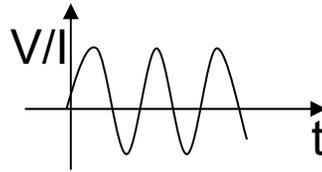
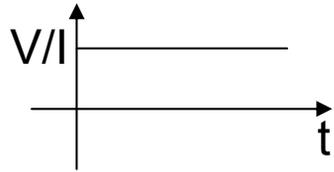


Signal numérique

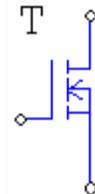
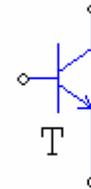
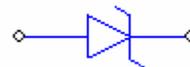
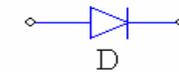
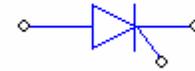
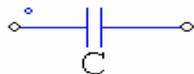
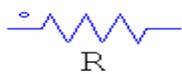
↓
discret



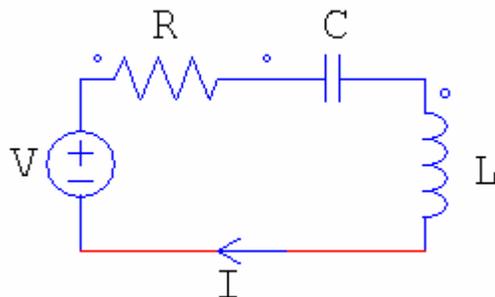
Courants et tensions :



Les éléments :



Les circuits :



$$V = R \cdot I + I \cdot Z_C + I \cdot Z_L$$

Rappels

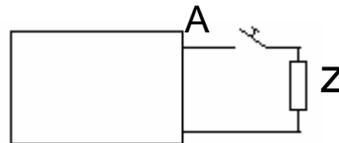
Électronique analogique

Loi d' Ohm : $U / I = R$

Lois de Kirchoff : $\sum I_k = 0$ $\sum R_k I_k = 0$

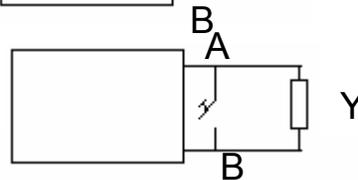
Théorèmes :

Thevenin



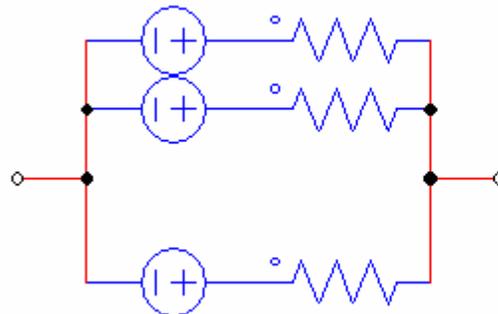
$$I = V_{ab} / (Z + Z_{ab})$$

Norton



$$V_y = I_{ab} / (Y + Y_{ab})$$

Millman



≈



$$V = (1 / \sum (1/R_i)) \sum V_i / R_i$$

Rappels

Électronique analogique

L'outil mathématique principal de calcul :

La transformation de Laplace

L'outil principal de représentation graphique:

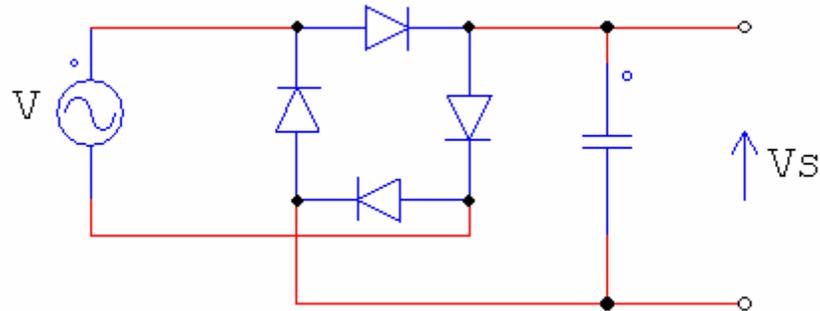
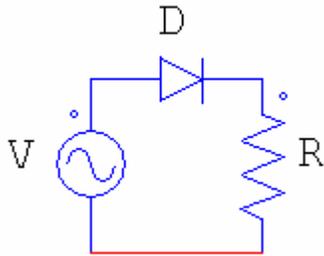
La représentation de Bode

Rappels

Électronique analogique

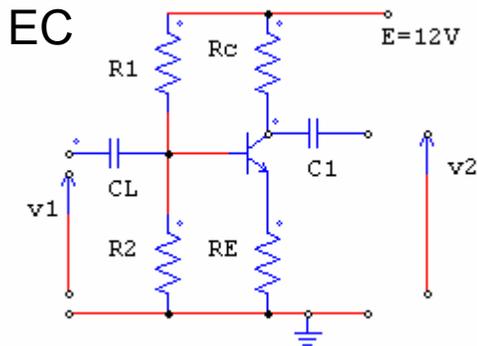
Circuits à base de diodes :

Redressement



Circuits à base de transistors :

Amplification

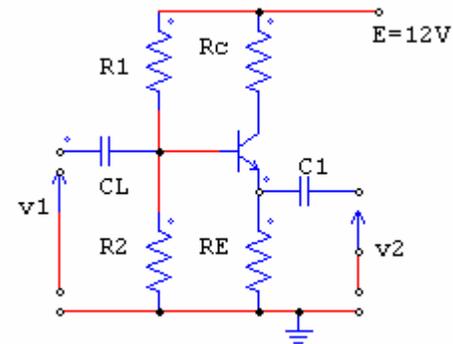


$$V_s = K_v * V_e$$

$$I_s = K_i * I_e$$

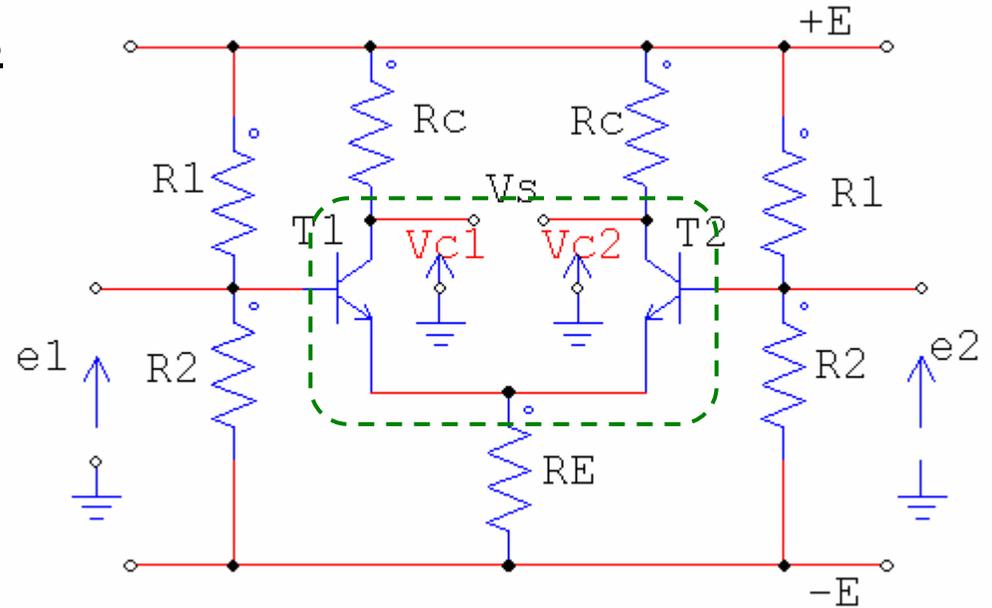
$$Z_e, Z_s$$

CC

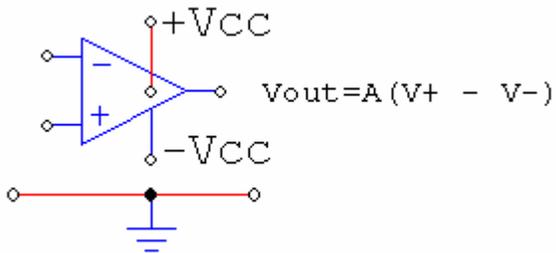


Rappels

Les amplificateurs différentielles



Les amplificateurs opérationnels



Gain

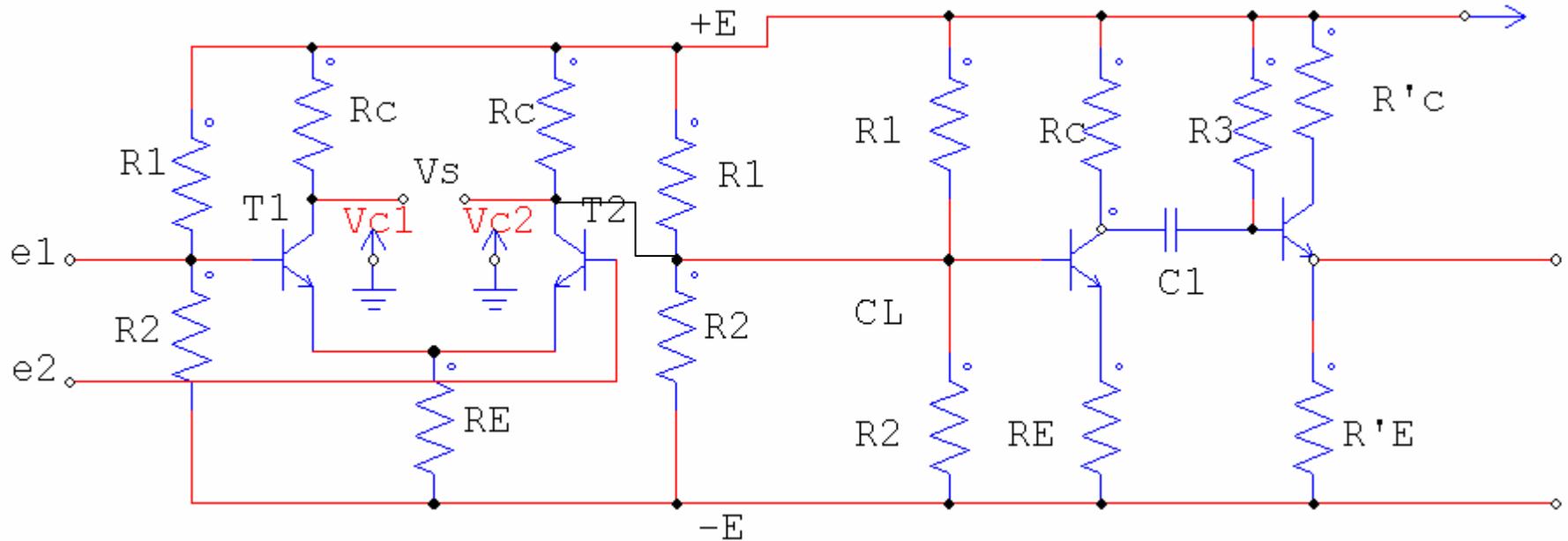
Impédance

Amplification AC et DC

Dérives

Rappels

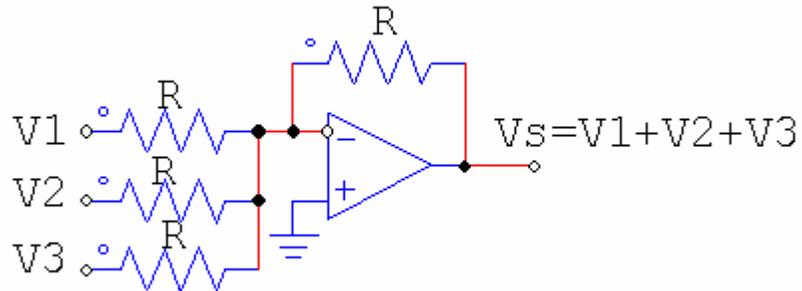
L'amplificateur opérationnel



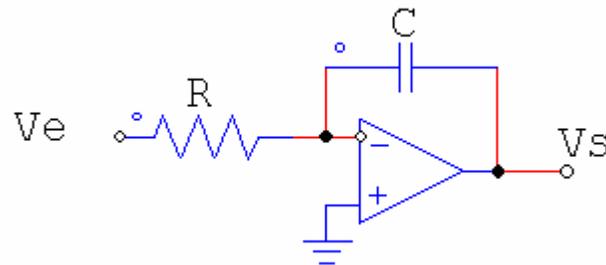
Rappels

Réaction négative

Sommateur

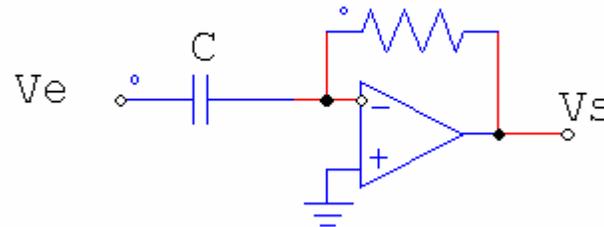


Intégrateur



$$V_S = \int V_e dt$$

Différentiateur



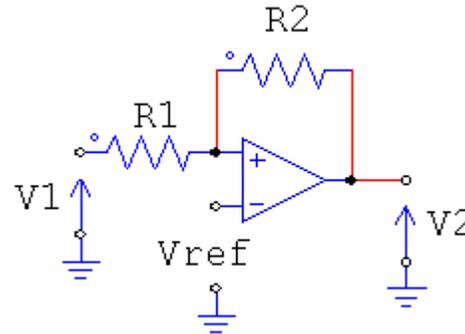
$$V_S = dV_e/dt$$

Filtres

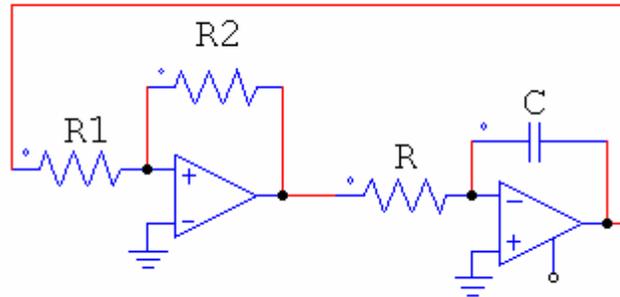
Rappels

Réaction positive

Trigger de Schmitt



Générateurs de signaux



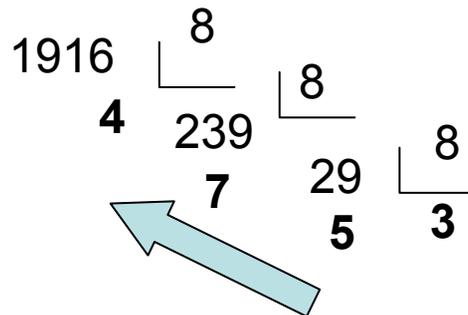
Filtres

La numération et représentation des nombres

La numération est la science qui traite de la dénomination et de la représentation graphique des nombres.

Le problème qui se pose est de représenter tous les entiers naturels et les décimaux à l'aide d'un ensemble fini de symboles (souvent des chiffres) rassemblés selon des règles (le code).

$$(2563)_{10} = 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 3 \times 10^0$$



$$(1916)_{10} = (3574)_8$$

Codage binaire des nombres entiers

$$A = (N)_b = a_n \dots a_i \dots a_1 a_0 = a_n b^n + \dots a_i b^i + \dots a_1 b^1 + a_0 b^0$$
$$= \sum a_i \cdot b^i$$

Où b est la base

La base 2: **b=2**
 $a_i \in \{ 0 , 1 \}$
 $A \in \{ 0 , 2^n - 1 \}$

Ex. passage de la base 10 à la base 2

Codage binaire des nombres décimaux

Codage binaire des nombres entiers relatifs

Définitions :

Chiffre binaire ou **bit** (Binary digit) : la plus petite unité d'information Binaire de valeur 0 ou 1

Octet (**byte**) : nombre binaire de 8 bits

Mot (**word**) : élément d'information mémorisé ou traité d'un seul bloc.
(16 , 32 , 64 ... etc.)

Le traitement de l'information dans les calculateurs (processeurs) s'effectue Sur des mots de 8 bits 16, 32, ... etc.

Il faut aussi représenter les nombres positives et les nombres négatives.

Rappels

Électronique numérique

Pour 8 bits :

C'est le bit le plus fort qui représente le signe du mot traité.

$$A = - a_{n-1} \times 2^{n-1} + \sum_{i=0}^{N-2} a_i \times 2^i ;$$

$$a_i \in \{ 0 , 1 \}$$

$$A \in [-2^{n-1} , + 2^{n-1} - 1]$$

$$A = - a_{n-1} \times 2^{n-1} + \sum_{i=0}^6 a_i \times 2^i$$

a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-	+	+	+	+	+	+	+
-128	+64	+32	+16	+8	+4	+2	+1

Les valeurs max et min ?

Rappels

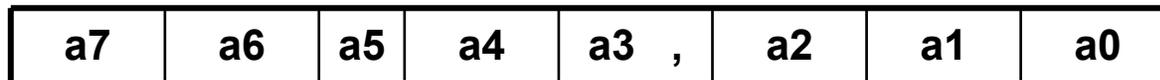
Électronique numérique

Deux représentations des nombres :

* Virgule fixe

Avec 8 bits, 28 nombres = 256 combinaisons différentes

La précision = $1 / 256$



Avec plusieurs bits on a plus de précision

++ rapidité de calcul

-- dynamique limitée (...)

Rappels

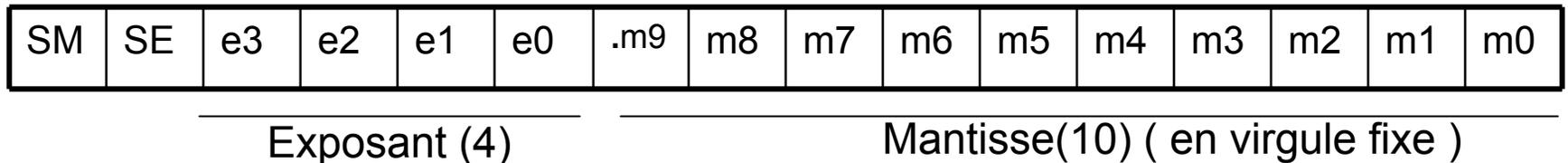
Électronique numérique

* Virgule flottante

Tout nombre est représenté sous forme :

$$N = M \cdot b^E$$

Ex: b_{10}, b_2



++ dynamique large

++ plus de précision

-- plus de temps de calcul

Ex: représenter $(+2.5)_{10}$ sur 2 octets (16 bits)

Le code décimal code binaire (DCB ou BCD)

Théorèmes fondamentaux de l'algèbre de BOOLE

BOOLE 1847

Algèbre qui s'applique à des fonctions logiques de variables logiques.

(variables Booléennes)

Toute fonction logique peut être réalisée à l'aide d'un petit nombre de fonctions logiques de base appelées **opérateurs logiques** ou **portes (gates)**.

