Institut Universitaire de Technologie de Marseille
Aix*Marseille Université

Aix*Marseille université

Spécialité ISN

ARCHITECTURE DES MACHINES

(Rev 2.0 – 01/2013)

Patrick GUMUCHIAN
patrick.gumuchian@univ-amu.fr
IUT GEII Marseille

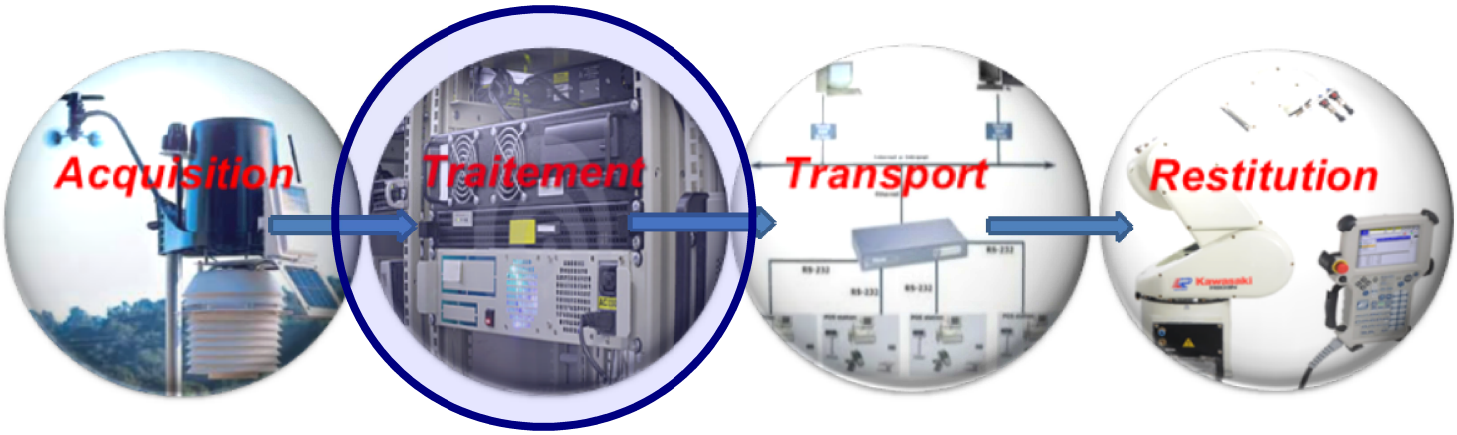
Marc SILANUS
marc.silanus@ac-aix-marseille.fr
http://www.silanus.fr/sin/
Lycée A. Benoit L'Isle sur la Sorgue

SOMMAIRE

- Les circuits logiques
- Mémoires
- Processeurs
- Ordinateur
- Composants Logiques Pragrammables
- Machine de MOORE et de MEALY

INTRODUCTION

Chaîne de traitement de l'information



INTRODUCTION

L'ensemble des équipements de notre vie quotidienne utilise des circuits programmables :

- **Electroménager (TV, machine à laver, four …), ordinateur, baladeur, lecteur video, automobile, Téléphone …**

INTRODUCTION

L'ensemble des équipements de notre vie quotidienne utilise des circuits programmables :

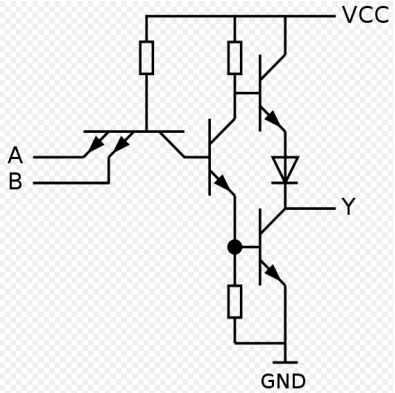
- **2 grandes familles de circuits programmables :**
  - programmation logicielle (µP, µC …)
  - programmation matérielle (CPLD, FPGA...)

LES CIRCUITS LOGIQUES

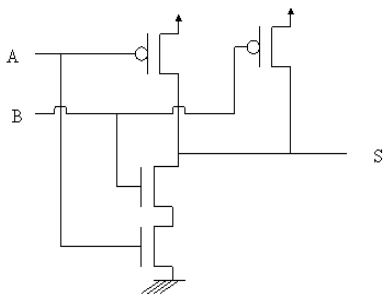
CIRCUITS LOGIQUES

2 grandes technologies :

- ☞ **TTL** : "Transistor-Transistor Logic" => transistors bipolaires
- ☞ **CMOS** : Complementary Metal Oxyde Semiconductor => transistors MOS



TTL

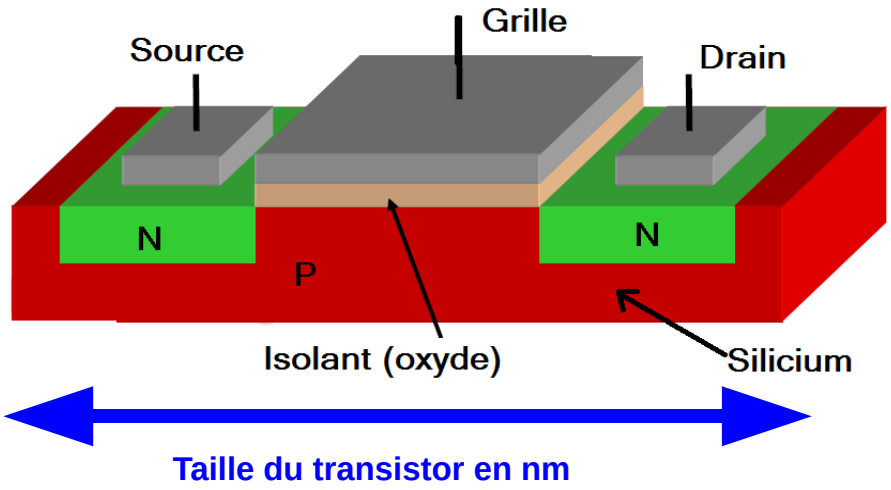


CMOS

CIRCUITS LOGIQUES

Technologie la plus utilisée aujourd'hui : CMOS (Complementary Metal Oxyde Semiconductor)

- **2 transistors MOS Complémentaires : Canal N et Canal P**



TRANSISTOR MOS CANAL N

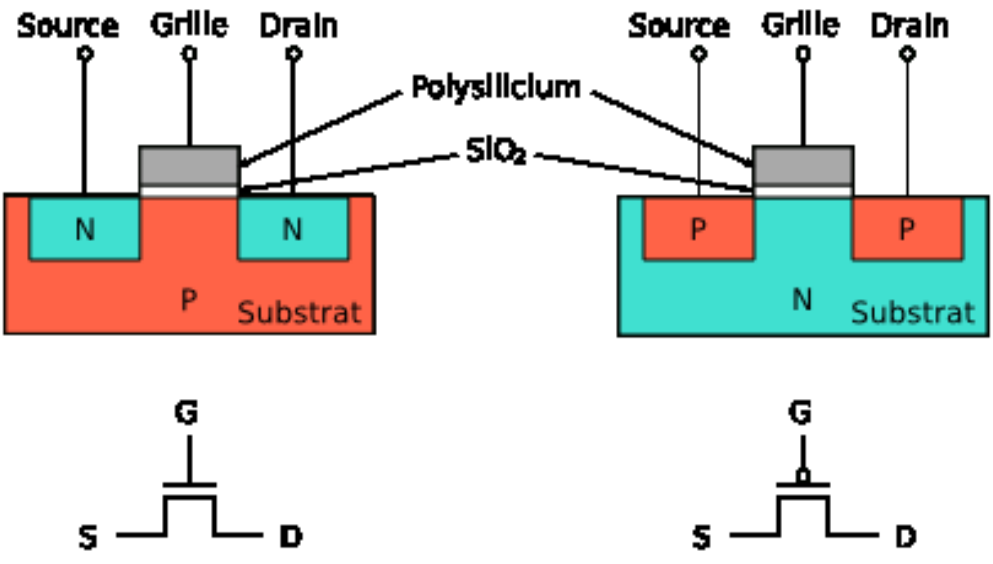
Taille de gravure du Core i7 : 32 nm

Taille du transistor en nm

CIRCUITS LOGIQUES

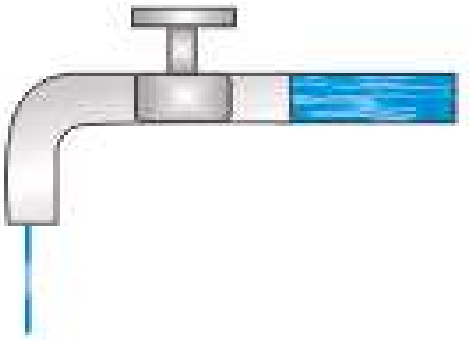
Technologie la plus utilisée aujourd'hui : CMOS (Complementary Metal Oxyde Semiconductor)

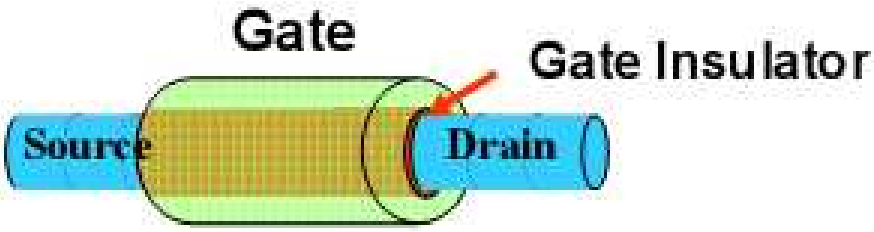
- **2 transistors MOS Complémentaires : Canal N et Canal P**



CIRCUITS LOGIQUES

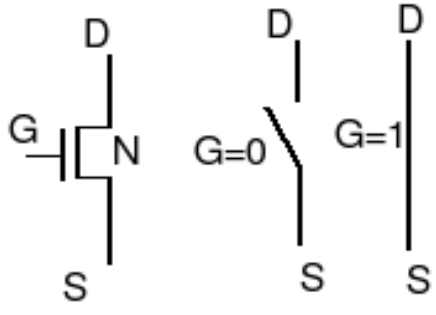
Technologie la plus utilisée aujourd'hui : CMOS (Complementary Metal Oxyde Semiconductor)



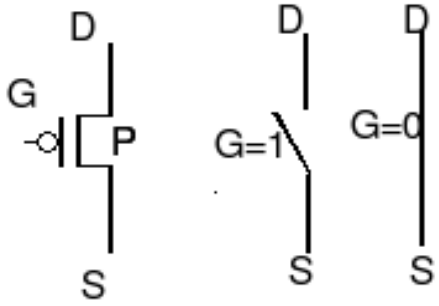


CIRCUITS LOGIQUES

Technologie la plus utilisée aujourd'hui : CMOS (Complementary Metal Oxyde Semiconductor)



Transitors nMOS

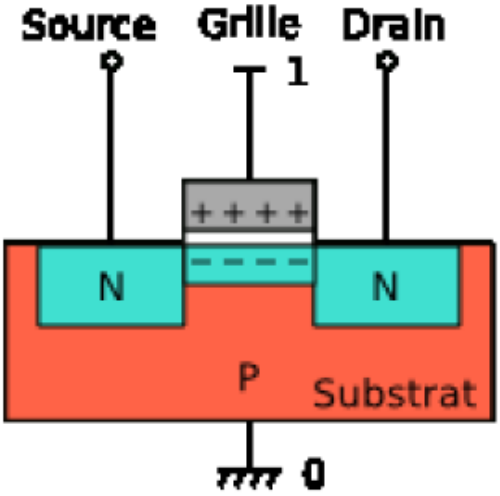


Transitors pMOS

CIRCUITS LOGIQUES

Pourquoi faire toujours plus petit ?

- **Réduire la taille de la puce (téléphone plus petit)**
- **Problème de consommation électrique**
- **Augmenter la fréquence de travail des circuits**



Condensateurs parasites :

- $P = 0{,}5 \cdot C \cdot U^2 \cdot F$ pendant les commutations

Solutions :

- Diminuer C en diminuant la taille
- Diminuer U (technologie des semiconducteurs)

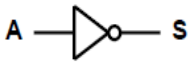
LA LOGIQUE COMBINATOIRE

LOGIQUE COMBINATOIRE

Les circuits logiques de base :

NON

A	$S = \overline{A}$
0	1
1	0



ET

A B	S = A·B
0 0	0
0 1	0
1 0	0
1 1	1

OU

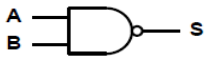
A B	S = A+B
0 0	0
0 1	1
1 0	1
1 1	1

XOR

A B	S = A⊕B
0 0	0
0 1	1
1 0	1
1 1	0

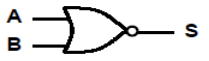
NAND

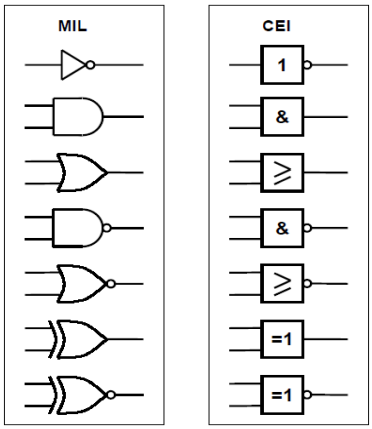
A B	$S = \overline{A \cdot B}$
0 0	1
0 1	1
1 0	1
1 1	0



NOR

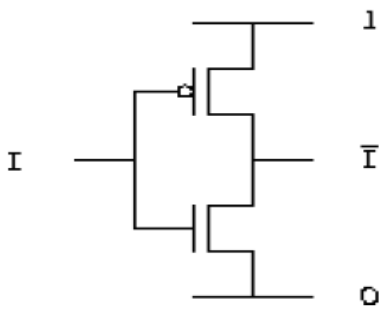
A B	$S = \overline{A+B}$
0 0	1
0 1	0
1 0	0
1 1	0

