

LICENCE  
EEA

2 - Rappels

# SYSTEMES NUMERIQUES & PROCESSEURS EMBARQUES

# Plan

- **Généralités**
- **Systèmes Combinatoires**
  - Multiplexeurs, Codeurs, Décodeurs
  - Incrémenteurs
  - Comparateurs
  - Nombres Signés - Complément à 2
  - Addition
- **Systèmes Séquentiels**
  - Bascules
  - Registres
  - Compteurs

C2

2

# Remembrance of things past...

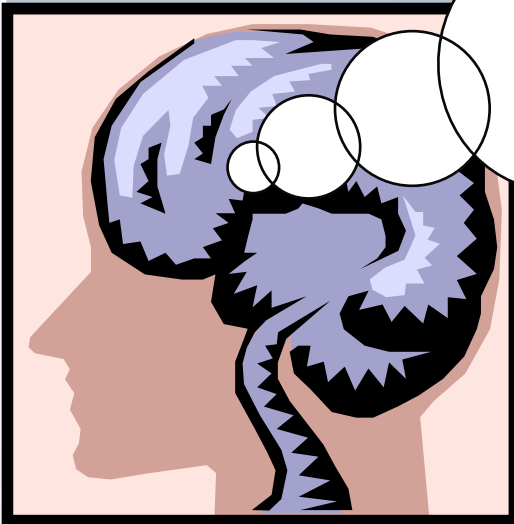
ELECTRONIQUE NUMERIQUE

Combinatoire

Séquentiel

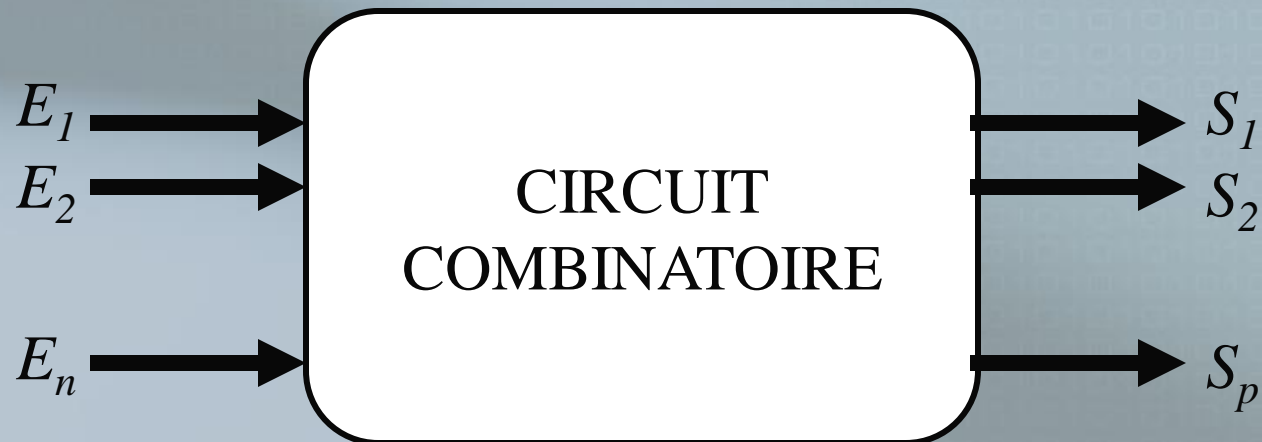
C2

3



# Électronique Combinatoire

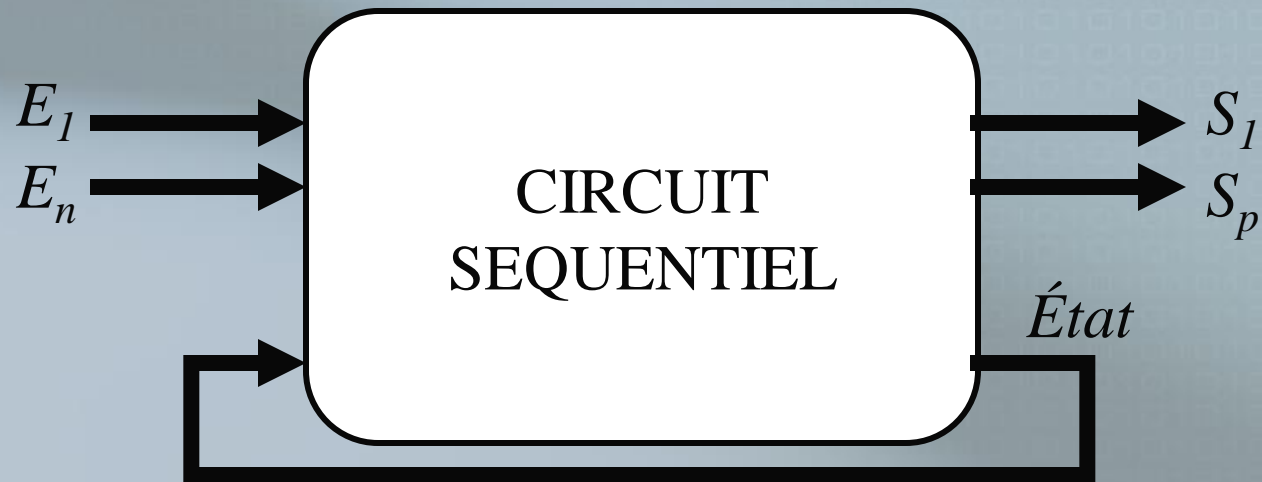
- Un circuit est dit combinatoire si la valeur de ses sorties ne dépend que de la valeur actuelle des entrées



- Une même combinaison en entrée DONNERA TOUJOURS la même valeur en sortie (après un temps de latence)

# Électronique Séquentielle

- Un circuit est dit séquentiel si la valeur de ses sorties dépend de la combinaison de ses entrées ET de l'état du système



- Une même combinaison en entrée NE DONNERA PAS TOUJOURS la même valeur en sortie

# Rememberance of things past...

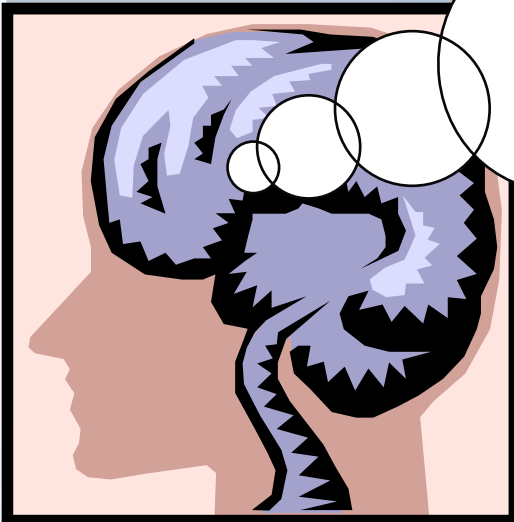
ELECTRONIQUE NUMERIQUE

Combinatoire

Séquentiel

C2

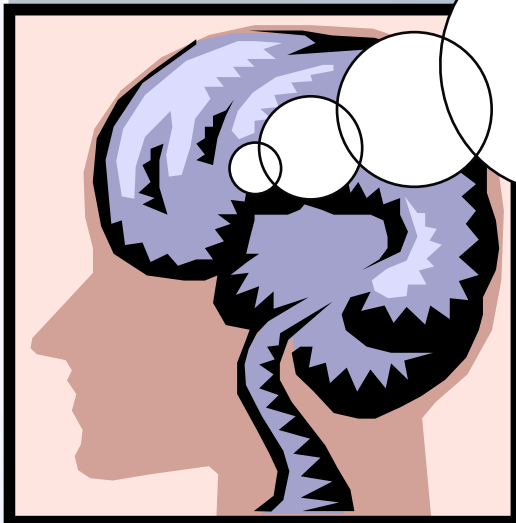
6



# Rappels de Combinatoire

C2

7



Portes Logiques

Additionneurs  
Soustracteurs

Multiplexeurs  
Codeurs/Décodeurs

Comparateurs

Buffers Tristate

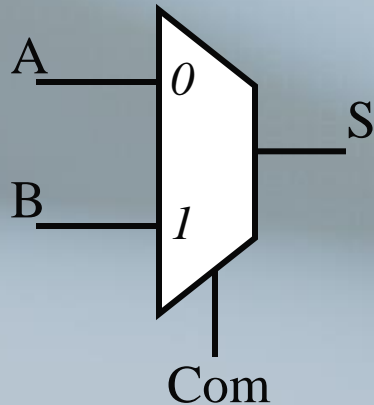
Nombres Signés

Incrémenteurs

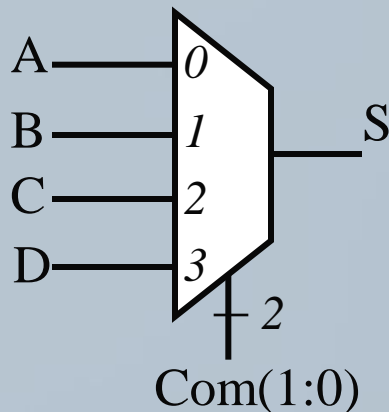
# Multiplexeurs

## ■ Dispositif d'aiguillage

- Une des entrées passe en sortie



$$S = \overline{Com}.A + Com.B$$



$$S = \overline{Com(1)}.\overline{Com(0)}.A + \\ \overline{Com(1)}.Com(0).B + \\ Com(1).\overline{Com(0)}.C + \\ Com(1).Com(0).D$$

C2

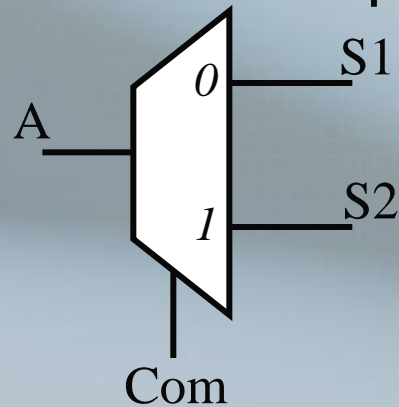
8



# Démultiplexeurs

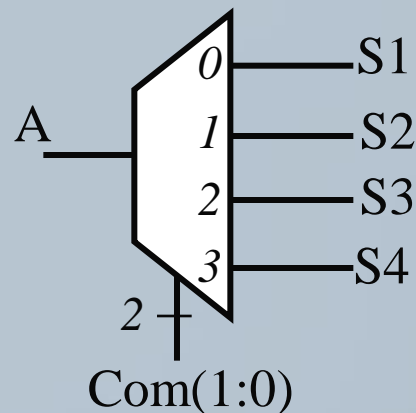
## ■ Dispositif d'aiguillage

- L'entrée passe sur une des sorties



$$S1 = \overline{Com}.A$$

$$S2 = Com.A$$



$$S1 = \overline{Com(1)}.\overline{Com(0)}.A$$

$$S2 = \overline{Com(1)}.Com(0).A$$

$$S3 = Com(1).\overline{Com(0)}.A$$

$$S4 = Com(1).Com(0).A$$

# Codeurs

- $2^N$  entrées  $\rightarrow$  N sorties

E3	E2	E1	E0	S1	S0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$S1 = \overline{E3}.E2.\overline{E1}.\overline{E0} + E3.\overline{E2}.\overline{E1}.\overline{E0} = \overline{E1}.\overline{E0}.(E3 \oplus E2)$$

$$S0 = \overline{E3}.\overline{E2}.\overline{E1}.E0 + \overline{E3}.\overline{E2}.E1.\overline{E0} = \overline{E2}.\overline{E0}.(E3 \oplus E1)$$

