



**HAL**  
open science

# Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC, x86 et PA-RISC

André Seznec, Thierry Lafage

► **To cite this version:**

André Seznec, Thierry Lafage. Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC, x86 et PA-RISC. [Rapport de recherche] RR-3188, INRIA. 1997. inria-00073501

**HAL Id: inria-00073501**

**<https://inria.hal.science/inria-00073501>**

Submitted on 24 May 2006

**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.

***Évolution des gammes de processeurs MIPS, DEC Alpha,  
PowerPC, SPARC, x86 et PA-RISC***

André Seznec, Thierry Lafage

**N° 3188**

Juin 1997

THÈME 1



***Rapport  
de recherche***



# Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC, x86 et PA-RISC

André Seznec, Thierry Lafage

Thème 1 — Réseaux et systèmes  
Projet CAPS

Rapport de recherche n° 3188 — Juin 1997 — 153 pages

**Résumé :** Aujourd'hui, les microprocesseurs sont utilisés dans un grand nombre de systèmes matériels : stations de travail, multiprocesseurs, systèmes temps réels, etc. L'évolution est extrêmement rapide dans ce domaine que ce soit au niveau de l'intégration sur le composant ou au niveau de l'architecture, du logiciel ou de la fréquence d'horloge.

Ce rapport reprend le travail de [1] et le complète. Le lecteur trouvera dans ces pages une présentation des architectures de processeur MIPS, DEC Alpha, PowerPC, SPARC, x86 et PA-RISC. Pour chacune d'entre elles, la plupart des microprocesseurs qui en constituent une implantation sont exposés à travers leurs principales caractéristiques. Pour chaque architecture aussi, nous donnons un aperçu de l'avenir en présentant succinctement les futurs microprocesseurs, dans la mesure du possible.

**Mots-clé :** Veille technologique, architecture de microprocesseurs, MIPS, DEC, PowerPC, SPARC, x86, PA-RISC.

*(Abstract: pto)*

Ce travail a été partiellement soutenu par la DRET (convention DRET-INRIA n° 96-2519 A/DRET).

# Evolution of the microprocessors MIPS, DEC Alpha, PowerPC, SPARC, x86, and PA-RISC

**Abstract:** In this report, we try to present in a synthetic form the evolution of different families of microprocessors. MIPS, DEC Alpha, PowerPC, SPARC, x86, and PA-RISC families are considered. We introduce the main architectural characteristics and present the principal actors of their development.

**Key-words:** Technological watch, microprocessor architecture, MIPS, DEC, PowerPC, SPARC, x86, PA-RISC.

## Avertissement au lecteur

*Devant le nombre incommensurable d'acronymes apparaissant aujourd'hui dans les écrits scientifiques et surtout informatiques, nous avons voulu permettre au lecteur une compréhension plus aisée en faisant un index des acronymes (p. 145). Cet index renseigne sur le sens de l'acronyme en fournissant sa forme étendue et en donnant le numéro de la première page qui en contient la référence. Cette référence n'expliquera pas forcément l'acronyme, le but de ce rapport se situant ailleurs (une explication pourra peut-être être trouvée dans le glossaire). Les acronymes jugés peu communs seront quelquefois développés dans une note de bas de page, lors de leur première référence.*



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Les microprocesseurs MIPS</b>	<b>3</b>
1.1 Évolution des microprocesseurs MIPS	3
1.1.1 Jeu d'instructions	4
1.1.2 Pipelines	10
1.1.3 Unités arithmétiques	11
1.1.4 Hiérarchie mémoire	13
1.1.5 Support des systèmes d'exploitation	14
1.1.6 Support multiprocesseur	17
1.1.7 Technologie	18
1.1.8 Remarques	18
1.2 Les microprocesseurs MIPS récents et à venir	19
1.2.1 Le microprocesseur MIPS R5000	19
1.2.2 Le microprocesseur RM7000	22
1.2.3 Comparaison des deux microprocesseurs	23
1.2.4 La famille RM52xx	23
1.3 Développement de systèmes à base de microprocesseurs MIPS	25
1.3.1 Rappels sur les microprocesseurs dérivés des MIPS R3000 et R4000	26
1.3.2 Les microprocesseurs dérivés du MIPS R3000	26
1.3.3 Les microprocesseurs dérivés du MIPS R4000	26
1.3.4 Microprocesseurs MIPS divers	29
<b>2 Les microprocesseurs Alpha</b>	<b>31</b>
2.1 L'architecture Alpha	31
2.1.1 Caractéristiques générales	32
2.1.2 Modifications apportées depuis fin 1995	32
2.2 Rappels et mises à jour sur les microprocesseurs DEC 21064, 21164 et dérivés	34
2.2.1 Jeu d'instructions	35
2.2.2 Architecture: vue générale	35
2.2.3 Séquencement et exécution des instructions	37
2.2.4 Unités arithmétiques	38
2.2.5 Hiérarchie mémoire	38
2.2.6 Support des systèmes d'exploitation	39
2.2.7 Support multiprocesseurs	41
2.2.8 Mesures de performances	41
2.2.9 Technologie	41
2.2.10 Dérivés	41
2.3 Le microprocesseur DEC 21264	42
2.3.1 Jeu d'instructions	43

2.3.2	Architecture: vue générale . . . . .	43
2.3.3	Séquencement et exécution des instructions . . . . .	43
2.3.4	Détail des unités fonctionnelles de calcul . . . . .	45
2.3.5	Hierarchie mémoire . . . . .	46
2.3.6	Bus système . . . . .	47
2.3.7	Support des systèmes d'exploitation . . . . .	47
2.3.8	Support multiprocesseurs . . . . .	48
2.4	Perspectives . . . . .	48
<b>3</b>	<b>Les architectures POWER et PowerPC</b>	<b>49</b>
3.1	L'architecture POWER . . . . .	49
3.1.1	Le jeu d'instructions POWER . . . . .	50
3.1.2	L'architecture POWER . . . . .	51
3.2	L'architecture PowerPC . . . . .	56
3.2.1	Caractéristiques de l'architecture PowerPC . . . . .	56
3.2.2	Les microprocesseurs d'architecture PowerPC . . . . .	58
3.2.3	Les micro-contrôleurs d'architecture PowerPC . . . . .	63
3.2.4	Perspectives pour l'architecture PowerPC . . . . .	66
<b>4</b>	<b>Les microprocesseurs SPARC</b>	<b>69</b>
4.1	Architecture SPARC . . . . .	69
4.1.1	Généralités . . . . .	70
4.1.2	La norme SPARC-V9 . . . . .	70
4.1.3	Extension multimédia . . . . .	73
4.2	Le SuperSPARC . . . . .	74
4.2.1	Le SuperSPARC-II . . . . .	75
4.3	Le MicroSPARC . . . . .	75
4.3.1	Les autres MicroSPARC . . . . .	76
4.4	L'HyperSPARC . . . . .	76
4.5	L'UltraSPARC . . . . .	77
4.5.1	L'UltraSPARC-II . . . . .	79
4.5.2	L'UltraSPARC-IIi . . . . .	79
4.6	Le SPARC64 . . . . .	80
4.7	Le TurboSPARC . . . . .	81
4.8	Les microprocesseurs SPARC embarqués . . . . .	83
4.8.1	La famille SPARC <i>lite</i> 930 . . . . .	83
4.8.2	La famille SPARC <i>let</i> . . . . .	84
<b>5</b>	<b>Les microprocesseurs x86</b>	<b>87</b>
5.1	L'architecture x86 . . . . .	87
5.1.1	Jeu d'instructions . . . . .	87
5.1.2	Modes de fonctionnement . . . . .	91
5.1.3	Registres . . . . .	91
5.1.4	Organisation de la mémoire . . . . .	91
5.2	Le Pentium et les compatibles Pentium . . . . .	92
5.2.1	La famille Pentium d'Intel . . . . .	92
5.2.2	Le 6x86 (M1) de Cyrix . . . . .	96
5.2.3	Le K5 d'AMD . . . . .	98
5.2.4	Le Nx586 de NexGen . . . . .	100
5.2.5	Le MediaGx de Cyrix . . . . .	100

5.3	Le PentiumPro et les compatibles PentiumPro . . . . .	102
5.3.1	Le PentiumPro d'Intel . . . . .	102
5.3.2	Le M2 de Cyrix . . . . .	105
5.3.3	Le K6 d'AMD . . . . .	107
5.4	Récapitulatif sur les microprocesseurs d'architecture x86 . . . . .	111
5.5	Les microprocesseurs x86 à venir . . . . .	111
5.5.1	Le Pentium-II (Klamath) d'Intel . . . . .	111
5.5.2	Autres . . . . .	113
<b>6</b>	<b>L'architecture PA-RISC</b>	<b>115</b>
6.1	Description de l'architecture . . . . .	115
6.1.1	Version 1.1 . . . . .	116
6.1.2	Version 2.0 . . . . .	120
6.2	Les microprocesseurs d'architecture PA-RISC . . . . .	124
6.2.1	Les microprocesseurs d'architecture PA-RISC 1.1 . . . . .	124
6.2.2	Le microprocesseur d'architecture PA-RISC 2.0 (PA-8000) . . . . .	134
6.3	Microprocesseurs à venir . . . . .	138
6.3.1	Le PA-8200 . . . . .	138
6.3.2	Le PA-8500 . . . . .	138
	<b>Annexe : caractéristiques principales des microprocesseurs hautes performances</b>	<b>141</b>
	<b>Bibliographie</b>	<b>143</b>
	<b>Index des acronymes</b>	<b>145</b>
	<b>Glossaire</b>	<b>149</b>

# Table des figures

1.1	Vue générale du MIPS R10000. . . . .	5
1.2	Utilisation de <i>ALNV.PS</i> avec <i>LUXC1</i> du jeu d'instructions MIPS V. . . . .	7
1.3	Différents types de pipeline selon les unités fonctionnelles dans le MIPS R10000. . . . .	11
1.4	Vue générale du MIPS R5000. . . . .	20
2.1	Architecture du DEC 21064 . . . . .	35
2.2	Architecture du DEC 21164 . . . . .	36
2.3	Schéma du DEC 21264. . . . .	44
2.4	Les pipelines du DEC 21264. . . . .	44
2.5	Interfaçage du DEC 21264 avec le cache secondaire et le reste du système. . . . .	47
3.1	Le processeur P2SC d'IBM : synoptique. . . . .	52
3.2	Relation entre les jeux d'instructions POWER et PowerPC. . . . .	57
4.1	Fenêtres de registres sur l'architecture SPARC-V9. . . . .	72
4.2	Les composants de l'HyperSPARC. . . . .	76
4.3	Synoptique du TurboSPARC. . . . .	81
4.4	Le pipeline du TurboSPARC. . . . .	82
4.5	Mécanisme de gestion des branchements dans le TurboSPARC. . . . .	82
5.1	Formats des instructions de l'architecture x86. . . . .	88
5.2	Synoptique du P55C (Pentium avec MMX). . . . .	95
5.3	Les pipelines du P55C. . . . .	96
5.4	Pipelines du microprocesseur <i>Cyrix</i> 6x86. . . . .	97
5.5	Synoptique du MediaGX de Cyrix. . . . .	101
5.6	Le pipeline du MediaGX. . . . .	101
5.7	Architecture du PentiumPro. . . . .	103
5.8	Pipelines du PentiumPro. . . . .	104
5.9	Synoptique du M2 de Cyrix. . . . .	106
5.10	Synoptique du K6 d'AMD. . . . .	108
5.11	Les pipelines du K6. . . . .	109
6.1	Décimal empacoté de l'architecture PA-RISC. . . . .	116
6.2	Adresses physiques de la mémoire et des entrées/sorties. . . . .	118
6.3	Adresse virtuelle de l'architecture PA-RISC 1.1. . . . .	119
6.4	Modèle d'adressage virtuel de PA-RISC 2.0. . . . .	121
6.5	Moyenne parallèle sur les demi-mots dans l'extension MAX-2. . . . .	121
6.6	Le cœur du « Snake ». . . . .	125
6.7	Le pipeline du cœur du « Snake ». . . . .	125
6.8	Le pipeline du PA-7100. . . . .	127
6.9	Configuration uniprocasseur avec le PA-7100. . . . .	129

6.10	Configuration à deux PA-7100. . . . .	129
6.11	Configuration haut de gamme à plusieurs PA-7100. . . . .	129
6.12	Synoptique du PA-7100LC . . . . .	130
6.13	Configuration multiprocesseurs à base de PA-7200. . . . .	133
6.14	Le PA-7300LC : synoptique. . . . .	133
6.15	Architecture du PA-8000. . . . .	135
6.16	Les pipelines du PA-8000. . . . .	136
6.2	Prédiction de branchement à deux bits selon l'algorithme de Smith . . . . .	152
6.3	Diagramme d'états finis du protocole MESI. . . . .	152

# Liste des tableaux

1.1	Modifications du jeu d'instruction apportées par MIPS V. . . . .	7
1.2	Évolution du jeu d'instructions MIPS. . . . .	9
1.3	Taille des pages des microprocesseurs MIPS. . . . .	15
1.4	Évolution de la technologie des microprocesseurs MIPS. . . . .	18
1.5	Latences et débits (en cycles) du MIPS R5000 et du R4600. . . . .	20
1.6	Caractéristiques principales des microprocesseurs MIPS R5000 et RM7000. . . . .	24
1.7	Caractéristiques des principaux microprocesseurs embarqués à base d'architecture MIPS depuis 1992 . . . . .	27
1.8	Les principales différences dans les microprocesseurs CW4001, 4002, et 4003. . . . .	29
1.9	Les principales différences dans les microprocesseurs CW4010, 4011, et 4020. . . . .	30
2.1	Extension multimédia au jeu d'instructions Alpha. . . . .	34
2.2	Caractéristiques technologiques des microprocesseurs DEC Alpha. . . . .	41
3.1	Caractéristiques technologiques des microprocesseurs POWER. . . . .	56
3.2	Récapitulatif des données technologiques des microprocesseurs PowerPC. . . . .	64
3.3	Les principales différences entre les micro-contrôleurs PowerPC 403. . . . .	66
4.1	Formats et types de données de la norme SPARC-V9 . . . . .	70
4.2	Différences entre les microprocesseurs de la famille SPARClite 930. . . . .	83
5.1	Les instructions de l'extension multimédia MMX. . . . .	90
5.2	Latences et débits d'exécution des unités du K6 en nombre de cycles. . . . .	110
5.3	Les microprocesseurs d'architecture x86 actuels (1996-97). . . . .	112
6.1	Les types de données de l'architecture PA-RISC. . . . .	116
6.2	Les instructions de l'extension MAX-2. . . . .	122
6.3	Blocages du pipeline du PA-7100. . . . .	127
6.4	Latences et débits en nombre de cycles des opération flottantes du PA-7100. . . . .	128
6.5	Exécution superscalaire de degré deux sur le PA-7100LC. . . . .	131
6.6	Caractéristiques principales des microprocesseurs hautes performances. . . . .	141

# Introduction

Le rapport de veille technologique présente la plupart des microprocesseurs de quelques architectures, à savoir MIPS, DEC Alpha, POWER et PowerPC, SPARC, x86, et PA-RISC. Elle fait suite à [1] et en constitue une mise à jour. Cependant, ce rapport peut être consulté indépendamment de son prédécesseur puisque les principales informations s'y retrouvent. Un retour à [1] ne sera nécessaire que pour chercher de plus amples détails sur certains produits ou spécifications plus anciennes ; dans ces cas-là, le lecteur trouvera une référence explicite. Pour une étude détaillée de certains processeurs de même génération, le lecteur pourra se référer à [2], [3], [4], [5] et [6].

Cette étude, tout comme la précédente, ne se veut pas exhaustive en ce qui concerne les architectures abordées et les microprocesseurs de chaque architecture, mais nous avons tâché de faire en sorte que les unes et les autres soient représentatifs. Ainsi apparaît l'évolution des techniques employées depuis les premiers jusqu'aux plus récents microprocesseurs de chaque architecture.

Cette approche a aussi pour but de rendre compte des tendances du marché des microprocesseurs et des acteurs qui y prennent part à travers les différentes alliances formées entre les concepteurs, les fondeurs, les intégrateurs de systèmes, etc.

Les six chapitres qui constituent le corps de ce rapport peuvent être abordés indépendamment les uns des autres. Ils référencent souvent des notions techniques que le lecteur averti connaîtra, mais dont le novice pourra trouver quelque explication dans le glossaire (p. 149). Attention toutefois, les données technologiques (intégration, fréquence d'horloge, etc.) évoluent très rapidement et sont susceptibles de changements ; les propos tenus ici ne sont que le reflet des informations que nous en avons au mois de mai 1997. Ainsi, ce rapport sera plutôt considéré comme une photographie à un instant précis de l'état de l'art de chaque architecture et de chaque microprocesseur en constituant une implantation.

Les informations présentées dans ce document ont été principalement extraites de la revue *Microprocessor Report - The Insider's Guide To Microprocessor Hardware*, des manuels de références de chaque architecture et des divers sites *World Wide Web* relatifs aux concepteurs et/ou fondeurs des microprocesseurs étudiées. La plupart de la littérature utilisée ici est en langue anglaise, et pour cela nous prions le lecteur de bien vouloir excuser l'utilisation (plus commode pour nous) de barbarismes (tels « renommage » de registres, unité « entière », microprocesseur « pipeliné », etc.) dont la compréhension ne pose heureusement pas de problème.



# Chapitre 1

## Les microprocesseurs MIPS

*MIPS Technology (MTI)* est une filiale de la société *Silicon Graphics* (depuis 1992). Les microprocesseurs *MIPS* sont utilisés par plusieurs vendeurs de station de travail et de PC. Mais, sur ce marché, leur principal utilisateur est la maison mère *Silicon Graphics*.

Le succès des microprocesseurs *MIPS* est également dû au marché des systèmes embarqués (*set-top boxes*, consoles de jeux, *PDA*<sup>1</sup>). La croissance de ce secteur et les enjeux économiques qu'il représente laissent augurer un avenir florissant pour MIPS.

Nous présentons dans un premier temps les microprocesseurs qui ont fait le succès de *MIPS* dans le domaine des stations de travail, à savoir les microprocesseurs MIPS R3000, MIPS R4000, MIPS R8000 et MIPS R10000. Dans la mesure où ces microprocesseurs ont déjà été le sujet de nos précédentes études [2, 3, 5, 6], nous n'entrons pas dans les détails de ceux-ci mais nous présentons les principales différences d'une génération à l'autre.

Puis, nous étudions brièvement les microprocesseurs les plus récents, le MIPS R5000 et le MIPS RM7000. Ils ne constituent pas une nouvelle génération de microprocesseurs dans la mesure où les techniques qu'ils implémentent ne sont pas nouvelles, mais seulement des versions différentes destinées à des marchés particuliers tels que les systèmes embarqués haut de gamme ou les stations de travail d'entrée de gamme. Et c'est pour cela que nous ne les incluons pas dans la partie précédente.

Enfin, nous étudions quelques uns des microprocesseurs dérivés des architectures MIPS R3000 et MIPS R4000 (ce sont actuellement les seuls microprocesseurs à avoir des descendants). Cette partie nous permettra d'introduire la stratégie commerciale de *MIPS* à travers ses partenaires fondateurs. En effet, *MIPS* a essentiellement une activité de développement de microprocesseurs, alors que la fonderie est assurée par des partenaires extérieurs. L'ensemble de ces fondateurs a d'autre part une activité d'innovation qui contribue au développement de systèmes à base de microprocesseurs *MIPS* et qui accroît ainsi la pénétration de cette architecture sur le marché des systèmes embarqués.

### 1.1 Évolution des microprocesseurs MIPS

En 1985, *MIPS Computer Systems* lance son premier microprocesseur, le **MIPS R2000**. Ce microprocesseur dérivé des travaux de l'université de Stanford est un des premiers microprocesseurs RISC (le premier microprocesseur VLSI selon *MIPS*). Comme il n'est plus commercialisé depuis longtemps, nous n'en parlerons pas dans les parties suivantes.

Annoncé en 1988, le **MIPS R3000** fut le premier microprocesseur développé par *MIPS* à vocation industrielle. L'une des principales caractéristiques de ce microprocesseur est sa simplicité. En effet, la force des microprocesseurs MIPS réside dans la relation très étroite entre l'architecture du microprocesseur et le compilateur associé<sup>2</sup>.

---

1. *Personal Digital Assistant*.

2. MIPS signifie *Microprocessor without Interlocked Pipeline Stages*.

En 1991, *MIPS* commercialise le **MIPS R6000** à la fréquence de 66 MHz. Développé par *Bipolar Integrated Technology*, ce microprocesseur reprend pour l'essentiel l'architecture du R3000 en introduisant certaines particularités telles qu'un cache secondaire externe, un identificateur de processus dans les caches primaires et dans le cache secondaire, une prédiction de branchement, une table de traduction d'adresses séparée de l'unité entière et implémentée sur le cache secondaire, etc. La technologie *ECL* a été utilisé pour fabriquer ce microprocesseur. Bien que cette technologie soit potentiellement plus rapide que la technologie CMOS, les avantages de cette dernière en matière d'intégration, de consommation et de coût de revient l'imposèrent comme la plus utilisée sur le marché. Le MIPS R6000 reste à ce jour le microprocesseur le plus connu réalisé en *ECL*, mais ne connut pas le succès escompté ; il ne sera pas décrit plus avant dans ce rapport.

Le **MIPS R4000** constitue la troisième génération de microprocesseurs MIPS (1992). Il est le premier à mettre en œuvre une architecture 64 bits qui reste cependant compatible avec les générations précédentes (un mode 32 bits assure la compatibilité). Afin d'offrir une cadence de séquençement très élevée à l'époque (dès sa sortie, le R4000 est annoncé à la fréquence de 100 MHz), la technique dite du superpipeline a été utilisée. Destiné au marché des stations de travail, ce microprocesseur fut initialement prévu en trois versions : R4000PC (avec cache primaire seulement), R4000SC (avec cache secondaire) et R4000MC (version destinée aux systèmes multiprocesseurs).

En juin 1994, *MIPS* annonçait le **MIPS R8000**. De par ses performances, ce microprocesseur visait clairement le marché des applications scientifiques et techniques (stations de travail graphiques 3D), les serveurs de base de données, ainsi que le domaine des multiprocesseurs. Ce microprocesseur est le premier à mettre en œuvre une architecture superscalaire. Jusqu'à quatre instructions peuvent être séquencées en parallèle à chaque cycle grâce à ses deux unités flottantes, ses deux unités entières et ses deux unités de lecture/écriture. Par ailleurs ce microprocesseur met en œuvre des mécanismes originaux afin d'accroître ses performances. Le MIPS R8000 est commercialisé à la fréquence de 75 MHz.

Fin 1995, le **MIPS R10000** (FIG. 1.1) voit le jour. Ce microprocesseur va remplacer le MIPS R8000 sur le marché des stations de travail haut de gamme. Il met en œuvre une architecture superscalaire découplée et peut séquencer jusqu'à six instructions en parallèle. Le MIPS R10000 est à ce jour le plus puissant de la gamme des microprocesseurs développés par *MIPS*<sup>3</sup>, il a été annoncé à une fréquence de 275 MHz. Cependant, il ne semble être disponible qu'à 200 MHz.

Nous décrivons dans les paragraphes suivants les principales caractéristiques de ces microprocesseurs en comparant leur jeu d'instructions, leurs pipelines, la structure des caches et les mécanismes de gestion de la mémoire. Nous attacherons plus d'importance aux deux microprocesseurs les plus récents parmi ceux cités ci-avant (MIPS R8000, R10000) car ils mettent en œuvre des mécanismes architecturaux relativement complexes. Le **MIPS R5000** qui a été annoncé début 1996 est plus récent que le R10000, mais il ne comporte aucun concept architectural nouveau, c'est pour cela que nous n'en parlerons pas dans cette partie. Il sera cependant introduit plus loin, en partie 1.2.1 (p. 19). Le **RM7000** n'est pas conçu par *MIPS Technologies*, mais par *QED*. Il est cependant d'architecture MIPS et est le dernier microprocesseur de la famille à avoir été conçu. De même que pour le R5000, nous le détaillons dans une autre partie (section 1.2.2, p. 22).

### 1.1.1 Jeu d'instructions

Le jeu d'instructions MIPS est un jeu d'instructions RISC (voir glossaire p. 153) et depuis la sortie du premier microprocesseur MIPS, il a beaucoup évolué. Il en est aujourd'hui à sa cinquième version. Notons que chaque mise à jour du jeu d'instructions a consisté en un sur-ensemble du précédent.

Nous présentons dans une première partie les concepts généraux de ce jeu d'instructions avant de détailler son évolution.

---

3. Récemment (mai 1997), le MIPS R12000 a été annoncé ; c'est en fait un R10000 avec quelques améliorations ponctuelles pour en augmenter les performances.

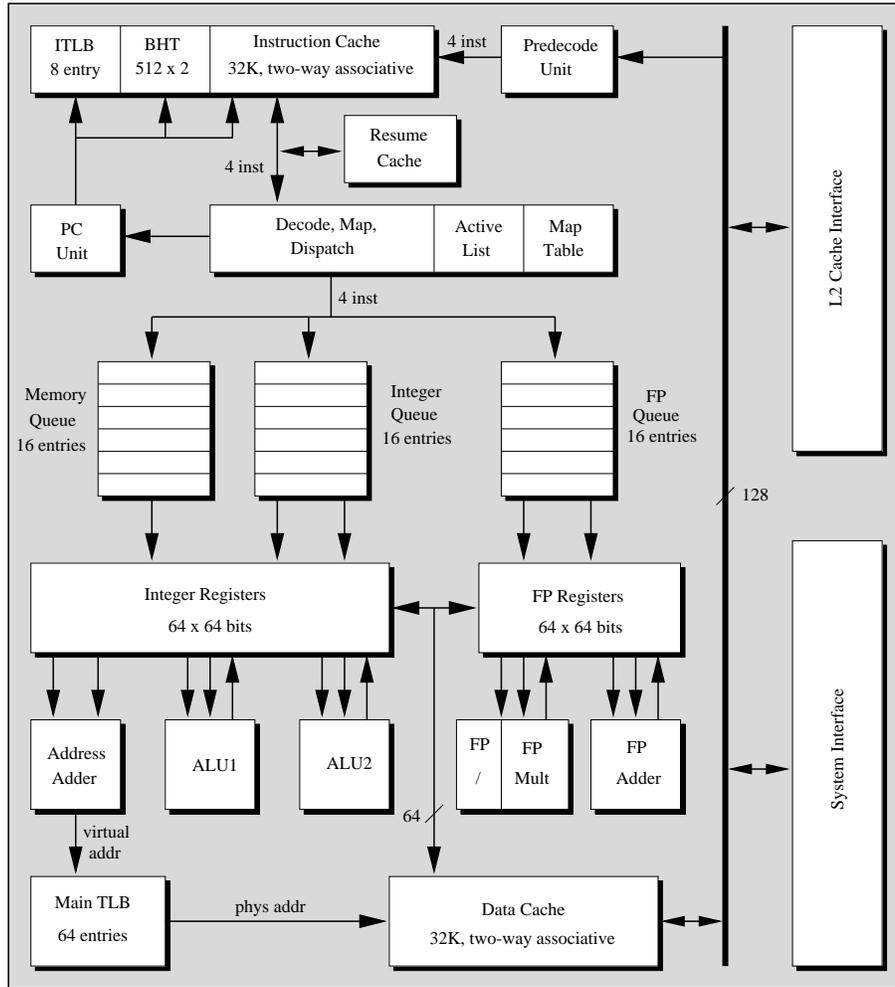


FIG. 1.1 – Vue générale du MIPS R10000.

## Jeu d'instructions d'origine

MIPS définit cinq classes d'instructions qui sont :

- Les instructions de lecture/écriture. Ces instructions concernent le transfert de données entre la mémoire et les registres généraux (entiers ou flottants). Aucune opération d'accès mémoire autre que des instructions de lecture et d'écriture n'est permise par le jeu d'instructions (architecture *load/store*).
- Les instructions arithmétiques. Ces instructions concernent l'ensemble des calculs, les opérations logiques et de décalages.
- Les instructions de saut et de branchement conditionnel. Ces instructions modifient le contrôle de flot du programme. Les sauts peuvent être soit absolus (l'adresse est alors codée sur 26 bits), soit indirects (l'adresse est alors contenue dans un registre et a donc une largeur de 32 bits). Les instructions de branchement conditionnel permettent des déplacements adressés sur 16 bits par rapport au compteur de programme (PC).
- Les instructions des coprocesseurs. L'architecture MIPS définit quatre coprocesseurs (*CP0*, . . . , *CP3*). Le coprocesseur *CP1* désigne l'unité flottante, alors que les *coprocesseurs 2* et *3* sont réservés pour de futures définitions de l'architecture MIPS. Le *coprocesseur 0* est chargé de la gestion du système de mémoire virtuelle, des exceptions ainsi que des transitions entre les modes *noyau*, *superviseur* (R4000, R10000) et *utilisateur*. Cette unité contrôle également les caches et fournit des diagnostics de contrôles d'erreur, etc. Elle n'est accessible qu'en mode *superviseur*. Les instructions de calcul et de transfert sont spécifiques aux coprocesseurs.
- Les instructions dites spéciales. Ces instructions concernent des tâches diverses de transfert de données entre registres spécialisés et registres généraux, des exceptions et des points d'arrêt.

Toutes les instructions sont codées sur 32 bits. Trois principaux formats d'instructions sont utilisés : immédiat (*I-type*), saut (*J-type*) et registre (*R-type*) (voir [2, 3, 5] pour plus de détails).

## Évolution du jeu d'instructions MIPS

Le jeu d'instructions MIPS a connu quatre mises à jour, chacune correspondant à un agrandissement de l'ensemble des instructions afin d'assurer une totale compatibilité binaire avec les versions plus anciennes. Nous ne détaillerons pas chaque version du jeu d'instruction MIPS — le lecteur en trouvera une synthèse dans le tableau 1.2 (p. 9) et il pourra consulter [7] pour une spécification complète du jeu d'instructions MIPS IV avec son évolution depuis MIPS I — pour nous concentrer sur la dernière version, récemment annoncée, qui offre des particularités intéressantes, ainsi que sur les extensions optionnelles récentes au jeu d'instructions (résumées elles aussi dans le tableau 1.2).

### MIPS V

Annoncée en octobre 1996, la nouvelle version du jeu d'instruction MIPS, MIPS V, vise à améliorer les performances des calculs sur les graphiques géométriques 3D et du calcul scientifique en général, en permettant la parallélisation des opérations sur les réels simple précision.

Pour ce faire *MIPS* a défini un nouveau format de données flottantes : le format *paired-single (PS)*. Ce format permet de stocker deux opérands flottants simple précision dans un seul registre flottant double précision. MIPS V inclut aussi la modification de certaines instructions pour qu'elles puissent supporter ce format de données, ainsi que de nouvelles instructions qui permettent de le manipuler (TAB. 1.1).

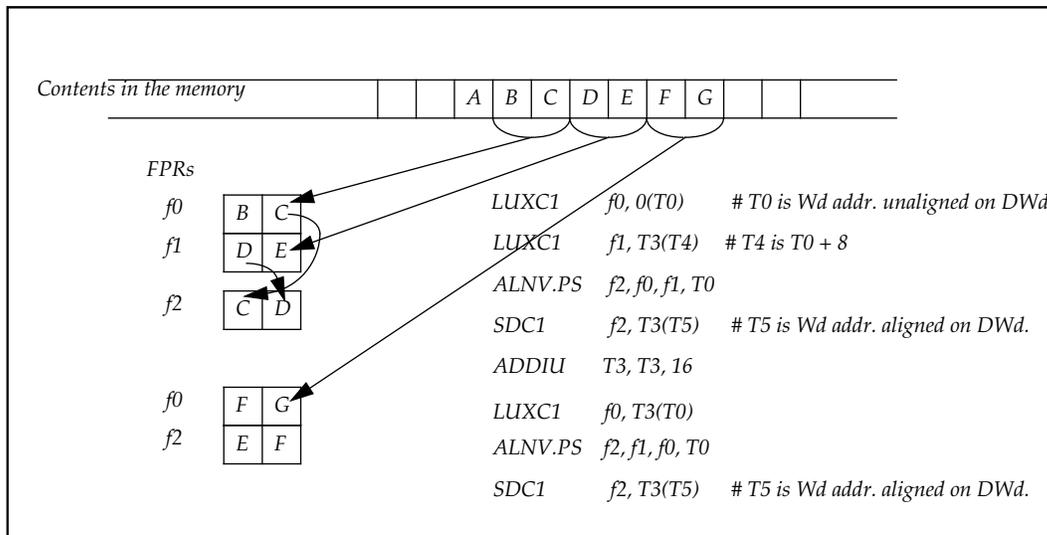
L'instruction *C.xxx* permet de comparer les deux valeurs contenues dans une donnée PS ; elle met à jour les bons codes condition. Les instructions *MOVx* ont été modifiées pour pouvoir faire des déplacements de valeurs de type PS de registre à registre selon la valeur (vraie ou fausse) d'un code condition.

Instructions modifiées	
ADD, SUB, MUL, ABS, MOV, NEG	Opérations de calcul arithmétique de base.
MADD, MSUB, NMADD, NM-SUB	Instructions de multiplication-addition/soustraction.
C.xxx	Comparaison parallèle.
MOVE, MOVT	Déplacements conditionnels.

Nouvelles instructions	
LUXC1, SUXC1	Chargement de 8 octets sans alignement.
ALNV	Alignement de donnée.
PLL, PLU, PUL, PUU	Réarrangement d'une donnée PS.
CVT.PS.S, CVT.S.PU, CVT.S.PL	Conversion de/vers une donnée PS.

TAB. 1.1 – Modifications du jeu d'instruction apportées par MIPS V.

Les nouvelles instructions *LUXC1* et *SUXC1* chargent et sauvegardent 64 bits alignés en ignorant les trois bits de poids faibles de l'adresse en opérande. Ces instructions permettent de travailler sur des paires d'opérandes simple précision quand les opérandes font partie d'un vecteur qui n'est pas aligné sur 64 bits. L'instruction *ALNV* peut ensuite correctement réaligner les données. Un exemple (FIG. 1.2) peut être utile.



**Wd** = Word (32 bits); **DWd** = Double Word (64 bits); **fx** = registre flottant; **Tx** = registre entier.

FIG. 1.2 – Utilisation de *ALNV.PS* avec *LUXC1* du jeu d'instructions MIPS V.

Les instructions *Pxx* permettent de copier la moitié haute ou basse d'une valeur PS vers une moitié haute ou basse d'une autre valeur PS. Pour créer une valeur PS (resp. une valeur flottante simple précision) à partir de deux valeurs flottantes simple précision (resp. une valeur PS) on utilise l'instruction *CVT.PS.S* (resp. *CVT.S.PU* ou *CVT.S.PL* selon que l'on retire la partie haute ou basse).

La modification des instructions arithmétiques permet à MIPS V de mettre en œuvre une approche SIMD. En effet, une opération sur une donnée PS (64 bits) réalise en fait deux opérations flottantes simple précision ; le résultat est aussi au format PS.

### MDMX

Le nouveau MDMX<sup>4</sup> annoncé en même temps que MIPS V, n'est pas un jeu d'instructions à part entière, mais une extension optionnelle<sup>5</sup> au jeu d'instructions MIPS, compatible avec MIPS IV et les versions plus récentes.

MDMX définit un ensemble de registres *médias* (qui correspondent physiquement à des registres flottants), de nouveaux types de données entières 8 et 16 bits, et des instructions qui opèrent sur ces types de données. La réelle nouveauté est que les registres *médias* de 64 bits contiennent huit valeurs de 8 bits (*Oct Byte*) ou bien quatre valeurs de 16 bits (*Quad Half*) entières. Les instructions arithmétiques définies par MDMX sur ces type de données permettent donc une approche, ici aussi, SIMD, comme dans le cas des données *PS* de MIPS V, et supportent trois modes d'exécution :

- mode *vecteur-vecteur* : l'instruction opère sur chaque octet (ou demi-mot) du premier registre source et sur l'octet (ou demi-mot) se trouvant à la même place dans le second registre source ;
- mode *vecteur-scalaire* : un unique octet (ou demi-mot) sélectionné dans n'importe quelle partie d'un registre *média* est combiné avec chaque octet (ou demi-mot) du second registre source ;
- mode *vecteur-immédiat* : chaque octet (ou demi-mot) du registre source est combiné avec une valeur immédiate (codée sur 5 bits).

MDMX définit aussi un large registre *accumulateur* de 192 bits qui permet de stocker les résultats de multiplications en mode *vecteur-vecteur* sans débordements et sans perte de précision. Il peut être partitionné en huit champs de 24 bits — cela permet l'accumulation de 256 multiplications d'octets (8 × 8 bits) — ou en quatre champs de 48 bits — cela permet l'accumulation de 65 536 multiplications de demi-mots (16 × 16 bits).

Ainsi, les données multimédias, codées sur 8 ou 16 bits, pourront être traitées avec beaucoup plus de rapidité.

### MIPS<sub>16</sub>

Destiné au marché embarqué, le jeu d'instruction MIPS<sub>16</sub> qui a été développé par *MIPS* et un de ses partenaires *LSI Logic Corporation*, date lui aussi d'octobre 1996. MIPS<sub>16</sub> est un mode particulier où toutes les instructions sont codées sur 16 bits : ceci permet de réduire la taille mémoire totale du code et, ainsi, les coûts des systèmes embarqués.

En fait MIPS<sub>16</sub> est, tout comme *MDMX*, une extension optionnelle qui est compatible, elle, avec toutes les versions du jeu d'instruction MIPS. On peut mixer des instructions MIPS<sub>16</sub> avec les instructions MIPS (I, II, III, IV ou V) existantes par le biais suivant : des instructions de branchement (JALX, JR, JALR) spéciales permettent de passer d'un mode à l'autre.

MIPS<sub>16</sub> comporte presque toutes les instructions générales définies par MIPS I et MIPS II, mais aussi de nombreuses lacunes (on ne peut pas réduire la taille du code et conserver toutes les instructions !). Parmi celles-ci, notons l'absence de toutes les opérations gérées par le coprocesseur ; ce qui signifie l'impossibilité d'exécuter du code MIPS<sub>16</sub> sur des données flottantes. On peut aussi remarquer que le nombre de registres généraux passe de 32 à 8 (codage sur 3 bits au lieu de 5), que la plupart des opérations registre-registre sont destructrices (écrasement d'un des deux registres source) et que les opérandes de type immédiat sont réduits de 16 à 8 bits (les *offsets* dans les instructions de branchement passent de 26 à 10 bits).

4. MIPS Digital Media eXtension, aussi connu sous le nom de Mad Max.

5. Le nom donné par *MIPS* est *Application Specific Extension (ASE)*.

Instructions	Date	Microprocesseurs correspondants	Caractéristiques principales
MIPS I	1984	R2000, R3000	Conforme au jeu d'instructions de base (travail en 32 bits).
MIPS II	1990	R6000	Transfert coprocesseur à 64 bits ; mise à jour atomique de mots ; instructions <i>branch on likely</i> ; déclenchement conditionnel d'exceptions ; instruction de synchronisation des accès à la mémoire partagée.
MIPS III	1991	R4000	Travail en 64 bits.
MIPS IV	1994	R8000, R10000, R5000, RM7000	Adressage <i>registre + registre</i> pour l'unité flottante ; instructions de déplacement conditionnel ; instructions de <i>prefetch</i> ; extension des codes condition flottants ; instruction de multiplication-addition flottante.
MIPS V	1996	—	Format de données flottantes <i>paired-single (PS)</i> et instructions de calcul et de manipulation.
MDMX	1996	—	Formats de données <i>oct byte (OB)</i> et <i>quad half (QH)</i> et instructions de calcul et de manipulation (définition de 8 codes condition spéciaux) ; accumulateur de 192 bits.
MIPS <sub>16</sub>	1996	TinyRISC	Instructions réduites sur 16 bits ; instructions de passage en mode 32 bits (possibilité d'exécuter du code MIPS « classique »)

TAB. 1.2 – Évolution du jeu d'instructions MIPS.

Le lecteur est invité à consulter [8, 9, 10] pour de plus amples détails sur le jeu d'instructions MIPS V et les extensions optionnelles MDMX et MIPS<sub>16</sub>.

### 1.1.2 Pipelines

#### Pipeline entier

Les quatre microprocesseurs étudiés dans ce chapitre sont représentatifs des différentes approches des pipelines RISC actuellement disponibles sur le marché.

Le MIPS **R3000** implémente un pipeline de cinq étages typique d'un microprocesseur RISC traditionnel (voir glossaire, p. 151). La gestion des interblocages au sein du pipeline est effectuée par logiciel.

Le MIPS **R4000** met en œuvre un pipeline très profond de huit étages. Les étages critiques ont été divisés en deux, voire en trois, pour permettre à ce microprocesseur d'atteindre des fréquences élevées.

Contrairement au MIPS R3000, ce microprocesseur met en œuvre une gestion matérielle des interblocages. L'acronyme *Microprocessor without Interlock Pipeline Stage* n'est désormais plus respecté, et ne le sera plus pour les versions postérieures de microprocesseurs MIPS (compatibilité binaire oblige).

Le pipeline entier du MIPS **R8000** diffère largement des structures de pipeline employées par MIPS jusque-là. Premier microprocesseur superscalaire réalisé par MIPS, il peut traiter jusqu'à quatre instructions en parallèle à chaque cycle et en émettre quatre vers les unités d'exécution, parmi deux instructions entières, deux opérations mémoires et deux instructions flottantes.

D'une profondeur de cinq étages, le pipeline du R8000 offre la particularité d'inverser les étages d'exécution et de calcul d'adresses, ceci afin de supprimer la dépendance entre la lecture d'un opérande et son utilisation.

Par ailleurs, ce microprocesseur met en œuvre un découplage au niveau de l'exécution des instructions entières et flottantes. Alors que les opérations entières et les générations d'adresses sont exécutées immédiatement, les instructions flottantes sont rangées dans une file d'attente appelée *Floating-Point Queue (FPQ)*. Cette file, de 15 entrées, évite l'encombrement du mécanisme d'émission par des instructions flottantes en attente d'opérandes et découple l'unité flottante de l'unité entière. Ceci permet de masquer la latence d'accès au cache. La gestion des dépendances internes à l'unité flottante est traitée au sein même de celle-ci.

Enfin, le MIPS R8000 implémente une prédiction de branchement effectuée par le biais d'un cache d'adresses de branchement de 1024 entrées accédé au même moment que le cache d'instructions.

Le MIPS **R10000**, lui aussi superscalaire, va plus loin que son prédécesseur dans le découplage de l'architecture. En effet, le principe des files d'attente est repris et amplifié ; les unités fonctionnelles sont alimentées par trois files d'instructions indépendantes<sup>6</sup> de 16 instructions chacune :

- une file pour les instructions entières (*Integer Queue*) ;
- une file pour les instructions flottantes (*FP Queue*) ;
- une file pour les instructions mémoires (*Memory Queue*).

À partir de ces files, jusqu'à six instructions peuvent être émises vers les unités d'exécution (deux instructions arithmétiques entières, deux instructions arithmétiques flottantes, une instruction lecture/écriture et une instruction de branchement conditionnel flottant) et exécutées dans le désordre.

Quand les instructions sont émises vers leur file respective, elles sont également entrées dans une liste d'instructions dites « actives ». Cette liste contient jusqu'à 32 instructions consécutives et garantit que l'exécution des instructions effectuée de manière spéculative et dans le désordre est conforme à un modèle ordonné pour le reste du système. Une instruction peut être extraite de cette liste quand

6. MIPS appelle cette architecture *ANDES: Architecture with Non-sequential Dynamic Execution Scheduling*.

toutes les instructions qui la précèdent ont été effectuées. Par ailleurs ce concept de liste d'instructions actives est particulièrement utile pour la gestion des branchements ou pour traiter les exceptions. En cas d'exception, la liste est utilisée pour « défaire » les résultats des instructions exécutées dans le désordre.

