

Polytech'Nantes

Département GE_3

Informatique Industrielle - partie I

- D. DELFIEU -

Reproduction interdite sans autorisation de l'auteur et de l'école



POLYTECH'
Premier réseau national
des écoles d'ingénieurs
polytechniques des universités

■ **Site de la Chantrerie**
Rue Christian Pauc - BP50609
44306 Nantes cedex 3 - France
Tél. +33 (0)2 40 68 32 00
Fax. +33 (0)2 40 68 32 32
www.polytech.univ-nantes.fr

■ **Site de Gavy**
Gavy Océanis - BP152
44603 St-Nazaire cedex - France
Tél. +33 (0)2 40 90 50 30
Fax. +33 (0)2 40 90 50 24



Logique

D. DELFIEU

January 3, 2023

1 / 178

Plan de la présentation

- 1 introduction
 - Les systèmes basés
 - L'arithmétique des systèmes basés
- 2 Logique Combinatoire
 - Fonctions logiques
 - Simplification des Fonctions Logiques
 - Règles générales de simplification
 - Règles de simplification arithmétiques
 - Les circuits Combinatoires
- 3 Logique Séquentielle
 - La notion de mémoire
 - Les différents types de bascules
 - La Bascule D positive edge triggered
 - La Bascule T
 - Les Maîtres esclaves
 - Les séquenceurs

2 / 178

Objectifs du cours

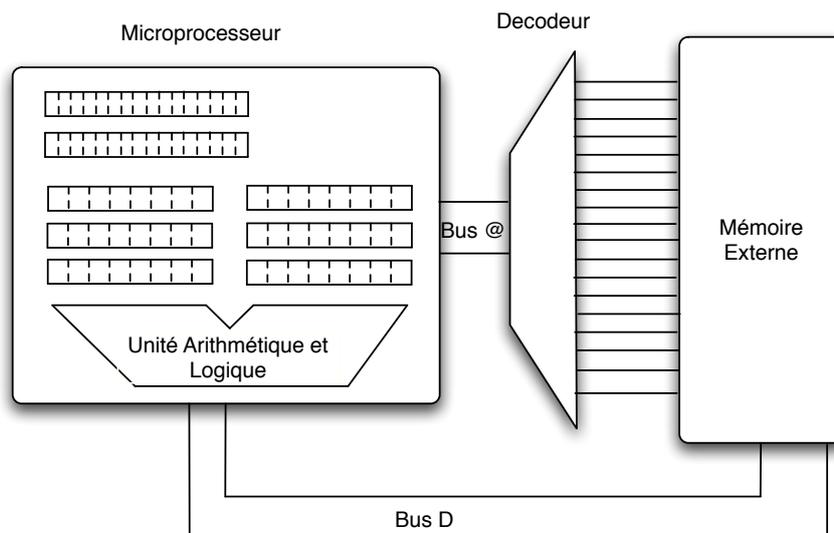
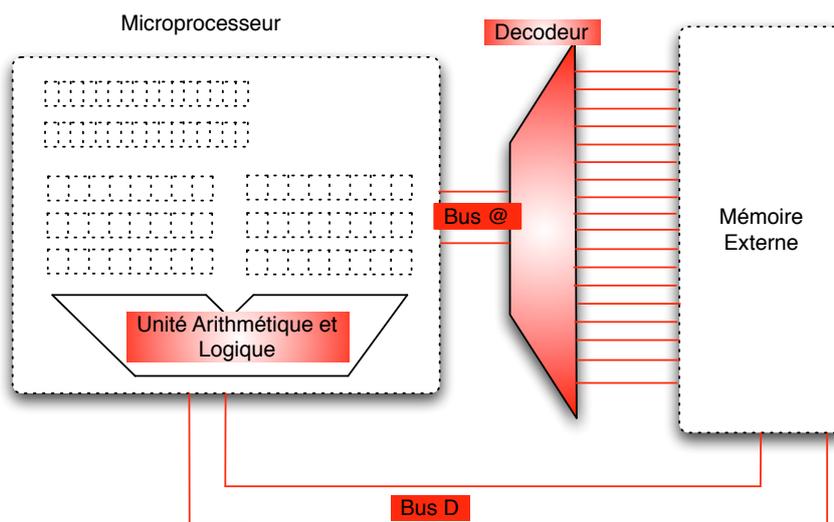


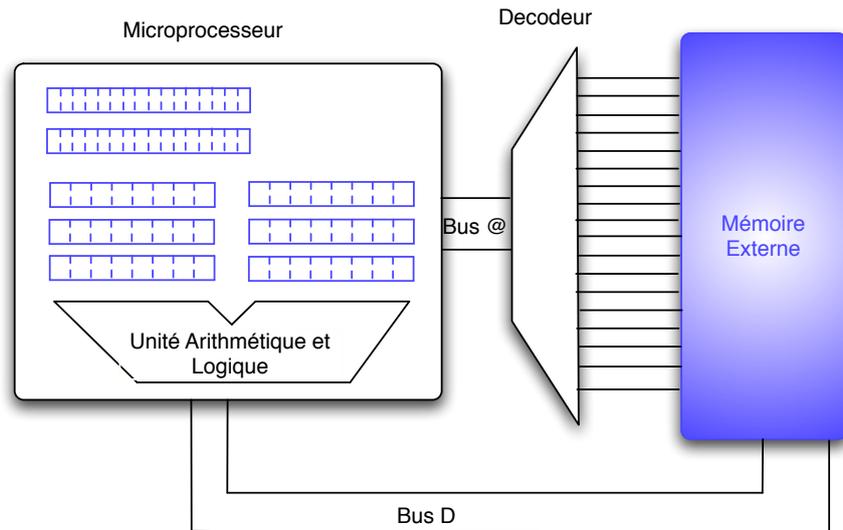
Figure: Objectifs du cours

Combinatoire



Composants relevant de la LOGIQUE COMBINATOIRE

Figure: Partie combinatoire



Composants relevant de la LOGIQUE SEQUENTIELLE

Figure: Partie séquentielle

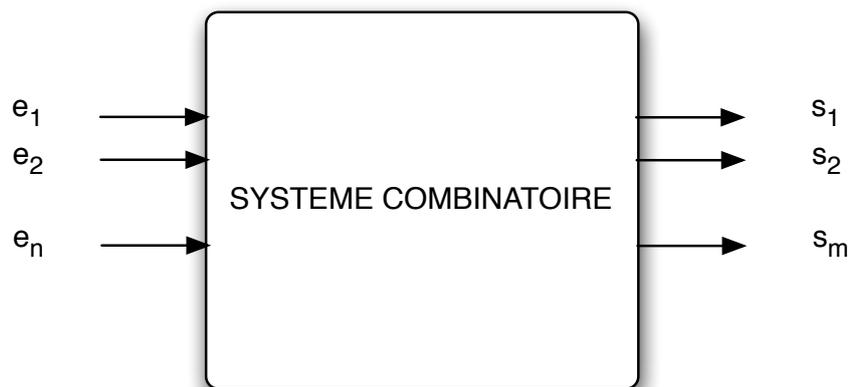
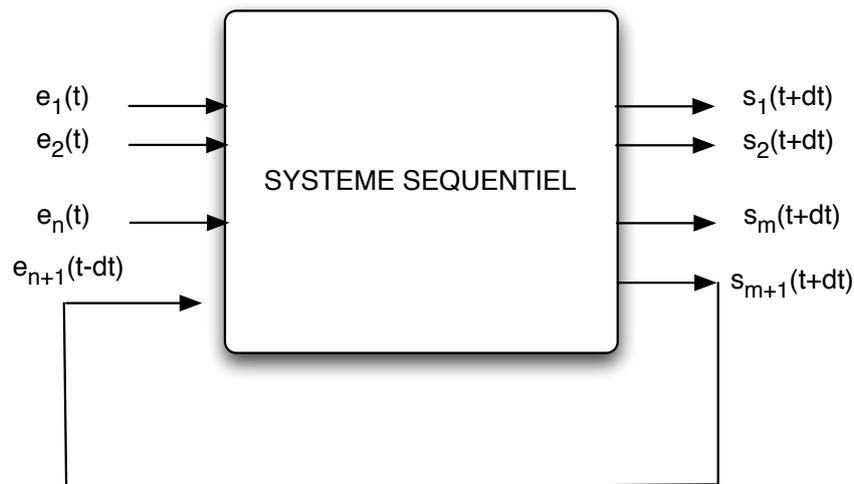


Figure: Système Combinatoire



Définition d'un système basé

Partie Entière (PE) :

- Base B : $0 \dots B - 1$
- Nombre N_B : juxtaposition de symboles exprimés en base B .
- $N_B = a_n a_{n-1} \dots a_0$
- $N_{10} = \sum_{i=0}^n a_i B^i$

Partie Fractionnaire (PF) :

- $N_B = 0, a_{-1} a_{-2} \dots a_{-m}$
- $N_{10} = \sum_{i=-m}^{-1} a_i B^i$

Méthode de conversion des PE base 10 \rightarrow base B

Méthode par soustraction

- 1 Soit la plus grande puissance, B^i et $a_j \in [0, B - 1]$ le plus grand entier tel que $a_j * B^i < N$
- 2 On pose a_j au rang i
- 3 On réalise $N = N - a_j * B^i$
- 4 On itère le processus à l'étape 1 avec la nouvelle valeur de N tant que N est supérieur à B
- 5 On dépose le dernier reste au rang zéro.

Exercice : 77_{10} en base 3 ?

Méthode par division

Diviser N autant de fois que nécessaire par B jusqu'à obtenir un quotient nul, on écrit alors les restes de la division dans l'ordre inverse où ils ont été obtenus (le premier reste est le poids faible du résultat).

Exercice : 77_{10} en base 3 ?

Méthode de conversion des PF base 10 \rightarrow base B

Méthode par multiplication

- 1 Multiplier successivement N par B ;
- 2 On soustrait à N sa partie entière qui devient un résultat que l'on écrit à droite du précédent résultat ;
- 3 On itère cet algorithme avec la nouvelle valeur de N ;
- 4 On arrête le processus lorsque l'on obtient une valeur de N nulle ou lorsque l'on a atteint un niveau de décimale correspondant à la précision exprimée. Sinon on reboucle à l'étape 1.

Exemple : $0,45_{10}$ en base 2 ?

Les autres conversions

- Base 2 et base 2^n

On regroupe par paquets de n bits et on convertit à l'intérieur de chaque paquets de n bits

Exemple : $0,01110011_2 \implies 0,011100110 \implies 0,346_8$

- base i vers j

Prendre la base relais 2 ou 10.

Représentation des nombre négatifs

- Signe et valeur absolue
- Complément à deux : CA_2
- Complément à un : CA_1

Signe et valeur absolue

7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0+	0000
0-	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

- Un bit de signe :
 - ▶ 1 pour "-"
 - ▶ 0 pour "+"
- Deux représentations pour le zéro :
 - ▶ 0000 pour 0⁺
 - ▶ 1000 pour 0⁻
- Nécessité d'implémenter une table de soustraction

Complément à 1 : Complément Restreint

7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0+	0000
0-	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000

Soit n le nombre de bits de représentation des nombres :

$$CA_{B-1}(N) = B^n - N - 1$$

Base 2 :

$$CA_1(N) = 2^n - N - 1$$

S'obtient facilement par inversion binaire du positif.

Complément à 1 : Exercices

En utilisant la formule précédente :

- Sur 4 bits, trouver la représentation en base 2 de -5 en complément restreint ?
- Sur 4 bits, trouver la représentation en base 2 de -1 en complément restreint ?

Complément à 2 : Complément Vrai

7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

Soit n le nombre de bits de représentation des nombres :

$$CA_B(N) = B^n - N$$

Base 2 :

$$CA_2(N) = 2^n - N$$

- S'obtient par inversion binaire du positif puis en rajoutant 1.
- "De la droite vers la gauche, reproduire les zéros, au premier "un" rencontré, le reproduire puis inverser tous les bits à sa gauche"

En utilisant la formule précédente :

- Sur 4 bits, trouver la représentation en base 2 de -3 en complément vrai ?
- Sur 4 bits, trouver la représentation en base 2 de -7 en complément vrai ?

Plage de représentation des nombres

- CA_1
 - ▶ plage des positifs : $2^{n-1} - 1 \dots 0^+$
 - ▶ plage des négatifs : $0^- \dots -(2^{n-1} - 1)$
- CA_2 :
 - ▶ plage des positifs : $2^{n-1} - 1 \dots 0$
 - ▶ plage des négatifs : $-1 \dots -(2^{n-1})$

Exemples

- Posons $n=8$:
- Plage positif : 0 ... +127
- Plage négatifs : -1 ... -128
- Représentation négative de 100 ? $100 = 01100100_2$
 - ▶ $CA_1(100) = 2^8 - 100_{10} - 1 = 256_{10} - 100_{10} - 1 = 155_{10} = 10011011_2$
10011011 s'obtient facilement par complément binaire de 01100100
 - ▶ $CA_2(100) = 2^8 - 100_{10} = 256_{10} - 100_{10} = 156_{10} = 10011011_2$
10011011 s'obtient moins facilement par complément binaire et addition de +1:
 $01100100 \xrightarrow{Comp.Bin.} 10011011 \xrightarrow{+1} 10011100$

Bilan

Complément à un :

- Deux zéros
- Une correction de +1 à réaliser dans certains cas
- facile à implémenter

Complément à deux :

- Un seul zéro,
- pas de correction
- plus difficile à implémenter
- dans les négatifs on a un nombre de plus.

Dépassement de capacité

Un débordement de capacité se produit lors de l'addition de deux nombres positifs ou lors de l'addition de deux nombres négatifs. Il se produit alors un changement de signe du résultat.

Exemple sur 4 bits :

$$\begin{array}{r} 0111 \\ +0001 \\ \hline 1000 \end{array} \begin{array}{l} (+7) \\ (+1) \\ (-8!!!) \end{array} \qquad \begin{array}{r} 1000 \\ +1110 \\ \hline 0111 \end{array} \begin{array}{l} (-7) \\ (-2) \\ (+7!!) \end{array}$$

Définition

Il y a débordement lorsque les retenues entrantes et sortantes dans l'addition du bits de signe sont différentes.

Le Codage

Il y a deux grandes familles de codes :

Code Pondéré

Un code est dit pondéré si la position de chaque symbole dans chaque mot correspond à un poids fixé. En multipliant chaque symbole par son poids et en additionnant le tout, on obtient la conversion en décimal.

Code Non Pondéré

Les autres codes pour lesquels on ne peut repérer de poids sont appelés non pondérés. Ils sont définis par des tables de correspondances ou par le biais de symétrie et/ou de propriétés logiques ou arithmétiques.

Le Codage

- Code Pondéré : binaire, octal, hexadécimal, Décimal Codé Binaire
Le "Décimal Codé Binaire" (BCD ou Binary Coded Decimal) est un code dans lequel chaque chiffre de la représentation décimale est codé sur un groupe de 4 bits.
 - ▶ Exemple : 1789 se code 0001 0111 1000 1001
 - ▶ Avantage : Affichage décimal grandement facilité
 - ▶ Inconvénient : Code redondant 6 combinaisons sur 16 ne sont pas utilisées

Remarque : 1789 prend 13 bits en BCD, seulement 11 en binaire naturel.

- Code Non Pondéré : Code excess 3
comme BCD, ce code permet une transcription rapide en décimal. Il permet de plus une précision infinie en arithmétique (limité par les temps de calcul) au dépend d'une représentation qui gaspille des combinaisons.
Il apporte par rapport au BCD une rapidité par rapport à la soustraction.
- Code Non Pondéré : Code de Gray

Codage de Gray

Code réfléchi, circulaire, distance de hamming de 1:

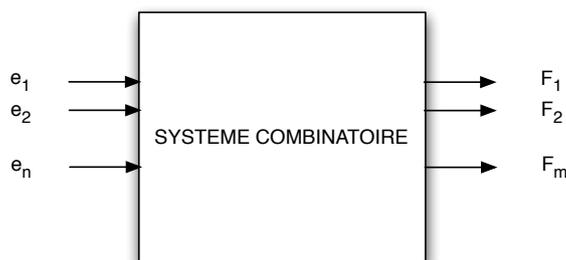
				0 0		0 0 0
				0 1		0 0 1
0		0 0		1 $\bar{1}$		0 1 $\bar{1}$
1		0 1		1 0		0 1 0
$\bar{1}$	Rem.	1 $\bar{1}$	Sym.	$\bar{1}$ 0	Rem.	1 $\bar{1}$ 0
0		1 0		1 1		1 1 1
				0 1		1 0 1
				0 0		1 0 0

Variable logique

- Une variable logique est une variable qui prend ses valeurs dans $[0, 1]$.
- $[0, 1]$ est un domaine qui peut s'interpréter comme $[Faux, Vrai]$ et également comme $[0V, 5V]$.
- L'algèbre de Boole (Georges Boole 1815 - 1864) est une algèbre binaire n'acceptant que deux valeurs numériques : 0 et 1. Cette algèbre est définie par la donnée d'un ensemble non vide muni de trois lois de composition interne : *ET*, *OU*, *NON* satisfaisant à certain nombre de propriétés (commutativité, distributivité...).

Fonction logique

Un système combinatoire peut se définir par un ensemble de fonctions logiques :



Fonction logique

Une fonction logique F_i est une fonction de n variables qui est définie par sa valeur dans $[0, 1]$ pour les 2^n combinaisons de ces variables d'entrées. Ces 2^n définitions constituent ce que l'on appelle la table de vérité d'une fonction.

- Traduire un problème en équations logiques ;
- Simplifier ces équations ;
- Implémenter ces équations à l'aide de portes et/ou de composants logiques.

Traduction d'un problème en équations logiques

Un problème de type combinatoire - c.a.d . qui ne requiert pas le concept de mémoire - peut généralement s'exprimer dans le cas d'une algèbre :

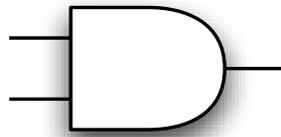
$$\mathcal{A} = (. , + , ^{-}, (0, 1))$$

ET

Noté "." on trouve aussi \wedge , $\&$:

- il a un élément neutre noté 1: $1.x = x$
- un élément absorbant noté 0 : $0.x = 0$
- il est commutatif $x.y = y.x$
- il est associatif $x.(y.z) = (x.y).z$
- il est distributif sur le OU : $x.(y + z) = xy + xz$
- il est idempotent : $x = x.x.....x$

Son symbole électronique est :

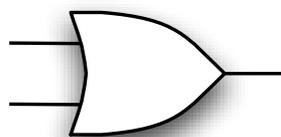


OU

Noté "+" on trouve aussi \vee :

- il a un élément neutre noté 0: $0 + x = x$
- un élément absorbant noté 1 : $1 + x = 1$
- il est commutatif $x + y = y + x$
- il est associatif $x + (y + z) = (x + y) + z$
- il est distributif sur le ET : $x + (y.z) = (x + y).(x + z)$
- il est idempotent : $x = x + x.... + x$

Son symbole électronique est :

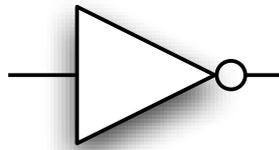


NON

Noté $\bar{}$ on trouve aussi \neg , c'est une fonction unaire.

- il est involutif $\overline{\overline{x}} = x$
- $x + \bar{x} = 1$
- $x \cdot \bar{x} = 0$
- théorème de Morgan (Augustus De Morgan 1806-1871)
$$\overline{x \vee y \vee z \vee \dots} = \bar{x} \wedge \bar{y} \wedge \bar{z} \wedge \dots$$
$$\overline{x \wedge y \wedge z \wedge \dots} = \bar{x} \vee \bar{y} \vee \bar{z} \vee \dots$$

Son symbole électronique est :

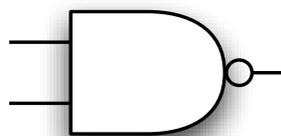


NAND

Noté $\overline{a.b}$

- il n'est pas associatif : $\overline{\overline{a.b.c}} \neq \overline{\overline{a.b}.c}$
- **NAND** est un opérateur logique COMPLET :
 - ▶ On peut réaliser le NOT avec un NAND : $\bar{a} = \overline{a.1}$
 - ▶ On peut réaliser le OU avec un NAND : $a + b = \overline{\overline{a.1.b}.1}$
 - ▶ On peut réaliser le ET avec un NAND : $a.b = \overline{\overline{a.b}.1}$

Son symbole électronique est :

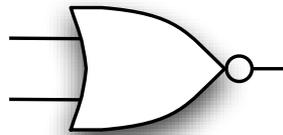


NOR

Noté $\overline{a + b}$

- il n'est pas associatif : $\overline{\overline{a + b} + c} \neq \overline{a + \overline{b + c}}$
- *NOR* est un opérateur logique COMPLET :
 - ▶ On peut réaliser le NOT avec un NOR : $\bar{a} = \overline{a + 0}$
 - ▶ On peut réaliser le ET avec un NOR : $a.b = \overline{\overline{a + 0} + \overline{b + 0}}$
 - ▶ On peut réaliser le OU avec un NOR : $a + b = \overline{\overline{a + b} + 0}$

Son symbole électronique est :

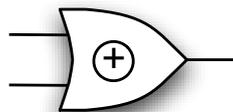


OU EXCLUSIF : XOR

Noté $a \oplus b = a.\bar{b} + \bar{a}.b$

- il est associatif : $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- *XOR* n'est pas un opérateur logique COMPLET :
 - On ne peut réaliser que le NOT avec un XOR : $\bar{a} = a \oplus 1$

Son symbole électronique est :



IMPLIQUE : \Rightarrow

Noté $a \Rightarrow b = \bar{a} + b$

a	b	$a \Rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Equivalence (\Leftrightarrow) : noté en logique \odot

Noté $a \Leftrightarrow b = a.b + \bar{a}.\bar{b} = \overline{a \oplus b} = a \odot b$

a	b	$a \Leftrightarrow b$
0	0	1
0	1	0
1	0	0
1	1	1

- $F(x, y) = x + \bar{x}.\bar{y} = x + \bar{y}$
- $F(x, y) = x + \bar{x}.y = x + y$
- $F(x, y) = x + x.y = x$
- $f(a, b, c) = \bar{a}\bar{b}.c + \bar{a}.b.c + a(\bar{b}\bar{c} + b\bar{c} + \bar{b}c + bc) = a + c$

Simplification

Simplification de fonction logique

C'est réduire la complexité d'une fonction logique et donc diminuer le coût et la consommation électrique du système électronique qui implémente la fonction.

Représentation d'une fonction logique

- Fonction logique ou représentation algébrique.
- Table de vérité
- Table de Karnaugh

Représentation Algébrique d'une fonction logique

- \exists infinité de formes équivalente à f :
 - ▶ Forme disjonctive : Opérateurs de plus haut niveau sont des *OU*
 - ▶ Forme conjonctive : Opérateurs de plus haut niveau sont des *ET*
- On peut se ramener à une forme unique : Forme Canonique

Théorème de décomposition de Shannon

Shannon

Toute fonction logique peut se décomposer par rapport à l'une de ces variables sous la forme d'une somme de deux produits logiques tel que :

$$F(x, y, \dots, z) = x.F(1, y, \dots, z) + \bar{x}.F(0, y, \dots, z)$$

Toute fonction logique peut se décomposer par rapport à l'une de ces variables sous la forme d'un produit de deux sommes logiques tel que :

$$F(x, y, \dots, z) = [\bar{x} + F(1, y, \dots, z)].[x + F(0, y, \dots, z)]$$

Démonstration du théorème de Shannon

Pour la première assertion posons $x=1$, on a :

$$\begin{aligned} F(1, y, \dots, z) &= 1.F(1, y, \dots, z) + \bar{1}.F(0, y, \dots, z) \\ &= 1.F(1, y, \dots, z) + 0.F(0, y, \dots, z) \\ &= F(1, y, \dots, z) \text{ ce qui est VRAI} \end{aligned}$$

Posons $x=0$, on a :

$$\begin{aligned} F(0, y, \dots, z) &= 0.F(1, y, \dots, z) + \bar{0}.F(0, y, \dots, z) \\ &= 1.F(0, y, \dots, z) \\ &= F(0, y, \dots, z) \text{ ce qui est VRAI} \end{aligned}$$

Pour toutes les valeurs de x , la première assertion est valide. On fera un raisonnement similaire pour la seconde assertion.

Exemple Décomposition disjonctive de Shannon

Soit la fonction F : définie par $F(x, y) = x + \bar{x}.\bar{y}$

$$\begin{aligned} F(x, y) &= x.F(1, y) + \bar{x}.F(0, y) \\ &= x.[y.F(1, 1) + \bar{y}.F(1, 0)] + \bar{x}.[y.F(0, 1) + \bar{y}.F(0, 0)] \\ &= x.y.1 + x.\bar{y}.1 + \bar{x}.y.0 + \bar{x}.\bar{y}.1 \\ &= x.y + x.\bar{y} + \bar{x}.\bar{y} \end{aligned}$$

Lorsque $F(.,.) = 0$, il y a une absorption du terme.

Remarques

Appliquer le théorème de Shannon pour la forme disjonctive revient, in fine, à la lecture de la table de vérité sur les "uns".

Exemple Décomposition conjonctive de Shannon

Soit la fonction F : définie par $F(x, y) = x + \bar{x}.\bar{y}$

$$\begin{aligned} F(x, y) &= [\bar{x} + F(1, y)].[x + F(0, y)] \\ &= [\bar{x} + (\bar{y} + F(1, 1))(y + F(1, 0))][x + (\bar{y} + F(0, 1))(y + F(0, 0))] \\ &= [\bar{x} + (\bar{y} + 1)(y + 1)][x + (\bar{y} + 0)(y + 1)] \\ &= [\bar{x} + 1] [x + \bar{y}] \\ &= 1 . [x + \bar{y}] \end{aligned}$$

Lorsque $F(.,.) = 1$, il y a une absorption du terme.

Remarques

Appliquer le théorème de Shannon pour la forme conjonctive revient, in fine, à la lecture de la table de vérité sur les "zeros", en inversant termes et opérateurs.

Table de vérité et Shannon en forme disjonctive

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

- Lecture sur les "uns".
- "zéro" est absorbant et élimine les autres lignes.
- Les termes sont associés en conjonction, les lignes sont en disjonction,
- $f(a, b, c) = \bar{a}\bar{b}.c + \bar{a}.b.c + a.\bar{b}\bar{c} + a\bar{b}c + a.b\bar{c} + a.bc$

Table de vérité et Shannon en forme Conjonctive

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

- Lecture sur les "zéros".
- "un" est absorbant et élimine les autres lignes.
- Les termes inversés sont associés en disjonction, les lignes sont en conjonction,
- $f(a, b, c) = (a + b + c).(a + \bar{b} + c)$

Table de vérité

- Une table de vérité sert à exprimer exhaustivement un cahier des charges.
- Une table de vérité est composée de deux parties : les combinaisons et la valeur de vérité de la fonction.
- Partie combinaison on exhibe l'ensemble 2^n combinaisons correspondant aux n variables.
- Si une combinaison est interdite ou si la valeur de la fonction est indifférente pour cette combinaison :
Alors la valeur de vérité de la fonction reçoit *.
- La valeur de vérité de la fonction est exprimée sur 0, 1, *.
- * signifie que la valeur vaut un ou zéro.

Règles de simplification

- Tables de vérité, tables de karnaugh
- Simplification algébrique

Karnaugh

Construction d'une table de Karnaugh pour une fonction à n variables :

- Partager l'ensemble des variables en 2 sous-ensembles s_1 de dimension n_1 et s_2 de dimension n_2 avec $n_1 + n_2 = n$.
- Ecrire un tableau avec 2^{n_1} lignes et 2^{n_2} colonnes.
- Construire un codage de Gray à n_1 et à n_2 variables, que l'on placera dans la première colonne et dans la première ligne du tableau.
- A partir de la table de vérité, reporter la (les) valeur(s) de vérité de(s) la fonction(s) à partir des combinaisons des variables d'entrée du tableau.

Exemple : Table de vérité et table de Karnaugh.

Exemple :

$$f(a, b, c) = \bar{a}\bar{b}.c + \bar{a}.b.c + a(\bar{b}\bar{c} + b\bar{c} + \bar{b}c + bc)$$

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

Sous-ensembles :

$$s_1 = c, s_2 = (a, b)$$

$f(a,b,c)$

		a				
		b		c		
c	a	b	0 0	0 1	1 1	1 0
	0	0	1	0	0	1
1	2	3	1	1	1	1

Règles de simplification

- Dans une table de karnaugh on regroupe deux paquets de 2^n "uns" pour former un paquet de 2^{n+1} variables.
- Le principe de regroupement de 2 paquets X est basé sur le théorème suivant :

$$X.\bar{x} + X.x = X$$

- Ainsi un regroupement de 2^n variables représente n applications SUCCESSIVES de ce théorème.

