



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Manuscrit de Thèse

présenté par

Serge PHILIP

pour l'obtention du titre de

Docteur de l'Université de Metz

en Micro-électronique

Etude et développement d'une nouvelle architecture de processeur DSP dédiée aux applications modem en télécommunications sur câble T.V

Soutenue le 11 décembre 1998 devant la commission d'examen :

Composition du Jury :

Rapporteurs :

Christian LANDRAULT
Michel ROUSSEL

Directeur de recherche CNRS, Montpellier
Professeur, IUT de Troyes

Directeur de thèse :

Bernard LEPLEY

Professeur, Université de Metz

Examineurs :

Abbas DANDACHE
Philippe JEAN
Fabrice MONTEIRO
Serge

Maître de conférences, Université de Metz
Ingénieur, Free Alter Soft
Maître de conférences, Université de Metz
, Université de Nancy 1

BIBLIOTHEQUE UNIVERSITAIRE DE METZ



022 282641 7

Labo

sein du
icro-électronique- CLOES

Manuscrit de Thèse

présenté par

Serge PHILIP

pour l'obtention du titre de

Docteur de l'Université de Metz

en Micro-électronique

BIBLIOTHEQUE UNIVERSITAIRE SCIENCES ET TECHNIQUES - METZ	
No IRs	1998/205
Cote	S/M3 98/58
Loc.	Magasin
Cat	

Etude et développement d'une nouvelle architecture de processeur DSP dédiée aux applications modem en télécommunications sur câble T.V

Soutenue le 11 décembre 1998 devant la commission d'examen :

Composition du Jury :

Rapporteurs :

Christian LANDRAULT
Michel ROUSSEL

Directeur de recherche CNRS, Montpellier
Professeur, IUT de Troyes

Directeur de thèse :

Bernard LEPLEY

Professeur, Université de Metz

Examineurs :

Abbas DANDACHE
Philippe JEAN
Fabrice MONTEIRO
Serge WEBER

Maître de conférences, Université de Metz
Ingénieur, Free Alter Soft
Maître de conférences, Université de Metz
Professeur, Université de Nancy 1

Thèse préparée au sein du
Laboratoire Interfaces Composants Micro-électronique- CLOES

REMERCIEMENTS

Je tiens avant tout à remercier :

Monsieur Bernard LEPLEY, Directeur du Laboratoire LICM/CLOES pour m'avoir accueilli au sein de son laboratoire et dirigé mes travaux. Qu'il trouve dans ces quelques lignes ma profonde sympathie et toute ma reconnaissance.

Monsieur Christian LANDRAULT, directeur de recherche CNRS au LIRMM de l'Université de Montpellier II, pour m'avoir fait l'honneur de présider le jury de cette thèse. Je tiens à lui témoigner toute ma reconnaissance et ma gratitude pour avoir été à l'origine de cette thèse, et pour ses conseils avisés en tant que rapporteur de ce travail.

Monsieur Michel ROUSSEL, Professeur à l'IUT de Troyes pour avoir accepté d'être rapporteur de ce travail, ainsi que pour ses conseils avisés.

Monsieur Serge WEBER, Professeur à l'Université de Nancy 1, pour avoir accepté d'être membre de ce jury.

Monsieur Philippe JEAN pour m'avoir accueilli au sein de la société I&Tcom. Je tiens à le remercier pour avoir facilité mon intégration et toute l'aide apportée lors de ce travail.

Monsieur Abbas DANDACHE et Monsieur Fabrice MONTEIRO pour avoir encadré mes travaux ainsi que pour leur soutien moral et logistique.

Enfin, je n'oublierai pas toutes les personnes, membres du Laboratoire, membres de ma famille ou amis, qui par leurs actions, gestes, paroles ou écoutes m'ont soutenu tout au long de cette thèse. Je citerai tout particulièrement (et par ordre alphabétique !) : Angélique, Charly, Francky, Gaël, Gaëlle, Hervé, Jean-François, Jean-Philippe, Joule, Maman, Papa, Rémy, Seb, Thierry, Zep.

Remerciements

Special Thanks to T. D. D. M

Je dédie la totalité de ce travail à une personne trop tôt disparue, mon père.

RESUME

Ce travail a été effectué dans le cadre d'un contrat CIFRE entre le laboratoire L.I.C.M (Laboratoire Interface Composant Micro-électronique) de l'Université de Metz et la société I&T COM associé à un projet Européen EURICO. Ce projet a pour but de développer un modem pour câble T.V à 1.5 Mbit/s.

La réalisation des applications modem dans le domaine des télécommunications est sujette ces dernières années à une grande évolution. Une nouvelle gamme d'applications propre au multimédia est apparue nécessitant des débits de plus en plus élevés et une multitude de traitement. Cette émergence a pour conséquence un développement de nouvelles structures matérielles. La norme du réseau DAVIC répond aux contraintes des applications multimédia et fournit les structures adaptées à l'interopérabilité.

Une étude des spécifications DAVIC (Digital Audio-VISual Council) sur câble T.V a permis de définir les contraintes fonctionnelles associées aux applications modem pour deux modes de transmission (unidirectionnelle et bidirectionnelle). La réalisation d'un modem compatible avec les réseaux DAVIC impose de traiter l'ensemble des fonctions contenues dans les spécifications.

Dans le cadre du projet EURICO un prototype est réalisé autour d'un processeur DSP du commerce en respectant des contraintes inspirées de DAVIC pour le débit de 1.5 Mbits/s. Les fonctions complexes imposées par DAVIC s'avèrent trop consommatrice en temps de traitement pour qu'une solution programmée puisse prendre à sa charge la totalité de l'application modem.

Sur la base de ce constat, le travail effectué consiste à développer une nouvelle architecture de processeur adapté à ces applications. L'architecture étudiée est basée sur un jeu de modules ADMs (Auxiliary Dedicated Module) agencés autour d'un cœur de DSP. Ces ADMs sont dédiés au traitement des fonctions complexes avec une partie opérative câblée. Une architecture modulaire, paramétrable et hautement parallèle est définie pour offrir la souplesse, la flexibilité et un niveau de performances suffisant pour réaliser les applications modems.

L'implantation sur FPGA d'une fonction complexe, le CRC, dans un ADM validé a permis de vérifier les performances de notre approche. Les résultats comparés avec ceux d'une solution programmée confirment les meilleures dispositions de notre processeur à traiter les applications modem.

Mots clés

réseau haut débit, multimédia, modem, DAVIC, DSP, architecture de processeur

ABSTRACT

This work is a part of an EURICO European project carried out in collaboration with the L.I.C.M laboratory (laboratoire Interfaces Composants Micro-électronique) and the I&Tcom society. The target of this project is the design of a T.V cable modem with a data rate of 1.5 Mbit/s.

In the last years, the design of modems has been subject to great advancements, in the telecommunication domain. A new range of applications has emerged dedicated to the multimedia domain requiring very high data rates and complex processing. The consequence has been the need for new design architectures.

The DAVIC (Digital Audio-VIsual Council) normalisation group proposes adapted structures for interoperability between application on TV cable. The design of unidirectional and bi-directional modems on TV cable can be designed respecting the functional constraints from the DAVIC specifications.

A prototype design for the EURICO project was made using a standard DSP processor, complying the DAVIC specification and allowing a data rate up to 1.5 Mbits/s. However, the time constraints are hard to respect using a pure software solution as most of the DAVIC functions are critical.

Thus, a dedicated DSP architecture was designed to fit the time constraints of the applications. The architecture is based on a DSP core and a set of Auxiliary Dedicated Modules (ADM). Each ADM processes a time-critical function through a specific hardware data path. However, the architecture is strongly parallel, modular and configurable granting a high level of performance, as required by the TV modem applications.

The implementation of an ADM on an FPGA device, dedicated to the CRC function, proved the effectiveness of this solution. These results, largely above those of the pure software solution, confirm the good adequacy of our DSP architecture to the TV modem applications.

Keywords

High speed networks, multimedia, modems, DAVIC, DSP, processor architecture

SOMMAIRE

REMERCIEMENTS	2
RÉSUMÉ	4
ABSTRACT.....	5
SOMMAIRE	6
RÉFÉRENCES BIBLIOGRAPHIQUES.....	11
INTRODUCTION	14
CHAPITRE I L'ÉVOLUTION DES MODEMS DANS LE DOMAINE DES	
TÉLÉCOMMUNICATIONS.....	17
I.1 GENERALITES SUR LA TRANSMISSION D'UN SIGNAL NUMERIQUE	18
I.1.1 <i>Transmission en bande de base</i>	18
I.1.2 <i>La transmission en bande transposée</i>	19
I.2 PRÉSENTATION ET GÉNÉRALITÉS SUR LE MODEM	20
I.2.1 <i>Les modems standards</i>	20
I.2.2 <i>Les fonctionnalites d'un modem</i>	21
I.2.2.1 Les techniques de modulation.....	22
I.2.2.2 Les techniques de filtrages.....	25
I.2.2.3 Les techniques de codages	27
I.2.2.4 Les codes cycliques.....	28
I.2.2.5 Les codes convolutifs.....	28
I.3 LE SUPPORT ÉTUDIÉ : LE CÂBLE T.V	29
I.3.1 <i>Présentation</i>	29
I.3.2 <i>les différents types de bruit</i>	30
I.3.3 <i>Modulation et techniques à acces multiples</i>	32
I.4 LES DIFFÉRENTES APPLICATIONS.....	33
I.4.1 <i>La diffusion vidéo</i>	34
I.4.2 <i>La vidéo à la demande</i>	34
I.4.3 <i>La télévision avancée</i>	34
I.4.4 <i>Le son numérique</i>	34
I.4.5 <i>La vidéo conférence</i>	35
I.4.6 <i>Connexion aux réseaux PC</i>	35
I.4.7 <i>Les jeux électroniques</i>	35
I.4.8 <i>Les caractéristiques techniques</i>	35
I.5 L'ÉVOLUTION DES TÉLÉCOMS VERS DES STRUCTURES RÉSEAUX.....	37
I.5.1 <i>Le réseau DAVIC (Digital Audio-VIsual Council)</i>	38
I.5.2 <i>Le réseau MCNS (Multimedia Cable Network System)</i>	38
I.5.2.1 Les spécifications techniques.....	39
I.5.3 <i>Le groupe IEEE 802.14</i>	40

I.5.4	Les spécifications de la couche MAC (Medium Access Control)	40
I.6	CONCLUSION	40
CHAPITRE II	TECHNIQUES DE CONCEPTION D'UN MODEM POUR CÂBLE T.V	42
II.1	INTRODUCTION	43
II.2	ARCHITECTURE GÉNÉRALE DU MODEM	43
II.3	LA NORME DAVIC : SPÉCIFICATIONS ET COMPLEXITE FONCTIONNELLE.....	44
II.3.1	<i>Spécification du mode de transmission unidirectionnel</i>	45
II.3.1.1	Structure des trames	45
II.3.1.2	Flot d'information au format MPEG-2	46
II.3.1.3	Structure de la trame supportant du MPEG-2-TS	46
II.3.1.4	Flot d'information au format ATM (Asynchronous Transfer Mode).....	46
II.3.1.5	La structure de trame encapsulant des cellules ATM.....	47
II.3.1.6	Le correcteur de haute sécurité (HRM)	48
II.3.1.7	Le codage du canal.....	48
II.3.1.8	Convertisseur octet / symbole	50
II.3.1.9	Quadrature Amplitude Modulation (QAM)	51
II.3.1.10	Information sur les débits	53
II.3.2	<i>Bilan et conclusion</i>	53
II.3.3	<i>Spécification pour le mode de transmission birectionnel</i>	54
II.3.3.1	Spécification de l'interface physique pour la voie descendante	54
II.3.4	<i>Spécifications de la voie montante</i>	59
II.3.4.1	La modulation QPSK en voie montante.....	59
II.3.4.2	Accès multiples par division de temps TDMA (Time Division Multiple Access)	60
II.3.4.3	Structure du slot sur voie montante.....	60
II.3.4.4	La signalisation de contrôle de la voie montante	60
II.3.5	<i>Bilan et conclusion</i>	61
II.3.6	<i>Les fonctionnalités de la couche MAC (Media Access Control)</i>	61
II.3.6.1	Modèle de référence MAC.....	62
II.3.6.2	Transport de l'information en sous-couche MAC.....	62
II.3.6.3	Les différents types de messages MAC	63
II.3.7	<i>Bilan et conclusion</i>	64
II.4	RÉALISATION D'UNE SOLUTION CABLEE	65
II.4.1	<i>Vue générale</i>	65
II.4.2	<i>Limitation et problématique</i>	66
II.5	CONCEPTION D'UNE CARTE AUTOUR D'UN DSP	66
II.5.1	<i>Présentation du prototype réalisé</i>	67
II.5.1.1	Spécifications pour les clients.....	68
II.5.1.2	Les spécifications pour le Serveur	69
II.5.2	<i>presentation du processeur DSP d'Hyperstone</i>	70
II.5.3	<i>Le protocole d'échange choisi : Le Polling</i>	71
II.5.4	<i>La synchronisation</i>	72
II.5.5	<i>Spécifications du composant FPGA en flux descendant</i>	72

II.5.5.1	Interfaçage avec le DSP d'HyperStone	72
II.5.6	<i>Spécifications du composant FPGA en flux montant</i>	73
II.5.6.1	Interfaçage avec le DSP	73
II.5.7	<i>contraintes supportées par le DSP en soft</i>	73
II.5.8	<i>Résultats des simulations temporelles</i>	74
II.5.8.1	Le Reed-Solomon	74
II.5.8.2	Bilan coté client	75
II.5.8.3	Bilan du coté du serveur.....	75
II.5.9	<i>Limitations et problématique</i>	76
II.6	LA SOLUTION HYBRIDE PROPOSÉE	76
II.7	CONCLUSION	78
CHAPITRE III	ARCHITECTURE DU DSP	80
III.1	INTRODUCTION	81
III.2	LES DIFFÉRENTES STRATÉGIES ARCHITECTURALES.....	81
III.3	UN PROCESSEUR PARTICULIER : LE D S P	83
III.3.1	<i>les aspects architecturaux</i>	83
III.3.1.1	Générateur d'adresses	83
III.3.1.2	Le compteur de répétition.....	84
III.3.1.3	Les architectures des parties opératives	84
III.3.2	<i>Le cœur de DSP</i>	86
III.4	NOTRE PROCESSEUR DSP	88
III.5	LE CŒUR DE DSP.....	90
III.5.1	<i>Une architecture VLIW</i>	90
III.5.2	<i>La technique de pipeline</i>	91
III.5.3	<i>Le séquenceur</i>	92
III.5.4	<i>Les unités de calcul</i>	93
III.6	LE JEU D'INSTRUCTION	93
III.7	LES ADMS	95
III.8	LE PROTOCOLE DE COMMUNICATION INTERNE	97
III.9	CONCLUSION	98
CHAPITRE IV	L'ARCHITECTURE DES ADMS.....	100
IV.1	INTRODUCTION	101
IV.2	VUE GÉNÉRALE	102
IV.3	L'INTERFACE ÉLECTRIQUE.....	103
IV.3.1	<i>La gestion des chaines de priorité</i>	103
IV.3.2	<i>Le contrôle des accès aux bus</i>	106
IV.4	L'INTERFACE DE CONTRÔLE	108
IV.4.1	<i>Le decodeur d'adresse</i>	110
IV.4.2	<i>Le décodeur</i>	111
IV.4.3	<i>Le contrôleur</i>	111

IV.4.3.1	Le contrôle principal.....	111
IV.4.3.2	Le contrôle de protocole.....	112
IV.5	LE CŒUR D'ADM.....	112
IV.5.1	<i>La partie contrôle</i>	113
IV.5.2	<i>La partie opérative</i>	113
IV.6	CONCLUSION.....	114
CHAPITRE V EXEMPLE D'IMPLANTATION DU CŒUR D'ADM		116
V.1	INTRODUCTION.....	117
V.2	LE CRC.....	118
V.2.1	<i>Principe</i>	118
V.2.2	<i>Algorithme</i>	118
V.3	LA SOLUTION ARCHITECTURALE DU CŒUR D'ADM	119
V.3.1	<i>Réalisation de la fonction de calcul du CRC</i>	120
V.3.2	<i>Réalisation des fonctions de transfert</i>	124
V.3.2.1	Architecture pour l'acquisition des données.....	124
V.3.2.2	Architecture pour l'émission du résultat.....	125
V.3.3	<i>Réalisation de la gestion des paramètres</i>	125
V.4	LA PARTIE CONTRÔLE DU CŒUR D'ADM	126
V.5	SIMULATION ET RESULTATS.....	127
V.5.1	<i>Présentation des résultats de la simulation du cœur d'ADM</i>	127
	<i>Résultats de simulation de l'ADM complet</i>	129
V.5.3	<i>Interprétation des résultats</i>	130
V.5.4	<i>Estimation de la surface</i>	131
V.6	CONCLUSION.....	132
CHAPITRE VI CONCLUSION ET PERSPECTIVES		134

REFERENCES BIBLIOGRAPHIQUES

Références

- [1] STEIN, "Les modems pour la transmission de données numériques" Collection technique et scientifiques des télécommunications, 2^{ème} édition. Masson 1991.
- [2] PJ. HUGUES, J. COOK, "Adaptive Filters – a review of Techniques , BT Technology Journal, Vol. 10, No. 1, January 1992.
- [3] PAGE, PAULEY, "An Application Of DSP To Voiceband Modems", BT Technology Journal, vol. 10, No. 1, pp. 80-100, January 1992.
- [4] CCITT (Comité consultatif International de Téléphonie et Télécommunication) "Transmission de données" Recueil des recommandations du CCITT, tome VIII, série V 1991.
- [5] MATTIS, "A new modem structure for Data Transmission". IEEE Transactions On Consumer Electronics, vol. 39, No. 4, pp. 878-886, 1992
- [6] CHEUNG, "Performance of CE16QAM modem in a regenerative satellite system". International Journal of satellite communication, vol. 7, pp. 425-434. 1989
- [7] G. FORNEY, "Maximum Likelihood Sequence Estimation Of Digital Sequences In The Presence Intersymbol Interference", IEEE Transaction On Information Theory, vol. IT-18, No. 3, pp. 363-378, May 1972.
- [8] D. MESSERSCHMITT, "Echo Cancellation In Speech And Data Transmission", IEEE Selected Areas In Comms, vol. 2, No. 2, March 1984.
- [9] MATSUSHIMA, HIRASAWA, "Parallel Encoder And Decoder Architecture For Cyclic Codes", IEICE Trans. Fundamentals, vol. E79-A, No. 9, pp. 1313-1323, September 1996.
- [10] POLI , HUGUET "Codes Correcteurs. Théories et applications". Collection "Logique Mathématiques Informatiques", vol. 1, Masson 1989.
- [11] C. ELDERING, N. HIMAYAT, F. GARDNER "CATV Return Path Characterization for Reliable Communications". IEEE Communications Magazine, pp 62-68, August 1995.
- [12] SPATARU, "Fondements de la théorie de la transmission de l'information" Presses Polytechniques Romandes, 1987.
- [13] G. PUJOLLE, "Les réseaux", Eyrolles, Juillet 1997
- [14] D. GINGOLD "Integrated Digital Services for Cable Networks". Master of Science in Technology and Policy at the Massachussetts Institute of Technology, September 1996.
- [15] DE JONG, "Access to a DAVIC-Based Video on Demand System Using the World Wide Web". 11th International Conference on Information Networks. Tapei, Taiwan, January 27-29, 1997.

- [16] ATM Forum, "Video on Demand Specifications 1.0". Audio-visual Multimedia Services January 1996
- [17] DAVIC, "The Specification DAVIC 1.3, ". Digital Audio Visual Council. Berlin, November 1997.
- [18] DOCSIS, "Data-Over-Cable Service Interface Specifications", Ref. SP-CMCI-I02-980317, 1998.
- [19] IEEE Project 802.14/a "Cable-TV Access Method and Physical Layer Specification", Draft2, Revision 2, july 1997.
- [20] HENNESSY, PATTERSON, "Computer Architecture: A Quantitative Approach", 2nd edition. Morgan Kaufmann Inc., San Francisco, 1995.
- [21] TANENBAUM, "Computer Networks". Prentice Hall Inc., Englewood Cliffs, NJ, 1994.
- [22] MOON, EBCIOGLU, "On Performance and Efficiency of VLIW and Superscalar". International Conference on Parallel Processing, vol. 2, pp. 283-287. CRC Press, Ann Arbor, 1993.
- [23] H. AHMED, R. KLINE, "Recent Advances in DSP Systems". IEEE Communications Magazine, pp. 32-45, May 1991
- [24] J.F DUBOC, "Spécification et Modélisation du Cœur d'un Processeur de Traitement du Signal (DSP)".Thèse de doctorat, EDSI Nice Sophia-Antipolis, 1993
- [25] ANALOG DEVICES "ADSP-2106X SHARC DSP Microcomputer Family" Data sheet, Rev B, 1998.
- [26] TEXAS INSTRUMENTS "TMS320C62X/C67X" Programmer's Guide, Literature Number SPRU198B, February 1998.
- [27] PEI, ZUKOWSKI, "High-speed parallel CRC circuits in VLSI". IEEE Trans. Commun., vol.40, no.4, April 1992.
- [28] MONTEIRO, PHILIP, DANDACHE, LEPLEY, "A New Processor Architecture Dedicated to Digital Modem Applications". IEEE ISCAS, Monterey, California, May 1998.
- [29] PHILIP, MONTEIRO, DANDACHE, LEPLEY, "A High-Speed Parallel DSP Architecture Dedicated to Digital Modem Applications". IEEE ICECS, Lisboa, September 1998.

INTRODUCTION

De nos jours, l'utilisation de modems pour la transmission des données est devenue le trait d'union entre le domaine de l'informatique et celui des télécommunications. Situés au carrefour des techniques numériques et analogiques, les modems ont connu sous l'impulsion conjuguée du progrès technologique et du développement des applications en télécommunication une évolution extrêmement rapide.

La diversité des supports de transmission et des modes d'exploitation a conduit à une grande variété d'équipements.

Il est alors difficile pour un ingénieur en charge de mettre en place un modem, de disposer de tous les éléments lui permettant d'effectuer un choix structurel judicieux parmi les matériels de transmission pour satisfaire les nombreuses contraintes fonctionnelles.

Dans ce domaine pluridisciplinaire il faut, en effet, acquérir de connaissances suffisantes dans des domaines variés pour en extraire les contraintes propres à chacun : le support de transmission (atténuation, bruit,...), les principes de la transmission numérique (débit, codage, filtrage, modulation, ...), les modes d'exploitation (type de données, format de trame, synchronisation) ainsi que les méthodes de maintenance (test, validation, messages de signalisation,...).

D'un côté, au niveau de ces aspects physiques, des efforts de standardisation sous forme de réseaux de communication sont en cours tels que : DAVIC (Digital Audio-VIsual Council), MCNS (Multimedia Cable Network Standard), IEEE ...

Ces réseaux ont pour but de définir une chaîne fonctionnelle unique pour chaque type d'application de manière à satisfaire le besoin grandissant d'interopérabilité entre différents sites. Ces normalisations ont l'avantage de fixer des spécifications fonctionnelles garantissant le respect des contraintes de l'application visée. Deux inconvénients majeurs émanent de ces spécifications, l'usage de fonctions complexes et l'emploi de nombreuses fonctionnalités pour favoriser la compatibilité entre plusieurs structures de réseaux.

D'un autre côté, confrontée à ces nouveaux besoins en communication et à la complexité des fonctions liées aux traitements de l'information, la réalisation d'un modem est devenue totalement dépendante des progrès technologiques en micro-électronique.

La micro-électronique, élément central de ce développement technologique, est directement confrontée à ce challenge.

La réponse actuelle est l'intégration toujours plus poussée des fonctions requises au sein de circuits intégrés.

L'apport de ces évolutions technologiques dote les processeurs d'une plus grande capacité d'intégration et d'une plus grande vitesse de traitement. Une intégration de cinq à six millions de transistors, sur une seule puce est devenue une réalité. Il s'agit alors de bien gérer le potentiel de conception induit par le nombre élevé de fonctions intégrables et d'améliorer les conditions de développement des circuits numériques.

L'objectif initial de cette thèse était de développer une carte modem pour câble T.V dans le cadre d'un contrat CIFRE entre le laboratoire LICM (Laboratoire d'Interface et de Composants en Micro-électronique) et la société I&T COM pour un projet européen EURICO.

Afin de ne pas aboutir à un produit marginal, il est nécessaire de respecter les standards industriels. Dans le domaine des télécommunications, ces standards sont représentés par des normes telles que DAVIC, MCNS... Une étude approfondie d'une de ces normes (DAVIC) s'est donc imposée. Cette étude s'est poursuivie par la réalisation d'une carte modem prototype en conformité avec la norme suscitée.

Les performances limitées d'une telle réalisation, suffisantes pour satisfaire au cahier des charges du projet EURICO, interdisent la généralisation de cette approche aux applications modem haut débit à venir. Un nouvel objectif de recherche est venu compléter les objectifs précédents : développer une nouvelle architecture de processeur DSP (Digital Signal Processor) dédiée aux applications modem sur câble T.V.

Le premier chapitre, consacré à l'état de l'art, permet de définir l'appellation modem au niveau du traitement du signal. Dans un deuxième temps nous présentons les différents éléments nécessaires à la réalisation d'un modem pour câble T.V.

Le deuxième chapitre, est dédié aux différentes techniques de conception liées à la réalisation d'un modem pour câble T.V. Pour cela, une présentation du cahier des charges est proposée sur la base de la norme DAVIC. Des solutions purement câblées sont énoncées provenant de compagnies actives dans l'élaboration de la norme. Nous proposons comme autre solution, un prototype que nous avons réalisé dans le cadre d'un contrat européen sous forme d'un circuit imprimé. Au vue de l'expérience acquise lors de la conception du prototype et des études faites sur les deux premières solutions, nous développons une troisième solution technologique sous forme d'un processeur DSP dédié aux applications dites "modem"

répondant au mieux aux différentes exigences.

Le troisième chapitre est consacré à l'architecture de notre processeur basé sur un ensemble de modules spécifiques (ADM : Auxiliary Dedicated Module) autour d'un cœur de DSP. Nous justifions dans un premier temps les différents choix de notre architecture avant de détailler le principe de fonctionnement et les différents éléments permettant d'optimiser les performances. Nous nous attachons tout particulièrement à la description du protocole de communication interne entre le cœur du processeur et les modules auxiliaires appelés "ADM". Ces ADMs jouent un rôle très important dans la mesure où ils ont en charge la réalisation des fonctions complexes très consommatrices de temps sur les processeurs classiques.

Le quatrième chapitre décrit les ADMs sous leur forme structurelle. Cette description est accompagnée d'explications concernant le fonctionnement des trois parties principales des ADMs : l'interface électrique pilotant les niveaux électriques de tous les signaux entrant ou sortant de l'ADM, l'interface de contrôle gérant et garantissant le respect du protocole de communication interne et externe à l'ADM à travers les différents signaux et le cœur d'ADM réalisant la fonction souhaitée.

Le dernier chapitre présente les solutions d'implantation de fonctions complexes à l'intérieur des ADMs. Le choix d'implantation est fonction de la complexité de l'algorithme étudié, du coût en surface engendré et de la souplesse de l'architecture offrant plusieurs configurations. Ce travail exige une étude approfondie et spécifique pour chacune des fonctions complexes. Au travers de l'exemple de la fonction de codage appelée CRC (Cyclical Redundancy Checking) nous proposons une stratégie et une solution d'implantation au niveau du cœur d'ADM. L'architecture obtenue a fait l'objet d'une implantation sur un composant FPGA. Les résultats de simulation obtenus donnent lieu à une évaluation des performances de cette approche comparée aux résultats obtenus lors de la mise en œuvre du prototype (étudié dans le deuxième chapitre).

CHAPITRE I

L'EVOLUTION DES MODEMS

DANS LE DOMAINE DES

TELECOMMUNICATIONS

I.1 GENERALITES SUR LA TRANSMISSION D'UN SIGNAL NUMERIQUE

Dans le domaine des télécommunications, une communication s'effectue par un transfert d'informations entre un émetteur et un récepteur. La transmission des données entre deux éléments de part et d'autre du canal de transmission est soumise aux caractéristiques de ce dernier. De ce fait, une chaîne de transmission se schématise de la manière suivante (voir figure 1) :

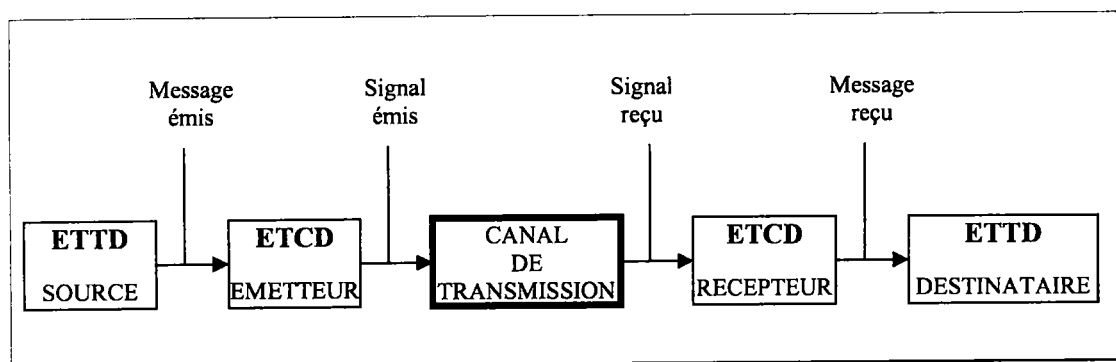


figure 1 : chaîne de transmission

Une liaison numérique est composée de quatre composants, deux ETTD (Equipement Terminal de Traitement de Données) et deux ETCD (Equipement Terminal des Circuits de Données) [1]. Les ETTD situés en début et en fin de la ligne de transmission, permettent d'échanger des informations sur le support de transmission. Les ETCD placés de part et d'autre du support de transmission, permettent d'assurer la conversion des signaux numériques, issus des ETTD, en signaux analogiques compatibles avec le support de transmission, et inversement.

I.1.1 TRANSMISSION EN BANDE DE BASE

Une transmission en bande de base est assimilée à un mode de transmission numérique. Cette technique représente la manière la plus simple de réaliser une transmission de données puisqu'elle consiste à transmettre directement le signal sans le transformer. Un signal en bande de base est un signal qui n'a pas subi de translation dans le domaine des fréquences.

Ainsi les supports utilisés pour ce genre de transmission sont des supports qui n'introduisent pas d'écart de fréquence entre signaux émis et signaux reçus. Lors d'une transmission d'un signal en bande de base, le récepteur a pour but de reconstituer le signal émis à partir du signal reçu. La remise en forme du signal s'avère souvent difficile à cause des

perturbations dues au canal de transmission. Ces perturbations agissent principalement dans le cadre d'une bande passante limitée en provoquant des distorsions d'amplitude et de phase.

Cette difficulté peut être réduite en utilisant des méthodes particulières de représentation de l'information dites de codage. Il existe un grand nombre de telles méthodes (codes binaires, codes bipolaires, codes biphases, codes linéaires, codes non linéaires, etc....) chacune d'elles se singularisant soit par de la redondance soit par une augmentation du nombre de niveaux soit enfin par une modification du spectre de puissance soit par une annulation de l'énergie. La sélection d'une méthode de codage consiste donc à rechercher le meilleur compromis entre les avantages du code utilisé et la dégradation du rapport signal sur bruit [2]. Ce compromis dépend avant tout des caractéristiques du support de transmission.

Quelle que soit la méthode de codage adoptée, le signal reçu doit être échantillonné avant décodage au rythme d'une « horloge » d'échantillonnage. L'un des problèmes essentiels lié à la transmission numérique est la génération dans le récepteur de ce signal d'échantillonnage. Il existe globalement deux méthodes pour transférer de l'émetteur vers le récepteur les informations de fréquence et de phase nécessaires à la génération locale du rythme d'échantillonnage, la transmission de l'horloge indépendamment du signal codé et l'utilisation des transitions du signal codé.

Cette méthode de transmission permet d'obtenir des débits très élevés sur des distances courtes. Elle conduit à des réalisations simples et économiques mais la transmission en bande de base n'est pas possible sur tous les supports.

I.1.2 LA TRANSMISSION EN BANDE TRANSPOSEE

La plupart des supports de transmission ne permettent pas la transmission directe d'un signal numérique en bande de base. De plus, il est souvent nécessaire de faire coïncider la bande de fréquence occupée par le signal transmis avec la bande passante du rapport de transmission.

Ces deux problèmes peuvent être résolus simultanément en modulant une sinusoïde porteuse de fréquence convenable par le signal à transmettre. L'opération de modulation équivaut en effet, à une translation du spectre du signal dans le domaine des fréquences et permet de centrer son énergie à l'intérieur de la bande passante du support. A cette translation s'ajoute celle due au décalage de fréquence introduit par la ligne. L'opération de démodulation consiste à restituer le signal numérique sous sa forme initiale en effectuant une translation égale à la somme des deux translations et de signe opposé.

Les différentes techniques de modulation utilisées en transmission numérique sont

réalisées en manipulant soit l'amplitude de la sinusoïde porteuse (modulation d'amplitude), soit son angle de phase (modulations de phase) soit de fréquence (modulation de fréquence) soit encore deux ou trois de ces caractéristiques à la fois (modulations combinées de phase et d'amplitude tel que la modulation d'amplitude et de deux porteuses en quadrature).

De plus en plus, les réalisations modernes font appel à des méthodes de traitement numérique du signal [3], que ce soit sous forme matérielle (hardware) ou logicielle (software).

L'un des intérêts du traitement numérique est d'offrir la souplesse et le coût des solutions programmées et une grande robustesse aux perturbations. L'une des façons possibles de procéder pour moduler un signal d'une façon numérique consiste à réaliser les opérations filtrage et de multiplication nécessaires à la modulation.

I.2 PRESENTATION ET GENERALITES SUR LE MODEM

Le modem est l'un des champs d'application privilégiés des techniques numériques de traitement du signal. Le modem réalise l'interface entre la source d'information et le support de transmission. Dans ce but, plusieurs fonctions lui sont attribuées, dont la plus importante est d'adapter au mieux les données à transmettre au support physique de transmission.

Toutefois l'objectif principal d'un modem est d'envoyer des données numériques sous forme analogique sur un canal de communication avec le moins d'erreurs.

Un modem possède deux modes de fonctionnement, l'un étant rattaché à l'aspect de MODulation communément appelé le mode émetteur, l'autre faisant référence à l'aspect de DEModulation appelé le mode récepteur. En mode émetteur, le modem a pour but de convertir les données à émettre en un signal analogique adapté au support de transmission et transposant ce signal émis à l'intérieur des bornes de la bande passante. En mode récepteur, le modem récupère le signal reçu lequel a été dégradé par le canal de transmission et toute la difficulté de ce mode réside dans les différents traitements à appliquer afin de reconstituer les données émises.

I.2.1 LES MODEMS STANDARDS

Il existe une large gamme de modems standards (voir tableau 1) lesquels ouvrent les débits de 300bits/s à 56200 bits/s [4]. La plupart de ces modems sont destinés à opérer sur le Réseau Téléphonique Commuté (RTC) avec la possibilité d'opérations « full-duplex ».

CCITT Standard	Débit en bit/s	R.T.C	Duplex
V.21	300/300	OUI	OUI
V.22	1200/1200	OUI	OUI
V.22 bis	2400/2400	OUI	OUI
V.23	1200/75	OUI	OUI
V.26	2400	NON	NON
V.26 bis	2400	OUI	NON
V.26 ter	2400/2400	OUI	OUI
V.27	4800	NON	NON
V.27 bis	4800	NON	NON
V.27 ter	4800	OUI	NON
V.29	9600	NON	NON
V.32	9600/9600	OUI	OUI
V.32 bis	14400/14400	OUI	OUI
V.33	14400	NON	NON
V.34	28800	OUI	OUI
V.34 bis	33600	OUI	OUI
V.90	56200	OUI	OUI

tableau 1 : Les modems standards V.séries

Plusieurs techniques de modulations sont utilisées dans les modems standards. Dans le cas des modems à faible vitesse où le débit n'est pas élevé, le choix de la technique de modulation et de la largeur de bande n'est pas pénalisant pour le respect des performances. Dans ce cas, une simple technique de modulation en fréquence (FM) est utilisable. Cette technique plus connue sous le nom de modulation FSK (Frequency Shift Keying) est présentée plus loin dans le paragraphe 1.2.2.1. Toutefois, cette technique est utilisée jusqu'à des débits de 1200bits/s. De plus, ce type de modem peut opérer de manière « full-duplex », les données sont alors envoyées dans les deux directions (voie d'émission et voie de réception) en même temps. Dans ce cas, l'usage de deux fréquences différentes s'avère indispensable.

I.2.2 LES FONCTIONNALITES D'UN MODEM

Un modem est essentiellement caractérisé par les fonctions de MODulation en mode émission et de DEModulation en mode réception comme l'indique son mot. Toutefois, la

fonctionnalité d'un modem complexe dépasse le cadre de ces deux seules fonctions [3], [5]. Un modem doit assurer la bonne transmission et récupération des données. Or la qualité du signal lors de son passage dans le canal de transmission est contrariée par des phénomènes parasites ou des phénomènes de bruit, d'interférence ou d'écho. Des fonctions de filtrages de codages, d'adaptation sont réalisées par le modem pour respecter la fonctionnalité comme le montre la figure 2.

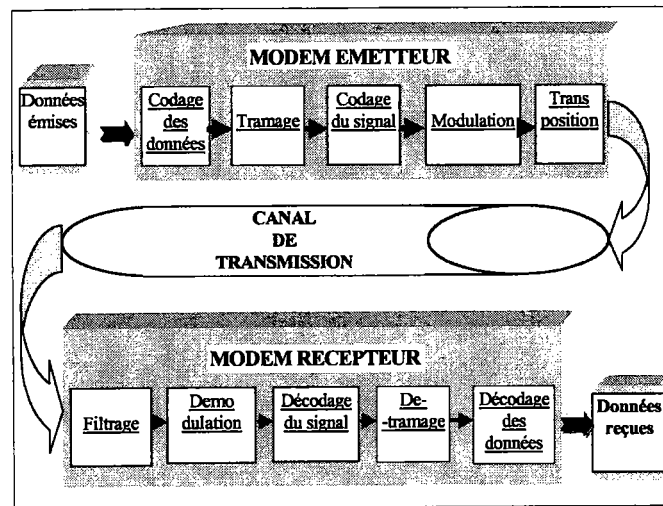


figure 2 : Les blocs fonctionnels de modem à modem

1.2.2.1 Les techniques de modulation

1.2.2.1.1 La modulation d'amplitude

Elle est peu employée pour la modulation de données numérique. En effet la reconstitution se fait par rapport à l'amplitude du signal reçu et généralement le rapport signal/bruit se détériore lors de la démodulation d'où un taux d'erreur très élevé.

Il existe essentiellement trois techniques [1], la modulation du tout ou rien (On Off Keying, OOK), la modulation à décalage (Amplitude Shift Keying, ASK) et la modulation à Bande Latérale Unique (BLU).

