

Les systèmes de traitement numérique du signal

Yann Bajot, Habib Mehrez

▶ **To cite this version:**

Yann Bajot, Habib Mehrez. Les systèmes de traitement numérique du signal. [Rapport de recherche] lip6.2001.021, LIP6. 2001. hal-02545594

HAL Id: hal-02545594
https://hal.science/hal-02545594

Submitted on 21 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Les systèmes de traitement numérique du signal

Bajot,Yann
Mehrez,Habib

LIP6.2001.021

Les systèmes de traitement numérique du signal

Sommaire :

1.1 INTRODUCTION 2
1.2 LE MARCHE DES CIRCUITS DE TRAITEMENT DU SIGNAL 3
1.2.1 Etat du marché 4
1.2.2 Domaines d'application 5
1.3 CARACTERISTIQUES DES SYSTEMES DE TRAITEMENT NUMERIQUE DU SIGNAL 6
1.3.1 Propriétés des systèmes numériques 6
1.3.2 Caractéristiques communes aux applications 8
1.3.3 Différentes contraintes pour différents domaines 10
1.3.3.1 Performance 10
1.3.3.2 Puissance consommée 12
1.3.3.3 Coût 13
1.3.3.4 Les compromis possibles 13
1.4 SOLUTIONS D'IMPLEMENTATION 14
1.4.1 Le coût des circuits numériques 14
1.4.2 Les ASICs 15
1.4.3 Les FPGAs 18
1.4.4 Les Processeurs programmables 19
1.4.4.1 Les processeurs DSP 19
1.4.4.2 Les microcontrôleurs 20
1.4.4.3 Les microprocesseurs 21
1.4.5 Comparatif 26
1.4.5.1 Matériel ou Logiciel 26
1.4.5.2 Microcontrôleurs, Microprocesseurs ou DSPs 29
1.5 EXEMPLE D'UN SYSTEME SUR PUCE POUR TELEPHONE MOBILE 32

1.1 Introduction

A moins de vivre totalement coupé du monde, sans télévision ni radio ni presse ou Internet, personne aujourd'hui ne peut ignorer le concept magique du «**numérique**». Cet adjectif béni des publicitaires est utilisé pour vanter les mérites de produits de diverses natures, du téléphone mobile à la télévision, et plus généralement de tout ce qui touche à l'électronique grand public. Cependant, par abus de langage ou volonté de simplification, ce terme est très souvent utilisé pour caractériser des systèmes réalisant des fonctions de ***traitement numérique du signal*** (en anglais *Digital Signal Processing* ou ***DSP***) : « télévision numérique » pour les réseaux câblés et satellites, « son digital » pour les téléphones mobiles, « appareil photo numérique » sont autant d'exemples faisant implicitement référence à la technologie DSP. Au-delà du phénomène purement marketing, ce succès traduit bel et bien la part désormais prépondérante de cette technologie dans les produits de grande consommation.

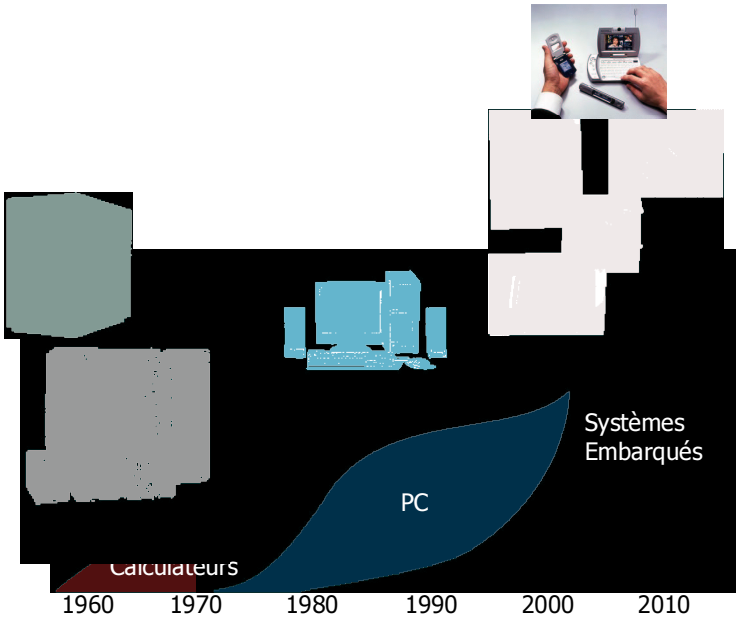


Figure 1 Les trois vagues de l'industrie des semi-conducteurs

Apparu dans les années 60-70 en même temps que les premiers calculateurs (Figure 1), le traitement numérique du signal est pendant longtemps resté cantonné à quelques domaines d'application particuliers: radar et sonar, industrie pétrolière, aéronautique et espace, et imagerie médicale. Ceci s'explique principalement par le coût prohibitif des ordinateurs à cette époque: le traitement numérique du signal constituait alors un luxe que seuls les militaires ou des industries particulièrement riches pouvaient s'offrir. L'arrivée des ordinateurs personnels dans les années 80 a contribué au développement de la technologie numérique, occasionnant d'énormes progrès dans la maîtrise des processus de fabrication, l'évolution des architectures matérielles et la performance des circuits intégrés. Mais ces efforts ont avant tout portés sur les microprocesseurs généralistes utilisés dans l'industrie des PC. Les circuits spécialisés en traitement du signal ont eux aussi profité de l'évolution

technologique mais sont globalement restés « à la traîne » par rapport aux microprocesseurs et microcontrôleurs, qui sont les moteurs de l'industrie de l'époque. Les quelques circuits implémentant des fonctions DSP sont des blocs câblés non programmables réalisant des traitements simples de type Filtre ou Transformée de Fourier.

Il faut attendre le milieu des années 80 pour voir apparaître les premiers processeurs programmables spécialisés dans le traitement du signal (en anglais *Digital Signal Processors*, ou *DSP*). L'architecture de ces processeurs reste alors relativement basique et découle directement de la structure des filtres numériques.

La véritable révolution survient au début des années 90 avec l'émergence des applications multimédias, de l'Internet et de la téléphonie mobile. La complexité de ces applications et les fortes contraintes en terme de performance liées à ces domaines obligent les concepteurs de circuits spécialisés DSP à envisager de nouvelles solutions matérielles et logicielles. C'est ainsi qu'on a vu apparaître dans les processeurs de traitement du signal des techniques architecturales modernes tels les jeux d'instructions VLIW, les architectures superscalaires et fortement pipelinées, les architectures multiprocesseurs, etc. Dans le même temps, les microprocesseurs généraux se sont dotés d'extensions matérielles spécialisées pour le multimédia (ex : l'extension MMX du Pentium) et les fabricants de FPGA ont commencé à proposer des librairies de macro-cellules orientées DSP. Les coprocesseurs de type ASIC (*Application Specific Integrated Circuit*) ne sont cependant pas en reste puisqu'ils restent incontournables pour implémenter les fonctions complexes qui ne peuvent se satisfaire d'une solution partiellement logicielle.

Aujourd'hui, les domaines d'application dans lesquels intervient le traitement du signal sont très nombreux, et la diversité des solutions matérielles d'implémentation est elle aussi très grande. Ce rapport présente le « domaine traitement du signal » et fait le point sur les différentes solutions d'implémentation. La section suivante donne un aperçu de la place des circuits DSP dans le marché des semi-conducteurs et présente les différents domaines d'application. La troisième section examine les caractéristiques particulières des applications DSP vis-à-vis du problème de l'intégration matérielle. La quatrième section détaille les différentes solutions d'implémentation en soulignant leurs avantages et leurs faiblesses respectifs. En conclusion, la dernière section présente un exemple de système DSP sur puce spécialisé pour la téléphonie mobile.

1.2 Le marché des circuits de traitement du signal

De plus en plus de fabricants d'appareils électroniques grand public font figurer la mention « DSP inside » ou « DSP powered » sur leurs produits comme argument de la puissance dudit produit. Aujourd'hui plus que jamais, le traitement numérique est à la mode à tel point qu'une compagnie comme Texas Instruments, pesant plusieurs milliards de dollars et dont les

activités couvrent un très large spectre dans le domaine des semi-conducteurs, affirme que son avenir dépendra avant tout de sa place dans le marché des circuits orientés DSP.

1.2.1 Etat du marché

L'évolution du marché des produits DSP s'inscrit dans le cadre de la croissance globale observée sur le marché des semi-conducteurs. Selon la société WSTS (World Semiconductor Trade Statistics), celui-ci a connu pour l'année 2000 une progression de 33 % par rapport à 1999, représentant une valeur totale de plus de 200 milliards de dollars. Les prévisions pour les quatre années à venir tablent sur une progression annuelle d'au minimum deux chiffres, avec une moyenne à 14% [1].

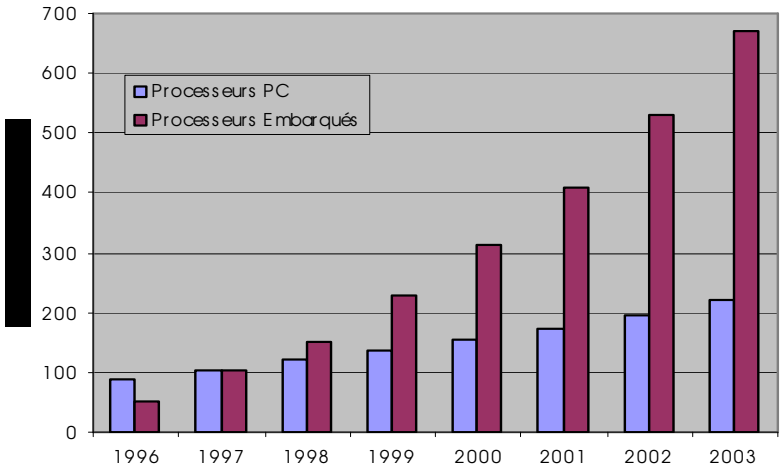


Figure 2 Comparaison du volume de vente des processeurs PC / Embarqués

Il faut cependant noter que le monde des micro-processeurs connaît depuis quelques années un bouleversement important: la part de marché des microprocesseurs destinés aux machines PC diminue graduellement au profit des processeurs destinés aux applications embarquées (Figure 2) [2]. Le rapport en 2003 devrait être de 3 pour 1 en faveur de ces derniers ; c'est ce que les analystes ont baptisé l'ère « Post-PC », dominée par les réseaux de communication, les applications clients-serveurs et les systèmes embarqués.

Dans ce contexte de forte croissance, le marché des produits DSP affiche à lui seul une progression de 57% [3], ce qui confirme son rôle moteur dans l'industrie des semi-conducteurs et s'explique principalement par le succès des circuits DSP dans les applications embarquées. Et ce phénomène semble devoir durer, comme le montrent les prévisions présentées Figure 3.

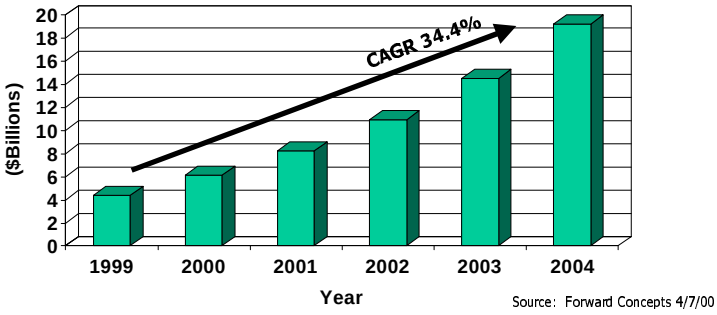


Figure 3 Evolution du marché des circuits DSP

1.2.2 Domaines d'application

Comme on l'a fait remarqué précédemment, c'est l'émergence de nouvelles applications utilisant massivement des algorithmes de traitement du signal et réclamant de plus en plus de puissance de calcul qui a contribué à l'essor du marché des circuits DSP. Les domaines d'application couverts par le traitement du signal sont particulièrement nombreux, comme l'illustre la Figure 4.

