

R. MERAT
R. MOREAU
L. ALLAY
J.-P. DUBOS
J. LAFARGUE
R. LEGOFF



Électronique numérique

NATHAN

R. MÉRAT
R. MOREAU
L. ALLAY
J.-P. DUBOS
J. LAFARGUE
R. LEGOFF

BEP - BAC PRO

Électronique numérique

NATHAN

Sommaire

1. Systèmes de numération	4
2. Système binaire. Opérations	8
3. Codes	12
4. Opérations logiques	16
5. Théorèmes fondamentaux	20
6. Formes canoniques	24
7. Table de Karnaugh	28
8. Circuits numériques	32
9. Famille logique TTL	36
10. Famille logique CMOS	40
11. Bascule RS	44
12. Bascules synchrones	48
13. Trois bascules synchrones	52
14. Bascules. Initialisation. Synchronisation	56
15. Monostable	60
16. Horloge intégrée	64
17. Compteurs asynchrones	68
18. Compteurs synchrones	72
19. Registre à décalage	76
20. Additionneurs et soustracteurs	80
21. Multiplieurs binaires	84
22. Décodeurs	88
23. Codeurs	92
24. Multiplexeurs	96
25. Démultiplexeurs	100
26. Conversions	104
27. Convertisseur numérique-analogique	108
28. Convertisseur analogique-numérique	112
29. Traitement des signaux	116
30. Voltmètre numérique	120
31. Multimètre numérique	124

Ce livre s'adresse aux utilisateurs de matériels électriques, techniciens ou non, qui ont besoin de disposer rapidement d'informations théoriques ou pratiques. Il est également destiné à tous ceux qui souhaitent se familiariser avec l'électricité, lecteurs occasionnels ou autodidactes que la lecture d'un manuel scolaire ne motive pas toujours suffisamment. Parce qu'il regroupe les résultats essentiels de l'électricité élémentaire cet ouvrage peut aussi constituer un aide-mémoire très utile aux étudiants et aux auditeurs de la formation permanente.

La présentation retenue doit permettre un accès facile à l'information recherchée et un apprentissage aisé. En effet, à chaque notion importante correspond un chapitre. Sur quatre pages, le lecteur trouve une loi ou le modèle d'un composant ou une méthode d'étude d'un circuit accompagnée d'exemples d'applications et d'exercices avec leurs solutions. Ainsi, le technicien ou l'étudiant, qui désire retrouver un résultat ou une explication, est immédiatement dirigé par la table des matières vers le sujet qui l'intéresse. Et celui qui désire apprendre l'électricité voit son travail grandement facilité : les difficultés de l'étude sont réparties entre des chapitres courts qui constituent une suite d'étapes simples. Après chacune d'elles les connaissances peuvent être contrôlées et confortées avant de passer à la suivante. Cette démarche qui apporte simultanément progressivité et assurance en soi sera certainement appréciée de tous ceux que l'étude des sciences intéresse.

Le contenu de chaque chapitre reste toujours simple, les auteurs ayant choisi de privilégier l'observation et l'exploitation d'expériences plutôt que l'exposé de théories et le recours aux mathématiques. Pour un utilisateur, une bonne compréhension des applications d'une loi est souvent plus importante que les éventuelles équations différentielles qui peuvent lui correspondre. En effet, ces connaissances peuvent suffire pour acquérir une bonne maîtrise de nombreux équipements électriques, ce qui est l'objectif de tout utilisateur, la complexité des systèmes techniques excluant de plus en plus la possibilité d'acquérir les capacités nécessaires pour le dépannage et même parfois la maintenance.

Les auteurs

1 Base d'un système de numération

En électronique numérique, les systèmes les plus utilisés sont : le système binaire, le système octal, le système décimal et le système hexadécimal.

Se rappeler que : $a^0 = 1$

Si des opérations sont effectuées dans un système unique, il est inutile de préciser chaque fois la base du système. Ainsi lorsqu'on utilise uniquement le système à base 10, $(2356)_{10}$ s'écrit 2356.

Dans le cas d'un nombre décimal présentant une virgule (15,43 par exemple) la virgule sépare la partie entière de la partie fractionnaire. Lorsque le nombre est écrit sous la forme d'un polynôme, la virgule indique la séparation entre les puissances de dix positives et les puissances de dix négatives :
 $15,43 = 1 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 3 \times 10^{-2}$

La base d'un système de numération est le nombre de chiffres différents qu'utilise ce système de numération.

Système décimal

C'est le système à base 10 que nous utilisons tous les jours. Il comprend dix chiffres différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.

Prenons l'exemple du nombre 2356 de ce système ; nous l'écrivons $N = (2356)_{10}$.

L'indice 10 indique la base dans laquelle le nombre est écrit.

Ce nombre N peut être écrit sous la forme du polynôme suivant :

$$N = 2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

Dans chaque monôme, nous trouvons un chiffre du nombre N multiplié par une puissance de la base.

Si nous appelons α le chiffre et b la base, le monôme peut s'écrire sous la forme générale suivante : $\alpha \cdot b^p$

Dans cette expression :

- p est le **rang du chiffre** α , ce rang étant compté en commençant par la droite et le chiffre des unités ayant pour rang 0.

Exemple : dans le nombre N , 0 est le rang du chiffre 6, 1 est le rang du chiffre 5, 2 est le rang du chiffre 3, 3 est le rang du chiffre 2.

- b^p est le **poids du chiffre** α .

Exemple : 10^0 est le poids du chiffre 6, 10^1 est le poids du chiffre 5, 10^2 est le poids du chiffre 3, 10^3 est le poids du chiffre 2.

De manière générale, tout nombre décimal N entier de $n+1$ chiffres pourra s'écrire sous la forme :

$$N = \alpha_0 \times 10^0 + \alpha_1 \times 10^1 + \dots + \alpha_n \times 10^n$$

$$\text{Ou encore : } N = \sum_{i=0}^{i=n} \alpha_i \times 10^i$$

α_i est l'un des chiffres formant le nombre N ; l'entier positif i est le rang du chiffre α_i de poids 10^i ; n est le rang du chiffre de poids le plus fort, 0 est le rang du chiffre de poids le plus faible.

En utilisant n bits, on peut former 2^n nombres différents et le plus grand d'entre eux est égal à $(2^n - 1)$.

Exemple : avec un dispositif à 8 bits, on peut représenter $2^8 = 256$ nombres différents dont le plus grand est : $(11111111)_2 = (255)_{10}$.

Dans le système binaire, la succession des nombres par ordre croissant est la suivante :

- nombres à un chiffre : 0, 1 ;
- nombres à deux chiffres : 10, 11 ;
- nombres à trois chiffres : 100, 101, 110, 111, etc.

Dans le système octal, la succession des nombres par ordre croissant est la suivante :

- nombres à un chiffre : 0, 1, 2, 3, 4, 5, 6, 7 ;
- nombres à deux chiffres : 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, etc.

Dans le système hexadécimal, la succession des nombres par ordre croissant est la suivante :

- nombres à un chiffre : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Les symboles littéraux A, B, ..., F correspondent respectivement aux nombres décimaux 10, 11, ..., 16.
- nombres à deux chiffres : 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, ... etc.

Système binaire

Ce système dit à base 2 comprend deux symboles qui sont des chiffres : 0 et 1. Chacun d'eux est appelé aussi *bit* (contraction de *binary digit*) ou *élément binaire*.

Exemple : $N = (10110)_2$

Ce nombre, écrit sous la forme d'un polynôme, a pour expression :

$$N = (10110)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (22)_{10}$$

Le dernier chiffre de droite est le chiffre de poids le plus faible, son rang est 0 ; le 1 le plus à gauche est le chiffre de poids le plus fort, son rang est 4. L'expression générale d'un nombre binaire, présentée sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n} \alpha_i \times 2^i$$

avec $\alpha_i = 0$ ou 1.

Système octal

Ce système, dit à base 8, comprend huit symboles qui sont des chiffres : 0, 1, 2, 3, 4, 5, 6, 7.

Exemple : $N = (6543)_8$

3 est le chiffre de poids le plus faible ; 6 est le chiffre de poids le plus fort.

Ce nombre, écrit sous la forme d'un polynôme, a pour expression :

$$N = (6543)_8 = 6 \times 8^3 + 5 \times 8^2 + 4 \times 8^1 + 3 \times 8^0 = (3427)_{10}$$

L'expression générale d'un nombre octal présenté sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n} \alpha_i \times 8^i$$

Chaque α_i étant l'un des huit chiffres du système.

Système hexadécimal

Ce système, dit à base 16, comprend seize symboles, dix chiffres et six lettres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F.

Exemple : $N = (AC53)_{16}$

3 est le symbole de poids le plus faible ; A est le symbole de poids le plus fort.

Ce nombre, écrit sous la forme d'un polynôme, a pour expression :

$$\begin{aligned} N &= (AC53)_{16} = A \times 16^3 + C \times 16^2 + 5 \times 16^1 + 3 \times 16^0 \\ &= 10 \times 16^3 + 12 \times 16^2 + 5 \times 16^1 + 3 \times 16^0 = (44115)_{10} \end{aligned}$$

L'expression générale d'un nombre hexadécimal présentée sous la forme d'un polynôme est :

$$N = \sum_{i=0}^{i=n} \alpha_i \times 16^i$$

α_i étant l'un des chiffres de la base.

2 Changement de base

Conversion d'un nombre décimal en un nombre d'un système d'une autre base

Problème : un nombre N étant donné en base 10, cherchons à l'écrire dans un système de base b .

Première méthode : Nous cherchons le plus grand multiple de la plus grande puissance entière de b contenu dans N puis nous le retranchons de N ; il faut recommencer le processus avec le reste obtenu et ainsi de suite.

Exemple : conversion de $N = (3786)_{10}$ en nombre hexadécimal ($b = 16$).

Nous recherchons d'abord la plus grande puissance de 16 contenue dans N :

$$3786 < 16^3 = 4096 \quad \text{et} \quad 3786 > 16^2 = 256$$

Nous retenons donc : 16^2 .

Recherchons ensuite le plus grand multiple de 16^2 contenu dans N .

Nous obtenons : $3786 = 14 \times 16^2 + 202$.

Il faut recommencer avec le reste 202 et ainsi de suite jusqu'à l'obtention d'un reste nul. Soit :

$$202 = 12 \times 16^1 + 10 \quad \text{puis} \quad 10 = 10 \times 16^0$$

Ce qui donne :

$$N = (3786)_{10} = 14 \times 16^2 + 12 \times 16^1 + 10 \times 16^0$$

Soit encore :

$$N = (3786)_{10} = E \times 16^2 + C \times 16^1 + A \times 16^0$$

Donc : $N = (3786)_{10} = (ECA)_{16}$

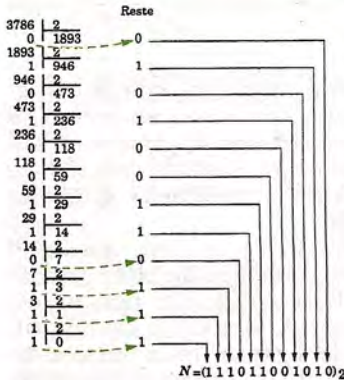
Deuxième méthode : Nous divisons le nombre décimal à convertir par la base b et nous conservons le reste. Le quotient obtenu est divisé par b et nous conservons le reste. Il faut répéter l'opération sur chaque quotient obtenu.

Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans le système de base b .

i	16^i
0	1
1	16
2	256
3	4096

Tableau 1

$$3786 = 14 \times 256 + 202$$



Exemple : Conversion de $N = (3786)_{10}$ en un nombre du système binaire ($b = 2$). La suite des divisions par 2 (fig. 1) conduit au résultat suivant :

$$N = (111011001010)_2$$

Autres conversions

- Conversion d'un nombre octal en un nombre binaire

Chaque symbole du nombre écrit dans le système octal est remplacé par son équivalent écrit dans le système binaire (voir le tableau 2).

Exemple :

$$N = (257)_8 \Rightarrow N = (\underbrace{010}_2 \underbrace{101}_5 \underbrace{111}_7)_2$$

- Conversion d'un nombre binaire en un nombre octal

C'est l'opération inverse de la précédente. Il faut donc regrouper les 1 et 0 du nombre par trois en commençant par la droite, puis chaque groupe est remplacé par le chiffre octal correspondant.

Exemple :

$$N = (11001101111)_2 = (\underbrace{11}_3 \underbrace{001}_1 \underbrace{101}_5 \underbrace{111}_7)_2$$

$$\Rightarrow N = (3157)_8$$

- Conversion d'un nombre hexadécimal en un nombre binaire

Chaque symbole du nombre écrit dans le système hexadécimal est remplacé par son équivalent écrit dans le système binaire (voir le tableau 2).

Exemple :

$$N = (ECA)_{16} = (\underbrace{1110}_E \underbrace{1100}_C \underbrace{1010}_A)_2$$

- Conversion d'un nombre binaire en un nombre hexadécimal

C'est l'inverse de la précédente. Il faut donc regrouper les 1 et 0 du nombre par quatre en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

Exemple :

$$N = (100001101111)_2 = (\underbrace{1000}_8 \underbrace{0110}_6 \underbrace{1111}_F)_2$$

$$\Rightarrow N = (86F)_{16}$$

base			
16	10	8	2
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

Tableau 2 : Correspondance entre nombres de différentes bases.

1 Nombres signés

Dans le système binaire décrit au chapitre précédent, nous n'avons tenu compte que des nombres positifs. La plupart des dispositifs numériques traitent également les nombres négatifs, ce qui impose de prendre en compte deux symboles supplémentaires : + et -.

Les dispositifs numériques « n'acceptant » que des 0 et des 1 aussi ajoute-t-on un autre bit pour symboliser le signe. Lorsqu'il est égal à 0, le nombre est positif; lorsqu'il est égal à 1, le nombre est négatif. Ce bit est appelé *bit de signe* et le nombre ainsi formé est dit *signé*. Ainsi, avec deux sortes de symboles, 0 et 1, il est possible de définir complètement un nombre binaire.

Notation en grandeur exacte

C'est la première notation qui vient à l'esprit. Elle comprend :

- un ensemble de bits qui représente la norme du nombre : ce sont les bits de grandeur;

exemple : $(47)_{10} = (101111)_2$

- un bit indiquant si le nombre est positif ou négatif : c'est le bit de signe.

Le bit de signe est le premier à gauche de l'ensemble de bits représentant le nombre binaire.

Exemples : $+ (47)_{10} = 0101111$ et $- (47)_{10} = 1101111$

Dans le cas d'un nombre binaire, le bit de poids le plus fort est appelé le MSB (most significant bit) et le bit de poids le plus faible est appelé le LSB (least significant bit).

Le complément de 1 est 0; le complément de 0 est 1.

Complément à 1 d'un nombre binaire

Complémenter à 1 un chiffre binaire revient à remplacer 0 par 1 et 1 par 0.

Complémenter à 1 un nombre binaire revient à complémenter à 1 chaque bit de grandeur.

Exemple : -47 s'écrit 1101111 en notation exacte et son complément à 1 est 1010000 , le bit de signe étant inchangé.

Complément à 2 d'un nombre binaire

Pour obtenir le complément à 2 d'un nombre binaire, il faut prendre le complément à 1 de ce nombre et lui ajouter 1.

Dans le cas d'une addition, un nombre binaire négatif est écrit sous la forme du complément à 2 de son grandeur car cela facilite la réalisation de l'opération.

Exemple :

-47 s'écrit 1101111 en notation exacte, son complément à 1 est 1010000 , son complément à 2 s'écrit 1010001 .

2 Addition

L'addition binaire classique est analogue à l'addition décimale. Il faut commencer par le bit de poids le plus faible en utilisant l'algorithme suivant :

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$ avec un report de 1

Lorsque l'on additionne deux 1 et un report de 1, l'algorithme précédent doit être complété par la ligne suivante :

$1 + 1 + 1 = 11$ soit 1 avec un report de 1

Exemple :

Report :	111111
1 ^{er} nombre	101101
2 ^e nombre	+ 110111
	<u>1100100</u>

Lors de l'addition en notation complément à 2, les deux nombres doivent avoir le même nombre de chiffres (on a tendance à dire même nombre de bits).

Les circuits numériques ne permettent pas d'additionner plus de deux nombres binaires à la fois. Les additions sont, en fait, une succession d'additions de deux nombres : le résultat de l'addition du premier et du deuxième est ajouté au troisième et ainsi de suite.

Dans les circuits numériques, une addition dure moins d'une microseconde et la succession d'additions qui remplace une addition de plus de deux nombres a une durée qui reste très brève.

L'addition binaire est l'opération la plus importante des circuits numériques car les autres opérations comme la soustraction, la multiplication et la division découlent de l'addition.

L'addition de nombres signés est utilisée dans la plupart des circuits numériques.

Trois cas peuvent se présenter :

- l'addition de deux nombres positifs,
- l'addition de deux nombres de signes contraires,
- l'addition de deux nombres négatifs.

La méthode consiste à écrire les nombres positifs en notation exacte et à remplacer les nombres négatifs par leur complément à deux avant d'additionner. Si le résultat est positif, il est en notation exacte, s'il est négatif, il est en notation complément à 2.

Addition de deux nombres positifs

Effectuons l'addition $(+17)_{10}$ et $(+12)_{10}$.

L'écriture de l'addition est la suivante :

1 ^{er} nombre $(+17)_{10}$		0 10001
2 ^e nombre $(+12)_{10}$	+	0 01100
		<u> </u>

Résultat	0 11101
	↑ bit de signe

Addition de deux nombres de signes contraires

Deux cas se présentent :

- la grandeur du nombre positif est supérieure à celle du nombre négatif.

Effectuons l'addition des nombres suivants :

$(+17)_{10} \rightarrow 010001$ et $(-12)_{10} \rightarrow 101100$

$(-12)_{10}$ étant négatif, il faut le remplacer par son complément à 2. Pour cela écrivons d'abord le complément à 1 de $(-12)_{10}$ (rappelons que les 1 deviennent des 0 et les 0 des 1, excepté le bit de signe) : $(-12)_{10} \Rightarrow 110011$

Puis écrivons le complément à 2 de $(-12)_{10}$ (en ajoutant 1 au complément à 1) : $(-12)_{10} \Rightarrow 110100$

Effectuons l'addition :

1 ^{er} nombre		0 10001
2 ^e nombre	+	1 10100
		<u> </u>

Résultat :	1 000101
	↑ bit de signe
	↓ débordement à éliminer

Remarque : nous avons additionné les bits de signe et la retenue; cela peut entraîner, comme dans le cas présent, un débordement qui est toujours rejeté. Ce résultat est général. La somme étant positive, le résultat est en notation exacte : $000101 = (+5)_{10}$

- La grandeur du nombre positif est inférieure à celle du nombre négatif.

Effectuons l'addition de $(-17)_{10}$ et de $(+12)_{10}$:

$$(-17)_{10} \Rightarrow 110001 \quad \text{et} \quad (+12)_{10} \Rightarrow 001100$$

Le complément à 2 de $(-17)_{10}$ s'écrit :

$$(-17)_{10} \Rightarrow 101111$$

L'écriture de l'addition est alors la suivante :

Report :	11
1 ^{er} nombre	101111
2 ^e nombre	+ 001100
	111011
Résultat :	111011 ↑ bit de signe

Cette somme est négative, le résultat est le complément à 2 du total cherché qui s'écrit en notation exacte : $100101 = (-5)_{10}$

Addition de deux nombres négatifs

Effectuons l'addition de $(-17)_{10}$ et de $(-12)_{10}$.

Les compléments à 2 s'écrivent :

$$(-17)_{10} \rightarrow 101111 \quad \text{et} \quad (-12)_{10} \rightarrow 110100$$

L'écriture de l'addition est alors la suivante :

Report :	1111
1 ^{er} nombre	101111
2 ^e nombre	+ 110100
	1100011
Résultat	1100011 ↑ bit de signe ↑ débordement à éliminer

Cette somme est négative, le résultat est le complément à 2 du résultat cherché qui s'écrit en notation exacte : $111101 = (-29)_{10}$

Remarque : Soustraire deux nombres peut se ramener à faire une addition. En effet, soustraire B de A revient à additionner au nombre A écrit sous sa forme exacte, le complément à 2 de B, bit de signe compris (ainsi un nombre positif devient négatif et un nombre négatif devient positif).

Nous nous plaçons dans le cas de nombres signés. Les circuits numériques font l'addition de nombres signés en utilisant la notation dite complément à 2. Lors de cette addition, le bit de signe est traité comme les bits de grandeur.

$$A - B = A + (-B)$$

Dans le cas de la soustraction de deux nombres binaires non signés, on peut aussi utiliser l'algorithme élémentaire suivant :

0 - 0 = 0
0 - 1 = 1 avec un report de 1
1 - 0 = 1
1 - 1 = 0

le report est à retrancher du chiffre de rang supérieur.

3 Multiplication

$$(45)_{10} \times (19)_{10} = (855)_{10}$$

Pour obtenir le résultat de la multiplication d'un nombre décimal par 10, en base 10, il suffit de lui ajouter un 0. De même on obtient le résultat de la multiplication d'un nombre binaire par $(10)_2$ en ajoutant un 0 à ce nombre.

Exemple : En base 2 :

$$10 \times 10 = 100; 1001 \times 10 = 10010$$

Ajouter deux nombres binaires égaux revient à multiplier par $(10)_2$ en base 2 donc à lui ajouter un zéro.

Exemple :

$$1 + 1 = 10;$$

$$1001 + 1001 = 10010$$

Algorithme :

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

La multiplication de deux nombres binaires se fait suivant l'algorithme indiqué dans la marge.

Exemple :

$$\begin{array}{r} 101101 \\ \times 10011 \\ \hline 101101 \\ 101101 \\ 101101 \dots \leftarrow \text{décalage dû aux deux 0} \\ \hline 110101011 \end{array}$$

Les circuits numériques n'additionnent pas l'ensemble des produits partiels directement mais deux par deux : le premier avec le deuxième puis la somme obtenue avec le troisième et ainsi de suite si nécessaire.

Si les nombres sont négatifs, ce sont leurs compléments à 2 qui sont pris en compte avant la multiplication et le produit est en notation exacte. Si l'un des nombres est négatif, on prend son complément à 2 avant la multiplication, le résultat est le complément à 2 du produit cherché.

4 Division

La méthode est identique à celle d'une division de deux nombres décimaux. Exemple :

$$\begin{array}{r} 110110 \mid 1010 \\ -1010 \leftarrow 101 \\ \hline 00111 \\ -00000 \leftarrow \\ \hline 1110 \\ -1010 \leftarrow \\ \hline 0100 \end{array}$$

Pour la clarté de l'opération, on a reporté dans la colonne de gauche le produit de chaque chiffre du quotient par le diviseur (en face de la flèche).

$$\text{Soit : } (110110)_2 = (1010)_2 \times (101)_2 + (100)_2$$

$$\text{ou encore, en base 10 : } 54 = 10 \times 5 + 4.$$

Si les nombres sont signés, la division s'effectue de la même façon que la multiplication :

- si les nombres sont négatifs, on prend leurs compléments à 2 avant la division, le quotient est en notation exacte;
- si l'un des nombres est négatif, on prend son complément à 2 avant la division, le résultat est le complément à 2 du quotient cherché.

Les systèmes numériques traitent des signaux qui représentent les symboles que sont les éléments binaires (bits) utilisés dans les chapitres précédents. La correspondance entre signaux et bits est définie par un code binaire. Les codes binaires permettent aussi la détection et la correction des erreurs.

1 Codes pondérés

Dans les codes pondérés, les positions des éléments binaires sont affectées d'un poids :

$$(N)_b = \sum_i a_i \cdot b^i$$

N est le nombre encore appelé **mot**, b est la **base**, i le **rang**, a_i le **chiffre** de rang i et b^i le **poids** du chiffre a_i .

Pour les calculatrices électroniques, les entrées (au clavier) comme les sorties (à l'affichage) sont des nombres décimaux. Les multimètres numériques affichent des nombres décimaux. Les traitements de données font appel au code DCB.

Code DCB : décimal codé binaire

Ce code conserve les avantages du système binaire naturel et du système décimal. A chaque chiffre d'un nombre du système décimal correspond son équivalent écrit dans le système binaire. Le code binaire du chiffre décimal le plus élevé 9 étant 1001, mot de quatre bits, chaque chiffre en notation binaire est codé avec 4 bits. D'où le tableau suivant :

Code décimal	0	1	2	3	4	5	6	7	8	9
Code DCB	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

L'écriture des 10 chiffres du code DCB est celle du code binaire naturel (avec éventuellement des 0 pour compléter à 4).

Très peu d'ordinateurs utilisent le code DCB car il nécessite plus de bits que le code binaire naturel, donc plus de place dans la mémoire de l'ordinateur. De plus les opérations en DCB sont plus complexes donc plus lentes.

Exemple : le nombre décimal 2583 s'écrit, en utilisant le code DCB :

$$\underbrace{0010}_2 \quad \underbrace{0101}_5 \quad \underbrace{1000}_8 \quad \underbrace{0011}_3$$

Remarques :

- Il y a 16 mots binaires différents de 4 bits. Le code DCB n'en retient que 10. Les mots 1010, 1011, 1100, 1101, 1110 et 1111 n'étant pas utilisés, l'apparition de l'un d'entre eux signifie qu'une erreur s'est produite.
- $(58)_{10}$ s'écrit 111010 en binaire naturel (mot de 6 bits); 0101 1000 en DCB (mot de 8 bits).

Ce résultat est général : le code DCB nécessite plus de bits que le code binaire naturel.

Les opérations arithmétiques en DCB sont plus complexes qu'en binaire naturel.

Exemple : Effectuons en DCB la somme correspondant à $(148)_{10} + (23)_{10}$; on utilise l'addition binaire classique pour chaque colonne de chiffres :

$$\begin{array}{r} 148 \\ + 23 \\ \hline 171 \end{array} \qquad \begin{array}{r} 0001\ 0100\ 1000 \\ + 0000\ 0010\ 0011 \\ \hline 0001\ 0110\ 1011 \\ (1\ 6\ «?»)_{10} \end{array}$$

L'une des combinaisons interdites apparaît dans la colonne de droite. Cela se produit chaque fois que la somme de deux chiffres décimaux donne un résultat supérieur à 9.

Pour effectuer la correction, on montre qu'il faut ajouter 6 soit 0110 en DCB (cela correspond au fait que l'on saute les 6 combinaisons non valides), d'où :

$$\begin{array}{r} 148 \\ + 23 \\ \hline 171 \end{array} \qquad \begin{array}{r} 0001\ 0100\ 1000 \\ + 0000\ 0010\ 0011 \\ \hline 0001\ 0110\ 1011 \\ + \qquad \qquad \qquad 0110 \\ \hline 0001\ 0111\ 0001 \\ (1\ 7\ 1)_{10} \end{array}$$

6 est appelé **facteur de correction**. Pour additionner en DCB, il faut effectuer une addition binaire classique pour les rangs dont la somme est inférieure à 9. Pour chacun des rangs donnant une somme supérieure à 9, il faut ajouter une correction de 6 (soit 0110 en DCB). Cela entraîne alors un report de 1 sur le rang placé immédiatement à gauche.

Code «plus 3» (ou majoré de trois)

C'est un code semblable au code DCB : 3 est ajouté à chaque chiffre décimal avant d'appliquer le code DCB. Ce qui donne le tableau de conversion suivant :

Code décimal	0	1	2	3	4	5	6	7	8	9
Code «plus trois»	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

Les trois combinaisons situées avant 0011 (c'est-à-dire 0000, 0001 et 0010) et les trois combinaisons situées après 1100 (c'est-à-dire 1101, 1110 et 1111) ne sont pas utilisées.

Exemple : $(2583)_{10}$ s'écrit en code «plus 3» :

$$\begin{array}{cccc} \underbrace{0101} & \underbrace{1000} & \underbrace{1011} & \underbrace{0110} \\ 2 & 5 & 8 & 3 \end{array}$$

2 Codes non pondérés

Dans les codes non pondérés, les positions binaires ne sont pas affectées d'un poids. Aussi ne conviennent-ils pas aux opérations arithmétiques.

Transitions en code Gray.

Si les transitions des différents bits ne sont pas simultanées, un codage binaire naturel risque d'introduire des états transitoires parasites. Le tableau ci-après montre que le passage de 3 à 4 en code binaire naturel peut donner lieu à l'affichage transitoire de nombres correspondant à 7 et 5 qui constituent des aléas.

Ce défaut est éliminé par l'utilisation du code Gray. Le passage de 3 à 4 ne nécessite qu'une seule modification (chiffre de gauche).

Code décimal	3		4
	(7)	(5)	
Code binaire naturel	011	→ 111	→ 101
			→ 100
Code Gray	010	→	110

Quelques éléments du code ASCII avec bit de parité paire :

Caractère	Élément du code
A	0100 0001
B	0100 0010
C	1100 0011
D	0100 0100
0	0011 0000
1	1011 0001
2	1011 0010
3	0011 0011
\$	0010 0100
(	0010 1000
)	1010 1001
,	1010 1100
-	0010 1101

Code cyclique

Souvent, dans les conversions analogiques-numériques, on a besoin d'un code dans lequel les grandeurs successives ne diffèrent que d'un caractère : c'est le cas du code de Gray encore appelé « binaire réfléchi ». Les positions binaires ne sont affectées d'aucun poids.

Exemple : Écriture de l'ensemble des nombres entiers décimaux compris entre 0 et 15 en binaire réfléchi.

0	0
1	1
11	2
10	3
110	4
111	5
101	6
100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

De 0 à 1, une position binaire suffit ; de 2 à 3, il en faut 2 ; de 4 à 7, il en faut 3 et de 8 à 15 il en faut 4. On peut vérifier que l'on passe d'un nombre au suivant en ne modifiant qu'un bit (exemple : 13 s'écrit 1011, 14 s'écrit 1001).

Code alphanumérique

Un ordinateur est prévu pour traiter des informations non numériques, c'est-à-dire qu'il doit « reconnaître » des caractères de l'alphabet, des caractères spéciaux, des chiffres, ... Tous ces éléments sont associés à un code appelé alphanumérique.

On a pour habitude de considérer que le minimum de caractères d'un code alphanumérique est 92 :

Le code ASCII est utilisé pour la transmission d'informations alphanumériques entre l'ordinateur et ses périphériques. La séquence de transmission est accompagnée de signaux indiquant le début (START) et la fin (STOP) du message. Cette transmission s'effectue avec une certaine « vitesse » exprimée en bauds : c'est l'inverse de la durée nécessaire à la transmission d'un bit lorsque celle-ci est exprimée en secondes :
 $300 \text{ bauds} \Leftrightarrow 300 \text{ bits/s}$.

26 caractères pour les lettres minuscules, 26 pour les majuscules, 10 pour les chiffres décimaux et 30 pour les caractères spéciaux (<, %, +, ...).

Le code le plus connu est le code ASCII (American Standard Code for Information Interchange). C'est un code quasi universel pour la transmission des informations.

Chaque symbole nécessite au moins 7 positions binaires, ce qui donne $2^7 = 128$ combinaisons binaires différentes (6 positions binaires offrent $2^6 = 64$ combinaisons, ce qui serait insuffisant pour le minimum de 92 caractères souhaités). Le plus souvent ce code est défini avec 8 bits, le huitième étant généralement un bit de parité permettant de détecter des erreurs de transmission comme nous le décrirons dans le paragraphe suivant.

3 Codes de détection d'erreurs

La transmission de données dans les circuits numériques s'accompagne parfois d'erreurs qui doivent être éliminées même si elles ne sont pas nombreuses. La méthode du bit de parité est l'une des méthodes utilisées pour les détecter. C'est un bit supplémentaire associé au nombre binaire.

Ces erreurs peuvent être dues à des tensions parasites se propageant sur les lignes de transmission.

On utilise le code de parité pour les erreurs dites de type « monobit » : en effet les erreurs sur deux bits, plus improbables, ne changent pas la parité du caractère codé et le code de parité est alors inutile.

Le code de parité permet de détecter des erreurs « monobit » mais non de les corriger car le récepteur de l'information ignore quel est le bit inexact.

Bit de parité paire

Le bit supplémentaire est fixé à une valeur (0 ou 1) telle que, pour chaque « mot », le nombre total de 1, y compris le bit de parité, soit pair.

Exemples : Un circuit numérique doit transmettre le caractère « , » (virgule); le code ASCII de ce caractère est 010 1100 : il est formé de trois 1, le bit de parité doit donc être 1 pour que le nombre total de 1 soit pair. Le code ASCII du caractère « , », avec bit de parité, est : 10101100 (le bit de parité est généralement placé à la gauche du nombre binaire).

Si le caractère à transmettre est « D », dont le code ASCII est 1000100, le bit de parité doit être 0 pour que le nombre total de 1 soit pair. Le code ASCII du caractère « D », avec bit de parité, est : 01000100.

Bit de parité impaire

C'est une méthode analogue à la précédente : le bit supplémentaire est fixé à une valeur (0 ou 1) telle que le nombre total de 1 dans le « mot », y compris le bit de parité, soit impair.

4 OPÉRATIONS LOGIQUES

1 Variables booléennes

Une variable logique (dite booléenne) est une grandeur binaire : elle peut prendre deux valeurs, 0 ou 1. Elle peut être utilisée pour représenter une proposition ou l'état d'un objet.

Exemple : Elle sera égale à 1 pour un moteur (ou tout autre dispositif) à l'état marche, 0 s'il est à l'état arrêt.

2 Opérations logiques élémentaires

Bien que le signe + soit aussi le symbole de l'addition binaire, nous verrons plus loin la différence entre *addition binaire* et *addition logique*.

Par convention, à une borne d'entrée ou de sortie désignée par une lettre capitale (par exemple E) nous associons une variable désignée au moyen de la même lettre mais écrite en italique (E dans l'exemple choisi). Rappelons que la valeur de cette variable est une tension électrique appliquée entre la borne considérée et la masse du montage.

Opérations

On peut définir des opérations mathématiques portant sur des variables logiques.

On définit trois opérations logiques élémentaires :
- l'*addition logique* généralement symbolisée par le signe + ;

- la *multiplication logique* généralement symbolisée par le signe ;

- l'*inversion ou complément* logique généralement symbolisée par le surlignement de la variable.

En électronique, des opérateurs dont les grandeurs d'entrée et de sortie sont des tensions représentant des variables logiques, permettent d'obtenir le résultat de ces opérations.

Rappelons qu'à l'état d'une entrée (ou d'une sortie) sont associées la valeur d'une variable d'entrée (ou de sortie) et la valeur d'une tension : l'état 1 correspond à une tension voisine de x volts (5 V par exemple), à l'état 0 correspond une tension voisine de y volts (0 V par exemple).

Table de vérité

E	S
0	1
1	0

Symbole

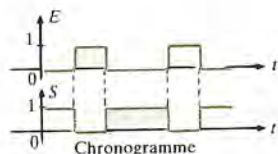
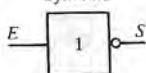


Fig. 1. Inversion logique.

Fonction à une variable logique

C'est une fonction à une variable d'entrée.

- Opération NON

L'opération NON est représentée par l'expression :

$$S = \bar{E}$$

Lire : « S est le COMPLÈMENT de E » ou « $S = \text{NON } E$ »

- Opérateur NON

L'opérateur NON (fig. 1), appelé couramment inverseur, effectue cette opération. C'est un opérateur à une seule entrée.

A une valeur $E = 0$ en entrée correspond la valeur $S = 1$ en sortie et, inversement, à une valeur 1 en entrée correspond la valeur 0 en sortie.

Fonctions à deux variables logiques

Ce sont des fonctions à deux variables d'entrée.

- Opération OU

L'opération OU s'écrit : $S = E_1 + E_2$

Lire : « S est égale à E_1 OU E_2 »

L'opération OU se nomme aussi *somme logique*.

- Opérateur OU

L'opérateur OU (fig. 2) est un opérateur à deux entrées.

A une valeur 1 à l'une des entrées ou aux deux entrées correspond la valeur 1 en sortie. A une valeur 0 aux deux entrées correspond la valeur 0 en sortie.

- Opération ET

L'opération ET s'écrit :

$$S = E_1 . E_2$$

Lire : « S est égale à E_1 ET E_2 »

La fonction ET se nomme aussi *produit logique*.

- Opérateur ET

L'opérateur ET (fig. 3) est un opérateur à deux entrées. A une valeur 1 aux deux entrées correspond la valeur 1 en sortie. A une valeur 0 à l'une des deux entrées correspond la valeur 0 en sortie.

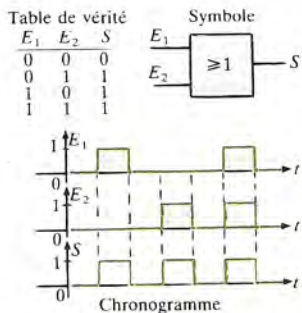


Fig. 2. Somme logique : fonction OU.

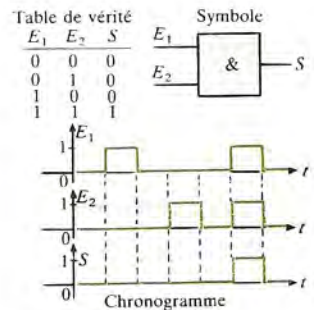


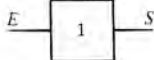
Fig. 3. Produit logique : fonction ET.

3 Autres opérations logiques de base

Table de vérité

E	S
0	0
1	1

Symbole



$$S = E$$

Lire : « S est IDENTIQUE à E »

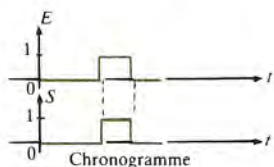


Fig. 4. Fonction OUI logique.

Table de vérité

E_1	E_2	S
0	0	1
0	1	0
1	0	0
1	1	0

Symbole

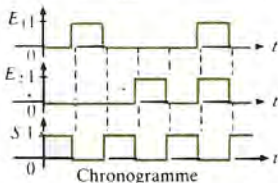
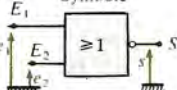


Fig. 5. Fonction NON-OU.

Table de vérité

E_1	E_2	S
0	0	1
0	1	1
1	0	1
1	1	0

Symbole

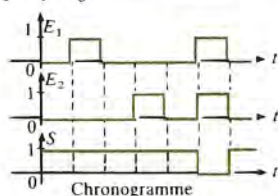
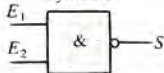


Fig. 6. Fonction NON-ET (NAND).

- Opération OUI

L'opération OUI s'écrit :

$$S = E$$

- Opérateur OUI

L'opérateur OUI (fig. 4) est un opérateur à une seule entrée. L'état logique de sortie est constamment identique à celui de l'entrée.

- Opération NON-OU

L'opération NON-OU permet l'obtention du complément d'une somme logique. Elle s'écrit :

$$S = E_1 + E_2$$

Lire : « S est égale au complément de ($E_1 + E_2$) »

- Opérateur NON-OU

L'opérateur NON-OU (fig. 5) comporte deux entrées. La sortie est à l'état 1 seulement si toutes les entrées sont à l'état 0.

- Opération NON-ET

L'opération NON-ET permet l'obtention du complément d'un produit logique. Elle s'écrit :

$$S = E_1 \cdot E_2$$

Lire : « S est égale au complément de ($E_1 \cdot E_2$) »

- Opérateur NON-ET

L'opérateur NON-ET (fig. 6) comporte deux entrées. Si l'une des entrées est à l'état 0, la sortie est à l'état 1. Elle est à l'état 0 si les deux entrées sont à l'état 1.

- Opération OU EXCLUSIF

L'opération OU EXCLUSIF est encore appelée opération d'anticoïncidence car la variable de sortie ne vaut 1 que lorsque les deux variables d'entrée ont des valeurs différentes.

L'opération OU EXCLUSIF s'écrit :

$$S = E_1 \cdot E_2 + \bar{E}_1 \cdot E_2$$

- Opérateur OU EXCLUSIF

L'opérateur OU EXCLUSIF (fig. 7) comporte deux entrées.

Une entrée E est dans l'état 1 si la variable $E = 1$. Elle est à l'état 0 si $E = 0$.
 $E = 0$ si la tension est inférieure à un certain seuil V_{OLmax} .
 $E = 1$ si la tension est supérieure à un certain seuil V_{OHmin} .

Le OU EXCLUSIF est parfois représenté par le symbole $\oplus$: ce qui donne l'équation logique suivante : $S = E_1 \oplus E_2$

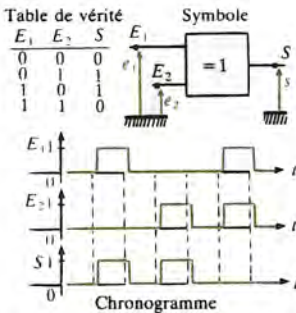


Fig. 7. Fonction OU EXCLUSIF.

