AE 3

Du binaire à l'information

-

Quelques circuits élémentaires

Objectifs

- **Connaître les circuits élémentaires :**
  - **Fonctions arithmétiques : additionneurs, comparateurs**
  - **Fonctions combinatoires : codeurs, décodeurs, multiplexeurs, démultiplexeurs**

- **Du binaire à l'information**

Tout est dit dans le cours « codage »

A partir d'un bit….

- **Fermé = 1 = le courant passe**
- **Ouvert = 0 = le courant ne passe pas**

- **Un bit (BInary digIT) peut prendre la valeur 0 ou 1 et permet de représenter 2 informations**
  - **Présent/absent**
  - **Ecoute/dort**
  - **Homme/femme**
  - **Droite/gauche**
  - **Pour/contre**
- **Et si on associait plusieurs bits?**

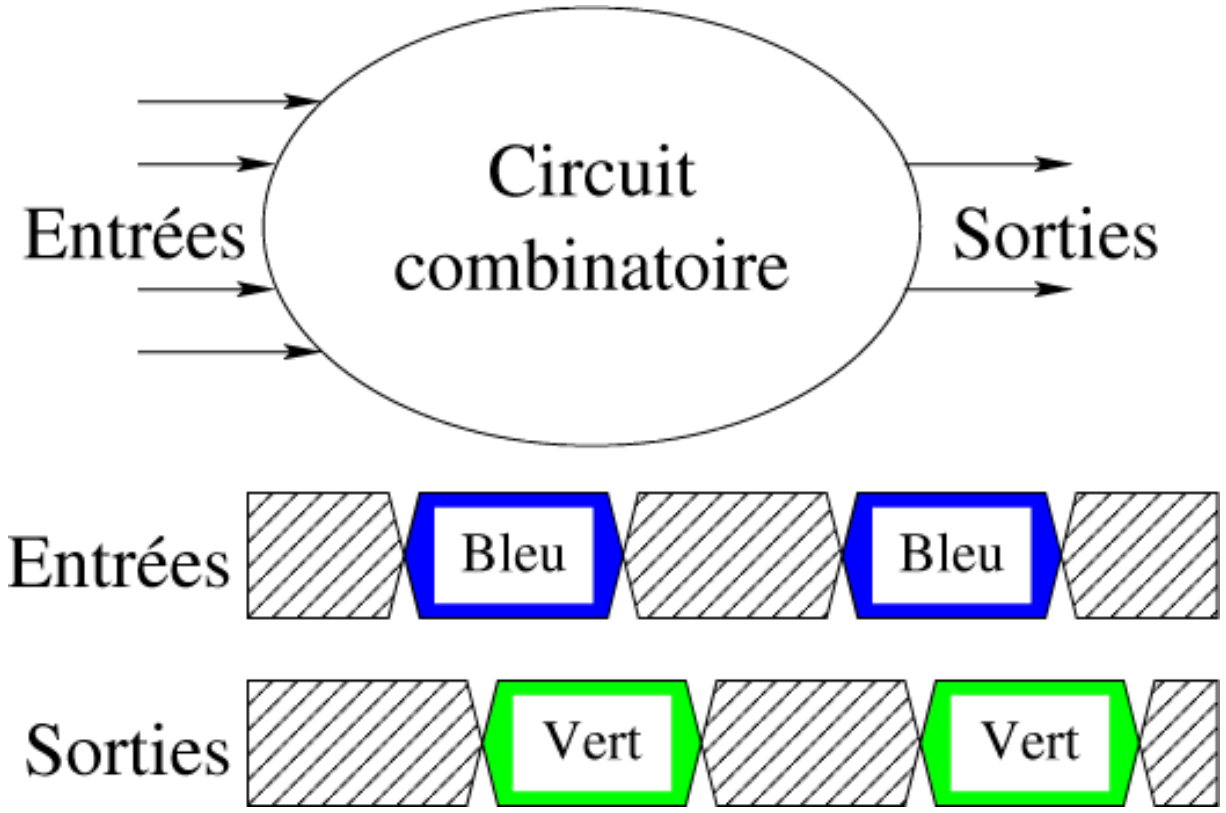
Plusieurs Bits

- 2 bits = 00 01 10 11 soit 4 informations différentes
  - Nord, Sud , Est, Ouest
- 3 bits = 000 001 010 011 100 101 110 111 ,soit 8 informations
- N bits = 2^n informations différentes
- On parle de mots:
  - 8 bits (octets)
  - 16, 32, 64, 128 …
  - Taille multiple de 2, car l'adresse sera codé en binaire aussi

- L'utilisateur (ou le constructeur) donne une signification à la valeur d'un mot

