DIXIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

ETUDE ET CONCEPTION D'UN ENSEMBLE DE CIRCUITS VLSI EN VUE DE LA REALISATION D'UN COMPOSANT DE NORMALISATION

C. BOZZO G.BUENO M.DUMAS

CSEE - CETIA 150 Av. Marcelin Berthelot TOULON EST 83088 TOULON CEDEX

RESUME

RESUME :

Nous presentons brièvement ici une approche originale du problème de la normalisation d'un bruit additif à un signal de façon à en améliorer la détection.

L'originalité du travail n'est pas dans l'algorithme choisi, mais dans l'approche matérielle de l'opérateur.

Nous espérons ainsi accroître de façon déterminante le rapport coût/performance.

SUMMARY

ABSTRACT :

We present here an original approach of the normalization of the noise in order to make the signal detection more accurate.

The originality of the work is not in the algorithm but in the hardware approach of the operator.

Using this way, we hope to increase the effectiveness and decrease the cost in term of time consuming.

ETUDE ET CONCEPTION D'UN ENSEMBLE DE CIRCUITS VLSI EN VUE DE LA REALISATION D'UN COMPOSANT DE NORMALISATION

1. - INTRODUCTION

Le problème de la normalisation du bruit additif à un signal est sous jacent à toute détection. En effet, les principaux algorithmes supposent le bruit gaussien et surtout stationnaire. Or, il est bien connu que ces deux hypothèses sont rarement vérifiées. Si dans certains cas on peut, en première approximation, supposer qu'elles le sont, il est parfois difficile de mettre en place une procédure de détection valable si un premier traitement n'a pas permis de se placer dans un bon voisinage de ces hypothèses. Notre propos, dans cette communication, n'est pas d'étudier tous les algorithmes de normalisation possibles, mais de montrer que cette "opération" telle qu'elle est faite aujourd'hui, peut être abordée sous l'angle d'une réalisation matérielle, sous forme d'un ensemble de circuits VLSI.
Nous avons été amenés à faire quelques hypothèses sur les différentes variables traitées. Compte tenu de l'avancement des travaux, ces hypothèses sont encore du domaine du paramètre et n'ont servi qu'à fixer les limites de "l'épure". On trouve donc, outre cette introduction, quatre chapitres. Le chapitre 2 précise l'approche algorithmique et la configuration globale de l'opérateur.
Nous présentons les principaux algorithmes utilisés, après avoir précisé le type de signaux et la simulation qui nous a permis de les évaluer. A partir de cette étude, relativement succincte, l'algorithme étant choisi, nous analysons le schéma-bloc de l'algorithme sous l'angle de la réalisation VLSI.
On est à même d'isoler des fonctionnalités de base qui sont étudiées au plan du principe et des schémas logiques. Cette analyse est faite dans le chapitre 2.
Une fois le schéma logique obtenu, la simulation des opérateurs est faite à l'aide du logiciel TEGAS 5. Cette simulation a permis d'analyser et de mettre en évidence les problèmes de séquencement, de conflit et de donner une évaluation concrète de la complexité et des performances que l'on peut espérer atteindre. Nous concluons ce chapitre 3 par l'évaluation en nombre de portes et la technologie qui nous paraît la plus appropriée pour une telle réalisation.
La conclusion reprend les principales idées développées dans ce travail et met en évidence les points forts et les lacunes de cette approche "matérielle" de l'algorithme.

2. - APPROCHE ALGORITHMIQUE

2.1 - Introduction

Nous présentons dans ce chapitre les différents algorithmes que nous avons étudiés. Toutefois, notre propos n'étant pas à proprement parler l'étude d'un algorithme nouveau, nous avons repris ceux qui nous paraissent les plus utilisés et surtout les plus robustes bien que, peut-être, largement sous-optimaux. On verra toutefois que, compte tenu de l'approche modulaire adoptée, cette restriction n'en est, en fait, pas une et l'évolution possible a été prévue au niveau du matériel (dans certaines limites toutefois). La validation de ces méthodes sera faite sur une simulation d'écho sonar entaché d'un bruit additif particulièrement non stationnaire. Nous avons, par ailleurs, défini une variable qui permettra de juger qualitativement du résultat obtenu.

