Logique combinatoire et séquentielle

Logique combinatoire

- Fonction élémentaire
- Représentation logique
- Simplification logique
- Additionneur
- Décodeur
- Multiplexeur

Logique combinatoire et séquentielle

Fonction élémentaire

❑ État logique (binaire ou discret)

- **Élément nul** : valeur binaire **0** (faux, non, bas, ouvert, éteint, vide)
- **Élément unité** : valeur binaire **1** (vrai, oui, haut, fermé, allumé, plein)

❑ Variable logique (bit : binary digit)

- **Grandeur** représentée par un symbole (lettre ou signe) qui peut prendre **2 états logiques** dans le cadre de l'algèbre de Boole.

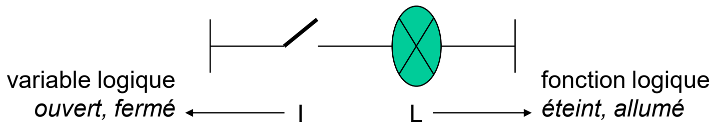
❑ Fonction logique

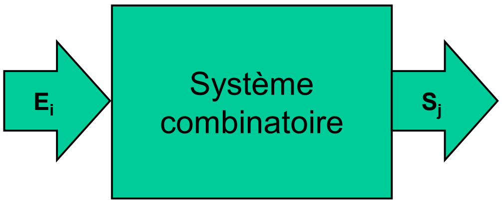
- **Fonction** représentée par des **groupes de variables** réliés par des **opérateurs** logiques qui ne peut prendre que 2 états logiques **0** (point faux) ou **1** (point vrai).

Logique combinatoire et séquentielle

Représentation logique

système binaire :





Logique combinatoire et séquentielle

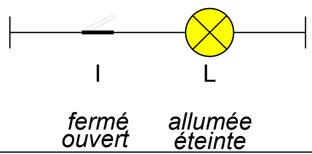
Représentation logique

- ❑ **Algébrique (forme littérale) :**
  - Équation, proposition, expression
  - Formes technologiques
  - Formes canoniques
- ❑ **Graphique :**
  - Table de vérité
  - Tableau de Karnaugh
  - Diagramme d'Euler ou de Venn (théorie des ensembles)
- ❑ **Temporelle :**
  - Chronogramme
- ❑ **Symbolique :**
  - Logigramme
- ❑ **Numérique (écriture condensée)**

Logique combinatoire et séquentielle

La table de vérité (TV):

- Soit F, une fonction de n variables. La table de vérité de F est un tableau de n+1 colonnes et 2^n lignes dans lequel apparaissent toutes les combinaisons d'entrées associées à la valeur correspondante de la fonction.



I	L	
0	0	Point faux
1	1	Point vrai

Logique combinatoire et séquentielle

Convention d'écriture de la TV :

Les variables a, b, et c représente un mot binaire $(a\ b\ c)_2$

Ordre binaire naturel

	a	b	c	f(a,b,c)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Points vrais

Points faux

$(N)_{10}$ est l'équivalent décimal du mot $(a\ b\ c)_2$ avec :
$(N)_{10} = 2^0$´c + 2^1´b + 2^2´a (codage binaire) où :
- c représente le bit le moins significatif (LSB) ou bit de poids faible et
- a représente le bit le plus significatif (MSB) ou bit de poids fort

Exemple : $(1\ 0\ 1)_2 = 2^0$ ´ 1 + 2^1 ´ 0 + 2^2 ´ 1 = 1 + 0 + 4 = $(5)_{10}$

Logique combinatoire et séquentielle

– Exercice :

- *Table de vérité à 4 variables*
- *Fonction f(a,b,c,d)*

	a	b	c	d	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

148

Logique combinatoire et séquentielle

NON ou PAS (NOT)

❑ fonction complément ou fonction inverse. C'est une fonction f d'une variable x telle que :

$$f(x) = \overline{x}$$

système binaire :

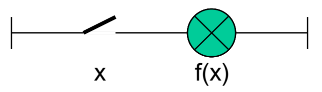
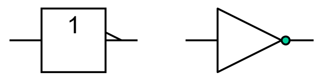


table de vérité :

x	f(x)
0	1
1	0

symbole :



norme CEI norme IEEE

149

Logique combinatoire et séquentielle

ET (AND) : produit logique

- C'est une fonction f de plusieurs variables équivalente à l'intersection en théorie des ensembles. **Elle prend la valeur 1 si toutes les variables sont simultanément égales à 1**. Soient x et y, deux variables booléennes, f(x,y) s'écrit: $\boxed{f(x,y) = x \cdot y = x \wedge y}$

système binaire :
Interrupteurs branchés en série

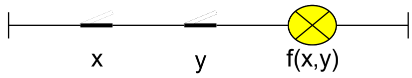
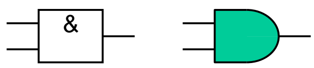


table de vérité :

x	y	f(x,y)
0	0	0
0	1	0
1	0	0
1	1	1

symbole :



norme CEI norme IEEE

Logique combinatoire et séquentielle

OU (inclusif) (OR) : somme logique (produel)

- C'est une fonction f de plusieurs variables équivalente à l'union en théorie des ensembles. Elle prend la valeur ***1*** si au ***moins*** une variable est égale à ***1.*** Soient x et y, deux variables booléennes, f(x,y) s'écrit : $\boxed{f(x,y) = x + y = x \vee y}$

système binaire :
Interrupteurs branchés en parallèle

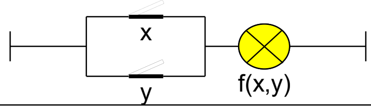
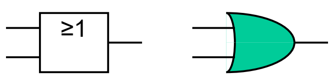


table de vérité :

x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	1

symbole :



norme CEI norme IEEE

Logique combinatoire et séquentielle

Opérateurs d'une variable :

- **fonction nulle : f(x) = 0**

table de vérité :

x	f(x)
0	0
1	0

- **fonction unité : f(x) = 1.**

table de vérité :

x	f(x)
0	1
1	1

- **OUI : fonction identité : f(x) = x.**

système binaire :

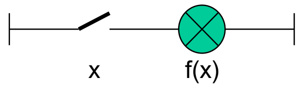
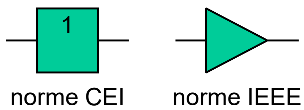


table de vérité :

x	f(x)
0	0
1	1

symbole :



Logique combinatoire et séquentielle

OU Exclusif (XOR) :

- Elle prend la valeur 1 si et seulement si le ***nombre*** de variables égales à 1 est ***impair***. Soient x et y, deux variables booléennes, f(x,y) s'écrit :

$$\boxed{f(x,y) = x \oplus y = \overline{x} \cdot y + x \cdot \overline{y}}$$

système binaire :

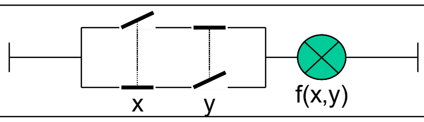
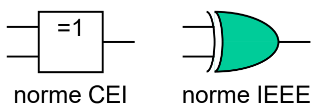


table de vérité :

x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

symbole :



Logique combinatoire et séquentielle

Coïncidence:

- **elle prend la valeur *1* si et seulement si le *nombre* de variables égales à *1* est *pair*.** Soient x et y, deux variables booléennes, f(x,y) s'écrit :

$$\boxed{f(x,y) = \overline{x \oplus y} = \overline{x}\cdot\overline{y} + x\cdot y}$$

système binaire :

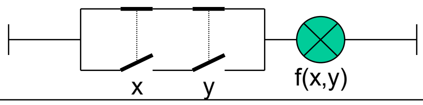
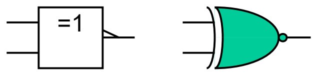


table de vérité :

x	y	x ⊕ y	f(x,y)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

symbole :



norme CEI norme IEEE

Logique combinatoire et séquentielle

NON ET (NAND ou ON) :

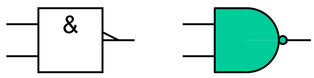
- **Elle prend la valeur 1 si au moins une variable est égale à 0.** C'est un opérateur complet car il permet de réaliser les trois opérateurs de base de l'algèbre de Boole. Soient x et y, deux variables booléennes, f(x,y) s'écrit :

$$\boxed{f(x,y) = \overline{x \cdot y}}$$

table de vérité :

x	y	x · y	f(x,y)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

symbole :



norme CEI norme IEEE

Logique combinatoire et séquentielle

NON OU (NOR ou NI) :

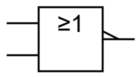
- **Elle prend la valeur 1 si toutes les variables sont simultanément égales à 0.** C'est aussi un opérateur complet. Soient x et y, deux variables booléennes, f(x,y) s'écrit :

$$\boxed{f(x,y) = \overline{x + y}}$$

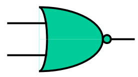
table de vérité :

x	y	$x + y$	f(x,y)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

symbole :



norme CEI



norme IEEE

Logique combinatoire et séquentielle

Table des fonctions logiques f à 2 variables

	(0,0)	(0,1)	(1,0)	(1,1)	f
$f_0(x,y)$	0	0	0	0	**0**
$f_1(x,y)$	0	0	0	1	$\mathbf{x \cdot y}$
$f_2(x,y)$	0	0	1	0	$\mathbf{x \cdot \overline{y}}$
$f_3(x,y)$	0	0	1	1	$\mathbf{x}$
$f_4(x,y)$	0	1	0	0	$\mathbf{\overline{x} \cdot y}$
$f_5(x,y)$	0	1	0	1	$\mathbf{y}$
$f_6(x,y)$	0	1	1	0	$\mathbf{x \oplus y}$
$f_7(x,y)$	0	1	1	1	$\mathbf{x + y}$
$f_8(x,y)$	1	0	0	0	$\mathbf{\overline{x + y}}$
$f_9(x,y)$	1	0	0	1	$\mathbf{\overline{x \oplus y}}$
$f_{10}(x,y)$	1	0	1	0	$\mathbf{\overline{y}}$
$f_{11}(x,y)$	1	0	1	1	$\mathbf{x + \overline{y}}$
$f_{12}(x,y)$	1	1	0	0	$\mathbf{\overline{x}}$
$f_{13}(x,y)$	1	1	0	1	$\mathbf{\overline{x} + y}$
$f_{14}(x,y)$	1	1	1	0	$\mathbf{\overline{x \cdot y}}$
$f_{15}(x,y)$	1	1	1	1	**1**

