Systèmes combinatoires 18

18.1 Numération, codage, transcodage

Remarque

On considère acquis les notions de systèmes de numérations décimaux, binaires et hexadécimaux ainsi que le passage d'une base à l'autre.
On considère aussi qu'il existe des notions sur la norme ASCII.

18.1.1 Systèmes de codages utiles en technologie

Code binaire naturel

Ce code pondéré correspond à donner un nombre selon sa valeur en système de numérotation binaire.

C'est le seul code qui permet de réaliser des opérations.

Inconvénient : il introduit des erreurs lors d'un changement de code. Pour passer de $(01)_2$ à $(10)_2$, donc de 1 à2, il faut modifier chaque digit et afficher (même brièvement) $(11)_2$ (soit 3) ou $(00)_2$ (soit 0). Cela peut provoquer des perturbations ou des erreurs.

Entrée				Sortie
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

TABLE 18.1 – Table de vérité du code binaire naturel

Code décimal codé binaire (DCB)

C'est un code pondéré base sur le code binaire naturel mais qui est adapté à la représentation des nombres en base 10. En effet le code binaire pur n'associe pas des bits spécifiques aux unités, dizaines, centaines, ... La propriété du code DCB est d'associer 4 bits différents à chaque puissance de 10. Ainsi,

$$(1664)_{10} \quad \Longrightarrow \quad (0001.0110.0110.0100)_{DCB}$$

Ce code est utilisé pour les afficheurs 7 segments. Chaque afficheur reçoit le chiffre codé en binaire sur 4 bits.

Inconvénient : Cette représentation adaptée à la représentation binaire des nombres décimaux utilise un nombre de bits supérieur à celui du binaire naturel, et donc une place plus important en mémoire de l'ordinateur.

Remarque

On peut réaliser la même typologie pour le code hexadécimal. Notons ce codage HCB. Ainsi,

$$(0C3F)_{16} \quad \Longrightarrow \quad (0000.1100.0011.1111)_{HCB}$$

Code binaire réfléchi ou code Gray

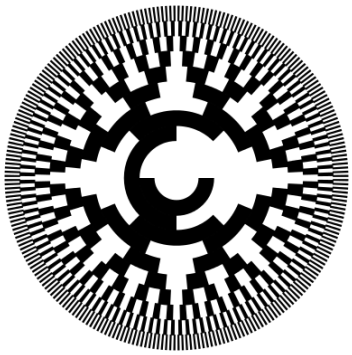


FIGURE 18.1 – Disque codage Gray

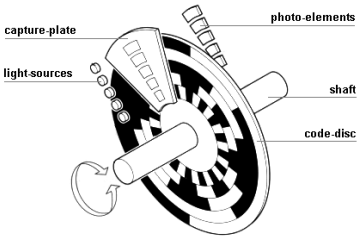


FIGURE 18.2 – Disque codage Gray

Ce code non pondéré est un arrangement du système binaire. Le passage d'un nombre à l'autre se fait en changeant l'état d'un seul bit. Il est très utilisé pour décrire des automatismes (un changement d'état d'un composant correspond à un bit qui change), en particulier dans les codeurs de position absolue.

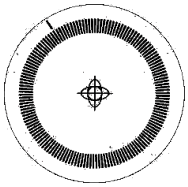
Avantage : il apporte une garantie d'interprétation avec une erreur maximale d'incrémentation.

Pour obtenir le code Gray, il faut faire toutes les 2^1, 2^2, 2^3 ... lignes, une symétrie en commençant par le bit de droite et changer la valeur du bit de gauche.

Décimal	Binaire pur				Binaire réfléchi			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

18.1.2 Les codeurs optiques

Le codeur incrémental



Un codeur incrémental est constitué d'un disque généralement muni de deux pistes ainsi que de trois couples de DEL – récepteurs. Une des deux pistes est percée d'une seule fente. Le détecteur noté Z permet de détecter le passage de cette fente. La seconde piste est composée de n fentes. Les deux autres DEL (A et B) détectent le passage des fentes sur cette piste.

Dans certains cas, les impulsions peuvent être mesurées par des capteurs à effets Hall.

