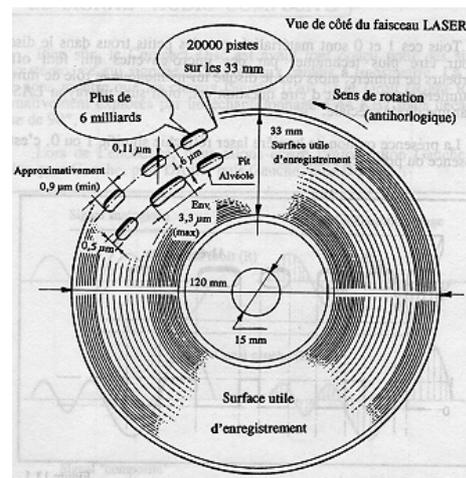
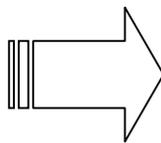
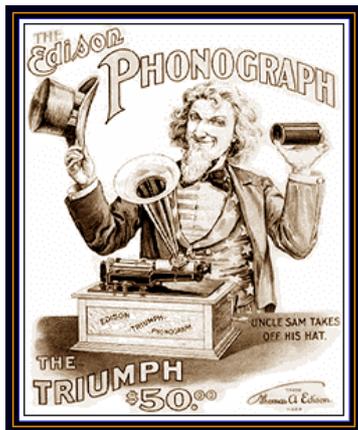


# CHAPITRE VII

## Les Convertisseurs Analogiques Numériques



## SOMMAIRE

<b>LES CONVERTISSEURS À INTÉGRATION .....</b>	<b>3</b>
<b>I LE CONVERTISSEUR SIMPLE RAMPE .....</b>	<b>3</b>
I.1 PRINCIPE .....	3
I.2 PHASES DE FONCTIONNEMENT .....	3
I.3 CARACTÉRISTIQUES .....	4
<b>II LE CONVERTISSEUR À RAMPE NUMÉRIQUE.....</b>	<b>4</b>
II.1 PRINCIPE.....	4
<b>III LE CONVERTISSEUR DOUBLE RAMPE (OU PAR INTÉGRATION).....</b>	<b>4</b>
III.1 PRINCIPE.....	4
III.2 PHASES DE FONCTIONNEMENT.....	5
<b>IV LE CONVERTISSEUR PAR PESÉES (APPROXIMATIONS) SUCCESSIVES .....</b>	<b>5</b>
C'EST UNE VIEILLE APPROCHE QUI EST EN PHASE D'ÊTRE REMPLACÉE PAR LES CONVERTISSEUR PIPELINE.....	5
IV.1 PRINCIPE .....	5
IV.2 MISE EN ŒUVRE .....	6
<b>V LE CONVERTISSEUR FLASH (OU PAR COMPARAISON DIRECTE).....</b>	<b>7</b>
V.1 PRINCIPE.....	7
<b>VI LE CONVERTISSEUR SEMI-FLASH .....</b>	<b>8</b>
VI.1 PRINCIPE .....	8
<b>VII BILAN COMPARATIF RAPIDE DES CAN.....</b>	<b>8</b>
<b>VIII CONVERTISSEUR PIPELINE .....</b>	<b>8</b>
VIII.1 PRINCIPE.....	8
VIII.2 CAS DU PIPELINE N ÉTAGES 1 BITS .....	9
<b>IX CONVERTISSEUR ALGORITHMIQUE.....</b>	<b>9</b>
<b>X CONVERTISSEUR À SURÉCHANTILLONNAGE.....</b>	<b>10</b>
X.1 PRINCIPE.....	10
X.2 INFLUENCE SUR LE BRUIT DE QUANTIFICATION .....	10
X.3 LE MODULATEUR DELTA.....	11
X.4 STRUCTURE DELTA SIGMA .....	12
<b>XI BILAN DES CONVERTISSEURS (1997).....</b>	<b>13</b>

# Les Convertisseurs Analogiques Numériques

Le but du CAN est de convertir un signal analogique continu en un signal discret et cela de manière régulière (à la fréquence d'échantillonnage).

Il existe différents types de convertisseur qui vont se différencier par leur temps de conversion et leur coût (Surface de silicium).

$$V \xrightarrow{\text{can}} N = \sum_0^{n-1} b_i \cdot 2^i$$

## Les convertisseurs à intégration

### I Le convertisseur simple rampe

#### I.1 Principe

A la valeur de la tension d'entrée on fait correspondre une impulsion dont la largeur est proportionnelle à cette tension. Cette impulsion vient contrôler l'autorisation à s'incrémenter d'un compteur. On génère ainsi le code binaire de sortie en comptant plus ou moins longtemps en fonction de l'amplitude du signal à convertir.