C2

10

# Décodeurs

- N entrées  $\rightarrow 2^N$  sorties



E1	E0	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$S3 = E1.E0$$

$$S2 = E1.\overline{E0}$$

$$S1 = \overline{E1}.E0$$

$$S0 = \overline{E1}.\overline{E0}$$

C2

11

# Détecteur d'égalité

- Pour deux mots de 1 bit

E1	E0	Egal
0	0	1
0	1	0
1	0	0
1	1	1



$$Egal = \overline{E1 \oplus E0}$$

*NB: Détecteur d'inégalité = XOR*

- Pour deux mots de n bits
  - Égalité si les bits sont égaux deux à deux

$$Egal = \overline{A_0 \oplus B_0} \cdot \overline{A_1 \oplus B_1} \dots \overline{A_{n-1} \oplus B_{n-1}}$$

# Détecteur de Supériorité

- Exemple sur deux mots A et B de 4 bits

3725

1832

2576

2799

1620

4821

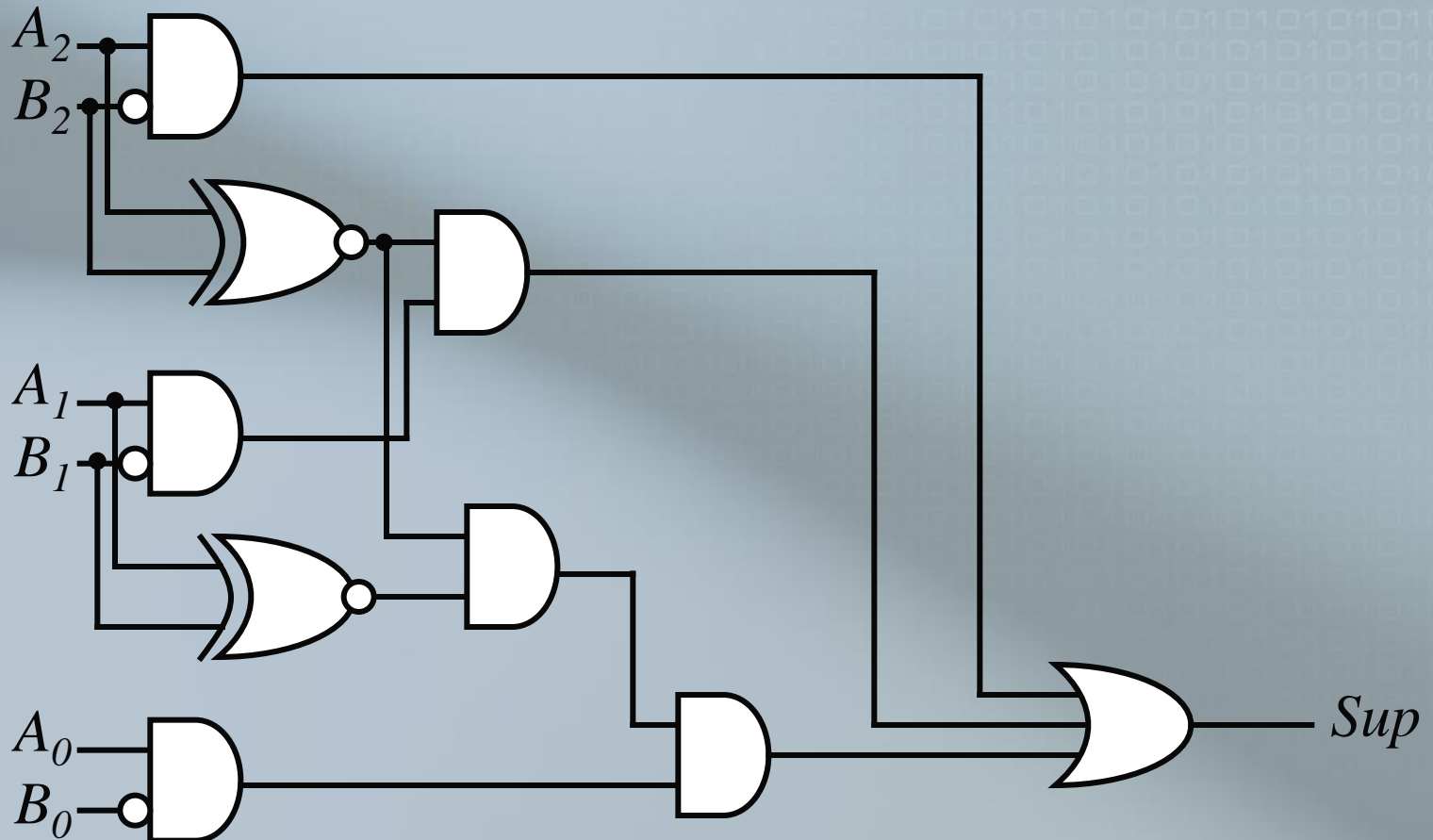


C2

13

# Détecteur de Supériorité

## ■ Schéma Portes (pour 3 bits)



C2

14

# Détecteur d'infériorité

- Raisonnement analogue à supériorité

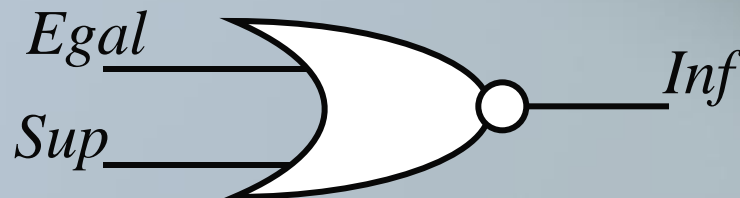
**OU**

- $A < B$  si

- A n'est pas égal à B

**ET**

- A n'est pas supérieur à B



C2

15

# Incrémenteur

E3	E2	E1	E0	S3	S2	S1	S0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

$$S0 = \overline{E0}$$

$$S1 = E1 \oplus E0$$

$$S2 = E2 \oplus (E1.E0)$$

$$S3 = E3 \oplus (E2.E1.E0)$$

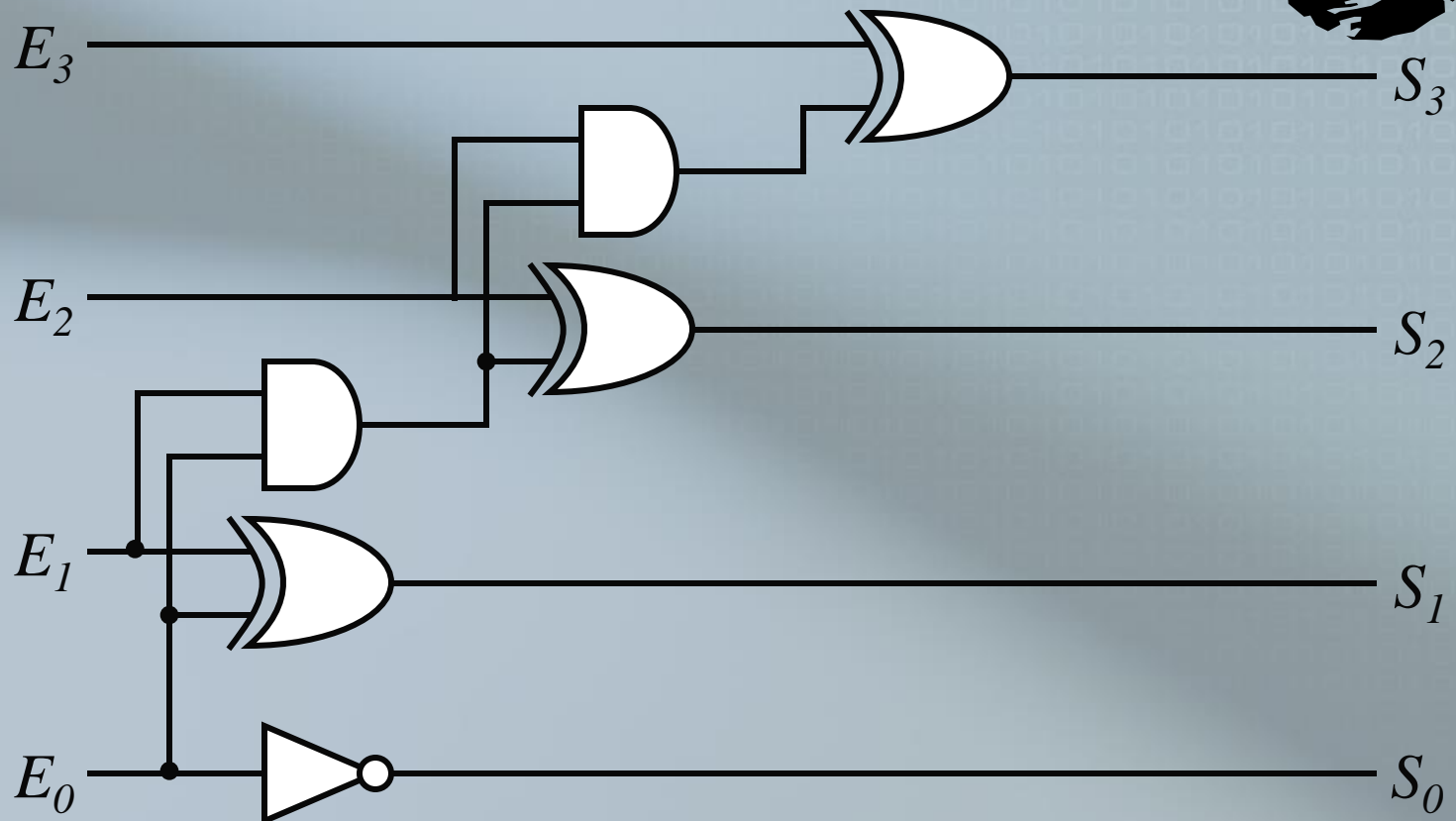
C2

16



# Incrémenteur

## ■ Schéma Portes



C2

17

# Nombres Signés

- Codage complément à 2
  - Introduction d'un bit de signe
    - Bit de poids fort (*MSB: Most Significant Bit*)
    - 0: Nombre positif
    - 1: Nombre négatif

- Exemple

- +3: 0 0 1 1

$$= -0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

- -3: 1 1 0 1

$$= -1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

C2

18

# Complément à 2

## ■ Passage d'un nombre à son complément à 2

### ■ Méthode 1:

- Nombre à complémenter
- Complément à 1
- Ajouter 1
- Complément à 2

$$\begin{array}{r} 0\ 1\ 0\ 1\ 0\ 0\ 0\ (+40) \\ 1\ 0\ 1\ 0\ 1\ 1\ 1 \\ + \qquad \qquad \qquad 1 \\ \hline 1\ 0\ 1\ 1\ 0\ 0\ 0\ (-40) \end{array}$$

### ■ Méthode 2:

- A partir du poids faible (LSB: Least Significant Bit)
- Laisser inchangés tous les bits jusqu'au 1er '1' inclus
- Complémenter tous les bits restants

$$\begin{array}{r} 0\ 1\ 0\ 1\ 0\ 0\ 0\ (+40) \\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ (-40) \end{array}$$

# Nombres Signés

- Intérêt de la complémentation à 2
  - Le codage des nombres positifs est identique
  - Unicité du 0
  - Utilisation des opérateurs non signés
    - Pour l'addition
    - Pour la soustraction

C2

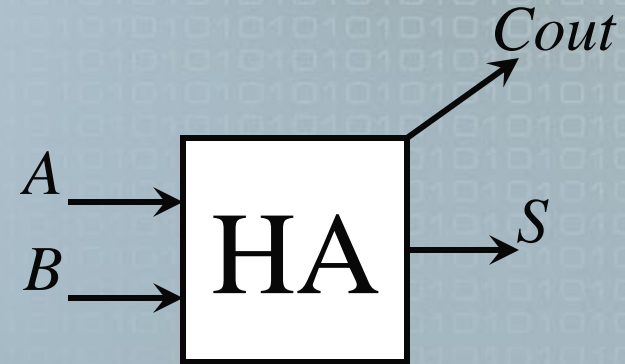
20



# Demi-Additionneur

- Half-Adder (HA)