2.2 Simulation

Il s'agissait de simuler un signal largement non stationnaire de façon à se placer d'emblée dans des circonstances se rapprochant le plus de l'environnement réel.

Dans le cas d'un signal SONAR, le bruit dit "de réverbération" répond de façon assez satisfaisante aux critères ci-dessus. La simulation de ce type de signaux a été faite en intégrant les notions :

- de réflexion d'un écho sur une cible,
- de réflexion non stationnaire des bruits additifs.

Ceci nous conduit à :

- la génération d'un écho sous la forme d'une superposition de fonction :

$$S_i(t) = \frac{\sin(\pi w_i t)}{\pi w_i t}$$

  décalée dans le temps,
- la génération par le calculateur d'une séquence gaussienne pseudo-blanche que l'on déstationnarisera par une fonction exponentielle décroissante.

La superposition de ces deux signaux a conduit à la simulation voulue.

Par ailleurs, nous avons calculé un rapport que l'on peut assimiler au rapport signal/bruit.
Ce rapport est obtenu par :

$$R = \frac{\text{moyenne de l'amplitude des pics de l'écho}}{\text{écart type du bruit}}$$

L'écart type du bruit est calculé en se donnant le rapport R et l'amplitude des différents pics de l'écho simulé. Il s'agit, bien entendu, du bruit avant déstationnarisation. Cette variable R permettra de tester la qualité des différents algorithmes. La figure 1 montre le résultat de cette simulation dans deux cas bien précis (pour un R = 20 dB) :
- signal lointain,
- signal proche.

2.3 - Présentation des algorithmes

2.3.1 - Introduction

Les algorithmes de normalisation étudiés reposent tous sur des estimations de la moyenne et du bruit au voisinage de chaque échantillon. En effet, pour chaque échantillon x_i il s'agit de calculer la valeur X_i telle que :

$$X_i = \frac{x_i - \hat{m}_i}{\hat{\sigma}_i}$$

$\hat{m}_i$ = estimation de la moyenne autour de x_i

$\hat{\sigma}_i$ = estimation de l'écart type du bruit autour de x_i

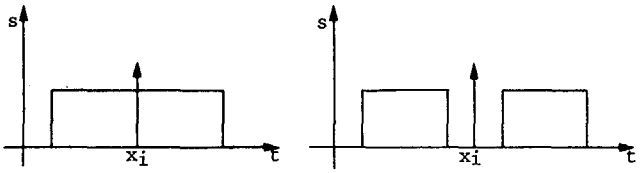
Les méthodes pour calculer $\hat{m}_i$ et $\hat{\sigma}_i$ pourraient s'appuyer sur des résultats de filtrages bien connus en traitement du signal. Nous avons préféré nous reposer sur des méthodes plus pragmatiques mais dont les résultats sont bien maîtrisés. Nous avons suivi en cela l'article de LEFAUDEUX (1).
On trouvera aussi dans (2) une approche relativement originale de la normalisation. On verra d'ailleurs que la méthode choisie possède l'avantage de laisser la porte ouverte à une investigation ultérieure.

2.3.2 - Les différents algorithmes

Tout le problème consiste donc à trouver des estimateurs fiables de la moyenne et de l'écart type autour d'un échantillon de signal. Pour cela, on définit au moyen d'une fenêtre de pondération, un voisinage de l'échantillon à normaliser. Cette fenêtre "glisse" ensuite sur toute la longueur du signal.

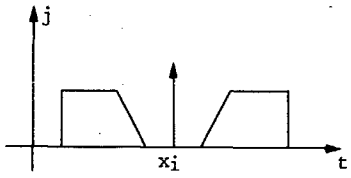
Sur cette fenêtre, les calculs de la moyenne et de l'écart type peuvent se faire de différentes façons :
- en une passe,
- en deux passes.