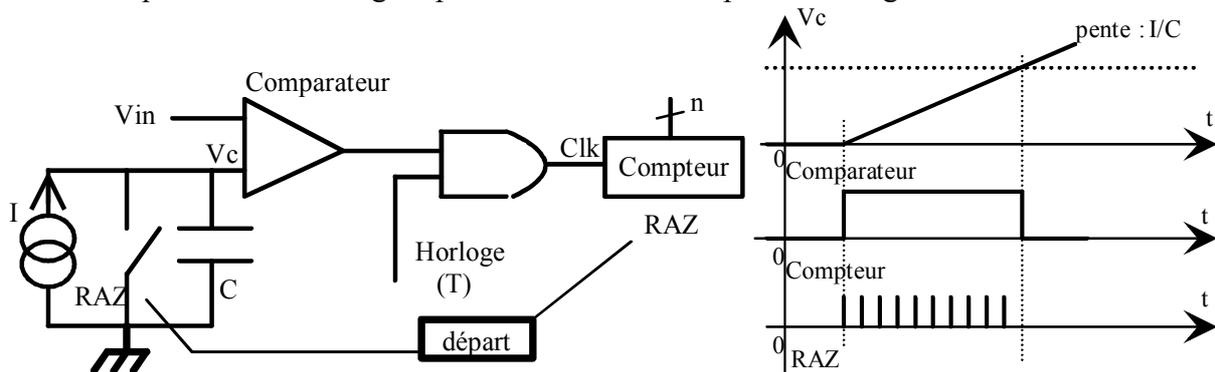


Figure 1 : Principe du convertisseur simple rampe

#### I.2 Phases de fonctionnement

Phase 1 : RAZ  $V_c=0$  et  $N=0$

Phase 2 : Intégration aux bornes du condensateur sous un courant constant tant que la tension du condensateur  $V_c$  est inférieure à la tension à convertir  $V_{in}$ .

$$V_c = \frac{1}{C} \int I dt = \frac{I}{C} t$$

$V_c < V_{in} \Rightarrow$  on compte tous les  $T$ ,  $T$  période de l'horloge système.

Phase 3 :  $V_c=V_{in}$

Le comparateur bascule et bloque le compteur à sa dernière valeur  $N$  :

$$N = \frac{C V_{in}}{I T}$$

On obtient une valeur comptée  $N$  qui est fonction de  $V_{in}$ ,  $C$ ,  $I$  et  $T$ .

### I.3 Caractéristiques

+ Avantages :

- Simple et peu coûteux.

- Inconvénients :

- $N$  dépend de  $C$  donc de la tolérance sur  $C$ .
- Lent car nécessite  $2^N$  cycles d'horloges pour effectuer une conversion.
- Comme il n'y a pas de synchronisme entre l'horloge et le RAZ, cela induit une imprécision de 1 période au début et à la fin de la conversion soit une erreur moyenne de 1,5 quantum.

## II Le convertisseur à rampe numérique

### II.1 Principe

Dans ce cas, on remplace l'intégrateur analogique par un convertisseur N/A :

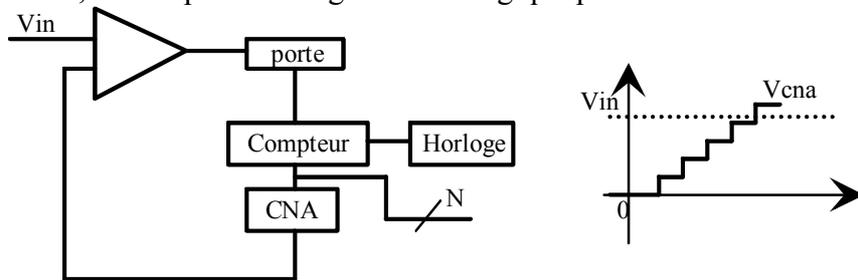


Figure 2 : Principe du convertisseur à simple rampe

La rampe est ainsi réalisée de manière numérique. Le temps n'intervient plus comme variable.

