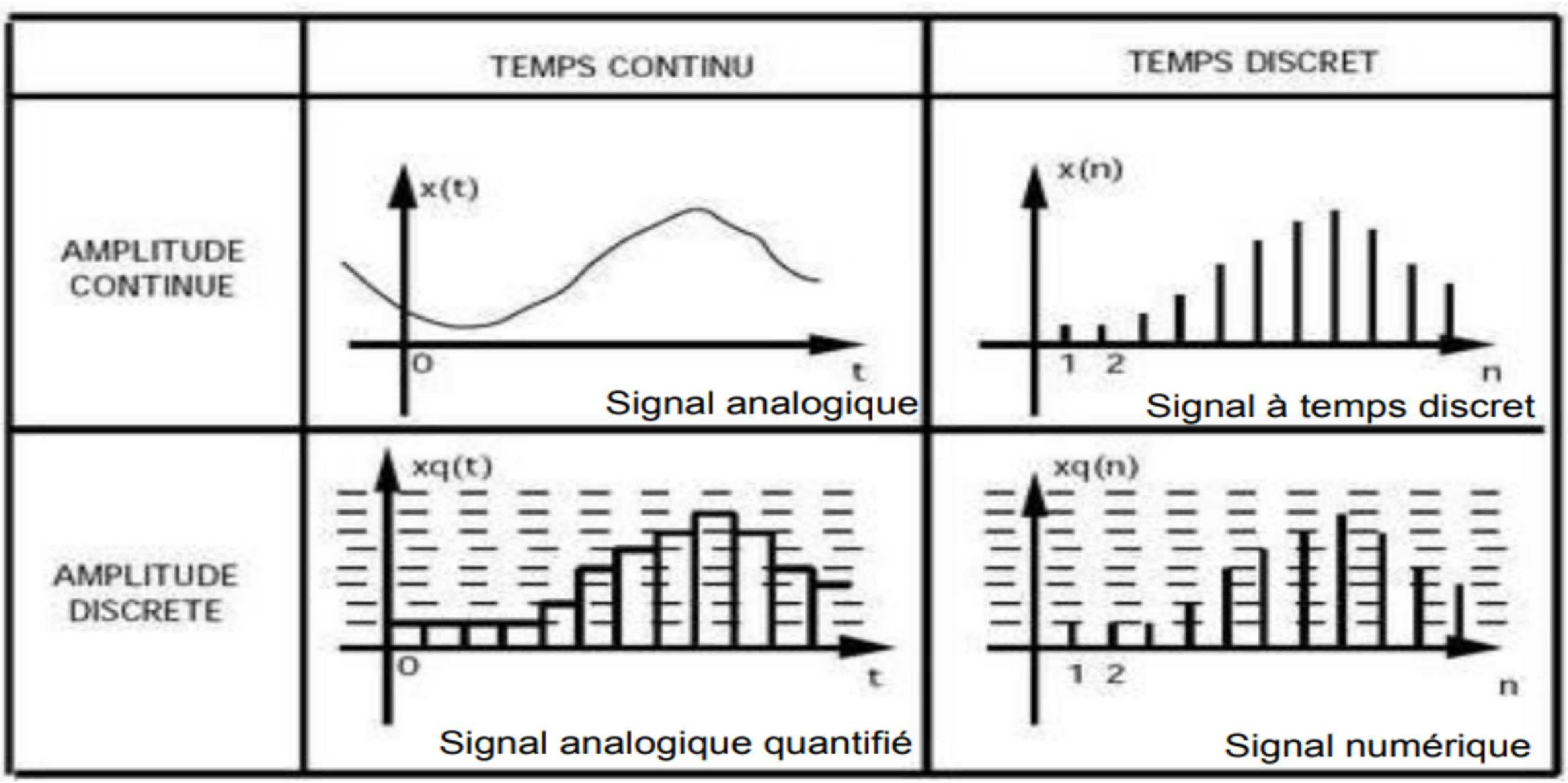
DSP : Processeurs de Traitement du Signal

Ch1 : Généralité &Arithmétique des DSPs

Chapitre 1

1. généralité
2. Représentation en virgule fixe
3. Codage en virgule flottante
4. Codage Virgule fixe Vs codage virgule flottante

Classification des signaux



Traitement numérique du signal

Pourquoi faire du traitement numérique du signal ?

Les principaux avantages du calcul numérique par rapport au calcul analogique s'appliquent aussi au traitement du signal. On a par exemple :

- Grande résistance aux bruits:
- variations des tensions d'alimentation
- variations de température
- interférences électromagnétiques (EMI)
- Précision arbitraire
- Stabilité dans le temps
- Stockage des données sans dégradation
- Duplication des valeurs sans altération
- Programmation flexible
- Développement rapide

Traitement numérique du signal

Exemples d'applications:

- Radars et sonars
- (Télé)communications :
  - modems
  - Réseaux
  - téléphones cellulaires
- Disques durs
- Appareils audio
  - lecteurs CD, DVD, MP3. . .
  - prothèses auditives
  - Synthétiseurs
  - reconnaissance de la parole
- Vidéo
- Automobile
- Robotique

Exemples d'algorithmes:

- Filtrage
- Transformées
- Codage/décodage
- Compression/décompression
- Cryptage/décryptage
- Contrôle
- Reconnaissance de la parole
- Synthèse de signaux
- Elimination d'échos
- Estimation spectrale

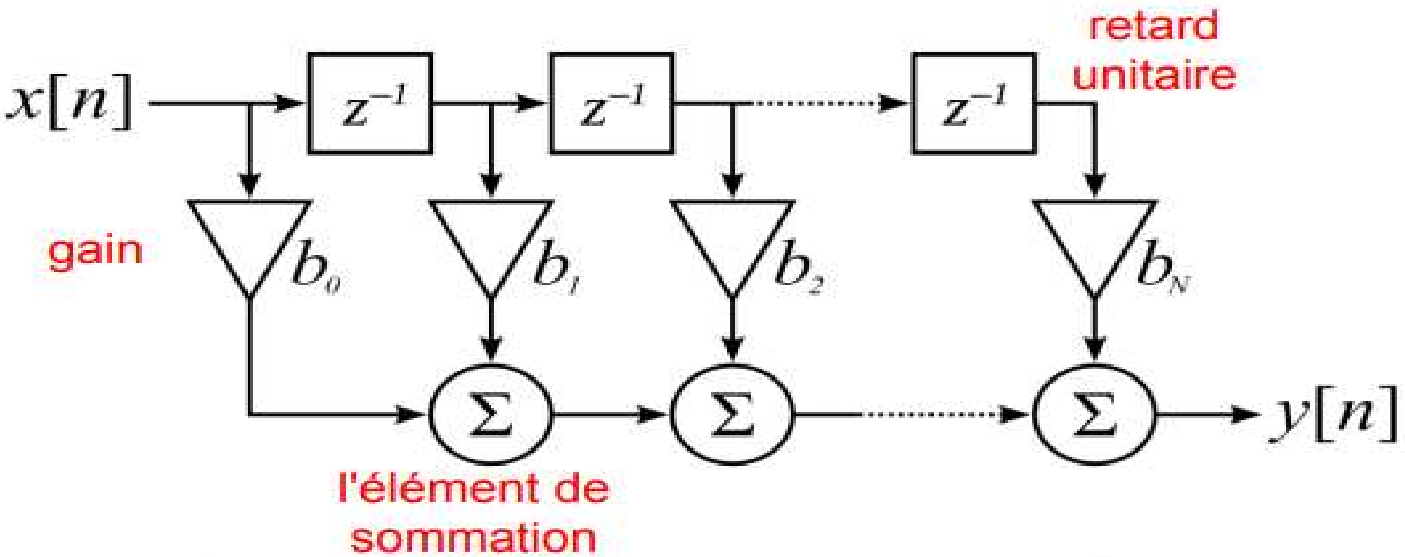
Equations des algorithmes

Algorithm	Equation
Finite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)$
Infinite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)+\sum_{k=1}^{N} b_k\, y(n-k)$
Convolution	$y(n)=\sum_{k=0}^{N} x(k)h(n-k)$
Discrete Fourier Transform	$X(k)=\sum_{n=0}^{N-1} x(n)\exp[-j(2\pi/N)nk]$
Discrete Cosine Transform	$F(u)=\sum_{x=0}^{N-1} c(u).f(x).\cos\left[\frac{\pi}{2N}u(2x+1)\right]$

Filtres RIF (non récursif)

$$y[n] = \sum_{k=0}^{N-1} b_k \cdot x[n-k]$$

N = Nombre de coefficients (ordre du filtre)
b_k = Coefficients de la fonction de transfert du filtre



- Les filtres RIF sont forcément stables (peu importe les coefficients utilisés).
- La complexité d'un filtre RIF est moindre que celle d'un filtre RII du même ordre.
- Les filtres RIF sont moins sensibles aux erreurs de quantification que les filtres RII (pas de récursivité empêche les erreurs cumulatives)

Filtre RII (récursif)

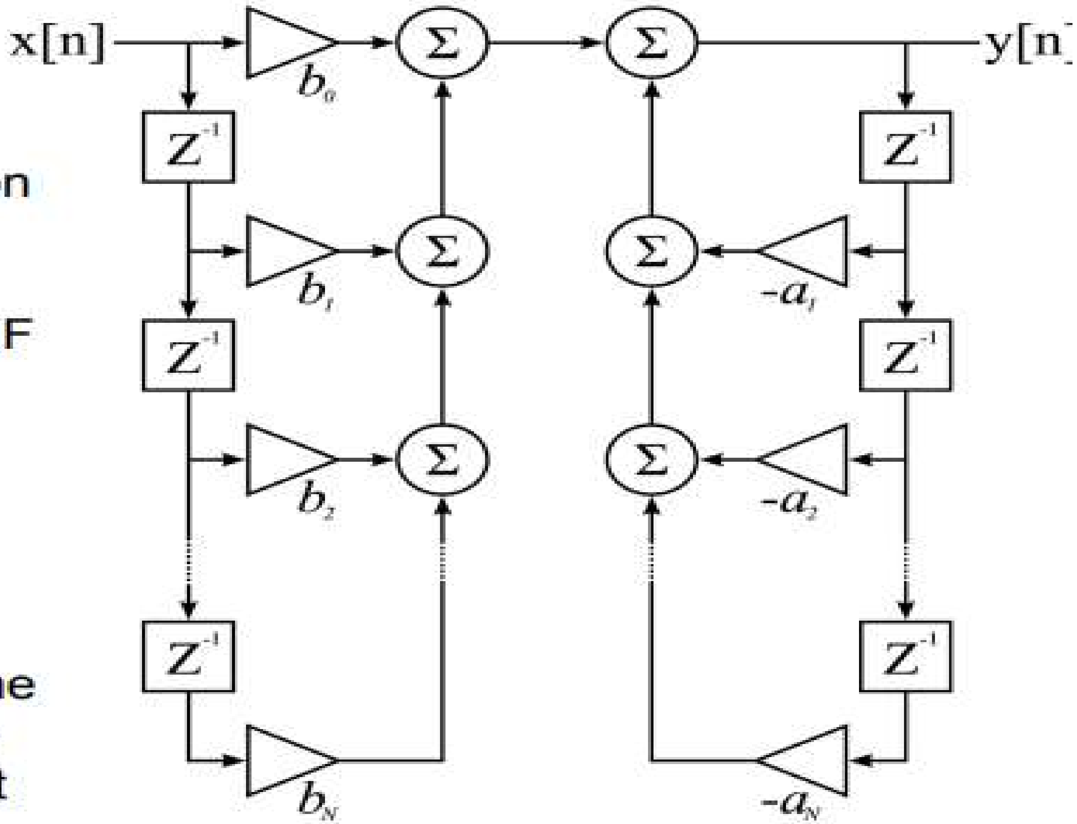
$$y[n] = \sum_{k=0}^{N} b_k \cdot x[n-k] - \sum_{k=1}^{M} a_k \cdot y[n-k]$$

