



**HAL**  
open science

# ANALYSE DE DISPOSITIFS DIGITAUX

Bernard Couty

► **To cite this version:**

| Bernard Couty. ANALYSE DE DISPOSITIFS DIGITAUX. 2018. hal-01731093

**HAL Id: hal-01731093**

**<https://hal.science/hal-01731093v1>**

Preprint submitted on 13 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# ANALYSE DE DISPOSITIFS DIGITAUX

Au cœur de la fabrication

**Résumé :** Ce troisième essai prend pour prétexte la transformée de Fourier rapide pour explorer les dispositifs techniques capables de l'effectuer : échantillonneur, convertisseur, analyseur. L'attention de l'ergologue est attirée sur le fait que des machines complexes sont en réalité composées de dispositifs unitaires, les portes logiques, qui assurent l'essentiel des opérations arithmétiques et logiques. Cette observation conduit à s'interroger sur les rapports de la modélisation algébrique avec la production industrielle des dispositifs.

**Mots-clés :** additionneur, amplificateur opérationnel, analogique, automatique, bascule DQ, convertisseur analogique/numérique, dispositif, échantillonnage, engin, épitactique, filtrage, filtre, fréquence d'échantillonnage, harmonique, horloge, impulsion de Dirac, machine, matériau, multiplieur, numérique, peigne de Dirac, polytrophe, polytropie, porte logique, quartz, soustracteur, spectre de fréquence, spectre de module, tâche, TFR, théorétique, transformation de Fourier, trigger de Schmitt

**Auteur :** Bernard Couty, MCF 71<sup>e</sup> section (retraité)

Chercheur associé au LiRIS, EA 7481, Université de Rennes 2

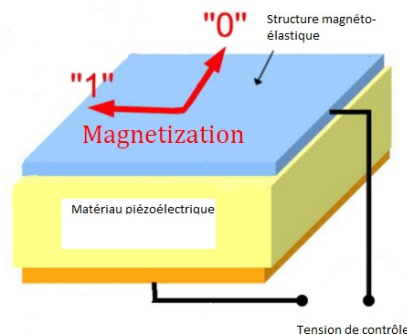
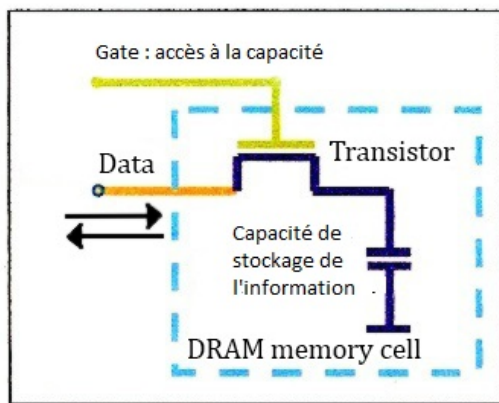
Troisième causerie.

## L'INDUSTRIE NUMÉRIQUE

*Où nous allons mettre les mains dans le cambouis.*

Nous allons aborder le passage des signaux analogiques aux signaux numériques. La causerie précédente portait essentiellement sur l'image, mais il y a bien d'autres types de signaux que ceux tirés du rayonnement photonique.

Auparavant, signalons un fait technologique récent : la conception d'un type nouveau de « mémoire » par un laboratoire Russe ([https://mipt.ru/english/news/researchers\\_create\\_magnetic\\_ram](https://mipt.ru/english/news/researchers_create_magnetic_ram)) apportant un progrès notamment dans la consommation d'énergie des ordinateurs. À gauche, une mémoire « classique », à droite la mémoire magnéto-élastique.



Cela montre à l'évidence que rien n'est virtuel, mais au contraire que tout est *fabriqué*. Nous avons ici des dispositifs dont la tâche est le stockage d'unités de charge (tensions notées *théoriquement* 0 ou 1) accessibles par un processeur lorsque le programme le lui commande.

Je voudrais également attirer l'attention sur les « représentations de l'art » évoquées par Bruneau et Balut dans *Artistique et Archéologie* (Paris, Presses de l'Université de Paris-Sorbonne, 1997, p 122). Cela pose tout le problème de la « production scientifiquement calculée », des modélisations scientifiques de *l'ars fabricandi*. Je pense qu'il faut distinguer, comme le font les auteurs, une visée descriptive (ostension par la description et le schéma : *Darstellung* en Allemand) à laquelle je référerai désormais par le vocable de *théorique*, d'une visée impérative, sorte de « recette » de fabrication, ou calcul de production (*Vorstellung*), à qui convient, ce me semble, le vocable d'*épitactique*. Autrement dit, on peut décrire et expliquer par le langage et les algèbres (théorique) et prescrire également par le langage et les algèbres (épitactique). Mais l'outil, la *fabrication* abstraite, sont autonomes par rapport aux algèbres.

Venons-en au propos de cette causerie.

# 1. De l'analogique au numérique et ce qui s'ensuit.

## 1.a. Le très discret $\Delta t$ .

Dans la première causerie, j'ai fait la distinction entre un phénomène -par exemple un rayonnement naturel- et un signal : « un signal, pour continu qu'il soit, est *souvent* en réalité, par un procédé à la fois logique et industriel, discrétisé, de telle manière que l'on puisse comparer (donc contraster ?) deux valeurs successives :  $y$ ,  $y+\Delta y$ ,  $x$ ,  $x+\Delta x$ . » Je pense qu'il faut garder cela à l'esprit. Pour produire un courant électrique alternatif, on fait tourner un rotor (induit) dans un champ magnétique (inducteur) : le mouvement est strictement continu tant qu'un moteur le permet. Par conséquent, le champ électrique induit variera continûment et un galvanomètre -analogique par construction- pourra mesurer exactement le courant produit. Mais dès que l'on veut *étudier* le phénomène, on est amené à distinguer dans la rotation, entre deux moments  $T$  et  $T+\Delta T$  un écart angulaire concomitant  $\Delta\theta$  et tout ce qui s'ensuit.  $\Delta$  témoigne en quelque sorte de notre capacité humaine à discrétiser en même temps qu'à comparer une quantité moindre à une quantité supérieure. Cependant on ne peut que constater qu'entre deux quantités  $Q_1$  et  $Q_2$  il existe une infinité de valeurs. Par conséquent  $\Delta$  témoigne aussi de la récursivité : dans l'intervalle choisi, on opère une nouvelle découpe discrète, et l'opération pourrait se réitérer un nombre infini de fois. Nous y mettons cependant une limite. Celle-ci peut être arbitraire, donc objet de convention, ou dépendre de nos moyens techniques de mesurer.

Cette mesure, en ce qui concerne le temps, est évidemment celle opérée par les horloges. Le temps ; c'est un paramètre de mesure, certes, mais il peut être éliminé des équations, comme le montre le calcul de la vitesse de la lumière par Fizeau (1849). Il n'empêche: Nicolas Tesla nous disait : « *if you want to find the secrets of the universe, think in terms of energy, frequency and vibration* ». Et  $t$  est un paramètre fondamental en matière de numérisation. Et, bien sûr, nous calculons les fréquences en mégahertz, voire térahertz, autant de rotations par seconde du vecteur de Fresnel<sup>1</sup>. On retrouve ici la rotation, celle de la Terre sur elle-même en 86400 secondes, celle d'un induit dans le champ magnétique inducteur etc... Autrement dit le nombre de rotations du vecteur par unité de mesure temporelle. Ne nous laissons pas bernier par l'emploi du système de numération décimal : 1 milliseconde (0,001S) vaut en réalité 0,06 tierce (0,001/60<sup>-1</sup>seconde), une microseconde 0,000006 tierce et ainsi de suite. En dépit des apparences, le temps demeure cyclique. Une fréquence<sup>2</sup> de 1 MHz correspond à un passage de  $\{A^+ \rightarrow 0 \rightarrow A^- \rightarrow 0 \rightarrow A^+\}$  une fois par microseconde, un million de fois par seconde.

---

<sup>1</sup> Mathématiquement parlant, bien entendu.

<sup>2</sup> La fréquence est le rapport de l'unité de mesure à la période, et la période correspond au temps nécessaire au passage de deux maxima du vecteur de Fresnel (ou d'une onde !).

## 1.b. L'horloge.

Où veux-je en venir ? Simplement au fait que pour choisir un  $\Delta t$ , il faut disposer d'un dispositif technique adéquat pour mesurer le temps. Si je veux mesurer la valeur instantanée d'un signal sinusoïdal en employant la formule très classique  $A_m \sin(\omega t)$ , il me faut mesurer  $t$ , donc choisir un  $\Delta t$  de  $1 \mu\text{S}$ ,  $1 \text{mS}$  etc... L'horloge est donc un dispositif indispensable, et elle se trouve non seulement dans tout ordinateur, mais aussi dans tout dispositif intervenant dans une machine dont la fin est de numériser.

Nous appellerons « horloge » tout dispositif capable de fournir des impulsions (électriques ou mécaniques) distantes d'un  $\Delta t$  constant connu. En électronique numérique, le système le plus simple

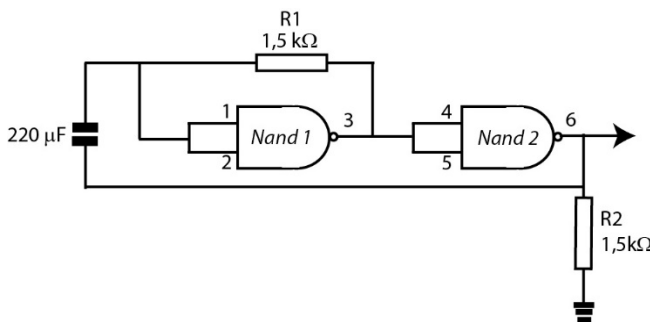


Image 34 : trigger de Schmitt

est le *trigger de Schmitt* comportant deux inverseurs (Nand1 et Nand2) connectés de telle manière que la sortie de la Nand2 (borne 6) réagisse sur l'entrée de la Nand1 via la capacité  $220 \mu\text{F}$  et la résistance de  $1,5 \text{k}\Omega$  qui constituent une « base de temps » que l'on peut modifier en changeant les valeurs de la capacité et de R1.

Cependant ce n'est pas entièrement satisfaisant, un tel circuit finit par dériver, aussi peut-on le stabiliser par un résonateur à quartz, la géométrie du cristal lui permettant de résonner selon une fréquence propre et stable. On trouve un quartz dans le circuit d'horloge près du microprocesseur sur la carte-mère d'un ordinateur, par exemple sur le petit Arduino Uno (voir image 35), véritable micro-ordinateur utile à qui veut s'initier à l'électronique numérique.

Je reviendrai plus tard à la quincaillerie (hardware). Pour le moment, examinons le principe théorique de la numérisation.

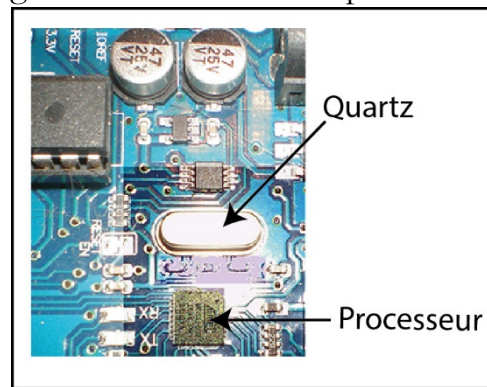


Image 35 : L'horloge et le processeur (Arduino Uno)

## 1.c. La fonction d'échantillonnage.

Pour cela, fabriquons (par exemple avec Excel) un signal sinusoïdal correspondant à une fréquence  $F$  de  $1 \text{Hz}$ , donc une rotation du vecteur de Fresnel par seconde. Comme il est numériquement impossible de produire un signal continu, si près de zéro que soit  $\Delta t$ , nous devons simuler l'horloge, c'est-à-dire, partant de 0, choisir un accroissement  $\Delta t$  convenable. Supposons que l'on prenne  $1024$  tops d'horloge =  $1 \text{S}$ , l'accroissement sera  $\Delta t = 1/1024 = 0,00097656 \text{S}$ , soit  $0,97656 \text{mS}$  ou  $976,56 \mu\text{S}$ . Ainsi le temps va-t-il courir selon  $T_i := T_{i-1} + \Delta t$ , partant de  $i=0$  avec  $i$  prenant les

valeurs [1..1024]. Donnons-nous (pour ne pas obtenir un affichage trop important en hauteur) une amplitude  $A_m$  de 1, calculons  $\omega=2\pi F$ , appliquons l'équation classique d'une sinusoïde  $A_i = A_m \cdot \sin(\omega T_i)$ , et nous obtenons une série de 1024 valeurs. Reportons-les sur un graphique, et nous obtenons la courbe de l'image 36. Mais nous sommes bien déjà dans le numérique, puisque nous avons discrétisé avec  $\Delta t$ . Maintenant, il s'agit de mettre en route une autre horloge, ou d'utiliser la même mais d'une manière particulière<sup>3</sup>. Il se trouve que les maths nous offrent une solution

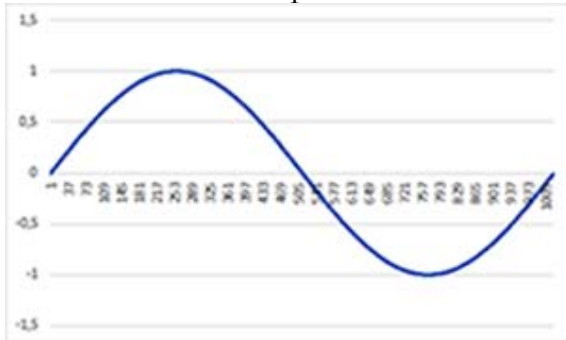
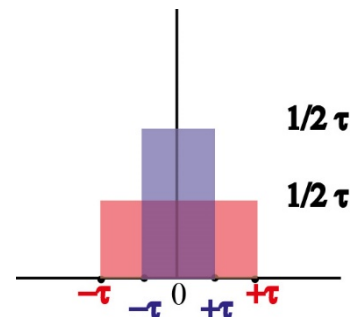


Image 36 : sinusoïde 1 Hz 1 seconde:= 1024 points

intéressante. Il existe une fonction  $\delta$  appelée «impulsion de Dirac», assez bizarre. On en trouve une très claire explication dans une [vidéo](#)<sup>4</sup>. Supposons une fonction  $d_\tau(t)$  définie comme  $d_\tau(t) = \begin{cases} \frac{1}{2\tau} & \text{si } -\tau < t < +\tau \\ 0 & \text{sinon} \end{cases}$ . Elle vaut 0 jusqu'à  $-\tau$  et se trouve  $>0$  entre  $-\tau$  et  $+\tau$ . L'aire du rectangle rouge vaut  $\int_{-\tau}^{+\tau} d_\tau(t) dt$  soit 1<sup>5</sup>.

