

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

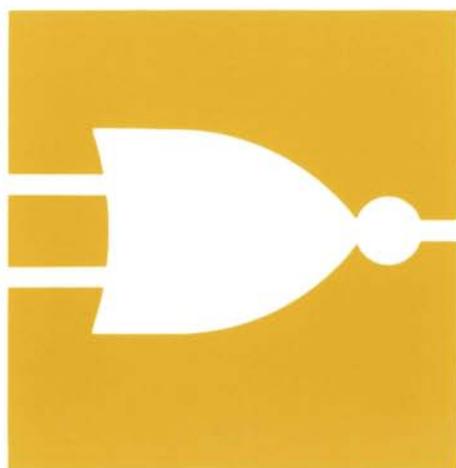
# Traité d'Électricité

PUBLIÉ SOUS LA DIRECTION DE JACQUES NEIRYNCK

VOLUME V

## ANALYSE ET SYNTHÈSE DES SYSTÈMES LOGIQUES

Daniel Mange



PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES

## TRAITÉ D'ÉLECTRICITÉ

Jusqu'à ce jour, les théories et les applications de l'électricité n'ont jamais fait l'objet d'un exposé systématique et unifié.

Les 22 volumes du Traité d'Electricité comblent cette lacune. Cet ouvrage rassemble de façon cohérente des connaissances jusqu'ici disparates et fragmentaires en fonction de la hiérarchie de modèles de plus en plus généraux, tels que les ingénieurs les ont conçus pour maîtriser des techniques de plus en plus complexes.

### PRÉSENTATION DU VOLUME V

## ANALYSE ET SYNTHÈSE DES SYSTÈMES LOGIQUES

□ Modèles logiques et modes de représentation des systèmes combinatoires □ Synthèse, simplification et analyse des systèmes combinatoires □ Définition des systèmes séquentiels, analyse et modes de représentation des bascules bistables □ Analyse, synthèse et décomposition des compteurs □ Systèmes séquentiels synchrones : analyse et synthèse □ Modèles asynchrones des systèmes combinatoires et séquentiels.

Certains circuits électriques sont caractérisés par des grandeurs physiques (des tensions par exemple) qui n'admettent que deux valeurs significatives, désignées généralement par 0 et 1; ces dispositifs seront décrits dans ce volume par un modèle unique : le système logique asynchrone. Celui-ci est un assemblage de composants ou éléments idéaux qui sont de deux types : l'élément combinatoire et l'élément de délai (ou retard).

L'étude des éléments combinatoires seuls et de leurs assemblages fait l'objet des chapitres 1 et 2 dans lesquels on introduit les rudiments de l'algèbre logique (algèbre de Boole). La synthèse d'un système logique combinatoire, c'est-à-dire sa réalisation à partir d'un certain cahier des charges, est limitée à trois types d'assemblages principaux : les multiplexeurs, les opérateurs NON, ET, OU munis ou non de l'opérateur OU-exclusif.

La bascule bistable est introduite au chapitre 3 : cet élément comporte des boucles de rétroaction et son fonctionnement est séquentiel. On présente alors le modèle quasi-synchrone et le modèle synchrone de la bascule qui sont deux descriptions relativement simplifiées de son comportement.

L'assemblage de bascules bistables et

TRAITÉ D'ÉLECTRICITÉ

**V**  
**ANALYSE ET SYNTHÈSE**  
**DES SYSTÈMES LOGIQUES**

Production .

Presses polytechniques  
et universitaires romandes  
CH-1015 Lausanne

2.3.2004

S-TTC — 73.80

S-HT — 72.10

-HT — 49.70

**TRAITÉ D'ÉLECTRICITÉ**

DE L'ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE  
PUBLIÉ SOUS LA DIRECTION DE JACQUES NEIRYNCK

VOLUME V

**ANALYSE ET SYNTHÈSE  
DES SYSTÈMES LOGIQUES**

par Daniel Mange



PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES

Le Traité d'électricité est édité sous la direction du Professeur Jacques Neiryck

- I INTRODUCTION À L'ÉLECTROTECHNIQUE
- II MATÉRIAUX DE L'ÉLECTROTECHNIQUE
- III ÉLECTROMAGNÉTISME
- IV THÉORIE DES RÉSEAUX DE KIRCHHOFF
- V ANALYSE ET SYNTHÈSE DES SYSTÈMES LOGIQUES
- VI THÉORIE ET TRAITEMENT DES SIGNAUX
- VII DISPOSITIFS À SEMICONDUCTEUR
- VIII ÉLECTRONIQUE
- IX ÉLECTROMÉCANIQUE
- X MACHINES ÉLECTRIQUES
- XI MACHINES SÉQUENTIELLES
- XII ÉNERGIE ÉLECTRIQUE
- XIII HYPERFRÉQUENCES
- XIV CALCULATRICES
- XV ÉLECTRONIQUE DE PUISSANCE
- XVI ÉLECTRONIQUE DE RÉGLAGE ET DE COMMANDE
- XVII SYSTÈMES DE MESURE
- XVIII SYSTÈMES DE TÉLÉCOMMUNICATIONS
- XIX FILTRES ÉLECTRIQUES
- XX TRAITEMENT NUMÉRIQUE DES SIGNAUX
- XXI ÉLECTROACOUSTIQUE
- XXII HAUTE TENSION



Le Traité d'Electricité est une publication des  
**Presses polytechniques et universitaires romandes**,  
fondation scientifique dont le but est principalement la diffusion des travaux de  
l'Ecole polytechnique fédérale de Lausanne.

Le catalogue de ces publications peut être obtenu aux  
**Presses polytechniques et universitaires romandes, CH-1015 Lausanne.**

<http://www.ppur.org>

---

Septième édition  
ISBN 2-88074-045-2

© 2004 Presses polytechniques et universitaires romandes  
CH - 1015 Lausanne  
Tous droits réservés.

Reproduction, même partielle, interdite sous quelque forme ou sur quelque support que ce  
soit sans l'accord écrit de l'éditeur.

Imprimé en Italie  
Imprimerie LegoPrint, Lavis

## INTRODUCTION

“L'enseignement doit être résolument retardataire. Non pas rétrograde, tout au contraire. C'est pour marcher dans le sens direct qu'il prend du recul; car, si l'on ne se place point dans le moment dépassé, comment le dépasser? Ce serait une folle entreprise, même pour un homme dans toute la force, de prendre les connaissances en leur état dernier; il n'aurait point d'élan, ni aucune espérance raisonnable.”

ALAIN, *Propos sur l'éducation*, Presses universitaires de France, Paris, 1965.

### Place du volume V dans le *Traité d'Electricité*

Certains circuits électriques sont caractérisés par des grandeurs physiques (des tensions par exemple) qui n'admettent que deux valeurs significatives (une tension haute, désignée par la valeur 1, et une tension basse, désignée par 0). Ces dispositifs qui, en théorie du moins, peuvent être analysés à l'aide des réseaux de Kirchhoff (vol. IV), seront décrits dans ce volume par un modèle plus dépouillé, le système logique asynchrone; dans celui-ci les variables continues (tension, courant) disparaissent et sont remplacées par des variables discrètes à deux états (0 et 1), sans dimension. Le modèle proposé peut donc s'appliquer à des dispositifs non seulement électriques, mais aussi fluidiques, mécaniques, etc...

Le système logique asynchrone est, comme tout modèle, un assemblage de composants ou éléments idéaux; ceux-ci sont de deux types: l'élément combinatoire, dont les valeurs des variables aux bornes d'entrée définissent à tout instant les valeurs des variables aux bornes de sortie, et l'élément de délai ou retard. Les règles d'assemblage précisent que deux bornes de sortie ne peuvent être reliées à une même borne d'entrée et que dans toute boucle de rétroaction il existe au moins un élément de délai. Dans un système logique asynchrone, le temps est une variable continue; celle-ci devient une variable discrète dans le cas particulier important des systèmes synchrones.

### Organisation générale du volume V

L'étude des éléments combinatoires seuls et de leurs assemblages fait l'objet des chapitres 1 et 2 dans lesquels on introduit les rudiments de l'algèbre logique (algèbre de Boole) ainsi qu'une famille de modes de représentation (tables de vérité, diagrammes de Venn, tables de Karnaugh, logigrammes). La synthèse d'un système logique combinatoire, c'est-à-dire sa réalisation à partir d'un certain cahier des charges, est limitée à trois types d'assemblages principaux: les multiplexeurs, les opérateurs NON, ET, OU munis ou non de l'opérateur OU-exclusif.

Un système logique asynchrone particulier, la bascule bistable, est introduit au chapitre 3; cet élément comporte des boucles de rétroaction et son fonctionnement

est alors séquentiel : les valeurs des variables de sortie ne dépendent pas seulement des valeurs des variables d'entrée au même instant, mais également des valeurs antérieures à cet instant. L'utilisation fréquente de ce composant nous conduit à présenter deux descriptions relativement simplifiées de son comportement qui sont le modèle quasi-synchrone et le modèle synchrone; les modes de représentation des systèmes séquentiels (chronogrammes, tables d'états, tables des transitions et graphes des états) illustrent alors différents types de bascules bistables.

L'assemblage de bascules bistables et d'éléments combinatoires permet de réaliser des compteurs (qui comportent une seule variable indépendante : le signal d'horloge) et des systèmes séquentiels synchrones ou quasi-synchrones (avec plusieurs variables indépendantes) dont l'analyse et la synthèse sont étudiées dans les chapitres 4, 5 et 6 à l'aide des modèles et modes de représentation proposés dans les trois premiers chapitres.

Enfin, l'étude générale des systèmes logiques asynchrones (chapitre 7) repose sur la définition de l'élément de délai inertiel et traite de l'analyse des assemblages quelconques sans boucle de rétroaction, puis avec boucles, ainsi que du cas particulier des bascules bistables et des systèmes séquentiels synchrones ou quasi-synchrones.

En annexe (chapitre 8) on trouvera un rappel sur les systèmes de numération ainsi que l'exposé détaillé d'une méthode de simplification des systèmes combinatoires, celle de McCluskey.

L'étude des systèmes logiques séquentiels est poursuivie dans le volume XI du *Traité*, intitulé *Machines séquentielles*. Elle est basée sur l'algèbre des événements, qui généralise l'algèbre de Boole de façon à prendre en compte la variable temps.

## Conventions

Le *Traité d'Electricité* est composé de volumes (vol.) repérés par un chiffre romain (vol. V). Chaque volume est partagé en chapitres (chap.) repérés par un nombre arabe (chap. 2). Chaque chapitre est divisé en sections (sect.) repérées par deux nombres arabes séparés par un point (sect. 2.3). Chaque section est divisée en paragraphes (§) repérés par trois nombres arabes séparés par deux points (§ 2.3.11). Les références internes stipulent le volume, le chapitre, la section ou le paragraphe du *Traité* auquel on renvoie. Dans le cas de la référence à une partie du même volume, on omet le numéro de celui-ci.

Les références bibliographiques sont numérotées continûment par volume et repérées par un seul nombre arabe entre crochets; les pages concernées sont éventuellement précisées entre parenthèses : [33] (pp. 12-15).

Un terme apparaît en *italique maigre* la première fois qu'il est défini dans le texte. Un passage important est mis en évidence lorsqu'il est composé en *italique gras*.

Un paragraphe délicat ou compliqué est marqué par le signe ■ précédant son repère numérique, tandis que pour les exercices ce même signe peut également annoncer des calculs longs et fastidieux. Un paragraphe qui n'est pas indispensable à la compréhension de ce qui suit est marqué par le signe □ précédant son repère numérique.

Les équations hors-texte sont numérotées continûment par chapitre et repérées par deux nombres arabes placés entre parenthèses et séparés par un point (3.14). Les figures et tableaux sont numérotés continûment par chapitre et repérés par deux nombres arabes précédés de Fig. (Fig. 4.12).

Etant donné le grand nombre des exercices, nous avons calculé les solutions pour un choix de ceux-ci.

## TABLE DES MATIÈRES

	INTRODUCTION .....	v
CHAPITRE 1	MODES DE REPRÉSENTATION DES SYSTÈMES COMBINATOIRES	
	1.1 Modèles logiques .....	1
	1.2 Fonctions d'une variable .....	5
	1.3 Fonctions de plusieurs variables .....	7
	1.4 Opérateurs complets .....	12
	1.5 Systèmes combinatoires universels .....	20
	1.6 Applications .....	25
	1.7 Fonctions incomplètement définies .....	32
	1.8 Conclusions .....	38
CHAPITRE 2	SYNTHÈSE ET ANALYSE DES SYSTÈMES COMBINATOIRES	
	2.1 Préambule .....	41
	2.2 Simplification des expressions algébriques .....	50
	2.3 Fonctions incomplètement définies .....	64
	2.4 Analyse des systèmes combinatoires .....	70
	2.5 Fonctions multiples .....	74
	2.6 Opérateurs OU-exclusif .....	77
	2.7 Systèmes itératifs .....	82
CHAPITRE 3	ANALYSE ET MODES DE REPRÉSENTATION DES BASCULES BISTABLES	
	3.1 Élément de mémoire $\bar{s}\bar{r}$ et systèmes séquentiels .....	87
	3.2 Bascule bistable $SR$ .....	93
	3.3 Modes de représentation analytiques .....	102
	3.4 Modes de représentation synthétiques .....	109
CHAPITRE 4	ANALYSE ET SYNTHÈSE DES COMPTEURS	
	4.1 Diviseurs de fréquence .....	117
	4.2 Compteurs synchrones .....	129
	4.3 Compteurs quasi-synchrones .....	144
	4.4 Décomposition des compteurs .....	148

<b>CHAPITRE 5</b>	<b>ANALYSE DES SYSTÈMES SÉQUENTIELS SYNCHRONES</b>	
5.1	Préambule . . . . .	159
5.2	Systèmes séquentiels synchrones . . . . .	166
5.3	Registres à décalage . . . . .	174
5.4	Réalisation séquentielle des systèmes combinatoires itératifs . . . . .	185
5.5	Systèmes séquentiels quasi-synchrones . . . . .	188
<b>CHAPITRE 6</b>	<b>SYNTHÈSE DES SYSTÈMES SÉQUENTIELS SYNCHRONES</b>	
6.1	Conception des systèmes à comportement asynchrone . .	195
6.2	Réalisation des systèmes à comportement asynchrone . .	205
6.3	Conception des systèmes à comportement synchrone . . .	218
6.4	Réalisation des systèmes à comportement synchrone . . .	223
6.5	Réalisation avec des registres à décalage . . . . .	231
<b>CHAPITRE 7</b>	<b>MODÈLES ASYNCHRONES DES SYSTÈMES LOGIQUES</b>	
7.1	Préambule . . . . .	241
7.2	Systèmes combinatoires . . . . .	244
7.3	Systèmes séquentiels . . . . .	257
7.4	Bascules bistables . . . . .	265
7.5	Systèmes séquentiels synchrones et quasi-synchrones . . .	277
<b>CHAPITRE 8</b>	<b>ANNEXES</b>	
8.1	Systèmes de numération . . . . .	283
8.2	Méthode de simplification de McCluskey . . . . .	286
	<b>SOLUTIONS DES EXERCICES . . . . .</b>	<b>297</b>
	<b>BIBLIOGRAPHIE . . . . .</b>	<b>325</b>
	<b>INDEX ANALYTIQUE . . . . .</b>	<b>331</b>
	<b>FORMULAIRE . . . . .</b>	<b>341</b>

## MODES DE REPRÉSENTATION DES SYSTÈMES COMBINATOIRES

### 1.1 MODÈLES LOGIQUES

#### 1.1.1 Introduction

Trois hypothèses simplificatrices permettent de décrire certains systèmes concrets à l'aide de grandeurs dépendant du temps et n'ayant que deux états physiques distincts. Le chapitre 1 est consacré à l'étude d'éléments dont les états de sortie dépendent uniquement des états d'entrée mesurés au même instant. On définit la fonction logique d'un tel élément comme étant le tableau de correspondance entre les états d'entrée et les états de sortie; divers modes de représentation de ces fonctions sont introduits : tables, diagrammes de Venn et expressions algébriques. L'algèbre logique introduite à cet effet repose sur un nombre restreint de postulats dont il découle des théorèmes qui sont démontrés et illustrés par des applications pratiques.

#### 1.1.2 Définition

On appelle *système concret* tout objet physique comportant un nombre fini d'accès ou bornes; certaines d'entre elles sont les bornes d'entrée, les autres sont les bornes

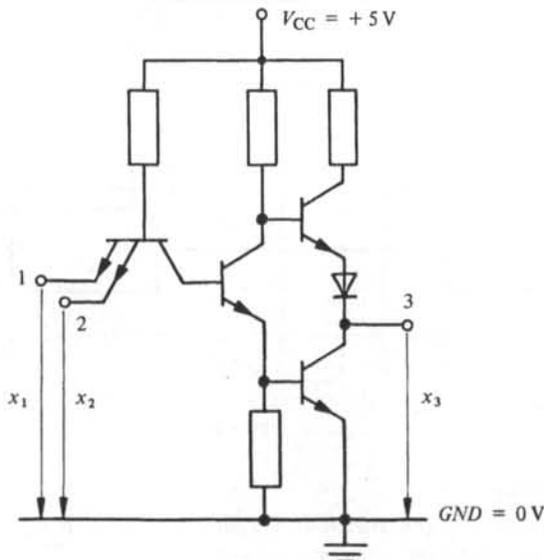


Fig. 1.1

de sortie et l'état physique des premières détermine de façon plus ou moins complexe l'état des secondes en fonction du temps.

### 1.1.3 Exemple

Le schéma électronique de la figure 1.1 présente un système concret comportant deux bornes d'entrée (désignées par 1 et 2) et une borne de sortie (désignée par 3).

### 1.1.4 Expérience

La figure 1.2 décrit la variation en fonction du temps  $t$  des tensions électriques  $x_1(t)$ ,  $x_2(t)$  et  $x_3(t)$  mesurées entre les trois bornes du schéma de la figure 1.1 et la terre; ce diagramme temporel ou *chronogramme* constitue le protocole d'une expérience. On constate que :

- les tensions d'entrée  $x_1$  et  $x_2$  ne prennent au cours du temps que deux états physiques distincts : 0V et +5V; ces deux états seront simplement désignés par les symboles 0 et 1;
- la tension de sortie  $x_3$  varie de façon continue et prend au cours du temps une infinité d'états physiques distincts.

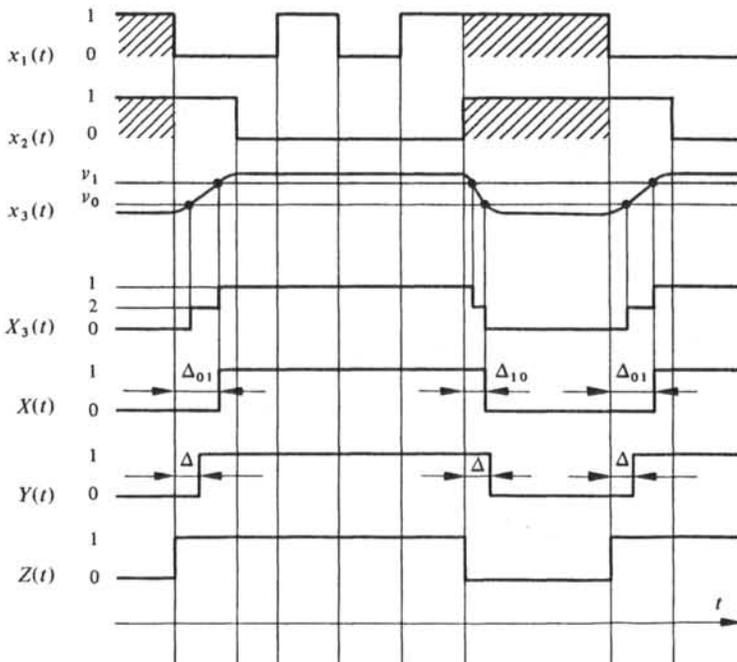


Fig. 1.2

### 1.1.5 Commentaire

Pour décrire le comportement du système concret de la figure 1.1, on cherche à établir une relation entre les variations des entrées et celles de la sortie; une telle relation serait facilitée si la tension de sortie  $x_3$  se présentait sous la forme des tensions d'entrée

$x_1$  ou  $x_2$ , c'est-à-dire si elle ne comportait que deux états 0 et 1. L'introduction de trois hypothèses simplificatrices permettra d'aboutir au résultat recherché : la représentation finale de  $x_3$  sera une traduction approximative de la réalité physique ou modèle du système concret.

### 1.1.6 Première hypothèse

En admettant l'existence de deux tensions constantes  $v_0$  et  $v_1$  (fig. 1.2) on peut déterminer à chaque instant une variable  $X_3(t)$  définie de la façon suivante :

- $X_3(t) = 0$  si  $x_3(t) \leq v_0$
- $X_3(t) = 1$  si  $x_3(t) \geq v_1$
- $X_3(t) = 2$  si  $v_0 < x_3(t) < v_1$

La variable  $X_3$  peut prendre trois valeurs distinctes ou états :  $X_3$  est un signal *discret* ou *quantifié*; la description de  $x_3$  par  $X_3$  est une *quantification* de  $x_3$ .

On constate dans la figure 1.2 que l'état  $X_3 = 2$  apparaît transitoirement, lors du passage direct ou inverse des états  $X_3 = 0$  à  $X_3 = 1$  : cette constatation suggère une deuxième hypothèse.

### 1.1.7 Deuxième hypothèse

En admettant que la durée des variations des entrées est nettement supérieure à la durée de l'état transitoire caractérisé par  $X_3 = 2$ , on peut négliger la représentation de cet état : la variable  $X(t)$  (fig. 1.2) illustre une telle simplification.

On constate alors que le signal  $X(t)$  ne comporte que deux états 0 et 1; on constate également l'existence de *retards* ou *délais* séparant l'action sur les entrées (variations de  $x_1$  et  $x_2$ ) des effets sur la sortie (variations de  $X$ ) : on remarque en particulier que le délai  $\Delta_{01}$  lors d'une variation de  $X$  de 0 à 1 n'est pas égal au délai  $\Delta_{10}$  de la variation inverse.

### 1.1.8 Troisième hypothèse

En admettant que les délais  $\Delta_{01}$  et  $\Delta_{10}$  sont égaux à une constante  $\Delta$ , indépendante du temps, la variable  $X(t)$  peut être remplacée par une nouvelle variable  $Y(t)$  (fig. 1.2).

### 1.1.9 Définition : modèle logique asynchrone

En admettant successivement les trois hypothèses de la quantification, de l'élimination des transitoires et de l'égalisation des délais on obtient un modèle particulier du système concret étudié : c'est le *modèle logique asynchrone* auquel se réfère l'ensemble du présent volume.

### 1.1.10 Définition : modèle logique combinatoire

En admettant que le délai  $\Delta$  de  $Y(t)$  est nul, on obtient la variable  $Z(t)$  (fig. 1.2) qui décrit un cas particulier très important du modèle asynchrone : c'est le *modèle logique combinatoire*.

### 1.1.11 Définitions : états

Chacune des  $2^2 = 4$  combinaisons possibles des valeurs des deux entrées (ou variables d'entrée)  $x_1$  et  $x_2$  est appelée un *état d'entrée* et sera symbolisée par  $(x_1, x_2)$ . De façon plus générale, chacune des  $2^n$  combinaisons des valeurs de  $n$  entrées  $x_1, x_2, \dots, x_n$  est un état d'entrée  $(x_1, x_2, \dots, x_n)$ .

Par analogie, chacune des  $2^1 = 2$  valeurs de la sortie (ou variable de sortie)  $Z$  est un *état de sortie*. De façon plus générale, chacune des  $2^r$  combinaisons des valeurs de  $r$  sorties  $Z_1, Z_2, \dots, Z_r$  est un état de sortie  $(Z_1, Z_2, \dots, Z_r)$ .

### 1.1.12 Définitions : élément combinatoire et fonction logique

L'examen des variables  $x_1, x_2$  et  $Z$  de la figure 1.2 permet de mesurer pour chacun des quatre états d'entrée  $(x_1, x_2)$  l'état de sortie  $Z$  correspondant. Le résultat de cette mesure est présenté dans le tableau ou *table de vérité* de la figure 1.3. On constate que pour chaque état d'entrée  $(x_1, x_2)$  il existe un unique état de sortie  $Z$  : l'état de sortie à un instant  $t$  est entièrement déterminé par l'état d'entrée à ce même instant. On appelle *élément combinatoire* tout système idéal décrit par un tel modèle (fig. 1.4). La table de vérité de la figure 1.3 définit une application qui est la *fonction logique* de cet élément.

$x_1$	$x_2$	$Z$
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 1.3

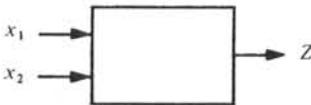


Fig. 1.4

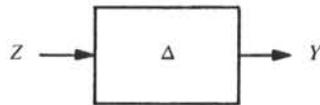


Fig. 1.5

### 1.1.13 Définition : élément de délai

L'examen des variables  $Y$  et  $Z$  de la figure 1.2 suggère la définition d'un système idéal ou *élément de délai* caractérisé par un retard  $\Delta$  et représenté par le schéma de la figure 1.5.

### 1.1.14 Définitions : systèmes logiques asynchrone et combinatoire

La théorie des systèmes logiques traite des assemblages que l'on peut former avec les deux types d'éléments idéaux définis ci-dessus : les éléments combinatoires et les éléments de délai. L'assemblage le plus général peut être représenté par la figure 1.6 et constitue un *système logique asynchrone* dont l'étude fera l'objet du chapitre 7. L'assemblage d'éléments combinatoires sans éléments de délai est appelé *système logique combinatoire*.

et sera étudié dans les chapitres 1 et 2. Un cas particulier de système asynchrone, la bascule bistable, sera introduit au chapitre 3; l'assemblage de bascules bistables et d'éléments combinatoires permettra d'étudier les compteurs et les systèmes séquentiels synchrones qui seront traités dans les chapitres 4, 5 et 6.

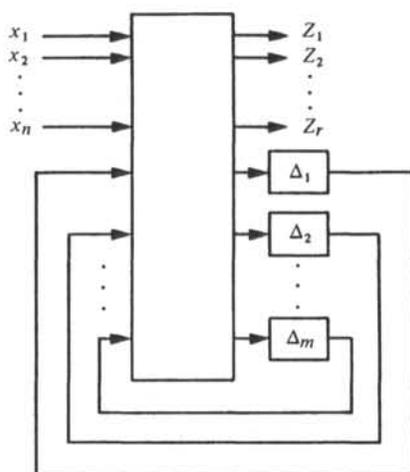


Fig. 1.6

## 1.2 FONCTIONS D'UNE VARIABLE

### 1.2.1 Description

Le système combinatoire le plus simple comporte une entrée  $a$  et une sortie  $Z$  (fig. 1.7) : il est possible d'énumérer toutes les fonctions logiques (au sens du § 1.1.12) d'un tel système en recourant à la table de vérité de la figure 1.8. Celle-ci présente autant de lignes que d'états d'entrée, soit deux. Pour chacun de ces états, la sortie  $Z$  peut prendre la valeur 0 ou 1 : on définit ainsi un ensemble de  $2^2 = 4$  fonctions d'une variable.



Fig. 1.7

$a$	$Z_0$	$Z_1$	$Z_2 = \bar{a}$	$Z_3$	$\bar{a}$
0	0	0	1	1	0
1	0	1	0	1	1

Fig. 1.8

### 1.2.2 Définitions : fonction NON

L'examen des quatre fonctions de la figure 1.8 met en évidence ce qui suit :

- $Z_0 = 0$
- $Z_3 = 1$
- $Z_1 = a$

Les fonctions  $Z_0$  et  $Z_3$  sont les deux *constantes logiques*; la fonction  $Z_1$  est l'*identité* ou la *forme vraie* de  $a$ .

La fonction  $Z_2$  est obtenue par permutation des valeurs de la variable  $a$ . Par définition on écrit alors :

$$Z_2 = \bar{a} \quad (1.1)$$

qui se traduit en disant que  $Z_2$  est le *complément*, la *négation*, l'*inverse* de  $a$  ou encore *a-barre*; on dit aussi que  $\bar{a}$  est la *forme complémentaire* de  $a$ .

La fonction  $Z_2$  est appelée *fonction NON*; elle peut être représentée à l'aide d'un *diagramme de Venn*. Un tel diagramme (fig. 1.9) fait apparaître dans un plan de référence deux régions (l'intérieur et l'extérieur du cercle) correspondant aux deux états d'entrée ( $a = 1$  et  $a = 0$ ). Il suffit de hachurer la région dans laquelle la fonction  $Z_2 = \bar{a} = 1$  pour obtenir la représentation désirée (fig. 1.10).

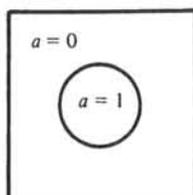
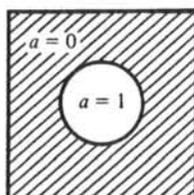


Fig. 1.9



$$Z_2 = \bar{a}$$

Fig. 1.10

### 1.2.3 Théorème

L'examen de la table de vérité (fig. 1.8) permet de vérifier la relation suivante :

$$\overline{(\bar{a})} = \bar{\bar{a}} = a \quad (1.2)$$

Il en découle que  $\bar{\bar{a}} \equiv a$ .

### 1.2.4 Définitions

On appelle *porte NON*, *opérateur NON* ou *inverseur* tout système combinatoire qui réalise la fonction NON.

De façon plus générale, on appellera *porte x* ou *opérateur x* tout système combinatoire réalisant la fonction  $x$ .

### 1.2.5 Logigrammes

Les schémas logiques ou *logigrammes* suivants représentent une porte NON selon trois systèmes de normes :

- les normes de la Commission électrotechnique internationale (CEI) [1] qui sont internationalement reconnues (fig. 1.11);
- les normes américaines MIL - STD - 806 B (MIL) [2] [3] très fréquemment utilisées dans la pratique (fig. 1.12);
- les normes allemandes DIN 40700, édition 1963 (fig. 1.13).

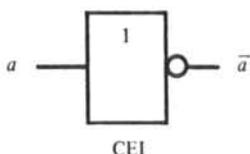


Fig. 1.11

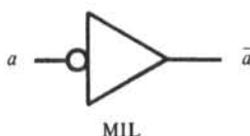


Fig. 1.12

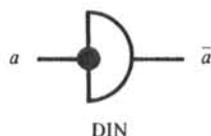
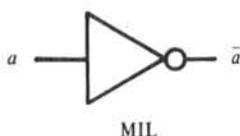


Fig. 1.13

### 1.3 FONCTIONS DE PLUSIEURS VARIABLES

#### 1.3.1 Description

L'ensemble des fonctions de deux variables est représenté par la table de vérité de la figure 1.14. On dénombre  $2^2 = 4$  états d'entrée ( $a, b$ ), soit quatre lignes de la table; on peut définir  $2^4 = 16$  fonctions  $Z$  distinctes, soit seize colonnes de la table. L'examen de ces fonctions met en évidence ce qui suit :

- $Z_0 = 0$  (constante)
- $Z_{15} = 1$  (constante)
- $Z_3 = a$  (identité)
- $Z_5 = b$  (identité)

D'autre part :  $Z_8 = \bar{Z}_7$  ;  $Z_9 = \bar{Z}_6$  ;  $Z_{10} = \bar{Z}_5$  ;  $Z_{11} = \bar{Z}_4$  ;  $Z_{12} = \bar{Z}_3$  ;  $Z_{13} = \bar{Z}_2$  ;  $Z_{14} = \bar{Z}_1$  ;  $Z_{15} = \bar{Z}_0$ . On peut donc calculer les huit fonctions  $Z_8, Z_9, \dots, Z_{15}$  par une simple négation des huit fonctions  $Z_7, Z_6, \dots, Z_0$ .

$a$	$b$	$Z_0$	$Z_1 = ab$	$Z_2$	$Z_3$	$Z_4$	$Z_5$	$Z_6$	$Z_7 = a + b$	$Z_8$	$Z_9$	$Z_{10}$	$Z_{11}$	$Z_{12}$	$Z_{13}$	$Z_{14}$	$Z_{15}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Fig. 1.14

### 1.3.2 Commentaire

Pour  $n$  variables, on dénombre  $2^n$  états d'entrée et l'on définit  $2^{2^n}$  fonctions différentes. On obtient ainsi [14] (p. 19) :

- $n = 3$  ;  $2^n = 8$  ;  $2^{2^n} = 256$
- $n = 4$  ;  $2^n = 16$  ;  $2^{2^n} = 65'536$
- $n = 5$  ;  $2^n = 32$  ;  $2^{2^n} = 4'294'967'296$

### 1.3.3 Définitions : fonction ET

Par définition on écrit :

$$Z_1 = a \cdot b = ab \quad (1.3)$$

qui se traduit en disant que  $Z_1$  est le *produit logique* ou *intersection* de  $a$  et  $b$ . La fonction  $Z_1$  est appelée *fonction ET*, traduisant le fait que  $Z_1 = 1$  si  $a$  ET  $b$  valent simultanément 1 (fig. 1.14).

Le diagramme de Venn d'une fonction de deux variables (fig. 1.15) attribue à chacune d'elles un cercle : les deux cercles se coupent et déterminent, à l'intérieur du plan de référence, quatre régions distinctes correspondant aux quatre états d'entrée  $a, b$ . En hachurant la région dans laquelle  $Z_1 = ab = 1$ , on obtient la représentation de la figure 1.16 qui met en évidence l'intersection des variables  $a$  et  $b$  représentées séparément dans la figure 1.15.

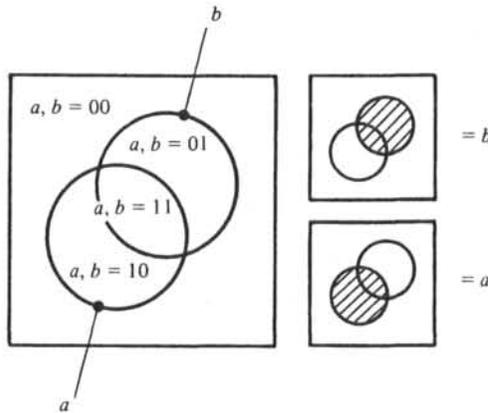
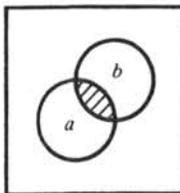
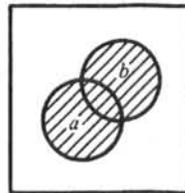


Fig. 1.15



$$Z_1 = ab$$

Fig. 1.16



$$Z_1 = a + b$$

Fig. 1.17

Enfin, les logigrammes des figures 1.18 à 1.20 montrent une porte ET dans les trois normes CEI, MIL et DIN.

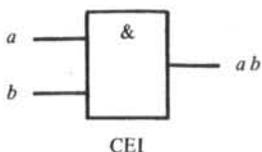


Fig. 1.18

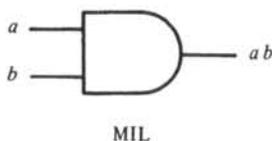


Fig. 1.19

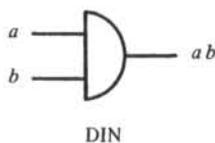


Fig. 1.20

### 1.3.4 Définitions : fonction OU

Par définition on écrit :

$$Z_7 = a + b \tag{1.4}$$

qui se traduit en disant que  $Z_7$  est la *somme logique* ou *réunion* ou encore *union* de  $a$  et  $b$ . La fonction  $Z_7$  est appelée *fonction OU* (parfois *OU inclusif*), traduisant le fait que  $Z_7 = 1$  si  $a$  OU  $b$  (c'est-à-dire l'une ou l'autre ou encore les deux variables simultanément) valent 1 (fig. 1.14).

En hachurant les régions du diagramme de Venn dans lesquelles  $Z_7 = a + b = 1$ , on obtient une représentation de la fonction OU (fig. 1.17) qui met en évidence l'union des variables  $a$  et  $b$  représentées séparément dans la figure 1.15. Enfin, les divers logigrammes d'une porte OU sont donnés par les figures 1.21 à 1.23.

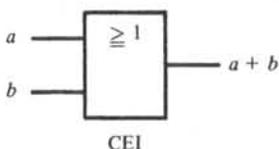


Fig. 1.21

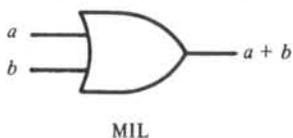


Fig. 1.22

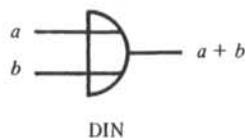


Fig. 1.23

### 1.3.5 Commentaire

Les logigrammes CEI sont des rectangles à l'intérieur desquels un symbole ( $\&$ ,  $\geq 1$ ) définit la fonction réalisée (ET, OU). Les logigrammes MIL et DIN indiquent par leur forme géométrique particulière (fig. 1.19, 1.20, 1.22 et 1.23) la fonction correspondante. L'expérience montre que si l'apprentissage de ces dernières normes est plus long que celui des normes CEI, par contre la lisibilité (c'est-à-dire la faculté de distinguer une fonction particulière dans un logigramme complexe) est supérieure.

Bien que les symboles graphiques du *Traité d'Electricité* soient ceux de la Commission électrotechnique internationale (CEI), on utilisera principalement dans ce volume, pour les raisons didactiques exposées plus haut, les normes américaines MIL. Soulignons enfin que la dernière version de celles-ci (IEEE Std 91-1973; ANSI Y32.14-1973) [3] reconnaît à la fois les logigrammes rectangulaires de la CEI et les logigrammes de forme MIL.

### 1.3.6 Théorèmes de commutativité

$$ab = ba \ ; \ a + b = b + a \quad (1.5)$$

Si  $a = b$ , la relation est vérifiée trivialement; si  $a \neq b$ , on vérifie dans la table de vérité (fig. 1.14) que  $Z_1 = ab = 0 \cdot 1 = 1 \cdot 0 = 0$  et que  $Z_7 = a + b = 0 + 1 = 1 + 0 = 1$ .

### 1.3.7 Théorèmes d'idempotence

$$a \cdot a = a \ ; \ a + a = a \quad (1.6)$$

On vérifie ces relations en imposant  $b = a$  dans la table de  $Z_1$  et  $Z_7$  (fig. 1.14); il en découle une forme généralisée de (1.6) :

$$a \cdot a \cdot \dots \cdot a = a \ ; \ a + a + \dots + a = a \quad (1.7)$$

### 1.3.8 Théorèmes des constantes

$$a \cdot 0 = 0 \ ; \ a + 0 = a \quad (1.8)$$

$$a \cdot 1 = a \ ; \ a + 1 = 1 \quad (1.9)$$

On impose  $b = 0$  ( $b = 1$ ) dans la table de  $Z_1$  et  $Z_7$  (fig. 1.14).

### 1.3.9 Théorèmes de complémentation

$$a \cdot \bar{a} = 0 \ ; \ a + \bar{a} = 1 \quad (1.10)$$

On impose  $b = \bar{a}$  dans la table de  $Z_1$  et  $Z_7$  (fig. 1.14).

### 1.3.10 Théorèmes de distributivité

On affirme que la fonction ET est distributive par rapport à la fonction OU :

$$a(b + c) = ab + ac \quad (1.11)$$

De même, la fonction OU est distributive par rapport à la fonction ET :

$$a + bc = (a + b)(a + c) \quad (1.12)$$

$a$	$b$	$c$	$b + c$	$ab$	$ac$	$a(b+c) = ab+ac$	$bc$	$a + b$	$a + c$	$a + bc = (a+b)(a+c)$
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	1	1
1	0	1	1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Fig. 1.24

A partir d'une table de vérité (fig. 1.24) comportant huit états  $a, b, c$  il est possible de calculer les valeurs de chacun des deux membres des relations (1.11) et (1.12). En vérifiant l'identité de ces deux membres dans les huit états, on démontre les théorèmes proposés.

### 1.3.11 Commentaire

Le calcul algébrique des expressions logiques admet, comme dans l'algèbre classique, que les produits sont toujours effectués avant les sommes. Il en découle la possibilité de supprimer toutes les parenthèses contenant un produit ainsi que le démontrent, par exemple, les relations (1.11) et (1.12).

### 1.3.12 Exemple

On cherche à démontrer la relation :

$$a + \bar{a}b = a + b \tag{1.13}$$

Dans une table de vérité à quatre états, il serait possible de vérifier cette relation en calculant ligne par ligne chacun des deux membres. La connaissance d'un certain nombre de théorèmes, vérifiés précédemment, permet une démonstration différente selon le schéma suivant :

- relation (1.12) :  $a + \bar{a}b = (a + \bar{a})(a + b)$
- relation (1.10) :  $(a + \bar{a})(a + b) = 1 \cdot (a + b)$
- relation (1.5) :  $1 \cdot (a + b) = (a + b) \cdot 1$
- relation (1.9) :  $(a + b) \cdot 1 = a + b$

### 1.3.13 Définitions

Les relations (1.5) à (1.12) ont été vérifiées par inspection systématique des tables de vérité correspondantes, tandis que la relation (1.13) a été démontrée par l'intervention successive de plusieurs théorèmes déjà vérifiés. Dans le premier cas on parle de *démonstration tabulaire*, dans le second cas de *démonstration algébrique*. Enfin, on appelle généralement *algèbre logique* ou *algèbre de Boole* (du nom de son créateur G. Boole [5]) l'algèbre des trois opérations NON, ET, OU appliquées aux éléments de l'ensemble  $\{0,1\}$ .

### 1.3.14 Théorèmes d'associativité

$$a(bc) = (ab)c = abc \tag{1.14}$$

$$a + (b + c) = (a + b) + c = a + b + c \tag{1.15}$$

La démonstration tabulaire est laissée au soin du lecteur.

### 1.3.15 Théorèmes du consensus

$$ax + b\bar{x} + ab = ax + b\bar{x} \tag{1.16}$$

$$(a + x)(b + \bar{x})(a + b) = (a + x)(b + \bar{x}) \tag{1.17}$$

La démonstration tabulaire est également laissée au soin du lecteur.

### 1.3.16 Exercice

On donne deux exemples de systèmes concrets (fig. 1.25). Est-il possible de décrire ces systèmes par un modèle logique combinatoire et si oui, quelle est la fonction réalisée par chacun d'eux ?

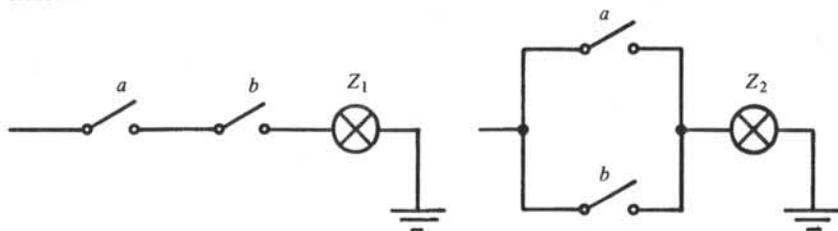


Fig. 1.25

### 1.3.17 Exercice

On demande la démonstration algébrique des relations suivantes :

$$a(a + b) = a \quad (1.18)$$

$$(a + b)(a + c) = a + bc \quad (1.19)$$

$$a + ab = a \quad (1.20)$$

$$ab + \bar{a} = \bar{a} + b \quad (1.21)$$

### 1.3.18 Exercice

En admettant le symbolisme bidimensionnel de Blanchard [97] (pp. 19 - 37) défini par :

$$a \cdot b = |ab|, \quad a + b = \begin{vmatrix} a \\ b \end{vmatrix} \quad (1.22)$$

on demande de vérifier l'écriture des théorèmes suivants :

- distributivité du ET par rapport au OU :  $a \cdot \begin{vmatrix} b \\ c \end{vmatrix} = \begin{vmatrix} a \cdot b \\ a \cdot c \end{vmatrix}$
- distributivité du OU par rapport au ET :  $\begin{vmatrix} a \\ b \cdot c \end{vmatrix} = \begin{vmatrix} a & a \\ b & c \end{vmatrix}$
- consensus :  $\begin{vmatrix} \bar{x} \cdot b \\ a \cdot x \end{vmatrix} = \begin{vmatrix} \bar{x} & b \\ a & x \end{vmatrix}$

## 1.4 OPÉRATEURS COMPLETS

### 1.4.1 Théorèmes de De Morgan

L'inverse d'un produit de deux variables est égal à la somme des inverses de ces variables; l'inverse d'une somme de deux variables est égal au produit des inverses de ces variables.

$$\overline{a \cdot b} = \bar{a} + \bar{b} \quad (1.23)$$

$$\overline{a + b} = \bar{a} \cdot \bar{b} \quad (1.24)$$

La démonstration tabulaire est donnée par une table de vérité (fig. 1.26).

$a$	$b$	$\bar{a}$	$\bar{b}$	$ab$	$\overline{ab} = \bar{a} + \bar{b}$	$a + b$	$\overline{a + b} = \bar{a} \cdot \bar{b}$
0	0	1	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0
1	1	0	0	1	0	1	0

Fig. 1.26

### 1.4.2 Corollaires

$$a \cdot b = \overline{\bar{a} + \bar{b}} \tag{1.25}$$

$$a + b = \overline{\bar{a} \cdot \bar{b}} \tag{1.26}$$

La démonstration algébrique découle du théorème (1.2) qui permet d'écrire :

$$a \cdot b = \overline{\bar{a} \cdot \bar{b}} \quad ; \quad a + b = \overline{\bar{a} + \bar{b}} \tag{1.27}$$

En remplaçant dans (1.27)  $\overline{\bar{a} \cdot \bar{b}}$  et  $\overline{\bar{a} + \bar{b}}$  par leurs expressions (1.23) et (1.24) on obtient les corollaires (1.25) et (1.26).

### 1.4.3 Transformation de De Morgan

Les théorèmes de De Morgan décrivent une transformation générale des expressions algébriques qui peut se résumer ainsi :

- à tout produit logique du premier membre correspond une somme logique dans le second membre, à toute somme logique du premier membre correspond un produit logique dans le second membre;
- à toute grandeur logique du premier membre correspond la grandeur complémentée dans le second membre.

Les théorèmes de De Morgan peuvent être appliqués à des produits logiques (des sommes logiques) comportant un nombre quelconque de variables :

$$\overline{a \cdot b \cdot c \cdot \dots \cdot z} = \bar{a} + \bar{b} + \bar{c} + \dots + \bar{z} \tag{1.28}$$

$$\overline{a + b + c + \dots + z} = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \dots \cdot \bar{z} \tag{1.29}$$

### 1.4.4 Exemple

Il faut prendre quelques précautions lors de l'application des théorèmes de De Morgan à des expressions algébriques quelconques. Considérons par exemple :

$$Z = \overline{a + b + \bar{c}d} \tag{1.30}$$

Une judicieuse organisation des parenthèses permet d'appliquer correctement les théorèmes (1.28) et (1.29) :

$$\begin{aligned} Z &= \overline{a + b + (\overline{c d})} = \bar{a} \cdot \bar{b} \cdot \overline{(\overline{c d})} = \bar{a} \cdot \bar{b} \cdot (\overline{\overline{c + d}}) = \bar{a} \cdot \bar{b} \cdot (c + d) \\ &= \bar{a} \bar{b} c + \bar{a} \bar{b} d \end{aligned} \quad (1.31)$$

On a donc utilisé deux fois la transformation de De Morgan.

#### 1.4.5 Définitions

Un produit de plusieurs variables apparaissant chacune sous la forme vraie ou sous la forme complémentaire s'appelle un *monôme*. Une somme de produits ou monômes est généralement appelée *polynôme*. L'apparition d'une variable sous la forme vraie ou complémentaire est une *lettre*.

L'exemple du paragraphe 1.4.4 traduit la transformation d'une expression algébrique quelconque (1.30) en un polynôme (1.31); chacun des deux monômes de ce polynôme comporte trois lettres.

#### 1.4.6 Définition : fonction NAND

La *fonction NON-ET* (en anglais : *NAND*) est définie algébriquement par l'expression :

$$Z = a \uparrow b = \overline{a \cdot b} = \bar{a} + \bar{b} \quad (1.32)$$

La table de vérité de la fonction NAND est représentée dans la figure 1.26 ainsi que dans la figure 1.14 (fonction  $Z_{14}$ ).

#### 1.4.7 Définitions : expressions et systèmes équivalents

On a déjà vu que tout élément combinatoire et, par extension, tout système combinatoire sont décrits par une table de vérité (§ 1.1.12) : cette table définit la fonction du système étudié. On dit alors qu'un système combinatoire réalise une certaine fonction.

Lorsque deux systèmes combinatoires distincts réalisent la même fonction, on dit que ces systèmes sont *fonctionnellement équivalents*.

Les diverses expressions algébriques qui décrivent la même fonction sont dites *équivalentes* ou *égales*, ce qui permet d'écrire des relations telles que (1.32) :  $\overline{a \cdot b} = \bar{a} + \bar{b}$ .

#### 1.4.8 Logigrammes

Les deux logigrammes MIL de la figure 1.27 représentent deux systèmes fonctionnellement équivalents : ces logigrammes traduisent en effet les expressions algébriques

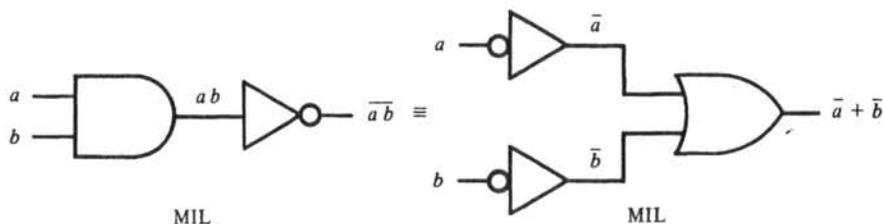


Fig. 1.27

équivalentes (1.32) et réalisent chacun la fonction NAND. Etant donné l'utilisation très fréquente de la porte NAND, on admet en général les logigrammes simplifiés des figures 1.28 (MIL) et 1.29 (CEI).

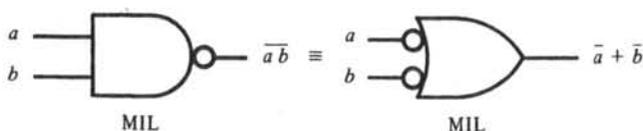


Fig. 1.28

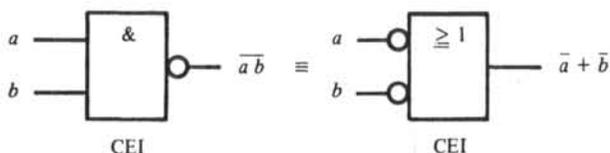


Fig. 1.29

### 1.4.9 Description

Il est possible de calculer les fonctions réalisées par les logigrammes NAND des figures 1.30 à 1.32. En utilisant la relation (1.32) il vient :

- $Z_a = \overline{a \cdot 1} = \bar{a}$  (fonction NON : fig. 1.30)
- $Z_b = \overline{a \bar{b} \cdot 1} = \overline{a \bar{b}} = a b$  (fonction ET : fig. 1.31)
- $Z_c = \overline{a \cdot 1} + \overline{b \cdot 1} = \bar{a} + \bar{b} = a + b$  (fonction OU : fig. 1.32)

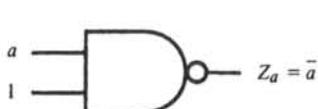


Fig. 1.30

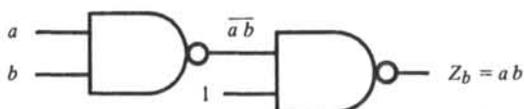


Fig. 1.31

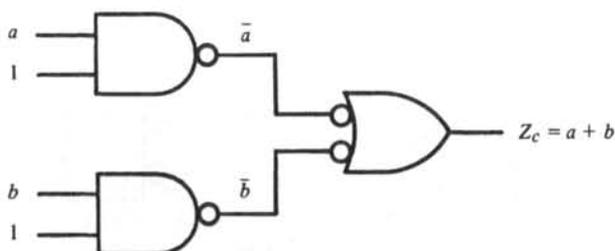


Fig. 1.32

On constate que ces trois assemblages de portes NAND réalisent les fonctions NON, ET et OU. On peut donc exprimer algébriquement ces trois fonctions à l'aide des seuls symboles  $\{0, 1, a, b, \uparrow\}$  :

$$Z_a = \bar{a} = a \uparrow 1 \tag{1.33}$$

$$Z_b = ab = (a \uparrow b) \uparrow 1 \quad (1.34)$$

$$Z_c = a + b = (a \uparrow 1) \uparrow (b \uparrow 1) \quad (1.35)$$

#### 1.4.10 Définition : opérateur complet

On appelle *opérateur complet* tout opérateur (porte) dont trois assemblages distincts permettent de réaliser les trois fonctions NON, ET et OU.

Le calcul précédent (§ 1.4.9) démontre que la porte NAND est un opérateur complet; les trois systèmes des figures 1.30 à 1.32 sont fonctionnellement équivalents aux portes NON, ET et OU.

#### 1.4.11 Définition : fonction NOR

La fonction *NON-OU* (en anglais *NOR*) est définie algébriquement par l'expression :

$$Z = a \downarrow b = \overline{a + b} = \bar{a} \cdot \bar{b} \quad (1.36)$$

La table de vérité de la fonction NOR apparaît dans les figures 1.26 et 1.14 (fonction  $Z_8$ ). Les deux logigrammes MIL de la figure 1.33 réalisent la fonction NOR, de même que les logigrammes simplifiés des figures 1.34 (MIL) et 1.35 (CEI).

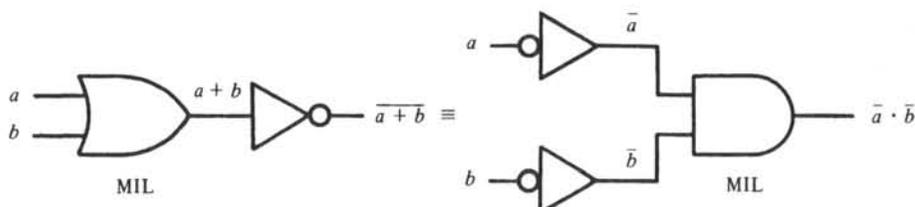


Fig. 1.33

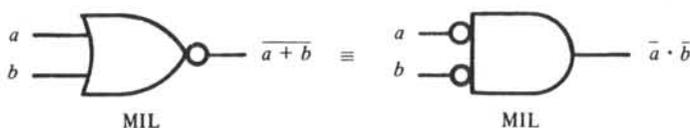


Fig. 1.34

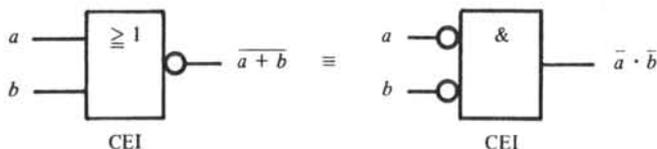


Fig. 1.35

#### 1.4.12 Exercice

Effectuer la démonstration algébrique des relations suivantes :

- $\overline{abcd} = \bar{a} + \bar{b} + \bar{c} + \bar{d}$  ;  $\overline{a + b + c + d} = \bar{a}\bar{b}\bar{c}\bar{d}$  (application des théorèmes de De Morgan aux fonctions de quatre variables et définition des fonctions NAND et NOR de quatre variables);

- $(a \uparrow b) \uparrow c \neq a \uparrow (b \uparrow c)$  ;  $(a \downarrow b) \downarrow c \neq a \downarrow (b \downarrow c)$  (les fonctions NAND et NOR ne sont pas associatives) ;
- $a \uparrow a = \bar{a}$  (réalisation de la fonction NON à l'aide d'une porte NAND).

On tracera enfin les logigrammes réalisant l'ensemble des relations calculées.

### 1.4.13 Exercice

A l'aide des théorèmes de De Morgan, on demande de transformer les expressions suivantes en polynômes :

$$Z_1 = \overline{ab + cd + ef} ; \quad Z_2 = \overline{abc + d + efg} ;$$

$$Z_3 = \overline{ab \cdot cd + ef} + gh ; \quad Z_4 = \overline{ab \cdot (cde + fg)}.$$

### 1.4.14 Exercice

Démontrer que la porte NOR est un opérateur complet au sens du paragraphe 1.4.10. A partir des logigrammes NOR réalisant les fonctions NON, ET et OU on demande d'exprimer algébriquement chacune de ces trois fonctions à l'aide des seuls symboles  $\{0, 1, a, b, \downarrow\}$ . Calculer enfin l'expression algébrique de la fonction NAND  $a \bar{b} c$  à l'aide des seuls symboles  $\{0, 1, a, b, c, \downarrow\}$ .

### 1.4.15 Transformation de logigrammes

Les théorèmes de De Morgan (1.23) (1.24) et leurs corollaires (1.25) (1.26) expriment des égalités algébriques; il en découle que les fonctions correspondantes sont réalisées par des logigrammes équivalents (fig. 1.36 à 1.39).

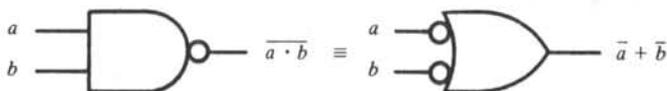


Fig. 1.36

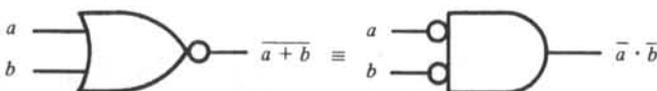


Fig. 1.37

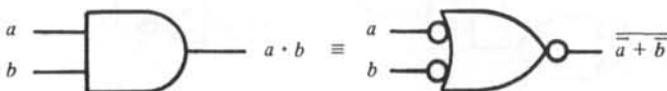


Fig. 1.38

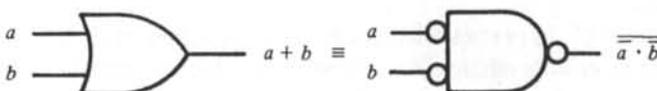


Fig. 1.39

L'expérience montre que les fonctions logiques qu'on cherche à réaliser s'expriment généralement à l'aide des opérations NON, ET et OU; on constate aussi que les opérateurs complets (NAND ou NOR) offrent des performances techniques et économiques meilleures que les opérateurs NON, ET, OU. L'intérêt de réaliser des expressions NON, ET, OU à l'aide d'opérateurs NAND (ou NOR) est alors évident; le calcul algébrique de ces transformations est souvent fastidieux : il se ramène à l'utilisation répétée des théorèmes de De Morgan [4]. On donne ci-après le principe d'une méthode graphique basée sur l'équivalence des logigrammes des figures 1.36 à 1.39 [11] (pp. 186-195) [12] (pp. 131-154).

#### 1.4.16 Exemple

On donne l'expression :

$$Z = [ab + cd + e + f(\bar{g} + h)]\bar{i} \quad (1.37)$$

Le logigramme de la figure 1.40 réalise la fonction  $Z$  à l'aide des opérateurs NON, ET et OU; on cherche un logigramme équivalent ne comportant que des portes NAND.

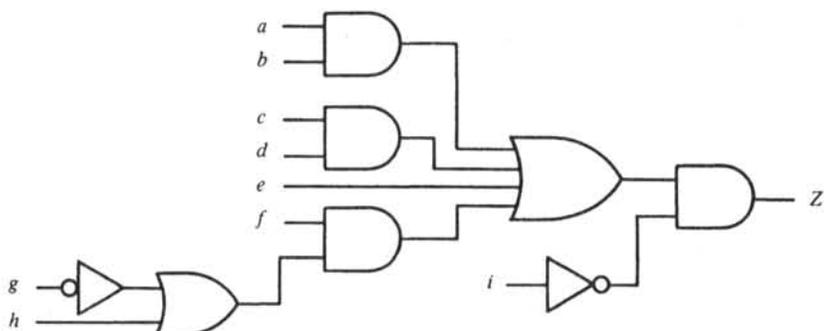


Fig. 1.40

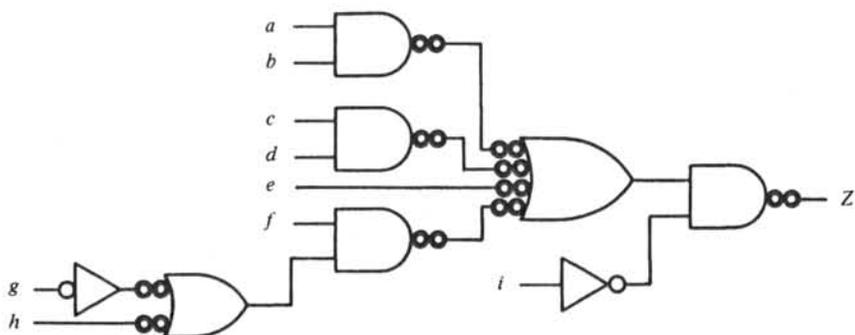


Fig. 1.41

#### ■ 1.4.17 Méthode

Le théorème (1.2) permet d'introduire sur n'importe quel fil deux inverseurs en série; on vérifie en effet la relation  $\bar{\bar{a}} = a$ . Il en découle la règle suivante : **sur chaque fil sortant d'une porte ET ou entrant dans une porte OU on introduit deux inverseurs en série**. Le nouveau logigramme (fig. 1.41) est fonctionnellement équivalent au précé-

dent (fig. 1.40); pour mettre en évidence la transformation opérée, on a symbolisé chaque inverseur rajouté par un cercle en trait fort.

En examinant les logigrammes équivalents de la porte NAND (fig. 1.36) il est possible de faire apparaître dans la figure 1.41 des portes NAND (à deux et quatre entrées) ainsi qu'un certain nombre d'inverseurs; ces inverseurs sont eux-mêmes réalisés à l'aide d'une porte NAND à deux entrées selon le schéma de la figure 1.30 : on obtient enfin le logigramme NAND recherché (fig. 1.42). On peut en déduire la forme algébrique de  $Z$  exprimée exclusivement à l'aide de la fonction NAND ( $\uparrow$ ) :

$$Z = \left| \left| (a \uparrow b) \uparrow (c \uparrow d) \uparrow (e \uparrow 1) \uparrow (f \uparrow [g \uparrow \{h \uparrow 1\}]) \right| \uparrow [i \uparrow 1] \right| \uparrow 1 \quad (1.38)$$

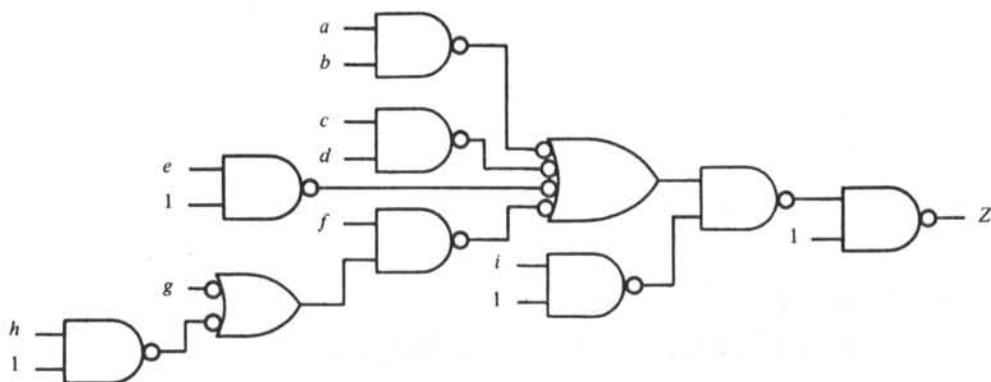


Fig. 1.42

**1.4.18 Exercice**

A l'aide de la méthode du paragraphe 1.4.17 (ou d'une variante de cette méthode) on demande de tracer :

- le logigramme réalisant les fonctions  $Z_1 = \overline{abcd}$  et  $Z_2 = \overline{abc} + \overline{cd}f + ae$  à l'aide de portes NAND à deux entrées seulement;
- le logigramme réalisant la fonction  $Z_3 = \overline{a + b + c + d}$  à l'aide de portes NOR à trois entrées;
- le logigramme réalisant l'expression (1.37) à l'aide de portes NOR exclusivement; puis le logigramme de la même fonction à l'aide de portes NAND à deux entrées et de portes NOR à trois entrées utilisées conjointement.

■ **1.4.19 Exercice**

On cherche à exprimer algébriquement la fonction réalisée par le logigramme de la figure 1.43; une forme immédiate est :

$$Z = \left\{ \left[ (\overline{S} \downarrow T) B \right] \downarrow \left[ \left\{ (\overline{K} \uparrow M) \uparrow B \right\} D \right] \downarrow A \right\} + RP \quad (1.39)$$

Si l'on désire exprimer la fonction  $Z$  sous la forme commode d'un polynôme, on peut procéder de deux façons :

- soit algébriquement, grâce aux définitions des fonctions NAND et NOR et par l'utilisation répétée des théorèmes de De Morgan;

- soit graphiquement, en transformant le logigramme donné (fig. 1.43) pour obtenir un logigramme NON, ET, OU équivalent et n'ayant des opérateurs NON qu'aux entrées; pour cette transformation, on utilisera les équivalences de logigrammes décrites dans les figures 1.36 à 1.39.

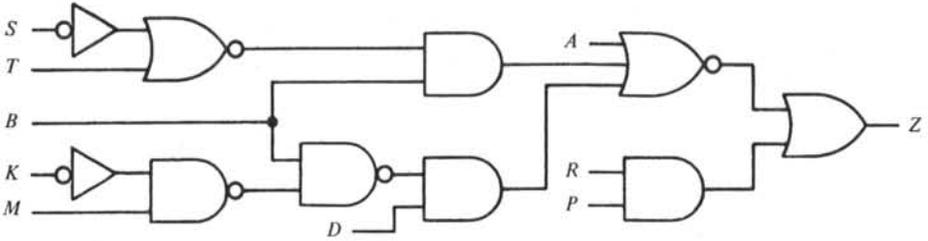


Fig. 1.43

### 1.4.20 Exercice

A l'aide de la transformation de logigrammes suggérée dans le paragraphe 1.4.19, on demande d'exprimer sous forme de polynômes les fonctions  $Z_1$  et  $Z_2$  réalisées par les expressions suivantes :

$$Z_1 = \{ [\bar{b} \downarrow c] \downarrow [(e \downarrow f) \downarrow \bar{d}] \downarrow a \} \downarrow 0 \quad ;$$

$$Z_2 = \{ \{ [(P \uparrow Q) \uparrow K] \uparrow L \} \uparrow \{ \bar{M} \uparrow K \} \} \uparrow D \uparrow \{ B \uparrow B \} \uparrow A$$

Tenter ensuite d'effectuer la transformation directe du logigramme NOR de  $Z_1$  en un logigramme NAND équivalent puis, inversement, convertir directement le logigramme NAND de  $Z_2$  en un logigramme NOR équivalent; en déduire une méthode graphique de transformation NOR-NAND et NAND-NOR.

## 1.5 SYSTÈMES COMBINATOIRES UNIVERSELS

### 1.5.1 Définition : minterme

On appelle *minterme* ou *fonction unité* de deux variables chacun des quatre monômes suivants de ces variables :

$$Z_8 = \bar{a}\bar{b} \ ; \ Z_4 = \bar{a}b \ ; \ Z_2 = a\bar{b} \ ; \ Z_1 = ab \quad (1.40)$$

$a$	$b$	$\bar{a}$	$\bar{b}$	$Z_8 = \bar{a}\bar{b}$	$Z_4 = \bar{a}b$	$Z_2 = a\bar{b}$	$Z_1 = ab$	$Z_9$	$Z$
0	0	1	1	1	0	0	0	1	$K_0$
0	1	1	0	0	1	0	0	0	$K_1$
1	0	0	1	0	0	1	0	0	$K_2$
1	1	0	0	0	0	0	1	1	$K_3$

Fig. 1.44

La table de vérité (fig. 1.44) et les diagrammes de Venn (fig. 1.45) montrent que chaque minterme vaut 1 pour un état d'entrée  $a, b$  et un seul.

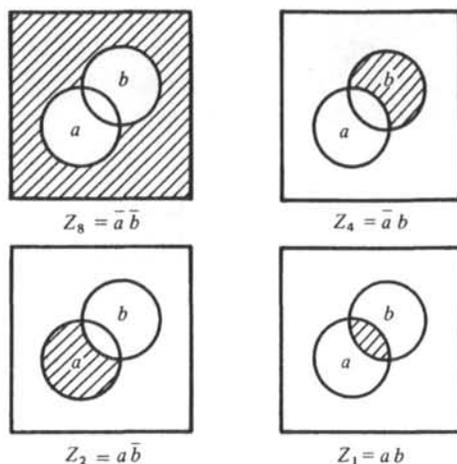


Fig. 1.45

### 1.5.2 Commentaire

On observe que l'expression algébrique de chaque minterme est un produit de toutes les variables ( $a$  et  $b$ ), chaque variable étant sous la forme vraie ou complémentaire selon que sa valeur logique dans l'état correspondant ( $a, b$ ) est 1 ou 0 : dans la figure 1.44 on constate par exemple que  $Z_4 = 1$  pour  $a, b = 01$  ; on déduit que  $Z_4 = \bar{a} b$ .

La généralisation du concept de minterme est immédiate : pour  $n$  variables, il existe  $2^n$  états d'entrée et  $2^n$  mintermes exprimés chacun à l'aide d'un monôme des  $n$  variables judicieusement complémentées.

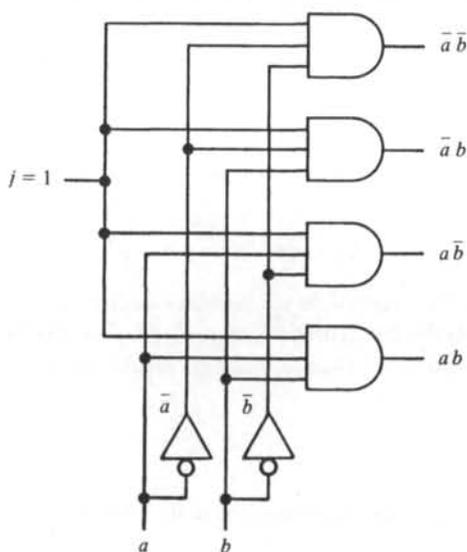


Fig. 1.46

### 1.5.3 Définition : démultiplexeur

On appelle *démultiplexeur* tout système combinatoire réalisant les  $2^n$  mintermes de  $n$  variables.

Le logigramme de la figure 1.46 est une réalisation possible d'un démultiplexeur à deux variables déduite de la relation (1.40); la figure 1.47 donne une représentation simplifiée dont le symbolisme suggère que la constante logique  $j = 1$  est aiguillée sur l'une des quatre sorties selon l'état d'entrée  $(a, b)$ .

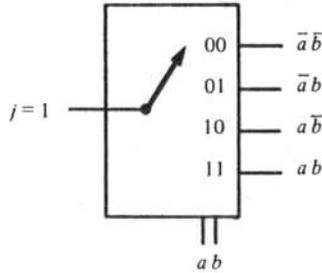


Fig. 1.47

### 1.5.4 Propriété

On constate que toute fonction logique peut être exprimée par la somme d'un ou plusieurs mintermes. On vérifie dans une table de vérité (fig. 1.44) que la fonction  $Z_9$ , par exemple, peut s'écrire sous la forme :

$$Z_9 = Z_8 + Z_1 = \bar{a}\bar{b} + ab \quad (1.41)$$

### 1.5.5 Définition : forme canonique

Toute somme de mintermes est appelée *forme canonique algébrique*; cette forme est unique pour une fonction donnée. L'expression (1.41) représente donc la forme canonique de la fonction  $Z_9$ .

### 1.5.6 Définitions : fonction logique universelle

La *fonction logique universelle* de deux variables  $a$  et  $b$  est définie par l'expression algébrique :

$$Z(a, b; K_0, K_1, K_2, K_3) = K_0 \cdot \bar{a}\bar{b} + K_1 \cdot \bar{a}b + K_2 \cdot a\bar{b} + K_3 \cdot ab \quad (1.42)$$

La fonction universelle dépend de six variables au total, soit les deux variables principales  $a, b$  et quatre variables auxiliaires  $K_0, K_1, K_2, K_3$  appelées *paramètres*. L'expression (1.42) est un polynôme dont chaque monôme résulte du produit d'un paramètre et d'un minterme.

### 1.5.7 Propriété

En calculant la valeur de la fonction universelle (1.42) pour chacun des quatre états  $a, b$  on trouve :

$$Z(a, b) = Z(0, 0) = K_0 \quad (1.43)$$

$$Z(a,b) = Z(0,1) = K_1 \tag{1.44}$$

$$Z(a,b) = Z(1,0) = K_2 \tag{1.45}$$

$$Z(a,b) = Z(1,1) = K_3 \tag{1.46}$$

On constate que dans chaque état  $a, b$  la fonction universelle prend la valeur d'un des quatre paramètres  $K_0, K_1, K_2, K_3$  : ces valeurs sont reportées dans la table de vérité de la figure 1.44. En remplaçant dans (1.42) les paramètres par leurs expressions (1.43) à (1.46) on obtient enfin :

$$Z(a,b) = Z(0,0) \cdot \bar{a}\bar{b} + Z(0,1) \cdot \bar{a}b + Z(1,0) \cdot a\bar{b} + Z(1,1) \cdot ab \tag{1.47}$$

### 1.5.8 Applications

En affectant à chacun des quatre paramètres  $K_0, K_1, K_2, K_3$  la valeur 0 ou 1, il est possible de retrouver la forme canonique de n'importe quelle fonction de deux variables.

La fonction  $Z_9$ , par exemple, est définie par une table de vérité (fig. 1.44) et par une forme canonique (1.41). En choisissant les valeurs suivantes des paramètres :

$$K_0 = 1 ; K_1 = 0 ; K_2 = 0 ; K_3 = 1 \tag{1.48}$$

et en introduisant ces valeurs dans l'expression (1.42) de la fonction universelle, on retrouve bien la forme canonique (1.41) de  $Z_9$  :

$$Z(a,b; K_0, K_1, K_2, K_3) = Z(a,b; 1,0,0,1) = \bar{a}\bar{b} + ab = Z_9 \tag{1.49}$$

Les seize états  $(K_0, K_1, K_2, K_3)$  définissent les seize fonctions de deux variables (fig. 1.14); chacune de ces fonctions est une *solution* de la fonction universelle.

Par extension, la fonction universelle de  $n$  variables s'exprime algébriquement sous la forme d'un polynôme de  $2^n$  monômes, chaque monôme étant le produit d'un paramètre et d'un minterme de  $n$  variables; une telle fonction dépend donc de  $n + 2^n$  variables au total. Les  $2^{2^n}$  états des  $2^n$  paramètres définissent les  $2^{2^n}$  fonctions de  $n$  variables qui sont les solutions de la fonction universelle.

### 1.5.9 Définition : système combinatoire universel

On appelle *système combinatoire universel* ou *multiplexeur à  $n$  variables* (en anglais : *data selector*) tout système réalisant la fonction universelle de  $n$  variables.

Le logigramme de la figure 1.48 illustre un exemple de multiplexeur à deux variables  $(a, b)$  réalisant la relation (1.42); la figure 1.49 donne une représentation simplifiée dont le symbolisme suggère que la sortie  $Z$  sélectionne l'une des quatre entrées  $K_0, K_1, K_2, K_3$  selon l'état  $a, b$ .

### 1.5.10 Exemple

On cherche à réaliser la fonction  $Z_9$  (fig. 1.44) à l'aide d'un multiplexeur à deux variables. Le choix des valeurs des paramètres a déjà été effectué :  $K_0 = 1; K_1 = 0; K_2 = 0; K_3 = 1$ . En introduisant ces valeurs dans le multiplexeur de la figure 1.50, on réalise la fonction recherchée (1.41). L'examen comparé des figures 1.50 et 1.44 montre

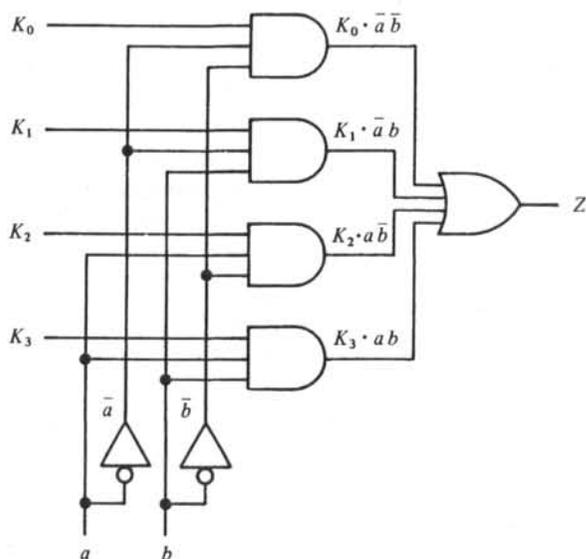


Fig. 1.48

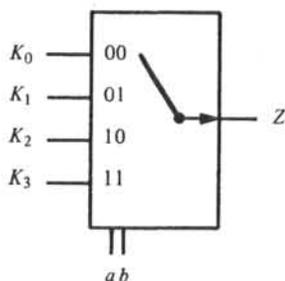


Fig. 1.49

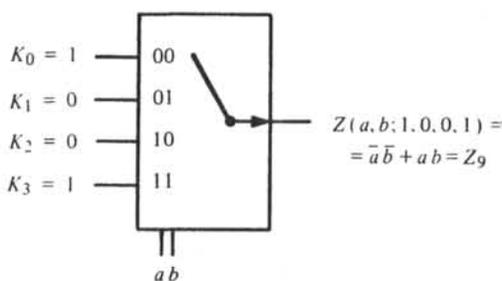


Fig. 1.50

que le codage des quatre paramètres  $K_0, K_1, K_2, K_3$  reproduit la table de vérité de  $Z_9$ ; la réalisation de n'importe quelle fonction de deux variables est donc immédiate sitôt donnée la table de vérité.

### 1.5.11 Exercice

Déterminer l'équation de la fonction universelle d'une variable et chercher une réalisation possible à l'aide d'un multiplexeur à deux variables. Réaliser ensuite à l'aide d'un assemblage de multiplexeurs à deux variables les fonctions suivantes :

$$Z_a = a \uparrow b \uparrow c; \quad Z_b = a \downarrow b \downarrow c; \quad Z_c = ab + \bar{a}c + bc; \quad Z_d = [ab + cd + e + f(\bar{g} + h)] \cdot \bar{i} \quad (\text{fig. 1.40}).$$

### 1.5.12 Exercice

Déterminer le logigramme d'un multiplexeur à trois variables puis réaliser les fonctions  $Z_a, Z_b, Z_c$  et  $Z_d$  du paragraphe 1.5.11 à l'aide d'un ou plusieurs de ces multiplexeurs. Concevoir ensuite un multiplexeur à trois variables à l'aide d'un nombre minimal de multiplexeurs à une variable; généraliser pour déterminer un logigramme réalisant la fonction universelle de  $n$  variables à l'aide de multiplexeurs à une variable.

### 1.5.13 Exercice

On appelle *maxterme* de deux variables  $(a, b)$  toute somme logique de ces variables apparaissant sous la forme vraie ou complémentaire  $(\bar{a} + \bar{b}, \bar{a} + b, a + \bar{b}, a + b)$ . Démontrer que toute fonction de deux variables peut s'exprimer algébriquement sous la forme d'un produit de maxtermes; en déduire l'équation d'une nouvelle fonction universelle et donner un logigramme; réaliser à l'aide de ce logigramme les fonctions  $Z_a = a \downarrow b$ ,  $Z_b = a \uparrow b$ ,  $Z_c = a \downarrow b$  et  $Z_d = \bar{a}\bar{b} + a\bar{b}$ .

## 1.6 APPLICATIONS

### 1.6.1 Définition : fonction OU-exclusif

La fonction *OU-exclusif* de deux variables (en anglais : *Exclusive OR* ou *XOR*) est définie algébriquement par la relation :

$$Z = a \oplus b = \bar{a}b + a\bar{b} \tag{1.50}$$

La table de vérité de cette fonction est représentée dans la figure 1.51 ainsi que dans la figure 1.14 (fonction  $Z_6$ ); il ressort du terme OU-exclusif que la fonction  $a \oplus b$  est égale à 1 si  $a$  OU  $b$  vaut 1 à l'exclusion du cas où  $a$  ET  $b$  valent simultanément 1. On remarque aussi que la fonction OU-exclusif prend la valeur 1 lorsque  $a = \bar{b}$ ; il en découle que la fonction inverse  $\overline{a \oplus b} = 1$  lorsque  $a = b$  (fig. 1.51).

$a$	$b$	$\bar{a}$	$\bar{b}$	$a \oplus b$	$\overline{a \oplus b}$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

Fig. 1.51

Les logigrammes des figures 1.52 à 1.54 représentent dans chacune des normes CEI, MIL et DIN deux réalisations équivalentes de l'opérateur OU-exclusif.

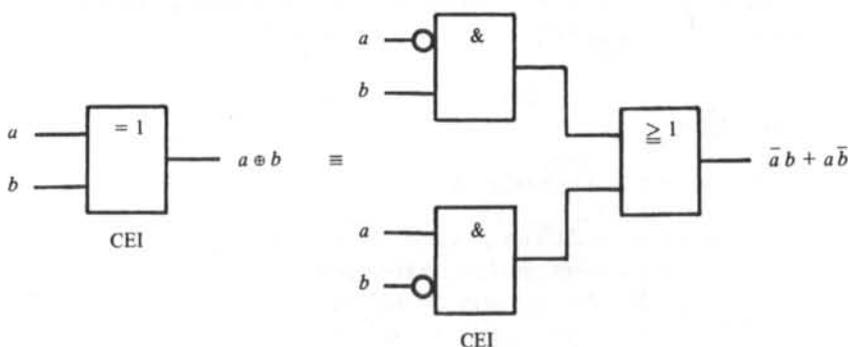


Fig. 1.52

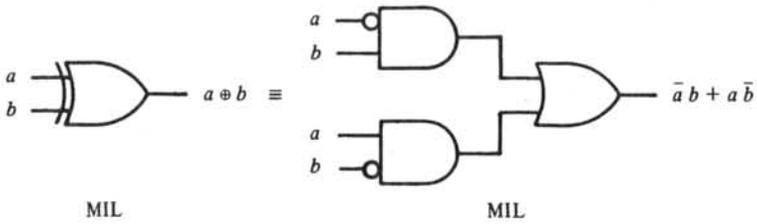


Fig. 1.53

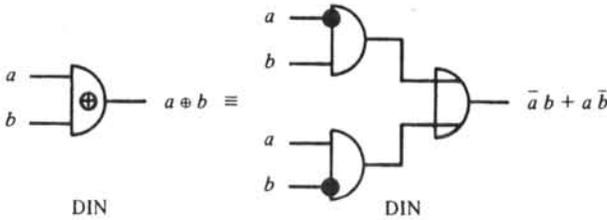


Fig. 1.54

### 1.6.2 Théorèmes

L'examen de la table de vérité (fig. 1.51) permet la démonstration tabulaire immédiate des relations suivantes :

$$a \oplus b = b \oplus a \quad (1.51)$$

$$a \oplus a = 0 \quad (1.52)$$

$$a \oplus 0 = a ; a \oplus 1 = \bar{a} \quad (1.53)$$

$$a \oplus \bar{a} = 1 \quad (1.54)$$

### 1.6.3 Théorème

$$\overline{a \oplus b} = \bar{a}\bar{b} + ab = a \oplus \bar{b} = \bar{a} \oplus b \quad (1.55)$$

La définition (1.50) permet de calculer :  $a \oplus \bar{b} = \bar{a}\bar{b} + a\bar{b} = \bar{a}\bar{b} + ab$ ;  $\bar{a} \oplus b = \bar{a}b + \bar{a}\bar{b} = \bar{a}\bar{b} + ab$ .

L'expression  $\bar{a}\bar{b} + ab$  est la forme canonique de la fonction  $\overline{a \oplus b}$  (fig. 1.51) : la relation (1.55) est démontrée.

### 1.6.4 Corollaire

$$\bar{a} \oplus \bar{b} = a \oplus b = \overline{a \oplus \bar{b}} = \overline{\bar{a} \oplus b} \quad (1.56)$$

Par la définition (1.50) on a :  $\bar{a} \oplus \bar{b} = \bar{a}\bar{b} + \bar{a}b = a\bar{b} + \bar{a}b = a \oplus b$ .

En inversant les divers membres du théorème (1.55) on obtient les égalités :  $\overline{\bar{a} \oplus \bar{b}} = a \oplus b = \overline{a \oplus \bar{b}} = \overline{\bar{a} \oplus b}$  qui démontrent la relation.

Les diverses expressions algébriques de la relation (1.56) correspondent à quatre logigrammes équivalents (fig. 1.55) : chacun d'eux réalise la fonction OU-exclusif.

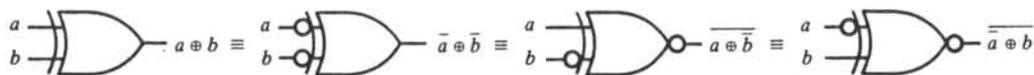


Fig. 1.55

**1.6.5 Définition**

La fonction *OU-exclusif* de  $n$  variables  $a \oplus b \oplus \dots \oplus n$  est définie par les propriétés suivantes :

- elle vaut 1 lorsqu'un nombre impair de variables prennent la valeur 1 ;
- elle vaut 0 lorsqu'un nombre pair de variables prennent la valeur 1.

On vérifie tout d'abord que la fonction OU-exclusif de deux variables jouit de la propriété (fig. 1.51) ; la table de vérité de la figure 1.56 donne ensuite l'exemple de la fonction OU-exclusif de trois variables dont la forme canonique s'écrit :

$$Z = a \oplus b \oplus c = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc \tag{1.57}$$

Pour alléger la présentation de la table de vérité (fig. 1.56) on peut négliger de reporter les 0 des divers mintermes.

$a$	$b$	$c$	$a \oplus b \oplus c$	$\bar{a}\bar{b}c$	$\bar{a}b\bar{c}$	$a\bar{b}\bar{c}$	$abc$
0	0	0	0				
0	0	1	1	1			
0	1	0	1		1		
0	1	1	0				
1	0	0	1			1	
1	0	1	0				
1	1	0	0				
1	1	1	1				1

Fig. 1.56

**1.6.6 Théorème d'associativité**

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c = a \oplus b \oplus c \tag{1.58}$$

La démonstration, tabulaire ou algébrique, est laissée au soin du lecteur.

**1.6.7 Théorème**

Si  $Z = a \oplus b$ , alors il est possible d'exprimer  $a$  ( $b$ ) en fonction des deux grandeurs restantes  $b$  et  $Z$  ( $a$  et  $Z$ ) :

$$a = b \oplus Z ; b = a \oplus Z \tag{1.59}$$

Démontrons algébriquement que si la relation  $a = b \oplus Z$  est vérifiée, alors  $Z = a \oplus b$  :

- par (1.59) :  $a \oplus b = (b \oplus Z) \oplus b$
- par (1.51) :  $a \oplus b = b \oplus (b \oplus Z)$

- par (1.58) :  $a \oplus b = b \oplus b \oplus Z$
- par (1.52) et (1.53) :  $a \oplus b = 0 \oplus Z = Z$

### 1.6.8 Définition

Si  $Z = 0$  dans la relation (1.59), cette expression devient :

$$a = b \quad (1.60)$$

et implique la relation :

$$a \oplus b = 0 \quad (1.61)$$

Une expression telle que (1.60) est généralement appelée *équation logique*; les relations (1.60) et (1.61) sont équivalentes.

### 1.6.9 Exercice

Peut-on démontrer les relations suivantes :

$$(a \oplus b) c \stackrel{?}{=} ac \oplus bc$$

$$(a \oplus b) + c \stackrel{?}{=} (a + c) \oplus (b + c)$$

$$(ab) \oplus c \stackrel{?}{=} (a \oplus c)(b \oplus c)$$

$$(a + b) \oplus c \stackrel{?}{=} (a \oplus c) + (b \oplus c)$$

$$a \oplus b \stackrel{?}{=} (a + b) \cdot \overline{ab}$$

### ■ 1.6.10 Exercice

Démontrer que l'opérateur OU-exclusif à deux variables n'est pas complet au sens du paragraphe 1.4.10.

### ■ 1.6.11 Exercice

Dans la pratique on associe fréquemment des opérateurs OU-exclusif à des portes NAND ou NOR; en utilisant les divers logigrammes équivalents de la figure 1.55 et en s'inspirant de la méthode graphique du paragraphe 1.4.17, on demande de réaliser les fonctions suivantes à l'aide de portes NAND et OU-exclusif ne comportant que deux entrées chacune :

$$Z_1 = bd \oplus a \oplus c; \quad Z_2 = (a\overline{b} + \overline{a}b + c) \oplus d \oplus ef; \quad Z_3 = \overline{a}c + ab\overline{c} + \overline{b}c;$$

$$Z_4 = a + cd + bd + \overline{b}\overline{d}; \quad Z_5 = (\overline{b}\overline{d} \cdot a + \overline{a}bd) \oplus c.$$

### ■ 1.6.12 Exercice

En s'inspirant de la méthode graphique du paragraphe 1.4.19 on cherchera à démontrer par des transformations de logigrammes, puis par des calculs algébriques simples, l'équivalence des relations suivantes :

$$[(a \uparrow b) \uparrow a] \uparrow [(a \uparrow b) \uparrow b] = a \oplus b$$

$$[(a \uparrow b) \cdot (\overline{a} \downarrow c)] \downarrow (b \oplus \overline{c}) = b \oplus c$$

$$\overline{(a \oplus b)(b \oplus c)} \uparrow [(\overline{a} \downarrow \overline{b}) + (a \downarrow c)] = \overline{b}c + (a \oplus b)$$

■ 1.6.13 Exercice

Montrer que la fonction universelle de deux variables (1.42)  $Z = K_0 \cdot \bar{a}\bar{b} + K_1 \cdot \bar{a}b + K_2 \cdot a\bar{b} + K_3 \cdot ab$  peut se mettre sous la forme :

$$Z = y_0 \oplus y_1 \cdot a \oplus y_2 \cdot b \oplus y_3 \cdot ab \tag{1.62}$$

où  $y_0, y_1, y_2, y_3$  sont quatre paramètres qui ne dépendent que des paramètres  $K_0, K_1, K_2, K_3$ . Réaliser l'expression (1.62) à l'aide d'un logigramme NAND, OU-exclusif.

1.6.14 Définition : fonction MAJORITÉ

La fonction MAJORITÉ de trois variables  $MAJ(a, b, c)$  prend la valeur 1 lorsque deux variables au moins valent 1 (fig. 1.57). Le logigramme de la figure 1.58 représente une porte MAJORITÉ selon les normes CEI (il n'existe pas de logigramme équivalent dans les normes MIL et DIN).

Par extension, la fonction MAJORITÉ de  $n$  variables (avec  $n$  impair) vaut 1 lorsque  $(n + 1)/2$  variables au moins sont égales à 1.

No	a	b	c	MAJ (a, b, c)	$\bar{a}\bar{b}c$	$a\bar{b}\bar{c}$	$ab\bar{c}$	$abc$
0	0	0	0	0				
1	0	0	1	0				
2	0	1	0	0				
3	0	1	1	1	1			
4	1	0	0	0				
5	1	0	1	1		1		
6	1	1	0	1			1	
7	1	1	1	1				1

Fig. 1.57

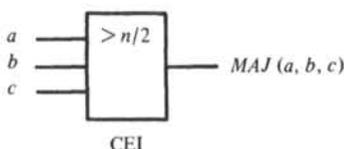


Fig. 1.58

1.6.15 Modes de représentation

Trois cercles (correspondant aux trois variables  $a, b, c$ ) qui se coupent selon la figure 1.59 déterminent huit régions dans un diagramme de Venn : chacune d'elles représente un état d'entrée  $(a, b, c)$ , c'est-à-dire une ligne de la table de vérité. En hachurant les régions dans lesquelles  $MAJ(a, b, c) = 1$ , on obtient une représentation de cette fonction (fig. 1.60).

Il est également possible de montrer les huit états  $a, b, c$  dans une table rectangulaire. La configuration proposée par Karnaugh (fig. 1.61) fait correspondre à chaque

cercle du diagramme de Venn (fig. 1.59) un carré (variables  $a$  et  $b$ ) ou un rectangle (variable  $c$ ). En inscrivant un 1 dans chaque case pour laquelle  $MAJ(a, b, c) = 1$  on obtient le mode de représentation de la figure 1.62.

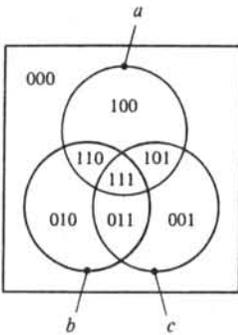


Fig. 1.59

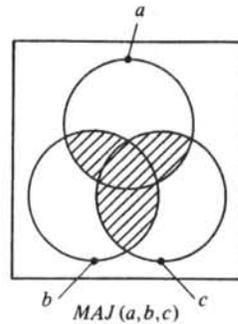


Fig. 1.60

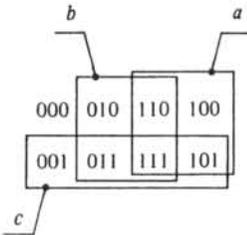


Fig. 1.61

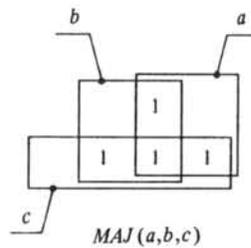


Fig. 1.62

La forme canonique découle de la table de vérité (fig. 1.57) :

$$MAJ(a, b, c) = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc \quad (1.63)$$

### 1.6.16 Définitions

Il est souhaitable de disposer d'un symbolisme plus succinct que la forme canonique (1.63) pour décrire la fonction  $MAJ(a, b, c)$ ; il est possible d'affecter à chaque état d'entrée  $a, b, c$  un *numéro décimal* (No dans la fig. 1.57) : ce numéro est l'équivalent décimal du nombre binaire  $a, b, c$  (sect. 8.1). On appelle *forme canonique décimale* d'une fonction la liste des numéros pour lesquels cette fonction vaut 1.

Dans le cas de la fonction MAJORITÉ on obtient (fig. 1.57) :

$$MAJ(a, b, c) = \Sigma 3, 5, 6, 7 \quad (1.64)$$

Lors du recours à la forme canonique décimale, il est impératif de préciser l'ordre et le nombre des variables pour pouvoir reconstituer la table de vérité ou la forme canonique algébrique originales.

### 1.6.17 Logigrammes

Le logigramme NON, ET, OU de la figure 1.63 réalise la forme canonique (1.63) de la fonction MAJORITÉ; la méthode graphique du paragraphe 1.4.17 permet de déterminer un logigramme NAND équivalent (fig. 1.64).

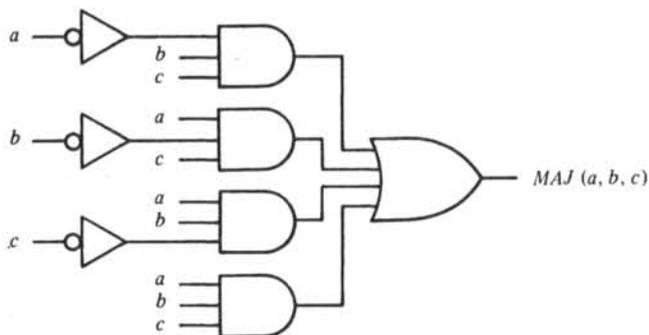


Fig. 1.63

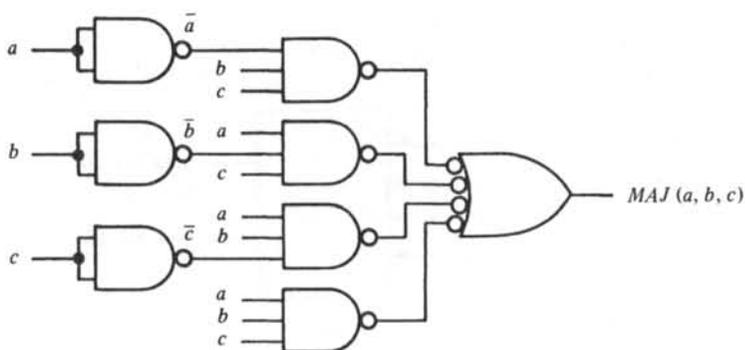


Fig. 1.64

### 1.6.18 Exercice

Vérifier à l'aide d'une table de vérité et d'un diagramme de Venn l'égalité des expressions suivantes :

$$\begin{aligned}
 MAJ(a,b,c) &= \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = ab + ac + bc \\
 &= (a + b)(b + c)(c + a)
 \end{aligned}
 \tag{1.65}$$

En déduire une vérification purement algébrique; comparer les logigrammes NAND réalisant ces diverses expressions.

### 1.6.19 Exercice

Trouver les formes canoniques algébrique et décimale de  $\overline{MAJ(a,b,c)}$  à l'aide d'une table de vérité puis, grâce aux théorèmes de De Morgan, à partir de l'expression (1.63) de  $MAJ(a,b,c)$ .

## 1.6.20 Exercice

On donne un système logique réalisant la fonction suivante :

$$Z = MAJ(a,b,c) \oplus \bar{d} \quad (1.66)$$

Etablir la table de vérité et les formes canoniques (algébrique et décimale) de  $Z$ ; pour quelles valeurs particulières des variables  $c$  et  $d$  est-il possible de réaliser les fonctions suivantes :  $Z_1 = a \cdot b$ ;  $Z_2 = a + b$ ;  $Z_3 = a \uparrow b$ ;  $Z_4 = a \downarrow b$ ;  $Z_5 = \overline{MAJ(a,b,c)}$ ?

## 1.7 FONCTIONS INCOMPLÈTEMENT DÉFINIES

## 1.7.1 Exemple

Il arrive fréquemment qu'on ait à réaliser un système logique selon un cahier des charges incomplet. Le schéma de la figure 1.65 illustre un système combinatoire dont :

- la fonction  $E(a,b,c,d)$  vaut 1 si et seulement si le nombre de variables valant 0 est égal au nombre de variables valant 1 dans l'état d'entrée  $(a,b,c,d)$ ;
- la fonction  $M(a,b,c,d)$  vaut 0 s'il y a une majorité de 0 et vaut 1 s'il y a une majorité de 1 dans l'état d'entrée  $(a,b,c,d)$ .

Ce cahier des charges détermine une fonction  $E$  unique dont la table de vérité est celle de la figure 1.66.

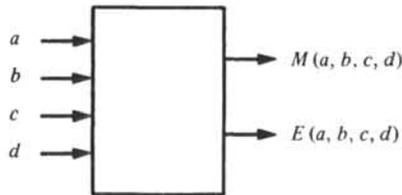


Fig. 1.65

No	a	b	c	d	$E(a, b, c, d)$	$M(a, b, c, d)$	$MAJ(a, b, c)$	$\lambda$
0	0	0	0	0	0	0	0	$K_0$
1	0	0	0	1	0	0	0	$K_1$
2	0	0	1	0	0	0	0	$K_2$
3	0	0	1	1	1	$\emptyset = K_3$	0	$K_3$
4	0	1	0	0	0	0	0	$K_4$
5	0	1	0	1	1	$\emptyset = K_5$	0	$K_5$
6	0	1	1	0	1	$\emptyset = K_6$	1	$K_6$
7	0	1	1	1	0	1	1	$K_7$
8	1	0	0	0	0	0	0	$K_8$
9	1	0	0	1	1	$\emptyset = K_9$	0	$K_9$
10	1	0	1	0	1	$\emptyset = K_{10}$	1	$K_{10}$
11	1	0	1	1	0	1	1	$K_{11}$
12	1	1	0	0	1	$\emptyset = K_{12}$	1	$K_{12}$
13	1	1	0	1	0	1	1	$K_{13}$
14	1	1	1	0	0	1	1	$K_{14}$
15	1	1	1	1	0	1	1	$K_{15}$

Fig. 1.66

**1.7.2 Définitions : fonction incomplètement définie**

La fonction  $M$  de l'exemple précédent n'est pas définie pour tout état d'entrée puisqu'on ne précise pas quelle valeur elle doit prendre pour les états  $a, b, c, d$  dans lesquels il y a égalité de 0 et de 1 : on dit alors que  $M$  est une *fonction incomplètement définie*. On introduit un symbole spécial  $\emptyset$  et l'on pose conventionnellement :

$$M(a,b,c,d) = \emptyset \tag{1.67}$$

pour tout état  $(a,b,c,d)$  dans lequel  $M$  n'est pas définie (fig. 1.66). Les états  $a, b, c, d$  tels que  $M(a,b,c,d) = \emptyset$  sont appelés *conditions*  $\emptyset$  ou *états*  $\emptyset$  de  $M$  (on dit aussi *conditions indifférentes* et en anglais : *don't care conditions* ou *don't happen conditions*).

Nous dirons qu'une fonction complètement définie  $Z(a,b,c,d)$  qui est égale à  $M(a,b,c,d)$  pour tout état  $a,b,c,d$  tel que  $M(a,b,c,d) \neq \emptyset$  est une *solution* de  $M$ ; on écrira dans ce cas :

$$Z(a,b,c,d) \in M(a,b,c,d) \tag{1.68}$$

Si  $M$  possède  $n$  états  $\emptyset$ , elle possède  $2^n$  solutions qui constituent une *famille*.

**1.7.3 Exemple**

La fonction  $MAJ(a,b,c)$  définie au paragraphe 1.6.14 et donnée par la relation (1.63) est représentée dans la table de vérité de la figure 1.66. On vérifie qu'elle est une solution de  $M(a,b,c,d)$ ; on peut donc écrire :

$$MAJ(a,b,c) \in M(a,b,c,d) \tag{1.69}$$

**1.7.4 Rappel**

La fonction universelle de quatre variables  $a, b, c, d$  se déduit de l'expression (1.42) et des considérations du paragraphe 1.5.8 :

$$\begin{aligned} \lambda &= Z(a,b,c,d; K_0, K_1, \dots, K_{15}) = \\ &= K_0 \cdot \bar{a}\bar{b}\bar{c}\bar{d} + K_1 \cdot \bar{a}\bar{b}\bar{c}d + K_2 \cdot \bar{a}\bar{b}c\bar{d} + K_3 \cdot \bar{a}\bar{b}cd + \\ &K_4 \cdot \bar{a}b\bar{c}\bar{d} + K_5 \cdot \bar{a}b\bar{c}d + K_6 \cdot \bar{a}bc\bar{d} + K_7 \cdot \bar{a}bcd + \\ &K_8 \cdot a\bar{b}\bar{c}\bar{d} + K_9 \cdot a\bar{b}\bar{c}d + K_{10} \cdot a\bar{b}c\bar{d} + K_{11} \cdot a\bar{b}cd + \\ &K_{12} \cdot ab\bar{c}\bar{d} + K_{13} \cdot ab\bar{c}d + K_{14} \cdot abc\bar{d} + K_{15} \cdot abcd \end{aligned} \tag{1.70}$$

Il existe dans ce cas seize variables auxiliaires  $K_0, K_1, \dots, K_{15}$  qui sont les paramètres de  $\lambda$  (fig. 1.66).

**1.7.5 Modes de représentation**

En énumérant les numéros des états de la table de vérité (fig. 1.66) pour lesquels la fonction incomplètement définie vaut respectivement 1 et  $\emptyset$ , on obtient la forme canonique décimale de  $M$  :

$$M(a,b,c,d) = \Sigma 7,11,13,14,15 + \emptyset 3,5,6,9,10,12 \tag{1.71}$$

Dans chaque état  $\emptyset$ , la fonction  $M$  peut prendre une valeur logique quelconque, c'est-à-dire 0 ou 1; cette alternative peut s'exprimer à l'aide d'une variable logique  $K$  ou paramètre. Six paramètres ( $K_3, K_5, K_6, K_9, K_{10}$  et  $K_{12}$ ) sont nécessaires pour décrire

$M$  (fig. 1.66). On peut alors considérer  $M$  comme la fonction universelle  $\lambda$  (1.70) dans laquelle les seize paramètres  $K_0, K_1, \dots, K_{15}$  ont :

- pris la valeur 0 pour tout état  $a, b, c, d$  tel que  $M(a, b, c, d) = 0$ ;
- pris la valeur 1 pour tout état tel que  $M(a, b, c, d) = 1$ ;
- conservé la forme d'une variable  $K$  pour tout état tel que  $M(a, b, c, d) = \emptyset$ .

On en déduit la forme canonique algébrique de  $M$  :

$$M(a, b, c, d) = \bar{a}b\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + a\bar{b}c\bar{d} + ab\bar{c}d + a\bar{b}c\bar{d} + K_3 \cdot \bar{a}\bar{b}c\bar{d} + K_5 \cdot \bar{a}b\bar{c}d + K_6 \cdot \bar{a}b\bar{c}\bar{d} + K_9 \cdot \bar{a}\bar{b}c\bar{d} + K_{10} \cdot \bar{a}\bar{b}c\bar{d} + K_{12} \cdot \bar{a}b\bar{c}\bar{d} \quad (1.72)$$

Chacun des  $2^6$  états ( $K_3, K_5, K_6, K_9, K_{10}, K_{12}$ ) détermine une forme particulière de (1.72) qui est une solution de  $M$ .

### 1.7.6 Application

La forme canonique décimale de la fonction universelle de quatre variables (1.70) s'écrit :

$$\lambda = Z(a, b, c, d) = \emptyset 0, 1, \dots, 15 \quad (1.73)$$

On peut donc considérer la fonction universelle de  $n$  variables comme une fonction incomplètement définie comportant  $2^n$  états  $\emptyset$ .

### 1.7.7 Définitions

On appelle *fonction maximale* ou *borne supérieure*  $Z_M$  d'une fonction incomplètement définie  $Z$  la solution obtenue en donnant à  $Z$  la valeur 1 pour tous les états  $\emptyset$ ; la *fonction minimale* ou *borne inférieure*  $Z_m$  de  $Z$  est la solution calculée en donnant à  $Z$  la valeur 0 pour tous les états  $\emptyset$ .

La *fonction indifférente*  $Z_\emptyset$  de  $Z$  est une fonction incomplètement définie valant  $\emptyset$  pour tous les états  $\emptyset$  de  $Z$  et valant 0 pour l'ensemble des états restants. Enfin, la *fonction différence*  $Z_\Delta$  de  $Z$  est la borne supérieure de sa fonction indifférente  $Z_\emptyset$ .

### 1.7.8 Exemple

En considérant la fonction  $M(a, b, c, d)$  décrite par les relations (1.71) et (1.72) il est possible de déterminer les formes canoniques décimales et algébriques des bornes supérieure  $M_M$  et inférieure  $M_m$ , ainsi que des fonctions indifférente  $M_\emptyset$  et différence  $M_\Delta$ ; l'ensemble de ces fonctions est représenté dans la table de vérité de la figure 1.67 :

$$M_M(a, b, c, d) = \Sigma 3, 5, 6, 7, 9, 10, \dots, 15 \\ = \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c}\bar{d} + a\bar{b}c\bar{d} + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} + abcd \quad (1.74)$$

$$M_m(a, b, c, d) = \Sigma 7, 11, 13, 14, 15 \\ = \bar{a}b\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + abc\bar{d} + abcd \quad (1.75)$$

$$M_\emptyset(a, b, c, d) = \emptyset 3, 5, 6, 9, 10, 12 \\ = K_3 \cdot \bar{a}\bar{b}c\bar{d} + K_5 \cdot \bar{a}b\bar{c}\bar{d} + K_6 \cdot \bar{a}b\bar{c}d + K_9 \cdot \bar{a}\bar{b}c\bar{d} + K_{10} \cdot \bar{a}\bar{b}c\bar{d} + K_{12} \cdot \bar{a}b\bar{c}\bar{d} \quad (1.76)$$

$$M_{\Delta}(a,b,c,d) = \sum_{3,5,6,9,10,12} = \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} \quad (1.77)$$

No	$\lambda$	a b c d	$M(a,b,c,d)$	$M_m$	$M_g$	$M_{\Delta}$	$M_M$
0	$K_0$	0 0 0 0	0				
1	$K_1$	0 0 0 1	0				
2	$K_2$	0 0 1 0	0				
3	$K_3$	0 0 1 1	$\emptyset$		$\emptyset$	1	1
4	$K_4$	0 1 0 0	0				
5	$K_5$	0 1 0 1	$\emptyset$		$\emptyset$	1	1
6	$K_6$	0 1 1 0	$\emptyset$		$\emptyset$	1	1
7	$K_7$	0 1 1 1	1	1			1
8	$K_8$	1 0 0 0	0				
9	$K_9$	1 0 0 1	$\emptyset$		$\emptyset$	1	1
10	$K_{10}$	1 0 1 0	$\emptyset$		$\emptyset$	1	1
11	$K_{11}$	1 0 1 1	1	1			1
12	$K_{12}$	1 1 0 0	$\emptyset$		$\emptyset$	1	1
13	$K_{13}$	1 1 0 1	1	1			1
14	$K_{14}$	1 1 1 0	1	1			1
15	$K_{15}$	1 1 1 1	1	1			1

Fig. 1.67

□ 1.7.9 Première propriété

Par définition de la fonction universelle  $\lambda$  (1.70) (1.73), des fonctions indifférente  $M_{\emptyset}$  (1.76) et différence  $M_{\Delta}$  (1.77) on peut vérifier tabulairement (fig. 1.67) ou algébriquement la relation :

$$M_{\emptyset} = \lambda \cdot M_{\Delta} \quad (1.78)$$

Tout produit d'une fonction complètement définie de  $n$  variables ( $M_{\Delta}$ ) par la fonction universelle de  $n$  variables ( $\lambda$ ) est une fonction indifférente ( $M_{\emptyset}$ ) de  $n$  variables.

□ 1.7.10 Seconde propriété

Grâce aux définitions du paragraphe 1.7.7 on peut écrire successivement :

$$M = M_m + M_{\emptyset} \quad (1.79)$$

$$M_M = M_m + M_{\Delta} \quad (1.80)$$

En introduisant (1.78) dans (1.79) :

$$M = M_m + \lambda \cdot M_{\Delta} = M_m (1 + \lambda) + \lambda \cdot M_{\Delta} = M_m + \lambda (M_m + M_{\Delta}) \quad (1.81)$$

En remplaçant (1.80) dans (1.81) on obtient enfin :

$$M = M_m + \lambda \cdot M_M \quad (1.82)$$

Toute fonction incomplètement définie de  $n$  variables ( $M$ ) est la somme de sa borne inférieure ( $M_m$ ) et de sa borne supérieure multipliée par la fonction universelle de  $n$  variables ( $\lambda \cdot M_M$ ).

□ 1.7.11 Définition

La représentation d'une fonction incomplètement définie  $M$  par une expression telle que (1.82) est appelée *forme paramétrique* de  $M$ .

□ 1.7.12 Exemple

A partir des expressions de  $M_m$  (1.75) et de  $M_M$  (1.74), il est possible de donner la forme paramétrique décimale de  $M$  :

$$M(a,b,c,d) = \Sigma 7,11,13,14,15 + \lambda \cdot (\Sigma 3,5,6,7,9,10,\dots,15) \quad (1.83)$$

La forme canonique algébrique de  $M$  (1.72) comporte six paramètres ( $K_3, K_5, K_6, K_9, K_{10}, K_{12}$ ) qui sont des variables indépendantes et dont les états définissent les  $2^6$  solutions de  $M$ . Par contre la forme paramétrique de  $M$  (1.83) fait apparaître un seul paramètre  $\lambda$  qui est la fonction universelle de quatre variables (1.70) et qui peut prendre  $2^{16}$  valeurs distinctes définies par les  $2^{16}$  états ( $K_0, K_1, \dots, K_{15}$ ). On constate donc qu'à chacune des  $2^6$  solutions de  $M$  correspondent  $2^{10}$  valeurs de  $\lambda$ .

□ 1.7.13 Définition

Pour exprimer qu'une fonction incomplètement définie  $M$  est comprise entre sa borne inférieure  $M_m$  et sa borne supérieure  $M_M$  on écrit :

$$M_m \leq M \leq M_M \quad (1.84)$$

Cette expression peut se décomposer en deux relations :

$$M_m \leq M \quad (1.85)$$

$$M \leq M_M \quad (1.86)$$

Chacune des expressions (1.85) et (1.86) est une *inéquation logique*.

□ 1.7.14 Propriétés

L'examen des états  $M_m, M$  dans une table de vérité (fig. 1.68) montre que si l'inéquation (1.85) est vérifiée, alors l'équation suivante est également vérifiée :

$$M_m \cdot \bar{M} = 0 \quad (1.87)$$

$M_m$	$M$	$M_m \leq M$
( $M$	$M_M$ )	( $M \leq M_M$ )
( $M_m$	$M_M$ )	( $M_m \leq M_M$ )
0	0	Possible
0	1	Possible
1	0	Pas possible
1	1	Possible

Fig. 1.68

Pour la même raison, l'inéquation (1.86) entraîne la relation :

$$M \cdot \bar{M}_M = 0 \quad (1.88)$$

Si les deux inéquations (1.85) et (1.86) sont vérifiées, alors les deux relations (1.87) et (1.88) sont également vérifiées et l'on peut écrire :

$$M_m \cdot \bar{M} + M \cdot \bar{M}_M = 0 \tag{1.89}$$

Cette équation est équivalente à la double inéquation (1.84); elle n'admet des solutions pour  $M$  que si  $M_m \leq M_M$ , c'est-à-dire si (fig. 1.68) :

$$M_m \cdot \bar{M}_M = 0 \tag{1.90}$$

La famille des solutions de  $M$  est une fonction incomplètement définie décrite par la forme paramétrique (1.82) :

$$M = M_m + \lambda \cdot M_M$$

Dans le cas particulier où  $M$  est complètement définie, on a  $M_m = M_M$ . Les relations (1.82) et (1.89) deviennent :

$$M = M_m \tag{1.91}$$

$$M_m \cdot \bar{M} + \bar{M}_m \cdot M = M_m \oplus M = 0 \tag{1.92}$$

L'expression (1.91) est une équation logique (§ 1.6.8) équivalente à la relation (1.92).

Des développements plus complets peuvent être trouvés dans la référence [22] (§ 4.2.2).

### 1.7.15 Exercice

On demande de vérifier les relations suivantes dans une table de vérité (voir fig. 1.66) :

$$MAJ(a, b, d) \in M(a, b, c, d) \tag{1.93}$$

$$MAJ(a, c, d) \in M(a, b, c, d) \tag{1.94}$$

$$MAJ(b, c, d) \in M(a, b, c, d) \tag{1.95}$$

Représenter dans cette même table la fonction  $\overline{M(a, b, c, d)}$  puis déterminer les bornes inférieure et supérieure, ainsi que les fonctions indifférente et différence. En s'inspirant des relations (1.93) à (1.95) peut-on trouver des solutions simples de  $\overline{M(a, b, c, d)}$  ?

### □ 1.7.16 Exercice

En considérant que  $M(a, b, c, d) = M_m + M_\emptyset = M_m + \lambda \cdot M_M$ , on demande de trouver les valeurs particulières de  $M_\emptyset$  et de  $\lambda$  définissant la solution  $MAJ(a, b, c)$  (fig. 1.66).

### □ 1.7.17 Exercice

Démontrer algébriquement que  $M = M_m + \lambda \cdot M_M$  est une solution de l'équation (1.89) quel que soit  $\lambda$  et pour autant qu'on respecte la condition (1.90). Démontrer enfin que :

$$\bar{M} = \bar{M}_M + \lambda \cdot \bar{M}_m \tag{1.96}$$

## 1.8 CONCLUSIONS

## 1.8.1 Postulats de l'algèbre logique

On a vu que toute fonction logique est une solution de la fonction universelle (§ 1.5.8); celle-ci peut toujours être exprimée à l'aide d'un polynôme. Pour deux variables, par exemple, on a obtenu l'expression (1.42) :

$$Z = K_0 \bar{a} \bar{b} + K_1 \bar{a} b + K_2 a \bar{b} + K_3 a b$$

dont un logigramme possible est illustré par la figure 1.48.

On peut donc décrire tout système combinatoire à l'aide :

- des constantes et des variables logiques,
- des trois fonctions logiques NON (fig. 1.69), ET et OU (fig. 1.70)

dont les définitions peuvent être résumées par quatre postulats qui constituent les fondements de l'algèbre logique.

$a$	$\bar{a}$
0	1
1	0

Fig. 1.69

$a$	$b$	$ab$	$a + b$	$a \uparrow b$	$a \downarrow b$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	0

Fig. 1.70

## 1.8.2 Postulat 1 : variable logique

$$a = 1 \text{ si et seulement si } a \neq 0 \quad (1.97)$$

## 1.8.3 Postulat 2 : fonction NON

$$\bar{0} = 1 ; \bar{1} = 0 \quad (1.98)$$

## 1.8.4 Postulat 3 : fonction ET

$$0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0 ; 1 \cdot 1 = 1 \quad (1.99)$$

## 1.8.5 Postulat 4 : fonction OU

$$0 + 0 = 0 ; 0 + 1 = 1 + 0 = 1 + 1 = 1 \quad (1.100)$$

## 1.8.6 Définition

Par analogie avec la définition des opérateurs complets (§ 1.4.10), on appelle *famille d'opérateurs complète* tout ensemble d'un ou plusieurs types différents d'opérateurs dont trois assemblages distincts réalisent les fonctions NON, ET et OU.

### 1.8.7 Familles d'opérateurs complètes

Il découle de la définition précédente que :

- la famille {NON, ET, OU} est complète;
- la famille {NAND} est complète, l'opérateur NAND étant lui-même complet;
- la famille {NOR} est complète, l'opérateur NOR étant également complet.

La fonction NAND est décrite par l'expression  $a \uparrow b = \overline{a \cdot b}$  (fig. 1.70); on peut donc réaliser un opérateur NAND à l'aide d'un opérateur ET et d'un opérateur NON; on en déduit que la famille {NON, ET} est également complète.

La fonction NOR est définie par  $a \downarrow b = \overline{a + b}$  (fig. 1.70) : la famille {NON, OU} est également complète.

Les familles {NON, ET} et {NON, OU} étant complètes, on déduit que l'un des quatre postulats de l'algèbre logique (le postulat 4 ou le postulat 3) est théoriquement superflu. Les familles {NAND} et {NOR} étant complètes, il découle que les trois postulats 2, 3 et 4 peuvent être remplacés par un seul des deux nouveaux postulats 5 et 6 définissant les fonctions NAND et NOR.

### 1.8.8 Postulat 5 : fonction NAND

$$0 \uparrow 0 = 0 \uparrow 1 = 1 \uparrow 0 = 1 ; 1 \uparrow 1 = 0 \tag{1.101}$$

### 1.8.9 Postulat 6 : fonction NOR

$$0 \downarrow 0 = 1 ; 0 \downarrow 1 = 1 \downarrow 0 = 1 \downarrow 1 = 0 \tag{1.102}$$

### 1.8.10 Exercice

Parmi les seize fonctions de deux variables (fig. 1.14) quelles sont celles dont la réalisation produit un opérateur complet ?

Démontrer ensuite que les familles {OU-exclusif, ET} et {OU-exclusif, OU} sont complètes. L'opérateur MAJ ( $a, b, c$ ) est-il complet ?

