LA RADIO LOGICIELLE
ou
LE TRAITEMENT NUMERIQUE DU SIGNAL
expliqué aux analogiciens par un analogicien.

Robert BERRANGER F5NB

Deuxième partie : le calcul numérique.

Article publié dans Radio-REF de février 2007.

Dans la première partie, nous avons traité de l'échantillonnage. Dans celle-ci nous allons aborder le traitement numérique du signal proprement dit.

Le domaine de la Radio logicielle (software radio) étant très vaste, nous nous sommes limités au cas spécifique d'un récepteur radio avec numérisation du signal antenne. Nous retrouvons le schéma synoptique général sur la fig.1.

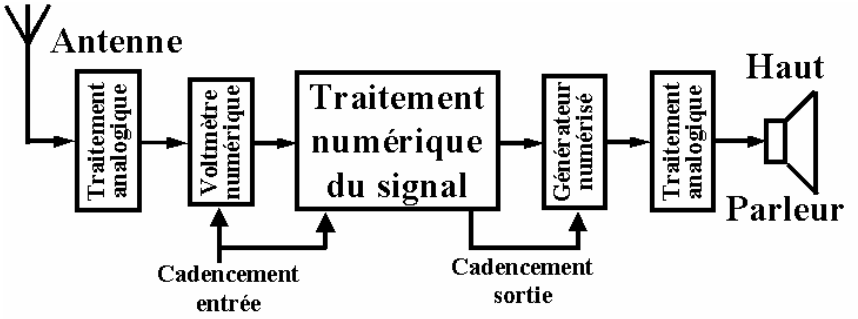

Figure 1

Nous avons vu le traitement analogique d'entrée, filtres, amplis, dither… Nous avons disserté sur le voltmètre numérique composé d'un échantillonneur bloqueur et d'un C.A.N. et vu les contraintes liées à l'échantillonnage.
Nous avons vu, plus brièvement, le « générateur numérisé », composée d'un C.N.A. et le dernier traitement analogique composé d'un filtre et d'un ampli de puissance.
Nous allons nous intéresser maintenant à la partie du milieu, le « traitement numérique du signal ».
On peut résumer simplement le processus en disant qu'on lit des valeurs numériques à la sortie du CAN, à une certaine cadence, puis après différents calculs, on fournit les résultats à un CNA, à une autre cadence.

Avant de décrire les différentes méthodes de calcul nous allons dire un mot du format numérique des données.
Le format naturel à la sortie d'un CAN est appelé « Offset binaire ». Par exemple, pour un CAN 8 bits, la sortie peut prendre une valeur entre 0 et 255[(1)]. Si nous voulons mesurer un signal sinusoïdal qui peut avoir une valeur négative, nous serons obligés de le superposer à une tension continue correspondant à une valeur de 128. Pour les valeurs allant de 128 à 255,

elles seront positives, et négatives pour les valeurs de 0 à 127. Mais pour qu'elles soient vraies, il faudrait leur retrancher 128 (l'offset). Ainsi, 127 deviendrait –1, 126, –2, etc…
On n'effectue pas cette transformation, on se contente d'inverser le bit MSB (l'offset) qui devient le signe (0 = positif, 1 = négatif). Le format obtenu, dit « complément à deux » permet d'effectuer directement les quatre opérations élémentaires (voir annexe A).
Le même format est aussi utilisé pour les nombres fractionnaires signés. Nous avons sur la figure 2, un mot de 8 bits, décliné sous les trois formes.

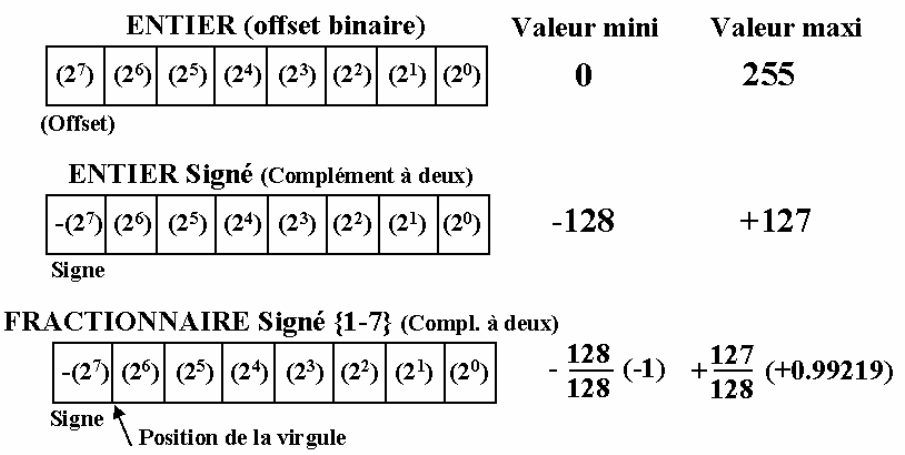

Figure 2

Si nous prenons, par exemple, le mot binaire « 11001101 », il prend la valeur 205 en offset binaire, –51 (256–205) en entier signé, et –0,398 (51/128) en fractionnaire.
Entre un entier et un fractionnaire, il n'y a pas de différence de méthode de calcul pour l'addition et la soustraction. Ce n'est plus vrai pour la division et la multiplication. Ce dernier cas est résumé sur la figure 3.