(cf. LEFAUDEUX (1))
Par ailleurs, un problème peut se poser si on ne prévoit pas un moyen pour éliminer l'influence du signal sur ces calculs.
On propose donc, bien souvent, de réaliser un "trou" autour de l'échantillon à normaliser.
Les fenêtres ont donc les formes suivantes :



fenêtre simple — fenêtre double

En ce qui concerne l'aspect rectangulaire de la fenêtre, on sait les problèmes que cela pose ; nous avons donc essayé des fenêtres à formes diverses en restant toutefois relativement simple. Nous proposons ici une fenêtre trapézoïdale autour de x_i :



fenêtre trapézoïdale

En ce qui concerne le calcul de l'estimation de la moyenne et de l'écart type, nous éliminerons dans chaque demi-fenêtre un certain nombre d'échantillons extrêmes qui risquent, par leurs seules présences, de perturber la qualité des estimations obtenues.

2.3.3 - Présentation des résultats

Les figures 2 et 3 montrent les résultats obtenus suivant :
- la position de l'écho (proche ou lointain),
- le rapport signal/bruit tel qu'il a été défini plus haut (pour le cas des fenêtres doubles à deux passes).

Nous avons constaté que si les algorithmes à deux passes l'emportent dans tous les cas, il paraît indispensable de prévoir un "trou" autour de l'échantillon à normaliser.
Par ailleurs, on peut espérer une amélioration des résultats en modifiant la forme de la fenêtre (coefficients de pondération).
L'estimateur, dans le cas rectangulaire, est donné par les formules suivantes, pour chaque demi-fenêtre :

$$\text{①}\quad \begin{cases} \sigma_A^2 = \dfrac{1}{NT} \displaystyle\sum_{j=1}^{NT} x_j^2 - m_A^2 \\ m_A = \dfrac{1}{NT} \displaystyle\sum_{j=1}^{NT} x_j \end{cases}$$

NT = Nombre d'échantillons sélectionnés

2.4 - Schéma-bloc et premières considérations de réalisations

L'algorithme ayant été maintenant présenté, nous allons en déduire le schéma-bloc qui va nous conduire à la définition de fonctionnalités de bases qui serviront à l'élaboration d'un composant de normalisation sous la forme d'un ensemble de circuits VLSI.

2.4.1 - Schéma-bloc

Compte tenu des équations ① le schéma bloc est donné à la figure 4.
On remarquera principalement :
- le parallélisme des traitements, au moins jusqu'au calcul de la moyenne et de la variance ;
- la répétitivité des fonctions de base.

On distingue, comme opération spécifique de la normalisation :
- le sélecteur : cet opérateur retient, parmi le flot de données qui le traverse, les n valeurs les plus grandes ;
- le moyenneur : cet opérateur calcule la moyenne des valeurs en sortie du sélecteur ;
- un élévateur au carré ;
- un opérateur de demi-somme.

Les opérations qui ne seront pas abordées ici concernent principalement le diviseur et l'extracteur de racine carrée, qui, dans un premier temps peuvent être réalisés par ailleurs.
Il faut noter, à ce propos, que cet opérateur doit être vu comme un co-processeur qui permet à l'unité centrale de se décharger de l'opération de normalisation dans le cadre d'un processus de traitement du signal programmé, par exemple, sur microprocesseur.

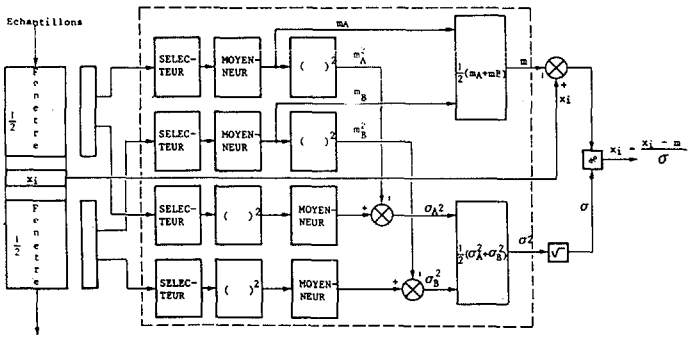


Figure 4 - Schéma global

2.4.2 - Format des données et prévision

Compte tenu de l'environnement général de ce type de machine, nous avons été amenés à faire les hypothèses suivantes :
- les données à traiter sont codées sur 16 bits,
- il s'agit de nombres entiers positifs.

