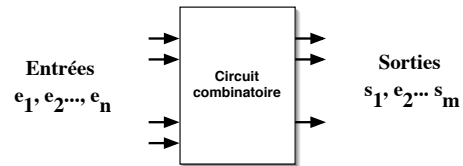


## Les circuits combinatoires

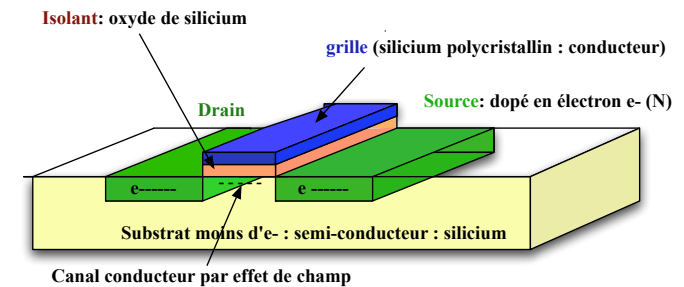
- Comment réaliser des fonctions booléennes à l'aide de circuits électroniques ?
- Dispositifs physiques à  $n$  entrées,  $m$  sorties



- Chaque sortie réalise une fonction booléenne de  $n$  variables
- $s_i = f_i(e_1, e_2, \dots, e_n)$
- 0 volt : 0 logique, 5 volts : 1 logique (aujourd'hui plutôt 2 à 3 Volts)
- Si on met des tensions électriques de 0 ou 5 volts sur les fils d'entrées, le fil de sortie porte la tension correspondant à la valeur

## L'élément de base : le transistor MOS

- MOS : Metal Oxide Semi-conducteur
- Un interrupteur miniaturisé
- Superposition de matériaux isolants, conducteurs ou semi-conducteurs
- Tension sur la grille: effet de champ électrique: formation d'un canal conducteur entre le drain et la source



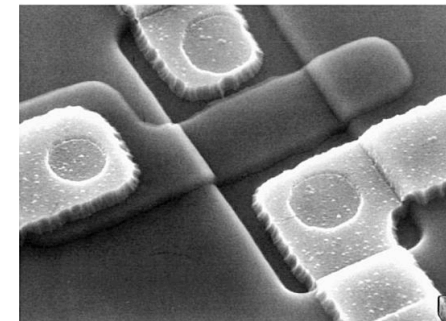
Coupe d'un Transistor NMOS

## Fonctionnement du transistor MOS

- Grille = 0 volts alors le drain et la source sont isolés électriquement
- Grille = 5 volts un canal conducteur se forme entre le drain et la source (électrons du substrat sont attirés sous la grille par effet de champ)
- Equivalent à un interrupteur
- Transistor PMOS fonctionne "à l'envers": passant si la grille est à 0 volt

## Caractéristiques d'un transistor

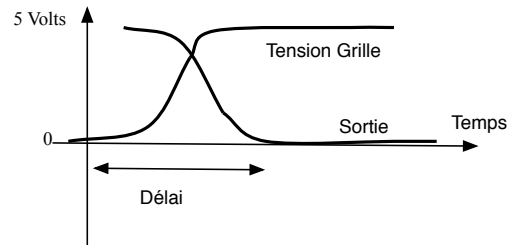
- **Taille :**  
Largeur de la grille : en micron ( $10^{-6}$  mètres, millième de millimètre)
  - Technologie actuelle :  $\sim 0,05$  micron = 50 nanomètres
  - Processeur : plus d'un Milliard de transistors sur 3 ou 4 cm<sup>2</sup>



## Caractéristiques

### • Temps de stabilisation

- Il faut du temps entre le moment où la tension sur la grille passe de 0 à 5v et le moment où le transistor devient vraiment passant
- En pico-seconde ( $10^{-12}$ )

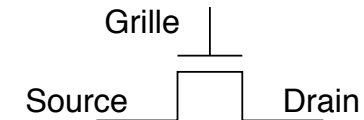


### • Consommation

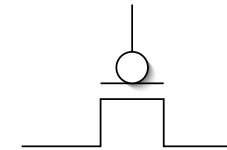
- Dépend des dimensions du transistor (largeur et longueur grille, drain...)