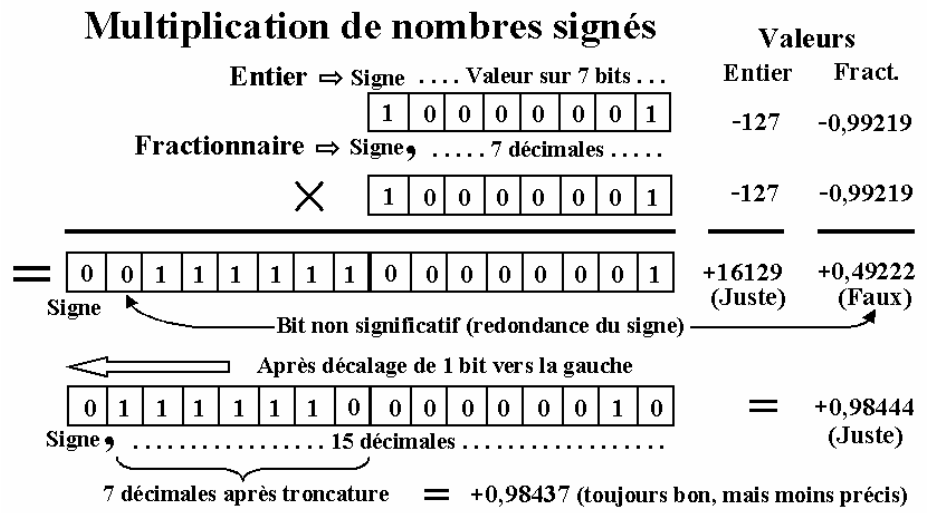

Figure 3

Noter que nous avons un format de nombre fractionnaire 1-7, c'est à dire qu'il y a 7 bits pour les décimales et un bit pour la partie entière, réduite ici au signe. Nous pourrions avoir un autre format, par exemple 3-4, alors les méthodes de calcul seraient différentes. Celles de la figure 3 sont implantées dans les DSP à virgule fixe. Il faut leur déclarer le type de nombre

utilisé, entier ou fractionnaire. Ils peuvent effectuer une multiplication signée en un seul coup d'horloge, à condition de travailler avec un format 1-N. Les divisions s'effectuent en N coups d'horloge, N étant le nombre de bits du diviseur.

L'utilisation de nombres en virgule fixe a des avantages de simplicité et de rapidité pour les calculs. Mais dans certains cas, il entraîne une perte de précision. Exemple (nombres 1-15) : Multiplions 0,9852 (0111111000011011B) par 0,01824 (0000001001010101B). Nous obtenons 0,017949 (0000001001001100B). Maintenant, divisons le résultat par 0,01824, et nous obtenons 0,9849, différent du nombre du début.
Si nous décalons la virgule du nombre binaire 0,01824 de cinq pas vers la droite, nous avons supprimé les cinq zéros de gauche et ajouté cinq bits significatifs à la partie décimale. Ce faisant nous avons multiplié le nombre par 32. Il suffira de s'en souvenir et de diviser par 32 le résultat final. En appliquant la méthode aux calculs ci-dessus, nous obtenons 0,98514 beaucoup plus proche du nombre du début.
Cette façon de déplacer la virgule pour garder la précision maxi est dite « travailler en virgule flottante ». Avec des DSP à virgule fixe, elle est compliquée à mettre en œuvre car le programmeur doit se souvenir des décalages qu'il a effectués. C'est pourquoi l'on utilise des nombres à virgule flottante avec des processeurs spécialisés, sans s'occuper des décalages.
Il existe plusieurs formats de nombres en virgule flottante, mais ils sont tous basés sur le même principe. Ils sont formés du signe plus deux nombres binaires :
- L'exposant qui est un entier signé ($2^{\pm m}$) ou offset binaire ($2^{m\text{-offset}}$).
- La mantisse qui est un nombre fractionnaire du type {1,XXX} (le 1 est implicite)

Le nombre s'obtient en multipliant la valeur de la mantisse par la valeur de l'exposant.
Les nombres à virgule flottante sont normalisés. Les principaux utilisés sont :
- Le single, sur quatre octets, 1 bit pour le signe, 8 bits pour l'exposant et 23 bits pour la mantisse ($10^{\pm 38}$).
- Le real, sur six octets, 1 bit pour le signe, 8 bits pour l'exposant et 39 bits pour la mantisse (10^{-39} à 10^{+38}).
- Le double, sur 8 octets, 1 bit pour le signe, 11 bits pour l'exposant et 52 bits pour la mantisse ($10^{\pm 308}$).

La précision est proportionnelle au nombre de bits de la mantisse.
Les calculs sur les nombres à virgule flottante sont plus compliqués que pour les nombres à virgule fixe.
Pour effectuer une multiplication, on additionne les deux exposants et on multiplie les deux mantisses. Comme il y a débordement, on décale la mantisse (division par deux) et on ajoute 1 à l'exposant. Même principe pour la division, avec soustraction des exposants moins 1, et division des mantisses en ayant décalé le diviseur (divisé par deux), puis décalage du quotient pour supprimer les zéros en tête (non significatifs).
Pour les additions et les soustractions, il faut d'abord normaliser les deux nombres avec le même exposant (décalage d'une mantisse).
Toutes ces opérations se font automatiquement avec les processeurs en virgule flottante, par exemple le co-processeur mathématique du PC. Mais elles ne peuvent se faire en un seul coup d'horloge comme avec les nombres en virgule fixe, et demandent beaucoup plus de « circuiterie » (plus de surface occupée sur la puce).

Nous allons maintenant entrer dans le vif du sujet en décrivant les procédés numériques utilisés pour remplir les principales fonctions radio.
Le synoptique général de la partie numérique d'un récepteur numérique n'est fondamentalement pas différent d'un récepteur analogique. La différence réside dans la manière de réaliser les fonctions qui se fait par calcul au lieu d'associer des composants à

fonction de transfert connue (R-L-C, résonateurs, transistors). Mais comme nous pourrons avoir d'excellentes performances, nous utiliserons une architecture simplifiée : la conversion directe en bande de base complexe I et Q[(2)]. Cette architecture est universelle et permet de décoder toutes les modulations, analogiques comme numériques. Nous l'avons sur la figure 4.