En terme de volume de production, trois domaines principaux se taillent aujourd'hui la part du lion : la téléphonie mobile, les modems et les contrôleurs de disque durs. Le marché des contrôleurs de disque durs, grand consommateur de processeurs DSP de faible puissance, voit son influence diminuer, l'accroissement continuel de la densité d'intégration faisant chuter les prix. Les modems analogiques restent un marché porteur grâce aux fournisseurs d'accès Internet qui les utilisent massivement pour leurs applications de type serveur, tandis que les solutions clients pour les usagers s'orientent plutôt vers les modems « soft », la puissance des PC actuels leur permettant d'effectuer ces taches de façon logicielle sans ralentissement notable du système.

Téléphone mobile	***Communication satellite***
Analyse sismique	***Cryptographie***
Détection de mouvement	***Téléphonie via Internet***
Commande Industrielle	***Sonar***
Radar	***Téléphone sans fil***
Appareils photo, caméra numérique	***Hi-fi***
Modems	***Vidéo-conférence***
Imagerie médicale	***Consoles de jeux***
Synthèse musicale	***Reconnaissance et synthèse vocale***
Navigation embarquée	***Acquisition de données***
Traitement d'image	***……***

Figure 4 Principaux domaines d'application

La téléphonie mobile, surtout, continue d'être le leader incontesté de la croissance du marché DSP grâce aux millions de téléphones vendus chaque année. Et l'avenir semble assuré avec l'arrivée prochaine des téléphones de troisième génération, intégrant des fonctionnalités Internet et Multimédia, qui devraient connaître un succès au moins égal à celui des anciennes générations de mobiles.

Parmi les technologies émergeantes et promises à un bel avenir, on peut citer l'*xDSL [4]* (principalement l'*ADSL* en France), qui semble enfin devoir décoller après des débuts difficiles : devant le succès de l'Internet par le câble, les opérateurs téléphoniques ont décider de réagir et cherchent à promouvoir cette technologie afin d'offrir à leurs clients des accès Internet haut-débit et illimité dans le temps. Plus généralement, les technologies offrant un accès large bande, comme l'ADSL, le câble ou le satellite, connaissent de plus en plus de succès ; les modems associés intègrent des protocoles plus complexes que pour les modems analogiques qui nécessitent de grandes capacités en terme de traitement du signal et donc l'utilisation de circuits spécialisés.

L'autre grand domaine en devenir concerne le Multimédia domestique, audio et vidéo. Les grands standards de codage multi-canaux (*Dolby Digital Surround, MPEG2, DTS, MP3*) utilisés en audio, ainsi que les formats de compression vidéo (*MPEG1 et 2*) utilisés par les décodeurs numériques ou les lecteurs DVD : tous ces algorithmes, complexes et coûteux en terme de puissance de calcul, sont implémentés sur des circuits spécialisés DSP.

Enfin, à plus long terme, la téléphonie sur Internet semble constituer un marché prometteur, au travers d'applications telles que *VoIP* (voice over IP) ou *FoIP* (fax over IP) dans lesquelles les traitements DSP seront prépondérants (encodage / décodage de voix) [5].

1.3 Caractéristiques des systèmes de traitement numérique du signal

Le traitement ***numérique*** du signal s'oppose au traitement ***analogique*** du signal, sur lequel se basait tous les anciens systèmes électroniques et qui manipule les signaux à leur état naturel, c'est-à-dire sous forme d'ondes (signaux *analogiques*). Le traitement numérique consiste à manipuler les signaux sous forme de suites d'échantillons numériques obtenus à partir de signaux physiques réels au moyen de convertisseurs analogique-numériques (*CAN*), leur appliquer les traitements mathématiques appropriés, puis éventuellement les restituer sous leur forme originelle à l'aide d'un convertisseur numérique-analogique (*CNA*) [6]. Les systèmes électroniques actuels, à l'image des téléphones mobiles, intègrent souvent les deux types de traitement, on parle alors de système mixte analogique/numérique.

1.3.1 Propriétés des systèmes numériques

Les systèmes numériques présentent plusieurs avantages par rapport aux systèmes analogiques. Le plus significatif d'entre eux est qu'un système numérique permet d'effectuer à peu de frais des tâches qu'il serait très difficile voire même impossible de réaliser en

analogique. Des applications telles que la reconnaissance de parole, la synthèse de parole ou les algorithmes de correction d'erreur utilisés dans les modems constituent de bons exemples : les traitements à effectuer ne sont pas systématiques mais dépendent des données reçues et sont donc très difficiles à implémenter en analogique.

Les deux autres avantages des systèmes numériques sont:

- *La résistance aux conditions d'environnement* : le comportement d'un système analogique dépend de facteurs extérieurs tels que la température, la proximité d'autres sources électromagnétiques, etc. Selon l'environnement, la réponse peut varier dans des proportions non négligeables. Les systèmes numériques, sauf conditions extrêmes, sont insensibles à l'environnement.
- *L'insensibilité à la tolérance et au vieillissement des composants* : deux systèmes analogiques absolument identiques n'auront pas exactement la même réponse du fait de la tolérance sur le comportement des composants, qui évolue aussi avec l'age. A contrario, deux systèmes numériques produisent des comportements rigoureusement identiques.

De ces deux caractéristiques, il résulte que les systèmes numériques ont un comportement *déterministe*, puisqu'ils sont insensibles aux conditions extérieures. Il est donc possible de concevoir des systèmes produisant des réponses exactes, chose impossible en analogique. De plus, sous réserve d'utilisation de circuits programmables, les systèmes numériques peuvent être *reprogrammables*, et peuvent donc effectuer d'autres tâches tout en se servant du même matériel.

Toutes ces considérations, ajoutées au fait que les systèmes numériques profitent des progrès toujours croissants réalisés dans le domaine de la VLSI (*Very Large Scale Integration*), font des systèmes numériques un support privilégié pour les applications de traitement du signal.

Compression audio et vidéo
Filtrage
Modulation/Démodulation
Codage Correction d'erreurs
Traitements audio / image (réduction de bruit, annulation d'écho)
Synthèse (parole, musique)
Détection / Extraction (reconnaissance de forme, de parole…)

Figure 5 Les principaux algorithmes DSP

1.3.2 Caractéristiques communes aux applications

Le succès des circuits spécialisés DSP s'explique par le fait que les architectures dites « classiques » ne peuvent satisfaire toutes les contraintes liées au domaine traitement du signal. Dans le cas contraire, il serait évidemment plus simple d'utiliser des circuits généraux comme des microprocesseurs type *Pentium*, qui présentent de nombreux avantages : faible coût, puissance, compatibilité avec le monde PC, richesse de l'environnement logiciel ,etc.

On peut classer ces contraintes en deux catégories: celles inhérentes à la nature même des algorithmes utilisés, et celles qui dépendent de l'application.

La Figure 5 présente les principaux algorithmes utilisés dans les applications DSP. Ils ont en commun certaines caractéristiques sur lesquelles se basent les architectures spécialisées:

- *Calculs répétitifs* : ces algorithmes sont généralement constitués de boucles de calcul effectuant le même traitement un grand nombre de fois. Les applications de filtrage, par exemple, sont représentées en langage C par une simple boucle *For* (voir Figure 6). C'est pourquoi les algorithmes DSP sont généralement considérés de type «flot de données », c'est à dire que leur comportement dynamique ne dépend que rarement de la valeur des échantillons traités. Ainsi, le temps passé dans une fonction de ce type est statistiquement presque toujours le même. A contrario, les applications d'usage général (outils CAO, suites bureautiques, compilateurs, interpréteurs…) sont plutôt de type « contrôle » (utilisant des structures de type *if-then-else, switch*…) et ont un comportement dynamique fortement dépendant des données. L'autre grande caractéristique des applications DSP est que la majorité des calculs effectués utilisent massivement deux principaux opérateurs arithmétiques : l'addition et la multiplication (Figure 7).

$$y = \sum_{i=0}^{N-1} x(i) * h(N-i)$$

Figure 6 Equation d'un filtre FIR

- *Précision numérique* : selon les applications visées, les systèmes de traitement du signal requièrent plus ou moins de ***précision*** numérique. Un domaine comme l'analyse sismique requiert une grande précision dans les données collectées puisque le but est de pouvoir détecter d'infimes variations du signal reçu ; à l'inverse, des domaines comme le multimédia y sont beaucoup moins sensibles. Un autre paramètre important est la ***dynamique*** du signal, défini par le rapport entre la plus petite et la plus grande valeur prise par le signal. Ces deux paramètres conduisent à choisir une méthode d'implémentation numérique parmi deux possibilités : l'arithmétique ***flottante,*** précise mais coûteuse en terme d'implémentation***,*** ou l'arithmétique ***entière***, plus complexe à

mettre en oeuvre mais plus « économique ». Il reste alors à définir la ***largeur*** de données utilisée, qui découle là encore d'un compromis coût / précision [6].

Algorithme	Equation	Type d'opération
Convolution, filtrage	y = y + x*h ;	Multiply/Accumulate (MAC)
Adaptation	yn = yn-1 + x*h ;	Multiply / Add (MAD)
FFT, multiplication complexe	$xr_n=xr_{n-1}.wr_{n-1}-xi_{n-1}.wi_{n-1}$; $xi_n=xr_{n-1}.wi_{n-1}+xi_{n-1}.wr_{n-1}$;	
Viterbi	a1 =x1+x2 ; a2=y1+y2 ; y= (a1>a2) ? a1 :a2 ;	Add/Compare/Select (ACS)

Figure 7 Fonctions classiques de traitement du signal

- *Bande passante mémoire importante* : parce que les algorithmes sont constituées de boucles de calcul consommant de grandes quantités de données, sous forme de flux (traitement audio) ou de matrices multidimensionnelles (traitement d'images), le débit de données nécessaire pour nourrir les unités de calcul impose une bande passante importante entre la mémoire et le cœur du système de calcul. Dans le cas contraire, le système est sous-utilisé et la contrainte-temps réel (voir ci-dessous) sera difficile à respecter.

- *Contrainte temps réel* : dans le domaine des applications générales type PC, la performance en terme de vitesse d'exécution est importante mais pas primordiale : on autorisera un traitement de texte à « ramer » de temps à autre, pourvu qu'en moyenne sa performance soit satisfaisante. Il n'en est pas de même pour les systèmes DSP qui reçoivent des flux de données provenant du monde extérieur à une cadence régulière, leur appliquent un traitement particulier et doivent ensuite les restituer en temps-réel. Un lecteur de CD audio, par exemple, lit un échantillon sonore à la fréquence de 44.1 kHz. Un système effectuant un post-traitement sur les données lues par le lecteur CD devra donc être capable d'effectuer toutes les taches nécessaires au traitement d'un échantillon dans un délai inférieur à 22.6 µs (1/44100 s). L'incapacité de maintenir cette cadence de travail conduirait à restituer un signal erroné et donc de mauvaise qualité. On demande donc aux systèmes DSP d'être particulièrement robustes vis-à-vis du temps-réel, c'est pourquoi on privilégie souvent les systèmes dont le comportement temporel est déterministe.

Les architectures des processeurs DSP sont conçues en tenant compte de ces propriétés ; elles intègrent par exemple des mécanismes matériels de gestion des boucles afin d'accélérer le traitement des noyaux de calculs répétitifs, des unités de calcul spécialisées tenant compte de la précision numérique choisie, plusieurs bus de données pour assurer la bande passante mémoire, etc.…

1.3.3 Différentes contraintes pour différents domaines

En plus des propriétés inhérentes aux algorithmes de traitement du signal eux-mêmes, chaque application fixe un cahier des charges qui influe fortement sur le choix du système d'implémentation. Dans le domaine de la micro-électronique, il existe quatre principaux facteurs critiques: la performance, le coût, la puissance consommée et le temps de conception. L'importance que revêt chaque facteur caractérise l'application et donne déjà des indications quant au choix du système d'implémentation.