- Les filtres RII ne sont pas forcément stables (la stabilité dépend de la position des pôles dans le plan complexe) ;

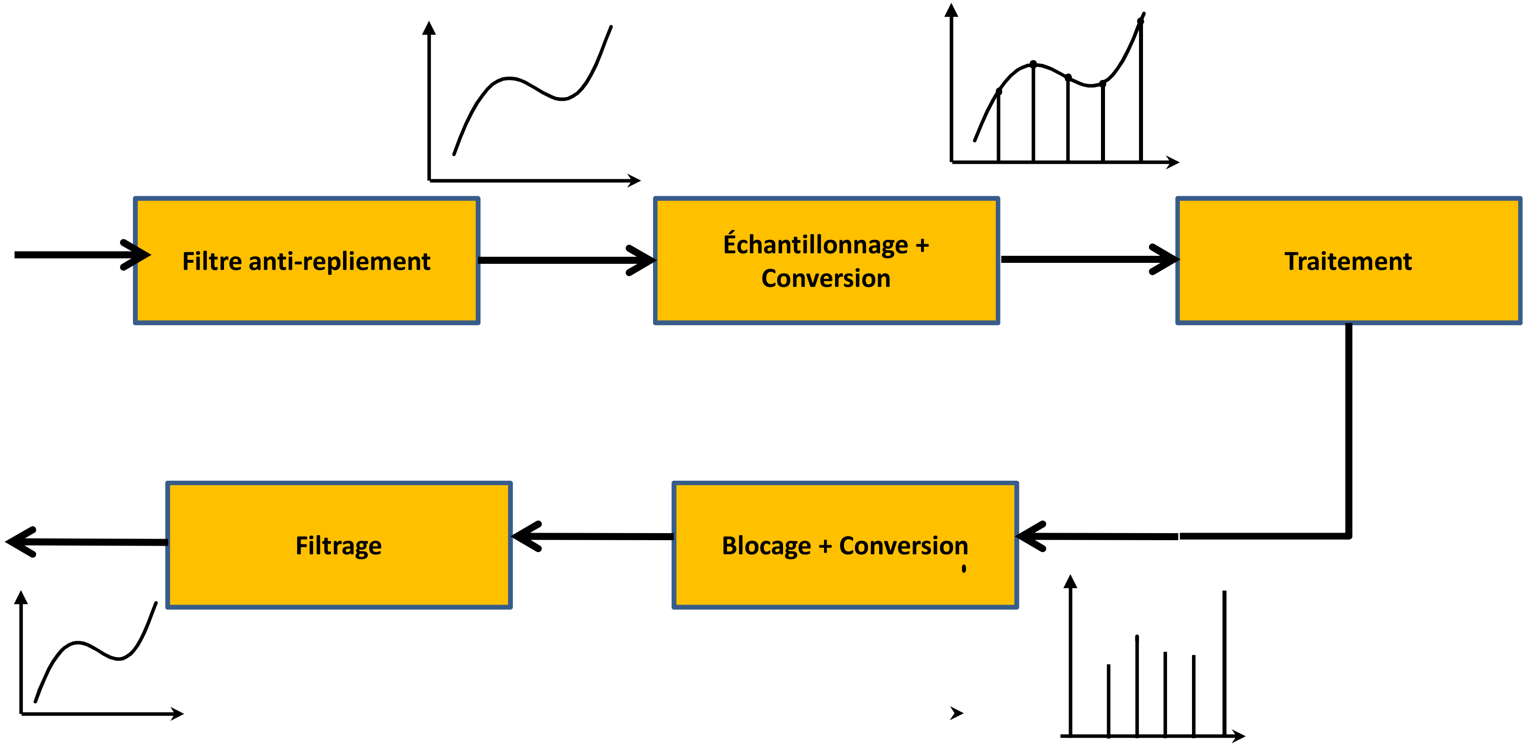
- Plus de calculs par rapport à un filtre RIF de même niveau ;

- Ils sont plus sensibles aux erreurs de quantification (La récursivité peut générer des erreurs cumulatives) ;

- Il est plus sélectif qu'un filtre RIF du même ordre, c'est-à-dire que la transition entre la bande passante et la bande rejetée est plus rapide.

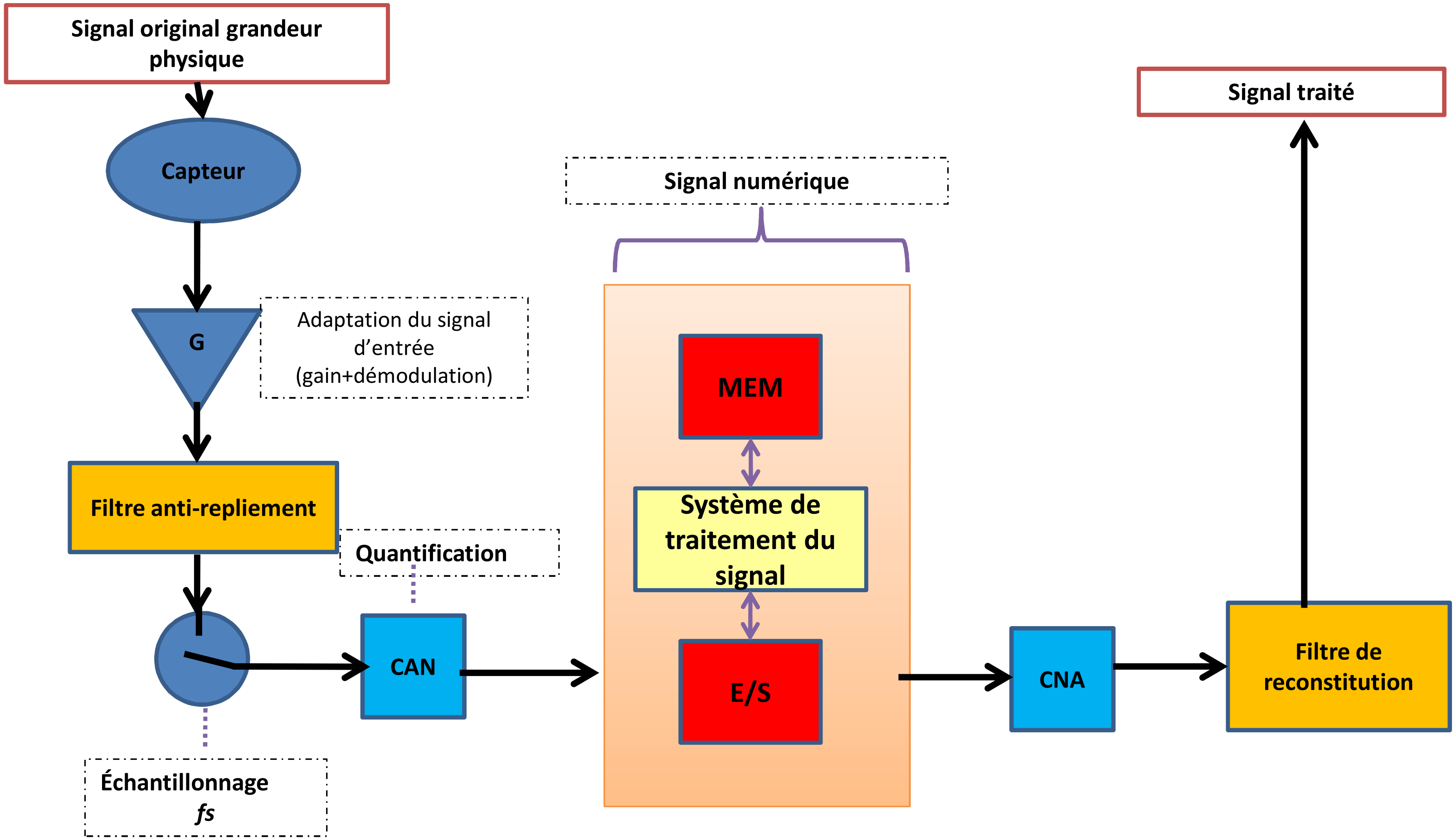


Traitement numérique du signal



- De quelles solutions dispose-t-on pour effectuer ce travail?

Chaine de traitement numérique du signal



Chaine de traitement numérique du signal

○ Filtre anti-repliement

❑ Phénomène de repliement spectral ou aliasing:

- Si la fréquence d’entrée f est supérieure à fs/2 (fs est la fréquence d’échantillonnage)
- Solution: ne pas injecter à l’entrée de l’échantillonneur de signal dépassant la valeur critique fs/2

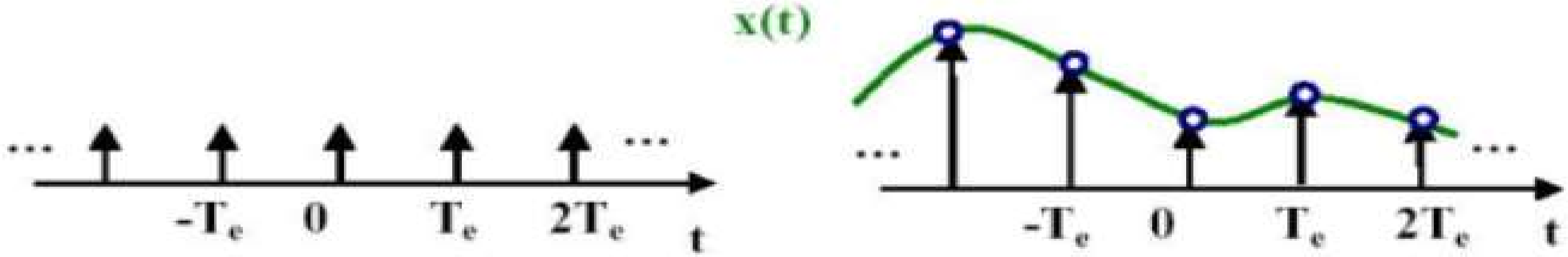
❑ Éviter le repliement spectral:

- Filtrage du signal analogique par un filtre passe-bas (appelé filtre anti-repliement ou anti-aliasing) avant son injection dans l’échantillonneur.
- Filtre anti-repliement idéal: pente infinie
- Filtre anti-repliement réel: pente finie induisant le passage d’une partie du spectre au delà de la fréquence critique. D’où la présence d’un repliement spectral local.

Chaine de traitement numérique du signal

○ Echantillonnage

L'échantillonnage idéal prélève des échantillons à la cadence T_e de façon instantanée.



T_e est la période d'échantillonnage, $F_e = \frac{1}{T_e}$ est la fréquence d'échantillonnage.

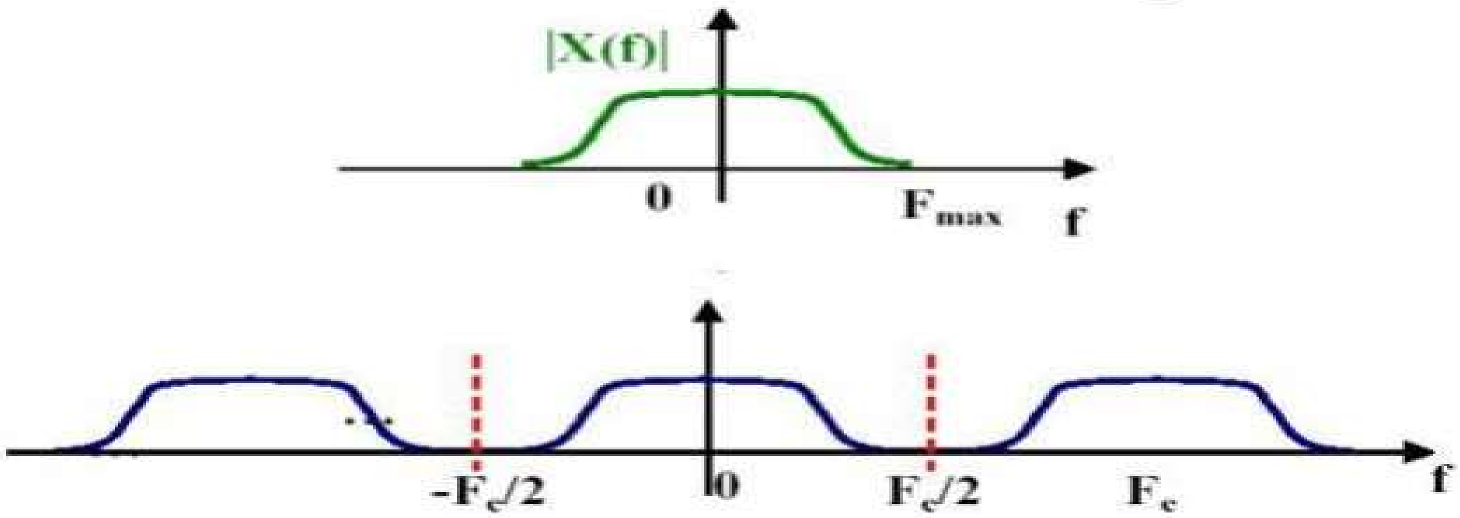
Le signal échantillonné est :

$$x_d(t) = \sum_{n \in \mathbb{Z}} x(nT_e)\delta(t - nT_e)$$

Un disque CD (Compact Disc) utilise une fréquence d'échantillonnage de 44 kHz.

