Archi : CH1 :

Codage des nombres en base 2, logique booléenne, portes logiques, Circuits Combinés

1] Interprétation des vecteurs de bits :

i) Vecteur de bits :

- une suite de bits (le bit $\in \{0,1\}$) finis
- 1 octet (bytes) = 8 bits
- 1 short = 16 bits = 2 octet
- 1 int (float) = 32 bits = 4 octet

ii) Ecriture en base 2 :

- Soit X un vecteur de n bits = $(x_{n-1}, \ldots, x_0)$ avec $x_i \in \{0,1\}$

Alors.
$$x = \sum_{i=n-1}^{0} x_i 2^i$$ (x interprété comme entier naturel)

- Bit le plus gauche (de poids 2^{n-1}) : MSB.
- Bit le plus droite (de poids 1) : LSB
- On a : $0 \leq x \leq 2^n - 1$ (min : 0....0 ; max : 111...1)

iii) Conversion binaire / décimale des entiers naturels :

* base 2 → base 10 :

On applique la formule : $x = \sum_{i=n-1}^{0} x_i 2^i$

* base 10 → base 2 :

$x = 2V_0 + x_0$

$V_0 = 2V_1 + x_1$

$V_i = 2 V_{i+1} + x_{i+1}$

Arrêt : lorsque $V_j < 2$

$$\Longrightarrow x = (V_j, x_j, \ldots, x_1, x_0)$$

iv) Conversion des entiers relatifs :

- Soit $X = (x_{n-1}, \ldots, x_0)$ un vecteur de n bits
- si X est interprété comme entier relatif

Alors
$$x = -x_{n-1} \cdot 2^{n-1} + \sum_{i=n-2}^{0} x_i 2^i$$

v) Representation de $-x$ à partir de x :

- $x = (x_{n-1}, \ldots, x_0)$
- Le complémentaire d'un bit :

$$a \in \{0, 1\}, \quad \bar{a} = \begin{cases} 1 & \text{si } a = 0 \\ 0 & \text{si } a = 1 \end{cases}$$

- Alors

$$-x = (\bar{x}_{n-1}, \ldots, \bar{x}_0) + 1$$

Rec :
- x_{n-1} est appelé le bit de signe (sign bit) ($x_{n-1} = 1 \Rightarrow x < 0$)
- $-2^{n-1} \leq x \leq 2^{n-1} - 1$
- n bits ($10\ldots0$: min, $01\ldots1$: max).

vi) Extension :

- Affectation d'un vecteur de taille n dans un vecteur de taille m :
  - $n > m$: troncature : il y a une perte
  - $n < m$: exige une extension
- Soit $(x_{n-1}, \ldots, x_0) = x$, $m > n$

Alors :

$$x = (y_{m-1}, \ldots, y_n, y_{n-1}, \ldots, y_0) \text{ tq :}$$

$$\begin{cases} y_{m-1} = y_{m-2} = \ldots = y_{n-1} = x_{n-1} \\ y_{n-2} = x_{n-2}, \ldots, y_0 = x_0 \end{cases}$$

vii) Nombres à virgule fixe :

- Soit X un vecteur : Partie entière sur n bits, partie fractionnaire sur m bits : $x_{n-1} \ldots x_0, y_1 y_2 \ldots y_m$

Alors,

$$x = \sum_{i=n-1}^{0} x_i 2^i + \sum_{i=1}^{n} y_i 2^{-i}$$

2] Notations binaire usuelles :

* L'hexa-décimal (base 16), L'octale (base 8) sont parfois utilisés

*

binaire	hexa	octal
0000	0	0
0001	1	1
0010	2	2
0011	3	3

* Notation :
  - Langage C : 0x : hexa, 0 : octal, 0b : binaire
  - VHDL : X"" : hexa, O" " : octal, B" " : binaire
  - python : 0x : hexa, 0o : octal, 0b : binaire

* Les unités : Kilo, Mega, Giga :
  $1K = 2^{10} = 1024 = 0x400$, $1M = 2^{20}$, $1G = 2^{30}$

3] Rappels sur les booléens :

i) Déf :

- $\mathbb{B} = \{0, 1\}$: ensemble des booléens
  $x \in \mathbb{B}$: variable booléenne

ii) Les opérations :

Operateur	notation info	notation math
non	not $x = \bar{x}$	$\neg x$
ET	x and $y = x \cdot y$	$x \wedge y$
OU	x or $y = x + y$	$x \vee y$

iii) Propriétés :

Associativité : $a + (b + c) = (a + b) + c$ / $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

