Calcul cablé d'une transformée de Fourier à très grand nombre d'échantillons, éventuellement multi-dimensionnelle

A. Vacher

▶ **To cite this version:**

A. Vacher. Calcul cablé d'une transformée de Fourier à très grand nombre d'échantillons, éventuellement multi-dimensionnelle. Micro et nanotechnologies/Microélectronique. Institut National Polytechnique de Grenoble - INPG, 1997. Français. NNT : . tel-00010763

HAL Id: tel-00010763
https://theses.hal.science/tel-00010763

Submitted on 26 Oct 2005

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.

THÈSE

présentée par

André VACHER

pour obtenir le grade de **DOCTEUR**

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 mars 1992)

(Spécialité : Microélectronique)

Calcul cablé d'une Transformée de Fourier à trés grand nombre d'échantillons, éventuellement multi-dimensionnelle

---

Date de soutenance : 8 janvier 1997

Composition du jury

Président : Guy MAZARÉ

Rapporteurs : Nicolas DEMASSIEUX

Habib MEHREZ

Examinateurs : Emmanuel BOUTILLON

Alain GUYOT

Joël LIÉNARD

Duc TRAN QUI

Thèse préparée au laboratoire TIMA

À mes parents.

Remerciements

Une thèse amène de multiples rencontres qui laissent des souvenirs trés divers. Lorsqu'un tel travail est soutenu en cinquième année d'inscription en thèse, la liste des rencontres est évidemment trés longue. Je ne vais pas écrire ce *who's who* en dessous du titre que vous venez de lire et que ceux qui auraient pu trouver leur nom ici ne m'en tienne pas rigueur. Je tiens toutefois à remercier :

Monsieur Bernard COURTOIS qui m'a accueilli dans le laboratoire qu'il dirige, il s'appelait alors TIM3, alors que j'arrivai avec mon sujet de thèse sous le bras,

Monsieur Guy MAZARÉ qui a accepté de présider le jury devant lequel cette thèse a été soutenue,

Messieurs Nicolas DEMASSIEUX et Habib MEHREZ qui ont bien voulu en être les rapporteurs

Messieurs Emmanuel BOUTILLON et Joël LIÉNARD qui ont lu attentivement mon manuscrit, m'ont fait part de leurs remarques et réactions avant de participer à ce jury,

Monsieur Duc TRAN QUI qui m'a inspiré le sujet de cette thèse au cours de nos conversations au temps où je travaillais dans son laboratoire et qui a consacré de nombreuses heures pour me présenter les principes et l'utilisation possible de la transformée de Fourier dans le domaine de la Cristallographie et a accepté de se joindre aux autres membres de ce jury,

Monsieur Alain GUYOT qui a bien voulu être le directeur de cette thèse et qui a toujours su conserver un oeil sur mes cheminements mêmes lointains,

le Réseau Doctoral en Architecture des Systèmes et des Machines Informatiques qui a financé un certain nombre de missions pour présenter mes travaux aux États-Unis, assister à des écoles de printemps et d'été en France et en Tunisie, participer à des journées diverses qui m'ont permis de rencontrer des thésards et des chercheurs d'horizons diverses et d'assumer des tâches qui jalonnent habituellement la vie d'un chercheur permanent (participation à des comités d'organisation et de lecture, etc ...),

Madame Lilianne PONTONNIER et Monsieur Philippe BRULARD sous les ordres desquels j'ai eu le plaisir d'enseigner la Physique Appliquée à l'Université,

Mesdames DE COOMAN et LOSSEAU grâce à qui j'ai pu participer à différentes manifestations scientifiques alors que l'appat du gain m'avait conduit à me trouver un métier qui puisse financer la danseuse qu'était devenue ma thèse

le personnel de TIMA que j'ai cotoyé pendant toutes ces années et que je ne pourrais entièrement nommer, commençons par tous ces visages féminins qui sont comme un ilot de civilisation dans un océan où les pirates sont rarement des hackers : Isabelle Amielh, Chantal BÉNIS, Pa-

tricia CHASSAT, Corinne DURAND-VIEL, Lydie HEUSCH, Isabelle ESSALHIENE, etc ... et tous les autres, notamment Hubert DELORI qui s'est converti sur le tard au monde du PC (... peut-être en entendant le *Start me up* des Rolling Stones racheté par Bill GATES) et Mikhael NICOLAIDIS que l'abondant courrier électronique mène à des heures de travail d'un thésard (à moins que ce ne soit pour lui qu'un moyen de vivre avec la nostalgie de sa jeunesse), Nadim KRIM, Jean-François PAILLOTIN, Richard PISTORIUS, Kholdum Torki, etc ...

les différentes personnes à qui j'ai été amené à faire appel à l'extérieur du laboratoire, notamment Alexandre CHAGOYA, Serge VIDAL du CIME et Chantal ROBACH du pôle grenoblois du réseau doctoral

les thésards du réseau doctoral susnommé, je citerai notamment Daniel CONIL, Jean-François GUILLAUD, Daniel HAGIMONT, Olivier JACQUIOT, Frédéric SAUNIER et deux étudiants qui ont su me donner un coup de main quand j'expérimentais les possibilités du logiciel Compass dont ils étaient un de fidèles adorateurs, David JACQUET et Raphael ROCHET,

les étudiants présents au laboratoire pour une durée plus ou moins longue suivant leurs statuts et notamment mes compagnons de galère, Rachid BOURAOUI, Hichem BOUTAMINE, Vincent COISSARD, Ihmed MOUSSA, Yustina KUSUMAPUTRI que je n'ai fais qu'entrevoir, Luis MONTALVO, Ali SKAF et certains stagiaires avec qui j'ai pu avoir d'intéressantes conversations, sans oublier les thésards d'autres équipes notamment Ricardo DUARTE et Vijay VIJAYA RAGHAVAN dont je me souviens des cours d'anglais par *e-mail*,

Monsieur Jean FRÉHEL qui sait incarner la coopération entre la recherche et l'industrie et qui me laissera quelques souvenirs de conversations possibles dans le monde scientifique.

Table des matières

Table des figures xi

Liste des tableaux xv

Avertissement 1

Partie I Bases scientifiques et applications à la conception 3

Chapitre 1
Motivations et présentation de cette thèse

I.1.1 Origine. 5
I.1.2 Le milieu ambiant. 5
I.1.3 L'idée directrice. 6
I.1.4 Quelques brefs exemples. 7
I.1.5 Topologie du manuscrit. 8

Chapitre 2
La transformée de Fourier, généralités

I.2.1 Définitions. 11
I.2.2 Développement de la transformée de Fourier rapide. 13
I.2.2.1 Généralités. 13
I.2.2.2 Entrelacement temporel. 14
I.2.2.3 Entrelacement fréquentiel. 18

I.2.3 Remarques sur la décomposition d'une transformée de Fourier rapide. 21
I.2.3.1 Les décompositions en base deux. 21
I.2.3.2 Autres bases et règles d'emploi. 22
I.2.3.3 Question de vocabulaire. 23
I.2.4 Les moyens de calcul en Traitement du Signal. 25
I.2.5 Exemples de configuration de transformée de Fourier. 26
I.2.5.1 Dimensions et nombre d'échantillons. 26
I.2.5.2 Taille des nombres codant les échantillons. 27
I.2.5.3 Circuits existants à un stade industriel. 29
I.2.6 Implantation cablée. 29
I.2.6.1 T.F.R. pipelinée sans et avec rebouclage 29
I.2.6.2 Papillon en base 2. 31
I.2.7 Conclusion. 32

Chapitre 3
Évaluation du calcul d'une T.F.R. en trois dimensions

I.3.1 Introduction. 35
I.3.2 Calcul d'une T.F.R. $3D$ grâce à des T.F.R. $1D$. 36
I.3.3 Notation avec des chiffres signés. 38
I.3.4 Multiplieurs à notation en complément à deux à trés haute vitesse. 39
I.3.5 Influence des notations arithmétiques sur une architecture. 43
I.3.5.1 Performances théoriques temporelles. 43
I.3.5.2 Performances d'opérateurs arithmétiques conçus, fabriqués et testés. . . . 46
I.3.5.3 Coûts en surface d'implantation. 49
I.3.6 Perspectives. 51

Chapitre 4
Implantations d'une T.F.R. avec des opérateurs en ligne

I.4.1 Introduction 53
I.4.2 Papillon en ligne 54
I.4.2.1 Considérations sur l'arithmétique. 54
I.4.2.2 Codage d'une notation redondante et propriétés. 58
I.4.2.3 Les composants de base d'une implantation redondante. 59
I.4.2.4 Architecture finalement choisie. 60
I.4.2.5 Commentaires. 62
I.4.3 Circuit réalisé 63
I.4.4 Perspectives de ce travail. 66

I.4.4.1 Généralités. 66
I.4.4.2 Fréquence de travail. 66
I.4.4.3 Architecture. 66
I.4.4.4 Conclusion. 69

Partie II Développements théoriques 71

Chapitre 1
Architecture à saturation de bus

II.1.1 Introduction. 73
II.1.2 Décomposition d'une T.F.R. sur plusieurs circuits. 73
II.1.2.1 Dimension d'une transformée et choix d'une macrobase. 73
II.1.2.2 Surface d'une T.F.R. réduite. 74
II.1.2.3 Coût de la mémoire. 76
II.1.2.4 Échanges de données entre papillons sériels. 77
II.1.3 Communication des données. 77
II.1.3.1 Généralités. 77
II.1.3.2 Architecture à saturation de bus. 78
II.1.3.3 Évolutions. 84
II.1.4 Adaptations. 84

Chapitre 2
Opérateurs sériels de taille variable

II.2.1 Introduction. 87
II.2.2 Principes de base. 88
II.2.2.1 Généralités. 88
II.2.2.2 Opérations arithmétiques et opérandes. 88
II.2.2.3 Quantifications des erreurs. 89
II.2.2.4 Intégration des erreurs. 90
II.2.3 Les voies traditionnelles de calcul. 91
II.2.3.1 Entrelacement temporel sans débordement par addition. 91

II.2.3.2 Entrelacement temporel avec division systématique par 2. 91
II.2.3.3 Entrelacement fréquentiel sans débordement par addition. 92
II.2.3.4 Entrelacement fréquentiel avec division systématique par 2. 92
II.2.3.5 Remarques. 92
II.2.4 Proposition. 95
II.2.4.1 Entrelacement temporel. 95
II.2.4.2 Entrelacement fréquentiel. 96
II.2.4.3 Comparaison. 96
II.2.5 Architecture à base d'opérateurs à taille variable. 97
II.2.5.1 Généralités. 97
II.2.5.2 Surface d'une implantation repliée. 97
II.2.5.3 Surface d'une implantation étalée. 98
II.2.5.4 Performance en temps 100
II.2.6 Applications de la croissance par pas. 100

Chapitre 3
Opérateurs arithmétiques parallèles de taille variable

II.3.1 Introduction. 103
II.3.2 Proposition. 104
II.3.3 Amélioration du temps de calcul. 104
II.3.3.1 Généralités. 104
II.3.3.2 Opérateurs à coefficients de proportionnalité linéaires. 105
II.3.3.3 Opérateurs à coefficients de proportionnalité logarithmiques. 105
II.3.4 Amélioration de la surface d'implantation. 106
II.3.4.1 Généralités. 106
II.3.4.2 Opérateurs à coefficients de proportionnalité linéaires. 106
II.3.4.3 Opérateurs à coefficients de proportionnalité logarithmiques. 108
II.3.5 Perspectives. 109
II.3.5.1 Généralités. 109

Chapitre 4
Bases de décomposition supérieures à 4

II.4.1 Introduction 111
II.4.2 Aspects théoriques. 112
II.4.2.1 Généralités. 112
II.4.2.2 Entrelacement temporel de base 8. 112
II.4.2.3 Entrelacement fréquentiel de base 8. 113

II.4.2.4Remarque sur le produit par $\sqrt{2}$. 115
II.4.2.5Entrelacement temporel de base 12. 116
II.4.2.6Entrelacement fréquentiel de base 12. 118
II.4.3 Amélioration en terme de vitesse. 120
II.4.4 Généralités. 120
II.4.4.1Nombre d'étapes successives de calcul. 122
II.4.4.2Nombre de multiplications. 123
II.4.5 Problème de la surface d'implantation. 124
II.4.6 Conclusion. 125

Partie III Cas particulier des matrices creuses **127**

Chapitre 1
La méthode cristallographique

III.1.1 Introduction. 129
III.1.2 Etude de la structure d'une molécule. 129
III.1.3 La méthode cristallographique. 129
III.1.4 Erreur dans une T.F.R. $3D$. 133

Chapitre 2
Erreurs dans une T.F.R. d'une matrice creuse

III.2.1 Spécificités de la reconstitution de données 137
III.2.2 Influence des données nulles sur la précision 138
III.2.2.1 Pas de débordement 138
III.2.2.2 Débordements possibles 138
III.2.3 Conclusion. 140

Chapitre 3
Avant de sombrer dans l'oubli

Index **145**

Bibliographie

Bibliographie **147**

Table des figures

2.1 chaine d'opérations successives des données incluant une transformée de Fourier. 13
2.2 schéma du traitement d'une T.F.R. de 8 échantillons à entrelacement temporel. . 16
2.3 schéma d'une T.F.R. de 8 échantillons à entrelacement temporel dans la littérature traditionnelle. 19
2.4 schéma du traitement d'une T.F.R. de 8 échantillons à entrelacement fréquentiel. 21
2.5 schéma d'une T.F.R. de 8 échantillons à entrelacement fréquentiel dans la littérature traditionnelle. 22
2.6 T.F.R. de seize échantillons pour une base 2 et une base 4. 23
2.7 schéma du traitement d'une T.F.R. de 8 échantillons avec une base mixte. 24
2.8 nombres caractéristiques d'échantillons d'une transformée de Fourier. 28
2.9 coefficients de proportionnalité de différents types de multiplieurs en fonction de la taille des opérandes. 29
2.10 T.F.R. pipelinée, parallélisée et totalement implantée. 30
2.11 T.F.R. pipelinée rebouclée. 31
2.12 papillon de TFR en base 2. 33

3.1 l'espace des données lors de chaque T.F.R. $1D$ et aprés la T.F.R. $3D$ globale. . . 36
3.2 rotation des données dans l'espace lors d'une T.F.R. $3D$. 37
3.3 architecture généreuse pour le calcul d'une T.F.R. $3D$. 37
3.4 architecture simplifiée pour une T.F.R. $3D$. 38
3.5 architecture ping-pong pour une T.F.R. $3D$. 38
3.6 additionneur sériel. 40
3.7 additionneur parallèle. 40
3.8 produit de deux nombres dissymétrique. 41
3.9 produit partiel et traitements associés. 42
3.10 diffusion de la retenue en fin de produit. 43
3.11 tranche de multiplieur à retenue à avance progressive de la retenue. 44
3.12 retard induit par les opérateurs en fonction du nombre d'échantillons, selon la notation des chiffres et la base de décomposition. 45
3.13 temps de calcul pour 256^3 échantillons selon la macrobase choisie. 46
3.14 temps de calcul pour diverses macrobases en fonction du nombre d'échantillons. . 47
3.15 exemple d'un opérateur arithmétique présentant des structures de pipelines transversales et longitudinales. 48
3.16 estimation simplifiée de la surface nécessaire pour une T.F.R. étalée en fonction du nombre d'échantillons et selon la base. 50
3.17 surface d'une T.F.R. repliée en fonction du nombre d'échantillons selon la base. . 52

4.1 extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés le premier des trois coups d'horloge qui amènent au résultat final. 56
4.2 extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés aprés le deuxième des trois coups d'horloge qui amènent au résultat final. . 57
4.3 extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés aprés le dernier des trois coups d'horloge qui amènent au résultat final. . . 57
4.4 schéma de principe d'un multiplieur en ligne à chiffre de poids fort en tête. 58
4.5 bloc PPM pour la synthèse des additionneurs à anticipation de retenue. 60
4.6 additionneur série à anticipation de retenue pour deux chiffres redondants. 60
4.7 additionneur série à anticipation de retenue pour trois chiffres redondants. 61
4.8 tranche pour additionneur parallèle à anticipation de retenue pour trois chiffres redondants. 61
4.9 papillon en ligne à base 2 aprés simplification et notamment présence de multiplieurs complexes. 62
4.10 structure du circuit choisie. 63
4.11 plan de masse du circuit. 64
4.12 dessin du circuit. 65
4.13 temps de calcul total d'une T.F.R. de 1024 points avec un seul circuit qui traite un nombre d'échantillons indiqué en abscisse pour différents architectures et implantations. 67
4.14 surface d'implantation en fonction du nombre d'échantillons. 68

1.1 T.F.R. 3D de 16 millions d'échantillons calculée avec des T.F.R. cablées de 256 échantillons. 75
1.2 T.F.R. 3D de 16 millions d'échantillons calculée avec des T.F.R. cablées de 256 échantillons et des T.F.R. cablées de 16 échantillons groupées dans un même circuit. 76
1.3 structure interne des circuits intégrant les T.F.R. de 256 et 16 échantillons. . . . 76
1.4 T.F.R. de 64 échantillons calculée avec une T.F.R. cablée de 8 échantillons. . . . 77
1.5 architecture à saturation de bus à deux niveaux. 79
1.6 nombre de papillons selon la macrobase en complément à 2. 81
1.7 temps de calcul en fonction de la macrobase en complément à 2. 82
1.8 nombre de papillons selon la macrobase en notation redondante. 83
1.9 temps de calcul en fonction de la macrobase en notation redondante. 83

3.1 temps de calcul relatif pour des opérateurs linéaires. 106
3.2 temps de calcul relatif pour des opérateurs logarithmiques. 107
3.3 surface relative d'une implantation à base d'opérateurs linéaires. 108
3.4 surface relative d'une implantation à base d'opérateurs logarithmiques. 109

4.1 valeurs remarquables des coefficients exponentiels pour une base 8. 113
4.2 structure mathématique d'un papillon de base 8 pour un entrelacement temporel. 114
4.3 architecture cablée d'un papillon de base 8 pour un entrelacement temporel. . . . 115
4.4 structure mathématique d'un papillon de base 8 pour un entrelacement fréquentiel. 116
4.5 architecture cablée d'un papillon de base 8 pour un entrelacement fréquentiel. . . 117
4.6 valeurs remarquables des coefficients exponentiels d'un papillon de base 12. . . . 118
4.7 structure mathématique d'un papillon de base 12 à entrelacement temporel. . . . 119
4.8 architecture cablée d'un papillon de base 12 à entrelacement temporel. 120
4.9 structure mathématique d'un papillon de base 12 avec un entrelacement fréquentiel. 121

4.10 architecture d'un papillon de base 12 avec un entrelacement fréquentiel. 122

1.1 phénomène de diffraction. 131
1.2 plans de l'espace et courbes d'iso-densité. 133
1.3 exemple d'architecture tri-dimensionnelle obtenue. 134

2.1 influence sur l'erreur de la présence d'échantillons nuls. 139
2.2 influence sur le risque de dépassement de la présence d'échantillons nuls. 140

Liste des tableaux

2.1 exemples de dimensions et de variables de transformées de Fourier. 26
2.2 exemples de tailles et de dimensions de transformées de Fourier rapides. 27
2.3 circuits ayant atteint un stade industrialisable aux cours des dernières années. . . 30

3.1 résultats des différentes implantations étudiées lors des comparaisons. 48

4.1 Évolution des grandeurs intermédiaires d'un produit. 56
4.2 codage d'une notation redondante. 58
4.3 codage d'une négation en notation redondante. 59
4.4 produit de deux chiffres binaires redondants. 59

1.1 maquettes de test de diverses techniques utilisées dans les architectures à saturation de bus. 85

2.1 coefficients d'erreur imposées par les moyens de calcul choisis. 90
2.2 coefficients d'erreur pour les différents entrelacements par les voies traditionnelles de calcul. 93
2.3 valeurs asymptotiques des coefficients d'erreur pour les différents entrelacements par les voies traditionnelles de calcul. 94
2.4 coefficients d'erreur pour les diverses solutions et entrelacement. 96

3.1 temps de calcul de différents types d'opérateurs. 104
3.2 surface fonction des opérateurs et de la vitesse. 107

4.1 opérations intervenant dans un papillon de base 8. 123
4.2 opérations intervenant dans un papillon de base 12. 124
4.3 puissances successives de 8 et 12 dans l'espace de travail. 124

1.1 Moyens d'étude et précision des résultats. 130
1.2 corps étudiés et données de mesure. 132

Avertissement

La Recherche se veut faire à un instant ce que l'Industrie ne peut pas faire avant plusieurs années. De même des avancées théoriques attendent souvent longtemps d'être mises en application, même dans un cadre de recherche, que des moyens fournis par d'autres sciences soient disponibles. La méthode cristallographique qui fait partie des méthodes d'investigation des structures moléculaires illustre ce fait. Mon passé professionnel m'a fourni cette piste pour mon sujet de thèse, une opportunité de travailler ailleurs que dans le vide, de contribuer à l'avance de mon temps et d'anticiper ce que l'Industrie fera dans quelques années. Les besoins de calcul qui apparaissent dans la méthode cristallographique sont tels qu'il apparait souhaitable de faire appel à des solutions spécialisées, donc à priori des circuits intégrés spécifiques, du moins dans une certaine mesure.

La région grenobloise dans laquelle s'est déroulé la préparation de cette thèse accueille des cristallographes, des concepteurs et des fabricants de circuits intégrés. Physicien appliqué de par ma formation initiale, je me suis trouvé à la frontière de ces domaines si différents et vous les trouverez donc dans ce manuscript de thèse, en amont ou en aval des travaux purement dédiés à la Physique Appliquée que j'ai pu y développer. Les années futures diront dans quelle mesure les résultats scientifiques qui découlent de ces années de travail peuvent servir, à qui et à quoi. Il y a toutefois un point qui semble émerger. Lors de la rédaction de ce manuscrit, une entreprise américaine[1] a mis sur le marché un circuit dédié au calcul de la transformée de Fourier à des fins scientifiques. Même si la démarche qui a amené ce produit est différente de la mienne puisqu'il s'agit du premier circuit intégré en nombres flottants commercialisé, mes objectifs initiaux ont finis par être rejoints par les ambitions commerciales d'une société de pointe du marché de l'industrie informatique. Ce qui n'est aprés tout que le destin d'un travail de recherche.

1. Texas Memory Systems, Inc.

Première partie

Bases scientifiques et applications à la conception

Chapitre 1

Motivations et présentation de cette thèse

Sommaire

I.1.1 Origine. 5
I.1.2 Le milieu ambiant. 5
I.1.3 L'idée directrice. 6
I.1.4 Quelques brefs exemples. 7
I.1.5 Topologie du manuscrit. 8

I.1.1 Origine.

UNE THÈSE EST UNE TRANCHE DE VIE d'un thésard, d'une certaine façon d'une équipe, *in the TIMA Laboratory on parle de group*, et d'un laboratoire, parfois même d'une entreprise. Le sujet de cette thèse est partie il y a quatre ans d'une discussion lors d'un repas. La méthode cristallographique, une méthode pour déterminer la structure spatiale d'une molécule d'un cristal, est connue dans ses principes depuis de nombreuses années, mais n'est mise en oeuvre par personne. Pour les cas simples, elle possède des concurrents qui ont fait leurs preuves. Pour les cas plus compliqués, ces concurrents montrent leurs limites, mais la méthode cristallographique reste inemployée. Sa mise en oeuvre demande des moyens de calcul en transformée de Fourier qui pourraient certes être mobilisés, mais dont le coût est tel que l'emploi de cette méthode est reportée en attendant que le développement des sciences et des techniques autorisent leur abaissement. La conception de la première bombe atomique a justifié la naissance du premier véritable ordinateur ayant pu exister, mais les circonstances sont ici quant même bien différentes !

I.1.2 Le milieu ambiant.

LA TRANSFORMÉE DE FOURIER peut être considérée comme la généralisation de la décomposition en séries de Fourier d'une fonction périodique à une fonction quelconque. Elle peut être définie pour une fonction ayant une variable, c'est le cas le plus courant, ou plusieurs. Le nombre de variables indépendantes d'une fonction est égal à la dimension de la transformée de Fourier qui peut lui être associée. La transformée de Fourier la plus connue et la plus utilisée est la transformée monodimensionnelle qui associe à une fonction temporelle une fonction fréquentielle. L'analyse spectrale et ses applications particulières comme le contrôle non destructif,

le fitrage adaptatif, les télétransmissions et télécommunications numériques sont des domaines où les ingénieurs et scientifiques ont trouvé trés tôt, trouvent encore actuellement ou pourraient trouver des applications extrêmement importantes qui ne sont souvent que limitées par la puissance de calcul à mettre en oeuvre. Les transformées multidimensionnelles les plus courantes sont appliquées à un plan ou à l'espace géométrique, en conjonction éventuelle avec le temps pour faire intervenir la dynamique temporelle du système étudié. Nous pouvons citer entre autres exemples le traitement des images en général, la tomographie et le scanner en mèdecine, l'exploitation des données fournies par un radar ou un sonar. Le frein à l'emploi des transformées de Fourier multidimensionnelles est le fait que l'existence de plusieurs dimensions augmente considérablement la quantité de calcul à effectuer et en réserve l'usage, dans la pratique, aux domaines où des considérations économiques permettent l'utilisation de moyens conséquents et aux domaines où la précision nécessaire des résultats permet de se satisfaire de moyens relativement courants. L'accélération du progrés technique fait qu'il existe une demande de plus en plus pressante pour rendre plus accessible l'utilisation de ces techniques à de nouveaux domaines et améliorer les performances obtenues là où son emploi est déja effectif. La méthode cristallographique n'est donc pas la seule application susceptible de tirer profit de travaux de recherche semblables à cette thèse.

I.1.3 L'idée directrice.

L'OBJECTIF DE NOTRE TRAVAIL était d'étudier les possibilités d'améliorer les moyens de calcul d'une transformée de Fourier à trois dimensions en la considérant simplement comme une transformée d'un trés grand nombre d'échantillons. Ceci en vue d'accélérer des calculs cristallographiques, tout en laissant ouverte la voie pour étendre les résultats obtenus à d'autres applications. Le développement de techniques ou de systèmes trop spécialisés conduit souvent à des impasses, en raison des moyens démesurés à mettre en oeuvre pour présenter un prototype qui puisse être confié à une personne autre que l'une de celles qui ont contribué à son élaboration et, pire encore, dont la mise à jour soit suivie, ne serait-ce que pendant quelques années. Il se trouve que la méthode cristallographique n'en est qu'à ses débuts, du point de vue de son utilisation pratique, en particulier à cause de la masse de calculs de transformée de Fourier à mettre en oeuvre. D'où la nécessité de mener un travail qui puisse être poursuivi ultérieurement pour accompagner les développements de cette méthode, par nous ou d'autres personnes. De plus, l'accés aux moyens de calcul suffisants pour la mettre en oeuvre amènera certainement à résoudre des problèmes propres à cette méthode qui limiteront sa diffusion dans un premier temps et dont l'importance est masquée par le problème de la puissance de calcul, phénomène classique dans le développement d'une technique nouvelle.

Il faut noter par ailleurs que la dimension d'une transformée est un paramètre variant peu pour un type d'applications donné. L'utilisation de ce paramètre pour décrire un nouvel algorithme ou améliorer un algorithme déja existant entraînerait éventuellement un saut des performances au moment de sa première mise en oeuvre. Mais il peut être facilement constaté que l'utilisateur de ce genre de techniques demande, au fil des années, de faire porter ces calculs sur un nombre croissant d'échantillons, quelque soit la dimension constatée. Cette évolution n'est que le résultat des performances de plus en en plus importantes offertes aux utilisateurs, la possibilité de faire mieux fait naître le besoin d'avoir encore mieux. D'où la nécessité d'étudier ce problème en priorité et de considérer la dimension des espaces de travail comme un paramètre non prioritaire.

I.1.4 Quelques brefs exemples.

HORMIS QUELQUES CAS PARTICULIERS , le développement de l'utilisation pratique de la transformée de Fourier a été permis par l'apparition de l'ordinateur. La complexité des applications a suivi l'augmentation de la puissance des modèles qui se sont succédés et profité des acquis de la science qu'il a engendré, l'informatique. Pour illustrer les divers problèmes qui se posent pour mettre en oeuvre le calcul d'une transformée et qui expliquent l'histoire de son développement, nous pouvons citer quelques exemples d'applications possibles ou effectives qui sont trés caractéristiques des demandes et des réponses des différents intervenants :

- le contrôle non destructif, par la quantité limitée d'informations à traiter dont il peut se satisfaire, a donné lieu à des applications industrielles assez larges depuis déja pas mal d'années.

- Le traitement des données fournies par un radar ou un sonar a connu un essor déja ancien en raison de son utilisation militaire, même si les applications civiles ont pu bénéficier avec un certain retard des progrés qui en ont découlé. L'importance stratégique, donc politique de ces questions a évidemment favorisé ce développement, mais il ne faut pas oublier que les phénomènes traités sont relativement lents, cas du sonar, ou présentent des caractéristiques qui évoluent relativement lentement, cas du radar. Un sous-marin n'atteint que quelques dizaines de kilomètres par heure et un avion même volant à haute vitesse ne change pas de vitesse ou de direction dans un temps extrêmement court.

- La tomographie médicale, bien que faisant appel à une transformée multidimensionnelle, a pu se développer relativement tôt malgré le coût nécessaire en raison de l'enjeu social sous-jacent et grâce au niveau relativement faible de la précision des calculs indispensable pour une exploitation des résultats.

- La radio et la télédiffusion numérique sont, au contraire des précédents exemples, des applications où la quantité des calculs à effectuer, joint à la nécessité de les effectuer en temps réel et de maintenir des coûts de fabrication relativement bas en raison de l'aspect grand public de ces produits, font que ces techniques ne se développent que depuis peu.

- Le filtrage numérique connaît toujours des développements dans l'utilisation de la transformée de Fourier.

- La méthode cristallographique permet de déterminer l'architecture tridimensionnelle d'une molécule avec une précision trés supérieure à celle des méthodes concurrentes, mais demande un nombre considérable de calculs dont la transformée de Fourier forme la plus grande part. La puissance de calcul nécessaire est le frein principal au développement et à la diffusion de cette méthode.

Ces différentes utilisations entrainent l'emploi de moyens de calcul, résultat d'un compromis entre les performances demandées, le coût acceptable d'un point de vue économique et la technologie accessible :

- solution logicielle (micro-contrôleurs, ordinateurs généralistes, processeurs vectoriels)

- solution logicielle dédiée (processeurs de traitement du signal)

- solution cablée (circuits spécialisés).

I.1.5 Topologie du manuscrit.

LES FOUS EUX-MÊMES ONT UNE LOGIQUE , mais le mieux pour être compris de ceux qui tenteront de lire ce manuscript a été pour moi de montrer au lecteur quels sont les cheminements et les écueils qu'il rencontrerait s'il tentait de développer une architecture spécialisée pour obtenir de hautes performances de calcul. Le lecteur peut d'ailleurs, s'il n'a pas la nécessité de tout lire, se contenter d'une seule partie ou de les lire dans un ordre différent que celui proposé. Chacune est consacrée à une démarche qui correspond à une optique bien précise. Toutefois, ceux qui daigneraient y consacrer le temps suffisant trouveront les raisons de mon cheminement à travers la science, libre à eux de ne pas partager mes raisonnements et mes conclusions.

Partons du réel et de l'accessible dans la partie I. Nous rappelons dans le chapître 2 ce que sont les transformées de Fourier, les méthodes et les moyens pour les calculer. Nous en profitons pour fixer quelques points qui, de vérités mathématiques, deviennent des nids d'ambiguité pour un calcul cablé. Ce qui nous amène à préciser les cas types de données dont nous visons le traitement. Le chapître 3 reprend une étude de faisabilité d'une transformée tridimensionnelle. Faite au début de cette thèse, elle était basée sur les opérateurs arithmétiques d'une thésarde qui venait de soutenir sa thèse [Kus93] et visait d'une part à appliquer les recherches sur l'implantation de cette transformée faites en grand nombre il y a une vingtaine d'années et d'autre part à démontrer la supériorité des arithmétiques dîtes redondantes, notamment dans ce cas précis. Menée plus ou moins en parallèle avec le travail décrit dans le chapître suivant, cela aurait pu en guider la suite. Nous avons repris ces travaux pour uniformiser le type de technologie utilisée par les deux parties de la comparaison en nous appuyant sur les travaux d'un autre thésard. Nous analyserons les nombreuses failles qui émaillaient cette étude, aidés par l'oeil aiguisé de comités scientifiques qui avaient eu vent du texte résultant. Le chapître 4 nous permet de présenter ce qui était effectivement possible de réaliser dans ce domaine en réutilisant le travail réalisé avant le début de cette thèse par d'autres étudiants en thèse de cette équipe dans le domaine des opérateurs arithmétiques. La collaboration de l'un d'entre eux [Ska95] et l'aide de deux stagiaires d'une école d'ingénieurs a été l'opportunité d'atteindre le tracé d'un circuit. Il ne réalise qu'un calcul de transformée de Fourier à une dimension et pour un faible nombre d'échantillons, mais avec des nombres ayant une taille classique dans le domaine du calcul scientifique. Ce qui le différencie des réalisations industrielles des dernières années. Ce travail permet de mettre en évidence les limites d'une telle démarche. Cette partie considère le problème de calculs faits sur des nombres à taille courante dans le monde scientifique, 32 bits, mais l'importance de la surface nécessaire pour les architectures de T.F.R. nous amène dans les parties suivantes à étendre, dans certains cas, nos diverses hypothèses de travail vers des tailles de nombres plus habituelles au monde industriel que scientifique.

Ayant perdu certaines de nos illusions, mais le coeur toujours vaillant et désormais riche de notre nouvelle expérience, nous laissons travailler notre imagination dans la partie II pour tracer les chemins qui pourraient mener à la conception d'architectures pour le calcul cablé de transformées de Fourier à grand nombre d'échantillons. Le faible nombre d'échantillons que peut traiter des circuits intégrés réalisables à court ou moyen terme nous amène dans le chapître 1 à proposer une architecture basée sur la communication d'un grand nombre de circuits de calcul. Cette démarche ne peut qu'entraîner des coûts de réalisation importants, ce qui nous amène à poser la question de la taille des opérandes du calcul pour limiter la surface d'implantation nécessaire à un nombre d'échantillons donné, donc à nous intéresser à la précision des calculs. Nous rappelons dans le chapître 2 les principes développés par d'illustres prédécesseurs sur ce dernier sujet et soulignons nos divergences, tant dans les résultats que dans leur interprétation. Cela nous permet de proposer une solution permettant un compromis entre les paramètres qui provoquent les

cauchemars des concepteurs de tels circuits. A savoir précision, surface d'implantation et vitesse des opérateurs élémentaires. Nous nous éloignons dans le chapître 3 des opérateurs sériels et étendons notre proposition aux opérateurs parallèles, ce qui devrait nous projeter dans un futur plus lointain. Cette suggestion ne nous empêche pas dans le chapître 4 de nous interroger sur la possibilité d'utiliser avec profit des bases supérieures à quatre et d'estimer le coût et le gain de telles solutions.

Ces projections dans un futur plus ou moins proche, les mauvaises langues diront plus ou moins illusoire, ne nous empêche pas dans la partie III de nous rappeler ce qui a déclenché cette thèse, à savoir la cristallographie. Le bref rappel dans le chapître 1 de ce qu'est la méthode cristallographique, nous fait poser la question d'un emploi éventuel des propriétés qui peuvent apparaître dans les données traitées par les transformées de Fourier apparaissant dans cette méthode. D'abord à travers les conséquences de la présence de plusieurs dimensions dans l'espace de travail, en particulier dans le domaine de la précision des calculs. Ensuite dans le chapître 2 nous étudions l'influence de la présence de nombreuses valeurs nulles parmi les échantillons sur la précision. Ceci dans le but d'étudier la possibilité, d'une part d'employer d'éventuels composants déja existants bien au delà de leurs possibilités prévues pour un cas général, d'autre part de limiter la taille des opérandes des opérateurs, donc en particulier de leurs surface d'implantation. Le chapître 3 nous permet, avant que le lecteur tourne la dernière page de ce manuscrit, de conclure notre travail en traçant les grandes lignes d'un futur accessible par nos travaux tout en le comparant à d'autres démarches tout aussi accessibles ... en se souvenant que nul n'est prophète dans son pays.

Chapitre 2

La transformée de Fourier, généralités

Sommaire

I.2.1 Définitions. **11**
I.2.2 Développement de la transformée de Fourier rapide. **13**
I.2.2.1 Généralités. 13
I.2.2.2 Entrelacement temporel. 14
I.2.2.3 Entrelacement fréquentiel. 18
I.2.3 Remarques sur la décomposition d'une transformée de Fourier rapide. **21**
I.2.3.1 Les décompositions en base deux. 21
I.2.3.2 Autres bases et règles d'emploi. 22
I.2.3.3 Question de vocabulaire. 23
I.2.4 Les moyens de calcul en Traitement du Signal. **25**
I.2.5 Exemples de configuration de transformée de Fourier. **26**
I.2.5.1 Dimensions et nombre d'échantillons. 26
I.2.5.2 Taille des nombres codant les échantillons. 27
I.2.5.3 Circuits existants à un stade industriel. 29
I.2.6 Implantation cablée. **29**
I.2.6.1 T.F.R. pipelinée sans et avec rebouclage 29
I.2.6.2 Papillon en base 2. 31
I.2.7 Conclusion. **32**

I.2.1 Définitions.

UNE TRANSFORMÉE DE FOURIER DE DIMENSION m est définie d'une façon générale par m intégrales. Soient une fonction f dont les variables forment l'ensemble $\{x_0, x_1, \cdots, x_{m-1}\}$. Soient F la fonction transformée de f et $\{y_0, y_1, \cdots, y_{m-1}\}$ l'ensemble des variables de F. Soit j le nombre tel que nous ayons $j^2 = \Leftrightarrow 1$. Nous avons par définition de la transformée de Fourier :

$$F(y_{m-1}, \cdots, y_1, y_0) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f(x_{m-1}, \cdots, x_1, x_0).e^{-2\pi j(x_{m-1}.y_{m-1}+\cdots+x_1.y_1+x_0.y_0)}.dx_{m-1}.\cdots.dx_1.dx_0$$

Ce qui peut s'écrire sous la forme :

$$F(y_{m-1},\cdots,y_1,y_0) = \int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty} f(x_{m-1},\cdots,x_1,x_0).e^{-2\pi j x_{m-1}.y_{m-1}}\cdots.e^{-2\pi j x_1.y_1}.e^{-2\pi j x_0.y_0}.dx_{m-1}.\cdots.dx_1.dx_0$$

Soit encore :

$$F(y_{m-1},\cdots,y_1,y_0) = \int_{-\infty}^{+\infty}\left[\cdots\left(\int_{-\infty}^{+\infty} f(x_{m-1},\cdots,x_1,x_0).e^{-2\pi j x_{m-1}.y_{m-1}}.dx_{m-1}.\right)\cdots.e^{-2\pi j x_1.y_1}.dx_1\right].e^{-2\pi j x_0.y_0}.dx_0 \quad (2.1)$$

Restreignons le problème à une transformée de Fourier à une dimension. Il existe deux cas où l'intégrale qui définit cette transformée peut s'écrire sous la forme d'une sommation discrète. Le signal étudié doit :

- être discret par sa nature même
- être fini
- avoir une bande passante finie, la plus grande composante fréquentielle du signal ayant une valeur inférieure à la moitié de la fréquence d'échantillonnage qui est utilisée pour le rendre discret.

Ces deux dernières conditions sont contradictoires et n'apparaissent conciliables que dans le cas d'un signal périodique dont une période est étudiée. Ce calcul n'amène donc qu'à une approximation de la réalité, mais généralement utilisable.

Supposons que ces conditions puissent être considérée comme vérifiée. Nous pouvons alors écrire :

$$F(k) = \frac{1}{N}\sum_{n=0}^{N-1} f(n) \times \exp \Leftrightarrow 2\pi j \frac{n.k}{N}$$

où N est le nombre des valeurs discrètes et successives des variables, n et k sont respectivement les variables discrètes et normalisées de l'espace original et de l'espace transformé, f est la fonction originale et F la fonction transformée de f. Avec ces conditions, nous pouvons écrire la transformée de Fourier inverse sous la forme :

$$f(n) = \sum_{k=0}^{N-1} F(k) \times \exp 2\pi j \frac{n.k}{N}$$

Certains auteurs introduisent un coefficient $\frac{1}{N}$ dans la transformée de Fourier directe pour avoir une similitude avec les formules du développement d'une fonction en série de Fourier, d'autres dans les transformées inverses. Nous laisserons ce problème à l'utilisateur de circuits spécialisés dans la transformée de Fourier qui décidera de quelles grandeurs il traite, et ce pour les deux raisons suivantes :

- il s'agit d'un gain qui s'applique sur tous les coefficients d'un même type
- l'utilisation de ce genre de circuits impose de normaliser les données traitées pour qu'elles puissent correspondre aux opérandes, taillés au plus juste pour réduire dans toute la mesure du possible la surface d'implantation nécessaire aux opérateurs arithmétiques. Il y a donc d'autres gains à introduire pour rendre interprétable les données fournies par les circuits de T.F.R. et reçues par l'utilisateur qui devra normaliser ces résultats. L'ensemble de ces opérations est symbolisé dans la figure 2.1.

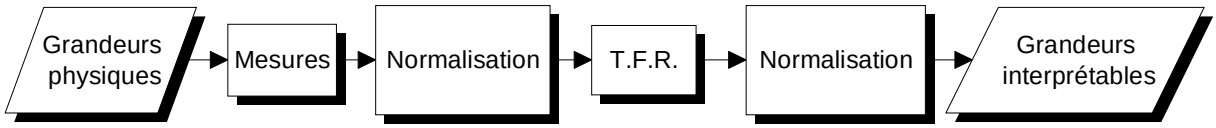

FIG. 2.1 – *chaine d'opérations successives des données incluant une transformée de Fourier.*

I.2.2 Développement de la transformée de Fourier rapide.

I.2.2.1 Généralités.

IL EXISTE TOUTE UNE LITTÉRATURE SUR QUI A DÉCOUVERT LES PRINCIPES qui permettent d'arriver aux algorithmes de transformée de Fourier rapide. Certains font remonter cette partie de l'Histoire à Gauss en 1805 ! Nous n'entrerons pas dans ces considérations. Signalons toutefois la propriété, démontrée par Danielson et Lanczos en 1942 et citée dans [PTF2s], qu'une transformée de Fourier discrète de taille N peut être calculée grâce à deux transformées de Fourier de taille $\frac{N}{2}$, l'une pour les indices pairs et l'autre pour les indices impairs. En 1965, Cooley et Tukey ont touvé un moyen de factoriser les calculs d'une transformée de Fourier, les rendant plus rapides [J.W65]. Différents algorithmes basés sur des principes semblables ont ensuite été proposés, créant toute une famille connue sous le nom de *Transformée de Fourier Rapide* (T.F.R.), utilisable si N est une puissance de p. La transformée globale est décomposée récursivement en des transformées minimales sur p valeurs. Cela conduit à $\frac{N}{p}$ calculs de base répétés durant $\log_p N$ étapes, chacun d'entre eux étant calculés par des cellules élémentaires appelées *papillons*. Nous devons nous rappeler que chaque calcul élémentaire d'une étape de la T.F.R. dépend des données fournies par l'étape précédente et seulement de celles-là. Cela apparaît dans les différents schémas de ce chapître représentant les différentes méthodes de décomposition et cas de figure envisagés. Ce qui entraine qu'un papillon est indépendant de ses semblables de même rang. Ainsi une T.F.R. peut être parallélisée. Les valeurs de p utilisées jusqu'à présent au niveau du calcul des données effectivement exécuté sont 2 et 4 parce qu'elles mènent à des simplifications intéressantes au point de vue du nombre d'instructions à exécuter.

Il existe deux approches pour établir les algorithmes de T.F.R., passer par une représentation matricielle de la définition ou sous une forme de séries arithmétiques. La première démarche traite l'ensemble des données et est trés adaptée à la description d'une réalisation logicielle de ces problèmes, la seconde ne décrit qu'une valeur particulière, mais est plus proche de la logique de l'architecture d'une implantation cablée de cette fonction. C'est donc par le biais de cette dernière que nous rappellerons le principe d'une transformée de Fourier rapide.

Pour ne pas trainer des formules trop longues, prenons le cas d'une transformée de Fourier d'un ensemble de huit échantillons et d'un algorithme à base 2. Nous étendrons ensuite le raisonnement à un nombre quelconque d'échantillons. Ecrivons les deux variables sous la forme de nombres binaires $n = n_0 + 2 \times n_1 + 4 \times n_2$ et $k = k_0 + 2 \times k_1 + 4 \times k_2$. Nous pouvons écrire :

$$\begin{aligned} F(k) &= \sum_{n=0}^{7} f(n) \times \exp \Leftrightarrow \frac{2\pi j.nk}{8} \\ &= \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \sum_{n_2=0}^{1} f(n) \times \exp \Leftrightarrow \frac{2\pi j}{8}(n_0 + 2.n_1 + 4.n_2)(k_0 + 2.k_1 + 4.k_2) \end{aligned}$$

Nous pouvons développer le terme de l'exponentielle et le simplifier, car l'exponentielle de tout nombre relatif multiplié par $2\pi j$ est égale à 1. Donc tout terme issu du développement de

$(n_0 + 2 \times n_1 + 4 \times n_2) \times (k_0 + 2 \times k_1 + 4 \times k_2)$ qui est multiple de huit disparait. Nous obtenons donc :

$$F(k) = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \sum_{n_2=0}^{1} f(n) \times \exp \Leftrightarrow \frac{2\pi j}{8}(n_0.k_0 + 2n_0.k_1 + 2n_1.k_0 + 4n_0.k_2 + 4n_1.k_1 + 4n_2.k_0) \tag{2.2}$$

C'est à ce moment du raisonnement qu'apparaissent les deux versions de la décomposition, selon que les sommations successives se font en factorisant les termes selon les n, cas d'un entrelacement temporel, ou les k, entrelacement fréquentiel. Pour la désignation des algorithmes, les auteurs français préfèrent le terme d'entrelacement à celui de décomposition plus prisé des Anglo-Saxons. Il tire son origine des schémas dus à un développement par les matrices.

I.2.2.2 Entrelacement temporel.

COOLEY ET TUKEY ont ouvert la voie au développement des algorithmes de transformée de Fourier rapides dans le sillage de l'informatique qui était à l'époque essentiellement la science du calcul. Inventeurs, ils ont donné leurs noms à cet algorithme.

$$\begin{aligned} F(k) &= \sum_{n_0=0}^{1} \left[\sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\frac{2\pi j}{8}(4n_2.k_0)} \right) .e^{-\frac{2\pi j}{8}(2n_1.k_0+4n_1.k_1)} \right] .e^{-\frac{2\pi j}{8}(n_0.k_0+2n_0.k_1+4n_0.k_2)} \\ &= \sum_{n_0=0}^{1} \left[\sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\pi j(n_2.k_0)} \right) .e^{-\frac{\pi j}{2}n_1(k_0+2k_1)} \right] .e^{-\frac{\pi j}{4}n_0(k_0+2k_1+4k_2)} \end{aligned} \tag{2.3}$$

Détaillons cette démarche connue pour aboutir aux équations de base. Nous pouvons faire apparaître dans la dernière étape de cette sommation la formule générale du calcul élémentaire de cet algorithme. Notons :

$$A = \sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\pi j(n_2.k_0)} \right) .e^{-\frac{\pi j}{2}n_1(k_0+2k_1)} \text{ pour } n \text{ tel que } n_0 = 0$$

$$B = \sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\pi j(n_2.k_0)} \right) .e^{-\frac{\pi j}{2}n_1(k_0+2k_1)} \text{ pour } n \text{ tel que } n_0 = 1$$

Développons l'exponentielle du terme n_0 :

$$\begin{aligned} e^{-\frac{\pi j}{4}n_0(k_0+2k_1+4k_2)} &= e^{-\frac{\pi j}{4}n_0(k_0+2k_1)}.e^{-\frac{\pi j}{4}n_0(4k_2)} \\ &= e^{-\frac{\pi j}{4}n_0(k_0+2k_1)}.e^{-\pi j n_0 k_2} \end{aligned}$$

Nous pouvons remarquer que :

$$e^{-\pi j n_0 k_2} = \begin{cases} 1 & \text{si } n_0 = 0, \text{ c'est à dire pour } A \\ 1 & \text{si } n_0 = 1 \text{ et } k_2 = 0, \text{ c'est à dire pour } B \text{ correspondant à } k_2 = 0 \\ \Leftrightarrow 1 & \text{si } n_0 = 1 \text{ et } k_2 = 1, \text{ c'est à dire pour } B \text{ correspondant à } k_2 = 1 \end{cases}$$

En notant $W = e^{-\frac{\pi j}{4} n_0 (k_0 + 2k_1)}$, nous obtenons :

$$F(k) = \begin{cases} A + B.W & \text{si } k_2 = 0 \\ A \Leftrightarrow B.W & \text{si } k_2 = 1 \end{cases}$$

La définition de $F(k)$ a une structure récursive qui apparaît dans l'équation 2.3. Les termes A et B ont la même définition et ne diffèrent que par la valeur de n_0 qui est contenue dans la variable n. Désignons donc A et B par le terme général $F'(k)$. Nous pouvons faire apparaître d'une manière identique dans la dernière étape de $F'(k)$ des termes tels que définis ci-dessous :

$$A' = \sum_{n_2=0}^{1} f(n).e^{-\pi j(n_2.k_0)} \text{ pour } n_1 = 0$$

$$B' = \sum_{n_1=0}^{1} f(n).e^{-\pi j(n_2.k_0)} \text{ pour } n_1 = 1$$

Développons l'exponentielle contenant le terme n_1 présente dans $F'(k)$:

$$\begin{aligned} e^{-\frac{\pi j}{2} n_1 (k_0 + 2k_1)} &= e^{-\frac{\pi j}{2} n_1 k_0}.e^{-\frac{\pi j}{2} n_1 2k_1} \\ &= e^{-\frac{\pi j}{2} n_1 k_0}.e^{-\pi j n_1 k_1} \end{aligned}$$

Nous pouvons remarquer que :

$$e^{-\pi j n_1 k_1} = \begin{cases} 1 & \text{si } n_1 = 0, \text{ c'est à dire pour } A \\ 1 & \text{si } n_1 = 1 \text{ et } k_1 = 0, \text{ c'est à dire pour } B \text{ correspondant à } k_1 = 0 \\ \Leftrightarrow 1 & \text{si } n_1 = 1 \text{ et } k_1 = 1, \text{ c'est à dire pour } B \text{ correspondant à } k_1 = 1 \end{cases}$$

En notant $W' = e^{-\frac{\pi j}{2} n_1 k_1}$, nous obtenons :

$$F'(k) = \begin{cases} A' + B'.W' & \text{si } k_1 = 0 \\ A' \Leftrightarrow B'.W' & \text{si } k_1 = 1 \end{cases}$$

Poursuivant notre raisonnement, désignons maintenant A' et B' par le terme général $F''(k)$ et définissons :

$$A'' = f(n) \text{ pour } n_1 = 0$$

$$B'' = f(n) \text{ pour } n_1 = 1$$

avec $W'' = 1$ pour garder la même struture dans les formules successives.

Nous pouvons faire la même remarque que dans les deux cas précédents :

$$e^{-\pi j n_2 k_0} = \begin{cases} 1 & \text{si } n_2 = 0, \text{ c'est à dire pour } A \\ 1 & \text{si } n_2 = 1 \text{ et } k_0 = 0, \text{ c'est à dire pour } B \text{ correspondant à } k_0 = 0 \\ \Leftrightarrow 1 & \text{si } n_2 = 1 \text{ et } k_0 = 1, \text{ c'est à dire pour } B \text{ correspondant à } k_0 = 1 \end{cases}$$

Et nous obtenons les expressions trés simples où les W'' peuvent être présents ou non :

$$F''(k) = \begin{cases} A'' + B''.W'' & \text{si } k_1 = 0 \\ A'' \Leftrightarrow B''.W'' & \text{si } k_1 = 1 \end{cases}$$

ou

$$F''(k) = \begin{cases} A'' + B'' & \text{si } k_1 = 0 \\ A'' \Leftrightarrow B'' & \text{si } k_1 = 1 \end{cases}$$

Ce qui nous donne le schéma de la figure 2.2 où les indices des W ont été omis par simplification. Nous pouvons noter la grande régularité du graphe et retrouver les remarques sur la parallélisation possible des calculs par suite de la structure obtenue.

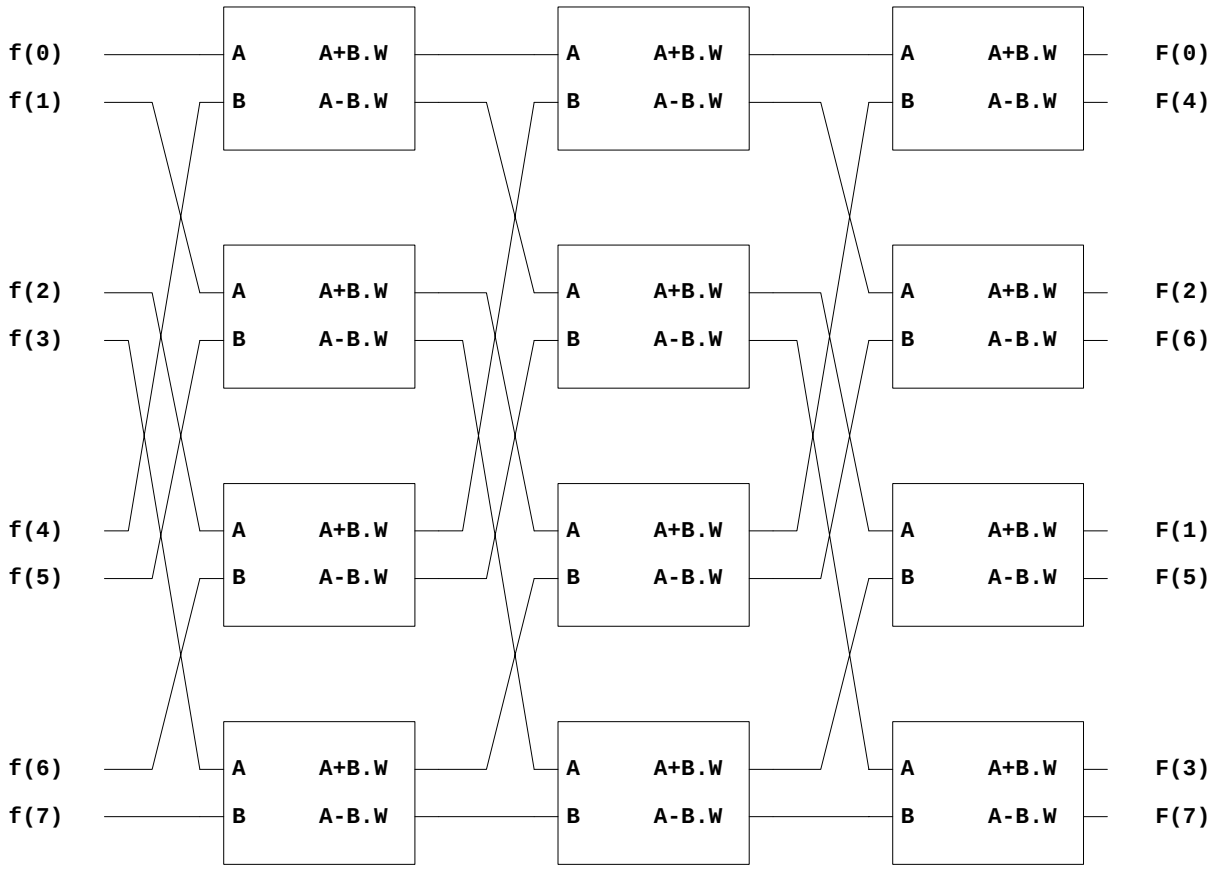

FIG. 2.2 – *schéma du traitement d'une T.F.R. de 8 échantillons à entrelacement temporel.*

Pour se raccrocher à la littérature plus traditionnelle sur la transformée de Fourier, nous donnons dans la figure 2.3 le schéma utilisé par les mathématiciens et informaticiens. Plus proche de leur raisonnement, il est la traduction d'une approche de la T.F.R. par une voie matricielle. Rappelons ce développement tel qu'il a été développé sous diverses formes par de nombreux auteurs [AR75] et en particulier par M. BELLANGER [Bel96].

$$\begin{vmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(N \Leftrightarrow 1) \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & & & \ddots & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)(N-1)} \end{vmatrix} \times \begin{vmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N \Leftrightarrow 1) \end{vmatrix}$$

Or nous avons :

$$\begin{vmatrix} F(0) \\ F(1) \\ \vdots \\ F(\frac{N}{2} \Leftrightarrow 1) \\ F(\frac{N}{2}) \\ \vdots \\ F(N \Leftrightarrow 1) \end{vmatrix} = \begin{vmatrix} F(0) \\ F(1) \\ \vdots \\ F(\frac{N}{2} \Leftrightarrow 1) \\ 0 \\ \vdots \\ 0 \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ F(\frac{N}{2}) \\ \vdots \\ F(N \Leftrightarrow 1) \end{vmatrix}$$

Nous pouvons donc écrire en ne considérant que les parties non nulles d'une matrice :

$$\begin{vmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(\frac{N}{2} \Leftrightarrow 1) \end{vmatrix} = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ 1 & W^2 & \cdots & W^{2(\frac{N}{2}-1)} \\ 1 & W^4 & \cdots & W^{4(\frac{N}{2}-1)} \\ \vdots & & \ddots & \vdots \\ 1 & W^{2(\frac{N}{2}-1)} & \cdots & W^{2(\frac{N}{2}-1)(\frac{N}{2}-1)} \end{vmatrix} \times \begin{vmatrix} f(0) \\ f(2) \\ f(4) \\ \vdots \\ f\left(2\left(\frac{N}{2} \Leftrightarrow 1\right)\right) \end{vmatrix}$$

$$+ \begin{vmatrix} 1 & 1 & \cdots & 1 \\ W & W^3 & \cdots & W^{N-1} \\ W^2 & W^6 & \cdots & W^{2(N-1)} \\ \vdots & & \ddots & \vdots \\ W^{\frac{N}{2}-1} & W^{3(\frac{N}{2}-1)} & \cdots & W^{(N-1)(\frac{N}{2}-1)} \end{vmatrix} \times \begin{vmatrix} f(1) \\ f(3) \\ f(5) \\ \vdots \\ f(N \Leftrightarrow 1) \end{vmatrix}$$

Pour simplifier les écritures définissons une matrice des coefficients exponentiels des échantillons pairs :

$$T_{\frac{N}{2}} = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ 1 & W^2 & \cdots & W^{2(\frac{N}{2}-1)} \\ 1 & W^4 & \cdots & W^{4(\frac{N}{2}-1)} \\ \vdots & & \ddots & \vdots \\ 1 & W^{2(\frac{N}{2}-1)} & \cdots & W^{2(\frac{N}{2}-1)(\frac{N}{2}-1)} \end{vmatrix}$$

Il a été démontré que la matrice des échantillons impairs est le produit d'une matrice diagonale par $T_{\frac{N}{2}}$. Cette matrice diagonale est égale à :

$$\begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & W & 0 & \cdots & 0 \\ 0 & 0 & W^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & W^{\left(\frac{N}{2}-1\right)} \end{vmatrix}$$

En se rappelant d'autre part que $W^N = 1$, nous obtenons les deux égalités matricielles suivantes :

$$\begin{vmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(\frac{N}{2} \Leftrightarrow 1) \end{vmatrix} = T_{\frac{N}{2}} \begin{vmatrix} f(0) \\ f(2) \\ f(4) \\ \vdots \\ f\left(2\left(\frac{N}{2} \Leftrightarrow 1\right)\right) \end{vmatrix} + \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & W & 0 & \cdots & 0 \\ 0 & 0 & W^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & W^{\left(\frac{N}{2}-1\right)} \end{vmatrix} T_{\frac{N}{2}} \begin{vmatrix} f(1) \\ f(3) \\ f(5) \\ \vdots \\ f(N \Leftrightarrow 1) \end{vmatrix}$$

$$\begin{vmatrix} F(\frac{N}{2}) \\ F(\frac{N}{2}+1) \\ F(\frac{N}{2}+2) \\ \vdots \\ F(N \Leftrightarrow 1) \end{vmatrix} = T_{\frac{N}{2}} \begin{vmatrix} f(0) \\ f(2) \\ f(4) \\ \vdots \\ f\left(2\left(\frac{N}{2} \Leftrightarrow 1\right)\right) \end{vmatrix} \Leftrightarrow \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & W & 0 & \cdots & 0 \\ 0 & 0 & W^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & W^{\left(\frac{N}{2}-1\right)} \end{vmatrix} T_{\frac{N}{2}} \begin{vmatrix} f(1) \\ f(3) \\ f(5) \\ \vdots \\ f(N \Leftrightarrow 1) \end{vmatrix}$$

En renouvellant ces décompositions matricielles, nous arrivons à des T.F.R. de deux échantillons dont la matrice T_i correspondante s'écrit :

$$T_2 = \begin{vmatrix} 1 & 1 \\ 1 & \Leftrightarrow 1 \end{vmatrix}$$

Cette décomposition amène la structure de la figure 2.3 .

En raison de la nature récursive de l'équation de définition d'une T.F.R. de taille quelconque, nous pouvons étendre ce schéma quelque soit le nombre d'échantillons traités pouvu qu'il soit une puissance de deux.

I.2.2.3 Entrelacement fréquentiel.

