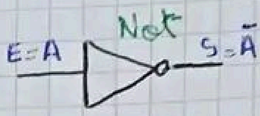
Algèbre de boole

opérateur logique : Non, ou, et.

opérateurs de base :

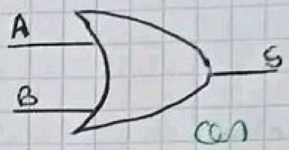
- la négation (non) : $\bar{A}$

E	S
0	1
1	0



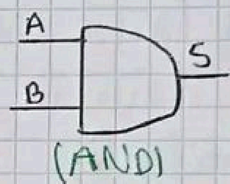
- la disjonction (ou) : $A + B$ (or)

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



- la conjonction (et) : $A . B$ (AND)

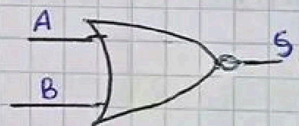
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1



opérateurs composés :

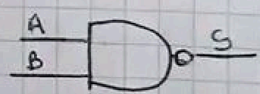
- opérateur NOR (Non ou) : $\overline{A + B} = A \downarrow B$

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



- opérateur Nand (Non et) : $\overline{A . B} = A \uparrow B$

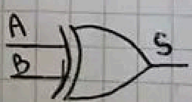
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



- opérateur XOR (ou exclusif).

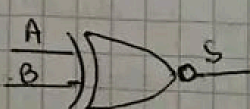
$A \oplus B = \bar{A} . B + A . \bar{B}$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



- opérateur XNOR : $\overline{A \oplus B} = A . B + \bar{A} . \bar{B}$

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1



Evaluation des expressions booléennes

Disjonction (+)	Conjonction (·)
$1+1=1$	$1.1=1$
$1+0=0+1=1$	$1.0=0.1=0$
$0+0=0$	$0.0=0$
$\bar{1}=0$	$\bar{0}=1$

Axiomes

propriété	Disjonction (+)	Conjonction (·)	
commutativité	$A+B=B+A$	$A.B=B.A$	تبديل
Associativité	$A+(B+C)=(A+B)+C$	$A.(B.C)=(A.B).C$	تجميع
Distributivité	$A+(B.C)=(A+B)(A+C)$	$A.(B+C)=A.B+A.C$	توزيع
élément neutre	$A+0=A$	$A.1=A$	العنصر الحيادي
" absorbant	$A+1=1$	$A.0=0$	العنصر الماص
complémentation	$A+\bar{A}=1$	$A.\bar{A}=0$	
idempotence	$A+A=A$	$A.A=A$	
involution	$\bar{\bar{A}}=A$	$\bar{\bar{\bar{A}}}=\bar{A}$	

Théorèmes

1. théorèmes de Morgan :

$\overline{A.B}=\bar{A}+\bar{B}$ $\qquad$ $\overline{A+B}=\bar{A}.\bar{B}$

2. théorèmes d'inclusion :

$A.B+A.\bar{B}=A$ $\qquad$ $(A+B).(A+\bar{B})=A$

3. théorèmes d'allégement :

$A.(\bar{A}+B)=A.B$ $\qquad$ $A+\bar{A}.B=A+B$

4. Thèremes d'absorption.

$A+A.B=A$ $\qquad$ $A.(A+B)=A$

5. théorème de dualité :

$A+A.B=A \Longleftrightarrow A.(A+B)=A \;;\; A+1=1 \Leftrightarrow A.0=0$

6 - Réflexion de Michaud :

Si $A = B \oplus C$ alors $B = A \oplus C$

- $F(a,b,c) = \bar{a}b + \bar{b}c$

$= \bar{a}b \cdot (c+\bar{c}) + (a+\bar{a}) \cdot \bar{b}c$

$= \bar{a}bc + \bar{a}b\bar{c} + a\bar{b}c + \bar{a}\bar{b}c$

- $F_1(a,b,c) = \underset{\substack{011\\3}}{\bar{a}bc} + \underset{\substack{010\\2}}{\bar{a}b\bar{c}} + \underset{\substack{101\\5}}{a\bar{b}c} + \underset{\substack{001\\1}}{\bar{a}\bar{b}c}$