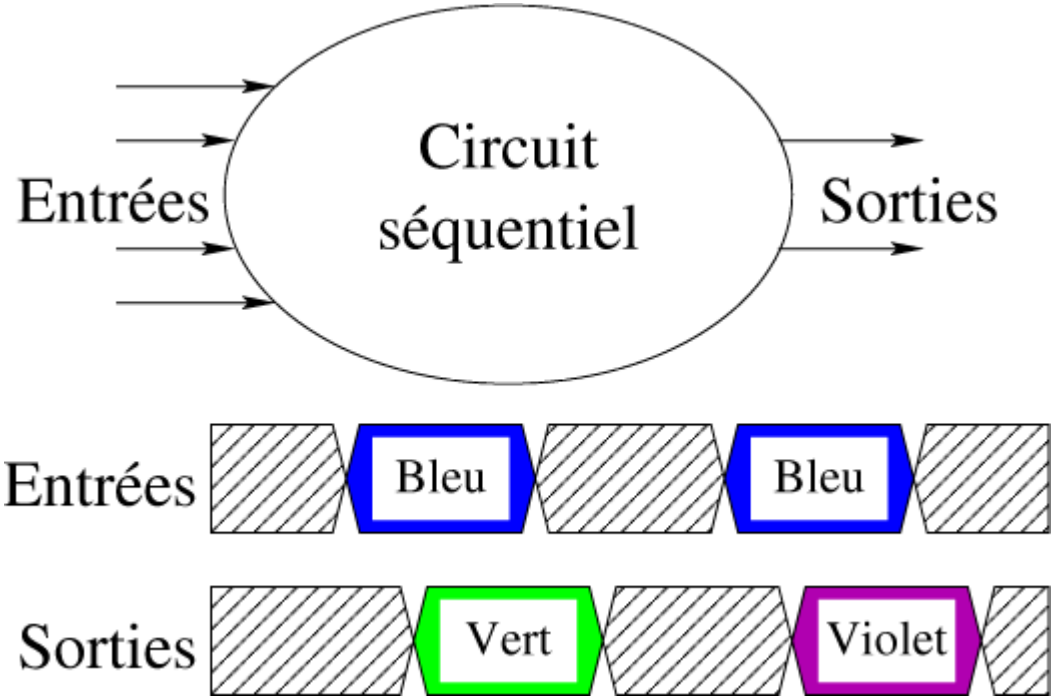
Circuits combinatoire

- A présentation, à des instants différents, des mêmes valeurs d'entrée produira à chaque fois les mêmes résultats



Circuits séquentiels

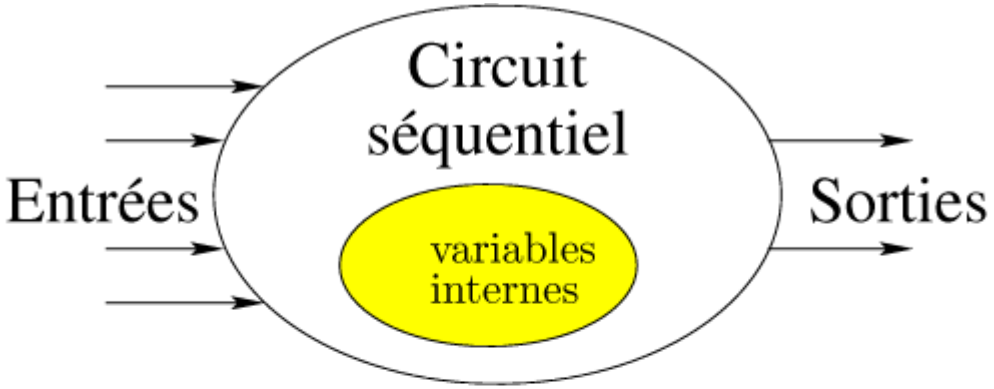
- **Les mêmes entrées du circuit ne produisent pas toujours le même résultat**