Logique combinatoire et séquentielle

Propriétés et théorèmes

- **identité (élément neutre) :**
  - $A + 0 = A$
  - $A \cdot 1 = A$

A	0	A + 0
0	0	0
1	0	1

A	1	A · 1
0	1	0
1	1	1

- **involution :**
  - $\overline{\overline{A}} = A$

A	$\overline{A}$	$\overline{\overline{A}}$
0	1	0
1	0	1

- **complémentarité :**
  - $A + \overline{A} = 1$
  - $A \cdot \overline{A} = 0$

A	$\overline{A}$	$A + \overline{A}$
0	1	1
1	0	1

A	$\overline{A}$	$A \cdot \overline{A}$
0	1	0
1	0	0

158

Logique combinatoire et séquentielle

Propriétés et théorèmes

- **Commutativité :**
  - $A + B = B + A$
  - $A \cdot B = B \cdot A$

A	B	A · B	B · A
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

- **Associativité :**
  - $A + (B + C) = (A + B) + C$
  - $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

A	B	C	B·C	A·(B·C)	A·B	(A·B)·C
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1

159

Logique combinatoire et séquentielle

Propriétés et théorèmes

- **distributivité :**
  - **ET sur OU** : A · (B + C) = (A · B) + (A · C) = AB + AC
  - **OU sur ET** : A + (BC) = (A + B) · (A + C)

A	B	C	B·C	A+(BC)	A+B	A+C	(A+B)·(A+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Logique combinatoire et séquentielle

Propriétés et théorèmes

- idempotence (pas d'exposant ou de coefficient) :
  - A · A = A
  - A + A = A

A	A	A · A
0	0	0
1	1	1

A	A	A + A
0	0	0
1	1	1

- Elément absorbant :
  - A · 0 = 0
  - A + 1 = 1

A	0	A · 0
0	0	0
1	0	0

A	1	A + 1
0	1	1
1	1	1

- Absorption :
  - A · (A + B) = A
  - A + (A · B) = A
  - *Démontrer algébriquement ces deux relations*

A	B	A·B	A+(A·B)
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Logique combinatoire et séquentielle

Propriétés et théorèmes

- **De Morgan :**
  - $\overline{A + B} = \overline{A} \cdot \overline{B}$
  - $\overline{A \cdot B} = \overline{A} + \overline{B}$

A	B	A·B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

- **Autre identité** *à démontrer algébriquement* :
  - $AB + \overline{A}B = B$
  - $(A + B) \cdot (\overline{A} + B) = B$
  - $A + \overline{A}B = A + B$
  - $A \cdot (\overline{A} + B) = AB$

- **Principe de dualité**
  - L'expression duale de toute expression logique (pas équation) s'obtient en permutant les opérateurs ET et OU et les éléments 0 et 1 apparaissant dans l'expression.

Logique combinatoire et séquentielle

– **Exercice :**

- *En utilisant les définitions, propriétés et théorèmes de l'algèbre de Boole développer et simplifier la fonction définie par l'équation suivante :*

$$\boxed{F(a,b,c,d,e) = \overline{a \cdot b \oplus b \cdot c + ce + \overline{de}}}$$

Logique combinatoire et séquentielle

Les formes technologiques

- **Première forme :** somme de monômes (produits de littéraux). C'est une forme **disjonctive**. Exemple : $F(x,y,z) = xy + xz + \bar{x}y$
- **Deuxième forme** : produit de monaux (somme de littéraux). C'est une forme **conjonctive**. Exemple : $F(x,y,z) = (x + y) \cdot (x + \bar{z})(x + y)$
- **Formes technologiques associées** : elles s'obtiennent d'après le théorème d'involution et celui de Morgan.

164

Logique combinatoire et séquentielle

Les formes normales ou canoniques

- **Une fonction logique** est sous forme **canonique** si chaque termes (monômes et monaux) contient toutes les variables. C'est aussi une forme technologique.
- **Forme normale disjonctive (1ère forme canonique)** : somme de monômes contenant chacun toutes les variables (intersection de base ou *min terme*). Exemple : $F(x,y,z) = x\,y\,z + \bar{x}\,\bar{y}\,z + \bar{x}\,y\,z$
- **Forme normale conjonctive (2ème forme canonique) :** produit de monaux contenant chacun toutes les variables (réunion de base ou *max terme*). Exemple : $F(x,y,z) = (x + y + z)(\bar{x} + \bar{y} + z)(\bar{x} + y + z)$

165

Logique combinatoire et séquentielle

Table de vérité → équation logique

a	b	c	f(a,b,c)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Points vrais :

$F(0,0,0) = 1 = \overline{a}\ \overline{b}\ \overline{c}$

$F(0,1,1) = 1 = \overline{a}\ b\ c$

$F(1,0,0) = 1 = a\ \overline{b}\ \overline{c}$

$F(1,0,1) = 1 = a\ \overline{b}\ c$

Forme normale disjonctive :
Elle ne comprend que les min termes pour lesquels la valeur particulière de la fonction est égale à **1** (points vrais).
Le nombre de termes de la réunion est égale au nombre de 1 de la fonction figurant dans la table de vérité.

$$\boxed{f(a,b,c) = \overline{a}\cdot\overline{b}\cdot\overline{c} + \overline{a}\cdot b\cdot c + a\cdot\overline{b}\cdot\overline{c} + a\cdot\overline{b}\cdot c}$$

166

Logique combinatoire et séquentielle

Table de vérité → équation logique

a	b	c	f(a,b,c)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Forme normale conjonctive :
Elle ne comprend que les max termes pour lesquels la valeur particulière de la fonction est égale à **0** (points faux).
Le nombre de termes de la réunion est égale au nombre de 0 de la fonction figurant dans la table de vérité.

Points faux :

$F(0,0,1) = 0 = a+b+\overline{c}$

$F(0,1,0) = 0 = a+\overline{b}+c$

$F(1,1,0) = 0 = \overline{a}+\overline{b}+c$

$F(1,1,1) = 0 = \overline{a}+\overline{b}+\overline{c}$

$$\boxed{f(a,b,c) = (a+b+\overline{c})\cdot(a+\overline{b}+c)\cdot(\overline{a}+\overline{b}+c)\cdot(\overline{a}+\overline{b}+\overline{c})}$$

167

Logique combinatoire et séquentielle

• Exercice 1 :

- *Extraire les équations logiques des tables de vérité des fonctions suivantes :*
  - *1ère forme canonique de la fonction f7*
  - *2ème forme canonique de la fonction f1*
  - *1ère forme et 2ème forme canonique de la fonction f6*
  - *1ère forme et 2ème forme canonique de la fonction f9*
- *En utilisant les règles de l'algèbre de Boole, simplifier ces fonctions.*

Logique combinatoire et séquentielle

Exercice 2 :

- *Mêmes questions que l'exercice 1 en utilisant la table de vérité suivante :*

a	b	c	f(a,b,c)
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

Logique combinatoire et séquentielle

Equation logique → Table de vérité

Compter le nombre de variables différentes dans l'équation et créer la table de vérité.

- **1ère méthode :** pour chacune des combinaisons de la table de vérité, évaluer l'équation et reporter le résultat dans la table.
- **2ème méthode :** mettre l'équation sous une forme technologique et pour chacun des termes de la forme choisie, reporter les 1 ou les 0 dans les cases correspondantes de la table de vérité (plusieurs reports par terme).

170

Logique combinatoire et séquentielle

Equation logique → Table de vérité

•Compter le nombre de variables différentes dans l'équation et créer la table de vérité.

- **3ème méthode :** mettre l'équation sous une forme canonique et pour chacun des termes de la forme choisie, reporter les 1 ou les 0 dans les cases correspondantes de la table de vérité (un seul report par terme).

171

Exemple :

- **$F(a,b) = a \oplus b \cdot c$**
- **3 variables a, b et c**

a	b	c	f(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

- $F(a,b) = a \oplus b \cdot c$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b \cdot c}$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot (\overline{b} + \overline{c})$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b} + a \cdot \overline{c}$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b} + a \cdot \overline{c}$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot (c + \overline{c}) + a \cdot \overline{c} \cdot (b + \overline{b})$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot c + a \cdot \overline{b} \cdot \overline{c} + a \cdot b \cdot \overline{c} + a \cdot \overline{b} \cdot \overline{c}$
- $F(a,b) = \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot c + a \cdot \overline{b} \cdot \overline{c} + a \cdot b \cdot \overline{c}$

Logique combinatoire et séquentielle

Les formes numériques

- Chaque combinaison est repérée par un numéro (en général, l'équivalent décimal) afin de condenser l'écriture.
  - –Exemple précédent :

$F(a,b,c) = \sum (3, 4, 5, 6)$

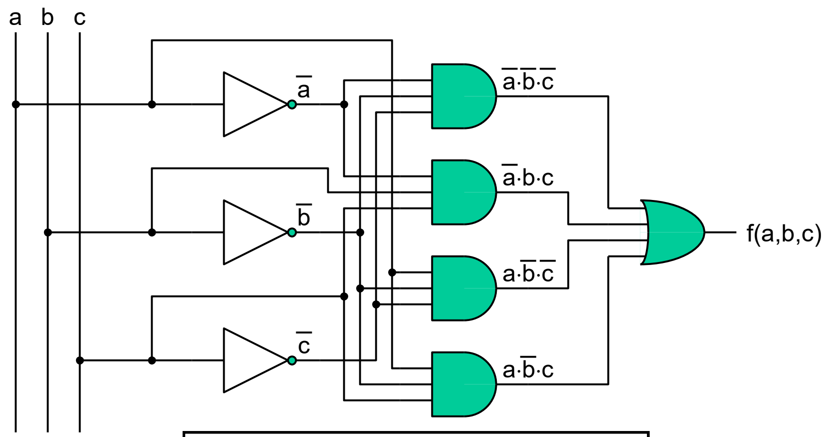
$F(a,b,c) = \prod (0, 1, 2, 7)$

Les logigrammes

- C'est une association des symboles utilisés pour représenter les fonctions logiques en vue de leur réalisation câblée ou programmée.
- Le logigramme le plus simple est celui qui utilise le moins d'opérateurs possible et de même type.