Echantillonnage et périodisation

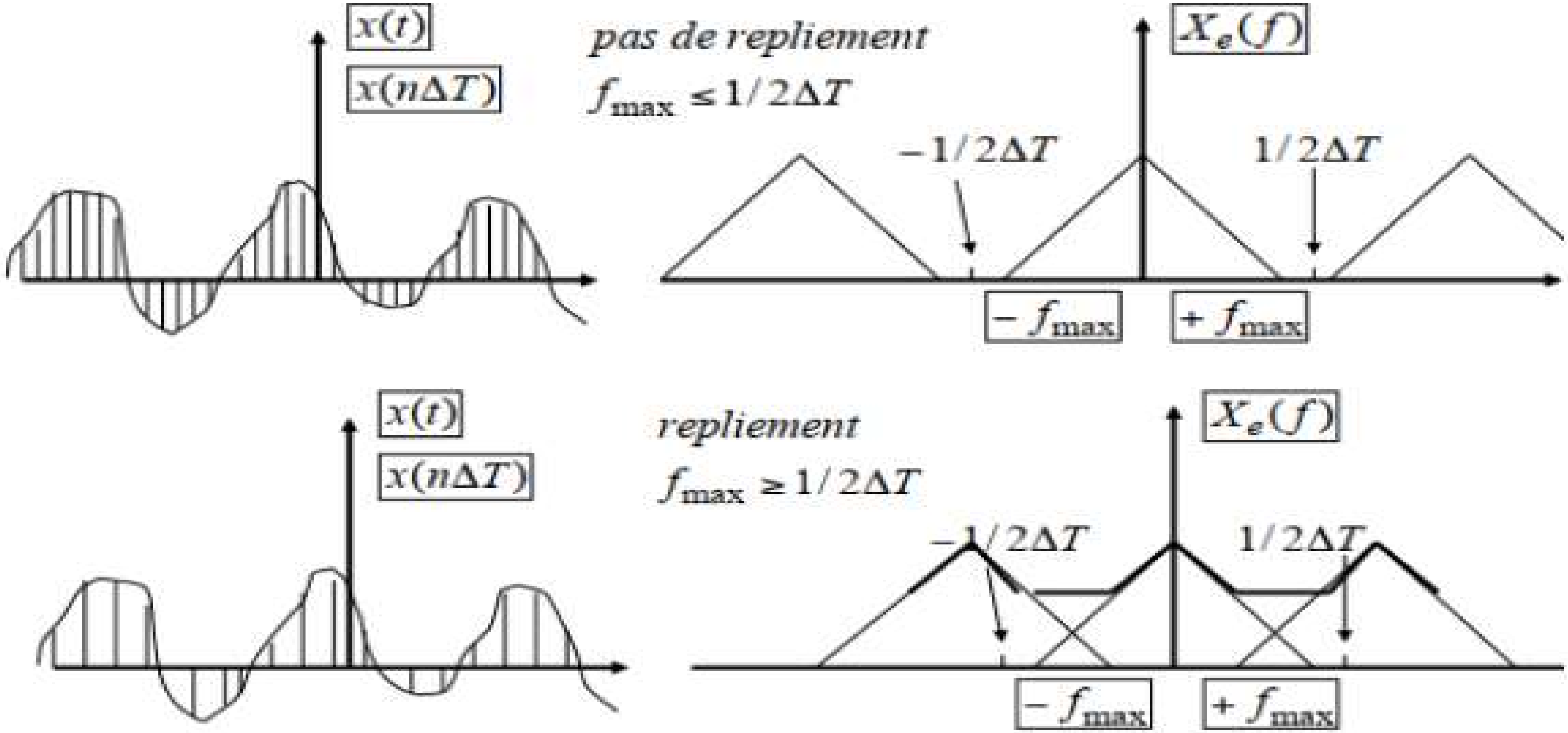
Échantillonnage temporel <=> périodisation en fréquence



Pour éviter une superposition des spectres élémentaires il est nécessaire d'imposer <u>le théorème de Shannon</u>

$$F_e/2 \geq F_{max} \rightarrow F_e \geq 2 \times F_{max}$$

Repliement de spectre dans le domaine fréquentiel



Pour éviter le repliement de spectre on élimine les fréquences contenues dans le signal analogique supérieures à $F_e/2$

Chaine de traitement numérique du signal

○ Quantification:

❑Quantification du signal

❑En sortie d'un échantillonneur bloqueur: signal analogique

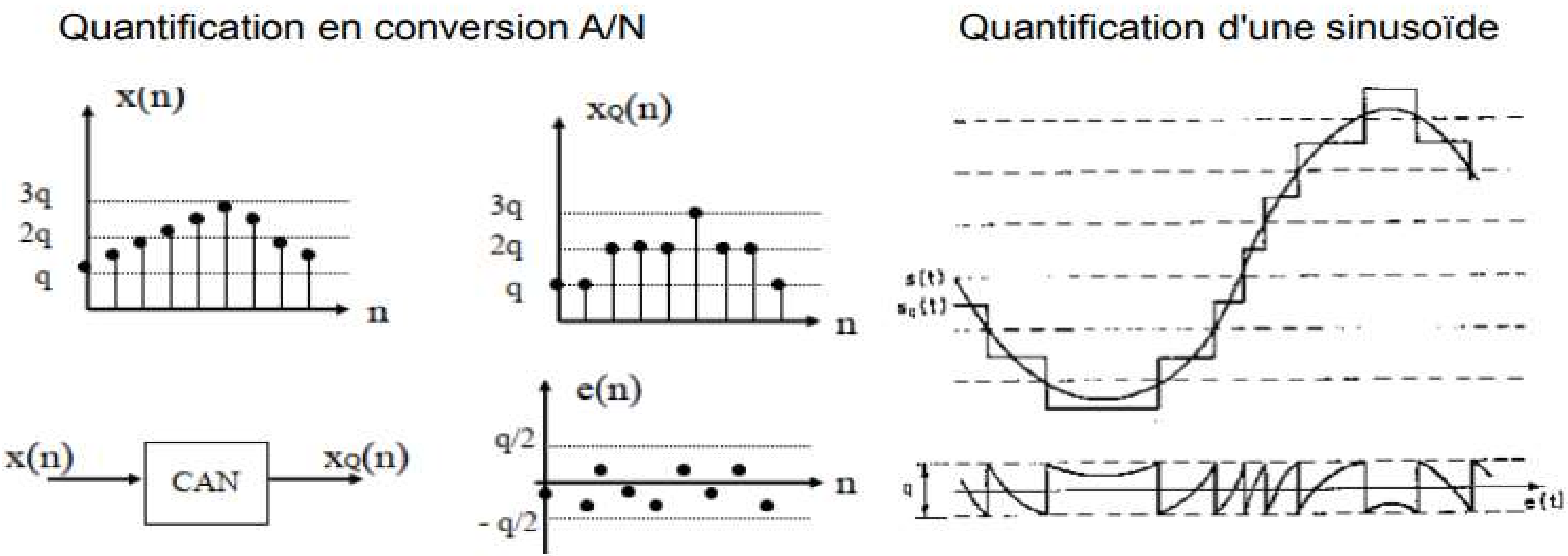
❑Conversion assurée par un circuit appelé convertisseur analogique numérique: numérisation du signal

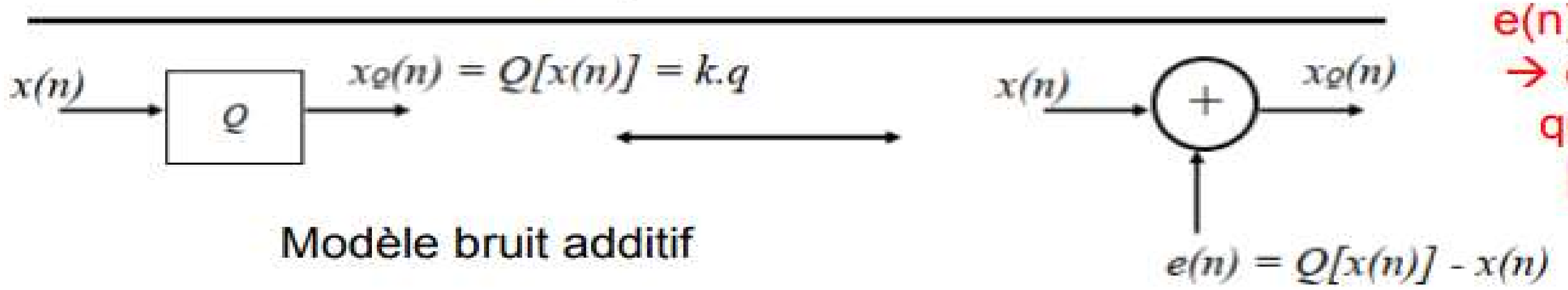
❑Transformation d'une valeur analogique d'entrée (tension ou courant) en un mot binaire codée sur n bits (n est un paramètre important dans un convertisseur: résolution)

❑Erreur de quantification maximale: exemple
- Les 8 bits donnent une résolution de 256 mots binaires possibles
- Erreur de quantification maximale du convertisseur est: 1/256 par pas

Chaine de traitement numérique du signal

- Quantification





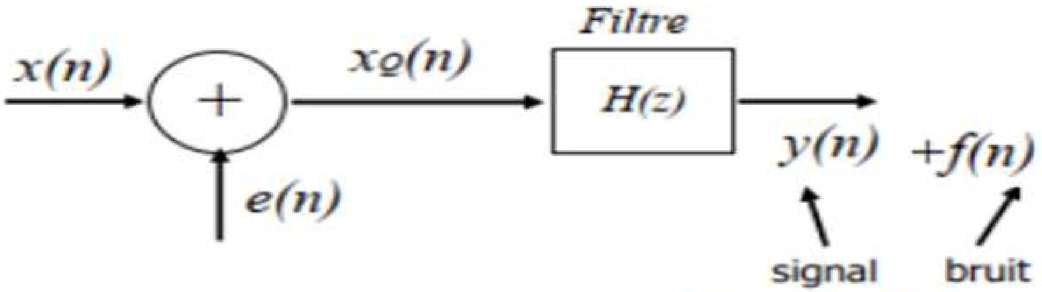
$e(n) = x_Q(n) - x(n)$

→ est l'erreur de quantification

$|e(n)| \leq q/2$

Chaine de traitement numérique du signal

- Quantification: rapport signal sur bruit



Filtrage du bruit de conversion

$$f(n) = e(n) * h(n)$$

En entrée du filtre: – Signal x(n) + Bruit de conversion e(n)

$$\sigma_e^2 = \frac{q^2}{12}, \sigma_x^2 \textit{ puissance du signal d'entrée}$$

