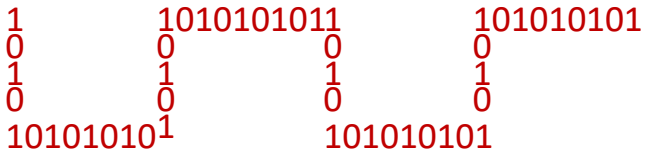
UNIVERSITE MARIEN NGOUABI

Ecole Nationale Supérieure Polytechnique (ENSP)



Architecture des Systèmes à processeurs

PhD. Fraidy Ghislain Bouesse

Architecture des Systèmes à processeurs

Volume horaire : 30h (cours)

Contenu

- Introduction à l'architecture : Système de représentation des données et Architecture de base d'un ordinateur
- Introduction au langage machine : Jeu d'instruction et Branchements
- L'assembleur 80x86 : l'assembleur, Segmentation de la mémoire, Adressage, Pile, procédures
- Notion de compilation
- Les interruptions
- Les mémoires
- Les entrées/sorties
- Les périphériques
- Les architecture actuelles

Introduction à l'architecture

Du numérique à l'analogie et de l'analogie au numérique

Domaine Numérique

Signaux discrets: le numérique possède Certains avantages sur l'analogique dans les applications électroniques.

Avantages:

- **Simplicité de conception.**
- **Bonne robustesse au bruit.**
- **Précision élevée et prévisible.**
- **Stockage de l'information aisé.**
- **Flexibilité (programmation).**
- **Calculs complexes sur les données.**
- **Stockage massif.**
- **Miniaturisation**
- **etc.**

Convertisseur Analogique - Numérique

Convertisseur Numérique – Analogique

Domaine Analogique

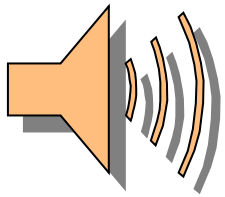
Signaux analogiques:
Signaux continus dans le temps.
La plupart des grandeurs mesurables dans la nature se présentent sous une forme analogique

Exemples:
- **la parole.**
- **le son.**
- **la température .**
- **etc.**

Microphone

Difficultés:

- **Stockage.**
- **Traitement.**
- **Fiabilité.**
- **etc.**

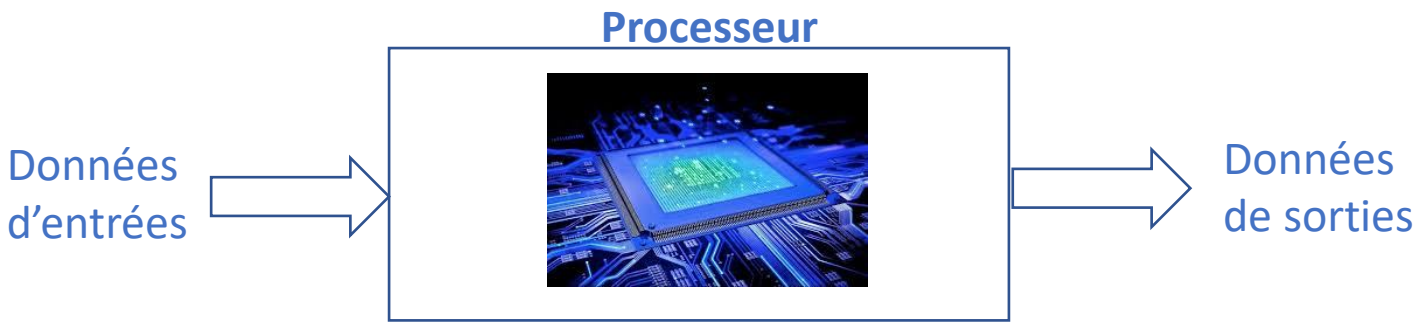


Haut -parleur

Introduction à l'architecture

Introduction : Processeur

Etymologiquement, le mot processeur vient de l'anglais (1957) « Processor » qui désignait une personne, puis un dispositif chargé de l'exécution d'une opération, d'un traitement. Le mot « Processor » est une dérivé du verbe anglais « to process » (exécuter , traiter) et du mot « process » (opérations).
De nos jours, on définit un processeur comme un calculateur électronique; une unité de traitement de données de petite taille représenté sous forme de circuit intégré (CI).



Ce sont des circuits ou composants programmables. Les traitements (fonctions) réalisées par un processeur peuvent être redéfinis. Ils reçoivent des données en entrées (généralement) sous forme binaire et délivrent des sorties sous la même forme.

Introduction à l'architecture

Introduction : Processeur – quelques jalons

- ENIAC (1946)
  - 19000 tubes
  - 30 tonnes
  - surface de 72 m²
  - consomme 140 kilowatts.
  - Horloge : 100 KHz.
  - ≈330 multiplications/s
- 1947: invention du transistor
- 1965: IBM S/360
  - Utilise déjà la plupart des techniques utilisées dans les machines actuelles
- 1971: 1er microprocesseur : Intel 4004
- 1978: microprocesseur Intel 8086 : Jeu d'instruction x86

