



Circuits logiques

Tanmoy MONDAL

tanmoy.mondal@lirmm.fr

Diapos de D. Delahaye et Chouki TIBERMACHINE



POLYTECH[®]
MONTPELLIER

Objectifs du chapitre

1. Maîtriser les bases de l'algèbre booléenne ;
2. Faire la synthèse et l'analyse d'un circuit combinatoire ;
3. Connaître les circuits logiques les plus importants ;
4. Appréhender les principes des circuits séquentiels et des bascules.

Notion de circuit logique

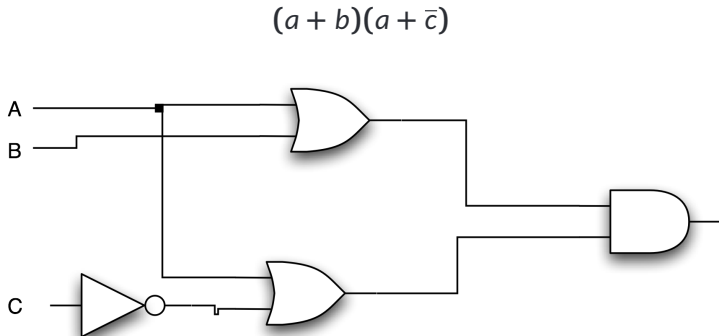
- Les circuits des machines électroniques modernes ont **2 états d'équilibre 0 et 1** (i.e., 2 niveaux de tension) \Rightarrow signal logique
- Une ligne permet de **transporter un signal logique**

Circuit logique

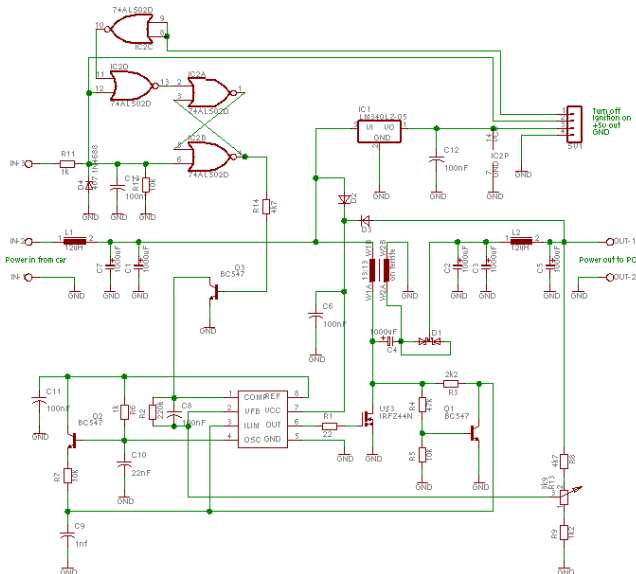
Représentation d'un circuit électronique. Exécute des opérations sur des variables logiques, transporte et traite des signaux logiques.

- **Circuit combinatoire**
 - circuit idéalisé
 - pas de prise en compte **du temps de propagation des signaux**
 - signaux de sortie dépendent que des signaux en entrée
- **Circuit séquentiel**
 - tiens compte du temps de propagation
 - mémoire
 - signaux de **sortie dépendent des signaux en entrée antérieurs**

Exemple de circuit logique (1/2)

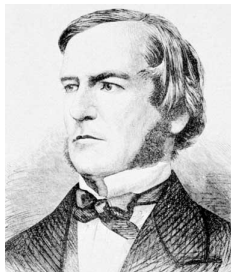


Exemple de circuit logique (2/2)



Algèbre de Boole

George Boole a défini (milieu du 19ème siècle) une algèbre qui s'applique à des fonctions logiques de variables logiques (variables booléennes).



- Toute fonction logique peut être réalisée à partir de fonctions logiques de base (opérateurs ou portes – *gates*).
- Les opérations arithmétiques peuvent être réalisées à l'aide d'opérations logiques de base.

Algèbre de Boole : définitions

Fonction logique

- Fonction définie par une **table de vérité** (i.e., tableau de **correspondance** entre les états d'entrée et les états de sortie) ;
- Toutes les combinaisons possibles des variables d'entrée ;
- Représentée avec des **diagrammes** ou **expressions algébriques** ;
- Trois opérateurs de base : NON, ET, OU.