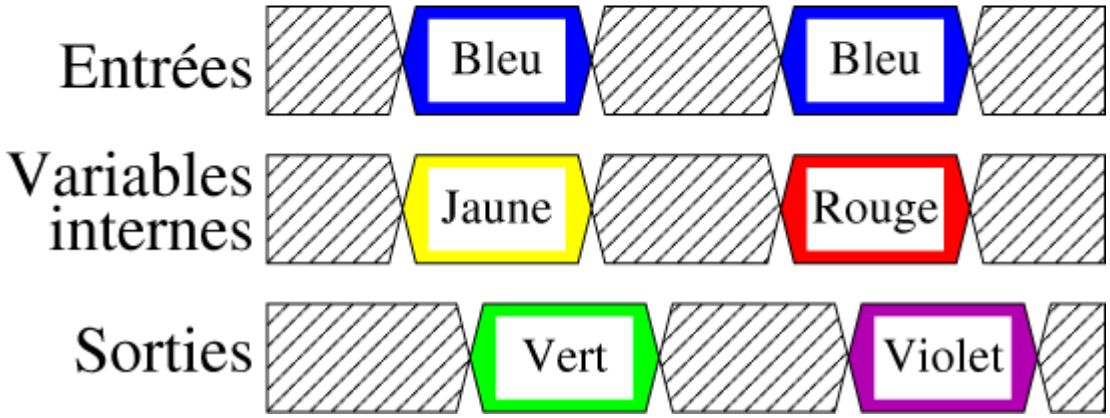


- **Comment est-ce possible?**

Des variables internes

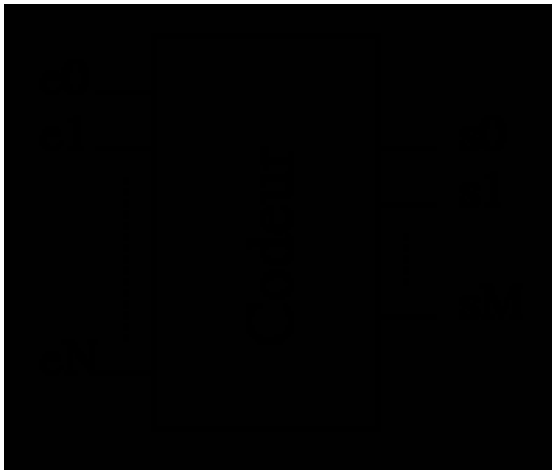
- **Variables supplémentaires internes au circuit dont la valeur évolue au cours du temps qui est la cause de ce non-déterminisme apparent**





Codeurs

- **Code une entrée à E entrées où il n ’y a qu'un seul 1 en binaire vers S sorties**
  - **S = log2(E)**
  - **Ici « codeur N+1 vers M+1 »**
  - **Exemple codeur 8 => 3**
    - **8 entrées dont une vaut 1**
    - **Produit une valeur entre 0 et 7**
    - **correspondant au numéro de l’entrée active**



Codeurs

- **Codeur 3 vers 2 : (N=2, M=1)**

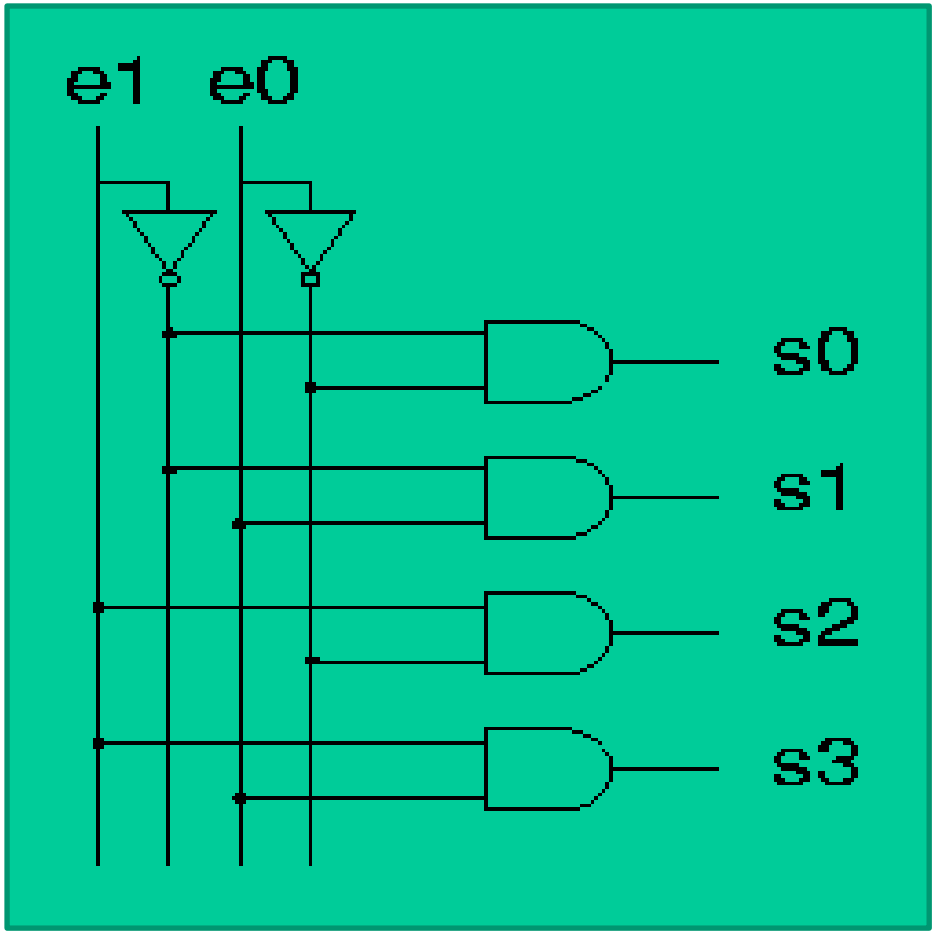
e2	e1	e0	s0	s1
0	0	0	*	*
0	0	1	0	0
0	1	0	0	1
0	1	1	*	*
1	0	0	1	0
1	0	1	*	*
1	1	0	*	*
1	1	1	*	*