## Représentation symbolique

### • Transistor NMOS

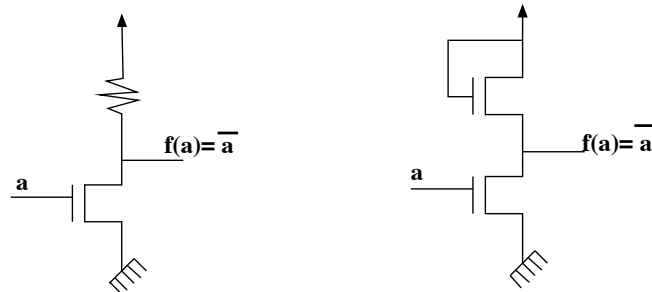


### • Transistor PMOS



## Réalisation de fonctions booléennes à l'aide de transistors MOS

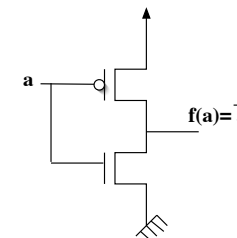
- Réalisation de la fonction  $f(a) = \bar{a}$ , l'inverseur
- Technologie NMOS



- $a = 5$  volts : transistor passant  $f(a) = 0$  volt
- $a = 0$  volt : transistor bloqué:  $f(a) = 5$  volts
- Sortie dégradée à cause de la résistance

## Technologie CMOS Complementary MOS

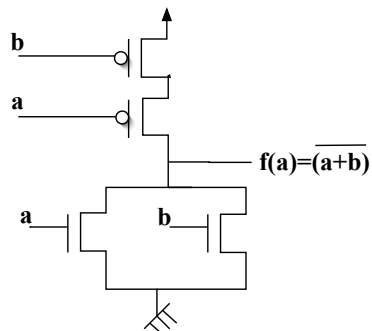
- On utilise des transistors NMOS et PMOS
- Exemple: l'inverseur



- Quand le transistor NMOS est passant, le PMOS est bloqué et inversement

## Autres exemples

- La fonction NOR (Non OU)



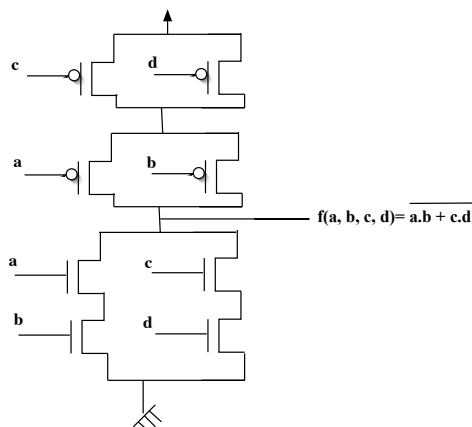
- Le réseau de transistors NMOS est le le dual du réseau PMOS

## Constructions de portes élémentaires

- Réseau de PMOS/réseau de NMOS
- Comment obtenir ces réseaux à partir de la fonction booléenne ?
- Sur le réseau NMOS :
  - transistors en parallèle  $\Leftrightarrow$  somme booléenne
  - transistors en série  $\Leftrightarrow$  produit booléen
  - Le réseau de transistor NMOS est équivalent au complément de la fonction à réaliser
- Le réseau PMOS se déduit du NMOS:
  - Transistors en parallèle / Transistor en série (et inversement)
- Exemple: le NON-AND: NAND
  - Réseaux NMOS : 2 transistors en série
  - Réseaux PMOS : 2 transistors en parallèle