SANDE ET TUKEY ont exploré l'autre voie permise par application directe des formules de transformées de Fourier discrètes pour arriver à la décomposition de ces transformées. Ce qui leur a valu le droit, en tant qu'auteurs de cet algorithme, de mettre leurs noms sur celui-ci.

$$F(k) = \sum_{n_0=0}^{1} \left[\sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\frac{2\pi j}{8}(n_0.k_0+2n_1.k_0+4n_2.k_0)} \right) .e^{-\frac{2\pi j}{8}(2n_0.k_1+4n_1.k_1)} \right] .e^{-\frac{2\pi j}{8}(4n_0.k_2)}$$

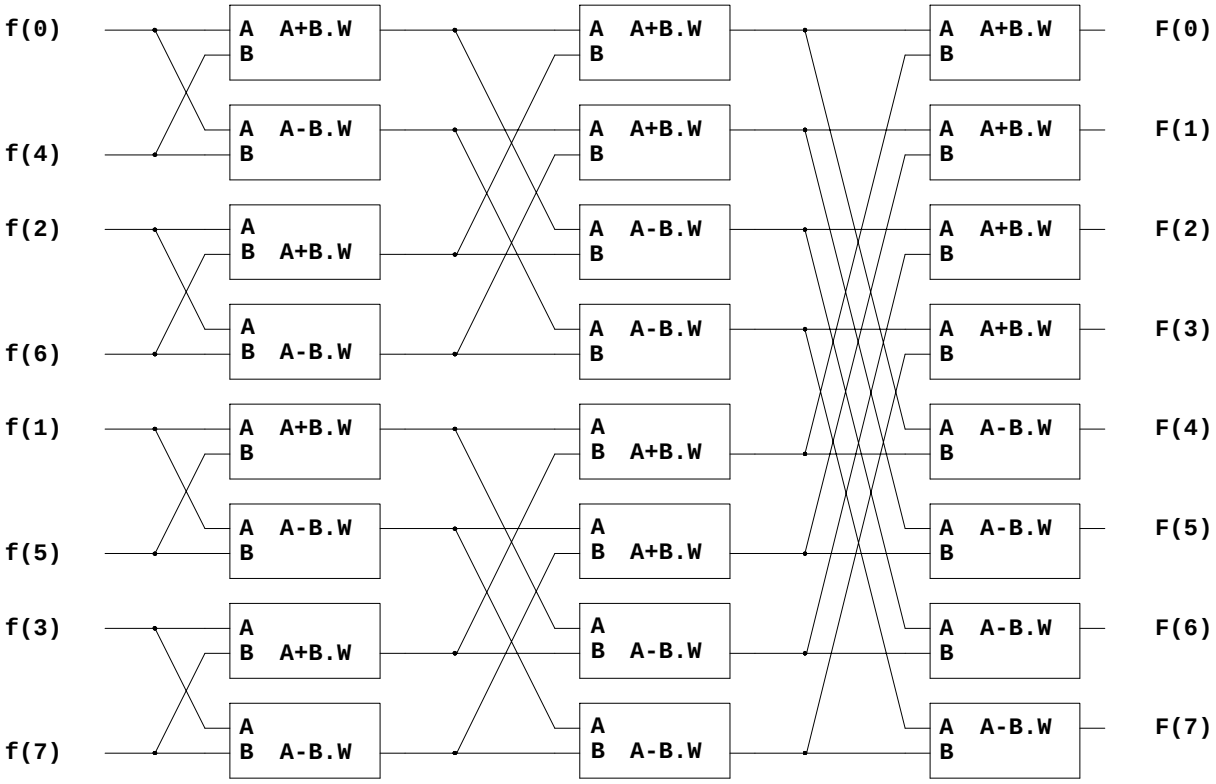

FIG. 2.3 – *schéma d'une T.F.R. de 8 échantillons à entrelacement temporel dans la littérature traditionnelle.*

$$= \sum_{n_0=0}^{1} \left[\sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} \right) .e^{-\frac{\pi j}{2}k_1(n_0+2n_1)} \right] .e^{-\pi j(k_2.n_0)}$$

Nous reprenons des définitions similaires pour A, B et W :

$$A = \sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} \right) .e^{-\frac{\pi j}{2}k_1(n_0+2n_1)} \text{ pour } n_0 = 0$$

$$B = \sum_{n_1=0}^{1} \left(\sum_{n_2=0}^{1} f(n).e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} \right) .e^{-\frac{\pi j}{2}k_1(n_0+2n_1)} \text{ pour } n_0 = 1$$

$$W = 1.$$

Nous avons les valeurs remarquables de l'exponentielle factorisée qui nous amènent aux expressions de $F(k)$ mises sous la forme qui apparait dans la suite du raisonnement pour garder le même type de formule.

$$e^{-\pi j n_0 k_2} = \begin{cases} 1 & \text{si } n_0 = 0, \text{ c'est à dire pour } A \\ 1 & \text{si } n_0 = 1 \text{ et } k_2 = 0, \text{ c'est à dire pour } B \text{ correspondant à } k_2 = 0 \\ \Leftrightarrow 1 & \text{si } n_0 = 1 \text{ et } k_2 = 1, \text{ c'est à dire pour } B \text{ correspondant à } k_2 = 1 \end{cases}$$

Formules les plus simples :

$$F(k) = \begin{cases} A + B & \text{si } k_2 = 0 \\ A \Leftrightarrow B & \text{si } k_2 = 1 \end{cases}$$

Formules les plus régulières :

$$F(k) = \begin{cases} A + B & \text{si } k_2 = 0 \\ (A \Leftrightarrow B).W & \text{si } k_2 = 1 \end{cases}$$

Poursuivons notre raisonnement comme pour la décomposition temporelle et définissons A', B' et W' :

$$A' = \sum_{n_2=0}^{1} f(n).e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} \text{ pour } n_0 = 0$$

$$B' = \sum_{n_2=0}^{1} f(n).e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} \text{ pour } n_0 = 1$$

$$W' = e^{-\frac{\pi j}{2}k_1.n_0}$$

Remarquons que $W' = 1$ pour $k_1 = 0$.

Décomposons l'exponentielle qui contient W' et notons les valeurs remarquables de l'exponentielle simple qui y apparaît pour arriver aux expressions de $F'(k)$.

$$\begin{aligned} e^{-\frac{\pi j}{2}k_1(n_0+2n_1)} &= e^{-\frac{\pi j}{2}k_1(n_0)}.e^{-\frac{\pi j}{2}k_1(2n_1)} \\ &= \underbrace{e^{-\frac{\pi j}{2}k_1.n_0}}_{W'}.e^{-\pi j k_1 n_1} \end{aligned}$$

$$e^{-\pi j n_1 k_1} = \begin{cases} 1 & \text{si } n_1 = 0, \text{ c'est à dire pour } A' \\ 1 & \text{si } n_1 = 1 \text{ et } k_1 = 0, \text{ c'est à dire pour } B' \text{ correspondant à } k_1 = 0 \\ \Leftrightarrow 1 & \text{si } n_1 = 1 \text{ et } k_1 = 1, \text{ c'est à dire pour } B' \text{ correspondant à } k_1 = 1 \end{cases}$$

$$F'(k) = \begin{cases} (A' + B') & \text{si } k_1 = 0 \\ (A' \Leftrightarrow B').W' & \text{si } k_1 = 1 \end{cases}$$

Remontant la récursivité de $F(k)$, nous définissons A'', B'' et W'' :

$$A'' = f(n) \text{ pour } n_2 = 0$$

$$B'' = f(n) \text{ pour } n_2 = 1$$

$$W'' = e^{-\frac{\pi j}{4}k_0(n_0+2n_1)}$$

Remarquons que $W'' = 1$ pour $k_0 = 0$.

Décomposons l'exponentielle contenant W'' et notons les valeurs remarquables de l'exponentielle simple qui y apparaît pour arriver aux expressions de $F''(k)$.

$$\begin{aligned} e^{-\frac{\pi j}{4}k_0(n_0+2n_1+4n_2)} &= e^{-\frac{\pi j}{4}k_0(n_0+2n_1)}.e^{-\frac{\pi j}{4}k_0(4n_2)} \\ &= \underbrace{e^{-\frac{\pi j}{4}k_0(n_0+2n_1)}}_{W''}.e^{-\pi j k_0.n_2} \end{aligned}$$

$$e^{-\pi j n_2 k_0} = \begin{cases} 1 & \text{si } n_2 = 0, \text{ c'est à dire pour } A'' \\ 1 & \text{si } n_2 = 1 \text{ et } k_0 = 0, \text{ c'est à dire pour } B'' \text{ correspondant à } k_0 = 0 \\ \Leftrightarrow 1 & \text{si } n_2 = 1 \text{ et } k_0 = 1, \text{ c'est à dire pour } B'' \text{ correspondant à } k_0 = 1 \end{cases}$$

$$F''(k) = \begin{cases} (A'' + B'') & \text{si } k_0 = 0 \\ (A'' \Leftrightarrow B'').W'' & \text{si } k_0 = 1 \end{cases}$$

Ce qui nous donne le schéma de la figure 2.4 qui peut être adapté à un nombre quelconque d'échantillons pourvu qu'il soit une puissance de deux. Pour reprendre les schémas traditionnels de la littérature dans ce domaine, la figure 2.5 reprend l'algorithme à entrelacement fréquentiel sous une approche par voie matricielle.

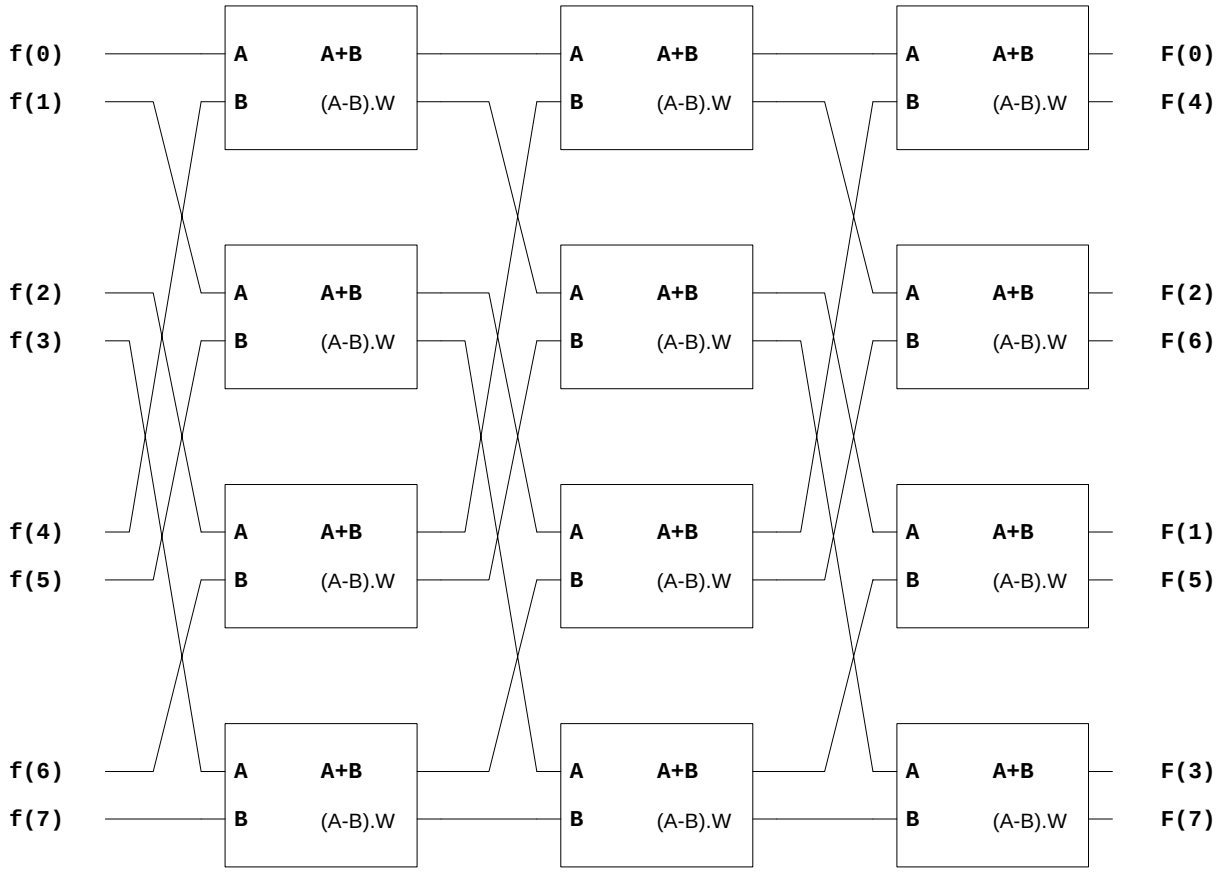

FIG. 2.4 – *schéma du traitement d'une T.F.R. de 8 échantillons à entrelacement fréquentiel.*

I.2.3 Remarques sur la décomposition d'une transformée de Fourier rapide.

I.2.3.1 Les décompositions en base deux.

LES COEFFICIENTS EXPONENTIELS suivent une évolution inverse dans les deux types de décomposition. W, W' et W'' dans la décomposition fréquentielle sont respectivement égales

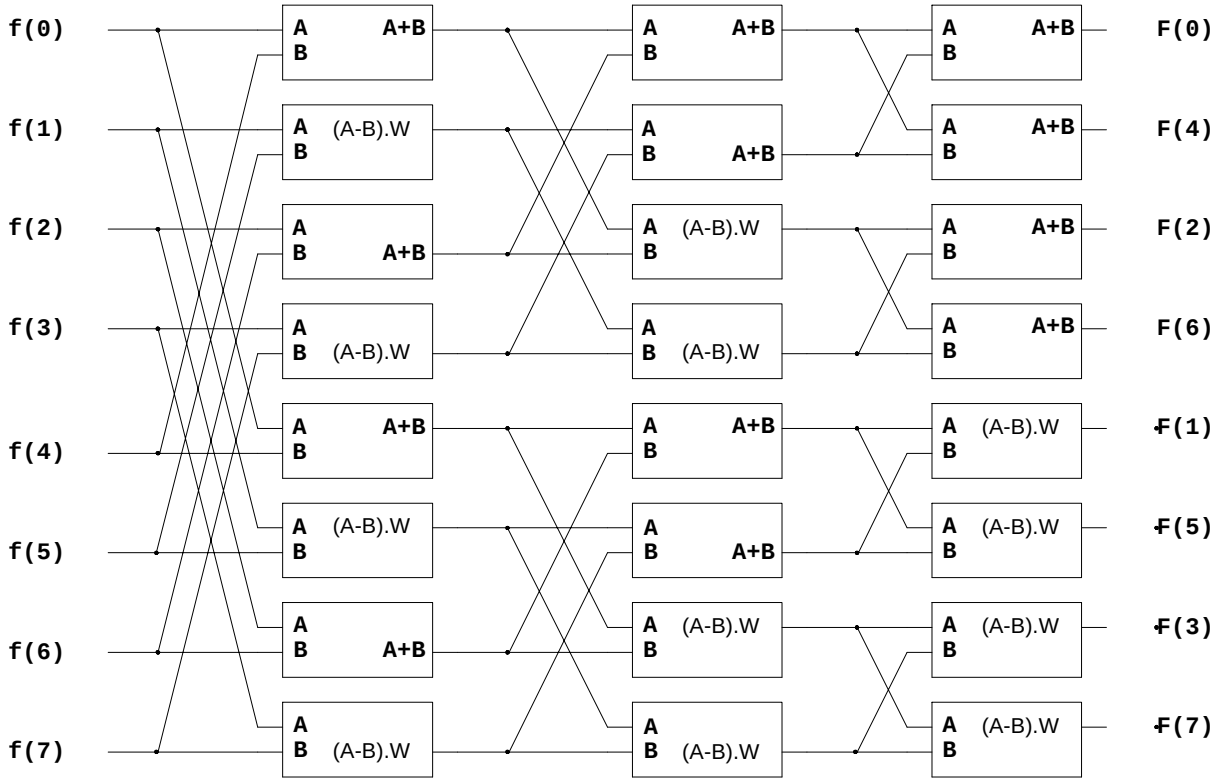

FIG. 2.5 – *schéma d'une T.F.R. de 8 échantillons à entrelacement fréquentiel dans la littérature traditionnelle.*

à W'', W' et W dans la décomposition temporelle. Cependant l'ensemble des valeurs qui apparaissent et leurs nombres respectifs d'apparition sont identiques dans les deux cas.

En comparant les figures 2.2 et 2.4 nous pouvons observer que les liens entre les cellules élémentaires de différents niveaux sont identiques pour les deux algorithmes lorsqu'ils sont implantés d'une façon cablée. Seules les équations à l'intérieur de chaque papillon change suivant le mode de décomposition.

I.2.3.2 Autres bases et règles d'emploi.

LA DÉCOMPOSITION EN BASE 2 À ÉTÉ ÉTENDUE À UNE BASE QUELCONQUE . Celle de base 4 [Joh84] amène des équations qui peuvent être regroupées sous le forme suivante :

$$\left(f(n) \pm f(n+\frac{N}{4}) + f(n+\frac{N}{2}) \pm f(n+\frac{3N}{4})\right).W^{n.k}$$

$$\left(f(n) \pm j.f(n+\frac{N}{4}) \Leftrightarrow f(n+\frac{N}{2}) \pm j.f(n+\frac{3N}{4})\right).W^{n.k}$$

C'est en fait la seule base au delà de 2 qui conduit à des simplifications évidentes dans la décomposition des termes exponentiels. En effet multiplier un nombre $a + j.b$ par j peut se ramener à transformer sa partie réelle en partie imaginaire et sa partie imaginaire en partie réelle en inversant son signe, ainsi qu'il est précisé ci-dessous :

$$j.(a + j.b) \quad = \quad a.j + b.j^2$$

$$= a.j \Leftrightarrow b$$
$$= \Leftrightarrow b + a.j$$

puisque $j^2 = \Leftrightarrow 1$.

Toutefois ces opérations, économes en surface d'implantation et en temps d'exécution pour une solution cablée, sont gourmandes en temps d'exécution pour une solution logicielle. Ces considérations ont amené les programmeurs à préférer les bases 2 et les concepteurs de circuits intégrés à privilégier les bases 4.

La figure 2.6 représente une T.F.R. de même taille, seize échantillons, pour les deux bases implantables de façon simple en circuit intégré. Le nombre de papillons pour une base 4 est plus faible, mais la structure interne d'un papillon et la connexion entre les papillons sont plus complexes. Une implantation cablée en base 4 amène toutefois un gain global important, en temps de calcul et en communications de données essentiellement. La complexité de la structure interne des papillons est due à un nombre plus grand d'opérations simples qui sont toutefois beaucoup moins coûteux en terme de surface qu'un multiplieur.

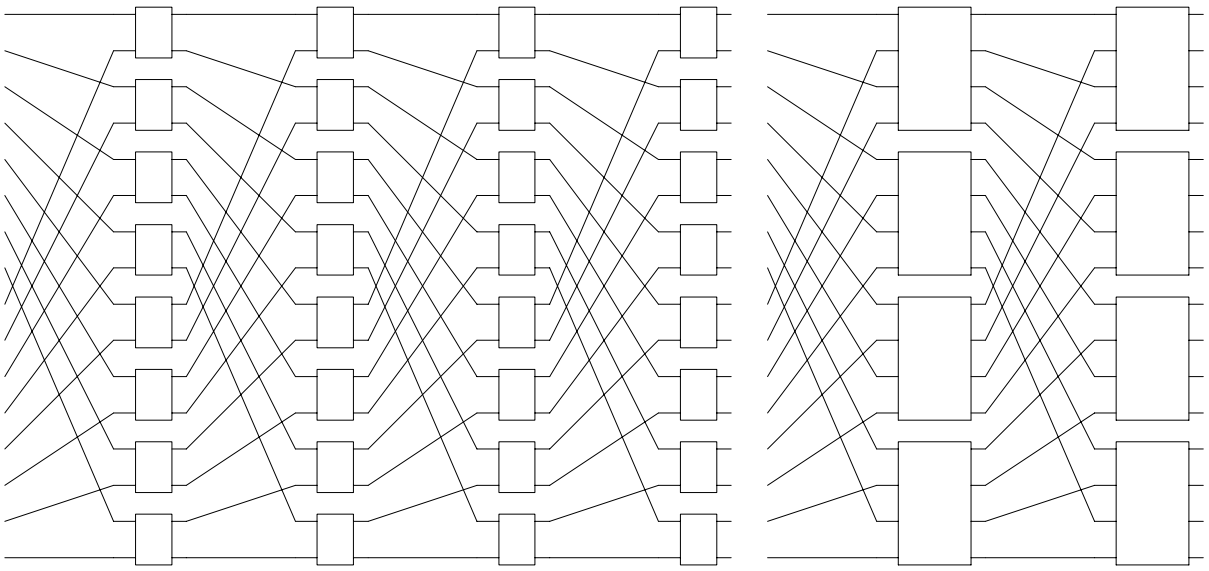

FIG. 2.6 – *T.F.R. de seize échantillons pour une base* 2 *et une base* 4.

Les *bases mixtes* sont constituées par l'emploi d'au moins deux bases pour la décomposition. En conséquence de ce qui vient d'être dit, une base double, 2 et 4, est employée dans des solutions cablée lorsqu'elle permet au nombre d'échantillons traités d'être compatible avec la base. C'est à dire lorsqu'il est de la forme $2 * 4^n$, toute puissance de deux supérieure à un pouvant être mise sous la forme d'un produit de 2 par une base 4. Une étape de calcul est faîte en base 2, les autres en base 4. Pour reprendre les exemples des paragraphes précédents illustrés par les figures 2.2 et 2.4, nous avons représenté dans la figure 2.7 une T.F.R. de base mixte pour huit échantillons.

I.2.3.3 Question de vocabulaire.

BEAUCOUP D'AUTEURS DÉSIGNENT EN FAIT SOUS LE MÊME TERME DE BASE , et sans plus de précision, deux termes qui sont identiques d'un point de vue purement mathématique, mais différents au point de vue calcul, le partitionnement et le calcul proprement dit. Lorsqu'une

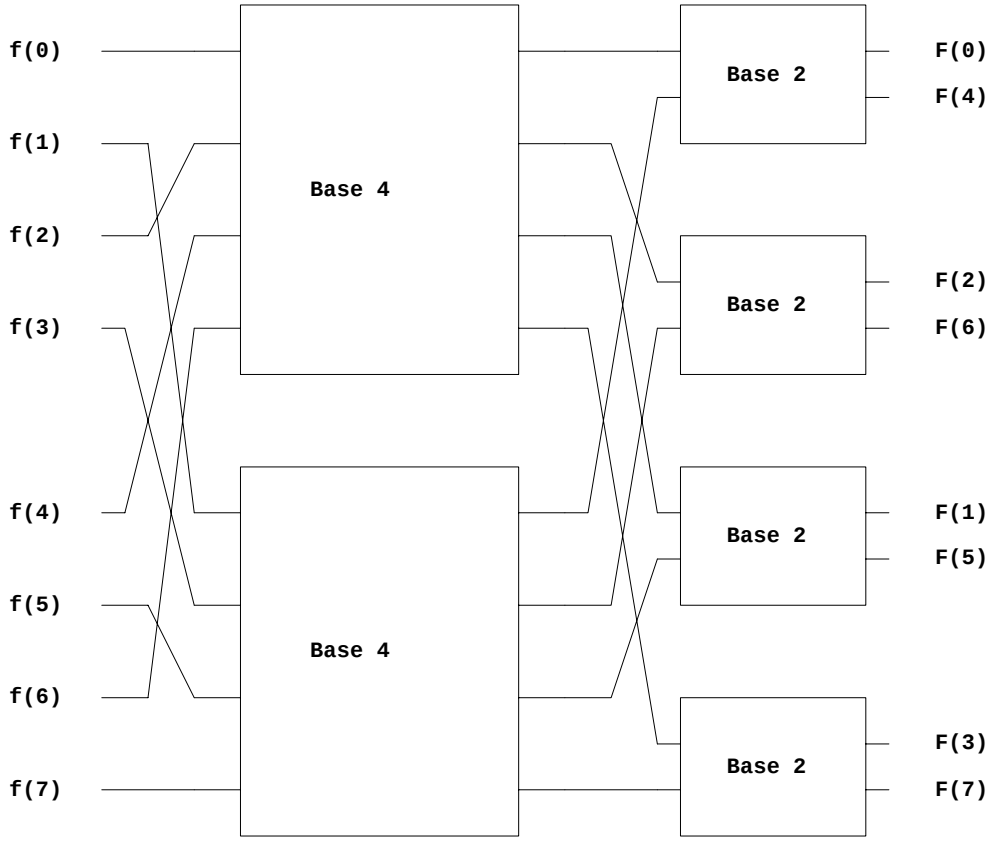

FIG. 2.7 – *schéma du traitement d'une T.F.R. de 8 échantillons avec une base mixte.*

transformée de Fourier comporte trop d'échantillons pour être calculée par un seul processeur ou pour être contenue dans la mémoire du processeur en question, elle est décomposée en transformées de taille plus faible qui peut être d'une taille trés supérieure à 4. Ceci qu'elles soient calculées chacune par un processeur différent dans le cas d'une architecture multiprocesseur ou par un seul et unique processeur, mais sans que celui-ci ait continuellement à faire appel à des mémoires lentes, comme des disques durs, dans le cas de transformées de grandes tailles disproportionnées avec la mémoire centrale du processeur. Par contre, lorsque le calcul arithmétique est réalisé, la base de décomposition est 2 ou 4. L'accroissement de la mémoire centrale des ordinateurs n'empêche pas ce phénomène. D'une part parce que la taille des T.F.R. croît au fil du temps en raison des exigences de leurs utilisateurs, d'autre part parce que les ordinateurs à forte taille mémoire centrale sont fréquemment multi-tâches, donc que la mémoire centrale réellement utile pour une tâche est plus faible que celle présente dans l'ordinateur. Pour prendre un exemple chiffré, une T.F.R. tridimensionnelle traitant une résolution de 256 échantillons par dimension et des nombres de 32 chiffres nécessite une mémoire de 64 mégaoctets. Si nous prenons le cas d'une T.F.R. $4 \Leftrightarrow D$ avec une résolution de 1024 échantillons par dimension et des nombres de 64 chiffres exige 8192 gigaoctets de mémoire.

Nous avions désigné par le terme de *pseudo-base* [VA94a] la base caractéristique de ces transformées réduites, mais nous préférons désormais l'appeler *macrobase*. Si nous poussons l'analogie avec la T.F.R. traditionnelle de base 2 ou 4, chacune de ces transformées de petite taille est équivalente à un papillon de la macrobase. À l'intérieur de ce papillon de la macrobase, le calcul est réalisé avec des papillons correspondant à une base de décomposition que nous appelons *base*

arithmétique, traditionnellement 2 ou 4.

I.2.4 Les moyens de calcul en Traitement du Signal.

POUR ACCROÎTRE LA VITESSE DE CALCUL, le domaine du traitement du signal nous offre trois possibilités de calcul :

1. un ordinateur généraliste
   - la solution la moins chère si l'installation n'est pas reproduite en un grand nombre d'exemplaires
   - utilisable à d'autres fins si l'utilisation qui en est faîte est peu intensive ou arrêtée
   - les fonctions du système sont trés peu utilisées hormis l'unité centrale et sa mémoire
2. un processeur de traitement du signal, plus couramment désigné par ses initiales anglo-saxonnes D.S.P. («Digital Signal Processor»)
   - l'architecture interne du circuit offre des fonctions cablées qui permettent d'accélérer les calculs
   - l'offre est vaste dans ce domaine, tant en ce qui concerne la taille des nombres (16, 24 ou 32 chiffres binaires) que leur format (virgule fixe ou flottante)
   - solution qui reste partiellement logicielle avec l'avantage de ne pas être figée tout en utilisant des circuits déja en production
3. une implantation cablée de la fonction souhaitée
   - la solution qui permet le plus court temps de traitement
   - un nombre d'échantillons réduit (d'un à quelques milliers pour les implantations actuellement disponibles commercialement qui ne font appel qu'à un seul circuit)
   - un choix limité pour la pécision des nombres traités (nombres ayant au plus 18 chiffres binaires pour une solution monopuce, *virgule glissante* au lieu d'une virgule flottante ; pour un ensemble de résultats d'une étape intermédiaire de la transformée de Fourier le dépassement sur l'un quelconque des calculs élementaires entraine la modification de la virgule pour l'ensemble considéré et donc de celles des résultats des étapes ultérieures qui en découlent).

Généralement ces deux dernières solutions concernent essentiellement des applications autonomes, citons la radio-diffusion et la télévision numériques ou les instruments de guerre électronique, mais peuvent être mises en oeuvre dans un ordinateur généraliste pour en faciliter le développement, bénéficier de ses interfaces et augmenter les performances de calcul de ce dernier. Celles-ci sont obtenues par association d'opérateurs spécialisés en plus ou moins grand nombre, mais jusqu'à présent seuls les processeurs de traitemant du signal ont été mises en oeuvre par ce biais, uniquement en raison des possibilités de vente estimées par les fabricants de ce genre de solution. Le facteur le plus limitant de ces circuits est la surface d'implantation nécessaire. Cela entraîne des conséquences sur le format des nombres qui ont été évoquées ci-dessus. Les processeurs de traitement du signal ont un marché suffisant pour mobiliser les fonds nécessaires au développement de solutions à virgule flottante. Les circuits de calcul cablés sont encore d'un emploi trop limité pour bénéficier des mêmes conditions et restent confinés au traitement de nombres à virgule glissante. Toutefois la vitesse qui est leur point fort nécessite d'évaluer les

dimensions	1D	2D	3D	4D
variables	x	(x,y) ou (x,t)	(x,y,z) ou (x,y,t)	(x,y,z,t)

TAB. 2.1 – *exemples de dimensions et de variables de transformées de Fourier.*

conséquences de ce fait sur la précision globale des calculs, en tenant compte des spécificités des données à traiter. Notons que le calcul d'une transformée de Fourier peut être décomposé pour respecter le nombre d'échantillons pouvant être traité par un circuit donné de ce type qui devient alors un papillon de la macrobase de la T.F.R. globale.

Ces toutes dernières années, le développement des D.S.P. a été tel que ces circuits offrent un compromis entre la puissance de calcul et le prix qui a élargi leur marché bien au-delà de leur domaine proprement dit, en particulier la commande numérique d'ensembles à réaction rapide. Cela ne fera qu'accentuer leur développement et leur diffusion.

I.2.5 Exemples de configuration de transformée de Fourier.

I.2.5.1 Dimensions et nombre d'échantillons.

POUR ILLUSTRER LES CONTRAINTES LIÉES À LA DIMENSION DE L'ESPACE DE TRAVAIL, nous pouvons examiner quelques exemples numériques de configurations correspondant à des cas de T.F.R. caractéristiques des différentes dimensions usuelles ou fréquemment souhaitées de l'espace mathématique de travail. Indiquons d'abord quelle est la signification des dimensions qui interviennent dans ces exemples :

- une transformée monodimensionelle porte en général sur le temps et son image fréquentielle, parfois sur une dimension géométrique
- une transformée bidimensionelle s'applique le plus souvent sur un plan qui est trés souvent une image photographique ou vidéo, parfois sur une dimension géométrique associée au temps
- une transformée tridimensionelle traite aussi bien des données liées au temps et à un plan, évolution d'une image photographique ou vidéo à travers le temps, que des données statiques dans l'espace géométrique
- une transformée quadridimensionelle traite généralement des données dynamiques dans l'espace géométrique.

Ces exemples d'utilisations actuelles ou prévisibles sont repris dans le tableau 2.1. Les variables x, y, z sont géométriques et la variable t est temporelle.

Pour les cas les plus typiques des utilisations actuelles ou prévisibles des dimensions les plus courantes, le tableau 2.2 indique la taille de la transformée notée N, c'est à dire le nombre d'échantillons concernés, la puissance de deux correspondante, notée M elle est égale au nombre de pas successifs de calcul pour une T.F.R. en base 2. Il est à remarquer que, pour les transformées multidimensionnelles, le grand nombre des échantillons entraine fréquemment une optimisation de celui-ci selon chaque dimension et, par conséquence, des transformées dont la taille sur chaque dimension n'est pas une valeur unique, donc non représentée sur ce tableau.

Pour rendre plus parlant ces chiffres, nous les avons représenté dans la figure 2.8. Le graphique représente le nombre d'échantillons d'une transformée en fonction du nombre d'échantillons par

N	1024	4096	16384	32768	65536	262144	1048576	2097152	16777216
M	10	12	14	15	16	18	20	21	24
1D	1024	4096	16384	32768	65536	262144	1048576	2097152	16777216
2D	32^2	64^2	128^2		256^2	512^2	1024^2		4096^2
3D		16^3		32^3		64^3		128^3	256^3
4D					16^4		32^4		64^4

TAB. 2.2 – *exemples de tailles et de dimensions de transformées de Fourier rapides.*

dimension, ceci pour les quatre cas dimensionnels évoqués. Les échelles sont logarithmiques en raison des écarts observés. Une implantation donnée autorise le calcul d'une T.F.R. d'un nombre d'échantillons donné. La dimension n'intervient pas dans ce nombre. Un utilisateur s'intéresse plutôt à la résolution de chaque dimension. Nous pouvons observer que la résolution par dimension, acceptable pour un calcul numérique, diminue trés rapidement avec le nombre de dimensions de l'espace de travail. Une T.F.R. de 16 millions de points est déja trés rare d'emploi en raison de sa taille, c'est pour cela que nous nous sommes limités à ce nombre. Il ne permet pourtant qu'une résolution trés faible, même dans un espace tridimensionnel.

Pour prendre un exemple, l'étude dynamique d'une image vidéo avec 16 millions de points, soit le temps et les deux dimensions géométriques, n'autorise qu'une résolution par dimension de 256 points. Ce qui est trés inférieur à celle des images les plus banales. En conservant une définition identique sur les trois axes de travail, cela donnerait une séquence vidéo formée d'images d'une définition de 256 prises à la cadence de 25 images par seconde pendant dix secondes ou une séquence d'images type VGA 640×480 pendant seulement deux secondes. Citons le cas de figure un peu identique de la vidéoconférence ou vidéophonie où le problème de la quantité de données à traiter est identique et les résultats aussi mauvais actuellement, en ce qui concerne la définition obtenue.

I.2.5.2 Taille des nombres codant les échantillons.

NOUS NE CONSIDÉRONS PAS LE CAS D'UNE NOTATION À VIRGULE FLOTTANTE qui est trop gourmande en surface d'implantation pour être envisagée pour des applications dédiées à hautes performances avec des coûts raisonnables. Pour les différents circuits de calcul cablé de transformées de Fourier qui ont pu être réalisé industriellement dans un passé récent, la taille des opérandes correspondent à la précision des convertisseurs analogique-numérique disponibles dans le même temps. Nous choisissons donc des exemples représentatifs de taille des nombres codant les échantillons et indiquons les applications les plus typiques qui leurs sont associées. Ce qui nous donne, en mettant de côté le traitement du signe, les cas de figure suivants pour les nombres :

- 8 chiffres binaires, trés utilisés pour les conversions analogique-numérique rapides (quelques centaines de mégahertzs)

- 12 chiffres binaires pour des conversions à moyenne vitesse (quelques dizaines de mégahertzs)

- 16 chiffres binaires pour les conversions à haute résolution (audio ou video)

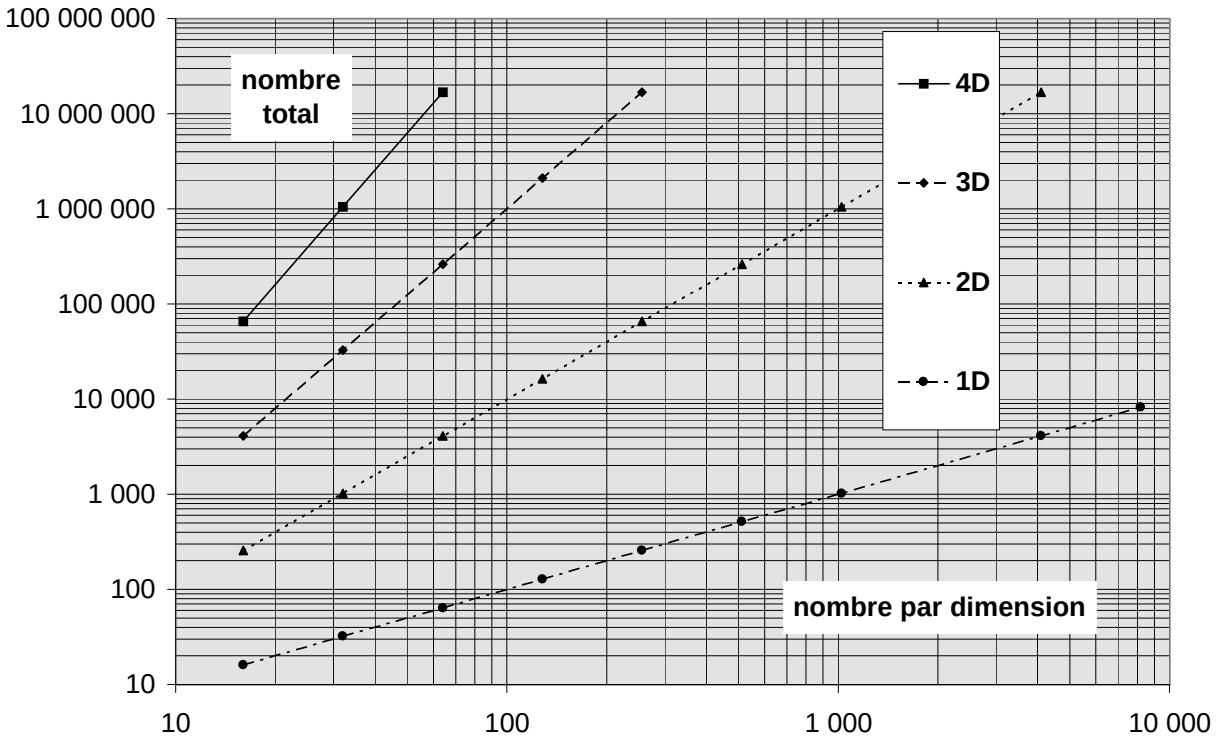

FIG. 2.8 – *nombres caractéristiques d'échantillons d'une transformée de Fourier.*

– 24 chiffres binaires pour les conversions à trés haute résolution (instrumentation de laboratoire).

Évidemment la répétition des calculs sur ces nombres et leurs dérivés imposent pour améliorer la précision l'utilisation d'opérandes plus grands, méthode coûteuse en terme de surface d'implantation, ou de techniques comme la virgule glissante, couteuse en terme de complexité de commande, donc de temps d'exécution pour des opérateurs parallèles et de retards pour des opérateurs sériels.

La virgule glissante a la préférence des solutions commerciales, car elles sont basées sur des opérateurs parallèles. Si les opérandes sont constitués de B chiffres binaires, la surface d'implantation des multiplieurs parallèles est proportionnelle à B^2 ou $B \times logB$. Toute augmentation de B pour satisfaire des besoins éventuels de précision est trés coûteuse en terme de surface. Par contre les opérateurs sériels qui constituent le fond de commerce de l'équipe qui a accueilli la préparation de cette thèse ont une surface seulement proportionnelle à B, d'où un coût nettement moindre. Un rapide coup d'oeil à la figure 2.9 nous permet d'évaluer les augmentations respectives du coefficient de proportionnalité de la surface pour une augmentation de la taille des opérandes de quelques chiffres binaires. Pour passer de 18 à 24 bits par exemple, il faut pour les opérateurs proportionnels à B, $B \times logB$ et B^2 respectivement 33%, 47% et 78% de surface en plus. Précisons qu'il s'agit d'une augmentation relative par rapport à la solution de même coefficient de proportionnalité ayant des opérandes plus petits. La surface d'une solution est égale au produit du coefficient de proportionnalité et d'un terme constant à l'intérieur d'une famille d'opérateurs, mais qui diffère d'une famille à l'autre. C'est pour limiter la surface d'implantation que nous étions partis sur l'a priori de l'emploi de nombres à virgule fixe d'une taille de 32 chiffres binaires et d'opérateurs sériels pour le travail en milieu scientifique. Au détriment du temps de calcul, proportionnel à B pour les opérateurs sériels. Le produit $S \times t$ utilisé pour comparer des

architectures concurrentes n'est pas utilisé dans notre approche.

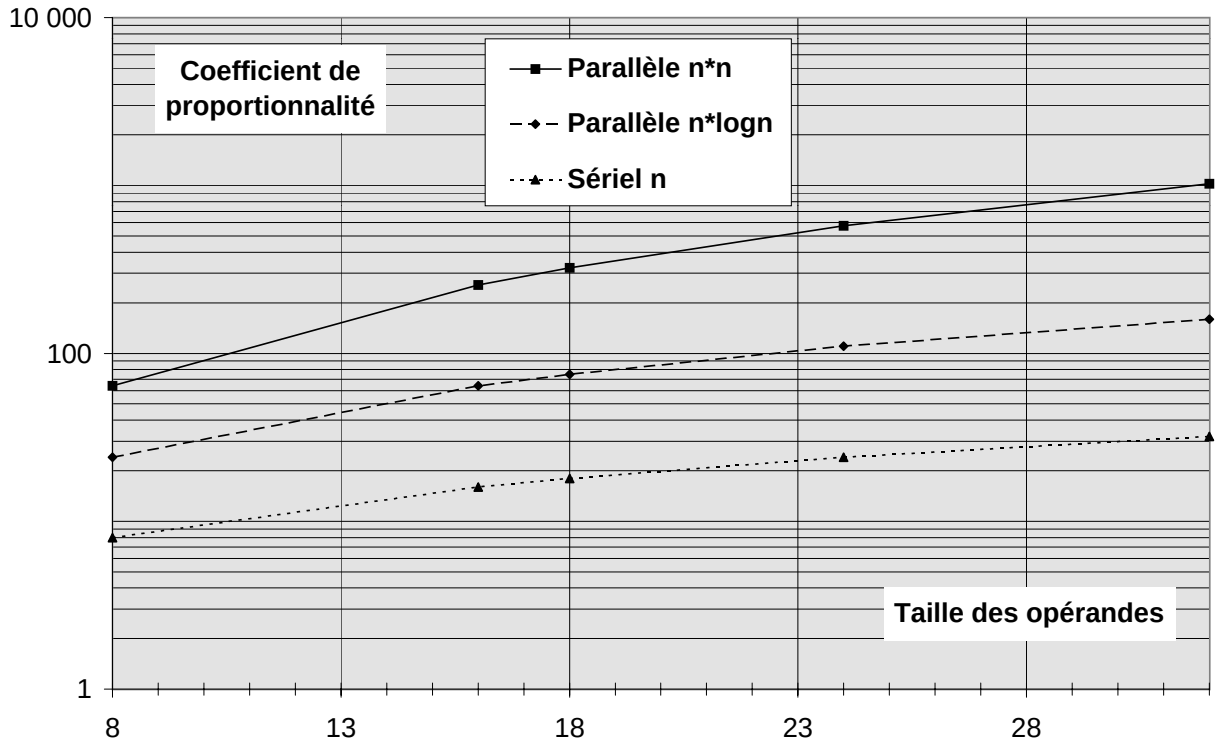

FIG. 2.9 – *coefficients de proportionnalité de différents types de multiplieurs en fonction de la taille des opérandes.*

I.2.5.3 Circuits existants à un stade industriel.

POUR ILLUSTRER NOS DIRES PRÉCÉDENTS , citons les produits commercialisés au cours des dernières années ou qui ont atteint un développement tel qu'ils pourraient l'être à terme. Dans le tableau 2.3, nous avons porté, outre le nom du fabricant et la taille maximale de la T.F.R. calculable par le circuit en question, la précision des opérandes et le marché visé. Comme nous l'avons déja signalé, le nombre d'échantillons et la taille des opérandes ont de faibles valeurs, hormis le produit Sharp qui fait appel à plusieurs puces.

I.2.6 Implantation cablée.

I.2.6.1 T.F.R. pipelinée sans et avec rebouclage

SI LE NOMBRE D'ÉCHANTILLONS À TRAITER N est une puissance de p qui est appelé la base de la transformée, le calcul est exécuté d'une façon générale en $\log_p N$ étapes de calcul, constituées chacune de $\frac{N}{p}$ ensembles de calculs élémentaires. Chacun de ces ensembles est réalisé par un papillon. Cela mêne donc à $(\frac{N}{p} * \log_p N)$ ensembles de calculs élémentaires dont les expressions diffèrent selon la version choisie de l'algorithme et sa base. La réduction du nombre de ces ensembles de calculs élémentaires en fonction de p s'accompagne d'une augmentation de la complexité des expressions mises en oeuvre et des connexions entre papillons de deux étapes successives, car chacun fait intervenir plus de variables

Industriel concerné	pays	taille de la T.F.R	taille des opérandes	marché visé
CNET (France Télécom)	France	8192	16 bits	télécommunications et télédiffusion
Dassault Électronique	France	1024	12 bits	militaire
Gec Plessey	Grande-Bretagne	1024	18 bits	militaire et industriel
Sharp	Japon	adaptable	24 bits	industriel et médical

TAB. 2.3 – *circuits ayant atteint un stade industrialisable aux cours des dernières années.*

Pour une base 2, nous avons donc à réaliser l'implantation d'une fonction mathématique réalisée en $\log_2 N$ étapes successives similaires. Chacune est constituée de $\frac{N}{2}$ calculs élémentaires indépendants réalisés par un papillon dont les entrées sont soit des constantes qui sont les coefficients exponentiels des formules mathématiques, soit des résultats de papillons de l'étape précédente de calcul. Une T.F.R. peut donc être parallélisée au sein de chaque étape. Ce qui nous donne le schéma de la figure 2.10 si chaque papillon est dédié à un calcul donné d'une étape donnée. Un transfert parallèle des données entre des papillons successifs créerait un faisceau de connexions trop dense dans le circuit. Une solution basée sur des opérateurs série s'impose donc. La relative lenteur des opérateurs série est compensée par leur nombre possible supposé important.

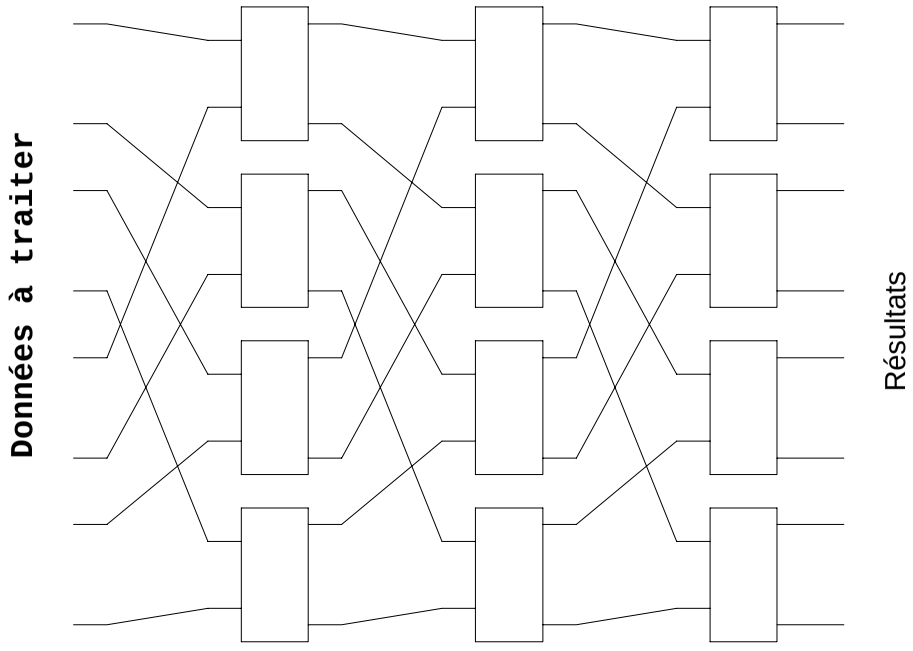

FIG. 2.10 – *T.F.R. pipelinée, parallélisée et totalement implantée.*

Si une seule T.F.R. doit être calculée, les opérateurs sont en sommeil, excepté la barette correspondant à l'étape en cours. Si des bascules à la sortie des papillons mémorisent les résultats, la surface du circuit peut être considérablement réduite en se contentant d'une seule barette de papillons et en changeant les valeurs des coefficients exponentiels à chaque étape du calcul. Un rebouclage des sorties sur les entrées permet de conserver un seul étage du circuit précédent. Un multiplexeur placé sur les entrées des données à traiter permet de sélectionner soit une donnée venant de l'extérieur, cas de la première étape du calcul, soit une donnée fournie par la sortie d'un papillon, cas des étapes suivantes. Ce qui nous donne la figure 2.11.

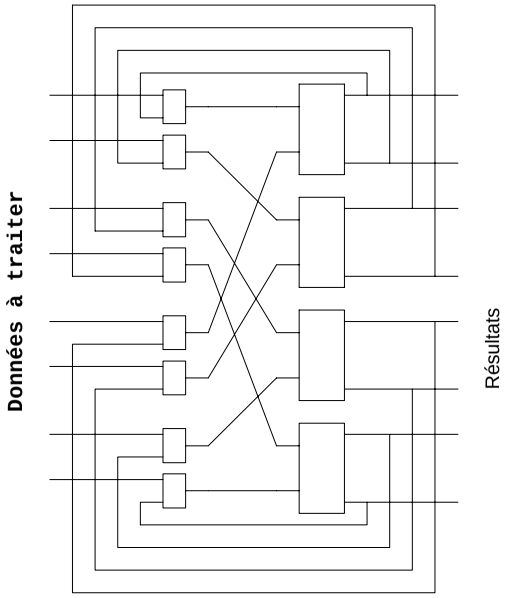

FIG. 2.11 – *T.F.R. pipelinée rebouclée.*

La redistribution des résultats intermédiaires peut être fait de différentes manières. La plus simple consiste à utiliser ce qu'on appelle couramment de sa dénomination anglo-saxonne *perfect shuffle exchange* [Sto71] [Par80] qui apparaît dans les figures 2.2 et 2.4. Il s'agit d'un réseau de communication qui assure en même temps et simplement la distribution et le reordonnancement des données entre deux étages de papillons. Lorsque le nombre de connexions est important, il a l'inconvénient d'être encombrant, nécessitant une surface proportionnelle au nombre d'échantillons. Toutefois pour des implantations comme celle décrite au chapître 4 où ce nombre est réduit, ce critère n'intervient que peu et il a l'avantage de la simplicité.

I.2.6.2 Papillon en base 2.

LES EXPRESSIONS À CALCULER sont, par assimilation aux variables dont elles découlent et aux fonctions auxquelles elles mènent, de la forme $F_{pap}(k) = [f_{pap}(n) \pm f_{pap}(n+\frac{N}{2}) * \omega^{n.k}]$. Nous désignons par $f_{pap}(n)$ les variables d'entrées des papillons et par $F_{pap}(k)$ les fonctions de sorties correspondantes et nous posons $\exp^{-2\pi j(n*k/N)} = \omega^{n.k}$. Comme tous les nombres sont complexes et que les opérateurs disponibles sont conçus à partir d'opérateurs élémentaires réels, nous devons décomposer ces expressions pour les implanter.

Soient a, b, c, d, e, f des nombres réels que nous emploierons pour simplifier la démontration, avec :

$$\begin{aligned} a &= \text{partie réelle de } f_{pap}(n) \\ b &= \text{partie imaginaire de } f_{pap}(n) \\ c &= \text{partie réelle de } f_{pap}(n+\frac{N}{2}) \\ d &= \text{partie imaginaire de } f_{pap}(n+\frac{N}{2}) \\ e &= \text{partie réelle de } \omega^{n.k} \\ f &= \text{partie imaginaire de } \omega^{n.k} \end{aligned}$$

Les expressions sont donc de la forme :

$$\begin{aligned} (a+jb) \pm (c+jd).(e+jf) &= a+jb \pm (c.e+jcf+jd.e+j^2d.f) \\ &= a+jb \pm c.e \pm jcf \pm jd.e \mp d.f \\ &= a \pm (c.e \Leftrightarrow d.f) + j[b \pm (cf+d.e)] \end{aligned}$$

Ce qui nous donne les expressions élémentaires suivantes à calculer :

$$\begin{aligned} a+(c.e \Leftrightarrow d.f) &= \text{partie réelle de } F_{pap}(k) \\ b+(cf+d.e) &= \text{partie imaginaire de } F_{pap}(k) \\ a \Leftrightarrow (c.e \Leftrightarrow d.f) &= \text{partie réelle de } F_{pap}(k+\frac{N}{2}) \\ b \Leftrightarrow (c.f+d.e) &= \text{partie imaginaire de } F_{pap}(k+\frac{N}{2}) \end{aligned}$$

Pour calculer les expressions $[f_{pap}(n) \pm f_{pap}(n+\frac{N}{2}) * \omega^{n.k}]$, il nous faut donc quatre multiplieurs et six additionneurs réels. Le schéma général est donné par la figure 2.12.

I.2.7 Conclusion.

Le rappel des bases de la transformée de Fourier qui autorisent la mise en oeuvre de la transformée de Fourier rapide nous a permis de distinguer les raisonnements liés à la nature de l'implantation de la T.F.R., cablée ou logicielle, et de mettre en évidence les notions de base arithmétique et de macrobase. Notions fondamentales dans le cas des architectures cablées, elles peuvent passer totalement inaperçues dans le cas d'une solution logicielle. En effet, les ordinateurs actuels ne permettent pas d'effectuer des arrondis aprés une suite de calculs successifs pour plusieurs traitements en parallèle comme le permet une solution développée à la demande. La notion de bases arithmétiques supérieures à 2 n'a donc plus grand sens. Il s'agit plutôt de macrobase de second niveau, de même qu'il existe des caches de niveaux différents dans les ordinateurs entre le processeur et la mémoire centrale.

Un tour d'horizon des applications éventuelles d'une T.F.R. cablée, en terme de caractéristiques des données à traiter, et des contraintes des implantations cablées, essentiellement la surface, nous donnent les ordres de grandeur des caractéristiques d'une architecture cablée à étudier. Les compromis choisis amènent des opérandes de 32 chiffres binaires pour des échantillons dont le nombre est supérieur au million, loin du millier autorisé par les circuits industriels

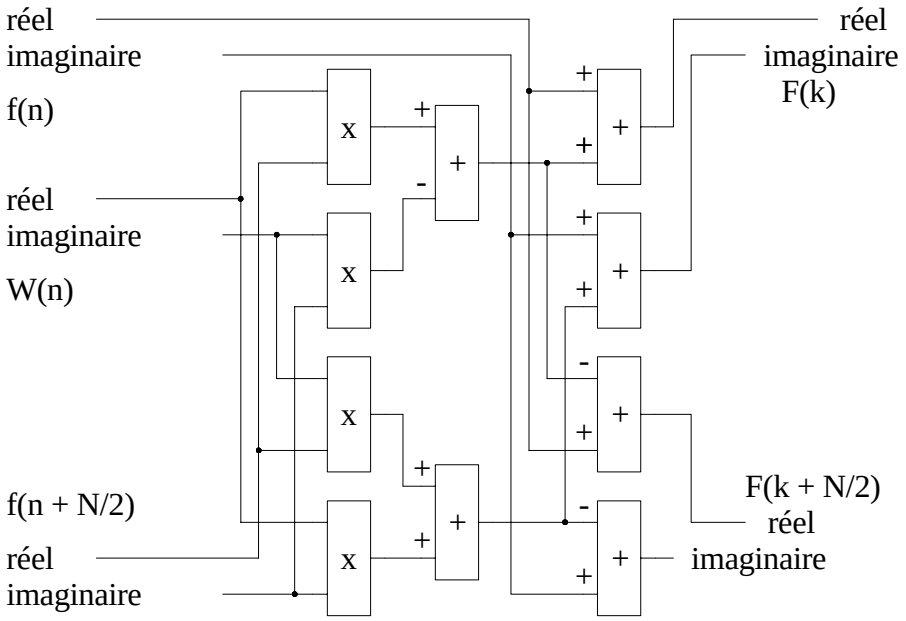

FIG. 2.12 – *papillon de TFR en base 2.*

existants. Le choix des opérateurs sériels qui ont été étudiés dans le passé pour ce genre d'applications [Meh83] [Meh91] avec les limites dûes à la technologie de l'époque nécessite l'optimisation de la surface, en utilisant en particulier la structure mathématique des équations à implanter qui apparaissent dans la structure de calcul de base qui forme le papillon.

Chapitre 3

Évaluation du calcul d'une T.F.R. en trois dimensions

Sommaire

I.3.1 Introduction. **35**
I.3.2 Calcul d'une T.F.R. $3D$ grâce à des T.F.R. $1D$. **36**
I.3.3 Notation avec des chiffres signés. **38**
I.3.4 Multiplieurs à notation en complément à deux à trés haute vitesse. **39**
I.3.5 Influence des notations arithmétiques sur une architecture. **43**
I.3.5.1 Performances théoriques temporelles. 43
I.3.5.2 Performances d'opérateurs arithmétiques conçus, fabriqués et testés. . . 46
I.3.5.3 Coûts en surface d'implantation. 49
I.3.6 Perspectives. **51**

I.3.1 Introduction.

Le calcul d'une t.f.r. multidimensionnelle par le biais de T.F.R. monodimensionnele est une technique connue comme nous le rappelons dans le paragraphe 3.2. Cette voie semble royale pour une implantation cablée, car toutes les implantations cablées réalisées jusqu'à ce jour ne permettent que des calculs sur des nombres réduits d'échantillons, du moins à l'échelle d'une transformée multidimensionnelle. Les implantations à base d'opérateurs sériels utilisant des chiffres signés constituent l'épine dorsale des divers travaux menés dans l'équipe qui a accueilli nos travaux. Nous en présentons trés brièvement les principes dans le paragraphe 3.3 et nous reviendrons plus en détail sur cette notation dans le chapître 4. Ils permettent des temps de calcul intéressants, encore faudrait-il prendre le temps de comparer ces solutions à celles qui utiliseraient des opérateurs avec des notations en complément à deux. Pour ce faire, nous décrivons dans le paragraphe 3.4 des multiplieurs à trés hautes performances en terme de vitesse. Le paragraphe 3.5.1 nous permet de mettre en valeur les attraits des notations redondantes qui utilisent des chiffres signés sur lesquelles nous revenons plus en détail dans le chapître 4 et d'en relativiser l'importance. La vitesse ayant un coût en surface d'implantation, nous nous intéressons dans le paragraphe 3.5.3 à la faisabilité des diverses solutions envisageables. Le paragraphe 3.6 nous amène à étudier les performances globales d'une implantation pour une T.F.R. $3D$ en fonction de

divers paramètres sur lesquels un utilisateur pourrait jouer selon ... les moyens financiers qu'il accepterait de mettre en oeuvre.

I.3.2 Calcul d'une T.F.R. $3D$ grâce à des T.F.R. $1D$.

UNE TRANSFORMÉE DE FOURIER À m DIMENSIONS peut être mise sous la forme d'une cascade d'intégrales ayant toutes la forme d'une transformée monodimensionnelle comme nous l'avons rappelé dans la formule 2.2. Lorsque les conditions pour réaliser ce type de calcul sous forme discrète sont réunies, cette formule devient :

$$F(k_{m-1},\ldots,k_0)=\sum_{n_0=0}^{N_0-1}\left[\ldots\left(\sum_{n_{m-1}=0}^{N_{m-1}-1}f(n_{m-1},\ldots,n0).e^{-2\pi j\frac{n_{m-1}.k_{m-1}}{N_{m-1}}}\right).....\right].e^{-2\pi j\frac{n_0.k_0}{N_0}}$$

où N_i est le nombre de valeurs discrètes des variables correspondant à la i^{eme} direction de l'espace, n_i et k_i les équivalents discrets et normalisés de x_i et y_i respectivement.

Si nous restreignons le problème à seulement trois dimensions et si chaque N_i est égal à une valeur unique N, la formule précédente devient :

$$F(k_3,k_2,k_1)=\sum_{n_3=0}^{N-1}\left[\sum_{n_2=0}^{N-1}\left(\sum_{n1=0}^{N-1}f(n_3,n_2,n_1).e^{-2\pi jn_1k_1}\right).e^{-2\pi jn_2k_2}\right].e^{-2\pi jn_3k_3}$$

Cette écriture d'une transformée de Fourier tridimensionnelle sous la forme d'une poupée russe dont la forme générale est une transformée monodimensionnelle permet de la calculer grâce à trois transformées monodimensionnelles successives [*et al.*84].

Les quelques circuits intégrés ayant été développés pour calculer une transformée de Fourier n'intégrant au mieux que quelques milliers d'échantillons dans une seule puce, il est actuellement illusoire d'espérer calculer une transformée $3D$ avec un seul circuit. Le fait de calculer une transformée $3D$ par le moyen de trois transformées $1D$ successives est donc un moyen aisé de décomposer la transformée globale en transformées de plus petite taille. Une conséquence heureuse de cette stratégie est la gestion trés simple des indices des échantillons et résultats intermédiaires à faire intervenir dans les transformées successives. Évidemment chaque transformée peut être calculée par une T.F.R. pour peu que N remplisse la condition d'être une puissance de deux.

Chaque série de T.F.R. $1D$ successives est calculée selon une variable différente de l'indice. Lors de la première étape, chaque T.F.R. traite les différentes valeurs de n_1, pour une valeur donnée de n_2 et de n_3. Les résultats intermédiaires produits correspondent à l'indice k_1. Lors de la seconde étape, n_3 et k_1 sont les paramètres et n_2 la variable du calcul. La troisième et dernière étape traite n_3, k_1 et k_2 étant fixés. Cela est résumé dans la figure 3.1.