Des résultats en simulation semblent montrer que ces deux hypothèses ne nuisent pas au bon fonctionnement des algorithmes (pour une plage d'utilisation donnée, bien entendu).
En ce qui concerne le moyenneur, de façon à ne pas fermer le système au plan algorithmique, nous faisons l'hypothèse que le nombre d'échantillons peut varier de 10 à 40.
Ceci nous permettra de connaître, pour une précision sur le calcul final donnée, et compte tenu du format des données, le nombre de bits nécessaires au codage des coefficients au calcul de la moyenne (on rappelle que ces coefficients sont toujours inférieurs à 1)

n \ p	8	9	10	11	12	13
10	2,34	0,39			0,14	
15		0,39			0,02	
20		2,34	0,39		0,39	
25			2,34	1,12	0,51	
30					0,39	
35					0,02	
40					0,39	

P : nbre de bits de codage pour les coefficients
n : nombre d'échantillons

On constate donc que la précision sur le calcul de la moyenne est toujours à 1 % si les coefficients sont codés sur 12 bits.
On peut montrer de façon relativement simple qu'en fait, compte tenu des nombres à représenter, seuls 9 bits sont nécessaires.

2.5 - Conclusion

Une fois les formules de calcul nécessaires à la normalisation du signal connues, nous nous sommes attachés à définir une architecture de machine dédiée à la normalisation.
Comme nous l'avons déjà dit, cette machine doit être considérée comme un co-processeur qui tire parti du parallélisme des traitements et de la simulation des processus utilisés.
La pile "FIFO" en entrée ainsi que les "aiguillages" ne sont pas inclus dans la machine (du moins dans cette première phase de l'étude) ainsi que les opérateurs de division et d'extraction de racine carrée.
Les pointillés de la figure 4 limitent notre champ d'investigation.
On rappelle pour mémoire que :
- les données sont codées sur 16 bits,
- les coefficients de moyennes sont codés sur 12 bits dont 9 sont pratiquement utilisés.