Règles de simplification

D'un point de vue graphique les regroupements peuvent se faire suivant les critères suivants :

- 2 cases adjacentes : Le Code de Gray assure que 2 codes adjacents ont une distance de Hamming de 1 ;
- un axe de symétrie partage la table de Karnaugh en deux parties égales ;
- 2 blocs de 2^n cases peuvent être regroupés si ils présentent un axe de symétrie ;
- Le Code de Gray étant circulaire les colonnes et les lignes extrêmes sont adjacentes.

Règles de simplification

- Commencer par les "uns" qui n'ont qu'une seule façon de se regrouper
- Faire les plus grands regroupements possibles

Regroupement de cases adjacentes

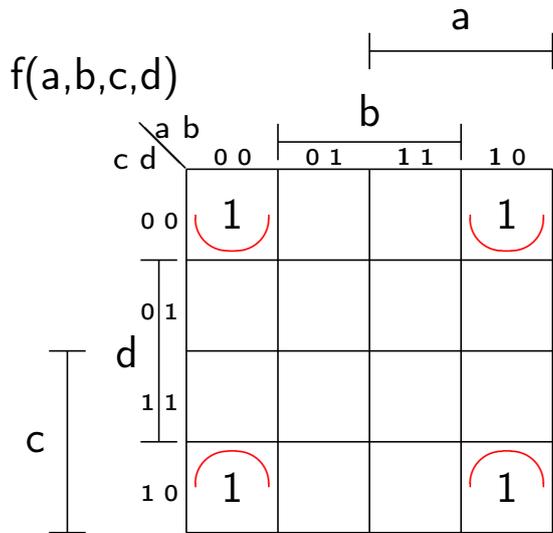
$f(a,b,c,d)$

		a			
		b			
c \ d	a \ b	0 0	0 1	1 1	1 0
d	0 0				
	0 1	1	1		
	1 1				
	1 0				

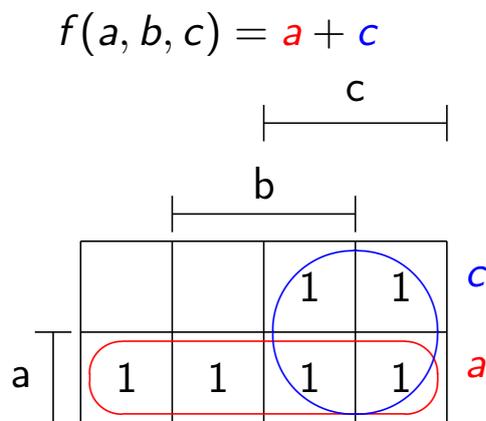
$f(a,b,c,d)$

		a			
		b			
c \ d	a \ b	0 0	0 1	1 1	1 0
d	0 0		1		
	0 1		1		
	1 1				
	1 0				

Regroupement de cases adjacentes

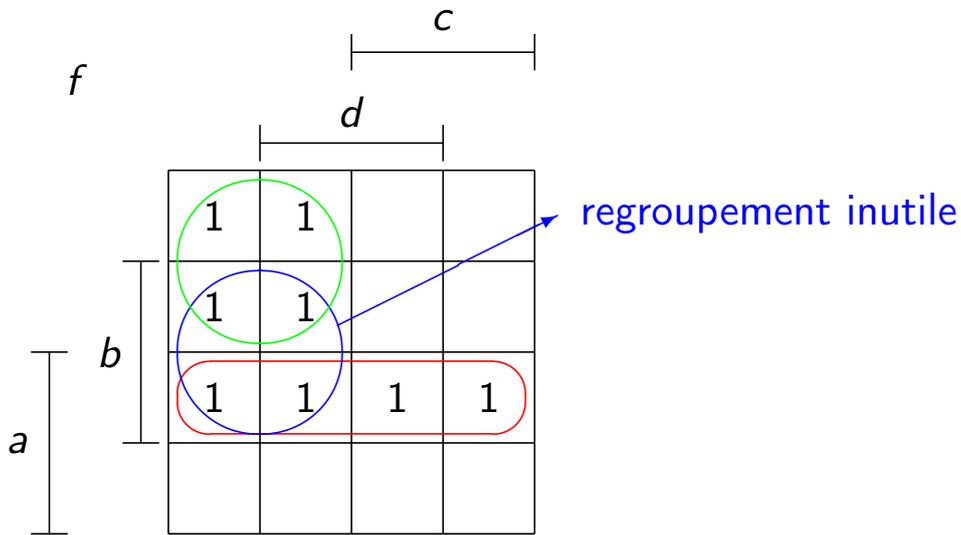


Regroupements de blocs de 2^n cases



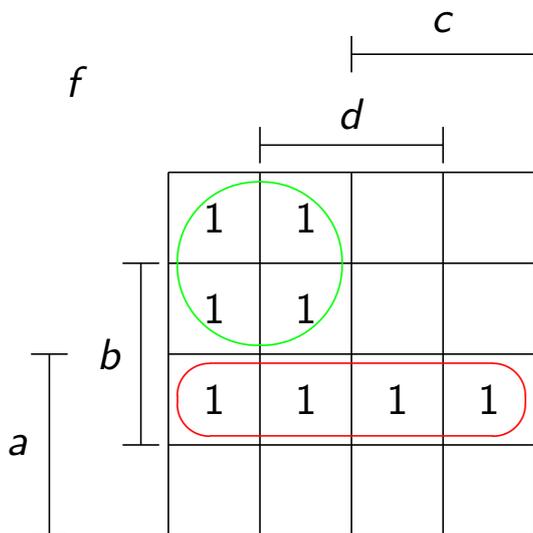
Regroupements de blocs de 2^n cases

$$f(a, b, c, d) = ab + \bar{c}b + \bar{c}\bar{a}$$

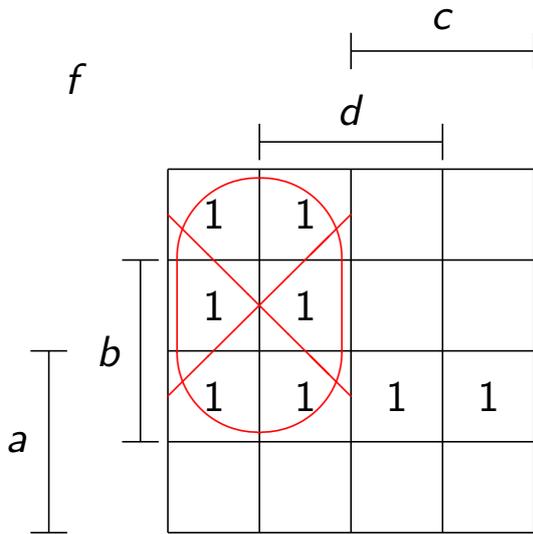


Regroupements de blocs de 2^n cases

$$f(a, b, c, d) = ab + \bar{c}\bar{a}$$



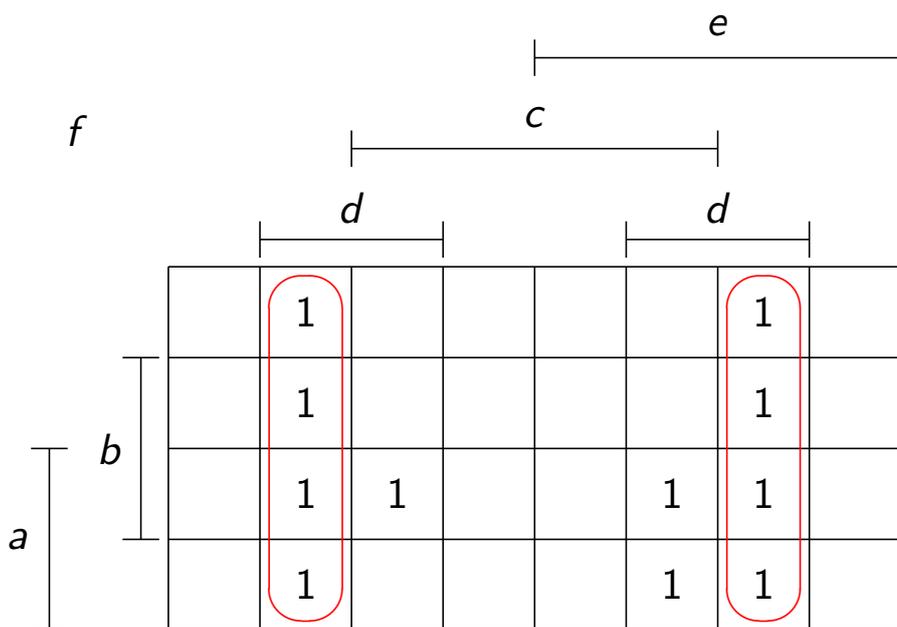
Attention !



Attention
Ce regroupement de six est interdit. Six n'est pas une puissance de 2.

Regroupements et axes de symétrie

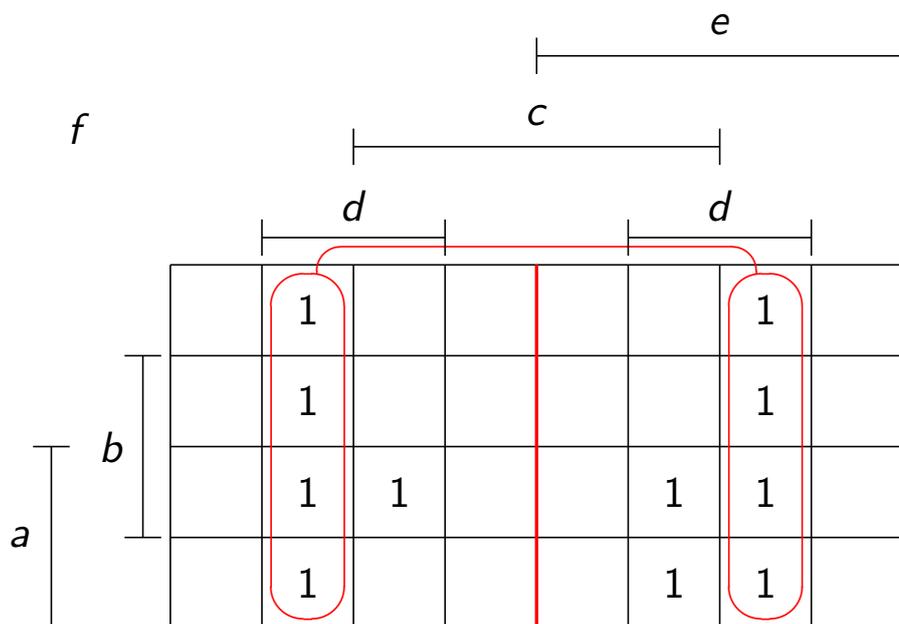
$$f(a, b, c, d, e) = \bar{c}d\bar{e} + \bar{c}de$$



Existe-t-il un axe de symétrie ?

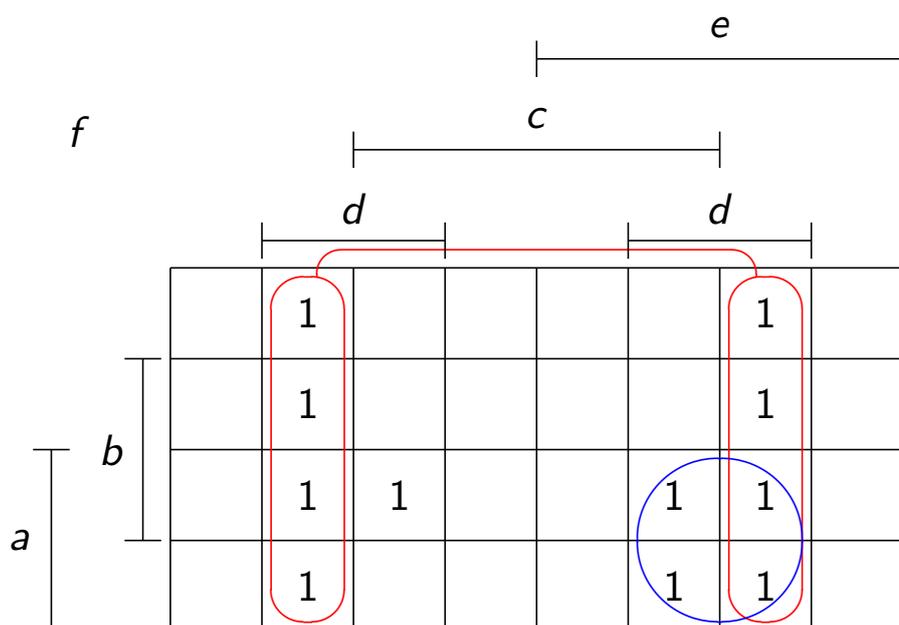
Regroupements et axes de symétrie

$$f(a, b, c, d, e) = \bar{c}d$$



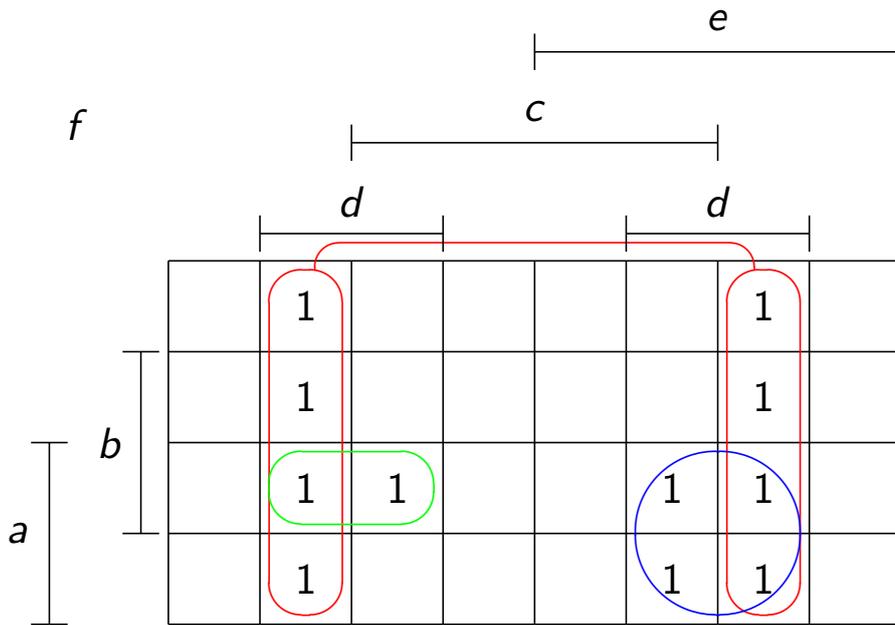
Regroupements et axes de symétrie

$$f(a, b, c, d, e) = \bar{c}d + ade$$



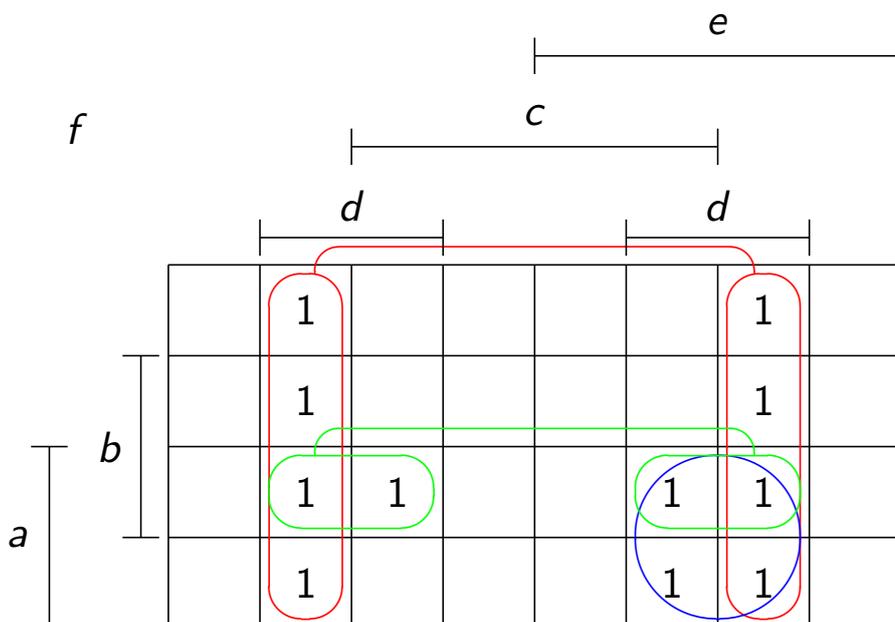
Regroupements et axes de symétrie

$$f(a, b, c, d, e) = \bar{c}d + ade + abd\bar{e}$$



Regroupements et axes de symétrie

$$f(a, b, c, d, e) = \bar{c}d + ade + abd$$



Cas des fonctions incomplètement définies

Combinaisons impossibles ou indifférentes

Si certaines combinaisons sont impossibles ou sont indifférentes, ces combinaisons sont appelées disponibles à la simplification et sont notées *.

Remarque :

Dans le cas où la combinaison est qualifiée d'impossible dans le cahier des charges et qu'il s'avère qu'elle se produit quand même (perturbation électromagnétique par exemple), cela provoquera un aléa dont la fréquence et la dangerosité nous amènera alors à revoir le cahier des charges et la réalisation du circuit électronique.

Cas des fonctions incomplètement définies : exemple

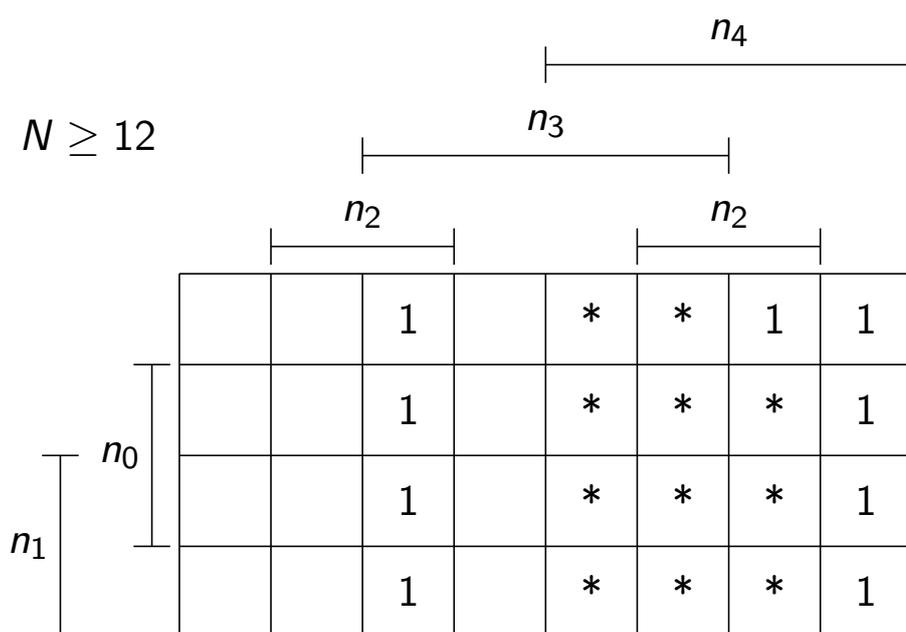
Cahier des charges

Soit N un chiffre décimal représentant une note sur 20 traduit en binaire. Donner la fonction logique qui permet de tester si $N \geq 12$.

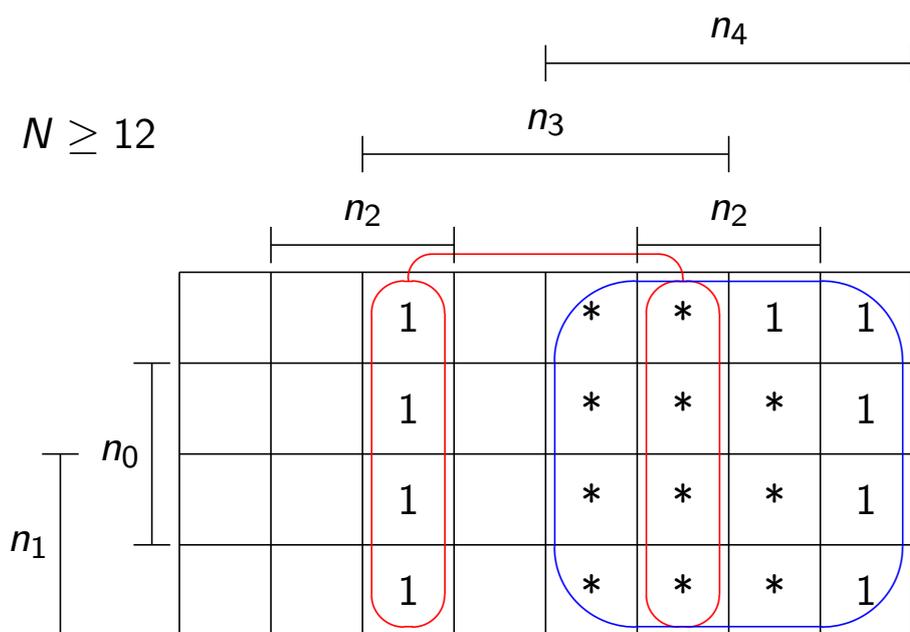
La table de vérité de $N \geq 12$

N	n_4	n_3	n_2	n_1	n_0	f
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	0
10	0	1	0	1	0	0
11	0	1	0	1	1	0
12	0	1	1	0	0	1
13	0	1	1	0	1	1
14	0	1	1	1	0	1
15	0	1	1	1	1	1
16	1	0	0	0	0	1
17	1	0	0	0	1	1
18	1	0	0	1	0	1
19	1	0	0	1	1	1
20	1	0	1	0	0	1
21	1	0	1	0	1	*
...	*
31	1	1	1	1	1	*

La table de Karnaugh de $N \geq 12$



La table de Karnaugh de $N \geq 12$



$$N \geq 12 = n_4 + n_2 \cdot n_3$$

La règle de remplacement

$$E + F = E + G \iff F \leq G \leq E + F$$

Exemple :

$$\begin{aligned} E(x_1, x_2) &= x_1 \\ F(x_1, x_2) &= \bar{x}_1 \cdot x_2 \\ G(x_1, x_2) &= x_2 \end{aligned}$$

x_1	x_2	E	F	G	E+F
0	0	0	0	0	0
0	1	0	1	1	1
1	0	1	0	0	1
1	1	1	0	1	1

D'où le nouveau théorème :

$$x_1 + \bar{x}_1 \cdot x_2 = x_1 + x_2$$

La règle de simplification

$$E + \bar{E}.F = E + F$$

Règle duale :

$$E.(\bar{E} + F) = E.F$$

La règle d'absorption

$$E + EF = E$$

Règle duale :

$$E.(E + F) = E$$

La règle du consensus

$$E.F + \bar{E}.G + FG = E.F + \bar{E}.G$$

Règle duale :

$$(E + F)(\bar{E} + G).(F + G) = (E + F)(\bar{E} + G)$$

Propriétés

$$a.b = x.y \Rightarrow a.b.u = x.y.u$$

$$a + b = x + y \Rightarrow a + b + u = x + y + u$$

PAR CONTRE :

$$x + a = x + b \not\Rightarrow a = b$$

$$a.x = b.x \not\Rightarrow a = b$$

les circuits combinatoires

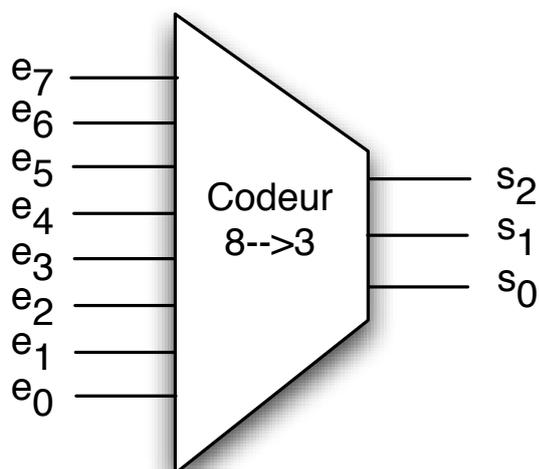
- Les transcodeurs
 - ① Les codeurs
 - ② Les décodeurs
- Les circuits d'aiguillages
 - ① Les multiplexeurs
 - ② Les démultiplexeurs

les circuits de transcodage

Ce sont les circuits qui transforment le codage d'une information.

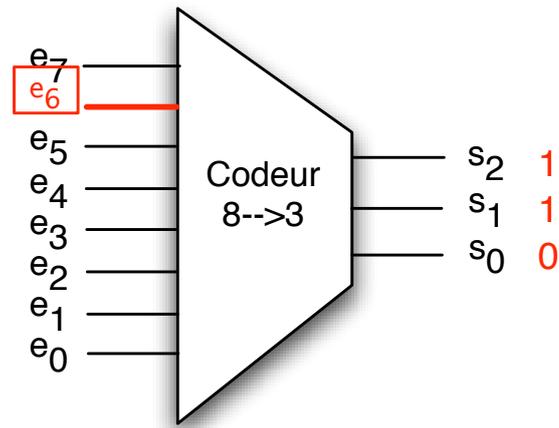
Codeur

Il transforme le codage d'une information de 2^n bits vers n bits.



Si une entrée e_i est active alors le nombre i sera représenté en binaire sur les sorties s_2, s_1, s_0

Codeur 8 vers 3



Equation des sorties d'un codeur non prioritaire :

$$s_0 = e_1 + e_3 + e_5 + e_7(\text{impair})$$

$$s_1 = e_2 + e_3 + e_6 + e_7$$

$$s_2 = e_4 + e_5 + e_6 + e_7$$

Le 74LS148 est un codeur prioritaire

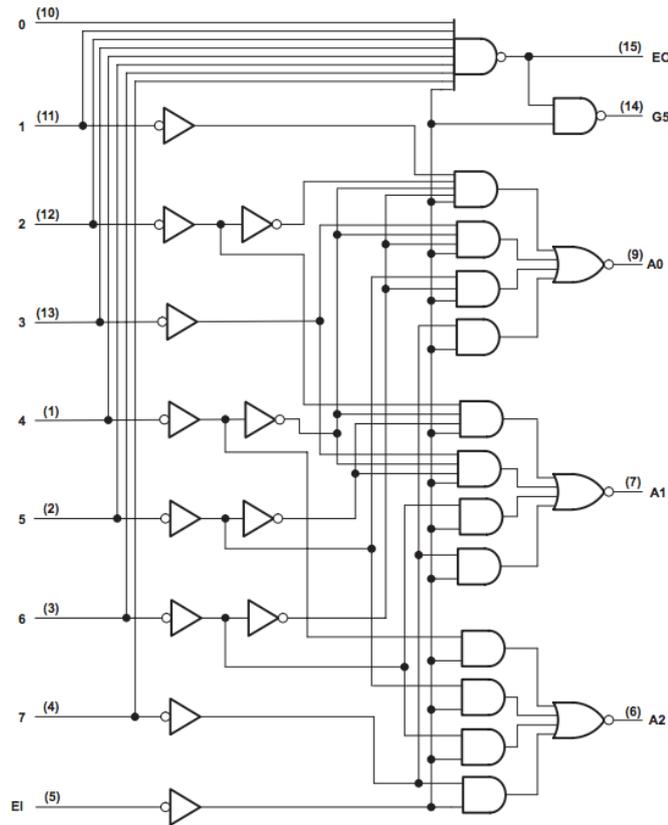
Exemple de codeur prioritaire: 74148

Table de vérité du 74148:

\overline{E}_I	Inputs								Outputs				
	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	\overline{A}_2	\overline{A}_1	\overline{A}_0	\overline{GS}	E_0
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

E_I (Enable Input) : Active le fonctionnement du codeur.

'148, 'LS148 logic diagram (positive logic)



Rôle de E_0 et GS

$\overline{E_I}$	Inputs								Outputs				
	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	GS	E_0
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

- E_0 (Enable Output) est activé si au moins une entrée a été sélectionnée
- GS (Gate Select) est activé si :
 - Soit aucune entrée n'a été sélectionnée
 - Soit E_I n'a pas été activé

Codeur 74148 : Sélection d'entrée

Table de vérité du 74148:

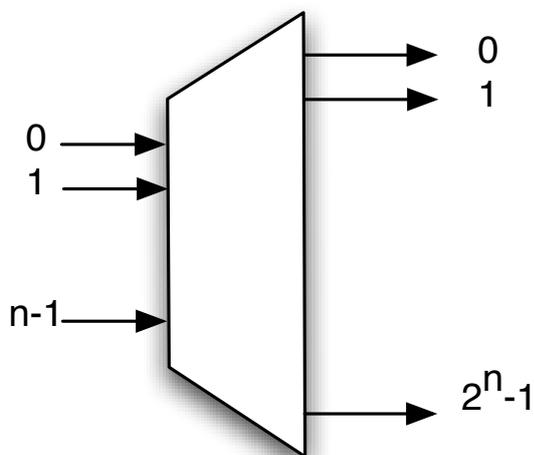
\overline{E}_I	Inputs								Outputs				
	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	\overline{A}_2	\overline{A}_1	\overline{A}_0	GS	E_0
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	L	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Si l'entrée 6 est active (par un niveau bas) alors on a sur les sorties $A_2, A_1, A_0 \rightarrow L L H$: 6 en logique négative

Les Décodeurs

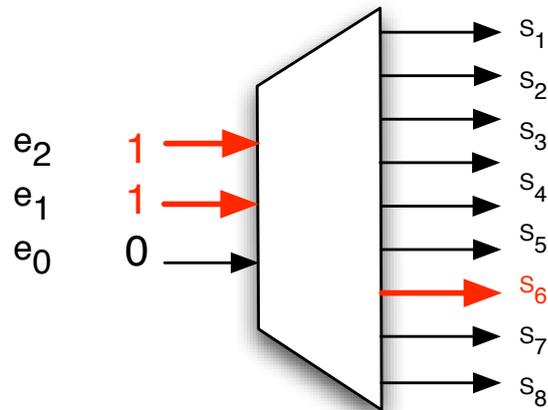
Décodeur

Il transforme le codage d'une information de n bits vers 2^n bits.



