Circuits numériques

Logique combinatoire

Plan

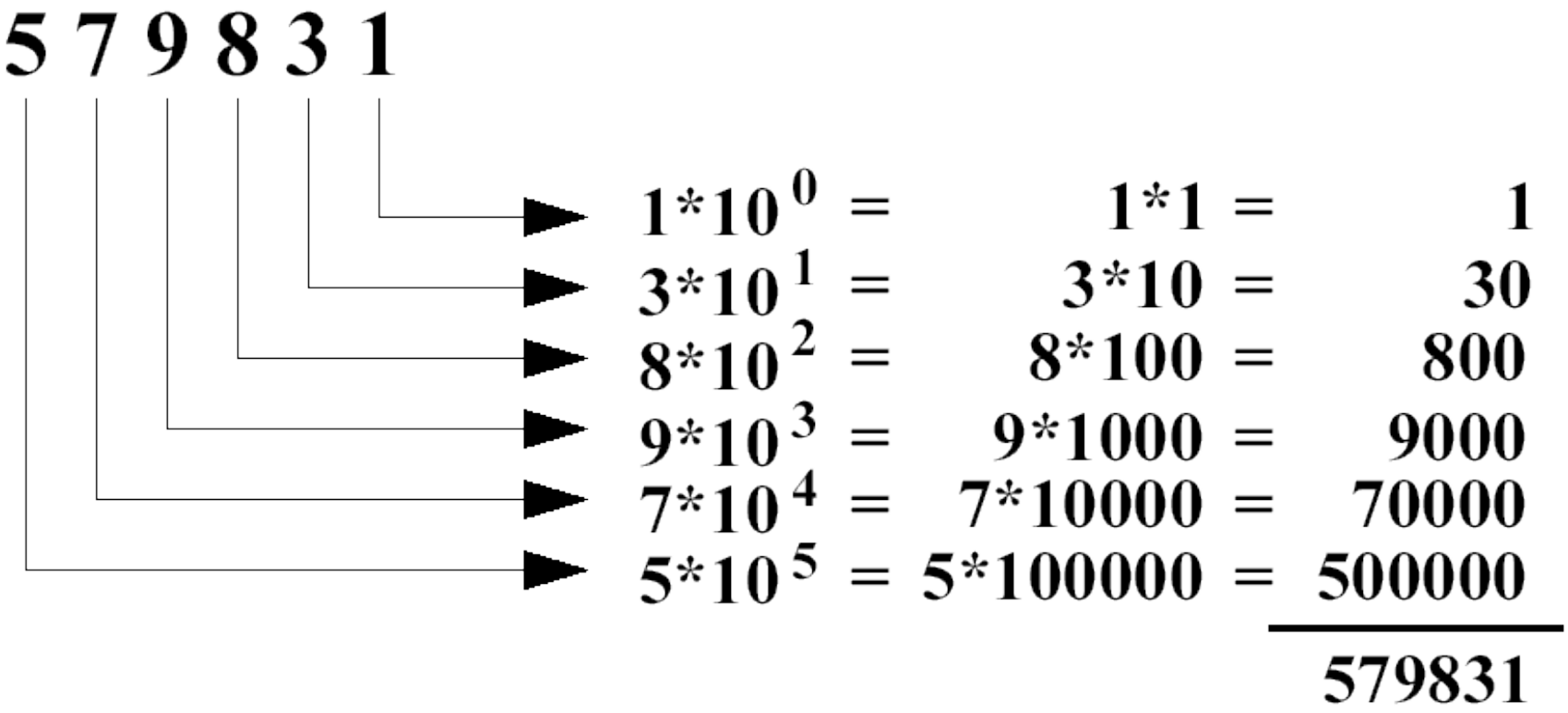
- De la numération et de l'Algèbre de Boole
- La logique combinatoire
  - Principes et notions fondamentales
  - Portes de base
  - Tableau de Karnaugh
  - Synthèse de fonctions logiques
- Des exemples de fonctions logiques

Des 1 et des 0

- Systèmes électroniques : communication par chaîne de BITs (Binary digiT) (John Tukey, popularisé par Shannon)
- L'information
  - Sa représentation : 0, 1
  - Son traitement : manipulation des 0 et des 1
- Réalisation physique de « 0 » et de « 1 »
  - électronique (tension, courant), optique (puissance optique, polarisation)
  - mécanique (levier, poulies), méca. flu. (fluides, vanne …)
  - Physique quantique (spin)
  - Objets : fiches perforées, pièces de monnaie

