

ARCHITECTURE DES ORDINATEURS

UVHC – ISTV – Licence 1
Informatique

Rabie Ben Atitallah

rabie.benatitallah@univ-valenciennes.fr

PROGRAMME

Séance 1 Un cours d'architecture pour des informaticiens

Séance 2 De l'électronique à l'informatique

Séance 3 Du binaire à l'information

Séance 4 Notion de circuit logique

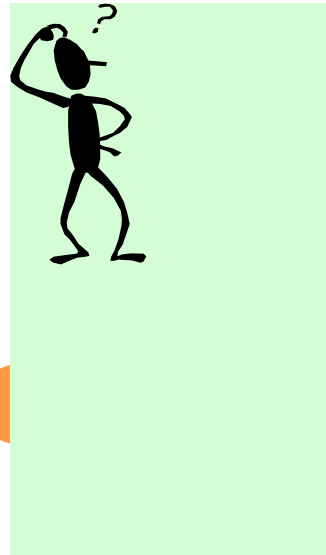
Séance 5 Un modèle d'exécution : Von Neumann

Séance 6 Comment mémoriser une donnée ?

Séance 7 Comment réaliser un calcul ?

Séance 8 Instruction et ordonnancement

CHAP. 2 : DU BINAIRE À L'INFORMATION – QUELQUES CIRCUITS ÉLÉMENTAIRES



Les circuits élémentaires

Fonctions arithmétiques :

additionneurs, comparateurs

Fonctions combinatoires :

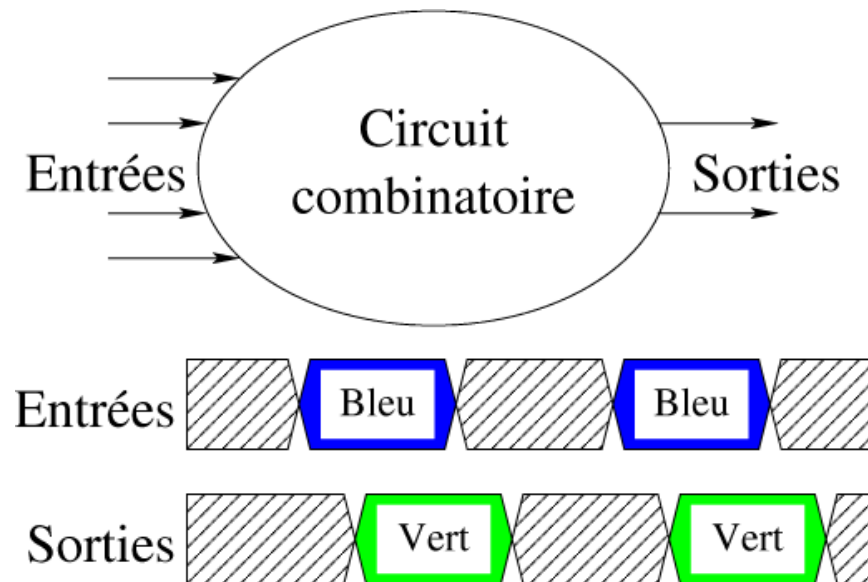
Codeurs, décodeurs, multiplexeurs,
démultiplexeurs

À PARTIR D'UN BIT

- Un bit (BInary digIT) peut prendre la valeur 0 ou 1 et permet de représenter 2 informations
 - Fermé = 1 = le courant passe
 - Ouvert = 0 = le courant ne passe pas
- Et si on associait plusieurs bits?
 - 2 bits = 00 01 10 11 soit 4 informations différentes
 - 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