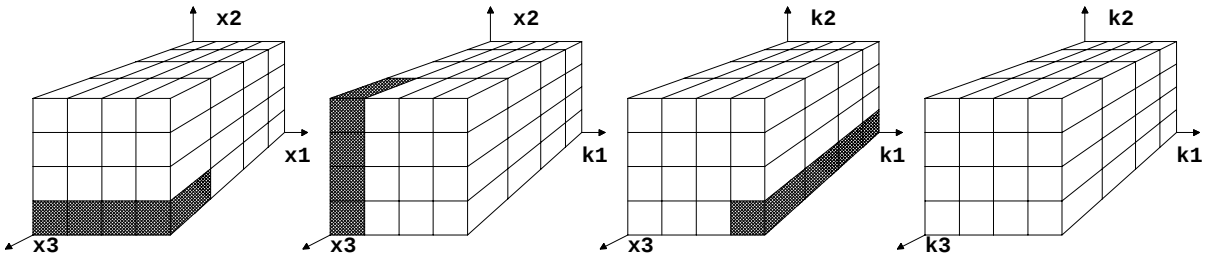

FIG. 3.1 – *l'espace des données lors de chaque T.F.R. $1D$ et aprés la T.F.R. $3D$ globale.*

Les données de l'une des T.F.R. en train d'être calculée lors de l'étape considérée sont représentées en grisé. Nous pouvons remarquer que la structure des données traitées ensemble subit une rotation dans l'espace au fil des étapes. Le résultat serait identique si les données subissaient une rotation entre chaque étape et que la structure de sélection des données pour chaque transformée élémentaire ait une orientation constante dans l'espace [GS87] comme le décrit la figure 3.2. Cela permet de grouper deux opérations distinctes dans l'adressage des données au cours de ce calcul. D'abord leur rearrangement au sein d'une T.F.R. dû à leur entrelacement, ensuite la redistribution entre les différentes T.F.R. d'une étape.

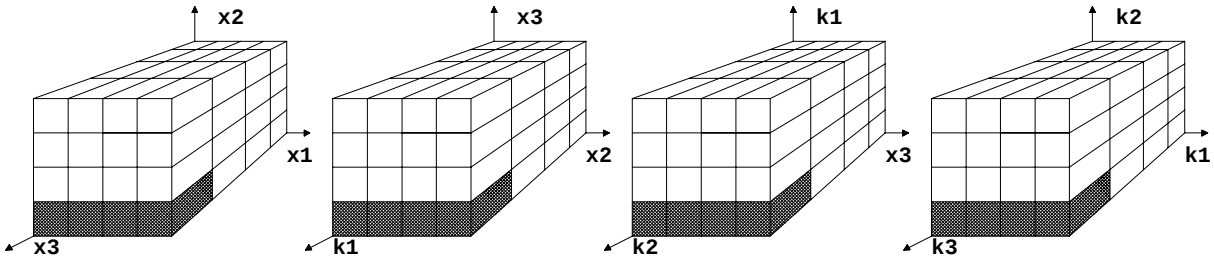

FIG. 3.2 – *rotation des données dans l'espace lors d'une T.F.R. 3D.*

Dans une implantation trés large, toutes les T.F.R. $1D$ d'une étape sont calculées simultanément. Les résultats sont ensuite rearrangés pour alimenter celles de l'étape suivante selon l'architecture représentée dans la figure 3.3.

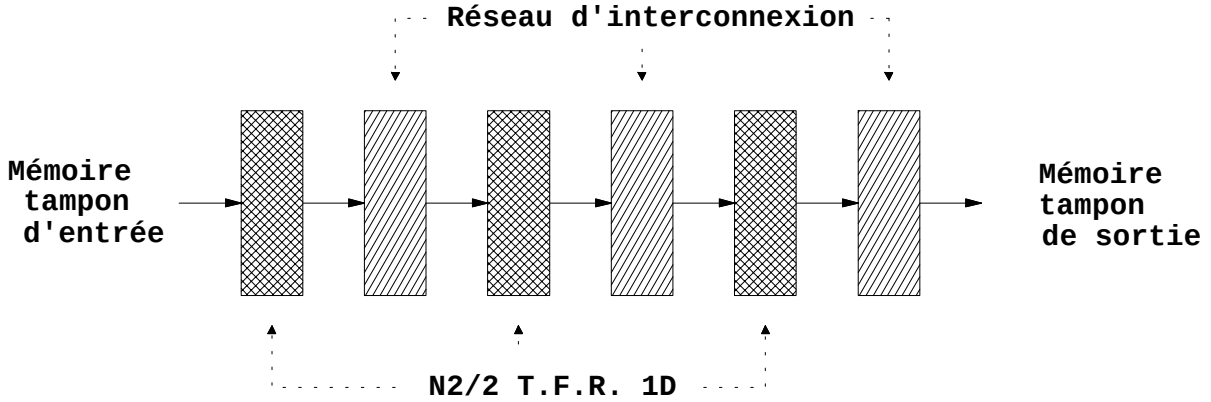

FIG. 3.3 – *architecture généreuse pour le calcul d'une T.F.R. 3D.*

Évidemment le coût d'une telle solution impose de réfléchir à des moyens plus économiques. Comme chaque étape est identique aux autres, le principe des architectures repliées rappelées au paragraphe 2.6.1 peut être étendu ici, les papillons étant remplacés par les T.F.R. $1D$. Ce qui donne l'architecture de la figure 3.4. Comme nous traitons N^3 valeurs à travers des T.F.R. de N points, il faut N^2 T.F.R. élémentaires.

Si nous prenons le cas d'une T.F.R. de 16 millions de points, cela mène à des T.F.R. $1D$ de 256 points, ce qui est tout à fait réaliste, mais demande 65536 T.F.R. $1D$, soit quelques milliers de puces à rassembler. Même si des solutions à base d'encapsulation multipuce, généralement appelées *Multi-Chip Modules*, sont envisageables, il est préférable d'envisager une solution plus rustique, au moins dans un cadre de recherche, où le calcul des T.F.R. serait multiplexé entre

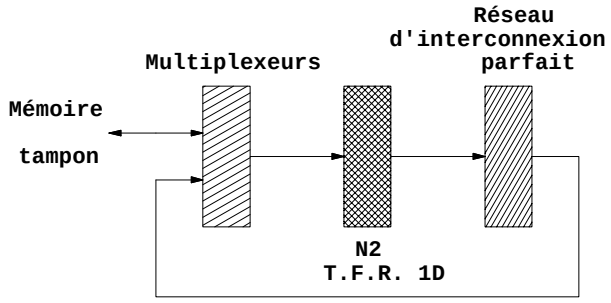

FIG. 3.4 – *architecture simplifiée pour une T.F.R. 3D.*

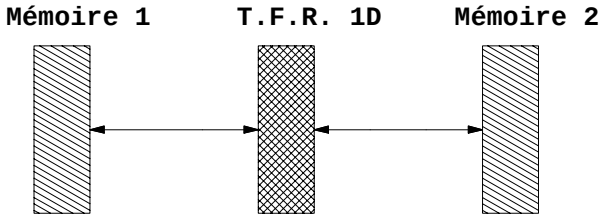

FIG. 3.5 – *architecture ping-pong pour une T.F.R. 3D.*

un ou quelques circuits. L'utilisation de deux mémoires entre lesquelles les données sont renvoyées au cours des étapes, un peu comme une balle de ping-pong, est illustrée dans la figure 3.5. Le rearrangement des données est ramené à un problème de gestion d'adresse d'une mémoire. L'utilisation de deux mémoires permet d'éviter tout conflit entre les entrées et les sorties des étages de calcul. Chaque adresse est l'image des trois indices, sa gestion est réduite à une traduction des trois parties la constituant : la rotation des données dans l'espace et éventuellement le rearrangement des données dû à l'entrelacement s'il n'est pas prévu dans le circuit de T.F.R. lui-même.

I.3.3 Notation avec des chiffres signés.

POUR JUSTIFIER LES CARACTÉRISTIQUES mises en oeuvre dans les circuits à notation à chiffres signés et utilisées dans nos comparaisons avec les circuits à notation en complément à deux, nous présentons cette technique à travers une addition de deux nombres de trois chiffres. Pour ne pas manier des variables logiques qui gardent malgré tout un aspect ésotérique, nous traitons le cas de nombres décimaux.

Soit deux nombres $x_2\ x_1\ x_0$ et $y_2\ y_1\ y_0$, dont l'addition donne le terme $z_3\ z_2\ z_1\ z_0$. Notons r_i la retenue générée par les chiffres x_i et y_i. Nous avons les sommes partielles :

$$x_i + y_i + r_{i-1} = 10r_i + z_i$$

. Précisons les :

$$\begin{aligned} x_2 + y_2 + r_1 &= 10r_2 + z_2 \\ &= z_3\ z_2 \\ x_1 + y_1 + r_0 &= 10r_1 + z_1 \\ x_0 + y_0 &= 10r_0 + z_0 \end{aligned}$$

Une addition telle que nous l'avons appris à l'école primaire se réalise en commençant par la droite. Essayons de la faire en commençant par la gauche, c'est à dire par le poids le plus fort. Nous devons ajouter $x_2 + y_2 + r_1$ en ignorant la valeur de r_1. Si nous connaissons x_1, y_1, bien que ne connaissant pas r_0, nous pouvons faire des hypothèses sur la valeur de r_1 :

- comme r_0 ne peut valoir que 0 ou 1 dans une addition de deux nombres, si $x_1 + y_1$ est inférieur à 9, alors $x_1 + y_1 + r_0$ est inférieur à 10. Donc r_1 vaut 0.
- Si $x_1 + y_1$ est supérieur ou égal à 9, alors $x_1 + y_1 + r_0$ risque d'être supérieur à 10. Donc r_1 peut valoir 1 et nous forçons r_1 à 1.

Dans l'étape suivante, c'est à dire l'addition de x_1 ety_1, nous opérons le même raisonnement en se souvenant que nous avons généré une retenue peut-être à tort. Si nous avons supposé que r_1 peut valoir 1 à priori et qu'il se révèle égal à zéro, c'est que nous nous trouvons dans la configuration $x_1 + y_1 + r_0 = 9 + 0$. Nous pouvons annuler cette retenue inutile avec un chiffre négatif, car $9 = 10 \Leftrightarrow 1$:

$$\begin{aligned} x_1 + y_1 + r_0 &= 9 \\ &= 9 + 0 \\ &= 10 \Leftrightarrow 1 \\ &= 10 \times 1 + (\Leftrightarrow 1) \\ &= 10 \times r_1 + z_1 \end{aligned}$$

Nous avons ainsi la possibilité de faire des opérations en commençant par les chiffres de poids le plus ou fort. Une autre conséquence de ce phénomène est la possibilité d'exécuter des additions en parallèle sans se soucier de la propagation des retenues. Contrairement à ce qui est généralement dit, une notation redondante n'empêche pas la propagation des retenues. Elle crée plus de retenues qu'il n'en faut, ce qui se traduit par une anticipation des retenues nécessaires, noyées dans celles qui ont été introduites inutilement.

I.3.4 Multiplieurs à notation en complément à deux à trés haute vitesse.

LA NOTATION REDONDANTE A COMME PRINCIPAL AVANTAGE de contenir la propagation des retenues au voisinage immédiat de leurs sources. La conséquence la plus intéressante pour les opérateurs sériels est la possibilité de faire débuter des opérations comme l'addition ou la multiplication par les chiffres de poids forts. La notation en complément à deux n'a pas cette possibilité. Toutefois une remarque astucieuse peut être faîte et exploitée dans certains cas particuliers. Si nous comparons un additionneur en série, représenté sur la figure 3.6, et un en parallèle, figure 3.7, le premier fonctionne à une fréquence supérieure au second, car la retenue ne se propage que d'un rang à la fois à travers la bascule de rebouclage. Le second est évidemment plus rapide puisqu'il traite tous les nombres en une seule opération, et ce même si la fréquence de l'horloge est plus faible.

Un multiplieur série-parallèle est constitué :

- d'un additionneur parallèle qui agit si le chiffre du multiplicateur le nécessite,
- d'un registre contenant le multiplicande

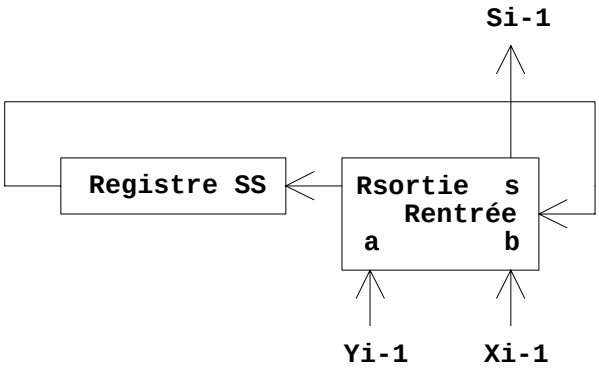

FIG. 3.6 – *additionneur sériel.*

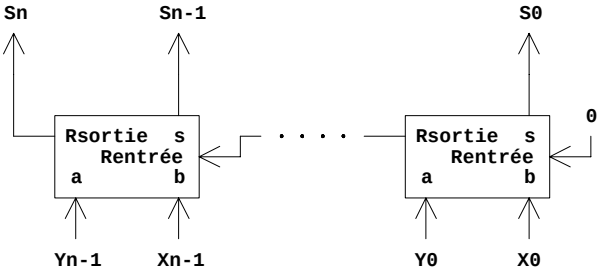

FIG. 3.7 – *additionneur parallèle.*

– d'un registre contenant le résultat, partiel jusqu'à la dernière opération.

De nombreux travaux ont eu les multiplieurs pour sujet, notamment dans leurs applications au traitement du signal [Lyo81] [Lyo84] [Lyo76]. Un moyen d'accélerer la fréquence de travail d'un multiplieur parallèle-série utilisant une notation en complément à deux consiste à imposer à la retenue de se déplacer d'un seul rang lors de chaque addition parallèle successive. Ainsi l'opérateur peut travailler à une fréquence plus haute, identique à celle d'un additionneur sériel. Le problème est que lors du traitement du dernier chiffre, une retenue peut encore à avoir à parcourir toute la partie de plus fort poids du résultat. Dans le cas où le multiplicande est chargé en série, Habib Mehrez [Meh83] [GKTMN85] a proposé d'utiliser le temps de chargement du multiplicande de l'opération suivante pour laisser la retenue influencer complètement le résultat. Ces travaux ont été repris par Per Larsson Edefors et Christer Svensson [LS93] pour fournir des multiplieurs trés rapides pour des réseaux de neurones artificiels dans un système de traitement d'images vidéo [CS90].

Comme l'ont souligné ces scientifiques, la taille des deux opérandes peut trés bien être différente. Pour illustrer le déroulement dans le temps de ces différentes opérations, nous prenons le cas d'un produit d'un nombre de quatre chiffres, $d = d_3d_2d_1d_0$, par un nombre de trois chiffres, $a = a_2a_1a_0$, qui est représenté sur la figure 3.8. Notons $r_{f,j}$ la retenue engendrée dans l'addition des produits partiels par les chiffres produisant le terme correspondant à la puissance 2^j du résultat final qui s'applique donc au terme de puissance 2^{j+1}.

Le multiplieur résultant de leurs travaux charge en série le multiplicande, puis effectue la suite des produits partiels. Ceux-ci sont ajoutés au résultat de l'étape précédente dont a été extrait le chiffre le moins significatif pour être envoyé en sortie du multiplieur. Cela revient à décaler le résultat final d'un rang vers les poids faibles. Comme les retenues doivent être transmises

			d_3	d_2	d_1	d_0
				a_2	a_1	a_0
		$d_3 * a_0$	$d_2 * a_0$	$d_1 * a_0$	$d_0 * a_0$	
	$d_3 * a_1$	$d_2 * a_1$	$d_1 * a_1$	$d_0 * a_1$		
$d_3 * a_2$	$d_2 * a_2$	$d_1 * a_2$	$d_0 * a_2$			
	$d_3 * a_2$	$d_2 * a_2$	$d_1 * a_2$	$d_0 * a_2$	$d_0 * a_1$	$d_0 * a_0$
		$+d_3 * a_1$	$+d_2 * a_1$	$+d_1 * a_1$	$+d_1 * a_0$	
			$+d_3 * a_0$	$+d_2 * a_0$		
$r_{f,5}$	$+r_{f,4}$	$+r_{f,3}$	$+r_{f,2}$	$+r_{f,1}$	$+r_{f,0}$	
$p_{6,0}$	$p_{5,0}$	$p_{4,0}$	$p_{3,0}$	$p_{2,0}$	$p_{1,0}$	$p_{0,0}$

FIG. 3.8 – *produit de deux nombres dissymétrique.*

mathématiquement aux chiffres de poids immédiatement supérieur à ceux qui l'ont produite, elles sont maintenues au niveau du même étage de cette configuration. Notons $r_{i,j}$ la retenue produite lors de l'addition de l'étape i par les chiffres produisant le terme correspondant à la puissance 2^j du résultat partiel. La figure 3.9 représente le déroulement d'un produit partiel et le traitement associé de l'étape i, addition et répartition des données résultantes.

Lorsque le dernier produit partiel a été effectué, le chargement du multiplicande de l'opération suivante a lieu en même temps que le traitement de la partie haute du résultat qui vient d'être généré, diffusion des retenues jusqu'au chiffre de plus fort poids et extraction de tous les chiffres jusqu'à celui-ci. Le tableau 3.10 représente ce phénomène. Nous avons en fait simplifié le problème en ne faisant pas apparaître le traitement du signe, ce qui ne change rien à ce principe de propagation contrôlée de la retenue. Dans la réalité, le multiplicande n'est pas forcé à zéro avant cette partie du traitement pour pouvoir mettre en oeuvre une propriété de la notation en complément à deux qui permet d'inverser un nombre en l'ajoutant autant de fois qu'il comporte de chiffres aprés l'avoir multiplié par deux à chaque étape. Une multiplication par 2 peut être implantée avec un décalage vers les poids forts, ce qui est compatible avec les opérations sur le multiplicande réalisé lors du chargement de celui du produit suivant, et les chiffres qui ont un rang dépassant ceux prévus par la taille des nombres sont oubliés par l'initialisation des retenues et du registre de résultat partiel avant la multiplication suivante. Nous présentons ce phénomène pour un nombre de trois chiffres plus signe appartenant à $[\Leftrightarrow 1, 1[$ en nous rappelant que le chiffre

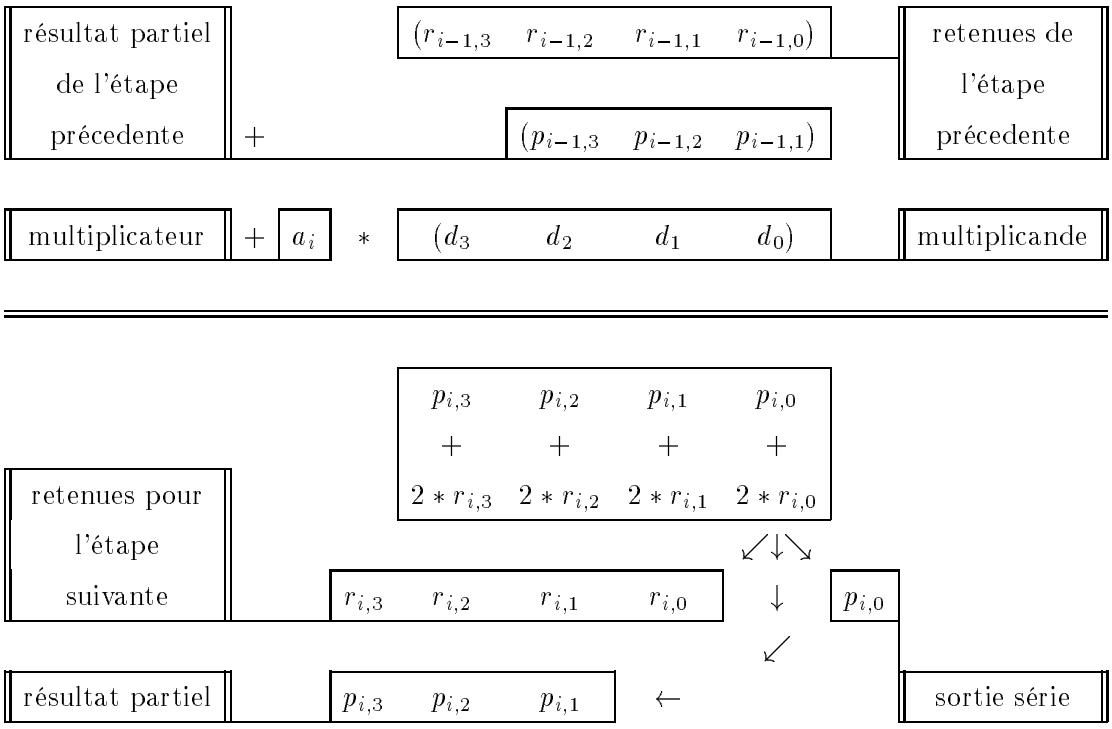

FIG. 3.9 – *produit partiel et traitements associés.*

de poids le plus fort est en fait le représentant du signe.

$$\begin{aligned}
d &= d_3\, d_2\, d_1\, d_0 \\
&= \Leftrightarrow d_3 + d_2 2^{-1} + d_1 2^{-2} + d_0 2^{-3} \\
\Longleftrightarrow \Leftrightarrow d &= \overline{d_3}\,\overline{d_2}\,\overline{d_1}\,\overline{d_0} + 2^{-3} \\
&= \Leftrightarrow(1 \Leftrightarrow d_3) + (1 \Leftrightarrow d_2)2^{-1} + (1 \Leftrightarrow d_1)2^{-2} + (1 \Leftrightarrow d_0)2^{-3} + 2^{-3} \\
&= \Leftrightarrow(1 \Leftrightarrow d_3) + (1 \Leftrightarrow d_2)2^{-1} + (1 \Leftrightarrow d_1)2^{-2} + 2(1 \Leftrightarrow d_0)2^{-3} + 2^{-3} d_0 \\
&= \Leftrightarrow(1 \Leftrightarrow d_3) + (1 \Leftrightarrow d_2)2^{-1} + (1 \Leftrightarrow d_1)2^{-2} + (1 \Leftrightarrow d_0)2^{-2} + 2^{-3} d_0 \\
&= \Leftrightarrow(1 \Leftrightarrow d_3) + (1 \Leftrightarrow d_2)2^{-1} + (1 \Leftrightarrow [d_0 + d_1])2^{-2} + 2^{-2} + 2^{-3} d_0
\end{aligned}$$

Nous retrouvons pour les puissances de $\Leftrightarrow 2$ le même type d'expression que précedemment pour les puissances de $\Leftrightarrow 3$, $(1 \Leftrightarrow [d_0 + d_1])2^{-2} + 2^{-2}$ au lieu de $(1 \Leftrightarrow d_0)2^{-3} + 2^{-3}$. Cette situation entraîne l'apparition d'un terme $(1 \Leftrightarrow d_1)2^{-1} + 2^{-2} d_1$ et le phénomène se renouvelle pour les puissances de $\Leftrightarrow 1$. Ce qui amène l'égalité suivante :

$$\begin{aligned}
\Leftrightarrow d &= \Leftrightarrow(1 \Leftrightarrow d_3) + (1 \Leftrightarrow [d_0 + d_1 + d_2]) + 2^{-1}(d_0 + d_1 + d_2) + 2^{-2}(d_0 + d_1) + 2^{-3} d_0 \\
&= d_3 \Leftrightarrow (d_0 + d_1 + d_2) + 2^{-1}(d_0 + d_1 + d_2) + 2^{-2}(d_0 + d_1) + 2^{-3} d_0 \\
&= \Leftrightarrow 2(d_0 + d_1 + d_2) + (d_0 + d_1 + d_2 + d_3) + 2^{-1}(d_0 + d_1 + d_2) + 2^{-2}(d_0 + d_1) + 2^{-3} d_0
\end{aligned}$$

Comme les nombres en complément à 2 sont tels qu'une puissance qui dépasse la fenêtre de valeurs considérées peut être considérées comme un terme nul, nous pouvons considérer tous ces

puissances de $\Leftrightarrow d$ comme quelconques. Ce qui nous donne :

$$\begin{aligned}
\Leftrightarrow d &= (d_0+d_1+d_2+d_3)+2^{-1}(d_0+d_1+d_2)+2^{-2}(d_0+d_1)+2^{-3}d_0 \\
&= 14d_3+6d_2+2d_1+(d_0+d_1+d_2+d_3)+2^{-1}(d_0+d_1+d_2)+2^{-2}(d_0+d_1)+2^{-3}d_0 \\
&= 14d_3+6d_2+2d_1+(d_3+2^{-1}d_2+2^{-2}d_1+2^{-3}d_0)+(d_2+2^{-1}d_1+2^{-2}d_0)+(d_1+2^{-1}d_0) \\
&= 14d_3+6d_2+2d_1+(d_3+2^{-1}d_2+2^{-2}d_1+2^{-3}d_0)+(d_2+2^{-1}d_1+2^{-2}d_0)+(d_1+2^{-1}d_0)
\end{aligned}$$

avant l'addition									aprés l'addition							
i	multipli-cande				retenue à propager				multipli-cande				produit partiel			sortie série
3	0	0	0	0	$r_{2,3}$	$r_{2,2}$	$r_{2,1}$	$r_{2,0}$	d_0	0	0	0	x	$p_{3,2}$	$p_{3,1}$	$p_{3,0}$
4	d_0	0	0	0	x	$r_{3,2}$	$r_{3,1}$	$r_{3,0}$	d_1	d_0	0	0	x	x	$p_{4,1}$	$p_{4,0}$
5	d_1	d_0	0	0	x	x	$r_{4,1}$	$r_{4,0}$	d_2	d_1	d_0	0	x	x	x	$p_{5,0}$
6	d_2	d_1	d_0	0	x	x	x	$r_{5,0}$	d_3	d_2	d_1	d_0	x	x	x	$p_{6,0}$

FIG. 3.10 – *diffusion de la retenue en fin de produit.*

La structure résultante représentée dans la figure 3.11 autorise une haute fréquence de fonctionnement et une excellente régularité pour une mise en oeuvre par une technique de *chemin de donnée*. Rappelons brièvement de ce dont il s'agit. Dans une architecture régulière, les opérateurs présentent une structure qui est la reproduction en grand nombre d'exemplaire d'un motif dont la description permet à un routeur de générer l'implantation correspondante, avant de le reproduire au nombre d'exemplaires voulu. C'est ce que fait à la main un concepteur traditionnel. Dans le cas d'une architecture étalée, il y a une double régularité, par le nombre d'étapes de calcul implantées et par le nombre de papillons implantés dans une barette consacrée à une étape de ce calcul. Une telle structure utilise trois signaux de commande qui dérivent d'un signal d'horloge général, deux pour le chargement des chiffres et un pour le pilotage des additionneurs.

I.3.5 Influence des notations arithmétiques sur une architecture.

I.3.5.1 Performances théoriques temporelles.

NOUS SOUHAITIONS COMPARER DEUX ARCHITECTURES, l'une à base d'opérateurs redondants et l'autre utilisant des multiplicateurs à notation en complément à deux trés rapides, donc d'un type similaire à ceux présentés dans le paragraphe 3.4. Les hypothèses de départ étaient une taille fixe, sans virgule flottante ou glissante. Nous nous étions intéressé au retard induit par les différents opérateurs pour évaluer les performances de telles solutions concurrentes. En effet ce retard, imposé par un circuit pour fournir ses résultats et plus couramment appelé *latence*, se traduit dans le temps de calcul d'une transformée de Fourier. Avec des opérateurs sériels de latence η dont les opérandes ont une taille B et qui opèrent sur M étapes successives, le temps de calcul rapporté à la période de fonctionnement est égal à $B + M \times \eta$. Précisons comment nous

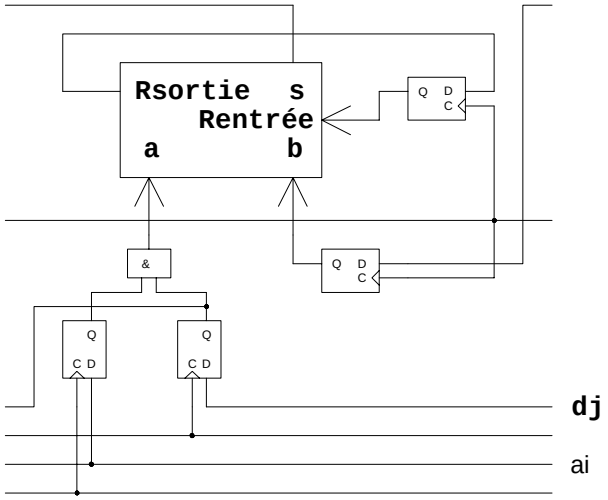

FIG. 3.11 – *tranche de multiplieur à retenue à avance progressive de la retenue.*

arrivons à cette formule. Lorsque la première donnée est introduite au temps zéro dans le premier étage de calcul, il faut attendre η coups d'horloge avant que le premier chiffre du résultat de cet étage ne soit produit, sachant que les chiffres suivants suivent immédiatement derrière au rythme d'un par coup d'horloge. Ce premier chiffre du résultat du premier étage alimente le deuxième étage de calcul qui produit le premier chiffre de son résultat avec de nouveau un retard de η, soit $2 \times \eta$ coups d'horloge aprés le temps zéro. Chaque étape ajoute η au retard général, soit $M \times \eta$ en tout. Les B chiffres du résultat nécessitent chacun un coup d'horloge supplémentaire, d'où la formule $B + M \times \eta$.

Résumons donc les latences à prendre en compte dans notre étude :

- un multiplieur en notation en complément à deux tel que celui que nous venons d'étudier traitant des nombres de B chiffres nécessite B impulsions pour recevoir les chiffres et opérer sa première phase de traitement, puis B autres impulsions pour laisser terminer la propagation d'éventuelles retenues et de fournir tous les chiffres du résultat. Seuls ceux fournis d'ailleurs dans la deuxième phase sont conservés dans la pratique, car les plus significatifs. En effet, les nombres conservent une taille constante au cours des calculs successifs, donc les chiffres les moins significatifs sont oubliés à chaque étape. La latence d'un tel multiplieur est donc égale à B.

- Un multiplieur et un additionneur sériels redondants tels que ceux que nous présentons au chapître 4 ont une latence respectivement égale à trois et deux. Dans le cas du multiplieur, les chiffres de poids faible ne sont jamais produits, car inutiles.

Les considérations suivantes sur les performances temporelles concernent des architectures complètement étalées ou dont les papillons de la macrobase de la transformée de Fourier ont une architecture interne complètement étalée pour minimaliser les temps de calculs.

Nous avons représenté dans la figure 3.12 le retard créé tout au long du calcul par des opérateurs sériels en complément à deux et redondants pour les deux types de base, 2 et 4, utilisées pour décomposer une T.F.R. au niveau du calcul arithmétique comme nous l'avons expliqué au paragraphe 2.3.3. Nous avons choisi une taille d'opérandes égale à 32 chiffres pour se placer dans une optique d'applications scientifiques. Nous avons tenu compte du fait que

dans les deux premières étapes, ou dernières selon le type de décomposition, tous les coefficients exponentiels sont simplifiables, car égaux à ± 1 ou $\pm j$. Il n'y a donc pas de multiplieurs et les retards associés.

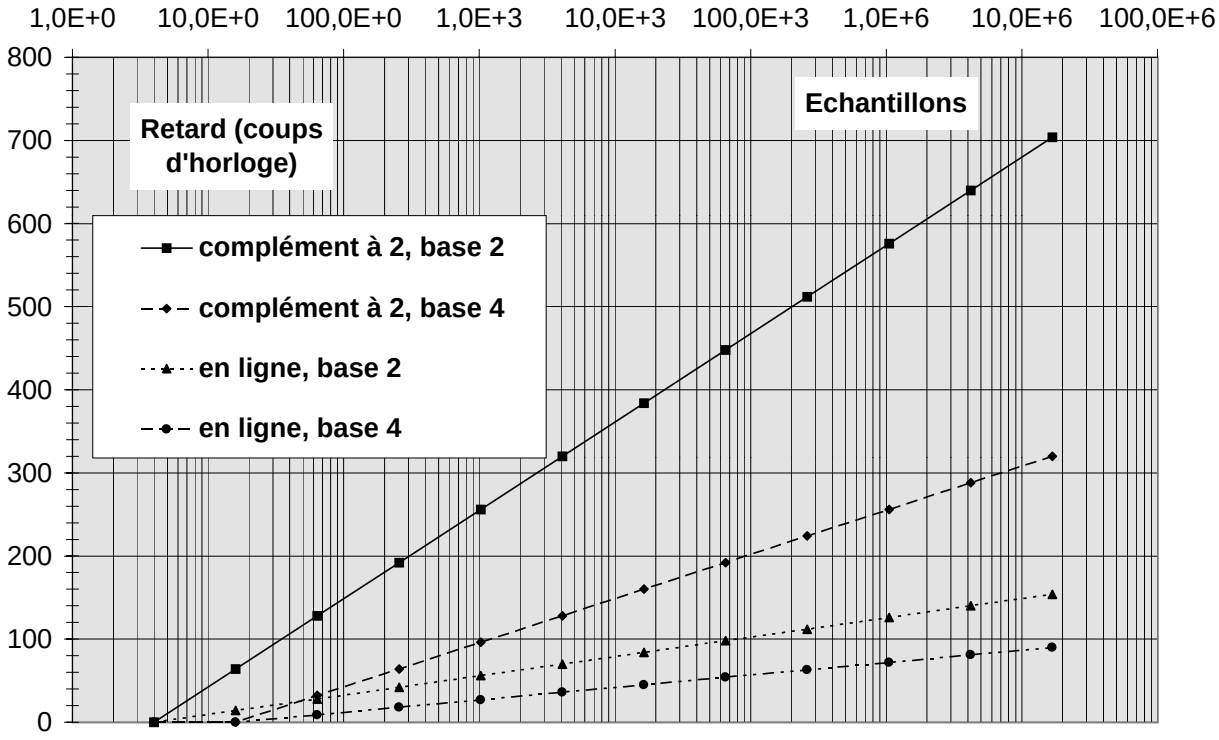

FIG. 3.12 – *retard induit par les opérateurs en fonction du nombre d'échantillons, selon la notation des chiffres et la base de décomposition.*

Nous pouvons remarquer qu'une décomposition selon une base 4 est nettement plus intéressante qu'une base 2 pour les deux notations arithmétiques. Ceci est la conséquence du nombre d'étapes du calcul, notons le M. Rappelons que si la base de décomposition de la T.F.R. est p et le nombre d'échantillons N, nous avons $M = \log_p N$.

Nous pouvons aussi noter le caractère trés avantageux d'une notation redondante qui induit un retard jusqu'à quatre fois plus faible que celui d'une base 4 en complément à deux, la meilleure de ses concurrentes.

Ce critère trés avantageux de comparaison doit toutefois être relativisé, car il correspond à une architecture étalée complètement implantée dont la réalisation est trés hypothétique à cause de la surface nécessaire. A moins de pipeliner tous les étages de calcul apparaissant dans une telle architecture, ce qui aurait un coût trés important en registres de stockage, il faut relativiser ces retards par rapport à la durée incompressible dans un calcul fait avec des opérateurs sériels. Il correspond au transfert des chiffres au cours des étapes successives, accompagné du traitement arithmétique afférent. Ce qui nous donne les courbes de la figure 3.13. Nous avons choisi le calcul d'une T.F.R. $3D$ de 256^3 échantillons avec différentes valeurs de la macrobase pour les deux bases arithmétiques et les deux notations possibles, sachant qu'un seul papillon est implanté. Le temps de calcul est exprimé en coups d'horloge et l'utilisation de plusieurs papillons diviserait les valeurs représentées par ce nombre. Bien que les problèmes de communication de données et les temps qui leur sont liés ne soient pas pris en compte nous pouvons toutefois noter le fort gain en terme de temps de calcul qu'il est possible d'espérer avec une solution cablée même relativement

modeste. Une solution avec une macrobase de 2 correspond à une solution logicielle ou à base de D.S.P. pour un seul papillon. Une solution avec une macrobase de 262144 correspond à une solution cablée certes déja imposante, car nécessitant plusieurs puces, mais le gain est voisin d'un million entre les solutions extrêmes. Les performances pour cette dernière macrobase et les diverses solutions de base et de notation arithmétique sont en fait l'image de celles d'une solution totalement implantée qui apparaissent dans la figure 3.12 pour un nombre d'échantillons de 262144.

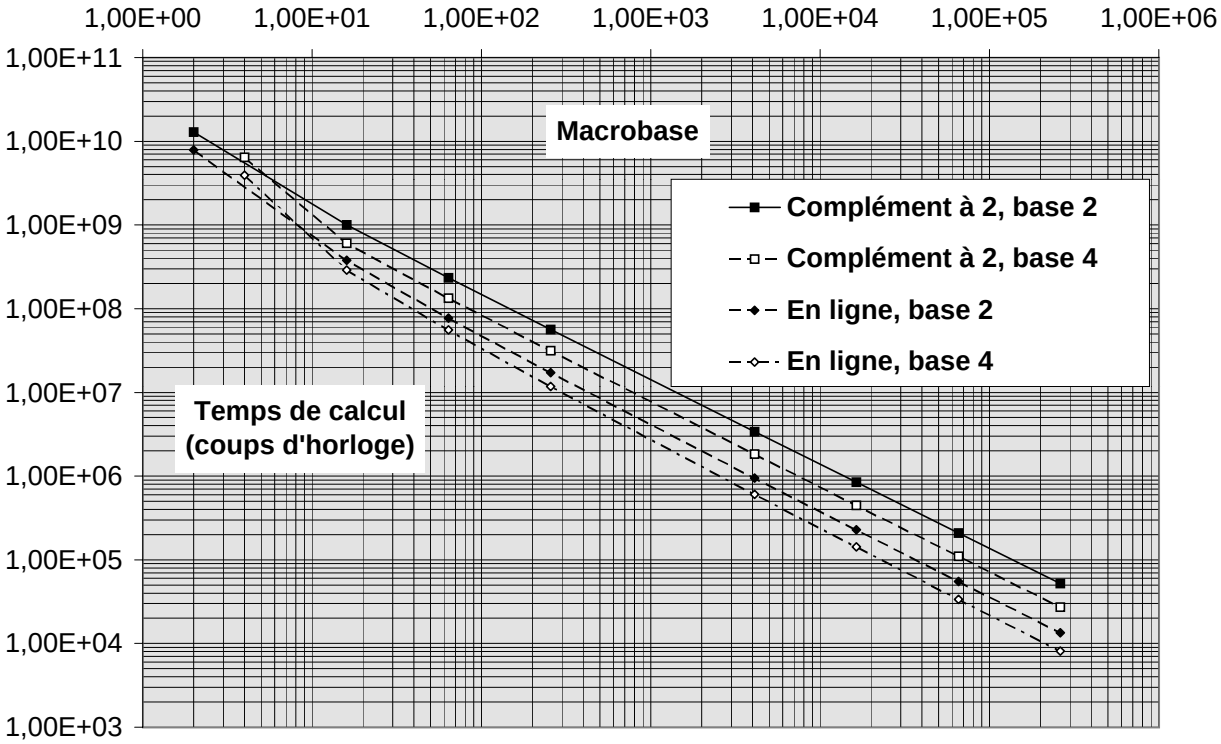

FIG. 3.13 – *temps de calcul pour* 256^3 *échantillons selon la macrobase choisie.*

Pour avoir une idée de la dynamique des temps de calcul selon la macrobase et le nombre total d'échantillons, nous avons représenté sur la figure 3.14, le temps de calcul d'une T.F.R. en fonction de son nombre d'échantillons pour diverses macrobases. Lue autrement, cette courbe permet de déterminer pour un temps de calcul donné, le nombre maximal d'échantillons acceptables pour une architecture, donc une macrobase, donnée. Ce qui est souvent le choix laissé à un utilisateur ! Un architecte de machines peut déterminer la macrobase nécessaire pour un temps de calcul et un nombre d'échantillons donnés.

I.3.5.2 Performances d'opérateurs arithmétiques conçus, fabriqués et testés.

Au vu de ces courbes, nous pouvons constater qu'une notation en complément à deux procure des performances moindres que celles d'une notation redondante. Même si la différence devient nettement plus faible dans une implantation réelle, utilisant un papillon d'une macrobase, que dans une implantation idéale, c'est à dire complètement étalée. Encore ne faut-il pas oublier la fréquence de travail des différents opérateurs. Nous avions à notre disposition les résultats de Yustina KUSUMAPUTRI [Kus93] et de Ali SKAF [Ska95], deux thésards ayant préparé leur thèse au sein de la même équipe que la notre, en ce qui concerne les opérateurs redondants. Ce

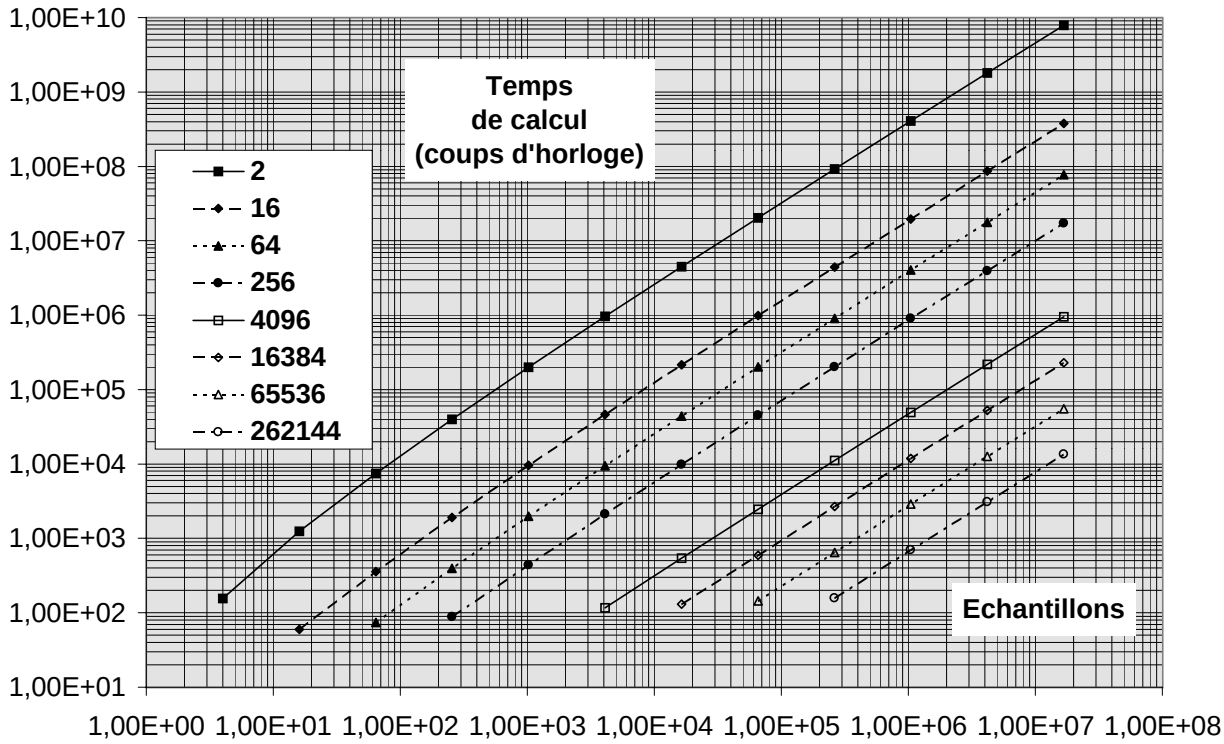

FIG. 3.14 – *temps de calcul pour diverses macrobases en fonction du nombre d'échantillons.*

qui permet une comparaison avec ceux obtenus par Per LARSSON et Christer SVENSSON qui ont utilisé une technologie plus récente que Habib Mehrez, ayant travaillé sur ce sujet beaucoup plus tard . Le tableau 3.1 récapitule les principaux résultats expérimentaux de ces différents travaux.

Les opérateurs peuvent présenter une structure en couche qui peut être mise à profit pour limiter le segment de propagation des signaux en cassant le parcours total avec des barrières temporelles et ainsi augmenter la fréquence de cadencement de ces opérateurs. Cette structure est dîte *pipeline*. La structure de pipeline la plus courante est celle qui utilise le fait que des données parcourent les opérateurs de l'entrée vers la sortie, c'est ce que nous appelons *pipeline longitutinale*. Il se trouve que certaines données traversent les opérateurs dans un sens perpendiculaire. Pour un opérateur arithmétique, il s'agit de signaux transmis d'un chiffre de poids plus faible vers un chiffre de poids plus fort ou inversement, c'est notamment le cas des retenues. Nous appelons l'exploitation de ce pnénomène *pipeline transversale*. Les opérateurs du type de ceux développés par Habib Mehrez, puis Per LARSSON et Christer SVENSSON bloquent la propagation des retenues au plus proche voisin de l'étage qui les a créées à chaque impulsion d'horloge. Leurs déplacements ultérieurs sont couplés avec l'avancement longitudinal que subissent par ailleurs les données qui sont traitées dans cette opération. Ils présentent donc une structure de pipeline transversale. Au contraire, les circuits redondants développés au laboratoire TIMA, bien que supportant des déplacements transversaux de retenues plus courts, doivent attendre la fin de ce déplacement pour que les données d'une tranche du pipeline longitudinal soient transmises à la tranche suivant du pipeline longitudinal qui pourrait être introduit. Pour résumer ces notions, nous représentons sur la figure 3.15 un opérateur arithmétique présentant les deux types de pipelines. Les additionneurs sont marqués d'un signe + et les registres d'une

Auteurs	règle de dessin (μm)	fréquence de travail (MHz)	surface en μm^2	
			multiplieur	additionneur
Yustina KUSUMAPUTRI	2	30	3469140	52488
Ali SKAF	1,2	27	1920000	13700
Habib Mehrez repris par Per LARSSON et Christer SVENSSON	1,2	100	253440	5280

TAB. 3.1 – *résultats des différentes implantations étudiées lors des comparaisons.*

lettre r. Le pipeline transversal opère sur la propagation des retenues du bloc interne complexe, le pipeline longitudinal sur les trois étages d'opérateurs apparaissant entre l'entrée et la sortie.

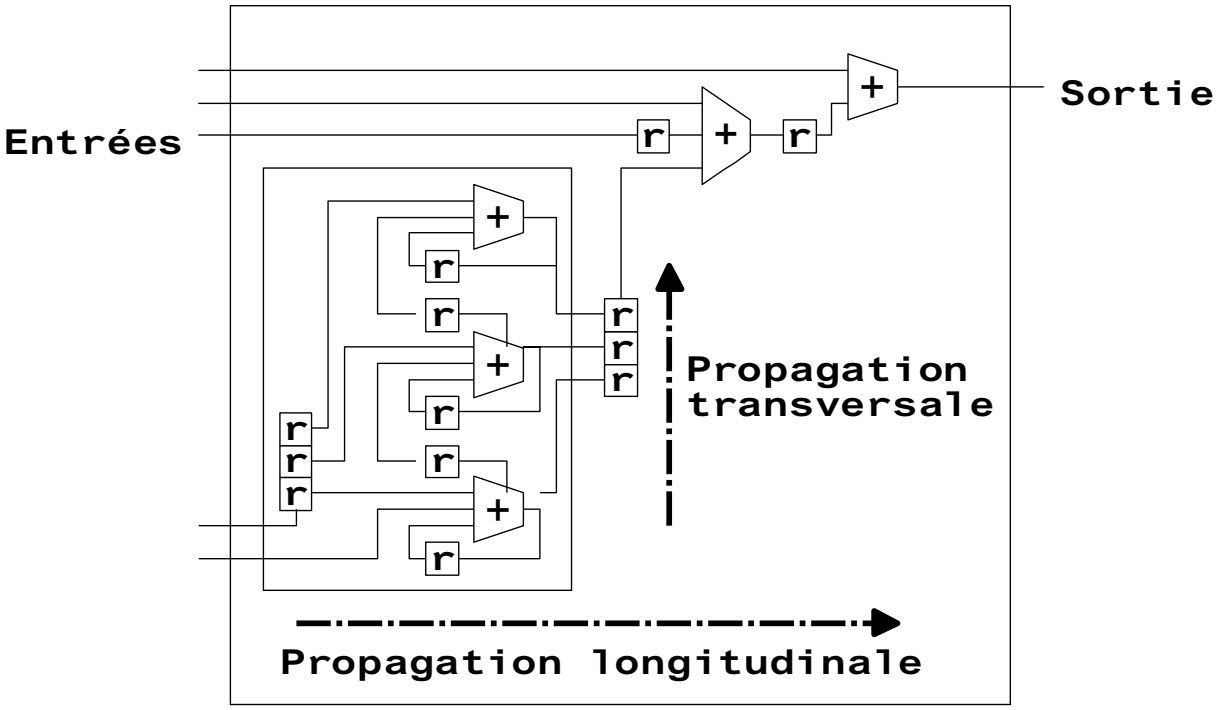

FIG. 3.15 – *exemple d'un opérateur arithmétique présentant des structures de pipelines transversales et longitudinales.*

Nous pouvons rapidement remarquer que :

– seule la surface de l'implantation effectuée par Ali SKAF a bénéficié du changement de technologie comparativement à celle de Yustina KUSUMAPUTRI. Il est vrai qu'il étudiait des fonctions complexes, du type polynômieur, et non l'implantation de multiplieurs et diviseurs comme elle. Ce qui justifie de faire porter ses efforts sur la surface utilisée au détriment de la fréquence de travail.

- Les fréquences des opérateurs redondants sont trois fois inférieures à celles des opérateurs en complément à deux étudiés ici. Cela est dû en partie à l'architecture pipeline de ces derniers, dans le sens de la largeur, alors que que les opérateurs redondants présentés ici n'ont aucune structure pipeline dans le sens de la profondeur comme le permettrait leur architecture.

- La surface d'un opérateur redondant devrait, à fréquence égale pour des architectures pipelinées, à priori être entre deux et quatre fois supérieure à celle d'un opérateur en complément à deux selon les architectures employées et donc leurs coefficients de proportionnalité, utilisant deux fois plus de variables binaires. Nous avons ici un rapport de presque huit, alors que les fréquences ne sont pas égales et que les opérateurs redondants cités ici ne mettent pas en oeuvre de structure pipeline, donc ne consomme pas de surface pour ce genre de fonction. Cela est certainement la conséquence de la différence de méthodes et de moyens en jeu comme nous le faisons remarquer au paragraphe 4.4.2 où nous comparons des résultats pour des opérateurs redondants dessinés par des voies différentes.

Ces résultats expérimentaux montrent la nécessité de développer des opérateurs redondants trés performants, tant en terme de vitesse que de surface d'implantation. En absence d'opérateurs ayant des caractéristiques comparables à leurs concurrents, la limitation de la propagation des retenues ne suffit pas pour rendre attractif une telle solution. Le faible nombre de produits en circulation dans le monde utilisant la notation redondante, donc théoriquement les meilleurs, n'est pas là pour nous démentir.

I.3.5.3 Coûts en surface d'implantation.

Commençons par examiner le cas d'une base 4, la plus favorable au niveau du temps de calcul par son nombre d'étapes. La figure 3.16 représente la surface nécessaire pour une implantation étalée en fonction du nombre d'échantillons pour les bases 2 et 4. Cette dernière correspond aux points indiqués par un symbole triangulaire. Nous avons limité le nombre d'échantillons à 256 qui correspond à une T.F.R. tridimentionnelle d'une taille déja trés raisonnable, 16 millions d'échantillons. Nous pouvons constater que seuls des opérateurs en complément à deux peuvent espérer tenir dans une surface autorisée par les contraintes technologiques pour 256 échantillons avec les technologies aujourd'hui disponibles, disons $0,5\mu m$ pour fixer un ordre de grandeur. Pour les opérateurs redondants, le nombre maximal n'atteint même pas 64.

Ceci nous amène à nous poser la question de la base de décomposition. La base 4 permet en divisant le nombre d'étapes par 2 de diminuer le temps de calcul, mais au prix d'une complexité des papillons supérieures. Rappelons que le nombre de papillons intégrant des multiplieurs n'est pas une fonction proportionnelle au nombre d'échantillons, si nous considérons le cas d'une architecture étalée qui permet de simplifier des papillons en supprimant les multiplications par ± 1 ou $\pm j$. Dans la figure 3.16 les surfaces nécessaires avec la base 2 pour des implantations avec une même technologie de $1,2\mu m$ est représentée par des courbes dont les points caractéristiques sont représentés par des carrés. Nous pouvons remarquer que :

- des nombres d'échantillons égaux à 4 ou 16 avec une base 4 ne permettent pas de calculer une T.F.R. plus grande, car les papillons concernés ne comportent pas de multiplieurs.

- La base 2 n'amène pas de différence suffisamment significative en ce qui concerne la réalisation sur une surface compatible avec les standards qui nous étaient accessibles, 190mm^2. Ce qui aurait pu être envisageable à priori avec les nombres d'échantillons intermédiaires qui n'apparaissent pas avec une base 4.

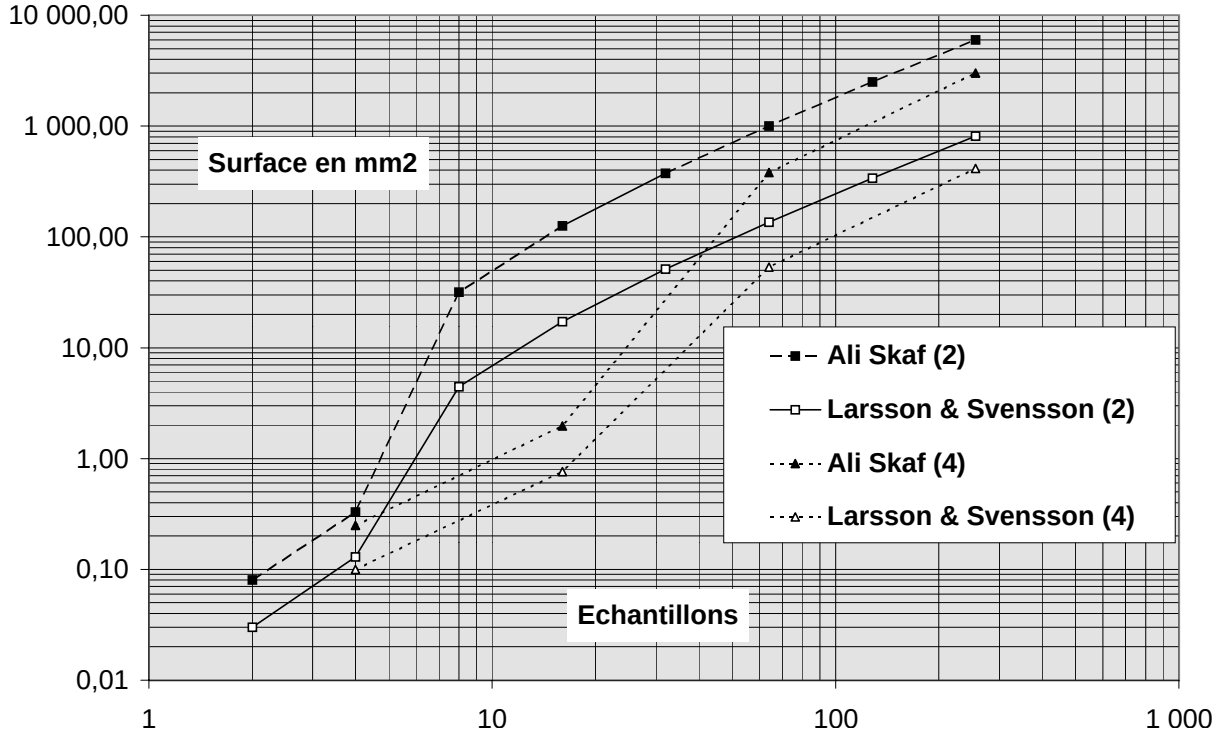

FIG. 3.16 – *estimation simplifiée de la surface nécessaire pour une T.F.R. étalée en fonction du nombre d'échantillons et selon la base.*

Toutefois il faudrait approfondir plus en détail ce problème de la surface à travers au moins les points suivants :

- les connexions sont plus complexes en base 4, tant à l'intérieur des papillons que pour l'alimentation de ces derniers en données à traiter, mais elles n'ont pas été prises en compte ici.
- Un papillon réalisant des calculs sur des opérandes communs, les règles de simplifications pour la conception d'un circuit permettent de réduire la surface d'implantation grâce à la diminution du nombre de registres, ceux en double qui apparaissent à des endroits différents du circuit, mais ne mémorisant des facteurs indentiques.
- Il existe un moyen de réduire le nombre de multiplieurs au détriment du nombre d'étapes à l'intérieur d'un papillon, donc au détriment du retard introduit et du temps de calcul.

Reprenons les équations à implanter dans un papillon telles que nous les avons données au paragraphe 2.6.2. Nous avons à calculer :

$$\begin{aligned}
a + (c.e \Leftrightarrow d.f) &= a + c.e \Leftrightarrow d.f + c.f \Leftrightarrow c.f \\
&= a + c.(e + f) \Leftrightarrow f.(c + d) \\
b + (c.f + d.e) &= b + c.f + d.e + d.f \Leftrightarrow d.f \\
&= b + f.(c + d) + d.(e \Leftrightarrow f) \\
a \Leftrightarrow (c.e \Leftrightarrow d.f) &= a \Leftrightarrow c.e + d.f + c.f \Leftrightarrow c.f \\
&= a \Leftrightarrow c.(e + f) + f.(c + d)
\end{aligned}$$

$$\begin{aligned} b \Leftrightarrow (c.f + d.e) &= b \Leftrightarrow c.f \Leftrightarrow d.e + d.f \Leftrightarrow d.f \\ &= b \Leftrightarrow f.(c + d) \Leftrightarrow d.(e \Leftrightarrow f) \end{aligned}$$

Si nous calculons $c.(e + f)$, $d.(e \Leftrightarrow f)$ et $f.(c + d)$, nous n'employons que trois multiplieurs au lieu de quatre, mais avec une étape d'addition en plus. La surface de ces papillons est donc diminué d'un tiers. Le retard est génant dans une notation redondante, il est augmenté d'un tiers, mais pas en complément à deux. Ce qui ne fait qu'accentuer l'avantage en faveur de ce dernier.

Nous en revenons au même point : sans opérateur redondant de performance en vitesse comparable à celle des opérateurs en complément à deux, l'avantage en temps de calcul du redondant dû ici au fait que ne sont calculés que les nombres de poids fort réellement utiles ne peut apparaître déterminant.

Aprés avoir considéré le problème d'une architectures étalée, examinons les caractéristiques en terme de surface pour les deux notations dans le cas d'une architecture repliée. La figure 3.17 nous les montre pour les trois cas d'opérateurs précités. Contrairement à une architecture étalée, la base 4 est plus gourmande en terme de surface. Ce qui s'explique facilement, puisque le nombre de papillons n'est plus réduit par le nombre des étapes. Par contre la surface nécessaire pour un nombre d'échantillons donné est plus faible et rend plus réaliste une implantation. Même avec une technologie $2\mu m$ et une arithmétique redondante, il était possible d'obtenir une solution cablée.

Le nombre réduit de papillons à priori implantable va toutefois dans un sens défavorable à la notation redondante, car ses avantages sont surtout intéressantes avec une architecture étalée. Une barette de papillons peut alimenter ceux de l'étape suivante alors que leurs calculs ne font que commencer, par le fait du calcul poids fort en tête. Ce qui est impossible avec une notation en complément à deux. Dans le cas de l'emploi d'une notation redondante, il vaut mieux prévoir une solution nécessitant plusieurs circuits pour profiter pleinement des avantages de celle-ci.

I.3.6 Perspectives.

TOUTES CES CONSIDÉRATIONS NE DOIVENT PAS FAIRE OUBLIER que dans les faits un concepteur peut détecter des simplifications telles que celles qui ont été mises en évidence dans le passé et qui sont utilisées avec profit dans le chapître 4. C'est pourquoi, compte tenu de ce que nous disons dans ce dernier chapître et dans les paragraphes ci-dessus, nous pouvons juger raisonnable la conception d'un circuit de 64 échantillons en redondant et de 256 en complément à deux. Ce qui impose dans le premier cas d'utiliser au minimum huit puces pour une T.F.R. tridimensionnelle de 16 millions de points, contre une seule en complément à deux. $256 = 4 * 64$, $256 = 2^{16}$ et $64 = 2^6$. Pour les puces de 64 échantillons, il faut deux barettes de 4 puces, sachant que dans la seconde barette les puces ne calculeront pas le nombre d'étapes maximal. Il est toutefois plus simple de prévoir une architecture utilisable plus ou moins partiellement que de concevoir deux puces différentes, question de coût de fabrication notamment.