Pour chaque combinaison de variables d'entrées on a une seule sortie active. Les combinaisons d'entrées sont appelées adresses car elles expriment en numérotation binaire le numéro décimal de la sortie activée.

Exemple : un décodeur 3 vers 8



La combinaison 110 active la sortie S_6 .

Exemple de décodeur : 74138

Table de vérité du 74138:

Inputs						Outputs							
Enable			Select										
G_{2A}	$\overline{G_{2B}}$	$\overline{G_1}$	A	B	C	$\overline{Y_7}$	$\overline{Y_6}$	$\overline{Y_5}$	$\overline{Y_4}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
L	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H	L
H	L	L	L	L	H	H	H	H	H	H	H	L	H
H	L	L	L	H	L	H	H	H	H	L	H	H	H
H	L	L	L	H	H	H	H	H	H	L	H	H	H
H	L	L	H	L	L	H	H	H	L	H	H	H	H
H	L	L	H	L	H	H	H	L	H	H	H	H	H
H	L	L	H	H	L	H	L	H	H	H	H	H	H
H	L	L	H	H	H	L	H	H	H	H	H	H	H

Si on a la combinaison $HHL = 6$ sur les entrées A, B, C alors la sortie $\overline{Y_6}$ sera active (L) en logique négative.

Le distributeur de boisson

Problème

Un appareil comporte 3 cuves contenant de l'eau, du concentré de cassis, du concentré de menthe. Ce distributeur permet d'obtenir de l'eau, de la menthe à l'eau et du cassis à l'eau par le moyen des boutons e , m , c qui commandent les électrovannes E , M , C . Le mélange cassis-menthe étant interdit. Une pièce p doit être introduite sauf pour l'eau pure qui est gratuite. L'appui sur un bouton e , m , c pour l'introduction déclenche une temporisation. Si celle-ci arrive à terme avant qu'un choix cohérent soit fait, la pièce éventuellement présente est rendue par la fonction de restitution P . Cette fonction est aussi activée si le choix est incohérent (mc).

Analyse du problème



La table de vérité

e	m	c	p	E	M	C	P
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0
0	0	1	1	1	0	1	0
0	1	0	0	0	0	0	0
0	1	0	1	1	1	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1

Tables de Karnaugh : E

$E(e,m,c,p)$

		e			
		m			
c	p	00	01	11	10
		00	0	0	0
01	0	1	1	1	
11	1	0	0	1	
10	0	0	0	0	

$$E(e, m, c, p) = \bar{m}.c.p + m.\bar{c}.p + e.\bar{m}.\bar{c}$$

Tables de Karnaugh : M

$M(e,m,c,p)$

		e				
		m				
c	p	e m	0 0	0 1	1 1	1 0
		0 0	0	0	0	0
		0 1	0	1	1	0
		1 1	0	0	0	0
		1 0	0	0	0	0

$$M(e, m, c, p) = m \cdot \bar{c} \cdot p$$

Tables de Karnaugh : C

$C(e,m,c,p)$

		e				
		m				
c	p	e m	0 0	0 1	1 1	1 0
		0 0	0	0	0	0
		0 1	0	0	0	0
		1 1	1	0	0	1
		1 0	0	0	0	0

$$C(e, m, c, p) = \bar{m} \cdot c \cdot p$$

Tables de Karnaugh : P

$P(e,m,c,p)$

		e			
		m			
c	p	e	m		
				0 0	0 1
	0 0	0	0	0	0
	0 1	1	0	0	1
	1 1	0	1	1	0
	1 0	0	0	0	0

$$P(e, m, c, p) = \bar{m}.\bar{c}.p + m.c.p$$

Mise en équations

$$E(e, m, c, p) = \bar{m}.c.p + m.\bar{c}.p + e.\bar{m}.\bar{c}$$

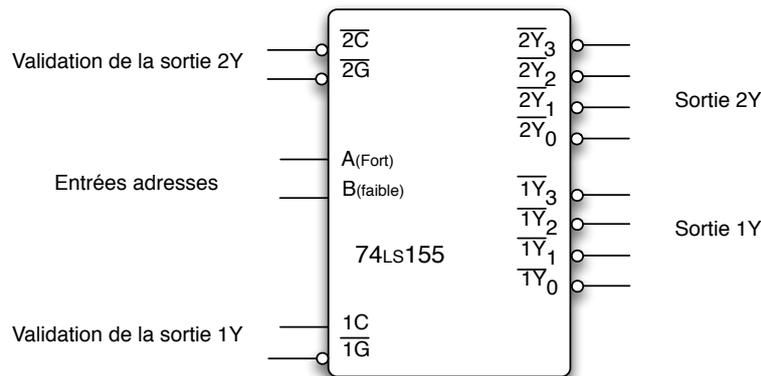
$$M(e, m, c, p) = m.\bar{c}.p$$

$$C(e, m, c, p) = \bar{m}.c.p$$

$$P(e, m, c, p) = \bar{m}.\bar{c}.p + m.c.p$$

Réalisation avec un 74LS155

Le 74155 peut être considéré comme un double décodeur 2 vers 4 ou bien comme un démultiplexeur 1 vers 4.



- A partir d'une même combinaison sur A et B on peut activer 2 lignes à la fois : une dans $1Y_i$ et une dans $2Y_i$
 - 1 $AB == 00$ les lignes $\overline{1Y_0}$ et $\overline{2Y_0}$ sont actives
 - 2 $AB == 10$ les lignes $\overline{1Y_2}$ et $\overline{2Y_2}$ sont actives
- Les entrées de validation de $\overline{1G}$ $1C$ et $\overline{2G}$ $\overline{2C}$ permettent de sélectionner le(s) groupe(s) de sortie actif(s).

Réalisation avec un 74LS155

Les équations du multiplexeur 74155 sont :

$$\overline{2Y_3} = A.B.\overline{2C}.\overline{2G}$$

$$\overline{1Y_3} = A.B.1C.\overline{1G}$$

$$\overline{2Y_2} = A.\overline{B}.\overline{2C}.\overline{2G}$$

$$\overline{1Y_2} = A.\overline{B}.1C.\overline{1G}$$

$$\overline{2Y_1} = \overline{A}.B.\overline{2C}.\overline{2G}$$

$$\overline{1Y_1} = \overline{A}.B.1C.\overline{1G}$$

$$\overline{2Y_0} = \overline{A}.\overline{B}.\overline{2C}.\overline{2G}$$

$$\overline{1Y_0} = \overline{A}.\overline{B}.1C.\overline{1G}$$

Réalisation avec un 74LS155

Dans toutes les expressions de E, M, C, P on trouve toujours m et c avec e ou p en facteur. Donc plaçons :

- m et c sur les entrées adresses A et B ;
- p sur l'entrée de validation $1C$;
- e sur l'entrée de validation $2G$;
- les validations $1G$ et $2C$ à la masse ;

Réalisation avec un 74LS155

Les équations du multiplexeur deviennent :

$$\overline{2Y_3} = m.c.\bar{e}$$

$$\overline{1Y_3} = m.c.p$$

$$\overline{2Y_2} = m.\bar{c}.\bar{e}$$

$$\overline{1Y_2} = m.\bar{c}.p$$

$$\overline{2Y_1} = \bar{m}.c.\bar{e}$$

$$\overline{1Y_1} = \bar{m}.c.p$$

$$\overline{2Y_0} = \bar{m}.\bar{c}.\bar{e}$$

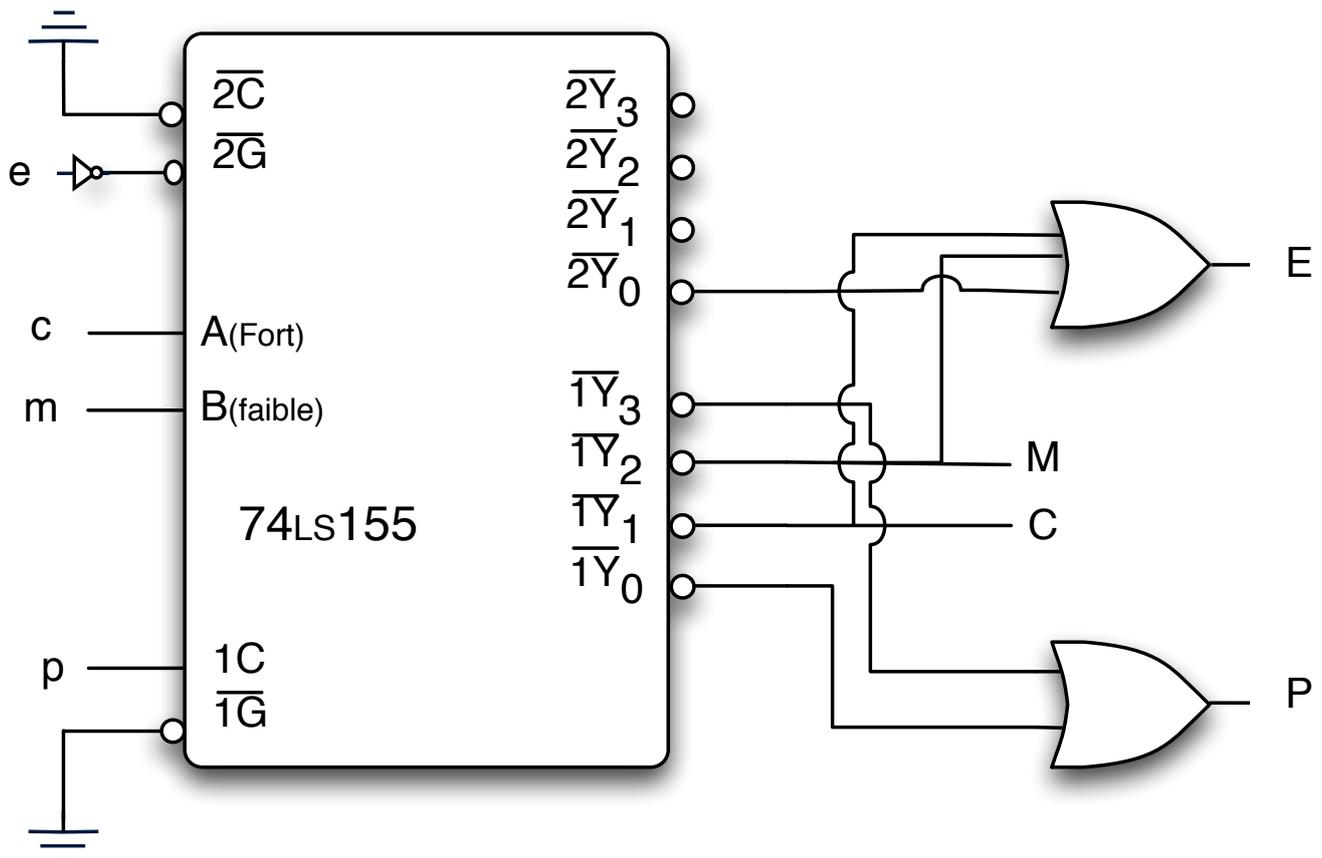
$$\overline{1Y_0} = \bar{m}.\bar{c}.p$$

Réalisation avec un 74LS155

D'ou les expressions de E, M, C, P en fonction des $i\bar{Y}_j$:

- $M = m.\bar{c}.p = \bar{1Y}_2$;
- $C = \bar{m}.c.p = \bar{1Y}_1$;
- $P = \bar{m}.\bar{c}.p + m.c.p = \bar{1Y}_0 + \bar{1Y}_3$;
- $E = \bar{m}.c.p + m.\bar{c}.p + \bar{m}.\bar{c}.e = \bar{1Y}_1 + \bar{1Y}_2 + \bar{2Y}_0$

Réalisation avec un 74LS155



Le Multiplexeur

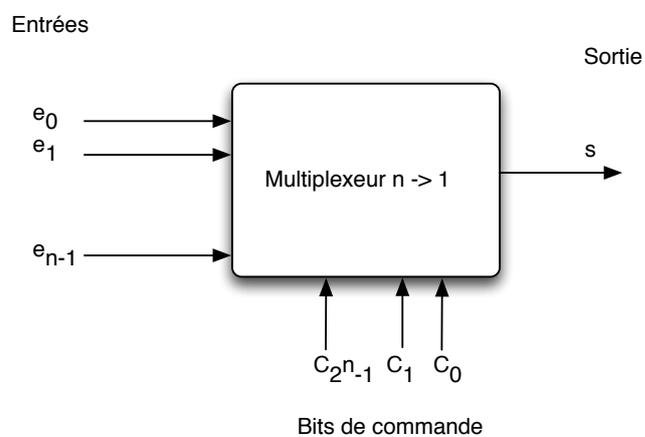
Multiplexeur

Un multiplexeur réalise l'aiguillage de 2^n entrées vers une sortie avec n bits de commande.

Le multiplexeur

Equation de la sortie :

$$\begin{aligned} s &= e_0 \cdot \overline{c_{n-1}} \cdot \overline{c_{n-2}} \cdot \dots \cdot \overline{c_1} \cdot \overline{c_0} \\ &+ e_1 \cdot \overline{c_{n-1}} \cdot \overline{c_{n-2}} \cdot \dots \cdot \overline{c_1} \cdot c_0 \\ &+ e_2 \cdot \overline{c_{n-1}} \cdot \overline{c_{n-2}} \cdot \dots \cdot c_1 \cdot \overline{c_0} \\ &+ \dots \\ &+ e_n \cdot c_{n-1} \cdot c_{n-2} \cdot \dots \cdot c_1 \cdot c_0 \end{aligned}$$



Le Démultiplexeur

Démultiplexeur

Un démultiplexeur réalise l'aiguillage d'une entrée vers 2^n sorties avec n bits de commande.

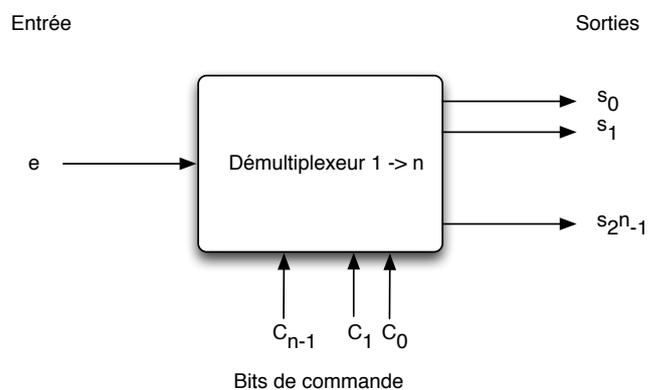
Le Démultiplexeur

Equation des sorties :

$$s_0 = e \cdot \overline{c_{n-1}} \cdot \overline{c_{n-2}} \cdot \dots \cdot \overline{c_1} \cdot \overline{c_0}$$

$$s_1 = e \cdot \overline{c_{n-1}} \cdot \overline{c_{n-2}} \cdot \dots \cdot \overline{c_1} \cdot c_0$$

$$s_n = e \cdot c_{n-2} \cdot \dots \cdot c_1 \cdot c_0$$



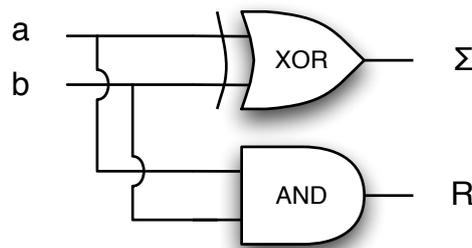
L'additionneur

C'est un circuit qui réalise l'addition de deux nombres binaires et produit la somme et la retenue :

$0 + 0$	$=$	0	retenue $= 0$
$0 + 1$	$=$	1	retenue $= 0$
$1 + 0$	$=$	1	retenue $= 0$
$1 + 1$	$=$	0	retenue $= 1$

$$\Sigma = \bar{a}.b + a\bar{b}$$

$$\text{Retenue} = a.b$$



L'additionneur complet

Pour réaliser une addition sur n bits, il faut un additionneur capable de réaliser l'addition sur 3 éléments : a_i , b_i et la retenue produite par le rang précédent r_i .

a_i	b_i	r_i	r_{i+1}	Σ_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

L'additionneur complet

$\Sigma_i(a_i, b_i, r_i)$

		a_i				
		b_i				
r_i	a_i	b_i	0 0	0 1	1 1	1 0
	0	1	0	1	0	1
0	0	1	5	4		
1	1	0	1	0		
1	2	3	7	6		

Equation de la somme
*Sigma*_i

$$\Sigma_i = a_i \oplus b_i \oplus r_i$$

L'additionneur complet

$r_{i+1}(a_i, b_i, r_i)$

		a_i				
		b_i				
r_i	a_i	b_i	0 0	0 1	1 1	1 0
	0	1	0	1	0	1
0	0	1	5	4		
1	1	0	1	0		
1	2	3	7	6		

Equation de la retenue r_{i+1}

$$\begin{aligned} r_{i+1} &= a_i \cdot b_i + r_i \cdot b_i + r_i \cdot a_i \\ &= a_i \cdot b_i + r_i \cdot (a_i + b_i) \end{aligned}$$

L'additionneur complet

$r_{i+1}(a_i, b_i, r_i)$

	a_i		b_i		
r_i	0 0	0 1	1 1	1 0	
0	0	0	1	0	
1	0	1	1	1	
	2	3	7	6	

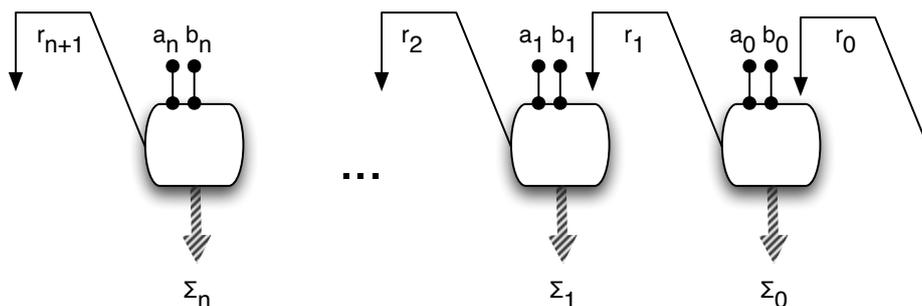
Equation de la retenue r_{i+1}

$$\begin{aligned}
 r_{i+1} &= a_i \oplus b_i \oplus r_i \\
 &= a_i \cdot b_i + r_i \cdot (\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i) \\
 &= a_i \cdot b_i + r_i \cdot (a_i \oplus b_i)
 \end{aligned}$$

L'additionneur complet

Equations de l'additionneur complet

$$\begin{aligned}
 \Sigma_i &= (a_i \oplus b_i) \oplus r_i \\
 r_{i+1} &= a_i \cdot b_i + r_i \cdot (a_i \oplus b_i)
 \end{aligned}$$



L'additionneur a retenue anticipée

Prenons la seconde forme de l'additionneur :

$$r_{i+1} = a_i \cdot b_i + r_i \cdot (a_i \oplus b_i) = a_i \cdot b_i + r_i \cdot (a_i + b_i)$$

Posons : $P_i = a_i + b_i$ et $G_i = a_i \cdot b_i$

On a :

$$r_1 = a_0 \cdot b_0 + (a_0 + b_0) \cdot r_0 = G_0 + P_0 \cdot r_0$$

$$r_2 = G_1 + P_1 \cdot r_1$$

$$= G_1 + P_1 \cdot (G_0 + P_0 \cdot r_0) = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot r_0$$

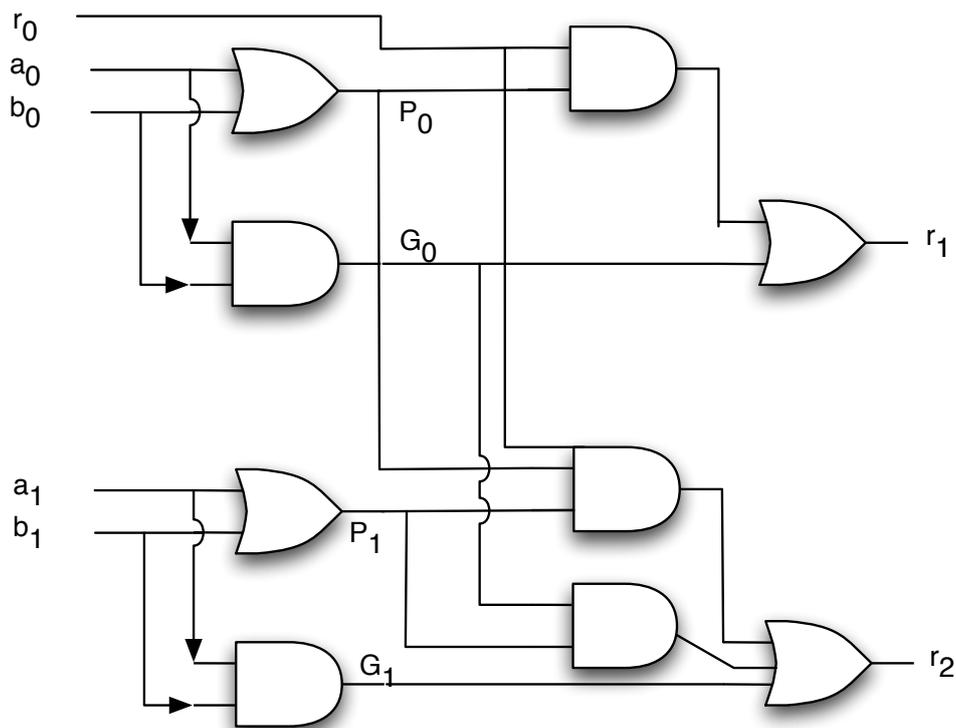
$$r_3 = G_2 + P_2 \cdot r_2$$

$$= G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot r_0)$$

$$= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot r_0$$

...

L'additionneur a retenue anticipée

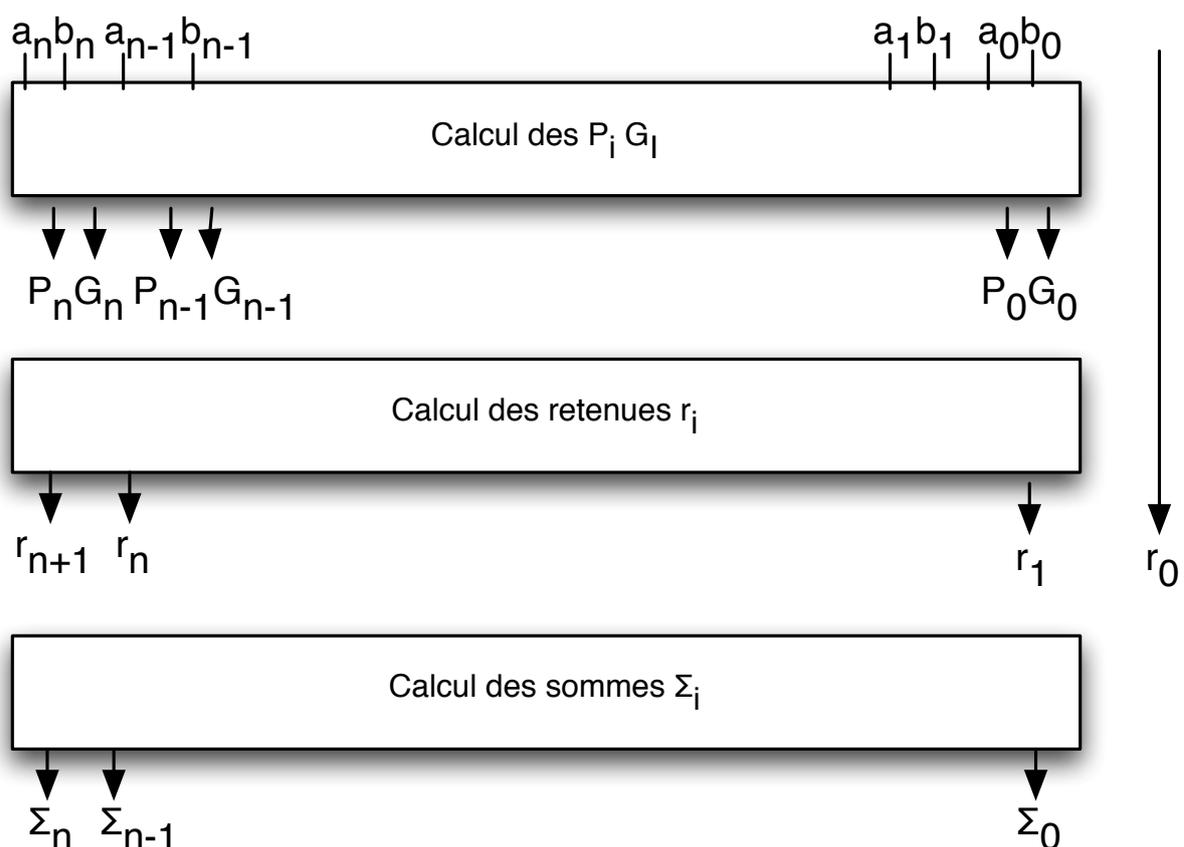


...

L'additionneur a retenue anticipée

- Les temps de calcul des retenues sont tous égaux à condition de réaliser des portes à plus de deux entrées.
- Ils correspondent aux temps de calcul :
 - ▶ des P_i, G_i ,
 - ▶ l'étage des ET
 - ▶ l'étage des OU .