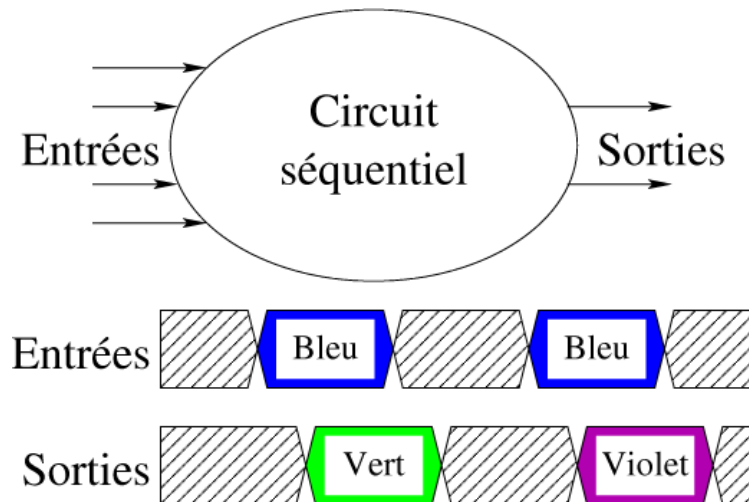
CIRCUITS COMBINATOIRES

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



CIRCUITS SÉQUENTIELS

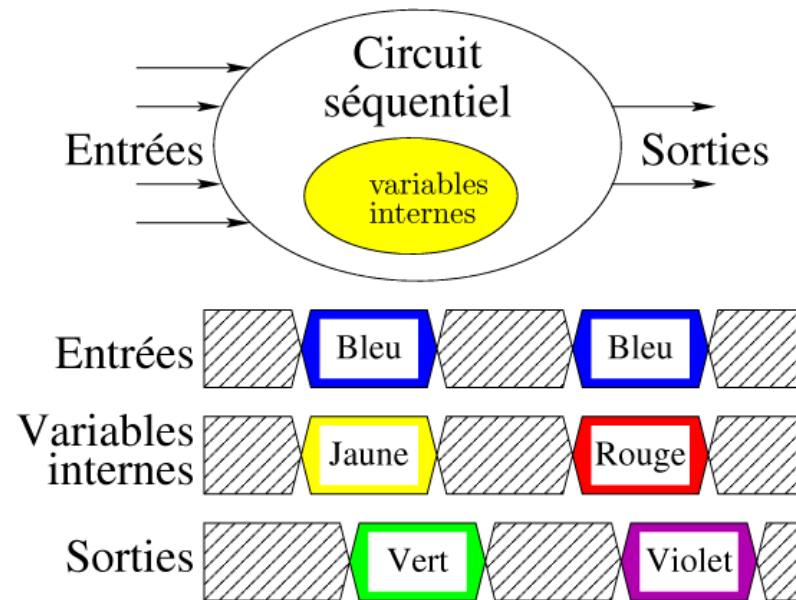
- Les mêmes entrées du circuit ne produisent pas toujours le même résultat



- Comment est-ce possible?

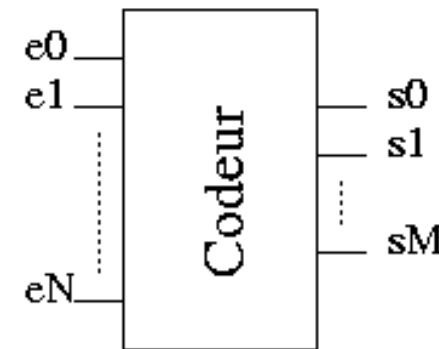
DES VARIABLES INTERNES

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



CODEURS

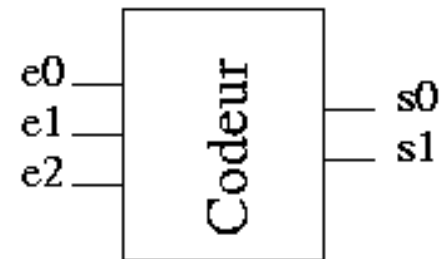
- Code une entrée à E entrées où il n'y a qu'un seul 1 en binaire vers S sorties
 - $S = \log_2(E)$
 - Ici « codeur N+1 vers M+1 »
- Exemple codeur 8 => 3
 - 8 entrées dont une vaut 1
 - Produit une valeur entre 0 et 7
 - correspondant au numéro de l'entrée active



CODEURS

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

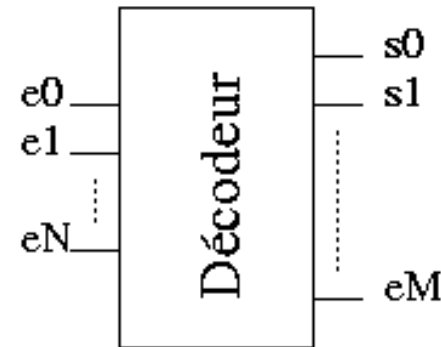
e2	e1	e0	s0	s1
0	0	0	*	*
0	0	1	0	0
0	1	0	0	1
0	1	1	*	*
1	0	0	1	0
1	0	1	*	*
1	1	0	*	*
1	1	1	*	*