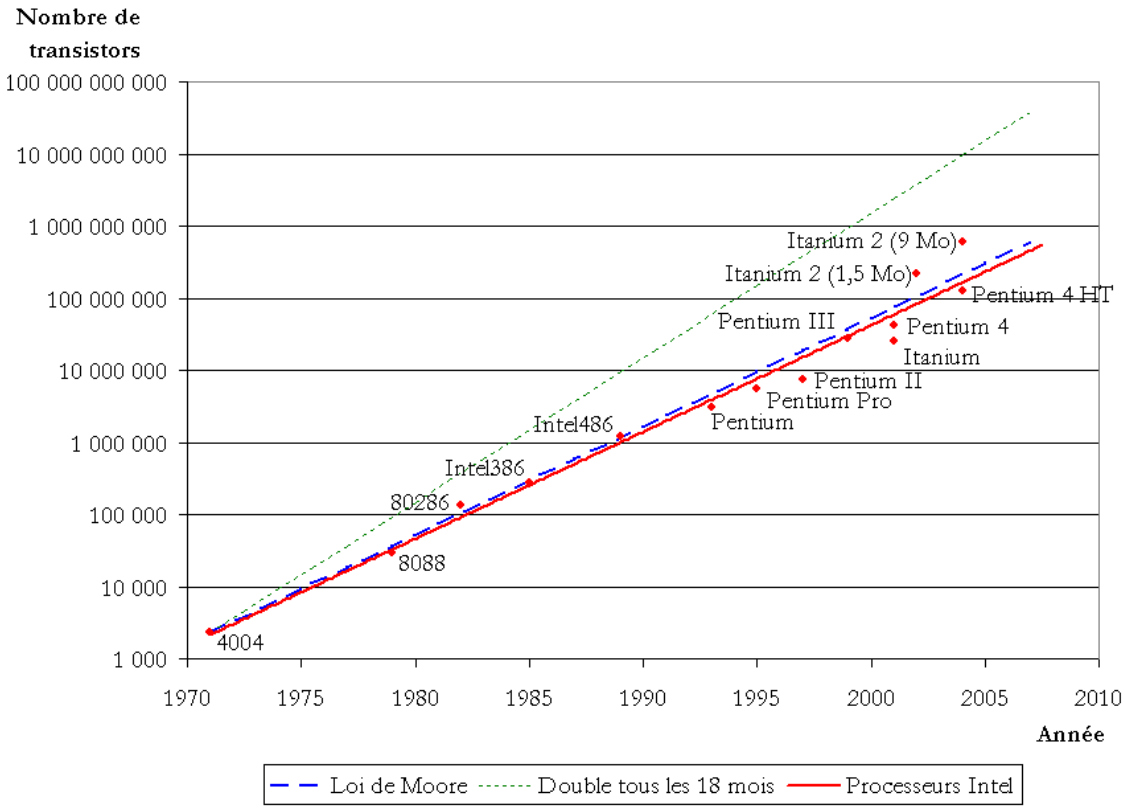
…….

- Mon portable (2006)
  - Intel Duo Processor 2GHz
  - 1 Go DRAM, 100 Go disque
  - 1, 9 kg

Introduction : Processeur

Le nombre de transistors sur une puce de silicium double environ tous les 2 ans

– 1965, Gordon Moore (co-fondateur d'Intel)



La surface d'un transistor est divisée par 2 environ tous les 2 ans

Introduction à l'architecture

Représentation des données

Les informations ou données manipulées dans les systèmes à processeurs sont représentées en binaire (système de numération binaire). Il existe plusieurs types de systèmes de numération: le système décimal, octal, hexadécimale et le système binaire.

Les systèmes de numérations, opérations et codes

Un système de numération est un système qui permet de **lire** et d'**écrire** des nombres.

Base d'un système de numération

La base d'un système de numération est le nombre de chiffres différents qu'utilise ce système de numération.

Introduction à l'architecture

Rappels sur le système décimal

C'est le système à base 10 que nous utilisons tous les jours et qui comprend dix chiffres : 0 -9.

Soit le nombre 1975 de ce système, nous l'écrirons :

$$N=(1975)_{10}.$$

Ce nombre peut également s'écrire sous la forme d'un polynôme :

$$N = 1*10^3+9*10^2+7*10^1+5*10^0$$

Chaque monôme est composé d'un chiffre du nombre **N** multiplié par une puissance de la base. Si nous appelons **a** chiffre et **b** la base, le monôme peut s'écrire sous la forme générale suivante :

$$a.b^p$$

- b^p est le poids du chiffre **a**; **p** est le rang du chiffre **a**

De manière générale, tout nombre décimal N entier de **n** chiffres peut s'écrire sous la forme :

$$N = a_n*10^{n-1} + a_1*10^1 +\ldots+ a_0*10^0 \rightarrow \quad N = \sum_{i=0}^{i=n-1} \alpha_i * 10^i$$

a_i est l'un des chiffres de **N** de rang i et de poids 10^i. **n** est le rang du chiffre de poids le plus fort et **0** le poids du chiffre de poids le plus faible.

Remarque : Les poids positionnels des fractions sont des puissances négatives de 10 et diminue de gauche vers la droite en commerçant par 10^{-1}.

$$\ldots 10^2\ 10^1\ 10^0\ ,\ 10^{-1}\ 10^{-2}\ 10^{-3}\ \ldots..$$

Introduction à l'architecture

Rappels sur le système Octal

Ce système utilise une base 8 et comprend huit symboles qui sont les chiffres compris entre 0 et 7.

Exemple : **N = $(1717)_8 = (975)_{10}$**, son expression sous forme polynomiale est :

$$N = 1*8^3 + 7*8^2 + 1*8^1 + 7*8^0$$

L'expression générale d'un nombre octal, présentée sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n-1} \alpha_i * 8^i$$

Rappels sur le système Hexadécimal

Ce système, dit à base 16, comprend seize symboles, dix chiffres et six lettres : 0-9 et A à F.

Exemple : **N = $(A7B7)_{16} = (42935)_{10}$**, son expression sous forme polynomiale est :

$$N = A*16^3 + 7*16^2 + B*16^1 + 7*16^0$$

L'expression générale d'un nombre hexadécimal présentée sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n-1} \alpha_i * 16^i$$

Introduction à l'architecture

Rappels sur le système binaire

Le système de numération binaire est plus simple que le système décimal, octal et hexadécimal puisqu'il possède que 2 chiffres. C'est un système à base de 2, composé du chiffre 1 et 0. Chacun d'eux est aussi appelé bit (contraction de binary digit : élément binaire).