- 2 entrées (A,B)
- 1 sortie somme (S)
- 1 sortie retenue (Cout)



A	B	Cout	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

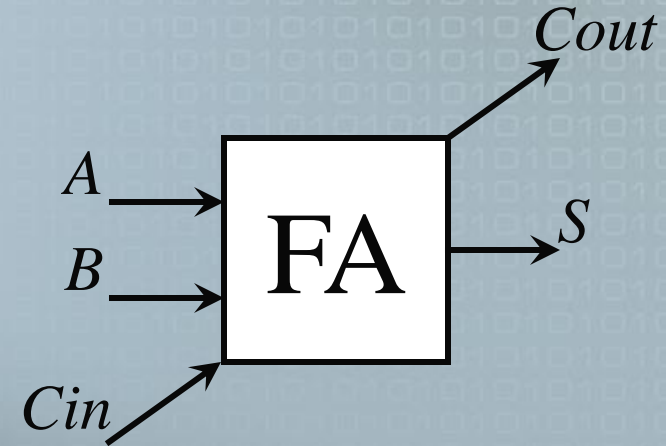
$$S = A \oplus B$$

$$Cout = A.B$$

# Additionneur Complet

## ■ Full-Adder (FA)

- 2 entrées (A,B)
- 1 entrée retenue (Cin)
- 1 sortie somme (S)
- 1 sortie retenue (Cout)



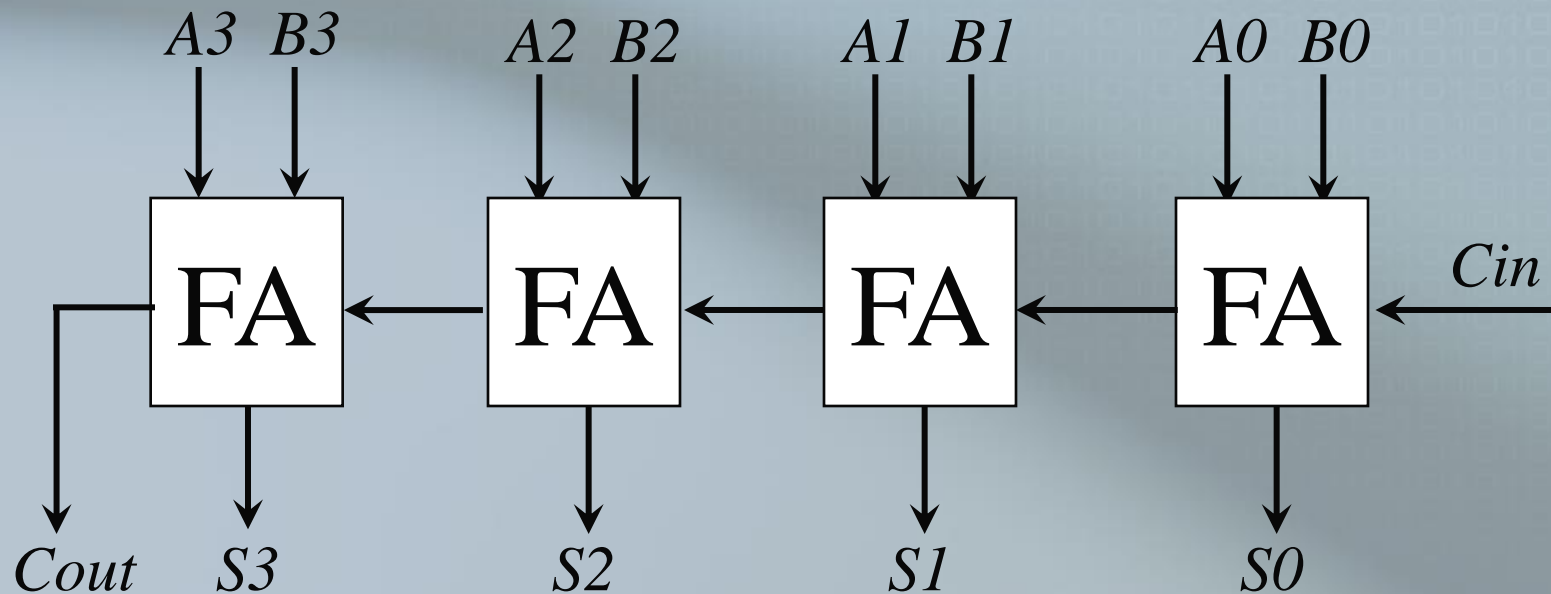
Cin	A	B	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A \oplus B \oplus Cin$$

$$Cout = A.B + Cin. A \oplus B$$

# Additionneur n bits

- Mise en Cascade de Full Adders
  - Exemple Additionneur 4 bits



C2

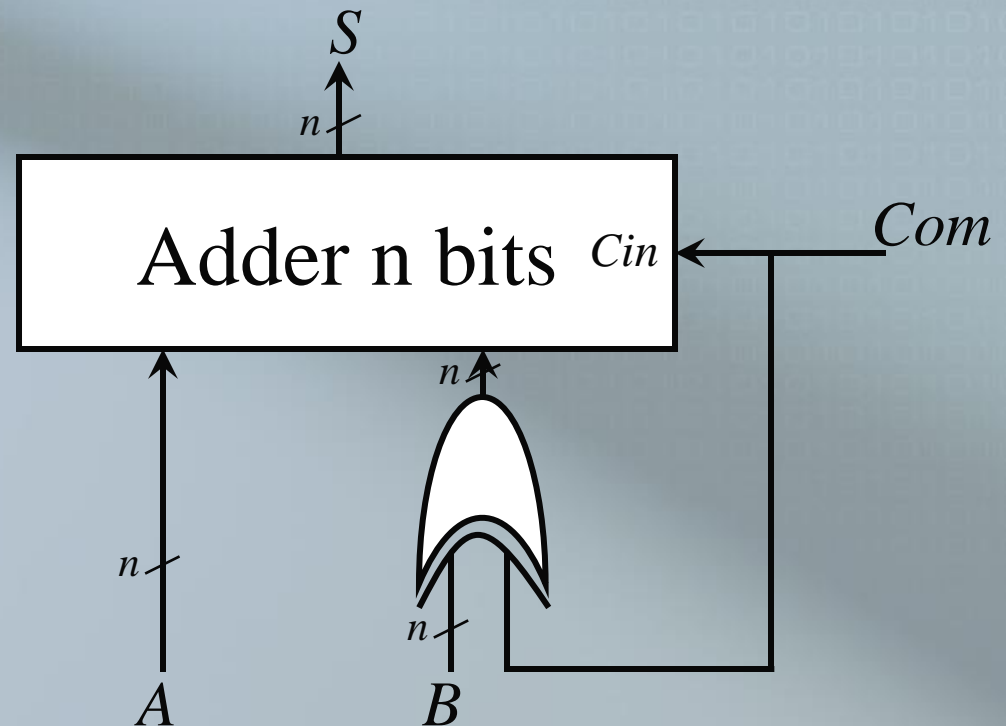
23

# Additionneur / Soustracteur

## ■ Utilisation du complément à 2

- $A - B = A + (-B) = A + \overline{B} + 1$
- Utilisation de l'additionneur n bits

Com	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



C2

24

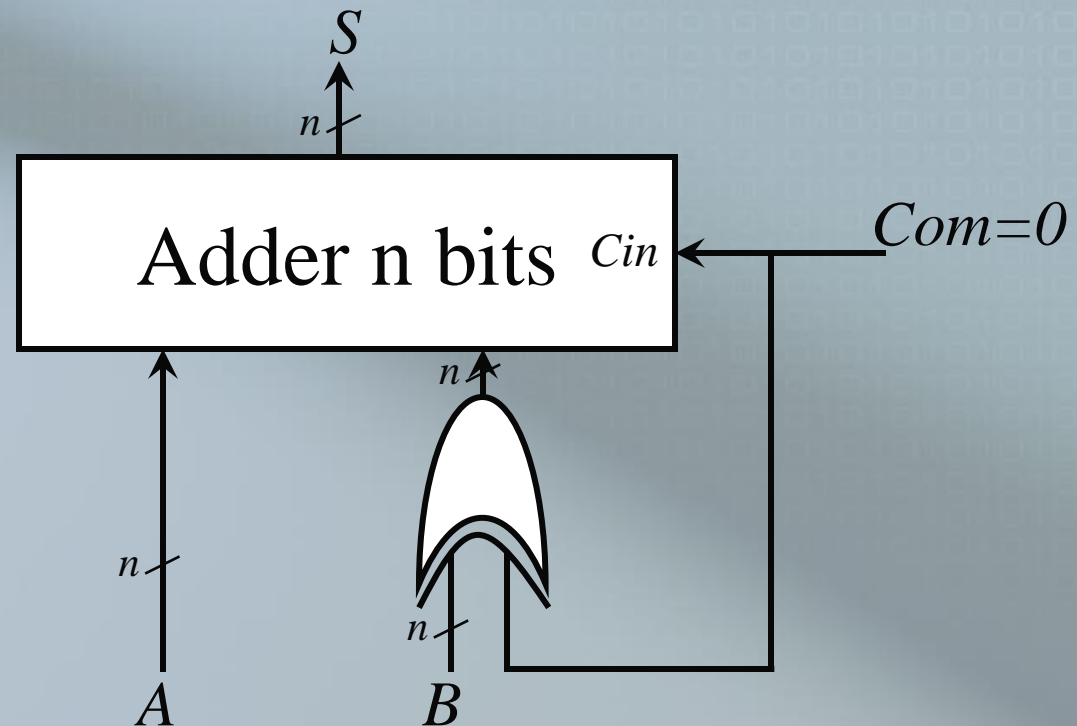


# Additionneur / Soustracteur

- $Com=0$ 
  - $B \text{ xor } 0 = B$
  - $S = A + B + 0$

➤ Addition

Com	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



C2

25

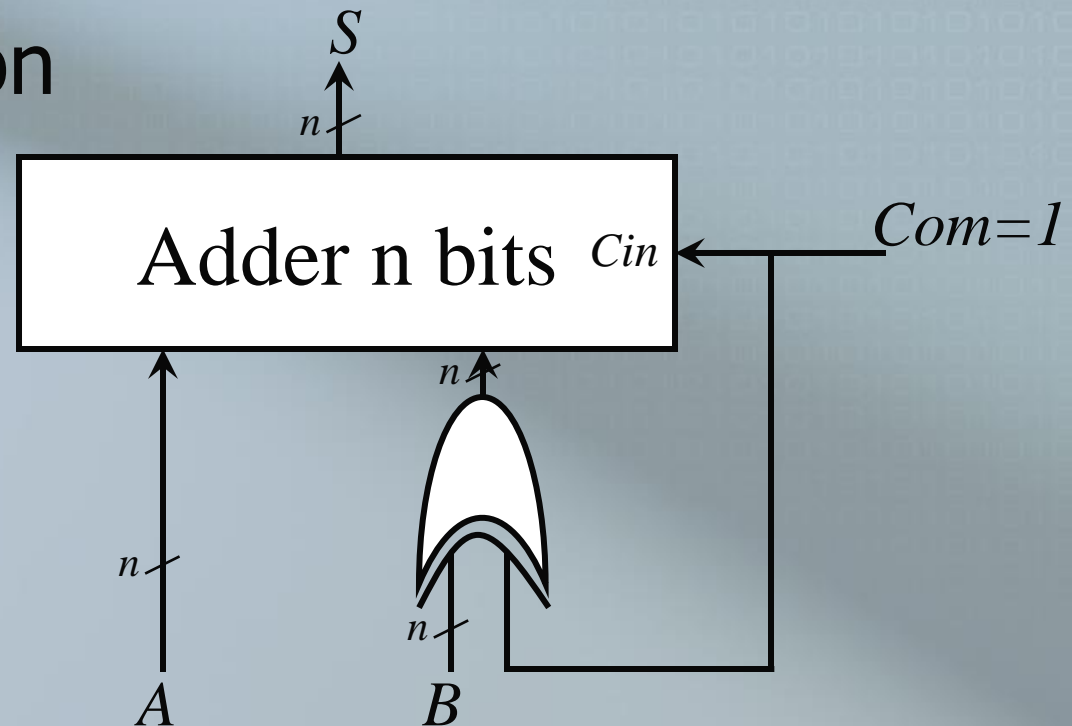
# Additionneur / Soustracteur

- $Com = 1$ 
  - $B \text{ xor } 1 = \overline{B}$
  - $S = A + \overline{B} + 1$
- Soustraction