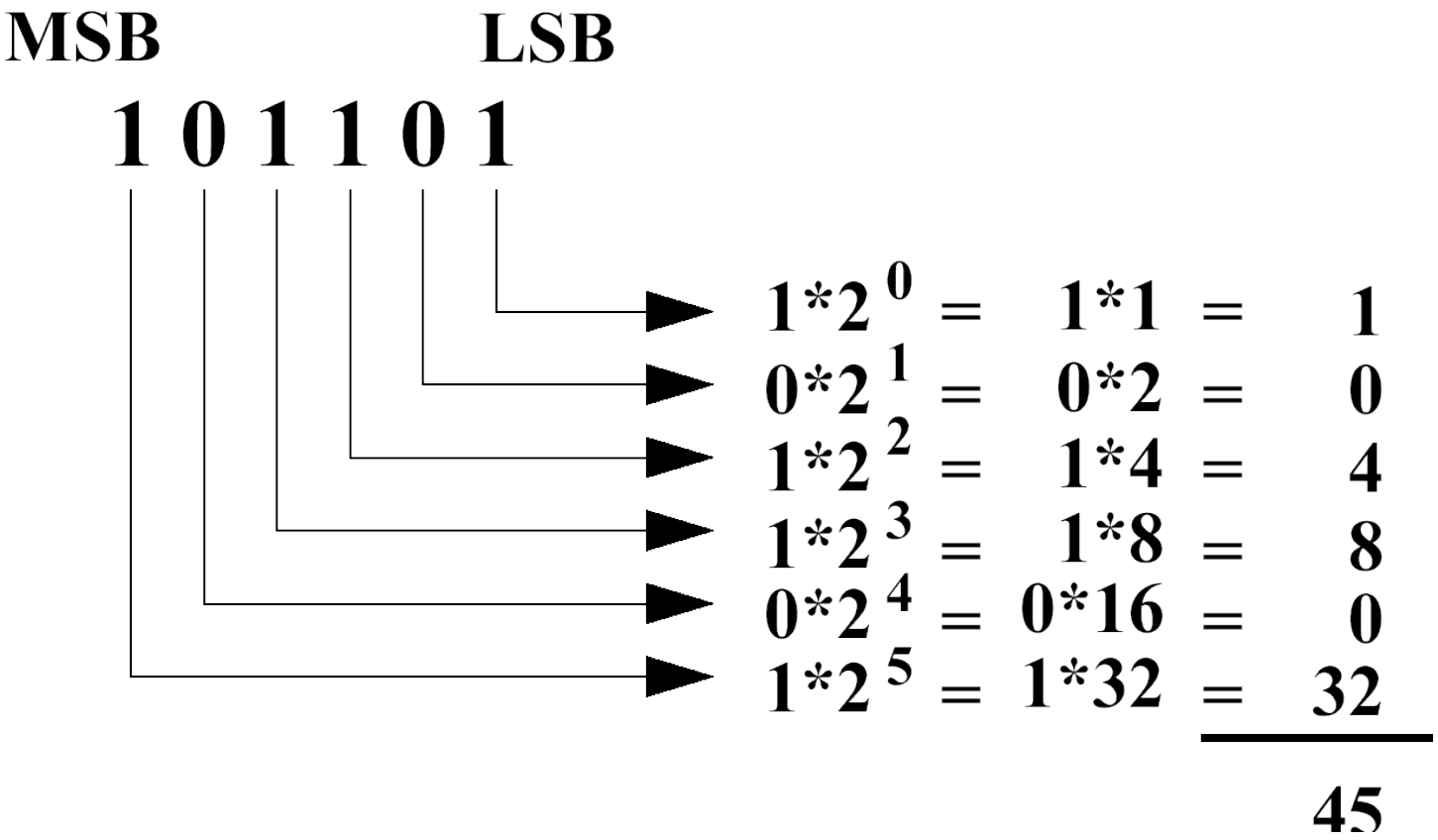
Représentation des nombres

- En base 10 : utilisée tous les jours
  - Élément de 0 à 9



Représentation des nombres

- En base 2 (1 ou 0)
  - Élément 0 ou 1



Codage des nombres

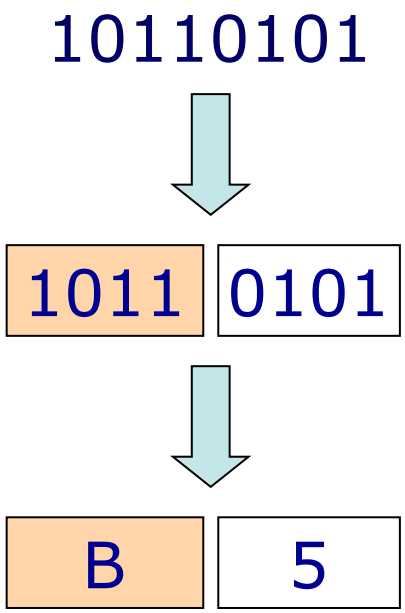
- octet (byte) = 8 bits
- mot (word) = 2 octets = 16 bits
- double mot (double word) = 2 mots = 32 bits

- kilo, mega, giga, tera … :
  - 1Ko = 2^{10} octets = 1024 octets
  - 1Mo = 2^{20} octets = 1024 ko = 1 048 576 octets
  - 1Go = 2^{30} octets = 1024 Mo = 1 073 741 824 octets
  - 1To = 2^{40} octets = 1024 Go = 1 109511627776 octets

Représentation des nombres

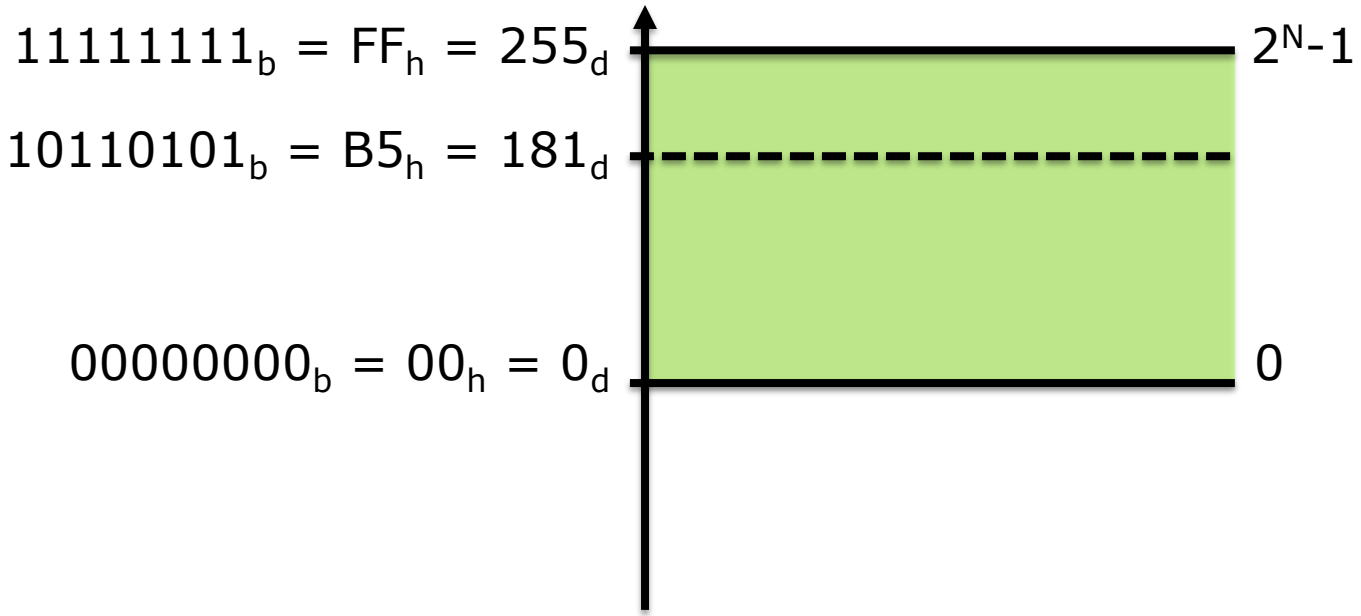
- En base 2 (1 ou 0) : binaire
- En base 16 (0, 1, 2 … D, E, F) : hexadécimal
  - représentation simple de groupes de données binaires
  - 1 quartet (2^4 combinaisons) = un caractère