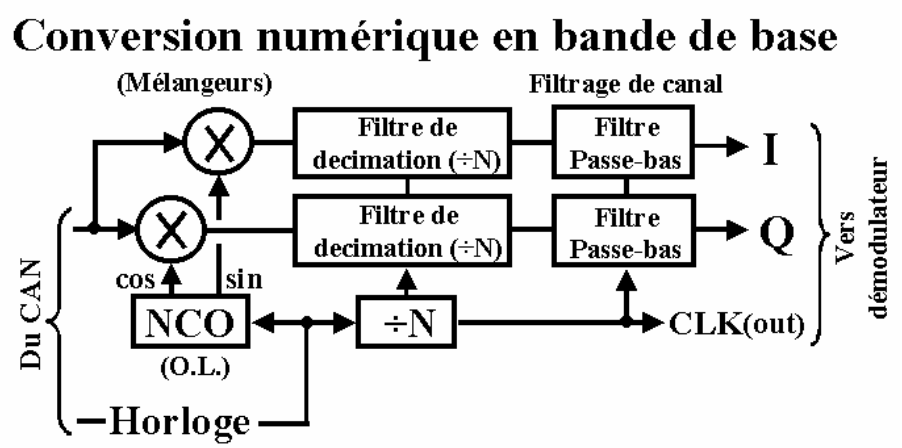

Figure 4

Fonction MELANGEUR

C'est une fonction très facile à réaliser en numérique. Il suffit de multiplier les échantillons d'entrée par les valeurs correspondant à un échantillonnage de l'Oscillateur Local, valeurs fournies par le NCO (Numerical Controlled Oscillator). Si l'on ne veut pas diminuer le rapport S/B du signal d'entrée, il faudra que l'OL soit défini avec au moins deux bits de plus que le signal d'entrée. Par exemple, si le signal d'entrée est sur 16 bits, l'OL sera sur 18 bits. Naturellement, nous aurons tous les problèmes des mélangeurs analogiques. En particulier, nous obtiendrons deux produits à la sortie. Avec une conversion en bande de base (OL=RF) le produit indésirable (OL+RF) est éliminé par le filtre de canal passe-bas. La fréquence image ne pouvant être filtrée, car contiguë, nous utiliserons deux voies en quadrature pour pouvoir l'éliminer en bande de base. Nous verrons en détail le processus dans le chapitre « démodulation ».

Fonctions GAIN et ATTENUATION

Un gain est obtenu en multipliant le signal par une constante supérieure à 1, et une atténuation, par une constante inférieure à 1. La multiplication câblée n'est pratiquement utilisée que pour ajuster le niveau d'une manière fine (ALC et CAG) et pour le filtrage.
Le plus souvent possible, nous nous arrangerons pour avoir à multiplier et diviser par des puissances de deux (multiples de 6 dB), et nous utiliserons des décalages de registres. En effet, un décalage vers la gauche de 1 bit multiplie par deux, un décalage de 2 bits, par quatre, etc… Même principe pour la division en décalant vers la droite.
Les DSP font une différence entre les décalages logiques (comme avec les microprocesseurs) et les décalages arithmétiques. Ceux-ci sont câblés pour un format {1-XXX}. Ils effectuent un décalage seulement sur la mantisse (XXX) en conservant le signe. Nous avons sur la figure 5 un exemple de multiplication par deux sur un nombre utilisant deux registres de 8 bits, et montrant les deux types de décalages.

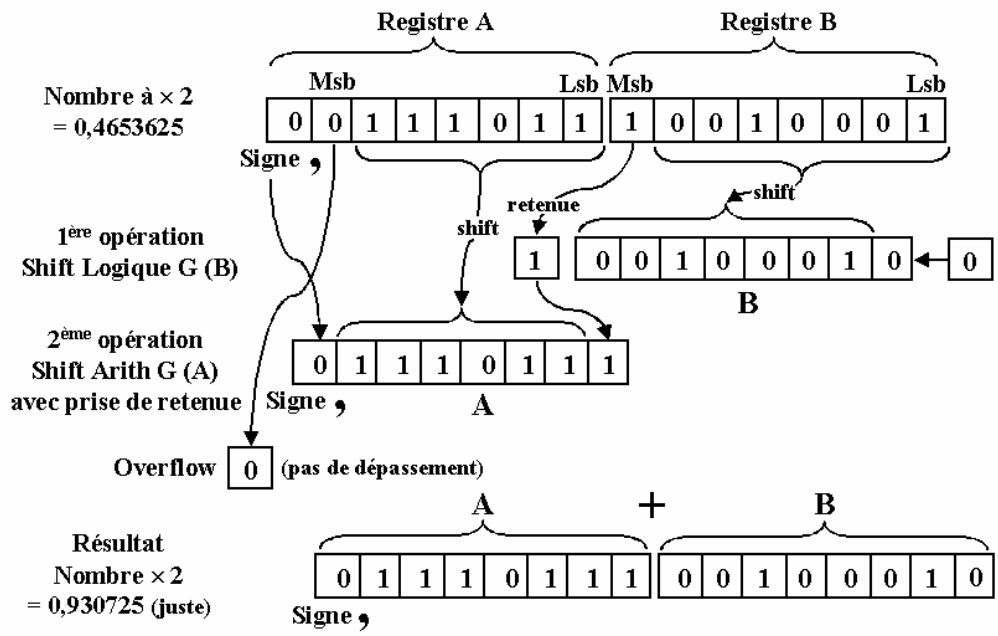

Figure 5

Les décalages sont très utilisés avec les DSP, car très rapides. Pas besoin de charger des registres et de gérer des pointeurs, qui ne sont pas forcément disponibles. Selon les DSP, on peut effectuer jusqu'à 16 bits de décalage sur un double mot en deux coups d'horloge.

Voyons maintenant les débordements. On ne peut pas les laisser passer, car ils occasionnent des erreurs importantes. La solution la plus simple consiste à saturer le registre, ce qui minimise l'erreur (même effet que la saturation d'un ampli analogique). La figure 6 nous donne un aperçu du problème.

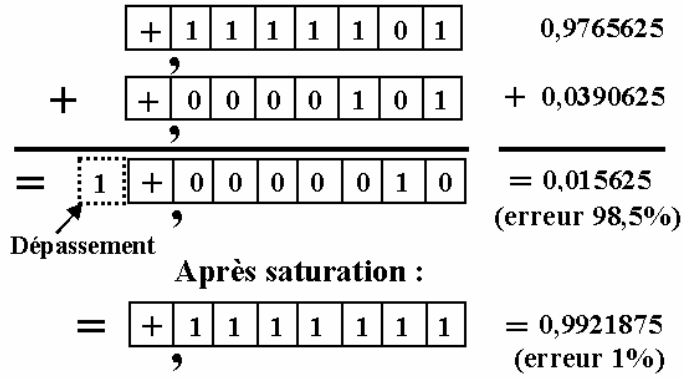

Figure 6

La saturation pour un nombre positif s'obtient en mettant tous les bits de la mantisse à un et la saturation d'un nombre négatif en mettant tous les bits de la mantisse à zéro.
C'est le programmeur qui teste le bit « overflow » en cas de doute et qui gère le dépassement, soit par la saturation, soit en recommençant les calculs avec une modification des paramètres. Dans certains composants spécialisés, comme les DDC (Digital Down Converter), les dépassements entraînent automatiquement la saturation. Les DSP ont généralement une instruction qui force la saturation conformément au signe (si dépassement, saturer le registre).

