

Architecture des ordinateurs

Cours 2

1er octobre 2012

Archi

1/20

Circuits combinatoires

Circuits arithmétiques

Spécification

Décrire clairement :

- les **entrées** :
 - données : ne sont pas des entrées de la table de vérité,
 - paramètres : bits de réglage
 - variables d'entrée.
- la **sortie** : pas forcément unique !
 - fonction logique : une seule valeur en sortie.
 - circuit : possiblement plusieurs fonctions pour obtenir le comportement voulu.
- le **rôle** de différents éléments :
 - À quoi sert le circuit ?
 - Qu'obtient-on en sortie ?
 - Quel rôle jouent les entrées ?
- la **table de vérité** (une table par fonction)

Archi

3/20

Circuits combinatoires

Archi

2/20

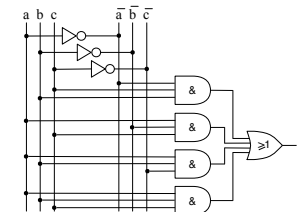
Circuits combinatoires

Circuits arithmétiques

Exemples : circuits pour la fonction majoritaire

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

fonction majoritaire
"classique"



a	b	c	d	S ₀	S ₁
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Pour tester la majorité sur un **nombre pair d'entrées**, une seule sortie ne suffit pas :

- soit les 0 sont majoritaires (sortie 00)
- soit les 1 sont majoritaires (sortie 01)
- soit il n'y a pas de majorité (10)

▷

Archi

4/20

Multiplexeur $2^n \times n$

Il existe un certain nombre de circuits combinatoires classiques... comme par exemple, le multiplexeur $2^n \times n$:

- **Entrées :**

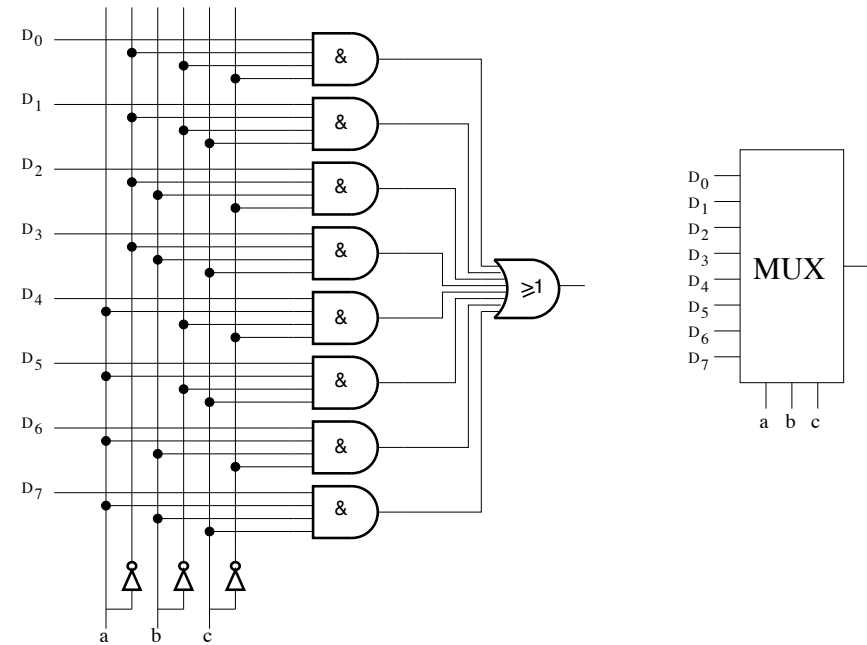
- 2^n lignes d'entrée (données) : D_0, \dots, D_{2^n-1}
- n lignes de *sélection* : a, b, c, \dots

- **Sortie :** Une seule sortie S

- **Rôle :** *Aiguiller* la valeur de l'une des 2^n lignes d'entrée vers la sortie S . La ligne d'entrée choisie est désignée grâce aux bits de sélection.

a	b	c	S
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

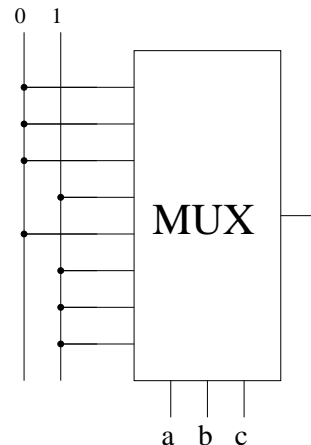
Câblage du multiplexeur 8×3



Exemple d'utilisation du multiplexeur

La fonction majoritaire avec un multiplexeur :