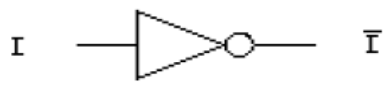


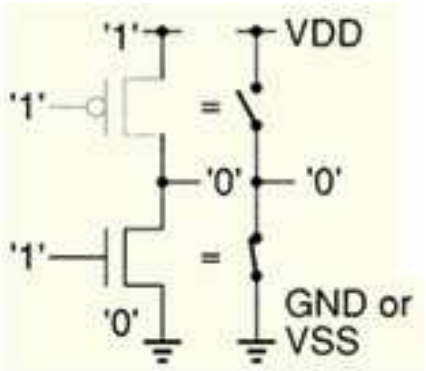


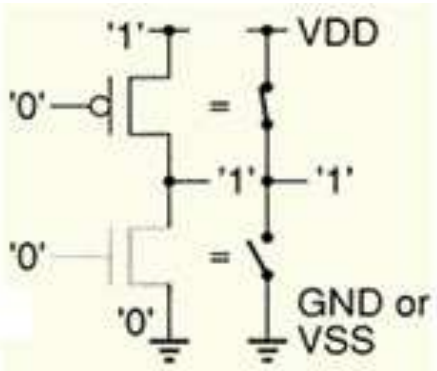
LOGIQUE COMBINATOIRE

Les circuits logiques de base :



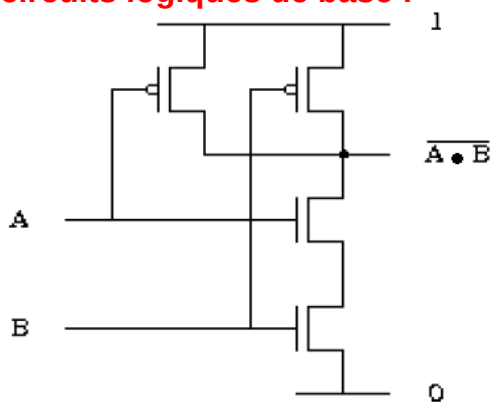


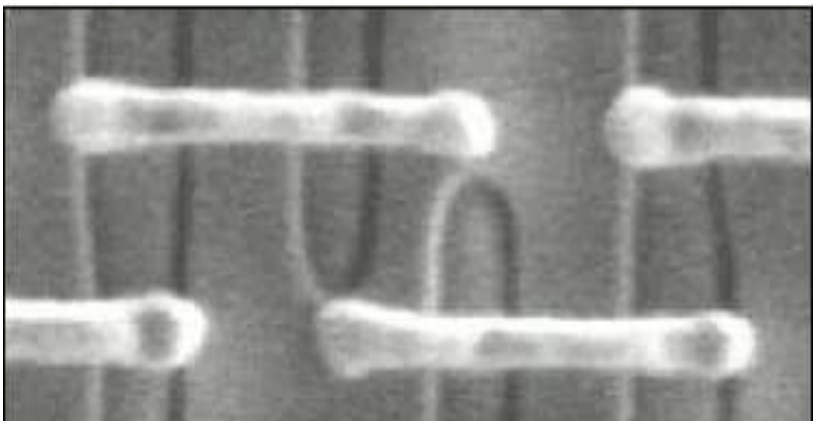


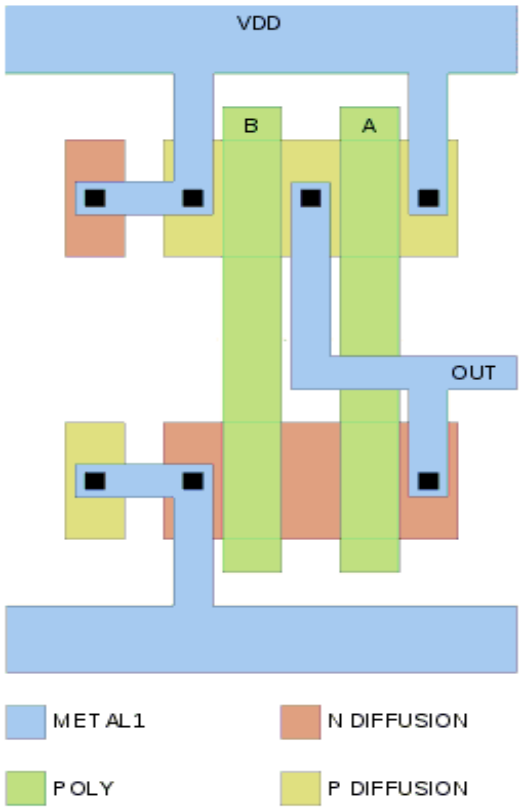


LOGIQUE COMBINATOIRE

Les circuits logiques de base :





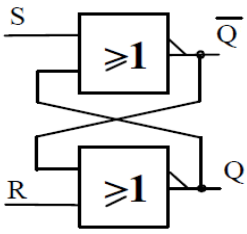


LA LOGIQUE SEQUENTIELLE

LOGIQUE SEQUENTIELLE

- La différence avec la logique combinatoire est que le système est rebouclé.
- Les sorties dépendent des entrées mais aussi de l'état précédent des sorties.
- 2 types de bascules : Synchrone ou Asynchrone

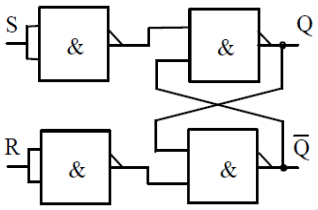
Bascule RS : Arrêt prioritaire



S	R	Q	$\overline{Q}$
0	0	Qp	$\overline{Qp}$
0	1	0	1
1	0	1	0
1	1	0	0

Etat mémoire

Bascule RS : Marche prioritaire

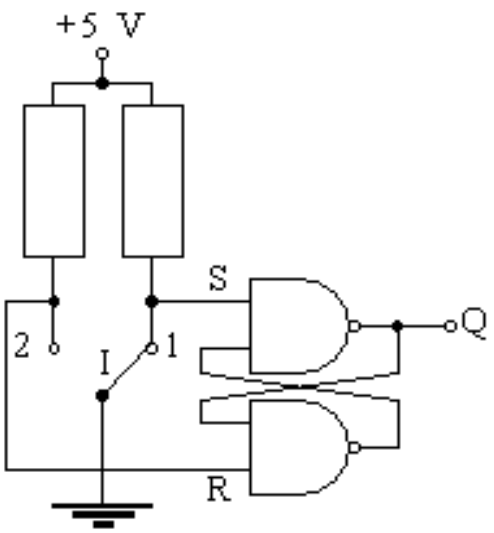


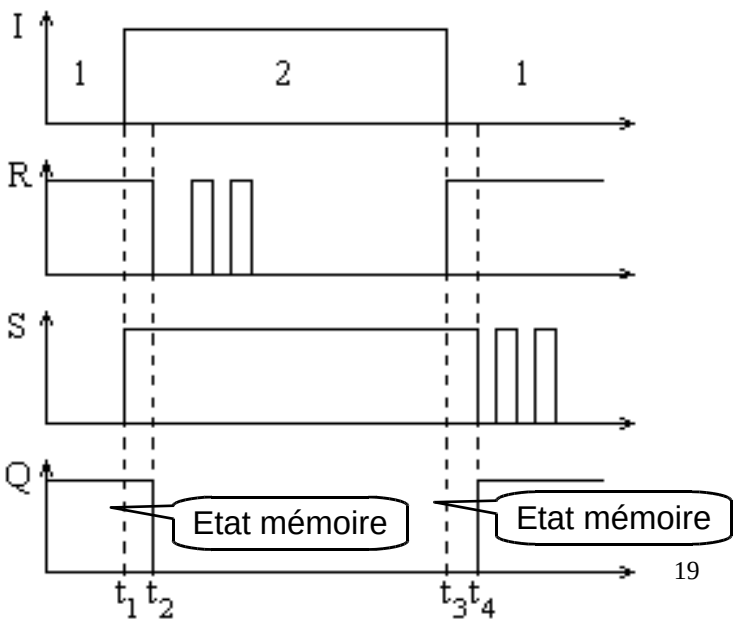
S	R	Q	Q
0	0	Qp	$\overline{Qp}$
0	1	0	1
1	0	1	0
1	1	1	1

Etat mémoire

LOGIQUE SEQUENTIELLE

- La différence avec la logique combinatoire est que le système est rebouclé.
- Les sorties dépendent des entrées mais aussi de l'état précédent des sorties.
- 2 types de bascules : Synchrone ou Asynchrone

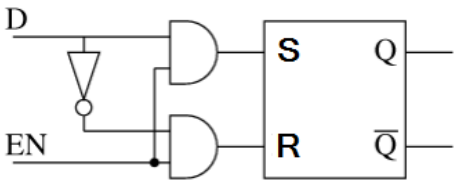




LOGIQUE SEQUENTIELLE

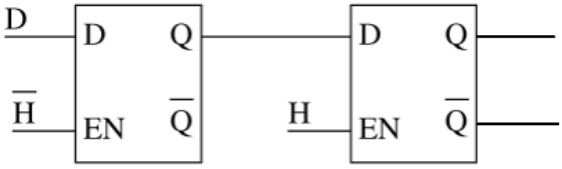
- **Bascule D**

Changement sur état



D	H	Q
0	1	0
1	1	1
X	0	Q_{n-1}

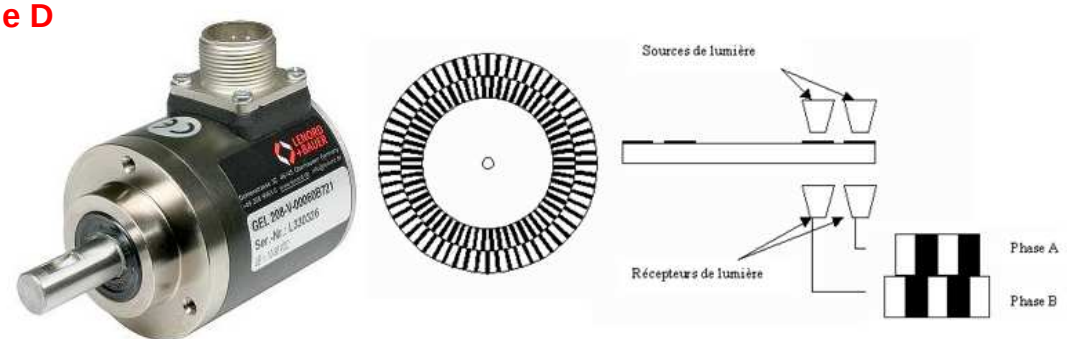
Changement sur front

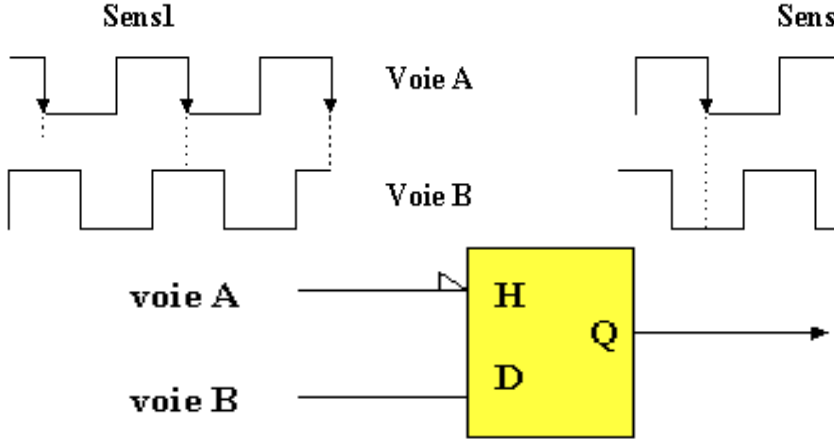


D	H	Q
0	↑	0
1	↑	1
X	0/1	Q_{n-1}

LOGIQUE SEQUENTIELLE

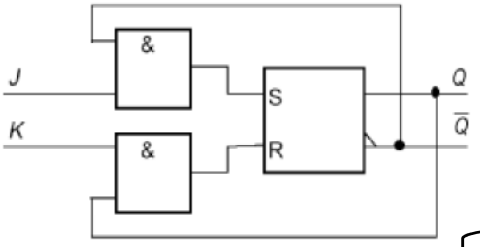
- **Bascule D**

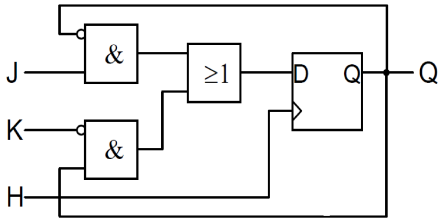




LOGIQUE SEQUENTIELLE

☞ **Bascule JK : Synchrone ou Asynchrone**





Etat mémoire

J	K	Q_{n-1}	Q_n
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

ENTREES				SORTIES	
CLEAR	CLOCK	J	K	Q	$\overline{Q}$
0	X	X	X	0	1
1	↓	0	0	Q0	$\overline{Q}$0
1	↓	1	0	1	0
1	↓	0	1	0	1
1	↓	1	1	TOGGLE	
1	1	X	X	Q0	$\overline{Q}$0
1	0	X	X	Q0	$\overline{Q}$0

