

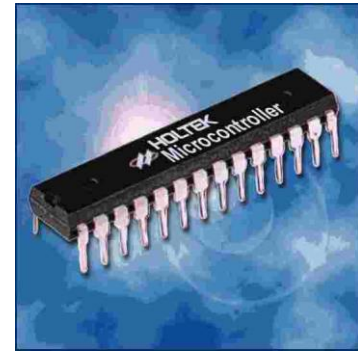
Réalisation de fonctions arithmétiques et logiques

Polytech Nice Sophia Antipolis

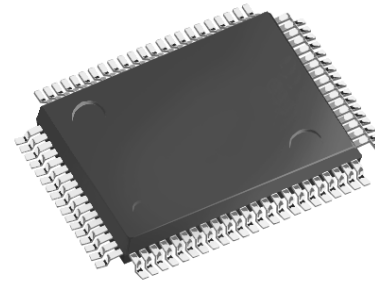
Circuits intégrés

Les types de boîtiers

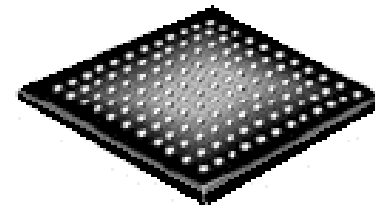
- DIP
- PGA
- Flatpack
- LCC
- QFP
- SOIC



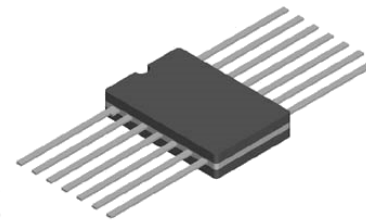
DIP



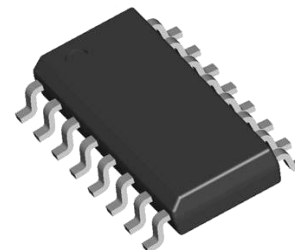
QFP



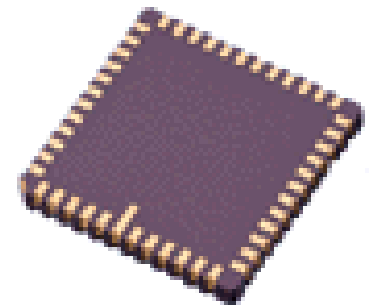
PGA



flatpack



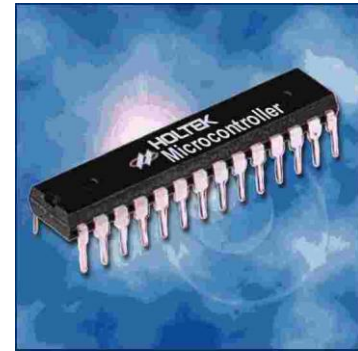
SOIC



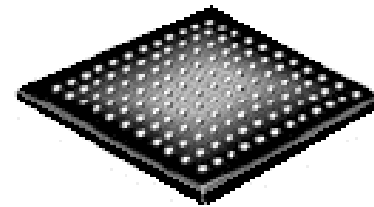
LCC

Les types de boîtiers

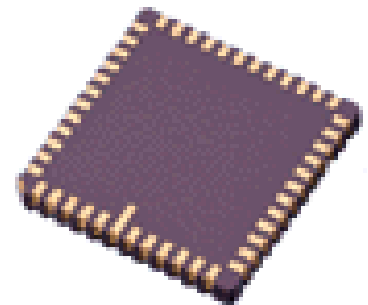
- DIP
- BGA
- Flatpack
- LCC
- QFP
- SOIC



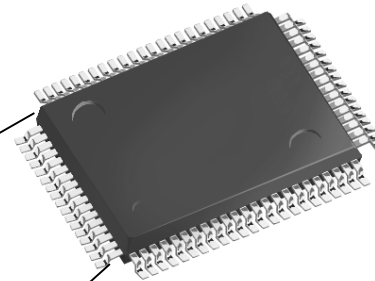
DIP



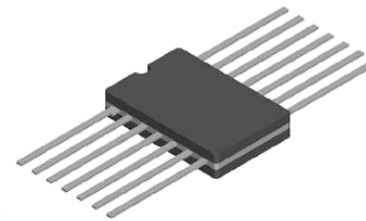
BGA



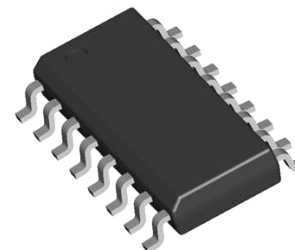
LCC



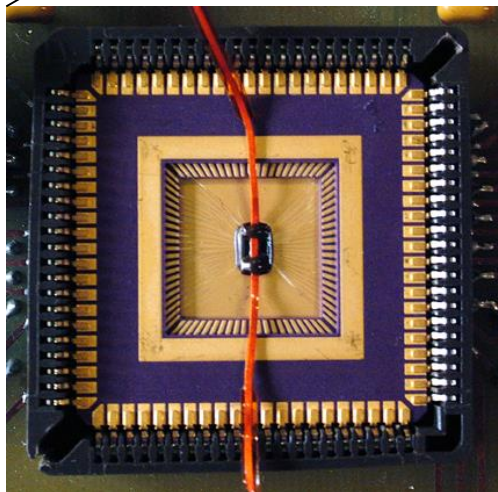
QFP



flatpack

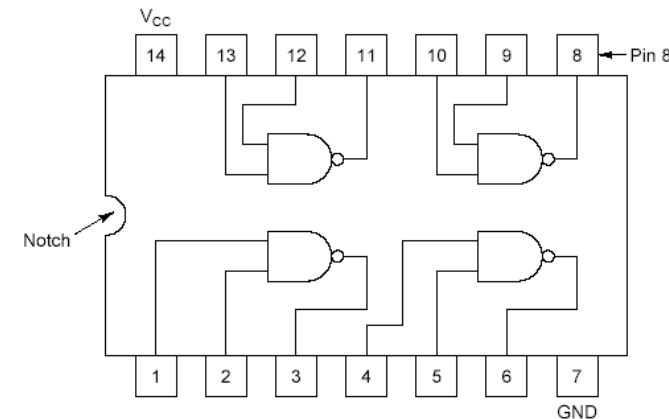


SOIC



Densité d'intégration

- SSI – Small Scale Integration (1960's)
 - 1 à 10 portes / circuit
 - Moins de 100 transistors
- MSI – Medium Scale Integration
 - 10 à 100 portes / circuit
 - Plus de 3000 transistors
- LSI – Large Scale Integration (1970's)
 - 100 à 10 000 portes / circuit
 - Plusieurs dizaines de milliers de transistors
- VLSI – Very Large Scale Integration (1980's)
 - + de 10 000 portes / circuit
 - 1 million de transistors
- ULSI – Ultra Large Scale Integration (2000's)
 - + de 100 000 porte / circuit



Les circuits logiques combinatoires

Le multiplexeur

- Un multiplexeur dispose de 2^n entrées, d'une sortie et de n lignes de sélection.
- Les sélecteurs permettent de choisir 1 entrée parmi les 2^n et de la router vers la sortie
- MUX_1_2ⁿ, 1 parmi 2ⁿ
- Comment faire ce circuit ?

Le multiplexeur

On commence par quoi ?

- Le circuit
- La table de vérité
- La forme algébrique
- La forme temporelle

Dans l'ordre

- 1.
- 2.
- 3.
- 4.

Le multiplexeur

On commence par quoi ?

- Le circuit
- La table de vérité
- La forme algébrique
- La forme temporelle

Dans l'ordre

1. La table de vérité
2. La forme algébrique
3. Le circuit
4. La forme temporelle

Le multiplexeur à deux entrées

MUX_2_1

- Choix d'une topologie
 - 0 -> e0
 - 1 -> e1

sel	e ₁	e ₀	s
0	0	0	
0	1	0	
0	0	1	
0	1	1	
1	0	0	
1	1	0	
1	0	1	
1	1	1	

Le multiplexeur à deux entrées

MUX_2_1

- Choix d'une topologie
 - 0 -> e0
 - 1 -> e1

$$s = \overline{sel}.e_0.\overline{e_1} + \overline{sel}.e_0.e_1 + sel.\overline{e_0}.\overline{e_1} + sel.e_0.e_1$$

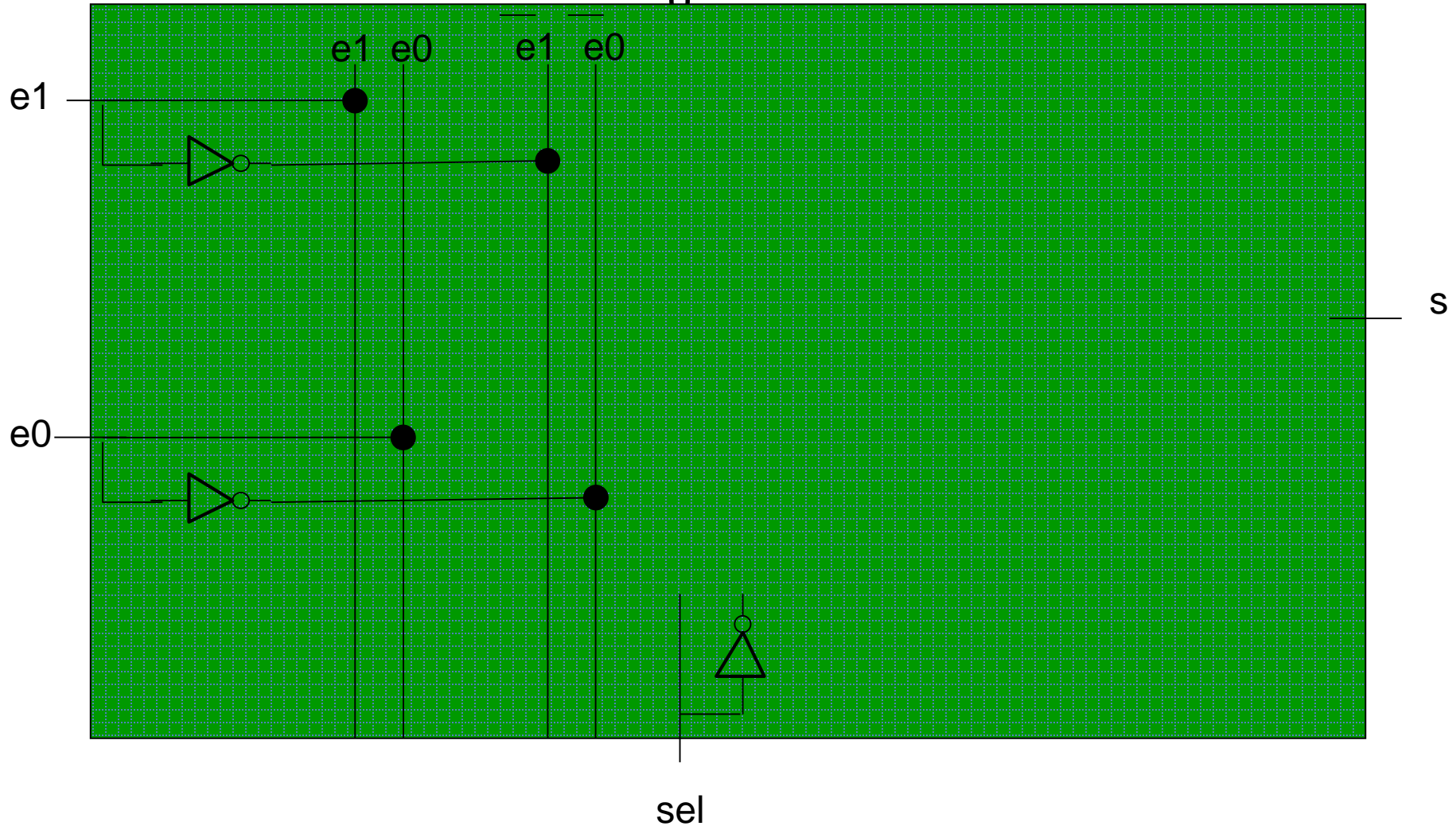
$$s = \overline{sel}.e_0(\overline{e_1} + e_1) + sel.e1(\overline{e_0} + e_0)$$

$$s = \overline{sel}.e_0 + sel.e1$$

sel	e ₁	e ₀	s
0	0	0	0
0	1	1	0
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	1
1	0	0	0
1	1	1	1