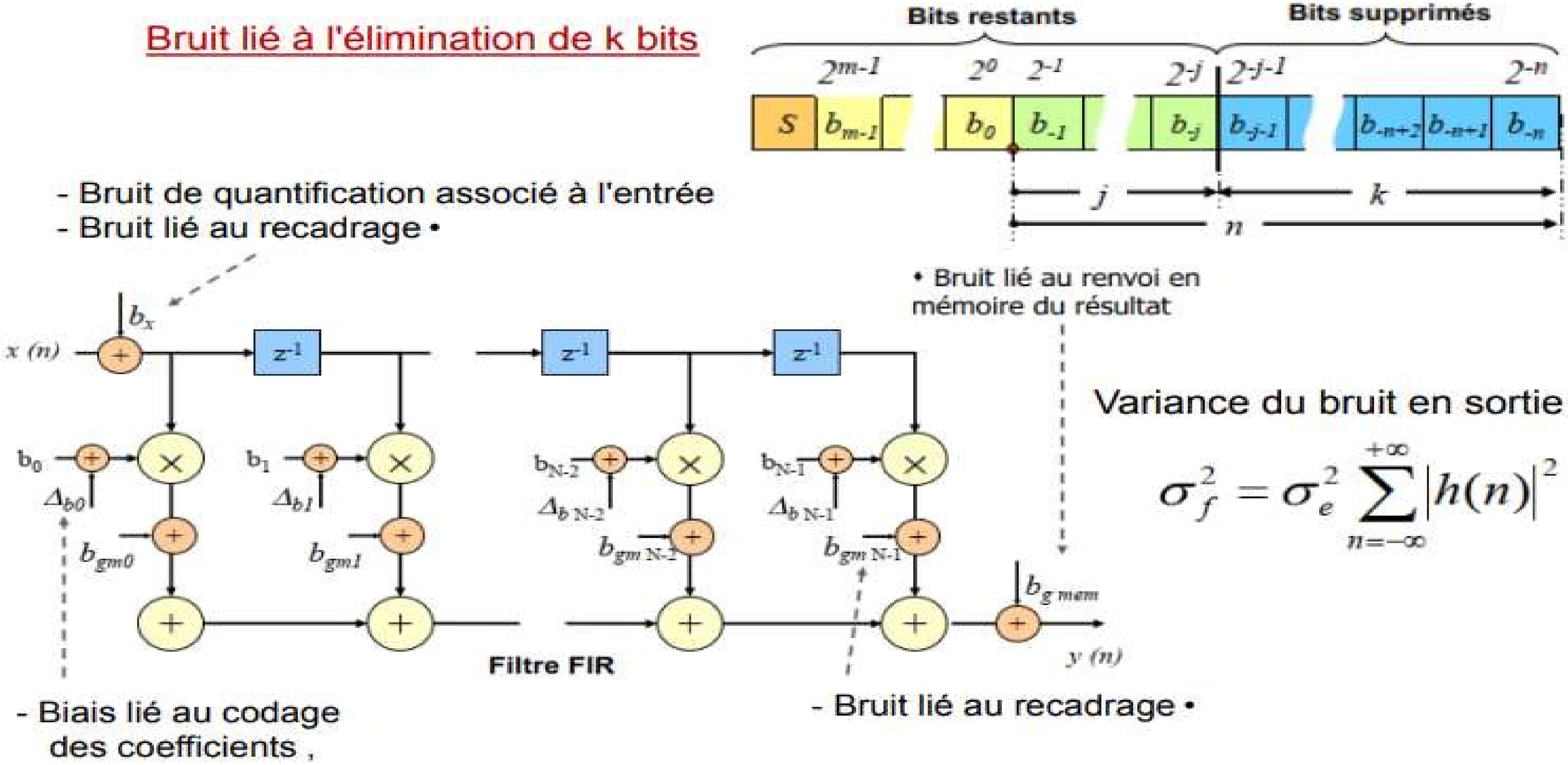
$$RSB = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{q^2/12} = 12.2^{2(b-1)}\sigma_x^2$$

$$RSB_{dB} = 10\log RSB = 6.02b + 4.77 + 10\log\sigma_x^2$$

Le RSB augmente de 6dB par bit ajouté

Chaine de traitement numérique du signal

- Source du bruit dans un filtre

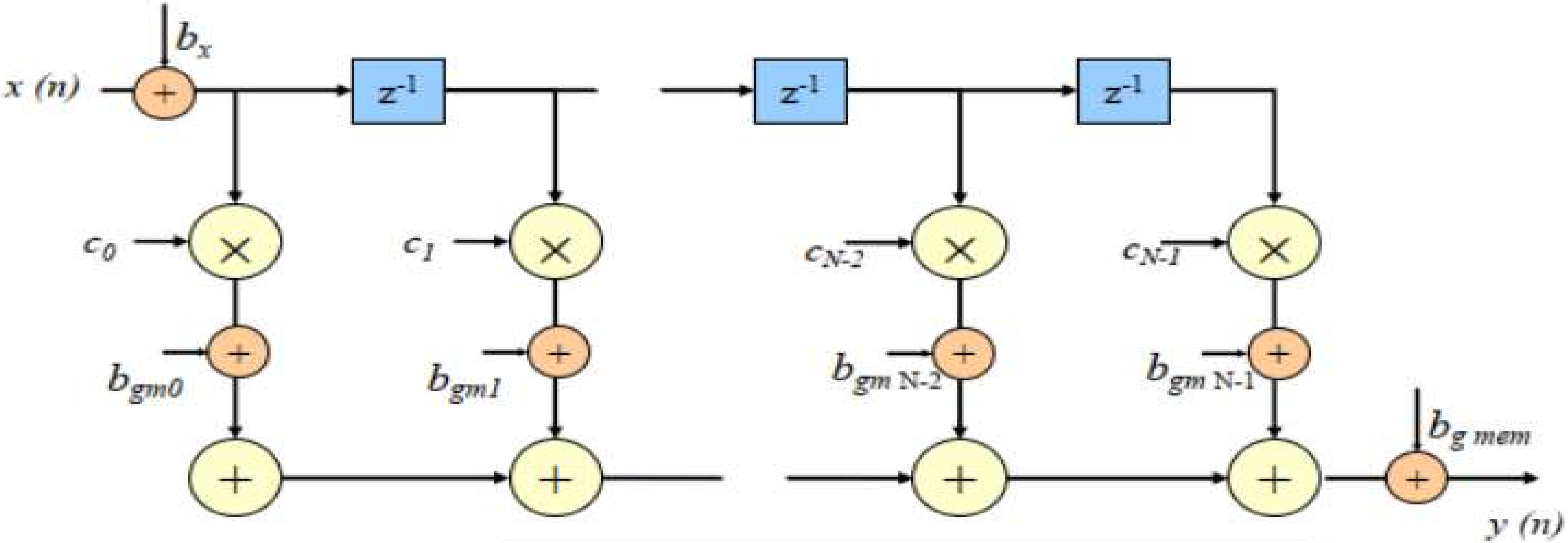


Variance du bruit en sortie

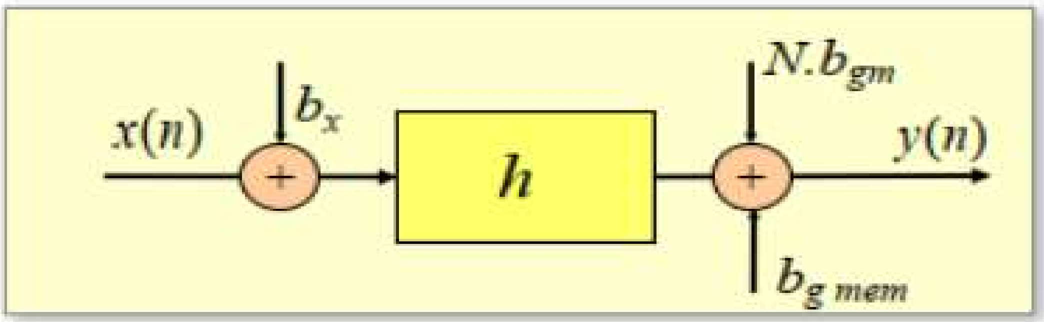
$$\sigma_f^2 = \sigma_e^2 \sum_{n=-\infty}^{+\infty} \left|h(n)\right|^2$$

Chaine de traitement numérique du signal

- Source du bruit dans un filtre



Modèle équivalent:



Chaine de traitement numérique du signal

- Puissance du bruit en sortie

<u>Quantification arrondi:</u>

Puissance du bruit dans les coefficients *h(n)*

Puissance du bruit dans l'entrée *x(n)*

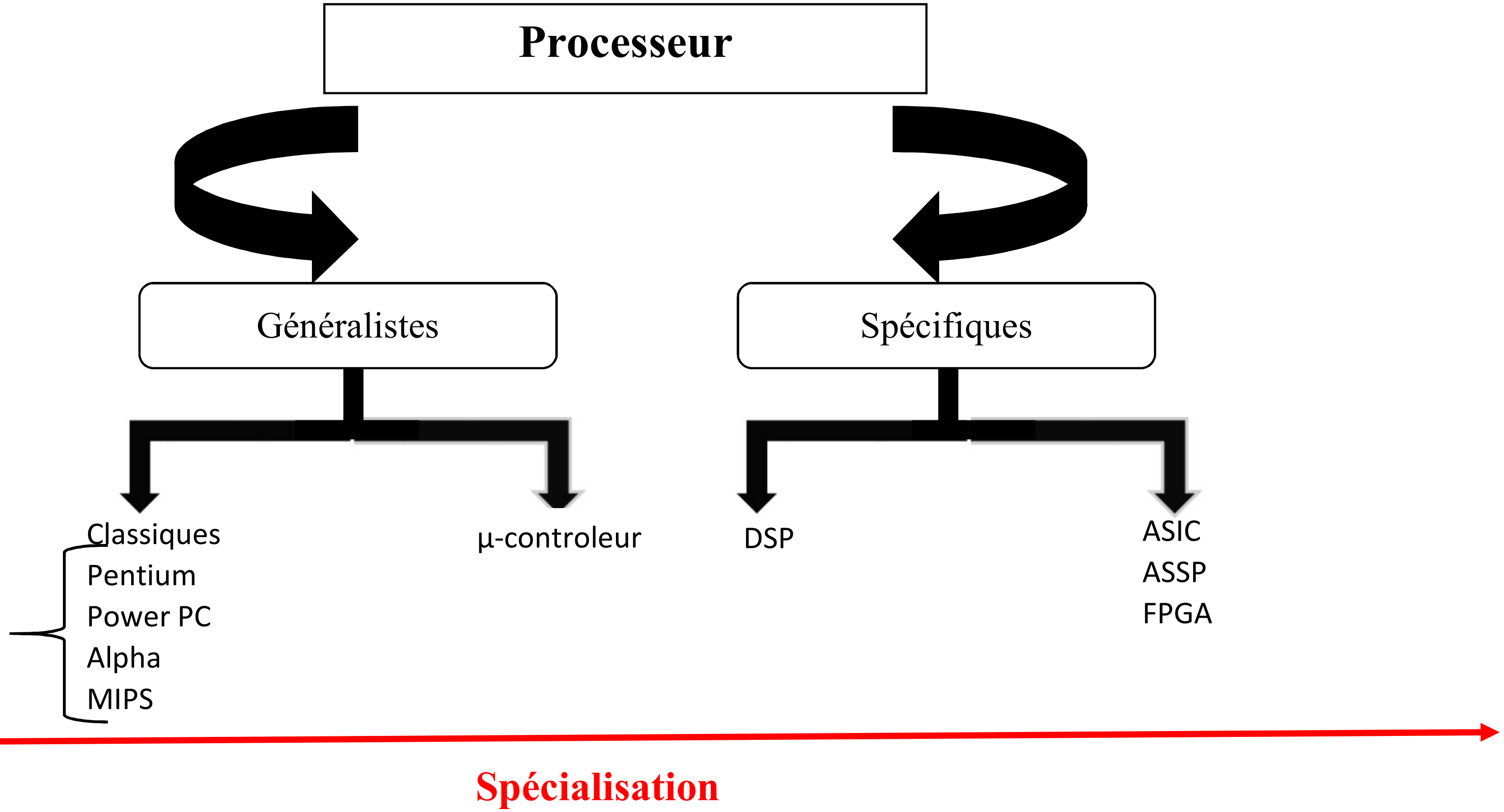
Puissance du bruit dans les sorties *y(n)*

$$\sigma_{b_y}^2 = \sigma_{b_e}^2 \sum_{m=-\infty}^{+\infty} |h(m)|^2 + \sigma_{b_{mem}}^2 + \sum_{i=0}^{N-1} \sigma_{b_{gm.i}}^2$$

Généralement
$q_e = q_{mem} = q_{mi}$

$$\sigma_{b_y}^2 = \frac{q_e^2}{12} \sum_{m=0}^{N-1} c_m{}^2 + \frac{q_{mem}^2}{12} + N.\frac{q_{mi}^2}{12}$$

Classification des processeurs



- ASIC: Application Specific Integrated Circuit
- ASSP: Application Specific Standard Product
- FPGA: Field Programmable Gate Array

Classification des processeurs

- Le processeur généraliste (GPP):

❑ Cout relativement élevé
❑ Forte consommation électrique
❑ Bonne performances en calcul numérique: certains surpassent les DSPs en puissance brute

GPP moins bon que les DSPs pour:

❑ La consommation électrique
❑ La gestion des entrées-sorties
❑ Le prix

- Le microcontroleur:

❑ Faible cout
❑ Mémoire limitée
❑ Peu adapté aux signaux numériques
❑ Adapté aux taches de contrôle
❑ Programmation bas niveau

Classification des processeurs

- ASIC/ASSP (Application Specific….)