## III Le convertisseur double rampe (ou par intégration)

### III.1 principe

On effectue une double intégration de manière à faire s'annuler les erreurs dues aux composants :

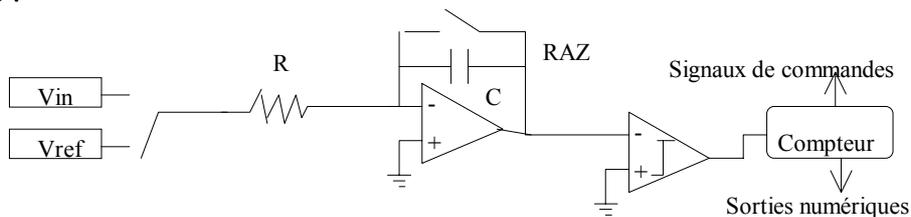
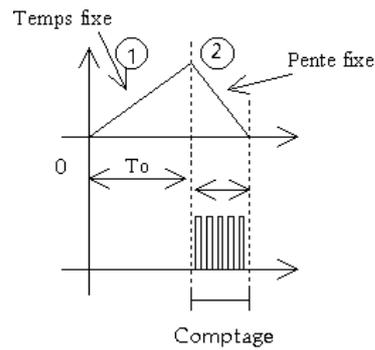


Figure 3 : Architecture du convertisseur double rampe

**III.2 Phases de fonctionnement**

Phase 1: On charge une capacité pendant un temps  $T_0$ , fixé, sous la tension à mesurer.  $T_0$  représente un cycle complet du compteur.

Phase 2 : On décharge la capacité sous une tension fixée  $V_{ref}$ . Durant cette décharge, on incrémente un compteur ( $n$  bits) qui une fois la décharge terminée, sera l'image numérique de la tension à quantifier.



On notera qu'il faut que  $V_{ref}$  et  $V_{in}$  soit de signe opposé.

La durée de fonctionnement du compteur est alors :  $T = T_0 \frac{V_{in}}{V_{ref}}$

On s'affranchit de l'incertitude sur la capacité. Seule la tension de référence intervient dans la mesure ainsi que le nombre ( $N$ ) d'impulsions  $T_e$  enregistrées durant  $T$ .

On a :  $T_0 = 2^n T_e$  et  $T = N T_e$ , d'où : 
$$N = 2^n \frac{V_{in}}{V_{ref}}$$

Ces convertisseurs offrent une bonne résolution, mais sont très lents. On peut les utiliser avec des cycles de conversion de 20ms de manière à s'affranchir de l'influence du secteur (50Hz). Au USA, on utilise des temps de conversion de 16.6ms du à la fréquence du réseau qui est de 60Hz.

Ils sont utilisés dans le cas de mesure de température, de valeurs quasi-constantes. En instrumentation basse fréquence, on peut atteindre une résolution de 18 bits.

On ne peut espérer des temps de conversion très courts car il nécessite au moins  $2 \cdot 2^N$  cycles d'horloge par acquisition.

**IV Le convertisseur par pesées (approximations) successives**

C'est une vieille approche qui est en phase d'être remplacée par les convertisseur Pipeline.

**IV.1 Principe**

On détermine les valeurs des différents bits l'un après l'autre en commençant par le MSB, un peu à la manière d'un marchand de marché :

$$V_{in} = \frac{V_{ref}}{2} b_{n-1} + \frac{V_{ref}}{4} b_{n-2} + \dots$$

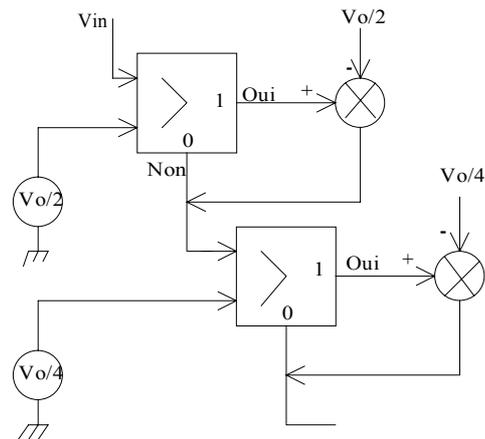


Figure 4 : Principe de la pesée successive

Le signal est comparé à une tension de référence:  $V_o/2$ . S'il est supérieur, on lui retranche cette valeur et on met le bit de comparaison à '1', sinon on met le bit de comparaison à '0' et on le compare à la tension suivante.

$$V_{in} \Leftrightarrow \frac{V_{ref}}{2} \quad \text{puis} \quad V_{in} - \frac{V_{ref}}{2} b_{n-1} \Leftrightarrow \frac{V_{ref}}{4} \quad \text{etc....}$$

On effectue ainsi un encadrement progressif de plus en plus fin.

Ainsi pour un CAN N bits, en N coups on obtient la conversion.

Il est plus lent que le Flash. Ainsi pour 16 bits, il lui faut en moyenne un temps de conversion de 10µs. Il est très adapté à des signaux audio.