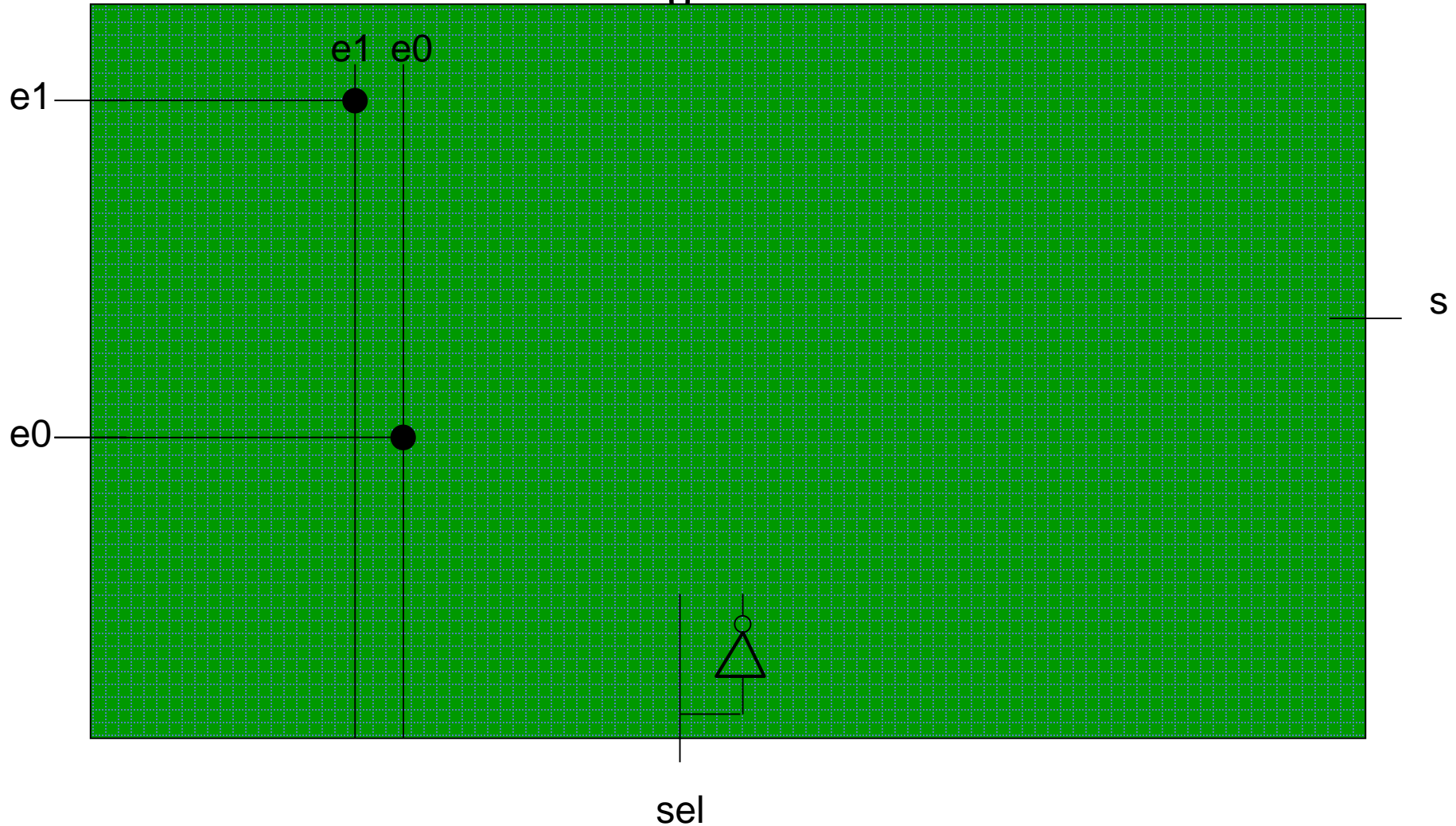
Implantation matérielle de

$$s = sel.e_0 + sel.e1$$

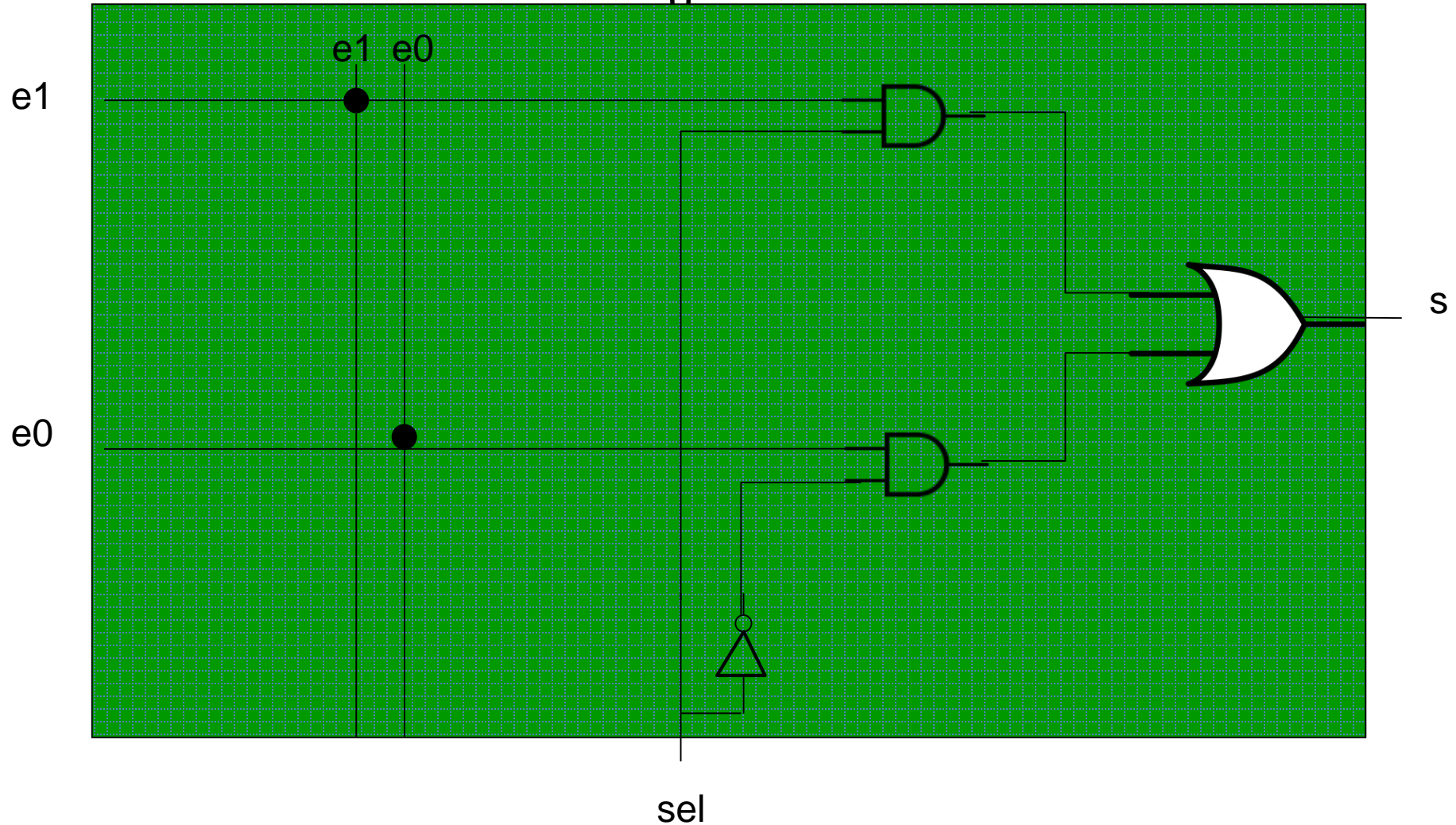


Implantation matérielle de

$$s = sel.e_0 + sel.e_1$$

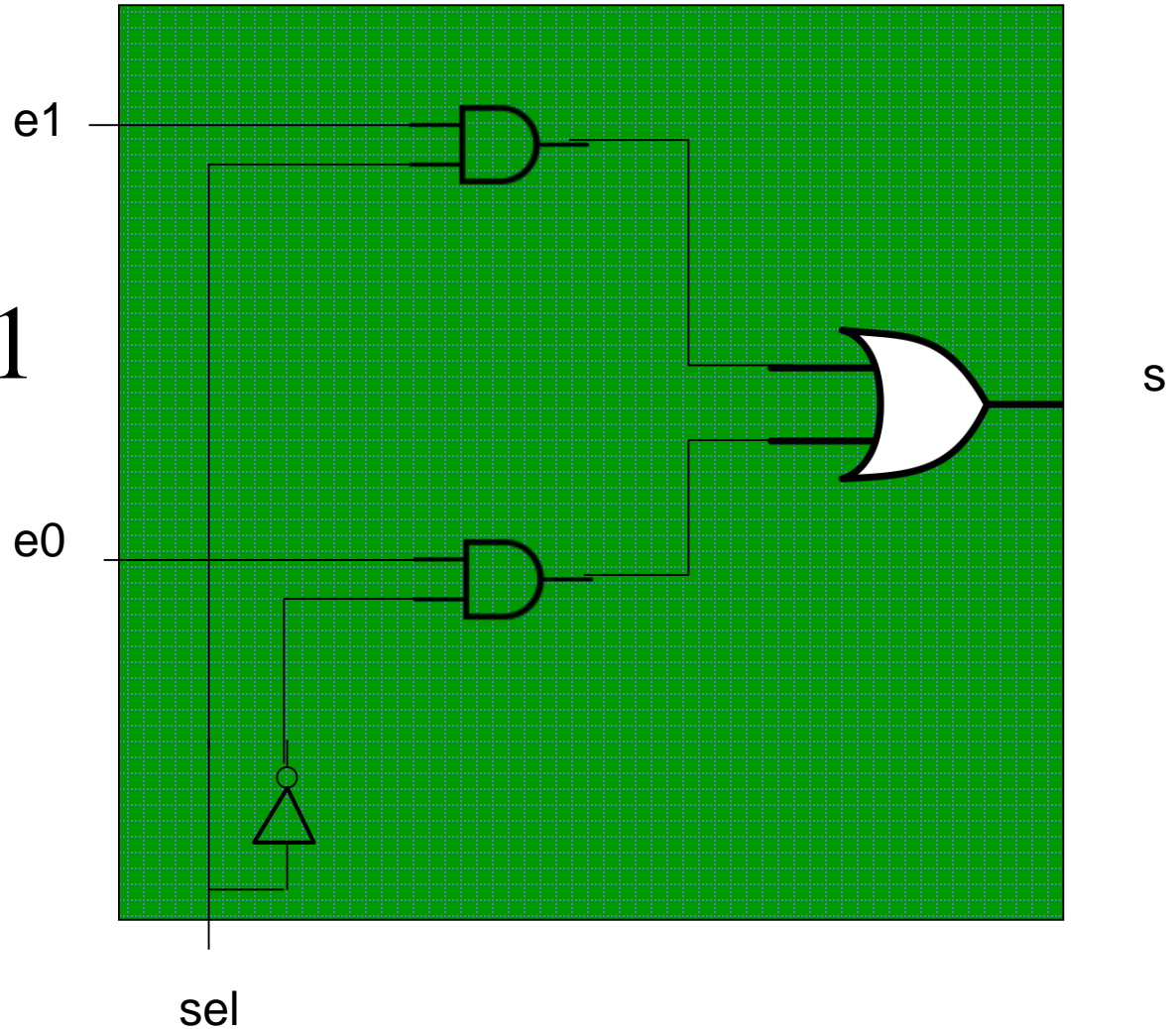


Implantation matérielle de

$$s = sel.e_0 + sel.e_1$$


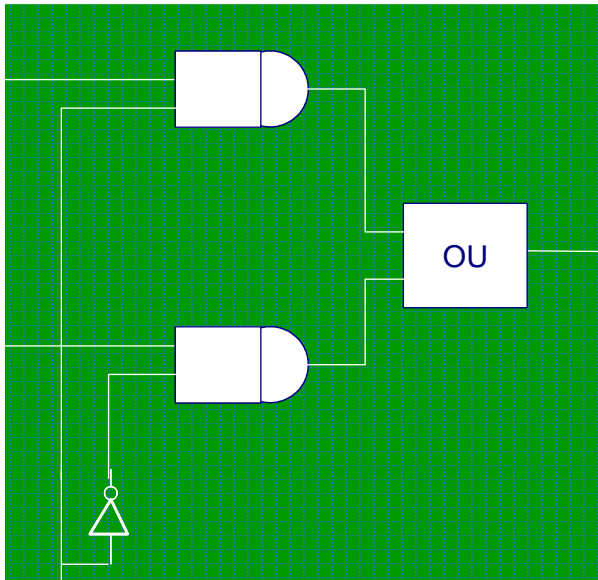
Implantation matérielle

$$s = \overline{sel}.e_0 + sel.e_1$$



Le multiplexeur à deux entrées

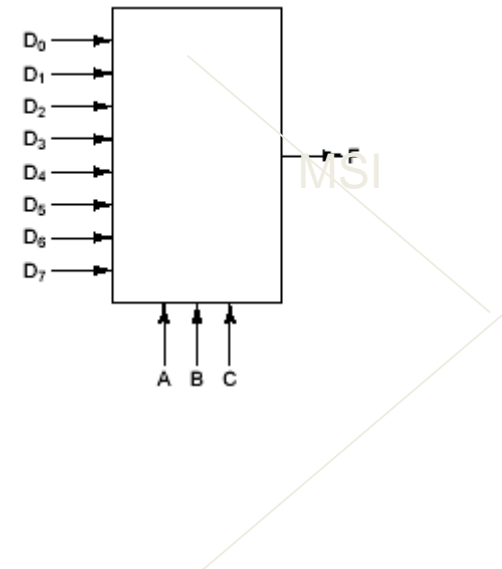
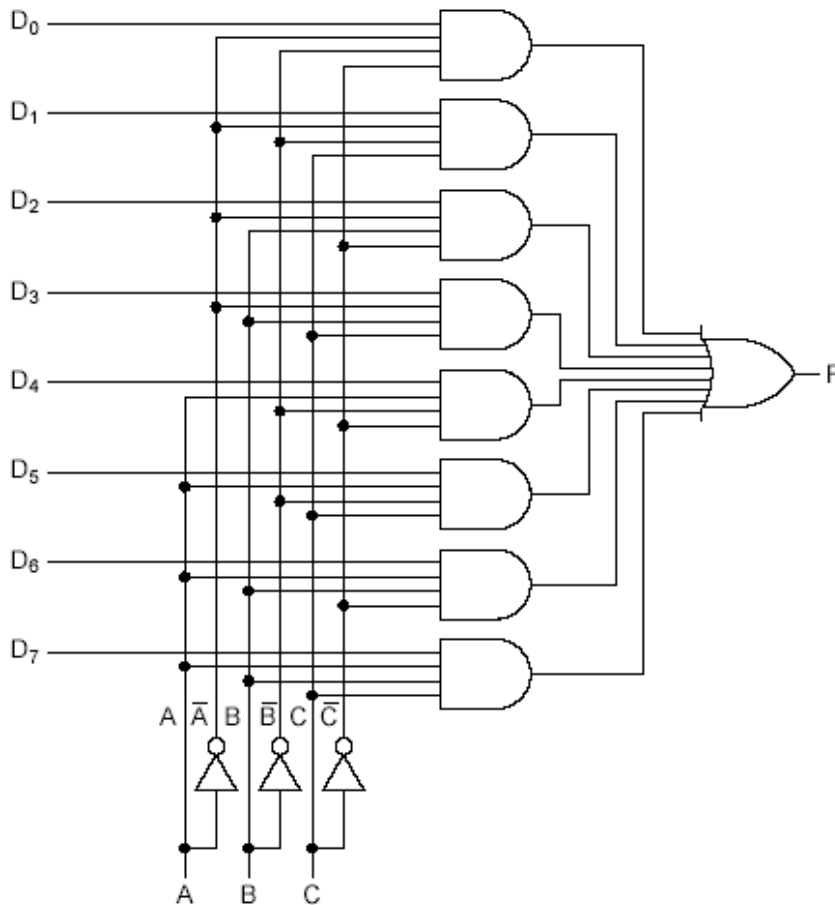
MUX_2_1



sel	e_0	e_1	s
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

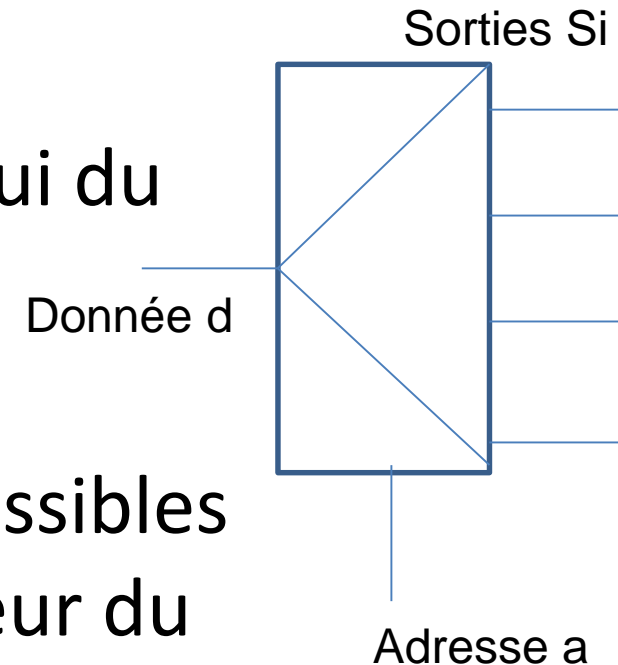
Le multiplexeur 8 vers 1

- 8 entrées = 2^3
- donc 3 sélecteurs



Le démultiplexeur

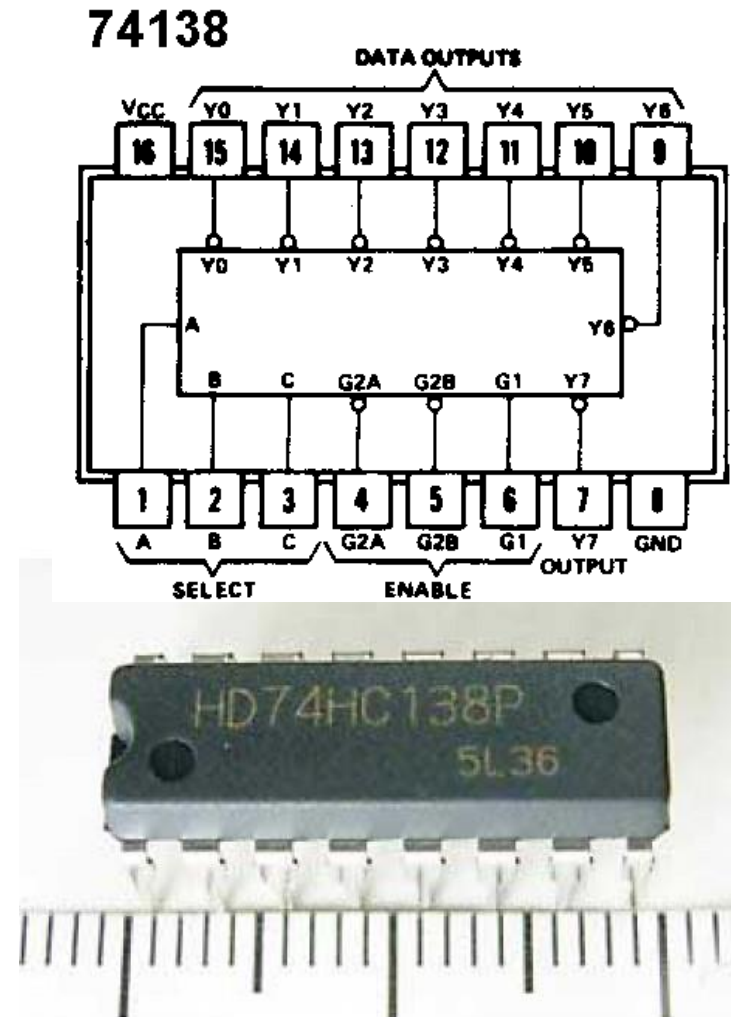
- Comme son nom l'indique, son comportement est inverse à celui du multiplexeur.