ALGEBRE DE BOOLE

Commutativité:
$A \cdot B = B \cdot A$
$A + B = B + A$

Idempotence:
$A \cdot A = A$
$A + A = A$

Constantes:
$A \cdot 0 = 0$
$A \cdot 1 = A$
$A + 0 = A$
$A + 1 = 1$

Complémentation:
$A \cdot \overline{A} = 0$
$A + \overline{A} = 1$

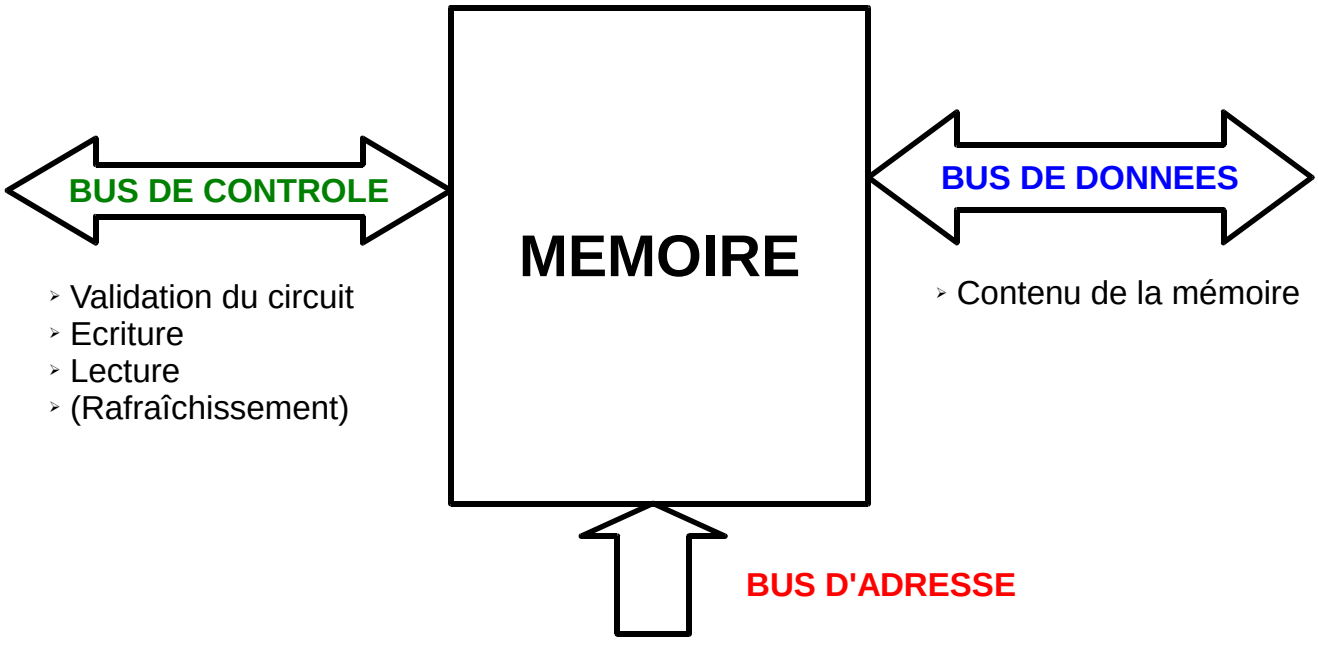
Distributivité:
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$
$A + (B \cdot C) = (A + B) \cdot (A + C)$

Associativité:
$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$
$A + (B + C) = (A + B) + C = A + B + C$

De Morgan:
$\overline{A \cdot B} = \overline{A} + \overline{B}$
$\overline{A + B} = \overline{A} \cdot \overline{B}$

LES MEMOIRES

LA MEMOIRE

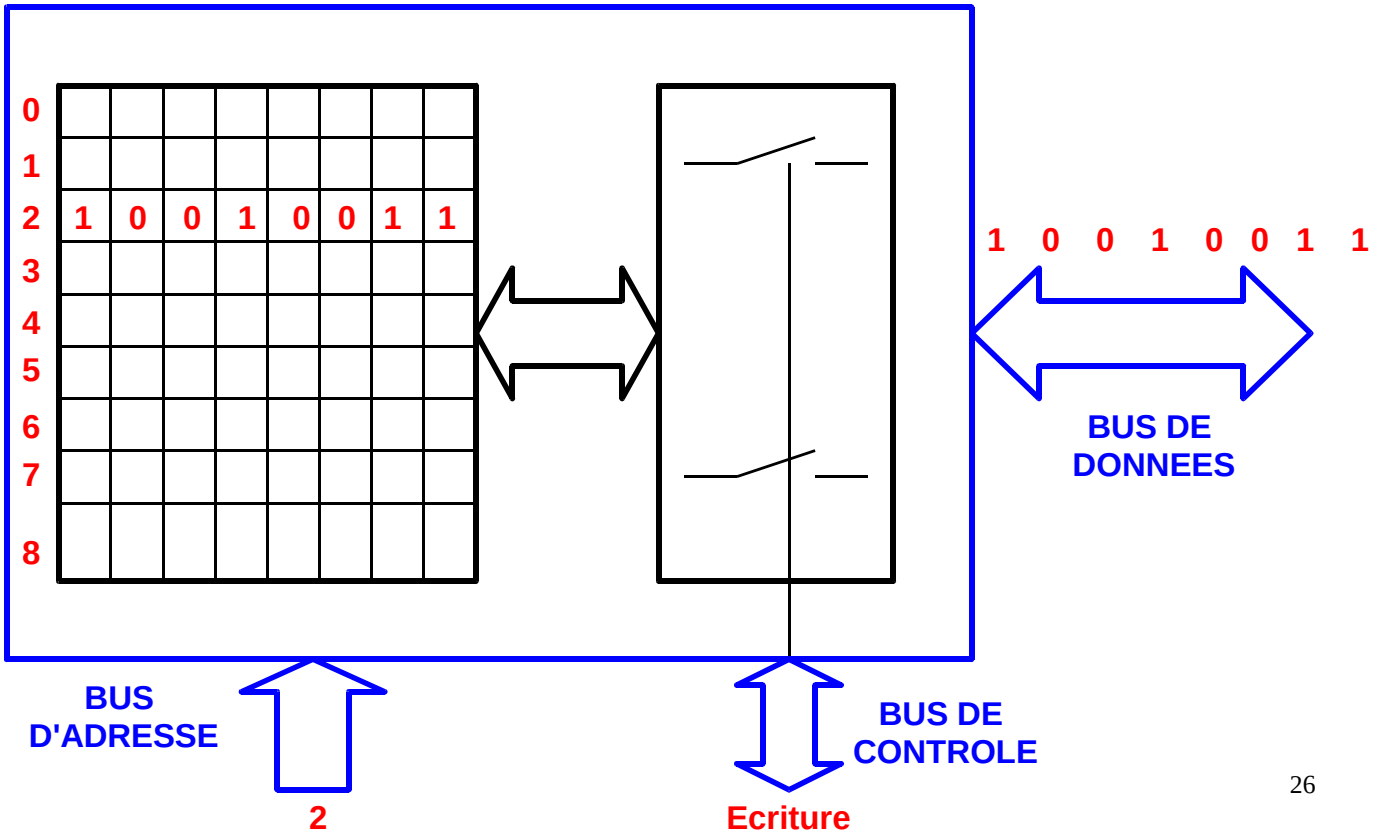


- Validation du circuit
- Ecriture
- Lecture
- (Rafraîchissement)

- Contenu de la mémoire

- Sélection d'un emplacement dans la mémoire

LA MEMOIRE



LA MEMOIRE

- C'est un composant qui est **constitué de cellules élémentaires** permettant de retenir ou de restituer une information binaire (0 ou 1).
- Les cellules élémentaires sont constituées de **transistors**.
- **Le bus d'Adresses** permet d'accéder aux cellules élémentaires.
- **Le bus de données** permet d'accéder au contenu de la cellule élémentaire.
- **Le bus de contrôle** permet de valider le fonctionnement du boîtier et d'indiquer s'il s'agit d'une écriture ou d'une lecture.

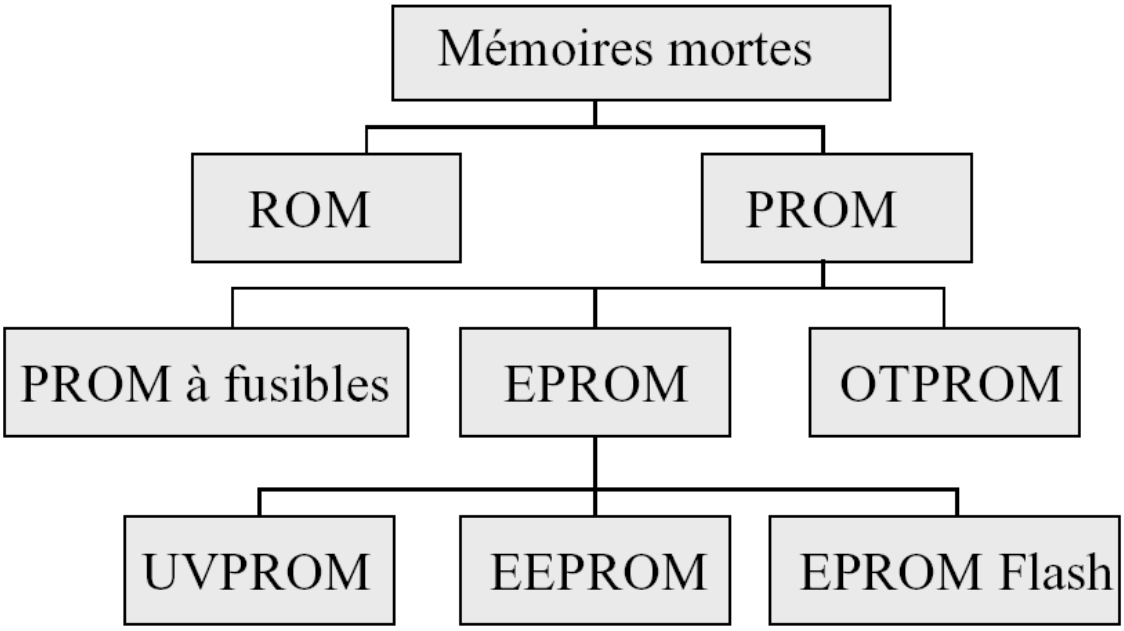
LA MEMOIRE

Les differents types :

- **RAM** dynamique ou statique (Random Access Memory)
- **ROM** (Read Only Memory)
- **PROM** ou **OTP** (Programmable Read Only Memory) ou (Once Time Programmable)
- **EPROM** (Erasable Programmable Read Only Memory)
- **EEPROM** (Electricaly Erasable Programmable Read Only Memory)
- **FLASH**

LA MEMOIRE

Les differents types de mémoires mortes :



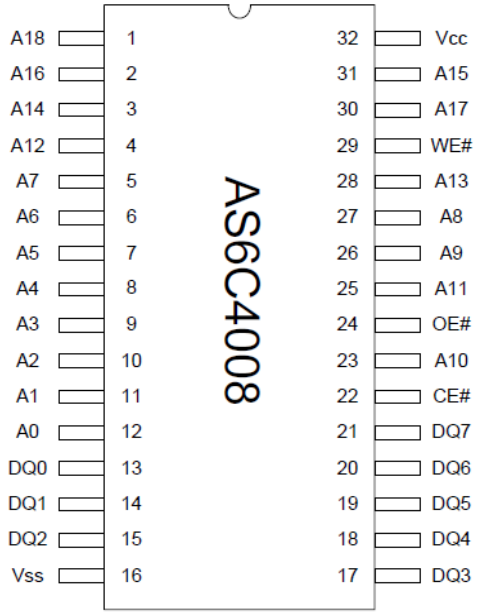
LA MEMOIRE : caractéristiques

- **Capacité** : c'est le nombre de bits (ou d'octets) que contient le circuit (128 bits, 1kbits, 1Mbits, etc)
- **Bus de données** soit série (avec protocole de communication : SPI, I2C, etc) soit parallèle
- **Temps de lecture** (qq ns)
- **Temps d'écriture** (de qq ns à qq ms)
- **Temps d'effacement** (de qq ns à qq ms)
- **Nombre de cycles d'écriture** (effacement et écriture)
- **Tension de programmation** (peut aller jusqu'à 21 V)
- **Durée de conservation des données** (Data retention) en année (on arrive à des durées > 200 ans pour les ROM)

LA MEMOIRE : exemple 1

RAM STATIQUE : 512 Ko

PIN CONFIGURATION



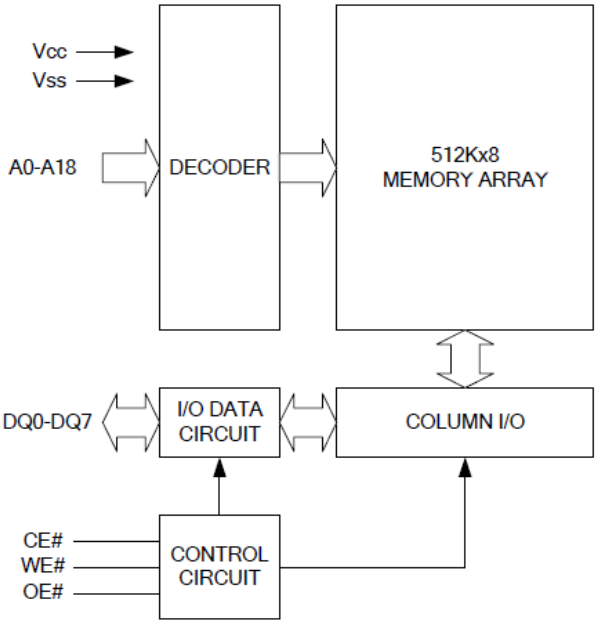
TRUTH TABLE

MODE	CE#	OE#	WE#	I/O OPERATION
Standby	H	X	X	High-Z
Output Disable	L	H	H	High-Z
Read	L	L	H	DOUT
Write	L	X	L	DIN

Note: H = V_{IH}, L = V_{IL}, X = Don't care.