L'additionneur a retenue anticipée



Le comparateur

Un comparateur compare deux nombres A et B sur n bits délivre six sorties :

- GT : sortie active si $A > B$
- EQ : sortie active si $A = B$
- LT : sortie active si $A < B$
- GE : sortie active si $A \geq B$
- LE : sortie active si $A \leq B$
- NE : sortie active si $A \neq B$

Le comparateur

Un comparateur compare deux nombres A et B sur n bits délivre six sorties :

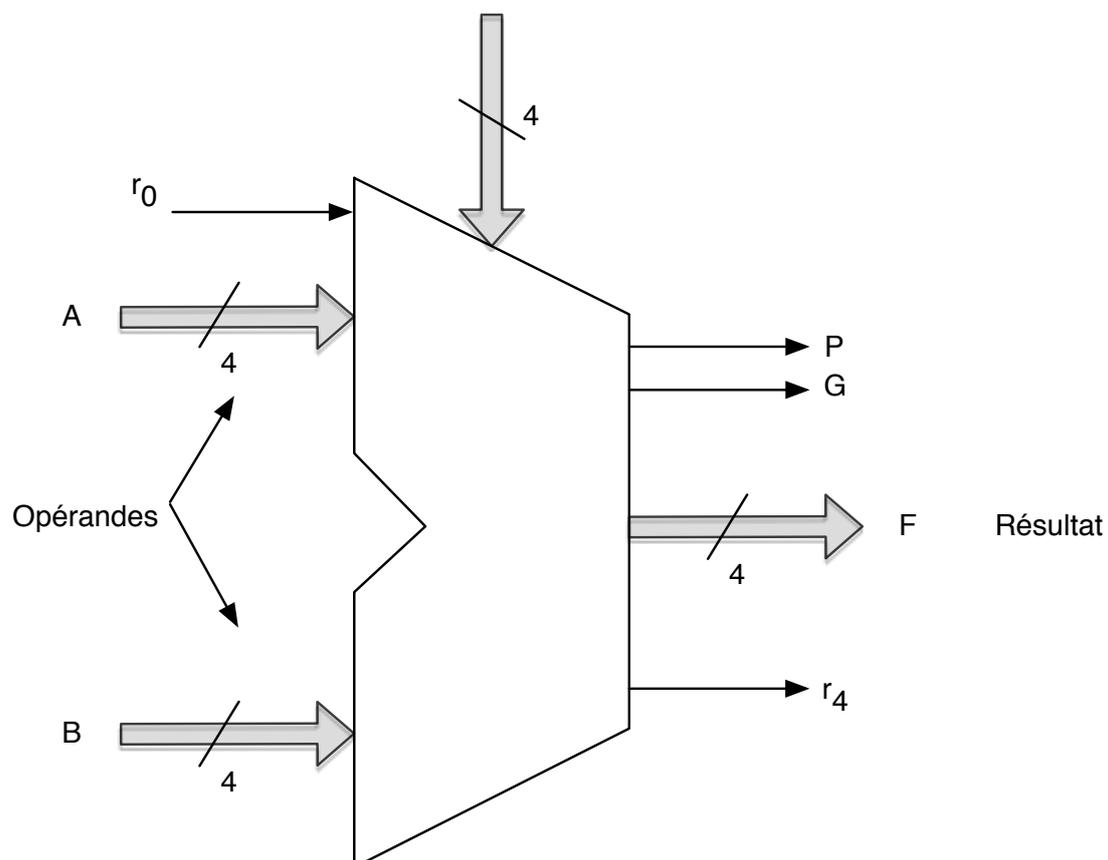
$$GT = A_n \cdot \bar{B}_n + (A_n \odot B_n) \cdot (A_{n-1} \cdot \bar{B}_{n-1}) + \dots + (A_n \odot B_n) \dots (A_1 \odot B_1) \cdot (A_0 \cdot \bar{B}_0)$$

EQ	=	$(A_n \cdot \bar{B}_n + \dots + A_0 \cdot \bar{B}_0)$
LT	=	\overline{GT}
GE	=	GT
LE	=	LT
EQ	=	\overline{EQ}

L'UAL est un circuit 4 bits capable d'effectuer :

- Arithmétiques :
 - ▶ Additions : $A + B$, $A + B + 1$ (+ = addition)
 - ▶ $-A = CA_2(A)$
 - ▶ Soustractions : $A - B$, $A - B - 1$
- Logiques :
 - ▶ ET : $(A_3.B_3).(A_2.B_2).(A_1.B_1).(A_0.B_0)$
 - ▶ OU : $(A_3 + B_3).(A_2 + B_2).(A_1 + B_1).(A_0 + B_0)$ (+ = ou)
 - ▶ $NOT(A) = CA_1(A)$
 - ▶ Nor, Nand ...
 - ▶ Décalages $A \leftarrow LSL(A) \dots$

Unité arithmétique et logique



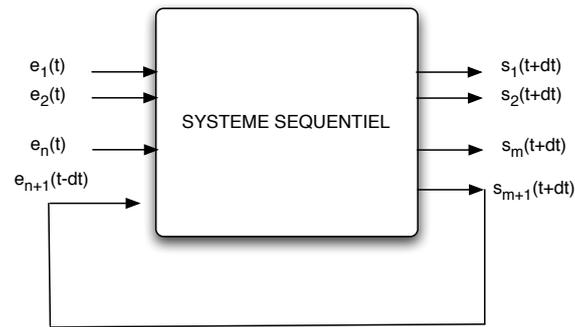


Figure: Partie séquentielle

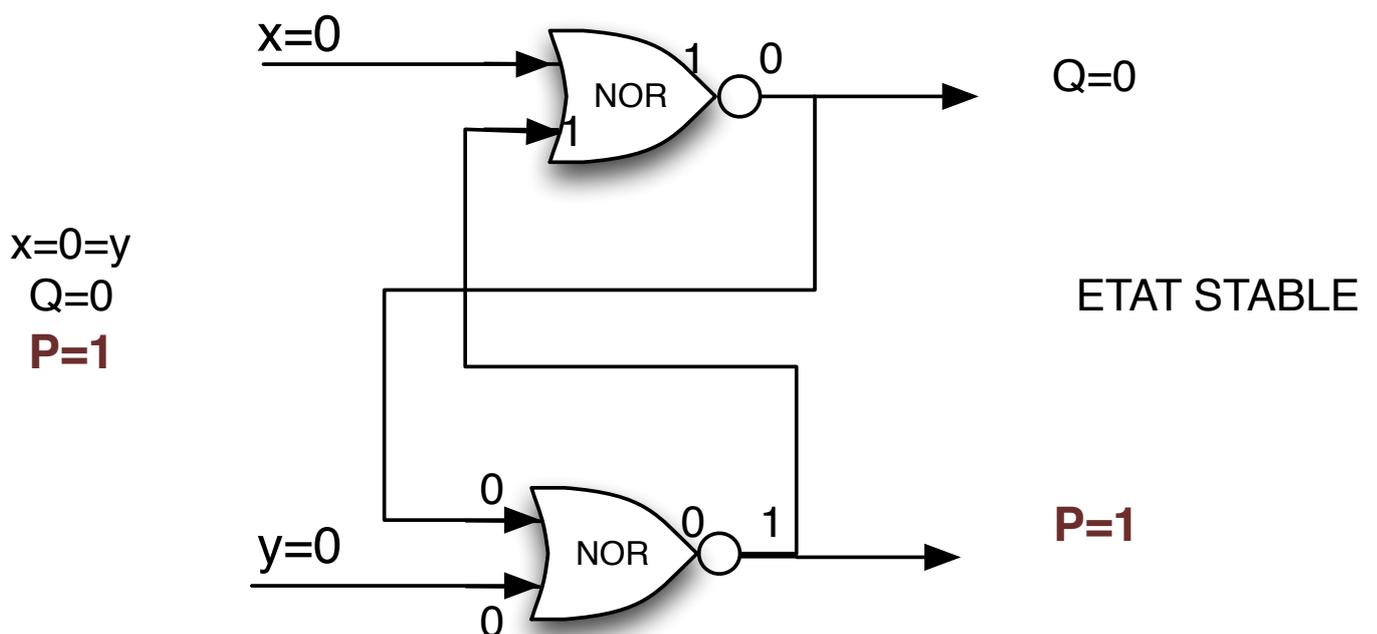
Objectifs du cours

- ① Introduire la notion de mémoire
- ② Circuit de bases :
 - ▶ les bascules,
 - ▶ les maîtres esclaves
 - ▶ les séquenceurs
- ③ L'analyse de circuits séquentiels

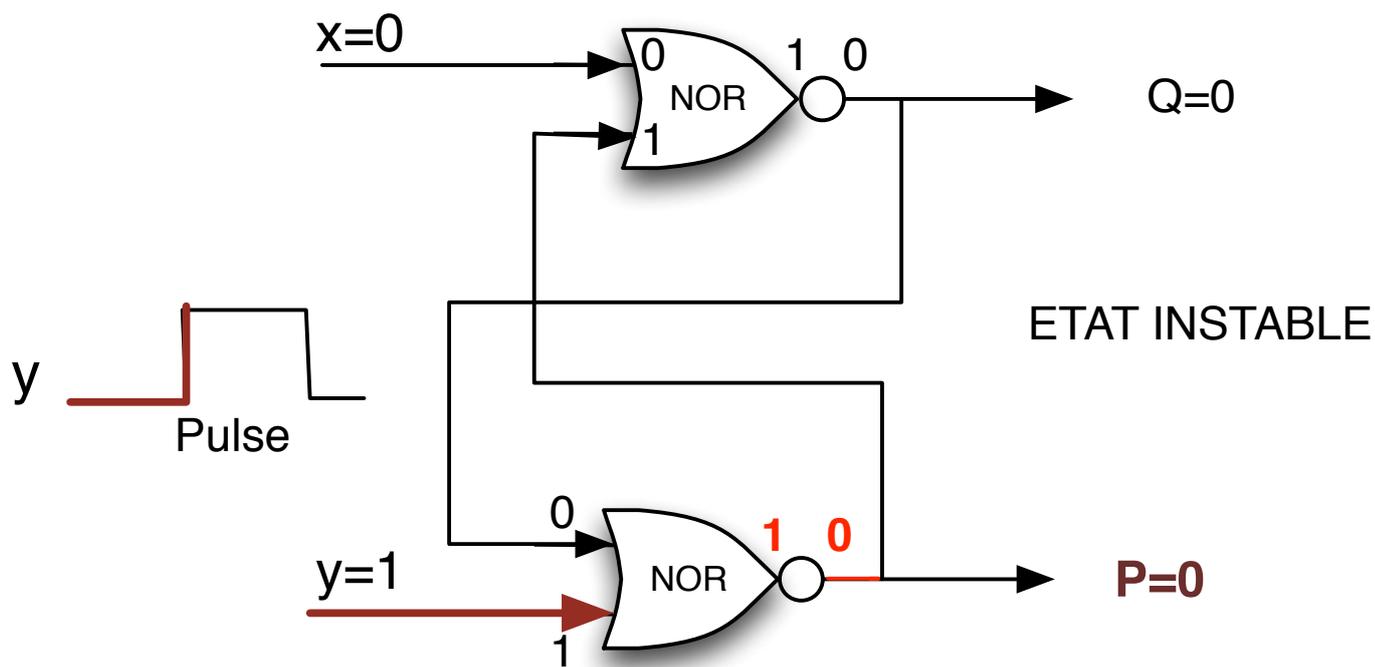
L'élément mémoire élémentaire

- La lecture ;
- La mise à un ;
- La mise à zéro.

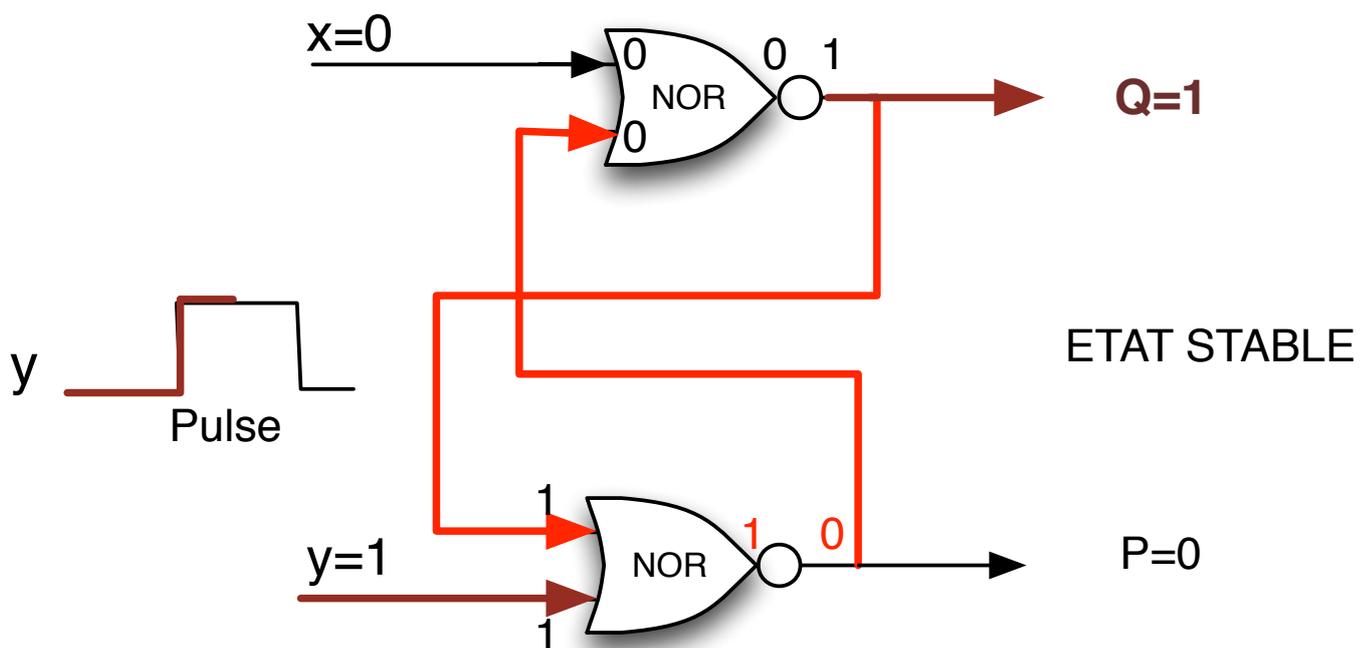
Le principe



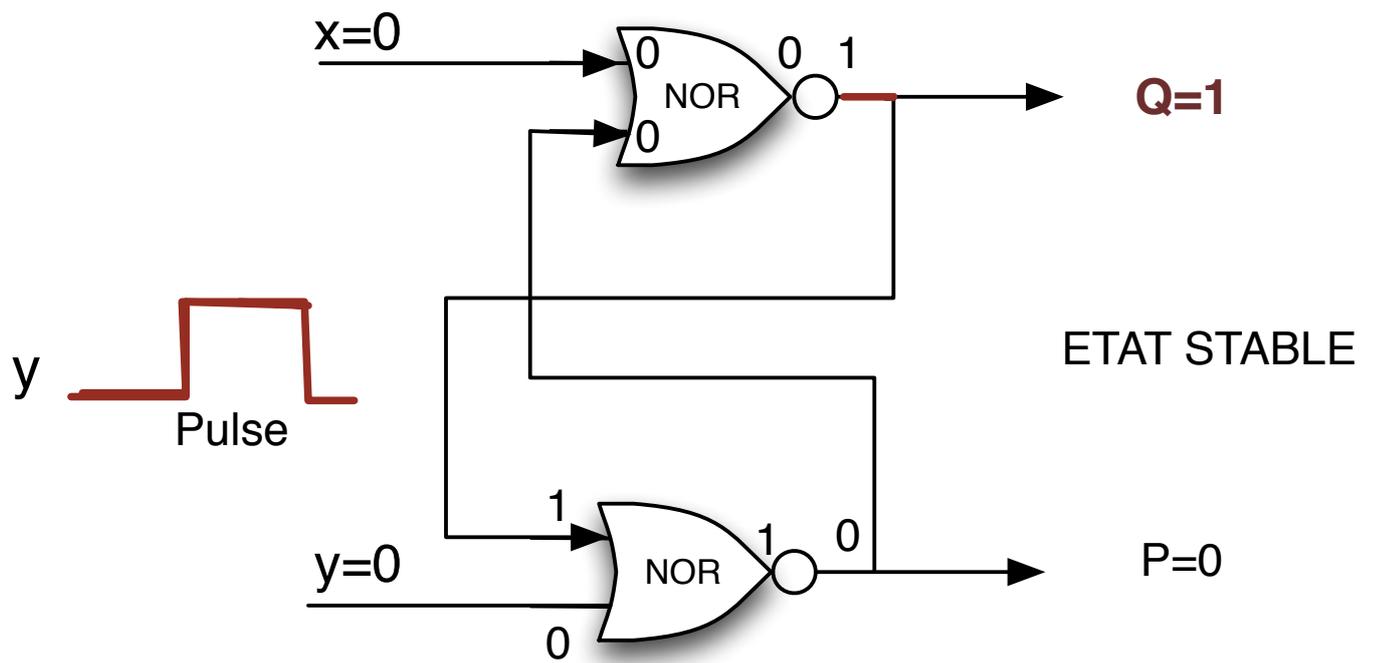
Le principe



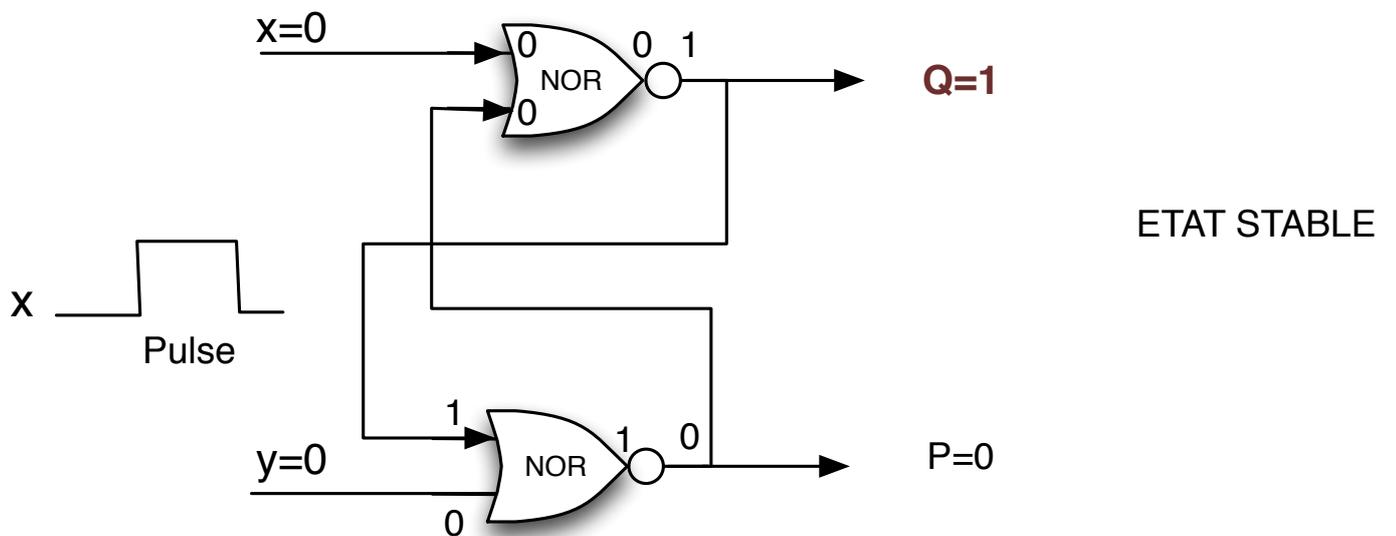
Le principe



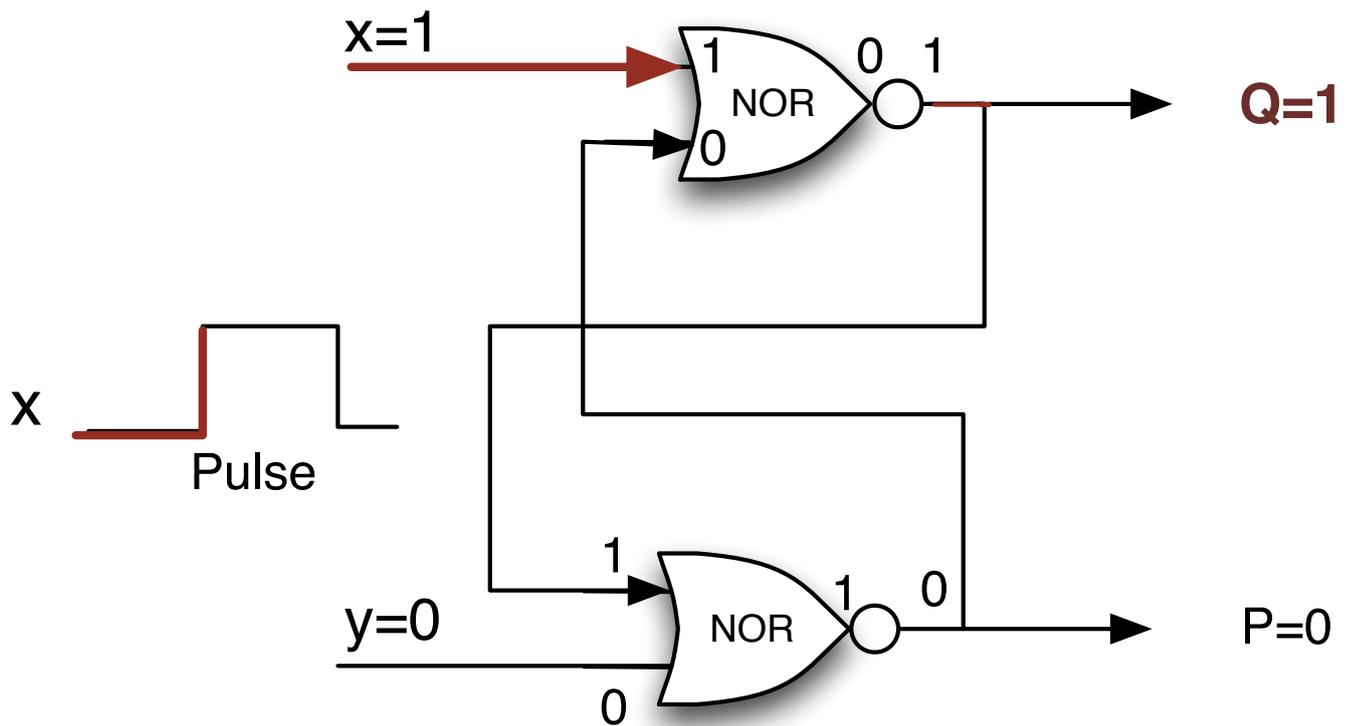
Le principe



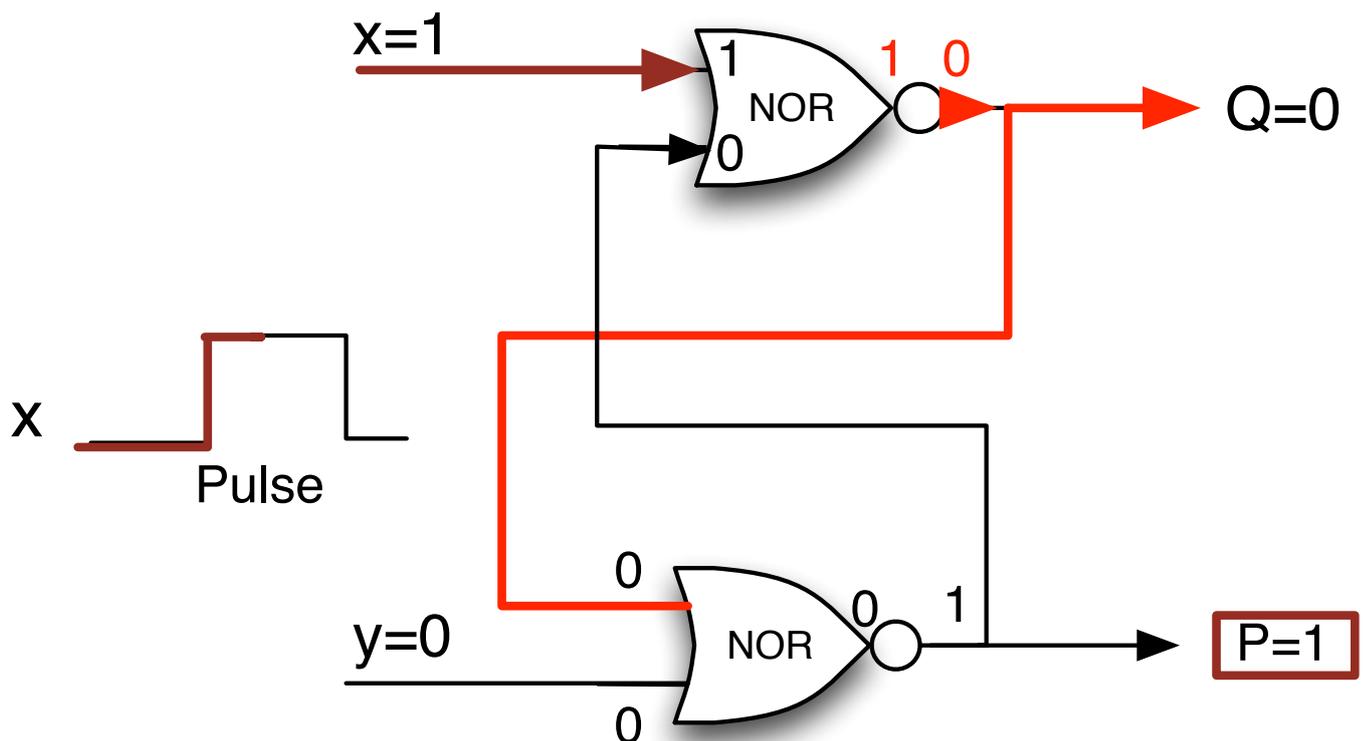
Le principe



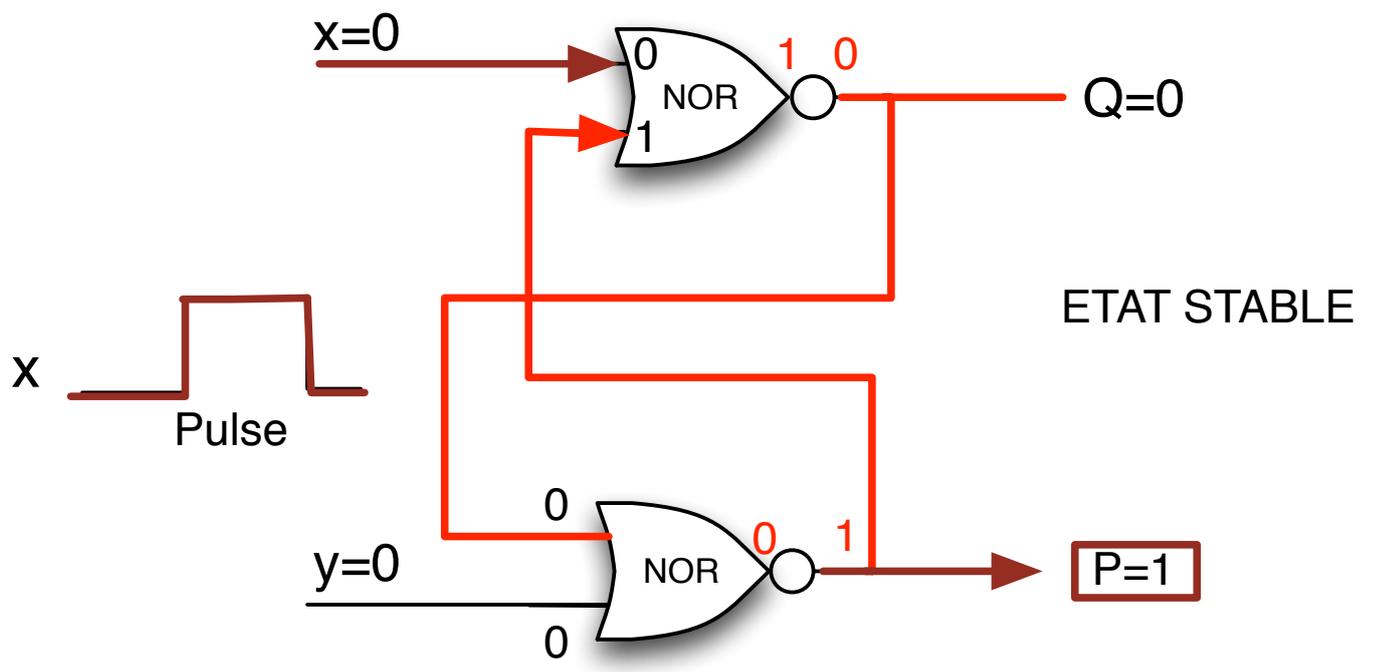
Le principe



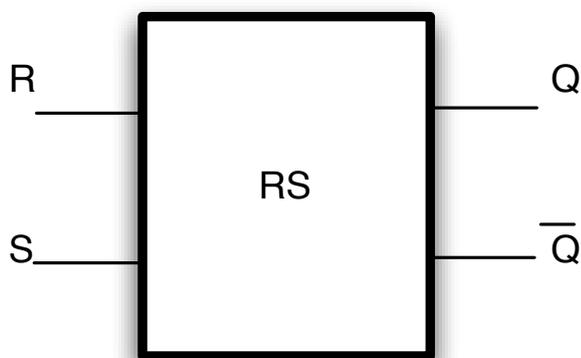
Le principe



Le principe



La RS asynchrone



R	S	Q^{t+1}	
0	0	Q^t	Mémoire
0	1	1	Set
1	0	0	Reset
1	1	*	Interdit

La RS asynchrone

$$Q^{t+1} = \overline{R}.\overline{S}.Q^t + \overline{R}.S + RS$$

$$= \overline{R}.Q^t + S$$

L'équation caractéristique est :

$$Q^{t+1} = \overline{R}.Q^t + S$$

$f(R,S,Q^t)$

		S		R	1 0
		0 0	0 1		
Q ^t	0 0	1	*	0	1 0
	1 1	1	*	0	

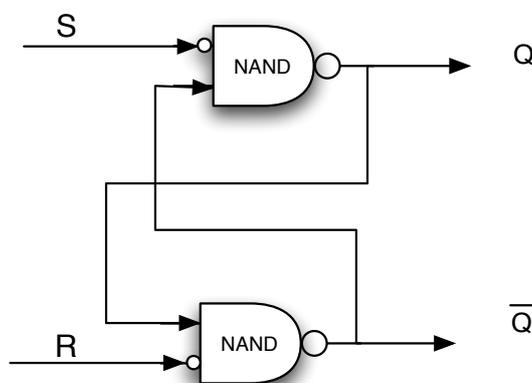
Diagram description: A truth table for the RS flip-flop characteristic function. The columns are labeled with RS pairs (00, 01, 11) and the output Q (1, 0). The rows are labeled with Q^t (0, 1) and the output Q (1, 0). Red circles highlight the '1' outputs in the (0,0) and (1,1) RS states. A 'fram' label is present at the bottom left of the table.