Si l'on resserre l'intervalle  $[-\tau, +\tau]$ , le rectangle aura une base plus étroite et une hauteur plus « haute », mais l'aire du rectangle bleu sera encore égale à 1. Il va de soi que  $\lim_{\tau \rightarrow 0} d_\tau(t)$  donnera encore 1, ce qui correspond exactement à la fonction de Dirac,  $\delta(t)$ . On a une impulsion que l'on représente par une flèche, mais dont l'aire vaut 1 en  $t=0$ , 0 partout ailleurs. Et cette fonction peut se translater sur l'axe du temps, sans que la valeur de son aire varie :  $\int_{-\infty}^{+\infty} \delta(t_i - t_0) dt = 1$ . On comprend que cela peut fournir un train d'impulsions échelonnées dans le temps. Mais comment la fonction de Dirac peut-elle servir à échantillonner ? Facile : lorsque l'on multiplie une quantité  $q$  par zéro, on obtient 0, et si on la multiplie par 1, on obtient  $q$ . Si j'ai une fonction  $f(t)$  et une fonction  $\delta(t)$  de Dirac,  $\int_{-\infty}^{+\infty} f(t) * \delta(t - t_0) dt$ ,



l'aire, vaudra 0 quand  $\int_{-\infty}^{+\infty} \delta(t - t_0) dt$  vaudra zéro et  $f(t_0)$  quand l'aire vaudra 1. Concrètement, on peut définir une série périodique d'impulsions de Dirac. Si je définis une horloge dont  $\Delta t = 64$  (j'ai 1024 tops de l'horloge principale, de la sinusoïde, et 64 est un diviseur entier de 1024 ( $1024 = 2^{10}$ )), pour  $i$  courant de 0 à 1024, chaque fois que  $i \bmod 64$  vaut 0 (c.à.d si  $i$  est

<sup>3</sup> On pourrait utiliser l'horloge du processeur, également, pour le mien 1,50 GHz, mais on obtiendrait à l'affichage une image un peu raplapla sans rien changer au problème !

<sup>4</sup> <<https://fr.khanacademy.org/math/differential-equations/laplace-transform/properties-of-laplace-transform/v/dirac-delta-function>>

<sup>5</sup> L'aire sous un graphe s'exprime par une fonction d'intégration notée  $\int$ .

exactement divisible par 64), j'indique 1, 0 dans les autres cas. On réalise ainsi un *peigne de Dirac* qui permettra de prélever 16 valeurs instantanées de la fonction sinusoïdale.

Par suite, à chaque impulsion  $i$  de l'horloge, on multiplie la valeur instantanée de la sinusoïde par 0 ou 1 selon la fréquence du peigne, et l'on prélève cette valeur dans le cas 1. C'est ce que

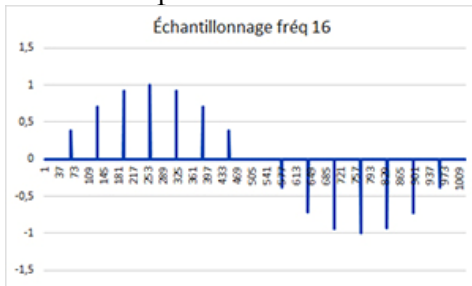


Image 38 : histogrammes d'échantillonnage

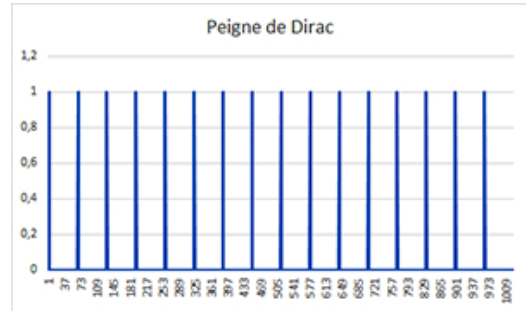


Image 37 : peigne de Dirac

montrent les histogrammes de l'image 38. Ces histogrammes représentent les valeurs prélevées sur la sinusoïde exactement à la fréquence du peigne de Dirac. L'image 39 ne fait que récapituler l'opération.

Il est important de noter que pour obtenir un échantillonnage permettant de restituer le signal d'origine, la fréquence d'échantillonnage doit répondre à la règle énoncée par Shannon :  $f_{\text{échantillonnage}} \geq 2 * f_{\text{max}}$ . En pratique, si l'on veut ne pas rater les « pics » d'amplitude, il vaut mieux que la fréquence d'échantillonnage soit  $\geq 2,56 * F_{\text{max}}$ . On choisit la fréquence maximale dans le cas de signaux composites, par exemple une onde modulée en amplitude par une onde modulante, ou une onde modulée en fréquence etc... Les signaux peuvent être bien moins réguliers qu'une simple sinusoïde. Dans notre exemple, nous avons une fréquence d'échantillonnage de 16 pour une fréquence de 1. L'image 39 montre bien que les histogrammes suivent fidèlement le graphe de la fonction. Avec une fréquence d'échantillonnage  $< 2 f_{\text{max}}$ , il y aurait beaucoup trop de perte d'information et l'on ne pourrait pas reconstituer le signal. Néanmoins, cela va de soi, tout échantillonnage est une perte d'information. Une fois de plus !

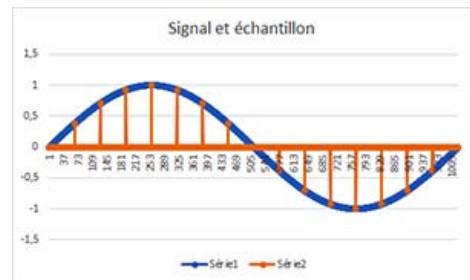


Image 39 : sinusoïde et échantillons

Nb : il n'y a pas que la fonction de Dirac pour échantillonner, on peut par exemple utiliser une fonction rectangulaire ou une fonction rampe, mais un Dirac est plus facile à mettre en œuvre (selon mon expérience).

### 1.d. L'échantillonneur électronique.

Revenons au « hardware », car là, on travaille vraiment sur de l'analogique pour en faire du numérique. Dans son principe, un convertisseur analogique-digital est résumée dans la figure 40. Le « switch » constitue le dispositif d'échantillonnage à

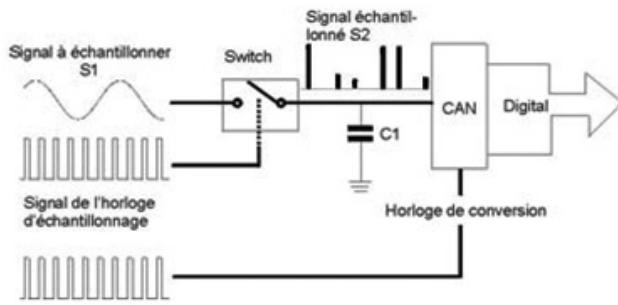


Image 40 : dispositif de numérisation.

est bloqué à la source S. Lorsque l'impulsion  $V_c$  est délivrée, le transistor est conducteur et  $V_a$  en sort par le drain D, est transmise en sortie par l'amplificateur opérationnel A et en même temps charge la capacité C qui sert en somme de « réservoir » intégrateur à  $V_a$ . Puis le générateur repasse à 0 : le transistor n'est plus conducteur et la tension  $V_a$  est bloquée, donc A ne transmet plus rien. La capacité permet de maintenir en  $V_0$  la tension, maintien suffisamment long pour que s'opère la conversion analogique/digitale dans les circuits CAN. C se décharge ensuite à travers la résistance  $R_2$  afin être disponible pour le prochain échantillon. On peut théoriquement modéliser l'impulsion venue du générateur et polarisant la « gate », mais en réalité il s'agit d'une impulsion électrique. Le dispositif ne dépend pas de ce que l'on en dit !

proprement parler ; un exemple de réalisation en est donné par l'image 41. L'entrée du signal analogique se fait en  $V_a$ . Un générateur G fournit une impulsion  $V_c$  contrôlée en fréquence, de l'ordre de 0 à 5V, à la « porte » (Gate) d'un transistor à effet de champ (encadré rouge). Lorsque  $V_c$  est à 0, le transistor ne conduit pas et le signal  $V_a$

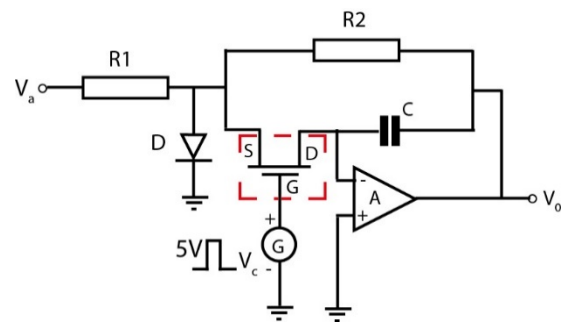


Image 41 : échantillonneur

### 1.e. Le convertisseur analogique-numérique.

Le CAN reçoit une tension électrique qui est encore analogique, et son rôle est d'en faire une quantification numérique. Autrement dit : ce n'est pas l'échantillonnage en soi qui « numérise », mais l'appareil convertissant une grandeur analogique en grandeur numérique. Mais comment passer de l'une à l'autre ? Grâce à un dispositif électronique. L'idée est d'employer les propriétés d'un « engin » analogique capable de recevoir en entrée une tension et de produire un « état » (0 ou 1) en sortie. Cet appareil, c'est l'amplificateur opérationnel (AOP ou ALI, je préfère le premier sigle). C'est un petit circuit intégré. L'image 42 montre les circuits de l'AOP 741 très courant. En fait, comme nous l'avons vu, l'AOP est représenté par une « boîte noire » (rectangulaire ou triangulaire) avec ses entrées et sorties, comme sur

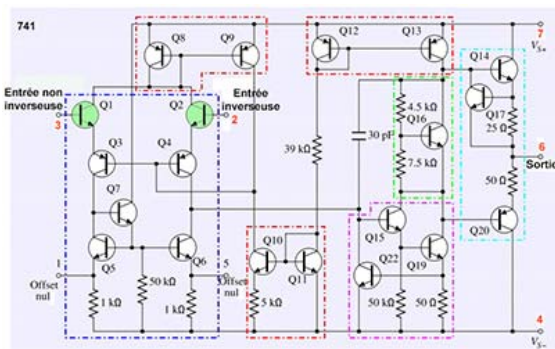


Image 42 Schéma de l'AOP 741

l'image 43 (sur laquelle, l'image 43 : Symbole AOP

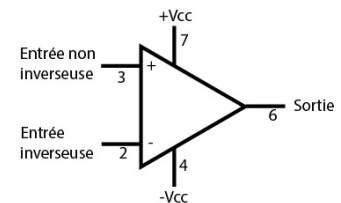


Image 43 : Symbole AOP



contrairement aux habitudes, j'ai fait figurer les alimentations symétriques  $+V_{cc}$  et  $-V_{cc}$ ). Raison : la sortie délivre une tension égale à celle de l'alimentation : si on alimente en 5V, la sortie sera à (très près de) 5V et si je définis l'état 1 à 5 V et l'état 0 à 0 V, j'ai une sortie binaire, non ? Les AOP peuvent servir à de multiples usages (polytropie) : amplification, opérations arithmétique, intégration etc.... mais c'est le fonctionnement en comparateur qui nous intéresse ici. Il y a deux entrées, l'inverseuse et la non-inverseuse ; si une tension de référence  $U_r$  est appliquée à la borne - (2) et une tension à comparer  $U_c$  appliquée à la borne + (3), dès que  $U_c > U_r$ , la sortie (6) délivre une tension très proche de celle de l'alimentation. Si l'AOP fonctionne en 5V, tant que  $U_c < U_r$ , la tension en sortie est nulle. Attention : le basculement ne s'opère pas si  $U_c = U_r$ , mais seulement à partir du moment où  $U_c > U_r$ , ce qui entraîne évidemment une incertitude sur la quantité  $U_c$  !

On peut ainsi réaliser un *convertisseur* comme celui de la figure 44. Un générateur  $E_1$  fournit les tensions de référence (niveaux indiqués par les flèches vertes) par l'intermédiaire d'un diviseur de tension (loi de Kirchoff + loi d'Ohm). Ces tensions sont appliquées aux entrées inverseuses de quatre AOP (connexion noire).

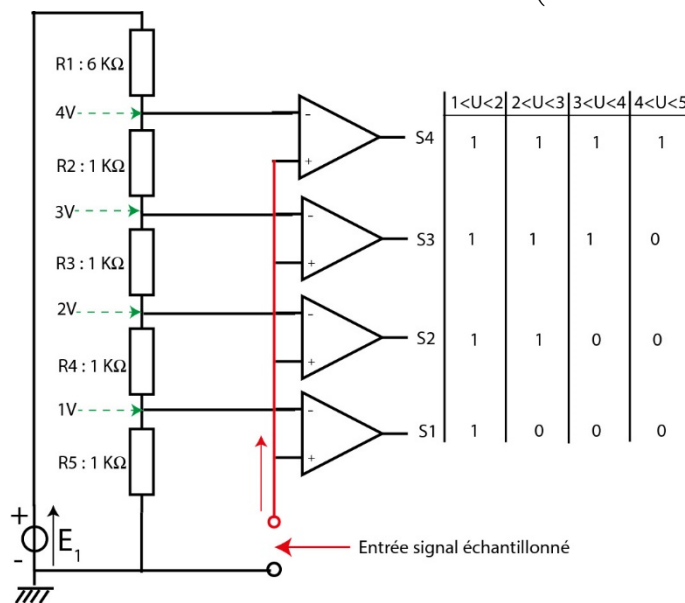


Image 44 : un convertisseur "flash"

l'intermédiaire d'un diviseur de tension (loi de Kirchoff + loi d'Ohm). Ces tensions sont appliquées aux entrées inverseuses de quatre AOP (connexion noire).

La tension  $U_c$  provenant de l'AOP de l'échantillonneur est appliquée aux entrées non-inverseuses (+, connexion rouge).

Les AOP sont réglés de telle manière qu'ils basculent lorsque  $U_c > U_r$ . Par exemple pour une amplitude d'échantillon  $1 < U_c < 2$ , la sortie S1 délivre une tension que l'on notera 1. Dans ce cas, tous les

AOP en amont délivrent une tension nulle (0). On comprend que tous soient à l'état 1 lorsque  $4 < U_c < 5$ . C'est pratique, certes, puisqu'au nombre de sorties différentes de zéro, on connaît l'amplitude « mesurée » 1110 indique 3V, par exemple. L'ennui est que ça s'arrête à 4V, alors qu'en théorie on devrait pouvoir mesurer une amplitude de 5V ! Et si j'entre une amplitude de 3.85V, elle sera évaluée par excès comme 4V, tandis qu'une amplitude de 3.28V ne déclenchera pas le basculement et sera donc réduite à 3 V. On travaille donc par « plages », ici une plage de 5V, que l'on peut schématiser par un diagramme (figure 45), mais qui ne prend pas en compte 5V, car passée la valeur de 4V le convertisseur ne changera plus d'état. On peut y remédier de plusieurs façons, d'abord en faisant intervenir le

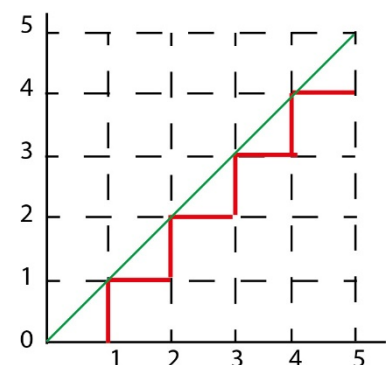


Image 45 : plage de conversion

basculement à  $\frac{1}{2}$  de la valeur de chaque palier (3.5 pour 4, par exemple) comme le montre la figure 46. On peut aussi augmenter le nombre de comparateurs, en choisissant un pas de référence plus petit (davantage de *résolution*). Cela devient assez rapidement coûteux. On minimise l'erreur, et les procédés techniques ne manquent certes pas, mais on ne l'élimine jamais complètement.