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 $M+1$ »



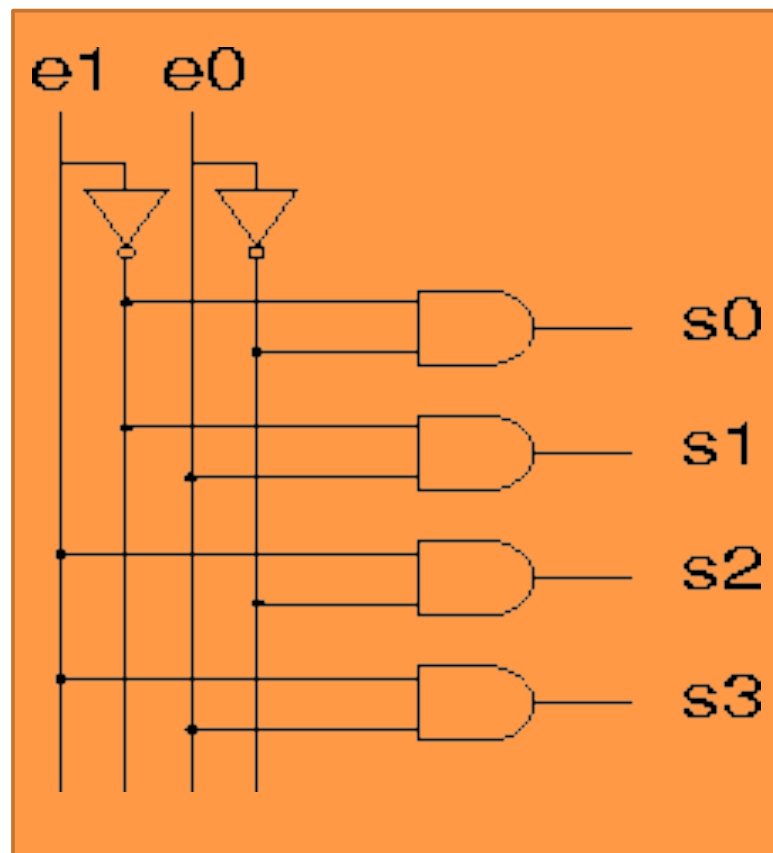
○ Exemple :