- Il autorise le routage d'un signal d'entrée vers une des sorties possibles du demux en fonction de la valeur du signal d'adresse, aussi appelé le sélecteur.



$$\begin{cases} S_0 = \bar{a}_1 \bar{a}_0 d \\ S_3 = a_1 a_0 d \end{cases}$$

7400 Integrated Circuits

- On retrouve par exemple ce type de fonctions logiques dans les circuits de série 7400 de Texas Instrument.
- Circuits TTL : Transistor-Transistor Logic
- Exemples :
 - 74138 = demux 1:8
 - 74151A = mux 8:1



Notre bibliothèque de portes

MSI

■ MAJ_

■ MUX_

■ DEMUX_

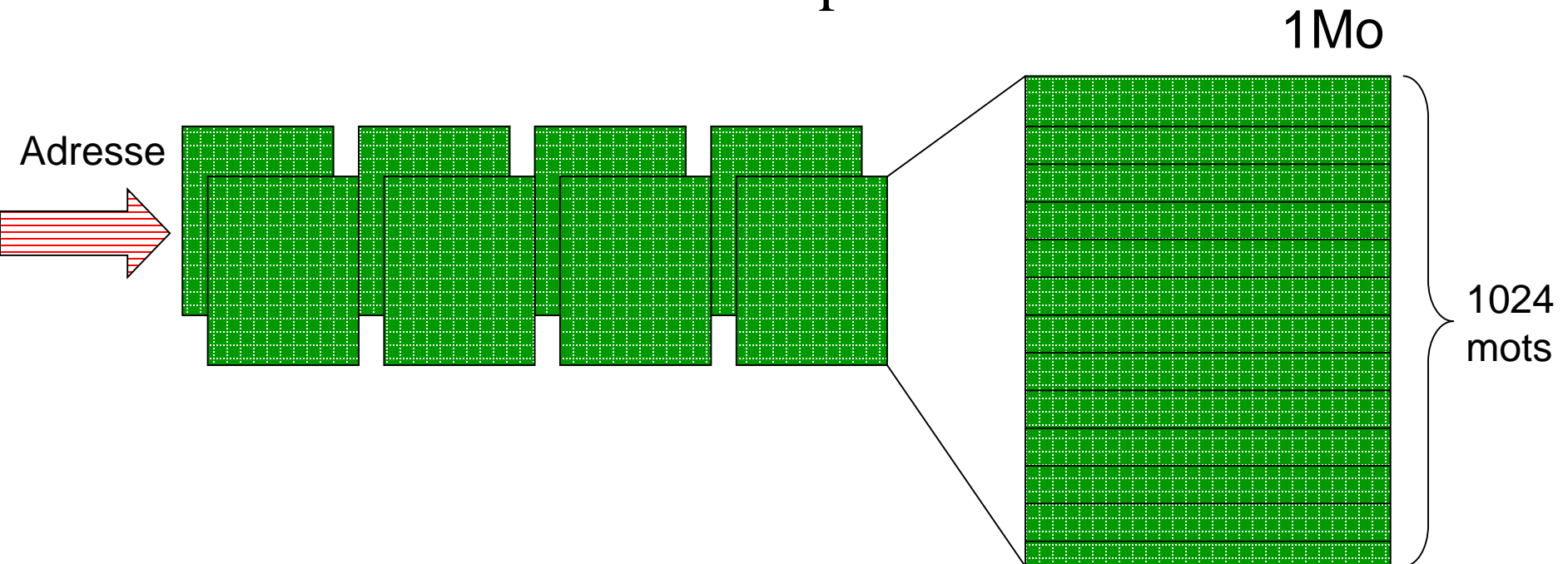
Le décodeur

- Il traduit l'information binaire en entrée pour rendre active la sortie dont le numéro correspond
- Par exemple un circuit mémoire utilise un décodeur pour aller chercher l'information présente à l'adresse n parmi ses 2^n emplacements