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



▷ Multiplexeur = Système combinatoire **universel**

Démultiplexeur $2^n \times n$

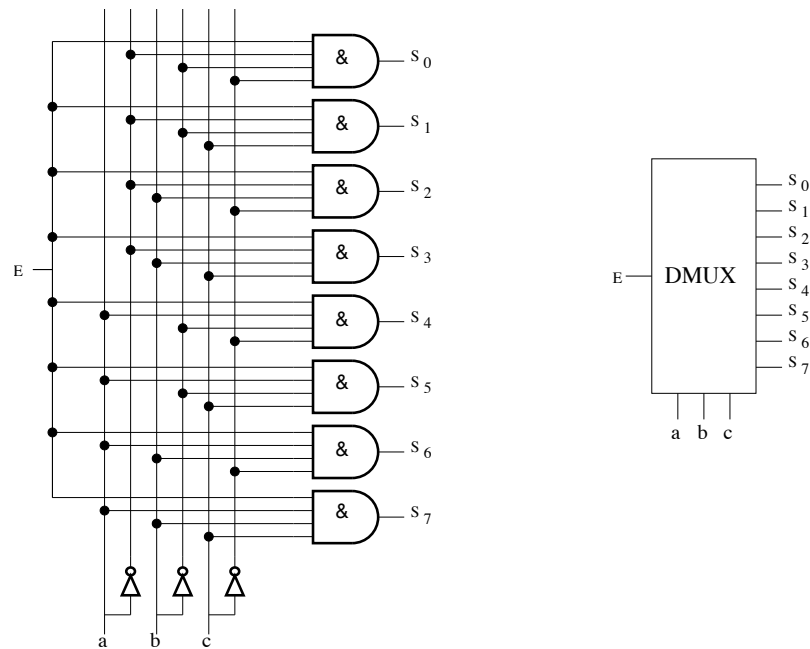
- **Entrées :**

- une ligne d'entrée (donnée) : E
- n lignes de *sélection* : a, b, c, \dots

- **Sortie :** 2^n lignes de sortie S_0, \dots, S_{2^n-1}

- **Rôle :** *Aiguiller* l'entrée E vers l'une des 2^n lignes de sortie. La ligne de sortie est désignée grâce aux bits de sélection.

a	b	c	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	0	0	E	0	0	0	0	0	0	0
0	0	1	0	E	0	0	0	0	0	0
0	1	0	0	0	E	0	0	0	0	0
0	1	1	0	0	0	E	0	0	0	0
1	0	0	0	0	0	0	E	0	0	0
1	0	1	0	0	0	0	0	E	0	0
1	1	0	0	0	0	0	0	0	E	0
1	1	1	0	0	0	0	0	0	0	E

Câblage du démultiplexeur 8×3 

Archi

9/20

Circuits combinatoires

Circuits arithmétiques

Décodeur $2^n \times n$

- **Entrées :**
 - n lignes de *sélection* : a, b, c, \dots
- **Sortie :** 2^n lignes de sortie S_0, \dots, S_{2^n-1}
- **Rôle :** *Sélectionner* (mettre à 1) l'une des 2^n lignes de sortie. La ligne de sortie est codée par les bits de sélection.

a	b	c	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

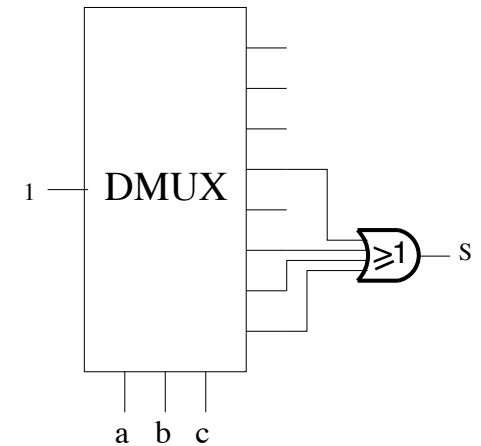
Archi

11/20

Exemple d'utilisation d'un démultiplexeur

La fonction majoritaire avec un démultiplexeur :