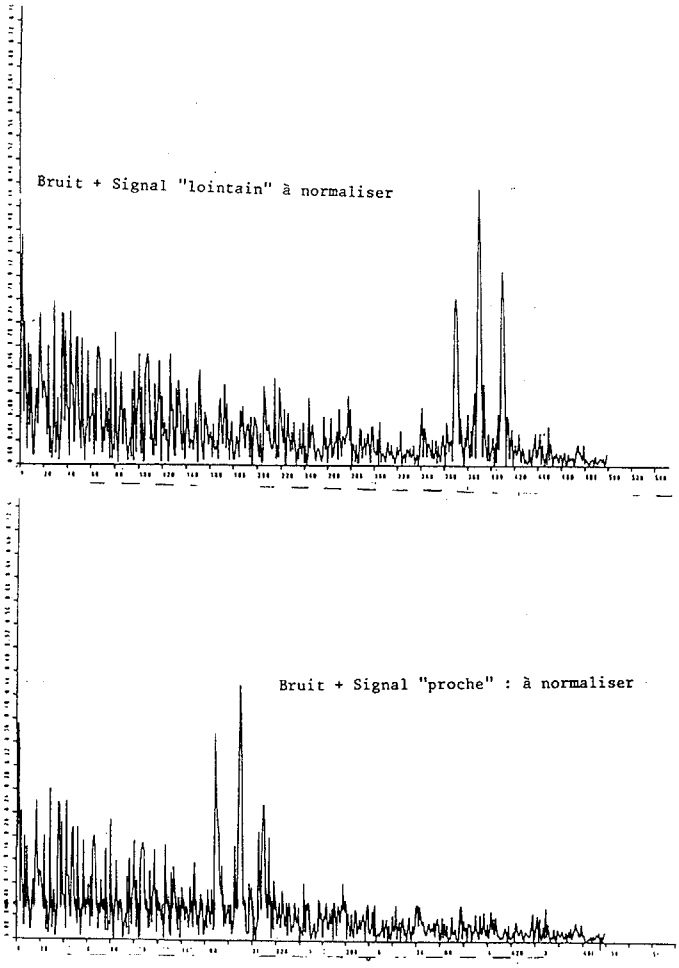


Figure 1 - Résultat de la simulation

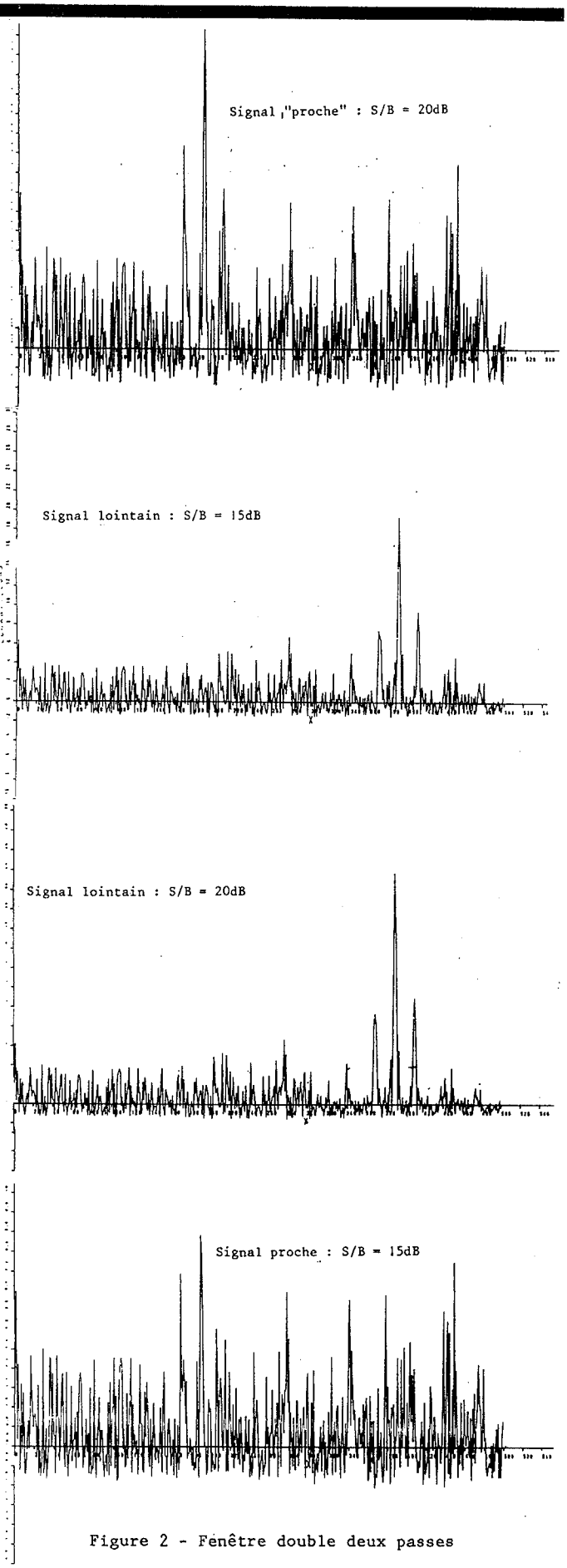


Figure 2 - Fenêtre double deux passes

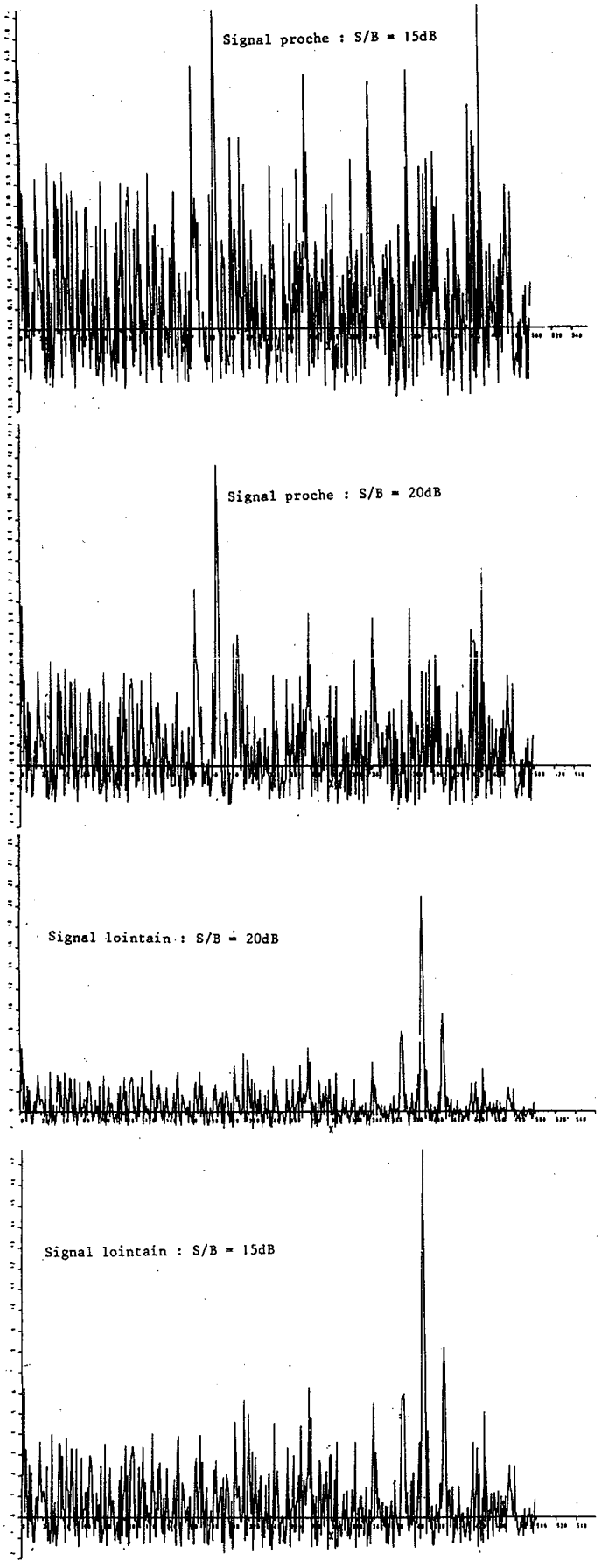


Figure 3 - Fenêtre double trapézoïdale deux passes

3 - DEVELOPPEMENT MATERIEL DES MODULES SPECIFIQUES

- Le sélecteur
- Le moyenneur.

3.1 - Introduction

L'architecture du composant, telle qu'elle est présentée, permet de constater un double parallélisme des traitements :
- parallélisme au niveau du calcul de la moyenne et de la variance,
- parallélisme au sein de chaque calcul.

L'aspect synchrone des résultats est assuré par la similarité des traitements dans chaque "pipe-line". Ainsi le problème du cadencement est résolu de lui-même.

On peut toutefois envisager, dans le cas d'un calcul de la moyenne avec des coefficients variables (fenêtre trapézoïdale par exemple), un automate qui permettrait d'assurer la prise en compte de ces grandeurs par les multiplieurs.

