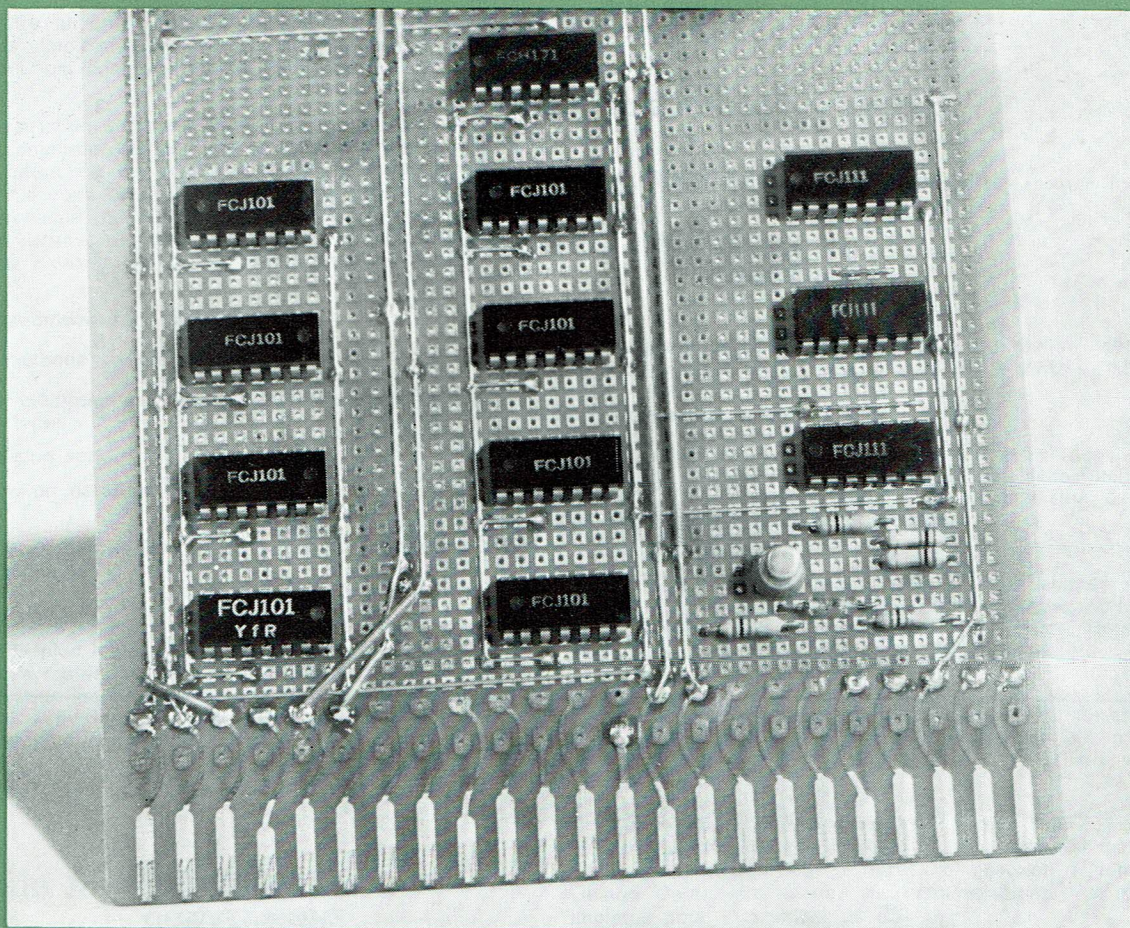


La conversion des grands nombres décimaux en nombres binaires

62 | 1970



Lorsqu'il est nécessaire d'effectuer plusieurs opérations sur un nombre binaire, on s'aperçoit qu'il est plus économique d'utiliser le code binaire pur, plutôt que le code binaire décimal (BCD).

Cette information a pour but de présenter quelques méthodes de codage d'un nombre décimal, si grand soit-il, en nombre binaire pur.

TABLES DES MATIERES

pp.

CHAPITRE I — INTRODUCTION

3

CHAPITRE II — METHODES DE CONVERSION

3

II 1 : LA METHODE DU DECOMPTAGE

3

II 2 : LES METHODES PAR DIVISIONS SUCCESSIVES

4

II 2 - 1 : La conversion série

5

II 2 - 2 : La conversion parallèle

8

CHAPITRE III — COMPARAISON DES TROIS METHODES

10

CHAPITRE I — INTRODUCTION

Il est aisé et relativement peu onéreux de coder un nombre décimal d'un chiffre significatif en code binaire, à l'aide de portes NAND. On peut ensuite effectuer les opérations arithmétiques souhaitées sur le résultat, puis repasser au code décimal.

Mais le problème se complique à mesure que le nombre à coder comporte de plus en plus de chiffres significatifs.