Le décodeur

Mémoire de 8 Moctets

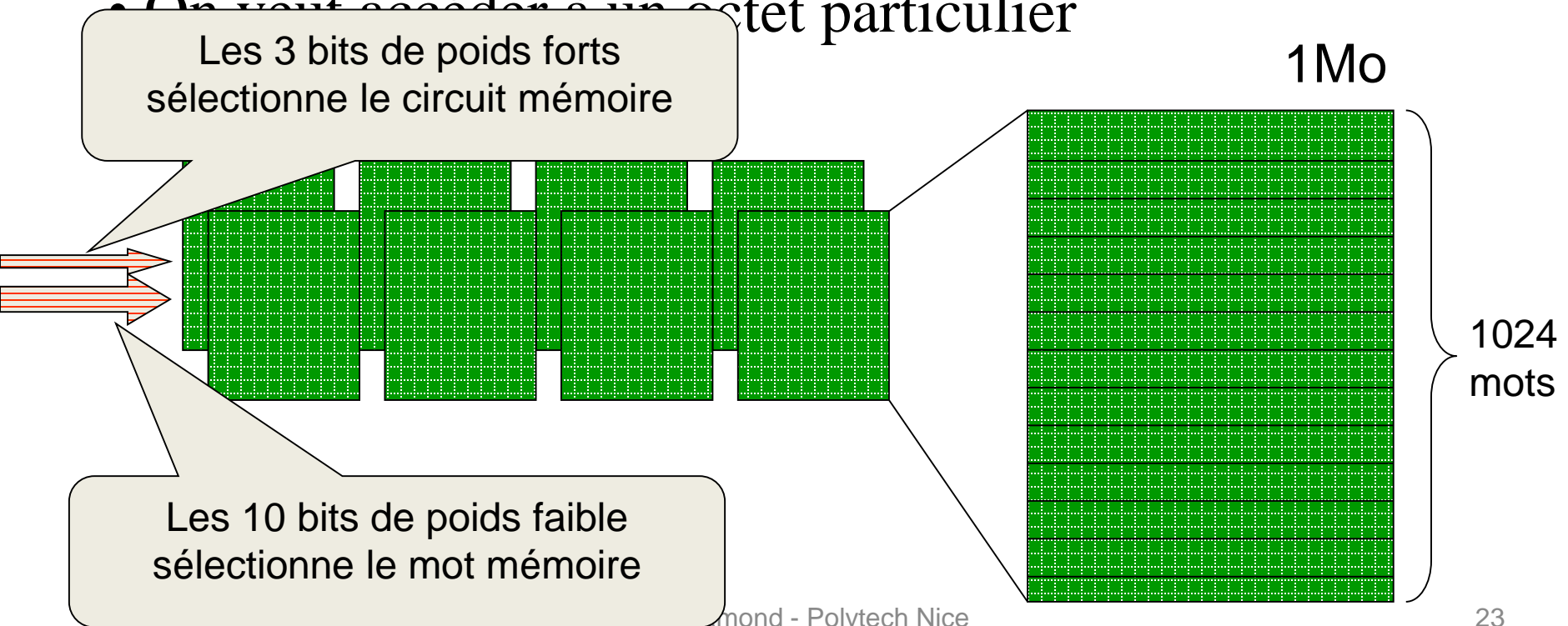
- composée de 8 circuits mémoires
- Chaque circuit contient 1 Moctet
- On veut accéder à un octet particulier



Le décodeur

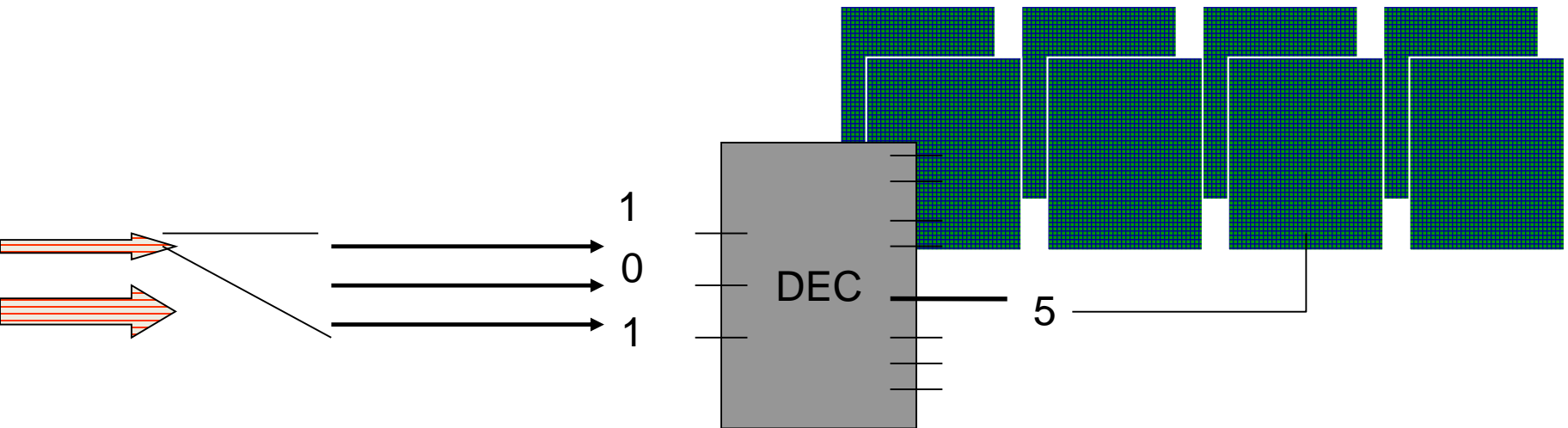
Mémoire de 8 Moctets

- composée de 8 circuits mémoires
- Chaque circuit contient 1 Moctet
- On veut accéder à un octet particulier



Le décodeur

- *Entrée = code 101*
- *Sortie = activer la sortie 5*
Mettre les autres sorties à 0



Le décodeur

e_2	e_1	e_0	s_0	...	s_7
0	0	0	1		0
1	0	0	0		0
0	1	0	0		0
1	1	0	0		0
0	0	1	0		0
1	0	1	0		0
0	1	1	0		0
1	1	1	0		1

$$s_0 = \overline{e_2} \overline{e_1} \overline{e_0}$$

$$s_1 = \overline{e_2} \overline{e_1} e_0$$

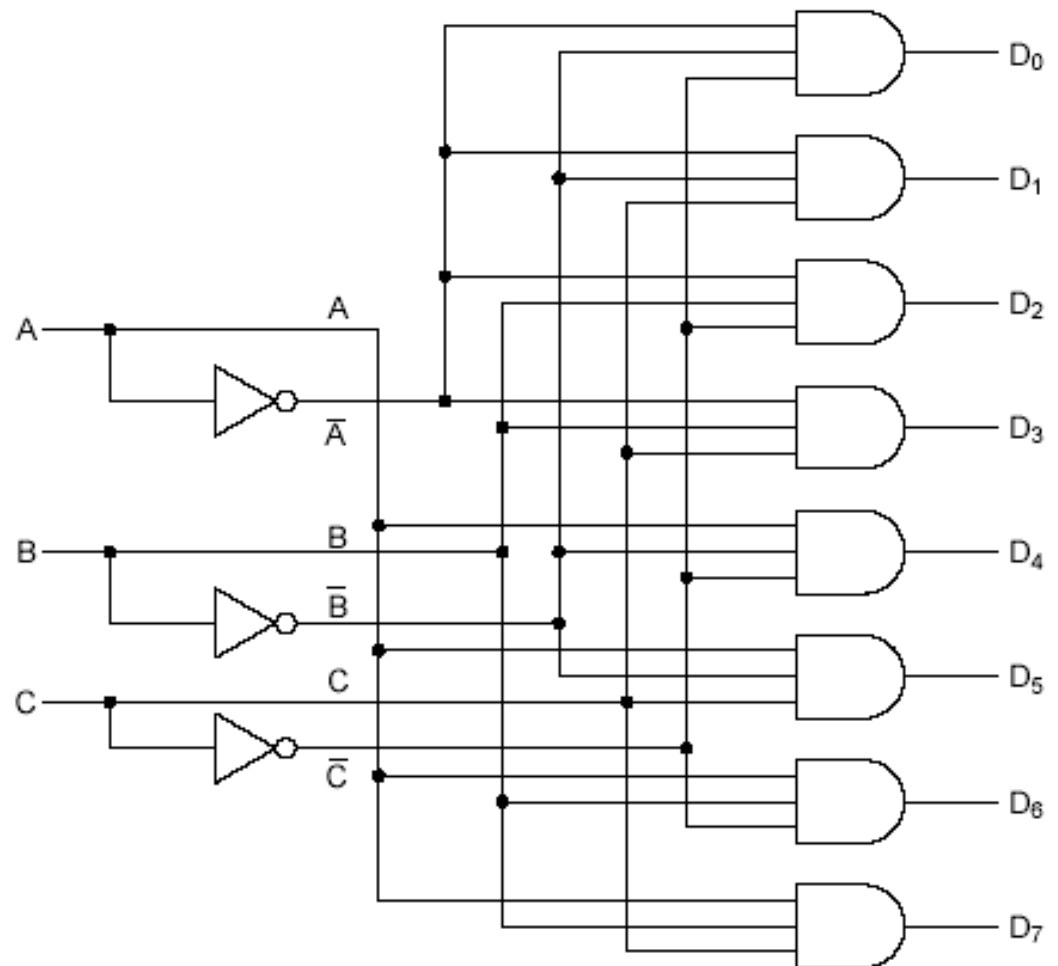
$$s_2 = \overline{e_2} e_1 \overline{e_0}$$

...

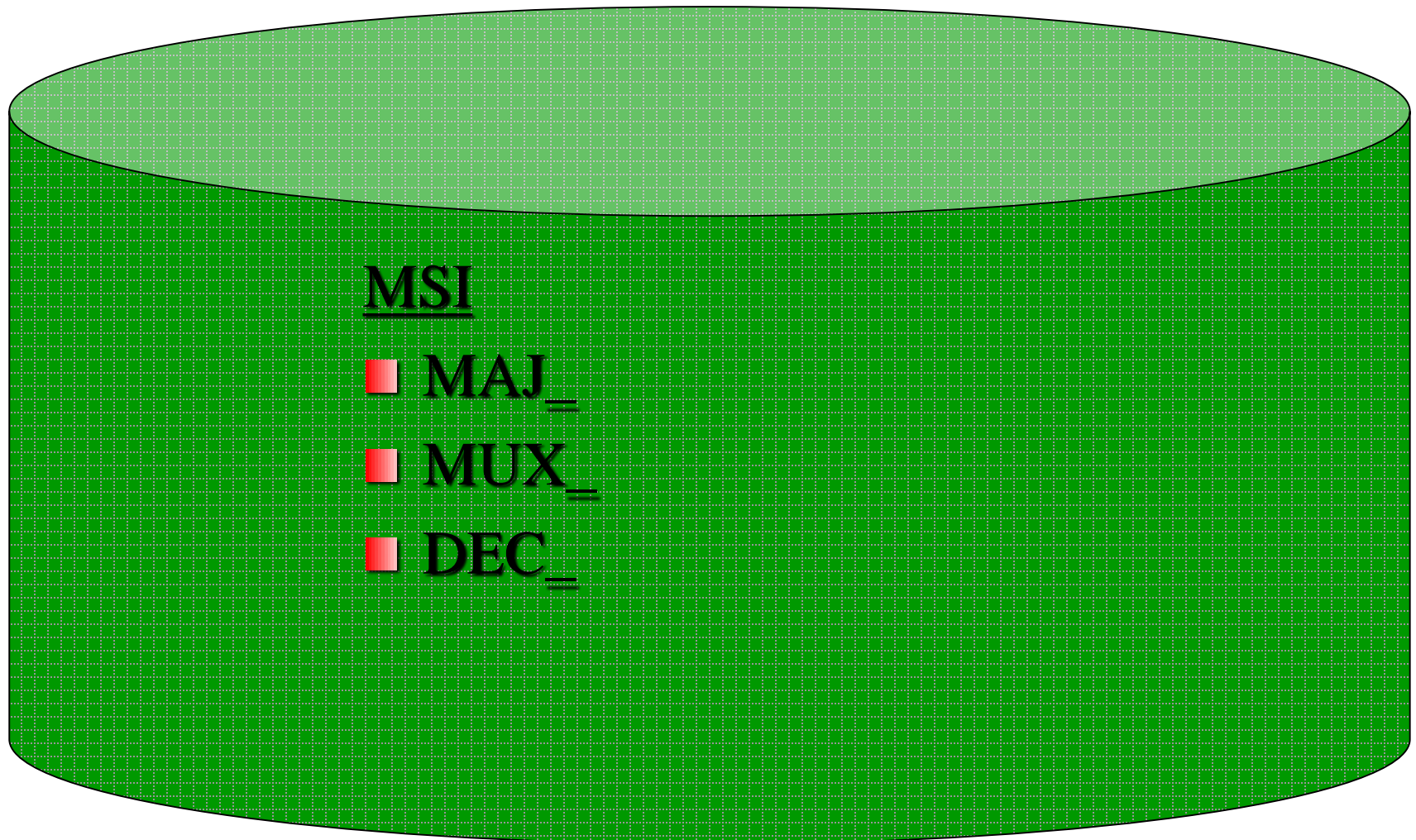
$$s_7 = e_2 e_1 e_0$$

Le décodeur

...