Fonction OSCILLATEUR (NCO)

C'est une fonction sinusoïdale par rapport au temps, échantillonnée à la cadence de l'horloge. Le théorème de Shannon nous dit que la fréquence ne pourra pas dépasser la moitié de celle de l'horloge, et nous aurons un problème de repliements qu'il faudra filtrer. Avec une conversion directe, la fréquence de l'OL est égale à celle du signal d'entrée, et la gestion des repliements de celui-ci est valable pour l'OL.
Un signal sinusoïdal a son amplitude fonction du sinus d'un angle qui tourne de ω radians par seconde ($\omega = 2\pi F$).
Echantillonner un signal sinusoïdal F1 à la fréquence Fe revient à fournir les valeurs successives du sinus d'un angle φ qui augmente régulièrement de $\Delta\varphi$ en fonction du rapport F1/Fe.
L'angle de l'horloge (Fe) tournant de 2π radians entre deux échantillons (une période), il est facile de calculer $\Delta\varphi$ qui est égal à : $2\pi(F1/Fe)$ radians ou $360(F1/Fe)$ degrés.
Reste à calculer $\sin(\varphi_n)$ avec $\varphi_n = \varphi_{n-1} + \Delta\varphi$.

Oscillateur à fréquence fixe

On s'arrange pour avoir un rapport simple entre F1 et Fe, de façon que l'angle φ reprenne la même valeur (modulo 2π radians) au bout d'un nombre (n) minimum de coups d'horloge. Le calcul du sinus se limitera alors à lire séquentiellement une table de (n) valeurs, ce qui ne demande qu'un coup d'horloge avec les DSP. Par exemple, soit F1 = Fe/3,2. Il suffira de 16 valeurs pour couvrir cinq périodes ($16=3,2\times5$) avec un $\Delta\varphi = 112,5°$ ($16\times112,5 = 5\times360°$).

Cas particulier : Fe/F1 = 4.

Alors $\Delta\varphi = 90°$, et le sinus prend les valeurs remarquables 0, 1, 0 et –1. Ceci simplifie grandement la génération de l'OL et l'opération de mélange. C'est pourquoi on utilise ce rapport pour la conversion d'une FI en bande de base complexe, l'entrée étant affectée alternativement à la voie I et à la voie Q, avec une négation une fois sur deux.

Oscillateur à fréquence variable

On peut considérer un OL variable comme le résultat d'un choix parmi un certain nombre d'OL fixes. Cette méthode devient vite inapplicable dès que le choix dépasse quelques unités. Par exemple, si nous voulions créer un OL variable de 1 à 20 MHz, avec un pas de 1 Hz, il nous faudrait pas moins de vingt millions de tables de sinus.
L'alternative est de calculer le sinus à chaque coup d'horloge. La manière la plus simple consiste à utiliser la fonction câblée d'un processeur de signal, comme le co-processeur math du PC. Mais le calcul demande un grand nombre de coups d'horloge (>100) et n'est possible que si l'horloge du processeur est bien plus rapide que celle du CAN. C'est la solution que nous utiliserons pour la conversion d'une FI basse, ou un traitement BF, à l'aide d'un PC.
Dans un récepteur numérique large bande, l'horloge du processeur de signal est la même que celle du CAN, qui est déjà très élevée (des dizaines de MHz). Il faudra donc trouver une méthode pour calculer un sinus en un seul coup d'horloge.
On pourrait utiliser une table de sinus, adressée par la valeur de l'angle φ. Mais avec un signal défini sur 16 bits, l'OL le sera sur 18 bits. Alors la précision du sinus serait obtenue avec une table de plus de cent mille mots de 18 bits, ce qui prend beaucoup de place sur le silicium.

La solution consiste à utiliser une table réduite aux gros pas et à faire une interpolation pour les petits pas. Nous avons sur la figure 7 un exemple d'interpolation linéaire.

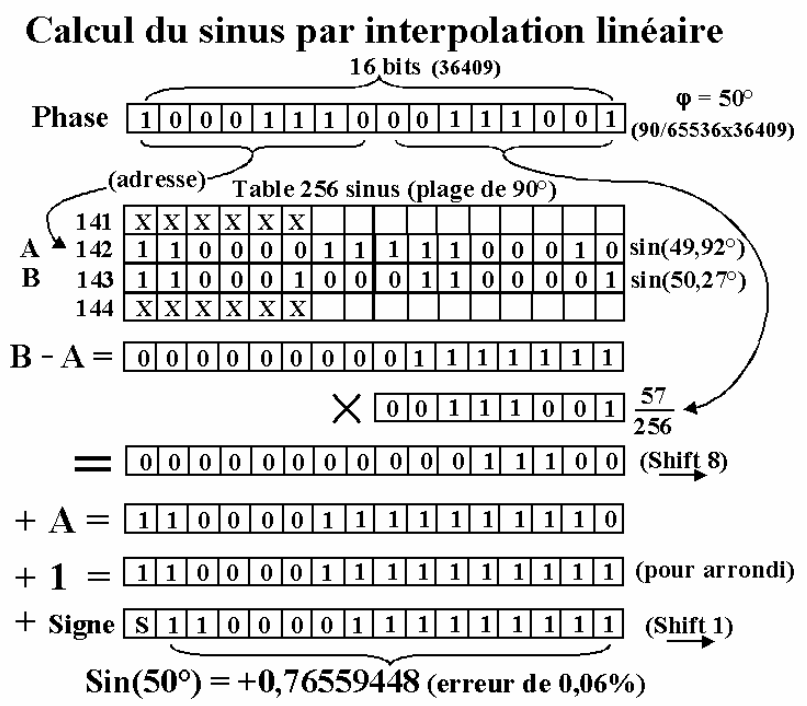

Figure 7

N-B : *L'erreur sur le sinus constitue un bruit de phase. Celui-ci a le même effet que le bruit de phase analogique, en particulier il entraîne du mélange réciproque*[3].