Si nous supposons que tous les circuits fonctionnaient à $100MHz$, cela amène des temps de calcul respectifs de $585ms$ et $880ms$ pour une base 2 et $287ms$ et $377ms$ pour une base 4. Ces temps pourraient être abaissés par l'emploi de plusieurs puces en parallèle ou batterie de puces pour les solutions à notation redondante. Nous pouvons comparer ces résultats avec ceux du produit que la société Texas Memory Systems a mis sur le marché au printemps 1996. Il travaille sur des nombres flottants de 32 bits et assure le calcul d'une T.F.R. de 1 million de points en $110ms$. Ce qui, par extrapolation, donne un temps de calcul de $2,112s$. Nous pouvons

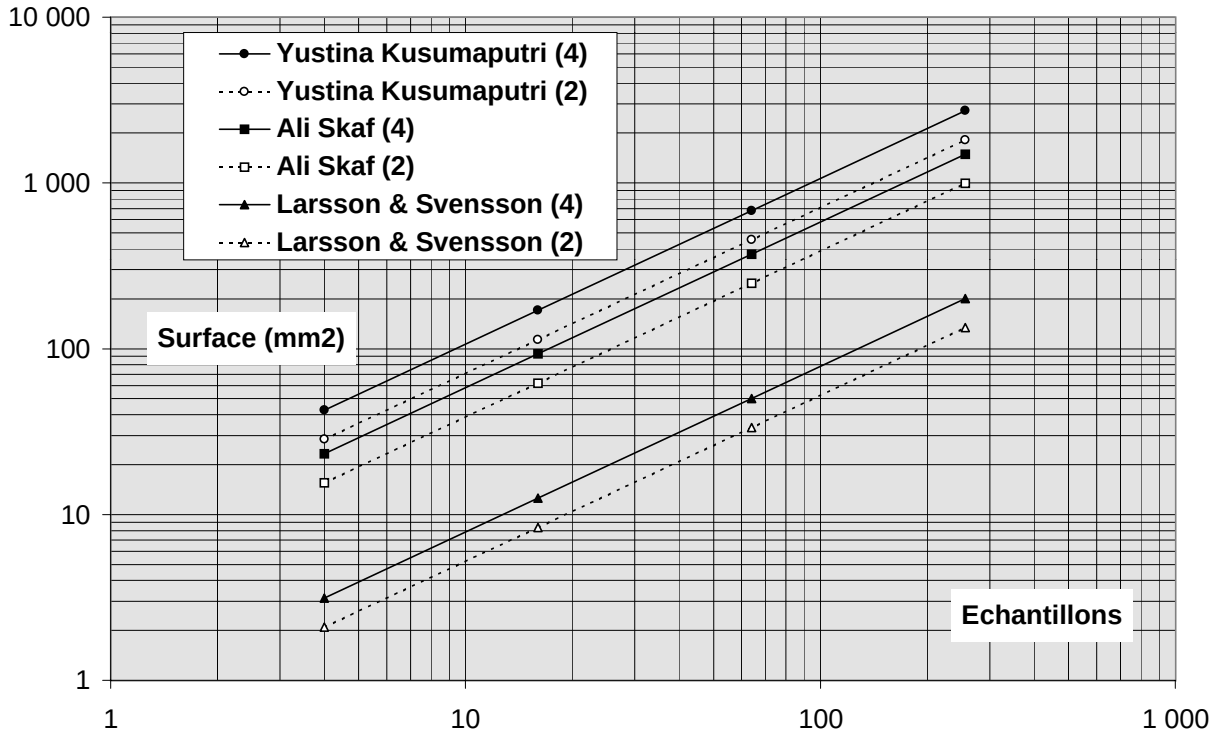

FIG. 3.17 – *surface d'une T.F.R. repliée en fonction du nombre d'échantillons selon la base.*

donc remarquer que le travail avec des nombres entiers permet des gains de vitesse intéressant en supposant que les calculs considérés puissent être faits avec des entiers, notamment avec une notation redondante et une base 4, avec les problèmes de surface déja évoqués.

Il semble donc que pour les T.F.R. à trés grand nombre d'échantillons les opérateurs sériels apportent des performances intéressantes qui sont améliorées grâce à la notation redondante, au prix d'une surface d'implantation importante. Cela demande le développement d'opérateurs redondants beaucoup plus compacts et rapides que ceux imaginables à partir de ceux disponibles. Il y a toutefois un domaine intéressant d'applications bien différent de celui étudié, celui des transformées à trés faible nombre d'échantillons, donc totalement implantables et où ces performances de vitesse sont réellement accessibles à l'utilisateur. Tout système ayant des contraintes temps réel ou de vitesse pourrait en profiter.

Chapitre 4

Implantations d'une T.F.R. avec des opérateurs en ligne

Sommaire

I.4.1 Introduction 53
I.4.2 Papillon en ligne 54
I.4.2.1 Considérations sur l'arithmétique. 54
I.4.2.2 Codage d'une notation redondante et propriétés. 58
I.4.2.3 Les composants de base d'une implantation redondante. 59
I.4.2.4 Architecture finalement choisie. 60
I.4.2.5 Commentaires. 62
I.4.3 Circuit réalisé 63
I.4.4 Perspectives de ce travail. 66
I.4.4.1 Généralités. 66
I.4.4.2 Fréquence de travail. 66
I.4.4.3 Architecture. 66
I.4.4.4 Conclusion. 69

I.4.1 Introduction

Le traitement du signal est devenu un domaine trés important et trés dynamique de recherche. Nous trouvons désormais dans le commerce des circuits dédiés au Traitement du Signal. C'est en particulier le cas de la T.F.R., à cause de ses nombreuses applications, tant dans le domaine scientifique qu'industriel. Malheureusement, les circuits cablés ne traitent qu'un faible nombre d'échantillons et surtout avec une dynamique réduite plus fréquente dans les applications industrielles que scientifiques. Il était donc tentant au cours d'une première expérience de conception de mettre l'accent sur ce dernier point ... et nous avons cédé à la tentation. Nous sommes partis aussi sur l'hypothèse que les échantillons étaient formés de nombres complexes. Pour obtenir une architecture la plus simple et la plus régulière possible, nous avions choisi une T.F.R. de base 2 à entrelacement temporel.

Nous détaillons d'abord dans le paragraphe 4.2 la structure du papillon qui apparait dans ce type d'algorithme. Ce qui nous amène à résumer les grands principes du fond de commerce de

l'équipe de recherche au sein de laquelle cette thèse a été préparée, la notation redondante et les opérations en ligne, c'est à dire sérielles et en commençant par les chiffres de plus fort poids. Les règles de fusion d'opérateurs en ligne nous permettent de proposer l'architecture optimisée d'un papillon en ligne. Avant de présenter dans le paragraphe 4.3 le circuit tel qu'il a été dessiné, nous évoquons le problème de la précision des calculs à travers la taille des opérandes et une façon de l'adapter aux désidératas des utilisateurs éventuels d'un tel circuit. Nous comparons ensuite notre circuit avec des travaux antérieurs. Ce qui nous amène dans le paragraphe 4.4 à établir les perspectives d'un tel travail.

I.4.2 Papillon en ligne

I.4.2.1 Considérations sur l'arithmétique.

UNE SÉRIE DE CALCULS étant faite sur un ensemble de données initiales, l'intégralité des chiffres produits par les calculs successifs ne peut être conservée. L'utilisation d'une numérotation classique comme la base binaire en complément à deux impose de calculer les chiffres les moins significatifs qui sont ensuite oubliés. Seules les retenues éventuelles produites à ce stade sont utilisées. Une notation redondante, telle que celle décrite par Avizienis en 1961 [Avi61], permet de commencer les calculs par les chiffres les plus significatifs et de traiter les suivants par poids décroissant. Cela permet soit de finir un calcul sans devoir évaluer les chiffres du résultat qui ne sont pas conservés, ou du moins la plus grande partie d'entre eux, soit de commencer le calcul de l'étape suivante avec comme seul délai la latence de l'étape considérée. D'abord réservée aux bases supérieures à 2, elle a ensuite été étendue à la base binaire.

Prenons le cas d'une base binaire redondante. Au lieu de pouvoir prendre deux valeurs, 0 et 1, un chiffre a l'une des trois valeurs suivantes : 0, 1 ou -1. Un nombre peut donc être mis sous plusieurs formes, prenons l'exemple du chiffre 3 : 11 (2+1, en décimal) ou 100-1 (4-1) par exemple. D'où le nom de redondant.

Les avantages de ces types de notation sont en fait basés sur une anticipation des retenues, ce qui a pour effet apparent de limiter dans une addition la propagation d'une retenue au plus proche voisin du chiffre qui a créé cette retenue. En effet avec une numérotation binaire en complément à deux, comme d'ailleurs avec la base décimale utilisée dans la vie courante depuis trés longtemps, la retenue créée par les chiffres de plus faible poids peut se propager, dans le pire des cas, jusqu'aux chiffres de plus fort poids. Outre les propriétés propres aux calculs itératifs cités précédemment, la limitation de la propagation d'une retenue au proche voisinage permet pour des additionneurs parallèles traitant des nombres redondants :

- des temps de réponse plus faibles,
- une vitesse indépendante de la taille des nombres traités.

Les opérateurs arithmétiques [M.D77] sont généralement construit avec, entre autres, des additionneurs parallèles. L'utilisation de la notation redondante permet donc la conception d'opérateurs plus rapides, sans pénalité pour la taille des opérandes. De quoi séduire des amateurs de calcul scientifique dont la particularité par rapport aux communs des mortels est un besoin d'une plus grande vitesse et d'une plus grande précision.

Nous prenons comme principe d'appeler *opérateur en ligne* un opérateur sériel traitant en premier les chiffres de poids fort. Considérons le problème de la multiplication en ligne [GK91] [SBGM92] [SG93] de deux nombres A et D codés sur B chiffres redondants.

Nous pouvons écrire :

$$\begin{cases} A &= \sum_{i=1}^{B} a_i * 2^{-i} \\ D &= \sum_{i=1}^{B} D_i * 2^{-i} \end{cases}$$

L'opérateur reçoit, par principe, les a_i et d_i en série selon les valeurs de i croissantes. Définissons trois grandeurs intermédiaires :

$$\begin{cases} A_g &= \sum_{i=1}^{g} a_i * 2^{-i} \\ D_g &= \sum_{i=1}^{g} d_i * 2^{-i} \\ P_g &= A_g * D_g * 2^g \end{cases}$$

Nous pouvons remarquer que :

$$\begin{cases} A_g &= A_{g-1} + a_g * 2^{-g} \\ D_g &= D_{g-1} + d_g * 2^{-g} \end{cases}$$

Examinons les conséquences de ces dernières relations :

$$\begin{aligned} P_g &= A_g * D_g * 2^g \\ &= (A_{g-1} + a_g * 2^{-g}) * (D_{g-1} + d_g * 2^{-g}) * 2^g \\ &= A_{g-1} * D_{g-1} * 2^g + A_{g-1} * d_g + D_{g-1} * a_g + a_g * d_g * 2^{-g} \\ &= A_{g-1} * D_{g-1} * 2^{g-1} * 2 + (A_{g-1} + a_g * 2^{-g}) * d_g + D_{g-1} * a_g \\ &= P_{g-1} * 2 + A_g * d_g + D_{g-1} * a_g \end{aligned}$$

Nous pouvons appliquer cette relation sur un exemple simple de trois chiffres pour chaque opérande. Nous avons donc $B = 3$. Pour le vérifier, écrivons l'opération comme si nous la réalisions à la main :

		d_1	d_2	d_3
		a_1	a_2	a_3
		$d_1 * a_3$	$d_2 * a_3$	$d_3 * a_3$
	$d_1 * a_2$	$d_2 * a_2$	$d_3 * a_2$	
$d_1 * a_1$	$d_2 * a_1$	$d_3 * a_1$		
$d_1 * a_1$	$d_2 * a_1 + d_1 * a_2$	$d_3 * a_1 + d_2 * a_2 + d_1 * a_3 d_3 * a_2 + d_2 * a_3$	$d_3 * a_3$	

L'évolution des coefficients au cours du temps a été résumé dans le tableau 4.1. Nous obtenons bien dans P_g le produit désiré lorsque $g = B$. Le multiplieur cablé fournit son résultat en série. Il s'agit des termes de P_g qui atteignent une puissance de deux suffisantes pour ne pas être modifié par la propagation des retenues crées par les nouveaux termes. Propagation limitée, rappelons-le, par le codage redondant.

Pour que le lecteur peu familier de ces techniques puisse mieux visualiser l'extension des différents coefficients au sein des différents développements traditionnels du produit nous les avons représenté sur la figure 4.1 4.2 4.3. Chaque coefficient est en minuscule, encadré et relié à son nom qui est en majuscule avec un encadrement doublé sur la verticale.

g	1	2	3
A_g	$a_1 * 2^{-1}$	$a_1 * 2^{-1} + a_2 * 2^{-2}$	$a_1 * 2^{-1} + a_2 * 2^{-2} + a_3 * 2^{-3}$
D_{g-1}	0	$d_1 * 2^{-1}$	$d_1 * 2^{-1} + d_2 * 2^{-2}$
P_{g-1}	0	$a_1 * d_1 * 2^{-1}$	$a_1 * d_1 + [a_1 * d_2 + a_2 * d_1] * 2^{-1}$ $+a_2 * d_2 * 2^{-2}$
P_g	$a_1 * d_1 * 2^{-1}$	$a_1 * d_1$ $+[a_1 * d_2 + a_2 * d_1] * 2^{-1}$ $+a_2 * d_2 * 2^{-2}$	$a_1 * d_1 * 2 + [a_1 * d_2 + a_2 * d_1]$ $+[a_1 * d_3 + a_2 * d_2 + a_3 * d_1] * 2^{-1}$ $+[a_2 * d_3 + a_3 * d_2] * 2^{-2} + a_3 * d_3 * 2^{-3}$

TAB. 4.1 – *Évolution des grandeurs intermédiaires d'un produit.*

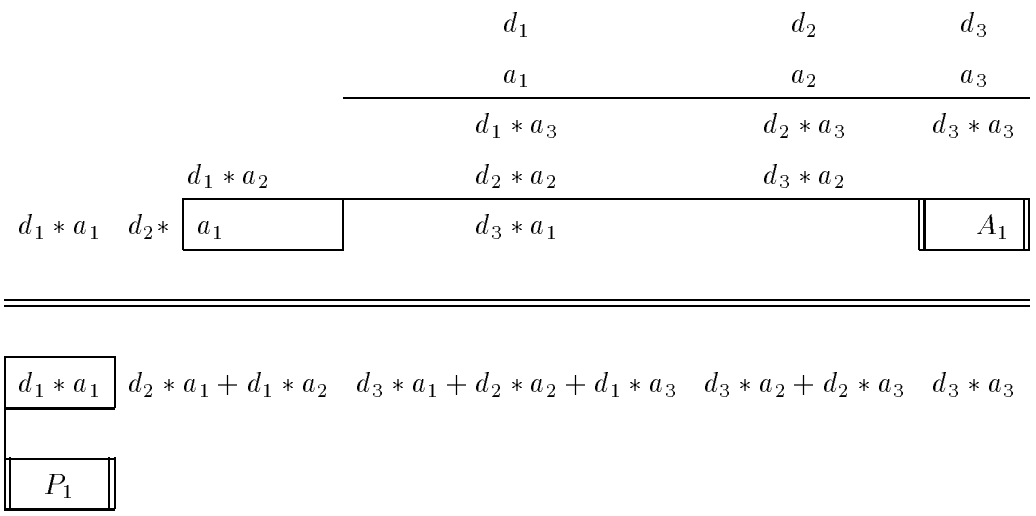

FIG. 4.1 – *extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés le premier des trois coups d'horloge qui amènent au résultat final.*

Un multiplieur redondant série-série est donc construit autour de trois registres (A_g, D_{g-1} et P_{g-1}), de deux barettes de multiplieurs élémentaires dont l'une des deux entrées est constituée par la sortie d'un registre de stockage des entrées (par exemple A_g) et l'autre par le chiffre provenant de l'autre entrée (d_g), d'un additionneur parallèle à trois entrées et d'une cellule finale de traitement. Cette dernière cellule sert, d'une part, à extraire les chiffres désormais indépendant du traitement à venir et, d'autre part, à limiter la taille des nombres redondants. En effet, les algorithmes de calcul commençant par les chiffres de poids fort sont en fait basés sur une anticipation de la retenue et peuvent faire apparaître des chiffres inutiles, appliquant le principe que $1 = 2 \Leftrightarrow 1$. Là où 1 est suffisant, 2 apparaît parfois ! Le schéma d'un tel multiplieur est représenté sur la figure 4.4.

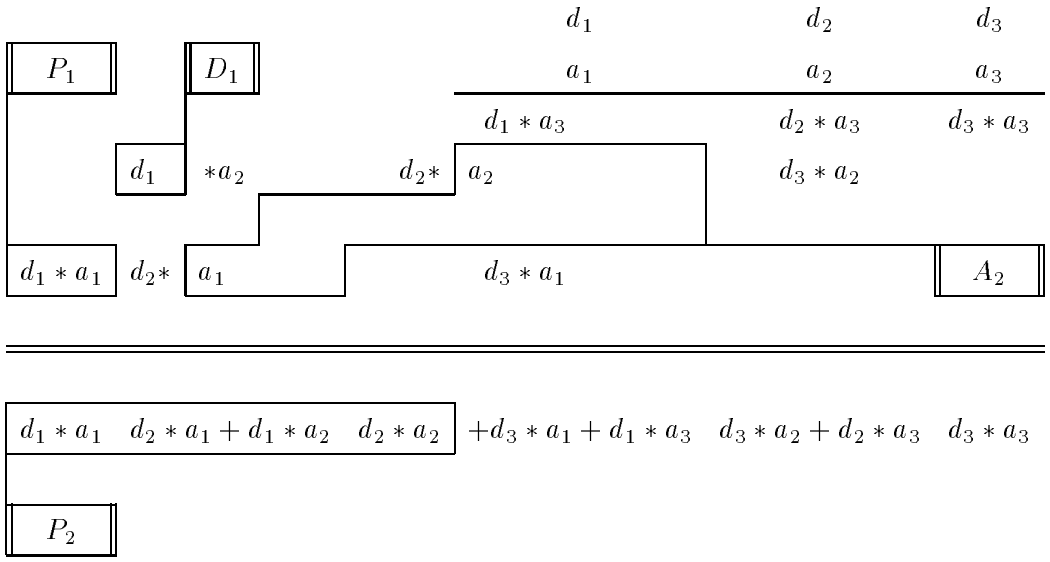

FIG. 4.2 – *extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés aprés le deuxième des trois coups d'horloge qui amènent au résultat final.*

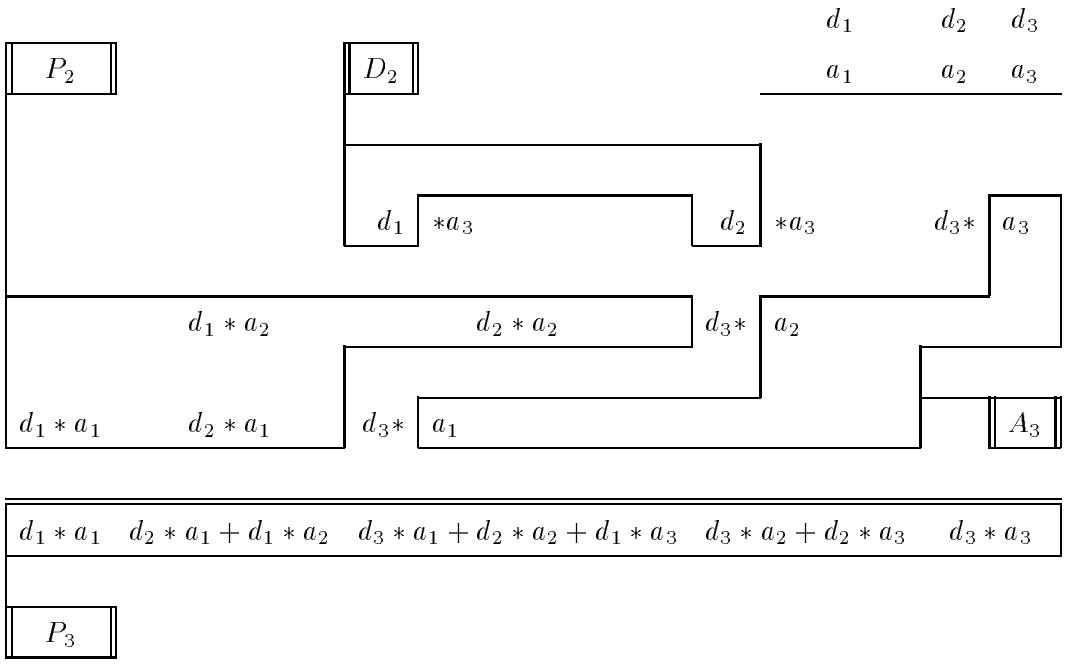

FIG. 4.3 – *extensions des différents coefficients A_i, D_i et P_i du produit à poids fort en tête aprés aprés le dernier des trois coups d'horloge qui amènent au résultat final.*

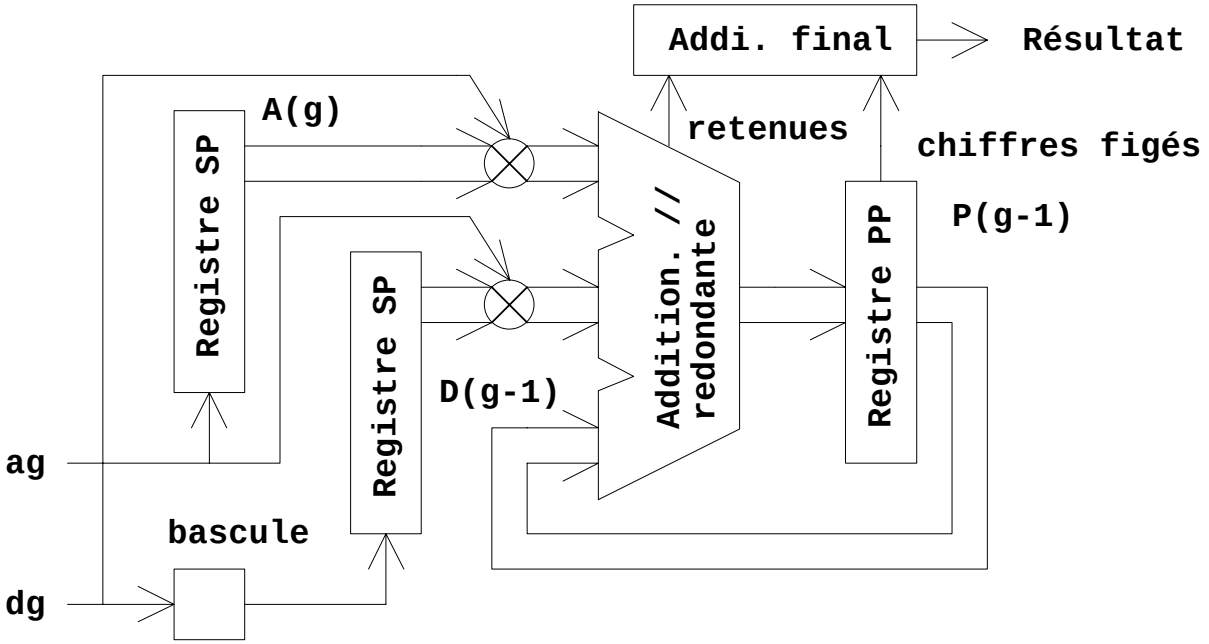

FIG. 4.4 – *schéma de principe d'un multiplieur en ligne à chiffre de poids fort en tête.*

chiffre redondant a_i	codage correspondant (a_{in}, a_{ip})
-1	10
0	00 ou 11
1	01

TAB. 4.2 – *codage d'une notation redondante.*

I.4.2.2 Codage d'une notation redondante et propriétés.

PRENONS L'EXEMPLE D'UN NOMBRE A. Avec une base binaire redondante $A = \sum_{i=1}^{B} a_i * 2^{-i}$ où chaque a_i prend ses valeurs dans l'ensemble $\{\Leftrightarrow 1, 0, 1\}$. Pour traduire ces chiffres mathématiques en une langue compréhensible par l'électronique numérique, nous utilisons une notation dîte de *chiffre binaire signé* ou à anticipation de retenue, *Borrow Save Notation* pour les puristes du franglais, formée de deux variables logiques a_{ip} et a_{in}. En assimilant chiffres binaires purs et variables logiques, nous avons la relation $a_i = a_{ip} \Leftrightarrow a_{in}$ résumée dans le tableau 4.2.

Dans une notation en complément à deux, l'inversion du signe est une opération relativement complexe qui découle de la définition de cette notation. Dans une notation redondante en général et binaire à anticipation de retenue en particulier, cette opération peut se réaliser trés simplement. Puisque tout chiffre redondant peut prendre une valeur positive ou négative, nous avons :

$$\Leftrightarrow A = \Leftrightarrow \sum_{i=1}^{B} a_i * 2^{-i} = \sum_{i=1}^{B} (\Leftrightarrow a_i) * 2^{-i}$$

Quant aux a_i, si nous jetons un coup d'oeil au tableau 4.3, nous pouvons remarquer que nous avons deux solutions pour les transformer en $\Leftrightarrow a_i$. Soit en permuttant les a_{in} et a_{ip}, soit en les

a_i	(a_{in}, a_{ip})	$\Leftrightarrow a_i$	$(\overline{a_{in}}, \overline{a_{ip}})$	(a_{ip}, a_{in})
-1	10	1	01	01
0	00	0	11	00
0	11	0	00	11
1	01	-1	10	10

TAB. 4.3 – *codage d'une négation en notation redondante.*

r		s		$t = r \times s$	
r_n	r_p	s_n	s_p	t_n	t_p
1	0	1	0	0	1
1	0	0	1	1	0
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1

TAB. 4.4 – *produit de deux chiffres binaires redondants.*

complémentant.

La conséquence la plus immédiate de cette propriété concerne la soustraction qui peut être effectuée avec un simple additionneur dont les entrées sont croisées au niveau de chaque chiffre redondant ou précédées d'un inverseur. Ceci restant vrai aussi bien pour les opérateurs sériels que parallèles.

I.4.2.3 Les composants de base d'une implantation redondante.

LA RÉALISATION D'OPÉRATEURS REDONDANTS exige un certain nombre de fonctions de base qu'il est nécessaire de réaliser pour pouvoir les utiliser comme éléments de bibliothèque et ainsi faciliter la conception des divers éléments souhaités. Nous avons besoin de concevoir :

- un multiplieur un vecteur par un chiffre qui n'est que la juxtaposition de muliplieurs élémentaires, un chiffre par un chiffre,
- un additionneur parallèle à trois entrées,
- un additionneur sériel à deux entrées pour fournir les résultats du papillon, il apparaît dans la figure 2.12 juste avant les sorties,
- un additionneur sériel à trois entrées pour la cellule finale du multiplieur.

Commençons par le multiplieur à un chiffre redondant binaire : $t = r \times s$. Une implantation possible est donnée par $t_n = \overline{r_n.s_n + r_p.s_p}$ et $t_p = \overline{r_n.s_p + r_p.s_n}$. Nous résumons dans le tableau 4.4 les cas de figure qui peuvent se rencontrer dans un tel produit.

La synthèse des additionneurs peut être faîte par la méthode des blocs *PPM*, pour Plus Plus Moins, et *MMP*, pour Moins Moins Plus. Ces briques de base sont assemblées selon le type de

l'additionneur réalisé : série ou parallèle, nombre d'entrées. La brique MMP est en fait équivalente à une brique PPM avec ses entrées complémentées. Considérons le bloc PPM. Il possède trois entrées, e_{1+}, e_{2+} et e_{3-} et deux sorties, s_{1+} et s_{2-}. Les équations logiques sont :

$$s_{1+} = \text{Majorité}\ (e_{1+}, e_{2+}, e_{3-})$$
$$s_{2-} = (e_{1+} \oplus e_{2+} \oplus e_{3-})$$

Le symbole et le schéma d'implantation utilisés sont donnés dans la figure 4.5. Rappelons qu'en *CMOS* un circuit est constitué de deux blocs de transistors complémentaires situés l'un entre l'alimentation et la sortie, l'autre entre la sortie et la masse. Selon les signaux d'entrée qui leurs sont appliqués, l'un force la sortie à zéro ou au plus de l'alimentation. Chaque signal d'entrée apparaît sur les deux blocs et agit de façon inverse dans l'un par rapport à l'autre, forçant un transistor au blocage et le transistor conjugué à la saturation.

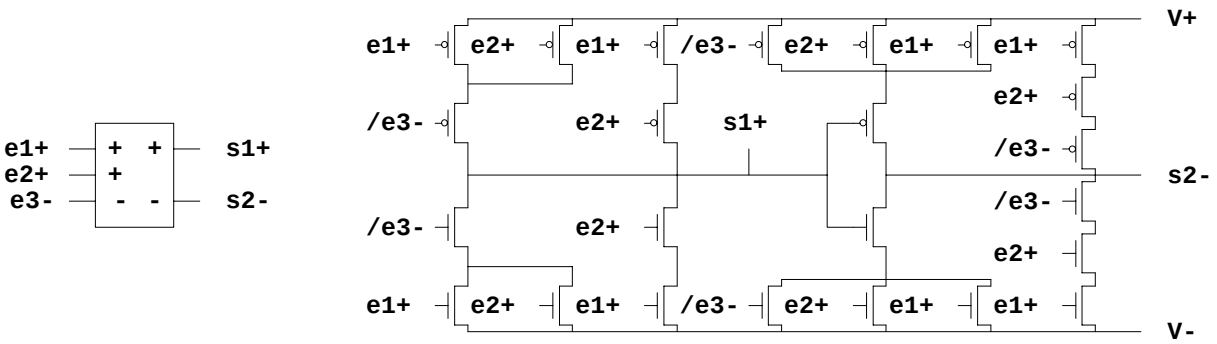

FIG. 4.5 – *bloc PPM pour la synthèse des additionneurs à anticipation de retenue.*

L'association des briques de base PPM et MMP nous a permis de concevoir les différents additionneurs qui s'étaient révélés nécessaires. Les figures 4.6, 4.7 et 4.8 représentent respectivement un additionneur en ligne pour deux chiffres, un additionneur en ligne pour trois chiffres et une tranche d'additionneur parallèle pour trois chiffres.

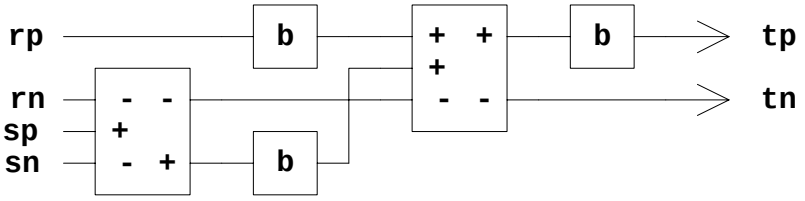

FIG. 4.6 – *additionneur série à anticipation de retenue pour deux chiffres redondants.*

I.4.2.4 Architecture finalement choisie.

DES RÈGLES DE SIMPLIFICATION D'UN CIRCUIT ont été établies dans le passé pour les cas où il est possible de mettre en commun certaines fonctions élémentaires entre différents opérateurs arithmétiques. Elles peuvent être mises en oeuvre pour concevoir des multiplieurs de nombres complexes [OVS94]. Rappelons les :

– Un des termes de chaque produit est connu avant le début de chaque opération puisqu'il ne dépend que du rang de l'étape et des indices des coefficients traités par le papillon,

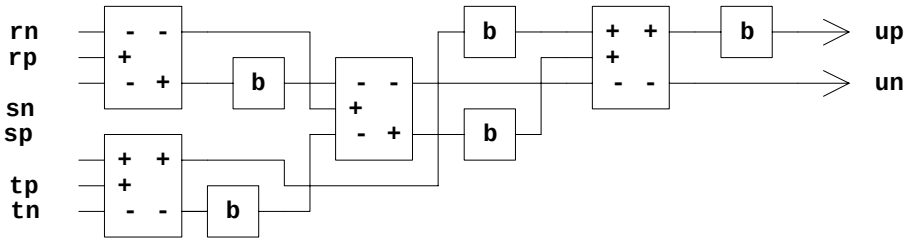

FIG. 4.7 – *additionneur série à anticipation de retenue pour trois chiffres redondants.*

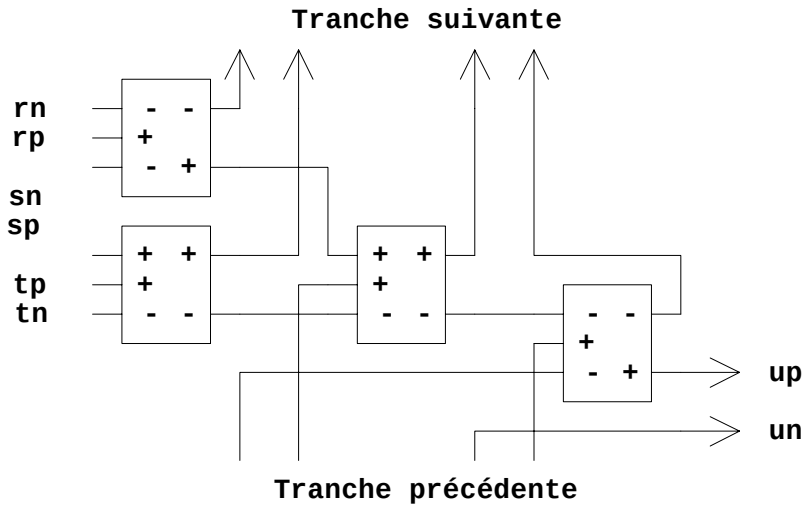

FIG. 4.8 – *tranche pour additionneur parallèle à anticipation de retenue pour trois chiffres redondants.*

c'est le terme exponentiel, se reporter à la figure 2.12. Nous pouvons donc le charger avant d'exécuter le produit et utiliser un multiplieur parallèle-série au lieu de série-série. Cela nous amène d'intéressantes simplifications. Si le nombre de registres est identique, l'un des multiplieurs d'un vecteur par un chiffre disparait.

– les termes réel et imaginaire de w^n sont communs à deux multiplieurs, les registres qui les mémorisent peuvent donc être mis en commun, d'où un gain en surface [BGW93].

– Le résultat des multiplieurs sont additionnés deux à deux, or la structure du multiplieur qui exploite ce registre de récursion est un additionneur. Il est donc possible de n'utiliser qu'un seul registre de récursion et de fondre les deux multiplieurs au niveau de l'additionneur final à travers un additionneur à trois entrées. Cela a comme conséquence heureuse que l'additionneur situé en sortie du papillon n'a plus que deux entrées.

Il faut noter que les données $f(n)$ arrivant en même temps que les $f(n + N/2)$, un registre tampon simule le retard nécessaire pour compenser le temps de traitement des multiplicateurs (latence de 3). Ce qui nous amène au schéma du papillon représenté dans la figure 4.9 où les lignes acheminant un chiffre redondant sont regroupés sous un seul trait. La structure générale du circuit ést donnée dans le figure 4.10.

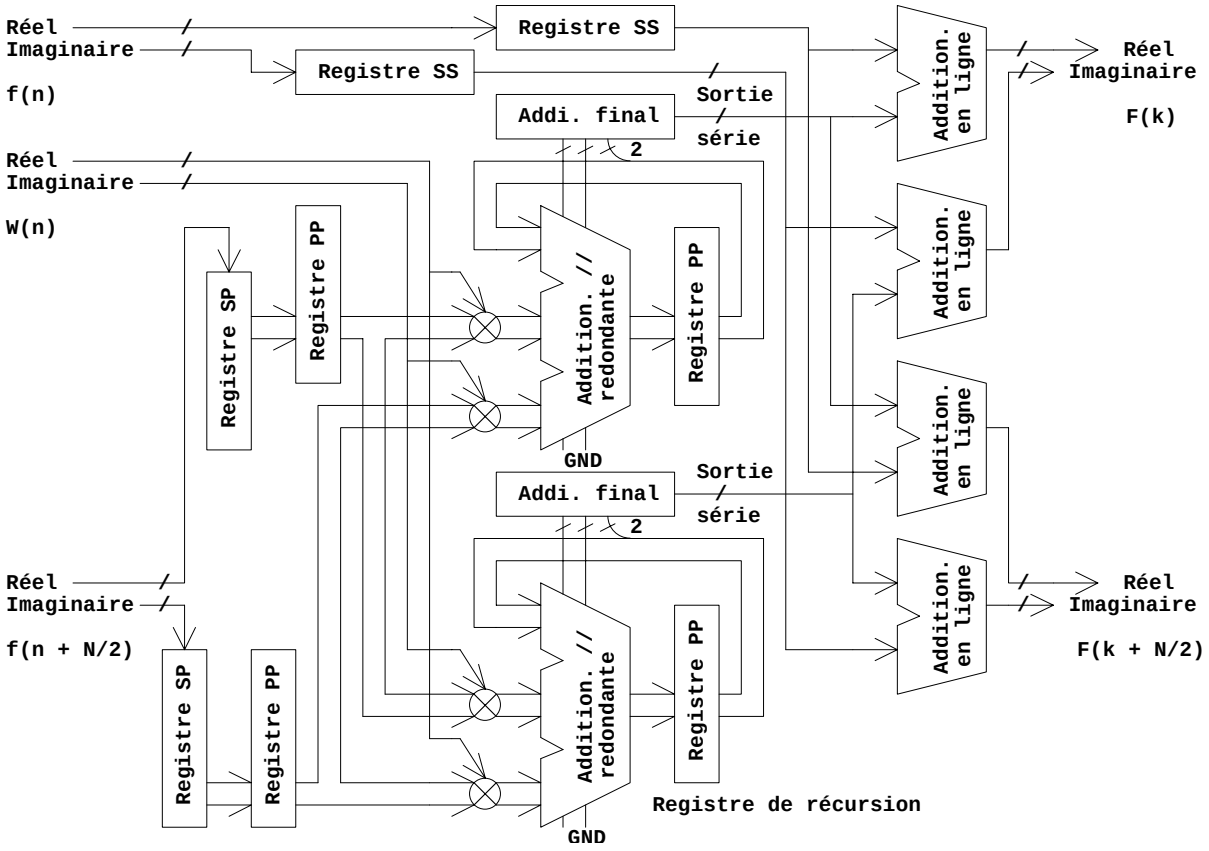

FIG. 4.9 – *papillon en ligne à base 2 aprés simplification et notamment présence de multiplieurs complexes.*

I.4.2.5 Commentaires.

L'EMPLOI D'OPÉRATEURS EN LIGNE A POUR AVANTAGE de pouvoir facilement redessiner un circuit selon les specifications d'un utilisateur. Cela est dû aux caractéristiques trés déterministes de ces opérateurs, tant au point de vue de la surface d'implantation, des communications entre opérateurs que de leurs temps de réponse.

Une modification de la taille des opérandes entraîne une dilatation de la taille des opérateurs associés. Hormis les additionneurs finaux, ils sont tous proportionnels à la taille des opérandes.

Les connexions entre des opérateurs sériels sont indépendants de la taille des nombres traités, ce qui n'est pas le cas d'opérateurs parallèles. Un changement de taille des opérandes à traiter ne se traduit donc que par une homothétie des cellules de base, sans remise en cause des connexions entre cellules. D'où une minimisation du travail lors du redessin du circuit pour l'adapter à une utilisation particulière.

En ce qui concerne les temps caractéristiques de fonctionnement du circuit d'autre part, le changement de taille des nombres n'entraine pas de problème ardu pour les déterminer. Une fois mesurés les temps des cellules de base, comme la retenue a une propagation limitée à son voisinage, le temps de calcul est entièrement déterminable et d'une forme $y = a \times x + b$ où b correspond à la latence du circuit et a à la taille des nombres de x chiffres.

Une application qui peut être citée concerne le choix de la précision des calculs. Cette dernière

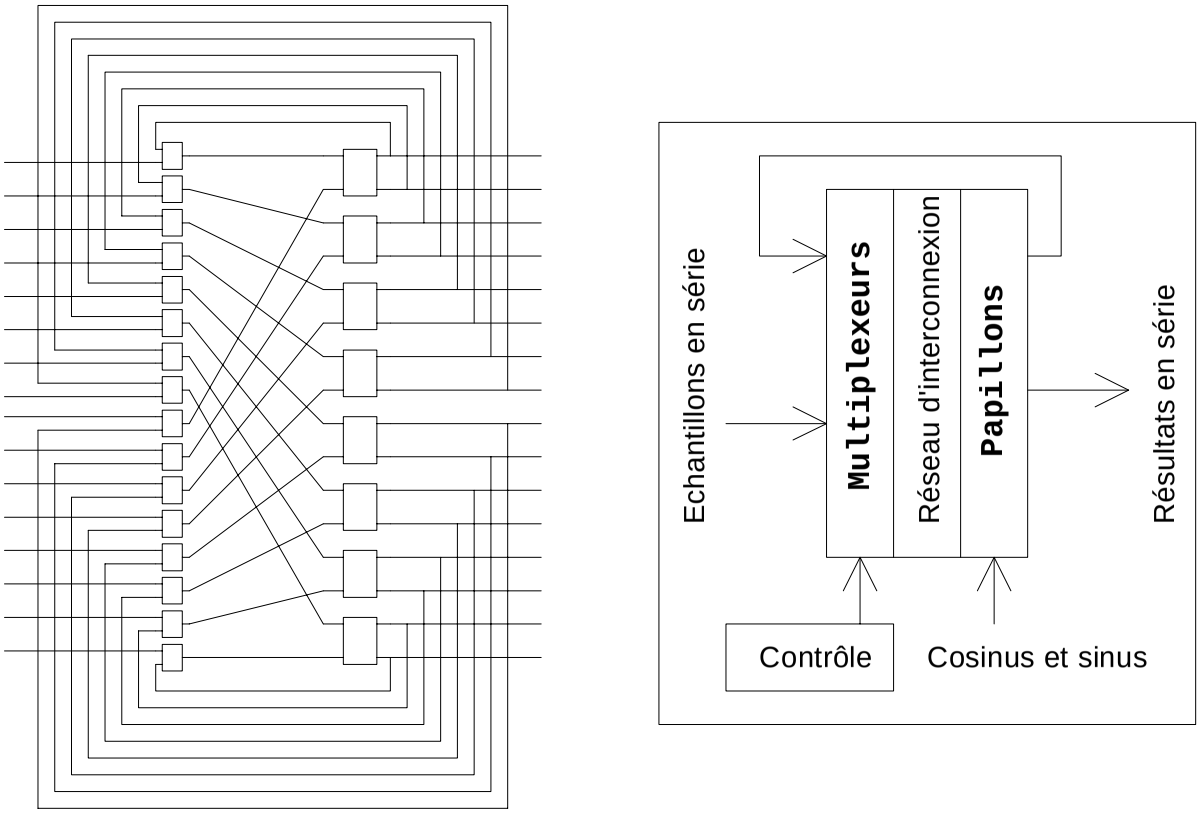

FIG. 4.10 – *structure du circuit choisie.*

étant liée au nombre d'échantillons, une variation de celui-ci entraine un changement de taille des opérandes pour adapter la précision du calcul. Avec de tels opérateurs, il est donc possible à l'utilisateur de travailler avec une précision à la demande, sans problème majeur de conception puisqu'il y a dilatation interne des papillons et aucun changement dans les temps caractéristiques de ces derniers.

I.4.3 Circuit réalisé

LE LABORATOIRE TIMA AVAIT À L'ÉPOQUE accés à une technologie lui permettant de faire fabriquer des circuits d'une surface inférieure ou égale à 190mm^2 et ayant au plus 144 broches [DJT+92]. Nous avons implanté [VBG+94] un circuit traitant seize échantillons qui atteint la limite des broches possibles, mais nécessite une surface plus faible que sa valeur maximale. Nous avons donc huit papillons qui réalisent leur calcul en quatre étapes successives. Les coefficients exponentiels sont stockés dans quatre ROMs, chacune correspondant à une étape de calcul. Le plan de masse du circuit est donné par la figure 4.11 et le dessin du circuit tel qu'il a été implanté par la figure 4.12.

Ce circuit n'était évidemment qu'une maquette vu le nombre d'échantillons traités qui nous permettait toutefois d'appréhender les problèmes qui restaient à résoudre pour mener à bien notre tâche.

Prévu pour la technologie $1,2\mu m$ de ES2, notre circuit peut travailler, selon les simulations électriques réalisées, avec une fréquence maximale de $27MHz$. Il peut ainsi calculer une T.F.R.

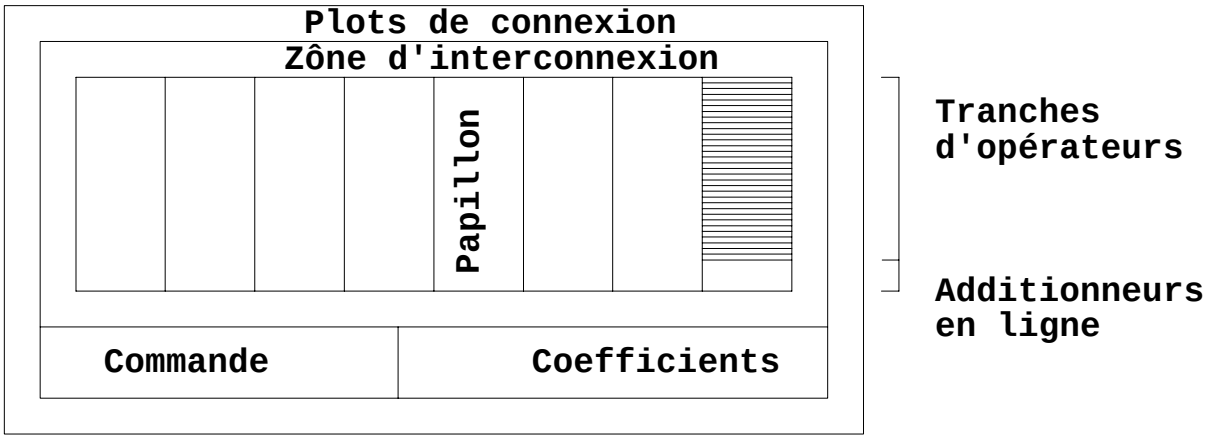

FIG. 4.11 – *plan de masse du circuit.*

de 16 échantillons de 32 bits en $3,43\mu s$, les nombres étant des entiers de 32 chiffres binaires redondants tout au long du calcul. Une extrapolation pour 1024 échantillons, avec la même précision, nous donne un temps de calcul de $877\mu s$ avec un seul de ces circuits et de $13,7\mu s$ si nous associons soixante-quatre de ces circuits en parallèle pour implanter la totalité de la barette de papillons qui respecte le schéma de l'architecture d'une T.F.R. repliée.

Comparons ces résultats avec des solutions similaires. Des processeurs généralistes de traitement du signal (TMS320C25 de Texas Instruments, ADSP2100 d'Analog Devices, DSP56001 de Motorola par exemple) permettaient à l'époque des calculs de cette précision, mais avec des temps de traitement plus long (entre 2,6 et $5ms$) [Col91]. Des circuits uniquement dédiés à la T.F.R. permettent des temps de calcul beaucoup plus courts ($10\mu s$) [F. 91] [J. 91] [Col91], mais avec une précision beaucoup plus faible, 12 ou 18 bits, adaptée au marché industriel qu'ils visent. Leur architecture interne est basée sur un nombre trés faible de papillons construits avec des opérateurs parallèles.

Face aux résultats de la concurrence, nous pouvons observer que nous faisons mieux qu'une solution à base de D.S.P. au niveau du temps, mais qu'une solution cablée à base d'opérateurs parallèles est plus performante, en terme de surface, en se souvenant que la taille de leurs opérandes, donc leur précision, est plus faible. Bien que parfaitement réalisable, notre solution à base de soixante-quatre puces est toutefois bien encombrante. Il faut toutefois noter que la réalisation industrielle qui se rapproche de notre démarche par la recherche d'une plus forte précision que la normale présente aussi les mêmes tendances. Il s'agit du produit conçu par les équipes de Sharp qui a connu une naissance difficile, pour des raisons apparemment plutôt commerciales, et qui utilise des opérateurs de vingt-quatre chiffres binaires.

L'examen de des performances de notre circuit amène à énoncer des pistes à explorer pour permettre la réalisation d'une architecture dédiée à la transformée de Fourier rapide utilisant des opérateurs en ligne :

- augmenter la fréquence de fonctionnement du système pour obtenir des temps de calcul plus courts,

- résoudre le problème de l'acheminement des données entre le circuit et son environnement, car le nombre maximal de broches possibles est atteint avec seulement seize échantillons, si un transfert sériel des données sans multiplexage est utilisé,

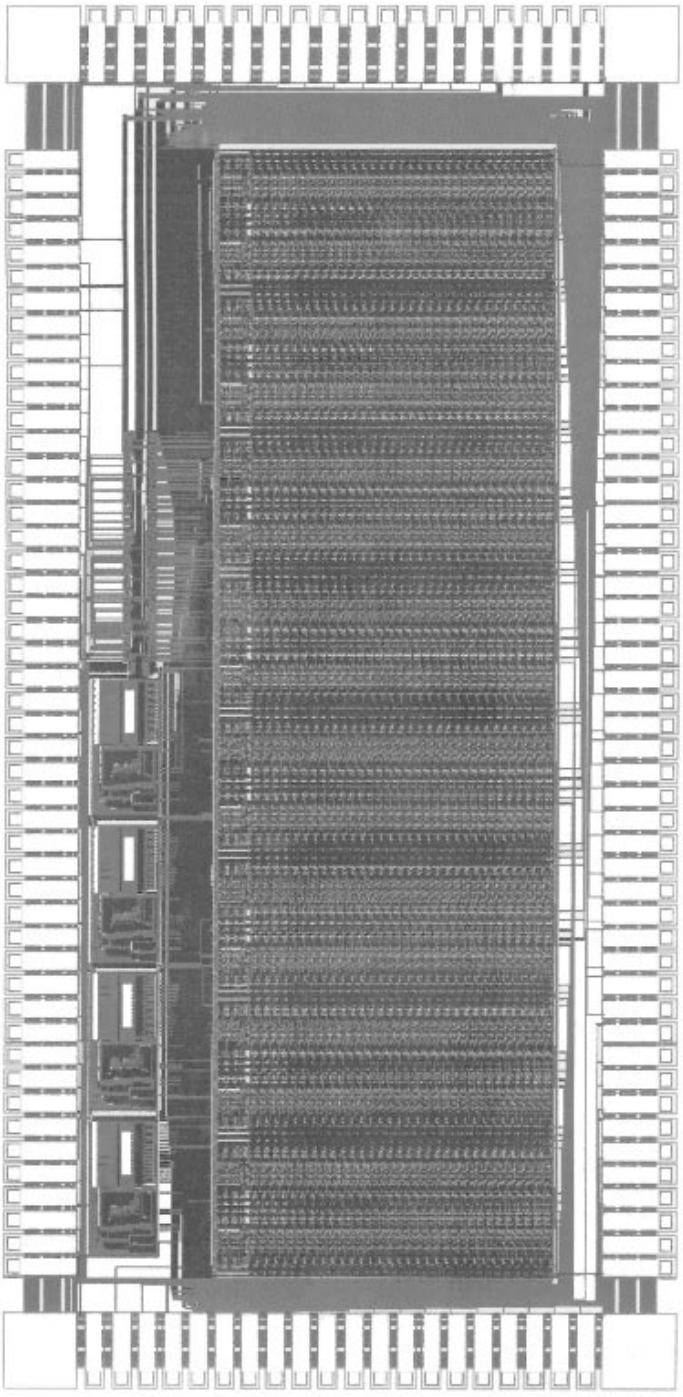

FIG. 4.12 – *dessin du circuit.*

– examiner les solutions existantes permettant un traitement d'un nombre d'échantillons plus important, en particulier quand leur nombre est trop grand pour être traité simultanément par un nombre suffisant de papillons ou en trouver d'autres.

I.4.4 Perspectives de ce travail.

I.4.4.1 Généralités.

L'ENCAPSULATION DES CIRCUITS INTÉGRÉS a fait des progrés notables qui peuvent repousser les problèmes du nombre de broches. Les technologies les plus avancées dans ce domaine autorisent plus de mille broches. Toutefois, les performances d'une solution basée sur les opérateurs tels qu'ils ont pu être conçus pourraient être fortement augmentés en considérant les différents points suivants.

I.4.4.2 Fréquence de travail.

LES DEUX STAGIAIRES QUI ONT DESSINÉ LES CELLULES DE BASE devaient mettre en oeuvre un logiciel de dessin de circuits intégrés destiné à faciliter la migration des implantations faîtes entièrement à la main, dîtes *full-custom*, au cours des migrations technologiques. La transformée de Fourier était l'occasion permettant de tester ce produit, PREFORM [DNJ90] [ND92], développé à l'École Nationale Supérieure des Télécommunications de Paris. Il ne faut toutefois pas chercher la cause de cette fréquence plutôt faible dans les contraintes de ce logiciel, car les cellules reprennent totalement les caractéristiques de celles conçues par Ali SKAF, qu'il s'agisse de la structure ou des performances en terme de surface ou de fréquence de travail obtenue par simulation électrique.

Nous avons ultérieurement repris cette étude, en abandonnant la technique de conception totalement à la main au profit d'un compilateur de chemin de données implanté par le logiciel de C.A.O. Compass. Même si dans l'absolu le résultat n'est pas optimum, car le fruit de compromis décidé ailleurs, cela permet de récupérer le savoir-faire et le travail des équipes d'ingénieurs ayant conçu les bibliothèques. Le résultat des simulations nous a donné, pour une même technologie de $1,2\mu m$ et une surface voisine, une fréquence de $80MHz$ en pipelinant les différents étages d'opérateurs successifs. Toutefois le gain qui peut être obtenu en vitesse ne peut masquer le problème de la surface d'implantation.

I.4.4.3 Architecture.

LE CHOIX DE L'ARCHITECTURE REPLIÉE pour implanter une T.F.R. a été fait à cause de l'aspect trés séduisant de cette technique qui permet, en simplifiant, de diviser le nombre de papillons par le logarithme du nombre d'échantillons. Cela est vrai si ce nombre est grand. Pour un nombre d'échantillons faible, cette architecture ne prend pas en compte les simplifications qui peuvent être faites au niveau des multiplications, lorsque les coefficients exponentiels font partie de l'ensemble $\{\pm 1, \pm j\}$. Or les opérateurs qui en résultent ont une surface d'implantation d'autant plus négligeable devant celle des multiplieurs que les opérandes sont grands, ce qui est notre cas avec notre choix d'une résolution à usage scientifique. Pour une T.F.R. étalée de seize échantillons, il ne faut pas plus de papillons comprenant des multiplieurs que la même T.F.R. repliée. Si une T.F.R. d'une grande taille est calculée avec des T.F.R. de seize échantillons qui constituent alors des papillons de la macrobase, le temps de calcul est bien plus faible avec une architecture étalée, car le débit est beaucoup plus grand.

Si nous mettons de côté le problème de l'acheminement des données entre le circuit et son environnement, nous pourrions implanter beaucoup plus de papillons. Ceux qui font partie de l'implantation qui a été menée jusqu'au dessin complet du circuit utilisent chacun une surface de $1,057mm \times 2,66mm \simeq 2,8\text{mm}^2$, soit en tout à peu prés 45mm^2 à rapporter aux 190mm^2 autorisés à l'époque.

C'est pourquoi, nous avons estimé le temps de calcul d'une T.F.R. de 1024 échantillons avec les deux architectures et différentes valeurs de la macrobase de la T.F.R., sachant que nous supposons que l'ensemble des données utilise un seul papillon de la macrobase considérée. Les résultats sont représentés sur la figure 4.13. Nous avons représenté le temps de calcul de la T.F.R. en fonction du nombre d'échantillons du circuit qui la calcule pour les deux familles d'architecture et pour les bases 2 et 4. Nous pouvons observer le gain extrêmement important obtenu avec une architecture étalée qui permet notamment d'enfoncer le plancher des $10\mu S$ obtenu par les implantations industrielles à base d'opérateurs parallèles. Le faible écart pour une architecture étalée entre les bases 2 et 4 provient du fait que la notation redondance autorise une latence faible pour les opérateurs arithmétiques. Donc la partie du temps de calcul liée aux retards créés par les circuits a, dans ce cas là, une influence plus faible que dans une notation en complément à deux.

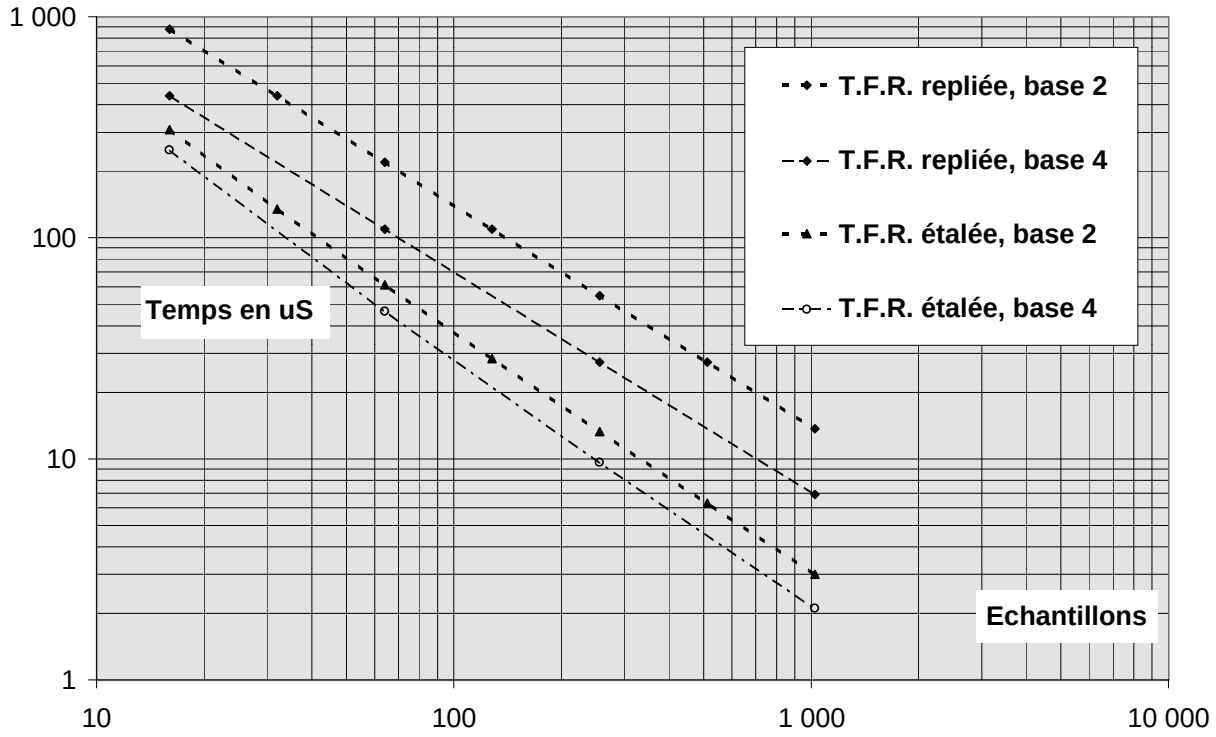

FIG. 4.13 – *temps de calcul total d'une T.F.R. de* 1024 *points avec un seul circuit qui traite un nombre d'échantillons indiqué en abscisse pour différents architectures et implantations.*

Il va de soi que, pour un nombre d'échantillons implantés donné et une base fixée, le nombre de papillons arithmétiques nécessitant un multiplieur n'est pas égal pour les deux architectures. Ce sont les papillons comportant des multiplieurs qui nécessitent le plus de surface. Pour pouvoir comparer les différentes solutions entre elles, nous avons représenté dans la figure 4.14 la surface de ces papillons en fonction du nombre d'échantillons implantés. Nous pouvons constater, bien

que cette estimation ne prend pas en compte les papillons se réduisant à des additionneurs, que par rapport aux estimations faîtes dans le chapître 3 nous obtenons une densité d'intégration suffisamment meilleure pour autoriser une implantation étalée de 64 échantillons. Ce qui peut nous laisser espérer avec les meilleures technologies disponibles à la fin de cette thèse qui atteignent les $0,35\mu m$ une implantation de 256 échantillons. En ce qui concerne les papillons arithmétiques en base 4 nous avons considéré que leurs surfaces devaient être approximativement quatre fois plus grandes, comportants quatre fois plus de multiplieurs.

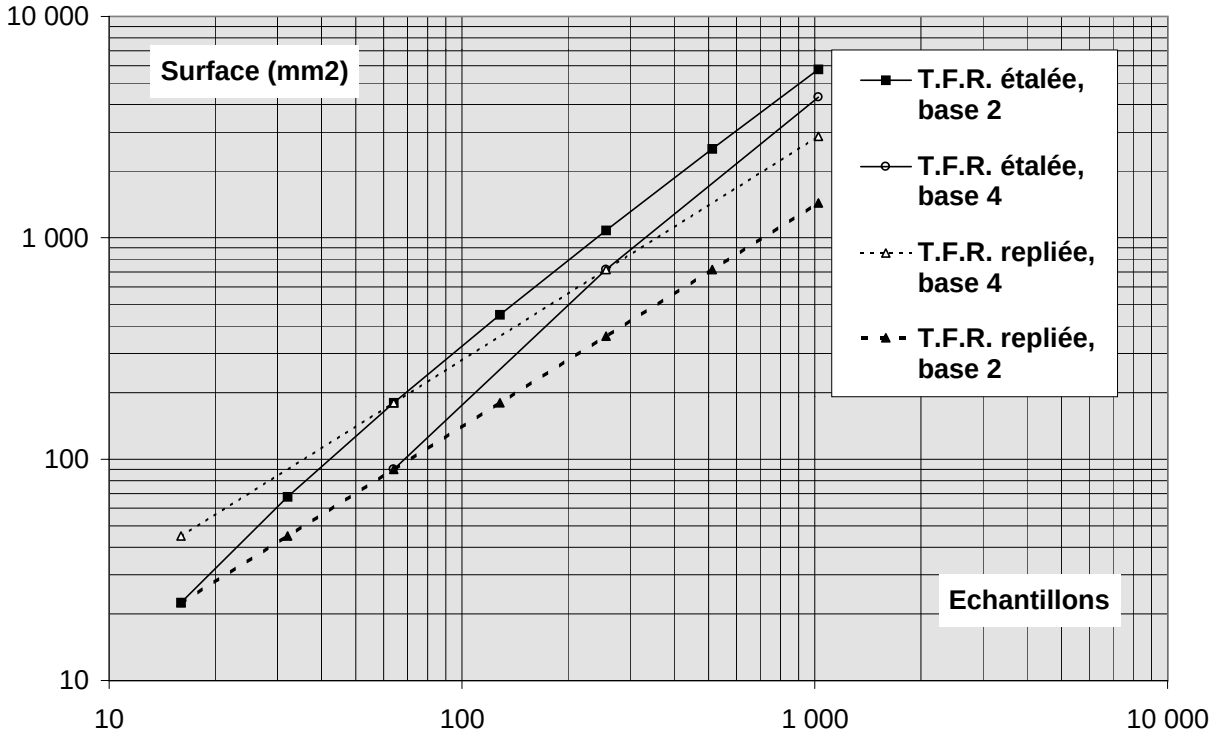

FIG. 4.14 – *surface d'implantation en fonction du nombre d'échantillons.*

Pour ce qui concerne le temps de calcul, nous devons nous souvenir que le circuit qui a été dessiné ne fonctionne qu'à moins de $30MHz$, alors qu'une conception faîte en pipelinant les opérateurs nous permettraient d'atteindre des fréquences trés supérieures comme nous l'avons fait remarquer au paragraphe 4.4.2. D'autre part la taille des opérandes est de 32 chiffres binaires. Si nous considérons ces deux facteurs, nous atteignons des temps de calcul tout à fait comparables aux réalisations industrielles concurrentes, en prenant une solution monopuce en $1,2\mu m$ pour notre produit. Nous pouvons nous souvenir que le circuit de Dassault Électronique a, lui aussi, été réalisé grâce à une technologie $1,2\mu m$ et met en oeuvre quatre puces pour atteindre les $10\mu s$, mais avec seulement une précision de 12 chiffres binaires. Ce qui à technologie et nombre de composants comparables, nous amène des résultats plus intéressants.