Réalisation de la RS asynchrone en NAND

En NAND, à partir d'une forme en somme:

$$\overline{\overline{Q^{t+1}}} = \overline{\overline{\overline{R}.Q^t + .S}} = \overline{\overline{R}.Q^t.\overline{S}}$$

D'où:



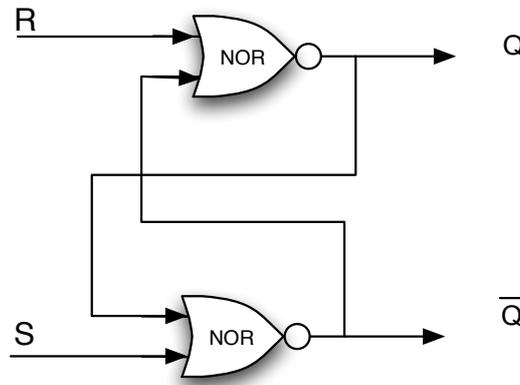
Réalisation de la RS asynchrone en NOR

En NOR, à partir d'une forme en produit :

$$Q^{t+1} = \overline{R} \cdot \overline{S} \cdot Q^t + \overline{R} \cdot S = \overline{R} \cdot (\overline{S} \cdot Q^t + S) = \overline{R} \cdot (Q^t + S)$$

$$Q^{t+1} = \overline{\overline{Q^{t+1}}} = \overline{\overline{\overline{R} \cdot (Q^t + S)}} = \overline{R + \overline{Q^t + S}}$$

D'où:



Race conditions

⇒ Que se passe-t-il si on met on applique R et S en même temps ?

- Si $S = R = 1$ alors $Q = \overline{Q} = 0$ (en NOR). Cet état est alors stable.
- Considérons maintenant l'application simultanée de $S = R = 0$, les deux portes n'étant jamais réellement identiques, l'une est plus rapide et donc, l'une commute sa sortie à un en premier verrouillant la seconde porte dont la sortie restera bloquée à zéro désormais.
- On a ici une situation de course : c'est la porte la plus rapide qui définit l'état de la bascule !!!

Race conditions : une première idée

Une première idée est de synchroniser les signaux d'entrée avec un signal d'autorisation par une porte *ET*.

- Si ce signal est à zéro, les entrées sont à zéros,
- Si ce signal est à UN, les entrées sont susceptibles de faire évoluer les sorties.

Ce concept permet de se prémunir contre les variations indésirables : les parasites.

Typologie des bascules

Il y a trois types de bascules :

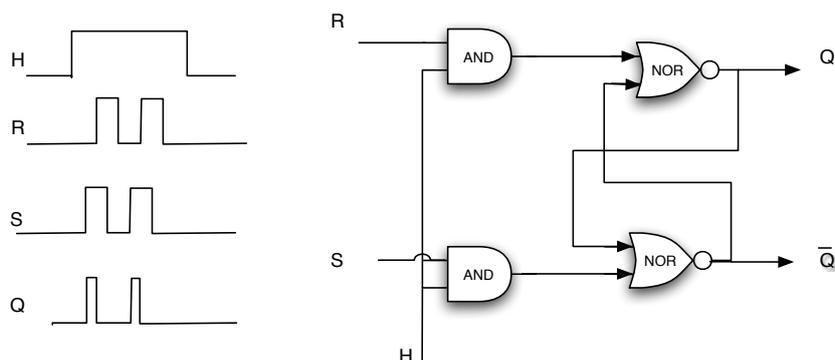
- les "Latch" : à Déclenchement sur niveau (haut ou bas) ;
- Les "Edge triggered" : à déclenchement sur front ;
- Les maîtres-esclaves.

Les "Latch"

Bascule Latch

C'est une bascule synchrone dont toute variation sur les entrées pendant le niveau d'activation du signal d'horloge est pris en compte sur ses sorties.

exemple :



exemple de LATCH : La bascule synchrone RST

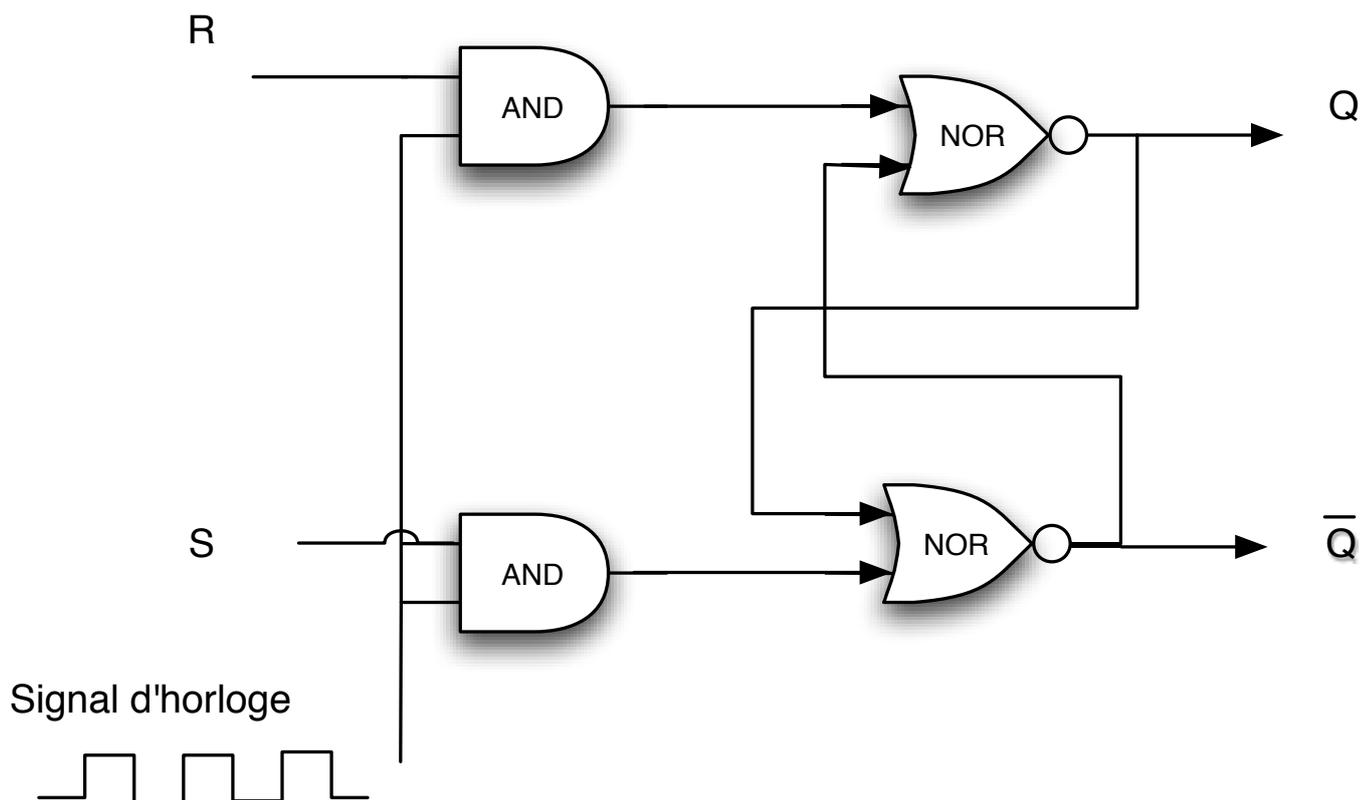
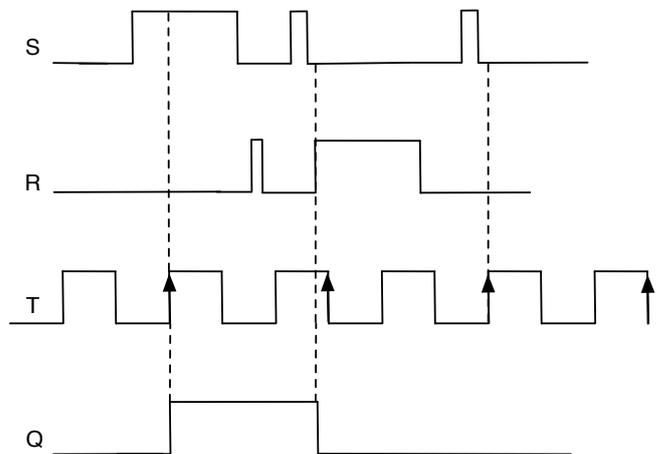


Table de vérité de la RST

T	R	S	Q^{t+1}
0	*	*	Q^t
1	0	0	Q^t
1	0	1	1
1	1	0	0
1	1	1	*

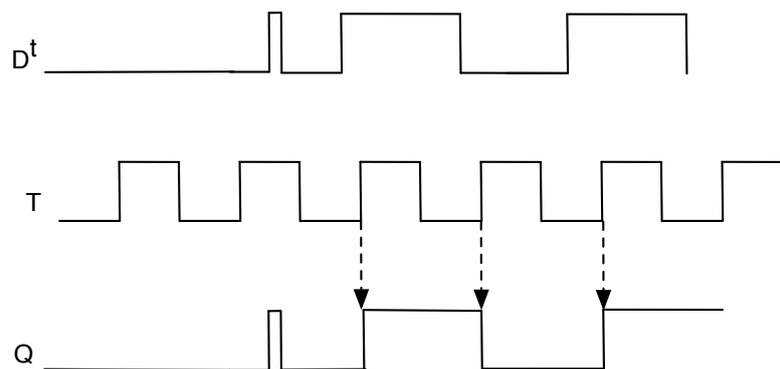


La bascule synchrone DT-LATCH

- Cette bascule possède une entrée D et une entrée horloge T .
- Cette bascule fonctionne en mode copie sur un niveau haut de T
- en mode mémoire sur un niveau bas de T

Table de vérité de la DT

T	D	Q^{t+1}
0	*	Q^t
1	0	0
1	1	1



Réalisation d'une DT-LATCH

A partir d'une RS :

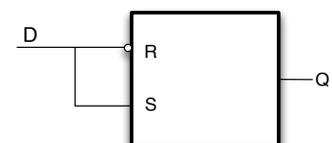
$$Q^{t+1} = S + \bar{R}.Q \quad (1)$$

$$Q^{t+1} = D \quad (2)$$

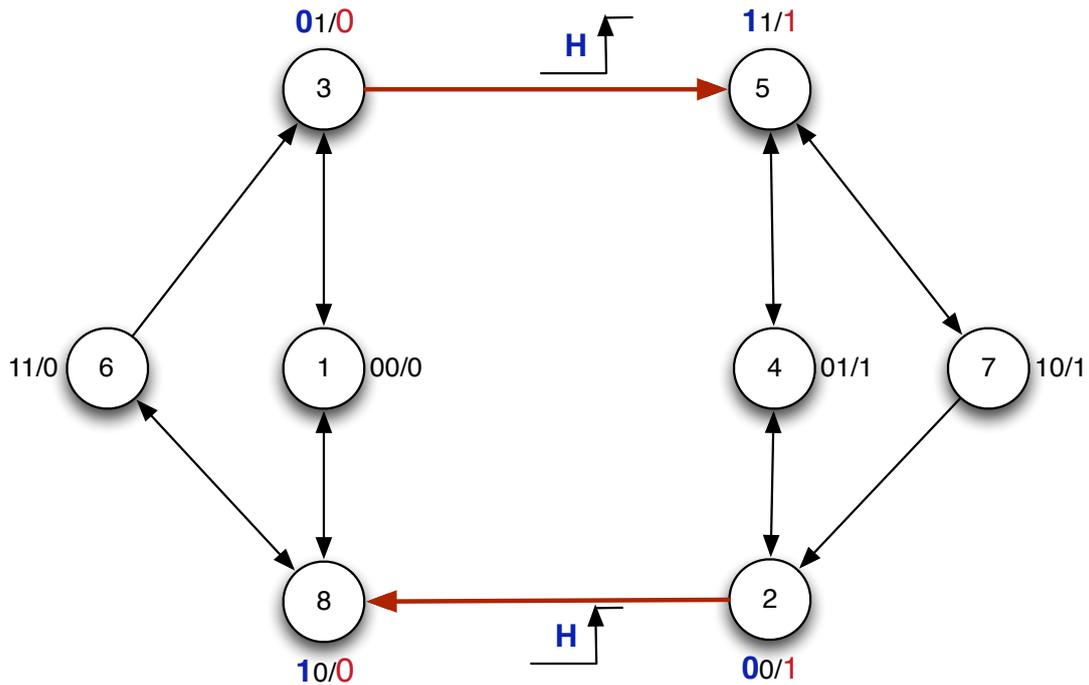
$$\text{or } D = D + DQ \quad (3)$$

$$\Rightarrow S = D \text{ et } D = \bar{R} \quad (4)$$

- (1) est l'équation caractéristique de la RS
- (2) est l'équation caractéristique de la D
- (3) correspond à un des théorèmes de la logique : $D + DQ = D(1 + Q) = D$
- (4) correspond à la façon de réaliser la bascule D^T



D positive edge triggered (HD/Q)

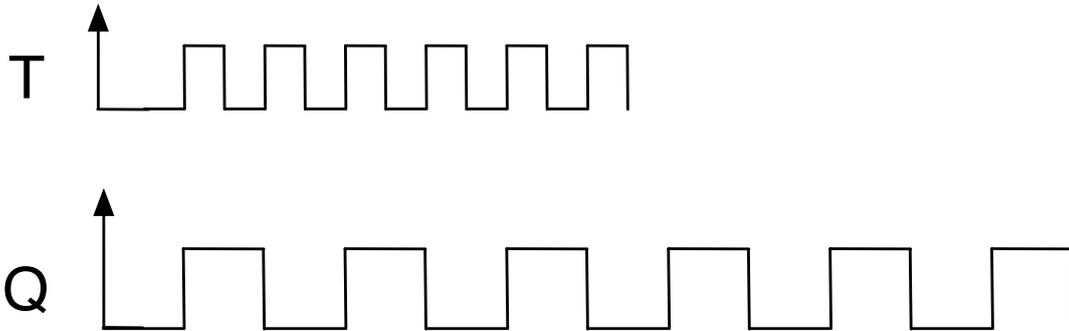


D positive edge triggered

- 1 Quand l'horloge est stable
 - ▶ $6 \leftrightarrow 8, 3 \leftrightarrow 1$ pour $Q = 0$
 - ▶ $5 \leftrightarrow 7, 4 \leftrightarrow 2$ pour $Q = 1$LA SORTIE NE VARIE PAS.
- 2 Quand l'horloge passe de 1 vers 0 : $5 \rightarrow 4, 8 \rightarrow 1$: LA SORTIE NE VARIE PAS
- 3 Quand l'horloge passe de 0 vers 1 avec $D = 0$ et $Q = 0$: $1 \rightarrow 8$
L'entrée D est appliquée sur la sortie mais comme Q était déjà à zéro il n'y a PAS DE CHANGEMENT D'ÉTAT.
- 4 Quand l'horloge passe de 0 vers 1 avec $D = 1$ et $Q = 1$: $4 \rightarrow 5$
L'entrée D est appliquée sur la sortie mais comme Q était déjà à un il n'y a PAS DE CHANGEMENT D'ÉTAT.
- 5 La sortie CHANGE D'ÉTAT lors d'un front montant de H avec $D \neq Q$ c.a.d. lors des transitions : $3 \rightarrow 5$ et $2 \rightarrow 8$

La bascule T

Une bascule fonctionnant suivant le type T dispose d'une entrée unique T . La sortie Q change d'état à chaque impulsion sur T . Elle divise la fréquence d'entrée de T par 2.



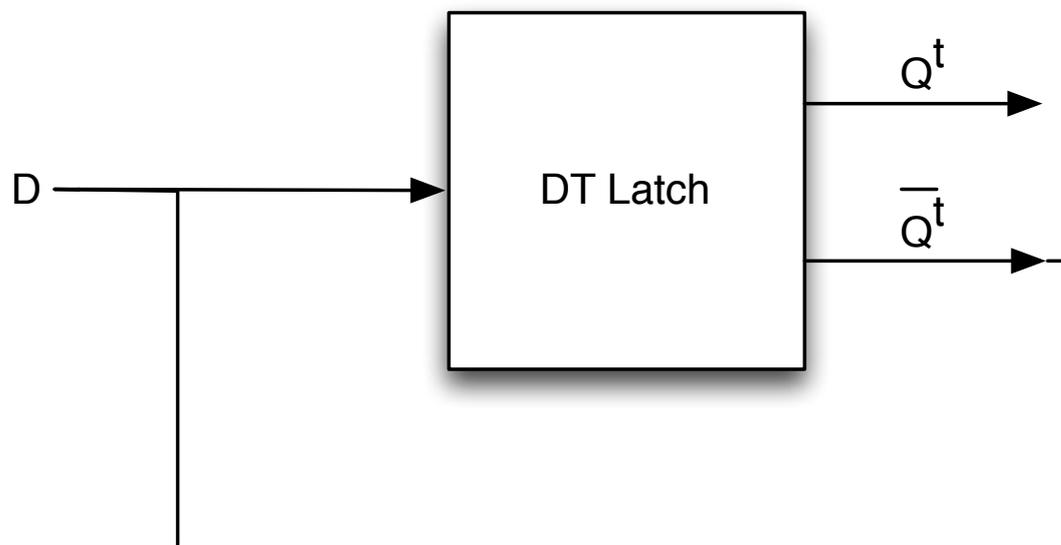
La bascule T

Réalisation :

$$D = Q_{t+1} = \overline{Q}_t$$

On réalise cette bascule en rebouclant la sortie Q sur l'entrée D d'une bascule Délai. Cependant, Cette bascule ne peut être réalisée avec une bascule de type "latch", mais avec une "Edge" : à déclenchement sur front.

La bascule T avec une DT



La JKT

- J et K similaire à S et R
- $J = K = 1$ est autorisé : fonctionne comme une bascule T

Réalisation d'une JKT

J	K	Q^{t+1}
0	0	Q^t
0	1	0
1	0	1
1	1	$\overline{Q^t}$

$$\begin{aligned}Q^{t+1} &= J.K.\overline{Q^t} + J.\overline{K} + \overline{J}.\overline{K}.Q^t \\&= J.K.\overline{Q^t} + J.\overline{K}(\overline{Q^t} + Q^t) + \overline{J}.\overline{K}.Q^t \\&= J.K.\overline{Q^t} + J.\overline{K}.\overline{Q^t} + J.\overline{K}.Q^t + \overline{J}.\overline{K}.Q^t \\&= J.\overline{Q^t}(K + \overline{K}) + \overline{K}.Q^t(J + \overline{J}) \\&= J.\overline{Q^t} + \overline{K}.Q^t\end{aligned}$$

La JKT

Pour la réaliser, on va identifier les termes 2 à 2 dans l'équation suivante :

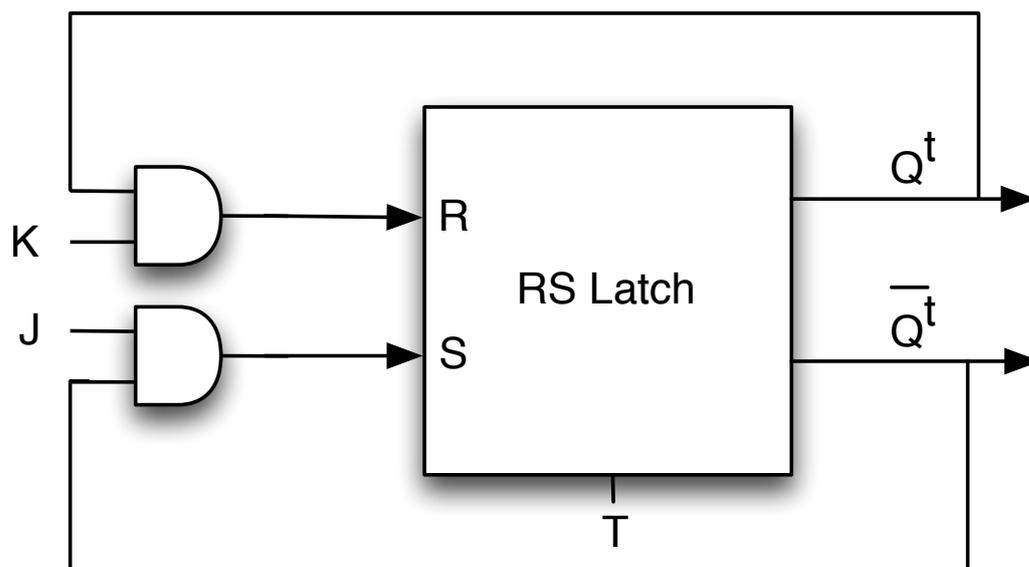
$$J.\overline{Q^t} + \overline{K}.Q^t = S + \overline{R}Q^t$$

En posant : $S = J.\overline{Q^t}$ et $R = K.Q^t$

En effet, avec ce choix pour R , on obtient alors :

$$\overline{KQ}.Q = (\overline{K} + \overline{Q}).Q = \overline{K}.Q$$

La bascule JKT avec une RS

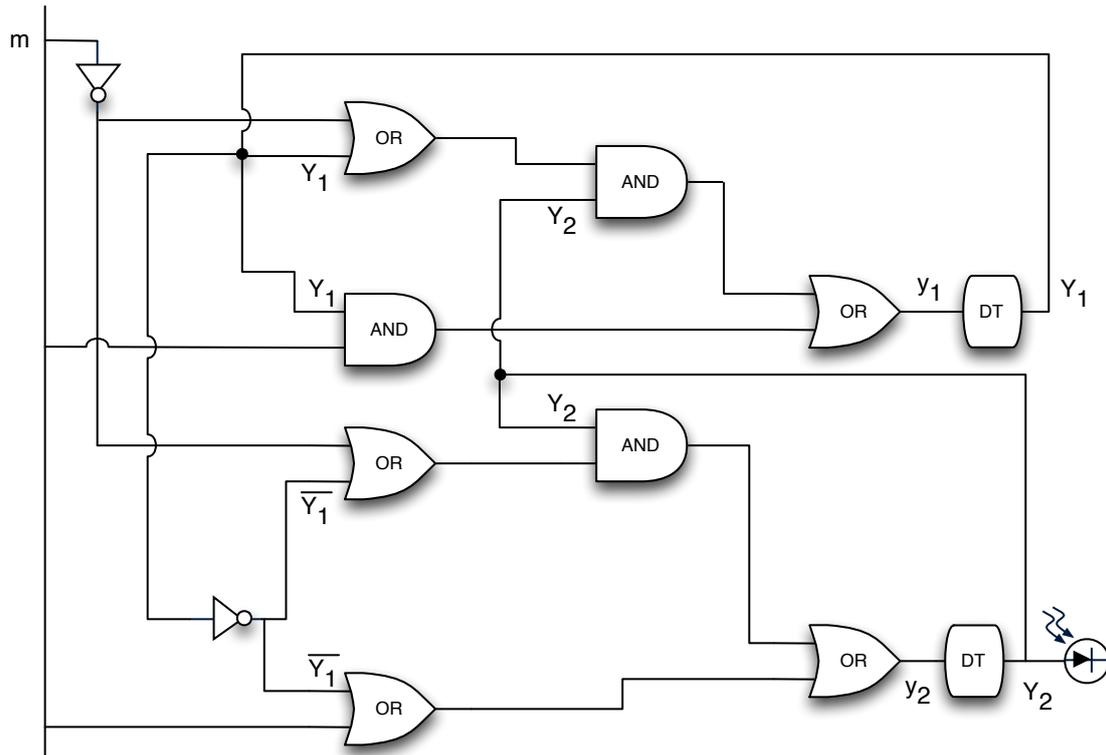


Avec une JKT

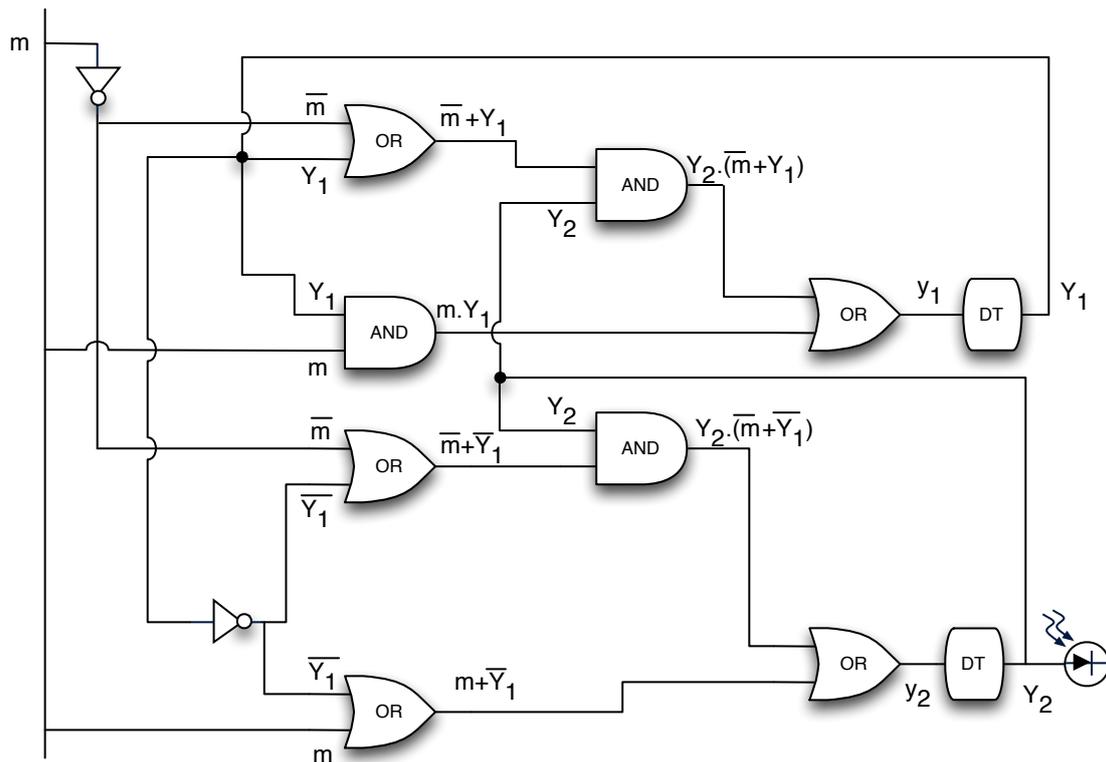
Avec une JKT on peut faire :

- Une DT , avec $J = \overline{K} = D$
- Une T , avec $J = K = 1$
- Une RST en s'interdisant $J = K = 1$

Analyse de circuits séquentiel



Analyse de circuits séquentiel



Analyse de circuits séquentiel

Equations logiques :

$$\begin{cases} y_1 = Y_2 \cdot (\bar{m} + Y_1) + m \cdot Y_1 \\ y_2 = Y_2 \cdot (\bar{m} + \bar{Y}_1) + m \cdot \bar{Y}_1 \\ L = Y_2 \end{cases}$$

Table d'excitation

Excitation

$Y_1 Y_2$		Y_2		Y_1
		0 0	0 1 1 1	
m	0	00 ₀	11 ₁ 11 ₁	00 ₀
	1	01 ₀	01 ₁ 10 ₁	10 ₀

Table d'excitation : état stables

Excitation

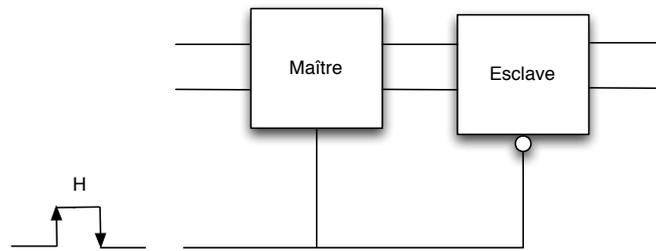
m		$Y_1 Y_2$			
		0 0	0 1	1 1	1 0
m	0	00 ₀	11 ₁	11 ₁	00 ₀
	1	01 ₀	01 ₁	10 ₁	10 ₀

Table des états nommés

Excitation

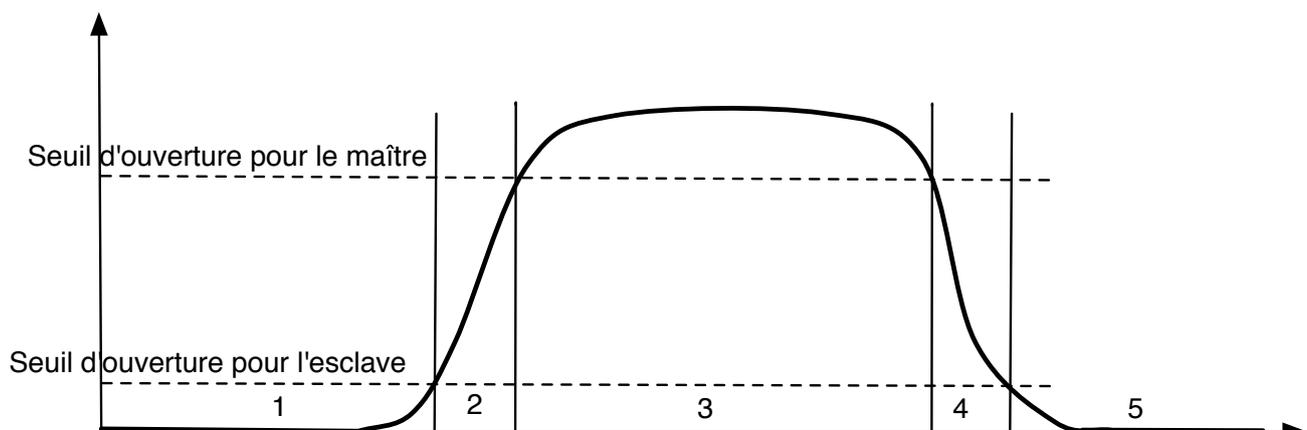
m		$Y_1 Y_2$			
		0 0	0 1	1 1	1 0
m	0	q ₀ /0	q ₃ /1	q ₃ /1	q ₀ /0
	1	q ₁ /0	q ₁ /1	q ₂ /1	q ₂ /0

Les Bascules Maîtres-esclaves

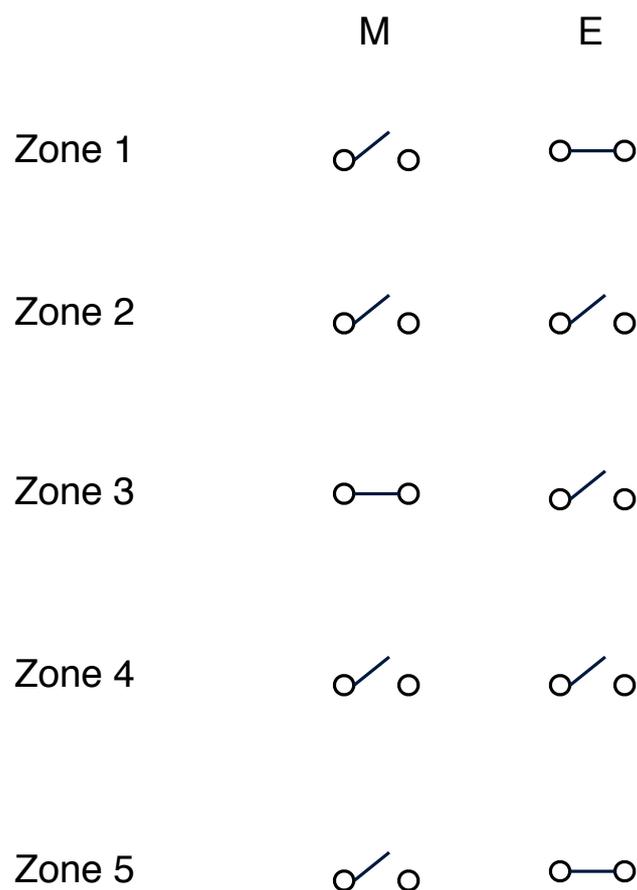


- Composées de 2 bascules synchrones (RS,JK, ou D)
- Maître et l'esclave
- Lecture et écriture de l'état de la bascules sur le même pulse.