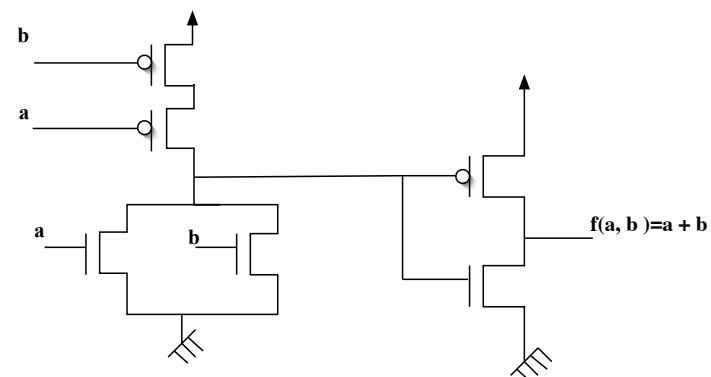
## Exemple

- La fonction  $f(a,b,c,d) = \overline{ab+cd}$
- Réseaux NMOS:
  - 2 transistors en série avec a et b
  - 2 transistors en série avec c et d

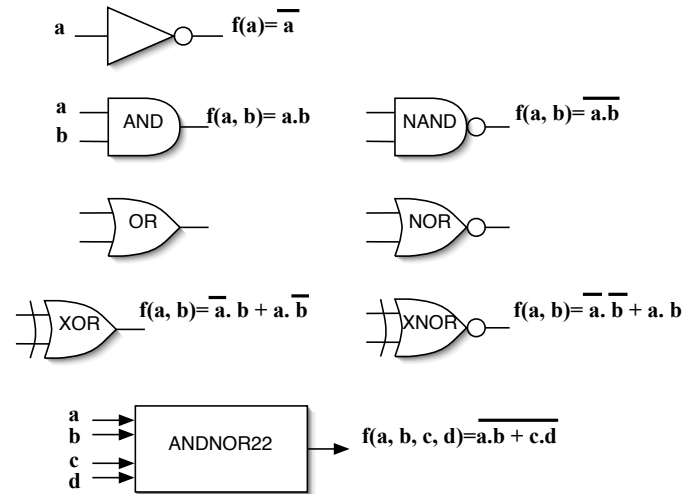


## Les portes AND et OR ?

- On échange la masse et l'alimentation en 5 volts ?
- D'un point de vue électronique cela ne marche pas bien
- On utilise donc un NOR suivi d'un inverseur pour réaliser le OR (idem pour le NAND)

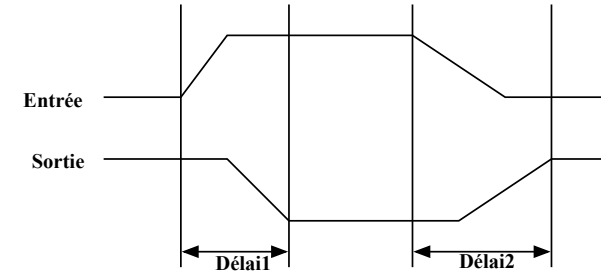


## Portes logiques de base et symboles utilisés



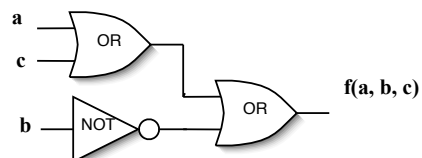
## Délai d'une porte

- Délai entre le changement des entrées et de la sortie
- Exemple un inverseur



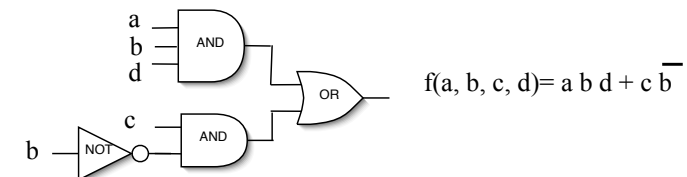
## Exemple de réalisation d'une fonction booléenne

- Bibliothèque de portes de base que le constructeur sait réaliser
- Exemple : AND2, OR2, INVERSEUR
- Soit la fonction :  $f(a,b,c) = a \cdot \overline{b} + a \cdot b + a \cdot \overline{c} + \overline{b} \cdot c$
- Minimisation:  $f(a,b,c) = a + c + \overline{b}$
- Adaptation à la bibliothèque (Mapping):
  - $f(a,b,c) = (a+c) + \overline{b}$
- Dessin du circuit à partir des portes de la bibliothèque