1.3.3.1 Performance

La performance requise par une application est directement fonction de la contrainte temps-réel et de la complexité des algorithmes mis en œuvre. Plus la contrainte temps-réel est forte, moins le système dispose de temps pour exécuter les algorithmes requis. Si les algorithmes eux-mêmes sont complexes, c'est-à-dire que le nombre d'opérations nécessaires pour les réaliser est très élevé, alors la *charge de calcul* (le nombre d'opérations à réaliser par unité de temps) requise par l'application est importante et le système devant la réaliser doit être puissant. En règle générale, la contrainte temps-réel d'un système est fortement liée à la fréquence d'échantillonnage des données qu'il traite. Les systèmes Radar fonctionnent à des fréquences très élevées (Figure 8) et doivent donc traiter de grandes quantités de données en un temps très court, mais les algorithmes mis en œuvre sont relativement simples ; au contraire, les modèles numériques utilisés en météorologie ne sont pas soumis à une contrainte de temps très forte (quelques prévisions dans la journée suffisent en général) mais la complexité des algorithmes est infiniment plus grande. Ces deux types d'application, bien que soumises à des contraintes différentes, requièrent donc toutes les deux des systèmes disposant d'une grande puissance de calcul.

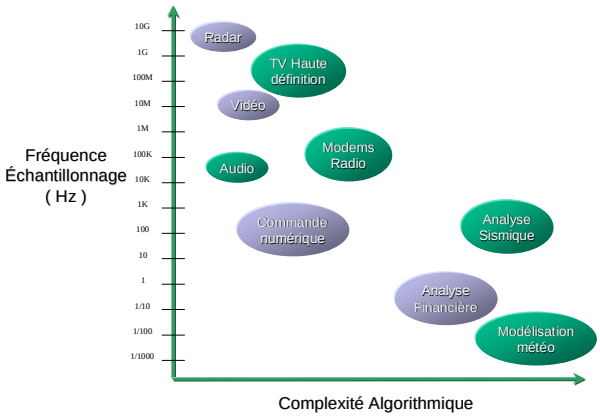


Figure 8 Estimation Fréquence / Complexité pour différents types d'application

Pour une même application, la complexité peut varier avec l'ajout de fonctionnalités. Prenons l'exemple des téléphones mobiles : les systèmes intégrés dans les premiers modèles se « contentaient » d'implémenter les algorithmes de base spécifiés par le standard GSM : codage canal, codage de voix, égalisation. Les progrès en terme de procédés de fabrication

ayant permis d'intégrer de plus en plus de matériel pour le même coût, les fabricants ont alors pu proposer des fonctions supplémentaires pour améliorer la qualité du produit : meilleure qualité sonore, reconnaissance de voix, modem intégré, tuner FM, fonctionnement bi-bande, navigation Internet (WAP), etc. Ces ajouts de fonctionnalité accroissent la charge de calcul nécessaire au respect de la contrainte temps réel, et poussent donc à l'utilisation de circuits numériques de plus en plus puissants.

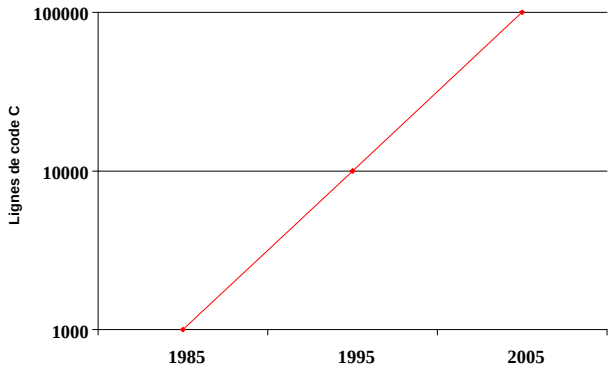


Figure 9 Evolution de la complexité des algorithmes DSP

La demande continuelle de plus grande performance, de meilleure qualité et de nouvelles fonctionnalités font que les algorithmes utilisés sont de plus en plus sophistiqués, et donc plus lourds à mettre en œuvre. Comme le montre la Figure 9, la complexité des algorithmes augmente d'un facteur dix tous les dix ans. Cette évolution exponentielle permet de tirer parti des puissances de calcul offertes par les circuits intégrés les plus performants, suivant en ce sens la loi de Moore décrivant les progrès en matière de procédés d'intégration (voir Figure 10) [7].

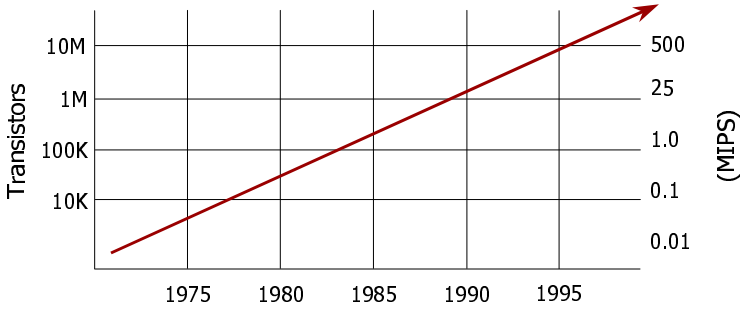


Figure 10 Loi de Moore

En toute logique, les circuits intégrés utilisés dans les systèmes électroniques devraient donc toujours utiliser au maximum les potentialités offertes par la technologie, en terme de fréquence de fonctionnement comme d'intégration matérielle, afin d'offrir des performances maximales. Or, ce n'est pas le cas. L'utilisation des processeurs DSP, par exemple, ne se justifie plus par l'écart de performance par rapport aux processeurs généraux, puisque cet écart tend vers zéro et est même dans certains cas défavorables aux processeurs DSP. La

raison d'être de ces processeurs est que leur architecture offre de meilleurs résultats en terme de puissance consommée et de coût matériel, qui sont les nouveaux facteurs critiques pour le domaine de l'électronique embarquée.

1.3.3.2 Puissance consommée

Alors que pendant des années les recherches en terme d'architecture de circuits intégrés se sont avant tout focalisées sur les moyens d'accroître la performance des circuits, un des grands défis actuel en matière de conception se situe maintenant au niveau de la réduction de la puissance consommée. En effet, malgré la diminution continue de la tension d'alimentation des circuits et l'amélioration des méthodes de conception, la puissance des microprocesseurs embarqués continue d'augmenter rapidement (d'un facteur 4 tous les 3 ans, cf. Figure 11). A coté de cela, la progression de la capacité des batteries portables est beaucoup plus faible (un facteur 2.5 en 30 ans, cf. Figure 12). Les demandes toujours croissantes en puissance de calcul des applications embarquées (en 2005, les terminaux 3G auront les mêmes capacités multimédia qu'un PC) réclament des circuits à haute performance et consommant peu. Or, avec les processeurs actuels, un système 3G aurait une autonomie de 10 minutes ou réclamerait une batterie de 30 kg ! [8].

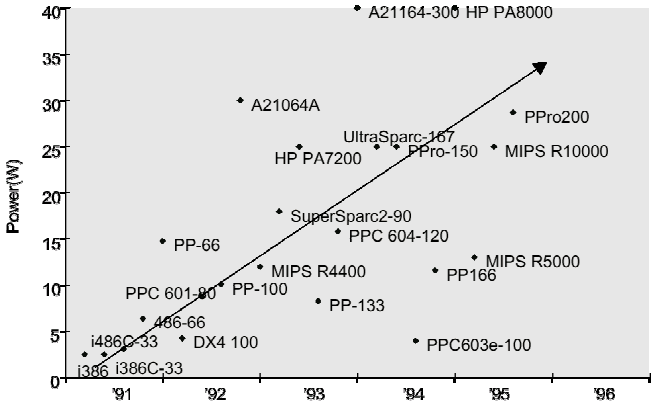


Figure 11 Evolution de la consommation des microprocesseurs

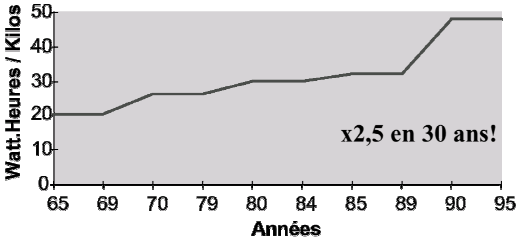


Figure 12 Evolution de l'autonomie des batteries portables

La consommation est ainsi devenu le critère le plus important pour le domaine de l'embarqué. L'autonomie est devenue un argument de vente majeur et les industriels cherchent à réduire au minimum la consommation des systèmes dans leurs produits. Les perspectives d'augmentation de la capacité des batteries étant limitées, les efforts se tournent vers la

conception de circuits assurant de grandes puissances de calcul pour une consommation la plus faible possible.

1.3.3.3 Coût

L'importance du coût d'un circuit numérique pour une application donnée varie selon deux principaux facteurs :

- la sensibilité de l'application à la notion même de coût. Les super-ordinateurs type *Cray* visent avant tout la performance et ne sont utilisés que pour des applications critiques par des industries aux moyens financiers très confortables. Ces machines sont en générale produites en faible quantité et peu d'efforts sont portés sur la réduction du coût. A l'inverse, les microcontrôleurs intégrés dans les disques durs visent un marché grand public très concurrentiel et réclamant une production massive de circuits. L'importance du coût à l'unité est alors beaucoup plus important, et peut devenir le critère de sélection entre plusieurs circuits offrant les mêmes performances. On voit au travers de ces exemples qu'en règle générale l'importance du coût est étroitement lié au volume de production requis.
- l'importance du coût du circuit dans le système global. Le coût d'un microcontrôleur utilisé dans une automobile n'aura évidemment pas la même importance que si il est utilisé dans une console de jeux.

1.3.3.4 Les compromis possibles

Pour une application donnée, le système idéal est celui qui assure la performance maximale tout en minimisant les trois autres paramètres. En réalité, ce système ne peut exister du fait de la forte corrélation qui existe entre les différents facteurs: une performance élevée ne peut généralement s'obtenir qu'au prix d'une consommation et d'un coût élevés. Concevoir le bon système est une affaire de compromis dans laquelle les considérations à prendre en compte sont autant d'ordre économique que scientifique.

Dans le domaine de l'électronique embarquée, les circuits numériques sont soumis principalement aux contraintes de consommation et de coût. Le défi pour les fabricants de téléphones mobiles consiste à la fois à offrir d'avantage de services à l'utilisateur et à améliorer l'autonomie, ce qui revient à augmenter la puissance de calcul tout en réduisant la puissance consommée. Dans le même temps, les appareils doivent être le moins cher possible si ils veulent avoir une chance de percer dans un marché sur lequel se battent plus d'une dizaine de fabricants. Ceci explique pourquoi on ne trouve toujours pas de Pentium dans les mobiles, puisque celui-ci consomme 100 fois plus et coûte 50 fois plus cher qu'un processeur DSP (cf. Figure 25, p.31).

A l'inverse, les systèmes de type serveur comme ceux équipant les stations réceptrices des réseaux téléphoniques doivent être capable de traiter en parallèle de grandes quantités

d'appels. Les circuits utilisés sont alors choisis avant tout en fonction de leur puissance de calcul, les critère coût et consommation passant en second plan. Cette différentiation des contraintes a une incidence directe sur la conception même des circuits intégrés. Ainsi, l'offre en matière de processeurs DSP tend actuellement à se séparer en deux grandes catégories : d'un coté les processeurs très faible consommation / performance moyenne, de l'autre les processeurs grande puissance / consommation importante.

Une fois définies les contraintes de coût, de performance et de consommation, le concepteur va devoir choisir quels types de circuits intégrer pour obtenir un système performant et économique. La section suivante examine les différentes solutions matérielles qui permettent l'implémentation d'un système DSP.

1.4 Solutions d'implémentation

Les processeurs DSP ne sont pas les seuls circuits à pouvoir prendre en charge les tâches de traitement du signal. Il existe de nombreuse alternatives parmi lesquelles on trouve les microcontrôleurs et processeurs généraux, les circuits programmables FPGAs et les ASICs.