Logique combinatoire et séquentielle

Logigramme d'une première forme canonique en norme américaine IEEE



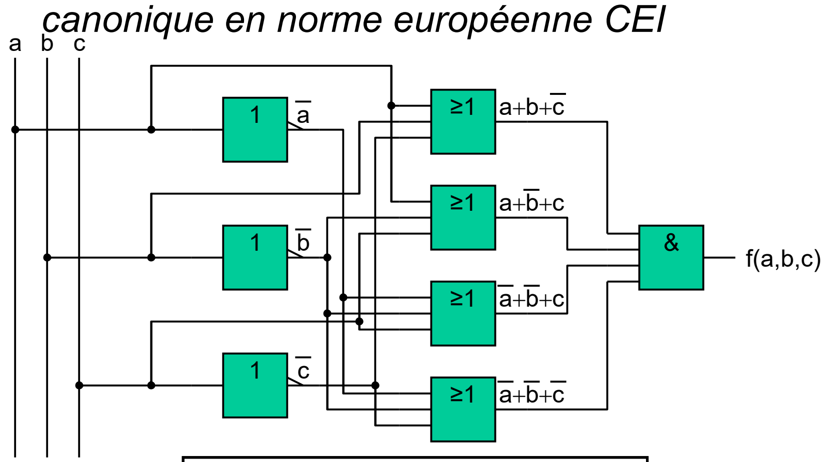
$$f(a,b,c) = \overline{a}\cdot\overline{b}\cdot\overline{c} + \overline{a}\cdot b\cdot c + a\cdot\overline{b}\cdot\overline{c} + a\cdot\overline{b}\cdot c$$

174

Logique combinatoire et séquentielle

– **Exercice** :

❑ *réaliser le logigramme d'une deuxième forme canonique en norme européenne CEI*



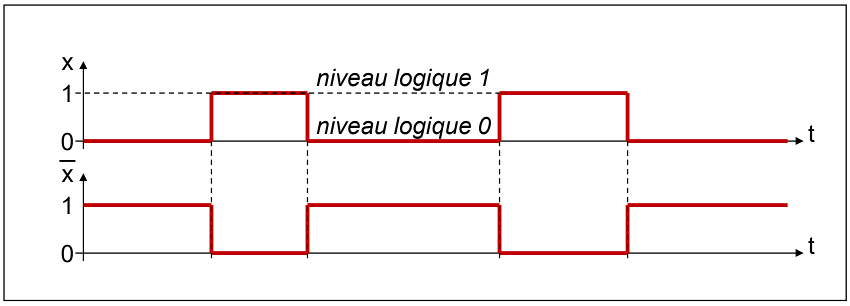
$$f(a,b,c) = (a+b+\overline{c})\cdot(a+\overline{b}+c)\cdot(\overline{a}+\overline{b}+c)\cdot(\overline{a}+\overline{b}+\overline{c})$$

175

Logique combinatoire et séquentielle

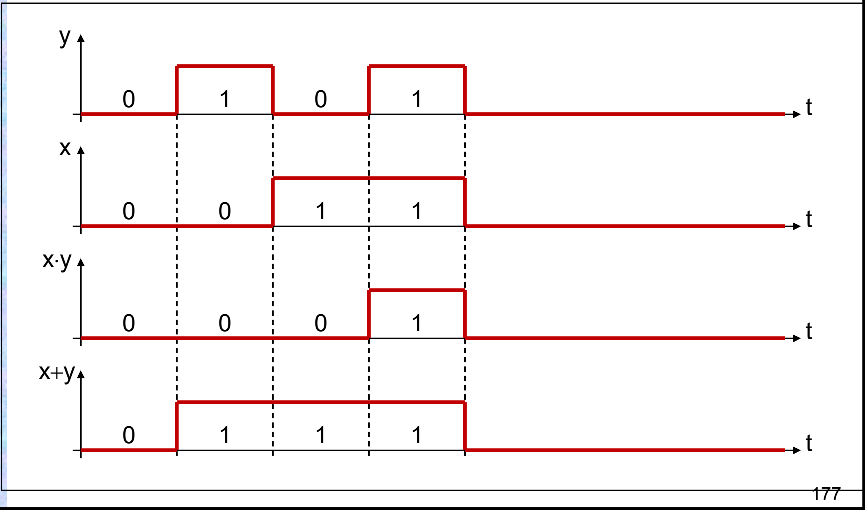
Les chronogrammes

- ❑ C'est le **graphique** d'évolution temporelle des variables et des fonctions logiques.
- ❑ **Exemple** : chronogramme de la fonction Non



Logique combinatoire et séquentielle

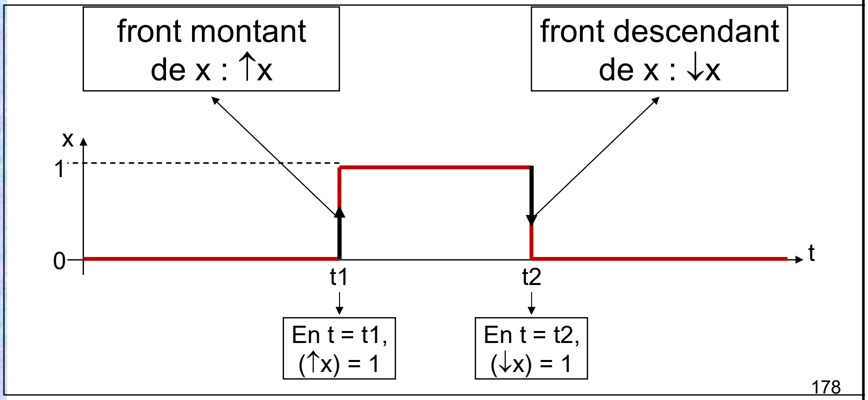
Chronogramme des fonctions ET et OU :



Logique combinatoire et séquentielle

Fronts des chronogrammes

- L'instant de passage de 0 à 1 est un front montant.
- L'instant de passage de 1 à 0 est un front descendant.
- La succession de ces deux fronts forme une impulsion.



178

Logique combinatoire et séquentielle

Les tableaux de Karnaugh

- Soit F, une fonction de *n* variables.
- Le tableau de Karnaugh est un tableau de 2^n cases correspondant aux 2^n combinaisons d'entrée dans lesquelles sont notées les valeurs correspondantes de la fonction.

a	b	c	f(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a \ bc	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	1 (4)	0 (5)	1 (7)	0 (6)

tableau de Karnaugh

Le passage d'une case à une case voisine se fait par changement de la valeur d'une seule variable à la fois (code binaire réfléchi ou code de Gray).

179

Logique combinatoire et séquentielle

Exercice :

– *Réaliser le tableau de Karnaugh de la fonction f de 4 variables a,b,c et d définie par la table de vérité suivante :*

a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

ab \ cd	00	01	11	10
00	0 $_{0}$	1 $_{1}$	0 $_{3}$	1 $_{2}$
01	1 $_{4}$	0 $_{5}$	1 $_{7}$	0 $_{6}$
11	0 $_{12}$	1 $_{13}$	0 $_{15}$	1 $_{14}$
10	1 $_{8}$	0 $_{9}$	1 $_{11}$	0 $_{10}$

180

Logique combinatoire et séquentielle

Méthodes de simplification

- **Méthode algébrique** : application des principes de l'algèbre de Boole
  - Mise en facteur ou développement
  - Idempotence...
- **Méthode graphique** : utilisation des tableaux de Karnaugh
  - Deux cases sont adjacentes si le passage de l'une à l'autre se fait uniquement par le changement d'état d'une seule variable.
  - Ce principe s'applique également pour des ensembles de cases adjacentes constitués de 2^n cases.

181

Logique combinatoire et séquentielle

– **Exemple** :

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

182

Logique combinatoire et séquentielle

Méthode de Karnaugh :

- Selon la forme recherchée, **regrouper les cases adjacentes de même valeur** (soit 0, soit 1) par des ensembles les plus grands possibles et correspondant à des puissances de 2.
- Tous ces regroupements correspondent à des **monômes premiers** (ou monaux) et constitue la base première complète.
- La somme de ces monômes donne une **expression simplifiée** de la fonction logique mais pas sa forme minimale.

183

Logique combinatoire et séquentielle

Méthode de Karnaugh :

- ❑ La forme minimale est obtenue en faisant la somme des **monômes premiers principaux** (un regroupement où il existe au moins une case qui ne peut être regroupée que par elle-même).
- ❑ Parfois, plusieurs possibilités sont offertes et on obéira aux règles suivantes :
  - Tous les 1 (ou tous les 0) doivent être regroupés au moins une fois.
  - Les regroupements les plus grands doivent être choisis en priorité.
  - Les cases à regrouper doivent l'être un minimum de fois (commencer par celles qui n'ont qu'une seule façon de se regrouper).

184

Logique combinatoire et séquentielle

– Exemple :

$F=\bar{a}\bar{b}\bar{c}\bar{d}+\bar{a}b\bar{c}\bar{d}+\bar{a}b\bar{c}d+\bar{a}bcd+ab\bar{c}d+abcd+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{b}\bar{c}\bar{d}+\bar{a}b\bar{c}\bar{d}+\bar{a}b\bar{c}\bar{d}+\bar{a}b\bar{c}d+\bar{a}b\bar{c}d+\bar{a}bcd+ab\bar{c}d+abcd+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}(\bar{b}+b)+\bar{a}b\bar{c}(\bar{d}+d)+bd(\bar{a}\bar{c}+\bar{a}c+a\bar{c}+ac)+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+\bar{a}b\bar{c}+bd[\bar{a}(\bar{c}+c)+a(\bar{c}+c)]+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+\bar{a}b\bar{c}+bd(\bar{a}+a)+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+\bar{a}b\bar{c}+bd+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+\bar{a}b\bar{c}(\bar{d}+d)+bd+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+\bar{a}b\bar{c}\bar{d}+\bar{a}b\bar{c}d+bd+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}(1+b)+bd(\bar{a}\bar{c}+1)+a\bar{b}c\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+bd+a\bar{b}c\bar{d}$

F est la somme des monômes premiers principaux (irredondants).

ab \ cd	00	01	11	10
00	1	0	0	0
01	1	1	1	0
11	0	1	1	0
10	0	0	0	1

$\bar{a}\bar{c}\bar{d}$

$\bar{a}b\bar{c}$

bd

$a\bar{b}c\bar{d}$

redondant

monômes premiers

185

Logique combinatoire et séquentielle

- **Exercice :**

❑ *Simplifier sous une 1ére et 2ème forme technologique la fonction définie par la table de vérité suivante en utilisant la méthode de Karnaugh :*

a	b	c	f(a,b,c)
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

186

Logique combinatoire et séquentielle

Cas des fonctions incomplètement définies :

❑ Certaines combinaisons **ne peuvent jamais exister**.

❑ La valeur de la fonction **n'a pas d'importance** pour certaines combinaisons de variables.

❑ La valeur de la fonction est dite **indifférente** ou la combinaison **interdite**. La valeur de la fonction est alors notée **Φ** ou **X** et peut prendre **indifféremment** la valeur **1 ou 0** selon qu'elle sert ou non à la **simplification**.