Exemple : **$N = (1010)_2 = (10)_{10,}$** sa représentation sous forme de polynôme est donnée par l'expression suivante :

$$N = 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$$

L'expression générale d'un nombre binaire présentée sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n-1} \alpha_i * 2^i$$

En utilisant **n** bits, on peut former **2^n** nombres différents dont le plus grand est égal à **2^n-1**.

Exemple : **n = 8 => 2^n=256 nombres différents dont le plus grand est $(11111111)_2 = (2^n-1) = (255)_{10}$**

Notions:
- Octet
- Most Significante Bit : MSB
- Least Significante Bit : LSB
- Little and Big endian (Little : LSB à droite; Big LSB à gauche)

Rappels sur les systèmes de numération

Correspondance entre nombres de différentes bases.

Base			
16	*10*	*8*	*2*
0	*0*	*0*	*0000*
1	*1*	*1*	*0001*
2	*2*	*2*	*0010*
3	*3*	*3*	*0011*
4	*4*	*4*	*0100*
5	*5*	*5*	*0101*
6	*6*	*6*	*0110*
7	*7*	*7*	*0111*
8	*8*	*10*	*1000*
9	*9*	*11*	*1001*
A	*10*	*12*	*1010*
B	*11*	*13*	*1011*
C	*12*	*14*	*1100*
D	*13*	*15*	*1101*
E	*14*	*16*	*1110*
F	*15*	*17*	*1111*

Rappels sur les systèmes de numération: Changement de base

Les systèmes de conversion permettent de changer la base du système de numération d'un chiffre. Ainsi quel que soit le nombre N défini dans la base b, ce nombre peut être représenté dans les différentes bases possibles du système de numération.

Conversion d'un nombre décimal en binaire et d'un nombre binaire en décimal

- Du binaire au décimal

La formule suivante permet de convertir un nombre binaire en décimal : $N = \sum_{i=0}^{i=n-1} \alpha_i * 2^i$

avec **n** le nombre de bit et $\mathbf{a_i}$ la valeur du bit de rang **i**.

Exemple :

1- Convertissez le nombre binaire 1101101 en décimal.

$N = 1*2^6+1*2^5+0*2^4+1*2^3+1*2^2+0*2^1+1*2^0$
N = 64 + 32 + 8 +4 + 1 = 109

2- Convertissez le nombre 0,1011en décimal.

$N = 1*2^{-1} + 0*2^{-2} + 1*2^{-3} + 1*2^{-4}$
N= 0.5 + 0.125 + 0.0625 = 0.6875

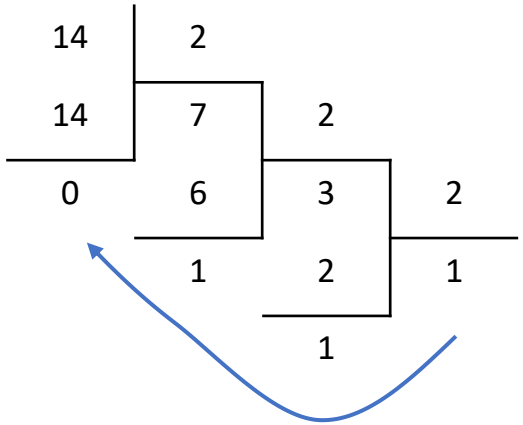
Rappels sur les systèmes de numération: Changement de base

➢ Du décimal au binaire

Il existe plusieurs méthodes de conversion d'un nombre décimal en binaire. Parmi ces nombreuses méthodes, nous nous limiterons dans ce cours, à la méthode par division.

Le nombre à convertir est divisé par la base **b** (dans notre cas **b=2**) et le reste de la division est conservé. Le quotient obtenu est divisé par b et le reste est à nouveau conservé. L'opération est répétée sur chaque quotient obtenu. Les restes successifs sont écrits, en commençant par le dernier, de la droite vers la gauche pour former l'expression de **N** dans le système de base **b**.

Exemple : convertissez la valeur $(14)_{10}$ en binaire



$(14)_{10}= (1110)_2$

Rappels sur les opérations arithmétiques en binaire

➢ Représentation des nombres signés

La représentation des nombres signés impose de prendre en compte deux symboles supplémentaires : **+** et **–**

La représentation de ces symboles est réalisée par l'ajout d'un bit en plus au nombre: le bit signe

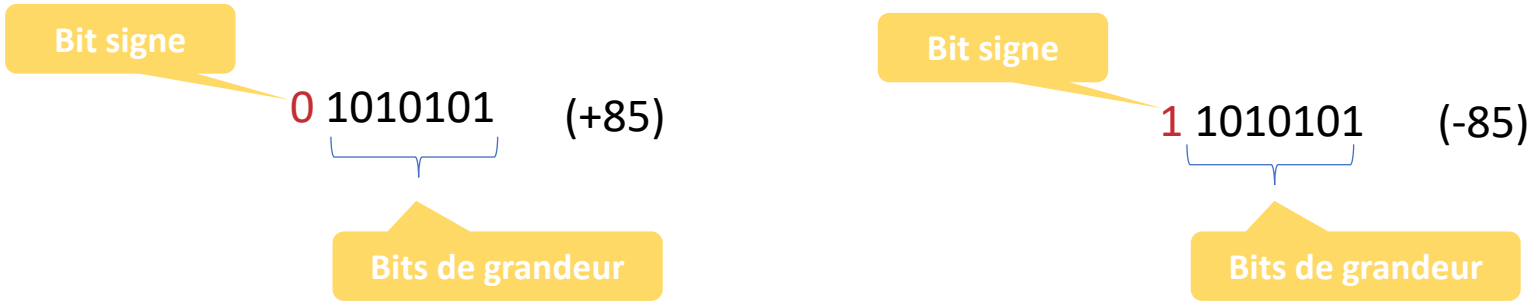
- Si le bit signe est 0, le nombre est positif
- Si le bit signe est 1, le nombre est négatif

Ainsi, on parle de nombres signés

❖ Notation en grandeur exacte

C'est la notation la plus évidente, elle est constituée de deux parties:

- une première partie qui représente les bits de grandeurs
- une seconde partie qui représente le bit signe



Introduction à l'architecture

❖ Complément à 1 d'un nombre binaire

Complémenter un bit à 1, revient à remplacer les bits 0 à 1 et les bits 1 à 0.
Complémenter à 1 un nombre binaire revient à compléter à 1 chaque bit de grandeur

(-85) 1 1010101 notation GE (Grandeur Exacte)
1 0101010 notation C1 (Complément à 1)

Le bit signe reste inchangé

D'un point de vue algébrique, le complément d'un nombre a écrit en base b sur n chiffres, s'obtient par $(b^n - 1) - a$

De -7 à +7 ….