Toutes ces considérations prometteuses ne doivent pas faire oublier que nous n'avons pas abordé le problèmes des entrées et sorties de nos circuits, ni celui du stockage des données intermédiaires. De plus le nombre de 1024 échantillons ne constitue pas un grand nombre d'échantillons, pour des transformées multi-dimensionnelles du moins.

I.4.4.4 Conclusion.

Cette étude d'implantation a eu le mérite de montrer qu'une implantation de papillons sériels, notamment en ligne, est envisageable au même titre que les solutions concurrentes à base d'opérateurs parallèles. Elle ne permet pas cependant d'espérer de miracle au niveau de la surface d'implantation. Notons toutefois que les règles de simplification, suppression de registres contenant des données stockées ailleurs notamment, développées par les concepteurs et rappelées dans le paragraphe 4.2.4 (page 60) apportent des gains significatifs par rapport aux extrapolations faîtes à partir des opérateurs simples de base telles celles du chapître 3 (page 35). Quand au problème du temps de calcul, il est difficile d'affirmer quelque chose de définitif, les équipes ayant développé des solutions industrielles n'ayant pas les mêmes moyens, ni les mêmes conditions de travail que les notres.

Deuxième partie

Développements théoriques

Chapitre 1

Architecture à saturation de bus

Sommaire

II.1.1 Introduction. **73**
II.1.2 Décomposition d'une T.F.R. sur plusieurs circuits. **73**
II.1.2.1 Dimension d'une transformée et choix d'une macrobase. 73
II.1.2.2 Surface d'une T.F.R. réduite. 74
II.1.2.3 Coût de la mémoire. 76
II.1.2.4 Échanges de données entre papillons sériels. 77
II.1.3 Communication des données. **77**
II.1.3.1 Généralités. 77
II.1.3.2 Architecture à saturation de bus. 78
II.1.3.3 Évolutions. 84
II.1.4 Adaptations. **84**

II.1.1 Introduction.

UNE ARCHITECTURE MASSIVEMENT PARALLÈLE de papillons permettrait d'obtenir de trés hautes performances pour le calcul de transformées de Fourier. Des opérateurs sériels autorisent plus facilement que des opérateurs parallèles une implantation de ce type. Nous avons pourtant vu dans les chapîtres précédents que cela reste de la fiction, à moins de mettre en oeuvre un nombre considérable de puces. Le paragraphe 1.2 nous permet d'envisager les moyens d'atténuer ce problème. Le choix d'opérateurs sériels répartis à l'intérieur d'un grand nombre de puces pose en lui-même le problème de la communication des données entre les différents circuits. Nous présentons dans le paragraphe 1.3 notre proposition pour résoudre ce problème et nous en évaluons les paramètres et les principales caractéristiques.

II.1.2 Décomposition d'une T.F.R. sur plusieurs circuits.

II.1.2.1 Dimension d'une transformée et choix d'une macrobase.

LE CALCUL DE TRANSFORMÉES MULTIDIMENSIONNELLES amène au calcul de transformées monodimensionnelle de taille plus réduite. Par exemple, une T.F.R. $3D$ de 16 millions de points conduit à des T.F.R. $1D$ de 256 points. Comme ce dernier nombre correspond à ce qui est implantable en silicium ou proche de l'être, il va de soi que ce genre de situation impose plus ou moins

le choix de la macrobase. Il ne faut toutefois pas oublier que le coté multidimensionnel d'une T.F.R. agit sur les valeurs des coefficients exponentiels des papillons, mais pas sur le devenir des résultats intermédiaires tout au long du calcul complet. Ce qui peut se dire autrement, à savoir que la communication des résultats intermédiaires entre les papillons des étapes successives reste constant par rapport au nombre de dimensions des espaces de travail, étant uniquement fonction du nombre total des échantillons à traiter. Par rapport à une T.F.R. $1D$ d'une taille équivalente à la T.F.R. $3D$ totale, ce dernier point est une constante.

Il ne faut donc pas écarter à priori la possibilité de macrobases mixtes si la technologie permet des implantations plus grandes que celles correspondant à chaque dimension, sachant que dans ce cas cela limiterait les échanges de données. En effet, plus la macrobase est grande, moins il y a d'étapes successives de T.F.R. réduites, phénomène identique au passage d'une base 2 à une base 4. Donc, moins il y a de transferts entre les barettes de papillons. Cela complique certes la gestion des coefficients, mais simplifie les échanges de données.

Supposons par exemple que nous puissions réaliser une T.F.R. de 4096 échantillons. Cela peut correspondre à une T.F.R. de 256 échantillons, une dimension complète de l'espace, suivie d'une série de 16, première partie de la deuxième dimension. L'étape suivante comprendrait une série de T.F.R. de 16 échantillons, dernière partie de la deuxième dimension, suivie d'une T.F.R. de 256 échantillons, la troisième dimension complète. Nous avons représenté sur la figure 1.1 une T.F.R. de 16 millions d'échantillons calculée avec une architecture cablée utilisant la décomposition en macrobase selon les dimensions de l'espace de travail. La figure 1.2 représente la même T.F.R., mais calculée avec des circuits intégrant des papillons d'une complexité équivalente à celle d'une T.F.R. de 4096 échantillons, le détail interne de ces derniers circuits est précisé dans la figure 1.3 où les entrées et sorties dans le sens du flot des données de l'architecture sont appelées respectivement fi et Fi, celles qui sont échangées entre les circuits appartenant à la même étape de calcul $f'i$ et $F'i$.

II.1.2.2 Surface d'une T.F.R. réduite.

RÉDUIRE LE NOMBRE DE PAPILLONS d'une T.F.R. est réalisable facilement en faisant l'opération inverse de celle que nous avons rappelé pour les T.F.R. multidimensionnelles : transformer une T.F.R. $1D$ en une T.F.R. $2D$.

Soit une T.F.R. de N échantillons et d'indices n et k. Reécrivons les indices pour permettre cette transformation $1D \rightarrow 2D$, $n = \sqrt{N}.n2 + n1$ et $k = \sqrt{N}.k1 + k2$. Nous avons les expressions suivantes :

$$\begin{aligned} F(k) &= \sum_{n=0}^{N-1} f(n) * \omega^{n*k} \\ &= \sum_{n_1=0}^{\sqrt{N}-1} \left(\sum_{n_2=0}^{\sqrt{N}-1} f(n_1, n_2) * \omega_{r2}^{n_2*k_2} \right) * \omega_{r1}^{n_1*k_1} \end{aligned}$$

où les ω_{r1} et ω_{r2} sont définis selon le type de décomposition, temporelle ou fréquentielle, de la même façon que dans la théorie de la T.F.R. de base 2.

Ceci est équivalent à avoir une T.F.R. à $2D$ avec une gestion un peu spéciale des coefficients exponentiels. Pour limiter la complexité de la communication des données Chowdary a proposé[*et al.*84] de calculer $\sqrt{N}$ T.F.R. avec une valeur fixe d'un indice et ensuite, aprés permutation des indices, $\sqrt{N}$ autres avec une valeur fixe de l'autre indice. Pour limiter le nombre de papillons, nous pouvons calculer successivement les différentes T.F.R. monodimensionnelles

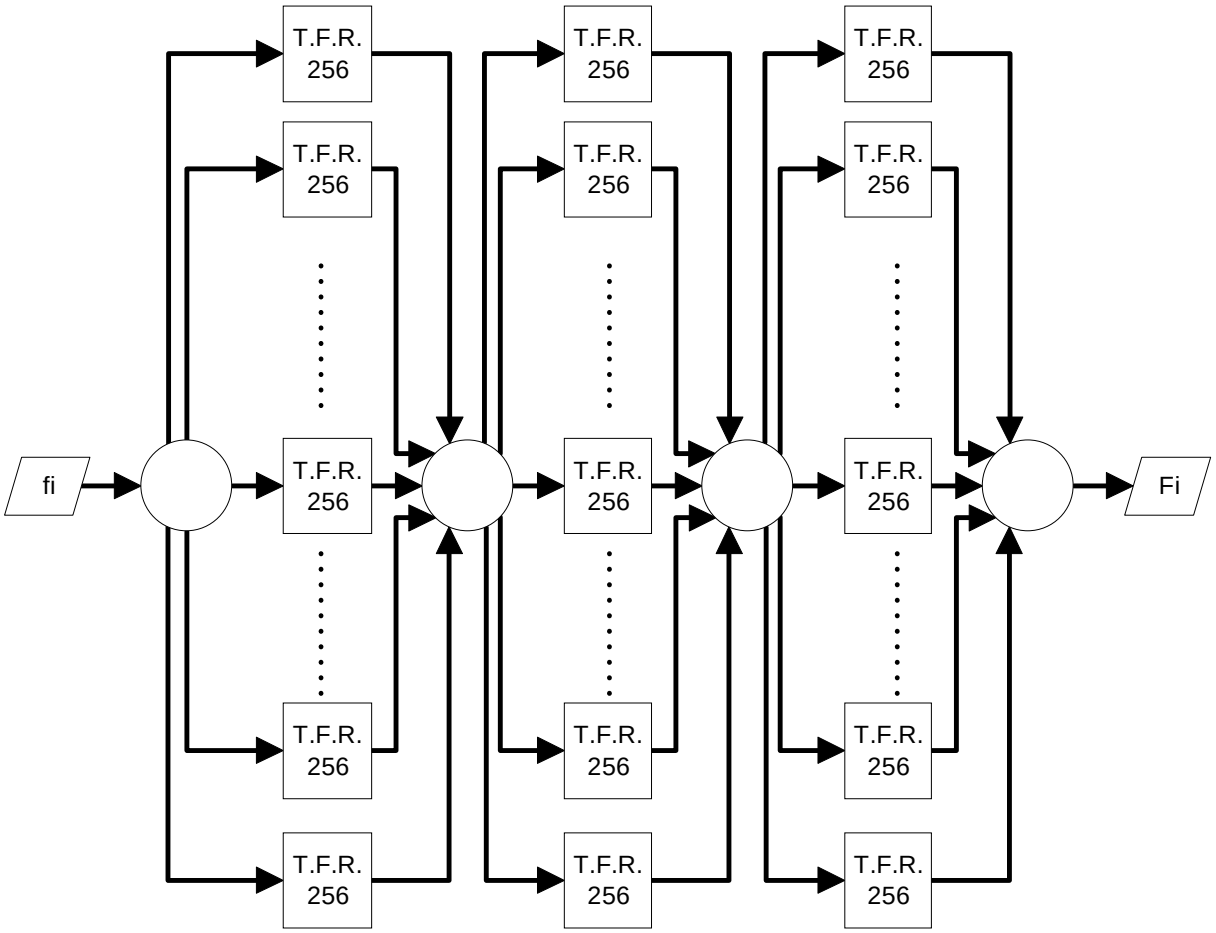

FIG. 1.1 – *T.F.R. 3D de* 16 *millions d'échantillons calculée avec des T.F.R. cablées de* 256 *échantillons.*

ayant apparu dans cette méthode. Cela impose l'utilisation d'une mémoire tampon dont la surface n'a toutefois rien à voir avec les papillons quelle fait disparaître et cela abaisse forcément la vitesse de calcul, mais c'est le prix à payer pour disposer d'une solution compatible avec des budgets du domaine du réalisable. La figure 1.4 montre un exemple d'une T.F.R. de 64 échantillons calculées en 16 T.F.R. successives de 8 échantillons.

La mémoire a une structure matricielle, les entrées se faisant selon les lignes de la structure matricielle et les sorties selon les colonnes. Nous avons ainsi, pour une complexité réduite, à la fois le stockage des données et la permutation des indices. Cette sorte d'opération pourrait être renouvelé, mais les circuits intégrés sont pour le moment conçu sur un plan et non dans l'espace. Les permutations de plus de deux indices sont donc beaucoup plus complexes pour être envisageables dans des cas comme ceux-ci.

Le fait d'utiliser de telles structures permet de réduire le nombre apparent d'échantillons de la T.F.R. véritablement cablée et donc de rendre intéressant le choix d'une architecture étalée pour bénéficier des simplifications dans les premières étapes au niveau des coefficients exponentiels.

Nous avons vu qu'une T.F.R. pouvait être implantable pour un nombre de 64 échantillons. Ce qui permet d'espérer des tailles de $64^2 = 4096$ échantillons.

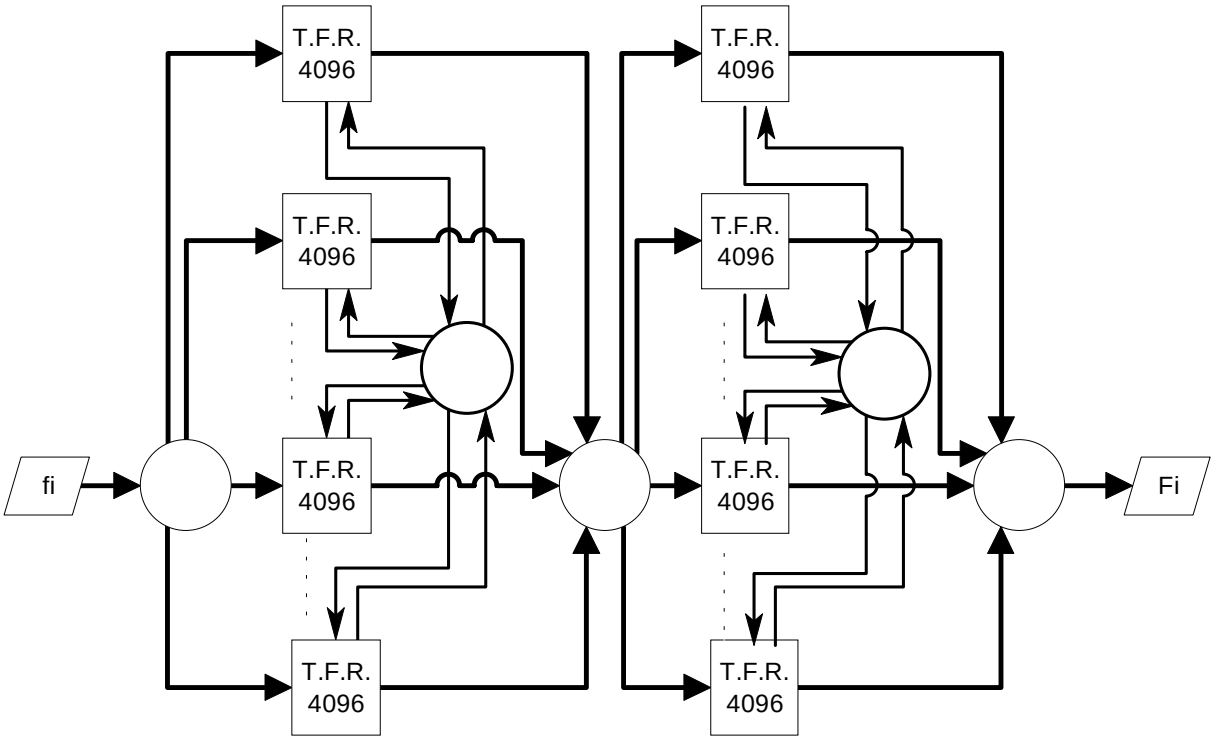

FIG. 1.2 – *T.F.R. 3D de* 16 *millions d'échantillons calculée avec des T.F.R. cablées de* 256 *échantillons et des T.F.R. cablées de* 16 *échantillons groupées dans un même circuit.*

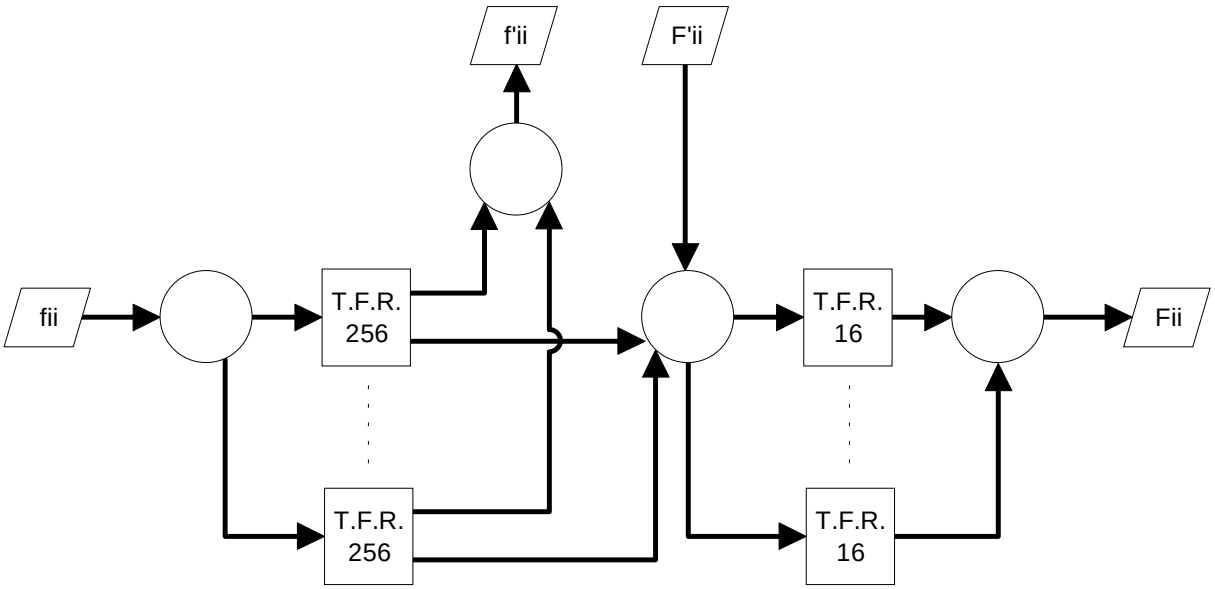

FIG. 1.3 – *structure interne des circuits intégrant les T.F.R. de* 256 *et* 16 *échantillons.*

II.1.2.3 Coût de la mémoire.

CETTE STRUCTURE MATRICIELLE demande que chaque cellule de cette matrice de transposi-

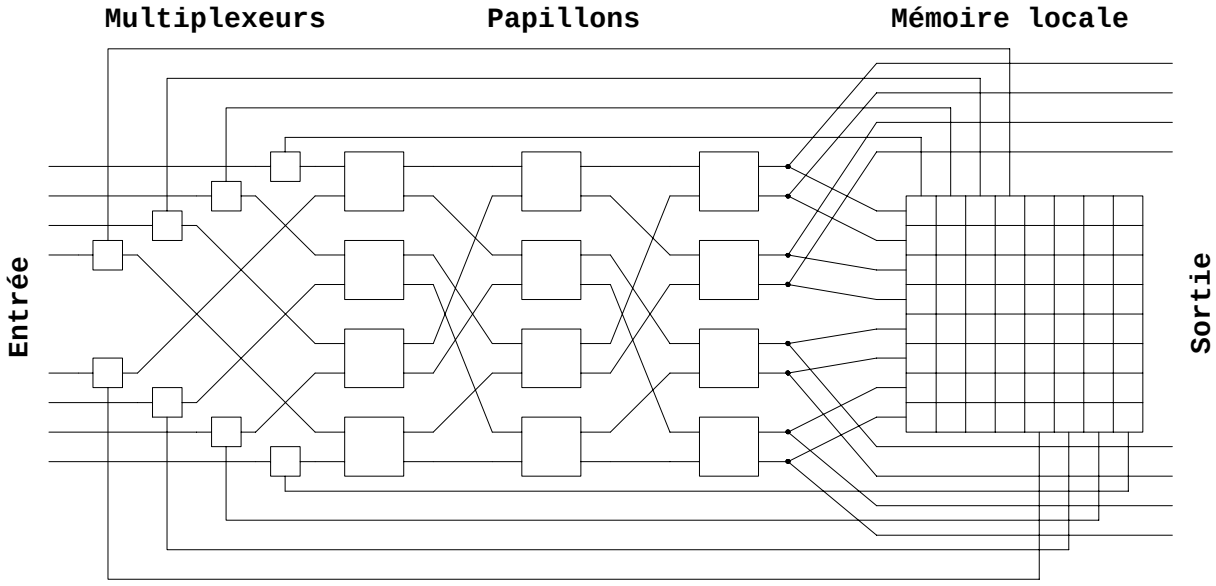

FIG. 1.4 – *T.F.R. de* 64 *échantillons calculée avec une T.F.R. cablée de* 8 *échantillons.*

tion implante un registre à décalage de taille équivalente au nombre à stocker. Nous pouvons l'estimer pour une configuration particulière, celle de la figure 1.4, soit 64 échantillons, avec des opérandes équivalents à ceux du chapître II.4, soit des nombres redondants de 32 chiffres. Ce qui amène $64 \times 32 \times 2$ cellules élémentaires d'un registre à décalage typiquement réalisées avec une bascule *R-S* commandée par une horloge nécessitant huit portes logiques à deux entrées type *OU-NON* ou *ET-NON*, soit 32768 portes ou encore 131072 transistors à comparer aux 4096 variables logiques stockées. Une solution à base de mémoire SRAM est beaucoup moins coûteuse, 6 transistors par point mémoire soit 24576 transistors non comptée la logique de commande, mais se révèle moins rapide d'accés. Pour une implantation atteignant des fréquences de 80 MHz ou supérieures comme celle qui a été étudiée partiellement dans le chapître II.4 une solution à base de RAM se révèlerait incompatible avec ses temps d'accés. Ce qui nous a amené à la solution exposée ci-aprés dans le paragraphe II.1.3.

II.1.2.4 Échanges de données entre papillons sériels.

SI LE NOMBRE DE PAPILLONS devient important, les techniques de perfect shuffle exchange se révèlent couteuses en surface, car proportionnelles au nombre d'échantillons N. D'autres techniques plus astucieuses, mais plus difficiles à mettre en oeuvre, ont été proposées dans le passé [G. 83]. Le coût en surface est proportionnel à $\log_2 N$.

II.1.3 Communication des données.

II.1.3.1 Généralités.

UNE T.F.R. COMPLÈTEMENT PARALLÉLISÉE conduit pour être implantable à un trés grand nombre de connexions entre les sorties de la barette de papillons d'un étage vers les entrées de la barette suivante, cas d'une architecture étalée, ou vers ses propres entrées, dans le cas d'une architecture repliée. Ce phénomène est coûteux en terme de surface si l'implantation est faîte dans un seul circuit. Il devient trés génant dans des solutions composées de plusieurs puces en raison du nombre de broches nécessaires. Le coût financier augmente extrêmement vite dés

que le nombre de broches sort des valeurs courantes en raison de leur trés faible emploi, cercle vicieux auquel se heurte bon nombre de technologies d'avant-garde. D'autre part, il ne faut pas négliger le coût induit par la surface de routage de ces connexions sur les circuits imprimés supportant ces cartes. Pour réduire le poids de ces deux points, il est possible de multiplexer les données transmises sans pénalité pour les opérateurs arithmétiques qui sont plus lents que la simple transmission de données. Nous pouvons toutefois remarquer que :

- la gestion correspondante est coûteuse en terme de surface et n'est pas partageable entre plusieurs circuits, sauf à utiliser de nombreux signaux de commande coûteux en nombre de broches, ce qui va à l'encontre du but poursuivi, et à résoudre les problèmes de respect de synchronisme que cela entraine.

- La présence d'une mémoire tampon, dans laquelle peuvent être entreposées les données entre les étapes de calcul et de transfert qu'elles subissent, permet de dissocier le mode de traitement des données au sein de ces deux opérations.

- Dans des papillons de macrobase p conçus avec une transformation $2D \rightarrow 1D$, la mémoire a des caractéristiques temporelles compatibles avec la fréquence de fonctionnement des opérateurs arithmétiques pour optimiser la surface d'implantation.

Si nous voulons conserver cette dernière caractéristique tout en multiplexant les données, nous avons le choix entre :

- lire les données de la mémoire en parallèle dans un premier temps et utiliser ensuite un tampon dont la vitesse est compatible avec l'interface de transfert,

- lire les données de la mémoire en série, puis multiplexer les signaux avant d'alimenter l'interface vers l'extérieur.

II.1.3.2 Architecture à saturation de bus.

NOTRE PROPOSITION est d'utiliser des bus parallèles pour transférer les données entre des cellules de calcul mettant en oeuvre des opérateurs sériels [VA94a]. Ces bus doivent fonctionner sans temps mort, au détriment si nécessaire du taux d'utilisation des opérateurs arithmétiques sériels réalisant les calculs proprement dits. L'optimum étant que les opérateurs travaillent à une fréquence telle qu'ils aliment les bus de connexion sans à avoir à attendre un intervalle de temps libre pour évacuer leurs données vers les étages suivant. Pour résumer la situation, *une architecture à saturation de bus* consiste à :

- utiliser des opérateurs sériels pour leur densité d'intégration et leur fréquence de travail,

- faire appel à des bus parallèles pour l'importance de leur débit, leur simplicité de mise en oeuvre et le fait que les problèmes dûs à leur emploi sont des phénomènes bien connus, donc plus facilement maîtrisables

- répartir au niveau des papillons de macrobase de la T.F.R. la mémoire contenant les données pour supprimer les problèmes de leur transfert entre une mémoire centralisée et les cellules de calcul et diminuer le débit d'information d'échange en ralentissant artificiellement le papillon de la macrobase,

- diminuer la surface d'implantation nécessaire pour les cellules de calcul.

Cette approche va à l'encontre d'autres démarches qui consistent à optimiser la gestion des flots de données entre une mémoire commune les contenant et la partie calcul de l'architecture spécialisée dédiée à ces calculs [Ver94].

Notre démarche permet de :

- conserver un cheminement cohérent des données tout au long des échanges entre puces de calcul et de supprimer la phase de sérialisation des données,
- limiter la fréquence des interfaces intégrées dans les puces et éventuellement déporter vers des puces spécialisées le problème du multiplexage comme représenté dans la figure 1.5, en faisant éventuellement appel à des technologies plus performantes que le CMOS en silicium, comme l'arséniure de gallium. Le nombre de bus n'est pas forcément égal pour les deux niveaux de communications.

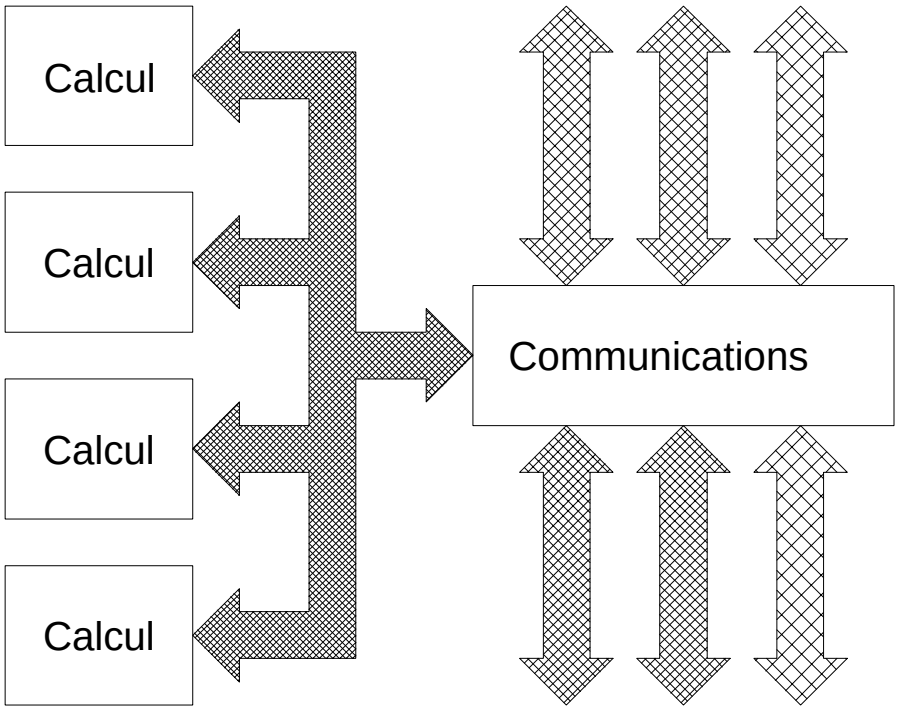

FIG. 1.5 – *architecture à saturation de bus à deux niveaux.*

Si le *Ga As* pourrait être utilisé pour les puces de calcul, cela pose le problème de l'intégration d'un trés grand nombre de portes qui n'est pas actuellent vraiment résolu. Par contre, son utilisation est suffisamment répandue en communication rapide pour pouvoir être envisagée dans cette fonction sans poser de problème à priori insurmontable. Le problème du double accés de la mémoire, l'un sériel et l'autre parallèle, n'est pas lui non plus un problème totalement nouveau, puisqu'il est à la base des Video-RAM. Il s'agirait donc d'une généralisation des principes utilisés dans ce dernier type de mémoire.

Soit N le nombre d'échantillons à traiter, B la taille des opérandes et p la macrobase de la T.F.R. générale, $T_c = \frac{1}{F_c}$ la période de l'horloge des opérateurs arithmétiques, $T_t = \frac{1}{F_t}$ celle des bus de communications. Il est tout à fait possible d'avoir plusieurs bus sur un circuit pour améliorer la vitesse de transfert. Appelons b le nombre d'opérandes qui sont transmis durant un

cycle du transfert. La variable b est une fonction du nombre de broches d'une puce consacrée aux bus et de B. En première approximation, nous avons égalité dans une solution optimale entre le temps de transfert des données d'une étape et le temps de calculs de ces données.

Avec une notation en complément à deux, chaque papillon de base arithmétique crée approximativement une latence de B cycles d'horloge pour exécuter le calcul d'un résultat intermédiaire. Chaque papillon de macrobase est constitué d'un calcul de $2 * \sqrt{p}$ T.F.R. élémentaires de $\log_2 \sqrt{p}$ étapes de calcul en base 2. Chaque échantillon ou résultat intermédiaire des étapes de calcul est constitué d'une partie réelle et d'une partie imaginaire. Nous avons donc :

$$\begin{aligned} \frac{N * T_t * 2}{b} &= [(B * T_c) * (1 + \log_2 \sqrt{p})] * 2 * \sqrt{p} \qquad (1.1) \\ &= B * T_c * \left(1 + \frac{1}{2} * \log_2 p\right) * 2 * \sqrt{p} \\ &= B * T_c * (2 + \log_2 p) * \sqrt{p} \\ \iff N &= \frac{B * b}{2} * \frac{T_c}{T_t} * \sqrt{p} * (2 + \log_2 p) \\ \iff N &= \frac{B * b}{2} * \frac{F_t}{F_c} * \sqrt{p} * (2 + \log_2 p) \qquad (1.2) \end{aligned}$$

La figure 1.6 présente le nombre d'échantillons, points à fond noir et échelle de droite, et de papillons correspondant, points à fond blanc et échelle de gauche, en fonction de la macrobase, pour un rapport $\frac{F_t}{F_c} = 2$ et pour différentes valeurs du produit $B \times b$ qui correspond à peu prés au nombre de broches consacrées aux bus de transfert. Lorsque la macrobase est égale à 4, cette configuration se rapproche d'une architecture étalée de base 4 classique. Dans ce cas, plusieurs papillons peuvent être intégrés dans une même puce. Lorsque la macrobase augmente, une puce est consacrée à un seul papillon de la macrobase et pour de fortes valeurs de celle-ci, probablement à partir de la dizaine de milliers avec les technologies submicroniques actuelles, il est nécessaire de consacrer plusieurs puces pour un seul papillon.

Il faut se souvenir que le rapport $\frac{F_t}{F_c}$ que nous avons considéré au niveau de la puce fait en réalité intervenir la fréquence de transfert du système global. En particulier, si des technologies comme le $Ga\ As$ permet des bus trés rapides, le nombre d'échantillons en sera augmenté d'autant. Il en va de même avec le nombre de bus. Cela permet donc d'obtenir des tailles de T.F.R. compatibles avec des espaces multidimensionnels de travail.

Le temps de calcul T peut lui aussi être estimé. La T.F.R. est constituée de $\log_p N$ étapes de calcul. Ce qui nous donne :

$$\begin{aligned} T &= [\{(B * T_c) * (1 + \log_2 \sqrt{p})\} * 2 * \sqrt{p}] * \log_p N \qquad (1.3) \\ &= B * T_c * \left(1 + \frac{1}{2} * \log_2 p\right) * 2 * \sqrt{p} * \log_p N \\ \iff T &= B * T_c * (2 + \log_2 p) * \sqrt{p} * \log_p N \qquad (1.4) \end{aligned}$$

Si p est suffisamment grand, nous avons $2 + \log_2 p \simeq \log_2 p$. Ce qui est atteint sans problème avec la transformation $1D \rightarrow 2D$ décrite ci-dessus et les valeurs de p autorisées par l'intégration telle qu'elle a été mise en oeuvre et décrite dans le chapître 4. Le temps de calcul peut être approché avec une valeur plus simple :

$$\begin{aligned} T &\simeq T_c * B * \log_2 p * \sqrt{p} * \log_p N \qquad (1.5) \\ &\simeq T_c * B * \log_2 p * \sqrt{p} * \frac{\log_2 N}{\log_2 p} \\ \iff T &\simeq T_c * B * \sqrt{p} * \log_2 N \qquad (1.6) \end{aligned}$$

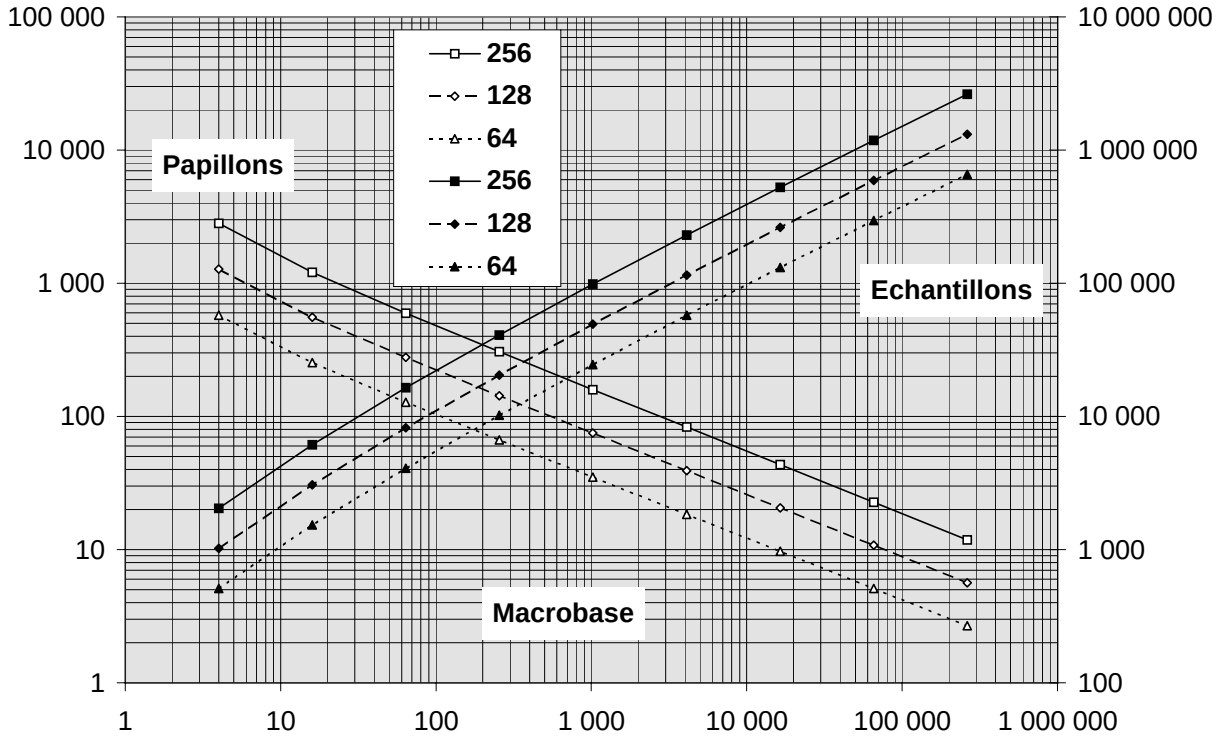

FIG. 1.6 – *nombre de papillons selon la macrobase en complément à* 2.

Nous pouvons remarquer que T est $\sqrt{p}$ fois plus grand que si le calcul avait été faît avec une T.F.R. n'utilisant pas cette technique de transformation $1D \rightarrow 2D$. Or cette méthode permet de diviser la surface par $\sqrt{p}$. Le produit $S \times T$ est donc constant.

La figure 1.7 représente le temps de calcul, points à fond blanc et échelle de gauche, en fonction de la macrobase pour la valeur $B \times b = 64$. Nous avons pris le cas de nombres d'une taille $B = 32$ pour rester dans les nombres de taille compatible avec des applications scientifiques. Pour d'autres valeurs de ce produit, la courbe reste proche et de même tendance. L'évolution du nombre d'échantillons traités en fonction de la macrobase est rappelé, points à fond noir et échelle de droite. Nous pouvons constater que le temps de calcul est grossièrement proportionnel au nombre d'échantillons N au lieu de l'être à $N \times log_2 N$.

Avec une notation redondante, chaque papillon de base arithmétique introduit une latence de 7 cycles d'horloge pour exécuter le calcul d'un résultat intermédiaire. Nous avons donc :

$$\begin{aligned}
\frac{N * T_t * 4}{b} &= [(B + 7 * \log_2 \sqrt{p}) * T_c] * 2 * \sqrt{p} \qquad (1.7)\\
&= \left[\left(B + 7 * \frac{1}{2} * \log_2 p\right) * T_c\right] * 2 * \sqrt{p}\\
&= [(2 * B + 7 * \log_2 p) * T_c] * 2 * \sqrt{p}\\
\Longleftrightarrow N &= \frac{2 * B + 7 * \log_2 p}{4} * \frac{T_c}{T_t} * b * \sqrt{p}\\
\Longleftrightarrow N &= \frac{2 * B + 7 * \log_2 p}{4} * \frac{F_t}{F_c} * b * \sqrt{p} \qquad (1.8)
\end{aligned}$$

Nous pouvons remarquer que, pour peu que $7 \times \log_2 p$ est grand devant $2 \times B$, N devient proportionnel à $\frac{7}{4} \times X$ au lieu de $\frac{B}{2} \times X$. Nous retrouvons la conséquence du plus faible temps

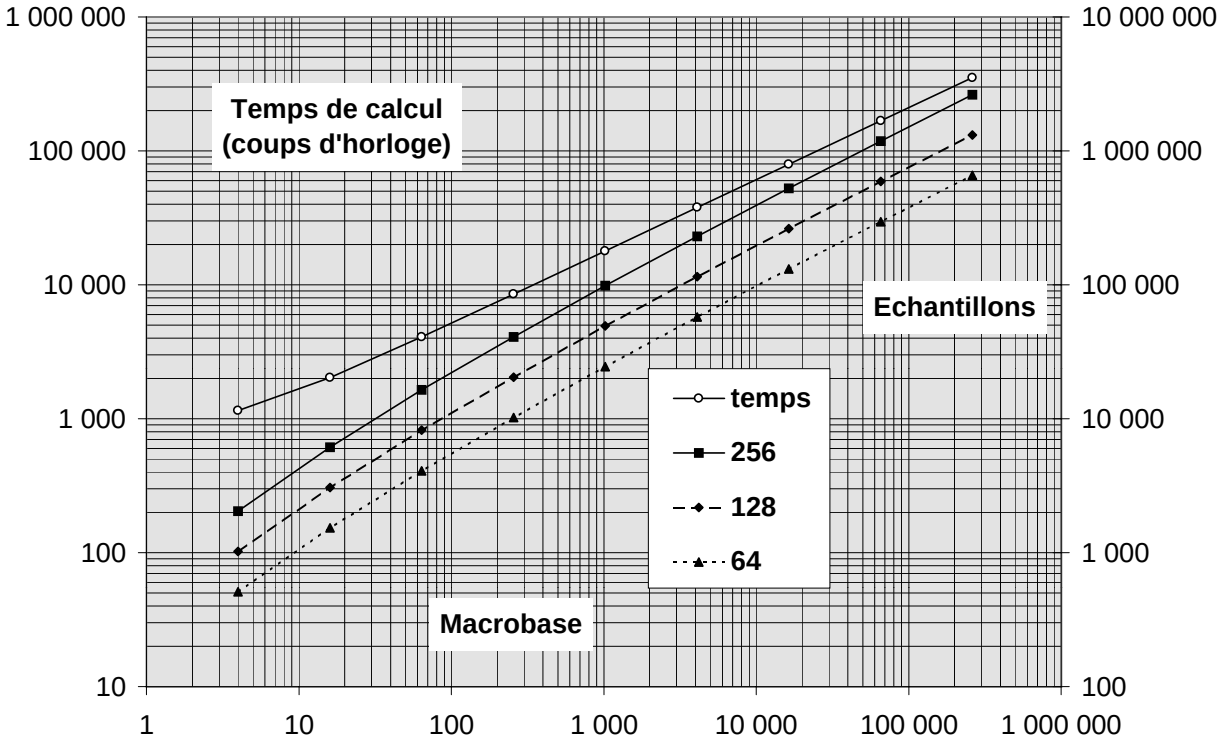

FIG. 1.7 – *temps de calcul en fonction de la macrobase en complément à 2.*

de réponse de cette notation, aggravée du fait qu'un chiffre redondant nécessite deux variables logiques au lieu d'une pour un chiffre binaire en complément à 2. Ce qui impose une plus grande valeur du rapport $\frac{T_c}{T_t}$ pour un même nombre d'échantillons. Nous pouvons visualiser ces résultats à travers les courbes de la figure 1.8 construites sur les mêmes principes que celles de la figure 1.6. Nous avons presque un rapport de 10 entre des courbes similaires pour les deux notations. À fréquences de travail égales, nous avons donc un nombre d'échantillons réduit.

Le temps de calcul T peut lui aussi être estimé. La T.F.R. est constituée de $\log_p N$ étapes de calcul. Ce qui nous donne :

$$T = [\{(B + 7 * \log_2 \sqrt{p}) * T_c\} * 2 * \sqrt{p}] * \log_p N \quad (1.9)$$

$$= \left[\left\{\left(B + \frac{7}{2} * \log_2 p\right) * T_c\right\} * 2 * \sqrt{p}\right] * \log_p N \quad (1.10)$$

$$\iff T = (2 * B + 7 * \log_2 p) * T_c * \sqrt{p} * \log_p N$$

Avec une notation redondante, le comportement de la surface et du temps de calcul des implantations réduites par le changement de dimension $1D \rightarrow 2D$ présente par rapport aux implantations complètes d'une T.F.R. le même phénomène que pour une notation en complément à 2. La figure 1.9 résume ces caractéristiques pour le temps de calcul.

Lorsque $7 * log_2 \gg B$ nous avons :

$$T = 7 * \log_2 p * T_c * \sqrt{p} * \log_p N = 7 * T_c * \sqrt{p} * \log_2 N$$

au lieu de $B * T_c * \sqrt{p} * \log_2 N$. Ce qui nous donne un temps de calcul beaucoup plus court, à nombre d'échantillons égal et pour peu que B soit grand devant 7, et surtout qui tend à devenir indépendant de B, ce qui dissimule la propriété des opérateurs sériels d'avoir un temps de calcul proportionnel à la taille de leurs opérandes.

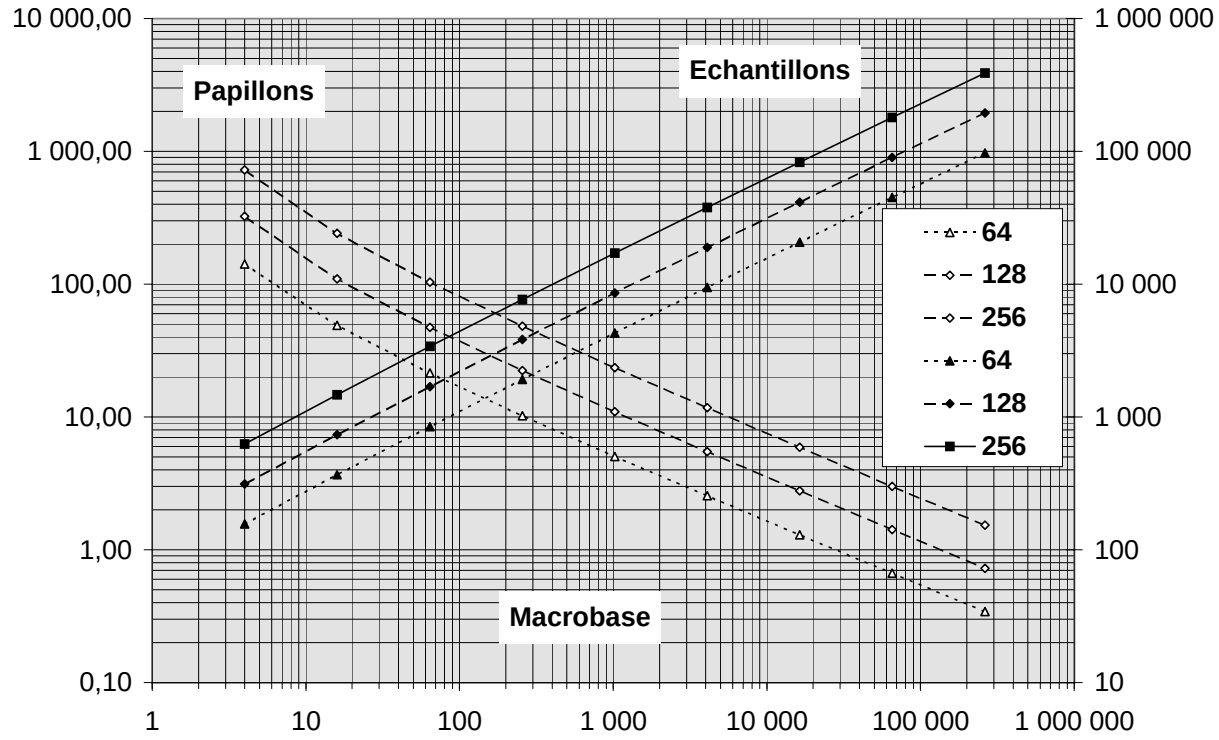

FIG. 1.8 – *nombre de papillons selon la macrobase en notation redondante.*

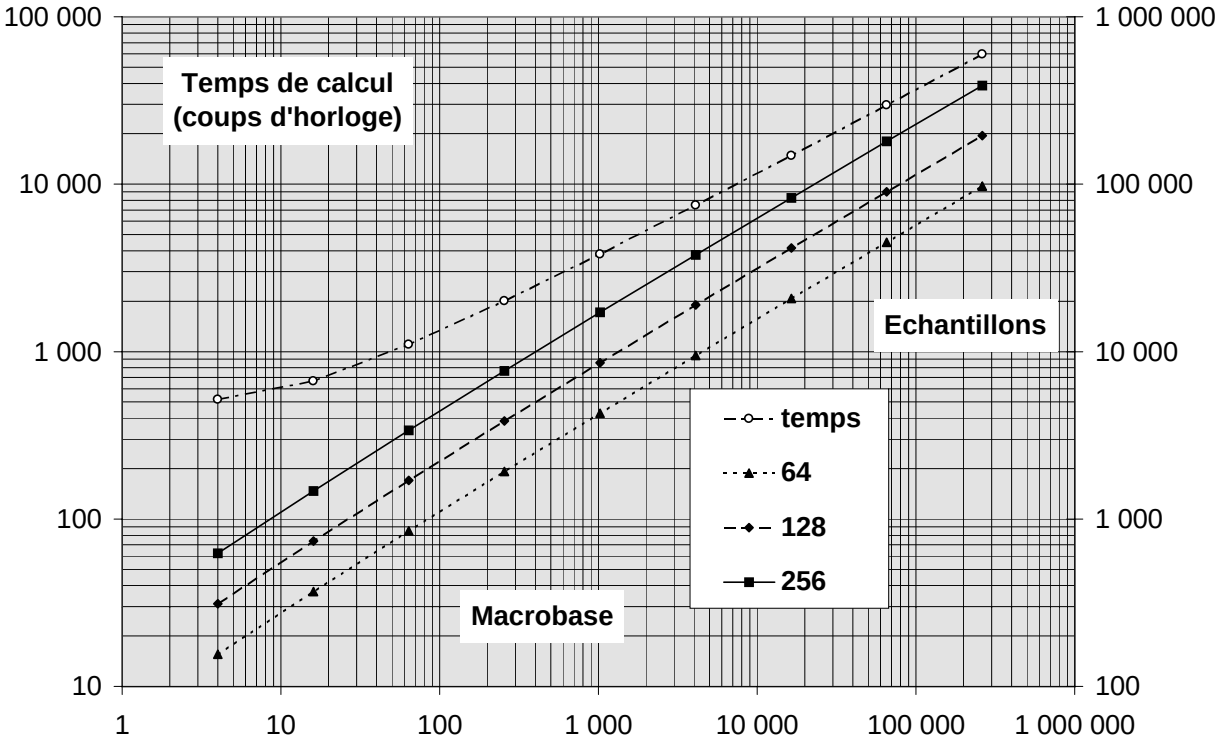

FIG. 1.9 – *temps de calcul en fonction de la macrobase en notation redondante.*

II.1.3.3 Évolutions.

AVEC UNE TECHNOLOGIE $1,2\mu m$ nous avions abouti dans le paragraphe 4.4.3 à la possibilité d'intégrer une architecture de 64 échantillons en redondant avec les simplifications possibles à l'intérieur des papillons arithmétiques. Ce qui doit correspondre d'aprés les résultats du paragraphe 3.5.3 à 256 échantillons en complément à 2. L'utilisation de technologie plus récentes permettrait probablement de gagner un facteur de 2, voire 4, selon la taille de la mémoire nécessaire. Cela permet des macrobases déja importantes, mais la taille de la mémoire risque de devenir pesante, voire limitative, dans l'utilisation de la surface. En effet sa loi de proportionnalité est en p, alors que celle des papillons est en $\sqrt{p}$.

Les notations redondantes telles qu'elles ont été développées jusqu'à présent ne font que gêner ce type d'architectures, surtout par le codage des chiffres. Ce qui se traduit par une surface nécessaire plus grande, donc une valeur plus faible de la macrobase, et un transfert de données à transmettre deux fois plus fort. En ce qui concerne ce dernier point, peut-être serait-il opportun d'étudier si le temps mort entre le calcul et le transfert d'un résultat ne pourrait pas être mis à profit pour recoder le chiffre redondant en complément à 2. Ce dernier n'est qu'un cas particulier du redondant, le chiffre de poids le plus fort valant 0 ou $\Leftrightarrow$1, les autres 0 ou 1. Au prix évidemment d'opérateurs supplémentaires, donc d'une surface nécessaire plus grande.

Une autre piste, qui ferait nous éloigner de la transformation simple $1D \rightarrow 2D$, serait de concevoir des papillons multipuces avec une transformation simple $1D \rightarrow nD$, mais au prix d'une gestion beaucoup plus compliquée de la mémoire. Nous rejoindrions alors les méthodes classiques de partitionnement des transformées de Fourier.

II.1.4 Adaptations.

CE QUI VIENT D'ÊTRE PRÉSENTÉ l'a été pour des bus parallèles et des opérateurs sériels. Pour avoir des coûts et des temps de réalisation plus faibles pour un prototype, il serait tout à fait envisageable d'utiliser des circuits de T.F.R. industriels tels que ceux présentés dans le chapître 2 à base d'opérateurs parallèles et tout ou partie de ces techniques, au moins pour en évaluer partiellement les problèmes et les performances. Cela concerne le rapport $\frac{F_t}{F_c}$, même s'il n'aurait plus tout à fait la même signification, et la transformation $1D \rightarrow 2D$. Nous esquissons dans le tableau 1.1 les principales caractéristiques de solutions minimales de la sorte. Notons que :

- ces circuits peuvent être utilisés pour moins d'échantillons que le nombre nominal, ce qui est nécessaire pour les cas où le nombre d'échantillons traités dans une étape ne correspond pas à ce nombre.

- Au moins certains autorisent de modifier les valeurs des coefficients exponentiels, ce qui est nécessaire pour les transformées tridimensionnelles dont le nombre d'échantillons selon une dimension est inférieur au nombre calculé par le circuit.

- L'architecture générale correspond à une T.F.R. repliée s'il y a plusieurs étapes qui ne sont pas forcément identiques (macrobases mixtes).

- La technique de transformation $1D \rightarrow 2D$ n'est pas toujours nécessaire pour les cas où il en faut deux pour réaliser le calcul. Si elle n'est pas utilisée, cela se traduit par un signe - dans la case correspondante à la valeur $\sqrt{p}$.

Société industrielle	nombre de puces par macrobase	taille des opérandes	$\sqrt{p}_1$	p_1	$\sqrt{p}_2$	p_2
Dassault	1	12 bits	256	$64k$	16	256
Électronique	4	12 bits	$1k$	$1M$	$\Leftrightarrow$	16
Gec Plessey	1	18 bits	$1k$	$1M$	$\Leftrightarrow$	16
CNET	1	16 bits	$4k$	$16M$	$\Leftrightarrow$	$\Leftrightarrow$

TAB. 1.1 – *maquettes de test de diverses techniques utilisées dans les architectures à saturation de bus.*

Nous pouvons constater que :

- la solution la plus précise au niveau des calculs nécessite deux passes et seize puces de T.F.R., elle est aussi la plus souple (programmabilité des coefficients, disponibilité des diverses cellules implantées dans ledit circuit sous forme de bibliothèque auprés du fabricant pour réaliser son propre ASIC par son intermédiaire)

- une solution compacte pourrait être réalisée avec un seul circuit de calcul pour tester les problèmes de transformation $1D \rightarrow 2D$, mais le produit n'a apparemment toujours pas trouvé place dans le catalogue du fondeur de la puce (SGS-Thomson).

Nous pouvons donc en conclure que les travaux que nous avons effectuer et que nous avons présentés dans ce chapître peuvent trouver des développements en utilisant des circuits de T.F.R. déja existant. Ceci peut permettre d'économiser dans un premier temps le temps d'étude pour ce dernier point et ainsi proposer ce genre de solution pour des applications pratiques.

Chapitre 2

Opérateurs sériels de taille variable

Sommaire

II.2.1 Introduction. **87**
II.2.2 Principes de base. **88**
II.2.2.1Généralités. 88
II.2.2.2Opérations arithmétiques et opérandes. 88
II.2.2.3Quantifications des erreurs. 89
II.2.2.4Intégration des erreurs. 90
II.2.3 Les voies traditionnelles de calcul. **91**
II.2.3.1Entrelacement temporel sans débordement par addition. 91
II.2.3.2Entrelacement temporel avec division systématique par 2. 91
II.2.3.3Entrelacement fréquentiel sans débordement par addition. 92
II.2.3.4Entrelacement fréquentiel avec division systématique par 2. 92
II.2.3.5Remarques. 92
II.2.4 Proposition. **95**
II.2.4.1Entrelacement temporel. 95
II.2.4.2Entrelacement fréquentiel. 96
II.2.4.3Comparaison. 96
II.2.5 Architecture à base d'opérateurs à taille variable. **97**
II.2.5.1Généralités. 97
II.2.5.2Surface d'une implantation repliée. 97
II.2.5.3Surface d'une implantation étalée. 98
II.2.5.4Performance en temps 100
II.2.6 Applications de la croissance par pas. **100**

II.2.1 Introduction.

L'a priori d'un travail sur des relatifs de 32 chiffres binaires peut être mis en question en raison des problèmes de surface d'implantation qui apparaissent comme trés critiques dans les implantations cablées. Encore faut-il se poser la question de la conséquence d'une diminution de celle-ci sur la précision des calculs. Dans le paragraphe 2.2, nous rappelons les principes de

base qui ont été dégagé par nos glorieux et nombreux prédécesseurs. Ce qui nous amène à préciser dans le paragraphe 2.3 les résultats qu'ils avaient obtenus pour les différents entrelacements et méthodes de calcul et à modifier certains points de vue qu'ils avaient adoptés et que nous ne partageons pas. Cela nous permet de proposer dans le paragraphe 2.4 une voie qui autorise un compromis entre la précision des calculs et l'un des critères gênants, la vitesse de calcul ou la surface d'implantation. L'architecture qui en découle est étudiée dans le paragraphe 2.5 avec ses caractéristiques. Ce qui nous conduit dans le paragraphe 2.6 de définir les règles d'emploi de cet espoir.

II.2.2 Principes de base.

II.2.2.1 Généralités.

La précision d'un calcul de T.F.R. dépend à la fois de celle des échantillons et de celle du calcul lui-même. Seul ce dernier point fait l'objet de notre étude qui se limite à la base 2. Rappelons les formules mathématiques de base qui apparaissent dans le calcul de chaque papillon : $y = x \pm x' \times \exp aj$ (entrelacement temporel), $y = x + x'$ et $y = (x \Leftrightarrow x') \times \exp aj$ (entrelacement fréquentiel).

L'erreur créée par les différents calculs élémentaires de la T.F.R. a deux origines :

- l'imprécision des coefficients exponentiels qui, sauf valeur particulière, ne sont pas des relatifs, mais des réels dont seule la partie la plus significative est traitée aprés arrondi,

- la troncature ou l'arrondi du résultat d'une opération arithmétique pour en limiter l'augmentation de taille par rapport aux entrées.

II.2.2.2 Opérations arithmétiques et opérandes.

Lorsque des nombres à virgule flottante ne sont pas utilisés , le calcul est réalisé traditionnellement avec des nombres à taille fixe. Une addition ou une multiplication, pour se restreindre au cas des opérations intervenant dans le calcul d'une T.F.R., fournit un résultat dont la taille est une fonction de celle des entrées. Dans le pire des cas et par rapport aux entrées, le résultat d'une addition a une taille augmentée d'un chiffre binaire et celui d'une multiplication a une taille égale à la somme de celles des opérandes d'entrée. Comme chaque calcul élémentaire fait intervenir plusieurs de ces opérations et qu'une T.F.R. est une succession de calculs élémentaires, il est impensable de conserver l'ensemble de ces accroissements de taille. En ce qui concerne les multiplications intervenant dans la T.F.R., elles portent sur des termes dont l'un est un sinus ou un cosinus. Par conséquent, la moitié la plus significative du résultat a le même poids que l'entrée autre que sinusoïdale. La solution la plus commune est de conserver celle-ci et ainsi maintenir à travers une multiplication le même poids aux nombres, hors coefficients exponentiels. Un éventuel débordement dû aux différentes additions est évité grâce à l'emploi d'une des techniques suivantes :

- utilisation de nombres suffisamment grands, nous parlerons d'une *solution sans débordement par addition*,

- division systématique par 2 des résultats intermédiaires, ce qui se réalise par un simple décalage vers la droite du nombre binaire considéré, à la condition que le module du nombre complexe correspondant soit inférieur à la moitié de la pleine échelle, nous parlerons d'une *solution avec division systématique par* 2,

– utilisation de nombres à *virgule glissante* . Il s'agit d'une amélioration de la technique précédente. Elle se rapproche de la technique des nombres flottants et est souvent appelée virgule floffante par bloc, car le cadrage de la mantisse est fait pour l'ensemble des résultats et non au niveau de chaque nombre. En conséquence l'exposant est une valeur unique pour cet ensemble. Ce qui la distingue des nombres flottants est le fait que le déplacement de la virgule ne se fait que dans un seul sens, il n'y a pas de retour en arrière. C'est pour cela que nous préferrons le terme de virgule glissante.

Le choix de la méthode de calcul est fait selon les moyens qui peuvent être mis en oeuvre et la précision demandée. Une solution matérielle est plus rapide qu'une solution logicielle, mais elle impose des nombres d'une taille plus limitée en raison de la surface nécessaire. Une solution à base d'opérateurs avec division par 2, systématique ou selon la nécessité, des résultats intermédiaires est la plus aisée à implanter. L'inconvénient est que cette division se traduit par la disparition du chiffre binaire de plus faible poids. Une division à la demande permet de limiter les pertes d'informations, donc l'erreur, mais au prix d'un surcoût en terme de complexité de gestion du système et donc aussi de surface d'implantation. Avec les ordinateurs actuels, la différence de temps de calcul entre des instructions arithmétiques portant sur des nombres courts et des nombres longs est faible, voire nulle suivant le processeur. L'élimination de décalages ou de tests dans un programme apporte un gain important en vitesse de calcul. Une solution basée sur des nombres suffisamment grands pour éviter un dépassement dû à une addition est donc la plus indiquée, du moins tant que le nombre d'échantillons et leur taille ont des valeurs courantes dans ce genre de calcul, typiquement quelques milliers d'échantillons codés sur seize à vingt chiffres binaires.

II.2.2.3 Quantifications des erreurs.

WEINSTEIN A DÉMONTRÉ que les effets dû à l'imprécision des coefficients est négligeable, comparée à celle dûe à la troncature ou à l'arrondi des resultats[Wei69]. Par conséquent, nous ne retiendrons que ces dernières dans notre étude et nous les ramenons comme nos prédécesseurs à un problème de bruit [Wel69] [D. 75]. Comme il a été démontré par ceux-ci qu'une troncature entraîne une erreur beaucoup plus forte qu'un arrondi, nous choisissons de ne traiter que le cas de l'arrondi des résultats et d'adapter les opérateurs en conséquence.