- ❑ Circuit intégré dédié à une application

  -dans une entreprise: ASIC

  - pour un marché (ex: téléphone):ASSP
- ❑ Ne sont pas programmables
- ❑ Mise en œuvre complexe et longue
- ❑ Cout élevé

- FPGA: sorte d'ASIC programmable

Classification des processeurs

- Caractéristiques classiques des DSP

- ❑ Chemin de données organisé pour traitement du signal
- ❑ Jeu d’instructions spécialisé
- ❑ Plusieurs mémoire et plusieurs bus
- ❑ Modes d’adressage spécifiques
- ❑ Périphériques spéciaux pour le traitement du signal
- ❑ Augmentation du parallélisme

Classification des processeurs

- Caractéristiques classiques des DSP
- *Augmentation du parallélisme*
  - Calculs
    - Unités de calcul en parallèle
  - Mémoire à accès multiples
    - Lecture/Écriture de plusieurs données simultanément
  - Pipeline
    - Découpage des instructions de façon à les exécuter à intervalles plus rapprochés

Classification des processeurs

	ASIC	FPGA	GPP	DSP
Performance	**Très élevée**	**élevée**	**moyenne**	**Moyen-élevée**
Flexibilité	**Très faible**	**élevée**	**élevée**	**élevée**
Consommation énergétique	**Très faible**	**faible**	**moyenne**	**faible-moyenne**
Temps de développement	**Long**	**moyen**	**court**	**court**

Le TNS:besoins

- Cahier des charges:

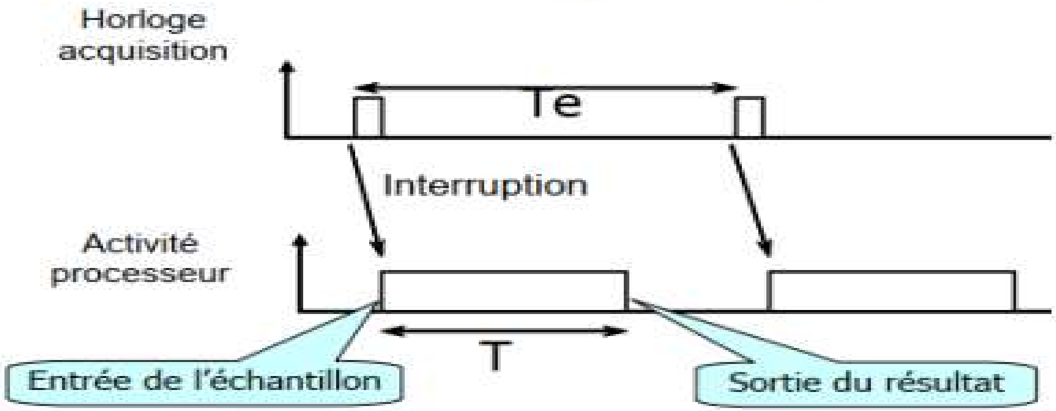
1. Calculs rapides
2. Contraintes temps réel (entrées/sorties à débit fixe)
3. Contraintes systèmes embarqués (taille limitée, faible consommation d'énergie)
4. Production de masse

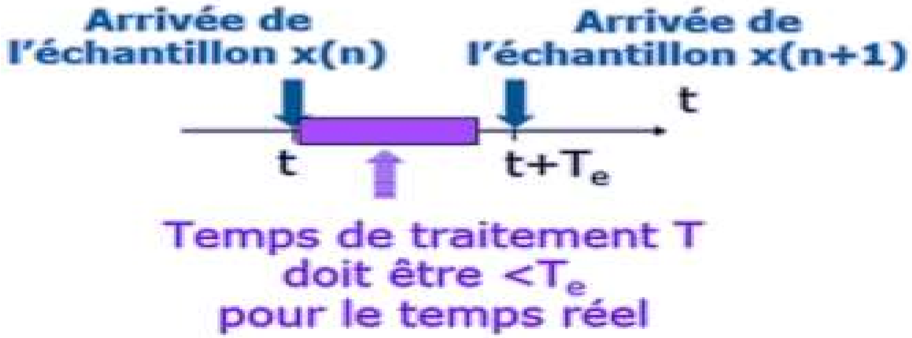
Les DSPs ont une architecture matérielle et logicielle dédiée permettant de répondre à ces besoins

Contrainte temps réel

- Contrainte de temps réel
- Ex : taux d'échantillonnage 48kHz
  $T_e = 20.833\ \mu s$, **Période d'échantillonnage**

Soit: T = Temps de traitement





- Objectifs :
  - Réduire les accès mémoire
  - Augmenter les accès mémoire simultanés
  - Réduire le temps passé à faire des calculs

Instruction MAC (multiplication-accumulation) en 1 seul cycle d'instruction

Limites de DSP

- Taux d'échantillonnage
  - La bande passante d'un système DSP est limitée par la fréquence d'échantillonnage du CAN et les périphériques matériels.
  - L'application peut être trop complexe pour être réalisée en temps réel
- Erreurs de quantification et arithmétiques
  - Les algorithmes DSP sont implémentés en utilisant un nombre fixe de bits avec une précision et une plage dynamique limitées

C’est quoi un DSP?

un peu d’histoire!!

- Les DSP ont été initialement développés pour des applications de radars militaires et de télécommunications cryptées dans les années 70.

- C'est Texas Instruments® qui en 1978 introduit un DSP pour la synthèse de la voix pour des applications très grand public. Il aura fallu 15 ans supplémentaires pour que les DSP deviennent des composants incontournables de l'électronique grand public.

Des microprocesseurs aux DSPs

- Le premier ordinateur électronique qui a vu le jour est ’ENIAC(Electronical Numerical Integrator And Calculator)

- Inconvénients de cette machine: lourdeur de programmation et complexité.

- Elle dispose de 2 mémoires séparées: une pour les données et l’autre pour les instructions du programme ⇒ Proposition de John Von Neumann: une seule mémoire centrale afin de simplifier la structure de l’ENIAC.

C'est quoi un DSP?

RISC/CISC:

❑CISC (Complex Instruction Set Computer): décomposition de chaque instruction en une suite d'opérations simples permises par le circuit interne du processeur

- Exemple: le noyau du processeur 8032 requiert 12 cycles machines par instruction. Certaines instructions, comme la multiplication, peuvent ainsi demander 36 cycles machines
- Problème principal: le calcul de la charge logicielle du processeur est très difficile du fait de la très large variabilité du nombre de cycles nécessaires pour l'exécution d'une instruction
- Besoin des applications « temps réel »: on ne dispose que d'un laps de temps donné pour réaliser certains calculs (ex: algorithme de traitement audionumérique en temps réel)

❑RISC (Reduced Instruction Set Computer)

- Exécution d'une instruction en un seul cycle d'horloge suite à l'optimisation du silicium afin de restreindre le jeu d'instruction

C’est quoi un DSP?

- Un DSP est un microprocesseur RISC dont la structure et le jeu d’instructions sont optimisés pour une exécution la plus rapide possible des algorithmes de traitement du signal

- Un DSP exécute une instruction en un cycle d’horloge (1 opération = 1instruction)

- Les DSPs réalisent 4 principaux types de calculs pour lesquels ils doivent être optimisés:

- Arithmétique/Logique: +,-, binaires
- Décalage logiques et arithmétiques lors de l’alignement des virgules
- Multiplication: opération assez courante en traitement du signal. Les DSPs disposent d’un multiplicateur câblé afin d’assurer l’exécution la plus rapide que possible
- Adressage: besoin d’avoir des accès spécifiques aux données en mémoire avec le minimum d’opérations depuis l’UAL

Profils d'utilisation du DSP

- Embarqué
  - Faible coût unitaire
  - Faible consommation : part importante de la consommation pour la mémoire
  - Architecture limitée au strict nécessaire
  - Temps réel

- Haute performance
  - Puissance : Calcul intensif
  - Parallélisme
    - Multiplication des unités de calcul internes
    - Interfaces multi-DSP
  - Interface avec un ordinateur hôte

Applications des DSP

- **Communications**
  - Filaire (xDSL, câble)
  - Sans fil (cellulaires, télévision numérique, radio numérique)
  - Modem
  - Cryptage
- **Image / vidéo**
  - Compression/Codage
  - Composition
  - Traitement
- **Audio**
  - Mixage et effets
  - Suppression de bruit
  - Annuleur d'echo
- **Militaire**
  - Imagerie : radar, sonar…
  - Cryptographie
  - Guidage de missiles

Applications des DSP

- **Automatisation**
  - Commande de machines
  - Contrôle de moteurs
  - Robots

- **Electronique Automobile**
  - Contrôle du moteur
  - Assistance au freinage
  - Aide à la navigation
  - Commandes vocales

- **Biomédical**
  - Équipements de monitoring
    - Signaux biophysiques
    - ElectroEncéphaloGramme (EEG)
    - ElectroCardioGramme (ECG)
  - Radiographie

- **Instrumentation**
  - Analyseurs de spectre
  - Générations de fonctions
  - Analyseurs de régimes transitoires