Décodeurs

- **Inverse du codeur**
- **Décode un mot binaire**
  - **Active le fil de sortie correspondant à la valeur en entrée**
  - **Les autres à zéro**
  - **« décodeur N+1 vers M+1 »**
- **Exemple :**
  - **N=1 M=2**
  - **« décodeur 2 vers 3 »**

e1	e0	s0	s1	s2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	0	0

Décodeur 2 vers 4



Quelques portes élémentaires suffisent

Décodeurs et fonctions logiques

- **Toute fonction logique f(x0, x1, … , xN) peut être réalisée avec un décodeur. C'est la table de vérité!**
- **Exemple l 'additionneur élémentaire :**

A	B	Ce	#	S	C
0	**0**	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**1**	**0**
0	**1**	**0**	**2**	**1**	**0**
0	**1**	**1**	**3**	**0**	**1**
1	**0**	**0**	**4**	**1**	**0**
1	**0**	**1**	**5**	**0**	**1**
1	**1**	**0**	**6**	**0**	**1**
1	**1**	**1**	**7**	**1**	**1**

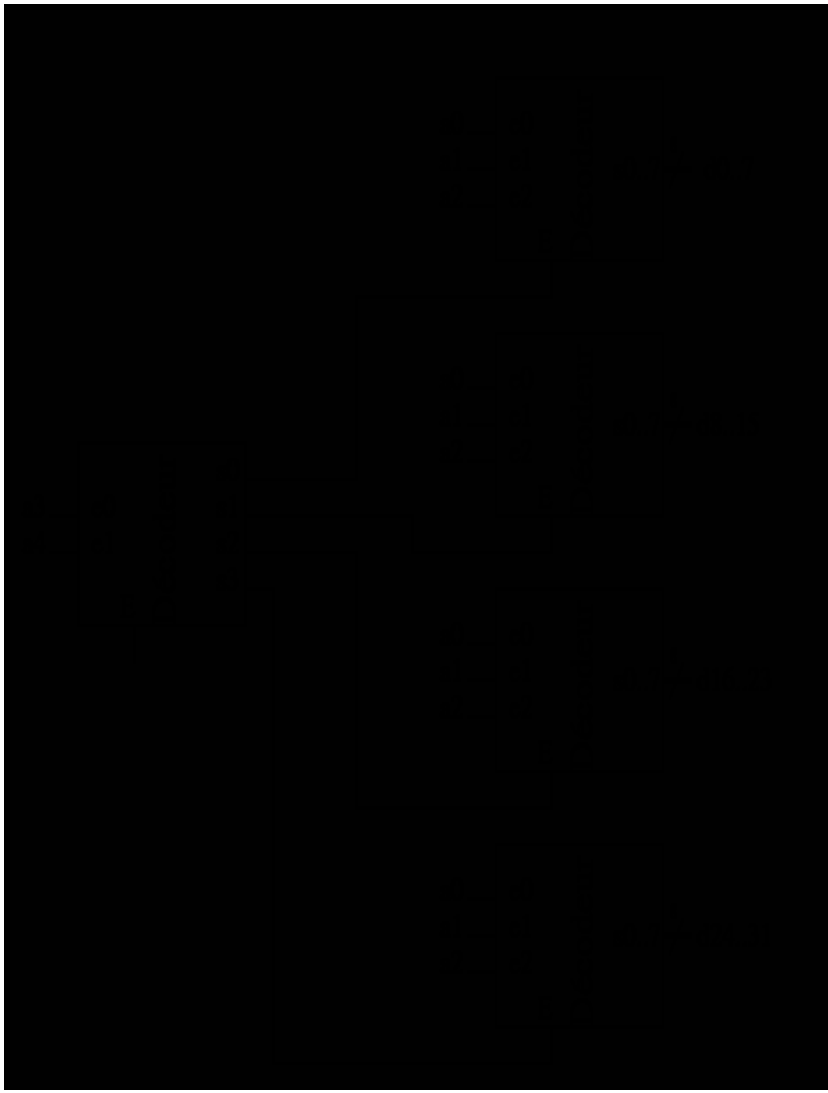
Association de décodeurs

- **Faire un 4=> 16 avec deux 3=>8**
- **On ajoute une entrée « Enabled »**
  - **si E = 0 alors toutes les sorties sont à 0**
  - **On peut continuer pour construire de plus « gros » décodeurs…**

Décodeur 4 vers 16 à partir de décodeur 3 vers 8

Un arbre de décodeurs

- **Les bits de poids forts permettent de sélectionner un seul décodeur**
- **Celui-ci décode alors les bits de poids faible**
- **Décodeur 5 vers 32 avec des décodeurs 3 vers 8 (et un 2 vers 4)…**