Concernant les autres sources d'errreur, nous pouvons résumer les principaux résultats des différents travaux qui ont précédés les notres, avec quelques modifications personnelles :

– en supposant que nous conservons la partie la plus significative du résultat de la multiplication dont les opérandes ont une taille de b chiffres sans compter le signe, la puissance de bruit d'arrondi peut être estimée à $\frac{2^{-2b}}{12}$ [G. 71].

– Dans les différentes versions de la décomposition d'une T.F.R., chaque résultat de la multiplication est en fait ajouté au résultat d'une autre multiplication qui sont au nombre de quatre dans un papillon. Si l'arrondi est réalisé au niveau de chaque multiplieur, l'erreur totale d'un papillon est la somme de quatre erreurs élémentaires. S'il est effectué aprés l'addition qui les suit, l'erreur correspondante est la somme de seulement deux erreurs élémentaires. Ce qui est le cas de notre réalisation présentée dans le chapître 4. Une solution logicielle optimisée en vitesse est un cas typique du premier cas de figure et une implantation matérielle optimisée au point de vue erreur celui du second. Pour unifier les resultats et caractériser cette difference, le paramètre α a été introduit. Il est égal à 4 dans le premier

cas et à 2 dans le second. L'erreur totale d'un papillon peut donc être mise sous la forme $\alpha\frac{2^{-2b}}{12}$.

- Un décalage vers la droite d'un chiffre binaire accompagné de l'élimination de celui de plus faible poids entraine l'ajout d'une erreur de puissance égale à $\frac{2^{-2b}}{4}$ aux erreurs précédentes dont la puissance est divisée par 4.

- Quelque soit l'algorithme, un papillon se termine avec une addition sur chaque sortie, sachant qu'il y a quatre sorties distinctes. Si l'architecture du papillon est telle que le décalage est réalisée sur la sortie de l'additionneur, il y a une seule source d'erreur par addition finale du papillon. Si le décalage considéré a lieu sur les grandeurs qui constituent les entrées de l'additionneur final, il existe deux sources d'erreurs par papillon. Le premier cas est typiquement celui d'une implantation matérielle optimisée en précision, le second celui d'une solution logicielle. Pour les mêmes raisons que ci-dessus, nous introduisons le paramètre β qui est égal à 4 dans le premier cas et à 8 dans le second. L'erreur résultante est ainsi égale à $\beta\frac{2^{-2b}}{4}$.

Les valeurs des coefficients α et β sont résumés dans le tableau 2.1 pour les différentes méthodes de calcul avec les critères qui les imposent en fonction de ces dernières.

solution	α	β	optimisation
logicielle	4	8	vitesse
cablée	2	4	précision

TAB. 2.1 – *coefficients d'erreur imposées par les moyens de calcul choisis.*

II.2.2.4 Intégration des erreurs.

NOTRE TRAVAIL EST BASÉ sur les différentes lois décrites par TRÂŃ-THÔŃG et BEDE-LIU [TTL76] que nous rappelons ici en précisant qu'il s'agit d'une T.F.R. de N échantillons, donc constituée de $M = \log_2 N$ étapes successives de calcul :

- une erreur introduite par un papillon dans le m^{eme} pas de la T.F.R. entraine l'apparition d'une erreur de même variance dans 2^{M-m} points au niveau de la sortie de la T.F.R.

- la variance des erreurs de chaque coefficient résultant de la T.F.R. est égale à la somme de la variance des erreurs crées par les étapes successives, le m^{eme} pas introduisant à son propre niveau 2^{M-m} erreurs

- dans l'algorithme à décomposition en temps, une relation de type T.F.D. de 2^m points existe entre un bloc d'éléments d'un pas de rang m et les éléments correspondants au niveau de l'entrée (se reporter au schéma de la figure 2.3)

- dans l'algorithme à décomposition en fréquence une relation de type T.F.D. de 2^{M-m} points existe entre un bloc d'éléments d'un pas de rang m et les éléments correspondants au niveau de la sortie (voir le schéma de la figure 2.5).

Dans tous les cas nous sommons les erreurs de tous les coefficients de sortie et la forme générale est :

$$\sum_{\text{coefficients}} \text{erreurs} = \sum_{\text{pas}} v.r.e.c$$

où :

$$\begin{cases} v &= \text{valeur de l'erreur} \\ r &= \text{poids relatif de l'erreur} \\ e &= \text{nombre d'erreurs apparaissant au cours d'une étape} \\ c &= \text{nombre de coefficients atteint par une erreur} \end{cases}$$

Pour un entrelacement temporel, nous pouvons remarquer que pour les deux premiers pas de la T.F.R. $\omega^{n.k} \in \{\Leftrightarrow 1, 1, \Leftrightarrow j, j\}$. Les multiplications correspondantes peuvent être changée par des opérations plus simples et n'apportant pas d'erreur comme « ne rien faire », « inversion du signe », « transformation d'un nombre réel en un imaginaire pur » et l'inverse de cette dernière. Ainsi la somme des erreurs causées par les multiplications est restreinte aux étapes dont l'indice varie de $m = 3$ à M plutôt que de commencer à $m = 1$. Ce phénomène se retrouve dans les différentes étapes du calcul avec une importance décroissante au fur et à mesure de l'avancement de celui-ci.

D'une façon similaire, nous pouvons remarquer pour un entrelacement fréquentiel que nous avons pendant les deux derniers pas de la T.F.R. $\omega^{n.k} \in \{\Leftrightarrow 1, 1, \Leftrightarrow j, j\}$. Les multiplications correspondantes peuvent être échangées contre des operations plus simples n'introduisant pas d'erreur comme « ne rien faire », « inversion du signe », « transformation d'un nombre réel en un nombre imaginaire » ou l'inverse de cette dernière opération. Ainsi la somme des erreurs dues aux multiplieurs des étapes successives est faîte de $m = 1$ à $M \Leftrightarrow 2$ au lieu de $m = 1$ à $m = M$.

II.2.3 Les voies traditionnelles de calcul.

II.2.3.1 Entrelacement temporel sans débordement par addition.

L'ERREUR SUR CHAQUE COEFFICIENT DE SORTIE de la T.F.R. est notée e_{tpda}. Nous pouvons remarquer que chaque paramètre ou variable a le même poids tout au long des étapes successives, donc $r = 1$. Par application de la définition de cette technique, les seules sources d'erreur sont les multiplieurs.

$$\begin{aligned} \sum_{i=0}^{N-1} e_{tpda} &= \sum_{m=3}^{M} \underbrace{\frac{\alpha.2^{-2B}}{12}}_{v} . \underbrace{\left(N \Leftrightarrow 4.2^{M-m}\right)}_{e} . \underbrace{2^{M-m}}_{c} \qquad (2.1) \\ \Leftrightarrow \sum_{i=0}^{N-1} e_{tpda} &= \frac{\alpha 2^{-2B}}{12} \sum_{m=3}^{M} \left(N \Leftrightarrow 4.2^{M}.2^{-m}\right) 2^{M}.2^{-m} \\ \Leftrightarrow \sum_{i=0}^{N-1} e_{tpda} &= \frac{\alpha 2^{-2B}}{12} \sum_{m=3}^{M} N^2 \left(1 \Leftrightarrow 4.2^{-m}\right) 2^{-m} \\ \Leftrightarrow \sum_{i=0}^{N-1} e_{tpda} &= \frac{\alpha 2^{-2B}}{12} \left(\frac{N^2}{6} \Leftrightarrow N + \frac{4}{3}\right) \qquad (2.2) \end{aligned}$$

II.2.3.2 Entrelacement temporel avec division systématique par 2.

NOTONS e_{dsd} l'erreur sur chaque coefficient de sortie de la T.F.R. et remarquons que diviser un résultat intermédiaire par 2 avant de le transmettre à l'étape suivante est équivalent à accorder au résultat de cette dernière un poids double à celui de la précédente. Selon la définition

de cette méthode, les erreurs sont engendrées aussi bien par les multiplieurs que par les décaleurs réalisant les divisions.

$$\sum_{i=0}^{N-1} e_{tdsd} = \underbrace{\sum_{m=3}^{M} \underbrace{\frac{\alpha 2^{-2B}}{12}}_{v} \underbrace{2^{2m}}_{r} \underbrace{\left(N \Leftrightarrow 4.2^{M-m}\right)}_{e} \underbrace{2^{M-m}}_{c}}_{\text{erreurs dues aux multiplications}} + \underbrace{\sum_{m=1}^{M} \underbrace{\frac{\beta 2^{-2B}}{4}}_{v} \underbrace{2^{2m}}_{r} \underbrace{N}_{e} \underbrace{2^{M-m}}_{c}}_{\text{erreurs dûes aux décalages}} \quad (2.3)$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tdsd} = \sum_{m=3}^{M} \frac{\alpha 2^{-2B}}{12} 2^{2m} \left(N \Leftrightarrow 4.2^{M}.2^{-m}\right).2^{M}.2^{-m} + \sum_{m=1}^{M} \frac{\beta 2^{-2B}}{4} 2^{2m}.N.2^{M}.2^{-m}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tdsd} = \sum_{m=3}^{M} \frac{\alpha.2^{-2B}}{12} 2^{2m}.N^{2}.\left(1 \Leftrightarrow 4.2^{-m}\right).2^{-m} + \sum_{m=1}^{M} \frac{\beta.2^{-2B}}{4} 2^{m}.N^{2}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tdsd} = 2^{-2B}\left[\left(\frac{\alpha}{6}+\frac{\beta}{2}\right)N^{3} \Leftrightarrow \left(\frac{\alpha}{3}M+\frac{\beta}{2}\right)N^{2}\right] \quad (2.4)$$

II.2.3.3 Entrelacement fréquentiel sans débordement par addition.

L'ERREUR SUR CHAQUE COEFFICIENT DE SORTIE est notée e_{fpda}. Chaque paramètre ou variable a le même poids tout au long des étapes successives du calcul, donc $r = 1$.

$$\sum_{i=0}^{N-1} e_{fpda} = \sum_{m=1}^{M-2} \underbrace{\frac{\alpha.2^{-2b}}{12}}_{v} \underbrace{\left(\frac{N}{2} \Leftrightarrow 2^{m}\right)}_{e} \underbrace{2^{M-m}}_{c}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{fpda} = \frac{\alpha.2^{-2b}}{12}.\left(\frac{N}{2} \Leftrightarrow M\right).N \quad (2.5)$$

II.2.3.4 Entrelacement fréquentiel avec division systématique par 2.

NOTONS e_{fdsd} l'erreur sur chaque coefficient de sortie de la T.F.R. au sujet de laquelle nous pouvons faire les mêmes remarques que pour l'entrelacement temporel.

$$\sum_{i=0}^{N-1} e_{fdsd} = \underbrace{\sum_{m=1}^{M-2} \underbrace{\frac{\alpha.2^{-2b}}{12}}_{v} \cdot \underbrace{2^{2m}}_{r} \cdot \underbrace{\left(\frac{N}{2} \Leftrightarrow 2^{m}\right)}_{e} \cdot \underbrace{2^{M-m}}_{c}}_{\text{erreurs dûes aux multiplications}} + \underbrace{\sum_{m=1}^{M} \cdot \underbrace{\frac{\beta.2^{-2b}}{4}}_{v} \cdot \underbrace{2^{2m}}_{r} \cdot \underbrace{2^{M-m}}_{n}}_{\text{erreurs dûes aux décalages}} \quad (2.6)$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{fdsd} = \frac{\alpha.2^{-2b}}{12}.2^{M}.\sum_{m=1}^{M-2} 2^{2m}.\left(\frac{N}{2} \Leftrightarrow 2^{m}\right).2^{-m} + \frac{\beta.2^{-2b}}{4}.2^{M}.\sum_{m=1}^{M} 2^{2m}.2^{-m}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{fdsd} = \frac{\alpha.2^{-2b}}{12}.N.\sum_{m=1}^{M-2} 2^{m}.\left(\frac{N}{2} \Leftrightarrow 2^{m}\right) + \frac{\beta.2^{-2b}}{4}.N.\sum_{m=1}^{M} 2^{m}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{fdsd} = 2^{-2b}\left[\left(\frac{\alpha}{72}+\frac{\beta}{2}\right).N^{3} \Leftrightarrow (\frac{\alpha}{12}+\frac{\beta}{2}).N^{2}+\frac{\alpha}{9}.N\right] \quad (2.7)$$

II.2.3.5 Remarques.

NOUS POUVONS RÉSUMER les principales propriétés de ces différentes solutions par le tableau 2.2. Nous retrouvons les effets de la remarque du paragraphe 2.3.1 :

– l'entrelacement fréquentiel a un nombre de coefficients exponentiels non simplifiables décroissant avec le rang de l'étape de calcul. Donc les sources d'erreur diffusent leurs effets vers un grand nombre de résultats finaux, sachant que dans le cas d'une division systématique par 2 leurs influences sont fortement déduites par les divisions successives.

– L'entrelacement temporel a un nombre de coefficients exponentiels non simplifiables croissant avec le rang de l'étape de calcul. Donc les sources d'erreur diffusent leurs effets vers un nombre moins grand de résultats finaux que s'il s'agissait d'un entrelacement fréquentiel. Par contre, dans le cas d'une division systématique par 2, leurs influences sont beaucoup moins atténuées par les divisions successives.

type de calcul	entrelacement	erreur	
		nom	formule
pas de débordement	temporel	e_{tpda}	$\frac{\alpha.2^{-2B}}{12}\left(\frac{N^2}{6} \Leftrightarrow N+\frac{4}{3}\right)$
par addition	fréquentiel	e_{fpda}	$\frac{\alpha.2^{-2B}}{12}.\left(\frac{N}{2} \Leftrightarrow M\right).N$
un décalage	temporel	e_{tdsd}	$2^{-2B}.\left[\left(\frac{\alpha}{6}+\frac{\beta}{2}\right).N^3 \Leftrightarrow \left(\frac{\alpha}{3}.M+\frac{\beta}{2}\right).N^2\right]$
sytématique par pas	fréquentiel	e_{fdsd}	$2^{-2B}\left[\left(\frac{\alpha}{72}+\frac{\beta}{2}\right)N^3 \Leftrightarrow \left(\frac{\alpha}{12}+\frac{\beta}{2}\right)N^2+\frac{\alpha}{9}N\right]$

TAB. 2.2 – *coefficients d'erreur pour les différents entrelacements par les voies traditionnelles de calcul.*

Ce qui amène le résultat suivant :

– pour une solution sans débordement l'entrelacement temporel est le plus précis,

– pour une solution avec division systématique l'entrelacement fréquentiel est préférable.

Signalons les points qui nous séparent de nos illustres prédécesseurs TRÂŃ-THÔŃG et BEDE-LIU, divergences toutes minimales :

– ils n'ont considéré que le cas $\beta = 4$,

– ayant appliqué une méthode légèrement différente, nous aboutissons dans le cas d'une division systématique par 2 à des résultats légèrement différents des leurs que nous rappelons ci-dessous aprés les notres.

  – $2^{-2B}.\left[\left(\frac{\alpha}{6}+2\right).N^3 \Leftrightarrow \left(\frac{\alpha}{3}.M+2\right).N^2\right]$ pour l'entrelacement temporel
    au lieu de $2^{-2B}.\left[\left(\frac{\alpha}{6}+2\right).N^3 \Leftrightarrow \left(\frac{\alpha}{3}.M+2+M\right).N^2\right]$
  – $2^{-2B}\left[\left(\frac{\alpha}{72}+2\right)N^3 \Leftrightarrow \left(\frac{\alpha}{12}+2\right)N^2+\frac{\alpha}{9}N\right]$ pour l'entrelacement fréquentiel
    au lieu de $2^{-2B}\left[\left(\frac{\alpha}{72}+2\right)N^3 \Leftrightarrow \left(\frac{\alpha}{12}+2+M\right)N^2+\frac{\alpha}{9}N\right]$.

Nous devons aborder de plus un point qui n'avait pas été abordé lors des travaux de nos prédécesseurs. À savoir, d'une part la valeur de B pour les solutions sans débordement dû à une addition et d'autre part la valeur de B pour les solutions avec division sytématique par deux. Dans le calcul d'une T.F.R. il y a des débordements dûs à des additions. Le nombre d'étapes du calcul où ils apparaissent dépend de la valeur des échantillons traités. Dans le pire cas hypothétique il y en a un à chaque étape. Comme il ne s'agit que d'un cas d'école et pour

limiter les erreurs de calcul en évitant les décalages inutiles, les implantations réelles mettent en oeuvre une solution avec division par deux en cas de nécessité que nous avons appelé virgule glissante. TRÂŃ-THÔŃG et BEDE-LIU avaient simulés le comportement de l'erreur pour deux cas, un ensemble d'échantillons dont les parties réelles et imaginaires valaient de façon aléatoire ± 1 d'une part et une sinuoïde d'autre part. L'erreur se situait grossièrement dans la partie supérieure de la zône séparant les cas extêmes « pas de dépassement » et « division systématique par deux ». Nous doutons personnellement de la représentativité réelle de ces deux cas face à la réalité de l'emploi d'une transformée de Fourier. Prenons le cas d'une gaussienne $A.\exp(\Leftrightarrow a^2.x^2)$, où a est l'inverse de sa demi-largeur. Sa transformée de Fourier est égale à $\frac{A.\sqrt{\pi}}{a}.\exp(\frac{-y^2}{4a^2})$. La valeur maximale de la transformée, donc le nombre de dépassements, est ainsi proportionnelle à la largeur de l'originelle, ce qui donne une idée de ses variations possibles ! Les données étant inconnues et, à notre connaissance aucun travail n'ayant abouti à prédire le nombre maximum de débordements au cours d'un calcul, il faut se rabattre sur le pire cas. Ce qui nous donne une largeur d'opérandes égale à $B+M$ au lieu de B. La formule d'erreur devient alors :

– pour l'entrelacement temporel

$$\frac{\alpha 2^{-2(B+M)}}{12}\left(\frac{N^2}{6} \Leftrightarrow N + \frac{4}{3}\right) = \frac{\alpha 2^{-2(B)}}{12}\left(\frac{1}{6} \Leftrightarrow \frac{1}{N} + \frac{4}{3N^2}\right)$$

– pour l'entrelacement fréquentiel

$$\frac{\alpha.2^{-2B}}{12}.\left(\frac{N}{2} \Leftrightarrow M\right).N = \frac{\alpha.2^{-2B}}{12}.\left(\frac{1}{2} \Leftrightarrow \frac{M}{N}\right)$$

.

Évidemment cela conduit à des implantations trés coûteuses en surface. Ce qui implique lors de la conception d'un circuit dédié à une application donnée de bien connaître le type de données traitées et de permettre des conceptions taillées sur mesure, voir à ce sujet les remarques du paragraphe 4.2.5.

En raison de la trés rapide décroissance des termes non constants pour des valeurs communes de N, nous indiquons dans le tableau 2.3 pour illustrer ces résultats théoriques les tendances asymptotiques de l'erreur de chaque cas, relativement au meilleur de sa catégorie.

type	référence	entrelacement	calcul	erreur relative
pas de débordement par addition	temporel câblé	temporel	logiciel	2
		fréquentiel	câblé	3
			logiciel	6
un décalage sytématique par pas	fréquentiel câblé	temporel	logiciel	2,3
			câblé	1,15
		fréquentiel	logiciel	2

TAB. 2.3 – *valeurs asymptotiques des coefficients d'erreur pour les différents entrelacements par les voies traditionnelles de calcul.*

Pour pallier à ces problèmes de taille des nombres, il est possible d'aborder le point de vue de l'erreur sous un angle de rapport de signal sur bruit. Encore faut-il avoir une idée des données à traiter, ce qui se résume aux cas pratiques d'application où celles-ci peuvent être caractérisées d'une façon commune, même si cela est assez simpliste. M. Bellanger a développé une approche assez simplifiée de ce genre [Bel96].

II.2.4 Proposition.

PRENDRE DES OPÉRANDES SUFFISAMMENT GRANDS pour éviter des dépassements dûs à des additions entraîne que les chiffres de plus fort poids de ces opérandes ne sont utilisés que progressivement au cours du calcul. C'est pourquoi nous proposons [VA94b] de calculer une T.F.R. avec des nombres à taille croissante pour compenser l'accroissement de taille dûe uniquement aux additions. Considérons le cas d'une *croissance sytématique d'un chiffre par pas* de calcul, ce qui correspond au pire cas semblable à celui décrit ci-dessus pour la solution sans débordement par addition. D'une manière identique à la virgule glissante, nous pouvons étendre notre proposition à une croissance suivant la nécessité, ce qui permet d'optimiser notre proposition en terme de vitesse au prix d'un surcoût de complexité de la commande du système envisagé. Pour ce qui concerne la quantification des problèmes d'erreur, nous retombons alors dans la problématique de la virgule glisante et son impossibilité à la traiter correctement, hormis la description de ses deux cas extrêmes, sans croissance et avec une croissance d'un chiffre par pas.

Notons e_{tcp} l'erreur sur chaque coefficient de sortie de la T.F.R. pour une solution de croissance sytématique d'un chiffre par pas. Nous pouvons remarquer que les seules erreurs sont engendrées par les multiplieurs et que celles-ci ont un poids qui dépend du rang de l'étape au cours de laquelle elles naissent. En supposant que les échantillons originaux aient été cadrés pour occuper toute la dynamique permise par B, l'erreur des premières étapes de calcul ont une importance relative plus forte que celles des dernières rapportées aux résultats finaux.

II.2.4.1 Entrelacement temporel.

$$\sum_{i=0}^{N-1} e_{tcsp} = \sum_{m=3}^{M} \underbrace{\frac{\alpha 2^{-2(B+m)}}{12}}_{v} \underbrace{\left(N \Leftrightarrow 4.2^{M-m}\right)}_{e} \underbrace{2^{M-m}}_{c} \tag{2.8}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tcsp} = \sum_{m=3}^{M} \frac{\alpha.2^{-2B}.2^{-2m}}{12} \left(N \Leftrightarrow 4.2^{M}.2^{-m}\right).2^{M}.2^{-m}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tcsp} = \frac{\alpha.2^{-2B}}{12}.N^2 \sum_{m=3}^{M} 2^{-3m}.\left(1 \Leftrightarrow 4.2^{-m}\right)$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tcsp} = \frac{\alpha 2^{-2B}}{12}\left(\frac{N^2}{840} \Leftrightarrow \frac{1}{7N} + \frac{1}{3,75N^2}\right) \tag{2.9}$$

II.2.4.2 Entrelacement fréquentiel.

$$\begin{aligned}
\sum_{i=0}^{N-1} e_{fcsp} &= \underbrace{\sum_{m=1}^{M-2} \underbrace{\frac{\alpha 2^{-2(B+m)}}{12}}_{v} \underbrace{\left(\frac{N}{2} \Leftrightarrow 2^m\right)}_{e} \underbrace{2^{M-m}}_{c}}_{\text{erreurs dûes aux multiplications}} \\
\Leftrightarrow \sum_{i=0}^{N-1} e_{fcsp} &= \sum_{m=1}^{M-2} \frac{\alpha.2^{-2B}.2^{-2m}}{12} \left(\frac{N}{2} \Leftrightarrow 2^m\right) 2^M.2^{-m} \\
\Leftrightarrow \sum_{i=0}^{N-1} e_{fcsp} &= \frac{\alpha.2^{-2B}}{12}.N \sum_{m=1}^{M-2} 2^{-3m}.\left(\frac{N}{2} \Leftrightarrow 2^m\right) \\
\Leftrightarrow \sum_{i=0}^{N-1} e_{fcsp} &= \frac{\alpha 2^{-2b}}{12}\left(\frac{N^2}{14} \Leftrightarrow \frac{32}{7.N} + \frac{16}{3.N^2} \Leftrightarrow \frac{1}{3}\right) \qquad (2.10)
\end{aligned}$$

II.2.4.3 Comparaison.

Les caractéristiques d'erreur des différentes solutions sont résumées dans le tableau 2.4. Pour le cas sans débordement, nous distinguons les deux cas, celui où B est considéré de façon générale sans nous soucier de sa valeur et désigné sous le terme de brut et celui où nous avons choisi le pire cas par rapport aux données initiales pour le nombre de dépassement.

type	entrelacement	erreur	
		nom	formule
pas de débordement	temporel	e_{tpda}	$\frac{\alpha.2^{-2B}}{12}\left(\frac{N^2}{6} \Leftrightarrow N + \frac{4}{3}\right)$
par addition, brut	fréquentiel	e_{fpda}	$\frac{\alpha.2^{-2B}}{12}.\left(\frac{N}{2} \Leftrightarrow M\right).N$
pas de débordement	temporel	e_{tpda}	$\frac{\alpha 2^{-2(B)}}{12}\left(\frac{1}{6} \Leftrightarrow \frac{1}{N} + \frac{4}{3N^2}\right)$
par addition, pire cas	fréquentiel	e_{fpda}	$\frac{\alpha.2^{-2B}}{12}.\left(\frac{1}{2} \Leftrightarrow \frac{M}{N}\right)$
un décalage	temporel	e_{tdsd}	$2^{-2B}.\left[\left(\frac{\alpha}{6} + \frac{\beta}{2}\right).N^3 \Leftrightarrow \left(\frac{\alpha}{3}.M + \frac{\beta}{2}\right).N^2\right]$
sytématique par pas	fréquentiel	e_{fdsd}	$2^{-2B}\left[\left(\frac{\alpha}{72} + \frac{\beta}{2}\right).N^3 \Leftrightarrow \left(\frac{\alpha}{12} + \frac{\beta}{2}\right).N^2 + \frac{\alpha}{9}.N\right]$
taille croissante	temporel	e_{tcsp}	$\frac{\alpha 2^{-2B}}{12}\left(\frac{N^2}{840} \Leftrightarrow \frac{1}{7N} + \frac{1}{3{,}75.N^2}\right)$
sytématique par pas	fréquentiel	e_{fcsp}	$\frac{\alpha 2^{-2B}}{12}\left(\frac{N^2}{14} \Leftrightarrow \frac{32}{7.N} + \frac{16}{3.N^2} \Leftrightarrow \frac{1}{3}\right)$

Tab. 2.4 – *coefficients d'erreur pour les diverses solutions et entrelacement.*

Nous pouvons constater qu'une solution sans débordement par addition brute présente une erreur beaucoup plus grande que la solution avec une taille croissante par pas. Cela résulte du fait que nous ne tenons pas compte des débordements possibles, donc des chiffres supplémentaires qu'il faut prévoir pour éviter ce fait. Nous avons donc consideré qu'une erreur est égale à $\frac{\alpha 2^{-2B}}{12}$ sans tenir compte de la taille utile des opérandes, c'est à dire la taille totale des opérandes diminuée du nombre de chiffres binaires de plus fort poids n'ayant qu'une valeur nulle ou égale à celle du chiffre de signe dans une notation en complément à deux dans les échantillons originels.

Cela est vrai pour le dernier pas parce qu'alors les deux grandeurs doivent être équivalentes. Pour les précédentes, il faut considérer le fait que la partie utile de l'opérande croît au cours du calcul, ce qui est équivalent au déroulement de la solution à taille croissante. Si nous faisons une étude du second ordre, nous pourrions démontrer qu'une solution sans débordement par addition est plus précise à cause de la taille constante des coefficients exponentiels. Cela exigerait de connaître le nombre de débordements qui auraient lieu au cours des étapes successives si rien n'était prévu pour combattre ce phénomène. Ce qui revient à connaître l'allure des données. Nous ne pouvons que prévoir le pire des cas et nous obtenons les résultats du tableau précité.

Si nous comparons ces résultats théoriques, nous pouvons constater que la solution par croissance systématique par pas autorise une nette amélioration par rapport à une division systématique par deux.

Cette solution de croissance des nombres au cours des calculs permet donc d'implanter des architectures qui peuvent être économiques en temps ou en surface d'implantation tout en permettant une excellente précision. Par rapport à la solution à taille fixe taillée pour éviter des débordements, elle permet d'utiliser au mieux le temps consacré aux calculs ou la surface en ne traitant que les chiffres existants. L'idéal serait de faire croître la taille des nombres quand les besoins se font sentir. Cela demande toutefois une gestion dont la complexité demande à être étudiée pour en estimer l'intérêt. Il semble toutefois que cette gestion serait intéressante lorsque B devient grand. Dans une première étape, il serait bon d'adopter une solution simple, c'est à dire prévoir le pire cas, donc une croissance systématique par pas.

II.2.5 Architecture à base d'opérateurs à taille variable.

II.2.5.1 Généralités.

UNE DES PROPRIÉTÉS DES OPÉRATEURS SÉRIELS est qu'ils permettent des calculs avec des opérandes de taille variable. Certains autorisent une taille quelconque, c'est le cas des additionneurs. D'autres la limitent à une valeur fixée à la conception du circuit, citons les multiplieurs, parce qu'ils utilisent des registres parallèles dont la taille est forcément un paramètre de construction.

Le temps de calcul et la surface d'implantation sont des fonctions de la taille des opérandes. Nous choisissons l'hypothèse correspondant aux opérateurs les plus courants, à savoir qu'une partie est constante et qu'une autre partie est proportionnelle à ce paramètre. Prenons le cas de la surface. La partie constante est celle correspondant aux additionneurs et à la tête des multiplieurs. La partie proportionnelle correspond au corps des multiplieurs.

Pour estimer l'amélioration qu'apporte une architecture à opérandes à taille variable en terme de vitesse et de surface, nous allons établir les facteurs de proportionnalité de ces deux grandeurs pour cette solution [VA95b], aprés avoir rappelé ceux des solutions traditionnelles. Nous unifions les cas de la notation en complément à deux et de l'arithmétique redondante. Pour cela nous introduisons le paramètre γ qui vaut 1 dans le cas de la notation en complément à deux et 0 dans le cas d'une notation redondante. Il représente le chiffre de signe en complément à deux et traduit le fait qu'une notation redondante coûte deux fois plus cher en terme de surface par rapport à une notation binaire traditionnelle.

II.2.5.2 Surface d'une implantation repliée.

PAR APPLICATION des principes énoncés ci-dessus , nous avons donc les coefficients de proportionnalité suivants pour la surface d'implantation d'une architecture repliée :

– division systématique par 2

$$(2 \Leftrightarrow \gamma)(\delta + B + \gamma) \tag{2.11}$$

– pas de débordement par addition

$$(2 \Leftrightarrow \gamma)(\delta + B + M + \gamma) \tag{2.12}$$

Le cas d'une taille croissante par pas est plus complexe. Dans une architecture repliée, les opérateurs doivent pouvoir traiter tous les opérandes successifs, en particulier ceux qui ont la plus grande taille. Il s'agit de ceux de la dernière étape qui sont identiques à ceux du cas sans débordement par addition. Le coefficient de proportionnalité est donc le même que celui donné par la formule 2.12.

II.2.5.3 Surface d'une implantation étalée.

SI NOUS NE TENONS PAS COMPTE des simplifications possibles pour conserver la meilleure régularité du schéma d'implantation d'une solution sans débordement ou avec division systématique par 2, il suffit de multiplier par $M = \log_2 N$ les coefficients d'une architecture repliée pour les obtenir. Pour une implantation à taille croissante, si nous considérons l'étape de calcul de rang m, la taille des opérandes est de la forme $\delta + B + m + \gamma$. Ainsi le coefficient global qui est l'intégrale des coefficients successifs élémentaires est égal à :

$$(2 \Leftrightarrow \gamma) \left[\delta + B + \gamma + \frac{1+M}{2}\right] \log_2 N \tag{2.13}$$

Pour obtenir une surface la plus réduite possible, nous pouvons nous souvenir des valeurs particulières que peuvent prendre les termes exponentiels. Dans le cas de l'algorithme dit à décomposition en temps, les deux premières étapes ne font intervenir que des termes qui entraînent une substitution d'opérateurs simples aux multiplieurs. Les étapes suivantes voient doubler à chaque étape le nombre de papillons qui ne peuvent subir ce traitement. Ce nombre peut être écrit sous la forme $\frac{N \Leftrightarrow 4.2^{M-m}}{2}$ où m est l'indice du pas de calcul. Pour la version de l'algorithme dîte à décomposition en fréquence, le phénomène est inversé et la formule donnant les papillons qui ne peuvent être simplifiés est égale à $\frac{N}{2} \Leftrightarrow 2^m$, mais les résultats globaux sont identiques. Nous ne portons qu'un exemple de calcul, celui de l'entrelacement temporel avec une division systématique par 2. Si les remplaçants simplifiés des multiplieurs, par leur faible taille et bien que leurs nombres ne soient pas constants au cours des étapes successives, sont globalisés avec les autres parties constantes des coefficients, nous obtenons les coefficients suivants :

– division systématique par 2

$$\begin{aligned}
&\sum_{m=3}^{M} (2 \Leftrightarrow \gamma)\,(\delta + B + \gamma)\,\frac{N \Leftrightarrow 4.2^{M-m}}{2} \\
&= (2 \Leftrightarrow \gamma)\,(\delta + B + \gamma)\,.\sum_{m=3}^{M} \frac{N \Leftrightarrow 4.2^{M}.2^{-m}}{2} \\
&= (2 \Leftrightarrow \gamma)\,(\delta + B + \gamma)\left[\sum_{m=3}^{M} \frac{N}{2} \Leftrightarrow \sum_{m=3}^{M} 2.N.2^{-m}\right]
\end{aligned}$$

$$
\begin{aligned}
&= (2 \Leftrightarrow \gamma)\,(\delta + B + \gamma)\left[(M \Leftrightarrow 3+1).\frac{N}{2} \Leftrightarrow 2.N.\sum_{m=0}^{M-3} 2^{-m-3}\right] \\
&= (2 \Leftrightarrow \gamma)\,(\delta + B + \gamma)\,.\left[(M \Leftrightarrow 2).\frac{N}{2} \Leftrightarrow 2.N.2^{-3}.\frac{(2^{-1})^{M-3+1} \Leftrightarrow 1}{2^{-1} \Leftrightarrow 1}\right] \\
&= (2 \Leftrightarrow \gamma)\,.\,(\delta + B + \gamma)\,.\left[(M \Leftrightarrow 2)\,.\frac{N}{2} + 2^{-2}.\frac{2^2 \Leftrightarrow N}{\frac{1}{2}}\right] \\
&= (2 \Leftrightarrow \gamma)\,.\,(\delta + B + \gamma)\,.\left[\frac{N}{2}(M \Leftrightarrow 3) + 2\right] \qquad (2.14)
\end{aligned}
$$

– pas de débordement par addition, nous prenons le pire cas où la taille augmente d'un chiffre binaire par pas de calcul.

$$(2 \Leftrightarrow \gamma)(\delta + B + M + \gamma)\left[\frac{N}{2}(M \Leftrightarrow 3) + 2\right] \qquad (2.15)$$

Dans le cas d'une taille croissante par pas, le coefficient de proportionnalité de chaque étape est de la forme $\delta + b + \gamma$. Ainsi le coefficient global est l'intégrale des coefficients successifs élémentaires et égal à :

– cas d'un algorithme à décomposition en temps

$$
\begin{aligned}
&\sum_{m=3}^{M} (2 \Leftrightarrow \gamma)\,(\delta + b + \gamma)\,\frac{N \Leftrightarrow 4.2^{M-m}}{2} \\
&= \sum_{m=3}^{M} (2 \Leftrightarrow \gamma)\,(\delta + B + m + \gamma)\,\frac{N \Leftrightarrow 4.2^{M-m}}{2} \\
&= (2 \Leftrightarrow \gamma)\,.\left[\sum_{m=3}^{M} (\delta + B + \gamma)\,.\frac{N \Leftrightarrow 4.2^{M}.2^{-m}}{2} + \sum_{m=3}^{M} m.\frac{N \Leftrightarrow 4.2^{M}.2^{-m}}{2}\right] \\
&= (2 \Leftrightarrow \gamma)\,.\left[(\delta + B + \gamma)\,.\sum_{m=0}^{M-3}\frac{N \Leftrightarrow 4.N.2^{-m-3}}{2} + \sum_{m=0}^{M-3}(m+3).\frac{N \Leftrightarrow 4.2^{M}.2^{-m-3}}{2}\right] \\
&= (2 \Leftrightarrow \gamma)\,.\left[(\delta + B + \gamma)\,.\left(\sum_{m=0}^{M-3}\frac{N}{2} \Leftrightarrow 2.N.\sum_{m=0}^{M-3}\frac{2^{-m}.2^{-3}}{2}\right)\right. \\
&\quad \left. + \sum_{m=0}^{M-3} m.\frac{N}{2} + \sum_{m=0}^{M-3} 3.\frac{N}{2} \Leftrightarrow \sum_{m=0}^{M-3} m.\frac{4.N.2^{-m}.2^{-3}}{2} \Leftrightarrow \sum_{m=0}^{M-3} 3.\frac{4.N.2^{-m}.2^{-3}}{2}\right] \\
&= (2 \Leftrightarrow \gamma)\,.\left[(\delta + B + \gamma)\,.\left(N.\frac{M \Leftrightarrow 3}{2} \Leftrightarrow N.2^{-3}.\frac{(2^{-1})^{M-3+1} \Leftrightarrow 1}{2^{-1} \Leftrightarrow 1}\right)\right. \\
&\quad + \frac{N}{2}.\frac{(M \Leftrightarrow 2).(M \Leftrightarrow 3)}{2} + (M \Leftrightarrow 3+1).3.\frac{N}{2} \Leftrightarrow \frac{N}{4}.\frac{\Leftrightarrow (M \Leftrightarrow 3+2).2^{-M+3+1} + 4}{2} \\
&\quad \left. \Leftrightarrow \frac{3.N}{4}.\frac{(2^{-1})^{M-3+1} \Leftrightarrow 1}{2^{-1} \Leftrightarrow 1}\right] \\
&= \frac{N}{2}(2 \Leftrightarrow \gamma)\left[\frac{M+1}{2} + (\delta + B + \gamma)(M \Leftrightarrow 3) \Leftrightarrow 10\right] \\
&\quad + (2 \Leftrightarrow \gamma)(\delta + B + \gamma) + 2(M+2) \qquad (2.16)
\end{aligned}
$$

– cas d'un algorithme à décomposition en fréquence

$$\begin{aligned}
&\sum_{m=1}^{M-2} (2 \Leftrightarrow\gamma) \,.\, (\delta + b + \gamma) \,.\, \left(\frac{N}{2} \Leftrightarrow 2^m\right) \\
&= \sum_{m=1}^{M-2} (2 \Leftrightarrow\gamma) \,.\, (\delta + B + m + \gamma) \,.\, \left(\frac{N}{2} \Leftrightarrow 2^m\right) \\
&= (2 \Leftrightarrow\gamma) \left[(\delta + B + \gamma) \left(\sum_{m=1}^{M-2} \frac{N}{2} \Leftrightarrow \sum_{m=0}^{M-3} 2^{m+1}\right) + \frac{N}{2}.\sum_{m=1}^{M-2} m \Leftrightarrow \sum_{m=1}^{M-2} m.2^m\right] \\
&= (2 \Leftrightarrow\gamma) \left[(\delta + B + \gamma) \left((M \Leftrightarrow 2 + 1).\frac{N}{2} \Leftrightarrow 2.\frac{2^{M-3} \Leftrightarrow 1}{2 \Leftrightarrow 1}\right)\right. \\
&\quad \left. + \frac{N}{2}.\frac{(M \Leftrightarrow 2 + 1).(M \Leftrightarrow 2)}{2} \Leftrightarrow (M \Leftrightarrow 2 \Leftrightarrow 1).2^{M-2+1} \Leftrightarrow 2\right] \\
&= \frac{N}{2}(2 \Leftrightarrow\gamma) \left[\frac{(M+1)(M \Leftrightarrow 2)}{2} + (\delta + B + \gamma)(M \Leftrightarrow 2) \Leftrightarrow (M \Leftrightarrow 3)\right] \\
&\quad + 2(2 \Leftrightarrow\gamma)(\delta + B + \gamma \Leftrightarrow 1)
\end{aligned} \tag{2.17}$$

II.2.5.4 Performance en temps

Nous reprenons les mêmes notations et introduisons le terme ϵ qui représente l'éventuelle propagation d'une retenue. Il dépend des opérateurs et de la technologie d'implantation pour la notation en complément à deux et est égal à zéro pour une notation redondante. Nous obtenons dans ces conditions :

– division systématique par 2

$$(1 + \gamma) \left[\delta + (B + \gamma)(1 + \epsilon)\right] * \log_2 N \tag{2.18}$$

– pas de débordement par addition, avec toujours le pire cas où la taille augmente d'un chiffre binaire par pas de calcul.

$$(1 + \gamma) \left[(\delta + (B + \gamma + M)\,(1 + \epsilon)\right] * \log_2 N \tag{2.19}$$

– taille croissante par pas où le coefficient de proportionnalité est $\delta + (b + \gamma)(1 + \epsilon)$ qui doit être intégré pour les étapes successives

$$(1 + \gamma) \left[\delta + \left(B + \gamma + \frac{1 + \log_2 N}{2}\right)(1 + \epsilon)\right] \log_2 N \tag{2.20}$$

II.2.6 Applications de la croissance par pas.

Si le phénomène d'erreur est une constante dans le cas d'architecture à croissance de la taille des opérandes à chaque étape du calcul, nous ne pouvons espérer bénéficier des deux propriétés de temps de calcul et de surface d'implantation tels que nous les avons évalués. Soit nous choisissons l'une, soit nous choisissons l'autre.

Dans une architecture repliée, nous bénéficions du gain en temps et utilisons progressivement au cours des étapes successives la surface consacrée aux opérateurs cablés. Le temps de calcul est optimisé, car les opérateurs trvaillent au débit maximal théorique.

Pour une implantation étalée les résultats d'une étape ne peuvent être transmis à l'étage suivant si celui-ci n'a pas terminé ses calculs précédents. Comme ce dernier traite des opérandes plus grands, il a besoin de plus de temps pour effectuer ses calculs, donc ses prédécesseurs doivent attendre d'être libéré de leurs résultats pour commencer de nouveaux calculs sans être engorgé de données en attente d'évacuation vers les étages ultérieurs. Les opérateurs présentent donc des temps morts qui diminuent au fur et à mesure que le rang de l'étage auquel ils appartiennent augmente. Pour une architecture étalée, c'est donc le taux d'utilisation du temps qui augmente au fil des étapes. La surface est optimisée, car tous les opérateurs travaillent en permanence si la structure considérée de la T.F.R. est alimentée avec suffisamment de données.

Chapitre 3

Opérateurs arithmétiques parallèles de taille variable

Sommaire

II.3.1 Introduction. 103
II.3.2 Proposition. 104
II.3.3 Amélioration du temps de calcul. 104
II.3.3.1Généralités. 104
II.3.3.2Opérateurs à coefficients de proportionnalité linéaires. 105
II.3.3.3Opérateurs à coefficients de proportionnalité logarithmiques. 105
II.3.4 Amélioration de la surface d'implantation. 106
II.3.4.1Généralités. 106
II.3.4.2Opérateurs à coefficients de proportionnalité linéaires. 106
II.3.4.3Opérateurs à coefficients de proportionnalité logarithmiques. 108
II.3.5 Perspectives. 109
II.3.5.1Généralités. 109

II.3.1 Introduction.

Le calcul réellement rapide d'une T.F.R. sur un grand nombre d'échantillons nécessite l'usage d'architectures dédiées basées sur des opérateurs cablés optimisés aussi bien pour la vitesse que la précision. Cette notion de rapidité est difficile à quantifier en raison de sa nature psychologique. Un utilisateur de T.F.R. souhaite obtenir un résultat le plus rapidement possible, avec la plus grande précision possible. Hormis le cas de problèmes ayant des contraintes de temps réel, domaine militaire essentiellement, le temps acceptable dépend de la patience de l'utilisateur $\cdots$ et de la possibilité de réduire la précision de ses calculs ou d'utiliser une méthode qui lui permette d'atteindre son objectif en se passant de ces calculs. Nous avons vu dans le chapître 2 que des opérateurs sériels adéquats permettaient d'atteindre cet objectif. Toutefois l'industrie a jusqu'à présent préféré des solutions plus économes en surface. La distribution des calculs entre quelques papillons réalisés à partir d'opérateurs parallèles est la solution à la mode. Les technologies disponibles ont longtemps justifié ces choix et, actuellement encore, une solution monopuce de quelques milliers d'échantillons reste encore un terrain privilégié pour cette stratégie. C'est pourquoi nous étendrons notre proposition d'opérateurs à taille variable

aux opérateurs parallèles dans le paragraphe 3.2. La variété de ceux-ci nous amène à évaluer dans le paragraphe 3.3 les caractéristiques qui en découlent en terme de temps de calcul. Les améliorations apportées par rapport aux solutions classiques ont un coût en surface que nous estimons dans le paragraphe 3.4. Ce qui nous amène dans le paragraphe 3.5 à envisager les règles d'emploi de tels opérateurs.

II.3.2 Proposition.

Opérateurs parallèles ou sériels présentent les mêmes phénomènes d'erreurs. Les lois concernant les sources, la propagation et les effets des erreurs sont identiques pour un type de fonctionnement donné des opérateurs, calcul sans débordement, avec division systématique par 2 ou avec des opérandes croissant en taille à chaque pas. Les mêmes causes produisant les mêmes effets, les estimations d'erreur faîtes dans le chapître traitant des opérateurs sériels restent valables. L'évaluation des améliorations apportées pour le temps de calcul par l'utilisation d'opérateurs parallèles demande à être refaîte, les lois gérant ce phénomène étant différentes. Il en est de même pour le coût en surface à supporter pour bénéficier de ces avantages.

Les opérateurs sériels imposent par principe un temps de calcul proportionnel à la taille des opérandes B , quelque soit la notation arithmétique employée. Les opérateurs parallèles offrent une palette plus large de possibilités [TYY86]. Cela permet, par exemple, d'utiliser des opérateurs ayant un temps de calcul proportionnel au logarithme de B, d'où un gain en vitesse de travail.

Le tableau 3.1 rappelle les coefficients de proportionnalité pour le temps de calcul des additionneurs des multiplieurs. Les solutions utilisant des opérateurs parallèles avec un coefficient

type d'opérations	opérateurs parallèles				opérateurs sériels
	naïf	usuel		optimisé	
$+$	b	$log_2 b$	$\sqrt[n]{b}$	1	b
$\times$	b^2	b		$log_2 b$	b

Tab. 3.1 – *temps de calcul de différents types d'opérateurs.*

égal à b^2 n'apportent rien par rapport à des solutions sérielles qui ont par principe un coefficient égal à b. Celles utilisant un coefficient égal à $\sqrt[n]{b}$ sont employées à cause de la difficulté de tester celles en $log_2 b$ et ne sont qu'un moyen d'approcher l'efficacité de ces dernières, le coefficient n dépend de l'architecture de l'implantation. Nous nous bornons dans notre étude à celles-ci, sachant que les résultats pour sa famille approchante en sont plus ou moins approchés. Nous ne prenons donc en considération que des solutions ayant un coefficient b, l'augmentation de performances est seulement dûe au parallélisme interne des opérateurs, ou $log_2 b$, la solution la plus intéressante en terme de vitesse.

II.3.3 Amélioration du temps de calcul.

II.3.3.1 Généralités.

Pour estimer les performances en terme de vitesse de calcul, nous devons intégrer les coefficients de proportionnalité pour les étapes successives de la transformée de Fourier. Soit une T.F.R. de $N = 2^M$ échantillons codés sur B chiffres binaires. La comparaison doit être faîte avec une solution basée sur des opérateurs parallèles d'une taille constante qui évite des

dépassements dûs à des additions. Dans le pire des cas, cela nécessite des opérandes de $B + M$ chiffres binaires.

Les hypothèses de travail pour cette étude sur le temps de calcul portent sur une transformée de Fourier implantée avec une architecture repliée, une architecture étalée étant trop irréaliste et ce même avec de technologies envisageables à moyen terme.

II.3.3.2 Opérateurs à coefficients de proportionnalité linéaires.

NOUS DEVONS INTÉGRER les coefficients caractéristiques le long des étapes pour les comparer à une solution à taille constante. Ce qui nous donne les relations suivantes :

$$\sum_{b=B}^{B+M} b \quad = \quad M \times B + \frac{M^2}{2} \tag{3.1}$$

au lieu de :

$$\sum_{b=B}^{B+M} B + M \quad = \quad M \times (B + M) \tag{3.2}$$

La figure 3.1 montre le temps de calcul d'une solution à augmentation de taille par pas, rapporté à celui d'une solution sans débordement prévue pour le pire des cas. Chaque courbe représente ce coefficient en fonction du nombre d'échantillons pour une valeur de B, la taille des échantillons. Les quantités d'échantillons communes vont de 1024 à 16 millions d'échantillons. Les gains obtenus peuvent être déja importants, mais ils sont d'autant plus importants pour une valeur de N que B est faible. Notons au passage qu'une valeur de B aussi faible que 8 n'est pas une hérésie, même pour des applications scientifiques comme la méthode cristallographique rappelée dans le chapître 1. En effet, cette valeur permet une définition suffisante pour différencier les valeurs des échantillons dans certaines configurations et l'augmentation de taille au cours du calcul assure le maintien d'une précision compatible avec celle-ci.

II.3.3.3 Opérateurs à coefficients de proportionnalité logarithmiques.

L'INTÉGRATION DES COEFFICIENTS caractérisant les opérateurs considérés nous permet de les comparer à ceux d'une solution à taille constante, puis à ceux à taille croissante dont les coefficients sont logarithmiques. Nous avons donc :

$$\sum_{b=B}^{B+m} \log_2 b \quad = \quad (B + m) \times \log_2(B + m) \Leftrightarrow B \times \log_2 B \Leftrightarrow \frac{m}{\ln 2} \tag{3.3}$$

au lieu de :

$$\sum_{b=B}^{B+m} \log_2 B \quad = \quad m \times \log_2(B + m). \tag{3.4}$$

La figure 3.2 montre pour les mêmes conditions que la figure 3.1 l'amélioration du temps de calcul grâce à l'emploi d'opérateurs à taille croissante. Nous pouvons noter que la diminution du temps de calcul pour des tailles d'opérandes et des nombres d'échantillons donnés est plus faible que pour des opérateurs linéaires, mais fait apparaître les mêmes constatations. Cette amélioration pourrait même apparaître comme trop faible pour être mise en oeuvre, en particulier pour

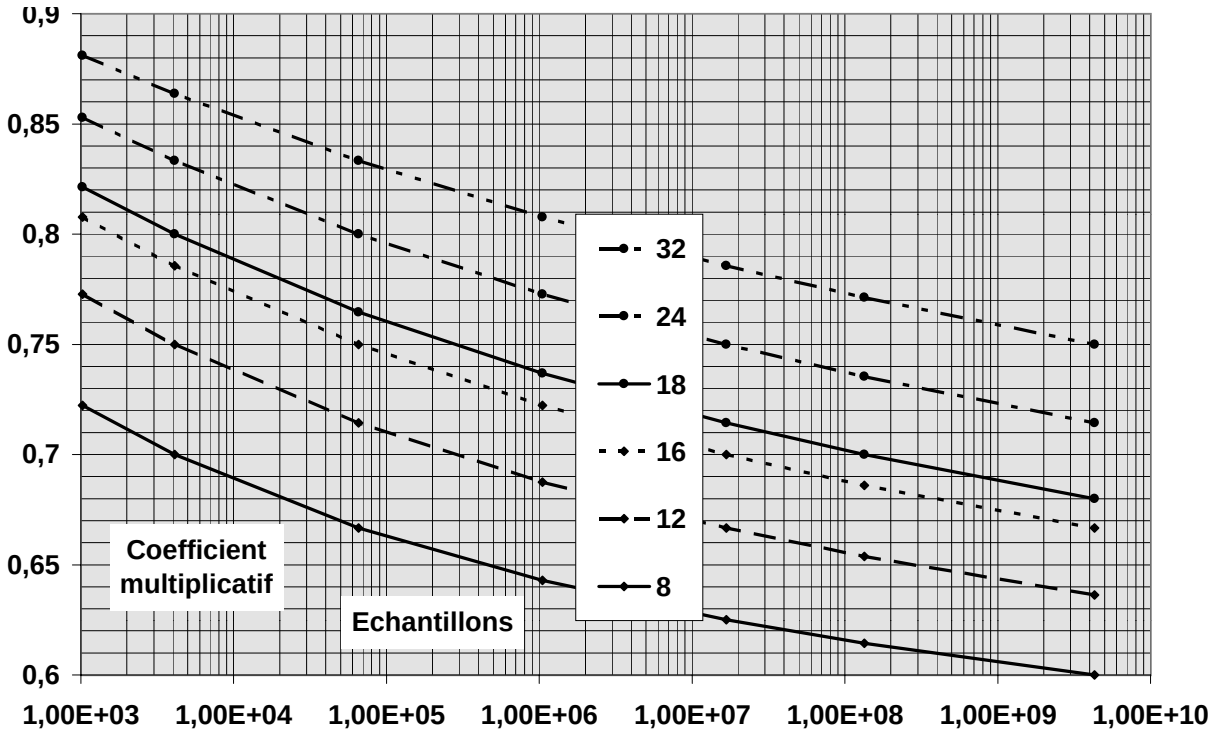

FIG. 3.1 – *temps de calcul relatif pour des opérateurs linéaires.*

les opérateurs à coefficients logarithmiques. Encore ne faudrait-il pas oublier que, par principe de ces opérateurs, la surface nécessaire pour les implanter est plus faible que celle d'opérateurs sans débordement. Il est donc possible d'implanter plus d'opérateurs travaillant en parallèle, donc par ce biais de diminuer encore plus le temps de calcul.

II.3.4 Amélioration de la surface d'implantation.

II.3.4.1 Généralités.

EN TERME DE SURFACE , nous devons intégrer les coefficients de proportionnalité tout au long des étapes successives de la T.F.R. pour en estimer les performances. Le tableau 3.2 indique les différents coefficients de surface selon le type d'opérateurs et leurs caractéristiques de vitesse. Comme les multiplieurs sont beaucoup plus encombrants que les additionneurs pour un coefficient de proportionnalité de temps de calcul donné, nous limitons notre estimation aux multiplieurs.

II.3.4.2 Opérateurs à coefficients de proportionnalité linéaires.

POUR COMPARER UNE SOLUTION à taille croissante à une solution à taille fixe dont les opérateurs sont suffisamment grands pour éviter un débordement dans le pire des cas, nous intégrons leurs coefficients respectifs de proportionnalité pour les diverses étapes de la T.F.R. et nous obtenons :

$$\sum_{n=B}^{B+M} n^2 \quad = \quad B^2 \times M + B \times M^2 + \frac{M^3}{3} \tag{3.5}$$

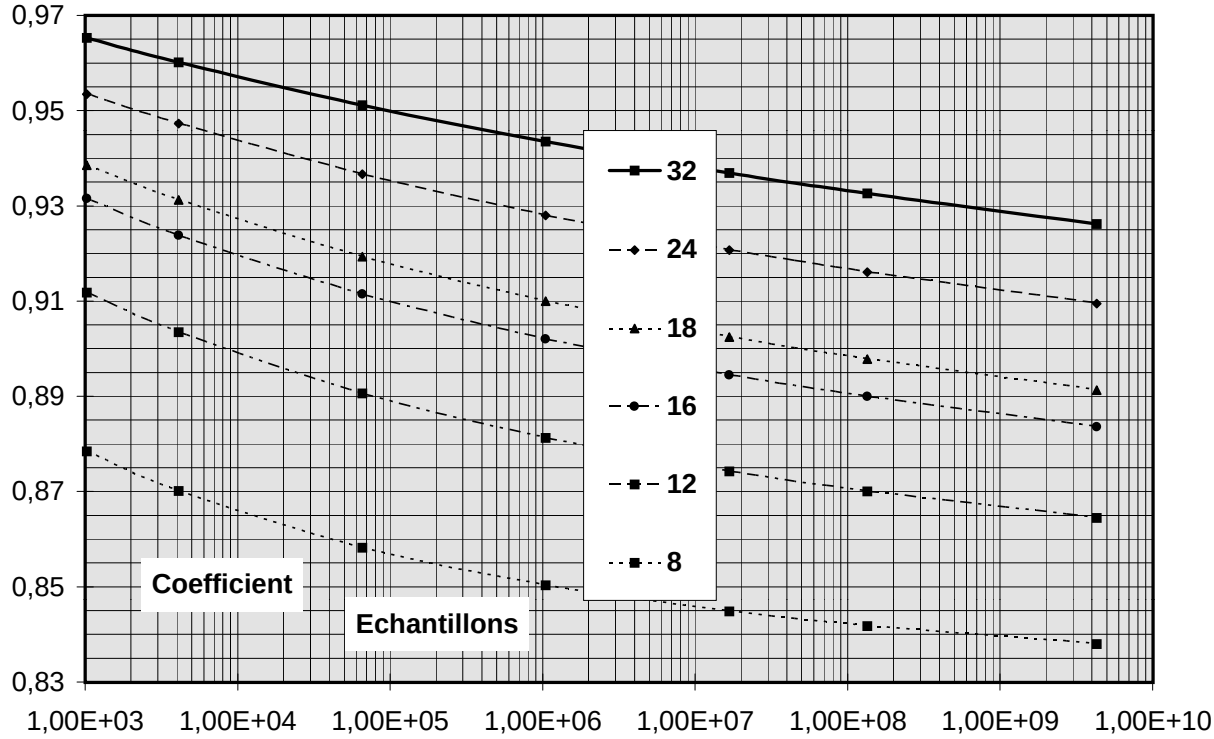

FIG. 3.2 – *temps de calcul relatif pour des opérateurs logarithmiques.*

au lieu de

$$M \times (B+M)^2 \quad = \quad B^2 \times M + 2B \times M^2 + M^3. \tag{3.6}$$

La figure 3.3 représente le rapport des deux précédentes valeurs pour étudier l'amélioration apportée par une solution à taille croissante des opérandes et ce avec les mêmes conditions de tracé que les figures 3.1 et 3.2. Si les courbes représentent les mêmes tendances que celles représentant le temps de calcul, l'amélioration est cependant plus forte. Pour reprendre le cas d'une transformée de Fourier tridimensionnelle de 16 millions de points, le gain varie de un tiers pour des nombres de 32 chiffres à plus de 50% pour des nombres d'une taille inférieure à 12 chiffres binaires, et même un peu plus. Ce qui signifie que, pour des cas de ce genre et pour une

coefficient de proportionnalité du temps de calcul	opérateur	coefficient de proportionnalité de la surface d'implantation
b	$+$	b
	$\times$	b^2
$\log_2 b$	$+$	$b \times \log_2 b$
	$\times$	$b^2 \times \log_2 b$

TAB. 3.2 – *surface fonction des opérateurs et de la vitesse.*

surface donnée, le nombre d'opérateurs peut être doublé, donc le temps de calcul de la T.F.R. divisé par deux, en plus du gain apporté par la simple réduction du temps de calcul propre à chaque opérateur.

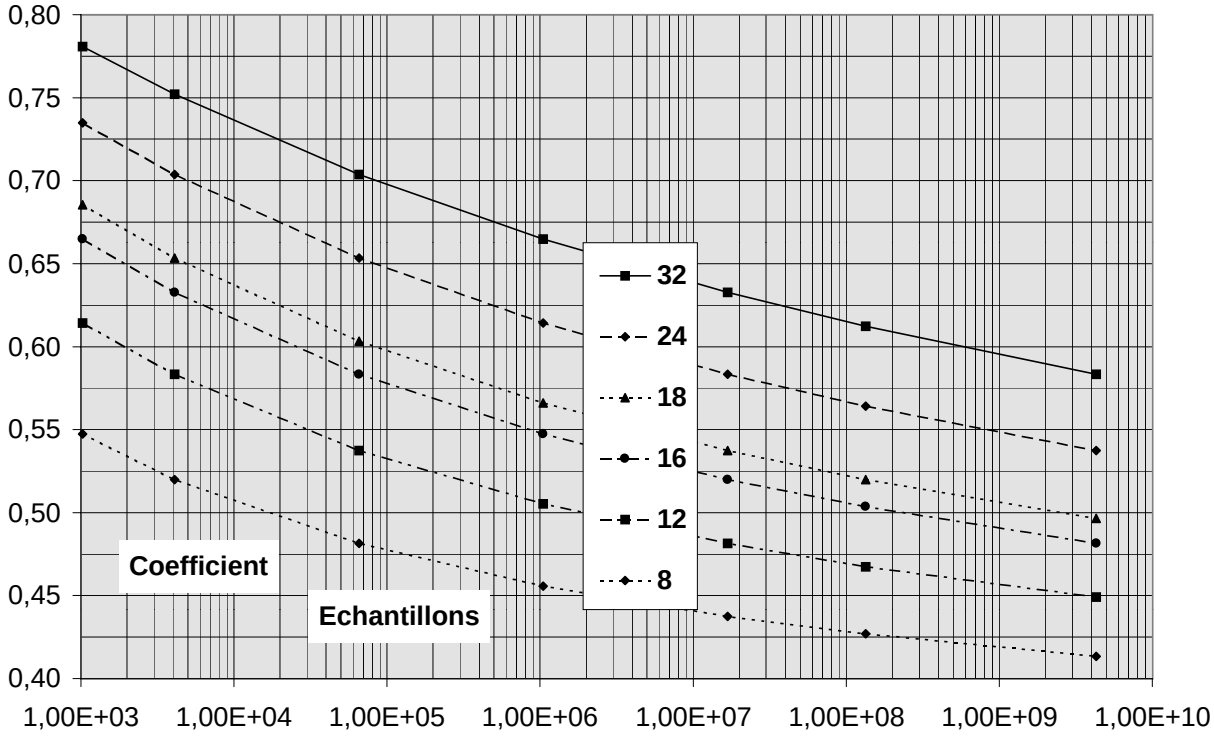

FIG. 3.3 – *surface relative d'une implantation à base d'opérateurs linéaires.*

II.3.4.3 Opérateurs à coefficients de proportionnalité logarithmiques.

