UFR SSMT – LICENCE 1 PC - version Juin 2016

INITIATION A L'INFORMATIQUE

Prof AMAN Angora
UFR SSMT- LAPA MF –Bât de Recherches, RDC
Email : angora.aman@gmail.com
Tél : +22507827752

Références biblio : T. Dumartin ; F. Pellegrini

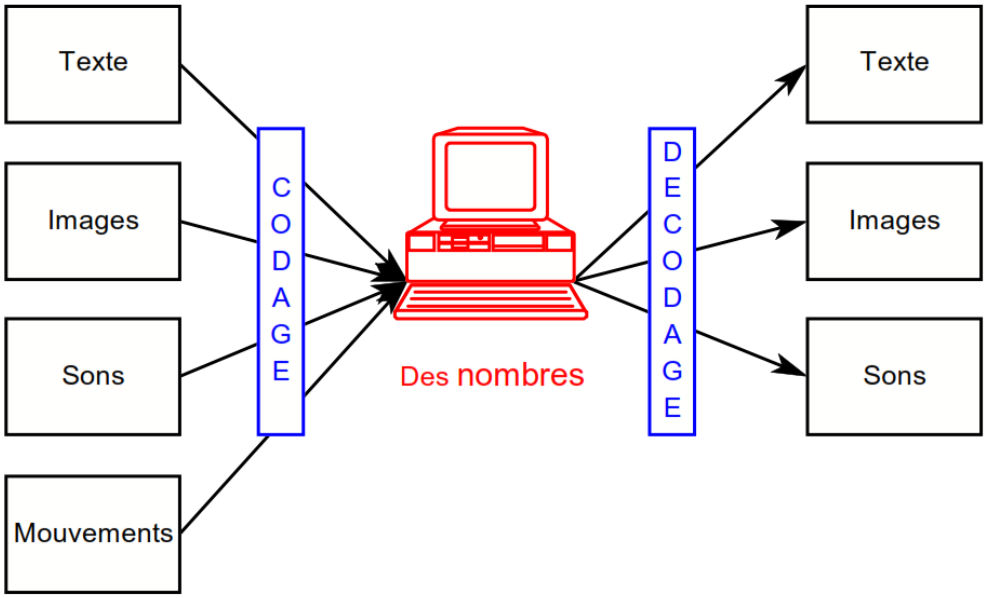
CHAPITRE 1: GENERALITES

INTRODUCTION

L'informatique, contraction d'information et automatique, est la science du traitement de l'information. Apparue au milieu du 20ème siècle, elle a connu une évolution extrêmement rapide. A sa motivation initiale qui était de faciliter et d'accélérer le calcul, se sont ajoutées de nombreuses fonctionnalités, comme l'automatisation, le contrôle et la commande de processus, la communication ou le partage de l'information.

L'architecture des systèmes à microprocesseurs expose les principes de base du traitement programmé de l'information. La mise en œuvre de ces systèmes s'appuie sur deux modes de réalisation distincts, le matériel et le logiciel. Le matériel (**hardware**) correspond à l'aspect concret du système : unité centrale, mémoire, organes d'entrées-sorties, etc…

Le logiciel (**software**) correspond à un ensemble d'instructions, appelé programme, qui sont contenues dans les différentes mémoires du système et qui définissent les actions effectuées par le matériel.



La numérisation au cœur du multimédia

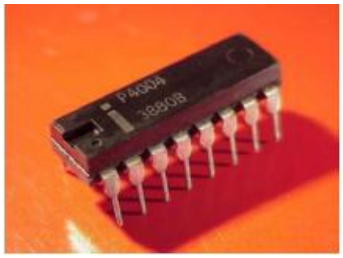
QU'ENTEND- T - ON PAR ARCHITECTURE ?

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

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

QU'EST-CE QU'UN MICROPROCESSEUR ?

Un microprocesseur est un circuit intégré complexe. Il résulte de l'intégration sur une puce de fonctions logiques combinatoires (logiques et/ou arithmétique) et séquentielles (registres, compteur, etc…). Il est capable d'interpréter et d'exécuter les instructions d'un programme. Son domaine d'utilisation est donc presque illimité.
Le concept de microprocesseur a été créé par la Société **Intel.** Cette Société, créée en 1968, était spécialisée dans la conception et la fabrication de puces mémoire. À la demande de deux de ses clients — fabricants de calculatrices de terminaux — Intel étudia une unité de calcul implémentée sur une seule puce. Ceci donna naissance, en **1971**, au premier microprocesseur, **le 4004**, qui était une unité de calcul **4 bits** fonctionnant

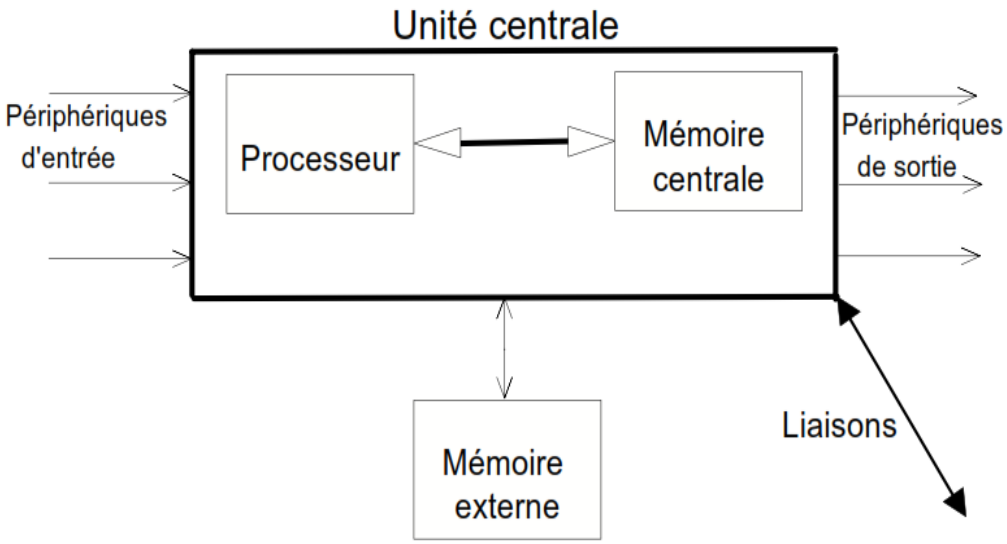


OU TROUVE-T-ON LES SYSTEMES A MICROPROCESSEURS ?

Les applications des systèmes à microprocesseurs sont multiples et variées :
- Ordinateur, PDA
- console de jeux
- calculatrice
- télévision
- téléphone portable
- distributeur automatique d'argent
- robotique
- lecteur carte à puce, code barre
- automobile
- avion
- instrumentation
- etc…

Les opérations élémentaires à effectuer seront désignées par **instructions**, les instructions sont regroupées **en programmes**, les programmes sont rangés **en**

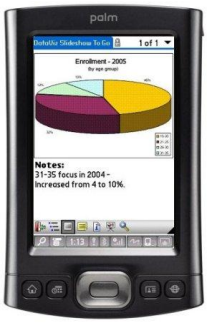
mémoire. La machine peut maintenant dérouler un programme sans intervention extérieure et sans avoir à recharger celui-ci chaque fois que l'on désire en lancer l'exécution.



LES TYPES D'ORDINATEURS.

- Ordinateurs de bureau PC (Personal Computer)

Ordinateurs portables



- Agendas, PIM (Personal Information Manager)

Netbooks (ce sont de gros agendas) **- Tablettes**

Les tablettes finiront par remplacer tous les autres types d'ordinateurs. Associées à des serveurs, c'est toute la connaissance dans son sac. Plus de clavier car tactiles.

- Téléphones portables

Avec la miniaturisation les téléphones portables rentrent dans le cercle et abandonnent leurs côtés autonomes pour devenir à la fois périphériques et unités centrales.
Avec l'avènement des tablettes, le téléphone deviendra un périphérique de la tablette, au même titre que la souris, le Wi-Fi ou les modules GPS.

LES SYSTEMES D'EXPLOITATION

Un système d'exploitation, ou logiciel système, ou *Operating System* (OS), est un logiciel qui, dans un appareil électronique, pilote les dispositifs matériels et reçoit des instructions de l'utilisateur ou d'autres logiciels (ou applications). Ces logiciels doivent être adaptés à un système d'exploitation.

Rôles du système d'exploitation

Dans un ordinateur, le système d'exploitation gère le ou les processeurs ainsi que la mémoire. Il fait fonctionner les périphériques (clavier, souris, surface tactile, écran, disque dur, lecteur de DVD, lecteur de cartes mémoire...). Dans un appareil photo, il fait fonctionner les différents mécanismes, gère l'affichage de l'écran et détecte les actions de l'utilisateur. Etc.

Les systèmes d'exploitation comportent aussi l'interface avec l'utilisateur. Dans un ordinateur, par exemple, c'est lui qui affichera les fenêtres et présentera le contenu des unités de stockage (disque dur, CD, DVD...).

Exemples de systèmes d'exploitations

Dans le secteur informatique, les systèmes d'exploitation les plus répandus sont Windows (pour lesPC), Mac OS (pour les ordinateurs d'Apple), Linux (pour les PC et les serveurs) et Unix(pour les serveurs). Pour les téléphones, on trouve Android, iOS (chez Apple), Symbian et Windows Phone.

CHAPITRE 2 : BASE DES SYSTEMES INFORMATIQUES - NUMERATION ET CODAGE DE L'INFORMATION

INTRODUCTION

- L'ordinateur traite de l'information digitale ; le support de l'information est système à deux états d'équilibre 0 et 1 appelés bits.
- A n bits correspondent 2^n configurations binaires possibles différentes.
- Mot : séquence $q_{n-1} \ldots q_0$ de n bits ; q_{n-1} est appelé bit de poids fort, q_0 bit de poids faible.
- Octet : mot de huit bits.
- Code : correspondance biunivoque entre les informations à représenter et les configurations binaires possibles.

L'alphabet de l'ordinateur est binaire : tout ce qui est écrit (codé) en mémoire centrale, les données comme les programmes exécutables, est écrit sous la forme d'une succession des symboles 0 et 1.

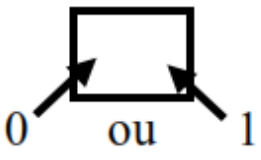
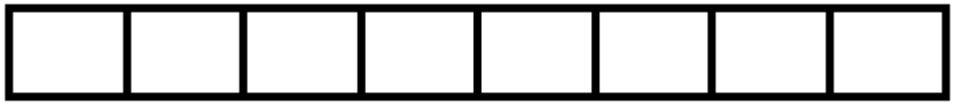


Figure 5-4 : un bit

BInart digiT (chiffre binaire)

Combien d'informations peut on écrire (coder) avec 1 bit ? 2 ? et avec 8 bits?



Exemple d'information codé sur 8 bits **1 1 0 0 1 0 1 1**

8 bits, 1 octet, 1 byte

1 ko = 2^{10} octets= 1024 octets

1 Mo = 2^{20} octets= 1048576 octets = 1024 ko

1 Go = 2^{30} octets = 1 073 741 824 octets = 1024 Mo

1 To = 2^{40} octets = 1 099 511 627 776 octets = 1024 Go

CODAGE DES ENTIERS NATURELS

Décomposition en binaire

En ne disposant que de deux symboles (0 et 1), la base 2 s'impose comme le seul système de numération adapté à la représentation interne des entiers naturels.

Corollaire 1 (Théorème 2) *Tout $a \in \mathbb{N}$, supérieur ou égal à 2, s'écrit de manière unique sous la forme :*

$$\begin{aligned} a &= q_{n-1}.2^{n-1} + \ldots + q_2.2^2 + q_1.2 + q_0 \\ &= \sum_{i=0}^{n-1} q_i.2^i \end{aligned}$$

avec $q_{n-1} = 1$ et $q_i \in \{0, 1\}$ pour $0 \leq i \leq n-2$.

EXEMPLE

L'entier 43_{10} se décompose en base 2 :

$$43_{10} = 1.2^5 + 0.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0$$

Autrement dit, 43_{10} sécrit en binaire 101011_2.

Entiers positifs ou nuls : On représente le nombre en base 2 et on range les bits comme pour les entiers naturels. Cependant, la cellule de poids fort est toujours à 1 : on utilise donc n-1 bits.

Le plus grand entier positif représentable sur n bits en relatif est donc $2^n - 1$

RESULTATS COMPLEMENTAIRES

Théorème 3 *Sur k bits, on peut représenter les entiers naturels a tels que $0 \leq a \leq 2^k - 1$.*

Preuve : Le plus grand entier représentable sur k bits est :

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1$$

Q.E.D.

- Sur un octet on compte de 0 à 255.
- Sur deux octets on compte de 0 à $32\,767$.
- Un mot de 32 bits permet de compter de 0 à $4\,294\,967\,295$.