Chaque unité fonctionnelle a son propre pipeline. Ainsi, la taille de celui-ci est variable (FIG. 1.3). De plus, afin d'éviter les faux aléas de dépendance de données, la technique du renommage de registres est employée sur ce microprocesseur.

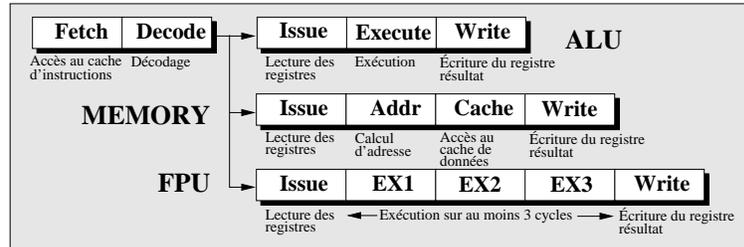


FIG. 1.3 – Différents types de pipeline selon les unités fonctionnelles dans le MIPS R10000.

Le R10000 utilise une prédiction de branchement à deux bits selon l'algorithme de Smith (table de 512 entrées). Les adresse cibles des branchements conditionnels sont recalculées et non prédites [6]. Sur le R10000 on trouve aussi un tampon d'adresse de retour de procédure à une entrée qui évite la lecture de cette adresse dans le banc de registres.

## Pipeline flottant

Sur tous les microprocesseurs étudiés, les pipelines flottants sont sensiblement identiques aux pipelines entiers, les étages de chargement et décodage des instructions étant communs. Cependant, contrairement au cas de l'unité entière dont l'étage d'exécution ne prend qu'un cycle, le temps requis pour exécuter une opération flottante est important en raison de la complexité des calculs effectués.

Les pipelines flottants du **R3000** et du **R4000** ont respectivement une profondeur de six étages et de huit étages.

Comme pour le **R8000**, le **R10000** implémente un pipeline flottant découplé du pipeline entier. Les étages d'exécution sont au nombre minimum de trois (dans le meilleur des cas) ; le pipeline comporte alors de sept étages.

### 1.1.3 Unités arithmétiques

#### Unité arithmétique entière

Sur le microprocesseur **R3000** l'ensemble des opérations est effectué sur 32 bits (arithmétique et calcul d'adresse). Pour cela, 32 registres généraux de 32 bits sont définis.

L'unité de multiplication-division utilise deux registres spéciaux pour stocker les parties haute et basse du résultat. Deux instructions spécifiques permettent de transférer le contenu de ces registres spéciaux dans les registres généraux.

Le MIPS **R4000** assure la compatibilité binaire avec le MIPS R3000 en conservant un mode 32 bits en plus de son mode naturel 64 bits. L'unité centrale fournit 32 registres généraux d'une largeur de 64 bits dont 32 ou 64 sont utilisés selon le mode de fonctionnement.

L'unité entière du **R8000** est sensiblement identique à celle du R4000 à l'exception d'un accroissement du nombre des unités d'exécution parce que l'architecture est superscalaire. Le banc de registres dispose toujours de 32 registres de 64 bits ; les ports de lecture et d'écriture de l'architecture scalaire

du MIPS R4000 ont été étendus (neuf en lecture et quatre en écriture) et permettent l'exécution de quatre instructions en parallèle.

L'unité entière du **R10000** comprend deux unités arithmétiques et logiques entières (ALU1 et ALU2), ainsi qu'une unité de calcul d'adresses alimentées en instructions par la même file d'attente, la *Integer Queue (IQ)* qui a 16 entrées (FIG. 1.1).

Les unités ALU1 et ALU2 traitent toutes deux les additions, les multiplications et les opérations logiques, mais ne sont pas identiques. L'ALU1 traite les instructions de branchement et de décalage, tandis que l'ALU2 s'occupe des multiplications et des divisions utilisant des algorithmes itératifs.

L'unité *load/store* calcule les adresses dans l'ordre du programme, mais les lectures peuvent dépasser les écritures, grâce à une détection des aléas *Read After Write* sur la mémoire.

La technique du renommage de registres qui permet de résoudre les fausses dépendances de données est mise en œuvre par le biais de 64 registres physiques entiers, soit le double de ceux « vus » par l'utilisateur.

### Unité arithmétique flottante (FPU)

L'architecture MIPS définit l'unité flottante comme le *coprocesseur 1 (CP1)*. Cette unité est responsable de l'ensemble des opérations arithmétiques flottantes. Elle est conforme à la norme *IEEE 754*.

L'unité flottante sur le MIPS **R3000** est externe et est appelée R3010. Cette unité dispose de 32 registres généraux de 32 bits (*FGR*). Par ailleurs, 16 registres de 64 bits sont également disponibles par concaténation de deux registres généraux adjacents (*FPR*) pour les opérations sur des valeurs en double précision. Ceux-ci sont des registres logiques, seuls des numéros pairs sont utilisés pour les adresser (numéros impairs invalides).

Le cœur R3000 a été repris pour de nombreux microprocesseurs embarqués, l'unité flottante et l'unité entière cohabitent alors sur le même composant.

Le MIPS **R4000** intègre l'unité flottante sur le même composant que l'unité entière. Elle dispose de 32 registres généraux de 64 bits. Pour assurer une totale compatibilité avec le R3000, un mode de fonctionnement permet de n'utiliser que 16 de ces registres. Par ailleurs, l'instruction de racine carrée a été mise en œuvre par matériel.

Sur le microprocesseur **R8000**, MIPS renoue avec les coprocesseurs flottants externes avec un composant R8010 conçu par *Weitek*. Ce composant inclut deux unités d'exécution à trois opérandes, 32 registres flottants de 64 bits et deux tampons de données de 32 entrées destinés aux opérations de lecture/écriture. Deux instructions peuvent être simultanément exécutées à chaque cycle par cette unité. Les deux files de données permettent d'anticiper l'exécution des opérations de lecture/écriture du fait de l'aspect découplé de cette architecture.

Notons aussi que les nouvelles instructions de multiplication-addition flottantes à trois opérandes ( $\pm(A \times B \pm C) \rightarrow D$ ) définies par MIPS IV ne sont pas conformes à la norme *IEEE 754* car l'arrondi entre la multiplication et l'addition n'est pas fait (on gagne ainsi en précision tout en diminuant la latence).

Le coprocesseur flottant du **R10000**, comme celui du R4000, se trouve sur le même composant. Il est composé de deux sous-unités : un additionneur et un multiplieur. Le multiplieur intègre deux unités d'exécution secondaires qui traitent les divisions, les racines carrées et les opérations à longue latence ; ces unités ne sont pas pipelinées mais peuvent opérer simultanément et en parallèle avec le multiplieur.

Le traitement des opérations de multiplication-addition se fait en deux parties : l'instruction est éclatée en deux opérations distinctes dans les étages de chargement et de décodage. Comme ensuite l'exécution se fait dans le désordre, le résultat de la multiplication est arrondi avant le passage dans

l'additionneur (contrairement au R8000). Ainsi cette instruction *MULT-ADD* est conforme aux exigences de la norme *IEEE 754*.

L'unité flottante du R10000 implémente aussi la technique du renommage de registres : elle inclut 64 registres physiques flottants, soit le double de ceux « vus » par l'utilisateur.

#### 1.1.4 Hiérarchie mémoire

L'architecture MIPS définit des microprocesseurs avec un premier niveau de caches d'instructions et de données séparés : le maintien de la cohérence cache d'instructions/mémoire doit être assuré par logiciel. Un second niveau de cache unifié et externe peut être éventuellement ajouté (il est même obligatoire pour le MIPS R8000 et le R10000).

##### Premier niveau de cache

Le MIPS **R3000** met en œuvre un premier niveau de cache d'instructions et de données externe au composant. Ces caches sont à écritures simultanées (*write-through*) et à correspondance directe (*direct mapped*), physiquement adressés et testés. Leur capacité est variable de 4 à 64 Koctets chacun.

Sur le MIPS **R4000**, les deux caches de premier niveau (instructions et données) sont internes. Ils ont tous deux une taille de 8 Koctets et sont à correspondance directe. Le cache de données met en œuvre une politique d'écriture différée (*write-back*). Ces deux caches sont adressés virtuellement mais sont vérifiés au moyen d'une étiquette physique.

Le MIPS **R8000** met en œuvre la structure de cache la plus particulière avec un premier niveau composé de deux caches internes, l'un pour les instructions, l'autre pour les données entières. Un second niveau de cache, externe au composant, sert de premier niveau de cache pour les données flottantes.

Le cache d'instructions a une capacité de 16 Koctets et est à correspondance directe. Il est structuré en lignes de quatre instructions pour alimenter en un cycle l'unité de chargement et de décodage des instructions. Ce cache est adressé virtuellement et à étiquettes virtuelles, ce qui permet d'utiliser directement l'adresse venant du programme.

Le cache de données (entières seulement) a une capacité de 16 Koctets et est accédé virtuellement mais vérifié au moyen d'une étiquette physique. Il met en œuvre un mécanisme original lors des écritures. Profitant de sa structure à correspondance directe, la donnée est systématiquement écrite dans le cache à l'unique place possible pendant le test de l'étiquette. Puis, si le test de l'étiquette révèle que le secteur est absent du cache, alors le (ou les) bit(s) de validité du mot écrit est (sont) mis à zéro. Ce type de fonctionnement n'est permis que parce que le cache de données est à correspondance directe et à recopie simultanée.

Le **R10000** intègre lui aussi deux caches primaires, un pour les instructions, l'autre pour les données, de 32 Koctets chacun.

Le cache d'instructions est associatif par ensembles à deux voies. Au chargement d'une ligne, les instructions sont partiellement décodées (afin que chaque instruction désigne l'unité d'exécution dans laquelle elle doit être lancée) dans un cache de 4 Koctets, avant d'être placées dans le cache primaire d'instructions. De plus, un bit de validité associé à chaque ligne permet de maintenir la cohérence du cache d'instructions avec la mémoire.

Le cache primaire de données est composé de deux bancs entrelacés associatifs par ensembles à deux voies. Ce cache est non-bloquant. Il utilise une politique d'écriture différée.

Les deux caches primaires sont à indexation virtuelle et à étiquetage physique, comme le R4000.

**À noter :** le R3000 a des caches indexés physiquement et à étiquettes physiques. Le R4000 et le R10000 ont des caches primaires indexés virtuellement et à étiquettes physiques, le cache de données du R8000 a les mêmes caractéristiques mais le cache d'instructions est virtuellement testé.

## Cache secondaire

MIPS propose une version d'entrée de gamme du **R4000** ne disposant pas d'interface pour un cache secondaire (R4000PC). Pour les versions avec cache secondaire, le R4000 dispose d'une interface permettant de gérer directement ce cache (test des étiquettes, ...).

Le cache secondaire du R4000 peut être utilisé comme un cache unifié, comprenant à la fois les données et les instructions, ou comme un cache subdivisé en un cache d'instructions et un cache de données. Sa taille est comprise entre 128 Koctets et 4 Moctets. Ce cache est, quelle que soit sa configuration, à correspondance directe avec la mémoire principale, et utilise une politique d'écriture différée pour la mise à jour des données.

Il est relié au microprocesseur à travers un bus de données de 128 bits qui assure une bande passante suffisante entre les deux niveaux de caches pour minimiser la pénalité en cas de défaut sur le cache primaire. La technique de protection utilisée est la technique *ECC*<sup>7</sup>.

Lorsqu'un défaut de cache survient à la fois dans les caches primaire et secondaire, le second cache est rechargé à partir de la mémoire principale et le premier à partir du second. Il est à noter que comme le microprocesseur assure la gestion des deux caches, les données transitent toujours par le microprocesseur entre la mémoire et le cache secondaire.

Le MIPS **R8000** est destiné au marché des applications numériques flottantes qui requièrent en général un ensemble de travail (ou *working set*) volumineux. Aussi, la structure de son cache secondaire externe privilégie une large bande passante (1.2 Goctets) et une grande capacité (de 4 à 16 Moctets) plutôt que le temps d'accès (accès pipeliné sur cinq cycles). Il est associatif par ensembles à quatre voies et met en œuvre une politique d'écriture différée.

Ce cache joue le rôle d'un cache secondaire pour le premier niveau de cache d'instructions et de données entières, et constitue un cache primaire vis-à-vis de l'unité flottante. Aussi, son efficacité est particulièrement déterminante pour les performances du système. C'est pourquoi les concepteurs ont choisi de faire appel à la technologie alors récente de mémoires synchrones ayant ici un temps de cycle de 12 ns et étant structurées en deux bancs entrelacés. À cause de son importante bande passante, ce cache est désigné par MIPS sous le nom de *Streaming Cache*.

Un cache secondaire externe associatif par ensembles à deux voies et unifié est obligatoire pour le **R10000** qui inclut toute la logique nécessaire à sa gestion. La taille de ce cache peut être comprise entre 512 Koctets et 16 Moctets.

Deux tailles de lignes sont possibles : 64 ou 128 octets. La politique d'écriture est de type différé et l'algorithme de remplacement est de type *LRU*. Le cache est physiquement indexé et physiquement étiqueté.

Il utilise avec le microprocesseur une interface de 128 bits pour les données et de 26 bits pour les étiquettes. Pour limiter la latence d'accès, une table de 8 Kbits est intégrée au microprocesseur. Elle mémorise le banc le plus récemment utilisé. Cette table peut recouvrir jusqu'à 2 Moctets de mémoire du cache secondaire.

La technique de protection des données est toujours la technique *ECC*.

### 1.1.5 Support des systèmes d'exploitation

Les microprocesseurs étudiés fournissent des mécanismes de gestion de la mémoire virtuelle et un support permettant de mettre en œuvre un système d'exploitation. Le *coprocesseur 0* désigne l'unité responsable de la gestion de la mémoire virtuelle.

L'architecture MIPS définit un espace d'adressage virtuel linéaire paginé, c'est-à-dire que l'adresse virtuelle est égale à l'adresse calculée par le programme. Elle est divisée en un numéro de page et un déplacement dans cette page (auxquels s'ajoute un identificateur de processus). L'unité de gestion de la mémoire manipulée par le *coprocesseur 0* est la page.

---

7. Error Correction Code

## Taille des pages

Un petit tableau (TAB. 1.3) vaut peut-être mieux qu'un long discours !

	Taille des pages	Remarques
<b>R3000</b>	4 Koctets	Format constant.
<b>R6000</b>	16 Koctets	Format constant.
<b>R4000</b>	4, 16, 64, 256 Koctets, 1 Moctets ou 4 Moctets	Format variable permettant à chaque processus d'avoir une taille de page spécifique pour adapter la gestion de la mémoire au type de l'application.
<b>R8000</b>	4, 16, 64, 256 Koctets, 1 Moctets ou 4 Moctets	Format variable ; possibilité d'allouer des tailles de pages différentes entre les instructions et les données.
<b>R10000</b>	4, 16, 64 Koctets, ..., 16 Moctets	Format variable permettant des pages encore plus grosses.

TAB. 1.3 – Taille des pages des microprocesseurs MIPS.

## Espace virtuel

Le MIPS **R3000** mettant en œuvre une architecture de type 32 bits, son espace d'adressage physique est limité à 4 Goctets. L'espace d'adressage virtuel se répartit en 2 Goctets pour l'utilisateur et 2 Goctets pour le noyau.

L'espace physique adressable par le **R4000** est de 64 Goctets (adresse physique sur 36 bits). Deux modes de fonctionnement sont utilisés sur ce processeur, un mode 32 bits et un mode 64 bits. Selon ce mode, l'adresse virtuelle est représentée sur 32 ou 64 bits. La taille maximale d'un processus utilisateur est de 2 Goctets dans le premier cas et de 1 Toctets dans le second (on ne tient compte que de 40 bits de l'adresse virtuelle).

Sur le MIPS **R8000**, les espaces d'adressage physique et virtuel ont été considérablement augmentés. L'espace d'adressage virtuel a été étendu à 48 bits alors que l'espace d'adressage physique est de 40 bits, le calcul d'adresses s'effectuant toujours dans l'un et l'autre cas sur 64 bits.

Le **R10000** assure lui aussi une compatibilité ascendante avec la famille des R3xxx, il supporte donc les deux modes d'adressage 32 et 64 bits. Le même *ASID*<sup>8</sup> est utilisé pour les instructions et les données (codé sur 8 bits, soit 256 processus), contrairement au R8000. Par rapport à ce dernier, l'adresse virtuelle a 4 bits de moins (donc 44 bits), et les bits de région ont été modifiés. Il est par conséquent pratiquement impossible d'utiliser des mêmes versions de système d'exploitation sur le R8000 et sur le R10000. L'espace d'adressage physique est toujours de 40 bits.

## Cache de traduction d'adresses (TLB)

Le MIPS **R3000** implémente un TLB totalement associatif, de 64 entrées, commun aux instructions et aux données. L'adressage physique des caches, entraînant des accès systématiques à cette table, explique le nombre relativement élevé d'entrées. Chacune des entrées de cette table est complétée par un identificateur de processus codé sur six bits (soit jusqu'à 64 processus simultanément actifs en mémoire). L'algorithme de remplacement utilisé est implémenté par logiciel (une possibilité de

8. *Address Space Identifier*.

verrouillage de certaines entrées permet de conserver dans les TLB la traduction d'adresses de sections de codes ou de données fréquemment référencées).

Le MIPS **R4000** implémente un TLB de 48 entrées également commun aux instructions et aux données. Chacune de ces entrées permet de représenter les adresses de deux pages virtuelles contiguës. Cette configuration permet ainsi de disposer de 96 entrées effectives. L'algorithme de remplacement utilisé après un défaut sur le TLB est le même que celui du R3000.

Le R4000 possède également un sous-TLB de deux entrées dédié aux instructions et inclus dans le TLB décrit précédemment. Il permet d'effectuer en parallèle la traduction d'une instruction et d'une donnée. En cas de défaut du sous-TLB d'instructions, celui-ci est rechargé depuis le TLB commun. Bien qu'il ne possède que deux entrées, ce dispositif supplémentaire par rapport au R3000 permet d'accélérer l'exécution car il supprime le conflit de banc lors d'un accès simultané aux caches d'instructions et de données et, en raison de la localité spatiale des instructions, deux pages sont dans la plupart des cas suffisantes [11]. Le regarnissage du sous-TLB d'instructions à partir du TLB général ne coûte que quelques cycles. Par ailleurs, l'identificateur de processus a été amené à huit bits (et ce, dès le MIPS R6000).

Le MIPS **R8000** met en œuvre un TLB de 384 entrées, associatif par ensembles à trois voies. L'indexage d'un TLB associatif par ensembles, où peuvent être rangés des descripteurs de pages de différentes tailles, est un problème délicat (choix de l'index). Il est ici résolu simplement par l'utilisation pour chaque *ASID* d'une seule taille de page ; cette taille fait partie du contexte du processus.

L'algorithme de remplacement du TLB est basé sur un algorithme de remplacement aléatoire matériel. Cependant, certaines entrées du premier banc du TLB peuvent être verrouillées afin d'éviter que certaines pages fréquemment accédées perdent la traduction de leur adresse par remplacement.

Le TLB du **R10000** contient 64 entrées (doubles comme le R4000) et est totalement associatif. Un sous-TLB de huit entrées est utilisé pour les instructions. En cas de défaut sur le TLB général, une exception est générée et le remplacement est fait par logiciel.

## Protection

Les microprocesseurs disposent de plusieurs modes de fonctionnement. Ces modes protègent en écriture ou en lecture certaines zones mémoire. Sur les microprocesseurs MIPS, la protection est réalisée à chaque accès à la mémoire au cours de la traduction d'adresse et s'effectue au niveau de la page ; les bits de poids fort des adresses sont utilisés pour distinguer le mode de fonctionnement.

Le MIPS **R3000** implémente les deux modes de fonctionnement communément utilisés : le mode *noyau* et le mode *utilisateur*. Comme on l'a dit, la mémoire virtuelle est partitionnée en deux zones de 2 Goctets, la zone d'adresses basses étant destinée au mode *utilisateur*, la deuxième partie étant réservée au mode *noyau*.

Le MIPS **R4000** se caractérise par un mode de fonctionnement supplémentaire : le *superviseur*. Deux bits d'adresse sont alors utilisés pour déterminer le mode de fonctionnement (trois en mode 32 bits). Ces trois modes sont :

- Le mode *noyau* : ce mode correspond au mode *superviseur* de la plupart des microprocesseurs. Le CPU entre en mode *noyau* à travers un appel explicite (instruction *SYSCALL*) ou lorsqu'une exception est détectée et y reste jusqu'à ce que le contexte dans lequel l'exception a été levée soit restauré.
- Le mode *superviseur* : ce mode représente une nouveauté par rapport au R3000. Il consiste en un mode intermédiaire entre le mode *noyau* et le mode *utilisateur*, et a été essentiellement mis en œuvre dans le but de permettre les systèmes d'exploitation implémentés en couches. Ceci assure une plus grande sécurité pour le système. Ainsi, certaines zones accessibles en mode *superviseur* ne le sont pas en mode *noyau* et vice-versa.

- Le mode *utilisateur* : ce mode est similaire à celui du R3000 mis à part que la taille de l'espace adressable par un processus utilisateur passe de 2 Goctets à 1 Toctets. L'espace utilisateur est accessible en mode *noyau* et en mode *superviseur* dans le R4000.

Le MIPS **R8000** ne supporte que deux modes d'exécution : le mode *noyau* et le mode *utilisateur*. Le mode *superviseur* de son prédécesseur, le R4000, n'a pas été implémenté mais est repris avec les deux autres modes pour le **R10000** sur lequel les 5 bits de poids fort de l'adresse sont utilisés pour les coder [6].

### 1.1.6 Support multiprocesseur

Le degré de performance atteint par les microprocesseurs étudiés les rendent attrayants comme brique de base des systèmes multiprocesseurs (en particulier à mémoire partagée et à bus unique). Dans de tels systèmes, il est nécessaire de disposer de supports matériels pour la cohérence de cache qui assurent l'exclusion mutuelle et la mise à jour des différents caches, afin que chaque microprocesseur puisse lire des données cohérentes.

Le MIPS **R4000** est le premier des microprocesseurs MIPS à inclure de telles fonctionnalités. Il permet les protocoles suivants :

- l'invalidation en écriture (*write invalidate*) : lorsqu'une donnée est modifiée par un microprocesseur, tous les caches possédant une copie de cette donnée mettent la ligne qui la contient dans l'état invalide. Ce protocole est également connu sous le nom de protocole *MESI* d'après les quatre états possibles d'une ligne de cache : *Modified*, *Exclusive*, *Shared*, *Invalid* (voir glossaire p. 152).
- la diffusion en écriture (*write update*) : lorsqu'une donnée est modifiée par un microprocesseur, la ligne qui la contient est diffusée sur le bus et chaque cache en contenant une copie la met à jour.

Le protocole de cohérence utilisé est déterminé au niveau de la page, par plusieurs bits dans le descripteur de page. Chaque page possède un attribut qui le spécifie. Ainsi, et contrairement au **R3000** qui ne disposait que du protocole d'invalidation en écriture, le R4000 peut simultanément utiliser l'invalidation en écriture, la diffusion en écriture ou l'absence de cohérence.

Par ailleurs, le R4000 intègre des instructions spécifiques utilisées lors des accès à la mémoire partagée. Ces instructions définies dès MIPS II offrent à l'utilisateur la possibilité de mettre en œuvre des primitives de synchronisation telles que : sémaphores, *test-and-set*, *bit-level locks*, séquenceurs et compteurs d'événements. Le MIPS R3000 était dépourvu de telles instructions.

Le protocole de cohérence de cache employé par le MIPS **R8000** est le protocole à invalidation en écriture. Ce protocole est actuellement le plus utilisé dans les systèmes incluant des caches à écriture différée en mémoire.

D'autre part, un des éléments importants d'un système bâti autour du MIPS R8000 est le *contrôleur de cache* qui est chargé de l'interface entre le processeur, la mémoire principale et le reste du système. Ce composant n'est pas proposé par MIPS, mais ses fonctionnalités sont spécifiées. L'intervention de ce *contrôleur de cache* est requise dans les cas suivants :

- interface avec le reste du système (chargement de données à partir de la mémoire principale ou mise à jour de la mémoire à partir du cache secondaire, transaction de cohérence avec les autres microprocesseurs) ;
- gestion de la cohérence des deux niveaux de caches (cache secondaire externe et cache interne) par maintien de la propriété d'inclusion<sup>9</sup> ;

9. Toute donnée présente dans le cache primaire doit être présente dans le cache secondaire.

- gestion de la cohérence du cache secondaire externe avec les autres caches du système.

Le support multiprocesseur du **R10000** a été considérablement amélioré par rapport aux générations précédentes. Le R10000 supporte une configuration en grappe permettant de connecter sur le même bus jusqu'à quatre microprocesseurs avec seulement l'adjonction d'un composant externe servant d'interface avec les autres ressources du système. Cette configuration réduit non seulement la complexité du système mais aussi son coût. Comme pour le R8000, le protocole de cohérence de cache du R10000 est de type MESI.

Le contrôleur de cache est intégré au R10000, ainsi qu'un mécanisme permettant de dupliquer en externe les étiquettes du cache de secondaire. De ce fait, seules les transactions de cohérence nécessitant une action sont vues par le microprocesseur.

Le bus système de 64 bits a complètement été redéfini à partir de celui des MIPS R4x00. Ce nouveau bus, dit « à avalanche », supporte le protocole à « transactions imbriquées » (*split-transaction protocol*) et permet ainsi jusqu'à huit transferts en même temps.

### 1.1.7 Technologie

Afin que le lecteur puisse se rendre compte des progrès technologiques effectués d'une génération de microprocesseurs à l'autre, un tableau (TAB. 1.4) en résume les données technologiques.

	Date	Intégration (CMOS)	Nombre de transistors (en millions)	Taille (en mm <sup>2</sup> )	Fréquences internes (en MHz)
<b>R3000</b>	1988	1,2 $\mu\text{m}$	—	—	40
<b>R4000</b>	1992	1,0 $\mu\text{m}$	1,1	213	100
<b>R8000</b>	1994	0,7 $\mu\text{m}$	3,4	298	75
<b>R10000</b>	1995	0,35 $\mu\text{m}$	6,7	297,1	200, 275
<b>R5000</b>	1996	0,35 $\mu\text{m}$	3,6	84	180, 200
<b>RM7000</b>	1997	0,25 $\mu\text{m}$	4	80	300

TAB. 1.4 – *Évolution de la technologie des microprocesseurs MIPS.*

On peut constater les différences considérables existantes entre le R8000 et le R10000, microprocesseurs sortis à peu de temps d'intervalle : les tailles des composants sont approximativement les mêmes, et pourtant le MIPS R10000 intègre le double de transistors ; la fréquence est aussi bien plus élevée.

### 1.1.8 Remarques

Le microprocesseur MIPS R10000, haut de gamme actuel des microprocesseurs MIPS, intègre quelques concepts du R8000 qu'il approfondit : architecture superscalaire découplée et principe des files d'attente. Cependant, on peut remarquer que d'autres techniques « exotiques » du R8000 ont été abandonnées (second niveau de cache servant de premier niveau pour les flottants, *ASID* différent pour les données et les instructions, ...) au profit de techniques plus simples ou plus anciennes mises en œuvre dans le R4000 (coprocesseur flottant intégré, sous-TLB d'instructions, remplacement logiciel du TLB, 3 modes de protection, ...). À noter aussi l'augmentation de l'intégration et avec elle, la capacité des premiers niveaux de cache (instructions et données) qui a quadruplée depuis le R4000.

## 1.2 Les microprocesseurs MIPS récents et à venir

*QED*<sup>10</sup> une firme créée en 1991 par d'anciens employés de *MIPS* se dédie principalement à la conception de microprocesseurs d'entrée de gamme, en particulier pour MIPS. Elle a développé pour MIPS le successeur des MIPS R4600 et R4700, le **MIPS R5000**, destiné au marché des stations de travail d'entrée de gamme. Annoncé en début 1996, le MIPS R5000 est dérivé du microprocesseur *Orion* (R4600). Il est commercialisé par *NEC*, *IDT* et *NKK*.

Depuis fin 1996, *QED* a acquis une licence de fabrication et de vente de produits MIPS. Ainsi, la firme propose son propre microprocesseur, le **RM7000**<sup>11</sup> qui devrait sortir en 1997. Le RM7000 est destiné au marché embarqué haut de gamme, mais *QED* n'exclut pas de l'intégrer dans des stations de travail d'entrée de gamme.

Dans ce qui suit, le lecteur trouvera une présentation succincte mais structurée des deux microprocesseurs à la lumière des informations disponibles sur ces deux produits. Les nouveaux concepts introduits par les précédents MIPS R8000 et MIPS R10000 ne seront pas développés, mais des références précises permettront de trouver des explications<sup>12</sup>.

Certaines comparaisons entre le R5000 et le RM7000 sont faites dans une dernière partie, sous forme d'un tableau récapitulatif.

### 1.2.1 Le microprocesseur MIPS R5000

#### Jeu d'instructions

Le MIPS R5000 est le troisième microprocesseur à implémenter le jeu d'instructions MIPS IV (voir section 1.1.1, p. 6), il est entièrement compatible avec les microprocesseurs MIPS R8000 et MIPS R10000. L'instruction de multiplication-addition flottante permet d'améliorer les performances de la FPU par rapport aux MIPS R4x00.

#### Architecture : vue générale

Le MIPS R5000 intègre un cœur superscalaire de degré 2, ainsi que de grands caches primaires de 32 Koctets pour les instructions et de 32 Koctets pour les données (FIG. 1.4).

#### Séquencement et exécution des instructions

Bien que le R5000 soit un dérivé du R4600 [1], l'addition de l'exécution superscalaire des instructions a remis en cause la logique de contrôle. À chaque cycle une instruction entière et une instruction flottante peuvent être émises vers les unités d'exécution (les « opérations entières » dont il est question ici sont les calculs arithmétiques entiers, les opérations mémoire et les branchements ; voir TAB. 1.6, p. 24). L'aspect superscalaire est réduit ici à sa plus simple expression : pour l'exécution de code uniquement entier, ce processeur est scalaire. Le pipeline entier est traditionnel et comporte les cinq étages classiques, comme celui du R4600.

Le MIPS R5000 intègre de nombreuses techniques empruntées au R10000, dont le prédécodage des instructions dans le cache primaire de données. Quatre bits sont ajoutés à chaque paire d'instructions pour indiquer si les deux instructions peuvent être séquencées ensemble ; ces bits sélectionnent aussi l'unité fonctionnelle pour chaque instruction.

Dans les deux unités fonctionnelles (entière, flottante), les calculs à longue latence (comme les divisions, les calculs de racine carrée, ...) sont pris en charge par une sous-unité. Ils sont ainsi exécutés en parallèle avec d'autres opérations à latence courte (telles les additions, soustractions, ...).

---

10. *Quantum Effect Design*.

11. Pour distinguer son microprocesseur MIPS, *QED* a choisi la désignation RM (pour RISCMark) au lieu de R.

12. Nous rappelons au lecteur la présence du glossaire en page 149.

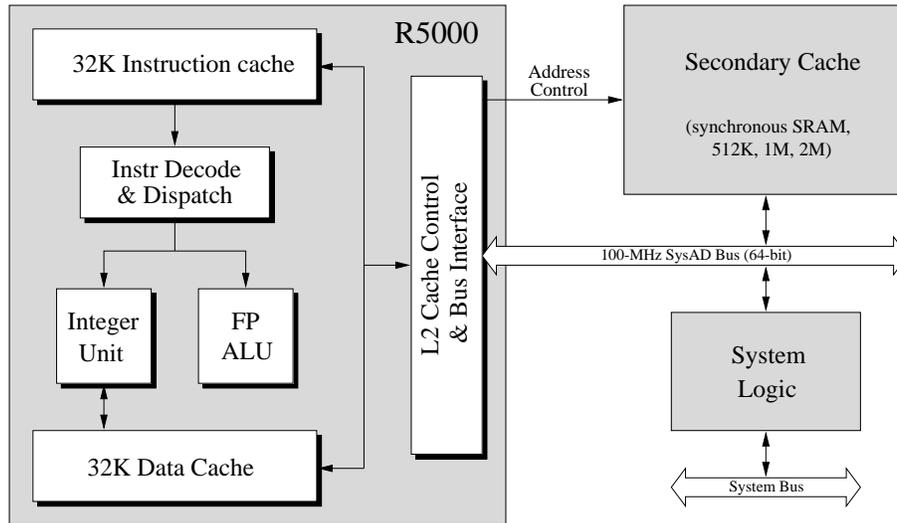


FIG. 1.4 – Vue générale du MIPS R5000.

L'unité flottante a été améliorée par rapport à celle de son prédécesseur, en particulier pour les calculs en simple précision. Remarquons que pour les applications uniquement flottantes, la structure superscalaire du R5000 est relativement efficace : les échanges avec la mémoire peuvent se faire en parallèle avec les opérations sur les registres flottants. Le tableau 1.5 montre les débits et les latences du MIPS R5000 et du R4600 pour les opérations entières et flottantes.

	R5000		R4600	
	Simple	Double	Simple	Double
Multiplieur entier	4/5	8/9	6/8	6/8
Diviseur entier	36/36	68/68	32/32	61/61
Additionneur flottant	1/4	1/4	4/4	4/4
Multiplieur flottant	1/4	2/5	6/8	6/8
Multi-add flottant	1/4	2/5	6/12	6/12
Diviseur flottant	19/21	34/36	32/32	61/61
Racine carrée	36/38	66/68	31/31	60/60

TAB. 1.5 – Latences et débits (en cycles) du MIPS R5000 et du R4600.

## Hiérarchie mémoire

Les tailles des caches primaires du MIPS R5000 ont été doublées (32 Koctets pour le cache d'instructions et 32 Koctets pour le cache de données) par rapport à celles du R4600, alors que la structure associative par ensembles à deux voies a été maintenue. Les deux caches sont virtuellement indexés et physiquement étiquetés ; la taille des lignes sur chaque cache est de 32 octets. Les écritures sur le cache de données peuvent être simultanées ou différées, la sélection se fait page par page.

Un tampon d'écriture est implémenté pour permettre au CPU de stocker des données à écrire dans la mémoire principale, lui évitant ainsi l'accès au bus système. Ce buffer peut contenir quatre paires d'adresses et de données sur 64 bits, ou une ligne de cache.

L'adjonction d'un cache secondaire externe connecté sur le bus principal de 64 bits est possible mais optionnelle: la logique de contrôle de ce cache est intégrée sur le microprocesseur, ce qui n'était pas le cas du R4600. Le bus principal de 64 bits est compatible avec le bus utilisé sur la plupart des MIPS R4x00: l'adaptation des cartes précédemment développées pour ces processeurs est donc possible. Le contrôleur de cache secondaire supporte plusieurs tailles de mémoire (512 Koctets, 1 Moctets et 2 Moctets) et les deux politiques d'écriture simultanée et différée. L'accès à la mémoire principale peut débiter en parallèle avec l'accès au cache secondaire, ce qui réduit la latence en cas de défaut sur le cache secondaire.

## Support des systèmes d'exploitation

Les microprocesseurs MIPS 4600 étaient à l'origine destinés aux PC sous *Windows NT*. Depuis l'arrêt du développement de *Windows NT* pour MIPS, le R5000 vise plutôt les stations de travail, tout en gardant certaines compatibilités avec ses prédécesseurs.

### Espace d'adressage

Comme pour le cas du R4000, les deux mode 32 et 64 bits sont supportés sur le R5000. Les autres caractéristiques de l'espace d'adressage sont identiques :

- en mode 32 bits, espace d'adressage utilisateur de 2 Goctets ;
- en mode 64 bits, espace d'adressage utilisateur de 1 Toctets ;
- espace d'adressage physique de 64 Goctets codés sur 36 bits ;

La taille des pages de l'espace virtuel est variable et suit le schéma de celles du R10000 : de 4 Koctets à 16 Moctets par multiples de quatre.

### Cache de traduction d'adresses

Le TLB du R5000 est, comme celui du R4000, commun aux instructions et aux données, et possède 48 entrées (doubles). Les défauts sur ce cache génèrent une exception et sont traités par logiciel.

### Protection

Les trois modes utilisés par le R4000 et le R10000 sont également présents sur le R5000 : mode *utilisateur*, mode *superviseur* et mode *noyau* (voir p. 16).

## Bus et interface système

Le bus système supporté par le R5000 est le *SysAD bus* que l'on retrouve sur de nombreux microprocesseurs MIPS. Les données et les adresses sont multiplexées sur 64 bits. Ce bus supporte aussi des accès pipelinés à une fréquence de 100 MHz. Il est de plus commun au reste du système et au cache secondaire externe s'il est implanté.

## Support multiprocesseurs

Comme le R4600, la version actuelle du MIPS R5000 ne peut fonctionner que dans un système uni-processeurs. Cependant, le boîtier inclut huit broches pour un support multiprocesseurs. *MIPS* pense intégrer ce support multiprocesseurs (jusqu'à quatre microprocesseurs) dans les prochaines versions du R5000.

## Technologie

Le MIPS R5000 est réalisé en technologie CMOS à 0.35  $\mu\text{m}$ . Il fonctionne à la fréquence de 200 MHz. *MIPS* espère plus tard le commercialiser à une fréquence de 250 MHz et prévoit d'améliorer

l'intégration (0.25  $\mu\text{m}$ ) en 1997. Ce microprocesseur est actuellement disponible chez *IDT*, *NEC* et *NKK*.

Remarquons de plus que le R5000 n'offre pas la compatibilité du brochage avec ses prédécesseurs, il ne pourra donc pas les remplacer dans les systèmes déjà existants sans modifications matérielles.

Pour de plus amples informations sur ce microprocesseur, le lecteur pourra consulter [12, 13].

## 1.2.2 Le microprocesseur RM7000

La philosophie de *QED* sur ce composant est : une mise en œuvre simple, mais efficace. En effet, aucun mécanisme n'est réellement nouveau, mais, par exemple, la première version du RM7000 est annoncée comme étant cadencée à 300 MHz.

Le RM7000 de *QED* devrait sortir au milieu de cette année, et la production en masse est prévue pour 1998. Ainsi, les informations disponibles sur ce composant sont assez limitées. Cependant, les originalités du microprocesseur sont clairement mises en avant par le concepteur, ce qui nous permet de les exposer.

### Jeu d'instructions

Le RM7000 implémente le jeu d'instructions MIPS IV (voir section 1.1.1, p. 6).

### Architecture : vue générale

Le RM7000 est un microprocesseur superscalaire de degré deux dont la principale caractéristique est d'intégrer de très larges caches non bloquants (288 Koctets au total occupant plus de 50 % de la surface totale du composant). Notons que contrairement au R5000, il intègre un deuxième niveau de cache unifié. Les concepteurs présentent ce processeur comme une mémoire SRAM avec un processeur attaché, et non le contraire.

### Séquencement et exécution des instructions

Le RM7000 est un processeur superscalaire de degré deux, mais avec des possibilités supérieures au R5000. La symétrie de traitement des instructions est quasi-totale (voir TAB. 1.6, p. 24), seules quelques combinaisons d'instructions ne peuvent pas avoir lieu en parallèle, à savoir :

- deux branchements ;
- deux références mémoire ;
- deux opérations flottantes ;
- une opération flottante avec un branchement.

De plus, l'ordre, dans les paires d'instructions qui peuvent être séquencées, n'intervient pas.

Le pipeline entier implémenté est un pipeline classique de cinq étages (cf. R3000). En ce qui concerne les branchements, le RM7000 n'implémente pas de prédiction ni d'exécution spéculative, mais, lorsqu'une instruction de branchement est décodée, le CPU, au cycle suivant, charge les deux instructions correspondantes au branchement pris (les deux instructions en séquence qui viennent d'être chargées sont mises dans un buffer). Ainsi, si le branchement est pris, les instructions sont disponibles; s'il n'est pas pris, les instructions en séquence sont aussi disponibles depuis le buffer prévu à cet effet.

Le pipeline flottant est inspiré du R5000, mais aucune information n'est encore disponible sur les unités d'exécution flottantes.

## Hiérarchie mémoire

L'originalité de ce microprocesseur réside dans sa hiérarchie mémoire. En effet, il intègre deux niveaux de cache. Le premier niveau comprend deux caches de 16 Koctets chacun (un pour les données, l'autre pour les instructions) associatifs par ensembles à quatre voies. La taille des lignes est de 32 octets dans les deux et les écritures sont différées. De plus, ces **caches primaires** sont non-bloquants — une lecture peut-être satisfaite même si une autre lecture précédente est en cours de traitement (donnée absente du cache primaire voire secondaire) — ce qui est une première pour *QED*.

Le **cache secondaire** intégré est unifié et a une capacité inégalée aujourd'hui de 256 Koctets. Comme les caches primaires, il est associatif par ensembles à quatre voies, à écriture différée, et non-bloquant. La taille des lignes est toujours de 32 octets.

De ce fait, les accès mémoire sont très rapides : un accès au cache primaire prend un cycle, et un accès au cache secondaire (défaut du cache primaire) est réalisé en trois cycles. L'augmentation de la bande passante entre les deux niveaux de cache, induite par l'intégration du second niveau, permet au RM7000 de ne pas être trop pénalisé par la relative (au R5000) petite taille de ses caches primaires.

Le RM7000 intègre aussi un contrôleur pour un éventuel **troisième niveau de cache** externe. La capacité de ce cache tertiaire peut aller jusqu'à 8 Moctets ; il est à correspondance directe avec la mémoire principale, et les écritures sont simultanées.

## Bus et interface système

Afin de complètement supporter l'aspect non-bloquant des caches, le RM7000 implémente une nouvelle broche qui permet au processeur de reprendre la main si une lecture est déjà en cours.

Une nouvelle facilité est donnée quant au choix de la fréquence externe : elle pourra être de 2 à 5 fois plus lente que la fréquence interne par pas de  $\frac{1}{2}$ .

## Technologie

Le RM7000 sera fabriqué en technologie CMOS à  $0.25 \mu\text{m}$ , et sera séquencé à 300 MHz. Selon *QED*, la dissipation maximale ne devrait pas dépasser 12,5 W (10 W sous des conditions typiques d'utilisation). L'alimentation se fait en 3,3 V, ce qui est assez étrange pour un composant CMOS à  $0.25 \mu\text{m}$  (les tensions appliquées avec une telle technologie sont désormais plutôt 2,5 ou 1,8 V).

Cependant, la dissipation thermique totale est réduite grâce à la présence des larges caches internes qui font que les contrôleurs de signaux externes très gourmands en puissance sont utilisés moins souvent : il y a moins de défauts de cache — les caches internes contribuent aussi à améliorer les performances du microprocesseur et sont donc doublement utiles.

### 1.2.3 Comparaison des deux microprocesseurs

Le MIPS R5000 et le RM7000 sont destinés aux mêmes applications (marché embarqué haut de gamme, stations de travail d'entrée de gamme), de ce fait, il peut être utile de les comparer.

Le tableau 1.6 (p. 24) reprend les caractéristiques générales des deux microprocesseurs. Les données technologiques de la plupart des microprocesseurs MIPS sont comparées dans le tableau 1.4, p. 18. La structure des pipelines étant la même sur les deux microprocesseurs, nous n'avons pas jugé utile de la référencer.