$F_1(a,b,c) = \Sigma(1,2,3,5)$

- $F_2(a,b,c) = \underset{\substack{001\\1}}{(a+b+\bar{c})}\underset{\substack{010\\2}}{(a+\bar{b}+c)}\underset{\substack{100\\4}}{(\bar{a}+b+c)}$

$F_2(a,b,c) = \Pi(1,2,4)$

Table de vérité

A	B	C	F Majorité	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A} \cdot B \cdot C$
1	0	0	0	
1	0	1	1	$A \cdot \bar{B} \cdot C$
1	1	0	1	$AB\bar{C}$
1	1	1	1	ABC

forme disjonctive

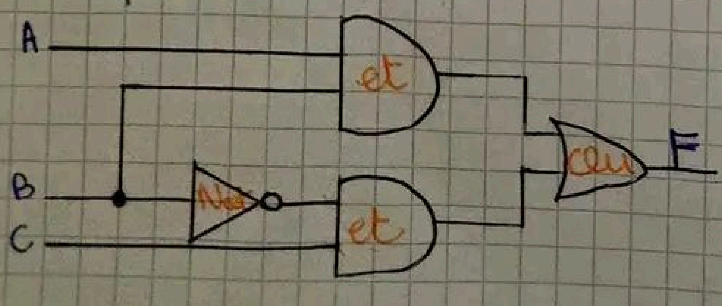
$Majorité(A,B,C) = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + ABC$

forme conjonctive

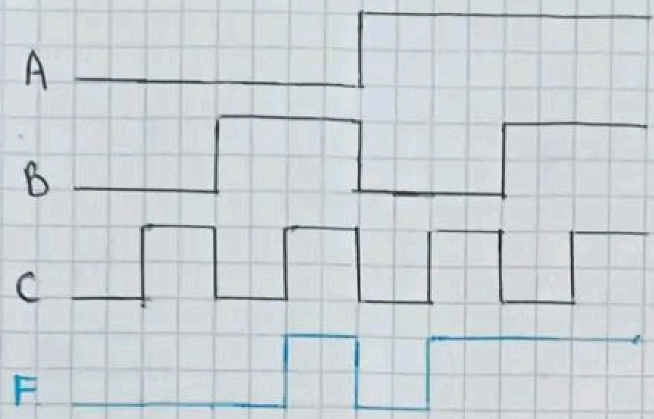
$= (A+B+C) \cdot (A+B+\bar{C}) \cdot (A+\bar{B}+C) \cdot (\bar{A}+B+C)$

logigramme :

exemple : $F = A \cdot B + \bar{B} \cdot C$

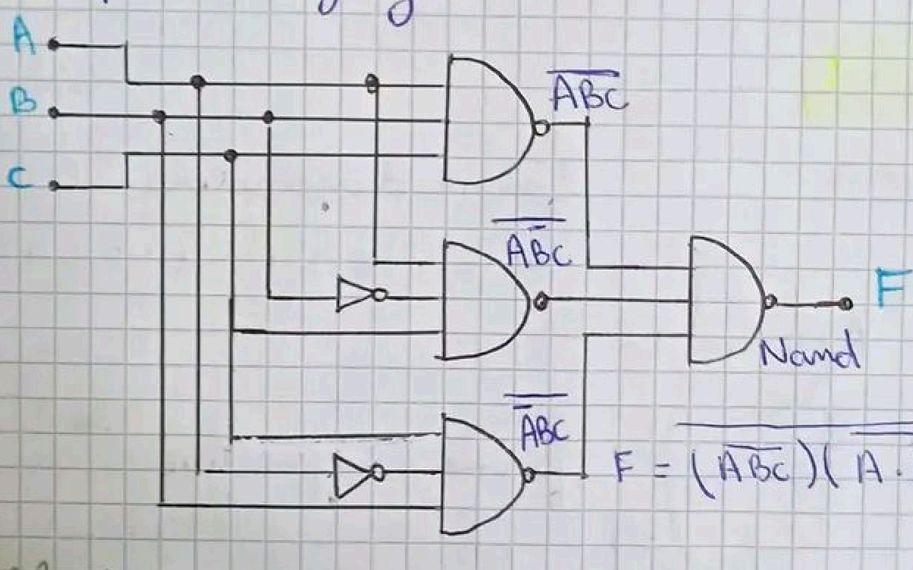


chronogramme :