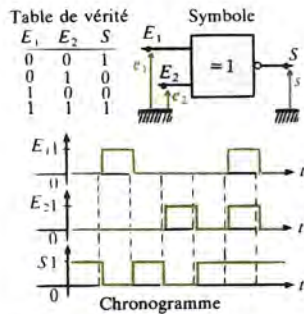


Fig. 8. Fonction NON-(OU EXCLUSIF).

- Opération NON-[OU EXCLUSIF]

L'opération NON-[OU EXCLUSIF] est encore appelée opération coincidence car la variable logique de sortie ne vaut 1 que si les deux variables d'entrée ont la même valeur logique.

L'opération NON-[OU EXCLUSIF] s'écrit :

$$S = \bar{E}_1 \cdot \bar{E}_2 + E_1 \cdot E_2$$

- Opérateur NON-[OU EXCLUSIF]

L'opérateur NON-[OU EXCLUSIF] (fig. 8) comporte deux entrées.

Annexe : Nous avons utilisé la norme française pour représenter les opérateurs logiques. La figure 9 donne la correspondance avec la représentation adoptée dans certains documents de constructeurs.

NON	OU/OR	ET/AND
NON-OU/NOR	NON-ET/NAND	OU EXCLUSIF

Fig. 9

1 Généralités

Écriture algébrique

C'est l'écriture qui permet, par les opérations somme logique et produit logique, de relier des variables booléennes entre elles pour exprimer la valeur d'une fonction logique.

Exemple :

Expression d'une fonction logique S, les variables étant E_1 , E_2 et E_3 :

$$S = E_1 \cdot E_2 \cdot E_3 + \bar{E}_1 \cdot \bar{E}_2 \cdot E_3 + \bar{E}_1 \cdot E_2 \cdot \bar{E}_3 + E_1 \cdot \bar{E}_2 \cdot \bar{E}_3$$

Table de vérité

C'est une table qui permet de connaître la valeur de la fonction logique S en fonction des diverses combinaisons des valeurs des variables E_1 , E_2 et E_3 .

Les valeurs des variables sont placées dans l'ordre binaire naturel.

Exemple : Table de vérité d'une fonction de trois variables :

$$S = E_1 \cdot E_2 \cdot E_3 + \bar{E}_1 \cdot \bar{E}_2 \cdot E_3 + \bar{E}_1 \cdot E_2 \cdot \bar{E}_3 + E_1 \cdot \bar{E}_2 \cdot \bar{E}_3$$

Pour une table de vérité :

- à deux entrées, il y a quatre lignes,
 - à trois entrées, il y a huit lignes,
 - à n entrées, il y a 2^n lignes.
- Cette représentation devient lourde lorsque le nombre de variables d'entrée est important. Nous verrons au chapitre 6 un autre type de tables permettant une simplification.

Pour l'exemple initial, la table de vérité s'écrit :

E_1	E_2	E_3	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

E_1	E_2	E_3	S
0	0	0	0
0	0	1	$1 \leftarrow \bar{E}_1 \cdot \bar{E}_2 \cdot E_3$
0	1	0	$1 \leftarrow \bar{E}_1 \cdot E_2 \cdot \bar{E}_3$
0	1	1	0
1	0	0	$1 \leftarrow E_1 \cdot \bar{E}_2 \cdot \bar{E}_3$
1	0	1	0
1	1	0	0
1	1	1	$1 \leftarrow E_1 \cdot E_2 \cdot E_3$

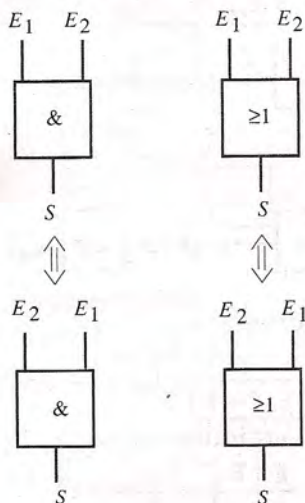
Le contenu de la dernière colonne est obtenu en calculant la valeur de la fonction S pour chaque ensemble de valeurs des trois variables E_1 , E_2 et E_3 .

Ces valeurs de S dépendent bien sûr de l'opération booléenne effectuée.

Dans le cas présent, le premier 1 de la colonne S (en descendant) correspond au terme $\bar{E}_1 \cdot \bar{E}_2 \cdot E_3$, le deuxième à $\bar{E}_1 \cdot E_2 \cdot \bar{E}_3$ et les suivants respectivement à $E_1 \cdot \bar{E}_2 \cdot \bar{E}_3$ et $E_1 \cdot E_2 \cdot E_3$.

2 Lois fondamentales

Les figures ci-après donnent quelques exemples de montages de principe permettant la vérification des lois fondamentales.



COMMUTATIVITE

On choisit trois variables booléennes E_1, E_2, E_3 et trois opérations NON, ET et OU.

Lois de fermeture

$E_1 \cdot E_2$ est une fonction booléenne définie par la table de vérité de l'opération ET.

$E_1 + E_2$ est une fonction booléenne définie par la table de vérité de l'opération OU.

Lois de commutativité

$$E_1 \cdot E_2 = E_2 \cdot E_1$$

$$E_1 + E_2 = E_2 + E_1$$

Lois d'associativité

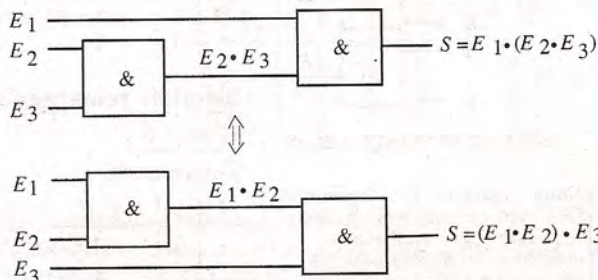
$$E_1 \cdot (E_2 \cdot E_3) = (E_1 \cdot E_2) \cdot E_3$$

$$E_1 + (E_2 + E_3) = (E_1 + E_2) + E_3$$

Vérification : nous nous bornerons à vérifier la validité de la dernière relation (la méthode appliquée dans cet exemple est générale) :

E_1	E_2	E_3	$E_2 + E_3$	$E_1 + (E_2 + E_3)$	$E_1 + E_2$	$(E_1 + E_2) + E_3$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

identité



ASSOCIATIVITE

Nous obtenons effectivement :

$$E_1 + (E_2 + E_3) = (E_1 + E_2) + E_3$$

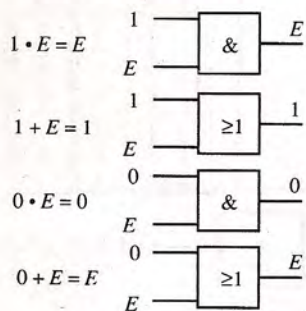
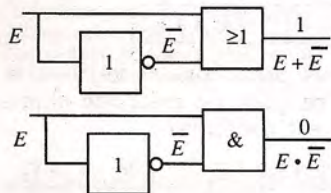
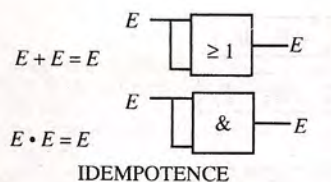
Lois de distributivité

- de l'opération ET sur OU

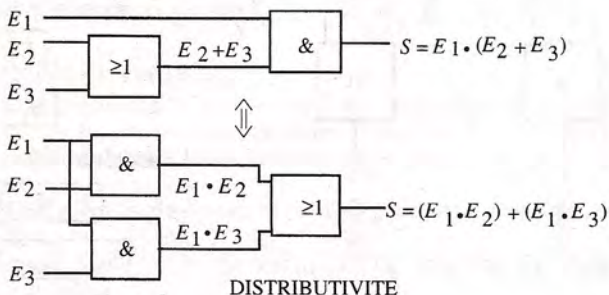
$$E_1 \cdot (E_2 + E_3) = (E_1 \cdot E_2) + (E_1 \cdot E_3)$$

- de l'opération OU sur ET

$$E_1 + (E_2 \cdot E_3) = (E_1 + E_2) \cdot (E_1 + E_3)$$



Nous voyons ici que l'opération OU, bien que notée d'un signe +, comme une addition ordinaire, en diffère puisque ici : $1 + 1 = 1$



Lois d'idempotence

$$E + E = E$$

Vérification :

E	E	$E + E$
0	0	0
1	1	1

$$E \cdot E = E$$

Vérification :

E	E	$E \cdot E$
0	0	0
1	1	1

Lois de complémentarité

$$E + \bar{E} = 1$$

Vérification :

E	$\bar{E}$	$E + \bar{E}$
0	1	1
1	0	1

$$E \cdot \bar{E} = 0$$

Vérification :

E	$\bar{E}$	$E \cdot \bar{E}$
0	1	0
1	0	0

Identités remarquables

$$1 \cdot E = E$$

Vérification :

1	E	$1 \cdot E$
1	0	0
1	1	1

$$1 + E = 1$$

Vérification :

1	E	$1 + E$
1	0	1
1	1	1

$$0 \cdot E = 0$$

Vérification :

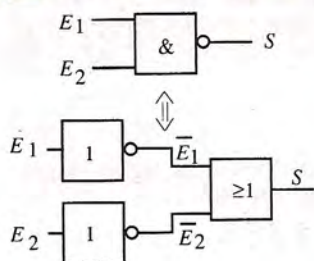
0	E	$0 \cdot E$
0	0	0
0	1	0

$$0 + E = E$$

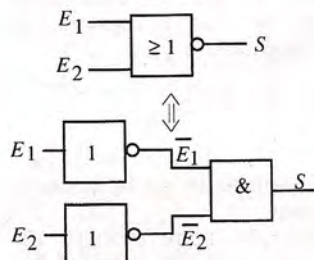
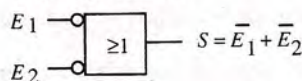
Vérification :

0	E	$0 + E$
0	0	0
0	1	1

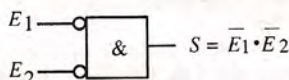
3 Théorèmes de De Morgan (Augustus)



Autre représentation



Autre représentation



Théorème 1 :

Le produit logique complémenté de deux variables booléennes est égal à la somme logique des compléments de ces variables :

$$\overline{E_1 \cdot E_2} = \bar{E}_1 + \bar{E}_2$$

Vérification :

E_1	E_2	$E_1 \cdot E_2$	$\overline{E_1 \cdot E_2}$	$\bar{E}_1 + \bar{E}_2$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

Les dernières colonnes sont identiques.

Remarque :

Cette relation peut être généralisée à n variables ; pour quatre variables nous avons :

$$\overline{E_1 \cdot E_2 \cdot E_3 \cdot E_4} = \bar{E}_1 + \bar{E}_2 + \bar{E}_3 + \bar{E}_4$$

Théorème 2

La somme logique complémentée de deux variables booléennes est égale au produit logique des compléments de ces variables :

$$\overline{E_1 + E_2} = \bar{E}_1 \cdot \bar{E}_2$$

Vérification :

E_1	E_2	$E_1 + E_2$	$\overline{E_1 + E_2}$	$\bar{E}_1 \cdot \bar{E}_2$
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

Les dernières colonnes sont identiques.

Remarque : Cette relation peut être généralisée à n variables ; pour quatre variables nous avons :

$$\overline{E_1 + E_2 + E_3 + E_4} = \bar{E}_1 \cdot \bar{E}_2 \cdot \bar{E}_3 \cdot \bar{E}_4$$

Se rappeler que le rond placé à l'entrée des symboles signifie une inversion logique, ce qui évite la représentation du symbole de l'inverseur.

Toutes les expressions booléennes peuvent donc être obtenues en utilisant des opérations NON, ET et OU. Cependant, il est possible de réaliser toutes les opérations booléennes au moyen d'une seule sorte d'opérateurs : opérateurs NON-ET ou opérateurs NON-OU par exemple.

1 Utilisation d'opérateurs NON-ET

Dans ce qui suit, S représente l'état de la sortie d'un montage. C'est la fonction logique dépendant des variables logiques (E ou E_1 , E_2 , etc.) figurées par l'état des entrées.

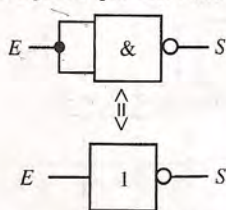


Fig. 1. Inverseur logique : opérateur NON-ET ou (NAND) à deux entrées réunies.

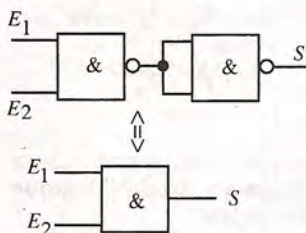


Fig. 2. Produit logique.

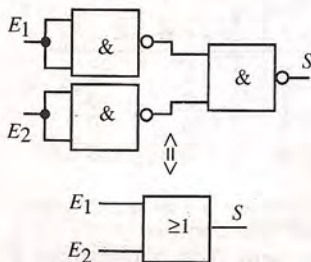


Fig. 3. Somme logique.

Par un montage convenable des opérateurs NON-ET, on réalise soit l'inversion ou complémentation logique, soit le produit logique, soit la somme logique.

Inversion logique

L'opérateur NON-ET monté selon la figure 1 réalise une inversion logique. En effet :

$$S = \overline{E} \cdot \overline{E} \Rightarrow S = \overline{E}.$$

Produit logique

Deux opérateurs NON-ET montés selon la figure 2 réalisent un produit logique.

En effet, à la sortie du premier opérateur, la variable a pour expression $\overline{E_1} \cdot \overline{E_2}$. Le deuxième opérateur étant monté en inverseur logique, la variable de sortie a pour expression :

$$S = \overline{\overline{E_1} \cdot \overline{E_2}}, \text{ soit : } S = E_1 \cdot E_2.$$

Somme logique

Trois opérateurs NON-ET montés selon la figure 3 réalisent une somme logique.

Les deux premiers opérateurs sont montés en inverseurs logiques et donnent en sortie les variables $\overline{E_1}$ et $\overline{E_2}$. A la sortie du troisième opérateur, la variable a pour expression :

$$S = \overline{\overline{E_1} \cdot \overline{E_2}}.$$

En appliquant le théorème de De Morgan nous obtenons :

$$S = \overline{\overline{E_1} \cdot \overline{E_2}} \Rightarrow S = E_1 + E_2$$

2 Utilisation d'opérateurs NON-OU

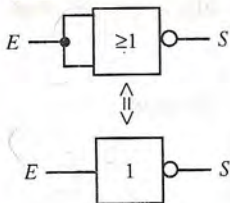


Fig. 4. Inverseur logique : opérateur NON-OU (ou NOR) à deux entrées réunies.

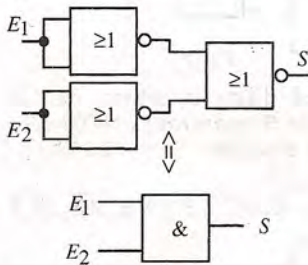


Fig. 5. Produit logique.

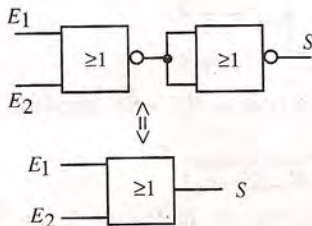


Fig. 6. Somme logique.

Par un montage convenable des opérateurs NON-OU, on réalise également soit l'inversion logique, soit le produit logique, soit la somme logique.

Inversion logique

L'opérateur NON-OU monté selon la figure 4 réalise une inversion logique. En effet :

$$S = \overline{E + E} \Rightarrow S = \overline{E}.$$

Produit logique

Trois opérateurs NON-OU montés selon la figure 5 réalisent un produit logique.

Les deux premiers opérateurs sont montés en inverseurs logiques et donnent en sortie les variables $\overline{E}_1$ et $\overline{E}_2$. A la sortie du troisième opérateur, la variable a pour expression :

$$S = \overline{\overline{E}_1 + \overline{E}_2}.$$

En appliquant l'un des théorèmes de De Morgan nous obtenons :

$$S = \overline{\overline{E}_1} \cdot \overline{\overline{E}_2} \rightarrow S = E_1 \cdot E_2$$

Somme logique

Deux opérateurs NON-OU montés selon la figure 6 réalisent une somme logique. En effet, la variable de sortie du premier opérateur, a pour expression $\overline{E}_1 + \overline{E}_2$. Le deuxième opérateur étant monté en inverseur logique, la variable S de sortie a pour expression :

$$S = \overline{\overline{E}_1 + \overline{E}_2} \Rightarrow S = E_1 + E_2.$$

3 Exemples d'application

E_1	E_2	S
0	0	0
0	1	1
1	0	1
1	1	0

$\leftarrow \overline{E}_1 \cdot E_2$
 $\leftarrow E_1 \cdot \overline{E}_2$

Tableau 1 : OU EXCLUSIF

Opérateurs NON-ET

Recherchons le montage d'opérateurs correspondant à la fonction :

$$S = E_1 \cdot \overline{E}_2 + \overline{E}_1 \cdot E_2$$

pour laquelle nous dressons la table de vérité ci-contre (tableau 1).

L'implantation de cette somme de produits peut s'effectuer grâce à des opérateurs NON, des opé-

rateurs ET et des opérateurs OU (fig. 7), mais elle peut aussi être réalisée à partir d'opérateurs NON-ET. Nous remarquons, en effet, que S peut s'écrire :

$$S = \bar{\bar{S}}$$

Or, d'après le théorème 2 de De Morgan :

$$\bar{S} = \overline{E_1 \cdot \bar{E}_2 \cdot \bar{E}_1 \cdot E_2}$$

Soit :

$$S = E_1 \cdot \bar{E}_2 \cdot \bar{E}_1 \cdot E_2$$

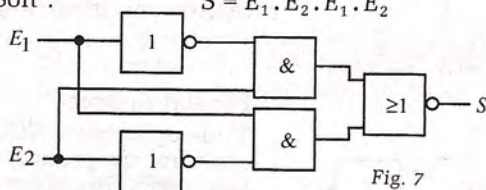


Fig. 7

Cette écriture conduit à l'implantation de la figure 8, réalisée à partir de 5 opérateurs NON-ET.

Remarque : S peut encore s'écrire :

$$S = E_1 \cdot (\bar{E}_1 + \bar{E}_2) + E_2 \cdot (\bar{E}_1 + \bar{E}_2) = \overline{E_1 \cdot (E_1 \cdot E_2)} + \overline{E_2 \cdot (E_1 \cdot E_2)}$$

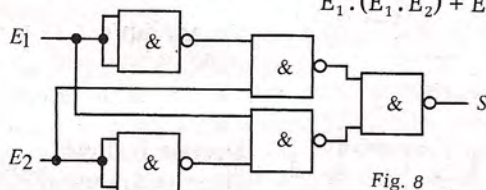


Fig. 8

La double inversion précédente ($S = \bar{\bar{S}}$) conduit alors à l'expression :

$$\overline{E_1 \cdot (E_1 \cdot E_2) \cdot E_2 \cdot (E_1 \cdot E_2)}$$

Celle-ci justifie l'implantation de la figure 9 réalisée à l'aide de 4 opérateurs NON-ET.

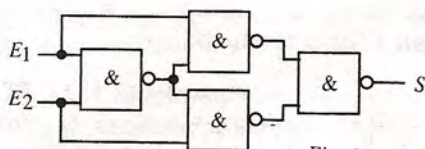


Fig. 9

Opérateurs NON-OU

En reprenant le même exemple ($S = E_1 \cdot \bar{E}_2 + \bar{E}_1 \cdot E_2$) et en remarquant que S peut encore s'écrire :

$$S = (E_1 + E_2) \cdot (\bar{E}_1 + \bar{E}_2)$$

nous en déduisons, en appliquant la propriété $S = \overline{\overline{S}}$, une nouvelle expression de S :

$$S = \overline{(\overline{E_1 + E_2}) + (\overline{E_1 + E_2})}$$

Cette écriture justifie l'implantation de la figure 10, réalisée à l'aide de 5 opérateurs NON-OU.

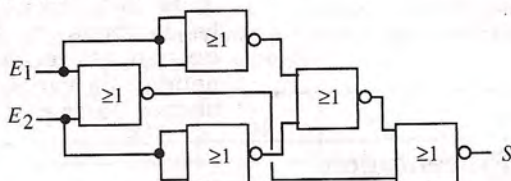


Fig. 10

4 Représentations équivalentes des opérateurs logiques

Pour faire le schéma des implantations, nous avons représenté les opérateurs logiques par leurs symboles normalisés. De plus en plus, d'autres symboles, découlant des symboles standard, sont utilisés : ils sont obtenus en complémentant entrées et sorties ainsi que par remplacement de l'opérateur ET par un opérateur OU, ou de l'opérateur OU par un opérateur ET. Le tableau 2 donne ces autres symboles.

E_1 E_2 $S = E_1 \cdot E_2$	$\Leftrightarrow$ ET	E_1 E_2 $S = \overline{E_1 + E_2}$ $S = \overline{E_1} \cdot \overline{E_2}$
E_1 E_2 $S = E_1 + E_2$	$\Leftrightarrow$ OU	E_1 E_2 $S = \overline{E_1} \cdot \overline{E_2}$ $S = \overline{E_1 + E_2}$
E_1 E_2 $S = \overline{E_1 \cdot E_2}$	$\Leftrightarrow$ NON-ET	E_1 E_2 $S = \overline{E_1 + E_2}$ $S = \overline{E_1} \cdot \overline{E_2}$
E_1 E_2 $S = \overline{E_1 + E_2}$	$\Leftrightarrow$ NON-OU	E_1 E_2 $S = \overline{E_1} \cdot \overline{E_2}$ $S = \overline{E_1 + E_2}$

Tableau 2

La méthode de la table de Karnaugh (ou *diagramme de Karnaugh*) permet d'écrire une équation booléenne, de la simplifier et de déduire une implantation des composants pour le montage correspondant.

Lorsque le nombre de variables devient important, le nombre de lignes d'une table de vérité devient également important rendant difficile l'emploi d'une telle table.

La méthode de la table de Karnaugh atténue un peu cet inconvénient. Cependant dès que le nombre de variables est supérieur à six, les problèmes posés sont traités par informatique.

1 Présentation

Une table dite de Karnaugh est constituée de lignes et de colonnes en nombre tel que la table soit la plus « carrée » possible.

Avec un nombre n pair de variables d'entrée ($n = 2p$), la table sera formée de 2^p lignes et 2^p colonnes (avec $n = 4$, il y a 2^2 lignes et 2^2 colonnes).

Avec un nombre n impair de variables d'entrée ($n = 2p + 1$), la table sera formée soit de 2^p lignes et 2^{p+1} colonnes soit de 2^{p+1} lignes et 2^p colonnes (avec $n = 3$, il y a 4 lignes et 2 colonnes ou 2 lignes et 4 colonnes).

Pour quatre variables E_1, E_2, E_3 et E_4 , la disposition est la suivante : la colonne correspondant à E_1E_2 et la ligne correspondant à E_3E_4 sont numérotées selon le code *binnaire réfléchi*.

Dans chacune des cases du tableau, nous écrivons l'expression logique correspondant au produit des variables égales à 1 et du complément des variables égales à 0. Ce qui donne

$E_1E_2 \backslash E_3E_4$	E_3E_4 0 0	E_3E_4 0 1	E_3E_4 1 1	E_3E_4 1 0
E_1E_2 0 0	$\bar{E}_1 \cdot \bar{E}_2 \cdot \bar{E}_3 \cdot \bar{E}_4$	$\bar{E}_1 \cdot \bar{E}_2 \cdot \bar{E}_3 \cdot E_4$	$\bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \cdot E_4$	$\bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \cdot \bar{E}_4$
E_1E_2 0 1	$\bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \cdot \bar{E}_4$	$\bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \cdot E_4$	$\bar{E}_1 \cdot E_2 \cdot E_3 \cdot E_4$	$\bar{E}_1 \cdot E_2 \cdot E_3 \cdot \bar{E}_4$
E_1E_2 1 1	$E_1 \cdot E_2 \cdot \bar{E}_3 \cdot \bar{E}_4$	$E_1 \cdot E_2 \cdot \bar{E}_3 \cdot E_4$	$E_1 \cdot E_2 \cdot E_3 \cdot E_4$	$E_1 \cdot E_2 \cdot E_3 \cdot \bar{E}_4$
E_1E_2 1 0	$E_1 \cdot \bar{E}_2 \cdot \bar{E}_3 \cdot \bar{E}_4$	$E_1 \cdot \bar{E}_2 \cdot \bar{E}_3 \cdot E_4$	$E_1 \cdot \bar{E}_2 \cdot E_3 \cdot E_4$	$E_1 \cdot \bar{E}_2 \cdot E_3 \cdot \bar{E}_4$

Pour associer une table de Karnaugh à une expression, il est préférable d'écrire celle-ci sous la forme d'une somme de produits car chaque produit est l'adresse d'une case ou d'un groupe de cases.

2 Utilisation

Exemple : Représentation de la table de Karnaugh de l'expression suivante :

$$S = E_1 \cdot \bar{E}_2 \cdot \bar{E}_3 + \bar{E}_1 \cdot E_2 \cdot \bar{E}_3 + \bar{E}_1 \cdot \bar{E}_2 \cdot E_3 + E_1 \cdot E_2 \cdot E_3$$

Avec trois variables, nous pouvons construire une table à deux lignes et quatre colonnes ou une table à quatre lignes et deux colonnes. Construisons, arbitrairement, une table à deux lignes et quatre colonnes, nous pouvons avoir :

$E_1 \backslash E_2 E_3$	00	01	11	10
0	I	II	III	IV
1	V	VI	VII	VIII

On convient de placer 1 dans une case si le produit qu'elle contient appartient à la combinaison étudiée. Dans le cas contraire, on y place un 0.

La case I correspond au produit $\bar{E}_1 \cdot \bar{E}_2 \cdot \bar{E}_3$ qui n'appartient pas à l'expression étudiée : on y place un 0. La case II correspond au produit $\bar{E}_1 \cdot \bar{E}_2 \cdot E_3$ qui appartient à l'expression étudiée : on y place un 1. En procédant de cette façon pour toutes les autres cases, nous obtenons le tableau suivant :

$E_1 \backslash E_2 E_3$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Ce tableau de Karnaugh ainsi obtenu est strictement équivalent à l'équation donnant S.

Réciproquement, à partir de la table de Karnaugh, nous pouvons retrouver la fonction S : S est la somme des produits contenus dans les cases où sont placés les 1.

Remarques :

- Chaque case de la table de Karnaugh correspond à une ligne de la table de vérité.
- Deux cases adjacentes, verticalement ou horizontalement, ne diffèrent que d'une variable.

Exemple :

- case II de la table de Karnaugh $\bar{E}_1 \cdot \bar{E}_2 \cdot E_3$,
- case III de la table de Karnaugh $\bar{E}_1 \cdot E_2 \cdot E_3$,
- case VII de la table de Karnaugh $E_1 \cdot E_2 \cdot E_3$.

Exemple :

$$S = \overline{E_1 \cdot E_3 + E_2 \cdot E_4}$$

En utilisant les théorèmes de Morgan nous avons :

$$\begin{aligned} S &= \overline{E_1 \cdot E_3} \cdot \overline{E_2 \cdot E_4} \\ &= (\overline{E_1} + \overline{E_3}) \cdot (\overline{E_2} + \overline{E_4}) \end{aligned}$$

Soit :

$$S = \overline{E_1} \cdot \overline{E_2} + \overline{E_1} \cdot \overline{E_4} + \overline{E_3} \cdot \overline{E_2} + \overline{E_3} \cdot \overline{E_4}$$

Nous obtenons la table de Karnaugh suivante :

$E_1 E_2 \backslash E_3 E_4$	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	0	0	0
10	1	1	0	0

3 Applications

Si le tableau met en évidence des cases adjacentes comportant des 1, les expressions qui leur correspondent peuvent être regroupées pour simplifier l'expression de la fonction S.

Deux cases sont dites adjacentes lorsqu'elles ont un côté commun ou lorsque l'une est la dernière de la colonne (ou de la ligne) et l'autre la première.

Réunion de doublets

En réunissant les expressions correspondant au contenu de deux cases adjacentes (verticales ou horizontales) comportant des 1, la simplification consiste à éliminer la variable qui change d'état.

Exemple : soit la table de Karnaugh suivante :

$E_1 E_2 \backslash E_3 E_4$	00	01	11	10
00	1 I	1 II	0 III	0 IV
01	0 V	0 VI	1 VII	0 VIII
11	0 IX	0 X	1 XI	1 XII
10	0 XIII	1 XIV	0 XV	1 XVI

Nous avons réuni les 1 dans des cases adjacentes.

Autre exemple de réunion de quartets : $S = \bar{E}_2 \cdot E_4 + E_1 \cdot E_2$

$E_3 E_4$ $E_1 E_2$	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	1	1	1	1
10	1	0	0	1

Remarque : Dans certains circuits, il est possible qu'une ou plusieurs combinaisons des variables d'entrée ne correspondent à aucun état particulier de la variable de sortie; cela se produit par exemple lorsque ces combinaisons ne doivent jamais exister. Dans la table de Karnaugh (comme dans la table de vérité) l'état de la sortie correspondant à ces combinaisons (dites indifférentes) est schématisé par une croix : X. Pour la simplification par réunion de 1 adjacents, les croix peuvent être remplacées par des 0 ou des 1 afin d'obtenir l'expression de la variable de sortie la plus simple.

Exemple :

$E_3 E_4$ $E_1 E_2$	00	01	11	10
00	1	1	0	0
01	1	X	0	0
11	0	0	0	1
10	0	0	X	1

L'expression de la variable de sortie sera la plus simple si les X sont remplacés par 1 ou 0 de la façon suivante :

$E_3 E_4$ $E_1 E_2$	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	0	0	0	1
10	0	0	0	1

D'où :

$$S = \bar{E}_1 \cdot \bar{E}_3 + E_1 \cdot E_3 \cdot \bar{E}_4$$

Il faut remarquer que :

- les cases II et XIV sont adjacentes,
 - les cases XI et XII, bien qu'adjacentes, n'ont pas été reliées car elles sont toutes deux prises en compte par ailleurs.
- En appliquant la règle de réunion des doublets, nous trouvons :

$$S = \bar{E}_1 \cdot \bar{E}_2 \cdot \bar{E}_3 + \bar{E}_2 \cdot \bar{E}_3 \cdot E_4 + E_2 \cdot E_3 \cdot E_4 + E_1 \cdot E_3 \cdot \bar{E}_4$$

Réunion de quartets

En réunissant les expressions correspondant au contenu de quatre cases adjacentes (verticales ou horizontales) comportant des 1, la simplification consiste à éliminer les deux variables qui changent d'état.

Exemple : soit la table de Karnaugh suivante :

$E_3 E_4$ $E_1 E_2$	00	01	11	10
00	1 I	1 II	0 III	1 IV
01	1 V	1 VI	0 VII	1 VIII
11	0 IX	0 X	0 XI	1 XII
10	0 XIII	1 XIV	0 XV	1 XVI

En appliquant la règle de réunion des quartets, nous obtenons : $S = \bar{E}_1 \cdot \bar{E}_3 + E_3 \cdot \bar{E}_4$

En effet :

- les seules variables communes aux cases I, II, V et VI sont $\bar{E}_1$ et $\bar{E}_3$: il faut éliminer les autres variables et il reste donc le produit $\bar{E}_1 \cdot \bar{E}_3$;
- les seules variables communes aux cases IV, VIII, XII et XVI sont E_3 et $\bar{E}_4$: il faut éliminer les autres variables et il reste donc le produit $E_3 \cdot \bar{E}_4$.

Réunion d'octets

En réunissant les expressions correspondant au contenu de huit cases adjacentes (verticales ou horizontales) comportant des 1, la simplification consiste à éliminer les trois variables qui changent d'état.

Remarque :

En réunissant les expressions correspondant aux cases adjacentes comportant des zéros nous obtenons l'expression de S. Ainsi pour l'exemple donné dans la marge nous pouvons écrire :

$$\bar{S} = E_1 \cdot \bar{E}_3 + E_3 \cdot E_4 + \bar{E}_1 \cdot E_3$$

Les opérateurs logiques NON, NON-ET, NON-OU... sont réalisés sous la forme de circuits intégrés. En général, chaque boîtier comporte plusieurs opérateurs identiques. Il existe deux grandes familles logiques : TTL (Transistor Transistor Logic) et CMOS (Complementary Metal Oxide Semiconductor).

Pour une application, le choix d'une famille logique est déterminé par ses caractéristiques : tension d'alimentation, consommation, caractéristique de sortie, immunité au bruit, temps de propagation...

Les opérateurs logiques se prêtent bien à la technique de l'intégration. Suivant le nombre d'opérateurs intégrés dans une même puce, l'intégration est dite à petite échelle (symbolisée par le sigle SSI : Small Scale Integration), à moyenne échelle (MSI ou Mean Scale Integration), à grande échelle (LSI ou large scale integration) ou à très grande échelle (VLSI ou Very Large Scale Integration).

Exemples :

- composants SSI : opérateurs élémentaires et bascules,
- composants MSI : compteurs et registres,
- composants LSI et VLSI : microprocesseurs.

1 Niveaux de tension et de courant

Niveaux de tension (fig. 1 et fig. 2)

- Niveau haut de tension d'entrée : V_{iHmin} .
C'est la tension minimale nécessaire à appliquer à l'entrée pour qu'elle soit à l'état 1.
- Niveau bas de tension d'entrée : V_{iLmax} .
C'est la tension maximale que l'on peut appliquer à l'entrée pour qu'elle soit à l'état 0.
- Niveau haut de tension de sortie : V_{oHmin} .
C'est la tension minimale de sortie correspondant à l'état haut.
- Niveau bas de tension de sortie : V_{oLmax} .
C'est la tension maximale de sortie correspondant à l'état bas.

Niveaux de courant

- Niveau haut de courant d'entrée : I_{iH} .
C'est l'intensité du courant traversant l'entrée lorsqu'elle est soumise à la tension V_{iH} .
- Niveau bas de courant d'entrée : I_{iL} .
C'est l'intensité du courant traversant l'entrée lorsqu'elle est soumise à la tension V_{iL} .

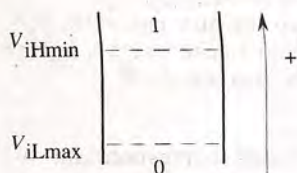


Fig. 1. Tensions d'entrée.

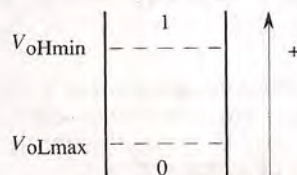


Fig. 2. Tensions de sortie.

- Niveau haut de courant de sortie : I_{oH} .
C'est l'intensité du courant traversant la sortie lorsqu'elle est au niveau haut de tension et pour une charge déterminée.
- Niveau bas de courant de sortie : I_{oL} .
C'est l'intensité du courant traversant la sortie lorsqu'elle est au niveau bas de tension et pour une charge déterminée.

2 Autres caractéristiques

Sortance

- Sortance à l'état haut : la sortance à l'état haut d'un circuit logique est le nombre maximal de charges logiques standard que peut commander sa sortie pour y imposer un 1 logique en régime dynamique.
- Sortance à l'état bas : la sortance à l'état bas d'un circuit logique est le nombre maximal de charges logiques standard que peut commander sa sortie pour y imposer un 0 logique en régime dynamique. Dans chacun des états, si ce nombre est dépassé, le fonctionnement des circuits commandés peut être défectueux.

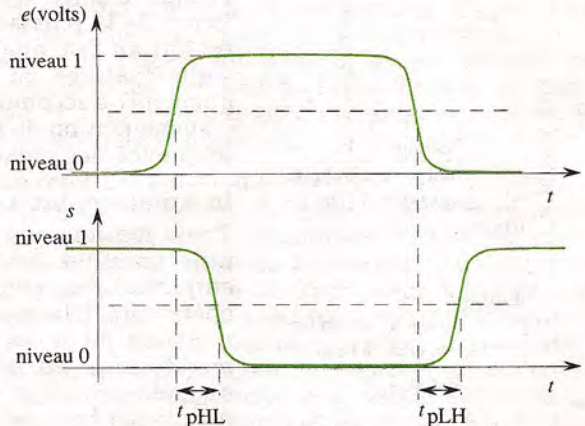


Fig. 3. Retards de propagation d'un circuit inverseur :

$$t_p = \frac{t_{pHL} + t_{pLH}}{2}$$

Retards de propagation

Le temps de propagation t_p d'un circuit logique est défini par la durée que met un signal logique pour « franchir » ce circuit. Cette durée de propagation varie selon que l'on passe du niveau 0 au niveau 1 ou du niveau 1 au niveau 0. Dans le premier cas

cette durée est appelée t_{pHL} et dans l'autre t_{pLH} (fig. 3). Ces deux retards sont généralement différents et varient avec la charge. On définit une durée moyenne qui représente le retard de propagation t_p :

$$t_p = \frac{t_{pHL} + t_{pLH}}{2}$$

Consommation

- Consommation statique : tout circuit numérique, au repos, est traversé par un courant électrique dont l'intensité diffère suivant que les sorties sont à l'état bas ou à l'état haut. Si V_{cc} est la tension délivrée par l'alimentation, I_{ccl} l'intensité du courant qu'elle fournit avec les sorties à l'état bas et I_{cch} l'intensité qu'elle fournit avec les sorties à l'état haut, la puissance moyenne consommée est définie par :

$$P_0 = V_{cc} \left(\frac{I_{ccl} + I_{cch}}{2} \right)$$

- Consommation dynamique : lorsque sous l'influence d'une ou plusieurs entrées variables (signaux rectangulaires), la sortie d'un circuit logique change d'état, on constate une augmentation de la puissance moyenne consommée. Cela est dû au fait que les régimes transitoires de la sortie (passage de l'état 1 à l'état 0 et réciproquement) s'accompagnent de pointes de courant. L'augmentation de puissance moyenne est liée à la fréquence des régimes transitoires.

Immunité au bruit

Toute tension parasite qui se superpose au signal utile constitue un bruit. Ce type de signal peut entraîner des commutations intempestives des opérateurs. L'immunité au bruit est mesurée par la variation du niveau d'entrée n'entraînant pas de modification du niveau de sortie; cette mesure s'appelle *marge de sensibilité aux bruits*.

Au niveau haut, elle est égale à $V_{oHmin} - V_{iHmin}$. Au niveau bas, elle est égale à $V_{iLmax} - V_{oLmax}$ (fig. 4).

Remarque : Un bruit assez faible pour ne pas entraîner de commutation intempestive est sans influence sur un système logique. Ce n'est pas le cas avec un système analogique où toute altération du signal au niveau d'un étage est répercutée sur les suivants (souvent avec amplification!).

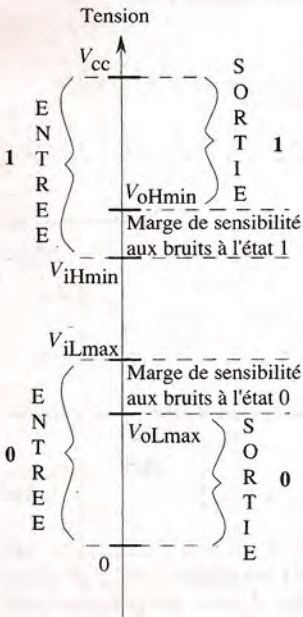


Fig. 4.

Facteur de mérite

C'est un coefficient qui permet de comparer les différentes familles logiques entre elles. Par convention, le facteur de mérite est égal au produit de la consommation statique par la durée de propagation. Ce produit est généralement exprimé en picojoules.

3 Association d'opérateurs en cascade

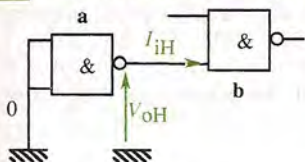


Fig. 5. Cascade à injection de courant.

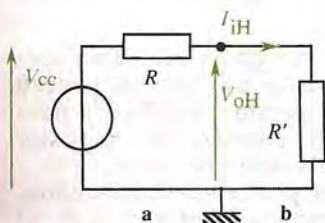


Fig. 6.

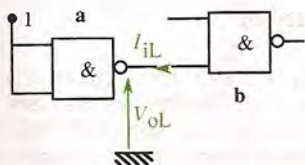


Fig. 7. Cascade à absorption de courant.

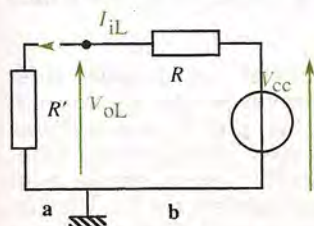


Fig. 8

Lorsque deux opérateurs logiques sont associés en cascade, la circulation du courant peut se faire de la sortie du premier vers l'entrée du second ou de l'entrée du second vers la sortie du premier : dans le premier cas la cascade est qualifiée à *injection de courant*, dans le second elle est qualifiée à *absorption de courant*.