Le code binaire décimal traite en effet chaque chiffre séparément, ce qui exige 4 digits par chiffre, tandis que le code binaire pur traite le nombre complet N , ce qui exige n digits, avec la relation : nombre à coder $N = 2^n$.

Ainsi, par exemple, le nombre 1024 est codé en binaire décimal 1248 par 0001.0000.0010.0100 soit par 16 digits, alors qu'en code binaire pur 10 digits suffisent ($1024 = 2^{10}$).

Par conséquent, lorsqu'une opération arithmétique doit se faire sur un nombre décimal converti, il est beaucoup plus simple d'opérer sur sa « traduction » en code binaire pur, qui comporte moins de digits.

Une restriction s'impose cependant : toutes les décades de comptage, de par leur principe même, fournissent la traduction du nombre décimal en code binaire décimal. Le code binaire pur n'est donc jamais directement accessible. Or, comme c'est précisément ce code que l'on désire, il faut transformer les informations reçues, et c'est le but des schémas décrits ci-après.

Trois méthodes sont applicables :

1. La méthode du décomptage

2. Les méthodes par divisions successives qui sont de deux types :

la conversion série

la conversion parallèle

Nous examinerons les avantages et les inconvénients de ces méthodes.

Dès l'abord, signalons que la méthode du décomptage est la plus lente, tandis que la conversion parallèle est la plus rapide. Un tableau comparatif permet, en fin de publication, de se rendre compte des caractéristiques des trois méthodes, pour quelques cas pratiques, avec une estimation du coût des circuits.

CHAPITRE II — METHODES DE CONVERSIONS

II. 1. LA METHODE DU DECOMPTAGE

Cette méthode consiste à mémoriser le nombre à convertir dans un compteur décimal travaillant en décompteur et associé à un compteur binaire.

Un générateur d'impulsions synchronise le mouvement des deux appareils. Ainsi, chaque impulsion de décomptage fait avancer le compteur binaire associé. Lorsque le décompteur atteint la position zéro, l'état du compteur binaire représente le nombre binaire recherché.

Cette méthode est la plus simple, mais aussi la plus lente, puisqu'il faut attendre $N-1$ périodes du générateur de synchronisation (clock generator) avant de connaître la réponse.

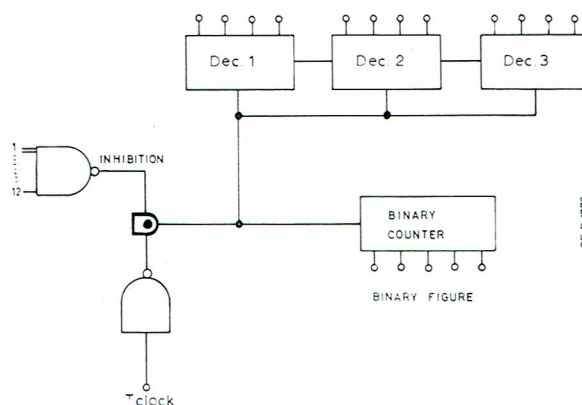


Fig. 1 — Principe de la conversion par décomptage, pour un nombre inférieur ou égal à 256.

La figure 1 montre une possibilité de réaliser cette conversion pour un nombre décimal de 3 chiffres, inférieur à 256.

Le nombre décimal est emmagasiné dans 9 décades (trois par chiffre) qui sont en fait des décompteurs asynchrones ; le nombre binaire est emmagasiné dans un compteur binaire pur ;

Un signal commun de récurrence T fait avancer le compteur et le décompteur. Dès que ce dernier affiche 0, la porte P inhibe l'impulsion T ; le nombre binaire cherché peut être lu dans le compteur binaire pur.

Les schémas détaillés de ces compteurs et décades sont décrits dans nos « Application Notes » n°s 006, 010, 016, 021, 045).

La durée du comptage vaut $(N-1) T$ soit $225 T$. Pour les circuits intégrés de la série FC, la fréquence maximale de comptage est 3 MHz en pratique (System Design Data, voir feuilles de caractéristiques) ce qui implique pour T la valeur de 330 ns.

La conversion suivant le schéma 1 nécessite

18 circuits FCJ 111

+ 3 circuits FCH 161

+ 8 diodes BAX 13

II. 2. LES METHODES PAR DIVISIONS SUCCESSIVES

Un autre procédé de conversion consiste à poursuivre la division par 2 du nombre décimal jusqu'à obtenir 0 comme résultat. En effet, un nombre décimal Z_0 se représente dans le système binaire par la relation suivante :

$$Z_0 = b_n 2^n + b_{(n-1)} 2^{n-1} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

La division par 2 de Z_0 donne un nombre entier Z_1 et un reste $\frac{1}{2} R_1$. On a :

$$\frac{1}{2} Z_0 = Z_1 + \frac{R_1}{2} = b_n 2^{n-1} + \dots + b_1 2^0 + b_0 2^{-1}$$

En comparant les deux membres de cette égalité, on constate que le reste R_1 correspond au plus petit nombre binaire b_0 .