C2

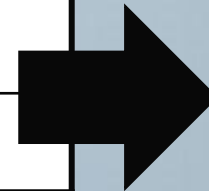
Com	B	XOR
0	0	0
0	1	1
<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>0</b>



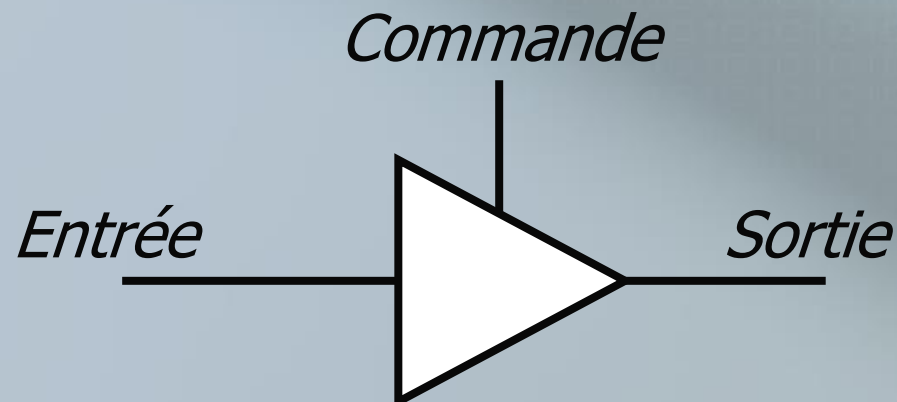
26

# Buffer Tristate

Commande	Entrée	Sortie
0	0	Z
0	1	Z
1	0	0
1	1	1



*Haute impédance  
(idem interrupteur ouvert)*

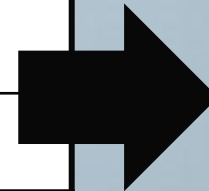


C2

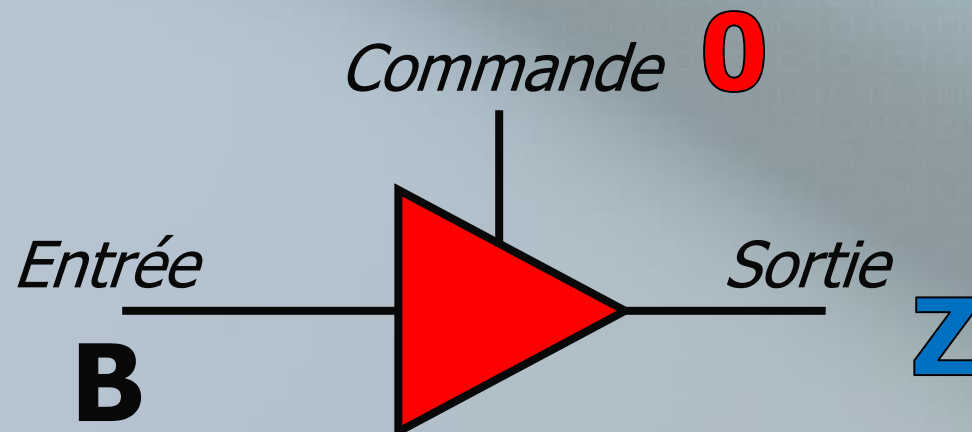
27

# Buffer Tristate

Commande	Entrée	Sortie
0	0	Z
0	1	Z
1	0	0
1	1	1



*Haute impédance*  
(idem interrupteur ouvert)

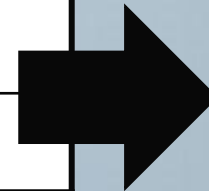


C2

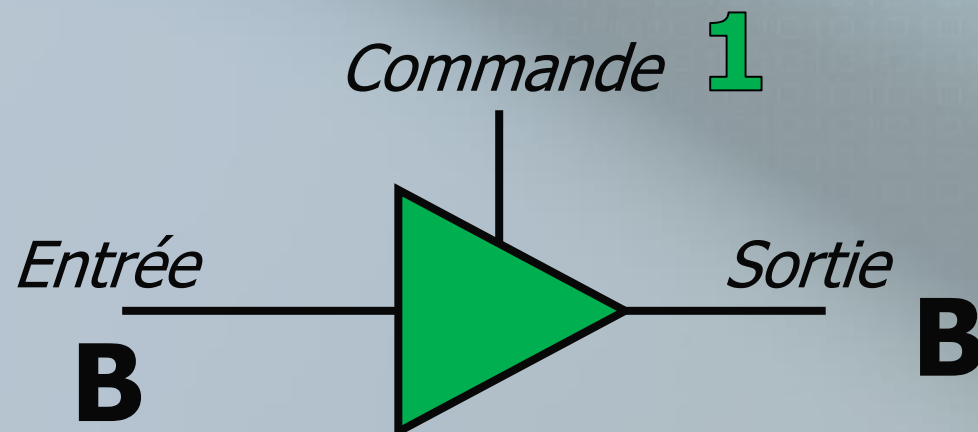
28

# Buffer Tristate

Commande	Entrée	Sortie
0	0	Z
0	1	Z
1	0	0
1	1	1



*Haute impédance  
(idem interrupteur ouvert)*



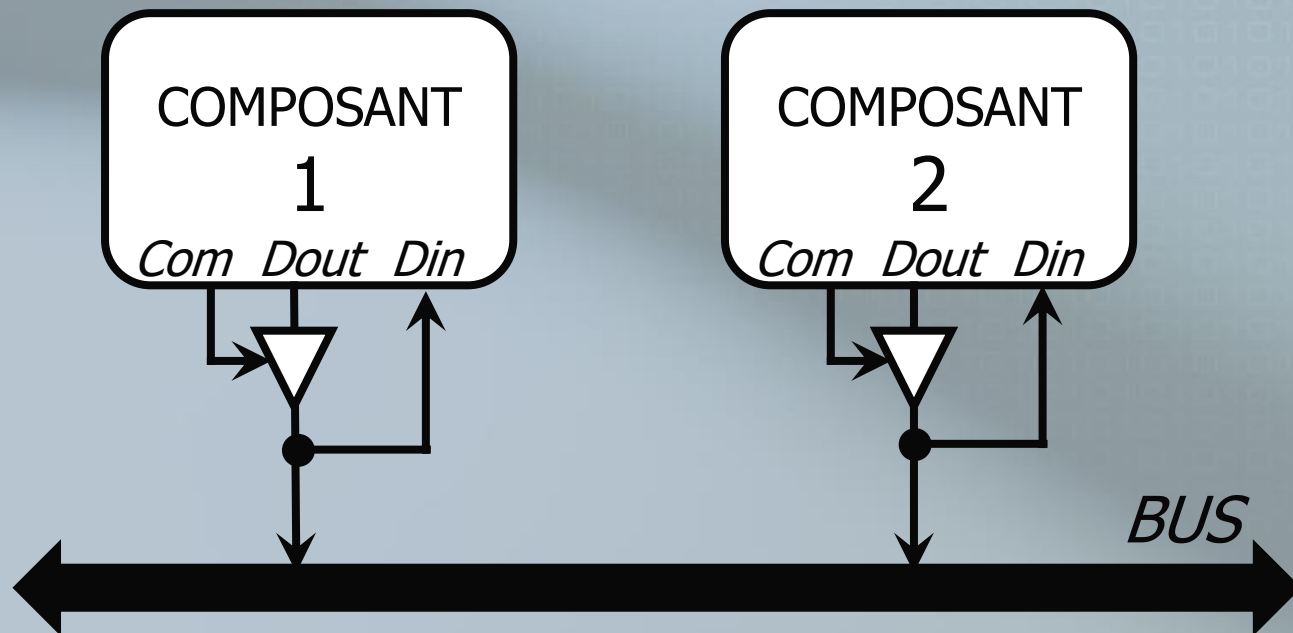
C2

29

# Buffer Tristate

## ■ Utilisation

- Accès à une ligne partagée bidirectionnelle  
(Bus d'un processeur)



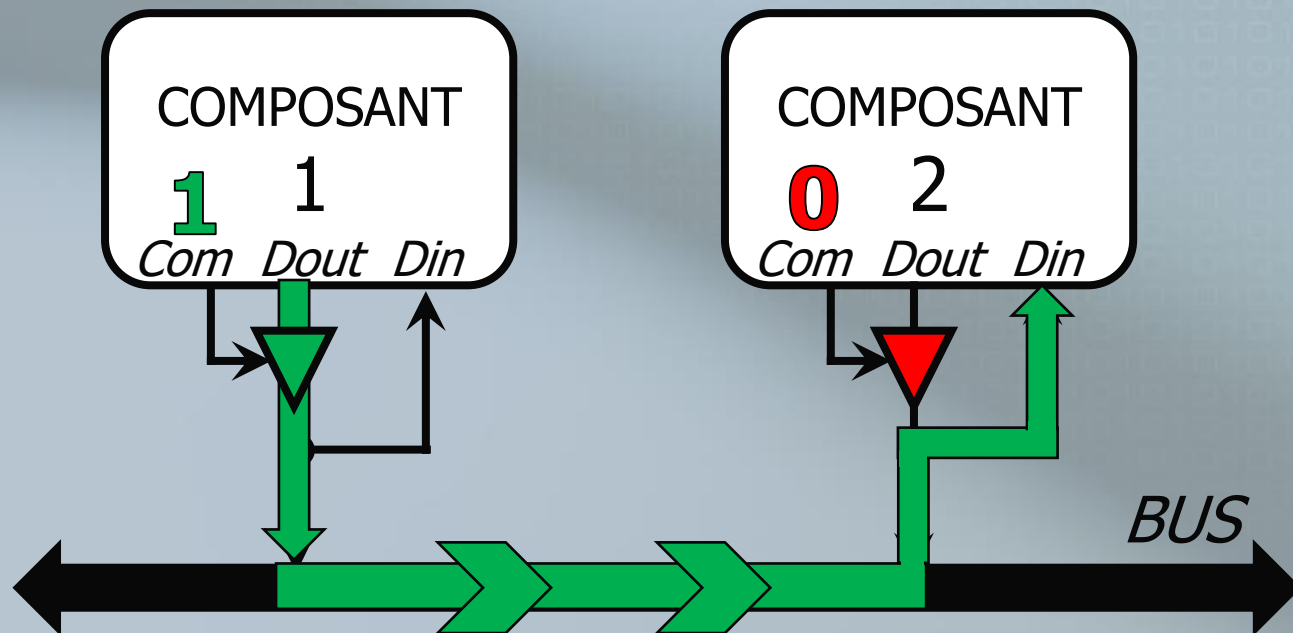
C2

30

# Buffer Tristate

## ■ Utilisation

- Accès à une ligne partagée bidirectionnelle (Bus d'un processeur)