On pourrait également imaginer une interpolation non linéaire qui utiliserait une deuxième table. Par ailleurs, on ne définit le sinus que sur 90°. En effet, il suffit d'inverser le signe pour les angles égaux à φ +180° et d'inverser l'adressage pour les angles φ+90° et φ+270°. Noter que la table est valable aussi pour le cosinus puisque cos(φ) = sin(φ+90°).
Mais, il nous faudra au moins quatre coups d'horloge pour calculer un sinus. Comment alors obtenir un résultat à chaque coup d'horloge ? La solution consistera à faire du traitement parallèle en pipe-line. Le principe est montré sur la figure 8.

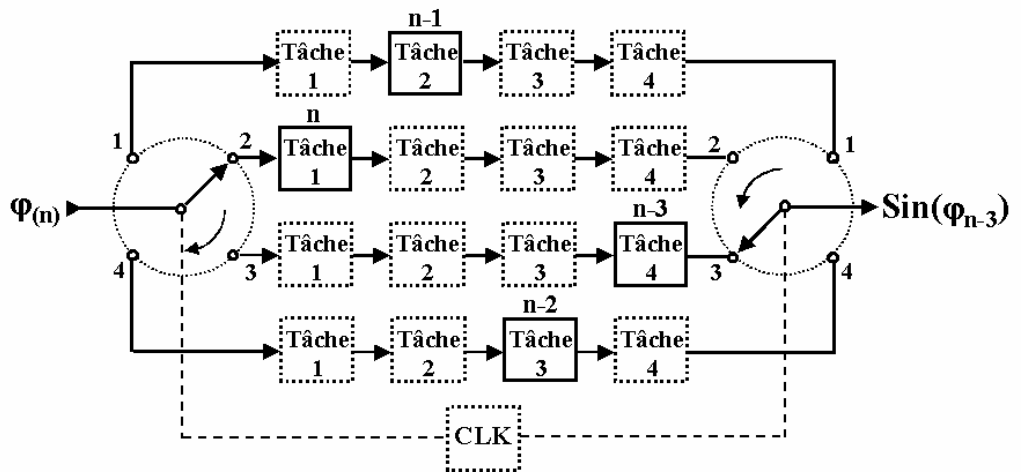

Figure 8

Nous avons ici quatre voies qui effectuent quatre tâches en pipe-line. Celles-ci sont décalées de façon à obtenir un résultat à chaque coup d'horloge. Le choix entre la taille de la table, la taille et l'organisation du pipe-line est fait par le concepteur en fonction de la place prise par le système sur la puce.
Lorsque le NCO est implanté dans un circuit spécialisé, c'est le processeur hôte qui calcule et fournit $\Delta\varphi$ en fonction de la fréquence désirée et de l'horloge.

Si je me suis attardé sur la génération d'un sinus, c'est que la méthode est valable pour n'importe quelle autre fonction mathématique (racine carrée, 1/X, arcTAN, etc…).

Fonction FILTRAGE

Nous verrons la décimation dans le prochain article et nous allons maintenant aborder (en réalité effleurer) le filtrage numérique.

En analogique, nous sommes habitués à la représentation fréquentielle de la réponse d'un filtre. Celle-ci est visible dans le haut de la figure 9.

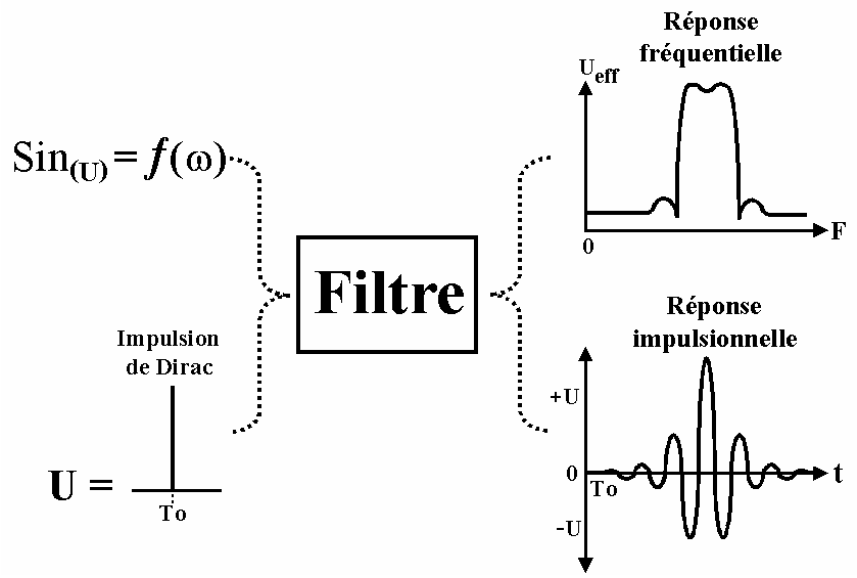

Figure 9

Pour l'obtenir, on fait varier la fréquence d'un signal sinusoïdal en entrée, et on trace la courbe de réponse point par point. Si nous injections à l'entrée un signal ayant un spectre contenant toutes les fréquences, l'enveloppe du spectre de sortie aurait exactement l'allure de la réponse fréquentielle. Un tel signal d'entrée, avec un spectre infini, existe, c'est l'impulsion de Dirac, impulsion de durée très brève et de très grande amplitude.
Maintenant, au lieu de regarder le spectre du signal de sortie, nous pouvons regarder la variation de la tension de sortie par rapport au temps, à partir de l'instant d'application de l'impulsion de Dirac. Nous obtenons la réponse impulsionnelle[(4)], comme montrée sur le bas de la figure 9 (vue d'artiste).

La réponse fréquentielle est la transformée de Fourier de la réponse impulsionnelle, et celle-ci est la transformée inverse de Fourier de la réponse fréquentielle.
Noter qu'à une réponse fréquentielle donnée, ne correspond qu'une seule réponse impulsionnelle, et inversement.