Chaque division par 2 donne un nouveau nombre binaire et un reste qui ne peut être que 0 ou 1.

Exemple : conversion du nombre 345

	reste
345	
172	1
86	0
43	0
21	1
10	1
5	0
2	1
1	0
0	1

La traduction de 345 en code binaire pur s'obtient en lisant les restes de bas en haut :

$$345 = 101011001$$

La conversion du nombre 345 nécessite donc neuf divisions successives ; ce serait prohibitif, si la division par deux n'était particulièrement simple : on divise par deux le chiffre du rang le plus élevé (3). Le reste éventuel (1) est ajouté à la décimale immédiatement inférieure, compte tenu de son poids : $10 + 4 = 14$.

Cette valeur est encore divisée par deux. Le nouveau reste (0) est ajouté au dernier chiffre (5) ; une dernière division par deux donne un reste (1) qui est le premier digit binaire.

La division par deux peut encore être réalisée par divisions simultanées. Chaque chiffre est divisé simultanément par deux. Si un reste apparaît dans une décade, on ajoute 5 à la décade inférieure. Le reste de la dernière décade constitue le digit binaire correspondant.

Reprenons l'exemple de 345.

- a) On divise 3 par 2, ce qui fait 1, reste 1
 On divise 4 par 2, ce qui fait 2, reste 0
 On divise 5 par 2, ce qui fait 2, reste 1

Ces trois divisions se font simultanément. On obtient le nombre 122.

La première division a donné un reste 1, on ajoute donc 5 au deuxième chiffre, ce qui fait 172.

La dernière division a donné un reste 1 qui constitue le premier digit binaire.

- b) On procède de même avec 172

- $1 : 2 = 0$ reste 1 ; on ajoute 50
 $7 : 2 = 3$ reste 1 ; on ajoute 5
 $2 : 2 = 1$ reste 0.

Le résultat total est 31, auquel on ajoute le total de 55, ce qui fait 86. Le dernier reste était 0, ce qui donne le deuxième digit binaire. On continue ces divisions jusqu'à ce qu'un résultat donne 0 ; le reste, dans ce cas-ci, est 1, ce qui donne le dernier digit binaire.

Comme nous l'avons signalé plus haut, le nombre décimal est toujours décodé par des décades. c'est-à-dire en code binaire décimal 1248. Les sorties de ces décades sont appelées D, C, B, A dans le tableau 1.

Tableau 1

	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

On voit que la division par deux peut se faire très simplement par décalage des digits vers la droite.

Nous exploiterons cette particularité pour deux types de conversion.

La conversion série et la conversion parallèle.

II. 2.-1 La conversion série

Chaque chiffre est mis en mémoire dans une décade-registre (RD) particulière, qui peut également fonctionner en registre à décalage (shift register, voir fig. 2).

On opère ensuite une division par 2 sur chaque chiffre, ce qui signifie un simple décalage à droite de l'information emmagasinée dans les registres comme expliqué au tableau 1.

Ce décalage est commandé par le circuit H.

Les restes des deux premières divisions sont emmagasinés dans les bascules FR 3 et FR 2 qui, éventuellement, bloquent le circuit de commande de décalage et ouvrent l'étage de commande d'un compteur « modulo 5 » (M5) qui ajoute 5 impulsions à la décade suivante. Un circuit monostable fait ensuite rebasculer FR 3 ou FR 2 de manière à bloquer à nouveau le circuit de commande de M5. Le dernier reste est affiché dans le registre d'adresse RA.

Le quotient de la division est à nouveau affiché dans les décades, et la succession des divisions se poursuit jusqu'à ce que le nombre binaire recherché se trouve complètement dans le registre d'adresse.

A l'exception de la décade d'ordre le plus élevé, qui est uniquement un registre à décalage, les décades suivantes constituent simultanément des registres (entrées R) et des compteurs (entrées T).

Réalisation à l'aide de circuits intégrés de la famille FC

La figure 3 donne le schéma de la conversion d'un nombre inférieur à 256 (256 mots ou bits), réalisée à l'aide d'éléments de la famille FC, qui sont des circuits intégrés DTL.