Analyse d'un circuit logique :

exemple : analysez le circuit suivant



$$F = \overline{(\overline{ABC})(\overline{A\cdot\bar{B}\cdot C})(\overline{\bar{A}\cdot BC})}$$

table de vérité :

A	B	C	$\overline{ABC}$	$\overline{A.\bar{B}C}$	$\overline{\bar{A}.BC}$	F
0	0	0	1	1	1	0
0	0	1	1	1	1	0
0	1	0	1	1	1	0
0	1	1	1	1	0	1
1	0	0	1	1	1	0
1	0	1	1	0	1	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1

la fonction simplifiée:

AB \ C	0	1
00		
01		1
11		1
10		1

Bc

Ac

$F = BC + AC$

Synthèse d'une fonction logique:

بال code BCD مثال الجدول عندما نجد مثلا 1|0|1|0 هادي
في F نضع 1 لأنه أكثر من 9 و أقل من 9 نضع 0

Simplification des fonctions:

1) Simplification algébrique:

exemple: $F = \bar{a}.b.\bar{c}.\bar{d} + a.b.\bar{c}.d + \bar{a}\bar{b}\bar{c}d + \bar{a}b\bar{c}d + ab\bar{c}d + a\bar{b}\bar{c}d$

$= b\bar{c}\bar{d}(\bar{a} + a) + \bar{c}d(\bar{a}\bar{b} + \bar{a}b + ab + a\bar{b})$

$= b\bar{c}\bar{d} + \bar{c}d(\bar{a}(\bar{b} + b) + a(b + \bar{b}))$

$= b\bar{c}\bar{d} + \bar{c}d(\bar{a} + a)$

$= b\bar{c}\bar{d} + \bar{c}d = \bar{c}(b\bar{d} + d) = \bar{c}(d + \bar{d}).(d + b) = \bar{c}(b + d)$

conjunction (AND) (et)

disjunction (or) (ou)

célèbres bases :

Décimale = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

binaire = {0, 1}

octale = {0, 1, 2, 3, 4, 5, 6, 7}

hexadécimale = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Changement de bases :

Décimale → base b

- divisions successives du nombre par la base jusqu'à avoir un résultat nul.

exemple $(\underbrace{143}_{}, \underbrace{526}_{})_{10} = (?)_2$ avec 7 chiffres après la virgule.

→ partie décimale : multiplications successives

partie entière : div successives

$$
\begin{array}{l}
143 \div 2 = 71 \text{ reste } 1\\
71 \div 2 = 35 \text{ reste } 1\\
35 \div 2 = 17 \text{ reste } 1\\
17 \div 2 = 8 \text{ reste } 1\\
8 \div 2 = 4 \text{ reste } 0\\
4 \div 2 = 2 \text{ reste } 0\\
2 \div 2 = 1 \text{ reste } 0\\
1 \div 2 = 0 \text{ reste } 1 \quad \text{stop}
\end{array}
$$

$= (10001111)$

0,526 x 2 = 1,052
0,052 x 2 = 0,104
0,104 x 2 = 0,208
0,208 x 2 = 0,416
0,416 x 2 = 0,832
0,832 x 2 = 1,664
0,664 x 2 = 1,328

=> $= (0,1000011)$

donc
$\Rightarrow (10001111,1000011)_2$

base b → décimale

il suffit de l'écrire sous sa forme polynomiale dans la base b, ensuite faire l'addition décimale.

exemple $(\overset{3}{1}\overset{2}{5}\overset{1}{7}\overset{0}{2}, \overset{-1}{3}\overset{-2}{6})_8 = (?)$

$$1\times 8^3 + 5\times 8^2 + 7\times 8^1 + 2\times 8^0 + 3\times \underbrace{8^{-1}}_{\frac{1}{8}} + 6\times \underbrace{8^{-2}}_{\frac{1}{8^2}} = 512 + 320 + 56 + 2 + 0{,}375 + 0{,}09375$$

$$= (890{,}46875)_{10}$$

Cas particuliers :

