INITIATION A L'ELECTRONIQUE

Suite voir n° 1758

ADDITION DE DEUX NOMBRES DE PLUS DE 4 BITS

Le montage de la figure 78 ne convient que pour des nombres de 4 bits chacun (au moins). Cela limiterait singulièrement les possibilités, puisque, avec ces nombres de 4 bits, on ne peut exprimer que de 0 à 15.

Aussi est-il indispensable de pouvoir additionner des nombres comportant un nombre de bits plus grand. Fort heureusement, c'est très facile avec des additionneurs du type HEF 4008, par exemple.
En effet, il suffit d'utiliser « n » circuits 4008 quand le plus grand des nombres à additionner comporte moins de 4 n (et plus de 4n–4 bits).

On les monte comme l'indique la figure 79, qui suppose que l'on se limite à des nombres de 12 bits (puisqu'on y emploie trois additionneurs), mais le montage peut évidemment s'étendre à des nombres bien plus grands.

Le fonctionnement de cet ensemble est facile à comprendre. Les deux nombres à additionner, A et B, comportent chacun 12 bits, de A_0 à A_{11} pour A, et B_0 à B_{11} pour B.

L'additionneur qui est commandé par les poids minimaux (de A_0 à A_3 et de B_0 à B_3) a son entrée CI à la masse, puisqu'il n'y a pas de retenue à introduire là, exactement comme dans le cas de la figure 78. Il fournit donc les chiffres de poids minimaux de la somme, de Σ_0 à Σ_3.

Il fournit aussi une retenue, sur sa sortie C_0, mais, cette fois, on ne la considère plus comme le MSB de la somme (ce que nous avions fait dans le cas de la figure 78, parce qu'il s'agissait de deux nombres de 4 bits). Cette retenue est donc appliquée à l'entrée retenue (CI) du second additionneur, le circuit (2). Ce second additionneur nous fournit les chiffres de rang 4 à 7 (du cinquième au huitième chiffre, car il ne faut pas oublier que l'on commence au « rang zéro ») de la somme, plus une retenue.

Cette dernière est appliquée à l'entrée retenue du circuit (3), recevant les MSB des deux nombres. Comme on ne dépasse pas les nombres de douze bits, la sortie retenue du circuit (3) est, cette fois, considérée comme le 13e bit de la somme, d'où sa désignation sous le repère Σ_{12}.

LA « RETENUE ANTICIPEE »

Le montage de la figure 79 se généralise, nous l'avons dit, à un nombre quelconque de bits. Pour des nombres de 40 bits, il faudrait dix additionneurs, et la complexité du montage n'a rien d'inquiétant. Mais, dans ce cas, il peut intervenir une difficulté inattendue : les circuits ne vont pas assez vite. Vous le savez bien, les électroniciens sont toujours pressés : quand on leur dit qu'ils devront attendre une information pendant une microseconde, ils répondent que l'information est bien destinée à eux, pas à leur arrière-arrière-petit-fils.
Alors, supposons que, dans le cas de la figure 79, le nombre A soit fait entièrement de 1 (soit FFF en hexadécimal), le nombre B étant composé entièrement de 0. La somme est, bien entendu, égale à A, soit à $FFF (rappelons que le symbole $ qui représente normalement le dollar, signifie, quand il précède un nombre, que ce dernier est exprimé en hexadécimal).
Maintenant, modifions B pour le faire passer de 000 à 001 (en changeant uniquement B_0, qui passe de 0 à 1). La somme devient alors :

$1000

Normal, n'est-ce pas ? Oui, mais... la sortie retenue, C_0, du circuit (1) ne va pas passer à 1 au moment précis où l'on appliquera le niveau un à l'entrée B_0, car il faudra le temps qu'elle « transite » par les quatre tranches d'addition contenues dans le circuit (1).

Elle abordera donc le circuit (2) avec un certain retard. Ce circuit va ajouter le sien, car il faudra aussi un temps de passage pour que ses quatre sorties passent au 0 et que sa sortie C_0 passe au 1. Enfin, le circuit (3) ajoutera son retard de transmission à ceux des deux autres.
Nous ne disposerons donc de la valeur vraie de la somme qu'au bout... d'un temps fou (précisons que ce temps est tout de même inférieur à 150 ns, ou 0,15 µs, pour des HEF 4008 alimentés sous 12 V).