3.2 - Sélecteur

Cet organe (fig. 5) a pour but d'extraire de la suite des échantillons x_i l'échantillon le plus grand.
Il travaille sur des mots de 16 bits et se compose :
- du registre d'entrée REGØ,
- du registre de sauvegarde REG1 de l'échantillon le plus grand détecté jusqu'alors,
- du comparateur entre l'échantillon courant et l'échantillon le plus grand,
- du multiplexeur de choix entre le mot courant et le plus grand échantillon suivant le sens de la comparaison,
- du registre de sortie.

Utilisation du sélecteur :

Il est clair que si l'on veut évincer d'une suite les N plus grands échantillons, il est nécessaire de chaîner N sélecteurs (fig. 6).
A noter que la cohérence temporelle n'est pas conservée lors d'un tel traitement.

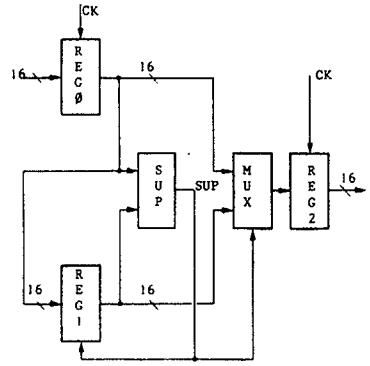


Figure 5 - Sélecteur

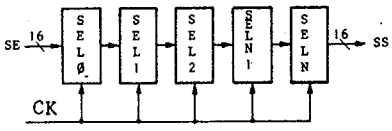


Figure 6 - Filtrage des N plus grandes valeurs d'échantillons

Légende : CK = Signal d'horloge d'échantillonnage
SE = Séquence d'entrée
SS = Séquence de sortie

3.3 - Moyenneur

Ce module (fig. 7) permet de réaliser la moyenne arithmétique pondérée sur les échantillons x_i préalablement sélectionnés.
Il faut donc calculer $A_i.X_i$ où A_i et X_i sont des entiers positifs codés respectivement sur
- A_i : 9 bits (chapitre 2.4.2)
- X_i : 16 bits .

Le moyenneur développé ici s'est voulu simple, facilement implantable en VLSI.
La solution retenue ici utilise des cellules de calcul élémentaires systoliques série (3) (fig. 8) qui sont cascadées pour réaliser ensemble le multiplieur base de l'outil (fig. 9).
Nous avons retenu la découpe fonctionnelle suivante:
- registre d'entrée REGB :
ce registre 16 bits à chargement parallèle, sortie série accueille la suite des échantillons X_i à moyenner.
- Multiplieur série :
ce multiplieur série systolique utilise une batterie de 9 multiplieurs élémentaires (fig. 10) ; ce nombre de multiplieur est imposé par la taille des coefficients multiplicateurs A_i dont la valeur est fonction de la fenêtre utilisée et précharges dans le registre parallèle REGA.
- Registre d'accumulation REGM :
c'est ce registre qui contient le résultat de la moyenne ainsi effectuée ; sa taille est déterminée par le nombre d'échantillons maximum à traiter.
Dans les conditions de l'application nous nous sommes permis d'aller jusqu'à 40 accumulations successives et par là même de fixer à 32 bits la taille de ce registre.
Le résultat de la moyenne pondérée se trouve alors sur les 16 bits de poids forts du registre.
Remarque :
L'opération de multiplication s'effectuant sur 32 bits il est nécessaire de fournir les données à moyenner sur 32 bits, pour cela tous les bits série sortant du registre d'entrée dont le poids est supérieur à 15 sont fixés à zéro.