La complexité des applications DSP varie de quelques dizaines de MOPS (*Million Operations per Second*) pour les applications simples de type « commande de moteur », jusqu'à plusieurs milliers de MOPS pour des applications complexes comme l'encodage MPEG2. La première question que doit se poser le concepteur est de savoir quels sont les systèmes pouvant supporter la charge requise par l'application. Il doit ensuite tenir compte des autres contraintes du cahier des charges (coût, puissance, temps de développement) afin de sélectionner le système approprié. Ce processus de sélection est loin d'‘être trivial et revêt une importance capitale, un mauvais choix pouvant avoir de graves conséquences sur la viabilité technique ou économique du circuit.

Avant d'étudier en détail les avantages et inconvénients des différentes solutions, il convient de revenir sur la notion de coût présentée dans la section précédente.

1.4.1 Le coût des circuits numériques

Le coût financier d'un circuit intégré du point de vue du fabricant est une notion plus complexe qu'il n'y paraît, et difficile à évaluer *à priori*, c'est-à-dire au tout début du flot de conception. On peut l'estimer comme la somme de deux composantes principales: le coût de production « brut » du composant, et le coût de développement.

Le coût de production dépend de nombreux facteurs tels que la technologie utilisée, le coût du test, le rendement de fabrication, le type de packaging employé, etc.[9] Du point de vue d'un concepteur de circuits, la surface de silicium est un facteur très important puisque le coût de production d'un circuit en fonderie est directement lié à la surface employée. Le nombre de « pattes » externes de l'interface a aussi une grande influence pour les circuits destinés à

l'intégration sous forme de composants discrets. Dans le cas qui nous intéresse, celui des cœurs de circuits intégrables dans des *System-On-Chip (Soc)*, ce dernier facteur n'intervient pas ; la recherche de la surface minimale devient donc le principal moyen pour réduire le coût d'un cœur de circuit.

Le coût de développement est quant à lui proportionnel au temps de conception et au coût humain, évalué en nombre d'*Hommes/Année*. Son importance par rapport au coût final est inversement proportionnel au volume de production. Les facteurs qui influent sur le coût de développement sont la complexité du circuit, le temps de test, la disponibilité et la qualité des outils de conception. A l'heure où l'industrie électronique utilise de plus en plus de standards (*MPEG* pour la vidéo, *GSM* et *UMTS* pour la téléphonie), la notion de flexibilité est devenue primordiale. Une solution flexible permet de prendre en compte les changements de standards au dernier moment, et évite ainsi de re-concevoir le circuit. Le gain est double : coût de conception réduit et temps d'accès au marché (*Time-to-Market*) accéléré, ce dernier point étant particulièrement critique dans les marchés très concurrentiels comme celui de la téléphonie mobile.

1.4.2 Les ASICs

Dans l'absolu et quelque soit l'application visée, le meilleur résultat en terme de coût matériel *et* de performance peut être obtenu par l'utilisation d'un circuit ASIC dédié. Dans l'espace des solutions d'implémentation matérielle, le circuit ASIC représente la solution « 100% matérielle » à un problème donné. Son architecture est conçue en fonction de l'application et spécialement étudiée pour réduire au minimum les facteurs de consommation et de surface, tout en assurant la performance requise par l'application.

Le caractère « flot de données » des algorithmes de traitement du signal est particulièrement adapté aux architectures de type *pipeline* et *systolique* [10], qui permettent la parallèlisation des calculs dans le temps et dans l'espace et offrent ainsi des performances de calcul très importantes. De nombreuses réalisations ASIC implémentant des algorithmes DSP reposent sur ces modèles d'architecture, dont la régularité facilite la réalisation matérielle et offre des performances très supérieures à celles que l'on peut obtenir avec des solutions programmables. Prenons l'exemple classique d'un filtre FIR. L'architecture systolique présentée Figure 13 permet d'obtenir un débit optimal d'un résultat par cycle dans le cas d'un filtre de longueur 4. Pour ce faire, tous les produits et sommes de l'algorithme sont exécutés en parallèle, des registres mémorisant les résultats intermédiaires requis par le découpage temporel de l'algorithme.

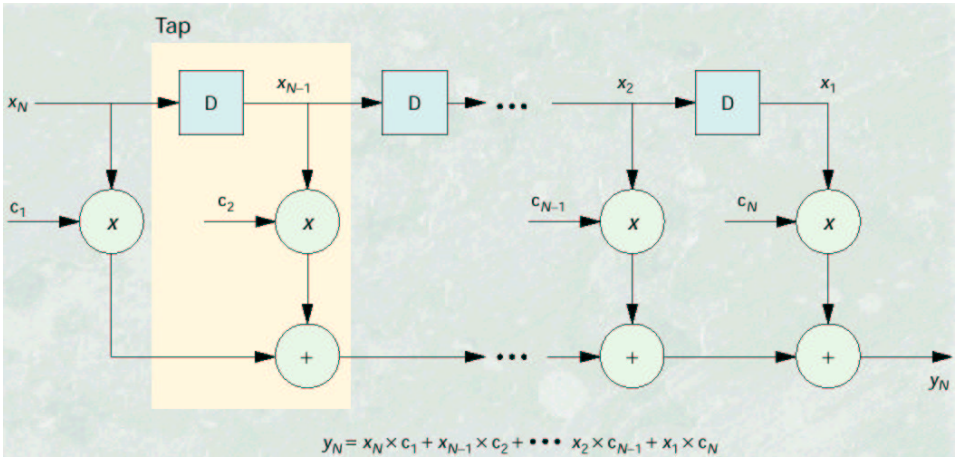


Figure 13 Architecture parallèle pour un filtre FIR

La conception de telles architectures optimisées pour un type d'application pose deux problèmes majeurs : un temps de conception très long, et une flexibilité nulle puisque la fonctionnalité du circuit est totalement figée après la fabrication. La volonté de maximiser le rapport *performance / coût matériel* impose que chaque étape du flot de conception, de la description de l'algorithme en langage de haut niveau jusqu'à la réalisation physique, fasse l'objet de longues phases d'études et d'optimisations. La réduction du nombre d'unités fonctionnelles utilisées et de la densité du chemin de données permettant de réduire le matériel passe par une étude très précise du flot de données inhérent à l'application, et la recherche d'un modèle d'architecture correspondant. Le flot de contrôle de l'algorithme est généralement réalisé sous forme d'un automate d'états, particulièrement délicat à mettre en œuvre et à tester. Pour la réalisation physique, les ASICs utilisent principalement de la logique « full-custom » en lieu et place des cellules pré-caractérisées, trop lentes et encombrantes. Le coût de développement en résultant est évidemment très grand, c'est pourquoi les ASICs sont avant tout utilisés soit dans des systèmes destinés à des applications grand public - le grand nombre d'unités produites permettant d'amortir le coût de conception - soit pour des applications dont les besoins en puissance de calcul ne peuvent être satisfaites par aucune des autres solutions.

La solution ASIC a aussi des limites liées à la complexité de l'algorithme à exécuter. Le filtre FIR présenté plus haut se modélise aisément en langage de haut niveau par une simple boucle *for*, et le parallélisme intrinsèque à cet algorithme permet de tirer parti d'architectures matérielles parallèles. Un premier problème survient lorsque l'algorithme n'est plus purement « data-flow » et utilise des instructions de contrôle type *if-then-else* dont le comportement est dépendant des données. Dans ce cas, le parallélisme de calcul est altéré, l'automate de gestion du flot de contrôle devient plus complexe. A plus forte raison, une application complète telle qu'un encodeur *MPEG* (plusieurs centaines de lignes de code C) composé de multiples fonctions, dont certaines très orientées « contrôle », est trop complexe pour une implémentation directe sur une structure de type ASIC. La méthode employée consiste en

général à identifier dans l'application les fonctions les plus exigeantes en puissance de calcul. Chacune de ces fonctions est alors réalisée sous forme d'un cœur d'ASIC, et le reste de l'application, moins critique, est exécuté sur un processeur programmable.

Les structures systoliques basées sur le parallélisme des calculs dans l'espace et dans le temps offre des débits de calculs très importants, mais au détriment d'une latence assez grande. L'exemple du filtre FIR de longueur 4 peut être implémenté sous forme d'une structure systolique totalement parallèle offrant un débit d'un résultat par cycle pour une latence de 4 cycles. Cette solution n'est avantageuse que si l'application requiert de calculer un grand nombre de résultats d'affilée afin de masquer l'effet de latence. Dans le cas extrême où les résultats sont calculés un par un, l'architecture systolique aura la même performance qu'un processeur DSP capable d'exécuter une Multiplication-Accumulation (MAC) par cycle, ce qui minimise son intérêt.

Outre le temps de conception, le grand désavantage de la solution ASIC provient de son manque de flexibilité. Le degré de spécialisation de l'architecture d'un ASIC étant très élevé (utilisation d'unités fonctionnelles ultra-spécialisées, topologie de chemin de données réduite car optimisée), un simple changement dans la spécification de l'application peut signifier la re-conception presque complète du circuit. Dans les applications basées sur les standards, les améliorations apportées aux normes sont régulières (*GSM* Phase I, *GSM* Phase II+, …). Pour cette raison, les concepteurs privilégient dans la mesure du possible des solutions plus souples comme les FPGAs ou les processeurs programmables.

1.4.3 Les FPGAs

L'architecture régulière et configurable des FPGA (*Field Programmable Gate Arrays*) en fait de très bonnes cibles pour l'implantation d'architectures matérielles parallèles accélérant les algorithmes DSP. Un FPGA est ainsi capable, au même titre qu'un ASIC, d'effectuer en un cycle une tache nécessitant plusieurs dizaines ou centaines de cycles sur un processeur programmable. Avantage essentiel par rapport aux ASICs, la fonctionnalité des blocs logiques configurables (CLB) et la topologie du réseau d'interconnexion entre blocs peuvent être reprogrammées, permettant ainsi de faire évoluer l'architecture du circuit selon les besoins (cf. Figure 14). Les densités d'intégration actuelles permettent d'intégrer plusieurs dizaines de milliers de CLBs par circuit, contre quelques dizaines seulement il y'a dix ans. De nombreuses réalisations implémentant des algorithmes très complexes utilisés pour la cryptographie ou la recherche de séquences génétiques ont prouver la puissance et la viabilité des solutions FPGA [11]. Les FPGAs sont aussi souvent utilisés pour le prototypage rapide de systèmes : le comportement du système est décrit en langage de haut niveau (type VHDL) et projeté directement sur un circuit FPGA. Le système est alors émulé par le FPGA, ce qui permet de tester et valider son comportement avant le passage à la réalisation physique sous forme d'ASIC.