Positif		Négatif	
0000	+0	1000	-7
0001	+1	1001	-6
0010	+2	1010	-5
0011	+3	1011	-4
0100	+4	1100	-3
0101	+5	1101	-2
0110	+6	1110	-1
0111	+7	1111	-0

Introduction à l'architecture

❖ Complément à 2 d'un nombre binaire

Pour obtenir le complément à 2 d'un nombre binaire, il faut prendre le complément à 1 de ce nombre et lui rajouter 1.

Complémenter à 1 un nombre binaire, revient à compléter à 1 chaque bit de grandeur

(-85)

```
  1 1010101 notation GE (Grandeur Exacte)
  1 0101010  notation C1 (Complément à1)
+         1
  1 0101011  notation C2 (complément à 2)
```

D'un point de vue algébrique, le complément d'un nombre a écrit en base b sur n chiffres, s'obtient par $b^n - a$

Positif		Négatif	
0000	+0	1001	-7
0001	+1	1010	-6
0010	+2	1011	-5
0011	+3	1100	-4
0100	+4	1101	-3
0101	+5	1110	-2
0110	+6	1111	-1
0111	+7	0000	-0

De -7 à +7 ….

Introduction à l'architecture

Addition (soustraction)

- Addition de deux nombres positifs
- Addition de deux nombres de signes contraires
- Addition de deux nombres négatifs

Base de l'addition	
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	0 plus retenue à 1

Multiplication

- Même principe qu'en décimal
- Différente architecture pour les nombres signés

Base de Multiplication	
0 x 0	0
0 x 1	0
1 x 0	0
1 x 1	1

Division

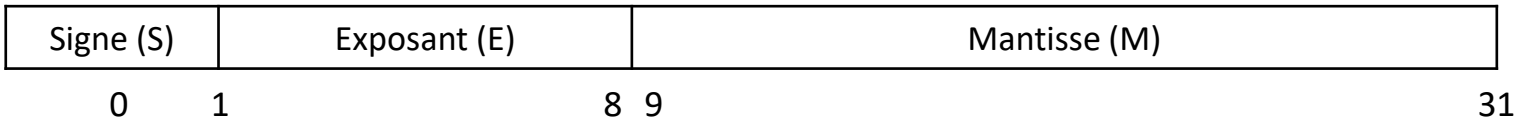
- Méthode de soustraction successive
- Même principe qu'en décimal

Introduction à l'architecture

❖ Représentation des nombres réels (Norme IEEE)

La représentation des nombres réels en binaire a été normalisée sous la forme $\pm 1, M \cdot 2^{\pm E}$. La représentation IEEE code séparément le signe, l'exposant et la mantisse (la suite de bits après la virgule).

Représentation sur 32 bits



- Le signe est représenté sur le bit de poids fort S, + est représenté par 0 et - par 1.
- L'exposant est codé sur les 8 bits E. On code en binaire la valeur *n* + 127.
- La mantisse est codée sur les 23 bits de poids faibles M.

Remarques :

1. Les exposants 00000000 et 11111111 sont interdits :
   - l'exposant 00000000 signifie que le nombre est dénormalisé ;
   - l'exposant 11111111 indique que l'on n'a pas affaire à un nombre (on note cette configuration NaN, Not a Number, et on l'utilise pour signaler des erreurs de calculs, comme par exemple une division par 0).
2. Le plus petit exposant est donc -126, et le plus grand +127

Introduction à l'architecture

❖ Représentation des nombres réels (Norme IEEE)

Encodage	Signe	Exposant	Mantisse	Valeur d'un nombre	Précision	Chiffres significatifs
32 bits	1 bit	8 bits	23 bits	$Sx1,M \cdot 2^{E+127}$	24 bits: Simple précision	≈ 7 bits
64 bits	1 bit	11 bits	52 bits	$Sx1,M \cdot 2^{E+1023}$	53 bits: Double précision	≈ 16 bits

- Bit de signe : 0 pour les nombres positifs et 1 pour les nombres négatifs ;
- Exposant : excentrement de 127 pour les nombres à simple précision et 1023 pour les nombres à double précision ;
- Mantisse 23 bits ou 52 bits ;
- Normalisation : 1,M… avec 1 comme bit caché. Le bit précédant la virgule n'est pas codé

Exemple: Codons le nombre -6, 625 en simple précision

- $(6, 625)_{10} = (110, 1010)_2$
- $110, 1010 = 1, 101010 \times 2^2$
- 10101000000000000000000
- $127 + 2 = (129)_{10} = (10000001)_2$
- 1 10000001 10101000000000000000000
- En hexadécimal : C0 D4 00 00

Rappels sur les codes

Les informations manipulées dans les systèmes numériques sont représentées sous forme de codes binaires. Ils permettent d'assurer une correspondance unique entre le symbole représentant une information et sa forme binaire. Les codes permettent également la détection et la correction des erreurs (transmission des données).

❖ Codes pondérés

Dans les codes pondérés, les positions des éléments binaires sont affectées de poids :

$$(N)_b = \sum_i a_i b^i$$

N est le nombre encore appelé **mot**, b est la base, i le **rang**, a_i le **chiffre** de rang i et b^i le **poids** du chiffre a_i

- Exemple: Code DCB - Décimal Codé Binaire

Code décimal	0	1	2	3	4	5	6	7	8	9
Code DCB	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Rappels sur les codes

❖ Codes non pondérés

Dans les codes non pondérés, les positions binaires ne sont pas affectées de poids. Ainsi, ils ne conviennent pas aux calculs arithmétiques.