- La modulation OOK consiste à effectuer l'émission d'un signal ou non suivant la valeur binaire de la donnée.
- La modulation ASK fait correspondre plusieurs bits de données à une amplitude particulière du signal transmis. Selon le choix n du nombre de bits, on pourra distinguer 2^n niveaux d'amplitude. Pour limiter le nombre de niveaux d'amplitude, un codage multi-niveaux et symétrique est préconisé.
- La modulation BLU est identique à une modulation analogique, elle permet de

translater le signal bande de base en fréquence et la largeur de son spectre peut être optimisée suivant le codage utilisé (exemple : un codage des données de type biphase aura un spectre plus grand que l'utilisation d'un codage NRZ).

I.2.2.1.2 La modulation par déplacement de fréquence

La modulation de fréquence en numérique est relativement simple. Elle ne varie qu'entre deux valeurs de fréquence, le principe consiste à décaler la fréquence centrale f_0 d'un oscillateur.

Par exemple l'état logique binaire "1" correspond à la fréquence $f_2 = f_0 + \Delta f$ et l'état logique "0" du signal numérique correspond à la fréquence $f_1 = f_0 - \Delta f$. Ainsi la modulation de fréquence s'effectue sur deux fréquences fixes alors que dans le cas analogique, des variations étaient possibles sur toute une plage de fréquence. Deux types de modulation de fréquence sont couramment distingués (voir figure 3) :

- La modulation de fréquence sans continuité de phase : cette modulation est basée sur l'utilisation de deux oscillateurs différents que l'on commute lorsque le signal binaire change d'état. Cette technique a pour conséquence des discontinuités au niveaux des transitions et un étalement du spectre des fréquences.

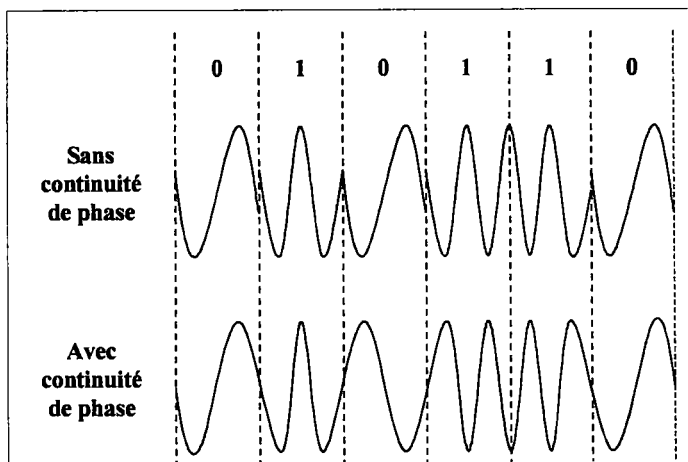


figure 3 : Deux types de modulation de fréquence

- La modulation de fréquence avec continuité de phase est plus souvent connue sous le nom de CPFSK (Continuous Phase Frequency Shift Keying). La phase varie de façon continue aux instants de transitions du signal binaire, limitant ainsi le spectre du signal modulé. Dans la plupart des cas, ce type de modulation est réalisé à l'aide d'un VCO (Voltage Controlled Oscillator), alors que la démodulation est réalisée à l'aide d'une PLL (Phase Locked Loop) [1].

I.2.2.1.3 La modulation par déplacement de phase

Cette modulation appelée PSK (Phase Shift Keying) consiste à transmettre une porteuse sinusoïdale dont seul le terme de la phase varie. Il existe deux procédés pour ce type de modulation:

- La PSK à déphasage absolu est obtenue en modulant le signal pendant l'intervalle de temps élémentaire. Ce signal se trouve alors sous la forme :

$$j(t) = A \cos(\omega_p t + \Phi_0 + \alpha_k)$$

avec A l'amplitude et Φ_0 la phase à l'origine constante et α_k le déphasage définit selon l'état du ou des symboles binaires à transmettre.

Dans ce cas le déphasage introduit sera référencé par rapport au signal sinusoïdal pur qui est : $A \cos(\omega_p t + \Phi_0)$

- La deuxième technique est appelé la modulation par sauts de phase ou déplacement de phase différentiel (Differential PSK). Cette modulation s'appuie sur un signal modulé de la forme :

$$j(t) = A \cos(\omega_p t + \Phi_k)$$

avec Φ_k défini par la relation suivante : $\Phi_k = \Phi_{k-1} + \alpha_k$

Cette technique est la plus utilisée car le démodulateur est relativement simple, contrairement au premier procédé qui nécessite d'avoir un oscillateur à la réception identique à celui de l'émission.

I.2.2.1.4 Les modulations combinées en amplitude et en quadrature (QAM)

Cette technique a été développée pour résoudre les problèmes d'indécisions à la réception du signal modulé. Lorsque le nombre de points est supérieur à huit, la modulation PSK atteint ses limites.

La modulation QAM [6], comme le montre la figure 4, permet de transmettre plusieurs bits simultanément pour un état de la porteuse. Cet avantage offre naturellement la possibilité d'obtenir des débits beaucoup plus importants en comparaison des modulations simples à deux états.

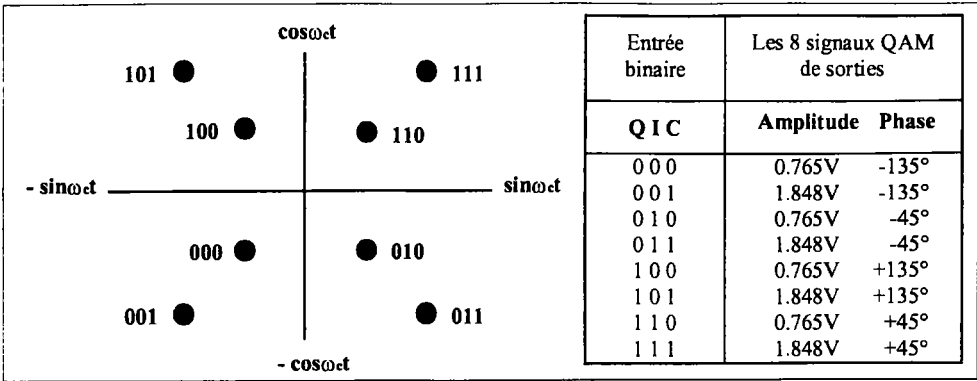


figure 4 : Une constellation QAM à 8 points

Toutefois l’augmentation du nombre d’états du signal modulé accroît la sensibilité du signal aux distorsions et aux perturbations. La modulation QAM est associée à une constellation de points. Chaque point de la constellation représente un état du signal modulé.

L’opération de démodulation consiste à identifier dans la constellation la position du point transmis à partir d’un point reçu. Le signal modulé ayant subi des distorsions d’amplitude et de phases, la position du point récupéré est modifiée par rapport à sa position initiale. Dans ce cas la décision est prise en comparant les énergies des deux états les plus proches. Or La position des points dans la constellation est allouée de façon à proposer une forte différence d’énergie des états représentés par deux points proches.

I.2.2.2 Les techniques de filtrages

I.2.2.2.1 L’interférence inter symbole

Le passage d’un signal à travers un canal de transmission génère des dégradations à l’arrivée des signaux au niveau du récepteur. Lors de la transmission de données à haute vitesse, une dégradation parmi les plus sérieuses est la distorsion de l’amplitude et de la phase [7].

Les conséquences sont une modification de l’inclinaison de l’amplitude et un retard de phase. Aussi de telles dégradations peuvent causer une dispersion dans le temps du signal reçu. Si aucune technique n’est développée pour compenser parfaitement ce phénomène, les performances du récepteur seront sérieusement dégradées.

L’effet de dispersion dans le temps d’un canal est illustré sur la figure 5 représentant la réponse impulsionnelle $h(t)$ classique.

Comme on peut le voir, la réponse s’étend sur plusieurs intervalles de symboles. Quand une séquence de symboles est transmise avec un intervalle T entre symboles, sous l’effet de la dispersion les symboles récupérés au niveau du récepteur se chevauchent. A chaque intervalle

T le symbole reçu est constitué de sa propre valeur à laquelle est rajoutée une partie de la valeur du symbole adjacent.

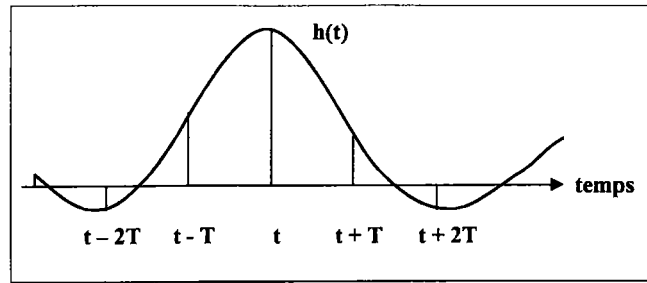


figure 5 : Une réponse impulsionnelle classique à travers un canal

Les modems complexe à haute vitesse sont construits avec des éléments à forte puissance de calcul [5]. C'est pourquoi des filtres de traitement du signal numérique sont utilisés pour compenser automatiquement les effets de dispersion du canal. Ces structures sont connues comme des égalisateurs adaptatifs.

1.2.2.2 Les égalisateurs

De par sa simplicité, la structure d'égalisateur la plus utilisée est celle des filtres à Réponse Impulsionnelle Finie (RIF). Cette structure, illustrée sur la figure 6, est constituée d'une ligne de retard composée d'éléments de mémoire stockant la donnée à chaque "top" d'intervalle T [2]. Le signal reçu $r(t)$ entre dans la ligne de retard (voir figure 6) et est décalé d'un élément de mémoire à chaque période T d'un symbole.

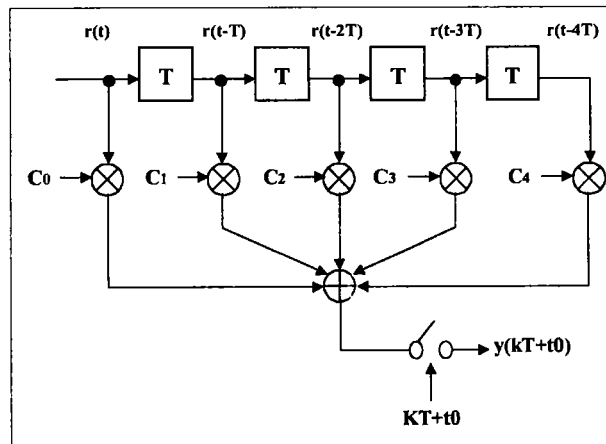


figure 6 : Structure d'un filtre à réponse impulsionnelle fini

Le signal sortant $y(kT+t_0)$ est calculé à chaque intervalle d'un symbole et est composé par l'addition des échantillons affectés par leur coefficient de chaque élément de mémoire.

$$y(kT + t_0) = \sum_{n=0}^{N-1} C_n r(kT + t_0 - nt)$$

Avec C_n les poids ou les coefficients,

$N-1$ représentant le nombre d'éléments mémoire.

I.2.2.2.3 L'annulation d'écho

Du point de vue du récepteur, le phénomène de l'écho se traduit par la présence de signaux non voulus tel que le bruit. La conséquence immédiate de ce phénomène est la diminution du rapport signal sur bruit [8]. L'ampleur de ce problème varie en fonction du type du niveau du signal reçu et des caractéristiques du canal.

Pour lutter efficacement contre les dégradations, les modems traitant des hauts débits de données doivent s'assurer de valeurs du rapport signal sur bruit de l'ordre de 25 dB. Le phénomène d'écho peut être modélisé comme une source constante de bruit pouvant atteindre une dégradation de 40 dB sur le signal reçu. Aussi les techniques dynamiques mise en place pour annuler ces échos doivent pouvoir compenser au moins 65 dB de rapport signal sur bruit.

Pour compenser et éliminer ces phénomènes d'écho, une technique classique consiste à utiliser un filtre adaptatif. Toutefois, cette technique fait apparaître plusieurs inconvénients.

Premièrement, le signal obtenu a été hautement corrélé avec la source de bruit et l'aptitude d'un filtre adaptatif à retrouver le comportement exact du signal est faible.

Deuxièmement, le débit d'échantillonnage doit s'effectuer sur une résolution d'au moins 12 points sur une période pour satisfaire le critère de Nyquist. Mais de telles contraintes nécessitent un traitement très intensif qui est difficile à atteindre.

Afin de traiter directement les données émises plutôt que les données du canal, une technique consiste à développer un modulateur transposé (voir figure 7). Cette structure appliquée à l'émetteur combine la modulation QAM avec l'annulation d'écho.

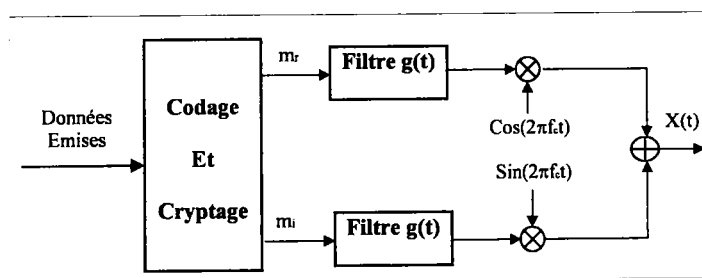


figure 7 : Un modulateur transposé

I.2.2.3 Les techniques de codages

L'égalisation adaptative et l'annulation d'écho sont amenées à traiter un type spécifique de dégradations. En effet, l'égalisation supprime les effets de distorsion linéaire dans le canal tandis que l'annulation d'écho enlève les phénomènes de résonance du canal.

Après la suppression de ces dégradations, le signal reçu est devenu une copie du signal émis à l'exception des modifications de valeurs dues aux problèmes des bruits, de contact, etc.... Aussi, la technique de codage utilisée, pour se protéger contre des erreurs de transmission, consiste à insérer de la redondance dans le message. Cette technique fait appel à des opérations de codage ajoutant au message numérique à transmettre des éléments binaires, dits de redondance suivant une loi donnée. Intuitivement, on peut concevoir que pour un message dépourvu de redondance, chaque élément binaire soit essentiel, aussi toute erreur de transmission conduit à une perte d'information irréversible. Au contraire des éléments de redondance introduits astucieusement vont corréliser les éléments binaires du message codé. Ainsi, sous certaines conditions, un ou plusieurs éléments binaires erronés au cours de la transmission pourront être détectés, voire corrigés.

Le décodeur vient donc tester si la loi utilisée à l'émission est respectée en réception ; si c'est le cas, aucune erreur n'est détectée. Dans le cas contraire, des erreurs sont présentes dans le message codé.

L'introduction du codage conduit toujours à une augmentation du débit numérique transmis, et donc à un accroissement de la bande occupée pour une modulation donnée.

Les codes peuvent être classés en deux grandes familles: les codes en blocs d'une part, les codes convolutifs ou récurrents d'autre part.

I.2.2.4 Les codes cycliques

Les codes cycliques représentent la classe la plus importante des codes en blocs linéaires (Reed-Solomon, BCH, ...). Leur mise en œuvre relativement aisée, à partir de registres à décalages et d'opérateurs logiques combinatoires simples (voir figure 8), en fait des codes très attractifs et très utilisés dans les systèmes de transmission [9].

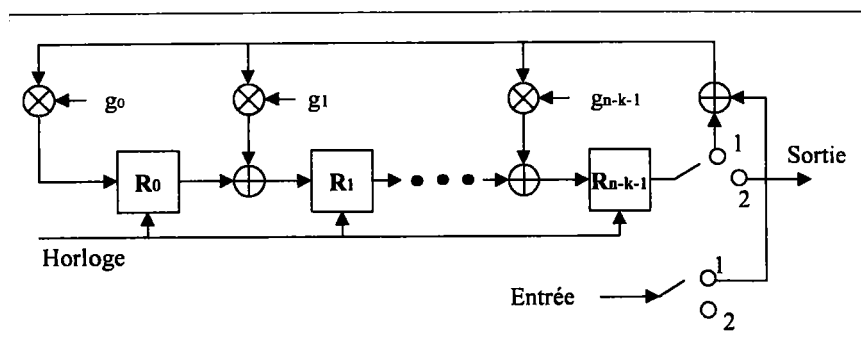


figure 8 : Schéma de principe d'un codeur pour code cyclique

I.2.2.5 Les codes convolutifs

Les codes convolutifs ou récurrents constituent une seconde famille de codes correcteurs

d'erreurs (Viterbi, codeur convolutif, etc.) au moins aussi importante que celle des codes en blocs cycliques [10]. Pour les codes convolutifs, chaque bloc de n éléments binaires en sortie du codeur dépend non seulement du bloc de k éléments binaires présents à son entrée mais aussi des m blocs présents précédemment. Les codes convolutifs introduisent par conséquent un effet de mémoire d'ordre m . La quantité $(m+1)$ s'appelle la *longueur de contrainte* du code. Le principe du code convolutif est illustré par la figure suivante:

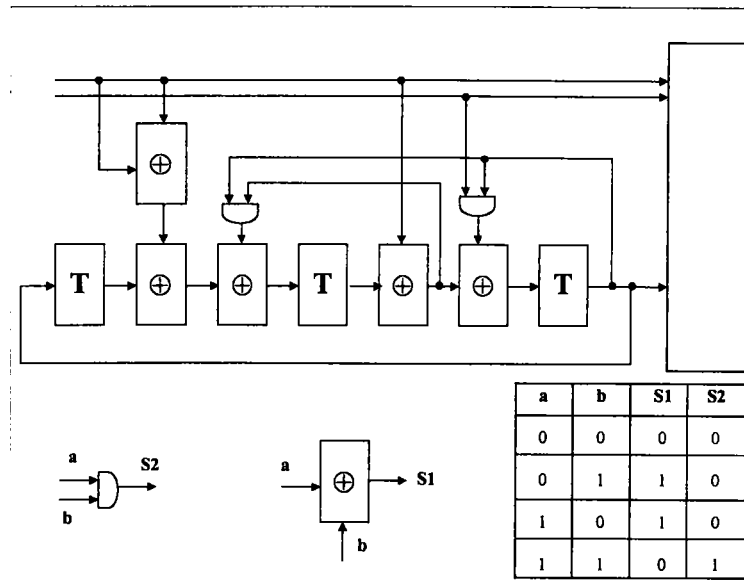


figure 9 : Schéma de principe d'un code convolutif

I.3 LE SUPPORT ETUDIE : LE CABLE T.V

I.3.1 PRESENTATION

Le câble coaxial est composé d'un conducteur central en cuivre et d'une tresse à la périphérie comme le montre la figure 10. Les différents câbles sont désignés par le diamètre utilisé en millimètres. Les deux plus courants sont les 2.9/9.5 et 1.2/4.4. Les connecteurs utilisés sur ces câbles sont les connecteurs BNC pour le petit coaxial et le Série N pour le gros coaxial [4].

La qualité de l'isolant est importante car il assure que les courants circulant sur l'âme ne sont pas perturbés par ceux de la tresse. Ainsi une bonne qualité de ce dernier assure une transmission fiable. De même, la bonne qualité des connecteurs augmente la qualité de la transmission. Un signal se propage à l'intérieur d'un câble coaxial de la même façon que sur un autre fil (tel que les fils torsadés utilisés dans le système téléphonique).

Toutefois le câble coaxial possède deux importantes caractéristiques qui permettent de

transporter les signaux de télévision avec plus d'efficacité.

Premièrement, un câble coaxial permet de limiter les effets d'interférences électromagnétiques et des signaux entre eux.

Deuxièmement, un câble coaxial permet de transporter efficacement des signaux sur une large gamme de fréquence avec très peu d'atténuation.

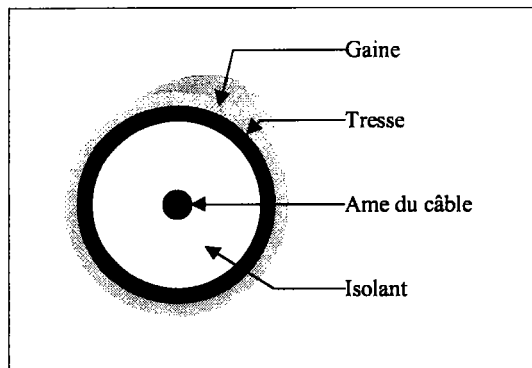


figure 10 : Coupe d'un câble coaxial

Mais le paramètre essentiel à connaître pour un câble coaxial ou pour tout autre fil, est l'atténuation induite sur les signaux à hautes fréquences et en particulier les signaux à faible puissance. Cette limitation s'exprime en décibels par unité de longueur. Cette atténuation augmente comme la racine carrée de la fréquence du signal.

La prise en compte de certains paramètres technologiques est nécessaire pour définir la qualité de la ligne de transmission. L'un de ces paramètres est la *capacité*. Ce dernier définit la quantité d'information qui pourra être supportée sur le canal. Cela s'exprime en bits par unité de temps.

En pratique, la capacité d'un canal est fonction des techniques de codage utilisées sur le signal, des caractéristiques du canal et particulièrement de la largeur de bande et du niveau de bruit. Cependant la relation fondamentale de Shannon exprime les limites de la capacité d'un canal bruité par :

$$C = W \log_2 \left(1 + \frac{P}{N} \right)$$

avec W la largeur de bande exprimée en nombre de cycles par unité de temps,

avec P/N le rapport signal sur bruit.

I.3.2 LES DIFFERENTS TYPES DE BRUIT

Le terme de bruit est utilisé en électronique pour représenter les fluctuations apparaissant dans les dispositifs, circuits ou systèmes électroniques.

Le bruit est un phénomène aléatoire : c'est une fluctuation, une variation autour d'une

valeur moyenne. Cette variation signifie que le bruit possède une composante spectrale. Le signal « bruité » se représentera alors par une raie principale et un nombre infini de raies plus petites allant des fréquences les plus basses aux fréquences les plus élevées. Aucune fréquence n'est privilégiée, mais dans le cas du câble coaxial, la bande 10 MHz à 30 MHz est très fortement bruité [11]. Cette gamme de fréquence constitue la bande allouée en voie descendante pour la diffusion sur télévision.

Une notation basée sur les propriétés statistiques du bruit permet d'effectuer la classification (voir figure 11) de tous les phénomènes de bruits impulsifs et autres opérant sur le câble [12].

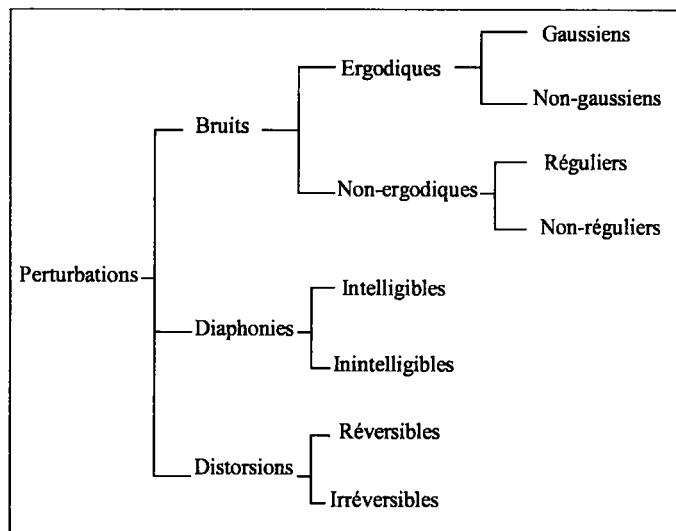


figure 11 : Classification des bruits

- *Bruit ergodique* : bruit dont les caractéristiques statistiques moyennes sont égales aux moyennes temporelles (au contraire du bruit impulsif).
- *Bruit non ergodique* : bruit caractérisé par une seule réalisation.
- *Bruit non ergodique régulier* : bruit pouvant être représenté par une fonction périodique. C'est le cas des systèmes d'allumage des moteurs à explosion. Ce type de bruit reprend de nombreux types de bruit d'origine artificielle car il représente un phénomène périodique dans le temps.
- *Diaphonie intelligente* : perturbations fortement cohérentes avec des signaux utiles d'autres canaux, ce phénomène se retrouve dans le cas où plusieurs utilisateurs disposeraient du même procédé de transmission. Ce bruit est analogue au bruit d'intermodulation.
- *Diaphonie inintelligible* : perturbations à faible cohérence avec les signaux utiles d'autres canaux, provoquées par des systèmes de transmission utilisant d'autres types de modulation (modulation de fréquence, d'amplitude,) ou plus simplement par des câbles

transportants de l'énergie.

- *Distorsion réversible* : perturbations cohérentes avec le signal utile correspondant à des distorsions de phase, d'amplitude dues par exemple à la voie de transmission suivant un paramètre connu l'évanouissement du signal.
- *Distorsion irréversible* : distorsions créées par un système non linéaire comme un écrêteur.

1.3.3 MODULATION ET TECHNIQUES A ACCES MULTIPLES

Dans la suite de ce paragraphe, trois techniques de base d'accès multiples sont présentées et comparées à travers leurs performances [11], [13]. Ces trois méthodes à Accès Multiples sont :

- TDMA (Time Division Multiplex Acces) permet d'obtenir un flux de données jusqu'à 25 Mbits/s. Cette technique affecte à chaque client un intervalle de temps pour utiliser le canal. Cette technique neutralise aisément les perturbations impulsives grâce au code correcteur d'erreur. Malheureusement le TDMA est excessivement vulnérable aux interférences à bande étroite. Aussi une solution combinant les deux techniques TDMA/FDMA est utilisée. Le combiné des deux techniques est moins vulnérable car moins d'interférences sont susceptibles d'apparaître dans une largeur de bande plus étroite.
- FDMA (Frequency Division Multiplex Acces) offre une porteuse propre à chaque signal. Cette technique se caractérise par l'usage de modulations multiporteuses et par l'affectation à chaque abonné des fréquences de ses canaux. Grâce à des allocations de fréquences intelligentes, la technique FDMA n'est presque pas affectée par les phénomènes d'interférences. De plus, la technique FDMA étale l'énergie de l'impulsion parmi les intervalles de fréquence, de telle manière qu'il faille une plus forte impulsion pour causer une erreur par rapport à un signal équivalent TDMA. Dans ce sens, le FDMA est plus résistant que le TDMA au bruit impulsif. Mais si ce dernier est assez important pour causer une erreur, alors il est probable qu'il affecte tous les canaux FDMA en même temps. En général cependant le FDMA fait une moins bonne utilisation de la bande passante (nécessité de laisser des espaces inter-canaux)
- CDMA (Code Division Multiplex Acces) associe à chaque abonné une séquence. Un récepteur CDMA étale le spectre d'une interférence pour obtenir une distorsion large bande. Aussi chaque interférence est supprimée par le gain du système à étalement de spectre. Mais un système à large bande admet plus d'interférences qu'un système à bande

étroite. Par conséquent aucune amélioration évidente n'est apportée par la technique d'étalement de spectre si les interférences sont uniformément distribuées sur le spectre. La méthode d'étalement de spectre a pour conséquence malheureuse qu'une interférence présente à la fréquence d'un utilisateur s'étend à tous les autres utilisateurs de CDMA pour les affecter de manière uniforme.

Une comparaison de la capacité d'un chemin de retour illustre parfaitement les différences fondamentales entre les trois techniques à accès multiples. Un résumé est donné dans la table 3, ci-dessous :

Techniques d'accès multiples	Paramètres	Type de modulation	Nbre de canaux à 64 kbits/s
TDMA	$\alpha = 0.5$ $\gamma = 1$	QAM 16 états	1040
CDMA Synchrone	$\alpha = 0.5$ $\gamma = 10$ $\rho = 260$	QAM 16 états	1040
CDMA Asynchrone	$\alpha = 0.5$ $\gamma = 10$ $\rho = 260$	BPSK	15 à 60
FDMA	$\alpha = 0.5$ $\gamma = 10$	QAM 16 états	1040

Table 1 : Comparaisons des capacités entre les différentes techniques

α : facteur d'excès de largeur de bande

ρ : facteur d'étalement

γ : facteur de crête

I.4 LES DIFFERENTES APPLICATIONS

Les réseaux câblés ont été conçus initialement pour permettre la diffusion d'un grand nombre de chaînes de télévision. Le potentiel des réseaux câblés modernes a fourni le support pour une large variété d'applications en communication [14]. Ainsi des applications différentes utilisent le support du câble pour transporter plusieurs types de données (ATM, MPEG,) à des débits variables (de 1.5 à 50 Mbits/s) [14], [16]. Cette prise en charge de toutes ces applications impose des contraintes des contraintes fonctionnelles propres à chacune des fonctions.

Les réseaux câblés ont été construits de manière à transmettre des signaux pouvant

atteindre des hautes gammes de fréquences et pouvant supporter des hauts débits de l'ordre de plusieurs gigabits par seconde. Une caractéristique des transmissions sur le câble est la non-symétrie des paramètres de la communication entre la voie montante et la voie descendante. Cette caractéristique a pour conséquence des allocations de bande passante plus étroite à la voie montante où le trafic est le moins chargé.

I.4.1 LA DIFFUSION VIDEO

Bien que les réseaux câblés aient toujours proposé la diffusion vidéo, de nouveaux aspects doivent être pris en compte pour offrir une meilleure diffusion des signaux numériques. En effet, la transmission des signaux vidéos compressés nécessite un débit de 2 à 3 Mbits/s sur le réseau câblé en utilisant des techniques de codage efficaces obligeant les canaux de diffusion à supporter dix fois plus de données. Toutefois, la diffusion de signaux numériques fournit aux opérateurs du câble la flexibilité due au numérique ainsi qu'une souplesse de contrôle intéressante pour des programmes de « pay per view ».

I.4.2 LA VIDEO A LA DEMANDE

Comme une substitution à la location de vidéo cassettes, la vidéo à la demande offre un nouveau type de service. Cette application nécessite des équipements spéciaux (modem, décodeur, vidéo serveur) [1]. Ces équipements techniques sont essentiellement liés aux techniques de télédiffusion. Cependant une étude plus fine est nécessaire au niveau de la voie montante pour satisfaire aux requêtes interactives. L'intérêt de la vidéo à la demande réside dans la souplesse du choix de la vidéo, du début et de l'arrêt du programme.

I.4.3 LA TELEVISION AVANCEE

Les standards proposés pour la télévision à haute définition (HDTV) fournissent des images vidéos de plus grande qualité que celles des applications existantes en télévision. Mais ce nouveau système nécessite un réseau câblé avec une plus grande capacité. En effet, une image non compressée sur de la HDTV (1240 par 720 pixels) exige un débit trois fois supérieur à celui d'un flot ordinaire de vidéo non compressée.

I.4.4 LE SON NUMERIQUE

Quelques systèmes développés sur le câble nécessitent la diffusion de son numérique. Pour obtenir un son numérique de la qualité d'un CD, cela exige un débit minimum de 0.7 Mbits/s

à 1.4 Mbits/s sous la forme non compressé. Ces applications, comme la diffusion de vidéos, sont délivrées par des services interactifs nécessitant une communication bidirectionnelle.

I.4.5 LA VIDEO CONFERENCE

Les applications actuelles de vidéo conférence utilisent la compression de l'information pour permettre des connections vidéos à double sens et pour travailler sur des canaux à bande limitée. Quelques applications de vidéo conférence peuvent travailler avec un modem standard de 28.8 kbits/s et avec une connexion Internet mais de telles applications sont marginales à ces vitesses. Les systèmes câblés devraient supporter une meilleur qualité vidéo à des débits compris entre 100 kbits/s à 1 Mbits/s.

I.4.6 CONNEXION AUX RESEAUX PC

Le câble est un moyen attractif pour accéder aux ordinateurs personnels et pour favoriser des connexions réseaux entre les PCs. En effet, le câble offre la possibilité de connexions dans des bandes passantes aux fréquences élevées et de partage de la bande sous forme d'allocation de canaux.

I.4.7 LES JEUX ELECTRONIQUES

Quelques services sur le câble consistent à délivrer des programmes informatiques pour des machines de jeux électroniques.

I.4.8 LES CARACTERISTIQUES TECHNIQUES

La tableau 2 résume les nécessités techniques de mise en place sur le câble pour satisfaire les différentes applications.

Application	Débit souhaité	Contraintes
Diffusion vidéo	2 Mbits/s en données compressées sur voie descendante	Nécessité de mécanisme à haute sécurité en vue des applications de "pay-per-view".
Vidéo à la demande	3 Mbits/s en données compressées sur voie descendante. Quelques kbits/s en voie montante.	Nécessité de mécanisme à haute sécurité en vue des applications de "pay-per-view".
Télévision avancée	Une dizaine de Mbits/s en données compressées sur voie descendante.	Similaire aux applications en télévision.
Son numérique	De l'ordre du Mbits/s en voie descendante.	Demande de réécoute similaire au service vidéo
Vidéo conférence	Débit variable autour de la dizaine de kbits/s sur deux voies en données compressées	La latence est un problème pour l'interaction. La confidentialité est importante.
Connexion aux réseaux PC	Le trafic sur deux voies autour de 10 kbits/s à 100Mbits/s ou plus.	Les caractéristiques du trafic et les besoins futurs sont fonctions de l'application
Jeux électroniques	Dépend de l'application.	Les jeux multi-joueurs interactifs exigent de faibles temps de latence sur des communications à deux voies.

tableau 2 : Les caractéristiques techniques des applications sur câble.

I.5 L'EVOLUTION DES TELECOMS VERS DES STRUCTURES RESEAUX

Pour délivrer des services sur des réseaux sur câble, un canal est alloué pour la voie descendante dans l'intervalle 5 MHz - 40 MHz réservé aux signaux de télévision et un autre canal dans l'intervalle 50 MHz - 90 MHz pour la voie de retour permettant une communication bidirectionnelle [4].

L'intérêt de développer une application sur une structure réseau réside dans la volonté de proposer une application multimédia. Le transport de données en utilisant plusieurs supports (câble, fibre optique, satellite, ...) est contraint aux respects des différentes caractéristiques des supports. Une structure réseau définie pour favoriser l'interopérabilité des applications alourdit sa complexité fonctionnelle pour satisfaire les contraintes liées aux supports de transmission et à la compatibilité entre les différents formats des données. Plusieurs formats de données peuvent partager la même ligne de transmission et un même format de données peut être utilisé sur plusieurs supports de transmission.

Pour gérer cette complexité, les fonctions sont hiérarchisées et classifiées et toutes les fonctions d'un même niveau hiérarchique sont regroupées dans une couche réseau. Plusieurs modèles de réseaux sont composés d'un nombre variable de couche (ex: Le modèle OSI « Open System Interconnection » a 7 couches, le modèle « Internet » a 4 couches, etc.) [13].

Un modem manipule les fonctions de la couche basse (couche physique) et éventuellement quelques fonctions des couches plus hautes [5] comme le montre la figure 12. Dans ce cas, le modem peut être assimilé à un « transceiver », soit un adaptateur de réseau.

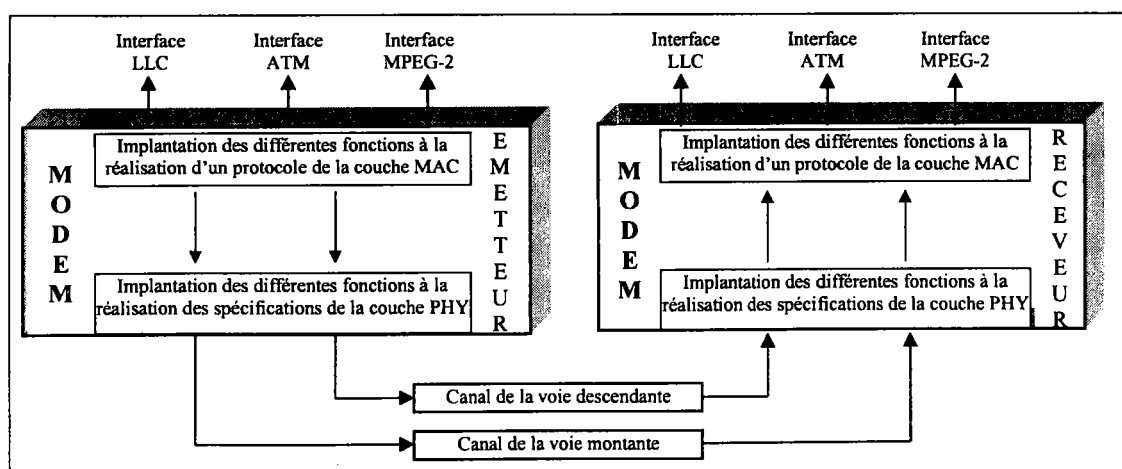


figure 12 : Contraintes réseaux pour application modem

I.5.1 LE RESEAU DAVIC (DIGITAL AUDIO-VISUAL COUNCIL)

Le réseau DAVIC (Digital Audio-Visual Council) [17] a pour but de développer des structures pour favoriser le succès des applications et des services ayant trait au domaine de l'audiovisuel. Dans cette optique, la mise en place en Août 1994 d'un consortium composé de membres DAVIC est constitué. Il regroupe plus de 200 compagnies provenant de 25 pays dans le monde et représentant tous les secteurs de l'industrie audiovisuelle :

- entreprises de production des ordinateurs,
- entreprises de composants électroniques,
- entreprises d'équipements de télécommunication,
- entreprises de services de CATV,
- entreprises de services de télécommunication,
- entreprises de services de radiodiffusion,

ainsi qu'un certain nombre de groupes administratifs et d'organisations pour la recherche.

Le succès grandissant de DAVIC provient de sa bonne vision du monde audiovisuel actuel, où les producteurs de contenu audiovisuel peuvent atteindre l'audience la plus large possible, où les utilisateurs ont un accès sans faille à l'information, et où les manufacturiers peuvent fournir sans restriction le matériel et les logiciels nécessaires à la production.

Le développement et la mise en place d'une norme tel que DAVIC a pour but de favoriser l'émergence des applications de l'audiovisuel numérique et des services associés dans le respect les accords internationaux. Ces accords régissent les différents protocoles existants et définissent des passerelles pour les faire communiquer entre eux.

La définition de spécifications techniques est nécessaire pour obtenir le maximum d'échanges d'opérations entre les pays et leurs différentes applications.

Les spécifications de la norme DAVIC sont présentées dans différentes versions : DAVIC 1.0, DAVIC 1.1, etc... Une présentation des spécifications techniques est détaillée dans le chapitre suivant.

I.5.2 LE RESEAU MCNS (MULTIMEDIA CABLE NETWORK SYSTEM)

MCNS [18] est un consortium se composant de Comcast, Cox, TCI, Time Warner, MediaOne, Rogers Cablesystems, CableLabs et Arthur D. Little qui s'est développé pour créer un standard rapide et "sûr" pour la transmission de données sur des réseaux câblées.

Le groupe chargé des spécifications est divisé en plusieurs équipes travaillant en collaboration étroite avec le comité des vendeurs de composants en télécommunication afin

de définir un réseau compatible avec les produits de l'industrie. Aujourd'hui plus de 35 sociétés ont planifié des produits compatible avec le réseau MCNS.

I.5.2.1 Les spécifications techniques

Voie descendante	Modulation	QAM 64 et 256 états
	Porteuse	6 MHz
	Débit	27 ou 36 Mbits/s
	Format de trame	MPEG-2
	Code correcteur d'erreur	Reed-Solomon
	Cryptage	DES
Voie montante	Modulation	QPSK et QAM 16 états
	Porteuse	Entre 200 kHz – 3.2 MHz
	Débit	320 kbits/s à 10 Mbits/s
	Code correcteur d'erreur	Reed-Solomon
	Cryptage	DES
Contrôle d'accès	MCNS	Basé sur la transmission en paquets, résolution des collisions et réservation par slots
Gestion	SNMP	Fournis par définitions MIB
Interface coté l'abonné	10 base-T	Planifié par USB et 1394 IEEE
Interfaces côté réseau	10 base-T 100 base-T ATM FDDI	

tableau 3 : Les spécifications du réseau MCNS.