Principaux sont: **NON, ET, OU + XOR, NAND, NOR**

Les théorèmes:

Théorèmes des constantes

$$a + 0 = a$$

$$a \times 0 = 0$$

$$a + 1 = 1$$

$$a \times 1 = a$$

Idempotence

$$a + a = a$$

$$a \times a = a$$

Complémentation

$$a + \overline{a} = 1$$

$$a \times \overline{a} = 0$$

Commutativité

$$a + b = b + a$$

$$a \times b = b \times a$$

Distributivité

$$a + (b \times c) = (a + b)(a + c)$$

$$a \times (b + c) = (a \times b) + (a \times c)$$

Associativité

$$a + (b + c) = (a + b) + c = a + b + c$$

$$a \times (b \times c) = (a \times b) \times c = a \times b \times c$$

Autres relations

$$a = \overline{\overline{a}}$$

$$a + (a \times b) = a$$

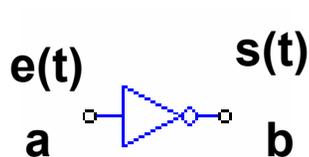
$$a + (\overline{a} \times b) = a + b$$

$$a \times (a + b) = a$$

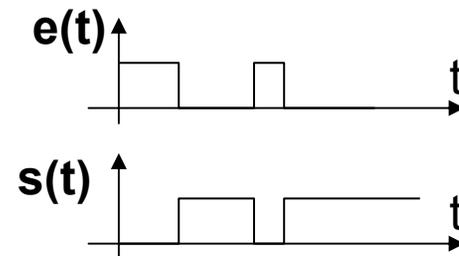
$$(a + b)(a + \overline{b}) = a$$

Les circuits logiques élémentaires (représentation des fonctions binaires)

La complémentation (inversion ou négation ou pas) : NON (NO)

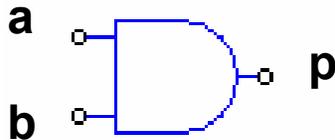


a	b
0	1
1	0

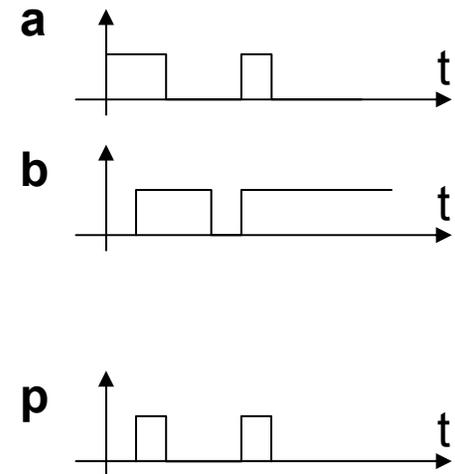


Les circuits logiques élémentaires

La produit (intersection ou multiplication logique) : ET (AND)



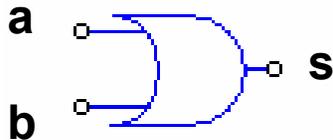
a	b	p
0	0	0
0	1	0
1	0	0
1	1	1



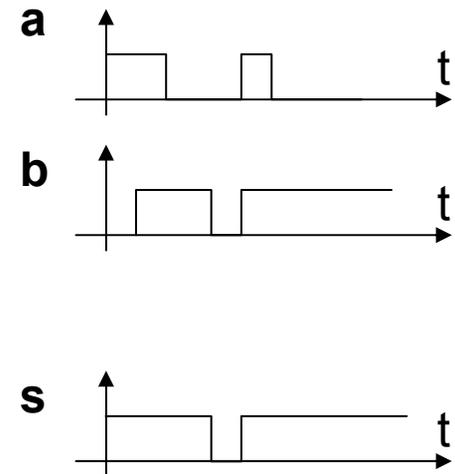
Les circuits logiques élémentaires

La produel (réunion ou addition logique) :

OU (OR)

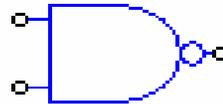


a	b	s
0	0	0
0	1	1
1	0	1
1	1	1



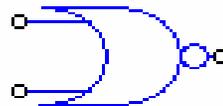
Les combinaisons :

Non Et



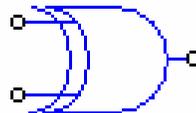
NAND

Non Ou



NOR

Ou exclusif



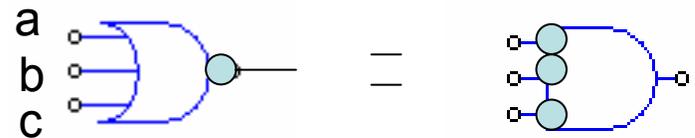
XOR

$$a \oplus b = a \bar{b} + \bar{a} b$$

Les théorèmes de Morgan:

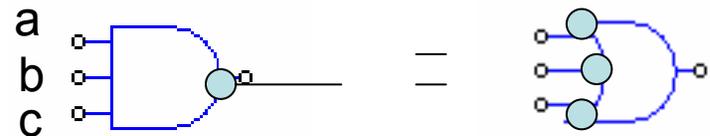
Le complément d'un produit est égal au produit des compléments des facteurs qui le composent.

$$\overline{a + b + c + \dots + q} = \bar{a} \bar{b} \bar{c} \bar{d} \dots \bar{q}$$



Le complément d'un produit est égal au produit des compléments des facteurs qui le composent.

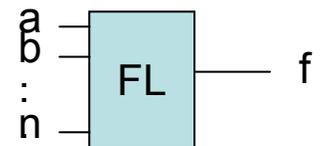
$$\overline{a b c d \dots q} = \bar{a} + \bar{b} + \bar{c} + \dots + \bar{q}$$



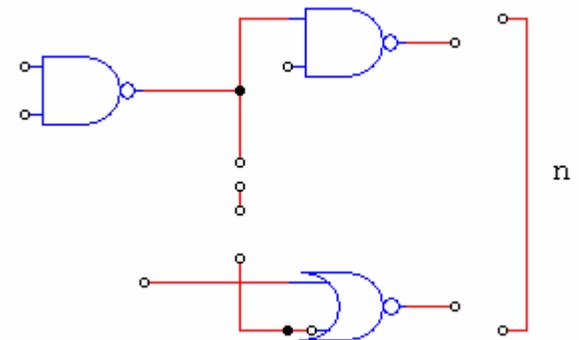
Les paramètres de base d'un élément logique

1 – La fonction logique réalisée

2 – Le coefficient de liaison en entrée : (nombre maximal d'entrées logiques de l'élément) (dépend de la technologie)
pour TTL $n = 8$



3 – Le coefficient de liaison en sortie : nombre maximal de branchement de la sortie
pour TTL ($n = 4 - 10$)
pour circuits spéciales $n = 30$



Les paramètres de base d'un élément logique

4 – La rapidité de fonctionnement

temps de réponse moyen:

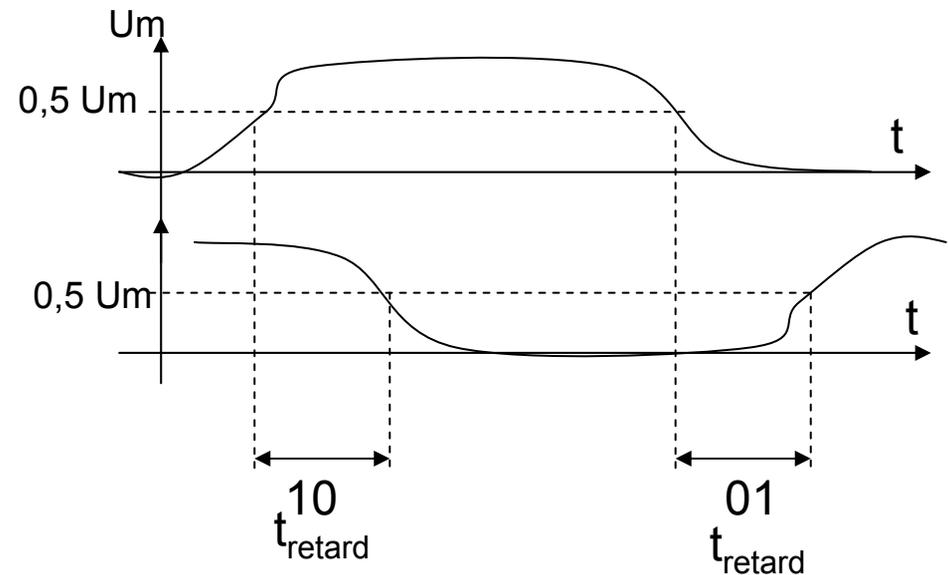
$$t_{\text{rep. moy.}} = (t_{\text{ret}}^{10} + t_{\text{ret}}^{01}) / 2$$

* Le temps de transition

$$t = \max \{ t_{\text{ret}}^{10}, t_{\text{ret}}^{01} \}$$

* La fréquence maximale

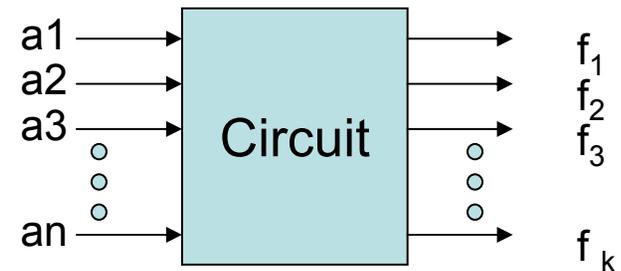
$$f_{\text{max}} = 1 / 2t$$



Les circuits numériques

Les circuits logiques 2 états d'équilibre = 2 niveaux de tension ou courant.

Les circuits logiques sont des dispositifs qui exécutent des opérations sur des variables logiques, transportent et traitent des signaux logiques.



- **Les circuits combinatoires** : circuits idéalisés où le temps de propagation des signaux n'est pas pris en considération. Les signaux de sortie ne dépendent que des signaux d'entrée, appliqués à l'instant considéré.

- **Les circuits séquentiels** : circuits où il faut tenir compte du temps de la propagation des signaux et de la mémoire du circuit .

Les signaux de sortie dépendent même des signaux d'entrée appliqués ultérieurement.

Les circuits numériques

La fonction logique d'un circuit peut se définir soit par :

1- tableau de correspondance = table de vérité

2- diagrammes de temps (temporaire)

3- expressions algébriques

4- schéma

Les circuits numériques

Les circuits combinatoires

La synthèse d'un circuit combinatoire:

La synthèse d'un circuit destiné à réaliser une fonction binaire donnée comprend trois étapes:

- Construire la table de vérité de la fonction logique.
- Écriture de l'expression de la fonction binaire.
- Simplification de l'expression en vue d'obtenir un circuit économique ou un circuit à temps de traversée minimal.
- Passage de l'écriture symbolique de l'expression simplifiée au schéma électronique du circuit.

Les circuits numériques

Les formes canoniques des fonctions binaires:

Toute fonction binaire peut s'exprimer:

- Soit par un produit de produits.
- Soit par un produit de produit.

en faisant intervenir toutes les variables directs ou complémentées

D'une façon general:

$$F = P_0 + P_1 + P_2 + \dots + P_N$$

$$F = S_0 S_1 S_2 \dots S_N$$

← Forme canonique
disjonctive

← Forme canonique
conjunctive

C'est deux expressions sont les deux formes canoniques de la fonction.

Avec: P= minterme S= maxterme

Simplification et minimisation des fonctions booléennes

Les formes canoniques des fonctions représentent des façons relativement compliquées d'écriture.

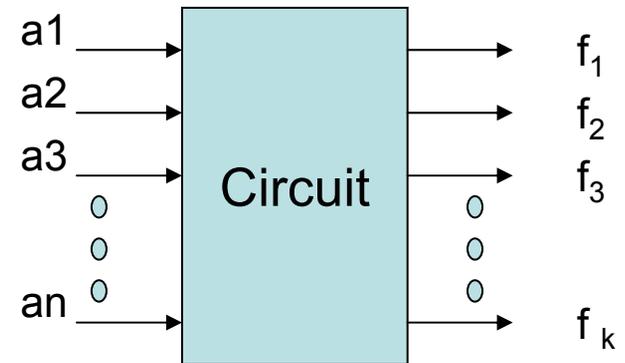
Simplifier une fonction booléennes consiste à mettre en œuvre des méthodes qui permettent d'écrire la fonction ou de réaliser le circuit correspondant sous sa forme la plus simple, tout en conservant les caractéristiques de la fonction.

Simplification par développement

Simplification par tables de Karnaugh

Exemples des circuits combinatoires

Fonction de transcodage



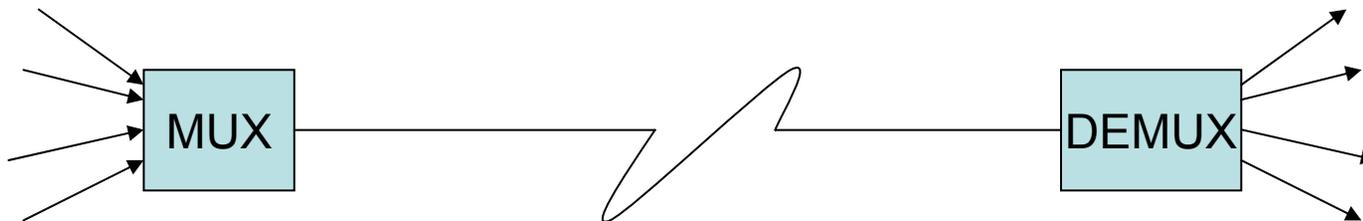
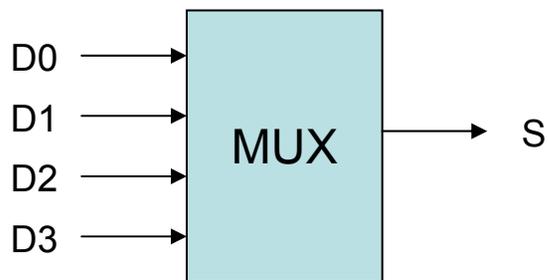
Le circuit est un circuit combinatoire si à chacun des combinaisons des variables a_1, a_2, \dots, a_n correspond une combinaison et une seule des fonctions f_1, f_2, \dots, f_n .

$k=1$, circuit logique

$k>1$, Additionneurs, multiplieurs, décodeurs, multiplexeurs ...

Exemples des circuits combinatoires

Multiplexeurs et démultiplexeurs

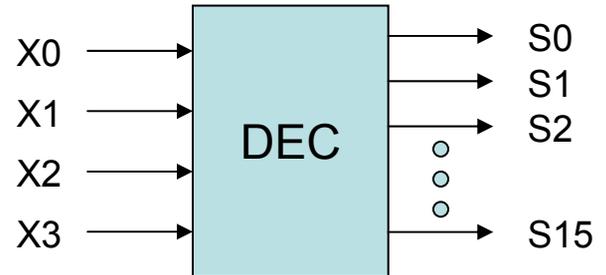


Conversion série parallèle

Exemples des circuits combinatoires

Décodeurs codeurs et transcodeurs

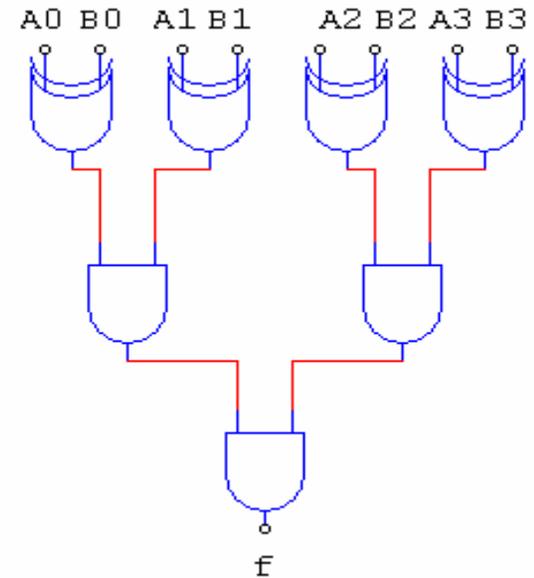
Le décodeurs fait correspondre à un code en entrée (sur n lignes) une seule Sortie active parmi les 2^n sorties possibles