#### IV.2 Mise en œuvre

On dispose d'un registre qui à chaque coup d'horloge va décaler le code initial pour arriver au code final :

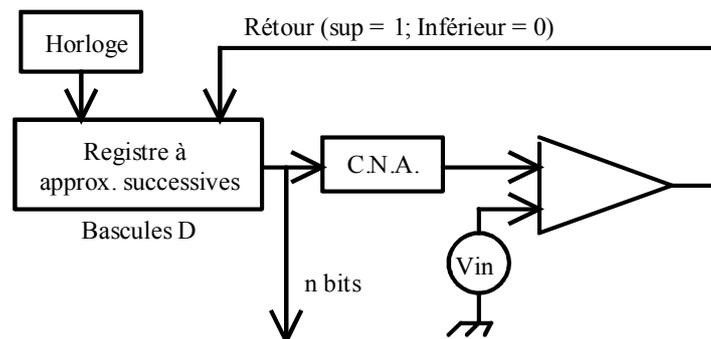


Figure 5 : Elaboration d'un convertisseur à approximations successives

On effectue une comparaison de la tension à convertir  $V_{in}$  avec la tension issue du CNA connecté au registre. Le premier code issu du registre est 1000 ( Cas d'un CAN 4 bits), code correspondant à la tension « moitié » ( $V_{ref}/2$ ). Puis on décale ce code vers  $*100$  puis  $**10$  etc... A la place de «\*», on vient placer le résultat de la comparaison. Si la tension d'entrée est supérieure on positionne un «1», si elle est inférieure on positionne un «0».

Ci-dessous un exemple de cycle de conversion :

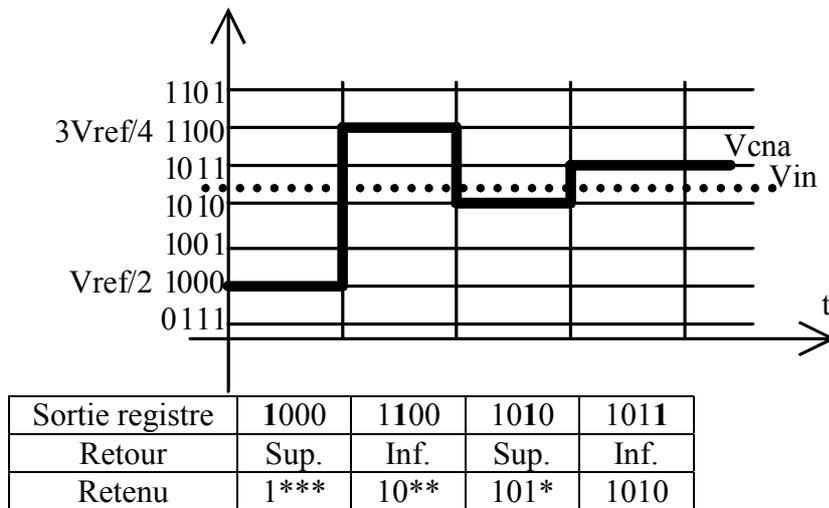


Figure 6 : Evolution du code au cours des pesées successives

## V Le convertisseur Flash (ou par comparaison directe)

### V.1 Principe

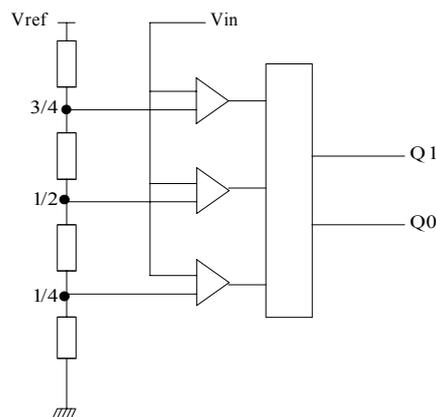


Figure 7 : le convertisseur Flash

C'est un réseau de comparateur mis en parallèle. Un codage sur n bits nécessite  $2^{n-1}$  comparateurs et résistances.

Le type de conversion est lié au choix des valeurs de résistances :

Position Résistance	Quantification linéaire Centrée	Quantification linéaire Par défaut
Connectée Vref	$3R/2$	R
...	R	R
...	R	R
Connectée Masse	$R/2$	R

La conversion est faite en un coup d'horloge, c'est un système qui est très rapide (1- >300Mhz) mais qui coûte très cher. Utilisé en vidéo (30Mhz), il est limité à 12 bits (coût et fabrication de l'encodeur).

Le passage d'un code « thermométrique » au code binaire est très gourmand en terme de surface de silicium. ( $2^{n-1}$  comparateurs) et consomme de la puissance.