Le tableau 3 résume les éléments clés des spécifications d'interface permettant de faire communiquer des données sur câble provenant de plusieurs systèmes [18]. Sur la base de ces spécifications, le réseau nord-américain a développé un standard de facto pour rendre compatible les différents câbles modem.

I.5.3 LE GROUPE IEEE 802.14

Le comité IEEE travaille actuellement pour définir un protocole standard pour les couches physiques et MAC (Medium Access Control) pour des applications sur réseaux câblés.

I.5.4 LES SPECIFICATIONS DE LA COUCHE MAC (MEDIUM ACCESS CONTROL)

Les standards IEEE concernent les couches physique et liaison (niveaux 1 et 2) du modèle OSI (Open Systems Interconnection) [19]. Nombre de ces standards sont regroupés sous le sigle IEEE 802. La majorité de ces standards sont repris par l'ISO sous la désignation ISO 8802.

Les fonctions du niveau 1 regroupent notamment:

- les codages et décodages des données;
- la synchronisation;
- la reconnaissance des trames.

Au niveau de la transmission sur le support physique deux techniques font actuellement l'objet de standards:

- la transmission en bande de base sur câble coaxial;
- la transmission large bande sur un câble de type CATV (câble d'antenne de télévision).

La couche MAC rend un service d'accès au support et fournit les messages de signalisation liés à un protocole de communication (Aloha, Polling, autre) pour contrôler les connections entre deux modems.

I.6 CONCLUSION

Ce chapitre a permis de mettre en évidence les différents éléments à considérer pour la réalisation d'un modem pour câble T.V. Actuellement, une application en télécommunication se situe à l'intersection de nombreux domaines : le traitement du signal, l'informatique, l'électronique et les structures en réseau. Chacun de ces domaines impose la présence de fonctions complexes pour respecter leurs contraintes. Aussi la réalisation d'une solution

technologique unique pour toutes ces applications s'avèrent très difficile. Dans le chapitre suivant nous présentons les solutions en restreignant ce cahier des charges à celui proposé dans DAVIC.

CHAPITRE II

TECHNIQUES DE

CONCEPTION D'UN MODEM

POUR CABLE T.V

II.1 INTRODUCTION

Dans ce chapitre, nous présentons les différentes solutions technologiques en vue de la réalisation d'un modem pour câble T.V à haut et moyen débit. Cette présentation s'accompagne d'une étude comparative des différentes solutions de manière à fournir les éléments architecturaux dédiés à ce type d'application. Cette étude s'appuie, de plus, sur la réalisation d'un prototype sous forme d'une carte modem.

En effet, le travail réalisé pendant la thèse s'intègre dans le cadre d'un projet Européen "EURICO" en collaboration avec la société I&TCom, lequel a pour objectif de réaliser un modem à 1.5 Mbits/s pour câble T.V.

Afin de proposer une solution innovante et actuelle, une étude approfondie est réalisée d'un modem numérique adapté aux structures complexes des réseaux câblés. Les spécifications des normes MCNS (voir I.5.2) et DAVIC sont sensiblement identiques pour les applications modems sur câble T.V., cette dernière étant déjà largement accessible au démarrage de cette étude-ci. Une étude détaillée de la norme DAVIC a permis l'extraction des spécifications concernant les applications modem sur câble T.V.

Une deuxième phase de travail a consisté à mettre en œuvre un prototype (dédié à l'application EURICO suscitée), sur la base des recommandations de la norme DAVIC, adaptées aux caractéristiques des transmissions européennes sur le câble.

Les limitations inhérentes à cette approche (architecture dédiée) nous ont conduit à la recherche d'une solution à la fois générale et efficace. Cette dernière solution est définie à travers une approche modulaire autorisant une adaptation facile aux spécifications de norme telles que DAVIC, MCNS et autres.

II.2 ARCHITECTURE GENERALE DU MODEM

En tenant compte des diverses contraintes évoquées dans le chapitre précédent (débits souhaités, support utilisé, applications ciblées, structure développée) une application modem pour câble TV se ramène à une chaîne fonctionnelle de modem à modem représentée par la figure 13 sous la forme d'un diagramme de blocs élémentaires.

Ainsi le modem se schématise du côté émetteur par cinq blocs fonctionnels (codage des données, mise en trame, codage du signal, modulation, transposition) et du côté récepteur par cinq autres blocs (filtrage, démodulation, décodage du signal, récupération des données utiles, décodages des données). Chacun de ces blocs fonctionnels peut être réalisé à l'aide de

plusieurs techniques différentes. Une chaîne fonctionnelle dédiée à une application modem doit posséder au moins une fonction de chacun des blocs. Plusieurs fonctions d'un même bloc peuvent être utilisées comme le CRC (Cyclic Redundancy Check) et le code détecteur et correcteur d'erreurs Reed-Solomon pour augmenter la sécurité de l'information.

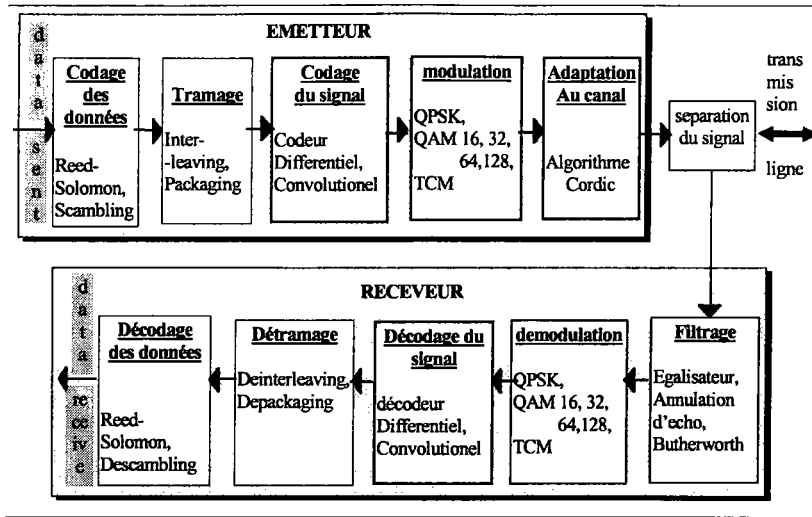


figure 13 : Blocs fonctionnels de la chaîne de transmission

II.3 LA NORME DAVIC : SPECIFICATIONS ET COMPLEXITE FONCTIONNELLE

La norme DAVIC (introduite au chapitre précédent) permet d'intégrer le transport de différents types de données (ATM, MPEG-2 et autres) sur un grand nombre de supports de transmission (fibre optique, satellite, ligne téléphonique, câble coaxial).

Dans la recommandation DAVIC, les contraintes des applications modem sur câble TV sont issues des spécifications liées à la HFC (Hybrid Fiber and Coaxial cable). Ces spécifications décrivent la structure complète des couches basses (couche physique et couche liaison) du réseau comprenant la structure d'encadrement, le codage du canal et la modulation de manière à acheminer les données entre deux points du réseau [17].

La couche physique et les spécifications encadrantes sont fournies pour deux modes de transmission sur le câble coaxial. Le premier mode permet de transmettre un flot d'information sur le câble coaxial de manière unidirectionnelle. Ce flot est essentiellement adapté au transport de données compressées sous forme MPEG-2.

Le deuxième mode est défini pour la transmission bidirectionnelle sur le câble coaxial. Ce mode autorise un flot d'information sur deux voies de communication l'une dite montante l'autre dite descendante.

A chacun de ces modes de transmission correspond des contraintes spécifiques de modulation, de codage, de débit et de structure de trames détaillées dans les paragraphes suivants.

II.3.1 SPECIFICATION DU MODE DE TRANSMISSION UNIDIRECTIONNEL

Le flot d'information associé à ce mode de transmission correspond au transport de données compressées à l'aide de l'algorithme MPEG-2.

Les spécifications DAVIC pour la couche physique présentent les différentes fonctions permettant la mise en place d'une interface pour la transmission unidirectionnelle sur les fréquences radio du câble coaxial (jusqu'à une bande passante de 1GHz). Cette interface est liée à une modulation QAM pour la transmission sur câble HFC (Hybrid Fiber Coaxial cable). De plus, deux structures de trames sont fournies pour transmettre les données sur la voie descendante. La première trame transporte des paquets au format MPEG2-TS et la deuxième est adaptée au transport de cellules ATM.

Une vue de l'allocation spectrale pour la couche physique sur câble coaxial est présentée sur la figure 14.

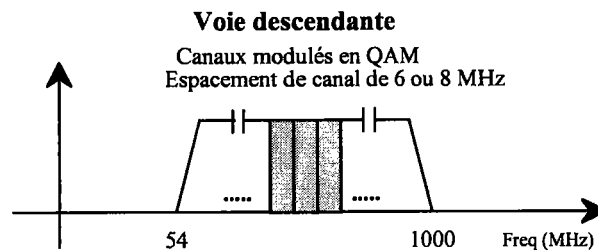


figure 14 : allocation spectrale

Les spécifications DAVIC pour le mode de transmission unidirectionnelle supportent des applications à débits variables (de 20 à 50 Mbit/s). Trois choix de modulation en quadrature d'amplitude et de phase (QAM 16 points, QAM 64 points et QAM 256 points) associés à une structure de trames configurable en fonction du débit autorisent un système à s'adapter aux caractéristiques de l'application. Le choix de la modulation QAM se justifie de par ses performances et une bonne efficacité spectrale.

II.3.1.1 Structure des trames

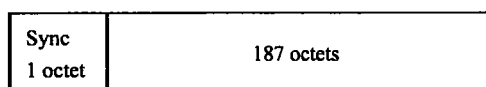
Dans ce mode de transmission la structure de trame permet d'encapsuler des données compressées au format MPG2-TS ou d'encapsuler des cellules ATM. Chacun de ces types de données étant régi par des organismes propres, la structure de la trame sera différente selon le choix des données à transmettre.

II.3.1.2 Flot d'information au format MPEG-2

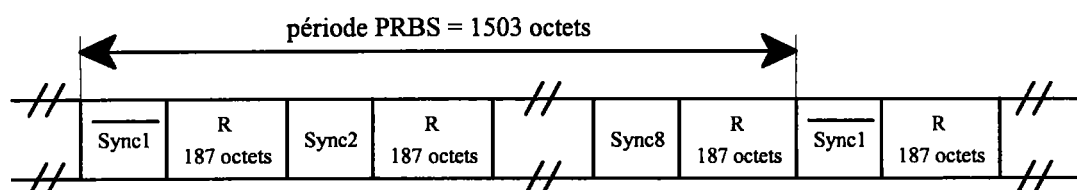
Le transport de flot de données MPEG2 est réglementé par l'ISO / IEC 13818-1. Dans ce cadre, les données MPEG2 sont constituées de paquets ayant 188 octets, avec un octet de synchronisation, trois octets d'en-tête contenant un identificateur de service suivi de 184 octets de MPEG-2.

II.3.1.3 Structure de la trame supportant du MPEG-2-TS

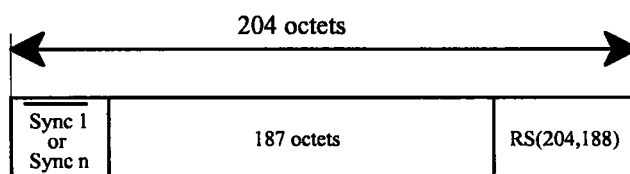
L'organisation pour la mise en trame est basée sur la structure de paquets MPEG-2 TS. Le système utilisé dans les spécifications DAVIC est montré sur la figure 15.



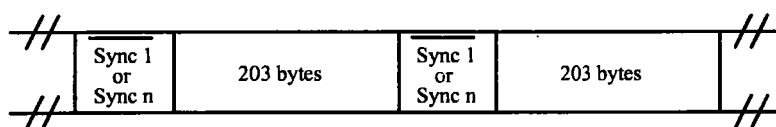
(a) Paquet MPEG2-T S



(b) packets brouillés: Les octets Sync et la séquence R brouillée



(c) Reed-Solomon RS(204,188, t=8)



(d) packets entrelacés avec un entrelacement de profondeur L

L=12 pour QAM 16 et 64, L=204 pour QAM 256

Sync 1 = not randomized complemented sync byte

Sync n = not randomized sync byte, n = 2...8

figure 15 : structure de la trame MPG2-TS

II.3.1.4 Flot d'information au format ATM (Asynchronous Transfer Mode)

Le flot d'information supportant des cellules ATM est réglementé par l'organisme UIT (Union Internationale des Télécommunications) [1]. Une cellule ATM est composée en paquets ayant 53 octets, avec une en-tête de cinq octets incluant un identificateur PID (Packet

Identifier) et un contrôle d'erreur HEC (Header Error Control) et de 48 octets de données ATM.

Les cellules ATM seront encapsulées dans la trame MPEG2 à titre de flot de données privé. Dans ce cas, le PID permet de distinguer un flot ATM d'un flot MPEG2.

II.3.1.5 La structure de trame encapsulant des cellules ATM

Une structure de trame spécifique au transport de cellules ATM est indispensable pour satisfaire les nombreuses applications réseaux sur le câble coaxial. Cette structure est dotée d'une organisation combinant la fonction d'entrelacement avec un alignement approprié des blocs FEC (Forward Error Correcting) de manière à résoudre le problème de synchronisation sur une nouvelle trame. Cette fonction est similaire à celle effectuée sur les trames des systèmes SDH (Synchronous Digital Hierarchy) et SONET (Synchronous Optical NETWORK).

Le transport des cellules ATM s'effectue sur la base des 188 octets de la trame MPEG2 en utilisant une organisation différente. Le format de la structure est montré sur la figure 16. Les étapes (b), (c) et (d) aperçues dans la figure 15 sont aussi appliquées dans le cas du transport de cellules ATM.

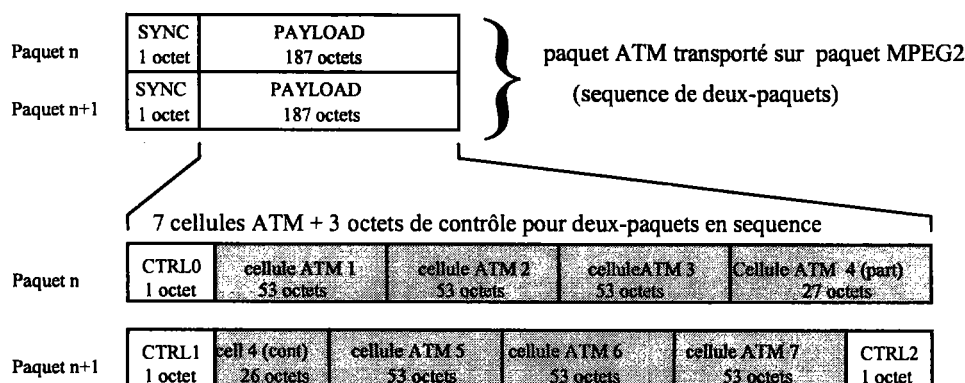


figure 16 : Trame ATM

Les différents champs de la trame SYNC, CTRL0, CTRL1, et CTRL2 sont définis et fixés à des valeurs particulières pour faciliter la synchronisation et respecter le format particulier de la trame MPEG2.

L'usage d'un même format pour transporter des cellules ATM ainsi que des paquets MPEG2 nécessite l'emploi de fonctions communes pour le respect des contraintes de la couche physique. L'utilisation de fonctions communes a l'avantage de figer une chaîne fonctionnelle quelle que soit l'application. Aussi le passage d'une application à une autre n'introduit pas de nouvelles contraintes fonctionnelles. Cependant le choix des fonctions doit être défini pour permettre de supporter l'application la plus coûteuse. Ce choix pénalise et

alourdi la chaîne fonctionnelle des autres applications. Cela se traduit par l'emploi de fonctions lourdes et coûteuses telles qu'un correcteur de haute sécurité HRM (High Reliability Marker), de nombreuses techniques de codage du canal (entrelacement convolutionnel, embrouilleur, Reed-Solomon), un convertisseur d'octet/symbole, et une modulation QAM configurable 16, 32 ou 256 points. Toutes ces fonctions détaillées dans la suite sont imposées à l'identique aux deux types de transports de données (MPEG2 et ATM).

II.3.1.6 Le correcteur de haute sécurité (HRM)

Le HRM (High Reliability Marker) est un outil transparent qui assure la robustesse de la synchronisation. Sa présence n'affecte en rien le fonctionnement du modem. Il est inséré au niveau des opérations de brouillage et d'espacement convolutionnel.

Un paquet de HRM est composé de 184 octets obtenus par un LFSR (Linear Feedback Shift Register). Il s'insère dans le flot de données à espacement régulier. Un paquet de HRM est transmis tous les 204 paquets de données. Le polynôme générateur $P(x)$ du LFSR est donné ci-dessous :

$$P(x) = x^8 + x^6 + x^5 + x + 1 \text{ avec une valeur initiale de } 10000000$$

Un paquet HRM est illustré par la figure 17:

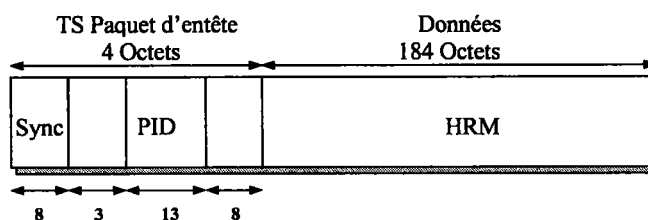


figure 17 : Trame HRM

II.3.1.7 Le codage du canal

Pour offrir un niveau approprié de protection vis à vis des erreurs lors de la transmission de données numériques sur le câble, un codage correcteur d'erreur FEC (Forward Error Correction) Reed-Solomon est appliqué. Une protection complémentaire contre les erreurs en salves est réalisé par un entrelacement des octets.

II.3.1.7.1 Embrouilleur

De manière à augmenter les transitions binaires, le flot de données en sortie du multiplexeur des paquets MPEG2 est embrouillé à travers la configuration décrite sur la figure 18.

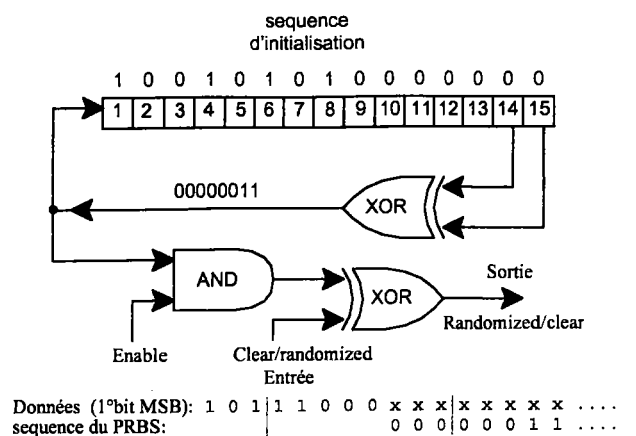


figure 18 : diagramme schématique de l'embrouilleur/désembrouilleur.

Le but de cet embrouillage aléatoire est de générer en sortie une séquence binaire avec le plus grand nombre possible de transitions "01". Pour cela, le flot de données est combiné à une séquence binaire pseudo-aléatoire PRBS (Pseudo Random Binary Sequence).

Le polynôme générateur du PRBS est le suivant:

$$P(X) = 1 + X^{14} + X^{15}$$

La séquence est initialisée dans les registres du PRBS à la valeur "1001010100000000" au début de chaque groupe de huit paquets.

II.3.1.7.2 Codeur Reed-Solomon

Le codage Reed-Solomon préconisé dans ce mode de transmission par DAVIC est appliqué sur chaque paquet MPEG2 effectuant la correction de 8 octets erronés au maximum par paquet. Ce traitement ajoute 16 octets de parité au paquet initial pour donner un mot de code de 204 octets (204,188,8).

Ce code Reed-Solomon est obtenu à partir des polynômes générateurs suivants :

- **Polynôme Générateur du code:** $g(x) = (x+\mu^0)(x+\mu^1)(x+\mu^2)\dots (x+\mu^{15})$
avec $\mu = 02\text{hex}$
- **Polynôme Générateur du champ:** $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

Ce code RS(204,188,8) est appelé un code réduit et est obtenu à partir du code complet de 255 octets. Toutefois, le calcul de ce code est obtenu sur la base du code complet dont les 51 octets finaux sont mis à zéros.

II.3.1.7.3 Entrelacement convolutionnel

Dans la spécification DAVIC une fonction d'*entrelacement convolutionnel* est appliquée aux paquets de données auxquels les bits du code correcteur d'erreurs ont été ajouté comme le montre la figure 15 (c). Cette fonction a pour but de mélanger les octets à l'intérieur de la

trame de façon à éviter en cas d'un fort niveau de bruit ponctuel, une transmission d'erreurs regroupées sur des octets adjacents. En effet, en cas de plus de huit octets erronés, le décodeur et correcteur d'erreur est inefficace. Le résultat est une trame entrelacée comme le montre la figure 15 (d).

Le traitement de cette fonction est basé sur la technique de Forney [7], [10] laquelle est compatible avec l'approche dite de Ramsey type III avec $I=12$ (pour une QAM de 16 et 64 points) et avec $I=12, 34$ ou 204 (pour une QAM 256 états).

L'entrelacement convolutionnel (voir figure 19) est composé de I branches, cycliquement connectées au flot des octets d'entrée par un sélecteur. Chaque branche représente un registre à décalage de type FIFO (First In First Out). Chaque cellule du FIFO a une capacité d'un octet. Le nombre de cellules de chaque branche est défini par la valeur : $(M * j)$

avec $M = N/I$,

avec $N = 204 =$ longueur de la trame,

où I est fonction de la QAM choisie

et où j est l'indice de la branche

Pour satisfaire les contraintes de synchronisation les octets "Sync" sont toujours acheminés vers la branche "0" de l'entrelacement convolutionnel ce qui correspond à un délai nul.

Le désentrelacement convolutionnel est similaire à l'entrelacement convolutionnel : seuls les indices des branches sont intervertis (i.e. $j = 0$ correspond au délai le plus long).

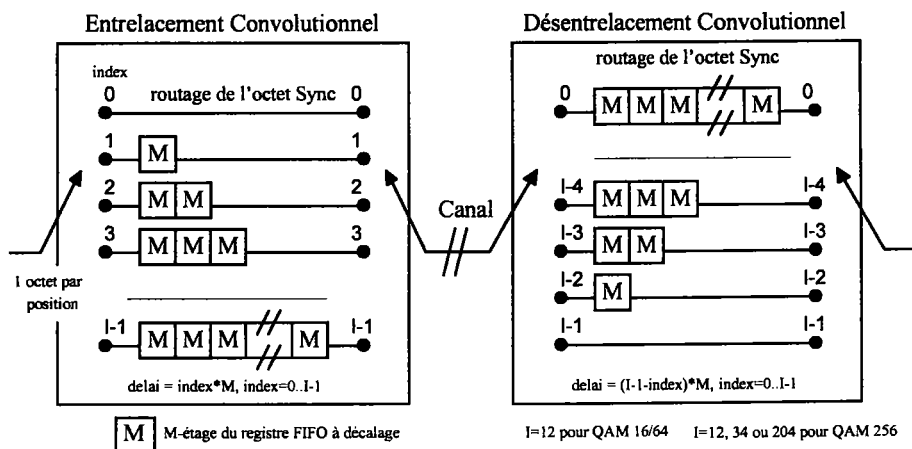


figure 19 : diagramme structurel de l'entrelacement et désentrelacement convolutionnel

II.3.1.8 Convertisseur octet / symbole

Après la fonction d'entrelacement convolutionnel, une conversion des octets en symboles doit être effectuée pour respecter le même système de représentation (en valeur I et Q) que l'on retrouve dans la constellation de la modulation QAM utilisé.

Les deux bits les plus significatifs MSBs de chaque octet sont traités par un codeur

différentiel de façon à obtenir une invariance rotationnelle de phase $\pi/2$ dans la constellation QAM. Le codage différentiel des deux MSBs est donné par l'expression booléenne suivante :

$$I_k = \overline{(A_k \oplus B_k)}(A_k \oplus I_{k-1}) + (A_k \oplus B_k)(A_k \oplus Q_{k-1})$$

$$Q_k = \overline{(A_k \oplus B_k)}(B_k \oplus Q_{k-1}) + (A_k \oplus B_k)(B_k \oplus I_{k-1})$$

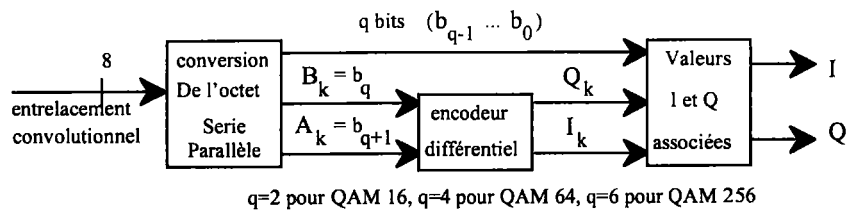


Figure 20 : Exemple d'implantation pour la conversion octet/symbole à l'aide d'un codeur différentiel des deux MSBs.

II.3.1.9 Quadrature Amplitude Modulation (QAM)

Bien que le modèle spectral soit identique, le niveau de performance de deux systèmes différents est fonction de la technique de modulation utilisée. En effet l'emploi d'un grand nombre de points pour un système avec une modulation QAM sera toujours plus performant qu'un système utilisant une modulation PSK. La raison essentielle étant que la distance entre deux points dans un système PSK est plus courte que celle obtenue entre deux points d'un système QAM comparable

La modulation QAM peut avoir plusieurs niveaux de valeurs discrètes. Les niveaux usuels sont la QAM 4 points, la QAM 16, 32, 64 et 256 points. Pour la spécification DAVIC, les niveaux sélectionnés sont QAM 16, 64 et 256 points. L'implantation de la (dé)modulation QAM est faite selon les spécifications du tableau 4 :

Modulation	QAM 16, 64, 256 avec invariance rotationnelle. Deux grades de niveaux QAM sont définis : Grade de niveau QAM A 16 et 64 points B 16 et 64 et 256 points Un émetteur devra au moins supporter un des niveaux QAM. Un récepteur doit pouvoir démoduler le grade A ou B
Espacement entre porteuse	Un espacement de 6 MHz ou 8 MHz est appliqué.
Gamme de fréquence	54 MHz – 1 GHz. Le récepteur doit opérer sur la totalité de la gamme
Débit	6 MHz : Le modem doit supporter au moins un débit compris entre 5-5.304 Mbaud. Pour les systèmes ayant un flux de contrôle, la valeur doit être multiple de 8 kbaud. (Information: La valeur recommandé est: 5.304 Mbaud) 8MHz : Le modem doit supporter au moins un débit compris dans 6-6.952 Mbaud. Pour les systèmes ayant un flux de contrôle, la valeur doit être multiple de 8 kbaud (Information : La valeur recommandé est 6.952 Mbaud)
Bruit de Phase	<-75 dBc/Hz à 1 kHz <-85 dBc/Hz à 10 kHz <-100 dBc/Hz à 100 kHz et plus
Codage du Signal	Codage à quadrant différentiel et codage de Gray dans le quadrant
Période de Modulation I/Q	$< 0.02 T$ (T = Période Symbole)
C/N à l'entrée du démodulateur (bruit blanc)	QAM 16,64 : ≥ 30 dB @ BER$<1 \times 10^{-12}$ (après correction d'erreurs) QAM 256 : ≥ 36 dB @ BER$<1 \times 10^{-12}$ (après correction d'erreurs) (i.e. 1 erreur pour 7 heures à 40 Mbit/s)
niveau de canal adjacent	≤ 3 dB (même types de canaux QAM) ≤ 6 dB (types différents de canaux QAM ; seulement pour les cas QAM 16/64 adjacent à une QAM 256)

tableau 4 : Spécifications pour la modulation QAM

II.3.1.10 Information sur les débits

Le tableau 5 donne des exemples de débits obtenus sur le câble en utilisant les diverses solutions QAM 16, 64 et 256 points.

Modulation employée	Débit usuel R_u (MPEG-2 couche transport) [Mbit/s]	Débit total R_t incluant RS(204,188) et HRM [Mbit/s]	débit par symbole sur Câble [Mbaud]	Largeur de bande occupée [MHz]
QAM 16	19.448	21.216	5.304	6.0
QAM 64	29.172	31.824	5.304	6.0
QAM 256	38.896	42.432	5.304	6.0
QAM 16	25.491	27.808	6.952	8.0
QAM 64	38.236	41.712	6.952	8.0
QAM 256	50.981	55.616	6.952	8.0
QAM 16 (HRM)	19.353*	21.216	5.304	6.0
QAM 64 (HRM)	29.030*	31.824	5.304	6.0
QAM 256 (HRM)	38.706*	42.432	5.304	6.0
QAM 16 (HRM)	25.367*	27.808	6.952	8.0
QAM 64 (HRM)	38.050*	41.712	6.952	8.0
QAM 256 (HRM)	50.732*	55.616	6.952	8.0

(*) Le débit présenté est le débit effectif excluant le HRM. Débit réel de MPEG-2-TS est le même sans

HRM, qu'avec insertion du HRM dans le multiplexage des MPEG-2-TS.

tableau 5 : Exemples de débits usuels R_u et de débits totaux R_t utilisant les débits par symbole recommandé (roll-off = 13% pour 6 MHz et roll-off = 15% pour 8 MHz).

II.3.2 BILAN ET CONCLUSION

Le mode de transmission unidirectionnel des spécifications DAVIC est principalement utilisé par les applications vidéos. Ces applications nécessitant des hauts débits de communication (jusqu'à 50 Mbit/s) et une haute sécurité de l'information, la chaîne fonctionnelle résultante de modem à modem est composée de fonctions de codage, de techniques de modulation et d'un format de trame dont l'élaboration est très coûteuse.

En effet, le codage et le décodage du code cyclique (voir I.2.2.4) Reed-Solomon (204, 188, 8) s'effectue à l'aide de divisions polynomiales successives très coûteuses. De même, l'utilisation d'une modulation QAM à 16, 32 ou 256 fait appel à un algorithme très contraignant pour l'identification des points de la constellation : le calcul des valeurs exactes de I et Q est très coûteux.

Seul un circuit spécifique ou un ensemble de composants appropriés sont en mesure de supporter l'enchaînement de l'ensemble de ces fonctions critiques tout en respectant le débit souhaité.

II.3.3 SPECIFICATION POUR LE MODE DE TRANSMISSION BIRECTIONNEL

L'utilisation du mode de transmission bidirectionnelle sur câble coaxial est destinée aux applications nécessitant deux voies de communication (par exemple un serveur et un ou plusieurs clients). Une voie est utilisée pour la communication d'un serveur vers un ou les clients (appelée descendante) et une autre pour la communication d'un client vers le serveur (dite montante). Pour ce mode de transmission, l'interface physique doit définir des contraintes fonctionnelles appropriées à chacune des voies de communication.

Dans ce mode de transmission, les spécifications DAVIC imposent l'utilisation de la modulation QPSK (Quaternary Phase Shift Keying). La structure de trame est ici adaptée au transport de cellules ATM.

L'émetteur et le récepteur en mode bidirectionnel devront respectivement opérer dans les gammes de fréquences indiquées sur la Figure 21.

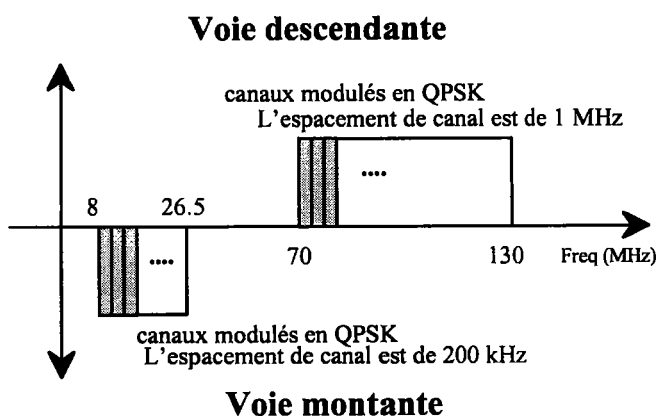


Figure 21 : allocation spectrale pour la transmission bidirectionnelle.

II.3.3.1 Spécification de l'interface physique pour la voie descendante

Les spécifications DAVIC pour le mode bidirectionnel autorisent deux flots d'information différents sur la voie descendante, l'un traitant les données à un débit de 1.544 Mbit/s, l'autre travaillant à un débit de 3.088 Mbit/s.

Dans les deux cas, l'information véhiculée sur la voie descendante utilise la modulation QPSK et une structure de trame spécifique dédiée au transport de cellules ATM. Ce choix de trame n'interdit cependant pas le transport d'autres types de données si le format proposé est respecté.

La modulation QPSK a été choisie au vu de ses performances face aux erreurs de transmission, de son efficacité spectrale.

II.3.3.1.1 La modulation QPSK (Quaternary Phase Shift Keying)

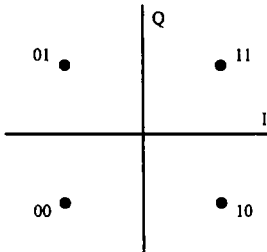
Débit de transmission	1.544 Mbit/s pour le grade A 3.088 Mbit/s pour le grade B Un démodulateur QPSK doit supporter le grade A ou B.															
Modulation	QPSK avec codage différentiel.															
Espacement de canal	1 MHz pour le grade A 2 MHz pour le grade B															
Intervalle entre porteuse	250 kHz															
Embrouilleur	Après l'ajout des octets FEC, toutes les données à 1.544 Mbit/s ou à 3.088 Mbit/s traversent un LFSR à six pour augmenter la distribution aléatoire des "1" et des "0". Le polynôme générateur est: $x^6 + x^5 + 1$.															
Le codage différentiel	Les octets arrivant dans le convertisseur octet/symbole sont découpés en quatre paires de bits, chaque paire génère un symbole QPSK. A chaque paire de (A,B) la phase générée change suivant la règle suivante: <table><tr><td>A</td><td>B</td><td>Phase</td></tr><tr><td>0</td><td>0</td><td>Aucune</td></tr><tr><td>0</td><td>1</td><td>+ 90 degrés</td></tr><tr><td>1</td><td>1</td><td>180 degrés</td></tr><tr><td>1</td><td>0</td><td>- 90 degrés</td></tr></table>	A	B	Phase	0	0	Aucune	0	1	+ 90 degrés	1	1	180 degrés	1	0	- 90 degrés
A	B	Phase														
0	0	Aucune														
0	1	+ 90 degrés														
1	1	180 degrés														
1	0	- 90 degrés														
Constellation du Signal	Les sorties I et Q repèrent l'état des phases comme suit: 															
La gamme des fréquences	70 à 130 MHz.															
Suppression des porteuses	> 30 dB															

tableau 6 : Spécifications pour la modulation QPSK en voie descendante

La modulation QPSK est utilisée comme un moyen de transmission d'informations numériques sur des lignes de transmission (fibre, câble). La technique de modulation est un cas particulier de la modulation de phase PSK (voir I.2.2.1.3). La modulation QPSK possède quatre niveaux correspondant à quatre déplacements différents de la phase. Une représentation du signal en quadrature implique le traitement de la phase comme une combinaison linéaire d'un cosinus et d'un sinus avec une phase initiale nulle.

Les systèmes basés sur la modulation QPSK requièrent l'emploi d'un codeur différentiel et du détecteur associé. En l'absence de ce détecteur, le récepteur n'a pas de moyen pour définir un point de référence à la réception des signaux. (voir I.2.2.1.3 modulation de phase).

L'implantation du (dé)modulateur QPSK répondant aux spécifications DAVIC est donné dans le tableau 6.

II.3.3.1.2 Structure de la trame

L'organisation de la trame est basée sur la structure dite de supertrame et sur la structure des cellules ATM.

II.3.3.1.2.1 Format de la supertrame

La structure de la supertrame correspond à un format de trame étendue comme l'illustre la figure 22.

Le flot de bit est découpé en blocs de 4632 bits représentant la capacité de la supertrame. Chaque supertrame est composée de 24*193-bits. Ces 193 bits sont constitués d'un bit d'entête (OH) et de 24 octets (192 bits) de données utiles.

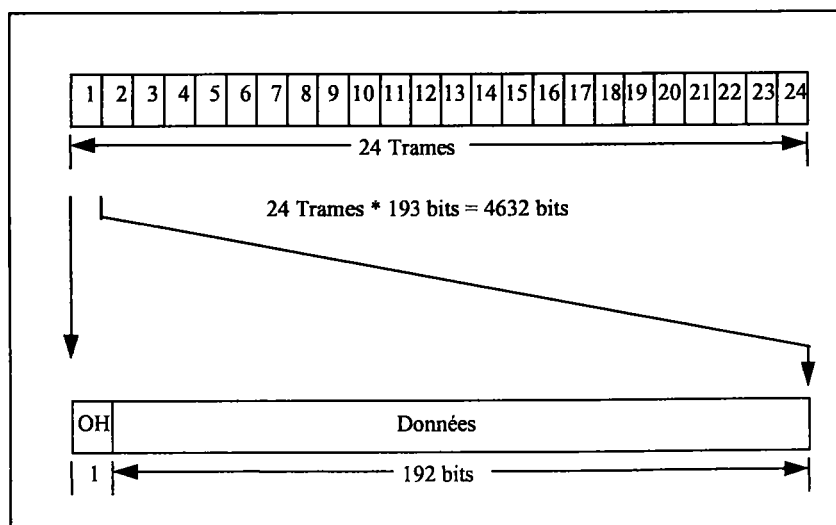


figure 22 : Structure de la supertrame

II.3.3.1.2.2 Organisation des bits d'en-tête

Dans la supertrame, les 24 bits d'en-tête sont placés à des positions précises comme le montre le tableau 7 et sont utilisés pour fournir des informations de contrôle (signalisation).

Numéro du paquet	Numéro du bit	Bit d'en tête	Données utiles (192 bits)	
1	0	M1		♦ Slot
2	193	C1		
3	386	M2		
4	579	F1 = 0		
5	772	M3		
6	965	C2		
7	1158	M4		
8	1351	F2 = 0		
9	1544	M5		♦ Slot
10	1737	C3		
11	1930	M6		
12	2123	F3 = 1		
13	2316	M7		
14	2509	C4		
15	2702	M8		
16	2895	F4 = 0		
17	3088	M9		♦ Slot
18	3281	C5		
19	3474	M10		
20	3667	F5 = 1		
21	3860	M11		
22	4053	C6		
23	4246	M12		
24	4439	F6 = 1		

tableau 7 : Structure repérant les bits d'en-tête

Ces 24 bits sont regroupés en deux mots de six bits et un mot de 12 bits. Un mot de six bits

(F1-F6) sert à l'alignement de la supertrame ; la valeur des six bits est fixée à ($F1 = 0, F2 = 0, F3 = 1, F4 = 0, F5 = 1, F6 = 1$).

Le deuxième mot de six bits (C1-C6) est utilisé pour transmettre la valeur du code de contrôle CRC (Cyclic Redundancy Check). La séquence C1-C6 représente la valeur du reste après division de la supertrame par le polynôme générateur x^6+x+1 . Cette fonction permet de vérifier la bonne réception de la supertrame en comparant la valeur du reste transmis C1-C6 à la valeur du reste calculé à la réception.