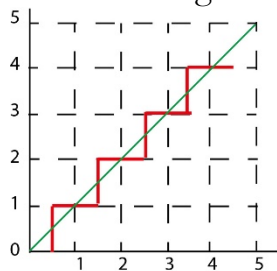


Image 46

Le type de convertisseur décrit ici, dit *convertisseur flash* n'est pas le seul possible, mais il est très employé notamment dans le procédé de numérisation aboutissant au compactage MP3. À la sortie du convertisseur, nous avons

bien une valeur numérique, mais qui n'est pas encore en code binaire. Pour effectuer cette opération de binarisation, il faut un *encodeur* capable de compter le nombre d'états 1 en sortie du convertisseur. Je ne décrirai pas ici le circuit dans la « boîte noire » et son fonctionnement, il fait appel à un assez grand nombre de portes logiques donc l'analyse du comportement serait fastidieuse.

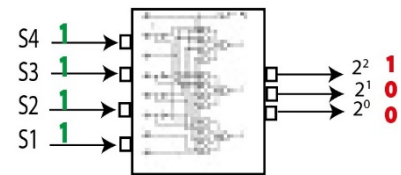


Image 47 principe de l'encodeur

| Sorties | Binaire | déc. |
|---------|---------|------|
| 0000    | 000     | 0    |
| 0001    | 001     | 1    |
| 0011    | 010     | 2    |
| 0111    | 011     | 3    |
| 1111    | 100     | 4    |

L'important est de savoir que le convertisseur *flash* choisi en exemple peut présenter en sortie plusieurs valeurs « hautes » (1111, 0111, 0011, 0001) et que l'encodeur doit donner la priorité à la sortie haute du numéro le plus élevé (4, 3, 2, 1). En sortie de l'encodeur, on obtient trois valeurs binaires ( $2^2$ ,  $2^1$ ,  $2^0$ ), ce qui autorise une plage de mesure acceptable allant jusqu'à 7V d'amplitude. Nous avons réalisé ici un codage sur 3 bits.

Pour montrer l'approximation de la CAN (en l'exagérant quelque peu, je le confesse !), j'ai fabriqué un petit programme qui met en œuvre la génération d'un signal sinusoïdal, son interception par un peigne de Dirac et la quantification avec en sortie une série d'histogrammes de base  $\Delta t$  et de hauteur  $A_i$  en tenant compte des amplitudes négatives. Les résultats sont récapitulés figure 48 :

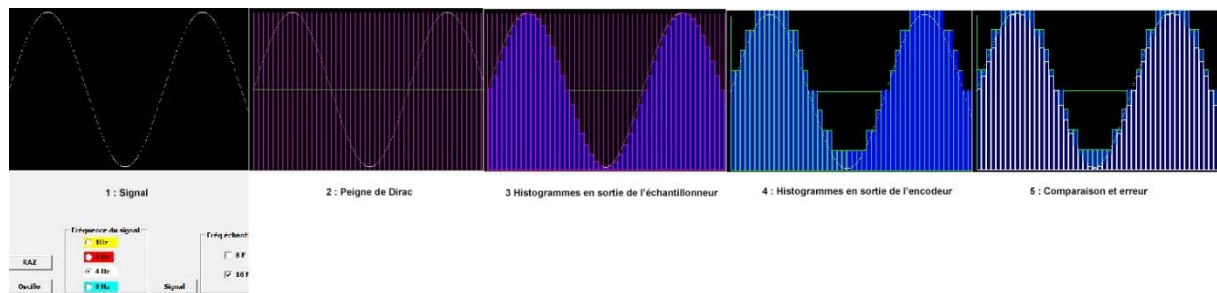


Image 48 : numérisation

(Bon, il faut un peu zoomer pour bien voir, mais on constate que si les histogrammes suivent bien la courbe en sortie de l'échantillonneur, ce n'est plus le cas après quantification binaire. La différence apparaît en bleu dans la dernière image)

## 1.f. Récapitulons.

Récapitulons : donc, nous prélevons des échantillons à une fréquence, rythmée par une horloge,  $F_{\text{échantillonnage}}$  au moins supérieure à 2,56 fois la fréquence maximale du signal (encore faut-il la connaître !) d'un signal d'amplitude  $V_m$  ou  $A_m$ . C'est la tâche de l'échantillonneur, dont chaque partie effectue des tâches distinctes, mais en synergie avec le convertisseur et l'encodeur. Le convertisseur donne une approximation aussi proche que possible de la valeur de l'amplitude du signal à chaque instant  $t + \Delta t$ . Enfin, l'encodeur utilise les sorties du convertisseur pour donner une valeur binaire de l'amplitude. L'opération de numérisation est terminée. Opération dont le résultat est nécessairement approximatif, mais nous savons désormais que le passage au numérique fait perdre de l'information.

Nous avons affaire -les schémas électroniques le montrent- à des *dispositifs* composites répondant à des *utilités* composites. Mais au fond, ce que je viens de faire est presque en tous points analogue à la vieille grammaire de l'École de ma jeunesse ! Il me semble extrêmement difficile d'analyser ergologiquement par exemple le *fabricant* en termes de *matériau* et d'*engin*. Car qu'est-ce qui fait du *matériau*, ici ? Dans le cas de l'horloge, par exemple, le quartz n'est matériau qui débité selon une géométrie particulière (en particulier en épaisseur, ce qui détermine sa fréquence de résonance), de même que pour les transistors constituant les portes (ou l'AOP, ça fonctionne aussi !) du trigger de Schmitt le silicium ne fait pas matériau à lui seul, mais débité en cristal et dopé d'impuretés positives ou négatives. Il y a certainement moins d'engins que de machines possibles. On peut à la rigueur analyser générativement le quartz résonnant et le transistor comme des engins. Mais ensuite, le transistor -polytrope évidemment- entre en unité quasi « syntagmatique » (j'ai jadis proposé le terme de *coalescence*) avec d'autres composants (capacités, résistances...) pour constituer des unités supérieures, par exemple la porte logique etc... Donc nous avons bien *une* analyse technique, mais *plusieurs* niveaux d'analyse *industrielle*, ce qui rend la tâche ardue. Quant au résultat, il n'est rien de mieux qu'une succession d'états physiques dans un *registre*, élément de mémoire absolument matériel. Cela étant, il faut ajouter que le CAN ne vaut qu'en ce qu'il « appelle » d'autres dispositifs avec lesquels il entrera en coalescence, par exemple un convertisseur MP3 puis un fichier puis un décompacteur MP3 puis un convertisseur numérique/analogique (CNA), un amplificateur, des écouteurs etc.... avant qu'Alicia (la dame de la seconde causerie) n'entende la *Toccata et Fugue* en ré mineur. Ou bien il servira autrement, tout dépend de ce à quoi industriellement on le destine.

Parenthèse : (la numérisation ne date pas des années soixante avec la vogue du « sampling » (échantillonnage). Pensons que les horloges mécaniques apparues à la fin du Moyen-Âge (1283, paraît-il) étaient déjà des échantillonneurs de la rotation de la Terre sur son axe (mais on ne le savait pas !), en prenant pour point zéro le moment de passage du Soleil au zénith, avec une course de  $2\pi$  rd par 24 heures. Le mécanisme

d'échappement, amélioré par Huygens puis Graham (échappement à ancre) confère au mouvement du balancier une période constante, un  $\Delta t$ .) Fin de parenthèse.

J'ai organisé jusqu'ici l'exposé en une partie théorique (avec notamment l'impulsion de Dirac) et une partie technique (avec l'échantillonneur en premier). Voilà qui permet de distinguer deux ordres : celui de la logique et celui de la technique. On ne peut pas dire que l'échantillonneur fonctionne avec des impulsions de Dirac : le top d'horloge est une impulsion électrique qui « ouvre » la porte (Gate) presque instantanément pour qu'une partie du signal passe de source en drain (cf image 41), puis, en disparaissant, la referme. Nous avons-là un procédé strictement technique. L'impulsion de Dirac combinée à l'équation du signal n'est en somme qu'une représentation logique, une *modélisation* parmi d'autres modélisations possibles. Bien entendu, le modèle peut être techniquement mis en œuvre, « *voilà ce que l'on peut mathématiquement faire pour concilier fréquence, amplitude, erreur minimale dans la numérisation ; à vous, électroniciens, de vous débrouiller pour en tirer parti !* » Il n'empêche : les électroniciens opèrent une toute autre analyse !

## 2. Transformée de Fourier et dispositif électronique de calcul numérique.

### 2.a. Signaux périodiques avec harmoniques.

Avançons. Un signal est rarement pur, il peut être perturbé par des bruits, résulter de la modulation d'une onde porteuse par une onde modulante etc... Et même, nous dit Fourier (1768-1830), une simple sinusoïde... En effet, démontre Fourier, tout signal périodique peut être décomposé en une somme de fonctions sinusoïdales. Un

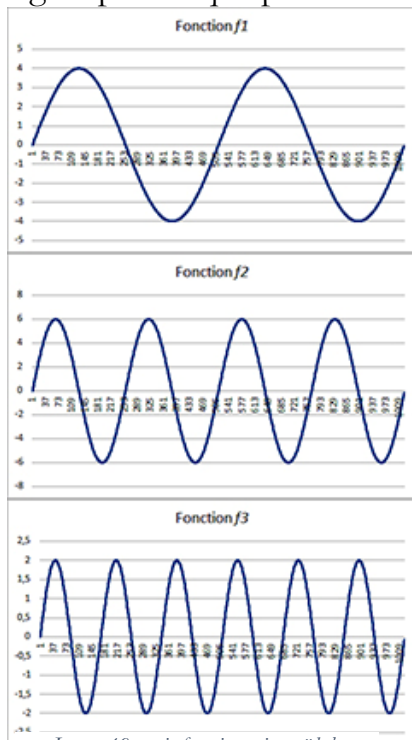


Image 49 :trois fonctions sinusoïdales

exemple : donnons-nous un  $\Delta t = 1/1024$ , définissons trois fonctions sinusoïdales  $f_1$  { Fréq<sub>1</sub> = 2 Hz, Ampl<sub>1</sub> = 4V},  $f_2$  {Fréq<sub>2</sub> = 4 Hz, Ampl<sub>2</sub> = 6V},  $f_3$  {Fréq<sub>3</sub> = 6 Hz, Ampl<sub>3</sub> = 2V}. Chacune est bien une sinusoïde comme l'indique la figure 49.

Additionnons ces trois signaux, et nous obtenons un signal plutôt sinusoïdal, mais loin d'être une sinusoïde parfaite (figure 50)

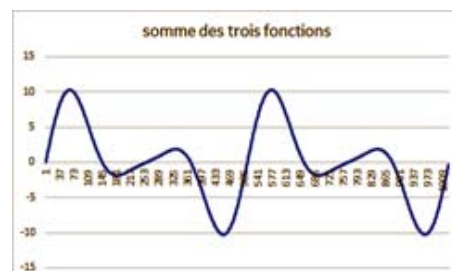


Image 50 Signal résultant

Cela pourrait être un signal quelconque, placé en entrée d'un CAN. Encore pourrait-on le compliquer à loisir en introduisant des décalages de phases entre les trois composantes ou *harmoniques* <sup>6</sup>.

<sup>6</sup> Harmonique est masculin. Le terme est employé de manière analogique à la physique des sons.

Dans la représentation de la figure 50, nous voyons ce que nous indiquerait un oscilloscope. Examinée ainsi sur l'axe du temps, la courbe ne nous donne aucune indication sur le *spectre* fréquentiel<sup>7</sup> ; en revanche, imaginons un axe des fréquences : il pourrait nous indiquer quelles sont les fréquences composant le signal, le *fondamental* et les harmoniques. Pourquoi connaître ce spectre ? Parce que quelquefois on a besoin de *filtrer* des fréquences parasites, il faut donc les identifier. La numérisation ne les élimine pas (et même peut introduire des harmoniques parasites sous de mauvaises conditions). L'accès aux fréquences des harmoniques d'un signal nous est donné par la *transformée de Fourier*.

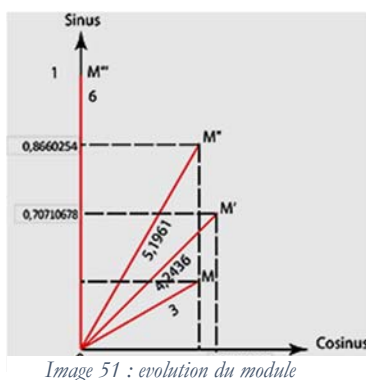
## 2.b. La transformée de Fourier (un peu de maths, mais ce n'est qu'un survol de la question).

Essayons de simplifier l'exposé. Dans l'exemple précédent, nous avons utilisé pour produire le signal trois fonctions sinus en *relation harmonique*, c'est-à-dire que si le fondamental a une fréquence F1, l'harmonique 1 aura une fréquence F2 = 2F, l'harmonique 2 une fréquence F3 = 3F etc... Nous avons ainsi F1 = 2 Hz, F2 = 4 Hz, F3 = 6 Hz... Il s'ensuit que si la pulsation du fondamental est  $\omega = 2\pi F1$ , celle de l'harmonique 1 sera  $2\pi F2 = 2\omega$ , celle de l'harmonique 2 sera  $2\pi F3 = 3\omega$ . On peut rassembler cela sous la forme d'une sommation :  $f(t) = \sum_{n=1}^N \text{Sin}(n \omega t)$  qui équivaut à  $f(t) = \text{Sin}(\omega t) + \text{Sin}(2\omega t) + \text{Sin}(3\omega t) + \dots + \text{Sin}(N\omega t)$ . Si l'on veut faire intervenir les amplitudes  $A_n$ , on écrira  $f(t) = \sum_{n=1}^N A_n \text{Sin}(n \omega t)$ .

Ce que l'on numérise, en fait, c'est une amplitude (tension, dans notre cas) à un instant  $t$ . On pourrait considérer cette amplitude comme un vecteur :

| Degrés | Radians    |         | sinus      | cosinus    | module     | t pour atteindre    | t mouvement $2\pi+n\Delta t$ |
|--------|------------|---------|------------|------------|------------|---------------------|------------------------------|
| 30     | 0,52359878 | $\pi/6$ | 0,5        | 0,8660254  | 3          | 42,6666667 $\Delta$ | 1,04166667 Sec               |
| 45     | 0,78539816 | $\pi/4$ | 0,70710678 | 0,70710678 | 4,24264069 | 64 $\Delta$         | 1,0625 Sec                   |
| 60     | 1,04719755 | $\pi/3$ | 0,8660254  | 0,5        | 5,19615242 | 85,3333333 $\Delta$ | 1,08333333 Sec               |
| 90     | 1,57079633 | $\pi/2$ | 1          | 0          | 6          | 128 $\Delta$        | 1,125 Sec                    |

Ce tableau est à comparer avec le schéma de l'image 51 :



Le vecteur -module en rouge- prend différentes valeurs OM, OM', OM'', OM''' en même temps que varie le temps, donc l'angle qu'il fait avec l'axe des cosinus. Je suis parti d'une pulsation  $\omega = 2\pi F$  avec  $F = 2$  et un  $\Delta t = 1/1024$ . L'angle varie de 0,08333333 rd par  $\Delta t$ , ce qui permet de calculer le temps qu'il faut pour atteindre un angle de  $\theta$ rd soit en partant de zéro, soit en partant de  $2\pi$  (ce qui revient au même).