Corollaire 2 *Pour tout entier naturel a, il existe un entier k strictement supérieur à 0, tel que $2^{k-1} \leq a < 2^k$.*

Remarque : La représentation binaire est malcommode à utiliser pour l'être humain : la longueur du codage et le manque de discrimination dû à l'utilisation de deux caractères seulement est facilement source d'erreurs...

Dans le système décimal, on écrit par exemple :

$$12,346 = 1\ 10^1 + 2\ 10^0 + 3\ 10^{-1} + 4\ 10^{-2} + 6\ 10^{-3}$$

En général, en base b, on écrit :

$$a_n a_{n-1} ... a_1 a_0, a_{-1} a_{-2} ... a{-p} = a_n b^n + a_{n-1} b^{n-1} + ... + a_0 b^0 + a_{-1} b^{-1} + ... + a_{-p} b^{-p}$$

NOMBRES FRACTIONNAIRES

Nombres fractionnaires On multiplie la partie fractionnaire par la base en répétant l'opération sur la partie fractionnaire du produit jusqu'a ce qu'elle soit nulle (ou que la précision voulue soit atteinte). Pour la partie entière, on procède par divisions comme pour un entier.

Exemple : conversion de (54, 25) en base 2
Partie entière : $(54_{10}) = (110110)_2$ par divisions.
Partie fractionnaire :

$$\begin{aligned} 0{,}25 \text{ x } 2 = 0{,}50 &\Longrightarrow a_{-1} = 0 \\ 0{,}50 \text{ x } 2 = 1{,}00 &\Longrightarrow a_{-2} = 1 \\ 0{,}00 \text{ x } 2 = 0{,}00 &\Longrightarrow a_{-3} = 0 \end{aligned}$$

$(54{,}25)_{10}$ = 110110,010

REPRESENTATION HEXADECIMALE

La numération hexadécimale représente le meilleur compromis pour tout système informatique qui utilise des mots dont le nombre de bits est divisible par 4.

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

DECOMPOSITION HEXADECIMALE

Corollaire 3 (Théorème 2) *Tout $a \in \mathbb{N}$, supérieur ou égal à 2, s'écrit de manière unique sous la forme :*

$$\begin{aligned} a &= q_{n-1}.16^{n-1} + \ldots + q_2.16^2 + q_1.16 + q_0 \\ &= \sum_{i=0}^{n-1} q_i.16^i \end{aligned}$$

avec

$$\begin{aligned} q_{n-1} &\in \{1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\} \quad \textit{et} \\ q_i &\in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\} \\ &\quad \textit{pour } 0 \leq i \leq n-2. \end{aligned}$$

EXEMPLE

L'entier 43_{10} se décompose en base 16 :

$$43_{10} = 2.16^1 + B.16^0$$

Autrement dit, 43_{10} s'écrit en héxadécimal $2B_{16}$.

CONVERSION D'UN ENTIER NATUREL

Deux méthodes :

1. la méthode des divisions successives ;
2. la méthode des soustractions successives.

METHODE DES DIVISIONS SUCCESSIVES

On a :

$$\begin{aligned} a &= q_{n-1}.b^{n-1} + q_{n-2}.b^{n-2} + \ldots + q_2.b^2 + q_1.b^1 + q_0 \\ &= \underbrace{(q_{n-1}.b^{n-2} + q_{n-2}.b^{n-3} + \ldots + q_2.b^1 + q_1)}_{a_1 \text{ (quotient)}}.b + \underbrace{q_0}_{\text{(reste)}} \end{aligned}$$

puis :

$$\begin{aligned} a_1 &= q_{n-1}.b^{n-2} + q_{n-2}.b^{n-3} + \ldots + q_2.b^1 + q_1 \\ &= \underbrace{(q_{n-1}.b^{n-3} + q_{n-2}.b^{n-4} + \ldots + q_2)}_{a_2 \text{ (quotient)}}.b + \underbrace{q_1}_{\text{(reste)}} \end{aligned}$$

et ainsi de suite...

ALGORYTHME

- l'entier décimal a est divisé par b, ce qui donne pour quotient a_1 et pour reste q_0 ;
- si $a_1 > 0$, il est divisé par b, ce qui donne pour quotient a_2 et pour reste q_1 ;
- l'opération de division mentionnée ci-dessus est renouvelée sur chacun des quotients obtenus jusqu'à obtenir un quotient $a_m = 0$;
- les restes obtenus successivement sont les chiffres de la représentation en base b de l'entier a, avec q_0 comme chiffre le moins significatif, q_1 comme chiffre suivant, et ainsi de suite.

EXEMPLE

Conversion du nombre 43_{10} en binaire :

$$\begin{aligned} 43 &= 21.2 + 1 \\ 21 &= 10.2 + 1 \\ 10 &= 5.2 + 0 \\ 5 &= 2.2 + 1 \\ 2 &= 1.2 + 0 \\ 1 &= 0.2 + 1 \end{aligned}$$

$43_{10} = 2.(2.(2.(2.(2.(2.0 + 1) + 0) + 1) + 0) + 1) + 1$ donc 43_{10} s'écrit en binaire 101011_2.

CAS DES BASES 2, 8 et 16

Ces bases correspondent à des puissances de 2 (2^1, 2^3 et 2^4) d'où des passages de l'une à l'autre très simples. Les bases 8 et 16 sont pour cela très utilisées en informatique, elles permettent de représenter **rapidement et de manière compacte des configurations binaires**.

La base 8 est appelée notation *octale*, et la base 16 notation *hexadécimale*.
Chaque chiffre en base 16 (2^4) représente un paquet de 4 bits consécutifs.

Le système octal utilise les huit symboles : 0, 1, 2, 3, 4, 5, 6, 7. Le passage de la base 2 à la base 8 se fait de façon immédiate en groupant les chiffres binaires 3 par 3. Ainsi :
$1011101{,}01101_2$ = 001 | 011 | 101,011 | 010 = $135{,}32_8$

La base 16 utilise les symboles 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
Le passage de la base 2 à la base 16 se fait de la même façon en groupant les chiffres binaires par bloc de 4 symboles. Ainsi :

$1011101{,}01101_2$ = 0101 | 1101,0110 | 1000 =
$5D, 68_{16}$

$$(10011011)_2 = (1001\ \ 1011)_2 = (9B)_{16}$$

OPERATIONS BINAIRES

- Addition
- Soustraction
- Multiplication
- Division
- Décalage logique à gauche
- Décalage logique à droite

TABLE D'ADDITION BINAIRE

$$
\begin{array}{lcl}
0+0 & = & 0 \\
0+1 & = & 1 \\
1+0 & = & 1 \\
1+1 & = & \underbrace{1}_{\textit{retenue}}\ 0
\end{array}
$$

Exemple :

$$
\begin{array}{rccccc}
 & & {}^{1}1 & {}^{1}1 & {}^{1}0 & 1 \\
+ & & & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 0
\end{array}
$$

TABLE DE SOUSTRACTION BINAIRE

$$
\begin{array}{lcl}
0-0 & = & 0 \\
1-0 & = & 1 \\
1-1 & = & 0 \\
0-1 & = & \underbrace{1}_{\textit{retenue}}\ 1
\end{array}
$$

Exemple :

$$
\begin{array}{rcccc}
 & 1 & 0 & 1 & 1 \\
- & {}^{1} & 1 & 0 & 1 \\
\hline
 & 0 & 1 & 1 & 0
\end{array}
$$

TABLE DE MULTIPLICATION BINAIRE

$$\begin{array}{|lcl|}\hline 0\times 0 & = & 0\\ 0\times 1 & = & 0\\ 1\times 0 & = & 0\\ 1\times 1 & = & 1\\ \hline\end{array}$$

Exemple :

$$\begin{array}{rcccccc} & & & 1 & 0 & 1 & 1\\ \times & & & & 1 & 0 & 1\\ \hline & & & 1 & 0 & 1 & 1\\ & & {}^{1}0 & 0 & 0 & 0 & \\ & 1 & 0 & 1 & 1 & & \\ \hline & 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

TABLE DE DIVISION BINAIRE

$$\begin{array}{|lcl|}\hline 0\div 0 & : & \text{indéfini}\\ 0\div 1 & = & 0\\ 1\div 0 & : & \text{indéfini}\\ 1\div 1 & = & 1\\ \hline\end{array}$$

Exemple :

$$\begin{array}{cccc|cc} 1 & 0 & 1 & 0 & 1 & 0\\ \cline{5-6} 1 & 0 & & & 1 & 0 & 1\\ \cline{1-2} 0 & 0 & 1 & & & & \\ & 0 & 0 & & & & \\ \cline{2-3} & 0 & 1 & 0 & & & \\ & & 1 & 0 & & & \\ \cline{3-4} & & 0 & 0 & & & \end{array}$$

Bases des systè

DEPASSEMENT DE CAPACITE

- Une retenue au delà du bit de poids fort produit un dépassement de capacité : sur quatre bits $7 + 12 = 19$ qui ne peut être représenté

REPRESENTATION DES ENTIERS RELATIFS – Notion de complément

Il faut ici coder le signe du nombre. On utilise le codage en ***complément à deux***, qui permet d'effectuer ensuite les opérations arithmétiques entre nombres relatifs de la même façon qu'entre nombres naturels.

- Sur n bits : 2^n configurations binaires
- Entiers signés $\rightarrow$ diviser cet ensemble de configurations par 2
- Une moitié des configurations pour les entiers positifs, l'autre moitié pour les entiers négatifs :

$$\underbrace{-2^{n-1} \longleftrightarrow 0 \longleftrightarrow 2^{n-1} - 1}_{2^n \text{ valeurs distinctes}}$$

- Donc $n - 1$ bits pour les entiers positifs, comme les entiers négatifs

BIT DE SIGNE

La distinction sur le signe découle du positionnement à 1 ou 0 du $n^{\text{ème}}$ bit de poids fort, appelé bit de signe :

$$\underbrace{q^{n-1}}_{\text{bit de signe}} \qquad \underbrace{q^{n-2} \ldots q^1 \; q^0}_{\text{valeur absolue } (n-1 \text{ bits})}$$

Bit de signe $= 0 \rightarrow$ entiers positifs
Bit de signe $= 1 \rightarrow$ entiers négatifs

CODAGE DES ENTIERS

- codage en signe et valeur absolue
- codage en complément à un
- codage en complément à deux

SIGNE ET VALEUR ABSOLUE

Distinction assurée seulement par le bit de signe

$$+1 \rightarrow 0001$$
$$-1 \rightarrow 1001$$

Problème : Deux représentations de zéro. . .

$$0^{+} \rightarrow 0000$$
$$0^{-} \rightarrow 1000$$

COMPLEMENT A UN OU COMPLEMENT RESTREINT Cr

On inverse les 1 et les 0

$$+1 \rightarrow 0001$$
$$-1 \rightarrow 1110$$

Problème : Deux représentations de zéro. . .

$$0^{+} \rightarrow 0000$$
$$0^{-} \rightarrow 1111$$

COMPLEMENT A DEUX OU COMPLEMENT VRAI Cv

Un entier négatif est tel qu'additionné à l'entier positif correspondant, on obtienne bien un zéro unique

$$\begin{array}{rcccc} & 0 & 0 & 0 & 1 \\ - & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 \end{array}$$

Codage de -1 :

$$\begin{array}{rcccc} & 0 & 0 & 0 & 0 \\ - & {}^{1}0 & {}^{1}0 & {}^{1}0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \end{array}$$

Cv = Cr +1

Exemple : soit à coder la valeur -2 sur 8 bits. On exprime 2 en binaire, soit 00000010. Le complément à 1 est 11111101. On ajoute 1 et on obtient le résultat : 1111 1110.

Remarques :

(a) le bit de poids fort d'un nombre négatif est toujours 1 ;

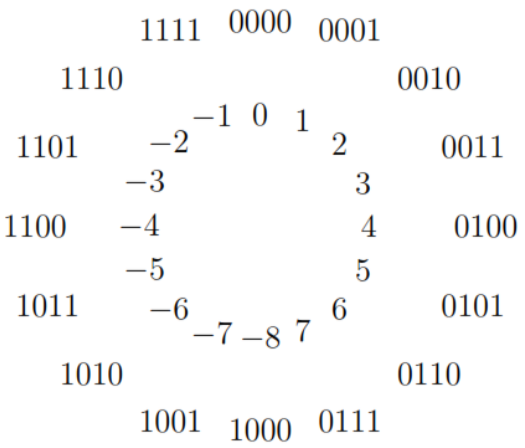
(b) sur n bits, le plus grand entier positif est

$+2^{n-1}-1= 011\ldots..1$;

(c) sur n bits, le plus petit entier négatif est

$-2^{n-1} = 10\ldots..0$

COMPLEMENT A DEUX SUR 4 BITS



REMARQUE : MAX +1 = MIN

Valeur décimale	Codage binaire
2	0000 0000 0000 0010
1	0000 0000 0000 0001
0	0000 0000 0000 0000
−1	1111 1111 1111 1111
−2	1111 1111 1111 1110

Table 2.2: Représentation complément à 2.