Exemples des circuits combinatoires

Les comparateurs

Détection de l'égalité de deux nombres



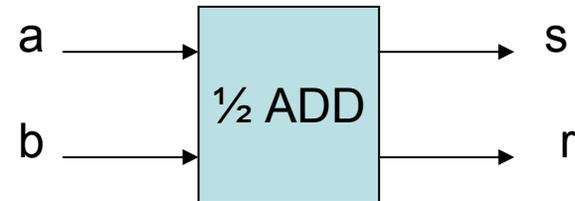
ALU

Décodeurs d'adresses

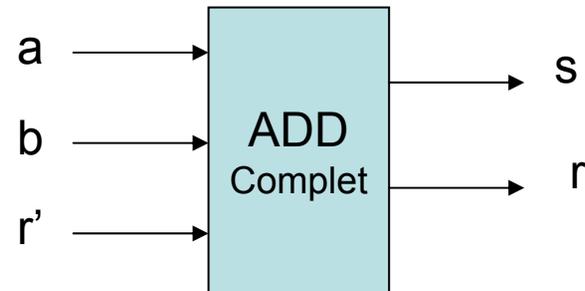
Exemples des circuits combinatoires

L'additionneur binaire

Le demi additionneur

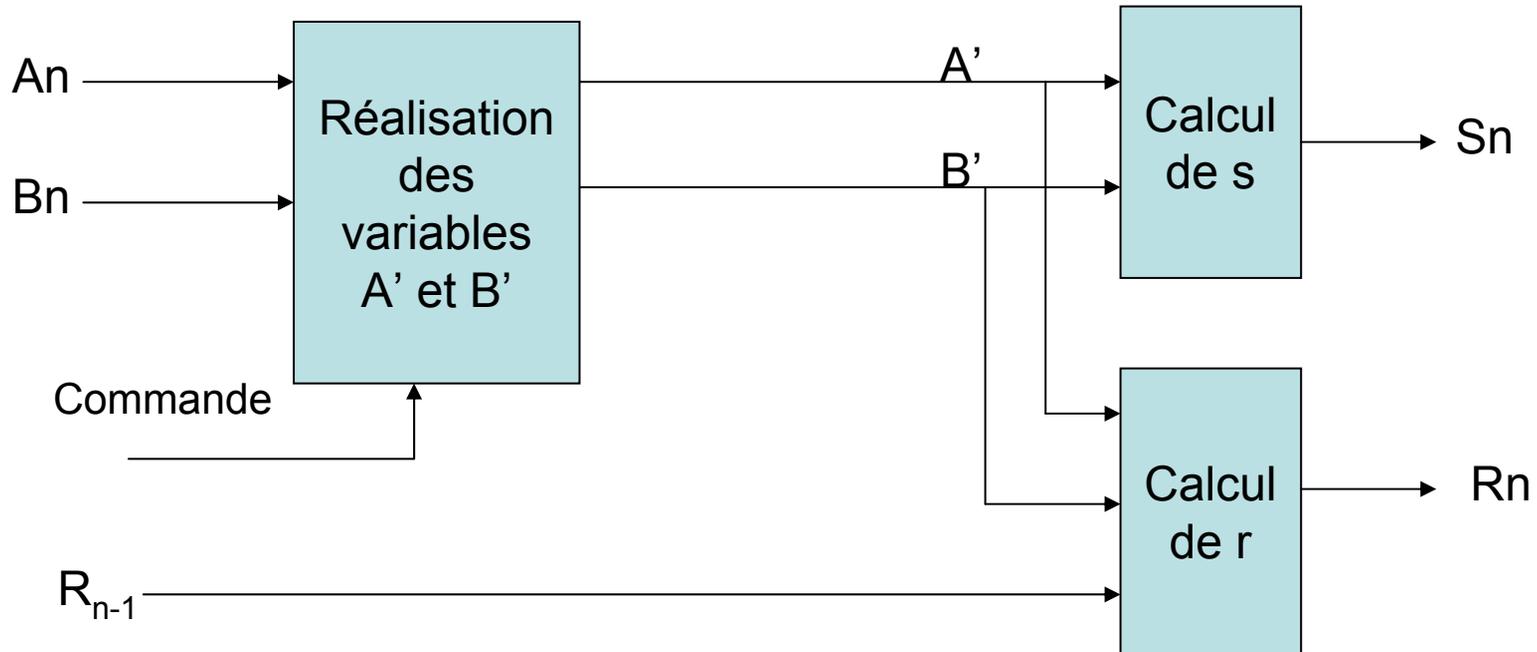


L'additionneur complet



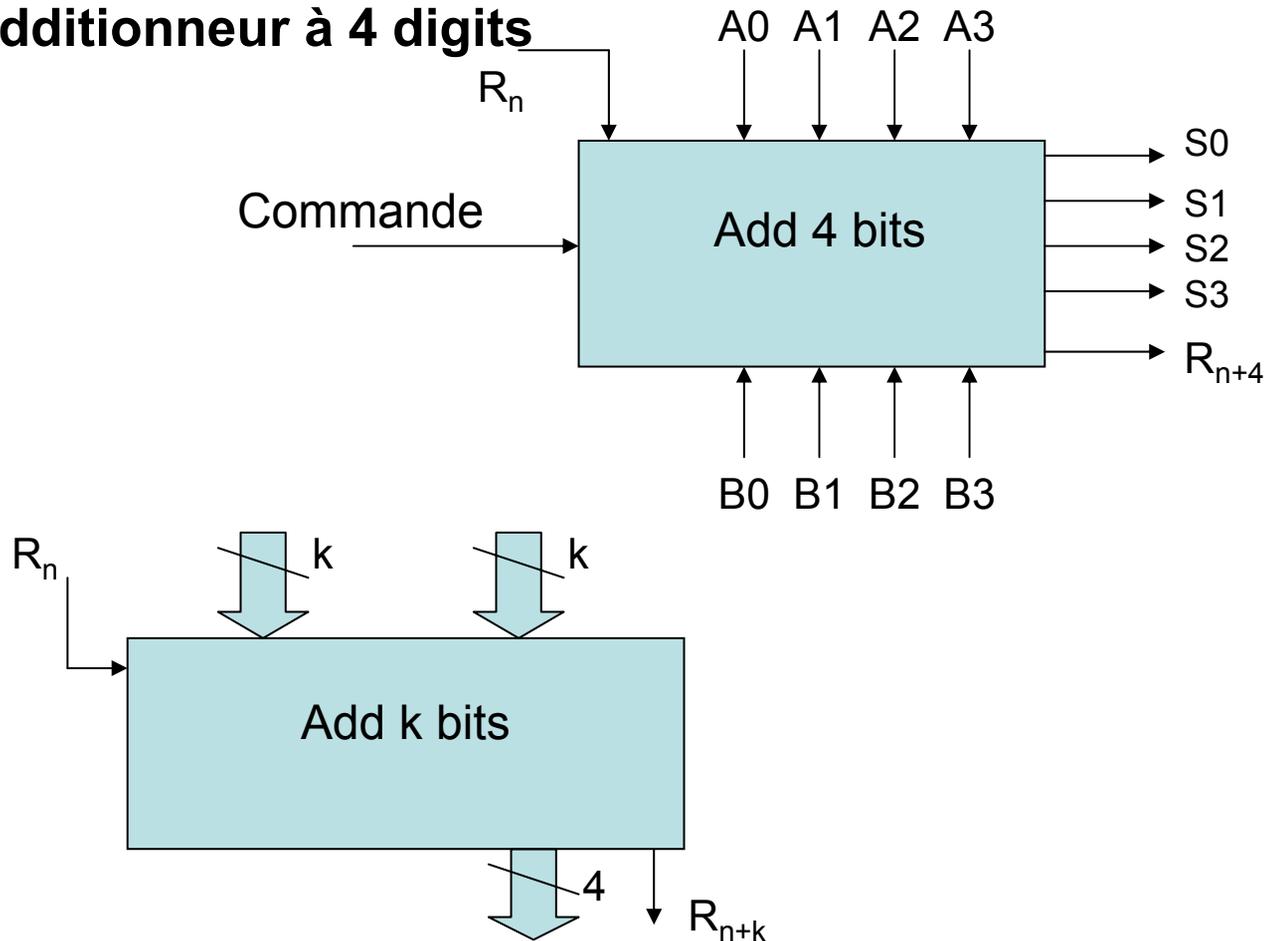
Exemples des circuits combinatoires

L'addition - soustraction



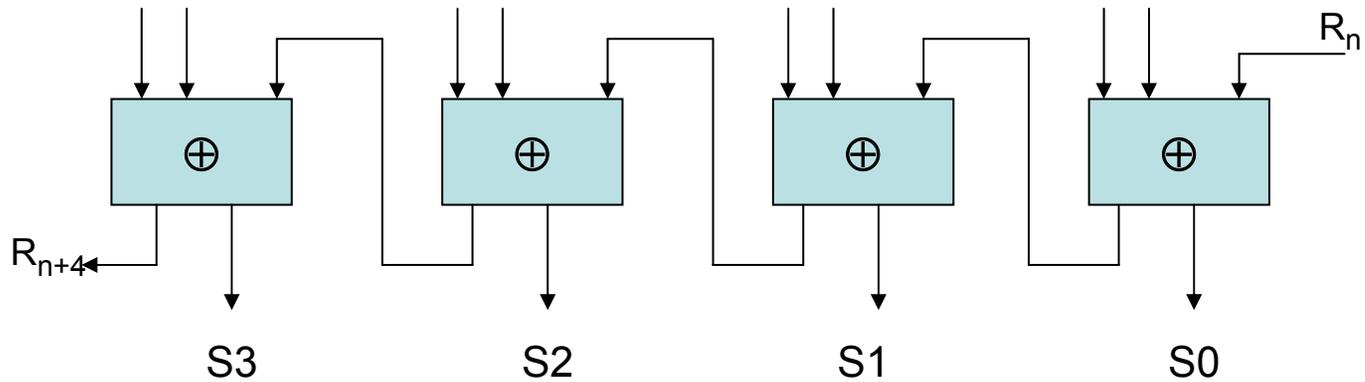
Exemples des circuits combinatoires

L'additionneur à 4 digits



L'additionneur

Propagation de la retenue



$$t = 4 \times t_{\text{add}}$$

SN74LS83

L'additionneur

La retenue anticipé

Les cas :

1 – pas de retenue $R_i = 0$

2 – La retenue propagée à travers l'étage

$$R_i = R_{i-1}$$

$$\text{Termes } P_i = a_i + b_i = 1$$

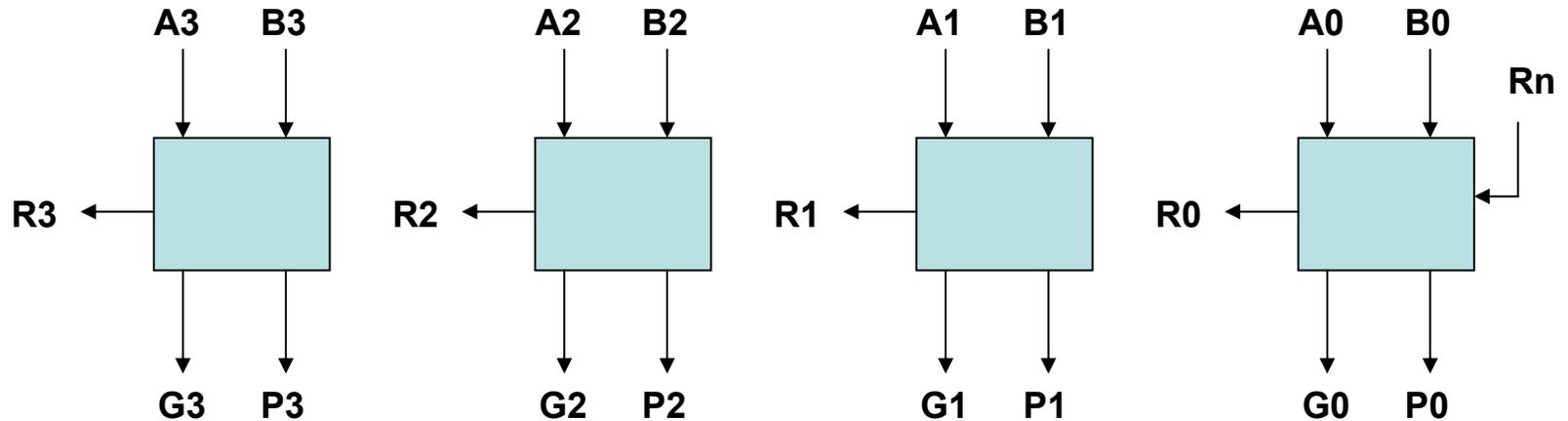
3 – La retenue y est indépendamment de R_i $R_i = 1$

$$\text{Termes de génération } G_i = a_i b_i = 1$$

A_i	B_i	R_{i-1}	S_i	R_i	N° de cas
0	0	0	0	0	} 1
0	0	1	1	0	
0	1	0	1	0	} 2
0	1	1	0	1	
1	0	0	1	0	} 3
1	0	1	0	1	
1	1	0	0	1	} 3
1	1	1	1	1	

L'additionneur

La retenue anticipé



Les expressions :

$$R_0 = G_0 + R_n P_0 \quad R_1 = G_1 + R_0 P_1 \quad R_2 = G_2 + R_1 P_2 \quad R_3 = G_3 + R_2 P_3 = R_{n+4}$$

$$R_{n+4} = R_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 R_n$$

3 couches de portes

$$t_{\text{add}} = 3 \times t_{\text{porte}}$$

SN74LS83A
SN74AS181A
SN74AS182 **transparents**

Les circuits numériques

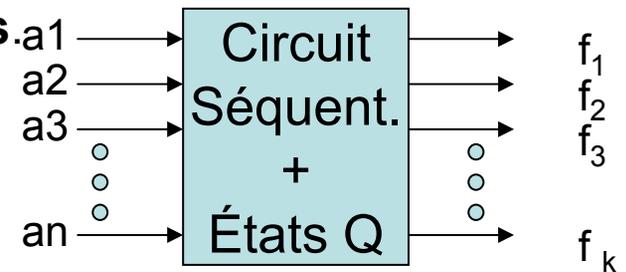
Les circuits séquentiels

Les circuits combinatoires n'ont pas de rétroactions et sont des circuits idéaux sans délai.

Les sorties ne dépendent que des entrées au même instant et l'étude repose sur **l'algèbre de Boole**.

Les circuits séquentiels possèdent des **rétroactions**.

Le circuit se rappelle des **Entrées**
et des **États** précédents



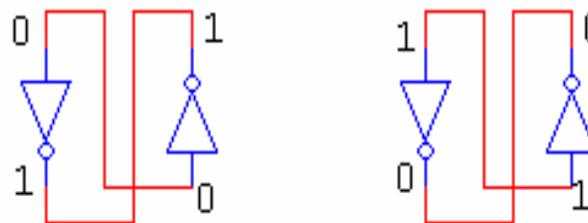
Il y a une mémoire du passé

L'étude des circuits séquentiels repose sur **la théorie des automates finis**.

Les bistables (les basculeurs ou flip-flops)

Deux inverseurs en opposition.

Deux états stables



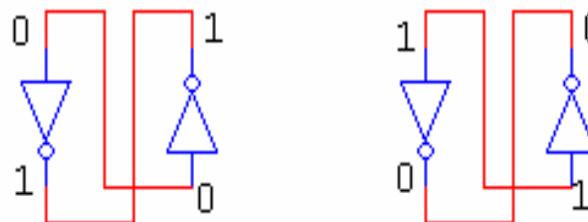
Plusieurs types :

R-S , D , T , J-K

Les bistables (les basculeurs ou flip-flops)

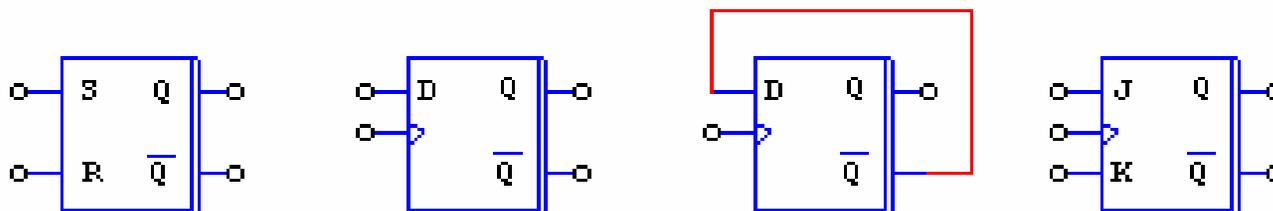
Deux inverseurs en opposition.

Deux états stables



Plusieurs types : sans et avec horloge de synchronisation

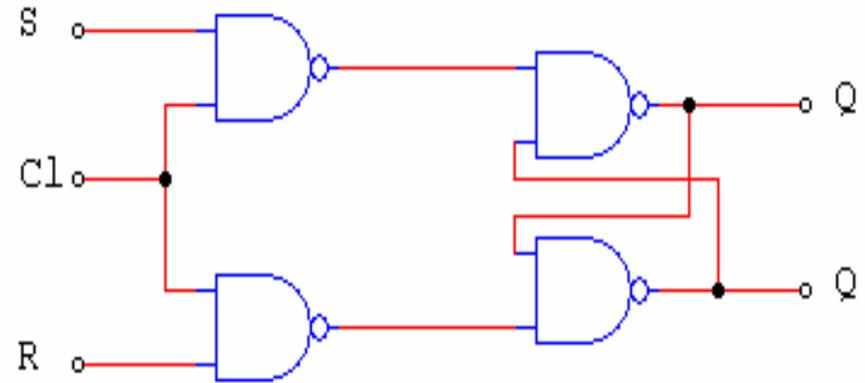
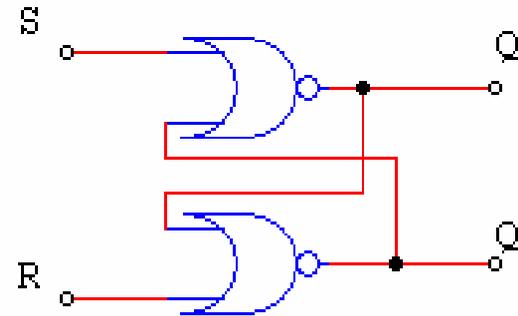
R-S , D , T , J-K



Les bistables :

Le bistable R-S.

R	S	Q	Q ⁺	Action
0	0	0	0	Q ⁺ =Q
0	0	1	1	Q ⁺ =Q
0	1	0	1	Mise a 1
0	1	1	1	Mise a 1
1	0	0	0	Effacement
1	0	1	0	Effacement
1	1	0	?	Indetermine
1	1	1	?	Indetermine

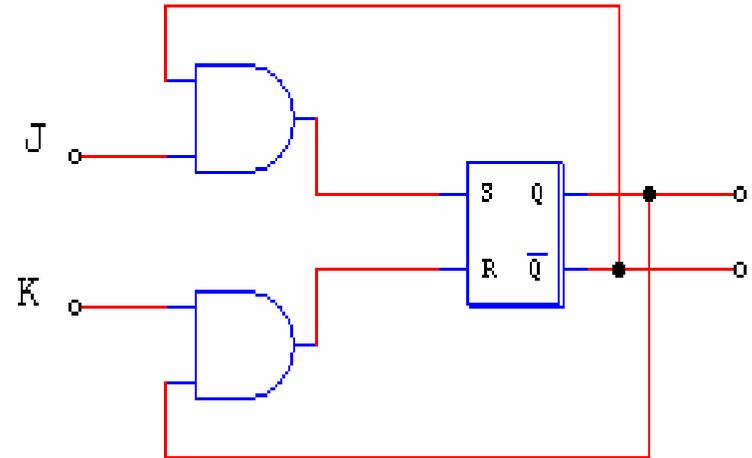


R-S synchronise

Les bistables :

Le bistable J-K.

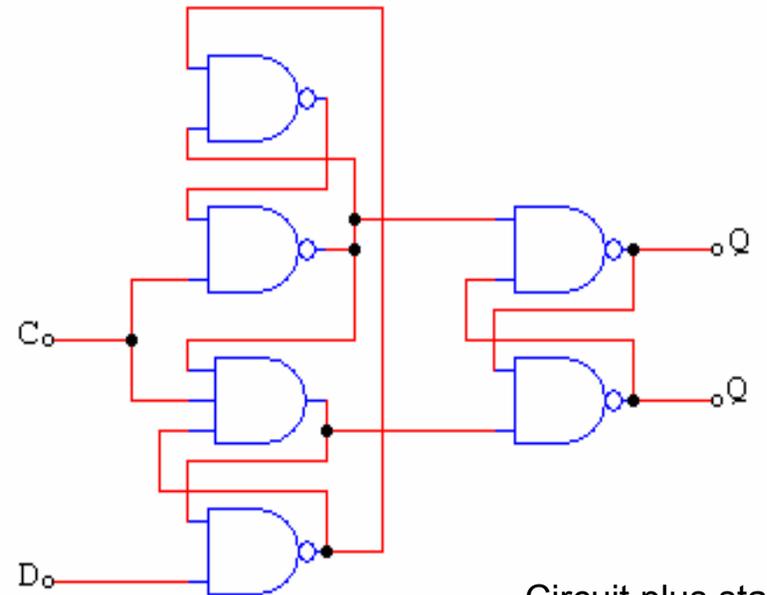
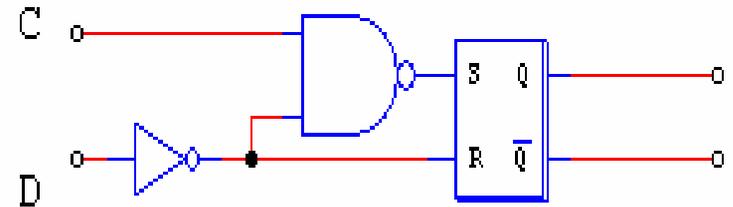
J	K	Q	Q ⁺	Action
0	0	0	0	Q ⁺ =Q
0	0	1	1	Q ⁺ =Q
0	1	0	1	Mise à 1
0	1	1	1	Mise à 1
1	0	0	0	Effacement
1	0	1	0	Effacement
1	1	0	1	Complément
1	1	1	0	Complément



Les bistables :

Le bistable D.

D	C	Q	Q ⁺	Action
0	0	0	0	Q ⁺ =Q
0	0	1	1	Q ⁺ =Q
0	1	0	0	Mise à 0
0	1	1	0	Mise à 0
1	0	0	0	Q ⁺ =Q
1	0	1	1	Q ⁺ =Q
1	1	0	1	Mise à 1
1	1	1	1	Mise à 1

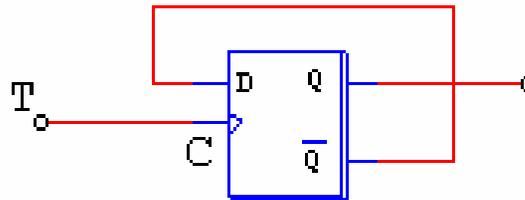
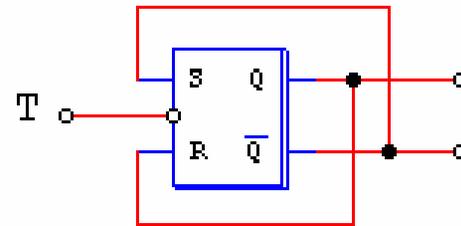


Circuit plus stable

Les bistables :

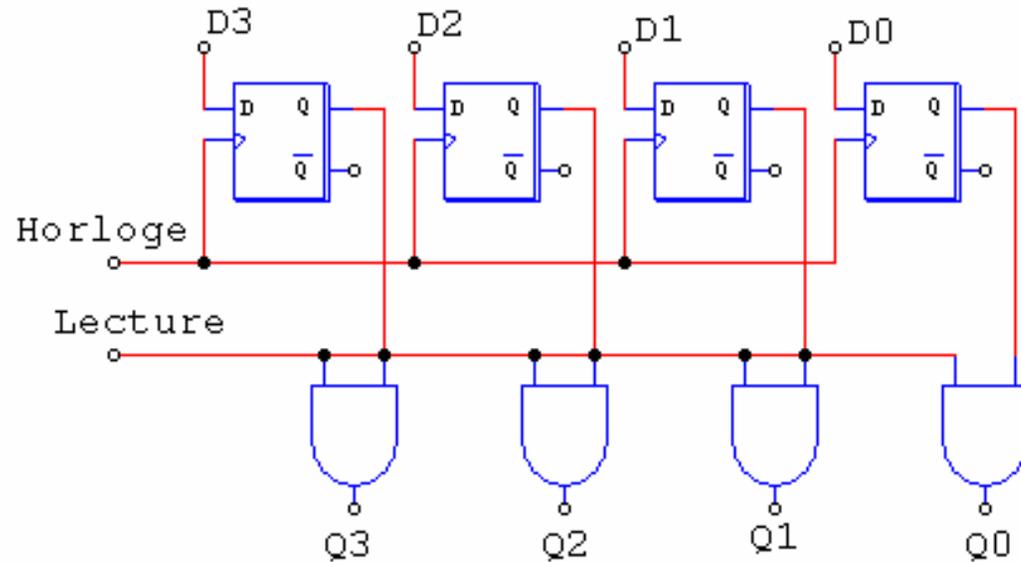
Le bistable T.