Algèbre de Boole : définitions -suite-

Table de vérité

- La table de vérité d'une fonction de n variables a autant de lignes que d'états d'entrée, soit 2^n .
- Comme pour chacun de ces états d'entrée, on peut avoir deux valeurs de sortie (0 et 1)
- Cela nous donne 2^{2^n} fonctions possibles à n variables.

pour 1 variable, 4 fonctions

pour 2 variables, 16 fonctions

pour 3 variables, 256 fonctions

pour 4 variables, 65536 fonctions

Fonctions d'une variable

entrées	fonctions			
a	Z_0	Z_1	Z_2	Z_3
0	0	0	1	1
1	0	1	0	1

$Z_0 = 0$ constante

$Z_1 = a$ identité

$Z_2 = \bar{a}$ complémentation

$Z_3 = 1$ constante

Opérateur NON

La seule fonction logique à une variable non triviale est la fonction de complémentation (Z_2) réalisée par l'opérateur logique NON (ou inverseur) :
 $\bar{1} = 0$ ou $\bar{0} = 1$

NON - Table de vérité

entrées	NON
a	\bar{a}
0	1
1	0

Fonctions de 2 variables (1/2)

Il existe 16 fonctions logiques à 2 variables. Les deux non triviales les plus importantes sont les fonctions de produit logique (intersection) et somme logique (réunion) réalisées par les opérateurs ET et OU, notés respectivement ab et $a + b$.

ET - Table de vérité

entrées		ET
a	b	ab
0	0	0
0	1	0
1	0	0
1	1	1

OU - Table de vérité

entrées		OU
a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Fonctions de 2 variables (2/2)

XOR (ou exclusif \oplus)

entrées		XOR
a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

NAND (non et)

entrées		NAND
a	b	\overline{ab}
0	0	1
0	1	1
1	0	1
1	1	0

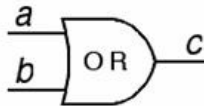
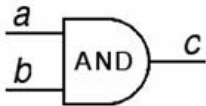
NOR (non ou)

entrées		NOR
a	b	$\overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

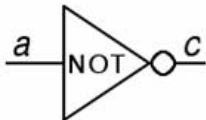
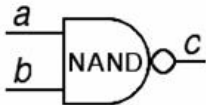
Opérateurs complets

- L'ensemble, utilisé en pratique, [ET, OU , NON] permet bien d'exprimer tous les opérateurs, mais il n'est pas minimal.
- On peut réaliser la fonction ET avec des OU et des NON et la fonction OU avec des ET et des NON.
- Il y a deux autres opérateurs importants du point de vue théorique dans l'algèbre de Boole : les opérateurs NAND (non et) et NOR (non ou).
- En effet, ces fonctions forment un ensemble complet ou minimal, c'est à dire qu'ils peuvent exprimer tous les opérateurs.

Symboles des principaux opérateurs logiques



...



Exercice

Circuit logique combinatoire

Dessiner le circuit logique combinatoire qui réalise la fonction :

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

Construire la table de vérité d'une fonction logique

Pour chaque combinaison de valeurs possibles des variables, on détermine la fonction booléenne.

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

a	b	c	$\bar{b}c$	$f(a, b, c)$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

Théorèmes fondamentaux de l'algèbre de Boole

(1/2)

Théorème des constantes	$a + 0 = a$	$a.0 = 0$
	$a + 1 = 1$	$a.1 = a$
Idempotence	$a + a = a$	$a.a = a$
Complémentation	$a + \bar{a} = 1$	$a.\bar{a} = 0$
Commutativité	$a + b = b + a$	$a.b = b.a$
Distributivité	$a + (bc) = (a + b)(a + c)$	
	$a(b + c) = (ab) + (ac)$	
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	
	$a(bc) = (ab)c = abc$	

Théorèmes fondamentaux de l'algèbre de Boole

(2/2)

Théorème de De Morgan

$$\overline{ab} = \bar{a} + \bar{b}$$

$$\overline{a + b} = \bar{a}\bar{b}$$

Autres relations

$$\overline{\bar{a}} = a$$

$$a + (ab) = a$$

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

$$a(a + b) = a$$

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

Méthodes des minterms et des maxterms

À l'aide des théorèmes précédents, il est possible d'exprimer toute fonction logique à l'aide des opérateurs NON, ET, OU.

Méthodes des minterms (somme logique des produits logiques)

La fonction peut être exprimée comme étant la somme logique des minterms correspondant à chaque sortie valant 1 dans la table de vérité. Chaque variable d'entrée est prise telle quelle si elle a la valeur 1, sinon elle est remplacée par son complément. Équivalent à la forme DNF en logique.