Le mot de 12 bits (M1-M12) fournit des informations pratiques au récepteur. Les bits M1, M5 et M9 servent de top d'horloge pour synchroniser le début des trames montantes. Le bit M12 spécifie le grade A (M12="1") ou B (M12="0") de la transmission. Les bits M2, M3, M4, M6, M7, M8, M10 et M11 forment un mot dont la valeur fixe le numéro du prochain slot de remontée.

II.3.3.1.3 Codage du canal

En suivant le principe vu dans le paragraphe II.3.1.7.2, un codage Reed-Solomon est exécuté sur la base du code court RS(55,53,1). Ce code est obtenu en utilisant les polynômes suivants :

Polynôme générateur : $g(x) = (x + \mu^0)(x + \mu^1)$, avec $\mu=02\text{hex}$

Polynôme de champs : $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

Un entrelacement convolutionnel sur le principe du paragraphe II.3.1.7.3 est appliqué aux cellules ATM encapsulées dans la supertrame. L'entrelacement choisi est composé de 5 lignes (I=5) sur une périodicité de 55 octets.

II.3.3.1.4 Structure de la cellule ATM

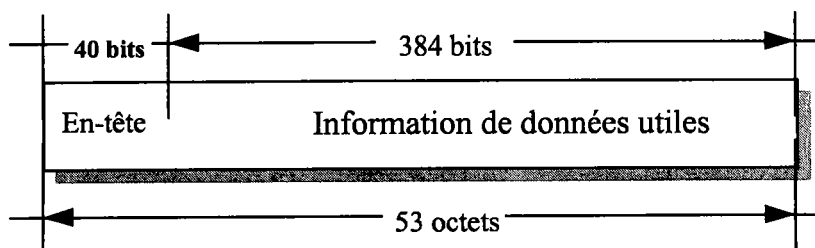


figure 23 : Structure de cellule ATM

Le format de chaque cellule ATM est illustré sur la figure 23. Cette structure et le codage des champs sont adaptés à la norme [ITU-T I361].

II.3.4 SPECIFICATIONS DE LA VOIE MONTANTE

Le transport des données sur la voie montante est réalisé en combinant la modulation QPSK et la structure de multiplexage par division de temps (TDMA).

II.3.4.1 La modulation QPSK en voie montante

Débit de transmission	Grade	Débit
	A	256 kbit/s
	B	1.544 Mbit/s
	C	3.088 Mbit/s.
Modulation	QPSK avec codage différentiel	
Espacement de canal	200 kHz pour grade A (256 kbit/s) 1 MHz pour grade B (1.544 Mbit/s) 2 MHz pour grade C (3.088 Mbit/s)	
Intervalle entre porteuses	50 kHz pour grade A, grade B et grade C	
Mot Unique	Le mot unique est de 4 octets : CC CC CC 0D hex.	
Embrouilleur	Le mot unique est transmis non embrouillé. Le brouillage est appliqué sur les 53 octets de données utiles et les 6 octets FEC. L'embrouilleur est exécuté en effectuant une addition modulo-2 avec une séquence pseudo-aléatoire. Le polynôme générateur est $x^6 + x^5 + 1$.	
Le codage différentiel	Voir tableau 6	
Constellation du Signal	Voir tableau 6 La constellation est utilisée pour la détection du mot unique lequel n'est pas codé.	
Gamme de fréquence	8 – 26.5 MHz.	
Suppression des porteuses	> 30 dB	

tableau 8 : Spécifications pour la modulation QPSK en voie montante

Une vue générale de la modulation QPSK a été fournie dans la section II.3.3.1.1 présentant les spécifications de la voie descendante. La modulation QPSK en voie montante reprend de nombreuses caractéristiques de celles fournies en voie descendante en y ajoutant un débit supplémentaire (256 kbit/s) et des fonctions de codage plus performantes dues au problème du

bruit plus important que celui rencontré dans la bande 25-30 MHz. Le tableau 8 liste l'ensemble des caractéristiques de la modulation QPSK à utiliser lors d'une transmission sur la voie montante.

II.3.4.2 Accès multiples par division de temps TDMA (Time Division Multiple Access)

La technique TDMA (voir I.3.3) repose sur l'utilisation de slots synchronisés sur l'horloge commune, celle de l'émetteur de la voie descendante. Cette synchronisation est faite par l'intermédiaire des bits M1, M5 et M9 de la supertrame descendante. Afin d'éviter le chevauchement des sources différentes dans le flot montant, des retards calculés différents seront appliqués avant émission. La durée cumulée du retard et du temps d'émission d'un slot ne doit pas dépasser celle séparant M1 de M5 (ou M5 de M9, ou M9 de M1).

II.3.4.3 Structure du slot sur voie montante

Le débit des slots en voie montante est de 3000 slots/s quand le débit des données est de 1.544 Mbit/s et de 500 slots/s quand le débit est de 256 kbit/s.

La structure de slot est la suivante :

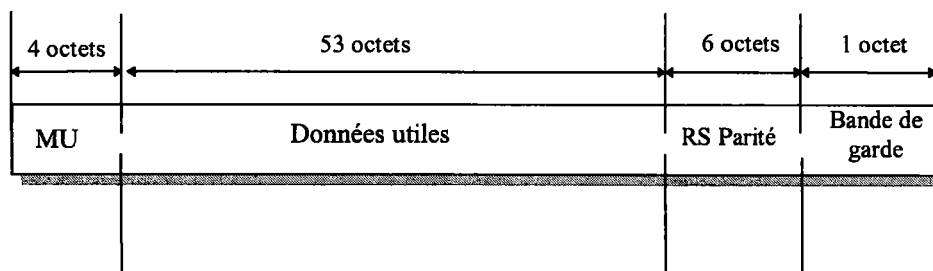


figure 24 : Structure du slot en voie montante

Le mot unique (MU) de 4 octets permet au système d'utiliser la méthode d'acquisition dite en "rafale". Les données utiles, de 53 octets de longueur, contiennent celles au format ATM. Un codage de ces données est réalisé à l'aide du code Reed-Solomon réduit RS(59,53,3) ce qui assure la détection et la correction d'un maximum de 3 octets erronés par paquet de 53 octets. Ce codage est effectué sur la base des polynômes suivants :

Polynôme générateur : $g(x) = (x + \mu^0)(x + \mu^1)(x + \mu^2) \dots (x + \mu^5)$, avec $\mu=02\text{hex}$

Polynôme de champ : $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

L'octet de bande de garde sans émission fournit un espacement de sécurité pour éviter un recouvrement avec le slot adjacent.

II.3.4.4 La signalisation de contrôle de la voie montante

La signalisation de contrôle de la voie montante est fournie par un canal de service utilisant

la modulation QPSK avec gestion des conflits (protocole Aloha) [13].

II.3.4.4.1 Le protocole Aloha

L'Aloha est utilisé pour la gestion des conflits de transmission sur un canal partagé. Pour le système DAVIC, ce protocole est utilisé pour résoudre les conflits d'accès inhérents à la connexion d'un nouvel abonné.

La technique Aloha fournit alors une solution pour palier le problème induit par des transmissions simultanées

Après la transmission d'un paquet de messages, l'émetteur doit déterminer si le signal a été reçu avec succès. Pour cela un accusé de réception est renvoyé à l'émetteur.

Un conflit se produit si deux ou plus émetteurs essaient de transmettre leur paquets de messages durant la même tranche de temps. Un conflit sera reconnu si l'émetteur ne reçoit pas l'accusé de réception. Dans ce cas l'émetteur sera conduit à reproduire la procédure.

II.3.5 BILAN ET CONCLUSION

Le mode bidirectionnel est essentiellement destiné aux applications Client/Serveur. Pour ce type d'applications, les spécifications DAVIC sont moins exigeantes en terme de débit (jusqu'à 3.088 Mbit/s) mais imposent la gestion de deux voies de communications (voie montante et voie descendante). Aussi les fonctions de la chaîne fonctionnelle doivent prendre en compte les phénomènes des bruit persistant dans la gamme 10-30 MHz de la voie descendante [11].

A l'instar du mode unidirectionnel des fonctions de codage tel que Reed-Solomon, entrelacement convolutionnel, embrouilleur ou CRC sont introduites dans la chaîne fonctionnelle en mode bidirectionnel avec les mêmes effets négatifs sur les performances. En effet toutes ces fonctions sont réalisées par des algorithmes ou des structures complexes ralentissant le traitement. La modulation QPSK et la transposition en fréquence de la porteuse sont de plus combinées sur chaque modem pour effectuer une communication sur plusieurs canaux. Ce dispositif constitue une source de problèmes, tant en termes de performance qu'en termes de réalisation.

II.3.6 LES FONCTIONALITES DE LA COUCHE MAC (MEDIA ACCESS CONTROL)

Ce paragraphe contient les spécifications du protocole de sous-couche MAC (Media Access Control) pour une communication entre la tête de réseau et ses clients sur câble HFC.

II.3.6.1 Modèle de référence MAC

Le flot de messages nécessaire au contrôle d'accès entre le serveur et le client se divise en trois catégories, les messages d'initialisation, les messages de reconnaissance et d'allocation, et les messages de gestion des liens et des connexions.

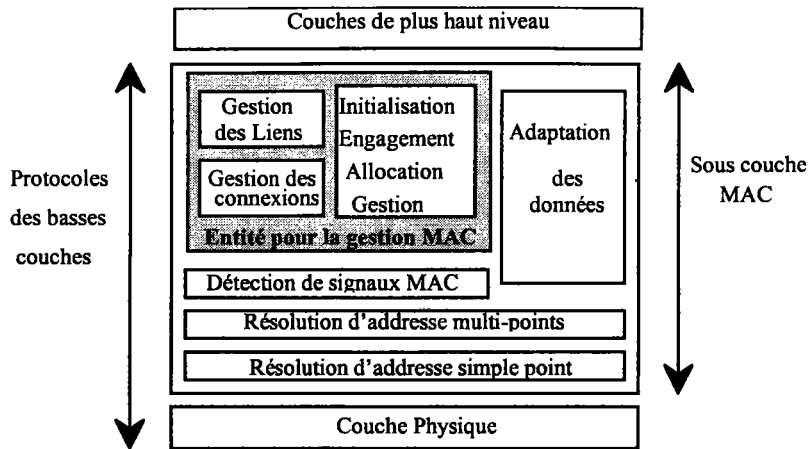


figure 25 : Modèle de référence MAC

(source IEEE 802.1F LAN et MAN)

Le protocole de couche MAC supporte la communication sur plusieurs canaux en voie descendante et en voie montante. Une fréquence hors bande additionnelle en flux descendant sert lors de la procédure de connexion. Une fois la connexion établie, un "canal de service" en voie montante sera utilisé par tous les clients.

II.3.6.2 Transport de l'information en sous-couche MAC

Dans DAVIC, la structure transportant l'information de la sous-couche MAC est conforme au protocole AAL5 (ATM Adaptation Layer 5), comme le spécifie [ITU-T I363]. Tous les messages MAC en flux montant sont restreints à un format unique de trame. La cellule MAC a un format de 40 octets de longueur.

Le format d'une trame de données MAC est illustré dans le tableau suivant :

MAC_message (){	Bits	Octets	Numéro du bit / Description
Configuration_Message		1	
Version_Protocole	5		7..3:{enum}
Indicateur_Syntaxe	3		2..0:{enum}
Type_Message	8	1	
if (Indicateur_Syntaxe == Adresse_MAC Incluse) {			
Adresse_MAC	48	6	
}			
Eléments _Information_MAC ()		N	
}			

Version Protocole

Protocol_Version est composé de 5 bits de type énuméré utilisé pour identifier la version courante de MAC.

Indicateur Syntaxe

Indicateur_Syntaxe est constitué de 3-bit de type énuméré qui indique le type d'adressage contenu dans le message MAC. Les messages en flux montant auront toujours leur adresse MAC incluse.

Adresse MAC

Adresse_MAC représente le mot d'adresse MAC de 48 bits unique d'un client.

II.3.6.3 Les différents types de messages MAC

Tous les types de messages MAC sont énumérés dans le tableau 9. Les types de message MAC sont divisés en trois groupes les messages d'initialisation, d'allocation et de reconnaissance, les messages pour établir une connexion et les messages de gestion des liens. Les messages MAC sont envoyés en utilisant soit un adressage multipoints (vers tous les clients) soit un adressage monopoint (vers un seul destinataire). Quand on utilise un adressage vers un seul destinataire, l'adresse MAC est indispensable pour identifier le client.

Nom du message	Type d'adressage
Message d'initialisation, d'allocation et de reconnaissance	
Message d'allocation de canal	Multipoints
Message de configuration par défaut	Multipoints
Message de demande de reconnaissance	Multipoints
Message réponse de reconnaissance	Monopoint
Message de calibrage	Monopoint
Message réponse de calibrage	Monopoint
Message d'initialisation complète	Monopoint
Messages pour établir une connexion	
Message pour se connecter	Monopoint
Message réponse de connexion	Monopoint
Message de demande de réservation	Monopoint
Message de réservation reconnue	Multipoints
Message pour confirmer la connexion	Monopoint
Message pour libérer la connexion	Monopoint
Message de réponse pour libérer la connexion	Monopoint
Message d'attente	Monopoint
Message d'accord pour une réservation	Multipoints
Message pour une réservation ID	Monopoint
Message de réponse pour la réservation ID	Monopoint
Messages de gestion des liens	
Message de Contrôle de la transmission	Monopoint
Message de nouvelle allocation	Monopoint
Message de réponse de gestion du lien	Monopoint
Message de vérification d'état	Monopoint
Message de réponse de l'état	Monopoint

tableau 9 : Liste des messages MAC

II.3.7 BILAN ET CONCLUSION

La prise en compte par le modem des trames de signalisation spécifiques à la couche MAC augmente la complexité fonctionnelle du modem. Parallèlement au traitement des données

utiles, chaque modem prend en charge le pilotage d'une fréquence supplémentaire, la définition des trames adéquates et la gestion de la communication entre le serveur et les clients. L'exécution d'un algorithme adapté au protocole de communication de la couche MAC est souvent coûteux en temps. Toute la difficulté consiste à effectuer le traitement MAC et celui des données utiles en respectant le débit imposé.

II.4 REALISATION D'UNE SOLUTION CABLEE

II.4.1 VUE GENERALE

Dans le cas général, l'application modem est représentée par une chaîne fonctionnelle complexe réalisant tous les types d'applications. Or DAVIC définit pour chaque application de nombreuses contraintes. La réalisation de l'application modem s'effectue en déterminant une chaîne fonctionnelle en fonction du mode de transmission souhaité, du débit désiré et du type de données à transporter. Le cahier des charges de la norme DAVIC se heurte cependant à la spécificité des fonctions critiques (CRC, Reed-Solomon, modulation QAM 256 points, modulation QPSK, etc.) à inclure dans la chaîne fonctionnelle.

En effet, chaque fonction critique (CRC, Reed-Solomon, modulation QAM 256 points, modulation QPSK, etc.) exige un dispositif adapté à son traitement qui satisfasse les contraintes de temps imposés par les applications des hauts ou moyens débits.

De ce fait, la première approche de conception consiste à reproduire la chaîne des blocs fonctionnels au travers d'un jeu de composants où chacun réalise une ou plusieurs des fonctions souhaitées. Cette approche a pour finalité le développement d'une carte réseau ou d'un composant permettant le raccordement à un réseau DAVIC. La même approche est possible pour un réseau MCNS. De plus, l'intégration de composants "hardware" assure à cette solution un haut niveau de performance en terme de vitesse d'exécution, indispensable pour le respect des hauts débits des différents modes de transmission.

Pour des raisons de coût en surface, le développement de composants dédiés à chaque fonction permet difficilement d'envisager la réalisation de toutes les chaînes fonctionnelles pour chaque application modem.

De nombreuses sociétés sont impliquées dans l'élaboration des normes DAVIC ou MCNS au sein de consortiums. Aussi des composants compatibles avec les différentes normes apparaissent sur le marché des télécommunications. Parmi celles-ci la présence de grands groupes comme Com 21, Motorola, Zénith, Hewlett Packard, IBM, Intel, Lucent, AT&T et

autres témoignent de l'attrait de ces normes dans les secteurs informatiques, électroniques et des télécommunications.

Dans cette optique la compagnie Com 21 propose le composant BCM 3118B réalisant une modulation QAM 64 et 256 points respectant les contraintes DAVIC coté client. Ce composant peut aussi jouer le rôle de démodulateur. Un codeur correcteur d'erreur est inclus ainsi qu'un convertisseur numérique/analogique.

II.4.2 LIMITATION ET PROBLEMATIQUE

Dans cette approche, l'application modem est réalisée grâce à un ensemble de composants spécifiques. L'avantage de cette approche réside dans le fait que la réalisation de composants sous une forme câblée permet un traitement performant de chacune des fonctions dans le respect des contraintes de temps. Les limites de cette solution purement "hardware" résident dans son absence de portabilité à d'autres types d'application et de souplesse face à des modifications de paramètres.

De plus les stratégies des conceptions actuelles tendent à intégrer l'application complète sur une seule puce. Cette approche confrontée à un problème de coût en surface satisfait difficilement les contraintes de la stratégie du "system on chip".

II.5 CONCEPTION D'UNE CARTE AUTOUR D'UN DSP

Une deuxième approche de conception consiste à reproduire la chaîne des blocs fonctionnels en intégrant un processeur DSP (Digital Signal Processing) lequel orchestrera les différentes actions à réaliser de l'application modem complète. Comme pour l'approche précédente le but est encore le développement d'un modem offrant la compatibilité au réseau DAVIC. La présence d'un DSP à l'intérieur de la carte permet une plus grande souplesse dans la conception du modem. Ainsi certaines fonctions seront implantées sous forme logicielle ("software") assurant à cette solution une grande souplesse à l'inverse de l'approche précédente tant au niveau de la mise à jour des fonctions qu'au niveau des configurations en cours de fonctionnement.

Cette option de conception a fait l'objet d'une étude complète aboutissant à la réalisation d'un prototype. Cette démarche s'inscrit dans le cadre d'un projet Européen EURICO associant la société I&Tcom et d'autres partenaires et dont le but est de réaliser un modem pour câble T.V autour du processeur DSP d'HyperStone.

II.5.1 PRESENTATION DU PROTOTYPE REALISE

Le projet Européen sur lequel s'intègre la réalisation de notre prototype impose des contraintes de débit de 1.5 Mbit/s et de transmission en mode bidirectionnel au modem sur câble TV. Le choix d'une solution permettant la compatibilité aux normes du réseau DAVIC a été dicté par la volonté d'obtenir une solution innovante et souple face aux futures applications. Ainsi les contraintes fonctionnelles de notre prototype sont inspirées des spécifications DAVIC en mode bidirectionnel sur câble TV dans le cas d'un débit de 1.544 Mbit/s définies dans le paragraphe II.3.3. Une vue fonctionnelle du modem est illustrée sur la figure 26.

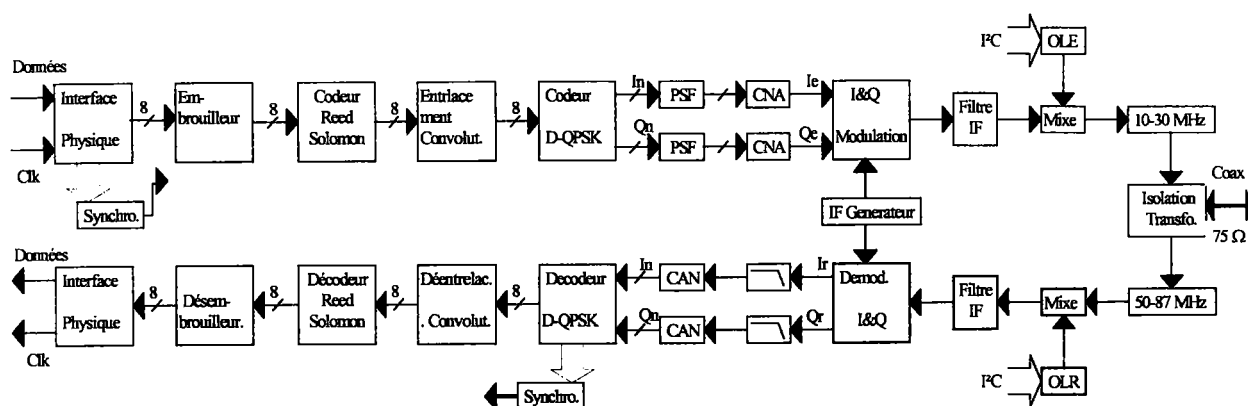


figure 26 : Vue fonctionnelle du prototype

Pour respecter le cahier des charges imposé par le projet Européen (réalisation d'un prototype à la fin de la première année d'étude), les phases de conception et de validation du fonctionnement ont été soumises à de strictes contraintes de calendrier. La réalisation de l'application modem a donc été menée en parallèle sur les deux parties numérique et analogique.

Une première phase a consisté à faire l'étude de toutes les fonctions numériques (CRC, brouillage, (dé)codage Reed-Solomon, (dé)entrelacement convolutif) proposées dans la spécification DAVIC et requise dans notre application. Cette étude a permis de définir des algorithmes adaptés à chaque fonction.

Une deuxième phase a eu pour but d'implanter ces algorithmes sur le processeur HyperStone et d'en évaluer les performances. Une dernière phase a consisté à optimiser les programmes tant au niveau codage qu'au niveau algorithme afin de respecter le débit de 1.5 Mbit/s de l'application modem.

Du fait des performances insuffisantes obtenues avec le DSP E1-32 d'HyperStone et du problème de synchronisation des trames, une répartition des tâches numériques entre un

composant FPGA spécifique et le DSP a été adoptée. Le composant FPGA spécifique a pour but de synchroniser le modem client sur l'arrivée des supertrames DAVIC et d'en extraire les différentes informations avant exploitation. Dans cette nouvelle stratégie le processeur DSP a la double tâche d'effectuer les fonctions de récupération des données utiles et de piloter l'ensemble des blocs fonctionnels de la carte.

Le travail effectué sur la partie analogique a consisté à étudier les fonctions de modulation, de filtrage et de transposition en fréquence dans le respect des spécifications, puis de déterminer un ensemble de composants pour chaque fonction garantissant une bonne compatibilité électrique et structurel avec le DSP E1-32. La solution structurel finale est présentée sur la Figure 27.

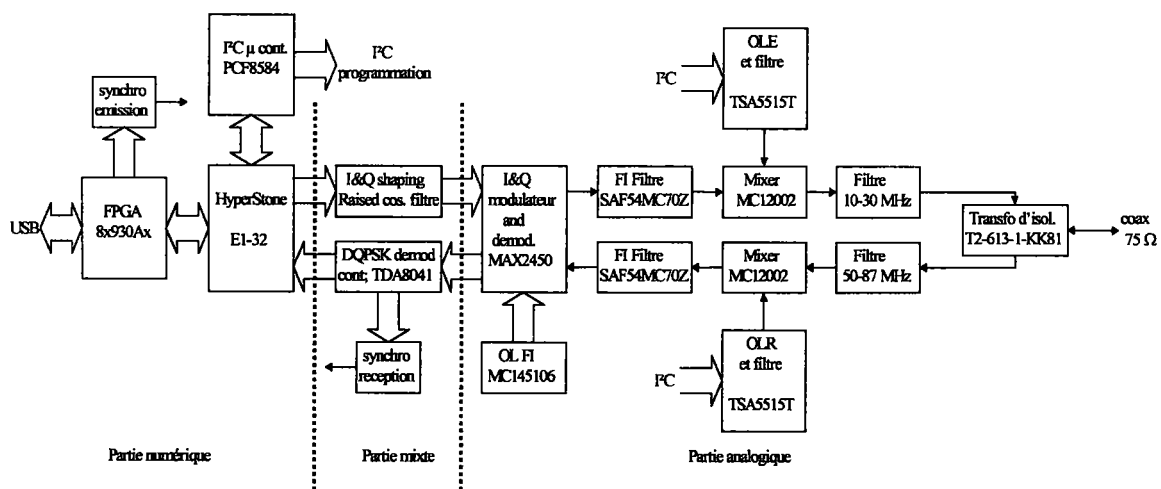


Figure 27 : vue générale du modem

La réalisation de ce prototype prend en compte les contraintes fonctionnelles définies au niveau du client et au niveau du serveur. Une étude séparée est cependant nécessaire du fait de la non-symétrie Client/Serveur, des fonctions de génération des trames et des bandes de fréquences utilisées.

II.5.1.1 Spécifications pour les clients

Le modem côté client doit réaliser les deux opérations suivantes :

- la récupération de la trame provenant du serveur en flux descendant
- l'émission de la trame de retour en flux montant.

Les traitements imposés par les spécifications DAVIC étant différents pour le flux descendant et le flux montant, la complexité du modem est augmentée par ces deux traitements.

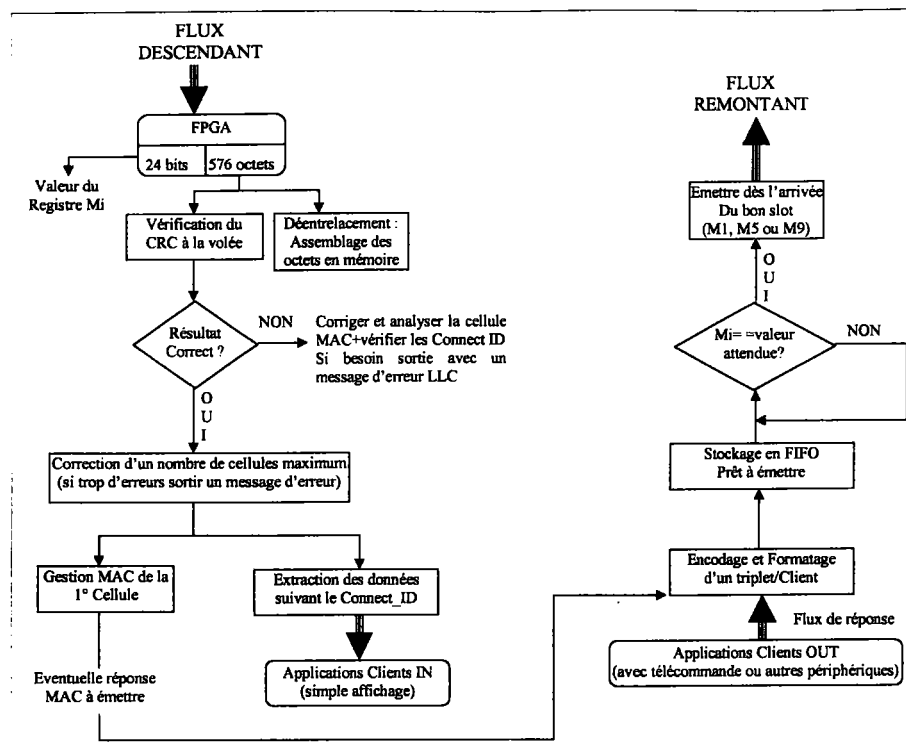


figure 28 : Synoptiques des tâches du côté client

Le slot de remontée constituant la réponse du client au serveur doit être émis lors de la détection d'un bit de position de la trame descendante. Dans ce cas, le client doit préparer son slot de réponse avant l'arrivée de la trame descendante. La réponse du client devra systématiquement être prête à l'avance, et stockée en FIFO.

Pour un même client, les spécifications imposent un délai d'attente de deux trames descendantes (dans le pire des cas) entre deux trames de réponses successives.

En terme de temps, le client est contraint de préparer sa réponse en moins de 6 ms (cela comprend la mise en forme et l'encodage de 9 slots)

En flux descendant un client est contraint en moins de 3 ms :

- de vérifier le CRC, de détecter les cellules erronées (parmi les 10 descendantes), et d'en corriger un nombre limite acceptable.
- de lire dans tous les cas la première cellule qui est la cellule d'information MAC et de la traiter si elle lui est destinée.

II.5.1.2 Les spécifications pour le Serveur

Les contraintes du serveur sont aussi composées de spécifications concernant les voies descendante et montante.

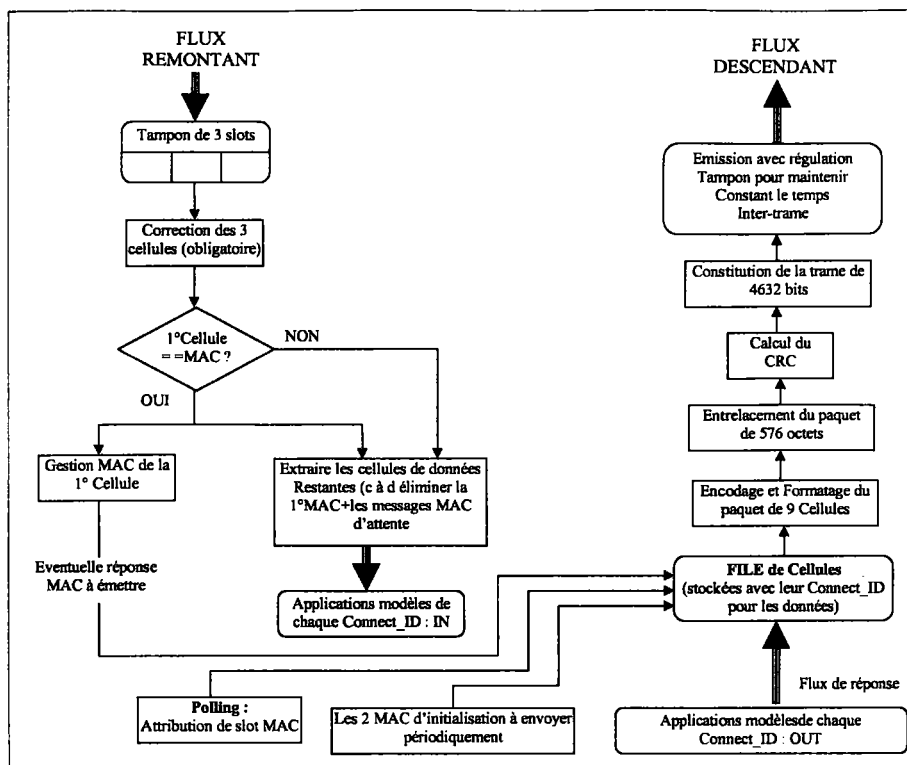


figure 29 : synoptique des tâches coté serveur

Les contraintes du serveur peuvent toutefois être respectées par un suréquipement du modem (un processeur dédié à chaque flux) dans la mesure où un seul serveur est requis dans le réseau. Les tâches affectées au niveau du serveur sont :

- Le codage des 10 cellules descendantes,
- La mise en forme du paquet de 576 octets,
- l'entrelacement,
- le calcul du CRC,
- La mise en forme de la trame complète de 4632 bits avec l'insertion des Mi, Ci et Fi.
- Le décodage des 9 éventuels slots de remontée

Ces tâches doivent être réalisées en moins de 3ms.

De plus, le serveur devra toujours exécuter en parallèle un algorithme de gestion des protocoles d'échange et de signalisation.

II.5.2 PRESENTATION DU PROCESSEUR DSP D'HYPERSTONE

Le processeur Hyperstone (El version 32 bits) représente une nouvelle classe de microprocesseur combinant une structure de processeur RISC à la puissance de traitement des architectures DSP. Il possède 4 Koctets de DRAM en mémoire interne, un cache instruction,

un format d'instructions variable (16, 32, 48 bits) et une exécution en 1 cycle pour la plupart des instructions.

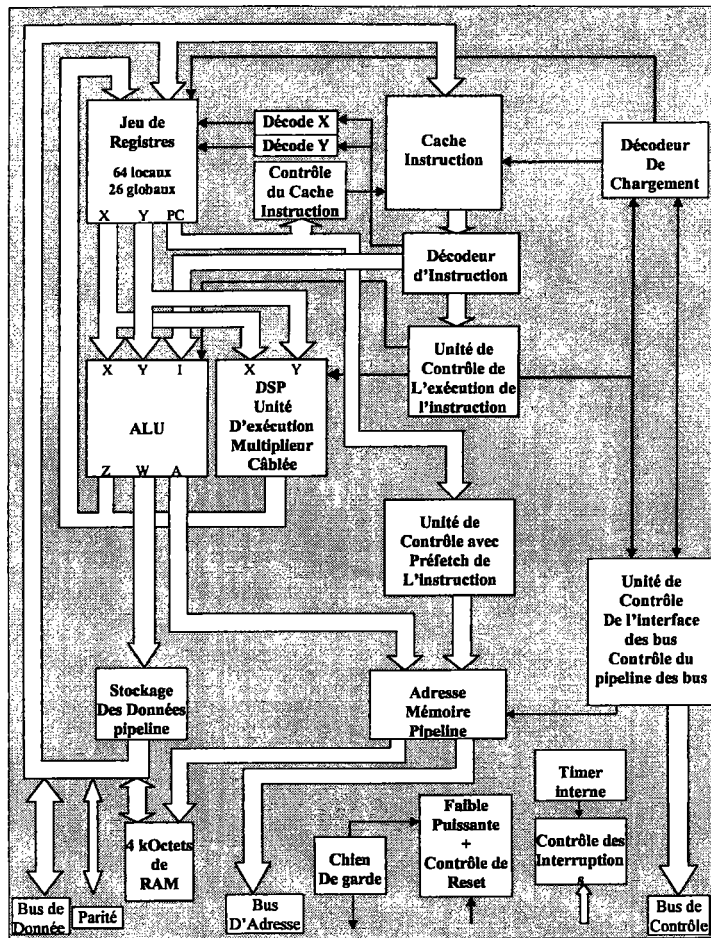


figure 30 : Architecture du DSP d'HyperStone

Ce processeur se singularise par l'adoption de deux parties opératives, l'une comportant une unité arithmétique et logique (ALU) pour traiter les opérations de base, l'autre spécialisée dans le traitement des instructions dites DSP. Afin d'augmenter les performances, le processeur procède à une exécution pipelinée des instructions DSP et permet le traitement d'instructions en parallèle sur les deux parties opératives.

Le diagramme ci dessous (figure 30) représente la structure interne du processeur avec les différents éléments le composant (RAM, DSP ..). Nous voyons également, sur ce diagramme, les différents bus ainsi que les ports intégrés.

II.5.3 LE PROTOCOLE D'ECHANGE CHOISI : LE POLLING

Dans le protocole Aloha, les trames sont sujettes à subir des collisions lors de leur émission. Le serveur doit donc être équipé d'un algorithme complexe de gestion de collisions. Le traitement de cet algorithme combiné à l'exécution des autres tâches du modem par le DSP

E1-32 ne permet plus le respect du débit utile. Aussi, l'option d'un protocole de type Polling a été préférée. Le serveur est maître de tous ses clients mais les temps de parole sont attribués sans conflit d'accès.

Le protocole du Polling oblige le serveur à interroger à tour de rôle chaque client en respectant un ordre préétabli. Cet ordre est défini dans une table repérant les clients connectés.

II.5.4 LA SYNCHRONISATION

La synchronisation consiste, pour le client, à repérer les bits M1, M5 et M9 de la supertrame et à émettre les slots de la voie montante sur ces positions. Dans un premier temps le DSP prenait à sa charge cette détection. Cependant, le DSP choisi ne pouvait traiter cette tâche supplémentaire sans faire chuter le débit utile. Aussi la solution développée dans le modem client consiste à insérer en entrée un composant FPGA programmé prenant à sa charge la synchronisation ainsi que des fonctions moins critiques pour soulager le traitement du DSP.

II.5.5 SPECIFICATIONS DU COMPOSANT FPGA EN FLUX DESCENDANT

Ce circuit est réalisé en FPGA. Son rôle est de récupérer les données provenant du démodulateur Q.P.S.K (Quadrature Phase Shift Keying) avec une rapidité de 0,772 Mbauds, soit un débit de 1,544 Mbits/s et d'effectuer les opérations suivantes :

- Un décodage différentiel
- Un désembrouillage dont le polynôme générateur est de la forme : x^6+x+1
- Générer les impulsions de synchronisation sur les bits M1, M5, M9
- Mémoriser dans un registre les 24 bits de signalisation M1 à M12, C1 à C6, F1 à F6
- Envoyer les données vers le DSP sous forme d'octets (576 octets) avec un signal d'interruption indiquant la réception d'un octet de données.

Les fonctions décrites sont réalisées avec des circuits programmables ALTERA. Les circuits ALTERA de la série EPM 7 sont en technologie C.M.O.S et peuvent supporter des fréquences de fonctionnement de l'ordre de 50 MHz.

II.5.5.1 Interfaçage avec le DSP d'HyperStone

Le DSP E-32 d'HyperStone a un bus de données de 32 bits de largeur. La mémoire est adressée par le DSP en mots de 8 bits. Le DSP possède 4 entrées pour les signaux

d'interruption (INT1 à INT4). Les signaux provenant du circuit FPGA vers les entrées d'interruption sont :

- le signal de synchronisation,
- le signal indiquant la réception d'un octet de données.

Seuls 8 bits sur 32 du bus de données sont utilisés sur le DSP, car le circuit FPGA fournit les données et les bits de signalisation sur 8 bits. La commande des tampons 3 slots ainsi que la sélection des registres sont faites grâce au bus d'adresses qui est sur 20 bits.

II.5.6 SPECIFICATIONS DU COMPOSANT FPGA EN FLUX MONTANT

Les tâches dévolues au FPGA incluent la récupération des données provenant du DSP E32 et le stockage provisoire dans une file d'attente de type FIFO (First In First Out) de 4 octets. Lors de l'envoi par le DSP du signal indiquant la demande d'émission d'un triplet de slots, ce circuit envoie les données de la FIFO:

- vers un embrouilleur dont le polynôme générateur est de la forme : x^6+x^5+1 ,
- puis vers un codeur différentiel.

Le circuit de la voie montante envoie une impulsion d'interruption vers l'HyperStone pour indiquer que la FIFO est vide, et pour commander la porteuse. Le DSP remplit la FIFO grâce à un signal d'écriture.

Les fonctions décrites ont été réalisées sur des circuits programmables ALTERA (série EPM 7).

II.5.6.1 Interfaçage avec le DSP

Seuls 8 bits sur 32 du bus de données du DSP sont utilisés. En effet, les données envoyées vers la FIFO ne font que 8 bits de large. Les signaux d'écriture dans la FIFO et d'envoi d'un triplet de slots sont générés par 2 bits du bus d'adresse qui est sur 20 bits.

II.5.7 CONTRAINTES SUPPORTEES PAR LE DSP EN SOFT

Les contraintes « soft » principales sont relatives aux exigences concernant le temps d'exécution des programmes (voir spécifications précédemment citées) .

Lors de la première phase de travail, la recherche d'algorithme réalisant chacune des fonctions numériques a permis de proposer un programme pour leurs exécutions. L'implantation de ces programmes sur le DSP ont fournit des temps d'exécution largement supérieur à ceux autorisés pour respecter le débit imposé.

Un travail d'optimisation des programmes (réduction du nombre d'instructions) et

d'optimisation algorithmique (réduction du nombre de boucles) a été effectué. Malgré un gain significatif (réduction de moitié du temps d'exécution) les résultats restaient critiques et surtout ne permettaient pas la prise en compte de la complexité liée à la synchronisation et à la gestion du protocole d'échange.

Pour permettre le respect des contraintes de temps, la solution adoptée du côté client a donc consisté à partager les tâches entre le FPGA et le DSP (le DSP prenant en charge les fonctions les plus coûteuses CRC, Reed-Solomon) et à s'écarter des spécifications DAVIC (emploi du protocole Polling au lieu du protocole Aloha).