Lorsque nous échantillonnons un signal analogique, chaque échantillon correspond à une impulsion de Dirac d'amplitude et de polarité équivalentes. Si nous appliquons ces échantillons à un filtre, chacun d'eux occasionnera une réponse impulsionnelle. La sortie du filtre sera la superposition (la somme) de toutes les réponses impulsionnelles des échantillons successifs[5]. Ceci est montré sur la figure 10.

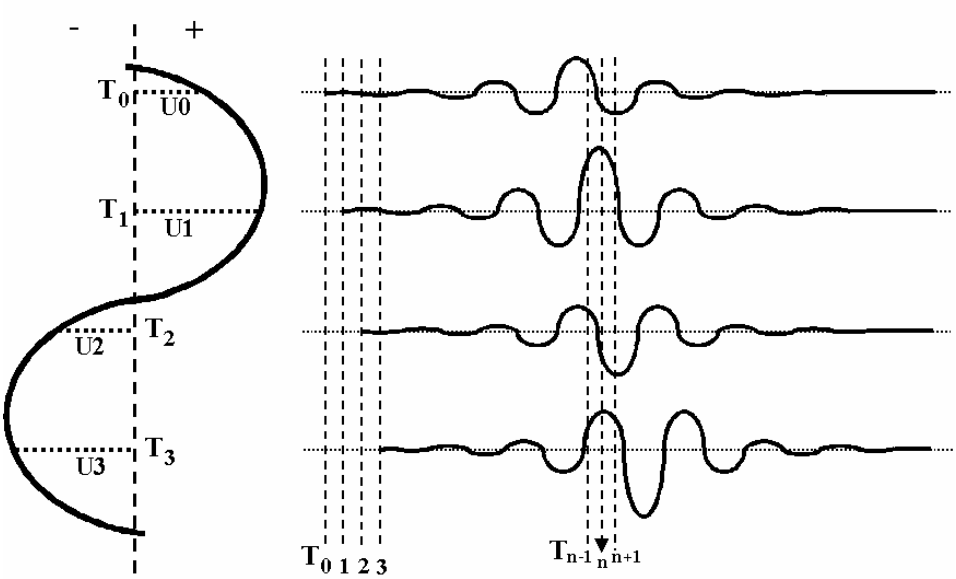

Figure 10

Ainsi, la tension de sortie à l'instant T_n sera la somme de toutes les réponses impulsionnelles appliquées aux échantillons de T_0 à T_n.

Les filtres I.I.R. (Infinite Impulse Response)

Leur synthèse est la mise en pratique du principe exposé sur la figure 10, en y incorporant une récursivité (retour à l'entrée d'une partie de la tension de sortie).
L'avantage de la récursivité est de rendre infinie une réponse impulsionnelle synthétisée par un nombre de coefficients limité. On peut ainsi synthétiser tous les filtres analogiques connus (Butterword, Chebychev, etc…). Par contre, les calculs sont complexes et le temps de propagation dans le filtre dépend de la fréquence d'entrée du signal (comme dans un filtre analogique). C'est pourquoi, on emploie peu les IIR, et on leur préfère les FIR qui, n'ayant pas de récursivité, n'ont pas ces inconvénients (ils en ont d'autres). Nous allons nous y attarder.

Les filtres F.I.R. (Finite Impulse Response)

La réponse impulsionnelle est dite « finie » car on prend un nombre limité de coefficients, et on les utilise en boucle ouverte. Alors, les calculs se résument à une suite de « multiplication-accumulations » (MAC).

La méthode est explicitée sur la figure 11 qui est une façon de réaliser strictement la fonction de la figure 10.

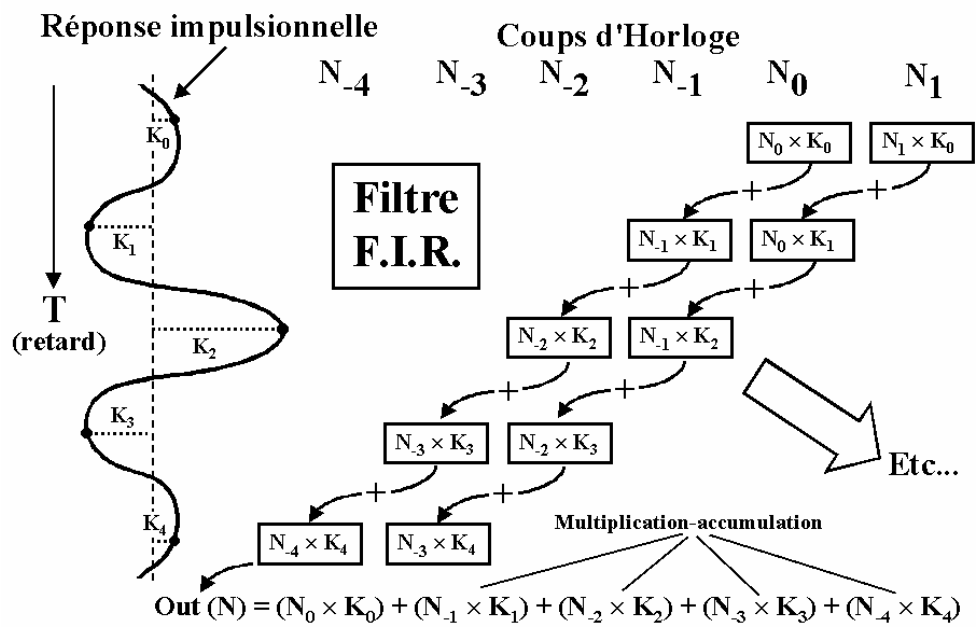

Figure 11

Ici, nous n'avons que cinq coefficients pour faciliter la compréhension. On imagine aisément que plus il y en aura, et plus le filtre se rapprochera de l'idéal, mais aussi, plus le retard sera important. Avec un FIR, le retard est indépendant de la fréquence d'entrée (boucle ouverte) et égal, en nombre de coups d'horloge, à la moitié du nombre de coefficients. Ceci permet la synchronisation exacte (mise en phase) de fonctions parallèles.