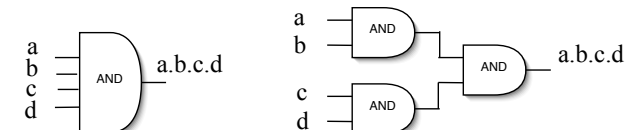


## De la Forme polynomiale aux portes AND, OR, INVERSEUR

- Un monôme à n variables: une porte AND à n entrées
- Une somme de k monômes : une porte OR à k entrées

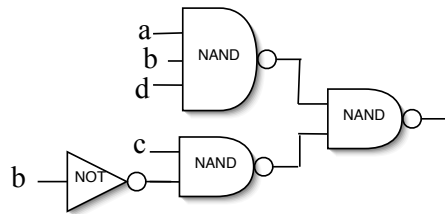


- Si le nombre d'entrée des portes est insuffisant :
  - décomposition possible :  $a \cdot b \cdot c \cdot d = (a \cdot b) \cdot (c \cdot d)$



## De la Forme polynomiale aux portes NAND, INVERSEUR

- $F = m_1 + m_2 + \dots + m_k = \overline{\overline{m_1 + m_2 + \dots + m_k}}$
- Par De Morgan:  $F = \overline{\overline{m_1} \cdot \overline{m_2} \cdot \dots \cdot \overline{m_k}}$
- Un monôme -> un NAND
- Exemple:  $f(a,b,c,d) = a.b.d + c.\overline{b} = \overline{\overline{a.b.d} \cdot \overline{c.\overline{b}}} = \overline{\overline{a.b.d} \cdot \overline{c} \cdot \overline{\overline{b}}}$



## Procédés de fabrication des transistors

- **Historique:**
  - Avant 60 : premiers circuits de calcul : relais pneumatique puis diodes
  - Fin 60 : quelques portes < 100 transistors
  - Début 70 : <1000 transistors, dessin à la main des transistors
  - Début 80 : <500.000 : naissance des outils de CAO (fonctions booléennes -> dessins des transistors de façon automatique)
  - Actuellement : centaines de millions de transistors

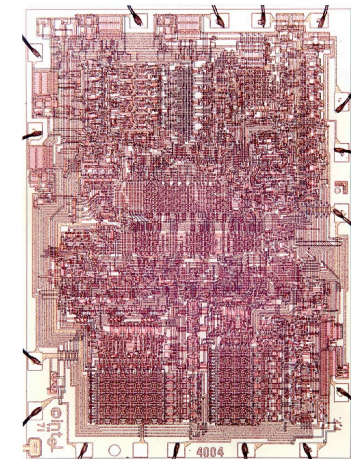
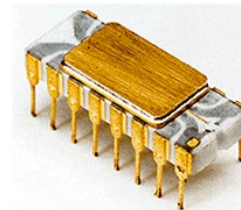
## Procédés de fabrication

- Galette de Silicium (semi-conducteur) de 10 à 40 cm de diamètre
- Dessin des masques correspondant aux différentes zones des Transistors PMOS et CMOS (grille, caisson P et N, fils de liaisons)



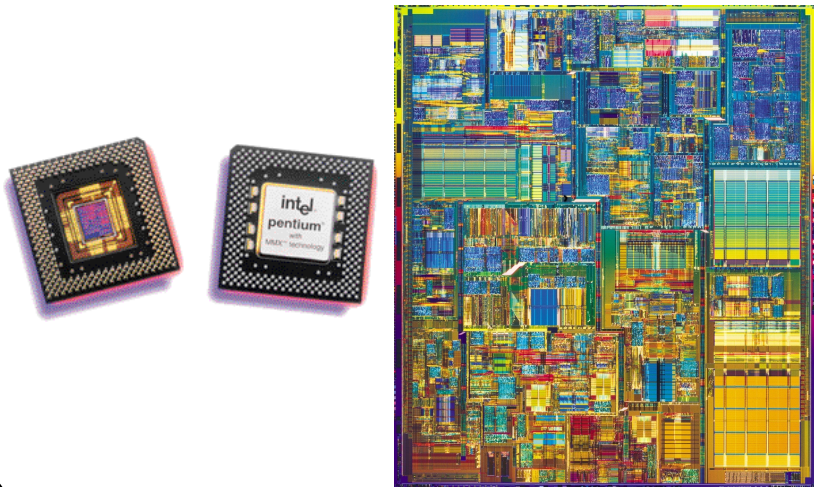
## Exemple de processeur

- **Le premier 1969: le Intel 4004**  
-2300 transistors, données de 4 bits, 108 khz