binaire → octale

$b_1 \xrightarrow{\text{fdg } 10} \xrightarrow{\text{ai mul}} b_2$

$8 = 2^3$ exemple : $(11010{,}0111)_2 = (?)_8$

$(011010{,}011100) = (32{,}34)_8$

octal	binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

octal → binaire

exemple : $(726{,}31)_8 = (?)_2 = (111010110{,}011001)_2$

binaire → hexadécimale

$16 = 2^4$

exemple :

hexa	binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

$\Rightarrow (11010{,}10111)_2 = (0001\ 1010{,}1011\ 1000)_2$

$= (1A{,}B8)_{16}$

hexadécimale → binaire

نفس الشيء

2 - Représentation interne des données

1 - addition :

0 + 0 = 0

0 + 1 = 1 + 0 = 1

1 + 1 = 0 avec retenue = 1

1 + 1 + 1 = 1 avec retenue = 1

exemple :

$$\begin{array}{r} 01110 \\ +\ 11100 \\ \hline 101010 \end{array}$$

2 - Soustraction :

0 - 0 = 0

1 - 0 = 1

0 - 1 = 1 avec '1' emprunté

1 - 1 = 0

exemple :

$$\begin{array}{r} 10110 \\ -\ 01100 \\ \hline 01010 \end{array}$$

3 - Multiplication :

0 x 0 = 0

1 x 0 = 0 x 1 = 0

1 x 1 = 1

exemple

$$\begin{array}{r} 10110 \\ \times\ 101 \\ \hline 10110 \\ +\ 00000. \\ +\ 10110.. \\ \hline 1101110 \end{array}$$

4 - division :

0 / 1 = 0

1 / 1 = 1

exemple

$$\begin{array}{r|l} 10010 & 10 \\ \cline{2-2} -10 & 1001 \\ \cline{1-1} 000 & \\ -00 & \\ \cline{1-1} 01 & \\ -00 & \\ \cline{1-1} 10 & \\ -10 & \\ \cline{1-1} 00 & \end{array}$$

- Représentation des entiers Signés :

a) SVA : Signe et Valeur absolue.

b) CA1 : complément à 1.

c) CA2 : complement à 2.

SVA : $(1101)_{SVA} = (-101)_2 = (-5)_{10}$

CA1 : $(1010)_{ca_1}$

$(+6)_{10} \rightarrow (+110)_2 \rightarrow (0110)_{SVA} \rightarrow (0110)_{ca_1}$

exemple opération sur 5 bits

1) $-13+4 = (-1101)_2 + (+0100)_2$

$= (11101) + (00100) \rightarrow SVA$

$= (10010) + (01011) \rightarrow ca_1 = (10110)_{ca_1}$

on a : $ca_1(ca_1(n)) = n$ $(11001)_{SVA}$

$= (-1001)_2$

$= (-9)_{10}$

2) $+13-4 = (+1101)_2 + (-0100)_2$

$= (01101) + (10100)_{SVA}$

$= (01101) + (11011)\ ca_1 =$

$$\begin{array}{r} {}^{1\,1\,1\,1} \\ 0\,1\,1\,0\,1 \\ +\ 1\,1\,0\,1\,1 \\ \hline = 1\,(0\,1\,0\,0\,0)_{ca_1} \end{array}$$

$(01001)_{ca_1}$

c'est un nombre positif : $(ca_1(n)) = n$

$= (01001)_{SVA}$

$= (+1001)_2 = (+9)_{10}$

complément à 2 :

exemple :

$(-5) \rightarrow -101_2 \rightarrow 1101_{SVA} \rightarrow 1010_{ca_1} \rightarrow 1011_{ca_2}$

$(+6) \rightarrow +110_2 \rightarrow 0110_{SVA} \rightarrow 0110_{ca_1} \rightarrow 0110_{ca_2}$