Principaux constructeurs de DSP

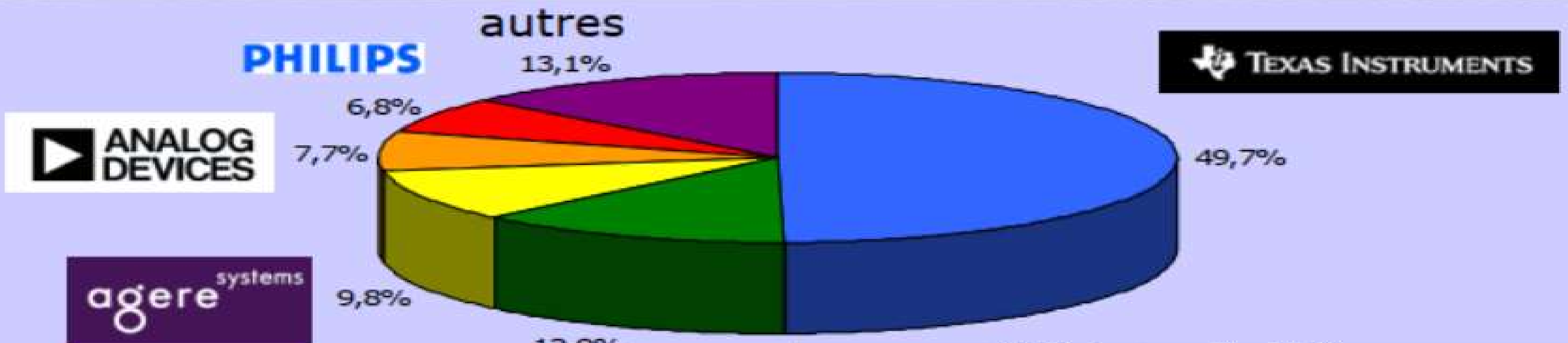


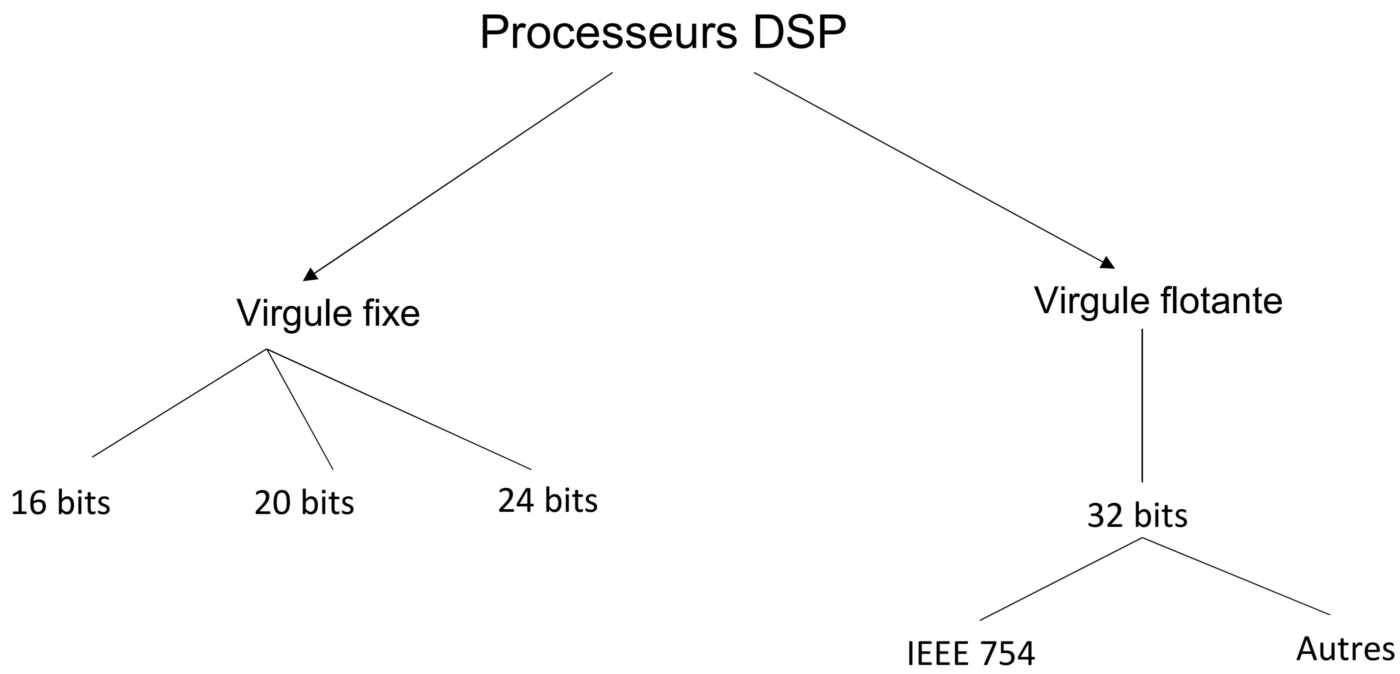
Table 1.2 Current commercially available DSP processors

Vendor	Family	Arithmetic type	Clock speed
Texas instruments	TMS320C24x	Fixed-point	40 MHz
	TMS320C28x	Fixed-point	150 MHz
	TMS320C54x	Fixed-point	160 MHz
	TMS320C55x	Fixed-point	300 MHz
	TMS320C62x	Fixed-point	300 MHz
	TMS320C64x	Fixed-point	1 GHz
	TMS320C67x	Floating-point	300 MHz
Analog devices	ADSP-218x	Fixed-point	80 MHz
	ADSP-219x	Fixed-point	160 MHz
	ADSP-2126x	Floating-point	200 MHz
	ADSP-2136x	Floating-point	333 MHz
	ADSP-BF5xx	Fixed-point	750 MHz
	ADSP-TS20x	Fixed/Floating	600 MHz
Freescale	DSP56300	Fixed, 24-bit	275 MHz
	DSP568xx	Fixed-point	40 MHz
	DSP5685x	Fixed-point	120 MHz
	MSC71xx	Fixed-point	200 MHz
	MSC81xx	Fixed-point	400 MHz

Chapitre 1

1. Généralité
2. Représentation en virgule fixe
3. Codage en virgule flottante
4. Codage Virgule fixe Vs codage virgule flottante

Représentation numérique des DSP commerciaux



Représentation numérique des DSP commerciaux

- Les DSP à virgule fixe: 16, 20, 24 bits
  - Les données sont représentées comme étant des nombres fractionnaires à virgule fixe, (exemple -1.0 à +1.0), ou comme des entiers classiques.
  - La représentation de ces nombres fractionnaires s'appuie sur la méthode du «complément à deux».
  - Permet facilement l'addition binaire de nombres positifs et négatifs.

- Les DSP à virgule flottante: 32 bits
  - Les données sont représentées en utilisant une mantisse et un exposant.
  - La représentation de ces nombres s'effectue selon la formule suivante :

$$n = signe \;\; mantisse \;\; 2^{exposant}$$

  - Généralement, la mantisse est un nombre fractionnaire (-1.0 à +1.0), et l'exposant est un entier indiquant la place de la virgule en base 2.

Représentation numérique des DSP commerciaux

- La représentation des nombres doit répondre à deux exigences contradictoires:
  - Précision: intervalle entre deux rationnels codés
    - doit être le plus petit possible
  - Dynamique: rapport entre le plus grand rationnel et le plus petit rationnel codés
    - doit être la plus étendue possible
- Dans ce contexte les unités de calcul des DSP travaillent
  - **soit en** *format fixe*
  - **soit en** *format flottant*

Représentation numérique des DSP commerciaux

Virgule fixe ou virgule flottante ?

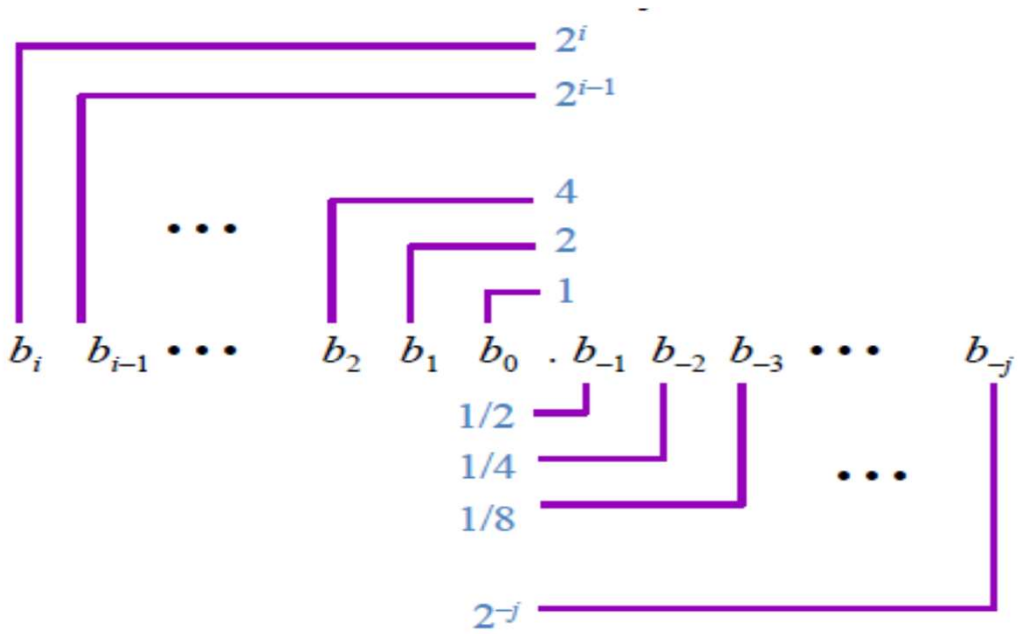
- Les profiles d'application nécessitant un processeur flottant sont :
  - Haute précision: Dans le cas d'un entier, le FLP (Floating-Point) a une plus grande précision plus que dans le cas d'un nombre réel
  - Dynamique importante: L'exponentiation augmente considérablement la plage dynamique

- Les inconvénients du DSP flottant sont :
  - Consommation
  - Hardware plus complexe, coût élevé
  - Moins performant que le DSP à virgule fixe

- 95% des DSP sont en virgule fixe.

Représentation en virgule fixe

Représentation du nombre fractionnaire binaire



Representation rationnelle du nombre binaire: $\sum_{k=-j}^{i} b_k . 2^k$

Représentation en virgule fixe

Conversion binaire-décimale

- $23.47_{10} = 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2}$
- $10.01_2 = 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 1 \times 2 + 0 \times 1 + 0 \times {}^1/_2 + 1 \times {}^1/_4 = 2 + 0.25 = 2.25_{10}$

Conversion décimale-binaire

- Ecrire le nombre comme la somme des puissances de 2

$$0.8125_{10} = 0.5 + 0.25 + 0.0625 = 2^{-1} + 2^{-2} + 2^{-4} = 0.1101_2$$

- Algorithme: Multiplier à plusieurs reprises la fraction par deux jusqu'à ce que la fraction devient nulle.

$0.8125 \rightarrow 1.625$
$0.625 \rightarrow 1.25$
$0.25 \rightarrow 0.5$
$0.5 \rightarrow 1.0$

Représentation en virgule fixe

Représentation en Complément à 2

$$x = -2^m S + \sum_{i=-n}^{m-1} b_i 2^i$$

Exemple1:

0	1	0	1	0	0	1	1

$= 2^6 + 2^4 + 2^1 + 2^0$=64+16+2+1=83

Exemple2:

1	0	1	0	1	1	0	0

$= -2^7 + 2^5 + 2^3 + 2^2$=-128+32+8+4=-84

$2^{N-1} - 1$

Nombre **positif**: codé comme un binaire naturel

Nombre **négatif**: Inversion des bits puis ajout de 1

-2^{N-1}

Nombre	Codage		
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0
-1	1	1	1
-2	1	1	0
-3	1	0	1
-4	1	0	0