- **Exemple: Code Cyclique - Code Gray**

  Code dans lequel les valeurs successives ne diffèrent que d'un caractère. On parle aussi de « binaire réfléchi ».

Code décimal	0	1	2	3	4	5	6	7
Code de Gray	000	001	011	010	110	111	101	100

- **Exemple: Code Alphanumérique**

  Il permet de représenter les caractères de l'alphabet, les caractères spéciaux, les chiffres etc. on considère que le minimum de caractères d'un code alphanumérique est de 92:

  - 26 pour les lettres minuscules
  - 26 pour les lettres majuscules
  - 10 pour les chiffres
  - 30 pour les caractères spéciaux: + - / % > < …

  Le code le plus utilisé est le codes ASCII

Rappels sur les codes

- Exemple: Code Alphanumérique : Code ASCII

Table des codes ASCII (hexadécimal)

00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL
08 BS	09 HT	0A NL	0B VT	0C NP	0D CR	0E SO	0F SI
10 DLE	11 DC1	12 DC2	13 DC3	14 DC4	15 NAK	16 SYN	17 ETB
18 CAN	19 EM	1A SUB	1B ESC	1C FS	1D GS	1E RS	1F US
20 SP	21 !	22 "	23 #	24 $	25 %	26 &	27 '
28 (	29)	2A *	2B +	2C ,	2D -	2E .	2F /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7A z	7B {	7C |	7D }	7E ~	7F DEL

Rappels sur les codes

❖ Codes de détection d'erreurs

La transmission des données dans les circuits numériques s'accompagne parfois d'erreurs qui doivent être détectées. La méthode de bit de parité est l'une des approches utilisée pour la détection.

- **Bit de parité paire**

  Le bit supplémentaire est fixé à une valeur (0 et 1) telle que, pour chaque « mot », le nombre total de 1 y compris le bit de parité, soit **pair**.

Caractère	0	1	A	B
ASCII (parité pair)	**0**011 0000	**1**011 0001	**0**100 0001	**0**100 0010

- **Bit de parité impaire**

  Le bit supplémentaire est fixé à une valeur (0 et 1) telle que, pour chaque « mot », le nombre total de 1 y compris le bit de parité, soit **impair**.

Caractère	0	1	A	B
ASCII (parité pair)	**1**011 0000	**0**011 0001	**1**100 0001	**1**100 0010

Introduction à l'architecture

Rappels : technologie des processeurs

L'élément de base constituant les CI (processeurs) est le transistor.

Le transistor est principalement utilisé comme « interrupteur commandé » en électronique numérique. Mais il permet également de stabiliser une tension, de moduler un signal et/ou de l'amplifier.

Le transistor est un dispositif semi-conducteur qui permet de contrôler grâce à une électrode d'entrée, un courant ou une tension sur une électrode de sortie.

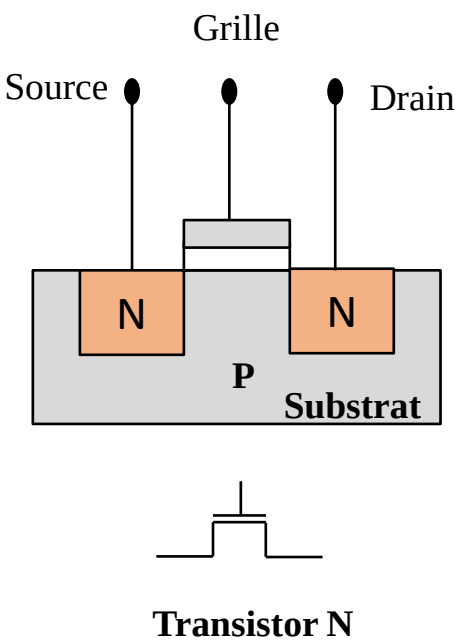
Deux types de transistors sont couramment utilisés pour la conception des CI.

- Les transistors bipolaires à jonction : pour les circuits à faible intégration
- les transistors à effet de champ à semi-conducteur: MOSFET (Metal Oxyde Semi-conductor Field Effect Transistor). Pour les circuits à très grande intégration – processeurs

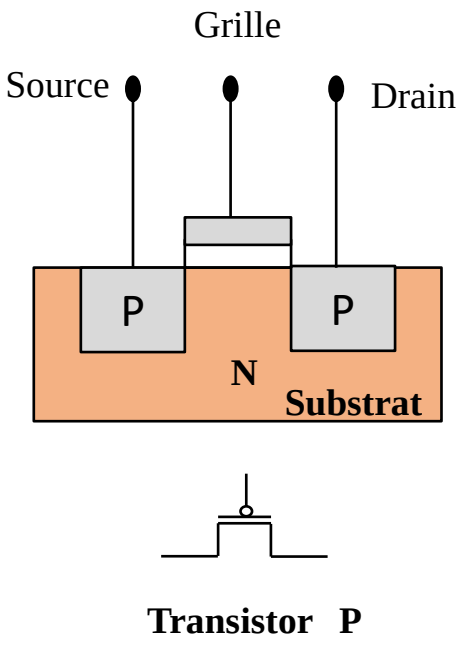
Rappels : technologie des processeurs

- Le transistor à effet de champ (MOSFET)
- Principale technologie utilisée: CMOS