8 bits	400 Mhz	2.7 W	6*8 mm <sup>2</sup>
6 bits	6 Ghz	2 W	3*4 mm <sup>2</sup>

## VI Le convertisseur semi-flash

### VI.1 Principe

La conversion se fait en deux étapes :

- Phase 1 : on utilise un premier CAN Flash qui détermine les principaux bits de poids fort.
- Phase 2 : on soustrait la tension des bits de poids fort à la tension d'entrée pour ensuite déterminer les bits de poids faible.

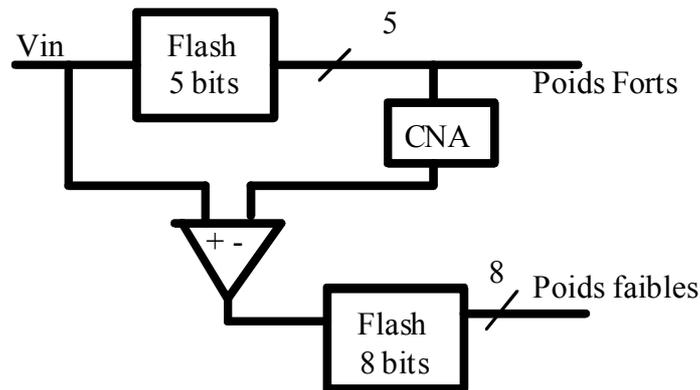


Figure 8 : le convertisseur semi-Flash

On utilise un premier CAN sur  $N_1$  bits pour déterminer le MSB. Puis un CNA sur  $N_1$  bits de manière à retrancher la partie entière, le résidu est converti à l'aide d'un CAN sur  $N_2$  bits.

Ainsi en deux coups d'horloge on effectue la conversion. L'avantage réside en la diminution de la surface de Silicium nécessaire comparé à un CAN Flash. On appelle aussi ce type de Convertisseurs des convertisseurs série parallèle. On peut ainsi augmenter la résolution en diminuant la surface de la puce comparativement au Flash, mais on augmentera le temps de conversion. On trouve des CAN sur ce principe en 8 bits à 40Mhz.

## VII Bilan comparatif rapide des CAN

	Durée de CV	Fréq. Utilisation	Nbre de bits	Coût
Double pente	$2^N$ cycles	kHz	> 16 bits	\$
Approximations	N cycles	50 khz	16 bits	\$\$
Flash	1 cycles	> 10 Mhz	10 - 12 bits	\$\$\$

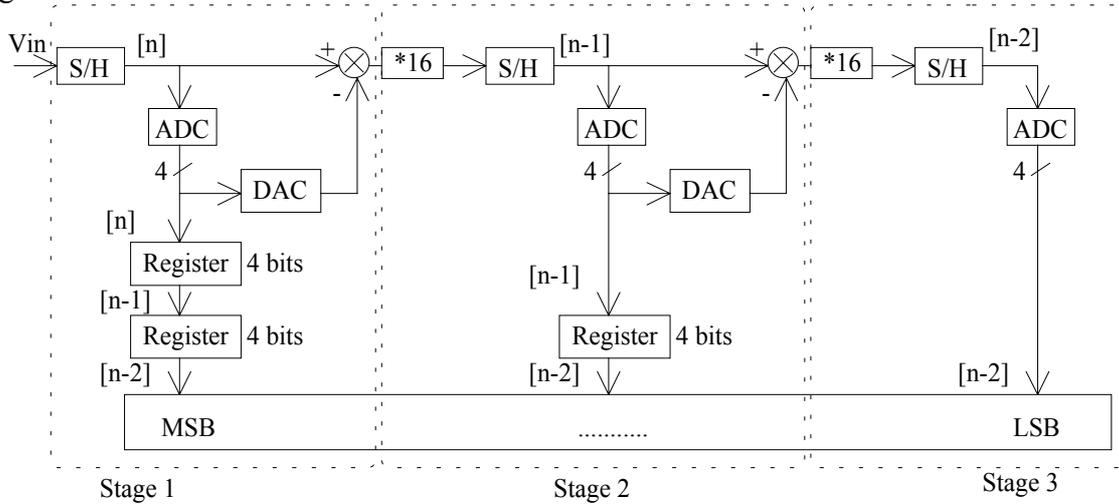
## VIII Convertisseur Pipeline

### VIII.1 Principe

C'est un convertisseur qui se comporte comme un semi flash auquel on a ajouté un E/B entre chaque étage.

A chaque coup d'horloge, on fait effectuer  $n$  conversions en parallèle. Chaque conversion étant dédiée à une partie du code. En traversant le convertisseur (en  $n$  clocks), la tension d'entrée est convertie en commençant par les bits de poids forts et finissant par les bits de poids faibles.

