

# Architecture des Ordinateurs

## Partie I : Concepts de Base

### 2. Composants élémentaires

David Simplot  
simplot@fil.univ-lille1.fr



## Objectifs

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

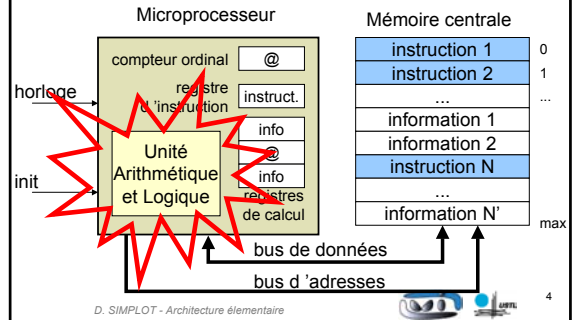


## Au sommaire...

- Circuits arithmétiques**
  - additionneurs, comparateurs
- Circuits combinatoires
  - codeurs, décodeurs, sélecteurs, multiplexeurs, démultiplexeurs
- Bascules, registres & mémoires
- Conclusion

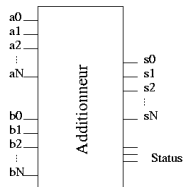


## Additionneur (1/7)



## Additionneurs (2/7)

But :



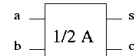
retenue, dépassement de capacité, zéro (carry, overflow, zero)



## Additionneurs (3/7)

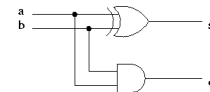
### Demi-additionneur

Demi-additionneur :

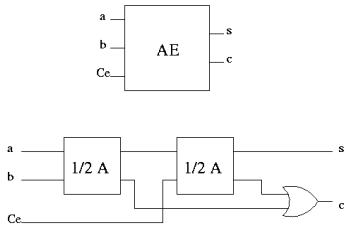


$$c = a \cdot b \quad s = a \oplus b$$

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



## Additionneurs (4/7) Additionneur élémentaire



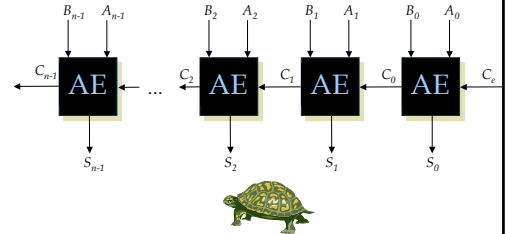
D. SIMPLOT - Architecture élémentaire



7

## Additionneurs (5/7)

- Additionneur n bits :



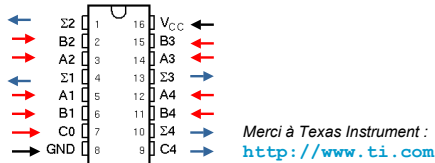
D. SIMPLOT - Architecture élémentaire



8

## Additionneurs (6/7)

- Il existe des circuits intégrés « additionneurs » 1 bit, 2 bits, 4 bits, etc... Le circuit 54/283 est un additionneur 4 bits :



D. SIMPLOT - Architecture élémentaire



9

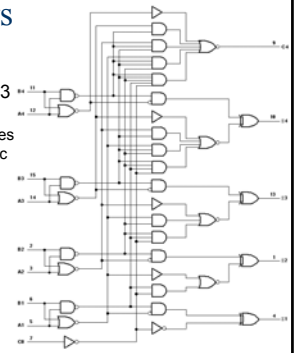
## Additionneurs (7/7)

- Circuit du 54 / 75283

→ ce n'est pas une association d'AE, c'est un additionneur avec anticipation de retenue...



TEXAS INSTRUMENTS



D. SIMPLOT - Architecture élémentaire



10

## Au sommaire...

- **Circuits arithmétiques**
  - additionneurs, **comparateurs**
- Circuits combinatoires
  - codeurs, décodeurs, sélecteurs, multiplexeurs, démultiplexeurs
- Bascules, registres & mémoires
- Conclusion

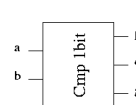
D. SIMPLOT - Architecture élémentaire



11

## Comparateurs (1/4)

- Comparateur 1 bit :



a	b	p	e	g
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

- p = « a plus **p**etit que b » =  $a.b$
- e = « a **é**gal b » =  $a \odot b$  (le XNOR/coïncidence/équivalence)
- g = « a plus **g**rand que b » =  $a.b$

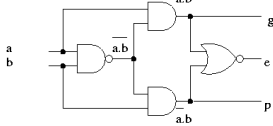
D. SIMPLOT - Architecture élémentaire



12

## Comparateurs (2/4)

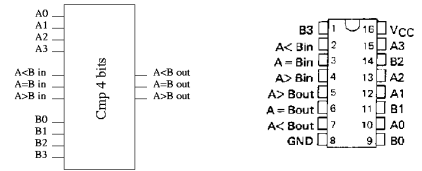
- Circuit du comparateur 1 bit :



- On peut utiliser les comparateurs 1 bit pour faire des comparateurs 2 bits. Puis on peut utiliser les comparateurs 2 bits pour faire des comparateurs 4 bits. Etc...