CALCUL DU COMPLEMENT A DEUX

Théorème 6 *Etant donné* $a \in [-2^{n-1}+1, 2^{n-1}-1]$, *la représentation de* $-a$ *en complément à* 2 *est obtenue à partir du complément à* 1 *de* a *auquel on ajoute* 1.

Exemple :

a	:		0	0	0	1
Complément à 1	:		1	1	1	0
ajoût de 1	:	+				1
-a	:		1	1	1	1

Il est important de bien comprendre comment se présentent les nombres dans ces divers formats afin de minimiser la place occupée sur le support mais également la place occupée en mémoire centrale et donc de la rapidité des traitements que l'on fera sur ces données. La représentation en virgule fixe occupe cependant une place importante quand on utilise de grands nombres et on lui préférera alors une autre forme de représentation dite virgule flottante.

DEPASSEMENT DE CAPACITE

Théorème 8 *L'addition UAL de deux entiers relatifs a et b fournit toujours un résultat correct :*

- *si a et b ne sont pas de mêmes signes*
- *si a et b sont de même signe, quand le bit de signe est égal à la retenue*

Exemple :

$$\begin{array}{rcl} 0111 & \rightarrow & 7 \\ + \quad 0010 & \rightarrow & 2 \\ \hline 1001 & \rightarrow & -7 \end{array}$$

LE CODAGE DES CARACTERES

Pour qu'une information (ou qu'un ensemble d'informations) soit acceptable par l'ordinateur, il faut que nous puissions la coder, la représenter sous la forme d'une série de nombres entiers. On sait qu'en réalité ces nombres entiers sont eux-mêmes écrits en binaire (et non en décimal).

Il est nécessaire que chacun des caractères utilisés trouve un numéro. Ceci a donné lieu à plusieurs systèmes de codage dont les plus connus sont le système ASCII et ANSI

- Code ASCII : données alphanumériques sur sept bits plus un bit de parité.
- Standard ISO 8859-1 (Latin 1) : évolution du code ASCII, huit bits pour représenter entre autres les caractères accentués.
- Unicode : deux octets pour coder des jeu de caractères non latins.

Le code **ASCII** (***American Standard Code for Information Interchange***) est l'un des codes les plus utilisés en informatique. Il a été défini aux USA en 1963 puis repris ensuite par les organismes de normalisation des transmissions internationales de données qui en ont fait le code ISO à 7 bits (International

Standard Organisation) ou code CCITT (Commission Consultation Internationale des Téléphones et Télécommunication).

Le code **ASCII** à 7 bits définit 128 combinaisons binaires différentes autorisant la codification de 128 caractères ou commandes. Le code **ASCII** est souvent assimilé à un code à 8 bits car on ajoute généralement aux 7 bits initiaux, un bit de contrôle (bit de parité) ASCII étendu = 8 bits.

Décimal	**Hexa**	**Binaire**	**Caractère**		**Décimal**	**Hexa**	**Binaire**	**Caractère**
0	0	00000000	NUL		32	20	00100000	ESPACE
1	1	00000001			33	21	00100001	!
2	2	00000010	STX		34	22	00100010	"
3	3	00000011	ETX		35	23	00100011	#
4	4	00000100	EOT		36	24	00100100	$
5	5	00000101			37	25	00100101	%
6	6	00000110	ACK		38	26	00100110	&
7	7	00000111	BEL		39	27	00100111	'
8	8	00001000			40	28	00101000	(
9	9	00001001			41	29	00101001	)
10	A	00001010	LF		42	2A	00101010	*
11	B	00001011			43	2B	00101011	+
12	C	00001100			44	2C	00101100	,
13	D	00001101	CR		45	2D	00101101	-
14	E	00001110			46	2E	00101110	.
15	F	00001111			47	2F	00101111	/
16	10	00010000			48	30	00110000	0
17	11	00010001			49	31	00110001	1

18	12	00010010			50	32	00110010	2
19	13	00010011			51	33	00110011	3
20	14	00010100	NAK		52	34	00110100	4
21	15	00010101			53	35	00110101	5
22	16	00010110			54	36	00110110	6
23	17	00010111			55	37	00110111	7
24	18	00011000			56	38	00111000	8
25	19	00011001			57	39	00111001	9
26	1A	00011010			58	3A	00111010	:
27	1B	00011011			59	3B	00111011	;
28	1C	00011100			60	3C	00111100	<
29	1D	00011101			61	3D	00111101	=
30	1E	00011110			62	3E	00111110	>
31	1F	00011111			63	3F	00111111	?
Décimal	**Hexa**	**Binaire**	**Caractère**		**Décimal**	**Hexa**	**Binaire**	**Caractère**
64	40	01000000	@		96	60	01100000	`
65	41	01000001	A		97	61	01100001	a
66	42	01000010	B		98	62	01100010	b
67	43	01000011	C		99	63	01100011	c
68	44	01000100	D		100	64	01100100	d
69	45	01000101	E		101	65	01100101	e
70	46	01000110	F		102	66	01100110	f
71	47	01000111	G		103	67	01100111	g
72	48	01001000	H		104	68	01101000	h

73	49	01001001	I		105	69	01101001	i
74	4A	01001010	J		106	6A	01101010	j
75	4B	01001011	K		107	6B	01101011	k
76	4C	01001100	L		108	6C	01101100	l
77	4D	01001101	M		109	6D	01101101	m
78	4E	01001110	N		110	6E	01101110	n
79	4F	01001111	O		111	6F	01101111	o
80	50	01010000	P		112	70	01110000	p
81	51	01010001	Q		113	71	01110001	q
82	52	01010010	R		114	72	01110010	r
83	53	01010011	S		115	73	01110011	s
84	54	01010100	T		116	74	01110100	t
85	55	01010101	U		117	75	01110101	u
86	56	01010110	V		118	76	01110110	v
87	57	01010111	W		119	77	01110111	w
88	58	01011000	X		120	78	01111000	x
89	59	01011001	Y		121	79	01111001	y
90	5A	01011010	Z		122	7A	01111010	z
91	5B	01011011	[		123	7B	01111011	
92	5C	01011100	\		124	7C	01111100	|
93	5D	01011101	]		125	7D	01111101	
94	5E	01011110	^		126	7E	01111110	~
95	5F	01011111	_		127	7F	01111111	

CHAPITRE 3 : NOTION D'ARCHITECTURE D'UN ORDINATEUR - PRESENTATION GENERALE DE L'ORDINATEUR

VARIABLE LOGIQUE

Un ordinateur ne manipule que des données binaires, on appelle donc variable logique une donnée binaire, c'est-à-dire une donnée ayant deux états possibles: 0 ou 1**.**

FONCTIONS LOGIQUES

On appelle «**fonction logique**» une entité acceptant plusieurs valeurs logiques en entrée et dont la sortie (il peut y en avoir plusieurs) peut avoir deux états possibles : 0 ou 1

PORTES LOGIQUES

Les fonctions logiques de bases sont appelées **portes logiques**. Il s'agit de fonctions ayant une ou deux entrées et une sortie:

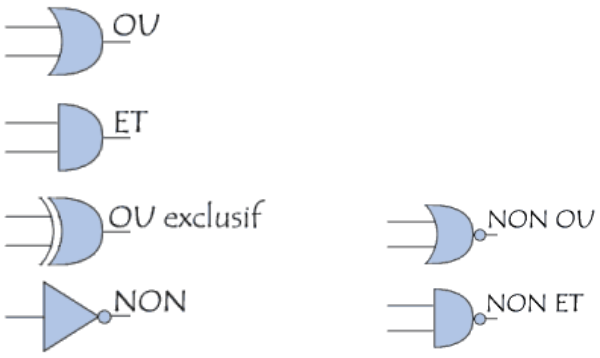
- La fonction **OU** (en anglais *OR*) positionne sa sortie à 1 si l'une ou l'autre de ses entrées est à 1
- La fonction **ET** (en anglais *AND*) positionne sa sortie à 1 si ses deux entrées sont à 1
- La fonction **OU EXCLUSIF** (en anglais *XOR*) positionne sa sortie à 1 si l'une ou l'autre de ses entrées est à 1 mais pas les deux simultanément
- La fonction **NON** (appelée aussi *inverseur*) positionne sa sortie à 1 si son entrée est à 0, et vice-versa
- On définit généralement les fonctions **NON OU** (couramment appelée *NOR*) et **NON ET** (*NAND*) comme étant la composition respective d'un NON avec un OU et un ET.

TABLE DE VERITE

Une **table de vérité** est un tableau permettant de décrire toutes les possibilités de sorties en fonction des entrées. On place donc les variables d'entrées dans les colonnes de gauche en les faisant varier de telle façon à couvrir l'ensemble des possibilités. La colonne (ou les colonnes si la fonction a plusieurs sorties) de droite décrit la sortie

Nom de la porte	**Entrée**		**Sortie**
	A	**B**	**S**
OU (OR)	0	0	0
	0	1	1
	1	0	1
	1	1	1
ET (**AND**)	0	0	0
	0	1	0
	1	0	0
	1	1	1
NON OU (**NOR**)	0	0	1
	0	1	0
	1	0	0
	1	1	0
NON ET (**NAND**)	0	0	1
	0	1	1
	1	0	1
	1	1	0
NON	0		1
	1		0

REPRESENATTION DES PORTES LOGIQUES



CIRCUITS LOGIQUES

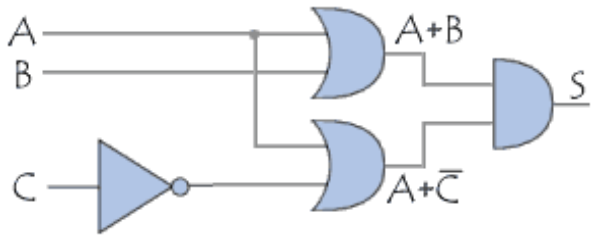
- Un circuit logique est un circuit qui ne manipule que deux valeurs logiques : 0 et 1
- À l'intérieur des circuits, on représente typiquement un état 0 par un signal de basse tension (proche de 0V) et un état 1 par un signal de haute tension (5V, 3,3V, 2,5V, 1,8V ou 0,9V selon les technologies)
- De minuscules dispositifs électroniques, appelées « portes », peuvent calculer différentes fonctions à partir de ces signaux

REALISATION DE CIRCUITS LOGIQUES

On appelle circuit logique (ou *circuit combinatoire*) un ensemble de portes logiques reliées entre elles pour répondre à une expression algébrique. Il s'agit donc d'aller transcrire en schéma électrique (à l'aide des représentations ci-dessus) l'expression algébrique que l'on a simplifiée grâce aux lois de composition.

Par exemple l'expression algébrique

(A+B).(A+/C)
sera schématisée comme suit:



- Quelle que soit la technologie, les portes NAND et NOR nécessitent moins de transistors que les portes AND et OR, qui nécessitent un inverseur en plus
- Les circuits des ordinateurs sont donc plutôt construits avec des portes NAND et NOR
  - Ces portes ont parfois plus de deux entrées, mais en ont rarement plus de 8 (problème de « fan-in/out »)
  - Les portes NAND et NOR sont dites « complètes », car tout circuit peut être implanté uniquement au moyen de l'un de ces types de portes

ALGEBRE DE BOOLE

- Pour décrire les circuits réalisables en combinant des portes logiques, on a besoin d'une algèbre opérant sur les variables 0 et 1
- Algèbre booléenne
  - G. Boole : 1815 – 1864
  - Algèbre binaire étudiée par Leibniz dès 1703

FONCTIONS BOOLEENNES

- Une fonction booléenne à une ou plusieurs variables est une fonction qui renvoie une valeur ne dépendant que de ces variables
- La fonction NON est ainsi définie comme :
  - f(A) = 1 si A = 0
  - f(A) = 0 si A = 1

ORDINATEUR ET LOGICIEL

- Les technologies numériques sont maintenant omniprésentes
  - Elles sont le moteur et l'objet de ce qu'on appelle la « révolution numérique »
- Elles sont basées sur l'interaction entre :
  - Des programmes, aussi appelés logiciels, décrivant des processus de traitement de l'information : biens immatériels
  - Des ordinateurs, capables d'exécuter ces programmes : biens matériels

REPRESENTATION DE L'INFORMATION

- L'information est représentée au sein des composants de l'ordinateur sous forme de différents états de la matière :
  - « Trou » ou « pas trou » sur la surface d'un cédérom ou DVD
  - Orientation nord ou sud d'un matériau magnétique
  - Lumière ou absence de lumière émise par un laser
  - Courant électrique ou non
- Ce sont souvent des représentations à deux états, c'est-à-dire « binaires »