Cascade à injection de courant

Prenons l'exemple de la figure 5 : il comporte deux opérateurs NON-ET. Les entrées du premier sont à l'état bas. La sortie de l'opérateur a est donc au niveau haut ; l'entrée de l'opérateur b, associée à la sortie de l'opérateur a, est traversée par un courant d'intensité I_{iH} . **Le premier opérateur impose alors un courant au second** ; ce dernier se comporte comme une résistance qui serait reliée entre la sortie de l'opérateur a et la masse. Le courant d'intensité I_{iH} provient de l'alimentation de l'opérateur a. La figure 6 est un schéma équivalent de Thévenin de l'ensemble.

Cascade à absorption de courant

Prenons l'exemple de la figure 7 : on y trouve deux opérateurs NON-ET. Les entrées du premier sont à l'état haut. La sortie de l'opérateur a est donc au niveau bas ; l'entrée de l'opérateur b reliée à la sortie de l'opérateur a, est traversée par un courant d'intensité I_{iL} . **Le second opérateur impose alors un courant au premier** ; ce dernier se comporte comme une résistance qui serait reliée entre l'entrée de l'opérateur b et la masse. Le courant d'intensité I_{iL} provient de l'alimentation de l'opérateur b. La figure 8 est un schéma équivalent de Thévenin de l'ensemble.

Dans une cascade à absorption de courant, quand un circuit logique est à l'état bas et qu'il commande un opérateur logique, il est traversé par le courant fourni par l'entrée de l'opérateur commandé.

9 FAMILLE LOGIQUE TTL

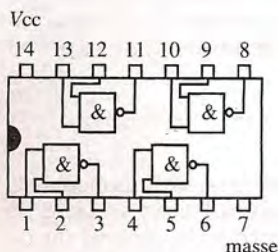


Fig. 1. Opérateurs NON-ET TTL.

1 Description du circuit de base

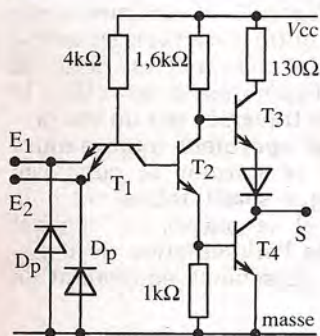


Fig. 2. Circuit de base d'un opérateur NON-ET. Les diodes D_p protègent les entrées contre les tensions négatives.

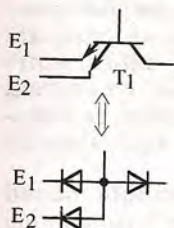


Fig. 3. Modèle équivalent en régime statique.

Dans ce chapitre, nous allons étudier l'une des familles les plus courantes, la famille TTL (Transistor Transistor Logic) : c'est une technologie à transistors bipolaires. C'est la famille la plus employée dans les circuits à intégration à petite échelle (SSI). Son utilisation pour l'intégration à moyenne échelle (MSI) diminue aux dépens de la famille CMOS que nous étudierons dans le chapitre suivant.

Tous les composants de la famille sont, bien sûr, compatibles entre eux. L'élément le plus courant de la famille est l'opérateur NON-ET. La figure 1 représente l'implantation des quatre opérateurs NON-ET dans le circuit standard de référence 7400.

Présentation

La figure 2 représente le circuit de base d'un opérateur NON-ET. Le transistor d'entrée est pourvu de deux émetteurs. Les opérateurs NON-ET pouvant avoir jusqu'à huit entrées, le transistor d'entrée peut avoir jusqu'à huit émetteurs.

Pour une compréhension plus aisée du fonctionnement du circuit, le transistor T_1 est remplacé par un modèle équivalent formé de trois diodes (fig. 3).

Etat des composants du circuit

- Sortie à l'état bas (fig. 4)

Les entrées E_1 et E_2 sont à l'état haut (tensions supérieures à 2,4 V). Elles sont traversées par des courants d'intensité très faible : ce sont les courants traversant les jonctions base-émetteur polarisées en inverse (une dizaine de microampères environ).

A vide, la sortie S a alors un potentiel très faible (inférieur à 0,4 V).

- Sortie à l'état haut (fig. 5)

L'une des entrées est à l'état bas (tension inférieure à 0,8 V); elle est traversée par un courant d'intensité assez élevée (de l'ordre du milliampère) : c'est le courant traversant la jonction base-émetteur polarisée en direct. L'autre entrée peut être à l'état haut (tension supérieure à 2,4 V); elle est alors traversée par un courant d'intensité très faible.

Fig. 4. Sortie à l'état bas.
Éléments conducteurs : D_3 ,
 T_2 , T_4 .
Éléments bloqués : D_1 , D_2 ,
 T_3 , D_4 .

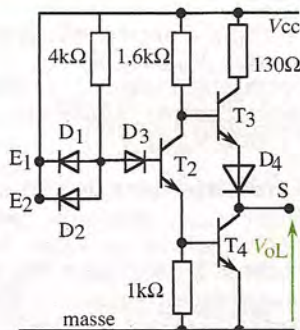
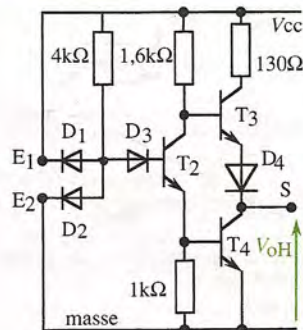


Fig. 5. Sortie à l'état haut.
Éléments conducteurs : D_2 ,
 T_3 , D_4 .
Éléments bloqués : D_1 , D_3 ,
 T_2 , T_4 .



2 Caractéristiques statiques

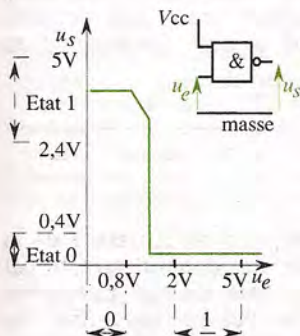


Fig. 6. Caractéristique de transfert à vide.

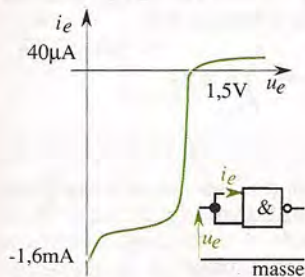


Fig. 7. Caractéristique d'entrée.

Pour les opérateurs TTL la tension d'alimentation V_{cc} doit être comprise entre 4,75 V et 5,25 V. Elle est généralement de 5,0 V.

Les entrées ne doivent jamais être soumises à des tensions supérieures à 5,5 V car il y a risque de claquage de la jonction base-émetteur du transistor d'entrée. Elles ne doivent jamais être soumises à des tensions inférieures à -0,5 V car, en réalité, des diodes de protection sont généralement placées entre les émetteurs du transistor d'entrée et la masse et pour des tensions plus négatives que -0,5 V ces diodes conduisent et sont traversées par un courant d'intensité importante.

Caractéristique de transfert

C'est la courbe donnant les variations de la tension de sortie u_s en fonction de la tension d'entrée u_e (fig. 6) pour une tension d'alimentation V_{cc} déterminée et pour une charge fixée (4 kΩ par exemple). Le constructeur garantit un état logique 1 en sortie, c'est-à-dire une tension minimale de sortie de 2,4 V (V_{oHmin}), si l'opérateur débite un courant d'intensité inférieure à 0,8 mA (I_{oHmax}).

Le constructeur garantit un état logique 0 en sortie, c'est-à-dire une tension maximale de sortie de 0,40 V (V_{oLmax}), si l'opérateur reçoit un courant d'intensité inférieure à 16 mA (I_{oLmax}).

Les valeurs courantes sont : $V_{oH} = 3,6$ V, $V_{oL} = 0,2$ V.

Caractéristique d'entrée

Elle est représentée figure 7.

A un état logique 1 en entrée correspond un

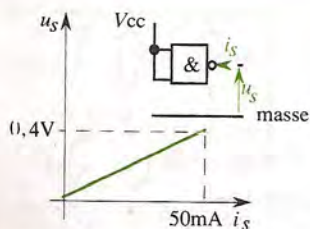


Fig. 8. Caractéristique de sortie à l'état bas.

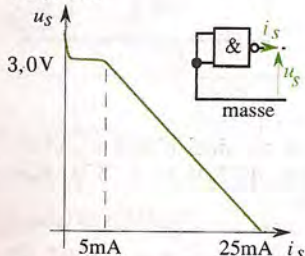


Fig. 9. Caractéristique de sortie à l'état haut.

courant d'entrée d'intensité $i_e \approx 40 \mu\text{A}$ (avec une tension $V_{OH} \geq 2,4 \text{ V}$).

A un état logique 0 en entrée correspond un courant d'entrée d'intensité $i_e \approx -1,6 \text{ mA}$ (avec une tension $V_{OL} \leq 0,4 \text{ V}$).

Caractéristiques de sortie

Deux caractéristiques de sortie sont tracées, l'une pour l'entrée à l'état bas (fig. 8), l'autre pour l'entrée à l'état haut (fig. 9). Pour les circuits TTL, la sortance dynamique étant à peu près égale à la sortance statique, à partir de ces caractéristiques on détermine la sortance de l'opérateur étudié (voir paragraphe 5).

Consommation statique

La consommation moyenne d'un opérateur de base NON-ET est de 2 mA environ (1 mA pour une sortie à l'état haut et 3 mA si elle est à l'état bas). Si l'alimentation est réalisée sous une tension $V_{cc} = 5 \text{ V}$, cela correspond à une consommation de 10 mW.

3 Caractéristiques dynamiques

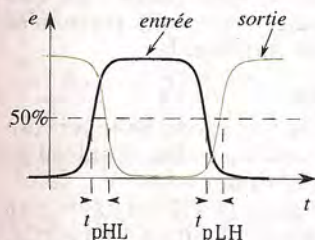


Fig. 10

Temps de propagation

Le temps de propagation dépend de la nature de la charge placée à la sortie de l'opérateur. Il est habituellement défini sur une charge purement capacitive : $C = 15 \text{ pF}$; la valeur moyenne du temps de propagation à la croissance du signal de sortie est $t_{pLH} = 11 \text{ ns}$ (la valeur maximale étant de 22 ns) alors que celle du temps de propagation à la décroissance du signal de sortie est $t_{pHL} = 7 \text{ ns}$ (la valeur maximale étant de 15 ns). Ce qui donne un temps moyen de retard de propagation :

$$t_{pd} = \left(\frac{t_{pLH} + t_{pHL}}{2} \right) = 9 \text{ ns}$$

Consommation dynamique

Les commutations répétées des opérateurs logiques contribuent à une augmentation de la puissance totale consommée. Cependant cette augmentation de puissance est faible par rapport à la consommation statique des circuits TTL et elle est souvent négligeable.

4 Autres caractéristiques

Entrées non utilisées

Toute entrée non connectée d'un circuit TTL se comporte comme une entrée à l'état haut. Il est préférable de raccorder toute entrée non utilisée :

- à l'alimentation par l'intermédiaire d'une résistance supérieure à 1 k Ω dans le cas des opérateurs ET et NON-ET,
- à la masse dans le cas des opérateurs OU et NON-OU (c'est une nécessité dans ce dernier cas).

Immunité au bruit

La marge d'immunité au bruit d'un circuit TTL est égale à :

$$V_{oHmin} - V_{iHmin} = 2,4 - 2 = 0,4 \text{ V au niveau haut}$$

$$\text{et } V_{iLmax} - V_{oLmax} = 0,8 - 0,4 = 0,4 \text{ V au niveau bas.}$$

Tout bruit générant des tensions parasites inférieures à 0,4 V est sans action sur le circuit.

Facteur de mérite

Il est représenté par le produit : consommation statique $\times$ durée de propagation. Il est environ égal à 100 picojoules-secondes, pour la série standard.

5 Autres séries TTL

Les valeurs numériques indiquées précédemment, concernent la série TTL dite normale ou standard. Il existe d'autres séries TTL possédant des caractéristiques légèrement différentes.

	7400	74L00	74H00	74S00
V_{cc} (V)	5 V			
$V_{oH(min)}$ (V)	2,4	2,4	2,4	2,7
$V_{oL(max)}$ (V)	0,4	0,3	0,4	0,5
$V_{iH(min)}$ (V)	2	2	2	2
$V_{iL(max)}$ (V)	0,8	0,8	0,8	0,8
retard (nS)	9	33	6	3
consommation (mW)	10	1	23	23
f_{max} (MHz)	25	3	40	80

Tableau 1

Sortance	7400	74L00	74H00	74S00
haut	10	12	10	25
bas	10	12	3	12

Tableau 2

La sortance est exprimée en nombre de charges TTC standard commandables en régime dynamique

Principales séries

- La série 74L00 est dite à faible consommation (Low power) : les résistances internes sont plus élevées que celles de la série standard, ce qui entraîne une réduction de consommation mais des temps de propagation plus importants.
- La série 74H00 est dite rapide (High speed) : les résistances internes sont plus faibles (et le transistor de sortie est un Darlington), ce qui entraîne une augmentation de consommation mais des temps de propagation réduits.
- La série 74S00 appelée série de Schottky utilise des transistors insaturables et des résistances plus faibles que celles de la série standard. Pour une consommation pratiquement égale à celle de la série 74H00, la série 74S00 est deux fois plus rapide.

Charge unitaire

C'est une charge TTL standard qui permet de déterminer la sortance d'un composant TTL (tableau 2). Elle est définie par :

- un courant d'intensité $I_{iH(max)} = 40 \mu A$ (celui qui entre par une entrée TTL à l'état haut), qu'elle reçoit quand on lui applique une tension de 5 V,
- un courant d'intensité $I_{oL(max)} = 1,6 \text{ mA}$ (celui qui est débité par une sortie TTL à l'état bas), qu'elle fournit quand on lui applique une tension nulle.

10 FAMILLE LOGIQUE CMOS

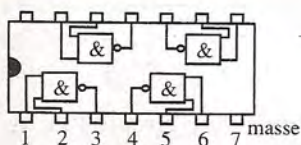


Fig. 1. Opérateur NON-ET CMOS 4011.

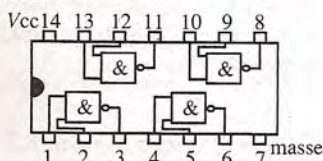


Fig. 2. Opérateur NON-ET CMOS 74C00.

1 Description

Dans ce chapitre, nous allons étudier une autre famille d'opérateurs logiques couramment utilisée, la famille CMOS à transistors à effet de champ MOS. Cette famille est de plus en plus employée dans les circuits à intégration à moyenne échelle (MSI).

Tous les composants de la famille sont compatibles entre eux. La figure 1 représente l'implantation des quatre opérateurs NON-ET dans un circuit de la série 4000 (référence 4011). Certains constructeurs ont créé d'autres séries comme la série 74C00 (fig. 2) dont les composants ont le même brochage que ceux, équivalents, de la série 7400 TTL.

On utilise deux sortes de transistors MOS : l'un à canal N et l'autre à canal P. Leur tension d'alimentation est habituellement désignée par V_{DD} . L'état haut de la sortie correspond à la tension V_{DD} ; son état bas correspond à 0 V.

Inverseur

La figure 3 représente le circuit de base d'un opérateur NON. Les deux transistors sont en série.

- Lorsque l'entrée est à l'état bas ($u_e \approx 0$ V) :
- le transistor à canal N est bloqué car $V_{GS2} \approx 0$: la résistance drain-source R_{DS2} atteint $10^{10} \Omega$;
- $V_{GS1} = V_{GS2} - V_{DD} \approx -V_{DD}$: le transistor à canal P est passant; sa résistance drain-source R_{DS1} est de l'ordre du kilohm.

R_{DS2} et R_{DS1} forment un pont diviseur de tension. La tension de sortie u_s est égale à :

$$u_s = \frac{R_{DS2}}{R_{DS2} + R_{DS1}} V_{DD} \approx \frac{R_{DS2}}{R_{DS2}} V_{DD} \approx V_{DD}$$

- Lorsque l'entrée est à l'état haut ($u_e \approx V_{DD}$) :
- $V_{GS2} \approx V_{DD}$: le transistor à canal N est passant; sa résistance drain-source R_{DS2} est de l'ordre du kilohm;
- $V_{GS1} = V_{GS2} - V_{DD} \approx 0$: le transistor à canal P est bloqué; sa résistance drain-source R_{DS1} est de l'ordre de $10^{10} \Omega$.

R_{DS2} et R_{DS1} forment un pont diviseur de tension. La tension de sortie u_s est égale à :

$$u_s = \frac{R_{DS2}}{R_{DS2} + R_{DS1}} V_{DD} \approx 10^{-7} V_{DD} \approx 0 \text{ V}$$

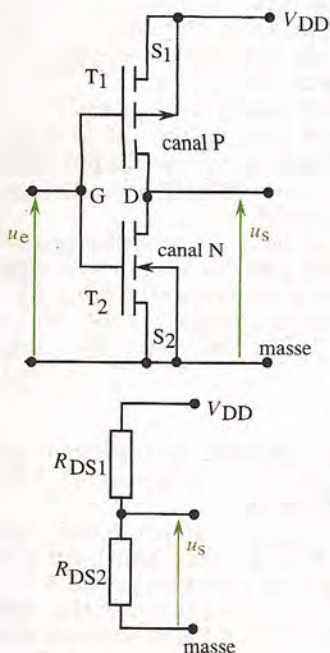


Fig. 3. Inverseur CMOS.

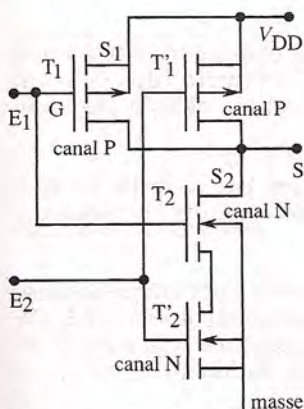


Fig. 4. Opérateur NON-ET CMOS.

E_1	E_2	S
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité d'un opérateur NON-ET

Opérateur NON-ET

La figure 4 représente le circuit de base d'un opérateur NON-ET.

En mettant l'entrée E_1 à l'état bas (la variable logique d'entrée E_1 prend la valeur 0), le transistor T_1 à canal P est passant ($V_{GS1} \approx -V_{DD}$) et le transistor T_2 à canal N est bloqué.

En mettant l'entrée E_1 à l'état haut (la valeur correspondante de la variable logique d'entrée E_1 est 1), le transistor T_1 à canal P est bloqué ($V_{GS1} \approx 0$ V) et le transistor T_2 à canal N est passant.

Nous obtiendrions un résultat identique en raisonnant avec les transistors T'_1 et T'_2 pour l'entrée E_2 . Pour que la variable de sortie S prenne la valeur 0, il faut que les deux variables d'entrée E_1 et E_2 aient la valeur 1; en effet, les deux transistors à canal N sont alors passants et, entre la sortie et la masse, le montage est équivalent à une résistance de faible valeur. Dans tous les autres cas, au moins l'un des transistors à canal P est passant et l'un des transistors à canal N est bloqué : la tension de sortie est à l'état haut.

2 Caractéristiques statiques

V_{DD}	5 V
V_{oHmin}	5 V
V_{iHmin}	3,5 V
V_{oLmax}	0 V
V_{iLmax}	1,5 V

Niveaux de tensions

Tensions

Pour les opérateurs CMOS la tension d'alimentation V_{DD} doit être comprise entre 3 V et 18 V. Si dans une même application des circuits CMOS et TTL sont utilisés, elle est généralement de 5,0 V afin de n'utiliser qu'une seule alimentation.

Le tableau ci-contre résume les niveaux de tension d'un circuit CMOS alimenté sous une tension $V_{DD} = 5,0$ V. Souvent on considère que :

$$\begin{aligned} V_{oHmax} &= V_{DD}, & V_{iHmin} &= 70 \% V_{DD}, \\ V_{oL} &= 0 \text{ V} & \text{et} & & V_{iLmax} &= 30 \% V_{DD} \end{aligned}$$

Caractéristique de transfert

C'est la courbe donnant les variations de la tension de sortie u_s en fonction de la tension d'entrée u_e (fig. 5) pour une tension d'alimentation V_{DD} et une charge fixées. A la valeur logique 1 de la variable de sortie correspond une tension de sortie $u_s \approx V_{DD}$, à la valeur logique 0 correspond une tension de sortie $u_s \approx 0$ V. Le changement d'état de la variable de sortie se produit chaque fois que la tension d'entrée u_e passe par une valeur proche de $V_{DD}/2$.

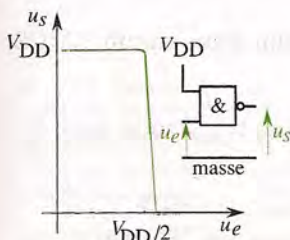


Fig. 5. Caractéristique de transfert à vide.

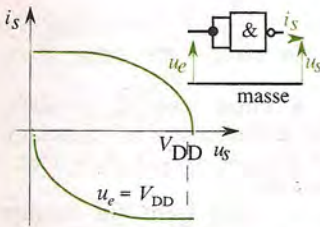


Fig. 6. Caractéristiques de sortie.

Caractéristique d'entrée

La résistance d'entrée d'un opérateur CMOS étant très élevée ($>10^{10} \Omega$), l'intensité du courant d'entrée est très faible et négligée dans la pratique.

Caractéristiques de sortie (fig. 6)

L'intensité du courant fourni par la sortie ne doit pas dépasser 10 mA pour les circuits classiques.

Consommation statique

La consommation moyenne de l'opérateur de base NON-ET CMOS est extrêmement faible : 2,5 nW pour une tension d'alimentation $V_{DD} = 5,0 \text{ V}$ et 10 nW lorsque cette tension est de 10 V.

3 Caractéristiques dynamiques

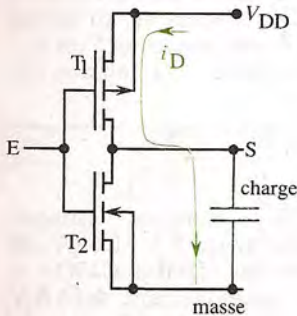


Fig. 7. Consommation dynamique d'un opérateur CMOS sur charge capacitive.

Temps moyen de propagation

C'est la durée que met un signal logique pour « franchir » le circuit. Elle dépend de la tension d'alimentation et de la charge placée en sortie. Elle est souvent définie avec une charge capacitive de 15 pF. Nous obtenons par exemple 35 ns pour une tension d'alimentation de 5 V et 25 ns pour une alimentation sous 10 V : la rapidité de l'opérateur CMOS augmente avec la tension d'alimentation.

Consommation dynamique

En alimentant les entrées du circuit logique par un signal rectangulaire, on observe une augmentation importante de consommation (même à vide) des opérateurs CMOS (fig. 7). Elle est proportionnelle à la fréquence (0,1 mW à 100 kHz et 1 mW à 1 MHz).

Immunité au bruit

La marge d'immunité au bruit d'un circuit CMOS est égale :

- au niveau haut à :

$$V_{oHmin} - V_{iHmin} = V_{DD} - 70 \% V_{DD} = 30 \% V_{DD};$$

- au niveau bas à :

$$V_{iLmax} - V_{oLmax} = 30 \% V_{DD} - 0 = 30 \% V_{DD}.$$

Pour une tension d'alimentation de 5,0 V, on obtient une marge d'immunité au bruit de 1,5 V, alors qu'elle n'est que de 0,4 V en TTL.

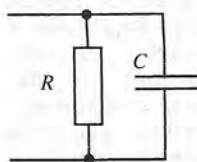


Fig. 8. Schéma équivalent.

Sortance

En régime dynamique, vu d'une entrée, un circuit CMOS se comporte à peu près comme un condensateur de capacité $C = 5 \text{ pF}$ en parallèle (fig. 8) avec une résistance très élevée ($> 10^{10} \Omega$). Cette capacité limite le nombre d'entrées d'un opérateur CMOS qu'une sortie d'opérateur CMOS peut commander : cette sortie voit une capacité équivalente à la somme des capacités des entrées commandées, ce qui augmente la durée de commutation de la sortie. Plus la fréquence d'utilisation est élevée, plus la sortance est petite. Jusqu'à 1 MHz, la sortance des opérateurs CMOS est limitée à 50.

Entrées non utilisées

Toute entrée non utilisée d'un circuit CMOS doit être soumise à une tension fixe (0 V ou V_{DD}) ou reliée à une autre entrée utilisée.

4 Association d'opérateurs de technologies différentes

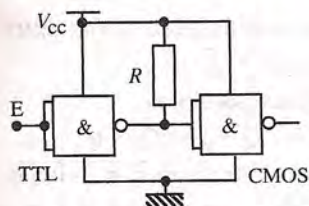


Fig. 9. Opérateur CMOS piloté par un opérateur TTL.

Opérateur CMOS piloté par un opérateur TTL

Nous nous plaçons dans le cas où les deux opérateurs sont alimentés sous la même tension d'alimentation $V_{DD} = 5,0 \text{ V}$. Rappelons que la tension V_{OH} d'un opérateur TTL peut atteindre 3,6 V et descendre à 2,4 V (V_{OHmin}) lorsqu'elle commande plusieurs charges TTL. Or la tension V_{IHmin} d'un circuit CMOS alimenté sous 5,0 V est de 3,5 V. Aussi est-il nécessaire de relever la tension V_{OH} . Le montage de la figure 9 est un exemple de réalisation.

Opérateur TTL piloté par un opérateur CMOS

Rappelons que $I_{IHmax} = 40 \mu\text{A}$ pour un opérateur TTL standard, ce qui est peu : une sortie CMOS à l'état haut peut donc commander plusieurs charges TTL standard. Ce n'est plus vrai lorsque la sortie CMOS est à l'état bas. En effet pour un opérateur TTL standard, $V_{OLmax} = 0,4 \text{ V}$ et $I_{ILmax} = 1,6 \text{ mA}$ (fig. 10). La résistance de sortie R_s d'un CMOS à l'état bas est de l'ordre du kiloohm et est traversée par le courant d'intensité I_{IL} , ce qui donne une tension $R_s I_{IL}$ supérieure à 0,4 V. Il est nécessaire de placer un circuit tampon entre l'opérateur CMOS et l'opérateur TTL capable de supporter le courant provenant de l'opérateur TTL (fig. 10).

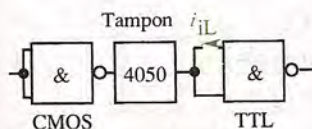


Fig. 10. Opérateur TTL piloté par une porte CMOS par l'intermédiaire d'un circuit tampon (4050 par exemple).

11 BASCULES RS



Fig. 1. Bascule.

Pour certains opérateurs l'état de la sortie dépend non seulement de la combinaison appliquée à l'entrée mais aussi de l'état précédent du circuit : ils sont dits séquentiels et ont un effet « mémoire ». L'état des variables de sortie dépend de la séquence appliquée à l'entrée, donc de l'ordre dans lequel ont été appliquées différentes combinaisons d'entrée. Le circuit de mémorisation le plus répandu est la bascule (fig. 1). Nous allons étudier le principe d'une bascule RS.

1 Bascule RS avec des opérateurs NON-OU

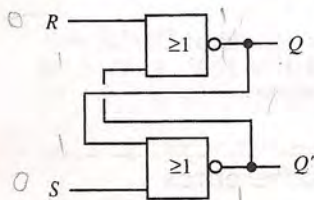


Fig. 2. Bascule RS en opérateurs NON-OU.

Présentation de l'état des sorties

Pour le montage de la figure 2, les états stables correspondent aux résultats des équations logiques suivantes qui donnent les états des sorties représentées par les variables Q et Q' :

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

- Plaçons nous dans le cas où : $R = 0$ et $S = 0$

Les relations précédentes se simplifient alors pour donner :

$$Q = \overline{Q'} \text{ et } Q' = \overline{Q}$$

Ce qui donne deux solutions possibles :

soit : $Q = 0$ et $Q' = 1$; soit : $Q = 1$ et $Q' = 0$

- Choisissons une autre combinaison des valeurs de R et de S : $R = 0$ et $S = 1$

Les relations initiales s'écrivent après simplification :

$$Q = \overline{Q'} \text{ et } Q' = 0$$

Une seule solution est possible :

$$Q = 1 \text{ et } Q' = 0$$

- Choisissons une troisième combinaison des valeurs de R et de S : $R = 1$ et $S = 0$

Simplifions les relations initiales; nous obtenons :

$$Q = 0 \text{ et } Q' = \overline{Q}$$

Une seule solution est possible :

$$Q = 0 \text{ et } Q' = 1$$

- Choisissons la dernière combinaison possible des valeurs de R et de S : $R = 1$ et $S = 1$.

Après simplification des relations initiales, nous obtenons :

$$Q = 0 \text{ et } Q' = 0$$

Conclusions

- A l'une des combinaisons d'entrée peuvent correspondre deux combinaisons en sortie (à $R = 0$ et $S = 0$ correspondent deux combinaisons de sortie $Q = 0$ et $Q' = 1$ ou $Q = 1$ et $Q' = 0$).
- Sauf pour la combinaison d'entrée $R = 1$ et $S = 1$, les sorties Q et Q' sont complémentaires.

Fonction mémoire

L'état initial, ou état de repos, correspond à la combinaison : $R = 0, S = 0$

Nous avons vu que dans ce cas, l'état des entrées ne suffit pas à déterminer l'état des sorties :

$$Q = 0 \text{ et } Q' = 1 \text{ ou } Q = 1 \text{ et } Q' = 0$$

Modifions uniquement la valeur de R deux fois de suite :

pour $R = 1, S = 0$: $Q = 0$ et $Q' = 1$;

puis pour $R = 0, S = 0$: les variables de sortie Q et Q' conservent les valeurs qu'elles avaient l'état précédent car elles correspondent à l'une des solutions possibles des équations logiques caractérisant les états stables de la bascule. Il y a « mémorisation » de l'état antérieur : $Q = 0$ et $Q' = 1$.

Reprenons l'état initial et modifions successivement uniquement la valeur de S . Nous obtenons :

- pour $R = 0, S = 1$: $Q = 1$ et $Q' = 0$;

- puis pour $R = 0, S = 0$: les variables de sortie Q et Q' conservent les valeurs qu'elles avaient l'état précédent car elles correspondent à l'une des solutions possibles des équations logiques caractérisant les états stables de la bascule. Dans ce cas aussi il y a « mémorisation » de l'état antérieur. Les variables de sorties sont : $Q = 1$ et $Q' = 0$.

Conclusions

La première séquence a permis de donner aux variables de sortie Q et Q' respectivement les valeurs 0 et 1 alors que la seconde séquence a conduit respectivement à 1 et 0.

Nous venons de décrire le fonctionnement d'une mémoire élémentaire ou bascule. En pratique, la solution correspondant à $R = S = 1$ n'est jamais utilisée.

R	S	Q	Q'
0	0	inchangé	
0	1	1	0
1	0	0	1
1	1	0	0 inutilisé

Fig. 3. Table de vérité d'une bascule RS NON-OU.

RS	00	01	11	10
Q_n	0	1	0	0
	1	1	0	0

Fig. 4.

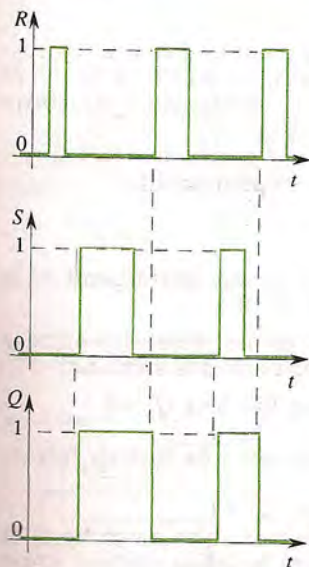


Table de vérité

Cette table (fig. 3) donne les combinaisons de sortie en fonction des combinaisons d'entrée. Elle montre que l'entrée R permet de porter Q à 0, l'entrée S permet de porter Q à 1. Si l'on interdit la combinaison $R = 1, S = 1$, alors les sorties Q et Q' sont complémentaires : les variables de sortie seront notées Q et $\bar{Q}$.

Si nous appelons Q_n l'état de la sortie Q avant la modification d'une des variables d'entrée et Q_{n+1} son état après cette modification, la table de Karnaugh associée à la sortie Q qui permet la détermination de Q_{n+1} est donnée figure 4.

A partir de cette table, nous en déduisons une équation logique associée à la bascule RS :

$$Q_{n+1} = \bar{R} \cdot S + \bar{R} \cdot Q_n = \bar{R} \cdot (S + Q_n)$$

Chronogrammes

Ils sont donnés figure 5.

Fig. 5. Chronogrammes. A $t = 0$, Q est supposé être à 0. Ils représentent l'allure des signaux d'entrée et de sortie de la bascule en négligeant les temps de propagation.

2 Bascule RS en opérateurs NON-ET

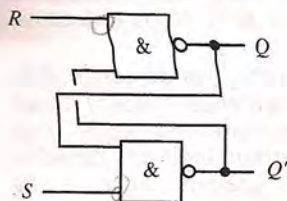


Fig. 6. Bascule RS en opérateurs NON-ET.

Présentation

La figure 6 illustre le principe d'une bascule en opérateurs NON-ET. Les états stables du circuit correspondent aux deux résultats des équations logiques suivantes :

$$Q = \bar{R} \cdot Q' = \bar{R} + \bar{Q}' \quad \text{et} \quad Q' = \bar{S} \cdot Q = \bar{S} + \bar{Q}$$

- $R = 0$ et $S = 0$ implique : $Q = 1$ et $Q' = 1$
- $R = 0$ et $S = 1$ implique : $Q = 1$ et $Q' = 0$
- $R = 1$ et $S = 0$ implique : $Q = 0$ et $Q' = 1$
- $R = 1$ et $S = 1$ implique : $Q = \bar{Q}'$ et $Q' = \bar{Q}$

Ce qui donne deux solutions possibles :

soit : $Q = 0$ et $Q' = 1$; soit : $Q = 1$ et $Q' = 0$

Comme dans le cas de la bascule RS NON-OU :

- à une combinaison des variables d'entrée peuvent correspondre deux combinaisons des variables de sortie,
- à l'exception de la combinaison d'entrée $R = 0$ et $S = 0$, les sorties Q et Q' sont complémentaires.

R	S	Q	Q'
0	0	1	1
0	1	1	0
1	0	0	1
1	1	inchangé	

Fig. 7. Table de vérité d'une bascule RS NON-ET.

RS	00	01	11	10
Q _n	1	1	0	0
1	1	1	1	0

Fig. 8 Table de Karnaugh

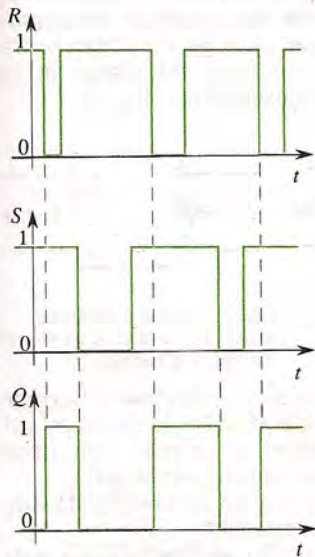


Fig. 9. Chronogrammes. A $t = 0$, Q est supposé être à 0. Ils représentent l'allure des signaux d'entrée et de sortie de la bascule en négligeant les temps de propagation.

Fonction mémoire

Dans le cas de cette bascule, l'état de repos correspond à la combinaison : $R = 1, S = 1$. Nous avons vu que l'état des sorties n'est pas déterminé par les entrées ($Q = 0$ et $Q' = 1$ ou $Q = 1$ et $Q' = 0$), il dépend de l'état antérieur de la bascule.

• Première séquence

- pour $R = 0, S = 1$: $Q = 1$ et $Q' = 0$;
- puis pour $R = 1, S = 1$: les variables de sortie Q et Q' conservent les valeurs qu'elles avaient l'état précédent car elles correspondent à l'une des solutions possibles des équations logiques caractérisant les états stables de la bascule. Les variables de sortie ont donc les valeurs : $Q = 1$ et $Q' = 0$

• Seconde séquence

Reprenons l'état initial et modifions successivement uniquement l'état de S. Nous obtenons :

- pour $R = 1, S = 0$: $Q = 0$ et $Q' = 1$;
- puis pour $R = 1, S = 1$: les variables de sortie Q et Q' conservent les valeurs qu'elles avaient l'état précédent (solution possible des équations logiques caractérisant les états stables de la bascule). Les variables de sorties sont donc : $Q = 0$ et $Q' = 1$

Conclusions

Nous retrouvons le fonctionnement d'une mémoire élémentaire. En pratique, la solution $S = R = 0$ n'est jamais utilisée et, dans cette mesure, on peut dire que $Q' = \bar{Q}$.

Table de vérité

Elle est donnée par la figure 7. Si nous appelons Q_n l'état de la sortie Q avant la modification d'une des variables d'entrée et Q_{n+1} son état après cette modification, la table de Karnaugh associée à la sortie Q qui permet la détermination de Q_{n+1} est donnée figure 8.

A partir de cette table, nous déduisons une équation logique associée à la bascule RS :

$$Q_{n+1} = \bar{R} + S \cdot Q_n$$

Chronogrammes

Ils sont donnés figure 9.

12 BASCULES SYNCHRONES

Les bascules semblables à celles décrites au chapitre 11 sont dites asynchrones (ou bascules non synchronisées). Pour ce type de bascule, le changement d'état de la sortie se produit au moment où la combinaison des valeurs d'entrée est changée (en négligeant les temps de propagation).

On fabrique d'autres bascules, dites synchrones, pour lesquelles le changement d'état de la sortie, qui correspond à une nouvelle combinaison des valeurs d'entrée, ne peut s'effectuer que lorsqu'un signal de contrôle appelé *signal d'horloge* prend lui-même une valeur donnée.

1 Structure d'une bascule synchrone

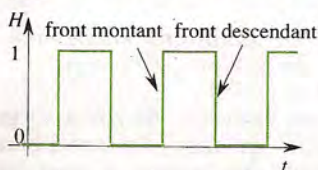


Fig. 1. Signal d'horloge.

Toute bascule synchrone dispose d'une entrée d'horloge. Généralement la variable logique associée à cette entrée est représentée par la lettre H . A cette entrée est appliquée une tension rectangulaire périodique (0, E). Le passage de l'état bas à l'état haut est le front montant du signal et le passage inverse, le front descendant (fig. 1).

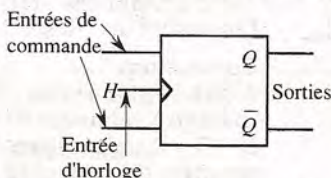


Fig. 2. Bascule synchrone active sur le front montant du signal d'horloge.

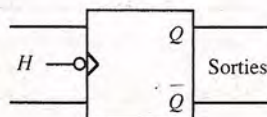


Fig. 3. Bascule synchrone active sur le front descendant du signal d'horloge.

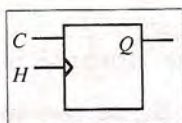
Sur le symbole d'une bascule, si l'entrée d'horloge aboutit à un petit triangle (fig. 2), la bascule est active sur le front montant du signal : ce front donne l'ordre de basculement du montage.

Si en plus de ce triangle il y a un cercle (fig. 3), elle est active sur le front descendant.

Les états des entrées déterminent l'état de la sortie de la bascule mais ce n'est qu'en présence du front actif du signal d'horloge que cet état est obtenu.

2 Exigences de synchronisation

Pour que le fonctionnement d'une bascule synchronisée soit efficace, deux conditions sont nécessaires : l'une concerne la durée de stabilisation et l'autre la durée de maintien.



Signal appliqué à l'entrée de commande C :

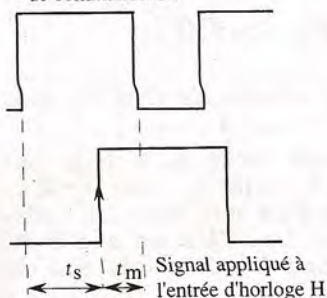


Fig. 4. Durées de stabilisation et de maintien.

Durée de stabilisation

Prenons l'exemple de la figure 4 : l'entrée de commande C permet de donner à la variable de sortie Q la valeur 1 : $C = 1$ impose $Q = 1$ après un front actif du signal d'horloge. Toutefois, pour qu'il en soit ainsi, il est nécessaire que C ait la valeur 1 depuis une certaine durée t_s avant l'application du signal d'horloge : t_s est la durée de stabilisation permettant un basculement sûr de la sortie (avec stabilisation de la sortie à l'état haut dans notre exemple).

La durée de stabilisation t_s est égale à l'intervalle de temps entre l'instant où l'entrée de commande est mise au niveau qui imposera l'état de la sortie qu'elle commande, et l'instant correspondant au front actif du signal d'horloge qui suit (pour un bon fonctionnement il faut que la durée t_s soit supérieure à une valeur minimale t_{sm}).

Généralement la durée minimale de stabilisation est voisine de quelques dizaines de nanosecondes.

Durée de maintien

Reprenons l'exemple précédent. Si le retour de l'entrée de commande à l'état 0 intervient trop tôt après l'application du front actif du signal d'horloge, la variable de sortie peut reprendre la valeur 0. Pour éviter ce mauvais fonctionnement, l'état 1 de l'entrée de commande doit être maintenu suffisamment longtemps : la valeur limite de la durée d'application est appelée durée minimale de maintien t_{mm} .