### 1.2.4 La famille RM52xx

*QED* a annoncé en mars 1997 la disponibilité d'échantillons de ses deux nouveaux microprocesseurs : le **RM5230** et le **RM5260** ; la production en masse est prévue pour la fin de l'année 1997. Ces deux microprocesseurs sont des dérivés du R5000 qui visent les mêmes marchés que ceux du RM7000 :

	<b>R5000</b>	<b>RM7000</b>
Jeu d'instructions .....	MIPS IV	MIPS IV
Séquencement d'instructions de type...		
ALU-ALU .....	—	oui
ALU-MEM .....	—	oui
ALU-Branch .....	—	oui
MEM-Branch .....	—	oui
Branch-Branch .....	—	—
MEM-MEM .....	—	—
FP-MEM .....	oui	oui
FP-ALU .....	oui	oui
FP-Branch .....	oui	—
FP-FP .....	—	—
Premier niveau de cache :		
Capacité en ko (I – D) .....	32 – 32	16 – 16
Associativité .....	2 voies	4 voies
Deuxième niveau de cache :		
Capacité en ko .....	—	256 (unifié)
Associativité .....	—	4 voies
Cache externe :		
Capacités en Moctets .....	de 0 à 2	de 0 à 8
Associativité .....	direct mapped	direct mapped
Fréquence de séquencement (MHz) ...	200	300

TAB. 1.6 – *Caractéristiques principales des microprocesseurs MIPS R5000 et RM7000.*

embarqué haut de gamme et stations de travail d'entrée de gamme, même si leur performances devaient être moindre. Pour cela, les deux microprocesseurs sont superscalaires de degré deux grâce à une unité entière et une unité flottante séparées. Leurs différences principales résident dans leur interface bus système et dans les fréquences de séquençement : le RM5230 atteint 133 MHz, alors que le RM5260 peut aller jusqu'à 150 MHz.

Le RM5230 et le RM5260 sont dit 64 bits et exécutent tous deux des instructions du jeu d'instructions MIPS IV. *QED* leur a permis aussi l'exécution d'une instruction de multiplication-addition.

Les deux microprocesseurs comportent deux caches séparés pour les instruction et les données, de 16 Koctets chacun. Ces caches sont associatifs par ensembles à deux voies et laissent le choix quant à la politique d'écriture (simultanée ou différée).

L'interface système du RM5230 prévoit un bus de 32 bits cadencé à, au plus, 67 MHz. Celle du RM5260 met en œuvre un bus de 64 bits cadencé à la même fraction de fréquence interne que son « collègue » ( $\frac{1}{2}$ ), soit 75 MHz maximum.

Le procédé de fabrication est CMOS à 0,35  $\mu\text{m}$  pour les deux microprocesseurs. Le RM5260 consomme 5 W (typ.) sous 3,3 V. Le RM5230 consomme moins, 3,3 W, avec la même alimentation, ce qui fait son intérêt (avec le coût) pour les systèmes embarqués.

### 1.3 Développement de systèmes à base de microprocesseurs MIPS

Après avoir étudié les microprocesseurs qui ont imposé MIPS sur le marché des RISC, nous abordons dans cette partie quelques uns des microprocesseurs dérivés de ces architectures. Les principaux systèmes sont bâtis autour des architectures R3000 et R4000, ou alors sont des implémentations des jeux d'instructions MIPS particulières aux constructeurs.

La stratégie de *MIPS Technologies* est le partenariat avec des fabricants de semi-conducteurs. *MIPS* conçoit des processeurs RISC qui sont ensuite fabriqués et vendus par les fondeurs. Ces constructeurs peuvent aussi développer leurs propres produits MIPS s'ils possèdent une licence technologique, ce qui permet de diversifier les applications de cette architecture, et de s'introduire sur de plus nombreux marchés, notamment celui très porteur des applications embarquées. Ainsi les produits MIPS peuvent être fabriqués en plus grand nombre, et ceci permet de financer le développement de microprocesseurs plus sophistiqués destinés à des marchés plus spécialisés tels celui des station de travail haut de gamme.

Voici une liste non exhaustive de fondeurs de microprocesseurs d'architecture MIPS licenciés (la firme *Quantum Effect Design* possède une licence mais n'assure pas la fabrication) :

- *NEC Electronics, Inc.* ;
- *NKK America, Inc.* ;
- *Toshiba* ;
- *Integrated Device Technology (IDT)* ;
- *LSI Logic Corporation* ;
- *Siemens components, Inc.* ;
- *Philips*.

Ce chapitre ne constitue pas une liste exhaustive de l'ensemble des microprocesseurs développés à partir de cette architecture, mais reprend le travail présenté dans [1], et le complète.

### 1.3.1 Rappels sur les microprocesseurs dérivés des MIPS R3000 et R4000

Depuis 1991 avec la sortie des R3051 et R3052, de très nombreux microprocesseurs destinés au marché embarqué ont vu le jour. Parmi ceux-ci, beaucoup ont été présentés dans [1]. Nous nous bornerons donc ici à faire un succinct rappel sur ces composants-là dans un tableau récapitulatif (TAB. 1.7), et nous présenterons les microprocesseurs apparus depuis dans les deux parties suivantes.

Le R3081 est représentatif de la famille variée à laquelle il appartient : les *RISControllers*. Cette famille compte aussi parmi ses membres les R3051/3052 (1991), le R3041 (1992), et les R3071/3071E (07/1994). Dans la même catégorie, citons aussi le *Performance Semiconductor PIPPER* (1991), et les *LSI Logic LR 33000* et *LR 33020*.

Parmi d'autres dérivés du R3000, il existe aussi le *Custom* de *LSI logic*, le *VR3000* (03/1993) de *NEC*, le *LR33120* (06/1993) de *LSI logic*, et le *R3100* (10/1993) de *Sony* et *HDL*.

On peut aussi signaler l'existence des microprocesseurs suivants qui sont dérivés du R4000 :

- le *R4700* (*IDT*, 10/1994), successeur du R4600 ;
- le *R4650* (*IDT*, *NKK*, 11/1994) et son très proche dérivé le *R4640* (mi 1996) ;
- le *R4100* (*NEC*, 03/1995) issu du R4200 (prédécesseur du R4300) ;
- le *R4300i* (*NEC*, 04/1995).

### 1.3.2 Les microprocesseurs dérivés du MIPS R3000

Depuis l'édition du rapport [1], aucun microprocesseur nouveau dérivé du MIPS R3000 n'est sorti, mais plutôt quelques systèmes intégrant des microprocesseurs déjà existants avec des entrées/sorties. Ces composants visent certaines applications embarquées, et peuvent être utilisés comme coprocesseurs.

Parmi le peu de systèmes dérivés du R3000, citons à titre d'exemple le **R3903F** proposé par *Toshiba* en octobre 1996. Ce microprocesseur est ce que *Toshiba* appelle un *Application Specific Standard Product (ASSP)*, destiné au marché des « ordinateurs de bord » automobiles.

Le R3903F est composé d'un R3900 (voir TAB. 1.7, p. 27, et [1]) auquel on a ajouté de nombreuses entrées/sorties et les fonctionnalités nécessaires à la gestion d'un écran à cristaux liquides couleur.

Les entrées/sorties fournies forment un ensemble assez complet :

- quatre canaux série UART<sup>13</sup> ;
- un port parallèle 16 bits ;
- deux canaux DMA<sup>14</sup>.

Le lecteur remarquera l'extrême spécification des produits dérivés du MIPS R3000 aujourd'hui, à travers cet exemple.

### 1.3.3 Les microprocesseurs dérivés du MIPS R4000

À notre connaissance, seuls deux dérivés du MIPS R4000 ont vu le jour en 1996<sup>15</sup> : le **VR4101** (octobre 1996) de *NEC* (à ne pas confondre avec le TR4101, dit *TinyRISC*, de *LSI Logic* présenté dans la partie suivante) et son successeur le VR4102 (avril 1997).

13. *Unidirectional Asynchronous Receiver and Transmitter*.

14. *Direct Memory Access*.

15. Le R4640 d'*IDT* sorti à la mi-1996 est très proche du R4650 : seul le bus système a été modifié, il passe de 64 à 32 bits.

	<b>R3081</b>	<b>R3900</b>	<b>R4400</b>	<b>R4600</b>	<b>4300</b>
Annoncé en...	01/1992	02/1995	début 1993	07/1993	05/1995
Fabricant	<i>IDT</i>	<i>Toshiba</i>	Ensemble des partenaires de <i>MIPS</i>	<i>QED</i>	<i>MIPS, NEC</i>
Jeu d'instructions	MIPS I	hybride (MIPS I, II)	MIPS III	MIPS III	MIPS III
FPU intégrée	oui	non	oui	oui	oui
Caractéristiques spéciales	Bus optionnel rapide, mode faible consommation.	Technique de <i>scoreboarding</i> , multiplieur 32 bits rapide, mode faible consommation.	Améliorations au R4000: caches plus grands, fréquence interne plus haute, tampon d'écriture supplémentaire.	Pipeline réduit à 5 étages (idem R3000), mode faible consommation.	Pipeline réduit à 5 étages (idem R3000).
Cache instructions Structure	16 Koctets ou 8 Koctets correspondance directe	4 Koctets (opt.) correspondance directe	16 Koctets correspondance directe	16 Koctets associatif 2 voies	16 Koctets correspondance directe
Cache données Structure Écriture	4 Koctets ou 8 Koctets correspondance directe simultanée	1 Koctets associatif 2 voies simultanée	16 Koctets correspondance directe différée	16 Koctets associatif 2 voies différée	8 Koctets correspondance directe différée
Deuxième niveau de cache (support pour)  Structure Écriture	non  — —	non  — —	de 128 Koctets à 4 Mcoctets (versions SC et MC)  correspondance directe différée	non  — —	non  — —
TLB	64 entrées, associatif (version E)	?.	48 paires d'entrées, associatif (idem R4000)	48 paires d'entrées, associatif (idem R4000)	32 paires d'entrées, associatif
Espionnage du bus	externe	oui	configurable (version MC)	non	non
Bus de données et d'adresses	multipléxé	séparé	multipléxé (SysAD Bus)	multipléxé (SysAD Bus)	séparé
Fréquences (en MHz)	20, 25, 33, 40, 50	de 15 à 50	150, 200, 250	150 (1994)	133
Technologie (CMOS)	?.	0,6 $\mu\text{m}$	0,6 $\mu\text{m}$	0,6 $\mu\text{m}$	?.

TAB. 1.7 – Caractéristiques des principaux microprocesseurs embarqués à base d'architecture MIPS depuis 1992

Le VR4101 est destiné au marché spécifique des *Personal Digital Assistant*, et a déjà (en novembre 1996) été intégré dans un produit le *MobilePro* qui est un PC de poche supportant *Windows CE*® (*Consumer Electronics*)<sup>16</sup>.

De même que les dérivés du R3000, celui du R4000 n'est pas réellement nouveau dans la mesure où il intègre un microprocesseur déjà existant, le R4100. Ce microprocesseur a été détaillé dans [1], donc nous nous contenterons de rappeler ici ses caractéristiques principales (mises à jour en rapport avec le VR4101), à savoir :

- Il supporte le jeu d'instruction MIPS III (excepté les instructions atomiques de mise à jour de mots : SC, LL), et inclut une nouvelle instruction de multiplication-addition de type (16 + 64 → 64 bits) utile pour les applications audio et video.
- Son pipeline comporte cinq étages.
- Le cache d'instructions a une capacité de 2 Koctets, celui de données a une capacité de 1 Koctets et est à écriture différée ; ils sont tous deux à correspondance directe.
- Trois modes de fonctionnements en consommation réduite sont possibles (*standby*, *suspend*, *hibernate*).
- Il est réalisé en technologie CMOS à 0,35  $\mu\text{m}$  et est cadencé à 33 MHz, fréquence à laquelle il consomme 250 mW (alimenté en 3,3 V).

Le VR4101 comporte en plus du R4100 de nombreuses interfaces d'entrées/sorties :

- interface pour deux canaux série, et un canal pour les signaux infrarouges et audios ;
- un contrôleur d'écrans à cristaux liquides ;
- un contrôleur DMA pour cinq canaux ;
- une interface de conversion analogique/numérique (utile pour la gestion d'écrans tactiles) ;
- une interface pour clavier ;
- un contrôleur PCMCIA<sup>17</sup> ;

Ainsi, bien plus qu'un microprocesseur, le VR4101 est plutôt un micro-contrôleur qui comporte de nombreuses fonctions d'entrées/sorties et est orienté vers une application enfouie spécifique.

Le **VR4102** est un direct successeur du VR4101 en cela qu'il ne lui est pas très différent :

- le cache d'instructions a une taille de 4 Koctets et complète le cache de données d'un Koctets ;
- le bus système est large de 32 bits ;
- des convertisseurs numériques/analogiques et analogiques/numériques ont été ajoutés ;
- la fréquence est de 66 MHz.

Ce microprocesseur devrait remplacer son prédécesseur sur le marché florissant des PDA en apportant de meilleures performances.

---

16. Ce système d'exploitation auparavant connu sous le nom de *Pegasus* est tout spécialement conçu pour les PDA et constitue un sous-ensemble de *Windows 95/NT*.

17. *Personal Computer Memory Card International Association* ou *People Can't Memorise Computer Industry Acronyms*

### 1.3.4 Microprocesseurs MIPS divers

D'autres microprocesseurs MIPS destinés au marché embarqué sont disponibles. Ils constituent des implémentations plus ou moins « exotiques » des architectures conçues par *MIPS* (à l'origine prévues pour des applications générales et des stations de travail), par un de ses partenaires : *LSI Logic*.

Les deux familles de processeurs suivantes sont conçues pour s'intégrer dans des systèmes réalisés à l'aide du programme de conception *CoreWare*. *CoreWare* permet à l'utilisateur de définir son propre système en intégrant des cœurs *LSI Logic*, de la mémoire, et de la logique de contrôle propre sur un seul composant (*system-on-chip design*). Le système ainsi défini est ensuite fondu et testé par le constructeur.

#### La famille MiniRISC

La famille de microprocesseurs MiniRISC est composée de six membres qui donnent à l'utilisateur un large choix dans l'optique *CoreWare*. Ces six produits forment, en fait deux catégories de microprocesseurs. La catégorie CW4001/2/3 met l'accent sur la consommation et le prix. L'autre catégorie, les CW4010/11/20, comporte des microprocesseurs embarqués aux performances nettement supérieures.

Cette famille, créée en 1995 — elle comptait alors 3 microprocesseurs : le CW4001, le CW4010 et le CW4020 — s'est agrandie en août 1996 avec la sortie des trois autres microprocesseurs.

##### Microprocesseurs CW4001/2/3

D'architecture 32 bits, les trois microprocesseurs de cette sous-famille ont une structure assez simple. En effet, ils intègrent un pipeline de seulement trois étages (*fetch*, *execute*, *write-back*), et se programment à l'aide du jeu d'instruction MIPS I (certaines instructions MIPS II peuvent toutefois être exécutées). D'autres instructions sont aussi définies afin de réduire la consommation du composant.

Les quelques informations supplémentaires que nous avons sur ces microprocesseurs se trouvent dans le tableau 1.8 et constituent leurs principales différences.

	Fréquence (MHz)	Dissipation (mW/MHz)	Technologie (CMOS)	Autre
<b>CW4001</b>	60	2	0,5 $\mu\text{m}$	
<b>CW4002</b>	35	2	0,6 $\mu\text{m}$	Cache interne unifié.
<b>CW4003</b>	80	1	0,35 $\mu\text{m}$	Interface <i>FlexLink</i> .

TAB. 1.8 – Les principales différences dans les microprocesseurs CW4001, 4002, et 4003.

On peut noter que ces différences résident principalement dans les aspects technologiques des composants. L'interface *FlexLink* permet une extension au jeu d'instruction MIPS II en fournissant une connexion directe au pipeline.

##### Microprocesseurs CW4010/11/20

Cette famille est la seule à implémenter le jeu d'instructions MIPS II sur des microprocesseurs superscalaires.

Deux pipelines de cinq étages permettent l'exécution de deux instructions par cycle. Une unité de multiplication/division est prévue, ainsi qu'une unité de multiplication/accumulation. Les CW4010/11/20 supportent aussi une instruction de *pre-fetch*, et mettent en œuvre une prédiction de branchement (précise à plus de 90% d'après *LSI*). Des instructions de réduction de consommation sont aussi disponibles.

Les caches internes d'instructions et de données, séparés, sont configurables : les tailles peuvent être choisies de 2 Koctets à 16 Koctets et les structures peuvent être à correspondance directe ou par ensembles à deux voies (un tampon d'écriture est disponible dans le cas de la recopie différée).

Ces processeurs intègrent aussi une unité de gestion de mémoire avec un TLB de 64 entrées.

Les principales différences entre les microprocesseurs de cette catégorie sont listées dans le tableau 1.9.

	Fréquence (MHz)	Dissipation (mW/MHz)	Technologie (CMOS)	32 ou 64 bits	Autre
<b>CW4010</b>	66	5	0,5 $\mu\text{m}$	32	
<b>CW4011</b>	80	3,5	0,35 $\mu\text{m}$	32	Fonctionnalité additionnelle de traitement numérique ( <i>DSP-like</i> ).
<b>CW4020</b>	250	3,5	0,35 $\mu\text{m}$	64	Interface cache et mémoire de 128 bits.

TAB. 1.9 – Les principales différences dans les microprocesseurs CW4010, 4011, et 4020.

### Famille TinyRISC

La famille TinyRISC est la toute dernière famille de microprocesseurs destinés au marché embarqué de *LSI Logic*, et ne compte à ce jour qu'un membre, le *TR4101*.

L'originalité de ce microprocesseur réside dans son jeu d'instructions. En effet, il implémente le nouveau MIPS<sub>16</sub> que nous avons détaillé en partie 1.1.1 (p. 8). Rappelons tout de même que ce nouveau jeu d'instructions conçu pour le marché embarqué permet la réduction de la taille totale du code puisque celui-ci est codé sur 16 bits.

D'autre part, il est aussi toujours possible de programmer les TinyRISC à l'aide du jeu MIPS II, en 32 bits.

De conception dépouillée, le TR4101 comporte un pipeline de trois étages, et ses chemins de données et d'instructions sont unifiés. L'unité de multiplication/division, présente dans les MiniRISC CW40xx est éliminée (elle reste cependant optionnelle). Le TR4101 est alimenté en 3,3 V et est fabriqué en CMOS 0,35  $\mu\text{m}$ , ce qui fait de lui le plus petit microprocesseur en taille actuellement (1,7 mm<sup>2</sup>). Sa consommation est ainsi réduite à 1 mW/MHz, et sa fréquence d'horloge est de 70 MHz en mode 16 bits, et 81 MHz en mode 32 bits.

Deux frères sont prévus pour le TR4101 en 1998, l'un plus rapide, et l'autre plus petit encore. Le TR4120 sera superscalaire de degré deux, et le TR4102 ne devrait mesurer que 1,5 mm<sup>2</sup>. Ces deux microprocesseurs seraient alimentés en 2,5 V ce qui leur permettrait de réduire encore leur consommation.

## Chapitre 2

# Les microprocesseurs Alpha

L'architecture Alpha<sup>1</sup> est l'architecture RISC 64 bits définie par *Digital Equipment Corporation (DEC)* et introduite en 1992. Cette architecture visait à fournir de hautes performances pour les clients VAX, et ainsi les faire perdurer au-delà des années 1990. Les critères du jeu d'instructions sont les suivants :

- hautes performances ;
- longévité (il devait à l'origine durer 25 ans) ;
- capacité d'exécuter les systèmes d'exploitation VMS et UNIX ;
- migration facile à partir des architectures VAX et MIPS.

Bien que ce ne fût pas la première architecture RISC produite par *DEC*, l'architecture Alpha a été la première de *DEC* à entrer sur le marché. Cette architecture est ouverte : tout fondateur peut en acquérir les droits et en fournir sa propre implémentation. À cet effet, *DEC* distingue bien la définition de l'architecture Alpha (jeu d'instructions plus indications de mise en œuvre) de son implémentation représentée par les différentes générations de processeurs.

Depuis peu (l'annonce date de juin 1996), *DEC* a créé une alliance avec le fabricant de semiconducteurs *Samsung Electronic Co.* pour la fabrication, la vente et la distribution de microprocesseurs Alpha, ainsi que le développement de nouveaux microprocesseurs dans cette famille. La firme japonaise *Mitsubishi Electric Co.* constitue quant à elle, une seconde source de microprocesseurs Alpha.

Nous développons brièvement dans une première partie les caractéristiques principales de l'architecture Alpha pour nous étendre sur ses nouveautés depuis la parution de [1], document dans lequel le lecteur trouvera de plus amples renseignements sur les bases de l'architecture. Dans une seconde partie, nous voyons quelles en ont été les implémentations à travers l'étude des microprocesseurs *DEC*.

Une troisième partie introduit le dernier microprocesseur *DEC* Alpha annoncé (octobre 1996), le 21264 (il devrait être disponible fin 1997). Nous réservons une partie à ce microprocesseur car il représente une nouvelle génération de microprocesseurs et intègre des mécanismes atypiques qui méritent une présentation séparée.

Puis, dans une dernière partie nous exposons les perspectives de l'architecture Alpha à travers celles de la firme *DEC*.

### 2.1 L'architecture Alpha

Comme nous l'avons dit précédemment, cette architecture a fait l'objet d'une section dans [1]. Nous rappelons ici les principales caractéristiques, et les complétons par les nouveautés.

---

1. On trouve aussi Alpha *AXP* qui désigne la même chose ; les raisons en sont « obscures et arcanes ».

### 2.1.1 Caractéristiques générales

Alpha est une architecture 64 bits, RISC et de type *load/store*. Toutes les instructions sont codées sur 32 bits, et elles opèrent sur des registres de 64 bits (32 registres entiers, 32 registres flottants). Les types de données supportés sont les suivants :

- les entiers sur 8 bits (octet), 16 bits (mot)<sup>2</sup>, 32 bits (double-mot) et 64 bits (quadruple-mot) ;
- les formats flottants *IEEE-754* à 32 bits (simple), 64 bits (double) et 128 bits (étendu<sup>3</sup>) ;
- les formats flottants VAX 32 bits (*F-floating*) et 64 bits (*G-floating*).

L'architecture Alpha définit cinq formats d'instructions :

- format des instructions mémoire ;
- format des instructions de branchement : branchement relatif au *PC*<sup>4</sup> avec un déplacement de 23 bits (codé sur 21 bits, les deux bits de poids faible étant toujours nuls) ;
- format des instructions arithmétiques entières. Ces instructions peuvent spécifier soit deux, soit trois registres ou une constante de 8 bits étendue à zéro sur 64 bits ;
- format des instructions arithmétiques flottantes ;
- format des instructions du *PALcode* : accès aux routines du *PALcode* (voir plus loin).

Trois modes d'adressage sont supportés : absolu, indirect et basé (le registre R31 est câblé à zéro).

Afin de favoriser et de faciliter l'exécution de plusieurs instructions en parallèle (implémentations superscalaires), certains choix ont été faits pour éliminer toute ressource spéciale ou cachée du processeur. Ainsi l'architecture Alpha n'implémente pas de codes condition, ni de branchement retardé (*branch delay slot*) et la gestion des exceptions arithmétiques est imprécise.

Alpha se veut indépendant des langages de programmation et des systèmes d'exploitation, et supporte :

- *OpenVMS Alpha* ;
- *Digital UNIX* ;
- *Windows NT Alpha*.

Pour cela, un ensemble de routines spécifiques à un système d'exploitation et à une implantation matérielle donnée fournit les primitives nécessaires à sa gestion, c'est le *PALcode*<sup>5</sup>. Un accès logiciel explicite à ces routines est possible par l'intermédiaire des instructions de type *CALL-PAL*.

### 2.1.2 Modifications apportées depuis fin 1995

Le jeu d'instructions a sensiblement évolué, puisque trois extensions majeures ont été faites depuis [1]. Ces extensions sont formées

- d'instructions de manipulation d'octets et de mots ;
- d'instructions *multimédias* ;

---

2. La définition d'un mot comme une quantité de 16 bits est une référence historique à l'architecture VAX, elle-même référence à l'architecture PDP-11. Dans tout ce chapitre, l'appellation de « mot » aura cette signification.

3. Ce format de données est uniquement « supporté », aucune instruction de manipulation n'est encore spécifiée, la gestion reste purement logicielle.

4. *Program Counter*.

5. *Privileged Architecture Library code*.

- d'instructions diverses :
  - comptage des 0 ou des 1 dans un registre (*populations count*) ;
  - déplacements inconditionnels de registres entiers vers flottants ou flottants vers entiers ;
  - racine carrée flottante.

D'autres instructions plus spécifiques ont aussi été ajoutées, elles concernent :

- l'ordre des écritures mémoire (instruction *WMB*) en définissant une *barrière de mémoire* pour les écritures (*Write Memory Barrier*) ;
- le test pour le support des extensions (masque d'architecture, version d'implémentation) ;
- l'amélioration de la performance du cache de données (éviction de bloc, indication d'écriture de 64 octets).

### Manipulation d'octets et de mots

Une des grandes modifications de l'architecture (introduite dans le 21164A en 1996) est la possibilité de lire/écrire des octets ou des mots en mémoire. En effet, ces formats de données étaient à l'origine accédés dans les registres à l'aide d'instructions de manipulation d'octets. Ce choix, justifié à l'époque par les co-architectes (D. Sites et R. Witek) permettait d'éviter :

- de ralentir les accès mémoire en 32 et 64 bits (en évitant le réalignement) ;
- la difficulté liée à la mise en place d'un système de correction d'erreurs au niveau des octets, dans le sous-système mémoire (sur les caches à écriture directe).

Précisons au passage que l'architecture privilégie une très haute fréquence d'horloge (l'ordre de grandeur actuel est 500 MHz), et que tout goulot d'étranglement, en particulier pour les accès mémoire, peut conduire à une horloge plus lente et par là engendrer des pertes de performances considérables.

Ce changement a été motivé en partie par l'entrée de *DEC* sur le marché des PC, à travers le support de *Windows NT*, et la volonté d'émuler du code x86 de manière performante (émulation désormais réalisée par un traducteur-émulateur spécifique, le *FX!32* : voir section 2.4, p. 48)

### Instructions multimédias

L'extension multimédia (fin 1996) a été introduite pour faciliter les applications multimédias et en particulier le codage vidéo *MPEG-2*<sup>6</sup> et porte le nom de *Motion-Video Instructions (MVI)*. En fait cette extension accélère une partie de la compression (celle qui consommait le plus de cycles CPU) en mettant à disposition des utilisateurs quelques nouvelles opérations entières sur des vecteurs d'octets (TAB. 2.1), sans toutefois y intégrer les opérations arithmétiques (additions, multiplications).

Nous n'explicitons pas plus longuement les particularités de ce jeu d'instructions, nous renvoyons le lecteur à [5] pour plus de détails à ce sujet, et à [14] pour une spécification complète de l'architecture Alpha.

---

6. *MPEG* pour *Moving Pictures Experts Group* est un comité qui génère des algorithmes standards (ISO) de compression numérique audio et vidéo. *MPEG-2* est le nom de l'algorithme qui concerne la vidéo.

Instructions	Commentaires
PERR Ra, Rb, Rc	<i>Pixel ERROR</i> : somme des valeurs absolues des différences entre chacun des 8 octets de Ra et Rb dans Rc.
PKxx Rb, Rc	<i>Pack</i> : extrait l'octet de poids faible de chaque mot ou double mot de Rb, et stocke les octets ainsi obtenus, côte à côte dans Rc.
UNPKxx Rb, Rc	Opération inverse de ci-dessus ; les octets « vides » sont mis à 0.
MINxxx Ra, Rb, Rc	Chaque octet (ou mot) de Rc contiendra le plus petit des octets (ou mot) correspondant de Ra ou Rb.
MAXxxx Ra, Rb, Rc	Chaque octet (ou mot) de Rc contiendra le plus grand des octets (ou mot) correspondant de Ra ou Rb.

TAB. 2.1 – *Extension multimédia au jeu d'instructions Alpha.*

## 2.2 Rappels et mises à jour sur les microprocesseurs DEC 21064, 21164 et dérivés

Ces microprocesseurs ont déjà été traités dans [1]. Aussi, nous ne rentrons pas dans le détail, mais mettons en valeur les évolutions qui ont eu lieu depuis. Ces évolutions concernent en général les progrès en matière de technologie (la fréquence en particulier) et actuellement, la tendance de *DEC* à baisser les prix de ses microprocesseurs, voire à en concevoir à moindre coût, pour se positionner sur le marché des PC haut de gamme (Windows NT).

Annoncé en 1992, le **DEC 21064** est le premier microprocesseur à implémenter l'architecture Alpha [3, 15]. Destiné au marché des serveurs et des stations haut de gamme, ce microprocesseur d'abord sorti à la fréquence de 150 MHz, a rapidement atteint les 200 MHz. Bénéficiant de l'avancée technologique, une deuxième version de cette architecture (DEC 21064A<sup>7</sup>) est commercialisée aux fréquences de 200, 233, 275 et 300 MHz (janvier 1996). Deux dérivés ont été annoncés en 1993, le **DEC 21066** et le **DEC 21068** destinés à des applications à moindre coût et/ou embarquées (voir partie 2.2.10, p. 41).

Le **DEC 21164**, annoncé en 1994, constitue la deuxième implémentation de l'architecture Alpha. Ce processeur vise essentiellement le marché des serveurs et des stations de travail haut de gamme. Réalisé en technologie CMOS à 0.5  $\mu\text{m}$ , il a atteint la fréquence de 333 MHz, puis une deuxième version, le DEC 21164A, a été annoncée en octobre 1995 pour le remplacer. La technologie CMOS 0.35  $\mu\text{m}$  lui a permis d'atteindre des fréquences inégalées : 366 MHz, 400 MHz (fréquence désormais abandonnée), 433 MHz et maintenant 500 MHz. Dans la suite, les références faites au DEC 21164A le seront dans la plupart des cas sous le nom de DEC 21164, ou simplement 21164, comme le fait *DEC*, sauf dans les cas où nous mettrons en évidence les différences entre les deux microprocesseurs. Une version « à moindre coût », le **21164PC**, destinée au marché des PC haut de gamme devrait être disponible avant la mi-97 (voir partie 2.2.10, p. 42).

7. Nous ne ferons la distinction entre 21164 et 21164A que dans les cas où nous exposerons leurs différences.

## 2.2.1 Jeu d'instructions

Les microprocesseurs **DEC 21064** et **DEC 21164** utilisent des jeux d'instructions conformes à l'architecture Alpha originale, sans les extensions décrites précédemment. Il faut attendre le **DEC 21164A** fin 1995 pour pouvoir lire/écrire des octets et des mots en mémoire, grâce à l'extension que nous avons décrite partie 2.1.2 (p. 32).

L'extension multimédia n'est pas encore été implémentée, elle le sera pour la première fois avec le DEC 21164PC (voir partie 2.2.10, p. 42).

## 2.2.2 Architecture: vue générale

Le **DEC 21064** est un processeur superscalaire (jusqu'à deux instructions par cycle). Il inclut trois unités fonctionnelles entièrement pipelinées (voir FIG. 2.1) :

- l'unité entière, *Ebox*, chargée d'effectuer les opérations arithmétiques entières ;
- l'unité flottante, *Fbox*, responsable de l'exécution des calculs arithmétiques flottants ;
- l'unité de lecture/écriture, *Abox*, qui gère les accès à la mémoire et effectue les calculs d'adresses.

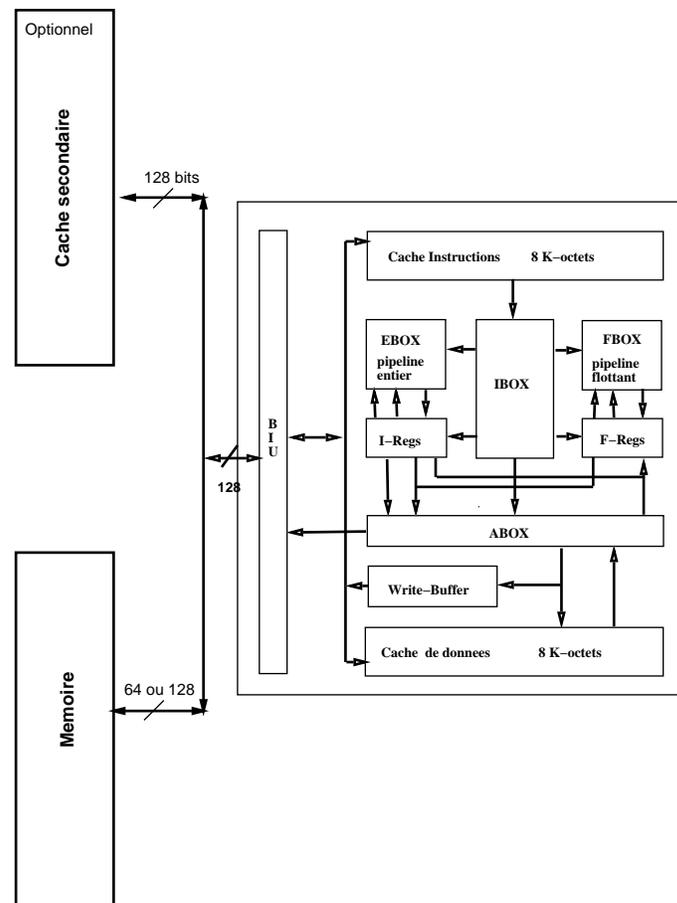


FIG. 2.1 – Architecture du DEC 21064

Par ailleurs, une unité centrale de contrôle, l'*Ibox*, est responsable du chargement des instructions, de leur lancement dans les différentes unités d'exécution, de la prédiction de branchement, de la gestion des conflits de ressources, ainsi que de la gestion des interruptions.

Le premier niveau de cache est intégré sur la puce et est composé d'un cache d'instructions et d'un cache de données séparés. Un second cache externe est optionnel.

Le **DEC 21164** est représentatif des microprocesseurs annoncés fin 1994 à travers une mise en œuvre superscalaire de degré quatre et une haute fréquence d'horloge (266 MHz à l'origine et 500 MHz aujourd'hui). D'autre part, il est un des rares (le premier à sa sortie) microprocesseur à inclure sur le même composant un deuxième niveau de cache (96 Koctets, unifié).

Le pipeline entier compte sept étages, le pipeline flottant neuf. Pour pouvoir exécuter quatre instructions en un cycle, l'architecture du DEC 21164 inclut quatre unités d'exécution (deux entières et deux flottantes, voir figure FIG. 2.2).

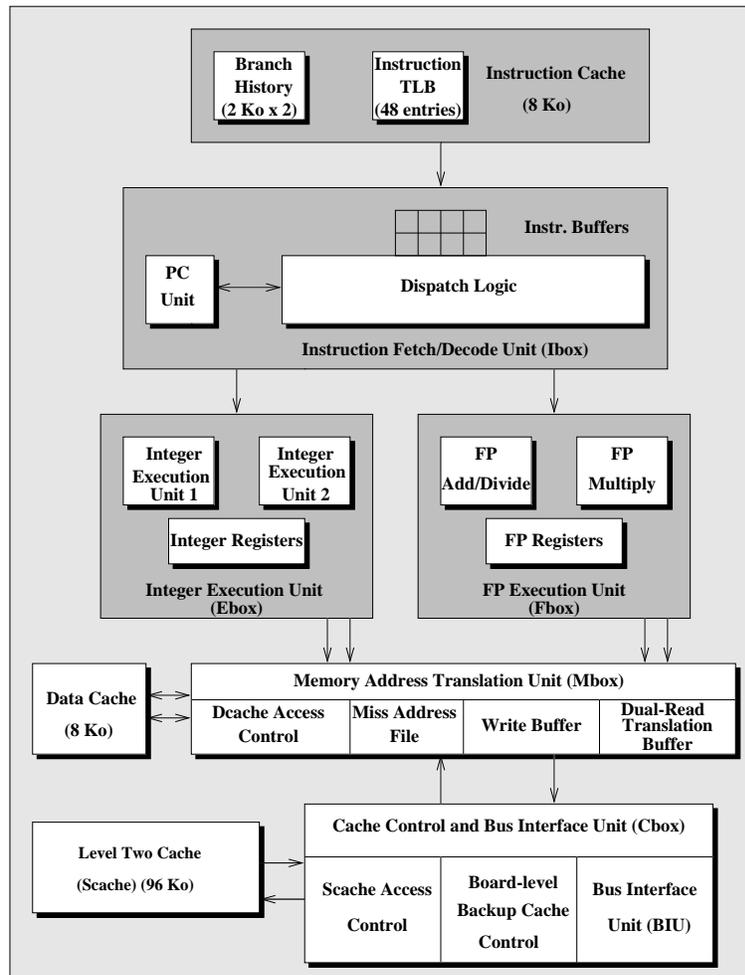


FIG. 2.2 – Architecture du DEC 21164

Le 21164 est constitué des mêmes types d'unités fonctionnelles que le 21064 (*Ibox*, *Ebox*, *Fbox*), l'*Abox* a été renommée en *Mbox* (pour *Memory box*) et qu'une nouvelle unité apparaît ici : la *Cbox* (*Cache Control and Bus Interface Unit*). Cette dernière sert d'interface entre la mémoire et le reste du système et contrôle le cache secondaire (*Scache*) ainsi que le troisième niveau de cache externe optionnel, le *Board-level Backup Cache (Bcache)*.

Les unités *Ebox* et *Fbox* comportent deux unités d'exécution chacune pour assurer, dans le meilleur des cas, le débit de quatre instructions par cycle.

### 2.2.3 Séquencement et exécution des instructions

#### Émission des instructions

L'*Ibox* du **21064** décode deux instructions en parallèle. Le séquencement est toujours fait dans l'ordre; ainsi, si une instruction sur les deux ne peut pas être émise, le séquencement est arrêté en attendant la résolution de l'interblocage.

Un tampon d'instruction à une seule entrée est disponible pour anticiper le chargement des instructions et réduire la pénalité en cas de défaut de cache.

Le DEC 21064 était le premier microprocesseur à mettre en œuvre une prédiction dynamique de branchement à un bit (réalisée dans l'*Ibox*). Cette technique a été abandonnée dans le DEC **21064A** au profit d'une prédiction (plus efficace et plus courante) de branchements à deux bits selon l'algorithme de Smith (voir glossaire p. 152); le lecteur pourra consulter [1] ou [3] pour plus de détails sur l'implémentation de la prédiction de branchement du DEC 21064.

Pour le **21164**, le partitionnement en étages statiques et dynamiques est resté identique, si ce n'est la prise en compte de quatre instructions à chaque cycle au lieu de deux sur le DEC 21064 (le tampon de préchargement d'instructions a été amené à quatre entrées).

La fréquence élevée de l'horloge limite la complexité des actions qui peuvent être réalisées en un cycle. Ainsi, l'émission des instructions est synchrone et se fait dans l'ordre: la partie statique du pipeline (toujours gérée par l'*Ibox*) reste bloquée tant que les quatre instructions chargées n'ont pas été envoyées vers les unités d'exécution.

L'exécution des instructions de branchement conditionnel a été retardée d'un cycle dans le but d'optimiser le débit du pipeline. En effet, cette astuce permet de séquencer au même cycle un calcul de condition et l'instruction de branchement directement dépendante. À chaque branchement, un cycle est systématiquement perdu sur la lecture des instructions. Cependant, l'exécution dans l'ordre des instructions sur le 21164 rend cette perte peu conséquente (voir [1]). L'algorithme de prédiction utilisé est toujours l'algorithme de Smith à deux bits.

#### Exécution des instructions

Le pipeline entier du DEC **21064** comporte sept étages :

- les quatre premiers (dits *statiques* car les instructions peuvent attendre à ce niveau la résolution des conflits de ressources) servent au chargement des instructions, à leur décodage, à la prédiction de branchement ainsi qu'à la résolution des conflits; ils sont communs aux pipelines entier et flottant.
- les trois suivants sont dits *dynamiques* et concernent l'exécution des instructions (qui ne peut être interrompue) dans les unités fonctionnelles correspondantes (*Ebox*, *Abox*).

Le pipeline flottant du DEC 21064 a une profondeur totale de dix étages dont quatre statiques, cinq pour les exécutions et le dernier de mise à jour des registres.

Le pipeline entier du **21164** est assez semblable avec quatre étages statiques et trois étages dynamiques.

Le pipeline flottant comporte quant à lui neuf étages au total, dont les quatre premiers statiques (traitements communs avec les autres instructions). Les cinq autres étages sont légèrement différents de ceux du 21064: le premier concerne l'accès aux registres (sur le 21064 c'était un étage d'exécution), les trois suivants sont réservés à l'exécution et le dernier à la mise à jour des registres.

## 2.2.4 Unités arithmétiques

### Unité arithmétique entière

L'*Ebox* du DEC **21064** comprend les opérateurs classiques de calcul arithmétique (additionneur, décaleur, etc.) et implémente les 32 registres généraux de 64 bits de l'architecture Alpha.

L'*Ebox* du DEC **21164** est composée de deux unités d'exécution entières qui ne sont pas identiques, même si elles gèrent toutes deux les opérations arithmétiques de base (additions, opérations booléennes) :

- l'une s'occupe plus particulièrement des décalages et manipulations d'octets ou de mots<sup>8</sup>, ainsi que des multiplications ;
- l'autre gère les branchements et les opérations d'écriture vers la mémoire.

Des circuits de *bypass* sont implémentés, ce qui permet au résultat de n'importe quelle opération entière d'être disponible au plus tôt pour toute autre instruction consécutive. Seule exception, la multiplication entière (déjà longue de 8 à 16 cycles) qui n'est pas à même de recevoir ses opérandes directement des autres unités fonctionnelles.

### Unité arithmétique flottante

L'unité flottante du DEC **21064** comporte 32 registres de 64 bits entièrement indépendants du banc de registres entiers (pour permettre un accès parallèle aux données entières et flottantes). Elle supporte les deux formats standards de la norme *IEEE 754* (le format étendu de 128 bits doit être géré par logiciel), ainsi que les formats flottants VAX *F-floating* et *G-floating*.

Le banc de registres comporte aussi un registre de contrôle, le *floating point control register*, de 7 bits permettant de spécifier le mode d'arrondi et le mode de contrôle d'exceptions arithmétiques.

Toutes les opérations, mise à part la division, sont entièrement pipelinées et peuvent donc être lancées au débit d'une instruction par cycle. Cependant, il n'est pas possible d'effectuer une autre opération flottante tant qu'une division n'est pas terminée.

L'unité flottante du DEC **21064A** est légèrement différente de celle de son prédécesseur. En effet, la prise en compte de certains *flags* a été améliorée ainsi que l'algorithme de calcul de la division.

Comme l'*Ebox*, la *Fbox* du **21164** est divisée en deux unités d'exécution, complètement différentes dans ce cas :

- une s'occupe des multiplications ;
- l'autre gère les additions (et soustractions), les comparaisons, les branchements et les divisions (qui ne sont pas pipelinées).

Les autres caractéristiques sont semblables à celles du 21064A en ce qui concerne le nombre des registres, les formats de données supportés et le *floating point control register*.

## 2.2.5 Hiérarchie mémoire

### Cache primaire

Le **21064** comporte un premier niveau de caches d'instructions et de données séparés de 8 Koctets chacun, tous deux à correspondance directe, indexés et étiquetés virtuellement. Le cache d'instructions inclut un identificateur de processus de 6 bits. Le cache de données est à écriture simultanée.

8. Sur le 21164A, on peut lire en mémoire des octets ou mots (16 bits) alignés et les utiliser ensuite plus directement.

La taille relativement petite de 8 Koctets des caches du DEC 21064 a été étendue à 16 Koctets sur le **21064A**. Ceux-ci sont adressés virtuellement et étiquetés physiquement. Ils comprennent aussi une protection par parité pour les deux caches de premier niveau. Les autres caractéristiques sont identiques à celles du 21064.