CONSTITUANTS ELEMENTAIRES

- Presque tous les ordinateurs sont construits à base de circuits électroniques
- Les circuits électroniques sont réalisés au moyen de transistors
  - Composant élémentaire, dont le courant de sortie dépend de deux valeurs d'entrée
    - Un transistor a donc trois « pattes »
      - Appelées : base, émetteur et collecteur
    - Analogue à un « robinet à électricité » : plus il arrive de courant sur la base, plus le courant circule de l' émetteur vers le collecteur

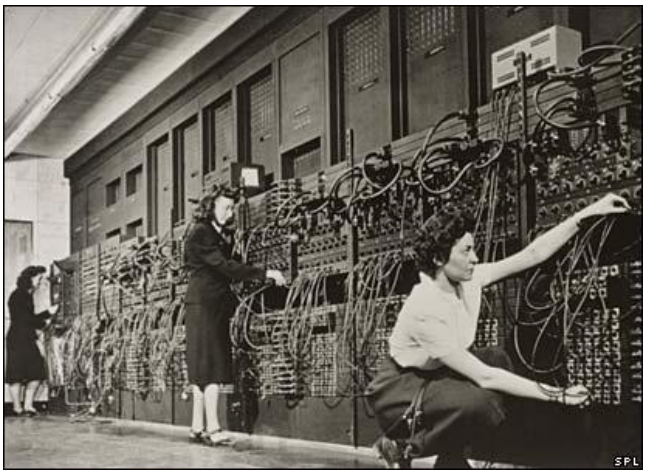
- Dans les ordinateurs, on utilise les transistors en mode saturé, c'est-à-dire « tout ou rien »
  - Fonctionnement analogue à celui d'un interrupteur
    - Robinet fermé ou ouvert en grand
    - Soit le courant passe, soit il ne passe pas du tout
  - Représentation des valeurs binaires « 0 » et « 1 »
- En combinant plusieurs transistors, on peut effectuer des calculs complexes
  - Sur la base de montages en série ou en parallèle
- Regroupement au sein de « circuits intégrés »

PERFORMANCE

- Les calculs des ordinateurs sont cadencés par une horloge
  - Plus la fréquence de l'horloge est élevée, et plus l'ordinateur pourra effectuer d'opérations par seconde (s'il n'est pas ralenti par autre chose...)
  - On mesure la fréquence d'une horloge en Hertz (Hz)
    - Nombre de battements par seconde
      - 1 kHz (kilo-Hertz) = 10^3 Hz
      - 1 MHz (méga-Hertz) = 10^6 Hz
      - 1 GHz (giga-Hertz) = 10^9 Hz
      - 1 THz (téra-Hertz) = 10^{12} Hz
- En fait, ce qui importe aux usagers, c'est le nombre d'opérations (plus généralement, « d'instructions ») qu'un ordinateur est capable d'effectuer par seconde
  - On la mesure en MIPS, pour « millions d'instructions par seconde »
- On pense souvent que la puissance d'un ordinateur dépend de sa fréquence de fonctionnement
  - C'est loin d'être toujours vrai !

EVOLUTIONS ARCHITECTURALES

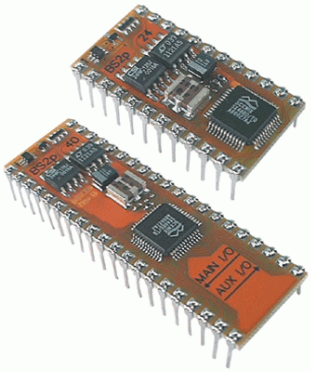
- 1946 : Ordinateur ENIAC
  - Architecture à base de lampes et tubes à vide : 30 tonnes, 170 m2 au sol, 5000 additions par seconde
  - 0,005 MIPS, dirons-nous...
- 1947 : Invention du transistor
- 1958 : Invention du circuit intégré sur silicium
  - Multiples transistors agencés sur le même substrat



ELECTRONIC NUMERICAL INTEGRATOR AND COMPUTER (ENIAC)

- 1971 : Processeur Intel 4004
  - 2300 transistors dans un unique circuit intégré
  - Fréquence de 740 kHz, 0,092 MIPS
- …40 ans d'une histoire très riche...
- 2011 : Processeur Intel Core i7 2600K
  - Plus de 1,4 milliards de transistors
  - Fréquence de 3,4 GHz
  - 4 cœurs, 8 threads
  - 128300 MIPS

Exemple de circuits intégrés



- Entre le 4004 et le Core i7 2600K :
  - La fréquence a été multipliée par 4600
  - La puissance en MIPS a été multipliée par 1,4 million
- La puissance d'un ordinateur ne dépend clairement pas que de sa fréquence !
- Intérêt d'étudier l'architecture des ordinateurs pour comprendre :
  - Où les gains se sont opérés
  - Ce qu'on peut attendre dans le futur proche

BARRIERE DE LA CHALEUR

- Plus on a de transistors par unité de surface, plus on a d'énergie à évacuer
- La dissipation thermique évolue de façon proportionnelle à V^2*F
  - La tension de fonctionnement des circuits a été abaissée
    - De 5V pour les premières générations à 0,9V maintenant
  - Il n'est plus vraiment possible de la diminuer avec les technologies actuelles
    - Le bruit thermique causerait trop d'erreurs
- La fréquence ne peut raisonnablement augmenter au delà des 5 GHz
- La tendance est plutôt à la réduction
  - « *Green computing* »
  - On s'intéresse maintenant à maximiser le nombre d'opérations par Watt
  - Mais on veut toujours plus de puissance de calcul !

- À surface constante, le nombre de transistors double tous les 2 ans
  - « Loi de Moore », du nom de Gordon Moore, co-fondateur d'Intel, énoncée en 1965
  - Diminution continuelle de la taille de gravage des transistors et circuits sur les puces de silicium
    - On grave actuellement avec un pas de 20 nm
- Limites atomiques bientôt atteintes...
  - Donc plus possible d'intégrer plus
  - Mais on veut toujours plus de puissance de calcul !

BARRIERE DE LA COMPLEXITE

- Que faire de tous ces transistors ?
  - On ne voit plus trop comment utiliser ces transistors pour améliorer individuellement les processeurs
  - Des processeurs trop complexes consomment trop d'énergie sans aller beaucoup plus vite
- Seule solution actuellement : faire plus de processeurs sur la même puce !
  - Processeurs bi-cœurs, quadri-coeurs, octo-cœurs, … déjà jusqu'à 128 cœurs !
  - Mais comment les programmer efficacement ?!

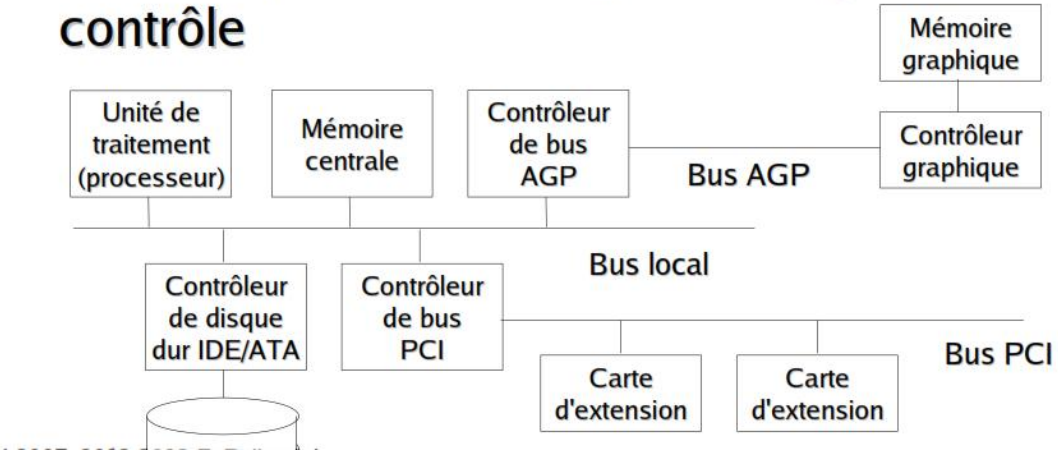
NOTION D'ARCHITECTURE DES ORDINATEURS

- L'architecture des ordinateurs a été l'un des secteurs de l'informatique qui a fait le plus de progrès
- Les ordinateurs d'aujourd'hui sont très complexes
  - 1,4 milliards de transistors pour le Core i7 8 cœurs
- Nécessité d'étudier leur fonctionnement à différents niveaux d'abstraction
  - Du composant au module, du module au système
  - Multiples niveaux de hiérarchie

- Un ordinateur est une machine programmable de traitement de l'information
- Pour accomplir sa fonction, il doit pouvoir :
  - Acquérir de l'information de l'extérieur
  - Stocker en son sein ces informations
  - Combiner entre elles les informations à sa disposition
  - Restituer ces informations à l'extérieur

- L'ordinateur doit donc posséder :
  - Une ou plusieurs unités de stockage, pour mémoriser le programme en cours d'exécution ainsi que les données qu'il manipule
  - Une unité de traitement permettant l'exécution des instructions du programme et des calculs sur les données qu'elles spécifient
  - Différents dispositifs « périphériques » servant à interagir avec l'extérieur : clavier, écran, souris, carte graphique, carte réseau, etc.

- Les constituants de l'ordinateur sont reliés par un ou plusieurs bus, ensembles de fils parallèles servant à la transmission des adresses, des données, et des signaux de contrôle



UNITE DE TRAITEMENT

- L'unité de traitement (ou CPU, pour « *Central Processing Unit* »), aussi appelée « processeur », est le cœur de l'ordinateur
- Elle exécute les programmes chargés en mémoire centrale en extrayant l'une après l'autre leurs instructions, en les analysant, et en les exécutant

- L'unité de traitement est composé de plusieurs sous-ensembles distincts
  - L'unité de contrôle, qui est responsable de la recherche des instructions à partir de la mémoire centrale et du décodage de leur type
  - L'unité arithmétique et logique (UAL), qui effectue les opérations spécifiées par les instructions
  - Un ensemble de registres, zones mémoires rapides servant au stockage temporaire des données en cours de traitement par l'unité centrale

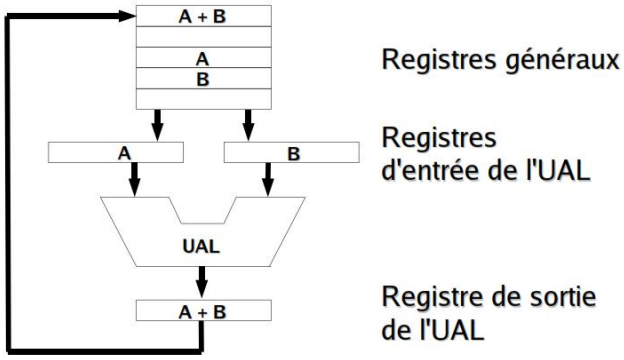
LES REGISTRES

- Chaque registre peut stocker une valeur entière distincte, bornée par la taille des registres (nombre de bits)
- Certains registres sont spécialisés, comme :
  - le compteur ordinal (« *program counter* ») qui stocke l'adresse de la prochaine instruction à exécuter
  - le registre d'instruction (« instruction register »), qui stocke l'instruction en cours d'exécution
  - l'accumulateur, registre résultat de l'UAL, etc.

LE CHEMIN DES DONNEES

- Le chemin de données représente la structure interne de l'unité de traitement
  - Comprend les registres, l'UAL, et un ensemble de bus internes dédiés
  - L'UAL peut posséder ses propres registres destinés à mémoriser les données d'entrées afin de stabiliser leurs signaux pendant que l'UAL calcule
- Le chemin des données conditionne fortement la puissance des machines
  - Pipe-line, superscalarité, …

- Chemin de données d'une machine de type « Von Neumann »



EXECUTION D'UNE INSTRUCTION

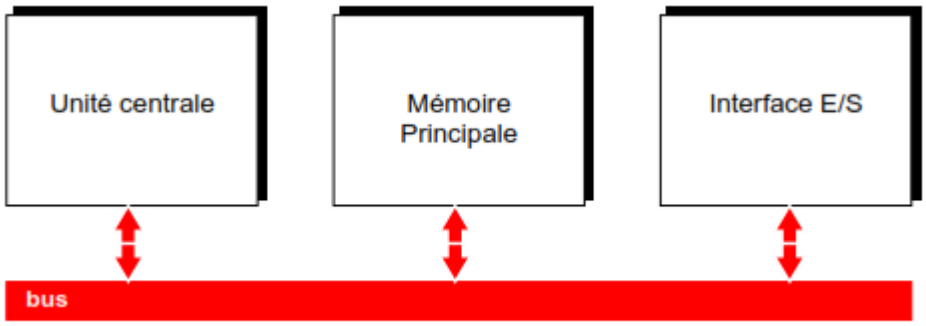
- L'exécution d'une instruction par l'unité centrale s'effectue selon les étapes suivantes :
  1. Charger la prochaine instruction à exécuter depuis la mémoire vers le registre d'instruction
  2. Décoder (analyser) l'instruction venant d'être lue
  3. Faire pointer le compteur ordinal vers l'instruction suivante (y compris dans le cas de branchements)
  4. Localiser en mémoire les données nécessaires
  5. Charger si nécessaire les données dans l'UAL
  6. Exécuter l'instruction, puis recommencer

Modèle deVon Neumann

Pour traiter une information, un microprocesseur seul ne suffit pas, il faut l'insérer au sein d'un système minimum de traitement programmé de l'information. John Von Neumann est à l'origine d'un modèle de machine universelle de traitement programmé de l'information (1946). Cette architecture sert de base à la plupart des systèmes à microprocesseur actuel. Elle est composée des éléments suivants :

- une unité centrale
- une mémoire principale
- des interfaces d'entrées/sorties

Les différents organes du système sont reliés par des voies de communication appelées **bus.**



L'UNITE CENTRALE

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

On caractérise le microprocesseur par :

– sa fréquence d'horloge : GHz
– le nombre d'instructions par secondes qu'il est capable d'exécuter : en MIPS
– la taille des données qu'il est capable de traiter : en bits

Le processeur ou microprocesseur

Le microprocesseur (CPU) permet à l'ordinateur d'effectuer les opérations (calculs) demandés.