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



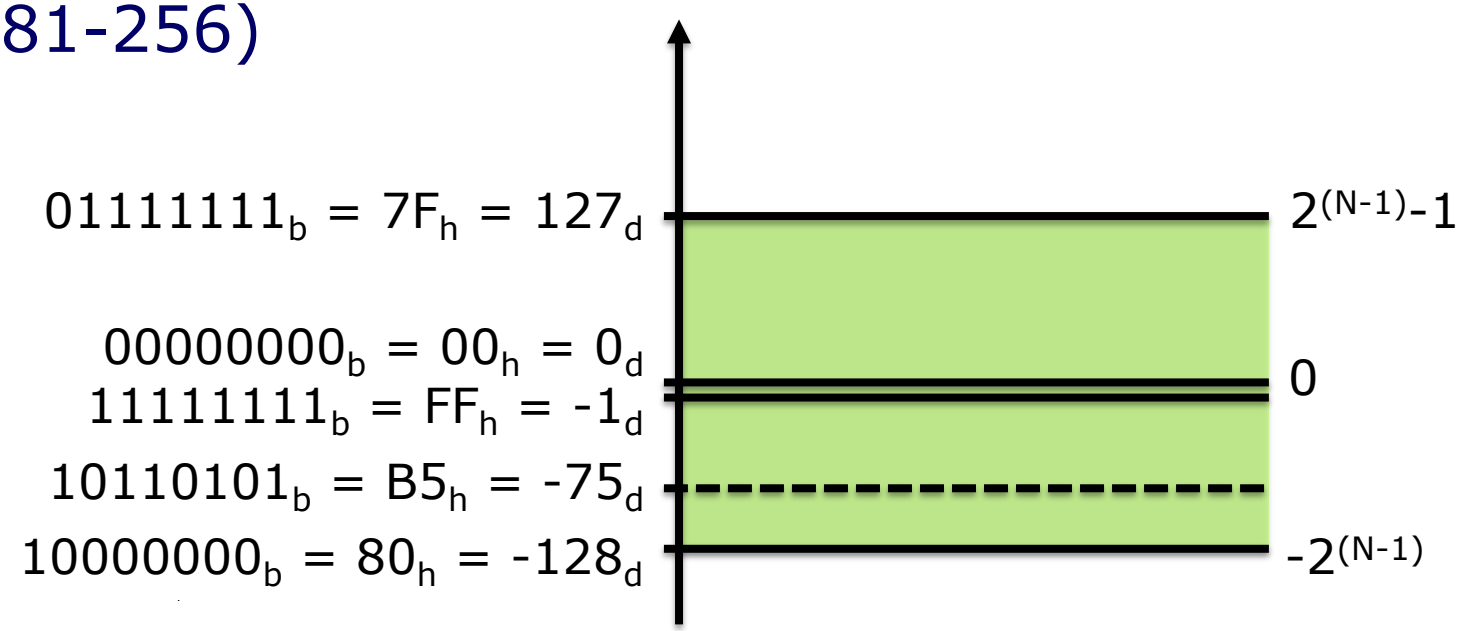
Virgule fixe

- Nombres entiers, non-signés
  - $0 \leq$ valeur $\leq 2^N-1$
  - $10110101 = 1*2^7+1*2^5+1*2^4+1*2^2+1*2^0 = 181$
  - $B5 = 11*16^1+5*16^0 = 181$
  - Erreur de quantification = $2^0/2$ (quantum, q)



Virgule fixe

- Nombres entiers, signés
  - MSB = bit de signe : 1=négatif, 0=positif
  - $-2^{(N-1)} \leq$ valeur $\leq 2^{(N-1)}-1$
  - $10110101 = 1*(-2^7)+1*2^5+1*2^4+1*2^2+1*2^0 = -75$
  - $B5 = 11*16^1+5*16^0$ (-2^N si MSB=1) = -75 (181-256)

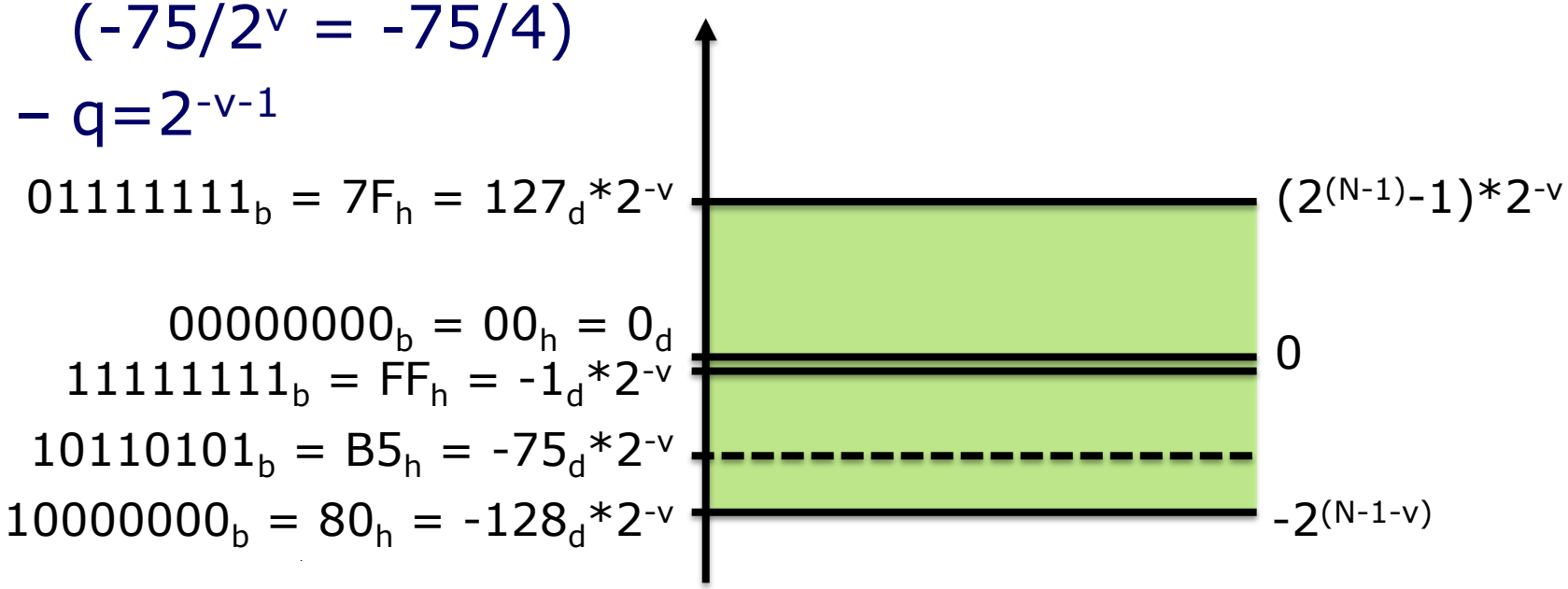


Complément à 2

- Représentation de nombres négatifs
- Elle exploite le nombre limité de bits (troncature à gauche) et permet la soustraction simple
- $-x = [2^N-x]-2^N$
  - 2^N = nombre non-représentable sur N bits
  - $2^N-x = /x + 1$
  - Ex.
    - $x=1_d=00000001_b$; $/x=11111110_b$; $/x+1 = 11111111_b$
    - $y-x = +4_d-1_d = 00000100_b + 11111111_b$ = (1 non représenté) $00000011_b = 3_d$
    - Complément à 10 : $-x=[10^N-x]-10^N$; ex. 40-1 = 40+[100-1] = (1 non représenté) 39