La durée de maintien t_m est égale à l'intervalle de temps séparant l'instant d'application du front actif du signal d'horloge et l'instant, qui suit ce front actif, où la tension d'entrée change d'état à nouveau.

La durée minimale de maintien est proche de quelques nanosecondes.

3 Exemple : bascule RS synchrone ou RSH

	R	S	Q_{n+1}
$\overline{R}$	0	0	Q_n
$\overline{H}$	0	1	0
$\overline{S}$	1	0	1
	1	1	inutilisé

Fig. 5. Bascule RSH et table de vérité associée.

La figure 5 donne la représentation symbolique d'une bascule RSH ainsi que la table de vérité associée.

Avec des opérateurs NON-ET, pour obtenir une bascule dont la table de vérité est celle de la figure 5, il est nécessaire d'utiliser quatre opérateurs.

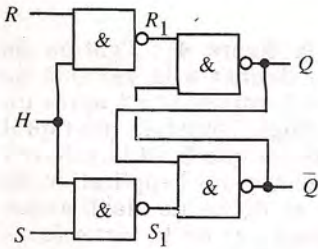


Fig. 6. Bascule RSH à opérateurs NON-ET.

Bascule RSH à opérateurs NON-ET

Nous travaillons à partir de la structure représentée sur la figure 6.

Les valeurs des variables R_1 et S_1 , correspondant aux sorties intermédiaires, sont données par le résultat des opérations :

$$R_1 = \overline{R} \cdot \overline{H} \text{ et } S_1 = \overline{S} \cdot \overline{H}$$

- Faisons $H = 0$:

Quelles que soient les valeurs de R et S , nous avons : $R_1 = 1$ et $S_1 = 1$

Pour modifier les valeurs de Q et de $\overline{Q}$ il faut modifier les valeurs de R_1 ou de S_1 . Avec $H = 0$, la modification de R ou de S ne sera donc pas transmise et la bascule reste dans l'état où elle était. C'est l'état qui correspond à la mémorisation de l'état précédent : $Q_{n+1} = Q_n$.

- Faisons $H = 1$:

Nous avons : $R_1 = \overline{R}$ et $S_1 = \overline{S}$

La modification de R ou de S peut modifier les valeurs des variables de sorties Q et $\overline{Q}$.

La bascule RSH se comporte alors comme une bascule RS.

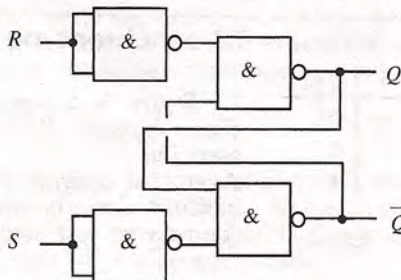
Les entrées S et R n'étant actives qu'en présence d'un signal d'horloge, la bascule est bien synchrone.

Table de vérité

Les sorties étant complémentaires, la table de vérité n'indique, pour la sortie, que les valeurs prises par Q (fig. 6). De plus, pour différencier les valeurs de Q avant et après le front actif du signal d'horloge, on note Q_n la valeur de Q avant le front actif et Q_{n+1} celle qui suit ce front.

L'équation logique associée à une bascule RSH est :

$$Q_{n+1} = R \cdot H + \overline{S} \cdot \overline{H} \cdot Q_n$$



R	S	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	1: inutilisé

Fig. 7. Bascule RS à opérateurs NON-ET : $Q_{n+1} = R + \overline{S} \cdot Q_n$.

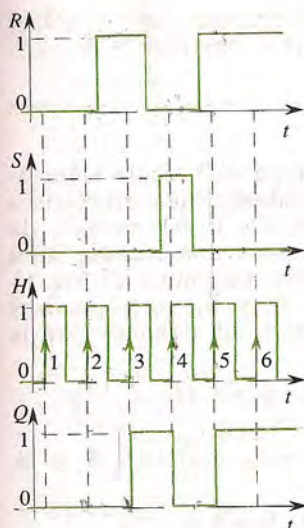


Fig. 8. Exemple de chronogrammes obtenus avec une bascule RSH.

Remarque : Prenons l'exemple de la bascule RS de la figure 7. A partir de la table de vérité qui lui est associée, on en déduit l'équation logique de Q_{n+1} :

$$Q_{n+1} = R + \bar{S} \cdot Q_n$$

Pour retrouver l'équation logique associée à la bascule RSH, il suffit de remplacer R par $R \cdot H$ et S par $S \cdot H$.

Chronogrammes

La figure 8 est un exemple ne tenant pas compte des durées de propagation.

Analyse des chronogrammes :

- Initialement les variables R, S et Q sont supposées être égales à zéro.
- La bascule est active sur le front montant du signal d'horloge. R et S ne prennent jamais simultanément les valeurs 1.
- Aux instants correspondant aux deux premiers fronts montants du signal d'horloge, R et S sont égales à 0; Q reste inchangée : avant chaque front montant, $Q = Q_n = 0$, après $Q = Q_{n+1} = 0$.
- Lorsque R prend la valeur 1, Q conserve la valeur 0 jusqu'à l'action du front montant n° 3 qui donne à Q la valeur 1.
- La mise à 0 de la variable R ne modifie pas la valeur de la variable de sortie Q : $Q = 1$.
- Lorsque S est mise 1, Q conserve la valeur 1 jusqu'à l'action du front montant n° 4 qui remet Q à 0.
- Le retour à 0 de la variable S ne modifie pas la valeur de la variable de sortie Q : $Q = 0$.
- Lorsque R est remise à 1, la variable Q ne reprend la valeur 1 que sous l'action du front montant d'horloge qui suit (le n° 5 dans notre exemple).
- La variable S restant toujours à 0, les fronts n° 6, n° 7, ..., sont sans action sur la tension de sortie de la bascule disponible à la borne Q.

Les variables d'entrée R et S sont les variables de commande synchrones; elles ont un effet sur les variables de sortie Q et $\bar{Q}$ uniquement en présence d'un front montant du signal d'horloge : ce signal déclenche le changement d'état des sorties dicté par les valeurs des variables d'entrée.

13 TROIS BASCULES SYNCHRONES

Les principales bascules synchrones sont appelées bascules RSH, bascules DH et bascules JKH.

1 Bascules RSH ou RS synchrones

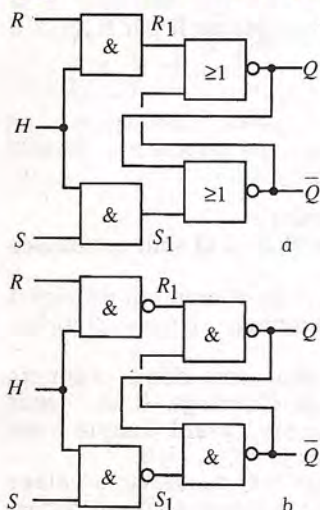


Fig. 1 : Bascule RSH à opérateurs NON-OU/ET et bascule RSH à opérateurs NON-ET.

Nous avons présenté un type de bascule élémentaire RSH au chapitre précédent : elle était réalisée avec des opérateurs NON-ET. Il est possible de réaliser une bascule RSH avec une bascule RS à opérateurs NON-OU et deux opérateurs ET (fig. 1). Les valeurs des variables R_1 et S_1 , correspondant aux sorties intermédiaires, sont données par le résultat des opérations :

$$R_1 = R.H \quad \text{et} \quad S_1 = S.H$$

- Faisons $H = 0$.

Quelles que soient les valeurs de R et S , R_1 et S_1 ont pour valeur :

$$R_1 = 0 \quad \text{et} \quad S_1 = 0$$

Or pour modifier les valeurs de Q et $\bar{Q}$ il faut modifier les valeurs de R_1 ou de S_1 .

Avec $H = 0$, la modification de R ou de S ne sera donc pas transmise et la bascule reste dans l'état où elle était : c'est le mode bloqué.

- Faisons $H = 1$.

Nous avons : $R_1 = R$ et $S_1 = S$

La modification de R ou de S peut agir et modifier les valeurs des variables de sortie Q et $\bar{Q}$: c'est un mode de fonctionnement asynchrone, la bascule RSH équivaut alors à une bascule RS.

C'est sur les fronts montants ou descendants du signal d'horloge que les valeurs des variables de sortie peuvent éventuellement être modifiées.

H	R	S	Q_n	Q_{n+1}	
0	x	x	0 1	$Q_{n+1} = Q_n$	bascule bloquée
1	1	0	0 1	$Q_{n+1} = 1$	fonctionnement asynchrone
	0	1	0 1	$Q_{n+1} = 0$	
↑ ou ↓	1	0	0 1	$Q_{n+1} = 1$	fonctionnement synchrone
	0	1	0 1	$Q_{n+1} = 0$	

Fonctionnement d'une bascule RSH à opérateurs NON-ET (analogue à la figure 1. b). C'est le mode de fonctionnement le plus couramment donné.

2 Bascules D

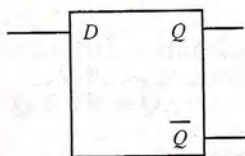


Fig. 2 : Symbole d'une bascule D.

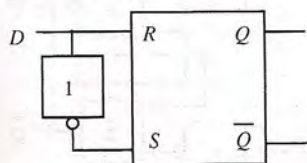


Fig. 3 : Bascule D élémentaire.

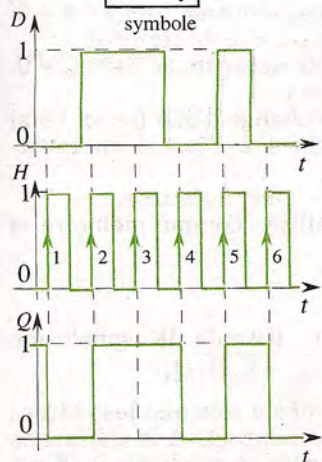
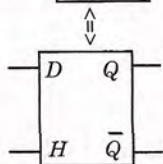
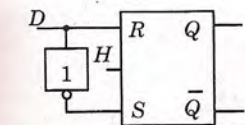


Fig. 4 : Exemple de chronogrammes obtenus avec une bascule DH.

Bascule D élémentaire

La figure 2 est la représentation symbolique de la bascule D.

Elle est réalisée à partir d'une bascule RS à opérateurs NON-ET (voir fig. 7, chap. 12) à laquelle un inverseur a été ajouté en entrée (fig. 3). De ce fait, $R = \bar{S}$ et les combinaisons $R = S = 0$ et $R = S = 1$ (combinaison interdite de la bascule RSH) n'existent pas.

On appelle D l'unique variable d'entrée. La table de vérité est :

D	Q_{n+1}
0	0
1	1

L'équation logique associée à la bascule D est :

$$Q_{n+1} = D$$

Bascule «D-latch» (latch : verrou)

C'est une bascule D synchrone (fig. 4) dont l'étude peut être conduite comme celle de la bascule RSH. L'équation logique associée à la bascule «D-latch» est :

$$Q_{n+1} = D \cdot H + \bar{H} \cdot Q$$

La bascule D est très utilisée dans les compteurs. On rencontre les bascules D à déclenchement sur front d'horloge comme élément de mémoire tampon dans les systèmes électroniques (affichage par exemple).

H	D	Q_n	Q_{n+1}	
0	x	0 1	$Q_{n+1} = Q_n$	bascule bloquée
1	1	0 1	$Q_{n+1} = D$	fonctionnement asynchrone
	0	0 1		
 ou 	1	0 1	1 0	fonctionnement synchrone
	0	0 1		

Modes de fonctionnement d'une bascule DH.

3 Bascules JK

Basculer JK élémentaire

Sur la figure 5, nous avons représenté la structure d'une bascule JK élémentaire. Cette bascule dérive d'une bascule RS dans laquelle $R = J \cdot \bar{Q}$ et $S = \bar{K} \cdot Q$.

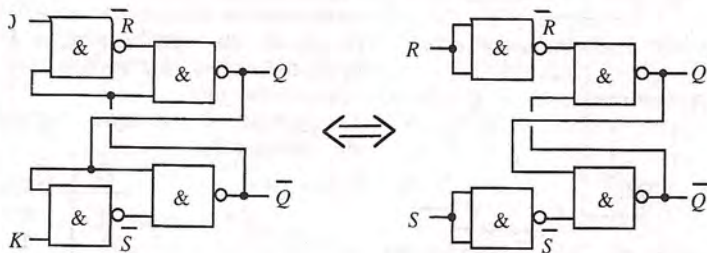


Fig. 5 : Basculer JK élémentaire : c'est une bascule RS dans laquelle $R = J \cdot \bar{Q}$ et $S = \bar{K} \cdot Q$.

Sa table de vérité est :

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

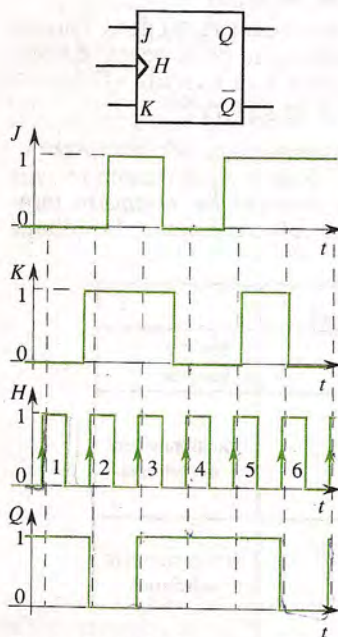


Fig. 6 : Exemple de chronogrammes obtenus avec une bascule JKH.

Son équation logique s'écrit : $Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n$. Ainsi, lorsque les variables d'entrée sont $J = K = 0$, la sortie reste dans l'état où elle se trouvait.

Si $J = 0$ et $K = 1$, la variable de sortie devient $Q = 0$.

Si $J = 1$ et $K = 0$, alors $Q = 1$.

Si $J = 1$ et $K = 1$, la sortie change d'état (selon l'état initial elle passe de l'état 1 à l'état 0 ou inversement).

Cette combinaison $J = K = 1$ est autorisée.

La bascule JK est très utilisée comme mémoire et dans les compteurs.

Basculer JKH

L'équation logique d'une bascule JK synchrone est :

$$Q_{n+1} = J \cdot H \cdot \bar{Q}_n + \bar{K} \cdot H \cdot Q_n$$

Les entrées J et K ont le même rôle que les entrées R et S d'une bascule RS synchrone. À la différence de ce dernier type de bascule, la condition $J = K = 1$ ne donne pas une condition ambiguë sur l'état des variables Q et $\bar{Q}$ (fig. 6).

Modes de fonctionnement d'une bascule JKH.

H	J	K	Q_{n+1}	
0	x	x	$Q_{n+1} = Q_n$	bascule bloquée
 ou 	0	0	$Q_{n+1} = Q_n$	fonctionnement synchrone
	1	0	$Q_{n+1} = 1$	
	0	1	$Q_{n+1} = 0$	
	1	1	$Q_{n+1} = \overline{Q_n}$	

Cas particulier des bascules T

La bascule T est une bascule à une seule entrée. Elle peut être réalisée avec une bascule JK en imposant (fig. 7) : $J = K$

Cette bascule est à entrée unique T.

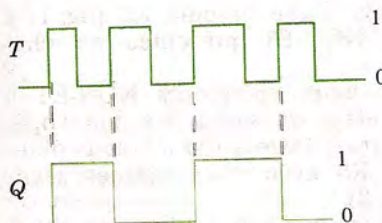
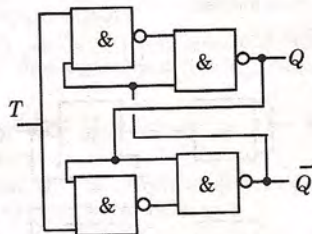


Fig. 7 : Bascule T

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0



Si l'on applique un signal rectangulaire à cette entrée T, la sortie change d'état pour chaque front montant d'entrée.

Son équation logique s'écrit :

$$Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n = T \oplus Q_n$$

Remarque :

$\oplus$ est le symbole du OU EXCLUSIF

- La bascule T synchrone possède une entrée T et une entrée d'horloge H (fig. 8).

Son équation logique s'écrit :

$$Q_{n+1} = (T \cdot H) \cdot \overline{Q_n} + \overline{T \cdot H} \cdot Q_n = (T \cdot H) \oplus Q_n$$

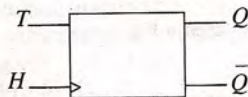


Fig. 8. Bascule T synchrone.

1 Initialisation des bascules

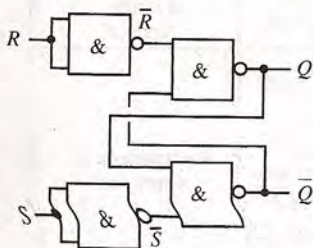


Fig. 1. Bascule RS à opérateurs NON-ET.



Fig. 2. Opérateurs NON-ET à trois entrées.

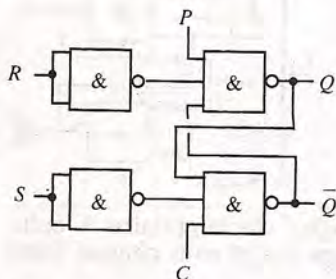


Fig. 3. Bascule RS à opérateurs NON-ET et entrées de prépositionnement.

Les lettres P et C proviennent des mots « Preset » et « Clear » (effacer en français).

Laisser en permanence les entrées P et C à l'état 1, revient, dans le cas particulier de cette bascule, à un fonctionnement en bascule RS sans entrée de prépositionnement.

Ce type de bascule est à la base des compteurs asynchrones.

Entrées asynchrones

Dans le chapitre précédent, nous avons présenté quelques exemples de bascules synchrones : l'état des variables de sortie de ces bascules est synchronisé par le signal d'horloge. Leurs entrées R, S, J, K, D et T sont appelées entrées synchrones.

En général, les bascules synchrones possèdent deux entrées supplémentaires agissant indépendamment des entrées synchrones et du signal d'horloge : elles sont dites asynchrones. Ces entrées sont prioritaires car elles imposent aux variables de sortie Q et $\bar{Q}$ des valeurs indépendantes de celles données aux variables d'entrée synchrones et indépendantes du signal d'horloge : ces entrées sont dites de forçage ou de prépositionnement.

Initialisation d'une bascule RS

Reprenons l'exemple d'une bascule RS (fig. 1) à quatre opérateurs NON-ET présentée au chapitre 12.

En remplaçant les deux opérateurs NON-ET à deux entrées de l'étage de sortie par deux opérateurs NON-ET à trois entrées (fig. 2), nous obtenons une bascule RS avec deux entrées asynchrones P et C (fig. 3).

L'équation logique associée à la bascule RS à opérateurs NON-ET démunie d'entrée asynchrone est :

$$Q_{n+1} = R + \bar{S} \cdot Q_n \quad (1)$$

L'équation logique associée à cette bascule RS avec entrées asynchrones est :

$$Q_{n+1} = R + \bar{P} + C \cdot \bar{S} \cdot Q_n \quad (2)$$

Nous résumons ci-après les différentes valeurs prises par Q_{n+1} suivant les combinaisons possibles des variables P et C.

$$- P = 0 \text{ et } C = 0 \Rightarrow Q_{n+1} = R + 1 = 1 \quad (a)$$

$$- P = 0 \text{ et } C = 1 \Rightarrow Q_{n+1} = \underbrace{R + 1}_{1} + \bar{S} \cdot Q_n = 1 \quad (b)$$

$$- P = 1 \text{ et } C = 0 \Rightarrow Q_{n+1} = R \quad (c)$$

$$- P = 1 \text{ et } C = 1 \Rightarrow Q_{n+1} = R + \bar{S} \cdot Q_n \quad (d)$$

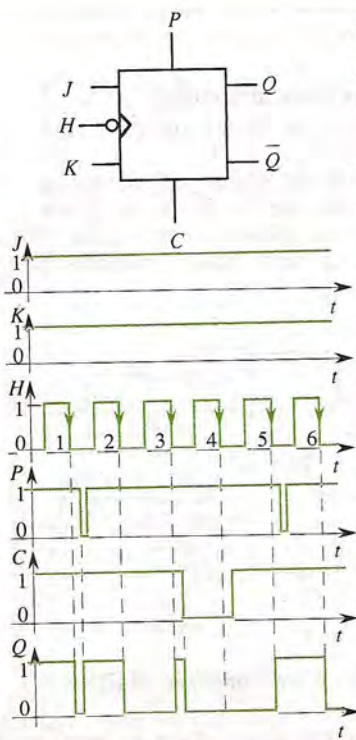


Fig. 4. Exemple de chronogrammes obtenus avec une bascule JKH un peu plus complexe que celle de la figure 3.

• Mise à 0 : exemple d'application.

Nous souhaitons forcer la sortie Q de la bascule RS à zéro alors que les valeurs des variables des entrées sont :

$$P = 1, C = 1, R = 0 \text{ et } S = 0$$

En remplaçant R et S par leurs valeurs dans la relation (d), nous obtenons :

$$Q_{n+1} = Q_n$$

Nous ne connaissons donc pas la valeur de la variable Q (la sortie Q est à l'état 0 ou 1). Portons l'entrée C à l'état 0 : la relation (c) nous indique que la variable Q prend la valeur de R qui dans le cas présent est égale à 0.

Portons à nouveau l'entrée à C à l'état 1 : $Q_{n+1} = Q_n$. Avant la mise à l'état 1 de C, $Q = 0$; après la mise à l'état 1 de C, la variable Q conserve la valeur 0.

L'entrée C est une entrée de mise à zéro de la sortie Q de la bascule par passage transitoire par la valeur zéro.

• Mise à 1 : exemple d'application.

Nous souhaitons forcer la sortie Q de la bascule RS à 1 alors que les valeurs des variables des entrées sont :

$$P = 1, C = 1, R = 0 \text{ et } S = 0$$

La sortie Q est à l'état 0 ou 1.

Portons l'entrée P à l'état 0 : la relation (b) nous indique que la variable Q prend la valeur 1.

Portons à nouveau la variable P à l'état 1 : $Q_{n+1} = Q_n$; or avant la mise à l'état 1 de P, $Q = 1$, après la mise à l'état 1 de P, la variable Q conserve la valeur 1.

L'entrée P est une entrée de mise à un de la sortie Q de la bascule par passage transitoire par la valeur zéro.

Initialisation d'une bascule JK synchrone

A titre d'exemple, nous donnons sur la figure 4 des chronogrammes d'une bascule JKH en circuits intégrés à entrées de prépositionnement actives à l'état bas (il en existe qui sont actives à l'état haut).

Exemple de fonctionnement :

$P = 0, C = 0$: combinaison inutilisée.

$P = 0, C = 1$: $Q = 1$

$P = 1, C = 0$: $Q = 0$

$P = 1, C = 1$: fonctionnement synchrone.

2 Déclenchement des bascules

Déclenchement sur un front d'horloge

Le front actif peut être le front montant ou le front descendant.

La vitesse de variation du signal d'horloge peut être sans influence (on parle alors de déclenchement par niveau) ou déterminante (elle doit alors être inférieure à une limite donnée par le constructeur).

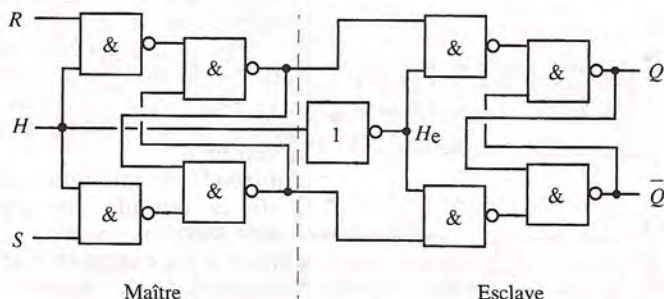


Fig. 5. Bascule RSH maître-esclave.

Déclenchement à partir d'une bascule maître-esclave

Une bascule maître-esclave est formée par un ensemble de deux bascules (fig. 5) commutant pour des valeurs différentes du signal d'horloge, ce qui correspond à des instants différents des fronts d'horloge (fig. 6). La première bascule est « le maître », la seconde est « l'esclave ». Maître et esclave fonctionnent alternativement. Dans l'action du signal d'horloge, nous pouvons distinguer quatre intervalles.

- *Intervalle a* : il y a séparation du « maître » de « l'esclave ».
 - *Intervalle b* : les informations d'entrée agissent sur la bascule « maître ».
 - *Intervalle c* : les entrées de la bascule « maître » sont verrouillées.
 - *Intervalle d* : la bascule « esclave » recopie la sortie de la bascule « maître ». Ainsi, à partir de l'instant t_4 aux valeurs $R = 1$ et $S = 0$ des entrées, par exemple, correspond bien la valeur $Q = 1$ de la sortie.
- Le fonctionnement est très sûr car la vitesse de variation du signal d'horloge est sans importance sur leur fonctionnement.

3 Synchronisation des bascules

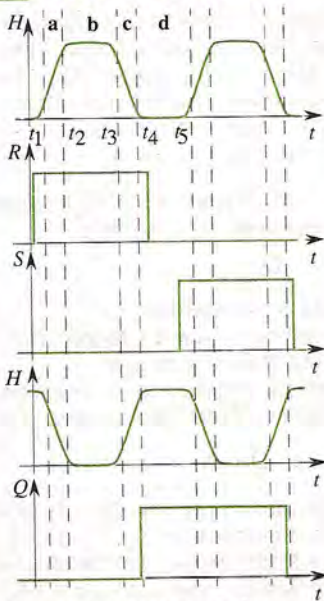


Fig. 6. Chronogrammes d'une bascule RSH maître-esclave active sur un front montant d'horloge : le premier front montant du signal H active la bascule maître; à ce front montant correspond un front descendant du signal d'horloge appliqué à la bascule esclave; la bascule esclave est activée lors du front descendant du signal H qui correspond à un front montant du signal He.

De manière générale, il faut retenir que les bascules en technologie CMOS réagissent moins vite aux signaux d'entrées que les bascules en technologie TTL.

La fréquence maximale du signal d'horloge doit être beaucoup plus petite pour les bascules CMOS que pour les bascules TTL.

Nous avons vu, dans le chapitre 12, que pour obtenir un fonctionnement adéquat d'une bascule synchronisée il faut :

- qu'un intervalle de temps suffisant sépare l'instant d'application du front actif d'horloge et l'instant où la tension d'entrée change d'état à nouveau : c'est la durée minimale de maintien;
- qu'un intervalle de temps suffisant sépare l'instant où l'entrée de commande prend son niveau définitif et l'instant correspondant à l'application du front actif du signal d'horloge : c'est la durée minimale de stabilisation.

D'autres paramètres entrent en jeu pour la synchronisation des bascules : les retards de propagation, la fréquence de synchronisation, les durées à l'état 1 et à l'état 0 du signal d'horloge, ainsi que les durées de transition de ce signal.

Retards de propagation

Chaque fois qu'un signal de commande est appliqué à une entrée de bascule, le changement n'apparaît à la sortie qu'avec un certain retard représenté dans les notices des fabricants par t_{pd} . Ces retards de propagation affectent les réponses à toutes les entrées qu'elles soient synchrones ou asynchrones.

Fréquence de synchronisation

Cette fréquence est en fait la fréquence maximale que peut avoir le signal d'horloge permettant un déclenchement certain de la bascule.

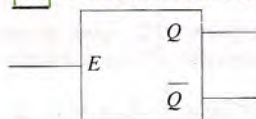
Durée des états 1 et 0

Chaque état du signal d'horloge doit avoir une durée minimale pour assurer un fonctionnement fiable de la bascule (la durée minimale à l'état haut n'est pas égale à celle de l'état bas).

Durées de transition du signal d'horloge

Les durées de transition du signal d'horloge représentent le temps de montée et le temps de descente de ce signal. Ces durées doivent être les plus courtes possible.

1 Présentation



Monostable

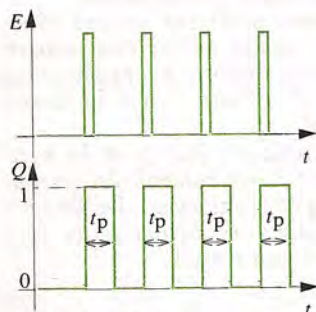


Fig. 1 : Exemple de chronogrammes de fonctionnement d'un monostable.

Les bascules étudiées précédemment sont des circuits logiques ayant deux sorties, en général complémentaires, auxquelles sont associées les variables Q et $\bar{Q}$ et deux états stables qui correspondent aux deux combinaisons suivantes :

$$Q = 0, \bar{Q} = 1 \text{ et } Q = 1, \bar{Q} = 0.$$

Il existe des circuits numériques à deux sorties complémentaires mais un seul état stable :

$$Q = 0, \bar{Q} = 1$$

Ces circuits sont appelés monostables.

Tant qu'aucun signal n'est appliqué à l'entrée d'un tel circuit, il reste à l'état stable indiqué.

Une impulsion de tension appliquée à l'entrée, provoque un changement d'état temporaire des sorties :

$$Q = 1 \text{ et } \bar{Q} = 0.$$

On dit que l'impulsion provoque le déclenchement du monostable (sous-entendu : déclenchement pour une phase de fonctionnement de durée limitée). En effet le monostable reste dans cet état pendant une durée t_p et revient de lui-même à l'état stable initial (fig. 1).

Cette durée t_p , appelée **durée de temporisation**, dépend des composants associés au circuit généralement.

Cette durée s'écrit souvent sous la forme :

$$t_p = kRC$$

Avec des opérateurs logiques de base, il est possible de réaliser un monostable. Beaucoup de montages, nécessitant des monostables, font appel à des circuits intégrés logiques spécialisés.

2 Monostable élémentaire

L'exemple proposé est réalisé à partir de deux opérateurs NON-ET en technologie CMOS (fig. 2). La résistance R_0 , placée à l'entrée du second opérateur NON-ET, a une valeur suffisamment élevée pour que l'on puisse négliger l'intensité du courant qui la traverse devant celles traversant la résistance et le condensateur.

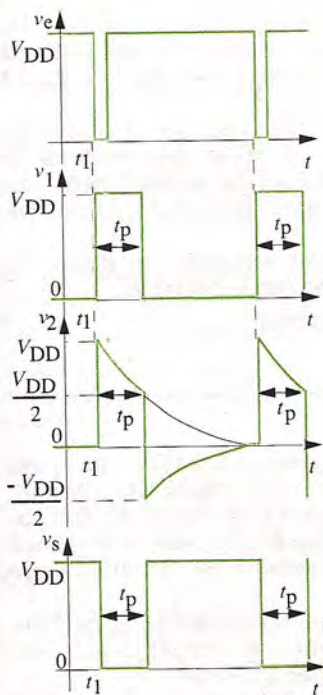
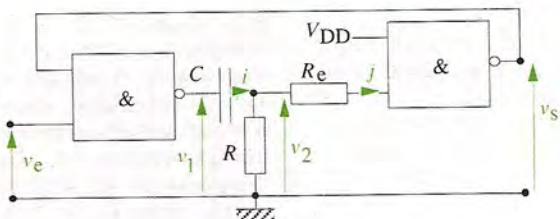


Fig. 2 : Chronogrammes d'un monostable élémentaire.



Rappels : Pour une tension d'entrée v_e d'un inverseur CMOS, inférieure à un seuil de transition $V_{tr} = \frac{V_{DD}}{2}$, la tension de sortie v_s de l'inverseur est égale à V_{DD} .

Pour une tension d'entrée v_e supérieure à V_{tr} la tension de sortie v_s de l'inverseur est nulle.

État stable

A l'état stable, aucun courant ne traverse le condensateur : $i = 0 \Rightarrow v_2 = 0$.

L'une des entrées du second opérateur NON-ET est à l'état 1, l'autre est à l'état 0; la sortie de cet opérateur est donc à l'état 1 : $v_s = V_{DD}$.

En choisissant $v_e = V_{DD}$, les entrées du premier opérateur sont à l'état 1, sa sortie est alors à l'état zéro : $v_1 = 0$. La tension aux bornes du condensateur est alors nulle.

La résistance R_e placée à l'entrée du second opérateur NON-ET fait que l'intensité j du courant la traversant est pratiquement nulle quelle que soit la valeur de la tension v_2 .

Fonctionnement

A un instant $t = t_1$, une impulsion portant temporairement la tension v_e à une valeur nulle, est appliquée à l'entrée de commande du premier opérateur : la tension v_1 prend instantanément la valeur V_{DD} .

Le condensateur n'admettant pas de discontinuité de tension, la tension à ses bornes reste nulle, ce qui impose $v_2 = V_{DD}$. Les deux entrées du second opérateur sont à l'état 1, sa sortie passe à l'état 0 : $v_s = 0$.

Le condensateur se charge à travers la résistance R avec la constante de temps $\tau = RC$, ce qui entraîne la décroissance de la tension v_2 . Lorsque $v_2 = V_{tr}$ (au bout d'une durée t_p) la sortie du second opérateur repasse à l'état 1, ce qui entraîne le

basculement du premier opérateur : $v_1 = 0$. Le condensateur n'admettant pas de discontinuité de tension, la tension à ses bornes reste égale à $V_{DD}/2$ ce qui impose $v_2 = -V_{DD}/2$.

Le condensateur se recharge avec une constante de temps voisine de RC . Ce n'est que lorsque le monostable est à nouveau à l'état stable (après une durée voisine de $5RC$) qu'il peut à nouveau être déclenché.

On montre que, pour cet exemple, la durée de temporisation est donnée par la relation :

$$t_p = 0,69RC$$

3 Utilisation de circuits intégrés

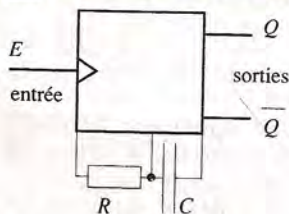


Fig. 3 : Monostable.

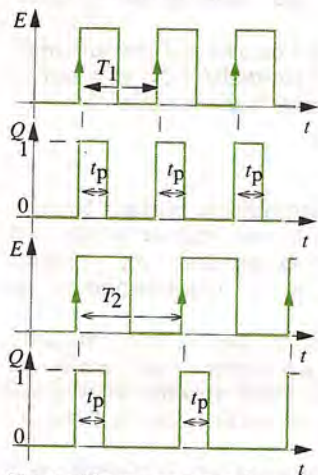


Fig. 4 : Chronogrammes de fonctionnement dans le cas où $t_p < \frac{T}{2}$ pour deux fréquences différentes de la tension d'entrée.

Avec des circuits intégrés spécialisés, on peut réaliser des monostables redéclenchables ou non. Quel que soit le monostable, la durée de temporisation est proportionnelle à la constante de temps d'un circuit RC qui est associé au circuit intégré (fig. 3) : $t_p = kRC$.

Le déclenchement des monostables peut se faire soit sur un front montant soit sur un front descendant du signal appliqué à l'entrée.

Nous allons expliquer le fonctionnement de ces deux sortes de monostables en les supposant actifs sur un front montant du signal d'entrée.

Monostable non redéclenchable

Choisissons un signal d'entrée périodique en créneaux de période T . Trois cas sont à considérer :

- $0 < t_p < T/2$ (fig. 4)

Sous l'action du premier front montant du signal d'entrée, la variable de sortie Q passe de 0 à 1. Elle reste égale à 1 pendant la durée t_p puis revient d'elle-même à 0 et ceci quelle que soit la période du signal d'entrée (en ayant toujours la condition $0 < t_p < T/2$).

- $T/2 < t_p < T$ (fig. 5)

Sous l'action du premier front montant du signal d'entrée, la variable de sortie Q passe de 0 à 1. Elle reste égale à 1 pendant la durée t_p puis revient d'elle-même à 0. Nous pouvons remarquer que l'action d'un front descendant, alors que le monostable n'est plus à l'état stable, ne le fait pas revenir à cet état stable.

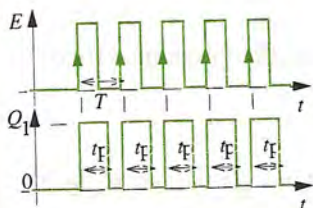


Fig. 5 : Chronogrammes de fonctionnement dans le cas où $\frac{T}{2} < t_p < T$. Au front descendant du signal, aucune modification de l'état du monostable ne se produit.

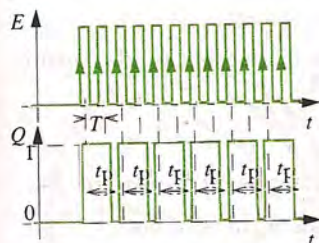


Fig. 6 : Chronogrammes de fonctionnement dans le cas où $t_p > T$.

• $t_p < T$ (fig. 6)

Sous l'action du premier front montant du signal d'entrée, la variable de sortie Q passe de 0 à 1. Elle reste égale à 1 pendant la durée t_p , puis revient d'elle-même à 0. L'action du second front montant (dans l'exemple choisi) du signal d'entrée est sans action sur le monostable.

Monostable redéclenchable

Le monostable précédent doit être revenu à son état stable avant de pouvoir être redéclenché. Ce n'est pas le cas d'un monostable redéclenchable (c'est d'ailleurs la seule différence entre ces deux sortes de monostables).

Dans l'exemple de la figure 7, la situation a illustre un fonctionnement décrit précédemment.

La situation b est différente : un signal d'entrée est appliqué alors que le monostable n'est pas revenu à l'état stable. Le front montant de ce signal redéclenche le monostable qui, à partir de cet instant reste t_p secondes supplémentaires dans cet état avant de revenir à l'état stable.

Dans la situation c, le premier front montant déclenche le fonctionnement du monostable. Les autres fronts montants appliqués alors que le monostable n'est pas à l'état stable provoquent une succession de redéclenchements.

Cet état intermédiaire cesse t_p secondes après le dernier front montant du signal d'entrée.

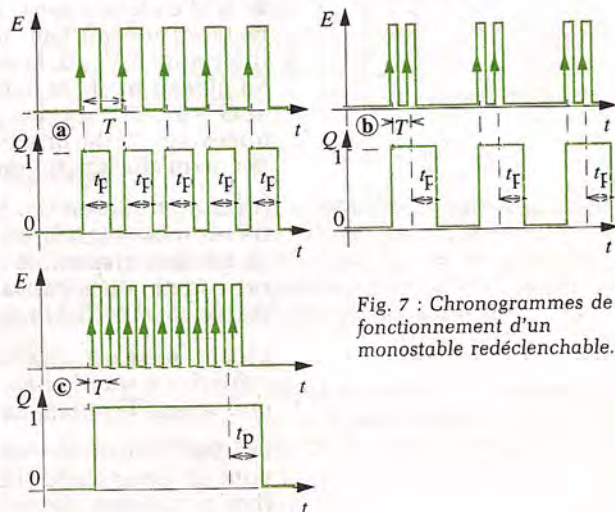


Fig. 7 : Chronogrammes de fonctionnement d'un monostable redéclenchable.

Souvent les monostables en circuit intégré ont deux entrées de déclenchement : l'une sur un front montant, l'autre sur un front descendant.

1 Description du circuit intégré NE 555

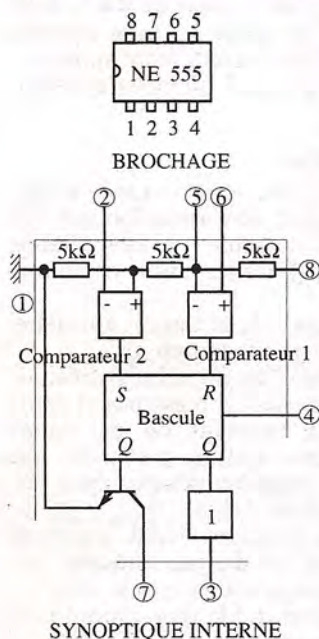


Fig. 1

C'est un composant très utilisé pour un fonctionnement en horloge.

Synoptique interne

Ce circuit comporte les éléments suivants (fig. 1) :

- un comparateur 1 (dit haut) capable de forcer une bascule RS à l'état 1;
- un comparateur 2 (dit bas) capable de faire passer la même bascule RS à l'état 0;
- une bascule bistable RS dont la sortie $\bar{Q}$ commande un transistor;
- un transistor à collecteur ouvert;
- un circuit tampon capable d'absorber ou de fournir de forts courants (200 mA).