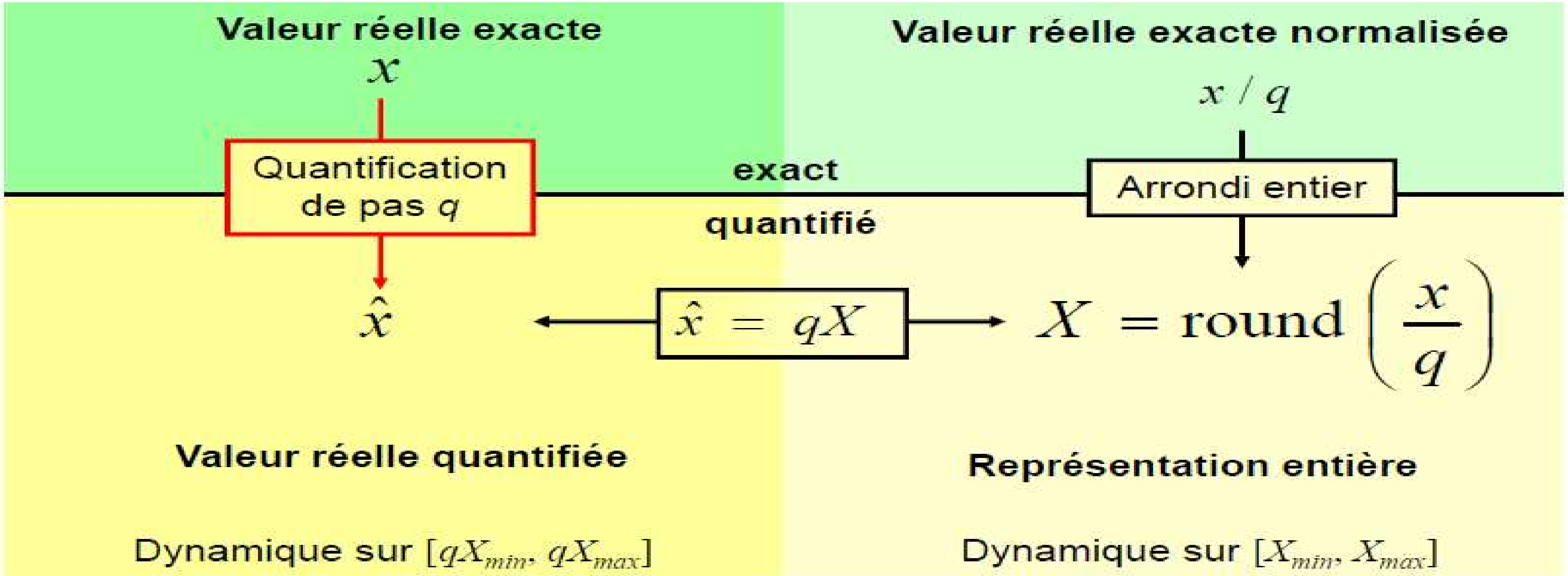
Signe

Représentation en virgule fixe

Calcul à Virgule fixe: codage des réels

- Format a virgule fixe : le format Qk

Attention à l'ambigüité entre $\hat{x}$ et X



Représentation en virgule fixe

·Aussi appelée représentation en “format fixe”

Définition ″format Q_k″ :
La représentation Q_k du réel x correspond à la représentation complément à 2 (C2) de l'entier y *tel que:*

$$y = \mathrm{round}(2^k x)$$

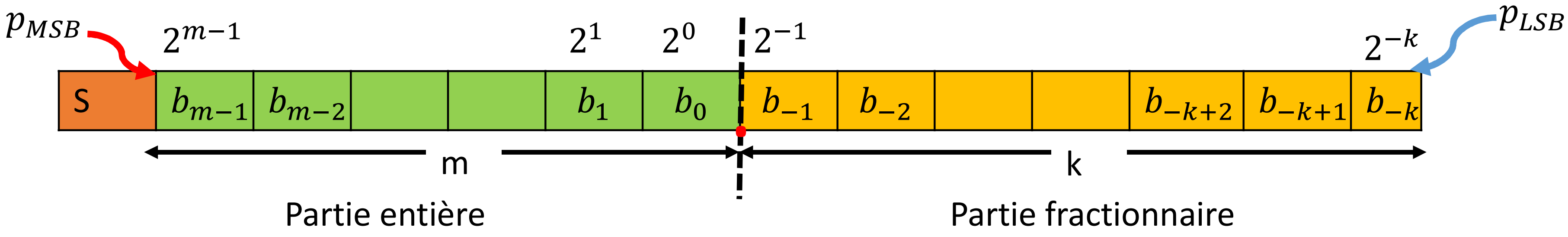
Propriétés:

- Partie fractionnaire codée sur k bits
- Partie entière codée sur N-k bits en C2 (dont 1 bit de signe dans le cas des nombres signés)

$\boldsymbol{Q_0}$ désigne le cas particuler des entiers signés en C2

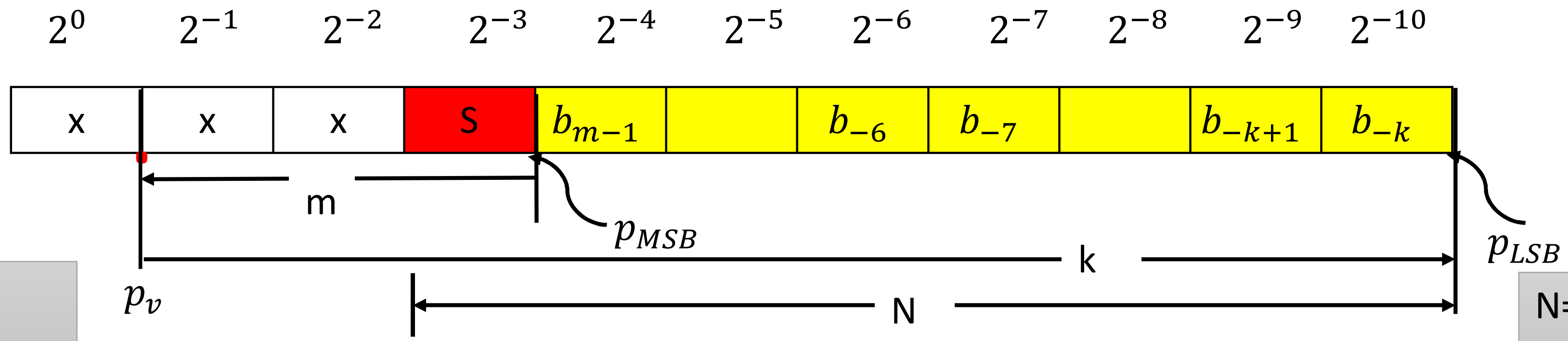
·Exemple: Le binaire 01011101 peut représenter :
Q_2 : 23.25 Q_4 : 5.8125 Q_7 : 0.7265625

Représentation en virgule fixe



- m: distance (en nombre de bits) entre la position du bit le plus significatif p_{MSB} et la position de la virgule p_v
- K: distance entre la position de la virgule p_v et la position du bit le moins significatif p_{LSB}

N=m+k+1 bits
Format: (N,m,k)



Ex:0.09472 (8,-3,10)
01100001
$0.0625+0.03125+2^{-10}$

N=8
m=-3
k=10

Représentation en virgule fixe

Exercice d'application:

Donnez la représentation en virgule fixe des formats (N,m,k) suivants: (5, 7, -3); (6,-2,7); (9, 0, 8); (7, 3, 3)

- D'autres notations de la représentation virgule fixe sont valables: $Q_{m.k}$ ou Q_k^N
- Pour passer d'un nombre décimal fractionnaire à son equivalent en virgule fixe au format Qm.k (en complément à 2) il faut multiplier ce nombre par 2^k, la partie entière du résultat (appelée valeur entière équivalente VEE) correspond au codage de la partie fractionnaire .

Exemple: Représentez le nombre 0.875 en format Q0.15 signé

Réponse: VEE= 0.875 x 2^{15}=28672 =2^{14}+2^{13}+2^{12}; Ainsi 0.875= 2^{-1}+2^{-2}+2^{-3};

la représentation en format Q0.15= 0111000000000000

Représentation en virgule fixe

On définit:

–$\boldsymbol{N_c}$ le nombre de valeurs représentables du codage.

–$\boldsymbol{D_R}$ Le domaine de définition correspond à l'intervalle regroupant l'ensemble des valeurs représentables par le codage. Les bornes min et max de cet intervalle sont respectivement X_{min} et X_{max}.

–La dynamique d'un codage **D** représente la différence entre la valeur minimale et maximale :$D = X_{max} - X_{min}$

$$D_{N(dB)} = 20.\log\left(\frac{\max(|x|)}{\min(|x|)}\right):$$

– La résolution **R** d'un codage représente la somme entre la valeur minimale et la valeur maximale: R= $Xmax + Xmin$

Représentation en virgule fixe

❑ Pour **la représentation complément à2** on a:

- $N_c = 2^N - 1$
- $D_R = [X_{min}; X_{max}] = [-2^m ; 2^m - 2^{-k}] = [-2^{N-1-k} ; \ 2^{N-1-k} - 2^{-k}]$
- $D = X_{max} - X_{min} = 2^{m+1} - 2^{-k}$
- Le pas de quantification $q = D/Nc = 2^{-k}$

q est la **précision**: intervalle entre 2 valeurs Quantifiés adjacentes

- Erreur maximale: $q/2 = 2^{-(k+1)}$

Exemples: Pour obtenir une dynamique sur l'intervalle [−1, 1[,utiliser le format Q_{N-1} sur N bits

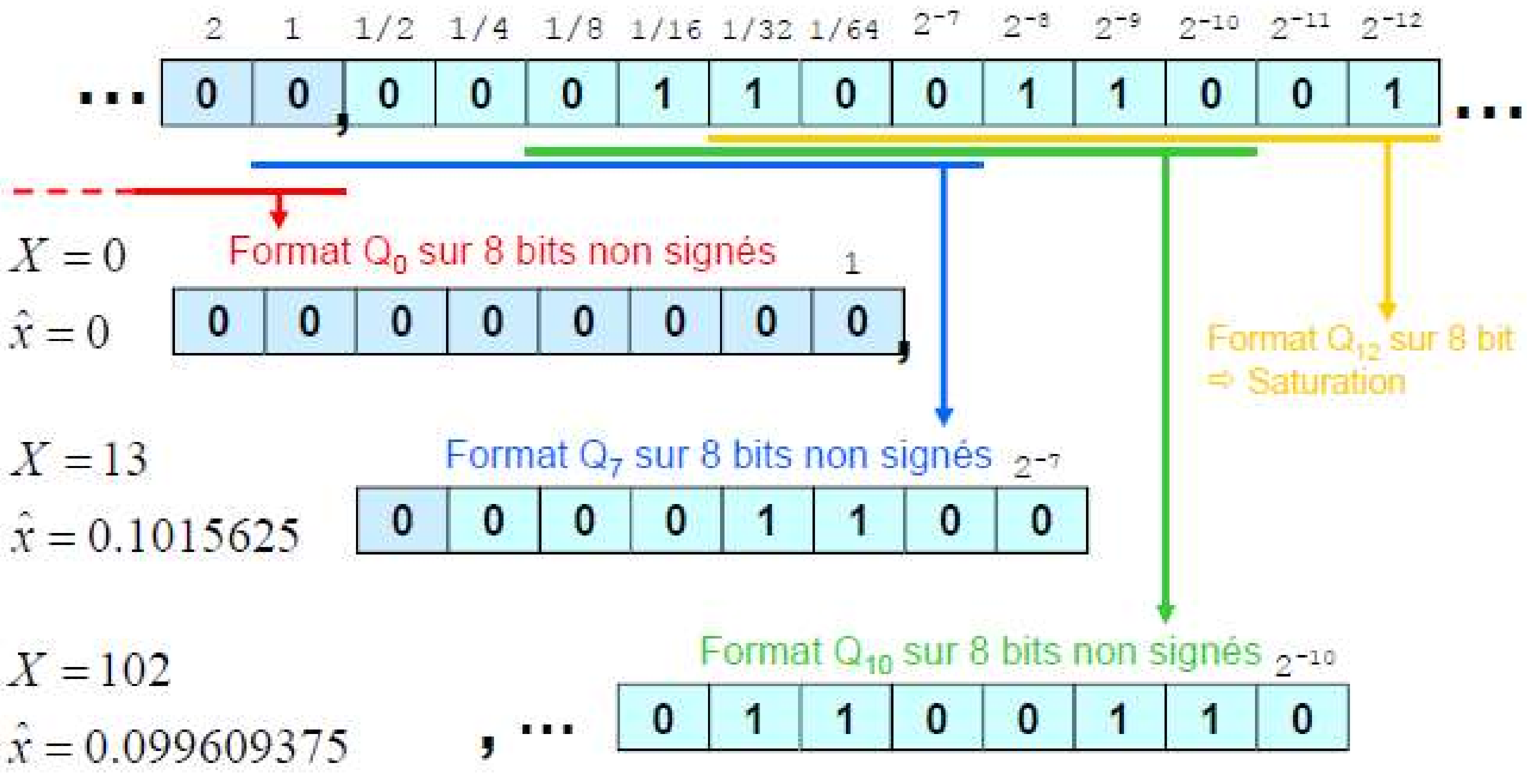
Format Q_{15}: sur 16 bits, le format Q_{15} permet de représenter tous les réels entre −1 et 1 avec une précision $q=2^{-15}$

Exemple de codage (6,3,2)

0111.11	7.75	
0111.10	7.5	
0000.11	0.75	
0000.10	0.5	
0000.01	0.25	
0000.00	0	
1111.11	-0.25	
1111.10	-0.5	0.5 − 8+7. 5
1111.01	-0.75	
1000.01	-7.75	-7.75 = -8+0.25
1000.00	-8	

Représentation en virgule fixe

- Exemple : Représentation Q_k non signée de x=0.1



Représentation en virgule fixe

❑ Choix du format Q_k

Pour le codage d'un ensemble de valeurs **x** en virgule fixe sur **N** bits signés, la méthode est la suivante:

1) Expliciter la dynamique prévue pour les valeurs de x

$$|x| \leq x_{max}$$

2) Estimer k_{max}, le k maximum permettant d'éviter la saturation, c'est-à-dire vérifiant:

$$k_{max} = \lfloor -\log_2(x_{max}) + N - 1 \rfloor$$

3) Toutes les valeurs k inférieures à k_{max} permettent d'éviter la saturation. On choisit la valeur permettant la plus grande précision, c'est à dire k_{max}.