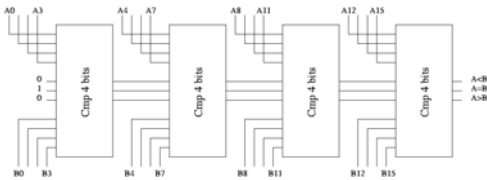
## Comparateurs (3/4)

- On peut également utiliser des circuits comparateurs dits « associatifs ». Par exemple, le SN54 / 7485 est un comparateur 4 bits associatif :



## Comparateurs (4/4)

- Application : construction d'un comparateur 16 bits :



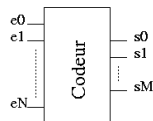
## Au sommaire...

- Circuits arithmétiques
  - ↳ additionneurs, comparateurs
- **Circuits combinatoires**
  - ↳ **codeurs, décodeurs, sélecteurs**, multiplexeurs, démultiplexeurs
- Bascules, registres & mémoires
- Conclusion

## Codeurs (1/2)

- Code une entrée à N entrée où il n'y a qu'un seul 1 en binaire.

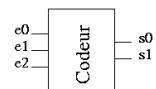
- ↳  $M = \log_2(N+1) - 1$
- ↳ « codeur N+1 vers M+1 »



## Codeurs (2/2)

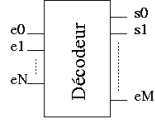
- Codeur 3 vers 2 : N=2, M=1

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 (1/4)

- Décode un mot binaire
  - $2^{N-1} \leq M < 2^N$
  - $s_J = 1$  ssi  $(e_{N-1} \dots e_1 e_0)_2 = J$
  - « décodeur N+1 vers M+1 »
- Exemple :
  - $N=1$   $M=2$
  - « décodeur 2 vers 3 »

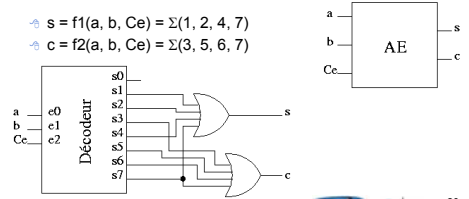


e1	e0	s0	s1	s2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	0	0



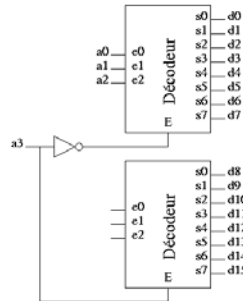
## Décodeurs (2/4)

- Toute fonction logique  $f(x_0, x_1, \dots, x_N)$  peut être réalisée avec un décodeur.
- Exemple l'additionneur élémentaire :
  - $s = f_1(a, b, Ce) = \Sigma(1, 2, 4, 7)$
  - $c = f_2(a, b, Ce) = \Sigma(3, 5, 6, 7)$

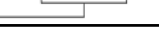


## Décodeurs (3/4)

- On ajoute une entrée « Enabled »
  - si  $E = 0$  alors toutes les sorties sont à 0
  - permet de construire de plus « gros » décodeurs...

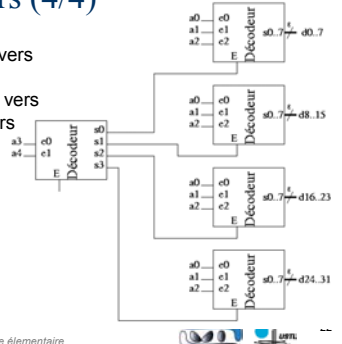


Décodeur 4 vers 16 à partir de décodeur 3 vers 8



## Décodeurs (4/4)

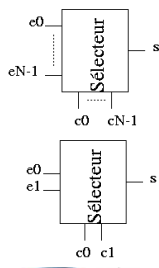
- Décodeur 5 vers 32 avec des décodeurs 3 vers 8 (et un 2 vers 4)...