Méthodes des minterms et des maxterms -suite-

Méthodes des maxterms (produit logique des sommes logiques)

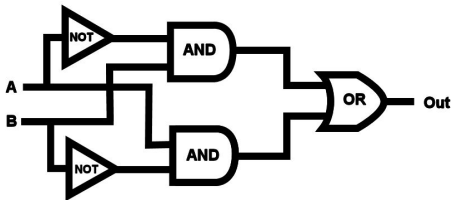
La fonction peut être exprimée comme étant le produit logique des maxterms correspondant à chaque sortie valant 0 dans la table de vérité. Chaque variable d'entrée est prise telle quelle si elle a la valeur 0, sinon elle est remplacée par son complément. Équivalent à la forme CNF en logique.

L'expression algébrique obtenu est dite forme **normale** (ou **canonique**).

Exemple : l'opérateur XOR

On veut exprimer la fonction XOR (ou exclusif) en n'utilisant que les fonctions ET, OU , NON :

- avec la méthode des minterms :
$$a \oplus b = \bar{a}b + a\bar{b}$$
- avec la méthode des maxterms :
$$a \oplus b = (a + b)(\bar{a} + \bar{b})$$



XOR - Table de vérité

entrées		XOR
a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Méthodes des minterms et des maxterms

Min and Max terms for two literal binary expressions

Input Variable (A)	Input Variable (B)	Minterm	Minterm notation	Maxterm	Maxterm notation
0	0	$\bar{A}\bar{B}$	m_0	$A+B$	M_0
0	1	$\bar{A}B$	m_1	$A+\bar{B}$	M_1
1	0	$A\bar{B}$	m_2	$\bar{A}+B$	M_2
1	1	AB	m_3	$\bar{A}+\bar{B}$	M_3

Min and Max terms for three literal binary expressions

Input Variable (A)	Input Variable (B)	Input Variable (C)	Minterm	Minterm notation	Maxterm	Maxterm notation
0	0	0	$\bar{A}\bar{B}\bar{C}$	m_0	$A+B+C$	M_0
0	0	1	$\bar{A}\bar{B}C$	m_1	$A+B+\bar{C}$	M_1
0	1	0	$\bar{A}B\bar{C}$	m_2	$A+\bar{B}+C$	M_2
0	1	1	$\bar{A}BC$	m_3	$A+\bar{B}+\bar{C}$	M_3
1	0	0	$A\bar{B}\bar{C}$	m_4	$\bar{A}+B+C$	M_4
1	0	1	$A\bar{B}C$	m_5	$\bar{A}+B+\bar{C}$	M_5
1	1	0	$AB\bar{C}$	m_6	$\bar{A}+\bar{B}+C$	M_6
1	1	1	ABC	m_7	$\bar{A}+\bar{B}+\bar{C}$	M_7

Exercice 1 : méthodes des minterms et des maxterms

Exemple

Appliquer les méthodes des minterms et des maxterms à la fonction :

Min and Max terms for three literal binary expressions

Input Variable (A)	Input Variable (B)	Input Variable (C)	Minterm	Minterm notation	Maxterm	Maxterm notation
0	0	0	$\bar{A}.\bar{B}.\bar{C}$	m_0	$A+B+C$	M_0
0	0	1	$\bar{A}.\bar{B}.C$	m_1	$A+B+\bar{C}$	M_1
0	1	0	$\bar{A}.B.\bar{C}$	m_2	$A+\bar{B}+C$	M_2
0	1	1	$\bar{A}.B.C$	m_3	$A+\bar{B}+\bar{C}$	M_3
1	0	0	$A.\bar{B}.\bar{C}$	m_4	$\bar{A}+B+C$	M_4
1	0	1	$A.\bar{B}.C$	m_5	$\bar{A}+B+\bar{C}$	M_5
1	1	0	$A.B.\bar{C}$	m_6	$\bar{A}+\bar{B}+C$	M_6
1	1	1	$A.B.C$	m_7	$\bar{A}+\bar{B}+\bar{C}$	M_7

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

Exercice 2 : méthodes des minterms et des maxterms

Exemple

Appliquer les méthodes des minterms et des maxterms à la fonction :