La solution proposée dispose pour respecter le débit de 1,5 Mbits/s au maximum de 6 ms en flux montant pour mettre en forme les réponses des 9 slots avec codage Reed-solomon et de 3 ms en flux descendant pour vérifier le CRC, de détecter les cellules erronées et récupérer les données utiles. De plus le client traite la cellule d'information MAC (en 1^o position).

Le modem du côté du serveur utilise deux DSPs dédiés chacun à un des flux afin de satisfaire les délais d'exécution (3 ms pour réaliser toutes ses tâches).

Du côté du flux descendant le serveur doit coder les 10 cellules, mettre en forme le paquet 576 octets, effectuer l'entrelacement, calculer le CRC, définir les valeurs des Mi, Ci et Fi avant de pouvoir mettre en forme la trame descendante complète.

Du côté du flux montant le serveur doit récupérer et décoder les 9 slots de réponse.

Parallèlement le serveur doit encore (et toujours) exécuter les algorithmes de gestion du Polling et des trames MAC.

II.5.8 RESULTATS DES SIMULATIONS TEMPORELLES

Les paragraphes suivants fournissent les résultats d'exécution des fonctions logicielles implantées sur le DSP et intégrant les différentes améliorations requises afin de respecter les contraintes temporelles.

II.5.8.1 Le Reed-Solomon

Deux fonctions de codage et de décodage des données par un algorithme Reed-Solomon sont introduites par DAVIC. L'une, imposée en voie descendante est un Reed-Solomon (53, 59, 3), l'autre utilisée en flux montant est un Reed-Solomon (53, 55, 1).

Le tableau suivant présente les temps d'exécutions sur le DSP E1-32 bits :

	Codage du RS (53, 59)	Décodage RS (53, 59)	Codage RS (53, 55)	Décodage RS (53, 55)
Durée en ms	0.145	0.5	0.066	0.217

II.5.8.2 Bilan coté client

Il est à noter que tous les temps sont mesurés ici sur un DSP E1-32 à 100 MHz.

Bilan total du coté client est évalué en effectuant la moyenne sur 10000 exécutions de manière à obtenir des résultats significatifs :

	Encodage (de 9 slots) Flux T	Détection de 10 cellules Flux T'	Calcul et comparaison du CRC	TOTAL
Durée en ms	1.55	0.4	0.405	2.355
% du temps	60.3	15.49	15.59	

On peut tout à fait séparer les deux flux T et T', car sur le principe du Polling les dates sur les réponses seront accordées par bouclage de la table des connectés. Donc à moins d'être seul ou à 2 sur le réseau, le client concerné a tout le temps de préparer sa réponse à l'avance. Ceci d'autant plus que le client a pris connaissance deux trames plus tôt de l'instant où le client devra émettre son slot de réponse.

Le temps de fabrication et d'encodage des 9 slots (au maximum) de réponse, ne doit pas être pris en compte simultanément avec celui de traitement du flux descendant. Le client peut ainsi traiter les 2 flux via un seul DSP, et ce même à 50MHz. De plus, la vérification du CRC se fera à la volée sur le prototype.

II.5.8.3 Bilan du coté du serveur

Dans les mêmes conditions que pour le client les résultats obtenus du côté serveur sont :

	Détection des 9 slots	Mise en forme des 4632 bits	Encodage Des 10 cellules ATM	Temps Total
Durée en ms	1.245	1.23	0.895	3.37
% du temps	30.51	30.08	21.88	

Comme il s'agit du serveur, on peut allouer à chacun des flux un DSP dédié.

II.5.9 LIMITATIONS ET PROBLEMATIQUE

L'application modem ici proposée n'est pas strictement conformes aux spécifications DAVIC dans le cas d'une transmission bidirectionnelle à 1.5 Mbits/s sur câble coaxial. Deux raisons essentielles sont les causes de cette incompatibilité :

- La norme DAVIC n'étant pas encore stabilisée, de nombreuses révisions des spécifications ont eu lieu au cours du temps nous obligeant à effectuer des mises à jour régulières. Le projet Européen ayant fixé un calendrier précis à la réalisation du modem, le choix d'une architecture a dû être figé rapidement.
- La deuxième raison provient de la complexité de la norme. La totalité des protocoles et des fonctions n'ont pu être implanté tout en conservant un débit de 1.5Mbit/s..

Cependant, la réalisation en "soft" des fonctions critiques sur le DSP offre de la souplesse à notre prototype. Les algorithmes utilisés permettent facilement une solution programmée paramétrable. Ainsi sur la base du même programme, le DSP exécute le Reed-Solomon (53,55,1) et le Reed-Solomon (53,59, 3).

Malheureusement la complexité des fonctions imposées par DAVIC ne nous permet pas d'envisager un traitement en "soft" de l'application modem complète. Cette constatation nous oblige à considérer qu'une architecture purement "software" n'est pas complètement adaptée à ce type d'application. La portabilité de cette solution vers d'autres applications à des débits plus élevés est très improbable.

II.6 LA SOLUTION HYBRIDE PROPOSEE

La première approche permet de fixer une solution à l'aide d'un ensemble de composants adaptés mais toute modification dans l'application implique l'intégration d'un nouveau composant ou d'une nouvelle configuration.

La deuxième approche répond partiellement à ce problème en autorisant une programmation des fonctions à l'aide d'un DSP. Dans ce cas, une modification des paramètres peut être compensée dans un programme par une affectation dynamique des variables ou par une nouvelle instruction. Malheureusement la plupart des blocs fonctionnels introduits au niveau des spécifications DAVIC sont réalisés à l'aide de fonctions très coûteuses et complexes lesquelles ne peuvent être traitées dans un délai raisonnable par la simple programmation.

Ainsi la conception d'un modem en respectant DAVIC dans les deux premières approches nous ramène à un constat connu de longue date en micro-électronique au sujet des

performances et des coûts en surface liés au choix d'implantation "hardware" ou "software". Les résultats de ce constat sont rappelés sous forme d'un tableau récapitulatif (voir tableau 10).

De plus l'expérience acquise lors de la réalisation du prototype spécialisé, a montré clairement qu'une architecture modulaire permettant l'adaptation à d'autres types d'application (chaîne ATM ou chaîne MPEG2) et une architecture paramétrable offrant la possibilité de réaliser plusieurs fonctions d'une même famille (selon le cas passage d'une QAM 16 ou 64 ou 256 points) constituent deux atouts indispensables à notre architecture pour rendre notre modem adaptable à plusieurs types d'applications modem.

	Performance en Temps d'exécution	Modularité et Flexibilité	Faculté d'adaptation	Coût en surface
Solution câblée "HARD"	HAUTE	AUCUNE	PAUVRE	FORT
Solution DSP "SOFT"	FAIBLE	BONNE	EXCELLENTE	FAIBLE
DSP dédié "MIXTE"	DE BONNE A HAUTE	MOYENNE A BONNE	BONNE	MOYENNE

tableau 10 : résumé des performances

La troisième approche consiste à développer une solution architecturale en tenant compte des inconvénients et des avantages des deux premières approches. Cette solution est présentée dans la suite sous la forme d'un processeur dédié aux applications modem télécom sur câble TV.

Ce processeur doit combiner une architecture DSP pour conserver de hautes performances face aux fonctions classiques de traitement du signal et une architecture câblée pour l'exécution des fonctions les plus coûteuses en temps de calcul. Cette solution "mixte" offre la souplesse des solutions programmées en la dotant des performances en rapidité d'exécution des solutions câblées.

L'architecture du processeur propose un jeu de modules spécifiques. Ces modules sont utilisés pour implanter une fonctionnalité coûteuse sous forme câblée. Ainsi l'aspect modulaire est obtenu à travers le choix des modules et l'aspect paramétrable est obtenu par des possibilités de configuration des modules exploités lors du passage d'instruction aux modules. Une telle architecture permet de modifier et d'adapter la configuration de la chaîne fonctionnelle aux contraintes d'une nouvelle application.

De plus, la plupart des fonctions critiques doivent être traitées simultanément pour permettre au modem d'atteindre les hauts débits (vers 50Mbit/s) pour les applications de

vidéo à la demande. Ce fait conduit nécessairement à un choix d'architecture parallèle.

II.7 CONCLUSION

Ce chapitre a permis de prendre connaissance des contraintes structurelles et fonctionnelles des applications modem sur câble TV. Les spécifications DAVIC fournissent en détail les caractéristiques des différentes chaînes fonctionnelles associées aux divers modes de transmission (unidirectionnel, bidirectionnel), aux types données (MPEG2, ATM) utilisées, aux débits (de 1.5Mbit/s à 50 Mbit/s) employés. Ces chaînes sont alors composées de fonctions complexes pour le codage des données (Reed-solomon, CRC, HRM) pour le codage du canal (codeur différentiel, entrelacement convolutionnel, embrouilleur) pour la modulation (QAM 16, 64, 256 points, QPSK). Les phénomènes de synchronisation et les trames de signalisation liées au protocole MAC introduisent une complexité supplémentaire.

Deux techniques de conception ont été proposées l'une câblée basée sur un jeu de composants dédiés, l'autre basée sur une réalisation programmée autour d'un DSP. La première approche a montré les avantages de cette solution pour permettre la réalisation des fonctions complexes tout en respectant le débit de l'application. Cependant cette solution introduit un grand nombre de composants pour satisfaire aux contraintes fonctionnelles de DAVIC. Cela donne lieu à un surcoût en surface ce qui fixe les limites de cette solution. L'architecture adoptée par cette approche ne permet pas la portabilité vers d'autres applications sans modification du jeu de composant.

La deuxième approche a montré les avantages de souplesse et de portabilité de cette solution par l'emploi de programmes ou d'instructions paramétrables. Toutefois le traitement en "soft" des différentes fonctions complexes ne permet pas le respect des contraintes de temps liées au débit.

Une troisième approche est proposée dans le chapitre III à partir du constat suivant : si une approche pouvait combiner les avantages des deux premières solutions on obtiendrait une solution hautement performante.

Cette approche consiste à développer une architecture de processeur DSP pour offrir la souplesse des solutions programmées combinées à des modules dédiés aux fonctions critiques câblées. De plus cette solution doit adopter une architecture modulaire paramétrable pour autoriser une portabilité aux différentes applications modem. Une architecture hautement parallèle est nécessaire pour satisfaire les contraintes des applications à haut débit (50 Mbit/s).

Le chapitre suivant présente l'architecture de ce processeur. Les choix des stratégies et des

techniques satisfaisant notre volonté d'une architecture modulaire, paramétrable et parallèle y seront justifiés et détaillés.

CHAPITRE III

ARCHITECTURE DU DSP

III.1 INTRODUCTION

L'expérience acquise lors de la réalisation du prototype avec un DSP du commerce (Hyperstone E1-32) a montré clairement l'intérêt et les avantages d'une architecture modulable et configurable. Elle a en outre permis de mettre en évidence les points délicats de l'application modem câble. Ainsi de nombreux blocs fonctionnels sont réalisés à partir d'algorithmes complexes qui s'avèrent coûteux en temps d'exécution et pour lesquels une solution purement logicielle n'est pas adaptée. Pour les hauts débits, un traitement sous forme câblée des fonctions critiques est nécessaire. Mais ces solutions purement câblées n'offrent pas la souplesse requise au développement des applications modem. En terme de performances et de coût, la meilleure solution consiste en un compromis entre ces deux voies.

De plus, les fonctions constituant l'application modem pouvant être traitées simultanément, une architecture à niveau élevé de parallélisme est hautement souhaitable. Une première partie dans ce chapitre présente les différentes architectures des familles classiques en insistant sur les architectures parallèles.

Les processeurs classiques sont classés en fonction de leur approche architecturale. Certaines stratégies déterminent un comportement du processeur (exécution d'une instruction en un ou plusieurs cycles) et influent sur la structure des éléments à insérer (séquenceur câblé ou micro programmé). D'autres stratégies définissent une organisation (une ou plusieurs unités de calcul, architecture à un, deux ou trois bus). Le choix d'autres aspects tel que le jeu d'instruction, la hiérarchie et les accès mémoire influe aussi sur l'architecture d'un processeur.

Les processeurs DSP sont caractérisés par la haute performance de leurs fonctions de traitement du signal mais leurs architectures peuvent combiner plusieurs stratégies.

La première partie de ce chapitre présente après un rapide rappel des principales stratégies employées dans les processeurs classiques, les choix architecturaux mis en œuvre dans les processeurs DSP et les bénéfices qui en découlent.

La suite du chapitre introduit l'architecture de notre processeur DSP et détaille les choix faits en les justifiant par rapport aux contraintes de l'application modem visée (portabilité, modularité, paramétrabilité).

III.2 LES DIFFERENTES STRATEGIES ARCHITECTURALES

Il est relativement simple de classer un élément de mémoire en fonction de sa capacité

(nombre de Moctets ou Goctets), de ses performances (temps d'accès), de son besoin ou non de cycles de réinitialisation (éléments statiques ou dynamiques); il est beaucoup plus difficile de caractériser un processeur. Plusieurs paramètres doivent être considérés tels que la cadence de traitement des instructions, les tailles des bus d'adresses et de données, l'usage ou non de mémoires partagées, etc... Quelques classifications ont été largement présentées popularisées dans l'étude des architectures des ordinateurs, basées sur la l'observation de quelques critères fondamentaux [21], [20].

Un critère important concerne la complexité du jeu d'instructions du processeur. La famille classique CISC (Complex Instruction Set Computer), essentiellement représentée par Intel et Motorola fournit de nombreuses et très puissantes instructions mais nécessite une partie contrôle très complexe, exigeant entre autre l'emploi de techniques de micro programmation. L'option RISC (Reduced Instruction Set Computer) a un jeu réduit d'instructions et un banc de registres important. Ce jeu est défini en fonction des instructions les plus employées sur les CISCs. Le câblage de ce jeu d'instructions réduit est fortement optimisé pour augmenter les performances. Chaque instruction est exécutée en un seul cycle d'horloge. MIPS, SUN, DEC, IBM sont parmi les plus représentatifs de cette famille.

Un autre critère est le niveau de parallélisme des architectures au niveau du traitement des instructions et/ou des programmes. Il existe deux cas de familles extrêmes les SISD (single Instruction Single Data stream) et les MIMD (Multiple Instructions Multiple Data streams). Une famille incarne un cas intermédiaire les SIMD (Single Instruction Multiple Data streams). Le cas des SISD est habituellement représenté par la machine Von Neumann.

Les ordinateurs hypercubes et à structure neuronale sont les exemples les plus connus des MIMD. Les caractéristiques fondamentales des MIMD sont le contrôle multiple (des processeurs multiples ou des contrôleurs multiples) et des chemins de données multiples (plusieurs registres locaux, éléments de mémoires, plusieurs entrées de données).

Dans la famille des SIMD sont regroupées des architectures très différentes. Ces architectures partagent toutes une seule partie de contrôle. Une architecture classique est représentée par plusieurs parties opératives identiques exécutant la même opération simultanément.

Une approche modifiée, s'approchant de la philosophie MIMD, est représentée par les processeurs Superscalaires et VLIW (Very Long Instruction Word). Dans ces deux cas, la technique consiste à construire une macro-instruction composée d'un petit nombre d'instructions élémentaires. Chaque instruction élémentaire sera traitée par une partie opérative spécifique permettant l'exécution parallèle. Ils diffèrent par la structure de la macro-

instruction. Dans le cas Superscalaire l'assemblage d'instructions élémentaires est sous le contrôle du processeur. Dans le cas du VLIW, l'assemblage de l'instruction doit être effectué par le compilateur ou le programmeur. L'architecture Superscalaire est beaucoup plus complexe que celle du VLIW [22].

Le parallélisme intrinsèque peut être obtenu par des techniques différentes : de multiples chemins de données pour la partie opérative ou le pipeline. Toutes les architectures précédentes font un usage abondant de ces deux techniques.

III.3 UN PROCESSEUR PARTICULIER : LE D S P

Un DSP (Digital Signal Processing) est une architecture de processeur optimisée pour les applications de traitement du signal. Un dispositif de traitement du signal sous forme numérique a pour objectif d'opérer sur un signal donné pour satisfaire certaines exigences techniques (filtrage, modulation, analyse spectrale, transformation de Fourier, ...). Ces opérations de traitement du signal sont généralement répétitives, agissant de façon régulière et systématique sur les données [19]. Ainsi l'implantation dans les composants de ces deux propriétés (régularité et répétitivité) est caractéristique d'une famille de processeurs dédiés au traitement numérique de signal (DSP).

III.3.1 LES ASPECTS ARCHITECTURAUX

Un processeur DSP s'articule autour du même principe de fonctionnement que les processeurs classiques. Une partie opérative effectue les calculs, un séquenceur contrôle l'exécution du programme et une mémoire stocke les données, les résultats et les instructions. Le partitionnement de la mémoire, les techniques de pipeline sont-elles aussi utilisées pour d'autres types de processeurs [20].

La présence des deux primitives nécessaires au traitement de tableaux que sont le générateur d'adresses et le compteur de répétition ainsi que l'architecture de la partie opérative spécifient l'architecture d'un processeur DSP [24].

III.3.1.1 Générateur d'adresses

En traitement du signal, le nombre de fois où l'on utilise une même donnée pour produire un résultat est souvent réduit (traitement dit à faible localité). Par contre, il faut accéder à de nombreuses données pour effectuer des acquisitions de coefficients successifs. Dans ce cas, la charge de calcul des adresses est comparable à la charge de calcul sur les données.

Le caractère régulier des accès (saut de composante à composante alignée selon un même vecteur) à l'intérieur d'un tableau est réalisé par l'opérateur câblé, le générateur d'adresses.

Le générateur d'adresses réalise en un cycle l'incrémentation d'adresses correspondant à l'accès de la composante suivante. Ce bloc est implanté deux fois dans la plupart des DSPs du commerce (une fois pour la mémoire dédiée aux données et une autre fois pour la mémoire dédiée aux coefficients). Le générateur d'adresses utilise la propriété de régularité des traitements.

III.3.1.2 Le compteur de répétition

Le compteur de répétition permet de répéter une séquence d'instructions sans y faire figurer d'instructions de contrôle de boucle. Le compteur réalise sous forme câblée le décompte des itérations et la reprise en début de séquence tant que le nombre requis de rebouclage n'est pas atteint.

La perte de temps dans la gestion d'une partie répétitive se limite alors au cycle d'horloge perdu à l'entrée de la boucle. Ce mécanisme est particulièrement rentable pour les boucles courtes, très fréquentes en traitement du signal. Le compteur de répétition permet d'utiliser la propriété de répétitivité des traitements.

III.3.1.3 Les architectures des parties opératives

Les processeurs DSP peuvent se classer en fonction des classes d'algorithme ciblé. Trois grandes familles au moins tendent à se démarquer par le choix d'une architecture dédiée de leur partie opérative. Ces architectures sont orientées vers le filtrage, le calcul complexe et la transformée de Fourier rapide (FFT) ou vers une architecture orthogonale combinant les deux autres premiers choix.

III.3.1.3.1 Architecture orientée filtrage

La première famille vise les calculs matriciels basés sur les opérations de multiplication/addition élément indispensable dans la plupart des transformations en traitement du signal et notamment dans les algorithmes liés au filtrage. Ces algorithmes que l'on retrouve dans le filtre à réponse impulsionnelle fini réalisant les égalisateurs (vu dans le paragraphe I.2.2.2.2) s'expriment généralement sous la forme suivante:

$$y(n) = \sum_{i(0,k-1)} a(i) * x(n-i)$$

La réalisation de ce type d'opérations nécessite la présence de deux unités de calcul (un multiplieur et un additionneur) et d'un registre accumulateur. La figure 31 donne le principe

de cette architecture.

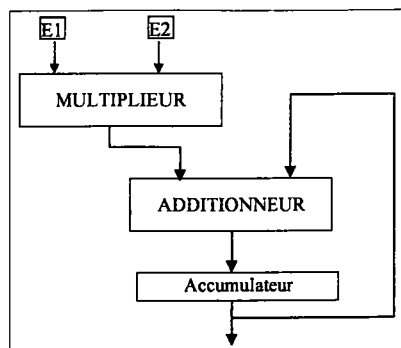


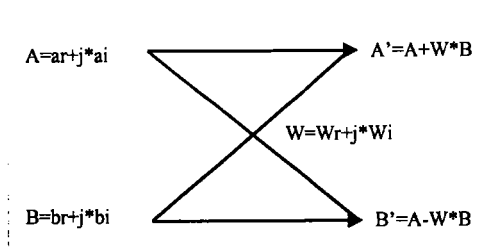
figure 31 : partie opérative orientée filtrage

A quelques variations près concernant le nombre et la place des registres, cette architecture de la partie opérative se retrouve dans les produits suivants :

- Famille TMS320CXX de Texas Instrument,
- DSP 16 et DSP 32 d'AT&T,
- Famille ST18 de SGS Thomson.

III.3.1.3.2 Architecture orientée FFT

La seconde famille vise la transformée de Fourier rapide dans son algorithme radix 2, qui se compose d'une suite de papillons à deux entrées. Le papillon se présente de la manière suivante:



Le papillon fait apparaître une opération supplémentaire, l'addition/multiplication sur deux opérandes identiques. L'architecture de la partie de traitement de ces DSP est alors présentée sur la figure 32.

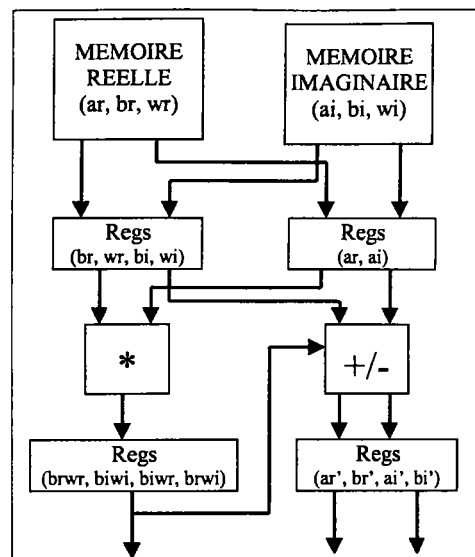


figure 32 : partie opérative orientée FFT

Les processeurs de cette famille sont moins nombreux que ceux de la précédente. Les processeurs MC 56000 et MC 96002 de Motorola sont les plus représentatifs de cette famille.

III.3.1.3.3 Architecture orthogonale

Une architecture est dite orthogonale si les chemins des données permettent de mettre en relation n'importe lequel des registres avec n'importe laquelle des unités de calcul. Ces architectures ont la propriété d'avoir un comportement assez proche de l'optimal sur la plupart des algorithmes. Ce type d'architecture se présente comme le montre la figure 33.

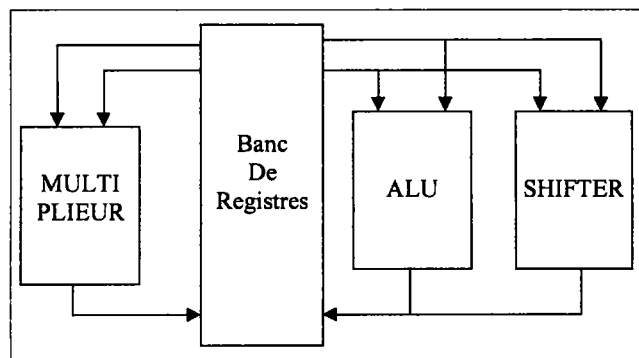


figure 33 : Architecture orthogonale

Cette architecture est représentée par le 21020 d'Analog Device.

III.3.2 LE CŒUR DE DSP

Les DSPs des premières générations étaient coûteux et leur seul but était de fournir une structure adaptée aux algorithmes complexes des fonctions du traitement du signal afin d'en améliorer les performances. Toutefois ces DSPs n'étaient pas adaptés au traitement complet d'une application spécifique. L'évolution technologique aidant, des produits spécialisés dans

la réalisation d'une application ont été introduits. Ces produits utilisent des unités de calcul dédiés au traitement du signal auxquelles s'ajoutent les caractéristiques d'un DSP programmable pour inclure les spécificités de l'application. Aussi une nouvelle famille de processeur est apparue appelée cœur de DSP. Ces cœurs de DSP fournissent une combinaison entre un DSP classique et les capacités d'un micro-contrôleur [24].

III.3.2.1.1 Exemples de cœur de DSP

Deux cœurs de DSP sont présentés l'un traitant les données en virgule flottante l'ADSP-21000 d'Analog Device (voir figure 34), l'autre traitant les données à virgule fixe le TMS320C60 de Texas Instrument (voir figure 35).

Le cœur de DSP ADSP-21000 est utilisé par la famille ADSP-2106X [25] (dont un exemple est le ADSP-21062 de la figure 34) autour duquel est construit un processeur (system on a chip) pour réaliser une application complète. Cette famille de DSP est un exemple du nouveau standard des processeurs de traitement du signal combinant une haute performance des opérations à virgule flottante à l'aide du cœur de DSP et les caractéristiques des systèmes intégrés incluant des interfaces de processeur, un contrôleur DMA et toute la connectivité nécessaire à un traitement multiprocesseur.

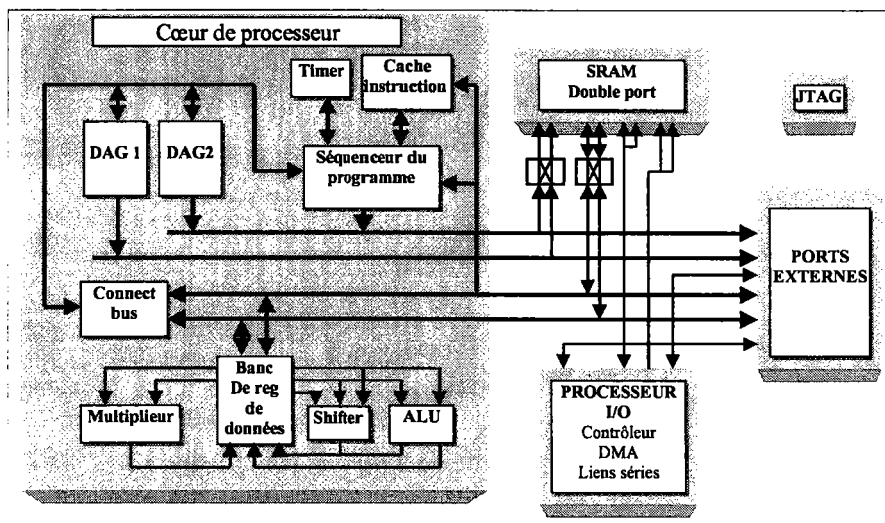


figure 34 : diagramme bloc du ADSP-21062/basé sur le cœur ADSP-21000

Le cœur de DSP de cette famille est composé de trois parties principales, une unité de calcul à architecture orthogonale, un générateur d'adresses (DAG1, DAG2) pilotant un "buffer" circulaire, et un séquenceur de programme coordonnant les différents signaux du processeur.

Le cœur de DSP TMS320C60 de Texas Instrument est utilisé comme base de la famille de processeur TMS320C6X [26]. De la même manière que pour l'exemple précédent les C6X

sont des processeurs élaborés pour réaliser une application complète. Pour cela le cœur de DSP TMS320C60 offre les hautes performances d'un DSP sur des opérations à virgule fixe.

Ces familles de processeurs offrent la structure interne aidant la connexion de plusieurs processeurs. Cette caractéristique permet un traitement en multiprocesseurs pour les applications coûteuses.

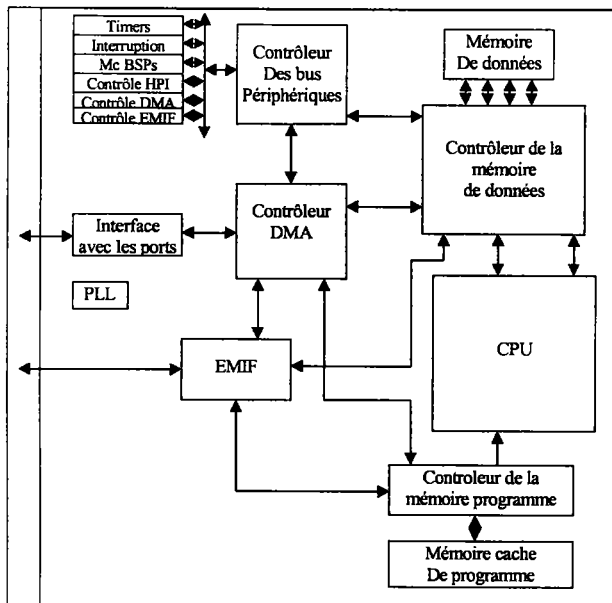


figure 35 : diagramme bloc du TMS320C62

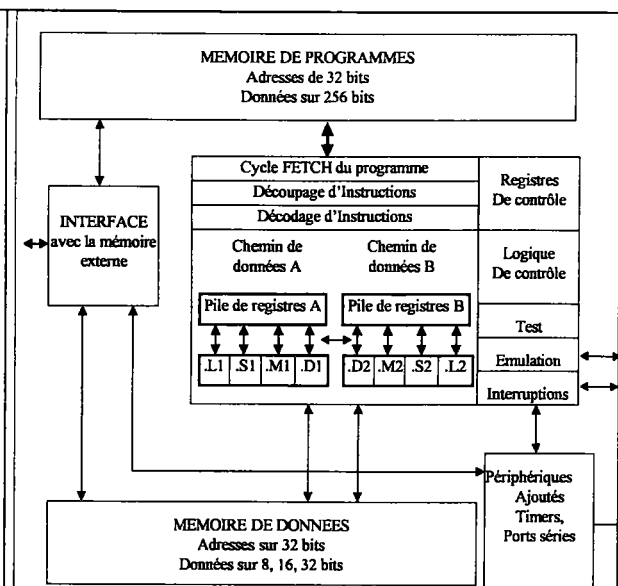


figure 36 : Diagramme bloc du CPU

Ce cœur de DSP se caractérise par une architecture parallèle en adoptant la philosophie VLIW. Cette architecture possède des mots d'instructions de 256 bits autorisant le traitement parallèle de 8 instructions élémentaires de 32 bits. De plus une organisation Harvard est adoptée qui permet d'accéder simultanément à deux blocs distincts de mémoire, la mémoire des programmes et celle des données.

III.4 NOTRE PROCESSEUR DSP

Dans ce paragraphe, le processeur présenté est dédié aux applications modem sur câble TV. Les résultats obtenus dans le chapitre précédent ont montré les avantages d'une architecture combinant une solution "software" et une solution "hardware".

L'architecture proposée s'articule autour d'un cœur de DSP et d'un jeu de composants spécialisés dans le traitement des fonctions complexes de notre application modem. La présence d'un cœur de DSP comme le montre les deux exemples du paragraphe précédent est parfaitement adapté au traitement des opérations classiques avec un haut niveau de performance.

Les composants spécifiques appelés ADM (Auxiliary Dedicated Module) sont des modules

dédiés au traitement d'une fonction trop consommatrice en temps d'exécution disposant d'une partie opérative câblée pour assurer à l'exécution de ces fonctions un niveau de performance proche d'une solution "hardware".

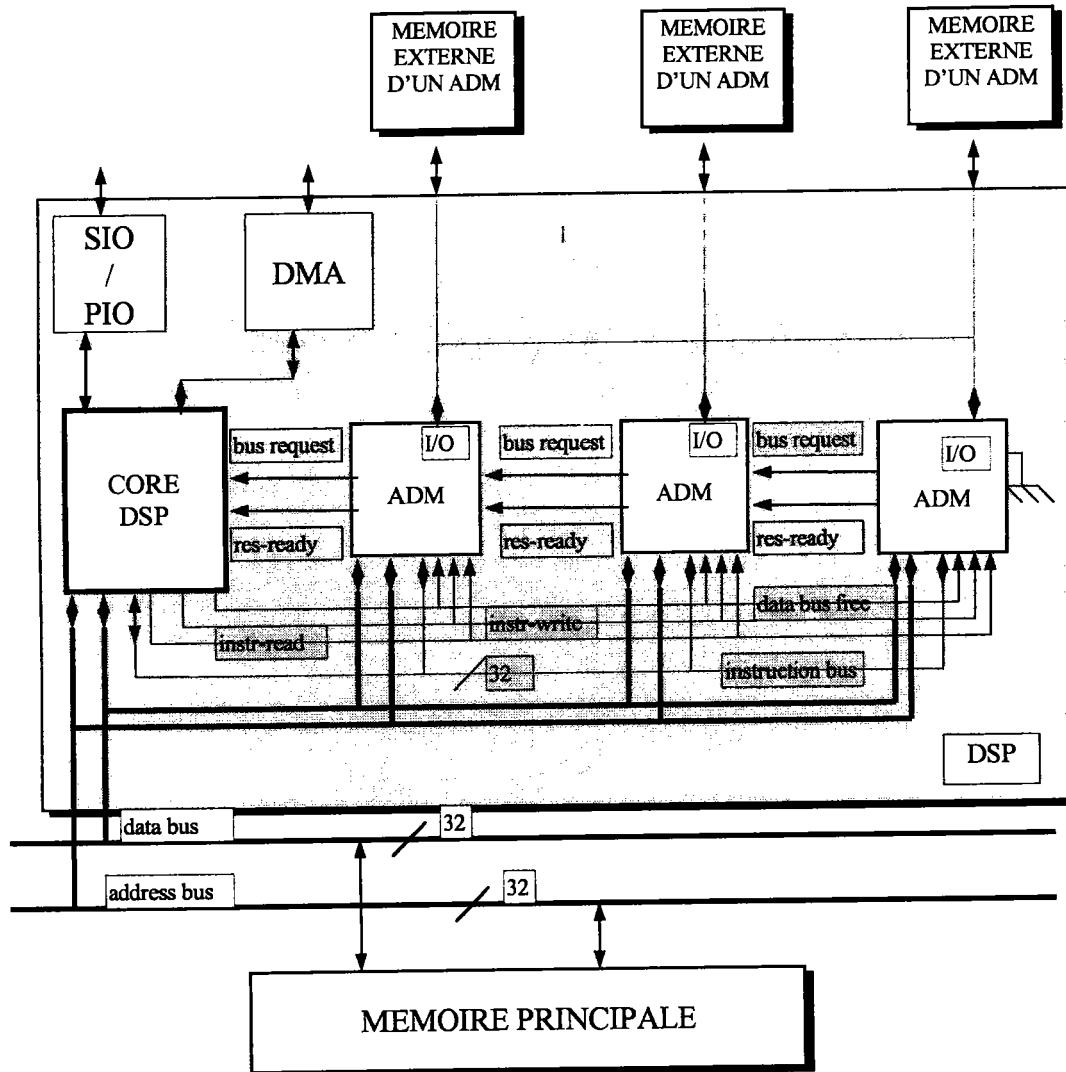


figure 37 : Vue générale de l'architecture du DSP

stratégie architecturale

La figure 37 présente une vue générale de l'architecture de notre processeur. Cette architecture a adopté une organisation à trois bus (*instr_bus*, *data_bus*, *address_bus*) de 32 bits. Ces trois bus sont connectés en mode bidirectionnel sur chacun des ADMs et sur le cœur de DSP.

Principe de fonctionnement

Le fonctionnement du processeur est basé sur le schéma maître/esclave. Le cœur de DSP est le maître du processeur. Seul le cœur de DSP est autorisé à récupérer les instructions en mémoire principale. Tous les codes instructions sont d'abord vérifiés par le contrôleur du cœur de DSP avant exécution. Selon le code de l'instruction, l'exécution est à réaliser soit par

une unité de calcul du cœur de DSP soit par un ADM. Dans le cas où l'instruction serait affecté à un ADM, le cœur de DSP envoie cette instruction à l'ADM ciblé, lequel démarre son exécution indépendamment du cœur. Ainsi pendant l'exécution d'une fonction d'un ADM, le cœur de DSP est libre de traiter de nouvelles instructions, voire d'initialiser un autre ADM. Ce principe est adopté pour favoriser le traitement parallèle des instructions sur les différentes unités de calcul.

III.5 LE CŒUR DE DSP

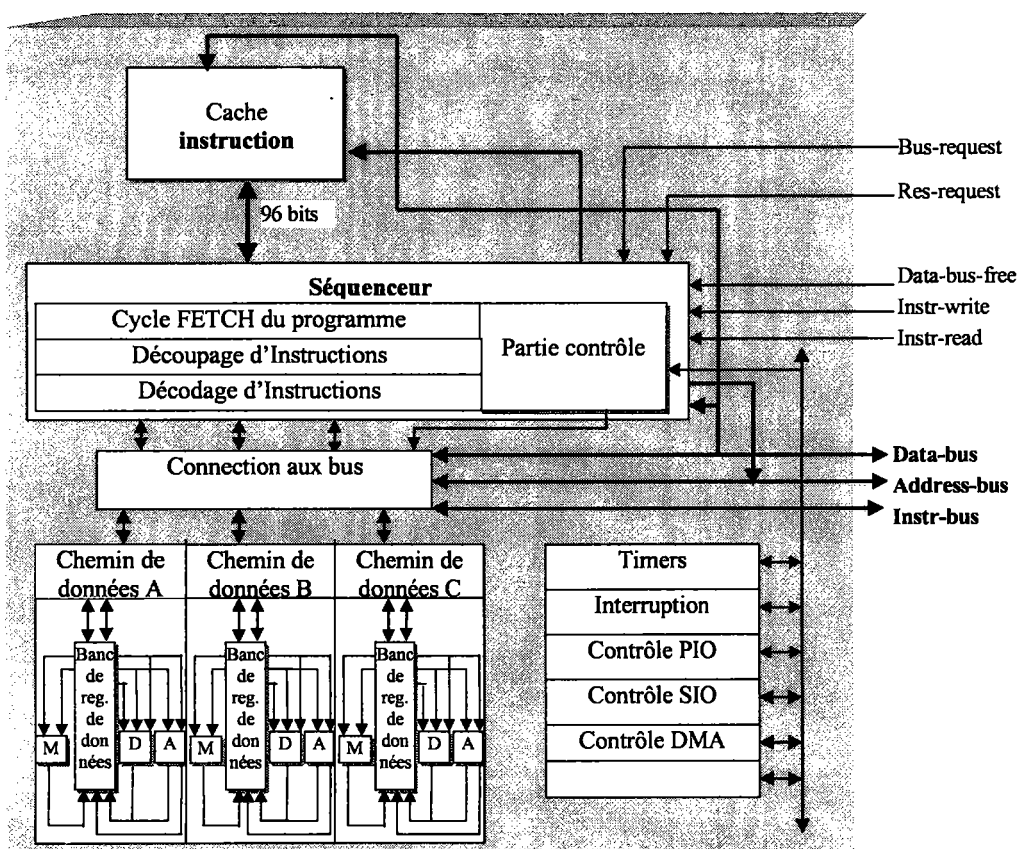


Figure 38 : diagramme bloc du cœur de DSP

Notre cœur de DSP est un cœur classique s'inspirant du cœur du TMS320C60 de Texas Instrument (voir figure 36) et du ADSP-21000 d'Analog Device. L'architecture bloc de notre cœur de DSP présenté sur la Figure 38 combine les avantages des deux cœurs précédemment énoncés. L'architecture de notre cœur de DSP est basée sur la philosophie VLIW [22] qui propose plusieurs niveaux de traitement en parallèle.

III.5.1 UNE ARCHITECTURE VLIW

L'architecture VLIW adoptée par notre cœur de DSP permet de traiter à chaque cycle fetch

un mot d'instruction composé de plusieurs instructions élémentaires. Dans notre cas, un mot d'instruction est constitué de trois instructions élémentaires de 32 bits soit 96 bits (3×32). Une architecture VLIW nécessite au niveau du contrôleur d'adapter les trois fonctions que sont la recherche du mot d'instruction, le découpage de l'instruction et le décodage de l'instruction à la philosophie du VLIW.