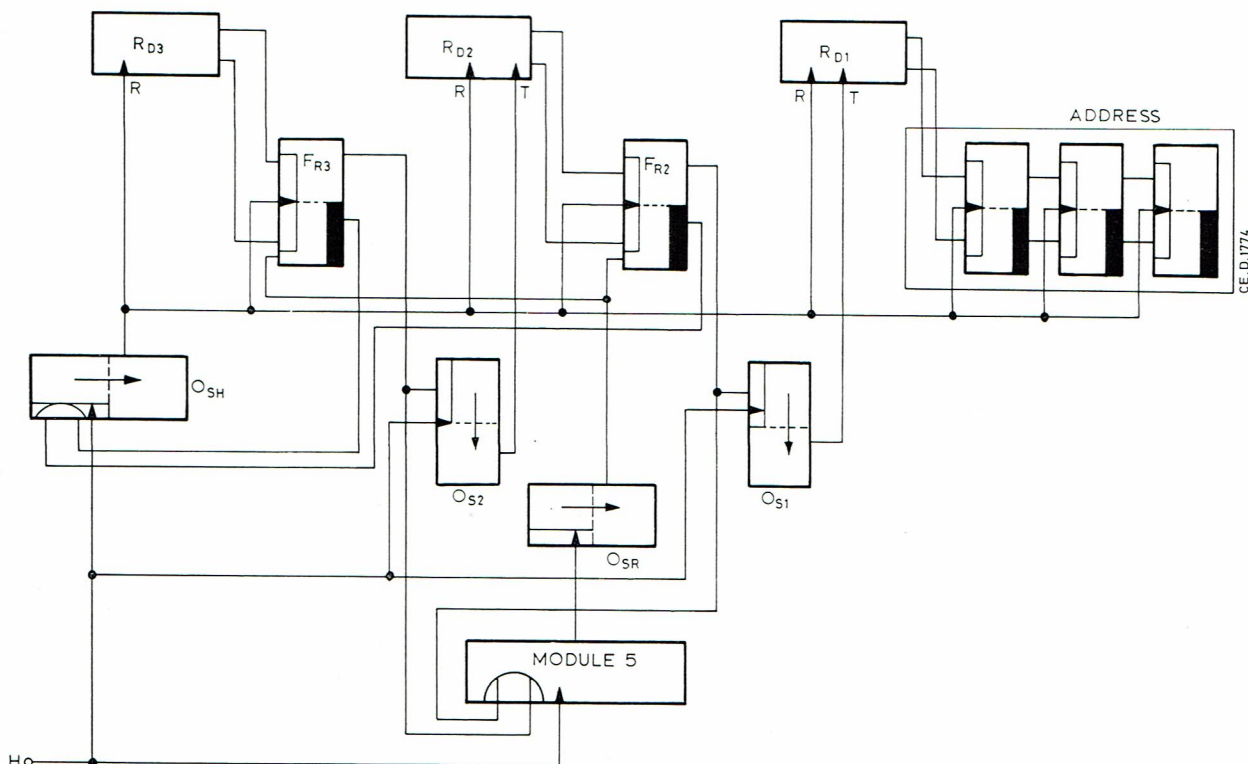


Fig. 2 — Principe de la conversion série.

Le nombre à convertir est mémorisé dans les trois décades-registres RD. L'étage de commande TS fournit l'impulsion de décalage qui réalise la division par deux ; les circuits de commande TC1 et TC2 fournissent les impulsions d'avance des décades D1 et D2. Le circuit de commande TCC libère le compteur « modulo 5 » lorsqu'une des bascules FR1 ou FR2 a changé d'état, indiquant ainsi qu'une des divisions par deux a donné un reste.

Le circuit TR sert à remettre ce compteur, ainsi que FR3 et FR2 à zéro. A chaque impulsion de décalage, on obtient une place vide qui se déplace dans le registre de la décade d'ordre le plus élevé vers celle d'ordre le plus bas.

On peut ainsi se passer du registre de sortie RA et mémoriser les digits binaires successifs dans les décades-registres initiales.

Dans ce but, le compteur Z2 enregistre les impulsions de décalage et termine le processus après 8 divisions ($256 = 2^8$).

En outre, après deux impulsions de décalage, il réalise l'interconnexion des décades D3 et D2 en registre ; après 6 impulsions, il fait de même avec les décades D1 et D2.

Le temps nécessaire est de 32 T.

La figure 4 donne le schéma de la conversion série de 1024 mots.

Après les 1^{re}, 5^{es} et 9^{es} impulsions de décalage, les décades D4 et D3, D3 et D2, D2 et D1 sont respectivement raccordées en registres, et les bascules FR4, FR3 et FR2 sont respectivement maintenues à l'état 0.

Le temps nécessaire est de 46 T.

La figure 5 décrit une décade qui peut être utilisée aussi bien en registre qu'en compteur binaire.