### 1.8.11 Exercice

A titre de récapitulation, on cherche à réaliser les fonctions de chacune des familles complètes {NON, ET, OU}, {NAND}, {NOR}, {OU-exclusif, ET}, {OU-exclusif, OU} à

$x \backslash y$	{NON, ET, OU}	{NAND}	{NOR}	{OU-exclusif, ET}	{OU-exclusif, OU}
NON	$\bar{a}$	$a \uparrow 1$	$a \downarrow 0$	$a \oplus 1$	$a \oplus 1$
ET	$ab$	$(a \uparrow b) \uparrow 1$	$(a \downarrow 0) \downarrow (b \downarrow 0)$	$ab$	$[(a \oplus 1) + (b \oplus 1)] \oplus 1$
OU	$a + b$	$(a \uparrow 1) \uparrow (b \uparrow 1)$	$(a \downarrow b) \downarrow 0$	$a \oplus b \oplus ab$	$a + b$
NAND	$\overline{a \cdot b} = \bar{a} + \bar{b}$	$a \uparrow b$	$[(a \downarrow 0) \downarrow (b \downarrow 0)] \downarrow 0$	$a \oplus b \oplus 1$	$(a \oplus 1) + (b \oplus 1)$
NOR	$\overline{a + b} = \bar{a} \cdot \bar{b}$	$[(a \uparrow 1) \uparrow (b \uparrow 1)] \uparrow 1$	$a \downarrow b$	$(a \oplus 1)(b \oplus 1)$	$(a + b) \oplus 1$
OU-exclusif	$a\bar{b} + \bar{a}b$	$[a \uparrow (b \uparrow 1)] \uparrow [b \uparrow (a \uparrow 1)]$	$\{[(a \downarrow 0) \downarrow b] \downarrow [(b \downarrow 0) \downarrow a]\} \downarrow 0$	$a \oplus b$	$a \oplus b$
ET	$ab$	$(a \uparrow b) \uparrow 1$	$(a \downarrow 0) \downarrow (b \downarrow 0)$	$ab$	$[(a \oplus 1) + (b \oplus 1)] \oplus 1$
OU-exclusif	$a\bar{b} + \bar{a}b$	$[a \uparrow (b \uparrow 1)] \uparrow [b \uparrow (a \uparrow 1)]$	$\{[(a \downarrow 0) \downarrow b] \downarrow [(b \downarrow 0) \downarrow a]\} \downarrow 0$	$a \oplus b$	$a \oplus b$
OU	$a + b$	$(a \uparrow 1) \uparrow (b \uparrow 1)$	$(a \downarrow b) \downarrow 0$	$a \oplus b \oplus ab$	$a + b$

$x$  : fonctions à réaliser;  
 $y$  : opérateurs à disposition.

Fig. 1.71

l'aide des opérateurs des quatre autres familles; les résultats finals sont résumés dans la figure 1.71 : on demande d'effectuer les calculs intermédiaires.

### 1.8.12 Bibliographie

L'étude théorique de l'algèbre logique constitue un chapitre important de mathématiques traité notamment dans les dictionnaires [6] et [7], ainsi que dans l'encyclopédie [23] (pp. 1 - 47). On trouvera des développements plus orientés vers les systèmes logiques dans les références [8] (pp. 93 - 100), [9] (pp. 11 - 80), [10] (pp. 23 - 67), [11] (pp. 54 - 108) et [14] (pp. 31 - 66).

## SYNTHÈSE ET ANALYSE DES SYSTÈMES COMBINATOIRES

### 2.1 PRÉAMBULE

#### 2.1.1 Introduction

On a étudié au chapitre 1 un certain nombre d'opérateurs réalisant des fonctions logiques simples; la représentation d'un système combinatoire quelconque par une ou plusieurs fonctions et la réalisation de celles-ci par un assemblage d'opérateurs constituent le problème plus général de la synthèse qui fait l'objet du chapitre 2. L'analyse, dont la démarche est inverse, permet de calculer la fonction logique réalisée par un logigramme donné.

#### 2.1.2 Hypothèses

On a déjà vu (§ 1.4.7) qu'une fonction logique peut être réalisée par différents logigrammes qui sont fonctionnellement équivalents; chacun d'eux est un assemblage de divers opérateurs (NON, ET, OU, NAND, OU-exclusif, multiplexeurs, etc...) : il existe donc un choix considérable de solutions pour la synthèse d'un système donné. Nous nous limiterons à trois types d'assemblages qui sont rencontrés très fréquemment dans la pratique :

- les multiplexeurs;
- les assemblages d'opérateurs { NON, ET, OU };
- les assemblages d'opérateurs { NON, ET, OU, OU-exclusif }.

La méthode de transformation des logigrammes NON, ET, OU en logigrammes NAND ou NOR (§ 1.4.17) permettra d'appliquer aux assemblages d'opérateurs NAND ou NOR les résultats que nous obtiendrons avec les assemblages { NON, ET, OU }.

Le problème général de la réalisation d'une fonction logique à l'aide d'opérateurs quelconques est la *décomposition* : l'étude des méthodes de calcul correspondantes sort du cadre de cet exposé; on consultera à ce sujet l'ouvrage de Curtis [13] ainsi que les références [14] (pp. 323 - 339), [10] (pp. 515 - 529), [32] (pp. 257 - 286), [15] (pp. 114 - 159), [16] (pp. 51 - 132) et [98] (pp. 193 - 214).

#### 2.1.3 Définitions : transformation et simplification

A chacun des trois types d'assemblages choisis (§ 2.1.2) correspond une expression algébrique particulière; il est en effet aisé de réaliser :

- la forme canonique d'une fonction à l'aide d'un multiplexeur;

- un polynôme à l'aide des opérateurs {NON, ET, OU};
- une expression algébrique incluant la fonction OU-exclusif à l'aide des opérateurs {NON, ET, OU, OU-exclusif}.

Les diverses expressions algébriques qui représentent une même fonction sont dites équivalentes ou égales. Une *transformation* d'une expression algébrique est le passage de cette expression à une expression équivalente.

Un cas particulier de ces transformations est le passage de la forme canonique à un polynôme contenant le nombre minimal de lettres (§ 1.4.5) : c'est la *simplification*.

#### 2.1.4 Plan

L'expérience démontre l'intérêt particulier de la simplification qui est associée aux assemblages {NON, ET, OU} : il en découle le plan du chapitre 2. Après deux exemples illustrant le calcul algébrique et graphique de la simplification (sect. 2.1), une méthode générale sera exposée dans la section 2.2 et sera appliquée aux fonctions incomplètement définies (sect. 2.3) et aux fonctions multiples (sect. 2.5); la transformation faisant apparaître la fonction OU-exclusif sera traitée dans la section 2.6 tandis que la section 2.7 présentera un cas particulier de décomposition : la décomposition d'un système en plusieurs sous-systèmes identiques. Enfin, l'analyse des systèmes combinatoires fait l'objet de la section 2.4.

#### 2.1.5 Exemple

La table de vérité de la figure 2.1 rappelle la fonction OU dont la forme canonique s'exprime sous la forme :

$$Z(a, b) = \Sigma 1, 2, 3 = \bar{a}b + a\bar{b} + ab \quad (2.1)$$

L'expression  $\bar{a}b + a\bar{b} + ab$  (2.1) est donc équivalente (ou égale) à l'expression  $a + b$  (1.4) qui définit la fonction OU, de sorte qu'on peut écrire :

$$\bar{a}b + a\bar{b} + ab = a + b \quad (2.2)$$

La forme canonique (2.1) de même que la définition (1.4) sont des polynômes; le premier comporte trois monômes (qui sont des mintermes de la fonction) et six lettres, le second comporte deux lettres seulement. Le passage de la première à la seconde expression est une simplification dont le calcul algébrique peut être effectué de deux façons distinctes.

No	a	b	Z
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Fig. 2.1

### 2.1.6 Première simplification algébrique

Le théorème d'idempotence (1.7) permet de transformer (2.1) :

$$Z = \bar{a}b + a\bar{b} + ab + a\bar{b} \quad (2.3)$$

dont on déduit après deux mises en évidence :

$$Z = a(b + \bar{b}) + b(a + \bar{a}) = a \cdot 1 + b \cdot 1 = a + b$$

### 2.1.7 Seconde simplification algébrique

En appliquant deux fois le théorème du consensus (1.16) on obtient :

$$\bar{a}b + ab = ba + b\bar{a} = ba + b\bar{a} + bb = ba + b\bar{a} + b = b(1 + a + \bar{a}) = b \quad (2.4)$$

$$ab + a\bar{b} = ab + a\bar{b} + aa = a(1 + b + \bar{b}) = a \quad (2.5)$$

En remplaçant (2.4) et (2.5) dans l'expression (2.3) il vient enfin :

$$Z = a + b$$

### 2.1.8 Conclusion

Le calcul algébrique peut toujours être mis en oeuvre pour chercher la simplification d'une forme canonique; l'application des divers théorèmes de l'algèbre logique se fait intuitivement et le résultat obtenu dépend avant tout de l'expérience et de l'habileté du logicien; l'utilisation du théorème du consensus peut être rendue systématique : on est alors conduit aux méthodes décrites dans les références [14] (pp. 110 - 121) et [58] (pp. 80 - 87, 111 - 124).

Nous préférons cependant l'emploi d'une méthode graphique dont une première illustration sera donnée au paragraphe 2.1.12 et qui sera traitée exhaustivement dans la section 2.2.

### 2.1.9 Logigrammes

En réalisant à l'aide des opérateurs NON, ET, OU les expressions algébriques (2.1) et (1.4) on obtient les logigrammes des figures 2.2 et 2.3 qui sont équivalents; la simplification algébrique entraîne naturellement une diminution du nombre des composants de l'assemblage final : par commodité de langage, on dira qu'on a simplifié le logigramme initial.

Le multiplexeur de la figure 2.4 réalise également la fonction  $Z$  décrite par la forme canonique (2.1) et par la table de vérité de la figure 2.1; il n'est pas possible d'affirmer que cette réalisation est plus ou moins simple que celles des figures 2.2 et 2.3. La simplicité relative de plusieurs logigrammes équivalents d'une même fonction ne peut être évaluée qu'au sein d'une même famille d'opérateurs.

Une discussion plus approfondie sur les critères de simplification peut être trouvée dans les références [10] (pp. 114 - 116), [12] (pp. 126 - 129), [18] (pp. 52 - 53) et [33] (pp. 114 - 139).

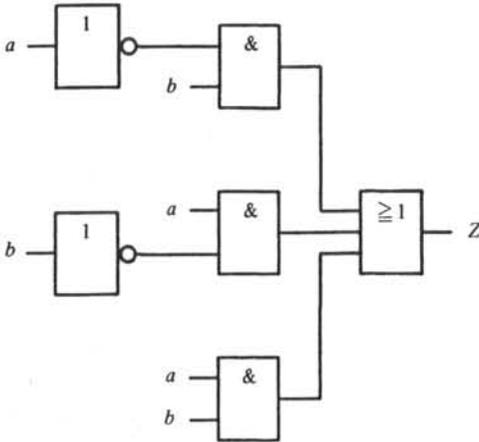


Fig. 2.2

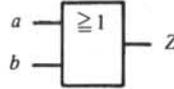


Fig. 2.3

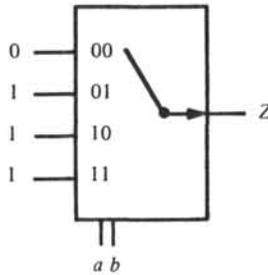


Fig. 2.4

### 2.1.10 Exemple

Etant donné les risques inhérents aux voyages aériens, une compagnie d'aviation recrute son personnel volant seulement si celui-ci satisfait l'une au moins des conditions suivantes :

- être un célibataire masculin de nationalité suisse (condition *a*);
- être célibataire, suisse et avoir moins de 25 ans (condition *b*);
- être une femme de nationalité étrangère, célibataire (condition *c*);
- être un homme âgé de moins de 25 ans (condition *d*);
- être célibataire et avoir plus de 25 ans (condition *e*).

On demande de réaliser une machine susceptible de choisir les candidats.

### 2.1.11 Modes de représentation

L'énoncé verbal du paragraphe 2.1.10 constitue le cahier des charges du système recherché : on dit que cet énoncé est exprimé dans un langage naturel; il en découlera une fonction logique et divers modes de représentation de celle-ci qui sont des langages formels.

Le problème posé se ramène à l'étude d'une fonction  $Z$  de quatre variables  $D, C, B, A$  si l'on convient du codage suivant :

- la variable  $D$  décrit la nationalité du candidat ( $D = 1$  : suisse;  $D = 0$  : étranger);
- la variable  $C$  décrit l'état civil ( $C = 1$  : célibataire;  $C = 0$  : marié);
- la variable  $B$  décrit le sexe ( $B = 1$  : masculin;  $B = 0$  : féminin);
- la variable  $A$  décrit l'âge ( $A = 1$  : moins de 25 ans;  $A = 0$  : plus de 25 ans);
- la fonction  $Z$  traduit la décision de la machine ( $Z = 1$  : candidat accepté;  $Z = 0$  : candidat refusé).

Le candidat est accepté ( $Z = 1$ ) si l'une au moins des cinq conditions  $a, b, c, d, e$  est satisfaite (c'est-à-dire égale à 1). On peut donc écrire :

$$Z = a + b + c + d + e \quad (2.6)$$

Chaque condition peut à son tour être exprimée en fonction des quatre variables  $D, C, B, A$  :

$$a = DCB; b = DCA; c = \bar{D}C\bar{B}; d = BA; e = C\bar{A} \quad (2.7)$$

En introduisant (2.7) dans (2.6) on a :

$$Z = DCB + DCA + \bar{D}C\bar{B} + BA + C\bar{A} \quad (2.8)$$

No	D	C	B	A	DCB	DCA	$\bar{D}C\bar{B}$	BA	$C\bar{A}$	Z
0	0	0	0	0						0
1	0	0	0	1						0
2	0	0	1	0						0
3	0	0	1	1				1		1
4	0	1	0	0			1		1	1
5	0	1	0	1			1			1
6	0	1	1	0					1	1
7	0	1	1	1				1		1
8	1	0	0	0						0
9	1	0	0	1						0
10	1	0	1	0						0
11	1	0	1	1				1		1
12	1	1	0	0					1	1
13	1	1	0	1		1				1
14	1	1	1	0	1				1	1
15	1	1	1	1	1	1		1		1

Fig. 2.5

Cette expression algébrique est un polynôme dont la valeur est calculée dans les seize états  $D, C, B, A$  : on obtient la table de vérité de la fonction  $Z$  (fig. 2.5) dont il découle la forme canonique :

$$Z(D,C,B,A) = \Sigma 3, 4, \dots, 7, 11, 12, \dots, 15 = \bar{D}\bar{C}BA + \bar{D}C\bar{B}\bar{A} + \bar{D}C\bar{B}A + \bar{D}CB\bar{A} + \bar{D}CBA + D\bar{C}BA + DC\bar{B}\bar{A} + DC\bar{B}A + DCB\bar{A} + DCBA \quad (2.9)$$

Une extension du diagramme de Venn de trois variables (fig. 1.59) à quatre variables est illustrée par la figure 2.6; la fonction  $Z$  (2.9) est alors représentée par la figure 2.7.

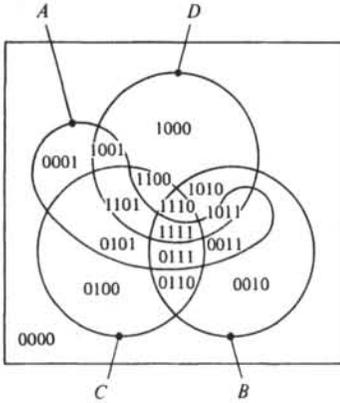


Fig. 2.6

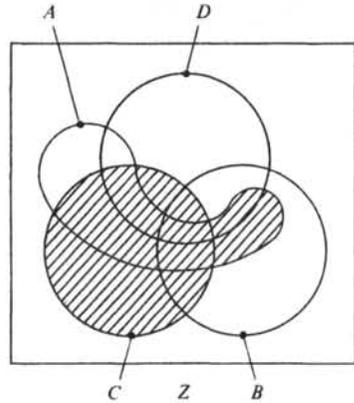


Fig. 2.7

A partir de la configuration proposée par Karnaugh (fig. 2.8) il est possible de donner une autre représentation de la fonction  $Z$  (fig. 2.9).

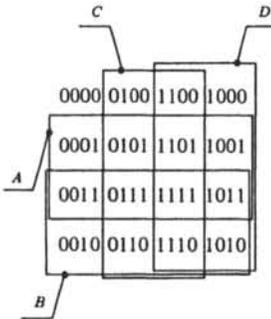


Fig. 2.8

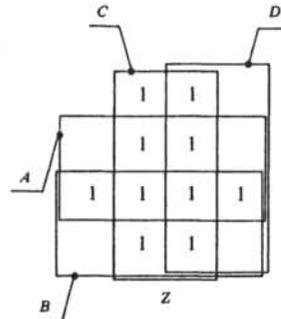


Fig. 2.9

### 2.1.12 Simplification graphique

L'examen du diagramme de Venn (fig. 2.7) et de la représentation de Karnaugh (fig. 2.9) met en évidence ce qui suit : la fonction  $Z$  est la réunion de la variable  $C$  et de l'intersection des variables  $B$  et  $A$ . On peut donc écrire :

$$Z = C + BA \quad (2.10)$$

Si l'on calcule la relation (2.10) dans la table de vérité (fig. 2.5) on constate qu'elle réalise la fonction  $Z$ . Les expressions algébriques (2.8), (2.9) et (2.10) sont donc équiva-

lentes; les passages de la forme canonique (2.9) aux polynômes (2.8) et (2.10) sont des transformations; on démontrera plus loin (sect. 2.2) que parmi tous les polynômes qui réalisent la fonction  $Z$ , celui décrit par la relation (2.10) comporte un nombre minimal de lettres (trois) : le passage de la forme canonique au polynôme (2.10) est donc une simplification de la fonction  $Z$ .

La relation (2.10) démontre que la fonction  $Z$  ne dépend que des trois variables  $C$ ,  $B$  et  $A$ . Une table de vérité à huit lignes (fig. 2.10) peut donc représenter  $Z$ ; il en découle une nouvelle forme canonique, équivalente aux expressions (2.8), (2.9) et (2.10) :

$$Z(C, B, A) = \Sigma 3, 4, \dots, 7 = \bar{C}BA + C\bar{B}\bar{A} + C\bar{B}A + CB\bar{A} + CBA \quad (2.11)$$

No	C	B	A	BA	Z
0	0	0	0		0
1	0	0	1		0
2	0	1	0		0
3	0	1	1	1	1
4	1	0	0		1
5	1	0	1		1
6	1	1	0		1
7	1	1	1	1	1

Fig. 2.10

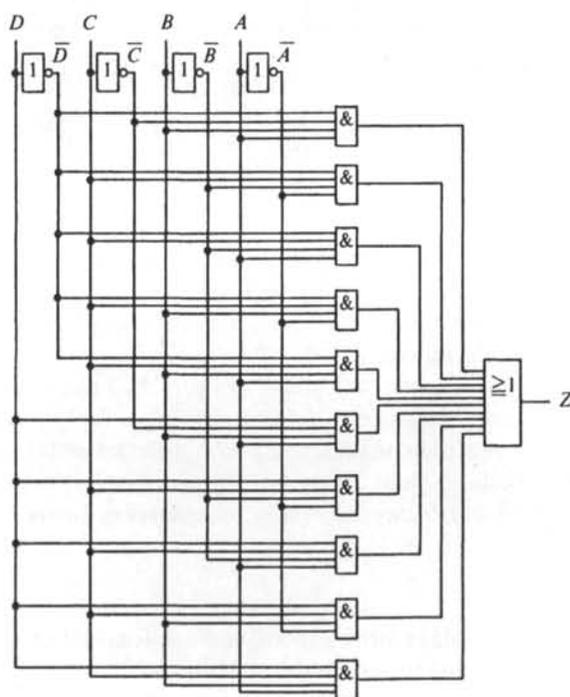


Fig. 2.11

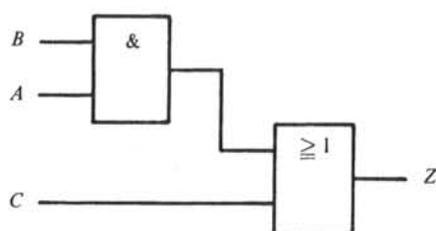


Fig. 2.12

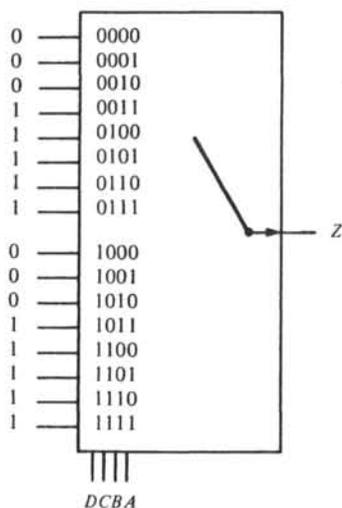


Fig. 2.13

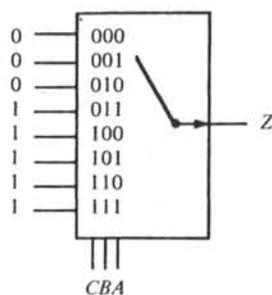


Fig. 2.14

### 2.1.13 Logigrammes

En réalisant à l'aide des opérateurs NON, ET, OU les relations (2.9) et (2.10) on obtient les logigrammes équivalents des figures 2.11 et 2.12 : on observe la diminution radicale du nombre d'opérateurs (ainsi que du nombre des fils reliant ces opérateurs) entraînée par la simplification.

Le multiplexeur à quatre variables de la figure 2.13 réalise la forme canonique (2.9) de la fonction Z; mais l'existence de la forme canonique (2.11) suggère une nouvelle réalisation de Z à l'aide d'un multiplexeur à trois variables seulement (fig. 2.14) : le nombre des entrées de celui-ci (huit paramètres et trois variables) est inférieur à celui du multiplexeur équivalent de la figure 2.13 (seize paramètres et quatre variables).

### 2.1.14 Conclusion

La simplification de la forme canonique d'une fonction de  $n$  variables peut montrer que cette fonction dépend seulement de  $p$  variables avec  $p < n$ . Dans ce cas, on dit que le multiplexeur à  $p$  variables réalisant la fonction donnée est plus simple que le multiplexeur à  $n$  variables équivalent.

### 2.1.15 Définitions

Étant donné un logigramme réalisant une fonction  $Z$ , on constate que chaque entrée de ce logigramme est reliée à la sortie par un ou plusieurs *chemins* composés d'une succession d'opérateurs et de fils. Il est possible de compter pour chacun de ces chemins le nombre d'opérateurs rencontrés : on appelle *profondeur* d'un logigramme (ou *nombre de niveaux* ou *nombre de couches*) le plus grand de ces nombres.

Les logigrammes des figures 2.11 et 2.12 comportent respectivement trois et deux niveaux; chaque niveau correspond dans ce cas à un opérateur NON, ET ou OU. Les logigrammes des figures 2.13 et 2.14 ont un seul niveau, puisque l'opérateur impliqué est un multiplexeur; si le multiplexeur était lui-même décrit par un logigramme détaillé du type NON, ET, OU (voir fig. 1.48), le nombre de niveaux serait égal à trois. La profondeur d'un système combinatoire dépend donc du choix des opérateurs qui constituent son logigramme.

### 2.1.16 Notation

Dans le chapitre 1 on a toujours représenté les diverses variables logiques dans l'ordre alphabétique normal, soit  $a, b, c, d, \dots$ . Cette représentation naturelle a été adoptée au début de ce volume pour en faciliter la compréhension; il s'avère que l'ordre alphabétique inverse, soit  $\dots D, C, B, A$  est en correspondance étroite avec les systèmes de numérotation pondérés (sect. 8.1) et permet aisément d'étendre des résultats obtenus pour un petit nombre de variables à un nombre plus élevé de celles-ci. Dès le chapitre 2, on adoptera donc l'ordre alphabétique inverse dans la plupart des problèmes traités.

### 2.1.17 Exercice

Démontrer à l'aide d'un diagramme de Venn les relations suivantes :

$$\bar{a}b + a\bar{b} + ab = a + b; \quad a + \bar{a}b = a + b; \quad ax + b\bar{x} + ab = ax + b\bar{x}; \\ (a + b)(a + c) = a + bc; \quad \overline{abc} = \bar{a} + \bar{b} + \bar{c}$$

### 2.1.18 Exercice

Effectuer algébriquement (par des mises en évidence et/ou par l'utilisation du théorème du consensus) les simplifications suivantes :

- $\bar{a}bc + a\bar{b}c + ab\bar{c} + abc = ab + ac + bc$
- $DCB + DCA + \bar{D}C\bar{B} + BA + C\bar{A} = C + BA$
- $\bar{C}BA + C\bar{B}\bar{A} + C\bar{B}A + CBA = C + BA$

### 2.1.19 Exercice

Déterminer la profondeur des logigrammes des figures 1.40, 1.42 et 1.43.

### 2.1.20 Exercice

A partir de la fonction  $Z$  définie par la figure 2.5 on demande la simplification algébrique et graphique de  $\bar{Z}$ ; tracer le logigramme NAND correspondant.

## 2.2 SIMPLIFICATION DES EXPRESSIONS ALGÈBRIQUES

### 2.2.1 Définition : table de Karnaugh

On a déjà suggéré que le diagramme de Venn (fig. 2.6) était équivalent à un autre mode de représentation (fig. 2.8) dont la forme habituelle est celle de la figure 2.15 et qu'on appelle *table de Karnaugh* à quatre variables [25]. Les figures 2.16 et 2.17 illustrent deux variantes de la même table : dans la seconde, chaque case contient le numéro décimal de l'état  $D, C, B, A$  correspondant.

On peut aussi considérer la table de Karnaugh comme une transformation de la table de vérité (fig. 2.5) : aux seize cases de la première correspondent les seize lignes de la seconde; on passe d'une représentation unidimensionnelle des seize états  $D, C, B, A$  (table de vérité) à une représentation bidimensionnelle (table de Karnaugh) dans laquelle chaque case  $D, C, B, A$  est l'intersection d'une colonne  $D, C$  et d'une ligne  $B, A$ .

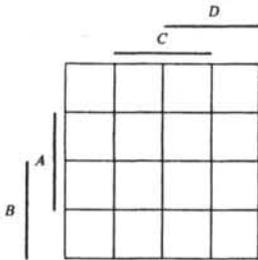


Fig. 2.15

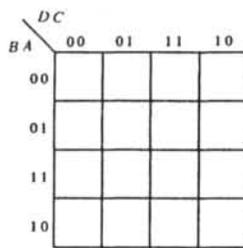


Fig. 2.16

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

Fig. 2.17

### 2.2.2 Représentation des variables

Chaque variable logique peut être représentée sous la forme vraie ( $D, C, B, A$ ) ou sous la forme complémentaire ( $\bar{D}, \bar{C}, \bar{B}, \bar{A}$ ) dans une table de Karnaugh. Les figures 2.18, 2.19 et 2.20 illustrent à titre d'exemple la représentation des variables  $D, A$  et  $\bar{A}$ ; on constate que chacune d'elles remplit un ensemble de huit cases; chacune de ces cases est marquée du symbole 1.

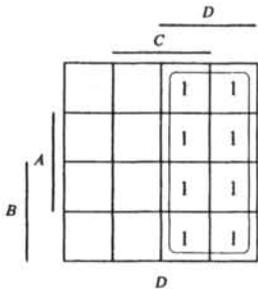


Fig. 2.18

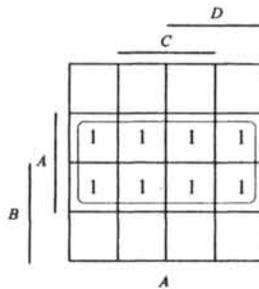


Fig. 2.19

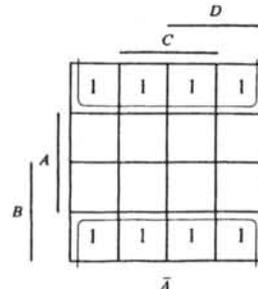


Fig. 2.20

### 2.2.3 Représentation des monômes

Tout produit de deux variables se traduit dans la table de Karnaugh par l'intersection des parties de la table affectées à ces variables. Les figures 2.21, 2.22 et 2.23 représentent les monômes  $DC, DA$  et  $\bar{C}\bar{A}$ ; chacun d'eux remplit quatre cases.

Tout produit de trois variables peut également être représenté dans la table de Karnaugh par l'intersection des variables en question : les monômes  $DCA$ ,  $D\bar{B}A$  et  $\bar{D}\bar{C}\bar{A}$  sont illustrés par les figures 2.24, 2.25 et 2.26 et remplissent chacun deux cases.

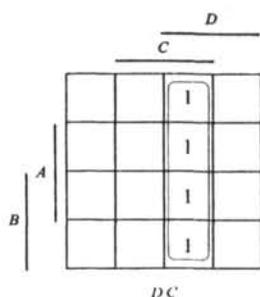


Fig. 2.21

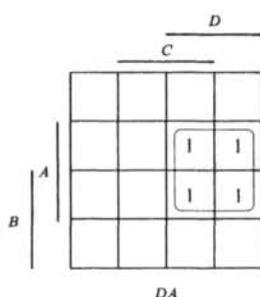


Fig. 2.22

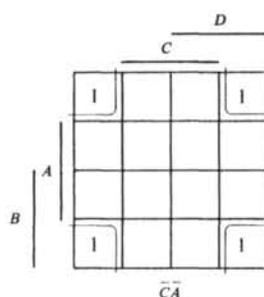


Fig. 2.23

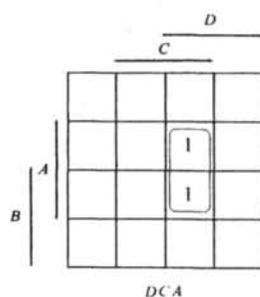


Fig. 2.24

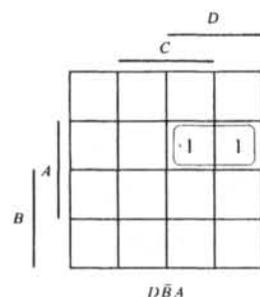


Fig. 2.25

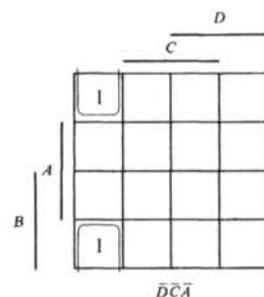


Fig. 2.26

### 2.2.4 Représentation des mintermes

Un minterme est un monôme particulier comportant les quatre variables  $D, C, B, A$ . Les figures 2.27 à 2.29 représentent les mintermes  $DCBA$ ,  $D\bar{C}\bar{B}A$  et  $\bar{D}\bar{C}\bar{B}\bar{A}$  qui remplissent chacun une seule case.

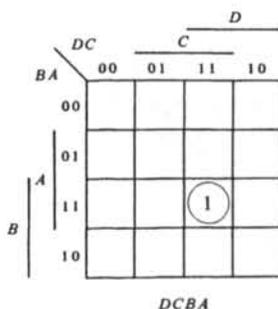


Fig. 2.27

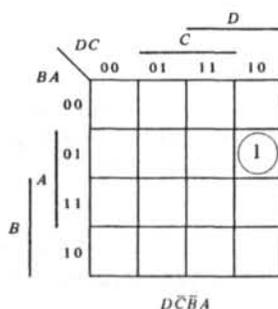


Fig. 2.28

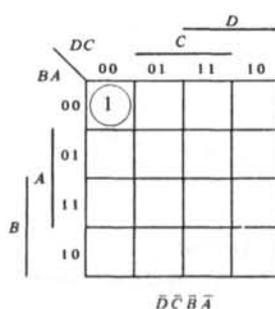


Fig. 2.29

### 2.2.5 Définition

Chaque variable ou monôme remplit un ensemble de cases dans la table de Karnaugh; un tel ensemble est appelé *bloc* associé à la variable ou au monôme.

### 2.2.6 Propriétés

On constate que tout bloc est constitué de  $p$  cases disposées en carré ou en rectangle, avec  $p$  égal à une puissance de deux (8, 4, 2, 1). Les blocs des figures 2.20, 2.23 et 2.26 peuvent être assimilés à des carrés ou à des rectangles si l'on imagine que la table de Karnaugh est refermée sur elle-même selon la forme toroïdale esquissée par la figure 2.30.

On remarque que les blocs les plus grands (huit cases) correspondent aux expressions algébriques les plus simples (une lettre), tandis que les blocs les plus petits (une case) sont décrits par les expressions les plus compliquées (quatre lettres).

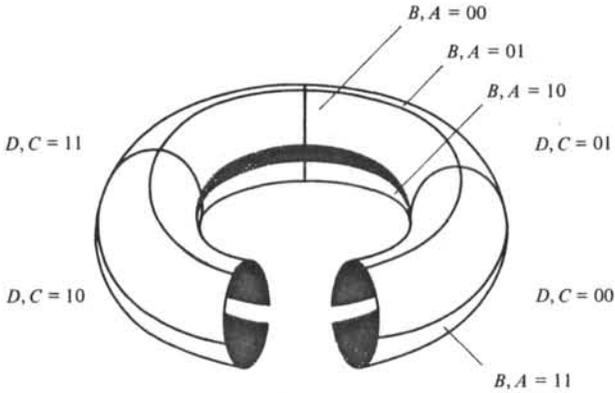


Fig. 2.30

### 2.2.7 Exercice

Les formes canoniques suivantes peuvent-elles être représentées par un bloc unique dans la table de Karnaugh :

$$Z_1(D, C, B, A) = \Sigma 0, 5; Z_2(D, C, B, A) = \Sigma 2, 4; Z_3(D, C, B, A) = \Sigma 2, 7, 8, 13$$

### 2.2.8 Exercice

A l'aide d'une table de Karnaugh, on demande d'exprimer les formes canoniques suivantes avec un seul monôme :

$$\begin{aligned} Z_1(D, C, B, A) &= \Sigma 4, 6; Z_2(D, C, B, A) = \Sigma 3, 11; Z_3(D, C, B, A) = \Sigma 3, 7, 11, 15; \\ Z_4(D, C, B, A) &= \Sigma 8, 9, 10, 11; Z_5(D, C, B, A) = \Sigma 2, 3, 10, 11; Z_6(D, C, B, A) \\ &= \Sigma 8, 10, 12, 14; Z_7(D, C, B, A) = \Sigma 0, 1, 2, 3, 8, 9, 10, 11; Z_8(D, C, B, A) \\ &= \Sigma 0, 1, \dots, 7; Z_9(D, C, B, A) = \Sigma 0, 1, \dots, 15 \end{aligned}$$

### 2.2.9 Exercice

Représenter dans une table de Karnaugh les monômes suivants :

$$Z_1 = \bar{B}A; Z_2 = \bar{D}\bar{A}; Z_3 = \bar{D}CB; Z_4 = DBA; Z_5 = \bar{C}\bar{B}\bar{A}; Z_6 = DCB\bar{A}$$

### 2.2.10 Exercice

A l'aide de l'algèbre logique et en tenant compte de la disposition de la table de Karnaugh, on demande de démontrer les propriétés du paragraphe 2.2.6.

### 2.2.11 Représentation des polynômes

Tout polynôme est représenté dans la table de Karnaugh par la réunion des blocs qui décrivent ses monômes. Considérons par exemple le polynôme  $DA + DC$  dont les monômes sont déjà illustrés par les figures 2.21 ( $DC$ ) et 2.22 ( $DA$ ). Leur réunion est effectuée dans la table de Karnaugh de la figure 2.31 qui représente le polynôme donné.

Les figures 2.32 et 2.33 représentent les polynômes  $\bar{C}\bar{A} + D\bar{B}A$  (voir les monômes sur les figures 2.23 et 2.25) et  $DCA + D\bar{B}A + \bar{D}\bar{C}\bar{A}$  (monômes : fig. 2.24 à 2.26).

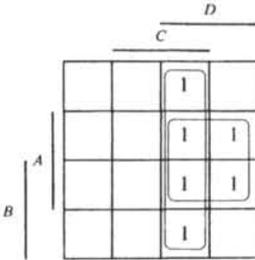


Fig. 2.31

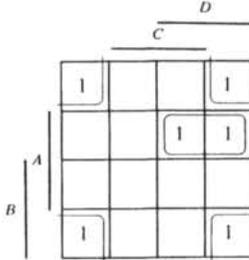


Fig. 2.32

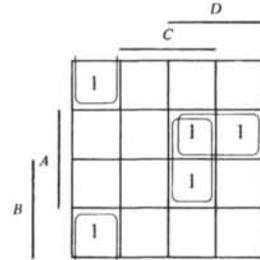


Fig. 2.33

### 2.2.12 Définition : impliquant premier

On appelle *impliquant premier* ou *monôme premier* d'un polynôme tout monôme qui n'est pas totalement inclus dans un monôme plus grand (c'est-à-dire comportant moins de lettres) du polynôme donné.

Si l'on considère le polynôme  $Z = DC + \bar{D}\bar{C}\bar{A} + DCA$  de la figure 2.34, on constate que le monôme  $DCA$  est totalement inclus dans le monôme  $DC$ . Le monôme  $DCA$  n'est donc pas un impliquant premier; par contre le monôme  $DC$  en est un.

Algébriquement, on vérifie la relation :

$$DC + DCA = DC(1 + A) = DC \quad (2.12)$$

On démontre ainsi qu'un monôme qui n'est pas un impliquant premier peut être supprimé du polynôme auquel il appartient.

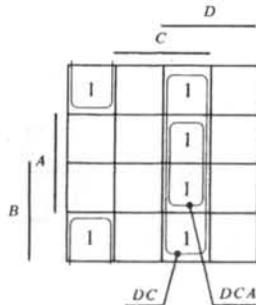


Fig. 2.34

### 2.2.13 Théorème

Tout polynôme réalisant une fonction avec un nombre minimal de lettres est une somme d'impliquants premiers de cette fonction.

### 2.2.14 Démonstration

Tout polynôme est une somme de monômes; par définition (§ 2.2.12), tout monôme qui n'est pas un impliquant premier est inclus dans un impliquant premier et peut être remplacé par ce dernier; l'impliquant premier comporte moins de lettres que le monôme inclus; en remplaçant chaque monôme par l'impliquant premier dans lequel il est inclus, on obtient un polynôme équivalent comportant un nombre minimal de lettres.

### 2.2.15 Méthode

La simplification d'une fonction logique est, par définition, le passage de sa forme canonique à un polynôme contenant le nombre minimal de lettres. Grâce au théorème du paragraphe 2.2.13 nous obtenons la méthode suivante :

- une fonction logique est introduite dans la table de Karnaugh;
- on recherche visuellement l'ensemble des blocs représentant les monômes du polynôme recherché;
- tout bloc totalement inclus dans un bloc plus grand est éliminé : le monôme correspondant n'est pas un impliquant premier;
- le polynôme final est représenté par la réunion des blocs restants : c'est une somme d'impliquants premiers.

### 2.2.16 Exemple

On cherche à simplifier la fonction

$$Z(D, C, B, A) = \Sigma 0, 1, 2, 3, 10, 11 \quad (2.13)$$

L'examen de la figure 2.35 fait apparaître deux blocs de quatre cases, correspondant aux impliquants premiers  $\overline{D}\overline{C}$  et  $\overline{C}B$ . Le polynôme obtenu est donc :

$$Z = \overline{D}\overline{C} + \overline{C}B \quad (2.14)$$

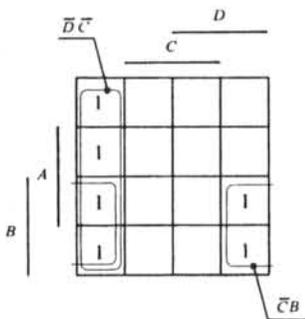


Fig. 2.35

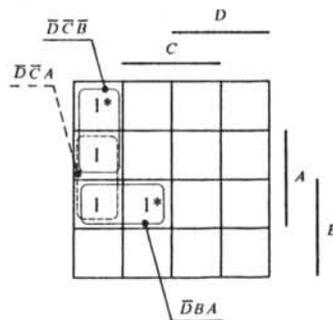


Fig. 2.36

### 2.2.17 Exemple

La figure 2.36 représente la fonction

$$Z(D, C, B, A) = \Sigma 0, 1, 3, 7 \quad (2.15)$$

La recherche des blocs fait apparaître dans la table de Karnaugh trois impliquants premiers :

$$Z = \bar{D}\bar{C}\bar{B} + \bar{D}\bar{C}A + \bar{D}BA \quad (2.16)$$

Il apparaît que l'impliquant premier  $\bar{D}\bar{C}A$  est inclus dans la réunion des deux autres impliquants  $\bar{D}\bar{C}\bar{B} + \bar{D}BA$ ; il peut donc être supprimé de l'expression algébrique finale de  $Z$  :

$$Z = \bar{D}\bar{C}\bar{B} + \bar{D}BA \quad (2.17)$$

### 2.2.18 Conclusion

Le théorème du paragraphe 2.2.13 a montré que la recherche des impliquants premiers était une condition nécessaire pour simplifier une fonction logique. L'exemple du paragraphe 2.2.17 démontre que cette condition n'est pas suffisante : le choix d'un nombre minimal d'impliquants premiers doit faire l'objet d'une seconde méthode exposée au paragraphe 2.2.20 et due à McCluskey [17] et Petrick [10] (pp. 155 - 159).

### 2.2.19 Définition

On appelle *table des impliquants premiers* d'une fonction logique  $Z$  une table de vérité particulière comportant autant de lignes que de mintermes de  $Z$  et autant de colonnes que d'impliquants premiers de  $Z$ .

La figure 2.37 présente la table des impliquants premiers de  $Z = \Sigma 0, 1, 3, 7$  : elle comporte quatre lignes et trois colonnes correspondant aux impliquants premiers du polynôme (2.16) dont les valeurs ont été calculées dans les quatre états 0, 1, 3 et 7.

No	D	C	B	A	$\bar{D}\bar{C}\bar{B} = a$	$\bar{D}\bar{C}A = b$	$\bar{D}BA = c$
0	0	0	0	0	1*		
1	0	0	0	1	1	1	
3	0	0	1	1		1	1
7	0	1	1	1			1*

Fig. 2.37

### 2.2.20 Méthode

Nous recherchons le plus petit sous-ensemble d'impliquants premiers satisfaisant la couverture de la fonction donnée, c'est-à-dire contenant l'ensemble de ses mintermes; en d'autres termes on cherche à minimiser le nombre des colonnes de la table des impliquants tout en garantissant dans chaque ligne la présence d'au moins un 1.

Il est possible de traduire à l'aide d'une proposition logique  $P$  le principe énoncé. Dans l'exemple de la figure 2.37, cette proposition doit affirmer que *le minterme 0 ET le minterme 1 ET le minterme 3 ET le minterme 7 doivent être inclus dans un impliquant au moins*. Grâce à l'algèbre logique, on peut écrire  $P$  sous la forme :

$$P = 0 \cdot 1 \cdot 3 \cdot 7 \quad (2.18)$$

L'examen de la table des impliquants (fig. 2.37) indique que :

- le minterme 0 est inclus dans l'impliquant premier  $a$ ;
- le minterme 1 peut être inclus dans  $a$  OU  $b$  ( $a + b$ );
- le minterme 3 peut être inclus dans  $b$  OU  $c$  ( $b + c$ );
- le minterme 7 est inclus dans l'impliquant  $c$ .

En introduisant ces conditions dans la relation (2.18) on obtient :

$$P = a(a + b)(b + c)c \quad (2.19)$$

Grâce au théorème (1.18) cette expression peut être ramenée à la forme plus simple :

$$P = a \cdot c \quad (2.20)$$

La signification de cette nouvelle forme de la proposition logique est la suivante : *les impliquants  $a$  ET  $c$  assurent la couverture de la fonction  $Z$* . Il en découle la simplification de  $Z$  :

$$Z = a + c = \bar{D}\bar{C}\bar{B} + \bar{D}BA \quad (2.21)$$

La suppression de l'impliquant  $b = \bar{D}\bar{C}A$  (§ 2.2.17) est donc démontrée.

### 2.2.21 Exemple

On cherche à simplifier la fonction suivante :

$$Z(D, C, B, A) = \Sigma 3, 5, 7, 8, 12, 13 \quad (2.22)$$

La recherche des blocs dans la table de Karnaugh (fig. 2.38) fait apparaître cinq impliquants premiers qui sont reportés dans la table des impliquants de la figure 2.39. On en tire une proposition logique  $P$  :

$$P = 3 \cdot 5 \cdot 7 \cdot 8 \cdot 12 \cdot 13 = a(b + c)(a + b)d(d + e)(c + e) \quad (2.23)$$

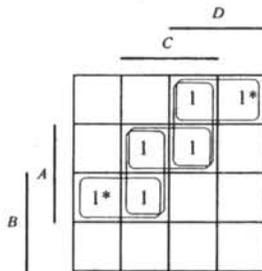


Fig. 2.38

Grâce aux théorèmes (1.18) et (1.20) il vient successivement :

$$\begin{aligned} P &= ad(b + c)(c + e) = ad(bc + be + c + ce) = ad(be + c), \text{ donc :} \\ P &= ad(c + be) = adc + adbe \end{aligned} \quad (2.24)$$

La forme finale (2.24) de  $P$  signifie que deux sous-ensembles d'impliquants premiers satisfont le problème posé :  $\{a, d, c\}$  et  $\{a, d, b, e\}$ . On dit que chacun de ces sous-ensembles est une solution de la proposition  $P$ ; on en déduit deux expressions équivalentes de la fonction  $Z$  :

$$Z = a + d + c = \bar{D}BA + D\bar{B}\bar{A} + C\bar{B}A \quad (2.25)$$

$$Z = a + d + b + e = \bar{D}BA + D\bar{B}\bar{A} + \bar{D}CA + DC\bar{B} \quad (2.26)$$

Le polynôme (2.25) comporte le nombre minimal de lettres et constitue la forme simplifiée de  $Z$ .

No	$\bar{D}BA = a$	$\bar{D}CA = b$	$C\bar{B}A = c$	$D\bar{B}\bar{A} = d$	$DC\bar{B} = e$
3	1*				
5		1	1		
7	1	1			
8				1*	
12				1	1
13			1		1

Fig. 2.39

### 2.2.22 Définitions

On appelle *impliquant premier essentiel* un impliquant qui est inclus dans toutes les solutions de la proposition logique  $P$ ; la variable définissant cet impliquant peut être mise en facteur dans l'expression algébrique de  $P$ . La relation (2.24), par exemple, démontre que les impliquants  $a$  et  $d$  sont essentiels : ils peuvent être mis en facteur. La relation (2.20) illustre un exemple où tous les impliquants de  $P$  ( $a$  et  $c$ ) sont essentiels.

On remarque que les impliquants premiers essentiels sont caractérisés dans la table des impliquants (fig. 2.37 et 2.39) par la présence d'un seul 1 dans une ligne au moins : ce type de ligne est marqué d'un astérisque (\*). Dans la table de Karnaugh (fig. 2.36 et 2.38), ces impliquants remplissent une case au moins qui n'est pas incluse dans un autre impliquant : on a marqué d'un astérisque (\*) ce type de case.

On appelle *impliquant premier non essentiel* (ou *impliquant premier à choix*) tout impliquant qui n'est pas essentiel mais qui est inclus dans une solution au moins de la proposition logique  $P$ ; l'existence d'un tel impliquant n'est possible que si cette proposition a plusieurs solutions. La relation (2.24) montre que les impliquants  $c$ ,  $b$  et  $e$  sont non essentiels : ils ne peuvent pas être mis en facteur dans le polynôme  $P$ .

Enfin les *impliquants premiers superflus* sont les impliquants qui ont été éliminés lors des transformations de la proposition logique  $P$ ; l'impliquant  $b$  de la relation (2.19) est superflu.

### 2.2.23 Commentaire

Le calcul de la proposition logique  $P$  est la transformation d'un produit de sommes en une somme de produits : ce calcul, même dans des problèmes simples, peut être très malaisé (voir l'exemple du paragraphe 8.2.10). On est donc intéressé par toute variante

de la méthode esquissée ci-dessus : le paragraphe 8.2.12 illustre la conception d'un système logique susceptible de calculer les solutions d'une proposition logique  $P$ ; d'autres méthodes systématiques [10] (pp. 148 - 159), [91] ou intuitives [92], [93] ont été également développées. Le choix d'un sous-ensemble minimal d'impliquants premiers constitue un cas particulier du problème de couverture qui joue un rôle important dans d'autres chapitres de la théorie des systèmes logiques et des machines séquentielles [94], [95].

### 2.2.24 Exercice

On demande de représenter les expressions algébriques suivantes dans une table de Karnaugh à quatre variables, puis d'en déterminer la forme canonique décimale; on cherchera ensuite l'ensemble des impliquants premiers ainsi qu'un sous-ensemble minimal de ceux-ci :

- $Z_1 = DCB + DCA + \bar{D}C\bar{B} + BA + C\bar{A}$  (2.8)
- $Z_2 = \bar{a}b + a\bar{b} + ab$  (2.1)
- $Z_3 = MAJ(a, b, c) = \bar{a}bc + a\bar{b}c + abc + abc$  (1.63)
- $Z_4 = \bar{a}b\bar{c} + \bar{a}c\bar{d} + \bar{a}b\bar{d} + ac + bc\bar{d}$
- $Z_5 = abc + \bar{a}bd$
- $Z_6 = (a + b)(b + \bar{c})(a + d)$
- $Z_7 = a \oplus b \oplus c \oplus d$

### 2.2.25 Exercice

Simplifier à l'aide d'une table de Karnaugh les formes canoniques suivantes et déterminer un sous-ensemble minimal d'impliquants premiers :

- $E(a, b, c, d) = \Sigma 3, 5, 6, 9, 10, 12$  (fig. 1.66)
- $Z_1(D, C, B, A) = \Sigma 1, 5, 6, 7, 11, 12, 13, 15$
- $Z_2(D, C, B, A) = \Sigma 0, 2, 4, 5, 10, 11, 13, 15$
- $Z_3(D, C, B, A) = \Sigma 2, 5, 10, 11, 13, 15$
- $Z_4(D, C, B, A) = \Sigma 0, 2, 4, 5, \dots, 9, 11, 13, 15$

Simplifier dans chaque cas la fonction inverse à l'aide d'une table de Karnaugh également.

### 2.2.26 Exercice

Effectuer la synthèse d'un système combinatoire à quatre entrées  $a, b, c, d$  et une sortie  $Z$ ; la sortie  $Z$  prend la valeur 1 chaque fois que le numéro décimal de l'état  $a, b, c, d$  est divisible entièrement par 4 ou 5. Simplifier la fonction  $Z$  et réaliser un logigramme NAND.

### 2.2.27 Exercice

Deux nombres binaires  $A$  et  $B$  sont représentés chacun à l'aide de deux bits :  $A_1, A_0$  pour  $A$  et  $B_1, B_0$  pour  $B$ . Effectuer la synthèse d'un système combinatoire destiné à déterminer si  $A < B$ ,  $A = B$  ou  $A > B$ , puis simplifier les fonctions et réaliser un logigramme NAND, OU-exclusif.

## 2.2.28 Exercice

A l'aide de deux variables de commande  $C$  et  $D$ , on désire modifier le comportement d'un système combinatoire pour réaliser selon l'état  $C, D$  l'une des quatre fonctions suivantes :

$$Z = A + B; \quad Z = A \cdot B; \quad Z = \overline{A \cdot B}; \quad Z = A \oplus B$$

Simplifier la fonction  $Z$  recherchée et tracer un logigramme NAND, OU-exclusif.

## 2.2.29 Exercice

Etant donné l'affichage à sept segments de la figure 2.40, on demande de calculer un système combinatoire comportant sept fonctions  $a, b, \dots, g$  de quatre variables  $D, C, B, A$  et réalisant la table de vérité de la figure 2.41. Chacune des sept fonctions sera simplifiée en supposant que l'état 0 correspond à un segment allumé.

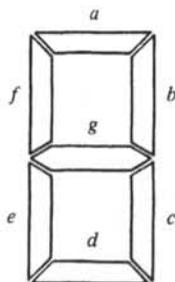


Fig. 2.40

No	D	C	B	A	Affichage
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Fig. 2.41

### 2.2.30 Fonctions de deux variables

La table de Karnaugh correspondante comporte quatre cases (fig. 2.42); son emploi est illustré par la simplification de la fonction OU (2.1) :

$$Z(a, b) = \Sigma 1, 2, 3 = \bar{a}\bar{b} + a\bar{b} + ab$$

L'examen de la figure 2.43 fait apparaître deux impliquants premiers essentiels; on retrouve la relation (2.2) :

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

		$\overline{a}$	
		0	1
$b$	0	0	2
	1	1	3

Fig. 2.42

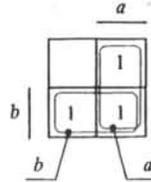


Fig. 2.43

### 2.2.31 Fonctions de trois variables

La table de Karnaugh comporte dans ce cas huit cases (fig. 2.44); on illustre son emploi par la fonction MAJORITÉ dont la forme canonique (1.64) est :

$$MAJ(a, b, c) = \Sigma 3, 5, 6, 7$$

L'examen de la table de Karnaugh correspondante (fig. 2.45) révèle l'existence de trois impliquants premiers essentiels :

$$MAJ(a, b, c) = ab + ac + bc \quad (2.27)$$

		$\overline{a}$			
		$\overline{b}$		$b$	
		00	01	11	10
$c$	0	0	2	6	4
	1	1	3	7	5

Fig. 2.44

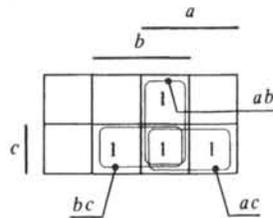


Fig. 2.45

### 2.2.32 Fonctions de cinq variables

La table de Karnaugh comporte alors  $2^5 = 32$  cases (fig. 2.46). On peut la construire en juxtaposant deux tables partielles de quatre variables  $D, C, B, A$  : la cinquième variable  $E$  distingue la table partielle de gauche ( $E = 0$ ) de celle de droite ( $E = 1$ ).

Illustrons l'emploi de cette table par l'exemple suivant :

$$Z(E, D, C, B, A) = \Sigma 0, 1, 2, 3, 6, 10, 11, 14, 16, 17, 18, 19, 26, 27 \quad (2.28)$$

BA		D				D				E
		000	001	011	010	100	101	111	110	
00	E DC	0	4	12	8	16	20	28	24	
		1	5	13	9	17	21	29	25	
11	A	3	7	15	11	19	23	31	27	
		2	6	14	10	18	22	30	26	
B		c				c				

Fig. 2.46

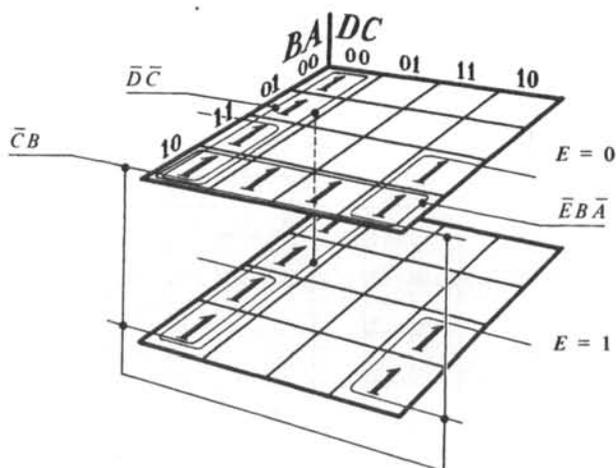


Fig. 2.47

La recherche des impliquants premiers est alors facilitée si l'on imagine que les deux tables partielles sont superposées dans l'espace (fig. 2.47) : on est alors amené à rechercher non seulement des blocs, mais également des cubes ou des parallélépipèdes rectangles comportant  $2^q$  cases. On détecte ainsi trois impliquants premiers essentiels :

- $\bar{D}\bar{C}$  représenté par un parallélépipède rectangle de huit cases;
- $\bar{C}B$  représenté par un cube de huit cases; ce cube est coupé par la moitié car il appartient à deux colonnes extrêmes des tables partielles : il faut donc garder à l'esprit la représentation toroïdale de la table de Karnaugh à quatre variables (fig. 2.30);
- $\bar{E}B\bar{A}$  représenté par un bloc de quatre cases.

La fonction simplifiée s'écrit alors :

$$Z = \bar{C}B + \bar{D}\bar{C} + \bar{E}B\bar{A} \quad (2.29)$$

dont la représentation habituelle est donnée par la figure 2.48.

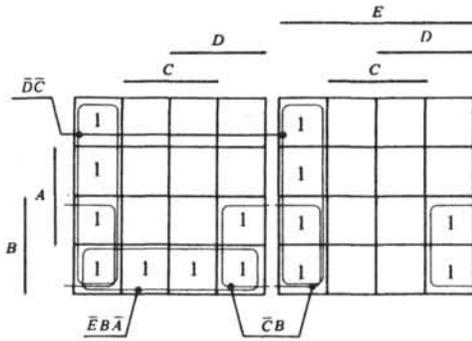


Fig. 2.48

### 2.2.33 Fonctions de six variables

La table de Karnaugh comporte  $2^6 = 64$  cases : elle peut se décomposer en quatre tables partielles de seize cases disposées dans le plan (fig. 2.49) ou dans l'espace (fig. 2.50).

DC		D				E			
		00	01	11	10	00	01	11	10
BA	00	1				1			1
	01	1				1			1
	11	1				1			1
	10	1				1			1
$F, E = 00$					$F, E = 01$				
DC		D				E			
		00	01	11	10	00	01	11	10
BA	00	32	36	44	40	48	52	60	56
	01	33	37	45	41	49	53	61	57
	11	35	39	47	43	51	55	63	59
	10	34	38	46	42	50	54	62	58
$F, E = 10$					$F, E = 11$				

Fig. 2.49

Dans la figure 2.50 on a simplifié la fonction  $Z$  :

$$\begin{aligned}
 Z(F, E, D, C, B, A) = & \\
 = \Sigma & 0, 1, 2, 3, 16, 17, 18, 19, 24, 25, 33, 35, 42, 43, 46, 47, 48, 49, 51, 56, 57 \\
 & \qquad \qquad \qquad (2.30)
 \end{aligned}$$

en procédant comme suit :

- on recherche dans chaque table partielle les plus grands blocs possibles;

- on cherche des blocs communs aux quatre tables partielles; on détecte l'impliquant  $\bar{D}\bar{C}A$ ;
- on cherche ensuite des blocs communs à deux tables partielles dont les états  $F, E$  sont distincts d'une seule variable : ces tables sont adjacentes dans l'espace à l'exception des deux tables extrêmes; on trouve ainsi les impliquants  $E\bar{C}\bar{B}$  et  $\bar{F}\bar{D}\bar{C}$ ;
- on considère enfin les blocs isolés, c'est-à-dire figurant dans une seule table partielle : c'est le cas de l'impliquant  $F\bar{E}DB$ .

La forme simplifiée de la fonction  $Z$  est alors :

$$Z = \bar{D}\bar{C}A + E\bar{C}\bar{B} + \bar{F}\bar{D}\bar{C} + F\bar{E}DB \quad (2.31)$$

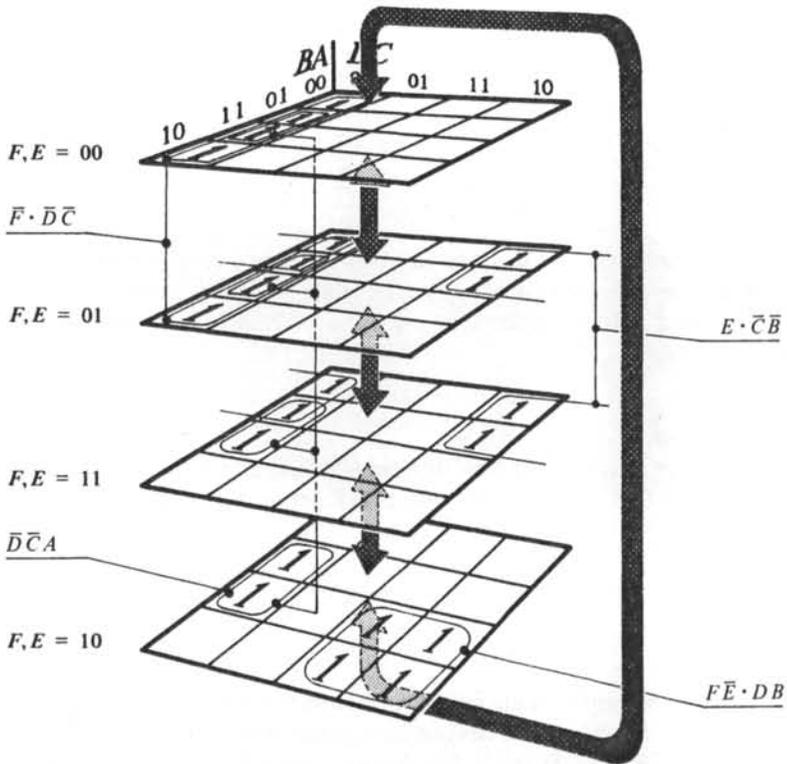


Fig. 2.50

### 2.2.34 Conclusion

La méthode de simplification de Karnaugh [25], [10] (pp. 103 - 128), [11] (pp. 109 - 128), [99] (pp. 168 - 252), [58] (pp. 63 - 71) est essentiellement visuelle et par conséquent quelque peu subjective; les résultats obtenus dépendent fortement de l'expérience et de l'intuition du logicien. Par ailleurs, on constate que l'emploi des tables de Karnaugh pour un nombre de variables supérieur à six est très malaisé, sinon impraticable. Pour tout problème comportant plus de six variables on utilisera des méthodes

systématiques dont la plus connue, celle de McCluskey, fait l'objet de la section 8.2. D'autres algorithmes de simplification ont été développés, en particulier dans l'optique du calcul automatique; les principaux sont dus à Scheinman [19], [20] (pp. 128 - 135) et à Tison [21], [14] (pp. 110 - 121), [33] (pp. 166 - 174), [58] (pp. 80 - 87, 111 - 124); la méthode de Tison, basée sur le théorème du consensus (§ 2.1.7), a été exploitée de façon approfondie par Dietmeyer [24] (pp. 245 - 299).

### 2.2.35 Exercice

Simplifier à l'aide d'une table de Karnaugh les expressions suivantes :

- $Z_1 = a \oplus b$
- $Z_2 = (b + \bar{c}) \cdot \bar{b}\bar{a}$
- $Z_3 (E, D, C, B, A) = \Sigma 0, 4, 9, 14, 25, 30$
- $Z_4 (E, D, C, B, A) = \Sigma 0, 2, 8, 11, 15, 18, 20, 21, 27, 28, 29, 31$
- $Z_5 (E, D, C, B, A) = \Sigma 0, 1, 4, 5, 6, 11, 12, 14, 16, 20, 22, 28, 30, 31$
- $Z_6 (E, D, C, B, A) = \Sigma 0, 1, 3, 4, 5, 7, 8, 9, 10, 12, 13, 21, 24, 25, 26, 28, 29$
- $Z_7 (F, E, D, C, B, A) = \Sigma 0, 24, 25, 28, 29, 32, 38, 39, 48, 54, 55, 56, 57, 60, 61$
- $Z_8 (F, E, D, C, B, A) = \Sigma 2, 3, 6, 7, 10, 14, 18, 19, 22, 23, 27, 37, 42, 43, 45, 46$

### 2.2.36 Exercice

Effectuer la synthèse d'un système combinatoire à cinq entrées  $E, D, C, B, A$  et une sortie  $Z$ ; cette sortie prend la valeur 1 chaque fois que le numéro décimal de l'état  $E, D, C, B, A$  est divisible entièrement par 6, 7 ou 9. Simplifier la fonction  $Z$  à l'aide d'une table de Karnaugh.

### 2.2.37 Exercice

Simplifier à l'aide d'une table de Karnaugh la fonction MAJORITÉ de cinq variables :  $Z = MAJ(E, D, C, B, A)$ ; simplifier ensuite la fonction inverse  $\bar{Z}$ .

## 2.3 FONCTIONS INCOMPLÈTEMENT DÉFINIES

### 2.3.1 Exemple

Un système combinatoire comporte sept sorties  $a, b, \dots, g$  et quatre entrées  $D, C, B, A$  (fig. 2.51) : chacune des sorties commande un segment lumineux qui peut être allumé (état 1) ou éteint (état 0) (fig. 2.52); à chacun des dix états  $D, C, B, A = 0000, \dots, 1001$  (numéros décimaux 0, 1, ..., 9) correspond l'affichage des chiffres 0, 1, ..., 9 selon la disposition de la figure 2.53.

### 2.3.2 Définitions

En représentant schématiquement les dix configurations de la figure 2.53 dans une table de Karnaugh à quatre variables (fig. 2.54) on peut résumer le cahier des charges du système recherché. Une telle table représente en fait les sept fonctions  $a, b, \dots, g$  : on l'appelle *table de Karnaugh principale*. Il est aisé d'en déduire sept tables partielles décrivant chacune une fonction : ce sont les *tables de Karnaugh secondaires* (fig. 2.55).

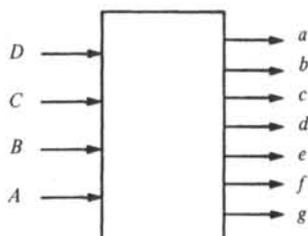


Fig. 2.51

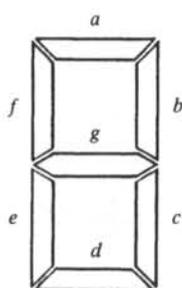


Fig. 2.52

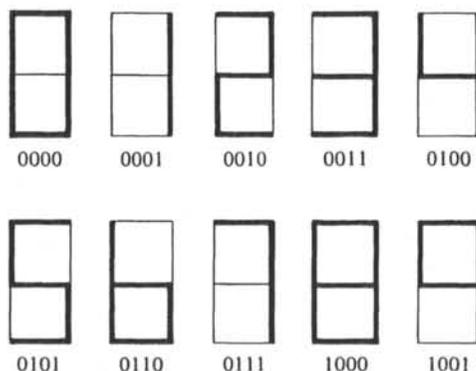


Fig. 2.53

	DC			
BA	00	01	11	10
00	0 0	4 4	12 12	8 8
01	1 1	5 5	13 13	9 9
11	3 3	7 7	15 15	11 11
10	2 2	6 6	14 14	10 10

Fig. 2.54

### 2.3.3 Mode de représentation

On constate que le cahier des charges du système étudié est incomplet : pour les six états  $D, C, B, A = 1010, \dots, 1111$  aucun affichage particulier n'est exigé; nous relevons donc l'existence de six états  $\emptyset$  (§ 1.7.2) communs aux sept fonctions : on représente alors ces états en reportant le symbole  $\emptyset$  dans les six cases correspondantes des tables de Karnaugh secondaires (fig. 2.55). Les fonctions  $a, b, \dots, g$  sont incomplètement définies et leur simplification nécessite une méthode particulière.

### 2.3.4 Théorème

La forme simplifiée d'une fonction incomplètement définie  $Z$  est une somme des impliquants premiers de la borne supérieure  $Z_M$  de cette fonction.

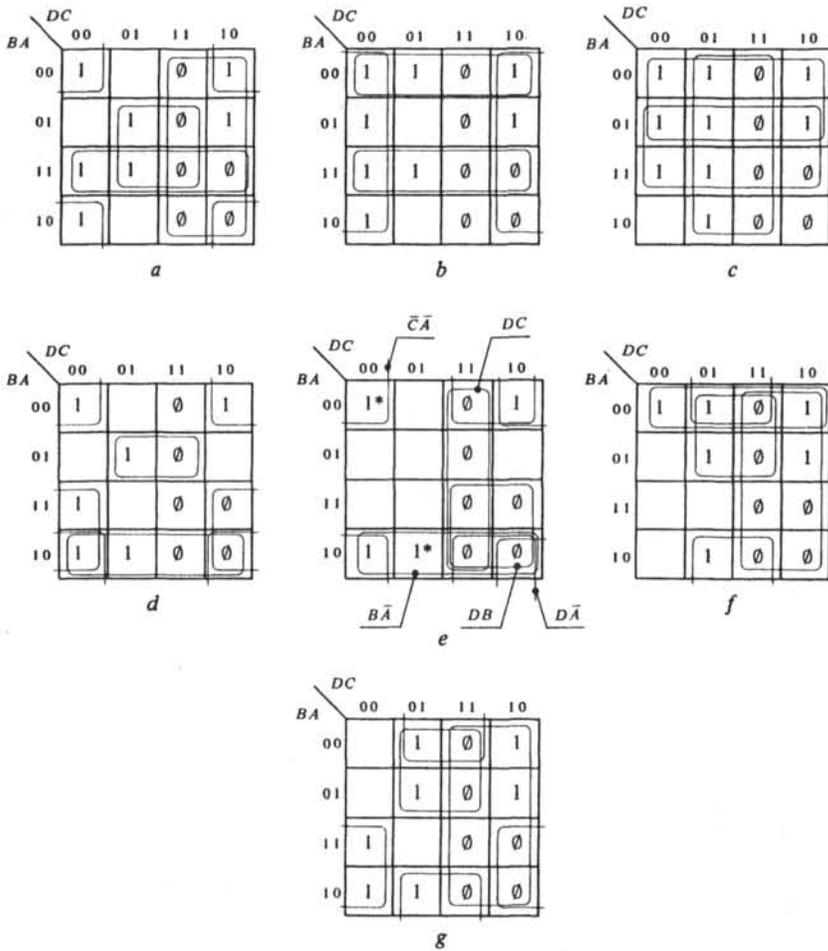


Fig. 2.55

### 2.3.5 Démonstration

On a déjà montré que toute fonction  $Z_x$  comprise entre la borne inférieure  $Z_m$  et la borne supérieure  $Z_M$  est une solution de  $Z$  (1.84); en particulier  $Z_M$  est une solution de  $Z$ . Il suffit alors de démontrer que la forme simplifiée de  $Z_M$  (la somme de ses impliquants premiers) est plus simple (comporte moins de lettres) que toute autre fonction  $Z_x < Z_M$ . La simplification de  $Z_x < Z_M$  donne une somme d'impliquants premiers qui sont inclus dans l'ensemble des impliquants de  $Z_M$  et qui sont par conséquent :

- soit des impliquants premiers de  $Z_M$ ;
- soit des monômes inclus dans les impliquants premiers de  $Z_M$ .

Dans ce dernier cas, on rappelle (§ 2.2.12) que les impliquants premiers de  $Z_M$  sont plus grands que les monômes inclus et sont par conséquent plus simples (ils comportent moins de lettres).

### 2.3.6 Méthode

Dans chaque état  $\emptyset$  on donne la valeur 1 à la fonction incomplètement définie  $Z$  : on obtient sa borne supérieure  $Z_M$ ; la recherche des impliquants premiers de  $Z_M$  se ramène alors à la méthode exposée dans le cas des fonctions complètement définies (§ 2.2.15).

### 2.3.7 Exemple

L'examen de la table de Karnaugh de la fonction  $e(D, C, B, A) = \Sigma 0, 2, 6, 8 + \emptyset 10, 11, \dots, 15$  (fig. 2.55) fait apparaître cinq blocs correspondant aux cinq impliquants premiers de la borne supérieure  $e_M$  :

$$\begin{aligned} e_M(D, C, B, A) &= \Sigma 0, 2, 6, 8, 10, 11, \dots, 15 \\ &= DC + DB + D\bar{A} + \bar{C}\bar{A} + B\bar{A} \end{aligned} \quad (2.32)$$

L'impliquant premier  $D\bar{A}$  est inclus dans la réunion des deux impliquants  $(DC + \bar{C}\bar{A})$ ; par ailleurs les impliquants premiers  $DC$  et  $DB$  prennent la valeur 1 uniquement dans les états  $\emptyset$  de la fonction  $e$ . On va montrer qu'il est possible de supprimer ces impliquants.

### 2.3.8 Conclusion

Le théorème du paragraphe 2.3.4 affirme que la recherche des impliquants premiers de la borne supérieure  $Z_M$  est une condition nécessaire pour simplifier la fonction incomplètement définie  $Z$ . L'exemple du paragraphe 2.3.7 suggère que cette condition n'est pas suffisante : le choix d'un nombre minimal d'impliquants premiers de  $Z_M$  doit faire l'objet d'une seconde méthode analogue à celle proposée pour les fonctions complètement définies (§ 2.2.20).

### 2.3.9 Méthode

On recherche le plus petit sous-ensemble d'impliquants premiers contenant l'ensemble des mintermes de la fonction  $Z$ ; ce sous-ensemble sera minimal si l'on couvre la borne inférieure  $Z_m$  de  $Z$  :  $Z_m$  est en effet la solution de  $Z$  comportant le nombre minimal de mintermes.

En reprenant l'exemple du paragraphe 2.3.7 (fig. 2.55) on calcule tout d'abord la borne inférieure  $e_m$  de la fonction  $e$  :

$$e_m(D, C, B, A) = \Sigma 0, 2, 6, 8 \quad (2.33)$$

A l'aide des cinq impliquants premiers de  $e_M$  (2.32) on dresse la table des impliquants de  $e_m$  (fig. 2.56) dont il découle une proposition logique  $P$  :

$$P = 0 \cdot 2 \cdot 6 \cdot 8 = V \cdot (V + W) \cdot W \cdot (V + X) = V \cdot W \quad (2.34)$$

L'expression simplifiée de  $e$  est unique : c'est une somme de deux impliquants premiers essentiels :

$$e = V + W = \bar{C}\bar{A} + B\bar{A} \quad (2.35)$$

La suppression des impliquants  $D\bar{A}$ ,  $DC$  et  $DB$ , suggérée par l'examen visuel de la table de Karnaugh, est donc démontrée.

No	$\bar{C}\bar{A} = V$	$B\bar{A} = W$	$D\bar{A} = X$	$DB = Y$	$DC = Z$
0	1*				
2	1	1			
6		1*			
8	1		1		

Fig. 2.56

### 2.3.10 Exemple

Avec un certain entraînement il est possible de détecter directement dans la table de Karnaugh un sous-ensemble minimal d'impliquants premiers couvrant la borne inférieure [18] (pp. 60 - 65), [10] (pp. 129 - 132). A titre d'exemples, on donne les simplifications des six autres fonctions de l'affichage étudié (fig. 2.55); dans deux cas (fonctions  $a$  et  $g$ ) on obtient deux polynômes distincts comportant le même nombre de lettres :

$$a_1 = D + \bar{C}\bar{A} + CA + BA ; a_2 = D + \bar{C}\bar{A} + CA + \bar{C}B \quad (2.36)$$

$$b = \bar{C} + \bar{B}\bar{A} + BA \quad (2.37)$$

$$c = \bar{B} + A + C \quad (2.38)$$

$$d = \bar{C}\bar{A} + \bar{C}B + B\bar{A} + C\bar{B}A \quad (2.39)$$

$$f = D + \bar{B}\bar{A} + C\bar{B} + C\bar{A} \quad (2.40)$$

$$g_1 = D + C\bar{B} + \bar{C}B + C\bar{A} ; g_2 = D + C\bar{B} + \bar{C}B + B\bar{A} \quad (2.41)$$

Les blocs tracés dans les tables de Karnaugh de  $a$  et  $g$  (fig. 2.55) illustrent les solutions  $a_1$  et  $g_1$ .

### 2.3.11 Logigramme

A partir des expressions simplifiées (2.35) à (2.41) il est possible de tracer un logigramme NAND (fig. 2.57) réalisant les sept fonctions  $a, b, \dots, g$ . On remarque qu'un choix a été opéré pour les fonctions  $a$  et  $g$  dont l'une des deux expressions algébriques a été arbitrairement retenue ( $a_1$  et  $g_1$ ).

L'examen des relations (2.35) à (2.41) montre que plusieurs impliquants premiers (par exemple  $\bar{C}\bar{A}$ ) sont communs à plusieurs fonctions ( $a_1, d$  et  $e$ ) : cette propriété est exploitée dans le logigramme de la figure 2.57 où une seule porte NAND réalise l'expression  $\bar{C} \uparrow \bar{A}$  qui est ensuite transmise aux trois portes NAND réalisant les fonctions  $a_1, d$  et  $e$ . L'intérêt pratique de cette mise en commun des impliquants premiers est évident : il en découle une méthode de simplification des systèmes à sorties multiples qui sera exposée dans la section 2.5.

### 2.3.12 Exercice

Calculer à l'aide d'une table des impliquants premiers la proposition logique de chacune des six fonctions  $a, b, c, d, f$  et  $g$  de la figure 2.55. Quels sont les impliquants premiers des bornes supérieures de ces fonctions qui sont superflus, essentiels ou à choix ?

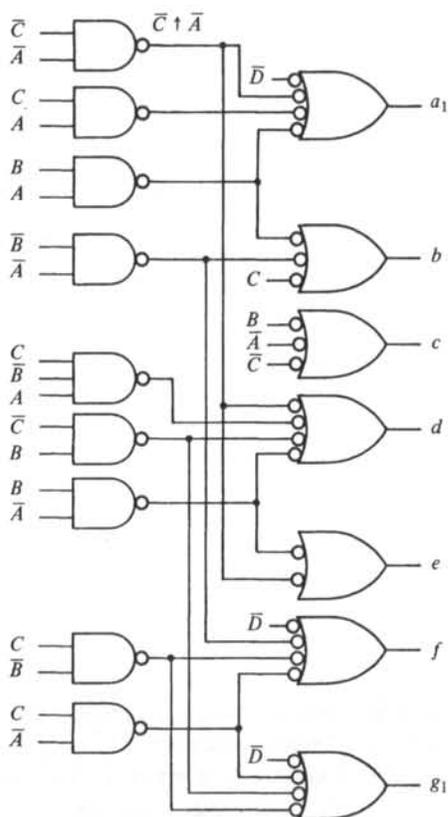


Fig. 2.57

### 2.3.13 Exercice

Simplifier à l'aide d'une table de Karnaugh et d'une table des impliquants premiers chacune des fonctions suivantes; donner l'ensemble des solutions de la proposition logique  $P$  :

- $M(a, b, c, d) = \Sigma 7, 11, 13, 14, 15 + \emptyset 3, 5, 6, 9, 10, 12$  (1.71)
- $Z_1(D, C, B, A) = \Sigma 0, 2, 5, 7, 10, 13, 15 + \emptyset 1, 4, 6, 14$
- $Z_2(D, C, B, A) = \Sigma 1, 2, 3, 6, 7, 14 + \emptyset 4, 8, 9, 11, 12, 13$
- $Z_3(D, C, B, A) = \Sigma 3, 4, 8, 15 + \emptyset 1, 2, 5, 6, 9, 10, 13, 14$

### 2.3.14 Exercice

Simplifier à l'aide d'une table de Karnaugh chacune des fonctions suivantes :

- $Z_1(D, C, B, A) = \Sigma 1, 8 + \emptyset 0, 3, 5, 6, 7, 9, 10, \dots, 15$
- $Z_2(E, D, C, B, A) = \Sigma 1, 2, 4, 5, 9, 10, 11, 15, 16, 18, 19, 22, 25, 27, 31 + \emptyset 0, 8, 13, 17, 24, 26, 29$
- $Z_3(E, D, C, B, A) = \Sigma 5, 6, 8, 11, 12, 19, 21, 22, 23, 24, 27, 28 + \emptyset 2, 10, 13, 14, 20, 29$

- $Z_4(a, b, c, d) = \Sigma 7, 8, 9, 10 + \emptyset 0, 1, 2, 13, 14, 15$
- $Z_5(a, b, c, d) = \Sigma 0, 1, 3, 6, 7, 8, 11 + \emptyset 2, 4, 10, 12, 13, 14$

Simplifier également les fonctions  $\bar{Z}_4$  et  $\bar{Z}_5$ .

### 2.3.15 Exercice

Un système combinatoire est défini par quatre fonctions  $D, C, B, A$  de quatre variables  $Q_1, Q_2, Q_3, Q_4$ . Simplifier chacune des fonctions à l'aide d'une table de Karnaugh et réaliser un logigramme NAND mettant en évidence le plus grand nombre possible d'impliquants communs à plusieurs fonctions :

$$D(Q_1, Q_2, Q_3, Q_4) = \Sigma 1, 2 + \emptyset 3, 4, 6, 9, 10, 13 ; C(Q_1, Q_2, Q_3, Q_4) = \Sigma 5, 7, 11, 15 + \emptyset 3, 4, 6, 9, 10, 13 ; B(Q_1, Q_2, Q_3, Q_4) = \Sigma 5, 11, 12, 14 + \emptyset 3, 4, 6, 9, 10, 13 ; A(Q_1, Q_2, Q_3, Q_4) = \Sigma 1, 5, 7, 8, 14 + \emptyset 3, 4, 6, 9, 10, 13$$

### 2.3.16 Exercice

Un convertisseur de codes BCD - Excédent 3 (§ 8.1.7 et 8.1.8) est défini par quatre fonctions  $d, c, b, a$  de quatre variables  $D, C, B, A$  :

$$d(D, C, B, A) = \Sigma 5, 6, \dots, 9 + \emptyset 10, 11, \dots, 15 ; c(D, C, B, A) = \Sigma 1, 2, 3, 4, 9 + \emptyset 10, 11, \dots, 15 ; b(D, C, B, A) = \Sigma 0, 3, 4, 7, 8 + \emptyset 10, 11, \dots, 15 ; a(D, C, B, A) = \Sigma 0, 2, 4, 6, 8 + \emptyset 10, 11, \dots, 15$$

Simplifier à l'aide d'une table de Karnaugh les fonctions  $d, c, b, a$  puis réaliser un logigramme NAND, OU-exclusif. On demande ensuite la synthèse du convertisseur inverse Excédent 3 - BCD : calculer chacune des expressions simplifiées de  $D, C, B, A$  en fonction des variables  $d, c, b, a$  puis tracer le logigramme NAND, OU-exclusif correspondant.

### 2.3.17 Exercice

Un affichage à sept segments absolument semblable à celui décrit au paragraphe 2.3.1 (fig. 2.51, 2.52 et 2.53) est caractérisé par une convention inverse : chaque segment lumineux est allumé pour la valeur 0 de la sortie et éteint pour la valeur 1. On demande de simplifier les sept nouvelles fonctions  $\bar{a}, \bar{b}, \dots, \bar{g}$  à l'aide d'une table de Karnaugh puis de réaliser un logigramme NAND en mettant en évidence le plus grand nombre d'impliquants communs à plusieurs fonctions.

## 2.4 ANALYSE DES SYSTÈMES COMBINATOIRES

### 2.4.1 Rappel

L'analyse est le calcul de la fonction logique réalisée par un logigramme donné (§ 2.1.1).

### 2.4.2 Exemple

Le logigramme de la figure 2.57 réalise sept fonctions  $a, b, \dots, g$  définies par les tables de Karnaugh de la figure 2.55. Le cahier des charges du système étudié étant incomplet, on ne connaît donc pas le comportement de son logigramme dans les six

états  $\emptyset$  ( $D, C, B, A = 1010, \dots, 1111$ ). Il peut être souhaitable de déterminer ce comportement lorsqu'on est amené (par accident ou pour des raisons de test) dans l'un de ces états  $\emptyset$  : le calcul correspondant est une analyse.

### 2.4.3 Analyse algébrique

L'établissement d'une table de vérité (fig. 2.58) comportant autant de lignes que d'états  $\emptyset$  (six) permet de calculer chacune des sept fonctions  $a_1, b, c, d, e, f$  et  $g_1$  [(2.35) à (2.41)] réalisées par le logigramme de la figure 2.57. Les valeurs 0 ou 1 de ces fonctions entraînent l'extinction ou l'allumage des segments lumineux correspondants : on obtient ainsi les configurations de l'affichage dans les six états  $\emptyset$  (fig. 2.59).

No	D	C	B	A	$\bar{C}\bar{A}$	CA	BA	$\bar{B}\bar{A}$	$\bar{C}B$	$B\bar{A}$	$\bar{C}\bar{B}\bar{A}$	$\bar{C}\bar{B}$	$\bar{C}\bar{A}$	$a_1$	b	c	d	e	f	$g_1$
10	1	0	1	0	1	0	0	0	1	1	0	0	0	1	1	0	1	1	1	1
11	1	0	1	1	0	0	1	0	1	0	0	0	0	1	1	1	1	0	1	1
12	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	1	1
13	1	1	0	1	0	1	0	0	0	0	1	1	0	1	0	1	1	0	1	1
14	1	1	1	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	1
15	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1	1

Fig. 2.58

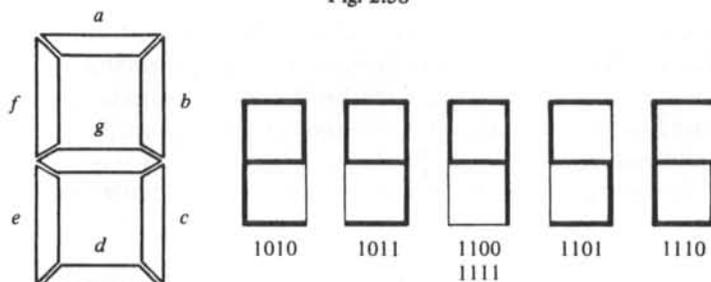


Fig. 2.59

Le calcul algébrique peut être simplifié si l'on remarque que les six états  $\emptyset$  sont caractérisés par la relation (fig. 2.58) :

$$D = 1 \quad (2.42)$$

En remplaçant (2.42) dans les expressions (2.36), (2.40) et (2.41) on calcule directement :

$$a_1 = f = g_1 = 1 \quad (2.43)$$

### 2.4.4 Analyse graphique

Tout polynôme peut être représenté dans une table de Karnaugh (§ 2.2.11). Chacune des expressions algébriques (2.35) à (2.41) est illustrée par une table de Karnaugh de la figure 2.55; chacun des symboles  $\emptyset$  de ces tables prend la valeur 1 si la case corres-

pondante appartient à un impliquant premier de la fonction, et prend la valeur 0 dans le cas contraire.

Examinons par exemple le cas de la fonction  $e$  (2.35) :

$$e = \bar{C}\bar{A} + B\bar{A}$$

Les deux impliquants premiers de  $e$  couvrent respectivement les cases 0, 2, 8, 10 et 2, 6, 10, 14 (fig. 2.55). La fonction  $e$  prend donc la valeur 1 dans les états  $\emptyset$  recouverts par un impliquant au moins (No 10 et 14) et prend la valeur 0 dans les états  $\emptyset$  restants (No 11, 12, 13 et 15). Les résultats obtenus pour les autres fonctions vérifient le calcul effectué dans la table de vérité (fig. 2.58); dans de nombreux cas, l'utilisation de la table de Karnaugh s'avère plus commode que la procédure algébrique.

### 2.4.5 Conclusion

L'opération d'analyse s'applique non seulement à des systèmes dont le cahier des charges est incomplet mais également à ceux dont le comportement est complètement inconnu et qui sont donnés par leur seul logigramme. Dans le cas où la traduction algébrique du logigramme sous la forme d'un polynôme est malaisée (utilisation d'opérateurs NAND, NOR, OU-exclusif, etc...) on peut effectuer au préalable les transformations graphiques citées au paragraphe 1.4.19. Dans tous les cas, l'analyse permet de calculer systématiquement la fonction réalisée par un logigramme donné : cette fonction est unique.

L'opération de synthèse est inverse; partant de la représentation unique d'une fonction dans une table de vérité, un diagramme de Venn ou une table de Karnaugh, on a mis en évidence l'existence d'un grand nombre d'expressions algébriques équivalentes réalisant cette fonction : la forme canonique (qui est unique), des polynômes (parmi lesquels se trouvent les formes simplifiées), etc... Chacune de ces expressions peut à son tour être réalisée par plusieurs logigrammes fonctionnellement équivalents. La synthèse implique donc le choix délicat d'un bon chemin parmi la multitude des voies conduisant de la fonction aux logigrammes qui la réalisent.

### 2.4.6 Exercice

On donne la fonction incomplètement définie :

$$Z(D, C, B, A) = \Sigma 0, 3, 5, 6, 10, 14 + \emptyset 2, 4, 7, 8, 12, 13 \quad (2.44)$$

Démontrer à l'aide des propriétés de la section 1.7 que l'inverse de l'expression simplifiée d'une fonction incomplètement définie ( $Z$ ) n'est pas forcément égal à l'expression simplifiée de la fonction inverse ( $\bar{Z}$ ). Vérifier cette affirmation sur l'exemple de la fonction (2.44) et l'illustrer dans une table de Karnaugh. Application au cas de la fonction  $M(a, b, c, d) = \Sigma 7, 11, 13, 14, 15 + \emptyset 3, 5, 6, 9, 10, 12$  (1.71).

### 2.4.7 Exercice

Démontrer à l'aide des tables de Karnaugh de la figure 2.55 qu'un logigramme réalisant les expressions  $a_2$  (2.36) et  $g_2$  (2.41) serait fonctionnellement équivalent au logi-

gramme de la figure 2.57 dans les six états  $\emptyset$ . Dédire de cette constatation que  $a_1$  et  $a_2$  ( $g_1$  et  $g_2$ ) sont deux expressions équivalentes d'une même solution de la fonction incomplètement définie  $a$  ( $g$ ).

#### 2.4.8 Exercice

Analyser le logigramme calculé dans l'exercice du paragraphe 2.3.17 et déterminer les configurations de l'affichage dans les six états  $\emptyset$ .

#### 2.4.9 Exercice

Analyser le logigramme de la figure 2.60 à l'aide d'une table de Karnaugh et montrer que les sept sorties  $a, b, \dots, g$  peuvent attaquer un affichage à sept segments (fig. 2.52) pour représenter les chiffres 0, 1, ..., 9 tandis que les quatre sorties  $w, v, y, x$  effectuent un codage particulier des quatre entrées  $D, C, B, A$ .

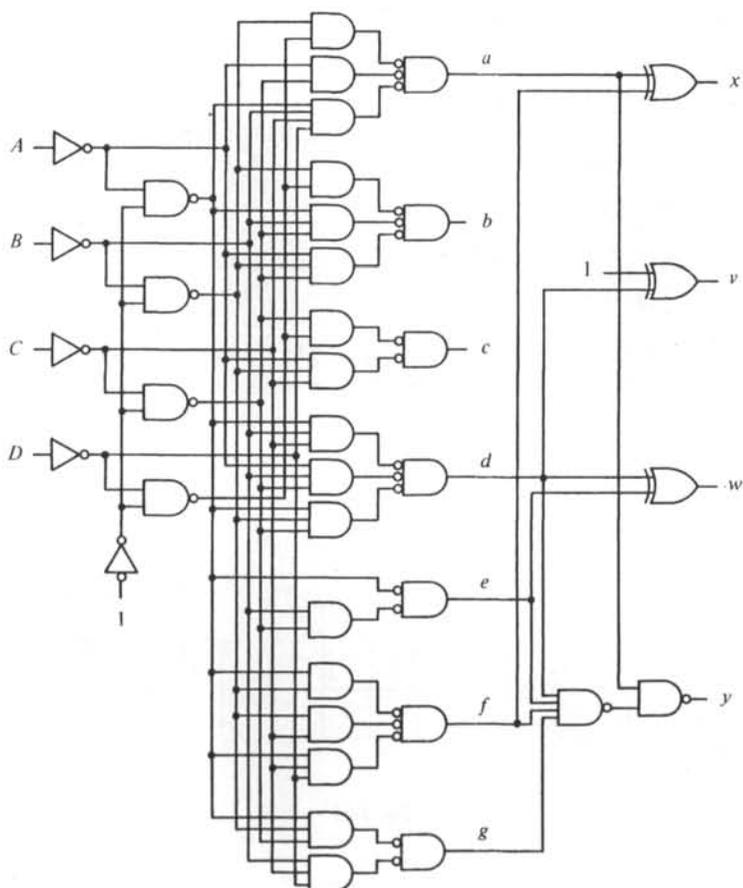


Fig. 2.60

## 2.5 FONCTIONS MULTIPLES

## 2.5.1 Introduction

On a déjà relevé la possibilité de réaliser des impliquants premiers communs à plusieurs fonctions d'un même logigramme (§ 2.3.11). Jusqu'ici on a toujours considéré les systèmes combinatoires à sorties multiples comme autant de systèmes à sortie unique et le calcul s'est borné à simplifier chaque fonction indépendamment de ses voisines. Dans cette section on étudie la simplification simultanée de plusieurs fonctions dépendant des mêmes variables afin d'obtenir un nombre maximum d'impliquants premiers ou de monômes communs.

## 2.5.2 Exemple

On cherche à réaliser un système à sorties multiples décrit par trois fonctions incomplètement définies :

$$R(D, C, B, A) = \Sigma 0, 4, 11, 15 + \emptyset 1, 3, 5, 8, 10, 14 \quad (2.45)$$

$$S(D, C, B, A) = \Sigma 0, 4, 5, 13 + \emptyset 2, 3, 6, 7, 9, 14, 15 \quad (2.46)$$

$$T(D, C, B, A) = \Sigma 5, 11, 13, 15 + \emptyset 3, 4, 9, 12 \quad (2.47)$$

En simplifiant chacune de ces fonctions dans une table de Karnaugh (fig. 2.61) on obtient les expressions suivantes :

$$R = \overline{D}\overline{B} + DB \quad (2.48)$$

$$S = \overline{D}\overline{A} + CA \quad (2.49)$$

$$T = C\overline{B} + DA \quad (2.50)$$

Chacun des monômes de  $R$ ,  $S$  et  $T$  est un impliquant premier essentiel; aucun impliquant n'est commun à plusieurs fonctions. Le logigramme réalisant les expressions (2.48) à (2.50) comporte neuf portes NAND à deux entrées chacune (fig. 2.63).

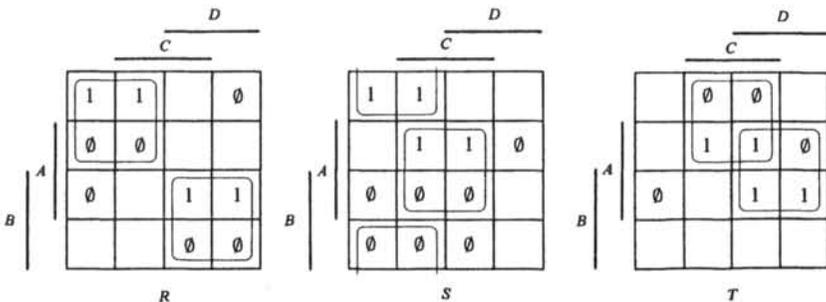


Fig. 2.61

## 2.5.3 Simplification simultanée

Les tables de Karnaugh de la figure 2.62 illustrent une simplification partielle des trois fonctions  $R$ ,  $S$  et  $T$  :

$$R = \overline{D}\overline{B}\overline{A} + DBA \quad (2.51)$$

$$S = \overline{D}\overline{B}\overline{A} + C\overline{B}A \quad (2.52)$$

$$T = DBA + C\overline{B}A \quad (2.53)$$

Aucun monôme de  $R$ ,  $S$  et  $T$  n'est un impliquant premier; par contre chacun des trois monômes  $\overline{D}\overline{B}\overline{A}$ ,  $DBA$  et  $C\overline{B}A$  est commun à deux fonctions. Le logigramme réalisant les expressions (2.51) à (2.53) comporte six portes NAND (fig. 2.64), dont trois à deux entrées et trois à trois entrées. Cette réalisation est moins complexe que la précédente (fig. 2.63) et justifie la mise en oeuvre de méthodes particulières.

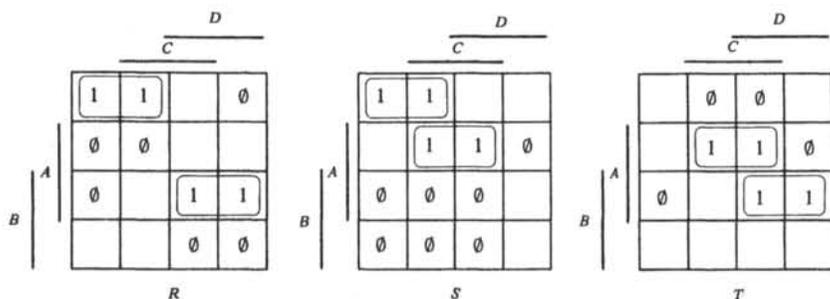


Fig. 2.62

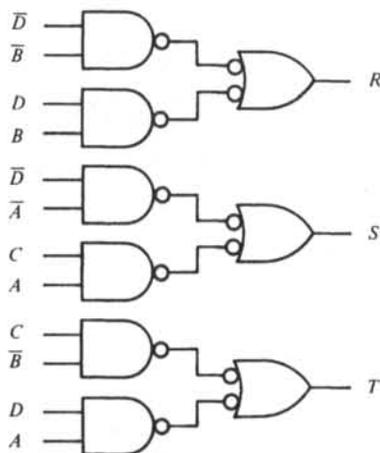


Fig. 2.63

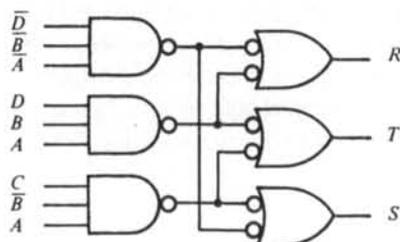


Fig. 2.64

### 2.5.4 Méthode

La recherche de monômes communs à plusieurs fonctions peut se ramener au calcul des impliquants premiers du produit de ces fonctions. Pour effectuer la simplification simultanée de  $r$  fonctions, il faut alors réaliser tous les produits possibles de ces fonctions deux à deux, trois à trois, etc... L'ensemble des impliquants premiers des  $r$  fonctions et de tous les produits qui en découlent est déterminé à l'aide des tables de Karnaugh; une table des impliquants premiers permet enfin le choix d'un sous-ensemble minimal d'impliquants couvrant les  $r$  fonctions données [10] (pp. 161 - 166).

### 2.5.5 Exemple

A partir des formes canoniques (2.45) à (2.47) on détermine tout d'abord les bornes supérieures de  $R$ ,  $S$  et  $T$  conformément à la méthode de simplification des fonctions incomplètement définies (§ 2.3.6) :

$$R_M(D, C, B, A) = \Sigma 0, 1, 3, 4, 5, 8, 10, 11, 14, 15 \quad (2.54)$$

$$S_M(D, C, B, A) = \Sigma 0, 2, 3, 4, 5, 6, 7, 9, 13, 14, 15 \quad (2.55)$$

$$T_M(D, C, B, A) = \Sigma 3, 4, 5, 9, 11, 12, 13, 15 \quad (2.56)$$

On détermine tous les produits de  $R_M$ ,  $S_M$  et  $T_M$  effectués deux à deux puis trois à trois :

$$R_M \cdot S_M = V(D, C, B, A) = \Sigma 0, 3, 4, 5, 14, 15 \quad (2.57)$$

$$R_M \cdot T_M = W(D, C, B, A) = \Sigma 3, 4, 5, 11, 15 \quad (2.58)$$

$$S_M \cdot T_M = X(D, C, B, A) = \Sigma 3, 4, 5, 9, 13, 15 \quad (2.59)$$

$$R_M \cdot S_M \cdot T_M = Y(D, C, B, A) = \Sigma 3, 4, 5, 15 \quad (2.60)$$

La simplification des sept expressions (2.54) à (2.60) peut être entreprise dans sept tables de Karnaugh; le choix d'un sous-ensemble minimal d'impliquants couvrant les bornes inférieures de  $R$ ,  $S$  et  $T$  est effectué à l'aide d'une table des impliquants premiers. Le calcul détaillé fait l'objet d'un exercice (§ 2.5.7).

### 2.5.6 Conclusion

Le calcul des impliquants premiers de  $r$  fonctions et de tous leurs produits nécessite la simplification de  $2^r - 1$  expressions; pour  $r \geq 4$ , on constate que cette méthode est fastidieuse. Dans la pratique, on peut toujours rechercher dans les tables de Karnaugh de plusieurs fonctions quelques monômes communs et se contenter ainsi d'une simplification approchée.

Une méthode tabulaire dérivée de l'algorithme de McCluskey [17] a été proposée par Bartee [27] : cette procédure est systématique et peut théoriquement s'appliquer à des systèmes comportant un nombre quelconque d'entrées et de sorties; la complexité de cette méthode la prédispose au calcul automatique : on trouvera dans la section 8.2 le traitement complet de l'exemple défini au paragraphe 2.5.2.

### 2.5.7 Exercice

Effectuer les calculs du paragraphe 2.5.5 puis comparer les résultats avec ceux découlant de la méthode systématique de la section 8.2.

### 2.5.8 Exercice

Un système à sorties multiples est défini par les trois fonctions :

$$Z_1(a, b, c, d) = \Sigma 2, 4, 10, 11, 12, 13$$

$$Z_2(a, b, c, d) = \Sigma 4, 5, 10, 11, 13$$

$$Z_3(a, b, c, d) = \Sigma 1, 2, 3, 10, 11, 12$$

Déterminer à l'aide des tables de Karnaugh l'ensemble des impliquants premiers de ces fonctions et de leurs produits  $Z_1 \cdot Z_2$ ,  $Z_1 \cdot Z_3$ ,  $Z_2 \cdot Z_3$  et  $Z_1 \cdot Z_2 \cdot Z_3$ ; calculer dans une table des impliquants premiers un sous-ensemble minimal de ceux-ci [10] (pp. 161-166).

### 2.5.9 Exercice

Simplifier simultanément les fonctions du comparateur calculé au paragraphe 2.2.27 en recherchant les impliquants premiers de ces fonctions et de leur produit; déterminer un sous-ensemble minimal de ces impliquants.

### 2.5.10 Exercice

Rechercher dans les tables de Karnaugh le plus grand nombre de monômes communs aux fonctions du décodeur calculé au paragraphe 2.3.15 et de l'affichage à sept segments représenté par la figure 2.55.

## 2.6 OPÉRATEURS OU-EXCLUSIF

### 2.6.1 Introduction

La transformation d'une forme canonique en une expression {NON, ET, OU, OU-exclusif} est motivée par les deux considérations suivantes : d'une part l'opérateur OU-exclusif est d'un emploi fréquent; d'autre part, on constate qu'une fonction logique dont la simplification n'est pas possible (ou produit un polynôme compliqué) peut généralement être réalisée par une expression algébrique simple comportant une ou plusieurs fonctions OU-exclusif. Dans cette section, on montre qu'il est possible d'utiliser la table de Karnaugh pour effectuer ce type de transformation.

### 2.6.2 Mode de représentation

Représentons dans deux tables de Karnaugh les fonctions  $a + b$  (fig. 2.65) et  $a \oplus b = \bar{a}b + a\bar{b}$  (fig. 2.66). On constate que la forme canonique  $\bar{a}b + a\bar{b}$  de la fonction OU-exclusif est une somme d'impliquants premiers qui ne peut pas être simplifiée.

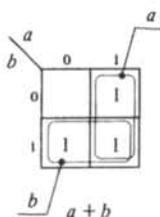


Fig. 2.65

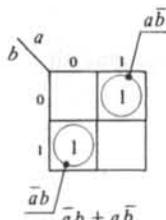


Fig. 2.66

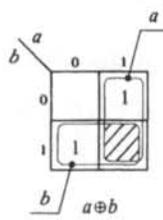


Fig. 2.67

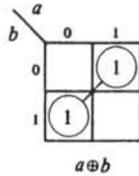


Fig. 2.68

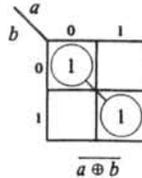


Fig. 2.69

### 2.6.3 Première propriété

Par définition, la fonction  $a \oplus b$  est égale à la fonction  $a + b$  pour tout état  $a, b$  à l'exception de  $a, b = 11$  (§ 1.6.1); il en découle la représentation de la figure 2.67 où la fonction  $a \oplus b$  prend la valeur 1 dans toute la partie de la table représentant  $a + b$  à l'exception de la case  $a, b = 11$  qui est hachurée et dans laquelle  $a \oplus b = 0$ .

Par définition, on a également vu que la fonction OU-exclusif de  $n$  variables vaut 0 lorsqu'un nombre pair de variables prennent la valeur 1, et vaut 1 dans le cas contraire (§ 1.6.5). Considérons par exemple le cas où  $n = 3$ ; la fonction  $a \oplus b \oplus c$  peut être représentée dans une table de Karnaugh (fig. 2.70) en partant de la définition. L'examen de cette table met en évidence la propriété suivante : *toute fonction OU-exclusif ( $a \oplus b \oplus c$  par exemple) peut être représentée dans une table de Karnaugh par la fonction OU correspondante ( $a + b + c$ ) à condition de supprimer les 1 de celle-ci dans tout état comportant un nombre pair de variables égales à 1 ( $a, b, c = 011, 101$  et  $110$ )*. Cette suppression des 1 peut être assimilée à une soustraction logique.

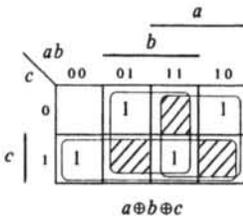


Fig. 2.70

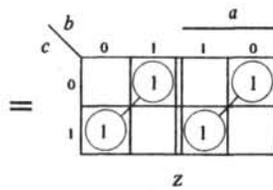


Fig. 2.71

### 2.6.4 Exemple

On cherche à réaliser les fonctions  $Z_1$  et  $Z_2$  (fig. 2.72 et 2.73) à l'aide d'opérateurs OU-exclusif. Une solution possible est :

$$Z_1 = C \oplus O \bar{R} \quad (2.61)$$

$$Z_2 = E \oplus \bar{C} \quad (2.62)$$

La simplification des mêmes fonctions donnerait :

$$Z_1 = CR + \bar{O}C + O\bar{C}\bar{R} \quad (2.63)$$

$$Z_2 = \bar{E}\bar{C} + EC \quad (2.64)$$

On constate que la transformation algébrique de (2.64) en (2.62) est immédiate (1.55) tandis que celle de (2.63) en (2.61) n'est pas évidente.

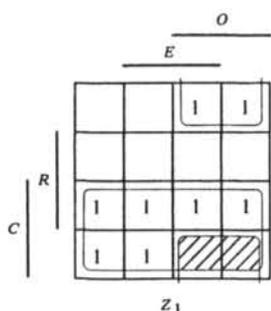


Fig. 2.72

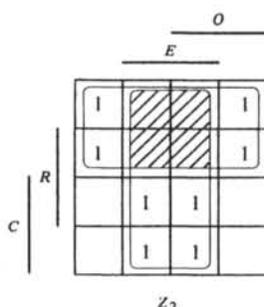


Fig. 2.73

### 2.6.5 Exemple

On désire réaliser simultanément les trois fonctions  $Z$ ,  $D_1$  et  $D_2$  (fig. 2.74 à 2.76) à l'aide d'opérateurs OU-exclusif. En simplifiant  $Z$  et en utilisant ses quatre impliquants premiers pour calculer  $D_1$  et  $D_2$  on obtient :

$$Z = x_1 \bar{x}_2 \bar{y}_1 + \bar{x}_1 x_2 y_1 + x_1 x_2 \bar{y}_2 + \bar{x}_1 \bar{x}_2 y_2 \quad (2.65)$$

$$D_1 = (x_1 \oplus x_1 \bar{x}_2 \bar{y}_1) + \bar{x}_1 x_2 y_1 \quad (2.66)$$

$$D_2 = (x_2 \oplus x_1 x_2 \bar{y}_2) + \bar{x}_1 \bar{x}_2 y_2 \quad (2.67)$$

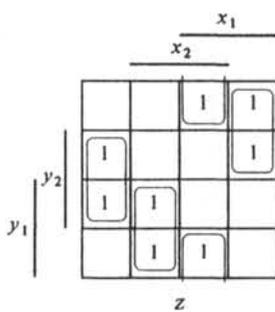


Fig. 2.74

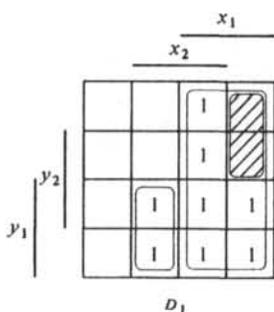


Fig. 2.75

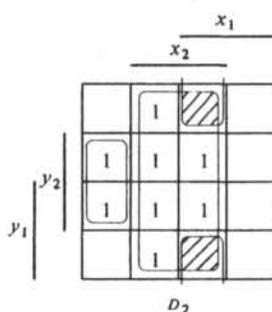


Fig. 2.76

### 2.6.6 Seconde propriété

L'examen des figures 2.68 et 2.71 suggère une seconde propriété : *toute fonction OU-exclusif peut être détectée dans une table de Karnaugh par une ou plusieurs paires de cases valant 1 et situées en diagonale*; on dit aussi qu'une paire de cases en diagonale est un *damier* et on relie ces cases par un trait oblique.

### 2.6.7 Exemple

En partant du damier défini par la fonction  $a \oplus b$  (fig. 2.68) et du damier complémentaire représentant  $a \oplus \bar{b}$  (fig. 2.69) on peut retrouver les expressions algébriques de fonctions OU-exclusif quelconques; l'examen de la figure 2.71 permet d'écrire :

- pour  $a = 0$  :  $Z = b \oplus c$
- pour  $a = 1$  :  $Z = \bar{b} \oplus c$

Il en découle la forme finale de  $Z$  :

$$Z = \bar{a}(b \oplus c) + a(\overline{b \oplus c}) = a \oplus b \oplus c \quad (2.68)$$

### 2.6.8 Exemple

On cherche à transformer la forme canonique de la fonction  $E(a, b, c, d)$  définie par la figure 1.66 et représentée par une table de Karnaugh (fig. 2.77). Pour mieux mettre en évidence les damiers recherchés, on décompose  $E$  en deux fonctions partielles  $E_1$  et  $E_2$  avec :

$$E = E_1 + E_2 \quad (2.69)$$

	$ab$			
$cd$	00	01	11	10
00			1	
01		1		1
11	1			
10		1		1

$E$

Fig. 2.77

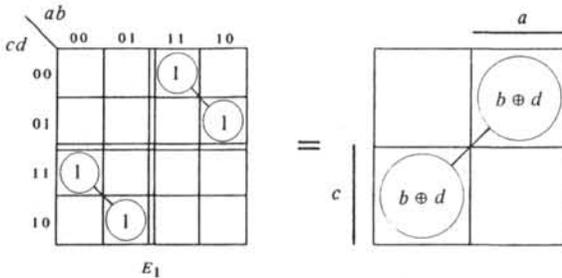


Fig. 2.78

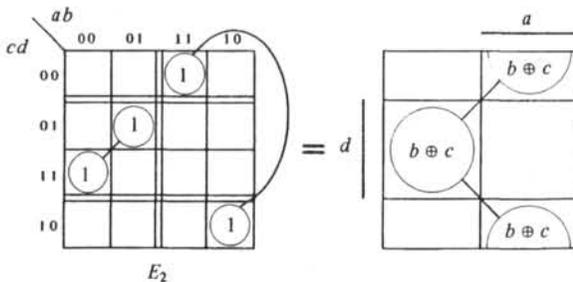


Fig. 2.79

Dans chacune des tables de Karnaugh de  $E_1$  et  $E_2$  (fig. 2.78 et 2.79) on fait apparaître deux damiers; chacun d'eux est exprimé par une fonction OU-exclusif ( $b \oplus d$  et  $b \oplus c$ ) qui est elle-même incluse dans un damier plus grand ( $a \oplus c$  et  $a \oplus d$ ). On remarque dans la figure 2.79 la représentation particulière des damiers définis par les lignes extrêmes de la table de Karnaugh. Les expressions algébriques de  $E_1$  et  $E_2$  sont alors :

$$E_1 = a\bar{c}(b \oplus d) + \bar{a}c(b \oplus d) = (a \oplus c)(b \oplus d) \quad (2.70)$$

$$E_2 = a\bar{d}(b \oplus c) + \bar{a}d(b \oplus c) = (a \oplus d)(b \oplus c) \quad (2.71)$$

En remplaçant  $E_1$  et  $E_2$  dans (2.69) on a enfin :

$$E = (a \oplus c)(b \oplus d) + (a \oplus d)(b \oplus c) \quad (2.72)$$

### 2.6.9 Conclusion

Il est donc possible de mettre en évidence la fonction OU-exclusif dans une table de Karnaugh grâce à deux propriétés (soustraction logique et représentation par des damiers); la méthode proposée est plus un tour de main [28] (pp. 89 - 96), [29] (pp. 97 - 98) qu'une procédure systématique [30]. L'emploi de la table de Karnaugh n'exclut ni le calcul algébrique, ni l'utilisation d'un système universel réalisant une fonction du type  $Z = y_0 \oplus y_1 \cdot a \oplus y_2 \cdot b \oplus y_3 \cdot ab$  (§ 1.6.13). Enfin l'équivalence des relations  $a = b$  et  $a \oplus b = 0$  (§ 1.6.8) suggère une méthode algébrique particulière exposée dans la référence [96].

### 2.6.10 Exercice

Simplifier les fonctions  $D_1$  et  $D_2$  des figures 2.75 et 2.76 puis comparer le logigramme NAND réalisant les expressions simplifiées de  $Z$  (fig. 2.74),  $D_1$  et  $D_2$  avec le logigramme {NAND, OU-exclusif} réalisant les relations (2.65) à (2.67). Effectuer une comparaison similaire pour les exemples des paragraphes 2.6.4 et 2.6.8.

### 2.6.11 Exercice

Un système transformant le code Gray en code binaire pur est défini par les quatre fonctions suivantes :

- $Z_3(D, C, B, A) = \Sigma 8, 9, \dots, 15$
- $Z_2(D, C, B, A) = \Sigma 4, 5, \dots, 11$
- $Z_1(D, C, B, A) = \Sigma 2, 3, 4, 5, 8, 9, 14, 15$
- $Z_0(D, C, B, A) = \Sigma 1, 2, 4, 7, 8, 11, 13, 14$

Réaliser un logigramme {NAND, OU-exclusif} puis effectuer la synthèse d'un système opérant la conversion inverse à l'aide d'opérateurs OU-exclusif.

### 2.6.12 Exercice

Réaliser les sept fonctions  $a, b, \dots, g$  (fig. 2.55) de l'affichage à sept segments avec le plus grand nombre d'opérateurs OU-exclusif; tracer le logigramme {NAND, OU-exclusif} et comparer sa complexité à celle du logigramme NAND (fig. 2.57).

### 2.6.13 Exercice

Réaliser les fonctions suivantes avec des opérateurs OU-exclusif :

- $\bar{E}(a, b, c, d) = \Sigma 0, 1, 2, 4, 7, 8, 11, 13, 14, 15$  (fig. 1.66)
- $Z_1(a, b, c, d) = \Sigma 3, 4, 8, 15 + \emptyset 1, 2, 5, 6, 9, 10, 13, 14$
- $Z_2(a, b, c, d) = \Sigma 2, 7, 8, 11, 13, 14$

Effectuer un calcul semblable pour  $\bar{Z}_1$  et  $\bar{Z}_2$ .

### 2.6.14 Exercice

Effectuer la synthèse d'un système à quatre entrées  $D, C, B, A$  et deux sorties  $Z_1, Z_2$ ; la sortie  $Z_1$  prend la valeur 1 chaque fois que le numéro décimal  $N$  de l'état  $D, C, B, A$  est entièrement divisible par 3, 5 ou 7 tandis que  $Z_2$  vaut 1 chaque fois que  $N$  est divisible entièrement par 4 ou 5. Tracer un logigramme {NAND, OU-exclusif} réalisant les fonctions  $Z_1$  et  $Z_2$ .

## 2.7 SYSTÈMES ITÉRATIFS

### 2.7.1 Introduction

L'emploi de la table de Karnaugh est limité à un petit nombre de variables (environ six); dans certaines applications, en particulier dans les calculatrices digitales, on représente les nombres par une succession de bits (en général 8, 16 ou 32). Lorsqu'on conçoit des systèmes combinatoires réalisant l'addition, la soustraction ou la comparaison de deux tels nombres [29] (pp. 220 - 263) on obtient des fonctions logiques de 16, 32 ou 64 variables. On est alors amené à décomposer le système donné en plusieurs sous-systèmes; chacun de ceux-ci pourra être calculé par les méthodes habituelles, en particulier par la table de Karnaugh.

### 2.7.2 Exemple

Deux nombres  $A$  et  $B$  sont exprimés dans le système binaire à l'aide de deux bits chacun ( $A_1, A_0$  et  $B_1, B_0$ );  $A$  et  $B$  sont définis par les relations arithmétiques habituelles (sect. 8.1) :

$$A = A_1 \cdot 2^1 + A_0 \cdot 2^0 ; B = B_1 \cdot 2^1 + B_0 \cdot 2^0 \quad (2.73)$$

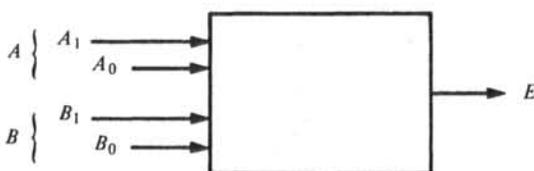


Fig. 2.80

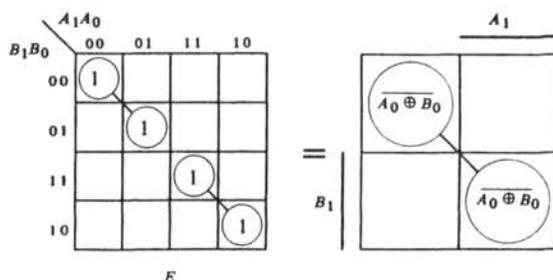


Fig. 2.81

On cherche à réaliser un système combinatoire (fig. 2.80) susceptible de détecter l'égalité ( $E = 1$ ) de ces deux nombres. Les deux tables de Karnaugh de la fonction  $E$  (fig. 2.81), dans lesquelles on a mis en évidence la fonction OU-exclusif, permettent d'écrire l'expression algébrique suivante :

$$E = (\overline{A_0 \oplus B_0}) (\overline{A_1 B_1} + A_1 B_1) = \overline{A_0 \oplus B_0} \cdot \overline{A_1 \oplus B_1} \quad (2.74)$$

Le logigramme de la figure 2.82 réalise l'expression (2.74).

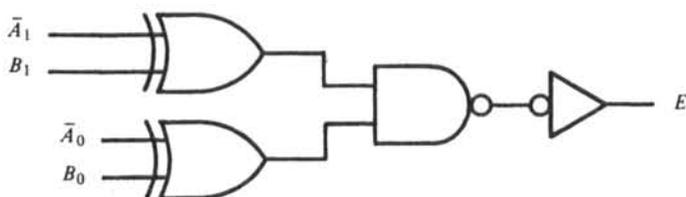


Fig. 2.82

### 2.7.3 Conclusion

La conception d'un système comparant deux nombres de trois bits chacun peut être entreprise selon la méthode traditionnelle du paragraphe 2.7.2 et entraîne l'emploi d'une table de Karnaugh à six variables; pour  $K$  bits, on est conduit à simplifier une fonction logique de  $2K$  variables. Dans l'exemple du paragraphe suivant on va tenter de décomposer un système comparant deux nombres  $A$  et  $B$  de  $n + 1$  bits chacun.

### 2.7.4 Exemple

Deux nombres  $A$  et  $B$  sont représentés dans le système binaire à l'aide de  $n + 1$  bits chacun et peuvent être définis par les relations arithmétiques suivantes (sect. 8.1) :

$$A = A_n \cdot 2^n + A_{n-1} \cdot 2^{n-1} + \dots + A_2 \cdot 2^2 + A_1 \cdot 2^1 + A_0 \cdot 2^0 \quad (2.75)$$

$$B = B_n \cdot 2^n + B_{n-1} \cdot 2^{n-1} + \dots + B_2 \cdot 2^2 + B_1 \cdot 2^1 + B_0 \cdot 2^0 \quad (2.76)$$

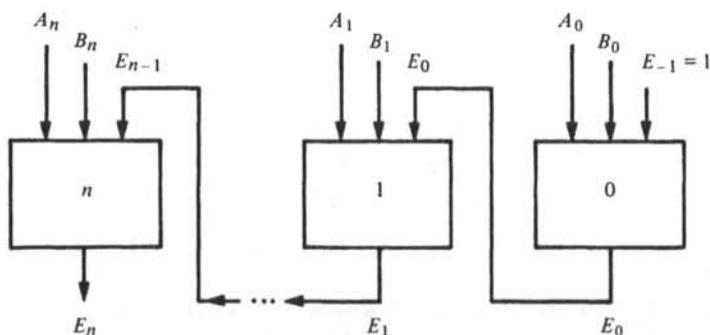


Fig. 2.83

On cherche à décomposer le système en  $n + 1$  sous-systèmes identiques selon le schéma de la figure 2.83. Chaque sous-système comporte trois variables ( $A_n, B_n, E_{n-1}$ ) et une sortie ( $E_n$ ) définie par le cahier des charges suivant :

- si  $E_{n-1} = 0$ , l'inégalité de  $A$  et  $B$  a déjà été détectée par un sous-système précédent; il en résulte que  $E_n = 0$ ;
- si  $E_{n-1} = 1$ , l'égalité de  $A$  et  $B$  est vérifiée par tous les sous-systèmes précédents; ce sont les valeurs de  $A_n$  et  $B_n$  qui déterminent celles de  $E_n$  ( $E_n = 1$  pour  $A_n = B_n$ ;  $E_n = 0$  pour  $A_n \neq B_n$ ).