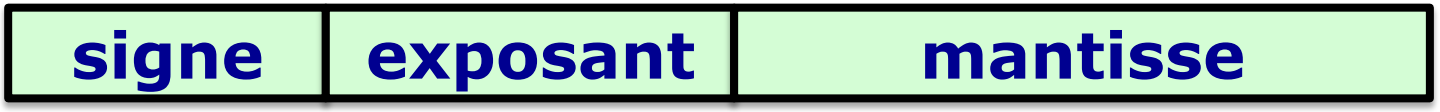
Virgule fixe

- Nombres fractionnaires, signés
  - Position de la virgule à v bits avant LSB
  - $-2^{(N-1-v)} \leq$ valeur $\leq (2^{(N-1)}-1)*2^{-v}$
  - v=2 : 10110101 (=101101.01) = $1*(-2^5)+1*2^3+1*2^2+1*2^0+1*2^{-2} = -18.75$
  - B5 = $11*16^1+5*16^0$ (-2^{N-v} si MSB=1) = -18.75 ($-75/2^v = -75/4$)
  - $q=2^{-v-1}$



Virgule flottante

- Nombre signé virgule fixe = mantisse, M
- Facteur de mise à l'échelle = exposant, E

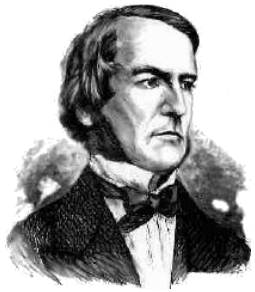


- Valeur = $M * 2^E$
- Permet une grande précision de représentation (échelle logarithmique grâce à l'exposant)
- Format typique : 32-bits, E sur 8 bits
- Ex. 8-bits, E sur 2 bits
  - 10110101 = 1 01 10101
  - $E=1_d$, $M=110101_b$=- /(110101 – 1) = -11
  - Valeur = $-11*2^1 = 22$

Algèbre de Boole

- Signaux numériques : 0 et 1
  - Nécessité d'une algèbre
  - Monde numérique : 2 valeurs, algèbre booléenne

- Boole : 1850



- Shannon : 1938 (machine à relais utilisant l'algèbre de Boole)

Algèbre de Boole

- Manipule des 0 et des 1
  - Résultat de même taille que les opérandes

  - Inversion (NOT)

    changement de valeur (complément)

A	S
0	1
1	0

  - ET logique (AND)

    A . B = 1 si A et B = 1

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

  - OU logique (OR)

    A + B = 1 si A ou B ou A et B = 1

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

Algèbre de Boole

- Idempotence

  A . A = A A + A = A

- Constantes

  A . 0 = 0 A + 0 = A A + 1 = 1 A . 1 = A

- Commutativité

  A . B = B . A A + B = B + A

- Complément

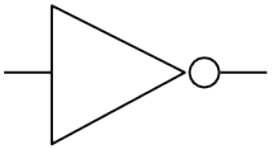
  A . /A = 0 A + /A = 1

- Théorème de De Morgan

  /A . /B = / (A + B) /A + /B = / (A . B)

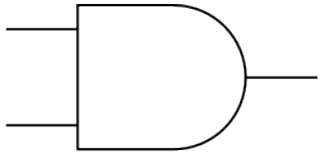
Portes logiques

- Inverseur (NOT)
- Non ET (NAND)
- Non OU (NOR)
- ET (AND)
- OU (OR)