## Exemple de processeur

- Plus récent 2000: Le Pentium 4 d'Intel
- ~50 millions de transistors, données de 32 bits, ~3 ghz



## Fabrication d'un transistor

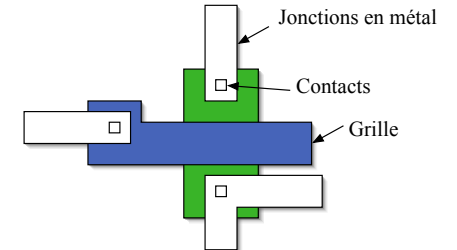
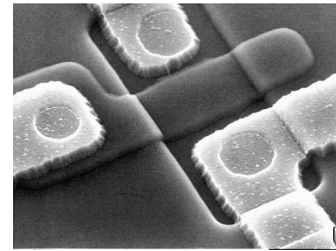
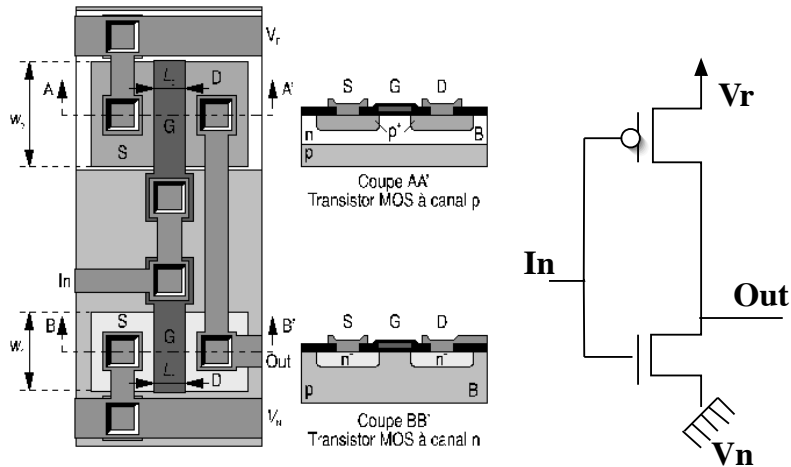


Photo au microscope électronique d'un transistor

Dessins du transistor (Layout)

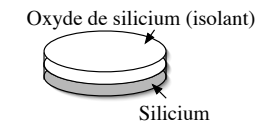
- Pour fabriquer un transistor il faut dessiner des rectangles
- Masques du circuits

## Exemple: dessin des masques d'un inverseur

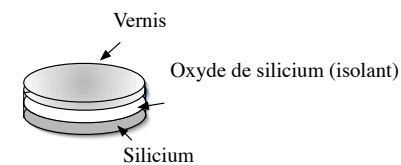


## Etapas de fabrication Chimique et photographique

- On part d'une galette de Silicium (Wafer)
- Traitement en "salle blanche"
- 1- Ajout d'une couche d'oxyde de silicium par procédé chimique

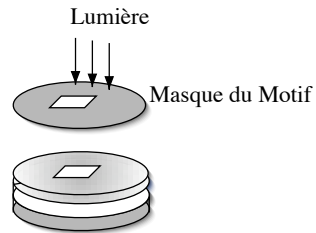


- 2- Ajout d'un vernis photosensible

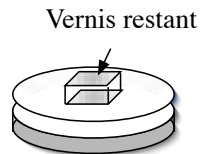


## Etapas de fabrication

- 3- Exposition à la lumière à travers un masque où sont dessinés les motifs (des grilles par exemple)