La représentation de  $E_n$  dans une table de Karnaugh (fig. 2.84) donne une forme canonique réalisée par le logigramme NAND de la figure 2.85 :

$$E_n = \bar{A}_n \bar{B}_n E_{n-1} + A_n B_n E_{n-1} = E_{n-1} (\bar{A}_n \oplus B_n) \quad (2.77)$$

Chacun des  $n$  sous-systèmes réalisant les fonctions  $E_0, E_1, \dots, E_{n-1}$  peut être réalisé par un logigramme équivalent à celui de la figure 2.85; on remarque qu'il faut imposer la valeur  $E_{-1} = 1$  pour tenir compte des exigences du cahier des charges. Le schéma de la figure 2.83 comporte  $n + 1$  niveaux (§ 2.1.15); en remplaçant dans cette figure chaque sous-système par son logigramme NAND (fig. 2.85) on obtiendrait pour le chemin le plus long entre une entrée ( $A_0$  ou  $B_0$ ) et une sortie ( $E_n$ ) une profondeur égale à  $2n + 3$ .

$A_n B_n$	00	01	11	10
$E_{n-1}$				
0				
1	1		1	
	$E_n$			

Fig. 2.84

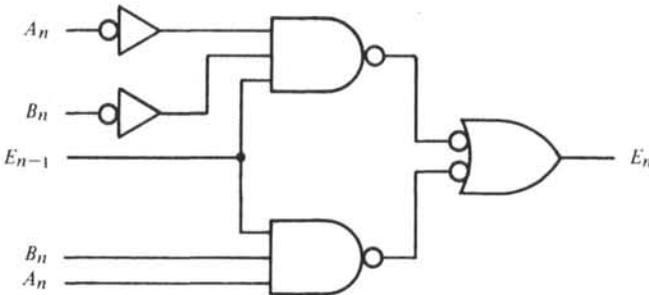


Fig. 2.85

### 2.7.5 Définitions

On appelle *système combinatoire itératif* tout système combinatoire décomposé en plusieurs sous-systèmes identiques entre eux, c'est-à-dire fonctionnellement équivalents. Le comparateur du paragraphe 2.7.4 (fig. 2.83) est donc un système itératif.

Lorsqu'un système donné admet une telle décomposition (par exemple le comparateur de deux nombres), on appelle généralement *structure parallèle* la réalisation de ce système sans décomposition (fig. 2.80 et 2.82) et *structure itérative* ou *pseudo-parallèle* la réalisation décomposée (fig. 2.83 et 2.85).

### 2.7.6 Conclusion

La synthèse d'un système itératif nécessite une décomposition en sous-systèmes identiques : celle-ci n'est possible que pour certains cahiers des charges. La recherche d'une bonne décomposition est une démarche essentiellement empirique.

La structure itérative d'un système admettant une décomposition en  $n$  sous-systèmes abaisse considérablement la complexité du problème combinatoire initial (dans un rapport de  $n$  à 1) mais augmente la profondeur de la réalisation finale (dans un rapport de 1 à  $n$ ). Dans la pratique, le logicien cherche à minimiser les deux paramètres suivants :

- un paramètre spatial ou coût : la quantité de matériel logique, généralement exprimée en nombre d'opérateurs du logigramme final (ou de boîtiers de circuits intégrés dans le cas d'une réalisation électronique) ;
- un paramètre temporel ou délai : la durée d'une ou plusieurs opérations logiques.

Si l'on rappelle qu'à chaque opérateur est attaché un certain retard (§ 1.1.7) on constate que tout système qui admet une décomposition (au sens du paragraphe 2.7.5) peut être réalisé :

- soit par une structure parallèle, d'un coût élevé (entraîné par le grand nombre des variables) et d'un faible délai ;
- soit par une structure itérative, d'un faible coût et d'un délai élevé (impliqué par la mise en série de  $n$  sous-systèmes).

On remarque qu'il existe une interdépendance entre les paramètres spatial et temporel dont la minimisation simultanée n'est généralement pas possible.

On montrera plus loin (sect. 5.4) que certains systèmes logiques séquentiels peuvent réaliser les systèmes combinatoires itératifs; la théorie correspondante est celle des transformations espace - temps : elle fait l'objet des références [18] (pp. 88 - 97), [10] (pp. 529 - 534), [11] (pp. 434 - 438), [28] (pp. 32 - 44), [31] (pp. 1 - 8 et pp. 359 - 453) et [26]. Ces transformations illustrent également l'interdépendance des paramètres spatial et temporel.

### 2.7.7 Exercice

Effectuer la synthèse d'un comparateur de deux nombres  $A$  et  $B$  représentés en binaire par trois bits chacun ( $A_2, A_1, A_0$  et  $B_2, B_1, B_0$ ). En s'inspirant des calculs du paragraphe 2.7.2, on cherchera à réaliser la fonction  $E$  à l'aide d'opérateurs OU-exclusif; puis on demande de trouver, par induction, la forme générale d'une fonction  $E$  susceptible de détecter l'égalité de deux nombres de  $n + 1$  bits chacun. La structure parallèle enfin obtenue sera comparée à la structure itérative calculée au paragraphe 2.7.4 (fig. 2.83 et 2.85).

### 2.7.8 Exercice

Effectuer la synthèse d'un système réalisant l'addition arithmétique de deux nombres  $A$  et  $B$ ;  $A$  et  $B$  sont exprimés en binaire à l'aide de  $n + 1$  bits chacun ( $A_n, A_{n-1}, \dots, A_1, A_0$  et  $B_n, B_{n-1}, \dots, B_1, B_0$ ). On demande de calculer successivement :

- la structure parallèle effectuant l'addition de  $A$  et  $B$ ; ceux-ci sont limités à deux bits chacun ( $A_1, A_0$  et  $B_1, B_0$ );

- la généralisation des équations obtenues pour déduire la structure parallèle réalisant l'addition de  $A$  et  $B$  avec  $n + 1$  bits chacun;
- la structure itérative réalisant l'addition de  $A$  et  $B$  avec  $n + 1$  bits chacun; dans ce dernier cas, on déterminera également la structure particulière réalisant l'addition arithmétique  $A + 1$ .

### 2.7.9 Exercice

Effectuer la synthèse d'un système itératif détectant si un nombre  $A$  (exprimé en binaire à l'aide de  $n + 1$  bits) est plus grand qu'un autre nombre  $B$  de  $n + 1$  bits ( $A > B$ ), égal ( $A = B$ ) ou plus petit ( $A < B$ ). Deux variantes seront calculées selon que la comparaison des deux nombres commence par les bits de poids fort ( $A_n$  et  $B_n$ ) ou de poids faible ( $A_0$  et  $B_0$ ).

## ANALYSE ET MODES DE REPRÉSENTATION DES BASCULES BISTABLES

### 3.1 ÉLÉMENT DE MÉMOIRE $\bar{s}\bar{r}$ ET SYSTÈMES SÉQUENTIELS

#### 3.1.1 Introduction

Les deux premiers chapitres ont traité les systèmes logiques sous forme d'assemblages d'éléments combinatoires idéaux (sans délai) dépourvus de rétroactions, c'est-à-dire de retours des sorties dans les entrées; en introduisant de telles rétroactions, on définit dans ce chapitre les systèmes séquentiels dont l'état de sortie ne dépend pas seulement de l'état d'entrée au même instant, mais également des états d'entrée antérieurs. Les bascules bistables sont des systèmes séquentiels élémentaires dont l'analyse temporelle est facilitée si l'on choisit une représentation discrète du paramètre temps; dans ce but, on introduit une variable logique particulière, le signal d'horloge, dont les passages à l'état 0 définissent une succession d'intervalles de temps. La description du comportement des bascules durant ces intervalles est possible à l'aide d'un modèle unique et de plusieurs modes de représentation.

#### 3.1.2 Définition : élément de mémoire $\bar{s}\bar{r}$

La figure 3.1 présente un assemblage de portes NAND appelé *élément de mémoire  $\bar{s}\bar{r}$*  (en anglais : *latch*, c'est-à-dire *verrou* ou *loquet*). On vérifie que le logigramme de la figure 3.2 est équivalent à celui de la figure 3.1 si l'on ferme l'interrupteur  $a$ , c'est-à-dire si l'on impose que les valeurs de  $y$  et  $y_1$  soient égales :

$$y = y_1 \tag{3.1}$$

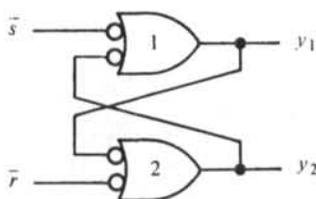


Fig. 3.1

#### 3.1.3 Commentaire

Les deux portes du logigramme de la figure 3.1 sont des systèmes logiques réalisant la fonction NAND; comme tous les systèmes logiques étudiés dans ce volume (sect. 1.1), ces deux portes NAND comportent chacune un retard (ou délai). Pour des raisons didac-

tiques, on analyse ce logigramme en supposant tout d'abord que ces retards sont nuls : il s'agit donc du modèle combinatoire de l'élément de mémoire; on montre ensuite la nécessité de tenir compte des délais et l'on introduit à cet effet le modèle asynchrone de l'élément de mémoire.

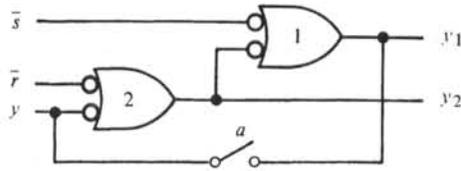


Fig. 3.2

### 3.1.4 Modèle combinatoire

Un tel modèle est dépourvu de délais (§ 1.1.10); le modèle combinatoire de l'élément de mémoire  $\bar{s}\bar{r}$  se confond donc avec les logigrammes des figures 3.1 et 3.2.

### 3.1.5 Modes de représentation

En ouvrant l'interrupteur  $a$  (fig. 3.2) l'élément de mémoire est transformé en un système combinatoire comportant deux fonctions  $y_1$  et  $y_2$  de trois variables indépendantes  $s, r, y$  :

$$y_1 = s + \bar{y}_2; y_2 = r + \bar{y} \quad (3.2)$$

En remplaçant  $y_2$  dans l'expression de  $y_1$  on obtient :

$$y_1 = s + \overline{r + \bar{y}} = s + \bar{r}y \quad (3.3)$$

Il est possible de calculer les valeurs de  $y_2$  (3.2) et de  $y_1$  (3.3) dans la table de vérité de la figure 3.3.

No	$s$	$r$	$y$	$\bar{r}y$	$y_1$	$\bar{y}$	$y_2$	Remarques
0	0	0	0	0	0	1	1	$y = y_1$
1	0	0	1	1	1	0	0	$y = y_1$
2	0	1	0	0	0	1	1	$y = y_1$
3	0	1	1	0	0	0	1	$y \neq y_1$
4	1	0	0	0	1	1	1	$y \neq y_1$
5	1	0	1	1	1	0	0	$y = y_1$
6	1	1	0	0	1	1	1	$y \neq y_1$
7	1	1	1	0	1	0	1	$y = y_1$

Fig. 3.3

En fermant l'interrupteur  $a$  (fig. 3.2) on vérifie la relation (3.1); l'élément de mémoire est alors décrit par les deux équations :

$$y_2 = r + \bar{y}_1 \quad (3.4)$$

$$y_1 = s + \bar{r}y_1 \quad (3.5)$$

L'examen de la table de vérité (fig. 3.3) montre que :

- trois états  $s, r, y$  (No 3, 4 et 6) ne vérifient pas la relation (3.1);
- les cinq états  $s, r, y$  restants (No 0, 1, 2, 5 et 7) vérifient cette même relation : ce sont les états possibles de l'élément de mémoire.

La figure 3.4 donne la table de vérité résumée de l'élément de mémoire dans laquelle sont représentés les cinq états possibles.

No	$\bar{s}$	$\bar{r}$	$s$	$r$	$y_1$	$y_2$
0	}	1	0	0	{	0
1						1
2		0	0	1	0	1
5		0	1	0	1	0
7		0	1	1	1	1

Fig. 3.4

### 3.1.6 Définitions : système logique séquentiel

On constate que pour un état d'entrée de l'élément de mémoire (fig. 3.4 :  $s, r = 00$ ) il existe deux états de sortie possibles ( $y_1, y_2 = 01$  et  $10$ ) : l'état de sortie à un instant  $t$  n'est donc pas entièrement déterminé par l'état d'entrée à ce même instant, mais également par un état d'entrée précédent. On appelle *système logique séquentiel* tout système qui admet pour un état d'entrée au moins plusieurs états de sortie possibles.

Le comportement d'un système séquentiel se décompose en un *régime permanent* (ou *établi* ou *stable*) tel qu'il est défini par la table de vérité (fig. 3.4) et un *régime transitoire* (ou *instable*); celui-ci décrit les passages d'un régime permanent à un autre régime permanent.

### 3.1.7 Description du régime transitoire

Considérons les séquences suivantes des états d'entrée  $s, r$  et déterminons dans chaque cas les états de sortie  $y_1, y_2$  résultants (fig. 3.4) :

- si l'on passe de  $s, r = 01$  à  $s, r = 00$  grâce à une variation de  $r$ , il est raisonnable de penser que le système va conserver par inertie son état de sortie initial  $y_1, y_2 = 01$  qui est également possible pour l'état d'entrée final;
- si l'on passe de  $s, r = 10$  à  $s, r = 00$  grâce à une variation de  $s$ , le système peut conserver pour la même raison son état de sortie initial  $y_1, y_2 = 10$ ;
- si enfin l'on passe de  $s, r = 11$  à  $s, r = 00$  grâce à une variation simultanée de  $s$  et  $r$  (ce qui est possible en imposant  $s = r$ ), alors le système quitte un état de sortie  $y_1, y_2 = 11$  pour aller soit en  $01$ , soit en  $10$  : le résultat final semble imprévisible.

### 3.1.8 Condition

Toute incertitude quant à la valeur finale de l'état de sortie  $y_1, y_2$  peut être éliminée si l'on interdit l'état d'entrée  $s, r = 11$ ; l'interdiction de l'état  $s, r = 00$  n'aurait aucun sens, puisqu'on supprimerait le caractère séquentiel de l'élément de mémoire.

Pour interdire l'état  $s, r = 11$ , il suffit d'imposer la condition suivante :

$$s \cdot r = 0 \quad (3.6)$$

L'examen de la figure 3.4 montre que si la relation (3.6) est vérifiée, alors on vérifie aussi l'égalité suivante :

$$y_2 = \bar{y}_1 \quad (3.7)$$

### 3.1.9 Modèle asynchrone

Le modèle combinatoire de l'élément de mémoire nous a permis de calculer son régime permanent; par contre, la description du régime transitoire (§ 3.1.7) a nécessité une explication intuitive qui peut être remplacée par un calcul rigoureux si l'on recourt au modèle asynchrone (§ 1.1.9). En admettant que chaque porte NAND de la figure 3.1 est mise en série avec un élément de délai (§ 1.1.13) caractérisé par un retard  $\Delta_1$  (pour la porte 1) et  $\Delta_2$  (pour la porte 2) on obtient l'assemblage de la figure 3.5 qui est un système logique asynchrone. L'étude générale de ces systèmes fait l'objet du chapitre 7 : on se borne ici à analyser le cas particulier de l'élément de mémoire.

Si l'on fait tendre vers zéro les retards  $\Delta_1$  et  $\Delta_2$  on retrouve le logigramme de la figure 3.1 qui est le modèle combinatoire de l'élément de mémoire.

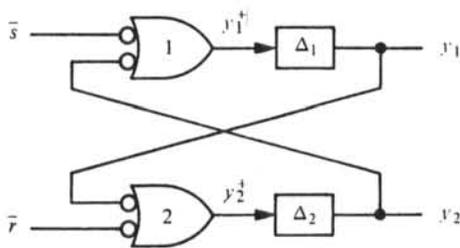


Fig. 3.5

### 3.1.10 Modes de représentation

Les entrées  $y_1^+$  et  $y_2^+$  des éléments de délai peuvent s'exprimer algébriquement en fonction des grandeurs  $s, r, y_1$  et  $y_2$  (fig. 3.5) :

$$y_1^+ = s + \bar{y}_2 \quad (3.8)$$

$$y_2^+ = r + \bar{y}_1 \quad (3.9)$$

Par définition (§ 1.1.13) la sortie  $y_1$  ( $y_2$ ) de l'élément de délai  $\Delta_1$  ( $\Delta_2$ ) est égale à l'entrée  $y_1^+$  ( $y_2^+$ ) du même élément retardée dans le temps d'une durée  $\Delta_1$  ( $\Delta_2$ ) :

$$y_1(t + \Delta_1) = y_1^+(t) \quad (3.10)$$

$$y_2(t + \Delta_2) = y_2^+(t) \quad (3.11)$$

Le régime permanent est caractérisé par des retards nuls; on vérifie alors que :

$$y_1(t) = y_1^+(t) \quad (3.12)$$

$$y_2(t) = y_2^+(t) \quad (3.13)$$

En remplaçant les expressions (3.12) et (3.13) dans (3.8) et (3.9) on retrouve l'équation (3.5) décrivant le régime permanent de l'élément de mémoire  $\bar{s}\bar{r}$  :

$$y_1 = s + \bar{y}_2 = s + \overline{r + y_1} = s + \bar{r}y_1$$

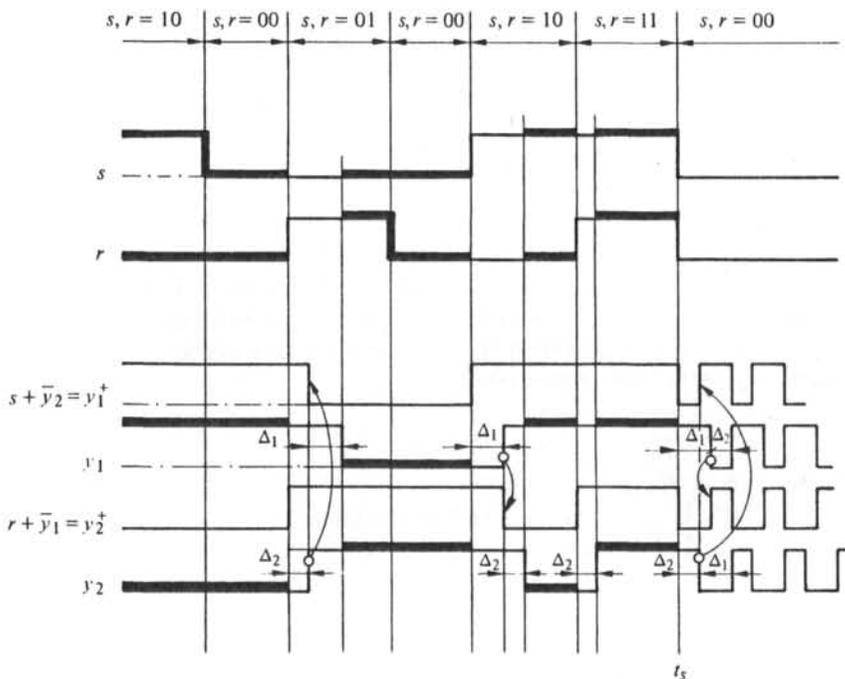


Fig. 3.6

Les régimes permanent et transitoire peuvent être représentés dans un chronogramme (fig. 3.6) en procédant comme suit :

- on admet une séquence d'états d'entrée  $s, r$  ;
- on admet au départ un régime permanent : le premier état d'entrée ( $s, r = 10$ ) produit un état de sortie  $y_1, y_2 = y_1^+, y_2^+ = 10$  (fig. 3.4) ;
- l'apparition du deuxième état d'entrée ( $s, r = 00$ ) n'implique aucune variation des valeurs de  $y_1^+$  et  $y_2^+$  calculées par (3.8) et (3.9); l'état de sortie  $y_1, y_2$  est donc inchangé;
- l'apparition du troisième état d'entrée ( $s, r = 01$ ) entraîne une variation de  $r$  et, par conséquent, de  $y_2^+$  (3.9); après un retard  $\Delta_2$ , cette variation apparaît sur la sortie  $y_2$  et provoque à son tour un changement de  $y_1^+$  (3.8); après un nouveau

retard  $\Delta_1$ , la variation de  $y_1^+$  est retransmise sur la sortie  $y_1$  : le régime transitoire est terminé (ou amorti) ;

- le calcul se poursuit de façon identique jusqu'à l'instant  $t_s$  où l'on impose la variation  $s, r = 11 \rightarrow 00$ ; il apparaît dans ce cas un régime transitoire sans amortissement : les sorties  $y_1$  et  $y_2$  oscillent indéfiniment avec une période égale à  $\Delta_1 + \Delta_2$ .

Le régime permanent de l'élément de mémoire est représenté en trait fort dans la figure 3.6 : ce régime est celui de la table de vérité du modèle combinatoire (fig. 3.4). Tout régime transitoire disparaît après une durée maximale égale à  $\Delta_1 + \Delta_2$  sauf dans le cas de la séquence  $s, r = 11 \rightarrow 00$ ; cette dernière situation doit être interdite par la condition (3.6) :  $s \cdot r = 0$ . L'expérience montre en fait que l'oscillation calculée est amortie : l'élément de mémoire atteint l'un des deux états permanents  $y_1, y_2 = 01$  ou  $10$ ; l'explication de cet écart entre le comportement du modèle asynchrone et celui du système concret est donnée au chapitre 7.

### 3.1.11 Conclusion

Les systèmes logiques séquentiels se distinguent des systèmes combinatoires par leur structure, qui comporte une ou plusieurs boucles de rétroaction (fig. 3.2), et par leur comportement; celui-ci se décompose en un régime permanent, qui admet pour un état d'entrée au moins plusieurs états de sortie possibles, et en un régime transitoire, généralement amorti. La détermination du régime transitoire nécessite la présence de délais dans la ou les boucles de rétroaction.

### 3.1.12 Description intuitive

Si l'on vérifie la condition  $s \cdot r = 0$  (3.6), le régime permanent de l'élément de mémoire peut se résumer de la façon suivante (voir la figure 3.4) :

- pour  $s = \bar{r}$ , la sortie  $y_1$  est égale à l'entrée  $s$  au même instant  $t$  :

$$y_1(t) = s(t) = \bar{r}(t) \quad (3.14)$$

On dit que l'élément de mémoire a enregistré la valeur de l'entrée  $s = \bar{r}$ .

- Pour  $s = r = 0$  la sortie  $y_1$  à l'instant  $t$  est égale à la valeur antérieure de l'entrée  $s = \bar{r}$  définie à un instant  $(t - \Delta t)$  :

$$y_1(t) = y_1(t - \Delta t) = s(t - \Delta t) = \bar{r}(t - \Delta t) \quad (3.15)$$

L'élément de mémoire a donc mémorisé la valeur  $s = \bar{r}$  précédemment enregistrée.

Le régime transitoire (fig. 3.6) montre qu'un certain retard  $\Delta$  (au maximum égal à  $\Delta_1 + \Delta_2$ ) sépare la variation de l'état d'entrée  $s, r$  du changement de la sortie  $y_1$ .

L'élément  $\bar{s}\bar{r}$  est une mémoire élémentaire qui peut conserver un bit d'information, c'est-à-dire la valeur logique de l'entrée  $s = \bar{r}$ . L'abréviation  $\bar{s}\bar{r}$  provient de l'appellation anglaise *set-reset* (*set* : mise à 1; *reset* : remise à 0).

### 3.1.13 Exercice

Analyser les modèles combinatoire et asynchrone du logigramme de la figure 3.7; mettre en évidence les parentés avec l'élément de mémoire  $\bar{s}\bar{r}$ .

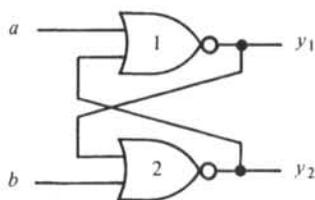


Fig. 3.7

## 3.2 BASCULE BISTABLE SR

### 3.2.1 Définitions

La figure 3.8 est un assemblage de portes NAND appelé *bascule bistable SR* (en anglais *SR flip-flop*); on reconnaît la présence de deux éléments de mémoire  $\bar{s}\bar{r}$  (fig. 3.1) qui sont des systèmes séquentiels : la bascule SR est donc un tel système. Le premier des éléments de mémoire est généralement surnommé le *maître* et le second l'*esclave* (en anglais : structure *master-slave*).

Le schéma de la figure 3.9 met en évidence des entrées ( $S, R, \overline{CK}, \overline{PR}, \overline{CLR}$ ) et des sorties ( $Q, \bar{Q}$ ) qui sont appelées dans ce volume :

- les entrées d'excitation  $S$  et  $R$  (en anglais : *Set* et *Reset*);
- le signal d'horloge  $CK$  ou son inverse  $\overline{CK}$  (en anglais : *Clock pulse*);
- les entrées asynchrones  $PR$  et  $CLR$  ou leurs inverses  $\overline{PR}$  et  $\overline{CLR}$  (en anglais : *Preset* et *Clear*);
- les sorties  $Q$  et  $\bar{Q}$ .

La valeur de la sortie  $Q$  est l'*état* (ou l'*état interne*) de la bascule.

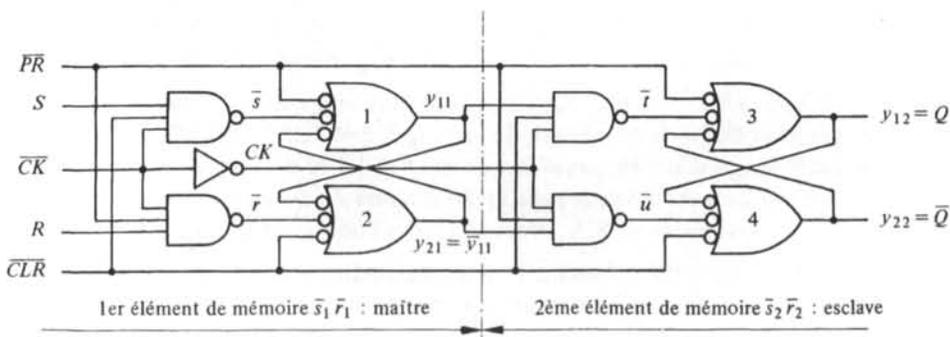


Fig. 3.8

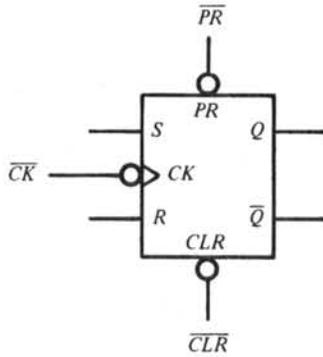


Fig. 3.9

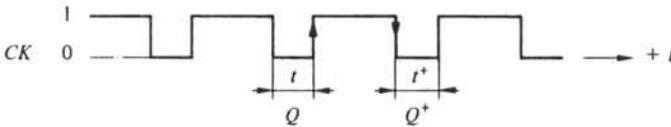


Fig. 3.10

### 3.2.2 Définitions : instants d'horloge

Le signal d'horloge  $CK$  est une variable logique privilégiée qui joue le rôle de signal de référence pour la bascule bistable : il peut être périodique ou non. On appelle *intervalles d'horloge* ou, par abus de langage, *instants d'horloge* les intervalles de temps pendant lesquels  $CK = 0$ . Si un instant d'horloge quelconque est l'*instant présent*  $t$ , l'instant d'horloge immédiatement suivant est appelé *instant futur*  $t^+$  (fig. 3.10). L'état  $Q$  de la bascule à l'instant  $t$  est l'*état présent*, tandis que l'état  $Q^+$  à l'instant  $t^+$  est l'*état futur*.

On appelle *montée* du signal d'horloge  $CK$  le passage de 0 à 1 et *descente* de  $CK$  le passage inverse (fig. 3.10).

### 3.2.3 Modèle quasi-synchrone

Toutes les portes NAND de la bascule  $SR$  (fig. 3.8) comportent un retard (sect. 1.1) : il en découle un modèle asynchrone qui est étudié au chapitre 7. Pour des raisons didactiques on cherche un modèle simplifié de la bascule qu'on appellera *modèle quasi-synchrone* et qui est obtenu en admettant les deux hypothèses suivantes :

- tous les délais des portes NAND de la bascule sont nuls à l'exception de ceux des deux éléments de mémoire (portes 1, 2, 3 et 4 de la figure 3.8);
- **toute variation des entrées d'excitation ( $S$  et  $R$ ) et des entrées asynchrones ( $PR$  et  $CLR$ ) est interdite pendant les instants d'horloge, c'est-à-dire lorsque  $CK = 0$ ; les valeurs de  $S$ ,  $R$ ,  $PR$  et  $CLR$  sont donc constantes quand  $CK = 0$ .**

La première hypothèse simplifie le modèle asynchrone en éliminant certains régimes transitoires : le régime permanent est inchangé.

La seconde hypothèse impose aux entrées de la bascule des restrictions de fonctionnement; tant que ces restrictions sont respectées, le modèle quasi-synchrone est valable; dans le cas contraire on est renvoyé au modèle asynchrone du chapitre 7.

### 3.2.4 Analyse du premier élément de mémoire $\bar{s}_1\bar{r}_1$

Celui-ci comporte deux portes NAND à trois entrées chacune (fig. 3.8 et 3.11); pour se ramener à la forme habituelle de l'élément de mémoire (fig. 3.1) on effectue deux transformations de logigramme successives (fig. 3.12 et 3.13) dont on déduit les expressions de  $\bar{s}_1$  et  $\bar{r}_1$ , puis à l'aide du théorème de De Morgan, celles de  $s_1$  et  $r_1$  :

$$\bar{s}_1 = \bar{s} \cdot \overline{PR} = \overline{S \cdot \overline{CLR} \cdot \overline{CK} \cdot PR} \quad (3.16)$$

$$s_1 = S \cdot \overline{CLR} \cdot \overline{CK} + PR \quad (3.17)$$

$$\bar{r}_1 = \bar{r} \cdot \overline{CLR} = \overline{R \cdot \overline{PR} \cdot \overline{CK} \cdot \overline{CLR}} = (\bar{R} + PR + CK) \cdot \overline{CLR} \quad (3.18)$$

$$r_1 = R \cdot \overline{PR} \cdot \overline{CK} + CLR \quad (3.19)$$

En respectant la condition  $s_1 \cdot r_1 = 0$  (3.6), c'est-à-dire en vérifiant la relation :

$$(S \cdot \overline{CLR} \cdot \overline{CK} + PR)(R \cdot \overline{PR} \cdot \overline{CK} + CLR) = 0 \quad (3.20)$$

qui se résume à :

$$S \cdot R \cdot \overline{PR} \cdot \overline{CLR} \cdot \overline{CK} + PR \cdot CLR = 0 \quad (3.21)$$

alors on peut décrire l'élément de mémoire avec les deux équations (3.7) et (3.5) :

$$y_{21} = \bar{y}_{11} \quad (3.22)$$

$$y_{11} = s_1 + \bar{r}_1 \cdot y_{11} = S \cdot \overline{CLR} \cdot \overline{CK} + PR + (\bar{R} + PR + CK) \cdot \overline{CLR} \cdot y_{11} \quad (3.23)$$

Cette dernière expression peut être simplifiée et devient enfin :

$$y_{11} = PR + (S \cdot \overline{CK} + \bar{R} \cdot y_{11} + CK \cdot y_{11}) \cdot \overline{CLR} \quad (3.24)$$

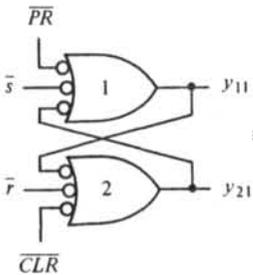


Fig. 3.11

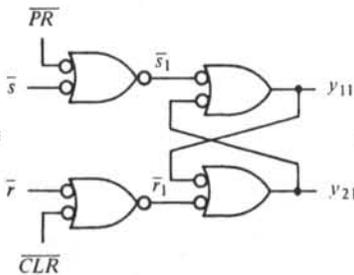


Fig. 3.12

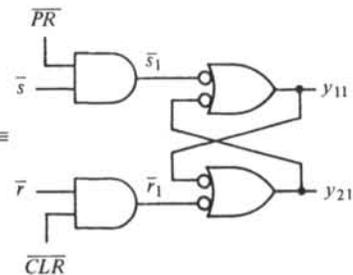


Fig. 3.13

### 3.2.5 Analyse du second élément de mémoire $\bar{s}_2\bar{r}_2$

Une transformation de logigramme analogue à celle du paragraphe 3.2.4 nous conduit au schéma de la figure 3.14 dont on tire les expressions de  $\bar{s}_2$ ,  $s_2$ ,  $\bar{r}_2$  et  $r_2$  :

$$\bar{s}_2 = \bar{t} \cdot \overline{PR} = \overline{y_{11} \cdot \overline{CLR} \cdot CK \cdot PR} \quad (3.25)$$

$$s_2 = y_{11} \cdot \overline{CLR} \cdot CK + PR \quad (3.26)$$

$$\bar{r}_2 = \bar{u} \cdot \overline{CLR} = \overline{\bar{y}_{11} \cdot \overline{PR} \cdot CK} \cdot \overline{CLR} = (y_{11} + PR + \overline{CK}) \cdot \overline{CLR} \quad (3.27)$$

$$r_2 = \bar{y}_{11} \cdot \overline{PR} \cdot CK + CLR \quad (3.28)$$

La condition  $s_2 \cdot r_2 = 0$  (3.6) devient dans ce cas :

$$PR \cdot CLR = 0 \quad (3.29)$$

Si la relation (3.21) est satisfaite, alors la relation (3.29) l'est aussi : la condition (3.29) est donc incluse dans (3.21). En respectant celle-ci on peut décrire le second élément de mémoire avec les deux équations (3.7) et (3.5) :

$$y_{22} = \bar{y}_{12} = \bar{Q} \quad (3.30)$$

$$Q = y_{12} = s_2 + \bar{r}_2 y_{12} = y_{11} \cdot \overline{CLR} \cdot CK + PR + (y_{11} + PR + \overline{CK}) \cdot \overline{CLR} \cdot Q \quad (3.31)$$

dont la forme simplifiée est :

$$Q = PR + (CK \cdot y_{11} + \overline{CK} \cdot Q) \cdot \overline{CLR} \quad (3.32)$$

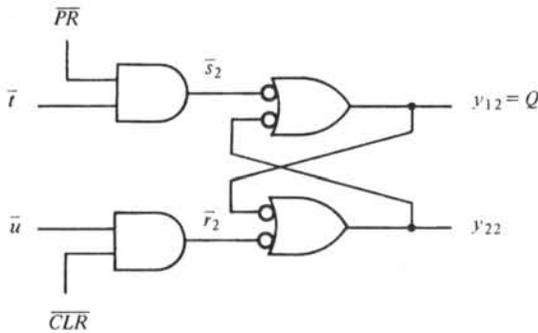


Fig. 3.14

### 3.2.6 Définitions

Les deux entrées asynchrones  $PR$  et  $CLR$  déterminent quatre états :

$PR, CLR = 00$  caractérise le *fonctionnement normal* de la bascule ; les équations des deux éléments de mémoire (3.24) et (3.32) deviennent dans ce cas :

$$y_{11} = S \cdot \overline{CK} + (\bar{R} + CK) y_{11} \quad (3.33)$$

$$Q = CK \cdot y_{11} + \overline{CK} \cdot Q \quad (3.34)$$

avec la condition (3.21) :

$$S \cdot R \cdot \overline{CK} = 0 \quad (3.35)$$

$PR, CLR = 10$  est la *mise à 1* de la bascule ; la variable  $y_{11}$  (3.24) et la sortie  $Q$  (3.32) prennent en effet la valeur suivante :

$$y_{11} = Q = 1 \quad (3.36)$$

tandis que la condition (3.21) est satisfaite pour toutes les valeurs de  $S, R$  et  $CK$ .

$PR, CLR = 01$  est la *remise* à 0 de la bascule dont la variable  $y_{11}$  (3.24) et la sortie  $Q$  (3.32) valent alors :

$$y_{11} = Q = 0 \quad (3.37)$$

La condition (3.21) est toujours satisfaite.

$PR, CLR = 11$  est un état interdit par la condition (3.21).

### ■ 3.2.7 Mode de représentation : fonctionnement normal

On désire représenter les régimes permanent et transitoire de la bascule  $SR$  (fig. 3.8) dans le fonctionnement normal; un chronogramme peut être établi en procédant comme suit (fig. 3.15) :

- on choisit une séquence d'états d'entrée  $S, R$  qui vérifie la condition (3.35) (état  $S, R = 11$  interdit lorsque  $CK = 0$ ) et qui valide le modèle quasi-synchrone de la bascule (§ 3.2.3 : variations de  $S$  et  $R$  interdites lorsque  $CK = 0$ ).
- On considère la bascule comme une juxtaposition des deux éléments de mémoire  $\bar{s}_1 \bar{r}_1$  et  $\bar{s}_2 \bar{r}_2$ ; les expressions particulières de  $s_1, r_1, s_2$  et  $r_2$  valables dans le fonctionnement normal se déduisent des relations (3.17), (3.19), (3.26) et (3.28) en faisant  $PR = CLR = 0$  :

$$s_1 = S \cdot \overline{CK} \quad ; \quad r_1 = R \cdot \overline{CK} \quad (3.38)$$

$$s_2 = y_{11} \cdot CK \quad ; \quad r_2 = \bar{y}_{11} \cdot CK \quad (3.39)$$

A partir de la séquence des états  $S, R$  on peut tracer le chronogramme de  $s_1$  et  $r_1$  (3.38).

- On considère le premier instant d'horloge  $t_1$ ; grâce à la description intuitive de l'élément de mémoire  $\bar{s}\bar{r}$  (§ 3.1.12), les valeurs de  $s_1$  et  $r_1$  (3.38) déterminent la valeur de  $y_{11}$  (3.14) qui caractérise le régime permanent de l'élément  $\bar{s}_1 \bar{r}_1$  :

$$y_{11}(t_1) = \bar{y}_{21}(t_1) = s_1(t_1) = \bar{r}_1(t_1) = 1 \quad (3.40)$$

On constate dans le chronogramme que les valeurs  $s_1(t_1) = \bar{r}_1(t_1) = 1$  entraînent une variation des valeurs de  $y_{11}$  et de  $y_{21}$ ; conformément à la description du paragraphe 3.1.12 et de la figure 3.6, ce changement n'est pas instantané : les retards séparant les variations de  $s_1, r_1$  de celles de  $y_{11}$  et  $y_{21}$  sont représentés dans la figure 3.15 et traduisent le régime transitoire de l'élément  $\bar{s}_1 \bar{r}_1$ .

- Au premier instant d'horloge  $t_1$  on peut également déterminer la valeur de la sortie  $Q$  du second élément de mémoire  $\bar{s}_2 \bar{r}_2$  en régime permanent; la relation (3.15) valable pour  $s_2 = r_2 = 0$  donne en effet :

$$Q(t_1) = y_{12}(t_1) = y_{12}(t_0) = s_2(t_0) = \bar{r}_2(t_0) = 0 \quad (3.41)$$

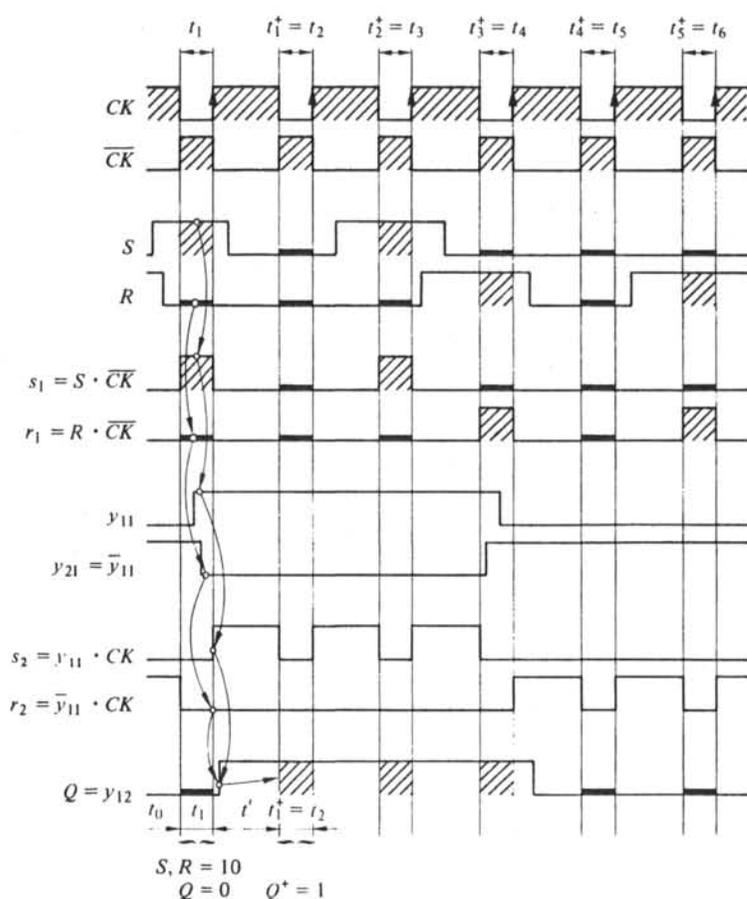


Fig. 3.15

- Dès la fin du premier instant d'horloge  $t_1$  et jusqu'au deuxième instant d'horloge  $t_2 = t_1^+$ , c'est-à-dire pendant l'intervalle de temps  $t'$ , on détermine successivement les valeurs en régime permanent de  $y_{11}$  (3.15), de  $s_2$  et  $r_2$  (3.39) et de  $y_{12} = Q$  (3.14) :

$$y_{11}(t') = y_{11}(t_1) = 1 \quad (3.42)$$

$$s_2(t') = y_{11} \cdot CK = 1 ; \quad r_2(t') = \bar{y}_{11} \cdot CK = 0 \quad (3.43)$$

$$Q(t') = y_{12}(t') = s_2(t') = \bar{r}_2(t') = 1 \quad (3.44)$$

La sortie  $Q$  prend sa nouvelle valeur après un certain retard dû au régime transitoire.

- Au deuxième instant d'horloge  $t_2$  on obtient :

$$y_{11}(t_2) = y_{11}(t') = 1 \quad (3.45)$$

$$s_2(t_2) = r_2(t_2) = 0 \quad (3.46)$$

$$Q(t_2) = Q(t') = 1 \quad (3.47)$$

La sortie  $Q$  n'a pas changé.

- Le tracé du chronogramme se poursuit de manière identique jusqu'à l'instant d'horloge  $t_6$ .

### 3.2.8 Description intuitive : fonctionnement normal

On remarque dans le chronogramme (fig. 3.15) qu'au premier instant d'horloge  $t_1$  (instant présent) l'élément de mémoire  $\overline{s_1} r_1$  enregistre les valeurs des entrées d'excitation  $S$  et  $R$  tandis que le second élément  $\overline{s_2} r_2$  conserve en mémoire son état de sortie précédent  $Q$ . A la montée du signal d'horloge  $CK$  (fin de l'instant présent) l'élément  $\overline{s_1} r_1$  mémorise les valeurs de  $S$  et  $R$  enregistrées précédemment et les transmet au second élément qui les enregistre et dont la sortie  $Q$  peut changer. Au second instant d'horloge (instant futur) le premier élément enregistre de nouvelles valeurs des entrées d'excitation tandis que le second élément mémorise sa précédente valeur de sortie. On peut donc assimiler la bascule bistable à un sas dont les deux éléments maître et esclave sont alternativement ouverts, puis refermés.

Les trois caractéristiques suivantes de la bascule seront d'un grand intérêt pratique :

- une variation de l'état  $Q$  de la bascule ne peut se produire qu'immédiatement après une montée du signal d'horloge  $CK$ ;
- l'état  $Q$  de la bascule est constant pendant les instants d'horloge ( $CK = 0$ );
- les valeurs des entrées  $S$ ,  $R$  et de la sortie  $Q$  à l'instant présent  $t$  déterminent complètement la valeur  $Q^+$  de la sortie à l'instant futur  $t^+$ ; les flèches du chronogramme (fig. 3.15) mettent en évidence cette dernière propriété.

### 3.2.9 Modes de représentation : mise à 1 et remise à 0

On vient de constater que l'état d'une bascule en fonctionnement normal ne peut changer qu'à un moment particulier (à la montée du signal d'horloge); tout en respectant le modèle quasi-synchrone (§ 3.2.3 : variations de  $S$ ,  $R$ ,  $PR$  et  $CLR$  interdites lorsque  $CK = 0$ ) on peut chercher à modifier cet état indépendamment de l'action du signal d'horloge. Les valeurs  $PR, CLR = 10$  et  $PR, CLR = 01$  des entrées asynchrones imposent à la bascule des régimes permanents caractérisés par  $y_{11} = Q = 1$  (3.36) et  $y_{11} = Q = 0$  (3.37); l'étude du régime transitoire des éléments de mémoire (§ 3.1.10 et fig. 3.6) suggère qu'un certain retard sépare les variations de  $PR, CLR$  de celle de la sortie  $Q$ . Le chronogramme de la figure 3.16 illustre le comportement global (régimes permanent et transitoire) d'une bascule soumise à certaines variations des entrées d'excitation (changement de  $Q$  lors des montées du signal d'horloge) auxquelles se superposent des variations des entrées asynchrones  $PR$  et  $CLR$ ; le tableau de la figure 3.17 résume les régimes permanents définis par les quatre états  $PR, CLR$  (§ 3.2.6).

### 3.2.10 Modèle synchrone

Un cas particulier du modèle quasi-synchrone de la bascule (§ 3.2.3) peut être obtenu en admettant les deux hypothèses suivantes :

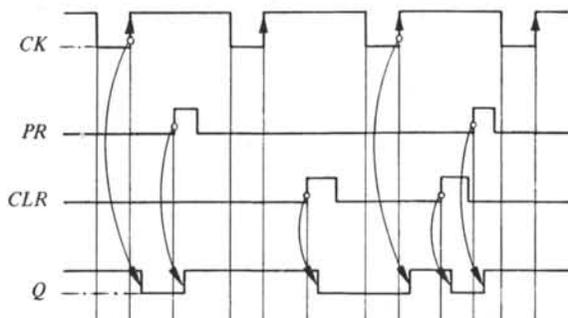


Fig. 3.16

PR	CLR	$y_{12} = Q$	$y_{22}$	Fonctionnement
0	0	$\begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix}$		Normal
0	1	0	1	Remise à 0
1	0	1	0	Mise à 1
1	1	1	1	Interdit

Fig. 3.17

- *le fonctionnement est normal* ( $PR = CLR = 0$ );
- *l'état  $Q$  est défini pendant les seuls instants d'horloge, c'est-à-dire lorsque  $CK = 0$ .*

En d'autres termes on représente le comportement de la bascule tel qu'il est décrit par la figure 3.15 (fonctionnement normal) en se limitant aux instants d'horloge ( $CK = 0$ ).

Il découle de ce modèle des modes de représentation très commodes (§ 3.2.11) : pendant les instants d'horloge les entrées d'excitation  $S, R$  sont invariantes par hypothèse (§ 3.2.3 : modèle quasi-synchrone) tandis que l'état  $Q$  est également invariant par construction (§ 3.2.8).

### 3.2.11 Modes de représentation

La description intuitive du paragraphe 3.2.8 nous suggère d'établir un tableau (fig. 3.18) donnant pour les huit états  $S, R, Q$  (définis à l'instant présent  $t$ ) l'état  $Q^+$  (défini à l'instant futur  $t^+$ ) :

- la condition (3.35) interdit les valeurs  $S = R = 1$ ; les états  $S, R, Q$  correspondants (Nos 6 et 7) produisent un état futur indéterminé qu'on notera  $Q^+ = \emptyset$  (selon les définitions du paragraphe 1.7.2);
- les six états  $S, R, Q$  restants sont visibles dans le chronogramme de la figure 3.15; on relève pour chacun d'eux l'état  $Q^+$  correspondant.

Un tel tableau est la *table d'états* de la bascule  $SR$  appelée fréquemment, par abus de langage, table de vérité; une présentation résumée est celle de la figure 3.19, tandis

qu'une disposition conforme à celle de Karnaugh (fig. 2.44) est donnée dans la figure 3.20. Cette dernière représentation suggère une simplification algébrique selon la méthode de la section 2.2. Si l'on respecte la condition (3.35) qui, lors d'un instant d'horloge ( $CK = 0$ ), se résume à :

$$S \cdot R = 0 \quad (3.48)$$

l'état futur  $Q^+$  s'exprime en fonction de l'état présent et des variables d'excitation par :

$$Q^+ = S + \bar{R}Q \quad (3.49)$$

L'expression (3.49) est l'équation caractéristique de la bascule  $SR$ ; associée à la condition (3.48), elle décrit complètement la table d'états originale (fig. 3.18 ou 3.20).

No	S	R	Q	$Q^+$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	$\emptyset$
7	1	1	1	$\emptyset$

Fig. 3.18

S	R	$Q^+$
0	0	Q
0	1	0
1	0	1
1	1	$\emptyset$

Fig. 3.19

		SR			
		00	01	11	10
Q	0			$\emptyset$	1
	1	1		$\emptyset$	1
		$Q^+$			

Fig. 3.20

### 3.2.12 Définitions

Les modes de représentation du paragraphe 3.2.11 déterminent le comportement futur de la bascule (état  $Q^+$ ) à partir d'un logigramme observé au moment présent (état  $Q$ , entrées d'excitation  $S$  et  $R$ ) : on dira qu'il s'agit de *modes de représentation analytiques* (dont l'étude fait l'objet de la section 3.3) par opposition aux modes de représentation synthétiques (sect. 3.4).

Par analogie avec les définitions des systèmes combinatoires, on dit que la table d'états de la figure 3.18 définit une application qui est la *fonction caractéristique* de la bascule bistable (voir le paragraphe 1.1.12); on dit aussi qu'une bascule réalise une certaine fonction caractéristique ou une certaine table d'états. Deux bascules distinctes qui réalisent la même fonction sont *fonctionnellement équivalentes* (voir le paragraphe 1.4.7); les diverses équations caractéristiques qui décrivent la même fonction sont équivalentes aussi.

Par analogie avec les définitions des fonctions incomplètement définies (voir le paragraphe 1.7.2), on dit qu'une fonction complètement définie  $q^+$  qui est égale à  $Q^+$  dans tout état où  $Q^+ \neq \emptyset$  est une *solution* de  $Q^+$  et l'on écrit :

$$q^+ \in Q^+ \quad (3.50)$$

Si  $Q^+$  possède  $n$  états  $\emptyset$ , elle a  $2^n$  solutions qui constituent une *famille de fonctions*.

La table d'états de la bascule *SR* (fig. 3.20) définit ainsi une famille de  $2^2 = 4$  fonctions qui constituent chacune une solution de  $Q^+$ ; l'expression (3.49) décrit l'une d'entre elles.

### 3.2.13 Conclusion

On a analysé dans cette section une structure possible de la bascule *SR* dite maître-esclave; d'autres structures peuvent réaliser la même fonction caractéristique : leur analyse et leur synthèse seront discutées au chapitre 7.

### 3.2.14 Bibliographie

Une justification de la structure maître-esclave fait l'objet de la référence [12] (pp. 167 - 178) tandis qu'une discussion détaillée sur le régime transitoire des bascules et la fréquence maximale de leur signal d'horloge se trouve dans [28] (pp. 161 - 170). L'introduction du modèle synchrone et des modes de représentation associés est principalement due à Phister [34] (pp. 112 - 142).

### 3.2.15 Exercice

En analysant le logigramme de la bascule *SR* (fig. 3.8) on demande de compléter le chronogramme de la figure 3.16 avec une représentation des variables  $S, R, s_1, r_1, y_{11}, y_{21}, s_2, r_2$  et  $y_{22}$ ; on décrira les régimes permanent et transitoire de ces variables et l'on vérifiera en particulier l'action des entrées asynchrones *PR* et *CLR* sur les sorties  $Q$  et  $y_{22}$ . Enfin, on montrera les conséquences de l'application d'une séquence d'états (normalement interdite)  $PR, CLR = 11 \rightarrow 00$ .

## 3.3 MODES DE REPRÉSENTATION ANALYTIQUES

### 3.3.1 Définition : bascule *D*

La bascule *D* (en anglais *Delay flip-flop*) est une bascule *SR* qui vérifie l'équation

$$S = \bar{R} = D \quad (3.51)$$

Le logigramme de la figure 3.21 représente une bascule *D* réalisée à partir d'une bascule *SR* (voir figure 3.9) tandis que la figure 3.22 donne le schéma conventionnel; dans ce dernier cas, on a admis à titre d'exemple que le signal d'horloge était disponible sous la forme vraie (*CK*).

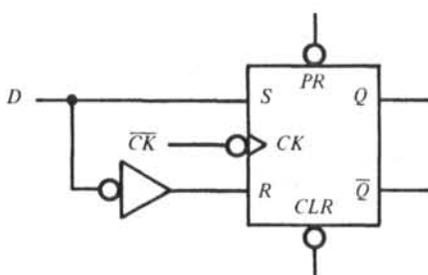


Fig. 3.21

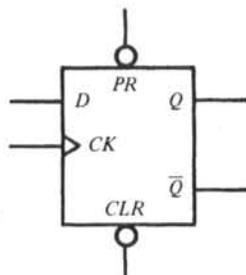


Fig. 3.22

### 3.3.2 Modes de représentation

En introduisant la relation (3.51) dans l'expression (3.49) de la bascule  $SR$  on obtient l'équation caractéristique de la bascule  $D$  :

$$Q^+ = D + DQ = D(1 + Q) = D, \text{ c'est-à-dire :} \\ Q^+ = D \quad (3.52)$$

La condition (3.48) devient alors :

$$SR = D\bar{D} \equiv 0 \quad (3.53)$$

Cette dernière équation est identiquement nulle; il n'y a donc aucune condition à respecter lors de l'emploi de la bascule  $D$  dans les limites du modèle synchrone (§ 3.2.10).

Les diverses tables d'états (fig. 3.23, 3.24 et 3.25) peuvent être obtenues soit à partir des tables correspondantes de la bascule  $SR$  (fig. 3.18 à 3.20) en vérifiant la relation  $S = \bar{R} = D$ , soit à partir de l'équation caractéristique  $Q^+ = D$ .

No	D	Q	Q <sup>+</sup>
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	1

Fig. 3.23

D	Q <sup>+</sup>
0	0
1	1

Fig. 3.24

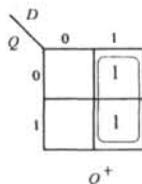


Fig. 3.25

### 3.3.3 Commentaire

L'état futur  $Q^+$  est donc égal à l'état d'entrée présent  $D$  : la bascule  $D$  est un système séquentiel introduisant un délai élémentaire (égal à la période du signal d'horloge) entre l'entrée  $D$  et la sortie  $Q$ .

### 3.3.4 Analyse d'une bascule $sr$

Le logigramme de la figure 3.26 représente une telle bascule dont l'équation caractéristique est :

$$q^+ = D = s + \bar{r} Q \quad (3.54)$$

La table d'états correspondante est celle de la figure 3.27; il n'y a pas de condition sur les états  $s, r, Q$ .

En comparant les tables d'états de la bascule  $SR$  (fig. 3.20) et de la bascule  $sr$  (fig. 3.27) on constate que la fonction  $q^+$  de celle-ci est une solution de celle-là (selon la définition du paragraphe 3.2.12) si l'on fait  $s \equiv S$  et  $r \equiv R$ .

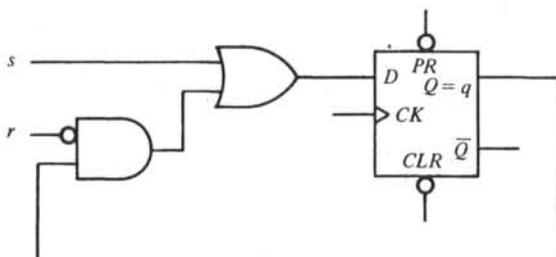


Fig. 3.26

$sr$ $Q$	00	01	11	10
	0		1	1
1	1		1	1
	$q^+$			

Fig. 3.27

### 3.3.5 Définition : bascule $JK$

La bascule  $JK$  (en anglais : *JK flip-flop* où  $J$  et  $K$  n'ont aucune signification particulière) est l'assemblage d'une bascule  $SR$  et de deux portes ET qui réalisent les fonctions suivantes :

$$S = J\bar{Q} ; R = KQ \quad (3.55)$$

Les logigramme et schéma correspondants sont ceux des figures 3.28 et 3.29.

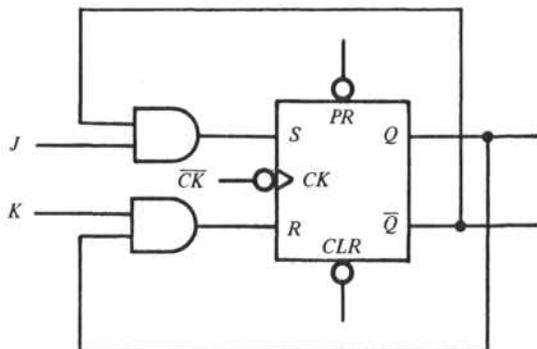


Fig. 3.28

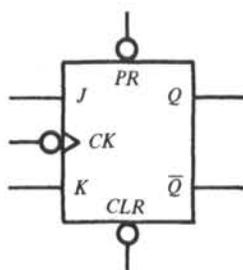


Fig. 3.29

### 3.3.6 Modes de représentation

En introduisant les deux relations (3.55) dans l'expression (3.49) de la bascule  $SR$  on détermine l'équation caractéristique de la bascule  $JK$  :

$$Q^+ = S + \bar{R}Q = J\bar{Q} + (\bar{K}\bar{Q})Q = J\bar{Q} + (\bar{K} + \bar{Q})Q = J\bar{Q} + \bar{K}Q, \text{ c'est-à-dire}$$

$$Q^+ = J\bar{Q} + \bar{K}Q \quad (3.56)$$

La condition (3.48) est identiquement nulle, car :

$$SR = J\bar{Q}KQ \equiv 0 \quad (3.57)$$

A partir de l'équation caractéristique (3.56) il est aisé de dresser les diverses tables d'états de la bascule  $JK$  (fig. 3.30 à 3.32).

No	J	K	Q	Q <sup>+</sup>
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Fig. 3.30

J	K	Q <sup>+</sup>
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

Fig. 3.31

Q	JK			
	00	01	11	10
0			1	1
1	1			1
				Q <sup>+</sup>

Fig. 3.32

### 3.3.7 Commentaire

La comparaison des tables d'états des bascules  $SR$  (fig. 3.18 et 3.20) et  $JK$  (fig. 3.30 et 3.32) démontre que la fonction de celle-ci est une solution de celle-là (au sens

de la définition du paragraphe 3.2.12). Il en découle la possibilité suivante : une bascule  $SR$  incluse dans un système logique quelconque peut être directement remplacée (en cas de panne, par exemple) par une bascule  $JK$  à condition de vérifier les relations  $S \equiv J$  et  $R \equiv K$ .

### 3.3.8 Définition : bascule $T$

La bascule  $T$  (en anglais *Trigger flip-flop*, c'est-à-dire bascule à déclenchement) découle d'une utilisation particulière de la bascule  $JK$  (fig. 3.29) caractérisée par la relation :

$$J = K = T \quad (3.58)$$

L'équation caractéristique se déduit immédiatement de l'expression (3.56) :

$$Q^+ = T\bar{Q} + \bar{T}Q = T \oplus Q \quad (3.59)$$

Les tables d'états sont celles des figures 3.33 à 3.35.

No	$T$	$Q$	$Q^+$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Fig. 3.33

$T$	$Q^+$
0	$Q$
1	$\bar{Q}$

Fig. 3.34

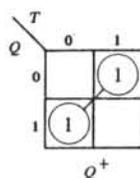


Fig. 3.35

### 3.3.9 Définition : diviseur de fréquence

On appelle généralement *diviseur de fréquence par deux* une bascule  $T$  vérifiant l'équation

$$J = K = T = 1 \quad (3.60)$$

L'équation caractéristique de la bascule  $T$  (3.59) devient dans ce cas :

$$Q^+ = \bar{Q} \quad (3.61)$$

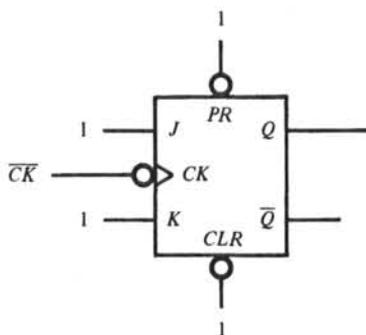


Fig. 3.36

Le système logique ainsi obtenu comporte une seule variable indépendante, le signal d'horloge  $\overline{CK}$  (fig. 3.36), et une sortie  $Q$ . La représentation de ces deux grandeurs en fonction du temps est l'objet du chronogramme de la figure 3.38; celui-ci décrit en fait le modèle quasi-synchrone du diviseur caractérisé notamment par la présence de certains retards (voir par exemple la figure 3.15); l'équation (3.61) est satisfaite lorsque  $Q$  et  $Q^+$  sont mesurés pendant deux instants d'horloge consécutifs ( $CK = 0$ , c'est-à-dire  $\overline{CK} = 1$ ).

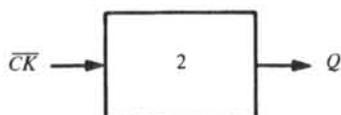


Fig. 3.37

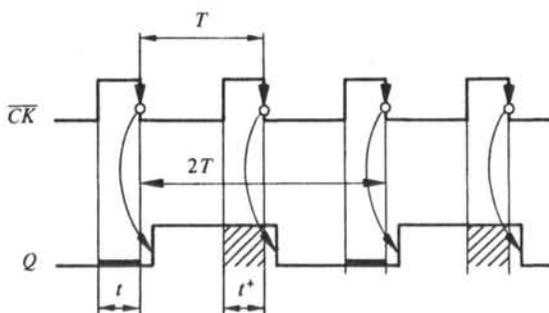


Fig. 3.38

### 3.3.10 Commentaire

Si le signal d'entrée  $\overline{CK}$  est périodique, de période  $T$ , on constate que le signal de sortie  $Q$  est également périodique, de période double  $2T$ . Le système logique étudié effectue donc une division par deux de la fréquence; on dit aussi qu'il compte par deux, puisqu'on remarque l'apparition d'une impulsion de la sortie  $Q$  ( $Q = 1$ ) pour deux impulsions de l'entrée  $\overline{CK}$  ( $\overline{CK} = 1$ ). Le schéma que nous adopterons par la suite est celui de la figure 3.37.

### 3.3.11 Conclusion

On a présenté dans cette section les bascules bistables qui sont le plus couramment utilisées dans la pratique; la recherche systématique de tous les types possibles de bascules comportant un nombre donné d'entrées d'excitation a fait l'objet de plusieurs travaux spécialisés [38] (pp. 75 - 91), [39], [41], [20] (pp. 227 - 229).

### 3.3.12 Exercice

Au sein d'un système logique quelconque, une bascule  $JK$  est défaillante. A quelles conditions est-il possible de la remplacer par une bascule  $SR$  si l'on vérifie les relations  $S \equiv J$  et  $R \equiv K$  ?

### 3.3.13 Exercice

En assemblant une bascule  $SR$  et deux portes ET réalisant les fonctions  $S = X \bar{V}$  et  $R = Y \bar{W}$  on définit une nouvelle bascule  $XYVW$  caractérisée par une certaine universalité d'emploi [35]; on demande d'établir l'équation caractéristique (avec une éventuelle condition) et la table d'états de cette bascule. Quelles sont les valeurs particulières qu'il faut attribuer aux entrées d'excitation  $X$ ,  $Y$ ,  $V$  et  $W$  pour obtenir une bascule  $JK$ ? une bascule  $T$ ? une bascule  $D$ ?

### 3.3.14 Exercice

Etant donné une bascule  $D$  on désire réaliser un système combinatoire permettant de la transformer en bascule  $JK$ , puis en diviseur de fréquence par deux. On demande ensuite la condition permettant de transformer une bascule  $JK$  en bascule  $D$ .

### 3.3.15 Exercice

On demande d'exprimer l'équation caractéristique d'une bascule  $\nu w$  réalisant la table d'états de la figure 3.39 [42]. Quel est le système combinatoire nécessaire pour transformer une bascule  $SR$  en une telle bascule?

$\nu$	$w$	$Q^+$
0	0	$Q$
0	1	0
1	0	1
1	1	$Q$

Fig. 3.39

### 3.3.16 Exercice

Une bascule  $D$  (fig. 3.22) est soumise à des variations de l'entrée d'excitation  $D$  et de l'entrée asynchrone  $\overline{CLR}$  décrites dans un chronogramme (fig. 3.40). En se référant au modèle quasi-synchrone de cette bascule et en supposant que  $PR = 0$ , on demande de tracer le chronogramme de la sortie  $Q$ .

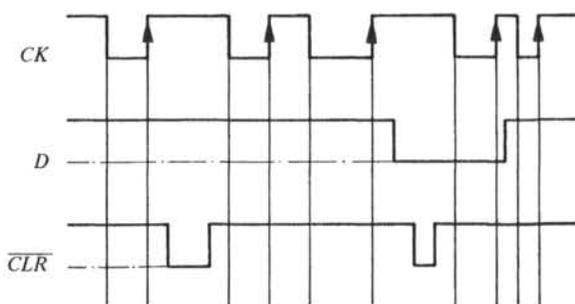


Fig. 3.40

### 3.3.17 Exercice

La table d'états de la bascule *SR* (fig. 3.20) définit quatre solutions (§ 3.2.12); deux d'entre elles ont déjà été analysées : les bascules *sr* (fig. 3.27) et *JK* (fig. 3.32); on demande de représenter les deux solutions restantes dans une table d'états, d'exprimer leur équation caractéristique et de les réaliser à l'aide d'une bascule *D* et d'un système combinatoire adéquat.

## 3.4 MODES DE REPRÉSENTATION SYNTHÉTIQUES

### 3.4.1 Définitions : transition

Tout couple  $Q, Q^+$  est appelé *transition* d'une bascule bistable (fig. 3.10). Il existe quatre transitions possibles qui peuvent être représentées par une *table des transitions* (fig. 3.41) ou par un *graphe des états* (fig. 3.42). Celui-ci est un graphe orienté qui comporte deux sommets (correspondant aux deux états possibles de la bascule :  $Q = 0$  et  $Q = 1$ ) et quatre flèches dont l'origine définit l'état présent ( $Q$ ) et l'extrémité l'état futur ( $Q^+$ ); chaque flèche représente donc l'une des quatre transitions  $Q, Q^+$ , numérotées de 0 à 3.

No de la transition	Transition $Q$ $Q^+$	Autre appellation [36] (pp. 33-37)
0	0   0	Maintien à 0 $\hat{=} \mu_0$
1	0   1	Enclenchement $\hat{=} \epsilon$
2	1   0	Déclenchement $\hat{=} \delta$
3	1   1	Maintien à 1 $\hat{=} \mu_1$

Fig. 3.41

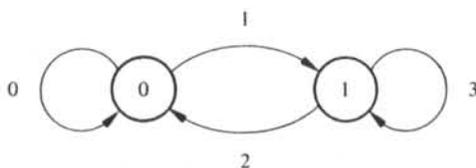


Fig. 3.42

### 3.4.2 Modes de représentation : bascule *SR*

La table des transitions de la bascule *SR* est obtenue en déterminant pour chaque transition  $Q, Q^+$  les valeurs des entrées d'excitation *S* et *R*. En partant de la table d'états (fig. 3.18) on élimine tout d'abord les états *S, R, Q* interdits (Nos 6 et 7 pour lesquels  $Q^+ = \emptyset$ ); les six états restants sont ensuite classés en quatre catégories correspondant chacune à une transition (fig. 3.43).

L'information contenue dans la figure 3.43 peut être condensée; l'examen de la transition  $Q, Q^+ = 00$ , par exemple, montre que :

No	No de la transition	$Q$ $Q^+$	$S$ $R$
0 } 2 }	0	0 0	{ 0 0 0 1
4 3	1	0 1	1 0
1 } 5 }	2	1 0	0 1
	3	1 1	{ 0 0 1 0

Fig. 3.43

No	$Q$ $Q^+$	$S$ $R$	$J$ $K$	$T$	$D$
0	0 0	0 $\emptyset$	0 $\emptyset$	0	0
1	0 1	1 0	1 $\emptyset$	1	1
2	1 0	0 1	$\emptyset$ 1	1	0
3	1 1	$\emptyset$ 0	$\emptyset$ 0	0	1

Fig. 3.44

- la valeur de  $S$  est égale à 0;
- la valeur de  $R$  peut être 0 ou 1, c'est-à-dire indifférente :  $R = \emptyset$ .

Il en découle la *table des transitions résumée* (fig. 3.44) dont le contenu peut être déterminé algébriquement. Dans l'exemple ci-dessus, on peut dire que la transition  $Q, Q^+ = 00$  est possible si  $S, R = 00$  OU si  $S, R = 01$ ; l'algèbre logique nous permet d'écrire une proposition  $P$  qui admet une simplification :

$$P = \bar{S}\bar{R} + \bar{S}R = \bar{S} \quad (3.62)$$

La forme simplifiée de  $P$  montre que la valeur de  $R$  est indifférente ( $R = \emptyset$ ) tandis que la valeur de  $S$  est égale à 0 ( $P = \bar{S}$ ).

En reportant dans la figure 3.42 les valeurs de  $S$  et  $R$  relatives à chaque transition on obtient la figure 3.45 qui est le graphe des états de la bascule  $SR$ .

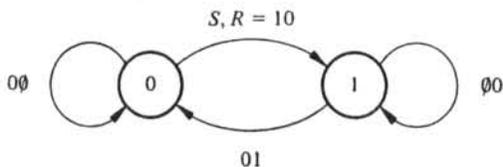


Fig. 3.45

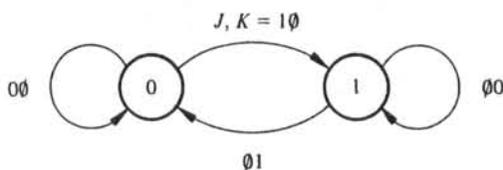


Fig. 3.46

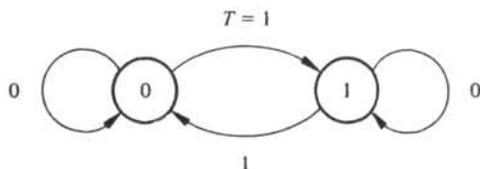


Fig. 3.47

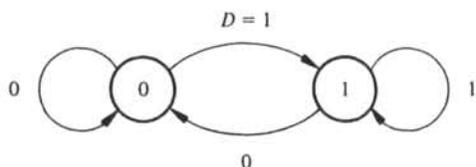


Fig. 3.48

### 3.4.3 Modes de représentation : autres bascules

En partant des tables d'états des bascules  $JK$  (fig. 3.30),  $T$  (fig. 3.33) et  $D$  (fig. 3.23) il est aisé de dresser les tables des transitions résumées de la figure 3.44. On remarque que les valeurs de  $T$  peuvent se déduire directement de celles de  $J$  et  $K$  si l'on vérifie la relation  $J = K = T$  (3.58), tandis que les valeurs de  $D$  se déduisent de celles de  $S$  et  $R$  en vérifiant l'expression  $S = \bar{R} = D$  (3.51). Les graphes correspondants sont ceux des figures 3.46 à 3.48; le cas particulier du diviseur de fréquence (§ 3.3.9 :  $T = 1$ ) est illustré par le graphe de la figure 3.49.

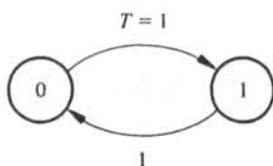


Fig. 3.49

### 3.4.4 Description

Il est intéressant d'établir une correspondance entre les graphes des états et les tables d'états; la comparaison des figures 3.45 et 3.50 (bascule  $SR$ ) montre que chaque sommet du graphe correspond à une ligne de la table et chaque flèche à une ou plusieurs cases. La simplification de la proposition logique  $P$  (3.62) peut alors être effectuée dans la table d'états par la méthode de Karnaugh.

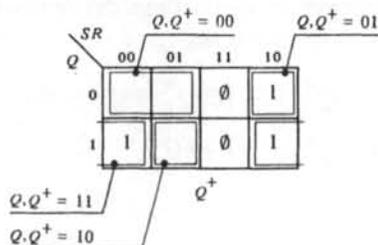


Fig. 3.50

### 3.4.5 Définition

La table des transitions et le graphe des états déterminent les fonctions (entrées d'excitation  $S$  et  $R$ , par exemple) nécessitées par un comportement souhaité (transition  $Q, Q^+$ ) : on dit qu'il s'agit de *modes de représentation synthétiques*.

#### □ 3.4.6 Mode de représentation : équations d'entrée

L'équation caractéristique de la bascule est l'expression algébrique de sa table d'états (voir le paragraphe 3.2.11 pour le cas de la bascule  $SR$ ). On peut aussi chercher une représentation analytique de la table des transitions, c'est-à-dire une expression algébrique des entrées d'excitation en fonction des variables  $Q$  et  $Q^+$ . On appellera *équations d'entrée* un tel mode de représentation.

#### □ 3.4.7 Exemple : bascule $D$

La forme particulièrement simple de l'équation caractéristique  $Q^+ = D$  (3.52) nous conduit directement à l'équation d'entrée de la bascule  $D$  :

$$D = Q^+ \quad (3.63)$$

La vérification de cette équation dans la table des transitions (fig. 3.44) est immédiate.

En général,  $Q^+$  est une fonction de plusieurs variables; quand cette fonction est incomplètement définie, on peut l'exprimer sous sa forme paramétrique (1.82) :

$$Q^+ = Q_m^+ + \lambda \cdot Q_M^+ \quad (3.64)$$

où  $Q_m^+$  est la borne inférieure de  $Q^+$ ,  $Q_M^+$  sa borne supérieure et  $\lambda$  la fonction universelle de toutes les variables de  $Q^+$  (§ 1.7.10). En reportant (3.64) dans (3.63) on obtient la nouvelle équation d'entrée :

$$D = Q_m^+ + \lambda \cdot Q_M^+ \quad (3.65)$$

dont l'expression (3.63) constitue le cas particulier valable lorsque  $Q^+$  est complètement définie, c'est-à-dire lorsque  $Q^+ = Q_m^+ = Q_M^+$ .

#### □ 3.4.8 Application : bascule $JK$

En examinant la table des transitions de la bascule  $JK$  (fig. 3.44) on constate que  $J$  et  $K$  sont des fonctions incomplètement définies des deux variables  $Q$  et  $Q^+$ ; grâce à la forme paramétrique (1.82) on peut écrire :

$$J = J_m + \lambda_J \cdot J_M = \bar{Q} Q^+ + \lambda_J (\bar{Q} Q^+ + Q) = \bar{Q} Q^+ + \lambda_J \cdot Q \quad (3.66)$$

$$K = K_m + \lambda_K \cdot K_M = Q \bar{Q}^+ + \lambda_K (Q \bar{Q}^+ + \bar{Q}) = Q \bar{Q}^+ + \lambda_K \cdot \bar{Q} \quad (3.67)$$

En remplaçant dans (3.66) et (3.67)  $Q^+$  par sa forme paramétrique (3.64) on obtient :

$$J = \bar{Q} (Q_m^+ + \lambda \cdot Q_M^+) + \lambda_J \cdot Q \quad (3.68)$$

$$K = Q(\overline{Q_m^+ + \lambda \cdot Q_M^+}) + \lambda_K \cdot \overline{Q} \quad (3.69)$$

La première propriété des fonctions incomplètement définies (voir le paragraphe 1.7.9) nous permet de poser :

$$\lambda = \lambda_J ; \quad \bar{\lambda} = \lambda_K \quad (3.70)$$

Les expressions (3.68) et (3.69) se transforment et se simplifient à l'aide notamment du théorème  $a + \bar{a}b = a + b$  (1.13) :

$$J = \overline{Q}Q_m^+ + \lambda_J \cdot \overline{Q}Q_M^+ + \lambda_J \cdot Q = \overline{Q}Q_m^+ + \lambda_J(Q + \overline{Q}Q_M^+) = \overline{Q}Q_m^+ + \lambda_J(Q + Q_M^+)$$

$$\begin{aligned} K &= Q(\overline{Q_m^+ \cdot \bar{\lambda}_K \cdot Q_M^+}) + \lambda_K \cdot \overline{Q} = Q(\overline{Q_m^+}[\lambda_K + \overline{Q_M^+}]) + \lambda_K \cdot \overline{Q} \\ &= \lambda_K \cdot Q\overline{Q_m^+} + Q\overline{Q_m^+}Q_M^+ + \lambda_K \cdot \overline{Q} = \lambda_K(\overline{Q} + Q\overline{Q_m^+}) + Q\overline{Q_m^+}Q_M^+ \\ &= \lambda_K(\overline{Q} + \overline{Q_m^+}) + Q(\overline{Q_m^+} + Q_M^+) \end{aligned}$$

Par définition  $Q_m^+ \leq Q_M^+$ , donc  $Q_m^+ + Q_M^+ = Q_M^+$  (fig. 1.68). On obtient enfin les équations d'entrée de la bascule JK :

$$J = \overline{Q}Q_m^+ + \lambda_J(Q + Q_M^+) \quad (3.71)$$

$$K = Q\overline{Q_M^+} + \lambda_K(\overline{Q} + \overline{Q_m^+}) \quad (3.72)$$

où  $\lambda_J$  et  $\lambda_K$  sont des fonctions universelles de toutes les variables.

Si la fonction  $Q^+$  est complètement définie, alors  $Q^+ = Q_m^+ = Q_M^+$  et les équations (3.71) et (3.72) se résument aux expressions (3.66) et (3.67).

#### □ 3.4.9 Conclusion

La recherche des équations d'entrée de la bascule JK est en fait une résolution de l'équation caractéristique (3.56) :

$$Q^+ = J\overline{Q} + \overline{K}Q$$

par rapport à chacune des variables  $J$  et  $K$ . Le problème général de la résolution d'une expression logique par rapport à l'une de ses variables n'entre pas dans le cadre de cet exposé : le lecteur est renvoyé au texte fondamental de Rudeanu [37] (pp. 341 - 411) ainsi qu'aux références [22] (§ 4.2.2), [20] (pp. 192 - 214), [38] (pp. 57 - 61), [34] (pp. 393 - 398) et [98] (pp. 193 - 214). Des résultats analogues ou voisins de ceux exposés dans les paragraphes 3.4.7 et 3.4.8 peuvent être trouvés dans les références [22] (§ 5.3.3), [34] (pp. 117 - 142), [38] (pp. 61 - 68), [20] (pp. 231 - 238) et [43] (pp. 112 - 127).

**3.4.10 Exercice**

Déterminer systématiquement les tables des transitions résumées des bascules suivantes en partant de leur table d'états (voir le paragraphe 3.4.4) :

- type  $XYVW$  (selon l'exercice du paragraphe 3.3.13);
- type  $\nu w$  (selon l'exercice du paragraphe 3.3.15);
- type  $LP$  donné par l'équation caractéristique :

$$Q^+ = LP + \bar{L}Q \quad (3.73)$$

- type  $RST$  donné par l'équation caractéristique [34] (pp. 129 - 132) :

$$Q^+ = S + T\bar{Q} + \bar{R}\bar{T}Q \quad (3.74)$$

avec la condition :

$$SR + RT + ST = 0 \quad (3.75)$$

- type  $xy$  donné par l'équation caractéristique :

$$Q^+ = x\bar{y} + \bar{y}Q \quad (3.76)$$

- type  $X$  donné par l'équation caractéristique :

$$Q^+ = XQ \quad (3.77)$$

**3.4.11 Exercice**

On demande d'établir et de comparer les tables des transitions résumées des quatre solutions de la bascule  $SR$  (selon l'exercice du paragraphe 3.3.17 et la figure 3.20).

**□ 3.4.12 Exercice**

Déterminer l'équation d'entrée de la bascule  $T$  en admettant que  $Q^+$  est une fonction incomplètement définie.

**□ 3.4.13 Exercice**

Déterminer les équations d'entrée de la bascule  $SR$  en admettant que  $Q^+$  est incomplètement définie; on examinera les deux cas suivants :

- pour tous les états  $\emptyset$  de  $Q^+$  on tolère que la condition  $SR = 0$  (3.48) ne soit pas vérifiée; on admet donc l'état  $S, R = 11$ ;
- pour tous les états  $\emptyset$  de  $Q^+$  on exclut les valeurs  $S = R = 1$  : on doit donc vérifier la condition  $SR = 0$ .

Montrer que les équations suivantes satisfont ce dernier cas [22] ( § 5.3.3) :

$$S = \bar{Q}Q_m^+ + \lambda_S \cdot \bar{\lambda}_R \cdot Q_M^+ \quad (3.78)$$

$$R = Q\bar{Q}_M^+ + \lambda_R \cdot \bar{\lambda}_S \cdot \bar{Q}_m^+ \quad (3.79)$$

où  $\lambda_R$  et  $\lambda_S$  sont deux fonctions universelles de toutes les variables.

□ **3.4.14 Exercice**

Calculer avec les équations caractéristiques et les équations d'entrée les systèmes combinatoires réalisant :

- une bascule  $D$  à l'aide d'une bascule  $JK$ ;
- une bascule  $LP$  (3.73) à l'aide d'une bascule  $JK$ ;
- une bascule  $JK$ , puis une bascule  $T$ , à l'aide d'une bascule  $SR$  (selon l'exercice du paragraphe 3.4.13);
- une bascule  $SR$  à l'aide d'une bascule  $T$  (selon l'exercice du paragraphe 3.4.12).



## ANALYSE ET SYNTHÈSE DES COMPTEURS

### 4.1 DIVISEURS DE FRÉQUENCE

#### 4.1.1 Définitions

On appelle *compteur quasi-synchrone* ou, plus brièvement, *compteur* tout assemblage d'une ou plusieurs bascules bistables et d'un système combinatoire comportant une seule variable indépendante : le signal d'horloge  $CK$ . La figure 4.1 représente le schéma d'un tel compteur réalisé dans le cas particulier avec des bascules  $JK$  (fig. 3.29).

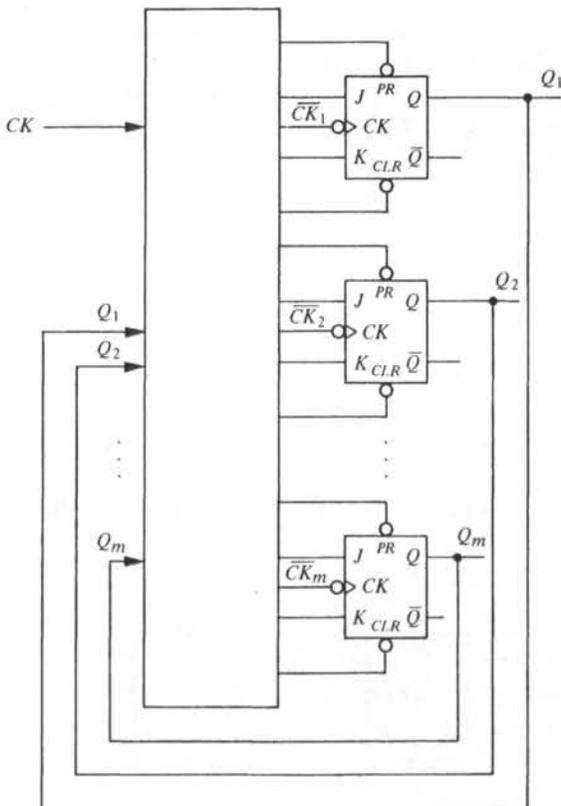


Fig. 4.1

Le *diviseur de fréquence* est un compteur dont toutes les bascules sont des diviseurs de fréquence par deux (voir les paragraphes 3.3.9 et 3.3.10).

Le *compteur synchrone* est un compteur dont toutes les bascules ont un fonctionnement normal (§ 3.2.6 :  $PR, CLR = 00$ ) et sont attaquées par le même signal d'horloge :  $\overline{CK} = \overline{CK}_1 = \overline{CK}_2 = \dots = \overline{CK}_m$ .

#### 4.1.2 Plan

Pour des raisons didactiques on traitera tout d'abord l'analyse et la synthèse des diviseurs de fréquence (sect. 4.1), puis des compteurs synchrones (sect. 4.2); l'analyse des compteurs quasi-synchrones fera l'objet de la section 4.3, tandis que la décomposition des compteurs, qui peut être étudiée indépendamment de leur définition, sera exposée dans la section 4.4.

#### 4.1.3 Hypothèses

On suppose dans ce chapitre que tout système combinatoire est décrit par son modèle combinatoire (§ 1.1.10) et que toute bascule bistable est décrite par son modèle quasi-synchrone (§ 3.2.3).

De plus, on admet que tous les retards attachés aux bascules et contenus dans le modèle quasi-synchrone ont des valeurs très voisines ou, en d'autres termes, que les différences entre ces valeurs sont négligeables; on admet aussi qu'aucun retard ne sépare les variations des sorties  $Q$  et  $\overline{Q}$  d'une bascule.

Dans toute situation où ces hypothèses ne sont pas vérifiées on doit recourir au modèle asynchrone du chapitre 7.

#### 4.1.4 Analyse

Le logigramme de la figure 4.2 est celui d'un diviseur de fréquence; chacune des bascules  $JK$  est en effet caractérisée par la relation  $J = K = 1$  (3.60). Les équations suivantes découlent de la relation (3.61) :

$$Q_1^+ = \overline{Q}_1 \quad ; \quad Q_2^+ = \overline{Q}_2 \quad ; \quad Q_4^+ = \overline{Q}_4 \quad ; \quad Q_8^+ = \overline{Q}_8 \quad (4.1)$$

Les signaux d'horloge des quatre bascules sont tous différents; d'après la figure 4.2 on a :

$$CK_1 = CK \quad ; \quad CK_2 = \overline{Q}_1 \quad ; \quad CK_4 = \overline{Q}_2 \quad ; \quad CK_8 = \overline{Q}_4 \quad (4.2)$$

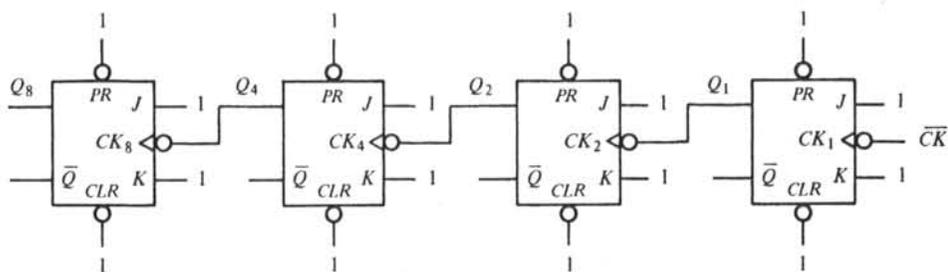


Fig. 4.2

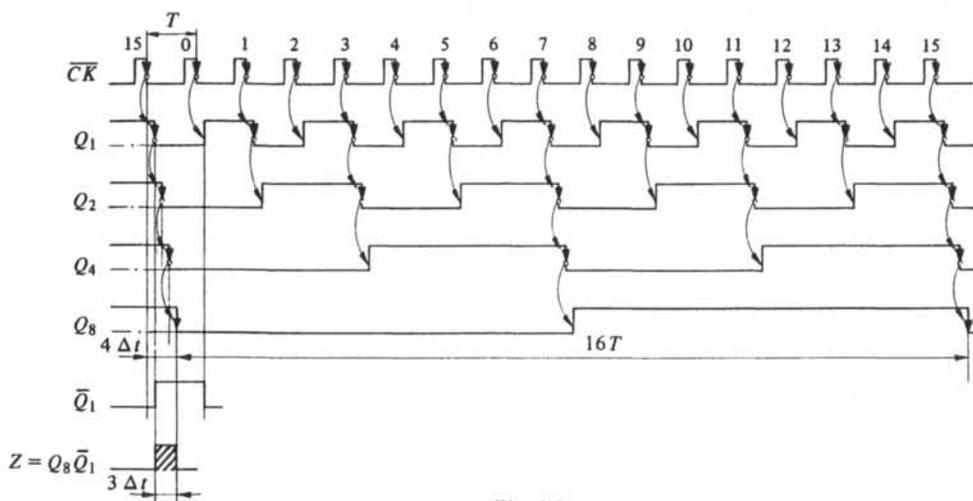


Fig. 4.3

#### 4.1.5 Mode de représentation : chronogramme

On considère la première bascule ( $Q_1$ ) de la figure 4.2 : c'est un diviseur de fréquence par deux dont l'entrée est  $\overline{CK}$  et la sortie  $Q_1$  ; le chronogramme de ce système a déjà été établi (fig. 3.38) : il est reporté dans la figure 4.3.

Un deuxième diviseur de fréquence par deux est constitué par la bascule dont l'entrée est  $\overline{CK}_2$  et la sortie  $Q_2$  ; à chaque descente du signal d'horloge  $\overline{CK}_2 = Q_1$  on obtient (après un certain retard) un changement de l'état  $Q_2$ .

L'établissement du chronogramme se poursuit de façon identique pour les deux bascules restantes : on en déduit les variations des états  $Q_4$  et  $Q_8$ .

#### 4.1.6 Définitions : états

Toute combinaison des valeurs des sorties  $Q_8, Q_4, Q_2$  et  $Q_1$  est un *état* (ou *état interne*) du diviseur et est symbolisée par  $(Q_8, Q_4, Q_2, Q_1)$ .

Dans le chronogramme de la figure 4.3 on distingue des retards séparant la descente d'un signal d'entrée ( $\overline{CK}$  par exemple) de la variation du signal de sortie ( $Q_1$  par exemple) ; la valeur du signal de sortie pendant ce retard est transitoire (§ 3.1.6). On appelle alors *état transitoire* tout état  $Q_8, Q_4, Q_2, Q_1$  dont l'une des sorties au moins a une valeur transitoire ; un *état permanent* ne comporte aucune sortie de ce type.

#### 4.1.7 Modes de représentation : graphes des états

Dans un tel graphe (§ 3.4.1) on représente chaque état par un sommet et chaque passage d'un état à l'état suivant par une flèche ; la lecture des états  $Q_8, Q_4, Q_2, Q_1$  est faite dans le chronogramme du diviseur (fig. 4.3) et l'on distingue dans le graphe (fig. 4.4) les états permanents (trait fort) des états transitoires (trait fin). En examinant la figure 4.4 on constate ce qui suit :

- le graphe est refermé sur lui-même ; il traduit le comportement périodique du diviseur ;

- un même état peut apparaître plusieurs fois (l'état  $Q_8, Q_4, Q_2, Q_1 = 0000$ , par exemple, apparaît une fois comme état permanent et trois fois comme état transitoire);
- en passant d'un état au suivant, une seule sortie ( $Q_8, Q_4, Q_2$  ou  $Q_1$ ) change de valeur.

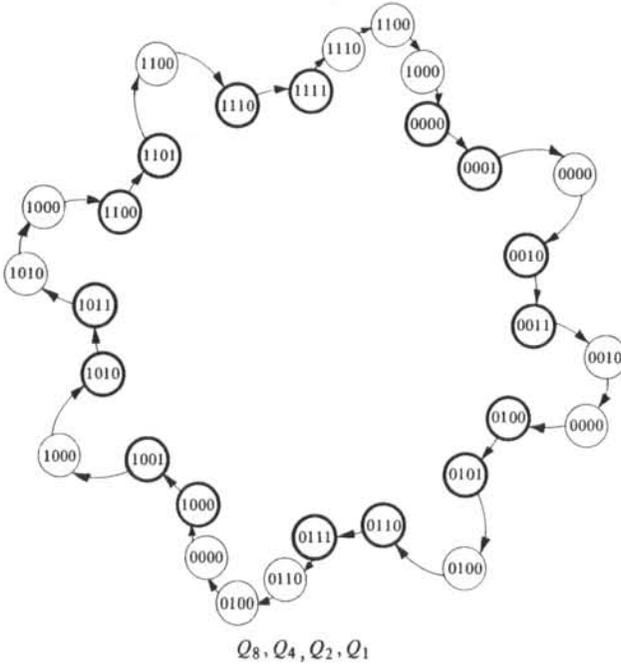


Fig. 4.4

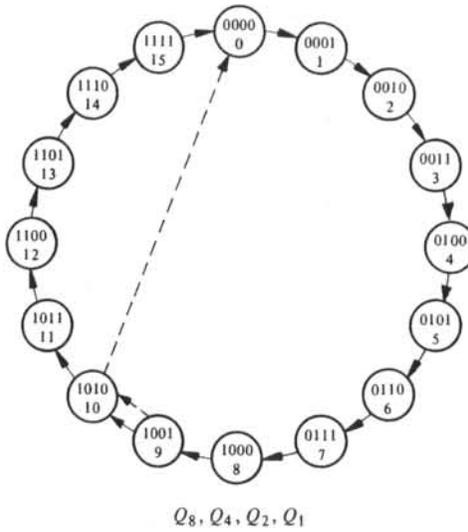


Fig. 4.5

En supprimant la représentation des états transitoires on obtient le nouveau graphe de la figure 4.5; on a donné dans chaque sommet le numéro décimal de l'état  $Q_8, Q_4, Q_2, Q_1$ . Dans ce graphe, un même état n'apparaît qu'une fois; on constate alors la présence des seize états possibles  $Q_8, Q_4, Q_2, Q_1$ .

#### 4.1.8 Définitions

La liste ordonnée des états permanents d'un compteur est la *séquence* ou le *code* de ce compteur. Le nombre des états permanents est la *mesure*  $p$ ; le *rappor de division* est la période du signal de sortie divisée par la période du signal d'entrée : ce rapport est généralement égal à la mesure  $p$ .

Le graphe des états de la figure 4.5 donne la séquence du diviseur de fréquence (fig. 4.2) qui peut être décrite symboliquement par :

$$Q_8, Q_4, Q_2, Q_1 = 0000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 0101 \rightarrow 0110 \rightarrow 0111 \rightarrow \\ \rightarrow 1000 \rightarrow 1001 \rightarrow 1010 \rightarrow 1011 \rightarrow 1100 \rightarrow 1101 \rightarrow 1110 \rightarrow 1111 \rightarrow \\ \rightarrow 0000$$

et qui est identique au code binaire pur des nombres décimaux compris entre 0 et 15 (§ 8.1.6). La mesure de ce diviseur est égale à 16; on dit aussi qu'on a un diviseur de fréquence par 16 ou un compteur par 16.

#### 4.1.9 Description

Un système combinatoire est défini par l'expression algébrique :

$$Z = Q_8 \bar{Q}_1 \quad (4.3)$$

Une représentation partielle de  $Z$  dans le chronogramme de la figure 4.3 fait apparaître une valeur transitoire  $Z = 1$  due au retard de  $Q_8$  par rapport à  $\bar{Q}_1$ ; la durée de ce régime transitoire est égale à la somme des retards des trois bascules  $Q_2, Q_4$  et  $Q_8$  ( $3 \Delta t$ ).

La structure du diviseur de fréquence entraîne un décalage dans le temps des variations des sorties  $Q_1, Q_2, Q_4$  et  $Q_8$  : il en découle des régimes transitoires qui peuvent être indésirables [54] (pp. 243 - 246).

#### 4.1.10 Conclusion

Le logigramme du diviseur que nous avons analysé (fig. 4.2) est caractérisé par une très grande simplicité (aucun système combinatoire n'est nécessaire) et par une structure itérative (§ 2.7.5). Son comportement (fig. 4.3 à 4.5) peut être défini par un code (binaire pur) et par une mesure ( $p = 16$ ).

De façon générale, on constate que la mise en série de  $m$  bascules (selon la disposition de la figure 4.2) produit un diviseur de fréquence dont le code est binaire pur et la mesure égale à  $2^m$ . Pour obtenir un code quelconque, il faut généralement réaliser un système combinatoire assurant le codage désiré; pour avoir une mesure quelconque, on doit appliquer une méthode de synthèse semblable à celle du paragraphe 4.1.15.

On remarque qu'il s'écoule un certain temps entre la descente du signal d'horloge  $\bar{CK}$  et l'apparition d'un état permanent; dans le pire des cas, une durée égale à la somme des retards de quatre bascules ( $4 \Delta t$ ) sépare la descente de  $\bar{CK}$  de celle de  $Q_8$  (fig. 4.3). Pour un diviseur par  $2^m$ , cette durée sera celle de la somme des délais de  $m$  bascules

( $m \Delta t$ ). La fréquence maximale de fonctionnement (ou la période minimale  $T$ ) des diviseurs est donc limitée non seulement par le retard de chaque bascule, mais également par leur nombre [44] (pp. 161 - 167), [54] (pp. 243 - 244).

#### 4.1.11 Exercice

Déterminer un logigramme équivalent à celui de la figure 4.2 et comportant uniquement des bascules  $D$  (§ 3.3.1 et 3.3.2). A quelles conditions les bascules  $D$  de ce nouveau logigramme, de même que les bascules  $JK$  de la figure 4.2, valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ?

#### 4.1.12 Exercice

Représenter dans le chronogramme de la figure 4.3 et dans le graphe de la figure 4.4 les fonctions suivantes :

$$Z_1 = \bar{Q}_2 \bar{Q}_1 ; Z_2 = Q_8 \bar{Q}_1 ; Z_3 = Q_8 \bar{Q}_4 \bar{Q}_2 \bar{Q}_1$$

Mettre en évidence les régimes transitoires de ces trois fonctions (§ 4.1.9).

#### 4.1.13 Exercice

Analyser un diviseur de fréquence obtenu à partir de la figure 4.2 en imposant les relations :

$$CK_1 = CK ; CK_2 = Q_1 ; CK_4 = Q_2 ; CK_8 = Q_4$$

Établir le chronogramme et les graphes des états correspondants, puis déterminer le code de ce compteur.

#### 4.1.14 Exercice

Déterminer le chronogramme et les graphes des états des diviseurs de fréquence définis par les figures 4.6 et 4.7. Montrer que dans le premier cas (fig. 4.6) le nombre des états permanents n'est pas égal au rapport de division (§ 4.1.8); représenter dans le second cas (fig. 4.7) la valeur de la sortie  $T$ .

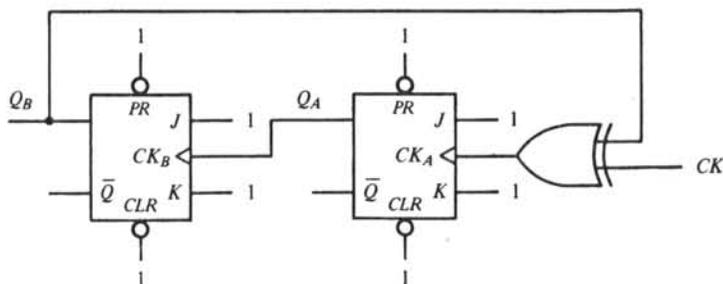


Fig. 4.6

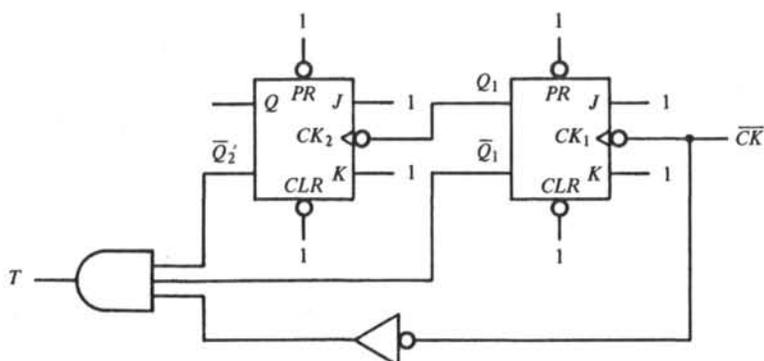


Fig. 4.7

#### 4.1.15 Synthèse : méthode

Pour déterminer le logigramme d'un diviseur de fréquence dont le code est binaire pur et dont la mesure  $p$  n'est pas une puissance de deux, on procède de la façon suivante [54] (pp. 246 - 248) :

- on calcule le nombre  $m$  de bascules grâce à la relation :

$$2^{m-1} < p < 2^m \quad (4.4)$$

- les  $m$  bascules sont assemblées selon la figure 4.2 et réalisent un diviseur par  $2^m$  ;
- les  $p$  états permanents du diviseur par  $p$  portent les numéros décimaux  $0, 1, \dots, p-1$  : ils se succèdent dans cet ordre; le premier état indésirable rencontré est donc l'état  $p$  : il est détecté à l'aide d'une fonction  $Z_p$  ;
- on réalise l'expression algébrique de  $Z_p$ , en la simplifiant au besoin ;
- la remise à 0 de toutes les bascules ( $PR, CLR = 01$  : § 3.2.6) est assurée par la fonction  $Z_p$  ; on regagne ainsi l'état initial du diviseur (état 0) sitôt après avoir détecté l'état  $p$ .

#### 4.1.16 Définition : état $\emptyset$

Un diviseur de fréquence par  $p$  comporte  $p$  états permanents qui constituent la séquence de comptage et  $2^m - p$  états superflus; par analogie avec les définitions des fonctions incomplètement définies (§ 1.7.2) ceux-ci sont appelés les *états  $\emptyset$*  du diviseur.

#### 4.1.17 Synthèse : exemple

On désire réaliser un diviseur par 10. La relation (4.4) devient dans ce cas :

$$2^3 < 10 < 2^4 \quad (4.5)$$

On doit donc assembler quatre bascules selon le logigramme de la figure 4.2; elles définissent  $2^4 = 16$  états  $Q_8, Q_4, Q_2, Q_1$  (No 0 à 15 : fig. 4.5). Les dix états du diviseur sont

numérotés de 0 à 9; l'état  $Q_8, Q_4, Q_2, Q_1 = 1010$  (No 10) est détecté par la fonction  $Z_p = Z_{10}$  dont la forme canonique algébrique s'écrit :

$$Z_{10} = Q_8 \bar{Q}_4 Q_2 \bar{Q}_1 \quad (4.6)$$

En attaquant les entrées asynchrones  $CLR$  des quatre bascules par la variable  $Z_{10}$ , on quitte l'état 1010 sitôt après l'avoir atteint et on gagne l'état initial 0000; ce comportement est illustré en trait discontinu dans le graphe des états (fig. 4.5). L'examen de celui-ci montre que les cinq états 1011, 1100, 1101, 1110 et 1111 sont des états  $\emptyset$  et peuvent contribuer à la simplification de la fonction  $Z_{10}$ . Une table de Karnaugh (fig. 4.8) donne alors :

$$Z_{10} = Q_8 Q_2 \quad (4.7)$$

Le logigramme final (fig. 4.9) réalise cette fonction à l'aide d'une porte NAND :

$$\bar{Z}_{10} = \bar{Q}_8 \bar{Q}_2 = \overline{CLR}_8 = \overline{CLR}_4 = \overline{CLR}_2 = \overline{CLR}_1 \quad (4.8)$$

	$Q_8 Q_4$			
	00	01	11	10
$Q_2 Q_1$	00		$\emptyset$	
	01		$\emptyset$	
	11		$\emptyset$	$\emptyset$
	10		$\emptyset$	1

$Z_{10}$

Fig. 4.8

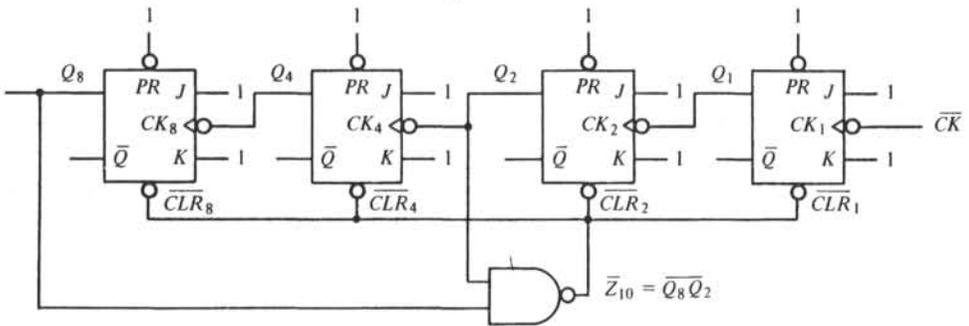


Fig. 4.9

#### 4.1.18 Analyse

L'analyse du logigramme de la figure 4.9 nous permet de déterminer les états transitoires du diviseur par 10 ainsi que le comportement de celui-ci à partir des états  $\emptyset$ . Les équations sont celles du diviseur par 16 (4.1) (4.2) auxquelles s'ajoute l'expression de la fonction  $Z_{10}$  (4.8).

#### 4.1.19 Mode de représentation : chronogramme

En partant de l'état initial  $Q_8, Q_4, Q_2, Q_1 = 0000$ , il est aisé de retrouver la séquence des dix états permanents  $0000 \rightarrow 0001 \rightarrow \dots \rightarrow 1001$  (§ 4.1.5) qui sont représentés par un chronogramme (fig. 4.10) identique à celui de la figure 4.3. On vérifie que dans l'intervalle de temps  $\tau$  la variable  $Z_{10}$  vaut toujours 0, dans les états permanents comme dans les états transitoires.

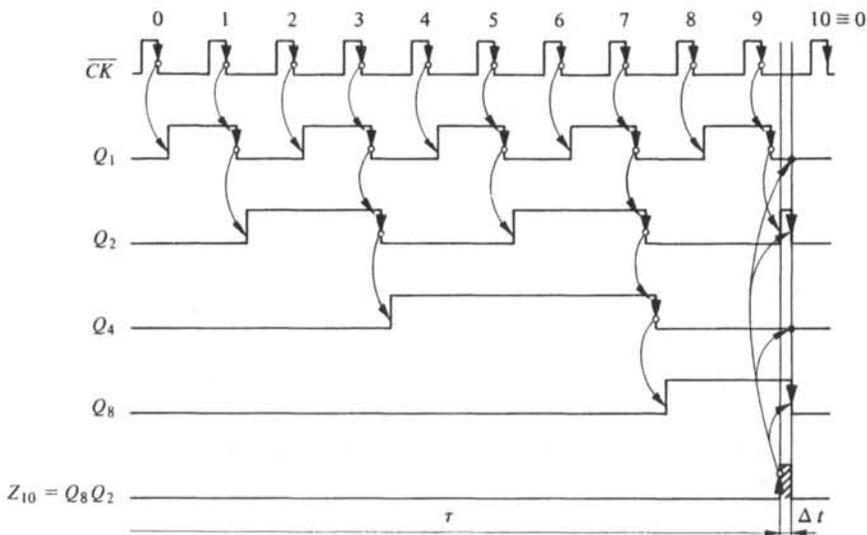


Fig. 4.10

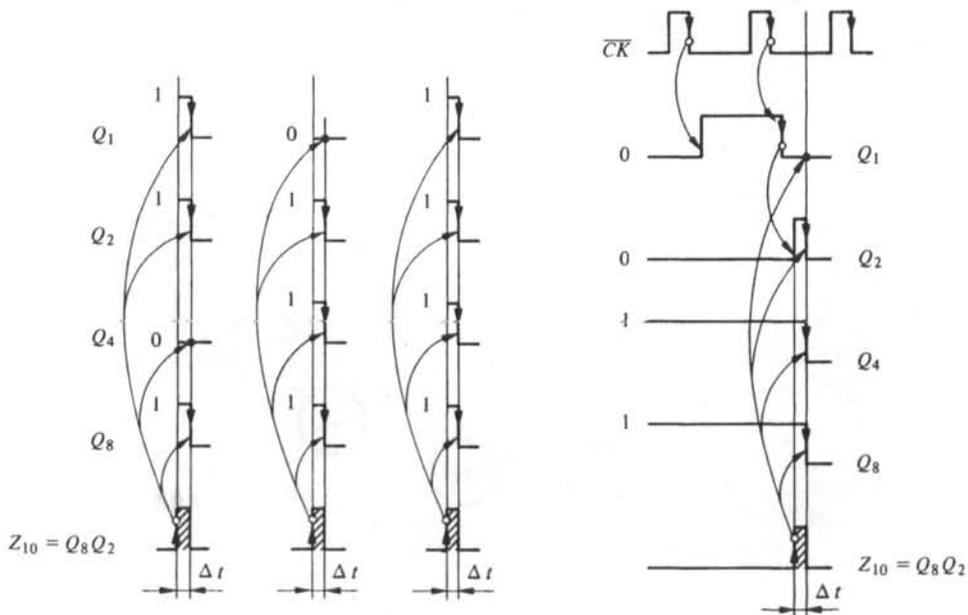


Fig. 4.11

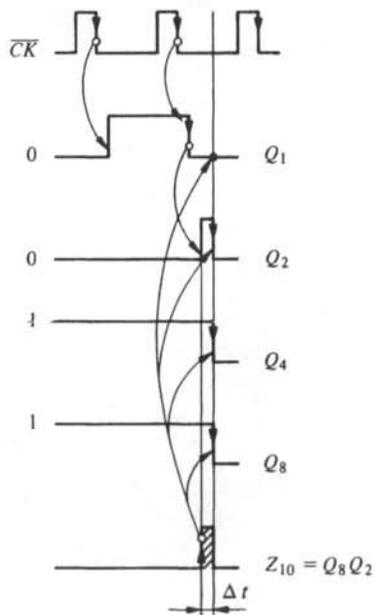


Fig. 4.12

Sitôt que l'état 1010 est atteint on a  $Z_{10} = Q_8 Q_2 = 1$ ; les entrées asynchrones  $CLR_8 = CLR_4 = CLR_2 = CLR_1 = Z_{10}$  sont donc égales à 1 et, après un certain retard  $\Delta t$  (§ 3.2.9 et § 4.1.3), elles imposent une remise à zéro de toutes les bascules : on atteint l'état initial 0000.

L'examen de la table de Karnaugh de  $Z_{10}$  (fig. 4.8) montre que cette fonction vaut 1 dans les trois états  $\emptyset$  suivants :

$$Q_8, Q_4, Q_2, Q_1 = 1011; 1110; 1111$$

Si le diviseur se trouve dans l'un de ces états,  $Z_{10}$  prend la valeur 1 et entraîne une remise à zéro des quatre bascules après un retard  $\Delta t$ ; ces trois états  $\emptyset$  sont donc transitoires : le diviseur regagne l'état initial 0000 (fig. 4.11).

Dans les deux états  $\emptyset$  restants caractérisés par :

$$Q_8, Q_4, Q_2, Q_1 = 1100; 1101$$

la fonction  $Z_{10}$  est égale à 0 (fig. 4.8). Le chronogramme de la figure 4.12 détaille alors le comportement du diviseur et fait apparaître une séquence de deux états permanents 1100  $\rightarrow$  1101 précédant le retour à l'état initial 0000.

#### 4.1.20 Modes de représentation : graphes des états

Le graphe des états de la figure 4.13 se déduit des chronogrammes des figures 4.10, 4.11 et 4.12; la valeur de  $Z_{10}$  est reportée sur chaque flèche et les états transitoires sont représentés en trait fin.

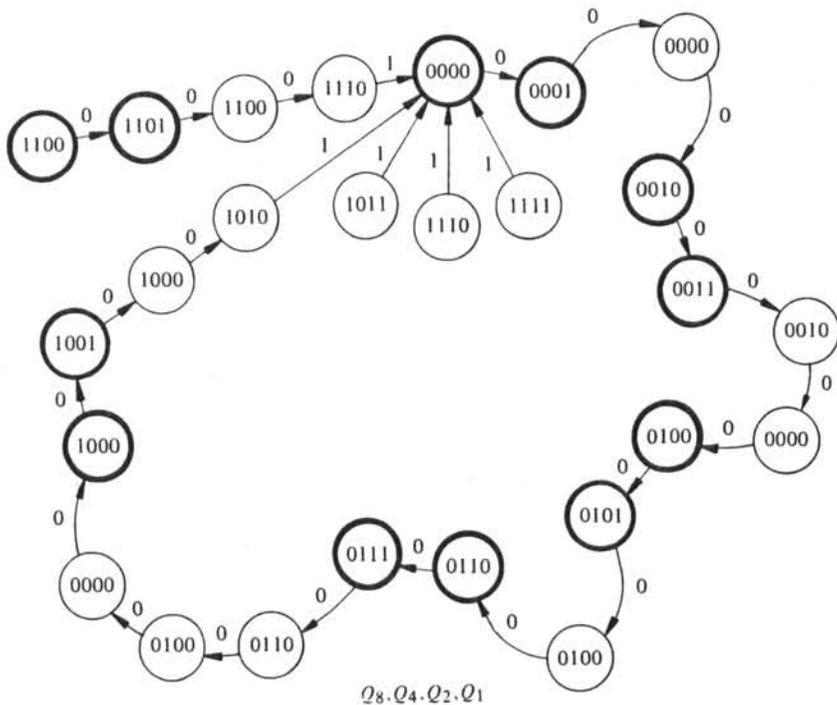


Fig. 4.13

On constate qu'à partir de tout état  $\emptyset$  le diviseur rejoint sa séquence après deux périodes du signal d'horloge au plus.

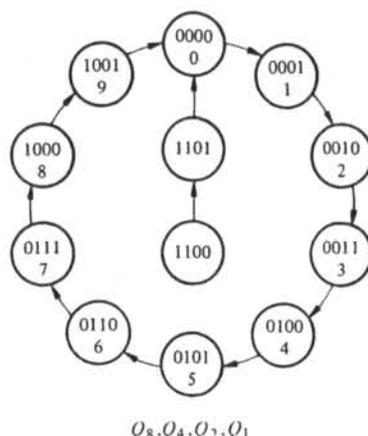


Fig. 4.14

#### 4.1.21 Commentaire

La variable  $Z_{10}$  est égale à un 1 pendant un intervalle de temps  $\Delta t$  (fig. 4.10) qui est le retard des bascules (§ 3.2.9); dans certains cas, cette durée  $\Delta t$  peut être trop brève pour assurer une remise à zéro correcte du diviseur. On peut alors mémoriser la valeur  $Z_{10} = 1$  dans un élément de mémoire  $\bar{s}\bar{r}$  (sect. 3.1) dont la remise à zéro est effectuée par le signal d'horloge : le logigramme correspondant est présenté dans les références [28] (p. 407) et [54] (pp. 247 - 249) ainsi que dans l'exercice du paragraphe 7.5.9 (fig. 7.63).

#### 4.1.22 Exercice

A quelles conditions les bascules  $JK$  de la figure 4.9 valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ?

#### ■ 4.1.23 Exercice

En analysant le chronogramme (fig. 4.10) du diviseur par 10 (fig. 4.9) démontrer que la remise à zéro de la première bascule est superflue et qu'on peut ainsi remplacer la relation  $\overline{CLR}_1 = \bar{Z}_{10}$  par  $\overline{CLR}_1 = 1$ ; modifier en conséquence les chronogrammes des figures 4.10 à 4.12 ainsi que le graphe de la figure 4.13, puis montrer qu'une simplification identique est impossible pour toute autre bascule.

#### 4.1.24 Exercice

Effectuer la synthèse d'un diviseur de fréquence par 10 selon le code Excédent 3 (fig. 8.1) : les dix états se succèdent de 0011 à 1100. Déterminer et simplifier la fonction de remise à zéro  $Z$ , rechercher le nombre minimal des entrées asynchrones  $PR$  et/ou  $CLR$  qu'il faut évaluer à la variable  $Z$ , tracer le chronogramme et les graphes pour tous les états du diviseur, y compris les états  $\emptyset$ .

## 4.1.25 Exercice

Déterminer le chronogramme et les graphes des états de chacun des trois diviseurs représentés par les figures 4.15 à 4.17. Quels sont le code et la mesure de ces compteurs ? La mesure est-elle toujours égale au rapport de division ?

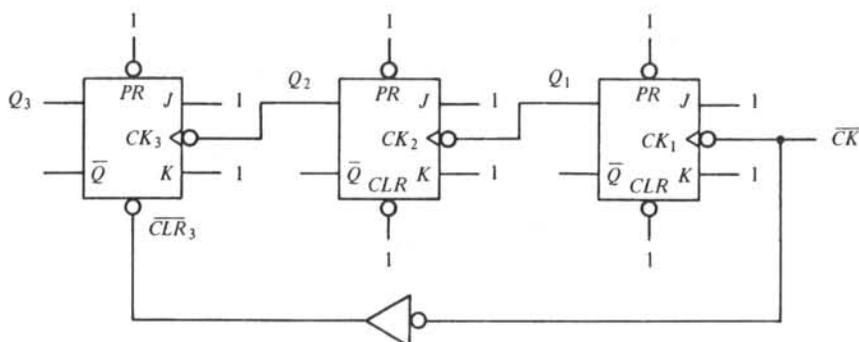


Fig. 4.15

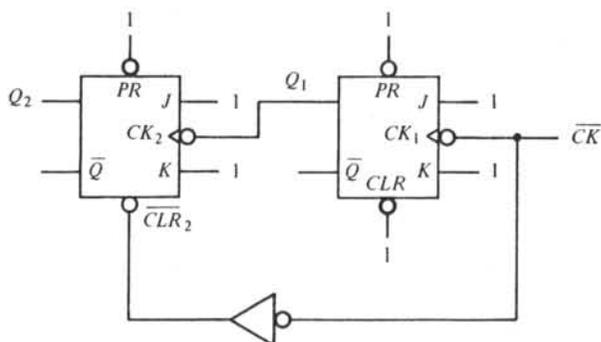


Fig. 4.16

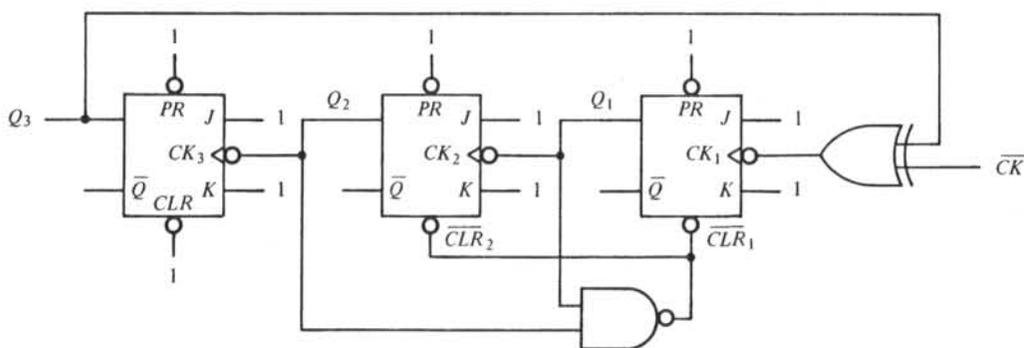


Fig. 4.17

## 4.1.26 Exercice

A partir d'un oscillateur délivrant un signal d'horloge  $\overline{CK}$  de période  $T = 1$  seconde on désire réaliser une horloge digitale comptant et affichant les secondes, les minutes et

les heures; cette horloge peut être décomposée en six diviseurs de fréquence (fig. 4.18) caractérisés chacun par une mesure  $p$  et un code binaire pur. Les diviseurs par 10 (No 1, 3 et 5) sont semblables à celui de la figure 4.9; les diviseurs par 6 (No 2 et 4) seront calculés par la méthode du paragraphe 4.1.15 tandis que le diviseur par 4 (No 6) se déduit directement du logigramme de la figure 4.2. L'interconnexion des diviseurs No 1, 2, 3, 4 et 5 se fait en examinant leurs chronogrammes respectifs : on tentera de trouver une structure itérative. L'interconnexion directe des compteurs No 5 et 6 produirait une division de fréquence par 40; on doit donc prévoir une rétroaction supplémentaire (fonction logique  $Z_{24}$ ) assurant la remise à zéro sitôt qu'on atteint 24 heures. Le diviseur par 24 sera l'objet d'une analyse (chronogramme et graphes des états) décrivant son comportement à partir des états  $\emptyset$  et détaillant l'apparition de tous les états transitoires.

## 4.2 COMPTEURS SYNCHRONES

### 4.2.1 Hypothèses

La définition des compteurs synchrones (§ 4.1.1) impose à ceux-ci un fonctionnement normal ( $PR = CLR = 0$ ) et l'existence d'un seul signal d'horloge  $CK$  commun à toutes les bascules; il en découle que les instants d'horloge, définis par  $CK = 0$ , sont également communs à toutes les bascules. Les hypothèses du paragraphe 4.1.3 peuvent être modifiées : le modèle quasi-synchrone des bascules bistables est remplacé par le modèle synchrone (§ 3.2.10).