<sup>7</sup> Spectre : on peut penser au spectre lumineux, la lumière blanche décomposée par un prisme nous donne le spectre des fréquences lumineuses. Il en va de même pour les signaux électromagnétiques, sonores etc...

À tout point M, M'... de coordonnées rectangulaires x et y, on peut faire correspondre l'ensemble  $z\{x,y\}$  de ces deux nombres donnés dans cet ordre. z est appelé *nombre complexe*. On appelle *module* de z la longueur OM ou OM' etc., tel que  $|z| = OM = \sqrt{x^2 + y^2}$ . x est la *partie réelle* de z et y sa *partie imaginaire*. Si l'on désigne par  $\rho$  la longueur de OM, et par  $\theta$  l'angle (ou *argument* de z), on a les relations  $\rho^2 = x^2 + y^2$ ,  $\text{tg}\theta = y/x$ ,  $x = \rho \cos \theta$ ,  $y = \rho \sin \theta$ . J'ai fait ce détour par les représentations trigonométriques des complexes pour faire comprendre que la transformée de Fourier utilise le module, le sinus et le cosinus. D'où la formule générale :

$$f(t) = A_0 + \sum_{n=1}^N A_n \text{Sin}(n \omega t) + \sum_{n=1}^N B_n \text{Cos}(n \omega t)$$

$A_0$ ,  $A_n$  et  $B_n$  sont les *coefficients* des séries de Fourier. S'il existe N harmoniques, il y aura N sommations.  $A_0$  représente la composante continue du signal ou *valeur moyenne*.  $A_n$  et  $B_n$  sont les composant d'amplitude de rang n du spectre. Tels sont les principes généraux.

Donc revenons à un signal composite (= comportant des harmoniques) : nous devons le décomposer en séries et calculer l'amplitude et la phase de chacun des harmoniques : c'est le calcul du *spectre*. Nous allons (Désolé ! Nous faisons beaucoup de math !) faire appel au calcul de l'*intégrale* d'une fonction. Intuitivement, c'est assez facile : si l'on trace le graphe de la fonction, l'intégrale est l'*aire* comprise entre les deux bornes de la fonction, la partie de graphe concernée et l'axe des abscisses



Image 52 : intégrale

comme le montre la figure 52 où l'intégrale est l'aire bleutée. Comment peut-on la calculer ? Il y a plusieurs méthodes, par exemple celle de Simpson, ou celle des rectangles, ou encore celle des trapèzes. Cette dernière est facile à mettre en œuvre dans le cas d'un signal numérisé, et elle donne une assez bonne approximation de l'intégrale. Le principe est

résumé figure 53. La surface en jaune, somme des surfaces des trapèzes, est une approximation de l'intégrale de la fonction sur l'intervalle [1..149] (on note  $\int_1^{149} f(t) dt$ ). Le tout est de se donner un *pas d'intégration* suffisamment petit pour obtenir un grand nombre de trapèzes. Or dans une série

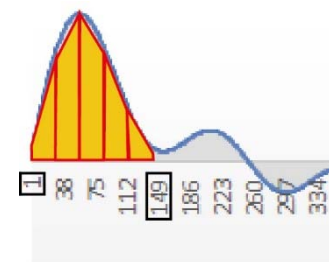


Image 53 : méthode des trapèzes

de valeurs numériques, une valeur  $V_t$  est séparée de la précédente  $V_{t-\Delta t}$  par la distance  $\Delta t$  qui est l'accroissement du temps, et si l'échantillonnage a lieu tous les 1/1024 de seconde, la hauteur de chaque trapèze vaut  $\Delta t$  tandis que ses bases valent  $a = V_{t-\Delta t}$  et  $b = V_t$ . On obtient ainsi une excellente approximation de l'intégrale. Il y a d'autres procédés d'intégration : Newton-Cotes, Gauss-Legendre, Monte-Carlo.

Il est évident que dans le cas d'une sinusoïde parfaite, l'intégrale vaut zéro, puisque l'aire comprise dans la partie supérieure de la courbe est égale en valeur absolue à celle de la partie inférieure, mais de signe opposé. Il n'en va pas de même si le signal

est composé de signaux en relation harmonique : l'intégrale est alors différente de zéro (je simplifie !).

Tout cela peut se vérifier facilement en créant un signal composite avec N harmoniques sous Excel, en se donnant un  $\Delta t$  de 1/1024. Mais revenons à Fourier. Si l'intégrale est non nulle, on passera au calcul des coefficients de Fourier, qui fait appel à l'intégration selon les formules :

$$A_0 = \frac{1}{T} \int_0^T f(t) dt, A_n = \frac{2}{T} \int_0^T f(t) \sin(n \omega t) dt, B_n = \frac{2}{T} \int_0^T f(t) \cos(n \omega t) dt$$
 <sup>8</sup>

T est la *période* du signal. On voit dans la figure 52 que le même motif de graphe se répète deux fois dans l'intervalle [1..1024], ce qui veut dire que si 1024 représente une seconde, le retour de la valeur du signal à sa valeur initiale demande exactement la moitié de 1024, soit 1/2 seconde : sa période est donc de 1/2 seconde et sa fréquence est de 2 Hz puisqu'il y a, dans l'exemple, deux rotations par seconde.

Donc, très schématiquement, pour trouver les composantes spectrales d'un signal, on prendra le signal d'origine  $f(t)$  que l'on multipliera par un autre signal -ce que signifie par exemple, dans la formule supra,  $f(t) \sin(n \omega t)$  et  $f(t) \cos(n \omega t)$  – et on intègre. Si l'intégrale = 0, le signal multiplicateur n'est pas un harmonique, alors on l'ignore ; dans le cas contraire, on calcule les coefficients. On prendra ensuite l'harmonique suivant et on répètera le processus.

On obtiendra donc un ensemble de coefficients qui permettront de calculer le spectre. Là, ce n'est pas trop fastidieux : on pose  $C_n$ , un nombre complexe, dont la valeur absolue est  $|C_n| = \sqrt{A_n^2 + B_n^2}$  qui correspond au *spectre d'amplitude* (ou de module) de chaque harmonique. On peut le représenter par un bâton dont la longueur est  $|C_n|$ . Le *spectre de phase* se calcule par  $\text{Arctan}(B_n/A_n)$ . Les figures 54 et 55 nous donnent un aperçu du spectre du signal composite de la figure 50 <sup>9</sup> :



Image 54

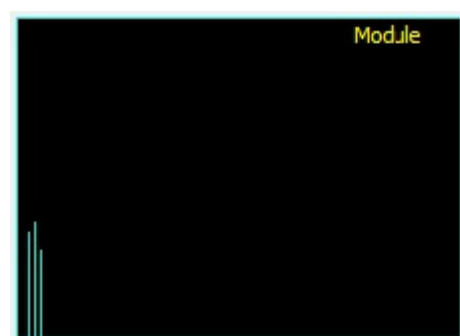


Image 55

<sup>8</sup> Dans ces formules, la notation dt est la survivance d'une ancienne conception de l'intégration, où on prenait la limite de sommes dans lesquelles intervenait un "petit accroissement de la variable" qu'on notait dt.

<sup>9</sup> La 54 est faite par Excel : on a bien une échelle, mais on ne distingue pas bien le fondamental et ses deux harmoniques. La 55 est faite par mon propre analyseur de Fourier (programme en Delphi), on distingue mieux les trois composantes (le fondamental est le premier bâton à gauche), mais je n'indique pas l'échelle, qui est logarithmique.

sur ces graphes, l'abscisse n'est plus le temps, mais la fréquence. Nous n'avons donc plus une fonction du type  $f(t)$ , mais une autre du type  $f(F)$ .

Ouf ! J'ai expliqué la transformation par la méthode qui me paraît la plus simple - sans avoir à s'occuper de l'exponentiation (développement des séries de Fourier en les réécrivant comme la partie réelle d'exponentielles complexes) - et la plus proche des méthodes informatiques dont j'ai l'habitude <sup>10</sup>. Bon... Il va de soi qu'une fois connus les coefficients, on peut opérer la reconstitution du signal en appliquant la transformation inverse, absolument symétrique de la précédente. Le tableau ci-dessous utilise le calcul par l'exponentielle  $F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-jt\omega} dt$  et l'inverse  $f(t) = \int_{-\infty}^{+\infty} F(\omega)e^{jt\omega} d\omega$ . Tableau de 8 lignes sur 1024 en réalité.

| $t + \Delta t$ | $f(t)$     | $-j*t*\omega$ | $e^{-j*t*\omega}$ | $F(\omega)$ | $j*t*\omega$ | $e^{j*t*\omega}$ | $f(t)$     |
|----------------|------------|---------------|-------------------|-------------|--------------|------------------|------------|
| 0,00097656     | 0          | 0,01227185    | 0,98780315        | 0           | 0,01227185   | 1,01234745       | 0          |
| 0,00195313     | 0,26994797 | 0,02454369    | 0,97575505        | 0,2634031   | 0,02454369   | 1,02484737       | 0,26994797 |
| 0,00292969     | 0,53970009 | 0,03681554    | 0,96385391        | 0,52019204  | 0,03681554   | 1,03750162       | 0,53970009 |
| 0,00390625     | 0,80906069 | 0,04908739    | 0,95209793        | 0,770305    | 0,04908739   | 1,05031213       | 0,80906069 |
| 0,00488281     | 1,07783449 | 0,06135923    | 0,94048533        | 1,01368752  | 0,06135923   | 1,06328081       | 1,07783449 |
| 0,00585938     | 1,34582677 | 0,07363108    | 0,92901436        | 1,2502924   | 0,07363108   | 1,07640962       | 1,34582677 |
| 0,00683594     | 1,61284358 | 0,08590292    | 0,91768331        | 1,48007964  | 0,08590292   | 1,08970054       | 1,61284358 |
| 0,0078125      | 1,8786919  | 0,09817477    | 0,90649046        | 1,70301629  | 0,09817477   | 1,10315557       | 1,8786919  |

L'image 56 montre un graphe tiré de la Transformée. Le pic correspond exactement au spectre de la fonction  $f(t)$ , qui est une sinusoïde, prise sur 1024 échantillons.

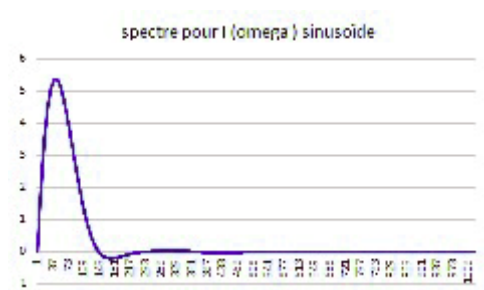


Image 56 : spectre de  $f(t)$

En y réfléchissant bien, une transformation de Fourier agit un peu comme un filtre. Imaginons un signal composite avec un fondamental  $\omega$  et une des composante non-harmonique par exemple  $(5*\sin(\omega t)) + (5*\sin(1,5*\omega t)) + (5*\sin(3*\omega t))$ . Si l'intégrale de la composition du fondamental avec la première composante est nulle, alors la première composante n'est pas en rapport harmonique, donc elle éliminée (coeff = 0). Elle n'apparaîtra donc pas dans la transformation inverse et le signal de sortie du dispositif  $S_e \rightarrow TF \rightarrow InvTF \rightarrow S_s$  sera différent du signal d'entrée. On peut aussi éliminer volontairement un harmonique d'une fréquence considérée comme parasite, par exemple supprimer du « souffle » en éliminant une harmonique de haute fréquence.

<sup>10</sup> Pour une approche plus fouillée, voir : [https://cdn.uclouvain.be/public/Exports%20reddot/fsa/documents/Travail\\_Complet.pdf](https://cdn.uclouvain.be/public/Exports%20reddot/fsa/documents/Travail_Complet.pdf)



## 2.c. La transformée de Fourier rapide (TFR) : l'algorithme.

Mais... Dans le monde numérique, en matière de réalisation technique, il faut aller vite. Or la quantité de calculs est énorme, car l'on effectue une transformée sur, qui plus est, des données trigonométriques<sup>11</sup>. Pour  $N=1024$  points comme dans nos exemples, il faut  $N^2$  soit 1 048 576 pas de calcul, ce qui prend du temps. C'est pourquoi on a inventé la transformée de Fourier rapide (TFR en Français, FFT ou Fast Fourier Transform, en technoricaïn). Celle-ci permet de gagner un facteur 100 sur le calcul de 1024 points :  $1024 * \log_2(1024) = 10\,240$  pas de calcul seulement (soit  $N * \log_2(N)$ ). Le principe est simple : au lieu d'effectuer une seule transformée complexe, on en calcule plusieurs élémentaires. Pour cela, on a recours au calcul matriciel, en effectuant des opérations et des permutations sur les éléments d'un tableau. Je vais tenter d'expliquer cela de manière simple, même si cela m'a demandé quatre jours de travail pour essayer de simplifier.

Voici une procédure -assez courte comme on le verra- de TFR appliquant l'algorithme dit de Cooley-Tukey, le plus employé<sup>12</sup>. La procédure est écrite en Pascal pour Delphi 7.

```
PROCEDURE tfr;
VAR x, x1, P2, auxiliaire, ip: INTEGER;
    demiP2, wr, wi, br, bi, ua, ur, ui, TR, TI: REAL;
BEGIN
    FOR x:= 0 TO exposant-1 DO
    BEGIN
        P2 := tableconstantes[x] ;
        demiP2 := P2/ 2;
        ur := 1;
        ui := 0;
        wr := COS (pi/ demiP2);
        wi := SIN(pi/ demiP2);
        FOR x1 := 0 TO TRUNC(demiP2)-1 DO
        BEGIN
            auxiliaire := x1;
            REPEAT
                br := PR[auxiliaire] ;
                bi := PI[auxiliaire] ;
                ip := auxiliaire+ TRUNC(demiP2);
                TR := PR[ip]* ur- PI[ip]* ui;
                TI := PR[ip]* ui+PI[ip]* ur;
                PR[ip] := br- TR;
                PI[ip] := bi- TI;
                PR[auxiliaire] := br+ TR;
                PI[auxiliaire] := bi+ TI;
                auxiliaire := auxiliaire+ TRUNC(P2);
            UNTIL auxiliaire>= taille;
            ua := ur* wr- ui* wi;
            ui := ur* wi+ ui* wr;
            wr:= ua;
```

---

<sup>11</sup> Un ordinateur ne dispose pas systématiquement de tables trigonométriques natives (built-in), il calcule les valeurs à l'aide de séries, et c'est long pour les faire converger, même si l'opérateur ne s'en rend pas compte

<sup>12</sup> Il existe un autre algorithme, dit de Sørensen, mais ne le pratiquant pas, je le connais mal.