Les caches primaires d'instructions et de données du **21164** sont séparés. Ils ont les mêmes caractéristiques que ceux de son prédécesseur (8 Koctets pour les données et 8 Koctets pour les instructions) mais le champ d'identification de processus est de sept bits (au lieu de six).

### Cache secondaire

Le cache secondaire du DEC **21064** est un cache externe optionnel dont la taille varie de 128 Koctets à 16 Moctets (de 256 Koctets à 16 Moctets pour le **21064A**). Il est à écriture différée. Le DEC 21064 fournit toute la logique de contrôle nécessaire à la mise en œuvre de ce cache. Les données n'ont pas à traverser le microprocesseur pour être inscrites dans ce cache.

L'une des différences majeures avec le DEC 21064 est la présence d'un deuxième niveau de cache interne, mixte, sur le **21164**. D'une capacité de 96 Koctets, associatif par ensembles à trois voies, à écriture différée, ce cache constitue une solution au problème de l'adressage d'un cache primaire à haute fréquence qui impose une petite capacité pour conserver un temps d'accès faible. L'associativité de ce niveau de cache compense les problèmes de conflits du niveau primaire.

### Cache tertiaire

Seul le DEC **21164** prévoit l'adjonction d'un cache de troisième niveau, externe. Sa taille est configurable de 1 à 64 Moctets (par puissances de 2). Ce cache est à correspondance directe, avec une politique d'écriture différée, et sa gestion est assurée par la *Cbox*.

Le **21164A** apporte une petite amélioration au 21164 pour le troisième niveau de cache : il supporte désormais les SRAM synchrones qui permettent des accès plus rapides.

### Remarques

La haute fréquence des microprocesseurs Alpha les confronte au problème de la latence d'accès à la mémoire. C'est pourquoi des mécanismes matériels sont mis en œuvre entre les divers niveaux de cache : le tampon d'instructions qui permet d'anticiper le chargement des instructions en est un exemple.

Sur le **21064**, la largeur du bus de donnée relativement importante (64 ou 128 bits sélectionnable) permet une bande passante soutenue pour les transactions entre les différents niveaux de mémoire (cache primaire ↔ cache secondaire ; cache secondaire ↔ mémoire principale)

Sur le **21164**, des mécanismes mis en œuvre dans les tampons de lecture et d'écriture présents entre chacun des niveaux de cache permettent de regrouper des requêtes vers une même ligne de données afin de les servir simultanément grâce à un bus de 128 bits. Un mécanisme de temporisation permet de conserver un certain temps les requêtes avant de les adresser au sous-système mémoire, ceci afin d'optimiser le mécanisme de regroupement.

## 2.2.6 Support des systèmes d'exploitation

L'architecture Alpha gère la mémoire virtuelle par l'intermédiaire d'une librairie de fonctions appelée *PALcode*. Plusieurs bibliothèques de *PALcode* sont proposées pour mettre en œuvre différents systèmes d'exploitation (*VMS*, *UNIX* et *Windows NT*).

Le mécanisme de traduction des adresses virtuelles en adresses physiques est identique à celui implémenté sur l'architecture MIPS. L'architecture Alpha met en œuvre un espace d'adressage virtuel linéaire paginé.

### Taille des pages

L'architecture Alpha spécifie que la taille des pages doit être constante, de 8, 16, 32 ou 64 Koctets. Le DEC **21064** a des pages de 8 Koctets, le **21164** aussi.

### Espace virtuel

L'architecture Alpha définit une adresse virtuelle sur 64 bits. Cependant, les applications actuelles ne nécessitant pas un tel espace d'adressage, le DEC **21064** n'utilise que 43 bits, les bits de poids fort étant juste validés. L'adresse physique est quant à elle codée sur 34 bits.

L'espace virtuel du **21164** est aussi effectivement codé sur 43 bits ; l'espace d'adressage linéaire physique a été étendu à 40 bits (soit un rapport de 64 avec celui du DEC 21064).

### Cache de traduction d'adresses

Le DEC **21064** possède deux TLB distincts : un TLB d'instructions (12 entrées) et un TLB de données (32 entrées). Le choix de TLB distincts provient de la nécessité de disposer de l'adresse physique à chaque accès aux caches (instructions et données). Ces deux caches de traduction d'adresses sont entièrement associatifs et utilisent la stratégie de remplacement de cache de type NLU (*Not Last Used*). Leur gestion est logicielle et se fait grâce aux fonctions du *PALcode*. Chaque entrée des TLB peut adresser 1, 8, 64 ou 512 pages contiguës et alignées, comme l'indique l'architecture Alpha.

Les TLB du **21164** sont semblables à ceux de son prédécesseur, sauf en ce qui concerne leur taille : 48 entrées pour les instructions et 64 entrées pour les données.

### Protection

Les DEC **21064** et **21164** possèdent les quatre modes d'exploitation suivants :

- le mode *noyau* ;
- le mode *superviseur* ;
- le mode *exécution* ;
- le mode *utilisateur*.

En mode *utilisateur* et *exécution*, l'usage de certaines instructions est interdit. En mode *noyau* et *superviseur*, tout est permis. Les modes *noyau* et *superviseur* sont généralement destinés à gérer le séquençement des processus, la gestion des interruptions, etc. Le système d'exploitation s'exécute dans un de ces modes-ci et contrôle l'ensemble du fonctionnement de la machine. Le *mapping* de la mémoire est différent selon le mode, *noyau* ou *superviseur*. Le mode courant détermine aussi les droits d'accès aux pages (lecture ou lecture/écriture).

Les appels au mode *noyau* se font via les bibliothèques du *PALcode*, servant d'intermédiaire entre le système d'exploitation et le matériel.

## 2.2.7 Support multiprocesseurs

L'architecture Alpha vise les applications hautes performances. Aussi elle permet l'intégration de ses implémentations dans des systèmes multiprocesseurs. De tels systèmes nécessitent la mise en œuvre de protocoles de cohérences de caches et des instructions de mise à jour atomique permettant d'implémenter des primitives de synchronisation. L'architecture Alpha définit également l'ordre des accès à la mémoire dans un système à mémoire partagée.

Nous n'avons pas de données en ce qui concerne l'implantation du support multiprocesseur sur le **21064**. En revanche, sur le **21164**, deux types de protocoles sont disponibles en fonction du coût d'intégration désiré de ce microprocesseur dans le système (protocole de type MESI ou « par éviction de bloc » dans le cas de systèmes d'entrée de gamme).

## 2.2.8 Mesures de performances

Les concepteurs de DEC ont introduit un mécanisme d'analyse de l'exécution des applications grâce à la mise en œuvre d'un moniteur de performances. Ce moniteur, implémenté sur les microprocesseurs DEC **21064** sous la forme de deux compteurs permet de mesurer divers paramètres (émission d'instructions, gel des pipelines, instructions de branchement, lecture/écriture, nombre de cycles exécutés en *PALmode*, etc.) au cours de l'exécution en vue d'une analyse ultérieure.

Le **21164** quant à lui comporte trois compteurs configurables par logiciel pour suivre certains événements.

## 2.2.9 Technologie

Le tableau TAB. 2.2 résume les caractéristiques technologiques des microprocesseurs DEC Alpha, et en montre l'évolution depuis 1992, année de parution du premier DEC Alpha.

	Technologie (CMOS)	Taille (mm <sup>2</sup> )	Nombre de transistors (en millions)	Puissance crête dissipée (en W)	Fréquences (MHz)
<b>21064</b>	0,65 $\mu\text{m}$	234	1,68	30 (à 200 MHz)	150, 166, 200
<b>21064A</b>	0,5 $\mu\text{m}$	164	2,8	33 (à 275 MHz)	200, 233, 275, 300
<b>21164</b>	0,5 $\mu\text{m}$	299	9,3	50 (à 300 MHz)	de 227 à 333
<b>21164A</b>	0,35 $\mu\text{m}$	209	9,3	25 (à 500 MHz)	366, 433, 500

TAB. 2.2 – *Caractéristiques technologiques des microprocesseurs DEC Alpha.*

On peut remarquer les nets progrès réalisés en matière d'intégration et de consommation entre le DEC 21164 et le 21164A. La puissance dissipée par le 21164A correspond à la moitié de celle dissipée par le 21164 à une fréquence nettement plus élevée. Ceci est dû en partie à l'alimentation qui passe de 3,3 V à 2 V (les entrées/sorties restent compatibles aux 3,3 V plus standards).

## 2.2.10 Dérivés

### Dérivés du DEC 21064

Basés sur la même architecture que le 21064, les microprocesseurs **DEC 21066** et **DEC 21068** ont été annoncés en 1993. Le DEC 21066 répond à la demande des utilisateurs sur le marché des ordinateurs personnels et des stations de travail d'entrée de gamme, alors que le DEC 21068 est plus

particulièrement destiné au marché des systèmes embarqués. Une deuxième version, le DEC 21066A, a été annoncée en novembre 1994. Il atteint la fréquence de 233 MHz (08/96).

Ces microprocesseurs reprennent pour l'essentiel l'architecture du DEC 21064, à travers le chargement des instructions, leur émission, la structure des pipelines, la résolution des conflits (prédiction de branchement, etc.), les tables de traduction d'adresses, l'espace d'adressage virtuel et physique (respectivement sur 43 et 34 bits) ainsi que l'organisation des caches (la taille et la structure du premier niveau de caches restent identiques, la capacité du cache secondaire externe peut varier de 128 Koctets à 2 Moctets).

Les principales différences résultent de l'adaptation de ces processeurs aux marchés des systèmes d'entrée de gamme (réduction du coût) et des microprocesseurs enfouis (contrôleur d'entrées/sorties).

### Dérivé du 21164

Développé par *DEC* en collaboration avec *Mitsubishi*, le **21164PC** est une version « à moindre coût » du 21164, destinée au marché des PC haut de gamme. Ce microprocesseur sera fabriqué et vendu par *Mitsubishi* et les systèmes bâtis autour devraient voir le jour fin 1997.

Ce microprocesseur sera le premier à pouvoir exécuter les instructions de l'extension multimédia décrite en 2.1.2 (p. 33). Cette particularité constitue la seule différence avec le cœur du 21164. La compatibilité binaire ascendante sera toutefois assurée, mais pas une compatibilité matérielle vu que le nombre de broches du composant diminue.

Le cache primaire d'instructions sera deux fois plus important que celui du 21164 avec 16 Koctets de capacité alors que le cache primaire de données reste le même (8 Koctets). Le composant ne sera pas plus gros, bien au contraire, puisque le cache secondaire interne du 21164 est éliminé. Il devient donc externe et sa capacité peut aller jusqu'à 4 Moctets.

L'interface système évolue aussi et permet ainsi la réduction des latences d'accès à la mémoire et au cache externe. En effet, elle est composée de deux chemins de données de 128 bits de large chacun : un pour le cache secondaire — cadencé à un quart de la fréquence CPU (133 MHz max.) — et l'autre pour la mémoire principale. Ceci augmente considérablement la bande passante avec le cache secondaire et limite les pertes de performances (par rapport au 21164) dues à la diminution de masse mémoire interne.

L'unité de gestion mémoire (MMU) est restreinte et supporte *Windows NT*, mais pas UNIX ni VMS.

Le même processus de fabrication CMOS à 0,35  $\mu\text{m}$  que le 21164 a été retenu, mais le composant sera plus petit (cache secondaire externe) : 140 mm<sup>2</sup>. Les fréquences disponibles sont 400, 466 et 533 MHz. Enfin, la puissance consommée diminue, le 21164PC devrait consommer au plus 24 W à 533 MHz grâce à une alimentation en 2 V.

## 2.3 Le microprocesseur DEC 21264

Prévu pour la fin de l'année 1997, le 21264 constitue réellement une nouvelle génération de microprocesseurs Alpha. En effet, alors que ses caractéristiques technologiques (fréquence, intégration) ne devraient pas être trop différentes de celles de son prédécesseur le 21164, le 21264 rompt avec la traditionnelle simplicité des implantations Alpha (simplicité forcée à l'origine par la haute fréquence d'horloge caractéristique des architectures Alpha) : il permet l'exécution des instructions « dans le désordre » et de manière spéculative, tout en conservant une fréquence élevée (les premiers 21264 devraient être séquencés à 500 MHz).

Les architectes de *DEC* ont innové en ce qui concerne la prédiction de branchement, les chemins de données et les interfaces système. Le 21264 est, tout comme ses prédécesseurs, un microprocesseur superscalaire (de degré quatre) utilisant aussi un pipeline très profond.

Le 21264 est naturellement orienté vers des applications haut de gamme au vu des hautes performances annoncées et du coût estimé qui n'a sans doute pas été un critère primordial dans les choix des concepteurs [16].

### 2.3.1 Jeu d'instructions

Le jeu d'instructions supporté par le 21264 est conforme à celui spécifié par l'architecture Alpha incluant toutes les extensions (voir partie 2.1.2, p. 32).

### 2.3.2 Architecture : vue générale

Le 21264 est un microprocesseur superscalaire de degré quatre : il peut lire et décoder quatre instructions par cycle. Jusqu'à six instructions peuvent être lancées en même temps dans les unités d'exécution.

Le schéma 2.3 permet de voir quelles sont les différentes unités fonctionnelles. Comme sur la plupart des microprocesseurs superscalaires actuels, nous avons :

- une unité de chargement et de décodage des instructions ;
- deux unités arithmétiques doubles : une flottante, une entière ;
- deux unités de mémoire cache de premier niveau : une pour les instructions (64 Koctets), l'autre pour les données (64 Koctets) ;
- une unité d'interfaçage avec un cache secondaire externe et avec le reste du système.

On peut remarquer l'absence de cache secondaire intégré, mais un large bus dédié permet d'en implanter un à l'extérieur du composant. De plus les caches primaires de données et d'instructions ont une taille importante : ils sont huit fois plus gros que sur le 21164.

### 2.3.3 Séquencement et exécution des instructions

Le 21264 est pipeliné sur sept étages afin de pouvoir être cadencé à haute fréquence. Les instructions contenues dans le cache associé ont fait l'objet, lors de leur chargement, d'un prédécodage qui permet de gagner du temps lors de la phase de décodage.

Une partie commune aux trois pipelines (calcul entier, calcul flottant, accès mémoire) concerne le chargement et le décodage des instructions : quatre instructions alignées sont chargées et décodées en même temps, puis assignées à la file d'instructions entières (20 entrées) ou à la file d'instructions flottantes (15 entrées). Dans chacune de ces files, les instructions dont les opérandes sont disponibles sont arbitrées pour le choix des unités d'exécution (la priorité est donnée à celles qui y sont depuis le plus longtemps). Les autres instructions attendent la disponibilité de leurs opérandes. À ce niveau-là, six instructions peuvent alors entrer en phase exécutive : la file entière comporte quatre sorties, la file flottante deux. Ainsi, la partie statique du pipeline ne reste pas bloquée si une des instructions ne peut pas passer à la phase d'exécution comme c'était le cas dans le 21164.

La figure 2.4 explicite les différents étages des pipelines.

Parmi les nouveautés présentes dans ce microprocesseur, on note l'augmentation de la latence d'accès au cache primaire de données : l'accès au cache lui-même prend presque entièrement un cycle, il faut donc un autre cycle pour acheminer la donnée dans le banc de registres correspondant. Ainsi, la pénalité dite de *load-use*<sup>9</sup> est doublée, mais dans un processeur où l'exécution des instructions se fait dans le désordre, la performance totale est moins affectée — les concepteurs auraient pu : soit réduire

---

9. C'est le nombre de cycles qu'une instruction doit attendre si elle a besoin d'une donnée provenant d'une instruction de lecture mémoire immédiatement précédente

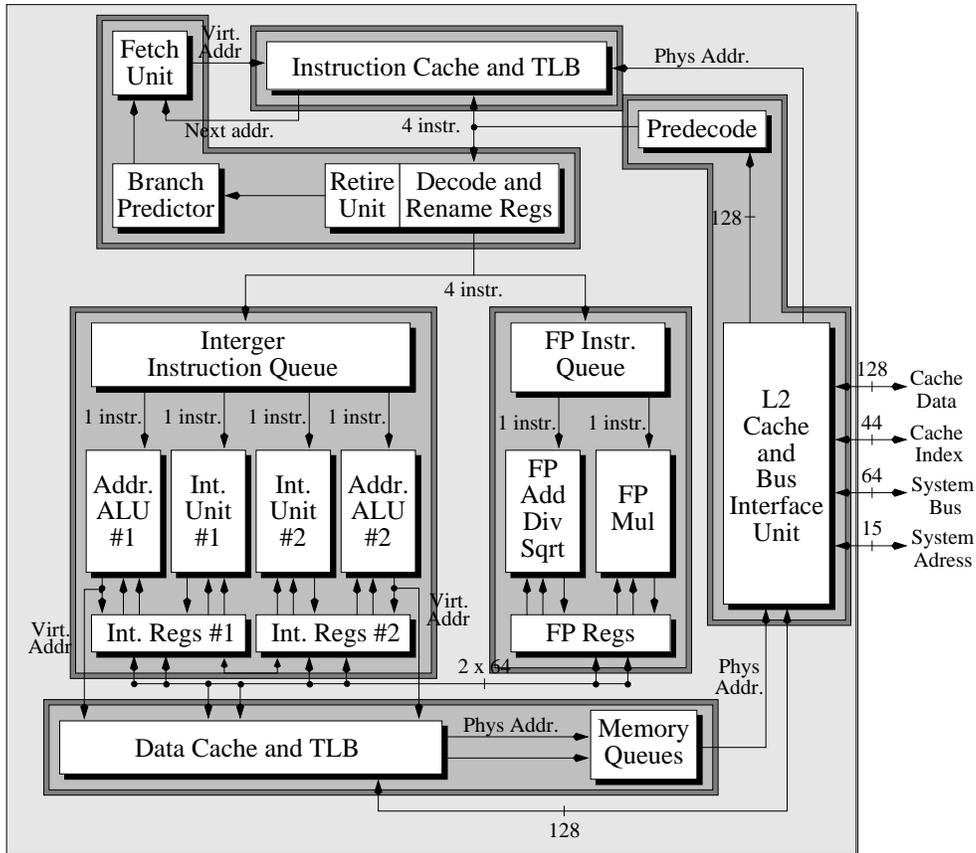


FIG. 2.3 – Schéma du DEC 21264.

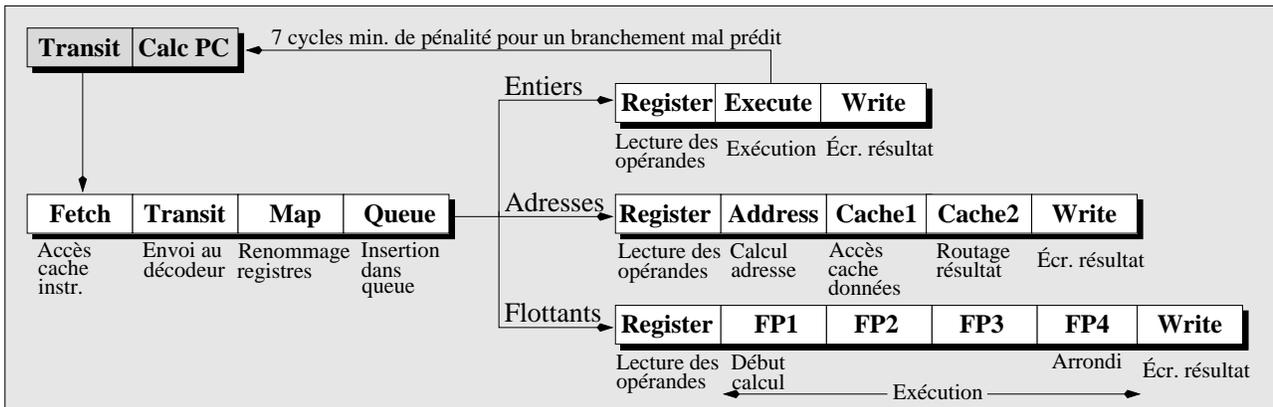


FIG. 2.4 – Les pipelines du DEC 21264.

la taille du cache primaire de données (et ainsi augmenter le taux de défaut de cache), soit réduire la fréquence d'horloge ce qui aurait engendré une perte de performance bien plus importante.

Comme de nombreux autres microprocesseurs récents, le 21264 comporte une pile d'adresses de retour de procédures accédée en même temps que le cache d'instructions. Celle-ci permet une prédiction précise à travers 32 niveaux d'appels de procédures.

### Prédiction de branchement

Une des plus importantes originalités du 21264 réside dans la prédiction de branchement. En effet, comme il se perd au minimum sept cycles par branchement mal prédit (sans compter les quatre à cinq cycles moyens d'attente dans la file entière ou flottante), ce microprocesseur nécessite une prédiction de branchement précise.

La méthode utilisée est celle que Scott Mc Farling avait décrite dans un article en 1993 [17] et qui consiste à combiner deux algorithmes différents et à choisir dynamiquement, pour une prédiction donnée, celui qui est le plus précis (méthode hybride). L'unité de prédiction de branchement comprend ainsi deux tables d'historique (une locale, une globale) et une table pour le choix de la prédiction. Le résultat de la prédiction pour un branchement donné est stocké dans la ligne du cache d'instructions associée à l'instruction concernée.

#### 2.3.4 Détail des unités fonctionnelles de calcul

Les unités fonctionnelles de calcul sont divisées en plusieurs unités d'exécution :

- deux unités arithmétiques entières ;
- deux unités de calcul d'adresses ;
- deux unités arithmétiques flottantes.

#### Unités entières et unités de calcul d'adresses

Les unités entières sont au nombre de quatre : deux unités pour les calculs entiers et deux autres pour les calculs d'adresses.

Les deux unités arithmétiques entières peuvent gérer toutes deux des opérations arithmétiques, des opérations logiques, des décalages de bits et des branchements. Elles ne sont cependant pas identiques car l'une traite les multiplications et l'autre les instructions de l'extension multimédia.

Les deux unités de calcul d'adresses gèrent les accès mémoire de données entières et flottantes mais peuvent aussi exécuter des opérations arithmétiques simples et des opérations logiques sur des entiers. Ainsi une instruction de calcul entier suffisamment simple peut s'exécuter *a priori* sur n'importe quelle unité d'exécution ; cependant, elle est pré-assignée à une unité d'exécution donnée dès son entrée dans la file d'instructions entières. Ceci a pour effet de simplifier l'arbitrage, qui a lieu dans la file, entre les instructions prêtes à l'exécution même si la performance s'en ressent légèrement.

Presque toutes les combinaisons de quatre instructions entières et/ou d'accès à la mémoire sont possibles. Pour cela une implémentation standard serait composée d'un banc de registres unique avec huit ports de lecture et six ports d'écriture. Un allongement du temps de cycle aurait alors été nécessaire à cause de la dilatation physique induite de la largeur du chemin de données.

Les concepteurs du 21264 ont préféré dupliquer le fichier de registres entiers. Chaque copie a ainsi quatre ports d'écriture et six ports de lecture et sert une unité de calcul entier et une unité de calcul d'adresses. Les deux ensembles ainsi formés sont appelés *clusters*.

Les deux fichiers de registres sont synchronisés, mais un cycle supplémentaire est nécessaire à une ALU d'un *cluster* pour écrire sur l'autre fichier de registres à cause de la distance physique existante

et du petit temps de cycle (le *bypass* n'existe qu'à l'intérieur d'un cluster donné). De cette façon, une séquence d'instructions dépendantes s'exécutera plus rapidement sur un même *cluster*.

Le renommage de registres réalisé en phase de décodage masque les fausses dépendances de données. Pour ce faire, 80 registres physiques entiers sont disponibles pour 32 registres logiques et 8 registres cachés réservés au *PALcode*.

### Unités flottantes

Les deux unités flottantes sont organisées de manière plus traditionnelle puisqu'elles se greffent sur un seul et même fichier de registres. Comme les opérations d'accès à la mémoire pour les opérandes flottants sont réalisées dans une des deux unités de calcul d'adresses, le 21264 peut maintenir un flot de quatre instructions flottantes par cycle dans les cas les plus propices.

Les deux unités flottantes ne sont pas identiques : l'une est dédiée aux multiplications, l'autre s'occupe des calculs arithmétiques simples (additions, soustractions, ...), des divisions et des racines carrées.

Toutes les opérations flottantes sont pipelinées, excepté les divisions et les calculs de racine carrée. La plupart des opérations flottantes ont une latence de quatre cycles (voir FIG. 2.4, p. 44), mais une division en double précision prend 16 cycles, et une racine carrée prend 33 cycles (aussi en double précision) ce qui constitue une amélioration par rapport au 21164.

Le renommage de registres a aussi lieu pour les données flottantes : le fichier de registres comporte ainsi 72 registres flottants physiques pour 32 logiques.

### 2.3.5 Hiérarchie mémoire

Une des grandes différences entre le 21264 et le 21164 réside dans la hiérarchie mémoire. En effet, alors que le 21164 comporte de petits caches primaires et un cache secondaire sur le composant, le 21264 est revenu à une conception plus classique des caches en intégrant seulement un niveau de cache, mais en augmentant largement sa capacité.

#### Cache primaire

Le premier niveau de cache est constitué de deux caches (un pour les instructions, l'autre pour les données) de 64 Koctets chacun, associatifs par ensembles à deux voies. L'importante capacité de ces caches permet d'abaisser le taux de défauts, mais augmente le temps d'accès. Ainsi un accès au cache primaire de données prend deux cycles (la haute fréquence d'horloge n'y est pas pour rien non plus). Les deux caches primaires sont non-bloquants.

Le cache de données est dit « double port » c'est-à-dire qu'il peut produire deux résultats indépendant de 64 bits par cycle et permet n'importe quelle combinaison de deux lectures/écritures. Cette unité est pipelinée et entame un nouvel accès tous les demi-cycles. Ainsi, la table des données opère à la fréquence de 1 GHz. Cependant, la table d'étiquettes est, elle, dupliquée.

#### Cache secondaire

Le cache secondaire du 21264 est externe au composant, contrairement au 21164. Il est contrôlé directement par le processeur et sa taille peut varier de 0 à 16 Moctets. Comme le cache primaire, il est non-bloquant. Entre les deux niveaux de cache, une file de huit requêtes en attente d'accès au cache secondaire est implantée.

Plusieurs bus sont mis en œuvre pour atteindre une bande passante énorme de 4 Go/s, dont un de 128 bits exclusivement pour les données (le bus système est séparé, contrairement à celui du 21164) dont la fréquence peut aller jusqu'à 333 MHz — les fractions de fréquences internes pour la fréquence des bus externes sont :  $\frac{1}{4}$ ,  $\frac{1}{3}$ ,  $\frac{1}{2}$  et  $\frac{2}{3}$ .

La figure 2.5 montre comment s'interface le cache secondaire (ainsi que le reste du système).

### 2.3.6 Bus système

Le bus système est séparé du bus de donnée du cache secondaire externe. Comme ce dernier, il peut être cadencé à des fractions de la fréquence du bus interne.

Deux parties séparées apparaissent ici : une pour les données de largeur 64 bits, l'autre pour le contrôle et les adresses. Afin de simplifier la conception, l'interface système met en œuvre une connexion point à point.

Le bus de données se connecte à un composant spécifique d'interface (*Tsunami D-Chip* par exemple) permettant l'accès à de la mémoire RAM à une fréquence plus adaptée. Le bus de contrôle et d'adresses se connecte lui aussi à des composants spécifiques (de type *Tsunami* par exemple) pour accéder à d'autres bus plus standard (type *PCI bus* dans le cas de l'exemple). La figure 2.5 reprend les exemples et montre aussi comment s'interface le cache secondaire.

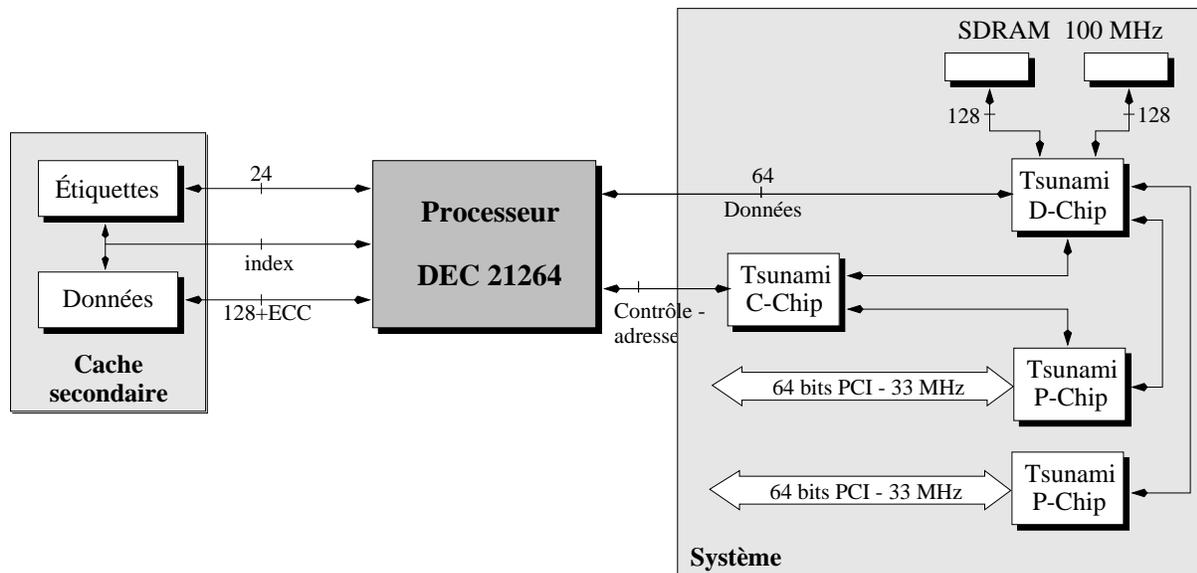


FIG. 2.5 – Interfaçage du DEC 21264 avec le cache secondaire et le reste du système.

Comme sur de nombreux microprocesseurs actuels, le protocole à « transactions imbriquées » (*split-transaction protocol*) est supporté par le bus système du 21264. Ce protocole permet dans ce cas-ci de traiter en même temps jusqu'à 16 références mémoire. Pour plus de détails sur le protocole à transactions imbriquées, voir [6] (p. 103).

### 2.3.7 Support des systèmes d'exploitation

Le support des systèmes d'exploitation est géré par les routines du *PALcode*, comme il est dit dans la spécification de l'architecture Alpha. Ainsi, le support d'un système d'exploitation donné sera assuré par un jeu de routines *PALcode* donné pour une implantation donnée. À titre de rappel, les systèmes d'exploitation actuellement supportés par les microprocesseurs d'architecture Alpha sont :

- *OpenVMS Alpha* ;
- *Digital UNIX* ;
- *Windows NT Alpha*.

Les TLB (instructions et données) sont bien plus importants que sur le 21164 puisqu'ils ont 128 entrées chacun.

Le TLB de données est totalement associatif. Il est de plus dupliqué car il est accédé en même temps que le cache primaire et doit ainsi assurer un haut débit. À chaque accès du cache primaire, le TLB de données est sollicité pour éviter les erreurs éventuelles due à des *alias* mémoire. En effet, les adresses mémoire physiques sont rangées, après la traduction, dans un tampon d'ordonnancement (*reorder buffer*) qui détecte les multiples accès à une même référence mémoire physique et impose, si besoin est, une exécution dans l'ordre aux instructions associées.

### 2.3.8 Support multiprocesseurs

Le support multiprocesseurs n'est pas assuré directement par le composant 21264, étant donné que le bus système mis en œuvre est une connexion point à point. Cependant, les composants spécifiques *Tsunami* sont prévus pour supporter un (unique) second processeur.

## Technologie

La technologie utilisée pour l'implémentation du 21264 n'est guère différente de celle utilisée pour le 21164A : il sera fabriqué en CMOS à 0,35  $\mu\text{m}$ , sur six couches de métal et devrait être cadencé à 500 MHz (puis plus par la suite).

Les différences fondamentales apparaissent dans le nombre de transistors qui est de 15,2 millions (contre 9,6 millions) et dans la taille du composant, conséquence logique, qui devrait être de l'ordre de 298  $\text{mm}^2$  (contre 209  $\text{mm}^2$ ). La puissance dissipée devrait atteindre les 60 W (à 500 MHz).

## 2.4 Perspectives

Les perspectives de cette partie concernent la stratégie commerciale de *DEC* avec son intrusion sur le marché des PC.

Le système d'exploitation *Windows NT* est supporté par l'architecture Alpha, et le récent départ de MIPS et PowerPC de ce marché incite d'autant plus *DEC* à faire concurrence aux processeurs x86.

Pour cela, *DEC* vient de développer un système logiciel gratuit permettant d'exécuter du code *Win32* standard sur les microprocesseurs Alpha : le **FX!32**. En fait, *FX!32* utilise une combinaison d'émulation et de traduction binaire sous-jacente ; deux « passages à exécution » sont nécessaires pour traduire une application en code Alpha : le premier émule complètement l'application et permet d'extraire les informations nécessaires à la traduction (en *tâche de fond* automatique dès la fin de l'exécution) ; le second peut exécuter du code Alpha sur les parties traduites — les parties non traduites à la première exécution sont émulées et traduites par la suite [18].

De cette façon, *DEC* espère à terme que les applications traduites seront seulement 30 % plus lentes que celles portées en code Alpha. Pour l'instant, les meilleures performances l'ont été sur du code entier (benchmark *byte* pour *Windows NT*), et le code traduit a atteint 65 % du code « natif » sur un 21164 à 500 MHz (octobre 1996). Les performances du code flottant sont moindres.

Outre ce logiciel ouvrant les capacités des processeurs x86 aux Alpha, *DEC* réduit fortement les prix de vente de ses 21164 aux fréquences les moins hautes. Au 30 décembre 1996, les prix des 21164 à 433 MHz avaient chuté de 50 %, ceux des modèles à 366 MHz de 48 % et ceux des modèles à 300 MHz de 43 %. Enfin, la disponibilité du 21164PC courant 1997 devrait permettre une politique encore plus agressive.

## Chapitre 3

# Les architectures POWER et PowerPC

L'offre d'*IBM* en matière de microprocesseurs s'articule autour de deux architectures RISC : POWER et PowerPC.

Nous étudions l'architecture POWER dans une première partie, et présentons les deux microprocesseurs principaux qui en constituent l'implémentation.

L'architecture PowerPC est issue d'une alliance scellée en 1991, entre *Apple*, *IBM* et *Motorola* pour contrer, à l'origine, le quasi-monopole détenu par *Intel* sur les processeurs de PC. Ces trois sociétés se sont unies pour définir et mettre au point une nouvelle famille de microprocesseurs RISC fabriqués à la fois par *Motorola* et par *IBM* et destinés aux ordinateurs *Apple* et *IBM* dans un premier temps. Cette alliance s'est ensuite étendue à d'autres constructeurs (*Bull* par exemple).

### 3.1 L'architecture POWER

*IBM* a introduit le premier système RISC 6000 (RS/6000) en février 1990. Les microprocesseurs d'architecture POWER, basés sur une implémentation multi-composants, constituent le cœur des systèmes RS/6000 ; ils ne sont intégrés que sur ces systèmes (qui vont du serveur à la machine parallèle en passant par la station de travail) et ne sont pas vendus en tant que tels.

L'architecture POWER regroupe les principales caractéristiques des microprocesseurs RISC : instructions de taille constante, architecture *load/store*, ensemble de registres généraux entiers et flottants, etc., mais est issue d'une orientation différente des constructeurs de microprocesseurs de l'époque.

L'objectif premier d'une architecture RISC est de privilégier la simplicité de manière à réduire au maximum le temps de cycle. Les concepteurs de POWER ont décidé de minimiser la durée totale d'exécution d'une tâche en incluant dans l'architecture des instructions plus évoluées. Le **POWER1** a été le premier microprocesseur à bénéficier de cette approche [2].

Deux ans plus tard, bénéficiant de l'avancée technologique dans le domaine de l'intégration qui permet de regrouper sur le même composant l'ensemble des unités fonctionnelles du POWER, un microprocesseur visant les systèmes d'entrée de gamme fut introduit : le **RSC**<sup>1</sup>. Ce microprocesseur est une version monocomposant du POWER1 et nous n'y ferons que très peu référence ici.

Le **POWER2** constitue une amélioration de cette architecture grâce à un séquençement superscalaire de degré six ainsi qu'un doublement de la largeur des bus et des unités fonctionnelles par rapport au POWER1.

L'avenir de cette architecture semblait compromis à cause de l'évolution vers l'architecture PowerPC (inspirée de POWER et plus récente). Cependant, *IBM* a introduit fin 1996 un nouveau processeur pour équiper les systèmes RS/6000 : le **POWER2 Super Chip** (P2SC) qui est en fait un POWER2 intégré sur un seul composant (d'où son nom).

---

1. *RISC Single Chip*.

Dans ce qui suit, il est fait un bref rappel sur le jeu d'instructions POWER (on consultera [1, 2] pour plus de détails). Puis nous décrivons le reste de l'architecture à travers les implémentations POWER [2] et POWER2 [5]. Le P2SC est développé plus en détail dans la sous-partie « Technologie des implantations POWER » (p. 55). Il en est aussi question dans la sous-partie « Vue générale [de l'architecture POWER] » (p. 51) où le lecteur peut consulter le schéma du processeur (très proche de celui du POWER2). Cependant, comme ce processeur ressemble beaucoup au POWER2, nous n'en parlons pas dans les autres parties.

### 3.1.1 Le jeu d'instructions POWER

L'architecture POWER est une des rares architectures 32 bits récentes (c'est-à-dire à manipuler des adresses 32 bits). Ainsi, les instructions mais aussi les données sont codées sur 32 bits (sauf les données flottantes qui le sont sur 64).

Le jeu d'instructions POWER est un jeu d'instructions qui se veut RISC. Un des défauts des architectures RISC réside dans la longueur du code dû à la simplicité des instructions inhérente au concept RISC. Pour pallier cela, *IBM* a imaginé des instructions composées assurant une extension du code sans pour autant revenir tout à fait au concept CISC.

#### Jeu d'instruction originel

Le jeu d'instructions POWER compte cinq formats principaux d'instructions (il y en a 11 en tout) :

- type *registre-registre* ;
- type *registre-immédiat* ;
- instructions de branchements conditionnels ;
- sauts inconditionnels et appels de procédures ;
- format flottant.

Les quatre formes courantes d'adressage (absolu, indirect, basé et indexé) et une forme d'adressage avec pré-incrémentation sont supportées.

Les accès à la mémoire se font via des instructions de *load/store*, et les instructions arithmétiques opèrent sur des registres. D'autre part, des instructions plus complexes que dans les jeux d'instructions RISC traditionnels existent, et parmi celles-ci on trouve (à titre d'exemple) :

- une instruction de multiplication-addition flottante ;
- une instruction de branchement conditionnel spéciale qui teste la valeur d'un registre spécial de comptage (32 bits) après en avoir décrémenté la valeur (utile pour les boucles) ;
- des instructions de manipulation de chaînes de caractères.

On notera l'absence d'instructions combinées de test et branchement car une technique différente a été mise en œuvre grâce à l'implémentation d'un registre de code conditions amélioré :

Dans le code opération de chaque instruction un bit assure une mise à jour optionnelle du registre de conditions. Cette mise en œuvre permet au compilateur de judicieusement réorganiser le code. Huit registres de conditions sont disponibles au total.

Cet espace conditionnel est inclus dans l'unité de branchement avec les registres d'adresses de branchement. Il n'est donc pas nécessaire d'accéder aux registres généraux pour évaluer un branchement : ceci permet de les mettre en œuvre sans retard.

## Modifications amenées par l'architecture POWER2

Le jeu d'instructions POWER2 est un sur-ensemble du jeu d'instructions POWER. Les quelques nouvelles instructions concernent surtout les nombres flottants ; l'extension comporte :

- des instructions flottantes d'accès à la mémoire sur des structures de 128 bits (*load quad, store quad*) ;
- une instruction de racine carrée ;
- des instructions de conversion de nombres flottants en entiers.

Pour plus de renseignements sur le jeu d'instructions POWER2, nous invitons le lecteur à consulter [19].

### 3.1.2 L'architecture POWER

#### vue générale

À l'origine, les microprocesseurs POWER sont multicomposants et se répartissent sur neuf (POWER1) ou huit (POWER2) composants. Pour le **POWER1**, les composants sont les suivants :

- une unité entière ( $FXU^2$ ) pour les calculs arithmétiques et logiques, les calculs d'adresses de données entières ou flottantes ainsi que les opérations de lecture/écriture de données entières ;
- une unité flottante ( $FPU$ ) pour les calculs flottants et les opérations de lecture/écriture de données flottantes ;
- une unité de chargement et de séquençement des instructions ( $ICU^3$ ) qui gère le chargement des instructions, leur émission vers les unités entières et flottantes ainsi que l'exécution des instructions de branchement et celles portant sur le registre de conditions ;
- une unité d'interface avec la mémoire principale ( $SCU^4$ ) ;
- quatre composants constituant le cache de données ( $DCU^5$ ) ;
- un contrôleur d'entrées/sorties.

Le contrôleur d'entrées/sorties a été intégré à l'unité d'interface mémoire ( $SCU$ ) sur le **POWER2**. Ceci constitue la principale différence entre POWER1 et POWER2 en ce qui concerne la nature des unités fonctionnelles, les autres composants demeurent les mêmes. Un autre apport important du POWER2 concerne le séquençement de six instructions au lieu de quatre sur le POWER1.

Dans le cas du dernier né de cette architecture, le **P2SC**, tous ces composants se retrouvent sur une seule puce selon le schéma 3.1 ; les unités fonctionnelles sont toujours les mêmes.

On remarquera que ce processeur, qui est un digne représentant de l'architecture POWER2 (il n'est guère différent de son prédécesseur en matière d'organisation interne), comporte de nombreuses unités d'exécution et de larges chemins de données.

---

2. *Fixed point Unit.*

3. *Instruction Cache Unit.*

4. *Storage Control Unit.*

5. *Data Cache Unit.*

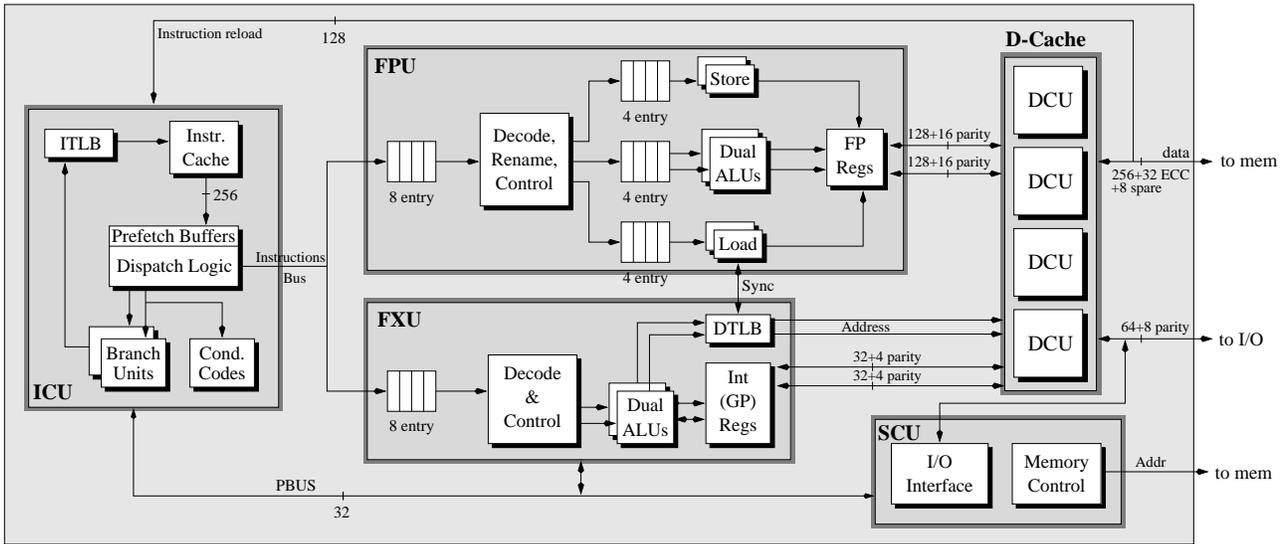


FIG. 3.1 – *Le processeur P2SC d'IBM: synoptique.*

## Pipelines

L'architecture POWER est une architecture résolument superscalaire. Le **POWER1** peut traiter simultanément quatre instructions par cycle: une instruction de branchement, une instruction sur les registres de conditions, une opération entière (ou calcul d'adresse) et une opération flottante. Ses pipelines comportent cinq étages pour les opérations entières, six lors d'opérations sur la mémoire et huit pour les opérations flottantes.