Dans les logigrammes des compteurs synchrones on néglige en général la représentation des entrées asynchrones ( $PR, CLR$ ) puisque celles-ci sont sans effet ( $PR = CLR = 0$ ).

### 4.2.2 Définitions : états

Si un compteur synchrone comporte  $m$  bascules on appellera *état* (ou *état interne*) de ce compteur toute combinaison des valeurs des sorties  $Q_1, Q_2, \dots, Q_m$  définies pendant un instant d'horloge ( $CK = 0$ ); on distinguera les *états présents*, mesurés à l'instant présent  $t$  et symbolisés par  $(Q_1, Q_2, \dots, Q_m)$ , des *états futurs*, mesurés à l'instant futur  $t^+$  et symbolisés par  $(Q_1^+, Q_2^+, \dots, Q_m^+)$ .

### 4.2.3 Propriétés

Les propriétés suivantes des compteurs synchrones découlent de celles des bascules bistables (§ 3.2.8) :

- tout état  $Q_1, Q_2, \dots, Q_m$  est constant pendant un instant d'horloge ( $CK = 0$ ); un tel état est donc permanent (selon la définition du paragraphe 4.1.6);
- toute variation d'un état  $Q_1, Q_2, \dots, Q_m$  ne peut se produire qu'immédiatement après une montée du signal d'horloge  $CK$ .

### 4.2.4 Synthèse

L'objectif de la synthèse est la réalisation des fonctions d'excitation ( $J$  et  $K$  dans l'exemple de la figure 4.1) à partir d'un code imposé.

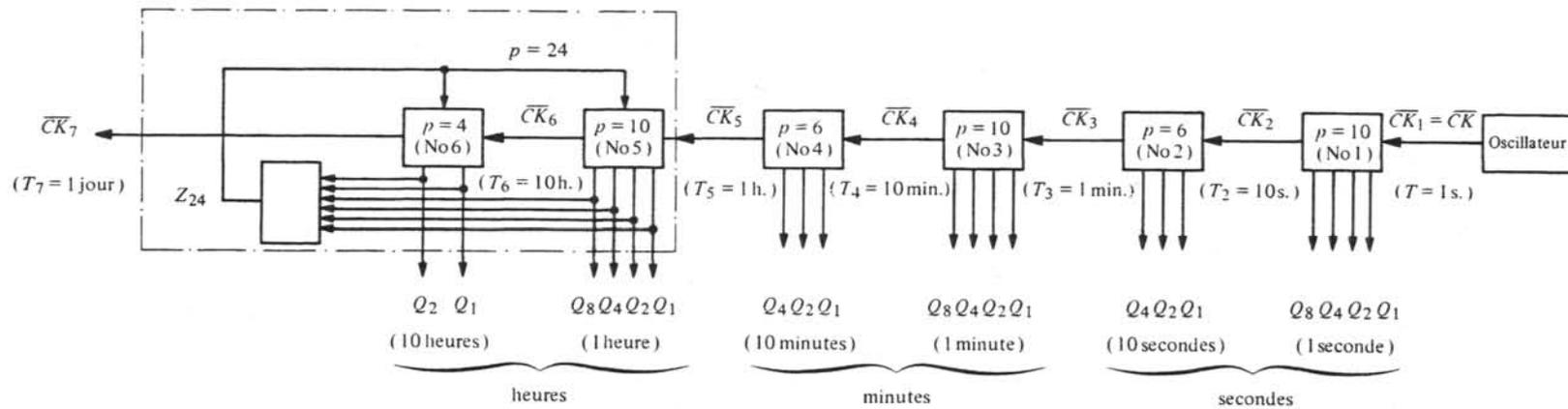


Fig. 4.18

Dans certains cas, on donne uniquement la mesure  $p$  du compteur désiré : le code peut être quelconque.

#### 4.2.5 Méthode

Les états du compteur synchrone se succèdent selon le code choisi. Pour chaque instant d'horloge on procède de la façon suivante :

- on relève l'état présent  $Q_1, Q_2, \dots, Q_m$  et l'état futur  $Q_1^+, Q_2^+, \dots, Q_m^+$ ;
- pour chaque bascule  $Q_j$  ( $j = 1, 2, \dots, m$ ) on obtient une transition  $Q_j, Q_j^+$ ;
- l'examen de la table des transitions (fig. 3.44) fournit les valeurs des fonctions d'excitation ( $J_j$  et  $K_j$  par exemple);
- les fonctions d'excitation, obtenues finalement sous la forme d'une table de vérité, peuvent être réalisées par les méthodes des systèmes combinatoires.

#### 4.2.6 Exemple

Un compteur synchrone comporte dix états  $Q_8, Q_4, Q_2, Q_1$  dans le code binaire pur (fig. 8.1); on désire réaliser deux variantes de ce compteur : l'une avec des bascules  $JK$ , l'autre avec des bascules  $D$ .

#### 4.2.7 Modes de représentation : graphe des états et table d'états

Le graphe des états de la figure 4.19 illustre le code imposé; on en déduit la table d'états (§ 3.2.11) de la figure 4.20 en procédant comme suit :

- seize lignes représentent les  $2^4 = 16$  états  $Q_8, Q_4, Q_2, Q_1$ ; ceux-ci sont toujours donnés dans le code binaire pur et affectés du numéro correspondant (colonne  $t$ );
- pour les dix états présents de la séquence désirée (No 0 à 9) on reporte dans la colonne  $t^+$  les dix états futurs correspondants tels qu'ils sont lus dans le graphe (fig. 4.19); l'état présent 1001, par exemple, est suivi de l'état 0000 : on reporte donc dans la ligne No 9 ( $Q_8, Q_4, Q_2, Q_1 = 1001$ ) l'état futur  $Q_8^+, Q_4^+, Q_2^+, Q_1^+ = 0000$ ;
- on constate qu'il existe six états qui n'apparaissent pas dans le code imposé : ce sont des états  $\emptyset$  (No 10, 11, ..., 15); les états futurs correspondants sont indifférents et l'on reporte le symbole  $\emptyset$  pour toutes les valeurs  $Q_8^+, Q_4^+, Q_2^+$  et  $Q_1^+$ .

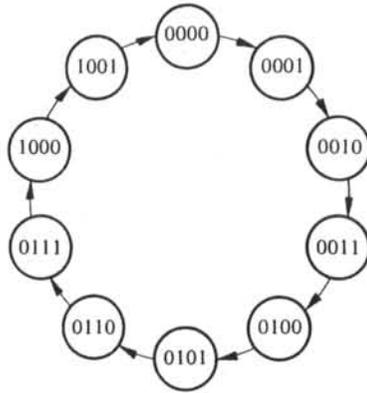
#### 4.2.8 Calcul tabulaire : bascules $JK$

On considère tout d'abord la bascule  $Q_8$ ; pour chacune des dix premières lignes de la table d'états (No 0 à 9 dans la figure 4.20) on détermine une transition  $Q_8, Q_8^+$ ; chacune de celles-ci donne un couple de valeurs  $J_8, K_8$  dans la table des transitions de la bascule  $JK$  (fig. 3.44). Les six états  $\emptyset$  se satisfont de transitions quelconques :  $J_8$  et  $K_8$  prennent la valeur  $\emptyset$  dans les six lignes correspondantes (No 10 à 15).

Il résulte de ce calcul la table de vérité des fonctions d'excitation  $J_8$  et  $K_8$  dont la forme canonique décimale est :

$$J_8(Q_8, Q_4, Q_2, Q_1) = \Sigma 7 + \emptyset 8, 9, \dots, 15 \quad (4.9)$$

$$K_8(Q_8, Q_4, Q_2, Q_1) = \Sigma 9 + \emptyset 0, 1, \dots, 7, 10, 11, \dots, 15 \quad (4.10)$$



$Q_8, Q_4, Q_2, Q_1$

Fig. 4.19

No	$t$ $Q_8 Q_4 Q_2 Q_1$	$t^+$ $Q_8^+ Q_4^+ Q_2^+ Q_1^+$	$Q_8 Q_8^+$	$J_8 K_8$	$J_4 K_4$	$J_2 K_2$	$J_1 K_1$
0	0 0 0 0	0 0 0 1	0 0	0 0	0 0	0 0	1 0
1	0 0 0 1	0 0 1 0	0 0	0 0	0 0	1 0	0 1
2	0 0 1 0	0 0 1 1	0 0	0 0	0 0	0 0	1 0
3	0 0 1 1	0 1 0 0	0 0	0 0	1 0	0 1	0 1
4	0 1 0 0	0 1 0 1	0 0	0 0	0 0	0 0	1 0
5	0 1 0 1	0 1 1 0	0 0	0 0	0 0	1 0	0 1
6	0 1 1 0	0 1 1 1	0 0	0 0	0 0	0 0	1 0
7	0 1 1 1	1 0 0 0	0 1	1 0	0 1	0 1	0 1
8	1 0 0 0	1 0 0 1	1 1	0 0	0 0	0 0	1 0
9	1 0 0 1	0 0 0 0	1 0	0 1	0 0	0 0	0 1
10	1 0 1 0	0 0 0 0	1 0	0 0	0 0	0 0	0 0
11	1 0 1 1	0 0 0 0	1 0	0 0	0 0	0 0	0 0
12	1 1 0 0	0 0 0 0	1 0	0 0	0 0	0 0	0 0
13	1 1 0 1	0 0 0 0	1 0	0 0	0 0	0 0	0 0
14	1 1 1 0	0 0 0 0	1 0	0 0	0 0	0 0	0 0
15	1 1 1 1	0 0 0 0	1 0	0 0	0 0	0 0	0 0

Fig. 4.20

En répétant le même calcul pour les trois bascules restantes on obtient les formes canoniques décimales des autres fonctions d'excitation :

$$J_4 = \Sigma 3 + \emptyset 4, 5, 6, 7, 10, 11, \dots, 15 \quad (4.11)$$

$$K_4 = \Sigma 7 + \emptyset 0, 1, 2, 3, 8, 9, \dots, 15 \quad (4.12)$$

$$J_2 = \Sigma 1, 5 + \emptyset 2, 3, 6, 7, 10, 11, \dots, 15 \quad (4.13)$$

$$K_2 = \Sigma 3, 7 + \emptyset 0, 1, 4, 5, 8, 9, \dots, 15 \quad (4.14)$$

$$J_1 = \Sigma 0, 2, 4, 6, 8 + \emptyset 1, 3, 5, 7, 9, 10, \dots, 15 \quad (4.15)$$

$$K_1 = \Sigma 1, 3, 5, 7, 9 + \emptyset 0, 2, 4, 6, 8, 10, 11, \dots, 15 \quad (4.16)$$

#### □ 4.2.9 Calcul algébrique : bascules JK

Le calcul algébrique des fonctions d'excitation nécessite l'emploi des équations d'entrée de la bascule JK (3.71) (3.72); appliquées au cas de la bascule  $Q_8$ , ces équations s'écrivent :

$$J_8 = \bar{Q}_8 Q_{8m}^+ + \lambda_J (Q_8 + Q_{8M}^+) \quad (4.17)$$

$$K_8 = Q_8 \bar{Q}_{8M}^+ + \lambda_K (\bar{Q}_8 + \bar{Q}_{8m}^+) \quad (4.18)$$

où  $Q_8^+$  est une fonction incomplètement définie donnée par la table d'états (fig. 4.20). L'examen de celle-ci nous permet d'écrire :

$$Q_8^+ = \Sigma 7, 8 + \emptyset 10, 11, \dots, 15 \quad (4.19)$$

dont on déduit les bornes inférieure et supérieure (§ 1.7.7) :

$$Q_{8m}^+ = \Sigma 7, 8 ; \bar{Q}_{8m}^+ = \Sigma 0, 1, \dots, 6, 9, 10, \dots, 15 \quad (4.20)$$

$$Q_{8M}^+ = \Sigma 7, 8, 10, 11, \dots, 15 ; \bar{Q}_{8M}^+ = \Sigma 0, 1, \dots, 6, 9 \quad (4.21)$$

avec :

$$Q_8 = \Sigma 8, 9, \dots, 15 ; \bar{Q}_8 = \Sigma 0, 1, \dots, 7 \quad (4.22)$$

En remplaçant les relations (4.20) (4.21) et (4.22) dans les expressions de  $J_8$  (4.17) et de  $K_8$  (4.18) on obtient :

$$J_8 = (\Sigma 0, 1, \dots, 7) \cdot (\Sigma 7, 8) + \lambda_J (\Sigma 8, 9, \dots, 15 + \Sigma 7, 8, 10, 11, \dots, 15) \quad (4.23)$$

$$K_8 = (\Sigma 8, 9, \dots, 15) \cdot (\Sigma 0, 1, \dots, 6, 9) + \lambda_K (\Sigma 0, 1, \dots, 7 + \Sigma 0, 1, \dots, 6, 9, 10, \dots, 15) \quad (4.24)$$

La définition des formes canoniques décimales (§ 1.6.16) et les propriétés des fonctions incomplètement définies (sect. 1.7) nous permettent alors de modifier les relations (4.23) et (4.24) :

$$\begin{aligned} J_8 &= \Sigma 7 + \lambda_J (\Sigma 7, 8, \dots, 15) = \Sigma 7 + \lambda_J (\Sigma 8, 9, \dots, 15) \\ &= \Sigma 7 + \emptyset 8, 9, \dots, 15 \end{aligned} \quad (4.25)$$

$$\begin{aligned} K_8 &= \Sigma 9 + \lambda_K (\Sigma 0, 1, \dots, 7, 9, 10, \dots, 15) \\ &= \Sigma 9 + \lambda_K (\Sigma 0, 1, \dots, 7, 10, 11, \dots, 15) \\ &= \Sigma 9 + \emptyset 0, 1, \dots, 7, 10, 11, \dots, 15 \end{aligned} \quad (4.26)$$

On retrouve bien les expressions précédentes de  $J_8$  (4.9) et de  $K_8$  (4.10).

Le calcul des fonctions  $J_4, K_4, J_2, K_2, J_1$  et  $K_1$  est laissé au soin du lecteur.

## 4.2.10 Simplification et réalisation : bascules JK

Chacune des fonctions d'excitation (4.9) à (4.16) peut être représentée dans une table de Karnaugh et peut être simplifiée séparément (fig. 4.21); les expressions algébriques obtenues sont les suivantes :

$$J_8 = Q_4 Q_2 Q_1 ; K_8 = Q_1 \quad (4.27)$$

$$J_4 = K_4 = Q_2 Q_1 \quad (4.28)$$

$$J_2 = \bar{Q}_8 Q_1 ; K_2 = Q_1 \quad (4.29)$$

$$J_1 = K_1 = 1 \quad (4.30)$$

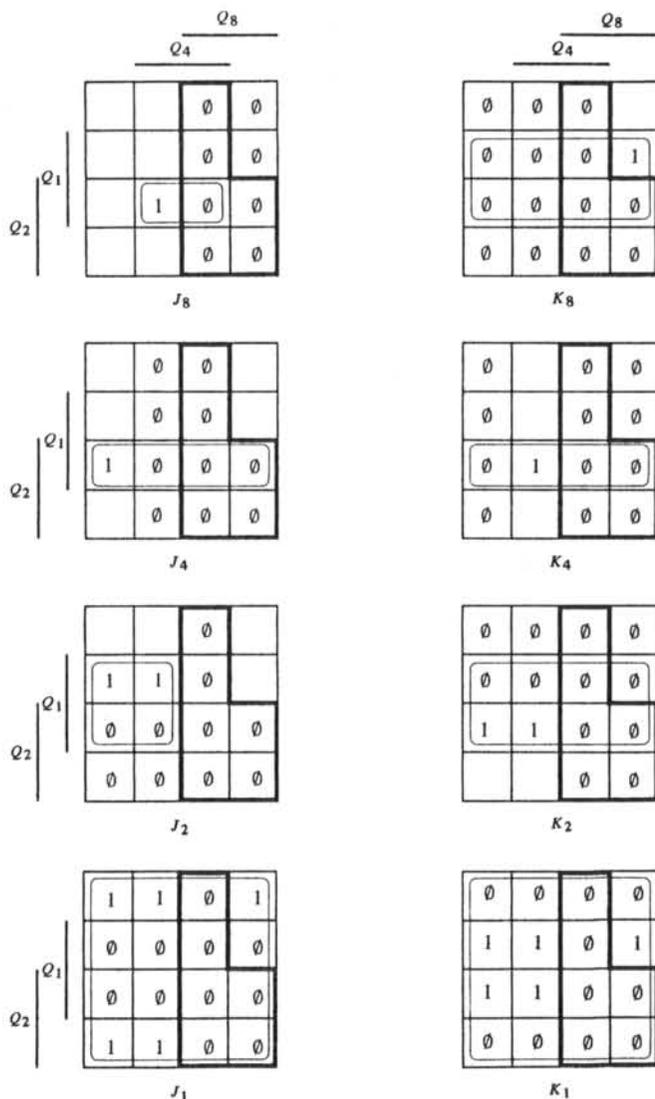


Fig. 4.21

Le logigramme ET de la figure 4.22 illustre une réalisation possible du compteur recherché.

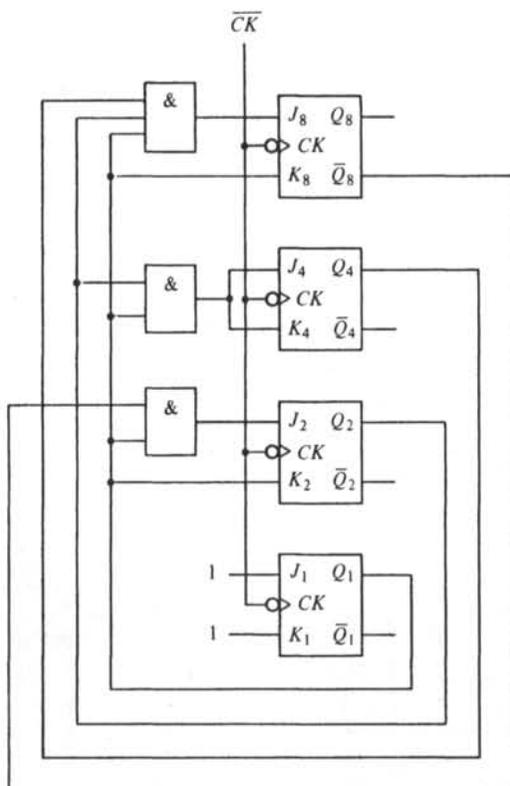


Fig. 4.22

#### 4.2.11 Calcul abrégé : bascules $D$

L'emploi de bascules  $D$ , dont l'équation d'entrée est simplement  $D = Q^+$  (3.63), entraîne un calcul abrégé. En partant du graphe des états (fig. 4.19), on représente dans une table de Karnaugh principale (§ 2.3.2) la séquence du compteur :

- chaque case de cette table (fig. 4.23) représente un état présent  $Q_8, Q_4, Q_2, Q_1$  ;
- on trouve dans chaque case l'état futur  $Q_8^+, Q_4^+, Q_2^+, Q_1^+$  tiré du graphe ;
- six cases représentent les états  $\emptyset$  (1010, 1011, ..., 1111) : on y reporte le symbole  $\emptyset$ .

On peut déduire de la table principale quatre tables de Karnaugh secondaires (§ 2.3.2) représentant les états futurs de chacune des bascules (fig. 4.23) ; l'équation d'entrée (3.63) devient dans ce cas :

$$D_8 = Q_8^+ ; D_4 = Q_4^+ ; D_2 = Q_2^+ ; D_1 = Q_1^+ \quad (4.31)$$

En simplifiant les tables de Karnaugh de la figure 4.23 on obtient l'expression algébrique des fonctions d'excitation :

$$D_8 = Q_8 \bar{Q}_1 + Q_4 Q_2 Q_1 \quad (4.32)$$

$$D_4 = Q_4 \bar{Q}_2 + Q_4 \bar{Q}_1 + \bar{Q}_4 Q_2 Q_1 \quad (4.33)$$

$$D_2 = \bar{Q}_8 \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1 \quad (4.34)$$

$$D_1 = \bar{Q}_1 \quad (4.35)$$

Une réalisation possible est celle de la figure 4.24.

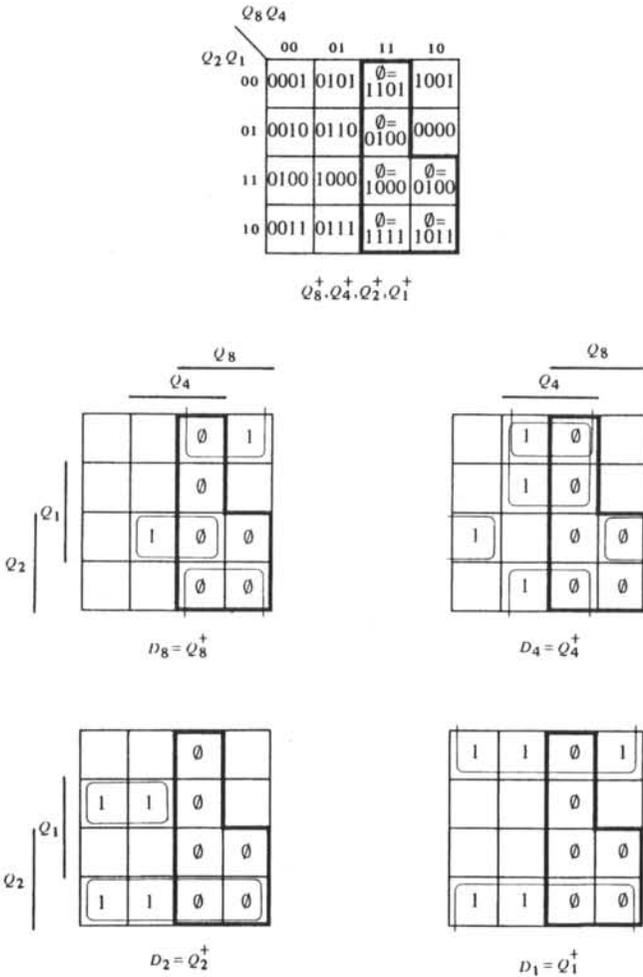


Fig. 4.23

#### 4.2.12 Commentaire

Les expressions algébriques (4.32) à (4.35) et le logigramme de la figure 4.24 (bascules  $D$ ) sont plus complexes que les relations (4.27) à (4.30) et la figure 4.22 (bascules  $JK$ ). Par ailleurs, un compteur synchrone à  $m$  bascules  $D$  exige le calcul de

$m$  fonctions d'excitation en lieu et place de  $2m$  si ces bascules sont du type  $JK$ . Le choix du type de bascules influence donc fortement la conception du système combinatoire réalisant les fonctions d'excitation.

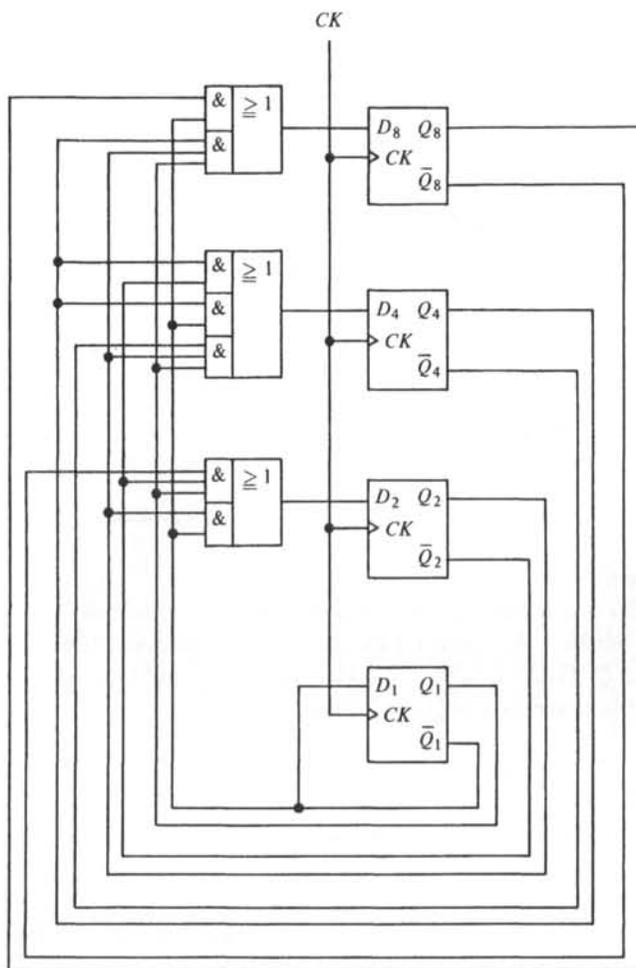


Fig. 4.24

#### 4.2.13 Exercice

Un compteur synchrone est défini par le graphe des états de la figure 4.5 (code binaire pur; mesure  $p = 16$ ). On demande de calculer deux variantes : l'une avec des bascules  $JK$ , l'autre avec des bascules  $D$ . Dans les deux cas on désire réaliser le système combinatoire avec des portes NON, ET, OU puis, en s'inspirant des considérations de la section 2.7, sous la forme d'un système itératif (voir aussi l'exercice du paragraphe 2.7.8). En admettant que chaque opérateur est affecté d'un certain retard, on comparera les deux structures obtenues (parallèle et itérative) sous l'angle du coût et de la rapidité de fonctionnement [44] (pp. 169 - 175), [45] (pp. 190 - 198), [54] (pp. 248 - 254).

On effectuera enfin la synthèse du même compteur avec des bascules  $T$  en recourant exclusivement au calcul algébrique du paragraphe 4.2.9 (voir aussi l'exercice du paragraphe 3.4.12).

#### 4.2.14 Exercice

Un compteur synchrone à seize états est défini par la *code négabinaire* de la figure 4.25. On effectuera la synthèse à l'aide de bascules  $JK$  et  $D$  puis on cherchera dans les deux cas un système combinatoire itératif réalisant les fonctions d'excitation.

$Q_8 Q_4$ $Q_2 Q_1$	00	01	11	10
00	0001	0101	1101	1001
01	0110	1010	0010	1110
11	0000	0100	1100	1000
10	0011	0111	1111	1011

$Q_8^+, Q_4^+, Q_2^+, Q_1^+$

Fig. 4.25

$AB$ $CD$	00	01	11	10
00	1010	1110	0110	0100
01	0111	1100	0101	1000
11	1011	1101	0001	0000
10	0011	0010	1111	1001

$A^+, B^+, C^+, D^+$

Fig. 4.26

#### 4.2.15 Exercice

Un compteur synchrone à seize états est défini par le code de la figure 4.26. Effectuer le calcul tabulaire, algébrique ou abrégé de ce compteur en utilisant successivement les bascules  $SR$ ,  $D$ ,  $T$ ,  $JK$  et  $RST$  (§ 3.4.10). Les résultats obtenus pourront être comparés à ceux de la référence [34] (pp. 117 - 132).

#### 4.2.16 Analyse

Le but de l'analyse est la détermination des modes de représentation du compteur synchrone (graphe des états et table d'états) à partir d'un logigramme donné.

Tout compteur à  $m$  bascules et  $p$  états possède  $2^m - p$  états  $\emptyset$  : le calcul du comportement à partir de ces états est un cas particulier important de l'analyse.

#### 4.2.17 Méthode

Pour chaque instant d'horloge, on procède de la façon suivante :

- on relève l'état présent  $Q_1, Q_2, \dots, Q_m$  ;
- les fonctions d'excitation (par exemple  $J$  et  $K$ ) sont données par le logigramme (fig. 4.1) : elles ne dépendent que des variables  $Q_1, Q_2, \dots, Q_m$  et l'on peut calculer leurs valeurs dans l'état présent ;
- l'examen de la table d'états des bascules utilisées (fig. 3.18, 3.23, 3.30 et 3.33) nous permet de déterminer l'état futur  $Q_j^+$  ( $j = 1, 2, \dots, m$ ) en fonction de l'état présent  $Q_j$  et des fonctions d'excitation ( $J_j$  et  $K_j$  par exemple) ;
- on reporte l'état présent  $Q_1, Q_2, \dots, Q_m$  et l'état futur  $Q_1^+, Q_2^+, \dots, Q_m^+$  dans la table ou le graphe des états du compteur.

#### 4.2.18 Exemple

On veut déterminer le comportement d'un compteur par 10 à partir de ses six états  $\emptyset$ ; l'analyse sera effectuée pour les deux variantes précédemment calculées et comportant des bascules  $JK$  (§ 4.2.10) ou  $D$  (§ 4.2.11).

#### 4.2.19 Calcul algébrique : bascules $JK$

Considérons tout d'abord la bascule  $Q_8$  dont l'équation caractéristique (3.56) s'écrit :

$$Q_8^+ = J_8 \bar{Q}_8 + \bar{K}_8 Q_8 \quad (4.36)$$

En remplaçant  $J_8$  et  $K_8$  par leurs expressions algébriques (4.27), l'équation (4.36) de  $Q_8^+$  devient :

$$Q_8^+ = \bar{Q}_8 Q_4 Q_2 Q_1 + Q_8 \bar{Q}_1 \quad (4.37)$$

On rappelle dans la figure 4.27 les six lignes de la table d'états représentant les six états  $\emptyset$  (fig. 4.20 : No 10 à 15); il est alors aisé de calculer, ligne par ligne, la valeur finale de  $Q_8^+$ .

Un calcul identique est effectué pour les trois bascules  $Q_4$ ,  $Q_2$  et  $Q_1$  à partir de l'équation caractéristique (3.56) et des fonctions d'excitation (4.28) à (4.30) :

$$\begin{aligned} Q_4^+ &= Q_2 Q_1 \cdot \bar{Q}_4 + (\overline{Q_2 Q_1}) \cdot Q_4 \\ &= \bar{Q}_4 Q_2 Q_1 + Q_4 \bar{Q}_2 + Q_4 \bar{Q}_1 \end{aligned} \quad (4.38)$$

$$Q_2^+ = \bar{Q}_8 \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1 \quad (4.39)$$

$$Q_1^+ = \bar{Q}_1 \quad (4.40)$$

Les valeurs de  $Q_4^+$ ,  $Q_2^+$  et  $Q_1^+$  sont alors calculées pour les six états  $\emptyset$  et reportées dans la table d'états (fig. 4.27).

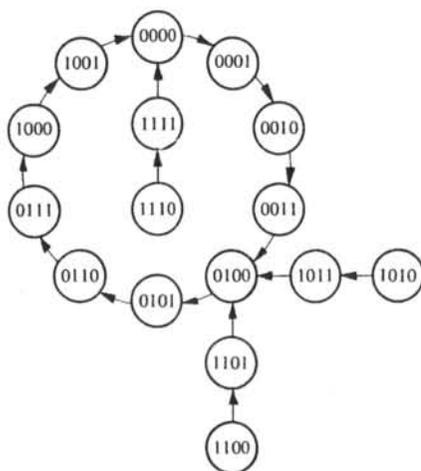
No	$t$ $Q_8, Q_4, Q_2, Q_1$	$t^+$ $Q_8^+, Q_4^+, Q_2^+, Q_1^+$
10	1 0 1 0	1 0 1 1
11	1 0 1 1	0 1 0 0
12	1 1 0 0	1 1 0 1
13	1 1 0 1	0 1 0 0
14	1 1 1 0	1 1 1 1
15	1 1 1 1	0 0 0 0

Fig. 4.27

#### 4.2.20 Mode de représentation : graphe des états

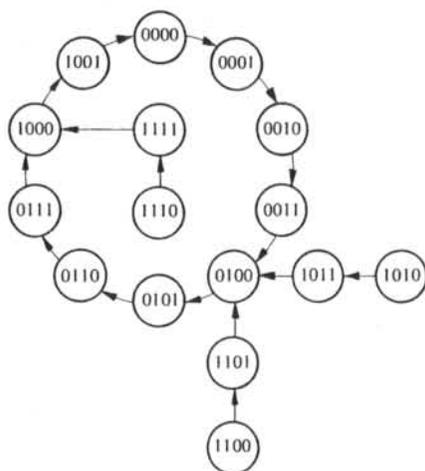
A partir de la table d'états (fig. 4.27) il est possible de compléter le graphe initial (fig. 4.19) : on en déduit le graphe des états final de la figure 4.28. L'examen de celui-ci

montre que le compteur rejoint toujours la séquence de comptage; dans le pire des cas (états 1010, 1100 et 1110) deux périodes du signal d'horloge sont nécessaires avant de regagner la séquence imposée.



$Q_8 \cdot Q_4 \cdot Q_2 \cdot Q_1$

Fig. 4.28



$Q_8 \cdot Q_4 \cdot Q_2 \cdot Q_1$

Fig. 4.29

#### 4.2.21 Calcul abrégé : bascules $D$

Conformément aux propriétés des tables de Karnaugh (§ 2.4.4) tout symbole  $\emptyset$  inclus dans un ou plusieurs blocs prend la valeur 1. Un simple examen des quatre tables secondaires de la figure 4.23 détermine les valeurs binaires de tous les états  $Q_j^+ = \emptyset$  ( $j = 8, 4, 2, 1$ ); ces valeurs sont directement reportées dans la table principale (fig. 4.23) où elles définissent les états futurs  $Q_8^+, Q_4^+, Q_2^+, Q_1^+$ . Il en découle un graphe des états (fig. 4.29) qui est légèrement différent de celui du compteur avec bascules  $JK$  (fig. 4.28).

#### 4.2.22 Définition

Lorsqu'un ou plusieurs états  $\emptyset$  d'un compteur constituent une séquence bouclée sur elle-même on observe un *cycle parasite*. Si le compteur se trouve dans un tel cycle, à l'enclenchement par exemple, il ne pourra jamais le quitter pour regagner la séquence de comptage désirée; généralement on corrige le logigramme original si l'analyse des états  $\emptyset$  révèle un cycle parasite.

#### 4.2.23 Conclusion

La méthode de synthèse des compteurs synchrones s'applique à des codes et à des mesures quelconques.

Le compteur synchrone peut fonctionner à une fréquence supérieure à celle d'un diviseur (§ 4.1.10) : les variations des bascules sont simultanées puisqu'elles sont provoquées par le même signal d'horloge; la fréquence maximale est donc limitée seulement par le retard de chaque bascule et non pas par leur nombre.

#### 4.2.24 Bibliographie

La synthèse des compteurs synchrones fait l'objet d'un grand nombre de méthodes recourant au calcul tabulaire [28] (pp. 147 - 159), [40] (pp. 37 - 146), [45] (pp. 190 - 206), [46] (pp. 131 - 137), [52], [53], [55] ou au calcul algébrique [34] (pp. 117 - 132), [38] (pp. 68 - 71), [20] (pp. 229 - 258). Une méthode originale, dite méthode de perturbation, est traitée dans [22] (§ 5.5.3).

On trouve également dans la littérature des logigrammes réalisant notamment :

- quinze compteurs ( $p = 2$  à 16) dans le code binaire pur, avec bascules  $JK$  [45] (pp. 209 - 223);
- onze compteurs ( $p = 3$  à 13) dans le code binaire pur, avec bascules  $JK$  [54] (pp. 272 - 277);
- quinze compteurs ( $p = 2$  à 16) dans des codes divers, sans aucun cycle parasite, avec bascules  $JK$  [28] (pp. 407 - 410);
- trente compteurs ( $p = 3$  à 32) dans des codes divers, avec bascules  $JK$  [53].

Une discussion sur la fréquence maximale de fonctionnement peut être trouvée dans les références [28] (pp. 161 - 170) et [44] (pp. 169 - 175).

#### 4.2.25 Exercice

En admettant les hypothèses du paragraphe 4.1.3, on tracera le chronogramme des seize états du compteur de la figure 4.22, décrit par le graphe de la figure 4.28. A quelles conditions les bascules  $JK$  de la figure 4.22 valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ? On représentera ensuite dans le même chronogramme la fonction  $Z = Q_8 \overline{Q_1}$  (4.3) dont les régimes transitoire et permanent seront mis en évidence; existe-t-il des comportements transitoires de  $Z$  semblables à celui de la figure 4.3 ?

#### 4.2.26 Exercice

Effectuer la synthèse et l'analyse (états  $\emptyset$ ) des compteurs synchrones définis par les séquences suivantes :

- $Q_4, Q_2, Q_1 = 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 000$
- $Q_C, Q_B, Q_A = 111 \rightarrow 110 \rightarrow 101 \rightarrow 011 \rightarrow 000 \rightarrow 111$
- $Q_C, Q_B, Q_A = 111 \rightarrow 100 \rightarrow 011 \rightarrow 001 \rightarrow 000 \rightarrow 111$

Calculer deux variantes : l'une avec des bascules  $JK$ , l'autre avec des bascules  $D$ .

#### 4.2.27 Exercice

Le logigramme de la figure 4.30 réalise un *compteur Johnson* [46] (pp. 134 - 137). On demande d'effectuer l'analyse de ce compteur et de mettre en évidence deux séquences indépendantes de huit états chacune; en choisissant l'une d'elles comme séquence de comptage, on cherchera à supprimer le cycle parasite constitué par l'autre. La correction du logigramme de la figure 4.30 doit modifier une seule fonction d'excitation :  $D_D = Q_D^+$ .

#### 4.2.28 Exercice

Des recherches ont été entreprises pour minimiser le système combinatoire des compteurs synchrones [40] (pp. 216 - 229), [48], [49] et/ou pour supprimer les

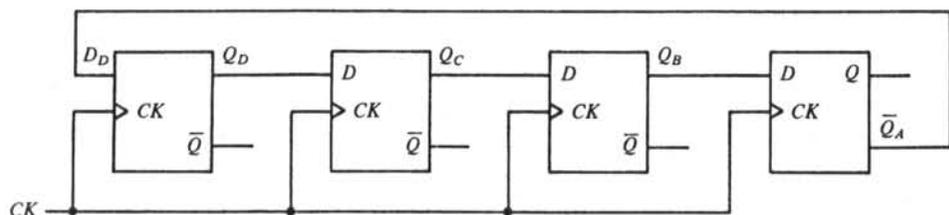


Fig. 4.30

cycles parasites produits par les états  $\emptyset$  [49], [50]. L'analyse des compteurs suivants illustre ces tentatives :

- compteur par 10 à bascules  $T$  :

$$T_1 = Q_2 \bar{Q}_3 \bar{Q}_4 ; T_2 = Q_3 Q_4 ; T_3 = \bar{Q}_1 Q_4 ; T_4 = 1$$

- compteur par 10 à bascules  $JK$  :

$$J_3 = \bar{Q}_1 ; K_3 = 1$$

$$J_2 = Q_3 ; K_2 = 1$$

$$J_1 = \bar{Q}_2 ; K_1 = 1$$

$$J_0 = K_0 = 1$$

- compteur à bascules  $JK$  :

$$J_2 = Q_0 ; K_2 = \bar{Q}_1$$

$$J_1 = Q_0 ; K_1 = Q_2$$

$$J_0 = Q_2 ; K_0 = Q_1$$

- compteur à bascules  $JK$  et  $D$  :

$$D_2 = \bar{Q}_0$$

$$J_1 = K_1 = Q_2$$

$$J_0 = Q_1 ; K_0 = \bar{Q}_2$$

On demande de tracer le graphe des états (y compris les états  $\emptyset$ ) de chacun de ces compteurs.

#### ■ 4.2.29 Exercice

Effectuer la synthèse d'un compteur synchrone dit *compteur en anneau*, comportant trois bascules  $Q_C, Q_B, Q_A$  et défini par le graphe des états de la figure 4.31; trois variantes seront calculées pour des bascules  $JK, D$  et  $SR$ . On demande de réaliser les expressions simplifiées des fonctions d'excitation et d'analyser les logigrammes finals pour tous les états  $\emptyset$ . On discutera enfin les mérites respectifs de chacune des trois variantes, ainsi que des mesures à prendre pour supprimer les éventuels cycles parasites.

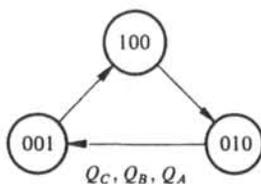


Fig. 4.31

#### ■ 4.2.30 Exercice

Effectuer la synthèse, puis l'analyse des états  $\emptyset$ , d'un compteur par cinq défini par le code suivant :

$$Q_4, Q_2, Q_1 = 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 000$$

Les bascules sont du type *LP* (voir l'exercice du paragraphe 3.4.10); on cherche un logigramme NAND minimal.

#### ■ 4.2.31 Exercice

Effectuer la synthèse, puis l'analyse des états  $\emptyset$ , d'un compteur par cinq défini par le code suivant :

$$Q_C, Q_B, Q_A = 000 \rightarrow 111 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 000$$

On calculera deux variantes : l'une avec des bascules *xy* (voir l'exercice du paragraphe 3.4.10) et l'autre avec des bascules *RST* (voir l'exercice du paragraphe 3.4.10 et la référence [52]); dans les deux cas on cherche un logigramme NAND minimal.

#### 4.2.32 Exercice

Pour une mesure donnée  $p$ , on observe qu'une augmentation du nombre des bascules du compteur synchrone entraîne une simplification de son système combinatoire. Dans cette optique, on effectuera la synthèse des deux compteurs par huit décrits par les figures 4.32 et 4.33 [51]; trois variantes seront calculées pour des bascules *JK*, *T* et *D*. L'analyse des états  $\emptyset$  du second compteur sera entreprise dans ces trois cas, et les éventuels cycles parasites seront corrigés.

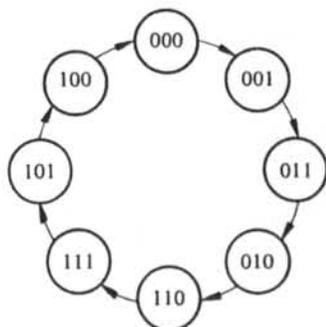
 $Q_C, Q_B, Q_A$ 

Fig. 4.32

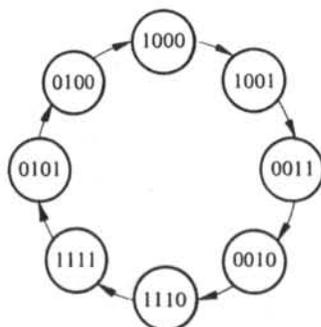
 $Q_D, Q_C, Q_B, Q_A$ 

Fig. 4.33

## 4.3 COMPTEURS QUASI-SYNCHRONES

## 4.3.1 Hypothèses

Par définition (§ 4.1.1) le compteur quasi-synchrone est l'assemblage quelconque d'un système combinatoire et d'une ou plusieurs bascules. Les hypothèses du paragraphe 4.1.3 sont admises sans modification.

## 4.3.2 Analyse

Le logigramme de la figure 4.34 ainsi que le schéma de la figure 4.35 représentent un compteur quasi-synchrone; celui-ci n'est ni un diviseur de fréquence (car  $J_2 = \bar{Q}_8$ , par exemple), ni un compteur synchrone (car  $CK_4 \neq CK_2$ ).

On remarque toutefois que la bascule  $Q_1$  est un diviseur de fréquence par deux ( $J_1 = K_1 = 1$  : § 3.3.9). On se limite donc à l'analyse du compteur quasi-synchrone formé par les trois bascules  $Q_8, Q_4, Q_2$  et attaqué par le signal d'horloge  $\bar{CK}_2 = Q_1$ .

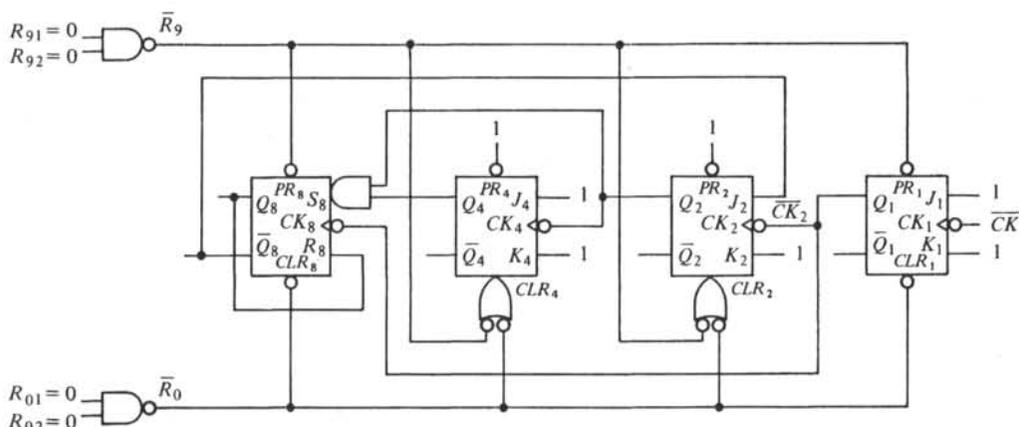


Fig. 4.34

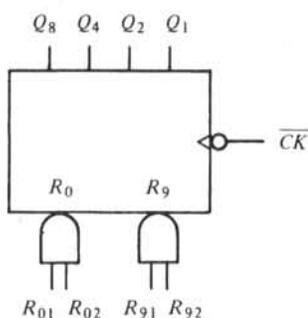


Fig. 4.35

## 4.3.3 Méthode

Chaque bascule est examinée séparément : le modèle synchrone (§ 3.2.10) et en particulier l'équation caractéristique décrivent l'état futur  $Q^+$  en fonction de l'état présent  $Q$  et des valeurs d'excitation ( $J$  et  $K$  par exemple).

Le compteur est ensuite analysé globalement : le modèle quasi-synchrone de chaque bascule (§ 3.2.3) ainsi que les hypothèses du paragraphe 4.1.3 nous permettent de tracer un chronogramme en procédant d'une façon semblable à celle des diviseurs de fréquence (§ 4.1.5 et 4.1.9).

#### 4.3.4 Exemple

Le logigramme de la figure 4.34 ainsi que les équations caractéristiques (3.56), (3.61) et (3.49) décrivent le fonctionnement des trois bascules  $JK$ ,  $T$  et  $SR$  :

$$Q_2^+ = J_2 \bar{Q}_2 + \bar{K}_2 Q_2 = \bar{Q}_8 \bar{Q}_2 \quad (4.41)$$

$$Q_4^+ = \bar{Q}_4 \quad (4.42)$$

$$Q_8^+ = S_8 + \bar{R}_8 Q_8 = Q_4 Q_2 \quad (4.43)$$

L'emploi d'une bascule  $SR$  ( $Q_8$ ) entraîne l'existence de la condition (3.48) :

$$S_8 R_8 = Q_8 Q_4 Q_2 = 0 \quad (4.44)$$

Les signaux d'horloge des bascules sont définis par les relations :

$$CK_2 = CK_8 = \bar{Q}_1 \quad ; \quad CK_4 = \bar{Q}_2 \quad (4.45)$$

tandis que les entrées asynchrones s'expriment par les équations :

$$PR_1 = PR_8 = R_{91} \cdot R_{92} = R_9 \quad ;$$

$$CLR_1 = CLR_8 = R_{01} \cdot R_{02} = R_0 \quad ;$$

$$CLR_2 = CLR_4 = R_{91} \cdot R_{92} + R_{01} \cdot R_{02} = R_9 + R_0 \quad (4.46)$$

La détermination du chronogramme (fig. 4.36) est effectuée de la façon suivante :

- en partant d'un état permanent  $Q_8, Q_4, Q_2$  quelconque ( $Q_8, Q_4, Q_2 = 000$  par exemple) on calcule les états futurs  $Q_8^+, Q_4^+, Q_2^+$  donnés par les expressions (4.41) à (4.43); on reporte ces valeurs ( $Q_8^+, Q_4^+, Q_2^+ = 011$ ) dans le chronogramme.
- A la première descente du signal d'horloge  $\bar{CK}_2$  les deux bascules attaquées par ce signal ( $Q_8$  et  $Q_2$ ) prennent les nouvelles valeurs  $Q_8^+$  et  $Q_2^+$  après un certain retard.
- La troisième bascule ( $Q_4$ ) est un diviseur de fréquence par deux ( $J_4 = K_4 = 1$ ) dont l'état change à chaque descente de  $\bar{CK}_4 = Q_2$ .
- La première descente du signal d'horloge  $\bar{CK}_2$  n'entraîne donc aucun changement des états  $Q_8$  et  $Q_4$  : le compteur atteint l'état permanent  $Q_8, Q_4, Q_2 = 001$ .
- En poursuivant le même calcul de proche en proche on fait apparaître dans le chronogramme une séquence de cinq états dans le code binaire pur :

$$Q_8, Q_4, Q_2 = 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 000$$

- On vérifie enfin que les valeurs des entrées asynchrones (4.46) sont, dans le cas particulier (fig. 4.34), sans effet (fonctionnement normal : § 3.2.6):

$$PR_1 = PR_8 = 0 \quad ; \quad CLR_1 = CLR_2 = CLR_4 = CLR_8 = 0 \quad (4.47)$$

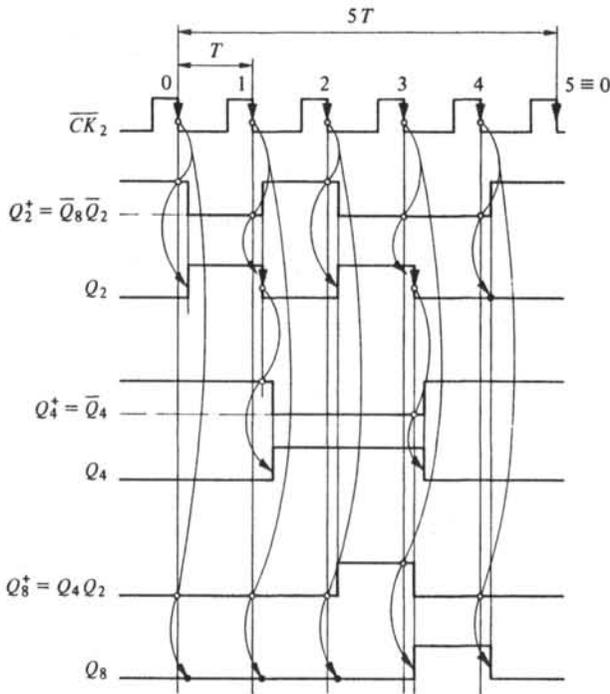


Fig. 4.36

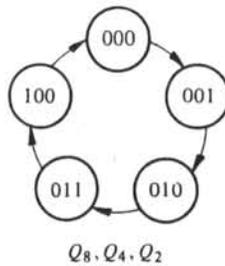


Fig. 4.37

On a donc obtenu un compteur par cinq dont le graphe des états est celui de la figure 4.37. En rappelant que la bascule  $Q_1$  est un diviseur par deux, on constate que le compteur quasi-synchrone de la figure 4.34 produit une séquence de dix états dans le code binaire pur : ce compteur est donc équivalent au diviseur de fréquence de la figure 4.9 et aux compteurs synchrones des figures 4.22 et 4.24.

#### 4.3.5 Exercice

Effectuer l'analyse des états  $\emptyset$  du compteur quasi-synchrone de la figure 4.34 en complétant le chronogramme de la figure 4.36. Déterminer le graphe des états transitoires et permanents. A quelles conditions les bascules de la figure 4.34 valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ? A quelles conditions la bascule SR valide-t-elle la relation (4.44) ?

## 4.3.6 Exercice

Le compteur quasi-synchrone des figures 4.34 et 4.35 est muni de quatre variables  $R_{01}$ ,  $R_{02}$ ,  $R_{91}$  et  $R_{92}$  agissant sur les entrées asynchrones de certaines bascules selon la relation (4.46). Effectuer l'analyse de ce compteur dans les trois cas suivants :

$$R_{01} = Q_2 ; R_{02} = Q_1 ; R_{91} = 0 ; R_{92} = \emptyset \quad (4.48)$$

$$R_{01} = Q_4 ; R_{02} = Q_1 ; R_{91} = 0 ; R_{92} = \emptyset \quad (4.49)$$

$$R_{01} = 0 ; R_{02} = \emptyset ; R_{91} = Q_4 ; R_{92} = Q_2 \quad (4.50)$$

Déterminer le chronogramme, le graphe des états permanents puis la mesure de chacun des compteurs. Quelles sont les valeurs de  $R_{01}$ ,  $R_{02}$ ,  $R_{91}$  et  $R_{92}$  qui permettraient de réaliser un compteur par 4 ? par 6 ? par 8 ? par 9 ?

## 4.3.7 Exercice

Effectuer l'analyse des compteurs quasi-synchrones des figures 4.38 à 4.40. Tracer les chronogrammes puis déterminer les codes et les mesures; montrer que celles-ci ne sont pas toujours égales aux rapports de division.

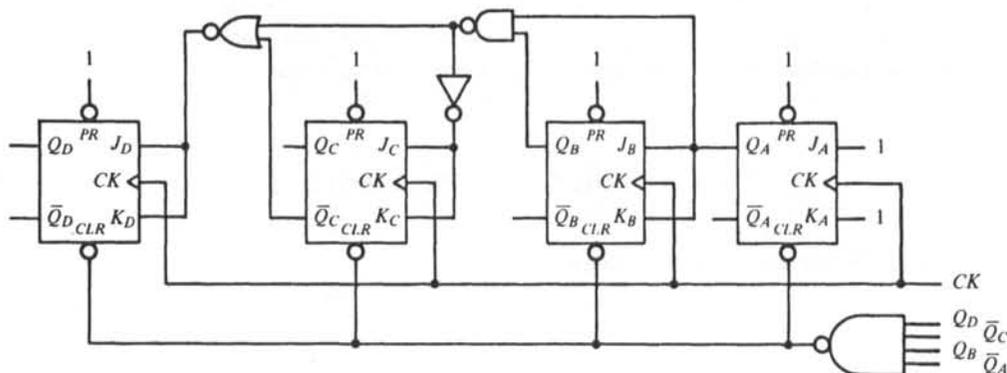


Fig. 4.38

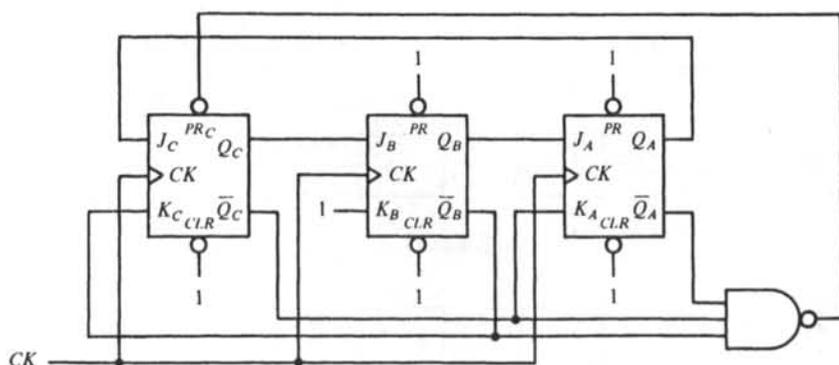


Fig. 4.39

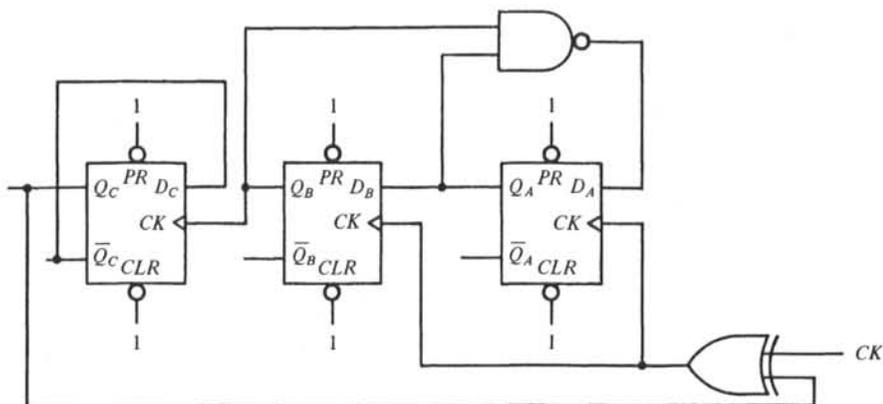


Fig. 4.40

### 4.3.8 Synthèse

Il existe des méthodes pour la synthèse de certains compteurs quasi-synchrones; les plus commodes de celles-ci sont décrites dans les références [12] (pp. 190 - 192), [46] (pp. 138 - 140), [53] et [55]; d'autres variantes font l'objet des références [44] (pp. 177 - 193) et [56]. Etant donné la variété des logigrammes réalisant ces compteurs, on ne dispose en fait d'aucune méthode générale et systématique.

## 4.4 DÉCOMPOSITION DES COMPTEURS

### 4.4.1 Définitions : décomposition en série

Lorsqu'un compteur par  $p$  peut être réalisé avec  $q$  compteurs assemblés selon le schéma de la figure 4.41 et dont les mesures  $p_1, p_2, \dots, p_q$  vérifient la relation arithmétique

$$p = p_1 \cdot p_2 \cdot \dots \cdot p_q \quad (4.51)$$

on dit que ce compteur admet une *décomposition en série*.

Les  $q$  compteurs par  $p_1, p_2, \dots, p_q$  sont les *compteurs partiels*; le compteur par  $p$  est le *compteur complet*.

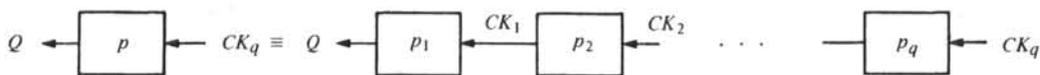


Fig. 4.41

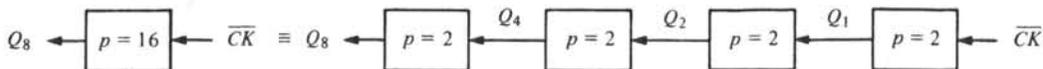


Fig. 4.42

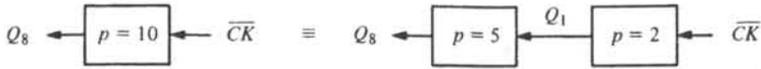


Fig. 4.43

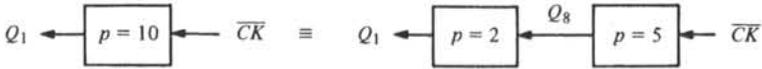


Fig. 4.44

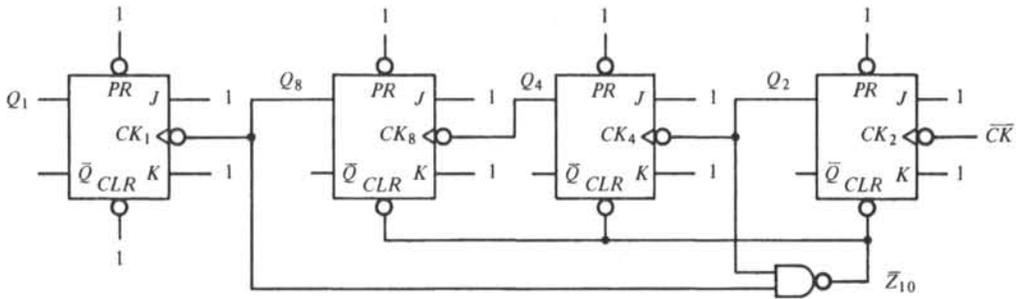


Fig. 4.45

#### 4.4.2 Exemples

Le diviseur de fréquence par 16 de la figure 4.2 admet une décomposition en série représentée dans la figure 4.42 et décrite par la relation arithmétique :

$$16 = 2 \cdot 2 \cdot 2 \cdot 2 \quad (4.52)$$

Le diviseur de fréquence par 10 de la figure 4.9 admet une décomposition en série (fig. 4.43) si l'on supprime la connexion  $\bar{Z}_{10} = \overline{CLR}_1$ , c'est-à-dire si  $\overline{CLR}_1 = 1$  (§ 4.1.23). Le compteur par 10 de la figure 4.34 admet la même décomposition si l'on vérifie la relation (4.47), c'est-à-dire si  $PR_1 = PR_8 = 0$  et si  $CLR_1 = CLR_2 = CLR_4 = CLR_8 = 0$ . Dans ces deux cas, la décomposition en série est décrite par l'expression arithmétique :

$$10 = 5 \cdot 2 \quad (4.53)$$

Si l'on permute les deux compteurs partiels de la figure 4.43 on obtient le schéma de la figure 4.44 décrit par la relation arithmétique :

$$10 = 2 \cdot 5 \quad (4.54)$$

Une réalisation possible de cette décomposition est celle de la figure 4.45 qui découle du logigramme du diviseur par 10 (fig. 4.9); du chronogramme de celui-ci (fig. 4.10) on déduit le nouveau chronogramme de la figure 4.46 et le graphe des états permanents  $Q_1, Q_8, Q_4, Q_2$  de la figure 4.47. Le compteur complet de la figure 4.45 a la même mesure que celui de la figure 4.9 ( $p = 10$ ) mais produit un code différent (voir la figure 4.14) même si l'on permute les variables  $Q_1, Q_8, Q_4$  ou  $Q_2$ . Le lecteur s'en convaincra en traçant, à titre d'exemple, le graphe des états permanents  $Q_8, Q_4, Q_2, Q_1$ .

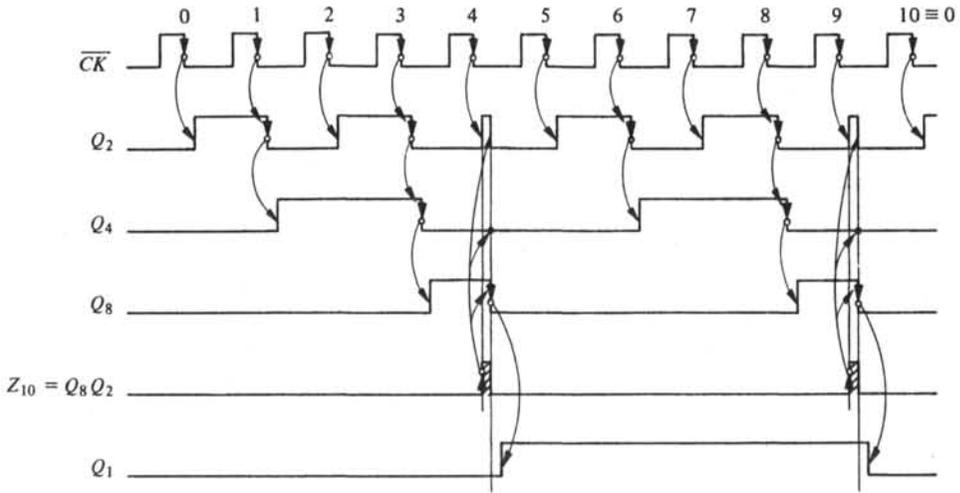
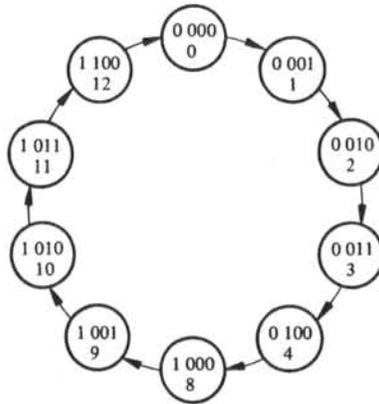


Fig. 4.46



$Q_1, Q_8, Q_4, Q_2$

Fig. 4.47

### 4.4.3 Propriétés

Les mesures  $p_1, p_2, \dots, p_q$  sont des nombres entiers. On déduit de la relation (4.51) qu'un compteur par  $p$  admet une (ou plusieurs) décomposition(s) en série si  $p$  peut être décomposé en facteurs, c'est-à-dire si  $p$  n'est pas un nombre premier.

Pour un produit de facteurs  $p_1 \cdot p_2 \cdot \dots \cdot p_q$  donné il existe en général plusieurs permutations distinctes de ces facteurs (par exemple  $10 = 5 \cdot 2 = 2 \cdot 5$ ); chacune d'elles est représentée par un schéma différent (fig. 4.43 et 4.44 par exemple) et produit un code différent (fig. 4.14 et 4.47). La décomposition en série n'est donc pas une opération commutative et nous distinguerons chacune des permutations de l'expression arithmétique (4.51).

Chaque compteur partiel peut être un diviseur de fréquence, un compteur synchrone ou un compteur quasi-synchrone; étant donné la présence de plusieurs signaux d'horloge distincts ( $CK_1, CK_2, \dots, CK_q$  : fig. 4.41) le compteur complet n'est en aucun cas un compteur synchrone.

#### 4.4.4 Définition : décomposition en parallèle

Lorsqu'un compteur par  $p$  peut être réalisé avec  $q$  compteurs assemblés selon le schéma de la figure 4.48 (dans laquelle  $SC$  est un système combinatoire) et dont les mesures  $p_1, p_2, \dots, p_q$  vérifient la relation arithmétique

$$p = p_1 \cdot p_2 \cdot \dots \cdot p_q \quad (4.55)$$

on dit que ce compteur admet une *décomposition en parallèle*.

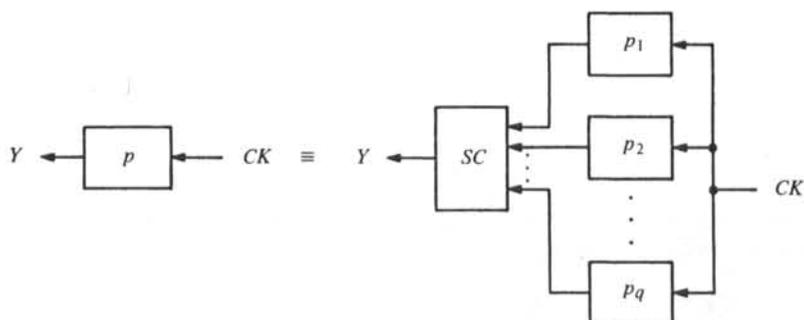


Fig. 4.48

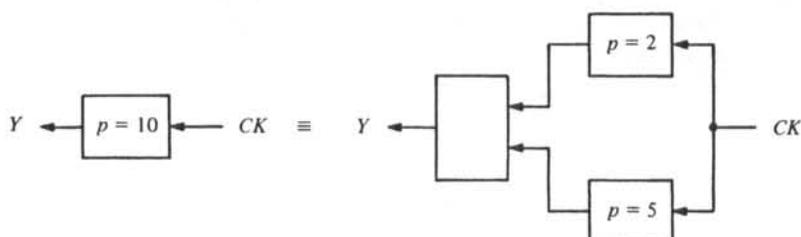


Fig. 4.49

#### 4.4.5 Exemple

La relation arithmétique (4.54) décrit une décomposition en parallèle qui peut être représentée par le schéma de la figure 4.49; une réalisation possible est celle de la figure 4.50 qui découle du logigramme de la figure 4.9 (dans laquelle on a remplacé la connexion  $\overline{CLR}_1 = \bar{Z}_{10}$  par  $\overline{CLR}_1 = 1$  selon le paragraphe 4.1.23). On assemble donc un diviseur de fréquence par 5 (bascules  $Q_8, Q_4, Q_2$ ) et un diviseur de fréquence par deux (bascule  $Q_1$ ) : ces deux diviseurs sont attaqués par le même signal d'horloge  $\overline{CK}$ .

Le chronogramme de la figure 4.51 décrit le comportement du compteur complet; en réalisant la fonction logique

$$X = Q_8 Q_1 \quad (4.56)$$

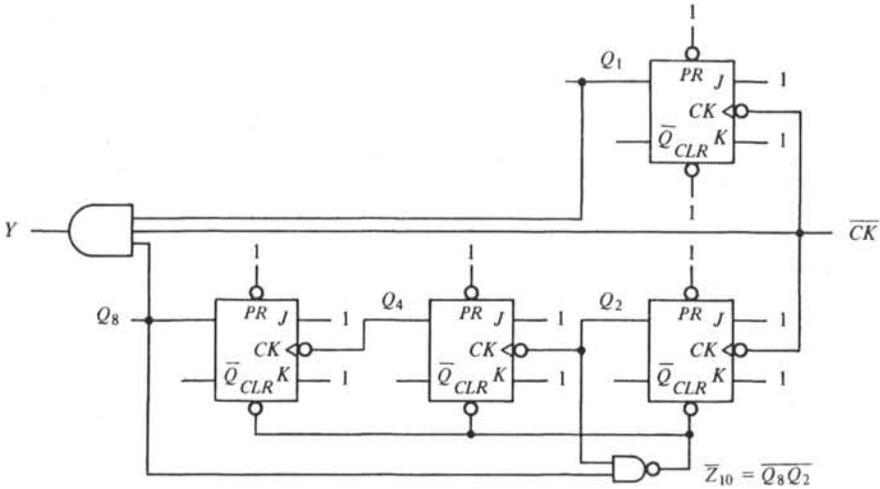


Fig. 4.50

on obtient un signal de sortie  $X$  dont la période est dix fois celle du signal d'entrée  $\overline{CK}$ . Il existe un régime transitoire  $X = 1$  de durée  $\Delta t$  (fig. 4.51) qui peut être supprimé en remplaçant  $X$  par la fonction

$$Y = X \overline{CK} = Q_8 Q_1 \overline{CK} \quad (4.57)$$

Le graphe des états permanents  $Q_8, Q_4, Q_2, Q_1$  de la figure 4.52 ne produit pas le code binaire pur qui caractérisait le diviseur de fréquence original (fig. 4.14).

On constate enfin que la décomposition en parallèle décrite par la relation arithmétique (4.53) est représentée par le même schéma (fig. 4.49) que celui réalisant la relation (4.54).

#### 4.4.6 Propriétés

Un compteur par  $p$  admet une décomposition en parallèle si  $p$  n'est pas un nombre premier (§ 4.4.3); cette condition est nécessaire, mais pas suffisante : les facteurs  $p_1, p_2, \dots, p_q$  de la relation (4.55) doivent être premiers entre eux. Une démonstration mathématique peut être trouvée dans la référence [47].

Contrairement à la décomposition en série, chaque permutation de la relation arithmétique (4.55) est représentée par un schéma identique et produit un même code : la décomposition en parallèle est donc une opération commutative.

Chaque compteur partiel peut être un diviseur de fréquence, un compteur synchrone ou un compteur quasi-synchrone. Si tous les compteurs partiels sont synchrones, le compteur complet est également synchrone (fig. 4.48).

#### 4.4.7 Exercice

Effectuer les décompositions en parallèle du diviseur de fréquence par 16 de la figure 4.2 qui sont décrites par les relations arithmétiques suivantes :

$$16 = 2 \cdot 2 \cdot 2 \cdot 2 \quad (4.58)$$

$$16 = 4 \cdot 4 \quad (4.59)$$

$$16 = 2 \cdot 8 \quad (4.60)$$

Tracer dans chaque cas un chronogramme et le graphe des états correspondant; en remarquant que les facteurs des expressions (4.58) à (4.60) ne sont pas premiers entre

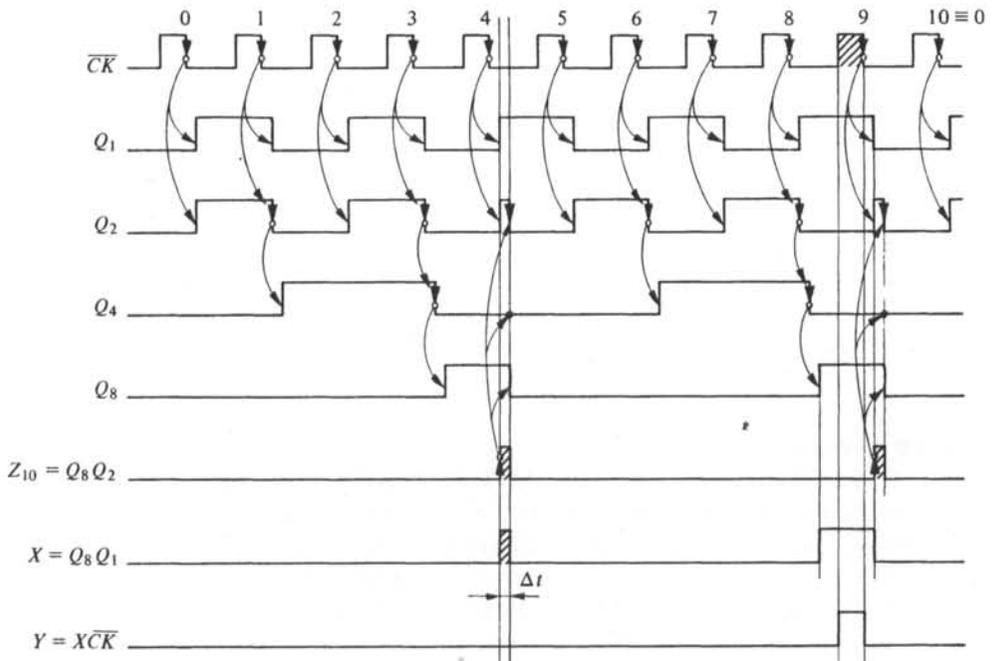
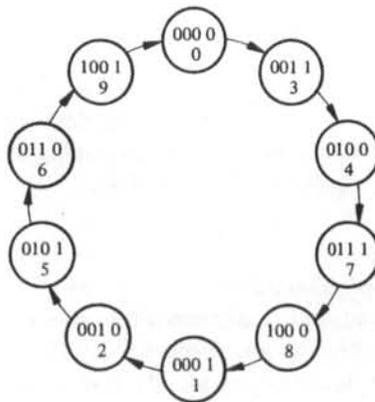


Fig. 4.51



$Q_8, Q_4, Q_2, Q_1$

Fig. 4.52

eux, déterminer les mesures des compteurs complets ainsi obtenus. Tirer une règle générale permettant de calculer la mesure d'une décomposition en parallèle dont les facteurs  $p_1, p_2, \dots, p_q$  ne sont pas tous premiers entre eux.

#### 4.4.8 Exercice

Compléter le chronogramme de la figure 4.51 et le graphe de la figure 4.52 avec tous les états  $\emptyset$  du compteur par 10 (fig. 4.50). Comparer les résultats avec ceux du même compteur non décomposé (fig. 4.11 à 4.14).

#### 4.4.9 Exercice

Réaliser le compteur par 10 de la figure 4.49 avec deux compteurs partiels synchrones et des bascules  $JK$ ; le code du compteur par 5 est binaire pur. Déterminer le logigramme détaillé, y compris la sortie  $Y$  et comparer sa complexité à celui de la figure 4.22; tracer ensuite un chronogramme et un graphe pour tous les états, y compris les états  $\emptyset$ . Existe-t-il des régimes transitoires indésirables de la sortie  $Y$ ? Quel est le code finalement obtenu? Est-il possible de choisir un code particulier du compteur par 5 produisant un code binaire pur pour le compteur complet?

#### 4.4.10 Exercice

La conception d'une horloge digitale (§ 4.1.26 et fig. 4.18) nécessite la synthèse de compteurs par 6 et par 24. Effectuer tout d'abord une décomposition en parallèle du compteur par 6 et réaliser une première variante avec deux compteurs partiels synchrones et des bascules  $JK$ ; une seconde variante sera réalisée à l'aide de compteurs synchrones en anneau (§ 4.2.29 et fig. 4.31) et de bascules  $JK$ . Dans les deux cas on tracera le chronogramme et le graphe de tous les états, y compris les états  $\emptyset$ . Effectuer ensuite une décomposition en parallèle du compteur par 24 et la réaliser avec des compteurs synchrones et des bascules  $D$ .

#### 4.4.11 Exercice

Un compteur par 10 admet les décompositions en série et en parallèle décrites par la relation arithmétique  $10 = 5 \cdot 2 = 2 \cdot 5$ . En partant des deux séquences à cinq états  $Q_C, Q_B, Q_A$  du paragraphe 4.2.26, on demande les graphes des compteurs complets construits selon les schémas des figures 4.43, 4.44 et 4.49.

#### 4.4.12 Définition : décomposition mixte

Lorsqu'un compteur admet une décomposition en série et qu'un (ou plusieurs) compteurs partiels de cette décomposition admettent à leur tour une décomposition en parallèle, on dit que le compteur complet admet une *décomposition mixte*. Inversement, lorsqu'un compteur admet une décomposition en parallèle et qu'un (ou plusieurs) compteurs partiels de cette décomposition admettent à leur tour une décomposition en série, on dit aussi que le compteur complet admet une décomposition mixte.

#### 4.4.13 Notations

Pour distinguer formellement les relations arithmétiques (produits de facteurs) décrivant les décompositions en série et les décompositions en parallèle on introduit les notations suivantes :

- un produit mis entre parenthèses décrit une décomposition en série :

$$p = (p_1 \cdot p_2 \cdot \dots \cdot p_q) \quad (4.61)$$

où  $p_1, p_2, \dots, p_q$  sont des facteurs quelconques; chaque permutation de l'expression (4.61) produit une décomposition différente (§ 4.4.3).

- Un produit mis entre crochets décrit une décomposition en parallèle :

$$p = [p_1 \cdot p_2 \cdot \dots \cdot p_q] \quad (4.62)$$

où  $p_1, p_2, \dots, p_q$  sont des facteurs premiers entre eux; chaque permutation de l'expression (4.62) produit la même décomposition (§ 4.4.6).

#### 4.4.14 Exemple

Etant donné un compteur par 12, on constate qu'il existe trois produits de facteurs qui sont égaux à cette mesure :

$$12 = 6 \cdot 2 = 4 \cdot 3 = 3 \cdot 2 \cdot 2 \quad (4.63)$$

Il découle de l'expression (4.63) sept permutations distinctes qui décrivent autant de décompositions en série (fig. 4.53) :

$$12 = (6 \cdot 2) ; 12 = (2 \cdot 6) \quad (4.64)$$

$$12 = (4 \cdot 3) ; 12 = (3 \cdot 4) \quad (4.65)$$

$$12 = (3 \cdot 2 \cdot 2) ; 12 = (2 \cdot 3 \cdot 2) ; 12 = (2 \cdot 2 \cdot 3) \quad (4.66)$$

Dans l'expression (4.63) il existe un seul produit de facteurs premiers entre eux; on en déduit l'unique décomposition en parallèle (fig. 4.53) :

$$12 = [4 \cdot 3] = [3 \cdot 4] \quad (4.67)$$

La recherche de facteurs premiers entre eux dans les expressions (4.64) à (4.66) met en évidence deux décompositions mixtes (fig. 4.53) décrites par les relations arithmétiques :

$$12 = ([3 \cdot 2] \cdot 2) = ([2 \cdot 3] \cdot 2) ; 12 = (2 \cdot [3 \cdot 2]) = (2 \cdot [2 \cdot 3]) \quad (4.68)$$

tandis que la recherche de produits de facteurs dans l'expression (4.67) entraîne une décomposition mixte (fig. 4.53) décrite par la relation :

$$12 = [(2 \cdot 2) \cdot 3] = [3 \cdot (2 \cdot 2)] \quad (4.69)$$

#### 4.4.15 Description

Tous les compteurs dont les mesures sont des nombres premiers n'admettent aucune décomposition (§ 4.4.3 et § 4.4.6); en d'autres termes toute décomposition (en

série, en parallèle ou mixte) d'un compteur complet peut être réalisée par un assemblage de compteurs partiels dont les mesures sont des nombres premiers. On rappelle alors la liste de ceux d'entre eux qui sont inférieurs à cent [7] (pp. 411 - 412) :

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, \\ 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97 \quad (4.70)$$

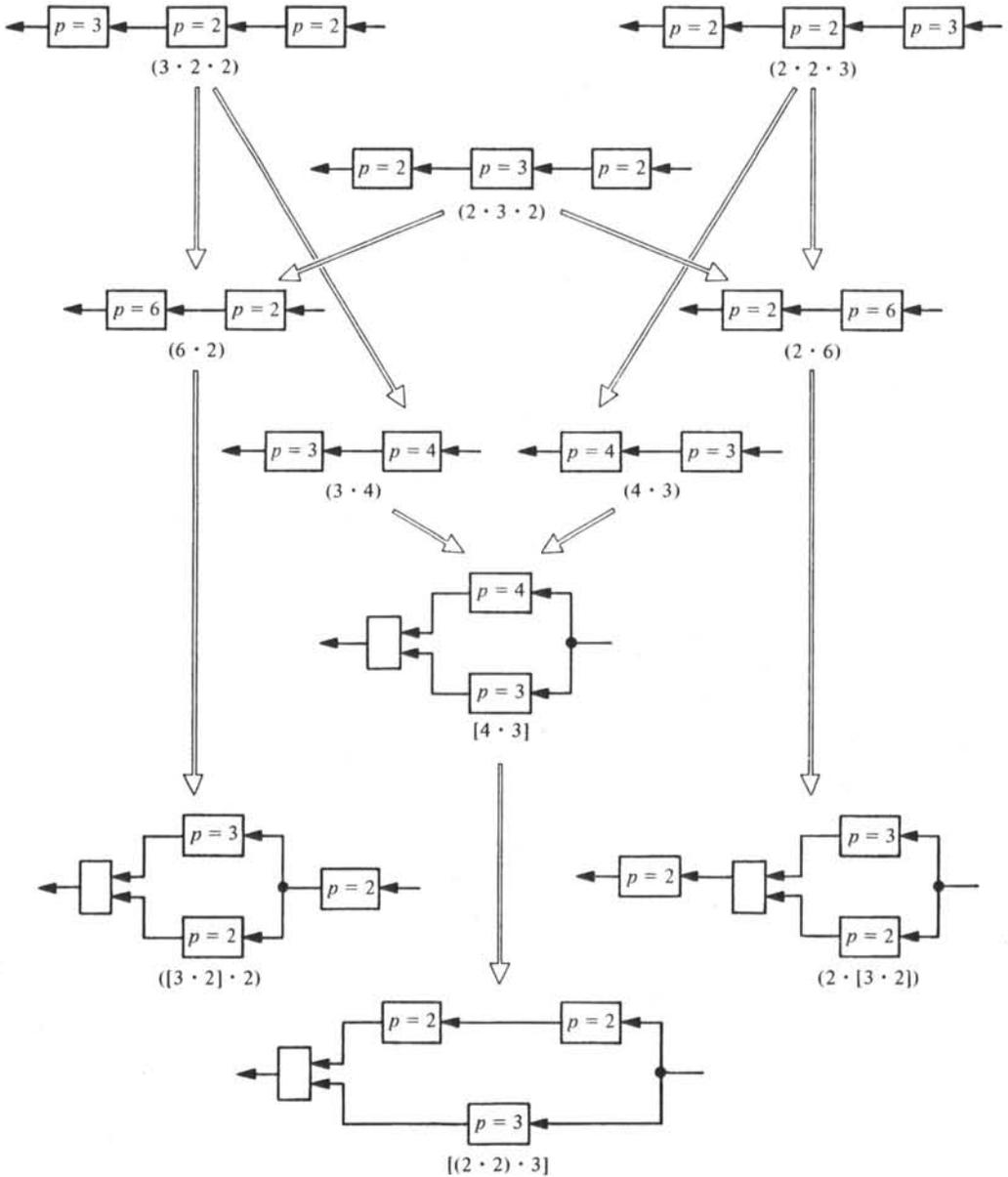


Fig. 4.53

#### 4.4.16 Exercice

On considère le compteur par 12 du paragraphe 4.4.14. En admettant que chaque compteur partiel produit un code binaire pur, on demande de déterminer et de comparer les graphes des états permanents des onze décompositions distinctes du compteur complet (fig. 4.53).

#### 4.4.17 Exercice

Déterminer l'ensemble des décompositions (en série, en parallèle et mixtes) :

- de tous les compteurs dont la mesure est égale ou inférieure à 16;
- du compteur par 24 de l'horloge digitale (§ 4.1.26 et fig. 4.18);
- d'un compteur par 30.

En admettant que chaque compteur par  $p$  comporte le nombre minimal de bascules  $m$  donné par l'expression (4.4) :

$$2^{m-1} < p \leq 2^m \quad (4.71)$$

on demande de calculer dans les trois cas le nombre de bascules nécessité par le compteur complet et par chacune de ses décompositions.

#### 4.4.18 Conclusion

Lorsqu'une décomposition est possible, la synthèse d'un compteur par  $p$  se ramène à la synthèse de compteurs partiels dont les mesures sont inférieures à  $p$  : la décomposition peut donc faciliter le calcul logique en abaissant le nombre des variables traitées.

La décomposition d'un compteur conserve sa mesure mais peut modifier son code; de plus, les décompositions en série et mixte excluent la réalisation de compteurs complets synchrones.

#### 4.4.19 Bibliographie

La décomposition en série fait l'objet d'une étude dans la référence [28] (pp. 171-178, 411) tandis que la publication [47] est entièrement consacrée à la décomposition en parallèle.



## ANALYSE DES SYSTÈMES SÉQUENTIELS SYNCHRONES

### 5.1 PRÉAMBULE

#### 5.1.1 Définitions

Nous appellerons *système séquentiel quasi-synchrone* tout assemblage de bascules bistables et d'un système combinatoire comportant plusieurs variables indépendantes  $x_1, x_2, \dots, x_n$ . La figure 5.1 représente le schéma d'un tel système réalisé avec des bascules  $D$ .

Le *système séquentiel synchrone* est un cas particulier important obtenu lorsque toutes les bascules ont un fonctionnement normal (§ 3.2.6 :  $PR, CLR = 00$ ) et sont attaquées par le même signal d'horloge :  $CK = CK_1 = CK_2 = \dots = CK_m$ . On néglige en général la représentation des entrées asynchrones ( $PR, CLR$ ) dans les logigrammes puisque celles-ci sont sans effet; par ailleurs, le signal d'horloge  $CK$  sera distingué des variables  $x_1, x_2, \dots, x_n$ .

Les  $n$  variables indépendantes  $x_1, x_2, \dots, x_n$  sont les *variables d'entrée* ou *variables primaires*; toute combinaison de leurs valeurs est un *état d'entrée* ou *état primaire* ( $x_1, x_2, \dots, x_n$ ).

Les sorties des  $m$  bascules  $y_1, y_2, \dots, y_m$  (on utilise également  $Q_1, Q_2, \dots, Q_m$ ) sont les *variables internes* ou *variables secondaires*; toute combinaison de leurs valeurs est un *état* (ou *état interne* ou *état secondaire*) symbolisé par  $(y_1, y_2, \dots, y_m) \equiv (Q_1, Q_2, \dots, Q_m)$ .

Toute combinaison des valeurs des  $n$  variables d'entrée et des  $m$  variables internes est un *état total* ( $x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m$ ).

Enfin, toute combinaison des valeurs des  $r$  sorties  $Z_1, Z_2, \dots, Z_r$  est un *état de sortie* ( $Z_1, Z_2, \dots, Z_r$ ).

#### 5.1.2 Applications

Les compteurs quasi-synchrones (§ 4.1.1) sont des systèmes séquentiels quasi-synchrones comportant une seule variable indépendante (le signal d'horloge  $CK$ ) et dont les seules sorties sont celles des bascules. Les compteurs synchrones (§ 4.1.1) constituent un cas particulier des systèmes séquentiels synchrones avec une seule variable indépendante (le signal d'horloge  $CK = CK_1 = CK_2 = \dots = CK_m$ ) mais sans variable d'entrée  $x_1, x_2, \dots, x_n$  et sans variable de sortie  $Z_1, Z_2, \dots, Z_r$ .

#### 5.1.3 Plan

Pour des raisons pédagogiques on traitera principalement les systèmes séquentiels synchrones dont un exemple simple, le compteur réversible, servira d'introduction (sect.

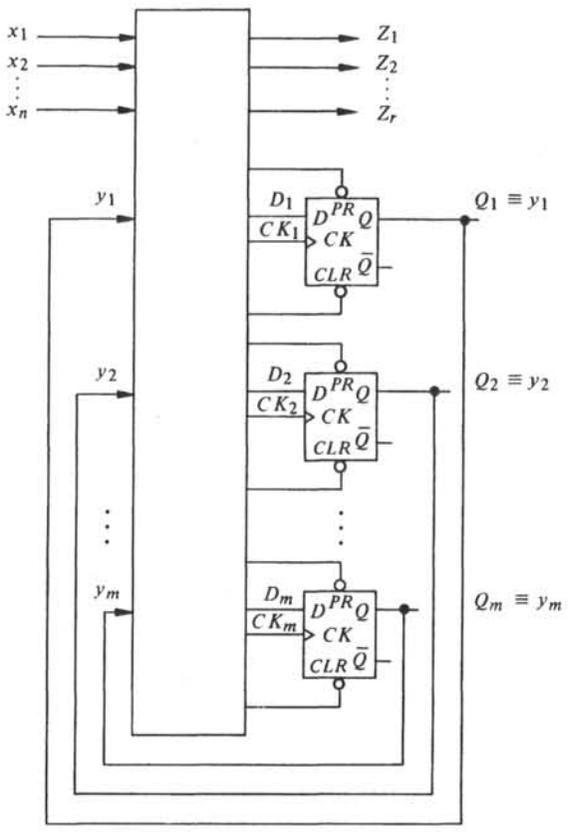


Fig. 5.1

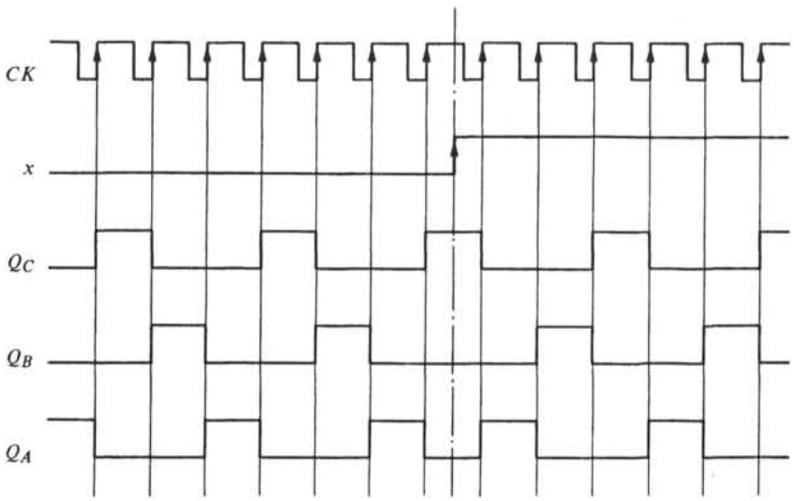


Fig. 5.2

5.1). L'analyse de systèmes synchrones quelconques fait l'objet de la section 5.2 : elle est suivie d'applications aux registres à décalage (sect. 5.3) et à la réalisation séquentielle des systèmes combinatoires itératifs (sect. 5.4). La section 5.5 est consacrée enfin à l'analyse de systèmes séquentiels quasi-synchrones.

Pour des raisons didactiques également, la synthèse des systèmes séquentiels synchrones est exposée à la suite de leur analyse dans le chapitre 6.

#### 5.1.4 Hypothèses

Les systèmes séquentiels synchrones ont un fonctionnement normal ( $PR = CLR = 0$ ) et un signal d'horloge  $CK$  unique (§ 5.1.1) : les instants d'horloge, définis par  $CK = 0$ , sont donc communs à toutes les bascules. On admet alors dans ce chapitre (à l'exception de la section 5.5) que tout système combinatoire est décrit par son modèle combinatoire (§ 1.1.10) et que toute bascule bistable est décrite par son modèle synchrone (§ 3.2.10).

Si ces hypothèses ne sont pas vérifiées on doit recourir au modèle asynchrone du chapitre 7.

#### 5.1.5 Exemple : compteur réversible

On veut effectuer la synthèse d'un système séquentiel synchrone comportant trois bascules  $Q_C, Q_B, Q_A$  du type  $D$  et une variable d'entrée  $x$ . Selon le chronogramme de la figure 5.2 ce système doit produire :

- une séquence  $Q_C, Q_B, Q_A = 100 \rightarrow 010 \rightarrow 001 \rightarrow 100$  pour  $x = 0$ ;
- la séquence inverse  $Q_C, Q_B, Q_A = 100 \rightarrow 001 \rightarrow 010 \rightarrow 100$  pour  $x = 1$ .

Le schéma de la figure 5.3 met en évidence les inconnues du problème : ce sont les fonctions d'excitation  $D_C, D_B$  et  $D_A$ ; les sorties  $Z_1, Z_2, \dots, Z_r$  de la figure 5.1 se confondent dans cet exemple avec celles des bascules ( $Q_C, Q_B$  et  $Q_A$ ).

On appelle généralement *compteur synchrone réversible* par trois (ou *compteur-décompteur*) un tel système séquentiel.

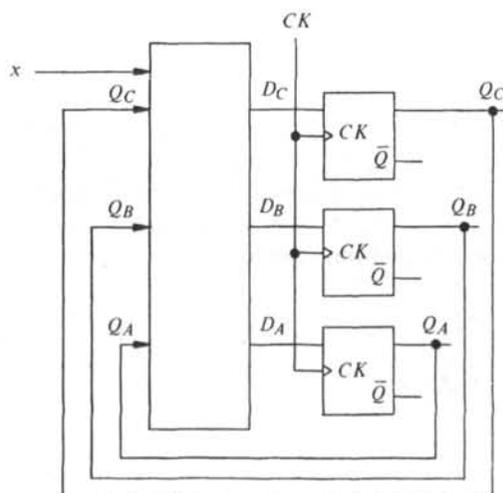


Fig. 5.3

### 5.1.6 Modes de représentation

La donnée du problème, ainsi que le chronogramme de la figure 5.2, font apparaître trois états seulement ( $Q_C, Q_B, Q_A = 100, 010$  et  $001$ ) qui sont représentés par les trois sommets du graphe de la figure 5.4. Les flèches de ce graphe relient les sommets selon les deux séquences définies dans le paragraphe 5.1.5. Pour distinguer ces séquences, on reporte sur chaque flèche la valeur correspondante de la variable d'entrée  $x$ .

A partir du graphe de la figure 5.4 on peut établir une table d'états (fig. 5.6) en procédant de la manière suivante :

- huit lignes représentent les  $2^3$  états présents possibles  $Q_C, Q_B, Q_A$  ;
- pour chaque état présent du graphe ( $Q_C, Q_B, Q_A = 100, 010$  et  $001$ ) on détermine (dans le graphe) les états futurs  $Q_C^+, Q_B^+, Q_A^+$  désignés par  $x = 0$  et par  $x = 1$ ; ceux-ci sont alors reportés dans la table;
- les états futurs non définis sont indifférents : on les représente par le symbole  $\emptyset$ .

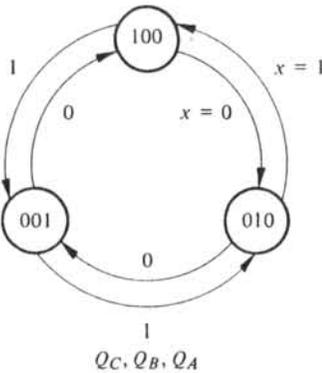


Fig. 5.4

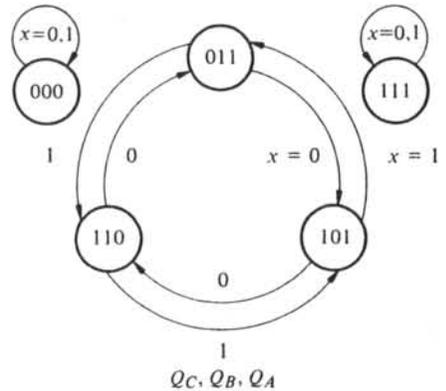


Fig. 5.5

### 5.1.7 Synthèse

L'emploi de bascules  $D$  (3.63) nous permet d'utiliser le calcul abrégé du paragraphe 4.2.11 :

- on construit tout d'abord une table de Karnaugh principale (fig. 5.7) : dans chaque case de celle-ci, c'est-à-dire pour chaque état total ( $x; Q_C, Q_B, Q_A$ ), on reporte l'état futur  $Q_C^+, Q_B^+, Q_A^+$  tiré de la figure 5.6;
- trois tables de Karnaugh secondaires (fig. 5.7) définissent les fonctions d'excitation  $D_C = Q_C^+$ ,  $D_B = Q_B^+$  et  $D_A = Q_A^+$ ; la simplification de celles-ci donne les expressions algébriques suivantes :

$$D_C = x Q_B + \bar{x} Q_A \quad (5.1)$$

$$D_B = x Q_A + \bar{x} Q_C \quad (5.2)$$

$$D_A = x Q_C + \bar{x} Q_B \quad (5.3)$$

Le logigramme de la figure 5.8 illustre une réalisation possible; un trait fort met en évidence le trajet de l'information défini par l'état d'entrée  $x = 0$ .

			$x = 0$			$x = 1$		
$Q_C$	$Q_B$	$Q_A$	$Q_C^*$	$Q_B^*$	$Q_A^*$	$Q_C^*$	$Q_B^*$	$Q_A^*$
0	0	0		$\emptyset$			$\emptyset$	
0	0	1	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0
0	1	1		$\emptyset$			$\emptyset$	
1	0	0	0	1	0	0	0	1
1	0	1		$\emptyset$			$\emptyset$	
1	1	0		$\emptyset$			$\emptyset$	
1	1	1		$\emptyset$			$\emptyset$	

Fig. 5.6

		$Q_C Q_B$			
		00	01	11	10
$Q_A x$	00	$\emptyset = 000$	001	$\emptyset = 011$	010
	01	$\emptyset = 000$	100	$\emptyset = 101$	001
$x$	11	010	$\emptyset = 110$	$\emptyset = 111$	$\emptyset = 011$
	10	100	$\emptyset = 101$	$\emptyset = 111$	$\emptyset = 110$

$Q_C^* \cdot Q_B^* \cdot Q_A^*$

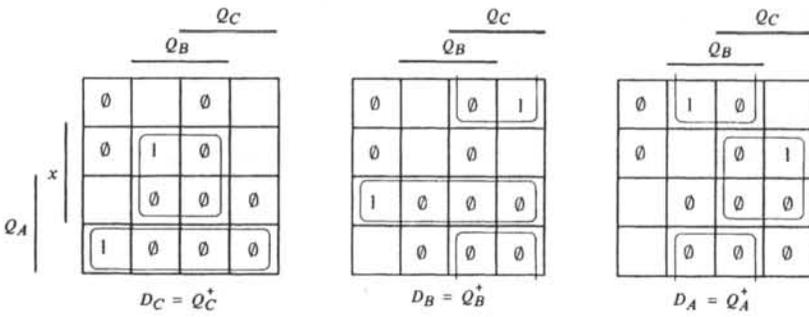


Fig. 5.7

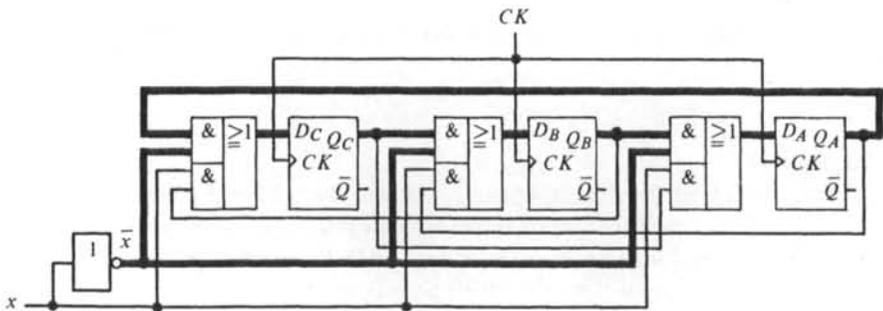


Fig. 5.8

### 5.1.8 Analyse

Le calcul abrégé du paragraphe 4.2.21 nous permet de déterminer dans les tables de Karnaugh secondaires (fig. 5.7) les valeurs binaires des conditions  $\emptyset$ . Les états  $Q_C^+$ ,  $Q_B^+$ ,  $Q_A^+$  recherchés apparaissent dans la table de Karnaugh principale (fig. 5.7) et sont représentés dans le graphe de la figure 5.5. On remarque alors l'existence de trois cycles parasites (§ 4.2.22); à partir d'un état  $\emptyset$  quelconque le compteur réversible de la figure 5.8 ne pourra jamais regagner le graphe de la figure 5.4.

### 5.1.9 Conclusion

La distinction fondamentale entre compteurs et systèmes séquentiels synchrones peut se résumer ainsi : l'état futur  $y_1^+, y_2^+, \dots, y_m^+$  de ceux-ci est défini non seulement par l'état présent  $y_1, y_2, \dots, y_m$  (cas des compteurs) mais encore par l'état d'entrée  $x_1, x_2, \dots, x_n$ . On observe alors dans le graphe (fig. 5.4 par exemple) plusieurs flèches quittant le même sommet; dans la table d'états (fig. 5.6 par exemple) il existe autant de colonnes que d'états d'entrée.

### 5.1.10 Exercice

A quelles conditions les bascules  $D$  du compteur réversible (fig. 5.8) valident-elles le modèle quasi-synchrone du paragraphe 3.2.3? Démontrer ensuite que les bascules d'un système séquentiel synchrone quelconque (§ 5.1.1) vérifient ce modèle à condition que toute variation de l'état d'entrée  $x_1, x_2, \dots, x_n$  soit interdite pendant les instants d'horloge ( $CK = 0$ ).

### 5.1.11 Exercice

En modifiant uniquement la fonction d'excitation  $D_C$  du compteur réversible (fig. 5.8) on demande de supprimer tous les cycles parasites représentés par la figure 5.5.

En partant de la table d'états du même compteur (fig. 5.6), effectuer la synthèse et l'analyse pour des bascules  $JK$  : les calculs seront tabulaire (§ 4.2.8) et algébriques (§ 4.2.9 et 4.2.19).

### 5.1.12 Exercice

Effectuer l'analyse (tables et graphes des états) des deux logigrammes des figures 5.9 et 5.10.

### 5.1.13 Exercice

Effectuer la synthèse d'un compteur synchrone réversible par 10 dans le code binaire pur (fig. 8.1) dont la variable d'entrée est  $x$  ( $x = 0$  : comptage direct;  $x = 1$  : comptage inverse) et dont les bascules sont du type  $D$ . En constatant que  $10 = 5 \cdot 2$  réaliser ensuite la décomposition en parallèle de la figure 5.11 (§ 4.4.4 à 4.4.6) : comparer le logigramme et le code ainsi obtenus à ceux du compteur réversible non décomposé.

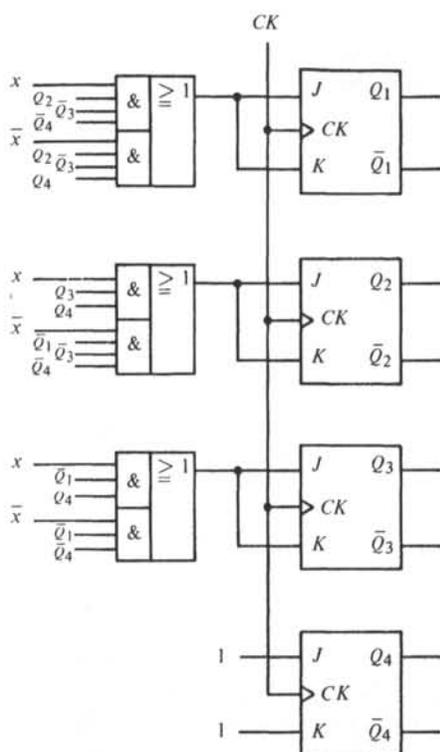


Fig. 5.9

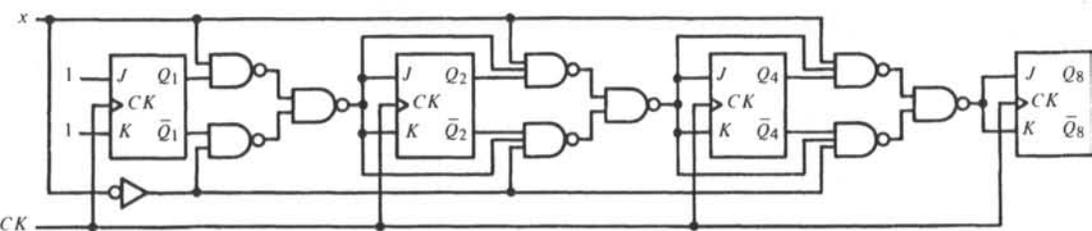


Fig. 5.10

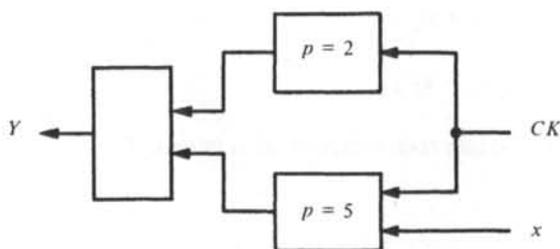


Fig. 5.11

### 5.1.14 Exercice

Effectuer la synthèse et l'analyse des états  $\emptyset$  d'un compteur synchrone réversible par 6 dans le code binaire pur (fig. 8.1) avec une variable d'entrée  $x$  ( $x = 0$  : comptage direct;  $x = 1$  : comptage inverse). Une variante sera calculée pour des bascules  $D$ , une autre pour des bascules  $JK$ .

## 5.2 SYSTÈMES SÉQUENTIELS SYNCHRONES

### 5.2.1 Analyse

Le but de l'analyse est la détermination de modes de représentation (table d'états, graphe des états) à partir d'un logigramme donné.

### 5.2.2 Méthode

En considérant le schéma de la figure 5.1 (avec  $PR = CLR = 0$  et  $CK = CK_1 = CK_2 = \dots = CK_m$ ) on procède de la façon suivante :

- l'analyse du système combinatoire donne une expression algébrique des sorties  $Z_1, Z_2, \dots, Z_r$  et des excitations  $D_1, D_2, \dots, D_m$  en fonction des variables d'entrée  $x_1, x_2, \dots, x_n$  et des variables internes  $y_1, y_2, \dots, y_m$ ;
- l'équation caractéristique des bascules utilisées (dans ce cas :  $y^+ \equiv Q^+ = D$ ) exprime l'état futur  $y_j^+$  ( $j = 1, 2, \dots, m$ ) en fonction des variables  $x_1, x_2, \dots, x_n$  et  $y_1, y_2, \dots, y_m$ ;
- on calcule pour tout état total ( $x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m$ ) la valeur des fonctions de sortie  $Z_1, Z_2, \dots, Z_r$  et des états futurs  $y_1^+, y_2^+, \dots, y_m^+$ ; la représentation de ces valeurs dans une table ou un graphe conclut la méthode.

### 5.2.3 Exemple : discriminateur du sens de rotation

Le logigramme de la figure 5.12 est celui d'un *discriminateur du sens de rotation* [67] (pp. 36 - 38, 51 - 55) dont la sortie  $Z$  et les excitations  $D_1, D_2$  s'expriment sous la forme algébrique suivante :

$$Z = \bar{x}_1 x_2 y_1 + x_1 \bar{x}_2 \bar{y}_1 + \bar{x}_1 \bar{x}_2 y_2 + x_1 x_2 \bar{y}_2 \quad (5.4)$$

$$D_1 = x_1 x_2 + x_1 y_1 + x_2 y_1 \quad (5.5)$$

$$D_2 = \bar{x}_1 x_2 + \bar{x}_1 y_2 + x_2 y_2 \quad (5.6)$$

On déduit de l'équation caractéristique de la bascule  $D$  (3.52) les expressions des états futurs :

$$y_1^+ \equiv Q_1^+ = D_1 = x_1 x_2 + x_1 y_1 + x_2 y_1 \quad (5.7)$$

$$y_2^+ \equiv Q_2^+ = D_2 = \bar{x}_1 x_2 + \bar{x}_1 y_2 + x_2 y_2 \quad (5.8)$$

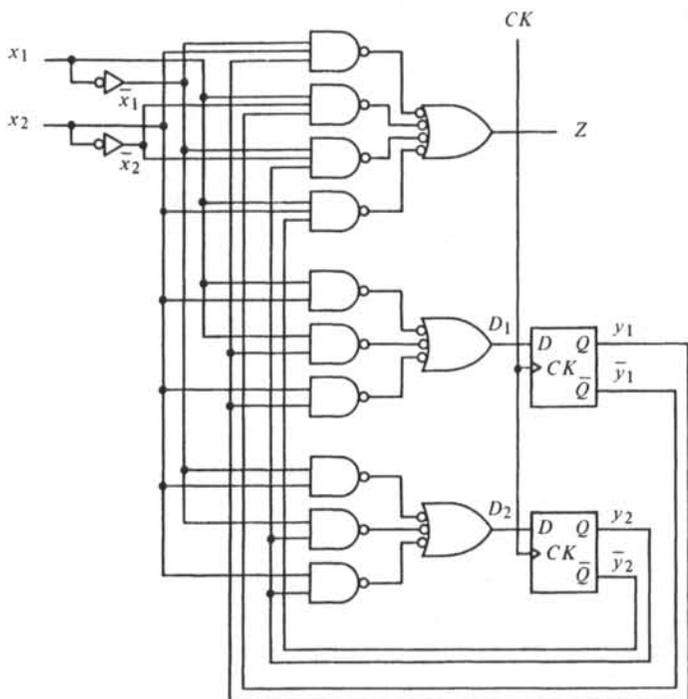


Fig. 5.12

#### 5.2.4 Mode de représentation : table d'états

La table d'états d'un système séquentiel synchrone comporte dans le cas général (fig. 5.1) :

- $2^n$  colonnes définies par les états d'entrée  $(x_1, x_2, \dots, x_n)$ ;
- $2^m$  lignes définies par les états internes  $(y_1, y_2, \dots, y_m)$ ;
- $2^{n+m}$  cases définies par les états totaux  $(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m)$ .

Dans chaque case (c'est-à-dire pour chaque état total) on reporte les états futurs  $y_1^+$ ,  $y_2^+$ , ...,  $y_m^+$  et les états de sortie  $Z_1, Z_2, \dots, Z_r$ .

La table d'états du discriminateur (fig. 5.13) comporte donc  $2^2 = 4$  colonnes;  $2^2 = 4$  lignes et  $2^4 = 16$  cases; dans chacune de celles-ci figurent un état futur  $y_1^+, y_2^+$  et un état de sortie  $Z$ . Le calcul de ces états est facilité si l'on choisit une disposition des variables  $x_1, x_2, y_1$  et  $y_2$  analogue à celle de la table de Karnaugh (§ 2.2.1) : à titre d'exemple, la représentation de la fonction  $y_1^+$  (5.7) est détaillée dans la figure 5.14. En procédant de façon identique pour les fonctions  $y_2^+$  (5.8) et  $Z$  (5.4) on complète la figure 5.13.

Chaque état présent  $(y_1, y_2)$  ou futur  $(y_1^+, y_2^+)$  peut être représenté par un symbole quelconque (un chiffre décimal, une lettre de l'alphabet, etc...). Deux exemples d'un tel codage sont illustrés par les figures 5.15 et 5.16 : les tables obtenues sont des *tables d'états non assignées*, tandis qu'on appelle *table d'états assignée* celle de la figure 5.13.

$y_1^+ y_2^+, Z$	$x_1 x_2$			
	00	01	11	10
00	00,0	01,0	10,1	00,1
01	01,1	01,0	11,0	00,1
11	01,1	11,1	11,0	10,0
10	00,0	11,1	10,1	10,0

$y_1 y_2$

Fig. 5.13

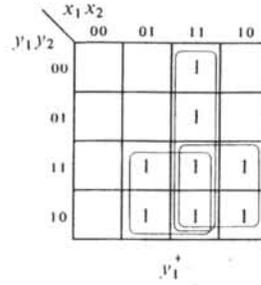


Fig. 5.14

$y_1^+ y_2^+, Z$	$x_1 x_2$			
	00	01	11	10
$00 \equiv a$	$a, 0$	$b, 0$	$d, 1$	$a, 1$
$01 \equiv b$	$b, 1$	$b, 0$	$c, 0$	$a, 1$
$11 \equiv c$	$b, 1$	$c, 1$	$c, 0$	$d, 0$
$10 \equiv d$	$a, 0$	$c, 1$	$d, 1$	$d, 0$

$y_1 y_2$

Fig. 5.15

$y_1^+ y_2^+, Z$	$x_1 x_2$			
	00	01	11	10
$00 \equiv 1$	1,0	2,0	4,1	1,1
$01 \equiv 2$	2,1	2,0	3,0	1,1
$11 \equiv 3$	2,1	3,1	3,0	4,0
$10 \equiv 4$	1,0	3,1	4,1	4,0

$y_1 y_2$

Fig. 5.16

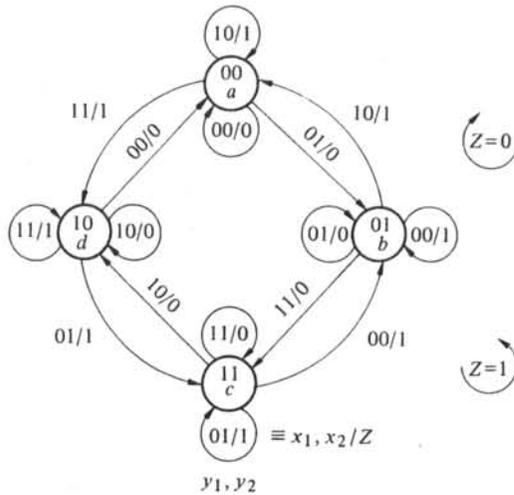


Fig. 5.17

**5.2.5 Mode de représentation : graphe des états**

Dans le cas général (fig. 5.1) le graphe des états d'un système séquentiel synchrone comporte :

- $2^m$  sommets définis par les états internes ( $y_1, y_2, \dots, y_m$ );
- $2^n$  flèches quittant chaque sommet et définies par les états d'entrée ( $x_1, x_2, \dots, x_n$ ).

Sur chaque flèche quittant un sommet (c'est-à-dire pour chaque état total) on indique l'état d'entrée et l'état de sortie correspondants :  $x_1, x_2, \dots, x_n / Z_1, Z_2, \dots, Z_r$ ; le sommet d'arrivée de la flèche définit l'état futur  $(y_1^+, y_2^+, \dots, y_m^+)$ .

Les tables d'états assignée (fig. 5.13) et non assignée (fig. 5.15) nous permettent de construire le graphe de la figure 5.17 : à l'intérieur de chaque sommet on a représenté l'état  $y_1, y_2$  sous ses formes binaire et alphabétique.

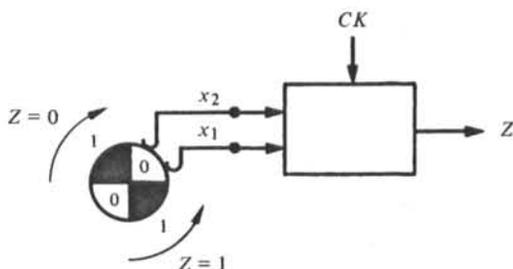


Fig. 5.18

### 5.2.6 Mode de représentation : langage naturel

A partir du schéma de la figure 5.18 et du graphe de la figure 5.17 il est possible de décrire dans un langage naturel le comportement du système étudié. Mécaniquement on constate que le dispositif est constitué d'un arbre tournant, muni de quatre secteurs isolés entre eux et reliés alternativement aux constantes logiques 0 et 1; deux frotteurs, dont l'espacement est inférieur au plus petit des arcs des secteurs, sont les entrées  $x_1$  et  $x_2$ .

Logiquement, on démontre que l'état de sortie  $Z$  indique le sens de rotation. Supposons que l'arbre soit animé d'un mouvement dans le sens des aiguilles d'une montre : les états d'entrée se succèdent dans la séquence  $x_1, x_2 = 00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$  et la sortie  $Z$  prend la valeur 0. La séquence inverse  $x_1, x_2 = 00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$  produit l'état  $Z = 1$ .

### 5.2.7 Définition : état total stable

On appelle *état total stable* d'un système séquentiel synchrone tout état total vérifiant la relation :

$$(x_1, x_2, \dots, x_n ; y_1, y_2, \dots, y_m) = (x_1, x_2, \dots, x_n ; y_1^+, y_2^+, \dots, y_m^+) \quad (5.9)$$

Pour un état d'entrée  $x_1, x_2, \dots, x_n$  donné l'état futur  $y_1^+, y_2^+, \dots, y_m^+$  est donc égal à l'état présent  $y_1, y_2, \dots, y_m$ .

En examinant la table d'états du discriminateur (fig. 5.13) on repère les états totaux stables qui vérifient la relation :

$$x_1, x_2 ; y_1, y_2 = x_1, x_2 ; y_1^+, y_2^+ \quad (5.10)$$

Ces états sont encadrés dans la table (fig. 5.19); ils sont représentés dans le graphe (fig. 5.17) par des flèches reliant un sommet à lui-même.

$y_1^+ y_2^+$	$x_1 x_2$			
	00	01	11	10
00	⊙00	→01	10	⊙00
01	⊙01	⊙01	11	00
11	01	⊙11	⊙11	10
10	00	11	⊙10	⊙10

$y_1 y_2$

Fig. 5.19

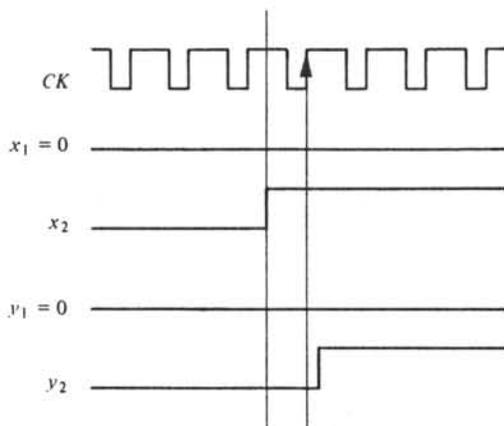


Fig. 5.20

On vérifie aisément la propriété suivante : seule une variation de l'état d'entrée  $x_1, x_2, \dots, x_n$  déloge le système séquentiel d'un état total stable; une telle variation se traduit par un déplacement horizontal dans la table d'états.

### 5.2.8 Définition : état total instable

Tout état total qui n'est pas stable (§ 5.2.7) est un *état total instable*. Il découle de la relation (5.9) que, pour un état d'entrée donné, l'état futur est différent de l'état présent : une variable interne au moins a changé de valeur.

Dans la table du discriminateur (fig. 5.19) tous les états totaux non encadrés sont instables. On relève par exemple pour  $x_1, x_2 = 01$  :

$$y_1, y_2 = 00 ; y_1^+, y_2^+ = 01 \neq y_1, y_2 \quad (5.11)$$

La propriété suivante est vérifiée : il suffit d'une variation du signal d'horloge (c'est-à-dire une montée de la variable  $CK$ ) pour quitter un état total instable et gagner un nouvel état total, stable ou instable : une telle variation se traduit par un déplacement vertical dans la table d'états.

### 5.2.9 Application

Dans l'exemple du discriminateur du sens de rotation, on admet que la fréquence du signal d'horloge  $CK$  est toujours plus élevée que celle des signaux d'entrée  $x_1$  et  $x_2$ ; il en découle le comportement suivant (fig. 5.17, 5.19 et 5.20) :

- le système occupe tout d'abord un état total stable ( $x_1, x_2 ; y_1, y_2 = 00, 00$  par exemple); les variations successives du signal d'horloge  $CK$  sont sans effet;
- une variation de l'état d'entrée ( $x_1, x_2 = 01$ ) entraîne instantanément le système dans un état total instable ( $x_1, x_2 ; y_1, y_2 = 01, 00$  avec  $y_1^+, y_2^+ = 01$ );
- à la prochaine montée du signal d'horloge le discriminateur gagne un nouvel état total ( $x_1, x_2 ; y_1, y_2 = 01, 01$ ); celui-ci est stable et ne sera abandonné qu'à la suite d'une nouvelle variation de l'état d'entrée.

### 5.2.10 Bibliographie

Le concept de la table d'états est essentiellement dû à Huffman [65] et son application aux systèmes séquentiels synchrones a été proposée par Mealy [66]. Des exposés classiques sur l'analyse de ces systèmes font l'objet des références [10] (pp. 226 - 230, 241 - 245), [11] (pp. 244 - 246), [12] (pp. 178 - 185) et [45] (pp. 147 - 164, 185 - 189).

### 5.2.11 Exercice

A quelles conditions les bascules  $D$  du discriminateur (fig. 5.12) valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ? Comparer les résultats de cet exercice avec ceux du paragraphe 5.1.10.

La fréquence du signal d'horloge  $CK$  peut-elle être plus basse que celle des signaux d'entrée  $x_1$  et  $x_2$  ? Sinon, montrer qu'il existe une condition liant ces fréquences pour que le système séquentiel fonctionne correctement.