Le processeur renferme deux unités fonctionnelles : l'unité de commande et l'unité de traitement

L'Unité de Traitement ou Unité Arithmétique et Logique (UAL)

L'unité de traitement assure l'exécution des opérations élémentaires "désignées" par l'unité de commande ("indicatif de traitement"). Les informations manipulées et les résultats intermédiaires sont rangés dans des éléments de mémorisation internes au processeur appelés registres.

Aucune opération ne se fait directement avec les cellules mémoire : celles-ci sont recopiées dans des registres, visibles ou non du programmeur, avant d'être traitées.

L'unité de commande, l'unité de traitement et les registres sont reliées entre eux par des connexions qui permettent le chargement ou la lecture en parallèle de tous leurs bits. Ces voies de communication sont appelées bus interne et sont aussi désignées par chemin de données

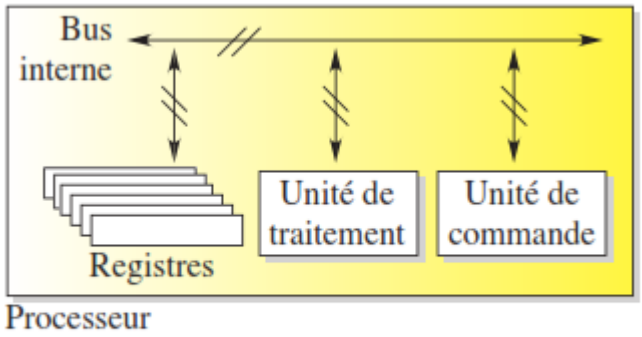


Figure 3.4 : Bus interne

L'unité de commande

L'unité de commande remplit trois fonctions. Elle "va chercher" les instructions en mémoire (fetch) :

1. Elle positionne assurant que le contenu de cette adresse est bien stable sur le bus de données (ou le bus d'instruction sur les machines disposant de bus distincts), elle charge dans un registre le code instruction ainsi obtenu.
2. Elle reconnaît l'instruction, opération désigne par décodage (décode),

3. Elle “indique” `a l’unité de traitement quels traitements arithmétiques ou opérations logiques il faut effectuer. C’est ce que l’on appelle le séquencement (execute).

L’unité de commande émet des signaux de commande (encore appelées microcommandes) pour assurer le déroulement des instructions reconnues après décodage.

L’unité de commande est composée de quatre modules : registre instruction, compteur ordinal, décodeur et séquenceur.

Les registres du processeur

Les registres de travail permettent de ranger des résultats intermédiaires sans accéder à la mémoire. La manipulation des données est ainsi simplifiée (on n’a pas à gérer d’adresse) et le temps d’accès aux informations est plus court. Dans la plupart des processeurs à architecture classique, le nombre des registres est de l’ordre de 8 à 32.

Registres accumulateurs
Pointeurs de pile
Registre des drapeaux
Registres d’adressage

LA MEMOIRE PRINCIPALE

Elle contient les instructions du ou des programmes en cours d’exécution et les données associées à ce programme. Physiquement, elle se décompose souvent en :
– une mémoire morte (**ROM** = Read Only Memory) chargée de stocker le programme. C’est une mémoire à lecture seule.
– une mémoire vive (**RAM** = Random Access Memory) chargée de stocker les données intermédiaires ou les résultats de calculs. On peut lire ou écrire des données dedans, ces données sont perdues à la mise hors tension.

Généralités sur les mémoires

Définition: Une mémoire est un circuit à semi-conducteur permettant d’enregistrer, de conserver et de restituer des informations (instructions et variables).
Il y a écriture lorsqu'on enregistre des informations en mémoire, lecture lorsqu'on récupère des informations précédemment enregistrées.

Organisation d'une mémoire

Une mémoire peut être représentée comme une armoire de rangement constituée de différents tiroirs. Chaque tiroir représente alors une case mémoire qui peut contenir un seul élément : des **données**. Le nombre de cases mémoires pouvant être très élevé, il est alors nécessaire de pouvoir les identifier par un numéro. Ce numéro est appelé **adresse**. Chaque donnée devient alors accessible grâce à son adresse.

Adresse	Case mémoire
7 = 111	
6 = 110	
5 = 101	
4 = 100	
3 = 011	
2 = 010	
1 = 001	
0 = 000	0001 1010

Caractéristiques d'une mémoire

- **La capacité :**

C'est le nombre total de bits que contient la mémoire. Mais elle s'exprime souvent en Mo ou Go

- **Le format des données :**

C'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.

- **Le temps d'accès :**

C'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.

- **Le temps de cycle :**

Il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.

- **Le débit :**

C'est le nombre maximum d'informations lues ou écrites par seconde.

- **Volatilité :**

Elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

Les différents types de mémoire

Les mémoires vives (RAM)

Une mémoire vive sert au stockage temporaire de données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure d'alimentation. Certaines d'entre elles, ayant une faible consommation, peuvent être rendues non volatiles par l'adjonction d'une batterie. Il existe deux grandes familles de mémoires RAM (Random Acces Memory : mémoire à accès aléatoire) :

- Les RAM statiques SRAM
- Les RAM dynamiques DRAM

DRAM

Les **mémoires dynamiques** (**DRAM**, *Dynamic Random Access Module*), peu coûteuses. Elles sont principalement utilisées pour la mémoire centrale de l'ordinateur. Dans les RAM dynamiques l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur.

Avantages :

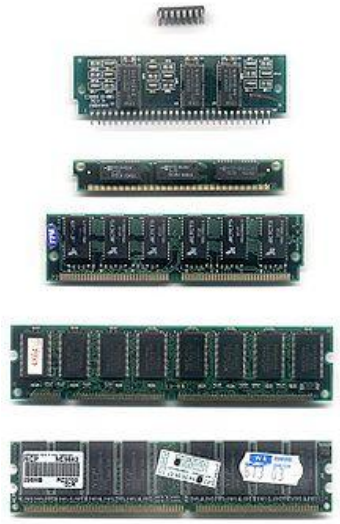
Cette technique permet une plus grande densité d'intégration, car un point mémoire nécessite environ quatre fois moins de transistors que dans une mémoire statique. Sa consommation s'en retrouve donc aussi très réduite.

- **Inconvénients :**

La présence de courants de fuite dans le condensateur contribue à sa décharge. Ainsi, l'information est perdue si on ne la régénère pas périodiquement (charge du condensateur). Les RAM dynamiques doivent donc être rafraîchies régulièrement pour entretenir la mémorisation : il s'agit de lire l'information et de la recharger. Ce rafraîchissement indispensable a plusieurs conséquences :

- il complique la gestion des mémoires dynamiques car il faut tenir compte des actions de rafraîchissement qui sont prioritaires.
- la durée de ces actions augmente le temps d'accès aux informations.

D’autre part, la lecture de l’information est destructive. En effet, elle se fait par décharge de la capacité du point mémoire lorsque celle-ci est chargée. Donc toute lecture doit être suivie d’une réécriture.



SRAM

Les **mémoires statiques** (**SRAM**, *Static Random Access Module*), rapides et onéreuses. Les SRAM sont notamment utilisées pour les mémoires cache du processeur

Les mémoires mortes (ROM)

Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires sont non volatiles.

Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM :

- PROM
- EPROM
- EEPROM
- FLASH EPROM.

La ROM Elle est programmée par le fabricant et son contenu ne peut plus être ni modifié, ni effacé par l'utilisateur.

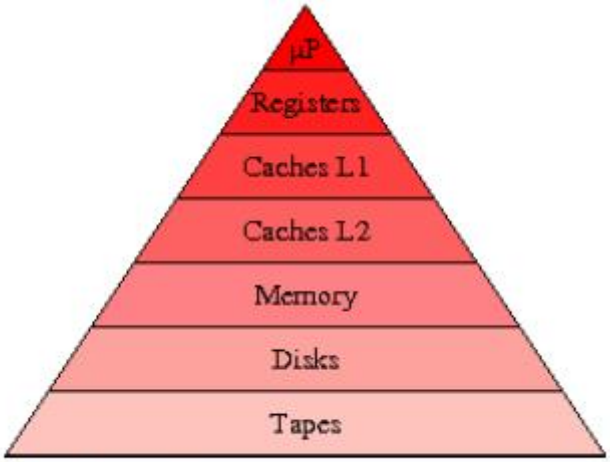
- **Avantages :**

Densité élevée
Non volatile
Mémoire rapide

- **Inconvénients :**

Écriture impossible
Modification impossible (toute erreur est fatale).
Délai de fabrication (3 à 6 semaines)
Obligation de grandes quantités en raison du coût élevé qu'entraîne la production du masque et le processus de fabrication.

Notion de hiérarchie mémoire



Hiérarchie de mémoire

Autres mémoires

Les registres sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.

- **La mémoire cache** est une mémoire rapide de faible capacité destinée à accélérer l'accès à la mémoire centrale en stockant les données les plus utilisées.
- **La mémoire d'appui** sert de mémoire intermédiaire entre la mémoire centrale et les mémoires de masse. Elle joue le même rôle que la mémoire cache.
- **La mémoire de masse** est une mémoire périphérique de grande capacité utilisée pour le stockage permanent ou la sauvegarde des informations.

LES BUS

Un bus est un ensemble de fils qui assure la transmission du même type d'information. On retrouve trois types de bus véhiculant des informations en parallèle dans un système de traitement programmé de l'information :

Caractéristiques d'un bus

Un bus est caractérisé par le volume d'informations transmises simultanément. Ce volume, exprimé en bits, correspond au nombre de lignes physiques sur lesquelles les données sont envoyées de manière simultanée. Une nappe de 32 fils permet ainsi de transmettre 32 bits en parallèle. On parle ainsi de « **largeur** » pour désigner le nombre de bits qu'un bus peut transmettre simultanément.

D'autre part, la vitesse du bus est également définie par sa **fréquence** (exprimée en Hertz), c'est-à-dire le nombre de paquets de données envoyés ou reçus par seconde. On parle de **cycle** pour désigner chaque envoi ou réception de données.

De cette façon, il est possible de connaître le **débit** maximal du bus (ou *taux de transfert maximal*), c'est-à-dire la quantité de données qu'il peut transporter par unité de temps, en multipliant sa largeur par sa fréquence. Un bus d'une largeur de 16 bits, cadencé à une fréquence de 133 MHz possède donc un débit égal à :

$16 * 133.10^6 = 2128*10^6$ bit/s,

soit $2128*10^6/8 = 266*10^6$ octets/s

soit $266*10^6 /1024 = 259.7*10^3$ Ko/s

soit $259.7*10^3 /1024 = 253$ Mo/s

Les bus, système de communication entre les composants d'un ordinateur.

Il(s) permet(tent) de connecter les différentes parties fonctionnelles de cet

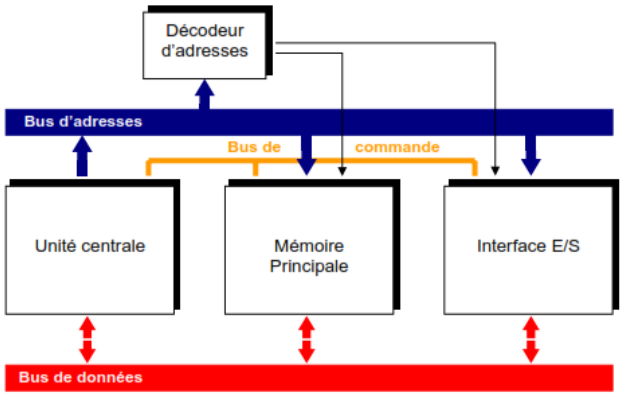
ordinateur entre elles.