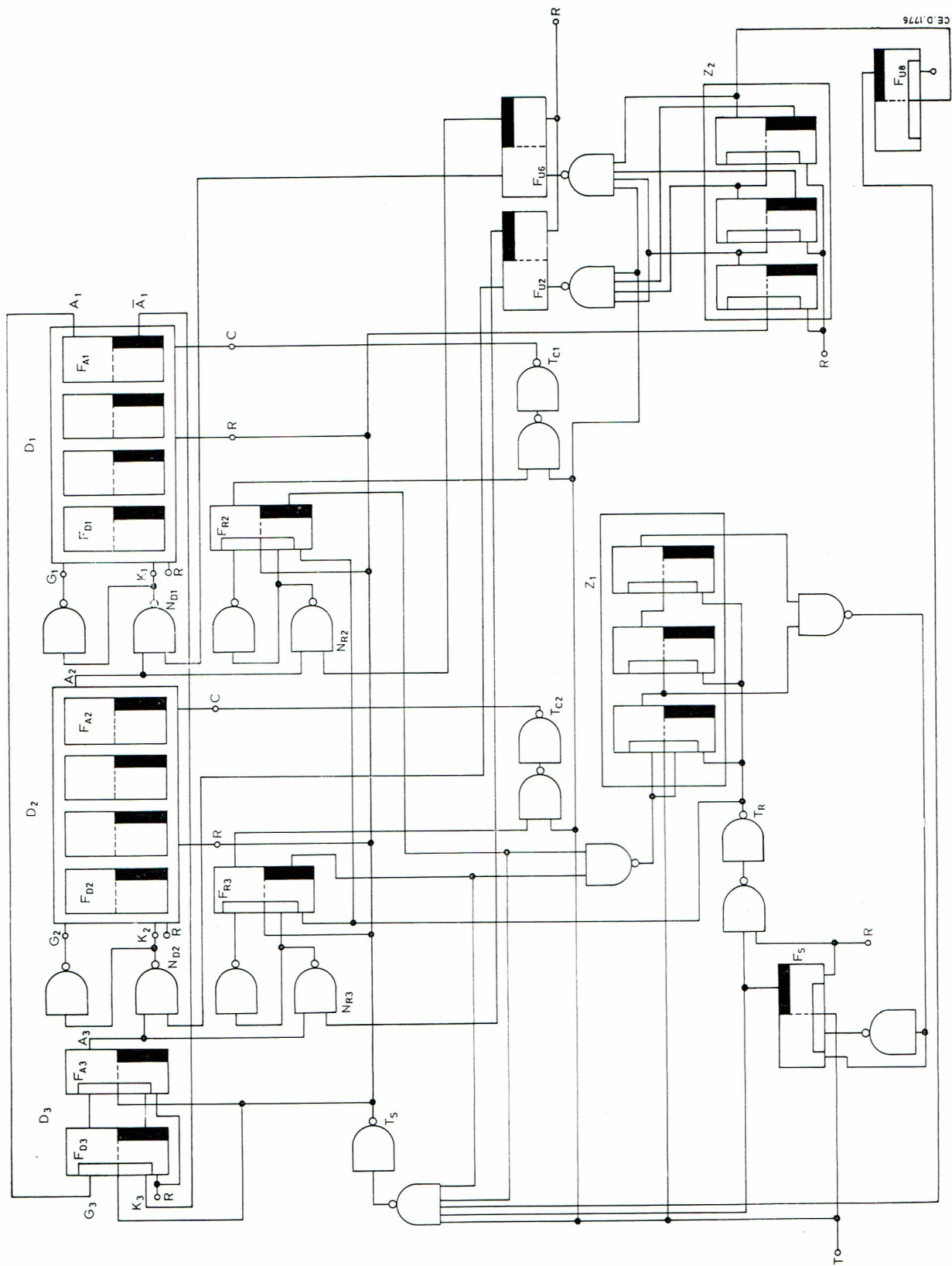


Fig. 3 — Conversion série de 256 mots.