### 5.2.12 Exercice

On demande une réponse circonstanciée à la lettre suivante [58] (pp. 249 - 254), [59] (pp. 75 - 76, 334 - 335) :

“ Cher ami,

Il y a quelque temps, j'ai acheté cette vieille maison, mais je me suis aperçu qu'elle était hantée par deux bruits d'outre-tombe : un Chant grivois, et un Rire sardonique. Elle est donc pratiquement inhabitable. J'ai cependant quelque espoir, car l'expérience m'a montré que leur comportement obéissait à certaines lois, obscures mais infaillibles, et que je pouvais le modifier en jouant de l'Orgue ou en brûlant de l'Encens.

Chaque minute, chaque bruit est présent ou absent. Ce que chacun d'eux fera au cours de la minute suivante dépend ainsi de ce qui s'est passé pendant la minute présente :

- le Chant conservera le même état (présent ou absent), sauf si, pendant la minute présente, l'Orgue jouait sans que le Rire se fasse entendre, auquel cas le Chant prendra l'état opposé;
- quant au Rire, si l'Encens brûlait, il se fera entendre ou non selon que le Chant résonnait ou non (de sorte que le Rire imite le Chant avec une minute de retard). Toutefois, si l'Encens ne brûlait pas, le Rire fera le contraire de ce que faisait le Chant.

Depuis plusieurs minutes, j'entends à la fois le Rire et le Chant. Vous m'obligeriez en m'indiquant à quelles manipulations d'Orgue et d'Encens je dois me livrer pour rétablir le calme définitivement et dans les plus brefs délais."

On décrira la maison hantée par un système séquentiel synchrone à deux variables d'entrée  $O$  et  $E$  (Orgue et Encens), deux variables internes  $y_1 = C$  et  $y_2 = R$  (Chant et Rire) qui sont aussi les sorties et deux bascules  $D$  ( $D_1 = y_1^+ = C^+$  et  $D_2 = y_2^+ = R^+$ ). La solution, exprimée en langage naturel, sera justifiée par une table d'états, un graphe des états et un chronogramme.

### 5.2.13 Exercice

Trois pièces de monnaie, numérotées de 1 à 3, présentent respectivement (dans l'ordre 1, 2, 3) les côtés pile, face, pile. Nous résumerons cette description par l'abréviation *pdf*. Le joueur peut retourner les pièces en se conformant à la règle suivante : à chaque coup l'une des trois pièces doit rester immobile, les deux autres doivent être retournées en même temps. On pourra caractériser un coup par le numéro de la pièce intacte. Le but du jeu est d'atteindre la situation *fff* (trois faces) à partir de *pdf* en une suite de trois coups exactement.

On demande d'interpréter les configurations *pdf*, *ppp*, etc... comme autant d'états d'un système séquentiel synchrone et de dessiner le graphe correspondant. On demande enfin de montrer sur ce graphe que le problème posé admet une solution [60] (p. 13).

### 5.2.14 Exercice

Déterminer les tables et les graphes des états de deux systèmes séquentiels synchrones construits avec des bascules  $D$  et décrits par les équations suivantes :

$$\begin{aligned} D_1 &= MAJ(\bar{x}_1, \bar{x}_2, y_1); D_2 = \bar{y}_2(\bar{x}_1 \oplus y_1) + y_2(x_2 \oplus y_1); \\ Z &= \bar{y}_2(x_2 \oplus y_1) + y_2(\bar{x}_1 \oplus y_1) \end{aligned} \quad (5.12)$$

$$\begin{aligned} D_1 &= x_1 \bar{y}_2 \bar{y}_3 + \bar{x}_1 \bar{x}_2 y_3 + \bar{x}_1 y_2 y_3 + x_1 x_2 \bar{y}_3 + x_2 y_1 \bar{y}_2 + \bar{x}_2 y_1 y_2; \\ D_2 &= y_2 y_3 + \bar{x}_2 y_3 + \bar{x}_1 \bar{x}_2 y_1 y_2 + x_1 \bar{x}_2 \bar{y}_1 y_2 + x_1 \bar{y}_1 y_3 + \bar{x}_1 y_1 y_3; \\ D_3 &= x_2; Z = y_1 \end{aligned} \quad (5.13)$$

Comparer le comportement de ces systèmes avec celui du discriminateur du sens de rotation (fig. 5.12 et 5.17).

### 5.2.15 Exercice

Analyser les deux systèmes séquentiels définis par les équations suivantes (bascules  $T$ , puis bascules  $SR$ ) et comparer leurs graphes des états :

$$T_1 = x_2 \bar{y}_1 + x_1 \bar{y}_1 + \bar{x}_1 \bar{x}_2 y_1; T_2 = \bar{x}_1 x_2 \bar{y}_1 + y_2; Z = x_1 \bar{x}_2 y_2 \quad (5.14)$$

$$\begin{aligned} S_1 &= x_1 + x_2; R_1 = \bar{x}_1 \bar{x}_2; S_2 = \bar{x}_1 x_2 \bar{y}_1; \\ R_2 &= y_2; Z = x_1 \bar{x}_2 y_2 \end{aligned} \quad (5.15)$$

Décrire le comportement en langage naturel.

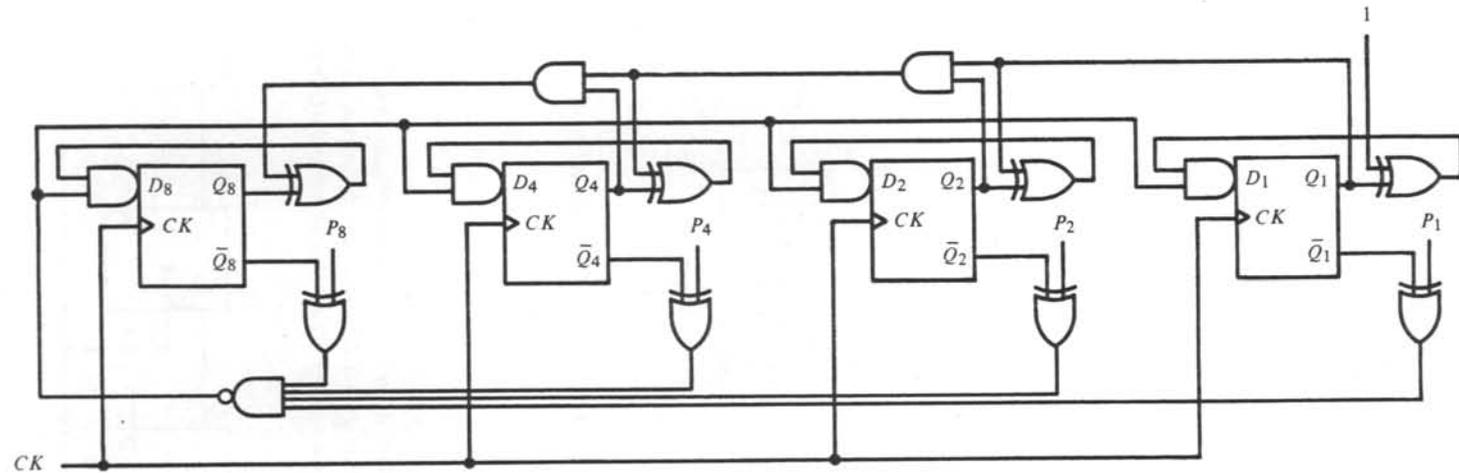


Fig. 5.21

### 5.2.16 Exercice

Analyser le logigramme de la figure 5.21 (graphe) et démontrer que ce système est un *compteur synchrone programmable* dont le code est binaire pur et dont la mesure  $p$  est définie par l'état d'entrée  $P_8, P_4, P_2, P_1$  de la figure 5.22. Est-il possible de généraliser l'emploi d'une telle structure pour obtenir une mesure supérieure à 16 ?

$P_8$	$P_4$	$P_2$	$P_1$	$p$
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

Fig. 5.22

## 5.3 REGISTRES À DÉCALAGE

### 5.3.1 Définition

Le logigramme de la figure 5.23 et les schémas de la figure 5.24 représentent un système séquentiel synchrone réalisé par la mise en série de trois bascules  $D$  : on dit qu'il s'agit d'un *registre à décalage* de trois bits.

Un registre à décalage de  $m$  bits est un assemblage de  $m$  bascules  $D$  connectées en série selon le modèle de la figure 5.23. D'après la définition du paragraphe 2.7.5, un tel registre est un système séquentiel itératif; chaque sous-système est une bascule  $D$ .

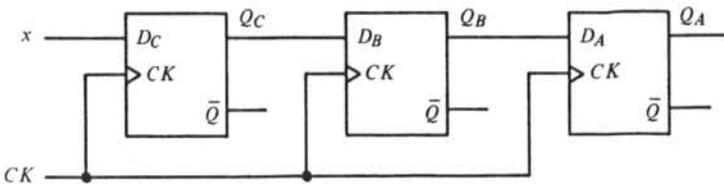


Fig. 5.23

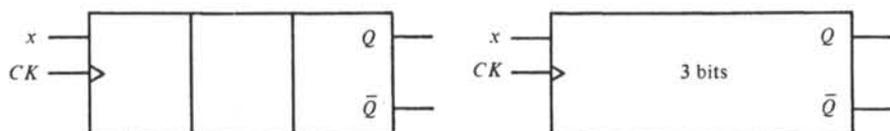


Fig. 5.24

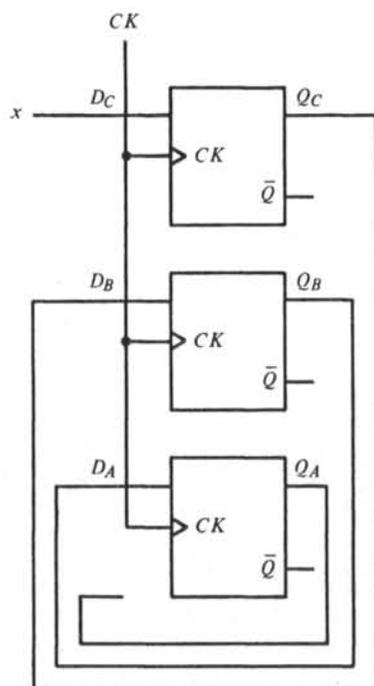


Fig. 5.25

### 5.3.2 Analyse

Le logigramme de la figure 5.23 peut être mis sous la forme du schéma des systèmes séquentiels (fig. 5.1); on obtient la figure 5.25 qui est décrite par les équations suivantes :

$$D_C = x ; D_B = Q_C ; D_A = Q_B \quad (5.16)$$

L'équation caractéristique (3.52) de la bascule  $D$  nous permet d'écrire :

$$Q_C^+ = x ; Q_B^+ = Q_C ; Q_A^+ = Q_B \quad (5.17)$$

On peut alors dresser la table d'états assignée de la figure 5.26 dans laquelle les états totaux stables sont encadrés; les sorties  $Z_1, Z_2, \dots, Z_r$  se confondent avec les sorties  $Q_C, Q_B, Q_A$  des bascules et ne sont donc pas représentées. Le graphe des états tiré de la table est illustré par la figure 5.27.

$Q_C^+$	$Q_B^+$	$Q_A^+$	$x$	
			0	1
0	0	0	0 0 0	1 0 0
0	0	1	0 0 0	1 0 0
0	1	0	0 0 1	1 0 1
0	1	1	0 0 1	1 0 1
1	0	0	0 1 0	1 1 0
1	0	1	0 1 0	1 1 0
1	1	0	0 1 1	1 1 1
1	1	1	0 1 1	1 1 1

$Q_C \quad Q_B \quad Q_A$

Fig. 5.26

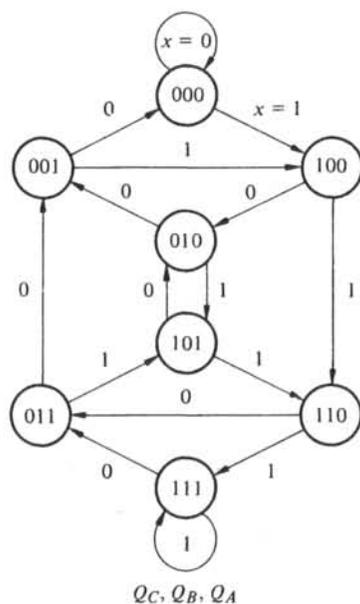


Fig. 5.27

### 5.3.3 Description

Admettons qu'à l'instant d'horloge  $t_0$  le registre à décalage (fig. 5.23) soit dans l'état  $Q_C, Q_B, Q_A = 000$  et supposons que les états d'entrée soient définis pendant trois instants d'horloge successifs :

$$x(t_0) = 1 \quad ; \quad x(t_1) = 1 \quad ; \quad x(t_2) = 0 \quad (5.18)$$

En tenant compte des propriétés de la bascule  $D$  (§ 3.3.3) il est aisé de décrire le comportement du registre de l'instant  $t_0$  jusqu'à l'instant  $t_3$  : la figure 5.28 illustre le décalage de gauche à droite de l'information, à la cadence d'un bit par instant d'horloge.

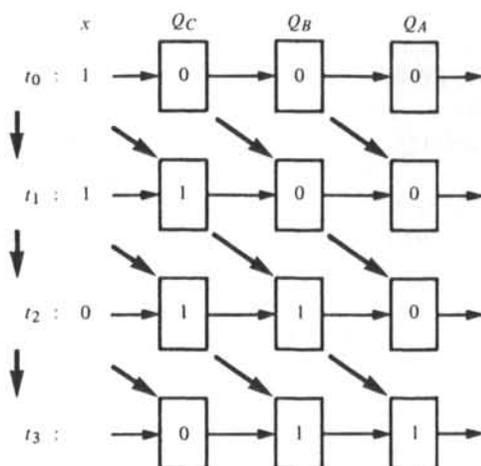


Fig. 5.28

En inversant le schéma habituel du registre (fig. 5.28), on obtient un décalage de droite à gauche qui a l'avantage de représenter l'information telle qu'elle apparaît dans un chronogramme (fig. 5.29). Selon le problème traité, nous choisirons l'un ou l'autre des sens de décalage.

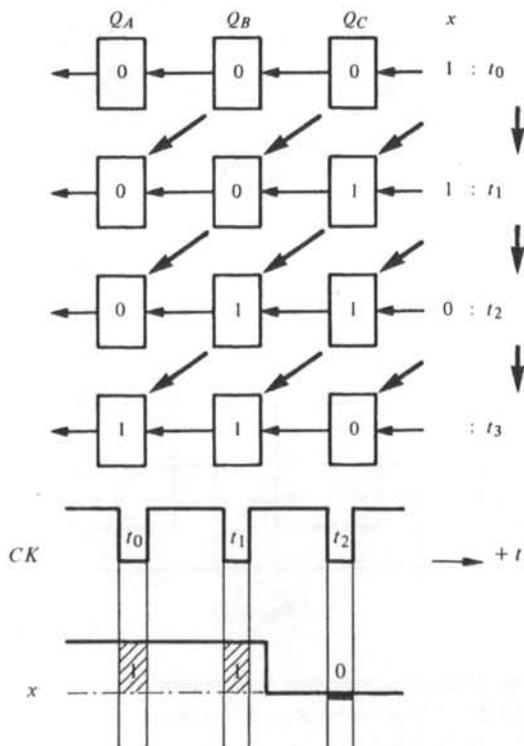


Fig. 5.29

### 5.3.4 Applications

La description du paragraphe 5.3.3 nous permet de construire directement le graphe des états d'un registre à décalage donné (fig. 5.27); à partir de n'importe quel sommet ( $Q_C, Q_B, Q_A = 100$  par exemple) il existe deux flèches ( $x = 0$  et  $x = 1$ ) conduisant à deux autres sommets ( $Q_C^+, Q_B^+, Q_A^+ = 010$  pour  $x = 0$  et  $Q_C^+, Q_B^+, Q_A^+ = 110$  pour  $x = 1$ ).

Les logigrammes des figures 5.30 et 5.32 sont ceux des registres de deux et quatre bits; on a construit les graphes des figures 5.31 et 5.33 sans établir les tables d'états correspondantes.

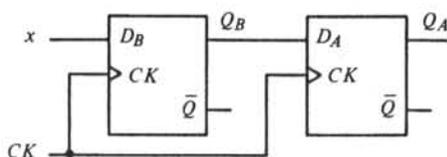


Fig. 5.30

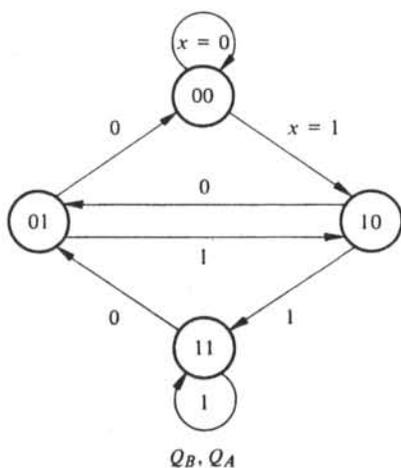


Fig. 5.31

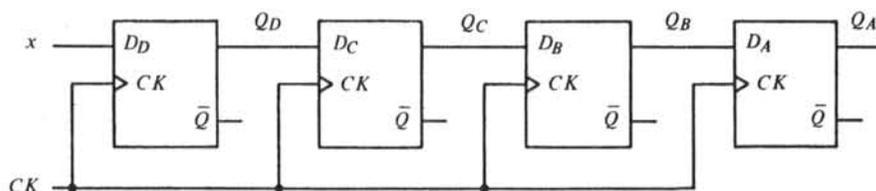
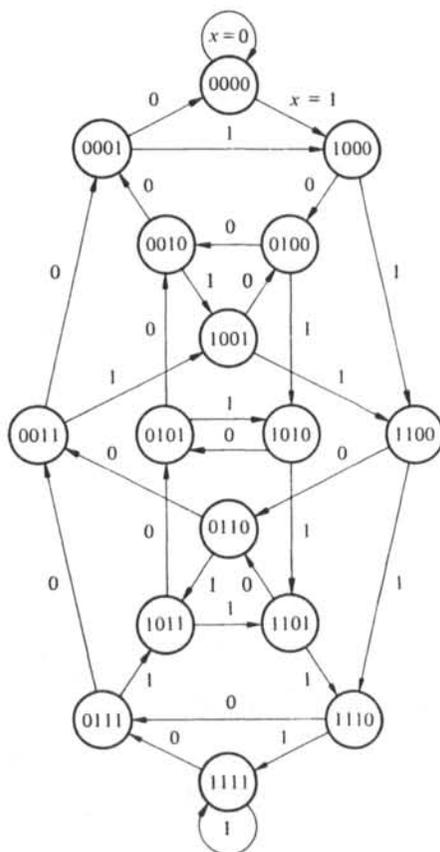


Fig. 5.32

### 5.3.5 Méthode : synthèse de compteurs synchrones

Il est possible de réaliser certains compteurs synchrones avec des registres à décalage [40] (pp. 147 - 182), [54] (pp. 285 - 307), [61]. A titre d'exemple, on choisit un compteur par six dont le code n'est pas précisé. La méthode de synthèse est la suivante :



$Q_D, Q_C, Q_B, Q_A$

Fig. 5.33

- le nombre  $m$  de bascules est calculé par la relation (4.4) :

$$2^2 < 6 < 2^3 = 2^m$$

Le registre à décalage est donc celui de la figure 5.23 ( $m = 3$ ).

- On recherche dans le graphe des états de ce registre (fig. 5.27) un chemin fermé reliant six sommets distincts : un tel chemin représente la séquence des six états du compteur. Plusieurs solutions sont possibles; la figure 5.34 illustre l'une d'entre elles.
- Le chemin retenu (fig. 5.34) détermine pour chaque état  $Q_C, Q_B, Q_A$  une valeur de l'entrée  $x$ ; celle-ci est donc une fonction des variables  $Q_C, Q_B$  et  $Q_A$  qu'on peut représenter dans une table de vérité (fig. 5.35) et dans une table de Karnaugh (fig. 5.36). Après simplification on obtient l'expression algébrique :

$$x = \bar{Q}_C \bar{Q}_A + \bar{Q}_B Q_A \quad (5.19)$$

dont une réalisation possible est celle de la figure 5.37.

- L'examen de la table de Karnaugh (fig. 5.36) indique les valeurs binaires des conditions  $\emptyset$  : on peut alors déterminer le comportement du compteur à partir des états  $\emptyset$  et compléter en trait discontinu le graphe des états (fig. 5.34); il n'existe aucun cycle parasite.

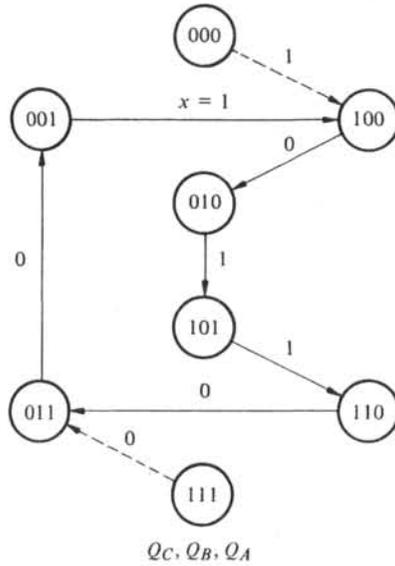


Fig. 5.34

$Q_C$	$Q_B$	$Q_A$	$x$
0	0	0	$\emptyset$
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	$\emptyset$

Fig. 5.35

		$Q_C Q_B$			
		00	01	11	10
$Q_A$	0	$\emptyset$	1		
	1	1		$\emptyset$	1
		$x$			

Fig. 5.36

Pour une mesure donnée, la variété des codes est limitée; il peut être nécessaire de réaliser un système combinatoire complémentaire si l'on veut obtenir un code particulier (le code binaire pur, par exemple).

Tout compteur dont la mesure est inférieure ou égale à  $2^m$  peut être réalisé avec un registre de  $m$  bits; la démonstration de cette propriété est malaisée : elle n'est pas détaillée ici.

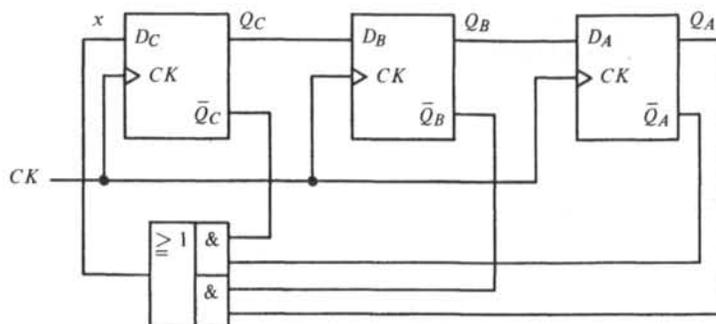


Fig. 5.37

### 5.3.6 Conclusion

L'étude des registres à décalage illustre les limites des modes de représentation introduits dans ce chapitre : l'analyse d'un registre de 10 bits, par exemple, impliquerait le dessin d'un graphe à  $2^{10} = 1024$  sommets ou le tracé d'une table d'états à 1024 lignes. Pour calculer des systèmes séquentiels comportant un nombre élevé de variables, on cherche soit à décomposer ceux-ci en sous-systèmes plus simples, soit à introduire de nouveaux modes de représentation : c'est notamment le cas du *langage de transfert de registres* [12] (pp. 237 - 267), [24] (pp. 494 - 511), [63], [64].

### 5.3.7 Exercice

Analyser le logigramme de la figure 5.38 et comparer le graphe ou la table d'états à ceux des figures 5.31 et 5.33.

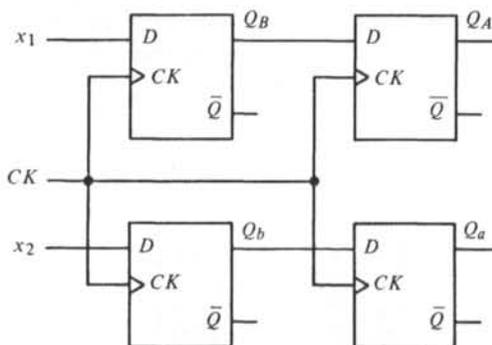


Fig. 5.38

### ■ 5.3.8 Exercice

Chercher dans le graphe de la figure 5.27 toutes les solutions possibles d'un compteur par 6 (l'une d'entre elles étant déjà illustrée par la figure 5.34). Déterminer dans chaque cas la fonction  $x$ , puis simplifier et analyser les états  $\emptyset$ ; corriger au besoin cette fonction pour éliminer tout cycle parasite.

Répéter le même exercice pour trouver tous les compteurs par 10 réalisables avec un registre de 4 bits (fig. 5.32 et 5.33).

## 5.3.9 Exercice

Analyser le compteur synchrone de la figure 5.39 (graphe des états) et corriger le logigramme pour éliminer les éventuels cycles parasites.

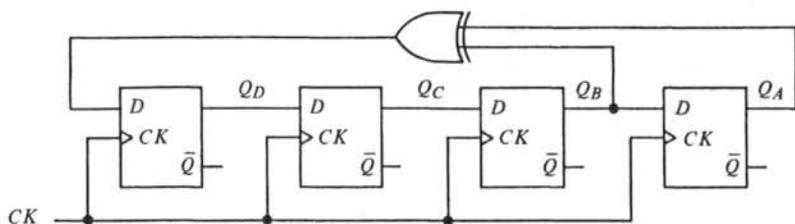


Fig. 5.39

## 5.3.10 Exercice

Analyser le logigramme de la figure 5.40 puis démontrer qu'il s'agit d'un compteur synchrone programmable dont la mesure  $p$  est variable; établir la correspondance entre les états d'entrée  $x_1, x_2$  et la valeur de  $p$ .

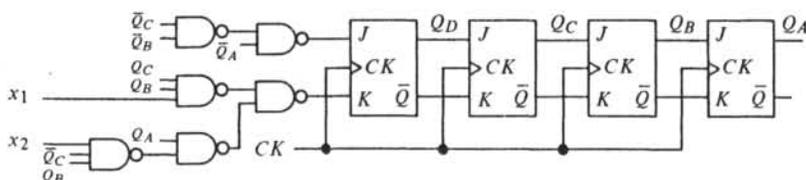


Fig. 5.40

## ■ 5.3.11 Exercice

Analyser le logigramme de la figure 5.41 puis vérifier la figure 5.42 qui donne pour chaque état d'entrée  $P_D, P_C, P_B, P_A$  la mesure  $p$  d'un compteur synchrone programmable. Quelles sont les valeurs de  $p$  pour les états  $P_D, P_C, P_B, P_A = 0000$  et  $1111$  ?

$P_D$	$P_C$	$P_B$	$P_A$	$p$
1	1	1	0	2
1	1	0	0	3
1	0	0	0	4
0	0	0	1	5
0	0	1	0	6
0	1	0	0	7
1	0	0	1	8
0	0	1	1	9
0	1	1	0	10
1	1	0	1	11
1	0	1	0	12
0	1	0	1	13
1	0	1	1	14
0	1	1	1	15

Fig. 5.42

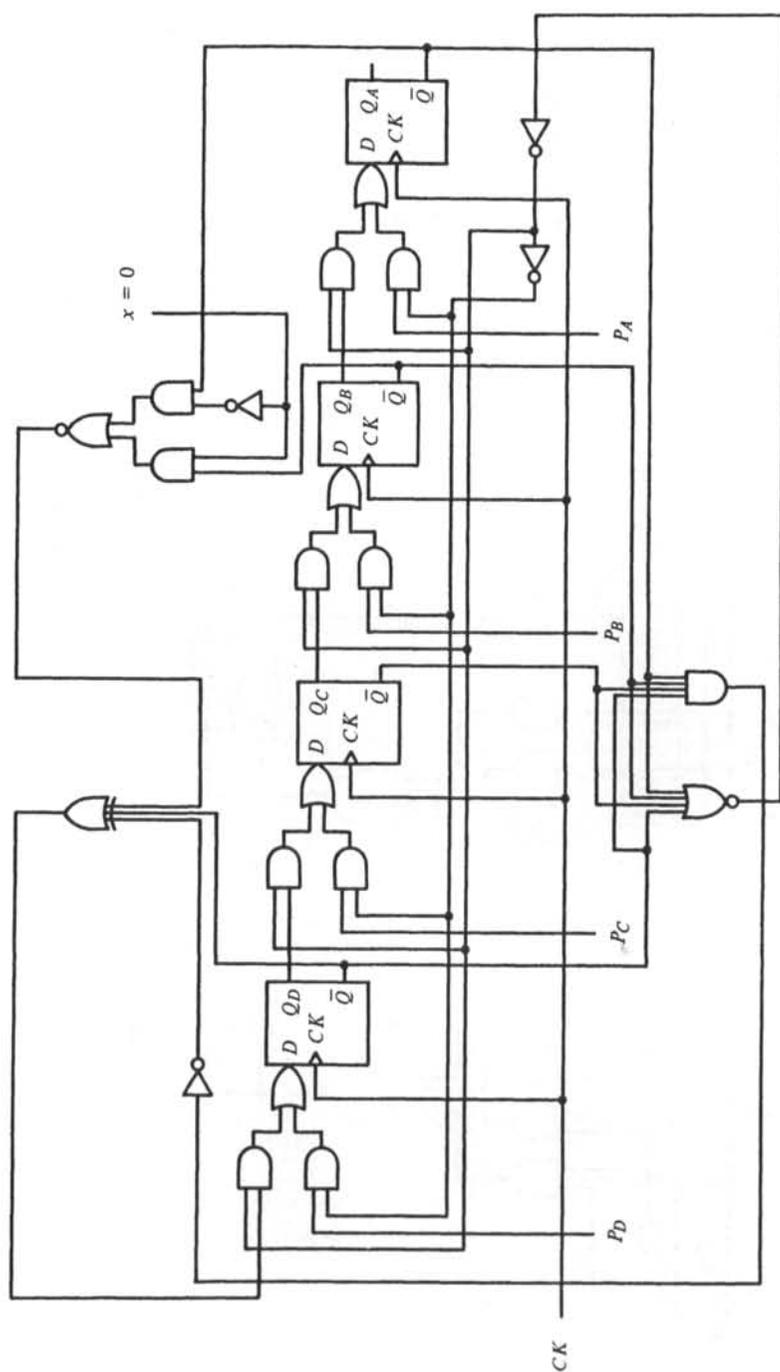


Fig. 5.41

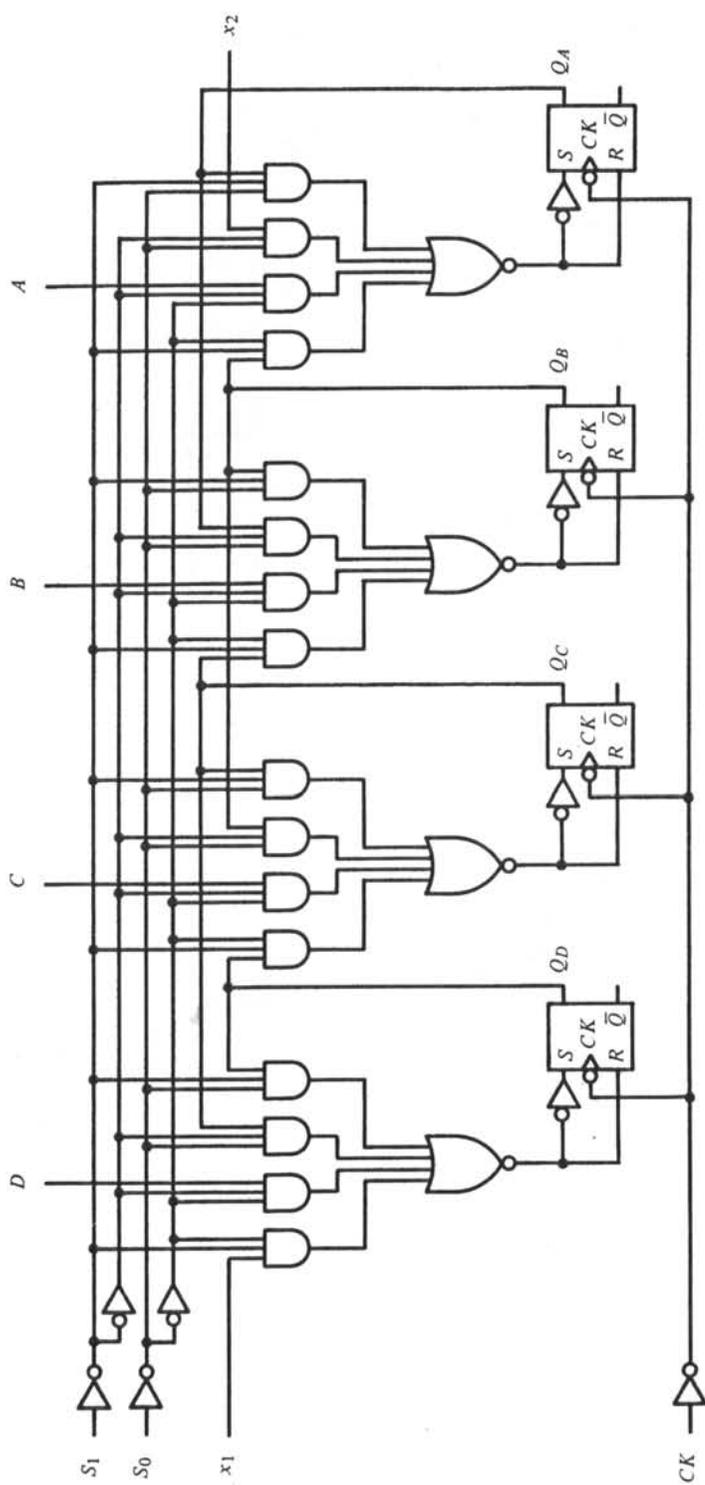


Fig. 5.43

### ■ 5.3.12 Exercice

On donne le logigramme de la figure 5.43. Déterminer les fonctions  $x_1$  et  $x_2$  afin d'obtenir le comportement suivant :

- pour  $S_0, S_1 = 00$  :  $Q_D^+, Q_C^+, Q_B^+, Q_A^+ = Q_D, Q_C, Q_B, Q_A$  (arrêt);
- pour  $S_0, S_1 = 10$  : comptage par 10, code quelconque;
- pour  $S_0, S_1 = 01$  : comptage par 10 avec la séquence inverse (décomptage);
- pour  $S_0, S_1 = 11$  :  $Q_D^+, Q_C^+, Q_B^+, Q_A^+ = D, C, B, A$  (saut dans un état présélectionné).

Simplifier les fonctions  $x_1$  et  $x_2$  puis analyser les états  $\emptyset$  de ce compteur réversible.

Réaliser ensuite un compteur synchrone programmable dont la mesure  $p$  soit variable ( $1 \leq p \leq 16$ ) et dépende de l'état d'entrée  $D, C, B, A$ . Déterminer et simplifier les fonctions  $x_1, x_2, S_0$  et  $S_1$ . Est-il possible de modifier un tel compteur pour qu'il soit réversible ?

## 5.4 RÉALISATION SÉQUENTIELLE DES SYSTÈMES COMBINATOIRES ITÉRATIFS

### 5.4.1 Description de logigrammes

On a défini dans la section 2.7 les systèmes combinatoires itératifs (§ 2.7.5) et décrit l'exemple d'un comparateur de deux nombres (§ 2.7.4). Le schéma de celui-ci (fig. 2.83) montre que les  $n + 1$  bits de chacun des nombres  $A$  et  $B$  sont présentés simultanément sur  $2(n + 1)$  entrées ( $A_0, B_0, A_1, B_1, \dots, A_n, B_n$ ) : on parle d'une *représentation spatiale* de l'information et le calcul de l'égalité (fonction  $E_n$ ) s'effectue de façon combinatoire.

En lieu et place de cette structure itérative (ou pseudo-parallèle) il est possible de réaliser le logigramme de la figure 5.44 dans lequel un seul comparateur des bits  $A_j$  et  $B_j$  calcule successivement les  $n + 1$  fonctions  $E_0, E_1, \dots, E_n$ ; chacun de ces calculs est effectué à un instant d'horloge différent : on obtient une *représentation temporelle* de l'information.

### 5.4.2 Définition

On appelle *structure série* d'un système combinatoire itératif (à  $n$  sous-systèmes) le système séquentiel synchrone réalisant la même fonction en  $n$  instants d'horloge, avec un seul sous-système.

### 5.4.3 Analyse

Dans le schéma de la figure 5.44 on distingue deux registres à décalage auxiliaires, chargés de mémoriser les nombres  $A$  ( $A_n, A_{n-1}, \dots, A_1, A_0$ ) et  $B$  ( $B_n, B_{n-1}, \dots, B_1, B_0$ ), et un système séquentiel synchrone; celui-ci résulte de l'assemblage d'un système combinatoire identique au comparateur de la figure 2.85 (avec  $j = n$ ) et d'une bascule  $D$  (produisant  $E_{j-1}$ ).

L'équation du système combinatoire se déduit de la relation (2.77) en remplaçant l'indice  $n$  par l'indice  $j$  (avec  $0 \leq j \leq n$ ) :

$$E_j = \bar{A}_j \bar{B}_j E_{j-1} + A_j B_j E_{j-1} \quad (5.20)$$

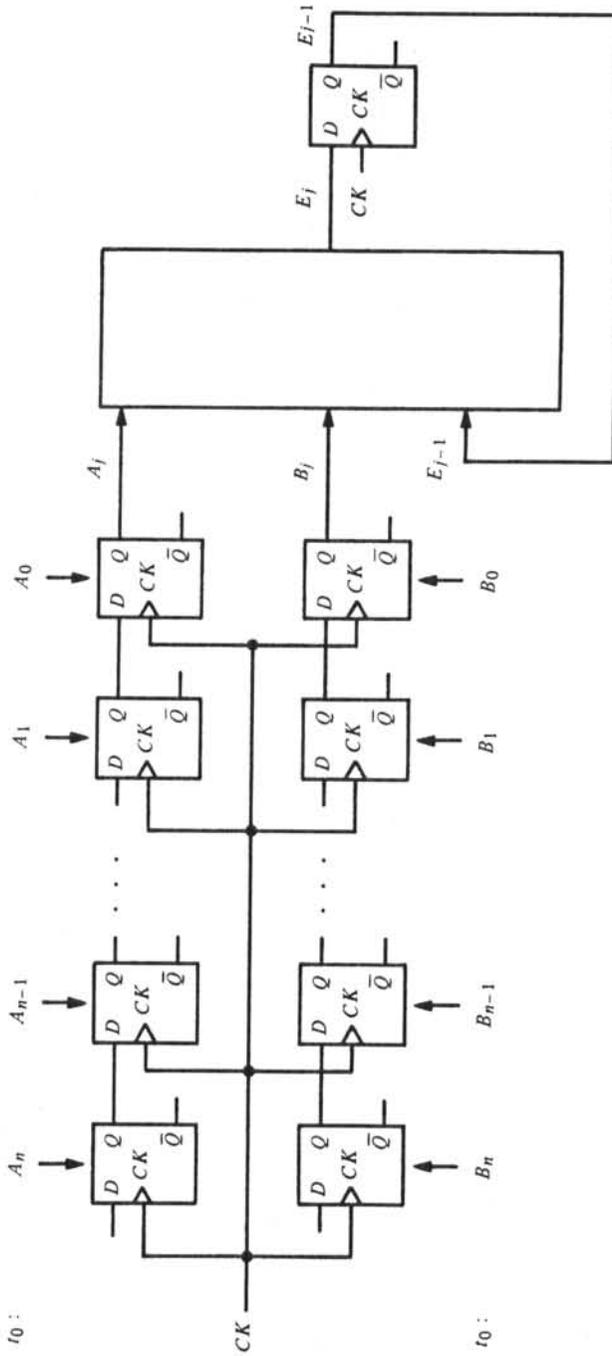


Fig. 5.44

L'équation caractéristique (3.52) de la bascule  $D$  donne l'état futur  $Q^+$ :

$$Q^+ = E_{j-1}^+ = E_j = \bar{A}_j \bar{B}_j E_{j-1} + A_j B_j E_{j-1} \quad (5.21)$$

dont la table et le graphe sont illustrés par les figures 5.45 et 5.46.

$E_{j-1}^+ = E_j$	$A_j B_j$			
	00	01	11	10
0	0	0	0	0
1	1	0	1	0

$E_{j-1}$

Fig. 5.45

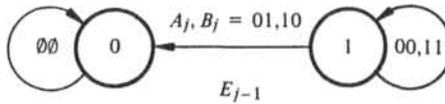


Fig. 5.46

#### 5.4.4 Description du comportement

Le comportement du logigramme de la figure 5.44 peut être décrit de la façon suivante :

- au premier instant d'horloge  $t_0$ , le système combinatoire calcule la fonction  $E_j = E_0$  (5.20) :

$$E_0 = \bar{A}_0 \bar{B}_0 E_{-1} + A_0 B_0 E_{-1} \quad (5.22)$$

Par analogie avec la structure pseudo-parallèle (§ 2.7.4 et fig. 2.83) on impose la valeur initiale de  $E_{j-1} = E_{-1}$  :

$$E_{-1} = 1 \quad (5.23)$$

- Après une montée et une descente de  $CK$  on atteint le deuxième instant d'horloge  $t_1$ ; les deux registres ont décalé les nombres  $A$  et  $B$  d'un bit vers la droite de même que la bascule  $D$  ( $E_{j-1} = E_0$ ). Le système combinatoire calcule alors :

$$E_1 = \bar{A}_1 \bar{B}_1 E_0 + A_1 B_1 E_0 \quad (5.24)$$

- A l'instant d'horloge  $t_n$  on obtient la valeur finale  $E_n$  (2.77) :

$$E_n = \bar{A}_n \bar{B}_n E_{n-1} + A_n B_n E_{n-1}$$

#### 5.4.5 Conclusion

La table d'états de la structure série (fig. 5.45) est formellement identique à la table de vérité de la structure pseudo-parallèle (fig. 2.84); la table d'états peut donc constituer un mode de représentation des systèmes combinatoires itératifs.

Tout système combinatoire qui admet une décomposition (au sens du paragraphe 2.7.5) peut être réalisé :

- soit par une structure pseudo-parallèle à  $n$  sous-systèmes et dont le délai est  $n \cdot \Delta t$  (où  $\Delta t$  est le délai d'un sous-système);
- soit par une structure série dont le coût est  $n$  fois plus faible (un seul sous-système) mais dont le délai est plus élevé ( $n \cdot T$  où  $T$  est la période du signal d'horloge).

Les structures pseudo-parallèle (fig. 2.83) et série (fig. 5.44) du comparateur sont unidimensionnelles [10] (pp. 529 - 534), [11] (pp. 434 - 438), [18] (pp. 88 - 97); l'extension à des structures bidimensionnelles ou multidimensionnelles est étudiée dans la théorie des transformations espace - temps [26], [28] (pp. 32 - 44), [31] (pp. 1 - 8, 359 - 453).

#### 5.4.6 Exercice

Déterminer la structure série d'un système réalisant l'addition arithmétique de deux nombres  $A (A_n, A_{n-1}, \dots, A_1, A_0)$  et  $B (B_n, B_{n-1}, \dots, B_1, B_0)$  puis appliquer les résultats au cas particulier où  $B = 1$ . Comparer le logigramme avec ceux des exercices des paragraphes 2.7.8 et 4.2.13.

#### 5.4.7 Exercice

Déterminer la structure série d'un système détectant si un nombre  $A (A_n, A_{n-1}, \dots, A_1, A_0)$  est plus grand qu'un nombre  $B (B_n, B_{n-1}, \dots, B_1, B_0)$  ( $A > B$ ), égal ( $A = B$ ) ou plus petit ( $A < B$ ). Deux variantes seront calculées selon que la comparaison commence par les bits de poids fort ( $A_n$  et  $B_n$ ) ou de poids faible ( $A_0$  et  $B_0$ ). Comparer les résultats avec ceux de l'exercice du paragraphe 2.7.9.

### 5.5 SYSTÈMES SÉQUENTIELS QUASI-SYNCHRONES

#### 5.5.1 Hypothèses

Par définition (§ 5.1.1) tout système séquentiel quasi-synchrone est l'assemblage d'un système combinatoire et d'une ou plusieurs bascules. On suppose que le premier est décrit par son modèle combinatoire (§ 1.1.10) et les secondes par leur modèle quasi-synchrone (§ 3.2.3).

De plus, on admet que tous les retards attachés aux bascules et contenus dans le modèle quasi-synchrone ont des valeurs très voisines (leurs différences sont négligeables); on suppose enfin qu'aucun retard ne sépare les variations des sorties  $Q$  et  $\bar{Q}$  d'une bascule.

Dans toute situation où ces hypothèses ne sont pas vérifiées on doit recourir au modèle asynchrone du chapitre 7.

#### 5.5.2 Analyse

Les logigrammes des figures 5.47 et 5.50 sont ceux de deux systèmes séquentiels quasi-synchrones; on observe dans le premier (fig. 5.47) deux signaux d'horloge distincts ( $CK_1 \neq CK_2$ ) et dans le second (fig. 5.50) une variable asynchrone non nulle ( $CLR_3 = Q_4$ ).

La méthode d'analyse sera celle des compteurs quasi-synchrones (§ 4.3.3) : après l'analyse séparée de chaque bascule (équation caractéristique), on entreprend l'analyse globale du système en tenant compte des divers signaux d'horloge et des variables asynchrones.

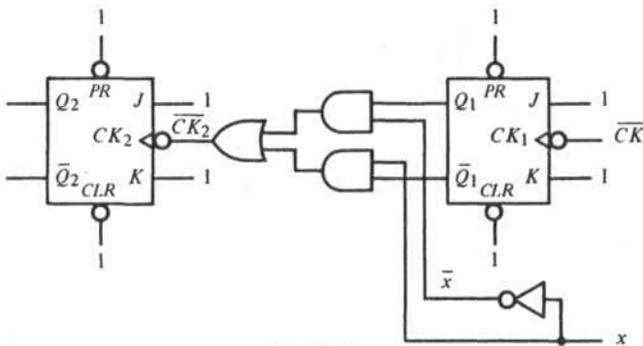


Fig. 5.47

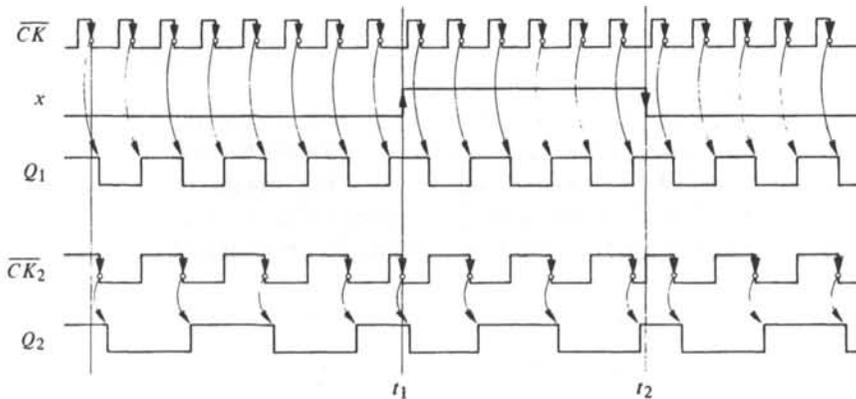


Fig. 5.48

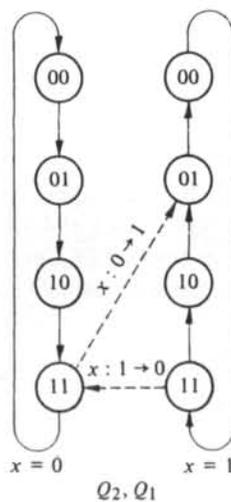


Fig. 5.49

### 5.5.3 Exemple : compteur réversible

Le logigramme de la figure 5.47 est celui d'un diviseur de fréquence par 4 (§ 4.1.1 et 4.1.4) modifié dont l'un des signaux d'horloge ( $CK_2$ ) dépend d'une variable d'entrée  $x$ . Les équations caractéristiques des bascules  $JK$  découlent de la relation (3.61) :

$$Q_1^+ = \bar{Q}_1 \quad ; \quad Q_2^+ = \bar{Q}_2 \quad (5.25)$$

tandis que les signaux d'horloge sont décrits par les expressions :

$$CK_1 = CK \quad ; \quad CK_2 = \overline{Q_1 \bar{x} + \bar{Q}_1 x} \quad (5.26)$$

Pour  $x = 0$ , on retrouve les équations (4.2) d'un diviseur de fréquence : la première partie du chronogramme de la figure 5.48 ( $x = 0$ ) est donc identique à la figure 4.3 pour les états  $Q_2, Q_1$ .

Pour  $x = 1$ , on constate que le signal d'horloge  $CK_2$  est inversé :

$$CK_2 = Q_1 \quad (5.27)$$

On peut alors compléter le chronogramme de la figure 5.48 où l'on représente deux changements successifs de la valeur de  $x$  (aux temps  $t_1$  et  $t_2$ ).

Dans la figure 5.49 on a tracé un premier graphe des états permanents valable pour  $x = 0$  : il représente un compteur par quatre dans le code binaire pur; un second graphe décrit le comportement du système pour  $x = 1$  : c'est toujours un compteur par quatre, dont le code est l'inverse du précédent. Deux passages direct et inverse du comptage ( $x = 0$ ) au décomptage ( $x = 1$ ) sont illustrés par les flèches en trait discontinu.

### 5.5.4 Conclusion

Dans certains systèmes séquentiels quasi-synchrones il existe plusieurs signaux d'horloge : il est par conséquent impossible d'utiliser les modes de représentation des systèmes séquentiels synchrones (table et graphe des états); le graphe de la figure 5.49 est incomplet : il ne représente pas tous les passages possibles du comptage au décomptage et vice versa. Le recours au modèle asynchrone du chapitre 7 est la seule possibilité d'effectuer une analyse systématique du logigramme de la figure 5.47.

### 5.5.5 Exemple : préleveur de période

Le logigramme de la figure 5.50 se décompose en un diviseur de fréquence par 4 (bascules  $Q_1$  et  $Q_2$  : voir les paragraphes 4.1.1 et 4.1.4) et un système séquentiel quasi-synchrone (bascules  $Q_3$  et  $Q_4$ ) à deux variables d'entrée  $x, Q_2$ . Les équations caractéristiques de ces deux dernières bascules découlent de la relation (3.56) :

$$Q_3^+ = J_3 \bar{Q}_3 + \bar{K}_3 Q_3 = 1 \quad (5.28)$$

$$Q_4^+ = J_4 \bar{Q}_4 + \bar{K}_4 Q_4 = Q_3 \quad (5.29)$$

tandis que leurs signaux d'horloge sont :

$$CK_3 = \bar{x} \quad ; \quad CK_4 = \bar{Q}_2 \quad (5.30)$$

Il existe enfin une variable asynchrone non nulle :

$$CLR_3 = Q_4 \quad (5.31)$$

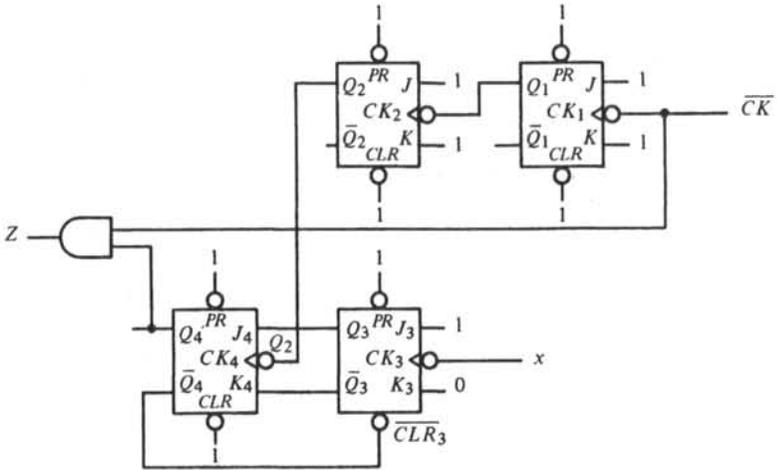


Fig. 5.50

Le chronogramme de la figure 5.51 est établi en admettant au départ que :

- le signal d'horloge  $\overline{CK}$  est périodique; la sortie  $Q_2$  produit une division par 4 de la fréquence (fig. 4.3);
- l'état  $Q_3, Q_4$  est égal à 00, tandis que la variable d'entrée  $x$  passe de 1 à 0 au temps  $t_0$ ;
- il existe une fonction de sortie  $Z = Q_4 \overline{CK}$ .

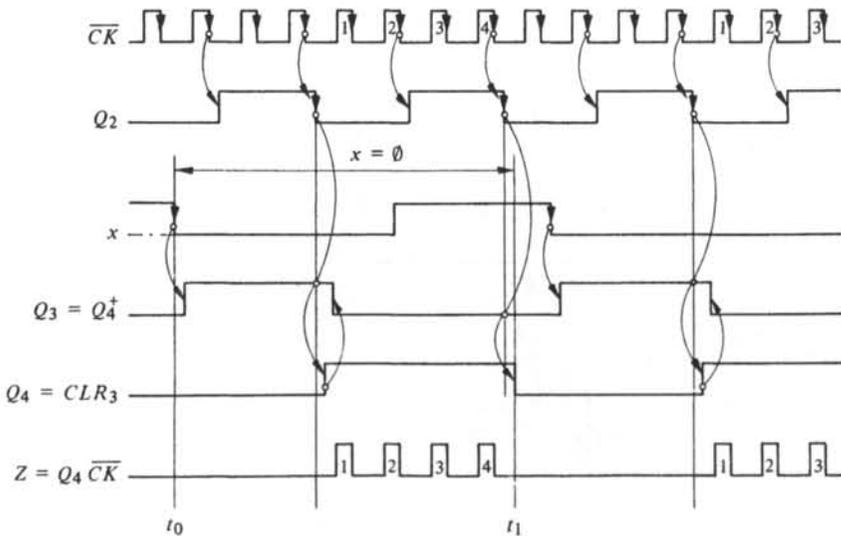


Fig. 5.51

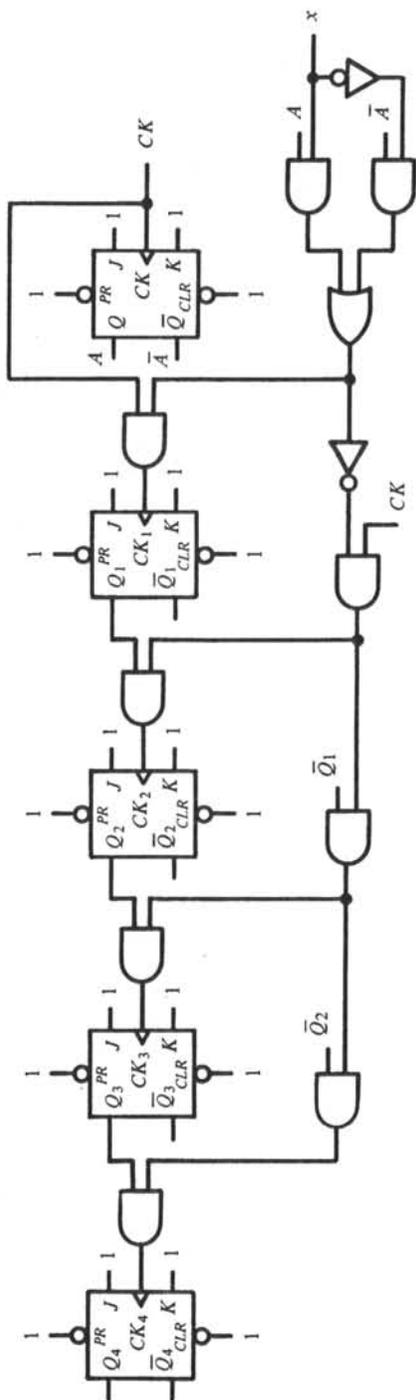


Fig. 5.52

Le système produit alors quatre impulsions (1, 2, 3, 4) sur la sortie  $Z$  pour toute descente de l'entrée  $x$  : c'est un *préleveur d'impulsions*; les bascules  $Q_3$  et  $Q_4$  constituent un système séquentiel appelé *préleveur de période*. On constate que dans l'intervalle de temps de  $t_0$  à  $t_1$  les variations de  $x$  sont sans effet ( $x = 0$ ); c'est seulement lorsque  $t > t_1$  qu'une descente de cette variable entraîne un nouvel envoi de quatre impulsions.

Toute description du logigramme de la figure 5.50 à l'aide des modes de représentation des systèmes séquentiels synchrones est impossible : on doit également recourir au modèle asynchrone du chapitre 7 pour une analyse systématique.

### 5.5.6 Synthèse

Étant donné la variété des logigrammes réalisant les systèmes séquentiels quasi-synchrones, il est difficile d'envisager une méthode de synthèse absolument générale et systématique; il existe cependant quelques tentatives dans cette direction [67] (pp. 141-175), [57] mais le plus souvent la synthèse reste empirique.

### 5.5.7 Exercice

Compléter le graphe du compteur réversible (fig. 5.49) en considérant les changements de  $x$  ( $0 \rightarrow 1$  et  $1 \rightarrow 0$ ) à partir de tous les états possibles  $Q_2, Q_1$ .

### 5.5.8 Exercice

A quelles conditions les bascules  $Q_3$  et  $Q_4$  du préleveur de période (fig. 5.50) valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ?

### 5.5.9 Exercice

Analyser les logigrammes des figures 5.52 à 5.54 et déterminer leur cahier des charges. On analysera également le logigramme de la figure 5.52 en admettant que les retards du modèle quasi-synchrone sont nuls.

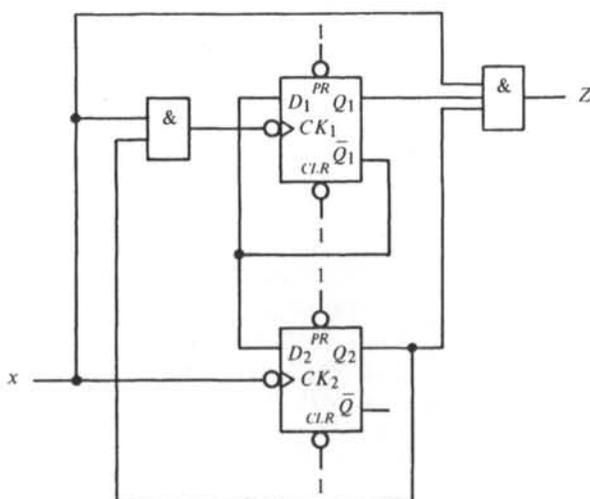


Fig. 5.53

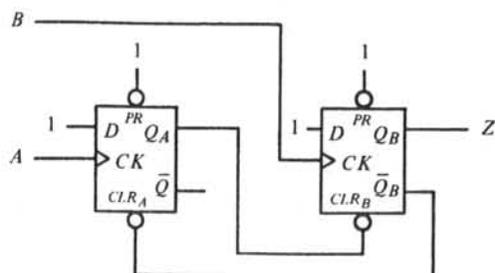


Fig. 5.54

## SYNTHÈSE DES SYSTÈMES SÉQUENTIELS SYNCHRONES

### 6.1 CONCEPTION DES SYSTÈMES À COMPORTEMENT ASYNCHRONE

#### 6.1.1 Introduction

L'objectif de la synthèse est la réalisation d'un logigramme à partir d'un cahier des charges imposé. On se limitera dans ce chapitre aux systèmes séquentiels synchrones (§ 5.1.1) dont la synthèse sera décomposée en deux opérations principales : la *conception*, qui est une représentation formelle du comportement (graphe des états, table d'états), et la *réalisation*, qui est la détermination d'un logigramme. Ces deux opérations seront illustrées par deux exemples caractéristiques : celui d'un discriminateur du sens de rotation (sect. 6.1 et 6.2) et celui d'un détecteur de séquence (sect. 6.3 et 6.4). Le cas particulier des réalisations avec registres à décalage fait l'objet de la section 6.5.

#### 6.1.2 Hypothèses

Comme pour l'analyse des systèmes séquentiels synchrones (§ 5.1.4) on admet dans ce chapitre que tout système combinatoire est décrit par son modèle combinatoire (§ 1.1.10) et que toute bascule bistable est décrite par son modèle synchrone (§ 3.2.10).

Si ces hypothèses ne sont pas vérifiées on doit recourir au modèle asynchrone du chapitre 7.

#### 6.1.3 Méthode

Dans ce chapitre on suivra généralement la méthode de synthèse suivante :

- à partir du cahier des charges imposé on détermine de façon empirique des modes de représentation formels : graphe des états et table d'états non assignés;
- une éventuelle manipulation des tables d'états nous permet de diminuer le nombre de leurs lignes : la conception est alors achevée.
- Le codage des tables d'états attribue à chaque ligne de celles-ci un état interne  $y_1, y_2, \dots, y_m$  (Fig. 5.1);
- avec la table des transitions des bascules utilisées on calcule les fonctions d'excitation (par exemple  $D_1, D_2, \dots, D_m$ ) en partant des états présents  $y_1, y_2, \dots, y_m$  et des états futurs  $y_1^+, y_2^+, \dots, y_m^+$ ;
- on simplifie les fonctions de sortie  $Z_1, Z_2, \dots, Z_r$  et les fonctions d'excitation ( $D_1, D_2, \dots, D_m$  par exemple), puis on les réalise avec un logigramme : la réalisation est terminée.

### 6.1.4 Exemple : discriminateur du sens de rotation

On veut déterminer le sens de rotation d'un arbre au moyen d'une variable logique  $Z$  qui prendra la valeur 0 si l'arbre tourne dans le sens des aiguilles d'une montre et la valeur 1 dans le sens opposé (fig. 6.1). Quatre secteurs isolés entre eux sont reliés alternativement aux constantes logiques 0 et 1, tandis que deux frotteurs (dont l'espacement est inférieur au plus petit des arcs des secteurs) sont les entrées  $x_1$  et  $x_2$  du discriminateur [67] (pp. 36 - 38, 51 - 55).

Cette description informelle, exprimée dans le langage naturel (§ 5.2.6), constitue le cahier des charges du système séquentiel.

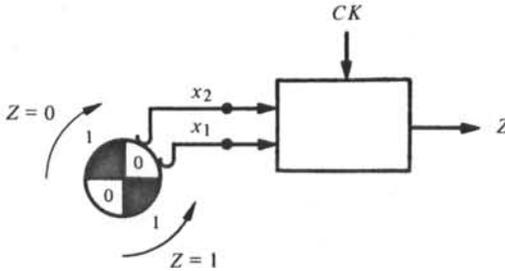


Fig. 6.1

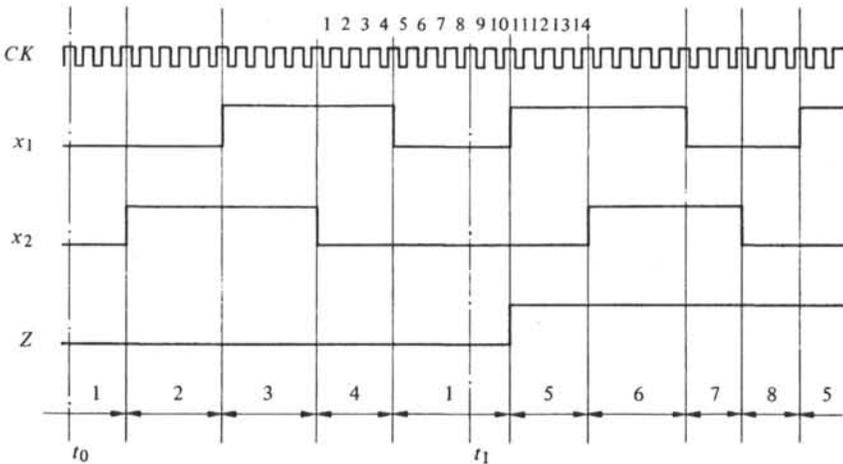


Fig. 6.2

### 6.1.5 Mode de représentation : chronogramme

Le cahier des charges définit une relation entre les états d'entrée  $x_1, x_2$  et l'état de sortie  $Z$ . Un chronogramme (fig. 6.2) peut décrire cette relation en fonction du temps; en supposant qu'au temps  $t_0$  l'arbre est dans la situation illustrée par la figure 6.1 ( $x_1, x_2 = 00, Z = 0$ ) et qu'il effectue un demi-tour dans le sens des aiguilles de la montre ( $Z = 0$ ) suivi dès l'instant  $t_1$  d'un demi-tour en sens opposé ( $Z = 1$ ), on parcourt huit étapes successives (numérotées de 1 à 8) ou *phases* décrites chacune par un état  $x_1, x_2, Z$ .

Dans le cas particulier, un état  $x_1, x_2, Z$  définit une seule phase; dans le cas général, un même état ( $x_1, x_2, \dots, x_n; Z_1, Z_2, \dots, Z_r$ ) peut définir plusieurs phases distinctes.

### 6.1.6 Définition : comportement asynchrone

On a déjà admis que la fréquence du signal d'horloge  $CK$  était plus élevée que celle des signaux d'entrée  $x_1$  et  $x_2$  (§ 5.2.9); on relève alors une caractéristique importante du discriminateur qui est bien visible dans la figure 6.2 : *l'état de sortie  $Z$  ne dépend pas de la durée des états d'entrée  $x_1, x_2$  mais seulement de leur ordre de succession*. On dira qu'un tel système possède un *comportement asynchrone*.

### 6.1.7 Mode de représentation : graphe des états

Il est possible de faire correspondre à chacune des huit phases 1, 2, ..., 8 du chronogramme (fig. 6.2) un état interne du système séquentiel et d'esquisser ainsi un graphe des états (fig. 6.3); celui-ci est établi en tenant compte du comportement asynchrone du discriminateur (§ 5.2.9) :

- chaque phase ( $x_1, x_2, Z$ ) détermine un état total stable (§ 5.2.7); le système reste dans le même état total pendant plusieurs périodes du signal d'horloge jusqu'au moment où une variation de l'état d'entrée  $x_1, x_2$  l'entraîne dans un autre état total stable;
- le passage d'un état total stable (par exemple l'état 1 avec  $x_1, x_2 = 00$ ) à un autre état total stable (l'état 2 avec  $x_1, x_2 = 01$ ) se fait par l'intermédiaire d'un état total instable (§ 5.2.8; dans cet exemple, il s'agit de l'état 1 avec  $x_1, x_2 = 01$ ) dont la durée est celle d'une période du signal d'horloge.

On complète ensuite le squelette de la figure 6.3 en constatant qu'il est possible d'inverser le sens de rotation dans chacun des états 2, 3 et 4 pour aboutir aux états 8, 7 et 6 avec  $Z = 1$ . Inversement, on peut regagner chacun des états 1, 4, 3 et 2 ( $Z = 0$ ) à la suite d'un changement du sens de rotation effectué dans les états 5, 6, 7 et 8. Le graphe final est celui de la figure 6.4.

La nécessité de compléter le graphe tel qu'il est issu du chronogramme montre que celui-ci (fig. 6.2) est un mode de représentation incomplet du cahier des charges.

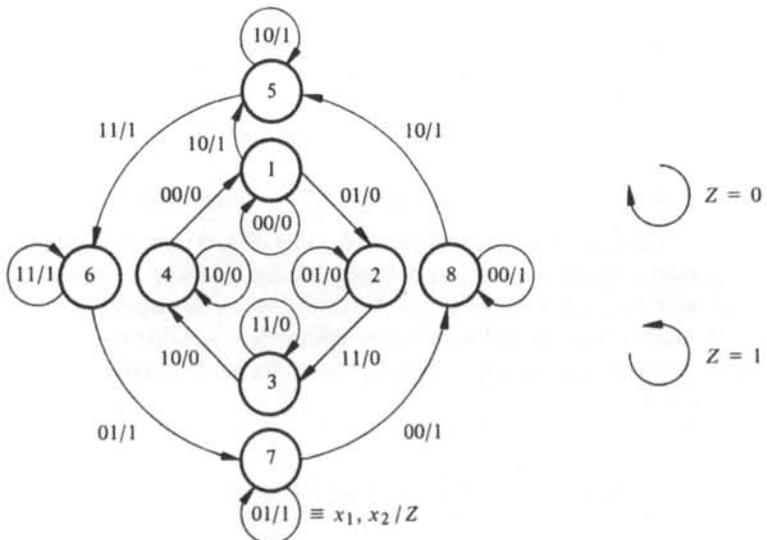


Fig. 6.3

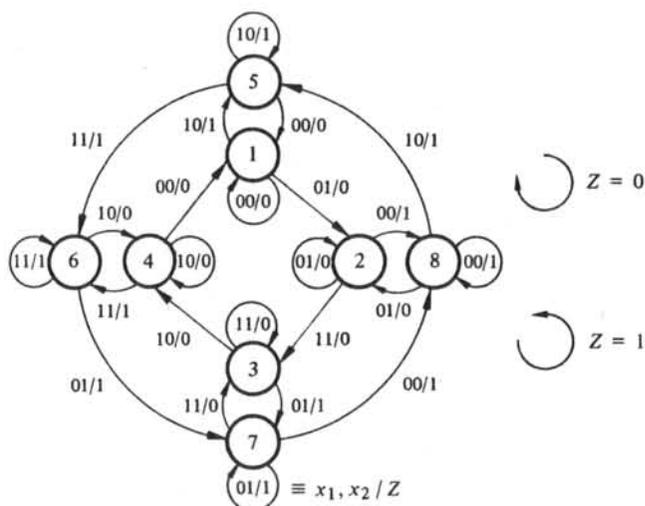


Fig. 6.4

### 6.1.8 Mode de représentation : table d'états

On peut déduire du graphe des états (fig. 6.4) une table d'états non assignée (§ 5.2.4) en procédant de la façon suivante (fig. 6.5) :

- chacun des quatre états d'entrée  $x_1, x_2$  définit une colonne;
- chacun des huit sommets du graphe définit une ligne; chaque ligne représente donc un état interne  $y_1, y_2, \dots, y_m$  dont l'ensemble sera symbolisé par  $Y$ ;
- chacune des 32 cases de la table est l'intersection d'une colonne avec une ligne et définit un état total  $(x_1, x_2; y_1, y_2, \dots, y_m)$ ;
- chaque flèche du graphe est caractérisée par un sommet de départ (c'est-à-dire un état interne  $y_1, y_2, \dots, y_m$ ) et par un état d'entrée  $x_1, x_2$ ; chaque flèche définit un état total, donc une case de la table;
- chaque flèche est également munie d'un sommet d'arrivée (c'est-à-dire un état futur  $y_1^+, y_2^+, \dots, y_m^+$  dont l'ensemble est symbolisé par  $Y^+$ ) et d'un état de sortie  $Z$ ; on reporte dans chaque case de la table la combinaison  $(y_1^+, y_2^+, \dots, y_m^+, Z)$ ;
- les états totaux stables (§ 5.2.7) sont encerclés.

La disposition particulière de la table d'états met en évidence l'ensemble des états totaux; il apparaît ainsi dans la figure 6.5 que huit d'entre eux ne sont pas définis (tirets). Il pourrait s'agir d'un oubli dans la formalisation du cahier des charges; mais dans l'exemple traité ces tirets traduisent un comportement irréalisable : les variations simultanées des entrées  $x_1$  et  $x_2$  sont rendues impossibles par le décalage géométrique des frotteurs (fig. 6.1).

### 6.1.9 Définitions : tables d'états complètement et incomplètement définies

Une table d'états est *complètement définie* lorsque les états futurs  $y_1^+, y_2^+, \dots, y_m^+$  et les états de sortie  $Z_1, Z_2, \dots, Z_r$  sont définis pour tout état total (fig. 5.15 par exemple).

Dans le cas contraire, on dit que la table d'états est *incomplètement définie* : la figure 6.5 illustre un tel exemple.

Soit une table d'états incomplètement définie  $T$ ; si une table d'états complètement définie  $T_1$  est identique à  $T$  pour tout état total où  $T$  est définie, alors on dit que  $T_1$  est une *solution* de  $T$ ; l'ensemble des solutions de  $T$  forme une *famille*.

$Y^*, Z$	$x_1, x_2$			
	00	01	11	10
1	(1,0)	2,0	-	5,1
2	8,1	(2,0)	3,0	-
3	-	7,1	(3,0)	4,0
4	1,0	-	6,1	(4,0)
5	1,0	-	6,1	(5,1)
6	-	7,1	(6,1)	4,0
7	8,1	(7,1)	3,0	-
8	(8,1)	2,0	-	5,1
$Y$				

Fig. 6.5

$Y^*, Z$	$x_1, x_2$			
	00	01	11	10
$a$	(a,0)	b,0	d,1	(a,1)
$b$	(b,1)	(b,0)	c,0	a,1
$c$	b,1	(c,1)	(c,0)	d,0
$d$	a,0	c,1	(d,1)	(d,0)
$Y$				

Fig. 6.6

1er graphe (Fig. 5.17)	2ème graphe (Fig. 6.4)
$a$	1,5
$b$	2,8
$c$	3,7
$d$	4,6

Fig. 6.7

### 6.1.10 Description

La comparaison des figures 5.17 et 6.4 met en évidence ceci :

- il s'agit de deux graphes des états;
- ces graphes décrivent le même cahier des charges : celui d'un discriminateur du sens de rotation;
- ils diffèrent par le nombre des états : quatre dans le premier (fig. 5.17) et huit dans le second (fig. 6.4).

La figure 6.7 propose une correspondance entre les états du premier graphe (fig. 5.17) et ceux du second (fig. 6.4).

### 6.1.11 Définitions : graphes et tables d'états équivalents

Lorsque deux graphes distincts décrivent le même cahier des charges, on dit qu'ils sont *équivalents*; les tables d'états qui leur sont associées sont aussi *équivalentes*.

Les graphes des figures 5.17 et 6.4 sont donc équivalents, de même que les tables des figures 5.15 et 6.5. La réalisation de ces tables nous conduirait à des logigrammes distincts satisfaisant tous deux le cahier des charges imposé : ces logigrammes seraient eux-mêmes équivalents.

Lorsque deux systèmes séquentiels synchrones distincts réalisent la même table d'états ou des tables d'états équivalentes, on dit que ces systèmes sont *fonctionnellement équivalents*.

### 6.1.12 Méthode : réduction des tables d'états

La recherche de tables équivalentes à une table d'états donnée, mais comportant un nombre de lignes plus faible, est appelée la *réduction*. Cette opération entraîne généralement une diminution du nombre des variables internes  $y_1, y_2, \dots, y_m$ , ce qui justifie son intérêt. Les méthodes générales de réduction des tables d'états complètement définies [68], [18] (pp. 127 - 135), [10] (pp. 256 - 275), [11] (pp. 268 - 278) et incomplètement définies [69], [18] (pp. 135 - 141), [10] (pp. 332 - 350, pp. 571 - 579), [11] (pp. 309 - 324) sortent du cadre de cet exposé et seront traitées dans le volume XI.

Néanmoins, un raisonnement intuitif nous permettra de réduire les huit lignes de la figure 6.5 aux quatre lignes de la figure 5.15.

### 6.1.13 Exemple

La figure 6.7 suggère qu'à toute paire d'états du second graphe correspond un seul état du premier. Illustrons cette réduction en partant de la paire 1,5 qui est extraite de la figure 6.5 :

$Y^+, Z$	$x_1 x_2$			
	00	01	11	10
1	1,0	2,0	-	5,1
5	1,0	-	6,1	5,1
$Y$				

On constate qu'une information identique à celle des deux lignes 1 et 5 peut être contenue dans une seule ligne 1,5 :

$Y^+, Z$	$x_1 x_2$			
	00	01	11	10
1,5	1,0	2,0	6,1	5,1
$Y$				

Remplaçons pour chaque état de  $Y$  et  $Y^+$  les chiffres (1, 2, 5, 6) par les lettres correspondantes (fig. 6.7) et nous obtenons enfin :

$Y^+, Z$	$x_1 x_2$			
	00	01	11	10
$a$	$a,0$	$b,0$	$d,1$	$a,1$
$Y$				

qui constitue la première ligne de la table réduite (fig. 5.15).

En répétant la même opération pour les paires d'états (2,8), (3,7) et (4,6), on obtient la table réduite de la figure 5.15; celle-ci est appelée dans la figure 6.6 avec les états totaux stables encerclés.

#### 6.1.14 Conclusion

Les limites de la conception sont de deux natures. Une limite qualitative est due à la détermination empirique du graphe des états (dans notre exemple la représentation du comportement dans un chronogramme constitue une façon de faire tout à fait spécifique au cas du discriminateur et nullement adaptée à l'ensemble des problèmes de synthèse). De véritables méthodes, telles que celles des langages réguliers, seront exposées dans le volume XI et sont également traitées dans les références [75] (pp. 1 - 63) et [32] (pp. 361 - 385). C'est surtout l'expérience, acquise notamment par des exercices, qui permettra au logicien d'acquérir un véritable savoir-faire.

Une limite quantitative est imposée par les modes de représentation choisis (tables d'états); pour traiter des problèmes dépassant  $2^5 = 32$  états il faut généralement :

- décomposer le système séquentiel donné en plusieurs sous-systèmes de dimension raisonnable [10] (pp. 289 - 300), [11] (pp. 418 - 425);
- utiliser d'autres modes de représentation tels que le langage de transfert de registres déjà cité au paragraphe 5.3.6;
- développer un logiciel pour microprocesseur : on abandonne dans ce cas la réalisation d'un système logique câblé au profit d'une programmation.

#### 6.1.15 Bibliographie

Les modes de représentation et méthode de ce chapitre reposent principalement sur les publications historiques de Huffman [65] et Mealy [66]. Des exercices et des applications se trouvent notamment dans les références [10] (pp. 241 - 249, pp. 329 - 332), [14] (pp. 305 - 322), [18] (pp. 121 - 125), [45] (pp. 154 - 163) et [76] (pp. 293 - 299).

#### 6.1.16 Exercice

Un système séquentiel comporte une sortie  $Z$  et deux entrées  $x_1, x_2$  dont les changements simultanés sont interdits (une période au moins du signal d'horloge sépare deux variations successives de  $x_1$  et  $x_2$ ); l'état de sortie  $Z$  est inchangé si  $x_2 = 0$  et prend la valeur de  $x_1$  si  $x_2 = 1$ . Déterminer le graphe et la table d'états, puis réduire éventuellement celle-ci.

#### 6.1.17 Exercice

Un système logique commande l'éclairage général  $Z$  de l'escalier d'une maison. Chacun des trois étages (rez-de-chaussée et deux niveaux) est muni d'un interrupteur ( $x_1, x_2, x_3$ ) de telle sorte qu'on puisse à tout étage et à tout moment soit éteindre l'éclairage  $Z$  ( $Z = 0$ ) si celui-ci est allumé, soit l'allumer s'il est éteint. Déterminer le graphe et la table d'états. Avec quel type particulier de système logique peut-on réaliser la table réduite ?

### 6.1.18 Exercice

Une porte automatique ne peut s'ouvrir que pour un sens de marche (de A à B), l'autre sens étant interdit. En A et B, c'est-à-dire de part et d'autre de la porte, sont disposées deux plates-formes mobiles qui actionnent à chaque passage deux contacts  $x_1$  et  $x_2$ ; ceux-ci constituent les entrées d'un système séquentiel dont la sortie est Z (ouverture de la porte). Déterminer le graphe et la table d'états, puis réduire celle-ci.

### 6.1.19 Exercice

Des barrières de passage à niveau sont situées sur une ligne de chemin de fer à voie unique, selon le schéma de la figure 6.8. On suppose que deux contacts  $x_1$  et  $x_2$  sont disposés sur la voie ferrée de part et d'autre des barrières et l'on admet que chaque véhicule du train actionne ces contacts : ceux-ci sont donc fermés tout au long du passage du train, quelle que soit sa longueur. Déterminer le graphe et la table d'états (éventuellement réduite) d'un système séquentiel dont la sortie Z commande l'abaissement des barrières.

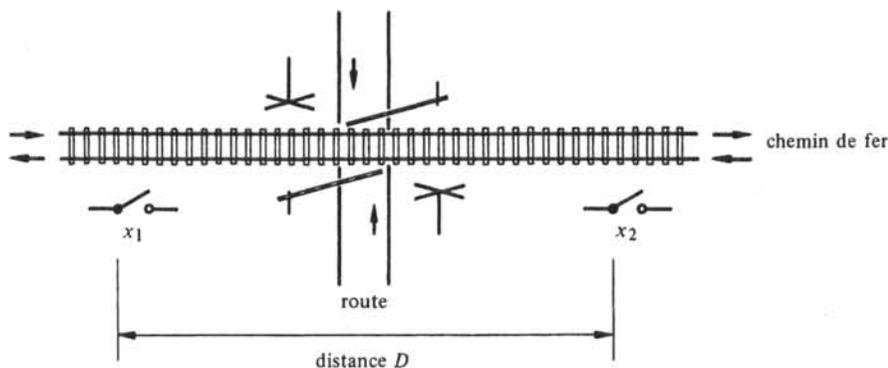


Fig. 6.8

### 6.1.20 Exercice

Une serrure électronique comporte deux entrées  $x_1, x_2$  et une sortie Z déclenchant l'ouverture. Z prend la valeur 1 à la fin de la séquence  $x_1, x_2 = 00 \rightarrow 01 \rightarrow 11$  et garde cette valeur pour toute variation de  $x_1, x_2$  jusqu'à l'apparition de la séquence  $x_1, x_2 = 11 \rightarrow 10 \rightarrow 00$ ; à la fin de celle-ci, Z reprend la valeur 0. Déterminer le graphe et la table d'états (éventuellement réduite) du système séquentiel en supposant que des variations simultanées de  $x_1$  et  $x_2$  sont irréalisables.

### ■ 6.1.21 Exercice

On veut réaliser une serrure électronique dont l'ouverture ( $Z_1 = 1$ ) est provoquée par une séquence particulière des états d'entrée ( $x_1, x_2$ ), toute autre séquence conduisant à l'enclenchement d'une alarme ( $Z_2 = 1$ ); il est également prévu une extinction de l'alarme ainsi qu'une fermeture du verrou, avec retour à l'état initial. La séquence d'ouverture comportera trois états d'entrée (sans compter l'état d'entrée initial), la séquence

d'extinction en comportera au moins deux et la séquence de fermeture trois. En interdisant les variations simultanées des entrées  $x_1$ ,  $x_2$  et en imposant huit états au maximum on demande le graphe et la table d'états du système séquentiel.

### 6.1.22 Exercice

On dispose de deux canaux d'information binaire appelés  $x_1$  et  $x_2$ ; en admettant que les variations simultanées de  $x_1$  et  $x_2$  sont irréalisables, on recherche un système séquentiel dont l'état de sortie  $Z_1$ ,  $Z_2$  est identique à l'avant-dernier état d'entrée  $x_1$ ,  $x_2$ : la figure 6.9 illustre un tel comportement. Déterminer le graphe et la table d'états (éventuellement réduite) de ce *décalage asynchrone*.

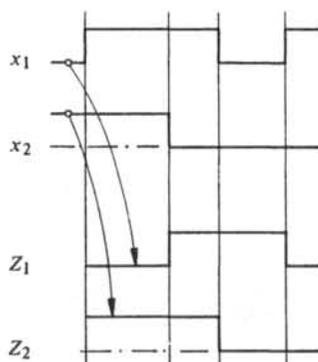


Fig. 6.9

### ■ 6.1.23 Exercice

Le préleveur de période est un système séquentiel quasi-synchrone comportant deux entrées  $x$ ,  $Q_2$  et une sortie  $Q_4$  (§ 5.5.5). A partir du cahier des charges de ce dispositif, on demande le graphe et la table d'états (éventuellement réduite) d'un système séquentiel synchrone équivalent.

### 6.1.24 Exercice

Un système séquentiel comporte trois entrées  $x_1$ ,  $x_2$ ,  $x_3$  et deux sorties  $Z_1$ ,  $Z_2$ . A partir du chronogramme de la figure 6.10 on demande de déterminer le graphe et la table d'états (éventuellement réduite).

### 6.1.25 Définitions : système séquentiel à impulsions

Soit un système séquentiel comportant trois variables d'entrée  $x_1$ ,  $x_2$ ,  $x_3$ ; on dit qu'il s'agit d'un *système séquentiel à impulsions* si :

- les seuls états d'entrée possibles sont  $x_1, x_2, x_3 = 000, 001, 010$  et  $100$ ;
- si les états d'entrée  $001, 010$  et  $100$  sont toujours séparés dans le temps par l'état de repos  $000$ .

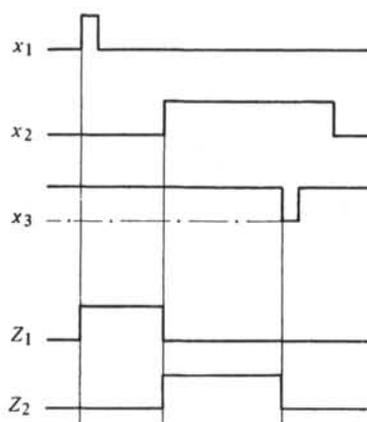


Fig. 6.10

On dit aussi que les états 001, 010 et 100 sont des *impulsions* de  $x_3$ ,  $x_2$  et  $x_1$ .

La figure 6.11 illustre le chronogramme d'un système séquentiel à impulsions; leur comportement particulier peut entraîner certaines simplifications dans la conception et dans la réalisation [10] (pp. 307 - 328), [11] (pp. 238 - 244, 247 - 253, 340 - 348).

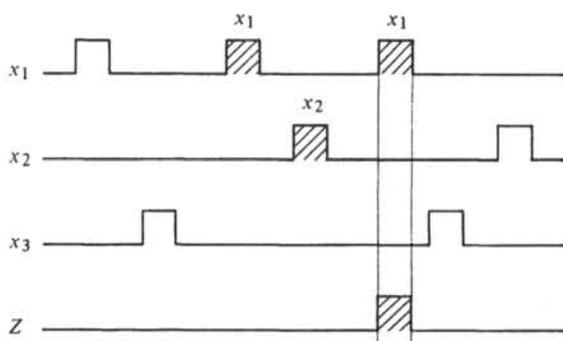


Fig. 6.11

### 6.1.26 Exercice : système séquentiel à impulsions

Établir le graphe et la table d'états d'un système à trois entrées  $x_1$ ,  $x_2$ ,  $x_3$  dont la sortie  $Z$  doit prendre la valeur 1 (conformément au chronogramme de la figure 6.11) seulement si les trois dernières impulsions ont été, dans l'ordre,  $x_1 \rightarrow x_1 \rightarrow x_1$  ou  $x_1 \rightarrow x_2 \rightarrow x_1$  (cette dernière séquence est celle de la figure 6.11).

### 6.1.27 Exercice : système séquentiel à impulsions

On veut réaliser la commande des barrières d'un passage à niveau selon le schéma de la figure 6.8. Contrairement au précédent exercice (§ 6.1.19) on suppose que seule une pédale montée sous la locomotive peut actionner les contacts  $x_1$  et  $x_2$ , de telle sorte qu'on satisfait la définition des systèmes à impulsions. Déterminer le graphe et la table d'états (éventuellement réduite) du dispositif dont  $Z$  est la sortie commandant l'abaissement des barrières.

## 6.2 RÉALISATION DES SYSTÈMES À COMPORTEMENT ASYNCHRONE

### 6.2.1 Introduction

La conception des systèmes séquentiels produit des tables dont les états sont désignés par des symboles quelconques (lettres de l'alphabet, chiffres décimaux, etc...). Nous savons déjà que des systèmes logiques particuliers, les bascules bistables, vont réaliser ces états. L'attribution d'une combinaison des valeurs de  $m$  variables internes  $y_1, y_2, \dots, y_m$  à chaque état d'une table est le *codage* ou l'*assignement* de cette table; cette opération traduit le passage d'un système discret (la table d'états non assignée) à un système logique (la table d'états assignée) et rend possible la réalisation d'un logigramme.

### 6.2.2 Définition : codage minimal

On cherche généralement à minimiser le nombre  $m$  des variables internes  $y_1, y_2, \dots, y_m$  dans le but de réduire à la fois le nombre total des variables du système combinatoire et celui des bascules bistables. Après une réduction de la table originale selon la méthode du paragraphe 6.1.12 on obtient le nombre minimal des états  $M$ ; le nombre minimal des variables internes  $m$  découle de la relation suivante, déjà rencontrée dans le calcul des compteurs (4.71) :

$$2^{m-1} < M \leq 2^m \quad (6.1)$$

Cette inéquation définit le *codage minimal* d'une table d'états.

### 6.2.3 Propriétés

Le nombre  $m$  étant déterminé, on dispose d'une liberté complète pour coder chacun des  $M$  états à l'aide de l'une des  $2^m$  combinaisons des valeurs  $y_1, y_2, \dots, y_m$ . Il existe  $K(M, m)$  façons de coder  $M$  états avec  $m$  variables internes [10] (pp. 275 - 277), [11] (pp. 294 - 296), [18] (pp. 141 - 142), [33] (pp. 263 - 268) :

$$K(M, m) = \frac{2^m!}{(2^m - M)!} \quad (6.2)$$

Cette formule sera justifiée par l'exemple du paragraphe 6.2.4. Dans le cas particulier où  $M = 2^m$  on obtient alors, en rappelant que  $0! = 1$  :

$$K(2^m, m) = 2^m! \quad (6.3)$$

Parmi les  $K(M, m)$  codages possibles il existe des familles de codes équivalents, qui ne se distinguent que par une simple permutation et/ou inversion des variables internes. Le nombre de ces familles  $F(M, m)$  est donc inférieur à celui de  $K(M, m)$  et limite ainsi le choix du code final. La référence [62] donne la formule :

$$F(M, m) = \frac{(2^m - 1)!}{(2^m - M)!m!} = \frac{K(M, m)}{2^m \cdot m!} \quad (6.4)$$

### 6.2.4 Application

Soit une table à cinq états; la relation (6.1) s'écrit :

$$2^{m-1} < 5 \leq 2^m \quad (6.5)$$

et a pour solution  $m = 3$ . Trois variables internes  $y_1, y_2$  et  $y_3$  définissent huit états distincts  $y_1, y_2, y_3$  de telle sorte que :

- la première ligne de la table peut être codée par l'un des huit états;
- la deuxième ligne par l'un des sept états restants, etc...;
- la cinquième (et dernière) ligne par l'un des quatre états restants.

Le nombre des codages distincts est donc :

$$K(M, m) = K(5, 3) = 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 = 6720 \quad (6.6)$$

qui peut s'écrire sous la forme :

$$K(M, m) = K(5, 3) = \frac{8!}{3!} = \frac{8!}{(2^3 - 5)!} = \frac{2^m!}{(2^m - M)!} \quad (6.7)$$

On a donc vérifié la validité de l'expression (6.2).

### 6.2.5 Exemple : discriminateur du sens de rotation

La table d'états réduite (fig. 6.6) comporte quatre états. Les relations (6.1), (6.3) et (6.4) définissent deux variables internes  $y_1, y_2$  qui produisent 24 codages réunis en trois familles; les trois colonnes de la figure 6.12 représentent ces familles et nous permettent de retrouver par permutation et/ou inversion des variables  $Y_1, Y_2$  n'importe lequel des 24 codages [10] (p. 276).

En choisissant la famille No 3 (fig. 6.12) et en effectuant les opérations

$$y_1 = \bar{Y}_2 \quad ; \quad y_2 = \bar{Y}_1 \quad (6.8)$$

on détermine un codage possible de la table d'états réduite (fig. 6.13). Chaque état présent ( $Y$ ) ou futur ( $Y^+$ ) de celle-ci (fig. 6.6) est alors remplacé par la combinaison des valeurs  $y_1, y_2$  de telle sorte qu'on obtient une table d'états assignée (fig. 6.14). L'état de sortie  $Z$  reste inchangé.

Etats $Y$	Famille No 1 $Y_1 Y_2$	Famille No 2 $Y_1 Y_2$	Famille No 3 $Y_1 Y_2$	Codage choisi $y_1 y_2$
$a$	0 0	0 0	0 0	1 1
$b$	0 1	1 1	1 0	1 0
$c$	1 1	0 1	0 1	0 1
$d$	1 0	1 0	1 1	0 0

Fig. 6.12

Fig. 6.13

$y_1^+ y_2^+, Z$	$x_1 x_2$			
	00	01	11	10
$a \equiv 11$	(11,0)	10,0	00,1	(11,1)
$b \equiv 10$	(10,1)	(10,0)	01,0	11,1
$c \equiv 01$	10,1	(01,1)	(01,0)	00,0
$d \equiv 00$	11,0	01,1	(00,1)	(00,0)

$Y \equiv y_1 y_2$

Fig. 6.14

$y_1 y_2$	$x_1 x_2$			
	00	01	11	10
$d \equiv 00$	11,0	01,1	00,1	00,0
$c \equiv 01$	10,1	01,1	01,0	00,0
$a \equiv 11$	11,0	10,0	00,1	11,1
$b \equiv 10$	10,1	10,0	01,0	11,1

$D_1, D_2; Z$

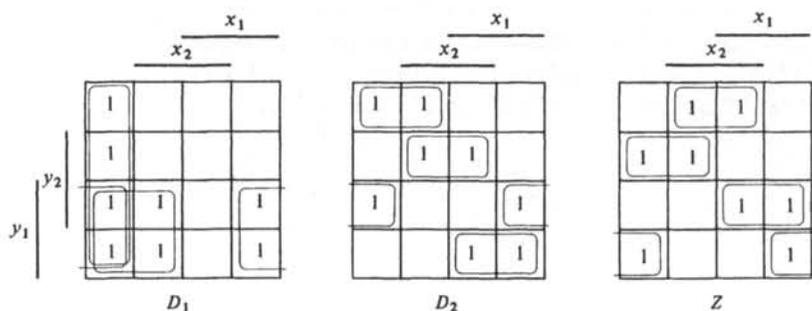


Fig. 6.15

### 6.2.6 Réalisation de logigrammes

Le logigramme doit comporter deux bascules bistables (dont les sorties sont les variables internes  $y_1, y_2$ ) et un système combinatoire réalisant la fonction de sortie  $Z$  et les fonctions d'excitation. Le calcul de celles-ci est semblable à celui des compteurs synchrones et peut être tabulaire (§ 4.2.8) [10] (pp. 277 - 289, pp. 350 - 355), [14] (pp. 311 - 320), [18] (pp. 142 - 150), [45] (pp. 165 - 189), [76] (pp. 299 - 310), algébrique (§ 4.2.9) ou, si l'on dispose de bascules  $D$ , abrégé (§ 4.2.11 et § 5.1.7). Illustrons ce dernier cas en rappelant que l'équation d'entrée (3.63) de la bascule  $D$  nous permet de poser :

$$D_1 = Q_1^+ \equiv y_1^+ ; D_2 = Q_2^+ \equiv y_2^+ \quad (6.9)$$

En permutant les lignes de la table d'états assignée (fig. 6.14) on trouve une table de Karnaugh principale (fig. 6.15) dont il découle trois tables secondaires. Les fonctions

d'excitation  $D_1$  et  $D_2$  peuvent s'exprimer sous la forme suivante :

$$D_1 = \bar{x}_1 \bar{x}_2 + \bar{x}_1 y_1 + \bar{x}_2 y_1 = MAJ(\bar{x}_1, \bar{x}_2, y_1) \quad (6.10)$$

qu'on peut aussi écrire sous la forme canonique décimale :

$$D_1(x_1, x_2, y_1) = \Sigma 0, 1, 3, 5 \quad (6.13)$$

$$\begin{aligned} D_2(x_1, x_2, y_1, y_2) &= \Sigma 0, 3, 4, 5, 10, 11, 13, 14 \\ &= \bar{x}_1 \bar{y}_1 \bar{y}_2 + x_1 y_1 \bar{y}_2 + x_2 \bar{y}_1 y_2 + \bar{x}_2 y_1 y_2 \\ &= \bar{y}_2 (\bar{x}_1 \oplus y_1) + y_2 (x_2 \oplus y_1) \end{aligned} \quad (6.11)$$

La détermination de la fonction de sortie est un problème purement combinatoire; la table de Karnaugh secondaire de la figure 6.15 nous permet de simplifier  $Z$  et d'écrire :

$$\begin{aligned} Z(x_1, x_2, y_1, y_2) &= \Sigma 1, 2, 4, 5, 10, 11, 12, 15 \\ &= x_1 y_1 y_2 + \bar{x}_1 \bar{y}_1 y_2 + x_2 \bar{y}_1 \bar{y}_2 + \bar{x}_2 y_1 \bar{y}_2 \\ &= y_2 (\bar{x}_1 \oplus y_1) + \bar{y}_2 (x_2 \oplus y_1) \end{aligned} \quad (6.12)$$

Une réalisation possible des fonctions d'excitation  $D_1$  (6.10),  $D_2$  (6.11) et de la fonction de sortie  $Z$  (6.12) est illustrée par le logigramme NAND, OU-exclusif de la figure 6.16, tandis que les multiplexeurs de la figure 6.17 (§ 1.5.9 et § 2.1.13) réalisent les formes canoniques de  $D_1$  (6.13),  $D_2$  (6.11) et  $Z$  (6.12).

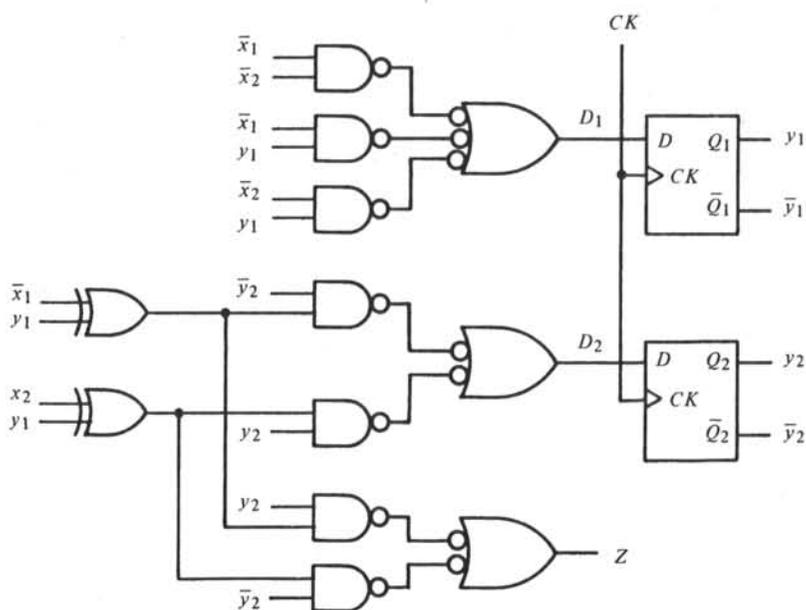


Fig. 6.16

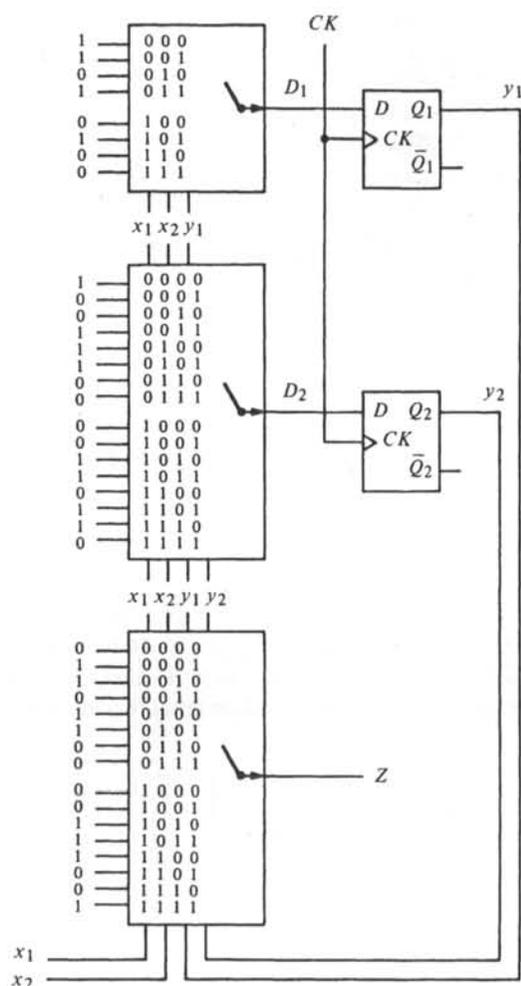


Fig. 6.17

### 6.2.7 Commentaire

La table d'états de la figure 5.15 illustre un autre codage du discriminateur, identique à celui de la famille No 1 de la figure 6.12 si l'on pose :

$$y_1 = Y_1 ; y_2 = Y_2 \quad (6.14)$$

Par définition (§ 6.2.3 et § 6.2.5) ce codage n'est pas équivalent au précédent (fig. 6.14) et décrit une réalisation dont les équations (5.5), (5.6) et (5.4) sont rappelées ci-dessous :

$$D_1 = x_1 x_2 + x_1 y_1 + x_2 y_1$$

$$D_2 = \bar{x}_1 x_2 + \bar{x}_1 y_2 + x_2 y_2$$

$$Z = \bar{x}_1 x_2 y_1 + x_1 \bar{x}_2 \bar{y}_1 + \bar{x}_1 \bar{x}_2 y_2 + x_1 x_2 \bar{y}_2$$

La comparaison de ces fonctions avec celles des expressions (6.10), (6.11) et (6.12) montre que les polynômes  $D_1$  et  $Z$  ont le même nombre de lettres (§ 1.4.5) tandis que le polynôme  $D_2$  a douze lettres dans le premier cas (6.11) et six dans le second (5.6).

Deux codages non équivalents ont donc donné deux réalisations du même discriminateur, d'une complexité différente. Il existe des méthodes pour choisir un bon codage minimal; mais le nombre de ceux-ci et la variété des réalisations (dépendant du type des bascules et des opérateurs combinatoires) limitent l'intérêt de ces méthodes qui seront traitées dans le volume XI ainsi que dans les références [10] (pp. 275 - 300), [11] (pp. 294 - 309), [18] (pp. 141 - 167), [70] (pp. 128 - 169) et [75] (pp. 155 - 210).

### 6.2.8 Exercice

A quelles conditions les bascules  $D$  du discriminateur du sens de rotation (fig. 6.16) valident-elles le modèle quasi-synchrone du paragraphe 3.2.3 ?

### 6.2.9 Exercice

Effectuer un codage minimal de la table d'états non réduite du discriminateur (fig. 6.5) avec le code de la figure 6.18, puis calculer et simplifier les fonctions d'excitation (pour des bascules  $D$ ) ainsi que la fonction de sortie  $Z$ . Comparer les résultats à ceux du paragraphe 6.2.7. Réaliser un logigramme, puis l'analyser pour déterminer les états futurs et de sortie qui ne sont pas définis dans la table d'états de la figure 6.5.

$Y$	$y_1$	$y_2$	$y_3$
1	0	0	0
2	0	0	1
3	0	1	1
4	0	1	0
5	1	0	0
6	1	0	1
7	1	1	1
8	1	1	0

Fig. 6.18

### 6.2.10 Exercice

Réaliser la table d'états assignée du discriminateur (fig. 6.14) avec des bascules  $JK$  : le calcul sera tabulaire (§ 4.2.8) ou algébrique (§ 4.2.9). Simplifier les fonctions d'excitation, tracer un logigramme et le comparer à celui de la figure 6.16.

### 6.2.11 Exercice

Effectuer un codage minimal des tables d'états réduites des exercices précédents : système de mémorisation (§ 6.1.16), porte automatique (§ 6.1.18), passage à niveau (§ 6.1.19), serrure électronique (§ 6.1.21), délai asynchrone (§ 6.1.22), préleveur de

période (§ 6.1.23) et passage à niveau (§ 6.1.27). Réaliser dans chaque cas deux logigrammes avec des bascules  $D$  et  $JK$ , puis analyser ceux-ci pour déterminer les états futurs et/ou les états de sortie qui ne sont pas définis dans la table d'états assignée.

### 6.2.12 Définition : codage 1 parmi $M$

Si l'on code une table de  $M$  états avec  $M$  variables internes de telle sorte qu'une seule de ces variables vaut 1 dans chaque état  $y_1, y_2, \dots, y_M$  on obtient un *codage 1 parmi  $M$* .

La figure 6.19 illustre un codage 1 parmi 4 de la table d'états du discriminateur (fig. 6.6); ce codage définit quatre variables internes  $y_d, y_c, y_b, y_a$  : il n'est donc pas minimal.

### 6.2.13 Propriétés

Tous les codages 1 parmi  $M$  sont équivalents, c'est-à-dire ne se distinguent que par une simple permutation des variables internes (§ 6.2.3). Pour une table de  $M$  états, il n'existe donc qu'une seule famille de codes et la valeur de  $F(M, m)$  (6.4) est égale à un.

Les  $M$  variables internes définissent au total  $2^M$  états dont  $M$  sont utilisés; il existe donc  $2^M - M$  états  $\emptyset$ . Dans le cas du discriminateur, les quatre variables internes produisent  $2^4 - 4 = 12$  états  $\emptyset$ .

$Y^+, Z$	$x_1 x_2$			
	00	01	11	10
$a \equiv 0001$	(a,0)	b,0	d,1	(a,1)
$b \equiv 0010$	(b,1)	(b,0)	c,0	a,1
$c \equiv 0100$	b,1	(c,1)	(c,0)	d,0
$d \equiv 1000$	a,0	c,1	(d,1)	(d,0)

$Y \equiv y_d y_c y_b y_a$

Fig. 6.19

$y_d^+ y_c^+ y_b^+ y_a^+, Z$	$x_1 x_2$			
	00	01	11	10
0 0 0 1	(0001,0)	0010,0	1000,1	(0001,1)
0 0 1 0	(0010,1)	(0010,0)	0100,0	0001,1
0 1 0 0	0010,1	(0100,1)	(0100,0)	1000,0
1 0 0 0	0001,0	0100,1	(1000,1)	(1000,0)

$y_d y_c y_b y_a$

Fig. 6.20

$y_d y_c$ / $y_b y_a$		$x_1 x_2$				
		00	01	11	10	
$a \equiv$	0 0 0 0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$y_a$
	0 0 0 1	0001,0	0010,0	1000,1	0001,1	
	0 0 1 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
$b \equiv$	0 0 1 0	0010,1	0010,0	0100,0	0001,1	$y_a$
	0 1 0 0	0010,1	0100,1	0100,0	1000,0	
$c \equiv$	0 1 0 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$y_a$
	0 1 1 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	0 1 1 0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	1 1 0 0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
$d \equiv$	1 1 0 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$y_a$
	1 1 1 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	1 1 1 0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	1 0 0 0	0001,0	0100,1	1000,1	1000,0	
	1 0 0 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$y_a$
	1 0 1 1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	1 0 1 0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	

$D_d, D_c, D_b, D_a; Z$

Fig. 6.21

### 6.2.14 Exemple : discriminateur du sens de rotation

Les figures 6.19 et 6.20 présentent la table d'états du discriminateur avant et après le codage 1 parmi 4, tandis que la figure 6.21 met en évidence les seize états internes dont douze sont inutilisés (états  $\emptyset$ ).

### 6.2.15 Réalisation avec multiplexeurs et démultiplexeurs

En admettant l'emploi de bascules  $D$  (3.63) on calcule les fonctions d'excitation par la méthode abrégée (§ 4.2.11, 5.1.7 et 6.2.6); celles-ci, ainsi que la fonction de sortie  $Z$ , peuvent être directement simplifiées dans la table de Karnaugh constituée par la figure 6.21 (où seule la simplification de  $D_d$  a été détaillée) :

$$y_d^+ = D_d = x_1 x_2 (y_d + y_a) + x_1 \bar{x}_2 (y_d + y_c) \quad (6.15)$$

$$= x_1 x_2 \cdot y_d + x_1 x_2 \cdot y_a + x_1 \bar{x}_2 \cdot y_d + x_1 \bar{x}_2 \cdot y_c \quad (6.20)$$

$$y_c^+ = D_c = \bar{x}_1 x_2 (y_d + y_c) + x_1 x_2 (y_c + y_b) \quad (6.16)$$

$$= \bar{x}_1 x_2 \cdot y_c + \bar{x}_1 x_2 \cdot y_d + x_1 x_2 \cdot y_b + x_1 x_2 \cdot y_c \quad (6.21)$$

$$y_b^+ = D_b = \bar{x}_1 \bar{x}_2 (y_c + y_b) + \bar{x}_1 x_2 (y_b + y_a) \quad (6.17)$$

$$= \bar{x}_1 \bar{x}_2 \cdot y_b + \bar{x}_1 \bar{x}_2 \cdot y_c + \bar{x}_1 x_2 \cdot y_a + \bar{x}_1 x_2 \cdot y_b \quad (6.22)$$

$$y_a^+ = D_a = \bar{x}_1 \bar{x}_2 (y_d + y_a) + x_1 \bar{x}_2 (y_b + y_a) \quad (6.18)$$

$$= \bar{x}_1 \bar{x}_2 \cdot y_a + \bar{x}_1 \bar{x}_2 \cdot y_d + x_1 \bar{x}_2 \cdot y_b + x_1 \bar{x}_2 \cdot y_a \quad (6.23)$$

$$Z = \bar{x}_1 \bar{x}_2 (y_c + y_b) + \bar{x}_1 x_2 (y_d + y_c) + x_1 x_2 (y_d + y_a) + x_1 \bar{x}_2 (y_b + y_a) \quad (6.19)$$

$$= \bar{x}_1 \bar{x}_2 \cdot y_c + \bar{x}_1 \bar{x}_2 \cdot y_b + \bar{x}_1 x_2 \cdot y_d + \bar{x}_1 x_2 \cdot y_c + x_1 x_2 \cdot y_d + x_1 x_2 \cdot y_a + x_1 \bar{x}_2 \cdot y_b + x_1 \bar{x}_2 \cdot y_a \quad (6.24)$$

Comme l'illustre l'exemple de  $D_d$  (fig. 6.21) on a volontairement omis les simplifications par rapport aux variables d'entrée  $x_1, x_2$ , ce qui facilitera la réalisation proposée.

La première expression de  $D_d$  (6.15) est identique à celle de la fonction universelle de deux variables  $Z(x_1, x_2; K_0, K_1, K_2, K_3)$  (1.42) dont les paramètres seraient :

$$K_0 = 0 ; K_1 = 0 ; K_2 = y_d + y_c ; K_3 = y_d + y_a \quad (6.25)$$

Un multiplexeur à deux variables (§ 1.5.9) peut donc réaliser  $D_d$  (fig. 6.22). Les expressions algébriques de  $D_c, D_b, D_a$  et  $Z$  (6.16)... (6.19) peuvent également être réalisées par des multiplexeurs conformément au logigramme de la figure 6.22.

La seconde expression de  $D_d$  (6.20) est un polynôme dont chaque monôme est le produit d'un minterme des variables d'entrée  $x_1, x_2$  avec une variable interne ( $y_d, y_c, y_b$  ou  $y_a$ ). Les quatre mintermes de  $x_1$  et  $x_2$  peuvent être réalisés par un démultiplexeur à deux variables (§ 1.5.3), les monômes de  $D_d$  par la mise en série d'une bascule bistable ( $y_d, y_c, y_b$  ou  $y_a$ ) et d'un démultiplexeur. Le logigramme de la figure 6.23 illustre cette réalisation de  $D_d$ , ainsi que celle des fonctions  $D_c, D_b, D_a$  et  $Z$  tirées des expressions (6.21) à (6.24).

#### □ 6.2.16 Commentaire

Dans la première réalisation (fig. 6.22) on dénombre autant de multiplexeurs que de fonctions d'excitation et de sortie, soit  $M + r$ ; chaque multiplexeur peut être précédé d'un certain nombre de portes OU : le nombre maximal de celles-ci s'élève à  $(M + r) \cdot 2^n$  (si  $n$  est le nombre des variables d'entrée), tandis que le nombre maximal des entrées de chacune de ces portes est égal à  $M$ .

Dans la seconde réalisation (fig. 6.23) on compte autant de démultiplexeurs que de variables internes, soit  $M$ ; le nombre maximal des portes OU est égal à  $M + r$ , tandis que le nombre maximal des entrées d'une telle porte s'élève à  $M \cdot 2^n$ .

#### 6.2.17 Définitions : systèmes séquentiels réparti et concentré

Les réalisations d'un codage 1 parmi  $M$  avec multiplexeurs (fig. 6.22) et démultiplexeurs (fig. 6.23) sont des cas très particuliers du schéma définissant les systèmes séquentiels synchrones (fig. 5.1 avec  $CK = CK_1 = CK_2 = \dots = CK_m$  et  $PR = CLR = 0$ ). Les premières font apparaître autant de systèmes combinatoires que de bascules (et de

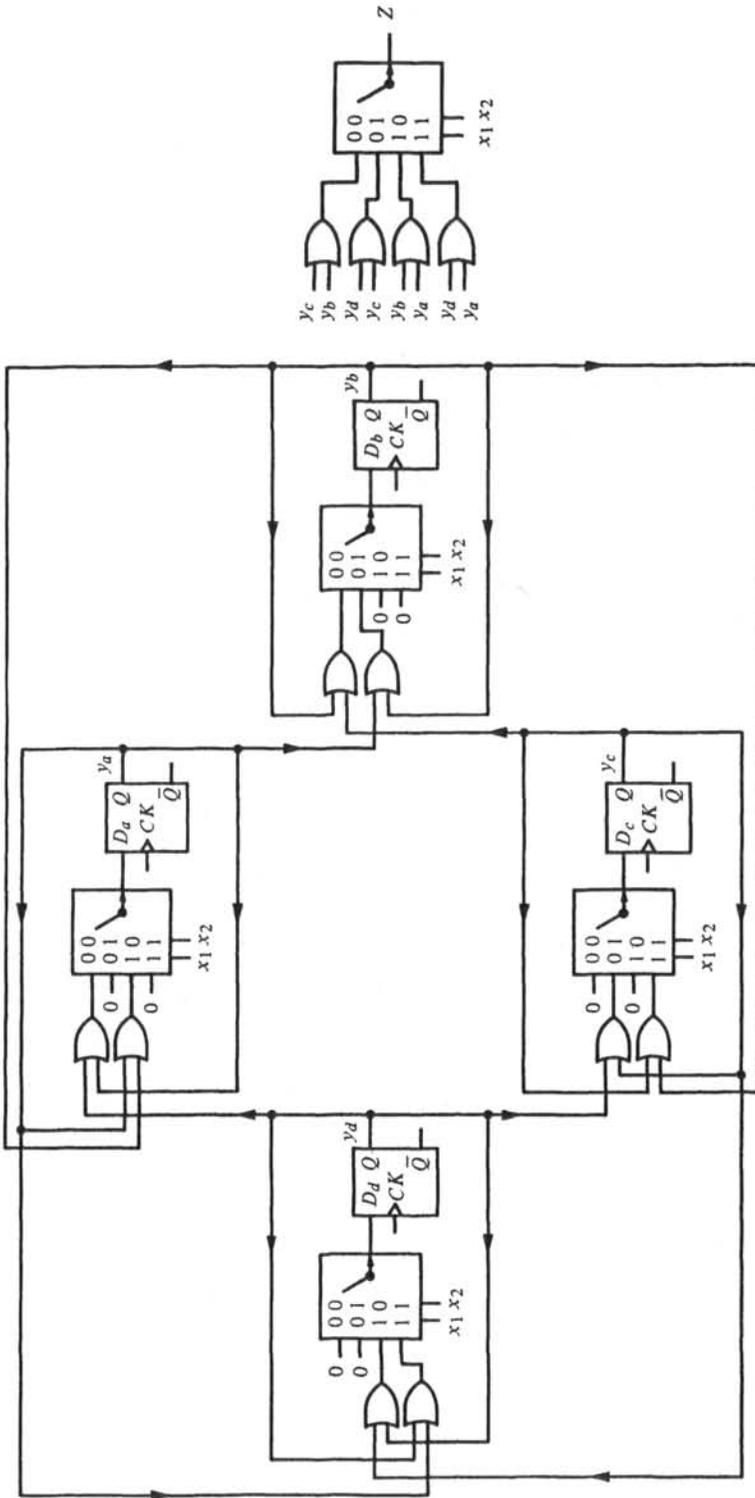


Fig. 6.22

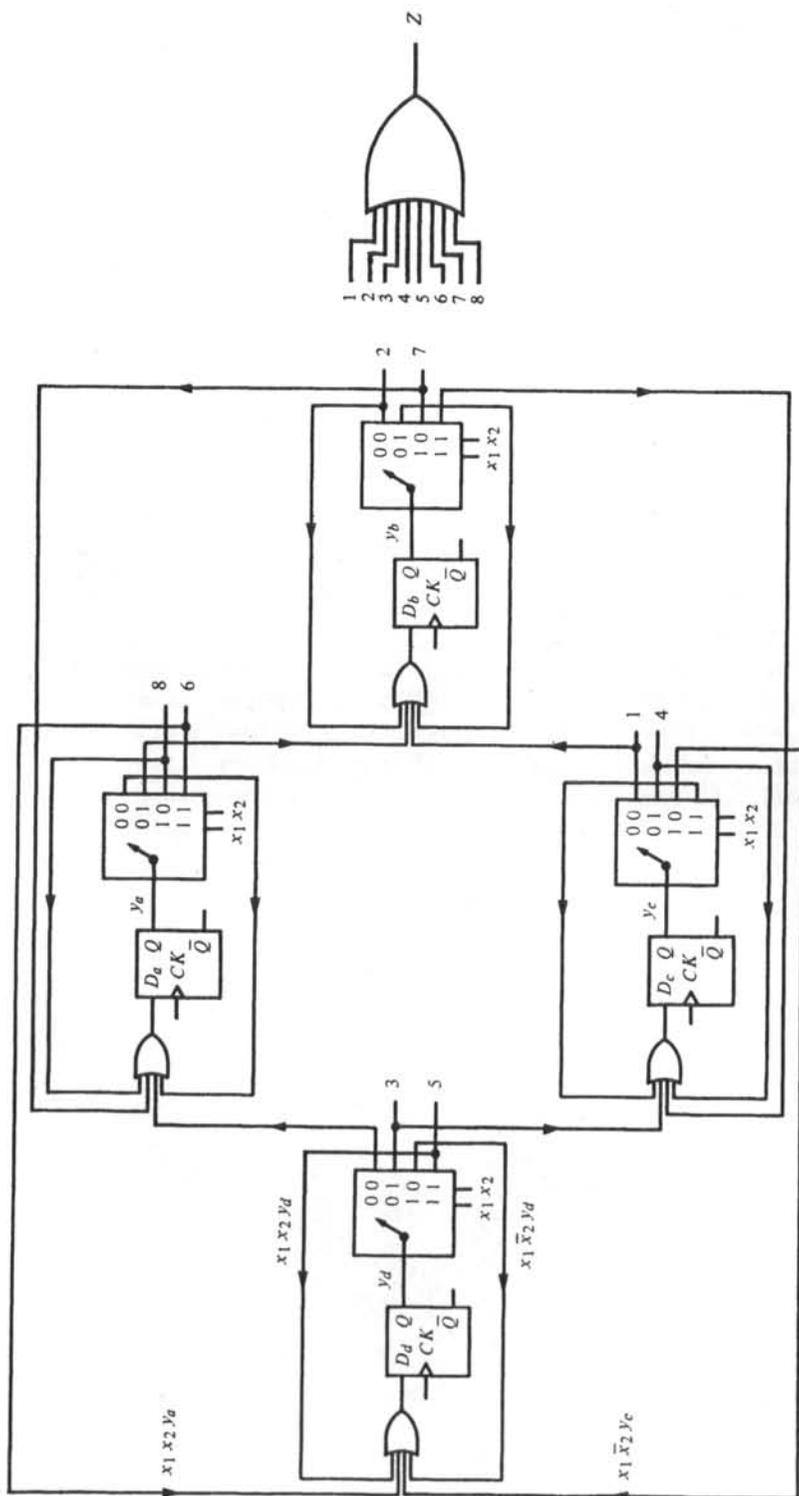


Fig. 6.23

sorties) et sont des *systèmes séquentiels répartis* (ou *cellulaires*), tandis que le second ne comporte qu'un seul système combinatoire et constitue un *système séquentiel concentré*.

### ■ 6.2.18 Propriétés

La comparaison du graphe de la figure 6.24 (identique à celui de la figure 5.17) et du logigramme de la figure 6.23 montre une étroite correspondance :

- chaque sommet du graphe ( $d, c, b, a$ ) est une bascule du logigramme ( $y_d, y_c, y_b, y_a$ );
- chaque flèche quittant un sommet est la sortie d'un démultiplexeur;
- chaque flèche atteignant un sommet est l'entrée d'une porte OU.