NOUS ACCOMPLISSONS À NOUVEAU notre tâche d'intégration de coefficients caractéristiques des opérateurs pour pouvoir comparer une solution à taille croissante par rapport à une taille fixe, dans un premier temps, et, ensuite, par rapport à une solution à taille croissante à coefficients linéaires. Nous avons :

$$\sum_{b=B}^{B+M} b^2 \log_2 b \quad = \quad \frac{[3\ln(B+M) \Leftrightarrow 1] \times (B+M)^3 \Leftrightarrow 3(\ln B \Leftrightarrow 1) \times B^3}{9 \ln 2} \tag{3.7}$$

au lieu de :

$$\sum_{b=B}^{B+M} (B+M)^2 \log_2(B+M) \quad = \quad M \times (B+M)^2 \times \log_2(B+M). \tag{3.8}$$

La figure 3.4 montre la surface d'une implantation à base de tels opérateurs rapportée à celle utilisant des opérateurs de taille constante. Nous pouvons remarquer que l'amélioration est encore plus importante en terme de surface que pour des opérateurs linéaires, ce qui est le contraire de ce qui se passe pour le temps de calcul. Nous retrouvons là le dilemme fréquent entre gagner en surface ou en temps, avoir le beurre ou l'argent du beurre.

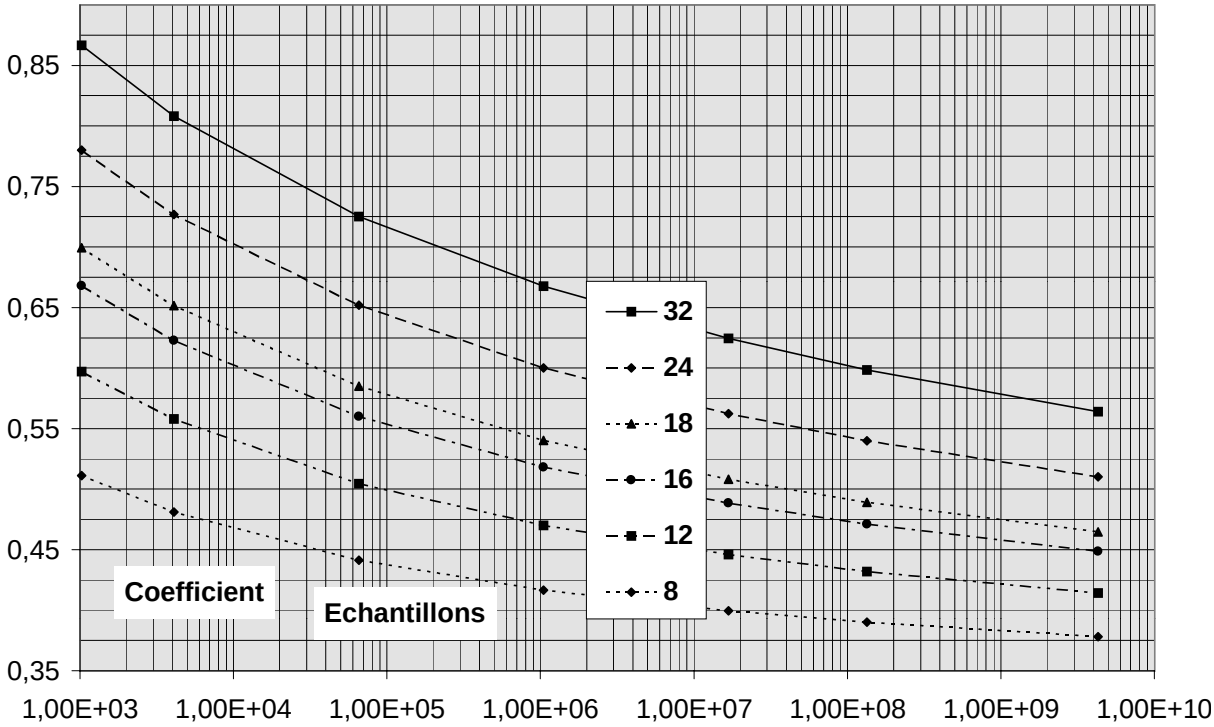

FIG. 3.4 – *surface relative d'une implantation à base d'opérateurs logarithmiques.*

II.3.5 Perspectives.

II.3.5.1 Généralités.

Le calcul d'une T.F.R. par le biais d'opérateurs parallèles présentant une taille croissant avec le rang de l'étape de calcul permet d'améliorer les performances par deux voies :

- pour un nombre donné de papillons, nous augmentons un peu la vitesse de calcul et diminuons la surface totale de l'implantation,
- pour une surface donnée nous augmentons beaucoup la vitesse de calcul, un peu grâce à l'amélioration de vitesse propre à un papillon et le reste grâce à l'augmentation du nombre de papillons dûe à la réduction de la taille nécessaire à un papillon.

L'utilisation d'architectures dédiées pour des calculs répétés et portant sur de grandes quantités de données amène à la fois de hautes performances et un fort taux d'utilisation des fonctions implantées en silicium, ce qui peut se traduire de façon plus imagée par une forte rentabilité de la surface utilisée. La surface d'implantation d'opérateurs parallèles est telle qu'il est totalement inenvisageable de dédier un papillon à chaque calcul élémentaire d'une T.F.R. à effectuer, même en réduisant artificiellement le nombre nécessaire grâce à des techniques comme les transformations $1D \rightarrow 2D$. Une conséquence heureuse de ce fait est que la régularité d'implantation n'est plus un paramètre dominant, ce qui laisse plus de liberté les différents blocs de l'implantation. Le gain en surface est donc plus proche de la réalité qu'avec une solution sérielle. Il est donc possible de profiter pleinement des possibilités de simplifier les papillons mettant en oeuvre des produits par des termes simples $(\pm 1, \pm j)$. Les papillons résultant ne mettent en oeuvre que des opérateurs économes en surface comme les additionneurs.

En ce qui concerne les règles d'emploi de ces opérateurs, nous définissons deux cas types qui correspondent à des stratégies souhaitables, car utilisant au mieux les propriétés de ces opérateurs, même si rien n'empêche de les utiliser autrement.

Nous pouvons nous souvenir que le nombre de coefficients exponentiels qui ne sont pas concernées par ces simplifications décroît avec le rang de l'étape de calcul pour un entrelacement fréquentiel. Pour maintenir un flux de données constant entre les différentes grappes de papillons calculant chacune une étape de la T.F.R., il faut un nombre de papillons qui croît avec le rang de l'étape de calcul pour compenser l'augmentation du temps de calcul d'une étape sur l'autre. Cette variation dépend du type d'opérateurs utilisés. Le nombre de coefficients exponentiels concernés par les simplifications double à chaque étape, mais la variation du nombre de papillons nécessaires est bien inférieure. Nous pouvons donc choisir d'imposer une surface d'implantation nécessaire plus ou moins constante avec le rang et ainsi de faire décroître le temps de calcul des étapes successives. Ceci est trés favorable à une implantation en plusieurs puces, car chaque circuit qui implante une étape peut occuper la taille maximale permise par une technologie et permettre de performances maximales en vitesse de calcul.

Un entrelacement temporel, plus précis comme nous l'avons vu dans le chapître 2, présente le phénomène inverse en ce qui concerne les coefficients exponentiels. Nous avons donc au fil des étapes des phénomènes qui cumulent leurs effets, augmentation de la taille des opérandes et du nombre de papillons comportant des multiplieurs. La possibilité de conserver pour les étapes successives une surface presque constante n'a pas lieu d'être. A moins de prévoir une architecture mettant en oeuvre des circuits dont le nombre croît avec le rang du calcul dans le cas où des impératifs de précision impose cette solution, l'entrelacement temporel s'applique mieux pour une solution monopuce où la surface peut être répartie entre les différentes étapes selon les choix et le travail du seul concepteur du circuit. Des solutions utilisant des modules multi-puces semblent donc le terrain idéal d'application de cette proposition d'opérateurs parallèles à taille croissante. Cela pour bénéficier des hautes performances qu'elles apportent dans un encombrement réduit, malgré leurs exigences.

Chapitre 4

Bases de décomposition supérieures à 4

Sommaire

II.4.1 Introduction 111
II.4.2 Aspects théoriques. 112
II.4.2.1Généralités. 112
II.4.2.2Entrelacement temporel de base 8. 112
II.4.2.3Entrelacement fréquentiel de base 8. 113
II.4.2.4Remarque sur le produit par $\sqrt{2}$. 115
II.4.2.5Entrelacement temporel de base 12. 116
II.4.2.6Entrelacement fréquentiel de base 12. 118
II.4.3 Amélioration en terme de vitesse. 120
II.4.4 Généralités. 120
II.4.4.1Nombre d'étapes successives de calcul. 122
II.4.4.2Nombre de multiplications. 123
II.4.5 Problème de la surface d'implantation. 124
II.4.6 Conclusion. 125

II.4.1 Introduction

Le choix d'une base d'une T.F.R. conditionne le nombre d'étapes successives qui apparaissent dans le calcul correspondant, donc la vitesse maximale de celui-ci. Une base égale à 2 ou 4 permet d'intéressantes simplifications dans les expressions de base apparaissant dans les décompositions d'une transformée de Fourier. Le paragraphe 4.2 nous permet de présenter le raisonnement qui nous a permis de proposer l'usage d'une base 8 pour le calcul d'une transformée de Fourier rapide [VA95a] pour un entrelacement temporel. Nous généralisons ce cas à un entrelacement fréquentiel et ensuite nous étendons nos arguments à la base 12. Nous étudions les gains en vitesse apportées par les différents opérateurs et implantations possibles dans le paragraphe 4.3 pour ces propositions. Nous évaluons le coût en surface qui en découle dans le paragraphe 4.5 et finalement dressons les conclusions de cet aspect de la T.F.R. dans le paragraphe 4.6.

II.4.2 Aspects théoriques.

II.4.2.1 Généralités.

Nous considérons le cas d'une T.F.R. de seulement $N = p^2$ échantillons, où p est la base de la décomposition pour éviter d'avoir à manipuler de trop grosses expressions. Nous supposons que les nombres sont complexes et que j est tel que $j^2 = \Leftrightarrow 1$. F est la transformée de Fourier de f, les indices sont respectivement k et n. Nous pouvons écrire ceux-ci avec la base p :

- $k = k_2 + p \times k_1$
- $n = p \times n_2 + n_1$

II.4.2.2 Entrelacement temporel de base 8.

$$\begin{aligned} F(k) &= \sum_{n_1=0}^{7} \left(\sum_{n_2=0}^{7} f(8n_2 + n_1) e^{-2\pi j \frac{nk}{64}} \right) \\ \Leftrightarrow F(k) &= \sum_{n_1=0}^{7} \left(\sum_{n_2=0}^{7} f(8n_2 + n_1) e^{-2\pi j \frac{(k_2 + p \times k_1)(p \times n_2 + n_1)}{64}} \right) \end{aligned} \tag{4.1}$$

comme $e^{-2\pi j a} = 1$ pour tout relatif, nous avons :

$$F(k) = \sum_{n_1=0}^{7} \left(\sum_{n_2=0}^{7} f(8n_2 + n_1) e^{-2\pi j \frac{k_2 n_2}{8}} \right) \times \underbrace{e^{-2\pi j \frac{k_2 n_1}{64}}}_{W} \times e^{-2\pi j \frac{k_1 n_1}{8}} \tag{4.2}$$

W est le terme exponentiel qui apparaît à la fin de chaque calcul élémentaire de la T.F.R., pour le premier pas il est égal à 1. Pour faire un parallèle avec la base 2 bien connue, rappelons la forme des expressions de base : $A \pm B \times W$ pour l'entrelacement temporel, $A + B$ et $(A \Leftrightarrow B) \times W$ pour l'entrelacement fréquentiel où A et B sont les coefficients d'entrées du papillon considéré.

Nous pouvons remarquer que $e^{-2\pi j \frac{k_i n_i}{8}} = e^{-\pi j \frac{k_i n_i}{4}}$. La figure 4.1 montre le cercle trigonométrique avec les points correspondant aux différentes valeurs des coefficients exponentiels selon n_i et k_i.

L'ensemble de valeurs de $e^{-\pi j \frac{k_i n_i}{4}}$ est constitué des nombres ± 1, $\pm j$ et $(\pm 1 \pm j) \times \frac{\sqrt{2}}{2}$. Il est à noter que seul le terme $\frac{\sqrt{2}}{2}$ peut être gênant. En effet multiplier par ± 1 (inversion du signe), $\pm j$ (permuter les parties réelles et imaginaires avec éventuellement inversion du signe) ou par $(\pm 1 \pm j)$ qui n'est que la combinaison des deux précédentes peut être réalisé avec à la fois un faible coût de surface d'implantation et de temps de calcul.

Revenons à une T.F.R. de base 2. Les expressions de base sont calculables en deux étapes, une de multiplication par des coefficients exponentiels, et une d'addition ou de soustraction. Selon l'entrelacement, l'une de ces étapes a lieu avant l'autre. Pour obtenir un gain en vitesse de calcul, nous devons conserver la même structure dans les papillons de base 8, c'est à dire une étape de multiplications et une d'opérations simples, additions éventuellement liées à des multiplications par ± 1, $\pm j$ ou $\pm 1 \pm j$. Les multiplieurs sont les opérateurs les plus coûteux en temps de calcul, soit directement pour les opérateurs sériels ou parallèles avec une notation en complément à deux, soit à travers la latence pour les opérateurs sériels avec une notation redondante. Pour conserver ces deux uniques niveaux d'opérations, il faut que les entrées qui doivent être multipliées par $\frac{\sqrt{2}}{2}$

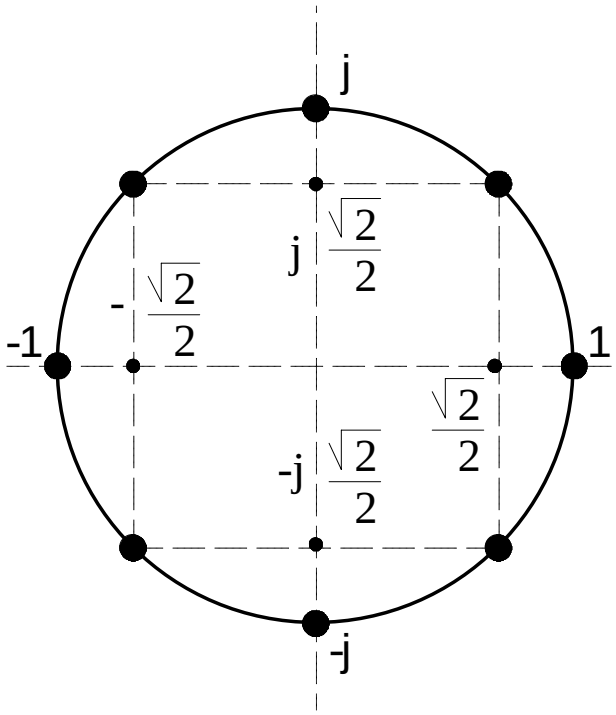

FIG. 4.1 – *valeurs remarquables des coefficients exponentiels pour une base* 8.

soient dédoublées [VA95a], une forme normale et une multipliée par $\frac{\sqrt{2}}{2}$. Les multiplications par $\frac{\sqrt{2}}{2}$ doivent être faîtes au même niveau que les multiplications par W, ce qui revient à réaliser deux multiplications en parallèle pour un seul coefficient exponentiel, une avec W et une autre avec $\frac{\sqrt{2}}{2} \times W$.

La structure mathématique du papillon en base 8 est représentée sur la figure 4.2 en regard de celle du papillon en base 2 pour visualiser la différence fondamentale entre les deux cas. Dans un papillon de base 2 le flot des données ne traverse que deux couches d'opérateurs, une d'opérateurs et une de multiplieurs. Dans un papillon de base supérieure à 4, ici 8, elles rencontrent trois couches d'opérateurs, dont deux de multiplieurs qui sont les plus pénalisant au point de vue temps de calcul. Toute accélération des calculs dûe à une diminution du nombre total de couches de papillon est atténuée par l'augmentation du nombre de couches internes des papillons. La figure 4.3 représente l'architecture résultant de notre proposition pour un papillon de base 8. Nous rappelons celle d'un papillon de base 2 et faisons apparaître l'architecture cablée avec son dédoublement des chemins de données, avec ou sans multiplication par $\frac{\sqrt{2}}{2}$, pour rétablir la structure en deux couches des papillons de base 2 qui est aussi celle des papillons de base 4.

II.4.2.3 Entrelacement fréquentiel de base 8.

$$F(k) = \sum_{n_1=0}^{7} (\sum_{n_2}^{7} f(8n_2 + n_1) e^{-2\pi j \frac{n.k}{64}})$$

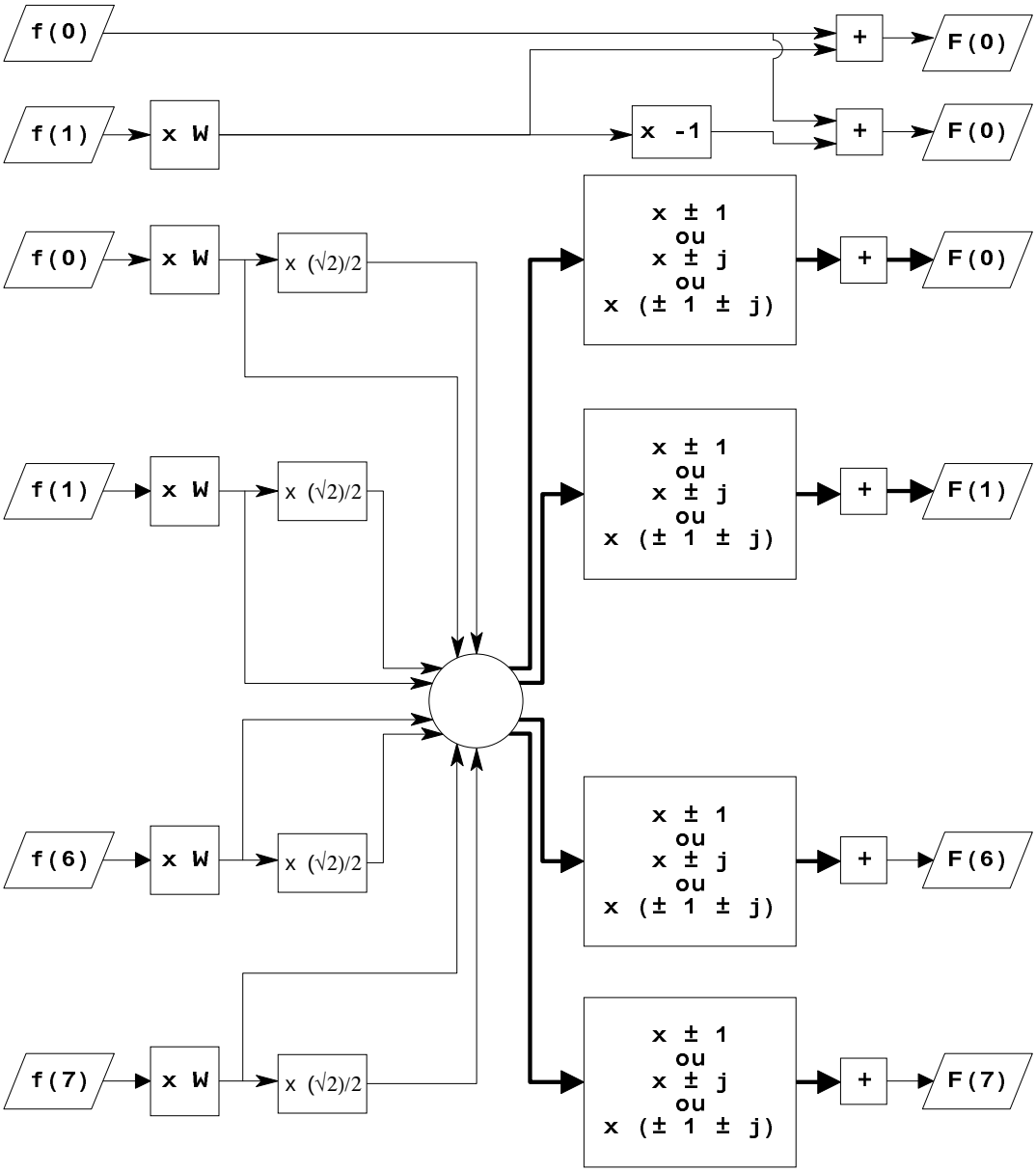

FIG. 4.2 – *structure mathématique d'un papillon de base* 8 *pour un entrelacement temporel.*

$$\begin{aligned} \Leftrightarrow F(k) &= \sum_{n_1=0}^{7}\left(\sum_{n_2}^{7} f(8n_2+n_1)e^{-2\pi j\frac{(8n_2+n_1).(8k_2+k_1)}{64}}\right) \\ \Leftrightarrow F(k) &= \sum_{n_1=0}^{7}\left(\sum_{n_2=0}^{7} f(8n_2+n_1)e^{-2\pi j\frac{k_2n_2}{8}} \times \underbrace{e^{-2\pi j\frac{k_2n_1}{64}}}_{W}\right) \times e^{-2\pi j\frac{k_1n_1}{8}} \end{aligned} \tag{4.3}$$

Reprenons la comparaison avec la base 2. L'équation 4.3 a la même forme que $A+B$ et $(A \Leftrightarrow B) \times W$. La multiplication par W est faite à la sortie du papillon, mais celle par $\frac{\sqrt{2}}{2}$ à l'entrée et sans porter sur toutes les entrées d'un ensemble concerné par une multiplication donnée de W. Cette structure est traduite dans le schéma de la figure 4.4. Nous ne pouvons

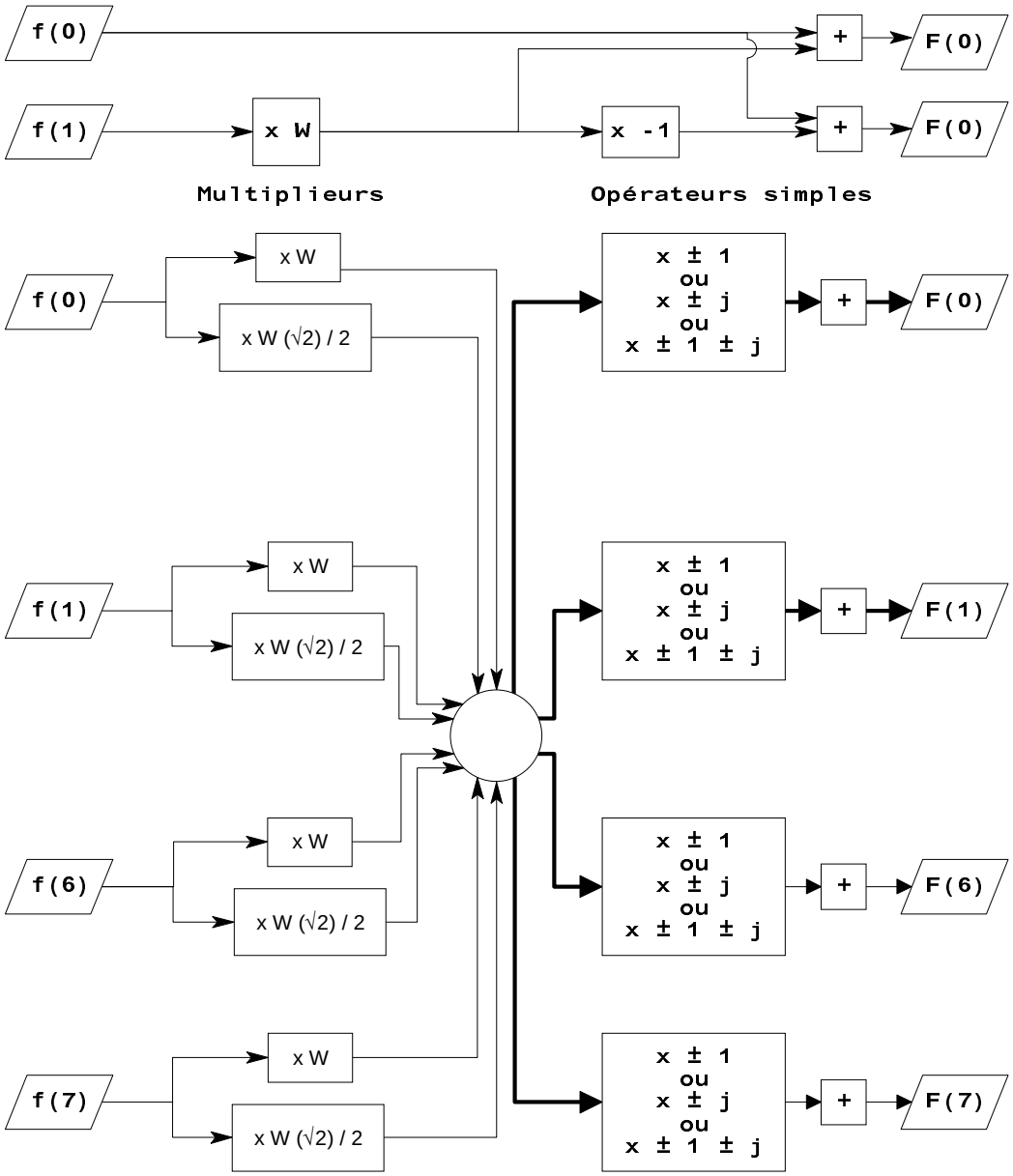

FIG. 4.3 – *architecture cablée d'un papillon de base* 8 *pour un entrelacement temporel.*

donc pas regrouper à l'intérieur du papillon les multiplications par W et par $\frac{\sqrt{2}}{2}$. Nous sommes obligés de déplacer ces produits par $\frac{\sqrt{2}}{2}$ dans les papillons précédents pour les regrouper avec les produits par W. Ce qui double les données à transmettre entre deux rangées successives de papillons. Cela donne l'architecture de la figure 4.5 où les termes multipliés par $\frac{\sqrt{2}}{2}$ en entrée et en sortie sont indiqués par un signe « ' » supplémentaire par rapport au terme équivalent sans ce produit.

II.4.2.4 Remarque sur le produit par $\sqrt{2}$.

DANS LE CAS D'UNE ARCHITECTURE CABLÉE avec des opérateurs parallèles ou sériels en com-

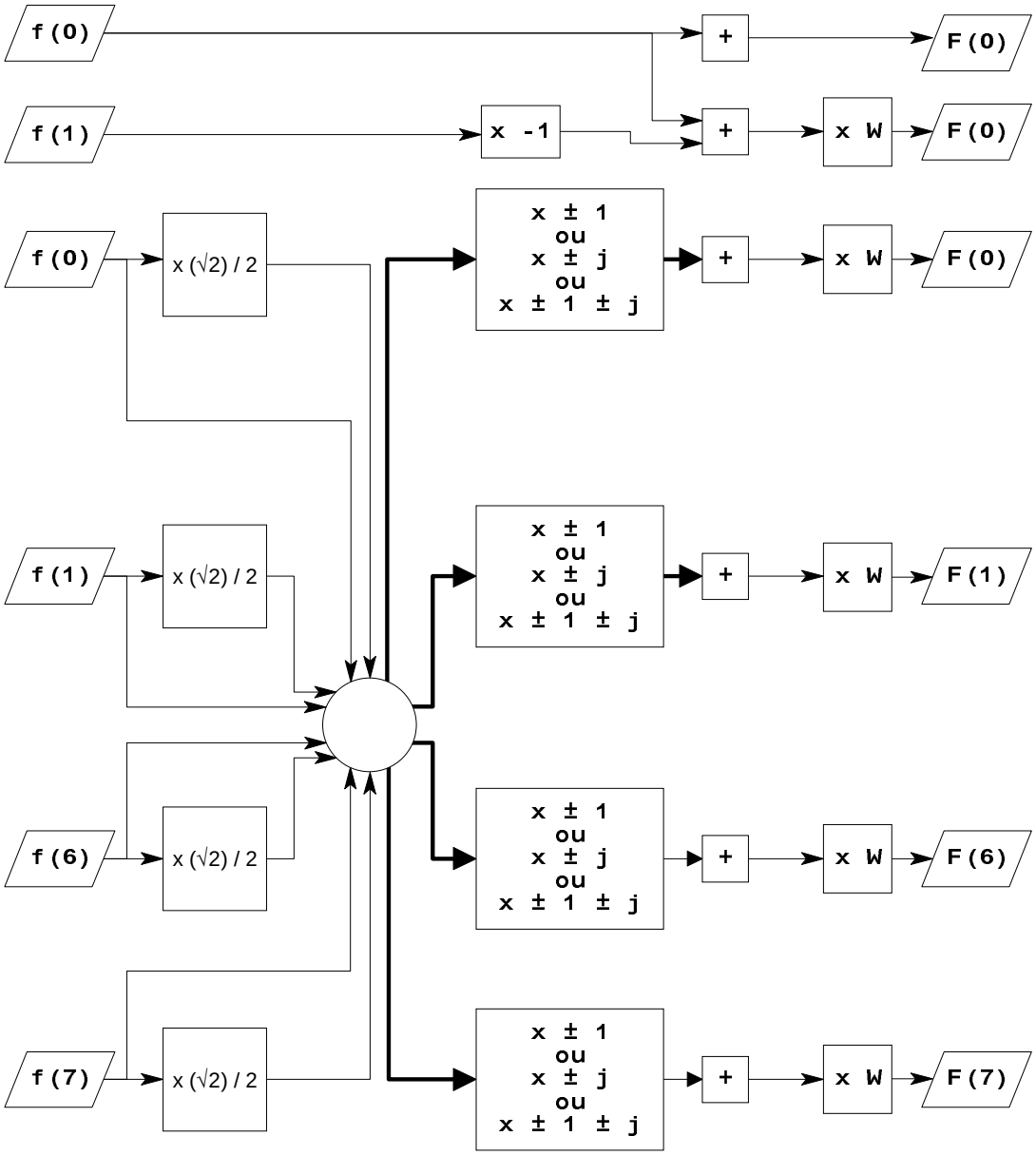

FIG. 4.4 – *structure mathématique d'un papillon de base* 8 *pour un entrelacement fréquentiel.*

plément à deux, nous pourrions envisager de garder la structure mathématique du papillon, sans utiliser le dédoublement des produits tel qu'il vient d'être décrit en raison de la valeur constante et particulière qu'est $\frac{\sqrt{2}}{2}$. Un multiplieur cablé et taillé pour cette valeur a un temps de réponse plus court qu'un multiplieur standard. Il est toutefois plus grand que celui des opérateurs simples, donc reste pénalisant au niveau temps de calcul. Il permet toutefois une surface d'implantation plus faible que notre solution.

II.4.2.5 Entrelacement temporel de base 12.

REPRENONS NOTRE RAISONNEMENT précédent pour $p = 12$. Notons que nous sortons des

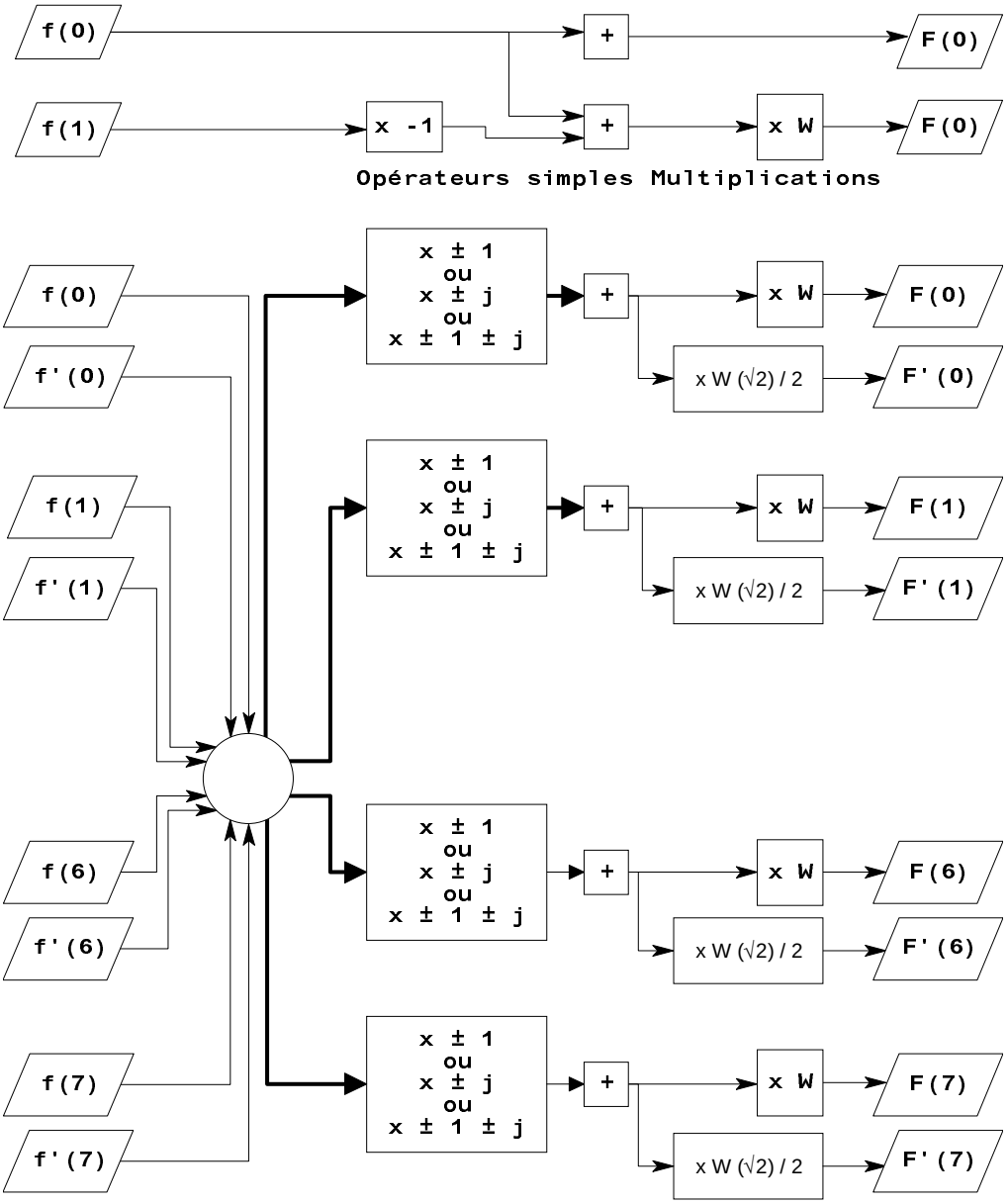

FIG. 4.5 – *architecture cablée d'un papillon de base* 8 *pour un entrelacement fréquentiel.*

puissances de 2 chère aux adeptes des T.F.R. de tout acabit.

$$\begin{aligned} F(k) &= \sum_{n_1=0}^{11}\sum_{n_2}^{11} f(12n_2+n_1)e^{-2\pi j\frac{nk}{144}} \\ \Leftrightarrow F(k) &= \sum_{n_1=0}^{11}(\sum_{n_2=0}^{11} f(12n_2+n_1)e^{-2\pi j\frac{k_2n_2}{12}}) \times \underbrace{e^{-2\pi j\frac{k_2n_1}{144}}}_{W} \times e^{-2\pi j\frac{k_1n_1}{12}} \end{aligned} \tag{4.4}$$

Nous avons $e^{-2\pi j\frac{k_in_i}{12}} = e^{-\pi j\frac{k_in_i}{6}}$. La figure 4.6 montre le cercle trigonométrique avec les points correspondant aux différentes valeurs remarquables selon n_i and k_i.

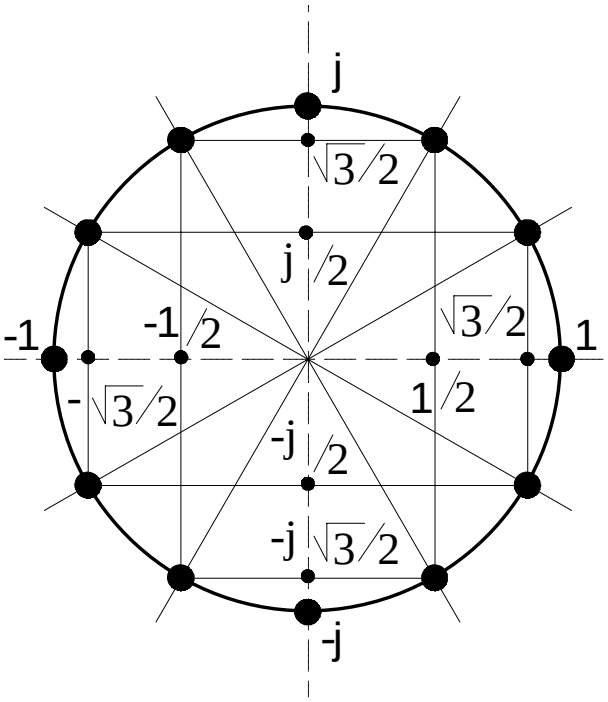

FIG. 4.6 – *valeurs remarquables des coefficients exponentiels d'un papillon de base* 12.

Avec le même raisonnement que dans le paragraphe 4.2.2, les valeurs exponentielles qui posent des problèmes à implanter sont celles dont les parties imaginaires sont de la forme $\pm\frac{1}{2}$ ou $\pm\frac{\sqrt{3}}{2}$. Notons cependant que diviser un nombre binaire par 2 a un coût trés faible en surface et en temps de calcul puisqu'il peut être réalisé au moyen d'un simple décalage. Il reste donc seulement $\sqrt{3}$ comme problème. Nous utilisons donc les mêmes techniques que pour les entrelacements de base 8. Les entrées devant être multipliées par un coefficient exponentiel égal à $\pm\frac{1}{2}\pm j\frac{\sqrt{3}}{2}$ ou $\pm\frac{\sqrt{3}}{2}\pm j\frac{1}{2}$ sont dédoublées. Une forme normale pour les coefficients exponentiels égaux à ±1 ou $\pm j$ et une forme dérivée pour les autres qui nécessitent aussi la première forme, à un coefficient $\frac{1}{2}$ prés. Cette division par deux est réalisée au même niveau que les opérations de même complexité, c'est à dire les multiplications par ±1 or $\pm j$. Cela nous donne le schéma de la figure 4.7. Dans la première partie du papillon W et $\sqrt{3}$ sont réunis pour générer la forme dérivée ainsi que l'illustre la figure 4.8.

II.4.2.6 Entrelacement fréquentiel de base 12.

L'ÉQUATION 4.5 peut être réorganisée pour retrouver la formule caractérisant ce type d'entrelacement avec la valeur particulière de $p = 12$.

$$F(k) = \sum_{n_1=0}^{11} \Big(\sum_{n_2=0}^{11} x(12n_2+n_1) e^{-2\pi j \frac{k_2 n_2}{12}} \times \underbrace{e^{-2\pi j \frac{k_2 n_1}{144}}}_{W} \Big) \times e^{-2\pi j \frac{k_1 n_1}{12}} \qquad (4.5)$$

Par analogie avec les raisonnements précédents, nous pouvons déduire les règles suivantes :

- les entrées du papillon sont dédoublées pour être présentes sous la forme brute et une forme multipliée par $\sqrt{3}$ qui est spécifiée grâce au signe « ' » qui lui est ajouté

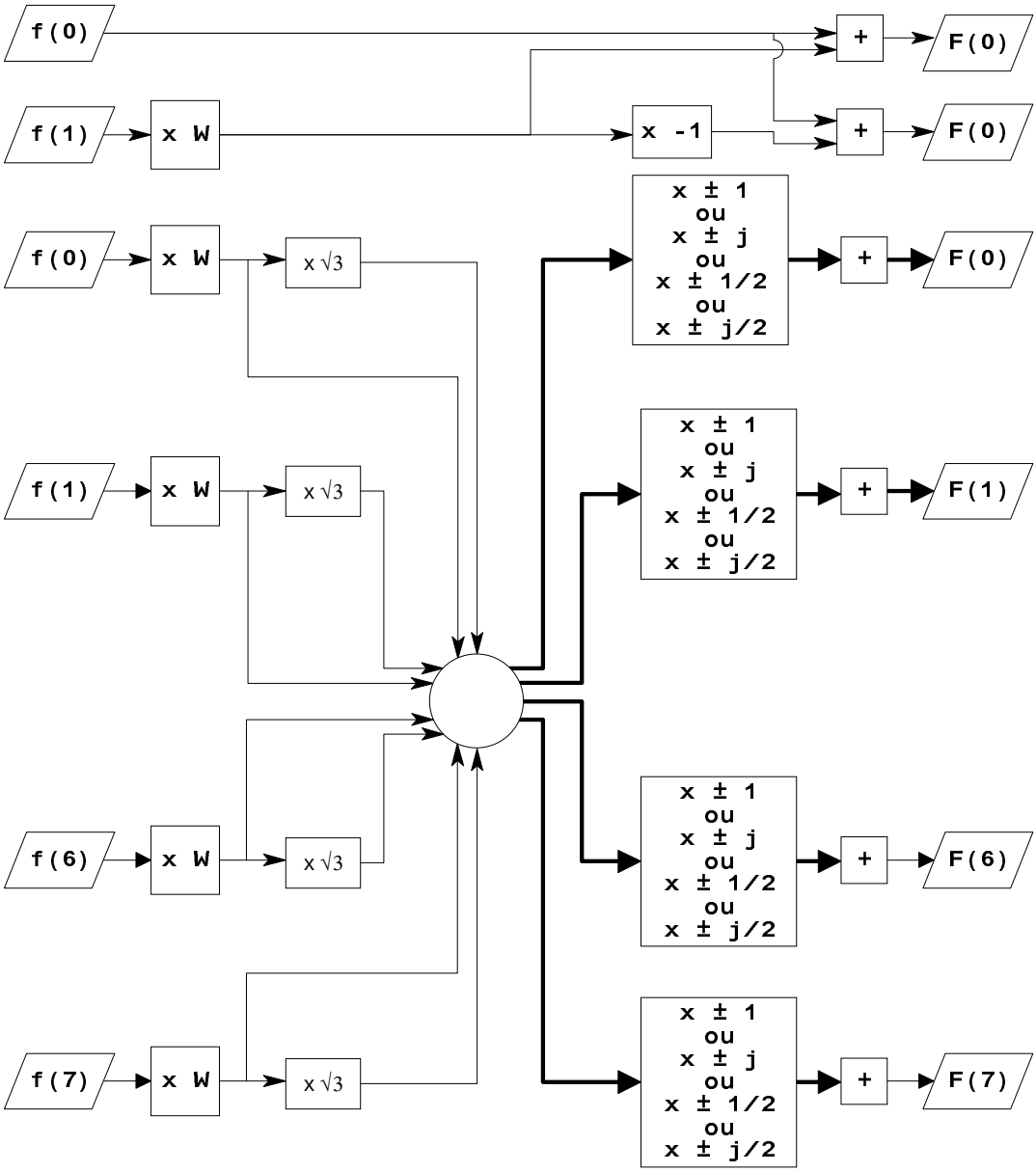

FIG. 4.7 – *structure mathématique d'un papillon de base* 12 *à entrelacement temporel.*

– les coefficients exponentiels de la forme $\pm\frac{1}{2} \pm j\frac{\sqrt{3}}{2}$ ou $\pm\frac{\sqrt{3}}{2} \pm j\frac{1}{2}$ sont reconstruits dans le second niveau du papillon qui implante les additions, inversions du signe, divisions par 2 et échange de parties réelles et imaginaires

– les produits par $\sqrt{3}$ d'un papillon donné migre dans les papillons précédents pour être amalgamé au terme W concerné.

Cela nous donne la structure mathématique de la figure 4.9 et l'architecture de la figure 4.10.

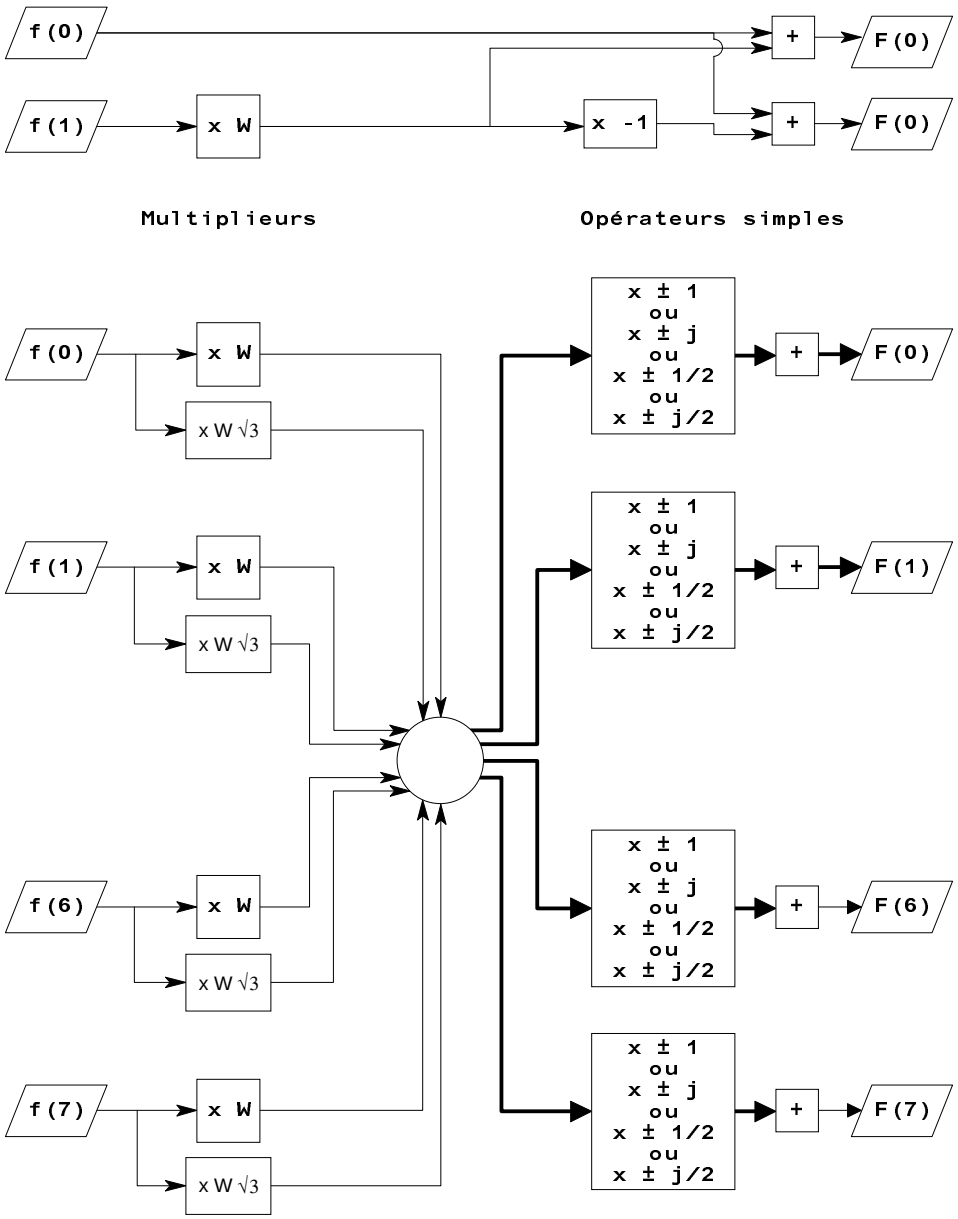

FIG. 4.8 – *architecture cablée d'un papillon de base* 12 *à entrelacement temporel.*

II.4.3 Amélioration en terme de vitesse.

II.4.4 Généralités.

IL Y A DEUX MOYENS D'ESTIMER la vitesse de calcul d'une T.F.R. que nous étudions ensuite successivement :

– par le nombre de multiplications, les opérations les plus coûteuses, ce qui correspond soit à une implantation logicielle, soit à une implantation câblée qui ne met pas en oeuvre autant de papillons que de calculs élémentaires nécessaires pour calculer au moins une étape de la T.F.R. complètement parallélisée. Une architecture basée sur des opérateurs parallèles en

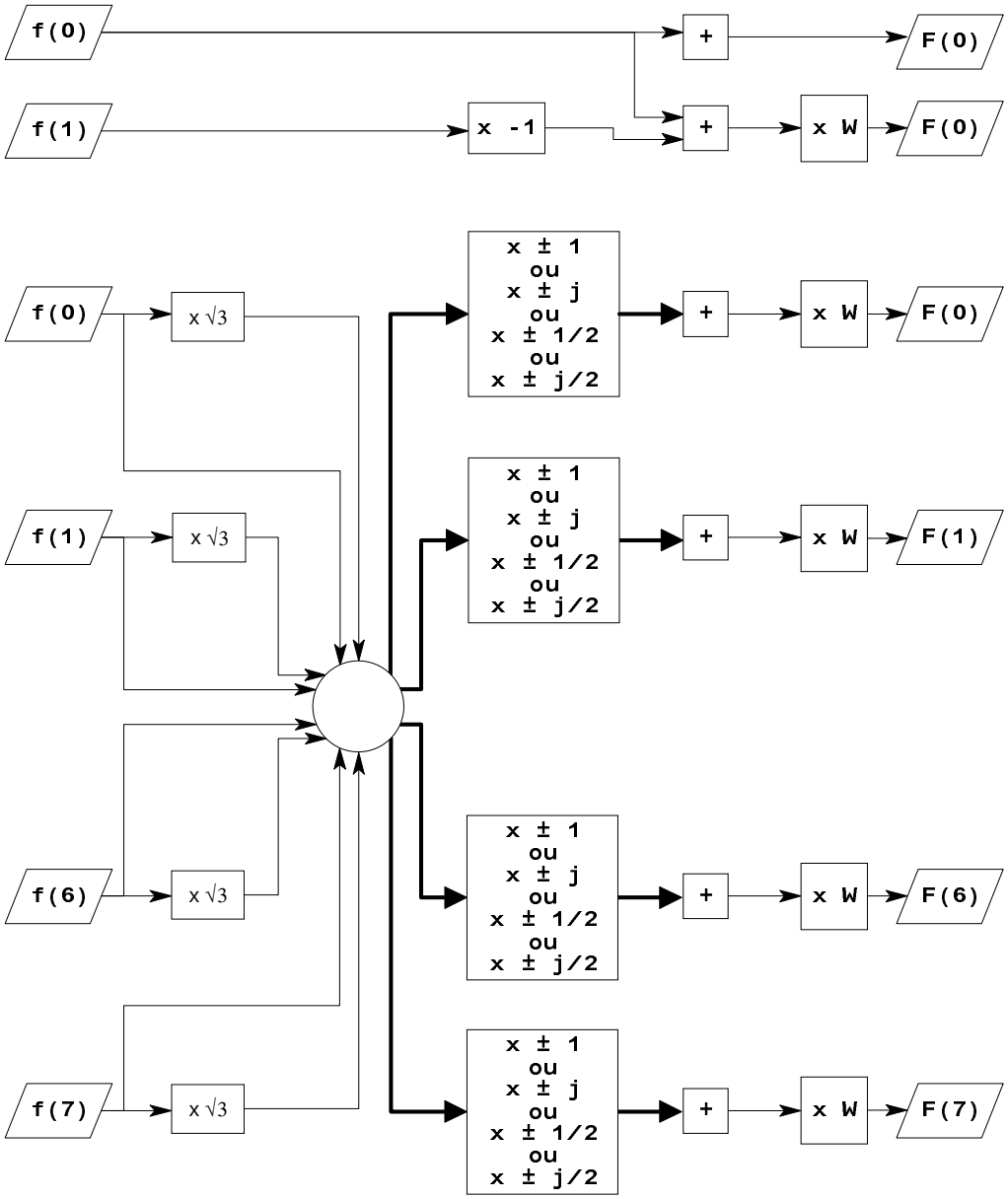

FIG. 4.9 – *structure mathématique d'un papillon de base* 12 *avec un entrelacement fréquentiel.*

est l'exemple typique.

– Le nombre d'étapes de calcul qui limite la vitesse maximale de calcul d'une solution matérielle suffisamment pourvue en papillons pour constituer au moins une barette effectuant en parallèle tous les calculs élémentaires d'une étape de la T.F.R., ce qui correspond actuellement pour ce qui concerne des réalisations envisageables à une implantation à base d'opérateurs sériels.

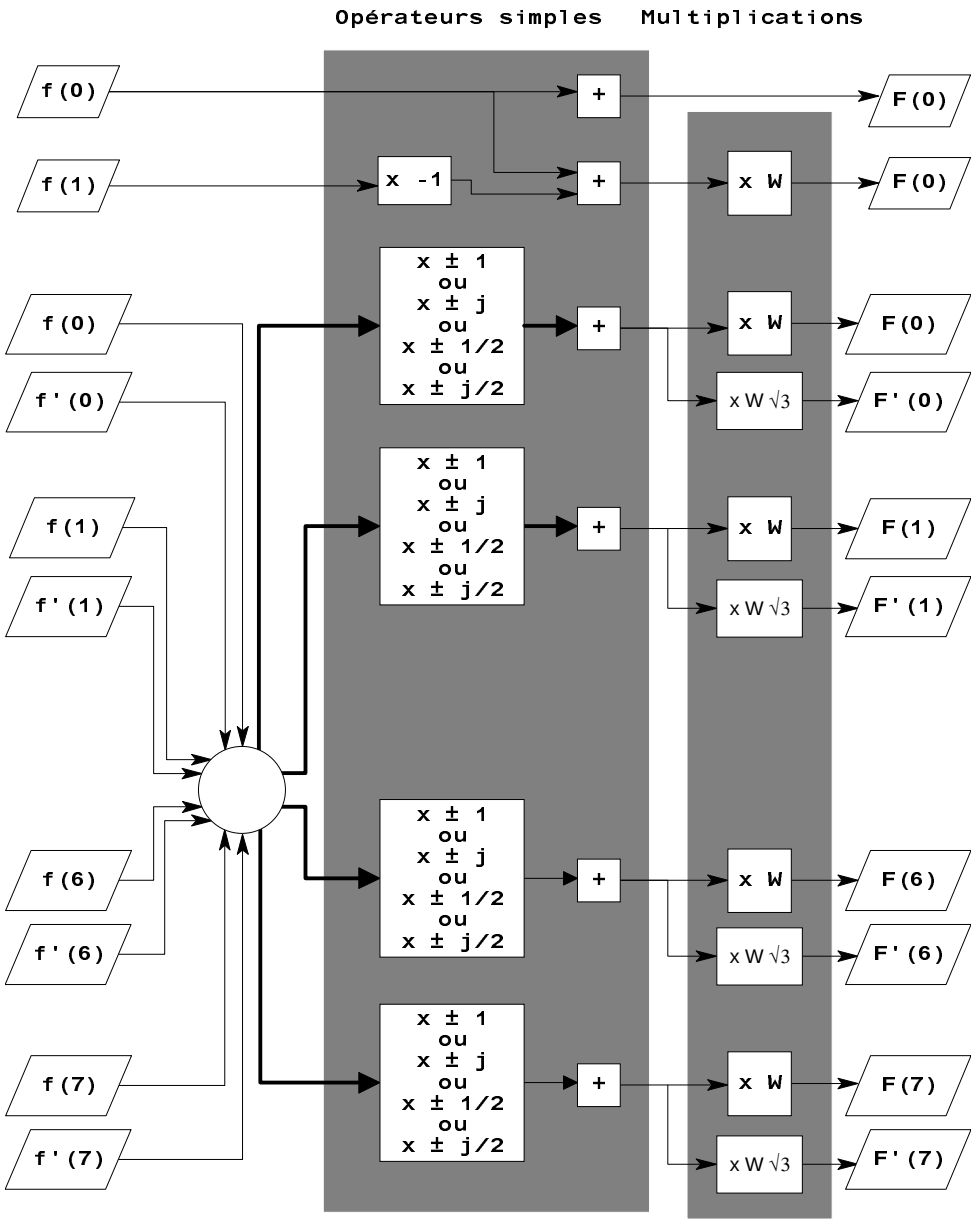

FIG. 4.10 – *architecture d'un papillon de base* 12 *avec un entrelacement fréquentiel.*

II.4.4.1 Nombre d'étapes successives de calcul.

UN NOMBRE DE CHIFFRES BINAIRES égal à M dans une base p a un nombre de valeurs différentes égale à p^M. Par application de cette propriété au principe de décomposition d'une transformée de Fourier selon la base de codage des indices, une base p implique M étapes. Pour comparer des bases différentes, le plus simple est de le faire par rapport à une référence que nous choisissons comme étant la base 2. Si nous considérons une T.F.R. de $N = p^m$ échantillons, nous avons $N = p^M = (2^r)^M = 2^{r.M}$. Ce qui se traduit par M étapes de calcul en base p et $r.M$ en base 2, d'où un gain en nombre d'étapes égal à r en faveur de la base p.

Pour une base 8, nous avons $8 = 2^3$, donc trois fois moins d'étapes. Pour une base 12, comme $\log_2 12 \approx 3.6$, le nombre d'étapes est divisé par 3.6. Cela doit être regardé comme une tendance, car aucun nombre n'est à la fois puissance de 8 et 12. Nous pouvons nous souvenir qu'une base 4 apporte un gain de 2 au point de vue du nombre d'étapes.

II.4.4.2 Nombre de multiplications.

REPRÉSENTONS SOUS FORME DE TABLEAUX les types d'opérations qui apparaissent pour chaque coefficient d'un papillon donné. Rappelons que leur nombre ne diffère pas selon le type de décomposition pour une base donnée. Chaque colonne est dédiée à une des données en entrée du papillon, chaque ligne à un des résultats fournis par le papillon. Chaque case du tableau contient donc le type des opérations que le signal d'entrée doit subir pour apparaître dans la valeur de sortie considérée. Les multiplications par W ne sont indiquées en raison de sa présence systématique. Un « $+$ » indique qu'au moins une addition doit être faîte, un « $\Leftrightarrow$ » qu'au moins une inversion de signe doit être réalisée, un « j » qu'il y a au moins une permutation entre partie réelle et imaginaire et « $\times$ » au moins une multiplication par $\frac{\sqrt{2}}{2}$ pour une base 8 ou par $\sqrt{3}$ pour une base 12.

Le tableau 4.1 concerne la base 8. Les multiplications par $\frac{\sqrt{2}}{2}$ concernent la moitié des données en entrées, ceux qui ayant un indice impair ne permettent pas une simplification de l'exposant de $e^{-\pi j \frac{k_i n_i}{4}}$ sous une forme $\Leftrightarrow \pi j \frac{a}{2}$ ou $\Leftrightarrow \pi j a$ où a est un entier. Le nombre de multiplications est indépendant de l'entrelacement. Pour un entrelacement temporel, les entrées subissent une multiplication par W et parallèlement pour une moitié d'entre eux une multiplication par $W \times \frac{\sqrt{2}}{2}$. Pour un entrelacement fréquentiel, les multiplications se font avant la sortie des résultats. Le nombre de multiplications pour une étape est donc multiplié par 1.5, mais le nombre d'étapes est divisé par 3. Le nombre global de multiplications pour une T.F.R. de base 8 est divisé par 2, le temps de calcul aussi. Cependant pour un système multiprocesseur, la réduction du nombre d'étapes diminue le nombre de données à transmettre dans le système. Ce qui se traduit par une simplification de la communication.

	0	1	2	3	4	5	6	7
0	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
1	$+$	$\times, j, \Leftrightarrow$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	j	$\times, j$
2	$+$	$\Leftrightarrow, j$	$\Leftrightarrow$	j	$+$	$\Leftrightarrow, j$	$\Leftrightarrow$	j
3	$+$	$\times, j, \Leftrightarrow$	j	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$
4	$+$	$\Leftrightarrow$	$+$	$\Leftrightarrow$	$+$	$\Leftrightarrow$	$+$	$\Leftrightarrow$
5	$+$	$\times, j, \Leftrightarrow$	$j, \Leftrightarrow$	$\times, j$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	j	$\times, j, \Leftrightarrow$
6	$+$	j	$\Leftrightarrow$	$j, \Leftrightarrow$	$+$	j	$\Leftrightarrow$	$j, \Leftrightarrow$
7	$+$	$\times, j$	j	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$

TAB. 4.1 – *opérations intervenant dans un papillon de base* 8.

Le tableau 4.2 est identique au tableau 4.1, mais pour une base 12. Les multiplications par $\sqrt{3}$ sont appliquées aux deux tiers des entrées, celles dont l'indice n'est pas un multiple de 3 et ne permet pas une simplification de l'exposant de $e^{-\pi j \frac{k_i n_i}{6}}$ pour amener une valeur de $\Leftrightarrow \pi j \frac{a}{2}$ ou

	0	1	2	3	4	5	6	7	8	9	10	11
0	+	+	+	+	+	+	+	+	+	+	+	+
1	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	j	$\times, j$	$\times, j$
2	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j$	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j$
3	+	$\Leftrightarrow, j$	$\Leftrightarrow$	j	+	$\Leftrightarrow, j$	$\Leftrightarrow$	$\Leftrightarrow, j$	+	j	$\Leftrightarrow$	$j, \Leftrightarrow$
4	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$
5	+	$\times, j, \Leftrightarrow$	$\times, j$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j$	$\times, j, \Leftrightarrow$	j	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$
6	+	$\Leftrightarrow$	+	$\Leftrightarrow$	+	$\Leftrightarrow$	+	$\Leftrightarrow$	+	$\Leftrightarrow$	+	$\Leftrightarrow$
7	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	j	$\times, j$	$\times, j, \Leftrightarrow$
8	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	+	$\times, j, \Leftrightarrow$	$\times, j$	+	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	0	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$
9	+	j	$\Leftrightarrow$	j	+	j	$\Leftrightarrow$	j	+	j	$\Leftrightarrow$	j
10	+	$\times, j$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	+	$\times, j$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$
11	+	$\times, j$	$\times, j$	$\Leftrightarrow, j$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	$\Leftrightarrow$	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$	j	$\times, j, \Leftrightarrow$	$\times, j, \Leftrightarrow$

TAB. 4.2 – *opérations intervenant dans un papillon de base* 12.

$\Leftrightarrow\!\!\!\!\pi ja$. Comparé à une T.F.R. de base 2, le nombre d'étapes est divisé par 3.6 et le nombre de multiplications est augmentée des deux tiers. D'où un facteur global de $0,465$ qui procure un gain en vitesse de calcul de $2,15$. Cela peut apparaître comme guère différent de l'amélioration apportée par une base 4, encore ne faut-il pas oublier là aussi le problème du nombre d'étapes qui réduit le transfert des données dans des systèmes multiprocesseurs.