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



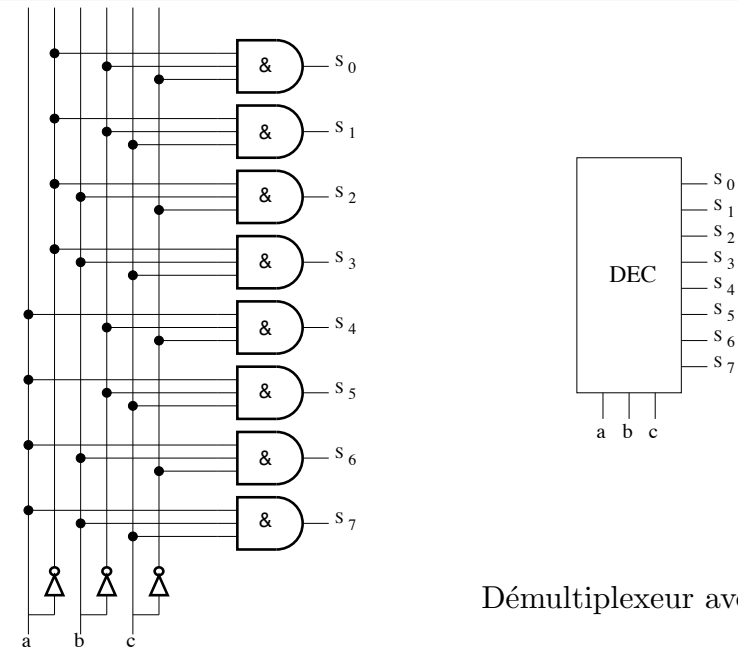
Archi

10/20

Circuits combinatoires

Circuits arithmétiques

Câblage du décodeur



Démultiplexeur avec $E=1$.

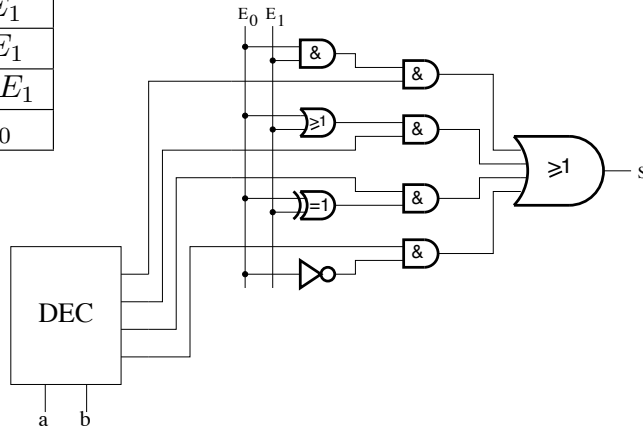
Archi

12/20

Exemple d'utilisation d'un décodeur

Activation de fonction : faire, au choix, l'une des 4 fonctions logiques sur les données E_0 et E_1 . Le choix de la fonction est déterminé par les valeurs de a et b .

a	b	S
0	0	E_0 ET E_1
0	1	E_0 OU E_1
1	0	E_0 XOR E_1
1	1	NON E_0



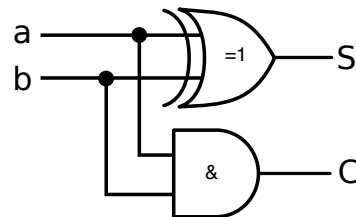
Archi

13/20

1/2 Additionneur

- **Entrées** : les deux bits à additionner a et b
- **Sortie** :
 - la somme $S = a + b$
 - la retenue C
- **Rôle** : Additionner a et b en conservant la retenue.

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Archi

15/20

Circuits arithmétiques

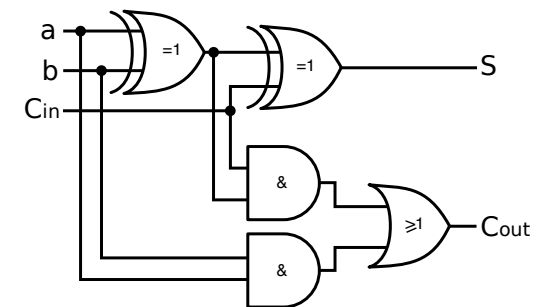
Archi

14/20

Additionneur complet (= 2 demi-additionneurs)

- **Entrées** :
 - les deux bits à additionner a et b
 - la retenue d'entrée C_{in}
- **Sortie** :
 - la somme $S = a + b + C_{in}$
 - la retenue C_{out}
- **Rôle** : Additionner a et b en prenant en compte la retenue d'entrée C_{in} et en conservant la retenue de sortie C_{out} .

a	b	C_{in}	S	C_{out}
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



C_{out} : 1 s'il y a une retenue.

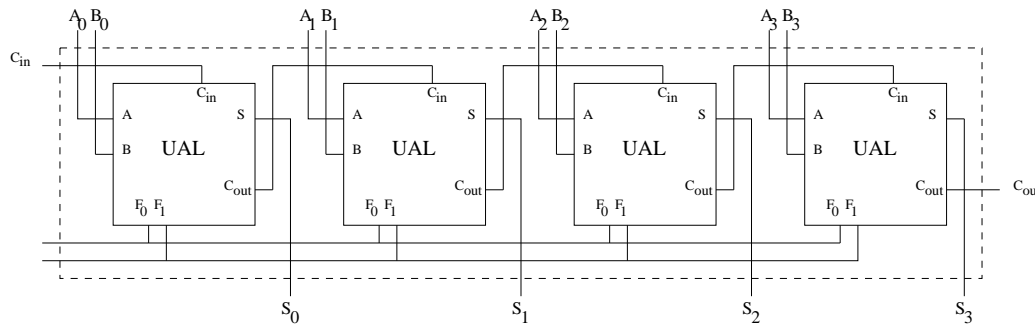
Archi

16/20

UAL 4 bits

Pour 2 bits d'entrée, l'UAL est un circuit qui a peut d'intérêt... Mais, en connectant judicieusement les retenues de n UAL, on obtient une UAL n bits, telle que :

- les opérations logiques sont des opérations bit à bit
- les opérations arithmétiques sont effectuées sur des entiers en complément à 2 sur n bits.