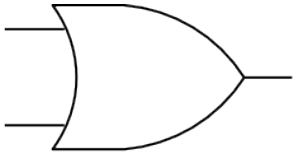
INVERSEUR

NOT



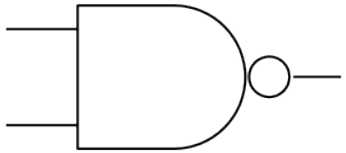
ET

AND



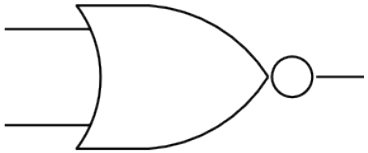
OU

OR



NON ET

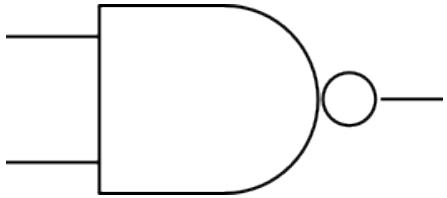
NAND



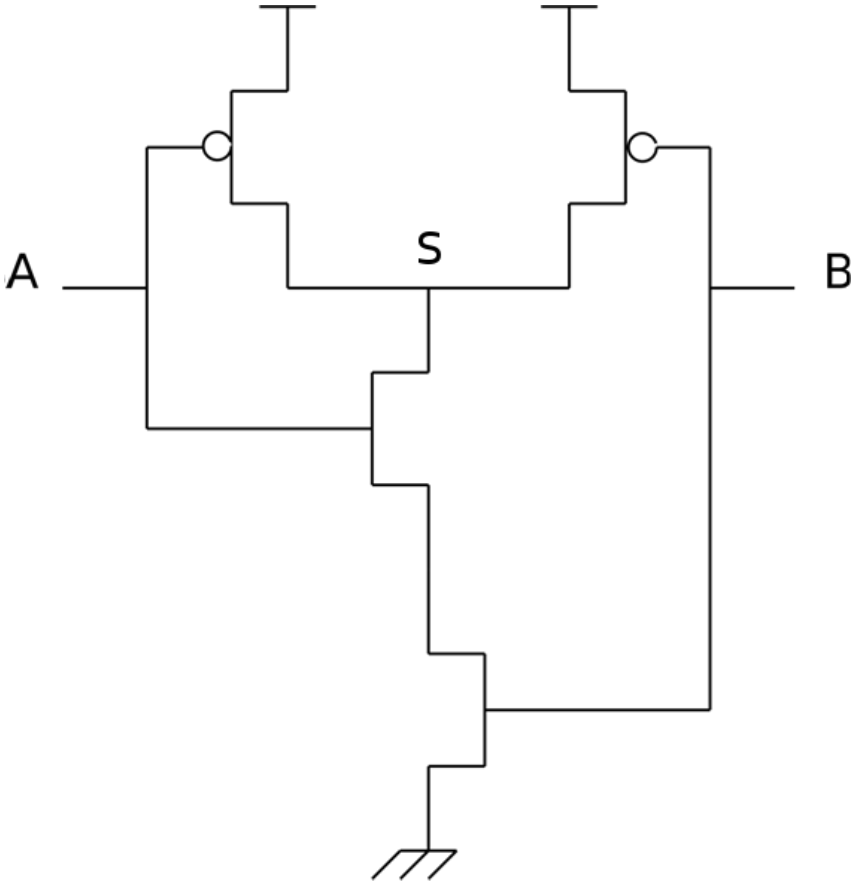
NON OU

NOR

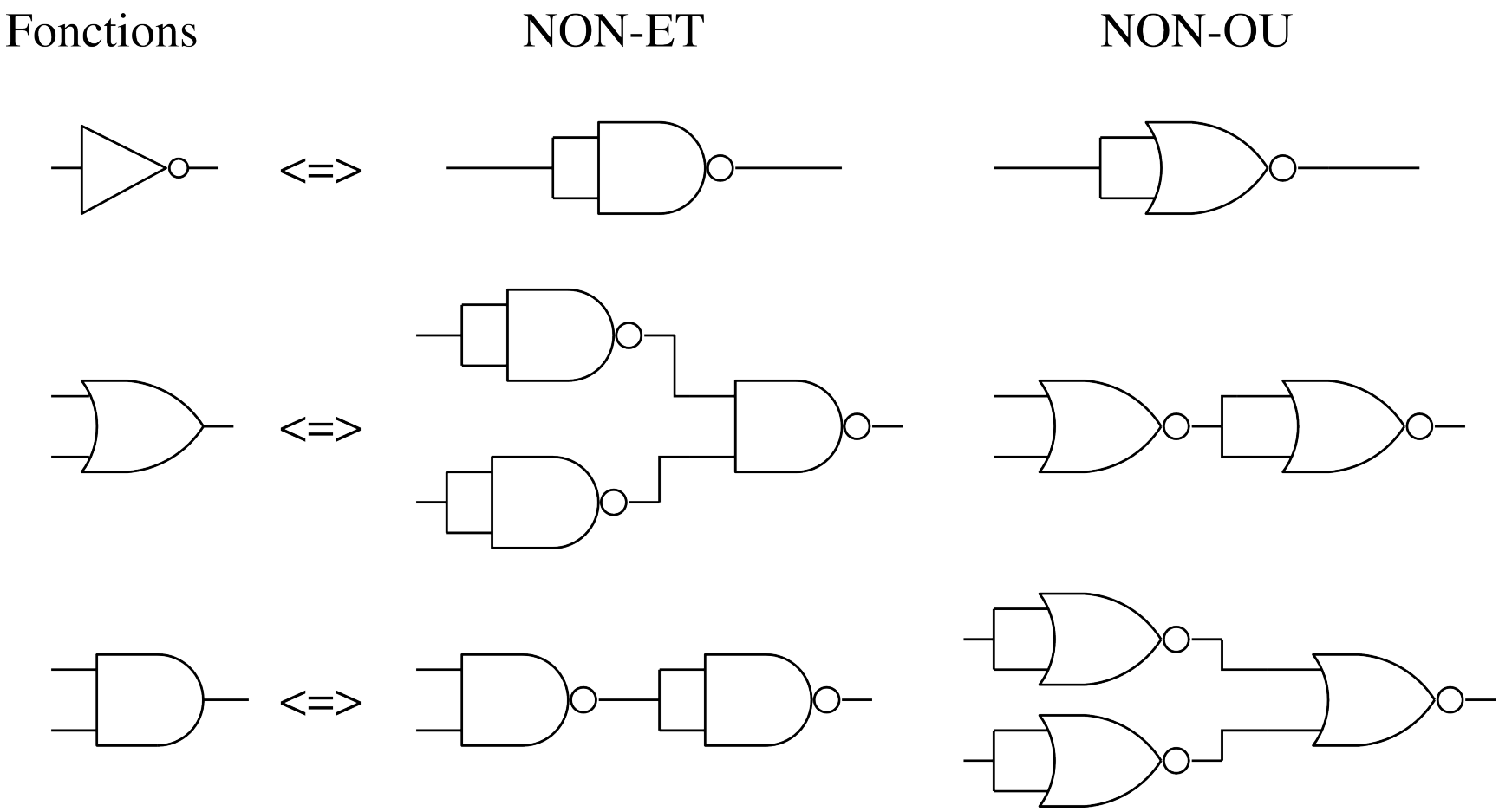
Portes logiques : niveau transistor



NON-ET (NAND)

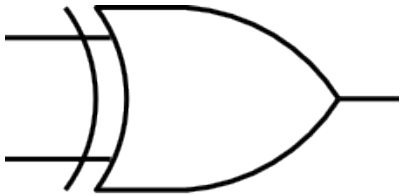


Portes logiques : équivalences



Autres portes logiques

- OU EXCLUSIF (XOR)



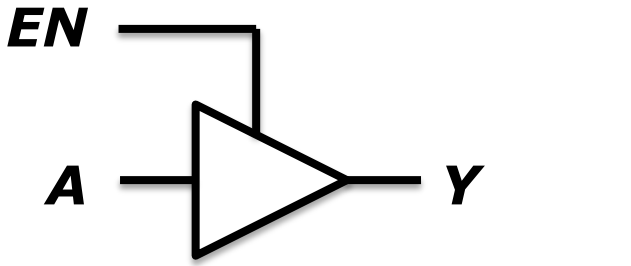
XOR

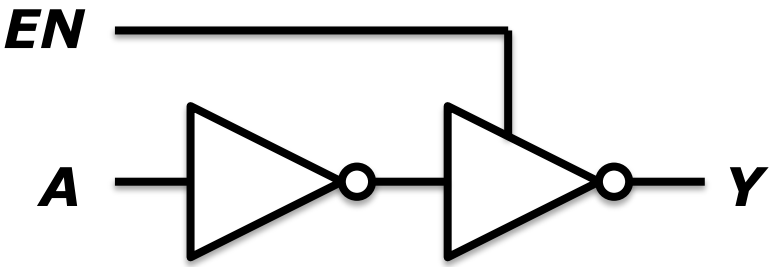
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$$S = A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