Nous prendrons ci-dessous le cas d'un convertisseur pipeline 12 bits décomposés en 3 étages de 4 bits chacun :



A chaque front d'horloge, on effectue 3 conversions en parallèle au travers de chaque cellule. Chaque conversion correspond à une partie du code binaire.

- Etage 1 : MSB de l'entrée correspondant à l'instant [n]
- Etage 2 : Bits intermédiaires de l'entrée correspondant à l'instant [n-1]
- Etage 3 : LSB de l'entrée correspondant à l'instant [n-2]

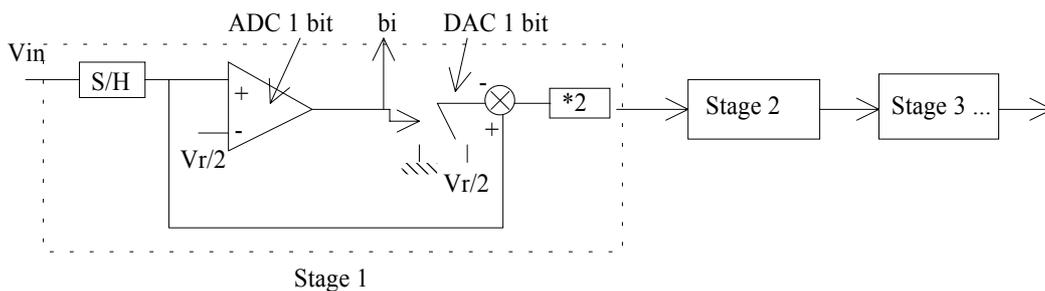
A la fin de chaque étape, on calcule le résidu de la conversion partielle, ce résidu est ensuite recalé à la pleine échelle par une multiplication..

Ce convertisseur possède un temps de latence nécessaire à la propagation de l'entrée dans les cellules (ici 3 coûts d'horloge).

Mais une fois le convertisseur « chargé », à chaque coût d'horloge il sort une data.

**VIII.2 Cas du pipeline N étages 1 bits**

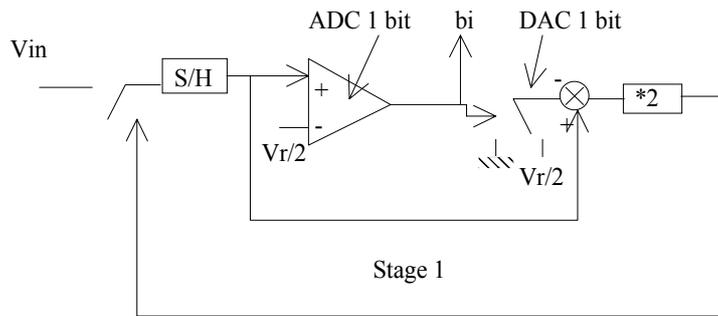
C'est une structure comparable au CV algorithmique :



Chaque étage marche en parallèle de manière décalée. Ils sont utilisables en video (10-14 bits ; 100khz 100Mhz). Par contre on ne peut faire d'asservissement numérique du à la présence du temps de latence.

**IX Convertisseur Algorithmique**

La conversion s'effectue bit après bit du MSB au LSB. Il suffit de reprendre le premier étage du pipeline (1 bit) et de le reboucler sur lui même :



On effectue un encadrement progressif du code binaire finale.

Il faut N coups d'horloge pour obtenir le code finale qui est obtenu de manière série. Bien évidemment, ce type de convertisseur occupe une surface de silicium réduite, consomme peu et n'est pas cher.

## X Convertisseur à suréchantillonnage

### X.1 Principe

Quant on veut améliorer la précision d'un convertisseur, on augmente le nombre de bits. Dans le cas d'un convertisseur à suréchantillonnage, on se base sur un codage minimaliste (un bit) qui se déroule à très haute fréquence bien au delà de la fréquence dite de shannon ( $F_e = 2F_{max}$ ).

De cette manière, on étale le spectre du bruit de quantification sur une plus grande gamme de fréquence, améliorant ainsi le rapport signal sur bruit. De plus, les convertisseurs à suréchantillonnage ont la particularité à repoussé le bruit en haute fréquence, diminuant d'autant ce bruit dans la bande passante. C'est donc par un échantillonnage à haute vitesse que l'on augmente la précision.

### X.2 Influence sur le bruit de quantification

Lors de la numérisation d'un signal sur N bits, le codage de l'amplitude sur un nombre fini de possibilité entraîne un bruit de quantification.