Description

Le circuit NE 555 est un composant à huit broches :

- broche 1 : à relier à la masse du montage;
- broche 2 : elle est reliée à l'entrée inverseuse du comparateur 2; tant que la tension v_2 de cette broche reste supérieure à $V_{cc}/3$ (V_{cc} étant la tension d'alimentation), la sortie du comparateur 2 reste au niveau bas; lorsque la tension v_2 devient inférieure à $V_{cc}/3$, la sortie du comparateur 2 passe au niveau haut : la bascule est mise à l'état 1 ($Q = 1$ et $\bar{Q} = 0$); ce sont des impulsions de tension appliquées sur cette broche qui permettent le déclenchement du fonctionnement du circuit;
- broche 3 : c'est la sortie du NE 555 qui peut recevoir ou fournir un courant d'intensité 200 mA (la tension disponible entre cette borne et la masse est commandée par la sortie Q de la bascule et a même valeur logique);
- broche 4 : c'est une entrée logique prioritaire qui initialise à zéro la bascule RS ($Q = 0$ et $\bar{Q} = 1$) quels que soient les états des entrées 2 et 6;
- broche 5 : cette broche est souvent inutilisée; le pont diviseur constitué par les résistances de 5 kΩ fixe la tension de référence du comparateur 1 à

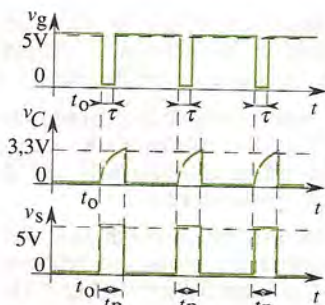


Fig. 3 : Chronogrammes de fonctionnement du NE 555 en monostable avec $\tau < t_p$.

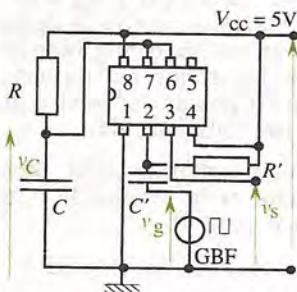


Fig. 4 : Autre montage du NE 555 en monostable.

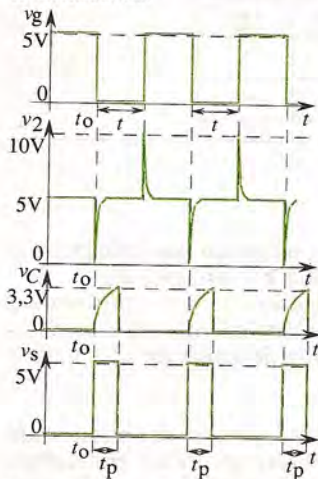


Fig. 5 : Chronogrammes de fonctionnement du NE 555 en monostable avec $\tau > t_p$.

La tension de déclenchement v_g (appliquée à la broche 2) est initialement supérieure à $V_{cc}/3$ (fig. 3). La bascule est à l'état 0 ($Q = 0$ et $\bar{Q} = 1$), ce qui implique :

- $v_7 \approx v_1 = 0$ V, donc $v_c = 0$ V car le transistor se comporte comme un court-circuit,

- $v_s = 0$ V.

A un instant t_0 , on applique brièvement (pendant une durée τ) une tension nulle à la broche 2. La tension v_2 devenant inférieure à $V_{cc}/3$, la sortie du comparateur 2 passe au niveau haut : la bascule est mise à l'état 1 ($Q = 1$ et $\bar{Q} = 0$), ce qui implique :

- $v_7 = 0$ V, car le condensateur n'est pas chargé,

- $v_s = V_{cc} = 5$ V.

A partir de cet instant, le condensateur se charge à travers la résistance R . C'est une charge sous tension constante ($V_{cc} = 5$ V).

La remise au niveau haut de la tension v_2 est sans action sur le circuit.

Dès que la tension v_C (donc v_6) atteint $2 V_{cc}/3$, la tension de sortie du comparateur 1 passe au niveau haut : la bascule est remise à l'état 0 ($Q = 0$ et $\bar{Q} = 1$); ce qui implique :

- $v_7 \approx v_1 = 0$ V,

- $v_s = 0$ V.

Le circuit est revenu à son état initial, état qu'il conservera jusqu'à la prochaine impulsion.

La durée de la temporisation est :

$$t_p = 1,1 RC$$

Remarque

Pour que le montage précédent fonctionne correctement, il est nécessaire que la durée de l'impulsion soit inférieure à la durée de la temporisation. Si les impulsions correspondant à la tension v_g sont trop longues, elles sont appliquées par l'intermédiaire d'un ensemble $R'C'$ tel que $R'C' \ll RC$. Ce circuit $R'C'$ (fig. 4) est monté en dérivateur : son rôle est de convertir des impulsions longues en impulsions brèves et ce sont les impulsions brèves qui vont déclencher le fonctionnement du monostable (fig. 5) comme nous l'avons décrit précédemment.

3 Utilisation en astable

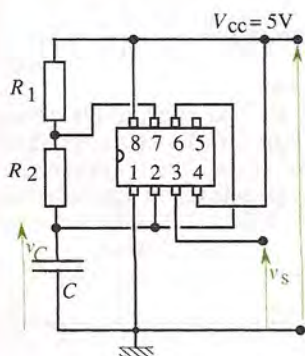


Fig. 6 : Montage du NE 555 en astable.

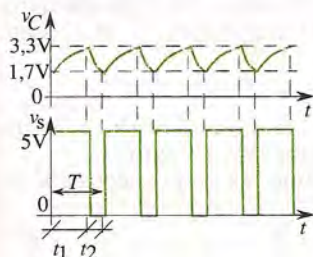


Fig. 7 : Chronogrammes de fonctionnement du NE 555 en astable.

Montage

La figure 6 représente un montage possible pour le fonctionnement du NE 555 en multivibrateur astable. La période T de l'oscillateur est fonction des trois composants extérieurs : la résistance R_1 , la résistance R_2 et le condensateur de capacité C .

Fonctionnement

Dès la mise sous tension, le montage se comporte comme un oscillateur (fig. 7). Nous nous plaçons en régime établi (nous ne nous intéressons pas à la naissance des oscillations).

Les broches 2 et 6 étant reliées, le condensateur subit une succession de charges et de décharges, la tension à ses bornes évoluant entre $V_{cc}/3$ et $2V_{cc}/3$. Durant la charge du condensateur, la tension de sortie est au niveau haut : le transistor de sortie est à l'état bloqué et la charge s'effectue à travers les deux résistances R_1 et R_2 avec la constante de temps $(R_1 + R_2)C$. Lorsque la tension v_C aux bornes du condensateur atteint la valeur $2V_{cc}/3$, la tension de sortie du comparateur 1 passe au niveau haut : la bascule est remise à l'état 0 ($Q = 0$ et $Q = 1$); le transistor de sortie est en court-circuit, ce qui implique une décharge du condensateur à travers la résistance R_2 avec la constante de temps R_2C . Lorsque la tension v_C atteint la valeur $V_{cc}/3$, la tension de sortie du comparateur 2 passe au niveau haut : la bascule est mise à l'état 1 ($Q = 1$ et $Q = 0$); le transistor de sortie est à l'état bloqué et la charge s'effectue à travers R_1 et R_2 .

Remarque :

On montre que la durée d'un état haut a pour expression : $t_h \approx 0,7 (R_1 + R_2)C$, et que celle d'un état bas est : $t_b \approx 0,7 R_2C$, ce qui donne pour expression de la période des oscillations :

$$T = t_h + t_b \approx 0,7 (R_1 + 2R_2)C$$

(Pour ce calcul, voir «Électronique analogique», chapitre 30, de la même collection Étapes).

La modification de C entraîne une modification de T , période des oscillations, proportionnelle à C .

La modification de R_1 ou de R_2 entraîne la modification du rapport cyclique et de la période de la tension de sortie (si $R_1 \ll R_2$, le rapport cyclique reste toujours voisin de 0,5 et la période ne dépend plus que de R_2 et de C).

17 COMPTEURS ASYNCHRONES

Par convention, à une borne d'entrée ou de sortie désignée par une lettre capitale (par exemple E) nous associons une variable logique désignée au moyen de la même lettre mais écrite en italique (E dans l'exemple choisi). Rappelons que la valeur de cette variable (0 ou 1) correspond à certaines valeurs d'une tension électrique appliquée entre la borne considérée et la masse du montage.

Une entrée E est dans l'état 1 si la variable $E = 1$. Elle est à l'état 0 si $E = 0$.

Les compteurs permettent d'établir une relation d'ordre de succession d'événements. Ils sont utilisés pour effectuer :

- du comptage,
- des opérations de séquençement,
- la division de fréquence,
- des opérations arithmétiques.

Dans ce premier chapitre sur les compteurs, nous nous intéressons aux compteurs asynchrones encore appelés compteurs série. Leur caractéristique principale est la propagation en cascade de l'ordre des changements d'état de chacun des éléments (des bascules) qui les composent.

1 Compteur asynchrone modulo 2ⁿ

Souvent, et nous admettrons que c'est le cas ici, si les entrées J et K sont laissées « en l'air », on a :

$$J = K = 1$$

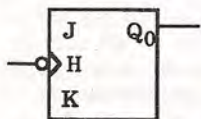


Fig. 1 : Compteur élémentaire modulo 2.

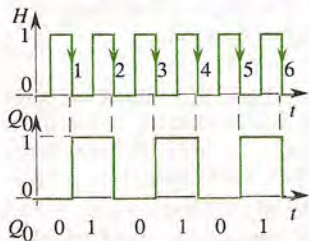


Fig. 2 : Chronogrammes d'un compteur élémentaire modulo 2.

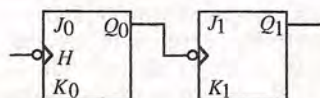


Fig. 3 : Compteur élémentaire modulo 4.

Compteurs élémentaires

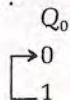
Premier exemple

Soit une bascule JK, active sur un front descendant du signal d'horloge, dont les entrées J et K sont placées dans l'état 1 (fig. 1) :

$$J = K = 1$$

A l'entrée H, appliquons un signal d'horloge ; nous obtenons les chronogrammes de la figure 2.

Le tableau ci-après regroupe les états successifs de la variable de sortie Q_0 :



Nous avons réalisé un compteur qui compte de 0 à 1, il revient à 0 et recommence.

En effet, rappelons que compter, par exemple, de 0 à 9 en base 2, c'est passer successivement de 0000 à 0001, puis de 0001 à 0010, puis de 0010 à 0011 et ainsi de suite pour arriver à 1111. Le chiffre de poids le plus faible passe alternativement de 0 à 1 et inversement : c'est le cas de Q_0 qui peut représenter le chiffre de poids le plus faible du nombre d'impulsions d'horloge exprimé en base 2.

Deuxième exemple

Prenons maintenant deux bascules JK associées en cascade (fig. 3). Un signal d'horloge est appliqué à l'entrée H de la première bascule. Le signal disponible à la sortie Q_0 de cette bascule est utilisé comme signal d'horloge pour la deuxième.

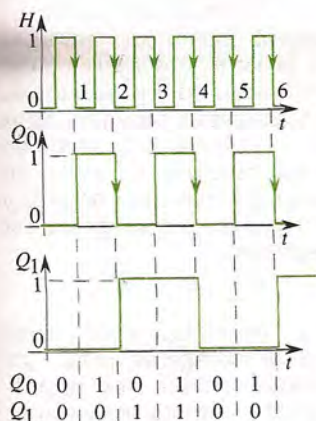


Fig. 4 : Chronogrammes d'un compteur élémentaire modulo 4.

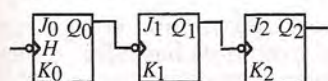


Fig. 5 : Compteur élémentaire modulo 8.



Fig. 6 : Chronogrammes d'un compteur élémentaire modulo 8.

Pour afficher le résultat du comptage dans le système décimal, il est possible d'utiliser un afficheur sept segments à décodeur incorporé.

En choisissant toujours des bascules actives sur un front descendant, nous obtenons les chronogrammes de la figure 4.

En regroupant les différents états successifs des variables de sortie Q_0 et Q_1 , nous avons :

	Q_1	Q_0
→ 0	0	
0	1	
1	0	
1	1	

État initial
après la 1^{re} impulsion
après la 2^e impulsion
après la 3^e impulsion

Pour ce compteur, quatre états différents sont possibles. Il permet le comptage dans le système binaire naturel de 00 à 11, il revient à 00 et recommence : c'est un compteur progressif.

Compteur octal

Avec trois bascules JK identiques aux précédentes et associées en cascade (fig. 5) nous avons un compteur octal.

En regroupant les différents états successifs des variables de sortie Q_0 , Q_1 et Q_2 , nous obtenons le tableau suivant :

Affichage			Nombre d'impulsions comptées
Q_2	Q_1	Q_0	
→ 0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Dans ce tableau, les deux flèches symbolisent le retour à zéro.

Il y a huit combinaisons différentes des variables de sortie Q_0 , Q_1 et Q_2 .

Q_0 est le bit de poids le plus faible, Q_2 est le bit de poids le plus fort.

A chaque nombre binaire correspond un nombre décimal qui est aussi le nombre d'impulsions du signal d'horloge émises après passage par la combinaison 0 0 0.

Conclusions

- Les bascules ne changent pas d'état simultanément mais les unes à la suite des autres : ces compteurs sont dits asynchrones ou à propagation.
- En observant les chronogrammes obtenus, nous constatons que la période de la tension de sortie de la dernière bascule est un multiple de celle du signal d'horloge. Dans le cas du compteur octal il y a division de la fréquence par 8 : un compteur est aussi un diviseur de fréquences.

Pour augmenter le « modulo » d'un compteur, il suffit d'augmenter le nombre de bascules.

Un compteur modulo 2^n est un diviseur de fréquence par 2^n .

Modulo d'un compteur

Les compteurs décrits précédemment sont cycliques. Dans le cas d'un compteur octal, un cycle correspond au dénombrement des impulsions d'horloge de 0 à 7, c'est-à-dire au passage par toutes les combinaisons binaires de 0000 à 1111. Le modulo d'un compteur est égal à ce nombre de combinaisons.

Un compteur octal est donc un compteur modulo 8. Il a été réalisé avec trois bascules. Avec n bascules, nous aurions un compteur modulo 2^n .

2 Compteur asynchrone à modulo inférieur à 2^n

Reprenons l'exemple d'un compteur modulo 8 (octal). Le montage est réalisé avec des bascules JK ayant chacune une entrée de mise à zéro reliée à la sortie d'un opérateur NON-ET (fig. 7).

Sortie de l'opérateur NON-ET à l'état 1

La mise à l'état 1 de la sortie de l'opérateur NON-ET n'affecte en rien le fonctionnement du compteur qui est alors à modulo 8.

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

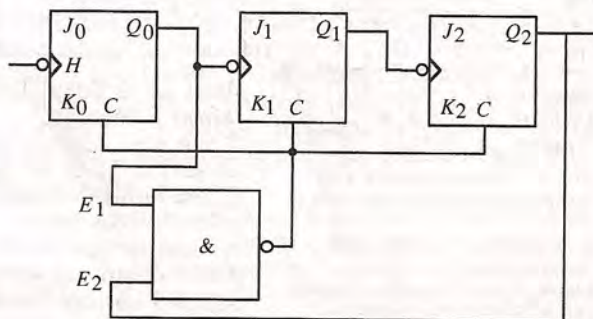


Fig. 7 : Compteur élémentaire modulo 5.

Pour forcer la mise à zéro du compteur, on utilise des bascules à entrées de prépositionnement supposées, ici, actives à l'état bas.

Sortie de l'opérateur NON-ET à l'état 0

La mise à l'état 0 de la sortie de l'opérateur NON-ET met toutes les sorties des bascules à l'état 0 et le compteur affiche 000 puis compte à nouveau.

Fonctionnement

Dans l'exemple choisi, les entrées E_1 et E_2 sont reliées respectivement aux sorties Q_0 et Q_2 des bascules. Nous pouvons donc écrire les équations logiques suivantes :

$$E_1 = Q_0 \quad \text{et} \quad E_2 = Q_2.$$

La sortie de l'opérateur NON-ET est à l'état 0, lorsque $E_1 = 1$ et $E_2 = 1$.

Cette condition est réalisée lorsque le contenu du compteur passe de 100 à 101, c'est-à-dire à la cinquième impulsion d'horloge.

L'entrée de prépositionnement des bascules est à ce moment mise à zéro, ce qui remet le contenu du compteur à 000. La mise à zéro des bascules supprime la double égalité $E_1 = E_2 = 1$, ce qui a pour effet de remettre l'entrée de prépositionnement des bascules à 1 : un nouveau comptage peut avoir lieu.

Conséquences : Le compteur compte de 000 à 100 et les états 101, 110 et 111 sont exclus. Il n'affiche plus que cinq états différents : c'est un compteur modulo 5.

Remarque : L'inconvénient majeur des compteurs asynchrones résulte du fait que chaque bascule est déclenchée par la transition de la sortie de celle qui la précède. Chaque bascule introduit un retard de propagation t_{pd} . Les bascules étant en cascade, les retards s'additionnent et la $n^{\text{ième}}$ bascule ne réagit que nt_{pd} secondes après l'arrivée de l'impulsion d'horloge.

Ces compteurs ne peuvent donc pas être utilisés à haute fréquence.

3 Décompteur asynchrone

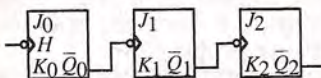


Fig. 8 : Décompteur élémentaire modulo 8.

Décompter consiste à faire régresser les états en partant du nombre le plus élevé (111 pour un décompteur octal) pour arriver à 000.

Le même circuit peut généralement fonctionner en compteur ou en décompteur.

Reprenons l'exemple du compteur octal à bascules JK. En utilisant les sorties complémentées (fig. 8), nous obtenons un compteur régressif modulo 8.

18 COMPTEURS SYNCHRONES

Dans le chapitre précédent, nous avons vu que l'association en cascade de bascules dans un compteur asynchrone entraînait des retards de propagation.

Ces retards limitent la fréquence maximale d'utilisation. Ces compteurs deviennent inutilisables pour des comptages longs.

1 Fonctionnement d'un compteur synchrone modulo 8

La complexité de la réalisation croît avec l'augmentation du modulo du compteur.

	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Fig. 1 : Cycle de cadencement d'un compteur modulo 8.

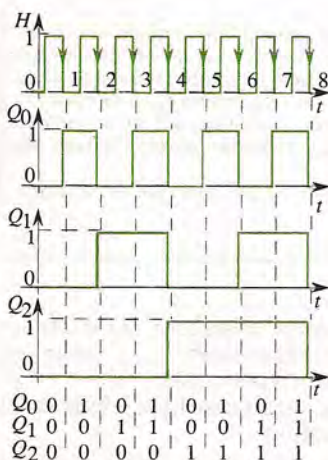


Fig. 2 : Chronogrammes d'un compteur élémentaire modulo 8.

Pour éviter ces retards, il est indispensable que toutes les bascules soient déclenchées simultanément. Ainsi lorsque leur déclenchement est synchronisé par le même signal d'horloge. De tels compteurs sont dits parallèles ou synchrones.

Les compteurs synchrones sont d'une plus grande complexité que les compteurs série mais leur fréquence maximale d'utilisation est bien plus élevée : elle n'est plus limitée que par le retard de propagation d'une seule bascule et non des n bascules.

Un compteur modulo 2^n nécessite n bascules : un compteur modulo 8, c'est-à-dire 2^3 , nécessite trois bascules.

Comme pour un compteur asynchrone modulo 8, le cycle de fonctionnement doit respecter le tableau de la figure 1.

Un compteur modulo 8 est un compteur à trois bits, disponibles sur les sorties Q_0 , Q_1 et Q_2 des bascules.

Cela veut dire que :

- la première bascule change d'état après chaque impulsion d'horloge afin d'obtenir l'alternance 010101...;
- la deuxième bascule change d'état toutes les deux impulsions d'horloge, son état dépendant également de celui de la première bascule;
- la troisième bascule change d'état toutes les quatre impulsions d'horloge, son état dépendant de celui des deux autres bascules.

La figure 2 donne les chronogrammes illustrant ce fonctionnement : ils sont identiques à ceux d'un compteur asynchrone modulo 8 car nous n'avons pas représenté les retards de propagation.

Ce principe de fonctionnement montre que la réalisation d'un tel compteur est plus complexe que celle d'un compteur asynchrone. En effet,

pour certaines impulsions d'horloge, les bascules doivent commuter et pour d'autres, elles ne le doivent pas. Aux trois bascules, il est nécessaire de rajouter un dispositif qui fait respecter le bon ordre des commutations.

2 Réalisation d'un compteur synchrone modulo 8

La figure 3 propose un exemple de réalisation possible avec des bascules JK actives sur les fronts descendants du signal d'horloge (fig. 4).

Fig. 3 : Compteur élémentaire modulo 8.

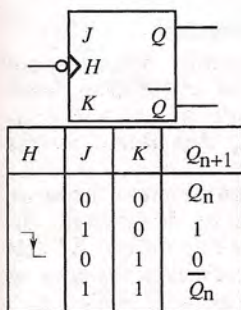
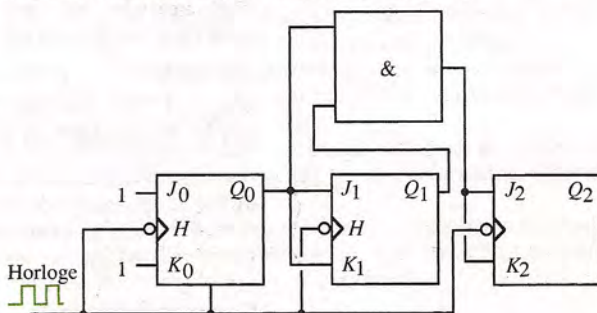


Fig. 4 : Bascule JK active sur un front descendant et sa table de vérité.



Commutation de la première bascule

Les entrées J_0 et K_0 de la première bascule sont à l'état 1 : elle commute à chaque impulsion d'horloge.

Commutation de la deuxième bascule

Les entrées J_1 et K_1 de la deuxième bascule sont reliées entre elles ainsi qu'à la sortie Q_0 de la première bascule. La table de vérité générale de cette bascule JK peut, dans ce cas, être simplifiée, car soit $J_1 = K_1 = 0$ soit $J_1 = K_1 = 1$ (fig. 5).

H	J	K	Q_{n+1}
0	0	0	Q_n
1	1	1	$\overline{Q_n}$

Fig. 5 : Table de vérité simplifiée.

Les valeurs de J_1 et de K_1 sont liées aux valeurs de Q_0 .

1^{er} cas : $Q_0 = 0$ avant l'arrivée d'un front descendant du signal d'horloge : $J_1 = K_1 = 0$.

L'arrivée de ce front sera sans action sur la commutation de cette bascule car elle est bloquée : Q_1 conservera la même valeur qu'avant le passage de ce front.

Cette observation est conforme au principe de fonctionnement décrit précédemment.

Exemples :

- avant l'arrivée du front 1 : $J_1 = K_1 = 0$ et $Q_1 = 0$, après le passage du front 1 : $Q_1 = 0$;
- avant l'arrivée du front 3 : $J_1 = K_1 = 0$ et $Q_1 = 1$, après le passage du front 3 : $Q_1 = 1$.

2^e cas : $Q_0 = 1$ avant l'arrivée d'un front descendant du signal d'horloge : $J_1 = K_1 = 1$; la bascule peut être activée par un front descendant du signal d'horloge et l'arrivée de ce front la fait basculer.

Exemples :

- avant l'arrivée du front 2 : $J_1 = K_1 = 1$ et $Q_1 = 0$, après le passage du front 1 : $Q_1 = 1$;
- avant l'arrivée du front 4 : $J_1 = K_1 = 1$ et $Q_1 = 1$, après le passage du front 3 : $Q_1 = 0$. Cette bascule commute toutes les deux impulsions d'horloge.

Commutation de la troisième bascule

Les entrées J_2 et K_2 de la troisième bascule sont reliées entre elles ainsi qu'à la sortie d'un opérateur ET dont les entrées sont reliées respectivement aux sorties Q_0 et Q_1 des deux autres bascules.

Cette troisième bascule commute lorsque $J_2 = K_2 = 1$, sous l'action d'un front descendant du signal d'horloge. La présence de l'opérateur ET fait que la condition $J_2 = K_2 = 1$ est réalisée lorsque sa sortie est à l'état 1, c'est-à-dire pour $Q_0 = Q_1 = 1$. Il est donc nécessaire que le front actif du signal d'horloge arrive après que l'on ait $J_2 = K_2 = 1$ ou encore $Q_0 = Q_1 = 1$.

Exemples :

- avant l'arrivée du front 4 : $J_2 = K_2 = 1$ et $Q_2 = 0$, après le passage du front 4 : $Q_2 = 1$.
- avant l'arrivée des fronts 5, 6 et 7, la condition $J_2 = K_2 = 1$ n'est pas réalisée et $Q_2 = 1$; après le passage de ces fronts cette troisième bascule ne commute pas : $Q_2 = 1$.
- avant l'arrivée du front 8 : $J_2 = K_2 = 1$ et $Q_2 = 1$, après le passage du front 8 : $Q_2 = 0$. Cette bascule commute toutes les quatre impulsions d'horloge.

19 REGISTRE À DÉCALAGE

1 Principe de fonctionnement

Un mot binaire de n bits est inscrit dans une cellule mémoire constituée de n mémoires élémentaires : la cellule mémoire est appelée aussi registre mémoire. Les mémoires élémentaires sont des bascules.

Exemple : 1011 est un mot binaire défini par quatre bits ($n = 4$); sa représentation dans un registre mémoire est :

1	0	1	1
---	---	---	---

C'est une cellule mémoire (ou registre mémoire) constituée de quatre cellules élémentaires.

La fonction décalage, consiste à faire glisser l'information de chaque cellule élémentaire dans des cellules élémentaires adjacentes.

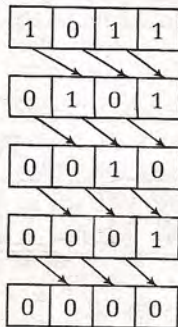
Si ce transfert a lieu vers les cellules élémentaires adjacentes de droite, le registre mémoire est un registre à décalage à droite.

Si ce transfert a lieu vers les cellules élémentaires adjacentes de gauche, le registre mémoire est un registre à décalage à gauche.

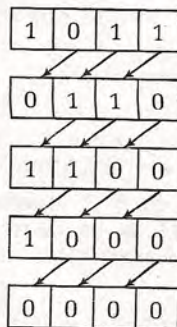
En l'absence d'un transfert d'informations d'un registre à un autre :

- lors d'un décalage à droite à l'intérieur du registre, la première cellule élémentaire (cellule de gauche) reçoit un zéro à chaque décalage;
- lors d'un décalage à gauche, la première cellule élémentaire (cellule de droite) reçoit un zéro à chaque décalage.

Exemples :



Décalage à droite



Décalage à gauche

Les caractéristiques dynamiques des registres sont spécifiées par les constructeurs, en particulier la fréquence maximale $f_{\max}$ et la largeur minimale $\theta_{\min}$ de l'impulsion d'horloge. Par exemple, pour le circuit intégré 4015, double registre à décalage à 4 bits, à bascules D : $f_{\max} = 9 \text{ MHz}$ et $\theta_{\min} = 50 \text{ ns}$.

2 Transfert du contenu d'un registre A dans un registre A'

Nous venons de voir le décalage des informations à l'intérieur d'un registre. Il est également possible de transférer le contenu d'un registre vers un autre.

Prenons l'exemple de deux registres à décalage de quatre bits dont le transfert registre à registre se fait en série (fig. 1).

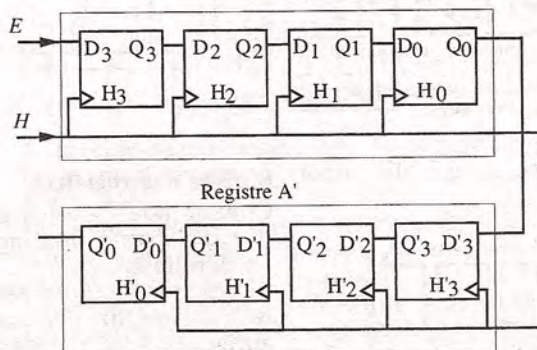
Les impulsions d'horloge déclenchent le transfert des informations suivant le tableau suivant :

registre A				registre A'				
Q_3	Q_2	Q_1	Q_0	Q'_3	Q'_2	Q'_1	Q'_0	
1	0	1	0	0	0	0	0	état initial
0	1	0	1	0	0	0	0	1 ^{re} impulsion
0	0	1	0	1	0	0	0	2 ^e impulsion
0	0	0	1	0	1	0	0	3 ^e impulsion
0	0	0	0	1	0	1	0	4 ^e impulsion

Comme nous l'avons vu précédemment, certains registres peuvent être à décalage à droite, d'autres à décalage à gauche. Il existe des registres à décalage ayant une commande de mode qui permet un fonctionnement soit en décalage à droite, soit en décalage à gauche : ce sont des registres bidirectionnels.

Au front montant de chaque impulsion d'horloge, chaque bascule reçoit la valeur mémorisée dans la bascule immédiatement à sa gauche. Il y a transfert progressif bit par bit du contenu du registre A dans le registre A'. Au bout de quatre impulsions, il y aura eu transfert des quatre bits du registre A dans le registre A'.

Fig. 1 : Transfert des informations d'un registre A dans un registre A'.



3 Généralisation

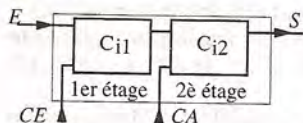


Fig. 2 : Cellule mémoire à deux étages.

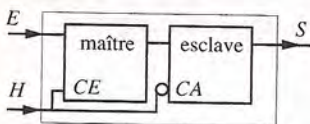


Fig. 3 : Bascule « maître-esclave ».

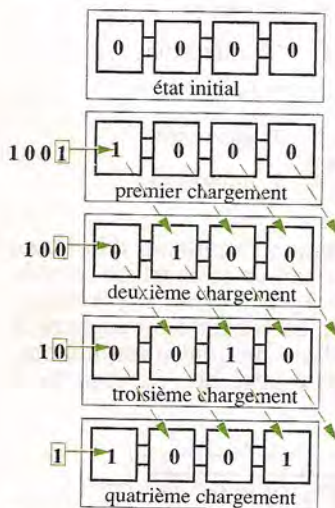


Fig. 4 : Registre quatre bits à chargement en série.

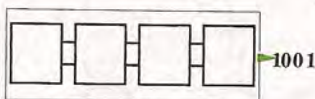


Fig. 6 : Registre quatre bits à déchargement en série.

Registres à décalage intégrés

Actuellement les registres à décalage se présentent sous la forme de circuits intégrés. Les cellules mémoires sont à deux étages (fig. 2) : une commande d'enregistrement CE transfère l'information d'entrée en sortie du premier étage sans modification du deuxième étage; une commande d'affichage CA transfère l'information en sortie du deuxième étage. Lorsque les entrées CE et CA sont complémentaires et commandées au rythme d'une horloge (fig. 3), la cellule a une structure « maître-esclave » (voir paragraphe 2 du chapitre 14).

Entrées des registres

Le « chargement » des registres à décalage peut se faire en série ou en parallèle. Suivant le mode de « chargement », un registre à décalage peut avoir une ou plusieurs entrées.

« Chargement » en série : le registre a une entrée; les différents éléments du nombre binaire arrivent les uns après les autres à l'entrée du registre (fig. 4) : le chargement d'un bit s'accompagne d'un décalage à droite (ou à gauche) du contenu de chaque mémoire élémentaire du registre.

« Chargement » en parallèle : le registre a autant d'entrées que de bits à charger; les différents éléments du nombre binaire arrivent simultanément aux entrées des différentes bascules du registre (fig. 5).

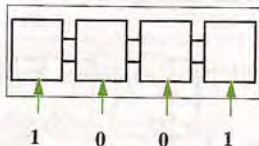


Fig. 5 : Registre quatre bits à chargement en parallèle.

Sorties des registres

Comme pour le « chargement », le « déchargement » des registres à décalage peut se faire en série ou en parallèle.

« Déchargement » en série : le registre a une sortie; le transfert du contenu du registre s'effectue bit après bit avec décalage simultané à droite (ou à gauche) du contenu de chaque mémoire élémentaire du registre (fig. 6).

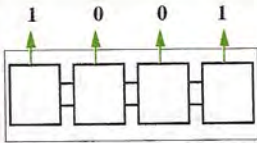


Fig. 7 : Registre quatre bits à déchargement en parallèle.

- « Déchargement » en parallèle : le registre a autant de sorties que de bits à transférer ; le transfert de de tous les bits est simultané (fig. 7).
Un transfert en parallèle est beaucoup plus rapide qu'un transfert en série. La mise en œuvre d'un transfert en série est beaucoup plus simple qu'un transfert en parallèle.
- Les combinaisons des différents modes de « chargement » et de « déchargement » sont possibles :
- registres série/série : « chargement » et « déchargement » se font en série ;
- registres série/parallèle : le « chargement » se fait en série et le « déchargement » en parallèle ;
- registres parallèle/série : le « chargement » se fait en parallèle et le « déchargement » en série ;
- registres parallèle/parallèle : « chargement » et « déchargement » se font en parallèle.

4 Exemple d'application : compteur en anneau

Un compteur en anneau est un registre à décalage dont la sortie de la dernière bascule est reliée à l'entrée de la première (fig. 8) : il y a rétroaction entre la sortie de la dernière bascule du registre et l'entrée de la première bascule.

En regroupant les différents états successifs de Q_0 , Q_1 , Q_2 et Q_3 , nous avons :

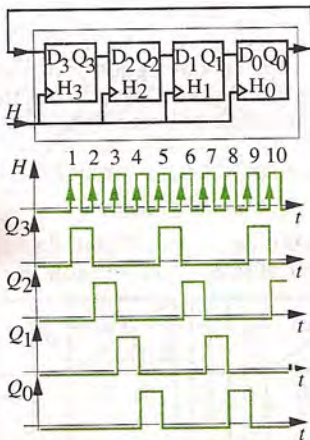


Fig. 8 : Compteur en anneau de quatre bits.

	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

A la cinquième impulsion, le compteur reprend l'état qu'il avait au début du comptage, c'est-à-dire 1000.

Nous constatons qu'un 1 circule dans le registre tant que le signal d'horloge est appliqué d'où le nom de compteur en anneau.

L'exemple choisi est un compteur modulo 4 car il y a quatre combinaisons différentes.

C'est un compteur très facile à décoder puisque pour chaque combinaison, une seule des sorties est à l'état 1.

1 Demi-additionneur

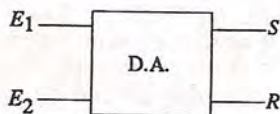


Fig. 1 : Demi-additionneur (DA).

E_1	E_2	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig. 2

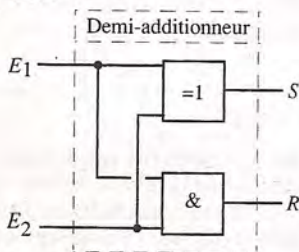


Fig. 3 : Structure du demi-additionneur (DA).

Principe

Un demi-additionneur (fig. 1) permet de réaliser l'addition de deux bits figurés par les variables E_1 et E_2 . Le résultat de l'addition est la variable S et éventuellement une retenue figurée par R .

Équations booléennes

La table de vérité est donnée figure 2 :

Les équations booléennes qui l'accompagnent sont :

$$S = \bar{E}_1 \cdot E_2 + E_1 \cdot \bar{E}_2$$

$$S = E_1 \oplus E_2 \quad (1)$$

et :

$$R = E_1 \cdot E_2 \quad (2)$$

Aux relations (1) et (2) correspondent les opérateurs OU EXCLUSIF et ET de la figure 3.

Limitations

Pour additionner deux nombres à plusieurs bits, il faut tenir compte de l'éventuelle retenue de l'étage précédent (revoir le chapitre 2). Un demi-additionneur ne peut donc réaliser une telle opération. Pour éliminer cette difficulté, il faut un circuit à trois entrées : c'est un additionneur complet.

2 Additionneur complet

Principe

Les variables S et R (fig. 4) correspondent au résultat de l'addition de E_1 , E_2 et R_{n-1} ; la sortie R étant toujours une éventuelle retenue R_n .

Tables logiques

Table de vérité

E_1	E_2	R_{n-1}	S	R_n
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

Tables de Karnaugh

Table de la variable S

$E_1 E_2$		R_{n-1}	
		0	1
0 0	0 1	0	1
0 1	1 0	1	0
1 1	0 1	1	1
1 0	1 0	1	0

Table de la variable R_n

$E_1 E_2$		R_{n-1}	
		0	1
0 0	0 0	0	0
0 1	0 1	0	1
1 1	1 1	1	1
1 0	0 1	0	1

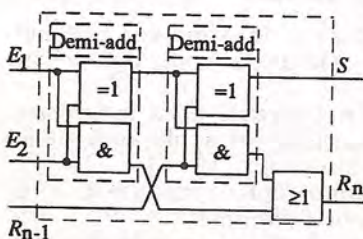


Fig. 4 : Structure de l'additionneur complet.

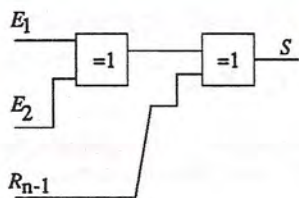


Fig. 5

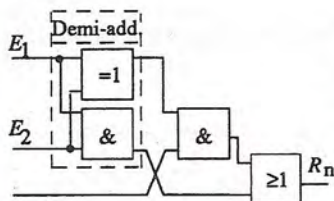


Fig. 6

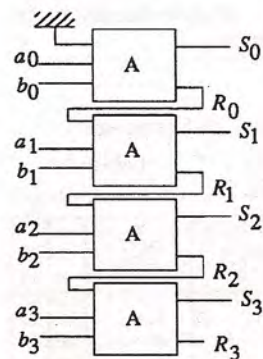


Fig. 7 : Structure d'un additionneur à quatre bits.

Équations booléennes

A partir des tables de Karnaugh précédentes nous obtenons les équations booléennes suivantes :

$$S = \bar{E}_1 \cdot \bar{E}_2 \cdot R_{n-1} + \bar{E}_1 \cdot E_2 \cdot \bar{R}_{n-1} + E_1 \cdot \bar{E}_2 \cdot \bar{R}_{n-1} + E_1 \cdot E_2 \cdot R_{n-1}$$

et :

$$R_n = \bar{E}_1 \cdot E_2 \cdot R_{n-1} + E_1 \cdot \bar{E}_2 \cdot \bar{R}_{n-1} + \bar{E}_1 \cdot E_2 \cdot \bar{R}_{n-1} + E_1 \cdot E_2 \cdot R_{n-1}$$

$$\text{Soit encore : } S = (E_1 \oplus E_2) \oplus R_{n-1} \quad (3)$$

$$R_n = (E_1 \oplus E_2) \cdot R_{n-1} + E_1 \cdot E_2 \quad (4)$$

L'équation (3) est réalisée par la mise en cascade de deux opérateurs OU EXCLUSIF (fig. 4).

L'équation (4) nécessite les opérateurs : OU EXCLUSIF, OU et ET (fig. 6).

La superposition des structures décrites figures 5 et 6 fournit la structure de l'additionneur complet qui apparaît comme l'association de deux demi-additionneurs (fig. 4).

Application : additionneur à quatre bits

Calculons la somme S de deux nombres binaires X et Y définis par quatre bits chacun :

$$X = (a_3 a_2 a_1 a_0) \quad \text{et} \quad Y = (b_3 b_2 b_1 b_0)$$

$$\begin{array}{r} X \\ + Y \\ \hline = S \end{array} \quad \begin{array}{r} a_3 \quad a_2 \quad a_1 \quad a_0 \\ + \quad b_3 \quad b_2 \quad b_1 \quad b_0 \\ \hline s_3 \quad s_2 \quad s_1 \quad s_0 \\ r_3 \quad r_2 \quad r_1 \quad r_0 \end{array}$$

Le calcul de la somme nécessite donc quatre additionneurs complets (fig. 7).

3 Demi-soustracteur

Principe

Un demi-soustracteur effectue la différence

$S = E_2 - E_1$ de 2 bits E_1 et E_2 avec, éventuellement, un report R.

Équations booléennes

La table de vérité est la suivante :

E_1	E_2	S	R
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

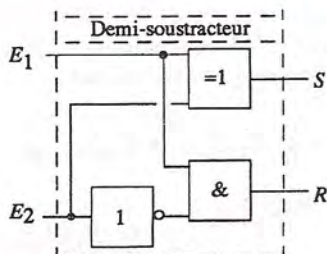


Fig. 8 : Structure d'un demi-soustracteur : $S = E_2 - E_1$.

4 Soustracteur complet

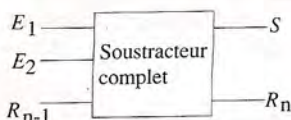


Fig. 9. Soustracteur complet.

Les équations booléennes qui accompagnent cette table de vérité sont :

$$S = E_1 \cdot \bar{E}_2 + \bar{E}_1 \cdot E_2 = E_1 \oplus E_2$$

et : $R = \bar{E}_2 \cdot E_1$.

A ces deux relations correspondent les opérateurs OU EXCLUSIF, NON et ET d'où l'implantation de la figure 8.

Principe (fig. 9)

Les variables de sortie S et R_n correspondent au résultat de la différence des trois variables d'entrée : $E_2 - (E_1 + R_{n-1})$, la variable de sortie R_n étant toujours un éventuel report.