LA MEMOIRE : exemple 1

FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION**

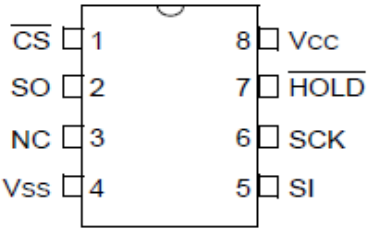
SYMBOL	DESCRIPTION
A0 - A18	Address Inputs
DQ0 – DQ7	Data Inputs/Outputs
CE#	Chip Enable Inputs
WE#	Write Enable Input
OE#	Output Enable Input
Vcc	Power Supply
Vss	Ground
NC	No Connection

$$C = \frac{2^{nb\ bits\ adresses}}{1024}$$

LA MEMOIRE : exemple 2

RAM STATIQUE : Bus SPI (Serial Peripheral Interface)

PDIP/SOIC/TSSOP
(P, SN, ST)



Pin Function Table

Name	Function
$\overline{CS}$	Chip Select Input
SO	Serial Data Output
Vss	Ground
SI	Serial Data Input
SCK	Serial Clock Input
$\overline{HOLD}$	Hold Input
Vcc	Supply Voltage

LA MEMOIRE : exemple 2

TABLE 2-1: INSTRUCTION SET

Instruction Name	Instruction Format	Description
READ	0000 0011	Read data from memory array beginning at selected address
WRITE	0000 0010	Write data to memory array beginning at selected address
RDSR	0000 0101	Read STATUS register
WRSR	0000 0001	Write STATUS register

FIGURE 2-1: BYTE READ SEQUENCE

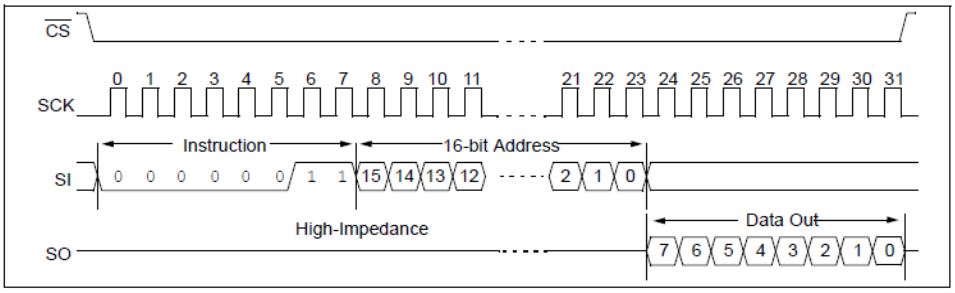
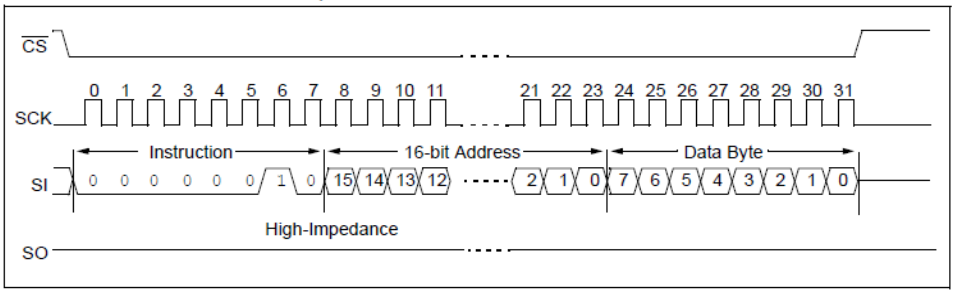
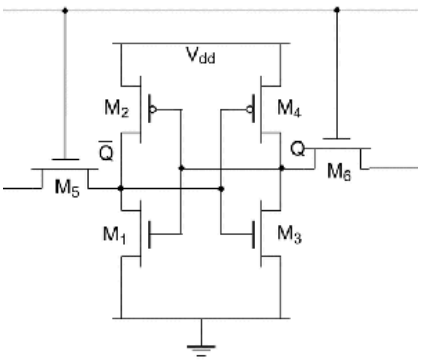


FIGURE 2-2: BYTE WRITE SEQUENCE



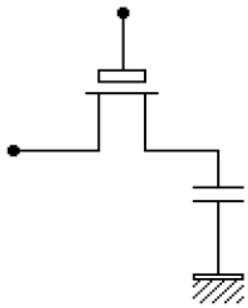
LA MEMOIRE : Technologie

RAM STATIQUE
6 transistors
Pour 1 bit



Plus rapide mais plus de transistors

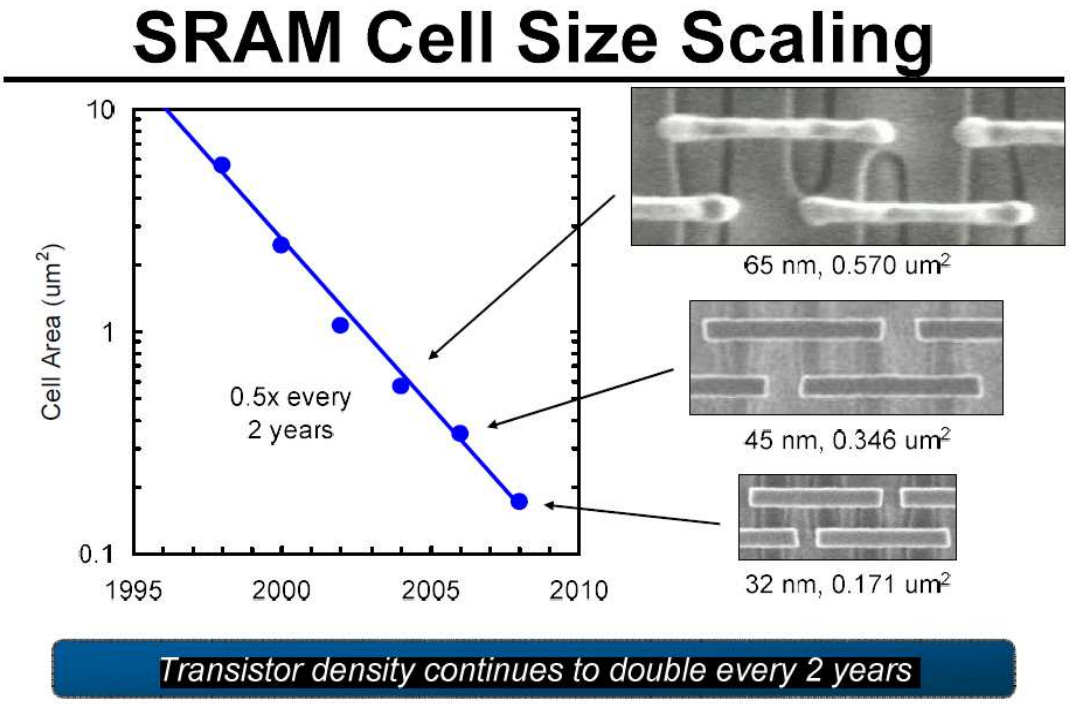
RAM dynamique
2 transistors
Pour 1 bit



Transistor MOS "dégénéré"

Besoin de rafraîchissement (recharger le condensateur)
Moins de transistors

LA MEMOIRE : Technologie

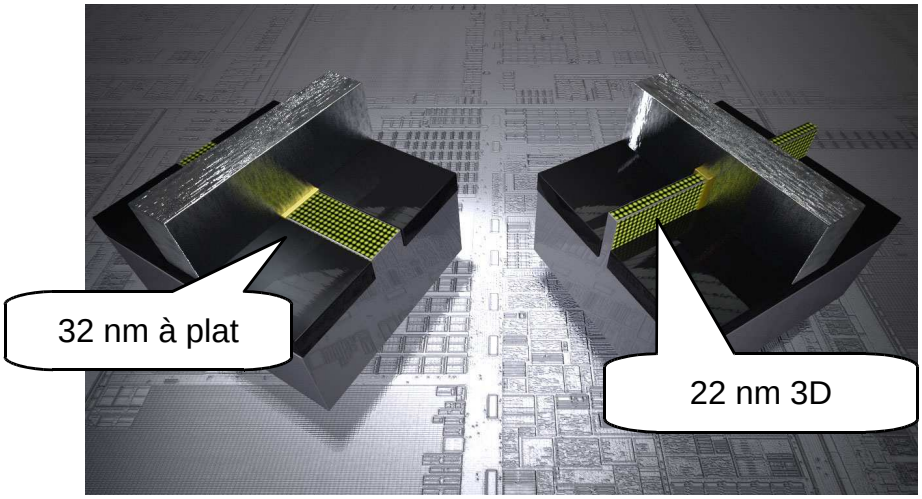


DOCUMENT INTEL

LA MEMOIRE : Technologie

2011 : 22nm en gravure 3D

Transistor Tri-Gate



DOCUMENT INTEL

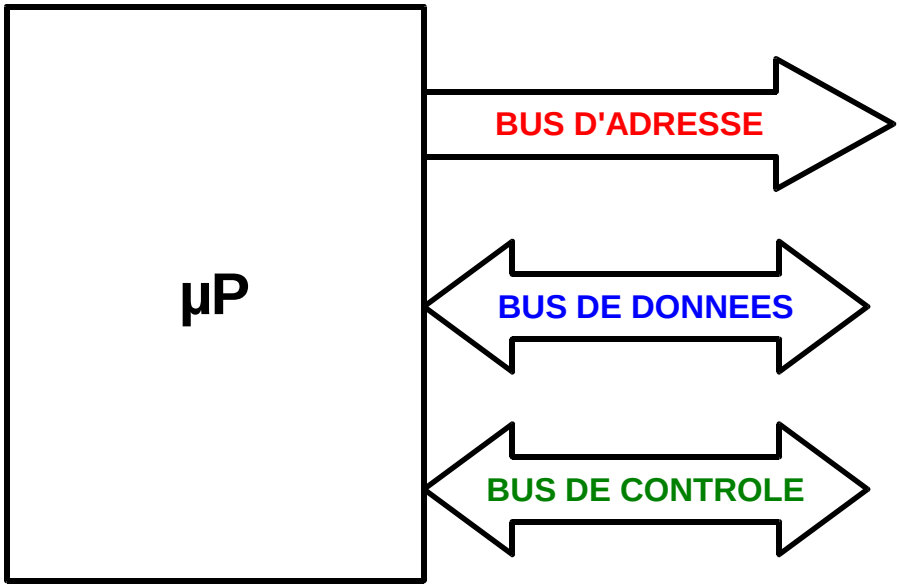
LE MICROPROCESSEUR

MICROPROCESSEUR

Date	Nom	Nb transistors	Gravure µm	Fréquence horloge	Bus adresse Bus données	MIPS
1971	4004	2300		108 kHz	4 bits / 4 bits	0,06
1974	8080	6000	6	2 MHz	8 bits / 8 bits	?
1979	8088	29000	3	5 MHz	16 bits / 8 bits	0,33
1982	80286	134 000	1,5	6 à 20 MHz	16 bits / 16 bits	1
1985	80386	275 000	1,5	16 à 40 MHz	32 bits / 32 bits	5
1989	80486	1 200 000	1	16 à 100 MHz	32 bits / 32 bits	20
1993	Pentium	3 100 000	0,8 à 0,28	60 à 233 MHz	32 bits / 64 bits	100
2000	Pentium 4	42 000 000	0,18 à 0,065	1,3 à 3,8 GHz	32 bits / 64 bits	1 700

MICROPROCESSEUR

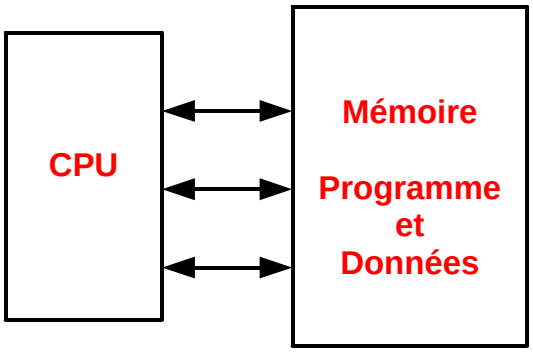
Vue de l'extérieur



MICROPROCESSEUR

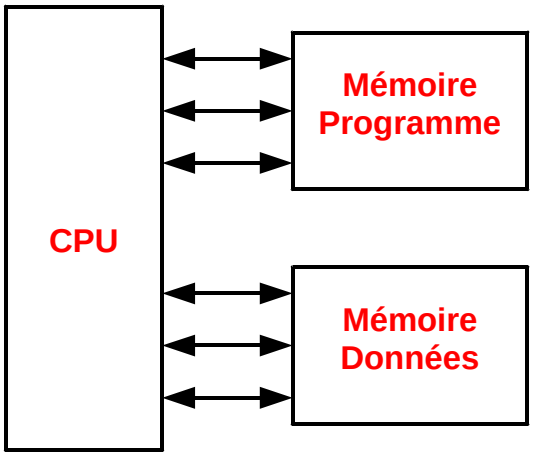
Architectures

Architecture : Von Neumann



Il faut 2 instructions différentes pour accéder à la mémoire programme et à la mémoire données

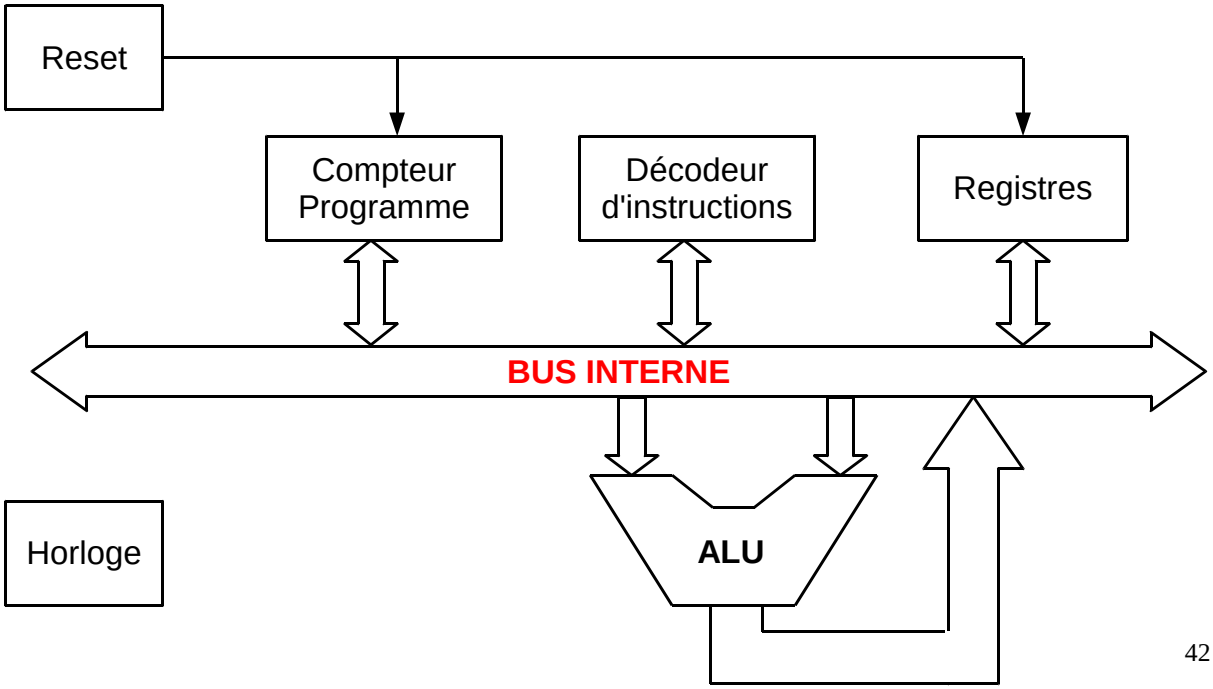
Architecture : Harvard



On accède à la mémoire programme et à la mémoire données avec 1 seule instruction

MICROPROCESSEUR

Vue de l'intérieur
Structure minimale



MICROPROCESSEUR

RESET : réinitialise le compteur programme et les registres du microprocesseur.