Min and Max terms for three literal binary expressions

Input Variable (A)	Input Variable (B)	Input Variable (C)	Minterm	Minterm notation	Maxterm	Maxterm notation
0	0	0	$\bar{A}.\bar{B}.\bar{C}$	m_0	$A+B+C$	M_0
0	0	1	$\bar{A}.\bar{B}.C$	m_1	$A+B+\bar{C}$	M_1
0	1	0	$\bar{A}.B.\bar{C}$	m_2	$A+\bar{B}+C$	M_2
0	1	1	$\bar{A}.B.C$	m_3	$A+\bar{B}+\bar{C}$	M_3
1	0	0	$A.\bar{B}.\bar{C}$	m_4	$\bar{A}+B+C$	M_4
1	0	1	$A.\bar{B}.C$	m_5	$\bar{A}+B+\bar{C}$	M_5
1	1	0	$A.B.\bar{C}$	m_6	$\bar{A}+\bar{B}+C$	M_6
1	1	1	$A.B.C$	m_7	$\bar{A}+\bar{B}+\bar{C}$	M_7

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

Simplification de fonction logique : méthode algébrique

On utilise les théorèmes de l'algèbre de Boole vu précédemment pour simplifier l'expression algébrique.

Exemple (minterms de l'exercice 2)

$$\begin{aligned}f(a, b, c) &= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}bc \\&= ab(c + \bar{c}) + a\bar{b}(c + \bar{c}) + \bar{a}bc \\&= ab + a\bar{b} + \bar{a}bc \\&= a(b + \bar{b}) + \bar{a}bc \\&= a + \bar{a}bc \\&= a + \bar{b}c\end{aligned}$$

Exercice : simplification (méthode algébrique)

Exemple (minterms de l'exercice 1)

Simplifier la fonction obtenue par la méthode des minterms à l'exercice 1.

Simplification de fonction logique : tables de Karnaugh

Basée sur l'inspection visuelle de tables judicieusement construites (\approx table de vérité à 2 dimensions).

- On attribue la valeur 1 aux cases correspondantes aux états d'entrée où la fonction est vraie, 0 sinon.
- Regroupement par blocs rectangulaires de 2, 4 ou 8 variables, des cases à 1 adjacentes.
 - Attention la table se referme sur elle-même.
 - Une case à 1 peut appartenir à plusieurs blocs.
 - Blocs les plus gros possibles (on utilise un bloc une seule fois).
- Pour chaque bloc :
 - Si une variable prend comme valeur 0 et 1, on ne la prend pas en compte.
 - On garde les variables dont la valeur ne varie pas.
 - Opérateur = ET.

Table de Karnaugh à 2 variables

Exemple

Table de vérité :

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

Expression algébrique
canonique (minterms) :

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

Table de Karnaugh :

$b \backslash a$	0	1
0	0	1
1	1	1

$$\Rightarrow f(a, b) = a + b$$

Table de Karnaugh à 3 variables

Exemple

Table de vérité :

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

Table de Karnaugh :

ab	00	01	11	10
c				
0	0	0	1	1
1	1	0	1	1

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

Expression (minterms) : $f(a, b, c) =$
 $\bar{a}\bar{b}c + ab\bar{c} + abc + a\bar{b}\bar{c} + \bar{a}bc$

Table de Karnaugh à 4 variables

Exemple

Expression algébrique
canonique (minterms) :

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

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

Table de Karnaugh :

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

$$\Rightarrow f(a, b, c, d) = \bar{a}\bar{b} + \bar{b}\bar{d} + \bar{b}\bar{c}d_{28/56}$$

Table de Karnaugh à 4 variables

Autre exemples

	<i>CD</i>	00	01	11	10
<i>AB</i>	00	1	1		
	01	1	1		
	11	1	1	1	1
	10	1	1		

$$\bar{C} + AB$$

	<i>CD</i>	00	01	11	10
<i>AB</i>	00	1		1	1
	01			1	
	11			1	
	10	1	1	1	1

$$A\bar{B} + CD + \bar{B}\bar{D}$$

Exercices

Donnez les fonctions simplifiées à partir des tables de Karnaugh suivantes :

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

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

Exercices : correction

Donnez les fonctions simplifiées à partir des tables de Karnaugh suivantes :

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

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

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

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

Règles de tables de Karnaugh

- Les groupes doivent contenir que 1 ou X
- Les groupes doivent contenir 1, 2, 4, 8, 16 (2^n) cellules
- Chaque cellule contenant un 1 doit appartenir à au moins un groupe
- Le groupe peut être horizontal ou vertical, mais pas en diagonale
- Les groupes doivent être aussi grands que possible
- Il devrait y avoir le moins de groupes possible
- Les groupes peuvent se chevaucher
- Les groupes peuvent s'asseoir autour de la table. La cellule la plus à gauche d'une rangée peut être groupée avec la cellule du bas