La Maître-esclave



La Maître-Esclave

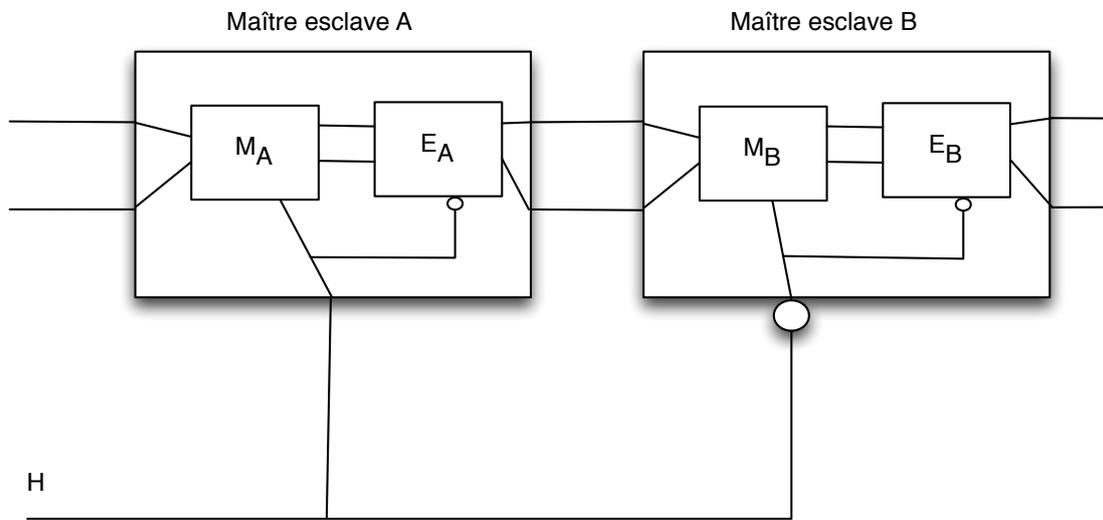


La Maître-Esclave

- Il n'existe pas de configuration où le maître et l'esclave sont passant en même temps et donc :
- On peut reboucler l'entrée et la sortie de la ME et donc :
- On peut implémenter : Echanger i, j ou $i=i+1$

Exemple 1

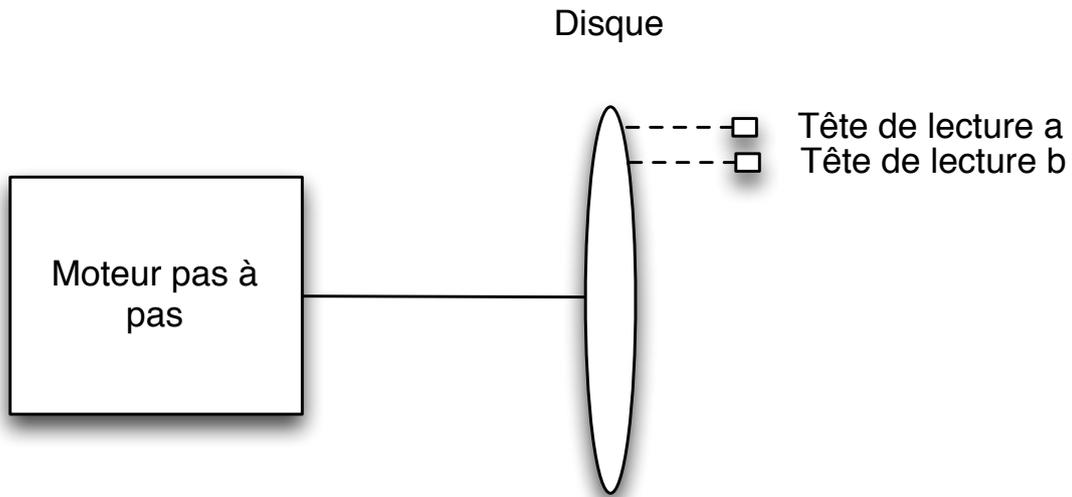
Echange de A et B



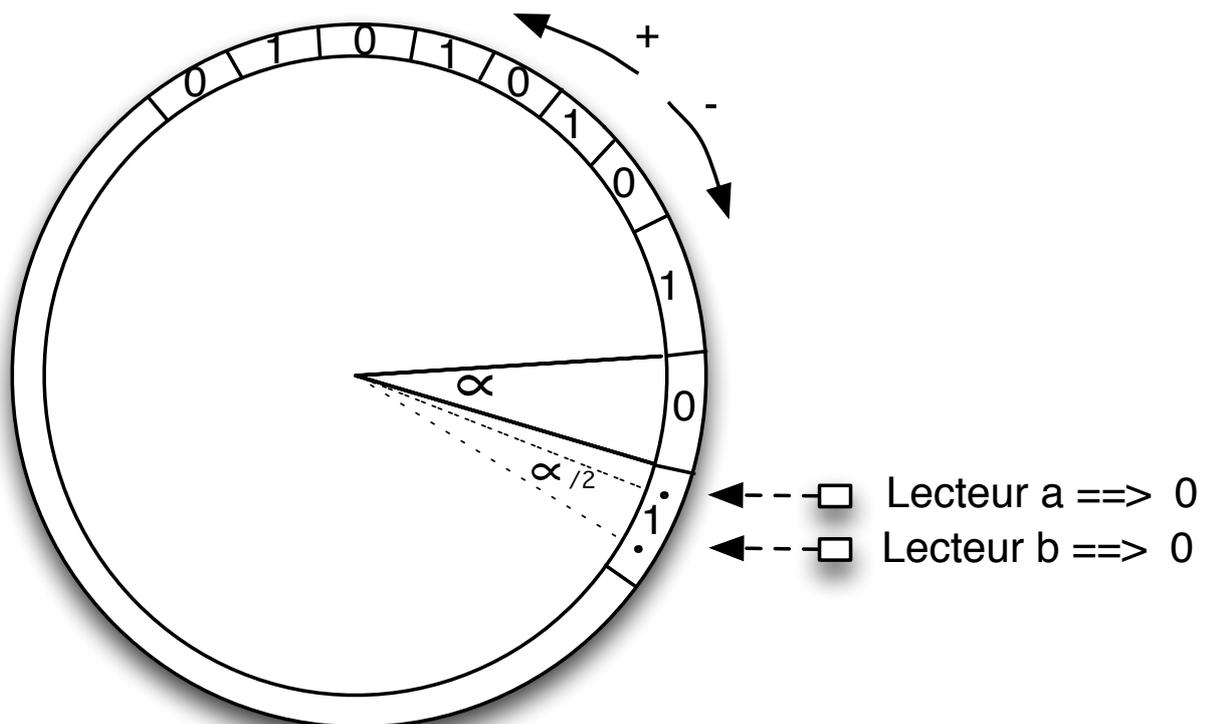
Exemple 1

	M	E	M_A	E_A	M_B	E_B
Zone 1			0	0	1	1
Zone 2			0	0	1	1
Zone 3			1	0	0	1
Zone 4			1	0	0	1
Zone 5			1	1	0	0

Exemple 2

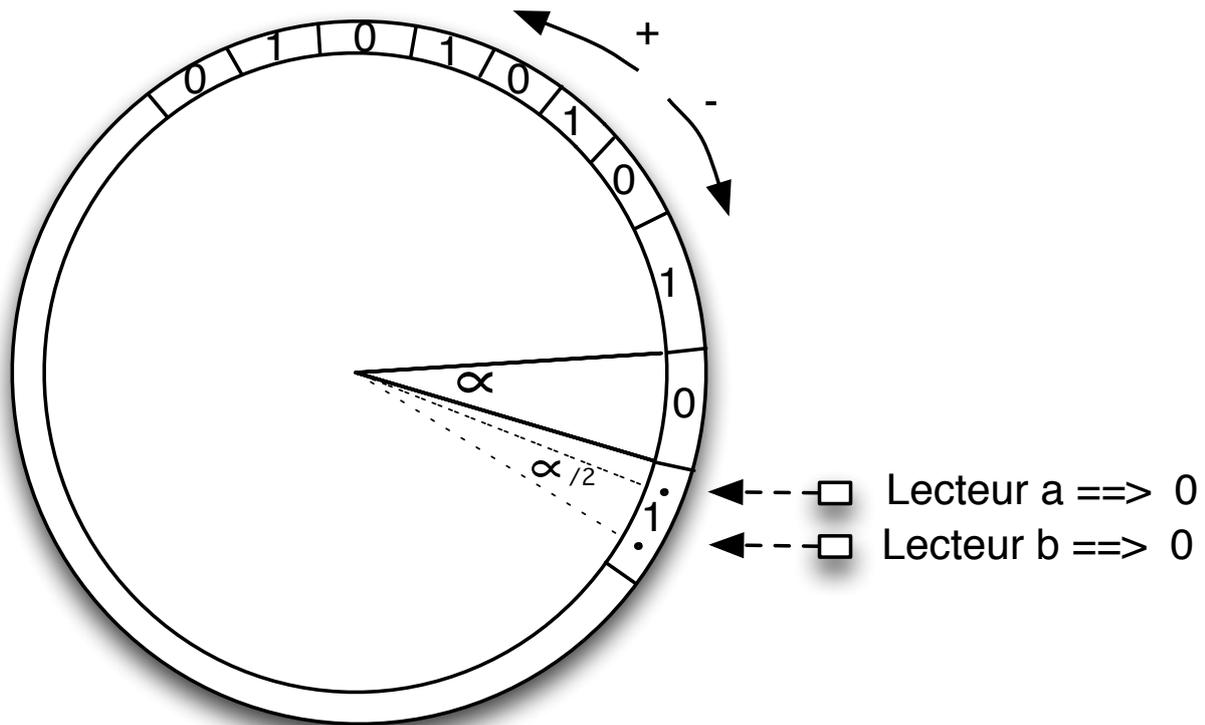


Exemple 2



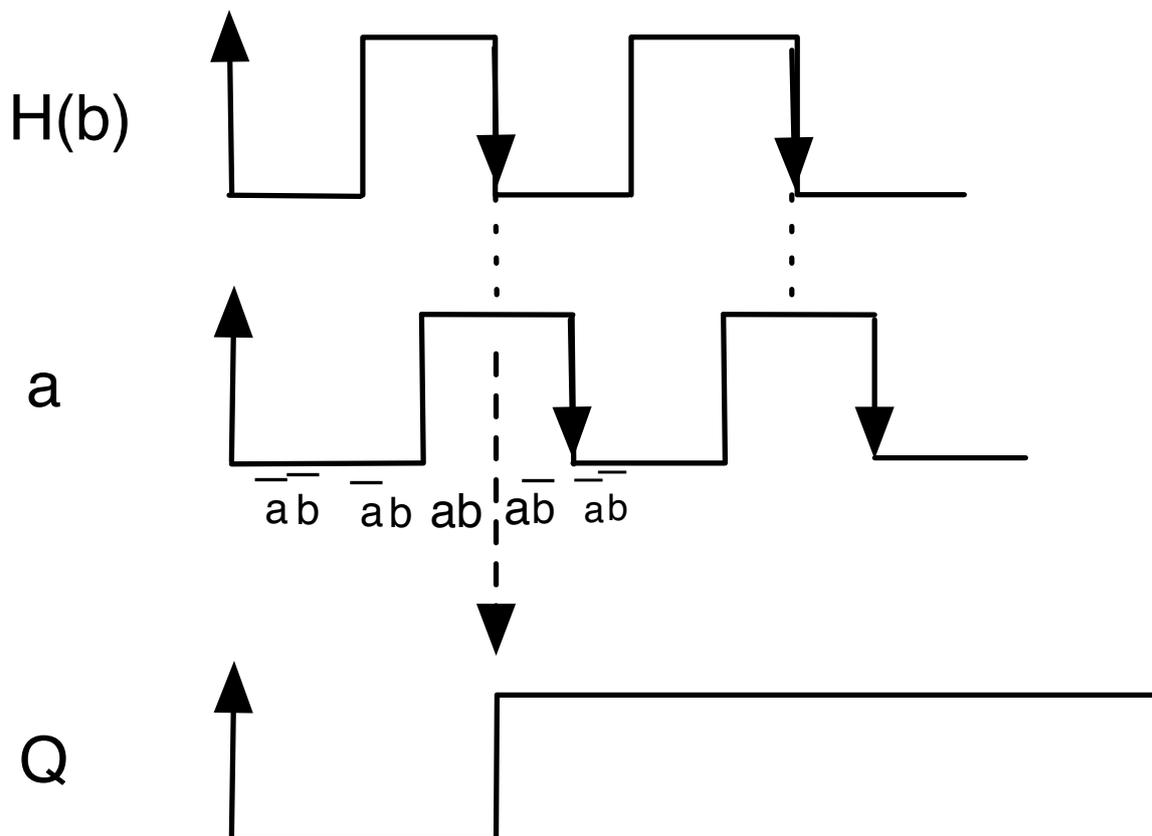
Logique négative : Les lecteurs renvoient 0 lorsqu'ils lisent 1

Exemple 2

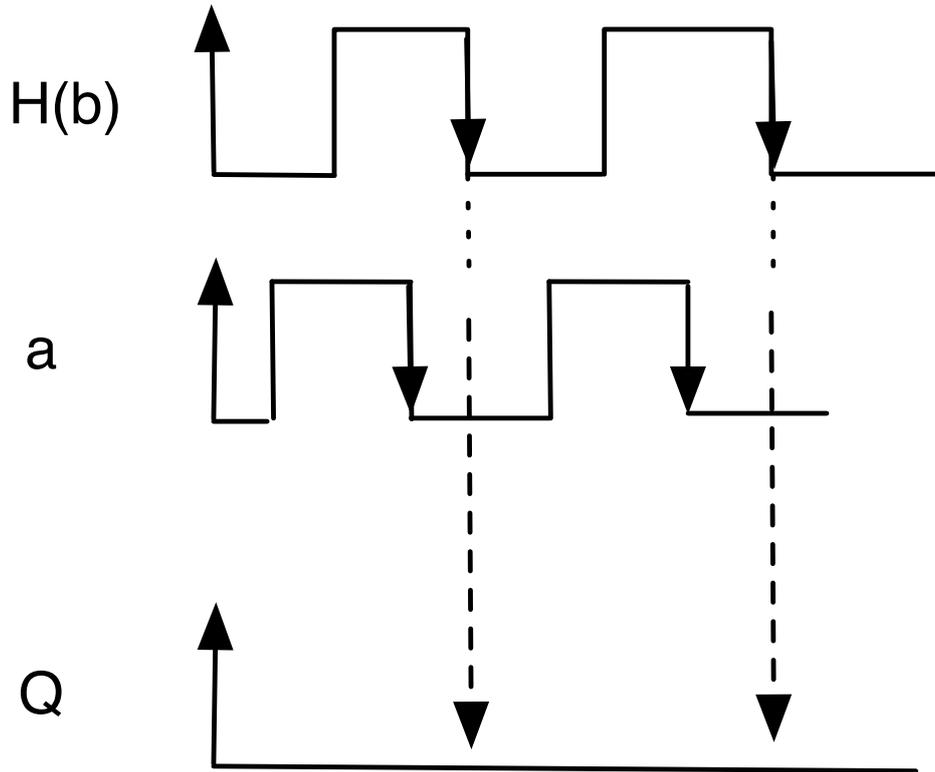


Logique négative : Les lecteurs renvoient 0 lorsqu'ils lisent 1

Exemple 2 : sens Positif



Exemple 2 : sens Négatif



Séquenceur

Séquenceur

Un séquenceur est un système logique séquentiel construit à partir de masques Maîtres-Esclaves qui implémente un automatisme. Cet automatisme peut délivrer des signaux sur des lignes à partir d'une horloge en fonction de son état courant.

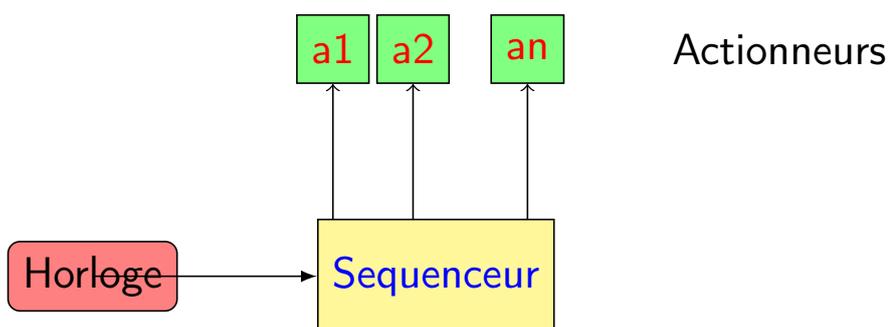


Table d'excitation de la bascule JK

J	K	Q^t	Q^{t+1}
0	*	0	0
1	*	0	1
*	1	1	0
*	0	1	1

- Cette table permet de déterminer le prochain état de la bascule JK .
- Ex : pour l'excitation $1 *$:
Avec $J = 1$ quel que soit l'état de K la bascule passe de l'état 0 à 1.

Exemple

- On veut réaliser un séquenceur qui affiche sur 4 leds la valeur de comptage en binaire.
- Ce compteur est cyclique : 0, 1, 2, 3 9, 0, 1, 2,
- On va réaliser ce séquenceur à l'aide de 4 bascules.

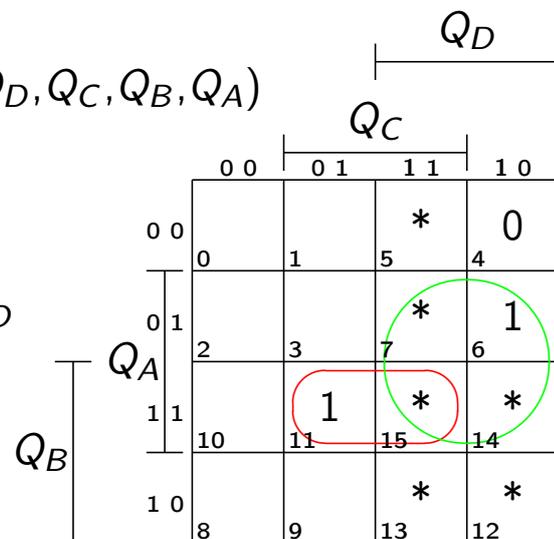
Exemple de séquenceur

etat	Q_D	Q_C	Q_B	Q_A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	*	0	*	0	*	1	*
1	0	0	0	1	0	*	0	*	1	*	*	1
2	0	0	1	0	0	*	0	*	*	0	1	*
3	0	0	1	1	0	*	1	*	*	1	*	1
4	0	1	0	0	0	*	*	0	0	*	1	*
5	0	1	0	1	0	*	*	0	1	*	*	1
6	0	1	1	0	0	*	*	0	0	*	1	*
7	0	1	1	1	1	*	*	1	*	1	*	1
8	1	0	0	0	*	0	0	*	0	*	1	*
0	1	0	0	1	*	1	0	*	0	*	*	1

$J_D K_D$

On pose $J_D = K_D$

$$\begin{aligned}
 J_D &= K_D \\
 &= Q_C \cdot Q_B \cdot Q_A + Q_A \cdot Q_D
 \end{aligned}$$



$J_C K_C$

On pose $J_C = K_C$

$$\begin{aligned} J_C &= K_C \\ &= Q_B \cdot Q_A \end{aligned}$$

$f(Q_D, Q_C, Q_B, Q_A)$

		Q_D			
		0 0	0 1	1 1	1 0
Q_B	0 0	0	0	*	0
	0 1	0	0	*	0
	1 1	1	1	*	*
	1 0	0	0	*	*

Cell indices: (0,0)=0, (0,1)=1, (0,5)=5, (0,4)=4, (1,2)=2, (1,3)=3, (1,7)=7, (1,6)=6, (1,10)=10, (1,11)=11, (1,15)=15, (1,14)=14, (1,8)=8, (1,9)=9, (1,13)=13, (1,12)=12

$J_B K_B$

On pose $J_B = K_B$

$$\begin{aligned} J_B &= K_B \\ &= Q_A \cdot \overline{Q_D} \end{aligned}$$

$f(Q_D, Q_C, Q_B, Q_A)$

		Q_D			
		0 0	0 1	1 1	1 0
Q_B	0 0	0	0	*	0
	0 1	1	1	*	0
	1 1	1	1	*	*
	1 0	0	0	*	*

Cell indices: (0,0)=0, (0,1)=1, (0,5)=5, (0,4)=4, (1,2)=2, (1,3)=3, (1,7)=7, (1,6)=6, (1,10)=10, (1,11)=11, (1,15)=15, (1,14)=14, (1,8)=8, (1,9)=9, (1,13)=13, (1,12)=12

Travaux Dirigés de Logique

D. Delfieu

October 17, 2022

1 NUMERATION

$$2^{-1} = 0,5 \quad 2^{-2} = 0,25 \quad 2^{-3} = 0,125 \quad 2^{-4} = 0,6125$$
$$2^{-5} = 0,03125 \quad 2^{-6} = 0,015625 \quad 2^{-7} = 0,0078125 \quad 2^{-8} = 0,00390625.$$

1.1 Conversions

Base 2 ? Trouver les équivalents décimaux de 1000, dans les bases $B = 8, 3, 2$.

Comparer le nombre de chiffres nécessaires pour exprimer un nombre N quelconque dans les systèmes binaires et octal.

Vers une base Octale Calculer l'équivalent octal des nombres décimaux suivants : 29, 1971, 899.

Vers base Décimale Trouver l'équivalent décimal des nombres suivants : $(44)_8, (587)_8, (A9F)_{16}$

Conversion en base 2 Convertir dans le système binaire les nombres suivants : $(228, 375)_{10}, (61)_8$

Conversion de bases quelconques Calculer l'équivalent binaire du nombre décimal $(7777)_{10}$ en passant par son équivalent octal.

B_1 vers B_2 : Effectuer les changements de base suivant :

Base	Nombre	Equivalent	Dans la base
10	214,420		4
10	3FE		4

Binaire Effectuer les opérations suivantes :

$$(0101011)_2 + (0111011)_2 =$$

$$(7FE)_{16} + (3AB)_{16} =$$

$$(4B9)_{16} + (FFF)_{16} =$$

Précision On veut cadrer une position entre 0 et 15 cm avec une précision supérieure ou égale à 0,1 mm :

Quel est le nombre de bits nécessaires ? Quelle est la précision obtenue finalement ? Donner la relation entre la longueur en mm et le code binaire. Quelle est la longueur qui correspond à $(72C)_{16}$?

1.2 Représentation des nombres négatifs

1.2.1 Nombres signés en "signe + valeur absolue"

Sachant que l'on dispose de 6 e.b. pour écrire les nombres avec leurs signes, donner les expressions en base 2 des nombres suivants en convention signe + valeur absolue : +24, +33, -15, +12, -17

1.2.2 Complément Vrai

En utilisant le Complément Vrai (CV), toujours avec 6 e.b., donner les expressions binaires des nombres suivants : -24, -15, -12

1.2.3 Complément Restreint

Combien faut-il d' e.b. pour réaliser les opérations suivantes : $15 - 3$, $-12 - 4$
Effectuer ces opérations en complément restreint.

1.2.4 Complément Vrai et débordements

Avec les nombres signés sur 9 e.b., en utilisant le complément vrai pour la représentation des nombres négatifs, faire les opérations suivantes :

$$232 - 25, 232 + 25, -232 + 25, -232 - 25$$

Détecter les éventuels débordements, en déduire un circuit de détection.

2 Logique combinatoire

2.1 Portes élémentaires

2.1.1 Logigramme

Dessiner les logigrammes ET-OU-NON des fonctions suivantes :

$$F_1(a, b, c, d, e, f) = a.b + cd + e.f$$

$$F_2(a, b, c, d, e, f) = (a + b).(c + d).(e + f)$$

$$F_3(A, B, C, D) = \bar{A}.\bar{B}.C + D.(\bar{A} + \bar{B})$$

Logigramme NAND Dessiner le logigramme NAND de la fonction suivante :

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

Logigramme NOR Dessiner le logigramme NOR de la fonction suivante :

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

2.1.2 Système logique complet

Montrer que NOR est un système logique complet

2.1.3 OU EXCLUSIF

Matérialiser la fonction "OU EXCLUSIF" de 2 variables a et b au moyen d'un circuit intégré 7400 (4 Nand à 2 entrées).

2.2 Karnaugh

2.2.1 Tables de karnaugh

Construire les tables de Karnaugh des fonctions logiques définies par les tables de vérité suivantes, puis les utiliser pour simplifier les expressions F_1 et F_2 :

a	b	c	d	F_2
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
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	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

A	B	C	F_1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Donner les expressions logiques les plus simples possibles des fonctions définies par les tableaux de KARNAUGH suivantes :

f	<div style="margin-left: 40px;"> $\overbrace{\hspace{10em}}^e$ </div> <div style="margin-left: 20px;"> $\overbrace{\hspace{6em}}^c$ </div> <div style="margin-left: 10px;"> $\overbrace{\hspace{3em}}^d$ </div> <div style="margin-left: 10px;"> $\overbrace{\hspace{3em}}^d$ </div> <table border="1" style="margin-left: 10px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0
0	0	0	0	1	0	0	1																										
1	0	0	1	1	0	0	1																										
1	1	1	1	1	1	1	1																										
0	1	1	0	0	1	1	0																										

$f(a,b,c,d)$	<div style="margin-left: 40px;"> $\overbrace{\hspace{10em}}^a$ </div> <div style="margin-left: 20px;"> $\overbrace{\hspace{6em}}^b$ </div> <table border="1" style="margin-left: 10px;"> <tr> <td style="border: none;"></td> <td style="border: none;">a</td> <td style="border: none;">b</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;">c</td> <td style="border: none;">d</td> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">00</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">10</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">00</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">00</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">01</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">01</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">11</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">11</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">10</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;"></td> <td style="border: none;">10</td> <td style="border: none;">01</td> <td style="border: none;">11</td> </tr> </table>		a	b			c	d						00	01	11			10					00	01	11			00	01	11			01	01	11			01	01	11			11	01	11			11	01	11			10	01	11			10	01	11
	a	b																																																											
c	d																																																												
		00	01	11																																																									
		10																																																											
		00	01	11																																																									
		00	01	11																																																									
		01	01	11																																																									
		01	01	11																																																									
		11	01	11																																																									
		11	01	11																																																									
		10	01	11																																																									
		10	01	11																																																									

		c	
	b		
	1	0	0
a	1	1	0

2.2.2 Lecture de table sur les zéros

On considère la fonction logique $F(a, b, c, d)$ définie par le tableau de Karnaugh suivant :

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

- Donner son expression sous la forme d'une somme de produits (à partir des "1" du tableau)
- Donner son expression sous la forme d'un produit de sommes (à partir des "0" du tableau puis par complémentation).
- Vérifier l'égalité de ces expressions.