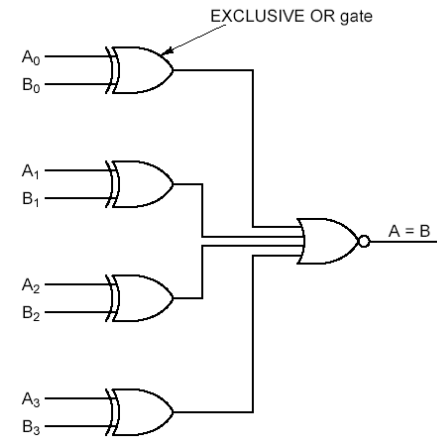


Notre bibliothèque de portes



Le comparateur

- Il effectue la comparaison de 2 mots de n bits A et B
- Renvoi
 - 1 si $A = B$,
 - 0 sinon



$$s = a_0 \oplus b_0 + a_1 \oplus b_1 + a_2 \oplus b_2 + a_3 \oplus b_3$$

Notre bibliothèque de portes

MSI

■ MAJ_

■ MUX_

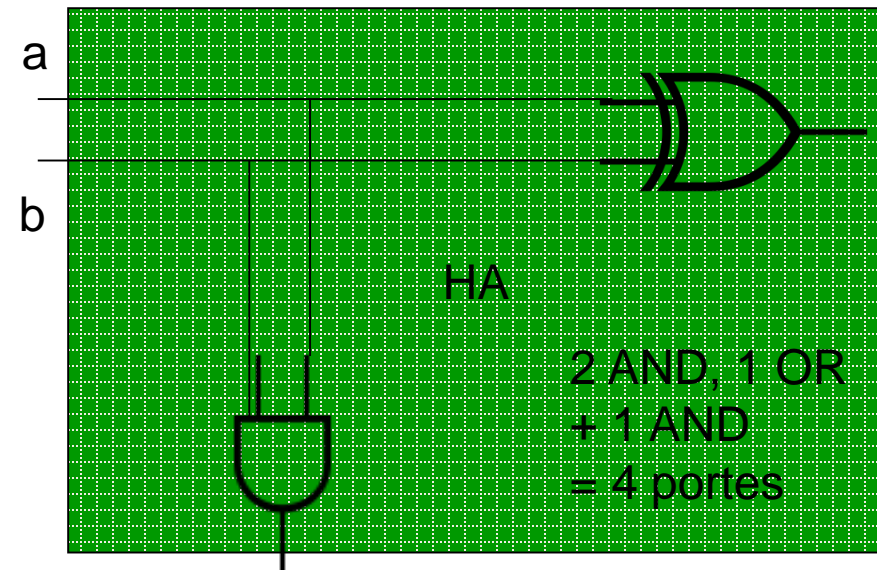
■ DEC_

■ COMP_

Les circuits arithmétiques

Addition & demi-additionneur (Half-Adder)

- sans retenue entrante
- $s(\text{omme}) = a \oplus b$
- $r(\text{etenue}) = a.b$



a	b	s	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Additionneur complet

$$s = \overline{a}\overline{b}R_e + \overline{a}bR_e + a\overline{b}\overline{R_e} + abR_e$$

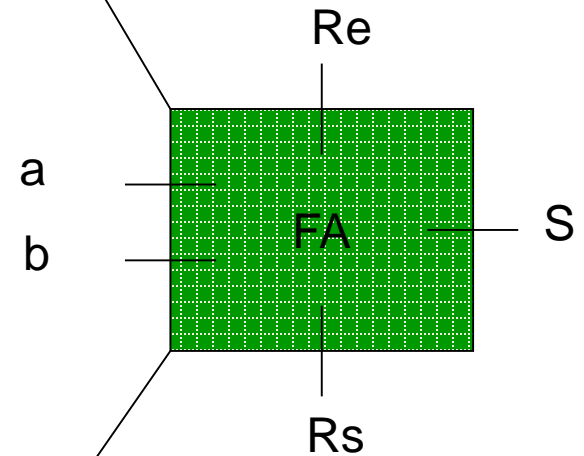
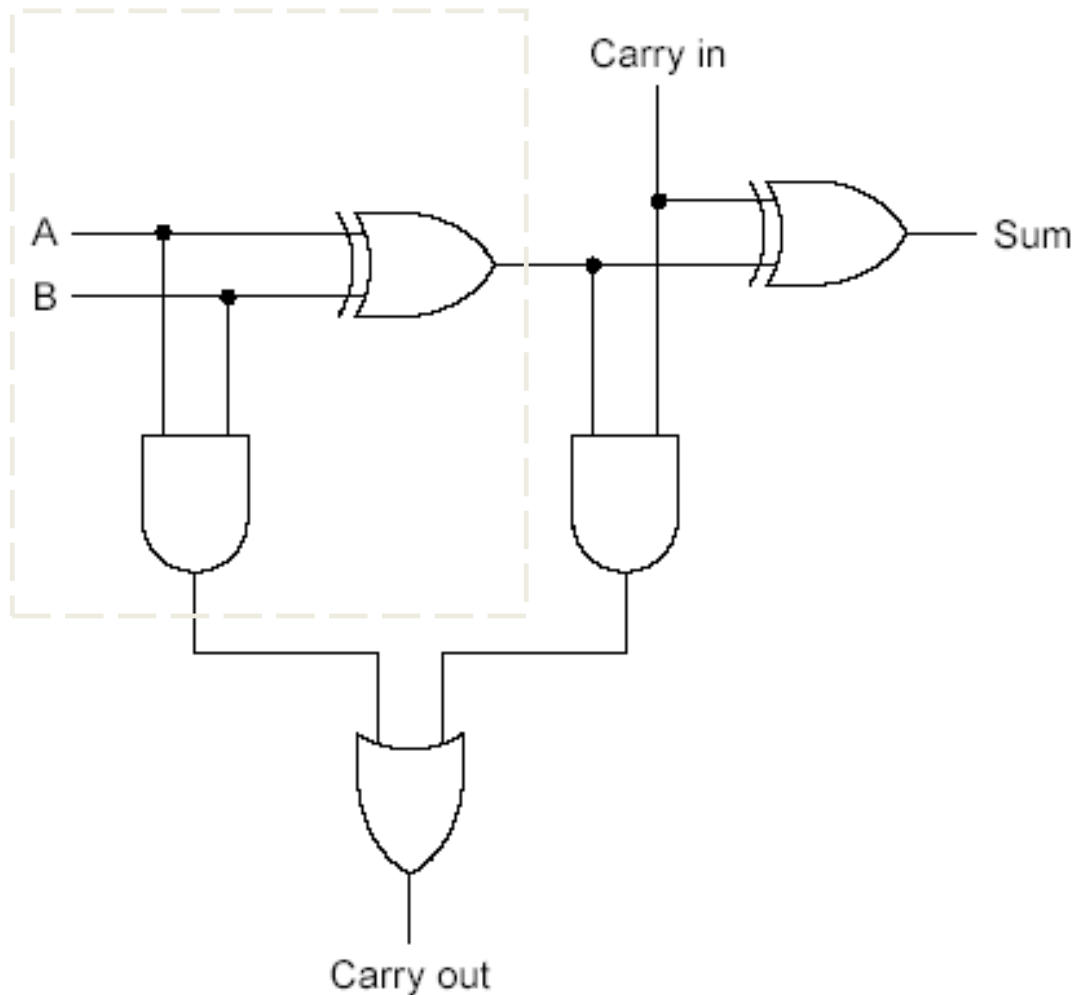
$$s = a \oplus b \oplus R_e$$

$$R_s = \overline{a}bR_e + a\overline{b}R_e + ab\overline{R_e} + abR_e$$

$$R_s = R_e(a \oplus b) + ab$$

a	b	Re	S	Rs
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

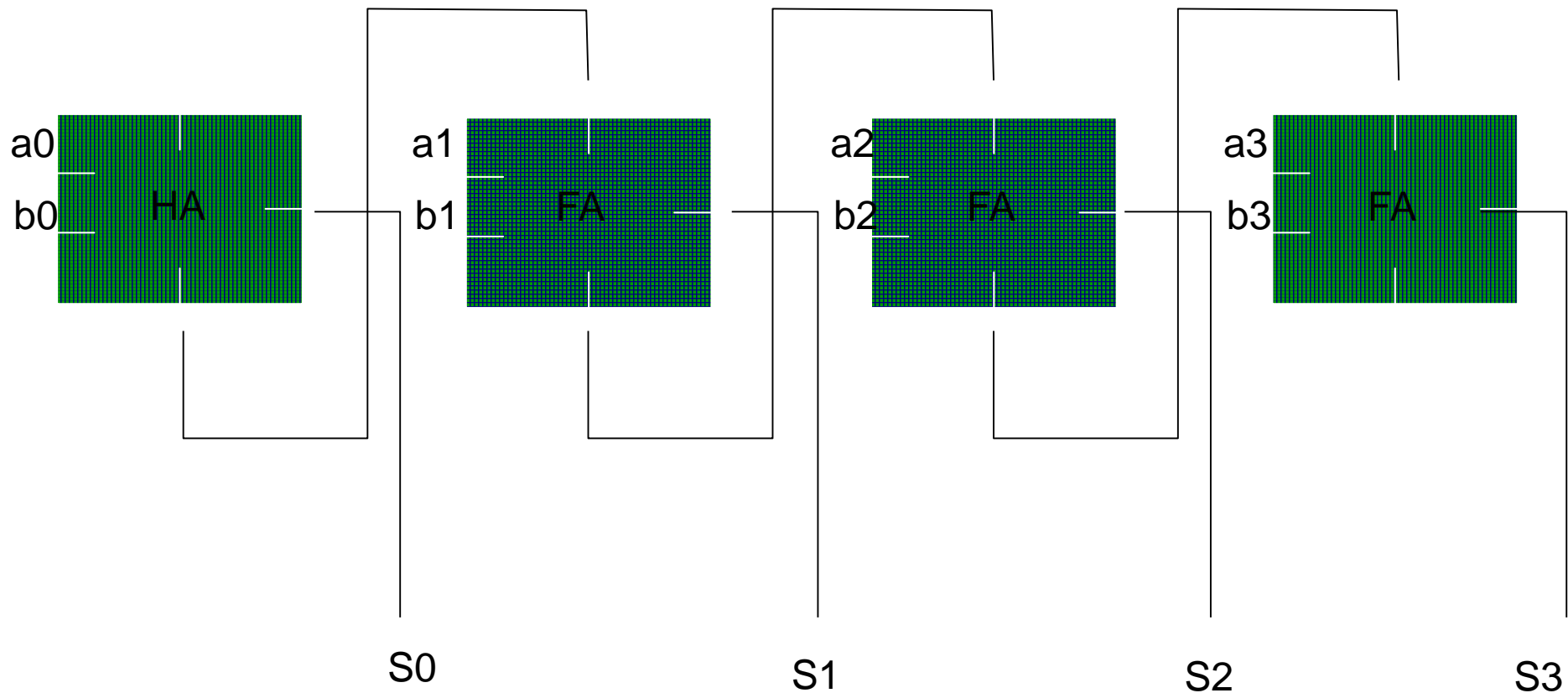
Full Adder (1 bit)