« Inadmissible ! », déclarera donc un électronicien, pensant même avec horreur que, dans le cas de dix circuits, ce retard « monstrueux » atteint une demi microseconde (de quoi se faire des cheveux blancs).
La situation est-elle insoluble ? Vous vous doutez bien que, si c'était le cas, l'auteur aurait lâchement laissé ce point dans l'ombre.

On réduit ce retard à une valeur bien plus courte par des circuits supplémentaires, qui, tenant compte des valeurs des chiffres de A et de B, fournissent la retenue de poids maximal avec un retard égal ou même inférieur à celui qui caractérise la sortie de la première retenue.

Ces circuits se nomment, en traduction un peu libre « regardant dans l'avenir pour donner une retenue anticipée » (« Anticipated look ahead carry » en VO). Mais leur fonctionnement ne nécessite ni boule de cristal ni marc de café.

En effet, on peut réaliser, par des ensembles de portes recevant sur leurs entrées les bits des nombres, des circuits qui ne traitent que les retenues. Ils ont l'avantage de fournir ainsi la retenue à peu près à l'instant où les additionneurs (toujours utilisés de leur côté), donnent les chiffres de la somme.
Nous nous garderons bien de donner une idée de leur structure, celle-ci relevant du

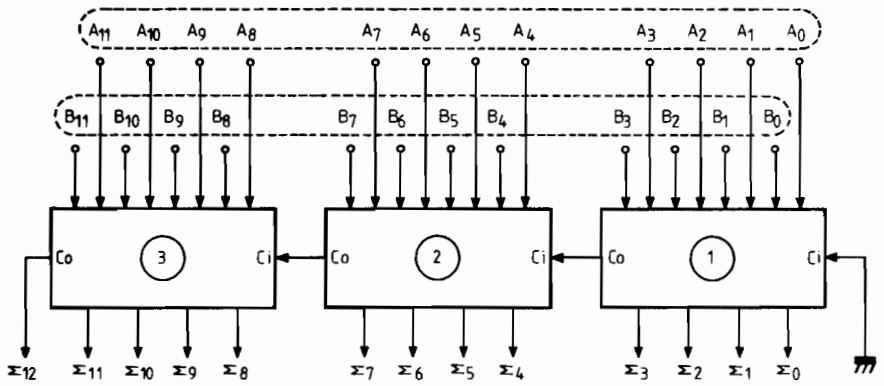

Fig. 79. – Additionneur composé, pour mots de 12 bits.

« cauchemar d'un électronicien malade après une nuit d'orgie », l'essentiel étant de savoir que ces circuits existent et que l'on peut donc obtenir très vite les bits de la somme.

CAS DE L'ADDITION DE PLUS DE DEUX NOMBRES PARALLELE

L'additionneur tel que le représentent les figures 78 et 79 est prévu pour ajouter *deux* nombres. Il a des « entrées A » et des « entrées B ». Or, il y a bien des cas où l'on désire faire la somme de trois, quatre... ou n nombres.
Comme c'est souvent le cas, l'électronique utilise alors la solution la plus... bête. Comme les additionneurs ne peuvent ajouter trois nombres, M, N et P, par exemple, ils ajoutent M et N, ce qui donne une somme S, puis ils ajoutent le nombre P à la somme S.
L'astuce utilisée pour arriver à ce résultat est intéressante, et elle illustre fort bien les possibilités des basculeurs D, dont nous avons déjà parlé (voir *Le Haut-Parleur*, n° 1757, octobre 1988, pages 68 à 70).
Le montage, que nous avons limité à 4 bits, pour ne pas surcharger le schéma, mais qui se généralise à un nombre quelconque de bits, est celui que représente la figure 80.
On voit que l'additionneur reçoit les différents nombres à ajouter sur ses entrées A (de A_0 à A_3). Les sorties somme, de Σ_0 à Σ_3, attaquent les entrées D de quatre basculeurs D, jouant le rôle de mémoire temporaire de la somme partielle.
Ce sont les sorties Q de ces basculeurs D qui vont attaquer les entrées « B » de l'additionneur. Le tout semble complexe, et l'on a un peu de peine à en démêler le fonctionnement, comme toujours quand un ensemble paraît « bouclé sur lui-même », mais nous allons voir que, en fait, on comprend facilement le déroulement des opérations.