2.3 Démonstrations algébriques

2.3.1 Egalités logiques

Démontrer les égalités logiques suivantes :

1. $a + \bar{a}b = a + b$
2. $(\bar{a} + b)(a + c) = a.b + \bar{a}c$
3. $(a + \bar{b})(b + \bar{c})(c + \bar{a}) = (\bar{a} + b)(\bar{b} + c)(\bar{c} + a)$

2.3.2 Complémentation (Théorème de Morgan)

Calculer les fonctions complémentaires des fonctions suivantes :

$$F_1 = a.(\bar{b} + c + d) + \bar{c}d$$

$$F_2 = (a + b).(b + c).(c + a)$$

Vérifier que les fonctions complémentaires des fonctions obtenues sont les fonctions données.

Simplifier les expressions logiques suivantes :

$$F_1 = a.b + \bar{c} + c.(\bar{a} + \bar{b})$$

$$F_2 = (a + b + c)(a + \bar{b} + c)(a + \bar{b} + \bar{c})$$

$$F_3 = (a + b)(a + c) + (b + c)(b + a) + (c + a)(c + b)$$

2.3.3 Simplification d'expressions logiques : comparaison de méthodes

Obtenir une expression plus simple des fonctions logiques suivantes, soit par le calcul booléen, soit en construisant leur tables de karnaugh :

$$F_1(A, B, C) = ABC\bar{C} + ABC + A\bar{C} + A\bar{B}C + \bar{B}C$$

$$F_2(A, B, C, D) = \bar{A}D + BC + ABD + ABC\bar{D}$$

$$F_3 = (a + b)(a + c) + (b + c)(b + a) + (c + a)(c + b)$$

2.4 Exercices

2.4.1 Serrure de coffre

Quatre responsables d'une société (A, B, C, D) peuvent avoir accès à un coffre. Ils possèdent chacun une clé différente (a, b, c, d) et il a été convenu que :

- A ne peut ouvrir le coffre que si au moins un des responsables B ou C est présent.
- B, C, D ne peuvent l'ouvrir que si au moins deux des autres responsables sont présents.

Donner l'équation logique de la serrure du coffre (S) en fonction de a, b, c, d .

2.4.2 Amplification sonore

Les trois haut-parleurs d'une salle de cinéma (soient a, b, c) sont branchés sur un amplificateur qui a deux sorties : une d'impédance 4Ω (soit S_4) et une autre d'impédance 8Ω (soit S_8).

- Lorsqu'un seul haut-parleur est utilisé, il doit être relié à la sortie de 8Ω .
- Lorsque deux haut-parleurs sont utilisés, ils doivent être reliés tous les deux à la sortie de $S_4 \Omega$.
- Le fonctionnement simultané des trois haut-parleurs est interdit.

Donner les équations logiques des sorties S_4 et S_8 en fonction de a, b, c .

2.4.3 Circuit de vote

Quatre délégués syndicaux représentent respectivement le nombre de voix suivants :

- $a = 100$ voix,
- $b = 150$ voix,
- $c = 250$ voix,
- $d = 175$ voix.

Pour être acceptée lors des réunions, une proposition doit recueillir au moins 50 % des voix représentées.

Donner l'équation logique d'un circuit S à 4 entrées a, b, c, d dont la valeur logique soit "1" lorsqu'une proposition est acceptée "0" lorsqu'elle est refusée.

2.4.4 Deviner un nombre

Un système devine la valeur d'un nombre compris entre 0 et 7 d'après les réponses à 3 questions :

- Le chiffre est-il impair ? Oui $A = 1$ Non $A = 0$
- La valeur de $2 * (N + 1) > 8$ Oui $B = 1$ Non $B = 0$
- Le reste de la division de $(N + 10)/4$ est-il égal à 1 ou 2 OUI $C = 1$ NON $C = 0$

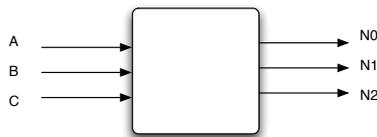


Figure 1: système qui devine un nombre

2.5 Problèmes

2.5.1 Décodeur

On souhaite réaliser un comparateur de deux nombres A et B codés en binaire. Soient A_0 et A_1 les bits du nombre A , et B_0 et B_1 les bits du nombre B (0 est l'indice de poids faible). Le comparateur aura 3 sorties SG , SE et SP telles que :

- $SG = 1$ si $A > B$
- $SE = 1$ si $A = B$
- $SP = 1$ si $A < B$

Etudier la réalisation de ce comparateur à partir d'un décodeur 4 entrées 16 sorties :



Figure 2: A1,B1 bits de poids fort

2.5.2 Multiplexeur

On considère la fonction suivante :

$$F_2 = \bar{a}cd + b\bar{c}.(e + a\bar{f}) + d\bar{f}.(\bar{a}e + \bar{e}a) + ab\bar{c} \bar{e}f$$

Matérialiser la fonction F à l'aide d'un multiplexeur 16 entrées et 4 entrées d'adresses :

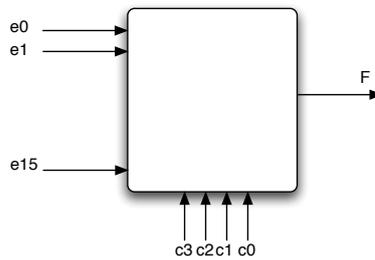


Figure 3: e_{15} , c_3 bits de poids fort

2.5.3 Multiplication combinatoire 2*2 bits

- Un chiffre A est codé en binaire sous la forme de 2 bits A_1 et A_0 (A_0 poids faible)
- Idem pour un second chiffre B codé en binaire sous la forme de 2 bits B_1 et B_0 (B_0 poids faible)
- On souhaite visualiser le résultat $A * B = C$ codé en binaire sur 4 bits C_3, C_2, C_1 et C_0
- Chercher les équations de C_3, C_2, C_1 et C_0
- Simplifier les équations de C_3, C_2, C_1 et C_0
- Proposer un schéma avec cellules NAND et OR pour C_2 et C_3 .

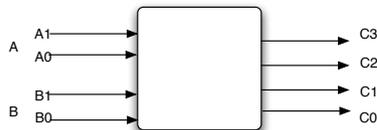


Figure 4: Multiplieur 2*2

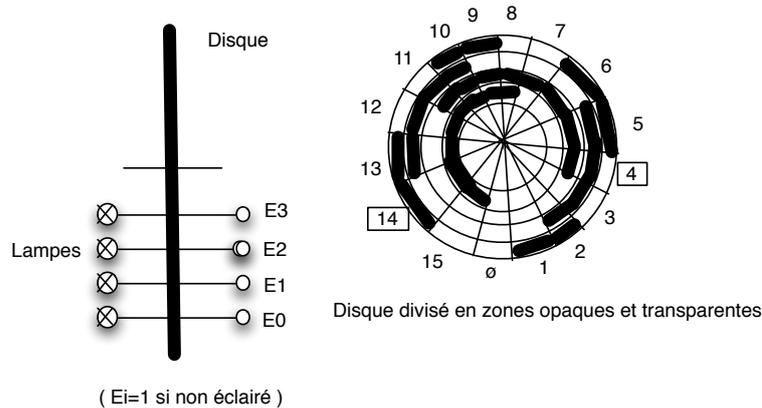


Figure 5: E_0 est noté comme le bit le plus à l'extérieur du disque

2.5.4 Codeur optique

Un codeur optique est composé d'un disque à secteur opaques et transparent et de 4 photo-détecteurs. On dispose de 4 informations (E_0 , E_1 , E_2 , E_3).

Réaliser un dispositif fournissant l'information de position en code BCD (Décimal Codé binaire).

Exemple Pour la position 2 on a : $E_3 = 0$, $E_2 = 0$, $E_1 = 1$, $E_0 = 1$.

Et le code DCB correspondant est $S_4 = 0$, $S_3 = 0$, $S_2 = 0$, $S_1 = 1$, $S_0 = 0$

En effet S_0 , S_1 , S_2 , S_3 code les unités et S_4 la dizaine.

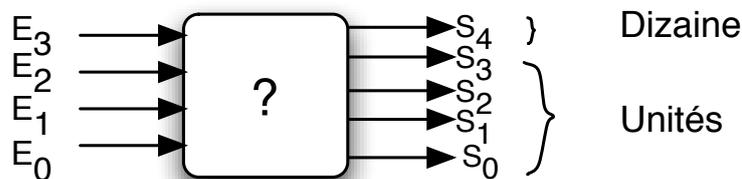


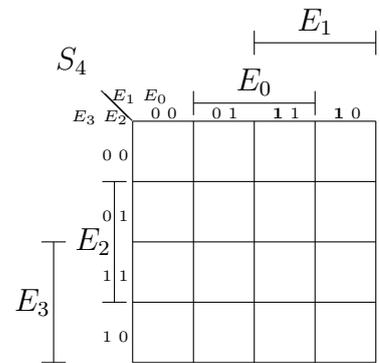
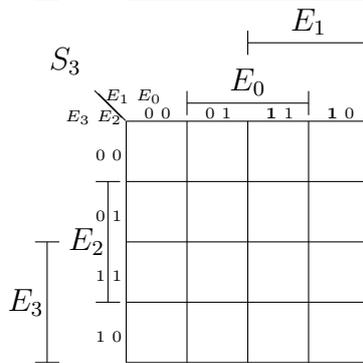
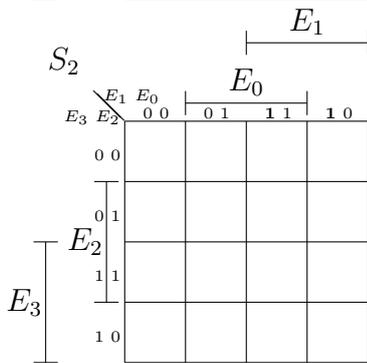
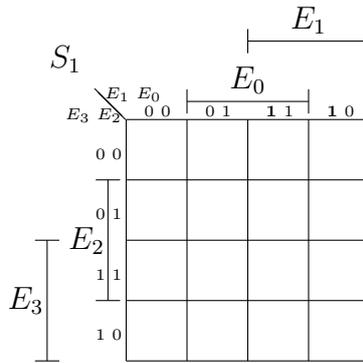
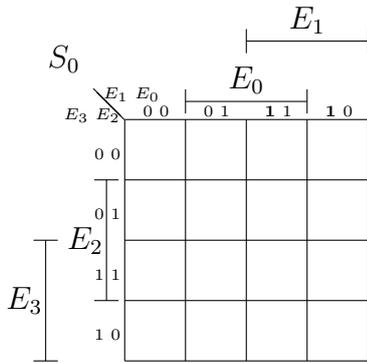
Figure 6: E_3 et S_4 sont les poids Forts

1. Tracer les schémas de S_1 , S_2 , S_3 , S_4 avec des portes *NAND*
2. S_0 sera matérialisé par un multiplexeur $8 \rightarrow 1$ (le 74151) avec E_2 , E_1 , E_0 pour les entrées d'adresses.

Tables :

<i>Pos</i>	<i>E3</i>	<i>E2</i>	<i>E1</i>	<i>E0</i>	<i>S4</i>	<i>S3</i>	<i>S2</i>	<i>S1</i>	<i>S0</i>
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11	1	1	1	0	1	0	0	0	1
12									
13									
14									
15									

Tables de Karnaugh :



2.5.5 L'afficheur 7-segments

L'afficheur 7-segments *TIL 302* est un circuit intégré formé de 7 diodes lumineuses en forme de bâtonnets permettant de représenter tout chiffre (0, ..., 9) sous la forme suivante : On appelle K_a, \dots, K_g les fonctions d'allumage des segments a, \dots, g . Notez que la mise à la masse de K_a c.a.d. $K_a = 0$ permet d'allumer le segment a . L'allumage d'un segment se fait par mise à zéro de la cathode K qui joue le rôle d'entrée, l'anode (+) étant à + 5v.



Figure 7: Forme des chiffres

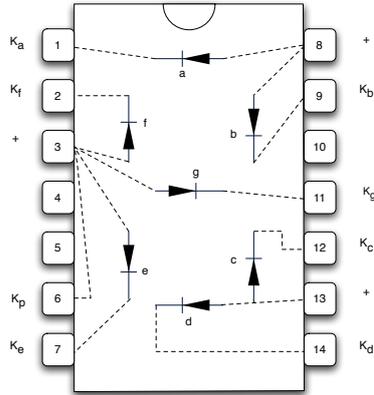


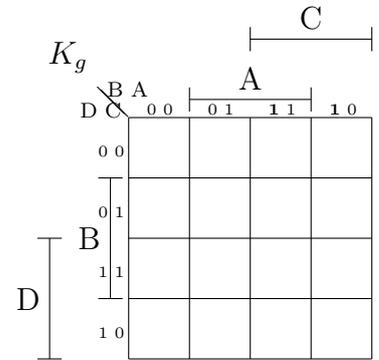
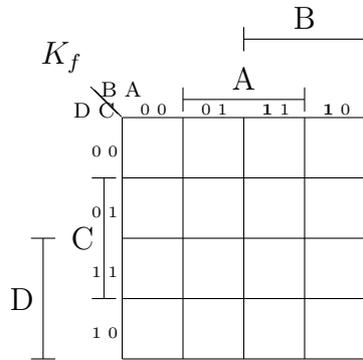
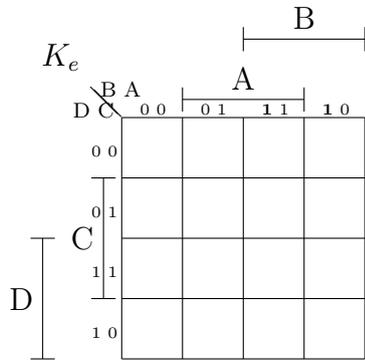
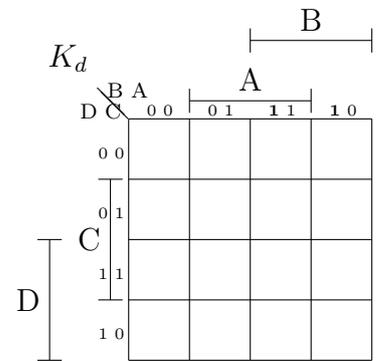
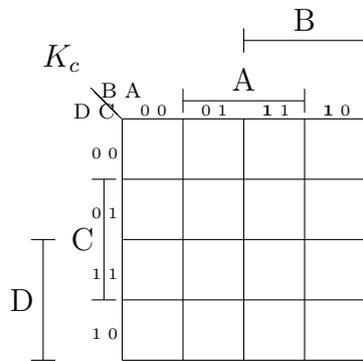
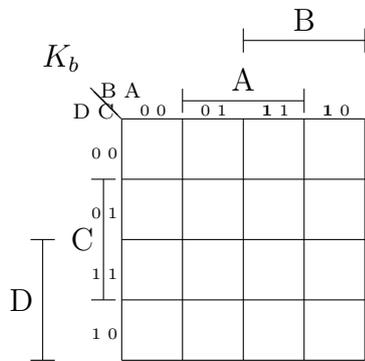
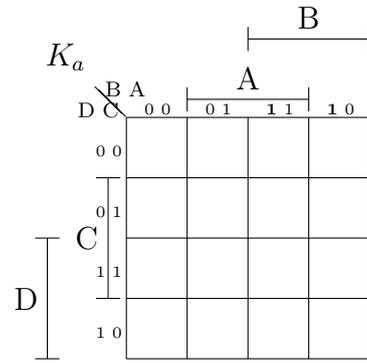
Figure 8: Le composant TIL 302

En fonction d'un nombre exprimé sur les entrées A, B, C, D , exprimer les équations de K_a, \dots, K_g .

<i>Entres</i>	<i>Fonction</i>
1	cathode a
2	cathode f
3	+ 5v
6	cathode du point
7	cathode e
8	cathode d
9	+ 5v
10	cathode c
11	cathode g
13	cathode b
14	+ 5v

Ecrire la table de vérité des fonctions, simplifier à l'aide des tables de Karnaugh, puis dessiner le logigramme du circuit en utilisant des portes $ET - NON$.

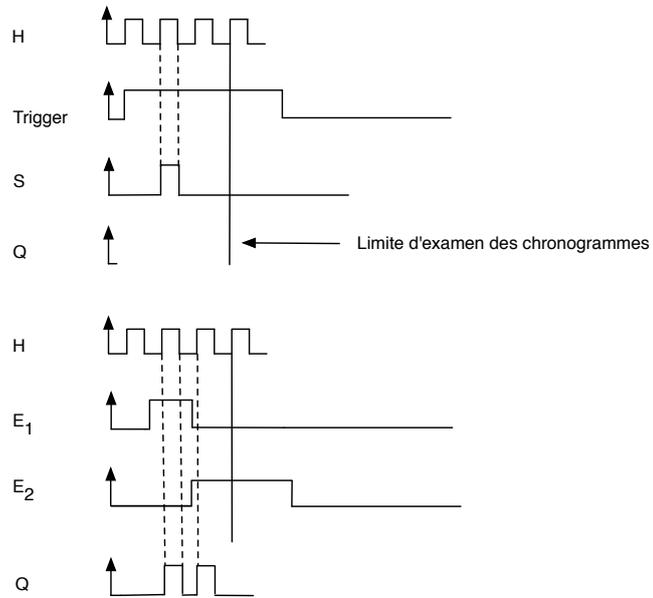
Nombre	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							
10	1	0	1	0							
11	1	0	1	1							
12	1	1	0	0							
13	1	1	0	1							
14	1	1	1	0							
15	1	1	1	1							



3 Logique Séquentielle

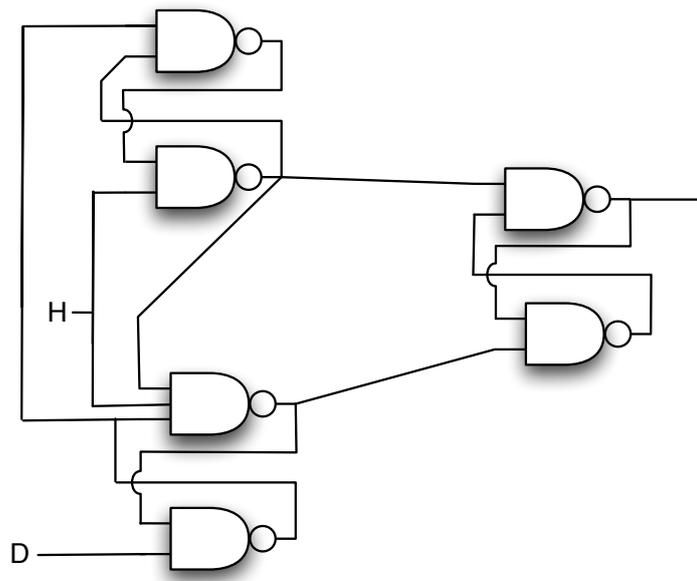
3.1 système séquentiel ou combinatoire ?

Les systèmes dont les entrées-sorties sont représentées par les chronogrammes suivant sont-ils séquentiel ou combinatoire ?



3.2 Analyse de circuits séquentiels

A partir du schéma, donner la machine de Moore explicitant le fonctionnement.

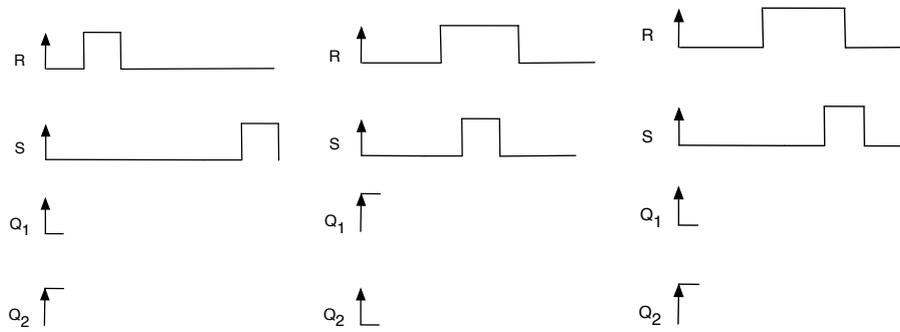
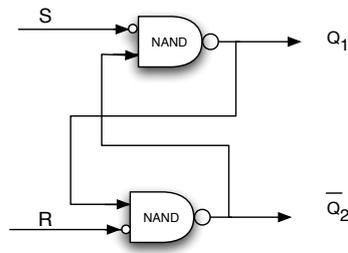


3.3 Les bascules

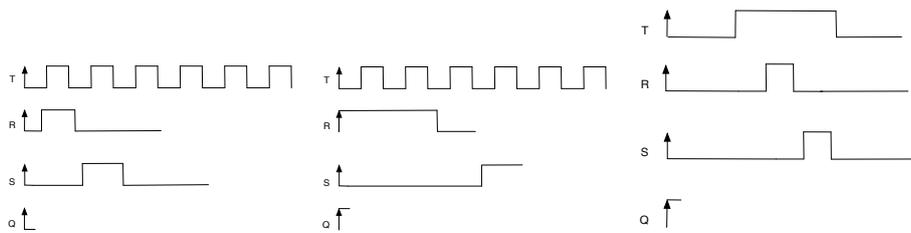
3.3.1 La bascule RS

Considérons la bascule RS asynchrone suivante :

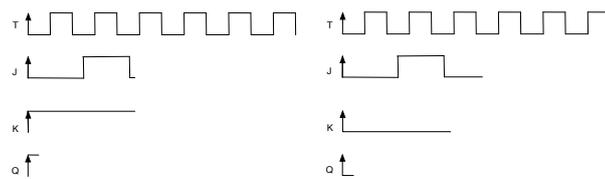
Bascule RS Donner dans les cas suivant les chronogrammes des sorties Q_1 et Q_2 :



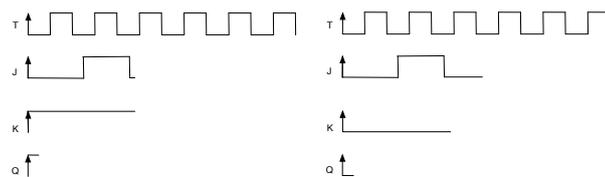
Bascule *RST* Donner dans les cas suivant les chronogrammes de la sortie *Q*:



Bascule *JK* synchronisée sur front montant Donner dans les cas suivant les chronogrammes de la sortie *Q*:

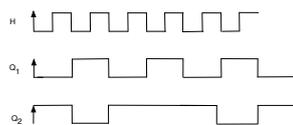


Bascule Maître-Esclave Donner dans les cas suivant les chronogrammes de la sortie *Q*:



3.4 Les séquenceurs

Réaliser le séquenceur correspondant au chronogramme suivant :



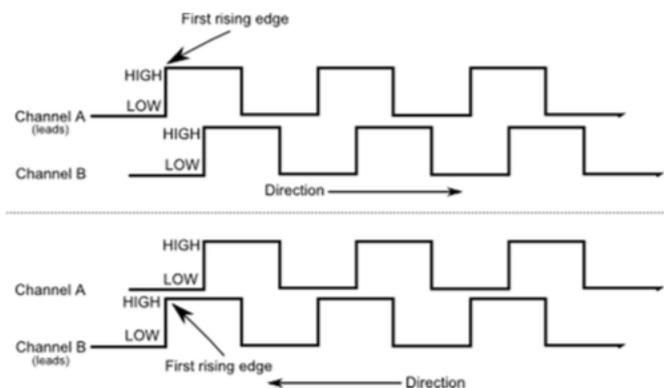
3.4.1 Les Compteurs/Décompteurs

Réaliser le séquenceur correspondant à la séquence :1, 2, 4, 15, 11, 13, 3, 0, 1, ...

3.4.2 Le Club Alpin Français

Une enseigne publicitaire pour le "Club Alpin Français" doit afficher successivement sur un afficheur 7-segments les lettres *C*, *A* et *F*. Proposer un schéma en *JK*.

3.4.3 Encodeur Quadratique



Un système comporte 2 entrées *A* et *B*, une une sortie *Direction*. A l'aide de deux bascules *JK* réaliser un encodeur quadratique : Il donne le sens en fonction de la succession de séquences d'états.

Travaux Pratiques de Logique

D. DELFIEU

January 3, 2023

Part I

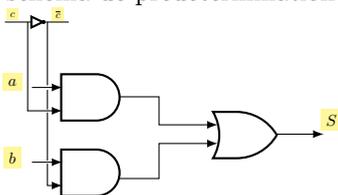
TP LOGIQUE COMBINATOIRE

1 Rédaction du compte rendu

Un compte rendu de travaux pratiques doit faire l'objet d'une introduction, de schémas accompagnés de légendes et de remarques et d'une conclusion générale. Elle peut évoquer les acquis apportés par le TP, les applications industrielles (ou dans la vie courante), les limites, une prise de recul.

2 Prédétermination

Un schéma de prédétermination doit ressembler à :



Il comporte des portes logiques et non des circuits intégrés !

2.1 Introduction

Le but de ces travaux est de se familiariser avec les différents circuits de base de la logique TTL-LS en traitant des exemples simples. Pour réaliser ces différents traitements on utilisera les composants dont la documentation est fournie en annexe. Ces travaux pratiques s'appuient sur la théorie de la logique combinatoire notamment sur les notions de décodeur et de mise en forme d'une expression logique en terme d'opérateurs spécifiques (Exclusiv-OR, *Nand*). Il y a 2 points accordés à la présentation.

2.2 Prédétermination

Pour cette étude il sera adjoint une représentation logique sous forme de portes logiques.

2.3 Décodeur 2/4

Donner des schémas de réalisation d'un décodeur 2/4. Un des montages utilisera le 74LS138 et l'autre des portes et, ou, non.

2.4 Générateur de parité

Après avoir étudié l'annexe I :

- Etablir l'équation logique d'un générateur de parité à 3 entrées.
- Dessiner le montage avec des portes , *Or* et *Not*, puis faire de même avec de portes Exclusive-Or.
- Etablir l'équation logique d'un circuit de détection d'erreur de parité à 4 entrées (a,b,c, parité)
- Réaliser le montage avec des portes Exclusive-OR
- Dans quel cas peut on mettre en défaut le détecteur ?

2.5 Additionneur

Après avoir étudié les annexes démontrer que :

- $S = A \oplus b \oplus R_{n-1}$
- $R_n = ab + (a \oplus b)R_{n-1}$

Réaliser un schéma avec des *Nand* et *Not*.

2.6 Sortie Totem Pole

Etude des sorties Totem Pole des 74LSxx. La sortie des portes logiques peut se schématiser ainsi :

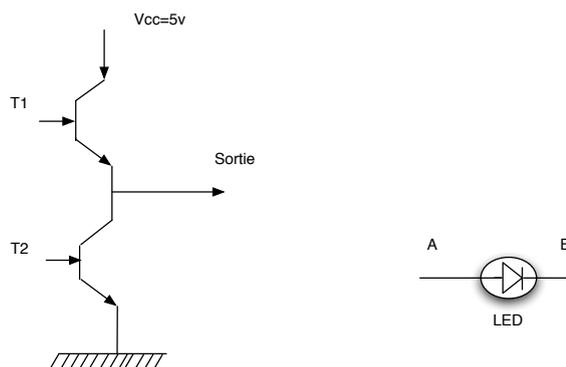


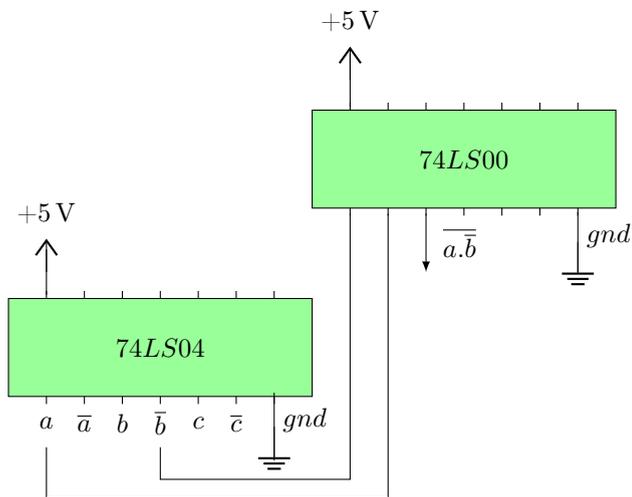
Figure 1: Transistor

- Si T_1 est bloqué et T_2 saturé alors $\text{Sortie} = 0$
- Si T_1 est saturé et T_2 bloqué alors $\text{Sortie} = 1$

Afin de visualiser l'état d'une sortie, nous disposons d'une *LED* et d'une résistance. En logique positive, proposer un montage simple permettant l'émission de la *LED* lorsque la sortie de la porte est au niveau haut. En logique négative proposer un montage permettant l'émission de la *LED* lorsque la sortie de la porte est au niveau bas.