Addition (Soustraction) en ca_2 :

exemple : $(-13+4) = (-1101)_2 + (+0100)_2$

$= (11101)_{SVA} + (00100)_{SVA}$

$= (10010) + (00100)_{ca_1}$

$= (10011) + (00100)_{ca_2}$

$= (10111)_{ca_2}$

on a : $ca_2(ca_2(n)) = n$

$= (11000)_{ca_1} + 1$

$= (11001)_{SVA} = (-1001) = -9$

2) $+13-4 = (+1101)_2 + (-0100)_2$

$= (01101)_{SVA} + (10100)_{SVA}$

$= (01101)_{ca_1} + (11011)_{ca_1}$

$= (01101)_{ca_2} + (11100)_{ca_2}$

$= (01001)_{ca_2}$

$$\begin{array}{r} 01101 \\ +\ 11100 \\ \hline 1\,(01001)_{ca_2} \end{array}$$

une retenue négligée

C'est un nombre positif :

$(ca_2(n)) = n$

$= (01001)_{SVA}$

$= (+1001)_2 = (+9)_{10}$

Débordement - overflows :

$+6+5 \left(\begin{array}{r} 0110 \\ +\ 0101 \\ \hline 1011 \end{array} \right)_{ca_2}$

3 - Représentation des nombres réels :

✓ Virgule fixe.

✓ Virgule flottante : Utilisée actuellement sur machine, définie par : $\pm m \cdot b^e$ • un signe (+) ou (−)

• une mantisse (m) (en virgule fixe)

• un exposant (e) (un entier signé)

• une base (b) (2, 8, 10, 16, ...)

exemple : $(+3,25)_{10} = (+11,01)_2 = [(+0,1101 \times 2^2) = (+1,101 \times 2^1)$
$= (+110,1 \times 2^{-1}) = (+1101 \times 2^{-2})]$ en virgule flottante

*** Codage en virgule flottante Normalisée :**

SM	Eb	M
1 bit	p bits	q bits

• SM : Signe de la Mantisse.

• Eb : exposant biaisé.

• M : Mantisse Normalisée.

exemple : $11,01 = 1,101 \times 2^1$

le Standard IEEE 754 :

- Simple précision (32 bits) :

SM	Eb	M
1 bit	8 bits	23 bits

exemple : $(35,5)_{10} = (?)_{IEEE\,754}$

$(35,5)_{10} = (100011,1)_2$ (VF) $= (1,000111 \times 2^5)$ (V.flo)

$Eb = E + 2^{p-1} = 5 + 8^{-1} - 1 = 5 + 127 = 132 = (10000100)_2$

$1.M = 1,000111 \rightarrow M = 00011100000000000000000$

SM = 0 ; c'est un nombre positif

0 10000100 00011100000000000000000

- Double précision (64 bits)

SM	Eb	M
1 bit	11 bits	52 bits

$Eb = E + 2^{p-1} - 1 = 5 + 2^{11-1} - 1 = 5 + 1023$

Addition en IEEE : elle se fait en trois étapes

1). Dénormaliser les deux nombres afin d'avoir le même exposant (le plus élevé)

2). additionner les mantisses.

3). Normaliser le résultat.

exemple : $2,3 \times 10^5 + 4,6 \times 10^4$

$2,3 \times 10^5 + 0,46 \times 10^5 = (2,3 + 0,46) \times 10^5$

$$\begin{array}{r} 1,11 \\ +\ 0,1001 \\ \hline 10,0101 \end{array}$$

Multiplication en IEEE : elle se fait en quatre étapes :

1) Dénormaliser les deux nombres (exposants naturels).

2) additionner les exposants naturels.

3) Multiplier les mantisses.

4) Normaliser le résultat.

exemple : $3 \times 10^7 \times 2 \times 10^5 = 6 \times 10^{12}$

$$\begin{array}{r} 1,11 \times 2^2 \\ \times\ 1,0011 \times 2^3 \\ \hline 10,000101 \times 2^5 \end{array}$$

1- le code BCD : (binary coded decimal) :

713

111 001 011

* Addition en BCD : exemple : $(39 + 58)_{10} = (00111001 + 01011000)_{BCD}$

$$\begin{array}{rr} 0011 & 1001 \\ + \ 0101 & 1000 \\ \hline 1000 & 10001 \\ 1 & 0110 \\ \hline 1001 & 0111 \end{array}$$

→ Nombre > 9 donc ajout de 6

2- le code Gray :

000
001
011
010
110
111
101
100

* Conversion binaire → Gray $= N \oplus N/2$

$0+0=0$ $0+1=1$ et $1+1=0$

exemple : $(11001101)_2$

$= (10101011)_{Gray}$

* Conversion Gray → binaire

exemple : $(11001101)_{Gray}$

$(10001001)_2$

3 - le code ASCII :

Des caractères imprimables universels. lettres anglaises minuscules (65 à 90) et majuscules (97 à 122), chiffres, symboles...

exemple :

convertir le mot Codage en ASCII.