Ce bruit de quantification se caractérise par sa puissance de bruit :

$$P_b = \frac{q^2}{12}$$

avec q le pas de quantification  $q = \frac{\Delta V_{max}}{2^N}$

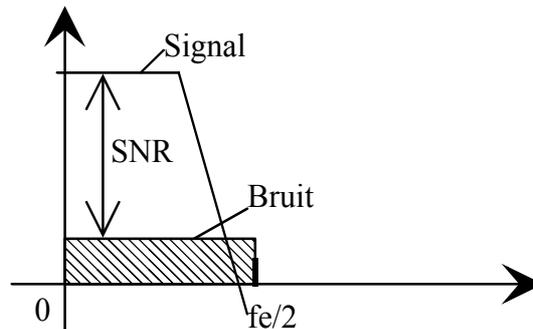
Ce qui donne un rapport signal sur bruit :

$$SNR_{db} = 10 \log\left(\frac{P_s}{P_b}\right) = 6,02N + 1,76dB$$

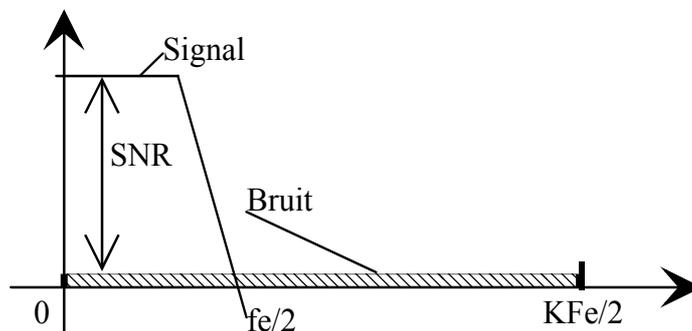
Un bit de code rajoute 6dB de SNR. Cela revient à dire qu'un SNR de 120dB correspond à une quantification sur 20bits.

La répartition spectrale de cette puissance de bruit donne une densité spectrale de puissance uniforme répartie entre  $-F_e/2$  et  $F_e/2$  avec comme amplitude :

$$D_{sb}(f) = \frac{q^2}{12F_e}$$



Si on échantillonne le même signal mais cette fois-ci à une fréquence  $K$  fois supérieure, on va diviser d'autant la densité spectrale du bruit qui va cette fois s'étaler entre  $-Kf_e/2$  et  $Kf_e/2$  :



Ainsi dans la bande de fréquence  $[0 ; f_e/2]$  la puissance du bruit est divisé par  $K$ , soit un SNR :

$$SNR_{db} = 6,02N + 1.76dB + 10\log(K)$$

Utiliser une fréquence d'échantillonnage 4 fois supérieure à la fréquence dite de Shannon revient à augmenter le SNR de 6 db soit un gain de 1 bit.

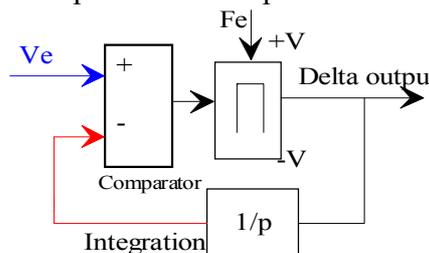
Par cette technique on va pouvoir améliorer le SNR d'une chaîne d'acquisition et soulager le filtre anti-repliement en entrée de chaîne. En effet il doit laisser passer jusqu'à  $F_{max}$  ( $f_e/2$ ) et couper au plus à  $Kf_e/2$ .

Pour revenir à un format et débit de données désiré, on utilise un filtre numérique qui permet de mettre en forme le bruit en le repoussant dans les Hautes Fréquences, puis une décimation (Comb filter) permet en moyennant les données de se recaler à un débit à la fréquence  $f_e$ .

\*\*\*\*\*  
Evolution du Spectre (voir transparent ESIEE Exposé)  
\*\*\*\*\*

### X.3 Le modulateur delta

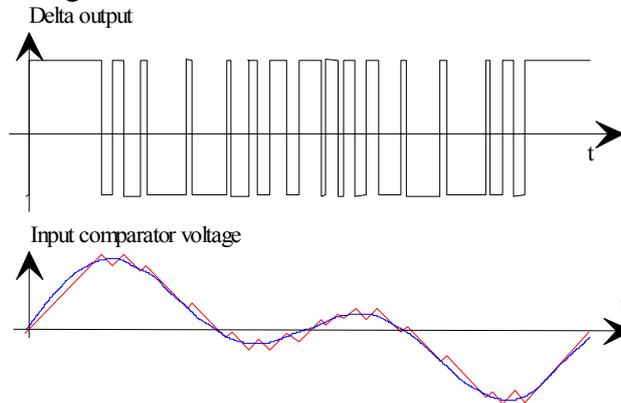
Un modulateur delta se compose d'un comparateur et d'un intégrateur.