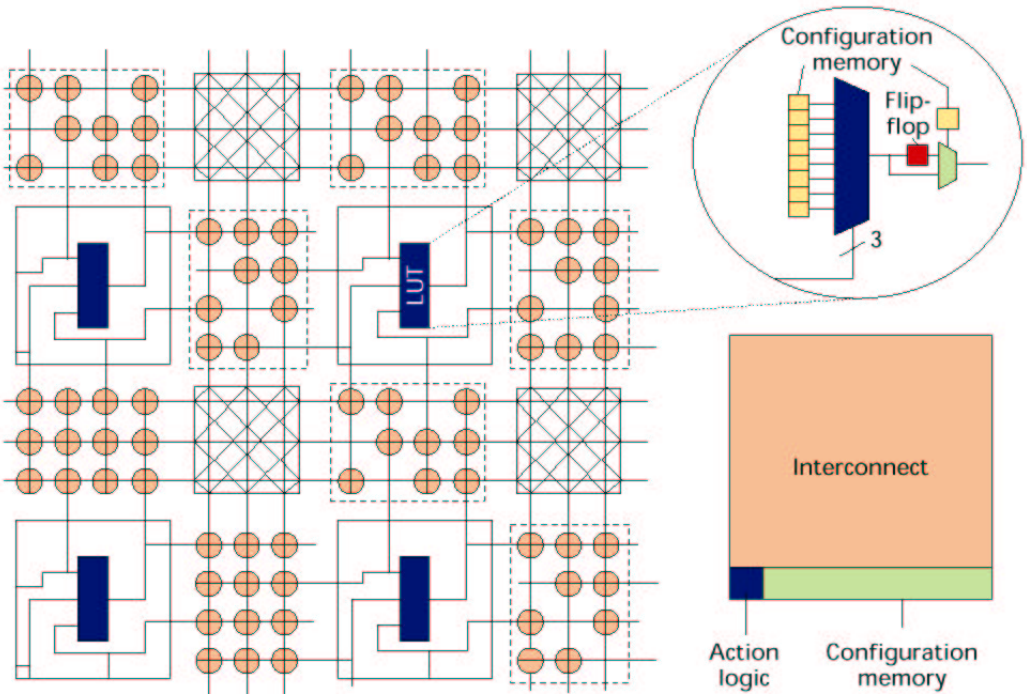


Figure 14 Architecture générale d'un FPGA

L'aspect configurable a évidemment un coût : la surface de silicium et la consommation électrique d'une solution FPGA sont bien supérieures à celles de son équivalent ASIC. Les interconnexions programmables, par exemple, utilisent la majeure partie de la surface d'un FPGA (voir Figure 14), tandis qu'un ASIC utilisera des connexions en nombre limité et optimisées en surface. De plus, l'implantation d'un algorithme sur FPGA n'utilise jamais

100% des capacités du circuit, laissant une partie du matériel inutilisé (mais consommant toujours !). Enfin, les FPGAs souffrent des mêmes limitations que les ASICs : ils sont avant tout efficaces pour des applications de complexité moyenne orientées « flot de données ». Pour palier à cette limitation, une solution consiste à adjoindre au circuit FPGA un cœur de processeur gérant les parties « contrôle » de l'application, comme dans l'architecture GARP [12].

1.4.4 Les Processeurs programmables

Une philosophie différente de celle des ASICs consiste à employer des processeurs programmables pour réaliser les traitements DSP désirés. On passe alors d'une solution totalement « matériel » à un système hybride « matériel/logiciel ». Il existe plusieurs types de processeurs programmables, des plus basiques comme les microcontrôleurs aux plus puissants comme les processeurs superscalaires ou VLIW, en passant par des processeurs spécialisés dits « processeurs DSP ». Cette section présente les principales caractéristiques des différents types de processeurs et leur « aptitude » au traitement du signal.

1.4.4.1 Les processeurs DSP

Comme leur nom l'indique, ces circuits ont pour vocation première l'exécution la plus efficace possible des algorithmes de traitement du signal. Ils diffèrent en ce sens des microprocesseurs du monde PC dont l'usage est beaucoup plus général., de part la diversité même des applications. Les concepteurs des premiers processeurs DSP sortis au début des années 80 se sont directement inspirés de la structure « flot de données » des algorithmes DSP, bien représenté par la structure du filtre FIR présenté Figure 13. Leur architecture est conçue de manière à pouvoir gérer efficacement les boucles de calculs intensives et à assurer une bonne précision numérique pour les données. Pour cela, ils disposent des caractéristiques suivantes :

- opérateurs arithmétiques spécialisés pour les calculs DSP
- grande bande passante pour les données liée à l'utilisation de bus indépendants pour les données et les instructions (Architecture Harvard).
- unités de calculs d'adresses permettant la parallèlisation des calculs et des accès aux données, et proposant des modes d'adressage spécifiques au traitement du signal.
- matériel embarqué permettant l'exécution rapide des boucles logicielles.

A l'heure actuelle, on peut distinguer deux types de processeurs DSPs . Les DSP « conventionnels » sont directement issus de la première génération, leur architecture n'a guère évoluée au fil du temps et l'amélioration de leurs performances tient avant tout aux progrès de la technologie (ex : l'*ADSP-218x d'Analog Devices*). La deuxième génération est apparue il y'a quelques années avec l'apparition du *TMS320C6x* de *Texas Instruments*, le

premier processeur basée sur une architecture VLIW. Ces processeurs, beaucoup plus puissants, intègrent des techniques matérielles évoluées identiques à celles que l'on trouve dans les microprocesseurs modernes. Ils visent avant tout le créneau des applications DSP hautes performances, tandis que les DSP conventionnels se placent sur le secteur du « faible coût / faible consommation ». Les différents types de processeurs DSP sont analysés en détail dans [13].

1.4.4.2 Les microcontrôleurs

Bien que les processeurs de type *Pentium* , *UltraSparc* ou *PowerPC* soient très largement médiatisés de par leurs positions dominantes dans le marché des PC ou des stations de travail, ce sont encore les microcontrôleurs 4 bits et 8 bits qui dominent largement le marché des processeurs du point de vue du volume de vente [23]. Comme leur nom l'indique, le rôle principal de ces circuits est d'effectuer des taches de contrôles. Un exemple typique consiste à gérer des signaux provenant du monde extérieur. Ces signaux peuvent provenir de capteurs analogiques ou numériques et sont connectés aux pattes d'entrées du microcontrôleur. Chaque entrée représente une part d'information sur l'état d'un dispositif externe. Les sorties servent de commandes pour ce dispositif, qui peut être composé de relais, de moteurs, etc. Le rôle du microcontrôleur est d'analyser les entrées et d'affecter les bonnes valeurs en sortie en fonction de l'état courant du système, à la manière d'un automate d'états. Dans les systèmes plus larges, les microcontrôleurs sont aussi utilisés pour la configuration de périphériques et de coprocesseurs, la signalisation et la synchronisation des différents éléments du système.

Le principal rôle des microcontrôleurs est donc de prendre des décisions en fonction de l'état d'un système et de la valeur de ses entrées. Leur architecture est spécialement conçue pour gérer efficacement les branchements conditionnels, les interruptions, les changements de contexte, les tests au niveau bit et les hiérarchies mémoires complexes. Leur capacité en terme de calcul arithmétique est faible : leur chemin de données est généralement composé d'un banc de registres à usage général couplé à une unité arithmétique et logique (ALU) capable d'effectuer les opérations élémentaires que sont l'addition, la soustraction et les fonctions logiques de base (cf. Figure 15). Les performances de ces circuits en traitement du signal sont faibles, pour deux principales raisons :

- Les données et les instructions utilisent le même bus mémoire. La bande passante disponible pour accéder aux données est donc faible : le filtre FIR nécessiterait 2 données pour l' exécution convenable d'une multiplication par cycle, alors que l'architecture Von Neumann (un bus de données et un bus d'adresses) des microcontrôleurs ne peut fournir qu'une donnée tous les 2 cycles.
- Les opérateurs arithmétiques sont peu performants. Les opérations de multiplication largement utilisées en traitement du signal sont soit émulées par logiciel en l'absence d'opérateur câblé spécialisé, soit réalisés sous forme d'opérateurs pipelinés. Dans tous

les cas, une multiplication prend donc plusieurs cycles. Il en va de même pour les opérations de saturation ou d'arrondi, indispensables pour la gestion de la précision lorsqu'on travaille en virgule fixe [14].

- Il n'existe pas de mécanisme matériel de gestion des boucles d'instructions et des modes d'adressage spécifiques aux algorithmes DSP. Les opérations de type « *Test and Branch* » effectuées à chaque fin de boucle sont très coûteuses en nombre de cycles. L'exécution séquentielle des calculs d'adresses et de données, et l'émulation des modes d'adressage spécifiques ajoutent encore à la perte de performance.

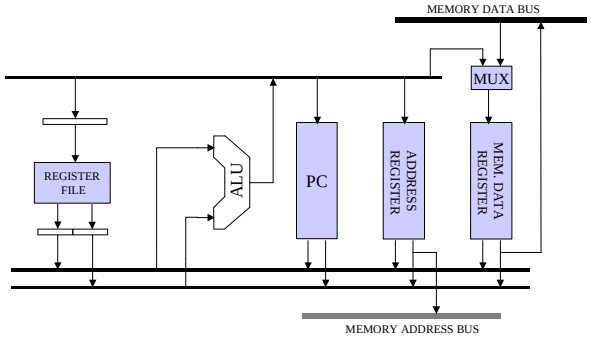


Figure 15 Chemin de données de type microcontrôleur

Portés par le succès des applications embarquées, les microcontrôleurs les plus récents tentent de compenser leur faible aptitude aux traitements DSP en modifiant leur architecture. C'est le cas par exemple du microcontrôleur *ARM7TDMI* : sorti à l'époque des premiers assistants numériques personnels (les fameux *PDAs*), il intègre un multiplieur câblé 16*16 accélérant les traitements numériques dans le but de concurrencer les processeurs DSP. La Figure 23 p.29 compare la performance de l'*ARM7TDMI* avec un processeur DSP faible coût fonctionnant à la même fréquence, *l'ADSP-218x*. La comparaison porte sur un ensemble de noyaux de calculs typiques des applications DSP regroupées en un benchmark nommé BDTIMark [15]. On voit que l'ajout d'un multiplieur seul ne suffit pas pour atteindre la performances des architectures spécialisées. Outre le fait qu'une multiplication prend plusieurs cycles (entre 2 et 5), l'ARM7 souffre de ses caractéristiques « microcontrôleur » : bande passante insuffisante, aucune gestion de la précision, modes d'adressage insuffisants. L'intérêt des microcontrôleurs provenant de leur très faible coût et leur basse consommation, l'ajout de fonctionnalités reste limité et explique la faiblesse de leurs performances par rapport à celles des autres types de processeurs.

1.4.4.3 Les microprocesseurs

Les microprocesseurs à usage général équipant les PC ou les stations de travail actuels (*Pentium*, *UltraSparc*, *PowerPc*, *MIPS Rxxx*, *DEC*) ont des performances autrement plus grandes que celles des microcontrôleurs classiques , leurs performances en traitements DSP pouvant même dépasser celles de certains des processeurs DSP, comme l'illustre la Figure 23

(voir les résultats du Pentium et de l'ADSP-218x). La puissance de ces processeurs provient de plusieurs caractéristiques :

- Fréquence de fonctionnement élevée : ces processeurs sont les moteurs de l'évolution technologique et bénéficient des progrès les plus récents en matière de densité d'intégration. Cela leur permet d'embarquer des millions de transistors sur la même puce tout en fonctionnant à des fréquences beaucoup plus grandes que celles des processeurs classiques ou DSP, ce qui leur confère un net avantage en terme de performance brut.

- Architecture superscalaire : le *Pentium* comme le *PowerPc* reposent sur des architectures dites superscalaires, c'est-à-dire capables d'exécuter plusieurs instructions en parallèle [16]. La technique consiste à regrouper les instructions devant s'exécuter séquentiellement dans une mémoire tampon, à effectuer une analyse dynamique des dépendances entre instructions puis à exécuter en parallèle celles qui ne sont pas interdépendantes. Cette technique permet d'augmenter le parallélisme d'instructions (ILP ou *Instruction Level Parallelism*) et donc d'accélérer significativement l'exécution des applications. Dans le cas du superscalaire, c'est le matériel qui effectue l'analyse dynamique et alloue les instructions aux différentes unités fonctionnelles. Il est à noter que ces processeurs intègrent des unités fonctionnelles arithmétiques performantes équivalentes à celles des processeurs DSPs, telles que des multiplieurs flottants. Ces unités sont parfois dupliquées pour permettre l'exécution des calculs en parallèle. Les architectures superscalaires ont deux avantages par rapport aux architectures VLIW, qui permettent elles aussi d'accroître l'ILP. Le premier est qu'un processeur superscalaire peut exécuter du code conçu pour un processeur séquentiel d'ancienne génération, du fait de la compatibilité ascendante du code binaire. C'est le cas par exemple du Pentium qui exécute indifféremment du code compilé pour un 8086, un i486 ou pour lui-même. Dans le monde PC, c'est une caractéristique cruciale : on imagine mal l'utilisateur de PC devoir changer de logiciels à chaque changement de processeur. Le second avantage est la compacité du code par rapport aux codes VLIW équivalents. La mémoire étant pour une grande part dans le coût des systèmes, c'est là encore un atout non négligeable. La Figure 16 présente un exemple d'architecture superscalaire d'ordre 2 (2 unités d'exécution) capable d'exécuter les instructions dans le désordre. Cette capacité requiert des mécanismes matériels complexes : utilisation d'un tampon de ré-ordonnancement des résultats (*Reorder Buffer*) pour maintenir la cohérence temporel des résultats dans le banc de registres, et des files d'instructions associées à chaque unité d'exécution (*Reservation Stations*, en violet sur la figure) pour permettre aux instructions n'ayant pas encore reçu leurs opérandes de ne pas bloquer les instructions suivantes.