Tables logiques

Table de vérité

E_1	E_2	R_{n-1}	S	R_n
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Tables de Karnaugh

Table de la variable S

$E_1 E_2$		R_{n-1}	
		0	1
0 0	0 1	0	1
0 1	1 0	1	0
1 1	0 1	1	0
1 0	1 0	1	0

Table de la variable R_n

$E_1 E_2$		R_{n-1}	
		0	1
0 0	0 1	0	1
0 1	0 0	0	0
1 1	0 1	0	1
1 0	1 1	1	1

Équations booléennes

A partir des tables de Karnaugh précédentes nous obtenons les équations booléennes suivantes :

$$S = (E_1 \cdot \bar{E}_2 + \bar{E}_1 \cdot E_2) \cdot \bar{R}_{n-1} + (\bar{E}_1 \cdot \bar{E}_2 + E_1 \cdot E_2) \cdot R_{n-1}$$

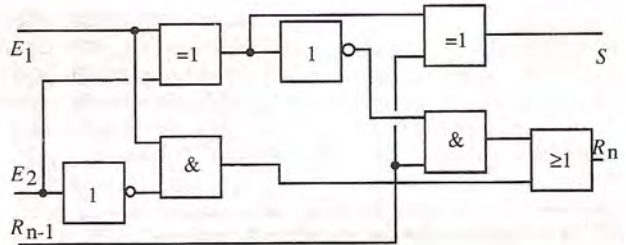
$$\text{et : } R_n = \bar{E}_2 \cdot E_1 + (\bar{E}_1 \cdot \bar{E}_2 + E_1 \cdot E_2) \cdot R_{n-1}$$

$$\text{Soit encore : } S = E_1 \oplus E_2 \oplus R_{n-1}$$

$$R_n = \bar{E}_2 \cdot E_1 + E_1 \oplus E_2 \cdot R_{n-1}$$

Aux relations précédentes, correspondent les opérateurs OU EXCLUSIF, OU et ET, d'où l'implantation de la figure 10 (le soustracteur complet comprend deux demi-soustracteurs).

Fig. 10 : Exemple de soustracteur complet : $S = E_2 - E_1$.



5 Additionneur-Soustracteur

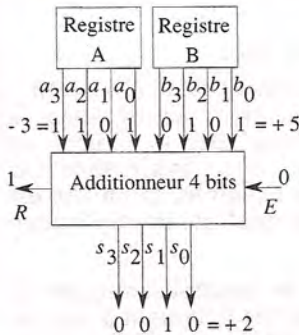
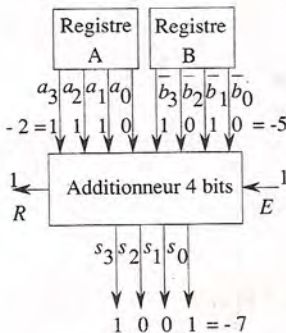


Fig. 11a : $-3 + 5 = +2$

$E = 0$: c'est une addition, on utilise les sorties normales du registre B : +5 est en écriture normale. -3 est remplacé par son complément à 2. Le résultat, +2 est en écriture normale.
 $R = 1$: dernière retenue à rejeter



Les figures 11a et 11b illustrent un exemple d'additionneur-soustracteur fonctionnant sur quatre bits utilisant des nombres signés dans la notation complément à 2 présentée au chapitre 64. Le nombre binaire $X = (a_3a_2a_1a_0)$ provient d'un registre A et l'autre nombre binaire $Y = (b_3b_2b_1b_0)$ provient d'un registre B (a_3 et b_3 sont des bits de signes).

Une entrée indique si l'opération à effectuer est une addition ($E = 0$) ou une soustraction ($E = 1$). Dans le cas d'une soustraction, le nombre à soustraire est complémenté, bit de signe compris : la soustraction se réduit alors à l'addition de deux nombres.

La figure 11a illustre l'addition : $-3 + 5 = +2$; la figure 11b illustre la soustraction : $-2 - 5 = -7$.

base10	base2	complément à 1	complément à 2
(+2)	= (0010)	= (0101)	= (0110)
(-2)	= (1010)	= (1101)	= (1110)
(+3)	= (0011)	= (0100)	= (0101)
(-3)	= (1011)	= (1100)	= (1101)
(+5)	= (0101)	= (0010)	= (0011)
(-5)	= (1101)	= (1010)	= (1011)
(+7)	= (0111)	= (0000)	= (0001)
(-7)	= (1111)	= (1000)	= (1001)

Fig. 11b : $-2 - 5 = -7$

$E = 1$: c'est une soustraction, on utilise les sorties complémentées du registre B (-5 est remplacé par son complément à 1, bit de signe non compris), et un 1 est ajouté à ce complément (entrée E) : cela revient à complémenter à 2 le nombre -5 puis à faire l'addition.

-2 est remplacé par son complément à 2 ainsi que le résultat, -7.
 $R = 1$: dernière retenue à rejeter.

Un multiplieur binaire permet de réaliser numériquement la multiplication de deux nombres binaires. Le premier est le multiplicande, $X = (a_3, a_2, a_1, a_0)$, le second est le multiplicateur $Y = (b_3, b_2, b_1, b_0)$. Le résultat de l'opération est noté $P = XY$.

1 Principe de la multiplication

Multiplication manuelle

Une multiplication manuelle, dans le système binaire, est une succession de décalages suivie d'une addition.

Exemple : $P = 1110 \times 1101$

Nous opérons comme dans le système à base 10. Pour effectuer l'opération, nous multiplions le multiplicande, $(1110)_2 = (14)_{10}$, par le chiffre de poids le plus faible du multiplicateur $(1101)_2 = (13)_{10}$. Dans notre exemple, ce chiffre est un 1 : le résultat est donc égal au multiplicande (fig. 1).

Ensuite nous multiplions le multiplicande par le chiffre du multiplicateur immédiatement à gauche du chiffre de poids le plus faible. Ce chiffre est un 0. Nous plaçons donc une ligne de 0 en la décalant d'un rang vers la gauche par rapport à la ligne écrite précédemment (fig. 2).

En remontant progressivement jusqu'au chiffre de poids le plus fort du multiplicateur, nous obtenons l'opération de la figure 3.

Le résultat de la multiplication est obtenu en faisant l'addition de toutes les lignes intermédiaires, comme indiqué figure 4.

```

1110  multiplicande
× 1101  multiplicateur
-----
1110  1er résultat partiel

```

Fig. 1

```

1110  multiplicande
× 1101  multiplicateur
-----
1110
0000

```

Fig. 2

```

1110  multiplicande
× 1101  multiplicateur
-----
1110
0000
1110
1110

```

Fig. 3

Fig. 4

```

1110  multiplicande
× 1101  multiplicateur
-----
1110
0000
1110
+ 1110
-----
10110110

```

$$(10110110)_2 = (182)_{10} = (14)_{10} \times (13)_{10}$$

Multiplication numérique

Dans l'exemple précédent, l'écriture des résultats intermédiaires se ramène à des décalages. Réaliser une multiplication nécessite donc l'utilisation de registres à décalage et d'additionneurs binaires.

Nous avons vu au chapitre 20 que les additionneurs binaires ne pouvaient traiter que deux nombres à la fois. Or l'algorithme que nous utilisons pour une multiplication manuelle nécessite l'addition de plus de deux nombres dès que le multiplicateur est formé de plus de deux chiffres. Il faut donc modifier quelque peu l'algorithme précédent pour le rendre compatible avec l'emploi d'additionneurs binaires.

Les résultats intermédiaires sont additionnés deux par deux et les sommes intermédiaires sont placées dans un registre. Il faut également utiliser le fait qu'additionner une ligne de zéros ne change rien au résultat final.

Reprenons l'exemple précédent (1110×1101) pour utiliser cet algorithme.

Le multiplicande est multiplié par le chiffre de poids le plus faible du multiplicateur. Nous réécrivons donc le multiplicande : 1110

En remontant vers le chiffre de poids le plus fort, nous rencontrons d'abord un zéro. L'addition d'une ligne de zéros étant inutile, il faut conserver le résultat précédent : 1110

Le troisième chiffre du multiplicateur qui est rencontré est 1 : il faut réécrire le multiplicande en le décalant de deux rangs vers la gauche (donc en ajoutant deux zéros à droite) : 111000.

Les deux nombres binaires sont alors additionnés :

$$\begin{array}{r} 1110 \\ + \quad 111000 \\ \hline 1000110 \end{array}$$

Le quatrième chiffre du multiplicateur est 1 : il faut réécrire le multiplicande en le décalant de trois rangs vers la gauche (et en ajoutant trois zéros à droite) : 1110000.

Ce nombre est additionné au résultat de l'addition binaire précédente :

$$\begin{array}{r} 1000110 \\ + \quad 1110000 \\ \hline 10110110 \end{array}$$

Le résultat final est le produit cherché.

2 Généralisation

Multiplication binaire :

	a_3	a_2	a_1	a_0	$\leftarrow X$	multiplicande			
$\times$	b_3	b_2	b_1	b_0	$\leftarrow Y$	multiplieur			
	c_3	c_2	c_1	c_0	$\leftarrow R_1$				
	d_3	d_2	d_1	d_0	0	$\leftarrow R_2$ résultats			
	e_3	e_2	e_1	e_0	0	0 $\leftarrow R_3$ intermédiaires			
	f_3	f_2	f_1	f_0	0	0 0 0 $\leftarrow R_4$			
	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	$\leftarrow P$ produit XY

Notons que les résultats intermédiaires R_1, R_2, R_3 et R_4 sont obtenus en multipliant un multiplicande noté X_0, X_1, X_2 ou X_3 obtenu à partir du multiplieur X décalé respectivement de 0, 1, 2 ou 3 rangs vers la gauche, par un chiffre du multiplieur, respectivement b_0, b_1, b_2, b_3 (en rouge ci-après) :

$$\begin{array}{ccccccc}
 a_3 a_2 a_1 a_0 \leftarrow X_0 & a_3 a_2 a_1 a_0 0 \leftarrow X_1 & a_3 a_2 a_1 a_0 0 0 \leftarrow X_2 & a_3 a_2 a_1 a_0 0 0 0 \leftarrow X_3 \\
 \times b_3 b_2 b_1 b_0 & \times b_3 b_2 b_1 b_0 & \times b_3 b_2 b_1 b_0 & \times b_3 b_2 b_1 b_0 \\
 \hline
 c_3 c_2 c_1 c_0 \leftarrow R_1 & d_3 d_2 d_1 d_0 0 \leftarrow R_2 & e_3 e_2 e_1 e_0 0 0 \leftarrow R_3 & f_3 f_2 f_1 f_0 0 0 0 \leftarrow R_4
 \end{array}$$

Un résultat intermédiaire est donc égal à :

$$R_{n+1} = X_n \cdot b_n$$

D'où, si :

$$b_n = 1; R_{n+1} = X_n; \text{ si } b_n = 0; R_{n+1} = 0$$

Le produit XY est égal à : $P = R_1 + R_2 + R_3 + R_4$.

Dans l'exemple précédent, $b_1 = 0$, d'où :

$$P = X_0 + X_2 + X_3$$

Ainsi après avoir comparé les valeurs des chiffres b_n à 0, la multiplication se ramène à une suite d'additions réalisées par des additionneurs à deux entrées qui effectuent les calculs suivants :

$$P_1 = R_1$$

$$P_2 = P_1 + R_2 = R_1 + R_2$$

$$P_3 = P_2 + R_3 = R_1 + R_2 + R_3$$

$$P_4 = P_3 + R_4 = R_1 + R_2 + R_3 + R_4$$

Soit :

$$P_{n+1} = P_n + R_{n+1}$$

Notons que :

$$\text{si } b_n = 1; P_{n+1} = P_n + X_n$$

$$\text{si } b_n = 0; P_{n+1} = P_n$$

3 Structure d'un multiplieur

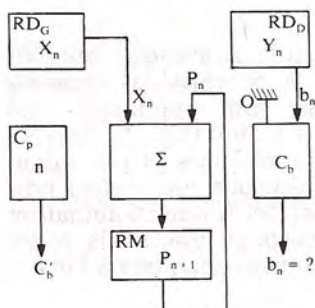


Fig. 5. Principe de la structure matérielle d'un multiplieur.

Un circuit multiplieur comprend (fig. 5) :

- un addresseur Σ qui calcule les termes P_{n+1} ,
- un registre mémoire RM pour le résultat de la multiplication,
- un registre à décalage à gauche RD_G qui fournit le multiplicande X_n ,
- un registre à décalage à droite RD_D qui fournit le multiplicateur Y_n ,
- un comparateur binaire C_b qui teste la valeur de b_n ,
- un compteur C_p qui enregistre le nombre n de pas de décalage,
- un deuxième comparateur C'_b qui teste la fin des opérations.

La séquence des opérations pour obtenir le produit XY est présentée sous forme d'organigramme (fig. 6).

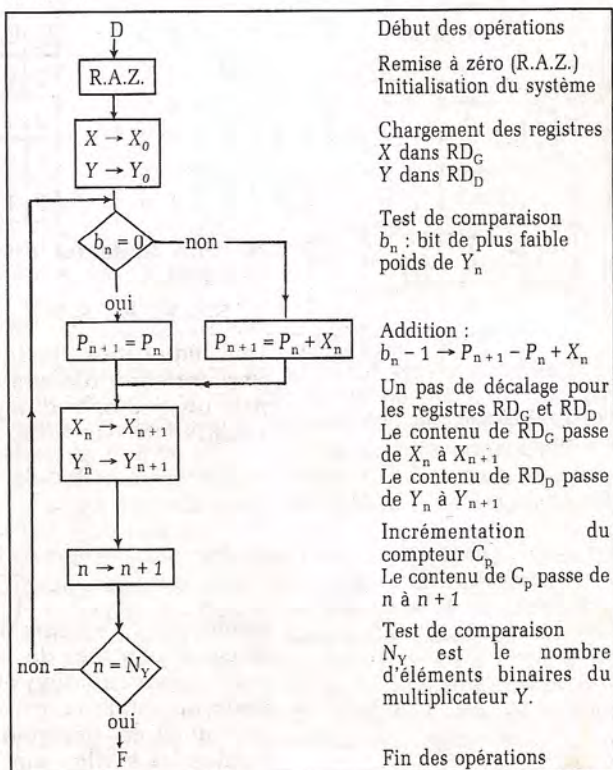


Fig. 6.

1 Décodeur binaire

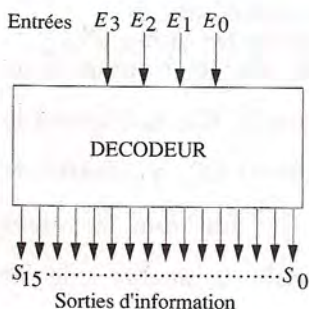


Fig. 1 : Schéma de principe d'un décodeur; exemple : quatre entrées et seize sorties.

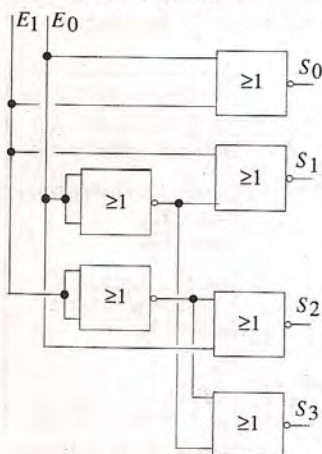


Fig. 2 : Décodeur binaire à deux entrées.

Description

Un décodeur est un circuit numérique qui fait correspondre 2^n sorties à n variables binaires d'entrées, sur lesquelles sont appliquées des variables binaires égales à 1 ou 0 (fig. 1).

Sauf pour quelques cas particuliers, il y a autant de sorties que de combinaisons (ou codes) possibles des n variables d'entrée. A une combinaison possible des entrées correspond une seule sortie active. Deux sorties ne peuvent donc être à l'état 1 en même temps.

Décodeur à deux entrées binaires

Aux deux variables d'entrées E_0, E_1 ce décodeur fait correspondre quatre variables de sortie S_0, S_1, S_2, S_3 . Sa table de vérité est la suivante :

E_1	E_0	S_3	S_2	S_1	S_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

A cette table on associe les équations logiques suivantes :

$$S_0 = \bar{E}_1 \cdot \bar{E}_0; S_1 = \bar{E}_1 \cdot E_0; S_2 = E_1 \cdot \bar{E}_0; S_3 = E_1 \cdot E_0$$

Des équations logiques, nous déduisons une implantation d'opérateurs logiques. Nous avons pris un exemple d'implantation utilisant des opérateurs NON-OU (fig. 2) :

$$S_0 = \overline{E_1 + E_0} = \bar{E}_1 \cdot \bar{E}_0$$

$$S_1 = \overline{E_1 + \bar{E}_0} = \bar{E}_1 \cdot E_0$$

$$S_2 = \overline{\bar{E}_1 + E_0} = E_1 \cdot \bar{E}_0$$

$$S_3 = \overline{\bar{E}_1 + \bar{E}_0} = E_1 \cdot E_0$$

Remarque : Certains décodeurs présentent une ou plusieurs entrées de validation; par exemple, une entrée de validation au niveau 1 permet au décodeur de fonctionner normalement; si elle est au niveau 0 le décodeur est bloqué au niveau 0 (toutes les sorties sont à l'état 0).

2 Décodeur DCB-7segments

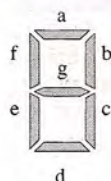


Fig. 3 : Afficheur 7 segments.

Fig. 4 : Segments apparents pour l'affichage des chiffres de 0 à 9.

Principe

La plupart des afficheurs numériques (montres, calculatrices, multimètres numériques...) permettent de représenter dix chiffres (de 0 à 9) et des lettres de l'alphabet. Ces symboles sont dessinés au moyen de sept segments lumineux (fig. 3) qui sont des diodes électroluminescentes ou des cristaux liquides. Pour l'affichage (fig. 4), certains de ces sept segments ne sont pas apparents.

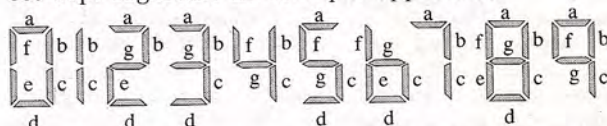


Fig. 5 : Table de vérité.

Chiffre décimal	Entrées D C B A				Sorties a b c d e f g							Afficheur
	D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	← 0
1	0	0	0	1	0	1	1	0	0	0	0	← 1
2	0	0	1	0	1	1	0	1	1	0	1	← 2
3	0	0	1	1	1	1	1	1	0	0	1	← 3
4	0	1	0	0	0	1	1	0	0	1	1	← 4
5	0	1	0	1	1	0	1	1	0	1	1	← 5
6	0	1	1	0	0	0	1	1	1	1	1	← 6
7	0	1	1	1	1	1	1	0	0	0	0	← 7
8	1	0	0	0	1	1	1	1	1	1	1	← 8
9	1	0	0	1	1	1	1	0	0	1	1	← 9

La figure 5 donne la table de vérité détaillant le fonctionnement du décodeur permettant un affichage des différents chiffres. Les variables d'entrée sont écrites en DCB : elles sont à décoder. Dans les colonnes de sortie le chiffre 1 indique un segment qui doit être allumé et le chiffre 0 un segment qui doit être éteint.

Exemple : pour afficher le chiffre 6, il faut éteindre les segments **a** et **b**, les autres étant allumés.

Pour écrire les équations logiques de ce décodeur, il faut construire les tables de Karnaugh correspondant à tous les segments. Il y en a donc sept. Dans ces tables, les 0 étant moins nombreux que les 1, il est préférable d'établir les équations correspondant à l'extinction des segments.

BA \ DC	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	x	x	x	x
10	1	1	x	x

Fig. 6 : Table de Karnaugh du segment **b** :

$$\bar{b} = A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

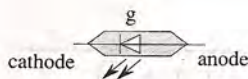


Fig. 8 : Segment à diode électroluminescente.

Certains décodeurs n'utilisent pas toutes les combinaisons possibles : les codes inutilisés appliqués à l'entrée n'activent aucune sortie.

Certains décodeurs intégrés ont leurs sorties actives à l'état bas, d'autres à l'état haut.

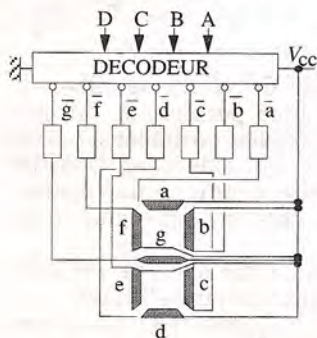


Fig. 9 : Sur cet exemple, les sorties du décodeur sont actives à l'état bas (d'où la présence du cercle indiquant la complément en sortie).

Dans la marge, nous donnons un exemple correspondant au segment **b** (fig. 6). A partir de l'équation logique correspondante, nous en déduisons une implantation d'opérateurs logiques (fig. 7). Une implantation complète nécessite le même travail pour tous les segments : le nombre important d'opérateurs logiques explique l'emploi de circuits intégrés.

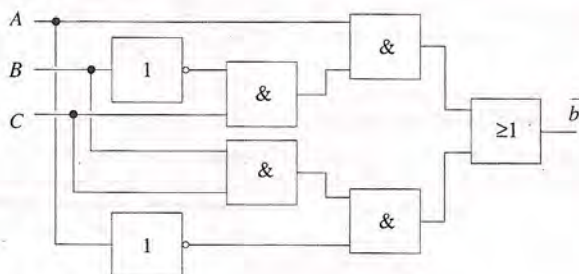


Fig. 7 : Élément d'un décodeur 7 segments commandant l'affichage du segment **b**.

Afficheurs

-Afficheur à diodes électroluminescentes (DEL)

Chaque segment est formé d'une ou deux diodes électroluminescentes (fig. 8). Toutes les anodes des segments sont reliées à l'alimentation (+ 5 V en général). Les cathodes sont connectées aux sorties du décodeur par l'intermédiaire de résistances limitant l'intensité du courant traversant les DEL à 20 mA environ (fig. 9).

-Afficheur à cristaux liquides

Un cristal liquide est un liquide dont certaines propriétés physiques changent de façon spectaculaire lorsque sa structure moléculaire est modifiée. Les molécules qui constituent ces cristaux sont très sensibles aux champs électriques et aux champs magnétiques.

Les afficheurs à cristaux liquides utilisent l'action d'un champ électrique qui modifie, entre autres, les propriétés optiques du cristal : tout en étant éclairé de l'extérieur, il apparaît « allumé » en présence de ce champ ou « éteint » en son absence (un cristal liquide n'émet pas de lumière).

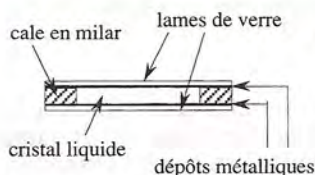


Fig. 10 : Afficheur à cristaux liquides (coupe).

Exemple : table de vérité de l'opérateur OU Exclusif alimentant le segment noté a sur la figure 11. Les variables d'entrée sont notées a et E, la variable de sortie est notée S.

a	E	S
0	0	0
0	1	1
1	0	1
1	1	0

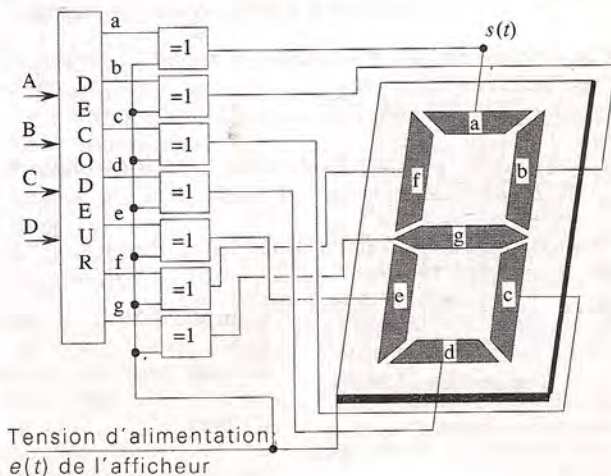
Fig. 11 : Commande d'un afficheur 7 segments à cristaux liquides.

Dans le cas d'un afficheur à sept segments, le cristal liquide est enfermé dans une capsule en verre (fig. 10). La face arrière comporte un dépôt métallique ayant la forme d'une plaque. Le dépôt métallique de la face avant a la forme des 7 segments qui seront visibles. La plaque et chaque segment constituent les armatures d'un condensateur. En appliquant une tension variable (de 4 V à 20 V en valeur maximale), de basse fréquence (entre 20 Hz et 60 Hz), entre ces armatures, le segment devient visible.

Dans la pratique, pour qu'un segment devienne visible, on applique entre, d'une part la plaque et la masse, d'autre part le segment et la masse, des tensions en créneaux décalées dans le temps. Pour effectuer ce décalage, on intercale un ensemble d'opérateurs OU EXCLUSIF entre le décodeur et les afficheurs (fig. 11).

Lorsqu'une sortie du décodeur est à zéro, la sortie de l'opérateur qui lui est associé reproduit exactement la tension en créneaux : la tension entre le segment correspondant et la plaque est nulle, le segment n'est pas visible.

Lorsqu'une sortie du décodeur est à l'état 1, la sortie de l'opérateur qui lui est associé reproduit une tension en créneaux mais décalée d'une demi-période par rapport à la tension d'entrée : la tension entre le segment correspondant et la plaque est à chaque instant différente de zéro, le segment est apparent.



1 Codeur binaire

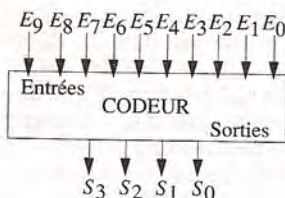


Fig. 1 : Schéma de principe d'un codeur; exemple : dix entrées et quatre sorties.

Définition

Un codeur binaire est un circuit numérique à N voies d'entrée et n sorties. Ce circuit doit faire correspondre une seule voie d'entrée à l'ensemble des sorties simultanément pour la représentation d'un mot binaire sur n bits (fig. 1).

Codage des chiffres décimaux 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9 en binaire naturel

Les dix variables d'entrée de ce codeur $E_0, E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$ et E_9 correspondent respectivement aux chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. Lorsque l'une de ces entrées est activée, c'est-à-dire placée dans l'état 1, cela revient à obtenir à la sortie du codeur l'équivalent binaire du chiffre décimal correspondant à cette entrée.

Les mots étant définis sur quatre bits, le codeur doit disposer de quatre sorties S_0, S_1, S_2, S_3 .

Utilisons la table de vérité (fig. 2) de ce codeur.

Fig. 2 : Table de vérité d'un codeur décimal binaire.

Dans ce tableau, nous avons isolé le chiffre décimal 3 qui correspond à l'entrée E_3 . Activer E_3 , revient à réaliser $E_3 = 1$ dans le but d'obtenir l'équivalent binaire du chiffre décimal 3, soit 11. Les variables de sortie S_0, S_1, S_2 et S_3 seront respectivement égales à 0, 0, 1 et 1.

Chiffre décimal	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	S_3	S_2	S_1	S_0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
→ 3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Les équations logiques associées aux sorties sont :

$$\begin{aligned}
 S_0 &= E_1 + E_3 + E_5 + E_7 + E_9 \\
 S_1 &= E_2 + E_3 + E_6 + E_7 \\
 S_2 &= E_4 + E_5 + E_6 + E_7 \\
 S_3 &= E_8 + E_9
 \end{aligned}$$

L'implantation d'opérateurs logiques accompagnant ces équations (fig. 3) est simple. Sur le schéma proposé, l'entrée E_0 n'est pas reliée car les sorties affichent 0000 lorsque toutes les autres entrées sont à l'état 0.

Ce codeur ne fonctionne convenablement que si une seule entrée est activée à la fois.

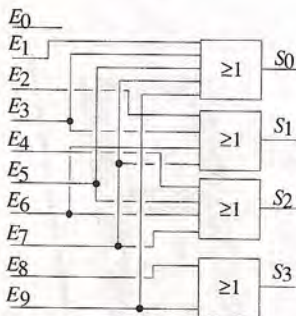


Fig. 3 : Codeur décimal binaire. L'entrée E_0 n'est reliée à aucun opérateur car lorsque toutes les autres entrées sont à 0, les sorties sont aussi à 0.

2 Codeur de priorité

Inconvénient d'un codeur binaire

Dans l'exemple précédent, activer deux entrées simultanément donne en sortie un résultat de codage erroné.

Exemple : Activons l'entrée E_2 seule, nous avons en sortie : 0010.

Avec l'entrée E_4 seule, le code de sortie serait : 0100.

Portons simultanément à l'état 1 les entrées E_2 et E_4 : en sortie nous obtenons le mot binaire : 0110, ce qui correspond au code binaire de l'entrée E_6 . Pour éviter ces erreurs, il faut utiliser un codeur de priorité, parfois appelé **encodeur prioritaire**.

E_0	E_1	E_2	E_3	S'_1	S'_0
1	0	0	0	0	0
x	1	0	0	0	1
x	x	1	0	1	0
x	x	x	1	1	1

Fig. 4 : Table de vérité d'un codeur de priorité.

$E_2 E_3$		00	01	11	10
E_1	0	0	1	1	0
	1	1	1	1	0

Fig. 5 : Table de Karnaugh de la sortie S'_0 d'un codeur de priorité 4 entrées/2 sorties :

$$S'_0 = E_1 \cdot \bar{E}_2 + E_3$$

$E_2 E_3$		00	01	11	10
E_1	0	0	1	1	1
	1	0	1	1	1

Fig. 6 : Table de Karnaugh de la sortie S'_1 d'un codeur de priorité 4 entrées/2 sorties : $S'_1 = E_2 + E_3$.

Principe d'un codeur de priorité

C'est un codeur binaire qui permet le choix du code correspondant au chiffre décimal le plus élevé des entrées activées.

Avec un codeur de priorité, l'activation simultanée des entrées E_2 et E_4 , comme dans l'exemple précédent, se traduit par l'obtention en sortie de 0100. C'est l'équivalent binaire du nombre correspondant à l'entrée E_4 : il représente le chiffre 4 ($4 > 2$, 2 étant chiffre décimal associé à E_2).

Il en est de même lorsque plus de deux entrées sont activées : avec un codeur de priorité, l'activation simultanée des entrées E_2 , E_4 et E_7 conduit à l'obtention, en sortie, de l'équivalent binaire du nombre correspondant à l'entrée E_7 .

Exemple : Codeur de priorité à quatre entrées et deux sorties

Ce codeur doit répondre aux conditions de fonctionnement suivant :

- Pour l'entrée E_0 mise dans l'état 1 et toutes les autres mises à l'état 0, les sorties doivent indiquer : 00.
- Les entrées E_2 et E_3 étant placées dans l'état 0, pour obtenir 01 en sortie, il faut activer l'entrée E_1 et ceci quel que soit l'état de l'entrée E_0 .
- L'entrée E_3 étant mise dans l'état 0, pour obtenir 10 en sortie, il faut activer l'entrée E_2 et ceci quels que soient les états des entrées E_0 et E_1 .
- Pour obtenir 11 en sortie, il faut activer l'entrée E_3 et ceci quels que soient les états des entrées E_0 , E_1 et E_2 .

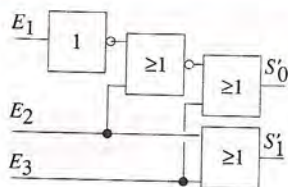


Fig. 7 : Codeur de priorité 4 entrées/2 sorties.

La table de vérité de la figure 4 résume ces conditions de fonctionnement.

A partir des tables de Karnaugh des variables S'_0 et S'_1 (fig. 5 et 6), nous en déduisons les équations logiques de S'_0 et S'_1 :

$$S'_0 = E_1 \cdot \bar{E}_2 + E_3 \quad \text{et} \quad S'_1 = E_2 + E_3$$

L'implantation d'opérateurs logiques de la figure 7 est un exemple de réalisation possible. L'entrée E_0 ne figure pas sur le schéma proposé car les sorties affichent 00 lorsque les entrées E_1 , E_2 et E_3 sont simultanément placées dans l'état 0.

3 Application : affichage d'un nombre décimal à l'écran d'une calculatrice

Nous désirons l'affichage du nombre 52. Nous travaillons à partir du schéma de la figure 8.

L'initialisation des bascules (circuit non représenté sur la figure) se fait en agissant sur leurs entrées de prépositionnement. Elle est telle que les bascules D du registre mémoire sont à zéro et que le contenu du compteur en anneau corresponde au nombre 01 (sortie Q de la première bascule JK à l'état 0 et sortie Q de la seconde à l'état 1).

- En appuyant sur la touche 5 de la calculatrice, l'entrée E_5 du codeur de priorité décimal-DCB est placée dans l'état 1, ce qui donne en sortie du codeur la combinaison 0101 (code DCB de 5).

Cette combinaison est appliquée :

- aux entrées parallèles des huit bascules D constituant un registre mémoire ;
- ainsi qu'aux entrées d'un opérateur OU dont la sortie, initialement placée dans l'état 0, passe à l'état 1, ce qui a pour effet de changer l'état des sorties des bascules JK du compteur : la sortie de la première passe de l'état 0 à l'état 1 et celle de la seconde de l'état 1 à l'état 0 (l'indication du compteur est alors 10). Les sorties du compteur étant reliées aux entrées d'horloge des huit bascules D actives sur un front montant (passage de l'état 0 à l'état 1), seules les quatre premières bascules sont soumises à un front montant : elles transmettent la combinaison 0101 au premier décodeur puis à l'afficheur correspondant, ce dernier indiquant le chiffre 5.

- En appuyant sur la touche 2, l'entrée E_2 du codeur est mise à l'état 1, ce qui donne en sortie du codeur la combinaison 0010 (code DCB de 2).

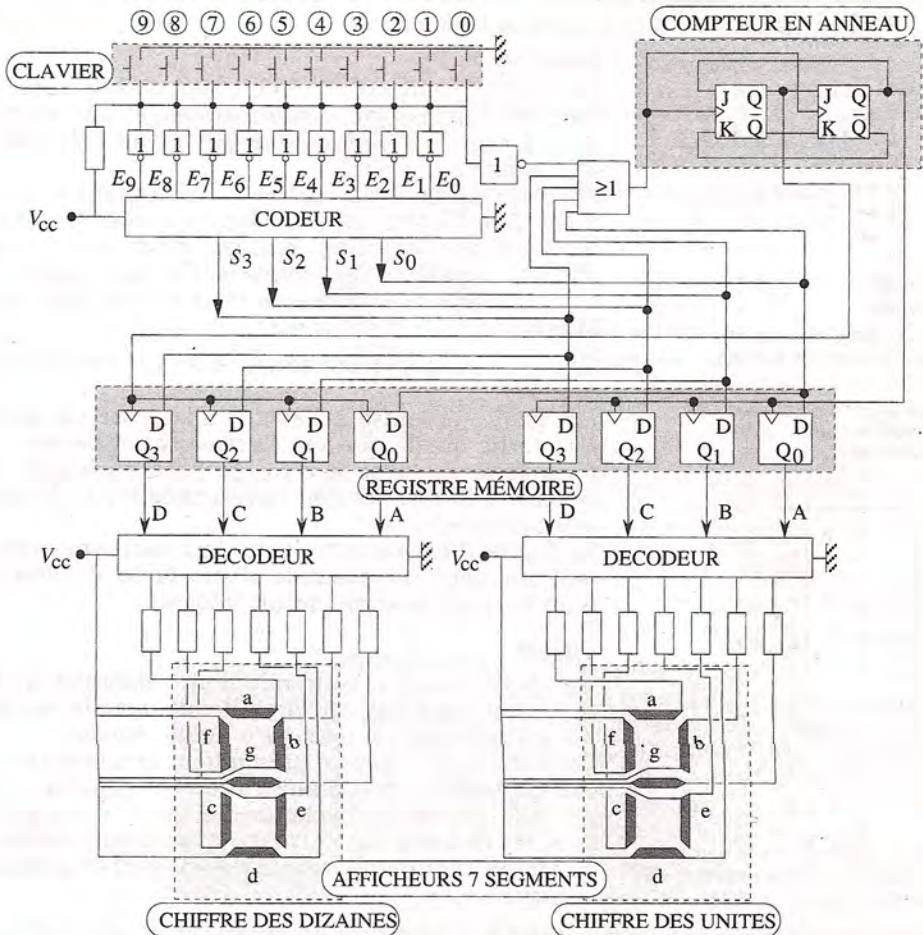


Fig. 8 : Affichage d'un nombre décimal de deux chiffres, entré au clavier d'une calculatrice.

Cette combinaison est appliquée :

- aux entrées des huit bascules D;
- ainsi qu'aux entrées de l'opérateur OU dont la sortie revenue à l'état 0 (du fait du relâchement de la touche 5) repasse à l'état 1; le compteur indique maintenant 1 (la première bascule passe de l'état 1 à l'état 0 et la seconde de l'état 0 à l'état 1); seules les quatre bascules D précédemment inactives sont soumises à un front montant du signal d'horloge : elles transmettent la combinaison 0010 au deuxième décodeur puis à l'afficheur qui lui est associé, ce dernier indiquant le chiffre 2.

A la fin de cette séquence, nous lisons le nombre 52.

1 Présentation

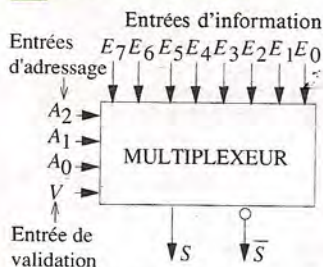


Fig. 1. Multiplexeur à 8 entrées d'information et 3 entrées d'adressage.

Ces entrées d'adresse s'appellent aussi entrées d'affichage.

V	A ₂	A ₁	A ₀	$\bar{S}$	S
H	X	X	X	H	L
L	L	L	L	$\bar{E}_0$	E ₀
L	L	L	H	$\bar{E}_1$	E ₁
L	L	H	L	$\bar{E}_2$	E ₂
L	L	H	H	$\bar{E}_3$	E ₃
L	H	L	L	$\bar{E}_4$	E ₄
L	H	L	H	$\bar{E}_5$	E ₅
L	H	H	L	$\bar{E}_6$	E ₆
L	H	H	H	$\bar{E}_7$	E ₇

Fig. 2. Table de vérité d'un multiplexeur à 8 entrées (L : low pour 0, H : high pour 1).

Description

Un multiplexeur est un circuit logique à N entrées dites de données (ou d'informations) et une sortie qui transmet les informations parvenant à l'entrée sélectionnée (fig. 1).

Pour sélectionner une entrée, le multiplexeur doit recevoir un ordre qui provient de n entrées complémentaires appelées entrées d'adresse : ces entrées aiguillent les données d'entrée choisies vers la sortie. Les nombres N et n sont liés par la relation : $N = 2^n$

Une entrée supplémentaire, dite de validation, permet :

- lorsqu'elle est mise à l'état 1, d'imposer un zéro à la sortie quelle que soit l'entrée sélectionnée ;
- lorsqu'elle est mise à zéro, de faire apparaître à la sortie l'état de l'entrée correspondant à l'adresse sélectionnée.

La figure 2 est un extrait de document constructeur donnant un exemple d'une table de vérité d'un multiplexeur en circuit intégré.

Intérêt

- Un multiplexeur joue le rôle d'un commutateur à plusieurs positions qui aiguillerait vers la sortie les informations de n'importe quelle entrée.
- Les entrées d'adressage permettent de connaître à chaque instant l'état d'une entrée de données.
- Par une utilisation cyclique des entrées d'adressage, les données des entrées, qui sont en parallèle à l'entrée du multiplexeur, se retrouvent en série à la sortie.

2 Exemples de réalisation

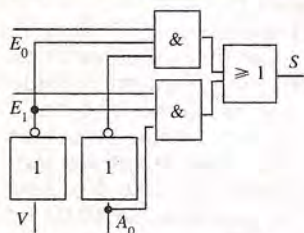


Fig. 3. Multiplexeur à 2 entrées de données, une entrée d'adresse et une entrée de validation.

Multiplexeur élémentaire à deux entrées

C'est un multiplexeur à deux entrées de données, donc à une entrée d'adressage ($N = 2$ et $n = 1$). Le circuit proposé (fig. 3) comporte cinq opérateurs logiques : deux inverseurs, deux opérateurs ET et un opérateur OU.

La valeur logique 1 ou 0 donnée à la variable d'adresse A_0 permet de choisir l'opérateur ET a ou ET b. Celui qui est passant transmet les données qui figurent à son entrée à l'opérateur OU pour qu'elles soient disponibles à la sortie S.

L'équation logique associée à ce montage est :

• $A_0 = 0$

$$S = E_0 \cdot \bar{A}_0 + E_1 \cdot A_0$$

L'équation précédente se réduit à : $S = E_0 \cdot 1$.

Donc : $S = E_0$.

Faire $A_0 = 0$ revient à sélectionner l'entrée E_0 : toutes les données logiques appliquées en E_0 se retrouvent à la sortie S du multiplexeur.

• $A_0 = 1$

L'équation précédente s'écrit : $S = E_0 \cdot 0 + E_1 \cdot 1$.