Cette propriété découle du codage 1 parmi 4 grâce auquel une seule des quatre variables internes (c'est-à-dire une seule bascule) vaut 1; cette variable (ou cette bascule) définit ainsi un seul sommet dans le graphe : c'est l'état du système à l'instant considéré. La sortie de chaque démultiplexeur réalise un état total ( $x_1 x_2 \cdot y_d$  par exemple) dont la représentation dans le graphe est une flèche ( $x_1, x_2 = 11$  quittant le sommet  $d$ ).

Les douze états  $\emptyset$  ne comportent aucune variable interne valant 1 ( $y_d, y_c, y_b, y_a = 0000$ ) ou plus d'une ( $y_d, y_c, y_b, y_a = 0011, 0101, \dots, 1110, 1111$ ). De tels états peuvent être représentés dans la table à quatre lignes (fig. 6.20) en imaginant qu'ils sont en dehors de cette table ( $y_d, y_c, y_b, y_a = 0000$ ) ou dans plusieurs lignes à la fois ( $y_d, y_c, y_b, y_a = 0011, 0101, \dots, 1110, 1111$ ); la même interprétation peut s'appliquer au graphe à quatre sommets de la figure 6.24. L'analyse des états  $\emptyset$  peut donc être effectuée sans calculer la table complète à seize états (fig. 6.21) ou le graphe correspondant.

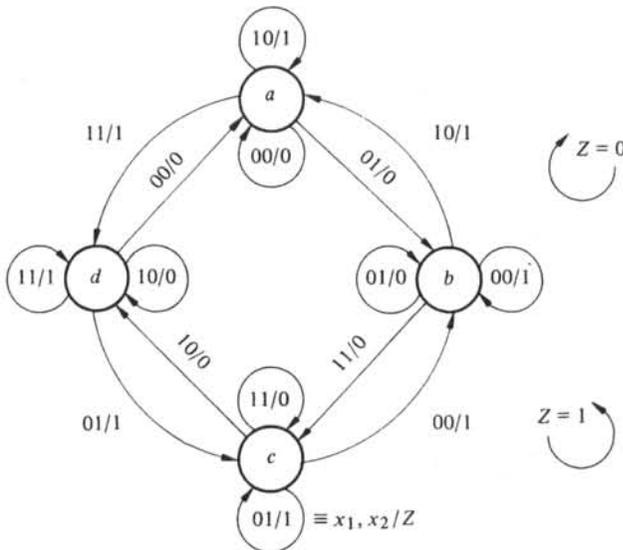


Fig. 6.24

### 6.2.19 Méthode

Les propriétés précédentes justifient la méthode de synthèse des systèmes séquentiels répartis :

- la conception produit un graphe ou une table (sect. 6.1) dont la réduction (§ 6.1.12) donne le nombre minimal d'états  $M$ ;
- un logigramme dont la topologie est identique au graphe des  $M$  états est réalisé soit avec des bascules et multiplexeurs (fig. 6.22), soit avec des bascules et démultiplexeurs (fig. 6.23);
- l'analyse des  $2^M - M$  états  $\emptyset$  est effectuée directement dans le graphe (ou la table) des  $M$  états.

Pratiquement, on supprime la nécessité d'analyser le logigramme en le munissant d'un système logique de correction : celui-ci peut être réalisé de façon asynchrone (entrées  $PR$  et  $CLR$  des bascules) ou synchrone (une solution est proposée dans l'exercice du paragraphe 6.2.21).

Une méthode semblable a été développée par David [71], [77], [78] pour réaliser des systèmes séquentiels asynchrones (c'est-à-dire sans bascules bistables).

### 6.2.20 Conclusion

Des codages distincts (minimal, 1 parmi  $M$ , etc...) produisent des systèmes séquentiels distincts (concentré, réparti, etc...) réalisant des tables d'états (ou des graphes) identiques. Le choix d'un tel codage répond à des critères empiriques et dépend souvent de contraintes non logiques : coût, fiabilité, facilité de conception et de dépannage, disponibilité de certains opérateurs, etc...

Comme dans le cas des systèmes combinatoires (§ 2.4.5), la synthèse des systèmes séquentiels est un problème à solutions multiples : du cahier des charges jusqu'au logigramme final existe une multitude de voies possibles. Au contraire, l'analyse de ces systèmes (sect. 5.2) nous permet de déterminer systématiquement les modes de représentation uniques (table et graphe) d'un logigramme donné.

#### ■ 6.2.21 Exercice

Pour effectuer l'analyse des états  $\emptyset$  du discriminateur, on demande de représenter les relations (6.21) à (6.24) dans la table de Karnaugh de la figure 6.21 et de tracer le graphe des états correspondant; en appliquant à chacun des états  $\emptyset$  la séquence des états d'entrée  $x_1, x_2$  donnée par la figure 6.2, on vérifiera que cette analyse est identique à celle entreprise directement dans la table de la figure 6.20 (ou dans le graphe de la figure 6.24) selon la méthode du paragraphe 6.2.18.

On modifie ensuite le logigramme de la figure 6.23 selon la figure 6.25; démontrer que ce système combinatoire itératif, rajouté au logigramme original, ramène le discriminateur (après une montée du signal d'horloge) d'un état  $\emptyset$  quelconque dans l'un des quatre états du graphe. Démontrer également la propriété particulière de l'entrée  $Y_d$  du dernier sous-système combinatoire :

$$Y_d \neq y_d^+ ; Y_d = 1 \quad (6.26)$$

puis déduire de cette relation une amélioration de la méthode de synthèse des systèmes séquentiels répartis (§ 6.2.19) ainsi qu'une simplification du logigramme de la figure 6.23.

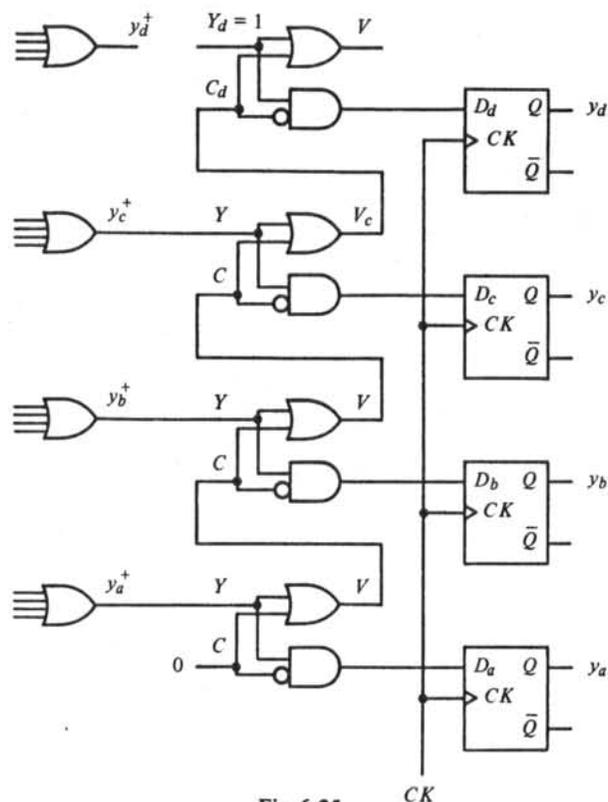


Fig. 6.25

### 6.2.22 Exercice

Effectuer le codage 1 parmi  $M$  des tables d'états réduites des exercices précédents : porte automatique (§ 6.1.18), passages à niveau (§ 6.1.19 et § 6.1.27) et serrure électronique (§ 6.1.21). Réaliser dans chaque cas un système séquentiel réparti pour démultiplexeurs, puis corriger le logigramme avec le système itératif du paragraphe 6.2.21 (fig. 6.25).

## 6.3 CONCEPTION DES SYSTÈMES À COMPORTEMENT SYNCHRONE

### 6.3.1 Exemple : détecteur de séquence

On veut reconnaître dans un texte français analysé caractère par caractère les mots terminés par *er* [72] (pp. 70 - 72). Le système séquentiel comporte une tête de lecture (fig. 6.26) lisant à chaque instant d'horloge un nouveau caractère; chacun d'eux appartient à l'une des quatre catégories suivantes qui constituent l'ensemble des états d'entrée  $X$  :

- la lettre  $e$ ;
- la lettre  $r$ ;
- l'ensemble des autres lettres de l'alphabet  $l, y$  compris le tiret;
- l'ensemble des signes de ponctuation  $p$  (avec l'espace blanc, mais sans le tiret).

La sortie  $Z$  prend la valeur 1 dans l'instant d'horloge qui suit immédiatement la détection d'un mot se terminant par  $er$ ; dans le cas contraire,  $Z$  vaut 0.

Il est possible enfin de coder chacun des quatre états d'entrée de  $X$  ( $e, r, l, p$ ) par une combinaison des valeurs de deux variables  $x_1, x_2$ ; la figure 6.27 illustre un tel codage.

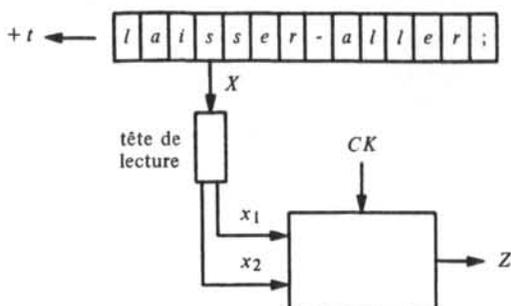


Fig. 6.26

$X$	$x_1$	$x_2$
$e$	0	0
$r$	0	1
$l$	1	1
$p$	1	0

Fig. 6.27

### 6.3.2 Mode de représentation : chronogramme

La recherche des mots terminés par  $er$  se ramène à la détection de la séquence d'entrée  $e \rightarrow r \rightarrow p$ ; une telle séquence, où les états d'entrée sont codés selon la figure 6.27, est représentée par le chronogramme de la figure 6.28.

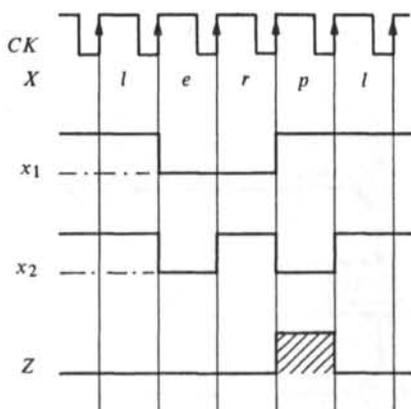


Fig. 6.28

Il est pratiquement exclu de décrire de la même façon toutes les séquences qui ne produisent pas l'état de sortie  $Z = 1$ ; comme dans le cas du discriminateur du sens de rotation (§ 6.1.5 et 6.1.7), le chronogramme est un mode de représentation incomplet dont on ne peut déduire que le squelette du graphe ou de la table d'états.

### 6.3.3 Définition : comportement synchrone

Conformément au cahier des charges et au chronogramme de la figure 6.28, on remarque que *l'état de sortie  $Z$  dépend à la fois de la durée des états d'entrée  $x_1, x_2$  et de leur ordre de succession*. Des séquences telles que  $e \rightarrow r \rightarrow r \rightarrow p$  ou  $e \rightarrow r \rightarrow r \rightarrow r \rightarrow p$ , où l'état d'entrée  $r$  apparaît pendant deux, respectivement quatre instants d'horloge consécutifs, ne sont pas conformes au cahier des charges et produisent une sortie  $Z = 0$ . On dira qu'un tel système possède un *comportement synchrone* par opposition au comportement asynchrone du paragraphe 6.1.6 qui ne distingue pas deux états d'entrée identiques successifs.

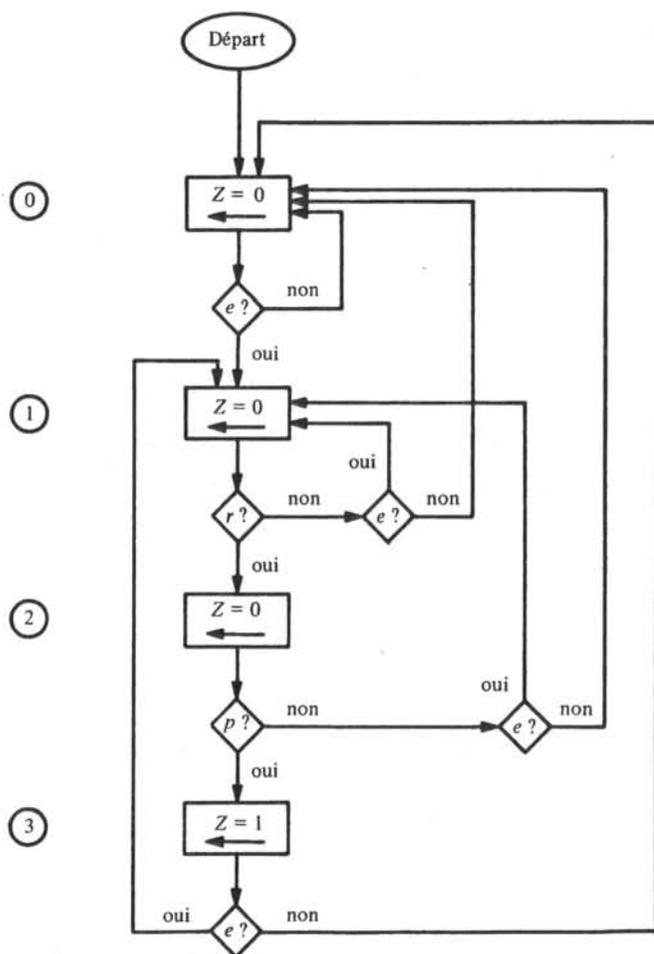


Fig. 6.29

### 6.3.4 Mode de représentation : organigramme

Un *algorithme* est une suite d'opérations fournissant la solution d'un problème et l'*organigramme* en constitue la représentation graphique.

L'organigramme de la figure 6.29 représente ainsi l'algorithme qui permet de tester n'importe quelle séquence d'entrée et de dire si elle est du type  $e \rightarrow r \rightarrow p$  (produisant un état de sortie  $Z = 1$ ) ou non ( $Z = 0$ ). Dans la figure 6.29, le symbole  $\leftarrow$  signifie : la tête de lecture avance au prochain caractère et lit l'état d'entrée  $x_1, x_2$  correspondant.

### 6.3.5 Modes de représentation : graphe et table d'états

En associant à chacun des rectangles de l'organigramme (fig. 6.29) un état interne ( $Y = 0, 1, 2$  et 3) on obtient le graphe de la figure 6.30 et la table d'états de la figure 6.32. On remarque que les deux dernières lignes de celle-ci sont dépourvues de tout état total stable (§ 5.2.7), ce qui illustre le comportement synchrone du système.

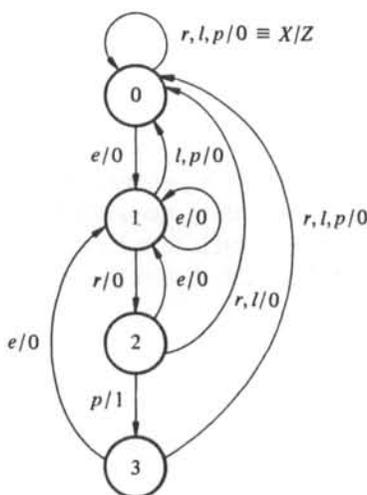


Fig. 6.30

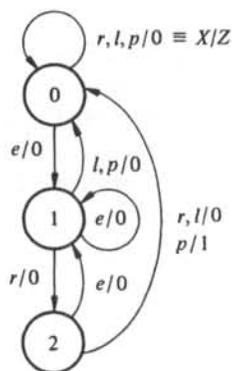


Fig. 6.31

### 6.3.6 Réduction de la table d'états

La table d'états de la figure 6.32 est complètement définie (§ 6.1.9). Les méthodes générales de réduction ont été citées au paragraphe 6.1.12; dans le cas particulier du détecteur de séquence, on remarque que les lignes 0 et 3 sont identiques (fig. 6.32) de telle sorte que l'état 3 est superflu et peut être remplacé par l'état 0. La table d'états réduite est celle de la figure 6.33, tandis que la figure 6.31 illustre le graphe correspondant.

### 6.3.7 Exercice

Un système séquentiel comporte deux entrées  $x_1, x_2$  et une sortie  $Z$ ; celle-ci prend la valeur 1 (durant une période du signal d'horloge  $CK$ ) après toute séquence d'entrée définie par :

$CK$	:	1	2	3	4	5
$x_1$	:	0	0	1	0	0
$x_2$	:	1	0	1	1	0
$Z$	:	0	0	0	0	1

où  $CK = 1, 2, \dots, 5$  désigne cinq instants d'horloge successifs. Déterminer le graphe et la table d'états (éventuellement réduite) de ce détecteur de séquence.

$Y^+, Z$	$x_1, x_2$			
	00 (e)	01 (r)	11 (l)	10 (p)
0	1,0	(0,0)	(0,0)	(0,0)
1	(1,0)	2,0	0,0	0,0
2	1,0	0,0	0,0	3,1
3	1,0	0,0	0,0	0,0
$Y$				

Fig. 6.32

$Y^+, Z$	$x_1, x_2$			
	00 (e)	01 (r)	11 (l)	10 (p)
0	1,0	(0,0)	(0,0)	(0,0)
1	(1,0)	2,0	0,0	0,0
2	1,0	0,0	0,0	0,1
$Y$				

Fig. 6.33

### 6.3.8 Exercice

Un système séquentiel comporte deux entrées  $x_1, x_2$  et une sortie  $Z$  qui doit détecter toute séquence de quatre états d'entrée successifs vérifiant la relation  $x_1 = x_2$ ; sitôt qu'une telle séquence est repérée, la sortie  $Z$  prend la valeur 1 et la conserve aussi longtemps que  $x_1 = x_2$ . Déterminer le graphe et la table d'états (éventuellement réduite) de ce comparateur de bits.

### 6.3.9 Exercice

Un compteur synchrone programmable (§ 5.2.16) admet les quatre fonctionnements suivants : blocage dans un état de repos, comptage par 2, par 3 ou par 4. Déterminer le graphe et la table d'états.

### 6.3.10 Exercice

Un système séquentiel possède une seule entrée  $x$  et doit détecter toute séquence présentant au moins deux 1 successifs et toute séquence présentant au moins quatre 0 successifs; sitôt qu'une telle séquence est repérée, la sortie  $Z$  vaut 1 et garde cette valeur jusqu'au changement suivant de la variable d'entrée  $x$ . Déterminer le graphe et la table d'états (éventuellement réduite).

### 6.3.11 Exercice

Un système compare des mots de quatre bits présentés séquentiellement sur deux entrées  $x_1, x_2$ . La sortie  $Z$  prend la valeur 1 (pendant une période du signal d'horloge) sitôt que deux mots comparés coïncident bit à bit. On observe par exemple les séquences suivantes :

CK	:	1	2	3	4	5	6	7	8	9	10	11	12
$x_1$	:	1	1	0	1	1	0	1	1	1	0	1	0
$x_2$	:	0	1	0	1	1	0	1	1	1	0	1	1
Z	:	0	0	0	0	0	0	0	1	0	0	0	0

Déterminer le graphe et la table d'états (éventuellement réduite) de ce comparateur de mots.

### ■ 6.3.12 Exercice

Un système séquentiel comporte deux entrées  $x_1, x_2$  et une sortie Z définie de la façon suivante :

- si  $x_2 = 0$ , la valeur de Z à l'instant d'horloge présent est égale à celle de  $x_1$  observée à l'instant d'horloge précédent;
- si  $x_2 = 1$ , la valeur de Z à l'instant présent est égale à celle de  $x_1$  observée deux instants d'horloge auparavant.

Déterminer le graphe et la table d'états (éventuellement réduite) de ce *décalage programmable*. S'inspirer au besoin des propriétés du registre à décalage (sect. 5.3).

## 6.4 RÉALISATION DES SYSTÈMES À COMPORTEMENT SYNCHRONE

### 6.4.1 Introduction

Par pure commodité on a distingué dans ce chapitre la conception des systèmes séquentiels à comportement asynchrone (sect. 6.1) de celle des systèmes à comportement synchrone (sect. 6.3). Les méthodes de réalisation sont identiques pour ces deux comportements, de telle sorte que les résultats de la section 6.2 (discriminateur du sens de rotation) peuvent être appliqués sans restriction à l'exemple du détecteur de séquence (sect. 6.4); réciproquement, toutes les réalisations de cette section sont également valables pour les systèmes à comportement asynchrone.

### 6.4.2 Codage minimal : détecteur de séquence

La table d'états réduite de ce dispositif (fig. 6.33) comporte trois états. Les formules (6.1) et (6.4) définissent deux variables internes  $y_1, y_2$  et trois familles de codes équivalents (fig. 6.34) [10] (p. 276). En choisissant la famille No 1 (fig. 6.34) avec :

$$y_1 = Y_1 ; y_2 = Y_2 \quad (6.27)$$

on détermine un codage possible de la table d'états (fig. 6.35).

Etats Y	Famille No 1 $Y_1 Y_2$	Famille No 2 $Y_1 Y_2$	Famille No 3 $Y_1 Y_2$
0	0 0	0 0	0 1
1	0 1	1 1	1 1
2	1 1	0 1	0 0

Fig. 6.34

$y_1^+, y_2^+, Z$	$x_1 x_2$			
	00	01	11	10
0 $\equiv$ 00	0 1,0	00,0	00,0	00,0
1 $\equiv$ 01	0 1,0	1 1,0	00,0	00,0
2 $\equiv$ 11	0 1,0	00,0	00,0	00, 1
10	-	-	-	-
$Y \equiv y_1, y_2$				

Fig. 6.35

Les deux variables internes  $y_1, y_2$  déterminent quatre états, de telle sorte que l'un d'eux ( $y_1, y_2 = 10$ ) est inutilisé : il existe par conséquent quatre états totaux indifférents, marqués par des tirets dans la figure 6.35, dont les états futurs ( $y_1^+, y_2^+$ ) et de sortie ( $Z$ ) ont la valeur  $\emptyset$ .

### 6.4.3 Réalisation de logigrammes

Si l'on dispose de bascules  $D$  (3.63), le calcul peut être abrégé (§ 6.2.6). En remarquant que la table d'états de la figure 6.35 est semblable à une table de Karnaugh, on peut obtenir directement les formes canoniques et simplifiées des fonctions recherchées :

$$y_1^+ = D_1(x_1, x_2, y_1, y_2) = \Sigma 5 + \emptyset 2, 6, 10, 14 = \bar{x}_1 x_2 \bar{y}_1 y_2 \quad (6.28)$$

$$y_2^+ = D_2(x_1, x_2, y_1, y_2) = \Sigma 0, 1, 3, 5 + \emptyset 2, 6, 10, 14 = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 \bar{y}_1 y_2 \quad (6.29)$$

$$Z(x_1, x_2, y_1, y_2) = \Sigma 11 + \emptyset 2, 6, 10, 14 = x_1 \bar{x}_2 y_1 \quad (6.30)$$

Les logigrammes des figures 6.36 et 6.38 illustrent deux réalisations possibles des relations (6.28) à (6.30) : la première avec des portes ET, OU, la seconde avec un démultiplexeur à quatre variables (§ 1.5.3) et une porte OU.

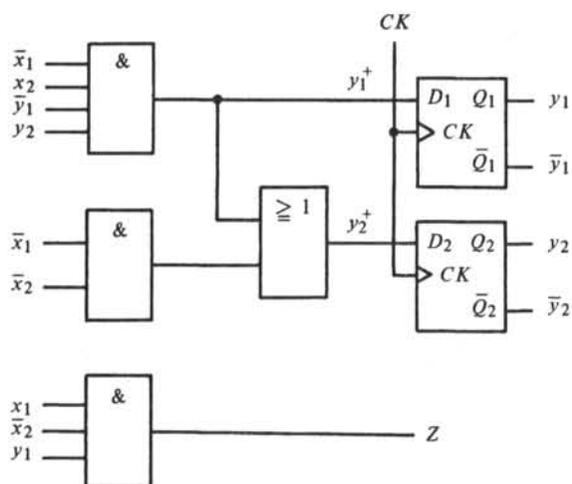


Fig. 6.36

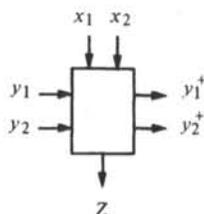


Fig. 6.37

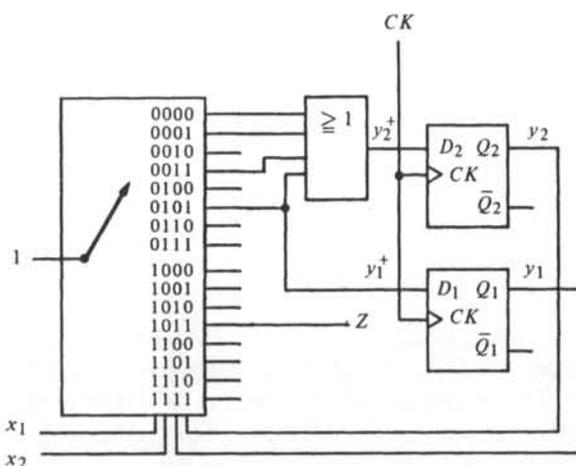


Fig. 6.38

#### 6.4.4 Commentaire

La réduction de la table d'états du détecteur de séquence (§ 6.3.6) n'a pas diminué le nombre des variables internes (pour  $M=3$  comme pour  $M=4$  on a  $m=2$ ), mais a produit un état indifférent ( $y_1, y_2 = 10$ ). Celui-ci introduit dans la table (fig. 6.35) des valeurs  $\emptyset$  (pour  $y_1^+, y_2^+$  et  $Z$ ) dont la présence a deux conséquences :

- elle peut améliorer la simplification des fonctions d'excitation et de sortie;
- elle nécessite une analyse afin de détecter d'éventuels cycles parasites.

Pour tout codage minimal, la formule (6.1) montre que si une table de  $M$  états, non réduite, nécessite  $m$  variables internes alors il faut que le nombre des états de la table réduite soit égal ou inférieur à  $2^{m-1}$  pour abaisser le nombre de ces variables.

#### 6.4.5 Exercice

Le logigramme de la figure 6.39 est une réalisation intuitive du détecteur de séquence. Calculer sa table d'états et démontrer que celle-ci réalise un codage particulier de la figure 6.34, obtenu par permutation et/ou inversion des variables  $Y_1, Y_2$  d'une famille.

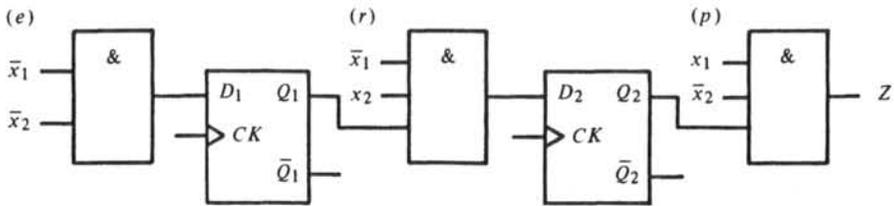


Fig. 6.39

#### 6.4.6 Exercice

Réaliser la table d'états assignée du détecteur de séquence (fig. 6.35) pour des bascules  $JK$ , déterminer deux logigrammes (l'un avec des portes ET, OU, l'autre avec un démultiplexeur et des portes OU) et comparer ceux-ci aux réalisations des figures 6.36 et 6.38.

#### 6.4.7 Exercice

Analyser les logigrammes des figures 6.36 et 6.38, puis déterminer les valeurs indifférentes des états futurs et de sortie dans la table de la figure 6.35. Tracer les graphes correspondants et comparer ceux-ci aux figures 6.30 et 6.31.

#### 6.4.8 Exercice

Effectuer un codage minimal des tables d'états réduites des exercices précédents : détecteur de séquence (§ 6.3.7), comparateur de bits (§ 6.3.8), compteur synchrone programmable (§ 6.3.9 : on demande dans ce cas le code binaire pur), détecteur de séquences (§ 6.3.10), comparateur de mots (§ 6.3.11) et délai programmable (§ 6.3.12 : s'inspirer dans ce cas des propriétés du registre à décalage). Réaliser pour chaque système deux logigrammes avec des bascules  $D$  et  $JK$ , puis analyser ceux-ci pour déterminer les états futurs et/ou les états de sortie qui ne sont pas définis dans la table d'états assignée.

#### 6.4.9 Codage 1 parmi $M$ : détecteur de séquence

En effectuant le codage 1 parmi 3 du graphe des états réduit (fig. 6.31), on obtient le graphe assigné de la figure 6.40 où  $y_2, y_1$  et  $y_0$  sont les trois variables internes. Pour améliorer la lisibilité de cette figure, on a supprimé tous les états de sortie  $Z = 0$ .

#### 6.4.10 Réalisation de logigrammes

La méthode de synthèse du paragraphe 6.2.19 peut s'appliquer au graphe de la figure 6.40 et produit deux systèmes séquentiels répartis : l'un avec multiplexeurs (fig. 6.41), l'autre avec démultiplexeurs (fig. 6.42). Les particularités de ces deux réalisations ont déjà fait l'objet d'un commentaire (§ 6.2.16).

#### 6.4.11 Exercice

Dresser la table d'états du graphe assigné de la figure 6.40 puis réaliser celle-ci avec un système séquentiel concentré (§ 6.2.17) comportant des bascules  $D$  et des portes NAND. Comparer le logigramme avec ceux des figures 6.36 et 6.39 (codage minimal), puis des figures 6.41 et 6.42 (codage 1 parmi 3 : système réparti).

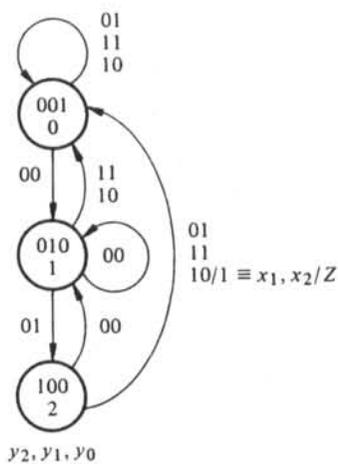


Fig. 6.40

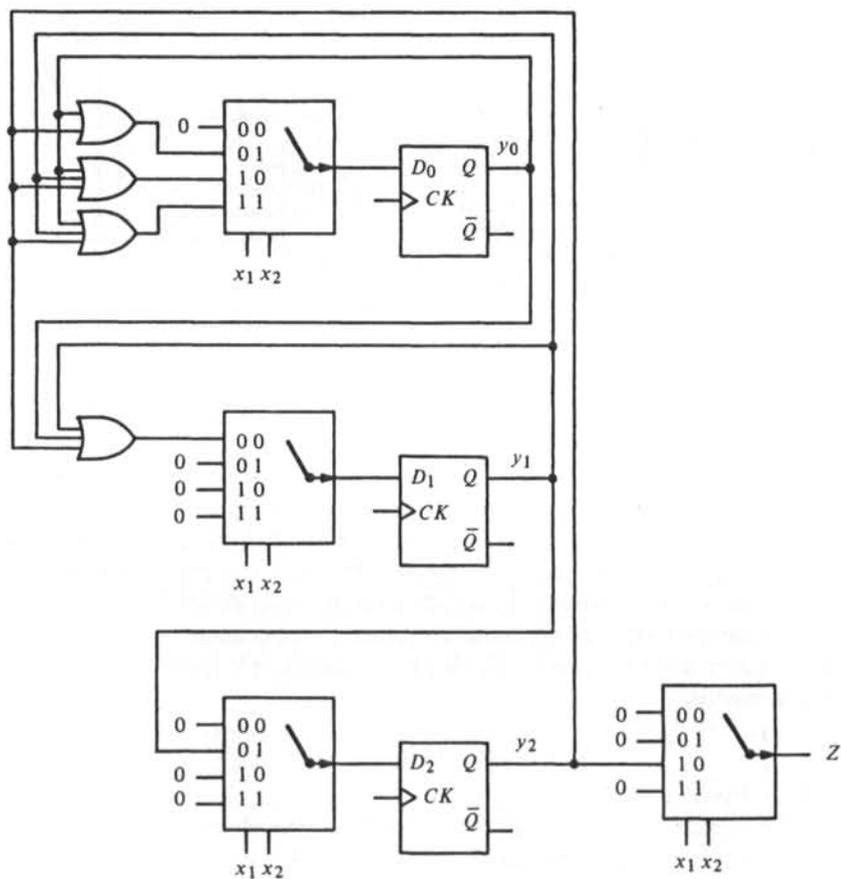


Fig. 6.41

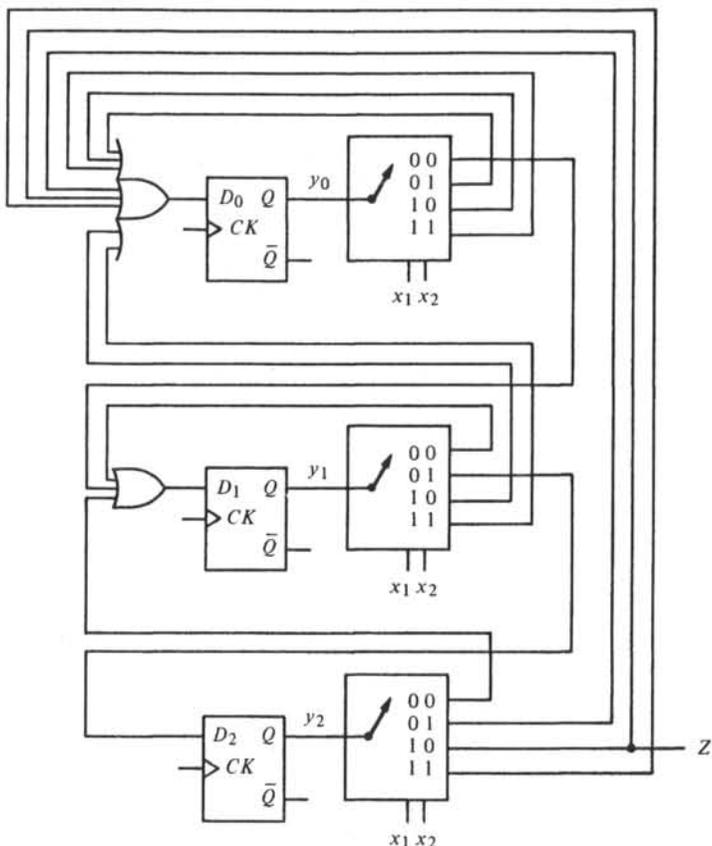


Fig. 6.42

#### ■ 6.4.12 Exercice

Analyser les logigrammes des figures 6.41 et 6.42 dans le graphe de la figure 6.40 pour tous les états  $\emptyset$  selon la méthode du paragraphe 6.2.18.

Corriger ensuite le logigramme de la figure 6.42 en ajoutant un système combinatoire itératif semblable à celui de la figure 6.25; démontrer ensuite que la relation (6.26) entraîne une simplification substantielle du logigramme modifié.

Examiner enfin le logigramme de la figure 6.41 et démontrer que la sortie de certaines portes OU peut être remplacée par une constante logique ou par la sortie  $\bar{Q} \equiv \bar{y}$  d'une bascule.

#### 6.4.13 Exercice

Effectuer le codage 1 parmi  $M$  des tables d'états réduites des exercices précédents : compteur synchrone programmable (§ 6.3.9) et délai programmable (§ 6.3.12). Réaliser dans chaque cas un système séquentiel réparti pour démultiplexeurs, puis corriger le logigramme avec le système itératif du paragraphe 6.2.21 (fig. 6.25).

### 6.4.14 Exercice

Effectuer la synthèse d'un compteur synchrone par quatre selon la méthode du paragraphe 6.2.19; réaliser le système séquentiel réparti pour démultiplexeurs et comparer le logigramme à celui d'un compteur en anneau (§ 4.2.29). Analyser les états  $\emptyset$  selon la méthode du paragraphe 6.2.18, puis corriger le logigramme avec le système itératif du paragraphe 6.2.21 (fig. 6.25) pour éliminer les éventuels cycles parasites.

### 6.4.15 Application : bascules bistables

Toute bascule bistable en fonctionnement normal (§ 3.2.6) est un système séquentiel synchrone, de telle sorte que les méthodes de synthèse de ce chapitre s'appliquent également à cet opérateur. Un problème classique est celui de la transformation d'une bascule donnée en une bascule d'un type différent : on cherche par exemple, dans un système logique quelconque (fig. 6.43), à remplacer une bascule défaillante (No 1 : type *SR*) par une autre bascule (No 2 : type *T*) et un éventuel système combinatoire d'interface. Cette transformation se ramène à la réalisation d'une table d'états assignée (celle de la bascule No 1 : *SR*) avec une bascule donnée (la bascule No 2 : *T*).

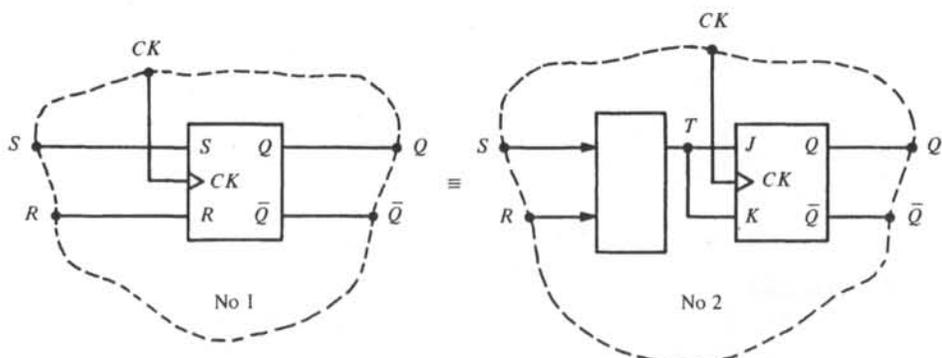


Fig. 6.43

### 6.4.16 Exemple

La table d'états de la bascule *SR* (fig. 3.20) est rappelée dans la figure 6.44. En recourant à la méthode tabulaire de synthèse (§ 4.2.8), il est possible de calculer pour chaque transition  $Q, Q^+$  la valeur de la fonction d'excitation  $T$ , déterminée par la table des transitions de cette bascule (fig. 3.44). On obtient directement une table de Karnaugh (fig. 6.45) qui donne, après simplification, l'équation recherchée :

$$T = S\bar{Q} + RQ \quad (6.31)$$

Une réalisation possible est celle du logigramme de la figure 6.46.

	SR			
$Q^+$	00	01	11	10
0	0	0	$\emptyset$	1
1	1	0	$\emptyset$	1
$Q$				

Fig. 6.44

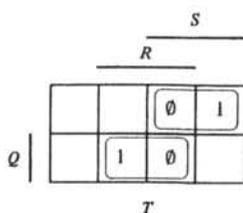


Fig. 6.45

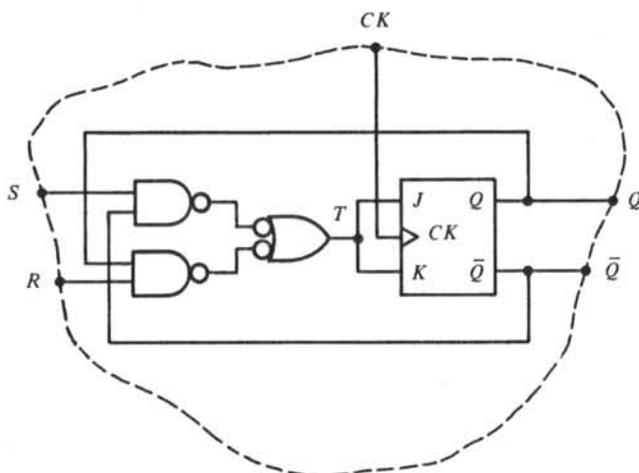


Fig. 6.46

## 6.4.17 Exercice

Déterminer l'équation caractéristique de la bascule réalisée par la figure 6.46 et démontrer que cette équation est identique à celle de la bascule JK (3.56) si l'on vérifie les relations  $S \equiv J$  et  $R \equiv K$ . Justifier ce résultat en se référant au commentaire du paragraphe 3.3.7.

No 2 \ No 1	SR	JK	T	D
SR		$J = S$ $K = R$	$T = S\bar{Q} + RQ$	$D = S + \bar{R}Q$
JK	$S = J\bar{Q}$ $R = KQ$		$T = J\bar{Q} + KQ$	$D = J\bar{Q} + \bar{K}Q$
T	$S = T\bar{Q}$ $R = TQ$	$J = T$ $K = T$		$D = T * Q$
D	$S = D$ $R = \bar{D}$	$J = D$ $K = \bar{D}$	$T = D * Q$	

Fig. 6.47

### 6.4.18 Exercice

La figure 6.47 résume toutes les réalisations possibles d'une bascule donnée (No 1) avec une bascule différente (No 2). Effectuer les calculs des fonctions d'excitation puis, dans le cas de la bascule *SR* (type No 1), analyser le logigramme et déterminer l'équation caractéristique de la bascule No 2. Lorsque la bascule No 2 est du type *SR*, vérifier l'équation de condition (3.48). Comparer enfin les résultats de la figure 6.47 avec ceux des exercices similaires (§ 3.3.14 et § 3.4.14).

## 6.5 RÉALISATION AVEC DES REGISTRES À DÉCALAGE

### 6.5.1 Définitions

Etant donné un système séquentiel à comportement synchrone, on dira qu'une succession de  $K$  états d'entrée est une *séquence d'entrée déterminante de longueur  $K$*  si l'état de sortie final produit par cette séquence est indépendant de l'état interne initial auquel elle est appliquée.

Le système séquentiel est dit à *mémoire finie* si toute séquence d'entrée de longueur égale ou supérieure à  $K$  est déterminante,  $K$  étant un entier fixe caractéristique du système.

### 6.5.2 Exemple : détecteur de séquence

On constate dans le graphe de la figure 6.31 que toute séquence d'entrée (de longueur égale ou supérieure à 3) terminée par :

$$X = e \rightarrow r \rightarrow p \text{ ou } x_1, x_2 = 00 \rightarrow 01 \rightarrow 10 \quad (6.32)$$

produit à partir de n'importe quel état interne (0, 1 ou 2) le même état de sortie  $Z = 1$ . Ces séquences (dont l'une est rappelée dans la figure 6.48) sont donc déterminantes.

On vérifie dans le même graphe (fig. 6.31) que toute autre séquence, de longueur égale ou supérieure à 3, appliquée à n'importe quel état interne produit le même état de sortie  $Z = 0$ .

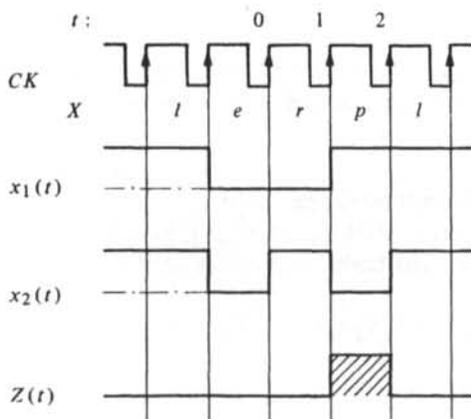


Fig. 6.48

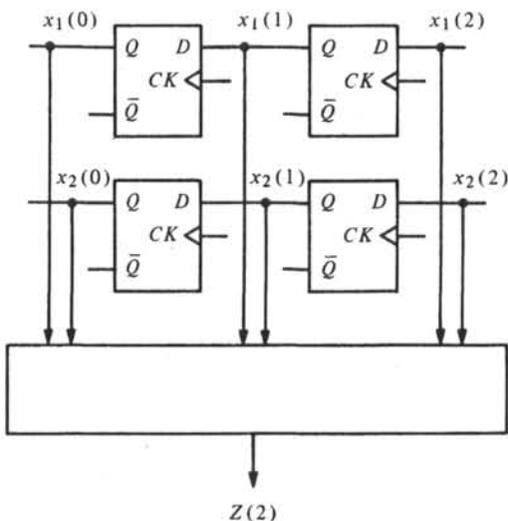


Fig. 6.49

Toute séquence d'entrée de longueur égale ou supérieure à 3 est déterminante : le détecteur étudié est donc un système séquentiel à mémoire finie.

### 6.5.3 Réalisation avec registres à décalage

Il est alors possible de réaliser le détecteur de séquence de la manière suivante :

- deux registres à décalage de deux bits chacun (§ 5.3.1 et 5.3.3) sont disposés en parallèle selon la figure 6.49; chacun des registres est attaqué par l'une des variables d'entrée  $x_1(t)$  et  $x_2(t)$ ; à un instant d'horloge donné  $t = 2$  (fig. 6.49), ils mémorisent les deux états d'entrée antérieurs  $x_1(0)$ ,  $x_2(0)$  et  $x_1(1)$ ,  $x_2(1)$ .
- Un système combinatoire comporte six entrées qui, à l'instant  $t = 2$ , sont  $x_1(0)$ ,  $x_2(0)$ ,  $x_1(1)$ ,  $x_2(1)$ ,  $x_1(2)$  et  $x_2(2)$ . La séquence déterminante (6.32) peut être détectée ou non, et la sortie  $Z(2)$  peut prendre la valeur 1 ou 0.

Les registres à décalage effectuent en fait une transformation temps-espace (§ 2.7.6 et 5.4.5) qui est aussi une conversion série-parallèle.

### 6.5.4 Système combinatoire à structure parallèle

Les six variables d'entrée  $x_1(0)$ ,  $x_2(0)$ ,  $x_1(1)$ ,  $x_2(1)$ ,  $x_1(2)$ ,  $x_2(2)$  déterminent  $2^6 = 64$  états; un seul d'entre eux traduit la présence de la séquence déterminante (6.32):

$$x_1(0), x_2(0), x_1(1), x_2(1), x_1(2), x_2(2) = 000110 \quad (6.33)$$

de telle sorte qu'on écrit directement la forme canonique de la fonction de sortie  $Z(2)$ :

$$Z(2) = \overline{x_1(0)} \cdot \overline{x_2(0)} \cdot \overline{x_1(1)} \cdot x_2(1) \cdot x_1(2) \cdot \overline{x_2(2)} \quad (6.34)$$

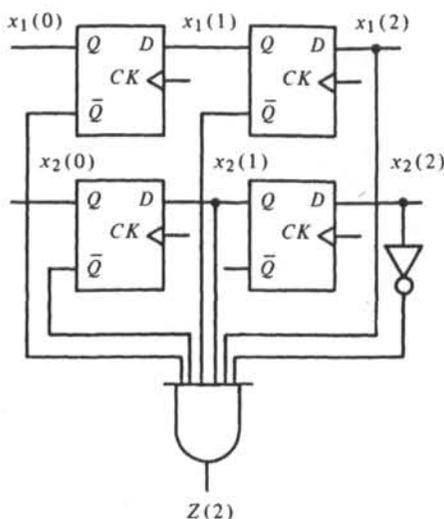


Fig. 6.50

dont une réalisation possible est illustrée par la figure 6.50; le logigramme de celle-ci peut être comparé à ceux des figures 6.36, 6.38, 6.39 (codages minimaux), 6.41 et 6.42 (codage 1 parmi 3).

### ■ 6.5.5 Système combinatoire à structure itérative

On a déjà effectué la synthèse d'un système séquentiel réalisant le détecteur de séquence (fig. 6.36) : à un instant d'horloge  $t$ , les variables  $x_1, x_2$  définissent un état d'entrée  $x_1(t), x_2(t)$  et la détection de la séquence (6.32) nécessite trois instants d'horloge consécutifs : on a une représentation temporelle de l'information d'entrée et le calcul de la fonction  $Z$  est séquentiel.

Dans la figure 6.49 on dispose simultanément de trois états d'entrée successifs  $x_1(t), x_2(t)$  avec  $t = 0, 1, 2$  : on obtient donc une représentation spatiale de l'information et la fonction  $Z(2)$  peut être calculée de façon combinatoire.

On rappelle alors les résultats de la section 5.4 consacrée à la synthèse d'un comparateur de deux nombres : la structure itérative à  $n + 1$  sous-systèmes combinatoires (représentation spatiale de l'information) est équivalente à la structure série comportant un seul sous-système et une bascule  $D$  (représentation temporelle); la table d'états de la structure série est identique à la table de vérité du sous-système combinatoire.

Cette équivalence de structures s'applique aussi à notre exemple du détecteur de séquence : en partant de la représentation spatiale donnée par les registres à décalage (fig. 6.49) on doit réaliser une structure itérative à trois sous-systèmes combinatoires dont la table de vérité est identique à la table d'états de la figure 6.35. Les équations décrivant ces sous-systèmes sont donc semblables aux relations (6.28), (6.29) et (6.30) :

$$y_1^+(x_1, x_2, y_1, y_2) = \Sigma 5 + \emptyset 2, 6, 10, 14 = \bar{x}_1 x_2 \bar{y}_1 y_2$$

$$y_2^+(x_1, x_2, y_1, y_2) = \Sigma 0, 1, 3, 5 + \emptyset 2, 6, 10, 14 = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 \bar{y}_1 y_2$$

$$Z(x_1, x_2, y_1, y_2) = \Sigma 11 + \emptyset 2, 6, 10, 14 = x_1 \bar{x}_2 y_1$$

qui sont réalisées dans la figure 6.36 et représentées par le schéma de la figure 6.37.

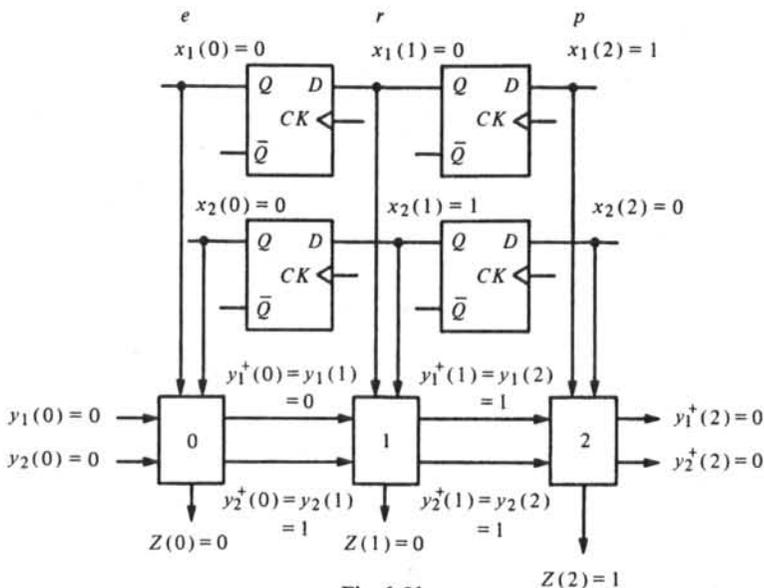


Fig. 6.51

$y_1^+ y_2^+, Z$	$x_1 x_2$			
	00 (e)	01 (r)	11 (l)	10 (p)
00	01,0	00,0	00,0	00,0
01	01,0	11,0	00,0	00,0
11	01,0	00,0	00,0	00,1
10	-	-	-	-
$y_1 y_2$				

Fig. 6.52

La structure itérative de la figure 6.51 est obtenue en juxtaposant les trois sous-systèmes dont le dernier produit la fonction  $Z(2)$  recherchée. Chacun de ces sous-systèmes réalise les fonctions  $y_1^+, y_2^+, Z$  de la table d'états (figure 6.52, identique à la figure 6.35) et simule à tout instant un état total de celle-ci. Pour illustrer ce fonctionnement, on a rappelé dans la figure 6.51 les valeurs de la séquence déterminante (6.32) ainsi que les états totaux correspondants qui sont encadrés dans la table (fig. 6.52). Les valeurs  $y_1(0)$  et  $y_2(0)$  sont des constantes qui représentent l'état interne initial  $y_1, y_2$ ; or une séquence déterminante, par définition, produit le même état de sortie à partir de n'importe quel état initial. On a donc choisi arbitrairement  $y_1(0), y_2(0) = 00$  (fig. 6.51 et 6.52), qui est l'un des trois états assignés (l'état indifférent  $y_1, y_2 = 10$  étant inutilisable).

La figure 6.51 montre que le système séquentiel global est lui-même itératif, chaque sous-ensemble étant constitué par le même assemblage de deux bascules et d'un système combinatoire.

### 6.5.6 Méthode

Tout système séquentiel à comportement synchrone et à mémoire finie peut être réalisé avec des registres à décalage en procédant de la façon suivante [10] (pp. 249-253) :

- on détermine le graphe ou la table d'états réduite et l'on y met en évidence la séquence d'entrée déterminante de longueur  $K$  produisant l'état de sortie  $Z = 1$ ;
- pour  $n$  variables d'entrée, il faut disposer en parallèle  $n$  registres à décalage de  $K - 1$  bits chacun;
- si le système combinatoire est réalisé avec une structure parallèle la fonction de sortie  $Z$  dépend de  $K \cdot n$  variables et peut être définie par une table de vérité; dans le cas d'une structure itérative, c'est la table d'états du système séquentiel qui définit les fonctions des sous-systèmes combinatoires.

### 6.5.7 Exercice

Analyser le logigramme de la figure 6.50 (voir l'exercice du paragraphe 5.3.7), puis comparer la table d'états calculée à celle de la figure 6.33. Est-il possible de réduire la première pour obtenir la seconde ?

### ■ 6.5.8 Exercice

Dans la structure itérative du détecteur de séquence (§ 6.5.5) chaque sous-système combinatoire est défini par une table de vérité qui est identique à la table d'états assignée de la figure 6.35 (codage minimal). On demande de calculer et de réaliser les sous-systèmes combinatoires définis par le graphe des états assigné de la figure 6.40 (codage 1 parmi 3), puis de comparer cette solution à la précédente.

### ■ 6.5.9 Exercice

Est-il possible de réaliser avec des registres à décalage (selon la méthode du paragraphe 6.5.6) les systèmes séquentiels des exercices précédents : détecteurs de séquences (§ 6.3.7 et 6.3.10), comparateur de bits (§ 6.3.8) et comparateur de mots (§ 6.3.11)? Dans l'affirmative, calculer les structures parallèle et itérative du système combinatoire.

### 6.5.10 Exemple : discriminateur du sens de rotation

Le graphe des états réduit de ce système (fig. 5.17) est rappelé dans la figure 6.53. *Si l'on admet un comportement synchrone*, c'est-à-dire s'il existe à chaque instant d'horloge un état d'entrée  $x_1, x_2$  différent du précédent (à l'exception des variations simultanées de  $x_1$  et  $x_2$  qui sont exclues par le cahier des charges), alors on constate que toute séquence d'entrée de longueur égale ou supérieure à 2, appliquée à n'importe quel état interne et terminée par :

$$x_1, x_2 = 00 \rightarrow 10 ; x_1, x_2 = 10 \rightarrow 11 ; x_1, x_2 = 11 \rightarrow 01 ; x_1, x_2 = 01 \rightarrow 00 \quad (6.35)$$

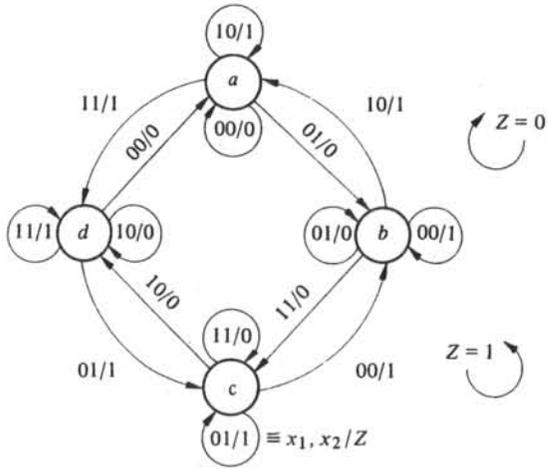


Fig. 6.53

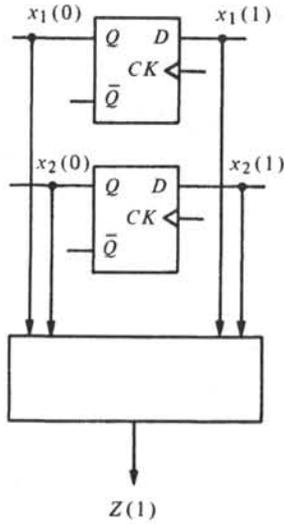


Fig. 6.54

	$x_1(0)x_2(0)$			
	00	01	11	10
$x_1(1)x_2(1)$	00	1	0	
	01	0	1	0
	11	0		1
	10	1	0	0
	$Z(1)$			

Fig. 6.55

produit un état de sortie  $Z = 1$ . Toute autre séquence (de longueur égale ou supérieure à deux) entraîne  $Z = 0$  à partir d'un état quelconque : le système est donc à mémoire finie.

### 6.5.11 Réalisation avec registres à décalage

On peut alors appliquer la méthode du paragraphe 6.5.6 : deux registres d'un seul bit chacun sont disposés selon la figure 6.54 et mémorisent au temps  $t = 1$  l'état d'entrée  $x_1(0), x_2(0)$ . La fonction  $Z(1)$  réalisant la structure parallèle peut être déterminée par la relation (6.35) et simplifiée à l'aide d'une table de Karnaugh (fig. 6.55) dans laquelle les conditions  $\emptyset$  traduisent des séquences irréalisables (variations simultanées de  $x_1$  et  $x_2$ , égalité de deux états d'entrée successifs :  $x_1(0), x_2(0) = x_1(1), x_2(1)$ ). L'expression algébrique de  $Z(1)$  s'écrit alors :

$$\begin{aligned} Z(1) &= x_2(0) \cdot \overline{x_1(1)} + \overline{x_2(0)} \cdot x_1(1) \\ &= x_2(0) \oplus x_1(1) \end{aligned} \quad (6.36)$$

Mais la fonction  $Z(1)$  réalise le cahier des charges seulement si l'on vérifie l'hypothèse admise plus haut : tout état d'entrée  $x_1, x_2$  diffère du précédent. Cette hypothèse n'est pas valable dans le comportement habituel du discriminateur, qui est asynchrone (fig. 6.2). Nous cherchons alors à réaliser un système séquentiel particulier, dit de synchronisation, capable de transformer une séquence d'entrée quelconque en une nouvelle séquence dont chaque état diffère du précédent.

### 6.5.12 Description : système séquentiel de synchronisation

Le logigramme de la figure 6.57 présente deux registres à décalage modifiés  $R_1$  et  $R_2$  (voir l'exercice du paragraphe 5.3.12), commandés par une variable auxiliaire  $S$ . Pour  $S = 1$ , ces registres ont le fonctionnement habituel et décalent l'information de droite à gauche à chaque montée du signal d'horloge, tandis que pour  $S = 0$  ils sont bloqués et conservent leur état interne inchangé ( $C_1^+, B_1^+, A_1^+ = C_1, B_1, A_1$  et  $C_2^+, B_2^+, A_2^+ = C_2, B_2, A_2$ ).

La fonction  $S$  est définie par l'équation :

$$S = (x_1 \oplus C_1) + (x_2 \oplus C_2) \quad (6.37)$$

et prend la valeur 1 seulement si l'état d'entrée  $x_1, x_2$  est différent de l'état  $C_1, C_2$ ; c'est dans cette situation seulement qu'un décalage des registres (d'un bit vers la gauche) est possible. Une succession d'états  $x_1, x_2$  quelconques est donc transformée en une succession d'états  $x_1, x_2$  distincts : toute répétition immédiate d'un même état d'entrée est éliminée.

A titre d'exemple, on peut appliquer à l'entrée des deux registres (préalablement remis à zéro) une partie de la séquence  $x_1, x_2$  de la figure 6.2; on obtient le chronogramme de la figure 6.58 où  $CK : 1, 2, \dots, 14$  représente quatorze instants d'horloge successifs et chaque flèche un décalage.

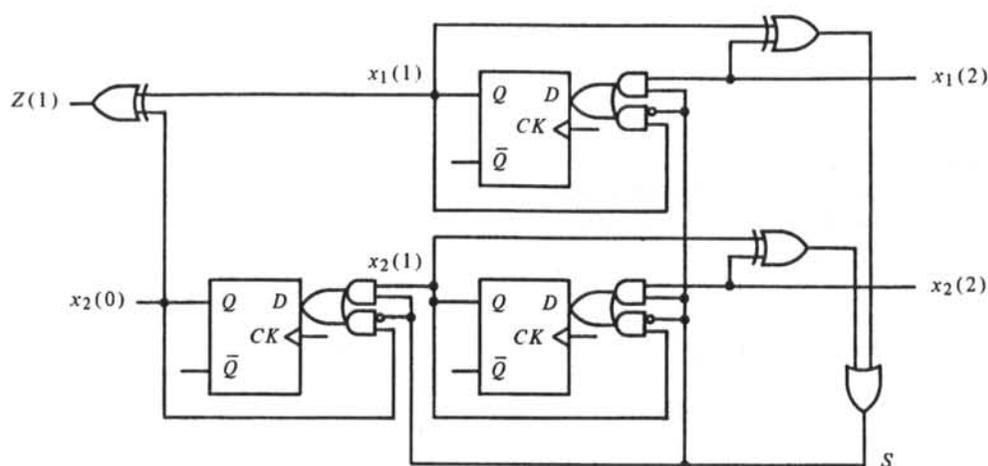


Fig. 6.56

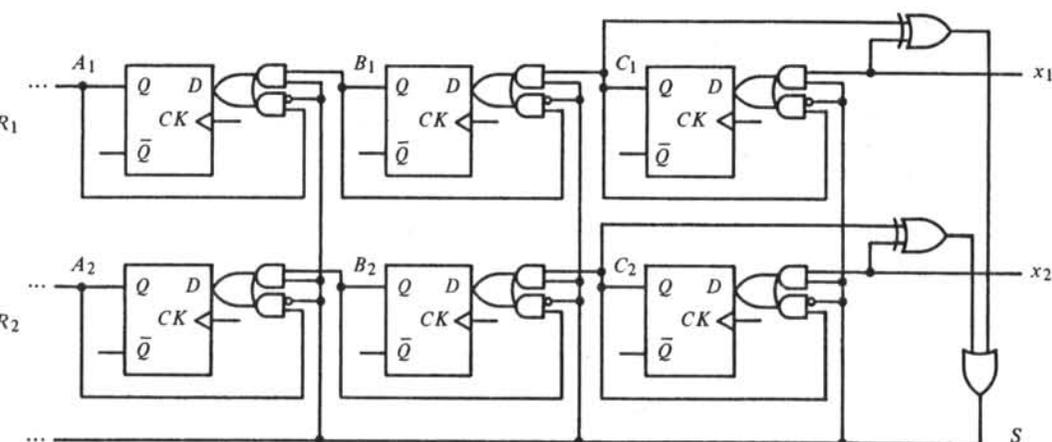


Fig. 6.57

### 6.5.13 Application : discriminateur du sens de rotation

En remplaçant les registres de la figure 6.54 par ceux du système de synchronisation de la figure 6.57 et en réalisant l'expression (6.36) de la fonction de sortie  $Z(1)$  on obtient le logigramme final de la figure 6.56; celle-ci peut être comparée aux figures 6.16, 6.17 (codage minimal), 6.22, 6.23 et 6.25 (codage 1 parmi 4).

### 6.5.14 Conclusion

La méthode du paragraphe 6.5.6 peut donc être étendue à plusieurs séquences d'entrée déterminantes produisant un certain état de sortie (le discriminateur en comporte quatre d'après la relation (6.35)) ainsi qu'aux systèmes séquentiels dont le com-

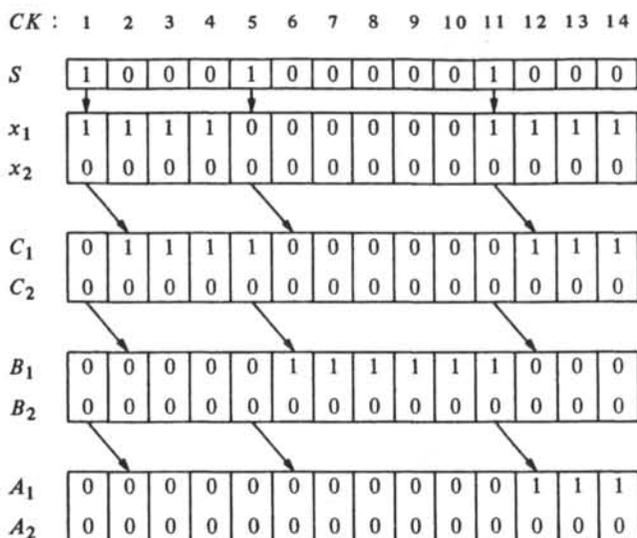


Fig. 6.58

portement est asynchrone : dans ce cas, les  $n$  registres de  $K - 1$  bits doivent être remplacés par les  $n$  registres modifiés à  $K$  bits du système de synchronisation (fig. 6.57).

Il existe des théories plus générales qui permettent de réaliser n'importe quel système séquentiel (et pas seulement ceux à mémoire finie) avec des registres à décalage : l'étude de ces méthodes sort du cadre de ce volume et fait l'objet des références [73] et [79] (pp. 321 - 343).

#### ■ 6.5.15 Exercice

Analyser le logigramme de la figure 6.56, comparer le graphe et la table d'états calculés à ceux des figures 6.4 et 6.5, mettre en évidence et justifier les différences. Corriger au besoin le chronogramme de la figure 6.2, préciser le cahier des charges du paragraphe 6.1.4, essayer de réduire la table d'états calculée pour obtenir enfin la table et le graphe réduits des figures 6.6 et 6.53.

Démontrer que le premier étage du système séquentiel de synchronisation (fig. 6.57) peut toujours être réalisé par les équations  $C_1^+ = x_1$  et  $C_2^+ = x_2$ .

#### ■ 6.5.16 Exercice

Un codage minimal de la table d'états du discriminateur (fig. 6.14) a produit les fonctions  $y_1^+ = D_1$  (6.10),  $y_2^+ = D_2$  (6.11) et  $Z$  (6.12). En s'inspirant de la méthode du paragraphe 6.5.5, est-il possible de réaliser la fonction  $Z(1)$  de ce discriminateur (§ 6.5.11) avec une structure itérative dont chaque sous-système serait défini par les relations (6.10), (6.11) et (6.12) ? Dans l'affirmative, tracer le logigramme d'une telle réalisation en précisant le choix des conditions initiales  $y_1(0)$  et  $y_2(0)$ .

**■ 6.5.17 Exercice**

Est-il possible de réaliser avec des registres à décalage modifiés selon la figure 6.57 les systèmes séquentiels des exercices précédents : passages à niveau (§ 6.1.19 et § 6.1.27), serrure électronique (§ 6.1.20), délai asynchrone (§ 6.1.22) et détecteur de séquences (§ 6.1.26) ? Dans l'affirmative, réaliser les structures parallèle et itérative du système combinatoire.

## MODÈLES ASYNCHRONES DES SYSTÈMES LOGIQUES

### 7.1 PRÉAMBULE

#### 7.1.1 Introduction

Dans ce qui précède on a étudié les systèmes combinatoires sans délai (décrits par leur modèle combinatoire : chapitres 1 et 2), les bascules bistables (décrites par leur modèle quasi-synchrone ou synchrone : chapitre 3), puis les assemblages de bascules et d'éléments combinatoires (compteurs, systèmes séquentiels quasi-synchrones et synchrones : chapitres 4, 5 et 6).

Le modèle logique asynchrone (§ 1.1.9 et sect. 7.1) nous permet d'aborder dans ce chapitre les problèmes non traités jusqu'ici, soit l'analyse des systèmes combinatoires avec délais (sect. 7.2), l'analyse des systèmes séquentiels quelconques, c'est-à-dire sans bascules (sect. 7.3), enfin l'analyse des bascules bistables (sect. 7.4) et des systèmes séquentiels synchrones ou quasi-synchrones (sect. 7.5) ne vérifiant pas le modèle quasi-synchrone du chapitre 3.

#### 7.1.2 Rappel : modèle logique asynchrone

On a montré dans la section 1.1 qu'en partant d'une grandeur  $x_3$  (fig. 1.2) qui varie de façon continue et qu'en admettant les trois hypothèses de la quantification, de l'élimination des transitoires et de l'égalisation des délais on obtient le modèle logique asynchrone du système concret étudié.

L'élément de délai (§ 1.1.13), caractérisé par un retard  $\Delta$ , est un système idéal dont l'existence est suggérée par le chronogramme de la figure 1.2 (variables  $Y$  et  $Z$ ). Nous allons préciser maintenant les caractéristiques de cet élément.

#### 7.1.3 Définition : élément de délai pur

Nous appellerons *élément de délai pur* (ou *délai pur*) le système logique idéal représenté par le schéma de la figure 7.1 et décrit par le chronogramme de la figure 7.3 : l'état de sortie  $y_p$  est égal à l'état d'entrée  $x$  retardé d'une valeur constante  $\Delta_p$  qui caractérise l'élément. On peut alors écrire l'équation :

$$y_p(t + \Delta_p) = y_p^+(t) = x(t) \quad (7.1)$$

qui est équivalente à :

$$y_p(t) = x(t - \Delta_p) \quad (7.2)$$

où  $t$ , le temps, est un paramètre qui varie de façon continue. Le délai pur retarde donc d'une durée  $\Delta_p$  tout signal d'entrée  $x$ .

Lorsque la relation  $x(t) = y_p(t)$  est vérifiée ( $\Delta_p = 0$ ), le délai pur admet un régime permanent (§ 3.1.6); mais lors de brèves variations de l'entrée, c'est-à-dire lorsque  $\Delta t < \Delta_p$  (fig. 7.3), on constate que l'équation  $x(t) = y_p(t)$  est validée pendant une durée  $\Delta_p - \Delta t$  alors que l'impulsion de  $x(t)$  n'est pas encore apparue à la sortie  $y_p$ : le régime est donc transitoire.



Fig. 7.1



Fig. 7.2

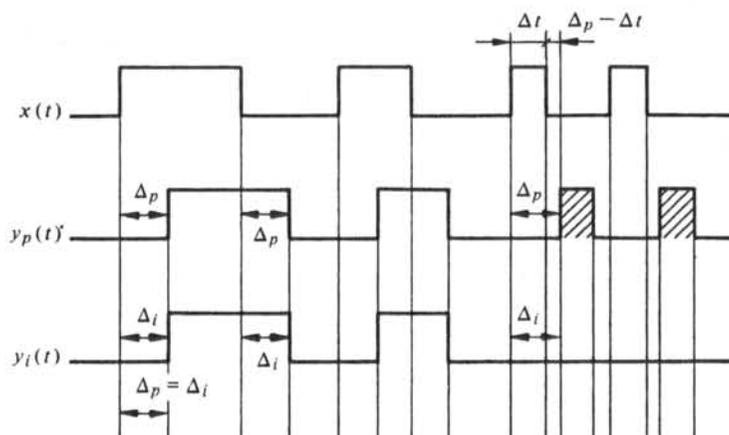


Fig. 7.3

#### 7.1.4 Définition : élément de délai inertiel

L'observation des systèmes concrets montre que ceux-ci se comportent différemment du délai pur si la durée des variations de l'entrée  $x$  est voisine ou inférieure à celle de la constante  $\Delta_p$ . On appellera *élément de délai inertiel* (ou *délai inertiel*) le système logique idéal représenté par le schéma de la figure 7.2 et décrit par le chronogramme de la figure 7.3; si  $\Delta t$  est la durée d'une variation de l'entrée  $x$  et si  $\Delta_i$  est une valeur constante qui caractérise l'élément, alors :

$$\begin{aligned} \text{pour } \Delta t < \Delta_i, & \quad y_i(t) = \text{constante}; \\ \text{pour } \Delta t \geq \Delta_i, & \quad y_i(t + \Delta_i) = y_i^+(t) = x(t) \end{aligned} \quad (7.3)$$

Le délai inertiel retarde donc d'une valeur  $\Delta_i$  tout signal d'entrée  $x$  dont la variation a une durée supérieure ou égale à  $\Delta_i$ , et conserve l'état de sortie  $y_i$  inchangé dans le cas contraire. Le délai inertiel est un délai pur (fonction de retard) pour des variations lentes de l'entrée, mais supprime les variations rapides de celle-ci (fonction de filtrage).

Contrairement au délai pur, tout régime caractérisé par la relation  $x(t) = y_i(t)$  est permanent (§ 3.1.6 et fig. 7.3); l'équation  $x(t) = y_i(t)$  définit donc le régime transitoire du délai inertiel.

### 7.1.5 Définitions : système logique asynchrone

On a déjà défini celui-ci (§ 1.1.14) : c'est l'assemblage d'un système combinatoire (sans délai) et d'un ou plusieurs éléments de délai (purs ou inertiels)  $\Delta_1, \Delta_2, \dots, \Delta_m$  (fig. 7.4). Par analogie avec les systèmes séquentiels quasi-synchrones (§ 5.1.1) nous dirons que :

- les  $n$  variables indépendantes  $x_1, x_2, \dots, x_n$  sont les variables d'entrée (ou primaires); toute combinaison de leurs valeurs est un état d'entrée (ou primaire);
- les sorties des  $m$  éléments de délai  $y_1, y_2, \dots, y_m$  sont les variables internes (ou secondaires); toute combinaison de leurs valeurs est un état (ou état interne ou état secondaire);
- toute combinaison des valeurs des  $n$  variables d'entrée et des  $m$  variables internes est un état total;
- toute combinaison des valeurs des  $r$  sorties  $Z_1, Z_2, \dots, Z_r$  est un état de sortie.

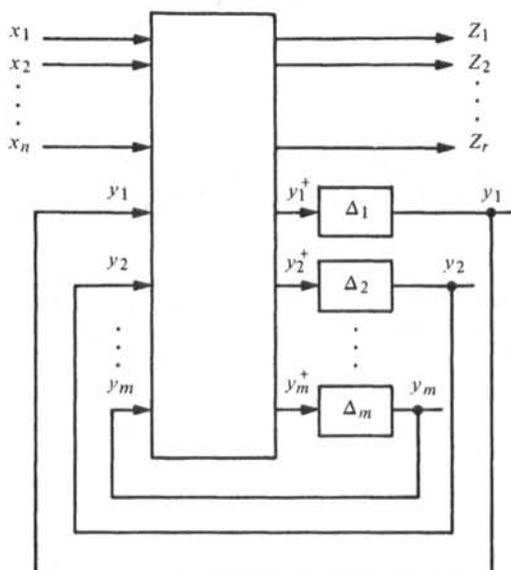


Fig. 7.4

Les entrées des  $m$  éléments de délai  $y_1^+, y_2^+, \dots, y_m^+$  sont les *excitations* du système; toute combinaison de leurs valeurs est un *état d'excitation*.

Enfin, tout état total vérifiant la relation (5.9) :

$$(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m) = (x_1, x_2, \dots, x_n; y_1^+, y_2^+, \dots, y_m^+)$$

est stable; tout état total qui n'est pas stable est instable (§ 5.2.7 et 5.2.8).

### 7.1.6 Propriété

Dans le cas où l'on dispose de délais inertiels (§ 7.1.4), on sait que les relations  $x \equiv y_i^+ = y_i$  et  $x \equiv y_i^+ = \bar{y}_i$  définissent les régimes permanent et transitoire de ces élé-

ments. Nous pouvons alors dire qu'un état total stable est synonyme de régime permanent et qu'un état total instable est synonyme de régime transitoire (§ 7.1.5).

Cette propriété, qui n'est pas vérifiée pour les délais purs (§ 7.1.3), facilitera l'analyse des systèmes asynchrones dont les délais sont inertiels.

### 7.1.7 Bibliographie

Une étude plus approfondie des délais pur et inertiel fait l'objet des références [102] (pp. 118 - 121), [82] (pp. 141 - 145), [84] (pp. 164 - 167, 182 - 191), [81] (pp. 265 - 267) et [103] (pp. 108 - 111).

Il existe d'autres modèles de délais, en général plus complexes, qui ont été décrits dans la littérature : ces délais comportent trois niveaux de quantification au lieu de deux [101], [10] (pp. 429 - 435) ou des temps de montée distincts des temps de descente [103] (pp. 108 - 111).

### 7.1.8 Exercice

On demande de réaliser avec un nombre minimal d'éléments de délai (pur et/ou inertiel) les assemblages des figures 7.5 ( $\Delta_p$  : délai pur) et 7.6 ( $\Delta_i$  : délai inertiel). En admettant que les  $n$  délais inertiels  $\Delta_{i1}, \Delta_{i2}, \dots, \Delta_{in}$  (fig. 7.6) ont la même valeur  $\Delta_i$  avec  $n \Delta_i = \Delta T$ , que se passe-t-il lorsque  $n$  tend vers l'infini avec  $\Delta T =$  constante ?



Fig. 7.5

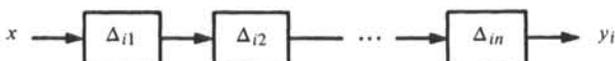


Fig. 7.6

### 7.1.9 Exercice

Réaliser un élément de délai caractérisé par un retard  $R$  et par la suppression de toutes les variations du signal d'entrée dont la durée est inférieure à  $0,2 \cdot R$ . Donner un logigramme comportant un nombre minimal d'éléments de délai pur et/ou inertiel. Est-il possible de tracer un logigramme équivalent avec des délais inertiels uniquement ?

## 7.2 SYSTÈMES COMBINATOIRES

### 7.2.1 Exemple

La figure 7.7 présente un système concret qui est un assemblage d'opérateurs NON, ET, OU sans rétroaction. Ces opérateurs comportent chacun un certain retard (sect. 1.1) qui est généralement sous-entendu dans le logigramme.

### 7.2.2 Modèle combinatoire

Ce modèle est dépourvu de délais (§ 1.1.10) : il se confond donc avec le logigramme de la figure 7.7 et il est décrit par l'équation suivante :

$$Z = ac + b\bar{c} \quad (7.4)$$

qui peut être représentée dans une table de vérité ou dans une table de Karnaugh (fig. 7.8). Celle-ci est unique : elle définit la fonction logique du système combinatoire et détermine son régime permanent (§ 3.1.6).

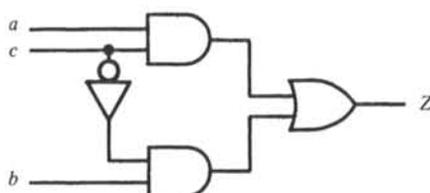


Fig. 7.7

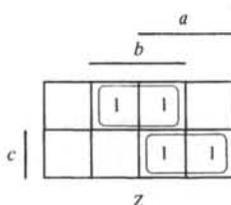


Fig. 7.8

### 7.2.3 Modèle asynchrone

Un tel modèle (fig. 7.9) est obtenu par exemple en insérant à la sortie de chaque opérateur NON, ET, OU de la figure 7.7 un élément de délai; l'analyse de ce modèle nous permettra de connaître le régime transitoire du système (§ 3.1.6), c'est-à-dire les variations éventuelles de l'état de sortie Z lors d'un changement de l'état d'entrée a, b, c.

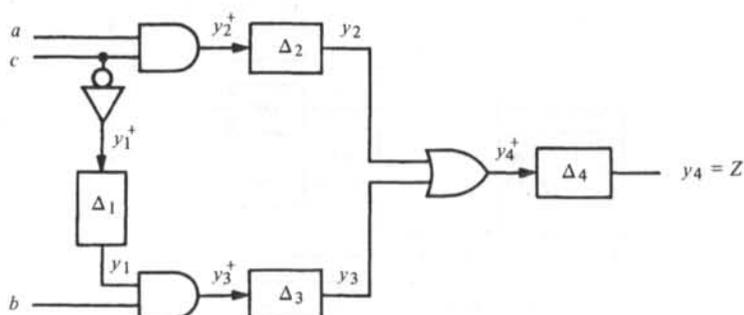


Fig. 7.9

### 7.2.4 Hypothèses

Il existe de nombreux modèles asynchrones décrivant un système logique donné. Nous dirons que nous avons déterminé *un modèle asynchrone* de ce système lorsque nous connaissons *le nombre des éléments de délai et leur place dans le logigramme*.

De plus, nous admettrons dans ce chapitre que *les éléments de délai d'un modèle asynchrone donné sont toujours inertiels* (§ 7.1.4) *et que leurs valeurs  $\Delta_1, \Delta_2, \dots, \Delta_m$  ne sont jamais nulles, ni infinies*.

### 7.2.5 Application

La figure 7.9 représente donc un modèle asynchrone du système concret de la figure 7.7. Les quatre délais  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  sont inertiels d'après l'hypothèse du paragraphe 7.2.4.

D'après cette même hypothèse, nous savons que si un ou plusieurs des délais  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  sont nuls nous obtenons un modèle asynchrone différent. Si les quatre délais sont nuls, on retrouve le modèle combinatoire; celui-ci constitue un cas particulier des modèles asynchrones, caractérisé par les relations :

$$y_1^+ = y_1 ; y_2^+ = y_2 ; y_3^+ = y_3 ; y_4^+ = y_4 \quad (7.5)$$

qui décrivent le régime permanent du système.

### 7.2.6 Modes de représentation : équations et table d'états

Le modèle asynchrone de la figure 7.9 peut être mis sous la forme d'un système logique asynchrone (fig. 7.4). Les excitations  $y_1^+, y_2^+, y_3^+$  et  $y_4^+$  des éléments de délai sont exprimées algébriquement :

$$y_1^+ = \bar{c} ; y_2^+ = a c ; y_3^+ = b y_1 ; y_4^+ = y_2 + y_3 \quad (7.6)$$

tandis que la sortie  $Z$  se confond avec la variable  $y_4$  (fig. 7.10).

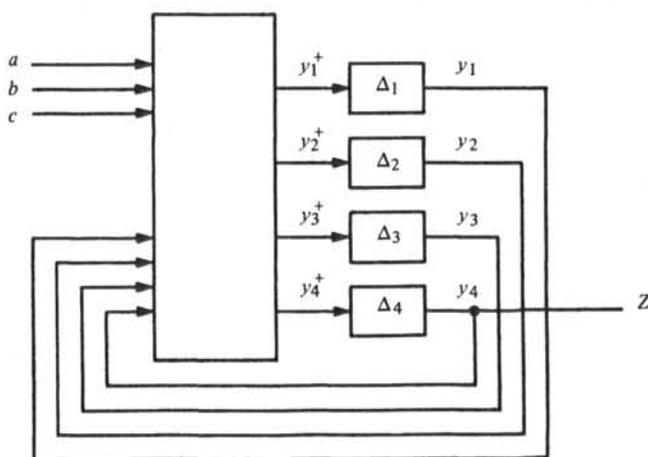


Fig. 7.10

Chaque élément de délai inertiel est défini par une expression semblable à (7.3) avec  $\Delta_i = \Delta_1, \Delta_2, \Delta_3$  ou  $\Delta_4$ . Le but de l'analyse est le calcul du régime transitoire pour toutes les valeurs possibles de  $\Delta_1, \Delta_2, \Delta_3$  et  $\Delta_4$  qui constituent ainsi les *paramètres* du modèle asynchrone; ces paramètres varient de façon continue, mais ne sont ni nuls, ni infinis (§ 7.2.4).

$y_1^+ y_2^+ y_3^+ y_4^+$		$a = 0$				$a = 1$			
		$bc$				$bc$			
$y_1$	$y_2$	00	01	11	10	00	01	11	10
00	00	1000	0000	0000	1000	1000	0100	0100	1000
00	01	1000	0000	0000	1000	1000	0100	0100	1000
00	11	1001	0001	0001	1001	1001	0101	0101	1001
00	10	1001	0001	0001	1001	1001	0101	0101	1001
01	00	1001	0001	0001	1001	1001	0101	0101	1001
01	01	1001	0001	0001	1001	1001	0101	0101	1001
01	11	1001	0001	0001	1001	1001	0101	0101	1001
01	10	1001	0001	0001	1001	1001	0101	0101	1001
11	00	1001	0001	0011	1011	1001	0101	0111	1011
11	01	1001	0001	0011	1011	1001	0101	0111	1011
11	11	1001	0001	0011	1011	1001	0101	0111	1011
11	10	1001	0001	0011	1011	1001	0101	0111	1011
10	00	1000	0000	0010	1010	1000	0100	0110	1010
10	01	1000	0000	0010	1010	1000	0100	0110	1010
10	11	1001	0001	0011	1011	1001	0101	0111	1011
10	10	1001	0001	0011	1011	1001	0101	0111	1011

$y_1 y_2 y_3 y_4$   
(Z)

Fig. 7.11

Les fonctions d'excitation  $y_1^+, y_2^+, y_3^+, y_4^+$  (7.6) sont alors représentées dans une table de Karnaugh (fig. 7.11) qui, par analogie avec les systèmes séquentiels synchrones (§ 5.2.4), est appelée la table d'états; on y a encadré les états totaux stables (§ 7.1.5): ceux-ci décrivent le régime permanent, car ils vérifient les relations (7.5).

Mais les propriétés des délais inertiels (contrairement à celles des bascules bistables) ne nous permettent pas encore de déterminer le régime transitoire (c'est-à-dire la succession des états internes) pour n'importe quelle valeur des paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ . La table d'états du modèle asynchrone est un mode de représentation incomplet: nous allons décrire par conséquent d'autres modes, tels que le chronogramme et le graphe des états.

### 7.2.7 Modes de représentation : chronogrammes

Pour limiter notre étude, on cherche à déterminer le régime transitoire à partir d'un régime permanent ( $a, b, c = 111$ ) et d'une variation de la variable  $c$  ( $c = 1 \rightarrow 0$ ). On dira qu'on a appliqué au système la séquence d'entrée :

$$a, b, c = 111 \rightarrow 110 \tag{7.7}$$

Les variables  $a$  et  $b$  étant constantes, on peut trouver les formes particulières des fonctions d'excitation (7.6):

$$y_1^+ = \bar{c} ; y_2^+ = c ; y_3^+ = y_1 ; y_4^+ = y_2 + y_3 \quad (7.8)$$

Mais le tracé d'un chronogramme nécessite aussi la connaissance des paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ . Un choix arbitraire de valeurs est représenté dans la figure 7.12, ce qui nous permet d'établir le chronogramme en procédant de la façon suivante :

- on a admis au départ un régime permanent avec  $a, b, c = 111$ ; pour tout instant  $t < t_0$ , on tire de (7.8) et (7.5) les relations :

$$y_1^+ = y_1 = 0 ; y_2^+ = y_2 = 1 ; y_3^+ = y_3 = 0 ; y_4^+ = y_4 = 1 \quad (7.9)$$

- au temps  $t = t_0$ , on impose la variation  $c = 1 \rightarrow 0$ ; on déduit de (7.8) les changements instantanés de  $y_1^+ = \bar{c} = 1$  et  $y_2^+ = c = 0$ ;
- au temps  $t_1 = t_0 + \Delta_2$  la variable  $y_2$  prend la nouvelle valeur  $y_2 = y_2^+ = 0$ ;
- cette variation de  $y_2$  entraîne une variation simultanée de  $y_4^+ = y_2 + y_3$  (7.8);

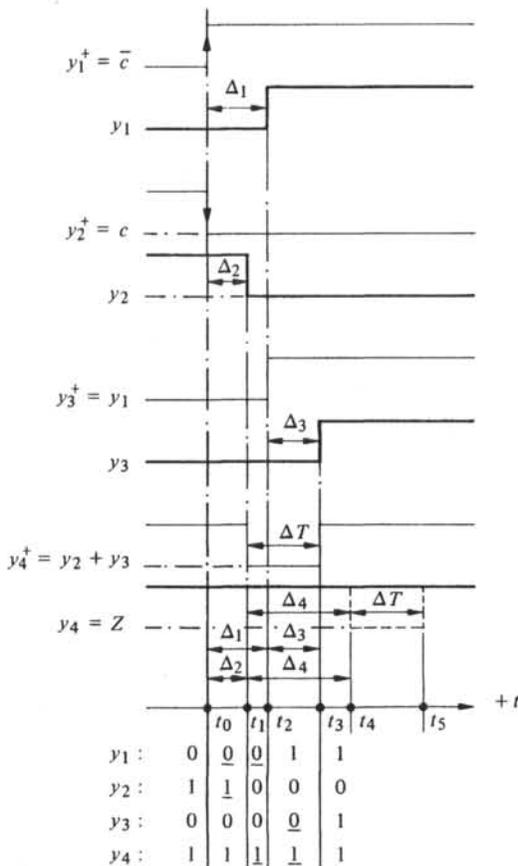


Fig. 7.12

- le calcul se poursuit selon le même principe : chaque variation d'une excitation ( $y_1^+, y_2^+, y_3^+$  ou  $y_4^+$ ) entraîne après un certain délai ( $\Delta_1, \Delta_2, \Delta_3$  ou  $\Delta_4$ ) un changement de la variable interne ( $y_1, y_2, y_3$  ou  $y_4$ ), et chaque variation de celle-ci implique un nouveau calcul des excitations (7.8);
- la définition du délai inertiel (§ 7.1.4) impose enfin la suppression des variations de sortie  $y_i$  dont la durée est inférieure au retard  $\Delta_i$  de l'élément; le choix des paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  dans la figure 7.12 montre que l'excitation  $y_4^+$  prend la valeur 0 pendant une durée  $\Delta T = \Delta_1 + \Delta_3 - \Delta_2$  qui est inférieure à  $\Delta_4$ , de telle sorte que la variable  $y_4$  reste inchangée (on a représenté en trait discontinu la variation de  $y_4$  qui serait obtenue si  $\Delta_4$  était un délai pur).

En résumé, les valeurs relatives suivantes des paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  :

$$\Delta T = \Delta_1 + \Delta_3 - \Delta_2 < \Delta_4 \quad \text{ou} \quad \Delta_2 + \Delta_4 > \Delta_1 + \Delta_3 \quad (7.10)$$

ont produit un état de sortie  $y_4 = Z$  invariant ( $Z = 1$ ) durant le régime transitoire, c'est-à-dire pendant l'intervalle où  $t_0 \leq t \leq t_3$ .

En choisissant une nouvelle valeur de  $\Delta_1$  ( $\Delta_2, \Delta_3$  et  $\Delta_4$  étant inchangés) qui

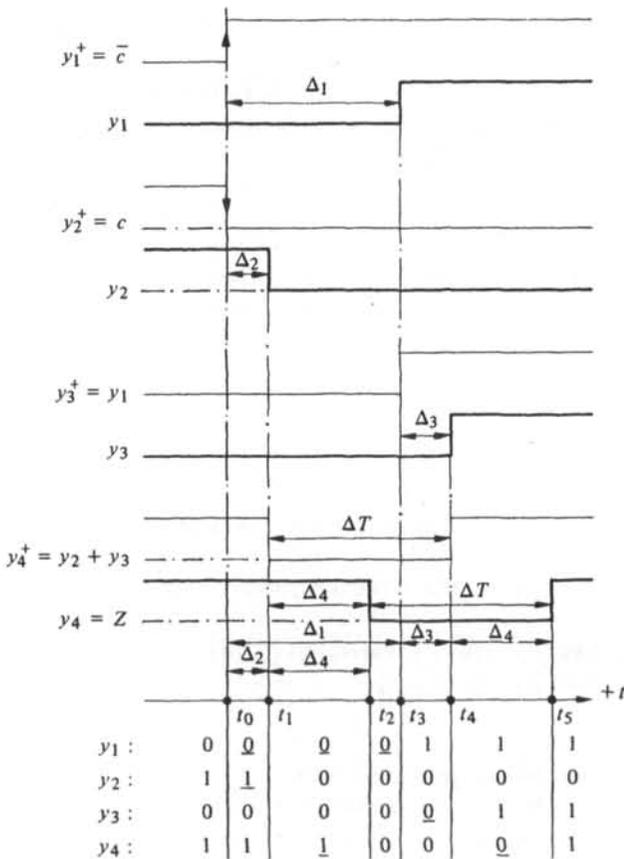


Fig. 7.13

vérifie l'inéquation :

$$\Delta T = \Delta_1 + \Delta_3 - \Delta_2 > \Delta_4 \quad \text{ou} \quad \Delta_2 + \Delta_4 < \Delta_1 + \Delta_3 \quad (7.11)$$

on trace un second chronogramme (fig. 7.13). Dans celui-ci on constate que l'état de sortie  $y_4 = Z$  a varié durant le régime transitoire (c'est-à-dire dans l'intervalle où  $t_0 \leq t \leq t_5$ ) selon la séquence suivante :

$$Z = 1 \rightarrow 0 \rightarrow 1 \quad (7.12)$$

### 7.2.8 Définitions : course

La séquence d'entrée (7.7) entraîne la variation simultanée des excitations  $y_1^+$  et  $y_2^+$ ; on constate alors, dans la figure 7.12 par exemple, que pour tout instant  $t$  tel que  $t_0 < t < t_1$  l'état d'excitation est différent de l'état interne :

$$y_1^+, y_2^+, y_3^+, y_4^+ = 1001 \quad ; \quad y_1, y_2, y_3, y_4 = \underline{0}\underline{1}01 \quad (7.13)$$

On remarque en effet que  $y_1^+ \neq y_1$  et que  $y_2^+ \neq y_2$ ; on dit que les variables internes  $y_1$  et  $y_2$  sont *instables* et on les souligne dans l'expression (7.13).

Par définition, nous dirons que tout état interne dont deux variables au moins sont instables définit une *course* entre ces variables.

Au bas des figures 7.12 et 7.13 on a rappelé la succession des états internes en soulignant les variables instables; chaque état souligné au moins deux fois définit donc une course.

### 7.2.9 Propriété

Une course entre plusieurs variables internes  $y_1, y_2, \dots, y_k$  produit des régimes transitoires distincts qui dépendent des valeurs des paramètres  $\Delta_1, \Delta_2, \dots, \Delta_k$ .

Deux choix arbitraires des valeurs  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  nous ont permis de tracer les chronogrammes précédents : dans le premier (fig. 7.12), on observe trois courses successives produisant la séquence suivante des états internes :

$$y_1, y_2, y_3, y_4 = \underline{0}\underline{1}01 \rightarrow \underline{0}\underline{0}0\underline{1} \rightarrow 1\underline{0}\underline{0}\underline{1} \rightarrow 1011 \quad (7.14)$$

tandis que dans le second (fig. 7.13) deux courses seulement ont produit la séquence :

$$y_1, y_2, y_3, y_4 = \underline{0}\underline{1}01 \rightarrow \underline{0}\underline{0}0\underline{1} \rightarrow \underline{0}\underline{0}0\underline{0} \rightarrow 1\underline{0}\underline{0}\underline{0} \rightarrow 101\underline{0} \rightarrow 1011 \quad (7.15)$$

Les régimes transitoires décrits par les expressions (7.14) et (7.15) sont distincts et dépendent des valeurs de  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ .

### 7.2.10 Mode de représentation : graphe des états

Toute course entre les variables internes d'un système asynchrone entraîne l'existence de plusieurs régimes transitoires, donc de plusieurs chronogrammes. Nous cherchons alors un moyen de représenter synoptiquement tous ces régimes : c'est le graphe des états dont la construction et l'interprétation font l'objet des paragraphes suivants.

■ 7.2.11 Méthode

- On part du régime permanent défini par l'état d'entrée  $a, b, c = 111$ , donc par l'état total stable :

$$a, b, c ; y_1, y_2, y_3, y_4 = 111, 0101 \tag{7.16}$$

qui est encerclé dans la table de la figure 7.11.

- Au temps  $t = t_0$ , on impose la variation  $c = 1 \rightarrow 0$ ; on est instantanément conduit dans l'état total instable :

$$a, b, c ; y_1, y_2, y_3, y_4 = 110, 0101 \tag{7.17}$$

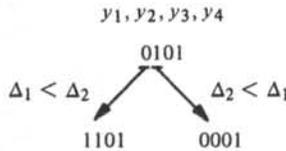
dont l'état d'excitation est alors (fig. 7.11) :

$$y_1^+, y_2^+, y_3^+, y_4^+ = 1001 \tag{7.18}$$

La comparaison de l'état interne (7.17) avec l'état d'excitation (7.18) montre que les deux variables  $y_1$  et  $y_2$  sont instables : il y a donc une course et nous écrivons, conformément à la notation du paragraphe 7.2.8 :

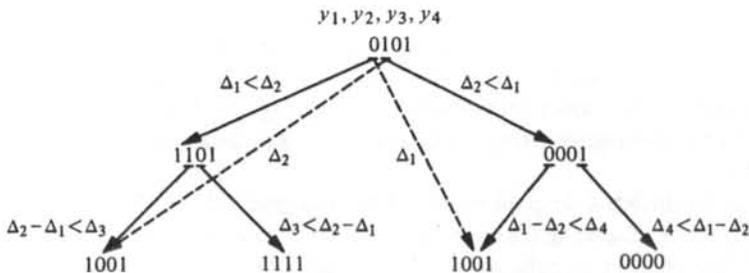
$$y_1, y_2, y_3, y_4 = \underline{0}101$$

- Selon les valeurs relatives des délais  $\Delta_1$  et  $\Delta_2$ , on atteint deux états internes distincts qui sont représentés par le schéma suivant :



L'égalité stricte des délais ( $\Delta_1 = \Delta_2$ ) constitue un cas très particulier qui n'est généralement pas représenté.

- A partir de chacun des états 1101 et 0001, on répète la même procédure : l'état d'entrée  $a, b, c = 110$  restant inchangé, on constate dans la figure 7.11 que les états internes 1101 et 0001 produisent respectivement les états d'excitation 1011 et 1000. Dans les deux cas il apparaît une course entre deux variables ( $y_2$  et  $y_3$ , respectivement  $y_1$  et  $y_4$ ):



Les flèches en trait discontinu indiquent la durée de certaines variations et facilitent la détermination des inéquations entre paramètres.

- En poursuivant systématiquement la méthode on obtient à l'extrémité de chaque branche du graphe le même état total stable (fig. 7.14) :

$$a, b, c ; y_1, y_2, y_3, y_4 = 110, 1011 \quad (7.19)$$

Celui-ci est représenté par une flèche bouclée sur elle-même; la construction du graphe, qui comprend six branches (No 1 à 6), est alors terminée.

Un graphe simplifié (fig. 7.15) peut être tiré du précédent (fig. 7.14) si l'on représente une seule fois chaque état.

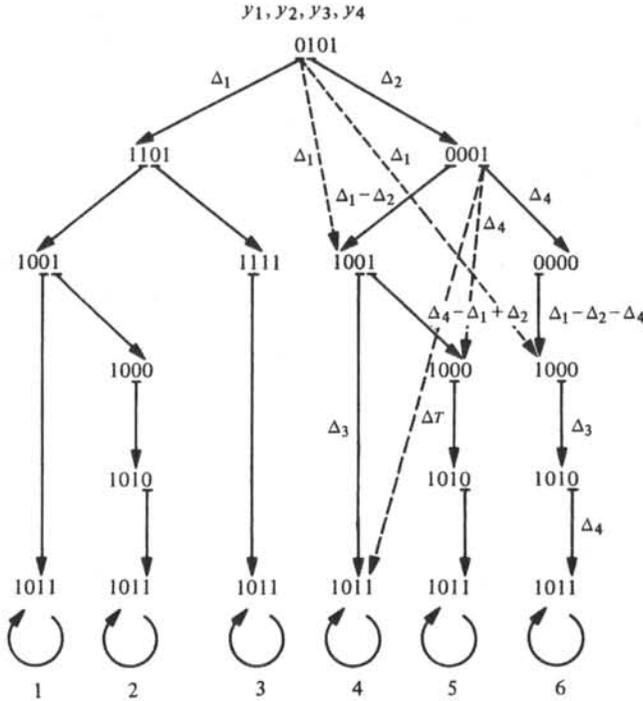


Fig. 7.14

### ■ 7.2.12 Description

La détermination des chronogrammes (fig. 7.12 et 7.13) et celle du graphe des états (fig. 7.14) découlent des mêmes équations (7.8); tandis que chaque chronogramme dépend d'un choix des valeurs de  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ , le graphe représente synoptiquement tous les régimes transitoires définis par des valeurs quelconques de  $\Delta_1, \Delta_2, \Delta_3$  et  $\Delta_4$ .

La branche No 4 du graphe (fig. 7.14) est équivalente au chronogramme de la figure 7.12 : on y observe la même séquence des états internes et les mêmes durées de chacun d'eux. L'effet du délai inertiel  $\Delta_4$  est également visible : la descente de  $y_4^+$  au temps  $t_1$  est représentée dans le graphe par l'état 0001 tandis que l'état total stable atteint au temps  $t_3$  est décrit dans le graphe par l'état 1011; la valeur  $y_4^+ = 0$  a persisté pendant une durée  $t_3 - t_1 = \Delta T$  inférieure au retard  $\Delta_4$  du délai inertiel : la sortie de

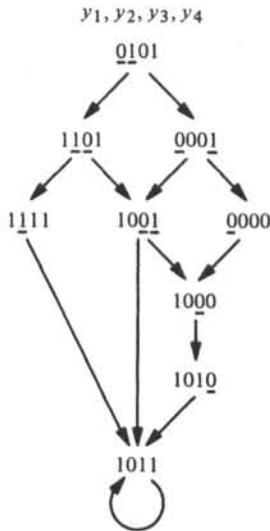


Fig. 7.15

celui-ci conserve donc sa valeur inchangée  $y_4 = 1$ . Les valeurs relatives des paramètres  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$  et  $\Delta_4$  peuvent être déterminées à leur tour :

- pour passer de  $0101$  à  $0001$ ,  $y_2$  doit gagner la course engagée avec  $y_1$  :

$$\Delta_2 < \Delta_1 \quad (7.20)$$

- de  $0001$  à  $1001$ ,  $y_1$  doit gagner la course avec  $y_4$  :

$$\Delta_1 - \Delta_2 < \Delta_4 \quad (7.21)$$

- pour passer de  $1001$  à  $1011$ ,  $y_3$  doit gagner la course avec  $y_4$  :

$$\Delta_3 < \Delta_4 - \Delta_1 + \Delta_2 \quad (7.22)$$

Les relations (7.20) à (7.22) peuvent s'exprimer sous la forme d'une inéquation triple qui inclut l'expression précédente (7.10) :

$$\Delta_2 + \Delta_4 > \Delta_1 + \Delta_3 > \Delta_1 > \Delta_2 \quad (7.23)$$

La branche No 6 du graphe des états (fig. 7.14) est équivalente au chronogramme de la figure 7.13 : on y observe notamment la même séquence de l'état de sortie :

$Z = y_4 = 1 \rightarrow 0 \rightarrow 1$ . Les paramètres  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$ ,  $\Delta_4$  vérifient dans ce cas l'inéquation :

$$\Delta_1 > \Delta_2 + \Delta_4 \quad (7.24)$$

Le graphe des états est bien adapté aux délais inertiels qui jouissent de la propriété du paragraphe 7.1.6 : sitôt qu'un état total stable est rencontré dans une branche (par

exemple  $y_1, y_2, y_3, y_4 = 1011$  dans la figure 7.14), le régime permanent commence et le régime transitoire est terminé. Cette propriété n'est pas forcément vérifiée pour les délais purs, de telle sorte que la représentation de leur comportement dans un graphe est malaisée.

### 7.2.13 Conclusion

L'analyse d'un système logique asynchrone (fig. 7.10) détermine donc une famille de régimes transitoires produits par toutes les valeurs possibles des paramètres de ce système ( $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ ).

La table d'états (fig. 7.11) est utile pour établir le graphe (fig. 7.14); celui-ci représente, pour une variation donnée de l'état d'entrée, la famille de tous les régimes transitoires possibles (six pour la séquence  $a, b, c = 111 \rightarrow 110$ ) et englobe ainsi le tracé de plusieurs chronogrammes.

Dans le cas particulier où il n'existe aucune course entre les variables internes, le graphe comporte une seule branche : celle-ci peut être directement lue dans la table d'états qui constitue alors un mode de représentation suffisant (séquence d'entrée  $a, b, c = 011 \rightarrow 010$  par exemple).

### 7.2.14 Définition : aléa statique

Les branches No 2, 5 et 6 du graphe des états (fig. 7.14) représentent la séquence suivante de l'état de sortie  $y_4 = Z$  :

$$Z = 1 \rightarrow 0 \rightarrow 1 \quad (7.25)$$

Lorsque la sortie d'un système logique présente un tel régime transitoire (ou la séquence complémentaire  $0 \rightarrow 1 \rightarrow 0$ ) pour certaines valeurs des paramètres ( $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ ) du modèle asynchrone, on dit qu'elle admet un *aléa statique*.

### 7.2.15 Propriété

L'aléa statique (7.25) est produit par la séquence d'entrée  $a, b, c = 111 \rightarrow 110$ ; celle-ci est représentée dans la table de Karnaugh de  $Z$  (fig. 7.16) par une flèche reliant les deux impliquants premiers de cette fonction ( $ac$  et  $b\bar{c}$ ). Sans modifier le modèle combinatoire de  $Z$  (c'est-à-dire sans changer la table de vérité de la fonction), on peut rajouter l'impliquant premier  $ab$  (en trait discontinu dans la figure 7.16) dans l'expression algébrique (7.4) qui devient alors :

$$Z = ac + b\bar{c} + ab \quad (7.26)$$

Une réalisation possible de (7.26) est donnée par la figure 7.17.

La présence de l'impliquant  $ab$  garantit la suppression de l'aléa statique (7.25) pour la séquence d'entrée  $a, b, c = 111 \rightarrow 110$ . On vérifie en effet que pour celle-ci :

$$Z(a, b, c) = Z(1, 1, c) = c + \bar{c} + 1 = 1 \quad (7.27)$$

La valeur de  $Z$  est donc une constante qui est indépendante de la variable  $c$ .

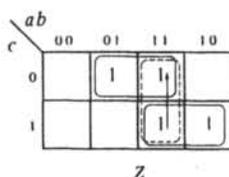


Fig. 7.16

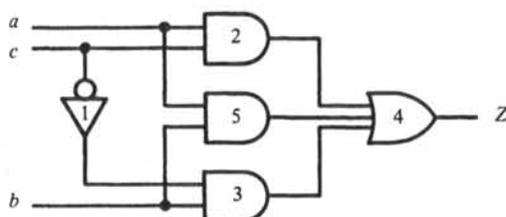


Fig. 7.17

On peut démontrer que tout aléa statique d'une fonction  $Z$  est généralement éliminé :

- s'il est provoqué par la variation d'une seule variable d'entrée;
- si la fonction est réalisée par un polynôme dont les monômes sont tous les impliquants premiers de celle-ci (essentiels, non essentiels et superflus, selon les définitions du paragraphe 2.2.22) [10] (pp. 394 - 398), [18] (pp. 203 - 204), [24] (pp. 170 - 173, 560 - 564).

### 7.2.16 Conclusion générale

On a étudié dans cette section un modèle asynchrone du système concret de la figure 7.7. Le choix de ce modèle dépend de la réalisation matérielle (électronique, hydraulique, pneumatique, etc...) des opérateurs logiques et découle de la connaissance des modèles physiques correspondants.

Un système concret électronique, par exemple, est constitué d'opérateurs reliés par des connexions; on constate que celles-ci présentent des délais purs (qui peuvent être négligés pour des signaux de fréquence moyenne ou basse), tandis que les opérateurs eux-mêmes sont caractérisés par des délais inertiels. Si des transistors bipolaires ou MOS (Metal Oxide Semiconductor) réalisent de simples portes (NON, NAND, NOR, etc...), on admet généralement qu'un seul délai inertiel, placé à la sortie de l'élément combinatoire idéal, constitue un modèle asynchrone acceptable [24] (pp. 552 - 553).

### 7.2.17 Bibliographie

L'analyse des systèmes logiques asynchrones et l'emploi du graphe des états sont notamment traités dans les références [84] (pp. 167 - 193), [24] (pp. 551 - 568) et [82] (pp. 192 - 244). En plus de l'aléa statique, certains auteurs ont défini l'*aléa dynamique* qui fait l'objet des références [10] (pp. 397 - 398), [18] (pp. 205 - 207), [24]

(pp. 565 - 568) et de l'exercice du paragraphe 7.2.23. Une théorie générale des aléas est présentée dans [81] et [104] à l'aide du calcul différentiel booléen. On trouvera enfin dans [80] une analyse des modèles asynchrones avec délais purs.

### 7.2.18 Exercice

Démontrer qu'en régime permanent (7.5) les équations du modèle asynchrone (7.6) sont équivalentes à l'expression (7.4) du modèle combinatoire. Etablir ensuite la table d'états décrite par les relations (7.8).

### 7.2.19 Exercice

Déterminer pour chacune des branches No 1, 2, 3 et 5 du graphe des états (fig. 7.14) une inéquation entre paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  telle que (7.23) ou (7.24). Exprimer par une inéquation semblable la présence ou l'absence de l'aléa statique de  $Z$  (§ 7.2.14). Compléter enfin le graphe (fig. 7.14) en admettant pour chaque course l'égalité des délais (ce cas particulier a été négligé dans le paragraphe 7.2.11).

### 7.2.20 Exercice

On applique la séquence d'entrée  $a, b, c = 011 \rightarrow 010$  au système asynchrone de la figure 7.9. Démontrer l'absence de toute course, tracer le graphe et vérifier que la table d'états de la figure 7.11 est un mode de représentation suffisant (§ 7.2.13). Combien de séquences d'entrée distinctes (successions de deux états  $a, b, c$ ) produisent-elles un régime transitoire sans course ?

### ■ 7.2.21 Exercice

Un modèle asynchrone du système de la figure 7.17 est obtenu en insérant à la sortie de chaque opérateur un délai ( $\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5$ ). Etablir les équations et la table d'états, vérifier la suppression de l'aléa statique (7.25) pour la séquence d'entrée  $a, b, c = 111 \rightarrow 110$ , puis déterminer les séquences d'entrée produisant une course. Tracer le graphe des états pour l'une d'elles. Existe-t-il un aléa statique de la sortie  $Z$  provoqué par la variation simultanée de plusieurs variables d'entrée ?

### 7.2.22 Exercice

Un système logique asynchrone réalisant (en régime permanent) la fonction  $Z = a \oplus b$  est donné par la figure 7.18. Etablir les équations et la table d'états, puis l'ensemble des graphes pour n'importe quelle variation de l'état d'entrée. Relever les séquences de l'état de sortie  $Z$  et contrôler l'existence éventuelle d'aléas statiques.

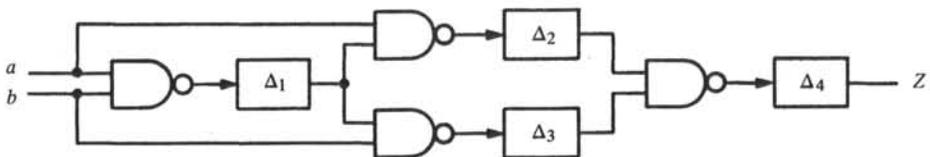


Fig. 7.18

■ 7.2.23 Exercice

On applique la séquence d'entrée  $x_1, x_2, x_3, x_4 = 1110 \rightarrow 0110$  au système asynchrone de la figure 7.19. Etablir les équations et la table d'états valables pour cette seule séquence, puis déterminer le graphe des états. Mettre en évidence dans celui-ci les divers régimes transitoires de la sortie  $Z$  (ceux-ci présentent notamment des aléas dynamiques définis par les séquences  $Z = 1 \rightarrow 0 \rightarrow 1 \rightarrow 0$  ou  $Z = 0 \rightarrow 1 \rightarrow 0 \rightarrow 1$ ).

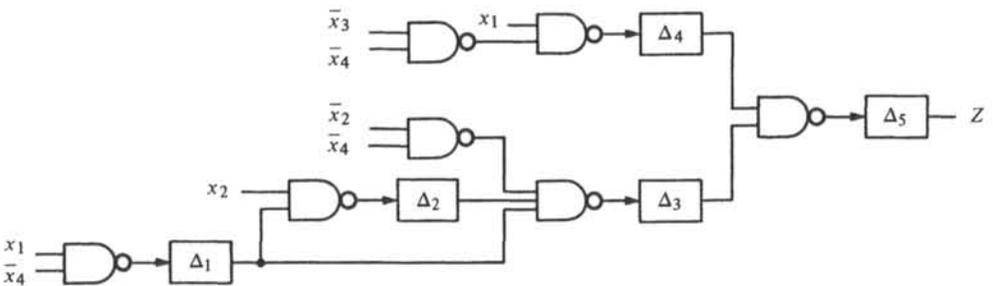


Fig. 7.19

7.3 SYSTÈMES SÉQUENTIELS

7.3.1 Exemple : élément de mémoire  $\bar{s}\bar{r}$

La figure 7.20 rappelle un système concret déjà étudié : l'élément de mémoire  $\bar{s}\bar{r}$  (§ 3.1.2); celui-ci est un assemblage de portes NAND avec une rétroaction : c'est donc un système séquentiel dont le modèle combinatoire, décrivant le régime permanent, a été analysé dans les paragraphes 3.1.4 et 3.1.5.

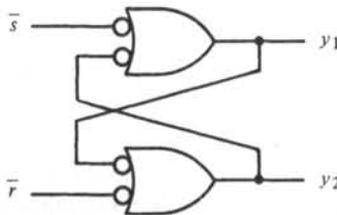


Fig. 7.20

7.3.2 Modèle asynchrone

Un modèle asynchrone de l'élément de mémoire (fig. 7.21) est obtenu en plaçant à la sortie de chaque porte NAND de la figure 7.20 un délai ( $\Delta_1$  et  $\Delta_2$ ). L'analyse de ce modèle déterminera le régime transitoire du système, c'est-à-dire le passage d'un état total stable à un autre état total stable lors d'un changement de l'état d'entrée.

D'après l'hypothèse du paragraphe 7.2.4, les délais  $\Delta_1$  et  $\Delta_2$  sont inertiels; le modèle asynchrone de la figure 7.21 est donc identique à celui de la figure 3.5, mais l'analyse sera différente puisqu'on a admis implicitement dans les paragraphes 3.1.9 et 3.1.10 que les délais étaient purs.

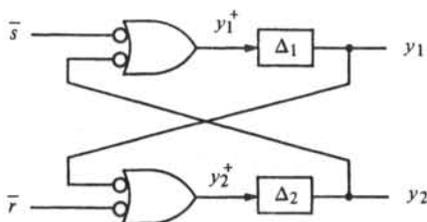


Fig. 7.21

### 7.3.3 Modes de représentation : équations et table d'états

Les équations (3.8) et (3.9) des excitations  $y_1^+$ ,  $y_2^+$  sont toujours valables :

$$y_1^+ = s + \bar{y}_2 \quad ; \quad y_2^+ = r + \bar{y}_1 \quad (7.28)$$

ceci nous permet d'établir la table d'états de la figure 7.22; les états totaux stables, qui décrivent le régime permanent, y sont encerclés.

$y_1^+ y_2^+$	$sr$			
	00	01	11	10
00	11	11	11	11
01	⊙01	⊙01	11	11
11	00	01	⊙11	10
10	⊙10	11	11	⊙10

$y_1, y_2$

Fig. 7.22

### 7.3.4 Définition : mode fondamental

Un système logique asynchrone (dont les délais sont inertiels) fonctionne dans le *mode fondamental* si toute variation de l'état d'entrée est effectuée à partir d'un état total stable.

En admettant le mode fondamental (ce qui a été fait implicitement dans la section 7.2), on facilite l'analyse et, en particulier, l'interprétation du graphe des états.

### 7.3.5 Mode de représentation : graphe des états

En admettant le mode fondamental, on constate dans la table d'états (fig. 7.22) la présence d'une seule course entre  $y_1$  et  $y_2$ ; en appliquant à l'état total stable  $s, r; y_1, y_2 = 11, 11$  la séquence d'entrée  $s, r = 11 \rightarrow 00$ , on atteint l'état total instable;

$$s, r; y_1, y_2 = 00, 11 \quad (7.29)$$

dont l'état d'excitation est :

$$y_1^+, y_2^+ = 00 \quad (7.30)$$

Les deux variables  $y_1$  et  $y_2$  sont instables (§ 7.2.8) et leur course est décrite dans un graphe des états (fig. 7.23). Pour  $\Delta_1 < \Delta_2$ , le système atteint l'état total stable  $s, r ; y_1, y_2 = 00, 01$ , tandis que pour  $\Delta_2 < \Delta_1$  il gagne l'état total stable  $00, 10$  qui est différent du précédent.

### 7.3.6 Définition : course critique

Une course entre  $k$  variables internes  $y_1, y_2, \dots, y_k$  qui produit des états totaux stables distincts selon les valeurs des paramètres  $\Delta_1, \Delta_2, \dots, \Delta_k$  est appelée *course critique*.

Ainsi la course entre  $y_1$  et  $y_2$  décrite par la figure 7.23 est critique. Pour éviter celle-ci, il suffit d'imposer la condition suivante (§ 3.1.8) :

$$s \cdot r = 0 \quad (7.31)$$

Par définition, la table d'états d'un système combinatoire avec délais ne comporte qu'un seul état total stable pour chaque état d'entrée (fig. 7.11) : aucune course n'est donc critique.

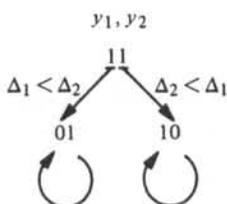


Fig. 7.23

### 7.3.7 Conclusion

Une première analyse asynchrone de l'élément de mémoire  $\bar{s}\bar{r}$  (§ 3.1.10), admettant l'existence de délais purs, a produit pour la séquence d'entrée  $s, r = 11 \rightarrow 00$  un régime transitoire sans amortissement (oscillation périodique).

En supposant que les délais sont inertiels, on a déterminé (§ 7.3.5) que le régime transitoire était amorti, mais que les états totaux finals dépendaient des valeurs des paramètres  $\Delta_1, \Delta_2$ .

Les mesures effectuées sur des systèmes concrets (électroniques) sont en accord avec cette dernière description et valident notre hypothèse du paragraphe 7.2.4 qui postule que tous les délais sont inertiels.

### 7.3.8 Définition : modèle asynchrone minimal

On a déjà relevé que le calcul du régime transitoire d'un système séquentiel nécessitait la présence d'un délai dans chaque boucle de rétroaction (§ 3.1.11).

Soit alors un système séquentiel dont le nombre minimal des boucles de rétroaction est égal à  $b$ . Nous appellerons *modèles asynchrones minimaux* tous les modèles asynchrones de ce système comportant  $b$  délais, chacun d'eux étant placé dans une boucle distincte.

### 7.3.9 Exemple : bascule bistable

Le logigramme de la figure 7.24 est celui d'une bascule bistable  $D$  (type 7474 de Texas Instruments). Le nombre minimal des boucles de rétroaction est de trois, de telle sorte qu'un modèle asynchrone minimal peut être représenté dans la figure 7.25 avec trois délais  $\Delta_1, \Delta_2, \Delta_3$ .

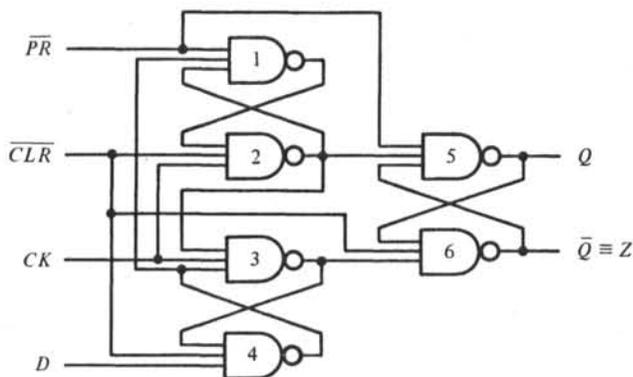


Fig. 7.24

On constate dans cet exemple que la détermination du nombre minimal des boucles n'est pas immédiate : il existe une méthode systématique, décrite dans la référence [74] (pp. 253 - 258, 267 - 271), qui détermine ce nombre.