Jetons par ailleurs un coup d'oeil aux puissances de 8 et 12. Le tableau 4.3 montre les premières puissances de 2, éventuellement 8 et 12, parfois mélangées. Nous pouvons remarquer une meilleure couverture de l'espace des nombres avec ces bases mixtes. Cela permet une meilleure adéquation entre l'espace de travail et son maillage. D'où une erreur moindre dans l'approximation entre l'espace calculé et l'espace réel.

8		16		32		64		128			256			512			1024
	12		24		48		96		144	192		288	384		576	768	

TAB. 4.3 – *puissances successives de* 8 *et* 12 *dans l'espace de travail.*

II.4.5 Problème de la surface d'implantation.

POUR UNE IMPLANTATION CABLÉE , l'accroissement de la surface nécessaire a deux origines :
– les opérations simples (additions, soustractions, échange des parties réelles et imaginaires) sont plus complexes, bien que du même ordre (multiplications par $\pm 1 \pm j$ pour la base 8 ou divisions par 2 pour la base 12). Les opérateurs correspondant ont une surface proportionnelle à la taille des opérandes pour une implantation à base d'opérateurs parallèles et constante pour une implantation à base d'opérateurs sériels.

– Le nombre de multiplications est augmenté de 50% pour une base 8 et des deux tiers pour une base 12. La complexité est proportionnelle à n^2 ou $n^2 \times \log n$ pour des opérateurs parallèles et à n pour des opérateurs sériels. L'influence des multiplieurs est donc prédominant et nous limitons notre étude à ceux-ci.

Nous avons calculé dans le paragraphe 4.4.2 le nombre total de multiplications d'une T.F.R. qui était divisé par 2 pour une base 8 et par 2.15 pour une base 12. Une architecture étalée implantant autant de papillons que de calculs élémentaires à effectuer, sa surface est proportionnelle au nombre total de multiplieurs mis en évidence ci-dessus. Ainsi les gains en surface sont les mêmes que ceux qui viennent d'être indiqués.

Dans une architecture repliée, seul un ensemble de papillons correspondant au calcul simultané d'une étape de la T.F.R. est implanté. Par rapport à une base 2, la surface est multipliée par le même facteur que le nombre de multiplieurs supplémentaires, 1.5 pour la base 8 et $(1 + \frac{2}{3}) \simeq 1,67$ pour la base 12. Dans ce type d'architecture, le gain est uniquement en vitesse de calcul et se paie en surface d'implantation.

II.4.6 Conclusion.

La structure des papillons de base 8 et 12 tels que nous les avons décrits permet de ne conserver qu'une étape de multiplications, à l'image des bases 2 et 4. Contrairement au passage de la base 2 à la base 4, il y a augmentation du nombre des multiplications par rapport à celles des papillons que ceux-ci remplacent. Malgré cela et bien que l'aspect des communications n'ait pas été abordé, l'utilisation d'une base égale à 8 ou 12 permet d'augmenter d'un facteur variant entre 2 et 3,6 les performances de calcul d'une T.F.R. dans des conditions proches pour ces deux nouvelles bases. Le choix de la base dépend du maillage désiré de l'espace de travail et le facteur d'amélioration dépend de l'implantation, logicielle ou cablée. La réduction du nombre d'étapes des diverses décompositions peut être mis à profit pour des implantations logicielles, dans des systèmes multiprocesseurs à cause des gains en vitesse de calcul et facilité de communications, et pour des implantations câblées, à cause des gains en vitesse de calcul et dans une moindre mesure en surface. Surtout valable dans le cas d'une architecture étalée, ce dernier point ne peut toutefois être mis à profit complètement d'une façon réaliste, à cause de l'importance de la surface nécessaire, bien que les architectures étalées présentent une meilleure utilisation de la surface d'implantation. Une architecture repliée peut toutefois se révéler intéressante dans les systèmes en temps réel où les impératifs de vitesse font que le surcoût en surface est oublié au profit du gain en temps de calcul lié à la réduction du nombre des étapes.

L'utilisation de bases mixtes prend désormais tout son sens. Le mixage des bases 12 et 2, ces dernières étant regroupées autant que possible en base 4 et 8, permet désormais d'atteindre un meilleur découpage de l'espace.

Signalons que dans un passé plutôt lointain des méthodes ont été proposées pour des bases 8 et 12. Nous n'avons pu retrouver ni les communications, ni même le nom des auteurs. La seule trace utilisable récupérée est le résultat chiffré des performances dans un ouvrage de traitement numérique du signal d'universitaires de Cambridge [PTF2s]. Citant les bases 4 et 8, ils indiquent un gain de 20 à 30%, avec en guise d'accompagnement le conseil de ne pas utiliser ces méthode à cause de leurs complexités. Des chiffres qui ne correspondent pas à notre approche, mais qui de toute manière ne devait viser que des approches logicielles.

Troisième partie

Cas particulier des matrices creuses

Chapitre 1

La méthode cristallographique

Sommaire

III.1.1 Introduction. **129**
III.1.2 Etude de la structure d'une molécule. **129**
III.1.3 La méthode cristallographique. **129**
III.1.4 Erreur dans une T.F.R. $3D$. **133**

III.1.1 Introduction.

L'ARCHITECTURE TRIDIMENSIONNELLE DES MACRO-MOLÉCULES peut désormais être connue avec une précision grandissantee grâce au développement des techniques expérimentales et à l'amélioration des technologies informatiques. Le paragraphe 1.2 nous permet de rappeler les différentes méthodes d'investigation dont la méthode cristallographique est l'une des plus performantes. Méthode précisée dans le paragraphe 1.3. Nous étudierons dans le paragraphe 1.4 l'influence de la présence d'une dimension supérieure à 1 sur les phénomènes d'erreur. Le gain négligeable qui en découle, nous amène dans le chapître suivant à nous tourner vers l'exploitation d'une des caractéristiques des structures moléculaires, la trés forte présence de valeurs nulles dans les données.

III.1.2 Etude de la structure d'une molécule.

LA MICROSCOPIE ET LA SPECTROSCOPIE (Résonance Magnétique Nucléaire, Infra-Rouges, . . .) permettent d'obtenir une précision variant du micromètre au dixième d'angström. Pour peu que le corps étudié ait une forme cristalline, il est possible d'améliorer ce facteur de quelques ordres de grandeur grâce à la méthode cristallographique. Le tableau 1.1 résume ces moyens d'investigation et indique la précision qui leur est associé. La méthode cristallographique a toutefois l'inconvénient actuel d'être beaucoup plus lente que ses concurrentes en raison du trés gros volume de calculs qu'elle nécessite [L. 73] [L. 77]. Réduire le temps de calcul correspondant est aujourd'hui le point le plus crucial à traiter pour favoriser l'utilisation de cette méthode et faire profiter de la précision de ses résultats [R. 78] [L. 76].

III.1.3 La méthode cristallographique.

TOUT CE QUI EST DÉCRIT CI-APRÉS RESTE VALABLE si les rayons X sont remplacés par des neutrons. Seuls la pénétration des rayons incidents et le pouvoir diffractant des éléments

techniques	précisions
microscope optique	quelques μm
microscope électronique	fraction de μm
spectroscopie	dixième d'Å
méthode cristallographique	de 10^{-2}Åà 10^{-4}Å

TAB. 1.1 – *Moyens d'étude et précision des résultats.*

agissant sur les rayons incidents sont différentes. Les neutrons permettent une étude des couches plus profondes que les rayons X. Les premiers sont diffractés par les noyaux des atomes constituant le cristal étudié (interaction neutron-noyau), les seconds par les électrons gravitant autour de ceux-ci (interaction photon-électron).

Un cristal est un matériau fortement ordonné. Ses propriétés physico-chimiques sont périodiquement répétées selon les trois dimensions de l'espace. Les trois vecteurs caractérisant cette périodicité définissent un volume élémentaire ou cellule de base. Les propriétés cristallographiques de la cellule de base font souvent apparaître elles-mêmes des symétries. Selon la théorie de la diffraction cinématique, les atomes de la cellule de base forment des familles de plans réfléchissant qui diffractent les faisceaux de rayons X lorsque la condition 1.1 est vérifiée.

$$2.d.\sin\Theta = n.\lambda \tag{1.1}$$

où :

- d est la distance entre deux plans cristallins successifs parallèles réfléchissants
- λ est la longueur d'onde des rayons X incidents
- Θ est l'angle entre la normale $\vec{H}$ d'un plan réfléchissant et le faisceau incident. Notons (h,k,l) les coordonnées de $\vec{H}$ selon les axes (x,y,z).
- n l'ordre de diffraction.

La figure 1.1 résume cette situation.

Les photons X qui sont transmis, réfractés, vers les plans suivants subissent le même phènomène. L'intensité qui pénètre à l'intérieur du cristal, si elle vérifie la relation 1.1 pour être diffractée, diminue rapidement à travers les différents plans successifs, jusqu'à ce que les rayons diffractés qui en résultent deviennent indétectables par l'électronique de mesure. Seules les premières couches du cristal sont donc accessibles par ce moyen d'investigation, typiquement une profondeur de 1000 Åpour un cristal de silicium de $300\mu m$ d'épaisseur.

Selon la théorie de la diffraction, la probabilité de présence des électrons dans un cristal qui est représentée par sa densité électronique notée $\rho(\vec{r})$, est obtenue par la transformée de Fourier des facteurs de structure notés F. Les coordonnées (X,Y,Z) sont les composantes du vecteur $\vec{r_j}$ représentant la position de l'atome j dans l'espace. Ce qui nous donne :

$$\rho(X,Y,Z) = \sum_{plans\ (h,\ k,\ l)} F.e^{-2\pi i(h.X+k.Y+l.Z)} \tag{1.2}$$

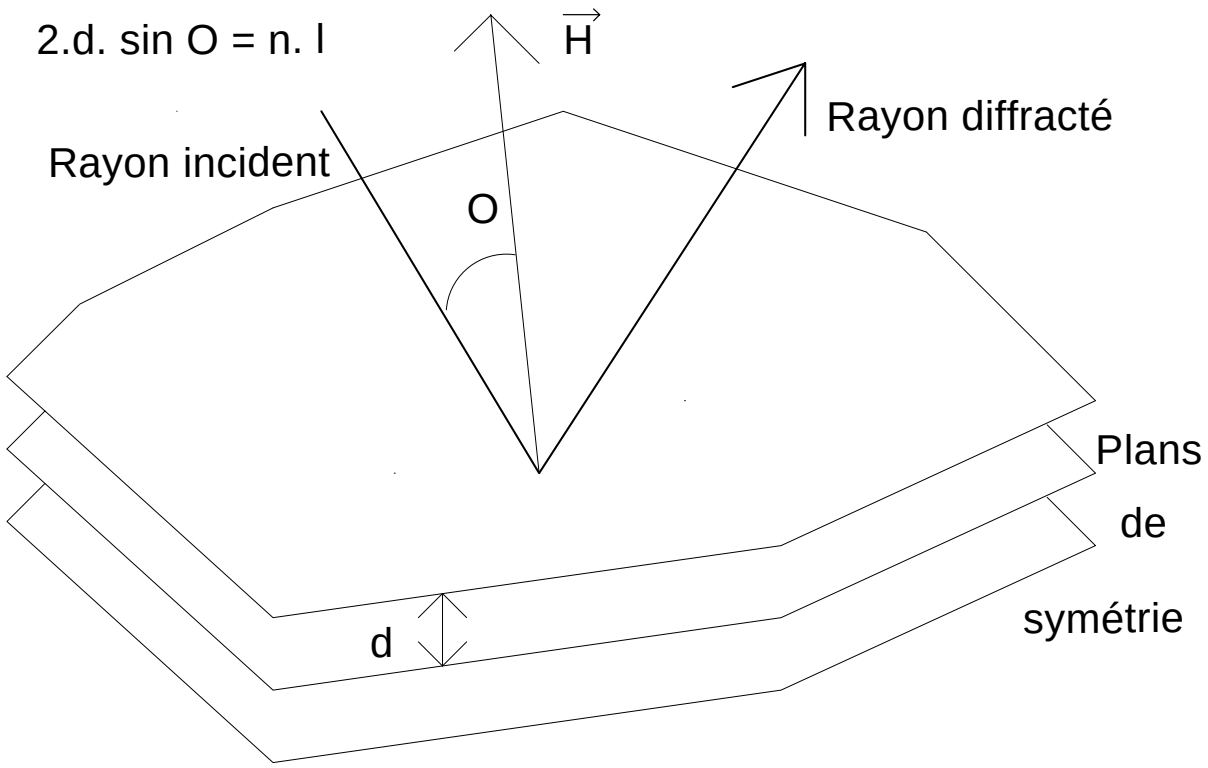

FIG. 1.1 – *phénomène de diffraction.*

Utilisant la réversibilité de la transformée de Fourier, nous pouvons écrire :

$$F_{\text{calculés}} = \sum_{\vec{r}} \rho(\vec{r}).e^{2\pi i(h.X+k.Y+l.Z)} \tag{1.3}$$

ou encore :

$$F = \sum_{atomes\ j} f_j.e^{2\pi i(h.X+k.Y+l.Z)}.t_{fj} \tag{1.4}$$

où :

- f_j est le pouvoir diffractant de l'atome j,
- t_{fj} est le facteur de température de l'atome correspondant qui traduit son mouvement dû à l'agitation thermique autour de sa position d'équilibre.

Parce que le détecteur électronique est sensible à l'intensité du rayonnement, c'est à dire à l'energie de celui-ci, seule l'amplitude de l'onde diffractée est accessible par une expérience de diffraction, comme indiqué par la relation 1.5. La phase de l'onde diffractée, nécessaire pour connaître complètement F, reste inconnue.

$$I_{diffractée} = L.p.|F.F^*| = L.p.|F|^2 \tag{1.5}$$

où apparaissent deux fonctions de l'angle Θ :

- L qui est le facteur de Lorentz qui dépend de la méthode utilisée pour la mesure,

– p qui est le facteur de polarisation des rayons incidents par le plan cristallin.

Les photons X qui arrivent sur le capteur de détection créent un signal qui est proportionnel au nombre de photons qui traversent le point de l'espace où le capteur se trouve, à condition que le détecteur ne présente pas de temps mort dû à une intensité trop forte. Dans le cas contraire, un atténuateur est plaçé entre le cristal et le détecteur. Ce signal est usuellement désigné par l'expression « nombre de coups reçus ». Chaque mesure, effectuée avec une direction des rayons X et une position du cristal et du capteur données, correspond à un vecteur $\vec{H}$ donné.

Quelques ordres de grandeur sont donnés dans le tableau 1.2 pour des composés type de ce genre de manipulation.

cristal de	nombre de plans (h,k,l)
silicium	quelques centaines
composé organique	quelques milliers
protéine	quelques dizaines de milliers

TAB. 1.2 – *corps étudiés et données de mesure.*

Par des méthodes statistiques ou par la méthode d'auto-corrélation, nous estimons un ensemble des φ, phases de F, telles que que $F = |F|.e^{i.\varphi}$. Puis, avec les $|F_{observé}|$ associés aux plans (h,k,l) nous calculons une première estimation des différentes valeurs $\rho(X,Y,Z)$ avec la relation 1.2.

Remarquons que la composition chimique du cristal doit être connue par ailleurs pour déterminer les différentes valeurs possibles de ρ qui sont des nombres réels. Nous découpons l'espace en plans parallèles et, sur chaque plan obtenu, nous traçons les courbes d'iso-densité électronique comme décrit par la figure 1.2. Notons que le tracé de ces courbes tient compte des quelques plans voisins les plus proches.

La position des pics de densité électronique en est déduite. Celles-ci nous donnent une nouvelle série de (X,Y,Z). La fonction $\rho(X,Y,Z)$ calculée avec les nouvelles coordonnées est en principe plus précise et contient des informations supplémentaires pour la détermination de la structure de la molécule étudiée. La transformée de Fourier inverse appliquée à travers la relation 1.3 à cette nouvelle approximation des valeurs réelles de $\rho(X,Y,Z)$ nous sert à calculer les valeurs du facteur de structure $F_{calculés}$.

La valeur des phases de $F_{calculés}$ sont attribuées aux valeurs correspondantes des $|F_{observés}|$, puis nous réitérons les calculs précédents avec ces nouvelles données. Elles nous permettent de converger, plus ou moins rapidement suivant la précision des $\varphi_{initiales}$ et la complexité du cristal, vers une représentation spatiale de la densité électronique du cristal étudié. Périodiquement, l'écart entre les $F_{observés}$ et les $F_{calculés}$ est estimé pour juger de cette convergence et décider de continuer ou d'arrêter ce calcul.

Cette dernière opération est trés subtile et constitue le point le plus délicat de ce genre de problèmes outre la masse des calculs nécessaires. La méthode nous donne une représentation tri-dimensionelle de l'architecture de la molécule étudiée comme représenté par la figure 1.3.

Parmi les différentes étapes qui constitutent l'implantation de la méthode cristallographique, le calcul des transformées de Fourier directes et inverses est le plus grand consommateur en temps d'exécution. Par conséquent, nous pouvons focaliser notre attention sur ce point précis pour améliorer la mise en oeuvre de cette méthode.

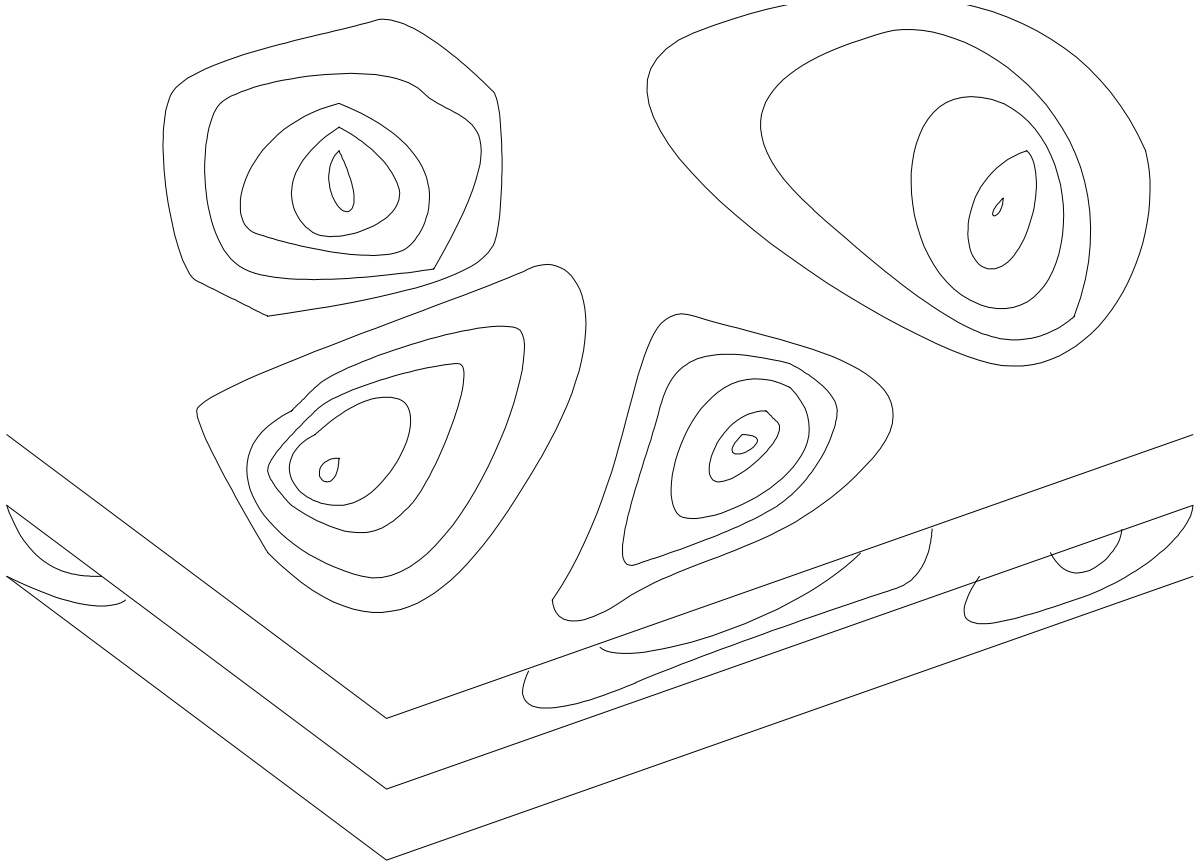

FIG. 1.2 – *plans de l'espace et courbes d'iso-densité.*

III.1.4 Erreur dans une T.F.R. $3D$.

NOUS AVONS PU ÉTUDIER dans le chapître 2 les phénomènes d'erreur dans une T.F.R. monodimensionnelle. Nous allons voir l'influence que peut avoir le fait de travailler dans un espace tridimensionnel. Par application des différentes lois mises en évidence précédemment, nous pouvons reécrire l'intégration des erreurs en sachant que :

- les phénomènes observés se reproduisent sans considération de dimension pour la propagation des erreurs à travers les diverses étapes qui traitent les N échantillons

- les sources sont semblables pour chaque dimension à celles d'une T.F.R. dont la taille serait celle de la dimension considérée, sachant qu'il y a trois dimensions.

Nous pouvons en déduire les règles suivantes :

- le problème des erreurs dûes aux divisions par deux sont identiques quelle que soit la dimension de l'espace de travail,

- les sources d'erreur concernant les multiplications sont à intégrer sur chaque dimension en tenant compte des étapes où elles n'apparaîtraient pas dans la T.F.R. monodimensionnelle équivalente.

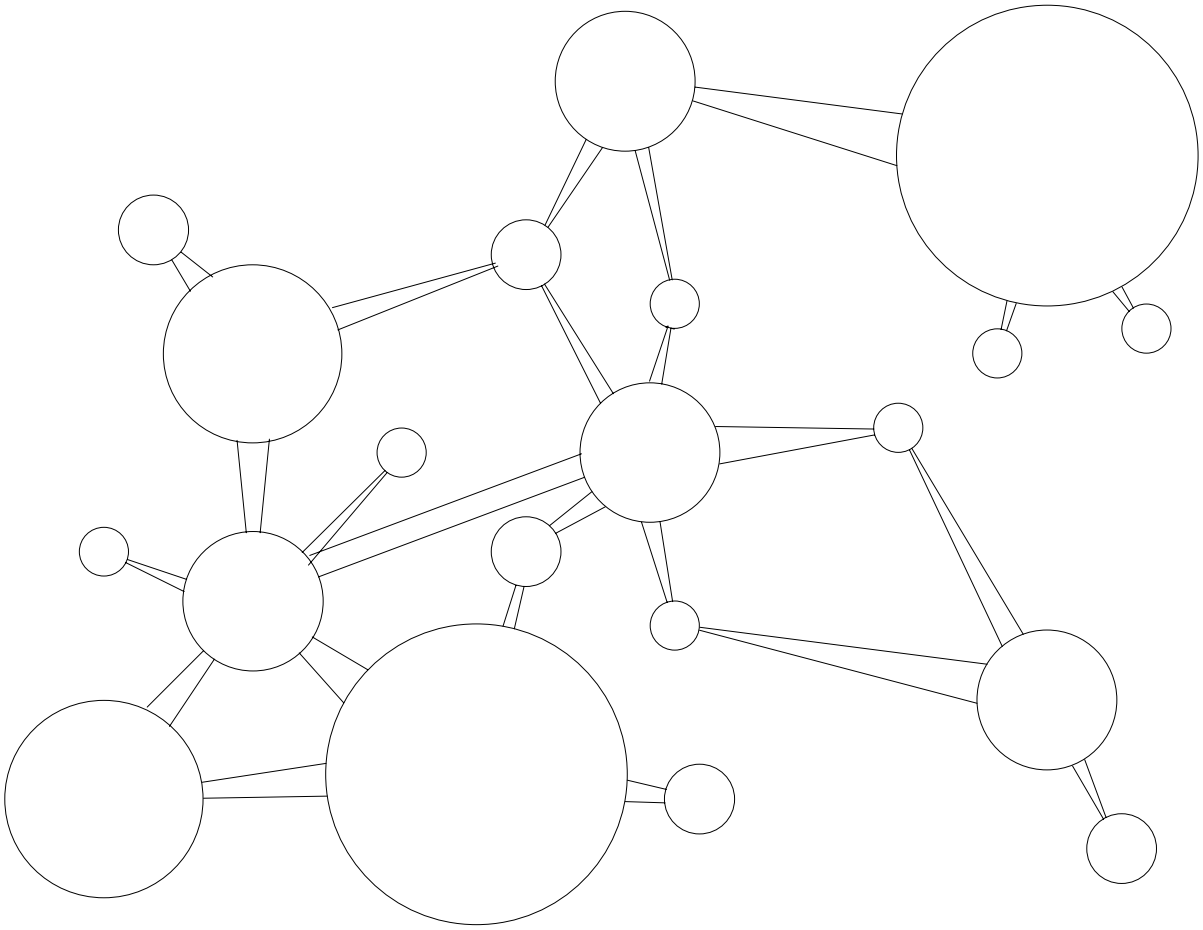

FIG. 1.3 – *exemple d'architecture tri-dimensionnelle obtenue.*

Nous nous contentons de traiter le cas d'un espace dont toutes les dimensions ont le même découpage pour être discrétisé et d'un entrelacement temporel.

$$\sum_{i=0}^{N-1} e_{tpda} = \sum_{m=3}^{\frac{M}{3}} \sum_{m=\frac{M}{3}+3}^{\frac{2M}{3}} \sum_{m=\frac{2M}{3}+3}^{M} \frac{\alpha.2^{-2B}}{12}.\left(N \Leftrightarrow 4.2^{M-m}\right).2^{M-m} \tag{1.6}$$

$$= \left[\sum_{m=3}^{M} \frac{\alpha.2^{-2B}}{12}.\left(N \Leftrightarrow 4.2^{M-m}\right).2^{M-m}\right]$$

$$\Leftrightarrow \left[\frac{\alpha.2^{-2B}}{12}.\left(N \Leftrightarrow 4.2^{M-m}\right).2^{M-m}\right]_{m_e} \tag{1.7}$$

Nous avons :

$$m_e = \{\frac{M}{3}+1, \frac{M}{3}+2, \frac{2M}{3}+1, \frac{2M}{3}+2\}$$

Ce qui nous donne :

$$\sum_{i=0}^{N-1} e_{tpda} = \frac{\alpha 2^{-2B}}{12}\left(\frac{N^2}{6} \Leftrightarrow N + \frac{4}{3}\right) \Leftrightarrow \left[\frac{\alpha.2^{-2B}}{12}.\left(N \Leftrightarrow 4.2^{M-m}\right).2^{M-m}\right]_{m_e}$$

$$\Leftrightarrow \sum_{i=0}^{N-1} e_{tpda} = \frac{\alpha 2^{-2B}}{12}\left(\frac{N^2}{6} \Leftrightarrow N + \frac{4}{3}\right) \Leftrightarrow \frac{\alpha.2^{-2B}}{12}.\left[\frac{3}{4}\sqrt[3]{N^5} \Leftrightarrow \frac{1}{2}\sqrt[3]{N^4} \Leftrightarrow \frac{5}{4}\sqrt[3]{N^2}\right] \quad (1.8)$$

Si nous nous bornons aux plus fortes puissances de N, nous obtenons un terme en $\left(\frac{N^2}{6} \Leftrightarrow \frac{3}{4}\sqrt[3]{N^5}\right)$ qui n'est atténué par rapport à $\frac{N^2}{6}$ que pour de faible valeurs de N. Le fait qu'une T.F.R. $3D$ puisse être une succession de T.F.R. $1D$ n'a qu'une influence négligeable sur l'erreur totale, du moins par le biais de la modélisation des erreurs que nous avons utilisée.

Chapitre 2

Erreurs dans une T.F.R. d'une matrice creuse

Sommaire

III.2.1 Spécificités de la reconstitution de données 137
III.2.2 Influence des données nulles sur la précision 138
III.2.2.1 Pas de débordement 138
III.2.2.2 Débordements possibles 138
III.2.3 Conclusion. 140

III.2.1 Spécificités de la reconstitution de données

Une T.F.R. qui apparait en tant que l'élément unique d'un traitement impose à la précision d'être un paramètre incontournable. La méthode cristallographique revient à reconstituer un ensemble complet de données, module et phase de F, à partir d'un sous-ensemble partiel, uniquement le module. Dans ce genre de problème il y a convergence des résultats vers une solution stable. Cette convergence est plus ou moins rapide selon :

- la validité des hypothèses de départ qui sont le reflet des connaissances et de l'habileté de l'utilisateur. Certaines zônes de la structure peuvent être identifiées comme étant identiques ou ressemblant à des motifs d'autres molécules déja connues.
- La précision des calculs qui ne vise pas la précision finale à laquelle ce procédé aboutit, mais la quantité d'information reconstituée à chaque étape de l'itération.
- La finesse du maillage permettant de numériser l'espace de travail.

L'utilisation d'opérandes de faible taille permet d'augmenter la puissance de calcul pour une surface d'implantation donnée, en nombre d'opérations par seconde, au détriment de la précision des calculs. Cela revient certes à augmenter le nombre d'itérations, mais aussi à calculer chacune d'entre elles en un temps beaucoup plus court. Différentes considérations, essentiellement expérimentales, permettent de justifier un gain global [VQ95a], sans qu'il soit possible dans l'état actuel de la connaissance de modéliser à l'heure actuelle et d'une façon extrêment précise le comportement de la méthode en fonction des données.

Une T.F.R. multidimensionnelle est synonyme d'un grand nombre d'échantillons, mais dans notre cas un pourcentage important d'entre eux a une valeur nulle. Ce qui réduit en conséquence

l'erreur du calcul, en particulier par le nombre de dépassements possibles de la capacité des opérandes. S'il est envisageable de réduire le nombre de coefficients à calculer en détectant les valeurs nulles, techniques de calcul sur des matrices creuses [Gue87] [Jut87], cela exige le développement de circuits ou d'architectures spécifiques [Law93], plutôt que l'utilisation des moyens existant tels que ceux décrit au paragraphe 2.5.3. Nous utilisons donc cette remarque uniquement pour ses conséquences heureuses sur l'erreur apparaisant au cours d'une des T.F.R. du traitement. Une diminution de celle-ci augmente la vitesse de convergence, donc vient contrebalancer les effets dûs au choix de limiter la taille des opérandes.

III.2.2 Influence des données nulles sur la précision

NOUS CONSIDÉRONS SUCCESSIVEMENT dans cette étude les deux cas typiques d'une solution à virgule glissante [VQ95b] : pas de débordement de capacité des opérandes au cours des étapes successives de la T.F.R. et division à la demande par 2 des résultats intermédiaires pour éviter un tel débordement. Nous nous limitons au multiplieur arrondissant ses résultats, plus précis que celui tronquant les siens.

III.2.2.1 Pas de débordement

DANS CE CAS, AUCUNE DIVISION PAR 2 n'est exécutée au cours du calcul, donc aucune erreur n'est engendrée par ce genre d'opérations. Notons p_m la probabilité qu'un résultat intermédiaire de l'étape m de la T.F.R. soit égal à zéro. La puissance équivalente de bruit générée par l'arrondi des résultats des multiplications de l'étape précédente ayant donné naissance à ce résultat doit être pondérée par la probabilité $(1 \Leftrightarrow p_m)$ qu'il ne soit pas nul. Nous avons la relation : $p_m = p_{m-1}^2$. Cela nous donne :

$$\begin{aligned} \sum_{i=0}^{N-1} e_i &= \sum_{m=3}^{M} \underbrace{\frac{\alpha 2^{-2B}}{12}}_{v} \cdot \underbrace{(N \Leftrightarrow 4.2^{M-m})}_{e} \cdot \underbrace{2^{M-m}}_{c} .(1 \Leftrightarrow p_m) \\ &= \frac{\alpha 2^{-2B}}{12} N^2 \sum_{m=3}^{M} (2^{-m} \Leftrightarrow 4.2^{-2m}).(1 \Leftrightarrow p_1^{2^{m-1}}) \end{aligned} \tag{2.1}$$

Le graphique 2.1 illustre l'influence du pourcentage d'échantillons non nuls dans une T.F.R. sur l'erreur globale de celle-ci. L'intégration a été faite pour un nombre d'échantillons égal à 10^6, mais les résultats sont sensiblement identiques pour des valeurs différents. La courbe tracée représente en fait le coefficient qu'il faut multiplier au terme $\frac{\alpha 2^{-2B}}{12} N^2$. En sachant que la valeur de ce coefficient, s'il n'y a pas d'échantillon nul, est proche de $\frac{1}{6}$, soit 0.167, et qu'il représente la valeur asymptotique de l'erreur telle qu'elle a été rappelée au chapître 2.

Nous rappelons que les applications considérées sont dans l'espace. La T.F.R. est multidimensionelle, donc le nombre d'échantillons est trés grand. Un million est un nombre raisonnable compte-tenu de la valeur que peut souhaiter un utilisateur et celle dont il se contente actuellement compte-tenu de la longueur des calculs.

Nous pouvons conclure que l'erreur est beaucoup plus faible avec un tel ensemble d'échantillons à traiter et que nous pouvons donc considérer une telle solution comme valable. Indiquons que les échantillons non nul ont une dynamique relativement faible et que l'ordre de grandeur de leur nombre a été indiqué pour différents cas de figure dans le tableau 1.2.

III.2.2.2 Débordements possibles

UN DÉBORDEMENT EST ÉVITÉ par une division par 2 des résultats intermédiaires en cas de

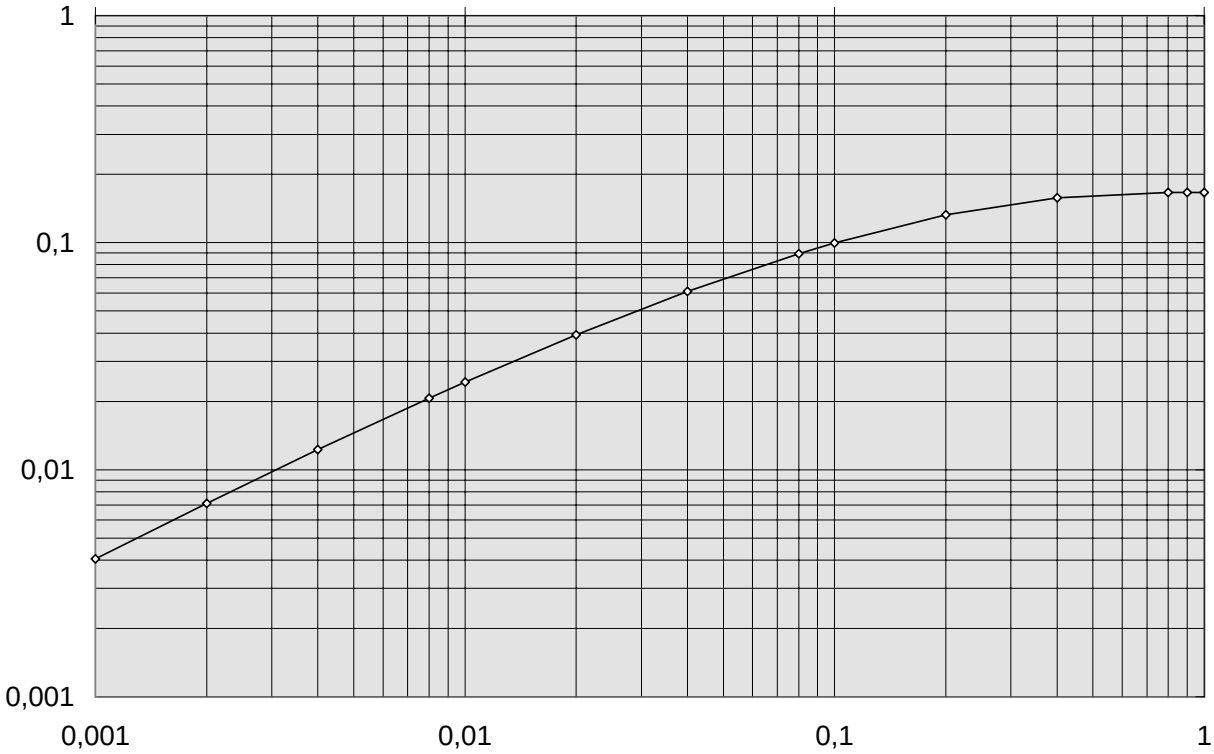

FIG. 2.1 – *influence sur l'erreur de la présence d'échantillons nuls.*

nécessité. Cela augmente l'erreur globale du calcul de la T.F.R., d'où l'intérêt d'avoir un nombre de divisions successives le plus faible possible. S'il y a possibilité de débordement, il y a nécessité d'une division par 2. La probabilité de ce fait est difficile à établir en raison de la présence de coefficients exponentiels. Dans le cas où ils seraient égaux à un, la probabilité serait celle d'un dépassement dans une addition. Elle est égale au produit de convolution des densités de probabilité de la valeur des deux signaux et vaut $0,125$ dans le cas où la valeur nulle serait aussi probable que les autres valeurs possibles pour les signaux d'entrée. Pour ne pas à avoir à intégrer toutes ces conditions, nous avons considéré que la probabilité d'un dépassement était égale à celle d'avoir une entrée non nulle. Cela augmente trés fortement cette probabilité.

Nous avons représenté dans le graphique 2.2 la probabilité que l'éventualité d'un débordement soit détecté au cours des étapes successives du calcul pour différents pourcentages d'échantillons nuls. Le pourcentage d'échantillons non nuls est le paramètre de cette famille de courbes. Nous pouvons déduire de l'étude de ce graphique que pour un faible pourcentage d'échantillons non nuls la probabilité d'un débordement est fortement réduite pour les premieres étapes et que ce phénomène se produit sur d'autant plus d'étapes que le nombre d'échantillons non nuls est faible en pourcentage. Qui dit faible risque de dépassement dit faible erreur crée par division par 2, donc calcul précis avec des opérateurs traitant des nombres de petite taille. Nous démontrons par ce biais l'intérêt d'utiliser des opérateurs cablés, éventuellement ceux existant déja au niveau industriel et même s'ils ne traitent que des nombres de faible taille.

Il faut remarquer avant d'envisager l'utilisation de cette propriété que le nombre d'échantillons non nuls est une valeur fixée par la mesure, mais que le nombre total d'échantillons est seulement fonction de la finesse du maillage de la portion de l'espace étudié. Le changement de la valeur du pas de maillage de l'espace peut permettre de se trouver dans le cas favorable évoqué ci-dessus.

La correspondance entre les coordonnées réelles et les valeurs discrètes qui leur sont associées est évidemment meilleure si le maillage est plus fin. Ceci est une source d'erreur qui est rarement prise en compte, car trés difficile à modéliser. Cette voie permet donc d'atténuer ce phénomène sans pénalisation au niveau du temps de calcul.

Cette propriété est donc exploitable directement dans le cas où elle existe avec le nombre d'échantillons choisis au départ ou en augmentant ce nombre dans le cas contraire.

Les étapes ultérieures du calcul peuvent faire apparaître des erreurs intolérables en raison de la taille des opérandes, mais ce phénomène peut être évité si nécessaire. Cela demande de séparer les premières étapes du calcul des suivantes pour concilier la rapidité des opérateurs pouvant traiter les premières et la précision de ceux devant traiter les suivantes.

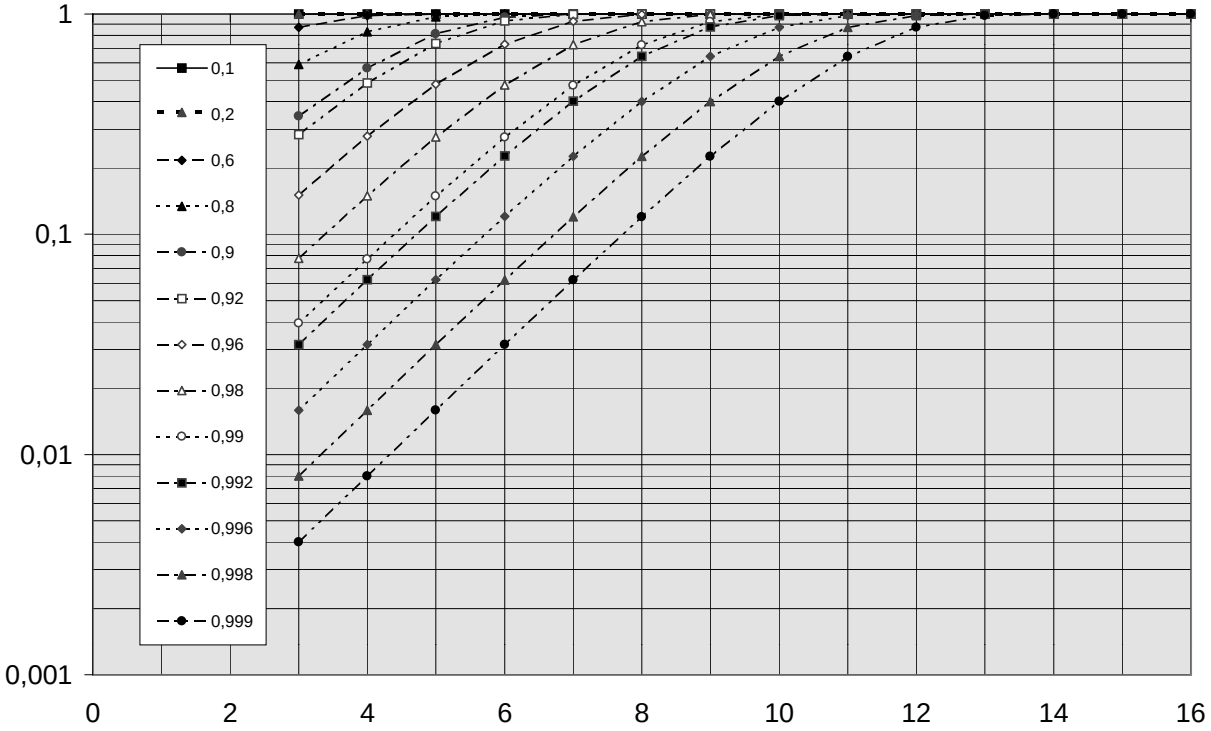

FIG. 2.2 – *influence sur le risque de dépassement de la présence d'échantillons nuls.*

III.2.3 Conclusion.

L'accélération des T.F.R. mises en oeuvre dans la méthode cristallographique au moyen d'opérations arithmétiques sur des opérandes de faible taille (16 chiffres binaires) est une voie pour développer et répandre celle-ci. Une optimisation de cette voie amène à se demander s'il est possible d'envisager une itération en plusieurs étapes, avec des tailles d'opérande croissantes pour répondre à l'importance croissante d'informations que contient le modèle calculé. Ceci pour utiliser des opérateurs cablés de T.F.R. dont la vitesse de calcul est inégalable, mais dont la surface d'implantation est importante.

Dans l'attente d'une telle possibilité, les applications présentant de nombreuses valeurs nulles dans leurs échantillons et de faible dynamiques de valeurs dans leurs données initiales comme la méthode cristallographique feront appel à des opérateurs existants semblables à ceux dont nous

rappelons l'existence dans le chapître 2.

Chapitre 3

Avant de sombrer dans l'oubli

PARTI DE RIEN ET ARRIVÉ NULLE PART, MAIS TOUT SEUL. Cette devise que les mauvaises langues se plaisent à coller au dos de beaucoup est souvent immérité. Dans de nombreux cas, il a fallu se mettre à plusieurs pour y arriver, mais certains coups de main sont, il est vrai, aussi discrets dans les autobiographies de leurs auteurs qu'efficaces lors de leur réalisation. Ceux qui sont philosophes devant la vie prétendent toutefois que nulle part est quand même mieux que rien, puisqu'on y rencontre beaucoup de monde. Certains vont même jusqu'à prétendre que là se trouve la raison qui pousse ceux qui s'y trouvent à vous y attirer en croyant vous y pousser.

L'augmentation de la taille des opérandes est une solution à privilégier pour une implantation logicielle, car gérer la nécessité d'une division est couteuse en terme de temps d'exécution et les fabriquants de processeurs ont adopté cette démarche. La tendance récente des ordinateurs généralistes est de proposer des instructions aussi rapides pour des opérandes de 8 à 32, voire 64 chiffres binaires. L'utilisation de grands nombres permet dans ce cas un gain en précision sans à avoir à optimiser la taille des nombres utilisés pour obtenir la meilleure vitesse. Toutefois certains constructeurs de microprocesseurs (Cyrix par exemple pour le monde des compatibles IBM-PC) ont implantés des opérations arithmétiques rapides pour des nombres d'une faible taille (16 chiffres binaires), pour des applications de multimédia notamment. Politique reprise par Intel pour les instructions MMX de sa toute prochaine génération de microprocesseur Pentium.

Le problème du calcul de T.F.R. avec des opérandes d'une faible taille reste donc posé dans des applications où le nombre des échantillons est grand, ce qui est le cas des T.F.R. multidimensionnelles. En effet les erreurs de calcul d'une T.F.R. se propagent aux différents coefficients aux cours des étapes successives. Ce qui peut être mis à profit pour concevoir des architectures à hautes performances, mais chères, car les circuits de base eux-mêmes doivent être développés. Le rapport signal-bruit résultant est donc d'autant plus dégradé que le nombre d'échantillons est important.

Les gains qui peuvent être obtenus par le concepteur d'une architecture dédiée à la T.F.R. ne peuvent malheureusement pas être miraculeux. Toutefois, nous avons pu explorer au cours de cette thèse différentes voies qui permettent un gain significatif, mis bout à bout. Ils ont toutefois l'inconvénient d'exiger des surfaces d'implantation conséquentes. L'idéal serait de pouvoir implanter des architectures massives d'opérateurs sériels à taille croissante pour des T.F.R. mixtes en base 8 et 12.

Certaines de ces techniques peuvent cependant être utilisées à plus faible échelle de réalisation. Que ce soit en ce qui concerne des opérateurs parallèles ou sériels à taille croissante pour de faible nombre d'échantillons, dans une application isolée ou en tant que papillon d'une T.F.R. ayant une macrobase. Un faible nombre d'échantillons pour une T.F.R. est souvent également une forte

valeur de la macrobase d'une autre transformée.

Le développement de tels circuits de calcul ou le choix d'utiliser des composants réalisés par ailleurs amène à l'emploi des transformations $1D \rightarrow 2D$ dont nous avons rappelé le principe et des architectures à saturation de bus que nous avons décrites. Il s'agit d'ailleurs de la solution la plus réaliste pour développer à l'intention d'utilisateurs finaux des architectures dédiées présentant suffisamment de hautes performances pour séduire un public demandeur, mais hésitant qui pourrait appuyer le développement de telles ambitions scientifiques..

Index

Il ne s'agit que d'un bref index permettant de retrouver les principales définitions introduites ou rappelées dans cette thèse que vous retrouverez facilement dans la page indiquée en vous souvenant qu'elles ont été mises en italique, sauf oubli de ma part.

architecture à saturation de bus, 78

base arithmétique, 25
bases mixtes, 23
Borrow Save Notation, 58

chemin de donnée, 43
chiffre binaire signé, 58
CMOS, 60
PRQ un chiffre par pas, 95

full-custom, 65

latence, 43

macrobase, 24
MMP, 59
Multi-Chip Modules, 37

opérateur en ligne, 54

papillons, 13
perfect shuffle, 31
pipeline, 47
pipeline longitutinale, 47
pipeline transversale, 47
PPM, 59
pseudo-base, 24

saturation de bus, 78
solution avec division systématique par 2, 88
solution sans débordement par addition, 88

Transformée de Fourier Rapide, 13

virgule glissante, 25, 89

Bibliographie

[AR75] N. Ahmed and K. R. Rao. *Orthogonal Transforms for Digital Signal Processing.* Springer Verlag, 1975.

[Avi61] Avizienis. Signed-digit number representation for fast parallel arithmetic. *IEEE Transactions on Electronic Computers*, 10 :389–400, September 1961.

[Bel96] M. Bellanger. *Traitement numérique du signal.* CNET-ENST. Masson, 5 edition, 1996.

[BGW93] R. Bouraoui, A. Guyot, and G. Walker. On-line operator for Euclidean distance. In *EDAC-EUROASIC93*, pages 192–195, Paris, France, 1993.

[Col91] Coles, Joe. Le DSP spécifique : une solution économique parfaite? *Electronique*, (12) :78–80, November 1991.

[CS90] K. Chen and C. Svensson. A 512 processor array for video/image processing. In optionnel, editor, *From pixel to features II*, pages 349–361, Bonas, France, aug 1990. optionnel, optionnel. optionnel.

[D. 75] D. V. James. Quantization errors in fast Fourier transform. *IEEE Trans. Acoustics, Speech, Signal Processing*, 23 :277–283, June 1975.

[DJT+92] H. Delori, J.F.Paillotin, K. Torki, A. Chagoya, C. Garnier, F. Martin, and B. Courtois. French MPC Annual Report. Technical report, CMP, 46 avenue Félix Viallet, 38 031 Grenoble Cedex, France, 1992.

[DNJ90] J.C. Dufourd, J.F. Naviner, and F. Jutand. Preform : A process independent symbolic layout system. In *ICCAD*, Santa-Clara, USA, nov 1990.

[*et al.*84] N. U. Chowdary *et al.* A high speed two dimensional FFT processor. In *Int. Conf. Acoust., Speech, Signal Processing*, pages 4.11.1–4.11.4, San Diego, California, USA, 1984.

[F. 91] F. Grosvalet. Réaliser un calcul de FFT complexe pour l'analyse de signal. *Electronique*, (12) :71–73, November 1991.

[G. 71] G. U. Ramos. Roundoff error analysis of the fast Fourier transform. *Math. Comput.*, 25 :757–768, October 1971.

[G. 83] G. Bongiovanni. Two VLSI Structures for the Discret Fourier Transform. *IEEE Transactions on Computers*, 32(8) :750–753, August 1983.

[GK91] Alain Guyot and Yustina Kusumaputri. Ocapi : prototype for high precision arithmetic. In *VLSI'91*, pages 1.2.1–1.2.8, Edinburgh, Great-Britain, aug 1991.

[GKTMN85] A. Greiner, C. Kara-Terki, H. Mehrez, and G. Noguez. A flexible high performance serial radix 2 fft butterfly arithmetic unit. In *ESSIRC*, pages 25–28, 1985.

[GS87] I. Gertner and M. Shamash. Vlsi architectures for multidimensional fourier transform processing. *IEEE Transactions on Computers*, 36(11) :1265–1274, November 1987.

[Gue87] A. Guerin. *CRASY: un Calculateur de R seaux Adaptatifs SYstoloique, application au calcul neuromim tique.* PhD thesis, INPG, 1987.

[J. 91] J. Pauthion. Un circuit spécifique pour une FFT à plus de 20 MHz. *Electronique*, (12) :73–78, November 1991.

[Joh84] J. A. Johnston. Generating multipliers for a radix-4 parallel fft algorithm. *Signal Processing*, (6) :61–66, 1984.

[Jut87] C. Jutten. *Calculs neuromim tiques et traitement du signal : analyse en composantes ind pendantes.* PhD thesis, INPG, 1987.

[J.W65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* , 19 :297–301, April 1965.

[Kus93] Y. Kusumaputri. *Op rateurs Arithm tiques Standards En Ligne tr s Grande Pr cision.* T se inpg, INPG, Grenoble, France, May 1993.

[L. 73] L. F. Ten Eyck. Crystallographic Fast Fourier Transforms. *Acta Crystallography*, (29) :183–191, 1973.

[L. 76] L. F. Ten Eyck and L. H. Weaver and B. W. Matthews. A method of obtaining a stereochemically acceptable protein model wich fits a set of atomic coordinates. *Acta Crystallography*, (23) :349–350, 1976.

[L. 77] L. F. Ten Eyck. Efficient Structure-Factor Calculation for Large Molecules by the Fast Fourier Transform. *Acta Crystallography*, (33) :486–492, 1977.

[Law93] J.C. Lawson. *SMART: Un Neurocalculateur parall le exploitant des Matrices Creuses.* PhD thesis, INPG, 1993.

[LS93] P. Larsson and C. Svensson. A serial-parallel multiplier for a narrow pitch layout. In *EDAC'93*, pages 191–196, Paris, France, Feb 1993. IEEE Computer Society Press.

[Lyo76] R. F. Lyon. Two's complement pipeline multipliers. *Transactions on Communications*, pages 418–425, April 1976.

[Lyo81] R. F. Lyon. A bit-serial vlsi architecture methodology for signal processing. In J. P. Gray, editor, *VLSI*, Edinburgh, 1981. Academic Press.

[Lyo84] R. F. Lyon. *VLSI Signal Processing*, section A Bit-Serial Multiprocessor for Signal Processing. IEEE Press, 1984.

[M.D77] M.D. Ercegovac and K.S. Trivedi. On-line algorithms for division and multiplication. *IEEE Transactions on Computers*, 26(7) :681–687, July 1977.

[Meh83] H. Mehrez. *Etude et classification des algorithmes de la Transformation de Fourier Rapide. Application la conception d'un circuit Papillon FFT int egr e : unit e arithm etique de base d'un processeur FFT.* Thèse de troisième cycle, Universit e de Paris-Sud, Orsay, France, June 1983.

[Meh91] H. Mehrez. *Des Architectures VLSI Pipelines pour les Algorithmes Num eriques à flots de Donn ees en Repr esentations Arithm etiques Virgule Fixe et Virgule Flottante.* Thèse d' etat, Universit e Pierre et Marie Curie, Paris, France, July 1991.

[ND92] J.F. Naviner and J.C. Dufourd. Preforme/agape : a synergy between symbolic cell design and assembly. In *EUROMICRO*, Paris, 1992.

[OVS94] V. G. Oklobdzija, D. Villeger, and T. Soulas. An integrated multiplier for complex numbers. *Journal of VLSI Signal Processing*, (7) :213–222, 1994.

[Par80] Parker, D. Stott, Jr. Notes on shuffle/exchange-type schwitching networks. *IEEE Transactions on Computers*, 29(3) :213–222, March 1980.

[PTF2s] W. H. Press, S. A. Teukolsky, and W. T. Vetterling B. P. Flannery. *Numerical Recipes in Fortran, The Art of Scientific Computing.* Cambridge University Press, 2 edition, 1992s.

[R. 78] R. C. Agarwal. A New Least-Squares Refinement Technique Based on the Fast Fourier Transform Algorithm. *Acta Crystallography*, (34) :791–809, 1978.

[SBGM92] A. Skaf, J.C. Bajard, A. Guyot, and J.M. Muller. On-line approximation of real functions using polynomials. In *ICM'92*, Monastir, Tunisia, dec 1992.

[SG93] A. Skaf and A. Guyot. VLSI design of on-line add/multiply algorithms. In *ICCD'93*, pages 264–267, Cambridge, Massachusetts, USA, oct 1993. IEEE Computer Society and IEEE Circuits and Systems Society, IEEE Computer Society Press.

[Ska95] A. Skaf. *Conception de Processeurs arithm tiques en ligne.* T se inpg, INPG, Grenoble, France, September 1995.

[Sto71] Stone, Harold S. Parallel Processing with the Perfect Shuffle. *IEEE Transactions on Computers*, 20(2) :153–161, February 1971.

[TTL76] Trâń-Thôńg and B. Liu. Fixed-point fast fourier transform error analysis. *IEEE Trans. Acoustics, Speech, Signal Processing*, 24(6) :563–573, December 1976.

[TYY86] N. Takagi, H. Yasuura, and S. Yajima. High-speed vlsi multiplier algorithm with a redundant binary addition tree. *IEEE Transactions on Computers*, 34(9) :789–796, September 1986.

[VA94a] A. Vacher and A.Guyot. A VLSI Implementation of Fast Fourier Transform for a Large Number of Samples. In *SPRANN'94*, Lille, France, May 1994. IMACS.

[VA94b] A. Vacher and A.Guyot. Error-Speed Compromise for FFT VLSI. In *SSST'94*, Athens, Ohio, USA, Mar 1994. IEEE Computer Society Press.

[VA95a] A. Vacher and A.Guyot. Radix-8 Butterflies for Folded FFT. In *SSST'95*, Starkville, Mississipi, USA, Mar 1995. IEEE Computer Society Press.

[VA95b] A. Vacher and A.Guyot. Spread and Folded Architectures for FFT. In *SSST'95*, Starkville, Mississipi, USA, Mar 1995. IEEE Computer Society Press.

[VBG+94] A. Vacher, M. Benkhebbab, A. Guyot, T. Rousseau, and A. Skaf. A VLSI Implementation of Parallel Fast Fourier Transform. In *EDAC-ETC-EUROASIC'94*, Paris, France, Feb 1994. IEEE Computer Society Press.

[Ver94] C. Verdier. *M moires parall les et r seau d'interconnexion pour architectures SIMD.* Thèse enst, ENST, Paris, June 1994.

[VQ95a] A. Vacher and D. Tran Qui. Hard-wired Computation of FFT : An Application to Cristallography. In *SSST'95*, Starkville, Mississipi, USA, Mar 1995. IEEE Computer Society Press.

[VQ95b] A. Vacher and D. Tran Qui. Hard-wired Computation of FFT : An Application to Cristallography. In *M2SABI'95*, Bruxelles, Belgique, May 1995. IMACS.

[Wei69] C. J. Weinstein. Quantization effects in digital filters. ASTIA Doc. AD-706862, MIT Lincoln Lab., nov 1969.

[Wel69] P. D. Welch. A fixed point fast fourier transform error analysis. *IEEE Trans. Audio Electroacoust.*, 17 :151–157, June 1969.

AUTORISATION DE SOUTENANCE

Vu les dispositions de l'arrêté du 30 Mars 1992 relatifs aux Etudes Doctorales

Vu les Rapports de présentations de :

Monsieur MEHREZ Habib

Monsieur DEMASSIEUX Nicolas

Monsieur VACHER André

est autorisé(e) à présenter une thèse en soutenance en vue de l'obtention du diplôme de **Docteur de l'Institut National Polytechnique de Grenoble,** spécialité MICROELECTRONIQUE.

Fait à Grenoble, le 07 JAN 1997

Bernard GUÉRIN
Vice-Président de l'INPG
Pédagogie et Formations

INPG - 46, avenue Félix Viallet - 38031 GRENOBLE CEDEX 1 - FRANCE - Tél. (33) 76 57 45 00 - Télex 320 305 F - Télécopie (33) 76 57 45 01 - 3615 code INPG

Résumé

Le calcul cablé d'une transformée de Fourier permet d'accélérer trés fortement son calcul. Des applications militaires ont vu des solutions pour de faibles nombres d'échantillons et avec des précisions limitées. Repousser ces barrières demande de diminuer la surface d'implantation. Un grand nombre de cellules de calcul, les papillons, utilisant des opérateurs sériels et travaillant en parallèle permet d'obtenir une meilleure précision et une forte vitesse. Le surcoût en surface a été vérifié au cours d'une implantation présentée avec ses perspectives. Une solutions multipuce impose le choix d'une architecture à deux niveaux, papillons sériels et bus de communication parallèles, dont l'un est privilégié au niveau taux d'utilisation et fréquence de travail. La précision est fonction de celles des données originales et du nombre d'étapes, donc d'échantillons. Des opérateurs à taille variable permettent de jouer sur la précision et la surface ou la vitesse selon le nombre de barettes de papillons implantées. Les paramètres des opérateurs optimisent l'architecture d'une transformée de Fourier pour une décomposition donnée de celle-ci. Les bases 2 et 4 sont les seules réellement utilisées pour la décomposition au niveau du calcul. L'estimation de la surface et du temps de calcul démontre un gain pour des solutions cablées pour les bases 8 et 12. Les transformées multidimensionnelles présentent un phénomène d'erreur plus faible, à nombre total d'échantillons égal, en raison du plus grand nombre de coefficients exponentiels simples. Celles-ci sont la cible des applications civiles à grand nombre d'échantillons, imagerie ou données dans l'espace. La méthode cristallographique en fait partie, avec en plus la présence de nombreux échantillons à valeur nulle. Ce qui amène à étudier l'erreur dans le cas des matices creuses, pour utiliser dans certains cas des circuits existants au delà de leurs applications originales. Ces différentes voies permettent d'envisager le développement d'architectures cablées pour les transformées de Fourier à grand nombre d'échantillons, particulièrement dans le cas de transformées multidimensionnelles.

Mots-clés : arithmétique, bus, cablé, calcul, Fourier, implantation, multidimensionnel, précision, opérateur, redondant, T.F.R., transformée.

Abstract

The hard-wired computation of a Fourier transform allows to increase its computation speed. Military applications have used such solutions, with a low number of samples and low accuracy. Pushing away these barriers needs decreasing the area of implementation. A large number of butterflies, cells of calculation, implementing serial operators and working simultaneously allows to obtain a better accuracy and a higher speed. The overcost in term of area has been studied through an implementation that is presented with its prospects. A multichip solution imposes the choice of a two-level architecture, serial butterflies and parallel communication buses, one is favoured in term of working ratio and frequency. The accuracy is a function of that of original data and of number of steps, then of number of samples. Operators with a variable size allow to favour accuracy and area or speed, according to the number of rows of butterflies. The parameters of operators optimize the architecture of a Fourier transform for a given decimation. 2 and 4 radices only are really used for decimation at the level of computation. The estimate of area and computation time demonstrates winnings for hard-wired solutions for 8 and 12 radices. Multidimensional transforms show a lower phenomenon of error, for a given total number of samples, because of a larger number of simple exponential coefficients. These are the target of civil applications for a large number of samples, images or spatial data. The crystallographic method is one of them, besides with the presence of many samples having a null value. That leads to study the error in the case of sparse matrices, in sight of using existing chips beyond their original applications. These different ways allow to contemplante implementing hard-wired architectures for Fourier transforms with a large number of samples, particularly in case of multidimensional transforms.

Key-Words: accuracy, arithmetic, bus, computation, Fourier, FFT, hard-wired, implementation, multidimensional, operator, redundant, transform.