Soit : $S = E_1$.

Faire $A_0 = 1$ revient à sélectionner l'entrée E_1 : toutes les données logiques appliquées en E_1 se retrouvent à la sortie S du multiplexeur.

Nous pouvons regrouper les résultats dans le tableau suivant :

$n = 1$	$N = 2$		
A_0	E_1	E_0	S
0		$\rightarrow E_0$	
1		$\rightarrow E_1$	

Multiplexeur à quatre entrées

C'est un multiplexeur à quatre entrées de données, donc à deux entrées d'adressage ($N = 4$ et $n = 2$).

Le circuit proposé (fig. 4) comporte deux inverseurs, quatre opérateurs ET à trois entrées et un opérateur OU à quatre entrées.

L'équation logique associée à ce montage est :

$$S = E_0 \cdot \bar{A}_1 \cdot \bar{A}_0 + E_1 \cdot \bar{A}_1 \cdot A_0 + E_2 \cdot A_1 \cdot \bar{A}_0 + E_3 \cdot A_1 \cdot A_0$$

La table de vérité est donnée dans la marge.

$n = 2$		$N = 4$				
A_1	A_0	E_3	E_2	E_1	E_0	S
0	0				$\rightarrow E_0$	
0	1			$\rightarrow E_1$		
1	0		$\rightarrow E_2$			
1	1	$\rightarrow E_3$				

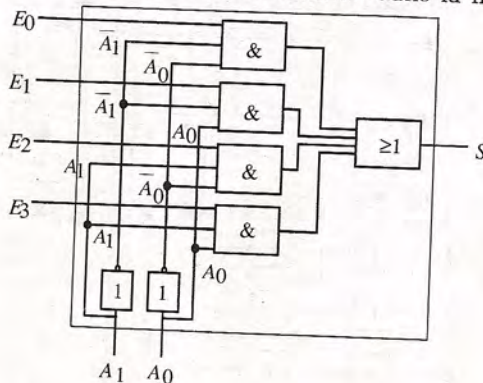


Fig. 4. Multiplexeur à 4 entrées.

La sortie S reproduit le niveau logique de la ligne de données sélectionnée par l'adresse A_0A_1 .

La combinaison $A_0 = 0, A_1 = 0$ permet de sélectionner l'entrée E_0 .

La combinaison $A_0 = 1, A_1 = 0$ permet de sélectionner l'entrée E_1 .

La combinaison $A_0 = 0, A_1 = 1$ permet de sélectionner l'entrée E_2 .

La combinaison $A_0 = 1, A_1 = 1$ permet de sélectionner l'entrée E_3 .

Multiplexeur avec entrée de validation et sortie complémentée

Ces multiplexeurs se présentent sous la forme de circuits intégrés (CMOS ou TTL). Ils sont disponibles en deux, quatre, huit ou seize entrées. Ils peuvent être combinés afin de pouvoir multiplexer un nombre d'entrées encore plus grand.

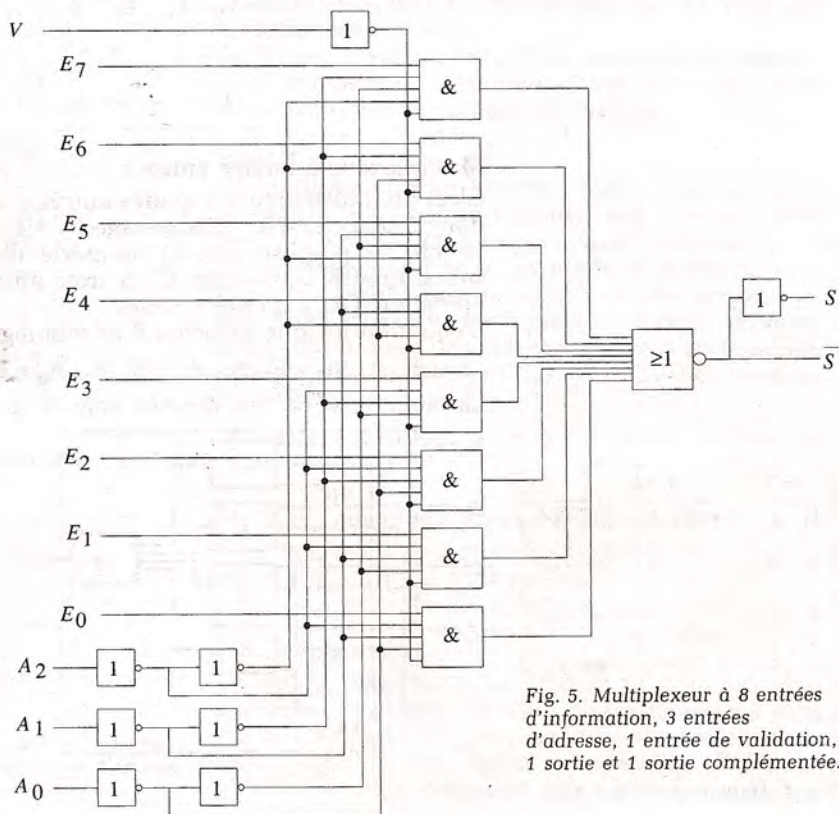


Fig. 5. Multiplexeur à 8 entrées d'information, 3 entrées d'adresse, 1 entrée de validation, 1 sortie et 1 sortie complémentée.

La figure 5 donne le synoptique interne d'un multiplexeur à huit entrées avec entrée de validation et sortie complémentée. Le tableau de la figure 2 illustre le fonctionnement d'un tel multiplexeur.

3 Applications

Code binaire naturel

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

Code binaire réfléchi

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

Fig. 6. Transcodage «code binaire naturel»/«code binaire réfléchi» : les données sont en parallèle à l'entrée, elles se retrouvent en série à la sortie.

Les multiplexeurs sont utilisés dans l'aiguillage des données, dans l'ordonnancement d'opérations, dans la conversion parallèle-série, dans la génération de fonctions logiques, ...

Prenons un exemple d'application : transcodage du «code binaire naturel» en «code binaire réfléchi». Nous nous limitons à trois bits.

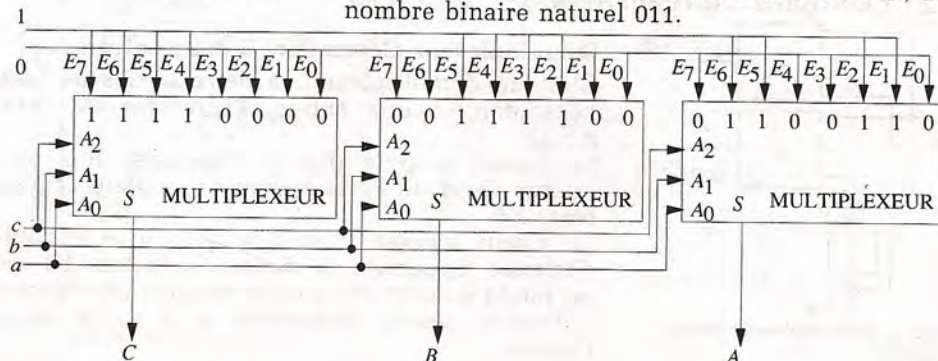
Il faut trois multiplexeurs dont les variables de sortie seront A, B et C. Le nombre cba pouvant prendre huit valeurs différentes, chaque multiplexeur sera muni de huit entrées d'information donc de trois entrées d'adressage. Il faudra appliquer à ces entrées d'aiguillage, les valeurs logiques de a, b et c. Le montage de la figure 6 donne la solution de ce problème.

Pour le transcodage de cba = 011, l'entrée E₃ est sélectionnée sur chaque multiplexeur :

- A reproduit la valeur de la variable d'entrée E₃ du troisième, soit 0;
- B reproduit la valeur de la variable d'entrée E₃ du deuxième, soit 1;
- C reproduit la valeur de la variable d'entrée E₃ du premier, soit 0.

A la sortie du montage, nous avons alors la combinaison : CBA = 010

010 est l'écriture, dans le code binaire réfléchi, du nombre binaire naturel 011.



1 Présentation

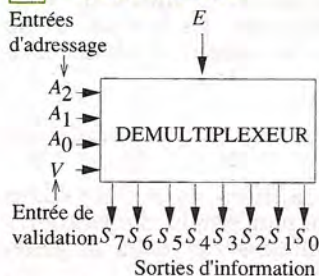


Fig. 1. Démultiplexeur à 8 entrées d'information et 3 entrées d'adressage.

Description

Un démultiplexeur est un circuit logique à une entrée de données ou d'informations et N sorties qui reçoivent les informations d'entrée (fig. 1).

Pour sélectionner la sortie qui doit être active, le démultiplexeur reçoit un ordre de n entrées d'adresse, comme dans le cas d'un multiplexeur. Ces entrées orientent les données d'entrée vers la sortie sélectionnée. Les nombres N et n sont liés par la relation : $N = 2^n$

Certains démultiplexeurs présentent une entrée de validation qui permet :

- lorsqu'elle est mise à l'état 0, d'imposer un zéro à l'ensemble des sorties quelle que soit la commande ;
- lorsqu'elle est mise à l'état 1, d'autoriser la transmission des données de l'entrée vers les sorties successivement sélectionnées.

Intérêt

- Un démultiplexeur joue le rôle d'un commutateur à plusieurs positions qui oriente les informations d'entrée vers chaque sortie choisie.
- Les entrées d'adressage permettent de connaître à chaque instant l'état d'une sortie.
- Par une utilisation cyclique des entrées d'adressage, les données des entrées, qui parviennent en série à l'entrée du démultiplexeur, sont fournies en parallèle à la sortie.

2 Exemples de réalisation

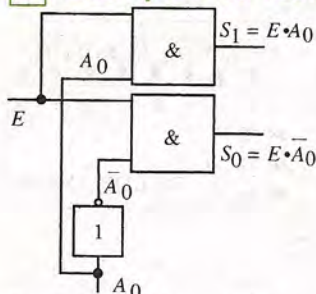


Fig. 2. Démultiplexeur à deux sorties.

Démultiplexeur élémentaire à deux entrées

C'est un démultiplexeur à deux sorties de données, donc à une entrée d'adressage ($N = 2$ et $n = 1$).

Le circuit proposé (fig. 2) comporte trois opérateurs logiques : un inverseur et deux opérateurs ET.

La valeur logique 1 ou 0 donnée à la variable d'adresse A_0 permet de choisir l'opérateur ET qui est rendu passant afin que les données qui figurent à l'entrée soient disponibles à la sortie sélectionnée.

Les équations logiques associées à ce montage sont : $S_0 = E \cdot \bar{A}_0$ et $S_1 = E \cdot A_0$

• $A_0 = 0$

Nous avons : $S_0 = E \cdot 1$ et $S_1 = E \cdot 0$

Soit : $S_0 = E$ et $S_1 = 0$

Faire $A_0 = 0$ revient à sélectionner la sortie S_0 : toutes les données logiques appliquées en E se retrouvent à la sortie S_0 du démultiplexeur.

• $A_0 = 1$

Nous avons : $S_0 = E \cdot 0$ et $S_1 = E \cdot 1$

Soit : $S_0 = 0$ et $S_1 = E$

Faire $A_0 = 1$ revient à sélectionner la sortie S_1 : toutes les données logiques appliquées en E se retrouvent à la sortie S_1 du démultiplexeur.

Nous pouvons regrouper les résultats dans le tableau suivant :

n=1			N=2
A ₀	E	E	S
0	└─┐	└─┐	$S_0 = E$
1		└─┐	$S_1 = E$

Démultiplexeur à quatre sorties

Le nombre de sorties N étant égal à 4, le nombre d'entrées d'adresse n est égal à 2.

Le montage proposé (fig. 3) comporte deux inverseurs et quatre opérateurs ET à trois entrées.

Les équations logiques associées à ce montage sont :

$$S_0 = E \cdot \bar{A}_1 \cdot \bar{A}_0; \quad S_1 = E \cdot \bar{A}_1 \cdot A_0;$$

$$S_2 = E \cdot A_1 \cdot \bar{A}_0; \quad S_3 = E \cdot A_1 \cdot A_0$$

La table de vérité est :

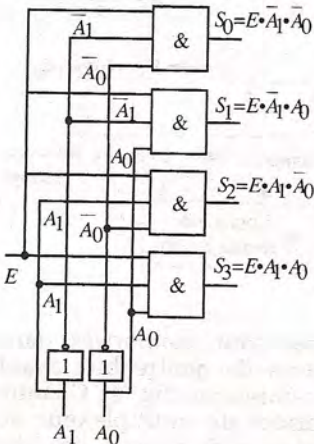


Fig. 3. Démultiplexeur à quatre sorties.

n=2						N=4
A ₁	A ₀	E	E	E	E	S
0	0	└─┐	└─┐	└─┐	└─┐	$S_0 = E$
0	1					$S_1 = E$
1	0					$S_2 = E$
1	1					$S_3 = E$

La combinaison $A_0 = 0, A_1 = 0$ permet de sélectionner la sortie S_0 : les informations d'entrée sont orientées vers la sortie S_0 .

La combinaison $A_0 = 1, A_1 = 0$ permet de sélectionner la sortie S_1 : les informations d'entrée sont orientées vers la sortie S_1 .

La combinaison $A_0 = 0, A_1 = 1$ permet de sélectionner la sortie S_2 : les informations d'entrée sont orientées vers la sortie S_2 .

La combinaison $A_0 = 1, A_1 = 1$ permet de sélectionner la sortie S_3 : les informations d'entrée sont orientées vers la sortie S_3 .

3 Application : transmission synchrone de données

Un système de transmission synchrone de données nécessite l'utilisation d'un multiplexeur et d'un démultiplexeur.

Le schéma de principe de la figure 4 illustre la transmission en série de quatre mots de quatre bits (seize bits en tout) d'un émetteur vers un récepteur.

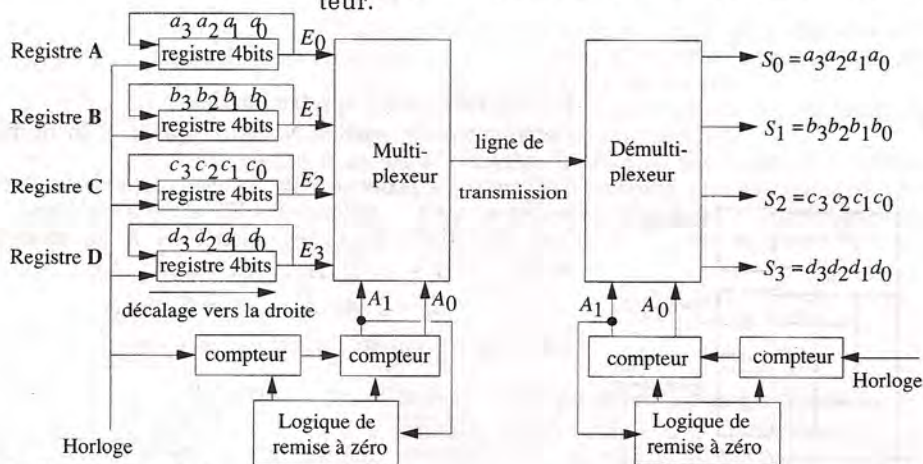


Fig. 4. Transmission de données.

Émetteur

Les quatre mots binaires sont mémorisés dans quatre registres à décalage de quatre bits ayant leurs entrées d'horloge communes (fig. 4). Chaque registre est relié aux entrées du multiplexeur de telle sorte que la transmission de chacun des bits se fasse en commençant par celui de poids le plus faible.

Deux compteurs commandent la transmission du contenu du registre sélectionné, contenu qui apparaît à la sortie du multiplexeur :

- l'un choisit le registre des données qui doivent être transmises à la sortie du multiplexeur ;
- l'autre contrôle que tous les bits de chaque registre ont été transmis avant de sélectionner le registre suivant : à chaque impulsion d'horloge, le compteur est incrémenté et le contenu du registre est décalé d'un rang vers la droite avec transfert vers la sortie du multiplexeur du bit le plus à droite. Après quatre impulsions, il est remis à zéro avant de passer au registre suivant : tous les bits du registre ont été transmis.

Ce sont donc des compteurs modulo 4 qui doivent être utilisés.

Chacun des seize bits est transmis, bit après bit, par ensemble de 4. Ces ensembles apparaissent à la sortie à des moments différents : les données sont dites multiplexées dans le temps.

Récepteur

Il est essentiellement formé d'un démultiplexeur dont le rôle est de séparer les quatre ensembles de données provenant de la sortie du multiplexeur afin de les orienter vers les quatre sorties du démultiplexeur. La logique de commande doit faire apparaître, bit après bit, le contenu :

- du registre A, à la sortie S_0 ,
- du registre B, à la sortie S_1 ,
- du registre C, à la sortie S_2 ,
- du registre D, à la sortie S_3 .

Comme pour l'émetteur, deux compteurs modulo 4 accompagnent le démultiplexeur :

- l'un sélectionne la sortie du démultiplexeur qui reçoit l'ensemble de quatre bits provenant, après multiplexage, de l'un des quatre registres à décalage ;
- l'autre contrôle que tous les bits de chaque ensemble ont été transmis avant de sélectionner la sortie suivante.

Remarque : En circuits intégrés, beaucoup de décodeurs peuvent être utilisés en démultiplexeurs et réciproquement : les entrées du décodeur sont des entrées d'adresse du démultiplexeur, l'entrée de validation du décodeur est l'entrée de donnée du démultiplexeur, les sorties étant les mêmes dans les deux cas.

1 Représentation d'une grandeur physique

La valeur numérique d'une grandeur physique peut être représentée soit de manière analogique, soit de manière numérique.

Représentation analogique

Prenons l'exemple du compte-tours d'une voiture. La rotation de l'aiguille au cours du temps se fait de façon **continue** : à une grandeur, la vitesse de rotation, correspond une autre grandeur, l'angle de rotation de cette aiguille, qui lui est proportionnelle. Cela caractérise la représentation analogique d'une grandeur.

Dans le cas d'un affichage numérique, les chiffres qui constituent l'heure évoluent de façon discontinue.

Représentation numérique

Prenons une montre à aiguilles à quartz et observons la rotation de la trotteuse : sa progression au cours du temps se fait de façon **discontinue** : à une grandeur, la durée, correspond une autre grandeur, l'angle de rotation, qui ne lui est plus proportionnelle ; chaque tour d'aiguille se fait par sauts : soixante sauts par tour pour l'aiguille indiquant les secondes. Cela caractérise la représentation numérique d'une grandeur.

2 Systèmes

Systèmes analogiques

La plupart des grandeurs qui doivent être mesurées (intensité, tension, température, pression, vitesse, ...) sont des grandeurs analogiques. Les systèmes analogiques couramment utilisés sont les oscilloscopes analogiques, les ampèremètres et voltmètres analogiques (progressivement remplacés par des appareils numériques), les récepteurs radiophoniques, les magnétophones, ...

Systèmes numériques

Dans ces systèmes les grandeurs physiques évoluent de manière discontinue. C'est le cas des ordinateurs, des calculatrices, des multimètres numériques, des machines-outils à commande numérique.

Les principaux avantages des systèmes numériques sont :

- d'être souvent programmables,
- d'être souvent plus précis et plus rapides que les systèmes analogiques,

- de pouvoir conserver en mémoire des données,
- d'être moins sensibles aux perturbations que les systèmes analogiques (pour certaines applications cela peut être un défaut).

Systèmes hybrides

En réalité, beaucoup de systèmes ne sont pas uniquement numériques.

Prenons l'exemple du voltmètre numérique. Il capte une tension, c'est-à-dire une grandeur analogique, et il affiche un nombre, c'est-à-dire une grandeur numérique, qui donne la mesure de cette tension. C'est un système que l'on peut qualifier d'hybride car des grandeurs analogiques et numériques interviennent simultanément dans cet appareil.

Dans des systèmes plus complexes, comme ceux utilisés en régulation, on capte une grandeur analogique (la vitesse par exemple) et elle est ensuite convertie en grandeur numérique par l'intermédiaire d'un convertisseur analogique-numérique. Cette information numérique est traitée par un ordinateur ou un automate qui renvoie, si nécessaire, une autre information numérique convertie ensuite au moyen d'un convertisseur numérique-analogique, en grandeur analogique. Cette dernière agit sur un régulateur qui permet l'obtention du résultat désiré. C'est l'exemple caractéristique d'un système hybride.

3 Conversion numérique-analogique (conversion N-A)

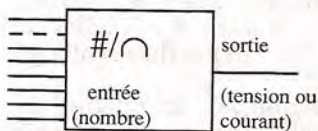


Fig. 1. convertisseur numérique-analogique ou CNA.

Lorsque le nombre d'entrée N est non signé, la tension (ou l'intensité du courant) de sortie n'a qu'une seule polarité : le convertisseur est unipolaire. Lorsque le nombre d'entrée N est signé, la tension (ou l'intensité du courant) de sortie est une grandeur algébrique : le convertisseur est bipolaire.

La conversion numérique-analogique réalise la transformation d'un signal numérique, généralement représenté en binaire naturel ou en décimal codé binaire (DCB), en une tension ou un courant. La figure 1 est la représentation symbolique du convertisseur qui réalise cette opération.

Poids de l'entrée

A chaque combinaison du nombre binaire d'entrée correspond une valeur différente de la tension de sortie.

A chaque chiffre d'un nombre binaire est affecté un poids différent suivant le rang du chiffre dans ce nombre.

La tension de sortie est une somme pondérée des différents poids (voir exemple dans la marge).

Pour un convertisseur de n bits, la résolution en pourcentage de la pleine échelle a pour expression :

$$\frac{1}{2^n - 1}$$

Exemple :

Nombre décimal de quatre chiffres : 2 5 8 3

La représentation générale de ce nombre est : $A_3 A_2 A_1 A_0$

Dans le code décimal :

$A_3 = 2, A_2 = 5, A_1 = 8, A_0 = 3$

Dans le code DCB :

$A_3 = 0010, A_2 = 0101,$
 $A_1 = 1000, A_0 = 0011$

L'équivalent en code DCB de 2583 s'écrit :

0010 0101 1000 0011

Avec le code DCB, ce nombre comporte 16 bits.

La représentation de chaque chiffre A est :

A_3	A_2
0 0 1 0	0 1 0 1
$E_3 E_2 E_1 E_0$	$E_3 E_2 E_1 E_0$
A_1	A_0
1 0 0 0	0 0 1 1
$E_3 E_2 E_1 E_0$	$E_3 E_2 E_1 E_0$

Comme dans le cas des convertisseurs numériques-analogiques, les convertisseurs analogiques-numériques peuvent être unipolaires ou bipolaires. La suite de l'étude des convertisseurs porte uniquement sur les convertisseurs unipolaires.

Résolution

La résolution d'un convertisseur N-A est égale au poids du bit de poids le plus faible. C'est la plus petite variation de la tension de sortie qui peut être provoquée par une modification du signal numérique d'entrée.

Exemple : Un convertisseur N-A de huit bits indique 00000000 pour une tension de 0 V et 00000001 pour une tension de 2 mV : la résolution de ce convertisseur est de 2 mV.

La résolution peut aussi être exprimée en pourcentage de la pleine échelle.

Reprenons l'exemple précédent. Un convertisseur N-A de huit bits, de résolution de 2 mV, a une tension de sortie pleine échelle qui correspond au nombre 11111111. Cette tension vaut donc $2 \text{ mV} \times 255 = 510 \text{ mV}$.

Lorsque le nombre binaire d'entrée évolue de 00000000 à 11111111 par pas de 1, la tension de sortie progresse de 0 à 510 mV par pas de 2 mV (il y a $[2^8 - 1]$ pas, c'est-à-dire 255 pas).

En pourcentage de la pleine échelle, la résolution du convertisseur est :

$$\frac{2}{510} \times 100 \% = 0,4 \%$$

(c'est aussi l'inverse du nombre de pas)

Code d'entrée DCB

Dans certains convertisseurs N-A on utilise le code DCB.

Pour quatre chiffres décimaux il faut un convertisseur 16 bits (4 bits par chiffre). La tension de sortie peut alors prendre 10^4 valeurs différentes (il y a 9999 pas).

Chaque chiffre A_0, A_1, A_2 ou A_3 , de quatre bits est représenté avec le code binaire naturel (voir exemple dans la marge). A_0 est le chiffre de poids le plus faible et A_3 celui de poids le plus fort. Dans chacun de ces chiffres, E_0 est l'élément binaire de poids le plus faible et E_3 celui de poids le plus fort.

La résolution d'un tel convertisseur est égale au poids de l'élément binaire E_0 du chiffre A_0 . En pourcentage de la pleine échelle, la résolution est représentée par l'inverse du nombre de pas. Pour notre exemple elle est égale à $1/9999$, c'est-à-dire $10^{-2} \%$.

4 Conversion analogique-numérique (conversion A-N)

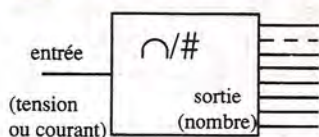


Fig. 2. Convertisseur analogique-numérique ou CAN.

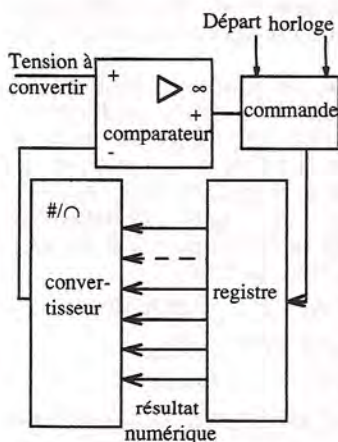


Fig. 3. Convertisseur analogique-numérique.

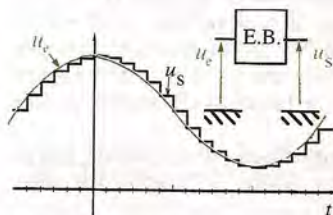


Fig. 4. Exemple de tension de sortie d'un échantillonneur-bloqueur lorsque la tension d'entrée comporte une composante alternative sinusoïdale superposée à une tension continue. La durée de conversion du CAN doit être inférieure à la durée de chacun des paliers de la tension u_s .

La conversion analogique-numérique réalise la transformation d'une tension ou d'un courant en un signal numérique (fig. 2), généralement représenté avec le code binaire naturel ou le code décimal codé binaire (DCB).

Principe de la conversion

Plusieurs catégories de convertisseurs analogique-numérique (CAN) correspondent au schéma de principe de la figure 3. Au début de la conversion, la sortie du comparateur est à l'état haut, elle autorise alors la conversion. La commande permet d'en lancer le processus; elle modifie, à la cadence imposée par l'horloge, le contenu binaire du registre jusqu'à ce que l'égalité entre la tension de sortie du CNA et la tension à convertir soit réalisée; la sortie du comparateur change alors d'état ce qui provoque l'arrêt de la conversion : à la sortie du registre on dispose alors du résultat de cette conversion (voir Etapes Electronique analogique).

Echantillonneur-bloqueur

Si la tension à convertir évolue pendant la conversion, le fonctionnement du convertisseur peut être perturbé. Pour éviter ce mauvais fonctionnement, la tension à convertir n'est pas directement appliquée à l'entrée du CAN mais passe par un échantillonneur-bloqueur qui mémorise la tension à convertir pendant toute la durée de la conversion (voir chapitre 29) : ainsi la tension appliquée à l'entrée du convertisseur est constante (fig. 4).

Multiplexage

Plusieurs sources peuvent fournir des tensions analogiques à un CAN. La conversion simultanée n'étant pas possible, on fait fonctionner le CAN en temps partagé par multiplexage des voies auxquelles les tensions sont appliquées. Une logique de commande sélectionne la voie d'entrée et donne l'ordre du début de la conversion. A la fin de cette opération, le convertisseur est relié à une autre voie pour opérer une nouvelle conversion et ainsi de suite jusqu'à ce que toutes les voies aient été « balayées ». Le cycle de fonctionnement est réglé par le signal d'horloge.

Un convertisseur numérique-analogique génère une tension électrique proportionnelle au nombre qui lui est fourni en entrée. En général ce nombre est représenté avec le code binaire naturel.

1 Échelle de résistances pondérées

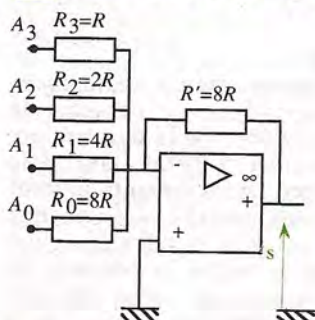


Fig. 1. Échelle de résistances pondérées.

A chaque bit d'un nombre binaire N de n bits, on fait correspondre une tension continue dont la valeur est 0 ou E selon que ce bit a lui-même pour valeur 0 ou 1. On effectue ensuite une somme pondérée des n tensions associées à N , en affectant à chaque tension le poids du bit correspondant au sein du nombre N .

Réalisons, par exemple, le montage sommateur inverseur de la figure 1 où N comporte quatre bits : $A_3 A_2 A_1 A_0$. Pour $N = 1001$, soit $(9)_{10}$, on peut, à l'aide d'une source de tension E unique et de commutateurs, associer à N les tensions $v_{A_3} = E$, $v_{A_2} = 0$, $v_{A_1} = 0$, $v_{A_0} = E$. Les résistances R_n (R_0, R_1, R_2, R_3) étant de la forme $R_n = 2^n R$, la tension de sortie u_s s'écrit sous la forme :

$$u_s = -(2^3 A_3 + 2^2 A_2 + 2^1 A_1 + 2^0 A_0)E$$

soit dans le cas particulier où $N = (9)_{10}$:

$$u_s = -(8 + 0 + 0 + 1)E = -9E$$

La tension u_s est donc ainsi proportionnelle au nombre N .

L'unique tension d'entrée E peut, par exemple, être égale à 1 V. La source qui fournit E doit avoir une résistance interne très faible devant R .

Pour un tel convertisseur à quatre bits, si $E = 1$ V, u_s évolue de 0 V lorsque $N = 0000$, à -15 V lorsque $N = 1111$. Suivant la valeur de N , il y a 2^n valeurs différentes possibles de la tension de sortie u_s (ici 16).

Plus le nombre n est important, plus l'écart entre les résistances doit être faible et meilleure doit être leur précision.

Cela limite l'utilisation d'une conversion par échelle de résistances pondérées à des nombres binaires formés de quatre ou cinq bits.

2 Réseau R-2R

Évaluons les tensions U_0 , U_1 et U_2 pour le montage de la figure 2.

En partant de la droite et en remontant progressivement le long du schéma vers la gauche, nous

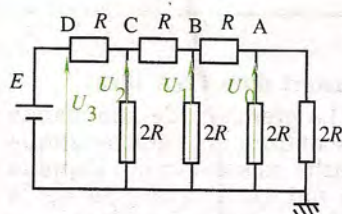
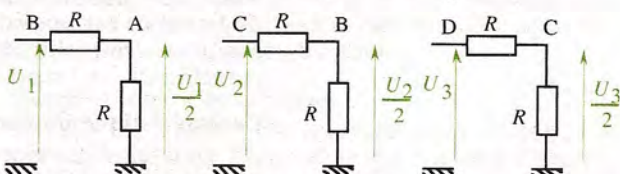


Fig. 2. Réseau R-2R.

Fig. 3. Schémas réduits.

obtenons les schémas réduits de la figure 3. De ces schémas, nous déduisons que :

$$U_0 = \frac{U_1}{2}, U_1 = \frac{U_2}{2} \text{ et } U_2 = \frac{U_3}{2}$$



Comme $U_3 = E$, nous obtenons :

$$U_0 = \frac{E}{8}, U_1 = \frac{E}{4}, U_2 = \frac{E}{2} \text{ et } U_3 = E$$

Deux valeurs de résistances seulement (R et $2R$) suffisent pour disposer d'une gamme de tensions pondérées.

En réalisant le montage de la figure 4, rien n'est changé quant aux potentiels de U_0, U_1, U_2 et U_3 . En effet, l'amplificateur est muni d'une contre-réaction : l'entrée non inverseuse étant à la masse, le potentiel de l'entrée inverseuse est égal à zéro. Le montage est un sommateur inverseur. La tension de sortie u_s est égale à : $-(U_0 + U_1 + U_2 + U_3)$,

$$\text{soit encore : } u_s = -\left(\frac{E}{8} + \frac{E}{4} + \frac{E}{2} + E\right)$$

En intégrant une logique de commande permettant la déconnexion de telle ou telle résistance égale à $2R$ (fig. 5), on fait disparaître le terme correspondant dans la somme, c'est-à-dire : $\frac{E}{8}$ ou $\frac{E}{4}$ ou $\frac{E}{2}$ ou E .

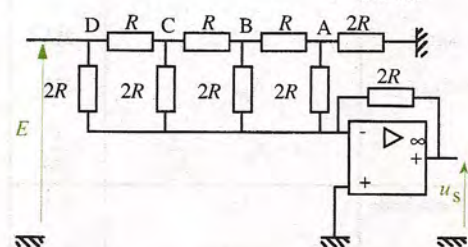


Fig. 4.

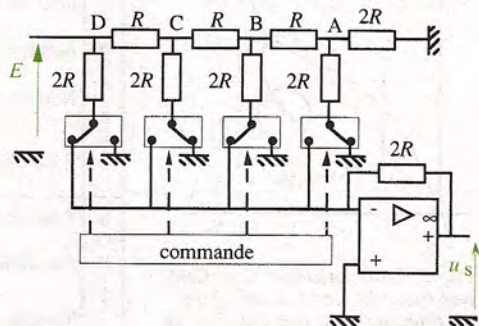


Fig. 5. C.N.A à réseau R-2R.

3 Caractéristiques d'un CNA

Caractéristique de transfert d'un CNA idéal

C'est la courbe $u_s(N)$. La grandeur de commande étant une grandeur discontinue, la caractéristique de transfert est une courbe en escalier qui s'appuie sur une droite (fig. 6) qui est l'enveloppe de la caractéristique : c'est la caractéristique idéale.

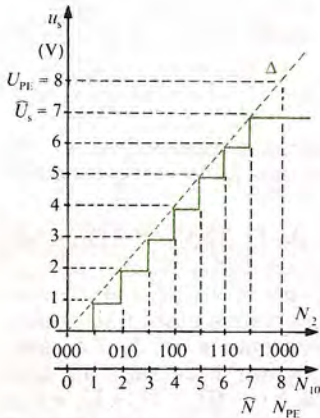


Fig. 6. Caractéristique de transfert d'un CNA à 3 e.b.

Caractéristique de transfert d'un CNA réel

En général, l'enveloppe de la caractéristique de transfert d'un CNA réel n'est pas une droite mais une courbe (fig. 7). Les écarts de la caractéristique réelle par rapport à la caractéristique idéale sont dus à la dispersion des composants utilisés.

Précision d'un CNA réel

La précision d'un CNA est l'écart maximal entre les enveloppes de la caractéristique réelle et de la caractéristique idéale, divisé par la valeur maximale de la tension de sortie appelée tension pleine échelle U_{pe} .

Temps de conversion

Le temps de conversion est la durée nécessaire pour que la tension de sortie soit acquise avec une précision déterminée lors du changement du nombre appliqué à l'entrée du CNA.

Caractéristiques essentielles

Les caractéristiques essentielles d'un CNA sont résumées dans le tableau ci-après :

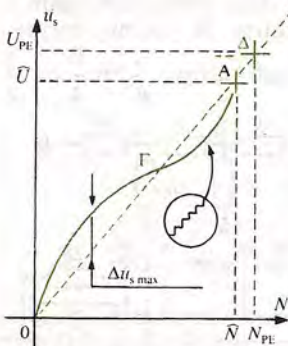


Fig. 7. Caractéristique d'un CNA réel (pour la clarté de la figure on a accentué la non-linéarité de la caractéristique réelle).

	Définition	Notation	Exemple
E N T R E E	Nombre d'éléments binaires	n	8
	Nombre de points	$N_{pe} = 2^n$	256
	Nombre maximal	$N_{pe} - 1$	255
	Résolution	$\mathcal{R} = (N_{pe})^{-1} = \frac{1}{N_{pe}}$	$\frac{1}{256}$
S O R T I E	Tension pleine échelle	U_{pe}	10 V
	Résolution	$S_0 = \mathcal{R}U_{pe}$	39 mV
	Tension maximale	$S_0 N$	9,96 V

4 Applications

Chaque fois qu'il faut transformer le signal de sortie d'un circuit numérique en une tension ou un courant, un convertisseur numérique-analogique est nécessaire. C'est le cas dans les asservissements numériques notamment.

Asservissement numérique

Réguler la vitesse de rotation d'un moteur, la température d'un four, créer des rampes d'accélération ou de décélération, piloter le fonctionnement de thyristors dans certains variateurs de vitesse, etc., nécessite des signaux numériques élaborés à partir de microprocesseurs se trouvant dans des ordinateurs ou des automates programmables. Des convertisseurs numériques-analogiques permettent la conversion des signaux numériques en grandeurs analogiques seules utilisables pour le fonctionnement des résistances, thyristors, moteurs, etc.

Bancs tests

Les constructeurs de machines électriques ont des bancs tests d'essais de machines entièrement automatisés. L'ensemble des essais est programmé sur ordinateur. Les signaux analogiques nécessaires à ces essais proviennent indirectement de l'ordinateur par l'intermédiaire, entre autres, de convertisseurs numériques-analogiques.

Il en est de même pour les essais automatiques de composants comme les diodes, les transistors, les amplificateurs opérationnels, etc.

Ces convertisseurs permettent la conversion d'une tension analogique en un nombre qui puisse être traité par un ordinateur.

1 Convertisseur simple rampe

La tension u_e à convertir est comparée à une rampe de tension $u_r(t)$ qui est obtenue en chargeant un condensateur de capacité C_0 par un courant d'intensité constante I_0 (fig. 1).

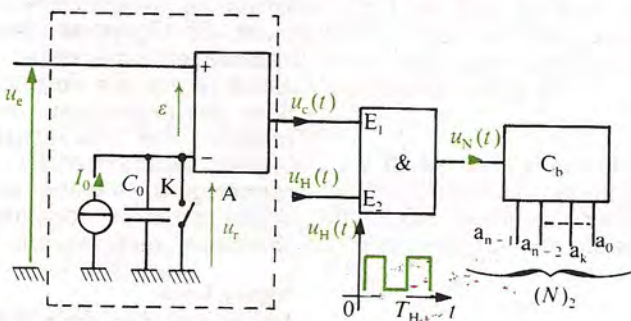


Fig. 1. Principe du convertisseur simple rampe.

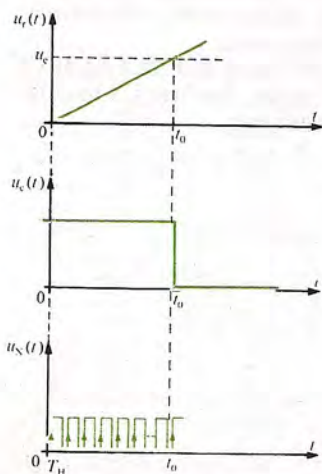


Fig. 2. Convertisseur simple rampe : évolution des tensions.

Tant que la tension u_e est supérieure à la référence $u_r(t)$, la sortie du comparateur A impose un niveau logique 1 à l'entrée E_1 de l'opérateur ET. Le compteur binaire C_b enregistre les impulsions d'horloge $u_H(t)$ de période T_H .

Dès que la tension rampe $u_r(t)$ atteint la valeur u_e , le comparateur A impose à l'entrée E_1 de la porte ET un niveau logique zéro, le comptage s'arrête. Le contenu du compteur est alors N :

$$N = k u_e \quad \text{avec :} \quad k = \frac{C_0}{I_0 T_H} \quad \text{si} \quad \varepsilon = 0$$

L'évolution des tensions est présentée sur la figure 2.

Limitations

La précision de conversion dépend de la valeur de la capacité C_0 et de la stabilité du signal d'horloge $u_H(t)$.

Il est possible d'améliorer la conversion en réalisant un circuit à double rampe.

2 Convertisseur double rampe

Le principe de ce convertisseur est représenté sur la figure 3.

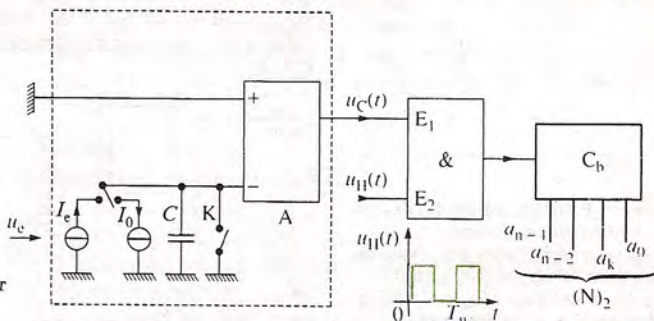


Fig. 3. Principe du convertisseur double rampe.

- Pendant une durée θ_1 fixée, invariable, le condensateur est chargé à l'aide d'un courant constant dont l'intensité I_e est proportionnelle à la tension u_e ($I_e = k u_e$). Au bout de cette durée, la tension aux bornes du condensateur est u_{Ce} .