END;  
 END;  
 END;

Examinons les variables.  $x$  est la variable de la première boucle *for*  $x := 0$  to *exposant*. L'exposant est 3 si l'on fait une transformée sur 8 échantillons, car  $2^3=8$ . La puissance de 2, notée par la variable P2, sera donc 8 quand  $x=3$ ; voici comment elle varie jusqu'à l'exposant 10 correspondant à une transformée sur 1024 échantillons :

|      |    |   |            |            |            |            |            |            |            |            |
|------|----|---|------------|------------|------------|------------|------------|------------|------------|------------|
| b    | 0  | 1 | 2          | 3          | 4          | 5          | 6          | 7          | 8          | 9          |
| P2   | 2  | 4 | 8          | 16         | 32         | 64         | 128        | 256        | 512        | 1024       |
| P2/2 | 1  | 2 | 4          | 8          | 16         | 32         | 64         | 128        | 256        | 512        |
| Wr   | -1 | 0 | 0,70710678 | 0,92387953 | 0,98078528 | 0,99518473 | 0,99879546 | 0,99969882 | 0,9999247  | 0,99998118 |
| Wi   | 0  | 1 | 0,70710678 | 0,38268343 | 0,19509032 | 0,09801714 | 0,04906767 | 0,02454123 | 0,01227154 | 0,00613588 |

( $b$  est la variable de boucle  $x$ ), P2/2 du tableau correspond à la variable demiP2. Les variables Wr et Wi sont des « twiddle factors », appelés habituellement « facteurs de rotation », mais qui plus prosaïquement, étant donné le sens péjoratif de « to twiddle », « facteurs de tripotage », ce qui sous-entend un petit bricolage trigonométrique pour que tout fonctionne bien<sup>13</sup>. *br*, *bi*, *ua*, *ur*, *ui*, *TR*, *TI* sont des variables de transfert utiles dans les opérations. PR et PI sont des tableaux contenant les valeurs des échantillons : PR pour la partie réelle (cosinus), PI pour la partie imaginaire (sinus).

La grande boucle *FOR*  $x := 0$  TO *exposant-1* DO initialise les facteurs utiles aux pas de calcul de la boucle imbriquée *FOR*  $x1 := 0$  TO *TRUNC(demiP2)-1* DO. L'essentiel des calculs se situe dans la boucle *REPEAT...UNTIL* *auxiliaire*  $\geq$  *taille*; mais à chaque pas de la boucle imbriquée, on recalcule les facteurs  $u_i$  et  $u_r$  en fonction des facteurs  $W_i$  et  $W_r$ . Ce qui est important, pour comprendre le mécanisme, est de savoir ce qui rentre et ce qui sort de la boucle *REPEAT... UNTIL*. On voit qu'il rentre PR[auxiliaire], PI[auxiliaire], PR[ip] et PI[ip], et que ces valeurs en sortent modifiées par le traitement

$$TR := PR[ip] * u_r - PI[ip] * u_i;$$

$$TI := PR[ip] * u_i + PI[ip] * u_r;$$

Les échantillons sont initialement rangés dans l'ordre temporel d'échantillonnage  $t_n - \Delta t$ ,  $t_n$  etc... Mais ils ne sont cependant pas présentés dans cet ordre. En fait, ils sont réarrangés selon une *permutation binaire*. Il s'agit d'une opération portant sur les indices des tableaux PR et PI, notés entre [ ] dans le programme. On prend l'indice, par exemple 4, on le convertit en binaire, 100, et on le retourne comme peau de lapin : 001. On permute donc les valeurs de PR[4] et PR[1] (voir tableau ci-dessous) portant sur 8 échantillons :

<sup>13</sup> De même que « to twitter » veut dire « jacasser », ce qui me semble bien correspondre à l'usage de Twitter !

| N° | Réel       | Imaginaire | origine | permuté | N° permuté | Réel       | Imaginaire |
|----|------------|------------|---------|---------|------------|------------|------------|
| 0  | 0,31722803 | 0,3015721  | 000     | 000     | 0 (0)      | 0,31722803 | 0,3015721  |
| 1  | 0,36713252 | 0,6829338  | 001     | 100     | 1 (4)      | 0,7528707  | 0,68840932 |
| 2  | 0,79593034 | 0,25083446 | 010     | 010     | 2 (2)      | 0,79593034 | 0,25083446 |
| 3  | 0,32481573 | 0,69129289 | 011     | 110     | 3 (6)      | 0,13844542 | 0,53302621 |
| 4  | 0,7528707  | 0,68840932 | 100     | 001     | 4 (1)      | 0,36713252 | 0,6829338  |
| 5  | 0,16279739 | 0,06044904 | 101     | 101     | 5 (5)      | 0,16279739 | 0,06044904 |
| 6  | 0,13844542 | 0,53302621 | 110     | 011     | 6 (3)      | 0,32481573 | 0,69129289 |
| 7  | 0,05846545 | 0,34604319 | 111     | 111     | 7 (7)      | 0,05846545 | 0,34604319 |

Par la suite, nous allons référer la position initiale entre parenthèses : 1 (4), 4 (1), juste pour se repérer dans les schémas. Regardons ce qui entre de PR[aux], PR[ip], PI[aux] et PI[ip] dans une TFR portant sur 8 échantillons :

|     |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|
| x   | 0    | 1    | 2    |      |      |      |      |
| x1  | 0    | 0    | 1    | 0    | 1    | 2    | 3    |
| aux | 0    | 0    | 1    | 0    | 1    | 2    | 3    |
| ip  | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
| aux | 2    | 4    | 5    | 8    | 9    | 10   | 11   |
| ip  | 3    | 6    | 7    | stop | stop | stop | stop |
| aux | 4    | 8    | 9    |      |      |      |      |
| ip  | 5    | stop | stop |      |      |      |      |
| aux | 6    |      |      |      |      |      |      |
| ip  | 7    |      |      |      |      |      |      |
| aux | 8    |      |      |      |      |      |      |
|     | stop |      |      |      |      |      |      |

On voit qu'à une valeur de l'auxiliaire « aux » correspond un indice des tableaux PR et PI ; il en va de même pour chaque valeur de ip. Pour mémoire, rappelons les indices avant la permutation binaire (ci-contre) :

Le calcul s'arrête lorsque *auxiliaire* >= *taille*, ici *taille* = 8.

En fin de traitement, les tableaux PR et PI contiennent les coefficients nécessaires au calcul du spectre. Par exemple, le spectre de module s'obtient en calculant :

$$\sqrt{PR[x]^2 + PI[x]^2}$$

et celui de phase par ARCTAN(-(PI[x]/PR[x])). En fait, on n'affiche que le calcul sur la moitié du tableau, l'autre partie étant symétrique (si tout va bien !). Voyons le schéma récapitulatif de la TFR sur notre série de 8 échantillons :

| x1  | 0    | Échantillon |
|-----|------|-------------|
| aux | 0    | (0)         |
| ip  | 1    | (4)         |
| aux | 2    | (2)         |
| ip  | 3    | (6)         |
| aux | 4    | (1)         |
| ip  | 5    | (5)         |
| aux | 6    | (3)         |
| ip  | 7    | (7)         |
| aux | 8    |             |
|     | stop |             |

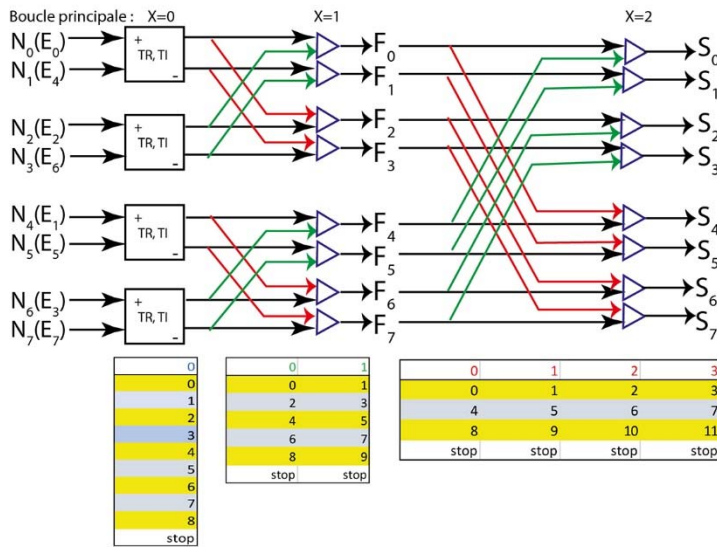


image 57 : schema de la TFR

Chaque bloc géométrique (carré, triangle), représente une transformation élémentaire. Les entrées sont notées par  $N_{[m]}$  avec rappel ( $E_{[m]}$ ) de la place initiale de l'échantillon avant permutation binaire. Sur la première ligne, correspondant à chacune des trois colonnes de transformées, la valeur  $x$  de la boucle principale. En bas, sur la même colonne, les variations des variables « auxiliaire » et « ip ». On a ainsi 8 transformations élémentaires dans la première colonne (2 par carré), 8 dans chacune des autres,

soit 24 transformées au lieu de 64. Les flèches de couleurs joignent les valeurs en entrée en entrée directe. Le graphe montre une structure des combinaisons-permutations assez classique appelée « papillon » (butterfly).

Voilà, je passe sur quelques millions de choses, en particulier les fonctions de fenêtrage et la convolution, les phénomènes d'*aliasing* etc... S'il est évident que la numérisation fait perdre de l'information par rapport à un rayonnement initial, la TFR (comme la transformation cosinus vue dans le cas du JPEG) en fait encore perdre, ne serait-ce que par l'utilisation d'irrrationnels comme  $e$  et  $\pi$  et la trigonométrie dont les fonctions sont en fait calculées par des séries qui convergent lentement<sup>14</sup>. Cela reste cependant dans des limites tolérables, puisque les signaux de sortie sont très proches des signaux d'entrée. Mais la TF n'est pas, pour mon propos, une fin en soi !

### 3. Les dispositifs de calcul.

#### 3.a. Les briques du jeu : des « portes logiques ».

Bon... Nous sommes intéressés par le côté ergologique du numérique, donc mettons les mains dans le cambouis. Toutes ces opérations de la TFR peuvent être accomplies par des circuits électroniques. Au fond, quand j'analyse un circuit en apparence complexe, ou quand je le fabrique sur un *bread-board* (planche de montage expérimental), je pense toujours au jeu de Lego ou au Meccano de mon enfance. Il y a des « briques » fondamentales, qui s'assemblent en unités plus vastes, qui elles-mêmes s'assemblent en modules selon une sorte de processus récursif.

<sup>14</sup> Essayez de calculer le sinus de  $\pi$  avec Excel, vous m'en direz des nouvelles !

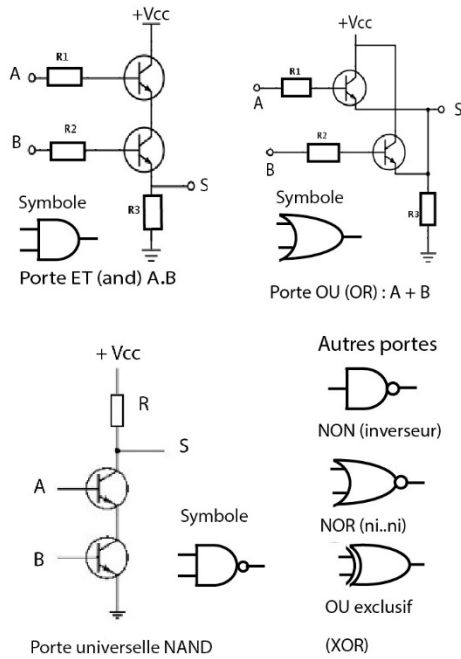


Image 58 : Portes logiques

Examinons ces briques-là, présentées dans l'image 58. Ce sont essentiellement des résistance (symbole rectangulaire), et des transistors. Leur mise en coalescence donne des ensembles fonctionnels que l'on appelle « portes logiques ». L'image montre les circuits des portes ET, OU, NAND, que l'on remplace par leur symbole dans les schémas électroniques. Je me contente d'indiquer seulement les symboles des fonctions NON, NOR, OU exclusif (Xor). Nous avons là les briques fondamentales des circuits numériques. Il y en a d'autres : capacités, bobines (inductances) etc. qui interviennent dans différentes tâches, on en a vu un exemple plus haut avec l'échantillonneur. Mais ceux que l'on présente ici sont essentiels pour réaliser des calculs binaires selon une *Algèbre de Boole*. On trouvera très

facilement sur l'internet des explications sur le calcul booléen, notamment les deux précieux théorèmes de de Morgan. Juste un exemple : l'intersection de deux ensembles, notée  $\cap$  à l'école,  $\wedge$  chez les logiciens,  $\bullet$  en algèbre de Boole, est réalisée par la porte ET dont la *table de vérité* est indiquée dans le tableau ci-contre. Autrement dit : de deux ensemble A et B on ne retient que les valeurs communes.

| A | B | A•B |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

### 3.b. Synopsis d'un dispositif effectuant une transformée de Fourier.

Nous allons voir que la transformée de Fourier se réalise tout aussi bien avec des circuits logiques. La figure 59 est un schéma synoptique de l'appareillage nécessaire à l'opération. On remarque que les calculs fondamentaux abordés dans la modélisation mathématique se retrouvent dans ce schéma, notamment la sommation  $\Sigma$ , les multiplications par le cosinus, par exemple  $\cos(\frac{2\pi k}{N})$  indiqué à côté des multiplicateurs. Précisons : N est le nombre d'échantillons, n le n<sup>ième</sup> échantillon, k le rang d'un harmonique dans la transformation. Les X indiquent les valeurs en entrée. Les temporisateurs

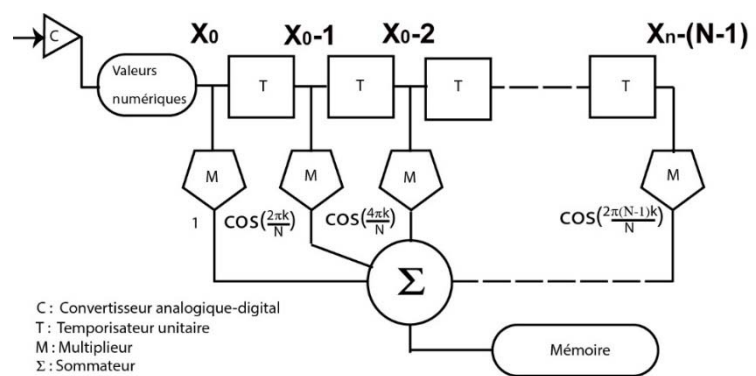


Image 59 : synopsis de la TFD

unitaires permettent le transfert des valeurs numériques d'un multiplieur à l'autre selon une séquence rythmée par les  $\Delta t$  d'une horloge.