En informatique, un bus permet le transport des informations entre différents composants d'un ordinateur. Il peut par exemple servir à relier le microprocesseur à la mémoire centrale, aux dispositifs de stockage ou aux périphériques. D'un point de vue technique, il est constitué d'un ensemble de fils (i.e. une nappe) ou de « pistes » sur un circuit intégré. Chacune de ces pistes permet de véhiculer une information en parallèle. Par conséquent, leur nombre influe sur la vitesse de transfert des données entre les composants de l'ordinateur. C'est pour cela qu'on parle par exemple de processeurs en 32 ou 64 bits (un bit est un nombre binaire, c'est-à-dire une information élémentaire pouvant circuler sur l'un de ces fils à un moment donné).
On retrouve trois types de bus véhiculant des informations en parallèle dans un système de traitement programmé de l'information :

- **un bus de données** : bidirectionnel qui assure le transfert des informations entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal à la capacité de traitement du microprocesseur.

- **un bus d'adresses**: unidirectionnel qui permet la sélection des informations à traiter dans un *espace mémoire* (ou *espace adressable*) qui peut avoir 2 emplacements, avec n = nombre de conducteurs du bus d'adresses.

- **un bus de commande**: constitué par quelques conducteurs qui assurent
la synchronisation des flux d'informations sur les bus des données et des adresses.



PRESENTATION DES COMPOSANTS D'UN ORDINATEUR

Les principaux éléments connectés è la carte mère

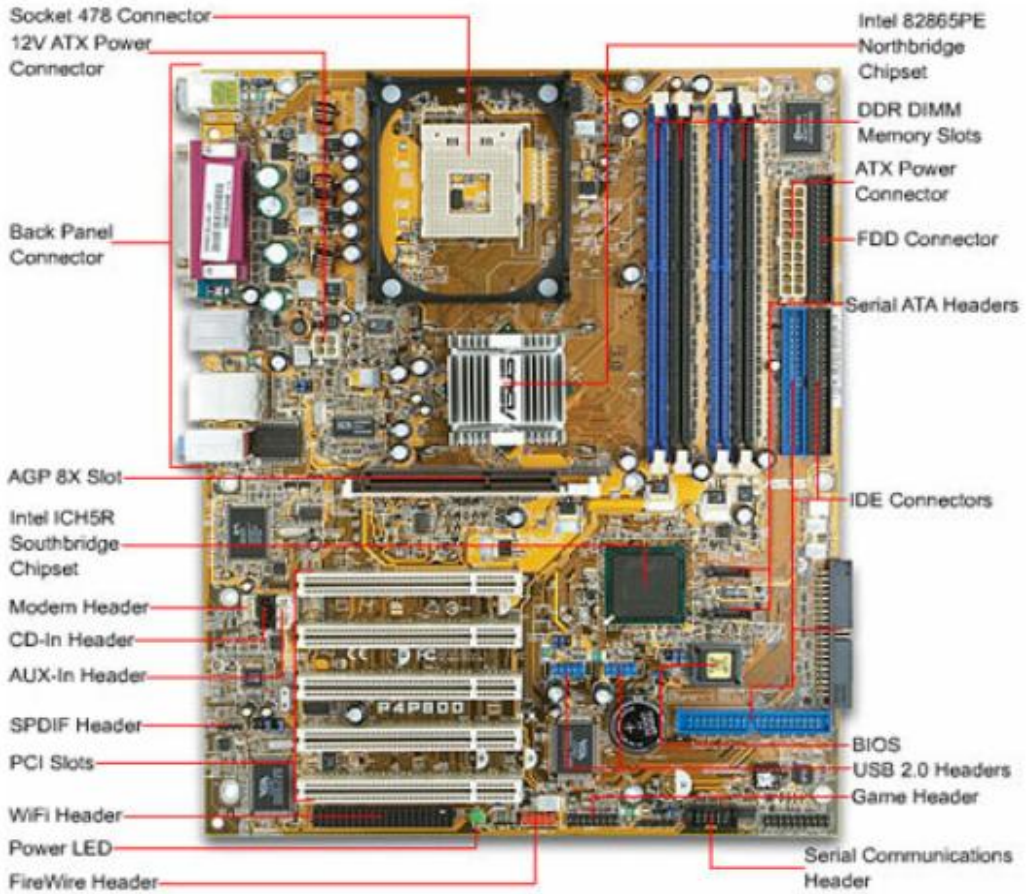
– La carte mère : qui relie tous les éléments constituant un ordinateur

La principale fonction d'une carte mère est la mise en relation de ces composants par des bus sous forme de circuits imprimés. Elle comporte notamment des emplacements (ou « slots »), prévus pour accepter différents types de composants. Il y a par exemple un ou plusieurs emplacements prévus pour le(s) processeur(s), pour les barrettes mémoire, et des emplacements génériques pour les périphériques.

La carte mère est l'un des éléments essentiels d'un ordinateur. Elle assure la connexion physique des différents composants (processeur, mémoire, carte d'entrées/sorties, ...) par l'intermédiaire de différents bus (adresses, données et commande). Plusieurs technologies de bus peuvent se côtoyer sur une même carte mère. La qualité de la carte mère est vitale puisque la performance de l'ordinateur dépend énormément d'elle. On retrouve toujours sur une carte mère/

le chipset :

C'est une interface d'entrée/sortie. Elle est constituée par un jeu de plusieurs composants chargé de gérer la communication entre le microprocesseur et les périphériques. C'est le lien entre les différents bus delà carte mère.



Le BIOS (Basic Input Ouput Service)**:**

C'est un programme responsable de la gestion du matériel : clavier, écran, disques durs, liaisons séries et parallèles, etc... Il est sauvegardé dans une mémoire morte (EEPROM) et agit comme une interface entre le système d'exploitation et le matériel.

l'horloge :

Elle permet de cadencer le traitement des instructions par le microprocesseur ou la transmission des informations sur les différents bus.

les ports de connexion :

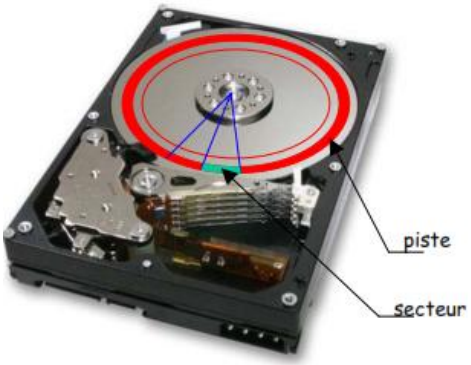
Ils permettent de connecter des périphériques sur les différents bus de la carte mère. Il existe des ports « internes » pour connecter des cartes d'extension (PCI, ISA, AGP) ou des périphériques de
stockage (SCSI, IDE, Serial ATA) et des ports « externes » pour connecter d'autres périphériques (série, parallèle, USB, firewire, etc …)

Le socket :
C'est le nom du connecteur destiné au microprocesseur. Il détermine le type de microprocesseur que l'on peut connecter.

INTERFACES D'E/S

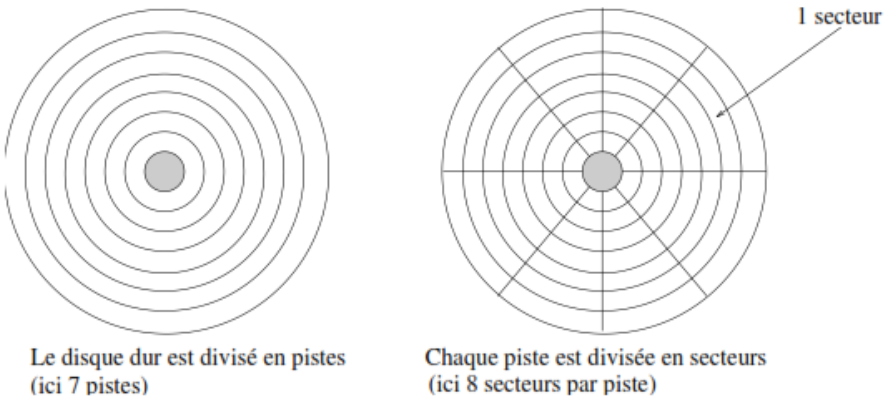
Les mémoires secondaires

- **Le disque dur**, qui stocke les informations des programmes et données de base de la machine.



Principe :

Le disque dur est constitué de plusieurs plateaux empilés, entre lesquels se déplace un bras comptant plusieurs têtes de lecture. Chaque plateau est recouvert d'une surface magnétique sur ses deux faces et tourne à une vitesse comprise entre 4000 et 15000 tr/min. La tête de lecture/écriture est composée par un aimant autour duquel est enroulée une bobine. Pour écrire, on fait passer un courant électrique dans la bobine ce qui crée un champ magnétique. Les lignes de champ magnétique traversent la couche d'oxyde et orientent celui-ci en créant de petits aimants dont le sens est donné par le sens du courant dans la bobine.
Pour lire, on fait passer la tête de lecture/écriture sur le support magnétisé qui crée un courant induit dans la bobine dont le sens indique s'il s'agit d'un 0 ou d'un 1.



Exemple de fiche technique d'un ordinateur

Caractéristiques techniques Hitachi Travelstar 7K1000 1 To

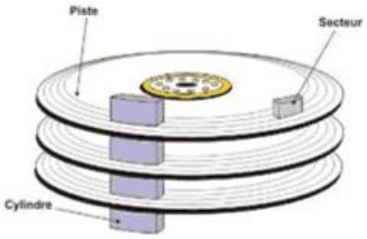
Capacité : **1 To**

- Vitesse de rotation : **7200 tpm**
- Mémoire tampon : **32 Mo**
- Interface : **Serial-ATA 6 Gb/s**
- Technologie **deux plateux**
- Consommation (lecture/écriture) : **1.8W**
- Hauteur du disque **: 9.5mm**
- Densité par plateau **: 500 Go**
- Temps de latence moyen **: 4.2 ms**
- Temps d'accès moyen **: 12 ms**
- Applications : PC portables, Consoles de jeux, Disques durs externes...

Le formatage :

Le formatage de *bas niveau* permet d'organiser la surface du disque en éléments simples (**pistes** et **secteurs**) qui permettront de localiser l'information. Le nombre total de pistes dépend du type de disque. Il est effectué en usine lors de la fabrication du disque. Chaque piste est découpée en secteurs. Toutefois l'unité d'occupation d'un disque n'est pas le secteur, trop petit pour que le système puisse en tenir compte. On utilise alors un groupe d'un certain nombre de secteurs (de 1 à
16) comme unité de base. Ce groupe est appelé Bloc ou Cluster. C'est la taille minimale que peut occuper un fichier sur le disque. Pour accéder à un secteur donné, il faudra donc déplacer l'ensemble des bras et attendre ensuite que ce secteur se positionne sous les têtes. L'accès à un bloc est aléatoire alors que l'accès à un secteur est séquentiel.
Une autre unité de lecture/écriture est le **cylindre**. Un cylindre est constitué par toutes les pistes superposées verticalement qui se présentent simultanément sous les têtes de lecture/écriture.

En effet, il est plus simple d'écrire sur les mêmes pistes des plateaux superposés que de déplacer à nouveau l'ensemble des bras.



La défragmentation :

A mesure que l'on stocke et supprime des fichiers, la répartition des fichiers sur les différents clusters est modifiée. L'idéal, pour accéder rapidement à un fichier, serait de pouvoir stocker un fichier sur des clusters contigus sur le même cylindre. La défragmentation permet de réorganiser le stockage des fichiers dans les clusters pour optimiser la lecture.

Les caractéristiques :

- capacité en To
- vitesse de rotation en tours minutes
- temps d'accès exprimé en millisecondes
- interface (IDE, SCSI, SATA)
- taux de transfert moyen exprimé en Mo par seconde

A noter que les disques durs actuels sont équipés de cache mémoire afin de diminuer les temps d'accès**.**

Et/ou le SSD

Le disque dur était encore il y a peu indispensable au fonctionnement de l'ordinateur, on peut aujourd'hui le remplacer par un **SSD**. Les disques SSD pour (Solid-state drive) permettent de stocker des données tout comme le fait un disque dur mais leur conception et leurs caractéristiques sont différentes

Clé USB : On peut également ajouter des périphériques d'entrée-sortie qui opèrent dans les deux sens : un lecteur de CD-ROM ou une clé USB, par exemple, permettent de stocker des données (sortie) ainsi de les charger (entrée).

Disque dur externe : permet de lire et d'enregistrer des données informatiques

Les périphériques connectés à la carte mère de l'ordinateur

Comme l'on vient de voir un ordinateur est composé d'un assemblage d'éléments principaux, mais pour rentré en relation et interagir avec lui, il lui

faut lui ajouter des éléments appelés périphériques

Les périphériques d'entrée

On y retrouve principalement : clavier (frappe de texte), souris (pointage), scanner (numérisation de documents papier), micro, webcam, etc.