Synthèse d'un circuit combinatoire

Méthode de synthèse

À partir d'une fonction logique, déterminer un circuit logique réalisant cette fonction et obtenir le meilleur (i.e., le plus simple en nombre de portes, de connexions) :

1. Construire la table de vérité de la fonction logique ;
2. En dériver une expression algébrique (par exemple par la méthode des minterms) ;
3. Simplifier cette expression (méthode algébrique ou tables de Karnaugh) ;
4. Réaliser la fonction logique à l'aide d'opérateurs divers (NON, ET, OU, XOR, NAND, NOR, etc.) pour obtenir un logigramme.

Exercice

Synthèse

Déduire le circuit de chacune des fonctions suivantes :

1. $xyz + x\bar{y} + \bar{x}y$;
2. $(x + z)(y + z) + y$.

Analyse d'un circuit combinatoire

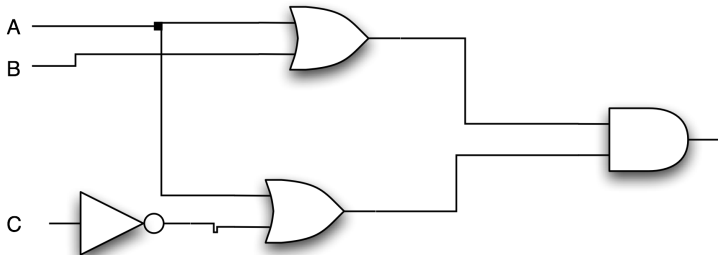
L'analyse est l'opération inverse de la synthèse.

Méthode de d'analyse

Retrouver la fonction d'un circuit dont on connaît uniquement le logigramme :

1. En procédant des entrées vers les sorties, donner, pour chaque opérateur l'expression de sa sortie en fonction de ses entrées, jusqu'à obtention d'une expression pour chaque fonction réalisée par le circuit;
2. Donner la table de vérité correspondante;
3. En déduire le rôle du circuit.

Exemple d'analyse

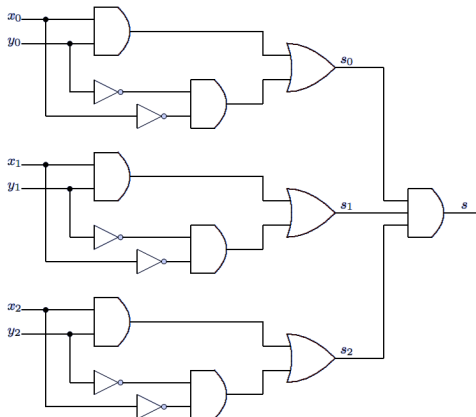


$$\Rightarrow f(a, b, c) = (a + b)(a + \bar{c})$$

Exercice

Analyse

Donner la fonction logique correspondant au circuit ci-dessous.
Simplifier la fonction et re-dessiner le circuit obtenu.



Circuits logiques les plus importants

- Demi-additionneur (addition sans gestion de la retenue) et additionneur complet (addition avec gestion de la retenue);
- Multiplexeur (plusieurs signaux en entrées, 1 seule sortie) et démultiplexeur (un seul signal en entrée et plusieurs sorties);
- Décodeur, codeur et transcodeur (e.g., conversion de base).

Synthèse d'un demi-additionneur

Circuit logique capable de faire la somme de 2 nombres binaires mais qui ne tient pas compte de la retenue éventuelle provenant d'une opération précédente.

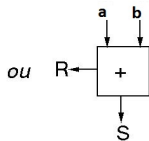
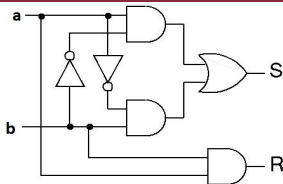
Demi-additionneur

<i>a</i>	<i>b</i>	Sortie <i>S</i>	Retenue <i>R</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

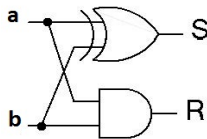
Expression algébrique
canonique (minterms) :

$$S = a\bar{b} + \bar{a}b = a \oplus b$$

$$R = ab$$



ou plus simple :



Synthèse d'un étage d'additionneur (1/2)

Circuit logique capable de faire la somme de 2 nombres binaires et d'une retenue provenant d'une opération précédente.