La fonction de recherche de l'instruction assimilé au cycle "fetch" doit effectuer l'acquisition du mot d'instruction de 96 bits comme une seule instruction, transmettre cette au traitement de découpage et générer l'adresse mémoire du mot d'instruction de 96 bits suivant.

Pour sa part, la fonction de découpage doit identifier les trois instructions élémentaires de 32 bits incluses dans le mot d'instruction et transmettre en parallèle les trois instructions à la fonction de décodage.

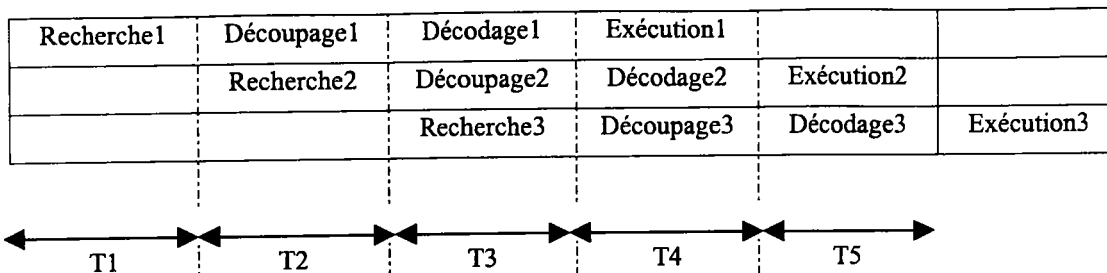
La fonction de décodage détermine en parallèle les trois codes opératoires de chacune des instructions et leurs différents champs avant de les transmettre aux unités de calcul adaptées à leurs exécutions.

Le choix d'une telle architecture fournit les éléments nécessaires au cœur de DSP pour le traitement parallèle de trois instructions élémentaires. L'exécution de 3 instructions arithmétiques, par exemple une addition, une soustraction et une multiplication, est réalisé en parallèle par les trois unités de calcul du cœur de DSP. Cette approche apporte un gain de temps en terme de nombre de cycles. Le bénéfice du VLIW est de permettre le traitement parallèle par duplication des chemins de données.

Toutefois le traitement parallèle n'est pas possible dans le cas d'instructions obligeant le séquenceur à effectuer un saut d'instruction ou un branchement à une adresse ne suivant pas la chronologie du programme. En effet le mot d'instruction de 96 bits est constitué dans ce cas de l'instruction de saut et de deux instructions NOP (pas d'opération). Cependant les fonctions de traitement du signal étant caractérisé par les propriétés de répétitivité et de régularité (voir III.3), cette stratégie d'architecture apporte un fort gain de performance lors du traitement de ces fonctions.

III.5.2 LA TECHNIQUE DE PIPELINE

Pour améliorer encore notre cœur de DSP en terme de performance, la technique de pipeline est introduite lors des phases de recherche, de découpage, de décodage et d'exécution. Comme le montre le tableau ci-dessous la technique de pipeline autorise le traitement de recherche de découpage de décodage d'autres instructions avant la fin d'exécution d'une première instruction.



Cette technique présente sur la plupart des processeurs n'est pas propre à notre architecture. Etant donné notre choix d'architecture parallèle au niveau des unités d'exécution, la combinaison entre la philosophie VLIW et la technique pipeline semble être un choix judicieux en terme de performance.

III.5.3 LE SEQUENCEUR

Dans notre architecture le séquenceur est en grande partie similaire aux séquenceurs utilisés sur les autres processeurs. Son rôle est d'assurer le séquençement des instructions à travers les fonctions de recherche, de découpage, de décodage et d'exécution. Le séquenceur génère dans ce but une adresse programme à la cache instructions qui se charge de récupérer les données en mémoire principale. Le séquenceur lit le mot d'instruction sur 96 bits. Si le séquenceur se contente de transmettre aux unités de calcul leur mot de commande, il supporte aussi certaines fonctions qui lui sont propres :

- la gestion des boucles,
- la gestion des sauts,
- les instructions conditionnelles,
- les interruptions,
- la gestion des sous programmes,
- la gestion des connections aux bus,
- le protocole de communication avec les ADMs.

Le séquenceur doit gérer la répétition d'une ou plusieurs instructions. Le séquenceur est capable d'effectuer des sauts à des adresses. La déclaration de ces sauts se fait par des labels dans le programme. Ces sauts peuvent être conditionnels. La gestion des sous programmes comprend le branchement et le retour automatique.

Le séquenceur possède une partie contrôle dont le rôle est de piloter le protocole de communication avec les ADMs et de gérer les connexions aux bus.

Les signaux *instr_read*, *instr_write*, *data_bus_free*, *bus_request*, *res_request* permettent le pilotage du protocole de communication avec les ADMs lors des tâches de :

- lecture du bus d'instructions (*instr_bus*),

- écriture sur le bus *instr_bus*,
- autorisation d'accéder aux bus *instr_bus*,
- gestion des accès à la mémoire par les ADMs,
- fin d'exécution et récupération du résultat d'un ADM.

La gestion des connexions aux bus consiste à sélectionner le chemin de données vers l'unité de calcul sollicitée pour exécuter l'opération.

III.5.4 LES UNITES DE CALCUL

En accord avec le choix de la stratégie VLIW, les trois instructions élémentaires doivent être traitées en parallèle ce qui nous impose d'implanter trois unités de calcul. Le choix des architectures de chacune des unités de calcul peut être soit toute identique soit proposer une architecture propre à chaque unité de traitement. Le choix d'architecture spécifique à chaque unité de calcul peut constituer une gêne à l'exécution de trois opérations identiques. De plus, une complexité supplémentaire au niveau du séquenceur apparaît lors du pilotage des chemins de données entre le séquenceur et les unités de calcul. Aussi l'option de trois unités de calcul identiques a été préférée. L'objectif de ces unités de calcul est d'être la plus générale possible quel que soit le type d'opération. Le choix de trois architectures orthogonales a été adopté dans ce sens. Ce même type d'architecture se retrouve dans le cœur de DSP ADSP-21000 d'Analog Device.

Cette architecture est constituée d'un banc de registres donnant accès en parallèle soit à un multiplieur soit à une unité arithmétique et logique (ALU) soit à un décaleur.

III.6 LE JEU D'INSTRUCTION

Les instructions de ce processeur se divisent en deux types. Le premier type d'instructions regroupent toutes les instructions traitées par le cœur de DSP. Le format de ces instructions est fixé à 32 bits et ces instructions sont divisées en trois classes.

Le deuxième type représente les instructions traitées par les ADMs. Le format de ce type d'instruction est de 96 bits.

Les deux premiers bits d'en tête d'un mot instruction permettent de distinguer une instruction ADM d'une instruction DSP. En effet lors de l'acquisition d'un mot instruction de 96 bits par le séquenceur, la présence de la combinaison «1,1» sur les deux premiers bits signale au séquenceur la présence d'une instruction ADM. Toute autre combinaison signifie que le mot d'instruction est constitué de trois instructions DSP de 32 bits. Le format des

différentes instructions est présenté sur la figure 39.

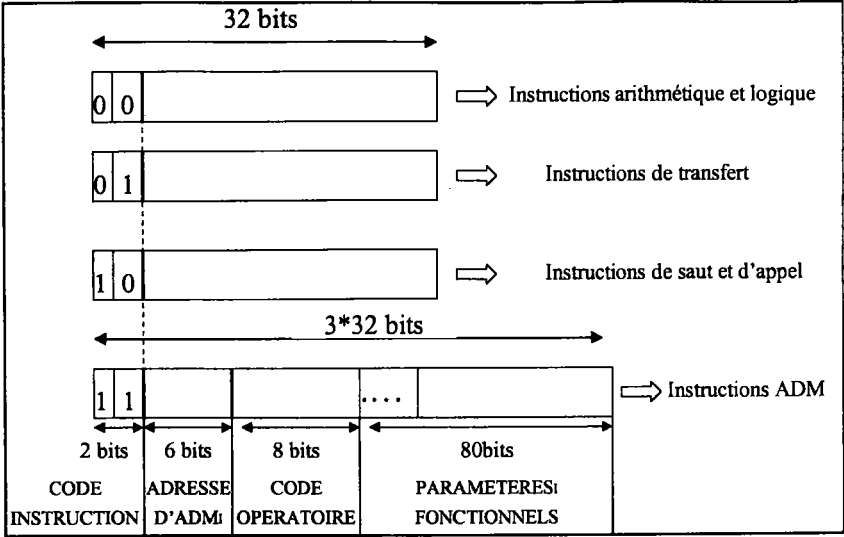


figure 39 : Format des instructions

Le séquenceur doit dès l'acquisition effectuer cette distinction car elle influence le comportement du séquenceur. Dans le cas d'une instruction ADM, le séquenceur doit piloter la connexion aux bus de façon à transmettre en parallèle les 96 bis à l'ADM sur le super bus constitué des trois bus de 32 bits *instr-bus*, *data-bus* et *address-bus*. Une instruction ADM est organisée selon un format à quatre champs. Le premier champ est le code instruction permettant de distinguer le type d'instruction sur deux bits. Le deuxième champ est composé de 6 bits désignant l'ADM considéré à travers son adresse d'ADM. Une adresse ADM différente est attribuée à chaque ADM. Le troisième champ est le code opératoire sur 8 bits. Ce code spécifie à l'ADM concerné la fonction à exécuter. Le dernier champ de 80 bits est utilisé de manière propre à chaque ADM pour fixer tous les paramètres nécessaires à l'exécution de la fonction.

Dans le cas où un mot d'instruction est composé de trois instructions DSP les deux premiers bits de chacune d'elles spécifie la classe de l'instruction. Etant donné que la combinaison "11" est interdite, les trois autres combinaisons de ces deux bits identifient la classe de l'instruction à traiter. La combinaison "00" est utilisée pour repérer une opération arithmétique et logique. Ces instructions peuvent bénéficier de l'architecture VLIW et peuvent être traitées en parallèle par les trois unités de calcul.

Les instructions nécessitant une action de transfert de données (move) sont repérées par la combinaison "01". Ces instructions sont utilisées dans différents types de transfert (d'un registre vers un registre, d'un registre vers la mémoire, de la mémoire vers un registre, d'une valeur immédiate dans la mémoire, d'une valeur immédiate dans un registre, d'une valeur immédiate d'initialisation). Ces instructions peuvent autoriser un traitement parallèle sur les

unités de calcul.

La dernière classe d'instruction caractérisée par la combinaison "10" est utilisée pour les instructions modifiant le compteur ordinal. Ces instructions exécutent des sauts vers une adresse (jump), saut vers un sous programme (call) et retour d'un sous programme (return). Ces instructions ne peuvent pas être traitées en parallèle. Elles sont associées dans le mot d'instruction à deux opérations (NOP).

III.7 LES ADMS

L'ADM (Auxiliary Dedicated Module) est un module dédié au traitement d'une fonction complexe. Ce module permet de soulager le cœur de DSP du traitement trop consommateur en temps de calcul par un jeu d'instruction DSP classique. Une implantation partielle (pour des raisons de coût en surface) sous forme câblée de l'algorithme fonctionnel permet aux ADMs d'obtenir des performances voisines des solutions "hardware". Ces ADMs ne visent pas à réaliser une fonction complexe particulière mais une famille de fonctions complexes. Un ADM dédié au CRC par exemple a pour objectif de réaliser la division polynomiale avec plusieurs polynômes diviseurs et ceci quelle que soit la taille du polynôme divisé. Ainsi l'ADM proposé répond aux contraintes architecturales de souplesse et de paramétrabilité.

Un ADM n'est pas un composant purement câblé mais un module auxiliaire de traitement dont la partie opérative câblée est capable d'exécuter toute une gamme de fonctions utilisant un même algorithme. Son architecture interne (voir prochain chapitre) possède une interface électrique, une interface de contrôle, un séquenceur et une partie opérative câblée. Ce choix architectural combine les avantages d'un système programmable avec les performances d'un système purement câblé.

Dans notre processeur, les ADMs sont les "esclaves" du cœur de DSP. Leur initialisation est effectuée par le cœur de DSP à l'aide d'une instruction de 96 bits. Cette instruction fournit différents paramètres pour sélectionner une fonction particulière (cette fonction peut être modifier par de nouveaux paramètres). Une instruction ADM est transmise par le cœur de DSP aux différents ADM en utilisant l'ensemble des trois bus de 32 bits internes IB (*instr_bus*), AB (*address_bus*) et DB (*data_bus*).

Une instruction ADM possède comme le montre la figure 39 quatre champs. Le premier champ fixe le type d'instruction pour ADM. L'adresse d'ADM est utilisé comme numéro identificateur pour désigné l'ADM concerné par l'instruction. Ce champ est composé de 6 bits. Ainsi notre structure peut distinguer au plus 2^6 (soit 64) ADMs différents. Toutefois une

adresse particulière est utilisée par le cœur de DSP pour communiquer avec tous les ADMs. Cette adresse dite adresse "broadcast" doit être reconnue par tous les ADMs en plus de leur propre adresse. Le cœur de DSP en disposant d'une telle adresse a la possibilité d'interrompre tous les processus en vue d'une nouvelle configuration ou d'un nouveau démarrage de l'instruction.

Le principe de fonctionnement

Le fonctionnement d'un ADM est basé sur un cycle d'instruction pouvant s'étendre sur plusieurs cycles d'horloge. Tous les ADM lisent les six bits d'adresse d'une nouvelle instruction présentés sur le bus IB. Seul l'ADM identifiant sa propre adresse (ou tous si l'adresse "broadcast" est employée) mémorise l'instruction complète de 96 bits à partir des bus IB, AB et DB. Ensuite les 8 bits du code de l'instruction sont traités par un contrôleur de façon à identifier la fonction exacte à traiter. L'organisation des 80 bits de paramètres étant propre à un code opératoire, une fois le code identifié les différents paramètres peuvent être acheminés vers la partie opérative câblée afin d'assurer l'exécution correcte de la fonction désirée. L'instruction reçue peut être une instruction de configuration ou une instruction d'exécution.

Les possibilités de communication d'un ADM

Un ADM possède les structures internes suffisantes pour communiquer avec l'extérieur. L'organisation de notre processeur permet à un ADM d'accéder à la mémoire principale ou à un autre ADM sous la supervision du cœur de DSP en utilisant les bus de données et d'adresse.

Les ADMs ont de plus la possibilité d'accéder à des données sans utiliser les bus AB et DB. Pour soulager les accès aux bus, une mémoire externe est dédiée à chaque ADM. La communication à cette mémoire est effectuée à l'aide des ports E/S. De plus ces ports autorisent la communication entre les ADMs sans monopoliser les bus de données et d'adresses. Cette opération est spécifiée à l'ADM lors du décodage de son instruction.

Dans ce cas, le contrôleur de l'ADM qui traite l'instruction, prend en charge l'émission ou la réception des données par ces ports d'E/S. L'intérêt de ce lien direct entre les ADM est multiple. D'une part il accroît la capacité de communication interne du processeur, d'autre part il dote l'architecture d'un chaînage entre les différents ADM offrant la possibilité d'une exécution pipelinée.

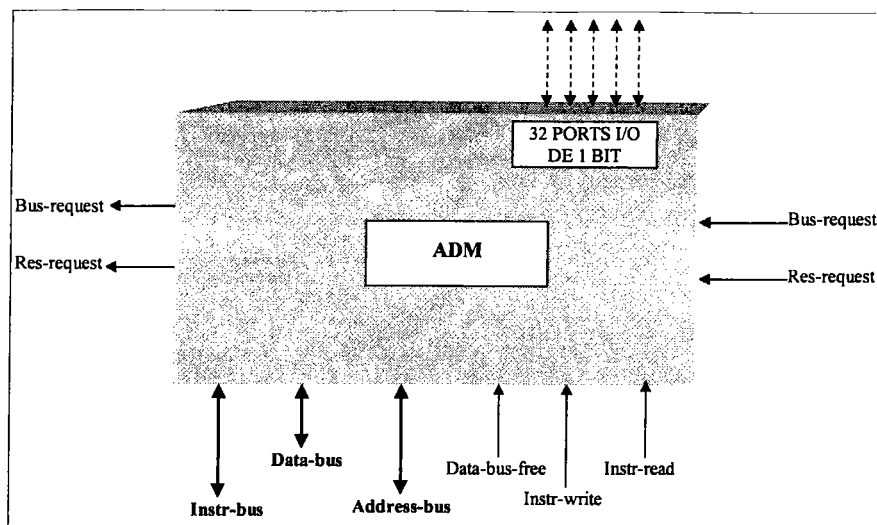


Figure 40 : Vue externe de l'ADM

III.8 LE PROTOCOLE DE COMMUNICATION INTERNE

Le protocole de communication s'attache à établir des règles de communication pour favoriser les transferts de données entre les différents blocs composants le processeur. Les deux composants principaux (cœur de DSP, ADM) de notre processeur sont soumis aux règles définies par ce protocole. Le protocole doit gérer trois types de communications, la communication d'un ADM vers le cœur de DSP, la communication du cœur de DSP vers un ou des ADMs et la communication entre ADMs.

LA COMMUNICATION D'UN ADM VERS LE CŒUR DE DSP

Dans ce sens de communication, l'ADM peut avoir à formuler deux types de requêtes au cœur de DSP, l'une signalant la fin d'exécution, l'autre sollicitant l'accès aux bus de données et d'adresses. Le problème à résoudre par le protocole dans cette communication est d'éviter les conflits d'accès de façon qu'un seul ADM à la fois puisse communiquer. Aussi deux chaînes dites de « priorité » sont mises en œuvre. Ces deux chaînes parcourent l'ensemble des ADMs à l'aide du signal BR (bus request) pour signaler une demande d'accès aux bus et du signal RR (Res request) pour signaler une fin d'exécution. Ces deux chaînes sont dites de priorité car un ADM peut activer une requête si sa position dans la chaîne est plus prioritaire qu'un autre ADM sollicitant une même requête. Dans le cas où l'ADM est seul à communiquer, il active sans problème son signal de requête sur la chaîne appropriée et bloque la chaîne. Le niveau de priorité d'un ADM sur un autre dépend de sa position dans la chaîne considérée.

LA COMMUNICATION DU CŒUR DE DSP VERS UN (OU LES) ADMs

Le cœur de DSP peut vouloir communiquer avec les ADMs dans deux cas :

- pour transmettre à un ADM une instruction,
- en réponse à une requête.

Pour transmettre une instruction à un ADM, le cœur de DSP va dans un premier temps activer le signal *instr_read* pour signaler à tous les ADMs l'arrivée prochaine d'une instruction. Au préalable, les 96 bits d'instruction sont transmis à l'aide des trois bus de 32 bits. Seul l'ADM (ou tous dans le cas d'une adresse broadcast) identifiant son adresse récupère les données.

Dans le cas où une requête d'accès au bus est reçue par le cœur de DSP. Le cœur de DSP active le signal *data_bus_free* pour laisser l'accès au bus. A tout moment il peut reprendre la main en désactivant ce signal.

Dans le cas où un résultat doit être transmis directement au bus, l'ADM devra signaler sa requête à l'aide du signal *res_request*. Le cœur de DSP répondra à cette requête en activant le signal *instr_write* pour autoriser l'ADM à transmettre son résultat via le bus IB.

La communication entre ADMs

Les ADMs ont la possibilité de communiquer entre eux leurs résultats pour respecter la chaîne fonctionnelle de l'application modem et pour améliorer les performances à l'aide d'un pipeline des différentes fonctions complexes sur plusieurs ADMs.

L'architecture du processeur adopte deux moyens de communications entre les ADMs :

- utilisation des bus de données pour transporter la donnée,
- utilisation des ports d'entrée/sortie.

L'accès au bus de données est régenté par le protocole de communication à l'aide d'une requête d'accès au bus au cœur de DSP. Cette procédure est alors bâtie sur la communication d'un ADM vers le cœur de DSP.

L'utilisation des ports E/S est conditionnée par un code opératoire spécifique dont seul l'ADM concerné est capable d'en extraire l'information. Chaque ADM dispose des éléments architecturaux nécessaires à l'accomplissement de ce transfert. La communication entre ADMs s'effectue indépendamment du contrôle par le cœur de DSP.

III.9 CONCLUSION

Ce chapitre a permis de présenter l'architecture d'un processeur dédié aux applications modem pour câble TV. Les différentes approches de conception et notamment l'expérience obtenue par le développement du prototype a précisé les choix architecturaux (modulaire,

paramétrable, parallèle) adaptés aux contraintes de notre application. De plus les fonctions numériques préconisées dans les spécifications des applications modem sont source de problèmes en terme de performances lors de leurs traitements par un DSP classique. En effet les jeux d'instructions des DSPs standards paraissent inadaptés aux traitements de l'application complète.

La réponse architecturale de notre processeur pour lutter contre ce manque de performance est l'emploi d'un cœur de DSP adoptant une philosophie VLIW avec possibilité de trois traitements parallèles. Le traitement des fonctions complexes est réalisé par des modules auxiliaires ADM en adoptant une architecture mixte avec une partie opérative câblée et une partie contrôle pour assurer un le haut niveau de performante associé des systèmes hardware et la souplesse des systèmes software.

L'architecture est modulable dans la mesure où le nombre d'ADMs peut être variable. Ainsi pour s'adapter aux contraintes d'une nouvelle fonctionnalité, l'architecture du processeur peut facilement inclure un nouveau ADM.

De plus, les fonctions implantées sous forme câblée sont exécutées à l'aide d'une instruction fixant les paramètres de la fonction. Cette approche dote l'ADM d'une architecture paramétrable.

Le chapitre suivant va préciser l'architecture interne des modules ADMs.

CHAPITRE IV

L'ARCHITECTURE DES

ADMs

IV.1 INTRODUCTION

Le processeur proposé combine un cœur de DSP et un jeu d'ADMs dédiés aux fonctions complexes. Ce processeur est cadencé par le cœur de DSP. Le cœur de DSP lit les instructions, les décode et les transmet aux unités de calcul adapté à leur traitement. Ce processeur traite deux types d'instructions.

Le premier type d'instructions définit les opérations de bases d'un processeur (opérations arithmétique et logique, opérations d'accès mémoire, opérations de saut d'adresse). Ces opérations sont traitées par le cœur de DSP.

Le deuxième type d'instruction requiert l'exécution d'une fonction complexe nécessaire à notre application modem (Reed-Solomon, CRC, modulation QAM, codeur Viterbi, filtrage). Chacune de ces fonctions est réalisée par un module de traitement spécifique (ADM). Ces modules dédiés chacun à une fonction complexe implantent dans leur unité de calcul une partie d'algorithme sous forme câblée. Ce choix d'architecture est fait pour augmenter les performances en terme de temps de calcul.

Par souci d'accroître les performances de notre processeur, les ADMs doivent pouvoir exécuter leur instruction sans monopoliser le cœur de DSP. Dans ce sens les ADMs prennent à leur charge la totalité de l'exécution et se dotent de moyens internes (contrôleur, séquenceur, interface de contrôle, générateur d'adresse) pour satisfaire les besoins de la fonction (décodage de l'instruction, accès aux bus bidirectionnel, recherche des données, acquisitions des données, stockage des résultats). Le cœur de DSP est alors sollicité seulement dans deux cas : requête pour accéder aux bus externes et une autre pour signaler la fin d'exécution.

Cette organisation a été adoptée pour soulager le cœur de DSP et augmenter le degré de traitement parallèle de notre processeur. Ainsi pendant le traitement d'un ADM, le cœur de DSP a la possibilité de traiter d'autres instructions voir d'initialiser un autre ADM.

Pour permettre à un ADM de posséder tous les éléments à son initialisation, le choix d'un mot d'instruction de 96 bits a été adopté. Ce grand format d'instruction est nécessaire pour fixer les différents paramètres de configuration.

L'ADM est un module complexe sachant traiter une instruction, prendre en compte les spécifications du protocole de communication avec le cœur de DSP. Ce chapitre s'attache à détailler l'architecture interne de ce module.



IV.2 VUE GENERALE

L'ADM est constitué de trois blocs principaux comme le montre la figure 41 :

- un cœur d'ADM spécifique à la réalisation d'une fonction complexe,
- deux interfaces de communications communes à tous les ADMs (l'interface électrique et l'interface de contrôle).

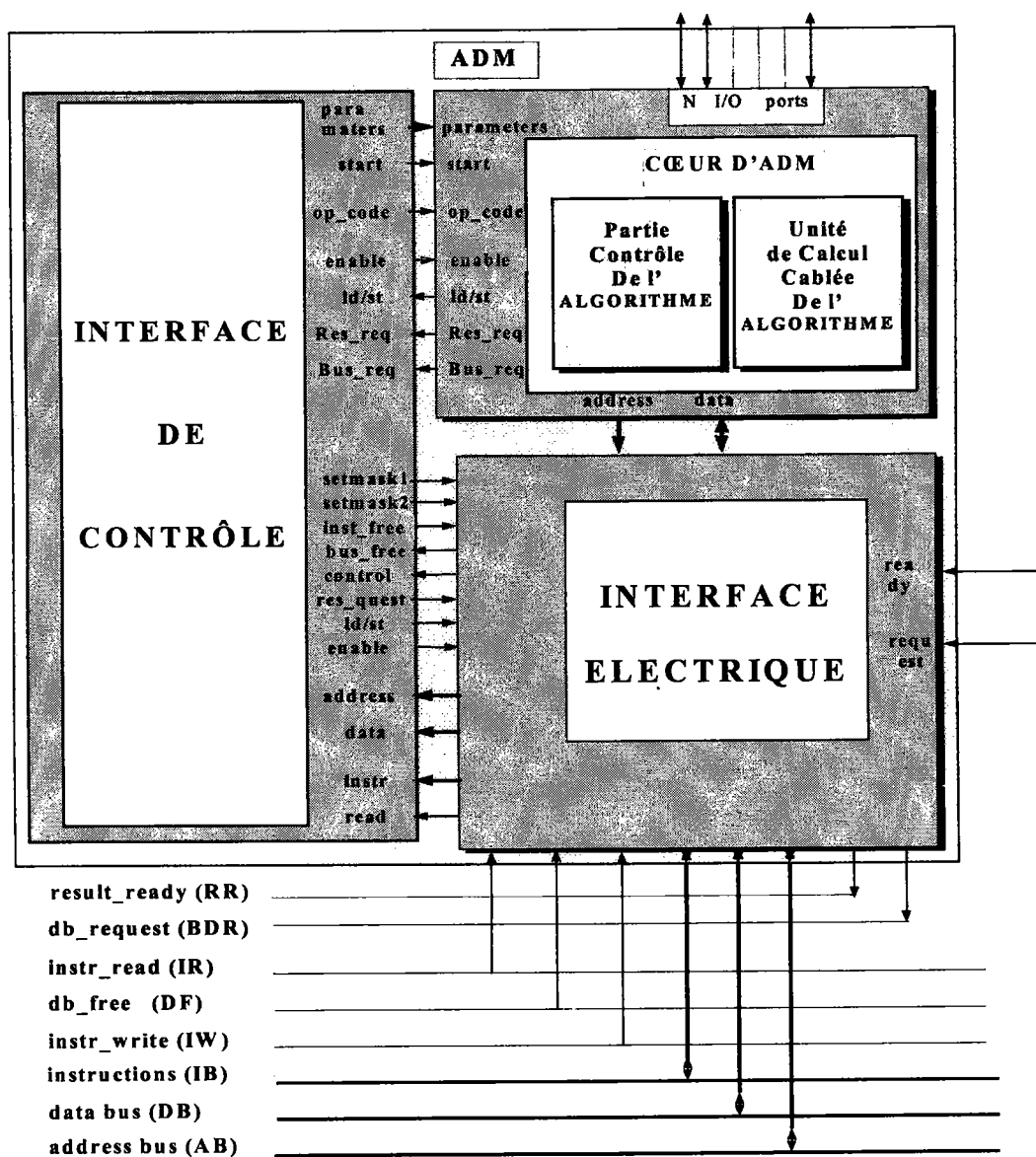


figure 41 : La structure d'un ADM

Le cœur d'ADM implante une ou plusieurs fonctionnalités, en utilisant une partie contrôle (permettant le décodage du code opératoire de l'instruction à réaliser et du passage des paramètres de la fonction choisie offrant la flexibilité) et une partie combinatoire câblée nécessaire pour traiter les fonctionnalités complexes sans dépassement des temps limites.

L'interface électrique est composée d'une logique combinatoire ajoutée servant d'interface

entre le cœur de l'ADM et les signaux de contrôle des trois bus de 32 bits (instruction_bus (IB), data_bus (DB) et le address_bus (AB)).

L'interface de contrôle génère tous les signaux nécessaires au bon fonctionnement de l'ADM ainsi que ceux nécessaires au pilotage l'interface électrique

IV.3 L'INTERFACE ELECTRIQUE

La structure de l'interface électrique est présentée sur la figure 42. Cette structure est définie pour adapter électriquement les signaux bus externes à l'ADM aux signaux et aux bus internes à l'ADM. L'architecture de l'interface consiste à définir des chemins logiques entre les signaux et bus internes de l'ADM (provenant de l'interface de contrôle et du cœur d'ADM) et les signaux et bus externes à l'ADM. Ces chemins sont composés d'éléments logiques tels que portes, inverseurs, suiveurs bloquants, bascules pour respecter le protocole de communication et de gérer les conflits d'accès aux bus. De ce fait l'interface électrique remplit essentiellement deux fonctions : la gestion des deux chaînes de priorité et le contrôle des accès aux bus *DB* et *AB*).

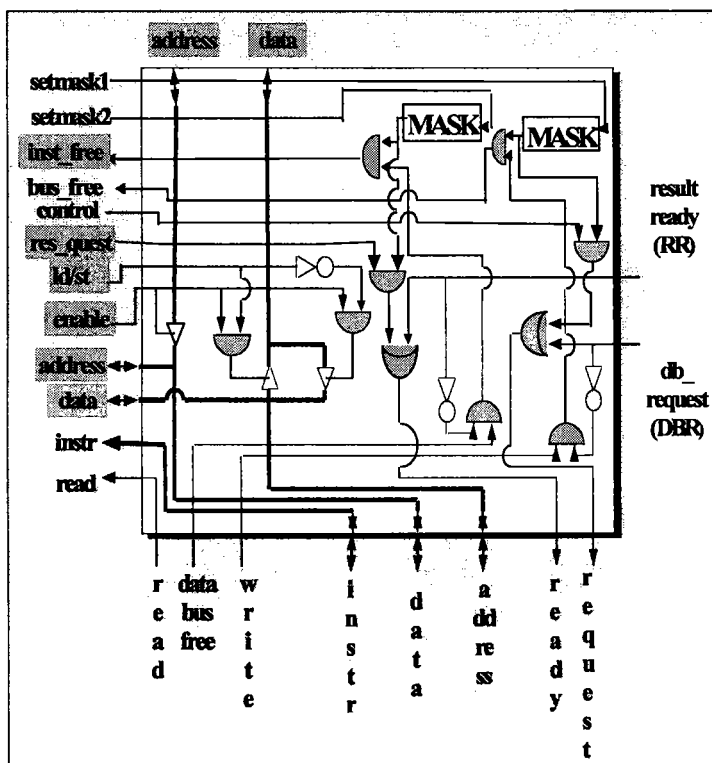


figure 42 : Structure de l'interface électrique

IV.3.1 LA GESTION DES CHAINES DE PRIORITE

Les deux chaînes de priorité sont fournies par le protocole de communication interne de

notre processeur. Ces chaînes ne requièrent pas de la part des ADMs un traitement particulier. Le but de ces deux chaînes est de communiquer au cœur de DSP une requête pour accéder aux bus de données et d'adresses (signal *request* actif) ou une requête pour signaler la fin d'exécution (signal *ready* actif). Aussi un signal pour chaque chaîne parcourt l'ensemble des ADMs. Les dispositifs mis en œuvre pour gérer ces deux chaînes par l'interface électrique des ADMs sont identiques.

Principe de fonctionnement

Le fonctionnement proposé sur la Figure 43 présente le principe de la chaîne de priorité dont le but est de signaler au cœur de DSP une requête de fin d'exécution. Le fonctionnement de la deuxième chaîne de priorité s'appuie sur le même principe.

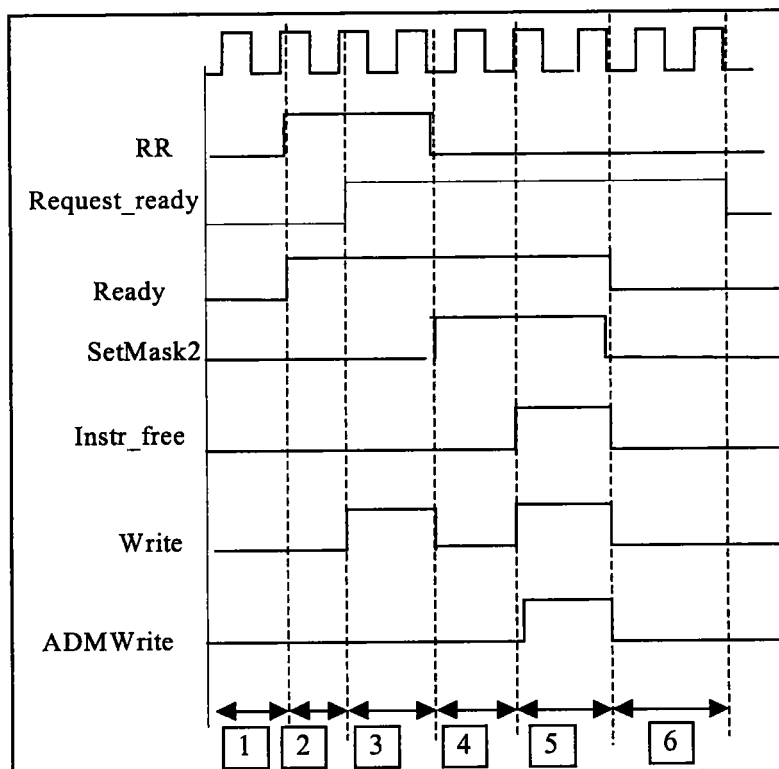


Figure 43 : Simulation du fonctionnement de la chaîne signalant la fin d'exécution.

Le principe de fonctionnement de la chaîne de priorité est présenté à travers les 6 étapes suivantes :

Étape 1 : Cette étape constitue une étape initiale où tous les signaux sont à '0'.

Étape 2 : Lors de cette étape, une requête est formulée par un ADM plus prioritaire (signal *RR* à '1'). Dans ce cas, l'ADM considéré propage cette information le long de la chaîne (*Ready* = '1').

Étape 3 : L'étape est caractérisée par deux requêtes, l'une provenant d'un ADM plus prioritaire (signal *RR* à '1'), l'autre par l'ADM considéré (signal *Request_ready* à '1'). Dans ce cas le signal *Ready* reste actif, le signal *Write* est passé à '1' autorisant un ADM à

communiquer au cœur de DSP sa fin d'exécution. Cependant l'ADM considéré n'étant pas le plus prioritaire le signal *instr_free* n'est pas actif et donc l'ADM n'active pas son signal *ADMwrite*.

Etape 4 : Lors de cette étape, le signal *RR* est repassé à '0' indiquant à l'ADM considéré qu'il est devenu le plus prioritaire dans la chaîne. Son signal *Setmask2* a configuré son registre de masquage à '1' l'autorisant à communiquer sa fin d'exécution dès que le cœur de DSP lui affectera un temps de parole à travers le signal *Write*.

Etape 5 : Le signal *Write* est passé à '1', la valeur du masque est toujours à '1', le signal *instr_free* est donc devenu actif autorisant la communication avec le cœur de DSP avec le signal *ADMwrite*.

Etape 6 : Cette étape représente un cas particulier du fonctionnement de la chaîne de priorité. Malgré sa position du plus prioritaire dans la chaîne, le cœur de DSP a configuré son masque à '0' (*setmask2* à '0') la requête est alors non propagée dans la chaîne (signal *Ready* à '0'). Cette configuration particulière permet au cœur de DSP d'octroyer un temps de parole à un ADM moins prioritaire dans la chaîne.

Dispositifs associés aux chaînes de priorité

Le dispositif mis en place consiste pour chaque chaîne à transmettre à l'ADM suivant, à l'aide du signal *request / ready* si une requête est formulé par un ADMs du début de chaîne *DBR*='1' resp. *RR*='1' ou par l'ADM considéré (signal *request*='1' / contrôle='1'). Dans ce cas une porte "OU" est utilisé dans les deux chaînes comme le montre la figure suivante.

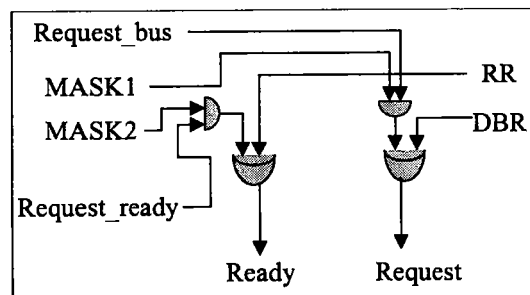


figure 44 : Première partie du dispositif des chaînes de priorité

Un autre dispositif plus complexe est nécessaire dans le cas d'une autorisation par le cœur de DSP d'accéder aux bus (signal *data_bus_free* actif) ou pour écrire le résultat (signal *write* actif). Ce dispositif est présenté sur la figure 45.

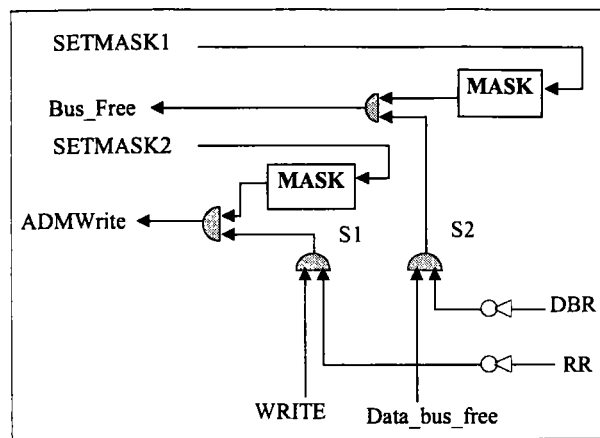


figure 45 : Deuxième partie du dispositif des chaînes de priorité

Dans le cas où les signaux WRITE ou Data_bus_free sont actifs, la première préoccupation est de s'assurer qu'aucun autre ADM plus prioritaire a formulé une requête.

Dans ce cas les signaux DBR ou RR sont mis à '1'. Le dispositif utilise alors une porte "ET" et un inverseur pour interdire aux signaux S1 ou S2 de prendre la valeur '1' signalant la réponse du cœur de DSP.

Dans le cas où les signaux provenant des ADMs plus prioritaire sont actifs, le dispositif permet de proposer à un ADM moins prioritaire de formuler une requête. Cette procédure particulière est imposée par le cœur de DSP pour modifier dynamiquement un niveau de priorité. Une instruction de configuration est alors utilisée pour masquer les requêtes des ADMs plus prioritaires. Ce masquage est réalisé à l'aide des signaux *setmask1* et *setmask2* dans une bascule "MASK". Si la valeur de la bascule "MASK" est à '0', toute requête est non propagée sur la chaîne de priorité. Ainsi le cœur de DSP peut communiquer avec un ADM quel que soit son niveau de priorité. Si la valeur de la bascule "MASK" est à '1', l'ADM le plus prioritaire formulant une requête sera mis en communication avec le cœur de DSP. Le signal *bus_free* (respectivement *ADM_write*) est mis à '1' si la valeur des masques et du signal S1 (respectivement S2) sont à '1'.