Clavier : Permet d'écrire

Souris : Permet de déplacer sur l'écran le curseur de pointage

Scanner: Permet l'acquisition d'images, de textes sur un ordinateur

Micro : Permet de capter une source sonore

Webcam : Permet de capter une source animée

Les périphériques de sortie

Les périphériques de sortie servent à faire sortir des informations du système informatique : écran, imprimante, haut-parleur, etc.

Écran : Permet de visualiser les informations venant de l'ordinateur

L'écran est un élément essentiel, souvent négligé, lors de l'achat d'un ordinateur confortable et donc productif. Il permet l'affichage de l'interface (graphique ou texte) qui permet à l'utilisateur d'interagir avec sa machine via le système d'exploitation.

Dans tous les cas l'écran est relié à l'unité centrale de l'ordinateur par un câble connecté à la sortie de la carte graphique. Le transfert se fait par un signal analogique ou numérique suivant la technologie supportée par la carte graphique et l'écran.

Principales caractéristiques d'un écran

Taille

Mesurée en pouces (1.7 cm à peu près) elle détermine la taille de la surface d'affichage de l'image en donnant la longueur de la diagonale du rectangle que forme l'écran. On retrouve les tailles standards suivantes : 15, 17, 19, 20, 21, 22, 24...

Résolution

Elle s'exprime en "nombre de lignes verticales" x "Nombre de lignes horizontales" et donne une indication sur la finesse de la représentation de l'image. Une résolution de 1024*768 (la résolution standard actuellement) permet d'afficher plus de détails et offre donc plus d'espace de travail qu'une résolution 800*600 (Basse résolution). En contrepartie de leur niveau de détail, les résolutions élevées peuvent affecter la vue et causer une fatigue du fait de la taille réduite des caractères affichés.

Contraste

Il s'exprime en x : 1 et donne une idée sur la capacité de l'écran à afficher une image de qualité aussi fidèle que possible à l'originale (c'est la même chose que pour un téléviseur).

Luminosité

Elle est mesurée au choix en Nits ou Cd/m^2 et n'est indispensable que pour regarder des vidéos sur écran. Pour une utilisation bureautique ou Internet, une faible luminosité est suffisante voire conseillée.

Temps De Réponse / Taux De Rafraîchissement

C'est la vitesse à laquelle l'image se renouvelle à l'écran. Plus le taux de rafraîchissement est élevé (temps de réponse bas) plus l'écran est réactif. Cet aspect est crucial pour une expérience de jeu satisfaisante.

Angle De Vision

Plus il se rapproche de 180° plus on peut voir l'image en se plaçant sur le coté. Pour les écrans classiques cet angle est de 180° en vertical et en horizontal, les LCD eux tournent autour de 150° à 178°, il faut alors se placer en face de l'écran pour profiter d'une image fidèle.

Imprimante: permet de mettre sur support d'impression (papier, carton, plastique, …) des images, des textes, … provenant de l'ordinateur.

Haut-parleur : permet d'émettre les sons provenant de l'ordinateur

<u>Les périphériques d'entrée-sortie</u>

Carte son : permet d'acquérir les sons extérieur ou de restituer les sons venant de l'ordinateur

Connexions à l'ordinateur

Sur les micro-ordinateurs, tous les périphériques sont reliés à la carte mère par

un connecteur que l'on insère, soit dans un port directement soudé à la carte mère.

Exemple des principaux ports de connexion :

- Ports **USB**, **Série**, **Parallèle** : qui permettent de connecter facilement des périphériques à l'ordinateur tels qu'une imprimante, une souris, un clavier, un disque dur externe, une webcam, un scanner, …
- Les ports **PCI**, **PCI Express** : qui permettent de connecter facilement à la carte mère des cartes additionnelles telles que la carte son, la carte vidéo, la carte réseau.
- Les ports **Séries** ou **Sérial-Ata** : Pour connecter le(s) disque(s) dur(s) interne à l'**unité centrale**, le(s) lecteur(s) de disque(s) (Cd-Rom, Dvd-Rom, Blue Ray), …

b) Soit dans un port disponible sur une carte d'extension, elle-même enfichée (soudé ou insérer dans un port) sur la carte mère.

Exemple de principaux ports de connexion :

- **Prise écouteurs**, **prise micro** : Relié en générale à la **carte son** qui nous permettent de connecter un micro et des écouteurs (haut-parleur) à notre **ordinateur**.
- Port **réseau** (ou **LAN**) : Relié à la **carte réseau** et qui nous permet de nous brancher notre ordinateur au réseau internet (Intranet) ou externe (exemple : Internet).
- Les ports **VGA** ou **DVI** : relié à la **carte graphique** et qui nous permettent de relier notre ou nos **écrans** à l'**ordinateur**.

La carte d'extension étant amovible, il est facile de la remplacer en cas de panne ou d'évolution technologique (exemple : carte son, carte réseau, …).

Pour reconnaitre Tous les composants reliés à la carte mère, l'ordinateur (ou le système d'exploitation installé sur le système informatique) doit disposer d'un logiciel (programme) qui lui permet de reconnaître le périphérique, c'est-à-

dire un logiciel chargé de communiquer avec lui et d'intégrer ses fonctionnalités au système d'exploitation.
Ce logiciel ou programme est appelé pilote ou driver (en anglais).

EXERCICES

Tout ce qui a germé au soleil du printemps devra supporter la chaleur du soleil de l'été afin de porter des fruits en automne (Natsu no Kansha).

Exercice 1

On utilise le code suivant pour représenter la date de naissance et le sexe d'un étudiant :

- *positions 1 et 2* : les deux dernières positions de l'année de naissance ;
- *position 3* : pour un homme le numéro du trimestre, pour une femme, le numéro du trimestre augmenté de 4.
- *Positions 4 et 5* : numéro d'ordre dans le trimestre du jour de la date de naissance.

a) Donnez votre date de naissance et codez la suivant ce code.

b) quels sont le sexe et la date de naissance d'un(e) étudiant(e) qui a le code 88587?

c) Ce code est il fiable ? Pourquoi ?

d) quelle information pourrait on rajouter si l'on dispose de trois positions supplémentaires pour rendre ce code fiable ?

Exercice 2

Compléter le tableau suivant

Binaire	Octal	Hexadécimal	Décimal
11101001101011011			
	5714		
		C9FA	

Exercice 3

On donne le tableau ci-dessous.

Chaîne de 4 bits	**Interprétation de son contenu en base 10 Entier positif**	**Interprétation de son contenu en base 10 Complément à un**	**Interprétation de son contenu en base 10 Complément à deux**
0000	…….	…….	…….
0111	…….	…….	…….
1000	…….	…….	…….
1010	…….	…….	…….
1111	…….	…….	…….

Exercice 4 : Numération binaire et hexadécimale

1) Convertir en binaire les nombres 397_{10}, 133_{10}, 110_{10}
puis en décimal les nombres 101_2, 0101_2, 1101110_2
et vérifier en convertissant pour revenir à la base d'origine.

2) Effectuer les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) 1100 + 1000
b) 1001 + 1011
c) 1100 - 1000
d) 1000 - 101
e) 1 + 1 + 1 + 1

3) Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) 1011 x 11
b) 1100 x 101
c) 100111 x 0110

4) Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) 100100 / 11
b) 110000 / 110

5) Convertir en binaire 127.75_{10} puis 307.18_{10}

Vous pourrez constater, à la réalisation de cet exercice, que la conversion du .18 peut vous entraîner « assez loin ». C'est tout le problème de ce type de conversion et la longueur accordée à la partie fractionnaire dépendra de la précision souhaitée.

6) Convertir en hexadécimal

a) 3167_{10} b) 219_{10} c) 6560_{10}

7) Convertir en décimal

a) $3AE_{16}$
b) FFF_{16}
c) $6AF_{16}$

8) Convertir en base 16

a) 128_{10}
b) 101_{10}
c) 256_{10}
d) 1001011_2
e) 1001011_2

9) Convertir en base 10

a) $C20_{16}$
b) $A2E_{16}$

10) Convertir en base 2

a) $F0A_{16}$
b) $C01_{16}$

Correction exercice 4

1)

Le résultat se lit en remontant : 1 1000 1101

Vérification

Puiss2	8	7	6	5	4	3	2	1	0
	256	128	64	32	16	8	4	2	1
x	1	1	0	0	0	1	1	0	1
	256	128				8	4		1

Soit 39710
13310 = 1000 0101 = 128 + 4 + 1
11010 = 110 1110 = 64 + 32 + 8 + 4 + 2

1012 = 4 + 1 = 510
01012 = idem, le zéro devant un nombre n'est pas significatif, en décimal ou en binaire
11011102 = 64 + 32 + 8 + 4 + 2 = 11010

2)
a) 1100 + 1000 = 10100
b) 1001 + 1011 = 10100
c) 1100 - 1000 = 0100
d) 1000 - 101 = 0011
e) 1 + 1 + 1 + 1 = 100 (en décomposant les additions)

3)

a) 1011 x 11 = 10 0001
b) 1100 x 101 = 11 1100
c) 100111 x 0110 = 1110 1010

4)

a) 100100 / 11 = **1100**

```
1       0       0       1       0       0               11

-       1       1                                       1
---------------------
0       0       1       1                               1
        -       1       1
        ------------------
                0       0       0                       0
                0       0       0       0               0
```

b) 110000 / 110 = 1000

5) Convertir en binaire 127.7510 puis 307.1810

127.7510

Partie entière : 111 1111

Partie fractionnaire :	.75 x 2	=	1.50
	.50 x 2	=	1.00

il ne reste plus de décimale, donc on s'arrête. ➔ **111 1111.11**

307.1810

Parte entière : 1 0011 0011

Parte fractionnaire :

.18 x 2	=	0.36
.36 x 2	=	0.72
.72 x 2	=	1.44
.44 x 2	=	0.88
.88 x 2	=	1.76
.76 x 2	=	1.52
.52 x 2	=	1.04 etc

=> 1 0011 0011.0010 111

6) Convertir en hexadécimal

a) 3167_{10} = C5F

3167	:	16	=	197	reste	15	soit	F
197	:	16	=	12	reste	5	soit	5
12	:	16	=	0	reste	12	soit	C

b) 219_{10} = DB (=13 x 16 + 11)

c) 6560_{10} = 19A0 en base 16

7) Convertir en décimal

c) $6AF_{16}$= 6 x 256+ 10 x 16 + 15 = 1536+160+15= 1711

a) $3AE_{16}$ = 3 x 16^2 + A (10) x 16 + E (14) x 1
= 3 x 256 + 10 x 16 + 14 = 942

b) FFF_{16} = 15 x 256 + 15 x 16 + 15 = 3840+240+15 = 4095

8) Convertir en base 16

a) 128_{10}

128 / 2 = 64 reste		= 0
64/2	= 32	= 0
32/2	= 16	= 0
16/2	= 8	= 0
8/2	= 4	= 0
4/2	=2	= 0
2/2	=1	= 0
1/2	= 0	= 1 ➔ 1000 0000 ➔ 80_{16}

b) 101_{10} = 65_{16}

c) 256_{10} = 100_{16}

d) 110_2 = 6_{16} en complétant le bloc de 4 bits = 0110

e) 1001011_2 = $4B_{16}$

9) Convertir en base 10

a) $C20_{16}$
On reconstitue le bloc de 4 bits ➔ 1100 0010 0000
Et on convertit le nombre binaire obtenu en décimal, 32 + 1024 + 2048 = 3104
b) $A2E_{16}$ ➔ 1010 0010 1110 ➔ 2+4+8+32+512+2048 = 2606 en base 10

10) Convertir en base 2

a) $F0A_{16}$ = 1111 0000 1010
b) $C01_{16}$ = 1010 0000 0001

Exercice 5.

Convertir en binaire, puis en octal, et enfin en hexadécimal les nombres suivants :
$(5A)_{16}$, $(CFBA)_{16}$, $(E10D)_{16}$, $(FF)_{16}$, $(B00)_{16}$, $(F000)_{16}$, $(FFFF)_{16}$.

Exercice 6

Effectuer les changements de base suivant comme indiquée en italique.

Base	Nombre	Equivalent	Dans la Base
10	255	*FF*	**16**
10	π	……	**2**
8	53,125	…….	**10**
2	0,1010	……..	**16**
16	0,2A	……..	**10**
10	0,2187	……	**2**
10	173,12	…….	**8**

Exercice 7

1) Qu'est ce qu'un code d'instruction ?
Une **instruction** est l'opération élémentaire que le processeur peut accomplir. Les instructions sont stockées dans la mémoire principale, en vue d'être traitée par le processeur. Une instruction est composée de deux champs :

- le **code opération**, représentant l'action que le processeur doit accomplir ;
- le **code opérande**, définissant les paramètres de l'action. Le code opérande dépend de l'opération. Il peut s'agir d'une donnée ou bien d'une adresse mémoire.