- $N=1$ $M=2$
- « 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

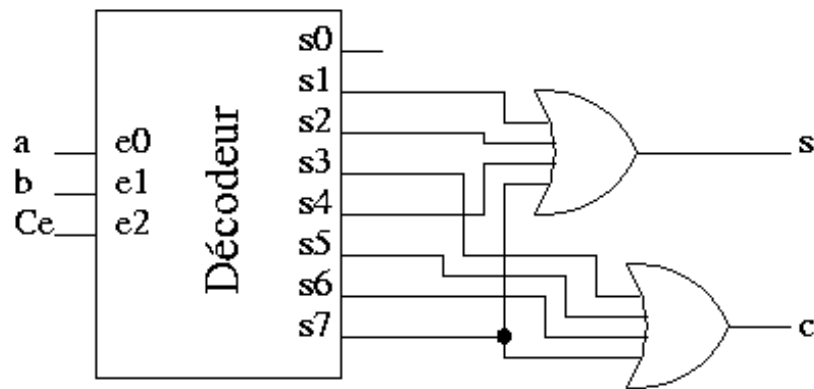
DÉCODEUR 2 VERS 4

- Quelques portes élémentaires suffisent



DÉCODEURS ET FONCTIONS LOGIQUES

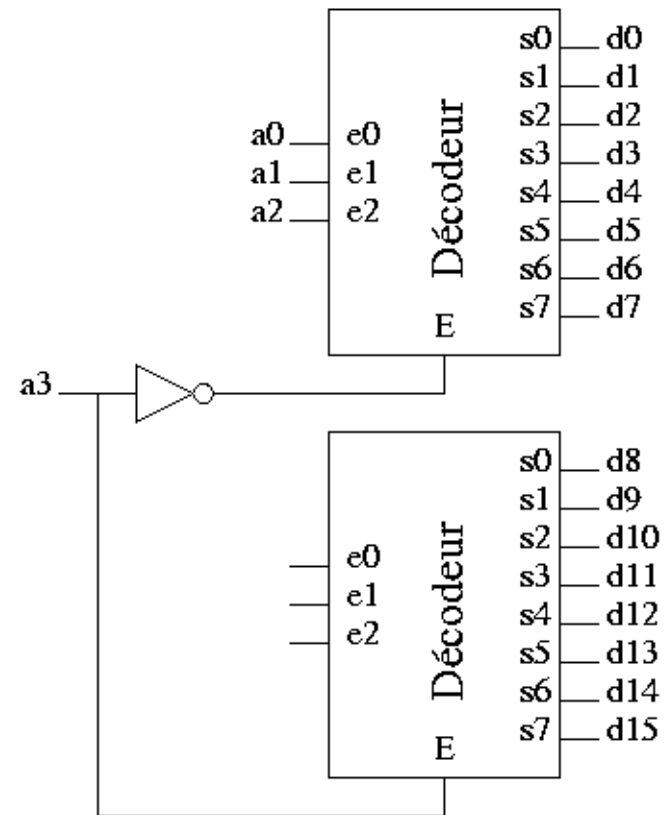
- Toute fonction logique $f(x_0, x_1, \dots, x_N)$ peut être réalisée avec un décodeur. C'est la table de vérité!
- Exemple 1 'additionneur élémentaire :



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

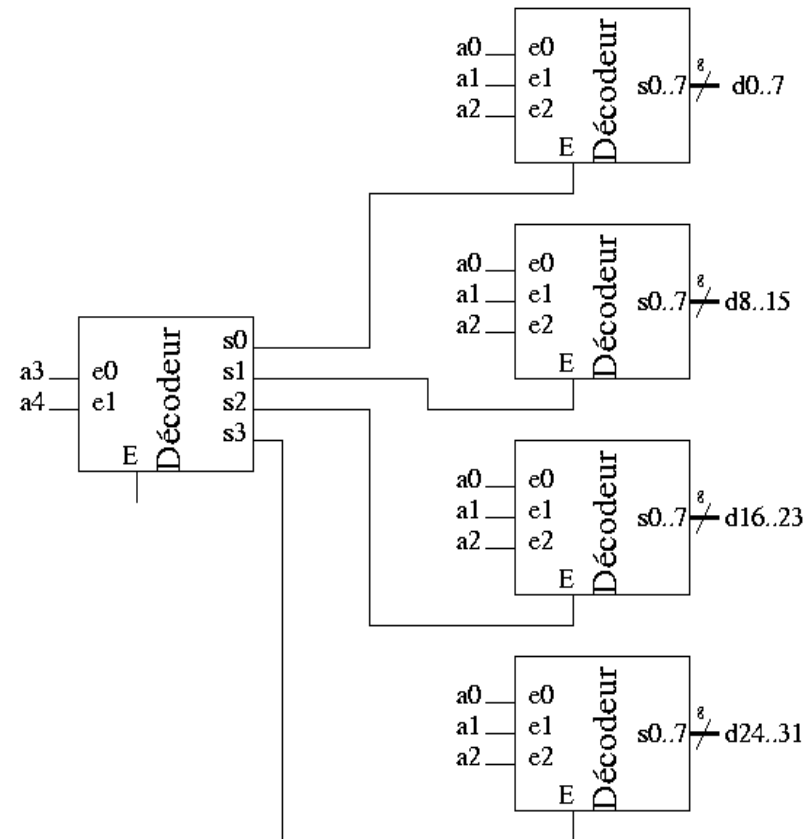
ASSOCIATION DE DÉCODEURS

- 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)...