IV.3.2 LE CONTROLE DES ACCES AUX BUS

La deuxième fonctionnalité de l'interface électrique est d'effectuer le contrôle des accès aux bus. Les bus AB et DB ont des transfert bidirectionnels à l'intérieur de l'ADM. En effet, les 96 bits d'instruction sont véhiculés à l'ADM à l'aide des trois bus de 32 bits IB, AB, DB. Dans ce cas, les trois bus sont dirigés dans le sens entrant de l'ADM. Dans le cas où le cœur d'ADM génère une adresse pour stocker le résultat en mémoire, les bus AB et DB sont utilisés dans le sens sortant. Ainsi un dispositif commandant un sens de transmission sans

créer des conflits est nécessaire pour ces deux bus.

Principe de fonctionnement

Un ADM peut au cours de son fonctionnement avoir besoin d'accéder à la mémoire principale. Les accès aux bus internes du processeur sont soumis à des règles par le protocole de communication. La simulation de la figure 46 présente les différentes étapes permettant d'accéder aux bus par un ADM.

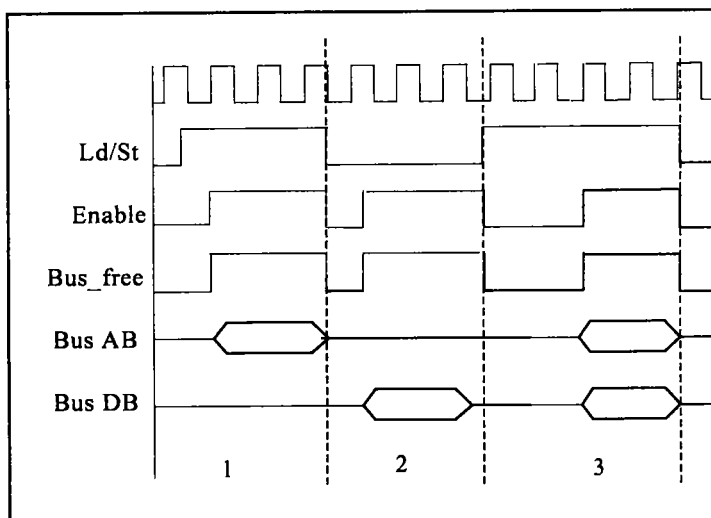


figure 46 : Simulation des accès aux bus par un ADM

Il existe trois actions pendant lesquelles un ADM peut solliciter un accès au bus de données et d'adresses.

Action 1 : Cette action correspond à la recherche d'une nouvelle donnée en mémoire principale. L'ADM signale au préalable une requête au cœur de DSP à l'aide de la chaîne de priorité *bus_request*. L'accès est alors autorisé par le signal *bus_free* à '1'. Cette information est transmise à l'interface de contrôle qui active le signal *enable* au niveau de l'interface électrique. Le signal *Ld/St* à '1' spécifie le sens sortant de la communication. L'ADM peut alors transmettre la valeur de l'adresse mémoire sur le bus AB.

Action 2 : Cette deuxième action correspond à la récupération de la nouvelle données par l'ADM. La procédure est identique à l'action précédente. Toutefois l'ADM doit indiquer à l'interface électrique le sens rentrant de la communication. Le signal *Ld/St* est alors mis à '0'. L'ADM récupère la donnée sur le bus DB.

Action 3 : Cette dernière action est utilisée en fin de traitement d'une fonction pour retourner le résultat en mémoire. Dans ce cas, l'action 1 est répéter à l'identique avec transmission d'une adresse mémoire sur le bus AB et d'une donnée sur le bus DB.

Dispositifs mis en œuvre pour gérer les accès aux bus

Le dispositif utilise des portes 3 états pour définir un sens de communication sur les bus

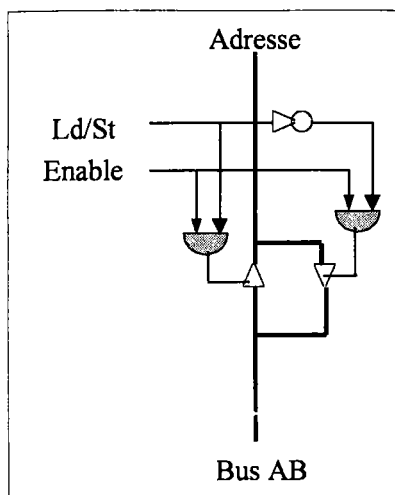


figure 47 : Exemple du dispositif pour le contrôle d'accès du bus d'adresse

. Le pilotage des portes 3 états est réalisé à l'aide de deux signaux (*Ld/St* et *Enable*). Le signal *Ld/St* fixe le sens de communication (*Ld/St* = '1' sens sortant *Ld/St* = '0' sens entrant). Le signal *Enable* est généré par l'interface de contrôle. Sa valeur est mis à '1' dans le cas où le cœur de DSP autorise l'accès au bus (*bus_free* = '1'). Une logique combinatoire permet de piloter la commande des suiveurs bloqueurs à l'aide de portes "ET" et d'un inverseur.

IV.4 L'INTERFACE DE CONTROLE

L'interface de contrôle a pour but de générer tous les signaux indispensables à la lecture d'une nouvelle instruction ainsi que ceux nécessaires à la gestion de l'interface électrique. Cette entité se charge aussi de transmettre les différents champs de l'instruction aux entités concernées (cœur d'ADM ou au contrôleur de protocole). Une vue de l'architecture de cette interface est présenté sur la figure 48.

L'ADM doit traiter deux types d'instructions. Le premier type d'instruction a pour but de fournir les éléments relatifs au contrôle du protocole de communication (exemple : interdire ou autoriser les privilèges d'accès à la mémoire). Le deuxième type d'instruction contient les paramètres et le code opératoire dédiés à une exécution fonctionnelle (exemple : calcul d'un CRC).

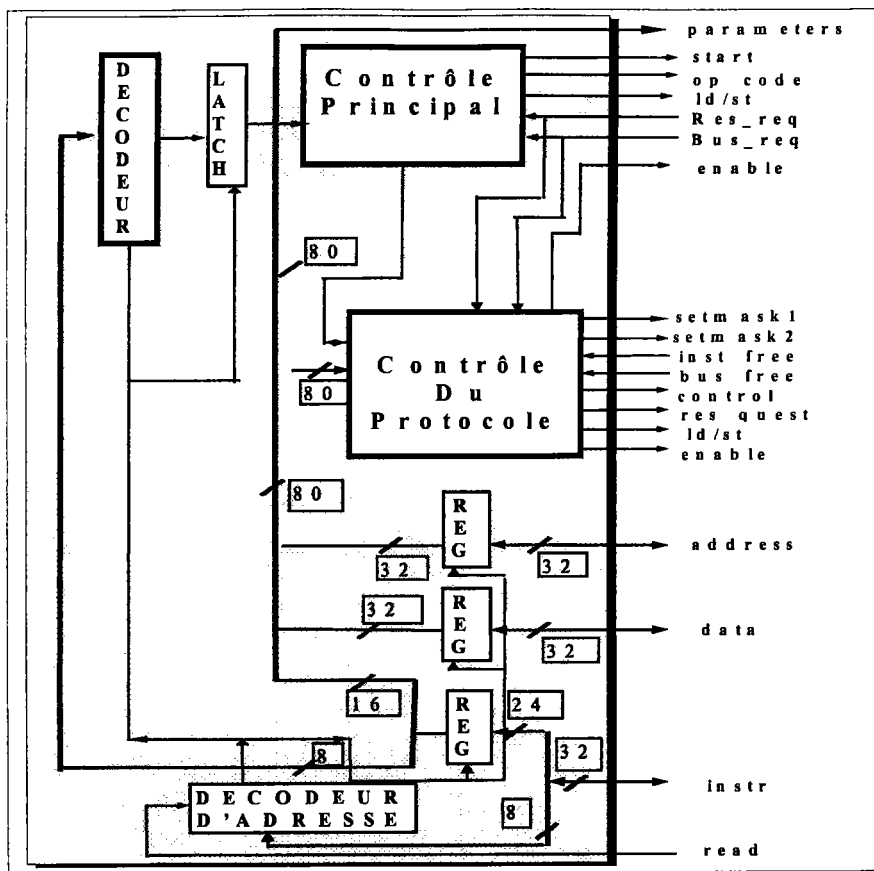


figure 48 : architecture de l'interface de contrôle

Principe du traitement d'une instruction ADM

La présence d’une instruction ADM sur les trois bus internes de 32 bits (IB ,AB, DB) du processeur a été au préalable annoncé par la mis à ‘1’ du signal *read* à tous les ADMs comme le montre la phase 1 de la Figure 49. Dans ce cas, les 8 premiers bits du bus d’instruction sont capturés sur niveau haut du signal *read* par le décodeur d’adresse et les 88 bits restants de l’instruction sont stockés dans trois registres (deux registres de 32 bits et un registre de 24 bits).

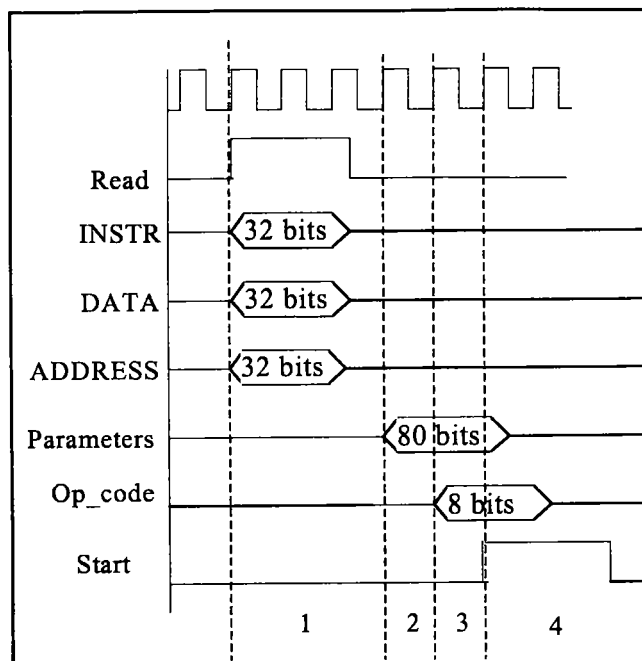


Figure 49 : Chronologie du traitement d'une instruction ADM

Si l'adresse transportée correspond soit à l'adresse identifiant cet ADM soit à l'adresse "broadcast", le décodeur génère un signal autorisant le transfert de l'information contenu dans les trois registres vers d'autres structures internes de cette interface. Les 8 bits du code opératoire sont acheminés vers le décodeur afin de transmettre au contrôle principal le code de la fonction à exécuter (*Op_code*). Le contrôle principal génère les signaux associés à ce code opératoire et les envoie au cœur d'ADM.

Dans le cas d'une instruction pour le cœur d'ADM (exécution d'une fonction complexe), les 80 bits de paramètres sont acheminés directement des registres vers le cœur d'ADM ce qui correspond à la phase 2 de la figure 49. Le contrôle principal transmet le code (*Op_code*) de la fonction à traiter (phase 3 de la figure 49). Puis le signal *start* est activé pour signifier le début de l'exécution par le cœur d'ADM (phase 4 de la figure 49).

Le code opératoire et les 80 bits de paramètres sont dirigés vers le contrôle de protocole dans le cas d'une instruction liée au protocole de communication (configuration des registres de masquage par exemple). Dans la suite de ce paragraphe, une présentation des différentes structures internes est proposée.

IV.4.1 LE DECODEUR D'ADRESSE

Le décodeur d'adresse est utilisé pour identifier l'adresse de l'instruction. Son but est de vérifier si l'adresse arrivante lui est destinée. Pour cela le décodeur compare l'adresse arrivante (les 8 bits premier bits de l'instruction) avec les 8 bits de l'adresse de cet ADM et

avec les 8 bits de l'adresse "broadcast". Ce décodeur d'adresse est constitué d'une fonction logique combinatoire dont le signal de sortie est à '1' lorsque les 8 bits d'adresse transmis sont identique aux 8 bits d'adresse de l'ADM préchargé ou aux 8 bits de l'adresse "broadcast".

IV.4.2 LE DECODEUR

Le décodeur est le dispositif utilisé pour le décodage du code opératoire. Il récupère en entrée les 8 bits du champ d'instruction destinés au code opératoire. Le décodeur d'adresse active son signal de début de traitement. Son traitement consiste à définir les différentes commandes actives associées au code opératoire et à les transmettre au contrôleur principale. Cette structure est réalisé en logique combinatoire nécessite une organisation particulière selon la fonction implantée. Cette structure est combinée en sortie à un "latch" de manière à maintenir les commandes actives pendant toute la durée de l'instruction.

IV.4.3 LE CONTROLEUR

Le contrôleur de l'ADM génère tous les signaux indispensables au bon fonctionnement de l'ADM. Cette interface pilote l'interface électrique et le cœur d'ADM. Dans un souci de diminuer la complexité, le contrôle de l'ADM est réalisé en deux structures : le contrôle principal et le contrôle de protocole. Toutefois une hiérarchie entre les deux parties de contrôle est nécessaire pour gérer correctement le fonctionnement de l'ADM. Cette hiérarchie donne au contrôle principale la gestion des grandes tâches de l'exécution et au contrôle de protocole la gestion du protocole de communication au travers des configurations de l'interface électrique.

IV.4.3.1 Le contrôle principal

Cette structure est le séquenceur de l'ADM. Elle a pour but de séquencer une instruction dans ces étapes principales. Ces étapes sont :

- le démarrage d'une instruction,
- le traitement d'une instruction,
- la fin d'exécution.

Le contrôle principal récupère les commandes actives provenant du décodeur. Sur la base de ces commandes il génère les signaux nécessaires à l'exécution de l'instruction.

Dans le cas d'une instruction destinée à un traitement par le cœur d'ADM, le contrôle principal génère pour le démarrage de l'instruction un signal *op_code* spécifique à la

réalisation d'une fonctionnalité. Ce signal est de taille variable dans chaque ADM. Sa taille est fixée par le nombre d'opérations différentes supportées par le cœur d'ADM. Le contrôle principale génère aussi le signal *start* qui lance le traitement de l'instruction à son passage sur niveau haut.

Lors de la phase de traitement de l'instruction le contrôle principal se contente d'effectuer de la surveillance. Il intervient pour activer le contrôle de protocole si une requête d'accès au bus par la combinaison des signaux *Ld/St* et *Bus_req* de la part du cœur de DSP survient pendant le traitement.

A la fin de l'exécution par le cœur de d'ADM de l'instruction le contrôle principale récupère le signal de *Res_req* avant d'exécuter la procédure prévue dans le cas du code opératoire transmis. En effet selon le code opératoire le résultat peut être transmis à un autre ADM via les ports E/S (dans ce cas le cœur d'ADM se charge du transfert) ou peut être envoyé en mémoire (dans ce cas le contrôle de protocole configure l'interface électrique) ou le résultat doit être retourner directement au cœur d'ADM dans chacun de ces cas le contrôle principale génère les signaux adéquats.

En cas d'une instruction de configuration de l'interface électrique pour satisfaire le protocole de communication interne, le contrôle de principale transmet le *op_code* au contrôle de protocole qui se charge de son traitement.

IV.4.3.2 Le contrôle de protocole

Le contrôle de protocole génère tous les signaux nécessaires lors des accès aux bus dans le respect du protocole de communication pour éviter tout conflit d'accès

Les principales tâches prises en compte par cette structure consiste à placer les valeurs des masques correspondantes aux requêtes sollicitées par le cœur d'ADM, à générer les signaux pilotant les suiveurs bloqueurs et à transmettre au cœur d'ADM l'autorisation (*bus_free* = '1') d'accéder aux bus en activant le signal *enable*.

IV.5 LE CŒUR D'ADM

Le cœur d'ADM a pour rôle d'implanter une fonction complexe de l'application modem en utilisant une partie contrôle et une partie opérative câblée. La partie contrôle est une machine d'état générant des micro-instructions pour l'exécution de la fonction. La partie opérative implante un chemin de données câblé qui traite la fonction complexe.

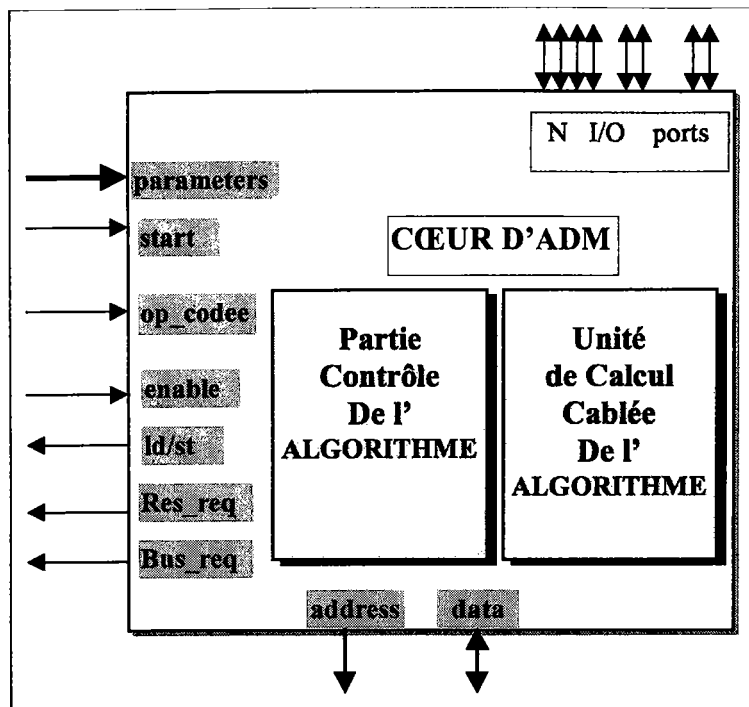


Figure 50 : Vue externe du cœur d'ADM

Le cœur d'ADM constitue la seule partie de l'ADM totalement spécifique à la fonction traitée. Ainsi le développement d'un nouveau ADM se caractérisera par une implantation différente de son cœur d'ADM.

IV.5.1 LA PARTIE CONTROLE

La partie contrôle du cœur d'ADM est en fait un sous contrôleur du contrôle principal. Il a pour but de générer sur la base du signal *op_code* du contrôleur principal, les micro-instructions nécessaires au traitement de la fonction. Cette structure est une machine d'état fini en relation direct avec la partie opérative.

Cette structure prend en charge en plus de l'exécution de la fonction, la génération d'adresses des données à calculer et la gestion des ports E/S. La transmission d'un résultat ou l'acquisition de données via les ports E/S est contrôlé directement par des signaux issus des micro-insructions de la machine d'état.

IV.5.2 LA PARTIE OPERATIVE

La partie opérative du cœur d'ADM est totalement dédiée à la réalisation d'une fonction complexe. Quelque soit la fonction à implanter, la partie opérative est développée de manière à proposer une architecture parallèle et paramétrable.

La stratégie adoptée consiste à rechercher dans un premier temps un algorithme

paramétrable permettant la réalisation d'un groupe de fonctions d'une même famille. Une étude de l'algorithme est alors effectuée pour favoriser un traitement parallèle des données.

L'objectif de cette étude est de proposer un algorithme exécutant la fonctionnalité par des boucles itératives. Ainsi la partie opérative implante le noyau de la boucle sous forme câblée en favorisant un traitement parallèle (exemple : par duplication de ce noyau). Le nombre de passage nécessaire à l'exécution complète d'une fonction est fixé par un paramètre de l'instruction. Cette information recueillie aussi par la machine d'état lui permet d'effectuer les commandes adéquates à l'exécution et au contrôle de la fonction complète.

La stratégie proposée peut ne pas convenir pour toutes les fonctionnalités. Une étude cas par cas pour chaque ADM doit être faite afin de proposer une solution architecturale offrant au mieux des contraintes de temps et de surface, une structure effectuant un traitement parallèle et paramétrable

Cependant toute partie opérative doit supporter un générateur d'adresse, des accès vers le bus de données et le bus d'adresse et un réseau de connexion adapté aux ports E/S

IV.6 CONCLUSION

Les ADMs constituent les éléments internes du processeur dédiés aux fonctions complexes de l'application modem. Ces modules ont en charge la réalisation de ces fonctions avec un haut niveau de performance autorisant le processeur à s'acquitter de la totalité des tâches pour différentes applications modem.

L'architecture des ADMs est développée de manière à respecter les différents aspects de notre processeur (protocole de communication, jeu d'instruction) ainsi que ceux liés aux traitements des fonctions complexes de l'application modem (solution câblée et paramétrable). L'architecture proposée est composée de trois parties principales : l'interface électrique, l'interface de contrôle et le cœur d'ADM.

L'interface électrique est un bloc logique combinatoire dont les composants logiques adaptent électriquement les signaux et bus externes aux signaux et bus internes à l'ADM. Dans ce sens la structure logique proposée dans cette interface fournit les éléments nécessaires à la gestion des chaînes de priorité et les éléments permettant de piloter les accès au bus sans conflit.

L'interface de contrôle est la partie commande de l'ADM. Elle génère les différents signaux pour activer une exécution ou une configuration. Cette interface prend en charge la réception, le décodage des instructions, le séquençement des grandes tâches de leurs

exécutions et le pilotage de l'interface électrique pour satisfaire les contraintes du protocole de communication du processeur.

Le cœur d'ADM est la partie de l'ADM effectuant le traitement d'une fonction complexe. Il est composé d'une partie contrôle et d'une partie opérative. La partie contrôle se présente sous forme d'une machine d'états finis exécutant un jeu de micro-instructions liés au code opératoire d'une instruction traitée par l'interface de contrôle. Cette partie contrôle est activée par l'interface de contrôle et génère les signaux pilotant l'unité de calcul. La partie opérative implante la fonctionnalité désirée caractérisant l'ADM. Cette structure est réalisée en logique câblée. Elle accepte des paramètres de configuration pour adapter son exécution à plusieurs fonctions d'une même famille.

Le choix de cette architecture pour les ADMs a été fait de façon à proposer un standard d'ADM où l'architecture des deux parties (interface électrique et interface de contrôle) est la même sur tous les ADMs. De ce fait la seule partie nécessitant le développement d'une architecture spécifique est le cœur d'ADM et plus particulièrement la partie opérative du cœur d'ADM. Ainsi la prise en compte le processeur d'une nouvelle fonction complexe se traduit par un travail spécifique sur une fonctionnalité et son algorithme sans remettre en cause l'architecture du processeur ni le protocole de communication.

Le chapitre suivant présente un exemple d'implantation de cœur d'ADM.

CHAPITRE V

EXEMPLE D'IMPLANTATION

DU CŒUR D'ADM

V.1 INTRODUCTION

Le chapitre précédent a présenté l'architecture des ADMs. Une architecture standard est proposée autour de l'interface électrique et de l'interface de contrôle. Cette architecture est générale pour tous les ADMs en tenant compte du protocole de communication (accès au bus, chaîne de priorité, réception d'une instruction, émission d'un résultat). Les ADMs sont destinés à la réalisation d'une fonction complexe à l'aide d'une partie opérative câblée. Cette propriété caractérise la spécificité de chaque ADM. L'implantation de l'algorithme est faite au niveau du cœur d'ADM. Les ADMs diffèrent les uns des autres seulement par leur cœur d'ADM.

Dans ce chapitre, une solution architecturale est proposée servant de modèle au développement de la partie contrôle et de la partie opérative d'un cœur d'ADM. Cette solution est dédiée au traitement de la fonction complexe, le contrôle de redondance cyclique (le CRC : Cyclical Redundancy Checking).

Cette fonction prise en compte dans les spécifications DAVIC s'est avérée être la fonction la plus consommatrice en temps de calcul dans la solution programmée de notre prototype.

L'implantation du cœur d'ADM nécessite une étude spécifique pour chacune des fonctions à réaliser (codage détecteur d'erreurs, codage correcteur d'erreurs, modulation, filtrage). Toutefois à l'aide de cet exemple une stratégie est proposée fournissant les différentes étapes de l'élaboration d'une solution d'implantation.

Dans un premier temps, nous présentons la fonction de codage du CRC. Un algorithme généraliste est proposé favorisant le traitement parallèle pour améliorer les temps de traitement.

Dans un deuxième temps, nous développons une architecture adaptée à l'algorithme proposé autorisant des configurations différentes par passage de nouveaux paramètres.

Dans un troisième temps, nous devons figer la machine d'états finis. Toutes les fonctions supportées par le cœur d'ADM sont décomposées en étapes pour déterminer les instructions principales de chaque traitement. Chacune de ces étapes est réalisée par notre architecture par activation de signaux et de commandes particulières. Les valeurs de ces signaux et de ces commandes sont regroupées dans la machine d'états finis pour servir de micro-instructions à une exécution

La dernière partie présente les résultats de simulation de l'implantation proposée (sur un composant FPGA) et donne lieu à une estimation des performances de notre structure.

V.2 LE CRC

La fonction CRC (Cyclical Redundancy Checking) est une technique de codage de données basée sur le principe des codes détecteurs d'erreurs. Cette fonction est utilisée par les spécifications DAVIC en émission et en réception. Elle permet de détecter dès la réception d'une trame de données la présence d'une ou plusieurs erreurs.

V.2.1 PRINCIPE

La fonction CRC consiste à effectuer la division polynomiale entre la trame de données et un polynôme diviseur connu de l'émetteur et du récepteur. Cette division a pour résultat une valeur de reste qui est insérée dans la trame de données. A la réception de la trame de données, le récepteur effectue la même opération. La comparaison de la nouvelle valeur du reste calculé avec la valeur du reste transmis permet de détecter la présence d'erreurs.

V.2.2 ALGORITHME

Les calculs effectués sur les symboles binaires de la trame peuvent être réalisés dans le corps des polynômes $K[x]$ (corps des polynômes modulo $x^n - 1$). Dans ce cas la trame de données $M = (m_0, m_1, \dots, m_{k-1})$ peut s'écrire comme le polynôme suivant :

$$M(x) = \sum_{i=0}^{k-1} m_i * x^i \text{ avec } m_i \in \{0,1\}$$

D'après le théorème d'Euclide, en considérant le polynôme diviseur noté $g(x)$ de degré r , il existe deux polynômes p et q tels que :

$$x^r M(x) = p(x)g(x) + q(x)$$

En utilisant un algorithme série, le nombre de cycles utilisé pour effectuer la division est supérieur au nombre de cycles nécessaire à la prise en compte des données. L'opération de division constitue dans ce cas le facteur limitatif de l'application. En parallélisant les opérations de codage et de décodage, ce facteur porte sur la lecture des données donc sur le débit d'entrée de l'ADM.

L'algorithme parallèle proposé [27] utilise le calcul de matrice basé sur la formule itérative suivante :

$$\begin{aligned}
[r_{d-1}^{k+1}, \dots, r_0^{k+1}] = & \\
& (e_{n(k+1)-1}, \dots, e_{n(k+1)-d}) \oplus \\
& (e_{n(k+1)-d-1}, \dots, e_{nk}, r_{d-1}^k, \dots, r_0^k) * [D]^{(1)}
\end{aligned}$$

Le niveau de parallélisme pour la division est donné par la valeur de n et le degré du polynôme diviseur est fixé par la valeur de d . La valeur de l'itération courante est fournie par k . Les e_i sont les bits d'entrée classés par ordre croissant. Le mot composé des bits r_i constitue la valeur du reste calculé lors de cette itération. Le reste final de la division sera donné lors de l'ultime itération.

A titre d'exemple, on prendra le polynôme d'ordre 6 de la forme $P_6(x) = x^6 + x + 1$ pour générer les matrices G et T de la façon suivante (1) :

$$[D] = \begin{bmatrix} G & & \\ G & * & T \\ - & & - \\ - & & - \\ G & * & T^{n-1} \end{bmatrix} \quad T_6 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_6 = [1 \ 1 \ 0 \ 0 \ 0 \ 0] \quad T_6 \text{ matrice transposée}$$

En utilisant un polynôme générateur d'ordre n , la même méthode définit les matrices G_n et T_n . Dans ce cas seuls les coefficients du polynôme et la taille des matrices présentent une différence avec l'exemple proposé.

V.3 LA SOLUTION ARCHITECTURALE DU CŒUR D'ADM

La solution architecturale proposée (voir figure 51) réalise la fonction de codage et de décodage du CRC avec la particularité d'effectuer ces fonctions à l'aide d'un bloc en logique câblée. Cette structure autorise le calcul d'une division polynomiale à partir de différents polynômes diviseurs. Comme pour tous les ADMs, la structure du cœur d'ADM est constituée de deux blocs. L'algorithme est traité par la partie opérative sous le contrôle d'une machine d'états finis (FSM; Finite State Machine).

La partie opérative fournit les éléments nécessaires pour réaliser trois fonctionnalités, le calcul de la fonction du CRC, les transferts des données et du résultat, et la gestion des paramètres.

L'architecture dédiée à chaque fonction est présentée dans les sections suivantes.

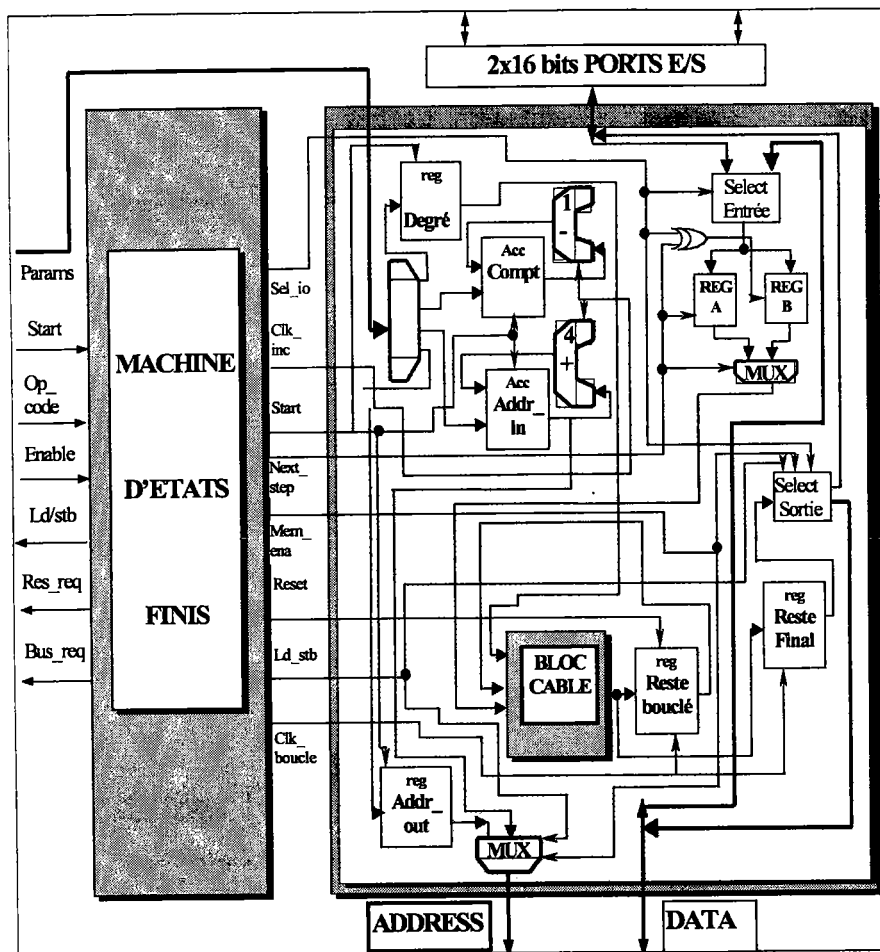


figure 51 : Le cœur d'ADM pour le CRC

V.3.1 REALISATION DE LA FONCTION DE CALCUL DU CRC

L'architecture dédiée au calcul de la fonction CRC se structure autour d'un bloc de logique câblé (bloc câblé).

L'architecture du bloc de logique câblée

Le bloc logique constitue la partie de traitement de l'algorithme sous forme câblée. L'architecture du bloc logique implante la formule (1) pour réaliser la division polynomiale. Cet algorithme améliore les performances des temps de calcul quand les données sont transmises en parallèle. Dans notre architecture, l'acquisition des données est faite à l'aide du bus de données de 32 bits autorisant un niveau de parallélisme sur 32 bits.

L'implantation de l'algorithme par la formule (1) s'appuie sur le constat suivant :

$$\text{Matrice [D]} = \begin{bmatrix} c_0^0 & - & - & c_0^{n-1} \\ - & - & - & - \\ - & - & - & - \\ c_{n-1}^0 & - & - & c_{n-1}^{n-1} \end{bmatrix}$$

$$(e_{n(k+1)-1}, \dots, e_{n(k+1)-d}) \oplus (e_{n(k+1)-d-1}, \dots, e_{nk}, r_{d-1}^k, \dots, r_0^k) * [D] = (r_0^{k+1}, \dots, r_n^{k+1})$$

avec

$$r_0^{k+1} = (e_{n(k+1)-d-1} * c_0^0) \oplus \dots \oplus (r_0^k * c_n^0) \oplus e_{n(k+1)-1}$$

$$r_n^{k+1} = (e_{n(k+1)-d-1} * c_0^n) \oplus \dots \oplus (r_0^k * c_n^n) \oplus e_{n(k+1)-d}$$

Comme le premier bit de données arrivé est de poids fort, une transformation de la matrice [D] peut ramener la formule à l'expression suivante:

1^{er} colonne de la nouvelle matrice [D] soit [d1]

$$\begin{bmatrix} 1 \\ 0 \\ - \\ 0 \\ c_0^0 \\ c_1^0 \\ - \\ c_{d-1}^0 \\ c_d^0 \\ 0 \\ - \\ 0 \\ c_{n-d}^0 \\ - \\ c_{n-2}^0 \\ c_{n-1}^0 \end{bmatrix}$$

$$(e_{n(k+1)-1}, \dots, e_{n(k+1)-d}, e_{n(k+1)-d-1}, \dots, e_{nk}, r_{n-1}^k, \dots, r_d^k, r_{d-1}^k, \dots, r_0^k) * [d1] = r_0^{k+1} \quad (2)$$

La valeur du premier bit de reste suivant à pour valeur :

$$r_0^{k+1} = e_{n(k+1)-1} \oplus (e_{n(k+1)-d-1} * c_0^0) \oplus \dots \oplus (r_0^k * c_n^0)$$

Le résultat est identique à la formulation précédente. La nouvelle matrice [D] sera complétée de la même façon en utilisant les coefficients appropriés. L'avantage de cette approche est de pouvoir fixer le format de la taille du mot considéré et la taille du mot de reste au niveau du parallélisme. Cette approche influe toutefois sur la taille de la matrice [D] qui

passer d'une matrice $(n \times n)$ à une matrice $(n \times 2n)$.

Le degré du polynôme générateur $G(x)$ a une influence sur le niveau de parallélisme à considérer. En effet la taille du mot de reste (fixé par le degré du polynôme) doit obligatoirement être inférieure par cet algorithme au niveau de parallélisme proposé. Or l'aspect paramétrable indispensable à une portabilité du CRC à d'autres applications nous oblige à proposer une architecture permettant la prise en compte de plusieurs polynômes générateurs. Aussi notre architecture est développée pour supporter les 16 polynômes diviseurs suivants :

$$P_1(x) = x + 1 ;$$

$$P_2(x) = x^2 + x + 1 ;$$

$$P_3(x) = x^3 + x + 1 ;$$

$$P_4(x) = x^4 + x + 1 ;$$

$$P_5(x) = x^5 + x^2 + 1 ;$$

$$P_6(x) = x^6 + x + 1 ;$$

$$P_7(x) = x^7 + x^3 + 1 ;$$

$$P_8(x) = x^8 + x^4 + x^3 + x^2 + 1 ;$$

$$P_9(x) = x^9 + x^4 + 1 ;$$

$$P_{10}(x) = x^{10} + x^3 + 1 ;$$

$$P_{11}(x) = x^{11} + x^2 + 1 ;$$

$$P_{12}(x) = x^{12} + x^6 + x^4 + x + 1 ;$$

$$P_{13}(x) = x^{13} + x^4 + x^3 + x + 1 ;$$

$$P_{14}(x) = x^{14} + x^{10} + x^6 + x + 1 ;$$

$$P_{15}(x) = x^{15} + x + 1 ;$$

$$P_{16}(x) = x^{16} + x^{12} + x^3 + 1 ;$$

Le choix de se limiter à ces 16 polynômes a été fait pour réduire la complexité et le coût en surface de notre architecture. De plus les applications proposent rarement l'emploi de fonction CRC avec un polynôme générateur de degré supérieur à 16. Notre architecture adapte alors son niveau de parallélisme au niveau le plus faible dans ces conditions soit $n = 16$.

La solution architecturale du bloc logique câblé peut effectuer une division polynomiale avec un des 16 polynômes proposés sur 16 bits parallèle. L'architecture proposée sur la figure 52 correspond au calcul d'un des 16 bits de reste à partir de la formule aménagée (2). L'architecture complète du bloc logique est constituée par 16 blocs identiques dédiés chacun au calcul d'un bit du reste.

La ROM permet de stocker les coefficients de la nouvelle matrice $[D]$ sous une forme particulière. Chaque bloc étant destiné au calcul d'un des r_i , la matrice $[D]$ ne regroupe pas les 16 colonnes de coefficients définies pour le calcul d'un polynôme mais les 16 colonnes de coefficients définies pour le calcul d'un r_i .

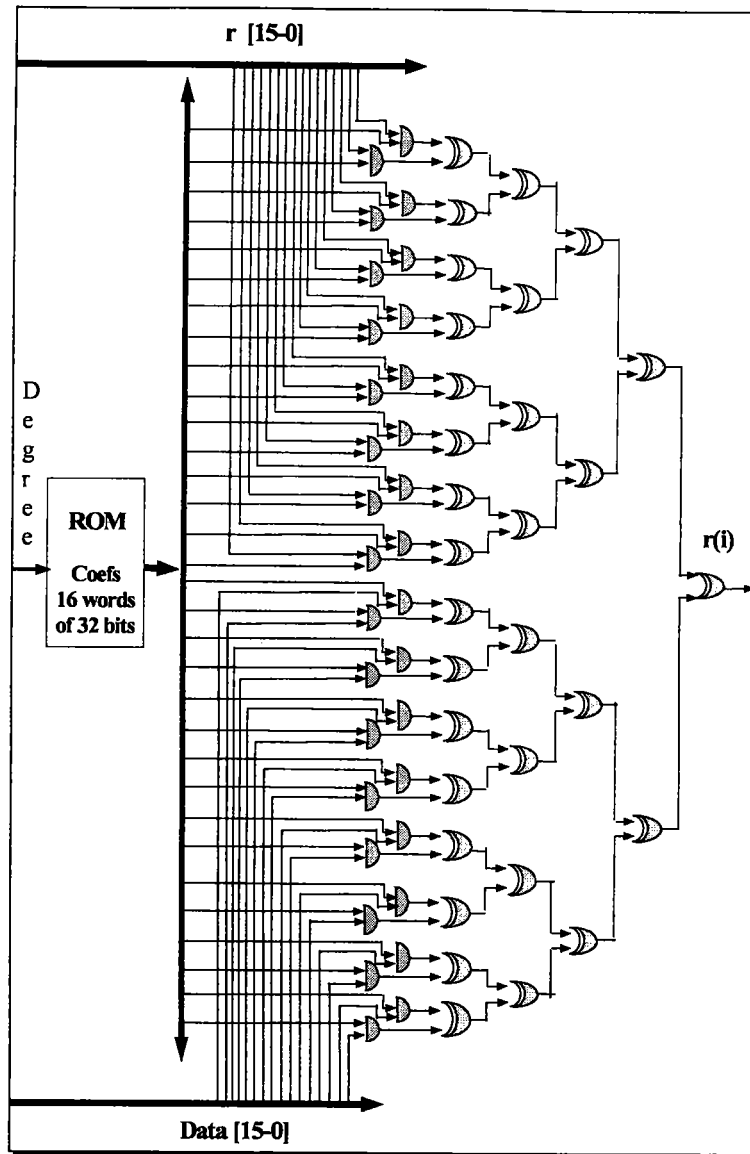


figure 52 : architecture d'un bloc de la structure câblée

Les entrées du bloc sont les 16 bits e_i de données arrivant en parallèle, les bits r_i du reste précédent et les 4 bits du degré. Les 4 bits du degré sont utilisés par la ROM pour sélectionner la colonne de coefficient relative au degré du polynôme transmis.