- Communication Composant 1 → Composant 2

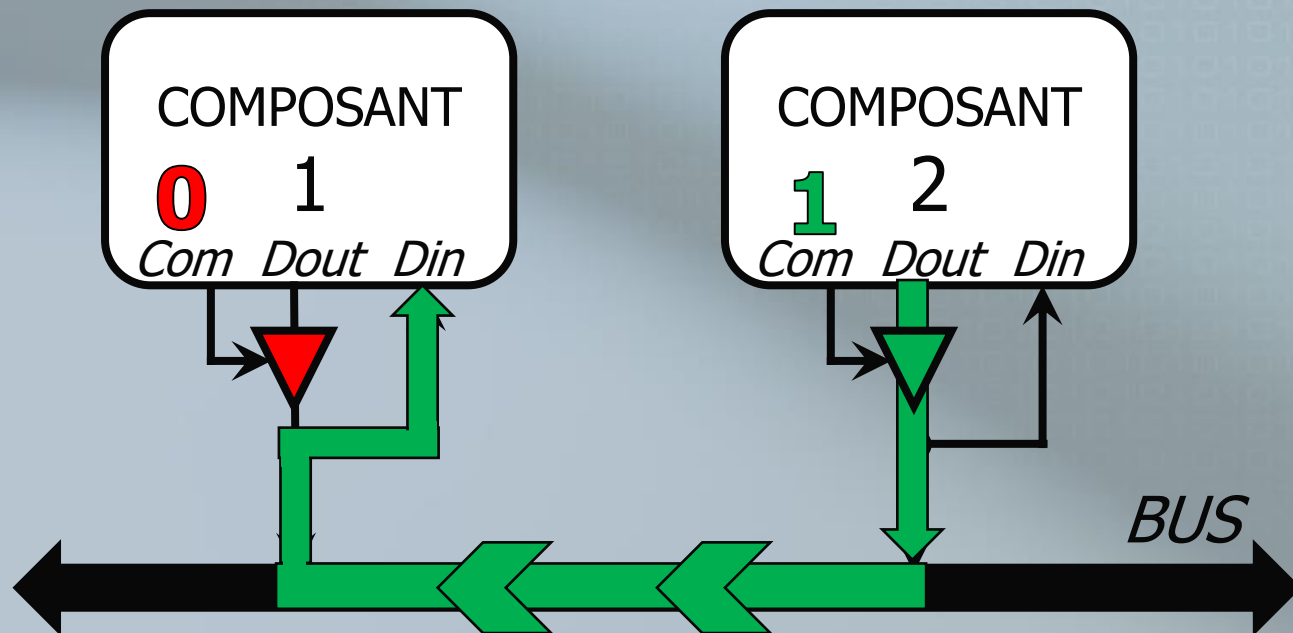
C2

31

# Buffer Tristate

## ■ Utilisation

- Accès à une ligne partagée bidirectionnelle (Bus d'un processeur)



- Communication Composant 2 → Composant 1

C2

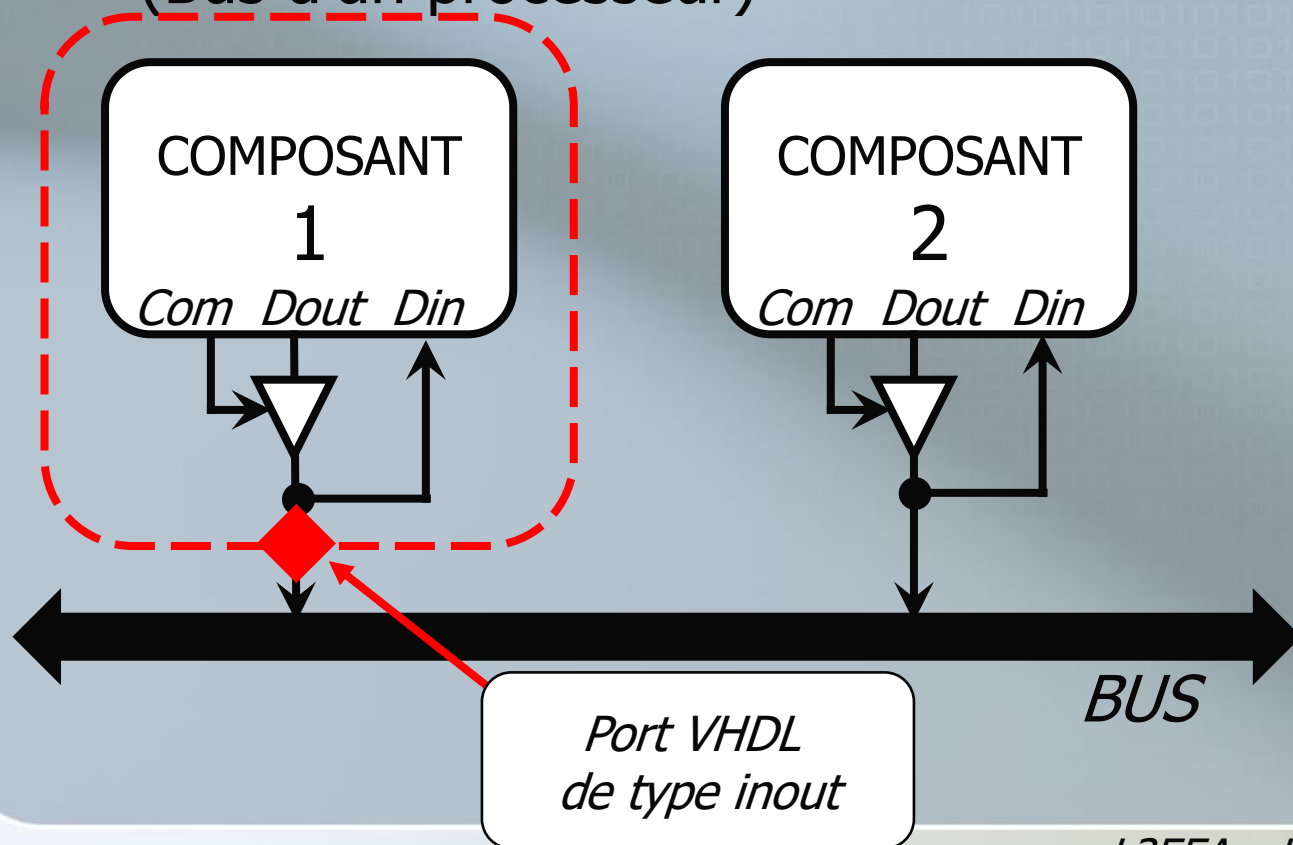
32



# Buffer Tristate

## ■ Utilisation

- Accès à une ligne partagée bidirectionnelle  
(Bus d'un processeur)



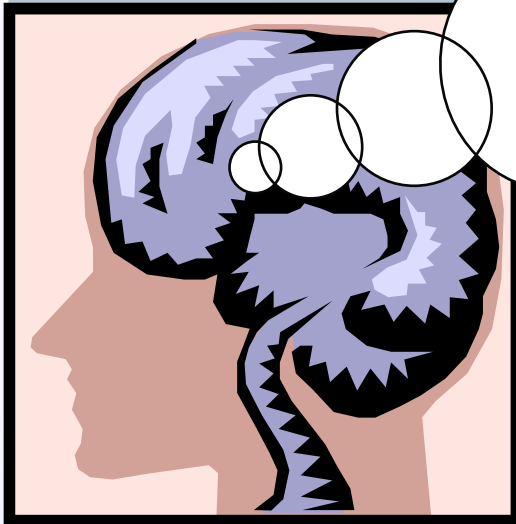
C2

33

# Rappels de Séquentiel

C2

34



Registres

Bascules  
Latches

Machines  
à états

Compteurs

# Bascules / Latches

- Éléments de Mémorisation

- Asynchrone

- Bascule JK
    - Bascule RS

- Synchrone sur niveau

- Latch D

- Synchrone sur front

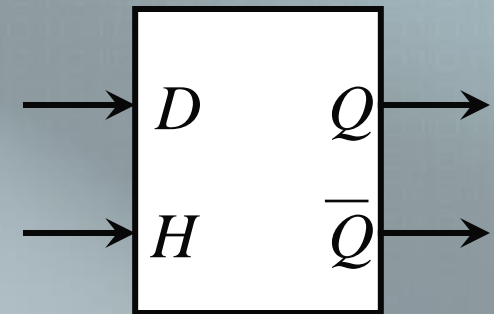
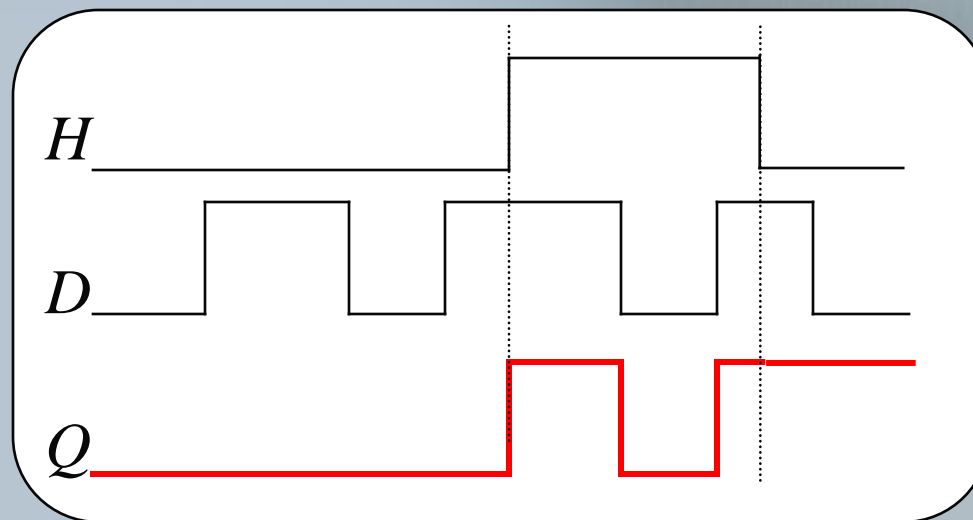
- Bascule D
    - Bascule T

C2

35

# Latch D

- Mémorisation sur Niveau
  - En fonction du signal d'horloge
    - $H=0 \rightarrow$  Verrouillage (Mémorisation):  $Q^+ = Q$
    - $H=1 \rightarrow$  Acquisition:  $Q^+ = D$