187

Logique combinatoire et séquentielle

Cas des fonctions incomplètement définies :

ab \ cd	00	01	11	10
00	1	0	0	0
01	1	1	Φ	0
11	0	Φ	1	0
10	Φ	Φ	0	1

$\bar{a}\bar{c}\bar{d}$

bd

$a\bar{b}\bar{d}$

$F=\bar{a}\bar{c}\bar{d}+bd+a\bar{b}\bar{d}$

Logique combinatoire et séquentielle

Exercice :

- *Simplifier sous une 1ére forme technologique la fonction suivante en utilisant la méthode de Karnaugh :*

ab \ cde	000	001	011	010	110	111	101	100
00	1	1	1	1	0	1	0	1
01	1	0	0	0	0	0	0	1
11	1	0	0	0	1	0	0	1
10	1	1	1	1	0	1	1	1

Logique combinatoire et séquentielle

– Exemple :

	a	b	c	d	f(a,b,c,d)
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Points vrais :

$F(0,0,0,1) = 1 = \bar{a}\bar{b}\bar{c}d$

$F(0,0,1,0) = 1 = \bar{a}\bar{b}c\bar{d}$

$F(0,1,0,0) = 1 = \bar{a}b\bar{c}\bar{d}$

$F(0,1,1,1) = 1 = \bar{a}bcd$

$F(1,0,0,0) = 1 = a\bar{b}\bar{c}\bar{d}$

$F(1,0,1,1) = 1 = a\bar{b}cd$

$F(1,1,0,1) = 1 = ab\bar{c}d$

$F(1,1,1,0) = 1 = abc\bar{d}$

Points faux :

$F(0,0,0,0) = 0 = a+b+c+d$

$F(0,0,1,1) = 0 = a+b+\bar{c}+\bar{d}$

$F(0,1,0,1) = 0 = a+\bar{b}+c+\bar{d}$

$F(0,1,1,0) = 0 = a+\bar{b}+\bar{c}+d$

$F(1,0,0,1) = 0 = \bar{a}+b+c+\bar{d}$

$F(1,0,1,0) = 0 = \bar{a}+b+\bar{c}+d$

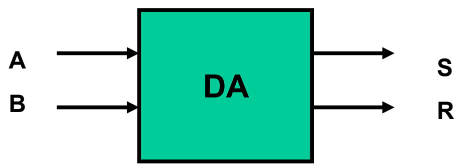
$F(1,1,0,0) = 0 = \bar{a}+\bar{b}+c+d$

$F(1,1,1,1) = 0 = \bar{a}+\bar{b}+\bar{c}+\bar{d}$

Logique combinatoire et séquentielle

Demi Additionneur

- Le demi additionneur est un circuit combinatoire qui permet de réaliser la somme arithmétique de deux nombres A et B chacun sur un bit.
- A la sotie on va avoir la somme S et la retenu R (Carry).



➔Pour trouver la structure (le schéma) de ce circuit on doit en premier dresser sa table de vérité

19

Logique combinatoire et séquentielle

Addition

- En binaire l'addition sur un seul bit se fait de la manière suivante:

$$\begin{cases} 0+0 &= 00 \\ 0+1 &= 01 \\ 1+0 &= 01 \\ 1+1 &= 10 \end{cases}$$

•La table de vérité associée :

A	B	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

De la table de vérité on trouve :

$$R = A.B$$

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

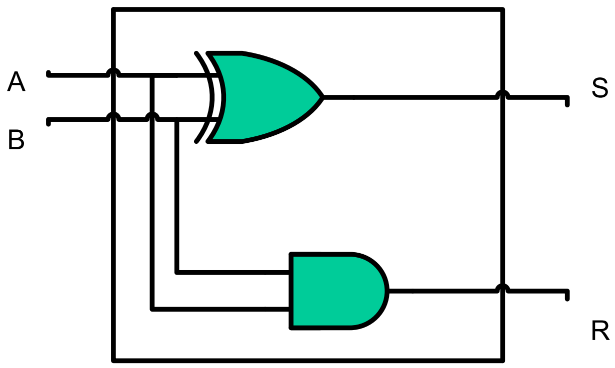
192

Logique combinatoire et séquentielle

Logigramme du demi additionneur

$$R = A.B$$

$$S = A \oplus B$$



193

Logique combinatoire et séquentielle

L'additionneur complet

- ❑ En binaire lorsque on fait une addition il faut tenir en compte de la retenue entrante.

$$
\begin{array}{cccccc}
 & r_4 & r_3 & r_2 & r_1 & r_0 = 0 \\
 & & a_4 & a_3 & a_2 & a_1 \\
+ & & b_4 & b_3 & b_2 & b_1 \\
\hline
 & r_4 & s_4 & s_3 & s_2 & s_1
\end{array}
\qquad\qquad
\begin{array}{ccc}
 & & r_{i-1} \\
 & & a_i \\
+ & & b_i \\
\hline
 & r_i & s_i
\end{array}
$$

194

Logique combinatoire et séquentielle

Additionneur complet 1 bit

- ❑ L'additionneur complet un bit possède 3 entrées :
  - a_i : le premier nombre sur un bit.
  - b_i : le deuxième nombre sur un bit.
  - r_{i-1} : la retenue entrante sur un bit.
- ❑ Il possède deux sorties :
  - S_i : la somme
  - R_i la retenue sortante

195

Logique combinatoire et séquentielle

Table de vérité d'un additionneur complet sur 1 bit

a_i	b_i	r_{i-1}		r_i	s_i
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

$$S_i = \overline{A}_i.\overline{B}_i.R_{i-1} + \overline{A}_i.B_i.\overline{R}_{i-1} + A_i.\overline{B}_i.\overline{R}_{i-1} + A_i.B_i.R_{i-1}$$

$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

196

Logique combinatoire et séquentielle

Simplification de l'équation logique d'un additionneur complet sur 1 bit

Si on veut simplifier les équations on obtient :

$$S_i = \overline{A}_i.\overline{B}_i.R_{i-1} + \overline{A}_i.B_i.\overline{R}_{i-1} + A_i.\overline{B}_i.\overline{R}_{i-1} + A_i.B_i.R_{i-1}$$

$$S_i = \overline{A}_i.(\overline{B}_i.R_{i-1} + B_i.\overline{R}_{i-1}) + A_i.(\overline{B}_i.\overline{R}_{i-1} + B_i.R_{i-1})$$

$$S_i = \overline{A}_i (B_i \oplus R_{i-1}) + A_i.(\overline{B_i \oplus R_{i-1}})$$

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

$$R_i = R_{i-1}.(\overline{A}_i.B_i + A_i.\overline{B}_i) + A_i B_i (\overline{R}_{i-1} +_i R_{i-1})$$

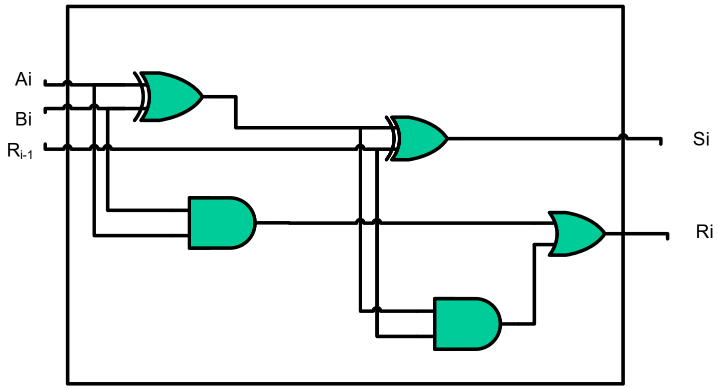
$$R_i = R_{i-1}.(A_i \oplus B_i) + A_i B_i$$

197

Logique combinatoire et séquentielle

Schéma d'un additionneur complet

$$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$$
$$S_i = A_i \oplus B_i \oplus R_{i-1}$$



Logique combinatoire et séquentielle

En utilisant des Demi-Additionneurs

$$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$$
$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

Si on pose $X = A_i \oplus B_i$ et $Y = A_i B_i$

On obtient :

$$R_i = Y + R_{i-1}.X$$
$$S_i = X \oplus R_{i-1}$$

et si on pose $Z = X \oplus R_{i-1}$ et $T = R_{i-1}.X$

On obtient :

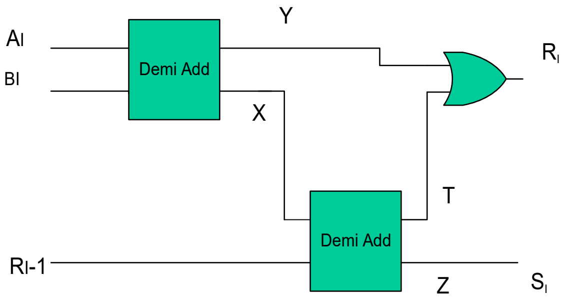
$$R_i = Y + T$$
$$S_i = Z$$

- On remarque que X et Y sont les sorties d'un demi-additionneur ayant comme entrées A et B
- On remarque que Z et T sont les sorties d'un demi-additionneur ayant comme entrées X et R_{i-1}

Logique combinatoire et séquentielle

En utilisant des Demi-Additionneurs

$$X = A_i \oplus B_i$$
$$Y = A_i . B_i$$
$$Z = X \oplus R_{i-1}$$
$$T = R_{i-1} . X$$
$$R_i = Y + T$$
$$S_i = Z$$



Logique combinatoire et séquentielle

Additionneur sur 4 bits

- Un additionneur sur 4 bits est un circuit qui permet de faire l'addition de deux nombres A et B de 4 bits chacun
  - $A(a_3a_2a_1a_0)$
  - $B(b_3b_2b_1b_0)$
  - En plus il tient en compte de la retenu entrante
- En sortie on va avoir le résultat sur 4 bits ainsi que la retenu (5 bits en sortie)
- Donc au total le circuit possède 9 entrées et 5 sorties.
- Avec 9 entrées on a 2^9=512 combinaisons !!!!!! Comment faire pour représenter la table de vérité ?????
- Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit ?