Étage d'additionneur

a	b	R_0	S	R_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

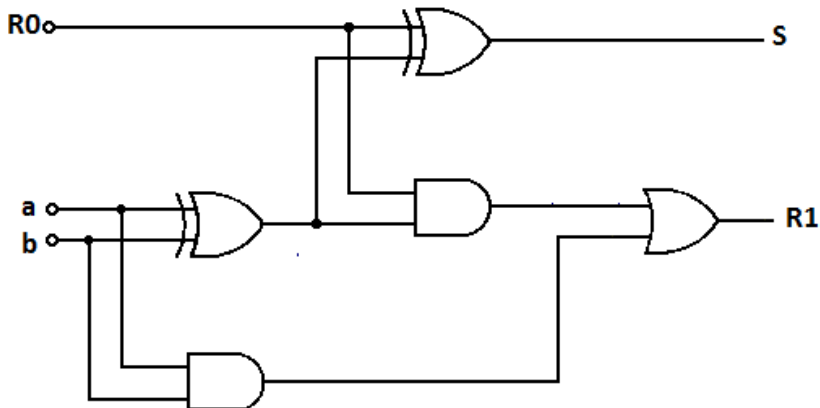
Expression algébrique canonique (minterms) :

$$\begin{aligned} S &= \bar{a}\bar{b}R_0 + \bar{a}b\bar{R}_0 + a\bar{b}\bar{R}_0 + abR_0 \\ &= R_0(\bar{a}\bar{b} + ab) + \bar{R}_0(\bar{a}b + a\bar{b}) \\ &= R_0(\overline{a \oplus b}) + \bar{R}_0(a \oplus b) \\ &= R_0 \oplus (a \oplus b) \end{aligned}$$

$$\begin{aligned} R_1 &= \bar{a}bR_0 + a\bar{b}R_0 + ab\bar{R}_0 + abR_0 \\ &= R_0(\bar{a}b + a\bar{b}) + ab(R_0 + \bar{R}_0) \\ &= R_0(a \oplus b) + ab \end{aligned}$$

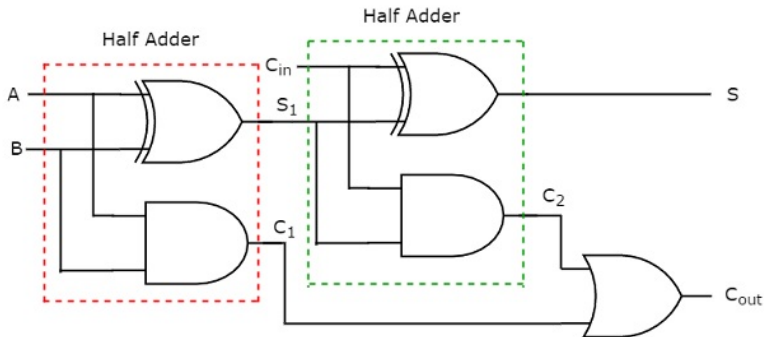
Synthèse d'un étage d'additionneur (2/2)

Logigramme correspondant :



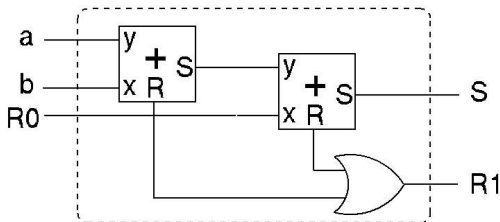
Synthèse d'un étage d'additionneur (2/2)

Logigramme correspondant :

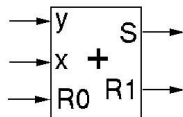


Additionneur binaire complet

- L'**étage d'additionneur** est composé de 2 demi-additionneurs et d'un OU. Il fait la somme de 2 bits en tenant compte d'une éventuelle retenue.



ou



- L'**additionneur complet** est obtenu en utilisant en parallèle plusieurs étages additionneurs (il faut autant d'étages que de bits composants les nombre binaires à additionner).

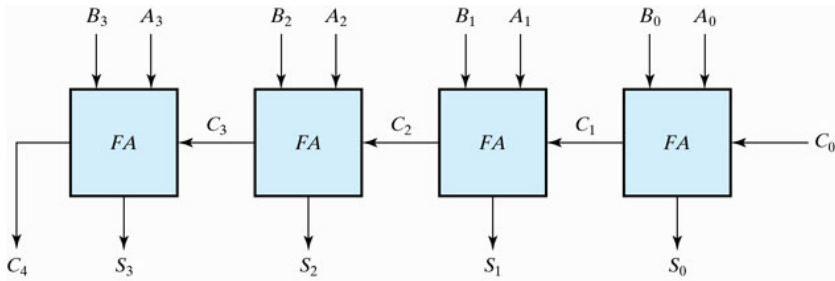
Exercice

Additionneur complet sur 4 bits

- Dessiner le schéma d'un additionneur de deux nombres sur 4 bits.