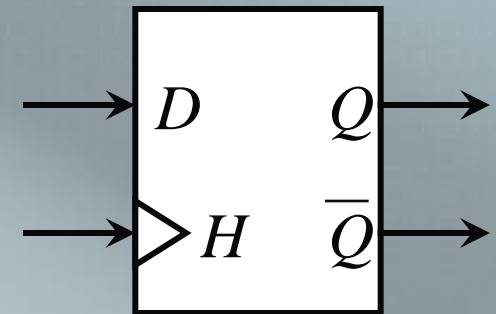
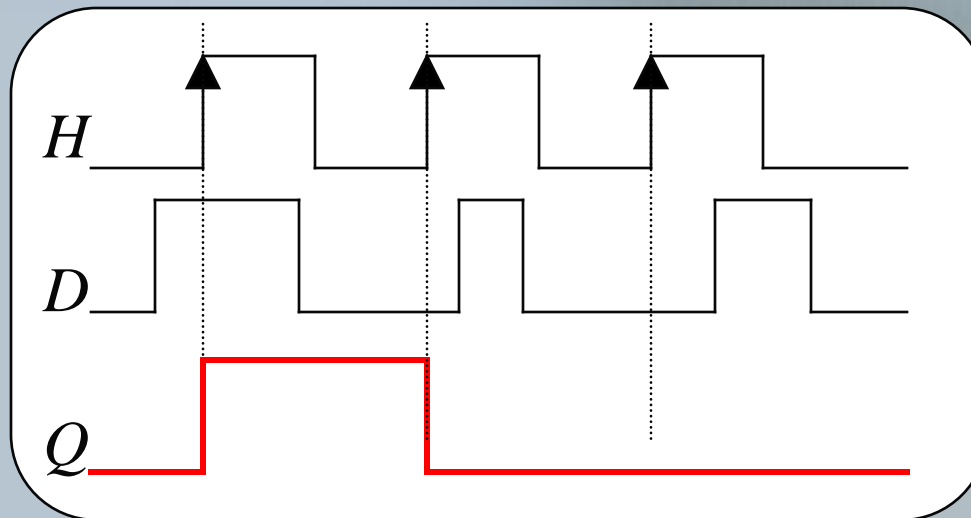


C2

36

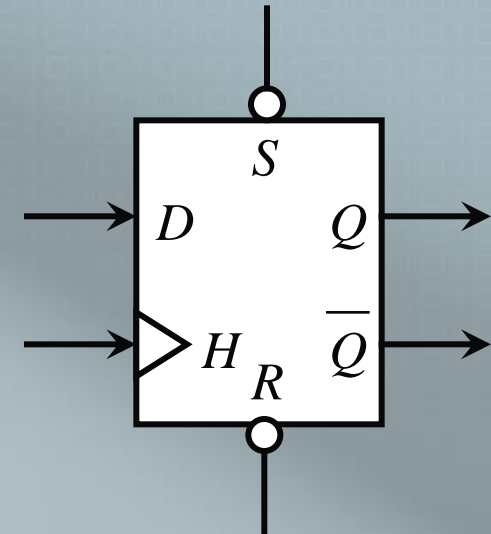
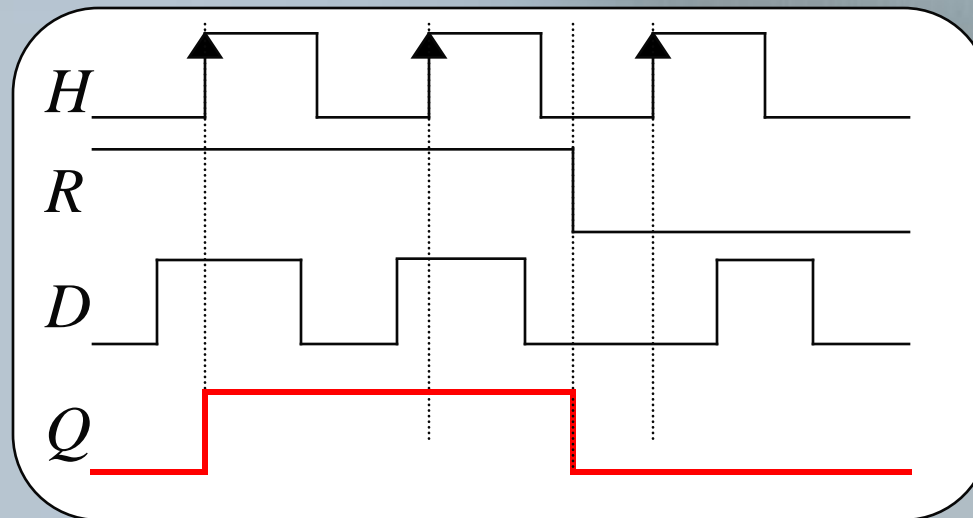
# Bascule D

- Mémorisation sur Front
  - En fonction du signal d'horloge
    - $H = \uparrow \rightarrow$  Acquisition:  $Q^+ = D$
    - $H = 0, 1, \downarrow \rightarrow$  Mémorisation:  $Q^+ = Q$



# Entrées de Forçage

- Initialisation ASYNCHRONE de la Bascule (Latch)
  - Entrée de Set -> Mise à 1
  - Entrée de Reset -> Mise à 0
- Intérêt: Permettent de démarrer un dispositif sans avoir de valeurs indéterminées dans les bascules



C2

38

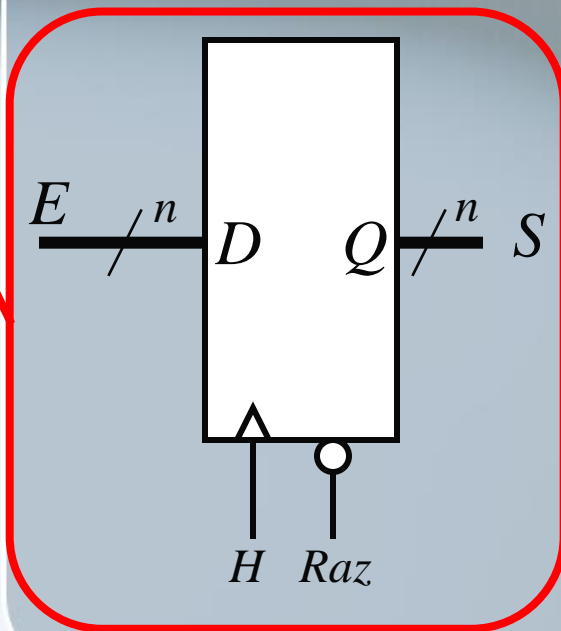
# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Chargement parallèle

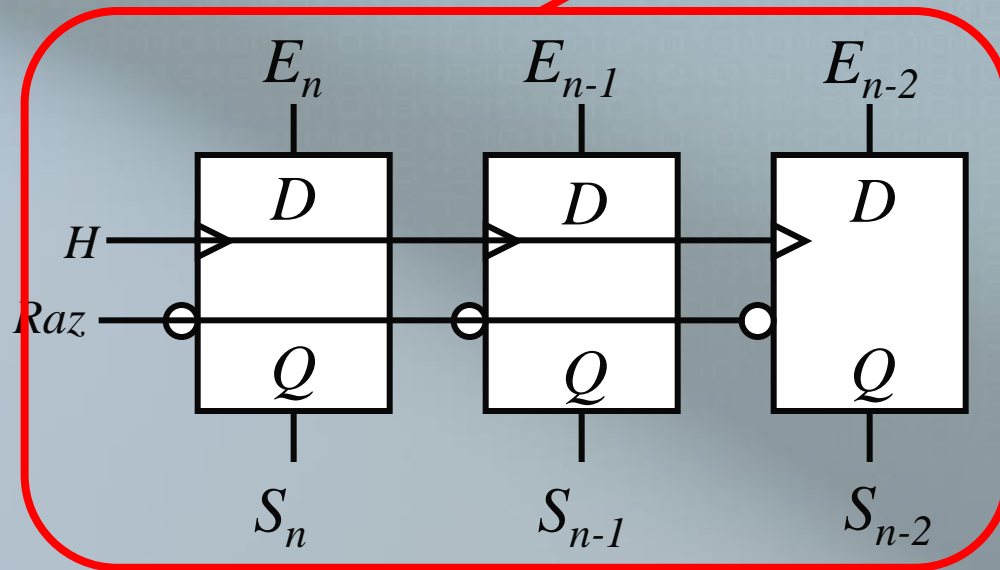
*Schéma  
Bloc*

C2

39

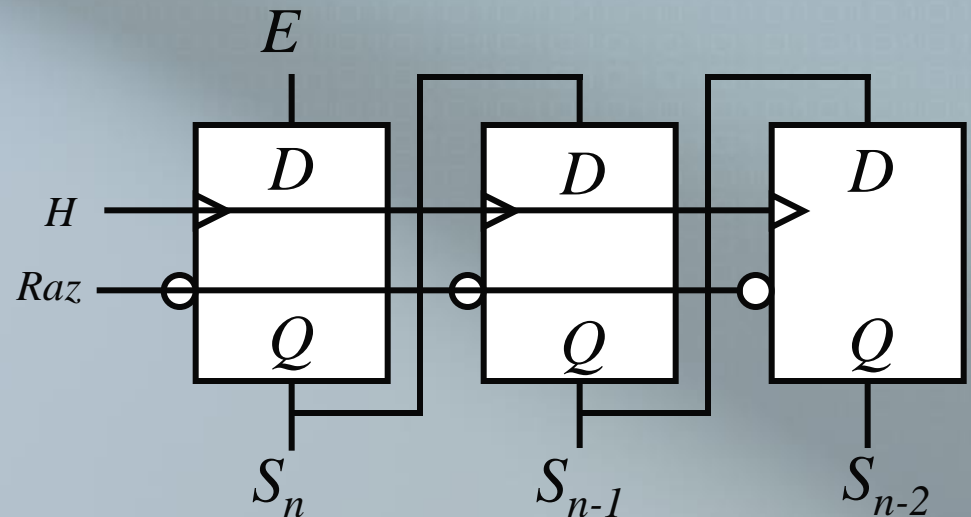
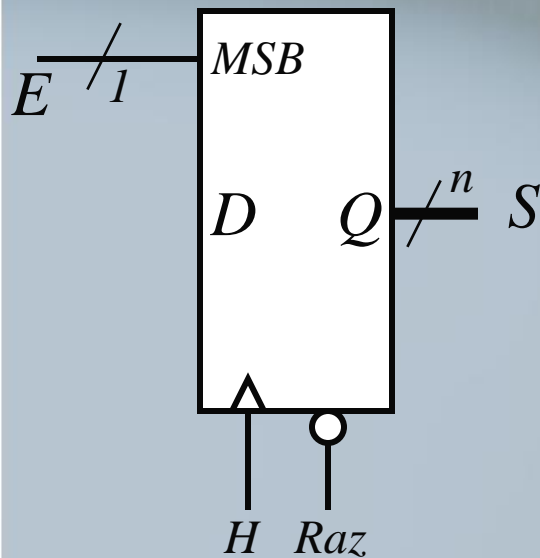


*Schéma  
Portes*



# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Décalage à Droite  
(Chargement Série par le MSB)



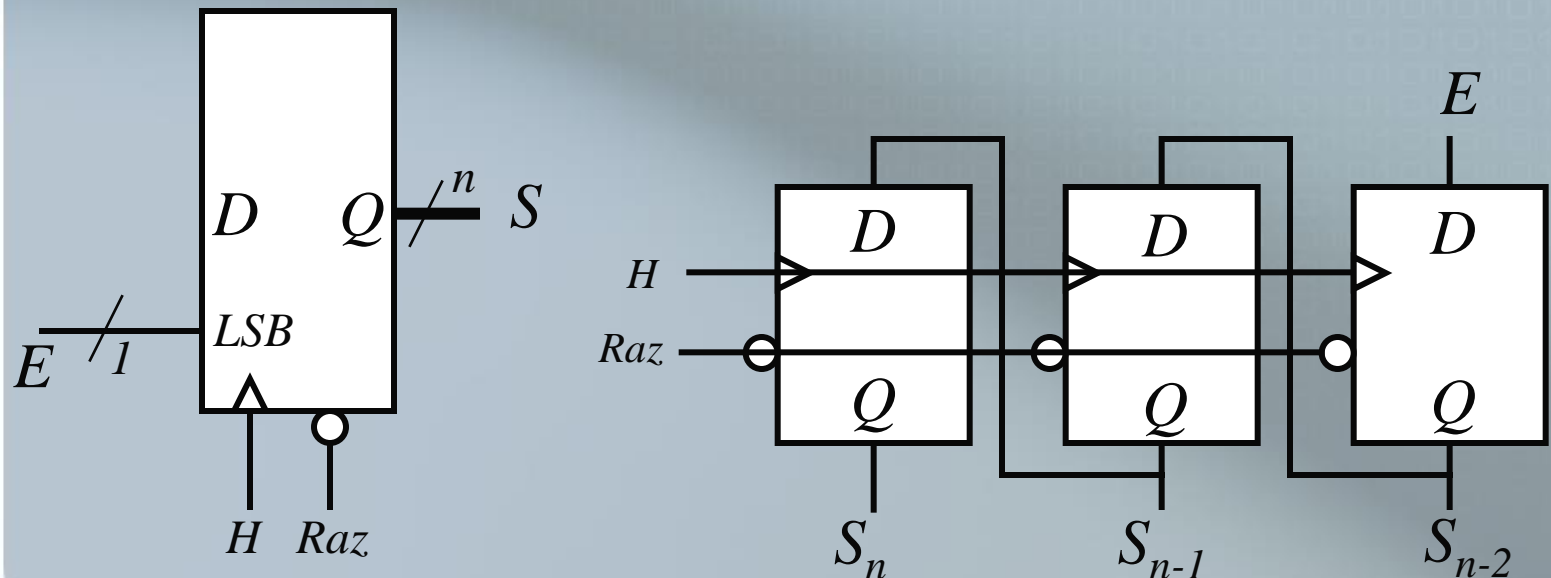
C2

40



# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Décalage à Gauche  
(Chargement Série par le LSB)