COMMENCONS PAR... AJOUTER ZERO AU PREMIER NOMBRE

Nous supposerons que nous avons commencé par envoyer une impulsion sur la ligne « Z », mettant au 0 les quatre basculeurs D, en agissant sur leurs entrées « Clr ». Leurs sorties Q sont donc au 0, et il y a 0 sur toutes les entrées B de l'additionneur.
Donc, si nous appliquons, sur les entrées A, un premier nombre, M, de 4 bits, comme on lui « ajoute 0 », nous retrouverons ce même nombre sur les sorties Σ . Il est donc présent sur les entrées D des quatre basculeurs, mais ceux-ci n'en tiennent pas compte, n'ayant pas reçu de signal d'horloge.
Envoyons donc ce signal sur la commande H. Les basculeurs D réagissent, transférant chacun sur sa sortie Q ce qu'il recevait sur son entrée D au moment du front montant du signal en H.
On pourrait penser qu'il va se produire une horrible pagaille : imaginez donc que le nombre M, qui vient d'être transféré sur les sorties Q des basculeurs, se trouve alors appliqué aux entrées B de l'additionneur, alors qu'il est encore appliqué sur les entrées A. Donc, l'additionneur ajoute M à lui-même, donnant, sur ses sorties, le nombre M + M (soit le double de M), qui va être immédiatement appliqué aux entrées D des basculeurs. Alors que vont faire ces derniers ?
Justement : ils ne feront... rien du tout. N'oubliez pas qu'un basculeur D est sensible à la valeur de son entrée D jusqu'au moment (exclu) où il reçoit une commande sur son entrée horloge (entrée CK). Il transfère alors, sur sa sortie Q, ce qu'il avait reçu sur son entrée D, mais *il cesse d'être sensible à son entrée D dès que la commande sur CK agit* et même un tout petit peu avant (rappelez-vous les problèmes de temps de pré-établissement et de maintien).
Donc, quand on a envoyé une impulsion en H, les bits du nombre M sont effectivement appliqués aux entrées B de l'additionneur, il y a effectivement un changement des sorties de ce dernier, puisqu'il ne reçoit plus M et 0, mais M et M. Mais, alors, les basculeurs D ne se soucient plus de ce qu'ils reçoivent sur leurs entrées D.
Le nombre M est maintenant appliqué aux entrées B de l'additionneur. Supposons que, à ce moment, nous cessions d'appliquer, sur les entrées A, le nombre M, pour y appliquer le nombre suivant, N. Les sorties de l'additionneur vont alors afficher la somme S = M + N, puisque le nombre N est appliqué aux entrées B de l'additionneur par les sorties Q des basculeurs D, alors que N est appliqué sur les entrées A.

LE DEUXIEME COUP DE L'HORLOGE

Tout est prêt pour recevoir un nouveau top d'horloge. Les valeurs présentes sur les entrées D sont alors transférées sur les sorties Q. On applique donc aux entrées B de l'additionneur les bits de la somme S = M + N.
De nouveau, on peut craindre qu'il se produise un « horrible cafouillage ». En effet, puisque N est toujours appliqué aux entrées A de l'additionneur, la somme M + N étant maintenant appliquée sur ses entrées B, on trouve donc :

M + N + N, ou : M + 2 × N

sur les sorties de l'additionneur ; mais les basculeurs D, de nouveau, ont la sagesse de ne tenir aucun compte de ces valeurs sans intérêt.
Ils ont fait consciencieusement leur travail : ils ont pris en mémoire la somme M + N, qui est

maintenant présente sur les entrées B, prête à être additionnée au nombre P, quand on appliquera ce dernier aux entrées A. Un troisième coup d'horloge, à ce moment-là, ira transférer sur les sorties Q des basculeurs la somme (M + N) + P, soit M + N + P. Astucieux, n'est-ce pas ?
Nous ne serions pas étonnés que cela vous rappelle quelque chose. Mais oui, cherchez dans votre mémoire (pas dans la RAM, dans vos neurones), et vous allez évoquer l'analogie avec l'instruction Basic :