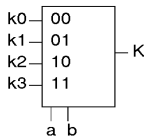
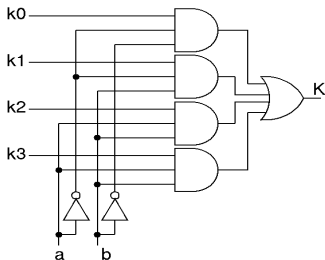
Additionneur complet sur 4 bits

- Dessiner le schéma d'un additionneur de deux nombres sur 4 bits.
- Les nombres binaires 4 bits : $A = A_3, A_2, A_1, A_0$ and $B = B_3, B_2, B_1, B_0$
- Utilisez quatre additionneurs complets, puisque le "carry" initial C_{in} est égal à zéro.



Multiplexeur

- 2^n entrées, n variables, 1 sortie
- La sortie (K) prend la valeur d'une des entrées selon la valeur des n variables :
une des 4 entrées est acheminé sur la sortie K.
 - Basé sur la combinaison des entrées présentes sur ces deux lignes de sélection



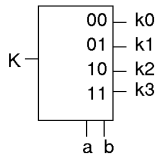
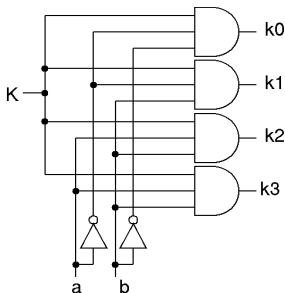
Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From Truth table, we can directly write the **Boolean function** for output, Y as

$$Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$

Démultiplexeur

- 1 entrée, n variables, 2^n sorties
- Une des sorties prend la valeur de l'entrée (K) selon la valeur des n variables (deux lignes de sélection) : **la variable K est acheminé sur l'une des 4 sorties.**



$$Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I$$

$$Y_1 = s_1' s_0 I$$

$$Y_0 = s_1' s_0' I$$

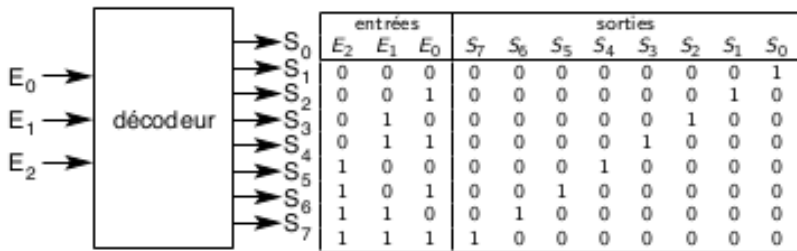
Selection Inputs		Outputs			
s_1	s_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Applications de multiplexeurs

- Fonction universelle (i.e., un multiplexeur à n variables peut réaliser les 2^{2^n} fonctions logiques à n variables ;
- Multiplexage (i.e., concentrer plusieurs lignes en une seule ou faire l'opération inverse) ;
- Codage, décodage, transcodage.

Décodeur (1/2)

Fait correspondre à un code en entrée (sur n lignes) une seule sortie active (i.e., a 1) parmi les 2^n sorties possibles.

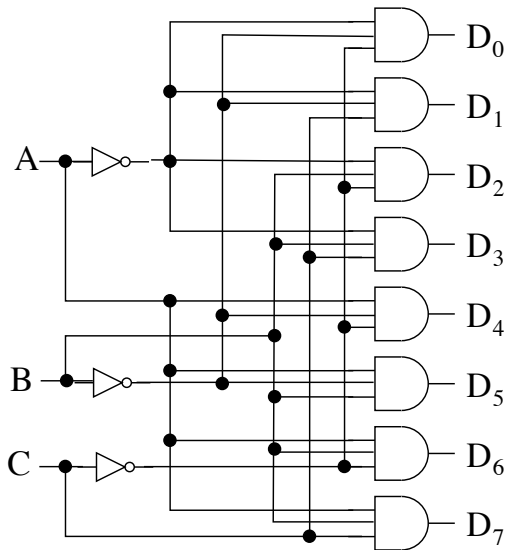


un décodeur 3 bits

la table de vérité.