Sélecteurs

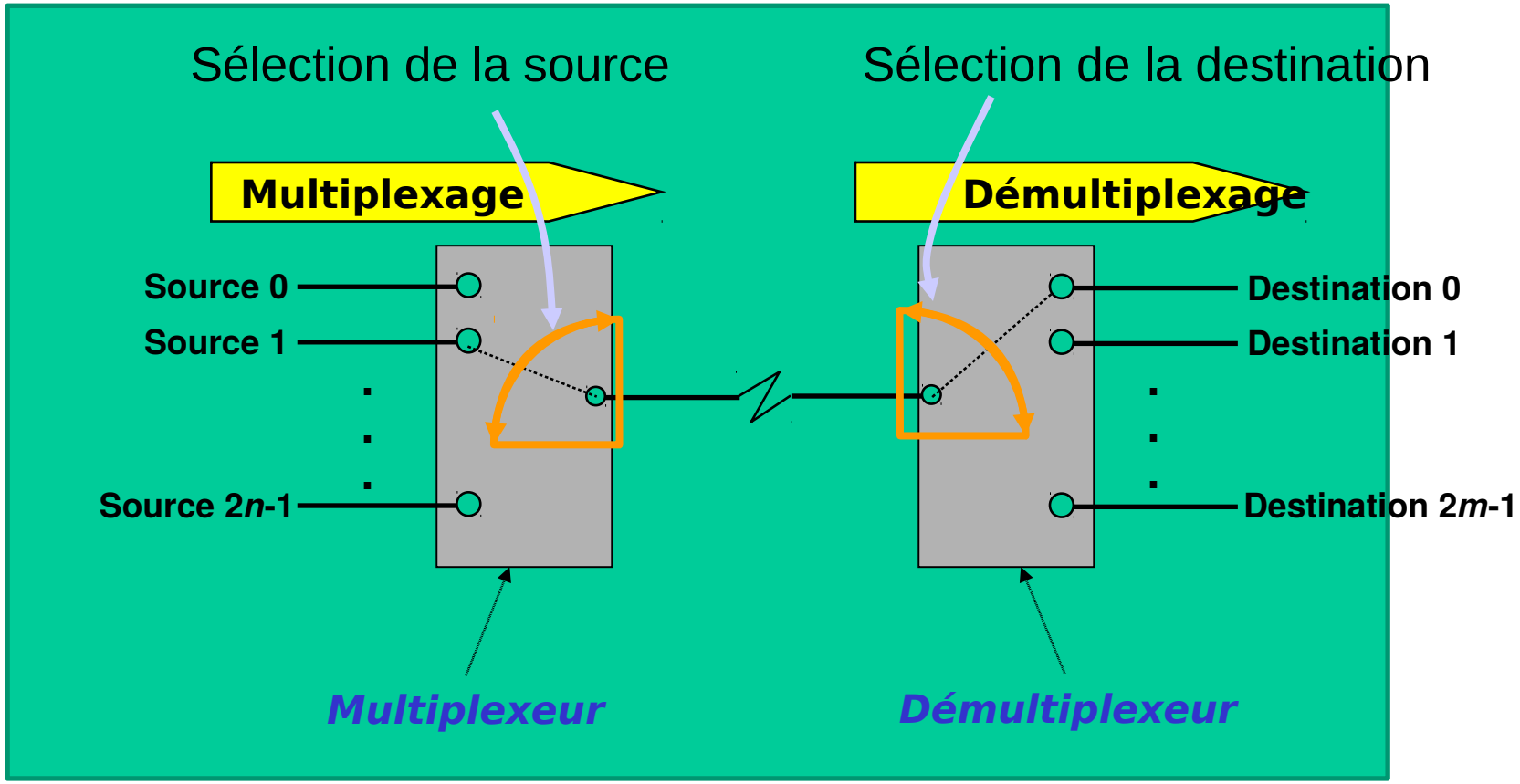
- Un sélecteur prend N entrées (e0..N-1) et N commandes (c0..N-1) et copie l 'entrée eJ si cJ est la seule à être à 1.
  - sJ = 1 ssi eJ = 1 et cJ est la seule commande à 1
- Exemple à 2 entrées :

c0	c1	s
0	0	*
0	1	e1
1	0	e0
1	1	*

- Beaucoup d'entrées!!!!

Multiplexeurs et Démultiplexeurs

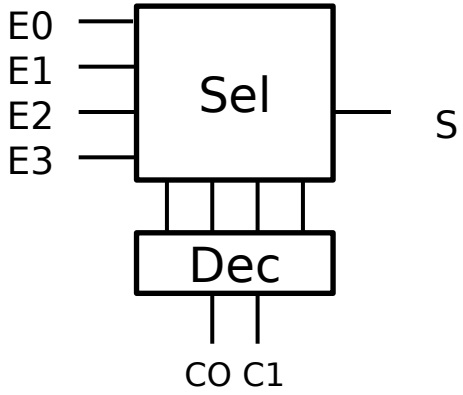
- **Utilisation : partage d'une ligne par plusieurs!!**