Logique combinatoire et séquentielle

Additionneur 4 bits (opération)

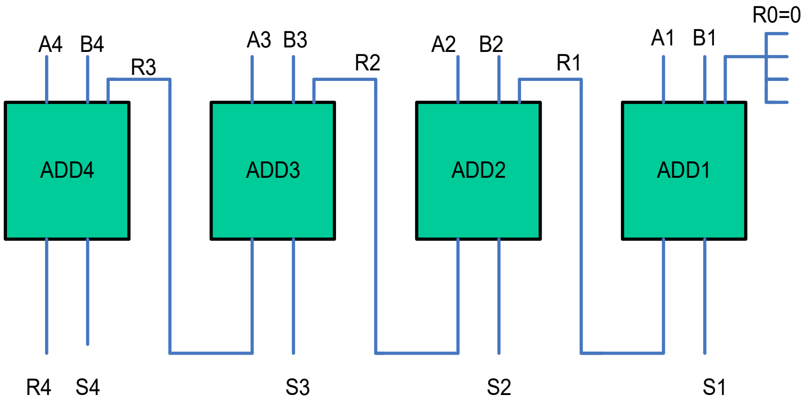
- Lorsque on fait l'addition en binaire , on additionne bit par bit en commençant à partir du poids fiable et à chaque fois on propage la retenue sortante au bit du rang supérieur.
  L'addition sur un bit peut se faire par un additionneur complet sur 1 bits.

$$
\begin{array}{ccccc}
 & r_3 & r_2 & r_1 & r_0 = 0 \\
 & a_4 & a_3 & a_2 & a_1 \\
+ & b_4 & b_3 & b_2 & b_1 \\
\hline
 & r_4\ s_4 & r_3\ s_3 & r_2\ s_2 & r_1\ s_1
\end{array}
$$

$\boxed{r4 \quad s_4 \quad s_3 \quad s_2 \quad s_1}$ **Résultat final**

Logique combinatoire et séquentielle

Additionneur 4 bits (schéma)



Logique combinatoire et séquentielle

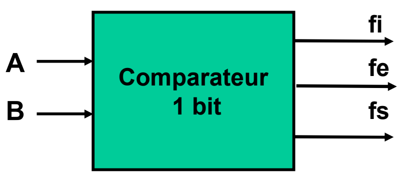
•Exercice

- Soit une information binaire sur 5 bits ($i_4i_3i_2i_1i_0$).
  Donner le circuit qui permet de calculer le nombre de 1 dans l'information en entrée en utilisant uniquement des additionneurs complets sur 1 bit ?
- **Exemple** :
  - Si on a en entrée l'information ($i_4i_3i_2i_1i_0$) =(10110) alors en sortie on obtient la valeur 3 en binaire (011) puisque il existe 3 bits qui sont à 1 dans l'information en entrée .

Logique combinatoire et séquentielle

Compteur

- C'est un circuit combinatoire qui permet de comparer entre deux nombres binaire A et B.
- Il possède 2 entrées :
  - A : sur un bit
  - B : sur un bit
- Il possède 3 sorties
  - fe : égalité (A=B)
  - fi : inférieur (A < B)
  - fs : supérieur (A > B)



Logique combinatoire et séquentielle

Comparateur sur un bit

A	B		fs	fe	fi
0	0		0	1	0
0	1		0	0	1
1	0		1	0	0
1	1		0	1	0

$$fs = A.\overline{B}$$

$$fi = \overline{A}B$$

$$fe = \overline{A}\,\overline{B} + AB = \overline{A \oplus B} = \overline{fs + fi}$$

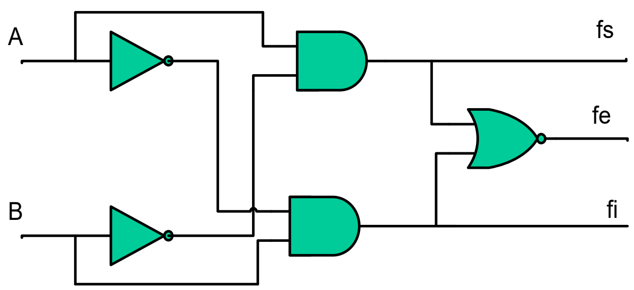
Logique combinatoire et séquentielle

Schéma d’un comparateur dur un bit

$$fs = A.\overline{B}$$

$$fi = \overline{A}B$$

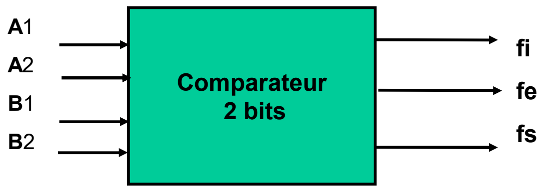
$$fe = \overline{fs + fi}$$



Logique combinatoire et séquentielle

Comparateur 2 bits

- Il permet de faire la comparaison entre deux nombres A (a_2a_1) et B(b_2b_1) chacun sur deux bits.



1. A=B si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1})$$

2. A>B si

A2 > B2 ou (A2=B2 et A1>B1)

$$fs = A2.\overline{B2} + (\overline{A2 \oplus B2}).(A1.\overline{B1})$$

3. A<B si

A2 < B2 ou (A2=B2 et A1<B1)

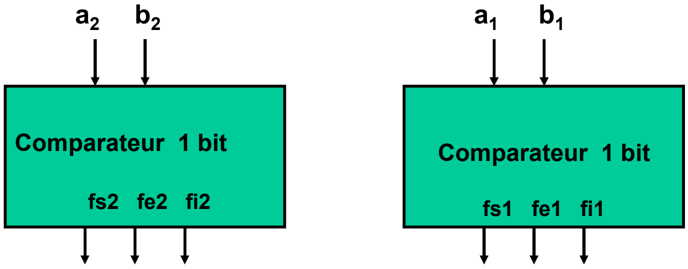
$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1)$$

A2	A1	B2	B1		fs	fe	fi
0	0	0	0		0	1	0
0	0	0	1		0	0	1
0	0	1	0		0	0	1
0	0	1	1		0	0	1
0	1	0	0		1	0	0
0	1	0	1		0	1	0
0	1	1	0		0	0	1
0	1	1	1		0	0	1
1	0	0	0		1	0	0
1	0	0	1		1	0	0
1	0	1	0		0	1	0
1	0	1	1		0	0	1
1	1	0	0		1	0	0
1	1	0	1		1	0	0
1	1	1	0		1	0	0
1	1	1	1		0	1	0

Logique combinatoire et séquentielle

Comparateur 2 bits avec des comparateurs 1 bit

- C'est possible de réaliser un comparateur 2 bits en utilisant des comparateurs 1 bit et des portes logiques.
- Il faut utiliser un comparateur pour comparer les bits du poids faible et autre pour comparer les bits du poids fort.
- Il faut combiner entre les sorties des deux comparateurs utilisés pour réaliser les sorties du comparateur final.



Logique combinatoire et séquentielle

Expressions logiques

1. **A=B** si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1}) = fe2.fe1$$

2. **A>B** si

A2 > B2 ou (A2=B2 et A1>B1)

$$fs = A2.B2 + (A2 \oplus B2).(A1.\overline{B1}) = fs2 + fe2.fs1$$

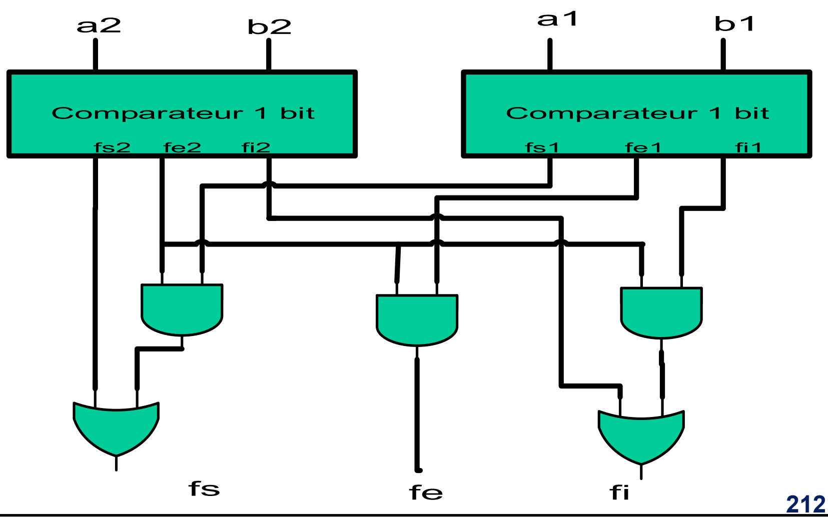
3. **A<B** si

A2 < B2 ou (A2=B2 et A1<B1)

$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1) = fi2 + fe2.fi1$$

Logique combinatoire et séquentielle

Logigramme



212

Logique combinatoire et séquentielle

Comparateur avec des entrées de mise en cascade

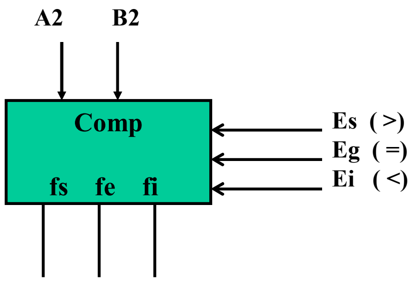
- On remarque que :
  - Si A2 >B2 alors **A > B**
  - Si A2<B2 alors **A < B**
- Par contre si A2=B2 alors il faut tenir en compte du résultat de la comparaison des bits du poids faible.
- Pour cela on rajoute au comparateur des entrées qui nous indiquent le résultat de la comparaison précédente.
- Ces entrées sont appelées des entrées de mise en cascade.