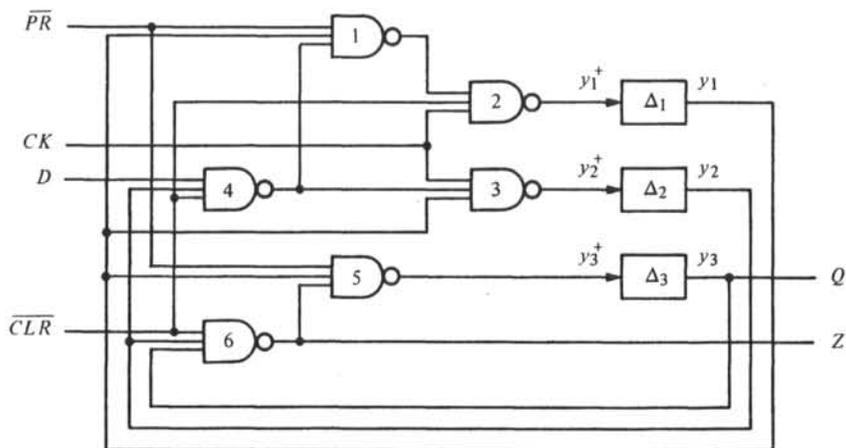


Fig. 7.25

### 7.3.10 Exemple : élément de mémoire $s\bar{r}$

La figure 3.2 montre que cet élément comporte au minimum une rétroaction; les modèles asynchrones minimaux ont donc un seul délai. L'un d'eux est représenté dans la figure 7.26 et décrit par les équations suivantes :

$$y_1^+ = s + \bar{r}y_1 \quad ; \quad y_2 = r + \bar{y}_1 \quad (7.32)$$

qui nous conduisent à la table d'états de la figure 7.27 (où  $y_2$  est une fonction de sortie). Les états  $y_1, y_2$  de cette table coïncident avec les états  $y_1, y_2$  de la table précédente (fig. 7.22) dans tout état total stable : le régime permanent est donc identique.

Par contre, la course critique résultant de la séquence d'entrée  $s, r = 11 \rightarrow 00$  (fig. 7.22) disparaît dans la nouvelle table (fig. 7.27) : en effet, le modèle asynchrone minimal de la figure 7.26 découle du modèle asynchrone précédent (fig. 7.21) en faisant  $\Delta_2 = 0$ , donc en admettant que :

$$\Delta_2 < \Delta_1 \quad (7.33)$$

On postule ainsi que la variable  $y_2$  gagne toujours la course engagée avec  $y_1$  et que l'état interne atteint dans le graphe (fig. 7.23) est  $y_1, y_2 = 10$  (fig. 7.27).

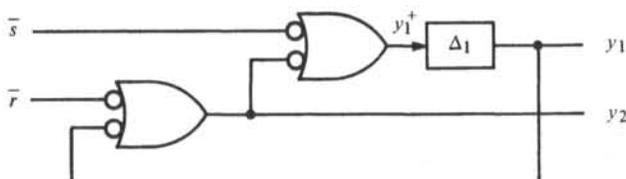


Fig. 7.26

$y_1^+, y_2$	$sr$			
	00	01	11	10
0	(0,1)	(0,1)	1,1	1,1
1	(1,0)	0,1	(1,1)	(1,0)
$y_1$				

Fig. 7.27

### ■ 7.3.11 Définition : systèmes asynchrones équivalents

Lorsque deux systèmes logiques asynchrones distincts peuvent être mis en correspondance de telle sorte que des séquences d'entrée identiques, respectant le mode fondamental, produisent des séquences de sortie identiques en régime permanent, on dit qu'ils sont *équivalents*, et leurs tables d'états sont *équivalentes*.

A condition d'interdire la course critique grâce à la relation (7.31)  $s \cdot r = 0$ , on constate dans les tables des figures 7.22 et 7.27 que les systèmes asynchrones des figures 7.21 et 7.26 sont équivalents : toute séquence d'entrée (débutant par exemple avec l'état  $s, r = 10$ ) appliquée aux deux systèmes et respectant le mode fondamental produit en régime permanent (c'est-à-dire dans les états totaux stables) la même séquence des états de sortie  $y_1, y_2$ . Le comportement de ces systèmes ne diffère que par les états totaux instables, c'est-à-dire par le régime transitoire.

### 7.3.12 Modèle asynchrone final : élément de mémoire $s\bar{r}$

Etant donné l'utilisation très fréquente de cet élément, on cherche un modèle asynchrone qui soit le plus simple possible. On constate alors dans la précédente table d'états (fig. 7.27) que  $y_2 = \bar{y}_1$  dans tout état total stable, sauf pour  $s, r = 11$ . En admet-

tant que la condition (7.31) est vérifiée, on définit alors le modèle asynchrone final de l'élément de mémoire  $\bar{s}\bar{r}$  qui est caractérisé par les équations :

$$y_1^+ = s + \bar{r}y_1 \quad \text{et} \quad y_2 = \bar{y}_1 \quad \text{si} \quad s \cdot r = 0 \quad (7.34)$$

La table d'états de la figure 7.28 et le schéma de la figure 7.30 représentent ce modèle. On a rappelé pour mémoire dans la figure 7.29 le logigramme habituel de l'élément  $\bar{s}\bar{r}$ .

D'après la définition du paragraphe 7.3.11, ce nouveau modèle asynchrone est équivalent à ceux des figures 7.21 et 7.26 si  $s \cdot r = 0$ .

En régime permanent ( $y_1^+ = y_1$ ) on retrouve les équations (3.5), (3.7) et (3.6) du chapitre 3.

$y_1^+, y_2$	$sr$			
	00	01	11	10
0	0,1	0,1	-	1,1
1	1,0	0,0	-	1,0

Fig. 7.28

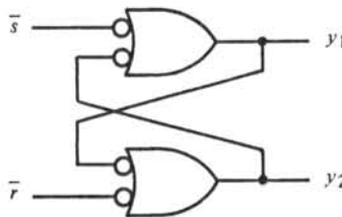


Fig. 7.29

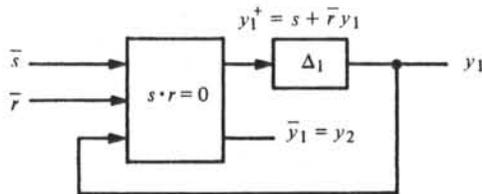


Fig. 7.30

### 7.3.13 Conclusion générale

Tout système séquentiel admet plusieurs modèles asynchrones distincts; ceux-ci sont équivalents s'ils décrivent (dans le mode fondamental) le même régime permanent. Si le régime transitoire est indifférent, il est manifeste qu'un modèle asynchrone minimal, équivalent à un modèle non minimal, est plus simple à calculer et plus commode à représenter : c'est cette démarche que nous avons adoptée pour analyser l'élément de mémoire  $\bar{s}\bar{r}$ .

### 7.3.14 Bibliographie

L'analyse du premier modèle asynchrone de l'élément de mémoire  $\bar{s}\bar{r}$  (fig. 7.21) est traitée dans les références [45] (pp. 127 - 129), [85] (pp. 31 - 37, 72 - 73), [84]

(pp. 172 - 181, 194 - 198) et [87]; l'analyse des systèmes séquentiels quelconques (c'est-à-dire sans bascules bistables) fait l'objet des références [10] (pp. 363 - 367, 401 - 408), [24] (pp. 551 - 576, 626 - 641), tandis que la synthèse de ceux-ci, découlant essentiellement des travaux de Huffman [65] et Unger [102], est notamment exposée dans les références [10] (pp. 367 - 394, 424 - 426), [14] (pp. 277 - 305), [18] (pp. 173 - 203), [24] (pp. 577 - 626), [45] (pp. 129 - 143) et sera traitée dans le volume XI.

■ 7.3.15 Exercice

On dispose de quatre portes NAND à deux entrées (No 1 à 4) dont le modèle asynchrone est obtenu en plaçant un délai inertiel ( $\Delta_1$  à  $\Delta_4$ ) sur la sortie. En connectant ces portes par paires, selon le logigramme de l'élément de mémoire  $\bar{s}\bar{r}$  (fig. 7.20), et en exploitant les résultats de la course critique (fig. 7.23), on demande quels sont les nombres minimal et maximal d'assemblages à réaliser pour obtenir expérimentalement un classement des quatre délais  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  selon leurs valeurs relatives (par exemple :  $\Delta_4 > \Delta_1 > \Delta_3 > \Delta_2$ ).

7.3.16 Exercice

On donne un modèle asynchrone de l'élément de mémoire  $\bar{s}\bar{r}$  (fig. 7.31). Etablir la table et le graphe des états pour la séquence  $s, r = 11 \rightarrow 00$  (mode fondamental). Déterminer les valeurs relatives des paramètres  $\Delta_1, \Delta_2, \Delta_3, \Delta_4$  qui produisent l'état final  $y_1, y_2 = 01$  ou  $10$ , puis comparer avec les résultats du paragraphe 7.3.5.

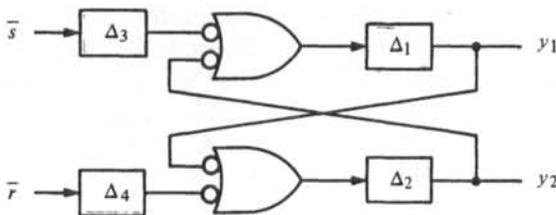


Fig. 7.31

7.3.17 Exercice

Analyser les systèmes asynchrones des figures 7.32 et 7.33 en établissant les tables et graphes des états, ainsi que les chronogrammes. Montrer que ces deux dispositifs sont des *oscillateurs logiques*. Quelle est leur période  $T$  ?

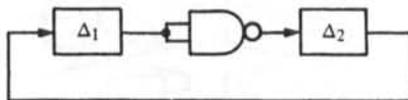


Fig. 7.32

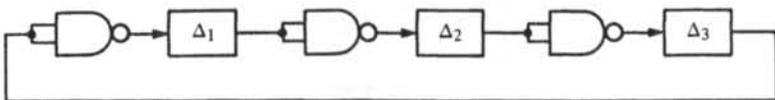


Fig. 7.33

## 7.3.18 Exercice

Établir la table d'états du système asynchrone de la figure 7.34. Existe-t-il, pour une variation de l'état d'entrée  $x_1, x_2$  (mode fondamental), un aléa statique de la sortie  $Z$ ? Dans l'affirmative, déterminer la durée  $\Delta T$  de celui-ci.

On constate que les portes NAND No 1 et 2 réalisent un élément de mémoire  $\delta\bar{f}$ . Est-il possible de décrire celui-ci par l'un des modèles asynchrones minimaux des paragraphes 7.3.10 ou 7.3.12? Dans l'affirmative, analyser le nouveau modèle asynchrone du système et comparer les résultats aux précédents.

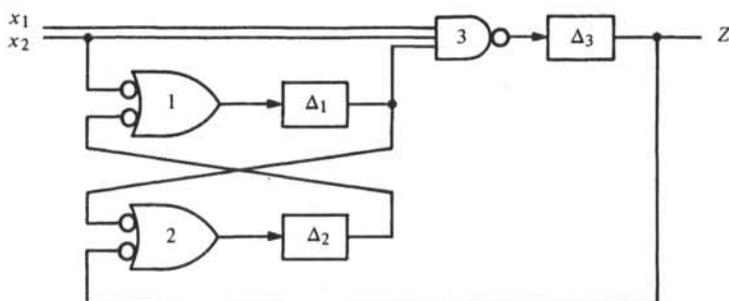


Fig. 7.34

## ■ 7.3.19 Exercice

Effectuer l'analyse (table d'états) de deux modèles asynchrones d'une bascule  $D$ : le premier est obtenu en insérant à la sortie de chaque porte NAND de la figure 7.24 (No 1 à 6) un élément de délai ( $\Delta_1$  à  $\Delta_6$ ); le second est celui de la figure 7.25 ( $\Delta_1, \Delta_2, \Delta_3$ ). Si l'on admet le fonctionnement normal ( $PR = CLR = 0$ ) et des changements d'une seule variable d'entrée ( $D$  ou  $CK$ ) dans le mode fondamental, les deux modèles sont-ils équivalents? La sortie  $Z$  est-elle toujours égale à  $\bar{Q}$  en régime transitoire? en régime permanent?

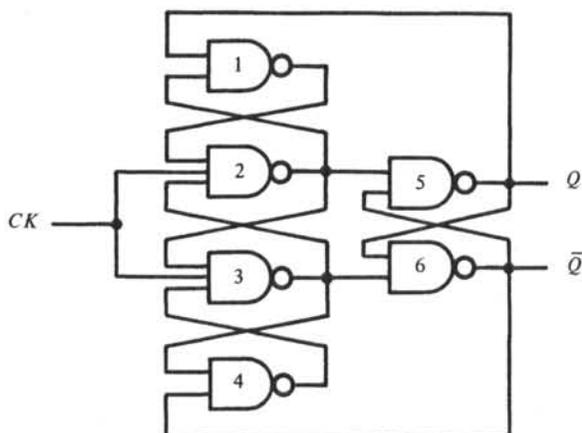


Fig. 7.35

Est-il possible de décrire les portes NAND No 5 et 6 par l'un des modèles asynchrones minimaux de l'élément de mémoire  $\bar{s}\bar{r}$  (§ 7.3.10 ou 7.3.12) ? Dans l'affirmative, reprendre l'analyse de la figure 7.25 et comparer les résultats aux précédents.

### ■ 7.3.20 Exercice

Effectuer l'analyse (table d'états) de deux modèles asynchrones d'un diviseur de fréquence (§ 3.3.9) : le premier est obtenu en insérant à la sortie de chaque porte NAND de la figure 7.35 (No 1 à 6) un élément de délai, le second en cherchant un nombre minimal de rétroactions. Démontrer que le fonctionnement de ce diviseur est indépendant des valeurs des délais. Les sorties  $Q$  et  $\bar{Q}$  sont-elles toujours complémentaires en régime transitoire ?

## 7.4 BASCULES BISTABLES

### 7.4.1 Exemple : bascule SR

Le logigramme de celle-ci (fig. 3.8) fait apparaître deux éléments de mémoire  $\bar{s}\bar{r}$  qui sont représentés plus simplement par les figures 3.13 et 3.14. En décrivant chacun de ces éléments avec le modèle du paragraphe 7.3.12, on obtient un modèle asynchrone de la bascule SR (fig. 7.36); celui-ci est minimal (§ 7.3.8).

### 7.4.2 Mode de représentation : équations

Les deux éléments de mémoire  $\bar{s}\bar{r}$  sont donc décrits par les relations (7.34) :

$$y_1^+ = s_1 + \bar{r}_1 y_1 \quad \text{si} \quad s_1 \cdot r_1 = 0 \quad (7.35)$$

$$y_2^+ = s_2 + \bar{r}_2 y_2 \quad \text{si} \quad s_2 \cdot r_2 = 0 \quad (7.36)$$

On tire de la figure 7.36 les expressions de  $s_1, r_1, s_2$  et  $r_2$  :

$$\bar{s}_1 = S \cdot \overline{CLR} \cdot \overline{CK} \cdot \overline{PR} ; \quad s_1 = S \cdot \overline{CLR} \cdot \overline{CK} + PR \quad (7.37)$$

$$\begin{aligned} \bar{r}_1 &= R \cdot \overline{PR} \cdot \overline{CK} \cdot \overline{CLR} = (\bar{R} + PR + CK) \overline{CLR} ; \\ r_1 &= R \cdot \overline{PR} \cdot \overline{CK} + CLR \end{aligned} \quad (7.38)$$

$$\bar{s}_2 = y_1 \cdot \overline{CLR} \cdot \overline{CK} \cdot \overline{PR} ; \quad s_2 = y_1 \cdot \overline{CLR} \cdot CK + PR \quad (7.39)$$

$$\begin{aligned} \bar{r}_2 &= \overline{y_1} \cdot \overline{PR} \cdot \overline{CK} \cdot \overline{CLR} = (y_1 + PR + \overline{CK}) \overline{CLR} ; \\ r_2 &= \overline{y_1} \cdot \overline{PR} \cdot CK + CLR \end{aligned} \quad (7.40)$$

qui sont ensuite reportées dans les équations (7.35) et (7.36) :

$$y_1^+ = PR + (S \cdot \overline{CK} + \bar{R} \cdot y_1 + CK \cdot y_1) \overline{CLR} \quad (7.41)$$

$$y_2^+ = PR + (CK \cdot y_1 + \overline{CK} \cdot y_2) \overline{CLR} \quad (7.42)$$

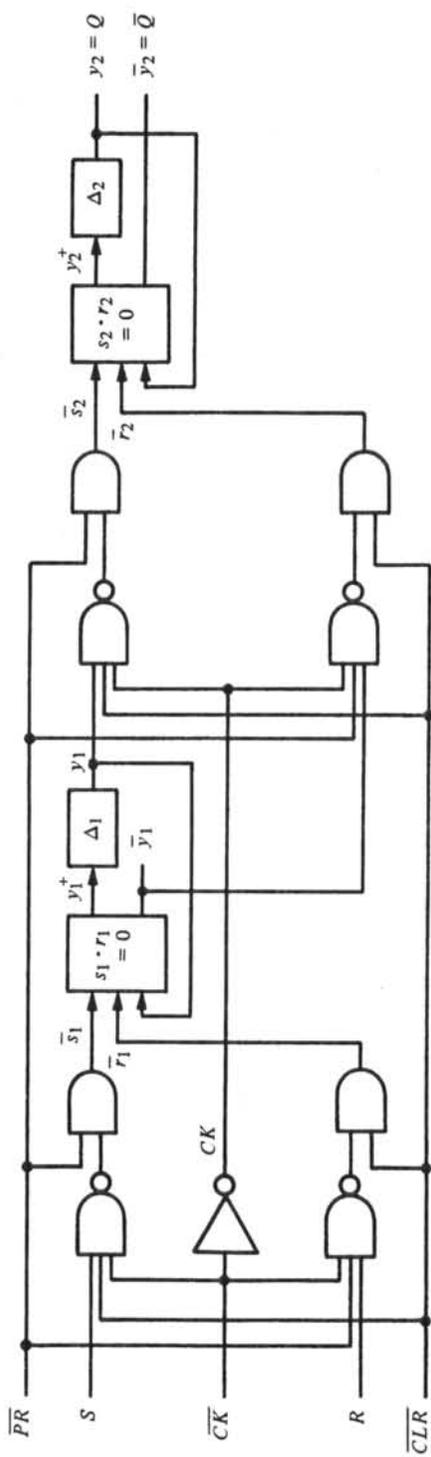


Fig. 7.36

Les relations (7.41) et (7.42) sont valables si l'on vérifie les deux conditions suivantes :

$$s_1 \cdot r_1 = S \cdot R \cdot \overline{PR} \cdot \overline{CLR} \cdot \overline{CK} + PR \cdot CLR = 0 \quad (7.43)$$

$$s_2 \cdot r_2 = PR \cdot CLR = 0 \quad (7.44)$$

Les équations (7.41) à (7.44) sont identiques aux relations (3.24), (3.32), (3.21) et (3.29) du chapitre 3 si l'on admet un régime permanent ( $y_1^+ = y_1$  et  $y_2^+ = y_2$ ) et si l'on remplace  $y_1$  par  $y_{11}$ ,  $y_2$  par  $Q$ .

### 7.4.3 Mode de représentation : table d'états

Par commodité, on partage celle-ci selon les quatre états  $PR, CLR$  (fig. 7.37). D'après les définitions du paragraphe 3.2.6, on distingue le fonctionnement normal ( $PR, CLR = 00$ ), la mise à 1 ( $PR, CLR = 10$ ) et la remise à 0 ( $PR, CLR = 01$ ); dans ces deux derniers cas, les fonctions d'excitation  $y_1^+$  (7.41) et  $y_2^+$  (7.42) sont indépendantes des variables d'entrée  $S, R, CK$ , de telle sorte qu'une seule colonne les représente dans la table d'états :

$$y_1^+(PR, CLR) = y_1^+(1, 0) = 1 ; y_2^+(PR, CLR) = y_2^+(1, 0) = 1 \quad (7.45)$$

$$y_1^+(PR, CLR) = y_1^+(0, 1) = 0 ; y_2^+(PR, CLR) = y_2^+(0, 1) = 0 \quad (7.46)$$

La condition (7.44) interdit l'état  $PR, CLR = 11$  (tirets dans la table).

Dans le fonctionnement normal ( $PR, CLR = 00$ ) les fonctions d'excitation  $y_1^+$  (7.41) et  $y_2^+$  (7.42) s'écrivent sous la forme suivante :

$$y_1^+ = S \cdot \overline{CK} + (\overline{R} + CK) y_1 \quad (7.47)$$

$$y_2^+ = CK \cdot y_1 + \overline{CK} \cdot y_2 \quad (7.48)$$

Il est donc aisé de dresser la partie correspondante de la table dans laquelle l'état d'entrée  $S, R, CK = 110$  est interdit (tirets) par la condition (7.43) :

$$s_1 \cdot r_1 = S \cdot R \cdot \overline{CK} = 0 \quad (7.49)$$

Pour  $PR, CLR = 00$ , la table d'états (fig. 7.37) ne présente aucune course; par contre, pour  $PR, CLR = 10$  et  $01$  il existe des courses entre  $y_1$  et  $y_2$  : étant donné le comportement combinatoire de la bascule dans ces cas, celles-ci ne sont pas critiques (§ 7.3.6). Si l'on admet le mode fondamental (§ 7.3.4), la table d'états de la figure 7.37 constitue alors un mode de représentation complet. On a encadré enfin la valeur de  $y_2$ , puisque celle-ci se confond avec l'état de sortie  $Q$  de la bascule (fig. 7.37).

### ■ 7.4.4 Modèles asynchrone et quasi-synchrone : fonctionnement normal

Le chronogramme de la figure 3.15 représente le fonctionnement normal d'une bascule  $SR$  ( $PR, CLR = 00$ ) validant le modèle quasi-synchrone du paragraphe 3.2.3 : toute variation des entrées d'excitation  $S, R$  est interdite pendant les instants d'horloge

$y_1^+ y_2^+$		$PR, CLR = 00$ $CK = 0$ $SR$				$CK = 1$ $SR$				$PR, CLR = 10$	$PR, CLR = 01$	$PR, CLR = 11$
		00	01	11	10	00	01	11	10			
0	0	00	00	-	10	00	00	00	00	11	00	-
0	1	01	01	-	11	00	00	00	00	11	00	-
1	1	11	01	-	11	11	11	11	11	11	00	-
1	0	10	00	-	10	11	11	11	11	11	00	-

$y_1, y_2 = Q$

Fig. 7.37

( $CK = 0$ ). En rappelant que  $y_{11} = y_1$  et  $Q = y_{12} = y_2$ , il est possible de représenter la séquence des états d'entrée  $S, R, CK$  et des états internes  $y_1, y_2$  du chronogramme (figure 3.15 examinée en partant de l'extrême gauche) dans la table d'états à condition de respecter le mode fondamental : on a illustré par des flèches dans la figure 7.38 le début de cette séquence. En achevant l'examen du chronogramme, on aurait parcouru dans la table l'ensemble des états totaux du modèle quasi-synchrone : ces états sont marqués en trait fort dans la figure 7.38.

$y_1^+ y_2^+$		$CK = 0$ $SR$				$CK = 1$ $SR$			
		00	01	11	10	00	01	11	10
0	0	00	00	-	10	00	00	00	00
0	1	01	01	-	11	00	00	00	00
1	1	11	01	-	11	11	11	11	11
1	0	10	00	-	10	11	11	11	11

$y_1, y_2 = Q$

Fig. 7.38

Le modèle asynchrone nous permet d'analyser des comportements interdits par le modèle quasi-synchrone. Le chronogramme et la table d'états des figures 7.40 et 7.39 illustrent l'exemple d'une variation momentanée de l'entrée  $S$  pendant un instant d'horloge ( $CK = 0$ ); on constate qu'une telle variation est mémorisée par la bascule et provoque un changement de l'état  $Q$  à la montée du signal d'horloge  $CK$ .

#### ■ 7.4.5 Modèles asynchrone et quasi-synchrone : mise à 1 et remise à 0

Le modèle quasi-synchrone interdit toute variation des entrées d'excitation ( $S, R$ ) et des entrées asynchrones ( $PR, CLR$ ) pendant les instants d'horloge ( $CK = 0$ ); le modèle asynchrone offre par contre la possibilité d'étudier de tels comportements. Le chronogramme et la table d'états des figures 7.42 et 7.41 illustrent l'exemple d'un changement momentanée de l'entrée  $PR$  pendant un instant d'horloge; on constate que l'effet d'une telle variation (mise à 1) disparaît lors de la montée du signal d'horloge (l'état  $Q$  reprend la valeur 0).

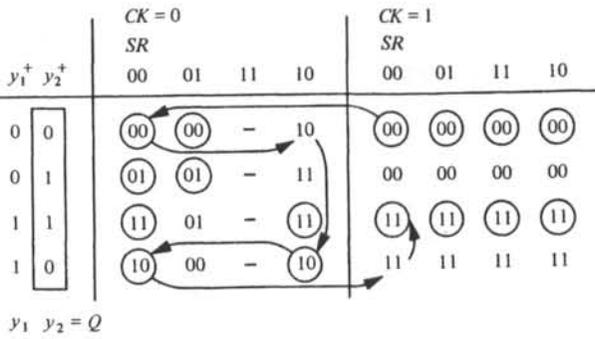


Fig. 7.39

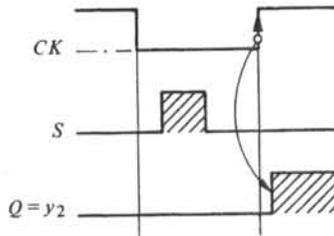


Fig. 7.40

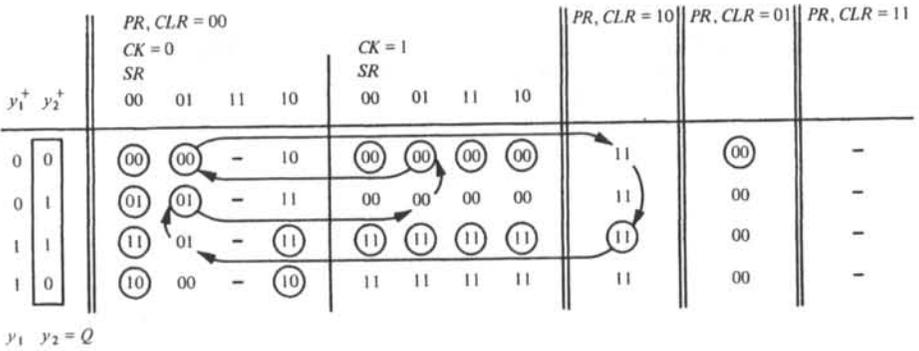


Fig. 7.41

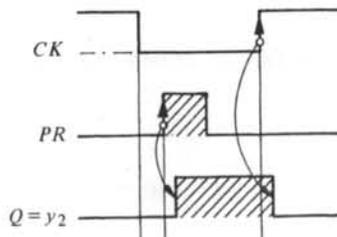


Fig. 7.42

### 7.4.6 Application : bascule JK

Celle-ci est constituée par l'assemblage d'une bascule SR et de deux portes ET, supposées sans délai et réalisant les équations suivantes (§ 3.3.5) :

$$S = J\bar{Q} = J\bar{y}_2 \quad ; \quad R = KQ = Ky_2 \quad (7.50)$$

En remplaçant  $S$  et  $R$  par leurs nouvelles valeurs (7.50) dans l'équation (7.41) nous obtenons :

$$y_1^+ = PR + (J \cdot \bar{CK} \cdot \bar{y}_2 + \bar{K} \cdot y_1 + CK \cdot y_1 + y_1 \bar{y}_2) \overline{CLR} \quad (7.51)$$

tandis que l'expression (7.42) reste inchangée :

$$y_2^+ = PR + (CK \cdot y_1 + \bar{CK} \cdot y_2) \overline{CLR} \quad (7.52)$$

et que les conditions (7.43) et (7.44) se résument à :

$$PR \cdot CLR = 0 \quad (7.53)$$

La figure 7.43 illustre la table d'états. Pour le fonctionnement normal ( $PR, CLR = 00$ ) on a représenté en trait fort les états totaux du modèle quasi-synchrone; cette partie de la table est équivalente (§ 7.3.11) à celle de la bascule 7476 de Texas Instruments [87] (pp. 281 - 283), [88].

$y_1^+ y_2^+$	$PR, CLR = 00$ $CK = 0$ JK				$CK = 1$ JK				$PR, CLR = 10$	$PR, CLR = 01$	$PR, CLR = 11$
	00	01	11	10	00	01	11	10			
0 0	00	00	10	10	00	00	00	00	11	00	-
0 1	01	01	01	01	00	00	00	00	11	00	-
1 1	11	01	01	11	11	11	11	11	11	00	-
1 0	10	10	10	10	11	11	11	11	11	00	-

$y_1, y_2 = Q$

Fig. 7.43

### 7.4.7 Application : bascule D

La bascule D est une bascule SR qui vérifie l'équation (§ 3.3.1) :

$$S = D \quad ; \quad R = \bar{D} \quad (7.54)$$

En introduisant les relations (7.54) dans l'équation (7.41) et en simplifiant celle-ci, on obtient :

$$y_1^+ = PR + (D \cdot \bar{CK} + CK \cdot y_1) \overline{CLR} \quad (7.55)$$

et, inchangée, l'équation (7.42) :

$$y_2^+ = PR + (CK \cdot y_1 + \overline{CK} \cdot y_2) \overline{CLR} \tag{7.56}$$

Les conditions (7.43) et (7.44) deviennent simplement :

$$PR \cdot CLR = 0 \tag{7.57}$$

Dans la table d'états de la figure 7.44, on a représenté en trait fort les états totaux validant le modèle quasi-synchrone pour le fonctionnement normal ( $PR, CLR = 00$ ); cette partie de la table est équivalente (§ 7.3.11) à celle de la bascule 7474 de Texas Instruments (voir la figure 7.24 et l'exercice du paragraphe 7.3.19) [46] (pp. 109 - 110), [84] (pp. 218 - 229), [87] (pp. 283 - 286).

$y_1^+ y_2^+$		$PR, CLR = 00$				$PR, CLR = 10$	$PR, CLR = 01$	$PR, CLR = 11$
		$CK = 0$		$CK = 1$				
		$D, CK$						
		00	10	11	01			
0	0	00	10	00	00	11	00	—
0	1	01	11	00	00	11	00	—
1	1	01	11	11	11	11	00	—
1	0	00	10	11	11	11	00	—

$y_1 y_2 = Q$

Fig. 7.44

**7.4.8 Définition : bascule commandée par un flanc**

Une variation momentanée de l'entrée  $D$  pendant un instant d'horloge, interdite dans le modèle quasi-synchrone, est illustrée par la table et le chronogramme des figures 7.45 et 7.47 : on remarque dans ce cas que cette impulsion est sans effet sur l'état  $Q$ . C'est la valeur de l'excitation  $D$  lors de la montée du signal d'horloge qui détermine définitivement la valeur de la sortie  $Q$ ; on dit qu'une telle bascule est *commandée par un flanc*, celui du signal d'horloge (en anglais : *edge-triggered* ou *edge-sensitive flip-flop*) [28] (pp. 121 - 124), [84] (pp. 218 - 229).

$y_1^+ y_2^+$		$D, CK$			
		00	10	11	01
0	0	00	10	00	00
0	1	01	11	00	00
1	1	01	11	11	11
1	0	00	10	11	11

$y_1 y_2 = Q$

Fig. 7.45

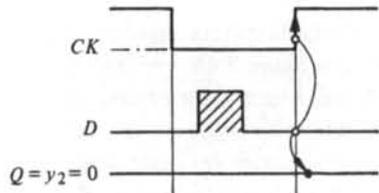


Fig. 7.47

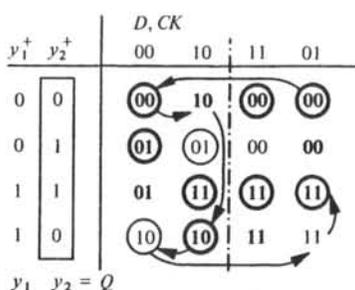


Fig. 7.46

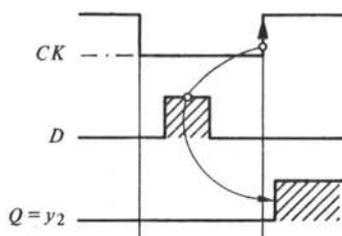


Fig. 7.48

### 7.4.9 Application : bascule $D$

Une bascule  $JK$  vérifiant les équations (fig. 6.47) :

$$J = D \quad ; \quad K = \bar{D} \quad (7.58)$$

réalise aussi une bascule  $D$ . En introduisant les relations (7.58) dans l'équation (7.51) on obtient :

$$y_1^+ = PR + (D \cdot \bar{C}\bar{K} \cdot \bar{y}_2 + D \cdot y_1 + CK \cdot y_1 + y_1 \bar{y}_2) \bar{C}\bar{L}\bar{R} \quad (7.59)$$

et, inchangées :

$$y_2^+ = PR + (CK \cdot y_1 + \bar{C}\bar{K} \cdot y_2) \bar{C}\bar{L}\bar{R} \quad (7.60)$$

$$PR \cdot CLR = 0 \quad (7.61)$$

La table d'états, représentée pour le fonctionnement normal ( $PR, CLR = 00$ ), est celle de la figure 7.46. Les états totaux du modèle quasi-synchrone (en trait fort) sont identiques à ceux de la précédente bascule  $D$  (fig. 7.44); parmi les états totaux restants, on remarque au contraire des différences entre les deux tables, démontrant que les modèles asynchrones des paragraphes 7.4.7 et 7.4.9 ne sont pas équivalents.

En appliquant à la nouvelle table d'états (fig. 7.46) la séquence d'entrée de la figure 7.47, on obtient un chronogramme différent (fig. 7.48) : la bascule a mémorisé l'impulsion de  $D$  et n'est donc pas commandée par le flanc du signal d'horloge.

#### 7.4.10 Conclusion

En admettant que les variations des entrées d'excitation et des entrées asynchrones sont interdites pendant les instants d'horloge, on peut représenter une bascule bistable par son modèle quasi-synchrone (ou synchrone) : celui-ci est unique pour un type donné ( $SR, JK, D$ , etc...).

Si, au contraire, l'on admet que les entrées d'excitation et les entrées asynchrones varient pendant les instants d'horloge, on doit décrire la bascule par un modèle asynchrone : pour un type donné ( $D$  par exemple) il existe une famille de modèles asynchrones qui dépendent du logigramme de la bascule et de la répartition des délais dans celui-ci. Tous les modèles d'une famille (ceux des paragraphes 7.4.7 et 7.4.9 par exemple) décrivent le même modèle quasi-synchrone et sont donc équivalents (§ 7.3.11) pour les séquences d'entrée validant celui-ci; pour toute autre séquence, ces modèles asynchrones peuvent présenter des comportements distincts et ne sont alors pas équivalents.

#### 7.4.11 Bibliographie

La distinction entre les modèles quasi-synchrones et asynchrones est mise en évidence dans les références [87] et [88], tandis que dans [67] (pp. 133 - 140) et [84] (pp. 198 - 232) on trouvera une étude systématique des différents modèles asynchrones des bascules  $SR, JK$  et  $D$ . La synthèse de ces bascules fait l'objet des références [10] (pp. 421 - 429), [46] (pp. 103 - 108), [67] (pp. 137 - 140, 157 - 159), [87] (pp. 273 - 276), [89] (pp. 156 - 184) et [90].

#### 7.4.12 Exercice

Un modèle asynchrone non minimal de la bascule  $SR$  est obtenu en rajoutant dans la figure 7.36 un délai  $\Delta_3$  à la sortie de l'inverseur produisant  $CK$  ( $y_3^+ = CK$ ). Effectuer l'analyse de ce modèle (équations, table d'états) pour  $PR, CLR = 00$ , puis tracer un graphe pour chaque variation de l'état d'entrée  $S, R, CK$  respectant le mode fondamental et produisant une course. Déterminer enfin les valeurs relatives des délais  $\Delta_1, \Delta_2, \Delta_3$  pour que ce système asynchrone soit équivalent (§ 7.3.11) à celui de la figure 7.36 (table d'états : figure 7.37).

Il apparaît dans cet exemple une course critique entre les variables internes  $y_1$  et  $y_3$ ; celle-ci est la manifestation d'un *aléa essentiel* dont on trouvera une étude détaillée dans les références [10] (pp. 399 - 401), [18] (pp. 207 - 210), [24] (pp. 572 - 574), [86] et [102] (pp. 143 - 163, 177 - 179, 258 - 259).

#### 7.4.13 Exercice

Un système concret, réalisé avec des transistors MOS (Metal Oxide Semiconductor), est décrit par le modèle asynchrone de la figure 7.49. Etablir la table d'états et représenter chaque course de celle-ci dans un graphe. Si  $S \cdot R \cdot \overline{CK} = 0$ , peut-on montrer que cette table est équivalente (§ 7.3.11) à celle de la figure 7.37 ( $PR, CLR = 00$ ) ?

#### 7.4.14 Exercice

Dans la table d'états de la figure 7.37 ( $PR, CLR = 00$ ), on veut déterminer la séquence des états d'entrée  $S, R, CK$  amenant la bascule :

- de l'état total  $S, R, CK$  ;  $y_1, y_2 = 000, 00$  à l'état total  $000, 01$  ;
- de l'état total  $S, R, CK$  ;  $y_1, y_2 = 100, 10$  à l'état total  $011, 00$ .

Dans les deux cas, on admet le mode fondamental et la variation d'une seule variable d'entrée à la fois.

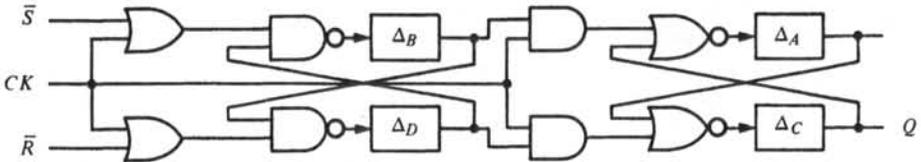


Fig. 7.49

#### 7.4.15 Exercice

Le logigramme de la figure 7.50 est celui d'une bascule  $SR$ . Déterminer un modèle asynchrone minimal en considérant que les deux éléments de mémoire  $\bar{s}\bar{r}$  sont décrits par les relations (7.34), puis analyser ce modèle (équations, table d'états) et comparer avec la figure 7.37 ( $PR, CLR = 00$ ); ces tables sont-elles totalement ou partiellement équivalentes? Mettre en évidence les éventuelles différences, puis examiner dans les deux cas l'effet d'une impulsion de  $S$  ( $R = 0$ ) selon le chronogramme de la figure 7.40.

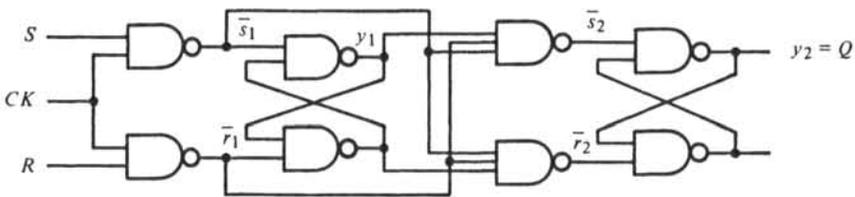


Fig. 7.50

#### 7.4.16 Exercice

On transforme une bascule  $D$  en une bascule  $JK$  (fig. 6.47) si l'on vérifie l'équation :

$$D = J\bar{Q} + \bar{K}Q = J\bar{y}_2 + \bar{K}y_2 \quad (7.62)$$

En admettant que le système combinatoire qui réalise l'expression (7.62) est sans délai, on peut introduire celle-ci dans les trois équations (7.55) à (7.57) de la bascule  $D$  et obtenir ainsi un nouveau modèle asynchrone de la bascule  $JK$ . Comparer sa table d'états avec celle de la figure 7.43 pour y rechercher une équivalence complète ou partielle et pour examiner l'effet d'une impulsion de  $J$  ( $K = 0$  ;  $PR, CLR = 00$ ) selon le chronogramme de la figure 7.51.

#### 7.4.17 Exercice

Le logigramme de la figure 7.52 est celui d'une bascule  $JK$ . Déterminer un modèle asynchrone minimal en considérant que les deux éléments de mémoire  $\bar{s}\bar{r}$  sont décrits par les relations (7.34), puis analyser ce modèle (équations, table d'états) et comparer

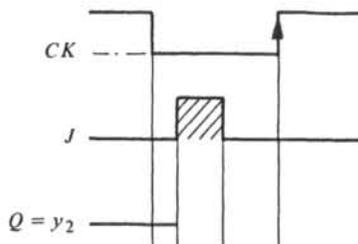


Fig. 7.51

avec la figure 7.43 ( $PR, CLR = 00$ ) : ces tables sont-elles totalement ou partiellement équivalentes ? Examiner dans les deux cas l'effet d'une impulsion de  $J$  ( $K = 0$ ) selon le chronogramme de la figure 7.51.

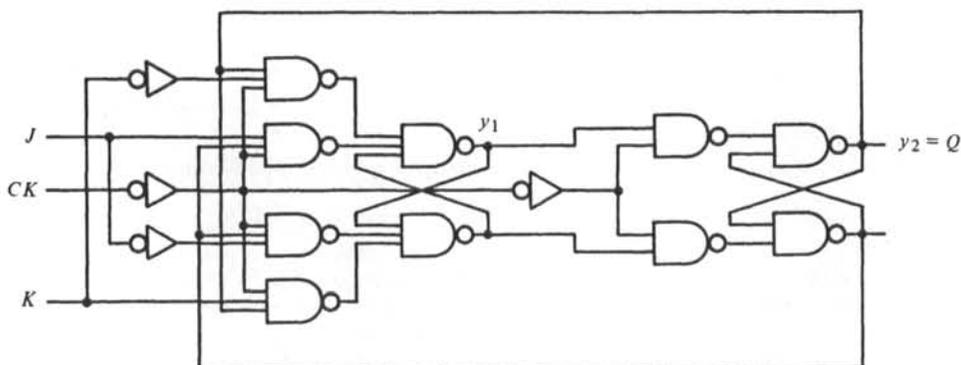


Fig. 7.52

#### ■ 7.4.18 Exercice

La table d'états de la figure 7.53 est celle d'une bascule  $JK$ . Représenter en trait fort les états totaux du modèle quasi-synchrone (§ 3.2.3), puis comparer cette table avec celle de la figure 7.43 ( $PR, CLR = 00$ ) : sont-elles totalement ou partiellement équivalentes ? Décrire à l'aide d'un chronogramme le comportement de cette bascule particulière (en anglais *data lockout flip-flop*, c'est-à-dire *bascule verrouillée*) [10] (pp. 420 - 421, 423 - 424).

#### 7.4.19 Exercice

Effectuer l'analyse du modèle asynchrone de la figure 7.25 (bascule  $D$ ; voir l'exercice du paragraphe 7.3.19). Comparer sa table d'états avec celle de la figure 7.44 : sont-elles partiellement ou totalement équivalentes ? Appliquer aux deux tables les séquences des figures 7.42 ( $CLR = 0, D = 0$ ) et 7.47 ( $PR, CLR = 00$ ) puis comparer les résultats.

#### 7.4.20 Exercice

La figure 7.54, extraite du catalogue [83] (pp. 76 - 77), décrit une bascule  $JK$  (type 7476) et une bascule  $D$  (type 7474).

$y_1^+ y_2^+ y_3^+$	$CK = 0$				$CK = 1$			
	$JK$				$JK$			
	00	01	11	10	00	01	11	10
0 0 0	000	000	000	000	100	100	001	001
1 0 0	000	000	000	000	100	100	100	100
0 0 1	011	011	011	011	001	001	001	001
0 1 1	011	011	011	011	111	010	010	111
1 1 1	011	011	011	011	111	111	111	111
0 1 0	000	000	000	000	010	010	010	010

$y_1 y_2 y_3$   
(Q)

Fig. 7.53

La bascule  $JK$  (7476) est équivalente à celle du paragraphe 7.4.6, représentée par la table d'états de la figure 7.43, tandis que la bascule  $D$  (7474) est équivalente à celle du paragraphe 7.4.7 (fig. 7.44). On demande de comparer les tables d'états avec les tableaux de la figure 7.54 et de mettre en évidence d'éventuelles différences ou omissions.

Bascule $JK$ (7476)					Bascule $D$ (7474)							
Entrées				Sorties		Entrées				Sorties		
$\overline{PR}$	$\overline{CLR}$	$\overline{CK}$	$J$	$K$	$Q$	$\overline{Q}$	$\overline{PR}$	$\overline{CLR}$	$CK$	$D$	$Q$	$\overline{Q}$
0	1	$\emptyset$	$\emptyset$	$\emptyset$	1	0	0	1	$\emptyset$	$\emptyset$	1	0
1	0	$\emptyset$	$\emptyset$	$\emptyset$	0	1	1	0	$\emptyset$	$\emptyset$	0	1
0	0	$\emptyset$	$\emptyset$	$\emptyset$	1*	1*	0	0	$\emptyset$	$\emptyset$	1*	1*
1	1	$\sqcap$	0	0	$Q_0$	$\overline{Q}_0$	1	1	$\uparrow$	1	1	0
1	1	$\sqcap$	1	0	1	0	1	1	$\uparrow$	0	0	1
1	1	$\sqcap$	0	1	0	1	1	1	0	$\emptyset$	$Q_0$	$\overline{Q}_0$
1	1	$\sqcap$	1	1	TOGGLE		1	1	0	$\emptyset$	$Q_0$	$\overline{Q}_0$

- $\sqcap$  : impulsion vers l'état 1; les entrées doivent être invariantes pendant que  $\overline{CK} = 1$ ; l'information des entrées est transmise à la sortie à la descente de l'impulsion;
- $\uparrow$  : transition de 0 à 1;
- $Q_0 =$  valeur de  $Q$  à l'instant précédant l'état d'entrée indiqué;
- TOGGLE : chaque sortie est inversée à chaque transition active (impulsion) du signal d'horloge;
- \* : cette configuration n'est pas stable, en ce sens qu'elle ne persistera pas lorsque les entrées  $\overline{PR}$  et  $\overline{CLR}$  reviendront à leurs valeurs inactives (1).

Fig. 7.54

7.5 SYSTÈMES SÉQUENTIELS SYNCHRONES ET QUASI-SYNCHRONES

7.5.1 Exemple : compteur synchrone réversible

La figure 7.55 donne le logigramme d'un tel compteur dont le modèle synchrone est décrit par les équations :

$$Q_y^+ = D_y = x \oplus Q_y \oplus Q_v \ ; \ Q_v^+ = D_v = \bar{Q}_v \tag{7.63}$$

On en déduit une table et un graphe des états (fig. 7.56 et 7.57).

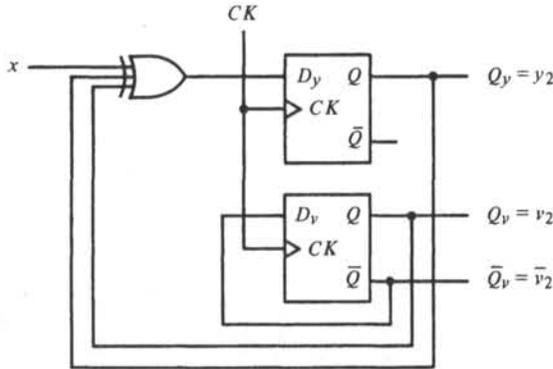


Fig. 7.55

$Q_y^+ Q_v^+$	$x$	
	0	1
00	01	11
01	10	00
11	00	10
10	11	01

$Q_y Q_v$

Fig. 7.56

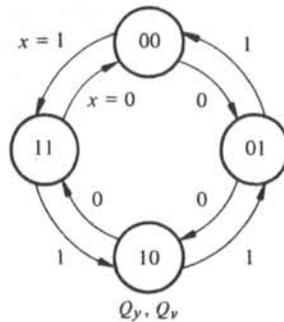


Fig. 7.57

7.5.2 Modèle asynchrone

On admet pour chaque bascule D le modèle asynchrone du paragraphe 7.4.9 en fonctionnement normal ( $PR, CLR = 00$ ). La première bascule est caractérisée par les variables internes  $y_1$  et  $y_2 = Q_y$  et la seconde par  $v_1$  et  $v_2 = Q_v$  de sorte que les équations (7.59), (7.60) et (7.63) deviennent, après quelques transformations :

$$\begin{aligned} y_1^+ &= D_y \cdot \bar{C}\bar{K} \cdot \bar{y}_2 + D_y \cdot y_1 + CK \cdot y_1 + y_1 \bar{y}_2 \\ &= (x \oplus y_2 \oplus v_2) (\bar{C}\bar{K} \cdot \bar{y}_2 + y_1) + CK \cdot y_1 + y_1 \bar{y}_2 \\ &= \bar{C}\bar{K} \bar{x} \bar{y}_2 v_2 + \bar{C}\bar{K} x \bar{y}_2 \bar{v}_2 + CK \cdot y_1 + \\ &\quad x y_1 v_2 + \bar{x} y_1 \bar{v}_2 + \bar{y}_2 y_1 \end{aligned} \tag{7.64}$$

$$y_2^+ = CK \cdot y_1 + \overline{CK} \cdot y_2 \tag{7.65}$$

$$\begin{aligned} v_1^+ &= D_v \cdot \overline{CK} \cdot \overline{v_2} + D_v \cdot v_1 + CK \cdot v_1 + v_1 \cdot \overline{v_2} \\ &= \overline{CK} \cdot \overline{v_2} + CK \cdot v_1 \end{aligned} \tag{7.66}$$

$$v_2^+ = CK \cdot v_1 + \overline{CK} \cdot v_2 \tag{7.67}$$

La table d'états de la figure 7.58 révèle l'existence de courses, mais l'analyse de celles-ci démontre qu'elles ne sont pas critiques.

$y_1^+ \ y_2^+$		$v_1^+ \ v_2^+$		$x = 0$		$x = 1$	
				$x, CK$			
				00	01	11	10
0	0	0	0	00 10	00 00	00 00	10 10
0	0	0	1	10 01	00 00	00 00	00 01
0	0	1	1	10 01	00 11	00 11	00 01
0	0	1	0	00 10	00 11	00 11	10 10
0	1	0	0	01 10	00 00	00 00	01 10
0	1	0	1	01 01	00 00	00 00	01 01
0	1	1	1	01 01	00 11	00 11	01 01
0	1	1	0	01 10	00 11	00 11	01 10
1	1	0	0	11 10	11 00	11 00	01 10
1	1	0	1	01 01	11 00	11 00	11 01
1	1	1	1	01 01	11 11	11 11	11 01
1	1	1	0	11 10	11 11	11 11	01 10
1	0	0	0	10 10	11 00	11 00	10 10
1	0	0	1	10 01	11 00	11 00	10 01
1	0	1	1	10 01	11 11	11 11	10 01
1	0	1	0	10 10	11 11	11 11	10 10

$y_1 \ y_2$        $v_1 \ v_2$   
 $(Q_y)$        $(Q_v)$

Fig. 7.58

■ 7.5.3 Description

Le modèle asynchrone du compteur nous permet d'analyser l'effet d'une variation de l'entrée  $x$  pendant un instant d'horloge. En admettant que la variable  $x$  est égale à 0 pendant plusieurs périodes du signal d'horloge (comptage) et qu'elle prend la valeur 1 (décomptage) dans l'état  $Q_y, Q_v = 01$ , pour  $CK = 0$ , on observe dans la table d'états (fig. 7.58) et dans un chronogramme (fig. 7.59) la séquence des états internes  $y_1, y_2,$

$v_1, v_2$ . On constate alors que l'inversion du sens de comptage, commandée par la variable  $x$ , n'est pas enregistrée par les bascules durant l'instant d'horloge  $t_0$ , mais seulement au cours de l'instant suivant  $t_1$ . Ce retard, mis en évidence dans un graphe partiel (fig. 7.60), est supprimé si l'on remplace la bascule  $D$  du paragraphe 7.4.9 par celle du paragraphe 7.4.7 (bascule commandée par le flanc du signal d'horloge).

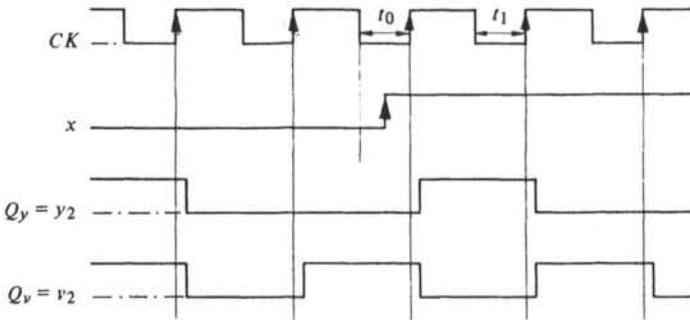


Fig. 7.59

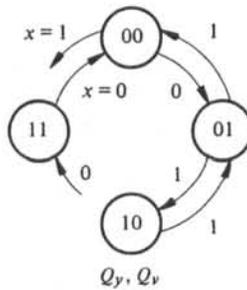


Fig. 7.60

**7.5.4 Conclusion**

L'existence d'une famille de modèles asynchrones pour chaque type de bascule bistable ( $D$  par exemple) entraîne la conséquence suivante : une même séquence d'entrée appliquée à des systèmes séquentiels distincts, mais décrits par le même modèle synchrone ou quasi-synchrone (équations (7.63) par exemple), peut produire des séquences de sortie différentes si elle viole les hypothèses de ces modèles (variations des entrées interdites pour  $CK = 0$  : voir le paragraphe 3.2.3).

**7.5.5 Exercice**

En admettant qu'une seule entrée ( $x$  ou  $CK$ ) varie à la fois, analyser toutes les courses de la figure 7.58 à l'aide du graphe des états et démontrer que celles-ci ne sont pas critiques. Souligner dans la table (fig. 7.58) tous les états totaux du modèle quasi-synchrone (§ 3.2.3), puis représenter dans un chronogramme toutes les séquences des états totaux ne validant pas ce modèle; obtient-on toujours, lors d'une inversion de  $x$  pour  $CK = 0$ , le retard décrit au paragraphe 7.5.3 ?

Un autre modèle asynchrone du compteur réversible (§ 7.5.1) est obtenu si l'on dispose de bascules  $D$  commandées par un flanc (§ 7.4.7); établir les équations et la table d'états, puis souligner dans celle-ci les états totaux du modèle quasi-synchrone (§ 3.2.3). En admettant la même séquence d'entrée  $x, CK$  et les mêmes conditions initiales, obtient-on un chronogramme identique à celui de la figure 7.59 ? Démontrer que ce modèle asynchrone n'est pas équivalent à celui du paragraphe 7.5.2 et qu'une variation de  $x$  pendant que  $CK = 0$  n'entraîne aucun retard dans l'inversion du sens de comptage.

### 7.5.6 Exercice

Déterminer un modèle asynchrone des deux systèmes séquentiels suivants : un compteur réversible (§ 5.5.3) et un préleveur de période (§ 5.5.5; bascules  $Q_3$  et  $Q_4$  de la figure 5.50). Dans les deux cas, on admet l'emploi de la bascule  $JK$  du paragraphe 7.4.6. Établir les équations et la table d'états, souligner dans celle-ci les états totaux du modèle quasi-synchrone (§ 3.2.3) puis comparer les résultats de cette analyse avec ceux des paragraphes 5.5.3 et 5.5.7 (compteur), 5.5.5 et 5.5.8 (préleveur).

### ■ 7.5.7 Exercice

Les trois bascules  $Q_3, Q_4, Q_2$  de la figure 4.34 réalisent un compteur par cinq (§ 4.3.4). En admettant l'emploi des bascules  $SR$  (§ 7.4.3) et  $JK$  (§ 7.4.6) on demande d'établir les équations et la table d'états du modèle asynchrone de ce compteur. Analyser les éventuelles courses (graphes des états), puis comparer le comportement calculé avec celui du paragraphe 4.3.4.

### 7.5.8 Exercice

La figure 7.61 est celle d'un système séquentiel quasi-synchrone. Déterminer un modèle asynchrone en admettant que la bascule  $JK$  est celle du paragraphe 7.4.6 et que la porte NAND présente un délai  $\Delta_3$  sur sa sortie  $y_3$ . Établir une table d'états et analyser les éventuelles courses. Pour quelles valeurs relatives des délais supprime-t-on toute course critique ?

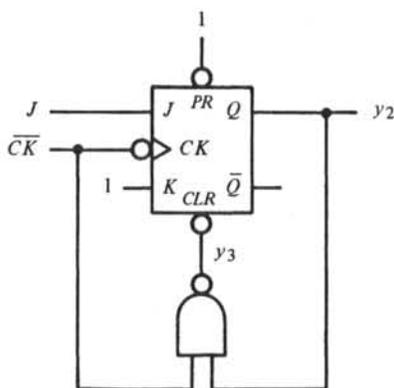


Fig. 7.61

## ■ 7.5.9 Exercice

Les figures 7.62 et 7.63 proposent deux variantes d'un diviseur de fréquence par 3 (§ 4.1.15). Déterminer dans chaque cas un modèle asynchrone à cinq délais en admettant que les bascules  $JK$  sont celles du paragraphe 7.4.6; la porte NAND de la figure 7.62 présente un délai  $\Delta_5$  sur sa sortie  $Z$  tandis que l'élément de mémoire  $\bar{s}\bar{r}$  de la figure 7.63 est décrit par le modèle du paragraphe 7.3.12. Etablir dans les deux cas une table d'états et analyser les éventuelles courses. Pour quelles valeurs relatives des délais obtient-on effectivement un comptage par trois ?

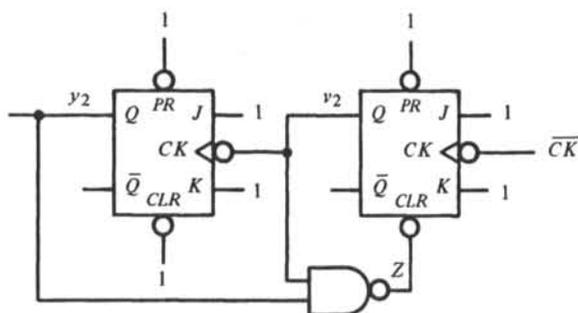


Fig. 7.62

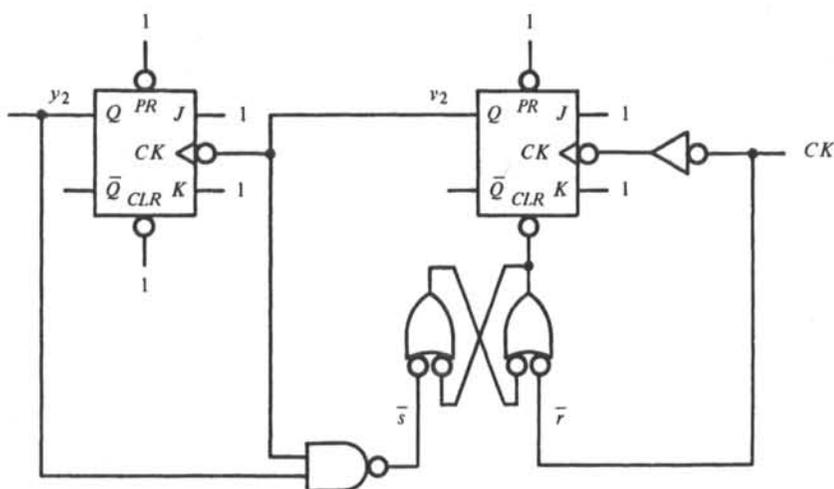


Fig. 7.63



## ANNEXES

## 8.1 SYSTÈMES DE NUMÉRATION

## 8.1.1 Notation

Dans la section 8.1 les opérateurs  $\cdot$  (produit) et  $+$  (somme) sont ceux de l'arithmétique.

## 8.1.2 Définitions : système décimal

Dans l'usage courant un nombre  $N$  écrit sous la forme :

$$N = 1975 \quad (8.1)$$

peut être exprimé par la relation arithmétique équivalente :

$$N = 1 \cdot 10^3 + 9 \cdot 10^2 + 7 \cdot 10^1 + 5 \cdot 10^0 \quad (8.2)$$

On dit alors que le nombre  $N$  est exprimé dans le *système de numération décimal* ou, plus brièvement, que  $N$  est un *nombre décimal*; les puissances successives de 10 sont les *poids* : ces poids sont sous-entendus dans la relation (8.1) et explicités dans l'expression (8.2); le nombre 10 est la *base* du système de numération. Il existe dix symboles distincts (0, 1, ..., 9) qui sont les *chiffres* de ce système.

## 8.1.3 Définitions : système binaire

Un nombre  $N$  peut être exprimé à l'aide d'une somme des puissances successives (ou poids) de 2 : on dit alors que  $N$  est représenté dans le *système de numération binaire* ou, plus brièvement, que  $N$  est un *nombre binaire*; le nombre 2 est la base de ce système qui comporte deux symboles distinctifs 0 et 1 : ce sont les *chiffres binaires* ou *bits*. La notation abrégée d'un nombre binaire  $N$  s'écrit sous la forme :

$$N = 110101 \quad (8.3)$$

qui est équivalente à l'expression arithmétique :

$$N = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \quad (8.4)$$

### 8.1.4 Méthode

La conversion d'un nombre binaire  $N$  dans son équivalent décimal est immédiate : elle découle du calcul (effectué dans la base 10) de l'expression (8.4); pour distinguer les expressions équivalentes de  $N$  dans deux bases différentes, on peut repérer celles-ci par un indice. On écrit alors :

$$N = (110101)_2 = 2^5 + 2^4 + 2^2 + 2^0 = (53)_{10} \quad (8.5)$$

### 8.1.5 Méthode

La conversion d'un nombre décimal  $N$  dans son équivalent binaire peut être effectuée par une succession de divisions par 2 selon l'algorithme suivant :

$$\begin{array}{l}
 N = 53 : 2 = 26, \text{ reste : } 1 \\
 \downarrow \\
 26 : 2 = 13, \text{ reste : } 0 \\
 \downarrow \\
 13 : 2 = 6, \text{ reste : } 1 \\
 \downarrow \\
 6 : 2 = 3, \text{ reste : } 0 \\
 \downarrow \\
 3 : 2 = 1, \text{ reste : } 1 \\
 \downarrow \\
 1 : 2 = 0, \text{ reste : } 1
 \end{array}
 \uparrow$$

Le nombre binaire recherché est formé par les restes successifs de la division, lus de bas en haut [10] (pp. 10 - 13); on retrouve alors :

$$N = (53)_{10} = (110101)_2$$

### 8.1.6 Définition : code binaire pur

Tout nombre décimal peut être exprimé dans le système binaire à l'aide de la méthode du paragraphe 8.1.5 : on appelle *code binaire pur* une telle représentation des nombres décimaux. La figure 8.1 illustre le code binaire pur des nombres décimaux entiers compris entre 0 et 21.

### 8.1.7 Définitions

Il existe une grande variété de représentations binaires des nombres décimaux. La figure 8.1 illustre deux codes très fréquemment utilisés :

- le code *Excédent 3* (en anglais : *Excess 3*) obtenu par une translation verticale du code binaire pur;
- le code *Gray* caractérisé par le changement d'un seul bit lors du passage d'un nombre décimal  $N$  au nombre suivant  $N + 1$ .

On remarque qu'il n'est pas possible d'affecter à chacun des bits de ces deux codes une puissance de 2 pour retrouver le nombre décimal original : ces codes sont *non pondérés*.

Représentation binaire		Système décimal		Représentations binaires		
Base: 10 2 symboles: 0 et 1		Base: 10 10 symboles: 0, 1, ..., 9		Base: 2 2 symboles: 0 et 1	2 symboles: 0 et 1	2 symboles: 0 et 1
Code BCD				Code binaire pur	Code Excédent 3	Code Gray
10 <sup>1</sup> 10 <sup>0</sup>		10 <sup>1</sup> 10 <sup>0</sup>		2 <sup>4</sup> 2 <sup>3</sup> 2 <sup>2</sup> 2 <sup>1</sup> 2 <sup>0</sup>		
0000		0		0 0 0 0	1 1	0 0 0 0
0001		1		0 0 0 1	1 0 0	0 0 0 1
0010		2		0 0 1 0	1 0 1	0 0 1 1
0011		3		0 0 1 1	1 1 0	0 0 1 0
0100		4		0 1 0 0	1 1 1	0 1 1 0
0101		5		0 1 0 1	1 0 0 0	1 1 1 0
0110		6		0 1 1 0	1 0 0 1	1 0 1 0
0111		7		0 1 1 1	1 0 1 0	1 0 1 1
1000		8		1 0 0 0	1 0 1 1	1 0 0 1
1001		9		1 0 0 1	1 1 0 0	1 0 0 0
0001 0000		1 0		1 0 1 0	1 1 0 1	
0001 0001		1 1		1 0 1 1	1 1 1 0	
0001 0010		1 2		1 1 0 0	1 1 1 1	
0001 0011		1 3		1 1 0 1	1 0 0 0 0	
0001 0100		1 4		1 1 1 0	1 0 0 0 1	
0001 0101		1 5		1 1 1 1	1 0 0 1 0	
0001 0110		1 6		1 0 0 0 0	1 0 0 1 1	
0001 0111		1 7		1 0 0 0 1	1 0 1 0 0	
0001 1000		1 8		1 0 0 1 0	1 0 1 0 1	
0001 1001		1 9		1 0 0 1 1	:	
0010 0000		2 0		1 0 1 0 0	:	
0010 0001		2 1		1 0 1 0 1	:	
:		:		:		

Fig. 8.1

### 8.1.8 Définition : code BCD

On constate que chacun des dix chiffres 0 à 9 peut être représenté dans le code binaire pur à l'aide de quatre bits au plus (fig. 8.1); on appelle code *binaire codé décimal* (en anglais : *Binary Coded Decimal* ou *BCD*) la représentation suivante des nombres décimaux :

- chaque chiffre décimal, indépendamment de son poids, est traduit dans le code binaire pur conformément à la figure 8.1;
- chaque nombre binaire ainsi obtenu est normalisé, c'est-à-dire représenté à l'aide de quatre bits; le cas échéant, on complète le nombre binaire (fig. 8.1) par un ou plusieurs 0 placés à gauche.

### 8.1.9 Exemple

Soit à exprimer le nombre décimal  $N = 103$  dans le code BCD :

$$N = 103 = 1 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0 \quad (8.6)$$

avec :

$$(1)_{10} = (1)_2 = (0001)_2 \quad (8.7)$$

$$(0)_{10} = (0)_2 = (0000)_2 \quad (8.8)$$

$$(3)_{10} = (11)_2 = (0011)_2 \quad (8.9)$$

En remplaçant (8.7), (8.8) et (8.9) dans (8.6) on obtient la représentation désirée :

$$N = (0001) (0000) (0011) \quad (8.10)$$

Le nombre  $N$  finalement obtenu (8.10) est caractérisé par :

- une représentation binaire : seuls les symboles 0 et 1 sont effectivement utilisés;
- une structure décimale : à chaque groupe de quatre bits correspond une puissance différente de 10.

La figure 8.1 représente dans le code BCD les nombres décimaux compris entre 0 et 21.

### 8.1.10 Bibliographie

Les différents systèmes de numération, les conversions entre systèmes de base quelconque et les divers codes binaires et BCD feront l'objet d'une étude approfondie dans le volume XIV; on peut également consulter les références [10] (pp. 9 - 21, 181 - 185), [18] (pp. 13 - 26), [12] (pp. 4 - 22) et [14] (pp. 17 - 30).

## 8.2 MÉTHODE DE SIMPLIFICATION DE McCLUSKEY

### 8.2.1 Définition

On appelle *monômes adjacents* deux produits qui sont identiques à l'exception d'une seule variable qui est vraie dans l'un et inversée dans l'autre.

Les monômes suivants sont adjacents :

- $abcd$  et  $abc\bar{d}$ ;
- $ac$  et  $a\bar{c}$ ;
- $\bar{D}BA$  et  $\bar{D}\bar{B}A$ .

### 8.2.2 Propriété

La somme de deux monômes adjacents de  $n$  lettres est égale à un seul *monôme résultant* de  $n-1$  lettres; on dit que les deux monômes adjacents sont inclus dans le monôme résultant.

La propriété découle du théorème (1.10) :  $a + \bar{a} = 1$ ; on a par exemple :

$$abcd + abc\bar{d} = abc(d + \bar{d}) = abc \quad (8.11)$$

$$ac + a\bar{c} = a(c + \bar{c}) = a \quad (8.12)$$

$$\bar{D}BA + \bar{D}\bar{B}A = \bar{D}A(B + \bar{B}) = \bar{D}A \quad (8.13)$$

### 8.2.3 Notation

Tout monôme d'une fonction de  $n$  variables peut être représenté à l'aide d'une notation ternaire comportant les trois symboles 0 (variable inversée), 1 (variable vraie) et - (variable absente). On a par exemple :

- si  $Z = Z(a, b, c, d)$  alors  $abcd = |1111|$ ;  $abc\bar{d} = |1110|$ ;
- si  $Z = Z(a, b, c, d, e)$  alors  $ac = |1-1--|$ ;  $a\bar{c} = |1-0--|$ ;
- si  $Z = Z(D, C, B, A)$  alors  $\bar{D}BA = |0-11|$ ;  $\bar{D}\bar{B}A = |0-01|$ .

Il découle de la définition (§ 8.2.1) et des exemples ci-dessus que deux monômes adjacents sont caractérisés par :

- le même nombre de symboles ternaires (0, 1, -);
- la même position relative des bits (0 ou 1) et des tirets (-);
- tous les bits sont égaux deux à deux à l'exception d'une paire d'entre eux.

Les expressions algébriques (8.11) à (8.13) peuvent s'écrire sous la forme :

$$|1111| + |1110| = |111-| \quad (8.14)$$

$$|1-1--| + |1-0--| = |1----| \quad (8.15)$$

$$|0-11| + |0-01| = |0--1| \quad (8.16)$$

### 8.2.4 Méthode

Les mintermes de la fonction à simplifier sont exprimés dans la notation ternaire (§ 8.2.3). On recherche systématiquement les monômes résultant de la somme de deux monômes adjacents; en poursuivant le calcul de proche en proche on obtient des monômes qui ne sont adjacents à aucun autre et qui, par conséquent, ne sont pas inclus dans des monômes plus grands : ce sont les impliquants premiers de la fonction (§ 2.2.12).

### 8.2.5 Exemple

On cherche à simplifier simultanément les trois fonctions incomplètement définies (2.45), (2.46) et (2.47) du paragraphe 2.5.2 :

$$R(D, C, B, A) = \Sigma 0, 4, 11, 15 + \emptyset 1, 3, 5, 8, 10, 14$$

$$S(D, C, B, A) = \Sigma 0, 4, 5, 13 + \emptyset 2, 3, 6, 7, 9, 14, 15$$

$$T(D, C, B, A) = \Sigma 5, 11, 13, 15 + \emptyset 3, 4, 9, 12$$

Ces trois fonctions sont représentées dans une table de vérité (fig. 8.2) en respectant les deux règles suivantes :

- les états  $D, C, B, A$  sont regroupés en cinq classes numérotées de 0 à 4; on remarque que la représentation de chaque état est identique à la notation ternaire du minterme correspondant (§ 8.2.3); chaque classe est constituée par des mintermes comportant le même nombre de 1;
- les états  $D, C, B, A$  produisant un état de sortie  $R, S, T = 000$  sont éliminés car les mintermes correspondants n'appartiennent à aucune fonction (dans l'exemple traité aucun d'eux n'est supprimé).

Classe	No	$D$ $C$ $B$ $A$	$R$ $S$ $T$	Contrôle
0	0	0 0 0 0	1 1 0	$V$
1	1	0 0 0 1	$\emptyset$ 0 0	$X$
	2	0 0 1 0	0 $\emptyset$ 0	$X$
	4	0 1 0 0	1 1 $\emptyset$	$V$
	8	1 0 0 0	$\emptyset$ 0 0	$X$
2	3	0 0 1 1	$\emptyset$ $\emptyset$ $\emptyset$	$X$
	5	0 1 0 1	$\emptyset$ 1 1	$V$
	6	0 1 1 0	0 $\emptyset$ 0	$X$
	9	1 0 0 1	0 $\emptyset$ $\emptyset$	$X$
	10	1 0 1 0	$\emptyset$ 0 0	$X$
	12	1 1 0 0	0 0 $\emptyset$	$X$
3	7	0 1 1 1	0 $\emptyset$ 0	$X$
	11	1 0 1 1	1 0 1	$V$
	13	1 1 0 1	0 1 1	$V$
	14	1 1 1 0	$\emptyset$ $\emptyset$ 0	$X$
4	15	1 1 1 1	1 $\emptyset$ 1	$V$

Fig. 8.2

#### ■ 8.2.6 Première étape

On recherche tous les monômes adjacents de la figure 8.2. Grâce au classement effectué dans la table de vérité, deux monômes (mintermes) adjacents sont situés dans deux classes voisines (0 et 1, 1 et 2, 2 et 3, 3 et 4) : une confrontation systématique produit les monômes résultants de la figure 8.3 qui sont représentés dans la notation

ternaire; chacun d'eux est désigné par les deux numéros des mintermes qu'il inclut. On remarque alors que :

- les monômes obtenus comportent trois lettres; ils sont regroupés en quatre classes (0-1, 1-2, 2-3 et 3-4); chaque classe inclut des monômes ayant le même nombre de 1;
- l'état de sortie  $R, S, T$  d'un monôme résultant est déterminé par les états de sortie des deux monômes adjacents selon la règle de la figure 8.5; tous les monômes dont l'état  $R, S, T = 000$  sont éliminés car ils n'appartiennent à aucune fonction.

La règle de formation des états de sortie (fig. 8.5) peut être justifiée en considérant deux monômes adjacents d'une seule fonction  $Z$  :

- si l'un au moins des monômes adjacents n'est pas inclus dans la fonction  $Z$  ( $Z = 0$ ), alors le monôme résultant ne peut être inclus dans  $Z$  ( $Z = 0$ );
- si l'un des monômes adjacents est inclus dans la borne inférieure de  $Z$  ( $Z = 1$ )

Classe	No	$D$ $C$ $B$ $A$	$R$ $S$ $T$	Contrôle
0-1	0,1	0 0 0 -	1 0 0	$V$
	0,2	0 0 - 0	0 1 0	$V$
	0,4	0 - 0 0	1 1 0	$a$
	0,8	- 0 0 0	1 0 0	$b$
1-2	1,3	0 0 - 1	$\emptyset$ 0 0	$X$
	1,5	0 - 0 1	$\emptyset$ 0 0	$X$
	2,3	0 0 1 -	0 $\emptyset$ 0	$X$
	2,6	0 - 1 0	0 $\emptyset$ 0	$X$
	4,5	0 1 0 -	1 1 1	$c$
	4,6	0 1 - 0	0 1 0	$V$
	4,12	- 1 0 0	0 0 $\emptyset$	$X$
	8,10	1 0 - 0	$\emptyset$ 0 0	$X$
2-3	3,7	0 - 1 1	0 $\emptyset$ 0	$X$
	3,11	- 0 1 1	1 0 1	$d$
	5,7	0 1 - 1	0 1 0	$V$
	5,13	- 1 0 1	0 1 1	$e$
	6,7	0 1 1 -	0 $\emptyset$ 0	$X$
	6,14	- 1 1 0	0 $\emptyset$ 0	$X$
	9,11	1 0 - 1	0 0 1	$V$
	9,13	1 - 0 1	0 1 1	$f$
	10,11	1 0 1 -	1 0 0	$V$
	10,14	1 - 1 0	$\emptyset$ 0 0	$X$
	12,13	1 1 0 -	0 0 1	$V$
	3-4	7,15	- 1 1 1	0 $\emptyset$ 0
11,15		1 - 1 1	1 0 1	$g$
13,15		1 1 - 1	0 1 1	$h$
14,15		1 1 1 -	1 $\emptyset$ 0	$V$

Fig. 8.3

Classe	No	D C B A	R S T	Contrôle
0-1-2	0, 1, 4, 5	0 - 0 -	1 0 0	<i>j</i>
	0, 2, 4, 6	0 - - 0	0 1 0	<i>k</i>
	0, 4, 1, 5	0 - 0 -	1 0 0	*
	0, 4, 2, 6	0 - - 0	0 1 0	*
1-2-3	2, 3, 6, 7	0 - 1 -	0 $\emptyset$ 0	<i>X</i>
	2, 6, 3, 7	0 - 1 -	0 $\emptyset$ 0	*
	4, 5, 6, 7	0 1 - -	0 1 0	<i>l</i>
	4, 5, 12, 13	- 1 0 -	0 0 1	<i>m</i>
	4, 6, 5, 7	0 1 - -	0 1 0	*
	4, 12, 5, 13	- 1 0 -	0 0 1	*
2-3-4	5, 7, 13, 15	- 1 - 1	0 1 0	<i>n</i>
	5, 13, 7, 15	- 1 - 1	0 1 0	*
	6, 7, 14, 15	- 1 1 -	0 $\emptyset$ 0	<i>X</i>
	6, 14, 7, 15	- 1 1 -	0 $\emptyset$ 0	*
	9, 11, 13, 15	1 - - 1	0 0 1	<i>p</i>
	9, 13, 11, 15	1 - - 1	0 0 1	*
	10, 11, 14, 15	1 - 1 -	1 0 0	<i>q</i>
	10, 14, 11, 15	1 - 1 -	1 0 0	*

Fig. 8.4

	Etat de sortie
1er monôme	0 0 0 1 1 1 $\emptyset$ $\emptyset$ $\emptyset$
2ème monôme	0 1 $\emptyset$ 0 1 $\emptyset$ 0 1 $\emptyset$
Monôme résultant	0 0 0 0 1 1 0 1 $\emptyset$

Fig. 8.5

et que l'autre monôme est inclus dans cette borne inférieure ( $Z = 1$ ) ou dans les états  $\emptyset$  de  $Z$  ( $Z = \emptyset$ ), alors le monôme résultant est inclus totalement ou partiellement dans la borne inférieure de  $Z$  ( $Z = 1$ );

- si les deux monômes adjacents sont inclus dans les états  $\emptyset$  de  $Z$  ( $Z = \emptyset$ ), alors le monôme résultant est également inclus dans ces états et n'est pas inclus dans la borne inférieure de  $Z$  ( $Z = \emptyset$ ).

### 8.2.7 Deuxième étape

On détecte les monômes adjacents de la figure 8.3 en comparant systématiquement tous les monômes de deux classes voisines; on obtient les monômes résultants de la figure 8.4 dans laquelle :

- chaque monôme comporte deux lettres;
- par la nature même de l'algorithme, chaque monôme apparaît deux fois.

### 8.2.8 Troisième étape

On recherche enfin les monômes adjacents de la figure 8.4; tous les monômes résultants produisent un état de sortie  $R, S, T = 000$  : la recherche de ces monômes est donc terminée.

### ■ 8.2.9 Quatrième étape

Avant de dresser une table des impliquants premiers, on cherche à éliminer tous les monômes des figures 8.2, 8.3 et 8.4 qui ne sont pas des impliquants premiers des trois fonctions ou de leurs produits, ainsi que tous les impliquants superflus.

On supprime ainsi :

- tous les monômes calculés à double dans la figure 8.4 : ils sont marqués d'un astérisque (\*);
- tous les monômes dont l'état de sortie  $R, S, T$  ne comporte aucun 1; ces monômes (dont l'état de sortie comporte au moins un symbole  $\emptyset$ ) sont inclus dans les états  $\emptyset$  d'une ou plusieurs fonctions mais ne sont inclus dans aucune borne inférieure de ces mêmes fonctions : ils sont marqués du symbole  $X$  (fig. 8.2, 8.3 et 8.4);
- tous les monômes adjacents appartenant à  $K$  fonctions (c'est-à-dire dont l'état de sortie comporte  $K - 1$ ) à condition que le monôme résultant appartienne à ces  $K$  fonctions; le monôme  $|000 - |$ , par exemple, appartient à la fonction  $R$ , de même que le monôme  $|0 - 0 - |$  dans lequel il est inclus :  $|000 - |$  peut être éliminé au profit de  $|0 - 0 - |$  qui est un impliquant premier de  $R$ ; par contre, le monôme  $|0 - 00|$  ( $R, S, T = 110$ ) est inclus dans les monômes  $|0 - 0 - |$  ( $R, S, T = 100$ ) et  $|0 - - 0|$  ( $R, S, T = 010$ ) :  $|0 - 00|$  n'est pas éliminé car il est impliquant premier de  $R \cdot S$ ; les suppressions ainsi effectuées sont marquées du symbole  $V$  (fig. 8.2 et 8.3).

Après l'élimination des monômes marqués des signes \*,  $X$  et  $V$ , il reste à considérer quinze impliquants premiers  $a, b, \dots, q$ .

### 8.2.10 Cinquième étape

Conformément à la théorie des fonctions incomplètement définies (§ 2.3.9) on cherche la couverture des bornes inférieures des trois fonctions  $R, S$  et  $T$ . On reporte alors dans la table des impliquants premiers (fig. 8.6) les douze mintermes des trois bornes inférieures  $R_m, S_m$  et  $T_m$  et les quinze impliquants premiers  $a, b, \dots, q$ ; chacun de ces impliquants couvre la ou les fonctions auxquelles il appartient selon la valeur de son état de sortie  $R, S, T$ .

La nécessité de couvrir simultanément les trois fonctions  $R_m, S_m$  et  $T_m$  à l'aide d'un sous-ensemble minimal d'impliquants premiers peut se traduire par une proposition logique  $P$  :

$$P = P_R \cdot P_S \cdot P_T \quad (8.17)$$

où  $P_R, P_S$  et  $P_T$  sont les propositions logiques relatives à chaque fonction  $R_m, S_m$  et  $T_m$ ; la lecture de la figure 8.6 donne alors :

Fonctions	No	a	b	c	d	e	f	g	h	j	k	l	m	n	p	q
$R_m$	0	1	1							1						
	4	1		1						1						
	11				1			1								1
	15							1								1
$S_m$	0	1									1					
	4	1		1							1					
	5			1		1						1		1		
	13					1	1		1					1		
$T_m$	5			1		1							1			
	11				1			1								1
	13					1	1		1				1			1
	15							1	1							1

Fig. 8.6

$$P_R = 0 \cdot 4 \cdot 11 \cdot 15 = (a + b + j)(a + c + j)(d + g + q)(g + q) \quad (8.18)$$

$$P_S = 0 \cdot 4 \cdot 5 \cdot 13 = (a + k)(a + c + k + l)(c + e + l + n)(e + f + h + n) \quad (8.19)$$

$$P_T = 5 \cdot 11 \cdot 13 \cdot 15 = (c + e + m)(d + g + p)(e + f + h + m + p)(g + h + p) \quad (8.20)$$

En utilisant la relation  $a(a + b) = a$  (1.18) on calcule successivement :

$$P_R = (a + b + j)(a + c + j)(g + q) \quad (8.21)$$

$$P_S = (a + k)(c + e + l + n)(e + f + h + n) \quad (8.22)$$

En introduisant (8.21), (8.22) et (8.20) dans (8.17) on obtient  $P$  :

$$P = (a + b + j)(a + c + j)(e + f + h + n)(e + f + h + m + p)(g + h + p)(d + g + p)(c + e + l + n)(c + e + m)(g + q)(a + k) \quad (8.23)$$

qui devient, en effectuant les parenthèses et en utilisant la relation  $(a + b)(a + c) = a + bc$  (1.19) :

$$P = (ag + apq + ahdq + jkg + jkpq + jkhdq + bckg + bckpq + bckhdq)(e + mn + fc + fml + hc + hml + npc) \quad (8.24)$$

La simplification optimale découle d'un sous-ensemble minimal d'impliquants premiers : elle implique la recherche d'un monôme de  $P$  comportant un nombre minimal de lettres. Sans effectuer complètement le calcul de (8.24) on peut détecter un tel monôme :

$$P_m = a \cdot g \cdot e \quad (8.25)$$

Il est donc possible de réaliser les fonctions  $R$ ,  $S$  et  $T$  à l'aide des trois impliquants  $a$ ,  $g$  et  $e$ ; leur répartition dans chacune des trois fonctions est déterminée par les relations (8.18), (8.19) et (8.20) dans lesquelles on donne la valeur 0 à tous les impliquants, à l'exception de  $a$ ,  $g$  et  $e$  :

$$P_R = a \cdot a \cdot g \cdot g = a \cdot g \quad (8.26)$$

$$P_S = a \cdot a \cdot e \cdot e = a \cdot e \quad (8.27)$$

$$P_T = e \cdot g \cdot e \cdot g = e \cdot g \quad (8.28)$$

Il en découle la forme simplifiée des trois fonctions étudiées :

$$R = a + g = \bar{D}\bar{B}\bar{A} + DBA \quad (8.29)$$

$$S = a + e = \bar{D}\bar{B}\bar{A} + C\bar{B}A \quad (8.30)$$

$$T = e + g = DBA + C\bar{B}A \quad (8.31)$$

### 8.2.11 Conclusion

Les solutions (8.29) à (8.31) vérifient les expressions (2.51) à (2.53) représentées sur les tables de Karnaugh de la figure 2.62.

La complexité de l'algorithme proposé le prédestine au calcul automatique; le calcul manuel de la proposition logique  $P$  est particulièrement fastidieux. L'emploi d'un système logique spécialisé (§ 8.2.12) permet de résoudre simplement le problème posé.

### □ 8.2.12 Description d'un système logique spécialisé

Le calcul de la proposition logique  $P$  (8.17) se ramène à la recherche des solutions de l'expression :

$$P(a, b, \dots, q) = P_R \cdot P_S \cdot P_T = 1 \quad (8.32)$$

On peut réaliser la fonction  $P(a, b, \dots, q)$  : tout état  $a, b, \dots, q$  vérifiant la relation (8.32) est une solution du problème et fournit un ensemble d'impliquants premiers couvrant  $R$ ,  $S$  et  $T$ . On peut concevoir un système logique spécialisé constitué (fig. 8.7) :

- d'un système combinatoire réalisant l'expression (8.23) de  $P$ ;
- d'un compteur (chap. 4) à quinze sorties  $a, b, \dots, q$ .

Le compteur doit parcourir les  $2^{15}$  états  $a, b, \dots, q$  pour donner l'ensemble des solutions de  $P$ ; en fait, seules les solutions incluant un nombre minimal d'impliquants nous intéressent : elles correspondent aux états  $a, b, \dots, q$  qui comportent un nombre minimal de 1. Il est donc avantageux de prévoir un code de comptage analogue à celui de la figure 8.8.

### 8.2.13 Bibliographie

La méthode de McCluskey [17] s'applique également à la simplification d'une seule fonction, complètement définie [18] (pp. 65 - 72), [10] (pp. 139 - 157), [11]

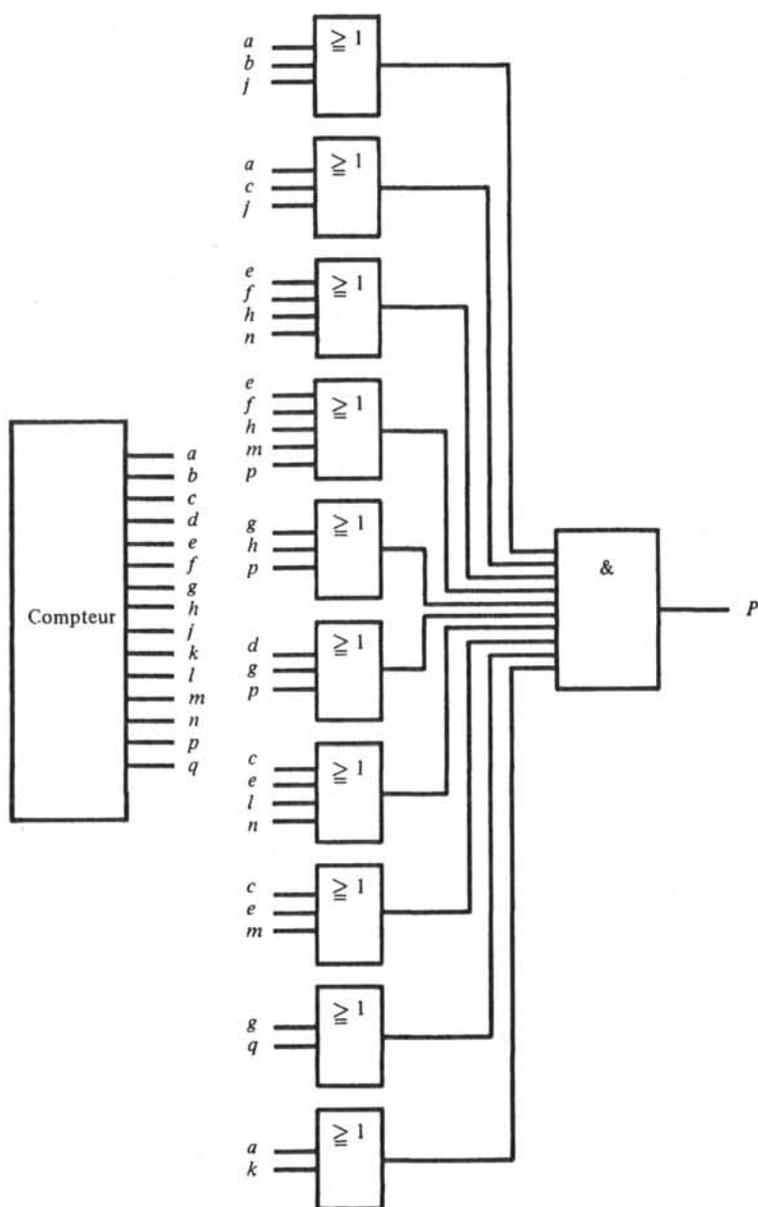


Fig. 8.7

(pp. 141 - 149), [14] (pp. 106 - 109) et [99] (pp. 131 - 167). L'extension aux fonctions incomplètement définies fait l'objet des références [18] (pp. 72 - 76), [10] (pp. 157 - 158), [11] (pp. 150 - 151) tandis que l'application aux fonctions multiples,

$a, b, c, d, \dots$	$m, n, p, q$	
0 0 0 0	... 0 0 0 0	}
0 0 0 0	... 0 0 0 1	}
0 0 0 0	... 0 0 1 0	}
0 0 0 0	... 0 1 0 0	}
0 0 0 0	... 1 0 0 0	}
	⋮	
0 0 0 1	... 0 0 0 0	}
0 0 1 0	... 0 0 0 0	}
0 1 0 0	... 0 0 0 0	}
1 0 0 0	... 0 0 0 0	}
	⋮	
0 0 0 0	... 0 0 1 1	}
0 0 0 0	... 0 1 0 1	}
0 0 0 0	... 1 0 0 1	}
	⋮	
1 0 1 0	... 0 0 0 0	}
1 1 0 0	... 0 0 0 0	}
	⋮	

$\downarrow$   
 $+t$

Fig. 8.8

traitée dans cette section, est due à Bartee [27] et se trouve notamment exposée dans les références [10] (pp. 166 - 167), [11] (pp. 151 - 154), [18] (pp. 83 - 88) et [100] (pp. 174 - 197).



## SOLUTIONS DES EXERCICES

### CHAPITRE 1

**1.3.16**  $Z_1 = ab ; Z_2 = a + b$

**1.3.17** Par (1.12) :

$$ab + \bar{a} = (a + \bar{a})(b + \bar{a}) = b + \bar{a} = \bar{a} + b \quad (1.21)$$

Par (1.16) :

$$ab + \bar{a} = ab + \bar{a} + b = b(1 + a) + \bar{a} = \bar{a} + b \quad (1.21)$$

**1.4.12**

$$\overline{abcd} = \overline{x \cdot d} = \bar{x} + \bar{d} \quad \text{avec } x = abc ;$$

d'où :

$$\bar{x} = \overline{abc} = \overline{y \cdot c} = \bar{y} + \bar{c} \quad \text{avec } y = ab ;$$

d'où :

$$\bar{y} = \overline{ab} = \bar{a} + \bar{b} ; \quad \bar{x} = \bar{a} + \bar{b} + \bar{c} ; \quad \overline{abcd} = \bar{a} + \bar{b} + \bar{c} + \bar{d}$$

**1.4.13**  $Z_1 = \overline{acde} + \overline{acdf} + \overline{bcde} + \overline{bcd\bar{f}}$

**1.4.14**  $\overline{abc} = [(a \downarrow 0) \downarrow (b \downarrow 0) \downarrow (c \downarrow 0)] \downarrow 0$

**1.4.18**

$$Z_1 = [(a \uparrow b) \uparrow 1] \uparrow [(c \uparrow d) \uparrow 1] \quad \text{ou} \quad Z_1 = \{ \{ \{ (a \uparrow b) \uparrow 1 \} \uparrow c \} \uparrow 1 \} \uparrow d$$

Réalisation de (1.37) avec des portes NOR : voir figure 1.72.

**1.4.19** La figure 1.73 nous donne :

$$\begin{aligned} Z &= [(K + \bar{M})B + \bar{D}] \cdot \bar{A} \cdot [\bar{S} + T + \bar{B}] + RP \\ &= [\bar{A}BK + \bar{A}B\bar{M} + \bar{A}\bar{D}] \cdot [\bar{S} + T + \bar{B}] + RP \\ &= \bar{A}BK\bar{S} + \bar{A}BK T + \bar{A}B\bar{M}\bar{S} + \bar{A}B\bar{M} T + \bar{A}\bar{D}\bar{S} + \bar{A}\bar{D} T + \bar{A}\bar{B}\bar{D} + RP \end{aligned}$$

**1.4.20**

$$Z_1 = a + b\bar{c} + de + df$$

$$Z_2 = \bar{A} + B + D\bar{K}L + DK\bar{M} + DLPQ$$

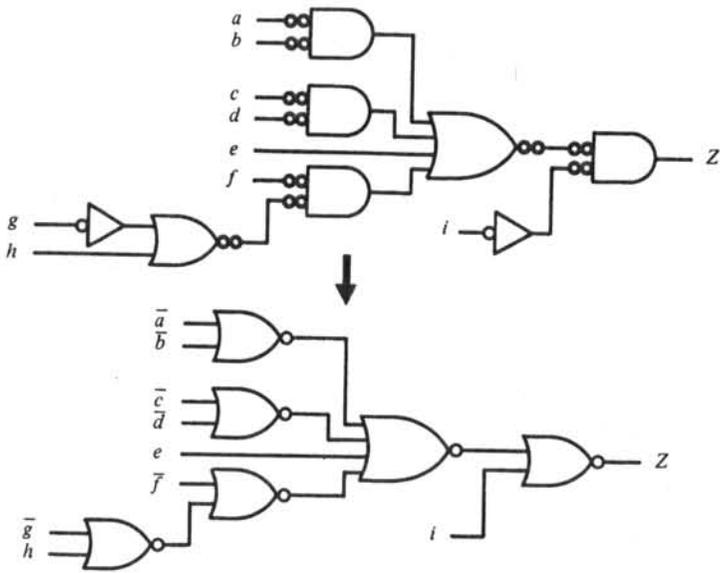


Fig. 1.72

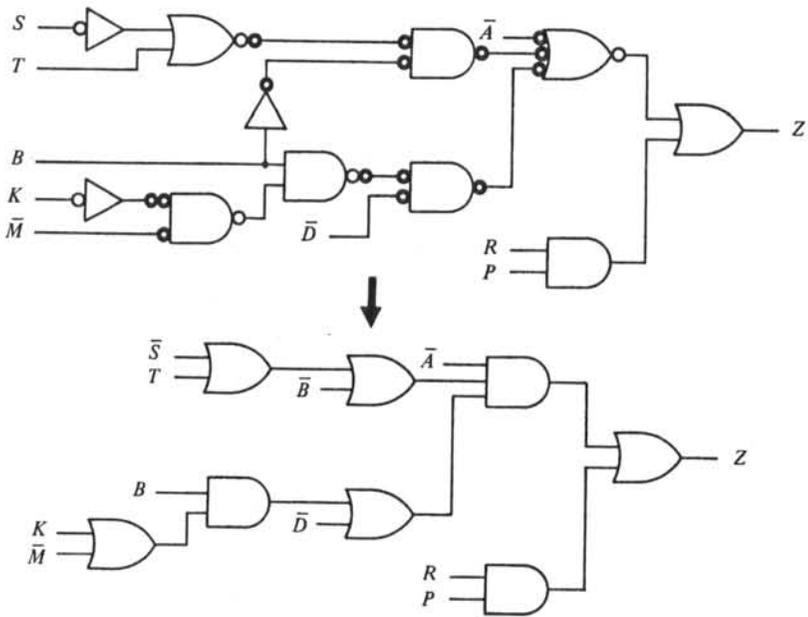


Fig. 1.73

1.5.11 La réalisation de  $Z_a$ ,  $Z_b$  et  $Z_c$  est faite avec l'assemblage de la figure 1.74.

1.5.12 Voir figure 1.75.

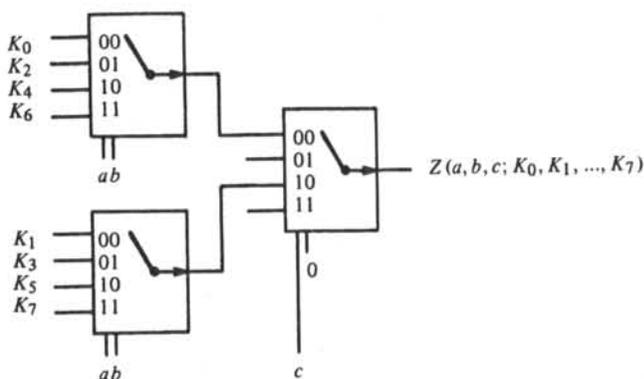


Fig. 1.74

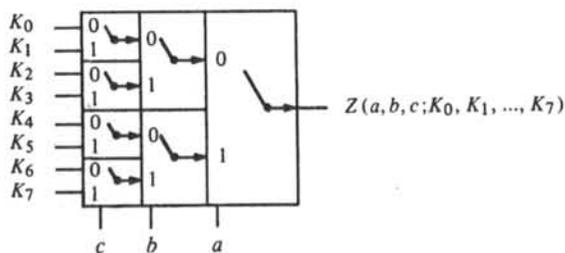


Fig. 1.75

## 1.5.13

$$Z(a, b; K_0, K_1, K_2, K_3) \\ = (K_0 + a + b)(K_1 + a + \bar{b})(K_2 + \bar{a} + b)(K_3 + \bar{a} + \bar{b})$$

## 1.6.9

$$(a \oplus b)c = ac \oplus bc; (a \oplus b) + c \neq (a + c) \oplus (b + c); \\ (ab) \oplus c \neq (a \oplus c)(b \oplus c); (a + b) \oplus c \neq (a \oplus c) + (b \oplus c); \\ a \oplus b = (a + b) \cdot \bar{a}\bar{b}$$

1.6.11 Voir figure 1.76 pour  $Z_2$ ;  $Z_3 = ab \oplus c$ .

1.6.12 Voir figure 1.77.

## 1.6.13

$$y_0 = K_0; y_1 = K_0 \oplus K_2; \\ y_2 = K_0 \oplus K_1; y_3 = K_0 \oplus K_1 \oplus K_2 \oplus K_3$$

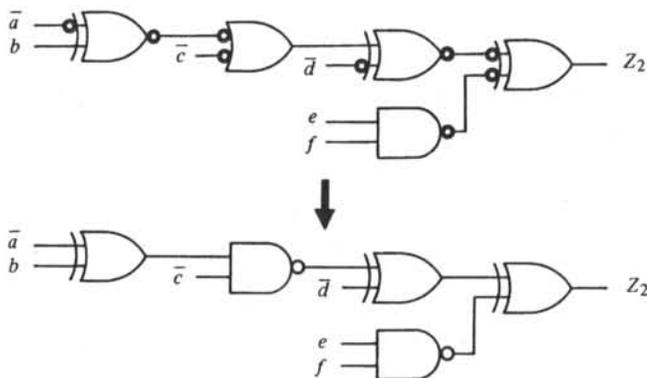


Fig. 1.76

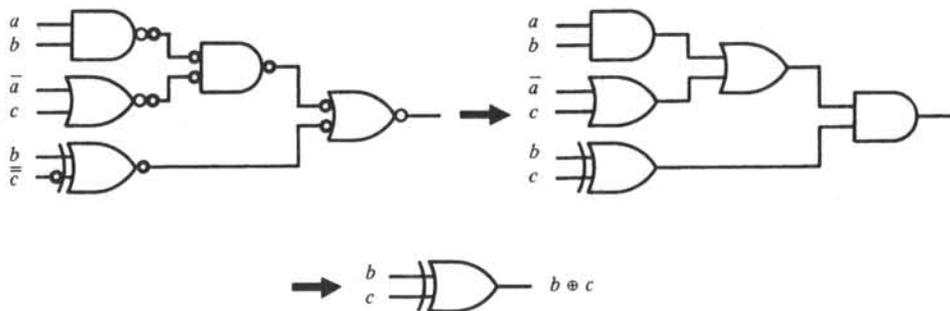


Fig. 1.77

## 1.6.20

$$Z_1 = MAJ(a, b, 0) \oplus 0 ; Z_2 = MAJ(a, b, 1) \oplus 0 ;$$

$$Z_3 = MAJ(a, b, 0) \oplus 1 ; Z_4 = MAJ(a, b, 1) \oplus 1 ;$$

$$Z_5 = MAJ(a, b, c) \oplus 1$$

## 1.7.15

$$\overline{MAJ(a, b, c)} \in \overline{M(a, b, c, d)} ; \overline{MAJ(a, b, d)} \in \overline{M(a, b, c, d)} ;$$

$$\overline{MAJ(a, c, d)} \in \overline{M(a, b, c, d)} ; \overline{MAJ(b, c, d)} \in \overline{M(a, b, c, d)}$$

## 1.7.16

$$M_0(a, b, c, d) = \Sigma 6, 10, 12$$

$$\lambda(a, b, c, d) = \Sigma 6, 10, 12 + \emptyset 0, 1, 2, 4, 7, 8, 11, 13, 14, 15$$

1.8.10 Opérateurs complets :  $Z_2, Z_4, Z_8, Z_{11}, Z_{13}, Z_{14}$ .

## CHAPITRE 2

2.1.18

$$\begin{aligned}
 & DCB + DC\overline{A} + \overline{D}C\overline{B} + \overline{B}A + C\overline{A} \\
 &= DCB + DC\overline{A} + DC + \overline{D}C\overline{B} + \overline{B}A + C\overline{A} + CB \\
 &= DC(B + A + 1) + \overline{D}C\overline{B} + \overline{C}B + \overline{B}A + C\overline{A} \\
 &= DC + \overline{D}C + \overline{D}C\overline{B} + \overline{C}B + \overline{B}A + C\overline{A} \\
 &= DC + \overline{D}C(1 + \overline{B}) + \overline{C}B + \overline{B}A + C\overline{A} = \overline{D}C + \overline{D}C + \overline{C}B + \overline{B}A + C\overline{A} \\
 &= C + DC + \overline{D}C + \overline{C}B + C\overline{A} + \overline{B}A = C(1 + D + \overline{D} + B + \overline{A}) + \overline{B}A \\
 &= C + \overline{B}A
 \end{aligned}$$

où  $\overline{x} \overline{\overline{x}}$  met en évidence les variables  $x$  et  $\overline{x}$  du théorème du consensus (1.16).

2.1.19 5 (fig. 1.40); 6 (fig. 1.42); 6 (fig. 1.43).

2.1.20  $\overline{Z} = \overline{C}\overline{B} + \overline{C}\overline{A}$

2.2.7 Non.

2.2.8

$$\begin{aligned}
 Z_1 &= \overline{D}C\overline{A} ; Z_2 = \overline{C}B\overline{A} ; Z_3 = \overline{B}A ; Z_4 = D\overline{C} ; \\
 Z_5 &= \overline{C}B ; Z_6 = D\overline{A} ; Z_7 = \overline{C} ; Z_8 = \overline{D} ; Z_9 = 1
 \end{aligned}$$

2.2.10 Cette démonstration repose sur l'emploi du théorème  $a + \overline{a} = 1$  (1.10) et sur le fait que deux cases voisines de la table de Karnaugh (disposées en ligne ou en colonne, mais pas en diagonale) représentent des états qui ne diffèrent que d'un seul bit.

2.2.24

$$\begin{aligned}
 Z_1 &= C + \overline{B}A ; Z_2 = a + b ; Z_3 = ab + ac + bc ; \\
 Z_4 &= \overline{a}\overline{c} + ac + c\overline{d} = \overline{a}\overline{c} + ac + \overline{a}\overline{d} ; \\
 Z_5 &= abc + \overline{a}bd ; Z_6 = ab + ac + bd
 \end{aligned}$$

2.2.25

$$\begin{aligned}
 Z_1 &= DC\overline{B} + DBA + \overline{D}C\overline{B} + \overline{D}\overline{B}A ; Z_2 = \overline{D}\overline{B}\overline{A} + C\overline{B}A + DBA + \overline{C}\overline{B}\overline{A} ; \\
 Z_3 &= \overline{C}\overline{B}A + DBA + \overline{C}\overline{B}\overline{A} ; Z_4 = \overline{D}C + DA + \overline{D}\overline{A} + D\overline{C}\overline{B}
 \end{aligned}$$

2.2.26  $Z = \overline{c}\overline{d} + \overline{a}b\overline{c} + a\overline{b}\overline{d} + abcd$

2.2.27 Si les états de sortie sont définis par  $(G, P) = 01$  pour  $A < B$ ,  $(G, P) = 10$  pour  $A > B$  et  $(G, P) = 00$  pour  $A = B$  on obtient :

$$G = A_1 \overline{B}_1 + A_0 \overline{B}_1 \overline{B}_0 + A_1 A_0 \overline{B}_0 ; P = \overline{A}_1 B_1 + \overline{A}_0 B_1 B_0 + \overline{A}_1 \overline{A}_0 B_0$$

2.2.28 Par exemple :

$$Z = D\bar{B}A + \bar{D}BA + DC\bar{A} + \bar{C}\bar{B}A + \bar{C}B\bar{A}$$

2.2.35

$$Z_2 = ab + \bar{b}\bar{c} ; Z_3 = \bar{E}\bar{D}\bar{B}\bar{A} + D\bar{C}\bar{B}A + DCB\bar{A} ;$$

$$Z_4 = EC\bar{B} + DBA + \bar{D}\bar{C}\bar{B}\bar{A} + \bar{E}\bar{C}\bar{B}\bar{A} ;$$

$$Z_7 = ED\bar{B} + F\bar{D}CB + F\bar{D}\bar{C}\bar{B}\bar{A} + \bar{E}\bar{D}\bar{C}\bar{B}\bar{A}$$

2.2.37

$$\begin{aligned} MAJ(E, D, C, B, A) = & EDC + EDB + EDA + ECB + ECA + EBA \\ & + DCB + DCA + DBA + CBA \end{aligned}$$

2.3.13

$$M = ad + bc = ab + cd = ac + bd ;$$

$$Z_1 = \bar{D}\bar{B} + CA + B\bar{A} = \bar{D}\bar{A} + CA + B\bar{A} ;$$

$$Z_2 = C\bar{A} + \bar{C}A + \bar{D}B ; Z_3 = \bar{D}\bar{C}B + \bar{D}C\bar{B} + DCB + D\bar{C}\bar{B} \text{ (par exemple).}$$

2.3.14

$$Z_1 = D + A ; Z_2 = E\bar{C} + DA + \bar{C}\bar{A} + \bar{E}\bar{D}\bar{B} + E\bar{D}B\bar{A} ;$$

$$Z_4 = \bar{b}\bar{c} + \bar{b}\bar{d} + bcd ; Z_5 = \bar{d} + \bar{a}\bar{b} + \bar{a}c + \bar{b}c$$

2.4.6

$$\overline{\bar{A} + C\bar{B} + \bar{D}B} \neq DA + \bar{C}\bar{B}A$$

2.5.7 Voir les paragraphes 8.2.5 à 8.2.12.

2.5.8

$$Z_1 = abc\bar{c} + \bar{b}c\bar{d} + a\bar{b}c + \bar{a}b\bar{c}\bar{d} ; Z_2 = a\bar{b}c + \bar{b}c\bar{d} + \bar{a}b\bar{c}\bar{d} ;$$

$$Z_3 = \bar{a}\bar{b}d + \bar{b}c\bar{d} + a\bar{b}c + abc\bar{d} \text{ [10] (pp. 161 - 166).}$$

2.5.9 En admettant les hypothèses de la solution du paragraphe 2.2.27, il n'existe aucun impliquant premier commun à  $G$  et  $P$ .

2.6.11

$$Z_3 = D ; Z_2 = D \oplus C ; Z_1 = D \oplus C \oplus B ; Z_0 = D \oplus C \oplus B \oplus A$$

2.6.13

$$\bar{E} = (a \oplus b \oplus c \oplus d) + bcd + \bar{b}\bar{c}\bar{d} = (a \oplus b \oplus c \oplus d) + (\bar{a}\bar{c} \oplus bd) ;$$

$$Z_1 = a \oplus b \oplus c ; Z_2 = (a \oplus c)(\overline{b \oplus d}) + ac(b \oplus d) = (a \oplus b \oplus c \oplus d)(a + c)$$

2.7.7

$$E = (\overline{A_n \oplus B_n}) \cdot (\overline{A_{n-1} \oplus B_{n-1}}) \cdot \dots \cdot (\overline{A_2 \oplus B_2}) \cdot (\overline{A_1 \oplus B_1}) \cdot (\overline{A_0 \oplus B_0})$$

2.7.8 Structure itérative :

$$R_n = A_n B_n + (A_n \oplus B_n) R_{n-1} = MAJ(A_n, B_n, R_{n-1}); S_n = A_n \oplus B_n \oplus R_{n-1}$$

2.7.9 Si les états de sortie sont définis par  $(G, P) = 01$  pour  $A < B$ ,  $(G, P) = 10$  pour  $A > B$  et  $(G, P) = 00$  pour  $A = B$  on obtient avec  $0 \leq j \leq n$  :

- en commençant par les bits de poids fort :

$$G_j = G_{j+1} + \overline{P}_{j+1} A_j \overline{B}_j; P_j = P_{j+1} + \overline{G}_{j+1} \overline{A}_j B_j$$

- en commençant par les bits de poids faible :

$$G_j = A_j \overline{B}_j + G_{j-1} A_j + G_{j-1} \overline{B}_j = MAJ(A_j, \overline{B}_j, G_{j-1})$$

$$P_j = \overline{A}_j B_j + P_{j-1} \overline{A}_j + P_{j-1} B_j = MAJ(\overline{A}_j, B_j, P_{j-1})$$

## CHAPITRE 3

3.1.13 Le logigramme de la figure 3.7 est équivalent à celui de la figure 3.1 si les sorties  $y_1$  et  $y_2$  sont inversées (avec  $a \equiv s$  et  $b \equiv r$ ).

3.2.15 Voir figure 3.51.

3.3.12 Si  $J \cdot K = 0$ .

3.3.13

$$Q^+ = X \overline{V} + \overline{Y} Q + W Q; X Y \overline{V} \overline{W} = 0$$

- Bascule JK :  $X = J, Y = K, V = Q, W = \overline{Q}$ ;

- bascule T :  $X = Y = T, V = Q, W = \overline{Q}$ ;

- bascule D :  $X = Y = 1, \overline{V} = W = D$ .

3.3.14

$$D = J \overline{Q} + \overline{K} Q; D = \overline{Q}$$

$$J = D; K = \overline{D}$$

3.3.16 Voir figure 3.52.

3.3.17

$$Q^+ = S \overline{R} + S Q + \overline{R} Q = MAJ(S, \overline{R}, Q); Q^+ = \overline{S R} + \overline{R} Q$$

3.4.10 Voir figure 3.53.

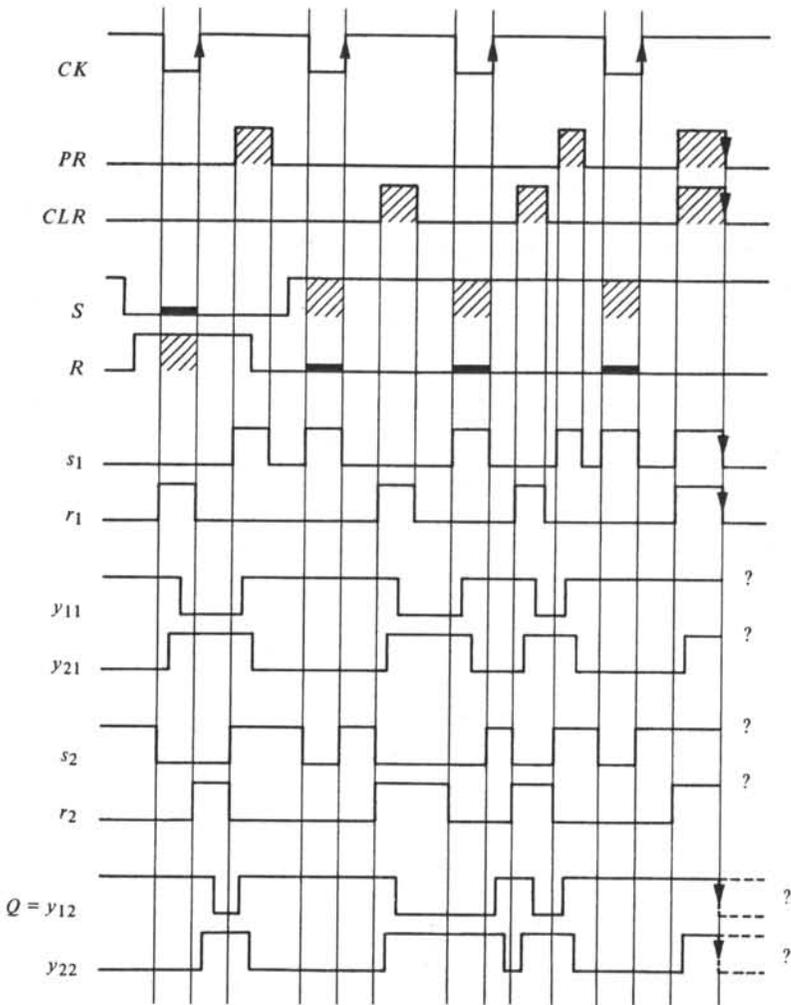


Fig. 3.51

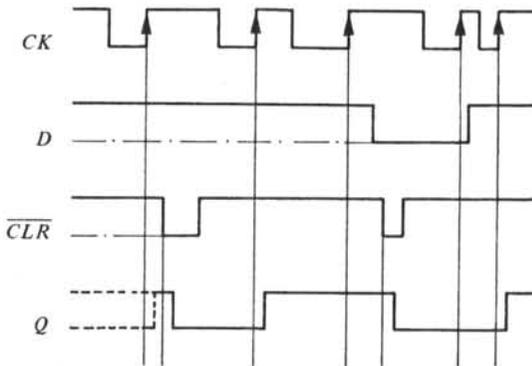


Fig. 3.52

No	$Q$	$Q^+$	$X$	$Y$	$V$	$W$	$v$	$w$	$L$	$P$	$R$	$S$	$T$	$x$	$y$	$X$
0	0	0	$\left\{ \begin{array}{l} 0 \ \emptyset \ \emptyset \ \emptyset \\ \emptyset \ \emptyset \ 1 \ \emptyset \end{array} \right.$				$\left\{ \begin{array}{l} 0 \ \emptyset \\ \emptyset \ 1 \end{array} \right.$		$\left\{ \begin{array}{l} 0 \ \emptyset \\ \emptyset \ 0 \end{array} \right.$		$\left\{ \begin{array}{l} \emptyset \ 0 \ 0 \\ \emptyset \ 0 \ 0 \end{array} \right.$			$\left\{ \begin{array}{l} 0 \ \emptyset \\ \emptyset \ 1 \end{array} \right.$		$\emptyset$
1	0	1	$\left\{ \begin{array}{l} 1 \ 0 \ 0 \ \emptyset \\ 1 \ \emptyset \ 0 \ 1 \end{array} \right.$				$\left\{ \begin{array}{l} 1 \ 0 \\ 1 \ 1 \end{array} \right.$				$\left\{ \begin{array}{l} 0 \ 0 \ 1 \\ 0 \ 1 \ 0 \end{array} \right.$			$\left\{ \begin{array}{l} 1 \ 0 \\ 1 \ 0 \end{array} \right.$		-
2	1	0	$\left\{ \begin{array}{l} 0 \ 1 \ \emptyset \ 0 \\ \emptyset \ 1 \ 1 \ 0 \end{array} \right.$				$\left\{ \begin{array}{l} 0 \ 1 \\ 1 \ 0 \end{array} \right.$		$\left\{ \begin{array}{l} 1 \ 0 \\ 1 \ 0 \end{array} \right.$		$\left\{ \begin{array}{l} 0 \ 0 \ 1 \\ 1 \ 0 \ 0 \end{array} \right.$			$\left\{ \begin{array}{l} \emptyset \ 1 \\ \emptyset \ 1 \end{array} \right.$		0
3	1	1	$\left\{ \begin{array}{l} \emptyset \ 0 \ \emptyset \ \emptyset \\ \emptyset \ \emptyset \ \emptyset \ 1 \end{array} \right.$				$\left\{ \begin{array}{l} \emptyset \ 0 \\ 1 \ \emptyset \end{array} \right.$		$\left\{ \begin{array}{l} 0 \ \emptyset \\ \emptyset \ 1 \end{array} \right.$		$\left\{ \begin{array}{l} 0 \ \emptyset \ 0 \\ 0 \ \emptyset \ 0 \end{array} \right.$			$\left\{ \begin{array}{l} \emptyset \ 0 \\ \emptyset \ 0 \end{array} \right.$		1

Fig. 3.53

3.4.12 Par (3.64) et (3.59) on a :

$$Q^+ = Q_m^+ + \lambda \cdot Q_M^+ = T \oplus Q$$

$$\begin{aligned} T &= Q \oplus Q^+ = Q\bar{Q}^+ + \bar{Q}Q^+ = Q(\overline{Q_m^+ + \lambda \cdot Q_M^+}) + \bar{Q}(Q_m^+ + \lambda \cdot Q_M^+) \\ &= \bar{Q}Q_m^+ + Q\bar{Q}_m^+ + \lambda \cdot \bar{Q}Q_M^+ + \bar{\lambda} \cdot Q\bar{Q}_m^+ \end{aligned}$$

Un diagramme de Venn (fig. 3.54) donne la forme simplifiée de  $T$  :

$$T = \bar{Q}Q_m^+ + Q\bar{Q}_m^+ + \lambda \cdot \bar{Q}_m^+Q_M^+$$

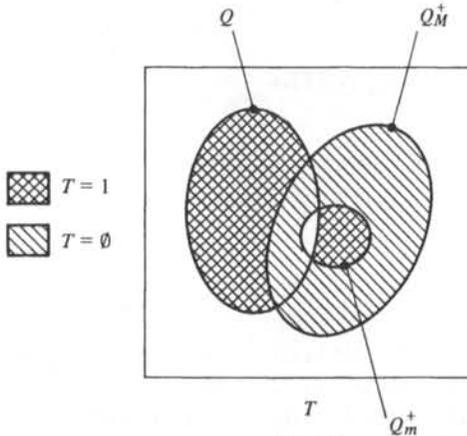


Fig. 3.54

3.4.13 Si  $S, R = 11$  est admis, on obtient :

$$S = \bar{Q}Q_m^+ + \lambda_S \cdot Q_M^+ ; R = Q\bar{Q}_m^+ + \lambda_R \cdot \bar{Q}_m^+$$

3.4.14 Réalisation :

- d'une bascule  $D$  avec une bascule  $JK$  :

$$Q^+ = D ; J = \bar{Q}D + \lambda_J \cdot Q \ni D ; K = Q\bar{D} + \lambda_K \cdot \bar{Q} \ni \bar{D}$$

- d'une bascule  $LP$  avec une bascule  $JK$  :

$$Q^+ = LP + \bar{L}Q \quad ; \quad J = \bar{Q}(LP + \bar{L}Q) + \lambda_J \cdot Q \ni LP \quad ; \\ K = Q(\overline{LP + \bar{L}Q}) + \lambda_K \cdot \bar{Q} \ni L\bar{P}$$

- d'une bascule  $JK$  avec une bascule  $SR$  (§ 3.4.13) :

$$Q^+ = J\bar{Q} + \bar{K}Q \quad ; \\ S = \bar{Q}Q^+ + \lambda_S \cdot Q^+ = (J\bar{Q} + \bar{K}Q)(\bar{Q} + \lambda_S) = J\bar{Q} + \lambda_S \cdot \bar{K}Q \ni J\bar{Q} \quad ; \\ R = Q\bar{Q}^+ + \lambda_R \cdot \bar{Q}^+ = (\overline{J\bar{Q} + \bar{K}Q})(Q + \lambda_R) = \bar{J}KQ + KQ + \lambda_R(\bar{J}K + \bar{J}\bar{Q}) \ni KQ$$

- d'une bascule  $T$  avec une bascule  $SR$  (§ 3.4.13) :

$$Q^+ = T \oplus Q \quad ; \\ S = \bar{Q}Q^+ + \lambda_S \cdot Q^+ = (T \oplus Q)(\bar{Q} + \lambda_S) = T\bar{Q} + \lambda_S \cdot \bar{T}Q \ni T\bar{Q} \quad ; \\ R = Q\bar{Q}^+ + \lambda_R \cdot \bar{Q}^+ = (T \oplus \bar{Q})(Q + \lambda_R) = TQ + \lambda_R \cdot \bar{T}\bar{Q} \ni TQ$$

- d'une bascule  $SR$  avec une bascule  $T$  (§ 3.4.12) :

$$Q^+ = Q_m^+ + \lambda \cdot Q_M^+ = (\overline{S\bar{R}} + \bar{R}Q) + \lambda(S + \bar{R}Q) \quad ; \\ T = \bar{Q}(\overline{S\bar{R}} + \bar{R}Q) + Q(\overline{S + \bar{R}Q}) + \lambda(\overline{S\bar{R}} + \bar{R}Q)(S + \bar{R}Q) \\ = \overline{S\bar{R}}\bar{Q} + \bar{S}RQ + \lambda \cdot SR \ni \overline{S\bar{R}} + RQ$$

## CHAPITRE 4

**4.1.11** Bascules  $D$  avec  $D = \bar{Q}$ . Modèle quasi-synchrone toujours valable pour bascules  $JK$  :  $J = K = \bar{P}\bar{R} = \bar{C}\bar{L}\bar{R} = 1$  ; toujours valable pour bascules  $D$  :  $D = \bar{Q}$  avec  $Q$  invariant pour  $CK = 0$  et  $\bar{P}\bar{R} = \bar{C}\bar{L}\bar{R} = 1$ .