T	Q	Q ⁺	Action
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



Les applications

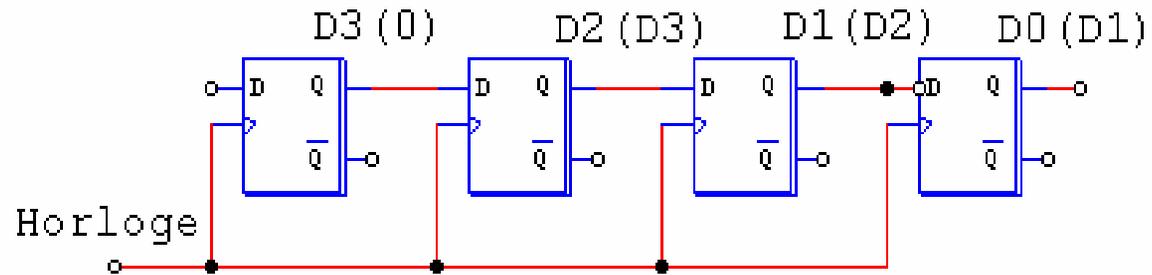
Le registre



Registre à 4 bits

Les applications

Le registre à décalage

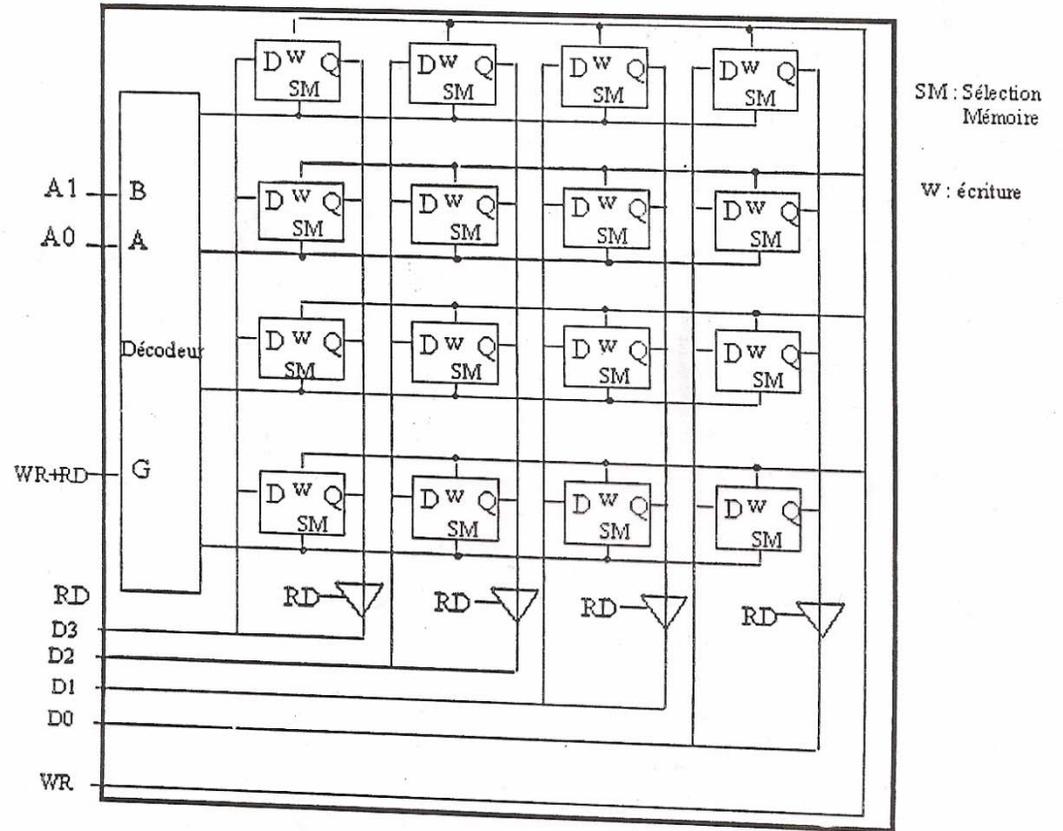


Diviseur sur 2

Convertisseur série/parallèle

Les applications

Mémoires rapides

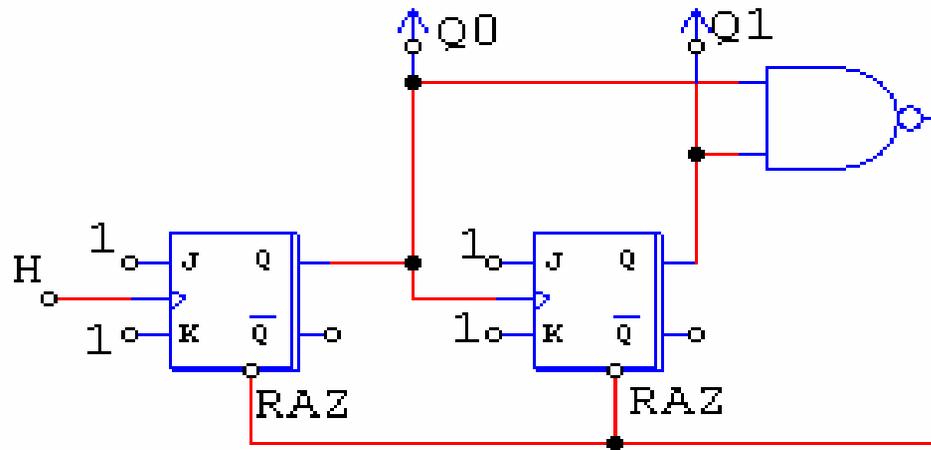


Les compteurs

Les compteurs réalisent la fonction de dénombrement des événements

Deux types : asynchrone et synchrone

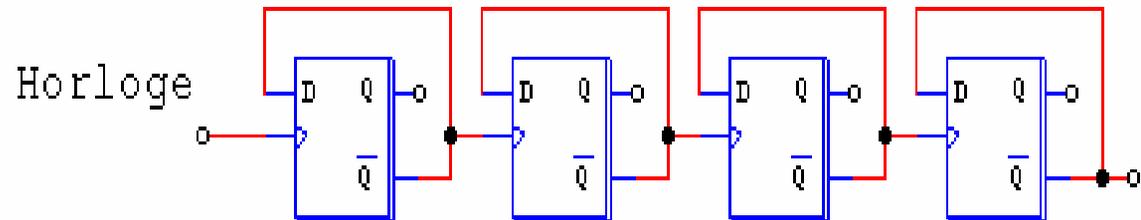
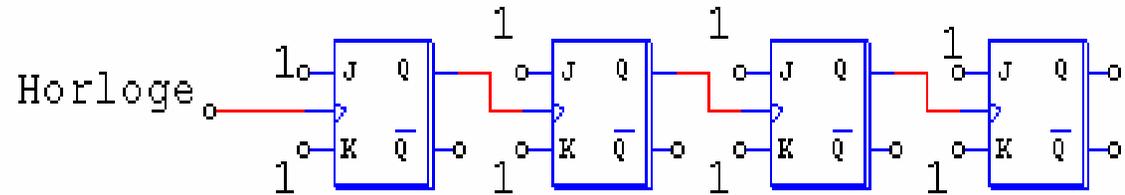
Asynchrone



Modulo - n

Les compteurs

Asynchrone

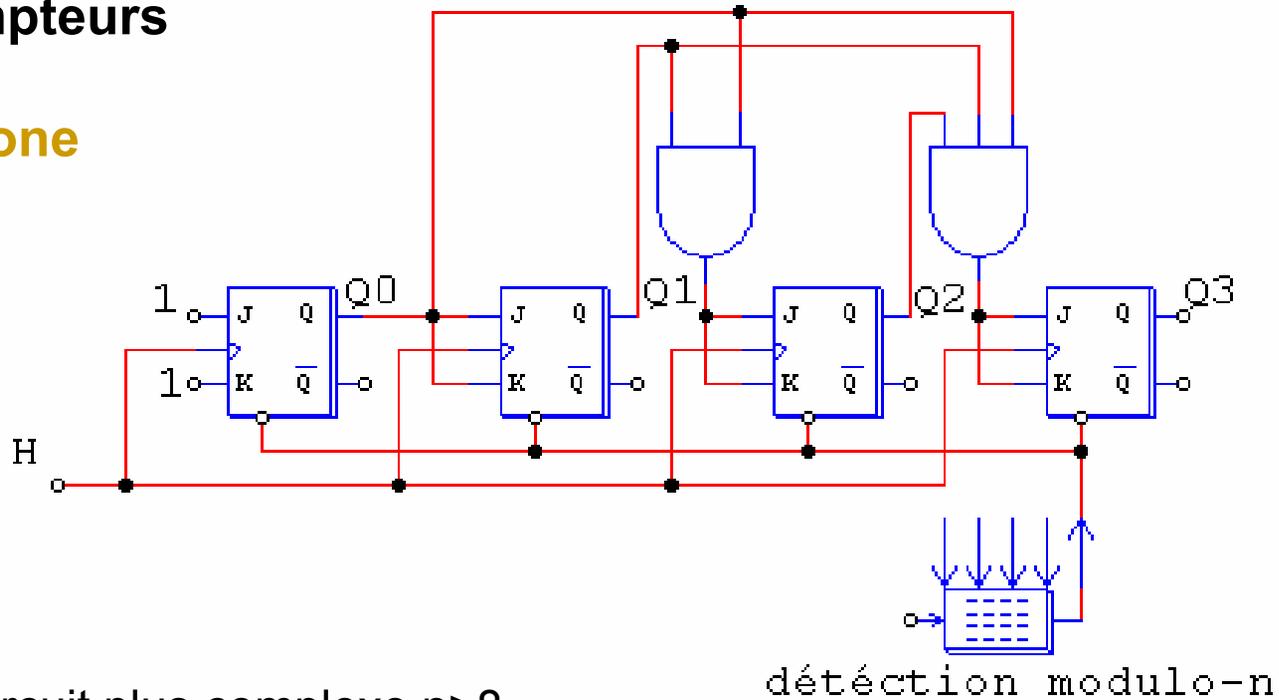


- + circuit plus simple
- Temps de comptage plus long
- Problème d'aléas

Les applications

Les compteurs

synchrone



- circuit plus complexe $n > 8$
- + Rapidité de comptage
- + pas de problème d'aléas

SN74LS190/191

Les architecture des processeurs

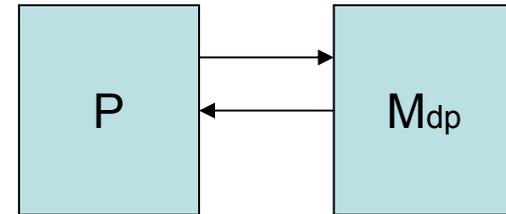
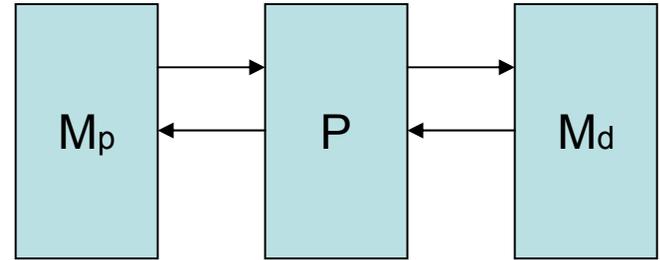
Historique :

Hardware

1944 Mark
1946 ENIAC

⊖ Organisation complexe

Von Neuman 1945



Devenu un standard pour les architectures a utilisations générales

ALU + shifter (add, sub, shift) opérations plus faciles à réaliser

Instr. Compl. (x , /) sont réalisées par une série de shift/ADD ou SUB

dans ROM

CISC

Les architectures des processeurs

Principes de fonctionnement

L'architecture de type Von Neuman (1945)

Caractéristiques :

- L'information est codée en binaire et représentée par un ensemble de mots.
- Les instructions et les données sont codées de la même façon et sont dans la même mémoire.
- Les mots dans la mémoire se distinguent par leurs adresses.
- L'utilisation d'un seul bus pour le transfert des données.

Les architectures des processeurs

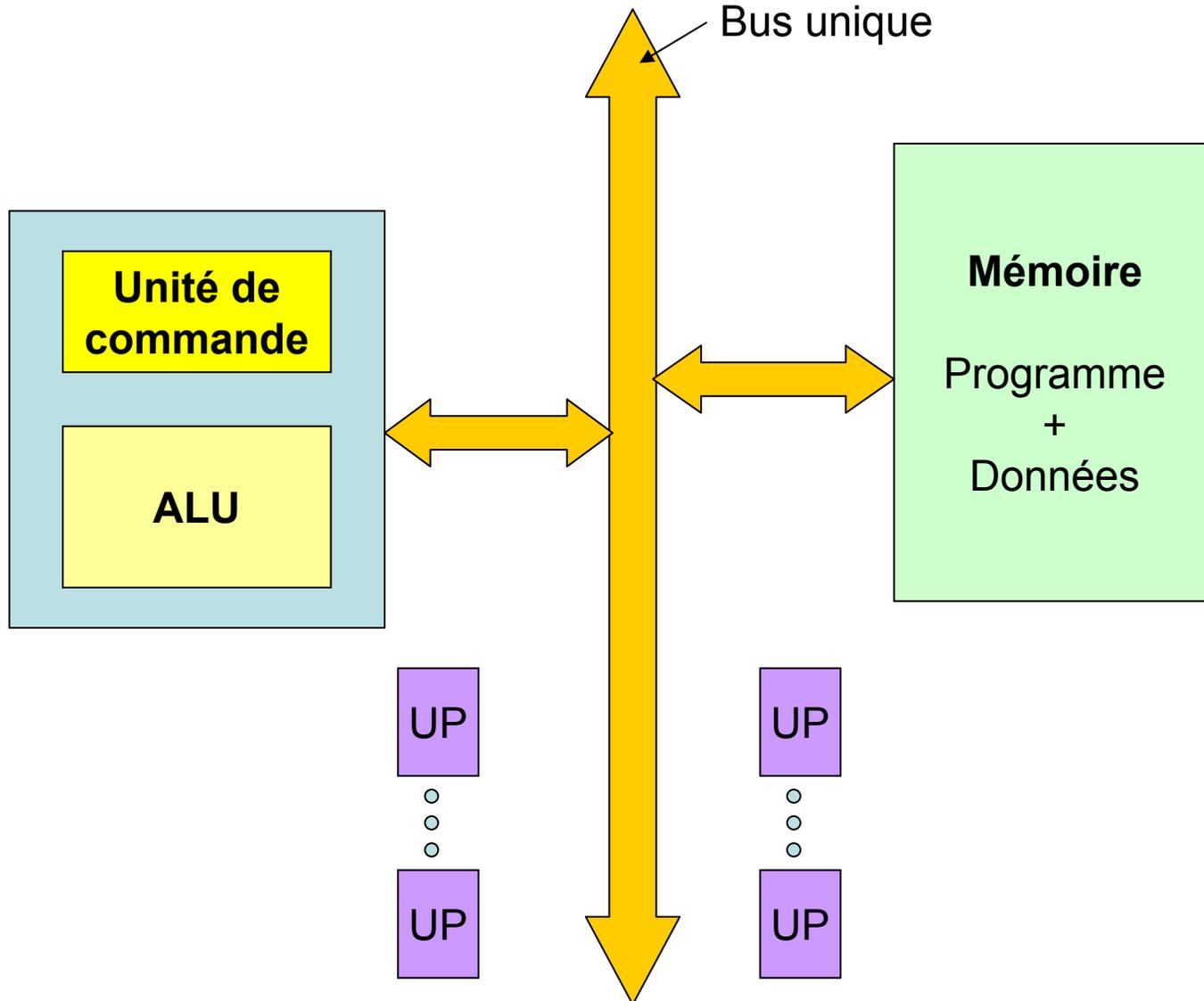
Principes de fonctionnement

L'architecture de type Von Neumann (1945)

Caractéristiques :

- L'algorithme pour résoudre le problème est réalisé sous forme de programme.
- Le programme est une suite des instructions.
- L'exécution séquentielle des instructions.
 - * Extraction de l'instruction;
 - * Décodage de l'instruction;
 - * extraction des opérandes;
 - * Exécution de l'opération;
 - * Stockage du résultat.

L'architecture Von Neumann



Les architectures des processeurs

Principes de fonctionnement

Définitions :

Instruction: ensemble de micro- opérations réalisées à un instant donné effectuant une opération spécifique.

Micro- opération: une des actions réalisée au niveau du processeur lors de l'exécution d'une instruction.

Les architectures des processeurs

Principes de fonctionnement

Plusieurs catégories d'architectures Von Neumann selon la manière d'adressage spécifiée dans l'instruction.

COP

COP	Adr. Opérande 1
-----	-----------------

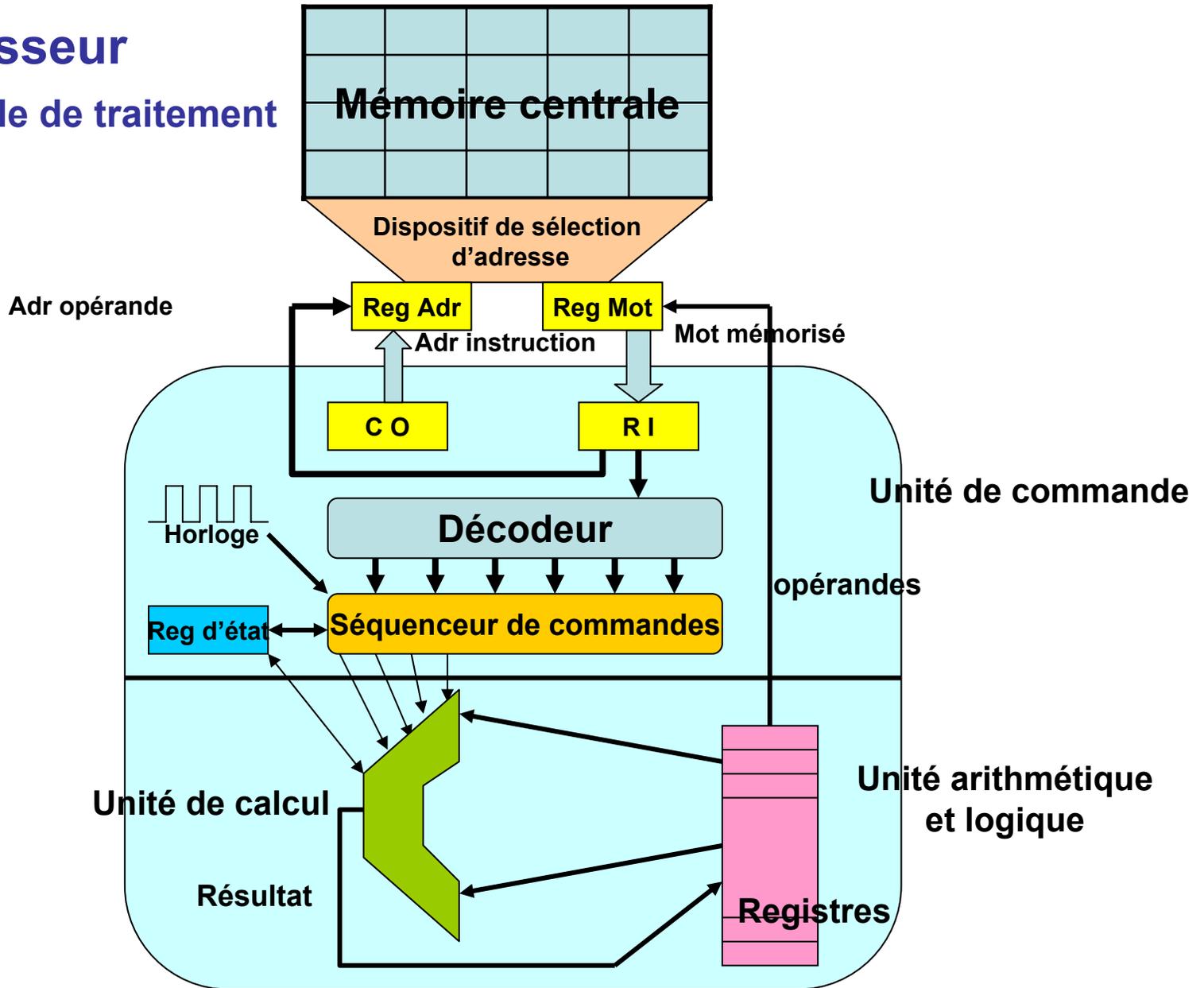
COP	Adr. Opérande 1	Adr. Opérande 2
-----	-----------------	-----------------

COP	Adr. Opérande 1	Adr. Opérande 2	Adr. résultat
-----	-----------------	-----------------	---------------

COP	Adr. Opérande 1	Adr. Opérande 2	Adr. résultat	Adr. Instr. Suiv.
-----	-----------------	-----------------	---------------	-------------------

Le processeur

Unité centrale de traitement
CPU



Composition d'un processeur:

Mémoire centrale (programmes et données)

Unité centrale de traitement (exécution des programmes)

Unités d'E/S (échange d'informations avec les unités périphériques)

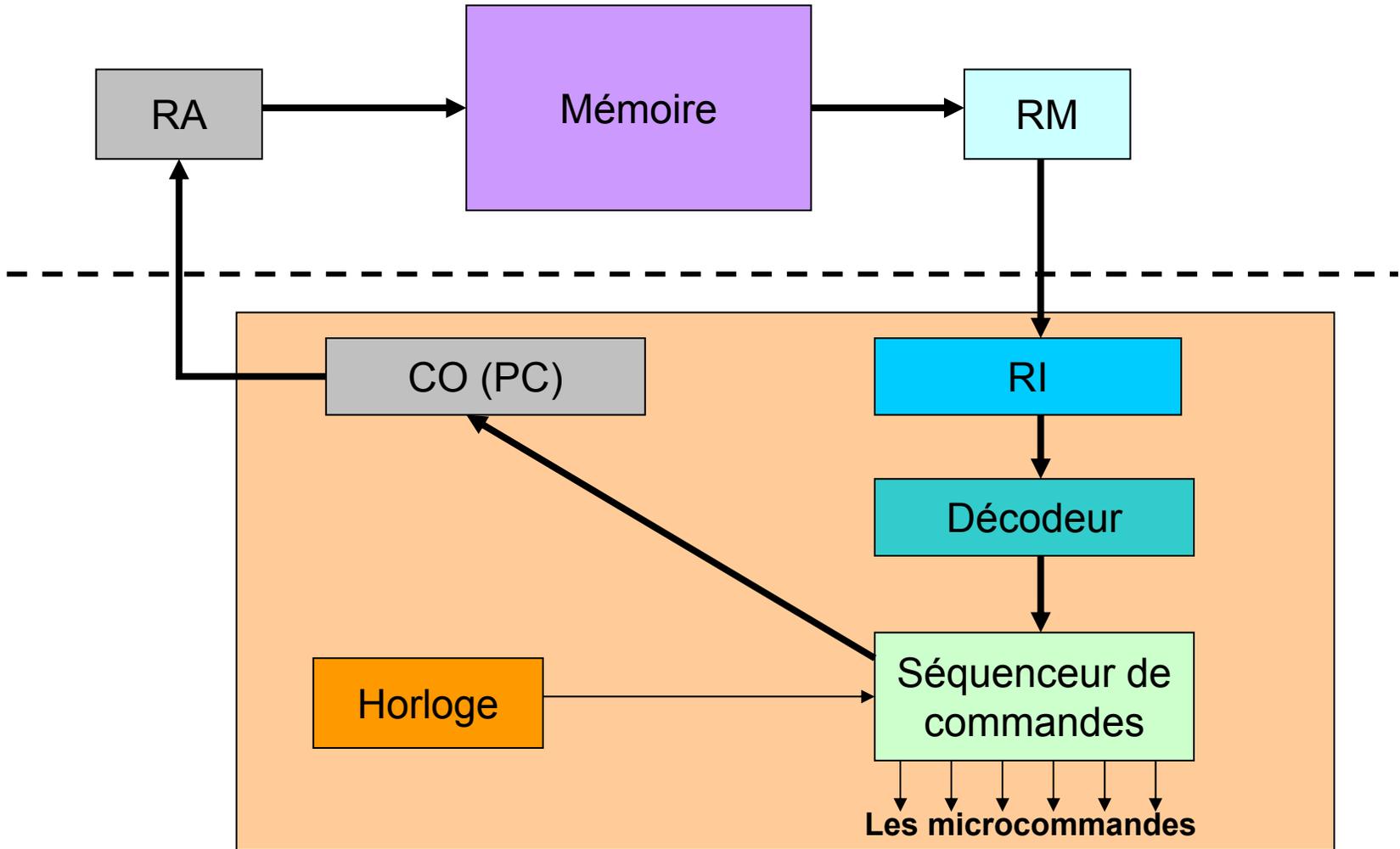
L'unité centrale de traitement

- **L'unité de commande** s'occupe de gérer l'exécution d'un programme.