213

Logique combinatoire et séquentielle

Comparateur avec des entrées de mise en cascade

A2	B2	Es	Eg	Ei		fs	fe	fi
A2>B2		X	X	X		1	0	0
A2<B2		X	X	X		0	0	1
A2=B2		1	0	0		1	0	0
		0	1	0		0	1	0
		0	0	1		0	0	1

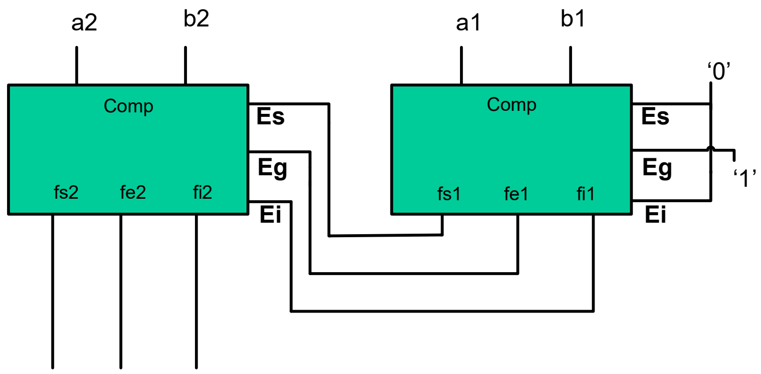


fs= (A2>B2) ou (A2=B2).Es
fi= (A2<B2) ou (A2=B2).Ei
fe=(A2=B2).Eg

214

Logique combinatoire et séquentielle

Comparateur avec des entrées de mise en cascade

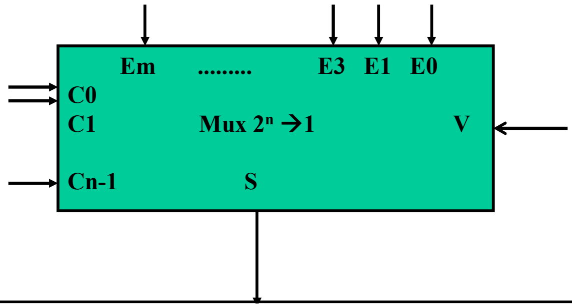


215

Logique combinatoire et séquentielle

Multiplexeur

- Un multiplexeur est un circuit combinatoire qui permet de sélectionner une information (1 bit) parmi 2^n valeurs en entrée.
- Il possède :
  - 2^n entrées d'information
  - Une seule sortie
  - N entrées de sélection (commandes)

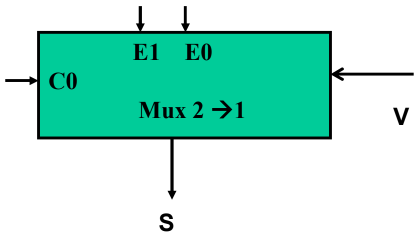


216

Logique combinatoire et séquentielle

Multiplexeur 2 →1

V	C_0		S
0	X		0
1	0		E0
1	1		E1



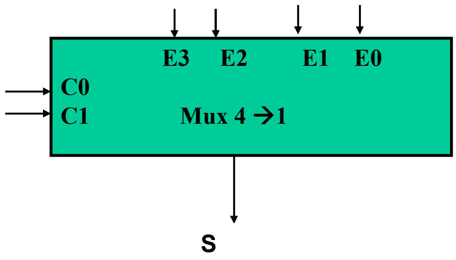
$$S = V.(\overline{C_0}.E0 + C_0.E1)$$

217

Logique combinatoire et séquentielle

Multiplexeur 4 →1

C1	C0		S
0	0		E0
0	1		E1
1	0		E2
1	1		E3

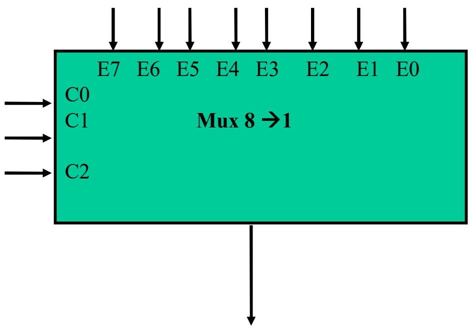


$$S = \overline{C1}.\overline{C0}.(E0) + \overline{C1}.C0.(E1) + C1.\overline{C0}.(E2) + C1.C0.(E3)$$

Logique combinatoire et séquentielle

Multiplexeur 8→1

C2	C1	C0		S
0	0	0		E0
0	0	1		E1
0	1	0		E2
0	1	1		E3
1	0	0		E4
1	0	1		E5
1	1	0		E6
1	1	1		E7

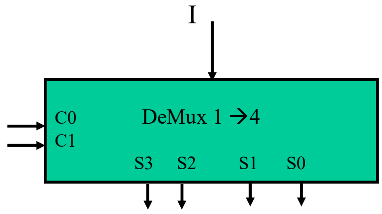


$$S = \overline{C2}.\overline{C1}.\overline{C0}.(E0) + \overline{C2}.\overline{C1}.C0(E1) + \overline{C2}.C1.\overline{C0}(E2) + \overline{C2}.C1.C0(E3) +$$
$$C2.\overline{C1}.\overline{C0}(E4) + C2.\overline{C1}.C0(E5) + C2.C1.\overline{C0}(E6) + C2.C1.C0(E7)$$

Logique combinatoire et séquentielle

Démultiplexeurs

- Il joue le rôle inverse d'un multiplexeur, il permet de faire passer une information dans l'une des sorties selon les valeurs des entrées de commandes.
- Il possède :
  - une seule entrée
  - 2^n sorties
  - N entrées de sélection (commandes)



Logique combinatoire et séquentielle

Démultiplexeur 1→4

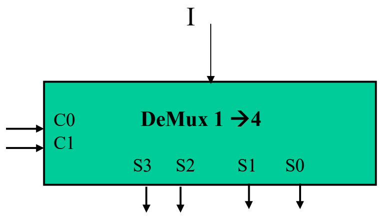
C1	C0		S3	S2	S1	S0
0	0		0	0	0	I
0	1		0	0	i	0
1	0		0	i	0	0
1	1		i	0	0	0

$$S0 = \overline{C1}.\overline{C0}.(I)$$

$$S1 = \overline{C1}.C0.(I)$$

$$S2 = C1.\overline{C0}.(I)$$

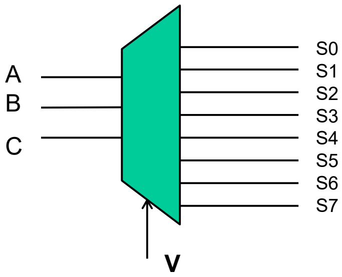
$$S3 = C1.C0.(I)$$



Logique combinatoire et séquentielle

Le décodeur binaire

- C'est un circuit combinatoire qui est constitué de :
  - N : entrées de données
  - 2^n sorties
  - Pour chaque combinaison en entrée une seule sortie est active à la fois

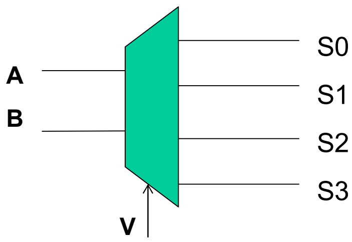


Un décodeur 3→8

Logique combinatoire et séquentielle

Décodeur 2→4

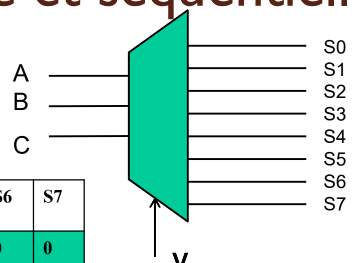
V	A	B		S0	S1	S2	S3
0	X	X		0	0	0	0
1	0	0		1	0	0	0
1	0	1		0	1	0	0
1	1	0		0	0	1	0
1	1	1		0	0	0	1



$$S_0 = (\overline{A}.\overline{B}).V$$
$$S_1 = (\overline{A}.B).V$$
$$S_2 = (A.\overline{B}).V$$
$$S_3 = (A.B).V$$

Logique combinatoire et séquentielle

Décodeur 3→8



A	B	C	S0	S1	S2	S3	S4	S5	S6	S7
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

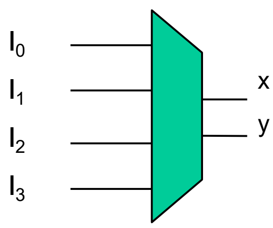
$$S_0 = \overline{A}.\overline{B}.\overline{C}$$
$$S_1 = \overline{A}.\overline{B}.C$$
$$S_2 = \overline{A}.B.\overline{C}$$
$$S_3 = \overline{A}.B.C$$
$$S_4 = A.\overline{B}.\overline{C}$$
$$S_5 = A.\overline{B}.C$$
$$S_6 = A.B.\overline{C}$$
$$S_7 = A.B.C$$

Logique combinatoire et séquentielle

L'encodeur binaire

- ❑ Il joue le rôle inverse d'un décodeur
  - Il possède 2^n entrées
  - N sortie
  - Pour chaque combinaison en entrée on va avoir sont numéro (en binaire) à la sortie.

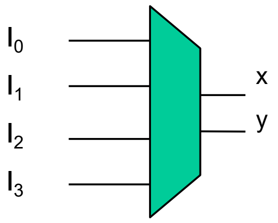
Encodeur 4→2



Logique combinatoire et séquentielle

L'encodeur binaire (4→2)

I_0	I_1	I_2	I_3		x	y
0	0	0	0		0	0
1	x	x	x		0	0
0	1	x	x		0	1
0	0	1	x		1	0
0	0	0	1		1	1



$$X = \overline{I0}.\overline{I1}.(I2 + I3)$$

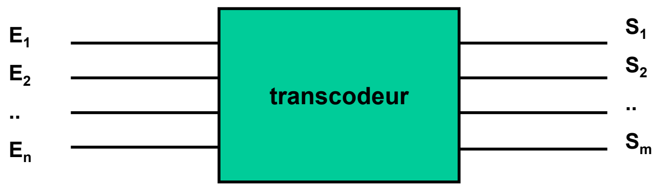
$$Y = \overline{I0}.(I1 + .\overline{I2}.I3)$$

226

Logique combinatoire et séquentielle

Le transcodeur

- ❑ C'est un circuit combinatoire qui permet de transformer un code X (sur n bits) en entrée en un code Y (sur m bits) en sortie.



227

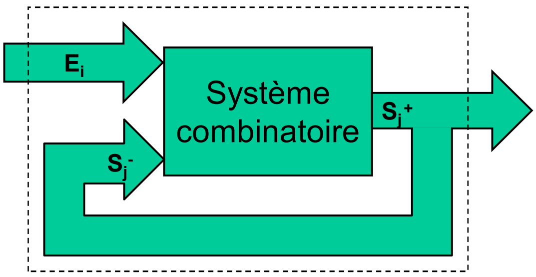
Logique combinatoire et séquentielle

Exemple : Transcodeur BCD/EXESS3

A	B	C	D		X	Y	Z	T
0	0	0	0		0	0	1	1
0	0	0	1		0	1	0	0
0	0	1	0		0	1	0	1
0	0	1	1		0	1	1	0
0	1	0	0		0	1	1	1
0	1	0	1		1	0	0	0
0	1	1	0		1	0	0	1
0	1	1	1		1	0	1	0
1	0	0	0		1	0	1	1
1	0	0	1		1	1	0	0
1	0	1	0		x	x	x	x
1	0	1	1		x	x	x	x
1	1	0	0		x	x	x	x
1	1	0	1		x	x	x	x
1	1	1	0		x	x	x	x
1	1	1	1		x	x	x	x

Logique combinatoire et séquentielle

Logique combinatoire



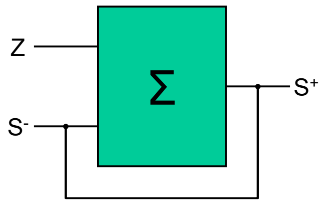
Système séquentiel

229

Logique combinatoire et séquentielle

Systèmes asynchrones

- Les sorties évoluent à la suite d'un changement de combinaison des entrées, ce qui provoque des états transitoires, des retards de durées différentes et des risques d'instabilité.