Avec un DSP, la mise en œuvre d'un filtre FIR est facilitée par l'opération câblée « MAC » qui permet de faire une multiplication-accumulation en un seul coup d'horloge. Les coefficients du filtre sont contenus dans une table stockée dans la mémoire programme (ROM), et les données sont stockées dans un buffer tournant (bouclé sur lui-même) de même longueur situé dans la mémoire données (RAM), ceci pour pouvoir lire les deux mémoires simultanément. L'opération de filtrage consiste à entrer l'échantillon courant dans le buffer des données, en remplacement du plus ancien, puis d'additionner successivement le résultat de chaque valeur du buffer multipliée par le coefficient de même rang relatif. Et on avance le pointeur du buffer tournant d'un pas pour l'échantillon suivant. Le traitement d'un échantillon demande alors un nombre de coups d'horloge égal au nombre de coefficients, plus quelques coups pour la préparation des pointeurs.

Le nombre et les valeurs des coefficients sont calculés par des programmes spécialisés, selon le filtre désiré. Les coefficients sont normalisés de façon que le filtre ait un gain unité pour sa réponse maxi (perte d'insertion = 0 dB).

Mais la simplicité du FIR a un prix. Pour un même nombre de coefficients, les performances d'un FIR sont nettement moins bonnes que celles d'un IIR.
En effet, une réponse impulsionnelle finie peut être considérée comme une réponse impulsionnelle infinie multipliée par une fonction rectangle. La réponse fréquentielle du filtre est alors l'addition de la transformée de Fourrier de la réponse impulsionnelle infinie (filtre idéal) avec celle du rectangle qui est une fonction sin(x)/x. Ceci est montré sur la figure 12 pour un filtre passe-bas.

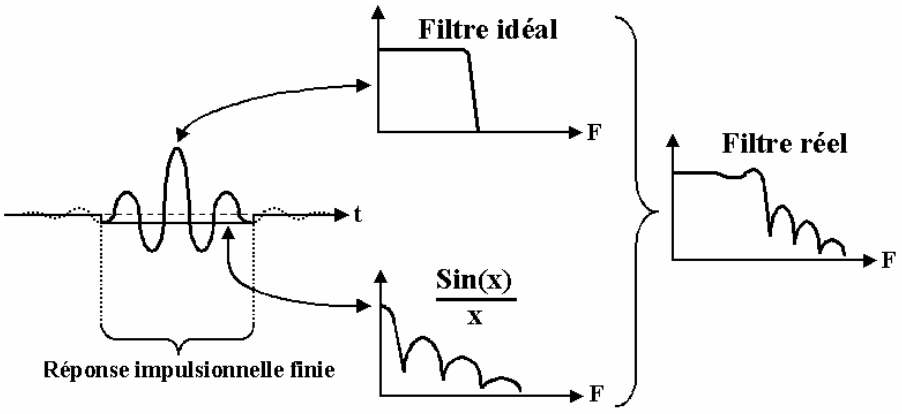

Figure 12

Plus nous aurons de coefficients, et moins la fonction rectangle aura d'importance. Mais on peut chercher à améliorer les choses sans augmenter le nombre de coefficients. La solution consistera à remplacer la fonction rectangle par une autre fonction ayant une transformée de Fourrier plus adéquate. Il n'en manque pas, et beaucoup de mathématiciens ont inventé la leur.
Toutes ces fonctions ont comme particularités de démarrer à zéro, de passer par un maximum au maximum de la réponse impulsionnelle, et de terminer à zéro. Ces fonctions sont appelées « fenêtres ». Chaque coefficient de la réponse impulsionnelle est multiplié par le coefficient correspondant de la fenêtre.
La plus simple est une fonction triangulaire symétrique. Le résultat ne fait que lisser les ondulations d'une fenêtre rectangulaire. Il existe des fenêtres de Hanning, de Hamming, de Kaiser, etc… Elles ont plus ou moins la forme d'un « omega ». Elles font chacune un compromis entre la remontée des lobes et la pente de la coupure, sachant que moins il y a de remontées, et plus la pente est douce.
Si l'on tolère une ondulation dans la bande, comme en audio par exemple, on peut modifier la réponse impulsionnelle finie pour obtenir une réjection dite « equiripple » (à ondulation régulière), c'est à dire que tous les lobes des remontées sont au même niveau, d'autant plus bas que l'ondulation dans la bande est importante. C'est la méthode qui permet d'obtenir les flancs les plus raides (pour un même nombre de coefficients), mais elle n'est pas toujours applicable. Nous avons sur la figure 13 les réponses obtenues en fonction de différentes fenêtres.

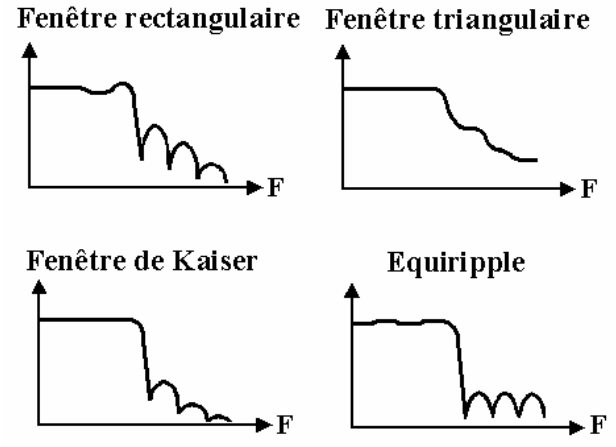

Figure 13

Il y a une différence fondamentale entre les FIR et les filtres analogiques. Pour ces derniers, à nombre de composants donné, la pente est une constante en dB par octave. Pour un FIR, à nombre de coefficients donné, la pente est une constante en absolu. Ceci fait qu'il faut beaucoup plus de coefficients pour un filtre passe-haut (Fc basse) par rapport à un filtre passe-bas (Fc élevée), si l'on veut la même pente en dB/octave. C'est pourquoi aussi, on s'arrange pour que la fréquence d'horloge soit la plus basse possible compte tenu du critère de Shannon et du filtre anti-repliement.
Par exemple, pour un filtre passe-bas de 3 kHz, on utilisera un FIR avec une fréquence d'horloge de l'ordre de 8 kHz. Si l'on filtrait à la fréquence de l'horloge d'entrée (des dizaines de MHz pour un poste HF), il nous faudrait des milliers de coefficients à calculer sur un coup d'horloge. Impensable, et c'est pourquoi on a recours à la décimation que nous verrons dans le prochain article.