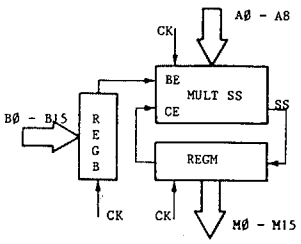


Figure 7 - Moyenneur

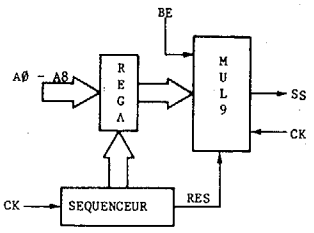


Figure 10 - MULT SS

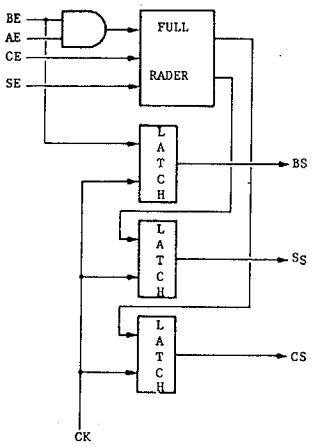


Figure 8 - MUL Ø

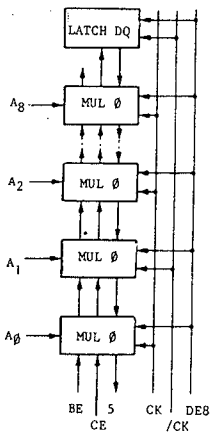


Figure 9 - MUL 9

3.4 - Spécifications

L'état d'avancement de l'étude a permis de réaliser les fonctions du sélecteur et du moyenneur.
Le décompte des portes, pour chaque réalisation, est :
- sélecteur : 200 portes,
- moyenneur : 800 portes,
- automate de prise en compte des coefficients A_i : 100 portes.

La fréquence maximale d'utilisation, compte tenu de la technologie utilisée, a été fixée à 25 MHz.
Pour une simplification des tâches de développement VLSI cette réalisation s'est faite dans une technologie prédiffusée CMOS 3 μm.

3.5 - Performances

Compte tenu de l'architecture adoptée (cf fig. 4) il n'y a pas accumulation des temps de calcul entre la moyenne et la variance, et, au sein de chacune d'elle, le parallélisme des traitements permet d'envisager, là aussi, des performances intéressantes.
En ce qui concerne le calcul de la moyenne, compte tenu de l'architecture du module, de la technologie utilisée, les simulations ont permis d'évaluer le temps d'exécution à 60 μ s pour effectuer la moyenne de 40 échantillons à une fréquence maximale (25 MHz), avec une vitesse de "pipe-line" de l'ordre de 1,3 μs, par échantillon.

4. - CONCLUSION

On rappelle les principales caractéristiques du composant :
- coprocesseur spécialisé,
- architecture parallèle et répétitive,
- fonction synchrone par les données,
- architecture interne des composants simples et facilement intégrables.

A noter que les fonctionnalités isolées peuvent trouver une application dans d'autres domaines, ce qui assure l'ouverture du système à d'autres applications.
Il est intéressant de noter qu'avec une telle approche, et bien que les multiplications systoliques série présentent un manque d'efficacité intrinsèque, nous pouvons améliorer d'un facteur 3 à 4 les opérations de normalisation.
En effet, lors d'un calcul de la moyenne sur 40 échantillons, une approche programmée nécessiterait près de 320 μs (4 X 80 μ s) alors que tout le traitement pipe-line ne demanderait que 60 μs.
On peut toutefois souligner la difficulté de réalisation de certains opérateurs (extracteur de racine carrée) sans engager des coûts de développement prohibitifs.

Remerciements

Nous tenons à remercier Messieurs AGRESTA, VIDAL, et VISCONTI pour leur collaboration lors de ce travail.

BIBLIOGRAPHIE

1. - M. LEFAUDEUX
CERSDM, LE BRUSC
ALGORITHME PRAGMATIQUE DE NORMALISATION
8ème colloque GRETSI 1981

2. - C. PLUMEJEAND, B. RAFINE et B. LUCAS
CERDSM, LE BRUSC
UN ALGORITHME DE NORMALISATION
9ème colloque GRETSI 1983

3. - C.BOZZO , M.FOUQUES et E.SIFFREDI
CSEE/CETIA, LA VALETTE DU VAR
A REAL TIME 2 - LEVEL SYSTOLIC 2D CONVOLUTION CHIP
22nd Annual ALLERTON Conference 1984