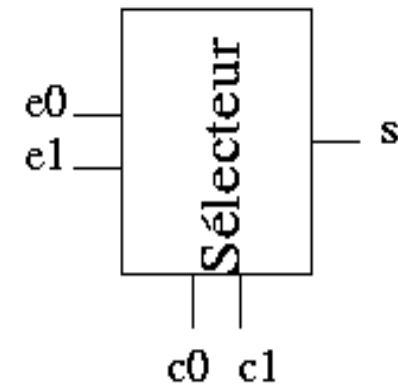
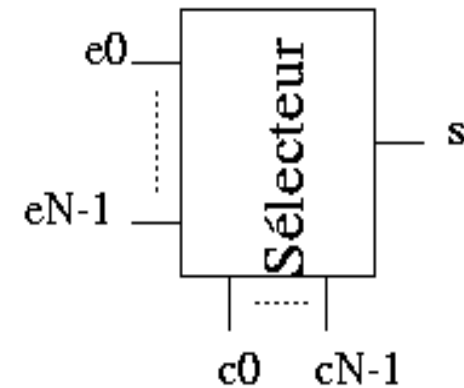
SÉLECTEURS

- Un sélecteur prend N entrées ($e_0..e_{N-1}$) et N commandes ($c_0..c_{N-1}$) et copie l'entrée e_J si c_J est la seule à être à 1.
- $s_J = 1$ ssi $e_J = 1$ et c_J est la seule commande à 1

- Exemple à 2 entrées :

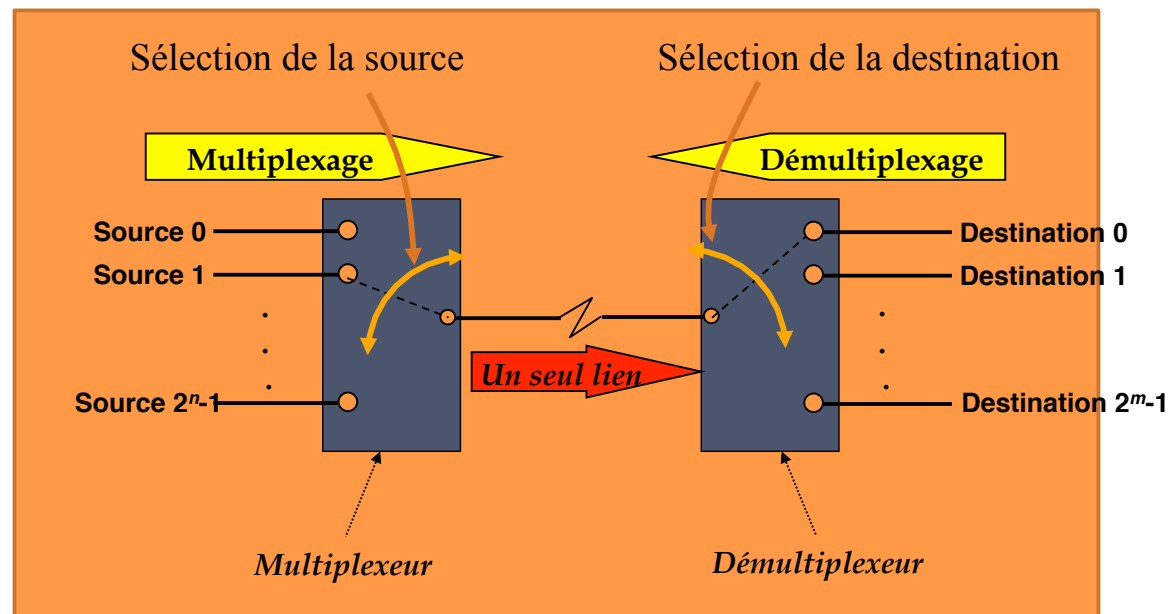
c0	c1	s
0	0	*
0	1	e1
1	0	e0
1	1	*

- Beaucoup d'entrées!!



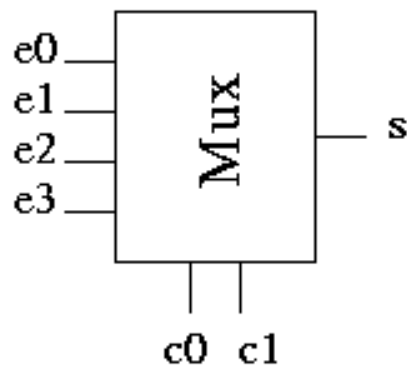
MULTIPLEXEURS ET DÉMULTIPLEXEURS

- Utilisation : partage d'un ligne par plusieurs !!