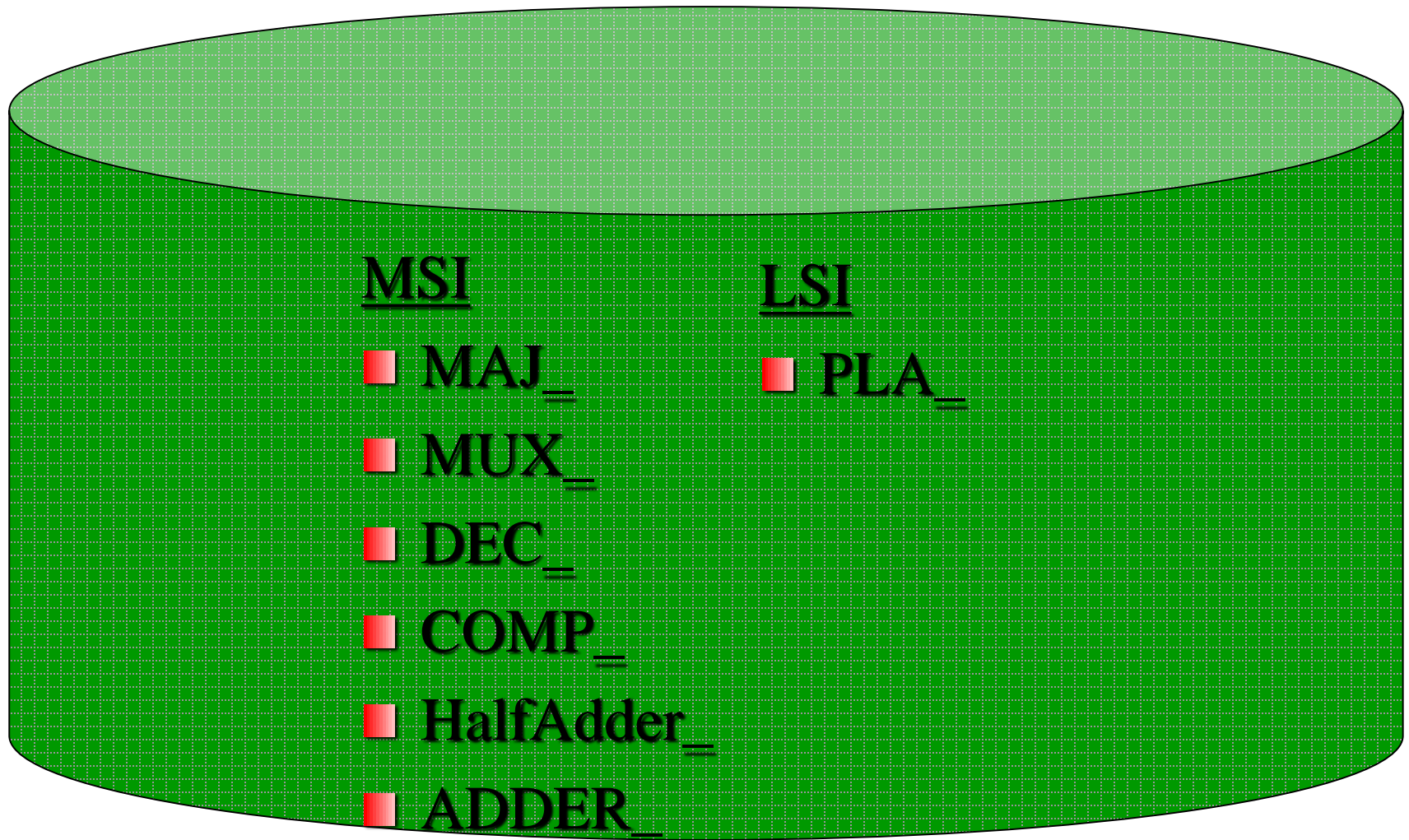
Full Adder 4 bits

additionneur à propagation de retenue

Temps de propagation très long

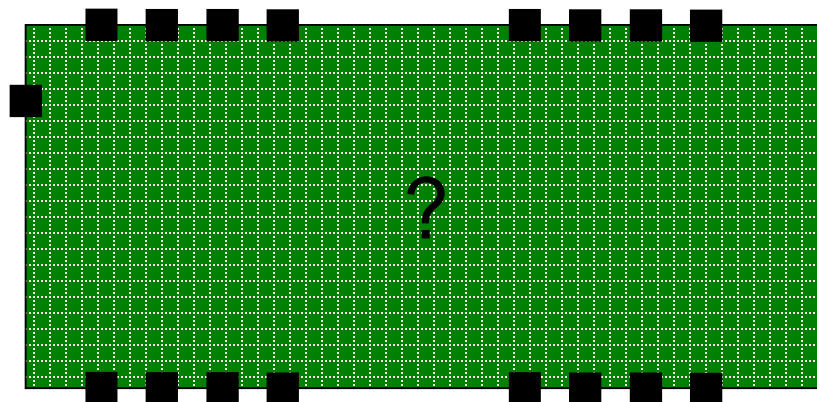


Notre bibliothèque de portes



Décaleur

- Le circuits dispose d'une entrée de donnée sur 8 bits et d'une ligne de commande ' c ' sur 1 bit
- Le circuit fournit en sortie, l'entrée décalée d'une place à gauche (\bar{c}) ou à droite en fonction de (c)



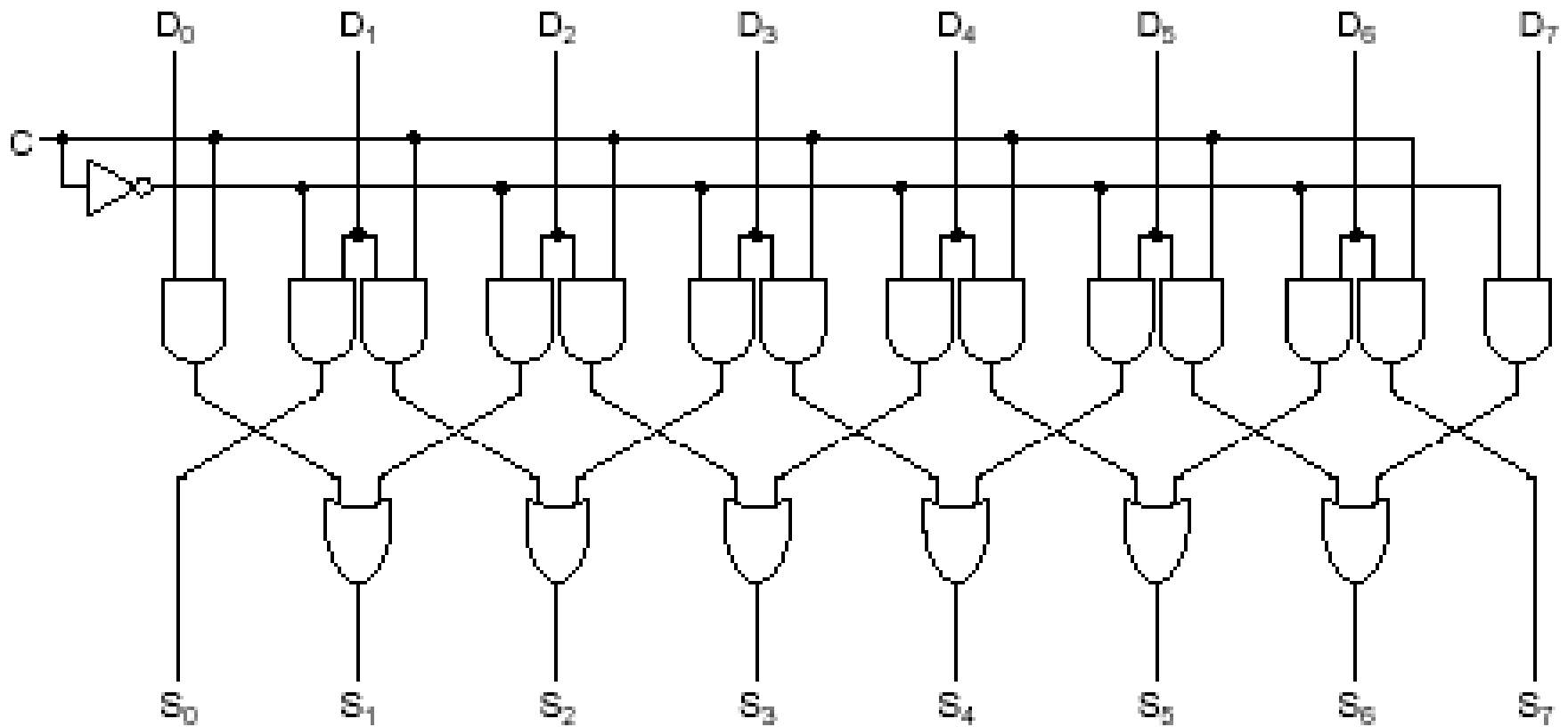
Décaleur

- Donner l'équation de chaque bit de sortie du décaleur 8 bits
- Dessiner ensuite le schéma en portes du décaleur d'une position à gauche ou à droite

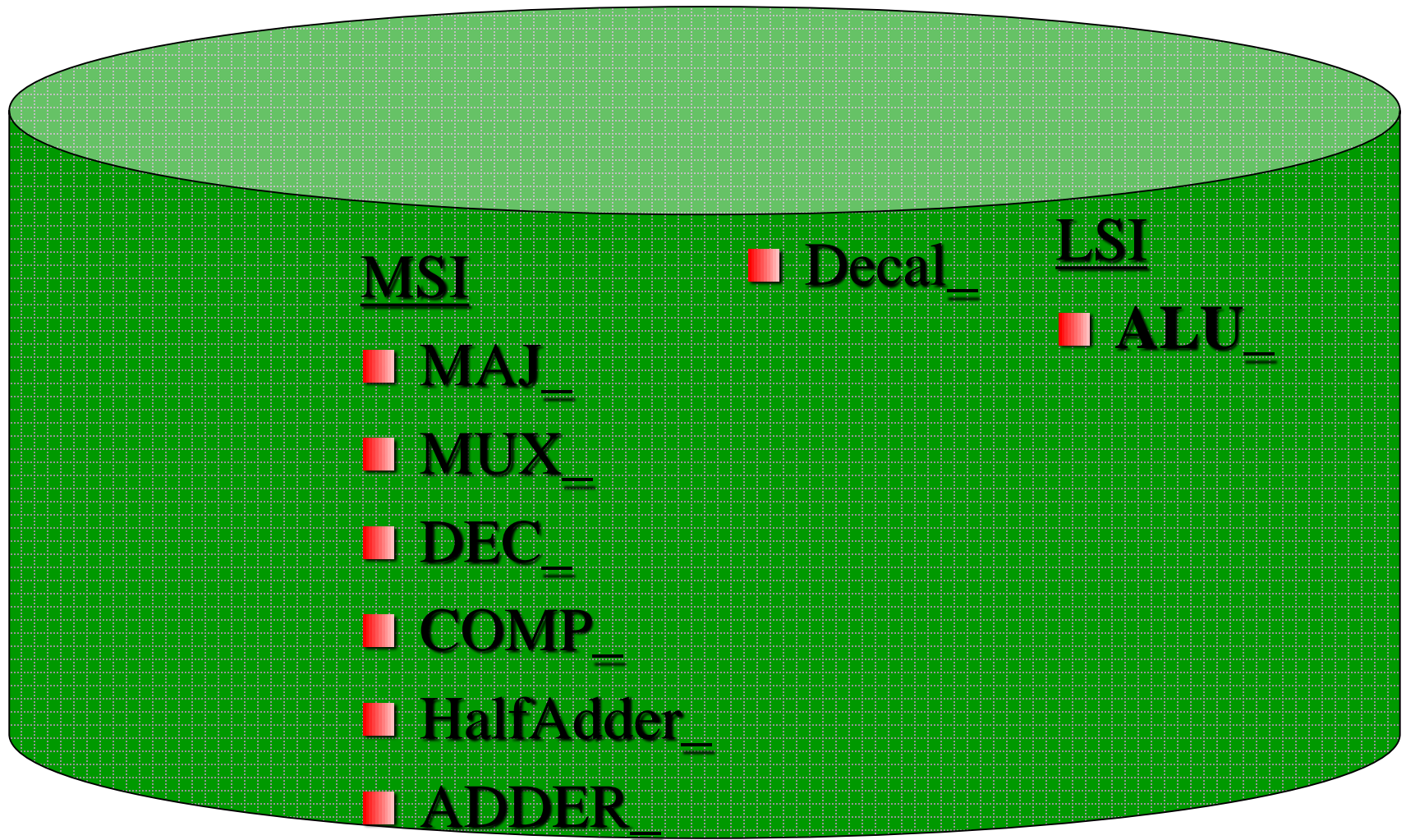
Décaleur

- Donner l'équation de chaque bit de sortie du décaleur 8 bits
- $$\begin{cases} S_0 = e_1 \cdot c \\ S_7 = e_6 \cdot \bar{c} \\ S_i = e_{i+1} \cdot c + e_{i-1} \cdot \bar{c}, i \in [1,6] \end{cases}$$
- Dessiner ensuite le schéma en portes du décaleur d'une position à gauche ou à droite

Décaleur



Notre bibliothèque de portes



Unité de calcul du processeur

- Un processeur doit pouvoir réaliser toute fonction logique et arithmétique
- L'unité matérielle responsable de ces calculs est appelée Unité Arithmétique et Logique
 - UAL
 - ou ALU
- Elle dispose de deux entrées dont la taille dépend de l'architecture du processeur
- Elle fournit en sortie le résultat de l'opération choisie par la troisième entrée