La gestion des dépendances de données est décentralisée vers les unités d'exécution afin d'exploiter au maximum le parallélisme entre les instructions. Aussi, l'*ICU* joue essentiellement un rôle de chargement des instructions et d'émission vers les unités entières et flottantes. Cette unité est également responsable de la gestion des conflits de contrôle. Elle dispose pour cela de deux tampons, l'un de huit entrées destiné au chargement des instructions du flot séquentiel et l'autre de quatre entrées destiné à recevoir les instructions issues du chemin prédit. Cette technique permet à l'*ICU* de disposer des deux chemins de données lors d'un branchement. Les instructions issues du flot séquentiel sont envoyées aux unités de manière conditionnelle. Si une mauvaise prédiction est détectée, ces instructions sont annulées et les instructions issues du tampon cible sont immédiatement émises.

Le **POWER2** peut traiter six instructions par cycle, car les unités arithmétiques ont été dupliquées par rapport à son prédécesseur. Il y a donc possibilité de traiter simultanément une instruction de branchement, une instruction sur les registres de conditions, deux opérations entières (ou calculs d'adresses) et deux opérations flottantes. Ceci a, bien entendu, des conséquences sur la largeur des chemins de données: le bus d'instructions interne situé entre le cache d'instructions et l'unité de *prefetch* a été élargi à 256 bits (huit instructions<sup>6</sup>) dans le cas de ce dernier.

Le POWER2 a conservé les mêmes pipelines que ceux implémentés sur le POWER1 et, de même, les dépendances de données ne sont pas gérés par l'*ICU*. Pour les branchements conditionnels, le même système se retrouve sur le POWER2 avec des tampons de taille plus importante: celui destiné au chargement des instructions du flot séquentiel a 16 entrées, l'autre contenant les instructions du branchement pris en huit.

6. Deux instructions supplémentaires sont inspectées pour la détection des branchements. Si un branchement est détecté dans la fenêtre des huit instructions inspectées et si l'adresse de la cible est disponible, le tampon cible est alors préchargé.

L'une des améliorations apportée par le POWER2 concerne la prise en compte des instructions sur le bus. Les unités entières et flottantes ne prennent sur le bus d'instructions (*Ibus*) que les instructions qui les concernent (sur le POWER1, elles étaient lues puis abandonnées). Pour absorber le flot d'instructions de cette architecture, la taille des tampons implantés sur les unités d'exécution a été systématiquement doublée comme nous le verrons dans la sous-partie suivante.

Sur tous les microprocesseurs d'architecture POWER, les calculs ne sont pas tous effectués par les unités arithmétiques : l'*ICU* se charge d'exécuter les instructions de branchement et celles modifiant les registres de conditions.

À noter, les processeurs POWER utilisent une architecture découplée où les instructions sont exécutées dans l'ordre sur les unités fonctionnelles : l'unité entière peut prendre une certaine avance sur l'unité flottante par exemple. D'autre part, aucune technique de prédiction de branchement dynamique n'est mis en œuvre sur ces deux processeurs. Cependant des mécanismes tels que compteurs de boucle et codes condition rendent de telles techniques moins fondamentales.

## Unités arithmétiques

### Unité arithmétique entière

Dans l'architecture POWER, cette unité joue plusieurs rôles :

- elle est responsable de l'ensemble des calculs arithmétiques entiers ;
- elle effectue les calculs d'adresses (pour les lectures/écritures de données entières mais aussi flottantes) ;
- elle assure la synchronisation avec l'unité flottante lors d'opérations d'accès à la mémoire ou encore lors de l'exécution d'instructions susceptibles de déclencher une exception ;
- elle est chargée de la gestion des dépendances de données entières.

L'unité arithmétique de l'architecture POWER a été considérablement améliorée entre les deux générations de microprocesseurs. Alors que le POWER1 réceptionne les instructions dans un tampon de quatre entrées et exécute une instruction à chaque cycle, le POWER2 dispose d'un tampon de huit entrées et de deux pipelines d'exécution (deux unités d'exécution entières non symétriques sont incluses). Au sein de cette unité, les instructions sont exécutées dans l'ordre, un additionneur à trois entrées permet toutefois d'améliorer les performances en exécutant en parallèle deux instructions dépendantes.

Une unité de multiplication/division permet de réaliser une multiplication en deux cycles (contre 3 à 5 cycles sur le POWER1) et une division en 13-14 cycles (19-23 sur le POWER1).

### Unité arithmétique flottante

L'architecture POWER a intégré un opérateur de multiplication-addition. Cet opérateur est pipeliné et permet de lancer une opération à chaque cycle avec une latence de deux cycles.

Par ailleurs, cette unité intègre un mécanisme de renommage de registres particulièrement utile pour résoudre les fausses dépendances de données (*WAW* et *WAR*). Sur les deux microprocesseurs, les instructions émises par l'*ICU* sont réceptionnées dans un tampon (huit entrées sur le POWER2, quatre sur le POWER1), au rythme de quatre instructions par cycle sur le POWER2 (deux sur le POWER1). Ces instructions sont par la suite décodées et renommées avant d'être envoyées dans des files d'exécution (tampon de décodage de quatre entrées sur le POWER1, file d'instructions arithmétiques de huit entrées sur le POWER2).

Comme l'unité entière, l'unité arithmétique flottante du POWER2 a été considérablement améliorée par rapport à celle du POWER1. Au lieu des 38 registres physiques du POWER1, le POWER2

dispose de 54 registres par rapport aux 32 vus par le programmeur. Il dispose de deux pipelines d'exécution pour chacune des instructions de type lecture/écriture et calcul flottant. Les unités travaillent indépendamment l'une de l'autre, des mécanismes matériels permettent de contrôler les interblocages (les deux pipelines d'exécution pour les instructions de type lecture et écriture sont précédés d'une file commune de six entrées au niveau de laquelle la cohérence des accès est vérifiée).

## Hiérarchie mémoire

L'organisation de la mémoire des microprocesseurs POWER est sensiblement la même, aux capacités des caches près. Le cache d'instructions du POWER2 a une taille de 32 Koctets et il est directement implanté sur l'*ICU* (sur le POWER1, ce cache est externe et a une taille de 8 Koctets). Il est associatif par ensemble à deux voies. Son adressage est physique.

Le cache de données du POWER2 a, quant à lui, une taille de 256 Koctets (quatre fois plus important que celui du POWER1), il est associatif par ensemble à quatre voies et met en œuvre une politique d'écriture différée. L'algorithme de remplacement des lignes utilisé est de type *LRU*. L'adressage de ce cache est physique.

## Support des systèmes d'exploitation

L'architecture POWER définit un espace d'adressage mémoire virtuel de type segmenté paginé. L'adresse virtuelle est composée des 28 bits de poids faible de l'adresse calculée auxquels s'ajoute un numéro de segment. Cette variante par rapport aux autres architectures permet de s'affranchir de la contrainte d'avoir de grands espaces d'adressage contigus, facilitant ainsi le remplacement des blocs de la mémoire principale, tout en alliant les avantages de la segmentation : protection séparée des instructions et des données, partage facilité, manipulation plus simple des structures de données dont la taille varie dynamiquement.

La taille des pages définie par l'architecture POWER est constante et égale à 4 Koctets.

### Espace virtuel

L'espace virtuel défini par l'architecture POWER n'est pas réellement segmenté au sens habituel du terme. L'approche mise en œuvre est une extension de l'espace virtuel paginé qui reste linéaire. Chaque processus ne voit qu'une mémoire virtuelle de 4 Go, découpée en 16 segments de 256 Moctets chacun. Cependant la véritable adresse virtuelle est obtenue par indirection. Les quatre bits de poids fort de l'adresse effective sur 32 bits sont utilisés comme numéro de registre de segment. Ce registre de segment contient 24 bits. L'adresse est alors étendue à  $(32 - 4) + 24 = 52$  bits, ce qui donne un espace virtuel de 4 Peta-octets. Le contenu des registres de segments est géré par le système d'exploitation, ainsi un processus ne peut accéder qu'aux 16 segments qui lui sont alloués sur les  $2^{24}$  segments disponibles. Cette notion d'espace virtuel étendu, empruntée à *Multics*, permet notamment d'intégrer le système de gestion des fichiers dans l'espace virtuel. L'adressage physique du POWER2 est de 32 bits.

L'approche d'*IBM* présente un compromis intéressant pour le partage de données ou de code. Tout en conservant l'espace virtuel linéaire paginé, deux processus partagent du code ou des données en partageant un même segment. Ce code ou ces données ne sont pas forcément à la même adresse virtuelle vue du processus puisque les quatre bits de poids fort de l'adresse de 32 bits peuvent être différents. On peut toutefois reprocher à l'architecture POWER le petit nombre de segments (16) dont un processus dispose.

### Caches de traduction d'adresses

Sur les microprocesseurs POWER, deux caches de traduction d'adresses sont implémentés pour les instructions et les données. Sur le POWER2, les TLB instructions et données ont une taille respective de 128 et 512 entrées (soit quatre fois la taille des TLB du POWER1). Ils sont tous deux associatifs par ensembles à deux voies et leur stratégie de remplacement est de type *LRU*.

## Support multiprocesseur

L'architecture POWER n'offre aucun support matériel pour la mise en œuvre d'une mémoire partagée (à la différence de l'architecture PowerPC). Les microprocesseurs sont cependant utilisés dans les systèmes *Scalable POWERparallel (SP)* qui sont à mémoire distribuée (avec échange de messages et réseau d'interconnexion). Ces systèmes sont utilisés dans le domaine scientifique ainsi que dans des architectures de type *client/serveur*. Notons aussi que désormais les microprocesseurs P2SC remplacent les POWER2 (qui eux-mêmes remplaçaient les POWER1) dans les nœuds de ces machines.

## Technologie des implantations POWER

Cette sous-partie concerne les caractéristiques technologiques des microprocesseurs d'architecture POWER. Nous présentons aussi les particularités du plus récent de cette famille : le P2SC, particularités qui sont essentiellement technologiques.

### Microprocesseurs de type POWER1

Les **POWER1** ont été les premiers à équiper les systèmes RS/6000. Les neuf composants de ce microprocesseur comportent au total 6,8 millions de transistors. Le procédé de fabrication CMOS à  $1\ \mu\text{m}$  a été utilisé, et la fréquence interne est de 62,5 MHz.

Le **RSC** regroupe sur un composant les différentes unités réparties du POWER.

### Microprocesseurs de type POWER2

Les 23 millions de transistors du **POWER2** se répartissent sur huit composants malgré l'utilisation d'une technologie CMOS à  $0,45\ \mu\text{m}$  sur quatre couches de métal, processus de fabrication compétitif en 1993 (date à laquelle ce microprocesseur a été annoncé). Les liaisons inter-composants sont assurées par un conditionnement céramique *MultiChip Module (MCM)* qui permet entre autres de minimiser la surface globale ( $1217\ \text{mm}^2$  au total).

La fréquence d'horloge est relativement faible pour un microprocesseur de cette époque : 66,5 MHz, ce qui n'empêche pas la consommation d'être élevée (aux environs de 65 W) à cause du nombre important de transistors.

Le **P2SC** intègre quant à lui 15 millions de transistors sur un unique composant, total encore jamais réalisé. Le processus de fabrication CMOS à  $0,29\ \mu\text{m}$  a été utilisé mais la taille du composant reste importante :  $335\ \text{mm}^2$ .

La fréquence interne a été largement augmentée par rapport au POWER2 puisqu'elle atteint 135 MHz (elle reste cependant faible comparé aux microprocesseurs actuels de même type). La dissipation thermique maximum de 30 W est assez faible pour un microprocesseur de cette taille ; elle est due à une alimentation en 2,5 V et à la relativement faible fréquence.

La réduction du nombre de transistors du P2SC par rapport à celui du POWER2 provient en grande partie de la réduction de moitié de la taille du cache de données (128 Koctets). Le cache de données permet deux accès provenant du CPU et un accès de la mémoire principale (*cache refill*) par cycle ; pour cela il est cadencé à 500 MHz.

Une autre caractéristique importante du P2SC réside dans son interface système. Celle-ci permet une connexion directe avec 2 Goctets de mémoire DRAM par un bus de 256 bits de large fournissant une bande passante de 2,2 Goctets/s (la fréquence du bus est la moitié de celle du CPU). Les connexions avec les périphériques sont assurées par un bus de 64 bits.

Le tableau 3.1 résume les caractéristiques technologiques des microprocesseurs d'architecture POWER.

	Date	Intégration (CMOS)	Nombre de composants	Nombre total de transistors (en millions)	Taille totale (en mm <sup>2</sup> )	Fréquences internes (en MHz)
<b>POWER1</b>	1990	1 $\mu\text{m}$	9	6,8	—	62,5
<b>POWER2</b>	1993	0,45 $\mu\text{m}$	8	23	1217	66,5
<b>P2SC</b>	1996	0,29 $\mu\text{m}$	1	15	335	135

TAB. 3.1 – *Caractéristiques technologiques des microprocesseurs POWER.*

## 3.2 L'architecture PowerPC

L'architecture PowerPC est issue de l'alliance scellée en septembre 1991 entre *IBM*, *Apple* et *Motorola* qui, à l'origine, devait s'imposer mondialement et s'opposer à la suprématie d'*Intel* sur le marché des PC. Pour cela, le consortium PowerPC a permis à un autre concepteur d'avoir une licence pour faire sa propre implémentation de PowerPC, *Exponential Technologies* avec son  $X^{704}$  (et peut-être bientôt *QED* et son PowerPC 603q).

D'autre part, des alliances sont passées entre le consortium PowerPC et d'autres firmes pour l'intégration et la vente de microprocesseurs PowerPC. Par exemple, *Mitsubishi* a désormais le droit de concevoir et de vendre des microprocesseurs à base de cœurs PowerPC parmi ceux destinés au marché embarqué (les PowerPC 603 et 604 en sont exclus).

Nous introduisons dans une première partie les caractéristiques du jeu d'instructions PowerPC. Puis nous abordons l'étude des microprocesseurs PowerPC existants — ce sont les PowerPC 601, 602, 603, 604, 620, A10 et A30 — et à venir ( $X^{704}$ ). Dans cette liste, la plupart des microprocesseurs ont été étudiés dans [1] et pour cela, nous ne rentrons pas trop dans le détail de ceux-ci, mais nous exposons les évolutions qu'ils ont subies.

Dans une troisième partie, nous présentons les microprocesseurs PowerPC dérivés, s'adressant à des marchés particuliers tels que les systèmes embarqués. Enfin nous faisons un survol des perspectives annoncées pour l'architecture PowerPC à travers les produits prévus pour l'avenir.

### 3.2.1 Caractéristiques de l'architecture PowerPC

Issu de l'architecture POWER, le jeu d'instructions PowerPC résulte de divers choix effectués en fonction de nouvelles exigences :

- permettre une implémentation dans des systèmes la plus large possible : du microcontrôleur au microprocesseur haut de gamme ;
- être doté d'une architecture suffisamment simple pour supporter une haute fréquence de fonctionnement ;
- minimiser les effets qui peuvent freiner la conception d'une architecture superscalaire performante ;
- inclure des fonctionnalités spécifiques à leur intégration dans des systèmes multiprocesseurs ;
- définir une architecture 64 bits assurant une compatibilité binaire avec l'ensemble des applications 32 bits.

#### Jeu d'instructions

L'architecture PowerPC est de type *load/store*. Les instructions sont de taille fixe (32 bits) et alignées en mémoire sur des frontières de mot.

Le jeu d'instructions PowerPC inclut la plupart des instructions du jeu d'instructions POWER à l'exception de celles exécutées peu fréquemment ou jugées trop complexes — celles-ci peuvent être remplacées par des instructions plus simples appartenant aux deux jeux d'instructions. Au total, 36 instructions ont été éliminées. Cependant, pour conserver une compatibilité binaire ascendante, l'architecture PowerPC indique que les instructions exclues doivent générer une interruption permettant ainsi leur émulation par le système d'exploitation.

D'un autre côté, le jeu d'instructions PowerPC inclut de nouvelles fonctionnalités, en particulier :

- instructions de calcul sur 64 bits ;
- instructions de calcul arithmétique flottant simple précision ;
- instructions de conversion de valeurs flottantes en valeurs entières ;
- extension des instructions de lecture/écriture.

La figure 3.2 schématise l'évolution des jeux d'instructions entre les architectures POWER et PowerPC.

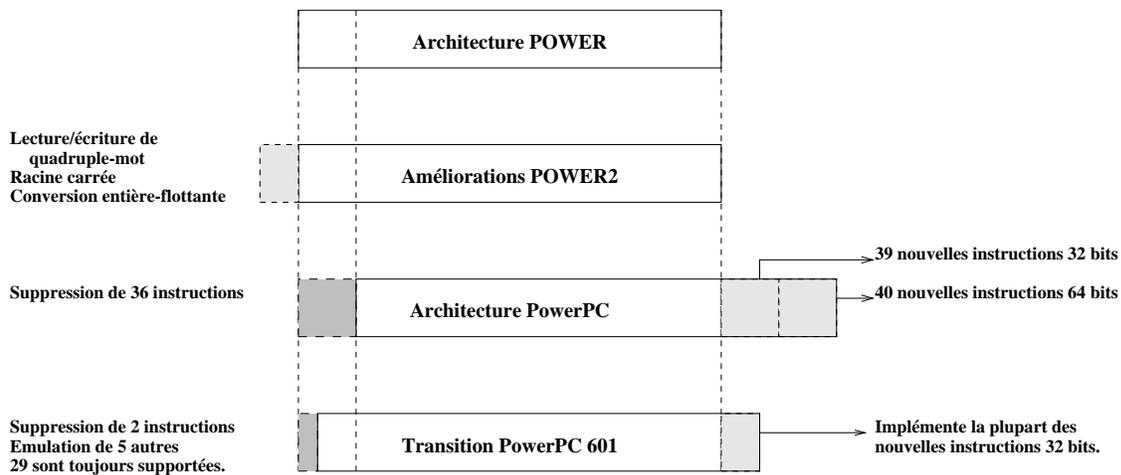


FIG. 3.2 – Relation entre les jeux d'instructions POWER et PowerPC.

## Gestion de la mémoire

L'architecture PowerPC diffère aussi de l'architecture POWER de par le modèle de mémoire et la gestion de celle-ci qui se veut plus flexible.

Des attributs associés à chaque page virtuelle et des instructions propres à l'architecture PowerPC permettent d'assurer plus facilement la cohérence des caches avec la mémoire et la gestion des entrées/sorties.

Deux autres nouvelles instructions constituent un moyen atomique d'accès à la mémoire : tout se passe comme si une lecture et une modification mémoire étaient une opération atomique.

Le rangement des données en mémoire se fait selon l'ordre *big-endian*, comme pour l'architecture POWER, mais aussi selon l'ordre *little-endian* ; le mode est sélectionné par logiciel.

## Systèmes d'exploitation

L'architecture PowerPC utilise un espace d'adressage segmenté paginé. La taille des pages est de 4 Koctets. Cependant, elle introduit un concept de *blocs*. Ces *blocs* peuvent être utilisés comme entité

de base et permettent d'accéder à de plus grands espaces mémoires. Leur taille varie de 128 Koctets à 8 Moctets.

L'architecture PowerPC n'est pas liée à un système d'exploitation particulier. Les microprocesseurs PowerPC supportent actuellement les systèmes d'exploitation suivants :

- Mac OS;
- Solaris;
- Windows NT (le support s'est arrêté fin 1996);
- AIX<sup>7</sup>

### 3.2.2 Les microprocesseurs d'architecture PowerPC

Les microprocesseurs d'architecture PowerPC sont relativement nombreux. Parmi ceux-ci, cinq (PPC 601, 602, 603, 604, 620) ont déjà été détaillés dans [1]. Ainsi, nous résumons leurs caractéristiques, et les mettons à jour en fonction des évolutions qui ont pu avoir lieu depuis (de nouveaux modèles ont vu le jour). Dans une dernière sous-partie, il est question du  $X^{704}$  d'*Exponential Technologies* qui fait preuve d'innovation en matière de technologie.

#### Le PowerPC 601

Sorti en 1993, ce microprocesseur est le premier à mettre en œuvre l'architecture PowerPC (consulter [4] pour plus de détails). Fortement inspiré de l'architecture POWER, il constitue une implantation intermédiaire entre les deux architectures. La plupart des instructions POWER devant être éliminées d'après les spécifications de l'architecture PowerPC sont toujours supportées par matériel. Ainsi, les applications existantes sur le POWER peuvent s'exécuter directement sur le PowerPC 601 sans être recompilées. D'autre part, le PowerPC 601 est un processeur 32 bits : les adresses restent sur 32 bits, les formats de données entières supportés sont de 8, 16 ou 32 bits et les formats de données flottantes supportés sont 32 ou 64 bits.

Le PowerPC 601 est, comme prévu par l'architecture PowerPC, un microprocesseur superscalaire de degré trois — une instruction arithmétique entière ou d'accès mémoire, une instruction arithmétique flottante et une instruction de branchement peuvent s'exécuter en parallèle. Il supporte par ailleurs des mécanismes de séquençement des instructions partiellement dans le désordre<sup>8</sup>. Comme sur le POWER et le POWER2, aucun mécanisme de prédiction de branchement n'est mis en œuvre.

Le 601 comporte un cache primaire unifié de 32 Koctets, associatif par ensembles à huit voies. Son adressage est physique et chaque accès mémoire nécessite le concours de l'unité de gestion de la mémoire (*Memory Management Unit, MMU*) qui intègre un TLB unifié de 256 entrées associatif par ensembles à deux voies. Le TLB et le cache utilisent un algorithme de remplacement de type *LRU*. Rien n'est mis en œuvre pour l'ajout d'un cache secondaire externe.

L'interface système du PowerPC 601 est composée d'un accès à un bus de données de 64 bits et à un bus d'adresses de 32 bits. L'une des différences essentielles avec l'architecture POWER est l'introduction d'un support aux systèmes multiprocesseurs à mémoire partagée. Le protocole de cohérence de cache à invalidation en écriture de type MESI est supporté ainsi que les instructions définies par l'architecture PowerPC permettant des accès mémoire atomiques.

Deux implantations de ce processeur ont vu le jour depuis sa conception. Le PowerPC 601 utilise une technologie CMOS à 0,6  $\mu\text{m}$  et est aujourd'hui cadencé à 80 MHz. Son remplaçant, le PowerPC 601v (ou 601+) utilise une technologie CMOS à 0,5  $\mu\text{m}$  et est cadencé à 110 MHz. L'alimentation de ce dernier est de 2,5 V au lieu de 3,6 V pour son prédécesseur et permet de réduire

7. *Advanced Interactive eXecutive*.

8. L'émission en désordre des instructions se fait ici au niveau de l'étage de décodage et non de celui d'exécution, comme c'est le cas sur le PowerPC 604 par exemple.

la consommation (passage de 8 à 4 W) malgré l'augmentation de fréquence. Le tableau 3.2 (p. 64) récapitule les données technologiques des différents microprocesseurs d'architecture PowerPC.

### Le PowerPC 603

Ce microprocesseur comporte trois versions qui sont, dans l'ordre chronologique, le 603, le 603e et le 603q. Les deux premiers sont conçus par *IBM* et *Motorola*, et le dernier par *QED*. Dans ce qui suit, nous détaillons brièvement le 603, et nous signalons les modifications apportées par les autres versions.

#### Le PowerPC 603

Annoncé en 1993, il est à l'origine destiné au marché des systèmes portables. L'effort a essentiellement porté sur la faible consommation, et la mise en veille de certains organes (processeur, disque, unités d'exécution). Ce microprocesseur a cinq unités d'exécution (entière, flottante, branchement, lecture/écriture et système). Il peut séquencer trois instructions par cycle (deux instructions arithmétiques ou accès mémoire et une de branchement). Les pipelines d'exécution (entier, flottant, branchement, accès à la mémoire) sont identiques à ceux du 601. Il est à noter cependant que ce microprocesseur inclut un mécanisme de renommage de registres pour les instructions entières et flottantes et qu'il implémente trois nouvelles instructions flottantes destinées à améliorer les performances graphiques.

À la différence du PowerPC 601, le 603 implémente des caches d'instructions et de données séparés de 8 Koctets. Ces deux caches sont associatifs par ensembles à deux voies, physiquement adressés. L'algorithme de remplacement des lignes est de type *LRU*.

Un protocole de cohérence à trois états, sous-ensemble du protocole MESI, est utilisé pour maintenir la cohérence lors de transferts de type DMA, mais ce système ne sert pas de support multiprocesseurs (à mémoire partagée).

Deux caches de traduction d'adresses de 64 entrées, associatifs par ensembles à deux voies pour les instructions et les données sont intégrés ainsi qu'un TLB de quatre entrées totalement associatif destiné à la traduction des adresses de blocs.

La largeur du bus de données de ce microprocesseur est de 32 ou 64 bits (au choix) et le bus d'adresses est toujours de 32 bits. Sa consommation typique est de 2,5 W. En plus de l'inhibition de certaines unités, ce microprocesseur dispose de trois modes de fonctionnement configurables par logiciel et qui permettent de mieux gérer sa consommation.

Le PowerPC 603 est cadencé aux fréquences de 66, 80 et 100 MHz (technologie CMOS à 0,5  $\mu\text{m}$ ).

#### Le PowerPC 603e

Le PowerPC 603 souffrait d'un manque de performances dû à la faible taille des caches (au total la moitié du cache unifié du 601). De ce fait *IBM*, *Motorola* et *Apple* ont annoncé une évolution de ce composant : le PowerPC 603e. Ce dernier conserve la même architecture avec des caches de taille double (16 Koctets), associatifs par ensembles à quatre voies et une fréquence d'horloge allant jusqu'à 200 MHz en technologie CMOS à 0,5  $\mu\text{m}$

#### Le PowerPC 603ev

Ce microprocesseur est une version technologiquement améliorée du 603e. Sa fréquence d'horloge va jusqu'à 240 MHz en technologie CMOS à 0,35  $\mu\text{m}$ . Comme ce microprocesseur n'est en rien différent, au point de vue logique, du 603e, la distinction n'est pas toujours faite entre les deux. D'autre part, on le trouve aussi sous le nom de PowerPC 603p.

Une nouvelle version proposée par *Motorola* est sortie en avril 1997. Celle-ci porte le nom de code **Goldeneye** et est fabriquée en technologie hybride comportant pour la première fois des transistors COMS à 0,25  $\mu\text{m}$ . La taille du composant tombe à 42 mm<sup>2</sup> alors que la fréquence grimpe à 300 MHz.

### Le PowerPC 603q

Le PowerPC 603q est le premier microprocesseur d'architecture PowerPC conçu par *QED*. Le but avoué du concepteur a été de fournir un microprocesseur aux performances comparables aux autres PowerPC 603 et à un prix moindre afin de l'introduire dans un premier temps sur le marché de la bureautique, puis sur celui des systèmes embarqués haut de gamme.

Le 603q est un PowerPC simplifié puisqu'il n'est pas superscalaire. Son unique pipeline est constitué des cinq étages classiques. Son cache de données est le même que celui du 603 (8 Koctets, associatif par ensembles à deux voies). Le TLB implanté est unifié, associatif par ensembles à deux voies et a une capacité de 64 entrées. De plus, le bus système est identique à celui du PowerPC 602 (multiplexé).

La simplicité de ce microprocesseur permet entre autres au composant d'être plus petit (69 mm<sup>2</sup>) que le 603, et permet une consommation moindre (1,6 W à 160MHz au lieu de 3,5 W pour le 603e à 120 MHz).

Cependant, ce microprocesseur a uniquement été annoncé : et il ne sortira peut-être pas car *QED* n'a pas encore acquis une licence de fabrication PowerPC.

### **Le PowerPC 602**

Ce microprocesseur, disponible depuis fin 1995, est un dérivé du PowerPC 603 (comme son nom ne l'indique pas!). Il vise le marché des microprocesseurs embarqués haut de gamme tels que les consoles de jeux, les *set-top boxes* et les PDA. Il constitue un compromis entre les performances et le coût de fabrication qui a été divisé par trois par rapport au PowerPC 603.

Les changements par rapport au 603 concernent :

- la réduction de moitié de la taille des caches instructions et données — ils font 4 Koctets chacun — qui deviennent non-bloquants (suppression des tampons de lecture/écriture) ;
- la suppression du séquençement superscalaire (sous certaines conditions, ce microprocesseur peut exécuter deux instructions grâce à la technique du *Branch folding* conservée du 603 et du 601 [1, 4]) ;
- la simplification des bus système en un bus unique multiplexé ;
- la réduction de la latence pour les multiplications entières.

La baisse du nombre total de transistors et la plus petite fréquence interne de ce microprocesseur par rapport au 603 permettent de réduire sa consommation à 1,2 W ; les autres données technologiques restent identiques au 603.

### **Le PowerPC 604**

Tout comme le 603, deux versions sont disponibles : les PowerPC 604 et 604e. Nous présentons brièvement le 604, puis les modifications apportées dans le 604e.

#### Le PowerPC 604

Annoncé au premier trimestre 1994, les premiers systèmes à base de PowerPC 604 ont vu le jour en juin 1995. Ce microprocesseur vise le marché des stations de travail (IBM) ainsi que les PC.

De nombreux aspects du 603 et du 601 auxquels il succède ont été améliorés ; ils concernent :

- le séquençement superscalaire de quatre instructions par cycle au lieu de trois pour ses prédécesseurs (intégration d'une seconde unité arithmétique entière) ;
- l'utilisation de nouvelles méthodes de gestion du séquençement comme
  - l'*exécution dans le désordre* et la mise en place de *stations de réservation* (de deux entrées) pour chaque unité d'exécution ;

- une *prédiction dynamique de branchement* selon l'algorithme à deux bits de Smith ;
- le *renommage de registres* qui ajoute un étage au pipeline « classique » ;
- la structure des caches qui est identique à celle de ceux implantés sur le 603e (16 Koctets pour les données, 16 Koctets pour les instructions) ;
- la gestion de la mémoire avec deux TLB pour les instructions et les données de 128 entrées chacun, associatifs par ensembles à deux voies ;
- le support multiprocesseurs avec le support du protocole de cohérence de caches de type MESI et la duplication du port d'accès aux étiquettes ;

Pour le séquençement des instructions, le 603 et le 604 ont des caractéristiques similaires (excepté celles exposées plus haut) dans la mesure où ils ont été développés en parallèle. D'autre part, la technologie CMOS à 0,5  $\mu\text{m}$  est aussi utilisée sur le 604, mais ses 3,6 millions de transistors font que sa taille est plus du double de celle du 603 (197  $\text{mm}^2$  ; voir TAB. 3.2, p. 64).

### Le PowerPC 604e

Les améliorations apportées par le PowerPC 604e par rapport au 604 sont du même ordre que celles amenées par le 603e sur le 603. En effet, les caches du 604 ont été doublés sur le 604e pour atteindre maintenant 32 Koctets chacun. D'autre part la technologie utilisée est désormais CMOS à 0,35  $\mu\text{m}$ , ce qui permet une augmentation considérable de la fréquence interne. Celle-ci atteint 225 MHz depuis juillet 1996. La taille du composant est de 148  $\text{mm}^2$ .

Une nouvelle version proposée par *IBM* est sortie en avril 1997, son nom de code est **Helmwind**. Le procédé de fabrication CMOS à 0,27  $\mu\text{m}$  lui permet d'atteindre 233 MHz. Le taille du composant se trouve ainsi réduite à 96  $\text{mm}^2$ .

### **Le PowerPC 620**

Annoncé à la fin de l'année 1994, le PowerPC 620 est destiné à un marché nécessitant des performances élevées telles que les stations de travail haut de gamme et le domaine des serveurs. Aussi, ce microprocesseur est le premier à mettre en œuvre une architecture PowerPC 64 bits, qui reste cependant compatible avec l'ensemble de l'offre 32 bits PowerPC. De nombreux problèmes étant intervenus lors de sa fabrication, ce microprocesseur vient à peine (avril 1996) de voir naître quelques échantillons et ne semble toujours pas disponible.

On retrouve au sein du 620 des caractéristiques déjà mises en œuvre sur le PowerPC 604. En revanche, l'accroissement de certaines fonctionnalités telles que les unités fonctionnelles, la hiérarchie mémoire, la largeur des bus et autres contribue à l'augmentation des performances.

Le PowerPC 620 a une architecture superscalaire de degré quatre (c'est-à-dire que quatre instructions sont chargées et décodées en parallèle). Six unités fonctionnelles sont présentes : trois unités d'exécution entière (deux pour les opérations simples et une pour les opérations complexes), une unité de lecture/écriture mémoire, une unité de branchement et une unité flottante.

L'exécution des instructions peut être effectuée dans le désordre grâce aux files d'instructions précédant chacune des unités (structure identique au PowerPC 604). Le pipeline a une structure classique (cinq étages). Quatre techniques sont employées pour la prédiction dynamique des branchements (voir [1]).

Le PowerPC 620 met en œuvre deux caches d'instructions et de données séparés, de 32 Koctets chacun, associatifs par ensembles à huit voies. Ces caches sont tous deux physiquement adressés. La politique de mise à jour des données de la mémoire est configurable (simultanée ou différée). Par ailleurs, le PowerPC 620 inclut un contrôleur de cache interne qui lui permet de rajouter un second niveau de cache de 1 à 128 Moctets. Le bus de données vers ce cache a une largeur de 128 bits. Le protocole de cohérence de cache de type MESI est utilisé sur ces deux niveaux de cache pour assurer l'intégration du 620 dans les systèmes multiprocesseurs à mémoire partagée.

L'interface système est elle aussi orientée vers les besoins des systèmes multiprocesseurs car elle met en œuvre un bus d'adresses de 40 bits et un bus de données configurable (à l'initialisation) en 32 ou 64 bits.

La technologie initialement prévue (CMOS à 0,5  $\mu\text{m}$ ) a été changée pour la sortie récente (avril 1996) des échantillons en CMOS à 0,35  $\mu\text{m}$ . La fréquence interne annoncée est 200 MHz.

## Le microprocesseur X<sup>704</sup>

*Exponential Technologies*, firme fondée en 1993, a annoncé fin 1996 son premier produit, le X<sup>704</sup>, qui devrait sortir à la mi-1997. Ce microprocesseur est destiné au marché des PC haut de gamme et se place, grâce à ses performances (comparables à celles du DEC 21164 pour le calcul entier), en haut de la gamme PowerPC (il est cadencé à 533 MHz et utilise une technologie BiCMOS<sup>9</sup> particulière).

Le X<sup>704</sup> est un microprocesseur d'architecture PowerPC, superscalaire de degré trois; il peut séquencer en parallèle une instruction de lecture/écriture mémoire, une instruction de calcul arithmétique (entier ou flottant) et une instruction de branchement<sup>10</sup>. Les instructions sont traitées dans l'ordre dans un pipeline typique profond de six étages. Une prédiction de branchement est mise en place et permet de ne pas avoir de retard dans le cas d'un branchement pris, correctement prédit (d'après *Exponential* la prédiction est précise à 75 %). Celle-ci prend la forme de deux bits d'historique pour chaque double-mot du cache d'instructions, avec une adresse cible.

Deux niveaux de cache sont intégrés sur le composant : deux caches primaires de données et d'instructions de 2 Koctets chacun et un cache secondaire unifié de 32 Koctets — la petite taille de ces caches a été imposé par la haute fréquence d'horloge privilégiée. Les caches de premier niveau sont à correspondance directe avec la mémoire tandis que le cache secondaire est associatif par ensembles à huit voies. Le cache primaire de données met en œuvre une politique d'écriture simultanée; celle du cache secondaire est configurable. L'interface bus du X<sup>704</sup> est compatible avec les bus système utilisés sur les PowerPC 60x.

Un des intérêts du X<sup>704</sup> réside dans ses aspects technologiques. En effet, *Exponential* a utilisé des transistors bipolaires, plus rapides que les CMOS, pour la partie logique du microprocesseur, et des transistors CMOS, d'intégration plus importante que les transistors bipolaires, pour la partie mémoire (caches). Ainsi, sur les 2,7 millions de transistors intégrés à ce processeur, environs 700 000 sont bipolaires. Le gros désavantage d'une telle technologie est la puissance dissipée puisque la consommation s'élève à 85 W.

Pour plus d'informations au sujet du X<sup>704</sup>, le lecteur peut consulter [20].

*Nous avons appris en dernière minute que Exponential Technologies a déposé son bilan (mai 1997).*

## Divers : PowerPC AS

Pour renouveler sa gamme de serveurs AS/400, *IBM* a développé à la mi-1995 deux microprocesseurs PowerPC AS (AS pour *Advanced Series*). Ils sont plus généralement appelés A10 et A30. Ceux-ci, tout comme les processeurs d'architecture POWER, ne sont pas vendus seuls mais uniquement dans les systèmes AS/400.

Ces deux microprocesseurs implémentent le jeu d'instructions PowerPC 64 bits complété par quelques nouvelles instructions et un nouveau mode d'adressage pour permettre une migration plus

---

9. *Bipolar-CMOS*

10. Celle-ci doit être la dernière instruction du groupe de trois pour qu'elle puisse être exécutée en parallèle avec les deux autres instructions.

souple depuis les anciens microprocesseurs équipant les systèmes AS/400 qui étaient d'architecture CISC.

L'**A30** est superscalaire de degré quatre et sa structure ressemble à celle des microprocesseurs POWER2 (voir FIG. 3.1, p. 52), même si les circuits sont complètement différents. L'unité centrale intègre un cache d'instructions de 8 Koctets. Le cache de données de 256 Koctets est composé de quatre circuits, et peut être accédé en un cycle. À noter que ce microprocesseur n'implémente pas de mécanismes de prédiction de branchement, mais que les deux branches d'un branchement sont émises vers les unités d'exécution. L'A30 destiné aux machines haut de gamme est composé de sept circuits au total, regroupés dans un boîtier MCM, et est réalisé en technologie BiCMOS 0.72  $\mu\text{m}$ . Il utilise une fréquence d'horloge de 154 MHz.

L'**A10**, destiné aux AS/400 d'entrée de gamme, est composé d'un seul circuit. L'A10 est un microprocesseur superscalaire de degré trois intégrant un cache d'instructions de 4 Koctets et un cache de données de 8 Koctets. Ce microprocesseur est réalisé en technologie CMOS à 0.65  $\mu\text{m}$ , et atteint la fréquence de 77 MHz.

Ces deux microprocesseurs ne semblent pas avoir évolué depuis leur sortie en 1995 car aucune information n'est diffusée à leur sujet, bien que *IBM* ait initialement prévu des améliorations technologiques, notamment en ce qui concerne l'intégration.

## Technologies des microprocesseurs PowerPC

Le tableau 3.2 récapitule les données technologiques des différents microprocesseurs d'architecture PowerPC.

### 3.2.3 Les micro-contrôleurs d'architecture PowerPC

Depuis la parution de [1], de nombreux microprocesseurs embarqués fondés sur l'architecture PowerPC ont été introduits. Ceci est dû au changement de cap des applications de l'architecture PowerPC qui ne s'est pas affirmée comme un véritable concurrent d'*Intel* sur le marché des PC. Pour ce secteur, *Motorola* et *IBM* développent leurs produits séparément.

Comme dans les autres parties, nous ne nous attardons pas sur les processeurs qui ont été décrits dans [1] pour céder la place aux nouveaux venus.

## Produits Motorola

*Motorola* a annoncé quatre micro-contrôleurs d'architecture PowerPC depuis le premier MPC 505 : le MPC 801, le MPC 821, le MPC 823 et le MPC 860.

### Le MPC 505

Le MPC 505 est le premier microprocesseur embarqué d'architecture PowerPC 32 bits développé par *Motorola*. Il est sorti en 1994. L'architecture PowerPC a été simplifiée ici afin de diminuer les coûts de production et le MPC 505 ne peut émettre qu'une instruction par cycle. Trois unités fonctionnelles sont présentes : une pour les calculs arithmétiques entiers, une autre pour les calculs arithmétiques flottants et une dernière pour les accès mémoire.

Le MPC 505 dispose d'un cache d'instructions et de données de 4 Koctets. Par souci de simplification de l'architecture, il n'intègre pas d'unité de gestion de la mémoire. Par contre, il inclut des caractéristiques spécifiques au domaine des microprocesseurs embarqués : gestion dynamique de la consommation, modes de faible consommation, intégration de contrôleurs d'entrées/sorties de type DMA, interruption, etc., de manière à simplifier au maximum son intégration dans les systèmes.

Le MPC 505 comporte 585 000 transistors et est cadencé à 40 MHz.

	Date	Intégration (CMOS)	Nombre total de transistors (en millions)	Taille totale (en mm <sup>2</sup> )	Fréquences internes (en MHz)
<b>PowerPC 601</b>	1993	0,6 $\mu\text{m}$	2,8	120	80
<b>PowerPC 601v</b>	1994	0,5 $\mu\text{m}$	2,8	74	110
<b>PowerPC 602</b>	1995	0,5 $\mu\text{m}$	1,0	50	66
<b>PowerPC 603</b>	1994	0,5 $\mu\text{m}$	1,6	83	66, 80, 100
<b>PowerPC 603e</b>	1995	0,5 $\mu\text{m}$	2,6	98	100, 120, 133, 150, 160, 166, 180, 200
<b>PowerPC 603ev</b>	1996	0,35 $\mu\text{m}$	2,6	81	160, 166, 180, 200, 225, 240
<b>PowerPC 603q</b>	??	0,5 $\mu\text{m}$	??	69	120, 160
<b>PowerPC 604</b>	1994	0,5 $\mu\text{m}$	3,6	197	100, 120
	1995				133, 150
	1996				166, 180
<b>PowerPC 604e</b>	1996	0,35 $\mu\text{m}$	5,1	148	150, 166, 180, 200, 225
<b>PowerPC 620</b>	1994- 97	0,35 $\mu\text{m}$	$\sim 7$	311 <sup>†</sup>	200
<b>X<sup>704</sup></b>	1996	0,5 $\mu\text{m}^*$	2,7	150	533
<b>A30</b>	1995	0,72 $\mu\text{m}^*$	—	447	154
<b>A10</b>	1995	0,65 $\mu\text{m}$	—	213	77

TAB. 3.2 – Récapitulatif des données technologiques des microprocesseurs PowerPC.

<sup>†</sup>. Cette taille est uniquement valable pour le processus de fabrication CMOS à 0,5  $\mu\text{m}$ . La donnée cohérente avec les autres paramètres indiqués n'est pas encore disponible.

\*. Cette valeur est valable pour la partie CMOS du microprocesseur.

Le MPC 860

Le MPC 860, sorti fin 1995, est le micro-contrôleur de référence de la gamme. Le microprocesseur au cœur du composant est un PowerPC qui est identique à celui du MPC 505 [1] avec en plus une unité de gestion de la mémoire et deux TLB séparés de 32 entrées pour les instructions et les données.