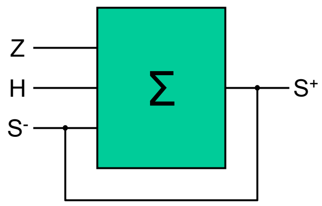


230

Logique combinatoire et séquentielle

Systèmes synchrones

- L'évolution des sorties est synchronisée par une commande externe appelée horloge afin d'éviter les multiples états transitoires notamment lorsque des entrées changent d'état simultanément.



231

Logique combinatoire et séquentielle

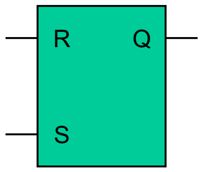
La bascule RS asynchrone

- La bascule RS asynchrone possède une entrée R (Reset) de mise à zéro, une entrée S (Set) de mise à 1 et une sortie Q.
- L'état R=S=0 (mode mémoire) maintient l'état de la sortie. L'état R=S=1 (mode interdit) est interdit car il conduit à mettre simultanément la sortie à 1 et à 0.

table de fonctionnement :

R	S	Q^+	
0	0	Q^-	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	Φ	Interdit

symbole :



Logique combinatoire et séquentielle

Réalisation 1 de bascule RS asynchrone

table de vérité :

R	S	Q^-	Q^+	
0	0	0	0	Mémoire
0	0	1	1	
0	1	0	1	Mise à 1
0	1	1	1	
1	0	0	0	Mise à 0
1	0	1	0	
1	1	0	Φ	Interdit
1	1	1	Φ	

tableau de Karnaugh :

RS \ Q^-	0	1
00	0	1
01	1	1
11	Φ	Φ
10	0	0

équation logique :

$$\boxed{Q^+ = S + Q^- \cdot \overline{R}}$$

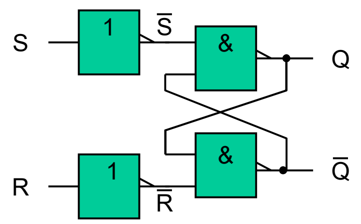
Logique combinatoire et séquentielle

Réalisation 1 de bascule RS asynchrone

logigramme :

$$Q^+ = \overline{\overline{S + Q^-\cdot\overline{R}}}$$

$$Q^+ = \overline{\overline{S}\cdot\overline{Q^-\cdot\overline{R}}}$$



- ❑ Cette bascule RS est prioritaire au 1 car, pour la combinaison R=S=1, la sortie Q est mise à 1 (les Φ ayant été fixés à 1 pour la simplification de Q).
- ❑ Remarque : le logigramme fait apparaître une sortie supplémentaire égale au complément de la sortie Q uniquement si la combinaison R=S=1 n'apparaît pas.

Logique combinatoire et séquentielle

Réalisation 2 de bascule RS asynchrone

table de vérité :

R	S	Q^-	Q^+	
0	0	0	0	Mémoire
0	0	1	1	
0	1	0	1	Mise à 1
0	1	1	1	
1	0	0	0	Mise à 0
1	0	1	0	
1	1	0	Φ	Interdit
1	1	1	Φ	

tableau de Karnaugh :

RS \ Q^-	0	1
00	0	1
01	1	1
11	Φ	Φ
10	0	0

équation logique :

$$\boxed{Q^+ = \overline{R}\cdot Q^- + S}$$

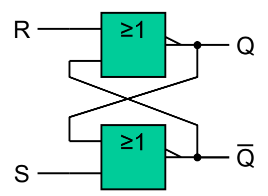
Logique combinatoire et séquentielle

Réalisation 2 de bascule RS asynchrone

logigramme :

$$Q^+ = \overline{\overline{\overline{R} \cdot Q^- + S}}$$

$$Q^+ = \overline{R + \overline{Q^- + S}}$$

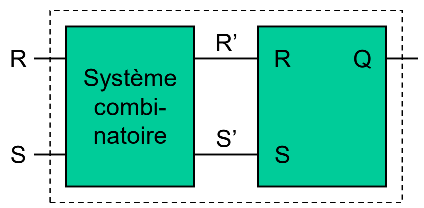


- ❑ Cette bascule RS est prioritaire au 0 car, pour la combinaison R=S=1, la sortie Q est mise à 0 (les Φ ayant été fixés à 0 pour la simplification de Q).
- ❑ Remarque : le logigramme fait apparaître une sortie supplémentaire égale au complément de la sortie Q uniquement si la combinaison R=S=1 n'apparaît pas.

Logique combinatoire et séquentielle

Cas des états interdits

- ❑ Afin de conserver une sortie complémentaire quel que soit la combinaison d'entrée, il convient de remplacer la combinaison R=S=1 par une autre combinaison en utilisant un circuit combinatoire selon le principe suivant :



Logique combinatoire et séquentielle

Mise à 1

- Cas R=S=1 ramené au cas R=0 et S=1

table de vérité :

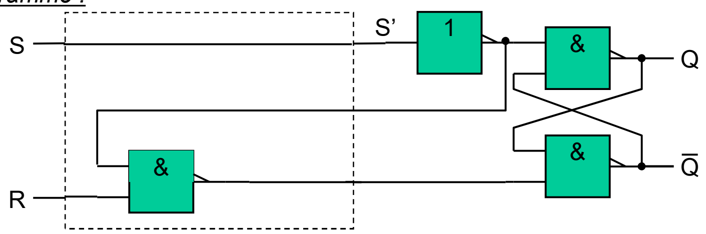
R	S	R'	S'
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	1

équations logiques :

$$S' = S$$

$$R' = R \cdot \overline{S} = \overline{\overline{R \cdot \overline{S}}}$$

logigramme :



Logique combinatoire et séquentielle

mise à 0

- Cas R=S=1 ramené au cas R=1 et S=0 ()

table de vérité :

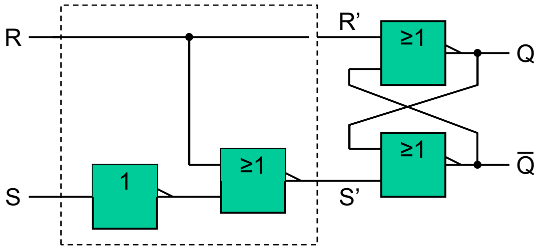
R	S	R'	S'
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

équations logiques :

$$R' = R$$

$$S' = \overline{R} \cdot S = \overline{R + \overline{S}}$$

logigramme :



Logique combinatoire et séquentielle

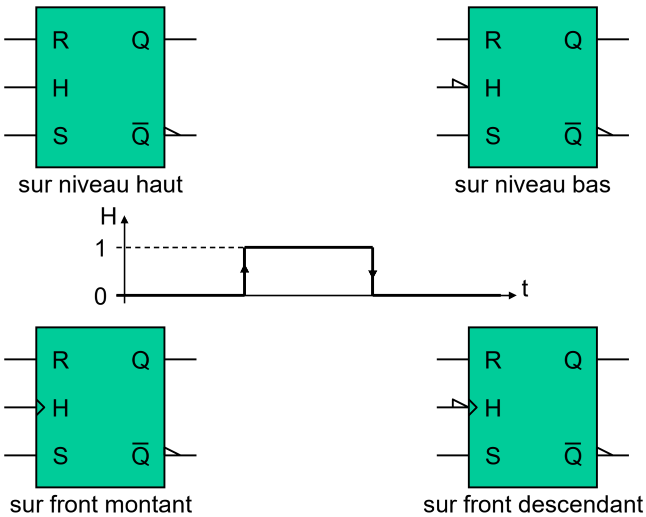
La bascule RS synchrone (RST ou RSH)

- La bascule RS synchrone possède une entrée R (Reset) de mise à zéro, une entrée S (Set) de mise à 1, une entrée d'horloge H et une sortie Q.
- La bascule RS synchrone fonctionne selon l'état de l'horloge :
  - si l'horloge est à 1 (niveau haut)
  - si l'horloge est à 0 (niveau bas)
  - si il y a un front montant sur l'horloge
  - si il y a un front descendant sur l'horloge
  - si il y a une impulsion sur l'horloge

240

Logique combinatoire et séquentielle

Modes de synchronisation des bascules RST



241

Logique combinatoire et séquentielle

La bascule RST synchronisée par le niveau haut de l'horloge :

table de fonctionnement :

H	R	S	Q^+	
0	Φ	Φ	Q^-	Mémoire
1	0	0	Q^-	
1	0	1	1	Mise à 1
1	1	0	0	Mise à 0
1	1	1	Φ	Interdit

- *Exercice : à partir de la table de vérité de cette bascule, déterminer l'équation de sa sortie et réaliser le logigramme avec des portes NAND uniquement.*

Logique combinatoire et séquentielle

Exercice:

H	Q^-	R	S	Q^+
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	Φ
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	Φ

HQ^- \ RS	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	Φ	0
10	0	1	Φ	0

$$Q^+ = Q^-\overline{R} + \overline{H}Q^- + HS$$

$$Q^+ = Q^-(\overline{R} + \overline{H}) + HS$$

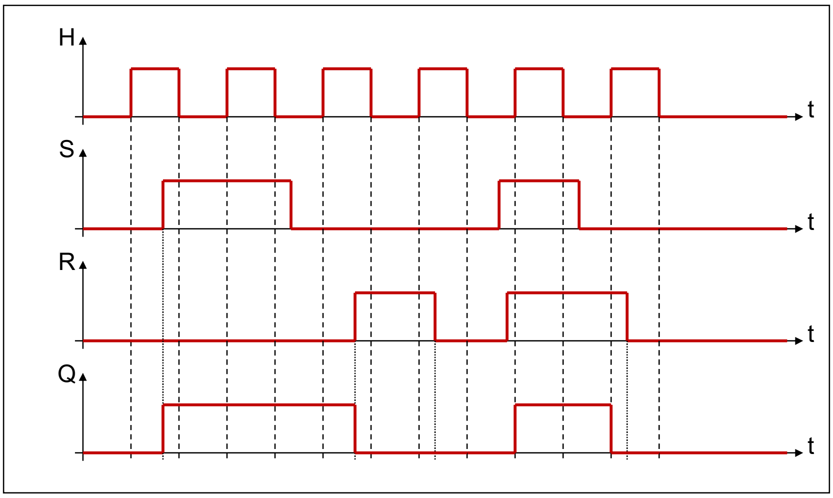
$$Q^+ = \overline{\overline{Q^-(\overline{RH}) + HS}}$$

$$Q^+ = \overline{\overline{Q^-(\overline{RH})}\;\overline{HS}}$$

Logique combinatoire et séquentielle

- **Exercice: Chronogram**

$Q^+ = \overline{\overline{Q^-\,(\overline{RH})}}\;\overline{\overline{HS}}$



Logique combinatoire et séquentielle

La bascule RST synchronisée par le front montant de l'horloge :