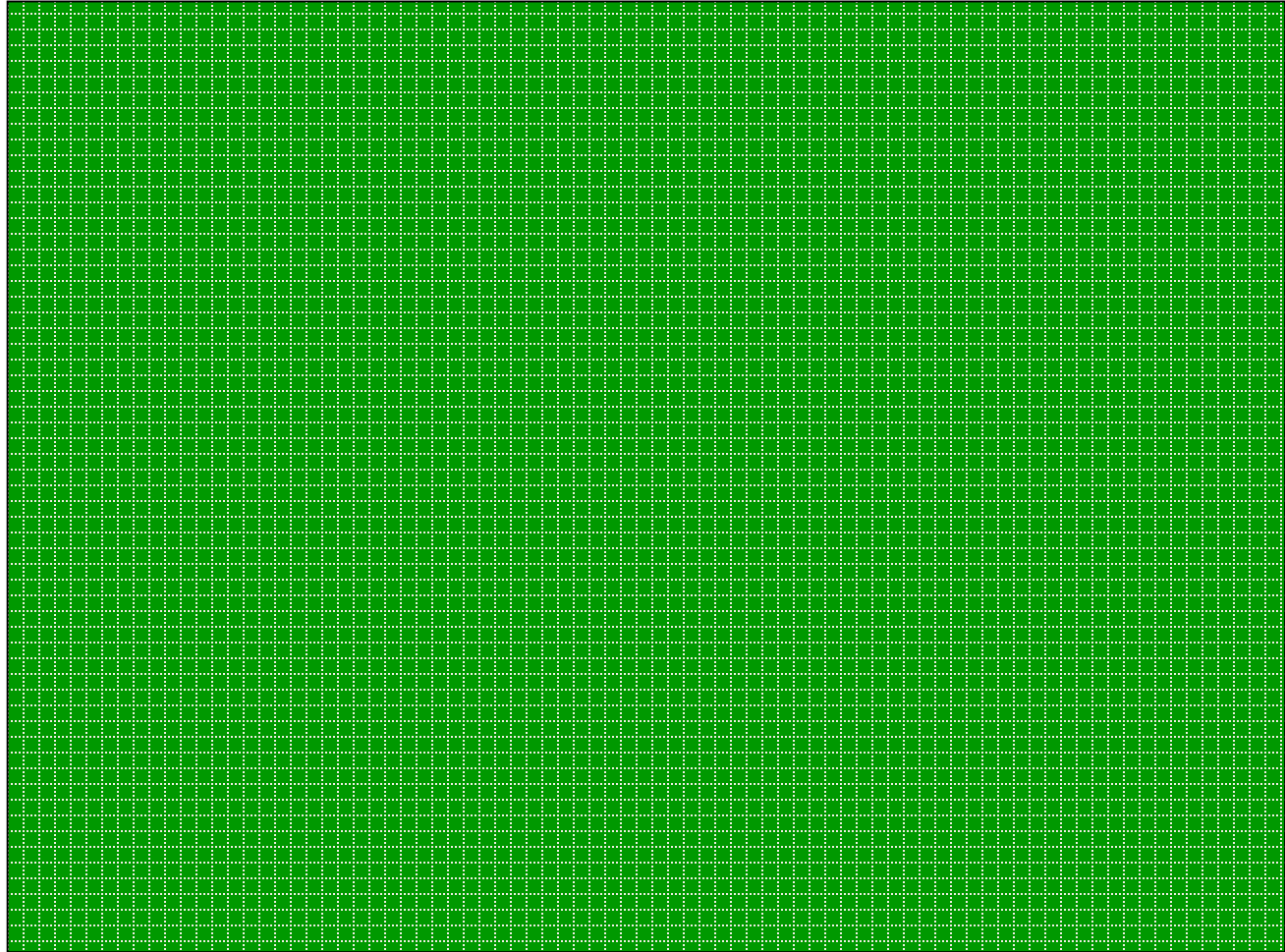
Réalisation structurelle d'une ALU

Cahier des charges :

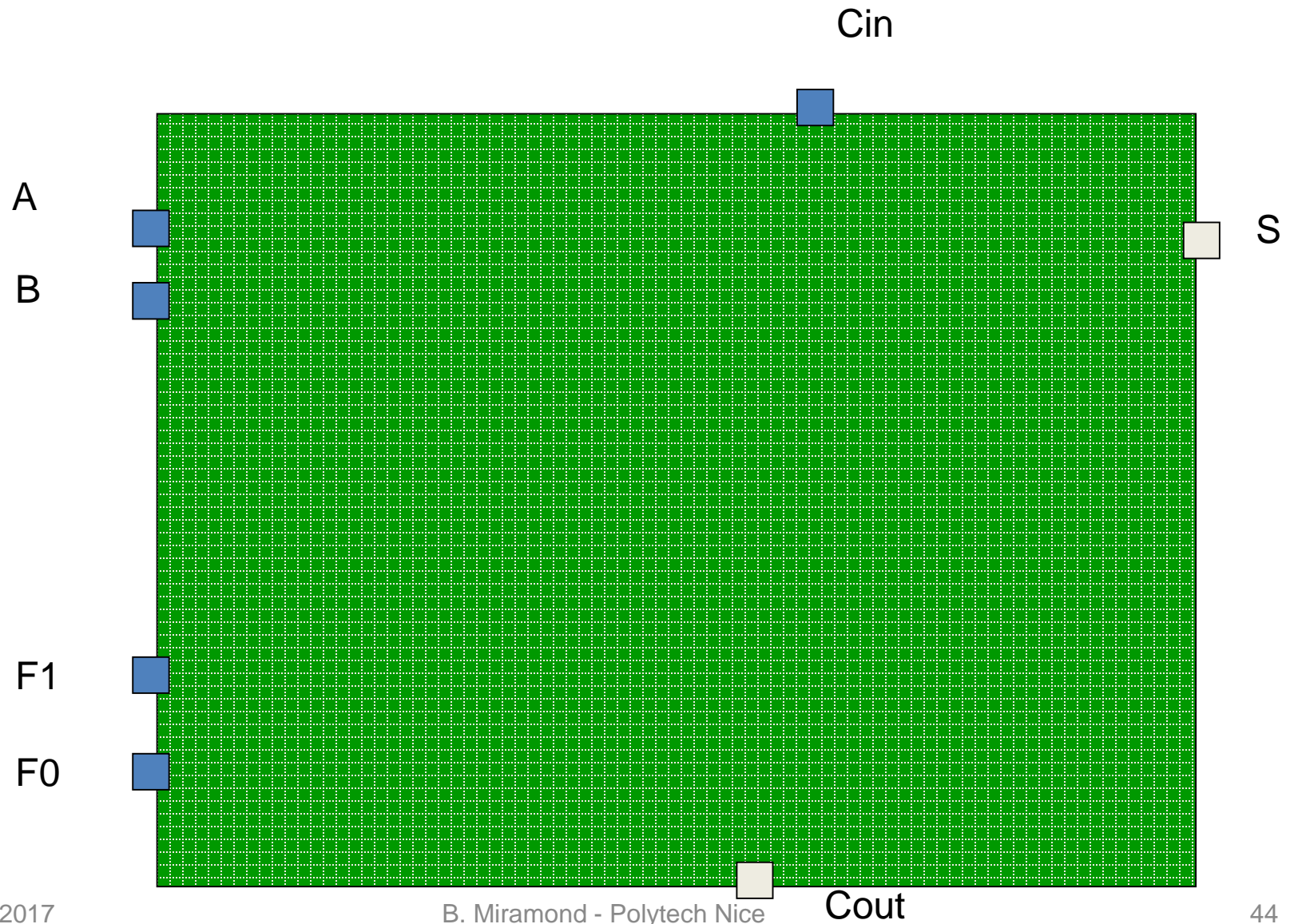
- Opérandes d'1 bit (A et B)
- Pouvant réaliser au choix les opérations
 - A or B
 - A and B
 - not B
 - A + B

Quels besoins ?

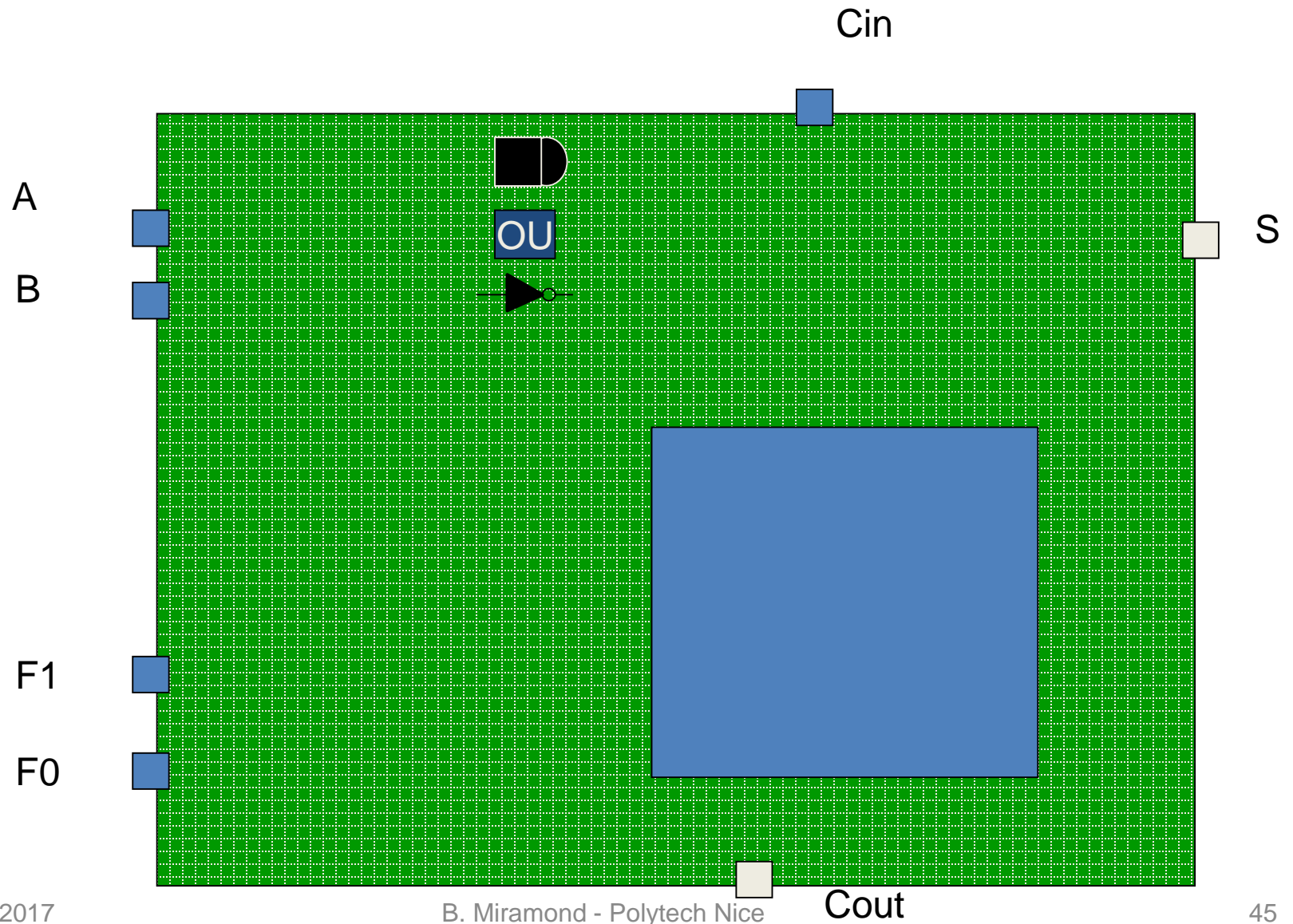
Interface ?



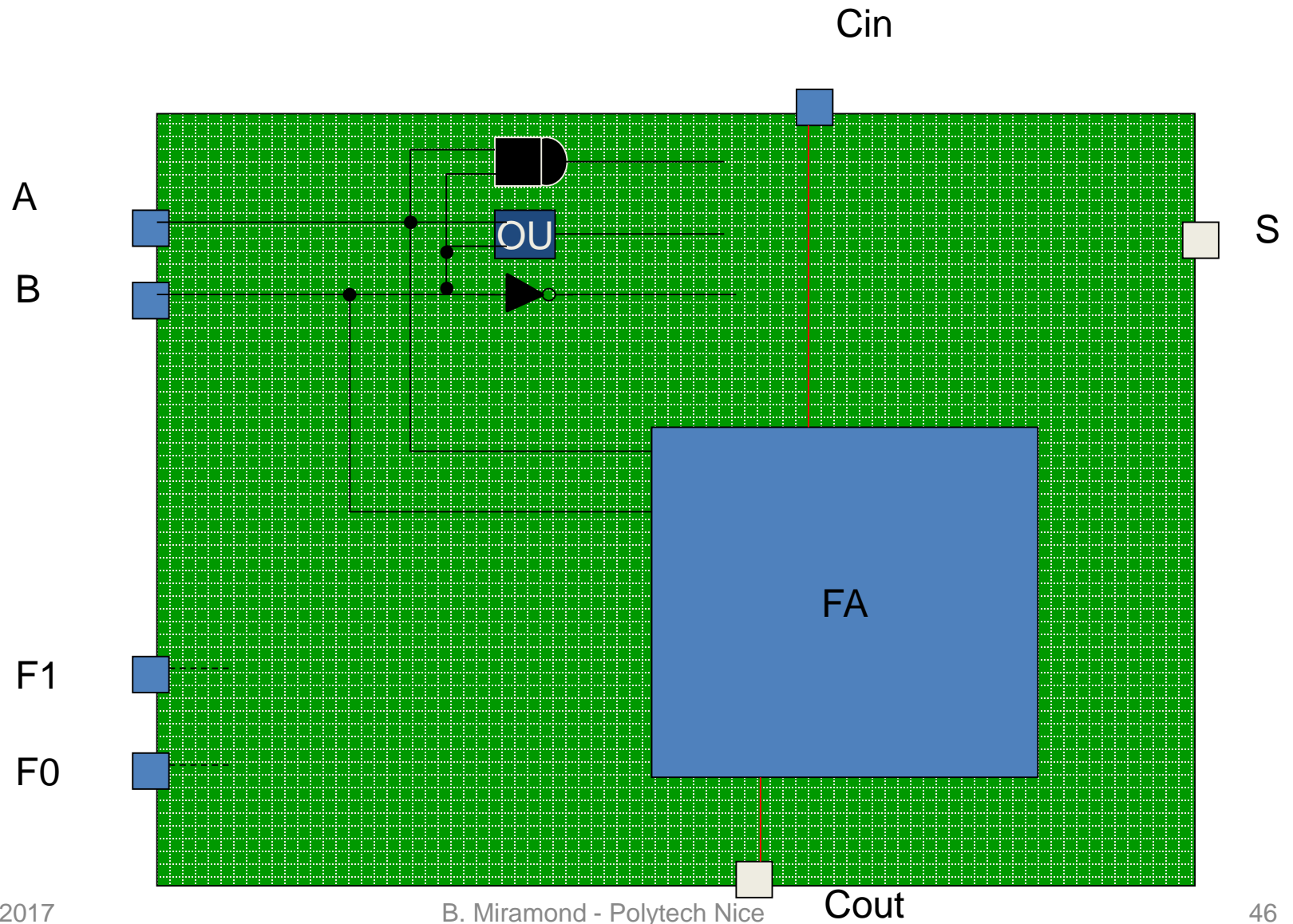
Interface ?