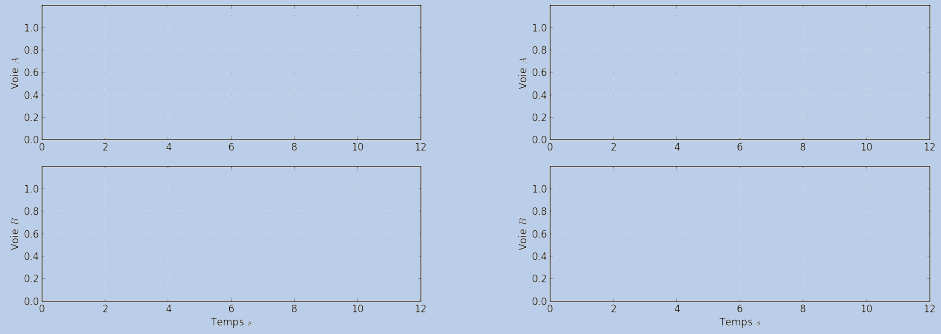
Ces codeurs permettent de réaliser une mesure incrémentale. Il faut donc fixer une référence (POM – Prise d'Origine Machine) pour connaitre la position absolue.

Ces codeurs sont utilisés pour mesurer les déplacements des axes sur les machines outils. On en retrouve aussi sur l'axe asservi Emericc ou encore sur le Tribar.

Exemple –

La résolution d'un codeur incrémental peut varier de 1 000 à 25 000 impulsions par tour pour des fréquences de rotation allant jusqu'à 10 000 tr/min.

1. Donner la résolution en degrés du codeur.
2. Quelle doit être la fréquence minimale du système d'acquisition afin de pouvoir gérer les informations provenant du codeur ?
3. Réaliser le chronogramme des sorties des voies A et B en faisant l'hypothèse que les DEL A et B sont «décalées» d'un quart de période.
4. Expliquer en quoi l'existence des deux LED permet de détecter le sens de rotation du disque.

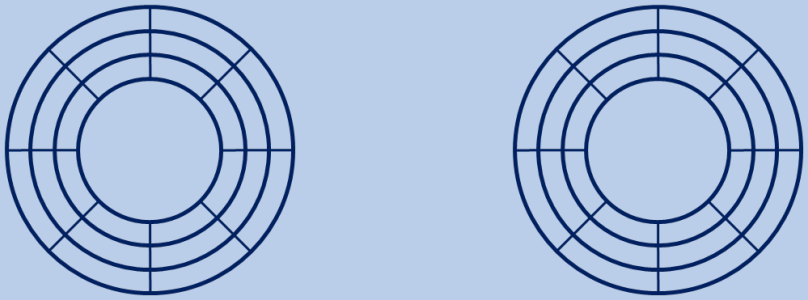


Le codeur absolu

Un codeur absolu permet de déterminer la position angulaire réel d'un système. Il est constitué de N pistes avec des fentes qui sont agencées selon le codage Gray. Une «rampe» de N couples de DEL – récepteurs permettent de détecter, à un instant donné une combinaison de bits correspondant à la position du disque.

Exemple –

1. Combien de pistes faut-il pour coder 360 avec une précision de $0,1$.
2. Combien faudrait-il de fentes avec codeur incrémental ?
3. Pour un codeur avec 3 pistes, teinter les différentes zones dans le cas d'un codage Gray puis dans le cas d'un codeur naturel.
4. Conclure sur l'intérêt du code Gray lorsque les DEL sont décalées.



Codage binaire naturel *Codage binaire réfléchi*

18.2 Systèmes combinatoires

18.2.1 Définitions

Variables binaires

De nombreux composants utilisés en automatisme ne peuvent normalement prendre que deux états différents : lampe allumée ou éteinte, bouton-poussoir actionné ou

relâché, moteur tournant ou à l'arrêt, vérin pneumatique sorti ou rentré.

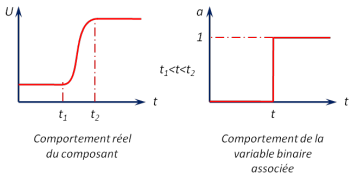
À chacun de ces composants, on peut associer une variable binaire (ou logique, Tout Ou Rien) qui ne peut prendre que deux valeurs notées 0 ou 1 (vrai ou faux, oui ou non).

Définition – Variable binaire

Une variable a est binaire si et seulement si elle peut prendre, à chaque instant, qu'une seule valeur parmi un ensemble de 2 valeurs possibles.

Exemple –

Lister les variables binaires pour un codeur incrémental avec une résolution de 0,1 degrés et pour un codeur absolu de même résolution.



Remarque

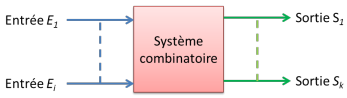
Le comportement tout ou rien (TOR) ne correspond qu'au comportement normalement prévu en régime stabilisé et en l'absence de tout dysfonctionnement. L'association d'une variable binaire à un composant ne peut pas rendre compte des états transitoires apparaissent entre deux états stables. C'est donc une simplification du comportement réel.