  Les MOSFET sont les commutateurs actifs des circuits CMOS



Transistor N



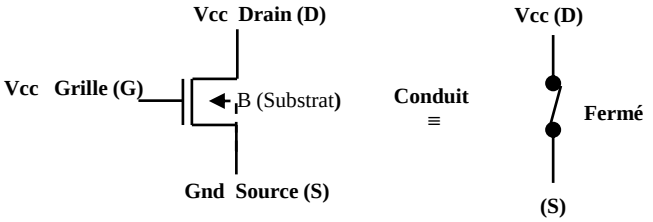
Transistor P

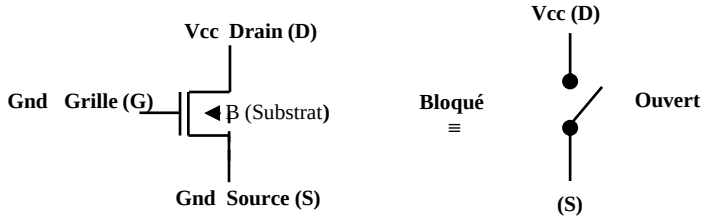
Introduction à l'architecture

Rappels : technologie des processeurs

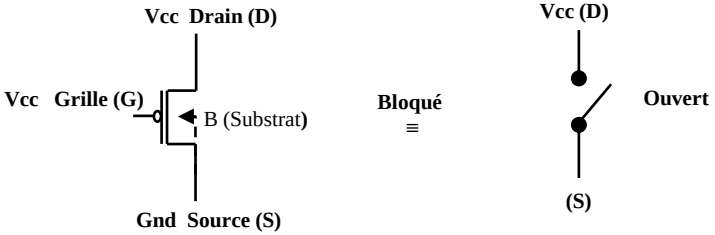
Fonctionnement en Interrupteur

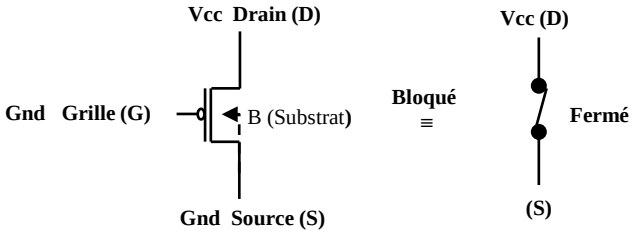
Transistor NMOS





Transistor PMOS

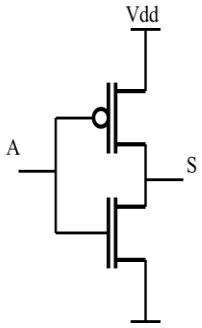




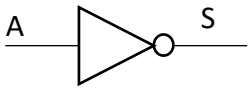
Rappels : technologie des processeurs

Exemples :

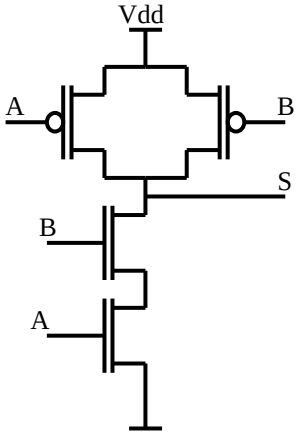
Description d'un circuit de base: l'inverseur en CMOS



- Si A = 0, le PMOS est passant et le NMOS bloqué
  => S = Vdd
- Si A = 1, le PMOS est bloqué et le NMOS passant
  => S = 0



Description d'un circuit de base: la porte Nand



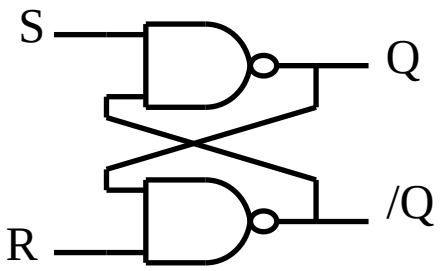
- Si A = B = "1", les NMOS sont passants, les PMOS bloqués,
  => S = "0"
- Si une des entrées est à "0", la branche NMOS est ouverte et un des PMOS est passant,
  => S = "1"

Rappels sur les éléments de mémorisation: les bascules

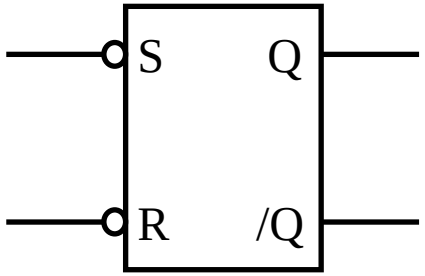
❖ La bascule RS

L'état de sortie de la bascule RS, dépend non seulement de la combinaison appliquée à l'entrée mais aussi de l'état précédent. Ils sont dits séquentiels et ont un effet de mémorisation. Le circuit de mémorisation le plus utilisé est la bascule RS.

- **Bascule RS** : en portes Nand
  - Entrées actives à l'état 0
  - Etat de repos de la bascule : S = R = 1
    - ATTENTION à l'état ambiguë !!



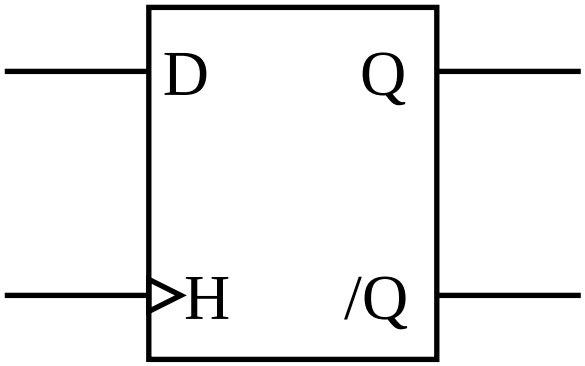
S R	sortie
0 0	ambiguë
0 1	Q = 1
1 0	Q = 0
1 1	inchangée