3 MANIPULATION

Faites ressortir distinctement les entrées-sorties des montages sur la plaquette de simulation. En cas de dysfonctionnement, faites vérifier vos montages par le responsable du laboratoire. Tous schéma de manipulation comporte toutes les connexions (V_{cc} , Gnd , LED , ...), de plus, toute connexion doit comporter un label ex $a\bar{b}$. Il devra ressembler à :



Dans la partie manipulation, tout montage doit être accompagné de commentaires et des remarques qui s'y rattachent :

- Donnez aussi la méthode de validation du (des) résultat(s)
- Limite ou défaut du montage
- Utilisation possible du montage

3.1 Liste des composants

Vous disposez des composants suivant :

- 3 * 74LS00, 1 * 74LS04, 2 * 74LS08
- 1 * 74LS32, 1 * 74LS86, 1 * 74LS138

Vérifier votre liste et identifier les fonctions logiques associées aux composants.

3.2 DECODEUR 2/4

Réaliser le montage du décodeur en portes et, ou, non.

3.3 Générateurs de parité

Réaliser le générateur de parité avec :

- Montage 1 : Composés de portes et, ou, *Not*.
- Montage 2 : Ne comportant qu'un seul circuit intégré

3.4 Circuit de détection d'erreur

Réaliser le circuit de détection d'erreur de parité et tester son fonctionnement.

3.5 Additionneur complet

Les équations d'un additionneur complet sont les suivantes :

$$\begin{aligned}
 S &= (a \oplus b) \oplus r_{n-1} \\
 r_n &= a.b + r_{n-1}.(a \oplus b)
 \end{aligned}$$

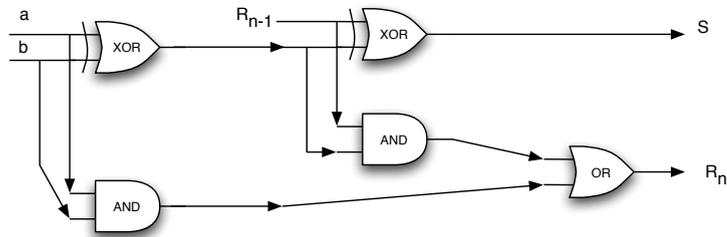


Figure 2: Additionneur complet

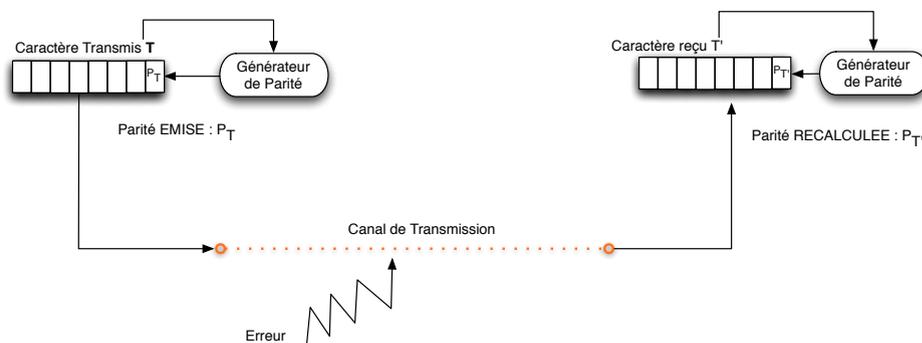
- Etablir $F = a \oplus b$ en *Nand*,
- Donnez les équations de r_n et S en *Nand*.
- Réaliser et tester cet additionneur complet 1 bit.

Annexe I : Générateur de parité

La probabilité d'une erreur dans le traitement, l'enregistrement ou la transmission d'informations numériques est faible mais non nulle. On peut utiliser pour pallier à ces erreurs des codes permettant de détecter les erreurs voir de les corriger de façon automatique.

Un moyen de détecter une erreur dans la transmission d'un caractère (7 bits en ASCII) est de connaître et de transmettre sa parité. Avant la transmission, on ajoute aux 7 bits de données un bit qui représente la parité : 1 (resp. 0) si le nombre de bits à un est impair (resp pair). Cela donne un octet : 7 bits de data + 1 bit de parité.

Exemple :



Il y a erreur si P_T , la parité transmise calculée sur le caractère T, n'est pas égale à la parité re-calculée $P_{T'}$, calculée sur le caractère reçu T'.

Annexe II : Datasheets


June 1989

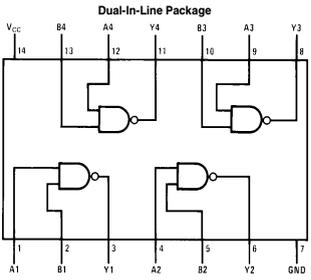
54LS00/DM54LS00/DM74LS00 Quad 2-Input NAND Gates

General Description
This device contains four independent gates each of which performs the logic NAND function.

Features

- Alternate Military/Aerospace device (54LS00) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6433-1

Order Number 54LS00DMQB, 54LS00FMQB, 54LS00MQB, DM54LS00J, DM54LS00W, DM74LS00M or DM74LS00N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H – High Logic Level
L – Low Logic Level

54LS00/DM54LS00/DM74LS00 Quad 2-Input NAND Gates


June 1989

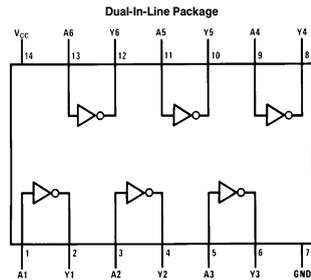
54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

General Description
This device contains six independent gates each of which performs the logic INVERT function.

Features

- Alternate Military/Aerospace device (54LS04) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6345-1

Order Number 54LS04DMQB, 54LS04FMQB, 54LS04MQB, DM54LS04J, DM54LS04W, DM74LS04M or DM74LS04N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H – High Logic Level
L – Low Logic Level

54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

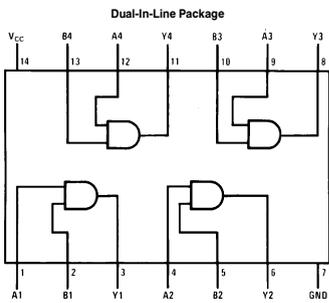
General Description

This device contains four independent gates each of which performs the logic AND function.

Features

■ Alternate Military/Aerospace device (54LS08) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6347-1

Order Number 54LS08DMQB, 54LS08FMQB, 54LS08LMQB, DM54LS08J, DM54LS08W, DM74LS08M or DM74LS08N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = AB$$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level
L = Low Logic Level

54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

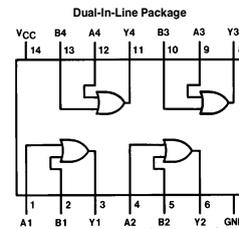
General Description

This device contains four independent gates each of which performs the logic OR function.

Features

■ Alternate Military/Aerospace device (54LS32) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6361-1

Order Number 54LS32DMQB, 54LS32FMQB, 54LS32LMQB, DM54LS32J, DM54LS32W, DM74LS32M or DM74LS32N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = A + B$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = High Logic Level
L = Low Logic Level

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

DM74LS86 Quad 2-Input Exclusive-OR Gate

General Description

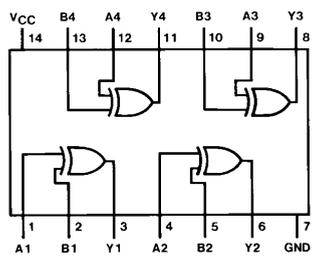
This device contains four independent gates each of which performs the logic exclusive-OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS86M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS86SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS86N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level



March 1998

DM74LS138, DM74LS139 Decoders/Demultiplexers

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

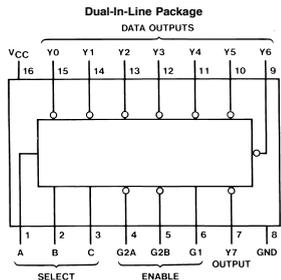
All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit.

All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

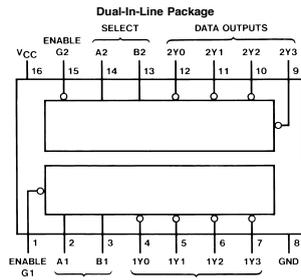
Features

- Designed specifically for high speed:
 - Memory decoders
 - Data transmission systems
- LS138 3-to-8-line decoders incorporates 3 enable inputs to simplify cascading and/or data reception
- LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
 - LS138 21 ns
 - LS139 21 ns
- Typical power dissipation
 - LS138 32 mW
 - LS139 34 mW
- Alternate Military/Aerospace devices (54LS138, 54LS139) are available. Contact a Fairchild Semiconductor Sales Office/Distributor for specifications.

Connection Diagrams



Order Number 54LS138DMQB, 54LS138FMQB, 54LS138LMQB, DM54LS138J, DM54LS138W, DM74LS138M or DM74LS138N
See Package Number E20A, J16A, M16A, N16E or W16A
Dual-In-Line Package



Order Number 54LS139DMQB, 54LS139FMQB, 54LS139LMQB, DM54LS139J, DM54LS139W, DM74LS139M or DM74LS139N
See Package Number E20A, J16A, M16A, N16E or W16A

DM74LS138, DM74LS139 Decoders/Demultiplexers

LS139 Switching Characteristics

at $V_{CC} = 5V$ and $T_A = 25^\circ C$

Symbol	Parameter	From (Input) To (Output)	$R_L = 2\text{ k}\Omega$				Units
			$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Select to Output		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Select to Output		27		40	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Output		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Output		24		40	ns

Function Tables LS138

Inputs		Outputs										
Enable		Select			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
G1	G2 (Note 8)	C	B	A								
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	L	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	L	H	H	H	H	H	H	L

H = High Level, L = Low Level, X = Don't Care
Note 8: $G2 = G2A + G2B$

LS139

Inputs		Outputs					
Enable		Select		Y0	Y1	Y2	Y3
G	B	A					
H	X	X	H	H	H	H	H
L	L	L	L	H	H	H	H
L	L	H	H	L	H	H	H
L	H	L	H	H	L	H	H
L	H	H	H	H	H	L	L

H = High Level, L = Low Level, X = Don't Care

Part II

TP LOGIQUE SEQUENTIELLE

1 Rédaction du compte rendu

Dans la partie manipulation, tous les montages doivent être accompagnés de commentaires :

- Procédure de test
- Limite ou défaut du montage
- Utilisation possible du montage

Tous schéma de manipulation comporte toutes les connections (V_{cc} , Gnd , LED , ...), de plus, toute connection doit comporter un label ex $Q_0\overline{Q_1}$.

Un compte rendu de travaux pratiques doit faire l'objet d'une conclusion générale. Elle peut évoquer les acquis apportés par le TP, les applications industrielles (ou dans la vie courante), les limites, une prise de recul.

2 Prédétermination

2.1 Introduction

Le but de cette manipulation est de se familiariser avec les différents circuits de logique séquentielle de type JK , RS et D ainsi que de compteurs programmables sur des exemples simples : Compteur à cycle quelconque et diviseurs de fréquence.

2.2 PRÉDÉTERMINATION

2.2.1 Compteur à cycle quelconque

On utilise quatre bascules JK (74LS73) pour la réalisation d'un compteur à cycle quelconque 4 bits (Q_0 , Q_1 , Q_2 , Q_3). La séquence à réaliser est la suivante :

$$0, 8, 12, 14, 7, 11, 13, 6, 3, 9, 4, 10, 5, 2, 1, 0$$

- Donner la table des entrées de la bascule JK
- Donner le tableau où sont représentés les états Q_i ainsi que les entrées J_i et K_i en fonction de chaque pas du compteur
- Donner les équations simplifiées des J_i et K_i : Au lieu de poser $J_i = K_i^1$, on pourra choisir de simplifier en posant $K_i = \overline{J_i}$ ce qui donnera des équations plus simples ! Dans ce cas on calcule d'abord l'équation de J_i puis on pose $K_i = \overline{J_i}$
- Donner le schéma complet du compteur
- Implanter un circuit de décodage de la combinaison interdite. Ce circuit permettra de remettre le compteur dans son cycle si par erreur cette combinaison apparaît.

2.2.2 Grafct

Rechercher sur internet ce que représente un *Grafct*. Rédiger alors avec vos propres mots, un résumé des principes de base (min 1 page, max 2 pages).

Réaliser ce grafct à partir de bascules RS et de portes et *Or*.

¹Comme on l'a vue en TD

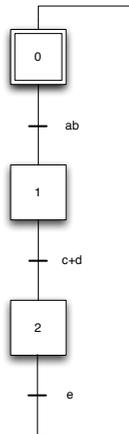


Figure 3: Grafcet

3 MANIPULATION

- Faites ressortir distinctement les entrées/sorties des montages sur la plaquette de simulation.
- Faites vérifier vos montages par un des intervenants de travaux pratiques
- Représenter sur votre compte-rendu le câblage complet (tel que vous l'avez réalisé) des boîtiers pour chaque application, dans vos schémas labelliser les liaisons.

3.1 Liste de composants

Vous disposez des composants :

- 1 * 74LS04 , 1 * 74LS08, 1 * 74LS32, 1 * 74LS86
- 2 * 74LS73, 1 * 74LS74

Vérifier votre liste de composants et énumérez les fonctions qu'ils réalisent.

3.2 Compteur à cycle quelconque

Attention sur les cartes *TP* logique dernière génération, les 74LS73 doivent être placée sur les emplacements où les V_{cc} et Gnd ne sont pas connectés. C'est à dire sur les deux supports en haut et à droite.

- Câblez le compteur sans la détection de combinaison interdite ;
- Établir un processus pour vérifier le fonctionnement du compteur sur toute la séquence.
- Câblez le compteur avec la détection de combinaison interdite ; Concernant la combinaison interdite, on peut minimiser l'emploi de portes, en utilisant un *Or* à la place d'un *et* et d'un *Not* : Expliquer ?

NB : On utilisera le générateur de signaux *Gbf* comme horloge.

3.3 Diviseur de fréquences

- Concevoir et réaliser un diviseur de fréquence par 2 et 4 à l'aide de la bascule 74LS74.
- Vérifier votre montage à l'aide d'un générateur de signaux.

3.4 Conclusion

Résumer les points importants du TP. Puis trouver les définitions sur internet et en faire un résumé des notions suivantes :

- Sortance
- Entrance
- Set up time

Annexe II : Datasheets

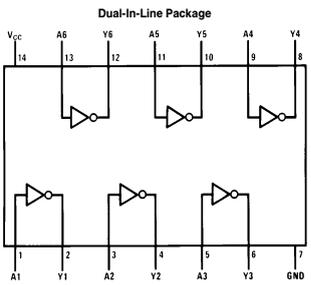

June 1989

54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

General Description
This device contains six independent gates each of which performs the logic INVERT function.

Features
■ Alternate Military/Aerospace device (54LS04) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6345-1

Order Number 54LS04DMQB, 54LS04FMQB, 54LS04MQB, DM54LS04J, DM54LS04W, DM74LS04M or DM74LS04N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = \bar{A}$

Input	Output
A	Y
L	H
H	L

H - High Logic Level
L - Low Logic Level

54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

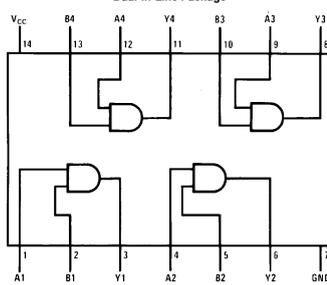

June 1989

54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

General Description
This device contains four independent gates each of which performs the logic AND function.

Features
■ Alternate Military/Aerospace device (54LS08) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6347-1

Order Number 54LS08DMQB, 54LS08FMQB, 54LS08MQB, DM54LS08J, DM54LS08W, DM74LS08M or DM74LS08N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H - High Logic Level
L - Low Logic Level

54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

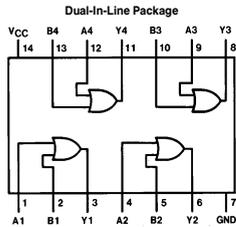
General Description

This device contains four independent gates each of which performs the logic OR function.

Features

- Alternate Military/Aerospace device (54LS32) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/6361-1

Order Number 54LS32DMQB, 54LS32FMQB, 54LS32LMQB, DM54LS32J, DM54LS32W, DM74LS32M or DM74LS32N See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = A + B$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = High Logic Level
L = Low Logic Level

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

DM74LS86 Quad 2-Input Exclusive-OR Gate

General Description

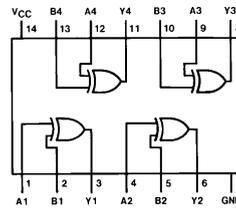
This device contains four independent gates each of which performs the logic exclusive-OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS86M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS86SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS86N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = A \oplus B = \bar{A}B + A\bar{B}$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

DM74LS86 Quad 2-Input Exclusive-OR Gate

**SN5473, SN54LS73A, SN7473, SN74LS73A
DUAL J-K FLIP-FLOPS WITH CLEAR**

SDLS118 - DECEMBER 1983 - REVISED MARCH 1988

- Package Options Include Plastic "Small Outline" Packages, Flat Packages, and Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

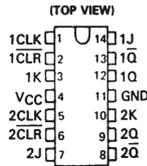
description

The '73, and 'H73, contain two independent J-K flip-flops with individual J-K, clock, and direct clear inputs. The '73, and 'H73, are positive pulse-triggered flip-flops. J-K input is loaded into the master while the clock is high and transferred to the slave on the high-to-low transition. For these devices the J and K inputs must be stable while the clock is high.

The 'LS73A contains two independent negative-edge-triggered flip-flops. The J and K inputs must be stable one setup time prior to the high-to-low clock transition for predictable operation. When the clear is low, it overrides the clock and data inputs forcing the Q output low and the \bar{Q} output high.

The SN5473, SN54H73, and the SN54LS73A are characterized for operation over the full military temperature range of -55°C to 125°C. The SN7473, and the SN74LS73A are characterized for operation from 0°C to 70°C.

SN5473, SN54LS73A . . . J OR W PACKAGE
SN7473 . . . N PACKAGE
SN74LS73A . . . D OR N PACKAGE



**'73
FUNCTION TABLE**

INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	\downarrow	L	L	Q ₀	\bar{Q} ₀
H	\downarrow	H	L	H	L
H	\downarrow	L	H	L	H
H	\downarrow	H	H	TOGGLE	

**'LS73A
FUNCTION TABLE**

INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	\downarrow	L	L	Q ₀	\bar{Q} ₀
H	\downarrow	H	L	H	L
H	\downarrow	L	H	L	H
H	\downarrow	H	H	TOGGLE	
H	H	X	X	Q ₀	\bar{Q} ₀

FOR CHIP CARRIER INFORMATION,
CONTACT THE FACTORY

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

**DM74LS74A
Dual Positive-Edge-Triggered D Flip-Flops with
Preset, Clear and Complementary Outputs**

General Description

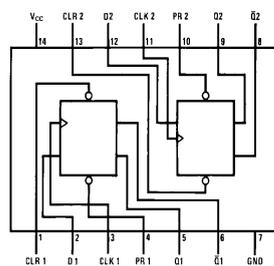
This device contains two independent positive-edge-triggered D flip-flops with complementary outputs. The information on the D input is accepted by the flip-flops on the positive going edge of the clock pulse. The triggering occurs at a voltage level and is not directly related to the transition time of the rising edge of the clock. The data on the D input may be changed while the clock is LOW or HIGH without affecting the outputs as long as the data setup and hold times are not violated. A low logic level on the preset or clear inputs will set or reset the outputs regardless of the logic levels of the other inputs.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS74AM	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS85ASJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS74AN	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

Inputs				Outputs	
PR	CLR	CLK	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H (Note 1)	H (Note 1)
H	H	\uparrow	H	H	L
H	H	\uparrow	L	L	H
H	H	L	X	Q ₀	\bar{Q} ₀

H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
L = LOW Logic Level
 \uparrow = Positive-going Transition
Q₀ = The output logic level of Q before the indicated input conditions were established.

Note 1: This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (HIGH) level.

Contrôle de Logique 2017

David Delfieu - Sans document ni calculatrice - 1H45

NOM :

PRENOM :

1 Arithmétique binaire (4 pts)

Soit des nombres exprimés sur n bits. Considérons deux nombres positifs A et B .

1. Développer l'expression de $A + CA_1(B)$ (0,5 pt)
2. Développer les expressions pour les cas $A > B$ et $A < B$ et déduire les propriétés du CA_1 . (2 pts)
3. En quelle base les équations suivantes ont-elles un sens ? (1,5 pts)
 - * $20=4*4$
 - * $200-60-50-20-2=5$

Solution

$$A + CA_1(B) = A + 2^n - B - 1$$

$$\text{SI } A > B \text{ Alors } R = A - B, \quad R > 0$$

$$1. \quad = R + 2^n - 1 = R - 1 : \text{Correction de } +1$$

$$\text{SI } A < B \text{ Alors } R = A - B, \quad R < 0$$

$$= 2^n - |R| - 1 = CA_1(|R|) : \text{Pas de correction}$$

$$2. \text{ Base } 8 : 2*8=4*4$$

$$2 * B^2 - 6B - 5B - 2B - 2 = 5$$

$$2 * B^2 - 13B - 7 = 0$$

$$3. \quad (2 * B + 1)(B - 7) = 0$$

$$B = 7$$

2 logique combinatoire (9,5 pts)

2.1 Etablir des théorèmes simplificateurs avec les expressions suivantes (2,5 pts)

$$T_1 : (X + a).(X + b)$$

$$T_2 : (X + \bar{a} + b).(X + a + \bar{b})$$

$$T_3 : (X + \bar{a}bc).(X + a\bar{b}c).(X + ab\bar{c})$$

2.2 Démonstrations contraintes (3 pts)

En utilisant au moins une fois les théorèmes précédents, simplifier :

1. $f_1(a, b, c, d, e, f, g) = (cf + a(b + c)e)(cf + b(\bar{a} + \bar{e})c)$
2. $f_2(a, b, c, d, e, f, g) = (cf + e + \bar{a}.\bar{b} + g)(cf + \bar{e}(a + b) + \bar{g})$
3. $f_3(a, b, c, d, e, f, g) = (a + \bar{e}.fg.h)(e(\bar{f} + \bar{g}).h + a)(\bar{h}.e.fg + a)$

Solutions

1. $T_1 : (X + a).(X + b) = X + ab$
2. $T_2 : (X + \bar{a} + b).(X + \bar{b}) = X + ab + \bar{a}\bar{b}$
3. $T_3 : (X + \bar{a}bc).(X + a\bar{b}c).(X + ab\bar{c}) = X.(X + ab\bar{c}) = X$
4. $f_1(a, b, c, d, e, f, g) \stackrel{T_1}{=} (cf + a(b + c)e).b(\bar{a} + \bar{e})c = cf$
5. $f_2(a, b, c, d, e, f, g) \stackrel{T_2}{=} (cf + (e + \bar{a}.\bar{b}) + \bar{g} + \bar{e}(a + b) + g)$
6. $f_3(a, b, c, d, e, f, g) \stackrel{T_3}{=} a$

2.3 Problème (4 pts)

Dans une usine de briques, on effectue un contrôle de qualité suivant quatre critères : Le Poids P , la Longueur L , la largeur l et la hauteur h . Si $P = 0$ (resp. $L = 0, l = 0, h = 0$) alors le Poids (resp. la Largeur, la longueur ou la hauteur) est incorrect(e). Cela permet de classer les briques en trois catégories de qualité décroissante :

A : Le poids P ainsi que au moins deux dimensions sont correctes ;

B : Le poids P est seul incorrect ou si le poids est correct au moins deux dimensions sont incorrectes ;

C : Le poids P est incorrect et au moins une autre dimension est incorrecte.

⇒ **Donner** les tables de karnaugh et les équations simplifiées de A, B et C .

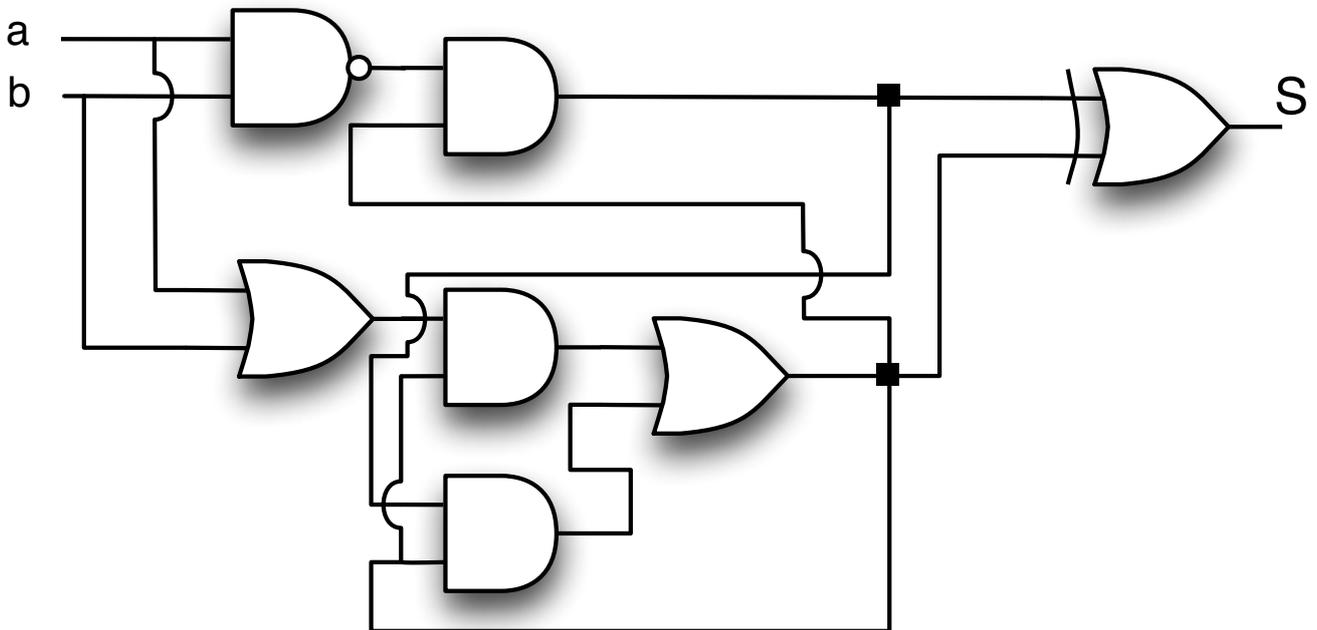
Solutions :

$$A = PLh + P\bar{L}l + P\bar{l}h$$

$$B = P\bar{L}h + P\bar{L}l + P\bar{l}h + \bar{P}.L.l.h$$

$$C = \bar{P}.\bar{L} + \bar{P}.\bar{l} + \bar{P}.\bar{h}$$

3 Logique séquentielle (6,5 pts)



- Identifier les rebouclages
- Ajouter sur le schéma les équations pour toutes les portes
- Analyser le système jusqu'à la machine à états.

Solution : Il y a 2 rebouclages et donc 2 variables internes Y_1 et Y_2 .

Equations

$$\begin{cases} y_1 = \bar{a}.Y_2 + \bar{b}.Y_2 \\ y_2 = a.Y_2 + b.Y_2 + Y_1.Y_2 \\ S = Y_0 \oplus Y_1 \end{cases}$$

Tables d'excitation et des états nommés

$$y_1(a,b,Y_1,Y_2)$$

		a				
		b				
Y ₁	Y ₂	00	01	11	10	
		00	0	0	0	0
		01	1	1	0	1
		11	1	1	0	1
	10	0	0	0	0	

$$y_2(a,b,Y_1,Y_2)$$

		a				
		b				
Y ₁	Y ₂	00	01	11	10	
		00	0	0	0	0
		01	0	1	1	1
		11	1	1	1	1
	10	0	0	0	0	

$$y_1y_2(a,b,Y_1,Y_2)$$

		a				
		b				
Y ₁	Y ₂	00	01	11	10	
		00	00	00	00	
		01	10	11	01	11
		11	11	11	01	11
	10	00	00	00	00	

$$y_1y_2(a,b,Y_1,Y_2)$$

		a				
		b				
Y ₁	Y ₂	00	01	11	10	
		00	00	00	00	
		01	10	11	01	11
		11	11	11	01	11
	10	00	00	00	00	

$$y_1y_2(a,b,Y_1,Y_2)$$

		a				
		b				
Y ₁	Y ₂	00	01	11	10	
		00	q ₀	q ₂	q ₄	q ₆
		01	q ₀	q ₃	q ₅	q ₇
		11	q ₁	q ₃	q ₅	q ₇
	10	q ₀	q ₂	q ₄	q ₆	

Machine de Moore : Les sorties S_1 et S_2 relative à chaque état sont représentés sous forme d'exposant dans l'état.