### 4.1.13

$$Q_8, Q_4, Q_2, Q_1 = 0000 \rightarrow (0001 \rightarrow 0011 \rightarrow 0111 \rightarrow) 1111 \rightarrow 1110 \rightarrow (1111 \rightarrow) \\ 1101 \rightarrow 1100 \rightarrow (1101 \rightarrow 1111 \rightarrow) 1011 \rightarrow 1010 \rightarrow (1011 \rightarrow) \\ 1001 \rightarrow 1000 \rightarrow (1001 \rightarrow 1011 \rightarrow 1111 \rightarrow) 0111 \rightarrow 0110 \rightarrow \\ (0111 \rightarrow) 0101 \rightarrow 0100 \rightarrow (0101 \rightarrow 0111 \rightarrow) 0011 \rightarrow 0010 \rightarrow \\ (0011 \rightarrow) 0001 \rightarrow 0000$$

Les états entre parenthèses sont transitoires.

**4.1.14** Figure 4.6 :  $Q_B, Q_A = 00 \rightarrow (01 \rightarrow) 11 \rightarrow 10 \rightarrow (11 \rightarrow) 01 \rightarrow 00$  ; nombre des états permanents : 4 ; rapport de division : 3. Les états entre parenthèses sont transitoires.

**4.1.22**  $J = K = 1$  ;  $\bar{P}\bar{R} = 1$  ;  $\bar{C}\bar{L}\bar{R}_8 = \bar{C}\bar{L}\bar{R}_4 = \bar{C}\bar{L}\bar{R}_2 = \bar{C}\bar{L}\bar{R}_1 = \bar{Z}_{10}$  : d'après la figure 4.10, il existe une variation de  $\bar{C}\bar{L}\bar{R}_4 = \bar{Z}_{10}$  pour  $CK_4 = \bar{Q}_2 = 0$  ; le modèle quasi-synchrone n'est pas vérifié dans ce cas.

4.1.23  $\overline{CLR}_1 = 1$  possible car  $Q_1 = 0$  lorsque  $Z_{10} = 1$ ;  $\overline{CLR}_2 = \overline{CLR}_4 = \overline{CLR}_8 = \overline{Z}_{10}$  car  $Q_2 = Q_8 = 1$  lorsque  $Z_{10} = 1$  et  $Q_4 = 0$  avec  $CK_4 = \overline{Q}_2 = 0 \rightarrow 1$  lorsque  $Z_{10} = 1 \rightarrow 0$ .

4.1.24  $Z = Q_8 Q_4 Q_1 = CLR_4 = PR_2$  (selon conventions des figures 4.2 et 4.9).

4.1.25 Figure 4.15 :  $Q_3, Q_2, Q_1 = 000 \rightarrow 001 \rightarrow (000 \rightarrow) 010 \rightarrow 011 \rightarrow (010 \rightarrow 000 \rightarrow) 100 \rightarrow 000$ ; nombre des états permanents : 5; rapport de division : 4.

Figure 4.16 :  $Q_2, Q_1 = 00 \rightarrow 01 \rightarrow (00 \rightarrow) 10 \rightarrow 00$ ; nombre des états permanents : 3; rapport de division : 2.

Figure 4.17 :  $Q_3, Q_2, Q_1 = 000 \rightarrow 001 \rightarrow (000 \rightarrow) 010 \rightarrow (011 \rightarrow 000 \rightarrow) 100 \rightarrow 101 \rightarrow (100 \rightarrow) 110 \rightarrow (111 \rightarrow 100 \rightarrow) 000$ ; nombre des états permanents : 6; rapport de division : 5.

Les états entre parenthèses sont transitoires.

#### 4.2.13

$$J_1 = K_1 = 1 \quad ; \quad J_2 = K_2 = Q_1 \equiv Q_1 \cdot J_1 \quad ;$$

$$J_4 = K_4 = Q_2 Q_1 \equiv Q_2 \cdot J_2 \quad ; \quad J_8 = K_8 = Q_4 Q_2 Q_1 \equiv Q_4 \cdot J_4 \quad ;$$

structure itérative :  $J_j = K_j = Q_{j-1} \cdot J_{j-1}$

$$D_1 = \overline{Q}_1 \equiv Q_1 \oplus 1 \quad ; \quad D_2 = Q_2 \oplus Q_1 \quad ;$$

$$D_4 = Q_4 \oplus Q_2 Q_1 \quad ; \quad D_8 = Q_8 \oplus Q_4 Q_2 Q_1 \quad ;$$

structure itérative (§ 2.7.8) :

$$R_n = A_n \cdot R_{n-1} \equiv Q_n \cdot R_{n-1} \quad ;$$

$$S_n \equiv D_n = A_n \oplus R_{n-1} \equiv Q_n \oplus R_{n-1}$$

4.2.15 Voir la référence [34] (pp. 117 - 132).

4.2.25 Modèle quasi-synchrone toujours valable pour bascules  $JK$  :  $J$  et  $K$  sont fonctions de  $Q_8, Q_4, Q_2, Q_1$  invariantes pour  $CK = 0$ ;  $PR = CLR = 0$ .  $Z = Q_8 \overline{Q}_1$  ne présente pas de comportement transitoire semblable à celui de la figure 4.3.

#### 4.2.26

$$J_4 = Q_2 Q_1; K_4 = 1; J_2 = K_2 = Q_1; J_1 = \overline{Q}_4; K_1 = 1.$$

Etats  $\emptyset$  :  $101 \rightarrow 010$ ;  $110 \rightarrow 010$ ;  $111 \rightarrow 000$ .

$$D_4 = Q_2 Q_1; D_2 = Q_2 \oplus Q_1; D_1 = \overline{Q}_4 \overline{Q}_1.$$

Etats  $\emptyset$  :  $101 \rightarrow 010$ ;  $110 \rightarrow 010$ ;  $111 \rightarrow 100$ .

4.2.27 Par exemple :  $D_D = \overline{Q}_A + Q_D \overline{Q}_B$ .

4.2.28 Compteur par 10 à bascules  $T$  :  $Q_1, Q_2, Q_3, Q_4 = 0000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 1101 \rightarrow 1100 \rightarrow 0101 \rightarrow 0110 \rightarrow 0111 \rightarrow 0000$ ; états  $\emptyset$  :  $1000 \rightarrow 1001 \rightarrow 1000$  ;  $1010 \rightarrow 1011 \rightarrow 1110 \rightarrow 1111 \rightarrow 1010$ .

Compteur par 10 à bascules  $JK$  :  $Q_3, Q_2, Q_1, Q_0 = 0000 \rightarrow 1011 \rightarrow 0100 \rightarrow 1001 \rightarrow 0110 \rightarrow 0001 \rightarrow 1010 \rightarrow 0101 \rightarrow 1000 \rightarrow 0111 \rightarrow 0000$ ; états  $\emptyset$  :  $0010 \rightarrow 0001 \rightarrow 0011 \rightarrow 0000$  ;  $1100 \rightarrow 0001$  ;  $1101 \rightarrow 0000$  ;  $1110 \rightarrow 0001$  ;  $1111 \rightarrow 0000$ .

4.2.29 Une solution possible avec des bascules  $SR$  :

$$S_C = Q_A \quad ; \quad R_C = Q_C \quad ; \quad S_B = Q_C \quad ; \quad R_B = Q_B \quad ;$$

$$S_A = Q_B \quad ; \quad R_A = Q_A$$

Etats  $\emptyset$  :  $000 \rightarrow 000$  ;  $011 \rightarrow 10\emptyset$  ;  $101 \rightarrow \emptyset 10$  ;  $110 \rightarrow 0\emptyset 1$  ;  $111 \rightarrow \emptyset\emptyset\emptyset$ .

4.2.30

$L_4 = Q_4 + Q_2, P_4 = Q_1$  (autres solutions :  $L_4 = Q_4 + Q_1, P_4 = Q_2$  ;  $L_4 = Q_2 + \bar{Q}_1, P_4 = Q_1$ ) ;  $L_2 = Q_1, P_2 = \bar{Q}_2$  ;  $L_1 = \bar{Q}_4, P_1 = \bar{Q}_1$ .

4.2.32 Figure 4.32 (bascules  $T$ ) :

$$T_C = \bar{Q}_A (Q_C \oplus Q_B) \quad ; \quad T_B = Q_A (\bar{Q}_C \oplus \bar{Q}_B) \quad ; \quad T_A = \overline{T_C + T_B} = \bar{T}_C \cdot \bar{T}_B$$

Figure 4.33 (bascules  $T$ ) :

$$T_D = T_C + T_B \quad ; \quad T_C = \bar{Q}_D \bar{Q}_A \quad ; \quad T_B = Q_D Q_A \quad ; \quad T_A = \bar{T}_D$$

4.3.5  $Q_8, Q_4, Q_2 = 101 \rightarrow (000 \rightarrow) 010$  ;  $110 \rightarrow 010$  ;  $111 \rightarrow (\emptyset 10 \rightarrow) \emptyset 00$

Les états transitoires sont entre parenthèses.

4.3.6 Pour  $(R_{01}, R_{02}, R_{91}, R_{92}) = (Q_2, Q_1, 0, \emptyset) : p = 3$  ;  $(Q_4, Q_1, 0, \emptyset) : p = 5$  ;  $(0, \emptyset, Q_4, Q_2) : p = 7$ .

Avec  $R_0 = R_{01} \cdot R_{02}$  et  $R_9 = R_{91} \cdot R_{92}$  on obtient pour  $p = 4, 6, 8, 9$  :  $(R_0, R_9) = (Q_4, 0)$  ;  $(Q_4 \cdot Q_2, 0)$  ;  $(Q_8, 0)$  ;  $(Q_8 \cdot Q_1, 0)$ .

4.3.7 Figure 4.38 :  $p = 10$  ; figure 4.39 :  $p = 3$  ; figure 4.40, nombre des états permanents : 6, rapport de division : 5.

4.4.7 Si les facteurs  $p_1, p_2, \dots, p_j$  ne sont pas premiers entre eux, la mesure de la décomposition en parallèle est égale au plus grand de ces facteurs.

4.4.9 Compteur par 5 (§ 4.2.26) :  $J_4 = Q_2 Q_1$  ;  $K_4 = 1$  ;  $J_2 = K_2 = Q_1$  ;  $J_1 = \bar{Q}_4$  ;  $K_1 = 1$ . Compteur par 2 :  $J = K = 1$ .  $Y = Q_4 Q$  (pas de régime transitoire indésirable).  $Q_4, Q_2, Q_1, Q = 0000 \rightarrow 0011 \rightarrow 0100 \rightarrow 0111 \rightarrow 1000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0101 \rightarrow 0110 \rightarrow 1001 \rightarrow 0000$ ; états  $\emptyset$  :  $1010 \rightarrow 0101, 1011 \rightarrow 0100, 1100 \rightarrow 0101, 1101 \rightarrow 0100, 1110 \rightarrow 0001, 1111 \rightarrow 0000$ .

4.4.11 Pour  $Q_C, Q_B, Q_A = 111 \rightarrow 110 \rightarrow 101 \rightarrow 011 \rightarrow 000 \rightarrow 111$  et  $Q = 0 \rightarrow 1 \rightarrow 0$  on obtient :  $Q_C, Q_B, Q_A, Q = 1110 \rightarrow 1111 \rightarrow 1100 \rightarrow 1101 \rightarrow 1010 \rightarrow 1011 \rightarrow 0110 \rightarrow 0111 \rightarrow 0000 \rightarrow 0001 \rightarrow 1110$  (fig. 4.43 avec  $CK_A = \bar{Q}$ );  $Q_C, Q_B, Q_A, Q = 1110 \rightarrow 1100 \rightarrow 1010 \rightarrow 0111 \rightarrow 0001 \rightarrow 1111 \rightarrow 1101 \rightarrow 1011 \rightarrow 0110 \rightarrow 0000 \rightarrow 1110$  (fig. 4.44 avec  $CK = \bar{Q}_C$ );  $Q_C, Q_B, Q_A, Q = 1110 \rightarrow 1101 \rightarrow 1010 \rightarrow 0111 \rightarrow 0000 \rightarrow 1111 \rightarrow 1100 \rightarrow 1011 \rightarrow 0110 \rightarrow 0001 \rightarrow 1110$  (fig. 4.49).

## 4.4.16

- Pour  $12 = (3 \cdot 2 \cdot 2) = (6 \cdot 2) = (3 \cdot 4)$  : code binaire pur.
- Pour  $12 = [4 \cdot 3] = [(2 \cdot 2) \cdot 3]$  (fig. 4.54) :  $Q_2, Q_1, Q_B, Q_A = 0000 \rightarrow 0101 \rightarrow 1010 \rightarrow 0011 \rightarrow 0100 \rightarrow 1001 \rightarrow 0010 \rightarrow 0111 \rightarrow 1000 \rightarrow 0001 \rightarrow 0110 \rightarrow 1011 \rightarrow 0000$ .
- Pour  $12 = ([3 \cdot 2] \cdot 2)$  (fig. 4.55 avec  $CK_1 = CK_B = \bar{Q}_A$ ) :  $Q_2, Q_1, Q_B, Q_A = 0000 \rightarrow 0001 \rightarrow 0110 \rightarrow 0111 \rightarrow 1000 \rightarrow 1001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 0101 \rightarrow 1010 \rightarrow 1011 \rightarrow 0000$ .

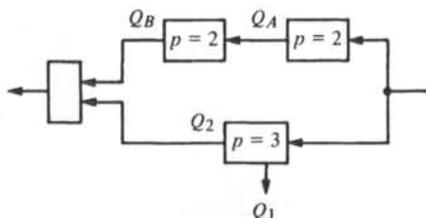


Fig. 4.54

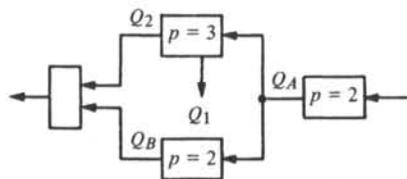


Fig. 4.55

4.4.17 Voir figure 4.56.

Mesure $p$	Décompositions	Nombre de bascules $m$	Observations
2	—	1	$p$ premier
3	—	2	$p$ premier
4	$(2 \cdot 2)$	2	
5	—	3	$p$ premier
6	$(3 \cdot 2) = (2 \cdot 3) = [3 \cdot 2]$	3	
7	—	3	$p$ premier
8	$(2 \cdot 2 \cdot 2) = (4 \cdot 2) = (2 \cdot 4)$	3	
9	$(3 \cdot 3)$	4	
10	$(5 \cdot 2) = (2 \cdot 5) = [5 \cdot 2]$	4	
11	—	4	$p$ premier
12	$(6 \cdot 2) = (2 \cdot 6) = (4 \cdot 3) = (3 \cdot 4) = (3 \cdot 2 \cdot 2)$ $= (2 \cdot 3 \cdot 2) = (2 \cdot 2 \cdot 3) = [4 \cdot 3] = [(3 \cdot 2) \cdot 2]$ $= (2 \cdot [3 \cdot 2]) = [(2 \cdot 2) \cdot 3]$	4	§ 4.4.14
13	—	4	$p$ premier
14	$(7 \cdot 2) = (2 \cdot 7) = [7 \cdot 2]$	4	
15	$(5 \cdot 3) = (3 \cdot 5) = [5 \cdot 3]$	5	4 bascules sans décomposition
16	$(2 \cdot 2 \cdot 2 \cdot 2) = (4 \cdot 2 \cdot 2) = (2 \cdot 4 \cdot 2) = (2 \cdot 2 \cdot 4)$ $= (4 \cdot 4) = (2 \cdot 8) = (8 \cdot 2)$	4	

Fig. 4.56

## CHAPITRE 5

**5.1.10** A condition que toute variation de  $x$  soit interdite pour  $CK = 0$ . Système séquentiel synchrone quelconque :  $PR = CLR = 0$ ; les fonctions d'excitation dépendent de  $x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m$  et sont invariantes pour  $CK = 0$  si  $x_1, x_2, \dots, x_n$  sont elles-mêmes invariantes ( $Q_1 \equiv y_1, Q_2 \equiv y_2, \dots, Q_m \equiv y_m$  étant constantes pour  $CK = 0$ ).

**5.1.11** Par exemple :  $D_C = \bar{Q}_C \bar{Q}_B \bar{Q}_A + \bar{Q}_C Q_B x + \bar{Q}_C Q_A \bar{x}$ .

**5.1.12** Figure 5.9 : compteur synchrone réversible par 10 dont le code est, pour  $x = 1$  :  $Q_1, Q_2, Q_3, Q_4 = 0000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 1101 \rightarrow 1100 \rightarrow 0101 \rightarrow 0110 \rightarrow 0111 \rightarrow 0000$ .

Figure 5.10 : compteur synchrone réversible par 16 dont le code des états  $Q_8, Q_4, Q_2, Q_1$  est, pour  $x = 1$ , binaire pur.

**5.1.13** Décomposition en parallèle (fig. 5.11) :  $Q_1 = 0 \rightarrow 1 \rightarrow 0$ ;  $Q_8, Q_4, Q_2 = 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 000$  (pour  $x = 0$ ); alors  $Q_8, Q_4, Q_2, Q_1 = 0000 \rightarrow 0011 \rightarrow 0100 \rightarrow 0111 \rightarrow 1000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0101 \rightarrow 0110 \rightarrow 1001 \rightarrow 0000$  (pour  $x = 0$ );  $Y = Q_8 Q_1$ .

**5.2.11** A condition que toute variation de  $x_1$  et  $x_2$  soit interdite pour  $CK = 0$ . La fréquence du signal d'horloge doit être supérieure (éventuellement égale) à celle des signaux d'entrée  $x_1$  et  $x_2$ .

**5.2.12** "Veuillez jouer de l'Orgue et éteindre l'Encens jusqu'à ce que le Rire et le Chant cessent, puis rallumer l'Encens à ce moment-là et le laisser brûler définitivement."

**5.2.13** Par exemple :  $pfp \rightarrow fff \rightarrow pfp \rightarrow fff$ .

**5.2.14** Les équations (5.12) et (5.13) réalisent le discriminateur du sens de rotation décrit par les graphes des figures 5.17 (avec  $a \equiv 11, b \equiv 10, c \equiv 01, d \equiv 00$ ) et 6.4 (à l'exception de certaines valeurs de la sortie  $Z$  et des états  $\emptyset$ ).

**5.2.15** Les équations (5.14) et (5.15) réalisent notamment le détecteur de séquence décrit par le graphe de la figure 6.31 et par la table de la figure 6.33.

**5.2.16** Voir la solution de l'exercice du paragraphe 4.2.13 (bascules  $D$  et structure itérative); la généralisation est possible.

**5.3.7** Voir aussi l'exercice du paragraphe 6.5.7.

**5.3.8** Figure 5.27 (compteur par 6) :  $x = \bar{Q}_A$ ;  $x = \bar{Q}_C Q_A + \bar{Q}_B \bar{Q}_A$ .

**5.3.9** Compteur par 15.

**5.3.10** Pour  $x_1, x_2 = 00, 01, 10$  ou  $11$  alors  $p = 15, 13, 14$  ou  $12$ .

5.3.11 Pour  $P_D, P_C, P_B, P_A = 0000$  ou  $1111$  alors  $p = 5$  ou  $1$ .

5.3.12 Par exemple :

$$x_1 = Q_D \bar{Q}_C + \bar{Q}_B \bar{Q}_A + \bar{Q}_D Q_C Q_A$$

$$x_2 = (\bar{Q}_D \oplus Q_A) + \bar{Q}_D \bar{Q}_B$$

5.4.6

$$R_j = A_j B_j + (A_j \oplus B_j) R_{j-1} = MAJ(A_j, B_j, R_{j-1}); S_j = A_j \oplus B_j \oplus R_{j-1}$$

5.4.7 Voir la solution de l'exercice du paragraphe 2.7.9.

5.5.7 Voir figure 5.55.

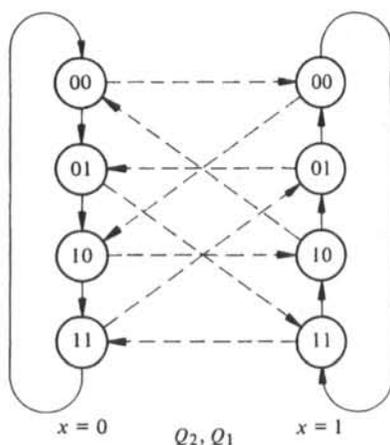


Fig. 5.55

5.5.8 Bascule  $Q_3$  : si  $\bar{Q}_4 = \overline{CLR}_3$  ne varie pas pour  $\bar{x} = 0$ ; bascule  $Q_4$  : si  $Q_3 = J_4$  ne varie pas pour  $\bar{Q}_2 = 0$ .

## CHAPITRE 6

6.1.16 Voir la solution de l'exercice du paragraphe 6.2.11.

6.1.17 Système logique combinatoire :  $Z = x_1 \oplus x_2 \oplus x_3$ .

6.1.18 Voir la solution de l'exercice du paragraphe 6.2.11.

6.1.19 Voir la solution de l'exercice du paragraphe 6.2.11.

6.1.20 Table d'états réduite : voir la figure 6.59.

6.1.21 Voir la solution de l'exercice du paragraphe 6.2.11.

$Y^*, Z$	$x_1 x_2$			
	00	01	11	10
$a$	$(a,0)$	$(\bar{a},0)$	$b,\emptyset$	$c,0$
$b$	$a,\emptyset$	$d,1$	$(b,1)$	$(b,1)$
$c$	$a,0$	$(c,0)$	$(c,0)$	$(c,0)$
$d$	$(d,1)$	$(d,1)$	$b,1$	$(d,1)$
$Y$				

Fig. 6.59

6.1.22 Voir la solution de l'exercice du paragraphe 6.2.11.

6.1.23 Voir la solution de l'exercice du paragraphe 6.2.11.

6.1.24 Table d'états réduite : voir la figure 6.60.

$Y^*, Z_1, Z_2$	$x_1 x_2 x_3$							
	000	001	011	010	100	101	111	110
$a$	-	$(a,00)$	$(a,00)$	$(a,00)$	-	$b,\emptyset$	-	-
$b$	-	$(b,10)$	$(b,01)$	$a,0\emptyset$	-	$(b,10)$	-	-
$Y$								

Fig. 6.60

6.1.26 Table d'états : voir la figure 6.61.

$Y^*, Z$	$x_1 x_2 x_3$			
	000	001	010	100
1	$(1,0)$	$(1,0)$	$(1,0)$	$2,0$
2	$3,0$	-	-	$(2,0)$
3	$(3,0)$	$1,0$	$6,0$	$4,0$
4	$5,0$	-	-	$(4,0)$
5	$(5,0)$	$1,0$	$6,0$	$(5,1)$
6	$7,0$	-	$(6,0)$	-
7	$(7,0)$	$1,0$	$1,0$	$8,\emptyset$
8	$3,\emptyset$	-	-	$(8,1)$
$Y$				

Fig. 6.61

6.1.27 Voir la solution de l'exercice du paragraphe 6.2.11.

6.2.8 A condition que toute variation de  $x_1$  et  $x_2$  soit interdite pour  $CK=0$ .

6.2.9  $D_1, D_2$  et  $D_3$  : voir la relation (5.13) du paragraphe 5.2.14.

$$Z = D_1$$

## 6.2.10

$$J_1 = \bar{x}_1 \bar{x}_2 ; K_1 = x_1 x_2 ; J_2 = \bar{x}_1 \bar{y}_1 + x_1 y_1 ; K_2 = \bar{x}_2 \bar{y}_1 + x_2 y_1$$

## 6.2.11

- Système de mémorisation (§ 6.1.16) :  $D = x_1 x_2 + \bar{x}_2 y_1 ; Z = y_1$ .
- Porte automatique (§ 6.1.18) :  $D = Z = x_1 \bar{x}_2 + x_1 y$ .
- Passage à niveau (§ 6.1.19) :  $D_1 = \bar{x}_1 x_2 + x_1 y_1 + y_1 \bar{y}_2 ; D_2 = x_1 \bar{x}_2 + x_2 y_2 + \bar{y}_1 y_2 ; Z = y_1 + y_2$ .
- Serrure électronique (§ 6.1.21) :  $D_1 = x_1 \bar{y}_3 + x_2 y_1 + \bar{x}_2 \bar{y}_1 y_2 + \bar{x}_2 \bar{y}_1 y_3 + y_1 \bar{y}_2 \bar{y}_3 + y_1 y_2 y_3 ; D_2 = \bar{x}_1 x_2 y_2 + \bar{x}_1 \bar{y}_1 y_2 + \bar{x}_1 y_2 y_3 + x_1 x_2 \bar{y}_1 y_3 ; D_3 = \bar{x}_1 x_2 \bar{y}_2 + \bar{x}_1 \bar{x}_2 \bar{y}_1 y_2 + \bar{x}_1 \bar{x}_2 y_2 y_3 + x_1 x_2 \bar{y}_1 y_3 ; Z_1 = \bar{x}_1 y_2 ; Z_2 = y_1 \bar{y}_2$ .
- Délai asynchrone (§ 6.1.22) :  $D_1 = x_1 \bar{x}_2 + x_1 y_1 + \bar{x}_2 y_1 ; D_2 = x_1 x_2 + x_1 y_2 + x_2 y_2 ; Z_1 = \bar{x}_1 y_1 + \bar{x}_1 y_2 + y_1 y_2 ; Z_2 = \bar{x}_2 y_2 + \bar{y}_1 y_2 + \bar{x}_2 \bar{y}_1$ .
- Préleveur de période (§ 6.1.23) :  $D_1 = \bar{x}_1 y_1 + x_2 y_1 \bar{y}_3 + x_2 y_2 \bar{y}_3 ; D_2 = \bar{x}_2 y_2 + x_2 \bar{y}_1 y_3 + \bar{x}_1 x_2 \bar{y}_1 \bar{y}_2 ; D_3 = \bar{x}_1 \bar{x}_2 y_1 + \bar{x}_1 \bar{y}_1 \bar{y}_2 + \bar{x}_1 y_1 y_3 + x_2 \bar{y}_1 y_3 + \bar{y}_1 \bar{y}_2 y_3 ; Z = y_1 \bar{y}_3 + y_2 \bar{y}_3$  avec  $x, Q_2 = x_1, x_2$  et  $Q_4 = Z$ .
- Passage à niveau (§ 6.1.27) :  $D_1 = x_1 y_1 + x_2 y_1 + \bar{x}_1 \bar{x}_2 y_2 ; D_2 = x_1 \bar{y}_1 + x_2 \bar{y}_1 + \bar{x}_1 \bar{x}_2 y_2 ; Z = y_1 + y_2$ .

6.2.21 Graphe des états  $\emptyset$  : voir la figure 6.62.

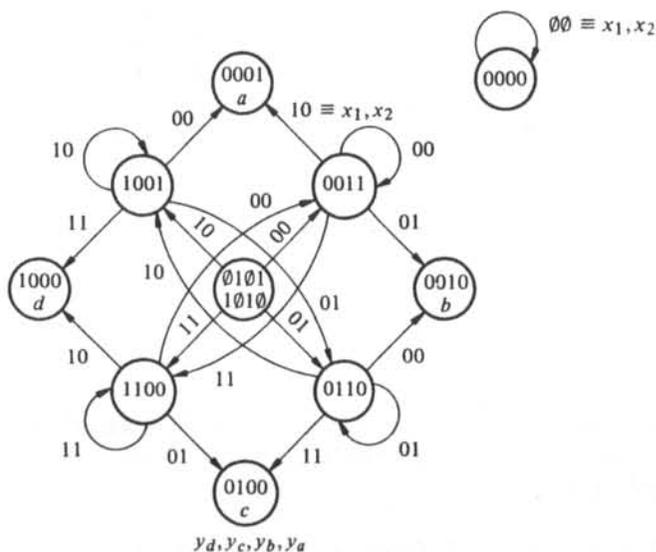


Fig. 6.62

Le système combinatoire itératif de la figure 6.25 se décompose en plusieurs sous-systèmes identiques à celui de la figure 6.63 et décrits par la table de la figure 6.64 : une variable  $D_j$  vaut 1 seulement lorsque  $y_j^+ = 1$  et lorsque toutes les variables précédentes  $y_{j-1}^+, y_{j-2}^+, \dots$  sont nulles (c'est-à-dire lorsque  $V_{j-1} = 0$ ); si  $D_j = 1$ , alors les variables suivantes  $D_{j+1}, D_{j+2}, \dots$  sont toutes nulles. Dans le cas où toutes les variables  $y^+$  sont nulles, on obtient  $D_d = Y_d = 1$  grâce à la disposition particulière du

dernier étage. L'analyse du logigramme de la figure 6.25 produit la table de vérité de la figure 6.65 qui démontre la propriété recherchée. La méthode du paragraphe 6.2.19 peut être modifiée ainsi : l'un des  $M$  états du graphe est décrit par une relation telle que (6.26); la bascule correspondante n'exige aucun câblage d'entrée ( $Y = 1$ ); en pratique on choisit la bascule qui comporte le nombre maximal d'entrées, par exemple  $y_0$  dans la figure 6.42.

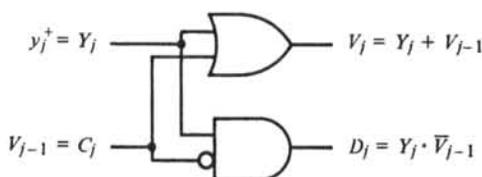


Fig. 6.63

	$Y_j$	
	0	1
$V_{j-1}$	0	1
	0,0	1,1
	1,0	1,0
	$V_j, D_j$	

Fig. 6.64

$y_d^+ y_c^+ y_b^+ y_a^+$	$D_d$	$D_c$	$D_b$	$D_a$
00 00	1	0	0	0
$a \equiv 00$ 01	0	0	0	1
00 11	0	0	0	1
$b \equiv 00$ 10	0	0	1	0
$c \equiv 01$ 00	0	1	0	0
01 01	0	0	0	1
01 11	0	0	0	1
01 10	0	0	1	0
11 00	0	1	0	0
11 01	0	0	0	1
11 11	0	0	0	1
11 10	0	0	1	0
$d \equiv 10$ 00	1	0	0	0
10 01	0	0	0	1
10 11	0	0	0	1
10 10	0	0	1	0

Fig. 6.65

**6.2.22** Tables d'états réduites : voir la solution de l'exercice du paragraphe 6.2.11. Adjonction du système itératif de la figure 6.25 : voir la solution de l'exercice du paragraphe 6.2.21.

**6.3.7** Voir la solution de l'exercice du paragraphe 6.4.8.

**6.3.8** Voir la solution de l'exercice du paragraphe 6.4.8.

**6.3.9** Voir la solution de l'exercice du paragraphe 6.4.8.

**6.3.10** Voir la solution de l'exercice du paragraphe 6.4.8.

**6.3.11** Voir la solution de l'exercice du paragraphe 6.4.8.

6.3.12 Voir la solution de l'exercice du paragraphe 6.4.8.

6.4.5

$$Q_1 \equiv y_1 = Y_1 ; Q_2 \equiv y_2 = \bar{Y}_2 \text{ (famille No 3 de la figure 6.34).}$$

6.4.6

$$J_1 = \bar{x}_1 x_2 y_2 ; K_1 = 1 ; J_2 = \bar{x}_1 \bar{x}_2 ; K_2 = x_1 + x_2 y_1$$

6.4.8

- Détecteur de séquence (§ 6.3.7) :  $D_1 = \bar{x}_1 x_2 y_2 \bar{y}_3 ; D_2 = \bar{x}_1 \bar{x}_2 y_1 + \bar{x}_1 \bar{x}_2 \bar{y}_2 y_3 + x_1 x_2 y_2 y_3 ; D_3 = \bar{x}_1 y_1 + \bar{x}_1 x_2 \bar{y}_2 + \bar{x}_1 x_2 y_3 + \bar{x}_1 \bar{y}_2 y_3 ; Z = \bar{x}_1 \bar{x}_2 y_1$ .
- Comparateur de bits (§ 6.3.8) :  $D_1 = \bar{x}_1 \bar{x}_2 y_1 + \bar{x}_1 \bar{x}_2 y_2 + x_1 x_2 y_1 + x_1 x_2 y_2 ; D_2 = \bar{x}_1 \bar{x}_2 \bar{y}_1 + x_1 x_2 \bar{y}_1 ; Z = \bar{x}_1 \bar{x}_2 y_1 \bar{y}_2 + x_1 x_2 y_1 \bar{y}_2$ . Voir aussi la solution de l'exercice du paragraphe 6.5.9.
- Compteur synchrone programmable (§ 6.3.9) :  $D_1 = x_1 \bar{y}_1 y_2 + x_1 \bar{x}_2 y_1 \bar{y}_2 ; D_2 = x_1 \bar{x}_2 \bar{y}_2 + x_2 \bar{y}_1 \bar{y}_2$ .
- Détecteur de séquences (§ 6.3.10) :  $D_1 = x ; D_2 = \bar{x} y_2 + \bar{x} y_3 ; D_3 = \bar{x} \bar{y}_2 ; Z = x y_1 + \bar{x} y_2 \bar{y}_3$ . La suppression éventuelle de l'état initial donne une table à 4 états : voir la solution de l'exercice du paragraphe 6.5.9 (fig. 6.68).
- Comparateur de mots (§ 6.3.11) :  $D_1 = y_1 \bar{y}_2 + \bar{x}_1 x_2 \bar{y}_2 + x_1 \bar{x}_2 \bar{y}_2 + \bar{x}_1 x_2 \bar{y}_1 y_3 + x_1 \bar{x}_2 \bar{y}_1 y_3 ; D_2 = \bar{y}_1 y_2 y_3 + y_1 \bar{y}_2 y_3 + \bar{x}_1 \bar{x}_2 \bar{y}_1 y_3 + x_1 x_2 \bar{y}_1 y_3 ; D_3 = y_1 \bar{y}_2 + \bar{x}_1 \bar{x}_2 \bar{y}_2 + x_1 x_2 \bar{y}_2 + \bar{x}_1 x_2 \bar{y}_1 y_3 + x_1 \bar{x}_2 \bar{y}_1 y_3 ; Z = \bar{x}_1 \bar{x}_2 y_2 \bar{y}_3 + x_1 x_2 y_2 \bar{y}_3$ . Si l'on pose  $x = \bar{x}_1 \oplus \bar{x}_2 = x_1 x_2 + \bar{x}_1 \bar{x}_2$  et  $\bar{x} = x_1 \oplus x_2$  alors on obtient :  $D_1 = y_1 \bar{y}_2 + \bar{x} \bar{y}_2 + \bar{x} \bar{y}_1 y_3 ; D_2 = \bar{y}_1 y_2 y_3 + y_1 \bar{y}_2 y_3 + x \bar{y}_1 y_3 ; D_3 = y_1 \bar{y}_2 + x \bar{y}_2 + \bar{x} \bar{y}_1 y_3 ; Z = x y_2 \bar{y}_3$ .
- Délai programmable (§ 6.3.12) :  $D_1 = y_2 ; D_2 = x_1 ; Z = \bar{x}_2 y_2 + x_2 y_1$ .

6.4.11

$$D_0 = x_1 + x_2 \bar{y}_1 ; D_1 = \bar{x}_1 \bar{x}_2 ; D_2 = \bar{x}_1 x_2 y_1 ; Z = x_1 \bar{x}_2 y_2$$

6.4.12 Figure 6.42 : en ajoutant le système itératif de la figure 6.25 on choisit  $Y_0 = 1$  (6.26), car la bascule  $y_0$  comporte le nombre maximal d'entrées (huit).

Figure 6.41 : voir la solution de l'exercice du paragraphe 6.4.11 (on vérifie en effet que  $y_0 + y_1 + y_2 = 1$  et que  $y_0 + y_2 = \bar{y}_1$ ).

6.4.13 Tables d'états réduites : voir la solution de l'exercice du paragraphe 6.4.8. Adjonction du système itératif de la figure 6.25 : voir la solution de l'exercice du paragraphe 6.2.21.

6.4.14 Pour le code  $y_d, y_c, y_b, y_a = 0001 \rightarrow 0010 \rightarrow 0100 \rightarrow 1000 \rightarrow 0001$  le logigramme corrigé est celui de la figure 6.25 avec :  $Y_d = 1, y_c^+ = y_b, y_b^+ = y_a$  et  $y_a^+ = y_d$ .

6.4.17

$$T = S\bar{Q} + RQ ; Q^+ = T\bar{Q} + \bar{T}Q = S\bar{Q} + \bar{R}Q = J\bar{Q} + \bar{K}Q \text{ si } S \equiv J \text{ et } R \equiv K.$$

6.4.18 Equations caractéristiques de la bascule *SR* (No 1) réalisée avec une bascule (No 2) du type *JK* et  $T: Q^+ = S\bar{Q} + \bar{R}Q$  (voir aussi la solution de l'exercice du paragraphe 6.4.17); du type *D*:  $Q^+ = S + \bar{R}Q$ .

Equations de condition (3.48) de diverses bascules (No 1) réalisées avec une bascule *SR* (No 2):  $SR = J\bar{Q} \cdot KQ \equiv 0$  (type *JK*),  $SR = T\bar{Q} \cdot TQ \equiv 0$  (type *T*) et  $SR = D \cdot \bar{D} \equiv 0$  (type *D*).

6.5.7 En posant  $x_1(2) = x_1, x_2(2) = x_2, x_1(1) = y_1, x_2(1) = y_2, x_1(0) = y_3, x_2(0) = y_4, Z(2) = Z = x_1\bar{x}_2\bar{y}_1y_2\bar{y}_3\bar{y}_4$  on obtient les tables d'états non réduite (fig. 6.66) et réduite (fig. 6.67).

$Y^+, Z$		$x_1, x_2$			
		00	01	11	10
0000	1	1,0	5,0	9,0	13,0
0001	2	1,0	5,0	9,0	13,0
0011	3	1,0	5,0	9,0	13,0
0010	4	1,0	5,0	9,0	13,0
0100	5	2,0	6,0	10,0	14,1
0101	6	2,0	6,0	10,0	14,0
0111	7	2,0	6,0	10,0	14,0
0110	8	2,0	6,0	10,0	14,0
1100	9	3,0	7,0	11,0	15,0
1101	10	3,0	7,0	11,0	15,0
1111	11	3,0	7,0	11,0	15,0
1110	12	3,0	7,0	11,0	15,0
1000	13	4,0	8,0	12,0	16,0
1001	14	4,0	8,0	12,0	16,0
1011	15	4,0	8,0	12,0	16,0
1010	16	4,0	8,0	12,0	16,0

$$y_1y_2y_3y_4 = Y$$

Fig. 6.66

$Y^+, Z$	$x_1, x_2$			
	00	01	11	10
$a \equiv 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16$	$b, 0$	$a, 0$	$a, 0$	$a, 0$
$b \equiv 1, 2, 3, 4$	$b, 0$	$c, 0$	$a, 0$	$a, 0$
$c \equiv 5$	$b, 0$	$a, 0$	$a, 0$	$a, 1$

$Y$

Fig. 6.67

6.5.8 Voir la solution de l'exercice du paragraphe 6.4.11 avec  $y_0^+ = D_0, y_1^+ = D_1$  et  $y_2^+ = D_2$ .

### 6.5.9

- Détecteur de séquence (§ 6.3.7); séquence déterminante:  $x_1, x_2 = 01 \rightarrow 00 \rightarrow 11 \rightarrow 01 \rightarrow 00$  ( $K=5$ ). Structure parallèle:  $Z(4) = x_1(0) \cdot x_2(0) \cdot x_1(1) \cdot x_2(1) \cdot x_1(2) \cdot x_2(2) \cdot x_1(3) \cdot x_2(3) \cdot x_1(4) \cdot x_2(4)$ . Structure

itérative : voir la solution de l'exercice du paragraphe 6.4.8 avec  $y_1^+ = D_1$ ,  $y_2^+ = D_2$  et  $y_3^+ = D_3$ .

- Détecteur de séquences (§ 6.3.10); séquences déterminantes :  $x = 1 \rightarrow 1$  et  $x = 0 \rightarrow 0 \rightarrow 0 \rightarrow 0$  ( $K = 4$ ). Structure parallèle :  $Z(3) = \overline{x(0)} \cdot \overline{x(1)} \cdot \overline{x(2)} \cdot \overline{x(3)} + x(2) \cdot x(3)$ . Structure itérative : la solution de l'exercice du paragraphe 6.4.8 donne une table à 5 états dont le premier (état initial) peut être supprimé : on obtient alors la table à 4 états de la figure 6.68.
- Comparateur de bits (§ 6.3.8); si l'on pose  $x = \overline{x_1} \oplus \overline{x_2} = x_1 x_2 + \overline{x_1} \overline{x_2}$  alors on a la séquence déterminante :  $x = 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$  ( $K = 4$ ). Structure parallèle :  $Z(3) = x(0) \cdot x(1) \cdot x(2) \cdot x(3)$ . Structure itérative : la solution de l'exercice du paragraphe 6.4.8 devient alors :  $y_1^+ = D_1 = x y_1 + x y_2$ ;  $y_2^+ = D_2 = x \overline{y_1}$ ;  $Z = x y_1 \overline{y_2}$ .
- Comparateur de mots (§ 6.3.11); il n'existe pas de séquence déterminante.

$Y^+, Z$	$x$	
	0	1
2	3,0	5,0
3	4,0	5,0
4	4,1	5,0
5	2,0	5,1

$Y$

Fig. 6.68

**6.5.15** Si l'on pose dans la figure 6.56  $x_1(1) = y_1$ ,  $x_2(1) = y_2$ ,  $x_2(0) = y_3$ ,  $Z(1) = Z$ ,  $x_1(2) = x_1$  et  $x_2(2) = x_2$  alors on a :  $y_1^+ = x_1 s + y_1 \overline{s}$ ;  $y_2^+ = x_2 s + y_2 \overline{s}$ ;  $y_3^+ = y_2 s + y_3 \overline{s}$ ;  $Z = y_1 \oplus y_3$ ;  $s = (x_1 \oplus y_1) + (x_2 \oplus y_2)$ . La table d'états calculée (fig. 6.69) est semblable à celle de la figure 6.5 à l'exception des états  $\emptyset$  et de certaines valeurs de  $Z$  dans des états totaux instables; cette dernière différence découle du cahier des charges initial (ainsi que du chronogramme de la figure 6.2) dans lequel on a implicitement admis que la variation de l'état d'entrée  $x_1, x_2$  provoquait instantanément une variation (éventuelle) de la sortie  $Z$ . Dans la réalisation de la figure 6.56 la fonction  $Z$  ne dépend que des variables internes ( $y_1$  et  $y_3$ ) et non des entrées ( $x_1$  et  $x_2$ ).

$Y^+, Z$		$x_1, x_2$			
		00	01	11	10
000	1	1,0	2,0	6,0	5,0
010	2	8,0	2,0	3,0	4,0
111	3	8,0	7,0	3,0	4,0
101	4	1,0	2,0	6,0	4,0
100	5	1,1	2,1	6,1	5,1
110	6	8,1	7,1	6,1	4,1
011	7	8,1	7,1	3,1	4,1
001	8	8,1	2,1	6,1	5,1

$y_1, y_2, y_3 \equiv Y$

Fig. 6.69

Dans la figure 6.56 on vérifie par le calcul algébrique que  $y_1^+ = x_1$  et  $y_2^+ = x_2$ ; en toute généralité on peut donc vérifier que  $C_1^+ = x_1$  et  $C_2^+ = x_2$  (fig. 6.57).

**6.5.16** Oui, avec  $y_1(0) = \emptyset$  et  $y_2(0) = \emptyset$ . Le logigramme final comprend un système séquentiel de synchronisation semblable à celui de la figure 6.57 et comportant quatre bascules dont les sorties sont  $B_1 = x_1(0)$ ,  $C_1 = x_1(1)$ ,  $B_2 = x_2(0)$  et  $C_2 = x_2(1)$ , ainsi que deux sous-systèmes combinatoires définis par les relations (6.10), (6.11) et (6.12).

### 6.5.17

- Délai asynchrone (§ 6.1.22); séquences déterminantes pour  $Z_1 : x_1, x_2 = 11 \rightarrow 10, 11 \rightarrow 01, 10 \rightarrow 00$  et  $10 \rightarrow 11$ ; séquences déterminantes pour  $Z_2 : x_1, x_2 = 11 \rightarrow 10, 11 \rightarrow 01, 01 \rightarrow 11$  et  $01 \rightarrow 00$ .
- Détecteur de séquences (§ 6.1.26); séquences déterminantes pour  $Z : x_1, x_2, x_3 = 000 \rightarrow 100 \rightarrow 000 \rightarrow 100 \rightarrow 000 \rightarrow 100$  et  $000 \rightarrow 100 \rightarrow 000 \rightarrow 010 \rightarrow 000 \rightarrow 100$ .

## CHAPITRE 7

**7.1.8** Figure 7.5 : un élément de délai pur avec  $\Delta_p = \Delta_{p1} + \Delta_{p2} + \dots + \Delta_{pn}$ . Figure 7.6 : mise en série d'un élément de délai inertiel  $\Delta_i$  et d'un élément de délai pur  $\Delta_p$ , avec  $\Delta_i$  égal au plus grand des délais  $\Delta_{i1}, \Delta_{i2}, \dots, \Delta_{in}$  et  $\Delta_p = \Delta_{i1} + \Delta_{i2} + \dots + \Delta_{in} - \Delta_i$ ; lorsque  $n \rightarrow \infty$  avec  $\Delta T = \text{constante}$ , on tend vers un délai pur avec  $\Delta_p = \Delta T$ .

**7.1.9** Logigramme minimal : mise en série d'un élément de délai inertiel  $\Delta_i$  et d'un élément de délai pur  $\Delta_p$  avec  $\Delta_i = 0,2 \cdot R$  et  $\Delta_p = 0,8 \cdot R$ . Logigramme avec délais inertiels seulement : mise en série de cinq éléments de délai inertiel  $\Delta_i$  avec  $\Delta_i = 0,2 \cdot R$ .

**7.2.18** Table d'états décrite par les relations (7.8): voir la figure 7.64.

**7.2.19** Branche No 1 :  $\Delta_2 + \Delta_4 > \Delta_1 + \Delta_3 > \Delta_2 > \Delta_1$ ; branche No 2 :  $\Delta_1 + \Delta_3 > \Delta_2 + \Delta_4 > \Delta_2 > \Delta_1$ ; branche No 3 :  $\Delta_2 > \Delta_1 + \Delta_3$ ; branche No 5 :  $\Delta_1 + \Delta_3 > \Delta_2 + \Delta_4 > \Delta_1 > \Delta_2$ . Absence d'aléa statique de  $Z : \Delta_2 + \Delta_4 > \Delta_1 + \Delta_3$ .

Cas particulier de l'égalité des délais : voir la figure 7.65 (traits forts) qui complète la figure 7.14.

### 7.2.20

$$y_1, y_2, y_3, y_4 = \underline{0000} \rightarrow \underline{1000} \rightarrow \underline{1010} \rightarrow 1011$$

**7.2.21** Suppression de l'aléa statique pour la séquence  $a, b, c = 111 \rightarrow 110$  : voir la figure 7.66 ( $y_4 = Z$ ).

**7.2.22** La figure 7.67 illustre l'exemple de la séquence d'entrée  $a, b = 00 \rightarrow 11$ . On remarque plusieurs aléas statiques de la sortie  $Z = y_4$ .

$y_1^* y_2^* y_3^* y_4^*$		$c$	
		0	1
00	00	1000	0100
00	01	1000	0100
00	11	1001	0101
00	10	1001	0101
01	00	1001	0101
01	01	1001	0101
01	11	1001	0101
01	10	1001	0101
11	00	1011	0111
11	01	1011	0111
11	11	1011	0111
11	10	1011	0111
10	00	1010	0110
10	01	1010	0110
10	11	1011	0111
10	10	1011	0111

$y_1 y_2 y_3 y_4$

Fig. 7.64

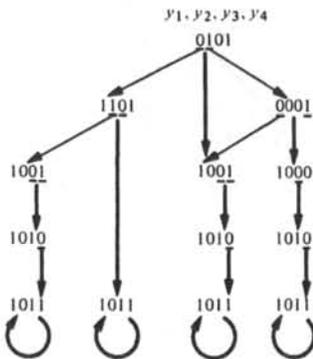


Fig. 7.65

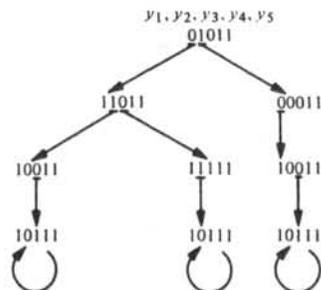


Fig. 7.66

7.2.23 Un exemple d'aléa dynamique peut être tiré du graphe :  $y_1, y_2, y_3, y_4, y_5 = \underline{0}11\underline{0}1 \rightarrow 1\underline{1}1\underline{1}01 \rightarrow 1\underline{1}1\underline{1}11 \rightarrow 1\underline{1}1\underline{1}10 \rightarrow 1\underline{1}01\underline{1}0 \rightarrow 1\underline{1}011 \rightarrow 1\underline{0}0\underline{1}11 \rightarrow 1011\underline{1}1 \rightarrow 101110$  avec  $Z = y_5 = 1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ .

7.3.15 Au minimum trois assemblages successifs de deux portes NAND avec, par exemple,  $\Delta_1 > \Delta_2, \Delta_4 > \Delta_3$  puis  $\Delta_2 > \Delta_4$  donc :  $\Delta_1 > \Delta_2 > \Delta_4 > \Delta_3$  (fig. 7.68); au maximum six assemblages successifs avec, par exemple,  $\Delta_4 > \Delta_1, \Delta_4 > \Delta_3, \Delta_2 > \Delta_3, \Delta_4 > \Delta_2, \Delta_1 > \Delta_3$  puis  $\Delta_2 > \Delta_1$  donc :  $\Delta_4 > \Delta_2 > \Delta_1 > \Delta_3$  (fig. 7.69).

7.3.16 Si  $\Delta_2 + \Delta_4 > \Delta_1 + \Delta_3$  alors  $y_1, y_2 = 01$ ; si  $\Delta_1 + \Delta_3 > \Delta_2 + \Delta_4$  alors  $y_1, y_2 = 10$ .

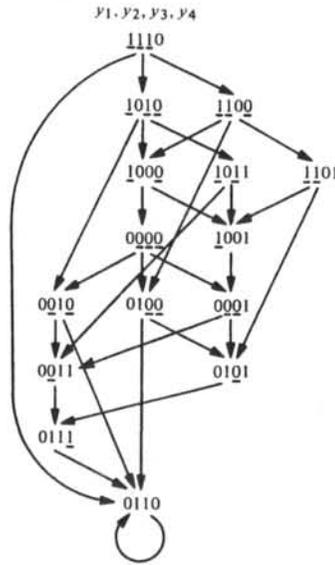


Fig. 7.67



Fig. 7.68

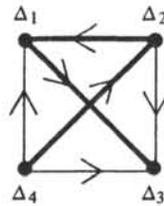


Fig. 7.69

7.3.17 Figure 7.32 :  $T = 2(\Delta_1 + \Delta_2)$ ; figure 7.33 :  $T = 2(\Delta_1 + \Delta_2 + \Delta_3)$ .

7.3.18 Pour  $x_1, x_2 = 10 \rightarrow 11$ , par exemple, il existe un aléa statique de  $Z$  avec  $\Delta T = \Delta_1 + \Delta_2 + \Delta_3$ .

7.3.19 Les deux modèles asynchrones sont équivalents; contrairement au régime permanent ( $Z = \bar{Q}$ ), la sortie  $Z$  n'est pas toujours égale à  $\bar{Q}$  en régime transitoire. Les équations et la table d'états du modèle de la figure 7.25 sont données dans la solution de l'exercice du paragraphe 7.4.19.

7.3.20 L'analyse du modèle à six délais démontre l'absence de toute course : le fonctionnement du diviseur est indépendant des valeurs des délais  $\Delta_1$  à  $\Delta_6$ ;  $Q$  et  $\bar{Q}$  ne sont pas toujours complémentaires en régime transitoire. Un modèle asynchrone minimal peut être décrit par les relations  $y_1^+ = y_1 y_3 + \bar{y}_2 + \bar{C}K, y_2^+ = \bar{y}_1 + y_2 \bar{y}_3 + \bar{C}K, y_3^+ = \bar{y}_1 + y_2 y_3$  où  $\Delta_1, \Delta_2$  et  $\Delta_3$  sont les éléments de délai placés à la sortie des portes No 2, 3 et 5 (fig. 7.35).

7.4.12 Courses critiques à partir des états  $y_1, y_2, y_3 = 001$  (pour  $S, R, CK = 100$ ) et  $y_1, y_2, y_3 = 111$  (pour  $S, R, CK = 010$ ). Condition d'équivalence :  $\Delta_1 + \Delta_2 > \Delta_3$ .

7.4.13 Pour  $S \cdot R \cdot \overline{CK} = 0$ , il n'existe plus aucune course : la table d'états calculée est équivalente à celle de la figure 7.37 ( $PR, CLR = 00$ ).

7.4.14

$S, R, CK = 000 \rightarrow 100 \rightarrow 101 \rightarrow 100 \rightarrow 000 \rightarrow 010 \rightarrow 000$

$S, R, CK = 100 \rightarrow 000 \rightarrow 010 \rightarrow 011$

7.4.15 En inversant le signal  $CK$  du logigramme de la figure 7.50, on constate que la table de celui-ci est équivalente à celle de la figure 7.37 ( $PR, CLR = 00$ ) pour les états totaux du modèle quasi-synchrone (fig. 7.38).

7.4.16 Table d'états : voir la figure 7.70; cette table est équivalente à celle de la figure 7.43 pour les états totaux du modèle quasi-synchrone qui sont représentés en trait fort. L'effet d'une impulsion de  $J$  est illustré par la figure 7.71 (selon table de la figure 7.43) et par la figure 7.72 (selon table de la figure 7.70).

$y_1^+ y_2^+$	$PR, CLR = 00$ $CK = 0$				$CK = 1$				$PR, CLR = 10$	$PR, CLR = 01$	$PR, CLR = 11$
	$JK$ 00	01	11	10	$JK$ 00	01	11	10			
0 0	00	00	10	10	00	00	00	00	11	00	—
0 1	11	01	01	11	00	00	00	00	11	00	—
1 1	11	01	01	11	11	11	11	11	11	00	—
1 0	00	00	10	10	11	11	11	11	11	00	—

$y_1 y_2 = Q$

Fig. 7.70

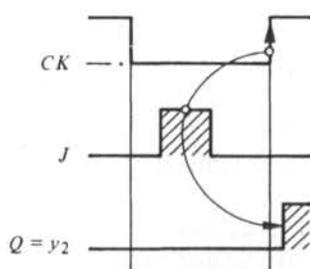


Fig. 7.71

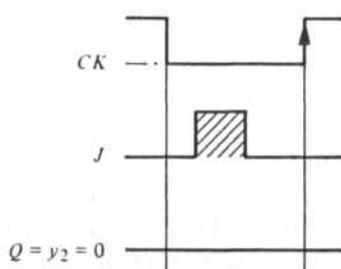


Fig. 7.72

7.4.17 Table d'états identique à celle de la figure 7.70 pour  $PR, CLR = 00$ ; l'effet d'une impulsion de  $J$  est illustré par la figure 7.72. Comparaison avec la figure 7.43 : voir la solution de l'exercice du paragraphe 7.4.16.

7.4.18 Résumé du comportement : à la montée de  $CK$  les valeurs de  $J$  et  $K$  sont enregistrées et définissent la valeur  $Q^+$  de la bascule qui apparaît sur la sortie  $y_2$  à la descente

suivante de  $CK$ . En inversant le signal  $CK$  de la figure 7.53, cette table est équivalente à celle de la figure 7.43 ( $PR, CLR = 00$ ) pour les états totaux du modèle quasi-synchrone.

7.4.19 Equations du modèle asynchrone de la figure 7.25 :

$$y_1^+ = \overline{CK} + CLR + (\overline{D}y_1 + y_1 \overline{y_2}) \overline{PR}$$

$$y_2^+ = \overline{CK} + \overline{y_1} + Dy_2 \cdot \overline{CLR}$$

$$y_3^+ = \overline{y_1} + PR + y_2 y_3 \cdot \overline{CLR}$$

A l'exception de l'état  $PR, CLR = 11$  (pour lequel la table de la figure 7.44 n'est pas définie) les tables des figures 7.73 et 7.44 sont totalement équivalentes. Les séquences des figures 7.42 et 7.47 sont inchangées.

$y_1^+ y_2^+ y_3^+$	$PR, CLR = 00$				$PR, CLR = 10$				$PR, CLR = 01$				$PR, CLR = 11$
	$CK = 0$		$CK = 1$		$CK = 0$		$CK = 1$		$CK = 0$		$CK = 1$		
	$D, CK$		$D, CK$		$D, CK$		$D, CK$		$D, CK$		$D, CK$		
	00	10	11	01	00	10	11	01	00	10	11	01	
00   0	111	111	011	011	111	111	011	011	111	111	111	111	-
00   1	111	111	011	011	111	111	011	011	111	111	111	111	-
01   1	111	111	(011)	(011)	111	111	(011)	(011)	111	111	111	111	-
01   0	111	111	011	011	111	111	011	011	111	111	111	111	-
10   0	110	110	(100)	(100)	111	111	001	001	110	110	(100)	(100)	-
10   1	110	110	100	100	111	111	001	001	110	110	100	100	-
11   1	(111)	(111)	011	101	(111)	(111)	011	001	110	110	100	100	-
11   0	(110)	(110)	010	100	111	111	011	001	(110)	(110)	100	100	-

$$y_1, y_2, y_3 = Q$$

Fig. 7.73

7.5.5 Retard obtenu par une inversion de  $x$  dans les états totaux  $x, CK; y_1, y_2, v_1, v_2 = (00, 0101), (00, 1001), (10, 0110)$  et  $(10, 1010)$ . Bascules  $D$  commandées par un flanc : les équations (7.65), (7.66) et (7.67) sont inchangées, tandis que la relation (7.64) devient  $y_1^+ = (x \oplus y_2 \oplus v_2) \overline{CK} + CK \cdot y_1$ .

7.5.6 Compteur réversible (§ 5.5.3) : voir la figure 5.55 (solution de l'exercice du paragraphe 5.5.7).

Préleveur de période (§ 5.5.5); si  $Q_2 = x_1$  et  $x = x_2$ , si  $v_1$  et  $v_2 = Q_3$  sont les variables internes de  $Q_3$ , si  $y_1$  et  $y_2 = Q_4$  sont les variables internes de  $Q_4$ , alors on a :

$$v_1^+ = x_2 \overline{v_2} \overline{y_2} + v_1 \overline{y_2} ; v_2^+ = \overline{x_2} v_1 \overline{y_2} + x_2 v_2 \overline{y_2}$$

$$y_1^+ = \overline{x_1} y_1 + v_2 y_1 + x_1 v_2 \overline{y_2} + y_1 \overline{y_2} ; y_2^+ = \overline{x_1} y_1 + x_1 y_2$$

On peut vérifier dans la table d'états de  $v_1^+, v_2^+, y_1^+, y_2^+$  le chronogramme de la figure 5.51.

7.5.7 Si  $y_1$  et  $y_2 = Q_8, v_1$  et  $v_2 = Q_4, u_1$  et  $u_2 = Q_2$  sont les variables internes des bascules  $Q_8, Q_4, Q_2$  et si  $\overline{CK}_2 = H$  on obtient les équations suivantes :

$$\begin{aligned}
 y_1^+ &= H v_2 u_2 + y_1 \bar{y}_2 + \bar{H} y_1 ; y_2^+ = \bar{H} y_1 + H y_2 ; H y_2 v_2 u_2 = 0 \\
 v_1^+ &= \bar{v}_2 u_2 + v_1 \bar{u}_2 ; v_2^+ = v_1 \bar{u}_2 + v_2 u_2 \\
 u_1^+ &= H \bar{y}_2 \bar{u}_2 + \bar{H} u_1 + u_1 \bar{u}_2 ; u_2^+ = \bar{H} u_1 + H u_2
 \end{aligned}$$

On retrouve dans la table d'états de  $y_1^+, y_2^+, v_1^+, v_2^+, u_1^+, u_2^+$  le graphe de la figure 4.37.

7.5.8 Toute course critique est supprimée si  $\Delta_1 < \Delta_2 + 2 \cdot \Delta_3$  où  $\Delta_1$  et  $\Delta_2$  sont les délais des variables internes  $y_1, y_2$  de la bascule.

7.5.9 Figure 7.62 : si  $y_1, y_2$  et  $v_1, v_2$  sont les variables internes des bascules, alors les équations sont :

$$\begin{aligned}
 y_1^+ &= \bar{y}_2 v_2 + y_1 \bar{v}_2 ; y_2^+ = y_1 \bar{v}_2 + y_2 v_2 \\
 v_1^+ &= \bar{C}\bar{K} \cdot \bar{v}_2 Z + C K \cdot v_1 Z ; v_2^+ = C K \cdot v_1 Z + \bar{C}\bar{K} \cdot v_2 Z \\
 Z^+ &= \bar{y}_2 + \bar{v}_2
 \end{aligned}$$

La table d'états présente une course critique (pour  $\bar{C}\bar{K} = 1 \rightarrow 0$ ) décrite par le graphe simplifié de la figure 7.74; le calcul de l'inéquation entre délais nécessiterait la représentation détaillée de chaque branche de ce graphe. On peut cependant vérifier dans la figure 7.74 que si  $\Delta_5$  est supérieur à chacun des quatre autres délais (ceux de  $y_1, y_2, v_1$  et  $v_2$ ), alors la course critique est supprimée (état final désiré :  $Z, y_1, y_2, v_1, v_2 = 10000$ ).

Figure 7.63; les équations précédentes sont inchangées, à l'exception de  $Z^+$  :

$$Z^+ = \bar{C}\bar{K} \cdot \bar{y}_2 + \bar{C}\bar{K} \cdot \bar{v}_2 + \bar{y}_2 Z + \bar{v}_2 Z \text{ avec } \bar{C}\bar{K} \cdot y_2 v_2 = 0$$

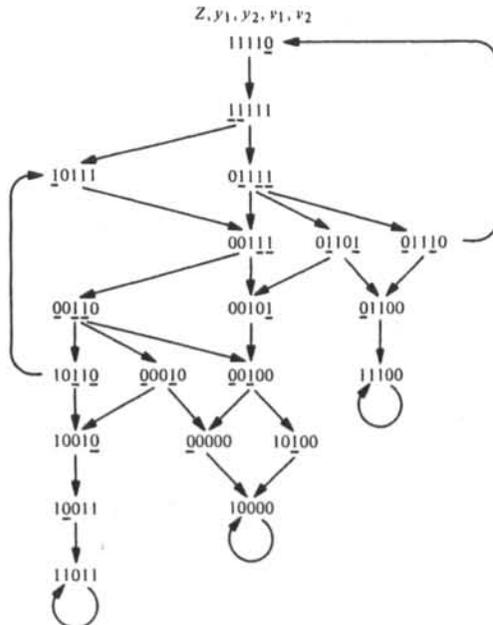


Fig. 7.74

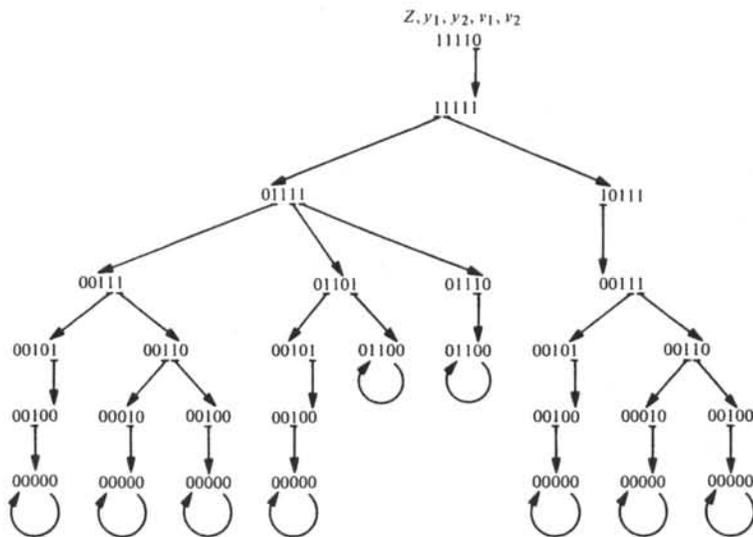


Fig. 7.75

La table d'états présente une course critique (pour  $\overline{CK} = 1 \rightarrow 0$ ) décrite par le graphe de la figure 7.75; l'inéquation supprimant cette course (état final désiré :  $Z, y_1, y_2, v_1, v_2 = 00000$ ) se résume à :

$$\Delta y_1 < \Delta v_2 + \Delta Z$$

où  $\Delta y_1$ ,  $\Delta v_2$  et  $\Delta Z$  sont les délais de  $y_1$ ,  $v_2$  et  $Z$ .

## BIBLIOGRAPHIE

- [1] COMMISSION ELECTROTECHNIQUE INTERNATIONALE, *Publication 117-15, Symboles graphiques recommandés, 15ème partie: Opérateurs logiques binaires*, Bureau Central de la Commission Electrotechnique Internationale, Genève, 1972.
- [2] *Military Standard, Graphic Symbols for Logic Diagrams, MIL-STD-806B*, Department of Defense, Washington, 1962.
- [3] ANSI, IEEE, *American National Standard, graphic symbols for logic diagrams (two-state devices), IEEE Std 91-1973, ANSI Y 32.14-1973*, Institute of Electrical and Electronics Engineers, New York, 1973.
- [4] H.C. WEHRFRITZ, Techniques for the Transformation of Logic Equations, *IEEE Transactions on Computers*, vol. C-23, No. 5, May 1974, pp. 477-480.
- [5] G. BOOLE, W.S. JEVONS, *Algèbre et logique*, Albert Blanchard, Paris, 1962.
- [6] A. WARUSFEL, *Dictionnaire raisonné de mathématiques*, Seuil, Paris, 1966.
- [7] D. FÈVRE, C. DE BARY, R. BOIREL, P. BUISSON, N. GAUTIER, M. GLAYMANN, M. PASCAL, F. RUSSO, A. WARUSFEL, Y. PESEZ, *Les dictionnaires du savoir moderne: les mathématiques*, Centre d'Etude et de Promotion de la Lecture, Paris, 1973.
- [8] T.L. BOOTH, *Digital Networks and Computer Systems*, Wiley, New York, 1971.
- [9] M. BOSOM, G. CHATY, *Mathématique et automatique*, Hachette, Paris, 1969.
- [10] F.J. HILL, G.R. PETERSON, *Introduction to Switching Theory and Logical Design, sec. ed.*, Wiley, New York, 1974.
- [11] H.T. NAGLE Jr., B.D. CARROLL, J.D. IRWIN, *An Introduction to Computer Logic*, Prentice-Hall, Englewood Cliffs, 1975.
- [12] M.M. MANO, *Computer logic design*, Prentice-Hall, Englewood Cliffs, 1972.
- [13] H.A. CURTIS, *A New Approach to the Design of Switching Circuits*, D. van Nostrand, Princeton, 1962.
- [14] P. DEBRAINE, *Machines de traitement de l'information, tome 1: étude logique et construction des circuits*, Masson, Sofradel, Paris, 1967.
- [15] G.A. MALEY, J. EARLE, *The Logic Design of Transistor Digital Computers*, Prentice-Hall, Englewood Cliffs, 1963.
- [16] D. ZISSOS, *Logic Design Algorithms*, Oxford University Press, London, 1972.
- [17] E.J. McCLUSKEY Jr., Minimization of Boolean Functions, *The Bell System Technical Journal*, vol. 35, November 1956, pp. 1417-1444.

- [18] D. LEWIN, L. MARET, *Systèmes logiques*, Spes, Lausanne, 1972.
- [19] A.H. SCHEINMAN, A Method for Simplifying Boolean Functions, *The Bell System Technical Journal*, vol. 41, July 1962, pp. 1337-1346.
- [20] O. FOELLINGER, W. WEBER, *Methoden der Schaltalgebra*, R. Oldenbourg, München, 1967.
- [21] P. TISON, Generalization of Consensus Theory and Application to the Minimization of Boolean Functions, *IEEE Transactions on Electronic Computers*, vol. EC-16, No. 4, August 1967, pp. 446-456.
- [22] M. DAVIO, *Cours polycopié: Systèmes logiques*, Faculté des Sciences appliquées de l'Université Catholique de Louvain (UCL), Louvain-la-Neuve, 1976.
- [23] J. BELZER, A.G. HOLZMAN, A. KENT, ed., *Encyclopedia of Computer Science and Technology*, vol. 1: A to AMPL, Marcel Dekker, New York, 1975.
- [24] D.L. DIETMEYER, *Logic Design of Digital Systems*, Allyn and Bacon, Boston, 1971.
- [25] M. KARNAUGH, The Map Method for Synthesis of Combinational Logic Circuits, *Transactions AIEE*, vol. 72, pt.I, November 1953, pp. 593-598.
- [26] F.C. HENNIE, *Iterative Arrays of Logical Circuits*, Wiley, New York, 1961.
- [27] T.C. BARTEE, Computer Design of Multiple-Output Logical Networks, *IRE Transactions on Electronic Computers*, vol. EC-10, No. 1, March 1961, pp. 21-30.
- [28] J.B. PEATMAN, *The Design of Digital Systems*, McGraw-Hill, New York, 1972.
- [29] A. BARNA, D.I. PORAT, *Integrated Circuits in Digital Electronics*, Wiley, New York, 1973.
- [30] M.H. MIESSLER, Use of exclusive-OR gates for Boolean minimisation, *Proceedings of the IEE*, vol. 119, No. 9, September 1972, pp. 1269-1272.
- [31] F.C. HENNIE, *Finite-state models for logical machines*, Wiley, New York, 1968.
- [32] J. CHINAL, *Techniques booléennes et calculateurs arithmétiques*, Dunod, Paris, 1967.
- [33] E.J. McCLUSKEY, *Introduction to the Theory of Switching Circuits*, McGraw-Hill, New York, 1965.
- [34] M. PHISTER, *Logical design of digital computers*, Wiley, New York, 1963.
- [35] D. MANGE, Equations logiques du flip-flop, *Neue Technik*, No. A3, 1967, pp. 165-173.
- [36] J. FLORINE, *La synthèse des machines logiques et son automatisation*, Dunod, Paris et Presses Académiques européennes, Bruxelles, 1964.
- [37] S. RUDEANU, *Boolean functions and equations*, North-Holland, Amsterdam, 1974.
- [38] G. BOULAYE, *Logique et organes des calculatrices numériques*, Dunod, Paris, 1970.
- [39] N.C. DE TROYE, A generalized two-input flip-flop and its realization, *Philips Research Reports*, vol. 21, October 1966, pp. 390-409.
- [40] R.M.M. OBERMAN, *Electronic counters*, Macmillan, London, 1973.
- [41] W. FLEISCHHAMMER, Eine Systematik der zusammengesetzten bistabilen Kippstufen, *Elektron. Rechenanl.*, vol. 10, No. 1, 1968, pp. 34-40.

- [42] J.J. SPARKES, Bistable Elements for Sequential Circuits, *Electronic Engineering*, vol. 38, No. 462, August 1966, pp. 510-515.
- [43] C.M. REEVES, *An Introduction to Logical Design of Digital Circuits*, Cambridge University Press, Cambridge, 1972.
- [44] G.K. KOSTOPOULOS, *Digital Engineering*, Wiley, New York, 1975.
- [45] W.E. WICKES, *Logic Design with Integrated Circuits*, Wiley, New York, 1968.
- [46] T. KOHONEN, *Digital Circuits and Devices*, Prentice-Hall, Englewood Cliffs, 1972.
- [47] R.P. HARRIS, Simplify sequencers with an improved counter, *Electronic Design*, vol. 21, No. 18, September 1973, pp. 124-127.
- [48] J.I. ACHA, J.L. HUERTAS, On Modulus 8 Gateless Synchronous Scalers with J-K Flip-Flops, *Proceedings of the IEEE*, vol. 64, No. 3, March 1976, pp. 374-375.
- [49] F.B. MANNING, R.R. FENICHEL, Synchronous Counters Constructed Entirely of J-K Flip-Flops, *IEEE Transactions on Computers*, vol. C-25, No. 3, March 1976, pp. 300-306.
- [50] M. DAVIO, G. BIOUL, *Report R279: Interconnection structure of cyclic counters made up of JK flip-flop*, M.B.L.E., Brussels, 1974.
- [51] B.W. SMITH, Remapping technique trades gates for flip-flop, *EDN*, vol. 16, No. 10, May 1971, pp. CH6-CH9.
- [52] N.N. BISWAS, The Logic and Input Equations of Flip-Flops, *Electronic Engineering*, vol. 38, No. 456, February 1966, pp. 107-111.
- [53] B. NOLET, Etude et réalisation des compteurs à l'aide de bascules JK et D, *Electronique et microélectronique industrielles*, No. 170, Avril 1973, pp. 75-81.
- [54] TEXAS INSTRUMENTS, *Designing with TTL Integrated Circuits*, McGraw-Hill, New York, 1971.
- [55] P. SIEBER, Der Entwurf von Folgeschaltungen, *Der Elektroniker*, vol. 6, No. 2, März 1967, pp. 72-77.
- [56] G.H. LANDERS, *Designing Integrated Serial Counters*, Application Note AN-194, Motorola, Phoenix, 1965.
- [57] M.J. CHARLAND, Look to asynchronous sequential logic for best speed-power performance, *Electronic Design*, vol. 22, No. 20, September 1974, pp. 98-103.
- [58] P. NASLIN, *Circuits logiques et automatismes à séquences*, 3ème édition, Dunod, Paris, 1970.
- [59] W.R. ASHBY, *Introduction à la cybernétique*, Dunod, Paris, 1958.
- [60] G. CORAY, *Langages réguliers et automates séquentiels finis*, Ecole polytechnique fédérale, Lausanne, 1972.
- [61] A.C. DAVIES, The Design of Feedback Shift Registers and other Synchronous Counters, *The Radio and Electronic Engineer*, vol. 37, No. 4, April 1969, pp. 213-223.
- [62] E.J. McCLUSKEY Jr., S.H. UNGER, A Note on the Number of Internal Variable Assignments for Sequential Switching Circuits, *IRE Transactions on Electronic Computers*, vol. EC-8, No. 4, December 1959, pp. 439-440.

- [63] C.G. BELL, J. GRASON, A. NEWELL, *Designing computers and digital systems*, Digital Press, 1972.
- [64] F.J. HILL, G.R. PETERSON, *Digital Systems: Hardware Organization and Design*, Wiley, New York, 1973.
- [65] D.A. HUFFMAN, The Synthesis of Sequential Switching Circuits, *Journal of The Franklin Institute*, vol. 257, No. 3, March 1954, pp. 161-190 and No. 4, April 1954, pp. 275-303.
- [66] G.H. MEALY, A Method for Synthesizing Sequential Circuits, *The Bell System Technical Journal*, vol. 34, September 1955, pp. 1045-1079.
- [67] J. FLORINE, *Automatismes à séquences et commandes numériques*, Dunod, Paris, 1969.
- [68] J. ZAHND, D. MANGE, Réduction des tables d'états complètement définies, *Systèmes logiques – Cahiers de la C.S.L.*, No.5, octobre 1973, pp. 287-304.
- [69] J. ZAHND, D. MANGE, Réduction des tables d'états incomplètement définies, *Systèmes logiques – Cahiers de la C.S.L.*, No.6, mars 1975, pp. 314-343.
- [70] J. KUNTZMANN, P. NASLIN, *Algèbre de Boole et machines logiques*, Dunod, Paris, 1967.
- [71] J.C. LAURENT, R. DAVID, Synthèse cellulaire de systèmes séquentiels complexes définis par leur graphe primitif. Intégration à grande échelle, *Revue Française d'Automatique, Informatique et de Recherche opérationnelle*, No. J-1, 1972, pp. 19-34.
- [72] V. PICCAND, *Introduction aux circuits de commutation, tome 2*, SEN électronique, Genève.
- [73] A.D. FRIEDMAN, Feedback in Synchronous Sequential Switching Circuits, *IEEE Transactions on Electronic Computers*, vol. EC-15, No. 3, June 1966, pp. 354-367.
- [74] J.P. PERRIN, M. DENOUE, E. DACLIN, *Systèmes logiques, tome 1*, Dunod, Paris, 1967.
- [75] J.P. PERRIN, M. DENOUE, E. DACLIN, *Systèmes logiques, tome 2*, Dunod, Paris, 1967.
- [76] W.M. PENNEY, L. LAU, ed., *MOS Integrated Circuits: Theory, Fabrication, Design, and Systems Applications of MOS LSI*, van Nostrand Reinhold, New York, 1972.
- [77] R. DAVID, Synthèse de réseaux séquentiels cellulaires, *Automatisme*, tome XV, No. 3, mars 1970, pp. 89-97.
- [78] R. DAVID, *Thèse: Réalisation de systèmes séquentiels asynchrones par interconnexion simple de cellules séquentielles identiques*, Université de Grenoble, 1969.
- [79] A.D. FRIEDMAN, P.R. MENON, *Theory & Design of Switching Circuits*, Computer Science Press, Woodland Hills, 1975.
- [80] J. BEISTER, A Unified Approach to Combinational Hazards, *IEEE Transactions on Computers*, vol. C-23, No. 6, June 1974, pp. 566-575.
- [81] A. THAYSE, Transient analysis of logical networks applied to hazard detection, *Philips Research Reports*, vol. 25, No. 5, October 1970, pp. 261-336.

- [82] R.E. MILLER, *Switching Theory, volume 2: Sequential Circuits and Machines*, Wiley, New York, 1965.
- [83] TEXAS INSTRUMENTS, *The TTL Data Book, Supplement to CC-401, for Design Engineers*, Texas Instruments, 1973.
- [84] J.A. BRZOZOWSKI, M. YOELI, *Digital Networks*, Prentice-Hall, Englewood Cliffs, 1976.
- [85] G.A. MALEY, *Manual of Logic Circuits*, Prentice-Hall, Englewood Cliffs, 1970.
- [86] S.H. UNGER, Hazards and Delays in Asynchronous Sequential Switching Circuits, *IRE Transactions on Circuit Theory*, vol. CT-6, No. 1, March 1959, pp. 12-25.
- [87] D. MANGE, Modèles asynchrones des bascules bistables, *Systèmes logiques – Cahiers de la C.S.L.*, No. 5, Octobre 1973, pp. 256-286.
- [88] J.P. DESCHAMPS, Synchronization of Finite Automata, *Philips Research Reports*, vol. 27, No. 2, April 1972, pp. 126-139.
- [89] J.J. SPARKES, *Transistor switching and sequential circuits*, Pergamon, London, 1969.
- [90] R. TREADWAY, Follow a logical sequence to synthesize circuits, *Electronic Design*, vol. 16, No. 6, March 1968, pp. 174-179.
- [91] T.L. DOLLHOFF, Program cuts logic-design costs, *Electronic Design*, vol. 22, No. 9, April 1974, pp. 186-189.
- [92] V. BUBENIK, Weighting Method for the Determination of the Irredundant Set of Prime Implicants, *IEEE Transactions on Computers*, vol. C-21, No. 12, December 1972, pp. 1449-1451.
- [93] P.S. NOE, V.T. RHYNE, S. SURARATRUNGSI, Comments on "Weighting Method for the Determination of the Irredundant Set of Prime Implicants", *IEEE Transactions on Computers*, vol. C-23, No. 6, June 1974, p. 646.
- [94] A. GRASSELLI, F. LUCCIO, A Method for Minimizing the Number of Internal States in Incompletely Specified Sequential Networks, *IEEE Transactions on Electronic Computers*, vol. EC-14, No. 3, June 1965, pp. 350-359.
- [95] W.S. MEISEL, A Note on Internal State Minimization in Incompletely Specified Sequential Networks, *IEEE Transactions on Electronic Computers*, vol. EC-16, No. 4, August 1967, pp. 508-509.
- [96] F.M. BROWN, Equational Realizations of Switching Functions, *IEEE Transactions on Computers*, vol. C-24, No. 11, November 1975, pp. 1054-1066.
- [97] R.L. VALLÉE, *Analyse binaire, tome 1: théorie et applications aux circuits combinatoires*, Masson, Paris, 1970.
- [98] B. GIRLING, H.G. MORING, *Logic and Logic Design*, International Textbook Company, Aylesbury, Bucks, 1973.
- [99] G.E. HOERNES, M.F. HEILWEIL, *Introduction à l'algèbre de Boole et aux dispositifs logiques*, Dunod, Paris, 1966.
- [100] D.D. GIVONE, *Introduction to Switching Circuit Theory*, McGraw-Hill, New York, 1970.

- [101] E.B. EICHELBERGER, Hazard Detection in Combinational and Sequential Switching Circuits, *IBM Journal of research and development*, vol. 9, No. 2, March 1965, pp. 90-99.
- [102] S.H. UNGER, *Asynchronous Sequential Switching Circuits*, Wiley, New York, 1969.
- [103] R.L. VALLÉE, *Analyse binaire, tome 2: clef des automates numériques*, Masson, Paris, 1970.
- [104] A. THAYSE, La détection des aléas dans les circuits logiques au moyen du calcul différentiel booléen, *Digital Processes*, vol. 1, No. 2, 1975, pp. 141-169.

# INDEX ANALYTIQUE

Les références sont celles des pages

- Adjacents (monômes ...), 286
- Aléa
  - dynamique, 255, 257
  - essentiel, 273
  - statique, 254
- Aléas (théorie générale des ...), 256
- Algèbre
  - de Boole (algèbre logique), 11
  - logique, 11
  - logique (postulats de l' ...), 38, 39
- Algébrique (démonstration ...), 11
- Algorithme, 221
- Analyse
  - algébrique des systèmes combinatoires, 71
  - des compteurs quasi-synchrones, 144
  - des compteurs synchrones, 138
  - des diviseurs de fréquence, 118
  - des systèmes combinatoires, 70
  - des systèmes séquentiels quasi-synchrones, 188
  - des systèmes séquentiels synchrones, 166
  - d'un compteur synchrone réversible, 164
  - d'une structure série d'un système combinatoire itératif, 185
  - du registre à décalage, 175
  - graphique des systèmes combinatoires, 71
- Anneau (compteur en ...), 142
- Assignée
  - (table d'états ...), 167
  - (table d'états non ...), 167
- Assignement (codage), 205
- Associativité
  - (théorème d' ... du OU-exclusif), 27
  - (théorèmes d' ...), 11
- Asynchrone
  - (comportement ...), 197
  - (délai ...), 203
  - (entrée ...), 93
  - (modèle ... d'un système logique), 246
  - (modèle ... minimal), 259
  - (modèle logique ...), 3, 241
  - (système logique ...), 4, 243
- Asynchrones (systèmes ... équivalents), 261
- Barre (complément, négation, inverse, forme complémentaire), 6
- Bartee, 76, 295
- Bascule bistable
  - commandée par un flanc (edge-triggered ou edge-sensitive flip-flop), 271
  - *D* (Delay flip-flop), 102, 260, 264, 270, 272, 275, 276
  - (état interne de la ...), 93
  - *JK* (*JK* flip-flop), 104, 270, 274, 275, 276
  - *LP*, 114
  - (modèle quasi-synchrone de la ...), 94
  - (modèle synchrone de la ...), 99
  - *RST*, 114
  - *SR* (*SR* flip-flop), 93, 265, 273, 274
  - *sr*, 103
  - *T* (Trigger flip-flop), 106
  - verrouillée (data lockout flip-flop), 275
  - *vw*, 108
  - *X*, 114
  - *xy*, 114
  - *XYVW*, 108
- Bascules bistables (transformation des ...), 229, 230, 231
- Base, 283
- BCD (code ...), 286
- Binaire
  - (code ... codé décimal), 286
  - (code ... pur), 284
  - (nombre ...), 283

- (système de numération ...), 283
- Binaires (chiffres ...), 283
- Binary Coded Decimal (code BCD, code binaire codé décimal), 286
- Bistable (basculer ...), voir bascule bistable
- Bits (chiffres binaires), 283
- Blanchard (symbolisme bidimensionnel de ...), 12
- Bloc, 51
- Boole (algèbre de ...), 11
- Borne
  - inférieure, 34
  - supérieure, 34
- Canonique
  - (forme ... algébrique), 22
  - (forme ... décimale), 30
- Cellulaire (système séquentiel ...), 216
- Chemin, 49
- Chiffres, 283
  - binaires (bits), 283
- Chronogramme, 2
  - d'un modèle asynchrone, 247
- Clear, 93
- Clock pulse (signal d'horloge), 93
- Codage (assignement), 205
  - minimal, 205
  - 1 parmi  $M$ , 211
- Code
  - BCD (Binary Coded Decimal, code binaire codé décimal), 286
  - binaire codé décimal, 286
  - binaire pur, 284
  - d'un compteur (séquence), 121
  - Excédent 3 (Excess 3), 284
  - Gray, 284
  - négabinaire, 138
  - non pondéré, 284
- Codes
  - BCD-Excédent 3 (convertisseur de ...), 70
  - Excédent 3-BCD (convertisseur de ...), 70
  - Gray-binaire pur (convertisseur de ...), 81
- Combinatoire
  - (élément ...), 4
  - (modèle logique ...), 3
  - (système ... itératif), 84
  - (système logique ...), 4
- Commission électrotechnique internationale, CEI (normes de la ...), 7
- Commutativité (théorèmes de ...), 10
- Complément (négation, inverse, barre, forme complémentaire), 6
- Complémentation (théorèmes de ...), 10
- Complet
  - (compteur ...), 148
  - (opérateur ...), 16
- Complète (famille d'opérateurs ...), 38, 39
- Complètement définie (table d'états ...), 198
- Comportement
  - asynchrone, 197
  - synchrone, 220
- Compteur
  - complet, 148
  - en anneau, 142
  - Johnson, 141
  - partiel, 148
  - quasi-synchrone (compteur), 117, 144
  - réversible, 190
  - synchrone, 118, 129
  - synchrone (état du ..., état interne du ...), 129
  - synchrone programmable, 174, 182, 185
  - synchrone réversible (compteur-décompteur), 161, 185, 277
- Compteur-décompteur (compteur synchrone réversible), 161, 185, 277
- Compteurs (décomposition des ...), 148
- Concentré (système séquentiel ...), 216
- Conception des systèmes séquentiels synchrones, 195
- Conditions
  - indifférentes (états  $\emptyset$ , conditions  $\emptyset$ , don't care conditions, don't happen conditions), 33
  - $\emptyset$ , 33
- Consensus (théorèmes du ...), 11
- Constante logique, 6
- Constantes (théorèmes des ...), 10
- Convertisseur de codes
  - BCD-Excédent 3, 70
  - Excédent 3-BCD, 70
  - Gray-binaire pur, 81
- Couches (nombre de ... d'un logigramme), 49
- Course, 250
  - critique, 259

- Coût (paramètre spatial), 85  
 Critique (course ...), 259  
 Curtis, 41  
 Cycle parasite, 140
- Damier, 79  
 Data  
 – lockout flip-flop (bascule verrouillée), 275  
 – selector (multiplexeur, système combinatoire universel), 23
- David, 217  
 Décalage (registre à ...), 174  
 Décimal  
 – (nombre ...), 283  
 – (numéro ...), 30  
 – (système de numération ...), 283  
 Décimale (forme canonique ...), 30  
 Déclenchement, 109  
 Décomposition, 41  
 – des compteurs, 148  
 – en parallèle des compteurs, 151  
 – en série des compteurs, 148  
 – mixte des compteurs, 154  
 Délai (paramètre temporel), 85  
 Délai (retard), 3  
 Délai  
 – asynchrone, 203  
 – (élément de ...), 4  
 – inertiel (élément de ...), 242  
 – programmable, 223  
 – pur (élément de ...), 241  
 Delay flip-flop (bascule  $D$ ), 102  
 Démonstration  
 – algébrique, 11  
 – tabulaire, 11  
 De Morgan  
 – (théorèmes de ...), 12  
 – (transformation de ...), 13  
 Démultiplexeur, 22  
 Descente, 94  
 Détecteur de séquence, 218  
 Déterminante (séquence d'entrée ... de longueur  $K$ ), 231  
 Diagramme de Venn, 6  
 Dietmeyer, 64  
 Différence (fonction ...), 34  
 DIN 40700 (normes ...), 7  
 Discret (signal ...), 3  
 Discriminateur du sens de rotation, 166,
- Distributivité (théorèmes de ...), 10  
 Diviseur  
 – de fréquence, 118  
 – de fréquence par deux, 106, 265  
 – de fréquence par trois, 281  
 – (état du ...), 119  
 – (état interne du ...), 119  
 Division (rapport de ... d'un compteur), 121  
 Don't care conditions (don't happen conditions, conditions indifférentes, états  $\emptyset$ , conditions  $\emptyset$ ), 33  
 Don't happen conditions, 33  
 Dynamique (aléa ...), 255, 257
- Edge-sensitive flip-flop (edge-triggered flip-flop, bascule commandée par un flanc), 271  
 Edge-triggered flip-flop, 271  
 Egales (expressions algébriques ...), 14  
 Élément  
 – combinatoire, 4  
 – de délai, 4  
 – de délai inertiel (délai inertiel), 242  
 – de délai pur (délai pur), 241  
 – de mémoire (latch, verrou, loquet), 87, 257, 260, 261  
 – de mémoire (modèle asynchrone de l'...), 90, 257, 260, 261  
 – de mémoire (modèle combinatoire de l'...), 88  
 Enclenchement, 109  
 Entrée  
 – asynchrone, 93  
 – d'excitation, 93  
 – (équation d'...), 112  
 – (état d'...), 4, 159, 243  
 – (séquence d'... déterminante de longueur  $K$ ), 231  
 – (variable d'...), 159, 243  
 Equation  
 – caractéristique de la bascule, 101  
 – d'entrée, 112  
 – logique, 28  
 Equivalentes  
 – (bascules fonctionnellement ...), 101  
 – (expressions algébriques ...), 14  
 – (tables d'états ...), 199, 261  
 Equivalents  
 – (graphes des états ...), 199  
 – (systèmes asynchrones ...), 261

- (systèmes fonctionnellement ...), 14
- (systèmes séquentiels synchrones fonctionnellement ...), 200
- Esclave (slave), 93
- Essentiel
  - (aléa ...), 273
  - (impliquant premier ...), 57
- ET (fonction ...), 8
- Etat (état interne, état secondaire), 159, 243
- Etat
  - de la bascule (état interne de la bascule), 93
  - d'entrée, 4
  - d'entrée (état primaire), 159, 243
  - de sortie, 4, 159, 243
  - d'excitation, 243
  - du compteur synchrone (état interne), 129
  - du diviseur (état interne), 119
  - futur de la bascule, 94
  - futur du compteur synchrone, 129
  - interne (état, état secondaire), 159, 243
  - interne de la bascule (état de la bascule), 93
  - interne du compteur synchrone (état), 129
  - interne du diviseur (état), 119
  - permanent, 119
  - présent de la bascule, 94
  - présent du compteur synchrone, 129
  - primaire (état d'entrée), 159, 243
  - secondaire (état, état interne), 159, 243
  - total, 159, 243
  - total instable, 170, 243
  - total stable, 169, 243
  - transitoire, 119
  - $\emptyset$ , 123
- Etats
  - (graphe des ...), 109
  - (table d'... de la bascule), 100
  - $\emptyset$  (conditions  $\emptyset$ , conditions indifférentes, don't care conditions, don't happen conditions), 33
- Excédent 3 (code ...), 284
- Excess 3 (code Excédent 3), 284
- Excitation, 243
  - (entrée d'...), 93
  - (état d'...), 243
- Exclusive OR (XOR, fonction OU-exclusif de deux variables), 25
- Expressions algébriques
  - égales (équivalentes), 14
  - équivalentes, 14
- Famille
  - de fonctions de l'équation caractéristique d'une bascule, 102
  - des solutions d'une fonction incomplètement définie, 33
  - des solutions d'une table d'états incomplètement définie, 199
  - d'opérateurs complète, 38, 39
- Finie (système séquentiel à mémoire ...), 231
- Flanc (bascule commandée par un ...), 271
- Flip-flop
  - (delay ...), 102
  - (*JK* ...), 104
  - (*SR* ...), 93
  - (trigger ...), 106
- Fonction
  - caractéristique de la bascule, 101
  - différence, 34
  - ET (produit logique, intersection), 8
  - incomplètement définie, 33
  - incomplètement définie (solution d'une ...), 33
  - indifférente, 34
  - logique, 4
  - logique universelle, 22
  - MAJORITÉ, 29
  - maximale, 34
  - minimale, 34
  - NAND (fonction NON-ET), 14
  - NON, 6
  - NON-ET (fonction NAND), 14
  - NON-OU (fonction NOR), 16
  - NOR (fonction NON-OU), 16
  - OU (somme logique, réunion, union, OU inclusif), 9
  - OU-exclusif de deux variables (exclusive OR, XOR), 25
  - OU-exclusif de  $n$  variables, 27
  - unité (minterme), 20
  - universelle (solution de la ...), 23
- Fonctionnellement
  - équivalentes (bascules ...), 101
  - équivalents (systèmes ...), 14
  - équivalents (systèmes séquentiels synchrones ...), 200

- Fonctionnement normal de la bascule, 96, 97, 99
- Fonctions
- de plusieurs variables, 7
  - d'une variable, 5
  - multiples, 74
- Fondamental (mode ...), 258
- Forme
- canonique algébrique, 22
  - canonique décimale, 30
  - complémentaire (complément, négation, inverse, barre), 6
  - paramétrique d'une fonction incomplètement définie, 36
  - vraie (identité), 6
- Fréquence
- (diviseur de ...), 118
  - (diviseur de ... par deux), 106, 265
  - (diviseur de ... par trois), 281
- Futur
- (état ... de la bascule), 94
  - (état ... du compteur synchrone), 129
  - (instant ...), 94
- Graphe des états, 109
- de l'élément de mémoire, 258
  - du détecteur de séquence, 221, 226, 227
  - du discriminateur du sens de rotation, 169, 197, 216, 235, 236
  - d'un compteur quasi-synchrone, 146
  - d'un compteur synchrone, 131, 139, 140
  - d'un compteur synchrone réversible, 162, 277, 279
  - d'un diviseur, 119, 126, 149, 152
  - d'un modèle asynchrone, 250
  - d'un système séquentiel synchrone, 168
  - du registre à décalage, 175, 178
  - simplifié d'un modèle asynchrone, 252
- Graphes des états équivalents, 199
- Gray (code ...), 284
- Horloge
- digitale, 128
  - (instant d'...), 94
  - (intervalle d'...), 94
  - (signal d'...), 93
- Huffman, 171, 201, 263
- Idempotence (théorèmes d'...), 10
- Identité (forme vraie), 6
- Impliquant premier (monôme premier), 53
- à choix (impliquant premier non essentiel), 57
  - essentiel, 57
  - non essentiel (impliquant premier à choix), 57
  - superflu, 57
- Impliquants premiers (table des ...), 55
- Impulsion, 204
- Impulsions
- (préleveur d'...), 193
  - (système séquentiel à ...), 203
- Inclusif (OU ...), 9
- Incomplètement définie
- (fonction ...), 33
  - (table d'états ...), 199
- Indifférente (fonction ...), 34
- Indifférentes (conditions ...), 33
- Inéquation logique, 36
- Inertiel
- (délai ...), 242
  - (élément de délai ...), 242
- Instable
- (état total ...), 170, 243
  - (régime ...), 89
  - (variable interne ...), 250
- Instant
- d'horloge (intervalle d'horloge), 94
  - futur, 94
  - présent, 94
- Interne
- (état ...), 159, 243
  - (variable ...), 159, 243
  - (variable ... instable), 250
- Intersection (produit logique, fonction ET), 8
- Intervalle d'horloge (instant d'horloge), 94
- Inverse (complément, négation, barre, forme complémentaire), 6
- Inverseur (porte NON, opérateur NON), 6
- Itératif (système combinatoire ...), 84
- Itératifs (réalisation séquentielle des systèmes combinatoires ...), 185
- Itérative (structure ...), 84
- Johnson (compteur ...), 141
- Karnaugh, 63
- (table de ...), 50, 60, 62

- Langage  
 – de transfert de registres, 181  
 – naturel, 169
- Latch (élément de mémoire, verrou, loquet), 87
- Lettre, 14
- Logigramme, 6
- Logigrammes (transformation de ...), 17
- Longueur  $K$  (séquence d'entrée déterminante de ...), 231
- Loquet (élément de mémoire, verrou, latch), 87
- Maintien  
 – à 0, 109  
 – à 1, 109
- Maître (master), 93
- MAJORITÉ (fonction ...), 29
- Master-slave, 93
- Maximale (fonction ...), 34
- Maxterme, 25
- McCluskey, 55, 64, 76, 286, 293
- Mealy, 171, 201
- Mémoire  
 – (élément de ...), 87, 257, 260, 261  
 – finie (système séquentiel à ...), 231
- Mesure d'un compteur, 121
- Metal Oxide Semiconductor (MOS), 255, 273
- MIL-STD-806B (normes ...), 7
- Minimal  
 – (codage ...), 205  
 – (modèle asynchrone ...), 259
- Minimale (fonction ...), 34
- Minterme (fonction unité), 20
- Mise à 1 de la bascule, 96, 99
- Mixte (décomposition ... des compteurs), 154
- Mode fondamental, 258
- Modèle  
 – combinatoire de l'élément de mémoire, 88  
 – logique asynchrone, 3, 241  
 – logique combinatoire, 3  
 – synchrone de la bascule, 99
- Modèle asynchrone  
 – (chronogramme d'un ...), 247  
 – de la bascule  $D$ , 271, 272  
 – de la bascule  $JK$ , 270  
 – de la bascule  $SR$ , 265, 267, 268  
 – de l'élément de mémoire, 90, 257, 260, 261
- d'un compteur synchrone réversible, 277
- d'un système logique, 246
- (graphe des états d'un ...), 250
- minimal, 259
- (table d'états d'un ...), 247
- Modèle quasi-synchrone de la bascule, 94  
 –  $D$ , 271, 272  
 –  $JK$ , 270  
 –  $SR$ , 267, 268
- Modes de représentation  
 – analytiques de la bascule, 101, 102  
 – synthétiques de la bascule, 109, 112
- Monôme, 14  
 – premier (impliquant premier), 53  
 – résultant, 287
- Monômes adjacents, 286
- Montée, 94
- MOS (Metal Oxide Semiconductor), 255, 273
- Multiples (fonctions ...), 74
- Multiplexeur (système combinatoire universel, data selector), 23
- NAND (fonction ...), 14
- Naturel (langage ...), 169
- Négabinaire (code ...), 138
- Négation (complément, inverse, barre, forme complémentaire), 6
- Niveaux (nombre de ... d'un logigramme), 49
- Nombre  
 – binaire, 283  
 – décimal, 283  
 – de couches d'un logigramme (nombre de niveaux, profondeur), 49  
 – de niveaux d'un logigramme, 49
- NON  
 – (fonction ...), 6  
 – (opérateur ...), 6  
 – (porte ...), 6
- Non essentiel (impliquant premier ...), 57
- NON-ET (fonction ...), 14
- NON-OU (fonction ...), 16
- NOR (fonction ...), 16
- Normes  
 – de la Commission électrotechnique internationale, CEI, 7  
 – DIN 40700, 7  
 – MIL-STD-806B, 7

- Numération  
 – (système de ...), 283  
 – (système de ... binaire), 283  
 – (système de ... décimal), 283  
 Numéro décimal, 30
- Opérateur (porte), 6  
 – complet, 16  
 – NON (porte NON, inverseur), 6  
 Opérateurs OU-exclusif (transformation en une expression avec des ...), 77  
 Organigramme, 221  
 Oscillateur logique, 263  
 OU (fonction ...), 9  
 OU-exclusif  
 – de deux variables (fonction ...), 25  
 – de  $n$  variables (fonction ...), 27  
 – (transformation en une expression avec des opérateurs ...), 77  
 OU inclusif, 9
- Parallèle  
 – (décomposition en ... des compteurs), 151  
 – (structure ...), 84  
 Paramètre  
 – spatial (coût), 85  
 – temporel (délai), 85  
 Paramètres  
 – du modèle asynchrone, 247  
 – d'une fonction logique universelle, 22  
 Paramétrique (forme ... d'une fonction incomplètement définie), 36  
 Parasite (cycle ...), 140  
 Partiels (compteurs ...), 148  
 Période (préleveur de ...), 190, 193  
 Permanent  
 – (état ...), 119  
 – (régime ...), 89  
 Petrick, 55  
 Phase, 196  
 Phister, 102  
 Poids, 283  
 Polynôme, 14  
 Pondéré (code non ...), 284  
 Porte (opérateur), 6  
 – NON (opérateur NON, inverseur), 6  
 Postulats de l'algèbre logique, 38, 39  
 Préleveur  
 – de période, 190, 193  
 – d'impulsions, 193
- Premier  
 – (impliquant ...), 53  
 – (impliquant ... essentiel), 57  
 – (impliquant ... non essentiel), 57  
 – (impliquant ... superflu), 57  
 – (monôme ...), 53  
 Premiers (table des impliquants ...), 55  
 Présent  
 – (état ... de la bascule), 94  
 – (état ... du compteur synchrone), 129  
 – (instant ...), 94  
 Preset, 93  
 Primaire  
 – (état ...), 159, 243  
 – (variable ...), 159, 243  
 Produit logique (intersection, fonction ET), 8  
 Profondeur d'un logigramme (nombre de niveaux, nombre de couches), 49  
 Programmable  
 – (compteur synchrone ...), 174, 182, 185  
 – (délai ...), 223  
 Pseudo-parallèle (structure ...), 84  
 Pur  
 – (délai ...), 241  
 – (élément de délai ...), 241
- Quantification, 3  
 Quantifié (signal ...), 3  
 Quasi-synchrone  
 – (compteur ...), 117, 144  
 – (système séquentiel ...), 159, 188
- Rapport de division d'un compteur, 121  
 Réalisation des systèmes séquentiels synchrones, 195  
 Réduction des tables d'états, 200  
 Régime  
 – établi (régime permanent, régime stable), 89  
 – instable (régime transitoire), 89  
 – permanent, 89  
 – stable, 89  
 – transitoire, 89  
 Registre à décalage, 174  
 Registres  
 – à décalage (réalisation de systèmes séquentiels synchrones avec des ...), 231, 232, 237, 238  
 – (langage de transfert de ...), 181  
 Remise à 0 de la bascule, 97, 99

- Réparti (système séquentiel ...), 216
- Représentation
  - spatiale de l'information, 185
  - temporelle de l'information, 185
- Reset, 92, 93
- Résultant (monôme ...), 287
- Retard (délai), 3
- Réunion (somme logique, union, fonction OU, OU inclusif), 9
- Réversible
  - (compteur ...), 190
  - (compteur synchrone ...), 161, 185, 277
- Rotation (discriminateur du sens de ...), 166, 196
- Rudeanu, 113
  
- Scheinman, 64
- Secondaire
  - (état ...), 159, 243
  - (variable ...), 159, 243
- Sens de rotation (discriminateur du ...), 166, 196
- Séquence
  - d'entrée déterminante de longueur  $K$ , 231
  - (détecteur de ...), 218
  - d'un compteur (code), 121
- Séquentiel (système logique ...), 89
- Série
  - (décomposition en ... des compteurs), 148
  - (structure ... d'un système combinatoire itératif), 185
- Set, 92, 93
- Signal
  - d'horloge (clock pulse), 93
  - discret, 3
  - quantifié, 3
- Simplification, 42, 50
  - algébrique, 43
  - graphique, 46
  - simultanée des fonctions multiples, 74
- Solution
  - de la fonction universelle, 23
  - de l'équation caractéristique d'une bascule, 102
  - d'une fonction incomplètement définie, 33
  - (table d'états complètement définie ... d'une table d'états incomplètement définie), 199
- Somme logique (réunion, union, fonction OU, OU inclusif), 9
- Sortie
  - (état de ...), 4, 159, 243
  - (variable de ...), 4, 159, 243
- Sorties de la bascule bistable, 93
- Spatial (paramètre ...), 85
- Spatiale (représentation ... de l'information), 185
- Stable
  - (état total ...), 169, 243
  - (régime ...), 89
- Statique (aléa ...), 254
- Structure
  - itérative (structure pseudo-parallèle), 84
  - parallèle, 84
  - pseudo-parallèle, 84
  - série d'un système combinatoire itératif, 185
- Superflu (impliquant premier ...), 57
- Symbolisme bidimensionnel de Blanchard, 12
- Synchrone
  - (comportement ...), 220
  - (compteur ...), 118, 129
  - (système séquentiel ...), 159, 166
- Synchronisation (système séquentiel de ...), 237
- Synthèse
  - de compteurs synchrones avec des registres à décalage, 178
  - des compteurs quasi-synchrones, 148
  - des compteurs synchrones, 129
  - des diviseurs de fréquence, 123
  - des systèmes combinatoires, 41
  - des systèmes séquentiels quasi-synchrones, 193
  - des systèmes séquentiels synchrones, 195
  - d'un compteur synchrone réversible, 162
- Système
  - concret, 1
  - de numération, 283
  - de numération binaire, 283
  - de numération décimal, 283
- Système combinatoire
  - itératif, 84
  - itératif (structure série d'un ...), 185
  - universel (multiplexeur, data selector), 23

- Système logique
- asynchrone, 4, 243
  - combinatoire, 4
  - séquentiel, 89
- Système séquentiel
- à impulsions, 203
  - à mémoire finie, 231
  - cellulaire (réparti), 216
  - concentré, 216
  - de synchronisation, 237
  - quasi-synchrone, 159, 188
  - réparti, 216
  - synchrone, 159, 166
  - synchrone (graphe des états d'un ...), 168
  - synchrone (table d'états d'un ...), 167
- Systèmes
- asynchrones équivalents, 261
  - combinatoires fonctionnellement équivalents, 14
  - combinatoires itératifs (réalisation séquentielle des ...), 185
- Systèmes séquentiels
- quasi-synchrones (analyse de ...), 188
  - quasi-synchrones (synthèse des ...), 193
  - synchrones (analyse des ...), 166
  - synchrones avec des registres à décalage (réalisation de ...), 231, 232, 237, 238
  - synchrones fonctionnellement équivalents, 200
  - synchrones (synthèse des ...), 195
- Table
- de Karnaugh, 50, 60, 62
  - de Karnaugh principale, 64
  - de Karnaugh secondaire, 64
  - des impliquants premiers, 55
  - des transitions, 109
  - des transitions résumée, 110
  - de vérité, 4
  - de vérité de la bascule (table d'états de la bascule), 100
- Table d'états
- assignée, 167
  - complètement définie, 198
  - de la bascule (table de vérité de la bascule), 100
  - de la bascule  $D$ , 271, 272
  - de la bascule  $JK$ , 270, 276
  - de la bascule  $SR$ , 267, 268, 269
  - de l'élément de mémoire, 258, 261, 262
  - du détecteur de séquence, 221, 222, 223, 224, 234
  - du discriminateur du sens de rotation, 167, 198, 199, 200, 206, 207, 211, 212
  - d'un compteur synchrone, 131, 139, 140
  - d'un compteur synchrone réversible, 162, 277, 278
  - d'une structure série d'un système combinatoire itératif, 187
  - d'un modèle asynchrone, 247
  - d'un système séquentiel synchrone, 167
  - du registre à décalage, 175
  - incomplètement définie, 199
  - non assignée, 167
- Tables d'états
- équivalentes, 199, 261
  - (réduction des ...), 200
- Tabulaire (démonstration ...), 11
- Temporel (paramètre ...), 85
- Temporelle (représentation ... de l'information), 185
- Texas Instruments, 260, 270, 271
- Tison, 64
- Toggle, 276
- Total
- (état ...), 159, 243
  - instable (état ...), 170, 243
  - stable (état ...), 169, 243
- Transfert (langage de ... de registres), 181
- Transformation
- de logigrammes, 17
  - des bascules bistables, 229, 230, 231
  - d'une expression algébrique, 42
  - en une expression avec des opérateurs OU-exclusif, 77
  - espace-temps, 85
- Transition d'une bascule, 109
- Transitions
- (table des ...), 109
  - (table des ... résumée), 110
- Transitoire
- (état ...), 119
  - (régime ...), 89
- Trigger flip-flop (bascule  $T$ ), 106
- Unger, 263
- Union (somme logique, réunion, fonction OU, OU inclusif), 9

- Universel (système combinatoire ...), 23
- Universelle (fonction logique ...), 22
  
- Variable
  - d'entrée, 4
  - d'entrée (variable primaire), 159, 243
  - de sortie, 4, 159, 243
  - (fonctions d'une ...), 5
  - interne (variable secondaire), 159, 243
  - interne instable, 250
  
- logique (postulat), 38
- primaire, 159, 243
- secondaire, 159, 243
- Variabes (fonctions de plusieurs ...), 7
- Venn (diagramme de ...), 6
- Verrou (élément de mémoire, loquet, latch), 87
- Verrouillée (bascule ...), 275
  
- XOR (fonction OU-exclusif de deux variables, Exclusive OR), 25

# FORMULAIRE

## MODÈLES LOGIQUES

### Modèle logique asynchrone

Celui-ci est le modèle général auquel se réfère l'ensemble du volume V; il est un assemblage d'éléments idéaux qui sont de deux types : l'élément combinatoire et l'élément de délai (§ 1.1.9). Les règles d'assemblage précisent que deux bornes de sortie ne peuvent être reliées à une même borne d'entrée et que dans toute boucle de rétroaction il existe au moins un élément de délai.

On admet généralement que les délais sont inertiels et que leurs valeurs ne sont ni nulles, ni infinies (§ 7.1.4 et 7.2.4).

Le temps est une variable continue.

### Modèle logique quasi-synchrone

Celui-ci est un modèle asynchrone particulier des bascules bistables obtenu en admettant que (§ 3.2.3) :

- tous les délais des éléments combinatoires sont nuls à l'exception de ceux des éléments de mémoire;
- toute variation des entrées d'excitation ( $S$  et  $R$  par exemple) et des entrées asynchrones ( $PR$  et  $CLR$ ) est interdite pendant les instants d'horloge, c'est-à-dire lorsque  $CK = 0$ .

Le temps est une variable continue.

### Modèle logique synchrone

Celui-ci est un cas particulier du modèle quasi-synchrone obtenu en admettant que (§ 3.2.10) :

- le fonctionnement de la bascule est normal ( $PR = CLR = 0$ );
- l'état  $Q$  est défini pendant les seuls instants d'horloge, c'est-à-dire lorsque  $CK = 0$ .

Le temps est alors une variable discrète.

### Modèle logique combinatoire

Celui-ci est un cas particulier du modèle asynchrone obtenu lorsque tous les délais sont nuls (§ 1.1.10).

Le paramètre temps n'intervient donc pas.

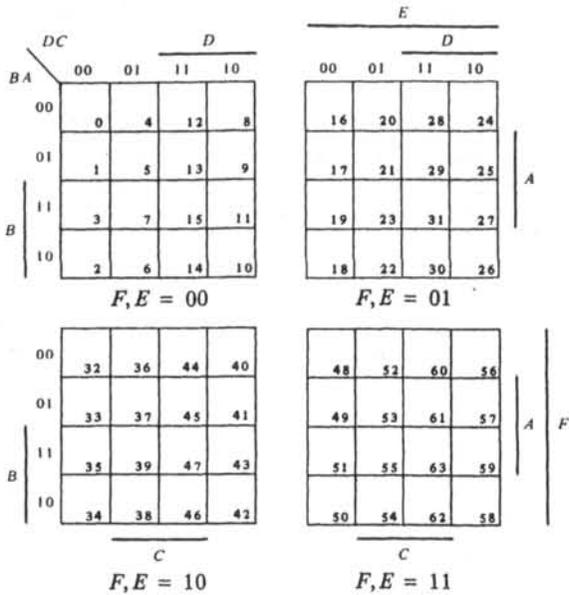
## SYSTÈMES LOGIQUES COMBINATOIRES

Fonctions logiques	Normes CEI	Normes MIL	Normes DIN
$\bar{a}$ (NON)			
$ab$ (ET)			
$a + b$ (OU)			
$\overline{ab} = \bar{a} + \bar{b}$ $a \uparrow b$ (NAND)			
$\overline{a + b} = \bar{a} \bar{b}$ $a \downarrow b$ (NOR)			
$a \oplus b = \bar{a}b + a\bar{b}$ (OU-exclusif)			

$a$	$\bar{a}$
0	1
1	0

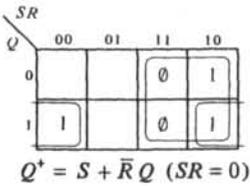
$ab$	$\bar{a}b$	$a\bar{b}$	$a + b$	$a \uparrow b$	$a \downarrow b$	$a \oplus b$
00	0	0	1	1	0	0
01	0	1	1	0	1	1
10	0	1	1	0	1	1
11	1	1	0	0	0	0

TABLE DE KARNAUGH À 6 VARIABLES  $F, E, D, C, B, A$

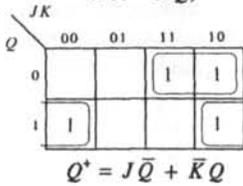


BASCULES BISTABLES

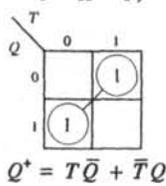
Type SR



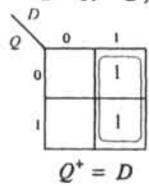
Type JK  
(Type SR avec  $S = J\bar{Q}$   
et  $R = KQ$ )



Type T  
(Type JK avec  
 $J = K = T$ )



Type D  
(Type SR avec  
 $S = R = D$ )

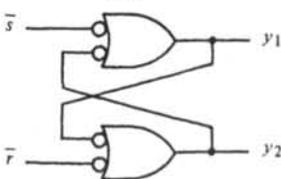


Tables des transitions

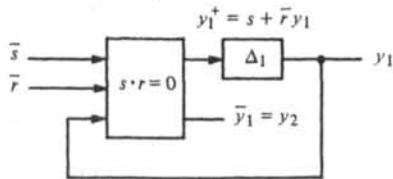
No	Q	Q <sup>+</sup>	S	R	J	K	T	D
0	0	0	0	∅	0	∅	0	0
1	0	1	1	0	1	∅	1	1
2	1	0	0	1	∅	1	1	0
3	1	1	∅	0	∅	0	0	1

ÉLÉMENT DE MÉMOIRE  $\bar{s}\bar{r}$

Logigramme

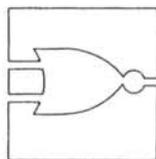


Modèle asynchrone minimal



Le Traité d'Electricité est l'œuvre collective  
des membres du Département d'Electricité de l'EPFL,  
assistés par quelques collaborateurs externes.  
A ce volume ont collaboré plus particulièrement :

- Christian Bernard* : matériel d'expérimentation  
*Eduard Cerny* : correction des épreuves  
*Jean-Daniel Chatelain* : direction de la production, dessins  
*Marc Davio* : critique du manuscrit  
*Florence de Bondeli* : correction de la première édition  
*André Decurnex* : résolution des exercices  
*Claire-Lise Delacrausaz* : composition du texte,  
coordination de la production  
*Yves Despond* : résolution des exercices  
*Walter Dübler* : photographie  
*Olivier Gardiol* : critique du manuscrit  
*Riccardo Grana* : composition de formules  
*Kurt Hofer* : dessins  
*Roland Jaques* : production  
*Roland Kallmann* : résolution des exercices  
*Allen Kilner* : mise en page et montage  
*Daniel Mange* : rédaction  
*Hubert Monbaron* : photographie  
*Jacques Neiryneck* : direction du Traité,  
critique du manuscrit  
*Jean-Daniel Nicoud* : critique du manuscrit  
*Jean-Michel Priou* : critique du manuscrit  
*Marco Rossi* : résolution des exercices  
*Eduardo Sanchez* : contrôle des exercices,  
correction des épreuves  
*Othmar Schaer* : photographie  
*André Spatz* : correction des épreuves  
*André Stauffer* : résolution des exercices  
*Eric Vittoz* : critique du manuscrit  
*Jacques Zahnd* : critique du manuscrit



d'éléments combinatoires permet de réaliser des compteurs (qui comportent une seule variable indépendante : le signal d'horloge) et des systèmes séquentiels synchrones ou quasi-synchrones (avec plusieurs variables indépendantes) dont l'analyse et la synthèse sont étudiées dans les chapitres 4, 5 et 6.

Enfin, l'étude générale des systèmes logiques asynchrones (chapitre 7) repose sur la définition de l'élément de délai inertiel et traite de l'analyse des assemblages quelconques sans boucle de rétroaction, puis avec boucles, ainsi que du cas particulier des bascules bistables et des systèmes séquentiels synchrones ou quasi-synchrones.



Marcel Imband

Daniel Mange est né à Lausanne (Suisse) en 1940. Il a poursuivi ses études à l'Ecole polytechnique fédérale de Lausanne où il a reçu les diplômes d'ingénieur électricien (1964) et de docteur ès sciences techniques (1968). Assistant, puis chargé de cours, il a été nommé professeur dans cette même Ecole en 1969; il est titulaire de la Chaire de systèmes logiques qui a édité de 1969 à 1975 six numéros de la revue "Systèmes logiques". Ses activités d'enseignement et de recherche sont désormais axées sur l'étude comparée des systèmes logiques câblés et programmés.

# Traité d'Électricité

I INTRODUCTION À L'ÉLECTROTECHNIQUE

II MATÉRIAUX DE L'ÉLECTROTECHNIQUE

III ÉLECTROMAGNÉTISME

IV THÉORIE DES RÉSEAUX DE KIRCHHOFF

V ANALYSE ET SYNTHÈSE DES SYSTÈMES LOGIQUES

VI THÉORIE ET TRAITEMENT DES SIGNAUX

VII DISPOSITIFS À SEMICONDUCTEUR

VIII ÉLECTRONIQUE

IX ÉLECTROMÉCANIQUE

X MACHINES ÉLECTRIQUES

XI MACHINES SÉQUENTIELLES

XII ÉNERGIE ÉLECTRIQUE

XIII HYPERFRÉQUENCES

XIV CALCULATRICES

XV ÉLECTRONIQUE DE PUISSANCE

XVI ÉLECTRONIQUE DE RÉGLAGE ET DE COMMANDE

XVII SYSTÈMES DE MESURE

XVIII SYSTÈMES DE TÉLÉCOMMUNICATIONS

XIX FILTRES ÉLECTRIQUES

XX TRAITEMENT NUMÉRIQUE DES SIGNAUX

XXI ÉLECTROACOUSTIQUE

XXII HAUTE TENSION