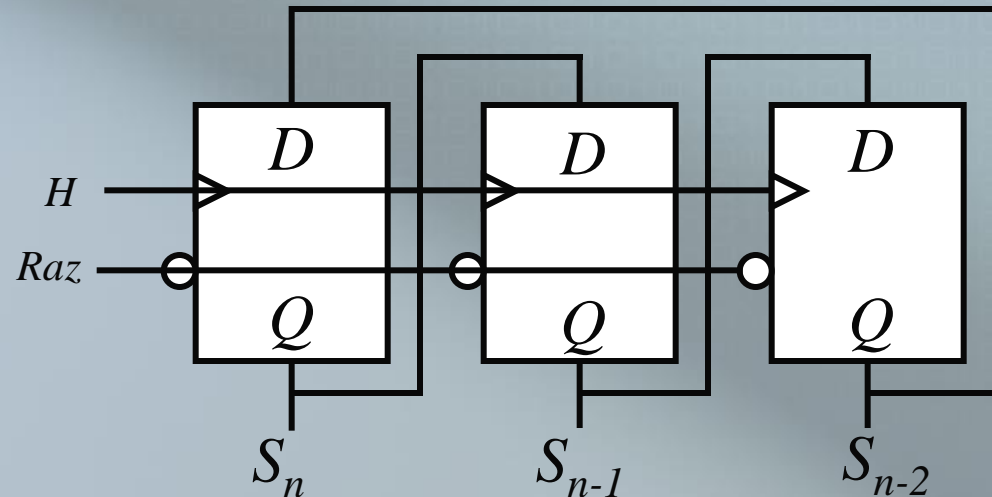
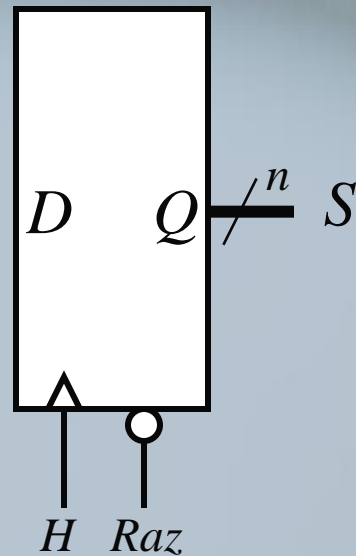


C2

41

# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Rotation à Droite (Rebouclage sur le MSB)

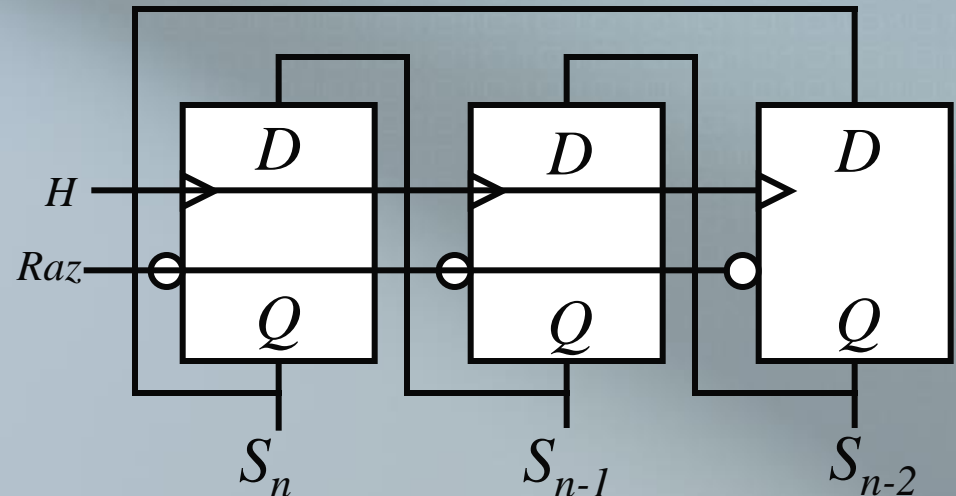
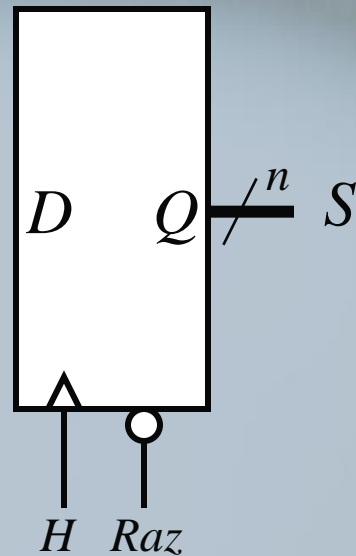


C2

42

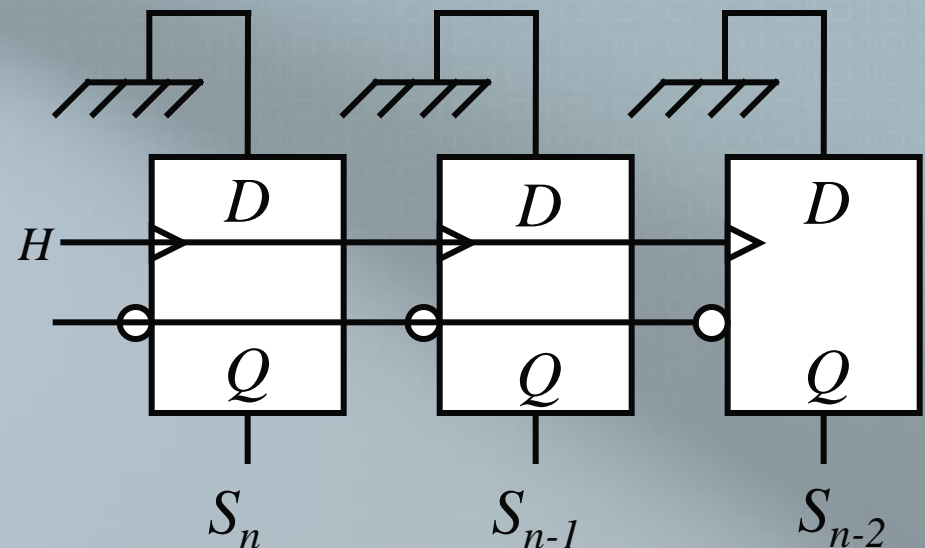
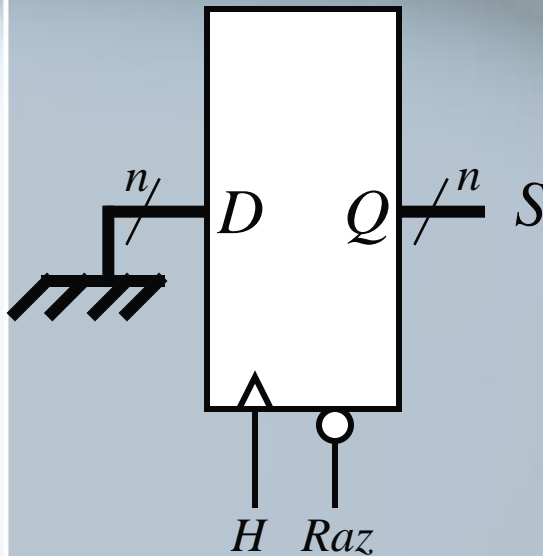
# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Rotation à Gauche (Rebouclage sur le LSB)



# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction RAZ Synchrone

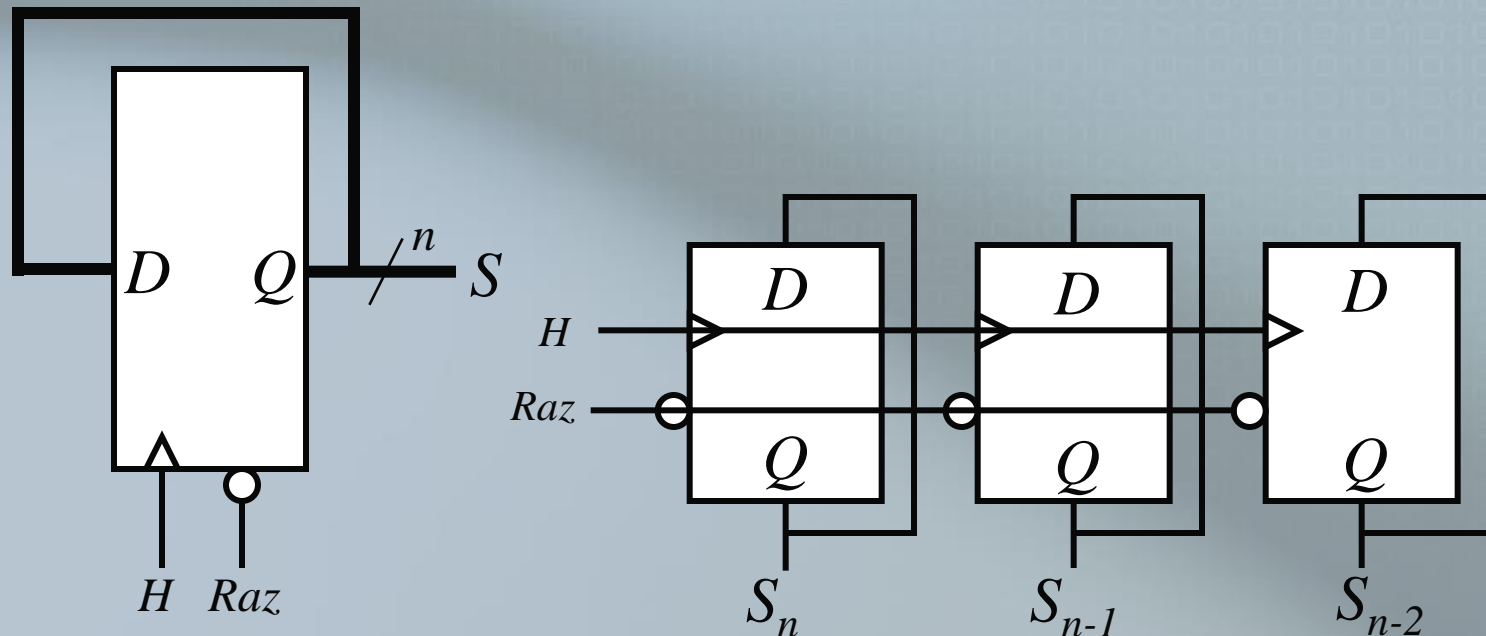


C2

44

# Registres

- Association de Bascules
- Réalisant la même fonction au même instant
  - Fonction Mémorisation (Maintien de la valeur)