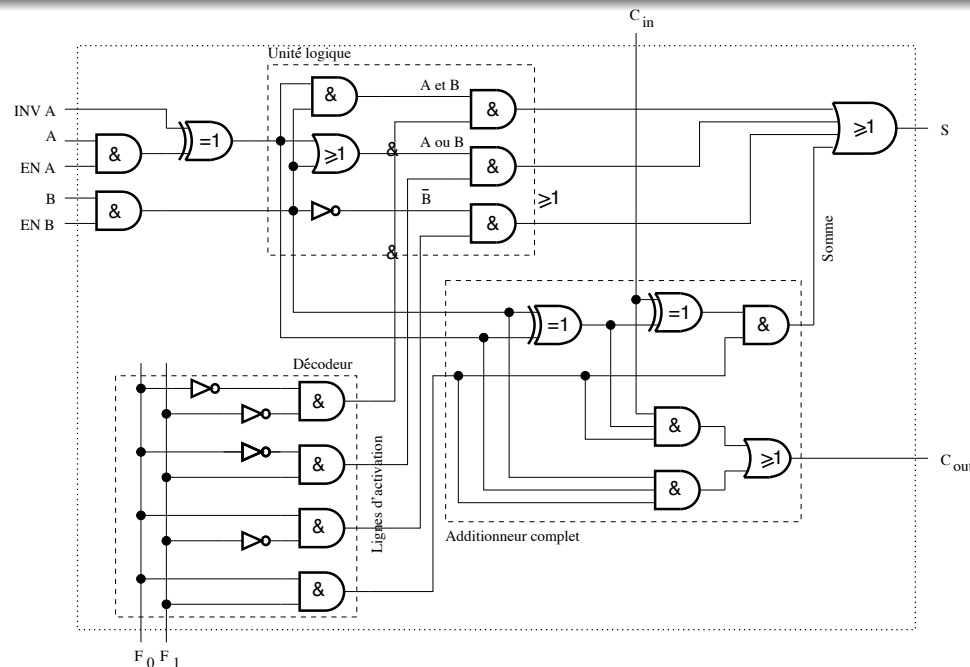
Archi

17/20

Circuits combinatoires

Circuits arithmétiques

Unité Arithmétique et logique



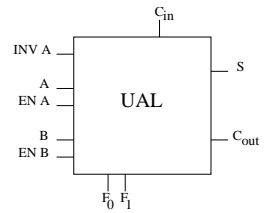
Archi

19/20

Unité Arithmétique et logique

Entrées :

- A et B : les variables (données)
- F_0 et F_1 : bits de choix du signal d'activation
- C_{in} : retenue entrante
- (optionnel) $EN A$ et $EN B$: bits inhibiteurs de A et B
- (optionnel) $INV A$: pour obtenir \bar{A}



Sortie :

- S : résultat de l'opération
- C_{out} : retenue de sortie

- Rôle : Faire l'une des 4 opérations (ou des variantes) :
 $A \text{ ET } B$, $A \text{ OU } B$, \bar{B} , $A + B + C_{in}$,
 en fonction des bits d'activation choisis.

Archi

18/20

Circuits combinatoires

Circuits arithmétiques

UAL - Résumé des fonctions

F_0	F_1	$EN A$	$EN B$	$INV A$	C_{in}	fonction
0	0	1	1	0	0	$A \text{ ET } B$
0	1	1	1	0	0	$A \text{ OU } B$
0	1	0	0	0	0	0
0	1	0	1	0	0	B
0	1	1	0	0	0	A
0	1	1	0	1	0	\bar{A}
1	0	1	1	0	0	\bar{B}
1	1	1	1	0	0	$A + B$
1	1	0	0	0	1	1
1	1	0	0	1	0	-1
1	1	0	1	0	1	$B + 1$
1	1	0	1	1	0	$B - 1$
1	1	1	0	0	1	$A + 1$
1	1	1	0	1	1	$-A$
1	1	1	1	0	1	$A + B + 1$
1	1	1	1	1	1	$B - A$

Archi

20/20