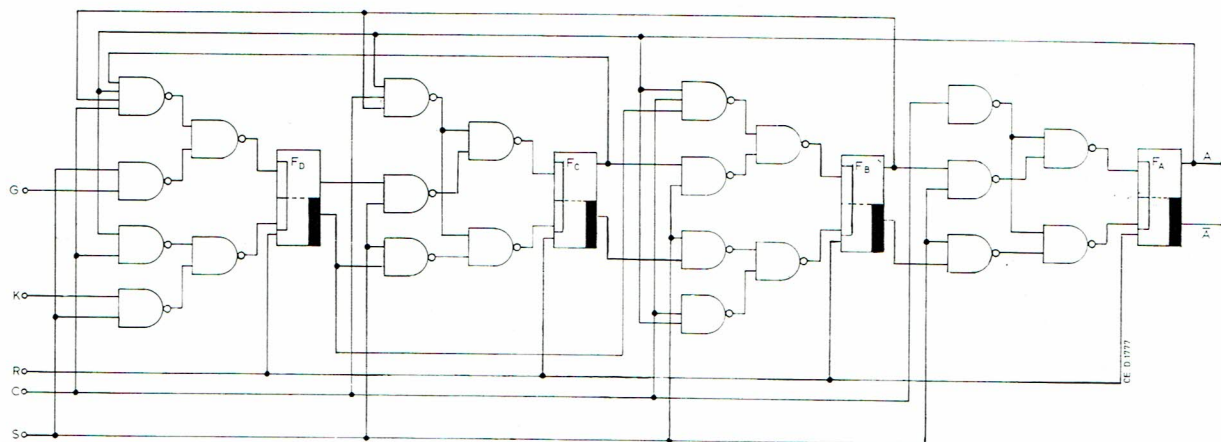


Fig. 5 — Décade utilisable en registre et en compteur binaire.

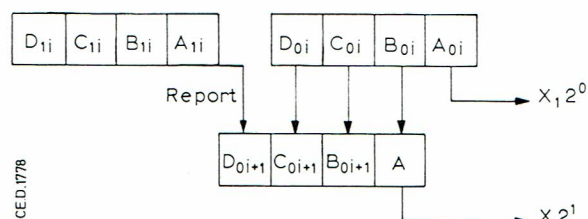


Fig. 6 — Principe de la conversion parallèle.

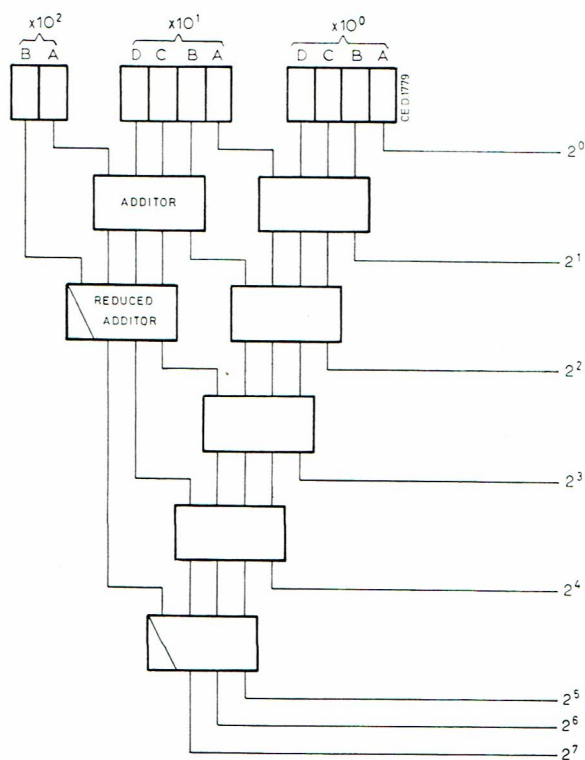


Fig. 7 — Conversion parallèle de 2^7 mots.

II. 2.2. Conversion parallèle

La division par deux d'un nombre écrit dans le code décimal 1248 peut aussi se faire de manière telle que le bit de poids le plus faible de la décade (colonne A

dans le tableau 1) représente le reste, et que le quotient de la $(i + 1)^{\text{e}}$ division soit

$$A_i + 1 = B_i$$

$$B_i + 1 = C_i$$

$$C_i + 1 = D_i$$

Ce principe est illustré par la figure 6. Si ce nombre se compose de plusieurs décades, et si un reste apparaît dans celles-ci (à l'exception de la décade inférieure), on ajoute 5 dans une additionneuse au nombre divisé par 2 de la décade suivante; le nombre ainsi obtenu est ensuite soumis à la même procédure, et ainsi de suite.

Par divisions successives par deux, on obtient les différents chiffres du nombre binaire. Ils sont chaque fois égaux au reste qui apparaît lors de la division par deux effectuée dans la décade inférieure.

Le temps nécessaire à la conversion de 256 mots est de $5 t_D$, où $t_D = t_{fd} + t_{rd}$ (voir les caractéristiques de nos circuits intégrés).

Constitution de l'additionneuse

Le but d'une additionneuse est d'ajouter 5 au nombre divisé par 2 d'une décade, si un reste apparaît lors de la division par 2 de celle immédiatement supérieure. (par exemple $A_{1i} = 1$)

La figure 7 montre comment on utilise les additionneuses dans un convertisseur pour 2^7 mots.

Une additionneuse comporte 15 portes NAND. La figure 8 en donne le schéma complet.

Aux sorties, on exploite la possibilité de se servir de la fonction OU connectée. Si la somme de l'additionneuse, c'est-à-dire $D_{0i} + 1$, vaut 0, ou peut se passer de la sortie $D_{0i} + 1$.