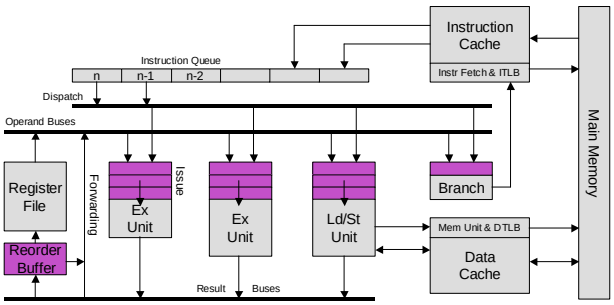


Figure 16 Architecture d'un processeur superscalaire d'ordre 2

- Mécanisme de prédiction de branchement. La complexité des architectures et la fréquence très élevées de fonctionnement de ces processeurs impose un pipeline très profond. Dans le cas d'une instruction de branchement, la cible est connue très tard dans le pipeline : cycle 5 pour le *MIPS R8000*, cycle 11 (!) pour le *Pentium Pro*. Afin de minimiser les pénalités dues aux branchements, ces processeurs utilisent des mécanismes de prédiction de branchement dynamiques [9]. Le principe consiste à garder un historique des derniers passages sur un branchement et de se servir de cet historique pour l'anticiper au passage suivant. On « parie » sur le comportement du branchement et on choisit d'exécuter en avance les instructions suivantes. Cela permet d'augmenter l'ILP et de minimiser le coût global des instructions de branchement. Bien sur, si la prédiction s'avère fausse, une correction doit être effectuée pour restituer l'état correct du processeur et exécuter la bonne branche du saut, opération qui fait perdre de nombreux cycles. Le jeu en vaut cependant la chandelle : si on suppose que la pénalité de branchement d'un processeur est de 10 cycles, et que le taux de bonne prédiction est d'environ 80%, la pénalité moyenne d'un branchement n'est plus que de 2 cycles, d'où une augmentation notable de la performance. L'utilisation ce cette technique requiert un matériel complexe basé sur une table d'historique (Branch History Table) qui mémorise le comportement des derniers branchements (pris ou non-pris) ; certains processeurs disposent d'une autre table mémorisant les adresses cibles des branchements (Branch Target Buffer), afin d'anticiper le chargement des prochaines instructions.

- Caches multi-niveaux : les processeurs modernes intègrent tous des caches données et instructions afin de réduire la latence d'accès entre le CPU et la mémoire. L'architecture superscalaire réclame à la fois une large bande passante afin de pouvoir nourrir les différentes unités fonctionnant en parallèle, et un temps d'accès rapide permettant de fonctionner à la vitesse du processeur. Les technologies mémoires actuelles (*SDRAM*) offrent de bonnes performances en terme de bande passante mais restent désespérément lentes. En effet, au fur et à mesure des années, la différence de vitesse entre les processeurs et les mémoires continue de s'accroître, atteignant un facteur 50 à l'heure actuelle. Le seul remède consiste alors à utiliser des mécanismes

de caches toujours plus complexes. Il est aujourd'hui courant d'utiliser deux niveaux de cache L1 et L2 pour assurer de bonnes performances. Le cache L1 est en général de petite taille, fonctionne à la même vitesse que le processeur et est intégré au coté de celui-ci. Le cache L2, plus grand, plus large et moins rapide, est habituellement situé à l'extérieur, mais son intégration tend à se généraliser.

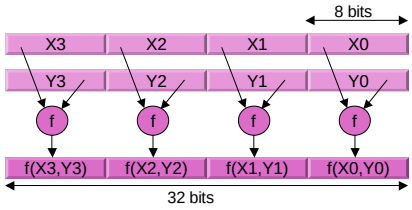


Figure 17 Opération SIMD

- Instructions spécialisées et extensions SIMD. Les applications multimédia utilisent massivement des algorithmes de type traitement du signal, qui exigent de grandes puissances de calcul et sont soumis à des contraintes temps-réel fortes. Tout naturellement, les concepteurs des processeurs généraux ont cherché à améliorer les performances de leurs circuits pour ce type d'applications devenues incontournables. C'est ainsi qu'on a vu apparaître des « extensions » de jeux d'instructions spécialement conçues pour les calculs multimédia : les extensions MMX pour le *Pentium*[17] , AltiVec pour *PowerPc*[18] et VIS *de Sun.* Toutes reposent sur le même constat : les algorithmes DSP et multimédia sont très riches en parallélisme de données, et peuvent donc tirer parti d'instructions de type SIMD (*Single Instruction Multiple Data*). Le principe du calcul SIMD consiste à traiter les données en paquet (sous forme de vecteur) et à appliquer globalement le même traitement à chaque donnée du paquet. La Figure 17 montre l'exemple d'une fonction SIMD prenant 2 opérandes de 32 bits et effectuant simultanément 4 calculs de largeur 8 bits. Une simple instruction SIMD fait donc le même travail que 4 instructions classiques. Certaines instructions réalisent des fonctions non présentes dans le jeu d'instruction de base du processeur, comme par exemple la gestion de l'arithmétique saturée qui prend en compte les problèmes de débordement arithmétique. La dernière extension du MMX intégrée au *Pentium III*, baptisée SSE (*Streaming SIMD Extension)*, gère les calculs vectoriels flottants et propose des mécanismes de contrôle de cache améliorant la gestion des flux de données. Le gain de performance en résultant est notable (cf. Figure 18)[19].

Transform Type	Integer Implementation	MMX	MMX & SSE
2D DCT 8*8	970	280	250
2D iDCT 8*8	970	320	290

Figure 18 Gain de performance MMX / SSE

Ne sont citées ici que quelques unes des fonctionnalités principales intégrées dans les microprocesseurs modernes. On pourrait aussi citer les mécanismes de renommage dynamique de registres, de translation adresses logiques / adresses physiques, d'écriture mémoire retardée, etc.[20] Toutes ces fonctionnalités ont évidemment un coût matériel non négligeable : utilisation de tables d'index, recopie de registres, logique de contrôle complexe, mémoires caches, registres dédiés supplémentaires pour extensions SIMD, etc. Si on ajoute que la gestion correcte des erreurs dans la prédiction de branchement, des interruptions et exceptions complexifie encore le matériel nécessaire, on imagine aisément l'incidence sur l'espace de silicium occupé et la consommation électrique dissipée. Mais les progrès technologiques offrent de telles densités d'intégration (120.000 transistors/mm2 en 0,18 µm) que l'encombrement n'est plus vraiment rédhibitoire, chaque nouvelle génération de procédé de fabrication doublant le nombre de transistors disponible. Quant à la consommation, il n'existe évidemment pas de problèmes d'alimentation électrique pour les PCs ou stations de travail fonctionnant sur le secteur. Seule pose problème la dissipation de chaleur qui doit être endiguée par des systèmes de refroidissement.

De manière générale, l'innovation dans la conception des microprocesseurs a pour l'instant toujours été guidée par la recherche de la performance, les problèmes d'encombrement et de consommation passant au second plan. C'est précisément pour cela que les microprocesseurs de ce type sont peu employés dans les systèmes embarqués, qui ne peuvent s'accommoder de leur consommation excessive, de leur encombrement et du prix prohibitif lié à la débauche de transistors utilisés (cf. Figure 25, [21]).

L'autre grand problème de ces processeurs est qu'on ne peut prévoir précisément leur comportement temporel pour une application donnée. Comme on l'a souligné précédemment, les applications DSP n'ont pas les mêmes contraintes que les autres du point de vue des performances. Pour les applications classiques, les performances requises sont généralement exprimées sous forme d'un temps de réponse moyen maximal : l'exigence ne s'applique pas à chaque exécution, mais seulement sur le comportement global. A l'inverse, les applications DSP ont des contraintes temps-réel très strictes : chaque traitement ne doit jamais dépasser une certaine durée, et ce à chaque instance d'exécution. Le programmeur doit ainsi être capable de prédire combien de temps l'application met à traiter un échantillon, ou du moins quel sera la durée dans le pire des cas.

Ce qui ne semble à priori pas être une exigence insurmontable devient pourtant problématique dès lors qu'on utilise un microprocesseur haute-performance. Ceci est lié aux mécanismes de contrôle dynamiques qu'utilisent ces processeurs. La prédiction de branchement, les mécanismes de caches ou l'exécution dans le désordre font que le comportement du processeur à un instant t dépend de paramètres comme la localité des données dans le cache, le comportement des branchements précédents ou le taux d'occupation des ressources du processeur. Ces paramètres ne sont pas prédictibles, ce qui fait que l'exécution de la même

portion de code peut prendre un nombre très variable de cycles selon ce qui s'est passé auparavant dans le processeur. Et dans ce cas, il devient très difficile de garantir le respect de la contrainte temps-réel.

La solution consistant à considérer l'exécution pire cas, en supposant par exemple que toutes les prédictions de branchement sont fausses et que les accès caches font systématiquement « miss », n'est pas non plus satisfaisante : l'estimation correspondante est très alarmiste et ne correspond pas à ce qui se passe réellement dans la plupart des cas, faisant paraître la solution microprocesseur beaucoup moins performante qu'elle ne l'est en réalité.

C'est pourquoi les architectures matérielles utilisés dans les processeurs DSP utilisent plus volontiers des mécanismes de contrôle statiques rendant la prédiction du temps d'exécution beaucoup plus fiable.

1.4.5 Comparatif

La Figure 19 présente un comparatif qualitatif des solutions matérielles envisageables pour l'implémentation d'un système numérique de traitement du signal. Chaque type de solution offre avantages et inconvénients.

Technologie	Performance	Surface	Consommation	Flexibilité	Temps de conception
ASIC	Excellente	Faible	Faible	Nulle	Très long
FPGA	Très bonne	Grande	Grande	Bonne	Court
Processeur DSP	Bonne	Moyenne	Moyenne	Très Bonne	Moyenne
Microprocesseur / Microcontrôleur	Moyenne / Mauvaise	Moyenne	Moyenne	Très Bonne	Moyenne

Figure 19 Solutions matérielles pour systèmes DSP

1.4.5.1 Matériel ou Logiciel

Du pont de vue de la performance de calcul brut, l'avantage va bien sur aux solutions matérielles parallèles de type ASIC ou FPGA. La Figure 20 présente les résultats de l'implémentation d'un filtre numérique FIR de longueur 8 pour les différents types d'architecture [11]. On y voit que les solutions ASIC et FPGA sont à la fois les plus rapides mais aussi celles qui présentent le plus faible coût *surface * temps*, due à l'excellente *densité de calcul* de ces solutions. Par densité de calcul, on entend le nombre d'opérations réalisables ramené par unité de surface. De ce point de vue, la faiblesse des processeurs programmables est due au fait qu'ils utilisent des mémoires de grandes capacités pour mémoriser le code logiciel. Ces mémoires coûtent cher en terme de surface de silicium, souvent plus cher que le processeur lui-même. A l'inverse, les ASICs ont un contrôle statique câblé en dur et occupant très peu de place. Quant aux FPGAs, la reconfigurabilité est assurée par les bits de configuration des CLBs et des interconnexions , qui forment une seule instruction très large

pour tout le circuit. L'utilisation d'une seule instruction pour mémoriser le comportement du matériel fait que les FPGAs représente de manière beaucoup plus dense l'état et la description d'un calcul donné.