A – Deux registre importants :

- 1 – **Le registre d'instruction (RI)** : contient l'instruction en cours d'exécution.
- 2 – **Le compteur ordinal (PC)** : contient toujours l'adresse de la prochaine instruction à exécuter. Il est automatiquement incrémenté (**taille**).

Le cycle de l'exécution des instructions

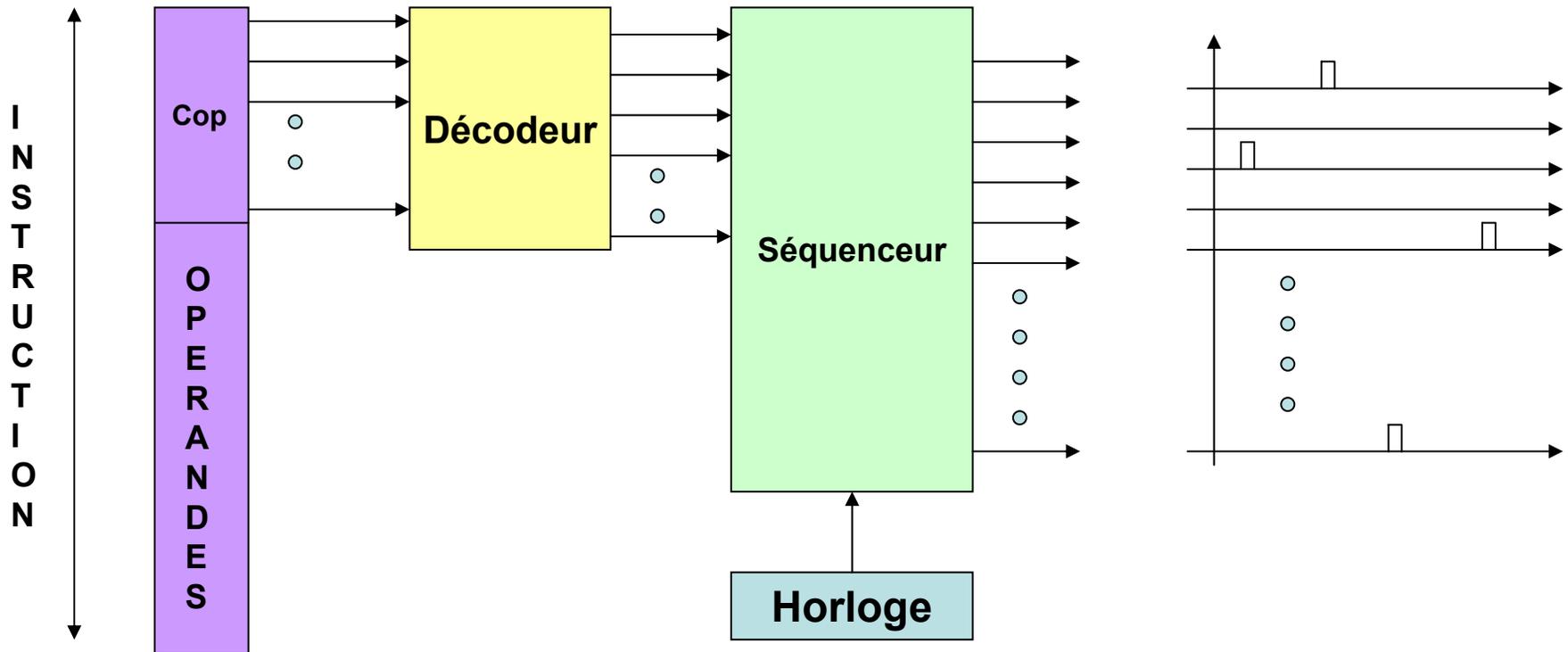


B – Décodeur et séquenceur de commandes :

1 – Le décodeur est un dispositif de décodage des instructions. C'est un circuit combinatoire qui permet à partir du champ du code opération de l'instruction de générer les différents signaux nécessaires à l'entrée du séquenceur.

2 – Le séquenceur de commande est un circuit séquentiel qui active les Circuits nécessaires à l'exécution de l'instruction en cours. Cette unité a besoin des signaux d'une horloge pour enchaîner les commandes

- Séquenceur câblé : réalisé entièrement à base de logique combinatoire et de la logique séquentielle



+ Plus rapide

- Complexité de réalisation

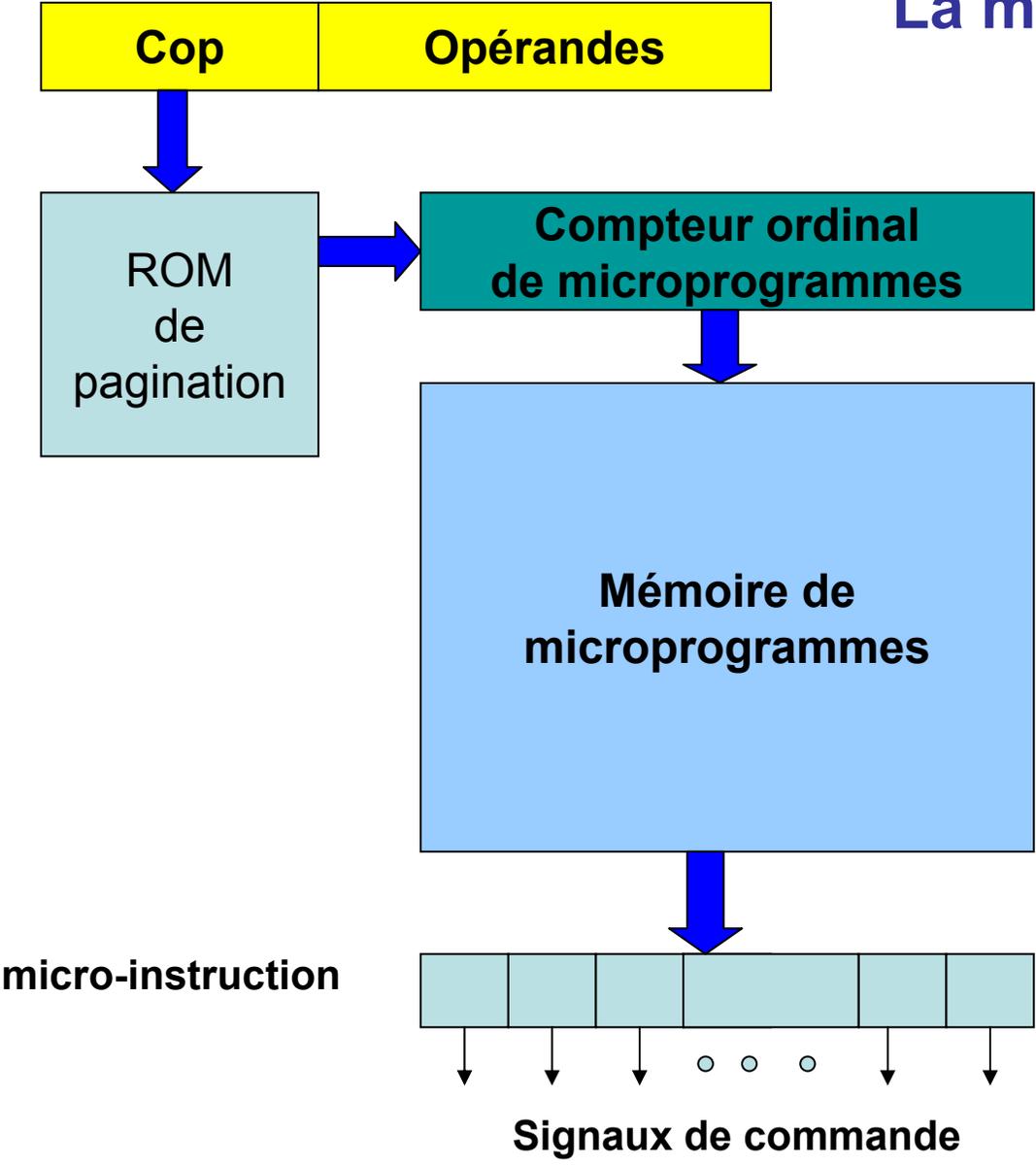
- **Séquenceur micro programmé** : réalisé entièrement à base d'une mémoire ROM de pagination et de microprogrammation.

Nécessite un compteur ordinal pour la lecture séquentielle des micro commandes à partir de la mémoire de microprogrammes.

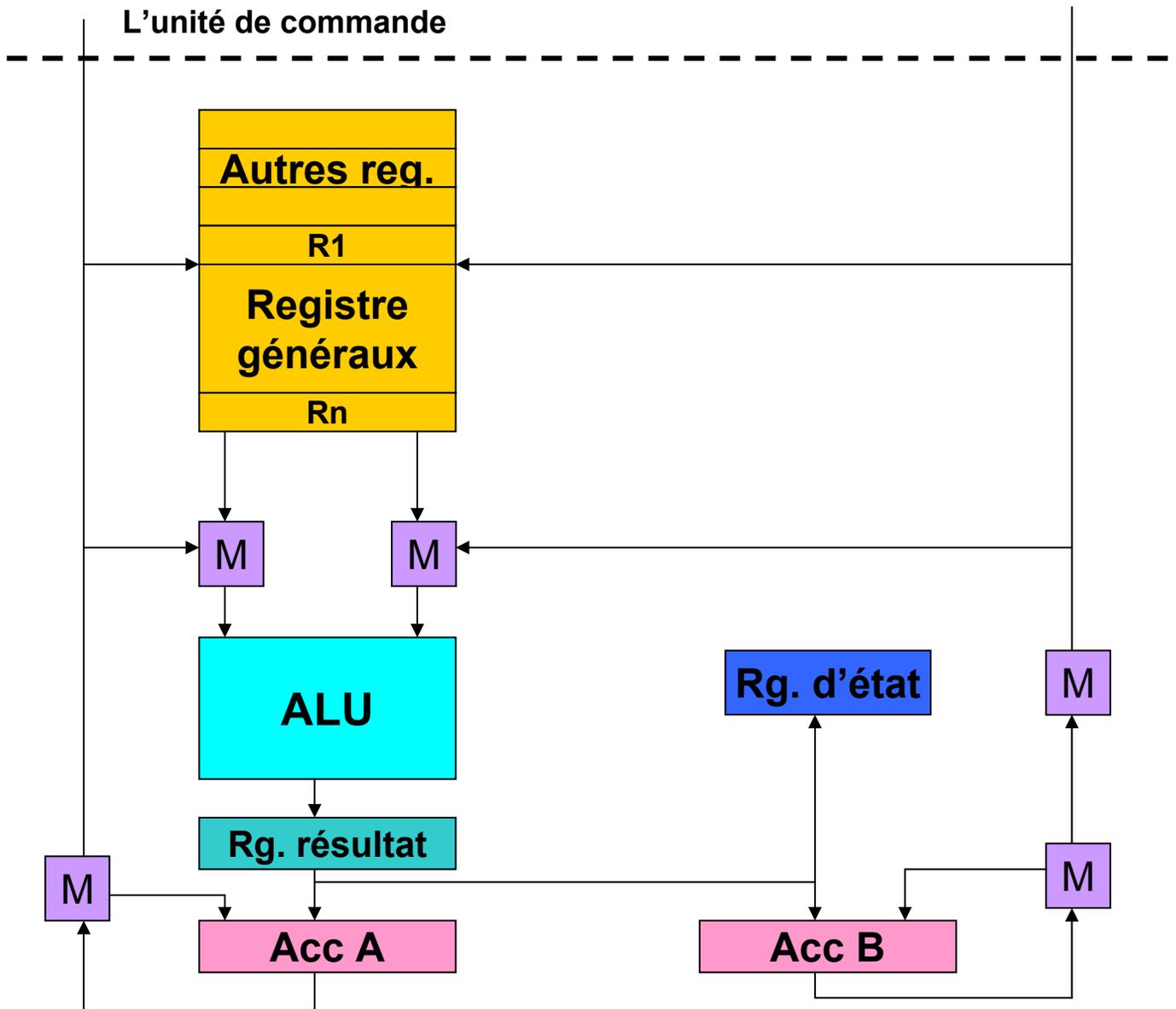
+ simple à réaliser

- Moins rapide

La micro programmation



L'unité arithmétique et logique



Les registres

- Les registres arithmétiques (Acc)

- Les registres de base et d'index

- Les registres banalisés

- Le registre d'état (PSW)

- Autres registres comme : Rgs à décalage (shift reg)
Rgs pour op. à VF

Le registre d'état (PSW)

Indique l'état du système après opération arithmétique ou logique

Les bits (drapeaux, flags) : indiquent l'état d'une condition particulière dans le CPU



C : Carry flag , Retenue

V : Overflow flag , Dépassement de capacité

Z : Zero

N : Negative

I : Interrupt mask

Les modes d'adressage

Pour faciliter la programmation, les fabricants offrent toute une gamme de méthodes pour adresser les opérandes. Le format des instructions prévoit un champ dans le Cop dont les bits indiquent le mode choisi.

- Adressage direct
- Adressage indirect
- Adressage immédiat
- Adressage implicite
- Adressage indexé
- Adressage basé
- Adressage relatif
- Une combinaison des modes

Ad. effective

Ad. De l'Ad (plusieurs niveaux)

l'opérande lui même

Indiqué dans le Cop

Ad=Champ Ad + Rgx

Ad=Champ Ad + Rgb

Ad=Champ Ad + CO

Le registre pointeur de pile SP – Stack pointer

La pile est une zone de stockage organisé en LIFO.

Le SP est un registre contenant le niveaux de remplissage de la pile.

2 opérations fondamentales :

PUSH

PULL

Le registre pointeur de pile SP – Stack pointer

- * **La pile est une structure dynamique.**
- * **La pile conserve l'ordre de l'exécution des événements.**
- * **En arithmétique la pile garde les op. et les rés. Int.**
- * **En appel à des sous-routines la pile garde l'adresse de retour.**
- * **En traitement des interruptions la pile garde l'état du Processeur.**
- * **En appel aux procédures la pile est utilisé pour la Passation des paramètres.**

Les mémoires

Dans un ordinateur deux caractéristiques essentielles :