On compare le signal d'entrée ( $V_e$ ) à la sortie de l'intégrateur ( $V_{s.int}$ ). En fonction de cette comparaison, on va venir modifier le signe de la tension d'entrée de l'intégrateur. De cette manière, la tension  $V_{s.int}$  va tendre à suivre l'évolution de la tension  $V_e$  :

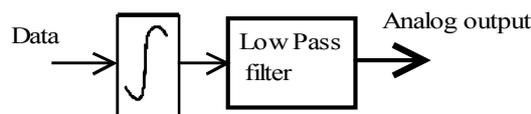
Si  $V_e > V_{s.int}$  : on va intégrer positivement de manière à « forcer »  $V_{s.int}$  à se rapprocher de  $V_e$ .

Si  $V_e < V_{s.int}$  : on va tendre à faire diminuer  $V_{s.int}$  en appliquant une tension négative à l'entrée de l'intégrateur.



On obtient une sortie sur un bit en fréquence élevée.

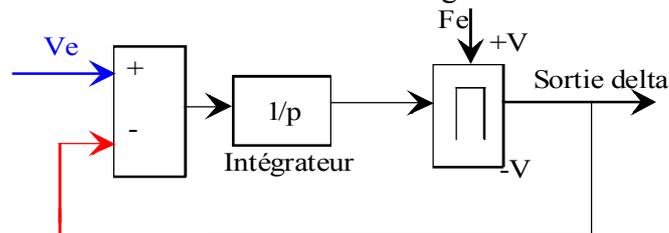
Par cette modulation, on effectue un codage de la pente (dérivée) du signal. La démodulation reprend la même structure du modulateur auquel on associe un filtre passe bas.



Remarque : il existe des structures de modulateur delta qui adapte leur pente d'intégration en fonction de la nature du signal (AN1544 Motorola)

#### X.4 Structure Delta sigma

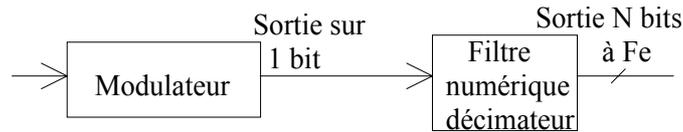
Cette structure découle du modulateur delta. En rajoutant sur l'entrée du signal un intégrateur, le modulateur delta effectue alors un codage de l'amplitude du signal d'entrée. le comparateur voit deux intégrateurs sur ses entrées, en les faisant alors glisser en sortie du comparateur, on obtient la structure du convertisseur sigma delta :



De manière macroscopique, on s'assure que la valeur moyenne de la sortie delta suive la valeur moyenne de la tension d'entrée. En cas d'écart, l'intégrateur vient compenser.

$$\langle V_{s\_delta} \rangle = \langle V_e \rangle$$

La sortie est dans ce cas sur un bit en très haute fréquence. On associe au convertisseur un filtre numérique décimateur qui a pour objectif de fournir un signal numérique au format désiré et à la fréquence souhaitée. De plus il élimine le bruit hors de la bande passante du signal.



\*\*\*\*\*  
 Modélisation :  
 (Le gain en SNR est plus que  $10\log(K)$  entre 0 et  $F_e/2$ ) - Cours Yves Blanchard (I5)  
 Evolution spectre  
 \*\*\*\*\*

## XI Bilan comparatifs des convertisseurs (Chiffres de 1997...)

Type de CV	Fréquence	Résolution	Surface Analogique	Surface Numérique	Consommation
Compteur	100Khz	14-16 bits	300 transistors	500 portes	10 mWatts
Simple rampe	100Khz	10-12 bits	<b>100 transistors</b>	<b>100 portes</b>	10 mWatts
Double rampe	100Khz	>16 bits	<b>150 transistors</b>	<b>150 portes</b>	10 mWatts
Approx. Successives	1Mhz	14-16 bits	300 transistors	500 portes	10 mWatts
Algorithmique	1Mhz	14-16 bits	100 transistors	100 portes	1 mWatt
Flash	<b>&gt;100Mhz</b>	12-14 bits	2500 transistors	5000 portes	1 Watt
Sub-ranging	50Mhz	12-14 bits	600 transistors	500 portes	100 mWatts
Pipeline	100Mhz	10-12 bits	800 transistors	200 portes	100 mWatts
Sigma-delta	1Mhz	<b>&gt;20 bits</b>	1000 transistors	5000 portes	100 mWatts

Remarque :

Comparateur : 10 transistors

Ampli-op : 50 transistors

DAC (N bits) :  $2^N$  transistors +  $2^N$  portes

Registre : 5 portes par bit

Compteur : 6 portes par bit