L'originalité de ce micro-contrôleur est qu'il comporte un coprocesseur pour la gestion des communications, semblable à celui du Motorola 68360 avec en outre une fonction multiplication/accumulation. Ce coprocesseur gère les signaux de quatre SCC<sup>11</sup>, de deux canaux série UART, d'un SPI<sup>12</sup> et d'un port I<sup>2</sup>C<sup>13</sup>. Une interface PCMCIA est aussi disponible sur ce micro-contrôleur. Deux fréquences sont proposées pour le MPC 860 : 25 et 40 MHz.

Le MPC 821

Sorti fin 1995, comme le MPC 860, le MPC 821 lui ressemble beaucoup. En effet les seules différences concernent les entrées/sorties et le domaine d'application (le coprocesseur de communications du MPC 860 reste le même). Le MPC 821 est destiné au marché des PDA, et pour cela intègre un contrôleur d'écrans à cristaux liquides. Les quatre SCC ont de plus été réduits à deux dont un seulement supporte le protocole Ethernet. Enfin, contrairement au MPC 860, le MPC 821 ne se vend qu'en une seule version cadencée à 40 MHz.

Le MPC 823

Le MPC 823, sorti à la mi-1996, est presque identique au MPC 821, mais vise un autre marché, celui des caméras numériques. Ainsi, le contrôleur d'écrans à cristaux liquides a été amélioré puisqu'il supporte des écrans couleurs 24 bits (au lieu d'écrans monochromes ou couleurs 4 bits du MPC 821). Cependant, d'autres caractéristiques ont été revues à la baisse :

- La taille des caches de 2 Koctets pour les instructions et 1 Koctets.
- La capacité des TLB est de : huit entrées pour les données et huit entrées pour les instructions.
- Deux canaux DMA et un canal série ont disparu par rapport au MPC 821.

La suppression des fonctionnalités précédentes se repercute naturellement sur le prix qui est revu à la baisse lui aussi par rapport au MPC 821.

Le MPC 801

Le MPC 801 qui date d'août 1996 est le plus récent et le plus simple de la gamme de micro-contrôleurs de *Motorola*, et le moins coûteux. Le cœur du composant est le même que celui du MPC 860. Les différences avec ce dernier sont les suivantes :

- La taille des caches a été diminuée. Le cache d'instruction passe de 4 Koctets à 2 Koctets, le cache de données de 4 Koctets à 1 Koctets, comme sur le MPC 823.
- Les TLB ont aussi été réduits : il ne contiennent plus que huit entrées chacun (idem MPC 823).
- Le bus d'adresses a été réduit à 26 bits.
- Le support PCMCIA a été éliminé.
- Le coprocesseur de communication et les entrées/sorties associées ont été remplacés par :
  - deux canaux UART ;
  - une interface I<sup>2</sup>C ;
  - un canal SPI.

Ce microprocesseur est disponible aux fréquences de 25 et 40 MHz.

---

11. Serial Communication Controller

12. Serial Peripheral Interface

13. *Inter-Integrated Circuit*.

## Produits IBM

La famille de micro-contrôleurs d'*IBM* s'est, elle aussi, agrandie depuis [1] à travers la sortie du PowerPC 401. Nous faisons tout de même un petit retour sur le PowerPC 403 qui a évolué, avant d'exposer les caractéristiques du 401.

### Le PowerPC 403

Sorti en 1994, le PowerPC 403 a à peu près les mêmes caractéristiques que le MPC 505 de *Motorola* décrit précédemment. Les instructions sont exécutées à raison d'une par cycle. Ce microprocesseur ne comporte pas de FPU, les calculs arithmétiques flottants sont donc émulés. Un cache d'instructions de 2 Koctets est disponible ainsi qu'un cache de donnée de 1 Koctets (sauf pour le dernier modèle sorti). La faible capacité de ces caches est légèrement compensée par un degré d'associativité de deux. Les caractéristiques spécifiques au domaine des microprocesseurs embarqués présentes dans le MPC 505 se retrouvent ici (gestion dynamique de la consommation, modes de faible consommation, intégration de contrôleurs d'entrées/sorties de type DMA, interruption, etc.). Quatre versions de ce micro-contrôleur sont proposées, leurs différences sont présentées dans le tableau 3.3.

	Fréquence (MHz)	Caches (I/D)	TLB	MMU	Canaux DMA
<b>403GA</b>	33, 40*	2/1 Koctets	—	non	4
<b>403GB</b>	28	2/1 Koctets	—	non	2
<b>403GC</b>	33, 40*	2/1 Koctets	64+4	oui	4
<b>403GCX<sup>†</sup></b>	66	16/8 Koctets	64+4	oui	4

\*. Fréquences les plus récentes (04/97).

†. Dernier né (04/97).

TAB. 3.3 – Les principales différences entre les micro-contrôleurs PowerPC 403.

### Le PowerPC 401

Ce microprocesseur est le dernier né (annoncé à la mi-1996) de la famille des micro-contrôleurs d'architecture PowerPC proposée par *IBM*. Son cœur est aussi le plus petit PowerPC, il mesure 4,5 mm<sup>2</sup> (la taille totale du composant est de 22 mm<sup>2</sup>). Deux versions sont proposées, une comportant uniquement le cœur et destiné au développement d'ASIC<sup>14</sup>, l'autre étant un vrai micro-contrôleur appelé PowerPC 401GF.

Les cinq étages de pipeline d'un PowerPC classique ont été réduits à trois étages. Ce micro-contrôleur intègre une unité de multiplication/division, un contrôleur de cache et une unité de gestion de la mémoire. Le 401GF comporte de plus 2 Koctets de cache d'instructions, 1 Koctets de cache de données et un port JTAG<sup>15</sup>.

Deux fréquences sont disponibles: 25 et 50 MHz. Le processus de fabrication est CMOS à 0,5 μm. De nouvelles versions devraient sortir prochainement, celles-ci seraient cadencées à 75 MHz, puis 100 MHz et la technologie CMOS devrait passer à 0,35 μm.

## 3.2.4 Perspectives pour l'architecture PowerPC

L'équipe PowerPC a mis en avant les produits qui devraient étendre la gamme de microprocesseurs PowerPC d'ici l'an 2000.

14. *Application Specific Integrated Circuit*.

15. *Join Test Action Group*

Courant 1997 le **G3** (nom de code **Arthur**) apparaîtra. Celui-ci est dérivé du 603e avec de plus larges caches (32 Koctets pour les instructions et 32 Koctets les données) et une interface système améliorée (bus pour cache secondaire séparé). Le G3 comporte aussi deux unités entières (une de plus que sur le 603). D'autre part un mécanisme de prédiction dynamique de branchement similaire à celui utilisé dans le 604 est repris pour le G3. L'aspect technologique comprend un passage au procédé de fabrication CMOS à 0,25  $\mu\text{m}$ . Le nombre de transistors atteint est de 6,4 millions; ceux-ci occupent une surface de 67  $\text{mm}^2$ . La fréquence annoncée est 250 MHz.

Après le G3, le **Mach 5** devrait sortir, toujours en 1997. Il constitue une amélioration du 604e, principalement due à un vrai procédé CMOS à 0,25  $\mu\text{m}$ . Ainsi, le processeur serait cadencé à 300 MHz. L'interface bus devrait être revue pour ne pas limiter les performances, mais le groupe PowerPC n'a pas encore donné de détails à ce sujet.

En 1998-99 la famille **G4** devrait commencer à être fabriquée. L'architecture de cette série serait totalement refaite, et la technologie physique encore améliorés (CMOS à 0,18  $\mu\text{m}$ ). Le nombre total de transistors des microprocesseurs de cette famille devrait avoisiner les 50 millions.

Plus tard, un nouveau type d'architecture devrait voir le jour, il porte pour l'instant le nom de *Project 2K*.



## Chapitre 4

# Les microprocesseurs SPARC

*SPARC Technology Business* est une division de *Sun Microsystems, Inc.* L'architecture SPARC, issue des travaux menés à l'université de Berkeley entre 1984 et 1987 est basée sur le concept RISC. Elle privilégie une architecture pipelinée, un jeu d'instructions simple accompagné d'un compilateur efficace<sup>1</sup>.

De même que MIPS ou DEC, Sun applique une politique commerciale ouverte. Les autres compagnies ont donc toute liberté pour développer leur propre implémentation de cette architecture. Cette stratégie contribue au développement d'une base logicielle importante et permet une évolution rapide de cette architecture. Pour la surveiller et en guider l'évolution, un comité, *SPARC International*, a été créé en 1989. Il regroupe des représentants de l'ensemble des développeurs ou vendeurs souhaitant influencer sur l'évolution du standard de cette architecture. On trouve entre autres et en dehors de *Sun* : *Fujitsu Ltd.*, *Texas Instruments* (qui assure entre autres la fabrication des microprocesseurs conçus par *Sun*), *ROSS Technologies Inc.*, *Hal Computer Systems*, *TEMIC*, ... *SPARC International* garantit la compatibilité binaire entre les diverses évolutions de ce standard. Ce comité joue un rôle essentiel sur l'avenir de cette architecture puisqu'il est à l'origine des normes SPARC-V8 puis V9 dont nous détaillons quelques caractéristiques.

Conséquence de la stratégie commerciale menée par *Sun*, les microprocesseurs d'architecture SPARC offrent une grande diversité. Aussi, dans la suite, nous présentons la plupart des familles de microprocesseurs SPARC existantes et certifiées par *SPARC international* (excepté l'UltraSPARC II) en précisant leurs évolutions depuis [1]. Cependant, dans un premier temps, nous exposons les caractéristiques générales de l'architecture SPARC et plus particulièrement celles de la version 9, la plus récente.

### 4.1 Architecture SPARC

Dans cette partie nous présentons les particularités de l'architecture SPARC en plusieurs points. Tout d'abord des généralités sur l'architecture sont exposées. Puis nous détaillons un peu plus la dernière version (SPARC-V9). Enfin, nous parlons de l'extension multimédia implémentée dans les processeurs UltraSPARC, plus connue sous le nom de *VIS*<sup>2</sup>.

Cette partie sera malgré tout assez succincte, étant donné que l'architecture SPARC a été détaillée dans [1] et qu'elle n'a pas subi de modifications notables depuis.

---

1. SPARC est un acronyme pour *Scalable Processor ARChitecture*.

2. *Visual Instruction Set*.

### 4.1.1 Généralités

L'architecture SPARC définie à l'origine par *Sun Microsystems* en 1985 est une architecture RISC, de type *load/store*. Cette architecture a évolué à travers plusieurs versions et la première à avoir fait l'objet d'une implantation physique a été la norme SPARC-V7; le microprocesseur correspondant est le *Sun SPARC* (celui-ci est détaillé dans [1] et nous n'en parlerons pas plus ici).

Actuellement, deux normes existent : la norme V8 spécifie une architecture 32 bits et la norme V9 propose une architecture 64 bits et amène des améliorations à la précédente. Il existe aussi une norme V8E destinée aux systèmes embarqués.

Dans ce qui suit, nous ne parlons que de la version 9 (des références précises à la version 8 se trouvent dans [1]) et de l'extension multimédia VIS implantée sur l'UltraSPARC.

### 4.1.2 La norme SPARC-V9

La norme SPARC-V9, disponible depuis fin 1993, constitue le plus grand changement apporté à l'architecture SPARC depuis sa création. Elle est conçue pour améliorer les performances des systèmes construits à partir de celle-ci et par là répondre aux exigences des utilisateurs des années 90 et au-delà. Pour cela, elle met tout en œuvre, entre autres, pour permettre des implantations superscalaires.

#### Jeu d'instructions

L'architecture définie par la norme SPARC-V9 est une architecture 64 bits, mais assure une compatibilité binaire ascendante avec la norme V8 et reconnaît donc les données et les adresses sur 32 bits. En effet, toutes les adresses sont calculées sur 64 bits; les formats de données supportés et les types de données associés sont exposés dans le tableau 4.1.

	octet	demi-mot	mot	mot étendu	mot étiqueté	double-mot	quadruple-mot
Nombre de bits	8	16	32	64	32 <sup>a</sup>	64	128
Entiers signés	x	x	x	x	x	x	
Entiers non signés	x	x	x	x		x	
Flottants			SP <sup>b</sup>			DP <sup>c</sup>	QP <sup>d</sup>

TAB. 4.1 – *Formats et types de données de la norme SPARC-V9*

<sup>a</sup> 30 bits de valeur plus 2 bits d'étiquette.

<sup>b</sup> Simple Précision.

<sup>c</sup> Double Précision.

<sup>d</sup> Quadruple Précision.

Les quatre modes d'adressage classiques (absolu, indirect, basé et indexé) sont conservés de la norme V8.

*SPARC International* divise le jeu d'instructions en sept groupes :

1. Instructions d'accès à la mémoire – Comme l'architecture SPARC est de type *load/store*, les seules instructions mémoire concernent la lecture et l'écriture des données (celles-ci doivent être alignées sur des frontières respectant le format de la donnée accédée) entre la mémoire et les

registres. L'ordre des octets en mémoire est de type *big-endian* mais peut-être configuré au format *little-endian*. Certaines instructions particulières sont proposées :

- instructions d'accès atomique à la mémoire ;
  - instructions de lecture ne provoquant pas d'exception (utiles par exemple pour entamer le chargement d'une donnée accédée par pointeur avant de tester si le pointeur est nul) ;
  - instructions de pré-charge pour une ou plusieurs lectures, pour plusieurs écritures, pour une page, . . .
2. Instructions arithmétiques entières – Ces instructions regroupent les opérations classiques de calcul entier, de test et de décalage. De plus, des instructions d'addition et de soustraction « étiquetées » sont utiles par exemple pour les langages *Lisp* et *Smalltalk*. L'étiquette concerne les deux bits de poids faible des opérands. Si l'une ou l'autre de ces étiquettes est non nulle ou si un débordement de capacité survient alors un *overflow* est signalé.
  3. Instructions arithmétiques flottantes – Ces instructions effectuent l'ensemble des calculs arithmétiques flottants conformes à la norme *IEEE 754-1985* sur les données 32 et 64 bits (pour les données 128 bits la norme en vigueur est la *IEEE Standard for Shared Data Formats 1596.5-1993*).
  4. Instructions de contrôle – Ces instructions concernent les sauts (conditionnels et inconditionnels) et sont pour la plupart retardées (un bit d'annulation est disponible ; dans ce cas-ci, l'instruction suivant le branchement n'est pas chargée depuis le cache). Les déplacements sont soit indirects (à travers un registre) soit relatifs au PC (de  $\pm 128$  Koctets à  $\pm 2$  Goctets selon la nature de l'instruction).
  5. Instructions de déplacements conditionnels – Une donnée contenue dans un registre source est copiée dans un autre registre selon la valeur d'un troisième registre entier ou la valeur d'un code condition entier ou flottant.
  6. Instructions d'accès aux registres d'état – Ces instructions permettent la lecture et l'écriture des registres d'états.
  7. Instructions de gestion de fenêtres de registres – L'architecture SPARC met en œuvre des fenêtres de registres pour accélérer les changements de contextes lors d'appels ou de retours de procédures. Le passage des paramètres est permis par le recouvrement des fenêtres de registres (voir paragraphe suivant).

## Autres caractéristiques de l'architecture

### Registres

L'architecture SPARC-V9 spécifie de 64 à 528 registres généraux entiers de 64 bits pour la mise en œuvre de fenêtres de registres. Une fenêtre de registres est composée de huit registres d'entrées, huit registres locaux et huit registres de sorties. Ainsi, à n'importe quel moment, une instruction peut accéder aux 24 registres de la fenêtre courante mais aussi à huit registres globaux indépendants du contexte (FIG. 4.1). La fenêtre courante est désignée par un pointeur (nommé *CWP* sur la figure pour *Current Window Pointer*).

La norme SPARC-V9 impose aussi pour les registres flottants :

- 32 registres simple précision (32 bits) ;
- 32 registres double-précision (64 bits) ;
- 16 registres quadruple-précision (128 bits) ;

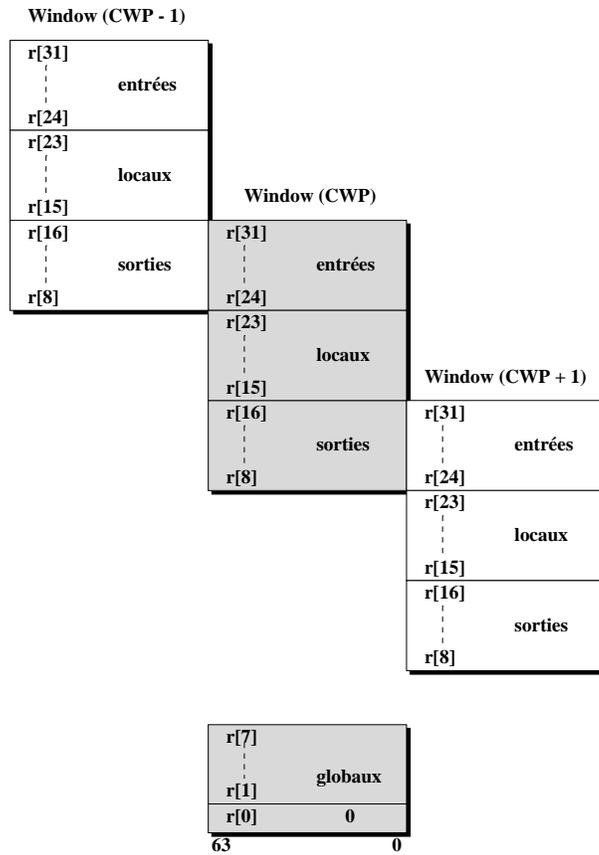


FIG. 4.1 – Fenêtres de registres sur l'architecture SPARC-V9.

Ceux-ci ne sont pas regroupés en fenêtres mais restent globaux.

### Modèle mémoire

La norme V9 spécifie plusieurs modes de comportement de la mémoire qui dépendent de son utilisation. Ceci permet de mettre en place des systèmes monoprocesseurs ou multiprocesseurs à mémoire partagée à travers la gestion de l'ordre des lectures et des écritures. SPARC-V9 propose les quatre modes suivants :

- *Strong Consistency* (ordre total des lectures ou écritures mémoire en provenance d'un même microprocesseur) ;
- *Total Store Ordering (TSO)* (ordre total des écritures en mémoire) ;
- *Partial Store Order (PSO)* (ordre partiel des écritures) ;
- *Relaxed Memory Order (RMO)* (peu de contraintes, permet de séquencer les accès à la mémoire (lecture ou écriture) dans n'importe quel ordre tant que le résultat reste juste).

Le jeu d'instructions défini par la norme SPARC-V9 inclut également des instructions spécifiques d'accès à la mémoire qui permettent de mettre en œuvre des primitives de synchronisation utilisées par les mécanismes d'exclusion mutuelle.

### Gestion des exceptions

Pour la gestion des exceptions, la norme SPARC-V9 implémente une pile de registres destinés à sauvegarder l'état courant du microprocesseur (pas de changement de fenêtre de registres). Quatre niveaux de pile constituent le minimum spécifié par la norme SPARC-V9 qui peut aller jusqu'à sept niveaux d'interruption. D'autre part, pour quitter les routines d'exception, des instructions spécifiques restituent l'état antérieur du microprocesseur de façon atomique.

### 4.1.3 Extension multimédia

Des instructions multimédias peuvent s'ajouter au jeu d'instructions SPARC-V9, elles sont regroupées dans ce qui s'appelle le *Visual Instruction Set (VIS)*. Implantées dans un premier temps sur l'UltraSPARC, elles sont destinées à améliorer les performances graphiques et le traitement d'images.

Deux types de données sont définis : les *pixels* et les données fixes.

Les pixels sont composés de quatre octets entiers non-signés (contenus dans un mot de 32 bits) et utilisés pour les images vidéo.

Les données fixes existent en deux formats : quatre demi-mots (16 bits) ou deux mots (32 bits) contenus dans un double mot (64 bits). Celles-ci sont utiles pour stocker des résultats intermédiaires lors de traitement d'images.

Les instructions du *VIS* utilisent les registres flottants.

Les différents types d'instructions sont les suivants :

- Instructions de conversion – Elles convertissent des données entre les formats spécifiques détaillés plus haut.
- Instructions arithmétiques et logiques – Dans une approche SIMD, celles-ci réalisent des calculs sur chaque octet, demi-mot ou mot en parallèle, dans un double mot (ou un mot dans le cas des *pixels*).
- Instructions de manipulation d'adresses – Elles favorisent le calcul d'adresses accédant des données non alignées en mémoire.
- Instructions d'accès mémoire – Elles permettent des écritures partielles de données de 8, 16 ou 32 bits et des transferts de blocs.

- Instruction d'estimation de mouvement – Celle-ci calcule la différence absolue entre les octets correspondants d'une paire de registres flottants double-précision, et accumule les « erreurs » dans le registre destination (cf. instruction *PERR* de l'extension multimédia du jeu d'instruction Alpha).

À noter que, pour l'instant, *VIS* est implanté uniquement sur l'UltraSPARC, mais à terme *Sun* envisage de l'implanter dans tous les processeurs équipant ses stations.

## 4.2 Le SuperSPARC

Annoncé par *Texas Instruments* et *Sun* en août 1991, le SuperSPARC est le premier microprocesseur superscalaire d'architecture SPARC. Il est destiné au marché des stations de travail et s'intègre dans celles fournies par *Sun*.

Le SuperSPARC comprend une unité entière intégrant deux ALU, une unité flottante, une unité de gestion de la mémoire ainsi que deux caches d'instructions et de données séparés. Par ailleurs, pour compléter l'unité de gestion de la mémoire, un TLB unifié de 64 entrées est mis en œuvre. Ce microprocesseur est conforme à l'architecture SPARC-V8.

Quatre instructions sont chargées directement à partir du cache vers deux files d'instructions de six entrées chacune. La première de ces files est destinée aux instructions du flot séquentiel, l'autre réceptionne les instructions issues des branchements (schéma similaire à l'architecture POWER). Quand le branchement est pris, les instructions issues du flot séquentiel sont invalidées. Trois instructions peuvent être émises simultanément vers les diverses unités d'exécution. Cette émission se fait toujours dans l'ordre.

Trois intructions peuvent s'exécuter en parallèle parmi les cinq suivantes :

- une instruction arithmétique entière ;
- une autre instruction arithmétique entière ;
- une instruction de branchement ;
- une instruction d'accès mémoire (lecture sur 32 bits seulement si elle est couplée avec une instruction arithmétique entière) ;
- une instruction arithmétique flottante ;

Le pipeline entier s'exécute sur quatre cycles mais comporte huit phases.

Le SuperSPARC comprend deux caches d'instructions et de données séparés, tous deux physiquement adressés. Le cache d'instructions a une taille de 20 Koctets et il est associatif par ensembles à cinq voies. Le cache de données a quant à lui une taille de 16 Koctets et est associatif par ensembles à quatre voies. La politique de mise à jour de la mémoire est configurable : écritures simultanées ou différées.

Un second niveau de cache peut également être ajouté à ce microprocesseur. Un contrôleur externe a été développé. Ce composant supplémentaire inclut les étiquettes de ce cache et permet l'ajout de 1 Moctets à correspondance directe.

L'interface bus supporte deux protocoles : soit celui utilisé par le contrôleur de cache externe soit, si ce dernier est absent, le MBus niveau 2.

Commercialement, ce microprocesseur n'a pas remporté un franc succès car il a subi de nombreux problèmes. Annoncé dans un premier temps à 50 MHz en 1991, les premiers microprocesseurs ne sont livrés qu'à 33 MHz. Il faudra attendre début 1993 avant d'atteindre les 40 MHz, puis le SuperSPARC+ commercialisé au deuxième semestre 1993 pour atteindre les 50 MHz. Finalement, le SuperSPARC n'a pas dépassé 60 MHz. Ce microprocesseur comprend 3.1 millions de transistors et est réalisé en technologie BiCMOS à 0.6  $\mu\text{m}$ .

### 4.2.1 Le SuperSPARC-II

Annoncé fin 1994, le SuperSPARC-II a été conçu pour remplacer le SuperSPARC ; il constitue une nouvelle version de celui-ci, et supprime les problèmes qui empêchaient le SuperSPARC d'atteindre des fréquences plus hautes et de meilleures performances.

Trois points majeurs ont été revus par rapport au SuperSPARC, en dehors de l'extension des files d'instructions à 12 entrées chacune :

1. Sur le SuperSPARC, le fichier de registres peut être accédé deux fois par cycle. Ces accès sont réalisés en deux phases en utilisant simplement quatre ports. Un vrai fichier 8 ports est implanté sur le SuperSPARC-II.
2. Le TLB unifié du SuperSPARC accédé lui aussi deux fois par cycle a été séparé en deux : un de 64 entrées pour les données, l'autre de 16 entrées pour les instructions.
3. L'unité de multiplication flottante aussi chargée de la division et de la racine carrée sur le SuperSPARC a été partagée. Ainsi l'horloge a pu être augmentée en permettant un plus grand nombre de cycles pour les division et racines carrées (les multiplications s'exécutent toujours en trois cycles).

Le même processus de fabrication BiCMOS à 0.6  $\mu\text{m}$  est utilisé pour le SuperSPARC et le SuperSPARC-II, mais la taille du SuperSPARC-II est supérieure de 17 %. Les fréquences disponibles sont 75 et 90 MHz.

## 4.3 Le MicroSPARC

Annoncé en octobre 1992 par *Texas Instrument* et *Sun Microsystems*, le MicroSPARC est destiné aux stations de travail d'entrée de gamme et au marché des portables.

L'originalité de ce microprocesseur réside dans sa facilité d'intégration dans les systèmes : il comprend le cœur du processeur (IU<sup>3</sup>, FPU et caches) ainsi que toute la logique d'interface vers la mémoire principale et le SBus<sup>4</sup>. Il est conforme à l'architecture SPARC-V8 et séquencé à 50 MHz.

L'unité entière utilise les cinq étages classiques d'un pipeline RISC et accède à sept fenêtres de registres (120 registres en tout). L'unité flottante est non pipelinée et les opérations les plus simples (additions) ont une latence minimum de cinq cycles.

Le MicroSPARC intègre deux caches primaires pour les instructions et les données de capacités respectives 4 Koctets et 2 Koctets. Ces deux caches sont à correspondance directe, physiquement adressés et à recopie simultanée. Leur faible capacité est compensée par une interface directe avec la mémoire principale qui permet de satisfaire les défauts en quatre cycles dans le meilleur des cas.

L'unité de gestion de la mémoire (MMU) comporte un TLB associatif de 32 entrées et un *micro-TLB* d'une entrée pour les instructions ; ce dernier est uniquement accédé lorsque le TLB principal est inaccessible. La taille des pages gérées par l'unité de gestion de la mémoire est paramétrable et est de 4 Koctets, 256 Koctets ou 16 Mcoctets.

Le MicroSPARC est réalisé en technologie CMOS à 0,8  $\mu\text{m}$  et est séquencé à 50 MHz. Sa taille est de 225 mm<sup>2</sup>.

---

3. *Integer Unit*.

4. Introduit en 1990 dans la gamme des stations de travail SPARC, le Sbus est devenu le standard utilisé par ces systèmes. Il fonctionne à une fréquence de 25 MHz.

### 4.3.1 Les autres MicroSPARC

Le **MicroSPARC II** a été annoncé en mars 94 : c'est une version améliorée du MicroSPARC. Les caches internes ont été agrandis (8 Koctets pour les données et 16 Koctets pour les instructions) ainsi que le nombre de fenêtres de registres (8 fenêtres, soit 136 registres). La technologie utilisée pour ce microprocesseur est CMOS à  $0,5 \mu\text{m}$  ; il est séquencé à 85 MHz dans un premier temps, et atteint 110 MHz en 1995.

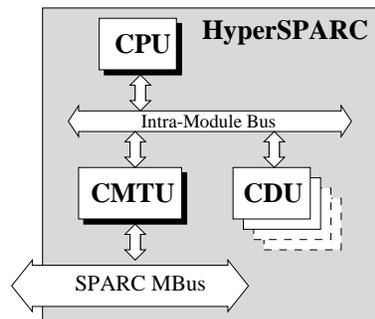
D'autre part, le MicroSPARC II s'enrichit d'une interface PCI en remplacement du Sbus classique. Ce dernier microprocesseur, appelé **MicroSPARC IIep** est disponible depuis la fin 1996 et est semblable au MicroSPARC-II. Il est séquencé à 100 MHz.

Le **MicroSPARC-III**, version améliorée du MicroSPARC II, était initialement prévu pour la mi-1995. Cependant, les concepteurs, après plusieurs changements d'avis ont décidé de sortir un microprocesseur bien plus puissant contenant un cœur d'UltraSPARC. Finalement, ce microprocesseur porte le nom d'**UltraSPARC-III** et devrait sortir fin 1997 (voir p. 79).

## 4.4 L'HyperSPARC

Annoncé en avril 1992, l'HyperSPARC constitue une implémentation multi-composants de l'architecture SPARC-V8, conçue par *Ross Technology*, filiale de *Fujitsu*. Depuis, plusieurs modèles sont sortis (le *Pinnacle*, puis le *Colorado 1*, le *Colorado 2*, le *Colorado 3* et enfin le *Colorado 4*) et les fréquences internes s'étalent de 66 MHz à maintenant 200 MHz.

La figure 4.2 expose les différents composants de l'HyperSPARC.



CMTU : Cache controller, Memory management and Tag Unit.  
CDU : Cache Data Units.

FIG. 4.2 – Les composants de l'HyperSPARC.

Le CPU de l'HyperSPARC **Pinnacle** d'origine est superscalaire de degré deux. Il est constitué d'une unité entière, d'une unité flottante, d'une unité de lecture/écriture, d'une unité de branchement et d'un cache d'instructions de 8 Koctets associatif par ensembles à deux voies. Deux instructions sont chargées à chaque cycle et émises vers les unités d'exécution respectives si les dépendances de données le permettent. Le Pinnacle utilise un pipeline classique de cinq étages. L'unité entière inclut 136 registres (huit fenêtres) et deux pipelines d'exécution pour les opérations arithmétiques et de lecture/écriture. L'unité flottante comprend quant à elle 32 registres flottants ainsi qu'une file d'instructions séparée qui permet l'émission simultanée de deux instructions flottantes (addition ou multiplication).

Le second niveau de cache (externe au CPU) est unifié (il sert de cache primaire pour les données) et à correspondance directe. Il est virtuellement adressé mais physiquement testé, facilitant ainsi

le maintient de la cohérence dans un environnement multiprocesseurs. La politique d'écriture est configurable (différée ou simultanée). Sa taille peut être de 128 Koctets ou 256 Koctets.

La technologie utilisée pour le Pinnacle est CMOS à 0,8  $\mu\text{m}$ . La fréquence maximale atteinte est 66 MHz.

L'HyperSPARC de deuxième génération, le **Colorado 1** (1994) passe à la technologie MCM et CMOS à 0,5  $\mu\text{m}$ . Sa fréquence peut ainsi atteindre les 100 MHz. Son successeur, le **Colorado 2** (1995) atteint la fréquence de 125 MHz grâce à une technologie CMOS à 0,4  $\mu\text{m}$ .

Le **Colorado 3** (fin 1995) n'améliore pas la technologie physique de ses prédécesseurs, mais ajoute une seconde unité de calcul entier qui permet de paralléliser les calculs entiers simples avec des branchements, des instructions mémoire ou du calcul flottant. D'autre part, il permet d'intégrer un cache secondaire de taille plus importante (512 Koctets, 1 Moctets). Sa fréquence peut aussi être augmentée à 150 MHz car les chemins critiques ont été revus.

Le **Colorado 4** (1996) ajoute à son CPU un cache primaire de données de 16 Koctets et double la taille du cache d'instructions ce qui l'amène à 16 Koctets aussi. La FPU a été revue (les opérations simples se font en deux cycles au lieu de trois) et elle comporte désormais un multiplieur double précision. Le processus de fabrication CMOS à 0,35  $\mu\text{m}$  lui permet d'atteindre les 200 MHz.

## 4.5 L'UltraSPARC

L'UltraSPARC est le premier microprocesseur de *Sun's SPARC Technology Business* à implémenter l'architecture 64 bits SPARC-V9. Annoncé en septembre 1994, sa commercialisation a commencé fin 1995. Il équipe désormais les stations de travail *Sun* moyen et haut de gamme. Ce microprocesseur superscalaire de degré quatre, totalement compatible avec les applications 32 bits déjà existantes est destiné au marché des applications hautes performances ; aussi il privilégie une haute fréquence d'horloge (167 MHz à l'origine, 200MHz en mars 1997). Ce microprocesseur a déjà fait l'objet d'une étude dans [1] et [6], et pour cela nous n'exposerons ici qu'un résumé de ses caractéristiques en les actualisant. D'autre part, nous exposons dans deux autres parties les microprocesseurs UltraSPARC-II et UltraSPARC-III.

### Jeu d'instructions

L'UltraSPARC est conforme à la norme SPARC-V9, mais comprend aussi certaines autres caractéristiques :

- l'extension multimédia *VIS* (voir p. 73) ;
- des instructions de lecture/écriture par bloc de 64 octets ;
- des compteurs de performances (deux registres de 32 bits accessibles par logiciel) ;
- une instruction d'extinction du système (mise en veille du composant durant des périodes de moindre activité permettant une diminution de la consommation de 30 W à 20 mW).

### Séquencement et exécution des instructions

L'UltraSPARC est un microprocesseur superscalaire de degré quatre. Ainsi, quatre instructions sont lues dans le cache d'instructions à chaque cycle puis (au cycle suivant) décodées dans l'ordre et placées dans un tampon d'instructions de 12 entrées qui permet de découpler leur exécution (celles-ci peuvent ensuite être exécutées dans le désordre).

L'UltraSPARC met en œuvre des pipelines de neuf étages. Le pipeline entier contient trois étages supplémentaires de synchronisation avec le pipeline flottant. Cette approche élimine, selon les concepteurs, le besoin d'implémenter une file pour les instructions flottantes et facilite par ailleurs une gestion précise des exceptions pour toutes les instructions entières et flottantes.

Comme tous les microprocesseurs destinés à de hautes performances, l'UltraSPARC met en œuvre une prédiction dynamique des branchements. Ce mécanisme s'appuie sur l'algorithme à deux bits de Smith ; les adresses de branchement sont mémorisées à raison d'une adresse pour quatre instructions.

#### L'unité entière

L'unité entière contient deux unités arithmétiques et logiques, une unité de calcul d'adresses et diverses unités de contrôle pour les fenêtres de registres ou les pipelines (validation des chemins de *bypass*, écriture des résultats, etc.).

Le banc de registres inclut 144 registres entiers (soit huit fenêtres) avec sept ports de lecture (dont un dédié aux instructions d'écriture vers la mémoire) et trois ports d'écriture.

#### L'unité flottante et graphique.

L'unité flottante et graphique (*FGU*) comprend cinq unités d'exécution (une unité d'addition/soustraction, une unité de multiplication, une unité pour les divisions et racines carrées, et deux unités graphiques d'addition et de multiplication responsables de l'exécution des instructions du *VIS*) et un banc de 32 registres 64 bits. Toutes les opérations, à l'exception de la division et de la racine carrée, sont pipelinées. Le banc de registres dispose de cinq ports de lecture et de trois ports d'écriture permettant l'exécution concurrente de deux opérations flottantes et d'une opération de lecture/écriture. Toutes les opérations flottantes sont conformes à la norme *IEEE 754* (les opérations flottantes sur 128 bits sont émulées par logiciel).

#### Hierarchie mémoire

Deux caches primaires sont implantés sur l'UltraSPARC, un pour les données, l'autre pour les instructions.

Le cache de données est à correspondance directe et a une capacité de 16 Koctets. Ce cache est non-bloquant. Il est virtuellement indexé et physiquement testé. Ce cache n'a qu'un seul port d'accès mais la table d'étiquettes dispose d'un port de lecture et d'un deuxième port de lecture/écriture permettant ainsi l'exécution en parallèle d'une lecture des étiquettes avec une écriture précédente.

Les instructions sont chargées à partir d'un cache d'instructions de 16 Koctets, associatif par ensembles à deux voies. La politique de remplacement est de type LRU. Ce cache implémente un mécanisme de prédiction qui permet de sélectionner directement le bon ensemble réduisant ainsi la consommation et fournissant le même temps d'accès qu'un cache à correspondance directe (les deux étiquettes sont cependant lues afin de confirmer la prédiction). Il est physiquement adressé et testé.

Un deuxième niveau de cache unifié est obligatoire. Sa taille peut varier de 512 Koctets à 4 Moctets. Il est physiquement adressé et testé. Il implémente une protection par parité pour chaque octet et supporte le protocole de cohérence de type *MOESI*<sup>5</sup> ou tout autre sous-ensemble (*MESI*, *MSI*). À noter l'utilisation de *SRAM* synchrone avec le même temps de cycle que le microprocesseur.

#### L'unité de gestion mémoire.

L'UltraSPARC comporte deux unités de gestion de la mémoire indépendantes pour les instructions et les données. Chacune de ces MMU intègre un TLB associatif de 64 entrées géré par logiciel.

L'adresse virtuelle est codée sur 44 bits, l'adresse physique est, quant à elle, codée sur 41 bits. Le mécanisme de traduction d'adresses est assuré par logiciel [1, 6]. Quatre tailles de page sont supportées : 8 Koctets, 16 Koctets, 512 Koctets et 4 Moctets.

Une protection des données est assurée au niveau de chaque page sous la forme de trois modes principaux de fonctionnement :

- un mode *non privilégié* qui correspond au mode utilisateur classique ;
- un mode *privilégié* qui est équivalent au mode noyau de nombreuses machines ;

---

5. Protocole de cohérence de caches comportant cinq états : *Modified*, *Owned*, *Exclusive*, *Shared* et *Invalid*.

- un mode *RED\_state* (spécifique aux états *Reset*, *Error*, *Debug* et en cas d'erreur grave).

### Bus et interface système

L'UltraSPARC utilise un bus unique (128 bits pour les données) pour se connecter au reste du système qui comprend le cache secondaire, la mémoire principale et, éventuellement, des entrées/sorties ainsi que d'autres microprocesseurs. Ainsi un composant externe est nécessaire pour gérer l'interfaçage entre le cache secondaire (rapide et utilisant une protection par parité au niveau des octets) et les autres périphériques, dont la mémoire principale, synchronisés à des fréquences moins hautes et utilisant une protection plus sûre de type ECC. Ce composant est proposé par *Sun* et porte le nom d'*UltraSPARC Data Buffer (UDB)*.

### Support multiprocesseurs.

Le bus système de l'UltraSPARC permet d'interconnecter quatre microprocesseurs UltraSPARC et un contrôleur système. La cohérence du cache externe et du système est assurée par le protocole de cohérence MOESI.

### Technologie.

L'UltraSPARC intègre 5.2 millions de transistors. Il est réalisé en technologie CMOS à 0.5  $\mu\text{m}$  et mesure 315  $\text{mm}^2$ . Séquencé à la fréquence de 167 MHz à l'origine, il atteint maintenant les 200 MHz grâce à un processus CMOS à 0,4  $\mu\text{m}$  (cette version du microprocesseur est aussi connue sous le nom d'**UltraSPARC+** et mesure 265  $\text{mm}^2$ ).

## 4.5.1 L'UltraSPARC-II

Annoncé fin 1995, l'UltraSPARC-II est une version légèrement améliorée de l'UltraSPARC, qui a une fréquence d'horloge supérieure (actuellement 250 MHz<sup>6</sup>).

Ce microprocesseur comporte quelques nouvelles caractéristiques par rapport à son prédécesseur :

- des instructions de *prefetch* peuvent être exécutées ;
- le cache externe peut avoir une taille de 16 Moctets ;
- les SRAM synchrones utilisées dans le cache externe peuvent avoir une horloge correspondant aux  $\frac{2}{3}$  de celle du CPU ;
- l'interface bus supporte désormais plusieurs accès concurrents (trois lectures, deux écritures) ;
- la technologie CMOS à 0,29  $\mu\text{m}$  et l'alimentation sous 2,5 V permettent une fréquence interne de 250 MHz.

## 4.5.2 L'UltraSPARC-III

L'UltraSPARC-III proposé par *Sun* devrait apparaître fin 1997. Destiné au marché des stations de travail *Sun* d'entrée de gamme, il remplacera le MicroSPARC-II. L'UltraSPARC-III peut-être considéré comme un UltraSPARC-II « à moindre coût » car il comporte des interfaces lui permettant de s'intégrer facilement dans les systèmes, même si le microprocesseur en lui-même ne devrait pas être meilleur marché.

Le cœur (unités fonctionnelles) de l'UltraSPARC-III reste le même que celui de l'UltraSPARC-II : CPU superscalaire de degré quatre avec deux caches primaires (instructions et données) de 16 Koctets chacun.

---

6. Il semble qu'en mars 1997 ce processeur ne soit disponible qu'en très faible volume.

L'interface système a largement été simplifiée (au niveau de l'utilisation) par rapport à l'UltraSPARC-II. Celle-ci gère de manière distincte :

- le cache secondaire unifié (taille comprise entre 256 Koctets et 2 Moctets) grâce à un bus de 64 bits pour les données (plus 8 bits d'ECC et 18 bits d'étiquettes) ;
- la mémoire principale à travers un bus au format *UPA*<sup>7</sup> (données sur 64 bits + 8 bits d'ECC) ;
- un bus PCI à 33 MHz ou 66 MHz.

Ainsi les gestions externes au microprocesseur sont séparées et permettent d'éviter l'emploi de l'UDB des précédents UltraSPARC. En contre-partie les trois bus distincts mis en place imposent un grand nombre de broches pour le composant (entre 550 et 600).

Fabriqués en CMOS à 0,35  $\mu\text{m}$ , l'UltraSPARC-IIi mesurerait environ 144 mm<sup>2</sup> et devrait atteindre 300 MHz.

## 4.6 Le SPARC64

Développé par la société *Hal Computer Systems*, ce microprocesseur a été annoncé en mars 1995 et équipe les stations commercialisées par la société. Il vise le marché des stations de travail haut de gamme et des serveurs et se place donc sur le même plan que l'UltraSPARC.

Créé en 1990, *Hal* a été absorbé par le groupe *Fujitsu* début 1991. Cette fusion marque la volonté de *Fujitsu* d'être présent sur le marché des microprocesseurs milieu et haut de gamme. Effectivement, déjà propriétaire de *Ross Technology*, *Fujitsu* développe l'HyperSPARC. Le Sparc64 lui permet d'occuper le marché haut de gamme.

Ce microprocesseur met en œuvre une architecture découplée conforme à la norme SPARC-V9 ainsi qu'une exécution spéculative et dans le désordre (seulement après la phase de décodage).

Le Sparc64 est un microprocesseur multicomposants (conditionnement *MCM*) constitué de deux composants logiques et de quatre composants mémoire responsables des fonctionnalités suivantes :

- une unité centrale qui contient le cœur du séquençement superscalaire, un cache d'instructions de 4 Koctets et la logique de contrôle pour les deux caches d'instructions et de données externes ;
- deux caches externes pour les instructions et les données de 128 Koctets (quatre composants de 64 Koctets) virtuellement adressés et testés, et non bloquants ;
- une unité de gestion de la mémoire responsable des traductions d'adresses et gérant les interfaces vers la mémoire principale et les bus d'entrées/sorties.

L'unité de séquençement reçoit à chaque cycle quatre instructions qu'elle émet dans l'ordre, après décodage, vers les stations de réservation associées à chaque unité d'exécution (ou directement vers l'unité de branchement). Elle assure aussi un renommage des registres grâce à 38 registres entiers et 24 registres flottants *supplémentaires* (il y a 118 registres entiers au total qui correspondent à quatre fenêtres).