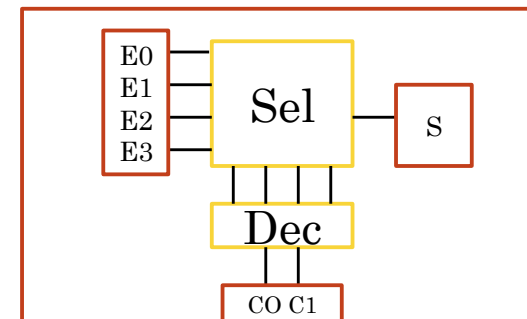


MULTIPLEXEURS

- Le multiplexeur (MUX) sélectionne, à l'aide de N entrées de commande ($c_0..c_{N-1}$), une des 2^N entrées d'information ($e_0..2^N-1$) et la dirige à la sortie.
 - La sortie est égale à 1 ssi l'entrée dont le numéro est $(c_0c_1...c_{N-1})_2$ est vraie.
 - Exemple: Multiplexeur 4 à 1.

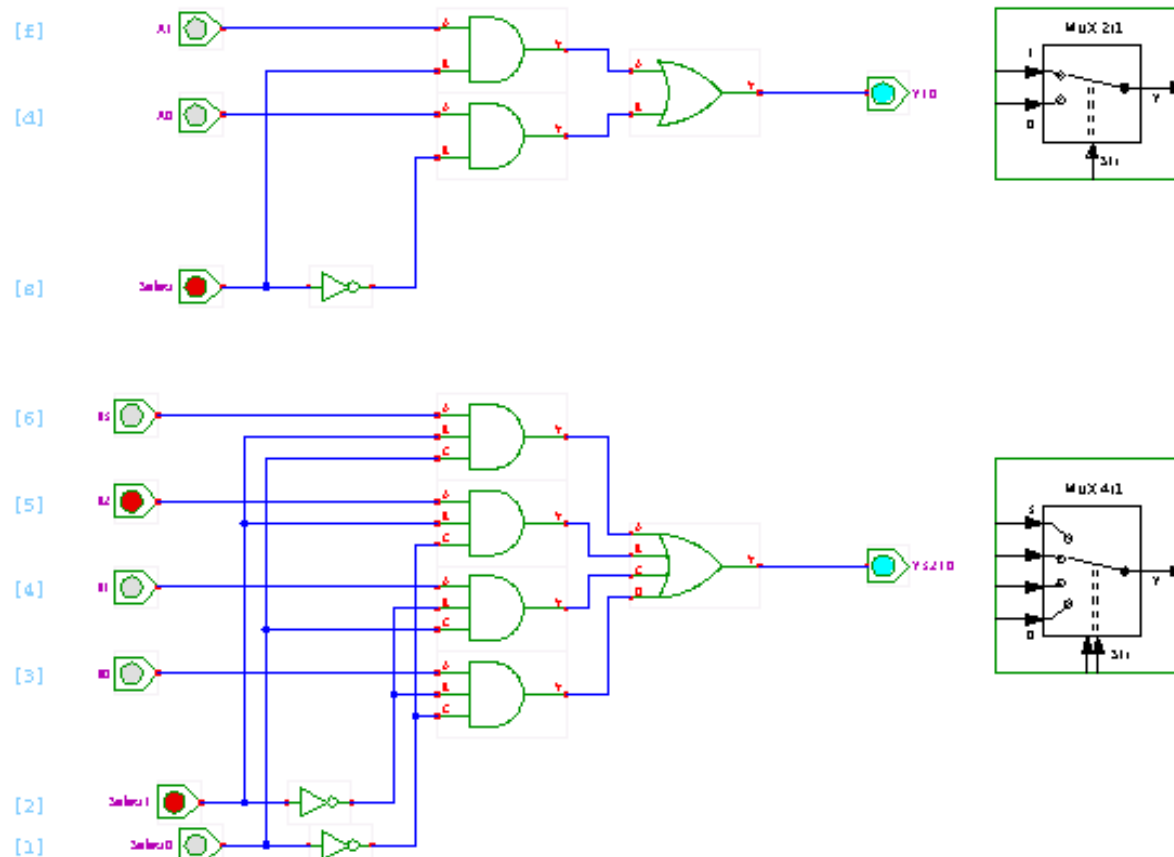


c0	c1	s
0	0	e0
0	1	e1
1	0	e2
1	1	e3



- On peut utiliser un décodeur et un sélecteur...