HORLOGE : fixe la vitesse de travail. Elle peut être configurée par des registres.

COMPTEUR PROGRAMME :
place sur le bus d'adresse, la prochaine adresse à lire dans la mémoire programme.

DECODEUR D'INSTRUCTION :
permet de lire la donnée présente sur le bus de donnée de la mémoire programme et de décoder l'instruction à réaliser.

UAL : unité qui permet au microprocesseur de faire tous les calculs (opérations sur les bits, décalage à droite ou à gauche, addition, multiplication, complémentation, ET, OU, OU exclusif, etc).

REGISTRES : ce sont des mémoires.

MICROPROCESSEUR

Architectures

On trouve des architectures diverses et variées pour exécuter les instructions:

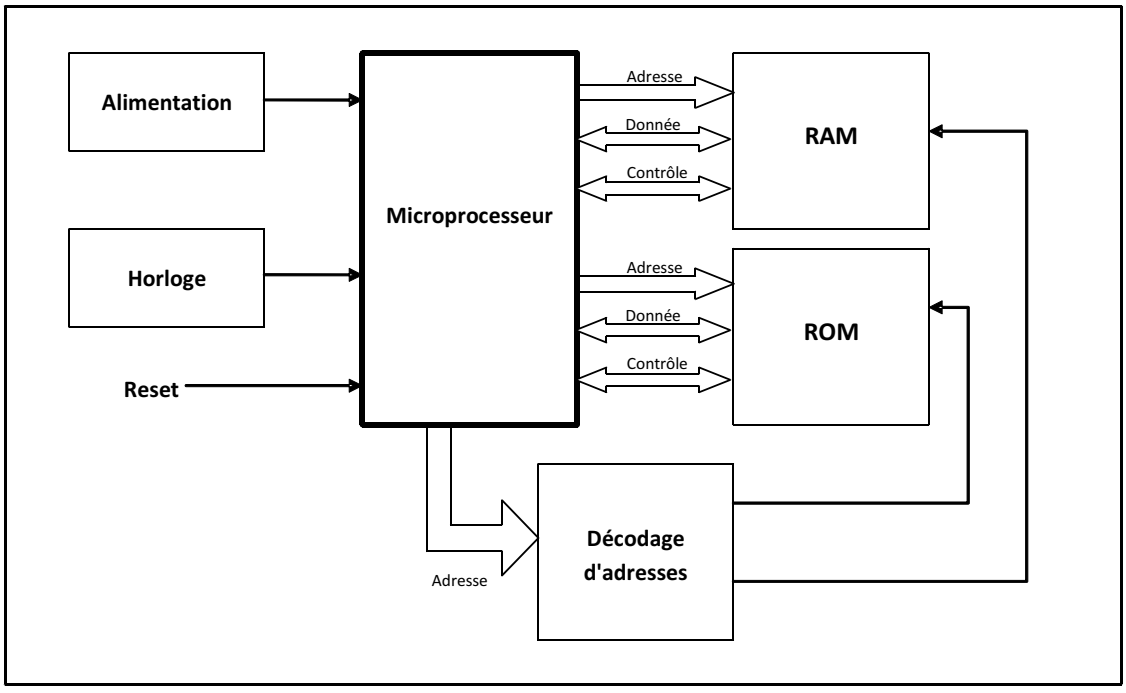
- **Instruction exécutée en "Pipeline" dans l'ordre ou le désordre**
- **Instruction "Superscalaire" : plusieurs UAL en //**

Thread = partie de programme indépendante du reste du programme

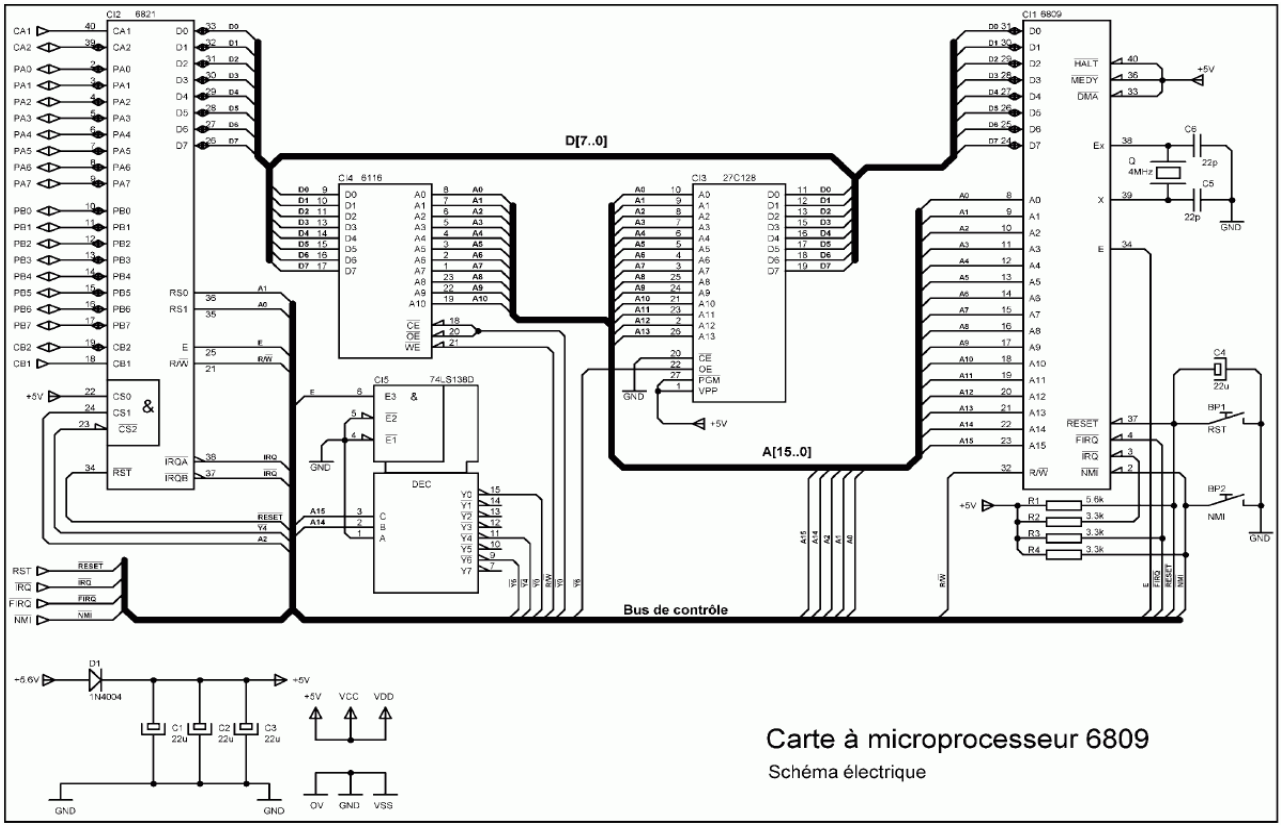
- **"Hyperthreading" : 2 processeurs virtuels (2 thread) avec un processeur réel**
- **"Multithreading" : exécution du plusieurs thread en même temps.**

MICROPROCESSEUR

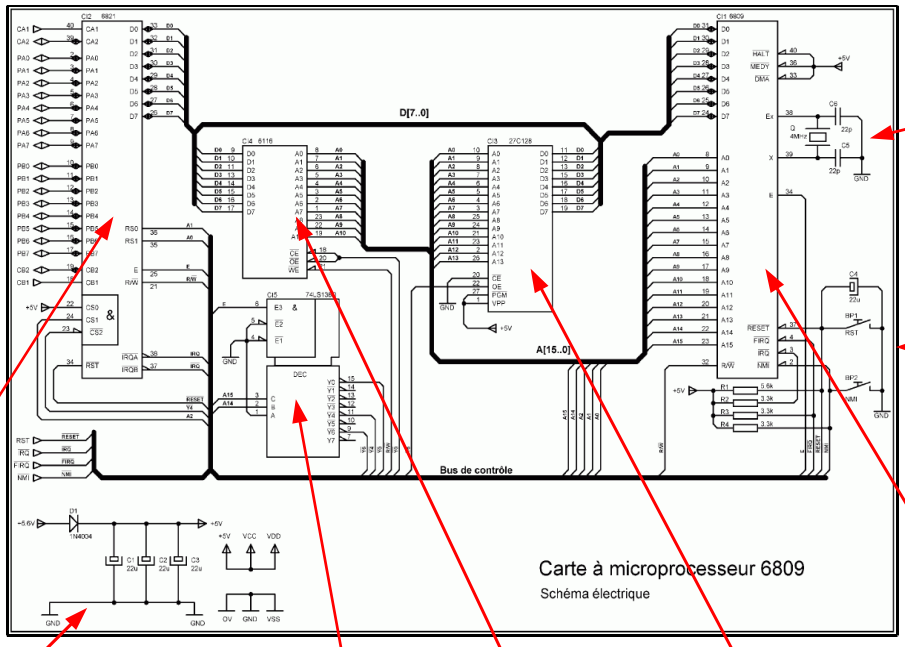
Système minimum



MICROPROCESSEUR



MICROPROCESSEUR



Horloge

Reset

Périphérique

Microprocesseur

Condensateurs de découplage

Décodage d'adresse

RAM

ROM

EXERCICE

MICROPROCESSEUR

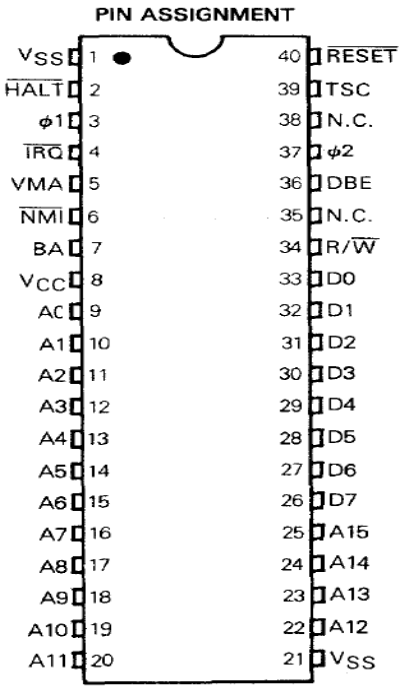
Exercice

Réalisation d'une carte avec un microprocesseur MOTOROLA MC6800.

- Quelle est la capacité d'adressage du microprocesseur ?
- A quelle adresse doit se placer la mémoire ROM et pourquoi ?
- On utilise une mémoire ROM de 16 Ko et une mémoire RAM de 8 Ko. Indiquer les adresses de début et de fin de chaque mémoire. Faire un plan d'adressage mémoire.
- Faire la logique de décodage d'adresse avec des circuits logiques.
- Réaliser un circuit de Reset avec une résistance et un condensateur (à dimensionner).