Le décodeur peut être utilisé pour convertir un nombre binaire en nombre décimal ou pour adresser une mémoire.

Décodeur (2/2)



Exercice

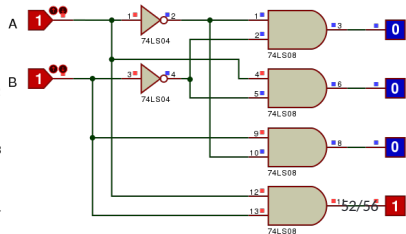
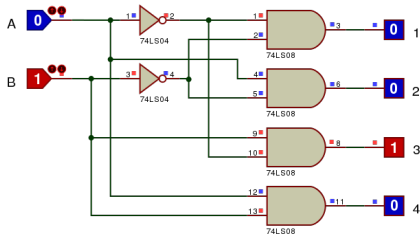
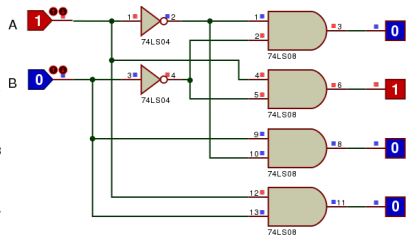
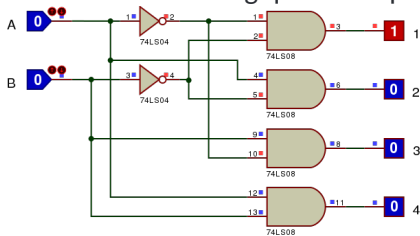
Décodeur 2 vers 4

- Construire la table de vérité d'un décodeur 2 vers 4 ;
- Dessiner le circuit logique correspondant.

Exercice : correction

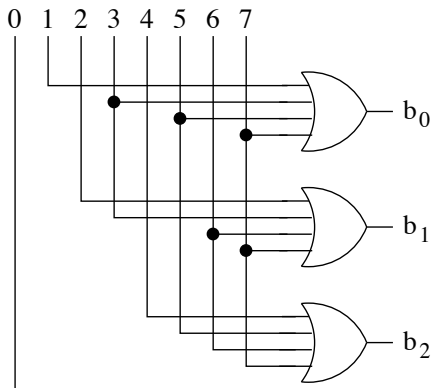
Décodeur 2 vers 4

- Dessiner le circuit logique correspondant.



Codeur

Fait correspondre à une entrée active, parmi les 2^n entrées, un code sur n lignes en sortie.



Un **transcodeur** fait correspondre une entrée sur n lignes correspondant à un certain codage, une sortie sur m lignes

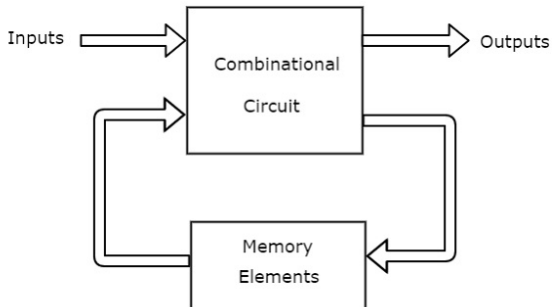
Circuits séquentiels

Circuits combinatoires \Rightarrow pas de rétroactions (i.e., de retours des sorties dans les entrées).

Les **circuits séquentiels** possèdent des rétroactions : les signaux de sortie ne dépendent pas uniquement des entrées mais aussi de leur séquence. Le circuit se rappelle des entrées et des états précédents : il a une **mémoire** du passé.

- Ajout des notions d'états et de mémoire.
- Ajout de la notion de temps (i.e., horloge).
- Repose sur la théorie des automates finis.

Circuits séquentiels



Combinational Circuits	Sequential Circuits
Outputs depend only on present inputs.	Outputs depend on both present inputs and present state.
Feedback path is not present.	Feedback path is present.
Memory elements are not required.	Memory elements are required.
Clock signal is not required.	Clock signal is required.
Easy to design.	Difficult to design.

Diapos et références

Diapos constuites sur la base du cours de :

David Delahaye, professeur à la FDS (mon prédécesseur)

Chouki TIBERMACINE, MCF à PolyTech-Montpellier (mon prédécesseur)

Références bibliographiques

- Paolo Zanella, Yves Ligier et Emmanuel Lazard. Architecture et technologie des ordinateurs - 6e éd. - Cours et exercices corrigés. Septembre 2018