(LET) U = U + V

(le LET est souvent facultatif) qui signifie, sous une forme très condensée :
« Prenez le contenu de la mémoire désignée par U, ajoutez-y le contenu de la mémoire désignée par V, et logez le résultat dans la mémoire désignée par U ».
Dans le cas du montage de la figure 80, les basculeurs D ont joué le rôle de cette mémoire désignée par U, puisqu'on en sort une des données et qu'on y rentre le résultat.
Il est d'ailleurs à noter que pas mal de gens protestent contre cette notation, qui sépare par un signe « égal » des choses qui ne sont manifestement pas égales. C'est pourquoi le langage Pascal, plus correct, note cela :

U : = U + V

ce qui choque moins nos habitudes.

LE PROCESSUS PAR «ITERATION «

On voit ici l'emploi d'une technique d'addition qui utilise la répétition d'une manœuvre autant de fois qu'il est nécessaire. C'est ce que l'on nomme (quand on veut avoir l'air cultivé) un « processus par itération ».
Il s'agit d'une méthode extrêmement utilisée. Supposez que, par exemple, vous ayez besoin des nombres binaires se succédant de sept en sept, comme 0, 7, 14, 21, 28, etc. (ou plutôt $000, $007, $00E, $015, $01C, etc.), la solution la plus simple est d'utiliser le montage de la figure 80 (généralisé à plus de 4 bits), en appliquant en permanence un niveau haut aux entrées A_0, A_1 et A_2 (0111 cela fait sept). A chaque impulsion d'horloge, les sorties seront « incrémentées » de 7.

« Idée bizarre », diront certains, je n'ai jamais besoin d'une telle suite ». Si telle est leur idée, qu'ils pensent une nouvelle fois à la programmation, en particulier à l'instruction « basico-pascalienne » (commune aux deux langages) :

FOR I = 1 TO 60 STEP 7...

qui correspond bien à ajouter 7, à chaque « tour », au contenu de la mémoire désignée par I.

On va retrouver ce processus dans la commande des tables traçantes. Si l'on veut qu'une telle table dessine une ligne biaise, on calcule combien on doit ajouter à la coordonnée X et à la coordonnée Y à chaque pas, et c'est un système analogue à celui de la figure 80 qui s'en charge pour X, un autre identique pour Y.

PASSONS A LA SOUSTRACTION

Quand on voit qu'il existe des circuits intégrés réalisant l'addition, on pense logiquement qu'il doit en exister pour les soustractions. Or, ce n'est pas le cas. Alors, comment faire ?

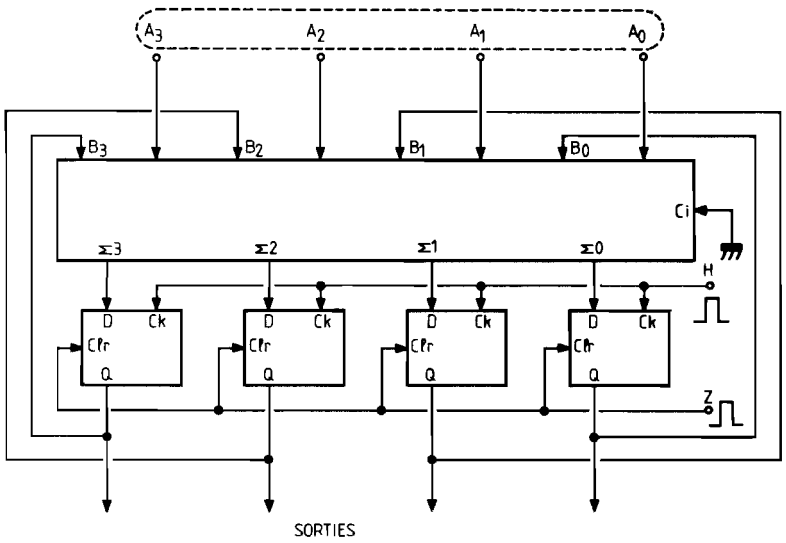

Fig. 80. – Additionneur itératif, pour mots de 4 bits, en nombre supérieur à 2.