Multiplexeurs

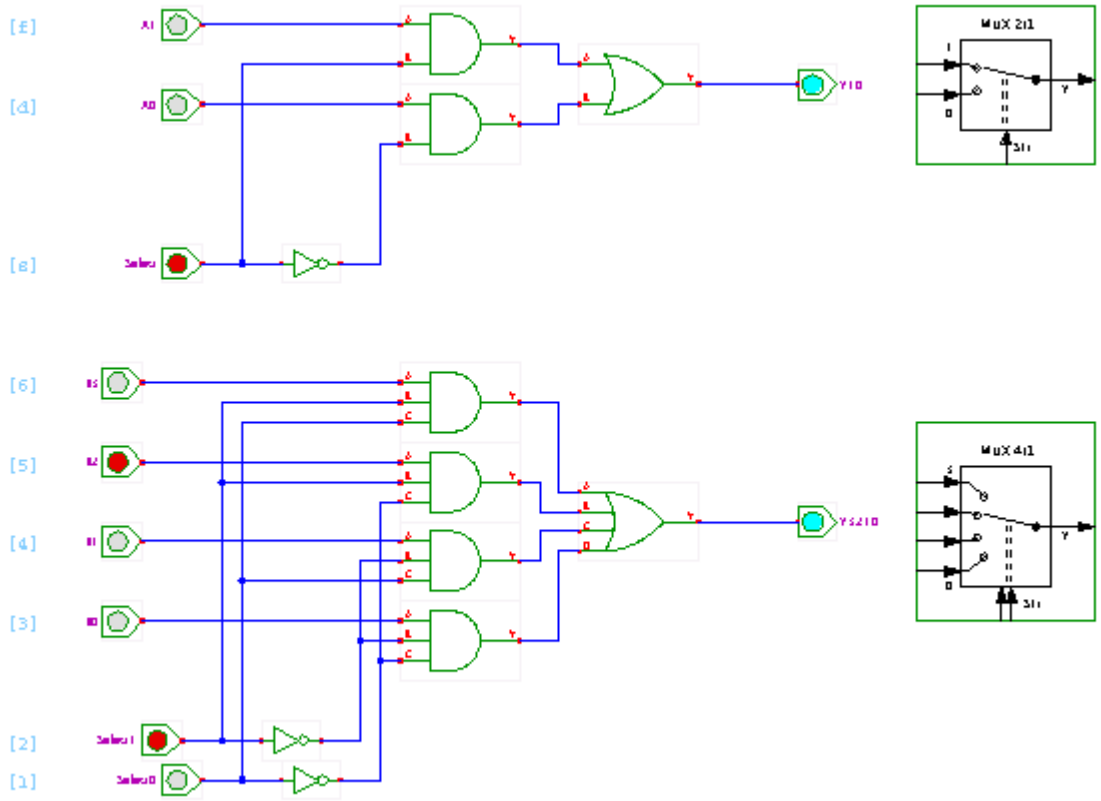
- **Le multiplexeur (MUX) sélectionne, à l'aide de N entrées de commande (c0..cN-1), une des *2*N entrées d'information (e0..2N-1)et la dirige à la sortie.**
  - **La sortie est égale à 1 ssi l 'entrée dont le numéro est (c0c1…cN-1)2 est vraie.**
- **Exemple: Multiplexeur 4 à 1.**

c0	c1	s
0	0	e0
0	1	e1
1	0	e2
1	1	e3



- **On peut utiliser un *décodeur* et un *sélecteur*…**

Avec des portes logiques…If ou case

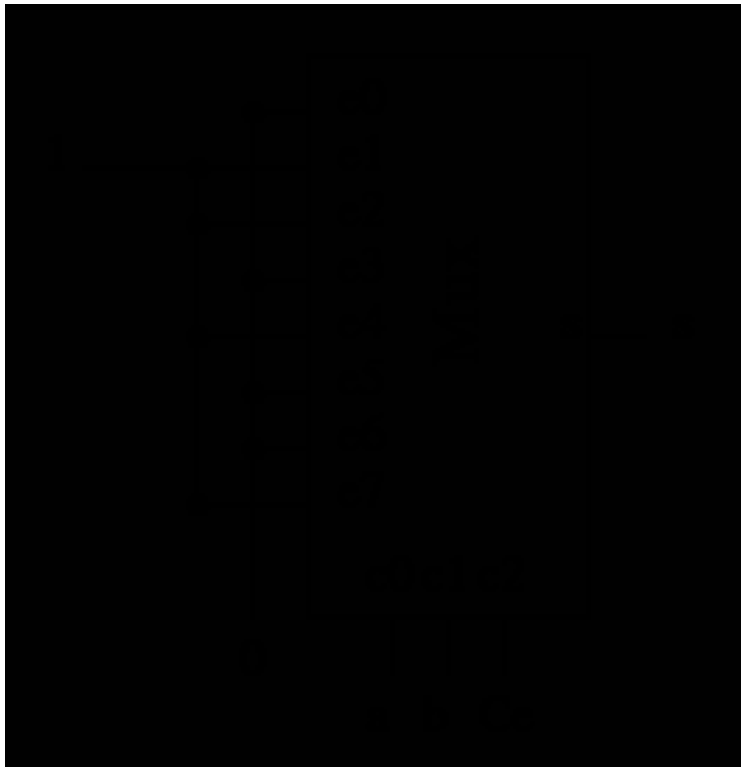


Multiplexeurs

- **Synthèse de fonctions logiques avec un multiplexeur**
- **Exemple de l ’AE… *(encore !)***

s = f1(a, b, Ce) = ❮(1, 2, 4, 7)