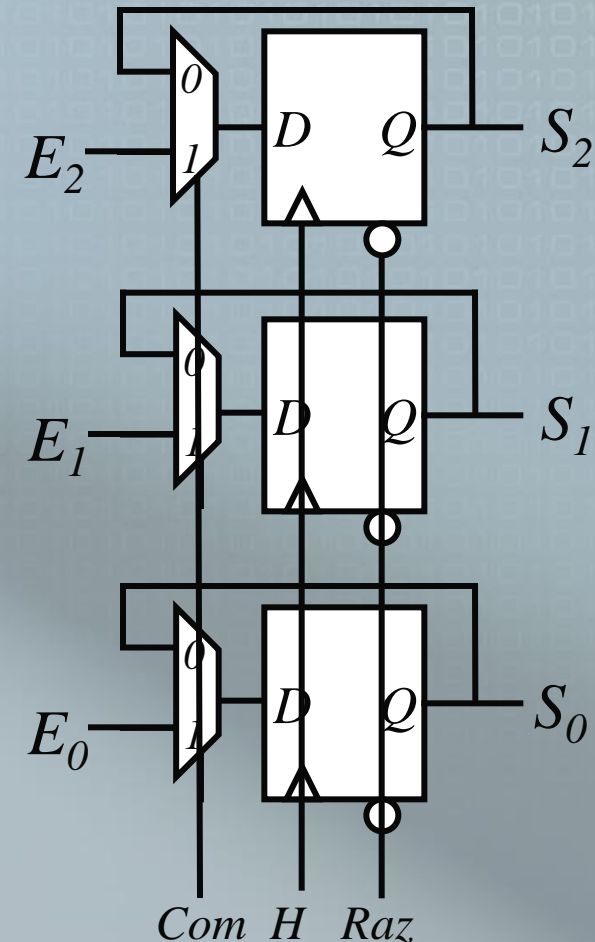
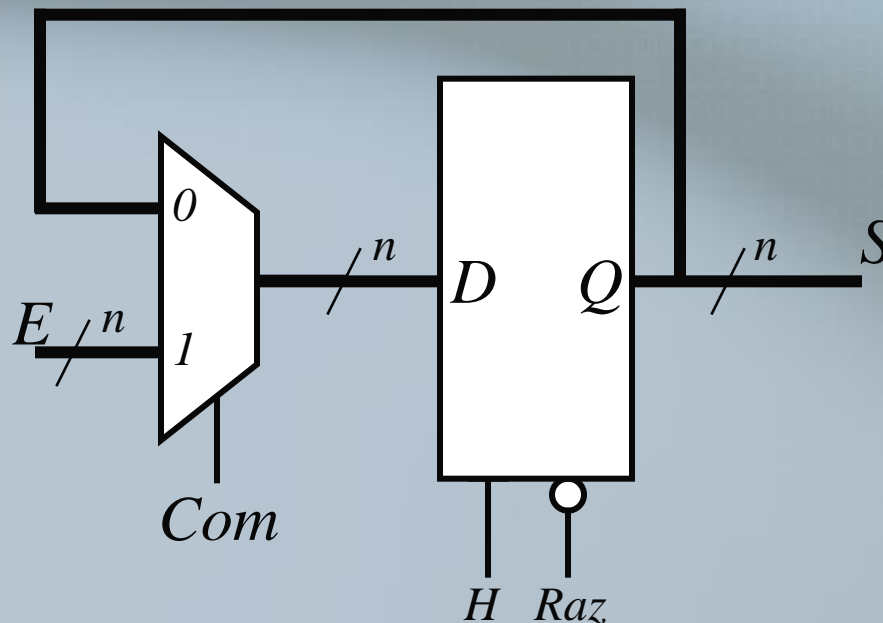


C2

45

# Registres Multifonctions

- Ajout d'un multiplexeur devant chaque bascule pour sélectionner la fonction désirée
  - Ex: chargement parallèle + mémorisation



C2

46

# Registres Multifonctions

- Ajout d'un multiplexeur devant chaque bascule pour sélectionner la fonction désirée
  - Ex: chargement parallèle + mémorisation

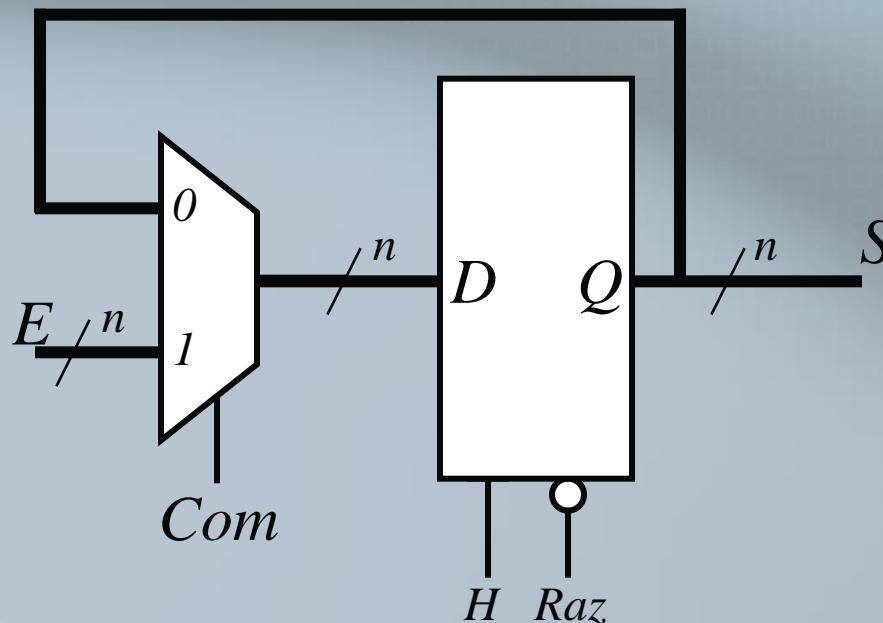


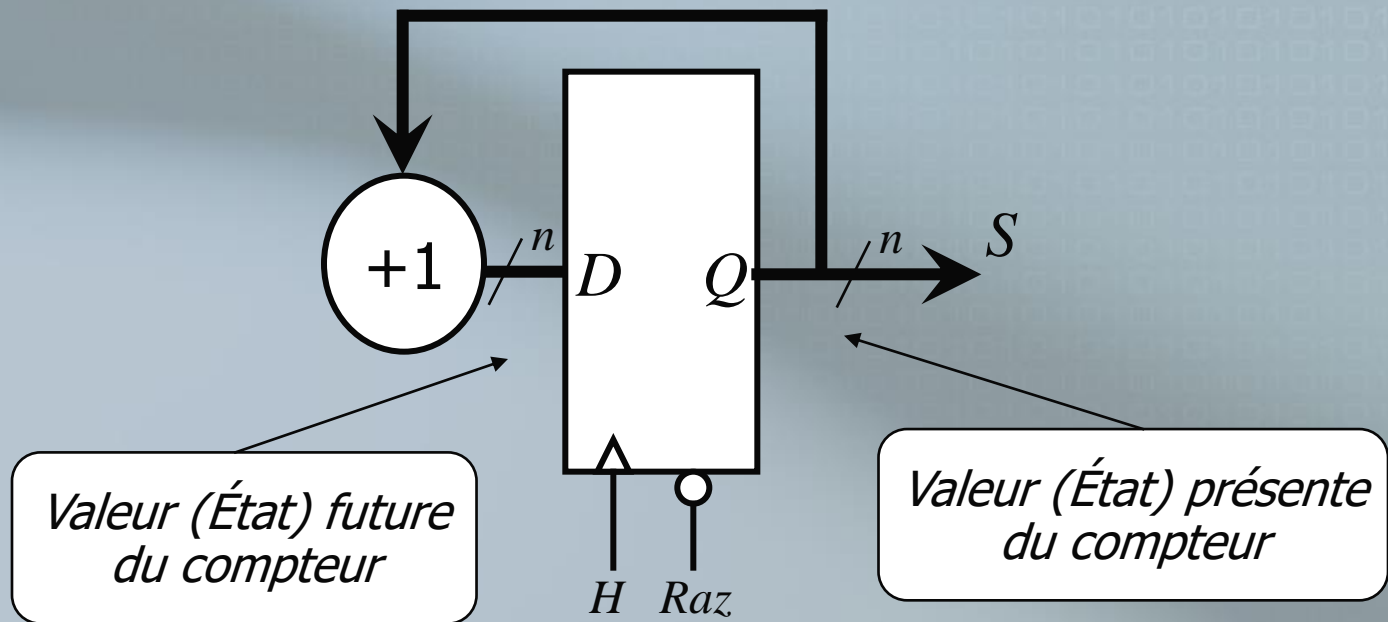
Table des Fonctionnalités

Fonction	Com
Mémo	0
Chgt //	1



# Compteurs

- **Registre + Incrémenteur**
  - État du Compteur mémorisé dans les bascules



C2

48



# Compteurs

## ■ Compteurs Multifonctions

- Architecture similaire à un registre multifonctions
- Exemple
  - Chargement Valeur Initiale
  - Remise à Zéro Synchrone
  - Incrémentation / Décrémentation

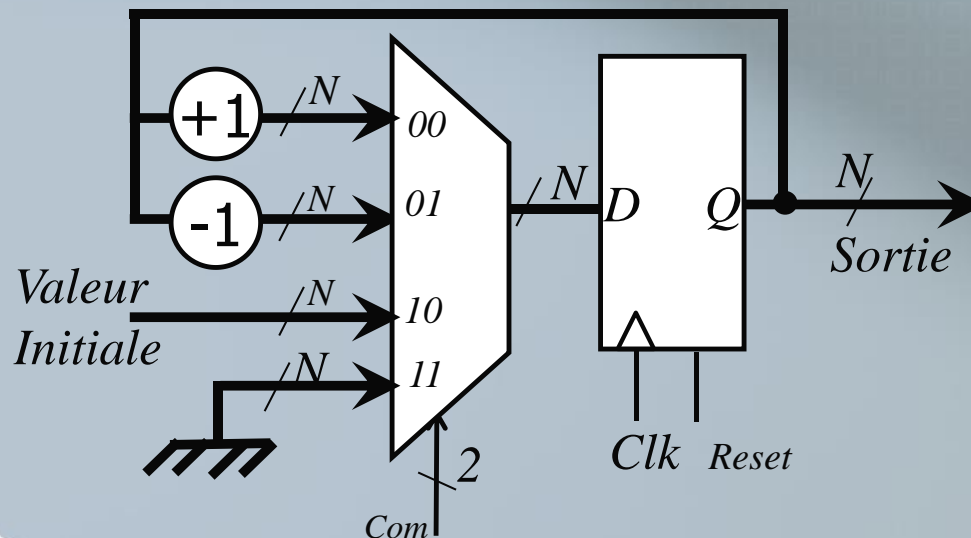


Table des Fonctionnalités

Fonction	Com
Incrémentation	00
Décrémentation	01
Chgt. Valeur	10
RAZ Synchrone	11

# Pour conclure...

- Il est important de maîtriser ces fonctions pour aborder correctement VHDL
- A partir de maintenant, vous êtes censés maîtriser ces architectures combinatoires et séquentielles.
- Si ça n'est pas le cas, remédiez-y...
  - Livres, polys d'anciens cours, questions...

C2

50



Source image: Fox