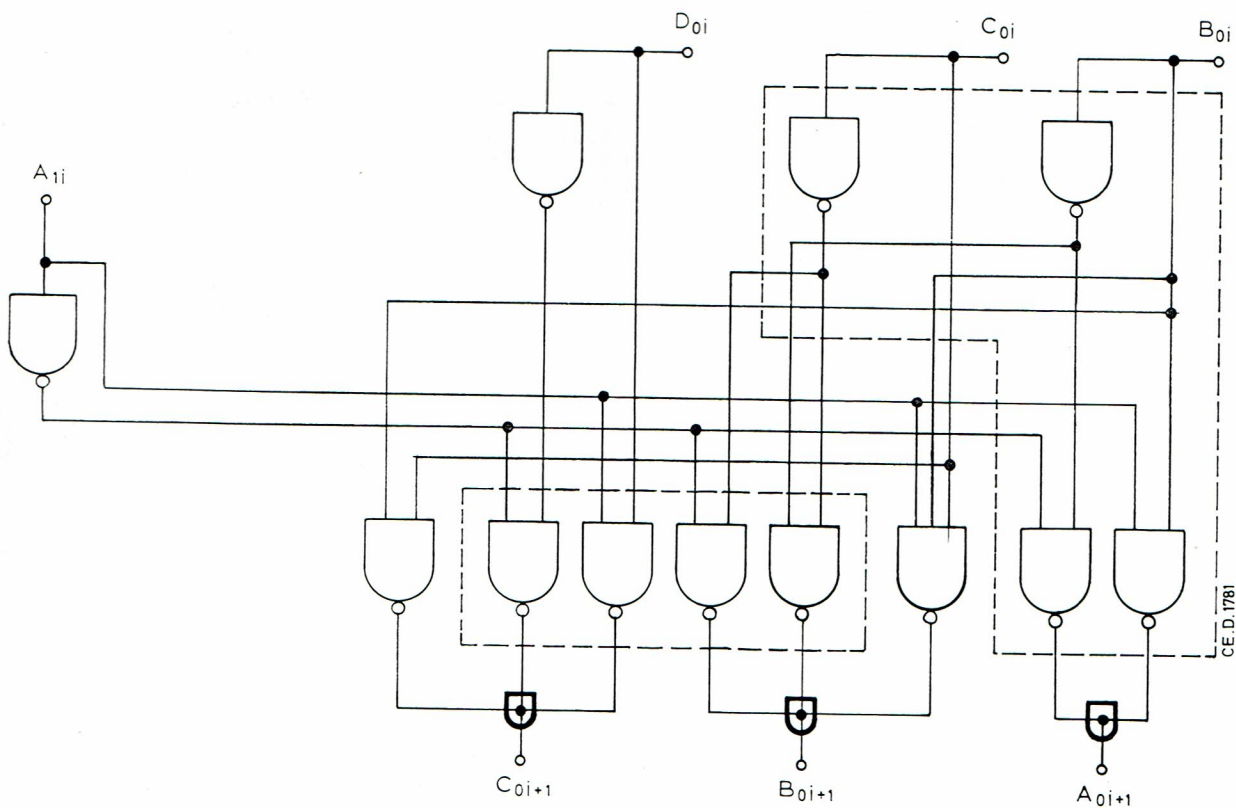
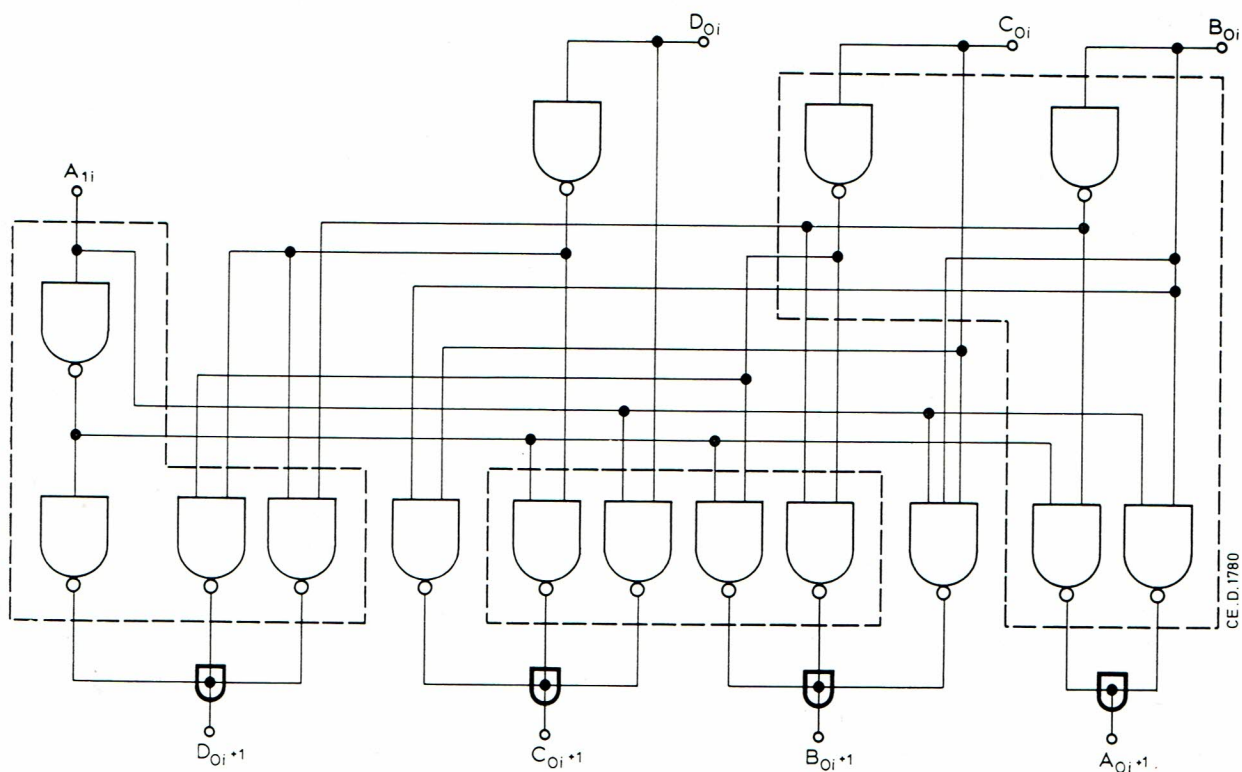
Dans ce cas, on obtient une additionneuse réduite suivant la figure 9, elle ne comporte que 12 portes NAND.