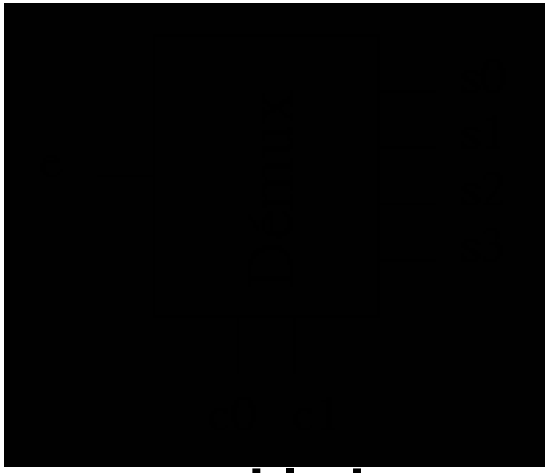
a	b	Ce	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Démultiplexeurs

- **Opérateur inverse du multiplexeur. Il distribue le bit e reçu en entrée unique vers l'une des 2N sorties possibles *D* (N: nombre d'entrées de sélection *S*).**
- **Exemple multiplexeur 4 sorties :**

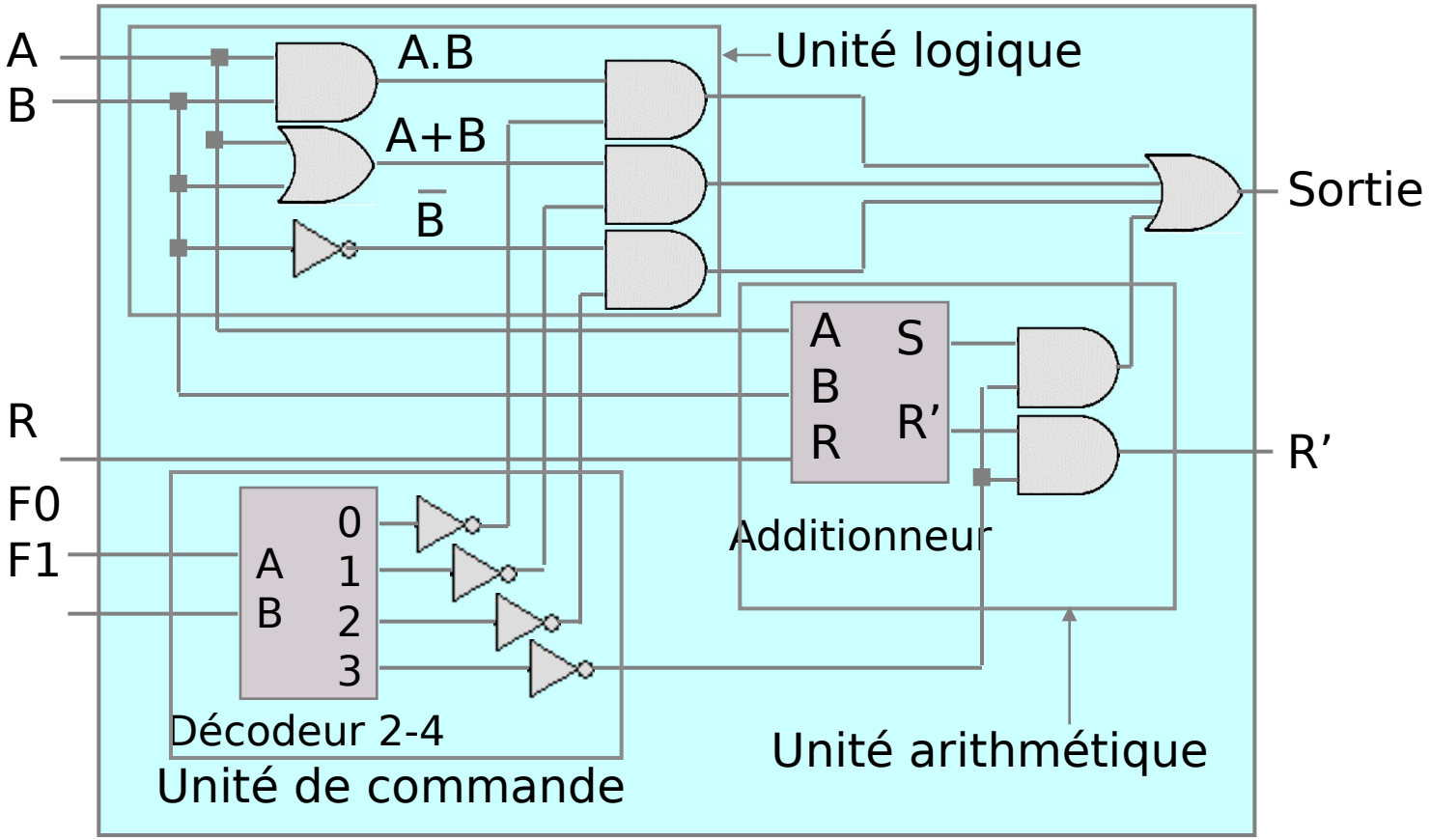
c0	c1	s0	s1	s2	s3
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e



- **C ’est juste un décodeur avec e comme « enabled ».**

Un début de processeur!