L'autre avantage de ces solutions est qu'elles fonctionnent à une granularité bit, contrairement aux processeurs qui utilisent des mots (16, 32, 64 bits). Un processeur devant traiter des informations de largeur 1 bit utilisera des opérateurs de largeur mot, faisant chuter l'efficacité en terme de densité de calcul.

Architecture	Feature Size (μm)	Area	Time	Area-Time/tap
32-bit RISC	0.75	125 million λ^2	66 ns/cycle X 6+ cycles/tap	50
16-bit DSP	0.65	350 million λ^2	50 ns/tap	17.5
32-bit RISC/DSP	0.25	1.2 billion λ^2	40 ns/tap	46
64-bit RISC	0.18	6.8 billion λ^2	2.3 ns/tap	16
XC4000	0.6	240 CLBs X 1.25 million λ^2 / CLB	14.3 ns / 8 taps	0.54
Altera 8000	0.30	30 LEs X 0.92 million λ^2 / LE	10 ns / tap	0.28
Full Custom	0.75	400 million λ^2	45 ns / 64 taps	0.28
	0.60	140 million λ^2	33 ns / 16 taps	0.28
	0.75	82 million λ^2	50 ns / 10 taps	0.41
Full Custom (fixed coefficient)	0.60	114 million λ^2	6.7 ns / 43 taps	0.018

Figure 20 Performance et coût d'un filtre FIR 8 sur différentes architectures matérielles

La meilleure utilisation du matériel dans les solutions câblées se traduit aussi au niveau de la consommation électrique. Ce paramètre, difficile à estimer pour un circuit complet, est cependant très fortement lié à la surface de silicium. Les FPGAs et les ASICs, plus denses en terme de surface et dont le taux d'utilisation des ressources est beaucoup plus élevé que pour les processeurs, ont donc des rendements performance / consommation bien meilleurs comme l'illustre la Figure 21 [22].

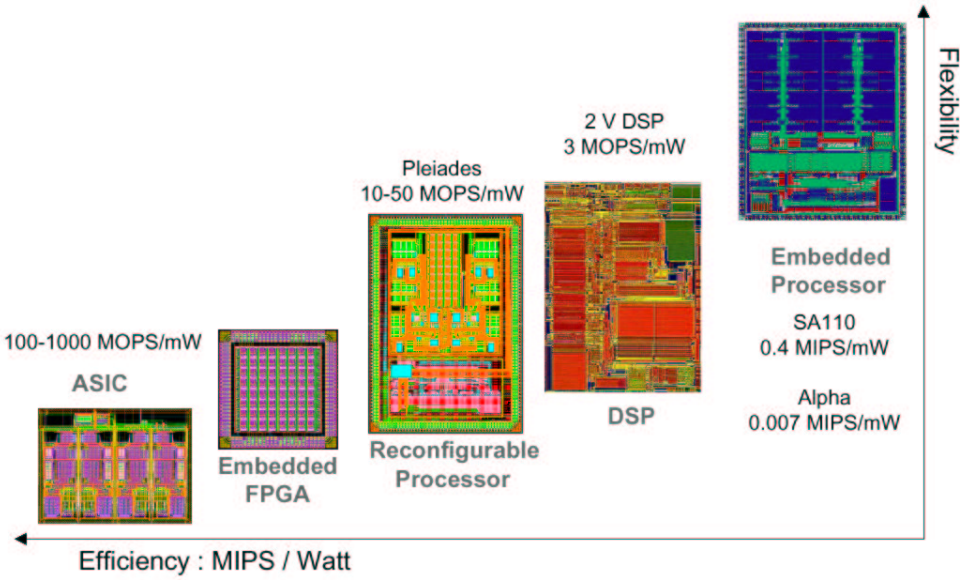


Figure 21 Performance, consommation et flexibilité des différentes architectures matérielles

L'utilisation de processeurs programmables spécialisés en lieu et place des solutions purement câblés est pourtant de plus en plus fréquente, particulièrement dans le domaine des circuits dédiés aux applications DSP [23]. La complexité croissante des applications constitue une première explication. Comme on l'a précédemment fait remarquer, les solutions câblés à structures de contrôle statiques atteignent leurs limites dès lors que les algorithmes s'éloignent du modèle purement « data-flow » en intégrant des structures de contrôle plus complexes. Dans le domaine de l'audio multimédia par exemple, les simples systèmes stéréophoniques utilisés dans les années 80 ont laissé la place à des standards de codage plus performants : *Dolby AC-3*, *MPEG2*, etc. Ces applications sont trop complexes pour pouvoir faire l'objet d'implémentation 100% matérielles, et utilisent en général des systèmes mixtes matériel/logiciel.

Dans le domaine de la téléphonie mobile, les circuits intégrés doivent être désormais capables de supporter plusieurs types de standard, pour pouvoir s'adapter aux normes téléphoniques utilisés dans les différents pays (*GSM*, *CDMA*, …). Chaque norme elle-même peut proposer plusieurs algorithmes différents pour réaliser la même fonction, comme le GSM qui supporte trois types d'encodage de voix différents : Half-Rate, Full-Rate et Enhanced Full Rate. La fusion des téléphones mobiles et des PDAs ajoute encore à la diversité des fonctions que doivent réaliser les circuits. Autant les ASICs ou FPGAs sont adaptés à l'implémentation d'un voire deux algorithmes, pourvu qu'ils ne soient pas trop complexes ni éloignés du point de vue de la fonctionnalité, autant la réalisation de toutes les fonctionnalités du GSM est impensable sur de telles architectures et ne peut se concevoir qu'à l'aide de systèmes programmables.

Le dernier argument penchant en faveur des systèmes programmables est leur flexibilité, qui permet de tenir compte des changements de standards ou de corriger les erreurs de conception.

La tendance à la standardisation est de mise dans le domaine des télécommunications (*GSM, GPRS, UMTS*) comme dans celui du multimédia. Ces normes sont évolutives et régulièrement soumises à changement, soit pour apporter des améliorations par rapport à la version précédente, soit pour corriger certaines erreurs. Ces évolutions sont généralement assez légères et ne remettent pas en cause l'ensemble du système. Dans le cas d'un système programmable, l'évolution consiste donc en une simple mise à jour du logiciel embarqué. Dans le cas du système câblé, le comportement de l'architecture est figé et ne peut être modifié. La prise en compte d'un changement de norme implique donc le changement de tout le système et la re-conception d'un circuit implémentant le nouveau standard. Compte tenu de la complexité et de la durée de conception d'un ASIC, le coût de l'évolution se révèle souvent prohibitif.

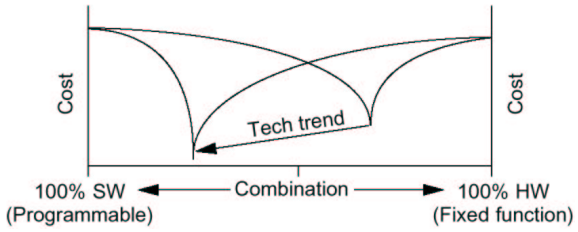


Figure 22 L'importance croissante du logiciel dans les systèmes mixtes

C'est ainsi que de manière générale, la tendance en matière de conception de systèmes intégrés est à l'utilisation de solutions mixtes matérielles/logicielles, dans lesquels la part du logiciel est de plus en plus prépondérante (Figure 22).

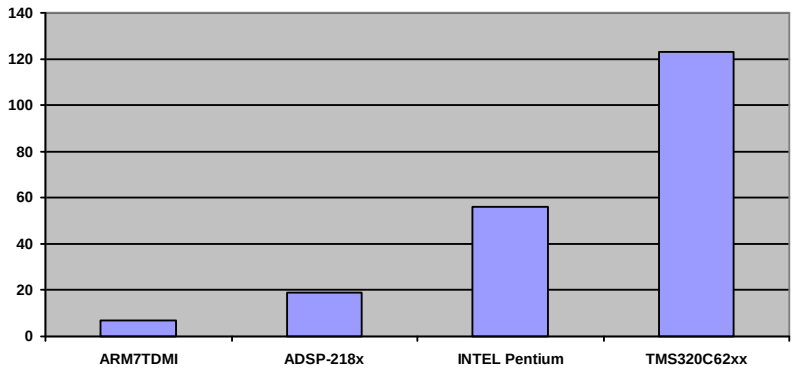


Figure 23 Résultats BDTImark

1.4.5.2 Microcontrôleurs, Microprocesseurs ou DSPs

Du strict point de vue de la performance de calcul, les microprocesseurs généraux n'ont rien à envier aux processeurs DSP (Figure 23), notamment grâce aux techniques matérielles très sophistiquées utilisés dans ces processeurs et absentes dans la plupart des processeurs DSP. A l'heure actuelle, seuls les DSPs évolués de type VLIW peuvent rivaliser en termes de performance.

Pourtant, les DSPs « classiques » intégrant des architectures mono-scalaires (comme l'*ADSP218x*) sont encore les plus nombreux et les plus utilisés en dépit de leurs performances moyennes. Comme on l'a souligné précédemment, les deux défauts rédhibitoires des microprocesseurs généraux du point de vue des systèmes embarqués sont leur coût et leur consommation électrique trop élevés. Quant aux microcontrôleurs, leur performance médiocre et leur faible aptitude aux calculs DSP n'en font pas des alternatives intéressantes pour les applications DSP.

La Figure 24 illustre ces considérations en comparant 4 types de processeurs en termes de performance brut (MIPS), de tension d'alimentation, de puissance moyenne dissipée et de rapport Performance/Consommation : un microcontrôleur RISC faible coût faible consommation (*CoolRisc*), un processeur DSP spécialement conçu pour la basse consommation (*TMS320C54x*), un processeur DSP VLIW principalement utilisé pour ses performances élevées (*TMS320C62x*), et un microprocesseur à usage général (*Dec Alpha*).

Processeur	MIPS	Vdd	Pmoy	MIPS/Watt
CoolRisc	14	3V	2.8mW	5000
TMSC54x	30-200	1.8V	460mW	1500-3000
TMSC6x	1600	2.5V	2W	800
Dec Alpha	500	1.8-3V	50W	7-10

Figure 24 Consommation et performance par type de processeur

La première chose marquante est la consommation « astronomique » du *Dec Alpha* par rapport à ses concurrents. Même le *TMS320C62x*, pourtant l'un des DSPs consommant le plus, dissipe vingt fois moins d'énergie pour une performance deux fois plus grande. Ces chiffres illustrent bien le fait que les microprocesseurs généraux équipant les PCs ou les stations de travail sont conçus avant tout pour la performance au détriment du critère de consommation. La Figure 25 confirme ce constat, et montre qu'il est aussi valable pour le critère du coût.