### 3.c. Coalescences de dispositifs basiques.

De quoi est composé un temporisateur ? L'idée est de retenir une valeur  $f(x)$  pendant une durée suffisante pour qu'elle soit exploitable par le multiplieur correspondant avant d'être transmise au multiplieur suivant. Pour cela, on utilise ce que l'on appelle une *bascule maître-esclave* elle-même constituée de deux bascules DQ dites élégamment (!!!) « flip-flop ».

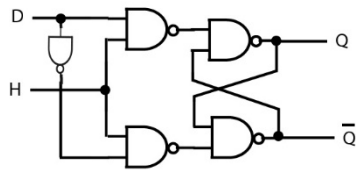


Image 60: Bascule DQ

avec son entrée D, son entrée d'horloge H et ses sorties Q et  $\bar{Q}$ . Au coup d'horloge (1), si  $D=1$ ,  $Q=1$  et  $\bar{Q}=0$ , si  $D=0$ ,  $Q=0$  et  $\bar{Q}=1$ . En associant deux bascules DQ,

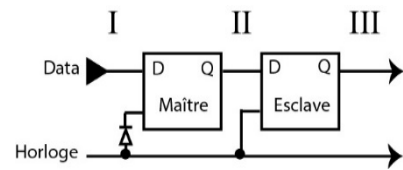


Image 61 : bascule maître esclave

maître-esclave (figure 61) dont le fonctionnement est expliqué figure 62.

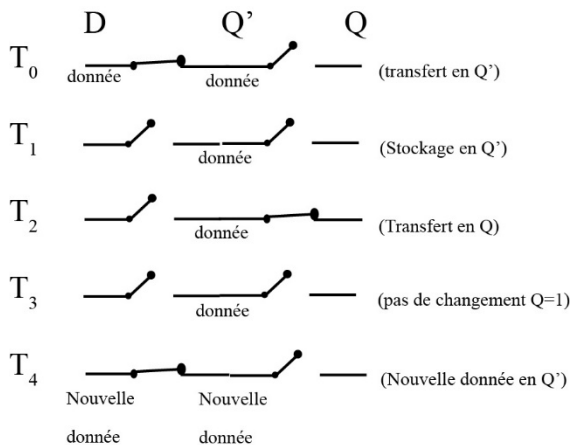


Image 62 : fonctionnement de la bascule DQ

Les interrupteurs fermés indiquent qu'un transfert de valeur s'effectue du maître à l'esclave. On remarquera qu'une valeur demeure en sortie-maître Q' durant quatre impulsions d'horloge, ce qui est suffisant pour qu'elle soit exploitée en  $T_4$  par le multiplicateur. Au cinquième temps ( $T_4$ ), la donnée se trouve à la sortie de la bascule esclave, Q, tandis qu'une nouvelle donnée entrée par D se trouve en Q'. Une donnée parcourt ainsi tout le *bus* jusqu'à la dernière bascule et au dernier multiplieur.

Pour comprendre comment fonctionne un multiplieur, il faut se souvenir qu'une multiplication, en quelque base de numération que ce soit, comporte l'addition des résultats partiels. Il faut donc déjà un dispositif capable de faire des additions, donc de gérer la retenue. Considérons donc un demi-additionneur (image 63a) : c'est un circuit simple comportant une porte OU exclusif (Xor) et, en parallèle, une porte ET.

| A | B | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

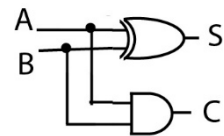


Image 63a

Le OU exclusif ne laissera passer que deux configurations :  $A=1$  et  $B=0$  ou  $A=0$  et  $B=1$ , pour que S soit à l'état 1. Dans les deux autres cas,  $S=0$ . Quant au ET, il sera à 1 uniquement si  $A=1$  et  $B=1$ . Si on additionne  $1+1$ , cela donne 2, soit 10 en base 2. Donc : à la sortie S,  $1 \oplus 1 = 0$  et à la sortie C,  $1 \cdot 1 = 1$ . Nous avons bien 10, mais diffracté en S (résultat de l'addition) et en C

(retenue !). C'est ce que résume la table et la figure 63a. Un additionneur complet devra prendre en compte une éventuelle retenue et fournir lui-même éventuellement une retenue. Il suffit pour cela de monter deux  $\frac{1}{2}$  additionneurs en cascade, comme il est montré à la figure 63b :

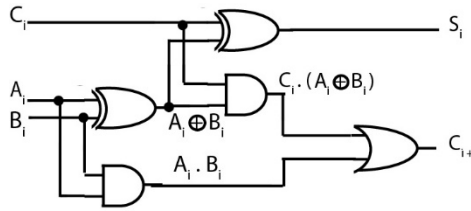


Image 63b

La table de ce montage figure ci-contre.  $C_i$  est l'éventuelle retenue issue d'un additionneur en amont. On voit que sortant du premier ou exclusif,  $A_i \oplus B_i$  est envoyé au second

| $A_i$ | $B_i$ | $C_i$ | $C_{i+1}$ | $S_i$ |
|-------|-------|-------|-----------|-------|
| 0     | 0     | 0     | 0         | 0     |
| 0     | 0     | 1     | 0         | 1     |
| 0     | 1     | 0     | 0         | 1     |
| 0     | 1     | 1     | 1         | 0     |
| 1     | 0     | 0     | 0         | 1     |
| 1     | 0     | 1     | 1         | 0     |
| 1     | 1     | 0     | 1         | 0     |
| 1     | 1     | 1     | 1         | 1     |

ou exclusif en même temps que  $C_i$ . Le premier ET renvoie une éventuelle retenue, comme le second ET qui effectue  $C_i \cdot A_i \oplus B_i$  (retenue éventuelle). La dernière porte, un OU *inclusif*, prendra en  $C_{i+1}$  la valeur 1 (retenue) sauf si  $C_i \cdot A_i \oplus B_i = 0$  et  $A_i \cdot B_i = 0$ . Le mieux, pour bien comprendre ce mécanisme, est de se reporter à la table et au schéma. Si l'on dispose de registres adéquats, un seul additionneur suffirait à la tâche pour additionner par exemple deux nombres de 8 bits (en fait 7, car cette sacrée retenue nous donne des soucis !). On agit alors séquentiellement, comme le montre page suivante (image 64) le dispositif mettant en coalescence un additionneur complet, une bascule DQ et une horloge :

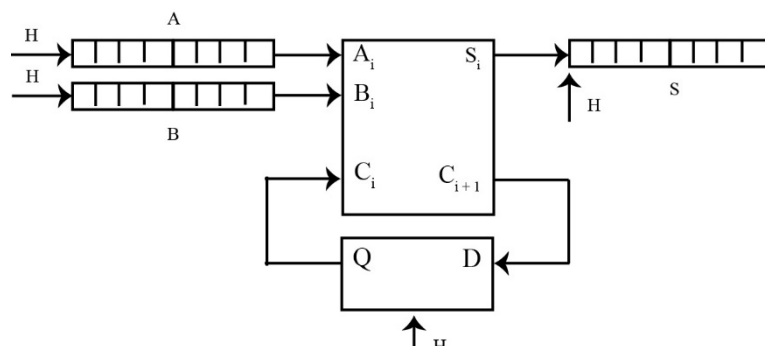


Image 64 Additionneur sériel

Les deux registres A et B sont lus à chaque top d'horloge, bit par bit en commençant par les bits de poids faible (à droite). Chaque bit  $A_t$  et  $B_t$  est transmis en  $A_i$  et  $B_i$ , l'état de la sortie  $S_i$  étant transmis au bit  $S_t$  de même poids. La retenue en  $C_{i+1}$  est transmise à la bascule en D et renvoyée

en  $C_i$  par la sortie Q de la bascule. Cependant ce traitement séquentiel est assez long, aussi préfère-t-on un traitement parallèle, soit un additionneur par bits de même poids, par exemple  $A_0$  avec  $B_0$ ,  $A_1$  avec  $B_1$  etc...

Et tant que nous y sommes : quid de la soustraction ? Car nos algorithmes de la transformée, il y a des soustractions, par exemple :  $TR := PR[ip]^* ur - PI[ip]^* ui$ . En fait il est électroniquement plus facile de procéder en additionnant un nombre A avec l'opposé du nombre B, soit -B. La méthode est assez simple à mettre en œuvre ; sur le papier, on fait le *complément à deux* de B en prenant chaque bit de B et en l'inversant : ainsi  $33_{(10)}$ <sup>15</sup>, soit  $100001_{(2)}$ , devient  $011110$  ; on ajoute 1 et -B devient  $011111$ . On peut maintenant l'additionner avec A, par exemple  $A=65_{(10)}$  soit  $1000001_{(2)}$ . On

<sup>15</sup> L'indice entre parenthèses indique la base de numération :  $33_{(10)}$  est 33 en base dix.

obtient 1100000 ; en éliminant le 1 de poids fort, on trouve 100000 soit  $32_{(10)}$ . Au fond, un soustracteur est un additionneur, mais avec un inverseur sur son entrée B et un forçage de la retenue à 1 pour le seul bit zéro. Dans le schéma 65, si on considère le bit  $B_0$ , on l'inverse (c'est pour cela qu'on insère une porte inverseuse NON) et on force l'entrée  $C_n$  à 1. Cette retenue se répercutera sur  $B_0$  et éventuellement sur les autres bits inversés, mais cette fois par le seul effet du résultat des additions. Aux inverseurs près, le schéma 65 ressemble au schéma 63(b).

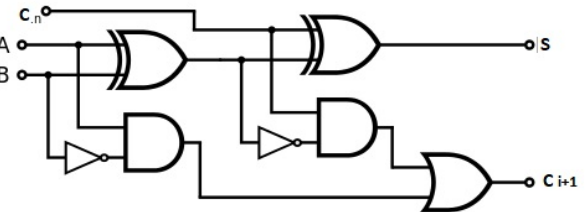


image 65 : soustracteur

Revenons à la multiplication, dont la table est vraiment facile à retenir. Nous savons comment additionner les résultats partiels avec l'additionneur. Maintenant, on procède exactement comme en base dix : prendre le bit  $B_0$  du multiplicateur et faire, pour chaque bit du multiplicande,  $B_0 \cdot A_0$ ,  $B_1 \cdot A_1$ , ...,  $B_n \cdot A_n$  et placer chaque résultat dans un registre  $R_0$ ,  $R_1$ , ... ,  $R_n$ . Il faut tout de même décaler les registres d'un rang vers la gauche dès  $R_1$ , comme on fait sur le papier. Et on renvoie tout cela aux additionneurs. On comprendra que  $B_n \cdot A_n$  renvoie tout simplement à une porte ET. On utilise habituellement un calcul parallèle, avec autant de portes ET qu'il y a de bits à traiter. Je donne figure 66 un extrait d'un schéma de multiplicateur (pris je ne sais plus où !). En raison de sa complexité, on comprendra que je ne souhaitais pas m'amuser à le dessiner 😊.

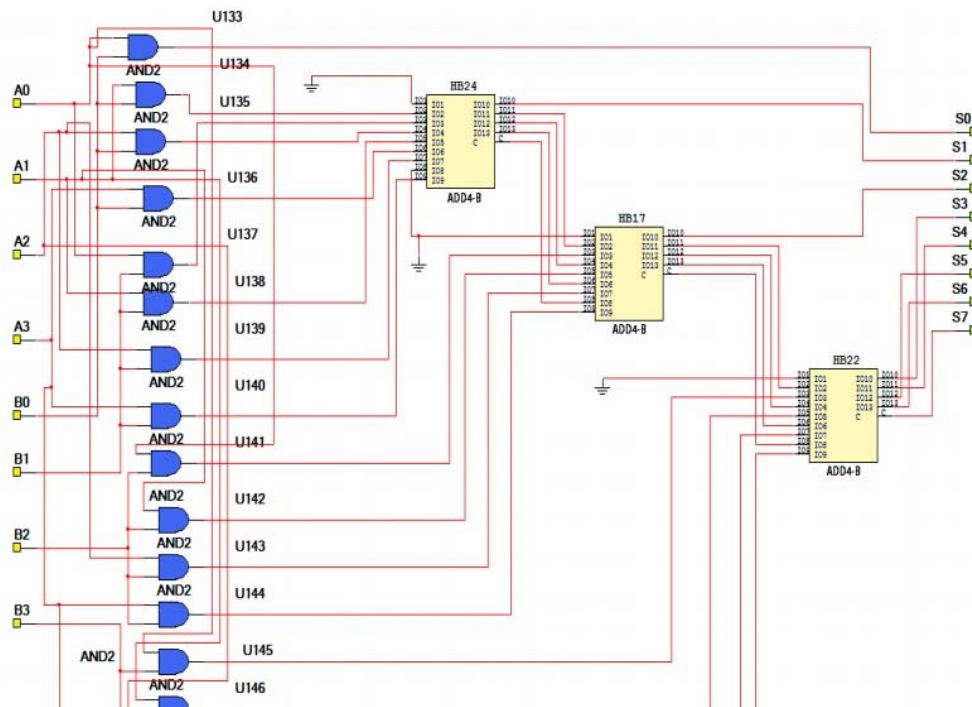


Image 66 : le multiplicateur



Bon, on imagine assez facilement que la division n'est rien d'autre qu'une suite de soustractions du diviseur au dividende selon une réitération  $\text{Dividende} = \text{Dividende} - \text{Diviseur}$ .

Je n'ai envisagé, pour simplifier l'exposé, que des entiers non signés. Le signe est toujours le bit de poids le plus fort (bit 8 pour un nombre sur un octet). Quant aux parties fractionnaires, après la virgule, elles sont traitées comme des parties entières, avec éventuellement une retenue vers la partie entière. Je simplifie à outrance, bien sûr.

### 3.d. Remarques.

1-En tous cas, nous avons tous les dispositifs permettant la transformée : temporisateurs, multiplieurs, additionneurs, soustracteurs. Toutes les pièces du Meccano, en somme. On trouve ainsi des analyseurs de Fourier « hardware » sur beaucoup d'appareils de traitement du signal. De même, sur nos ordinateurs, l'Unité Arithmétique et Logique dispose de sommateurs, multiplieurs etc... Cette plongée dans les fondamentaux de la technique numérique nous révèle cependant les limites pratiques du calcul numérique. Sans doute les algèbres de Boole permettent-ils de dériver à l'infini des théorèmes, mais, dans la production calculée, on se heurte aux réalités contraignantes de la technique. On rencontre constamment le conflit entre la vitesse d'exécution et l'architecture exécutante : on peut ainsi n'utiliser séquentiellement qu'un seul additionneur, en travaillant séquentiellement *en série*, mais le travail prend du temps. On peut aussi utiliser *en parallèle* plusieurs additionneurs ou plusieurs multiplicateurs, mais cela demande un grand nombre de composants. Je pense que ce problème se pose en fait dans toute activité industrielle (ce qui nous force à l'*empirie*), et notamment lorsque l'*efficacité* (je n'ai pas d'autre mot pour cela) ou le *rendement* (idem) de l'objet ouvré entre en conflit avec, au plan IV, la valeur d'*ophélimité* ou, plus souvent avec le sacrifice à consentir pour l'obtenir. Revenons aux limites techniques. J'ai à maintes reprises insisté sur la perte d'information ; nous travaillons le plus souvent sur des nombres appartenant à l'ensemble  $\mathbb{R}$  des réels signés, mais on comprend que l'on se heurte assez rapidement à la taille des registres (par exemple le type Extended, en Pascal, nécessite dix octets avec une précision de 19 à 20 chiffres). Il faut reconnaître aussi que la numération binaire consomme énormément de mémoire : la limite technique impose donc des arrondis.