MICROPROCESSEUR

Exercice



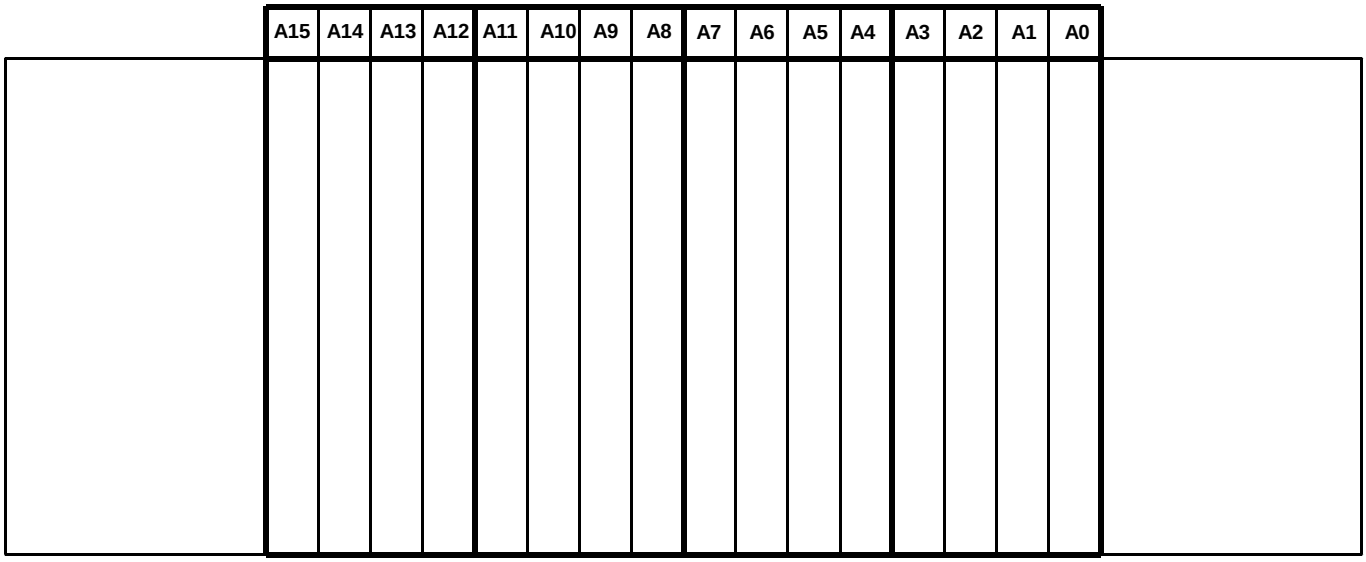
Adresse du vecteur RESET :
$FFFE - $FFFF

ROM : 16 Ko
RAM : 8 Ko

MICROPROCESSEUR

Exercice

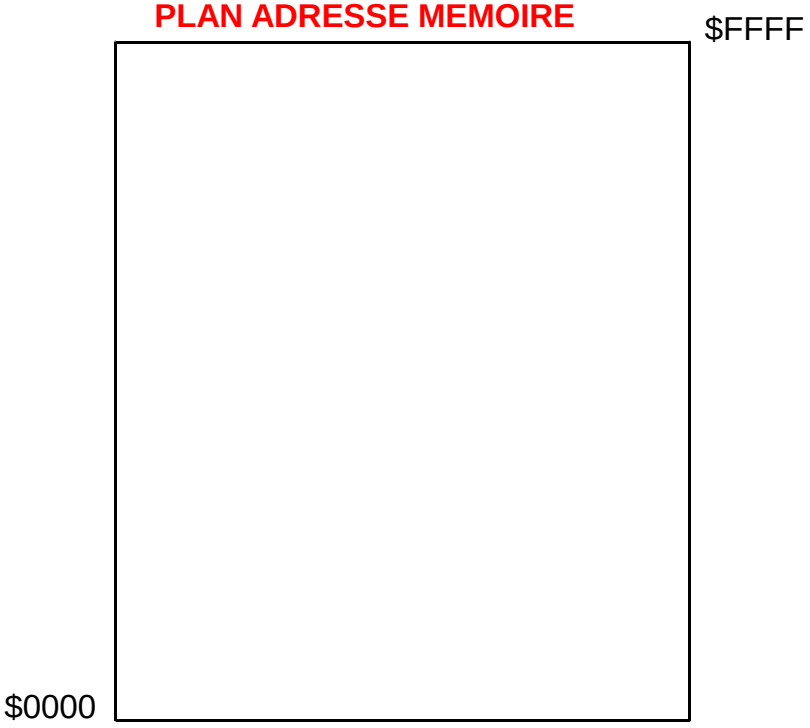
PLAN ADRESSE MEMOIRE



MICROPROCESSEUR

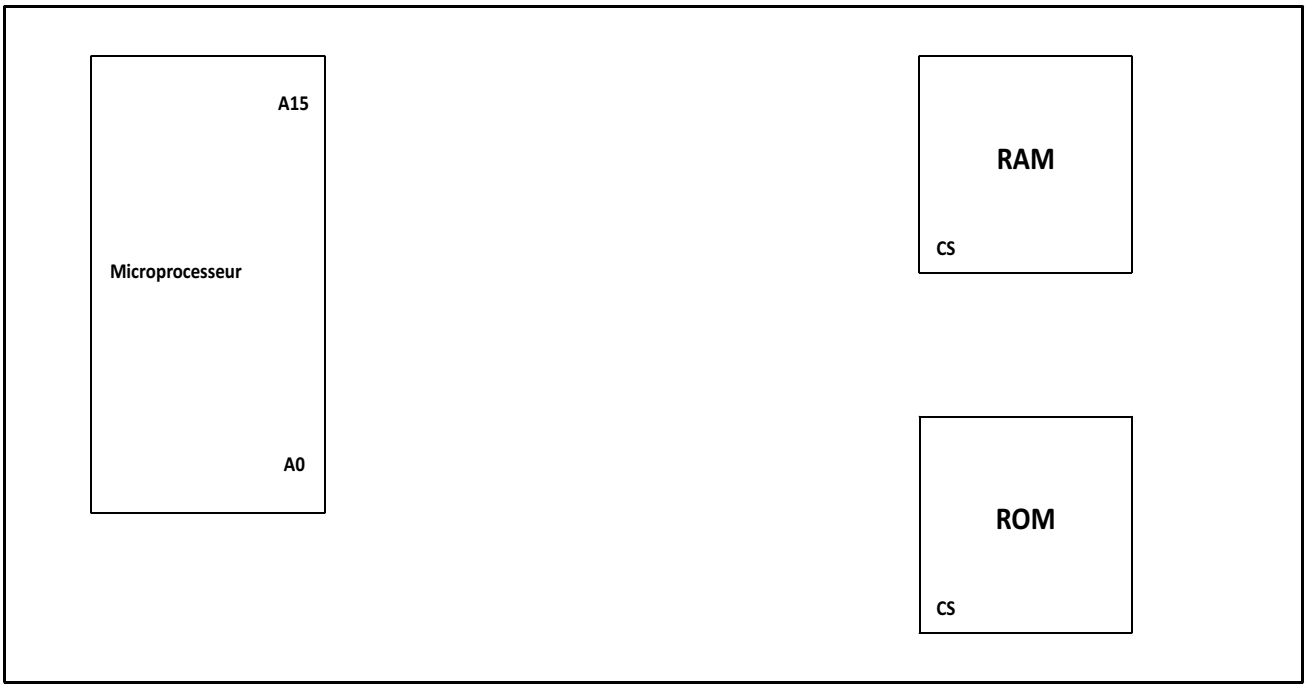
Exercice

PLAN ADRESSE MEMOIRE



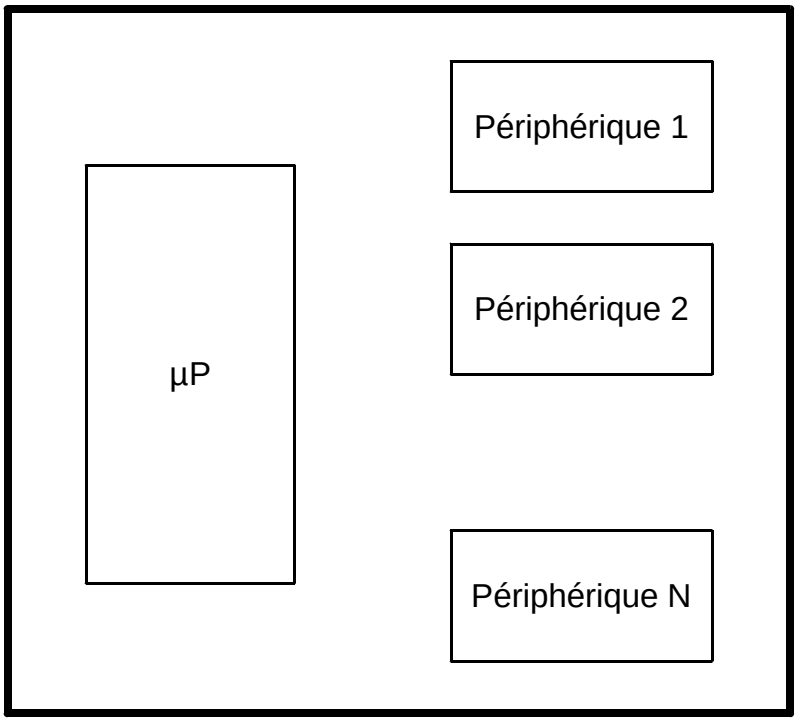
MICROPROCESSEUR

Exercice



LE MICROCONTROLEUR

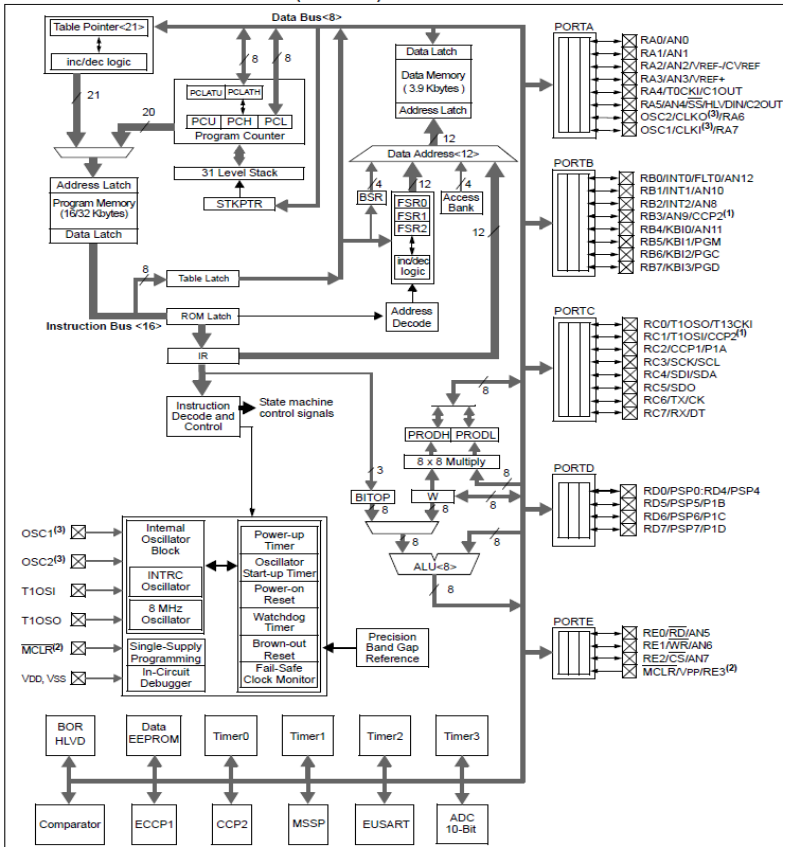
MICROCONTROLEUR



C'est un microprocesseur avec des périphériques intégrés dans un seul circuit

MICROCONTROLEUR

FIGURE 1-2: PIC18F4420/4520 (40/44-PIN) BLOCK DIAGRAM



Exercice avec un PIC 18F4520

Repérer les bornes externes du circuit

Identifier le microprocesseur

Identifier les périphériques

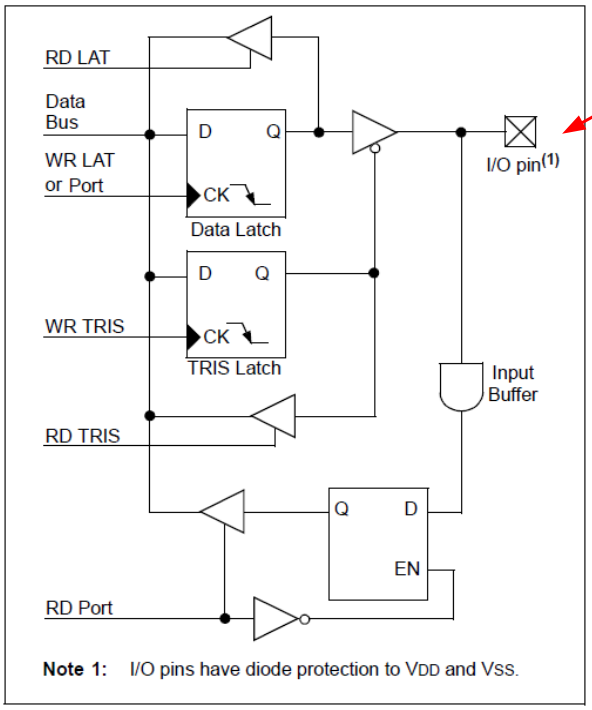
MICROCONTROLEUR

Quelques périphériques

- Port d'entrée / sortie
- Timer
- PWM ou MLI (Modulation à Largeur d'Impulsion)
- Conversion Analogique Numérique (ADC)
- Conversion Numérique Analogique (DAC)
- Liaison série (RS232, SPI, I2C...)
- Bus CAN (Automobile)
- Ethernet

MICROCONTROLEUR

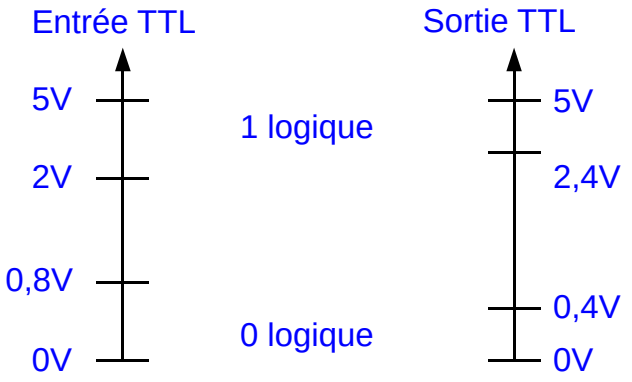
Port



Borne externe au circuit

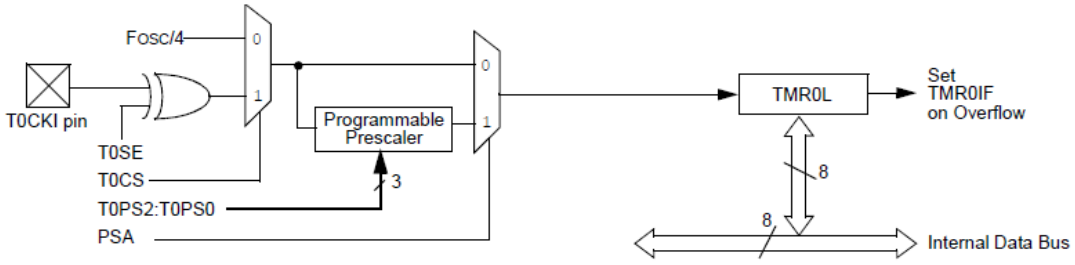
- C'est l'interface qui traduit les niveaux logiques en tensions et réciproquement.
- C'est la communication entre l'intérieur et l'extérieur du circuit.