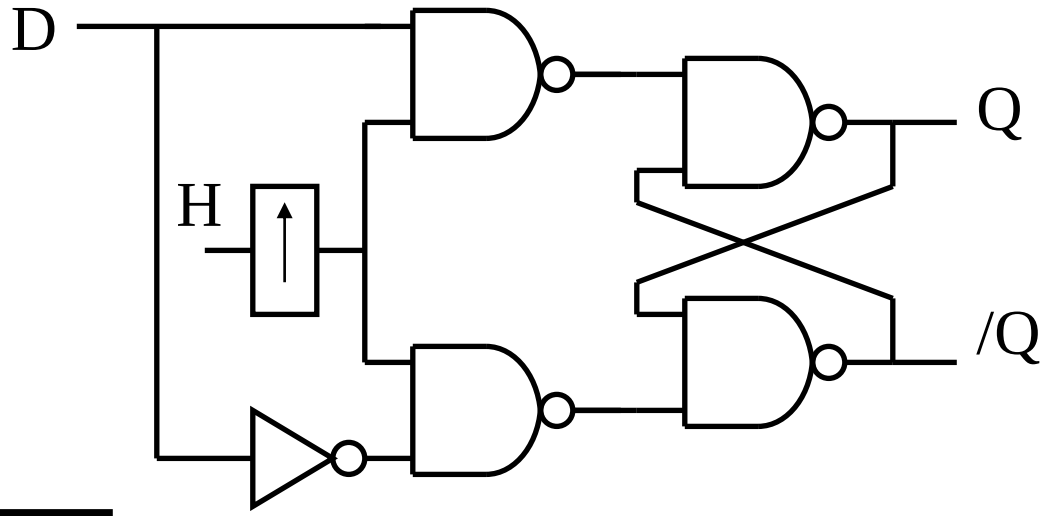


Rappels sur les éléments de mémorisation: les bascules

❖ Bascule D

- La plus utilisée car la plus simple !!
- Bascule synchrone en général à front d'horloge.



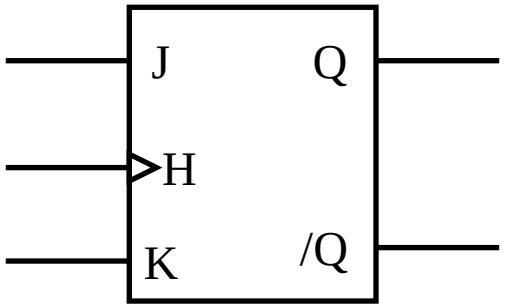


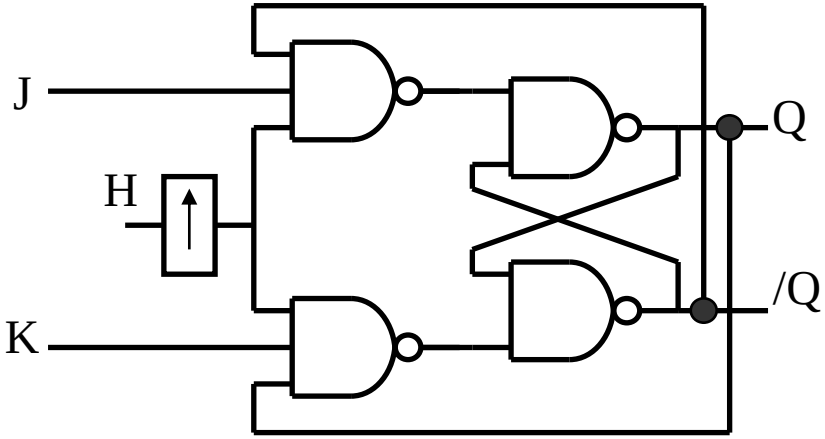
H	D	sortie
0	X	inchangée
1	0	Q = 0
1	1	Q = 1

Rappels sur les éléments de mémorisation: les bascules

❖ Bascule JK

- Proche de la bascule RS mais sans combinaison ambiguë
- J = set, K = reset





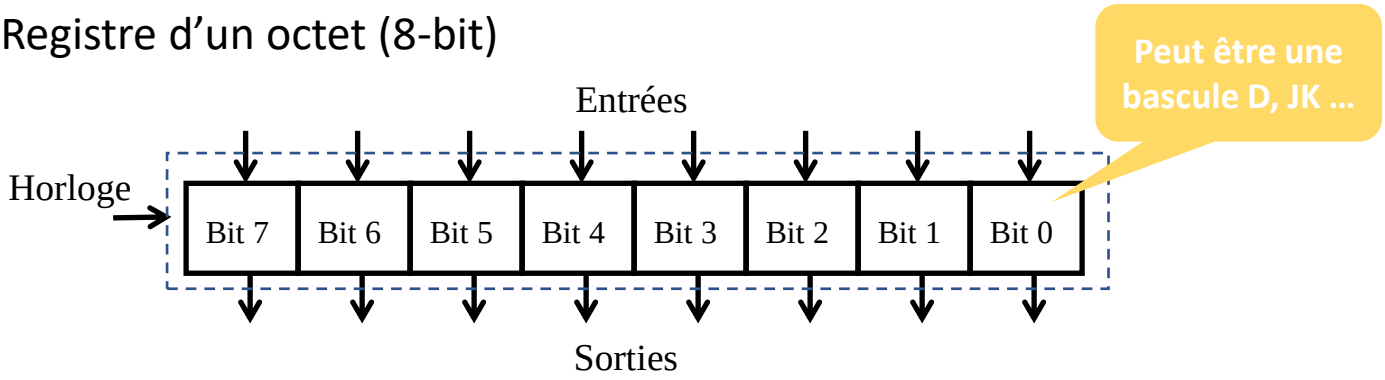
J	K	Q
0	0	inchangée
0	1	Q = 0
1	0	Q = 1
1	1	/Q

Rappels sur les éléments de mémorisation: les bascules

❖ Les registres

Un registre d'un mot de ***n bits*** est une cellule mémoire composée de ***n mémoires élémentaires*** (Bascules).

Exemple : Registre d'un octet (8-bit)



Les registres permettent de stocker et de transférer des données de manière simple :

- ❑ Toutes les bascules du registres ont la même horloge
- ❑ On peut stocker ou transférer les données en série ou en parallèle :
  - Conversion synchrone série // ou l'inverse

Les registres sont des interfaces "tampons" très utilisés dans des environnements logiques complexes (processeurs/Microcontrôleurs) .