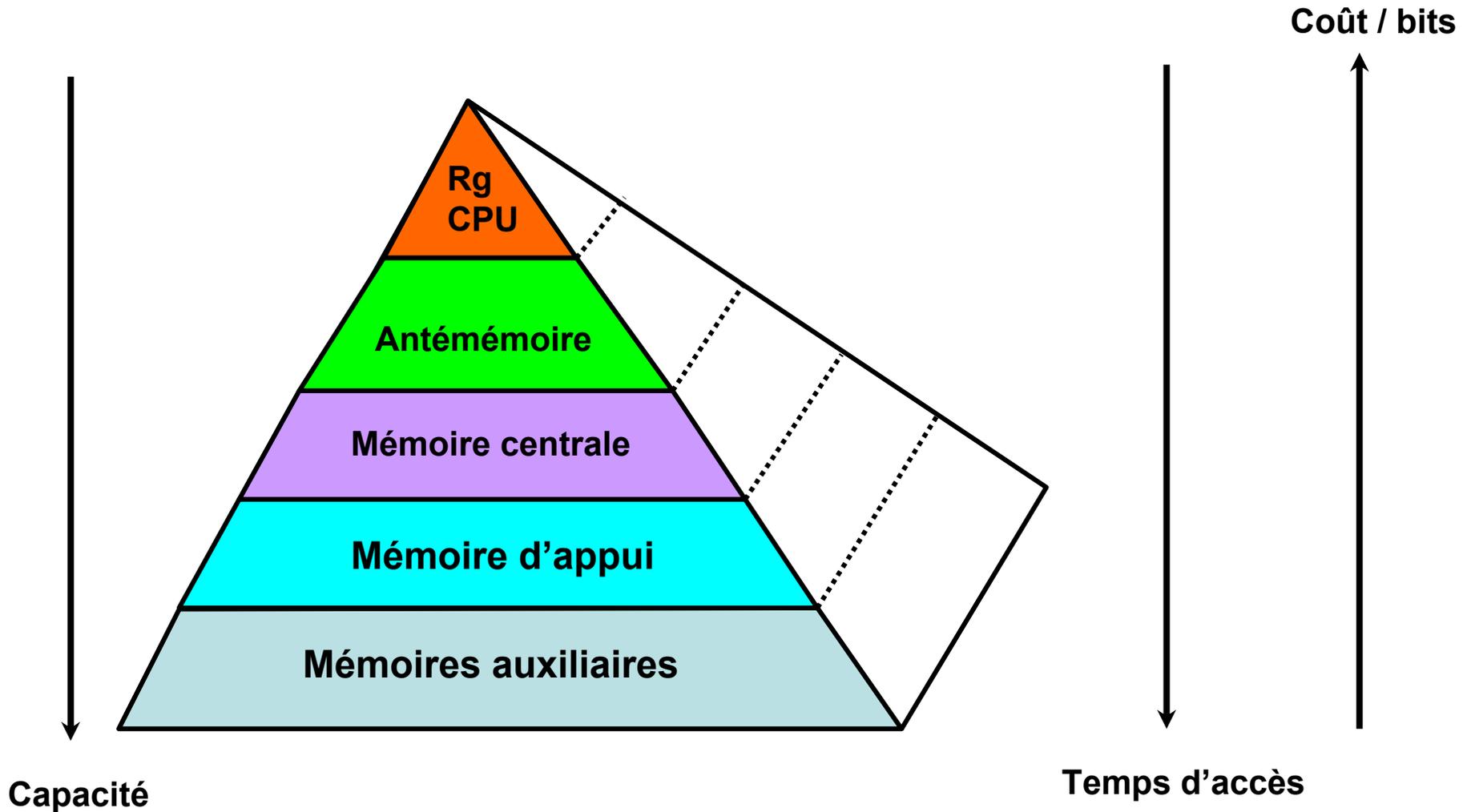
La vitesse de traitement

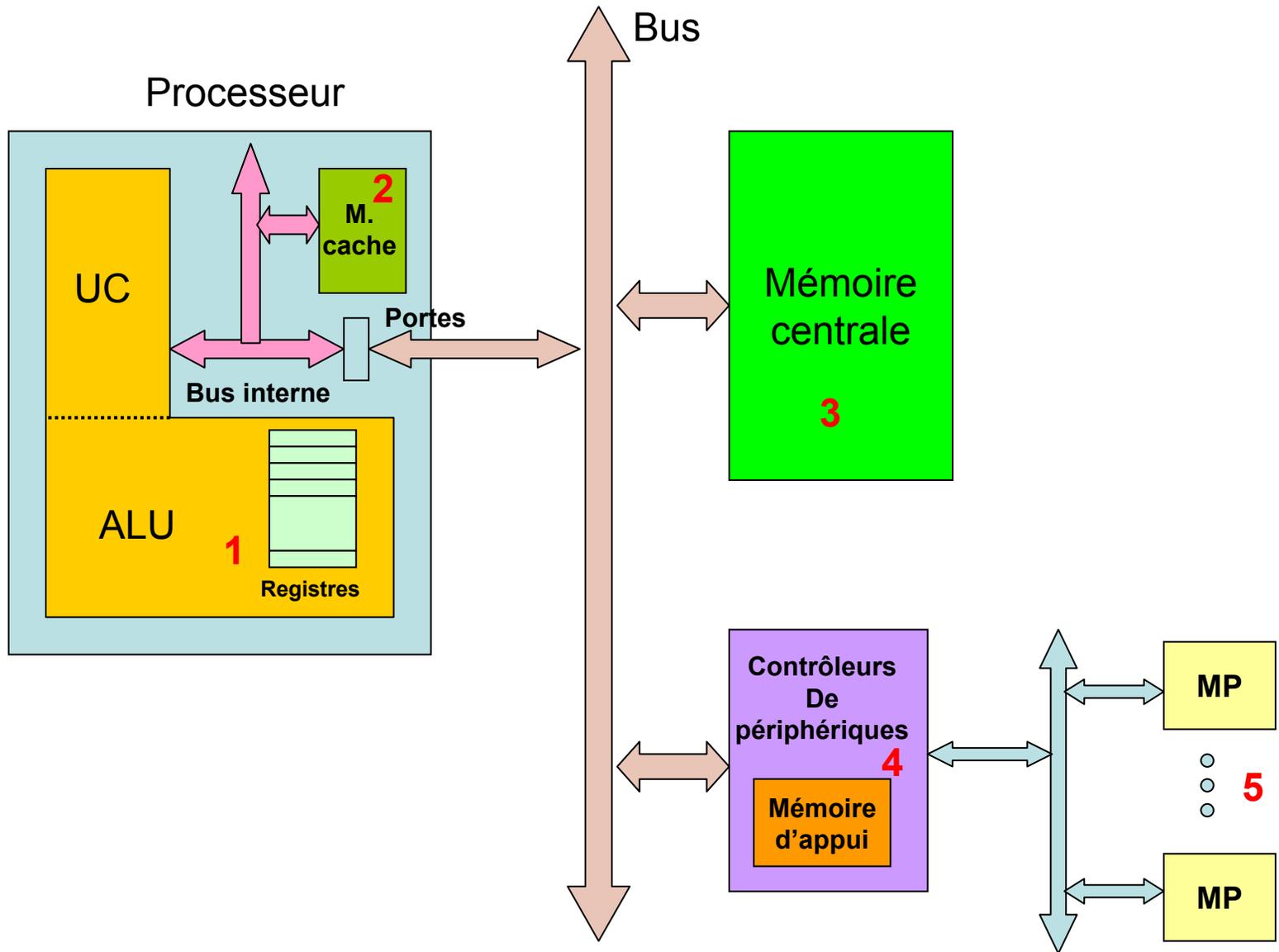
Capacité de mémorisation

Une mémoire : dispositif capable d'enregistrer, de conserver et de restituer des informations codées en binaire.

La mémoire se caractérise par sa **capacité**, son **le temps d'accès** et son **coût par bit**

Hiérarchie des niveaux de mémoires





Classification par mode d'accès

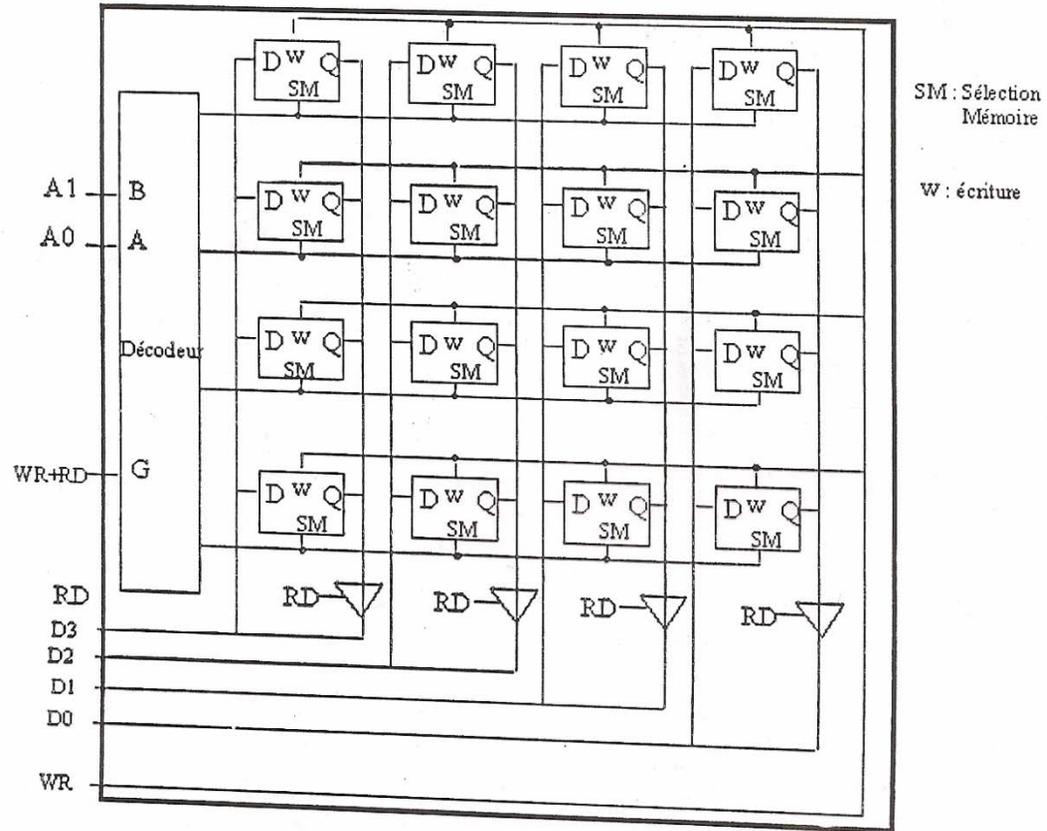
Mémoires à accès séquentiel

Mémoires à accès semi séquentiel

Mémoires à accès aléatoire

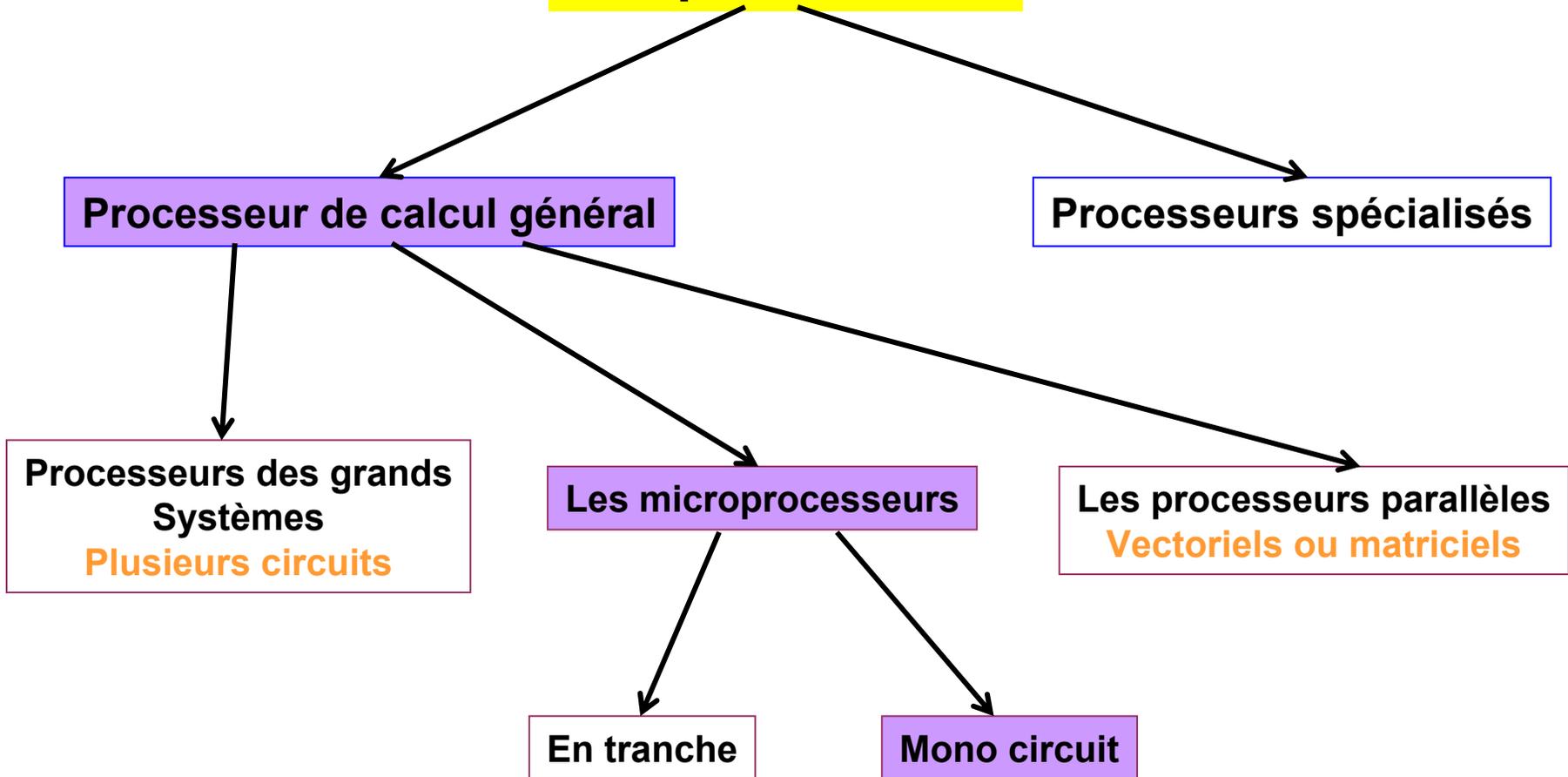
**Mémoires à accès par le contenu :
Mémoire associative**