Norme TTL



MICROCONTROLEUR

Timer



- C'est un compteur de front qui fait passer un bit à 1 lorsqu'il y a un dépassement de capacité.
- La source des fronts peut interne (horloge du micro) ou externe (signal appliqué sur une patte du circuit)

MICROCONTROLEUR

Convertisseur Analogique Numérique

Extérieur du µC

Entrée Analogique Vin

ADC

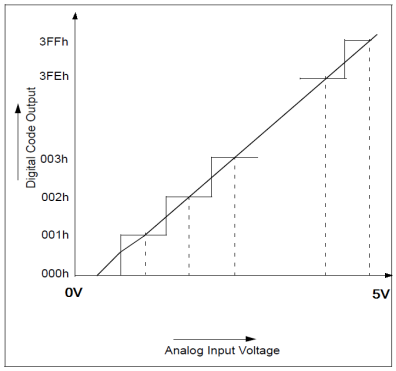
Intérieur du µC

Sortie Numérique N

$$N = \frac{Vin}{Vref}\left(2^n - 1\right)$$

- Vref : tension de référence en V
- n : résolution du convertisseur en bits.

Fonction de transfert



MICROCONTROLEUR

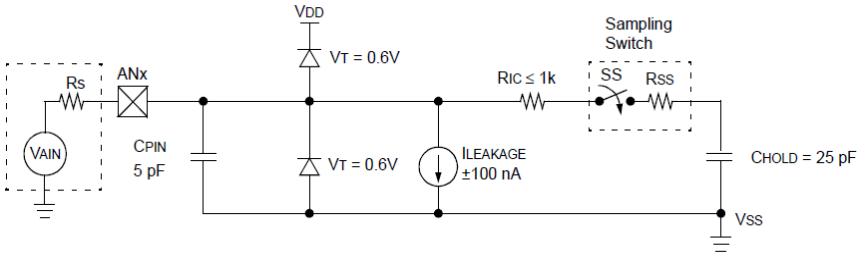
Convertisseur Analogique Numérique

Les caractéristiques :

- Résolution : nombre de bits pour le résultat numérique
- LSB ou quantum : c'est la différence de tension pour faire varier le bit de poids fable

$$LSB = \frac{Vref}{2^n - 1}$$

- Temps de conversion
- Erreur d'offset
- Erreur de linéarité
- Erreur de gain ou pleine échelle



MICROCONTROLEUR

Convertisseur Analogique Numérique

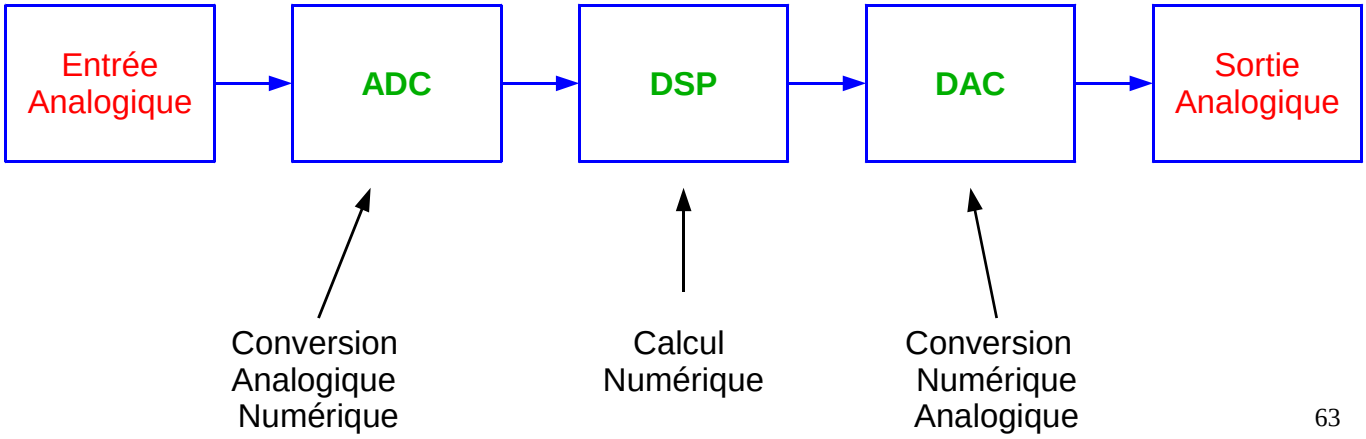
Quelques remarques :

- Un convertisseur 8 bits sera moins précis qu'un convertisseur 10 bits.
- La conversion dure un certain temps (Acquisition + Conversion).
- Problème de l'acquisition du signal : filtrage pour éliminer les parasites et règle de Shanon (la fréquence d'échantillonnage doit être 2 fois plus grande que la fréquence du signal).
- Il existe plusieurs types de convertisseurs : simple rampe, double rampes, approximation successive, flash, Sigma Delta...
- Celui utilisé dans les µC est en général un approximation successive car on maîtrise parfaitement le temps de conversion.

DSP

DSP : Digital Signal Processor (Processeur de Signaux Numériques)

- C'est un microcontrôleur optimisé pour le calcul numérique
- Ce qui le caractérise par rapport au µC c'est une instruction particulière appelée **MAC** : "**Multiply and ACcumulate**" qui est réalisée en un cycle instruction. Cette instruction permet de faire une addition et une multiplication en même temps, le résultat est stocké sur 40 bits.



DSP

Où trouve - t - on ces DSP ?

- **Dans les modem**
- **Les baladeurs**
- **La video**
- **Les GPS**
- **Les téléphones, etc....**
- **Partout où l'on traite du signal numérique (c'est le champion du calcul numérique)**

ORDINATEUR

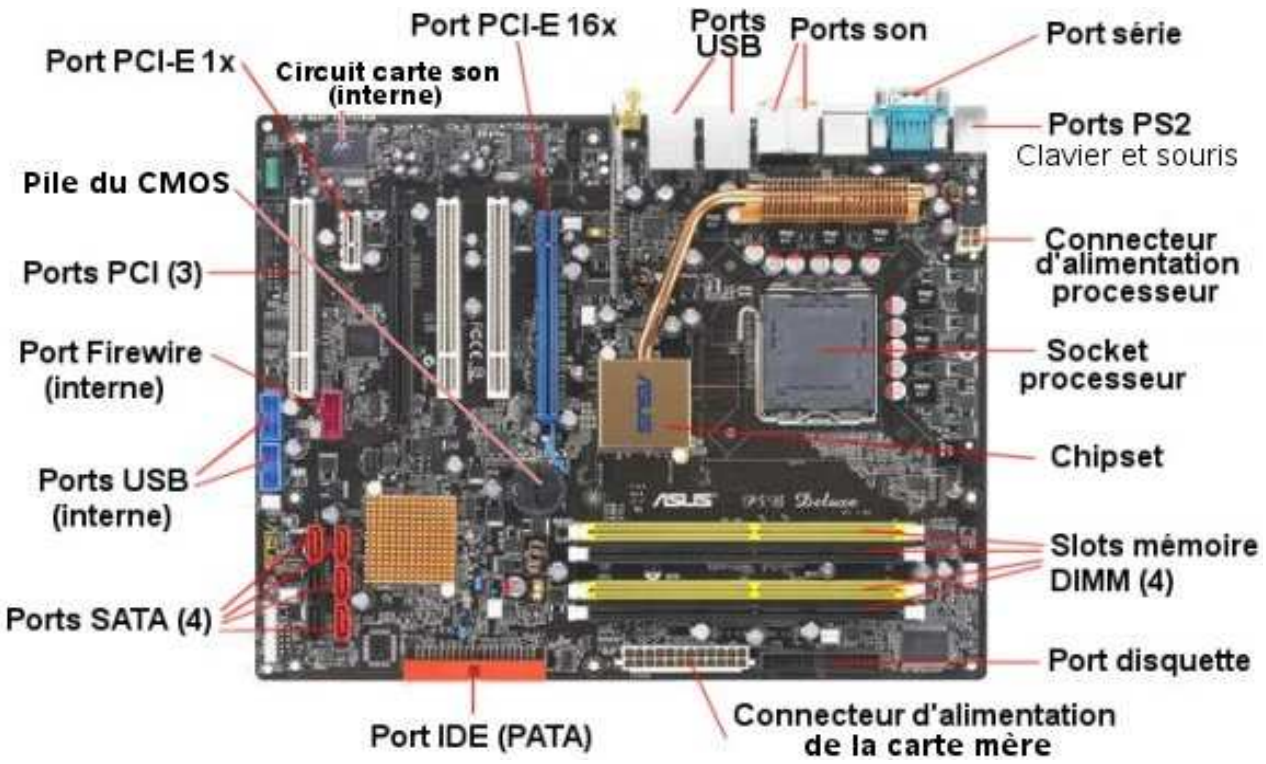
Clavier
Ecran
Souris...

Ce sont des périphériques

Unité centrale

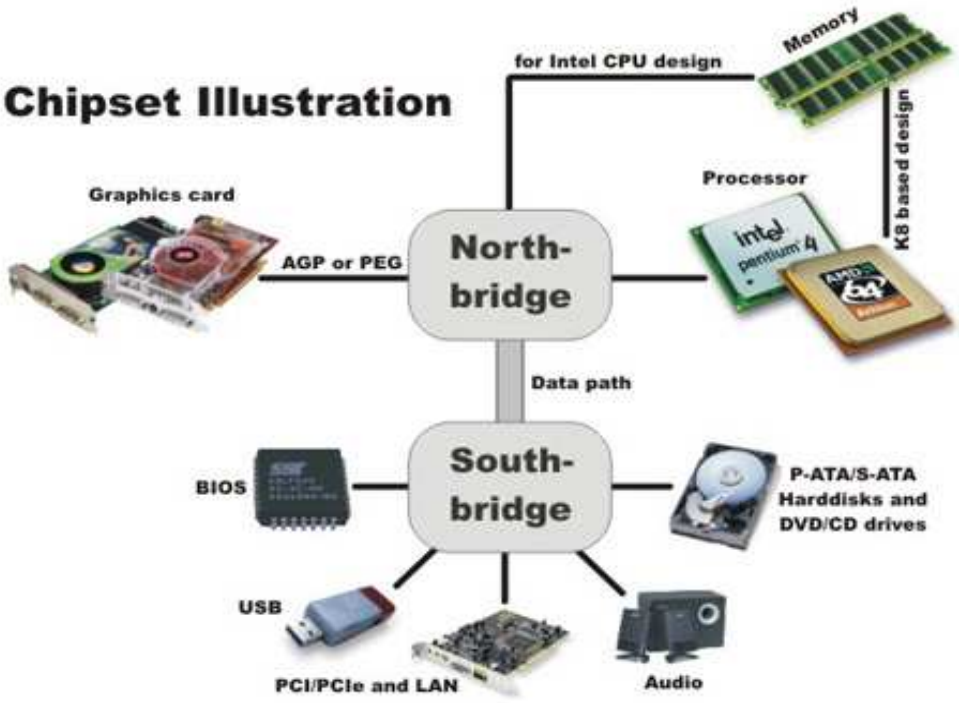
ORDINATEUR

Carte mère



ORDINATEUR

Carte mère



LES CIRCUITS LOGIQUES PROGRAMMABLES

CIRCUITS LOGIQUES PROGRAMMABLES

CPLD : Complex Programmable Logic Device
FPGA : Field Programmable Gate Array

Une stratégie différente :

Microcontrôleur :

- Circuit avec un jeu d'instruction définit par le constructeur
- Execution du programme séquentielle
- Les périphériques ne sont pas modifiables
- Programmation logicielle : Assembleur, C ...