Système binaire

Définition –

Un système est dit binaire ou logique si les variables d'entrée et de sortie sont binaires.

Systèmes combinatoire et séquentiel



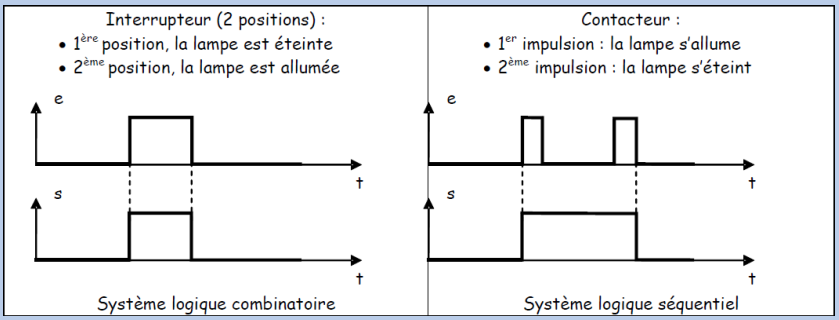
Définition –

Un système logique combinatoire est un système binaire pour lequel à un état des variables d'entrée E_i correspond un unique état des variables de sortie S_j. (La réciproque n'est pas vraie.)

Définition –

Un système logique est dit séquentiel si les sorties S_j ne dépendent pas uniquement des E_i.

Exemple –



18.2.2 Algèbre de Boole

La fonction logique qui caractérise le système est indépendante du temps. Pour traiter de tels systèmes, on utilise l'algèbre de Boole.

Définition

C'est un algèbre de propositions logiques mise au point par un mathématicien anglais, Georges Boole (1815 – 1864).

Définition –

Un ensemble E a une structure d'algèbre Boole si on a défini dans cet ensemble :

- une relation d'équivalence notée $=$;
- deux lois de composition interne $+$ (addition booléenne) et $\cdot$ (multiplication booléenne) ;
- une loi appelée complémentation ($\overline{a}$ complément de a).

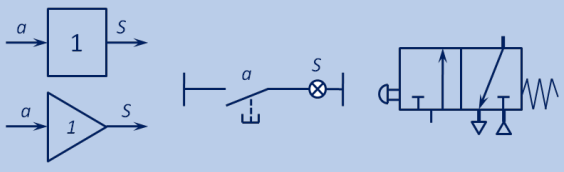
Une algèbre binaire est une algèbre de Boole dont les éléments B ne peuvent prendre que deux valeurs notées 0 ou 1 : $B = \{0, 1\}$.

Exemple –

Fonction OUI.

a	S
0	0
1	1

$S = a$



Fonction NON – appelée complément

Définition –

Fonction NON – appelée complément

$$B \longmapsto B$$
$$a \longmapsto \overline{a}$$

a	$\overline{a}$
0	1
1	0

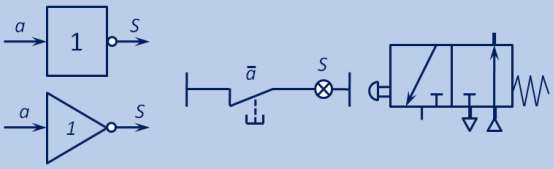
Il faut lire $\overline{a}$: NON a.

Exemple –

Fonction NON.

a	S
0	1
1	0

$S=\bar{a}$



Fonction ET – Produit booléen

Définition –

Fonction ET – Produit booléen

$$\begin{array}{rcl} B \times B & \longmapsto & B \\ (a,b) & \longmapsto & a \cdot b \end{array}$$

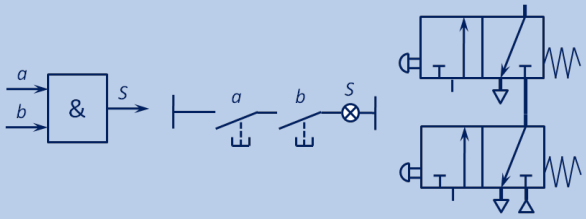
Il faut lire a ET b.

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Exemple –

a	b	S
0	0	0
0	1	0
1	1	1
1	0	0

$S=a.b$



Fonction OU – Somme booléenne

Définition –

Fonction OU – Somme booléenne

$$\begin{array}{rcl} B \times B & \longmapsto & B \\ (a,b) & \longmapsto & a + b \end{array}$$

Il faut lire a OU b.