Pour permettre de comprendre plus facilement le processus de la soustraction par la méthode « du complément vrai », nous allons imaginer une voiture bizarre, dont le totalisateur kilométrique ne comporte que deux chiffres, les unités et les dizaines.

Il ne peut donc indiquer que les valeurs allant de 00 (inclus) à 99 (inclus). Quand on arrive au kilomètre 99, si l'on roule encore un kilomètre, le compteur-totalisateur revient à 00. On dit qu'il a « recyclé ».

Donc, si le compteur indique une certaine valeur, et que l'on roule exactement cent kilomètres, l'affichage ne change pas. Mais si, le compteur affichant une valeur supérieure à 10, on roule exactement 90 km, l'affichage semble avoir *diminué* de 10.
En effet, si, par exemple, le compteur indique 60, au bout de 39 km, il marquera 99. Le kilomètre suivant (le quarantième) le fera recycler, et les cinquante kilomètres qui suivent provoqueront l'affichage de 50, ce qui est bien inférieur de dix à l'affichage précédent (60).

Autrement dit, comme on ne peut pas « décompter » sur le totalisateur (un parcours de 10 km en marche arrière est fortement déconseillé !), et qu'il est donc impossible de retrancher 10 à ce qu'il affiche, on augmente cet affichage de :

100 – 10 = 90

Or, comme on l'a vu plus haut, ajouter cent ne modifie pas l'affichage. Donc, ajouter « 100 moins 10 » revient à retrancher dix de l'affichage.
Les lecteurs sont d'ailleurs bien habitués à une manœuvre analogue. Quand on repasse à l'« heure d'hiver », renonçant, pour six mois, à « chercher midi à quatorze heures » (quand il est midi *vrai* en été, les montres marquent 14 heures), comme on ne peut généralement pas faire diminuer l'affichage des heures sur une montre numérique, on n'a pas d'autre solution que de l'avancer de :

24 – 1 = 23 heures

ce qui fait diminuer l'affichage de une heure.

LA NOTION DE « FORMAT «

Généralement, quand on veut exprimer un nombre, on sait qu'il est inutile (mais non faux) d'ajouter des zéros à gauche du chiffre le plus significatif du nombre. En effet, 000 013 ou 13 veulent dire la même chose.

Les circuits décodeurs accompagnant les compteurs sont d'ailleurs souvent prévus pour réaliser l'effacement des zéros non significatifs en tête, ce qui complique notablement leur structure.

Quand on veut raisonner sur les soustractions, il faut dire, une fois pour toutes, que les nombres sur lesquels on va opérer sont écrits systématiquement avec un nombre de chiffres. On dit qu'ils sont notés dans un « format » défini.

Pour rappeler ce « format », on doit donc obligatoirement mettre autant de zéros qu'il en faut à gauche du chiffre le plus significatif du nombre.

Si, par exemple, en décimal, on opère en « format à quatre chiffres », le nombre 13 devra s'écrire :

0013

Dans le cas de nombres décimaux en format à quatre chiffres, on peut afficher de 0000 inclus à 9999 inclus. Ajouter 10 000 à un affichage ne change pas ce dernier (dans notre exemple du compteur kilométrique, le format était de deux chiffres).

Donc, dans le cas d'un compteur décimal, affichant ses résultats dans un format à quatre chiffres, si l'on veut, par exemple, retrancher 372 à l'affichage (supposé supérieur à 0372), on peut le faire en ajoutant :

10 000 – 372 = 9 628

On dit que cette valeur (9 628) est le « complément vrai » de 372 dans le format à quatre chiffres décimaux.

LE « COMPLEMENT RESTREINT »

Soustraire 372 de 10 000 n'est pas une opération vraiment difficile, mais on peut la simplifier encore, en remarquant que :

10 000 = 9 999 + 1

Or, quand on retranche n'importe quel nombre (inférieur à 9 999) de 9 999, une circonstance favorable intervient : il n'y a jamais de retenue dans la soustraction.

Donc, si l'on retranche 372 de 9 999, une solution simple consiste à écrire 372 en format à quatre chiffres (soit 0372), puis à remplacer chaque chiffre par son « complément à neuf ».