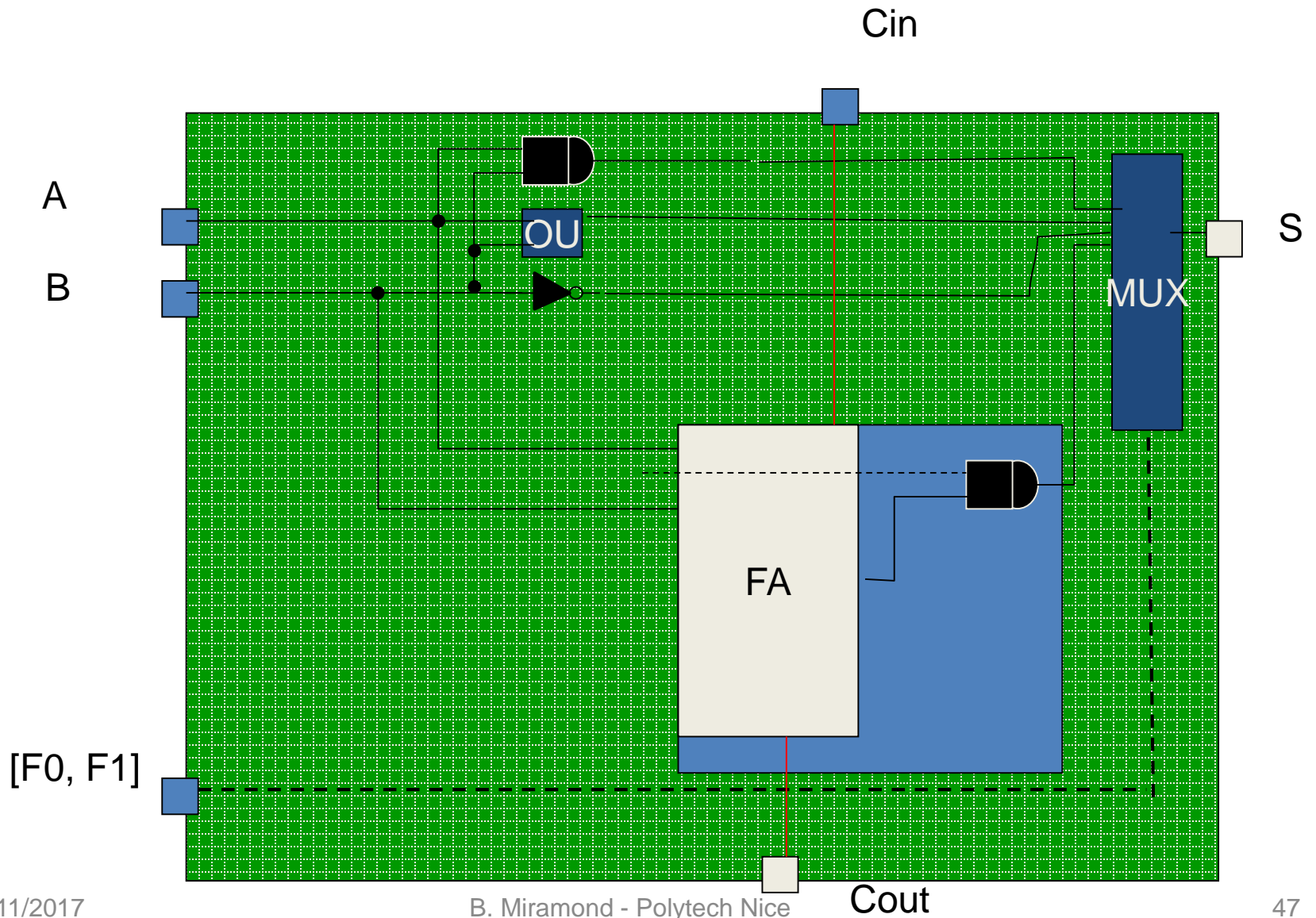
Vue structurelle



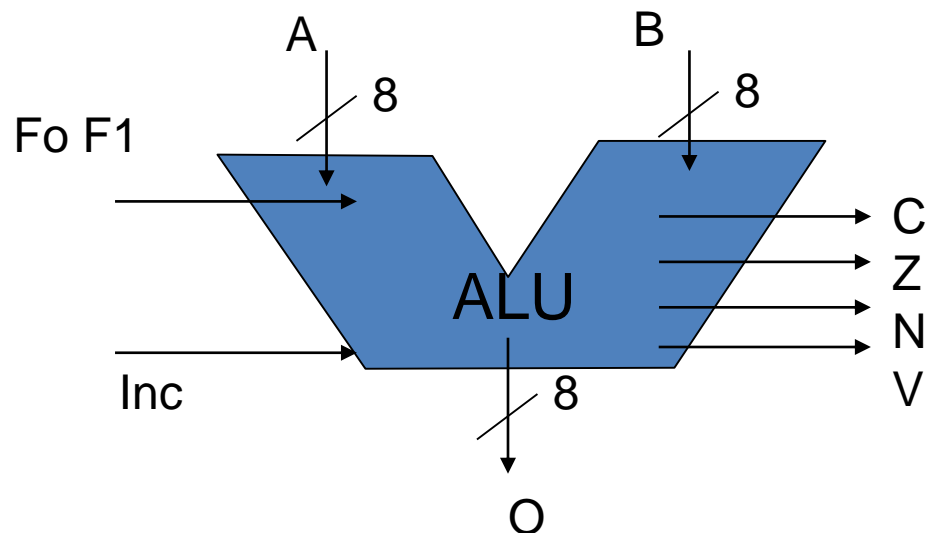
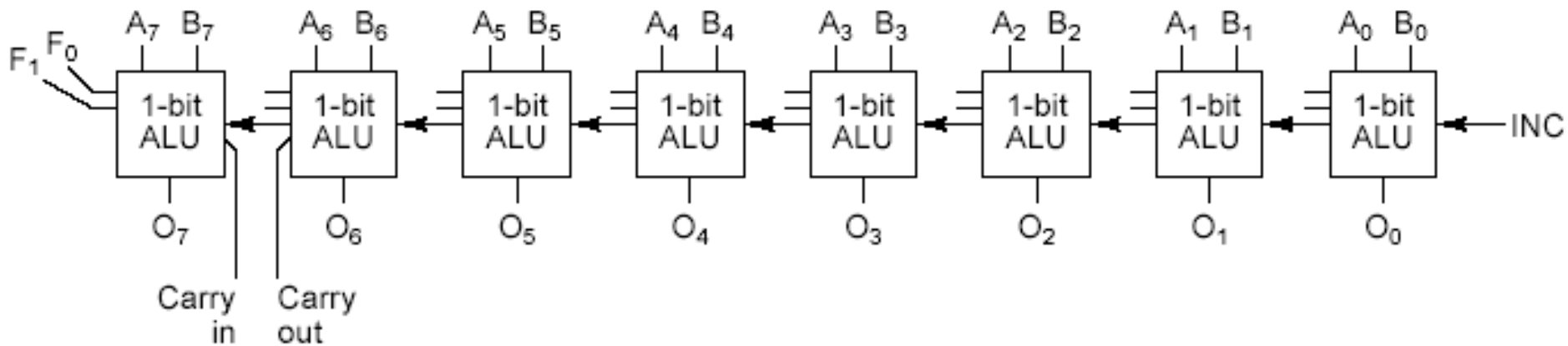
Vue structurelle



Vue structurelle



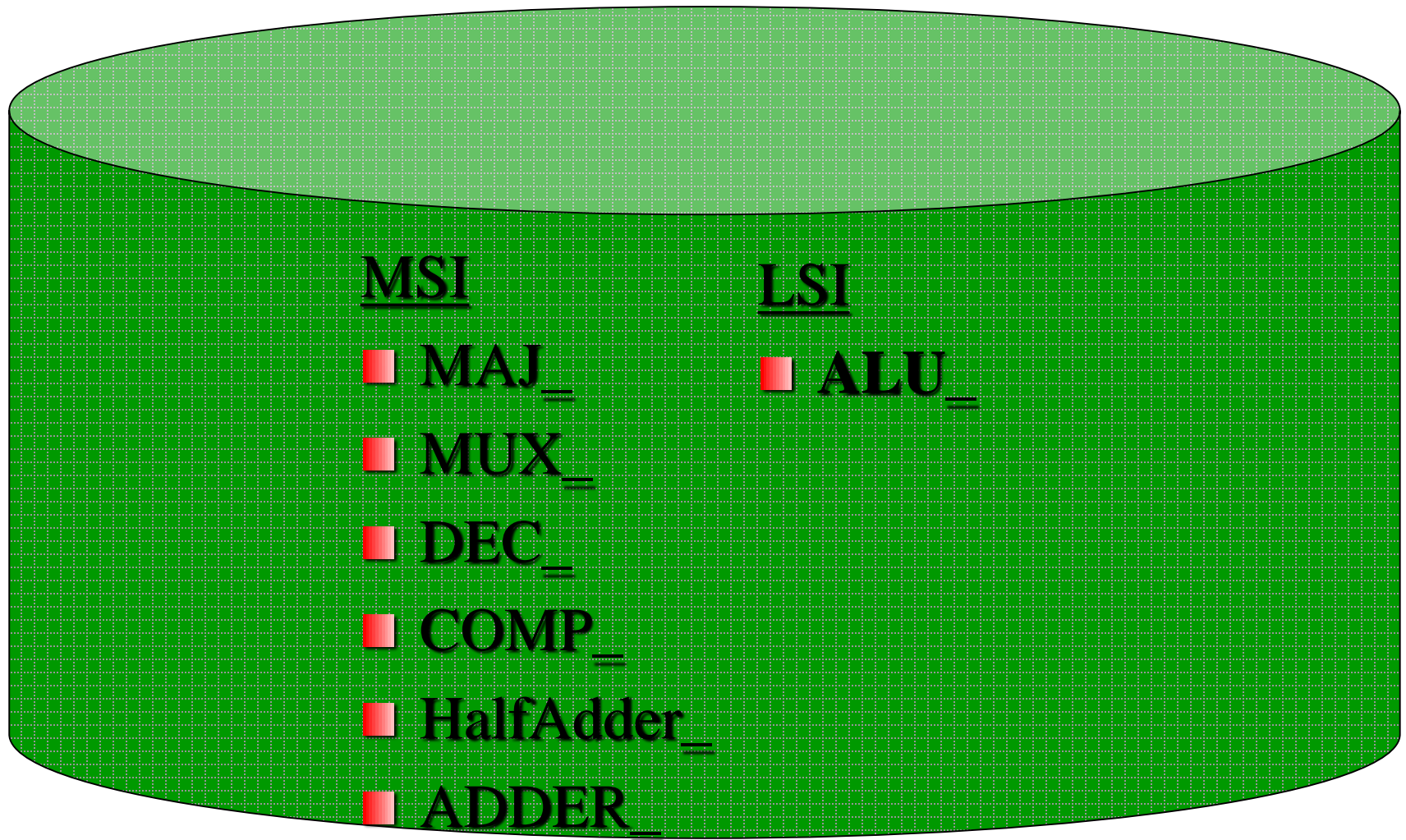
ALU_8bits



FLAGS

Carry
Zéro
Négatif
oVerflow

Notre bibliothèque de portes



Architecture générale de P-ARM

2. Contrôleur

3. Chemin de données