Commutativité : $a + b = b + a$ / $a \cdot b = b \cdot a$

Distributivité : $a + (b \cdot c) = (a + b) \cdot (a + c)$ / $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Absorption : $a + (a \cdot b) = a$ / $a \cdot (a + b) = a$

Complementation : $a + \bar{a} = 1$ / $a \cdot \bar{a} = 0$

Idempotence : $a + a = a$ / $a \cdot a = a$

Éléments neutres : $a + 0 = a$ / $a \cdot 1 = a$
$a + 1 = 1$ / $a \cdot 0 = 0$

Involution : $\bar{\bar{a}} = a$.

iv) Lois de Morgan :

- $\overline{a+b} = \overline{a} \cdot \overline{b}$ / $\overline{a \cdot b} = \overline{a} + \overline{b}$
- $\overline{\sum_{i=1}^{n} a_i} = \prod_{i=1}^{n} \overline{a_i}$ / $\overline{\prod_{i=1}^{n} a_i} = \sum_{i=1}^{n} \overline{a_i}$

- Principe de dualité :
  une loi valide peut être transformée en son dual en échangeant 0 avec 1 et + avec •

4] Représentation des fonctions :

i) Table de vérité

* Énumération de toutes les combinaisons possibles des entrées
  ⟹ n variables impliquent 2^n valeurs possibles
* $x, y \Rightarrow \overline{x}, x \cdot y, x+y, x \oplus y, \overline{x \cdot y}, \overline{x+y}, \overline{x \oplus y}$
* Il y a 2^{2^n} fonctions booléennes de n variable

ii) Formes canoniques d'équations booléennes :

* Somme de produits : (Forme : disjonctive)
  - 1 en sortie : obtenu par produit des littéraux
  - fonction obtenue : par somme de ces produits
* Produit de sommes : (Forme : conjonctive) :
  - 0 en sortie : obtenu par somme des littéraux
  - fonction obtenue : par produit de ces sommes

Ex :

x	y	f	SOP	POS
0	0	0		$x+y$
0	1	1	$\overline{x} \cdot y$	
1	0	0		$\overline{x}+y$
1	1	0		$\overline{x}+\overline{y}$

d'où

$$f = \overline{x} \cdot y = (x+y)(\overline{x}+y)(\overline{x}+\overline{y})$$

iii] Porte logique (logic gate),

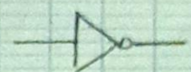
* **Def**

Porte logique : représentation schématique d'un opérateur booléen.

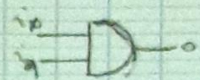
* Une porte possède :
  - des connecteurs d'entrée : i_j $j \in [\![0, n-1]\!]$
  - un connecteur de sortie : o
  - un comportement : $o = f(i_{n-1}, \dots, i_0)$

* **Portes de base :**

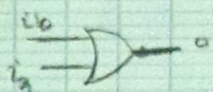
- $\bar{i}$: $o \Leftarrow$ not i



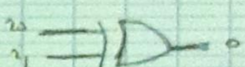
- $i_0 \cdot i_1$: $o \Leftarrow i_0$ and i_1



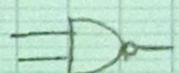
- $i_0 + i_1$: $o \Leftarrow i_0$ or i_1



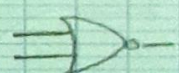
- $i_0 \oplus i_1$: $o \Leftarrow i_0$ xor i_1



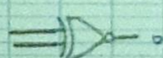
- $\overline{i_0 \cdot i_1}$: $o \Leftarrow i_0$ nand i_1



- $\overline{i_0 + i_1}$: $o \Leftarrow i_0$ nor i_1



- $\overline{i_0 \oplus i_1}$: $o \Leftarrow i_0$ xnor i_1



* **Aspects temporels :**
  - Temps $t_{pe}(i, o)$: le temps de propagation entre i et o
  - Temps $t_{te}(o)$: (transition) : le temps de passage de 0 à 1 ou de 1 à 0.

IV] Circuit Combinatoire :

*** Def :**

Circuit Combinatoire : Représentation schématique d'un ensemble de portes ou Circuits interconnectés

* Un Circuit Combinatoire possède :

- des Connecteurs d'entrée : i_j ; $j \in [\![0, n-1]\!]$
- des Connecteurs de Sortie : O_K : $0 \le K \le m-1$
- un Comportement : $(O_{m-1}, \dots, O_0) = f(i_{n-1}, \dots, i_0)$

*** Les interconnexions :**

- entrée primaire sur entrée porte
- Sortie de porte sur entrée(s) de porte(s)
- Sortie de porte(s) sur Sortie primaire
- entrée primaire sur Sortie primaire

Exemple

1) Réalisation en portes de l'équation : $t = \overline{(x+y) \cdot \bar{z}}$