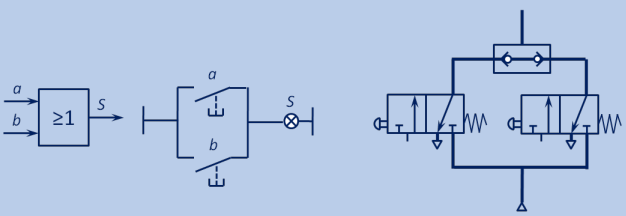
a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Exemple –

Fonction OU.

a	b	S
0	0	0
0	1	1
1	1	1
1	0	1

$S=a+b$



Propriétés des opérateurs de base de l'algèbre de Boole

Propriété –

- **Commutativité :** $a+b=b+a \quad a\cdot b=b\cdot a$.
- **Distributivité :** $a\cdot(b+c)=a\cdot b+a\cdot c \quad a+(b\cdot c)=(a+b)\cdot(a+c)$.
- **Associativité :** $a+(b+c)=(a+b)+c=a+b+c \quad a\cdot(b\cdot c)=(a\cdot b)\cdot c=a\cdot b\cdot c$.
- **Élément neutre :** $a+0=a \quad a\cdot 1=a$.
- **Élément absorbant :** $a+1=1 \quad a\cdot 0=0$.
- **Complémentarité :** $a+\overline{a}=1 \quad a\cdot\overline{a}=0$. Par ailleurs $\overline{\overline{a}}=a$.
- **Idem potence :** $a+a=a \quad a\cdot a=a$.
- **Identités remarquables :**
  - absorption : $a+a\cdot b=a$;
  - inclusion : $a+\overline{a}\cdot b=a+b$.

Théorème de Morgan

Theorem 18.2.1

$$\overline{\sum_{i=1}^{n} a_i}=\prod_{i=1}^{n}\overline{a_i} \qquad \overline{\prod_{i=1}^{n} a_i}=\sum_{i=1}^{n}\overline{a_i}$$

Exemple –

Déterminer les expressions complémentaires $\overline{P}$ et $\overline{Q}$ des expressions suivantes : $P=x\cdot y\cdot(z+\overline{t})$ et $Q=\overline{x}\cdot\overline{y}+y+x\cdot t$.

18.3 Fonctions logiques

Un système combinatoire est décrit par une fonction logique qui permet de définir de manière unique la sortie pour une combinaison des entrées.

18.3.1 Table de vérité

La table de vérité d'une fonction logique est une écriture systématique qui consiste à décrire toutes les combinaison des variables et à y associer les valeurs correspondantes de la fonction.

Exemple –

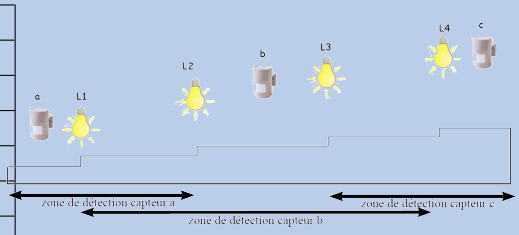
Compléter la table de vérité suivante et vérifier les théorèmes de De Morgan pour deux variables.

a	b	$\overline{a\cdot b}$	$\overline{a}+\overline{b}$	$\overline{a+b}$	$\overline{a}\cdot\overline{b}$
0	0				
0	1				
1	1				
1	1				

Exemple –

Soit 4 lampes L_1, L_2, L_3, L_4 permettant d'éclairer un escalier. L'allumage de ces 3 lampes est régit par 3 détecteurs de présences a, b et c. La table de vérité est la suivante :

a	b	c	L_1	L_2	L_3	L_4
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	1	0	0	1	1
0	1	0	0	1	1	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1
1	0	1	1	0	0	1
1	0	0	1	0	0	0



Donner l'expression de L_1, L_2, L_3, L_4.

Il est possible à l'aide de la table de vérité de définir une équation donnant les variables de sortie en fonction des variables d'entrée. Il existe 2 modes d'écriture particuliers :

- somme canonique (somme de produit) : $S = a \cdot b + c \cdot d$;
- produit canonique (produit de sommes) : $S = (a + b) \cdot (c + d)$.

Méthode – Détermination de la somme canonique

Pour chaque ligne où la sortie vaut 1, déterminer la combinaison d'entrées correspondante à l'aide de l'opérateur ET puis sommer ces combinaisons.

Exemple –

Donner l'expression sous forme canonique de L_1 et L_2 à partir de la table de vérité précédente.

Remarque

On utilise la forme de sommes canoniques si le nombre de 1 est inférieur au nombre de 0. Il est souvent nécessaire de simplifier ces équations pour réaliser technologiquement ces fonctions (voir plus loin).