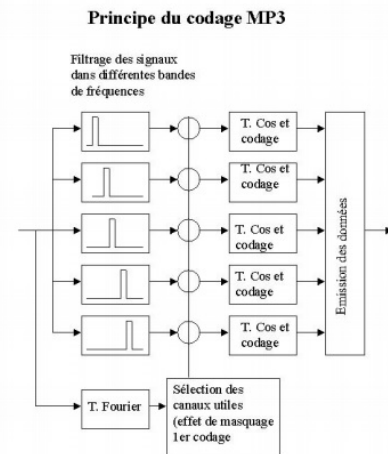
2-Je parlais des pièces du Meccano : c'est vrai que l'on assemble, mais pas n'importe comment. Cela va de soi : la coalescence des machines impose leur complémentarité, et celle-ci subit les contraintes des lois de la Physique. Par exemple, un signal s'atténue en passant par une porte logique, et la sortie n'est pas nécessairement toujours au niveau de tension que l'on attendait : dans ce cas, il faut compenser la perte. Autre exemple : une porte est physiquement limitée dans le nombre de portes du même type (TTL ou CMOS) qu'on peut connecter à sa sortie :

on appelle cela sa *sortance*<sup>16</sup>. *Au fond, je me risquerais bien à dire que l'analyse technique, plutôt que de résoudre des problèmes qui lui préexisteraient, en fait exister d'autres par sa nature même.*

3-Nous sommes arrivés à un niveau d'analyse nous permettant de comprendre ce sur quoi se fonde la « technologie » numérique : des tensions électriques correspondant à des seuils permettant de distinguer un état « bas » d'un état « haut »<sup>17</sup>, tensions obtenues grâce à un arrangement particulier de composants de base comme des transistors, des résistances etc... Telle est la réalité technique. « 0 » et « 1 » ne sont en fin de compte que des descriptions de l'art (théorétiques) que l'on remplace parfois par « L » (low) et « H » (high) ou « F » (faux) et « V » (vrai).

4-Revenons à cette transformée de Fourier, sur laquelle j'insiste longuement moins en raison de son importance en traitement du signal que du fait qu'elle constitue un biais excellent pour approcher les éléments fondamentaux des techniques numériques. Comme je l'ai dit, la TF n'est pas une fin en soi, elle ne vaut que par ses applications. J'ai évoqué au passage quelques emplois de la transformée, dont l'analyse fréquentielle des signaux -et ce qui en découle : l'identification d'un système linéaire, la suppression d'échos, de bruits etc.... En voici d'autres : analyse, synthèse et reconnaissance de la « parole<sup>18</sup> », transmissions analogiques et numériques, radio et télévision numériques, interférométrie et spectroscopie, RMN, cristallographie et, bien entendu, traitement d'image.

Pour coder des fichiers MP3 (son) comme indiqué ci-contre, la TFR intervient avec une autre transformée dont nous avons déjà parlé, la transformée cosinus. Mais puisque nous avons, au cours de ces causeries, beaucoup insisté sur l'image, il faut signaler que la transformée de Fourier s'y applique bien.



## 4. Transformée de Fourier et images.

Mathématiquement, nous avons affaire avec l'image à un espace à deux dimensions,  $f(x,y)$  ; l'étude des fréquences se fera par une double transformée de Fourier, selon une double intégration  $F(\mu, \nu) = \iint_{-\infty}^{+\infty} f(x, y) e^{-j2\pi(\mu x + \nu y)} dx dy$

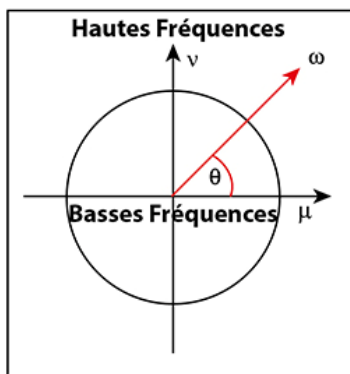
<sup>16</sup> « La sortance s'exprime par un nombre entier et représente la capacité que possède la sortie d'un circuit à piloter en même temps plusieurs entrées de circuits de même type. » ([https://www.sonelec-musique.com/electronique\\_bases\\_compatibilite\\_ttl\\_cmos.html](https://www.sonelec-musique.com/electronique_bases_compatibilite_ttl_cmos.html))

<sup>17</sup> <en logique CMOS, l'état bas, 0 est obtenu avec une tension inférieure à 1/3 de la tension d'alimentation de la porte considérée, l'état « haut » est obtenu par une tension supérieure à 2/3 de la tension d'alimentation.

<sup>18</sup> Il ne s'agit évidemment pas de langage, mais simplement de phonation à partir de textes, ou encore de « dictée » vocale.

selon la direction x (colonnes) avec  $\mu$  pour la fréquence spatiale et la direction y (lignes) avec  $\nu$  pour la fréquence spatiale correspondante. Malgré l'aspect rébarbatif de l'intégrale double, le calcul est assez simple : nous avons un tableau de nombres réels (appartenant à  $\mathbb{R}$ ), aussi pouvons-nous travailler sans trop nous inquiéter de  $j$  et des nombres complexes -nous l'avons déjà fait pour la TFR d'un signal. Chaque case du tableau représente l'intensité lumineuse de l'image en un point  $\{x,y\}$  selon la fonction  $f(x,y)$ . L'image ayant un nombre fini de points, les bornes des sommations seront ramenées à M et N :  $f(\mu, \nu) = \sum_{m=0}^M \sum_{n=0}^N f(m, n) e^{-j(\frac{2\pi}{m})\mu m} e^{-j(\frac{2\pi}{n})\nu n}$ . Hum... Bon...

La TFR nous permet de calculer le spectre de module et celui de phase ; le spectre de module décrit la distribution énergétique de l'image, en respectant aussi bien la périodicité que l'orientation des motifs de l'image. En fait,



l'important est de pouvoir interpréter la transformation en termes de hautes et de basses fréquences. Le repérage se fait selon le plan de Fourier (voir ci-contre) : les basses fréquences sont situées au centre, les hautes fréquences sont en périphérie. Les basses fréquences correspondent à des surfaces sans changement important d'intensité lumineuse, les hautes fréquences correspondent à de brusques changements d'intensité lumineuse, c'est notamment le cas des contours. L'image 67 montre les

variations du spectre de module selon le traitement d'une texture : les spectres (images du bas) varient d'une image à l'autre, mais présentent des caractéristiques communes, par exemple la présence de segments de droites passant par l'origine, d'épaisseur plus ou moins importante. Plus une droite est longue, plus elle porte des fréquences élevées. Quant à la direction des droites, elle indique les lignes de force de l'image d'origine, qui lui sont perpendiculaires. Enfin, si une

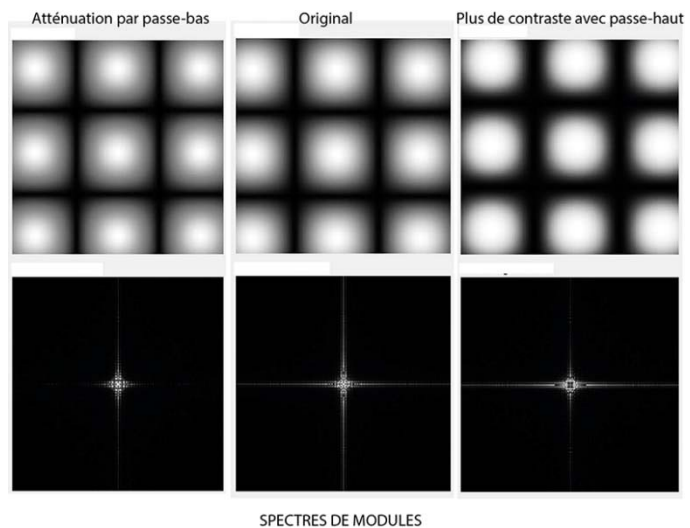


Image 67

droite est faite de points ou de stries alignées, cela indique une certaine périodicité de l'image dans des directions perpendiculaires. Si avec un filtre passe-bas on élimine l'amplitude des hautes fréquences, on obtient une apparence de flou -il y a moins de contrastes, cf l'image de gauche) ; si, au contraire, à l'aide d'un filtre passe-haut on augmente l'amplitude des hautes fréquences, accentue les contours, mais l'image paraît plus « bruitée ». Les mêmes observations peuvent être faites à partir des images ordinaires :

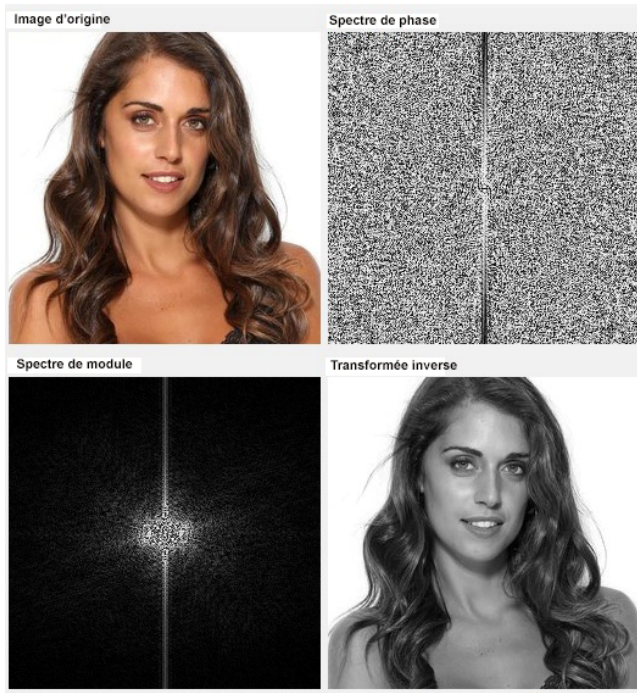


Image 68

Le spectre de phase est assez difficile à interpréter à cette échelle, mais le spectre de module comporte d'une part une concentration des basses fréquences autour de l'origine du plan de Fourier, d'autre part une forte concentration des hautes fréquences le long de la ligne verticale, ce qui correspond bien à une image ayant des contours très nets. La TF inverse apparaît en noir et blanc, car le logiciel dont je me sers travaille avec des niveaux de gris. Cela n'enlève rien à la luminance.

## 5. Le filtrage.

Pour finir l'exposé technique, nous avons beaucoup évoqué les filtres. J'aimerais donner une idée sur la manière dont peut fonctionner un filtre numérique. L'image 69 schématise le dispositif électronique capable d'effectuer un filtrage :

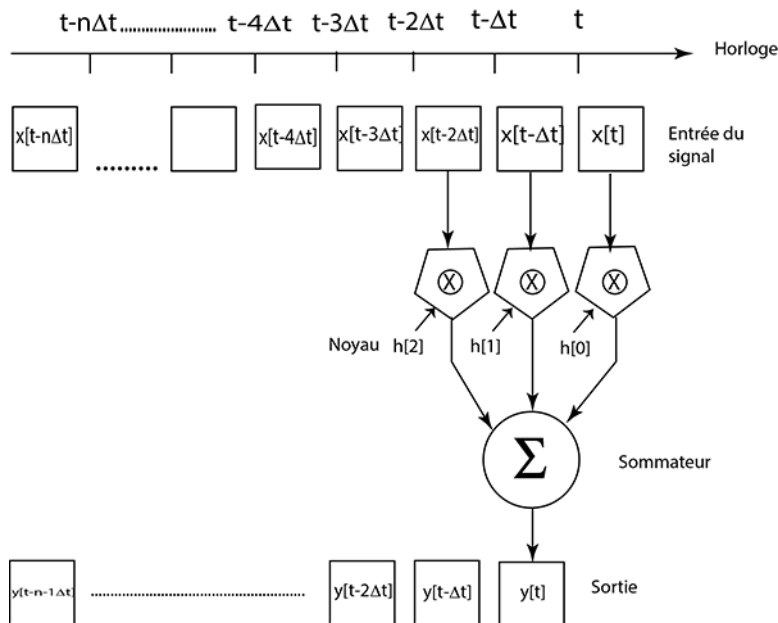


Image 69 : schéma d'un filtre

Cette image n'est pas sans rappeler l'image 59. En fait, on y retrouve les multiplieurs et le sommateur, je n'y reviendrai donc pas. On distingue en fait trois registres : celui contenant les valeurs du signal (les  $x$  correspondant aux valeurs échantillonnées de la fonction  $f(t)$ ), le noyau contenant les coefficients  $h$ , et le registre de sortie contenant les valeurs  $y$  de la fonction filtrée. On remarquera que les opérations s'effectuent sur trois valeurs successives

du signal entrant, et que par conséquent la sortie  $y[t-n\Delta t]$  sera en retard sur l'entrée. ce qui se traduit par un déphasage entre le signal d'entrée et celui de sortie.

Le noyau comporte ici trois termes,  $h_0, h_1, h_2$ . On effectue donc les opérations sur trois échantillons du signal ; c'est une combinaison linéaire. Si  $M$  est le nombre de termes du noyau, l'échantillon  $n$  de sortie est calculé sur les  $M$  derniers échantillons de l'entrée, soit  $y_{[n]} = \sum_{i=0}^{M-1} h_{[i]} * x_{[n-i]}$ . Cette opération, nommée *convolution discrète*, permet d'effectuer un filtrage par convolution. La formule nous montre l'utilité des multiplicateurs et du sommateur. J'ai indiqué ici les opérations effectuées par un filtre en *temps-réel*, c'est-à-dire simultanément à l'entrée du signal numérisé dans l'appareil, mais il est évidemment possible de travailler avec un tableau de valeurs correspondant à un échantillonnage borné du signal.

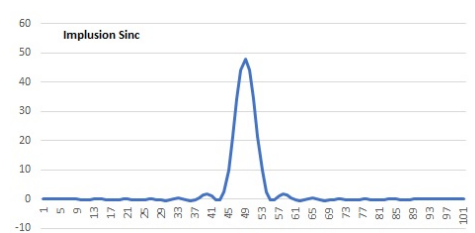
Je passe sur beaucoup de détails, évidemment, pour ne pas compliquer un exposé déjà trop mathématisé. Il suffit de mentionner qu'un filtre répond à un type d'impulsion (par exemple impulsion de Dirac, sinus cardinal etc..) et qu'il est défini par sa fréquence de coupure, c'est-à-dire les fréquences qu'il bloque et qu'il élimine du signal entrant (ceci dit grossièrement). Pour donner un exemple, revenons au signal de la figure 50, composé de trois harmoniques, celles-ci étant trois fonctions sinusoïdales  $f_1\{\text{Fréq}_1 = 2 \text{ Hz}, \text{Ampl}_1 = 4\text{V}\}, f_2\{\text{Fréq}_2 = 4 \text{ Hz}, \text{Ampl}_2 = 6\text{V}\}, f_3\{\text{Fréq}_3 = 6 \text{ Hz}, \text{Ampl}_3 = 2\text{V}\}$ . Supposons que je veuille éliminer la composante de haute fréquence (celle de  $f_3$ ). Il me faut un filtre dont la valeur de  $h$  à la fréquence de coupure soit  $h(fc) = \frac{1}{\sqrt{2}} * H_{Max}$ ,  $H_{max}$  étant l'amplitude maximale du signal traité. En se rapportant à la figure 50, on voit que l'amplitude maximale est de 10, donc  $h_{(fc)} = 7,07106...$  et des poussières. Je dois également connaître une grandeur  $a = fc/fe$  ( $fe =$  fréquence d'échantillonnage) ; puisque j'ai travaillé sur 1024 échantillonnages par seconde,  $fe = 1/1024$ . Ceci étant connu, je réalise un filtre passe-bas (qui, donc, laisse passer toutes les fréquences sauf celles supérieures la fréquence de coupure) répondant à une impulsion de type sinus cardinal

$sinc(x) = \frac{\sin(x)}{x}$  (voir l'impulsion ci-contre).