Circuit logique Programmable :

- Aucun jeu d'instruction, contient uniquement des circuits logiques
- Execution séquentielle ou combinatoire
- L'organisation et la connexion des circuits logiques sont modifiables
- Programmation matérielle : le VHDL

CIRCUITS LOGIQUES PROGRAMMABLES

Exemple de circuit simple

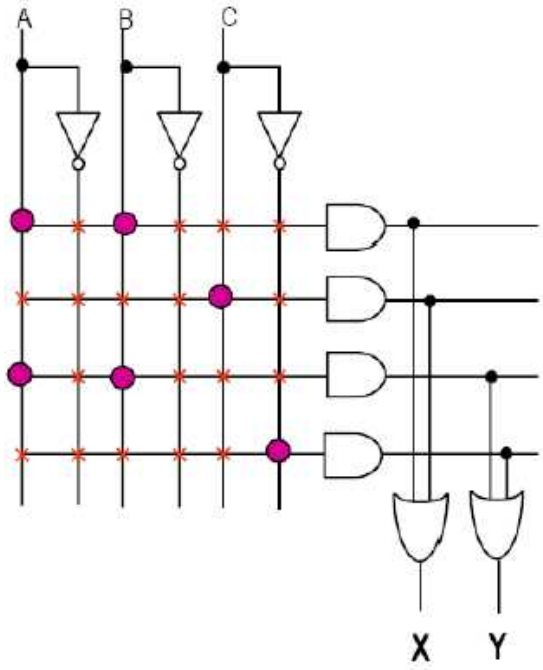
Equations logiques à réaliser :

$$X = A.B + C$$

$$Y = A.B + \overline{C}$$

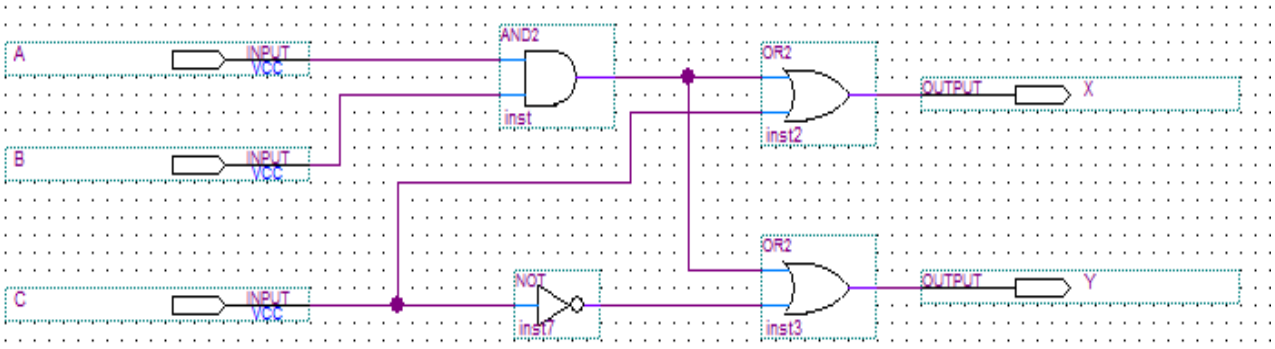
Pour réaliser ces équations :

Utilisation d'un logiciel qui permet une description matérielle (câblage des circuits logiques)



CIRCUITS LOGIQUES PROGRAMMABLES

Réalisation graphique du circuit



CIRCUITS LOGIQUES PROGRAMMABLES

Réalisation avec du code VHDL

```
library ieee;
use ieee.std_logic_1164.all;

entity Test is
    port
    (
        A : in std_logic;
        B : in std_logic;
        C : in std_logic;
        X : out std_logic;
        Y : out std_logic
    );
end Test;


architecture Circuit of Test is

begin
    X <=    (A and B) or C;
    Y <=    (A and B) or not (C);
end Circuit;
```

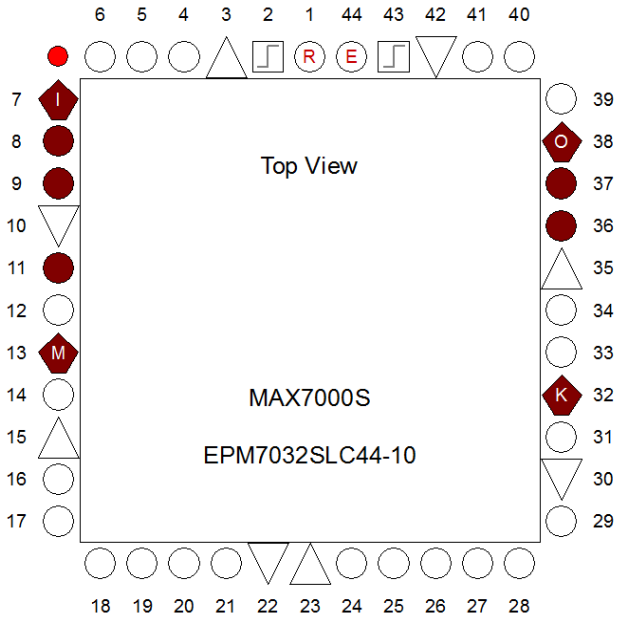
Utilisation des bibliothèques

Description des entrées sorties du circuit

Description du câblage

CIRCUITS LOGIQUES PROGRAMMABLES

Assignation des broches



Node Name	Direction	Location
A	Input	PIN_8
B	Input	PIN_9
C	Input	PIN_11
X	Output	PIN_37
Y	Output	PIN_36
TCK	Input	PIN_32
TDI	Input	PIN_7
TDO	Output	PIN_38
TMS	Input	PIN_13

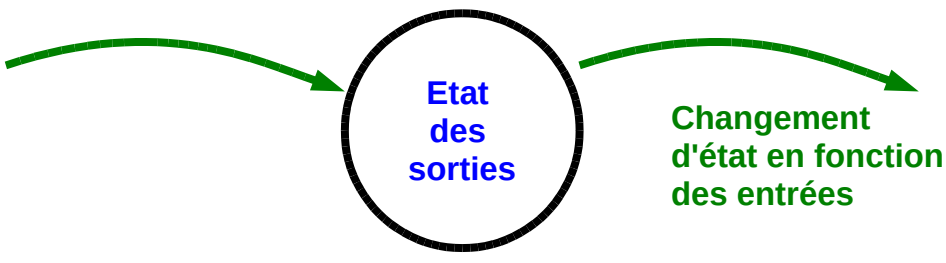
MODELE MATHEMATIQUE DE DESCRIPTION

DESCRIPTION MATHEMATIQUE

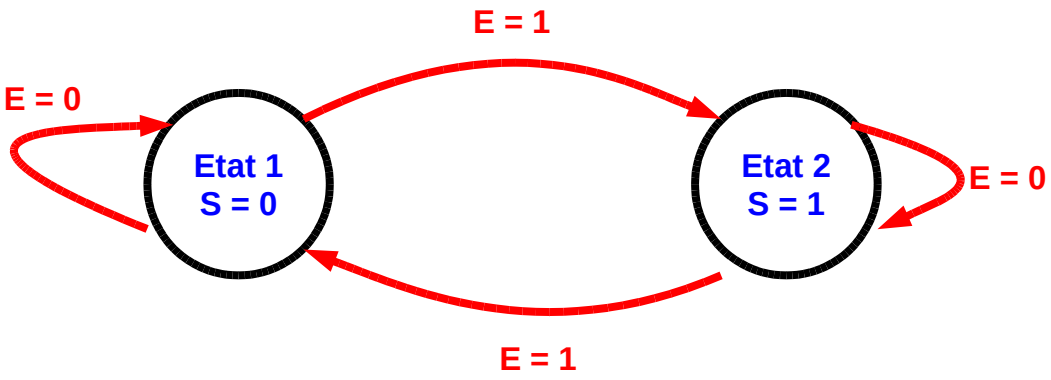
- 2 modèles de description : machine de MEALY et machine de MOORE

- Ces 2 machines fournissent un modèle mathématique pour décrire simplement le fonctionnement d'un système.

- On utilise une représentation sous forme de graphe de transition.

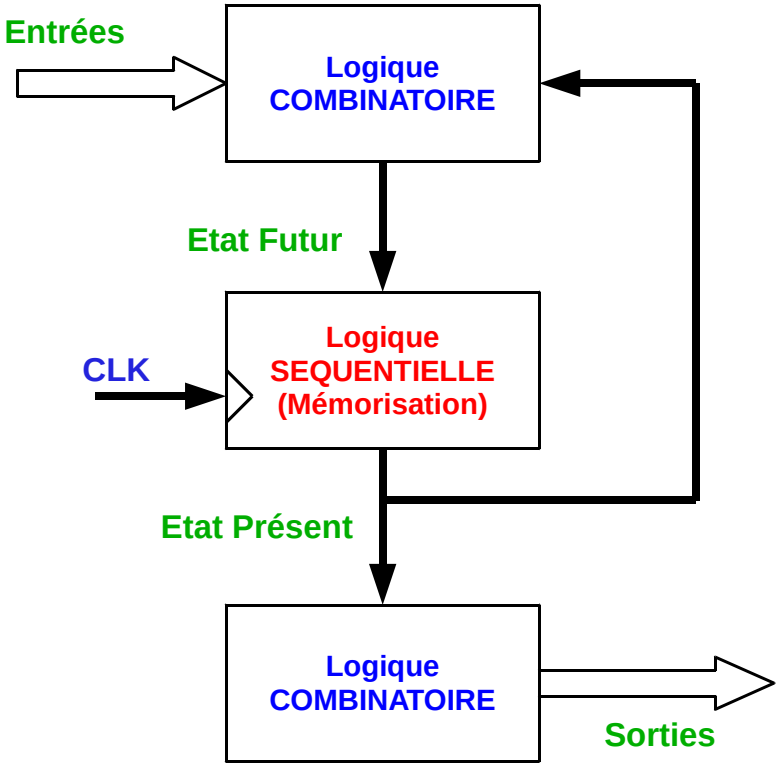


GRAPHE DE TRANSITION



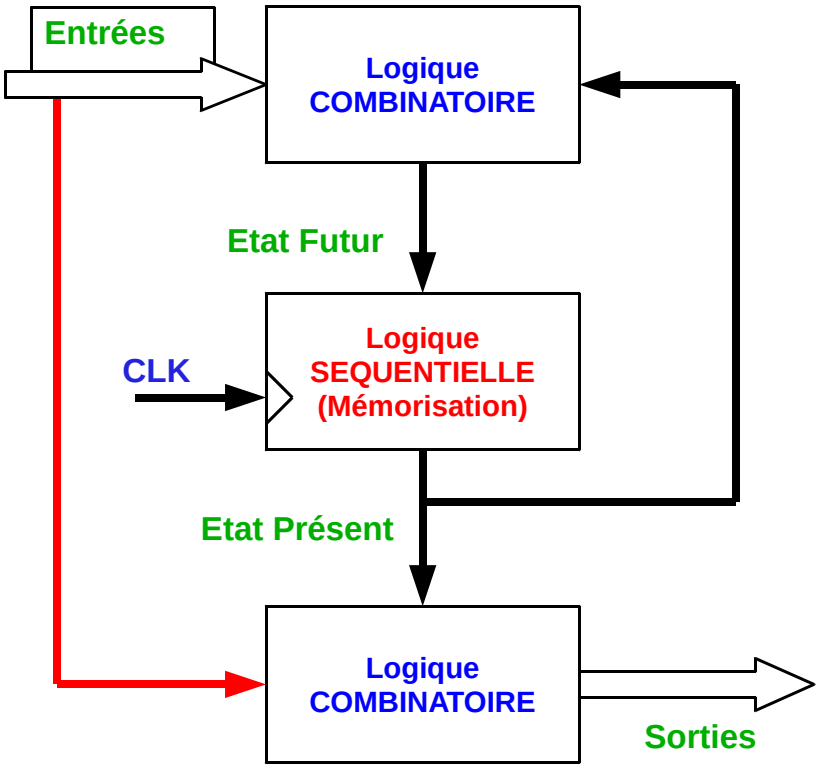
- Les cercles représentent les sorties
- Les flèches les entrées qui engendrent les transitions
- A chaque front d'horloge, on regarde l'état des entrées pour aller vers le nouveau cercle

MACHINE DE MOORE



- Les sorties d'une machine de Moore dépendent de l'état présent et changent sur front d'horloge de manière synchrone
- L'état futur est calculé à partir de l'état présent et des entrées.

MACHINE DE MEALY



- Les sorties d'une machine de Mealy dépendent de l'état présent et des entrées
- L'état futur est calculé à partir de l'état présent et des entrées.
- Sorties asynchrone

EXERCICE

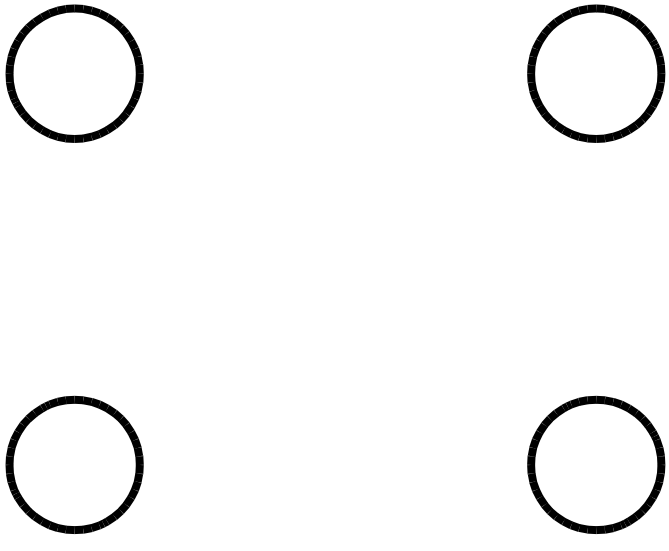
GRAPHE DE TRANSITION

Exercice : compteur – décompteur 2 bits

- Réaliser le graphe de transition d'un compteur décompteur 2 bits
- Lorsque C = 1 on réalise le comptage
- Lorsque D = 1 on réalise le décomptage
- Lorsque R = 1 on fait un Reset quel que soit l'état des autres entrées

GRAPHE DE TRANSITION

Exemple : compteur – décompteur 2 bits



LES CONSTRUCTEURS

- **MICROPROCESSEURS : INTEL et AMD**
- **MICROCONTROLEURS : Microchip**
  Atmel
  Texas Instruments
  Freescale (Motorola)
  Philips
- **DSP : Texas Instrument**
  Analog Devices
  Microchip
- **CPLD et FPGA : ALTERA**
  XILINK
  LATTICE
  ACTEL

CONCLUSION

- **2 stratégies : programmation logicielle et / ou matérielle**
- **programmation logicielle : µC coûte pas cher**
- **Programmation matérielle : FPGA, ASIC (Application Specific Integration Circuit) coûte plus cher mais circuit dédié à l'application donc plus performant.**

CONCLUSION

- **L'avenir s'oriente de plus en plus vers un FPGA avec un microcontrôleur intégré : SoC (System on Chip)**