2) Quels sont liens physiques entre le processeur et la mémoire ?
Les Bus : bus d'adresses – bus Commandes – bus de données

***Exercice 8*.**

Voici quelques caractéristiques de la fiche technique d'un ordinateur :

Processeur	**A**	**B**
	…….	AMD
		Athlon64 x2
		2Ghz
		512 Mo
Mémoire	**A**	**B**
		2Go
		DDR2
		667 Mhz
Disque dur	**A**	**B**
		500 Go
		7200 t/min
Logiciel	**A**	**B**
		Windows Vista Edition familiale Premium
Connexion	**A** connecteurs	**B** 8 ports USB 2.0, RJ45
Son	**A** Marque carte son	**B** RealTek
Communication	**A** Chipset Réseau	**B** LAN 10/100/1000

1) Remplir les lignes de la colonne A
2) Définir en deux lignes le rôle de chaque élément de cet ordinateur
3) Déterminer le nombre de cycles d'attente pour un cycle de transfert pour cet ordinateur. Quel commentaire faites-vous sur les performances de cette machine ?

Exercice 9 : *(Sujet Examen)*

Questions de cours

Définir les éléments du modèle de machine universelle de Von Neumann. Définir chaque élément.

Exercice 1

Une machine travaille sur 16 bits en complément à deux.

a) Dire pourquoi les machines ne travaillent pas en complément à un ?
b) Donner les codes du max et du min en complément à deux.
c) BA2C et 3D10 sont en complément à deux. Donner leurs valeurs décimales.
d) On effectue les opérations suivantes :
   32846 + 19188

16845 + 29300
32766 +1
Donner les valeurs affichées à l'écran.

Exercice 2

On suppose que l'on travaille sur 16 bits. On donne deux entiers :
$X= (2FAD)_h$ et $Y= (7A7D)_h$.
Soient S = X+Y et D =Y-X

a) Donner les valeurs décimales de X et Y.
b) Ecrire X et Y en binaire
c) Calculer les valeurs de S et D en décimal. Conclusion ?

Exercice 3

a) Soit $A= (101101)_2$ et $B= (101)_2$.
Donnez les valeurs décimales de A et B.
b) Effectuer A/B en binaire.
c) Convertir C= (111011.1111) en décimal

Correction sujet examen

Questions de cours (5pts)
Définir les éléments du modèle de machine universelle de Von Neumann. Définir chaque élément (voir cours)

Exercice 1 (6pts)

a) Les machines ne travaillent pas en complément en un car il existe deux représentations du 0 dans ce mode de représentation (0.5pt)
0000 0000 0000 0000 et 1000 0000 0000 0000 (1pt)
b) Sur 16 bits Max= 0111 1111 1111 1111 Min = 1000 0000 0000 0000 (1 pt)
c) BA2C et 3D10 sont en complément à deux.
BA2C = 1011 1010 0010 1100 est un nombre négatif. Le complément à un est 1011 1010 0010 1011 ; l'inversion des bits donne 0100 0101 1101 0100 qui vaut 17876 en décimal. BA2C = - 17876 (1pt)
3D10 est positif. Il est donc équivalent à 15632 en décimal (1pt)
d) On effectue les opérations suivantes : (2pts)
32846 + 19188. Le max sur 16 bits est 32767. Dépassement de la capacité (erreur)
16845 + 29300 = -19 391
32766 +1 = 32767 = Max

Exercice 2 (6 pts)

On suppose que l'on travaille sur 16 bits. On donne deux entiers :

X= $(2FAD)_h$ et Y= $(7A7D)_h$.
Soient S = X+Y et D =Y-X

d) Donner les valeurs décimales de X et Y.
   X= 0010 1111 1010 1101 est >0 il vaut 12205 (1pt)
   Y= 0111 1010 0111 1101 est >0 et vaut donc 31357 (1pt)
e) Ecrire X et Y en binaire (voir réponse ci-dessus) (2pts)
f) Calculer les valeurs de S et D en décimal. Conclusion ?
   S= X+ Y = 12205 + 31357 = 43562 alors que le max est +32767. On sait que Max+1 = Min=-32768. On trouve donc S= -21974 (1pt)
   D= 31357 – 12205 = 19152 (1pt)

Exercice 3 (3 pts)

d) Soit A= $(101101)_2$ et B= $(101)_2$.
   Donnez les valeurs décimales de A et B.
   A= 45 et B= 5 (1pt)
e) Effectuer A/B en binaire. La division binaire est égale à 1001 (1pt)
   Convertir C= (111011.1111) en décimal
   La partie entière 111011 = 59 et la partie entière vaut 0.9375 donc
   C= 59.9375 (1pt)

Exercice 10 : (sujet examen)

Exercice 1

(Les parties A, B et C sont indépendantes).

Partie A

Les adresses sont codées en <u>hexadécimal</u> sur 4 positions. Une donnée Xp est à l'adresse 7A4C.

a) Donnez l'adresse de Xp en décimal.
b) Donnez l'adresse du premier et du dernier emplacement en hexadécimal.
c) Une donnée Xc est située au 4402^e emplacement. Donnez son adresse en hexadécimal.
d) La mémoire est organisée en emplacement de 2048 bits. Calculer sa capacité en Mo.

Partie B

Chaîne de 6 bits	**Interprétation de son contenu en base 10 (Entier positif)**	**Interprétation de son contenu en base 10 (complément à un)**	**Interprétation de son contenu en base 10 (complément à deux)**
111011			
101111			
011110			

a) Compléter le tableau ci-dessus
b) Donner les valeurs décimales du Min et du Max si l'on travaille en complément à un ou en complément à deux.

Partie C

1) Qu'est ce qu'un système binaire ? Quel est l'intérêt de ce système de numération en informatique ? Pourquoi a t – on l'habitude d'utiliser le code hexadécimal au lieu du code binaire dans le codage de l'information ?

2) Dans un ordinateur multimédia :

a) Donnez 3 éléments qui permettent de définir sa performance?
b) Quel est le rôle de chacun de ces 3 éléments en quelques lignes ?

Exercice 2

Le braille est une écriture en relief pour les malvoyants. Les caractères de cette écriture sont constitués de six points en relief (trou ou boss) sur une grille de 3 lignes et 2 colonnes.

a) quelle es la quantité d'information d'un caractère de ce code ?
b) Montrer que ce code suffit pour écrire pratiquement tous les textes.

Exercice 3

a) Effectuez les opérations suivantes en binaire puis donnez les résultats en décimal.

1101010111 + 0101101111
1000011010 – 0111010111
1011 * 101

b) Effectuer les opérations suivantes en hexadécimal puis convertir les résultats en binaire, en octal et en décimal.

$(175A)_h + (B92C)_h$
$(DEEA)_h + (BCD7)_h$

Exercice 11:(sujet examen corrige)

Session 1 2014-1015

(Calculatrices, ordinateurs portables et téléphones portables non autorisés)
Bien indiquer la démarche suivie pour aboutir aux différents résultats

Exercice 1 **2.25 pts**

Complétez le tableau ci-dessous

Chaîne de 6 bits	Interprétation de son contenu en base 10 Entier positif	Interprétation de son contenu en base 10 Complément à un	Interprétation de son contenu en base 10 Complément à deux
100010	34 0.25pt	-29 0.25pt	-30 0.25 pt
011101	29 0.25pt	29 0.25pt	29 0.25 pt
111111	63 0.25pt	0 0.25 pt	-1 0.25 pt

Exercice 2 **6pts**

1- Effectuez les opérations suivantes :

a) Calculer X= $A3E_{(16)} + CEA_{(16)} = (1728)_{16}$ **1 pt**
b) Calculer Y= $C2A_{(16)} - 3AD_{(16)} = (87D)_{16}$ **1pt**
c) Convertir X et Y en base 2

$X = (1728)_{16} = (0001\ 0111\ 0010\ 1000)_2$ **1pt**

$Y = (87D)_{16} = (1000\ 0111\ 1101)_2$ **1 pt**

2- Adresse de mémoire et capacité de mémoire

d) La mémoire vive d'une petite machine est de 1 Ko. Elle est conçue à l'aide de transistors couplés à des condensateurs (état 0 ou 1). Combien de condensateurs trouve-t-on dans ce type de mémoire ?
1ko = 1027 octets = 1024x8 bits = 8192 bits = 8192 condensateurs **1pt**

e) Les emplacements de mémoire sont en mots de 8 bits. Donner le nombre d'emplacements mémoire.
Nombre d'emplacements : il suffit de diviser 8192 bits par 8 = 1024 emplacements 1pt

f) Nombre d'adresses de mémoire ?
Nombre d'adresses de mémoire ? il y a 1024 adresses, il faut donc 10 bits d'adresse. 1pt

Exercice 3 6pts

Une petite machine travaille en complément à deux sur 8 bits.

a) Donnez l'intervalle des valeurs entières acceptées par cette petite machine.
[-128, +127]
1pt

b) Donnez le code du maximum en complément à deux. En déduire la valeur du minimum
Code du max : 01111111 en faisant max+1 on obtient le code du min soit 10000000 1pt

c) Donnez le code de +1 en complément à deux sur 8 bits. En déduire celui de -1.
Code de 1 : 00000001 ; cr(-1) = 11111110 cv(-1)= cr(-1)+1 soit 11111111 1pt

d) On effectue les opérations suivantes :

d-1) 100 + 100

100 + 100 = 200 en utilisant max+1 = min, on aura 200 = 127 +1 +x avec x= 72,
or 127+1=min, donc on verra afficher **-56**
1pt

d-2) 145 + 125 **Erreur car 145 est en dehors de l'intervalle [-128, 127]**
1pt

d-3) 127 +25

Comme dans le premier cas, on ne verra pas afficher 152 mais **-104**
1pt

Donnez les valeurs affichées à l'écran dans le cas de ces 3 opérations.

Exercice 4 6.5 pts

On utilise le code suivant pour représenter la date de naissance et le sexe d'un habitant d'une banlieue Z:
On utilise le code suivant pour représenter la date de naissance et le sexe d'un habitant d'une banlieue Z :

- *Position 1& 2 :* numéro d'ordre dans le bimestre du jour de la date de naissance.
- *positions 3, 4, 5 & 6* : l'année de naissance
- Positions 7 et 8 : pour un homme le numéro du bimestre, pour une femme, le numéro du bimestre augmenté de 6.

Positions 1 & 2	Positions 3, 4,5 & 6	Positions 7 & 8

a) Codez la date de naissance d'un homme né le 29 Février 1975

Pas possible car pas de 29 Février en 1975 (pas une année bissextile)

0.5pt

b) quels sont le sexe et la date de naissance d'un habitant qui a le code 60198512?

60198512 : 60ème jour du 6ème bimestre de l'année 1985. Il s'agit d'une femme née le 30 Décembre 1985.

0.5pt

c) on se propose d'utiliser le code binaire pour représenter une information en lieu de place du code décimal. Combien de bits minimum faut-il utiliser pour représenter le code d'un habitant ?

- pour les positions 1&2, il y a au max 62 jours (6 bits max)

0.5pt

- positions 3, 4, 5&6 (on prend un max de 2100 pour l'année) : 2100 en base 2 = 1000 0011 0100, il faut donc 12 bits

0.5 pt

- Positions 7 & 8 : max = 12, il faudrait donc 4 bits

Total = 22 bits

0.5 pt

d) On ajoute 3 bytes pour coder les données biométriques d'un habitant puis on ajoute un petit en-tête en ASCI étendu comme indiqué dans le tableau ci-dessous : **Commune de Bingerville**

Information biométrique	Commune de Bingerville	Positions 1 &2	Positions 3, 4, 5 & 6	Positions 7 & 8

Donner le nombre de bits nécessaire utilisé pour représenter le code d'un habitant

- information biométrique 3 bytes, soit 24 bits

0.5 pt

- Commune de Bingerville codé en ASCII étendu donne

Commune (7 caractères), espace (1 caractère), de (2 caractères), espace (1 caractère), Bingerville (11 caractères), soit au total 7+1+2+1+11 = 22 caractères.

8 bits par caractères, donc 176 bits

2pts

Pour un habitant, il faudrait donc 22 bits+176 bits = 198 bits au total

0.5 pt

d) L'on suppose que la banlieue comprend 2047 individus. Combien de bits d'adresse doit-on utiliser pour référencer tous les habitants de la banlieue ?

2047 en binaire nécessite 11 bits d'adresse

0.5pt

e) Donnez la capacité de mémoire nécessaire pour archiver toutes les données sur les habitants de la banlieue.

2047 * 198 bits /8 = 50663 octets environ 49.5 Ko

0.5pt