Rappels sur les éléments de mémorisation: les bascules

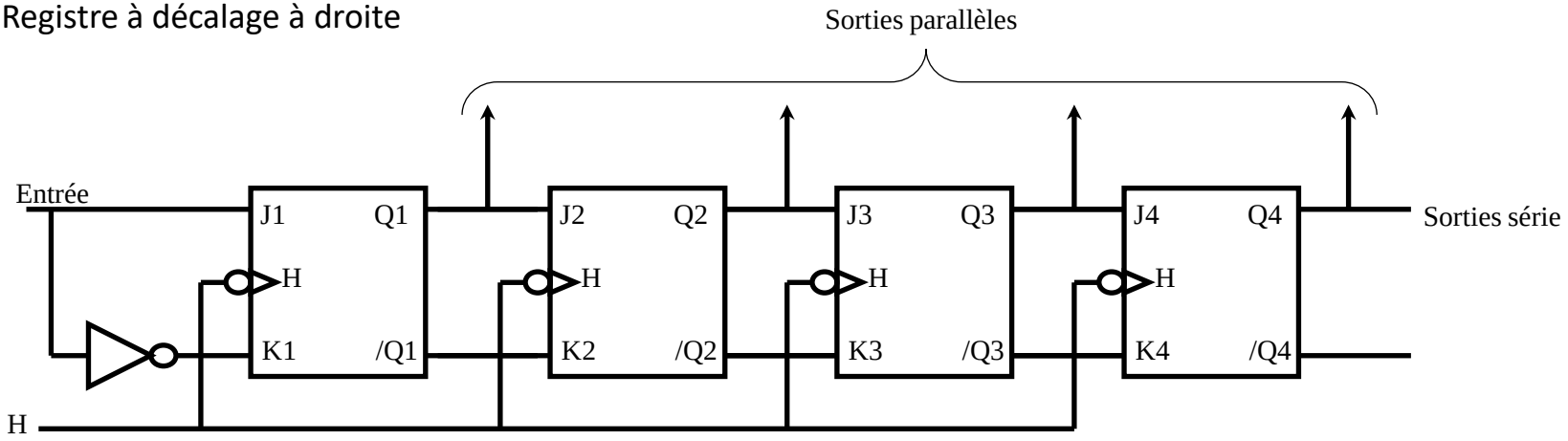
❖ Les registres à décalage (shift register)

La fonction de décalage consiste à faire glisser l'information de chaque bascule dans des bascules adjacentes.

- Si le décalage se fait vers la droite, on parle d'un registre à décalage à droite
- Si le décalage se fait vers la gauche, on parle d'un registre à décalage à gauche

La sortie peut être lue sur l'ensembles des bascules (sortie parallèle) ou uniquement sur la dernière bascule (sortie série).

Exemple: Registre à décalage à droite



Rappels sur les éléments de mémorisation: les bascules

❖ Les compteurs

Les compteurs permettent d'établir une relation d'ordre de succession d'événements. Ils sont utilisés pour effectuer :

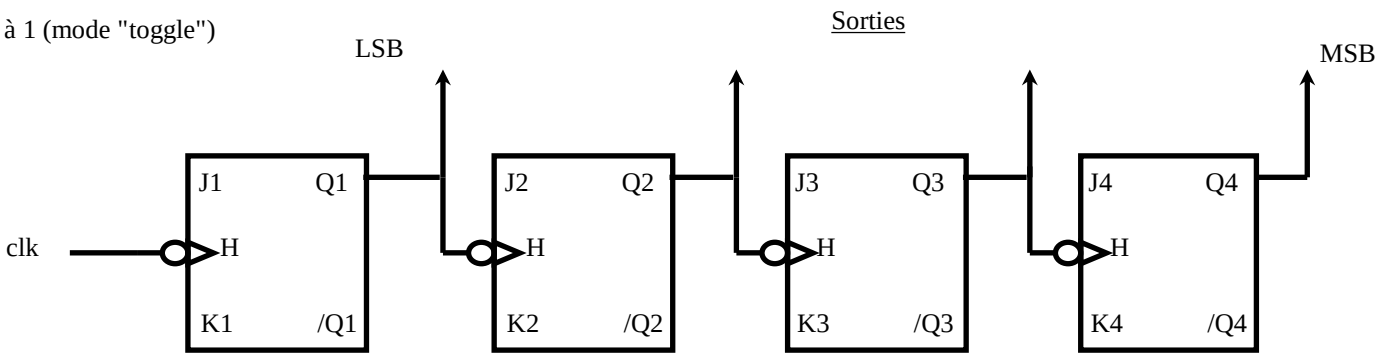
- Du comptage
- Des opérations de séquencement
- La division de fréquence
- Des opérations arithmétiques

Il existe deux types de compteurs

- Les compteurs asynchrones (compteur série): l'ordre des changements d'état de chacune des bascules se propage en cascade
- Les compteurs synchrones (compteur parallèle): les changements d'état est synchronisé par un signal d'horloge.

Exemple Compteur asynchrone modulo 16 à bascules JK :

Nota : Toutes les entrées sont à 1 (mode "toggle")



Rappels technologique

❖ Les compteurs

Les compteurs permettent d'établir une relation d'ordre de succession d'événements. Ils sont utilisés pour effectuer :

- Du comptage
- Des opérations de séquencement
- La division de fréquence
- Des opérations arithmétiques

Il existe deux types de compteurs

- Les compteurs asynchrones (compteur série): l'ordre des changements d'état de chacune des bascules se propage en cascade
- Les compteurs synchrones (compteur parallèle): les changements d'état est synchronisé par un signal d'horloge.

Exemple Compteur asynchrone modulo 16 à bascules JK :

Nota : Toutes les entrées sont à 1 (mode "toggle")