Réalisation à l'aide d'éléments de la série FC

On utilise des portes FCH 191 et FCH 161. Pour l'additionneuse complète de la figure 8, on assemble trois fois quatre portes NAND avec deux entrées (FCH 191); les trois NAND restant à réaliser sont fournies par un circuit FCH 161.

L'additionneuse réduite de la figure 9 est construite à l'aide de trois FCH 191, et d'une FCH 161.

Les portes non utilisées, (par exemple FCH 161) peuvent éventuellement être utilisées dans d'autres additionneuses.



CHAPITRE III — COMPARAISON DES TROIS METHODES DE CONVERSION

Le tableau n° 2 qui suit permet de se rendre compte des avantages respectifs des trois méthodes, si l'on réalise la conversion de trois nombres (256, 1024 et 4096), suivant les trois principes exposés, et avec nos circuits intégrés de la famille FC.

En consultant les caractéristiques particulières de ces circuits on peut fixer pour les temps nécessaires les valeurs :

$$T = 330 \text{ ns } (f_H = 3 \text{ MHz})$$

$$t_D = 300 \text{ ns}$$

Une estimation du coût de chaque réalisation est indiquée, sur base des prix actuels de vente au détail.

On y voit clairement que la méthode du décomptage est la plus économique, mais la plus lente, tandis que la conversion parallèle est la plus rapide, mais la plus onéreuse, à mesure que le nombre de mots augmente.

Tableau 2

	256 mots			1024 mots			4096 mots		
Principe	Durée	Circuit	Prix	Durée	Circuit	Prix	Durée	Circuit	Prix
Décomptage	255 T = 84,2 μ s	18 FCJ 111 3 FCH 161 8 BAX 13	2.534 F	1023 T = 337,5 μ s	23 FCJ 111 4 FCH 161 13 BAX 13	3.264 F	4095 T = 1,044 ms	27 FCJ 111 4 FCH 161 15 BAX 13	3.769 F
Conversion série	23 T = 10,5 μ s	20 FCJ 111 1 FCH 111 1 FCH 131 2 FCH 161 14 FCH 191 1,5 FCH 221 4 BAX 13 1 générateur de commande	4.232 F	46 T = 15,2 μ s	24 FCJ 111 1 FCH 111 1 FCH 131 1 FCH 161 1 FCH 171 20,75 FCH 191 7 BAX 13 2 FCH 221 1 générateur de commande	5.383 F	54 T = 17,8 μ s	26 FCJ 111 1 FCH 111 1 FCH 131 1 FCH 161 1 FCH 171 20,75 FCH 191 7 BAX 13 2 FCH 221 1 générateur de commande	5.633 F
Conversion parallèle	5 t_D = 1,5 μ s	20,5 FCH 191 5,67 FCH 161	2.327 F	7 t_D = 2,1 μ s	35,25 FCH 131 9 FCH 161 1 FCH 171	4.030 F	9 t_D = 2,7 μ s	53,25 FCH 191 15 FCH 161 1 FCH 171	6.164 F