Fonction de mémorisation : bascule et registre



Classification des processeurs

Les processeurs



Classification des processeurs

Les processeurs

Processeur de calcul général

Processeurs spécialisés

Processeurs parallèles

Cellules à fonction spécifique

Processeurs programmable

Processeurs à programme figé

ASIC

A Utilisation générale

Domaine spécifique

Définitions

Microprocesseur

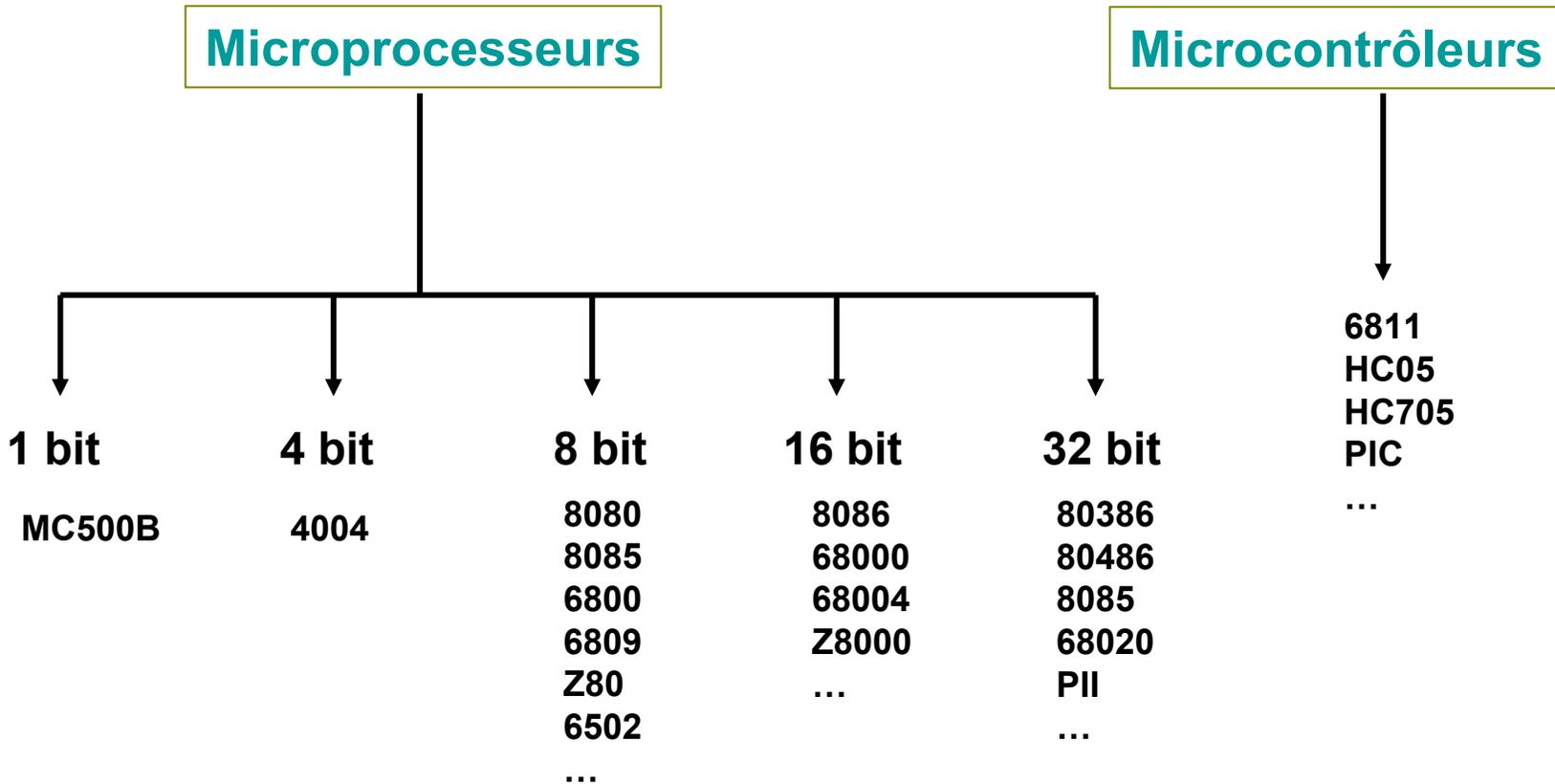
Microprocesseur en tranche

Système microprocesseur

Microcontrôleur

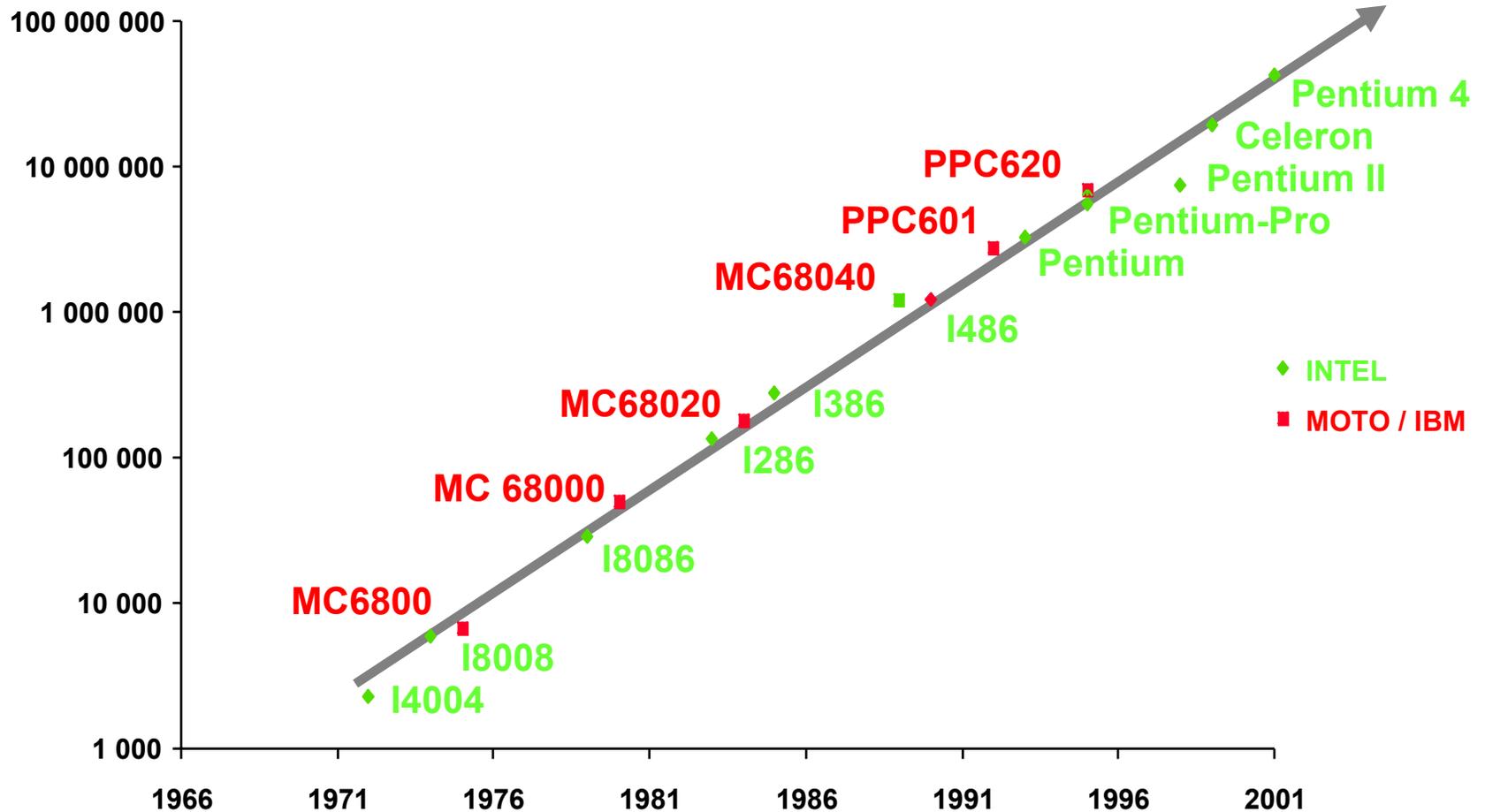
Famille microprocesseur

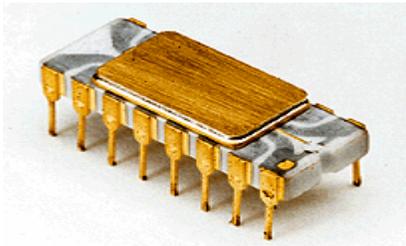
Types des microprocesseurs



Évolution de la complexité

Nb Tr



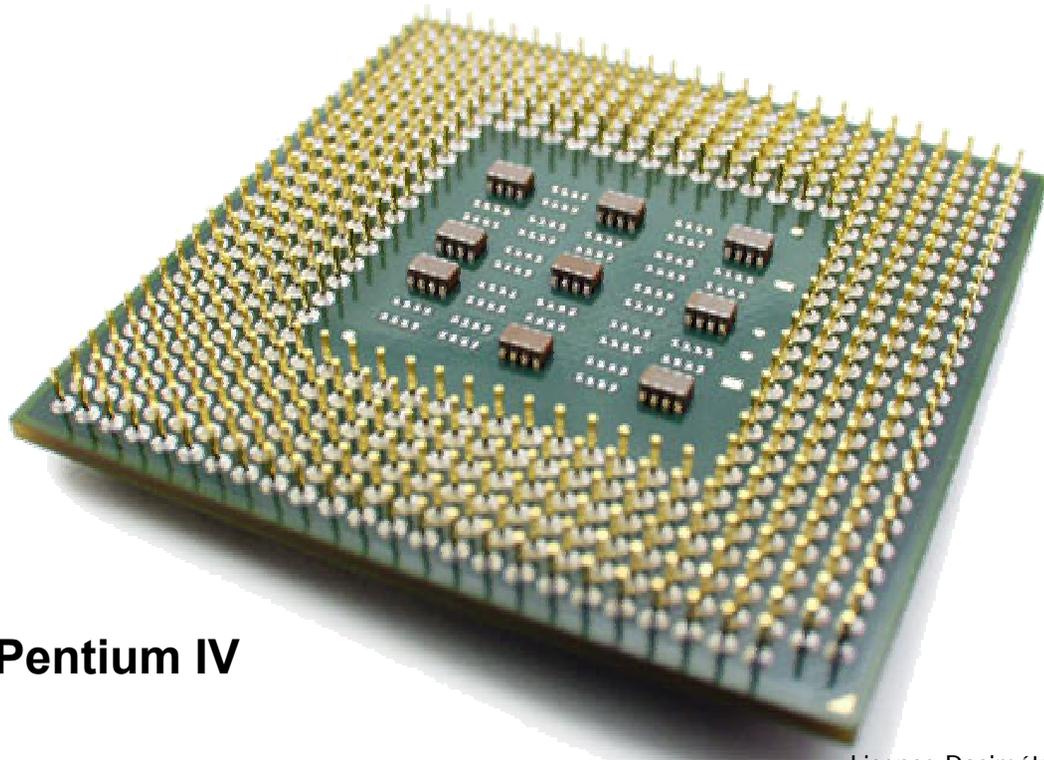


Intel4004

Pentium I



Pentium IV



La famille du microprocesseur MC6800 de MOTOROLA

MCM6800

Le microprocesseur

MCM6810

128x8-bits RAM

MCM6830

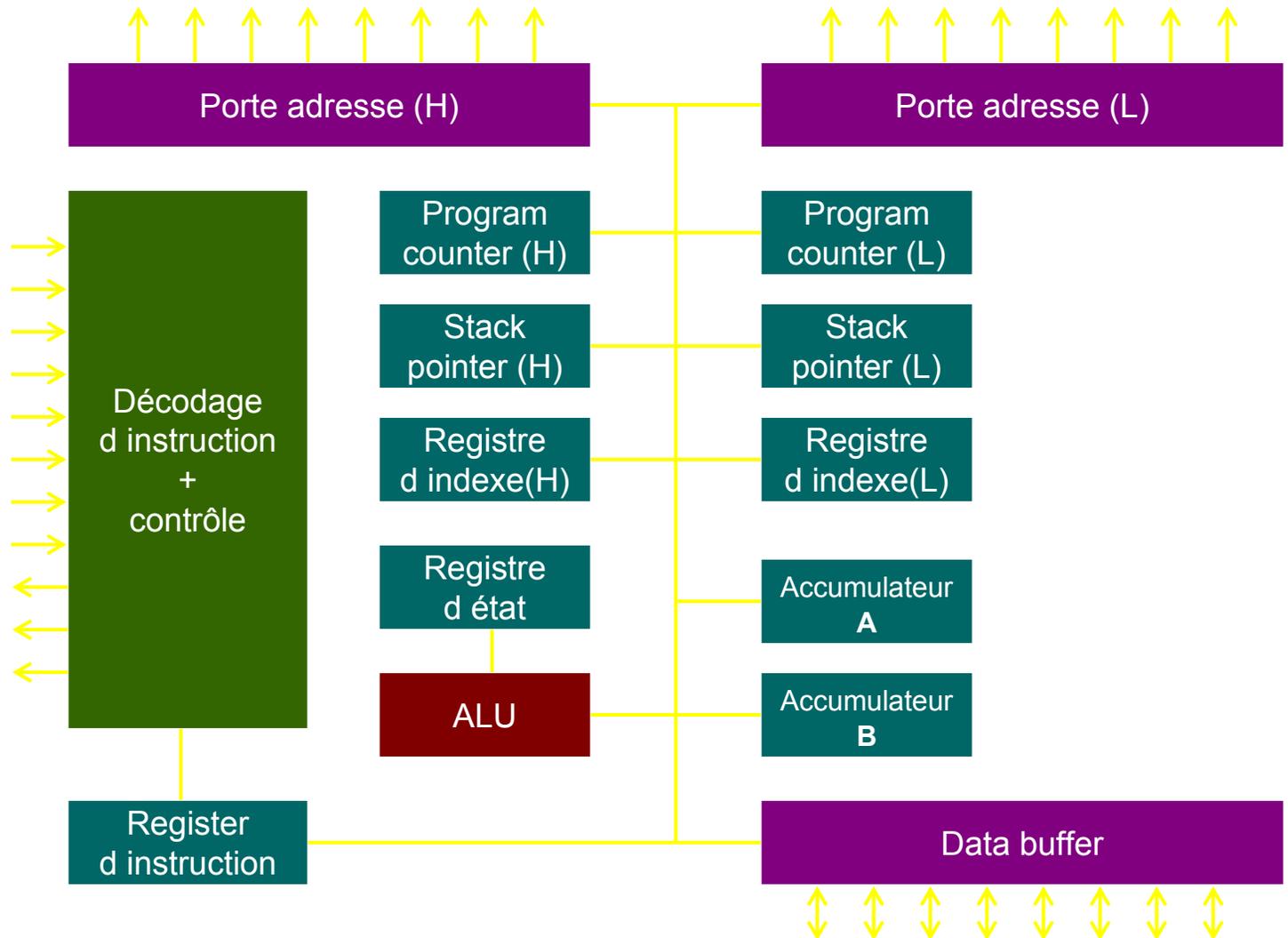
1024x8-bits ROM

MCM6821

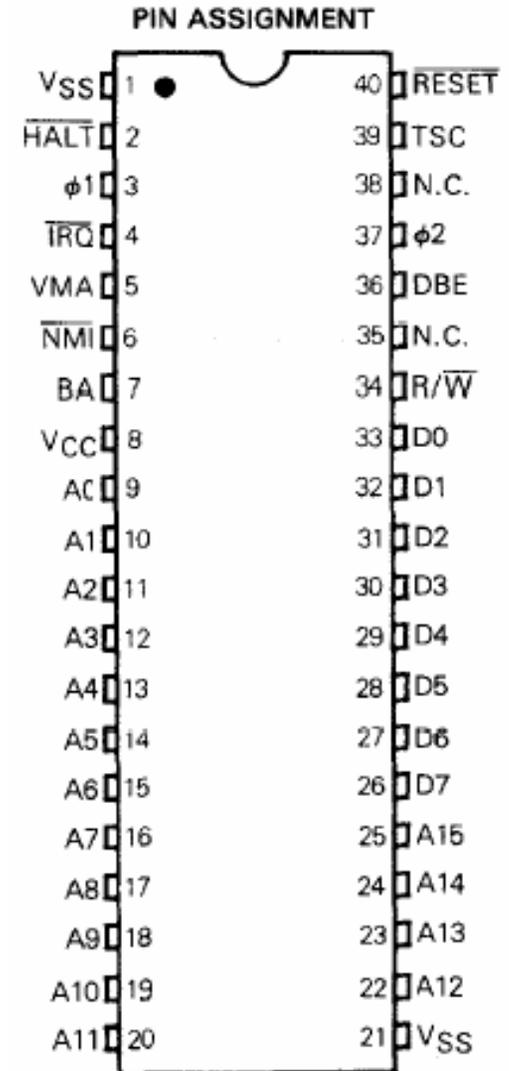
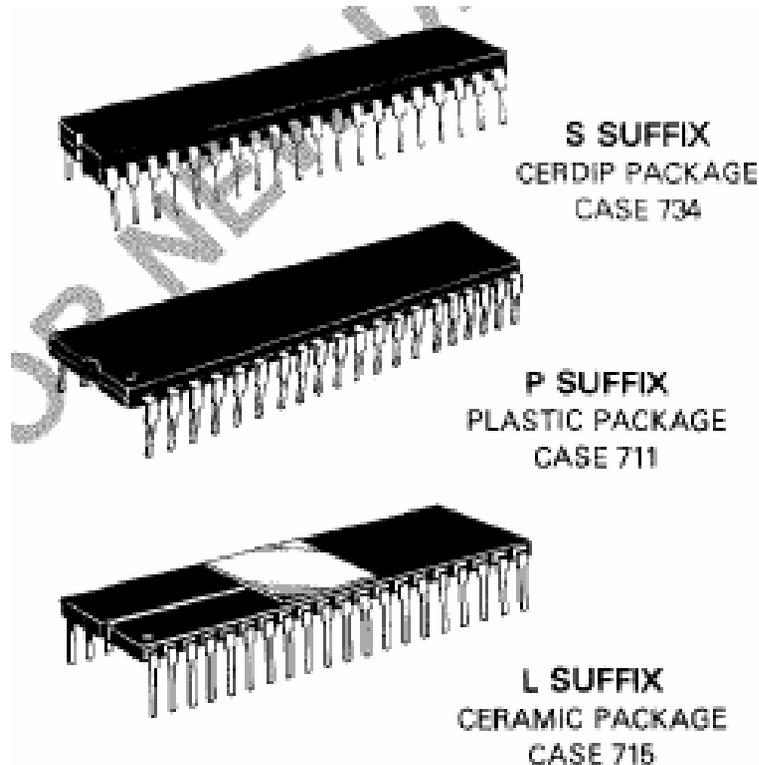
PIA

MCM6850

ACIA



Les pins de MC6800

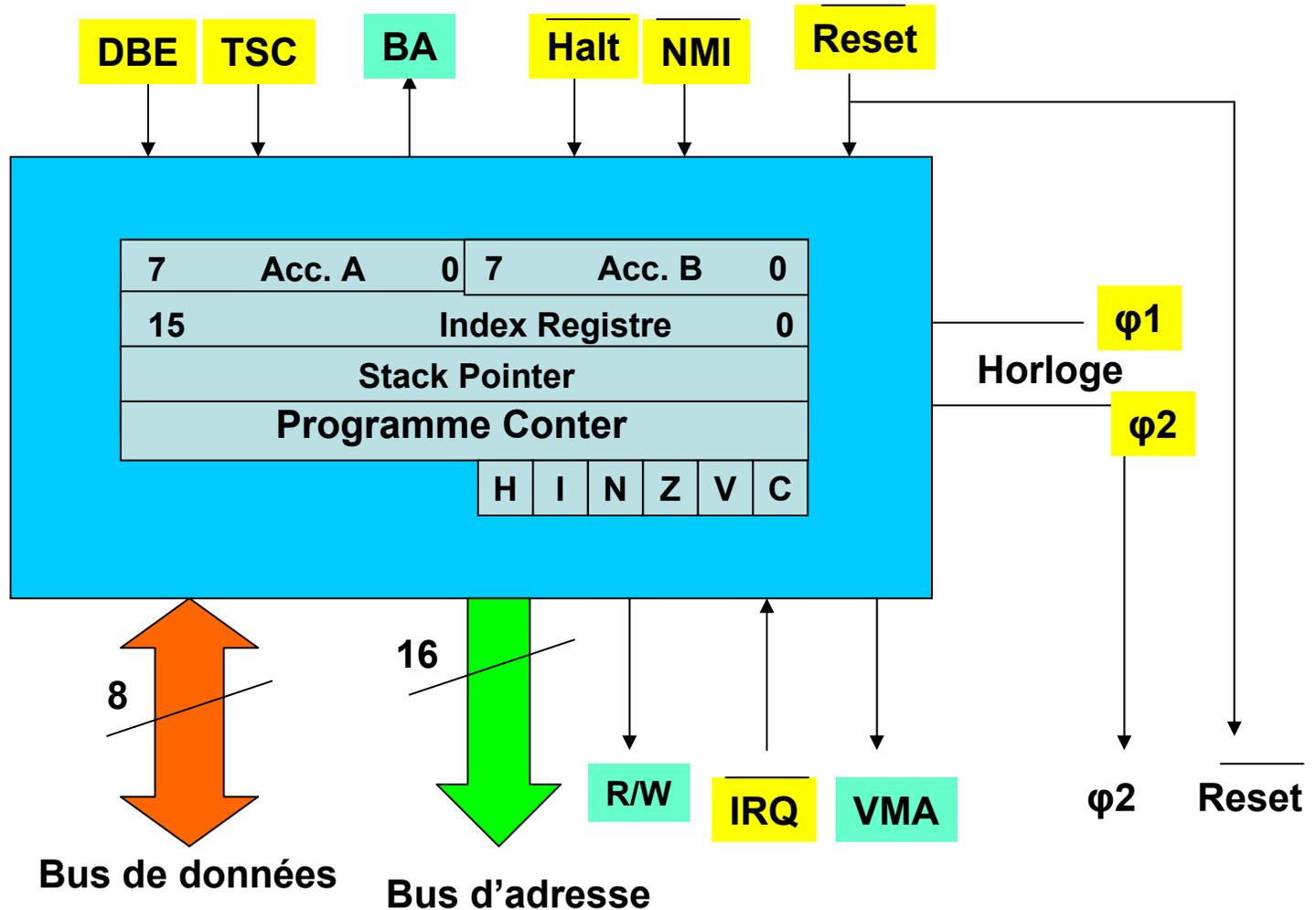


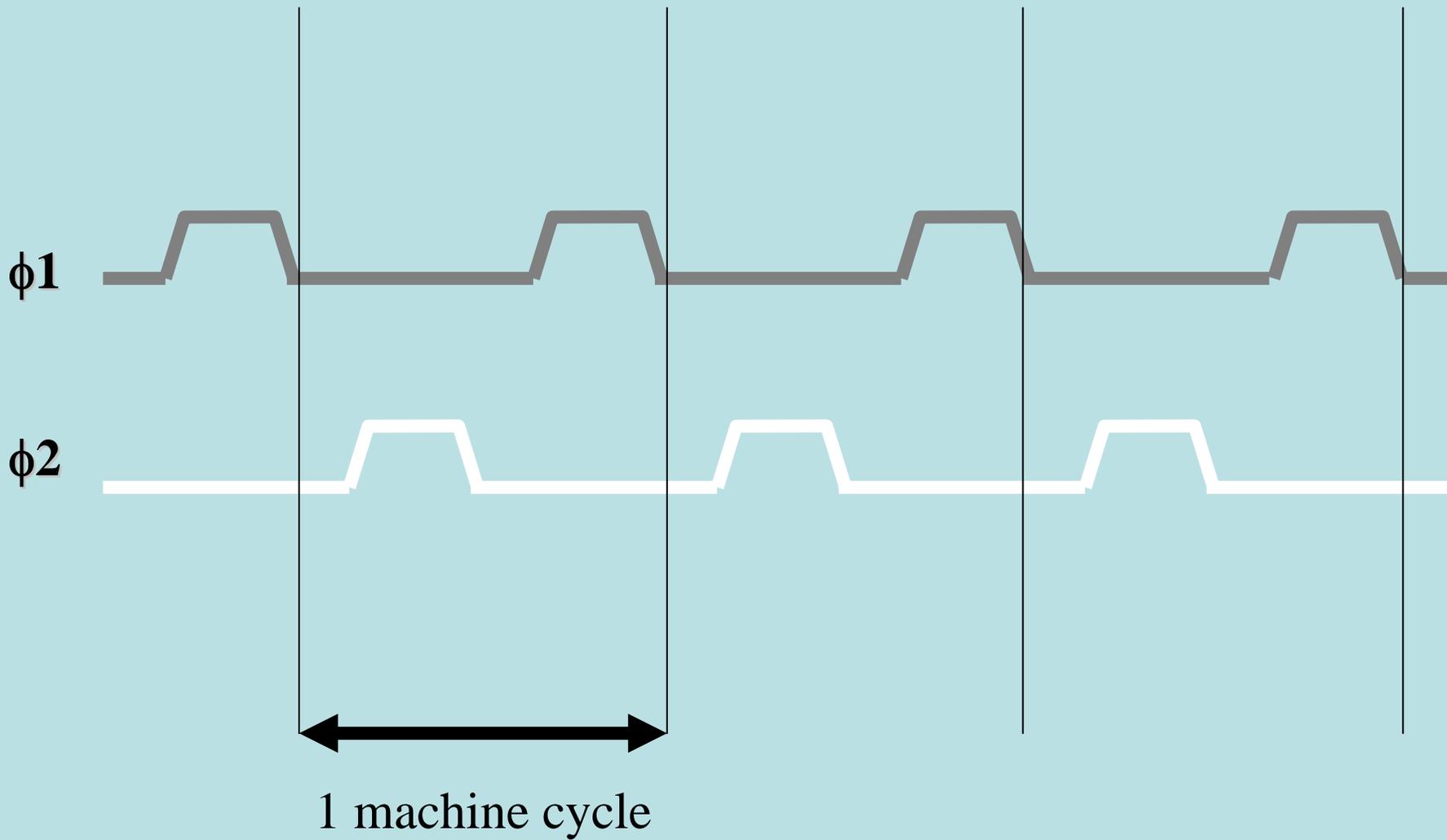
Les pins de MC6800

- Nom des signaux et type de pin

Signal name	Function	Data type
A0 – A15	Address lines	Output
D0 – D7	Data lines	Bidirectional
R/W	Read and Write lines	Output
VMA	Valid Memory Address	Output
DBE	Data Bus Enable	Input
RESET	Reset Line	Input
HALT	Halt line	Input
BA	Bus Available	Output
TSC	3-State Control	Output
IRQ	Interrupt Request	Input
NMI	Non- Maskable Interrupt	Input
$\phi 1, \phi 2$	Phase 1 and 2 Clocks	Input
V_{CC}, V_{SS}	Power and Ground	Input

Le microprocesseur MC6800 de MOTOROLA



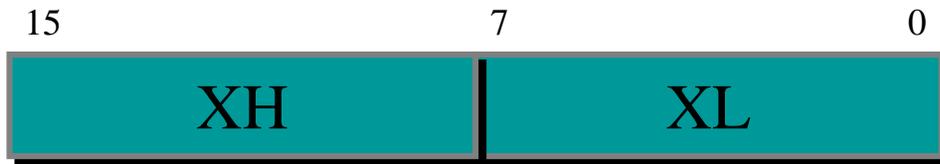




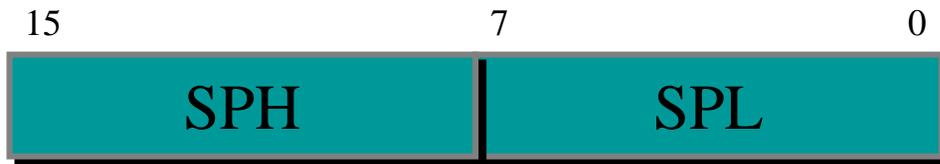
Accumulator A



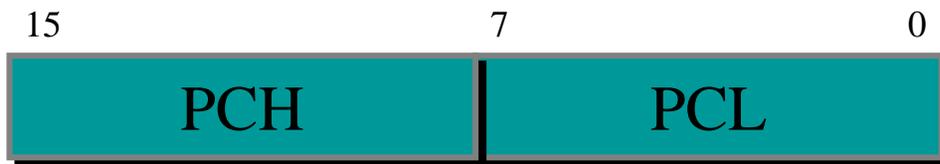
Accumulator B



Index Register (X)



Stack Pointer (SP)



Program Counter (PC)



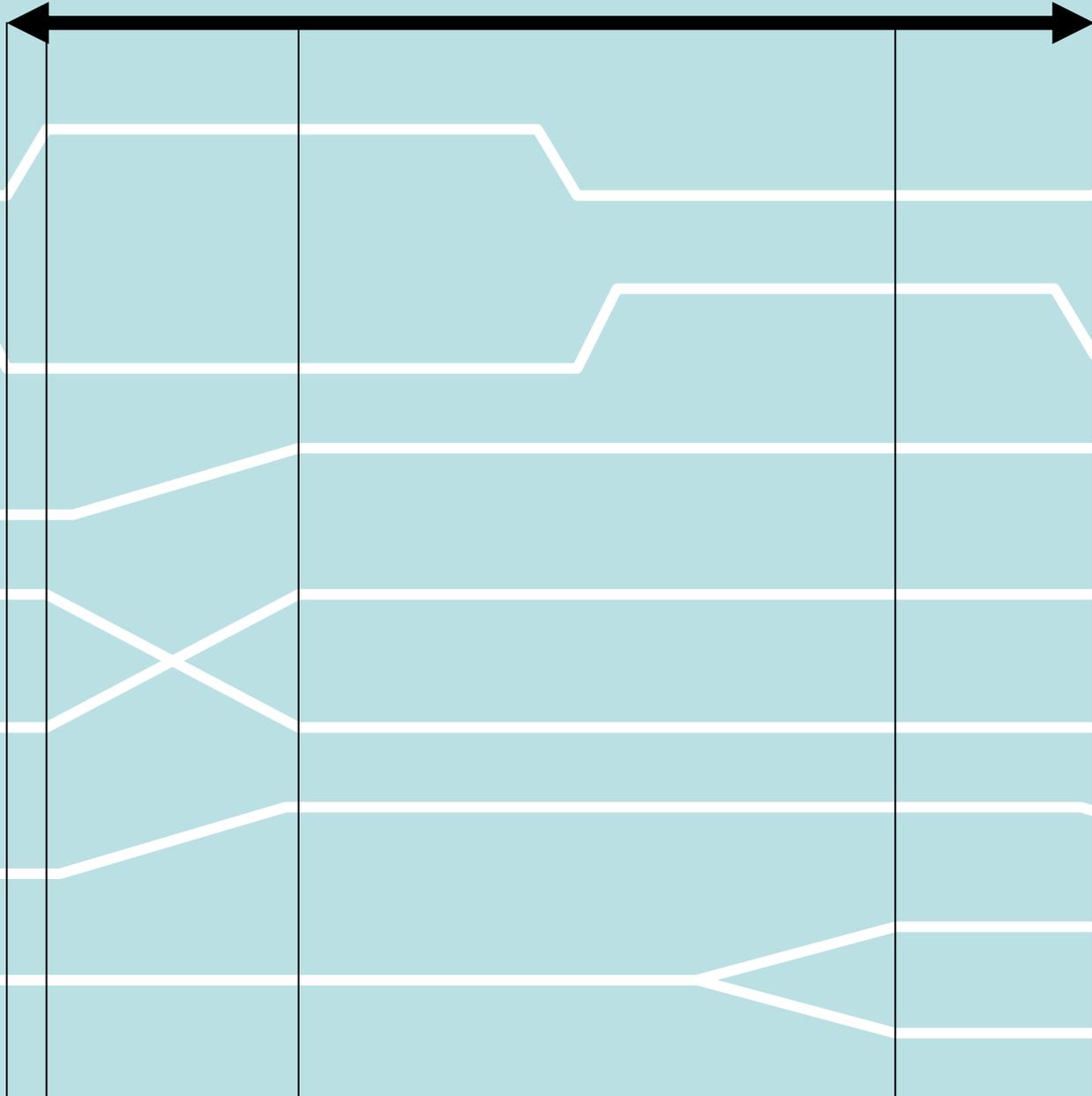
Condition Code Register (CCR)