- Le condensateur est ensuite déchargé à l'aide d'un courant constant, d'intensité I_0 indépendante de u_e , jusqu'à atteindre la tension $u_c = 0$, au bout d'une durée θ_2 (fig. 4).

- La quantité d'électricité emmagasinée pendant la charge est égale à celle fournie pendant la décharge.

Soit : $I_e \theta_1 = I_0 \theta_2 = C u_{Ce}$ avec $I_e = k u_e$

Si les nombres d'impulsions d'horloge, de période T , pendant les durées θ_1 et θ_2 sont respectivement N_1 et N_2 , nous pouvons écrire :

$$N_1 T k u_e = N_2 T I_0$$

Soit :

$$N_2 = \frac{k N_1 u_e}{I_0}$$

N_1 , I_0 , et k étant des constantes, le nombre d'impulsions N_2 est proportionnel à la tension à convertir.

Ces convertisseurs, bien que relativement lents, sont très utilisés dans les systèmes d'affichage comme ceux qui équipent les appareils de mesure numériques, notamment de nombreux multimètres.

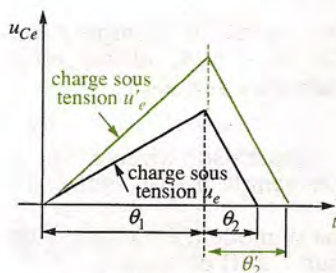


Fig. 4. Allure des rampes pour deux valeurs de la tension à mesurer : u_e et u'_e .

3 Conversion par comptage

Convertisseur à comparaison directe

La méthode consiste à comparer le signal analogique d'entrée u_e (fig. 5) à l'image analogique, u_{CNA} , du contenu d'un compteur binaire C_b , donnée par un convertisseur numérique-analogique CNA.

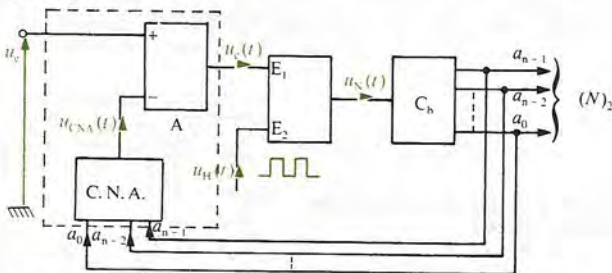


Fig. 5. Principe du convertisseur à comparaison directe.

On peut remarquer que, pour ce système, la sortie $(N)_2$ (nombre codé en binaire) réagit sur l'une des entrées du comparateur analogique A. Ce convertisseur est un système bouclé.

Tant que la tension u_e est supérieure à la tension u_{CNA} , la sortie du comparateur A impose un niveau logique 1 à l'entrée E_1 de l'opérateur ET. Le compteur binaire C_b est incrémenté, ce qui entraîne la modification de la valeur de u_{CNA} .

Dès que cette dernière atteint la valeur u_e , le comparateur A impose à l'entrée E_1 de l'opérateur ET un niveau logique 0 et le comptage s'arrête.

Le contenu du compteur est alors N :

$$N = ku_e \quad \text{avec : } k = \frac{1}{q_{CNA}} \quad \text{si } \varepsilon = 0$$

L'évolution des tensions est indiquée figure 6.

Limitations

Comme dans le cas des CAN à rampe, le temps de conversion est long.

La précision du convertisseur à comparaison directe dépend de celle du C.N.A. utilisé pour produire la tension de comparaison u_{CNA} .

Remarques

- La structure du CNA à comparaison directe (fig. 5) et celle du CNA à simple rampe (fig. 1) sont comparables.

La différence réside dans le mode d'élaboration de la tension de comparaison : $u_r(t)$ ou u_{CNA} .

- Le convertisseur à comparaison directe montre qu'il est possible de réaliser les fonctions CNA et CAN à partir de la seule fonction CNA.

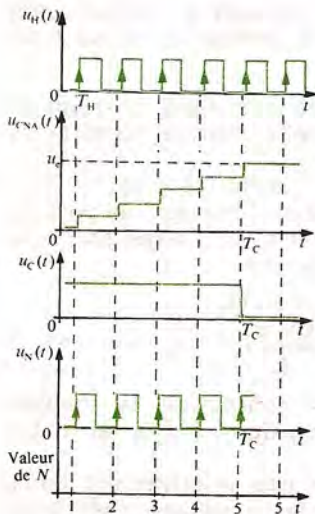


Fig. 6. Convertisseur à comparaison directe : évolution des tensions.

- Le convertisseur à comparaison directe présente une structure de circuit bouclé.

Autre conversion

La conversion par approximations successives met en œuvre un principe analogue à celui utilisé pour la mesure d'une masse à l'aide d'une balance. Le temps de conversion est beaucoup plus court que celui des CAN à intégration ou à comparaison directe.

4 Caractéristiques essentielles d'un CAN

CAN idéal

Les caractéristiques d'un CAN idéal sont résumées dans le tableau ci-après :

	Définition	Notation	Exemple
E N T R É E	Tension pleine échelle	U_{pe}	10 V
	Résolution	$S_0 = \mathcal{R}U_{pe}$	39 mV
	Tension maximale	$S_0 N$	9,96 V
S O R T I E	Nombre d'éléments binaires	n	8
	Nombre de points	$N_{pe} = 2^n$	256
	Nombre maximal	$N_{pe} - 1$	255
	Résolution	$\mathcal{R} = \frac{1}{N_{pe}}$	$\frac{1}{256}$

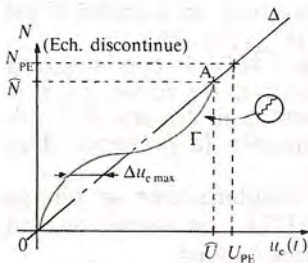


Fig. 7. Caractéristique d'un CAN réel (pour la clarté de la figure, la non-linéarité de la caractéristique a été accentuée).

CAN réel

L'enveloppe de la caractéristique de transfert d'un CAN idéal est une droite alors que celle d'un CAN réel est une courbe (fig. 7).

La précision d'un CAN est l'écart maximal $\Delta u_{e\max}$ entre la caractéristique réelle et la caractéristique idéale divisé par la valeur maximale de la tension d'entrée :

$$p = \frac{\Delta u_{e\max}}{U_{pe}}$$

La durée de conversion est le temps nécessaire pour que la grandeur numérique de sortie soit acquise lors du passage de la commande analogique de zéro à la valeur correspondant au maximum de l'échelle.

1 Acquisition des signaux

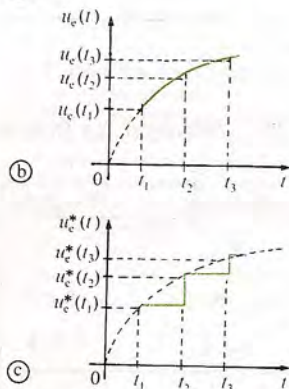
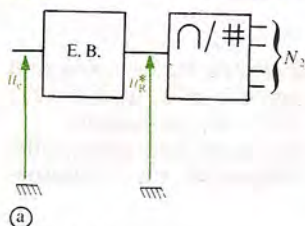


Fig. 1. Évolution du signal analogique à l'entrée d'un C.A.N. placé à la suite d'un échantillonneur-bloqueur.

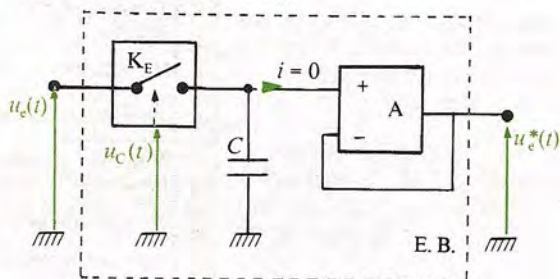
Fig. 2. Principe de l'échantillonneur-bloqueur.

La conversion analogique-numérique n'est pas un processus instantané. En conséquence, la tension analogique d'entrée u_e doit être maintenue constante entre les instants tels que t_1 et t_2 (fig. 1 a et b) afin de rendre possible la conversion. Nous avons vu que l'acquisition du signal, en vue de la conversion A-N, nécessite un dispositif supplémentaire appelé échantillonneur-bloqueur (fig. 2).

Principe de l'échantillonneur-bloqueur

- Phase d'échantillonnage

L'échantillonnage consiste à prélever périodiquement la valeur de la tension analogique u_e (fig. 2). Cette opération non linéaire est réalisée en utilisant un interrupteur électronique K_E commandé au rythme d'un signal $u_H(t)$ dont la période T_E est la période d'échantillonnage.



- Phase de maintien

A l'instant kT_E , le condensateur de capacité C est chargé sous la tension $u_e(kT_E)$ (fig. 3).

Entre les instants kT_E et $(k+1)T_E$, le condensateur maintient constante la tension de sortie u_{eb} à la valeur $u_e(kT_E)$ à condition que l'intensité i soit nulle. Cela rend indispensable la présence d'un l'amplificateur suiveur A .

A l'instant $(k+1)T_E$, le condensateur se charge sous la tension $u_e[(k+1)T_E]$. Le signal u_{eb} est appelée signal échantillonné-bloqué.

- Conversion

A l'entrée du convertisseur analogique-numérique (fig. 1), la tension analogique échantillonnée-bloquée u_{eb} est maintenue constante pendant la période d'échantillonnage T_E .

La conversion est possible dans la mesure où le temps de conversion T_c satisfait à la relation :

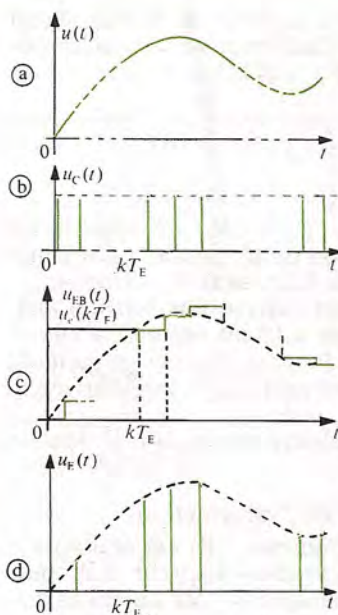


Fig. 3. Évolution des tensions.

- a) Tension d'entrée $u_e(t)$.
 b) Tension de commande de l'interrupteur.
 c) Tension de sortie de l'échantillonneur-bloqueur.
 d) Signal échantillonné :

$$\begin{cases} u_E(t) = 0 & \text{si } t \neq kT_E \\ u_E(t) = u_e(t) & \text{si } t = kT_E. \end{cases}$$

$$T_c < T_E$$

- Remarque

On appelle **signal échantillonné** $u_E(t)$ le signal théorique dessiné sur la figure 3 d.

Fréquence d'échantillonnage

- Position du problème

Lors de l'acquisition d'un signal, l'échantillonnage est une opération essentielle.

Un signal analogique u_e peut être déterminé expérimentalement point par point.

Des mesures successives à des instants $t = kT_E$ constituent l'échantillonnage de la fonction u_e : c'est le signal u_E (fig. 3 d). Un problème se pose ensuite : malgré l'absence d'information entre les divers échantillons, peut-on retrouver complètement le signal u_e à partir de u_E ? Cette reconstitution du signal u_e à partir de u_E s'appelle l'interpolation (fig. 4). Elle nécessite de choisir convenablement la période d'échantillonnage T_E .

- Théorème de l'échantillonnage (théorème de Shannon)

La reconstitution parfaite d'un signal analogique u à partir d'échantillons prélevés à la fréquence

d'échantillonnage $F_E = \frac{1}{T_E}$ n'est possible que si la

fréquence F_E est au moins deux fois plus grande que F_m , plus grande des fréquences du spectre du signal $u(t)$:

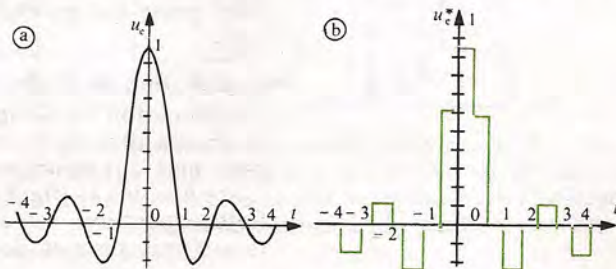
$$F_E \geq 2 F_m$$

Le théorème de Shannon montre que pour reconstituer parfaitement un signal, il n'est pas nécessaire de le transmettre intégralement ! Ainsi pour une communication téléphonique, puisque $F_m = 4 \text{ kHz}$, il suffit d'échantillonner le message à $F_E = 8 \text{ kHz}$. De même, dans le cas de disque numé-

Fig. 4. Influence de la fréquence d'échantillonnage sur la reconstitution d'un signal.

a) Signal d'entrée $u_e(t) = \frac{\sin \pi t}{\pi t}$.

b) Signal de sortie échantillonné bloqué avec $T_E = 0,5 \text{ s}$.



rique, pour la restitution haute fidélité d'un signal sonore ($F_m = 20 \text{ kHz}$); l'information est échantillonnée à la fréquence $F_E \approx 44 \text{ kHz}$.

2 Structure d'une chaîne numérique

Prise d'information

Le capteur C (fig. 5) a pour rôle de fournir un signal électrique fonction de la grandeur physique θ , ici la température du four, sans la perturber. Généralement le capteur délivre une tension analogique $u_c(t)$. La réponse $u_c(t)$ du capteur, à l'excitation θ , doit être indépendante des grandeurs d'influence qui peuvent également caractériser le système étudié.

Pour mesurer n grandeurs distinctes il faut n capteurs.

Traitement numérique de l'information

Le signal analogique d'entrée $u_c(t)$ est appliqué à l'entrée d'un échantillonneur-bloqueur E.B. puis converti en signal numérique N_E par un convertisseur analogique-numérique CAN codé au moyen de n éléments binaires. L'information est traitée par une unité de calcul.

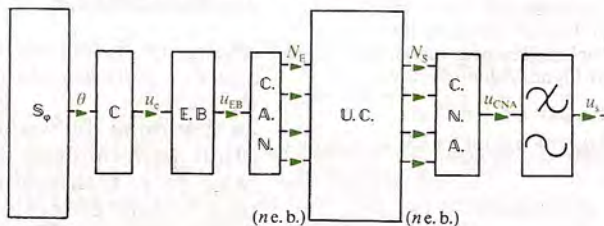


Fig. 5. Structure d'une chaîne numérique (chaîne directe). Mesure de la température θ d'un four.

A la sortie du calculateur, un convertisseur numérique-analogique CNA transforme le signal numérique N_s en signal analogique quantifié $u_{CNA}(t)$. Un filtre passe-bas permet de restituer un signal utile $u_s(t)$.

Exploitation de l'information

L'information numérique N_s obtenue en sortie de la chaîne directe de traitement peut être utilisée pour agir automatiquement sur la grandeur physique étudiée G (fig. 6).

Cette opération est réalisée par une chaîne de retour qui permet de comparer l'information de

sortie N_s à un nombre N_0 , image numérique de la grandeur de consigne θ_0 .

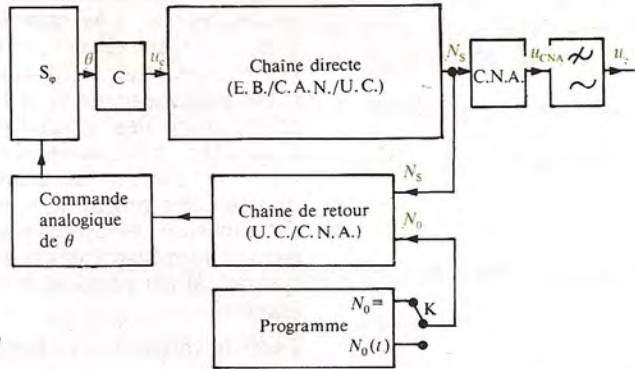


Fig. 6. Structure d'une chaîne numérique en boucle fermée. Asservissement de la température θ du four S_θ à la valeur de consigne $\theta_0(t) = kN_0(t)$.

Il y a asservissement de la grandeur physique θ lorsque la grandeur de consigne N_0 évolue en fonction du temps. Dans le cas particulier où la grandeur de consigne est indépendante du temps : $\theta_0 = C^{te}$ ou $N_0 = C^{te}$ on parle de régulation de la grandeur physique θ .

Chaîne d'acquisition

Une chaîne d'acquisition de données (fig. 7) permet de multiplier le nombre de grandeurs physiques mesurées. Ce système est construit autour d'une unique chaîne de mesure.

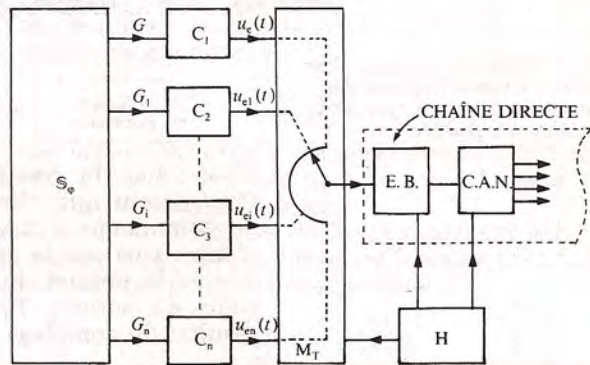


Fig. 7. Chaîne d'acquisition.

En conséquence, chaque voie de mesure doit être périodiquement dirigée vers l'entrée de la chaîne de traitement. Commandé au rythme de l'horloge H , le multiplexeur temporel M_T réalise cette opération d'aiguillage.

1 Présentation

Nous nous limitons à un exemple correspondant à un voltmètre très courant. L'organisation simplifiée de l'appareil est représentée sur la figure 1. Pour cette étude, afin de faciliter les explications, la tension mesurée v_x et la tension de référence $V_{\text{réf}}$ sont supposées positives. Cela nous a conduit à ajouter, à l'entrée du montage, un opérateur « $\times (-1)$ » dont la fonction est de faire correspondre une tension négative $-v_x$ à la tension mesurée v_x . La fonction électronique essentielle est réalisée par un montage intégrateur à amplificateur opérationnel. Il est possible d'appliquer à l'entrée de cet étage :

- soit la tension $-v_x$ lorsque K_1 seul est fermé ;
- soit la tension de référence $V_{\text{réf}}$ lorsque K_2 seul est fermé.

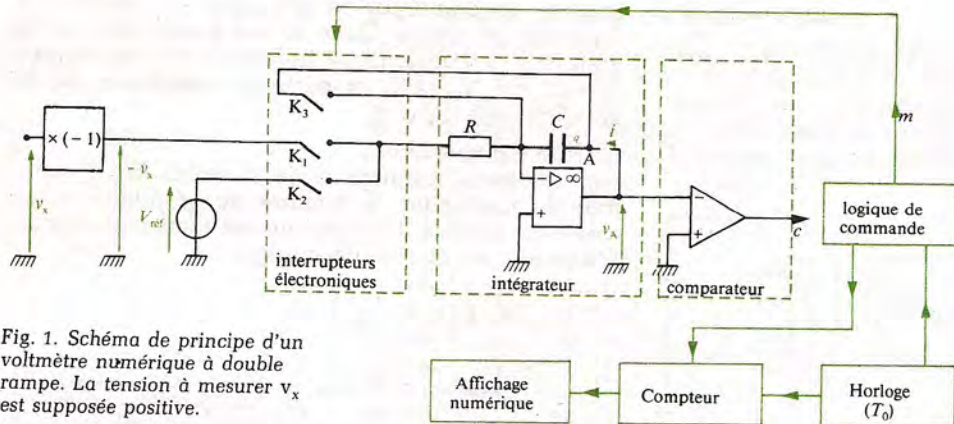


Fig. 1. Schéma de principe d'un voltmètre numérique à double rampe. La tension à mesurer v_x est supposée positive.

C'est l'état du comparateur placé à la suite de l'intégrateur qui détermine l'action de la logique de commande d'une part sur les interrupteurs, d'autre part sur le compteur. Lorsque cette commande le permet, le compteur reçoit des impulsions de période T_0 délivrées par l'horloge. Le résultat du comptage est transféré à un afficheur.

2 Fonctionnement

Nous allons étudier comment évolue la quantité d'électricité q_A , qui est portée par l'armature du condensateur reliée au point A, au cours d'un cycle de fonctionnement.

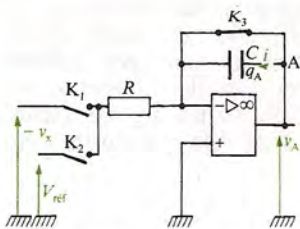


Fig. 2. Phase d'initialisation de l'intégrateur.

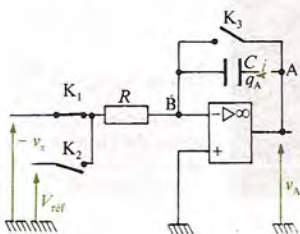


Fig. 3. Phase de charge du condensateur.

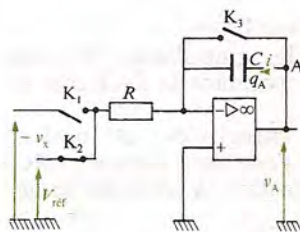


Fig. 4. Phase de décharge du condensateur.

A l'instant $t = 0$: le condensateur C est déchargé (fig. 2) : $q_A(0) = 0$

A partir de l'instant $t = 0^+$: la logique de commande place les interrupteurs dans l'état suivant (fig. 3) : K_1 fermé, K_2 et K_3 ouverts. Le condensateur se charge.

En appliquant la loi des nœuds au point B, nous obtenons l'intensité i du courant de charge (l'amplificateur opérationnel fonctionne en régime

linéaire) :

$$i = \frac{V_x}{R} = C^{\text{te}}$$

Puisqu'il s'agit d'une charge à courant constant, à un instant t :

$$q_A = it = \frac{V_x}{R} t$$

La durée $T_1 = t_1$ du régime de charge est égale à un nombre entier de périodes du signal d'horloge ; elle est fixée par le constructeur et ne dépend pas de la tension v_x :

$$T_1 = n_0 T_0$$

A la fin du régime de charge : le condensateur porte une charge $q_A(t_1)$:

$$q_A(t_1) = \frac{V_x}{R} T_1 = \frac{V_x}{R} n_0 T_0$$

A l'instant t_1 : la logique de commande impose un nouvel état pour les interrupteurs (fig. 4) : K_1 et K_3 ouverts, K_2 fermé.

Nous trouvons l'intensité i du courant en opérant comme précédemment :

$$i = - \frac{V_{\text{réf}}}{R}$$

Nous constatons qu'elle est constante et négative. Donc, à partir de l'instant t_1 le condensateur se décharge à courant constant.

A l'instant t_2 ce régime de fonctionnement cesse : la charge q_A est nulle. Entre les instants t_1 et t_2 le condensateur perd une charge égale à :

$$\left[\frac{V_{\text{réf}}}{R} (t_2 - t_1) \right]$$

C'est évidemment la charge $q_A(t_1)$ qui était posédée à la fin du régime de charge.

D'où :

$$q_A(t_1) = \frac{V_{\text{réf}}}{R} (t_2 - t_1)$$

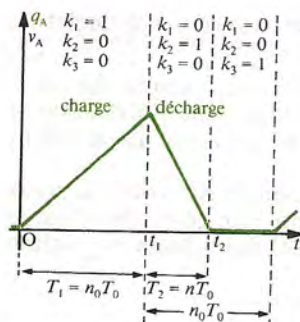
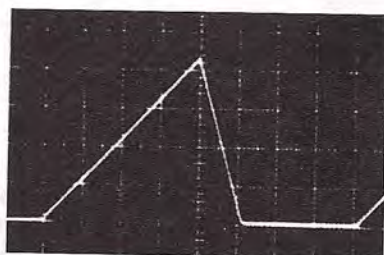


Fig. 5. Courbe $q_A(t)$ (double rampe).

— La durée de charge, T_1 , est imposée par la logique de commande et par l'horloge. Elle ne dépend pas de la mesure à réaliser.

— La durée de la décharge $T_2 = nT_0$ est proportionnelle à v_x .

Les variations de la quantité d'électricité q_A au cours d'un cycle de fonctionnement sont représentées sur la figure 5. Notons que cette courbe représente également, à une échelle différente, les variations de la tension de sortie v_A de l'intégrateur puisque q_A et v_A sont des grandeurs proportionnelles ($q_A = Cv_A$).



Courbe $v_A(t) = \frac{q_A(t)}{C}$;
c'est la tension de sortie de l'intégrateur qui est visualisée à l'oscilloscope.

3 Indication du compteur

Nombre d'impulsions enregistrées

Le compteur enregistre les impulsions fournies par l'horloge uniquement pendant la décharge du condensateur.

Le nombre d'impulsions enregistrées est égal au nombre n de périodes qui s'écoulent durant l'intervalle de temps $(t_2 - t_1)$ nécessaire à cette décharge :

$$n = \frac{t_2 - t_1}{T_0}$$

Relation entre n et v_x

Dans la relation donnant $q_A(t_1)$, remplaçons $(t_2 - t_1)$ par son expression en fonction de T_0 (nous supposons n entier) :

$$q_A(t_1) = \frac{V_{\text{réf}}}{R} nT_0 = \frac{v_x}{R} n_0 T_0$$

D'où :

$$n = \frac{n_0}{V_{\text{réf}}} v_x$$

Le nombre n de périodes d'horloge enregistré par le compteur est donc proportionnel à la tension à mesurer v_x .

Affichage

L'indication n du compteur étant proportionnelle à la tension v_x , il suffit de choisir un rapport $\frac{n_0}{V_{\text{réf}}}$ égal à une puissance entière de 10 pour que l'indication n du compteur soit un multiple décimal de la tension à mesurer v_x (à une unité près).

Exemple : Les caractéristiques du voltmètre présenté sont les suivantes :

— $T_0 = 20,0 \mu\text{s}$, $n_0 = 10^3$ donc $n_0 T_0 = T_1 = 20,0 \text{ ms}$.

— $V_{\text{réf}} = 10,0 \text{ V}$, $R = 20,0 \text{ k}\Omega$,

$C = 1,00 \mu\text{F}$, donc $RC = 20,0 \text{ ns}$

● *Premier exemple de mesure*

$$v_x = 3,520 \text{ V}$$

La décharge a lieu en un nombre de périodes égal à :

$$n = \frac{n_0}{V_{\text{réf}}} v_x = 352$$

Dans ce cas, la durée T_2 correspond à un nombre entier de périodes d'horloge. L'indication du compteur est égale à $n = 352$.

● *Deuxième exemple de mesure* $v_x = 3,525 \text{ V}$

Dans ce cas : $n = 352,5$

La durée T_2 est telle que : $352T_0 < T_2 < 353T_0$

Le contenu du compteur n'évoluant pas entre les instants $352T_0$ et $353T_0$, la quantité affichée sera $n' = 352 < n$. Il y a donc là une première cause d'erreur : le nombre affiché est approché à une unité près par défaut.

Remarque :

Pour tous les voltmètres à rampe, la durée de charge T_1 est égale à 100 ms soit 5 périodes du secteur en Europe ($f = 50 \text{ Hz}$) et 6 périodes en Amérique ($f = 60 \text{ Hz}$). Le fait que T_1 soit égal à un nombre entier de périodes du secteur permet de limiter les erreurs dues au « bruit ».

Les mesures des tensions alternatives, des intensités des courants continus ou alternatifs et des résistances peuvent se ramener simplement à une mesure de tension continue. Il est alors possible de munir un voltmètre de plusieurs autres fonctions de mesure : l'appareil obtenu est un multimètre numérique.

1 Voltmètre

Un voltmètre numérique à courant continu est une sorte de convertisseur analogique-numérique (CAN) puisque sa grandeur d'entrée est une tension continue réglable et sa sortie un nombre. Cependant, nous verrons en fin de chapitre que la fonction qu'il réalise est en fait beaucoup plus complexe. En particulier si la tension qu'on lui applique comporte des fluctuations périodiques, il en donne approximativement la valeur moyenne (cette valeur moyenne est donnée exactement pour certaines valeurs particulières de la fréquence des fluctuations). Il diffère également des convertisseurs analogiques-numériques par sa cadence de conversion : 4 ou 5 mesures par seconde alors qu'un circuit intégré CAN en réalise facilement 10^5 par seconde.

Le principe de la double rampe exposé au chapitre 92 est mis en œuvre dans la plupart des multimètres industriels. Un opérateur permet la mesure des tensions continues positives ou négatives : il est conçu pour fournir à l'entrée de l'intégrateur une tension de même valeur absolue que la tension à mesurer et de signe convenable.

Caractéristiques

- Calibres

Un voltmètre industriel est pourvu de plusieurs calibres. Un diviseur de tension placé à l'entrée de l'appareil permet d'obtenir une tension V_x à l'entrée de l'intégrateur au plus égale à la tension de référence $V_{\text{réf}}$.

- Nombre de points

Le nombre de points d'un appareil numérique est le nombre de valeurs positives discrètes qu'il est susceptible d'afficher. Par exemple, un voltmètre à 4000 points, utilisé sur le calibre 4 V, peut afficher 4000 tensions positives régulièrement réparties dans l'intervalle compris entre 0,000 V et 3,999 V.

Lorsque la tension à mesurer est très voisine de zéro volt, certains appareils affichent alternativement les signes + ou -. C'est le cas, par exemple du voltmètre MX40 utilisé sur le calibre 400 mV. Il « considère » comme nulle toute tension dont la valeur est comprise entre - 10 μ V et 10 μ V.

- **Étendue de la plage de mesure**

Pour des tensions continues, comme le signe est pris en compte et indiqué, l'étendue de la plage de mesure est le double de l'intervalle correspondant au nombre de points.

- **Résolution**

Elle correspond à une unité du chiffre de poids le plus faible (dernier chiffre de droite). Avec un voltmètre à 4000 points sur le calibre 4 V, la résolution est égale à 1 mV.

- **Nombre de digits**

Le voltmètre considéré dans l'exemple précédent est équipé d'un afficheur à quatre chiffres dont le premier chiffre à gauche ne peut être égal qu'à 0, 1, 2 ou 3. Conventionnellement, il est dit à «3 digits 1/2» pour exprimer le fait que le nombre maximal affiché est compris entre 10^3 et 10^4 lorsqu'il n'est pas tenu compte de la présence d'une virgule (ou point décimal).

- **Résistance d'entrée**

La résistance d'entrée d'un voltmètre numérique peut dépasser plusieurs centaines de mégohms et sa valeur peut être la même pour tous les calibres (par exemple : 10 M Ω).

Pratiquement, un voltmètre numérique se comporte comme un appareil qui ne dérive aucun courant dans les conditions de mesure les plus courantes. De ce point de vue, il possède un avantage considérable par rapport aux appareils analogiques.

Voltmètre alternatif RMS

La valeur efficace V d'une tension périodique $v(t)$, de période T , est la valeur d'une tension continue qui provoquerait dans une résistance le même effet thermique (même dégagement de chaleur) dans le même temps. On peut établir, que son expression mathématique est :

$$v = \sqrt{v^2(t)} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt}$$

Valeur efficace d'une tension périodique

Selon que l'appareil prend en compte la totalité de la tension $v(t)$ (fig. 1 a), ou seulement son ondulation (fig. 1 b), l'appareil est respectivement dit RMS - AC + DC ou RMS - AC.

- Il est utilisé pour mesurer les valeurs efficaces des tensions périodiques.

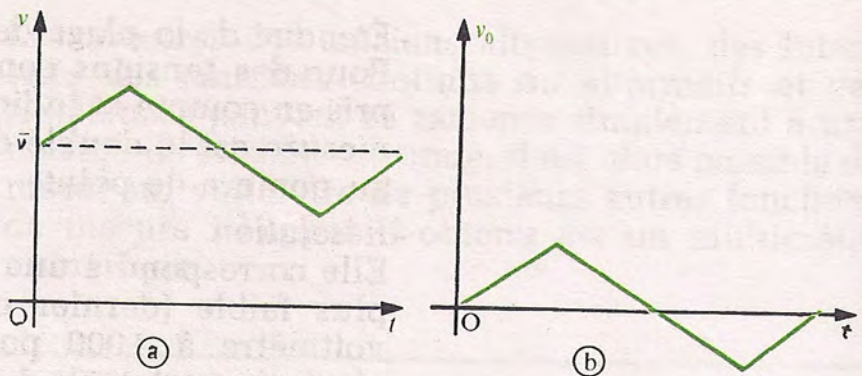


Fig. 1. Fonctions
RMS - AC + DC ou RMS - AC
 v est la somme d'une
composante continue $\bar{v}$ et d'une
ondulation v_0 :

$$v(t) = \bar{v} + v_0$$

— Fonction RMS - AC + DC :
l'appareil indique la valeur

efficace V de $v(t)$.
— Fonction RMS - AC :
l'appareil indique seulement la
valeur efficace v_0 de $v_0(t)$.

Relation. On peut montrer
que :

$$V = \sqrt{\bar{v}^2 + V_0^2}$$

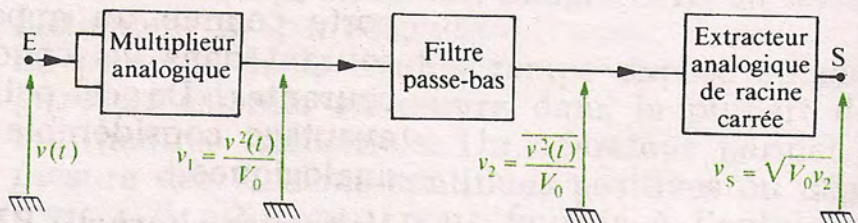
Fig. 2. Principe de la mesure des
valeurs efficaces (RMS de Root
Mean Square : racine de la
moyenne du carré).

- Le dispositif fournit une
tension continue v_s égale à la
valeur efficace V de la tension
 $v(t)$

$$v_s = \sqrt{V_0 v_2} = \sqrt{v^2(t)} = V$$

- Cette tension v_s est ensuite
mesurée au moyen du voltmètre
numérique et affichée.
- Facteur de crête : c'est le
quotient $\frac{\hat{v}(t)}{V}$. Le plus souvent les

caractéristiques du voltmètre
précisent un facteur de crête
maximal à ne pas dépasser (4
par exemple). Au-delà, la mesure
est mauvaise car la sortie du
multiplieur analogique est
transitoirement saturée.



2 Ampèremètre continu et alternatif

La figure 3 montre comment la mesure d'une
intensité est ramenée à celle d'une tension.
Suivant l'utilisation choisie le multimètre peut
indiquer les valeurs moyennes ou les valeurs
efficaces des intensités.

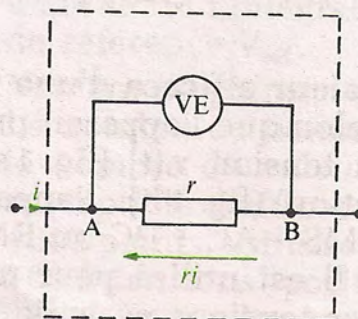


Fig. 3. Ampèremètre numérique.
Le voltmètre numérique mesure
la tension ri . En position
ampèremètre l'appareil affiche
directement la valeur de
l'intensité en tenant compte de la
résistance r du shunt (sa valeur
dépend du calibre).

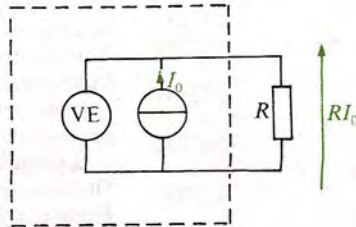
3 Ohmmètre

Lorsque cette fonction est sélectionnée, un générateur de courant débite un courant d'intensité I_0 dans la résistance inconnue (fig. 4). L'appareil mesure la tension RI_0 et affiche directement la valeur de la résistance.

Fig. 4. Ohmmètre numérique. L'intensité I_0 dépend du calibre de l'ohmmètre.

Exemples (MX40) :

Calibre	400 Ω	4 k Ω	4 M Ω
I_0	1 mA	$10^2 \mu\text{A}$	10^2nA



Mesure des faibles tensions

Avec le voltmètre de principe présenté au paragraphe 1, aucune rampe de charge ne peut être obtenue quand la tension d'entrée v_x devient nulle. Pour remédier à cet inconvénient et également pour faciliter la mesure des faibles tensions, un générateur de courant constant (intensité I_0) est parfois ajouté au montage (fig. a).

Dans ces conditions la charge acquise au bout de la durée T_1 est, avec les notations de la figure b :

$$q_A(t_1) = (i_x + I_0) T_1$$

$$\text{avec } i_x = \frac{v_x}{R}.$$

De ce fait, le comptage ne doit débuter qu'à l'instant t'_1 . La durée $\Delta t = t'_1 - t_1$ pendant laquelle le compteur reste bloqué, ne dépend que des valeurs de I_0 , $V_{\text{réf}}$ et R . Elle est fixée une fois pour toutes. La durée $T_2 = t_2 - t'_1$ et le résultat n du comptage ne sont pas modifiés par I_0 .

Conclusion : Une rampe est obtenue même si $v_x = 0$.

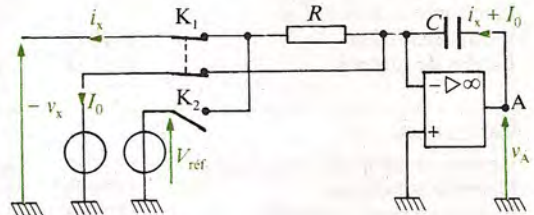


Fig. a

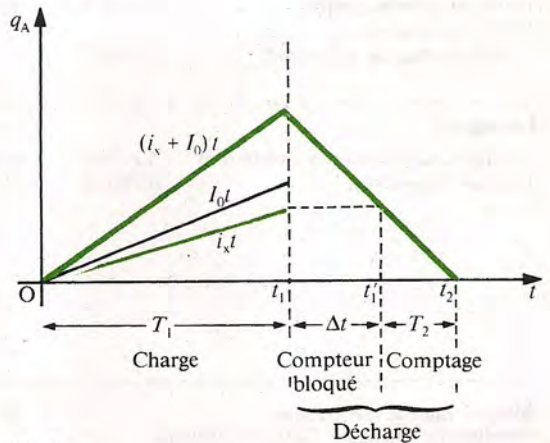


Fig. b

Mémentos

Action commerciale	177146	Finances publiques	177825
Automatismes industriels	176573	Hydraulique industrielle	177147
Comptabilité		La distribution	177733
et gestion appliquées	177155	La pratique des claviers	177988
Construction mécanique	177739	La publicité	176857
Dessin industriel	176576	Le plan comptable abrégé	177149
Droit	177139	Législation du travail	
Droit constitutionnel	177824	1995/1996	177123
Économie d'entreprise	177144	Marketing direct	177865
Économie générale	177981	Mathématiques appliquées	
Électricité - Courant continu	177787	à l'électricité - Tome 1	177150
Électricité appliquée		Mathématiques appliquées	
Exercices	177142	à l'électricité - Tome 2	176574
Électricité professionnelle		Mathématiques financières	
Tome 1	177720	appliquées	177148
Électricité professionnelle		Outils de gestion commerciale	176816
Tome 2	177785	Pneumatique industrielle	177748
Électricité professionnelle		Régulation - Tome 1	177145
Tome 3	177830	Régulation - Tome 2	177743
Électronique analogique	177788	Régulation - Tome 3	176575
Électronique de puissance	176966	Techniques du commerce	
Électronique numérique	177990	international	177140
Électrotechnique	177992	Unités et grandeurs	176891
Études de marché	176119		

Références

Communication des entreprises	177157	Gestion de la PME/PMI	177912
Denrées et boissons	177853	Maintenance	177840
Électricité professionnelle	177914	Marketing	177158
Électrochimie	177866	Mathématiques appliquées	177738
Français	177154	Organisation et gestion des entreprises	177837
Génie électronique	176577	Précis de mathématiques	
Génie électrotechnique	176578	et de statistiques	177979
Génie mécanique -			
Automatismes industriels	177737		

Lexiques

Lexique Automatismes industriels	177791	Lexique Électronique	177794
Lexique Éco-Droit	177926		

COLLECTION ÉTAPES



Cet ouvrage s'adresse aux utilisateurs de matériels électriques, techniciens ou non, qui ont besoin de disposer rapidement d'informations théoriques ou pratiques. Il est également destiné à tous ceux qui souhaitent se familiariser

avec l'électricité. Parce qu'il regroupe les résultats essentiels de l'électricité élémentaire, cet ouvrage peut aussi constituer un aide-mémoire très utile aux étudiants et aux auditeurs de la formation permanente.

Les auteurs ont choisi de privilégier l'observation et l'exploitation d'expériences plutôt que l'exposé de théories et le recours aux mathématiques. Ces connaissances peuvent suffire pour acquérir une bonne maîtrise de nombreux équipements électriques, la complexité des systèmes électriques excluant de plus en plus la possibilité d'acquérir les capacités nécessaires pour le dépannage et même parfois la maintenance.

BEP - BAC PRO

