**Introduction Architecture**

7 semaines, 1h cours + 3h TD

Sommaire :

- Encodages et représentations

- Représentations des nombres

- Ecriture positionnelle en base quelconque

- Représentation des nombres relatifs

- Nombres a virgules

- Operateurs

- Circuits

- Additionneurs

- Assembleur

DEF: Un but 1 est une unité d’information qui représente un état logique pouvant prendre deux valeur (0/1).

**Proposition:**

    Avec N bits, on peut représenter 2^n valeurs differentes

    1 bit : 0,1

    2 bits: 00 ,01 , 10 ,11

    3 bits: 000, 001 , 010, 011, 100 , 101 , 110 , 111

**Proposition:**

    Pour représenter K valeurs distinctes , il faut utiliser au moins [log2(k)] bits

    6 valeurs : log2(6) ≈ 2,59 -> 3 bits

    8 Valeurs : log2(8) = 3 -> 3 bits

    9 Valeurs : log2(9) ≈ 3,17 -> 4 bits

**ASCII** ( American Standard Code for Information Interchange)

* Representation des caracters (pour l'anglais ) sur 7 bits
* Developpee dans les annees 1960 revisee jusqu'en 1986

**OCTET:**

En pratique on considere les bits par groupes de 8 :

* 1 octet = 8 bits = 256 valeurs
* La memoire des ordinateurs est adressee par octets (chaque octet a un numero qui est son adresse)
* Les octets Peuvent etre interpretes comme une valeur entre 0 et 255

Pour les grands valeurs , on utilise des unites multiples d l'octet

**Historiquement** :

* 1 kilooctet (ko) = 1024 o
* 1 mégaoctet (Mo) =1024 ko
* 1 gigaoctet (Go) = 1024 Mo
* ...
* 1 kibioctet (kio) = 1024 o
* 1 mébioctet (Mio) = 1024 Mio
* 1 gibioctet (Gio) = 1024 Gio
* ...

**ASCII 2**

* La table a depuis été étendue de differentes manières pour ajouter des caracteres supplémentaires
* Ce sont les versions ISO 8859-l (latin l) et ISO 8859-15 (latin 9) qui correspondent le mieux au francais

**TEXTE**

* Si l'aphabet est simple , on peut utiliser un format ASCII (p.ex  ISO-latin-9 pour le francais)
* 1 octet par caractere

ASCII      HEX         BINAIRE

H           48          0100 1000

e           65          0110 0101

l            6c          0110 1100

l            6c          0110 1100

o           6f           0110 1111

Si on veut utiliser un aphabet plus large, on peut utiliser l'Unicode :

* 143859 Caracteres
* 154 systemes d'ecriture ( modernes et historiques )
* Symboles divers ( emoji, musique , etc.)

**Il existe plusieurs encodages Unicode** :

UTF-8 :

* 1 a 4 octets par caractere
* 0xxxxxxx
* 110xxxxx 10xxxxxx
* 1110xxxx 10xxxxxx 10xxxxxx
* 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
* Compatible ASCII
* Environ 97% des pages web actuelles

UTF-16 :

* 2 ou 4 octets par caractere
* Non compatible ASCII
* Utilise en interne

UTF-32 :

* taille fixe de 4 octets par caractere

**Images :**

* Pour representer une image, on peut decrire chacun de ses pixels
* Pour une image en couleurs de n x m pixels:
* n x m pixels
* 3 composantes par pixel (RGB)
* 1 octet pour chaque composante (0 a 255)
* -> 3 x n x m octets pour representer l'image ( 3 Mo pour une image 1000x1000)

**Audio**

* Pour representer une fonction continue (p.ex piste audio)
* Un ou plusieurs canaux independants ( mono: 1 canal, Stereo: 2 canaux)
* Mesures reguliees au cours du temps (echantillons) -> taux d'echantillonage en Hertz
* Taille fixee pour les echantillons( bit depth) -> nombre fini de valeurs possibles
* Taille d'un fichier audio (en octets) :
* taille d'un echantillon (en octets)
* taux d'echantillonage (en Hz)
* durée ( en secondes)
* Nombre de canaux
* Environ 10Mo par minute pour un fichier *wave* en stereo, 44.1 kHz, 16 bits par echantillon

TO BE CONTINUED... EP 2 NEXT WEEK