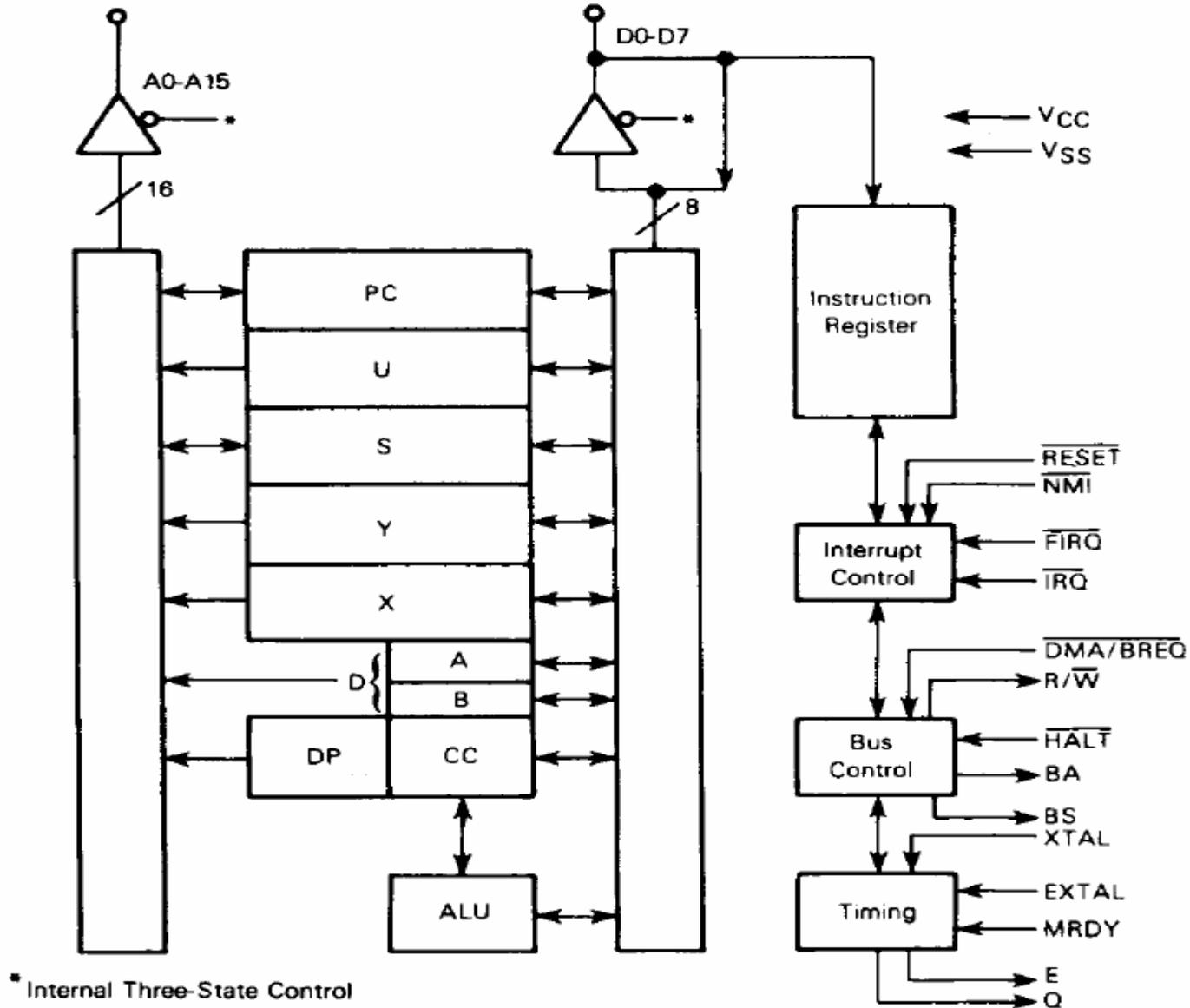
La table d'instruction

Voir MC6800.PDF

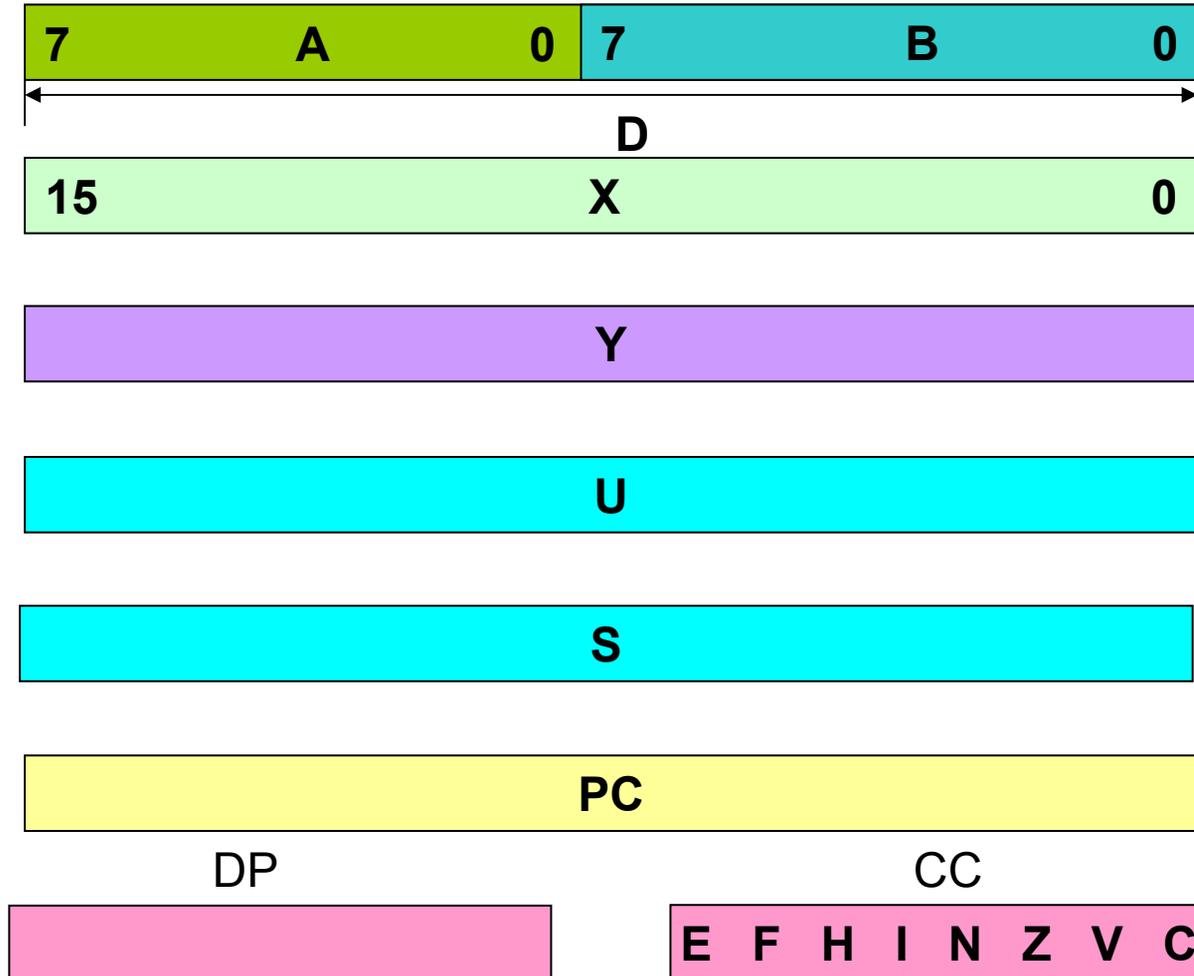
1 cycle



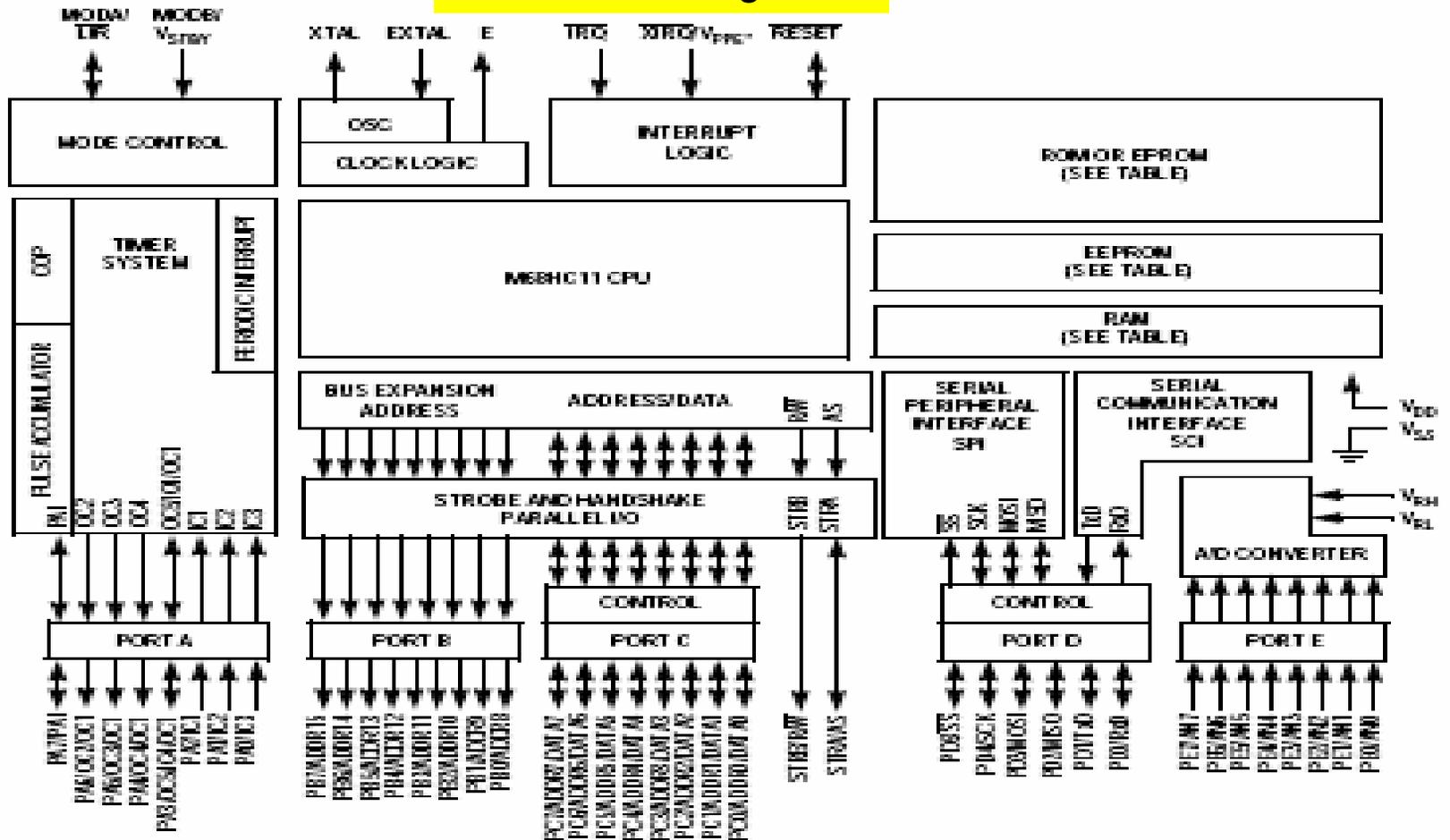
6809 Block Diagramm



Le 6809



6811 Block Diagramm

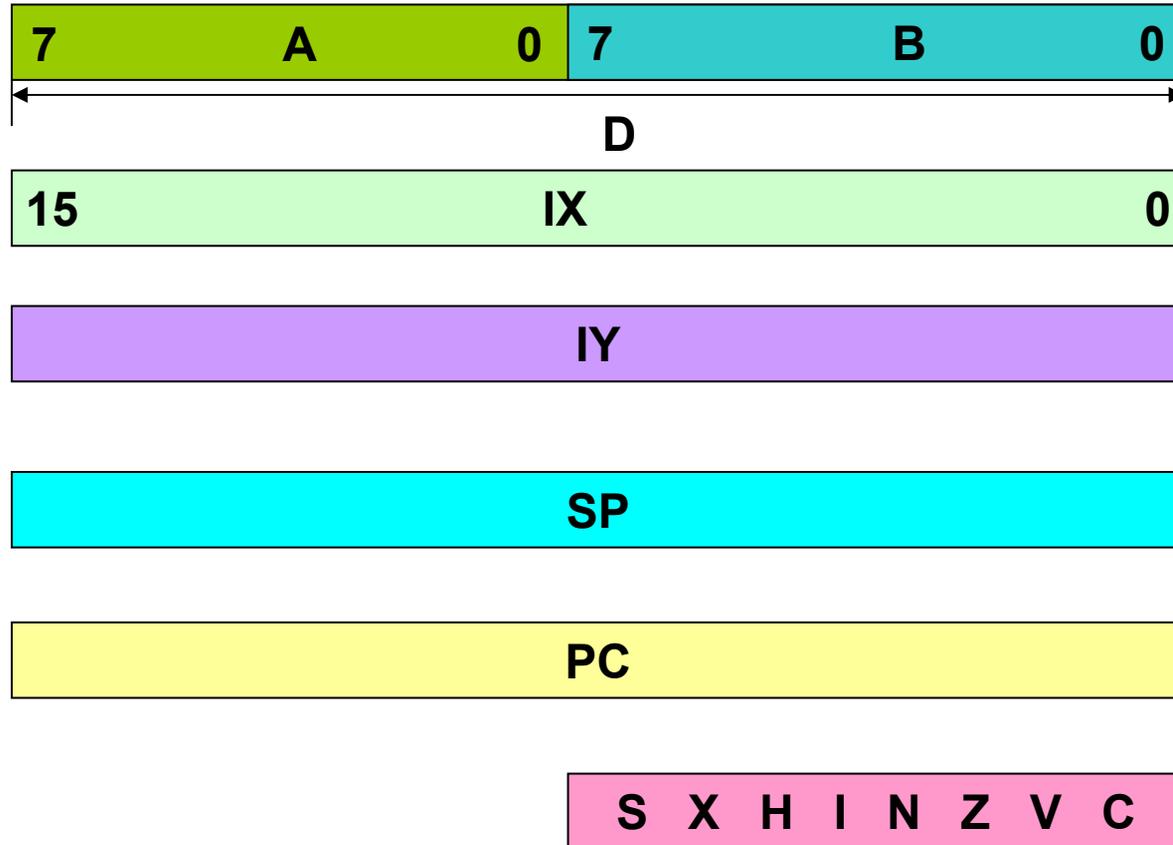


DEVICE	RAM	ROM	EPROM	EEPROM
M68HC11E0	512	—	—	—
M68HC11E1	512	—	—	512
M68HC11E8	512	12 K	—	512
M68HC11E8	512	—	12 K	512
M68HC11E20	256	20 K	—	512
M68HC11E20	256	—	20 K	512
M68HC811E2	256	—	—	2048

* V_{ppsc} applies only to devices with EPROM/OT PROM.

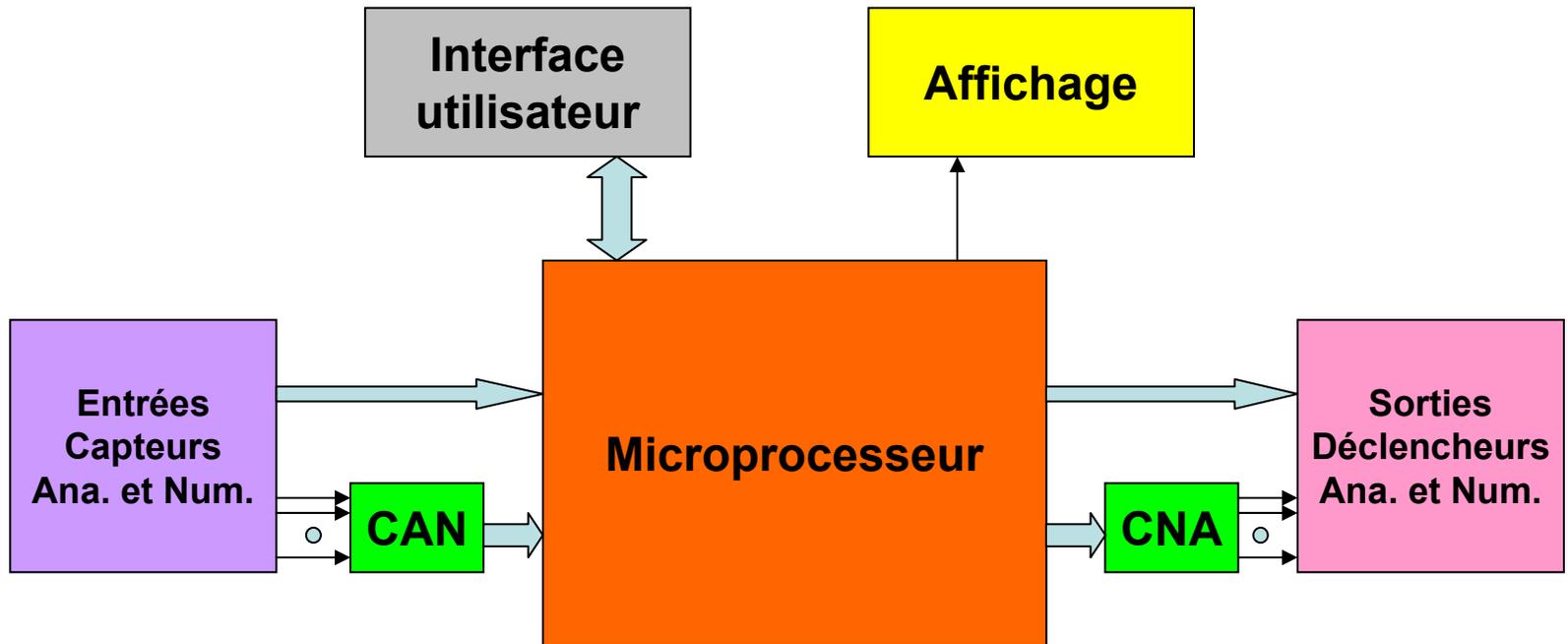
Figure 1-1. M68HC11 E-Series Block Diagram

Le 6811



La conception des circuits à microprocesseurs

Systeme microprocesseur type



La programmation

Quelques liens

<http://www.abcelectronique.com/>

<http://www.alldatasheet.com/>

<http://www.intel.com/>

<http://perso.orange.fr/xcotton/electron/constructeurs.htm>

<http://www.histoire-informatique.org/idx/>

<http://para.maxim-ic.com/>

<http://www.abcelectronique.com/>

<http://fanelectronique.free.fr/>

<http://www.mon-ordi.com/>

<http://www.ldlc.fr/>

http://cours.sofad.qc.ca/microinfo/page_princ.htm