Afin de suivre la trace de l'exécution des instructions, une étiquette de six bits est assignée à chaque instruction (d'où 64 instructions actives). Une instruction peut être dans un état parmi cinq (*issued, executing, waiting, completed, retiring*). Trois pointeurs sont associés au groupe d'instructions actives : l'*issue pointer* (dernière instruction émise), le *commit pointer* (dernière instruction en cours d'exécution) et le *retirement pointer* (instruction active la plus ancienne). Ce mécanisme, différent du traditionnel tampon de réordonnancement, permet de garder un état précis de l'exécution pour défaire les instructions issues de branchements mal prédits et pour traiter les exceptions. La prédiction de branchement mise en œuvre ici est basée sur l'algorithme de Smith.

---

7. UltraSPARC Port Architecture.

Le SPARC64 comporte neuf unités d'exécution :

- deux unités entières (station de réservation commune de huit entrées) non symétriques (une seule traite les multiplications et les divisions) ;
- deux unités de calcul d'adresses (station de réservation commune de huit entrées) pouvant en outre assurer des opérations entières simples ;
- deux unités d'accès mémoire (station de réservation commune de 12 entrées) ;
- deux unités flottantes (station de réservation commune de huit entrées) non symétriques (une seule est dédiée aux divisions et opérations à longue latence) ;
- une unité de branchements.

Ce microprocesseur est implémenté sous la forme d'un conditionnement *MCM* céramique. Il est conçu selon une technologie CMOS à  $0.4\ \mu\text{m}$  et contient au total 21.9 millions de transistors (l'essentiel étant constitué par la mémoire). Séquencé à la fréquence de 154 MHz, il dissipe environ 60 W.

Pour plus d'informations à propos du SPARC64, nous invitons le lecteur à consulter [1].

## 4.7 Le TurboSPARC

Afin de combler le manque créé par l'absence du MicroSPARC-III (et dans l'attente de l'UltraSPARC-III) *Fujitsu* a sorti fin 1996 le TurboSPARC. Ce microprocesseur est destiné aux stations de travail monoprocesseur d'entrée de gamme. Deux produits sont en fait vendus, le microprocesseur en tant que tel pour équiper les nouvelles stations de travail « à bas prix », et un *kit* de mise à jour pour les stations *SparcStation 5* existantes qui permet d'ôter le MicroSPARC-II et de le remplacer par un TurboSPARC.

Le TurboSPARC est de conception simple puisqu'il est scalaire et conforme à la norme SPARC-V8 (32 bits). Un schéma de ce microprocesseur est visible figure 4.3.

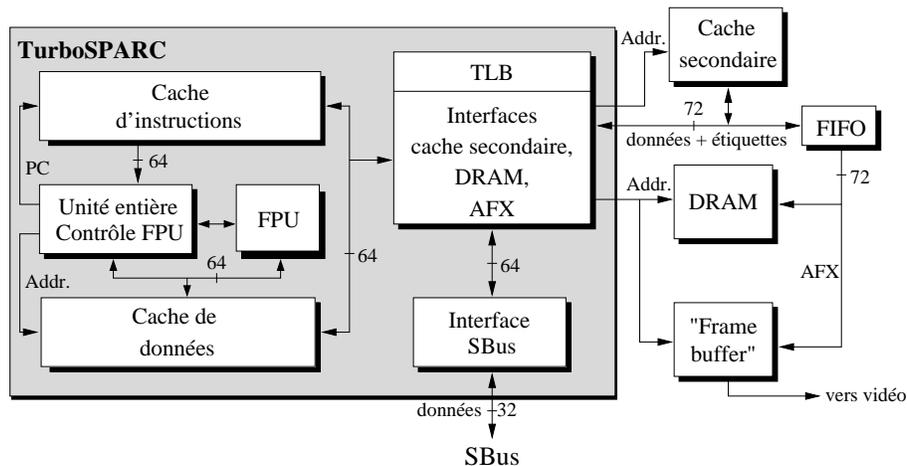


FIG. 4.3 – Synoptique du TurboSPARC.

### Pipeline

Le gain en performance par rapport au MicroSPARC-II provient de la structure du pipeline. Celui-ci comporte six étages pour les calculs entiers et huit pour les calculs flottants (FIG. 4.4) au lieu des cinq étages traditionnels du MicroSPARC. Son unicité permet de simplifier la logique de contrôle.

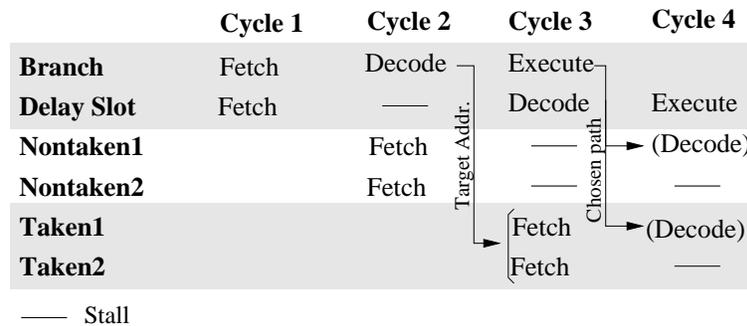
Fetch	Decode	Execute	Memory	Resolve	Write	FP Resolve	FP Write
2 instructions lues sur le cache d'instructions.	1 instruction décodée + lecture opérandes.	Opération entière.	Lecture étiquettes et données sur le cache.	Test étiquettes.	Écriture résultat dans registre entier.	Fin exécution opération flottante.	Écriture résultat dans registre flottant.

FIG. 4.4 – *Le pipeline du TurboSPARC.*

L'étage *Resolve* permet de découper l'accès au cache en deux cycles. Cette étape teste les étiquettes, et rejette la donnée lue au cycle précédent si un défaut apparaît.

L'étage *FP Resolve* est un étage d'exécution d'opérations flottantes car la plupart d'entre elles ont quatre cycles de latence.

Une des particularité du TurboSPARC est qu'il peut gérer les branchements sans prédiction et sans pénalité. Ceci est dû au fait que d'une part le microprocesseur est scalaire, d'autre part, deux instructions sont chargées du cache d'instruction à chaque cycle. Le mécanisme consiste en le chargement des deux chemins d'un branchement comme le montre la figure 4.5.

FIG. 4.5 – *Mécanisme de gestion des branchements dans le TurboSPARC.*

### H hiérarchie mémoire

Deux caches primaires d'instructions et de données sont intégrés sur le TurboSPARC ; leur taille est de 16 Koctets chacun et ils sont à correspondance directe avec la mémoire. Le cache de données est à recopie différée ; il est en outre indexé virtuellement et à étiquettes physiques — le cache d'instructions est virtuellement adressé et testé.

Le TurboSPARC, contrairement au MicroSPARC-II, supporte un cache externe unifié de taille comprise entre 256 Koctets et 1 Moctets. Celui-ci est à correspondance directe et implémente une politique d'écriture simultanée.

Dans ce cache, les étiquettes et les données sont contiguës. Parmi les 72 bits que comporte le bus interface, 64 correspondent aux données, 2 à la protection par parité, et 6 à l'étiquette. Ainsi en deux accès (la fréquence du bus peut être de la moitié de celle du CPU, ou du tiers), l'étiquette d'une ligne de cache (12 bits) est disponible pour le test.

Les accès à la mémoire principale sont entamés en même temps que ceux du cache secondaire et sont abandonnés si la donnée est présente dans ce dernier. Le gestionnaire de mémoire est à la norme SPARC-V8. Celui-ci comporte en outre de nombreuses tables de traduction d'adresses :

- un TLB de 256 entrées, à correspondance directe, pour les données ;
- un petit TLB de quatre entrées pour les instructions ;
- un TLB de 16 entrées pour les entrées/sorties ;

- un TLB de quatre entrées, associatif, pour de larges blocs de données contiguës en mémoire.

La taille des pages mémoire est fixe et de 4 Koctets.

### Interfaces système

Une interface pour la mémoire principale et le cache secondaire comporte 72 bits pour les données. Ce bus supporte aussi l'ajout d'une carte graphique proposée par *Sun* et appelée *AFX*.

D'autre part, tout comme le MicroSPARC, le TurboSPARC intègre une interface pour le SBus. Celui-ci peut être cadencé à 16 MHz ou 25 MHz.

### Technologie

Les quelques trois millions de transistors du TurboSPARC occupent 132 mm<sup>2</sup> grâce à un procédé CMOS à 0,35 μm. Alimenté en 3,3 V, il ne consomme pas plus de 9 W pour une fréquence de 170 MHz (le kit de mise à jour pour SPARCStation 5 comporte un TurboSPARC cadencé à 160 MHz).

## 4.8 Les microprocesseurs SPARC embarqués

### 4.8.1 La famille SPARClite 930

Proposée par *Fujitsu Microelectronics*, la famille SPARClite 930 réunit un ensemble de microprocesseurs destinés au marché embarqué. Ceux-ci sont compatibles avec la norme SPARC-V8. Le premier, le MB86930 est né en 1991. Depuis, quatre autres microprocesseurs ont agrandi la famille : le MB86932, le MB86933H, le MB86934 et le MB86936.

Tous les microprocesseurs de la famille SPARClite 930 comportent un pipeline classique de cinq étages. L'unité entière intègre 136 registres soit huit fenêtres de registres<sup>8</sup>. La plupart de ces microprocesseurs (tous sauf le MB86933H) comportent des caches d'instructions et de données séparés associatifs par ensembles à deux voies (le MB86933H comporte uniquement un cache d'instructions à correspondance directe).

Au niveau des entrées/sorties, sur chacun d'entre eux on trouve un contrôleur d'interruptions et une interface JTAG<sup>9</sup>.

Les principales différences entre les microprocesseurs de la famille SPARClite 930 se trouvent dans le tableau 4.2 (pour des raisons de mise en page, les noms des microprocesseurs ont été raccourcis ; pour avoir le nom complet, ajouter « MB86 » devant chaque nom).

	FPU	Cache instructions (Koctets)	Cache données (Koctets)	Bus données	Bus adresses	MMU	TLB	Canaux DMA	Techno (CMOS)	Fréquences (MHz)
<b>930</b>	non	2	2	32 bits	32 bits	oui	non	non	0,8 μm	20, 30, 40
<b>932</b>	non	8	2	32 bits	32 bits	oui	16 entrées	2	0,8 μm	20, 40
<b>933H</b>	non	1	—	32 bits <sup>†</sup>	28 bits	oui <sup>‡</sup>	—	non	0,8 μm	20, 25
<b>934</b>	oui	8	2	32 bits	32 bits	oui	—	2	0,5 μm	25, 50
<b>936</b>	oui	4	2	32 bits	32 bits	oui <sup>‡</sup>	—	3*	0,5 μm	25, 50

<sup>†</sup>. 64 bits pour les entiers.

<sup>‡</sup>. Contrôle de mémoire DRAM.

\*. Un de ces canaux DMA est réservé à l'interface vidéo.

TAB. 4.2 – Différences entre les microprocesseurs de la famille SPARClite 930.

8. Ceci n'est pas vrai pour le MB86933H qui n'intègre que 104 registres, soit six fenêtres.

9. Interface TAG sur le MB86932.

À noter que le MB86934 comporte en outre un contrôleur de SDRAM<sup>10</sup>, et le MB86936 une interface vidéo.

#### 4.8.2 La famille SPARClet

SPARClet désigne une famille de microprocesseurs SPARC embarqués proposée par *TEMIC*, une filiale de *Daimler-Benz* et *Matra MHS*. Cette famille est destinée aux contrôleurs de communication et comporte des caractéristiques de DSP<sup>11</sup> (multiplieur intégré et instruction *multiply-accumulate*). Le **TSC701**, annoncé en avril 1996 est le premier micro-contrôleur intégrant le cœur de cette architecture.

Les micro-contrôleurs appartenant à la famille SPARClet utilisent le jeu d'instructions défini par la norme SPARC V8 et un pipeline de quatre étages. Dans ce pipeline une seule instruction est chargée et décodée par cycle. Les instructions peuvent ensuite être exécutées en parallèle dans les quatre unités d'exécution (ALU, multiplieur, unité *load/store* et coprocesseur de communication). Le fichier de registres est composé de 136 registres, soit huit fenêtres.

Le coprocesseur de communication supporte les protocoles de communication HDLC<sup>12</sup> et V110, et traite les données en FIFO<sup>13</sup> à raison d'un bit par cycle.

Le TSC701 intègre un cache d'instructions de 16 Koctets et un cache de données de 8 Koctets. Ces deux caches sont associatifs par ensembles à quatre voies. D'autre part, en tant que micro-contrôleur, il intègre :

- un contrôleur de bus ;
- quatre canaux USART<sup>14</sup>/PCM<sup>15</sup> ;
- deux *timers* de 32 bits qui peuvent compter des événements extérieurs aussi bien que les cycles internes ;
- deux *timers* 32 bits pour système d'exploitation temps-réel ;
- un « chien de garde » 32 bits ;
- 10 interfaces PIA<sup>16</sup> ;
- un port de test JTAG.

Ce micro-contrôleur est réalisé en technologie CMOS à 0,6  $\mu\text{m}$  et deux fréquences d'horloge sont disponibles: 33 et 50 MHz.

Pour plus d'informations au sujet du TSC701, voir [21].

D'autres micro-contrôleurs sont prévus pour compléter la famille SPARClet :

- le **TSC701LV** qui est une version limitant la dissipation thermique (prévu début 1997) ;
- le **TSC701SW** dont la technologie CMOS à 0,35  $\mu\text{m}$  permet une fréquence de 70 MHz (prévu pour la mi-1997) ;
- le **TSC702**, version du TSC701SW optimisée pour les communications ATM<sup>17</sup> (prévu pour la fin 1997) ;

---

10. Synchronous Dynamic Random Access Memory.

11. *Digital Signal Processor*.

12. *High-level Data Link Control*.

13. *First In, First Out*.

14. Universal Synchronous/Asynchronous Receiver and Transmitter.

15. *Pulse Code Modulation*.

16. Peripheral Interface Adapter.

17. Asynchronous Transfert Mode.

- le **TSC711** qui est destiné au marché des contrôleurs de communication d'entrée de gamme
  - le CPU sera adapté pour exécuter du code 8 bits comme *JAVA* (prévu pour la fin 1997) ;



## Chapitre 5

# Les microprocesseurs x86

L'architecture CISC d'*Intel* est probablement la plus répandue des architectures de microprocesseurs. Sa domination écrasante du marché des ordinateurs personnels, ses querelles juridiques avec l'ensemble de ses concurrents (*Cyrix*, *AMD*<sup>1</sup>) développeurs de clones ainsi que sa « mésaventure » avec le diviseur flottant du microprocesseur Pentium ont contribué à faire connaître au grand public le nom de cette architecture.

*Intel* est à l'origine de l'architecture x86, mais cette société doit faire face à deux concurrents *AMD* et *Cyrix*. Ceux-ci développent des clones de microprocesseurs d'architecture x86, et petit à petit s'introduisent dans tous les secteurs du marché des PC (leurs derniers microprocesseurs rivalisent avec ceux du haut de gamme *Intel*). Cependant, certains accords sont passés, notamment entre *Intel* et *AMD* en ce qui concerne MMX, l'extension multimédia du jeu d'instructions.

Nous étudions brièvement dans un premier temps les caractéristiques de l'architecture x86 qui a été développée dans [1] en détaillant les nouveautés (*MMX* entre autres).

Puis nous abordons la famille Pentium avec les produits proposés par *Intel*, mais aussi *AMD* et *Cyrix*. Ensuite, nous exposons le PentiumPro d'*Intel* et les versions d'*AMD* et de *Cyrix* de même génération. La partie suivante comporte un tableau récapitulatif des microprocesseurs d'architecture x86 actuels (TAB. 5.3, p. 112).

Enfin, nous faisons un rapide survol des produits qui n'en sont encore qu'au stade de projets.

### Remarque :

Les microprocesseurs d'architecture x86 antérieurs au Pentium ne seront pas référencés ici car ils ne sont plus représentatifs de l'état de l'art actuel de cette architecture et tendent à disparaître du marché.

## 5.1 L'architecture x86

### 5.1.1 Jeu d'instructions

#### Types de données

Les types de données fondamentaux de l'architecture *Intel* sont les octets (8 bits), les mots (16 bits), les doubles-mots (32 bits) et les quadruples-mots (64 bits). Leur placement en mémoire est de type *little-endian*.

Les entiers peuvent être signés ou non-signés et sur 8, 16 ou 32 bits. Ils peuvent aussi être *BCD*<sup>2</sup>. Ces derniers sont des entiers non-signés sur 4 bits dont les valeurs valides sont comprises entre 0 et 9.

---

1. *Advanced Micro Devices*.

2. Binary-Coded Decimal.

Ceux-ci peuvent être *empaquetés* (deux entiers BCD par octet) ou *non-empaquetés* (un entier BCD occupe un octet et les 4 bits de poids fort sont inutilisés).

Les données supportées par l'unité flottante sont les suivantes (conformes aux normes *IEEE 754* et *854*) :

- les réels (réel simple sur 32 bits, réel double sur 64 bits et réel étendu sur 80 bits) ;
- les entiers signés (« entier mot » sur 16 bits, entier court sur 32 bits et entier long sur 64 bits) ;
- entiers BCD empaquetés et signés (sur 10 octets, le bit de poids fort est un bit de signe) ;

## Modes d'adressage

Le jeu d'instructions CISC permet à la plupart des instructions d'accéder aux données directement en mémoire.

Les quatre modes d'adressage courants (absolu, indirect, basé et indexé) sont supportés par l'architecture x86. Celle-ci fournit aussi un mode d'adressage supplémentaire, l'adressage basé indexé, qui fournit en fait de nombreux autres modes d'adressage (combinaisons possibles). L'adresse de la donnée accédée résulte du calcul suivant (il concerne l'*offset*, voir p. 91) :

$$\text{offset} = \text{base} + (\text{index} \times \text{facteur d'échelle}) + \text{déplacement}$$

La base et l'index sont des valeurs contenues dans des registres ; le facteur d'échelle et le déplacement sont des valeurs immédiates. Les combinaisons possibles sont obtenues par l'absence d'une ou plusieurs composantes de l'équation ci-avant (on peut avoir un *offset* correspondant par exemple à *base*, *base + déplacement*, ...).

Le mode d'adressage basé indexé facilite, selon *Intel*, l'adressage de tableaux à deux dimensions.

## Format des instructions

L'architecture x86 est une architecture CISC, ainsi, les instructions sont nombreuses et ont une taille variable. De ce fait, à travers des champs optionnels, de nombreux formats existent.

La taille des instructions peut varier d'un octet à 15 octets. Toutes les instructions se conforment cependant à un schéma donné, exposé en figure 5.1.

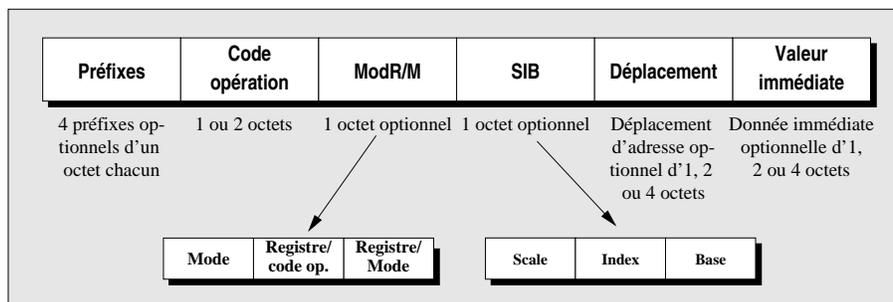


FIG. 5.1 – *Formats des instructions de l'architecture x86.*

Les *préfixes* permettent de compléter l'opération réalisée par l'instruction. Certains apportent une information supplémentaire quant au mode de fonctionnement, la taille des opérands (16 ou 32 bits), ...

Le *code opération* comporte le code de l'opération à réaliser.

Les champs *ModR/M* et *SIB* contiennent des informations sur le mode d'adressage utilisé (quel mode et accès de la donnée dans les registres ou en mémoire), pointent sur les registres quand ceux-ci interviennent et contiennent le facteur d'échelle (*Scale*) nécessaire au mode basé indexé.

La présence des champs *déplacement* et *valeur immédiate* est dépendante du mode d'adressage utilisé pour l'instruction.

Toutes les instructions ne réfèrent au plus que deux opérandes : le résultat est donc stocké à l'emplacement d'un des deux opérandes (jeu d'instruction dit destructeur).

## Types d'instructions

Les instructions de l'architecture x86 se divisent en plusieurs groupes dont voici les principaux :

- Instructions de transfert de données – Les données sont transférées sous forme d'octets, de mots, de doubles mots ou de quadruples mots entre la mémoire et les registres ou entre deux registres.
- Opérations sur les entiers – On retrouve les opérations arithmétiques et logiques classiques, ainsi que des décalages, des rotations et des tests de bits ou d'octets. D'autre part, des instructions convertissent les données entières au format BCD pour faire, à l'aide d'instructions arithmétiques binaires, du calcul décimal.
- Instructions de contrôle de flot – Ces instructions permettent les branchements conditionnels et inconditionnels. La condition des branchements conditionnels est située dans le registre *EFLAGS*.
- Instructions flottantes – Le banc de registres flottants est implémenté sous la forme d'une pile de registres de 80 bits. Les instructions sont réparties sur six groupes :
  - instructions de transfert de données (les données transférées entre la mémoire et la pile de registres sont converties au bon format et transférées à l'aide d'une seule instruction) ;
  - instructions arithmétiques (addition, multiplication, division, racine carrée, ... ) ;
  - instructions de comparaison (dont la mise à jour des codes conditions flottants ou entiers ; ces derniers sont nécessaires pour réaliser un saut conditionné par le résultat d'une instruction flottante car il n'existe pas d'instruction de contrôle de flot flottante) ;
  - instructions de calculs « complexes » ou « transcendantaes » qui réalisent des fonctions trigonométriques, logarithmiques ou hyperboliques ;
  - instructions de chargement de constantes (telles que 0, 1,  $\pi$ , ... ) ;
  - instructions de contrôle pour la gestion de l'état et du mode de fonctionnement de l'unité flottante.
- Instructions multimédias – Ces instructions sont regroupées sous l'appellation *MMX*<sup>3</sup>. Elles sont destinées à accélérer les traitements concernant le son et l'image, et constituent la plus importante modification apportée au jeu d'instruction x86 depuis 1985. MMX a été conçu de manière à s'intégrer facilement et pour cela n'a pas d'impact sur le système d'exploitation. Pour éviter de compliquer encore l'architecture, MMX n'introduit pas de nouveaux modes de fonctionnement, de nouveaux états, de nouvelles interruptions ni de nouvelles exceptions. D'autre part, huit nouveaux registres de 64 bits sont créés (MM0-MM7), ils correspondent physiquement aux registres flottants. Comme aucun mode de fonctionnement n'est associé à l'exécution de code MMX, rien n'empêche un mixage destructeur d'instructions flottantes et MMX, malgré quelques précautions prises par *Intel* (quand une donnée est chargée dans un registre MM, tous les registres flottants sont marqués « occupés », ce qui génère une exception en cas d'exécution d'instructions flottantes ; à la fin de la routine MMX, l'instruction *EMMS* doit être insérée pour libérer les registres).

Les quelques 57 nouvelles instructions (en réalité 57 codes opération) permettent une approche SIMD des calculs. En effet, trois nouveaux types de données 64 bits sont définis : huit *octets empaquetés* (*packed bytes*), quatre *mots empaquetés* (*packed words*) et deux *doubles mots empaquetés*

3. Selon Intel, ce n'est pas un acronyme pour *MultiMedia eXtension*, mais une marque à part entière.

(*packed double words*). Les instructions du jeu d'instructions MMX sont listées et explicitées dans le tableau 5.1 (les instructions dites SIMD opèrent sur certains formats de données — octets, mots, ... — en parallèle dans un quadruple mot et stockent le résultat sous la même forme — huit octets, quatre mots, ...).

Groupe	Mnémonique	Description
<b>Transfert de données, empaquetage, « dépaquetage »</b>	MOV[D,Q]	Transfère un [double, quad.] mot depuis ou vers un registre MM.
	PACKUSWB	Empaquète des mots dans des octets avec saturation non-signée <sup>†</sup> .
	PACKSS[WB,DW]	Empaquète des [mots dans des octets, doubles mots dans des mots] avec saturation signée <sup>‡</sup> .
	PUNPCKH[BW,WD,DQ]	Intercala le(s) [4 octets, 2 mots, double mot] de poids fort de chacun des deux registres sources dans le registre destination.
	PUNPCKL[BW,WD,DQ]	Idem que ci-dessus avec le(s) [4 octets, 2 mots, double mot] de poids faible.
<b>Arithmétique</b>	PADD[B,W,D]	Addition SIMD sur des [octets, mots, doubles mots].
	PADDS[B,W]	Addition SIMD avec saturation signée sur des [octets, mots] <sup>‡</sup> .
	PADDUS[B,W]	Addition SIMD avec saturation non-signée sur des [octets, mots] <sup>†</sup> .
	PSUB[B,W,D]	Soustraction SIMD sur des [octets, mots, doubles mots].
	PSUBS[B,W]	Soustraction SIMD avec saturation signée sur des [octets, mots] <sup>‡</sup> .
	PSUBUS[B,W]	Soustraction SIMD avec saturation non-signée sur des [octets, mots] <sup>†</sup> .
	PMULHW	Multiplication SIMD $16 \times 16 \rightarrow 32_{16:31}$ bits (bits de poids fort du résultat ; en relation avec l'instruction suivante).
	PMULLW	Multiplication SIMD $16 \times 16 \rightarrow 32_{0:15}$ bits (bits de poids faible du résultat ; en relation avec l'instruction précédente).
<b>Décalages</b>	PSLL[W,D,Q]	Décalage logique* à gauche sur tous les [mots, doubles mots, quadruples mots].
	PSRL[W,D,Q]	Décalage logique* à droite sur tous les [mots, doubles mots, quadruples mots].
	PSRA[W,D]	Décalage arithmétique** à droite sur tous les [mots, doubles mots].
<b>Logique</b>	PAND	ET logique bit à bit.
	PANDN	NON ET logique bit à bit.
	POR	OU logique bit à bit.
	PXOR	OU exclusif logique bit à bit.
<b>Comparaisons</b>	PCMPEQ[B,W,D]	Comparaison d'égalité entre les [octets, mots, doubles mots].
	PCMPGT[B,W,D]	Comparaison de supériorité (« plus grand que ») entre les [octets, mots, doubles mots].
<b>Divers</b>	EMMS	État MMX vide.

<sup>†</sup>. Si un débordement survient, le résultat est positionné à la valeur maximum. Le bit de signe n'est pas pris en compte.

<sup>‡</sup>. Si un débordement survient, le résultat est positionné à la valeur maximum ou à la valeur minimum (selon le signe *overflow* ou *underflow*). Les bits de signe sont conservés.

\*. Insertion de zéros.

\*\*. Insertion de bits égaux à celui du signe.

TAB. 5.1 – Les instructions de l'extension multimédia MMX.

– Instructions diverses – Diverses instructions dont quelques unes typiques d'un jeu d'instruction CISC sont proposées :

- instructions d'opérations sur les chaînes de caractères (lecture, écriture, comparaison, déplacement de chaînes) ;

- instructions d'entrées/sorties qui permettent des transferts de données entre les ports d'entrées/sorties du processeur et la mémoire ou les registres ;
- instructions pour les langages structurés qui permettent une gestion à plus haut niveau des appels et retours de procédures (voir [1]) ;
- instructions de manipulation de registres tels le registre *EFLAGS* et les registres de segments ;
- instructions « système » essentiellement destinées au support des systèmes d'exploitation.

### 5.1.2 Modes de fonctionnement

L'architecture Intel x86 supporte trois modes de fonctionnement qui déterminent les instructions accessibles et certaines caractéristiques architecturales :

- **Mode protégé** – C'est le mode *natif* : toutes les instructions et les fonctionnalités de l'architecture sont disponibles.
- **Mode d'adressage réel** (*Real-address mode*) – Ce mode est équivalent à l'environnement de programmation du microprocesseur *Intel* 8086 avec seulement quelques extensions.
- **Mode de gestion système** – Permet le contrôle de la consommation et la sécurité du système avec un espace d'adressage et des instructions particuliers.

### 5.1.3 Registres

L'architecture x86 prévoit trois types de registres entiers :

- Les huit registres dits « généraux » sont utilisés par l'ensemble des opérations entières (logiques, arithmétiques, calcul d'adresses). Ils ont une taille de 32 bits, mais pour assurer une compatibilité binaire ascendante, les quatre premiers peuvent être utilisés pour les opérandes 16 bits (une compatibilité limitée est aussi assurée pour les microprocesseurs 8 bits). On remarquera que des instructions n'utilisent que certains registres afin d'écourter la phase de décodage.
- Les six registres de segments contiennent les sélecteurs de segments (sur 16 bits) associés aux différentes formes d'accès mémoire. À tout moment, six segments de la mémoire sont immédiatement disponibles. Les registres de segments conservent donc le sélecteur associé à chacun de ces segments. Chaque registre est associé à un type de segment particulier.
- Les deux registres de contrôle et de statut permettent la modification de l'état du microprocesseur (codes condition notamment, dans le registre *EFLAGS*) .

Les registres flottants forment aussi trois groupes :

- huit registres de calcul de 80 bits, implémentés sous forme de pile ;
- trois registres de 16 bits contenant contrôle, statut et étiquette de l'unité flottante (en fait ce dernier indique le contenu de chacun des huit registres flottants) ;
- des registres pointeurs d'erreurs.

### 5.1.4 Organisation de la mémoire

Une adresse issue du programme est appelée *adresse logique*. Des mécanismes traduisent cette adresse logique en une adresse vers un espace d'adressage continu ; celle-ci porte le nom d'*adresse linéaire*. Puis des mécanismes de *pagination* convertissent cette adresse linéaire en une *adresse physique*.

Trois modèles de mémoire sont proposés pour obtenir une adresse linéaire :

- un **modèle de mémoire uniforme** ou non segmenté. Ce modèle de mémoire est le plus simple puisqu'ici l'adresse linéaire est confondue avec l'adresse logique (le code, les données et la pile de procédures se partagent le même espace d'adressage).
- un **modèle de mémoire segmenté**. Cette organisation découpe l'espace d'adressage en 16 383 *segments* qui sont autant d'espaces d'adressage indépendants et de tailles variables. Dans ce cas-ci, l'adresse logique est composée d'un descripteur de segment (situé physiquement dans un *registre de segment* pour le segment courant) et d'un déplacement dans ce segment codé sur 32 bits (chaque segment peut avoir une taille maximum de  $2^{32} = 4$  Goctets) ou 16 bits (mode 16 bits).
- un **modèle en « mode d'adressage réel »** (*real-address mode*). Celui-ci est utilisé pour assurer une compatibilité avec le processeur 8086. Pour plus de détails, voir [22].

L'adresse linéaire peut ensuite être directement utilisée pour pointer vers un vaste espace d'adressage physique, ou alors un mécanisme de *pagination* peut être utilisé pour simuler cet espace à partir d'une petite quantité de mémoire vive et d'espace disque (mémoire virtuelle). Quand la pagination est utilisée, l'adresse linéaire est convertie en une adresse physique ou une exception est générée. L'exception permet au système d'exploitation de charger la page concernée à partir du disque. L'architecture x86 impose une taille de pages fixe (4 Koctets).

## 5.2 Le Pentium et les compatibles Pentium

### 5.2.1 La famille Pentium d'Intel

#### Architecture Pentium

Le premier Pentium date de début 1993. Ce microprocesseur était destiné à remplacer le i486 sur le marché des PC. Avec une architecture superscalaire de degré deux, l'introduction d'une prédiction de branchement et l'amélioration de l'unité flottante, le microprocesseur Pentium constitue une profonde évolution par rapport à son prédécesseur.

#### Séquencement et exécution des instructions

Le Pentium présente deux types d'exécution des instructions :

- Les instructions simples les plus fréquemment utilisées ont un contrôle câblé. Ces instructions peuvent être pipelinées et couplées. Elles incluent la plupart des instructions présentes dans un jeu d'instructions RISC.
- Toutes les instructions complexes (celles qui permettent d'assurer la compatibilité ascendante dans la famille de microprocesseurs x86) sont gérées par un microcode (l'unité de contrôle est intégrée dans un « ordinateur miniature » comprenant une unité arithmétique, de la mémoire, un compteur ordinal, etc).

Le Pentium utilise deux pipelines entiers indépendants : le *pipeline U* qui peut exécuter toutes les instructions entières et flottantes, et le *pipeline V* qui ne peut exécuter que les instructions entières dites simples (câblées) ainsi que les instructions d'échange de registres flottants (*FXCH*).

Les deux pipelines ont une profondeur de cinq étages pour l'exécution de code entier. À noter que sur ce pipeline, comme sur celui du MIPS R8000, le calcul de l'adresse des opérandes est effectué un cycle plus tôt que les opérations arithmétiques. Ceci permet de s'affranchir de la pénalité de « load-use » des pipelines plus classiques dans lesquels l'exécution des opérations arithmétiques a lieu plus tôt.

Ce type de pipeline augmente toutefois la pénalité de branchement d'un cycle ; il est donc préférable d'accompagner celle-ci d'un mécanisme de prédiction :

Le Pentium implémente un tampon d'adresses de branchement (*BTB* pour *Branch Target Buffer*) associatif par ensembles à quatre voies, d'une capacité de 256 adresses cibles. Deux tampons de préchargement des instructions cibles viennent compléter ce dispositif. Quand le *BTB* détecte une instruction de branchement parmi les instructions préchargées, il prédit à partir d'un algorithme à deux bits si le branchement est pris ou non. Dans le cas d'une prédiction positive, le *BTB* fournit l'adresse cible sous forme d'adresse physique et permet ainsi de précharger les instructions dans l'un des deux tampons.

Le pipeline flottant du Pentium comporte huit étages dont cinq sont partagés avec le pipeline entier U.

L'émission des instructions dans le Pentium respecte toujours l'ordre du flot d'instructions. Le pipeline U exécute obligatoirement la première instruction de la paire. Il n'est possible d'émettre deux instructions dans le même cycle que sous certaines conditions :

- les deux instructions doivent être « simples » (câblées) ;
- elles ne doivent pas être dépendantes entre elles ;
- les modes d'adressage ne doivent pas être trop complexes (pas d'*immédiat* combiné avec un *déplacement*) ;
- la plupart des instructions avec préfixes doivent être traitées dans le pipeline U.

Ainsi, deux instructions entières peuvent être émises en même temps si elles respectent les règles ci-dessus. Une instruction flottante ne peut être émise avec une instruction entière, mais deux instructions flottantes peuvent être émises par cycle si la deuxième est l'instruction *FXCH* (qui échange le contenu d'un registre avec celui du haut de la pile).

### Hierarchie mémoire

Le Pentium dispose de deux caches primaires distincts pour les instructions et les données, d'une capacité de 8 Koctets chacun. Ces deux caches sont associatifs par ensembles à deux voies et utilisent une politique de remplacement de type *LRU*. Le cache d'instructions n'est accessible qu'en lecture<sup>4</sup>, alors que le cache de données est accessible en lecture et écriture. Les deux techniques de mise à jour de la mémoire peuvent être utilisées (écriture simultanée ou différée).

Le Pentium n'intègre pas de contrôleur de cache secondaire, mais *Intel* propose deux composants : le *82496* qui fait office de contrôleur de cache et le *82491*, un composant de mémoire SRAM spécialement conçu pour cette fonction. L'organisation du cache secondaire est associative par ensembles à deux voies et sa taille peut être de 256 Koctets ou 512 Koctets. Pour répondre plus rapidement, ce cache utilise la technique dite *MRU*<sup>5</sup> : la donnée adressée le plus récemment dans un ensemble est disponible à la fin du premier cycle. En cas de mauvaise prédiction, la donnée référencée est disponible à la fin du second cycle<sup>6</sup>.

La technique de mise à jour de la mémoire, préconisée pour le cache secondaire est la politique d'écriture différée (*write-back*), toutefois il est possible de l'utiliser comme un cache à écriture simultanée.

4. Dans le cas où un code veut accéder à une instruction en écriture (*self modifying code*), il ne peut pas, comme c'était le cas avec les microprocesseurs x86 précédents, modifier directement la ligne dans le cache d'instructions. La ligne d'instructions adressée est au préalable invalidée puis l'écriture est effectuée en mémoire.

5. *Most Recently Used*.

6. L'information *MRU* de chaque ensemble doit être répliquée dans chaque composant mémoire. Ceci explique, entre autres, la nécessité de composants mémoire spécifiques.

À noter que d'autres fondeurs proposent des solutions légèrement différentes pour le cache secondaire.

Le bus mémoire est, en principe, d'une largeur de 64 bits mais peut être étendu à 128 bits. Sur celui-ci, une protection par parité est mise en place.

### Support système d'exploitation

L'organisation de la mémoire du Pentium est conforme à celle décrite en p. 91 (segmentation, pagination).

Deux TLB, associatifs par ensembles à quatre voies sont implémentés pour les instructions et les données. La capacité de chacun d'entre eux pour des pages de 4 Koctets est de 64 entrées pour les données et 32 entrées pour les instructions. Un certain nombre d'entrées additionnelles permettent par ailleurs d'utiliser des pages de 4 Moctets.

Les systèmes d'exploitation supportés par le Pentium sont (entre autres) :

- MS-DOS ;
- Windows et Windows 95 ;
- Windows NT ;
- OS/2 ;
- Unix ;
- Solaris ;
- NextStep.

### Support multiprocesseurs

Le Pentium assure la cohérence des caches primaires et secondaires (utilisation de composants spécifiques) à travers le protocole MESI.

## **Diverses versions du Pentium d'Intel**

### Le P5

Le P5 est le premier microprocesseur Pentium. Il est conforme à ce qui a été exposé ci-avant. La technologie de fabrication du P5 est BiCMOS à 0,8  $\mu\text{m}$ . Les quelques 3,1 millions de transistors logent dans un composant de 296 mm<sup>2</sup>. Alimenté en 5 V, il est proposé à l'origine aux fréquences de 60 et 66 MHz.

### Le P54C[S]

Le P54C est la seconde version du Pentium (premier trimestre 1994). Outre une technologie physique revue, celui-ci inclut quelques améliorations :

- un **contrôleur d'interruptions** (*APIC*<sup>7</sup>) qui permet une configuration à deux P54C avec un cache secondaire partagé, sans logique supplémentaire ;
- une **gestion de la consommation** ;
- un **multiplieur d'horloge** (PLL<sup>8</sup>) qui permet au CPU d'être cadencé 1,5 ou 2 fois plus vite que le bus système.

---

7. *Advanced Priority Interrupt Controller.*

8. *Phase-Locked Loop.*

La fabrication du P54C est faite en BiCMOS à 0,6  $\mu\text{m}$ . Bien que le nombre de transistors soit légèrement supérieur au P5 (3,3 millions contre 3,1 millions), le P54C a une taille de 163  $\text{mm}^2$ . Une autre caractéristique importante du P54C est une alimentation en 3,3 V (contre 5 V pour le P5) qui permet de réduire la dissipation thermique tout en élevant la fréquence interne à 75, 90 et 100 MHz. À noter que le brochage du P54C est différent de celui du P5.

Une version encore améliorée (processus BiCMOS à 0,35  $\mu\text{m}$ ), le **P54CS**, a permis en 1995 d'accéder à des fréquences de 133, 150, 166 et 200 MHz (mi-1996).

### Le P55C

Disponible depuis janvier 1997, le P55C est le dernier microprocesseur d'*Intel* à être sorti à la date de rédaction de ce rapport (mai 1997). La plus grande modification par rapport à ses prédécesseurs concerne l'exécution de code MMX. Son nom est d'ailleurs officiellement « **Pentium Processor with MMX Technology** » (processeur Pentium avec technologie MMX). Ce microprocesseur se positionne dans le haut de gamme des Pentiums puisque ses performances en sont améliorées. D'un autre côté il sera certainement le dernier Pentium à être conçu avant la suprématie du PentiumPro sur le marché des PC (prévue en 1998 par *Intel*).

La figure 5.2 représente le schéma du P55C, celui-ci correspond à l'état de l'art en matière de Pentium. Notons que les parties grisées correspondent aux modifications apportées au P54C, et les parties noires sont nouvelles.

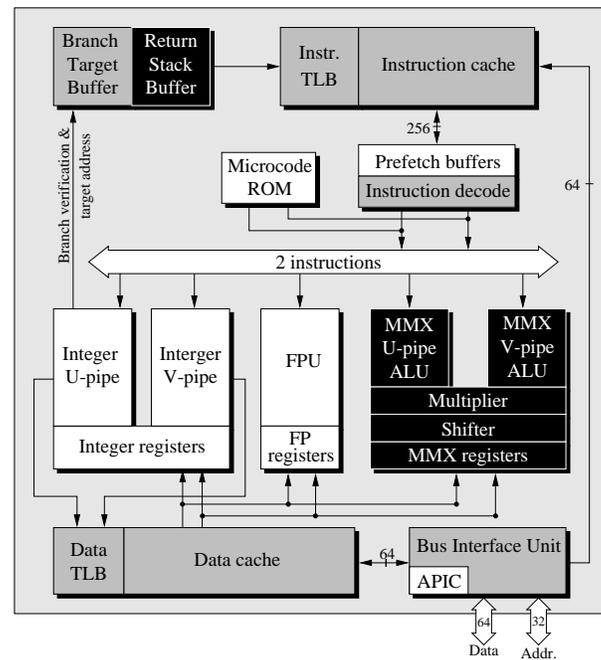


FIG. 5.2 – Synoptique du P55C (Pentium avec MMX).

L'intégration des instructions MMX dans le P55C commence tout d'abord avec une révision du décodeur afin que celui-ci puisse assurer un débit convenable. D'autre part, l'allongement du pipeline d'un cycle lui donne plus de temps pour le décodage des instructions les plus complexes (ceci permet aussi l'élargissement de chemins critiques). Les pipelines du P55C sont visibles figure 5.3.

Une phase supplémentaire de pipeline augmente la pénalité de mauvaise prédiction de branchement d'un cycle. Pour cela, l'algorithme mis en œuvre sur le P55C (similaire à celui du PentiumPro) est celui, plus récent que celui de Smith, proposé par Yeh et Patt (voir glossaire p. 152). Il est adaptatif

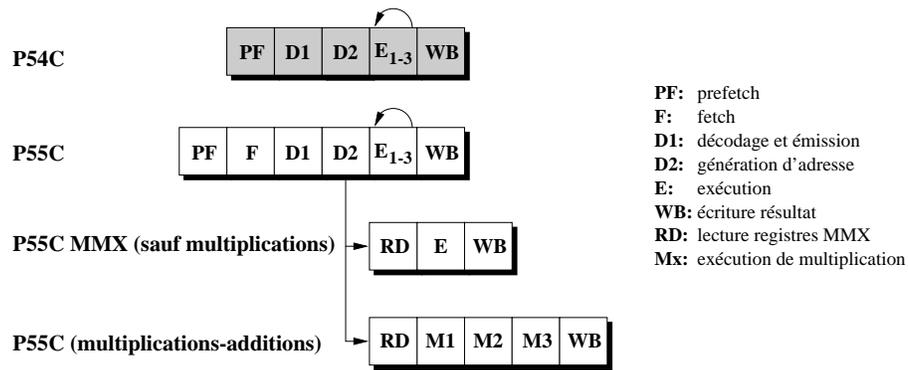


FIG. 5.3 – Les pipelines du P55C.

à deux niveaux. Pour les retours de procédures, une pile de quatre entrées est implémentée (*Return Stack Buffer*).

