

A- Élément booléen : le bit.

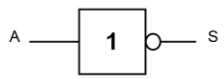
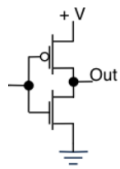

2 valeurs logiques possibles : '0' et '1'

Passant ou bloqué	Etats de fonctionnement d'un transistor dans un système numérique
'1' ou '0'	Utilisé pour représenter des nombres
HIGH ou LOW	Utilisé en électronique numérique : Niveaux logiques associés aux tensions.
Niveau haut ou niveau bas	
True ou False	Utilisé en informatique : valeurs des variables booléenne (vrai ou faux)

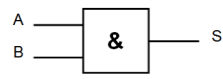
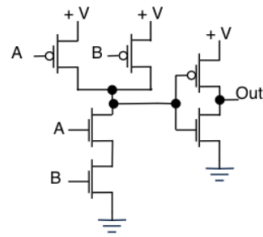

B- Les opérateurs élémentaires.

Fonction NOT.

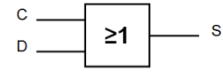
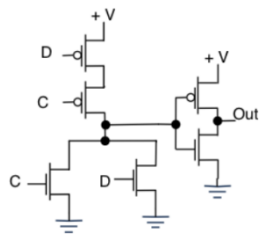
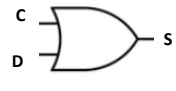
L'expression S est vraie si et seulement si A est fausse

Equation logique	Symbole européen	Table de vérité	Réalisation matérielle						
$S = \overline{e}$ se dit e barre		<table border="1"> <tr> <th>A</th> <th>S</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	S	0	1	1	0	
A	S								
0	1								
1	0								
Opération en python	Symbole américain								
$S = \text{not}(A)$									

Fonction AND

Equation logique	Symbole européen	Table de vérité	Réalisation matérielle															
$S = A \bullet B$		<table border="1"> <tr> <th>A</th> <th>B</th> <th>and</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	and	0	0	0	0	1	0	1	0	0	1	1	1	
A	B		and															
0	0		0															
0	1	0																
1	0	0																
1	1	1																
Opération en python	Symbole américain																	
$S = A \text{ and } B$																		

Fonction OR

Equation logique	Symbole européen	Table de vérité	Réalisation matérielle															
$S = C + D$		<table border="1"> <tr> <th>C</th> <th>D</th> <th>or</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	C	D	or	0	0	0	0	1	1	1	0	1	1	1	1	
C	D		or															
0	0		0															
0	1	1																
1	0	1																
1	1	1																
Opération en python	Symbole américain																	
$S = C \text{ or } D$																		



La fonction XOR (OU exclusif)

L'expression (E OUexclusif F) est vraie si et seulement si une et une seule des expressions C, D est vraie.

Equation logique	Symbole européen	Table de vérité	Réalisation matérielle (agencement de fonctions)															
$S = E \oplus F$		<table border="1"> <thead> <tr> <th>E</th> <th>F</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	E	F	S	0	0		0	1		1	0		1	1		On réalise la fonction en faisant la somme (fonction OU) des combinaisons égale à "1".
E	F		S															
0	0																	
0	1																	
1	0																	
1	1																	
Opération en python	Symbole américain																	
//																		

Ecrire la table de vérité. Retrouver l'agencement de fonctions permettant de la réaliser.

Application des opérations logiques sur des octets

$$\begin{array}{r}
 1011 \ 0110_{(2)} \\
 \text{and } 0011 \ 1100_{(2)} \\
 \hline
 _{(2)}
 \end{array}$$

$$\begin{array}{r}
 1111 \ 0101_{(2)} \\
 \text{or } 1101 \ 0101_{(2)} \\
 \hline
 _{(2)}
 \end{array}$$

$$\begin{array}{r}
 1011 \ 0110_{(2)} \\
 \text{xor } 0011 \ 1100_{(2)} \\
 \hline
 _{(2)}
 \end{array}$$

$$\begin{array}{r}
 1111 \ 0101_{(2)} \\
 \text{and } 1101 \ 0101_{(2)} \\
 \hline
 _{(2)}
 \end{array}$$