F.I.R. particuliers

Filtre passe-bas du 1er ordre

C'est l'équivalent d'un R-C. C'est un FIR dont les n coefficients sont égaux à 1/n. Alors le calcul se simplifie. On utilise un buffer tournant dont on remplace un élément à chaque coup d'horloge. On additionne le buffer et on divise le résultat par sa longueur. Celle-ci est prise égale à une puissance de 2, ce qui permet une division par simple décalage de bits. En fait, on n'additionne pas tout le buffer à chaque coup d'horloge, on soustrait d'un accumulateur une valeur qu'on remplace par l'échantillon courant que l'on ajoute à l'accu. On avance le pointeur et on copie l'accu avec un shift de la puissance de deux correspondant à la longueur du buffer.
Si $F_{entrée} < (Clk/n)$, nous avons un filtre passe-bas du premier ordre. Si $F_e >> (Clk/n)$, alors la sortie représente la moyenne arithmétique. Utilisé dans les boucles d'asservissement (ALC, CAG, CAF, etc…) et comme intégrateur.

Filtre transformée de Hilbert

C'est un FIR passe-bande de largeur maximale. A l'intérieur de celle-ci, le déphasage sortie/entrée est de 90°. Très employé pour le traitement des données I et Q d'une transposition en bande de base complexe. Le retard du filtre est égal à la moitié de son nombre de coefficients.

Ligne à retard (L.A.R.)

C'est un filtre passe-tout dont on n'utilise que son retard. Il suffit d'un buffer tournant. On lit une valeur qu'on remplace par la valeur de l'échantillon courant, puis on avance le pointeur d'un pas. On obtient ainsi un retard égal à la taille du buffer. Très utilisée pour compenser le retard d'un FIR (voies parallèles).

Ceci terminera la deuxième partie. Dans la prochaine, nous verrons la décimation, la démodulation, la CAG, le post-filtrage, etc…

F5NB.

Annexe A.

Le format dit « complément à deux ».

On pourrait travailler avec des nombres au format standard des nombres décimaux. Alors, la conversion des nombres « offset binaire » en nombres signés serait compliquée. De même les opérations élémentaires, car il faudrait appliquer la règle des signes. Le format dit « complément à deux » simplifie tout cela. Nous avons vu que la conversion d'un nombre offset binaire se faisait simplement par une inversion du bit LSB qui devenait le signe. Nous avons sur la figure 14 le principe général et la méthode pour passer d'un nombre négatif standard (signe plus valeur absolue) à un nombre « complément à deux ».

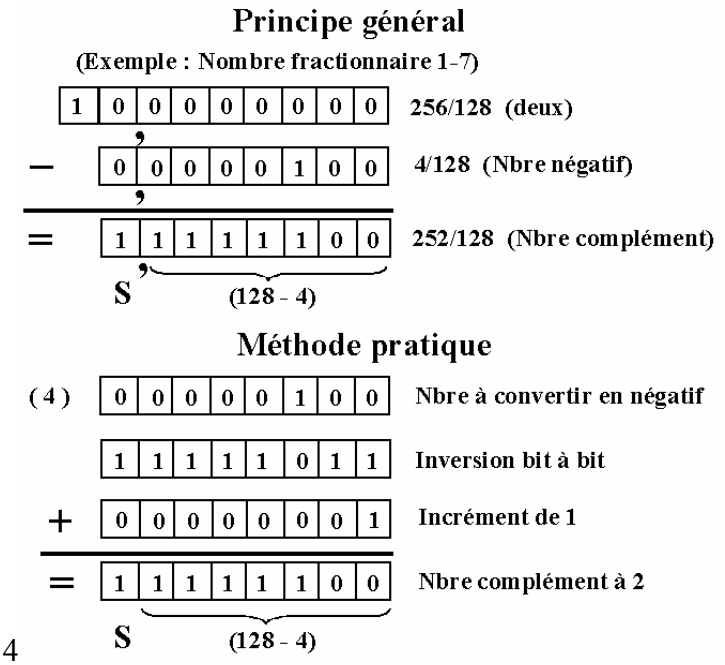

Figure 14

L'utilisation d'un nombre fractionnaire montre bien le principe de conversion. La partie entière du résultat est devenue le signe du nombre (1 = négatif). La méthode pratique est valable, que le mot représente un nombre entier ou fractionnaire. Elle est identique pour repasser du nombre complément à 2 au nombre absolu.
Sur la figure 15, nous avons une addition de nombres « complément à deux », montrant que celle-ci s'effectue sans s'occuper de la règle des signes, donc utilise le même additionneur que pour les nombres non signés.

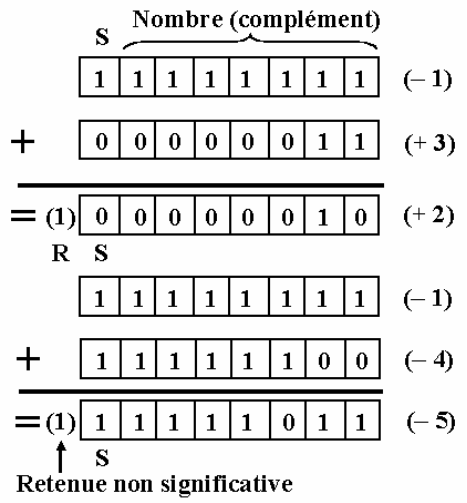

Figure 15

Noter qu'avec des nombres signés, la retenue n'est plus utilisée pour contrôler un dépassement. On examine les signes des données. S'ils sont différents, pas de dépassement possible. Sils sont identiques, le résultat doit être du même signe.

Notes.

(1) Pour un CAN à tension d'alimentation unique, comme les CAN rapides type « Radio ».
(2) La fréquence centrale du signal reçu est convertie à la fréquence zéro. Nous verrons cela en détail dans la troisième partie.
(3) Voir le précédent article.
(4) Réponse temporelle à l'impulsion de Dirac.
(5) C'est le principe de superposition applicable aux systèmes linéaires.