- **UAL élémentaire**



Logique avec ROM

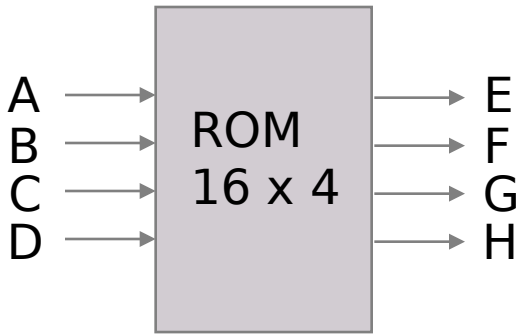
- Il est possible de réaliser des circuits logiques au moyen de mémoires ROM (Read-Only Memory).
  - Aucune simplification n'est nécessaire.
  - Les entrées de la table de vérité servent d'adresse dans la ROM
  - Le contenu de chaque adresse est la sortie désirée pour cette combinaison de variables d'entrée, la sortie pouvant avoir un ou plusieurs bits.

Exemple : transcodeur binaire à code Gray

- **Logique avec ROM**

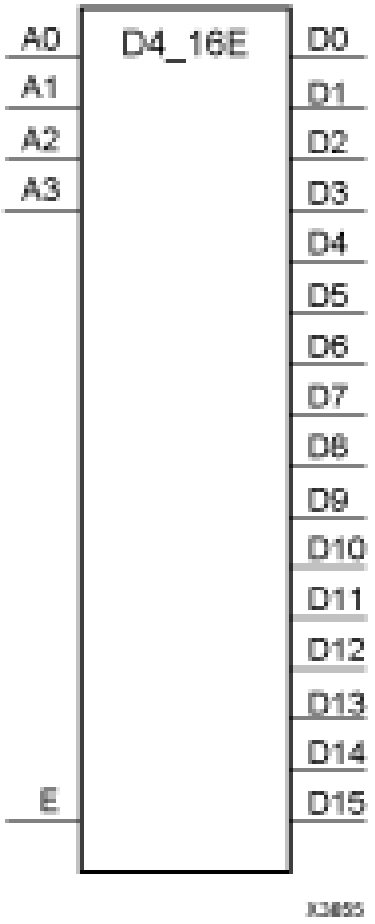
Table de vérité

ABCD	EFGH	ABCD	EFGH
0000	0000	1000	1100
0001	0001	1001	1101
0010	0011	1010	1111
0011	0010	1011	1110
0100	0110	1100	1010
0101	0111	1101	1011
0110	0101	1110	1001
0111	0100	1111	1000



D4_16E

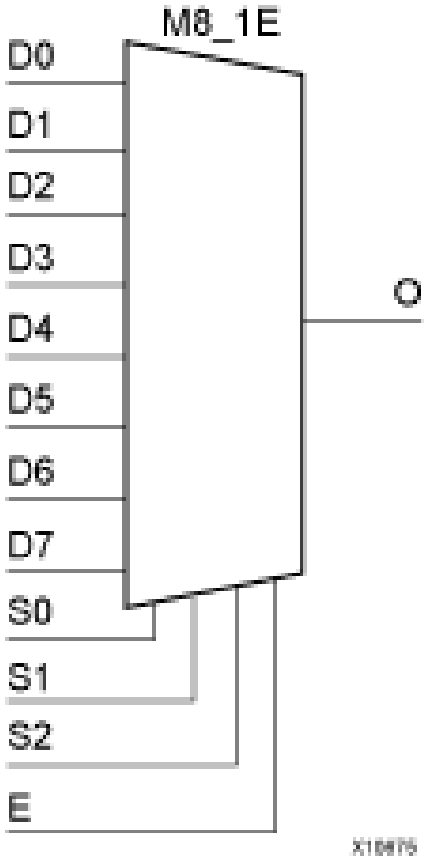
- Macro: 4- to 16-Line Decoder/Demultiplexer with Enable
- This design element is a decoder/demultiplexer. When the enable (E) input of this design element is High, one of 16 active-High outputs (D15 : D0) is selected with a 4-bit binary address (A3 : A0) input. The non-selected outputs are Low. Also, when the E input is Low, all outputs are Low. In demultiplexer applications, the E input is the data input.



M8_1E

Macro: 8-to-1 Multiplexer with Enable

This design element is an 8-to-1 multiplexer with enable. When the enable input (E) is High, the M8_1E multiplexer chooses one data bit from eight sources (D7 : D0) under the control of the select inputs (S2 : S0). The output (O) reflects the state of the selected input. When (E) is Low, the output is Low.



A lire

- http://tk5yp.fr/elec/elec.htm
- http://www.info.univ-tours.fr/~marcel/archi/node63.htm
- http://comelec.enst.fr/tpsp/eni/poly/ l