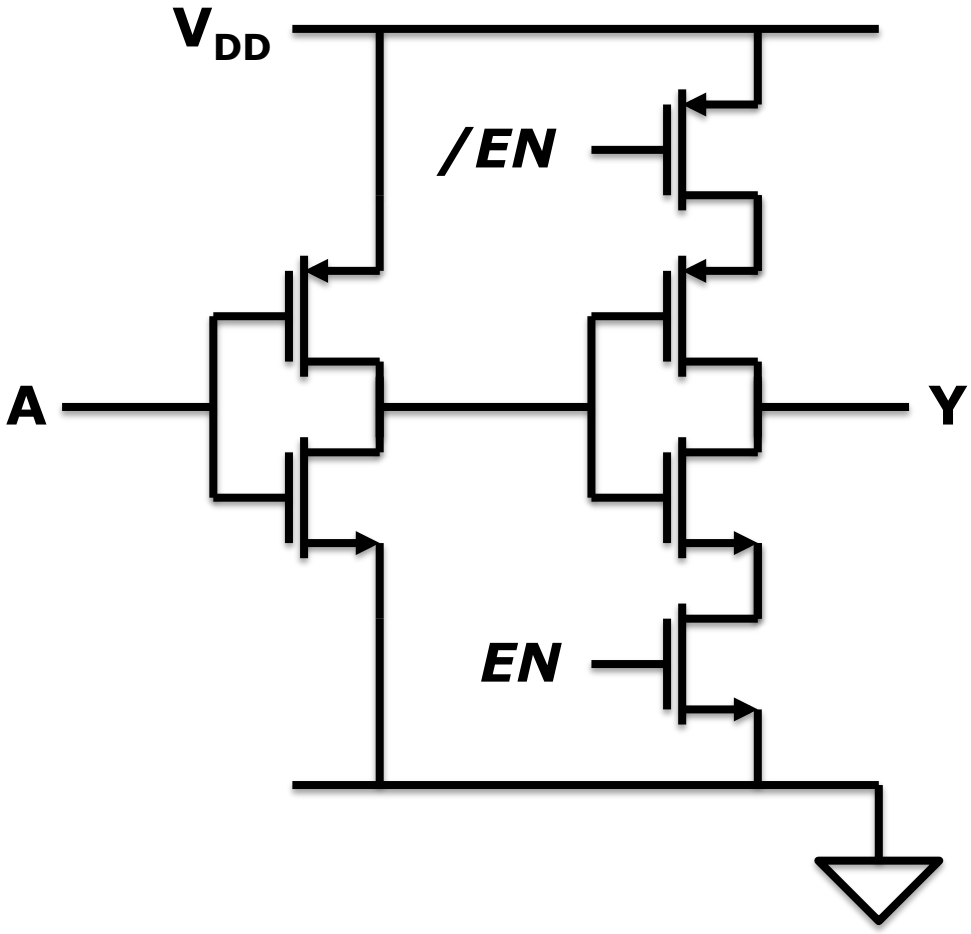
Autres portes logiques

- Buffer 3 états (0 / 1 / Z)
- Interrupteur avec regénération du signal



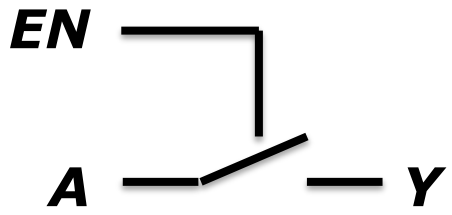


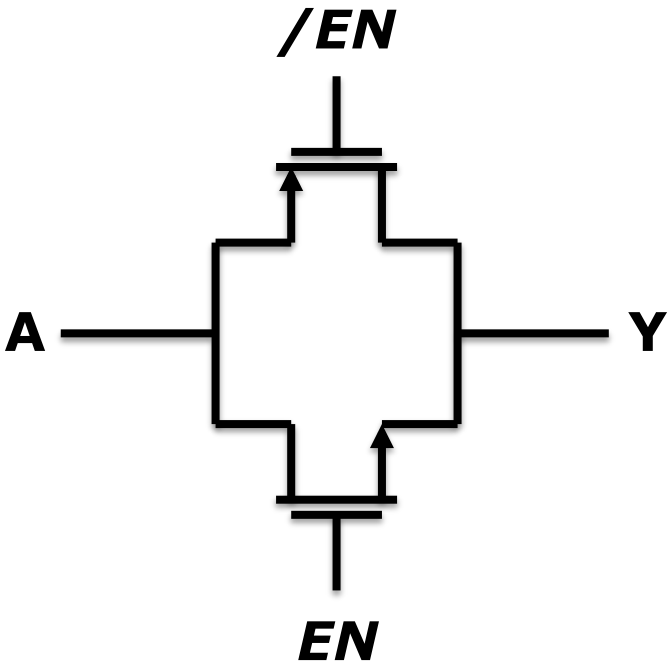
EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1



Autres portes logiques

- Porte de transmission
- Interrupteur sans regénération du signal



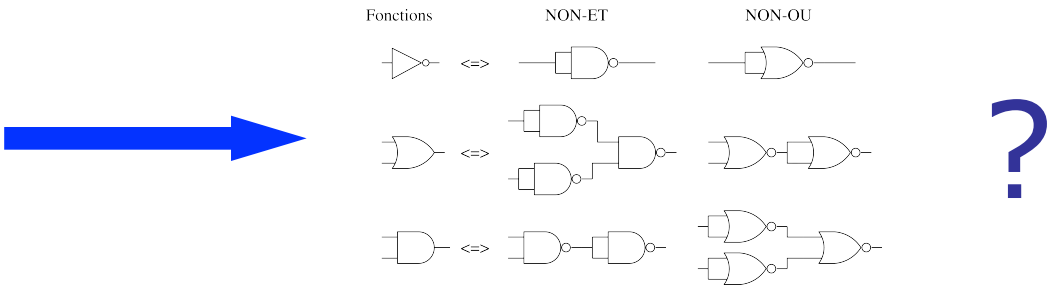


EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

Si EN=1 :
A=1 : PMOS passant
A=0 : NMOS passant
Si EN=0 :
A=X : PMOS, NMOS bloqués

Réalisation de circuits combinatoires

- Rechercher l'implémentation physique d'un problème
  - A x B



- Recherche d'une solution optimale en surface ou vitesse
  - Méthode:
    - Table de vérité : mise en forme du problème
    - Tableau de Karnaugh (ou algèbre !) : recherche d'équation minimale
    - Dessin du schéma électrique

Table de vérité: exemple de l'additionneur

- Ecrire les sorties en fonction des entrées (intuitif !)
  - Enoncé: addition binaire 3 entrées (donc il faut 2 sorties)

c_i	a	b	s	c_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Binaire codage naturel