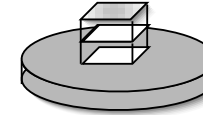


- 4- Suppression du vernis non exposé à la lumière par un bain chimique

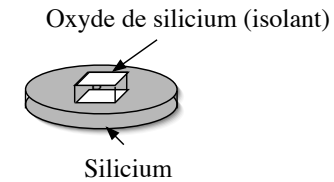


## Etapas de fabrication

- 5- Attaque de l'oxyde de silicium par un bain chimique auquel le vernis résiste

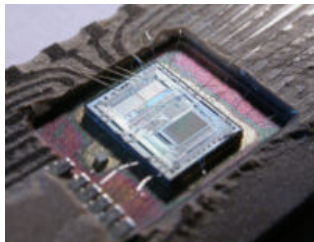


- 6- Suppression du vernis par un bain chimique particulier



## Etapas de fabrication

- Dopage en ions métalliques du silicium non recouvert par l'oxyde de silicium.
- On peut répéter ce type d'opérations pour les autres couches du transistor : silicium polycristallin, connexion en aluminium...
- Etapas finales:
  - Tests des circuits sur le Wafer
  - Découpage du Wafer
  - Mise en boîte avec connexion des broches d'entrées/sorties
  - Test de validation des circuits en fréquence et température



## Assemblage des puces

- Sur des cartes imprimées
  - Liaison dans la carte en aluminium
  - Plusieurs couches pour le croisement des fils
  - Soudure des puces sur la carte



## Technologies existantes

- ASIC Application Specific Integrated Circuit : circuit spécifique à une application (téléphone, machine à laver..) en opposition avec les circuits standards des ordinateurs (processeur, mémoire ...)
- Full Custom : construction à grande échelle (exemple: un processeur) Dessins et assemblage des transistors à la demande pour répondre à des contraintes de surface, délais, consommation...
- Half Custom (portes précaractérisées): Bibliothèques de portes standards déjà dessinées. Les dessins des masques des portes existent déjà. On les choisit et les relie suivant l'application.
- Circuit programmable: construction à petite échelle. Circuit déjà fabriqué dont on programme la logique : fonctions booléennes basique dans des EPROMs, on fixe les liaisons entre ces EPROMs
  - Circuit programmable une seule fois
  - Circuit programmable à volonté (lors de la mise sous tension, programmation à partir d'une ROM)

## Le flot de conception (De la fonction booléenne au circuit)

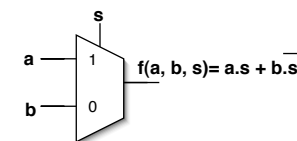
- Outils de CAO (Conception Assisté par Ordinateur)
- Programmes permettant d'aller de façon quasi-automatique des fonctions booléennes au circuit
- Les différentes étapes en Half-Custom:
  - 1- Synthèse:
    - Minimisation
    - Mapping: Adaptation à la technologie (suivant la bibliothèque de portes par exemple)
    - Simulation et vérification
  - 2- Placement et routage des portes standards
  - 3- Simulation
  - 4- Estimation des délais et de la surface
  - 5- Fabrication des masques et gravure proprement dite

## Synthèse de haut-niveau

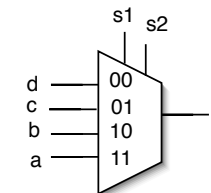
- Aujourd'hui on peut donner une description du circuit dans des langages adaptés (Standard: VHDL)
  - Description des Fonctions Booléennes (table de vérité, expression algébrique ...)
  - Automates d'états finis
  - Et même Algorithmes
- Exemple :
  - `add(a,b: int) returns resultat:int`  
`resultat= a+ b;`
  - Fabrication automatique d'un additionneur
- En TP pour comprendre on partira toujours des portes logiques de base (opérateurs booléens)

## D'autres circuits usuels

- Le multiplexeur 2 vers 1:



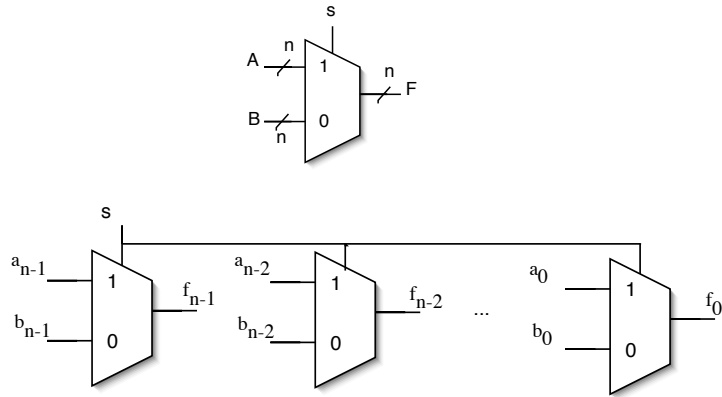
- Le multiplexeur 4 vers 1:



$$f(a, b, c, d, s1, s2) = a.s1.s2 + b.s1.\bar{s2} + c.\bar{s1}.s2 + d.\bar{s1}.\bar{s2}$$



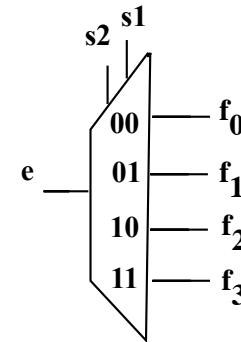
- Multiplexeur 2 vers 1 sur n bits



## Le démultiplexeur ou décodeur

- Décodeur 2 bits:

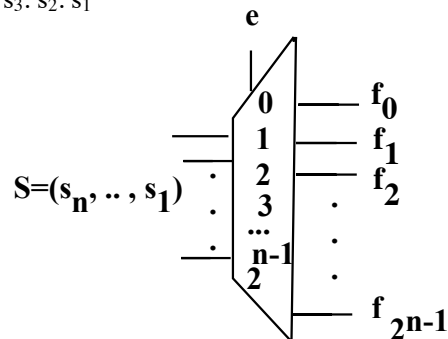
- 4 fonctions booléennes:
  - $f_0(e,s1,s2) = e \cdot \overline{s1} \cdot \overline{s2}$
  - $f_1(e,s1,s2) = e \cdot s1 \cdot \overline{s2}$
  - $f_2(e,s1,s2) = e \cdot \overline{s1} \cdot s2$
  - $f_3(e,s1,s2) = e \cdot s1 \cdot s2$



## Généralisation à n bits

- Décodeur n bits

- $2^n$  fonctions booléennes
- Exemple :
  - $f_3 = 1$  si ( $e=1$  et  $S=3$ ) avec S codé en base 2
  - Soit  $f_3 = e \cdot \overline{s_n} \cdot \overline{s_{n-1}} \cdot \dots \cdot \overline{s_3} \cdot s_2 \cdot s_1$

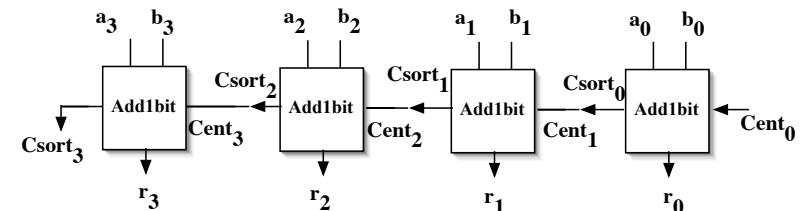


## Circuits réguliers

- Arithmétique

- Exemple : l'additionneur n bits

$$\begin{array}{r}
 \text{Cent}_1 \text{ Csort}_0 \\
 \begin{array}{cccc}
 \swarrow 0 & \swarrow 1 & \swarrow 0 & \\
 1 & 0 & 1 & 1 \\
 + & 0 & 1 & 0 \\
 \hline
 1 & 1 & 0 & 1
 \end{array}
 \end{array}
 \begin{array}{l}
 A = (a_3, a_2, a_1, a_0) \\
 B = (b_3, b_2, b_1, b_0) \\
 \text{RES} = (r_3, r_2, r_1, r_0)
 \end{array}$$



## Additionneur n bits

- On peut généraliser à n bits