AVEC DES PORTES LOGIQUES...IF OU CASE

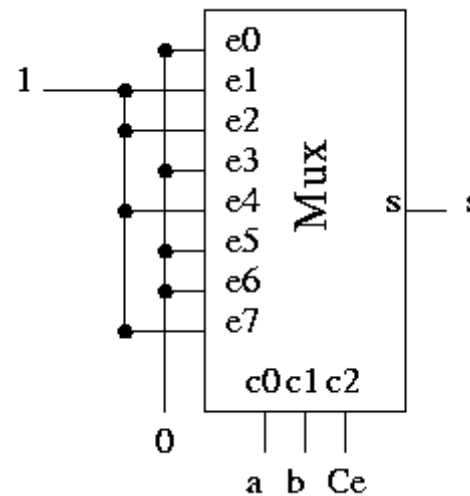


MULTIPLEXEURS

- Synthèse de fonctions logiques avec un multiplexeur

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

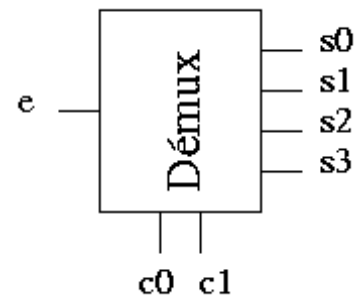
$$s = f1(a, b, Ce) = \Sigma(1, 2, 4, 7)$$



DÉMUPLEXEURS

- 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 :

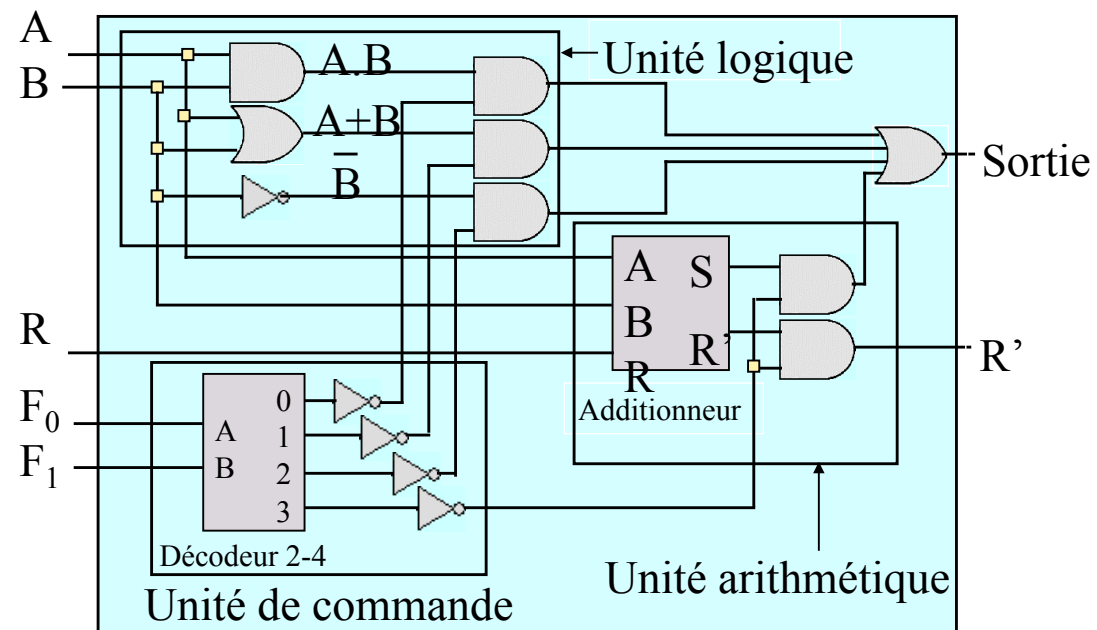
c0	c1	s0	s1	s2	s3
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e



- 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.