Méthode – Détermination du produit canonique

Déterminer l'expression de $\overline{S}$ ce qui revient à : Déterminer la combinaison d'entrées pour chaque ligne où la sortie vaut 0, les sommer, puis passez au complémentaire en utilisant les théorèmes de De Morgan.

Exemple –

Donner l'expression sous forme de produit canonique de L_1 et L_2.

Remarque

Cette méthode est utile lorsque la sortie comporte peu de 0 et beaucoup de 1 ou lorsque l'on recherche la fonction complémentaire.

18.4 Simplification des fonctions logiques

La simplification des expressions logiques est destinée à économiser le matériel nécessaire à la réalisation (utilisation d'un composant réalisant plusieurs fonctions identiques) ou diminuer l'importance des équations programmées.

18.4.1 Cellules universelles

Définition – Cellule universelle

Une cellule (ou fonction) est dite « universelle » si elle permet de réaliser les fonctions ET, OU, NON. Il est alors possible de réaliser toutes les fonctions logiques à l'aide de cette seule cellule.

Exemple –

Les cellules NAND (non et – $aNANDb = \overline{a \cdot b}$) et NOR (non ou – $aNORb = \overline{a + b}$) sont des cellules universelles.
Réalisation de la fonction NON : $NON(a) = \overline{a} = \overline{a \cdot a}$. Besoin d'une seule cellule NAND.
Réalisation de la fonction ET : $aETb = a \cdot b = \overline{\overline{a \cdot b}}$. Besoin de deux cellules NAND en cascade.
Réalisation de la fonction OU : $aOUb = a + b = \overline{\overline{a + b}} = \overline{\overline{a} \cdot \overline{b}}$. Besoin de trois cellules NAND en cascade.
De même, la cellule NOR est universelle.

Exemple –

Écrire l'expression logique suivante uniquement avec des opérateurs NAND puis indiquer le nombre minimal d'opérateurs utilisés.

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

18.4.2 Simplification algébrique d'une fonction

Il s'agit d'utiliser les propriétés, théorèmes et identités remarquables de l'algèbre de Boole afin de simplifier une équation. Cette méthode n'est pas forcément simple à utiliser et demande beaucoup d'intuition.

Exemple –

$F(a,b,c) = \overline{a} \cdot \overline{b} \cdot c + a \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot c = \overline{a} \cdot c \cdot (\overline{b} + b) + a \cdot \overline{b} \cdot c = \overline{a} \cdot c + a \cdot \overline{b} \cdot c = c(\overline{a} + a \cdot \overline{b})$
$= c\left(\overline{a}(1 + \overline{b}) + a \cdot \overline{b}\right) = c \cdot (\overline{a} + \overline{b})$

Exemple –

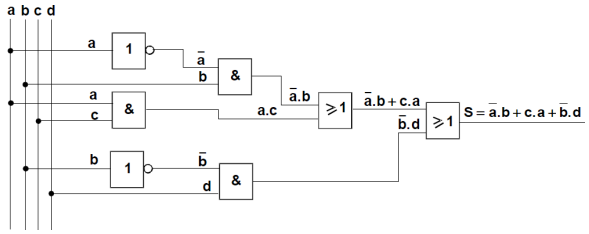
Simplifier par cette méthode l'expression des lampes L_1 et L_2.

18.5 Représentation des fonctions logiques

Dans un système combinatoire :

- les variables d'entrée sont des informations qui décrivent à un moment donnée l'état de certaines parties (position d'un chariot, d'un outil, ...)
- les variables de sortie permettent de décrire l'état dans lequel doivent se retrouver les organes récepteurs.

Une fois l'expression logique obtenue et simplifiée, il est possible d'exprimer graphiquement le comportement du système combinatoire. On utilise pour cela en général le logigramme ou le schéma électrique. On peut également utiliser un schéma pneumatique ou électronique.



18.5.1 Logigrammes pour un codeur incrémental

Exemple –

1. Réaliser le logigramme illustrant le comptage et le décomptage des impulsions provenant d'un codeur incrémental.
2. Réaliser le logigramme illustrant le comptage et le décomptage des impulsions provenant d'un codeur incrémental en multipliant par 4 sa résolution.

Remarque

On appelera front montant de a et notera $\uparrow a$ le passage de la variable a de l'état 0 à l'état 1.

Bibliographie

[1] Supports de cours de David Violeau, Lycée Saint-Louis, Paris.
[2] Supports de cours de Florestan Mathurin, Lycée Bellevue, Toulouse.