1000011 1101111 1100100 1100001 1100111 1100101

le codage UTF_8 :

	Nbre octets codant	format du code
2^7	1 0 a 127	0XXX XXXX
2^{11}	2 128 a 2047	110 XXXXX 10 XXXXXX
2^{16} 2048	3 a 65535	1110 XXXX 10 XXXXXX 10 XXXXXX
2^{21} 65 536	4 à 1114111	11110 XXX 10 XXXXXX 10 XXXXXX 10 XXXXXX

Exercices corrigés Arithmétiques binaires et transcodage

exercice 1 : effectuez les opérations suivantes en binaire :

- 110101,1101 + 11,011
- 100011,11 + 101,101
- 101101,11011 − 11101,111
- 110001,11 − 111,111
- 111000011,11101 × 1110,1
- 111,001 × 1,11
- 11011001,1101 / 101,1
- 100111011,1101 / 11,101

Solution :

$$
\begin{array}{r}
110101,1101 \\
+ \quad 11,011 \\
\hline
= 111001,0011
\end{array}
\qquad
\begin{array}{r}
100011,11 \\
+ \quad 101,101 \\
\hline
= 101001,011
\end{array}
$$

$$
\begin{array}{r}
101101,11011 \\
- \quad 11101,111 \\
\hline
= \text{[illegible]}
\end{array}
\qquad
\begin{array}{r}
110001,11 \\
- \quad 111,111 \\
\hline
=
\end{array}
$$

```
  1110000011,11101
×            110,1
------------------
  1110000 1111101
                0
+ 1110000 111110100
  11100001111101000
------------------
  10110111100 1011001
```

```
     111,001
×       1,11
------------
     111001
    111001•
   111001••
------------
   110001111
```

```
11011001,1101 | 101,1
              |------
```

=>

```
  110110011,101 | 1011
- 1011          |----------
  0010100       | 100111,10011
    1011
    10011
     1011
     10001
      1011
      01101
       1011
        10010
         1011
          1110
```

le code BCD :

exemple : le nombre est positif : +127

0000 0001 0000 0010 0000 0111 => code BCD

1100 000100100111 => packed BCD

- وعندما يكون العدد سالب في مكان الأصفار نضع "1" في BCD وفي packed BCD نضع 1101 .

le code gray

① Successeur et predeccesseur :

① 10010 => 10011

110110 => 110111

10101110 => 10101010

②

1101101 => 1101100

1101100 => 1100100

- 1Ko = 1000 octet (2^{10} bits) = 10^3 octets
- 1Mo = 1000 Ko = 1000 x 1000 octet = 10^6 octets
- 1Go = 1000 Mo = 1000 x 1000 Ko = 1000 x 1000 x 1000 octets = 10^9 octets
- 1To = 1000 Go = 1000 Go = 1000 x 1000 x 1000 x 1000 octets = 10^{12} octets
- peta, Exa, Zetta, Yotta)
- 4 bits est appelée quartet.
- 8 bits est appelée octet.

avec N bits, on peut représenter 2^N informations ou nombres,

$0 \leq N \leq 2^N - 1$

- pour 8 bits on peut représenter $2^8 = 256$ nombres : entre 0 .. 255

$$
\begin{array}{r|l}
10.11 & 100 \\
\cline{2-2}
-100 & 10,11 \\
\cline{1-1}
0011 & \\
-000 & \\
\cline{1-1}
110 & \\
-100 & \\
\cline{1-1}
0100 & \\
-100 & \\
\cline{1-1}
000 &
\end{array}
$$

!

- كلما جاء عدد جديد يجب عليه إيجاد رمز له لتمثيله
لحل هذه المشكلة إستعمال مجموعة محدودة من الرموز.

فائدة نظام العد الثنائي : سهولة تخزين المعلومات ونقلها