Pour chaque multiplicateur  $k [0..2]$  du noyau, le signal de sortie sera multiplié par :  $2a \frac{\sin(2\pi ka)}{2\pi ka}$ .

L'image 70, page suivante, indique le résultat : la

courbe bleue représente le signal d'origine, la courbe rouge représente le signal filtré par convolution. Le signal n'est plus composé que du fondamental et du premier harmonique (l'amplitude max, 4, est la sienne).



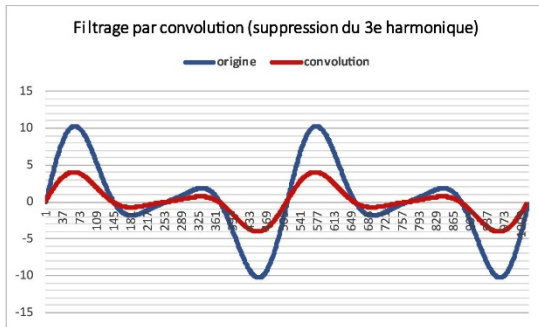
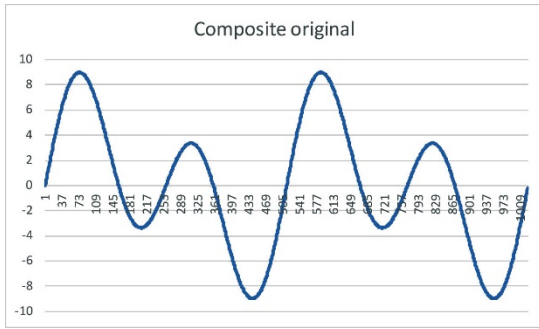


Image 70

Le principe est évidemment très semblable lorsqu'il s'agit d'une image. Comme on le sait, les pixels sont les échantillons. Le filtrage par convolution se fait selon les trois niveaux de gris, chacun correspondant à un niveau de couleur (voir seconde causerie). On opérera donc une convolution sur chaque couche, séparément.

Conventionnellement, on opère en {colonne, ligne}, ce qui correspond aux coordonnées (x,y) du pixel, et la réponse en impulsion du filtre est aussi un signal en deux dimensions, dont on peut noter les coefficients par  $h(m,n)$ . Si on note par  $X_{ij}$  la valeur d'un pixel (i,j), on obtiendra son image  $Y_{ij}$  par convolution, en combinant linéairement le pixel de valeur  $X_{ij}$  avec les

huit pixels de son proche voisinage (nous avons déjà vu une combinaison semblable de pixels lors de la transformée cosinus des images JPEG, cf seconde causerie). On voit une difficulté : les pixels des bords ne peuvent pas être traités de cette manière, puisque par exemple le pixel (0,0) n'a que trois voisins, ou encore, dans un tableau 12 x 12 pixels, le pixel (2,11) n'a que cinq voisins : il faudra un traitement spécial pour ces pixels du bord, le plus simple étant de ne pas les traiter du tout. L'image 71 indique le principe de ce filtrage par le noyau : le noyau est un ensemble de 9 coefficients. Dans l'exemple, il s'applique au pixel  $X_{ij}$  (en notation de tableau informatique, [j,i]). Le coefficient  $h_{11}$  s'applique à ce pixel, les autres coefficients s'appliquent aux huit pixels voisins. Généralement, les valeurs des pixels sont des entiers (de 0 à 255, comme on le sait), cependant le résultat de la convolution est exprimée en réels. Cela permet de limiter les erreurs d'arrondi lorsque l'on effectue plusieurs filtrages successifs. Ces valeurs sont converties en entiers lors de l'affichage ou lors de l'enregistrement de l'image.

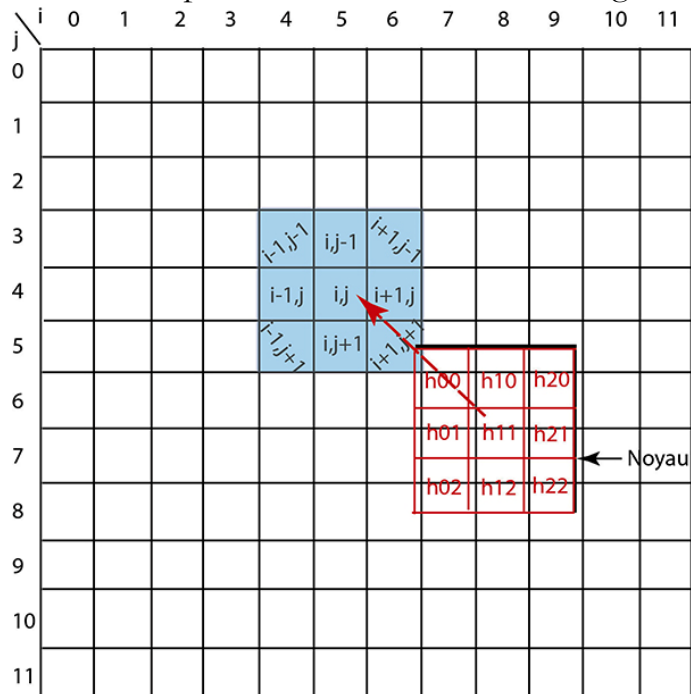

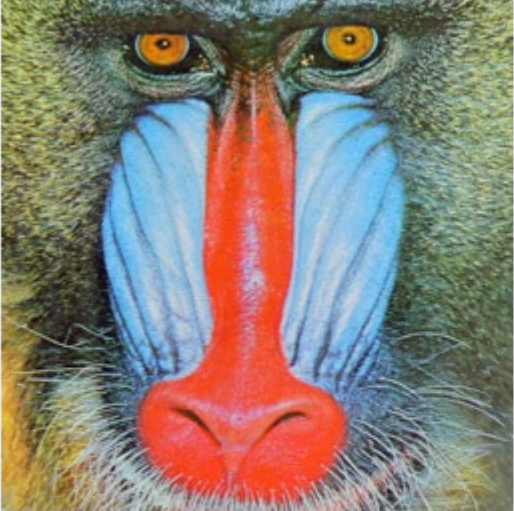


Image 71 Filtrage 2D par convolution

Pour terminer avec l'image, voici le résultat d'un filtrage par un filtre dit de *Prewitt Y*:

**Filtrage d'une image par convolution**

<http://www.f-legrand.fr/scidoc/simul/image/filtrageConvolution.html>



**Image**

Photo babouin ▾

**Filtre**

Prewitt Y (valeur absolue) ▾

|         |         |         |
|---------|---------|---------|
| -1.0000 | -1.0000 | -1.0000 |
| 0.0000  | 0.0000  | 0.0000  |
| 1.0000  | 1.0000  | 1.0000  |

Les coefficients du noyau sont indiqués dans le tableau. On constate une nette atténuation des basses fréquences, mais une amplification des hautes fréquences permettant de discerner les contours.

## 6. La commande numérique.

Avant d'en finir, j'aimerais ouvrir pas même une fenêtre, au plus un cernel, sur un domaine immense, celui de la commande numérique des machines. En effet, tous les exemples pris jusqu'ici concernent sans doute le physicien, mais surtout le grand public avec l'imagerie, le MP3, les iPhones et autres gadgets. Mais il ne s'agit que d'un petit domaine à côté de celui de l'automatique numérique qui fait fonctionner les machines à expresso, guide les fusées, commande les robots etc... Prenons juste un exemple. L'image 72, page suivante, montre comment un dispositif à cinq portes logiques (deux ET et trois inverseurs NON) peut piloter un processus industriels.

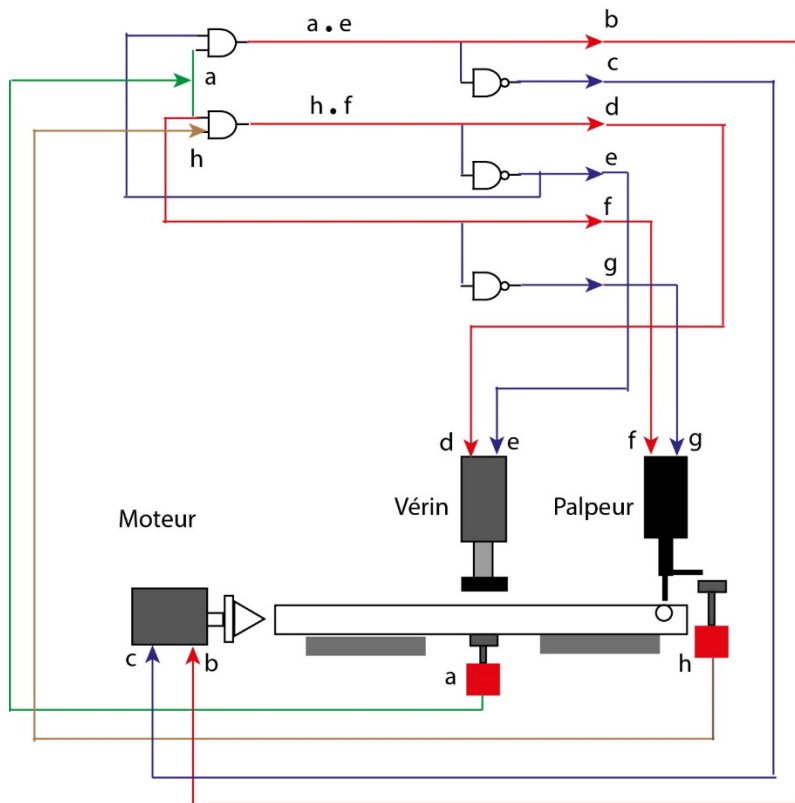


Image 72 : automatisation numérique

Il s'agit d'une situation que j'ai « modernisée » à partir de l'étude de dispositifs pneumatiques - car on peut aussi fabriquer des portes logiques pneumatiques ou hydrauliques, demandez aux mécaniciens !. On veut placer automatiquement un tube dans une position précise pour lui souder un embout. Le tube est percé d'un trou perpendiculaire à l'axe à une extrémité. Un autre dispositif place le tube sur le support et le capteur **a** détecte la présence de ce tube. **b** est alors à l'état 1 et le moteur s'avance, enclanche le galet

cônique dans le tube et fait doucement tourner le tube. Mais cela n'est possible que si le vérin est relevé, c'est-à-dire si **e** commandant son relevage est à 1. La rotation n'a d'intérêt que si le palpeur est descendu, afin de positionner l'ergot dans le trou du tube : **f** doit être à 1. La rotation s'arrête quand l'ergot est engagé, ce qui met **h**, un autre capteur, à l'état 1. Alors le vérin descend (**d** est à l'état 1) pour bloquer le tube en bonne position, le moteur recule (**c=1**) et s'arrête, le palpeur peut remonter (**g** passe à l'état 1). La soudage peut commencer après qu'un autre dispositif ait placé l'embout.

On remarque que finalement l'état du système dépend de celui des deux capteurs **a** et **h**. On remarque également que (b,c), (d,e), (f,g) sont antagonistes, ce qu'indique la présence d'inverseurs. Mais d'autres portes se complètent : par exemple, **b** est à l'état 1 si **a** et **e** sont à l'état 1, de même **d** est à l'état 1 si **h** et **f** sont à l'état 1 : de là le rôle des portes AND (ET).

La table des états est indiquée ci-contre : **b** commande l'avance du moteur, **c** son retrait. **d** fait descendre le vérin, **e** le fait remonter ; **f** fait descendre le palpeur, **g** le fait remonter.

| a | h | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Naturellement, la tension de sortie des portes n'est pas directement appliquée aux dispositifs mobiles comme le vérin, le palpeur : elle ne suffirait pas à les faire fonctionner. Mais ils commandent l'électronique de puissance, par exemple des triacs



qui commutent les circuits d'alimentation et fournissent la puissance nécessaire. Tout comme dans la machine à laver ou le four de la cuisine, à notre époque !

J'aurais bien voulu en dire davantage sur cette électronique numérique passionnante qu'est la commande des systèmes, mais le temps passe... et je dois rendre ma copie.

## Conclusion

Ainsi s'achève cette promenade au jardin numérique. Ce n'est qu'une vue échantillonnaire d'un domaine gigantesque. En relisant cette dernière causerie, je trouve que, par souci de simplification, j'ai été beaucoup elliptique dans l'explication et parfois cependant trop ésotérique -chacun de ces défauts n'excluant pas l'autre. Voilà qui risque de ne satisfaire ni le mathématicien, ni l'ingénieur, ni l'ergologue ! En alternant les approches mathématiques (quelquefois hâtives) et les approches électroniques (tout aussi hâtives), mon souci était de mettre en vis-à-vis les *modélisations* et les *fabrications*. Il me semble évident que les techniques numériques font appel à de la production calculée, mais j'espère avoir fait comprendre que l'on a affaire à deux modes rationnels propres, une analyse logique et une analyse technique, même s'ils interfèrent.

Nous avons également, au détour des exemples, rencontré des dispositifs vraiment polytropes comme les AOP, les transistors etc., et constaté que l'analyse technique pouvait démarrer à n'importe quel niveau : qu'est-ce qui fait de l'*engin* ? une résistance, une inductance, ou à un autre niveau la coalescence de deux diodes et d'une résistance, ou de deux transistors, voire quatre, dans une « porte logique » qui devient à son tour une brique du jeu de Lego ? Je crois que le niveau de départ tient surtout à la disponibilité industrielle de sous-ensembles qui évite d'avoir à chaque instant à réinventer la roue. On ne réinvente pas le marteau quand on veut enfoncer un clou : il est là, virtuellement et pratiquement disponible, comme la porte NAND ou le microprocesseur.

Au fond, il n'y aurait pas de « monde numérique » sans la capacité d'analyse technique et son réinvestissement industriel : certes, pour nous c'est un truisme, mais sans les portes logiques, l'algèbre de Boole ne demeurerait qu'une algèbre. Non, loin de là, qu'il s'agisse de la réalisation pratique d'une algèbre : c'est tout autre chose, même si *techno-logiquement parlant* on a l'impression que le résultat est le même.

J'ai donné quelques aperçus, sans vraiment baliser de pistes de recherche. J'espère néanmoins que les ergologues trouveront dans ce travail matière à réflexion.

BC.