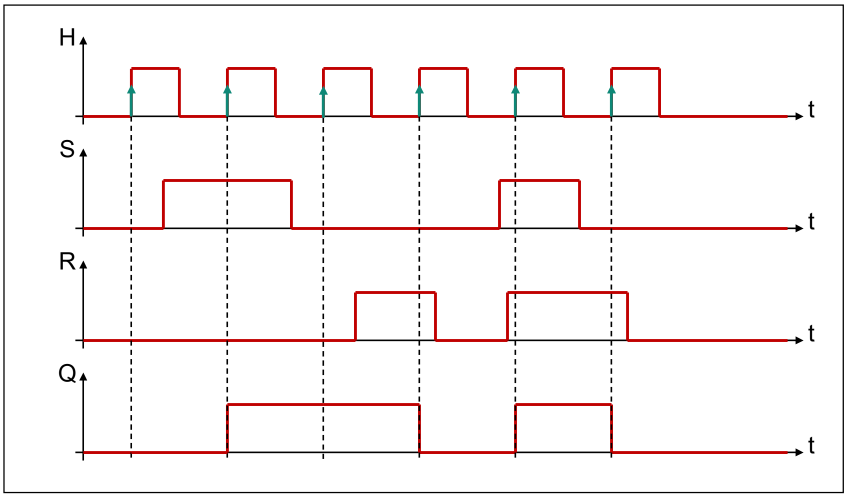
table de fonctionnement :

H	R	S	Q^+	
0	Φ	Φ	Q^-	
1	Φ	Φ	Q^-	Mémoire
↑	0	0	Q^-	
↑	0	1	1	Mise à 1
↑	1	0	0	Mise à 0
↑	1	1	Φ	Interdit

- **Réalisation** : la détection du front s'effectue par le jeu de 3 mémoires interne à la bascule ou par un circuit de dérivation du signal d'horloge.

Logique combinatoire et séquentielle

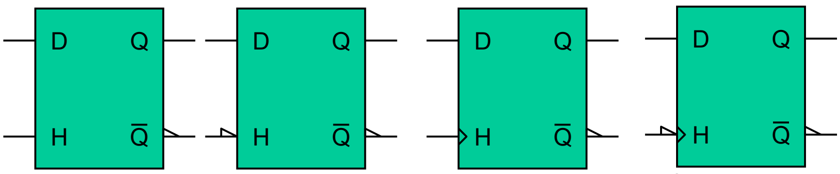
Réalisation: Chronogramme



Logique combinatoire et séquentielle

La bascule D

- La bascule D est une bascule synchrone qui possède une entrée de donnée D (Data), une entrée d'horloge H, une sortie Q et une sortie complément de Q.
- Le signal de synchronisation est actif :
  - soit sur un niveau (haut ou bas) de l'horloge (bascule D latch)
  - soit sur un front (montant ou descendant) de l'horloge (bascule D edge triggered)



sur niveau haut sur niveau bas sur front montant sur front descendant

Logique combinatoire et séquentielle

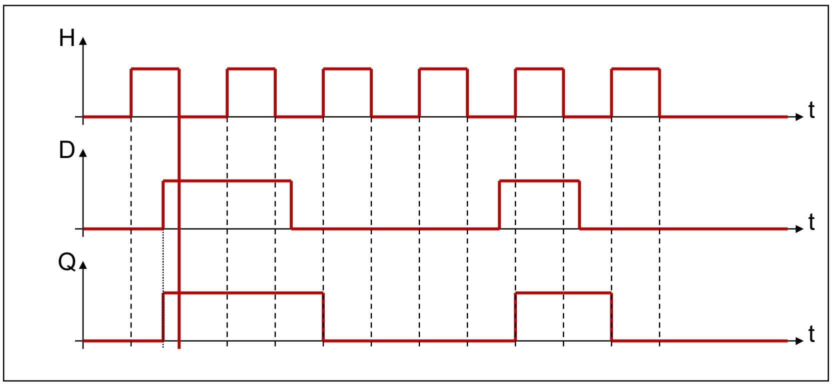
La bascule D latch :

- ❑ la sortie recopie l'entrée sur un niveau d'horloge. Sur l'autre niveau, la sortie est mémorisée.
  - Bascule D latch synchronisée par le niveau haut :

248

Logique combinatoire et séquentielle

La bascule D latch : Chronogramme



249

Logique combinatoire et séquentielle

La bascule D edge triggered :

- ❑ la sortie recopie l'entrée sur un front d'horloge sinon elle ne change pas d'état (maintien de l'état, mémorisation).
  - Bascule D synchronisée par le front montant (positive edge triggered):

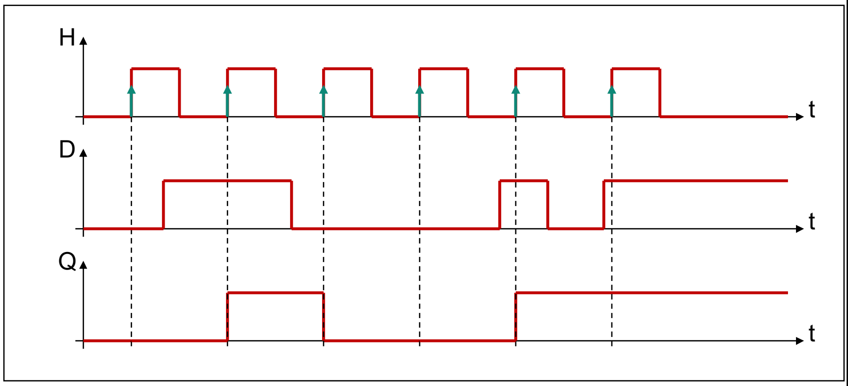
table de fonctionnement :

H	D	Q^+	
0	Φ	Q^-	Mémoire
1	Φ	Q^-	
↑	0	0	Recopie
↑	1	1	

250

Logique combinatoire et séquentielle

La bascule D latch : Chronogramme

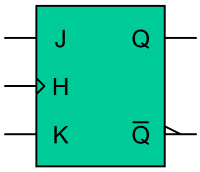


251

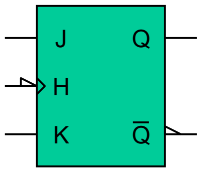
Logique combinatoire et séquentielle

La bascule JK

- ❑ La bascule JK est une bascule synchrone (le plus souvent sur front) qui possède une entrée J de mise à 1, une entrée K de mise à 0, une entrée d'horloge H, une sortie Q et une sortie complément de Q.
- ❑ Son fonctionnement diffère de celui d'une bascule RST pour la situation ambiguë R=S=1. Dans le cas J=K=1, la sortie est inversée.



bascule JK à déclenchement sur front montant



bascule JK à déclenchement sur front descendant

252

Logique combinatoire et séquentielle

Bascule JK à déclenchement sur front montant :

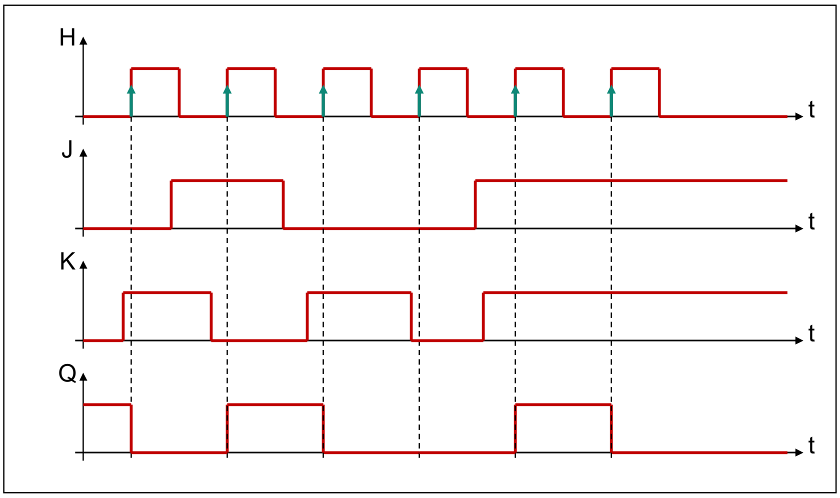
table de fonctionnement :

H	J	K	Q^+	
0	Φ	Φ	Q^-	
1	Φ	Φ	Q^-	Mémoire
↑	0	0	Q^-	
↑	0	1	0	Mise à 0
↑	1	0	1	Mise à 1
↑	1	1	$\overline{Q}^-$	Inversion

253

Logique combinatoire et séquentielle

Chronogramme :



254

Logique combinatoire et séquentielle

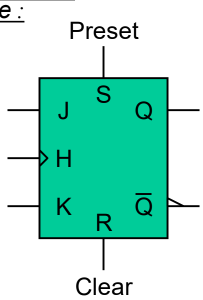
Entrées asynchrones :

- ❑ toutes les bascules synchrones commercialisées possèdent des entrées asynchrones de forçage de mise à 0 (R ou Clear) et de mise à 1 (S ou Preset) prioritaires sur toutes autres entrées.
  - Bascule JK à déclenchement sur front montant avec entrées de forçage non complémentées :

table de fonctionnement :

R	S	H	J	K	Q^+	
1	0	Φ	Φ	Φ	0	Forçage à 0
0	1	Φ	Φ	Φ	1	Forçage à 1
0	0	↑	0	0	Q^-	Mémoire
0	0	↑	0	1	0	Mise à 0
0	0	↑	1	0	1	Mise à 1
0	0	↑	1	1	$\overline{Q^-}$	Inversion
1	1	Φ	Φ	Φ	Φ	Interdit

symbole :



255

Logique combinatoire et séquentielle

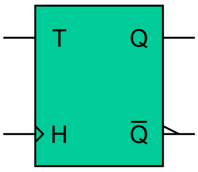
La bascule T

- La bascule T est une bascule synchrone qui possède une entrée de donnée T, une entrée d'horloge H, une sortie Q et une sortie complément de Q.
- Son fonctionnement est un cas particulier de la bascule JK ou les entrées J et K sont connectées ensemble (ou mises à 1).

table de fonctionnement :

H	T	Q^+	
0	Φ	Q^-	Mémoire
1	Φ	Q^-	
↑	0	Q^-	
↑	1	$\overline{Q^-}$	Inversion

symbole :



Logique combinatoire et séquentielle

La bascule T: Chronogramme :