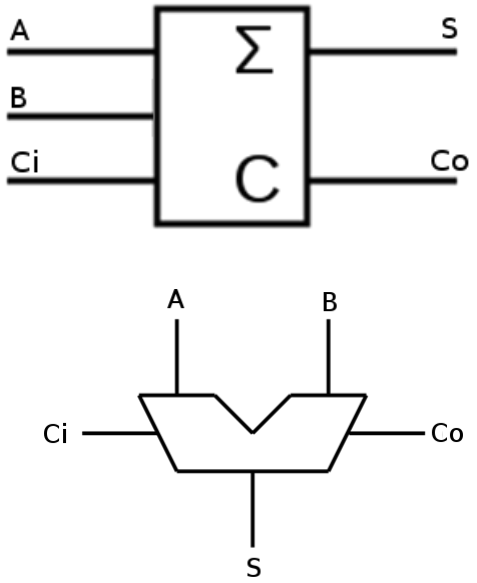


Tableau de Karnaugh: exemple de l'additionneur

- Pour chaque sortie, réécrire les valeurs différemment
  - Tableau de la sortie Co

Binaire codage gray

Ci \ B A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Valeurs de Co

  - Tableau de la sortie S

Ci \ B A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Valeurs de S

Tableau de Karnaugh: exemple de l'additionneur

- Faire les regroupements les plus grands possibles en base 2 entière (1,2,4,8…), Résolution sur les "1":
  - Tableau de la sortie Co

$$Co = Ci.A + Ci.B + A.B$$

Ci \ B A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Valeurs de Co

  - Tableau de la sortie S

Ci \ B A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Valeurs de S

$$S = Ci \cdot \overline{B} \cdot \overline{A} + \overline{Ci} \cdot \overline{B} \cdot A + Ci.B.A + \overline{Ci} \cdot B \cdot \overline{A} = Ci \oplus B \oplus A$$

Schéma électrique: exemple de l'additionneur

- Schéma libre (utilisation de toutes les portes de base)