## Sélecteurs

- Un sélecteur prend N entrées  $(e_0 \dots e_{N-1})$  et N commandes  $(c_0 \dots c_{N-1})$  et copie l'entrée J si  $C_J$  est la seule à être à 1.
  - $S = 1$  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	*



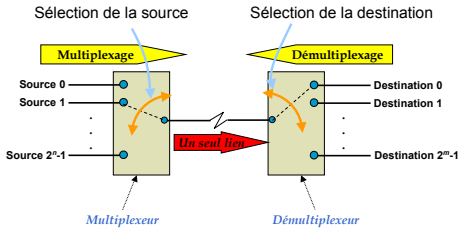
## Au sommaire...

- Circuits arithmétiques
  - additionneurs, comparateurs
- **Circuits combinatoires**
  - codeurs, décodeurs, sélecteurs, **multiplexeurs**, **démultiplexeurs**
- Bascules, registres & mémoires
- Conclusion



# Multiplexeurs et Démultiplexeurs

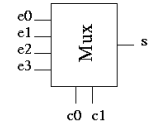
Utilisation :



# Multiplexeurs (1/2)

- Le multiplexeur (**MUX**) sélectionne, à l'aide de  $N$  entrées de commande ( $c0..n-1$ ), **une** des  $2^N$  entrées d'information ( $e0..2^N-1$ ) et la dirige à la sortie.
  - La sortie est égale à 1 ssi l'entrée ( $c0c1...cN-1$ )<sub>2</sub> est vraie.
- Exemple: Multiplexeur 4 à 1.

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



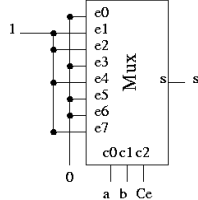
On utilise un **décodeur** et un **sélecteur**...

# Multiplexeurs (2/2)

- Synthèse de fonctions logiques avec un multiplexeur
- Exemple de l'AE... (encore !)

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

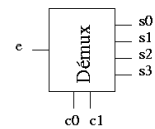
$$s = f1(a, b, Ce) = \Sigma(1, 2, 4, 7)$$



# Démultiplexeurs

- Opérateur inverse du multiplexeur. Il distribue le bit  $e$  reçu en entrée unique vers l'une des  $2^N$  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

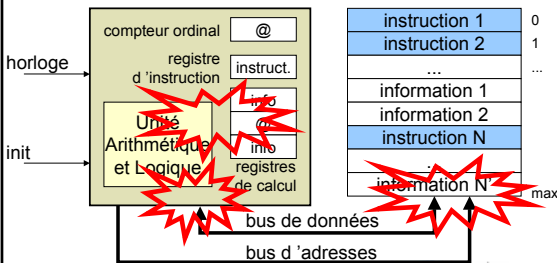


C'est un décodeur avec  $e$  comme « enabled ».

# Multiplexeurs et Démultiplexeurs

Microprocesseur

Mémoire centrale

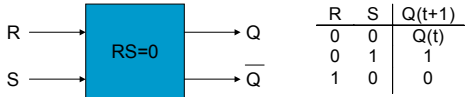


# Au sommaire...

- Circuits arithmétiques
  - additionneurs, comparateurs
- Circuits combinatoires
  - codeurs, décodeurs, sélecteurs, multiplexeurs, démultiplexeur
- Bascules, registres & mémoires**
- Conclusion

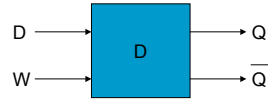
## Bascules, registres et mémoires (1/3)

- Bascule (bistable/flip-flop)
  - ↳ = moyen de mémoriser de l'information
- Différents type de bascules
  - ↳ RS=0, RS=0, RST, D, JK, JKT, RSME, JKME...
- Ex. Bascule RS=0



## Bascules, registres et mémoires (2/3)

- Ex. Bascule D



↳ La donnée stockée copie D si W=1 sinon, l'état de la bascule est inchangée

## Bascules, registres et mémoires (3/3)

- Mémoires persistantes
  - ↳ ROM, EEPROM, Flash
  - ↳ Read-Only Memory
- Mémoires volatiles
  - ↳ RAM (Random Acces Memory)
  - ↳ Deux types statique ou dynamique
    - SRAM et DRAM

## Au sommaire...

- Circuits arithmétiques
  - ↳ additionneurs, comparateurs
- Circuits combinatoires
  - ↳ codeurs, décodeurs, sélecteurs, multiplexeurs, démultiplexeur
- Bascules, registres & mémoires
- Conclusion

## En conclusion