La fonctionnalité est assurée par le dispositif de la figure 53. Cette structure est composée d'un bloc de logique câblé et de deux registres de mémorisation. Le bloc logique traite les données en parallèle par groupe de 16 bits. Un rebouclage du reste calculé (r_i) est nécessaire pour effectuer le calcul sur les 16 bits suivants (e_i). En sortie du bloc logique deux registres (latch) de 16 bits sont utilisés pour stocker un résultat intermédiaire (*Reste Bouclé*) et pour stocker la valeur finale de la division (*Reste Final*). Un signal de commande (*clk_boucle*) est émis par la machine d'états pour autoriser sur front montant la mémorisation des données par les registres. Un autre signal de commande (*Reset*) est utilisé pour initialiser le premier mot de reste à zéro.

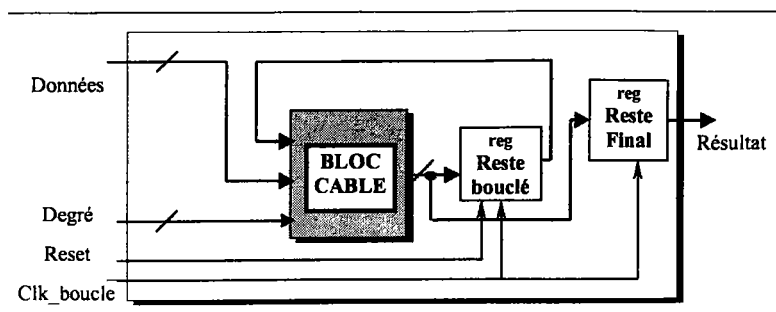


figure 53 : Architecture de la fonction calcul du CRC

V.3.2 REALISATION DES FONCTIONS DE TRANSFERT

Pour réaliser les transferts de données et du résultat, l'architecture doit fournir les éléments autorisant l'acquisition des données et la transmission du résultat soit par le bus de données soit par les ports E/S (*Select Entrée*, *Select Sortie*).

V.3.2.1 Architecture pour l'acquisition des données

L'architecture développée pour effectuer l'acquisition des données est proposée sur la figure 54. Le dispositif mis en place doit tenir compte de deux phénomènes :

- Les données peuvent utiliser deux voies de communication pour parvenir à la partie opérative.
- Le bloc logique traite les données sur 16 bits.

L'architecture adaptée à ces opérations est composée de quatre composants (*Select Entrée*, deux registres de 16 bits et un multiplexeur). Un composant "Select Entrée" permet en fonction du signal "Sel_e/s" de valider un chemin d'entrée (les ports E/S ou le bus de données). Ce composant découpe les 32 bits d'entrée en deux mots de 16 bits puisque le calcul du CRC par le bloc logique prend en compte des mots de 16 bits. Deux registres de 16 bits (registre A et registre B) permettent de les stocker (par convention les bits de poids forts sont mémorisés dans le registre A). Le calcul du CRC d'un mot de 32 bits s'effectue par deux passages dans le bloc logique. Le signal de commande *Next_Step* est utilisé pour passer au traitement d'une nouvelle itération.

Pour respecter l'algorithme utilisé, le bloc logique doit traiter en premier les bits de poids forts (contenus dans le registre A). La fonction logique "Sel_e/s XOR Next_Step" permet de repérer la première ou la deuxième itération du calcul d'un mot de 32 bits. Dans le cas où *sel_e/s* est à "1" signifiant la capture d'une nouvelle donnée, le calcul s'effectue alors sur le mot contenu dans le registre « A ». A la fin du traitement du mot A, le signal "Next_Step" redevient actif mais "Sel_e/s" ne l'est plus, le calcul est réalisé à partir du mot stocké dans le

registre B. Le multiplexeur (MUX) aiguille le mot de sortie des registres A ou B vers l'entrée du *bloc câblé*

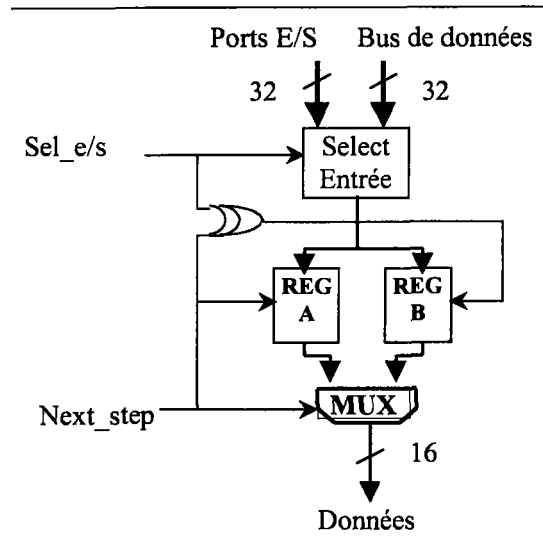


figure 54 : Architecture de la fonction d'acquisition

V.3.2.2 Architecture pour l'émission du résultat

L'architecture de la partie opérative insère le bloc "Select_Sortie" pour réaliser le transfert d'un résultat. Ce bloc a en entrée le mot du reste finale de la division polynomiale sur 16 bits. Ce bloc a pour fonction de compléter le mot de 16 bits en un mot de 32 bits en insérant des "0" sur les 16 bits de poids forts et de transmettre ce mot de 32 bits soit vers le bus de données soit vers les ports E/S. Pour réaliser cette fonctionnalité, ce bloc est piloté par trois signaux de commandes (*Ld_St*, *Mem_ena*, *Sel_e/s*).

Le signal *Ld/St* est actif si la structure veut émettre une donnée en sortie. Le signal *Mem_ena* est utilisé par la machine d'état pour préciser à la partie opérative si les accès aux bus sont autorisés. Le signal *Sel_e/s* spécifie la voie de communication à employer.

V.3.3 REALISATION DE LA GESTION DES PARAMETRES

La gestion des paramètres de l'instruction nécessite des registres pour capturer les différents paramètres, un générateur d'adresses pour déterminer les adresses mémoires des données à traiter et un décompteur pour fixer la fin du calcul itératif. Les paramètres sont transmis directement à la partie opérative à l'aide d'un bus de 80 bits. Les paramètres de l'instruction dédiée au CRC s'organise en quatre champs. Le premier champ fixe la taille du polynôme diviseur en utilisant 4 bits. Le second paramètre définit le nombre d'octets à considérer pour ce calcul sur 12 bits. Les deux derniers champs sont utilisés pour fixer l'adresse mémoire de la première donnée et l'adresse mémoire de retour.

L'architecture insère deux registres (*Degré* et *Addr_out*) et deux accumulateurs (*Compt* et *Addr_in*) pour mémoriser ces différents paramètres. Le nombre d'octet à considérer doit être décrémenté après chaque traitement d'un octet. Aussi l'architecture intègre un décrémenteur bouclé sur l'accumulateur "Compt". Pour générer une nouvelle adresse, l'architecture insère un compteur (+4) en relation avec l'accumulateur "Addr_in". La mémoire étant adressée en octet, ce compteur ajoute quatre à la valeur d'entrée (l'adresse précédente) pour récupérer les 32 bits de la nouvelle donnée.

V.4 LA PARTIE CONTROLE DU CŒUR D'ADM

L'algorithme de la fonction est implanté dans une partie câblée sous le contrôle d'une machine d'états finis (FSM). La FSM traduit le code de l'instruction fournie à travers les 8 bits du "op_code" bus et séquence alors les micro-instructions destinées à l'exécution de l'algorithme. Le signal "Start" légèrement déphasé par rapport au signal "Op_code" spécifie le début d'une exécution. Le traitement d'une instruction est découpé en plusieurs actions. A chaque action, le contrôleur active les commandes nécessaires à sa réalisation.

Un graphe d'états permet de fixer une chaîne de l'ensemble des actions possibles où les actions sont repérées en des états. Le graphe de la machine d'états est constitué comme le montre la figure 55 de trois sous-graphes : initialisation, terminal et traitement itératif.

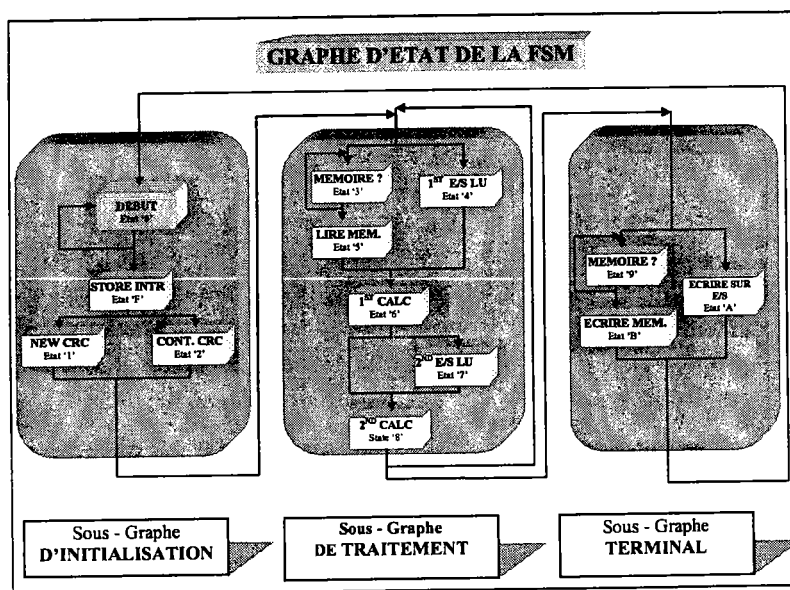


figure 55 : Graphe de la machine d'états pour le contrôleur du CRC

Le sous-graphe de traitement effectue le calcul du CRC sur un flot d'entrée de 2x16 bits en utilisant 4 cycles d'horloge. Ce résultat ne dépend pas du degré du polynôme. Ce sous-graphe est constitué de 6 états. L'état '3' est un état d'attente du signal "enable" pour accéder à la

mémoire. L'état '5' effectue la lecture des 32 bits de données en mémoire. Le traitement du premier mot de 16 bits dans le bloc logique est repéré par l'état '6' et le traitement du second mot de 16 bits est réalisé pendant l'état '8'.

Le sous-graphe d'initialisation débute à l'état '0'. La machine d'état reste dans cette état d'attente tant que le signal *start* n'est pas actif. Les signaux *op_code* et les *params* sont lus pendant l'état 'F'. Puis, en fonction du code de l'instruction un état d'initialisation est choisi (nouveau polynôme état '1' ou nouveau calcul avec le même polynôme état '2').

Le sous-graphe terminale commence dans l'état '9' en attente du signal "enable" autorisant l'accès à la mémoire. L'écriture en mémoire est faite sur l'état 'B'. Lors de cet état, le signal *res_req* indique la fin du traitement à l'interface de contrôle. Puis, la FSM revient à l'état '0' initial.

V.5 SIMULATION ET RESULTATS

Le cœur d'ADM dédié à la fonction CRC a été implanté sur un composant de la famille FPGA d'ALTERA le (FLEX10K). La description de l'architecture du cœur d'ADM a été faite dans le langage de modélisation VHDL (Very High Description Language).

La figure 56 présente le chronogramme de simulation concernant le traitement du mot de code instruction '81' en hexadécimal. L'interprétation de ce code instruction par la machine d'état a pour effet l'exécution d'un nouveau calcul de CRC sur des données provenant de la mémoire principale et dont le résultat sera stockée en mémoire principale. Dans cet exemple les 80 bits de paramètres sont respectivement 5 hex, 4 hex, 00000000 hex et 10000000 hex. La fonction à calculer dans cet exemple doit donc réaliser la division de 4 mots de 32 bits par un polynôme diviseur d'ordre 5.

V.5.1 PRESENTATION DES RESULTATS DE LA SIMULATION DU CŒUR D'ADM

Dans la figure 56, les différents champs du bus *params* de 80 bits sont découpés en trois bus *param1*, *param2* et *param3*. Le bus *param1* regroupe la valeur du degré '5' du polynôme diviseur et le nombre d'octet '4' à traiter. Le bus *param2* contient l'adresse mémoire du premier octet à traiter. Le bus *param3* contient l'adresse mémoire où sera stockée le résultat final. Pour faciliter la lecture des différentes étapes de la simulation, on visualise les étapes de la machine d'état par le signal *etat*.

Le calcul du CRC est composé de trois phases importantes, la phase d'initialisation (étape 1), phase de traitement (étapes 2,3,4,5) et une phase de fin d'exécution (étape 6).

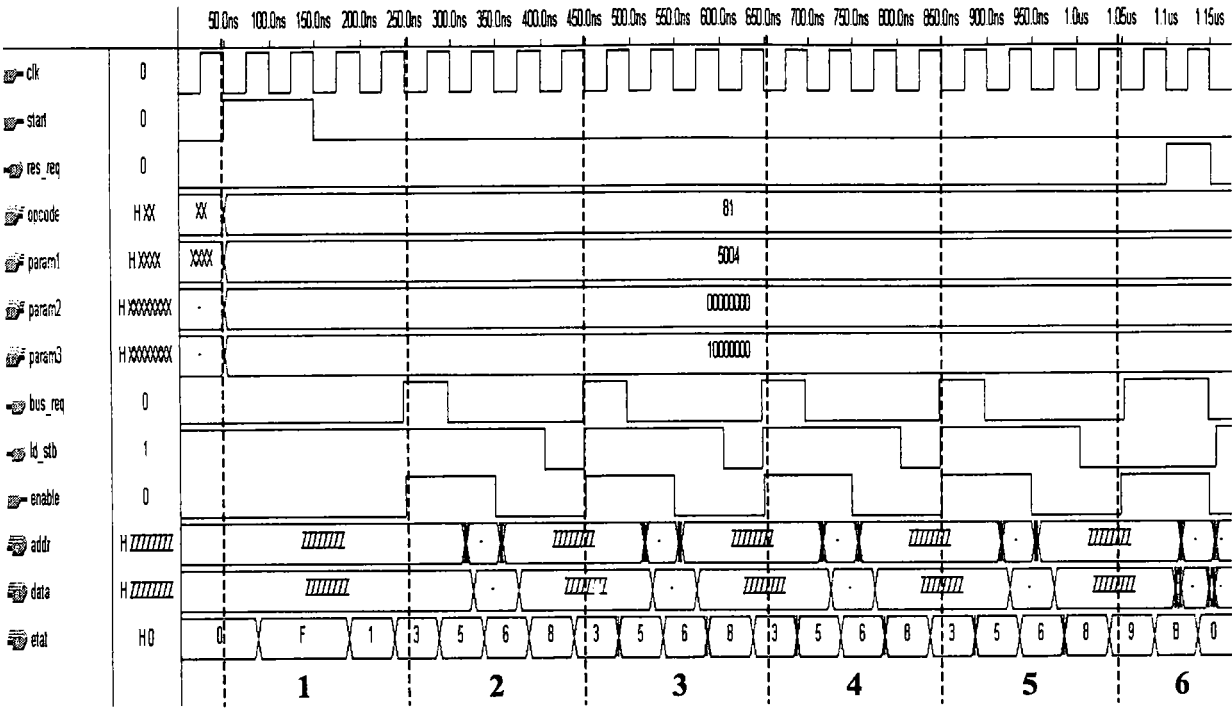


figure 56 : Chronogramme de simulation du cœur d'ADM sur un ALTERA/FLEX10K

L'étape 1 d'initialisation débute avec le signal *start* qui active le processus de la machine d'état. Le code de l'instruction (Op-code '81'hex) est alors traduit par la machine d'état (état 'F'). L'exécution d'un nouveau CRC est correctement interprétée par la FSM ayant pour conséquence le passage à l'état'1'.

La phase de traitement commence par l'état'3' sollicitant un accès au bus (signal *bus_req* actif) pour récupérer la donnée en mémoire. L'autorisation de l'accès à la mémoire se traduit par la mise à '1' du signal *enable*. L'adresse mémoire est alors véhiculée par le bus d'adresses *addr*. La lecture de la donnée (état '5') est faite sur le bus de données *data*. Le mot de 32 bits de données est découpé en deux mots de 16 bits pour respecter le niveau de parallélisme de notre implantation. Le calcul des 32 bits est réalisé par deux passages dans le bloc logique représentés par les états '6' et '8' de la machine d'état. La division d'un mot de 32 bits par le polynôme de degré '5' termine l'étape 2. Les étapes 3,4,5 sont utilisées pour le calcul du deuxième, troisième et quatrième octet de notre division. Leurs phases de traitement sont identiques à celle de l'étape 2.

La phase de fin d'exécution représenté par l'étape 6 débute par une requête d'accès à la mémoire pour y écrire le résultat(état '9'). La mise à '1' du signal *enable* autorise le transfert du résultat et de son adresse mémoire sur les bus *data* et *addr* (état '8'). Cette phase termine le traitement en activant alors le signal *res_req* pour indiquer au cœur de DSP la fin du traitement.

V.5.2 RÉSULTATS DE SIMULATION DE L'ADM COMPLET

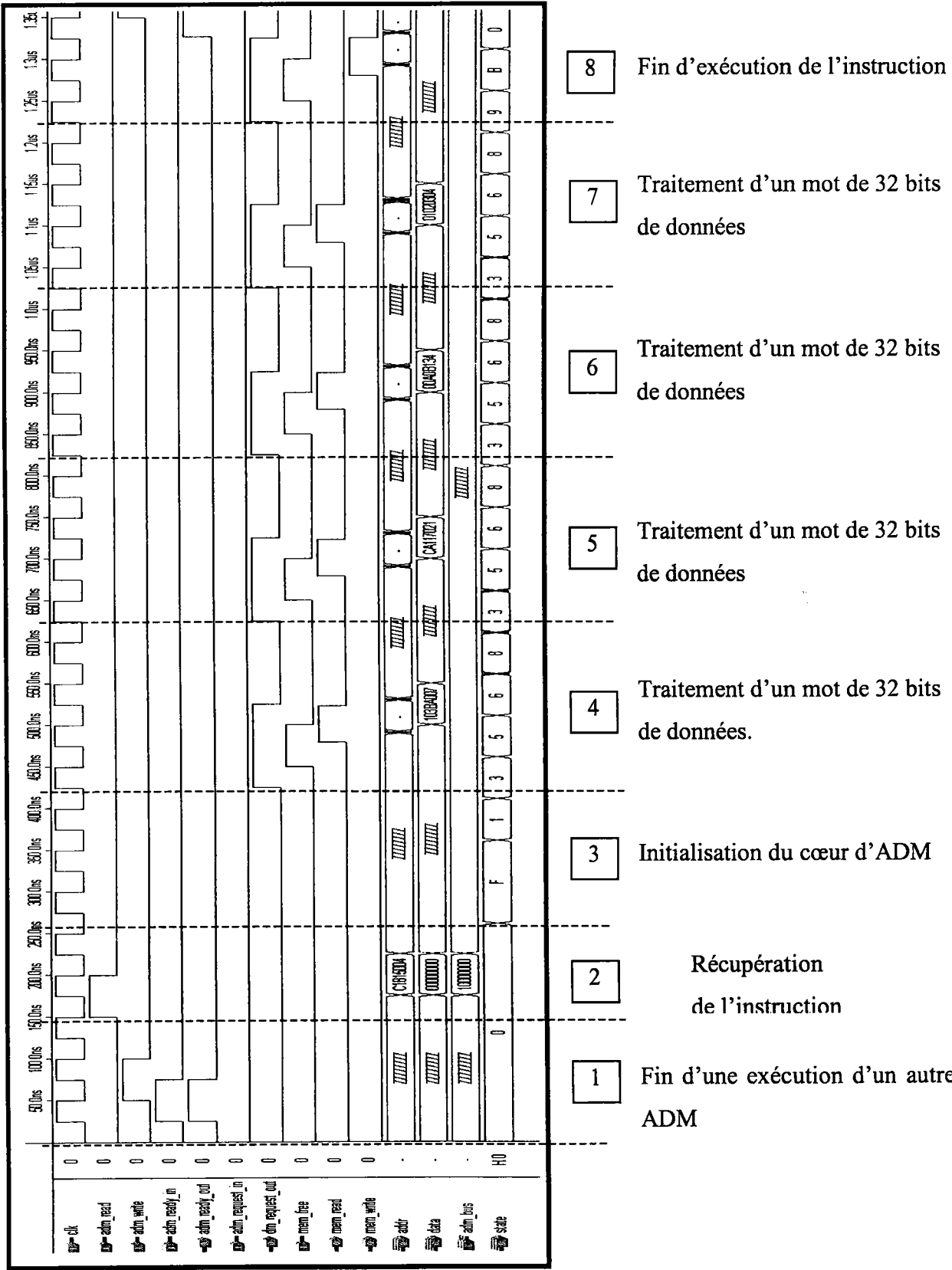


figure 57 : Simulation du même exemple pour l'ADM complet

La simulation présentée sur la

figure 57 fournit le comportement de l'ADM complet dans le cas de la même instruction que dans la simulation précédente. Cette simulation prend en entrée les signaux et bus d'entrée d'un ADM.

Le signal *adm_read* indique l'arrivée d'une nouvelle instruction ADM

Le signal *adm_write* est utilisé par le cœur de DSP comme accusé de réception d'un signal de fin d'exécution (*adm_ready_**).

Les signaux *adm_ready_in* et *adm_ready_out* gèrent les requêtes de fin d'exécution.

Les signaux *adm_resquest_in* et *adm_request_out* gèrent les requêtes d'accès aux bus.

Le signal *mem_free* est transmis par le cœur de DSP pour signaler la disponibilité des bus.

Les signaux *mem_read* et *mem_write* permettent à l'ADM de sélectionner le sens d'accès à la mémoire.

Les bus d'adresses, de données et d'instruction sont respectivement représentés par *addr*, *data* et *adm_bus*.

Cette simulation peut se diviser en 8 étapes principales.

L'étape 1 présente une requête de fin d'exécution d'un ADM plus prioritaire à travers les signaux *ADM_ready_in* et *ADM_ready_out*. Le cœur de DSP signale la prise en compte de cette requête en activant le signal *adm_write*.

L'étape 2 caractérise l'acquisition et le traitement d'une nouvelle instruction par un ADM. Celle-ci débute par la mise à '1' du signal *adm_read*. L'ADM prend en compte ce signal sur le front montant de l'horloge *clk*. L'ADM concerné par cette nouvelle instruction fait l'acquisition des 96 bits d'instruction par les 3 bus de 32 bits *addr*, *data* et *adm_bus*.

L'étape 3 de cette simulation est alors identique à l'étape 1 de la simulation précédente. De même les étapes 4, 5, 6, 7 de traitement des 4 mots de 32 bits de données correspondent aux quatre étapes 2, 3, 4, 5 de la première simulation. L'étape 8 de fin d'exécution correspond à l'étape 6 de la simulation précédente.

V.5.3 INTERPRÉTATION DES RÉSULTATS

Le cycle d'horloge pour l'implantation FPGA proposé est de 50 ns (soit 20 MHz). En considérant le temps de traitement de 32 bits sur le chronogramme (défini entre deux fronts montants du signal *adm_request_out* représentant une étape de traitement 4 ou 5 ou 6 ou 7), notre architecture peut effectuer la fonction CRC en respectant les contraintes d'un débit de 160 Mbits/s. Ce résultat est obtenu par une implantation FPGA ne dotant pas notre structure

des meilleurs technologies de conception. Une implantation ASIC devrait permettre de plus hauts débits.

Pour réaliser le prototype, une implantation "software" de cette fonction avec le même algorithme a été testée sur le DSP d'HyperStone à 100 MHz. Le même programme a aussi été évalué sur un PC utilisant un processeur Pentium 166 MHz d'INTEL et sur une station de travail Sparc 20 de Sun. Les résultats d'évaluation dans les trois cas ont montré que les débits limites de traitement pouvant être atteints sont respectivement : 1.67 Mbits/s pour l'HyperStone, 3.2 Mbits/s pour le Pentium et 4.5 Mbits/s pour la Sparc 20.

Ce résultat comparatif laisse présager de bonnes performances à notre processeur pour réaliser les applications modems complètes. Etant donné que la performance d'une chaîne pipelinée d'ADM est limitée par les performances de l'ADM réalisant la fonction la plus critique. Dans l'application modem mis en œuvre par le prototype, la fonction CRC était l'une des fonctions les plus critiques. Aussi le débit maximum des données ne devrait pas être détérioré dans l'architecture finale incluant tous les ADMs nécessaires à l'application modem.

V.5.4 ESTIMATION DE LA SURFACE

Une estimation de la surface occupée par l'architecture d'un ADM à partir des données obtenues lors de la synthèse sur un composant FLEX 10K20 d'ALTERA peut être calculée de la façon suivante.

Les caractéristiques du FLEX 10K20 par ALTERA indique une utilisation maximale de 1152 blocs élémentaires représentant 20000 portes logiques classiques. Les résultats de notre simulation dénotent une utilisation de 962 blocs logiques. Notre architecture occupe donc 83% des 1152 blocs logiques disponibles. La surface occupée est d'environ 16600 portes logiques. Un bloc élémentaire de la famille FLEX 10K est présenté sur la figure 58. Ce composant permet d'implanter plusieurs fonctions logiques. Les 962 blocs élémentaires nécessaires à notre architecture n'utilisent pas tous la complexité maximale d'un bloc élémentaire. Aussi l'estimation du coût de surface engendrée par l'implantation de l'ADM dédié au CRC doit être interprétée comme une valeur supérieure.

L'architecture du processeur dédiée aux applications modem sera composée de cinq ADMs (CRC, Reed-Solomon, QAM, filtrage, cryptage) et du cœur de DSP. La complexité de l'ADM dédié au CRC est de 16600 portes logiques. Cet ADM étant un des plus complexes, la borne supérieure du nombre de portes logiques d'un ADM ne devrait pas excéder les 18000 portes logiques. Dans le pire cas, la complexité des cinq ADMs est évaluée à 5×18000 portes

logiques (soit 90000 portes logiques).

L'architecture de notre cœur de DSP est inspiré de l'architecture du cœur de DSP TMS320C60 de Texas Instruments et sa complexité ne devrait être comparable. Les informations fournies par Texas Instruments [26] indiquent l'emploi de 550000 transistors pour réaliser leur cœur de DSP.

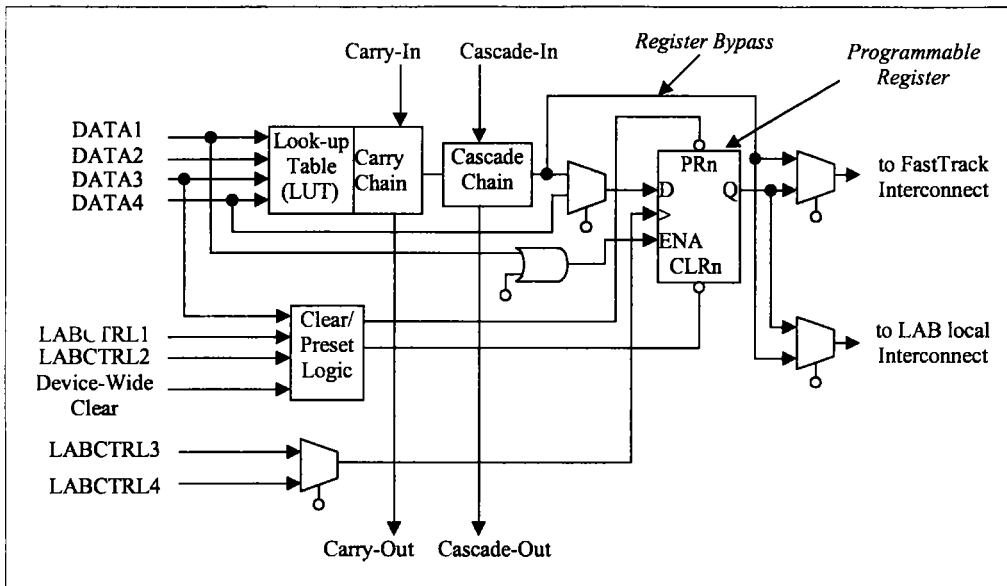


figure 58 : Architecture d'un élément logique pour la famille FLEX 10K d'ALTERA

V.6 CONCLUSION

L'objectif de ce chapitre est de réaliser une implantation d'un cœur d'ADM dédié à la fonction CRC. La fonction CRC constitue dans l'approche "software" un point critique limitant les performances. Au vue de ce résultat, l'amélioration des performances de cette fonction est une priorité pour la réalisation d'une application modem.

Afin d'augmenter la vitesse de traitement de cette fonction, l'architecture proposée insère un bloc de logique câblé pour implanter l'algorithme fonctionnel. Cette algorithme est défini pour autoriser le traitement des données en parallèle. En accord avec la stratégie adoptée, l'architecture propose une solution paramétrable et parallèle.

L'architecture de la partie opérative permet de réaliser une division polynomiale à partir de 16 polynômes diviseurs différents. L'algorithme fonctionnel implanté autorise un niveau de parallélisme sur 16 bits.

La description de l'architecture de ce cœur d'ADM par le langage de modélisation VHDL a permis une implantation de ce cœur d'ADM sur un composant de la famille FPGA d'ALTERA (le FLEX10K30). Les résultats de simulation obtenus ont montré que le cœur

d'ADM peut traiter la fonction CRC pour des applications nécessitant un débit de 160 Mbits/s. Ce résultat est d'autant plus prometteur qu'il devrait être fortement amélioré sur une réalisation ASIC. Il confirme les bonnes performances de notre approche pour réaliser les applications modems pour câble TV.

CHAPITRE VI

CONCLUSION ET

PERSPECTIVES

Conclusion et perspectives

Les travaux effectués dans cette thèse présentent les différentes approches envisagées pour réaliser des applications modem pour câble TV. Cette étude découle du constat lié à la prolifération d'applications multimédia dans le domaine des télécommunications (vidéo à la demande, télévision interactive, son numérique, etc.).

La diversité des traitements à prendre en compte avec l'apparition des applications multimédia, les débits très importants requis et l'évolution vers les réseaux câbles TV, exigent l'apparition de nouvelles normes et de nouveaux produits performants sur le marché. Un élément fondamental d'une telle structure est l'interface de connexion des dispositifs terminaux au réseau lui-même. L'interface devra assurer une interopérabilité matérielle et logicielle importante et pouvoir évoluer facilement pour s'adapter rapidement aux besoins du marché.

Cette interface se matérialise généralement sous la forme d'un modem. Celui-ci doit réaliser des fonctions diverses incluant la modulation et filtrage, le codage des données et le codage du canal. Les contraintes sur la vitesse de traitement et les volumes de données imposent le développement de nouvelles structures.

Une étude du réseau de communication DAVIC au travers de ces spécifications a permis de proposer une chaîne fonctionnelle pour deux modes de transmission sur le câble TV (unidirectionnelle et bidirectionnelle). Ces spécifications imposent l'emploi de fonctions complexes et coûteuses en temps d'exécution à la réalisation d'une application modem.

L'étude préalable du réseau DAVIC nous a mis devant une alternative : faut-il concevoir un modem dédié aux applications modem à base de composants spécialisés ? ou est il plus intéressant de choisir une conception d'un système autour d'un processeur ?

La conception d'un modem constitué de composants spécifiques est une solution que nous avons écartée pour plusieurs raisons : l'addition de plusieurs composants spécifiques est d'une part trop coûteux en surface pour une adaptation de ce modem à toutes les applications; d'autre part une telle structure manque de souplesse face à de nouvelles contraintes ou face à une nouvelle configuration. Cependant cette solution purement câblée dote le modem d'un traitement suffisamment rapide pour traiter les diverses fonctions dans le respect des débits.

La conception d'un modem pour câble TV autour d'un DSP à 1.5 Mbits/s a été réalisée

sous forme d'un prototype avec la société I&T Com dans le cadre d'un contrat CIFRE. Les problèmes rencontrés lors de la réalisation de ce prototype ont mis à jour les limites des solutions programmées pour respecter la contrainte de débit à partir des spécifications DAVIC.

L'expérience acquise lors de la réalisation du prototype nous a permis de définir les propriétés d'une architecture adaptée aux spécifications des applications modem. Une architecture modulaire, paramétrable et hautement parallèle est nécessaire pour fournir la souplesse, la flexibilité et les performances pour respecter les contraintes des applications modem.

Sur la base de ce constat nous proposons une autre voie d'étude qui consiste à développer un processeur adoptant l'architecture proposée ci-dessus et dont les parties opératives des fonctions très complexes sont câblées. L'architecture proposée est basée sur un cœur de DSP et sur un jeu d'ADM. Ces derniers sont des modules câblés dédiés susceptibles de traiter les fonctions les plus complexes. Un protocole de communication approprié établit le lien entre les ADM et le cœur. Son cœur de DSP est basé sur la philosophie VLIW qui offre plusieurs niveaux de traitement en parallèle.

Les principales caractéristiques de notre processeur sont la modularité grâce à l'utilisation de modules auxiliaires câblés appelés ADM (Auxiliary dedicated modules), sa capacité de traitement parallèle et sa flexibilité. Le haut niveau de parallélisme est assuré grâce au choix des interconnexions et du protocole existant entre les ADM et le cœur de DSP et par l'algorithme implanté dans chacun des ADM. Quant à la flexibilité, elle provient du fait que chaque ADM possède sa partie de contrôle (un petit jeu d'instruction propre à chaque ADM) et parce qu'il s'agit d'une architecture paramétrable et reconfigurable.

Un exemple d'implantation d'ADM est proposé pour la fonction CRC. Cette fonction réalise un codage par division polynomiale. Dans le souci perpétuel d'améliorer les performances de traitement. Un algorithme autorisant un traitement des données en parallèle est implanté sous forme câblée. De plus l'implantation est effectuée de manière à autoriser un paramétrage du polynômes diviseurs. La division polynomiale peut s'effectuer à partir d'un polynôme diviseur dont le degré est compris entre 1 et 16.

Une description de l'architecture de cet ADM à l'aide du langage de modélisation VHDL a permis une implantation sur un composant FPGA de la famille FLEX10K d'ALTERA. Les résultats de simulation ont confirmé les espoirs mis dans cette approche. Les temps de simulation ont montré que notre ADM réussirait à traiter un CRC pour une application nécessitant un débit jusqu'à 160 Mb/s. Les solutions programmées ne pouvaient plus suivre

à partir d'un débit d'une dizaine de Mbits/s.

Ce travail permet d'ouvrir une voie de recherche sur la réalisation d'un processeur dédié aux application en télécommunications. L'architecture présentée donne des résultats très encourageants. L'implantation des autres fonctions complexes est cependant nécessaire pour tester la capacité du processeur à traiter une chaîne fonctionnelle complète de l'application modem.

RESUME

Ce travail a été effectué dans le cadre d'un contrat CIFRE entre le laboratoire L.I.C.M (Laboratoire Interface Composant Micro-électronique) de l'Université de Metz et la société I&T COM associé à un projet Européen EURICO. Ce projet a pour but de développer un modem pour câble T.V à 1.5 Mbit/s.

La réalisation des applications modem dans le domaine des télécommunications est sujette ces dernières années à une grande évolution. Une nouvelle gamme d'applications propre au multimédia est apparue nécessitant des débits de plus en plus élevés et une multitude de traitement. Cette émergence a pour conséquence un développement de nouvelles structures matérielles. La norme du réseau DAVIC répond aux contraintes des applications multimédia et fournit les structures adaptées à l'interopérabilité.

Une étude des spécifications DAVIC (Digital Audio-VISual Council) sur câble T.V a permis de définir les contraintes fonctionnelles associées aux applications modem pour deux modes de transmission (unidirectionnelle et bidirectionnelle). La réalisation d'un modem compatible avec les réseaux DAVIC impose de traiter l'ensemble des fonctions contenues dans les spécifications.

Dans le cadre du projet EURICO un prototype est réalisé autour d'un processeur DSP du commerce en respectant des contraintes inspirées de DAVIC pour le débit de 1.5 Mbits/s. Les fonctions complexes imposées par DAVIC s'avèrent trop consommatrice en temps de traitement pour qu'une solution programmée puisse prendre à sa charge la totalité de l'application modem.

Sur la base de ce constat, le travail effectué consiste à développer une nouvelle architecture de processeur adapté à ces applications. L'architecture étudiée est basée sur un jeu de modules ADMs (Auxiliary Dedicated Module) agencés autour d'un cœur de DSP. Ces ADMs sont dédiés au traitement des fonctions complexes avec une partie opérative câblée. Une architecture modulaire, paramétrable et hautement parallèle est définie pour offrir la souplesse, la flexibilité et un niveau de performances suffisant pour réaliser les applications modems.

L'implantation sur FPGA d'une fonction complexe, le CRC, dans un ADM validé a permis de vérifier les performances de notre approche. Les résultats comparés avec ceux d'une solution programmée confirment les meilleures dispositions de notre processeur à traiter les applications modem.

Mots clés

réseau haut débit, multimédia, modem, DAVIC, DSP, architecture de processeur

ABSTRACT

This work is a part of an EURICO European project carried out in collaboration with the L.I.C.M laboratory (laboratoire Interfaces Composants Micro-électronique) and the I&Tcom society. The target of this project is the design of a T.V cable modem with a data rate of 1.5 Mbit/s.

In the last years, the design of modems has been subject to great advancements, in the telecommunication domain. A new range of applications has emerged dedicated to the multimedia domain requiring very high data rates and complex processing. The consequence has been the need for new design architectures.

The DAVIC (Digital Audio-VISual Council) normalisation group proposes adapted structures for interoperability between application on TV cable. The design of unidirectional and bi-directional modems on TV cable can be designed respecting the functional constraints from the DAVIC specifications.

A prototype design for the EURICO project was made using a standard DSP processor, complying the DAVIC specification and allowing a data rate up to 1.5 Mbits/s. However, the time constraints are hard to respect using a pure software solution as most of the DAVIC functions are critical.

Thus, a dedicated DSP architecture was designed to fit the time constraints of the applications. The architecture is based on a DSP core and a set of Auxiliary Dedicated Modules (ADM). Each ADM processes a time-critical function through a specific hardware data path. However, the architecture is strongly parallel, modular and configurable granting a high level of performance, as required by the TV modem applications.

The implementation of an ADM on an FPGA device, dedicated to the CRC function, proved the effectiveness of this solution. These results, largely above those of the pure software solution, confirm the good adequacy of our DSP architecture to the TV modem applications.

Keywords

High speed networks, multimedia, modems, DAVIC, DSP, processor architecture