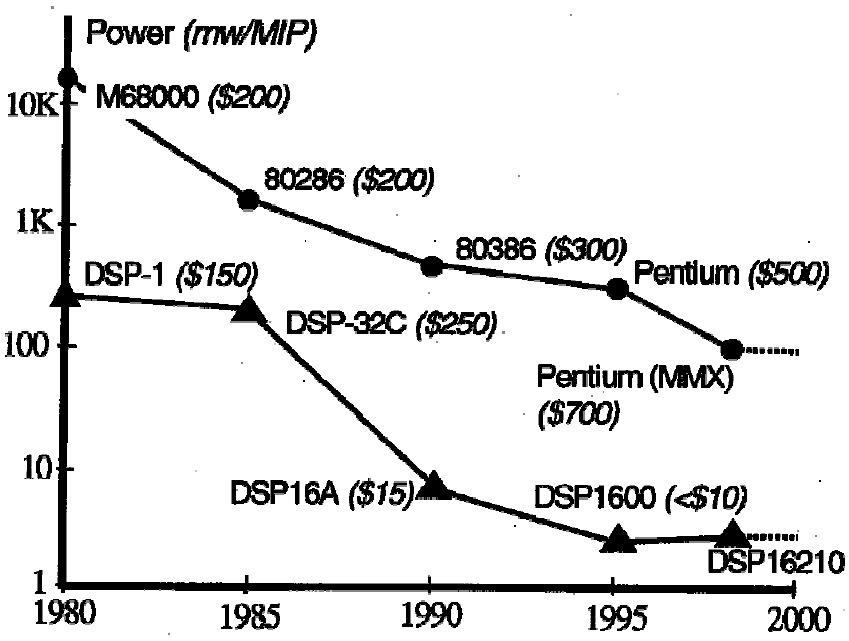


Figure 25 Coût et Consommation des DSPs / Microprocesseurs

La puissance de ces processeurs les rend cependant capables d'exécuter la plupart des applications multimédia fonctionnant sur PC, évitant ainsi le surcoût lié à l'utilisation d'un circuit spécialisé. C'est d'ailleurs la raison d'être des extensions multimédia (*MMX* ou *AltiVec)* intégrées aux processeurs récents. Cependant, pour les applications les plus exigeantes comme les jeux ou les outils graphiques professionnels, ces extensions sont insuffisamment puissantes et les utilisateurs doivent avoir recours à des cartes d'extension intégrant la plupart du temps des processeurs DSP ou des ASICs. On pourrait résumer le problème en disant qu'en terme d'applications DSP, les microprocesseurs d'aujourd'hui résolvent les problèmes d'hier : à l'heure actuelle, un *Pentium* est parfaitement capable de lire un flux audio encodé en *MPEG1* sans surcharger outre mesure le système, mais ne peut soutenir la puissance de calcul requise pour la lecture en temps réel d'une vidéo DVD utilisant un codage *MPEG2*, alors que cette tache peut être effectué à faible coût par un processeur DSP spécialisé.

La flexibilité, le faible coût, les architectures spécialisées en traitement du signal et une consommation électrique raisonnable sont les principaux arguments qui plaident en faveur de l'utilisation de processeurs DSP pour les applications de traitement du signal. Parmi les différentes solutions matérielles envisageables, les processeurs DSPs sont ceux qui présentent le compromis Performance/Consommation/Temps de développement/Coût le plus équilibré [24]. Ils ne sont les meilleurs dans aucune des catégories, mais ne sont mauvais nulle part, et c'est précisément ce qui fait leur succès.

1.5 Exemple d'un système sur puce pour téléphone mobile

La téléphonie mobile est un domaine d'application DSP très populaire et dont les systèmes utilisent plusieurs des architectures matérielles présentées précédemment. Les circuits spécialisés sont soumis aux contraintes liées à tous les systèmes embarqués : coût, encombrement, consommation. Le *56690* de *Motorola* présenté Figure 26 est un circuit dédié à la téléphonie mobile et conçu pour être utilisé dans les terminaux. Il est capable de supporter les différents protocoles mondiaux de communications : *CDMA*, *TDMA*, *GSM*, *iDEN* ou *GPRS*, offrant ainsi la possibilité d'un téléphone fonctionnant partout dans le monde. Il intègre un cœur de DSP (*56600*), un microcontrôleur RISC (*M-Core 210*) et plusieurs périphériques généraux ou spécialisés DSP, dont certains sont réalisés sous forme d'ASIC.

L'architecture hétérogène RISC / DSP / ASIC est très répandue dans le domaine de la téléphonie pour implanter la partie numérique du système; en général, le microcontrôleur gère le protocole, assure l'interface homme-machine et effectue les taches globales de contrôle et de configuration du circuit. Le processeur DSP est responsable de toutes les taches DSP de type « Baseband » (qui suivent la partie RF, la démodulation et la conversion analogique/numérique), constituées principalement des fonctions d'encodage/décodage canal et voix. Des ASICs sont aussi utilisés pour des taches de calcul trop exigeantes et/ou spécialisées ne pouvant être effectuées par le processeur DSP : l'accélérateur de Viterbi (décodage donnée du GSM), l'encryptage GPRS et l'accélérateur AMPS (pour les téléphones analogiques). Ce type d'architecture combinant plusieurs types de systèmes matériels découle directement de la nature très diverse des différents traitements : orientées contrôle , DSP, analogiques.

La technologie actuelle permet désormais d'intégrer des systèmes complets sur une seule puce (concept du « *SoC* » ou *System On Chip*), diminuant ainsi le nombre de composants discrets nécessaires pour une application donnée. Jusqu'à présent, la tendance consistait à utiliser deux composants distincts pour le RISC et le processeur DSP, l'ensemble du système étant implanté sur une carte PCB. Le 56690 propose de ramener ces deux processeurs sur un seul circuit, en y adjoignant de nombreux périphériques généraux ou spécialisés [25]. Le gain par rapport à un système basé sur des composants discrets est multiple :

- diminution du coût: meilleure utilisation du silicium, un seul packaging
- réduction de l'encombrement du système : moins de composants sur la carte embarquée
- réduction de la consommation : les communications entre les différents processeurs et périphériques se font en interne sur le silicium, et évitent d'utiliser les pattes d'entrée-sortie des circuits dont l'activation a une grande influence sur la consommation

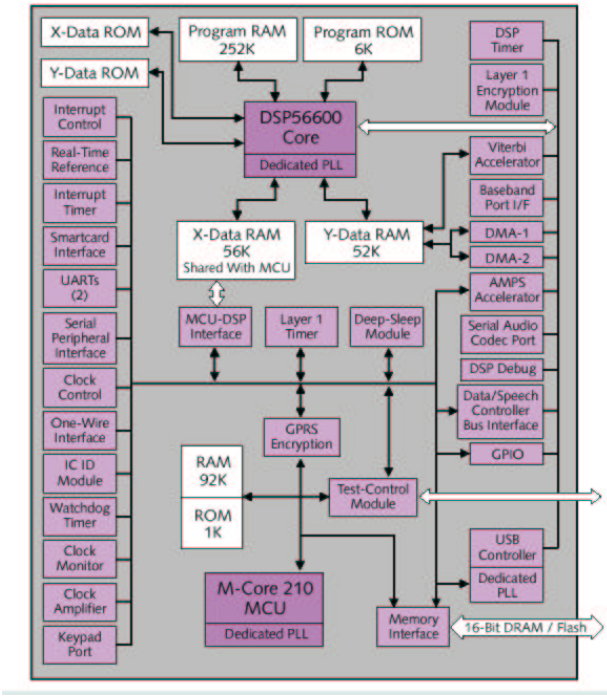


Figure 26 Motorola 56690

Cet exemple illustre les grandes tendances actuelles en matière de circuits DSP pour systèmes embarqués. D'une part, la tendance générale à l'intégration sur une seule puce pousse les concepteurs à concevoir les circuits DSP non plus comme des composants isolés mais comme des cœurs devant s'intégrer dans un système de plus haut niveau. Un effort particulier est porté sur la consommation électrique et l'encombrement qui sont les critères de sélection principaux, le critère de performance étant devenu nécessaire mais plus suffisant. Les processeurs DSP côtoient des blocs ASICs spécialisés implémentant efficacement et à moindre frais (matériel et consommation) des fonctions pour lesquelles les processeurs DSP ne sont pas la meilleure solution, souvent à cause de leur architecture trop générale.

Une alternative à la solution « processeur DSP + coprocesseurs ASIC » non présentée ici consiste à utiliser des processeurs DSP optimisés pour des domaines d'application particuliers. Ces processeurs, baptisés *ASIP* pour « Application-Specific Instruction-Set Processor » ont une architecture différente de celle des processeurs généraux qui leur permet d'accélérer le traitement des noyaux critiques de l'application, éliminant le besoin en circuits ASIC spécialisés. Le concept des ASIPs est présenté en détail en [26].

Bibliographie

1 **WSTS**, *Semiconductor Market Forecast Release*, http://www.wsts.org/

2 **HENNESSY J**., *The Future of System Research*, IEEE Micro, August 1999

3 **FORWARD CONCEPTS**, *DSP Market Bulletin*, Tempe, Arizona, http://www.forwardconcepts.com/

4 **TREZEGUET H., FESTE J.P.**, *Technologies xDSL: pour demain ou après ?* , Electronique, Mars 1999

5 **THOMSEN G., YASHVANT J**., *Internet telephony: going like crazy*, IEEE Spectrum, Mai 2000

6 **SMITH S,** *The Scientist and Engineer's Guide To Digital Signal Processing*, Second Edition, California Technical Publishing, http://www.dspguide.com/

7 **MOORE G**., interview à *Electronics Journal*, avril 1965

8 **SENTIEYS O**., *Estimation et réduction de l'énergie en conception CMOS numérique*, ENSSAT, Université de Rennes I, http://archi.enssat.fr

9 **HENNESSY J., PATTERSON D.**, *Architecture des Ordinateurs, une approche quantitative*, Deuxième édition, traduction de Daniel Etiemble, Thomson Publishing

10 **KUNG H.T.**, *Why Systolic Architectures*, IEEE Computer Magazine, janvier 1982

11 **DEHON A.**, *The Density Advantage of Configurable Computing*, IEEE Computer, Avril 2000

12**CALLAHAN T., HAUSER J., WAWRZYNEK J.**, *The Garp Architecture and C Compiler*, IEEE Computer, Avril 2000

13 **BERKELEY DESIGN TECHNOLOGY INC.**, *The evolution of DSP Processors*, lecture presented to U.C. Berkeley CS152, April 2000, http://www.bdti.com

14 **BALDWIN K., PIEDRA R.M.**, *Overflow Avoidance Methodologies in Commonly Used DSP Functions*, Proc. of the International Conference on Signal Processing Applications and Technologies, Toronto, Canada, 1998

15 **BERKELEY DESIGN TECHNOLOGY INC.**, *BDTIMark Benchmark Results*, http://www.bdti.com/

16 **LAFAGE T., SEZNEC A**., *Evolution des gammes de processeurs MIPS, DEC ALPHA, POWERPC, SPARC, X86 et PA-RISC*, publication interne IRISA, Juin 1997, http://www.irisa.fr/caps

17 **DIEFENDORFF K**., *Katmai Enhances MMX*, Microprocessor Report, October 1998

18 **TREZEGUET H**., *Nouvelle Architecture PowerPc estampillée Multimédia*, Electronique magazine, juin 1998

19 **INTEL Corp.**, *A Fast Precise Implementation of* 8x8 *Discrete Cosine Transform Using the Streaming SIMD Extensions and MMX Instructions*, http://developer.intel.com/software/idap/processor/ia32/mmx/

20 **DIEFENDORFF K**., *PC Processor Microarchitecture*, Microprocessor Report, July 1999

21 **ACKLAND B., NICOL C.**, *High Performance DSPs – What's Hot and What's Not?*, ISLPED98, August 1998, Monterey, CA, USA

22 **SENTIEYS O.**, *Processeurs de traitement du signal : Etat de l'art, Critères de choix, Perspectives*, ENSSAT – Université de Rennes 1, http://archi.enssat.fr

23 **PAULIN P., LIEM C., CORNERO M., NACABAL F., GOOSSENS G.**, *Embedded Software in Real-Time Signal Processing Systems: Application and Architecture Trends*, Proc. of the IEEE, Vol 85, March 1997

24 **RESTLE R**., *Choosing between DSPs, FPGAs, µPs and ASICs to implement digital signal processing*, ICSPAT2000, Dallas, USA, 2000

25 **HALFHILL T**., *Motorola Cellular DSP Does It All*, Microprocessor Report, December 1999

26 **PAULIN P., LIEM C., CORNERO M., NACABAL F., GOOSSENS G.**, *Embedded Software in Real-Time Signal Processing Systems: Application and Architecture Trends*, Proc. of the IEEE, Vol 85, March 1997