4) Le pas de quantification est alors: $q = 2^{-k_{max}}$

Dans le cas non signé, on a $x_{max} \leq 2^{N-k_{max}}$ $k_{max} = \lfloor -\log_2(x_{max}) + N \rfloor$

Représentation en virgule fixe

- Exemple: Q_5 sur 8 bits

- Partie entière codé sur 3 bits (dont 1 de signe)
- Partie fractionnaire codée sur 5 bits

- Valeurs comprises entre −4 et +3.96875

#/Poids	-2^3	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
-3.96875	1	0	0	0	0	0	0	1
-4	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
3.9375	0	1	1	1	1	1	1	0
3.96875	0	1	1	1	1	1	1	1

Représentation en virgule fixe

Représentation Qk sur 16 bits

k	Précision q	Dynamique pour 16 bits signés	Dynamique pour 16 bits non signés
-1	2	[-65536, 65535]	[0, 2^{16}-1]
0	**1**	**[-32768, 32767]**	**[0, 65535]**
1	1/2	[-16384, 16383]	[0, 32767]
14	2^{-14}	[-2, 2[	[0, 4[
15	2^{-15}	**[-1, 1[**	[0, 2[
16	2^{-16}	[-0.5, 0.5[	**[0, 1[**
17	2^{-17}	[-0.25, 0.25[	[0, 0.5[

Entiers Q0

[-1 1[Q15 sur 16 bits signés

[0 1[Q16 sur 16 bits non signés

Représentation en virgule fixe

❑ *Cadrage à gauche ou à droite*

Deux représentations particulières liées à la position de la virgule sont couramment utilisées:

- **Cadrage à Droite:** Lorsque la virgule est cadrée à droite la valeur codée est entière
- **Cadrage à Gauche**: Lorsque celle-ci est cadrée à gauche la donnée est fractionnaire.

Les caractéristiques de ces deux représentations sont présentées dans les tableaux suivants:

Représentation	Cadrage à gauche	Cadrage à droite	cadrage
Condition	m=0	k=0	k+m=N-1
Q	$2^{-(N-1)}$	1	2^{-k}
D	[-1;1-q]	$[-2^{N-1}; 2^{N-1} - q]$	$[-2^{m}; 2^{m} - q]$

Représentation CA2

Chapitre 1

1. Généralité
2. Représentation en virgule fixe
3. Codage en virgule flottante
4. Codage Virgule fixe Vs codage virgule flottante

Codage en virgule Flottante

- Aussi appelée représentation en "virgule flottante" est une représentation avec une précision finie, définie selon l'expression

$$x = (-1)^s M \times 2^E$$

- La **mantisse** M est exprimée sur m bits, avec un format Q_{m-1} en complément à 2 → détermine le nombre de chiffres significatifs

- L'**exposant** $\boldsymbol{E}$ est un entier signé sur $\boldsymbol{e}$ bits en binaire décalé → détermine la dynamique

$$1 + m + e = N$$

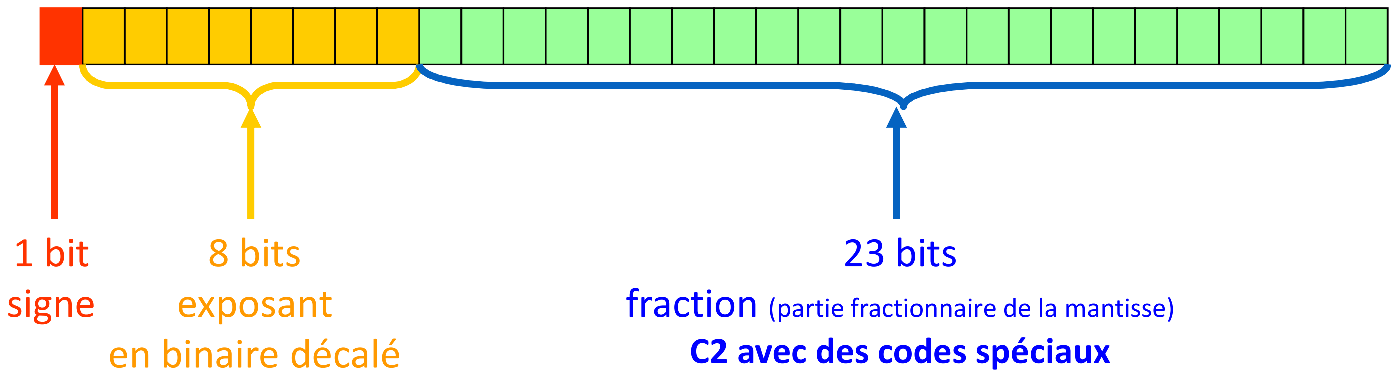
- Opération de normalisation:
  Pour rendre la représentation unique, M doit satisfaire en outre:

$$1 \leq |M| < 2$$

Codage en virgule Flottante

❑ Format virgule flottante IEEE 754

- *Format pour N=32 bits (float)*



$$x = (-1)^{signe} \times (1,fraction)_2 \times 2^{exposant-127}$$

- **Cas spéciaux**

  Zéro: tous les bits à 0

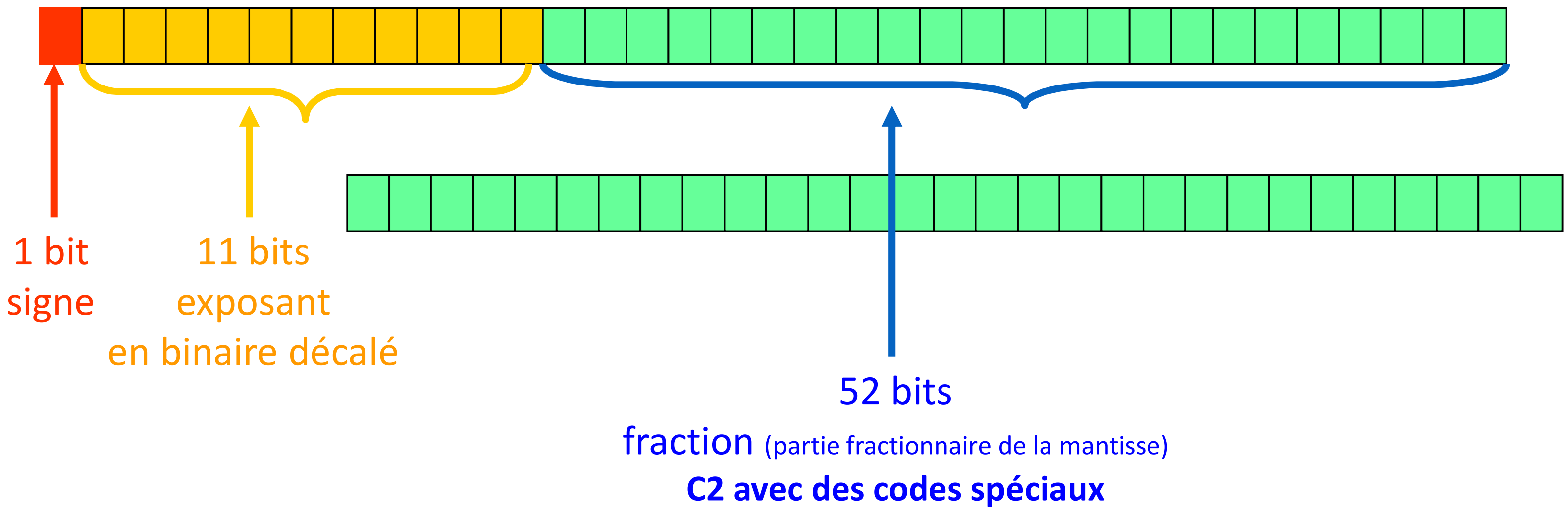
  Underflow: exposant = 00000000_2

  Overflow: exposant = 11111111_2

Codage en virgule Flottante

Format virgule flottante IEEE 754

- Format pour N=64 bits (double)



$$x = (-1)^{signe} \times (1{,}fraction)_2 \times 2^{exposant-1023}$$

Chapitre 1

1. Généralité
2. Représentation en virgule fixe
3. Codage en virgule flottante
4. Codage Virgule fixe Vs codage virgule flottante

Codage en virgule Flottante

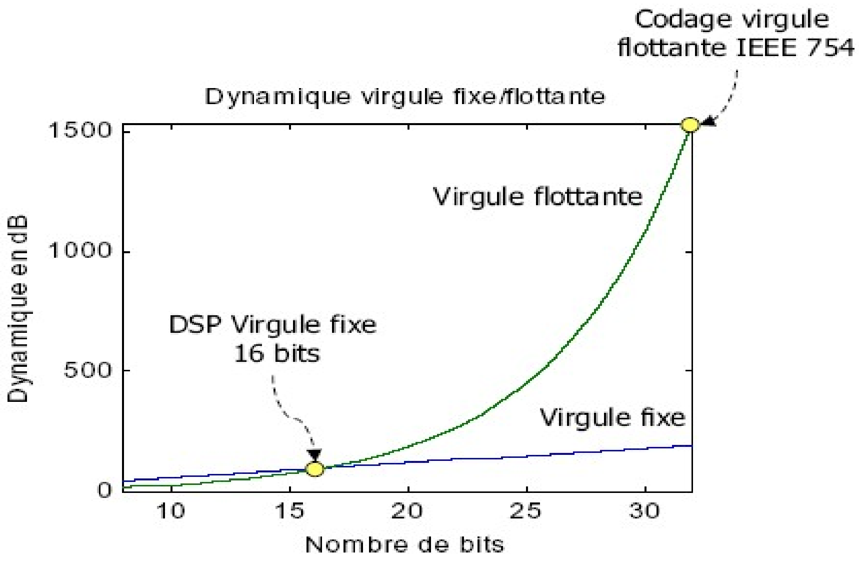
Analyse de la dynamique

$$D_{N(dB)} = 20.\log\left(\frac{\max(|x|)}{\min(|x|)}\right):$$

- **Virgule fixe:** Le niveau de dynamique exprimé en *dB* du codage en virgule fixe est linéaire par rapport au nombre de bits *N* utilisés par le codage
- **Virgule flottante:** Le niveau de dynamique pour une représentation en virgule flottante est fonction du nombre de bits *E* alloués pour l'exposant :

$$D_{N\ dB} = 20\log(2^{2K+1}) \quad \text{avec} \quad K = 2^{E-1} - 1$$

Pour cette courbe: codage en virgule flottante, la taille de l'exposant = 1/4 de la longueur totale.



Codage en virgule Flottante

Analyse du RSB

$$\rho_{dB} = 10.\,log_{10}\left(\frac{P_{signal}}{P_{bruit}}\right)$$

- Pour les signaux de dynamique faible, très sensibles à l'erreur de quantification, la représentation en virgule flottante permet d'obtenir un meilleur RSB.
- Pour des signaux de dynamique élevée et pour N identique, le RSB du codage en virgule fixe est supérieur à celui en virgule flottante.