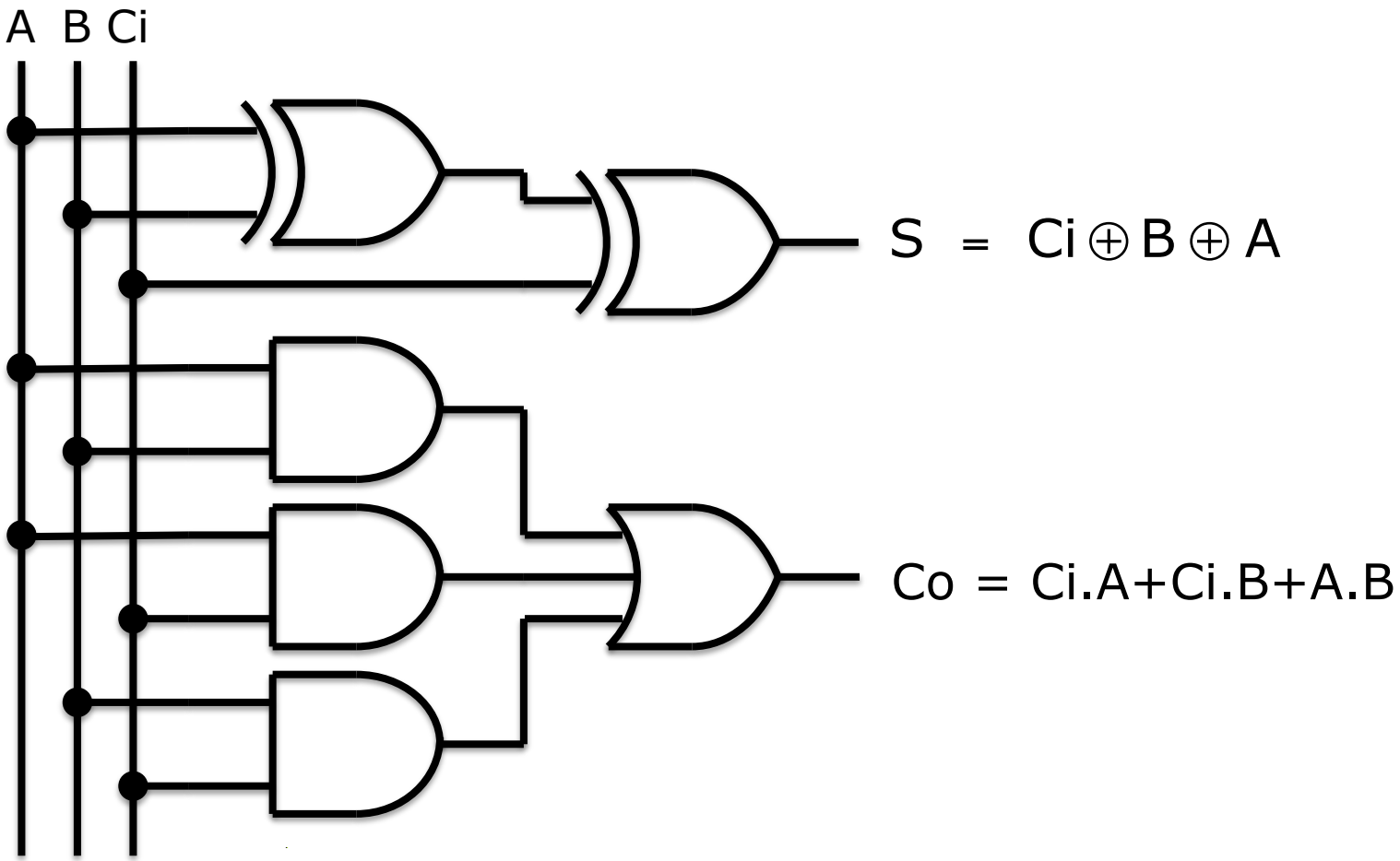


Schéma électrique: exemple de l'additionneur

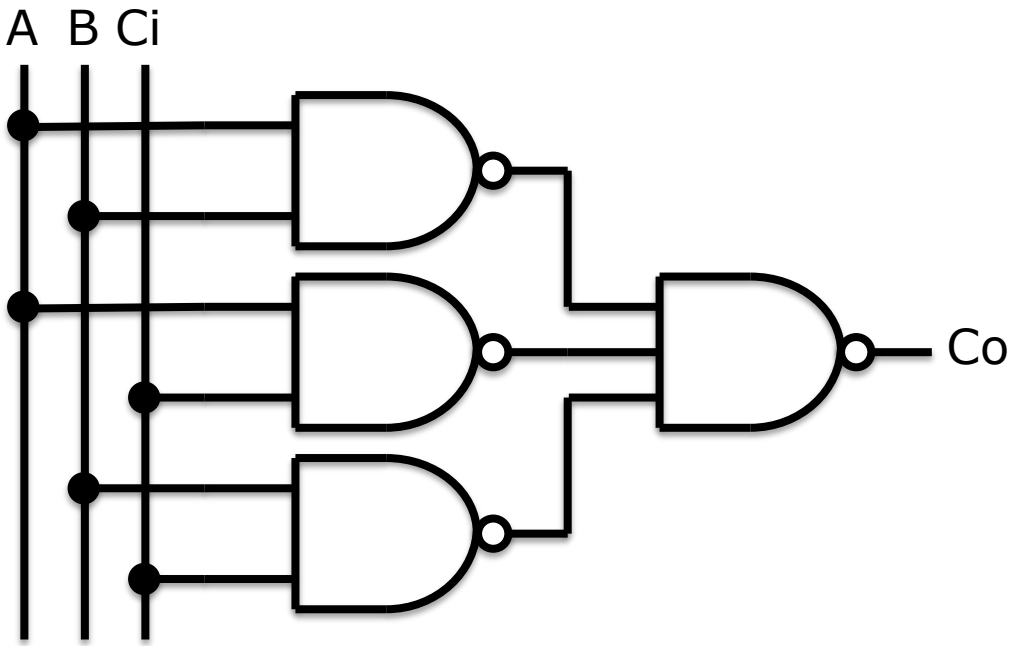
- Schéma imposé: utiliser uniquement des portes NAND
  - Théorème de De Morgan

$$S=\overline{\overline{Ci}\cdot B\cdot A+Ci\cdot\overline{B}\cdot A+Ci.B.A+Ci\cdot B\cdot\overline{A}}$$

$$S=\overline{\overline{Ci}\cdot B\cdot A.Ci.\overline{B}\cdot A.\overline{Ci.B.A}\,.\,\overline{Ci\cdot B\cdot A}}$$

$$Co=\overline{Ci.A+Ci.B+A.B}$$

$$Co=\overline{Ci.A.Ci.B.A.B}$$



Etat X : don't care

- X vaut 1 ou 0, sans incidence sur le résultat
- Ex. détecter les nombres premiers sur les nombres 0,1,2,3,4,5. On doit donc utiliser 3 bits pour coder les nombres possibles

E	E2	E1	E0	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	X
7	1	1	1	X

← **N'arrive jamais car ne fait pas partie des nombres d'entrée !**

Etat X : don't care

- Exemple: détecter les nombres premiers sur les nombres 0,1,2,3,4,5. On doit donc utiliser 3 bits pour coder les nombres

E2 \ E1 E0	00	01	11	10
0	0	1	1	1
1	0	1	0	0

sans X

$$S=\overline{E_1}\cdot E_0 + \overline{E_2}\cdot E_0 + \overline{E_2}\cdot E_1$$

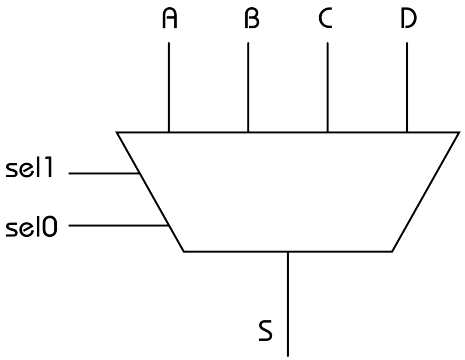
E2 \ E1 E0	00	01	11	10
0	0	1	1	1
1	0	1	X	X

avec X

$$S=E_0+E_1$$

Autres circuits utiles

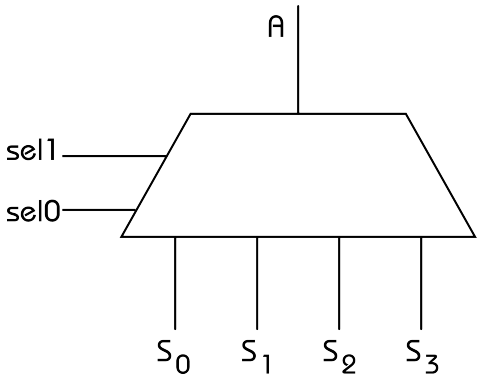
- Multiplexeur (MUX), Démultiplexeur (DEMUX)



{sel1, sel0}
permettent de
sélectionner 1 entrée
parmi 4

(MUX)

S = A , B , C ou D
selon sel1 sel0



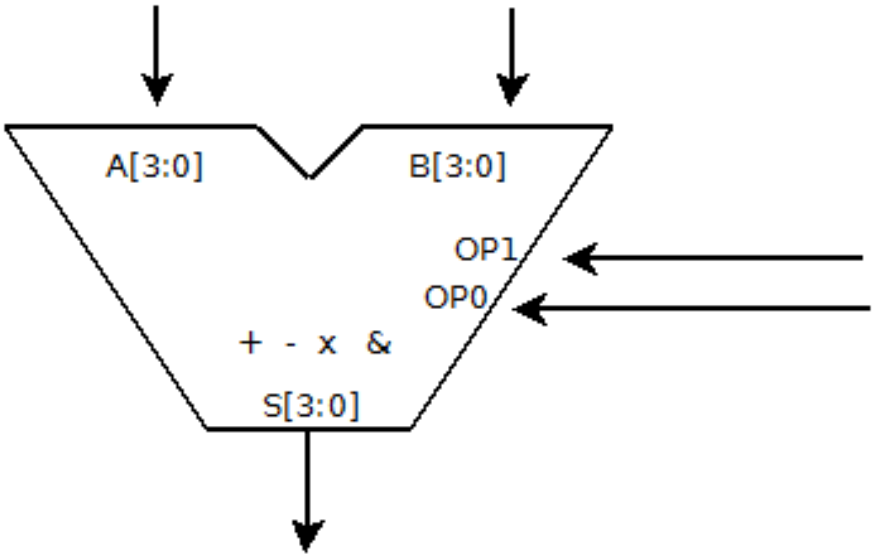
{sel1, sel0}
permettent de
sélectionner 1 sortie
parmi 4

(DEMUX)

A => S_0 , S_1 , S_2 ou
S_3 selon sel1 sel0

Autres circuits utiles

- Unité Arithmétique et Logique (UAL, ALU)



OP1	OP0	S
0	0	A+B
0	1	A-B
1	0	AxB
1	1	A.B

Bibliographie

- Logique combinatoire et séquentielle, C. Brie, coll. Technosup, Bibli ECL: 005.131 BRE
- Introduction aux circuits logiques, J. Letocha, McGraw-Hill, 1985