Ainsi :

le 0 sera remplacé par 9 (0 + 9 = 9)
le 3 sera remplacé par 6 (3 + 6 = 9)
le 7 sera remplacé par 2 (7 + 2 = 9)
le 2 sera remplacé par 7 (2 + 7 = 9)

et l'on obtiendra : 9 627.

Cette différence entre le nombre à soustraire (372) et le plus grand nombre affichable dans le format à quatre chiffres décimaux (9999) se nomme le « complément restreint » de 372.

On passera au « complément vrai » (10 000 – 372) en ajoutant une unité au complément restreint :

9 627 + 1 = 9 628

REVENONS AU SYSTEME BINAIRE

Nous admettons, par exemple, que nous utilisons un format à six chiffres binaires. Il nous permet de noter depuis :

000000 (zéro) à :
111111 (soixante-trois)

Le plus petit nombre qui ne puisse être noté dans ce format est donc 64, qui exigerait un format à sept chiffres binaires.

Soit le nombre binaire :

A = 011110 (trente)

dont nous voulons retrancher le nombre binaire

B = 001101 (treize)

Pour cela, nous allons chercher le « complément vrai » de B, soit le nombre C, qui, ajouté à 13, donne 64 (le plus petit nombre que l'on ne peut noter dans le format à six chiffres binaires).

Là aussi, pour obtenir le complément vrai, C, nous passerons par D, le « complément restreint » de B, c'est-à-dire le nombre que l'on doit ajouter à B pour obtenir *63* (le plus grand nombre que l'on puisse écrire dans le format à six chiffres.

Or, ce complément restreint est d'une facilité idéale à chercher, puisque 63 c'est :

111111

On voit tout de suite que l'addition ci-dessous est juste :

D = 110010
+ B = 001101
= 111111
(soixante-trois)

Il suffit que D ait des 1 partout où B a des 0, et des 0 partout où B a des 1 pour que la somme donne « un partout » (63).

Là aussi, on passe du complément restreint, D (manquant à B pour faire 63) au complément vrai, C (manquant à B pour faire 64) en ajoutant simplement une unité à D :

C = D plus un

D = 110010
donc :

C = 110011

C'est presque fini. Faisons la somme de A et de C :

A011110
+C110011
= (1)0¶0001

A noter que dans cette addition :

– nous avons indiqué des astérisques (*) en haut quand il y a une retenue venant des chiffres de poids inférieurs ;

– nous avons écrit le 1 le plus à gauche de la somme entre parenthèses, car il ne peut être écrit dans le format à six chiffres.

La somme, dans le format à six chiffres est donc :
0 0 0 1 soit 17, ce qui vaut bien 30 moins 13.

RECAPITULONS

Pour soustraire :

B = 001101 (treize) de :

A = 011110 (trente),

nous avons « inversé » tous les chiffres de B (0 remplacé par 1, 1 remplacé par 0), ce qui nous donne D, le « complément restreint » de B :

D = 110010,

nous avons ajouté une unité à D, ce qui nous donne le complément vrai, C :

C = 110011

et nous avons finalement ajouté ce complément vrai à A, en ignorant la retenue de gauche.

Donc, ayant écrit A et B dans le même format, pour soustraire B de A, il faut :

– Inverser tous les chiffres de B, ce qui donne D,
– ajouter 1 à D, ce qui donne C,
– ajouter C à A en ignorant la retenue de gauche.

On aurait aussi bien pu ajouter D à A et ajouter 1 ensuite à la somme.

On voit ici l'importance du « format », car tous les 0 à gauche du chiffre le plus significatif de B sont à transformer en 1.

Notons que, si nous n'avons pas écrit dans le résultat la retenue de gauche, il faut que cette retenue soit apparue dans l'opération pour que celle-ci soit valable.

De même, dans le cas de notre compteur kilométrique à deux chiffres, pour retrancher 10 (en ajoutant 90) il fallait, pour que la méthode soit valable pour réaliser une soustraction (retrancher 10), que le compteur, au départ, marque plus de 10. Cela implique que, au cours de l'addition de 90, il doit « recycler » », ce qui est l'équivalent de la sortie de cette retenue que nous n'avons pas écrite.

(à suivre)

J.-P. CŒHMICHEN