Une des grandes nouveautés amenées par le P55C concerne l'unité MMX. Celle-ci est distincte de la FPU et comporte deux ALU identiques. Deux instructions MMX peuvent être exécutées à la fois si une seule au plus accède à la mémoire ou aux registres entiers, ou réalise une multiplication, un décalage ou une des deux instructions PACK/UNPACK (voir p. 90). À noter que contrairement aux ALU entières, celles-ci peuvent toutes deux (à tour de rôle) faire des multiplications ou des décalages : le pipeline-U n'en a pas l'exclusivité.

Bien que les registres flottants et MMX soient physiquement distincts, ils restent logiquement assimilés. De ce fait, les instructions flottantes et MMX ne peuvent pas être mêlées dans le code, et ces deux types d'instructions ne s'exécutent pas en parallèle.

La taille des caches primaires a été multipliée par deux (16 Koctets chacun au lieu de 8 Koctets) et ceux-ci sont associatifs par ensembles à quatre voies (au lieu de deux voies pour le P54C).

Le tampon d'écriture mémoire passe de deux entrées à quatre. Les deux TLB du P55C sont associatifs et supportent des pages de 4 Koctets ou de 4 Moctets, éliminant ainsi l'utilité d'un TLB particulier pour des pages de 4 Moctets.

Les 4,5 millions de transistors du P55C occupent une surface de 140 mm<sup>2</sup> (contre 90 mm<sup>2</sup> pour le P54C) malgré un passage à une technologie CMOS (et non BiCMOS comme pour le P54C) à 0,28 μm. L'alimentation se fait en 2,8 V (au lieu de 3,3 V pour le P54C, mais les brochages sont compatibles). Les fréquences proposées sont 166 et 200 MHz.

## 5.2.2 Le 6x86 (M1) de Cyrix

Ce microprocesseur, appelé 6x86 (pour sixième génération de microprocesseurs x86) ou M1, fait appel à des techniques avancées d'architecture telles que prédiction de branchement, exécution spéculative et dans le désordre (limitée) des instructions et renommage de registres. Sorti fin 1995, il concurrence directement la famille des Pentiums d'*Intel*. Il est superscalaire de degré deux et comporte cinq unités fonctionnelles : une unité entière, deux caches primaires, une unité de gestion de la mémoire, une unité flottante ainsi qu'une interface bus.

## Séquencement et exécution des instructions

### Pipelines

La figure 5.4 schématise les deux pipelines d'exécution que l'on trouve sur ce microprocesseur. Les pipelines incluent des étages séparés pour la lecture des données, la phase de calcul et enfin l'étape de

mise à jour de la mémoire pour permettre des mécanismes de *bypass* lors de séquences d'opérations dépendantes.

D'autre part, les deux pipelines ne sont pas symétriques. Le « pipeline X » est le seul à gérer les branchements ainsi que les instructions flottantes.

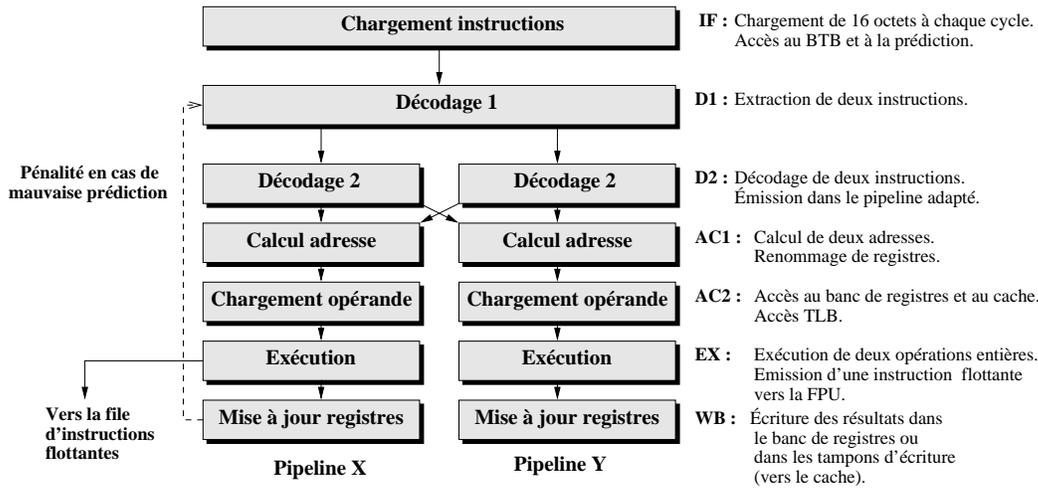


FIG. 5.4 – Pipelines du microprocesseur Cyrix 6x86.

### Prédiction de branchement

La prédiction de branchement (nécessaire avec un pipeline de cette taille) est identique à celle du Pentium : BTB (*Branch Target Buffer*) de 256 entrées, associatif par ensembles à quatre voies et prédiction à deux bits. Le microprocesseur de *Cyrix* implémente également une pile d'adresses de retours de procédures de huit entrées.

### Unité flottante

L'unité flottante n'est pas pipelinée. Les instructions sont envoyées vers une file d'instructions de quatre entrées. Une gestion précise des exceptions est maintenue.

### Renommage de registres et « bypass de la mémoire »

Le petit nombre de registres prévus par l'architecture x86 impose de nombreux accès mémoire qui retardent la disponibilité des données. Ceci crée en outre de fausses dépendances de données. Une solution classique à ce second problème est de mettre en œuvre un mécanisme de renommage de registres. La solution choisie par *Cyrix* implémente 32 registres physiques pour les huit registres logiques vus par le programmeur.

Le *bypass de la mémoire* est également une technique qui permet de réduire les effets du nombre limité de registres sur les microprocesseurs x86. Comme la plupart des microprocesseurs, le 6x86 fournit directement la donnée qui a été calculée (ou lue) au cycle précédent à l'unité fonctionnelle, sans passer par le banc de registres. Mais le 6x86 va au-delà en permettant le *bypass* même si la donnée est écrite en mémoire plutôt que dans le banc de registres. Un tel mécanisme prend tout son sens dans un microprocesseur d'architecture CISC.

### Exécution des instructions dans le désordre

L'exécution dans le désordre concerne ici l'exécution d'instructions postérieures à une instruction qui attend une résolution de défaut de cache. Dans ce cas-là, les instructions qui s'exécutent doivent toutes utiliser le même pipeline, alors que celle qui est bloquée utilise l'autre. Ainsi pour ne pas bloquer le pipeline-X qui est chargé des branchements et des opérations flottantes, les instructions d'accès à la mémoire seront plutôt dirigées vers le pipeline-Y.

Les instructions qui s'exécutent dans le désordre sont repérées avant leur émission pour assurer une gestion précise des exceptions et un modèle ordonné d'exécution. Pour cela, les accès au bus se font uniquement dans l'ordre.

## Hiérarchie mémoire

Cyrix a fait le choix d'un cache unifié de 16 Koctets (même capacité que les deux caches du Pentium réunis) couplé à un petit cache d'instructions de 256 octets servant à limiter les problèmes liés à l'accès concurrent aux instructions et aux données.

Le cache d'instructions, totalement associatif, utilise un *faux* double port<sup>9</sup> permettant une lecture et une écriture à chaque cycle. Pour gérer les accès en cas de conflits, le 6x86 utilise un algorithme qui alloue la plus haute priorité au chargement d'instructions puis aux lectures, puis aux écritures qui sont temporairement conservées dans des tampons.

Un TLB unifié à deux ports permet de générer deux adresses physiques à chaque cycle. Il a une capacité de 128 entrées et est à correspondance directe.

## Interface bus

Le bus système du 6x86 peut être cadencé à des fréquences deux ou trois fois moins hautes que celles du CPU. Ainsi pour un 6x86 à 150 MHz, la fréquence du bus peut être de 50 MHz ou 75 MHz.

## Technologie

Les différentes versions du 6x86 sont produites par *IBM* et *SGS-Thomson*. Conçu initialement en technologie CMOS à 0.65  $\mu\text{m}$ , le 6x86 fonctionnait à la fréquence de 100 MHz. Le passage à une technologie à 0.5  $\mu\text{m}$  lui a permis d'atteindre la fréquence de 133 MHz. En juin 1996, *IBM* a annoncé la production du 6x86 à 150 MHz dans une technologie CMOS 0,44  $\mu\text{m}$ . Celui-ci affiche des performances très légèrement supérieures à celles d'un Pentium à 200 MHz.

### 5.2.3 Le K5 d'AMD

Annoncé fin 1994 et commercialisé depuis mars 1996, ce microprocesseur vise au même titre que le 6x86 de *Cyrix*, le marché des Pentiums. Il constitue le premier composant d'AMD entièrement développé à partir d'un cœur *AMD*. En développant ainsi sa propre architecture, *AMD* tentait de se mettre à l'abri de toutes poursuites juridiques d'*Intel*; depuis, certains accords ont aussi eu lieu entre les deux compagnies. Le premier composant de cette série n'a pas donné les performances espérées, c'est pour cela que la première version n'est sortie que 18 mois plus tard.

Le cœur de ce microprocesseur est basé sur une architecture superscalaire de degré quatre permettant une exécution spéculative et dans le désordre, ainsi que le renommage de registres. Cette architecture va au-delà de celle du Pentium et du 6x86 en utilisant un découplage plus poussé et un décodage particulier.

Le K5 a six unités d'exécution : deux ALU non symétriques (seule une possède un décaleur, l'autre disposant d'un diviseur), deux unités de lecture/écriture, une unité de branchement et une unité de calcul flottant.

## Décodage des instructions

La taille variable des instructions x86 (jusqu'à 15 octets) les rend particulièrement délicates à décoder en parallèle. Les concepteurs du K5 ont évité ce problème en permettant un pré-décodage des

---

9. Le cache a une structure de 16 bancs entrelacés de 16 octets chacun permettant deux accès par cycle sur des bancs différents.

instructions au moment de leur chargement dans le cache, qui ajoute cinq bits à chaque octet. Ceux-ci indiquent, entre autre, si l'octet est le début ou la fin d'une instruction x86, ainsi que la localisation des codes opérations et des préfixes.

Après avoir été chargées dans le cache, ces instructions sont converties en une ou plusieurs micro-instructions, appelées selon la terminologie utilisée par *AMD*, « opérations RISC », ou *ROP* (pour *RISC Operations*<sup>10</sup>) dans quatre convertisseurs ROP (quatre instructions peuvent être converties par cycle). Jusqu'à quatre ROP peuvent ensuite être émises à chaque cycle.

La plupart des instructions x86 sont traduites directement en ROP. Les instructions complexes sont converties en de multiples ROP et le flot d'instructions est réarrangé pour qu'il conserve sa cohérence.

Toute instruction pouvant être traduite en une, deux, ou trois ROP est gérée entièrement par matériel. Pour des instructions plus complexes qui nécessitent quatre ROP ou plus, des « séquences ROP » (en fait du microcode) de la *MROM*<sup>11</sup> sont utilisées. La *MROM* peut ensuite émettre quatre ROP par cycle.

### Exécution des instructions.

Une fois les convertisseurs d'instructions x86 en ROP passés, le cœur du K5 est similaire à celui d'un microprocesseur RISC classique à l'exception de certaines caractéristiques destinées à supporter le jeu d'instructions x86 (notamment l'ajout de logique aux deux unités de lecture/écriture comme support aux modes d'adressage complexes x86). On peut remarquer que l'émission de quatre instructions par cycle est basée sur des ROP<sup>12</sup>.

Les ROP sont émises vers des stations de réservation où elles attendent la disponibilité de leurs opérandes. Chaque unité a deux stations de réservation à l'exception de l'unité flottante qui n'en possède qu'une. Le mécanisme d'émission est bloqué dès qu'une des stations de réservation est pleine.

Les branchements sont prédits à partir d'un schéma simple de prédiction à un bit associé à chaque ligne de cache. Ce bit d'historique reflète la dernière direction prise lors de l'exécution du branchement. L'entrée de prédiction pour chaque ligne de cache inclut un pointeur vers l'instruction cible, avec son index de cache et le déplacement dans la ligne. Ce pointeur permet de suivre un branchement sans cycle de pénalité. La pénalité maximale pour une mauvaise prédiction est de trois cycles (soit 12 ROP potentielles).

Un tampon de réordonnancement de 16 entrées récupère les instructions exécutées spéculativement. Tous les résultats sont d'abord inscrits dans ce tampon, puis le banc de registres est mis à jour si la prédiction s'est avérée bonne. Les opérandes peuvent être chargés soit à partir du banc de registres, soit à partir de ce tampon. Le mécanisme de renommage est mis en œuvre au sein même de ce tampon de réordonnancement qui facilite par ailleurs la prédiction des branchements ainsi qu'une gestion précise des exceptions.

Au total, le pipeline comporte six étages, mais seuls cinq comptent au niveau des performances puisque le sixième (le tampon de réordonnancement) ne les affecte pas (les opérandes peuvent y être lus).

L'unité flottante n'inclut pas la traditionnelle pile de registres flottants des microprocesseurs x86. Cette pile est en fait émulée dans le banc de registres généraux à travers un mécanisme spécifique de renommage.

---

10. Les ROP ne sont pas des instructions RISC au sens traditionnel du terme mais en partagent cependant deux caractéristiques principales : une taille constante et un codage des opérations simple et constant.

11. *Microcode ROM*.

12. On peut considérer que cela correspond en moyenne à environ deux instructions x86 à adressage 16 bits ou trois instructions x86 à adressage 32 bits par cycle.

## Hiérarchie mémoire

Deux caches primaires d'instructions et de données sont implémentés sur le K5. Le cache d'instructions a une taille de 16 Koctets et est associatif par ensembles à quatre voies. Le cache de données a une taille de 8 Koctets et est à recopie différée. La cohérence de ce dernier avec le système mémoire est maintenue par le protocole MESI.

## Technologie

Le processus de fabrication des différents modèles est CMOS à 0,35  $\mu\text{m}$ . Plusieurs fréquences sont proposées dans une première gamme : 75, 90 et 100 MHz. Dans une seconde gamme améliorée (élargissement de goulots d'étranglements internes) trois autres fréquences sont proposées : 90, 100 et 116,7 MHz. La version à 116,7 MHz offre des performances comparables à celles d'un Pentium à 166 MHz.

### 5.2.4 Le Nx586 de NexGen

Le Nx586 de *NexGen* est basé sur le même principe que le PentiumPro et le K5. Il traduit les instructions en micro-opérations (elles sont appelées *RISC86 instructions* par *NexGen*) pour les exécuter sur un cœur RISC. Une seule instruction x86 peut être décodée à chaque cycle, mais plusieurs micro-opérations peuvent être envoyées vers les unités d'exécution. Le Nx586 est un microprocesseur scalaire du point de vue CISC, mais superscalaire du point de vue RISC. Les micro-instructions sont lancées dans l'ordre, mais peuvent s'exécuter dans le désordre. Un mécanisme les retire dans l'ordre après la fin de leur exécution.

Le Nx586 intègre trois unités d'exécution, deux unités entières et une unité de calcul d'adresses, mais il n'y a pas d'unité flottante. En effet, *NexGen* a remplacé l'unité flottante par un contrôleur de cache pour le cache secondaire externe. Un second composant, le Nx587, est nécessaire pour traiter toutes les instructions flottantes.

Le Nx586 intègre un cache d'instructions et un cache de données de 16 Koctets chacun, ils sont associatifs par ensembles à quatre voies. Le contrôleur du cache secondaire est intégré au circuit et permet l'interfaçage d'un cache associatif par ensembles à quatre voies.

À noter que le bus externe est incompatible avec ceux des 486 et Pentium d'*Intel*. Il est nécessaire d'ajouter de la logique externe pour pouvoir l'intégrer dans un PC.

En novembre 1995, *NexGen* a annoncé le Nx586 à 120 et 133 MHz. Ce microprocesseur est réalisé par *IBM* en technologie CMOS à 0,44  $\mu\text{m}$ .

Une version légèrement améliorée est sortie en décembre 1995, elle intègre une FPU (en fait, le composant Nx587 proposé par *NexGen*) grâce à la technique *Multi Chip Module (MCM)* d'*IBM* que l'on retrouve sur son P2SC. Cadencé à 93 MHz, le Nx586 offre les performances d'un Pentium à 100 MHz.

En 1996, *AMD* qui avait racheté *NexGen* a décidé d'arrêter la fabrication de ce microprocesseur.

### 5.2.5 Le MediaGx de Cyrix

Prévu depuis octobre 1995 pour la mi-1996, le MediaGX n'est sorti qu'en février 1997. Son originalité n'est pas étrangère à ce délai. En effet ce processeur est un compatible Pentium, et est destiné à s'intégrer dans des PC à bas prix. Pour cela, il intègre un cœur de 5x86 et en particulier un contrôleur de mémoire, une unité graphique et une interface PCI. De ce fait, le composant ne s'intègre pas comme un Pentium sur les cartes. Il a donc fallu que Cyrix convainque les fabricants de cartes d'adopter son microprocesseur et que les composants d'interfaçage, plus spécifiques, soient fabriqués. Ceci explique le retard de la sortie de ce microprocesseur.

Le schéma du MediaGX est visible figure 5.5.

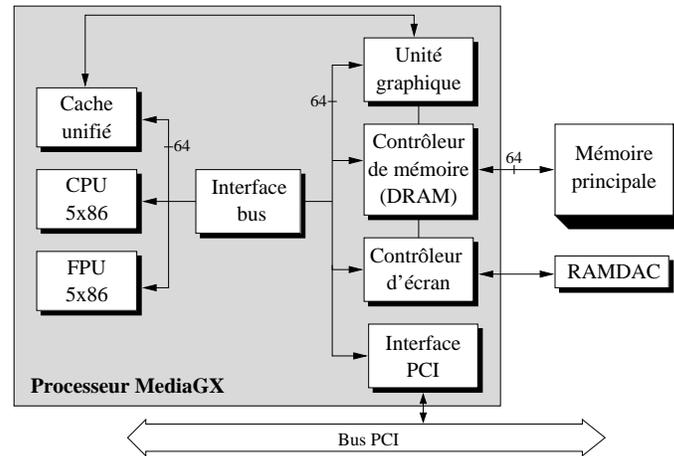


FIG. 5.5 – Synoptique du MediaGX de Cyrix.

## Cœur d'exécution

Le cœur du MediaGX est composé d'une unité entière et de calcul d'adresses 5x86 et d'une unité flottante 5x86 découplée. Le 5x86 est le prédécesseur du M1 détaillé en page 96 (en fait il est dérivé du M1 d'origine avec lequel les concepteurs ont eu de nombreux problèmes). Ce microprocesseur est scalaire et le pipeline comporte six étages (voir FIG. 5.6).

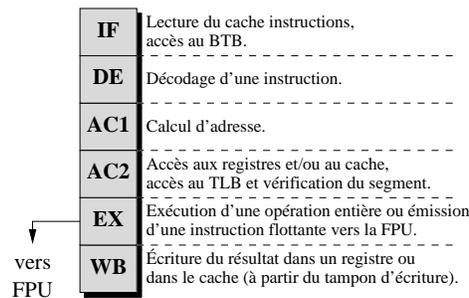


FIG. 5.6 – Le pipeline du MediaGX.

On remarquera que l'étage d'accès au cache est séparé de celui d'exécution. Ceci est intéressant pour les instructions arithmétiques dont une des opérands est en mémoire (pas de pénalité dans le cas où la donnée se trouve dans le cache).

D'un autre côté, un branchement peut engendrer quatre cycles de pénalité. Pour compenser cela, un « branch target buffer (BTB) » de 128 entrées similaire à celui du Pentium est implémenté ainsi qu'un tampon d'adresses de retours de procédures.

L'unité flottante est indépendante et comporte une file d'instructions de quatre entrées qui permet son découplage vis-à-vis de l'unité entière.

## Hiérarchie mémoire

Le MediaGX comporte un cache unifié de 16 Koctets associatif par ensembles à quatre voies. Il est de plus à recopie différée. D'autre part, certaines parties de ce cache peuvent être verrouillées pour conserver des données en rapport avec la *Virtual System Architecture (VSA)*<sup>13</sup>.

Le MediaGX ne peut pas se voir adjoindre un cache secondaire. On peut remarquer que le contrôleur de mémoire permet de lire un mot en mémoire en six cycles (dans le meilleur des cas), latence équivalente à celle d'un Pentium pour accéder à son cache secondaire.

## Unité graphique

Le MediaGX implémente une architecture à mémoire unifiée (*Unified Memory Architecture – UMA*). Ce qui veut dire que la mémoire vidéo (*frame buffer*) est stockée dans la mémoire principale. Pour ne pas trop réduire la bande passante entre le CPU et la mémoire, les concepteurs du MediaGX ont mis en place une technique de compression pour les données graphiques. Finalement, 2 Moctets sont réservés en mémoire pour la mémoire vidéo.

## Contrôleur d'écran

Le contrôleur d'écran interface le processeur avec un écran à travers un convertisseur de type RAMDAC<sup>14</sup>.

## Technologie

Le composant mesure 120 mm<sup>2</sup> et est fabriqué en CMOS à 0,5  $\mu\text{m}$ . Les fréquences disponibles sont 120 MHz et 133 MHz. Une version à 150 MHz est à l'étude et devrait sortir dans la deuxième moitié de 1997.

## 5.3 Le PentiumPro et les compatibles PentiumPro

### 5.3.1 Le PentiumPro d'Intel

Le PentiumPro a été annoncé par *Intel* en décembre 1995. Il dépasse largement les performances de ses prédécesseurs et vise dans un premier temps le marché des PC haut de gamme. Pour cela, il utilise des techniques déjà mises en œuvre par ses concurrents (le K5 d'AMD en page 98 et le Nx586 de *NexGen* en page 100). Les instructions x86 sont traduites dans un premier temps en « micro-instructions », puis exécutées sur une architecture superscalaire découplée (FIG. 5.7) utilisant les techniques de renommage de registres et d'exécution dans le désordre.

À noter: le PentiumPro implémente des instructions *conditional-move* qui permettent d'éviter de nombreux branchements. Ces instructions permettent un déplacement conditionné par le résultat de l'opération précédente (déplacement registre  $\rightarrow$  registre).

## Séquencement et exécution des instructions

### Pipeline et prédiction de branchement

Le PentiumPro utilise un pipeline de 12 étages (FIG. 5.8). Cette profondeur de pipeline s'explique par l'ajout de trois étages et demi pour traduire les instructions en micro-opérations.

13. Le concept de *Virtual System Architecture* est introduit par *Cyrix* afin de maintenir une compatibilité avec les cartes audio et vidéo standards (*VGA*, *Sound Blaster*, ...). Pour cela le MediaGX récupère toutes les tentatives d'accès à ces cartes et les émule (quelques nouvelles instructions utilisées en interne sont supportées).

14. *Random Access Memory Digital-to-Analog Converter*.

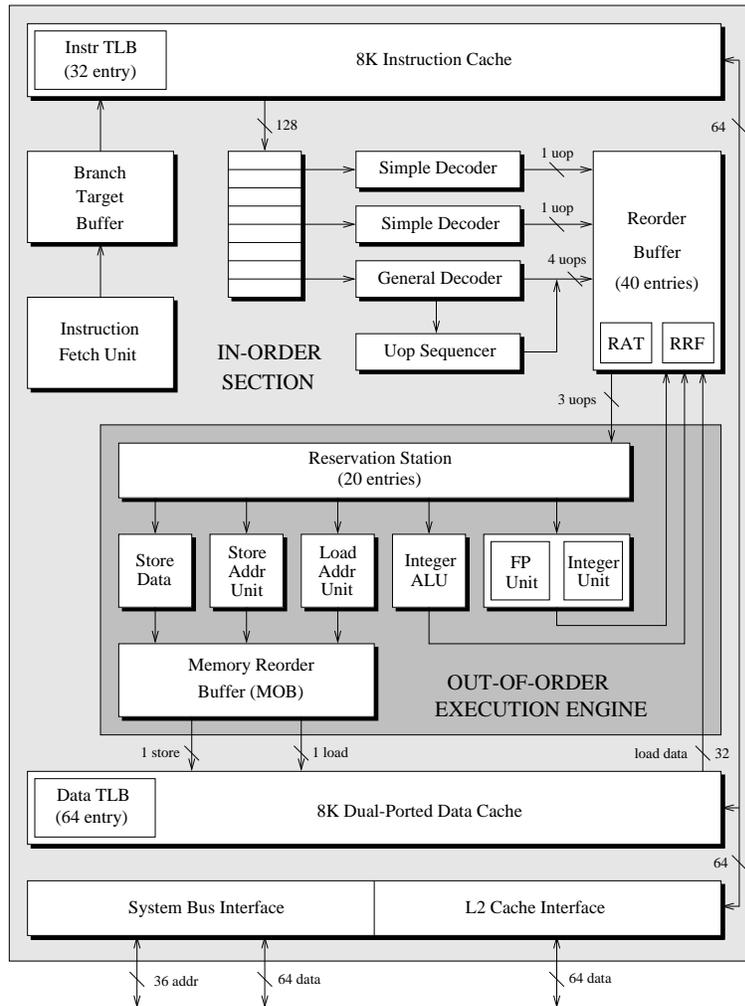
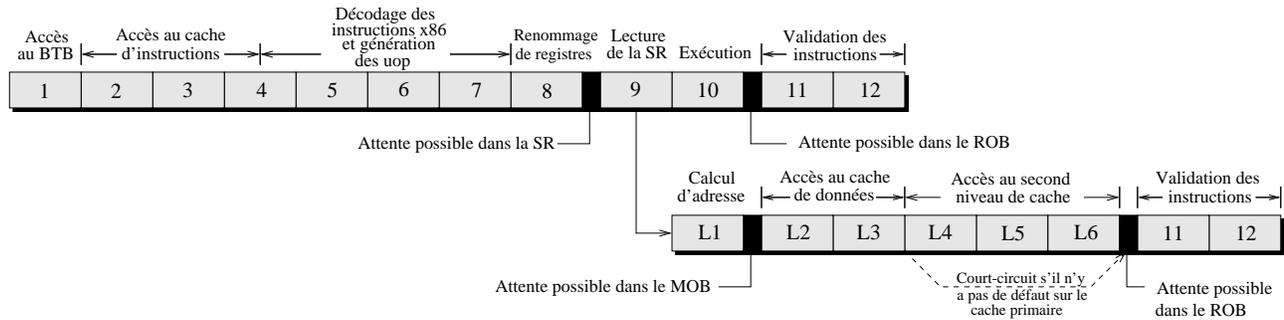


FIG. 5.7 – Architecture du PentiumPro.

L'un des principaux inconvénients d'un pipeline aussi profond est la pénalité liée à une mauvaise prédiction de branchement qui est ici d'au minimum onze cycles. Aussi, le PentiumPro se doit d'avoir un mécanisme de prédiction de branchement particulièrement performant. Il utilise pour cela un BTB de 512 entrées associatif par ensembles à quatre voies. Contrairement à la plupart des microprocesseurs, le PentiumPro n'utilise pas le traditionnel algorithme de Smith à deux bits mais fait appel à un algorithme plus récent, celui de *Yeh and Patt* [23] qui met en œuvre une prédiction adaptative à deux niveaux. Dans le cas où un branchement serait lui-même absent du BTB, une prédiction statique est alors utilisée (branchement prédit pris si déplacement négatif).



**SR** : station de réservation ; **MOB** : *Memory Order Buffer* ; **ROB** : *ReOrder Buffer*.

FIG. 5.8 – Pipelines du PentiumPro.

### Décodage et traduction des instructions en micro-instructions

À chaque cycle d'horloge, le PentiumPro peut charger 16 octets alignés en mémoire, à partir du cache dans un tampon d'instructions. Ce flot d'instructions est envoyé vers trois décodeurs d'instructions qui vont traduire les instructions en micro-opérations. Le premier décodeur peut traduire n'importe quelle instruction et produire jusqu'à quatre micro-opérations. Les deux autres décodeurs se restreignent aux instructions simples se traduisant en une seule micro-opération<sup>15</sup>. À noter que les instructions sont toujours décodées dans l'ordre. Les instructions les plus complexes, décodées par le décodeur général, utilisent un support matériel appelé *Microcode Instruction Sequencer* pour générer le flot de micro-opérations approprié. Ces instructions complexes peuvent occuper le décodeur pendant plusieurs cycles.

Les micro-instructions produites (appelées *uop* par *Intel*) ont une taille constante de 118 bits. Leur structure est régulière pour coder l'opération, les deux opérandes sources et la destination. Comme les instructions RISC, les micro-opérations utilisent un modèle de type *load/store* : les instructions, typiquement CISC, qui ont une opérande en mémoire, sont décomposées en séquences d'instructions RISC (chargement de l'opérande, puis calcul, puis éventuellement écriture mémoire). *Intel* estime qu'une instruction x86 produit en moyenne 1,5 à 2 micro-opérations.

### Séquencement des instructions

Après le décodage, les micro-opérations sont envoyées dans le *ReOrder Buffer (ROB)* ; ce tampon maintient l'ordre sémantique de l'exécution du programme : les micro-opérations sont exécutées dans le désordre mais validées dans l'ordre dans ce tampon. Chacune des 40 entrées du ROB possède un champ servant à mémoriser un résultat ou des codes condition modifiés par la micro-opération.

Les micro-opérations sont ensuite placées dans une station de réservation de 20 entrées. Elles vont y rester jusqu'à ce que leurs opérandes sources soient valides. À chaque cycle, jusqu'à cinq micro-opérations peuvent être lancées : deux opérations de calcul, un chargement depuis la mémoire, une

15. *Intel* ne diffuse pas explicitement la liste des instructions faisant partie de cette catégorie.

écriture mémoire, un calcul d'adresse pour une écriture (il faut deux micro-opérations pour réaliser une écriture mémoire : le calcul de l'adresse et le rangement lui-même).

Dans de nombreuses situations, il y a beaucoup plus de micro-opérations prêtes à être exécutées que d'unités fonctionnelles. Dans ce cas, la logique de lancement utilise des règles de priorité<sup>16</sup>.

## Hiérarchie mémoire

En plus du cache d'instructions et du cache de données de 8 Koctets chacun, le PentiumPro intègre dans le même boîtier (composants distincts) un second niveau de cache (256 ou 512 Koctets) chargé de compenser la faible capacité des caches primaires. L'intégration de ce cache sur le même boîtier le rend beaucoup plus performant. D'autre part, un tampon de réordonnement des accès vers la mémoire (*Memory Order Buffer* — *MOB*) contribue à l'efficacité de cette hiérarchie et assure la cohérence des accès mémoire.

Le cache de données est non-bloquant, il a été conçu de manière à supporter trois accès par cycle : un accès en lecture, un accès en écriture et un accès de mise à jour en provenance du cache secondaire. Le cache secondaire est lui aussi non-bloquant, il est associatif par ensembles à quatre voies.

Les caches primaires d'instructions et de données sont virtuellement adressés mais physiquement testés, le cache secondaire étant quant à lui physiquement adressé et testé. Deux TLB d'instructions et de données (32 et 64 entrées respectivement) totalement associatifs fournissent les traductions d'adresses.

## Bus système et support multiprocesseurs

Le bus système du PentiumPro a été complètement revu. Il comporte toujours 64 bits de large pour les données et est toujours cadencé à 66 MHz maximum, mais il supporte un protocole de transactions imbriquées qui assure ainsi une plus large bande passante.

D'autre part, il est entièrement dédié aux communications avec la mémoire principale ou les entrées/sorties parce que l'interfaçage avec le cache secondaire se fait avec un bus particulier distinct.

Par ailleurs, le bus système supporte jusqu'à quatre microprocesseurs PentiumPro sans logique de contrôle supplémentaire. De plus, on retrouve sur le PentiumPro le contrôleur d'interruptions du Pentium (système *APIC*).

## Technologie

Plusieurs versions existent. Le PentiumPro à 150 MHz est réalisé en technologie 0,5  $\mu\text{m}$ . Le PentiumPro à 200 MHz (11/1995) utilise une technologie BiCMOS à 0,35  $\mu\text{m}$ . Le cœur du microprocesseur occupe 196 mm<sup>2</sup> (5,5 millions de transistors). Le cache secondaire de 256 Koctets réalisé en technologie 0,5  $\mu\text{m}$  occupe une surface de 202 mm<sup>2</sup> (15,5 millions de transistors). Celui de 512 Koctets réalisé en technologie 0,35  $\mu\text{m}$  occupe 242 mm<sup>2</sup> (31 millions de transistors).

### 5.3.2 Le M2 de Cyrix

Annoncé en octobre 1996, le M2 sera disponible à la mi-1997. De par ses performances *espérées*, ce microprocesseur se place dans le haut de gamme des PC en tant que direct concurrent du P55C (p. 95) et du Pentium-II (p. 111).

Le cœur du M2 reste identique à celui du M1 (p. 96) et les techniques employées dans celui-ci se retrouvent dans celui-là (processeur superscalaire de degré deux, prédiction de branchement, exécution spéculative et partiellement dans le désordre, renommage de registres).

---

16. *Intel* ne diffuse pas ces règles, mais indique qu'en général les micro-opérations les plus anciennes ont priorité sur les plus récentes.

Les améliorations apportées par le M2 concernent les fréquences possibles du bus (en fait les fractions de fréquences d'horloge interne, pour le bus) et le support d'extensions au jeu d'instructions (notamment MMX), sans compter les augmentations de tailles des divers tampons et caches. Les parties grisées du schéma 5.9 mettent en valeur les différences entre le M1 et le M2.

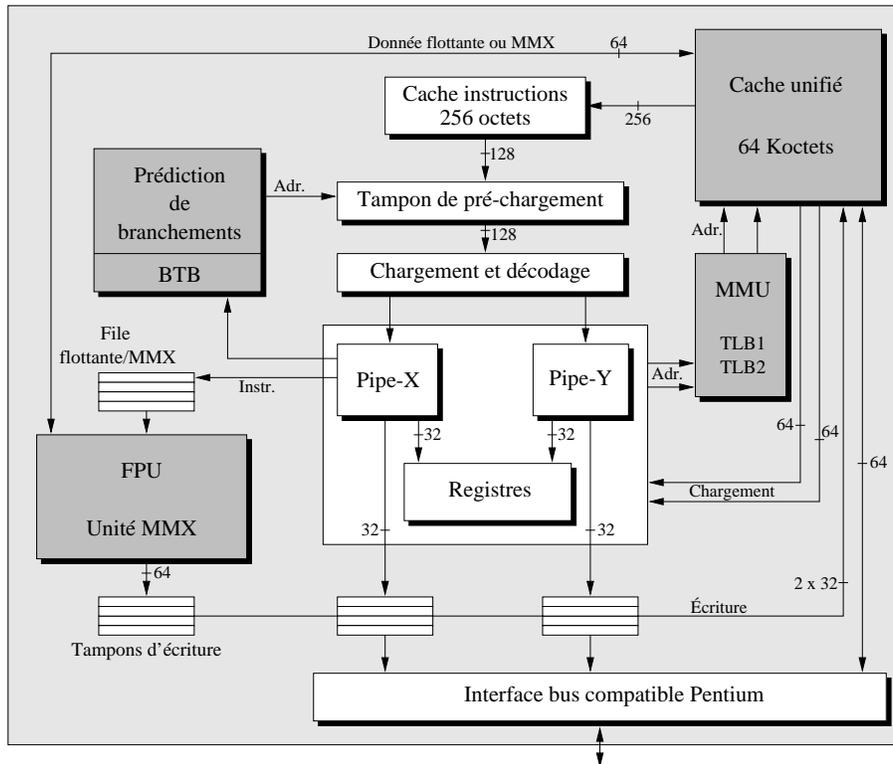


FIG. 5.9 – Synoptique du M2 de Cyrix.

Dans ce qui suit, nous détaillons les points qui ont fait l'objet d'améliorations, sachant que les autres restent identiques à ceux du M1.

## Séquençement et exécution des instructions

### Jeu d'instructions

Le M2 supporte le jeu d'instruction implémenté dans le PentiumPro (notamment les instructions de déplacements conditionnels) ainsi que l'extension MMX. Cette dernière est gérée au niveau de l'unité flottante (voir plus loin).

### Pipeline

Le pipeline du M2 reste identique à celui du M1 (voir FIG. 5.4, p. 97) si ce n'est la prise en compte d'instructions MMX.

### Prédiction de branchement

L'algorithme de prédiction de branchement est le même que celui du M1 (prédiction à deux bits identique à celle du Pentium d'*Intel*), mais la taille du BTB a été doublée (512 entrées).

## Unité flottante

L'unité flottante supporte les calculs flottants comme sur le M1 mais aussi les instructions multi-média MMX. Cependant, comme prévu par *Intel*, le mélange des deux types d'instructions n'est pas possible et les registres sont communs.

Les instructions de multiplication-addition sont totalement pipelinées et ont une latence de deux cycles. Ainsi, les opérations arithmétiques et logiques simples sont effectuées en neuf cycles au total et celles de multiplication ou multiplication-addition en dix cycles.

## Hiérarchie mémoire

Un des changements les plus importants concerne l'augmentation de la taille du cache primaire unifié : celle-ci est multipliée par quatre et atteint 64 Koctets. Couplé à celui-ci un petit cache d'instructions associatif de 256 octets permet de libérer le cache unifié pour les données. Ce dernier est associatif par ensembles à quatre voies et supporte deux accès par cycle. Chacune de ses lignes peut être verrouillée pour le forcer à conserver des données critiques.

Afin d'assurer une traduction d'adresses rapide, l'efficacité du TLB est essentielle. Ainsi, les concepteurs du M2 ont opté pour deux niveaux de TLB. Le TLB primaire est à correspondance directe et contient 16 entrées. Quand celui-ci est en défaut, le TLB secondaire peut être en mesure de le remplacer avec un cycle de pénalité. Il est associatif par ensembles à six voies et contient 384 entrées. Les deux TLB sont double-port pour supporter les traductions d'adresses de données et d'instructions.

## Bus système

L'interface bus du M2 est compatible avec celle du Pentium et la fréquence CPU peut être 2, 2,5, 3 ou 3,5 fois la fréquence du bus.

## Technologie

Le M2 sera construit dans un premier temps par *IBM*, selon un procédé CMOS à 0,33  $\mu\text{m}$ . *SGS Thomson* devrait aussi participer à la fabrication dans un deuxième temps. Les 6 millions de transistors seront alimentés en 2,5 V (3,3 V pour les entrées/sorties) et la taille du composant devrait être inférieure à 200  $\text{mm}^2$ . Les fréquences prévues sont 180, 200 et 225 MHz.

### 5.3.3 Le K6 d'AMD

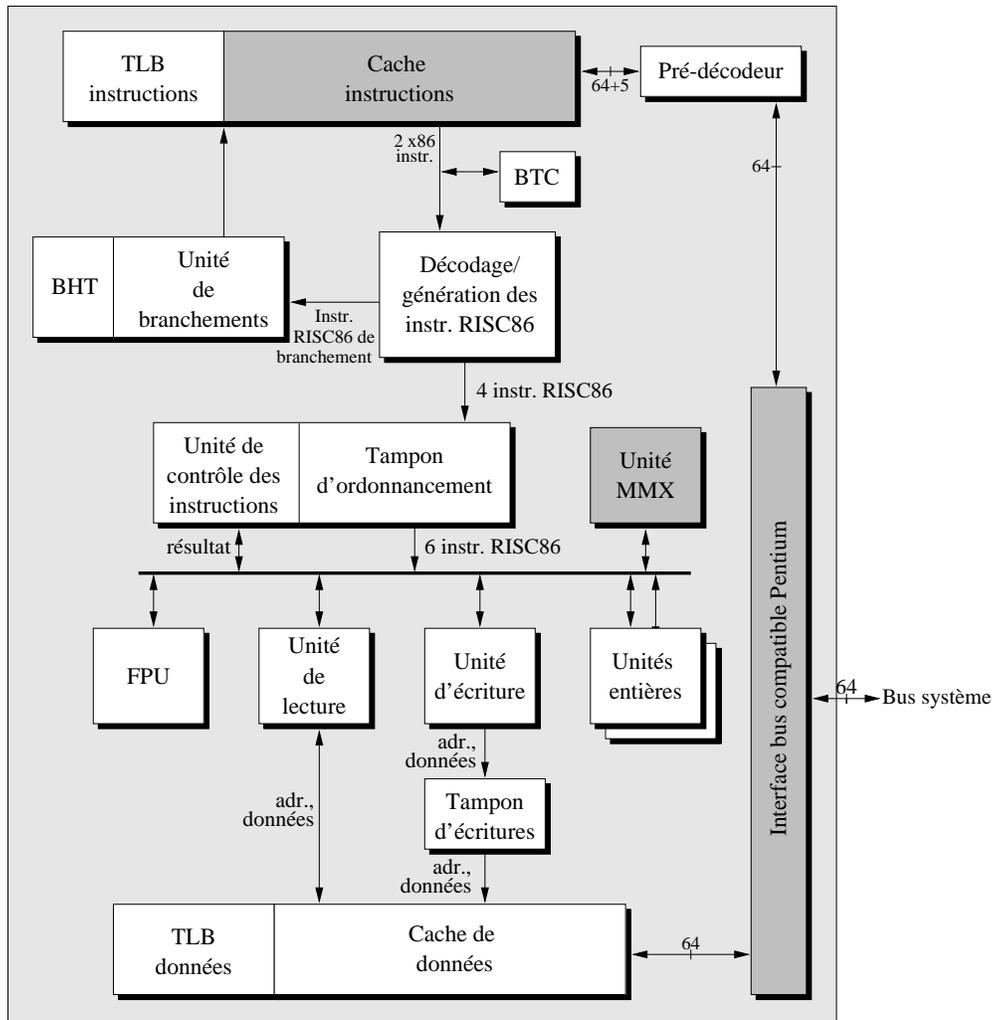
Le K6 est le processeur haut de gamme de chez *AMD* qui se veut concurrent du P55C et du futur Pentium-II, et de ce fait du M1. Présenté en octobre 1996, ce microprocesseur est basé sur l'architecture du Nx686 dévoilée un an auparavant par *NexGen* (société acquise par *AMD* en 1996). Il est disponible depuis mars 1997.

La figure 5.10 montre en grisé les modifications apportées au Nx686 pour former le K6. Il possède une architecture découplée, de nombreuses unités d'exécution et la traduction, comme sur les K5, Nx586 et PentiumPro, des instructions x86 en instructions de type RISC. *AMD* appelle ces instructions *RISC86*.

## Séquencement et exécution des instructions

### Décodage et émission des instructions

Tout comme le K5, le K6 pré-décodage les instructions avant leur chargement dans le cache d'instructions, en ajoutant cinq bits par octet. Ces bits indiquent le nombre d'octets restant avant la prochaine



**BHT** : Branch History Table.

**BTC** : Branch Target Cache.

FIG. 5.10 – Synoptique du K6 d'AMD.

instruction x86 et permettent au décodeur de mieux séparer les instructions x86 (qui ont une taille variable de 1 à 15 octets).

À chaque cycle, le K6 charge au plus 16 octets d'instructions (ou deux instructions x86) ainsi que les bits de pré-décodage associés, depuis le cache d'instructions ou depuis le *Branch Target Cache (BTC)* qui contient des instructions issues de branchements pris.

Ensuite, les instructions x86 sont traduites en instructions RISC86 dans un des trois ensembles de décodeurs. Le premier ensemble est constitué de deux décodeurs parallèles (décodeurs *courts*) qui peuvent traduire les instructions *simples* (elles constituent en général la plupart du code x86) en zéro, une ou deux instructions RISC86. Le deuxième ensemble de décodeur comporte un seul décodeur dit *long* car il traite les instructions x86 de plus de 7 octets et de moins de 11 octets. Il peut générer jusqu'à quatre instructions RISC86 par cycle. Les autres instructions x86, dites complexes, sont traduites par un décodeur dit *vectoriel* assisté d'une ROM interne où sont stockées des séquences d'instructions RISC86.

Ainsi, à chaque cycle un des trois ensembles de décodeurs est utilisé, et exactement quatre instructions RISC86 sont générées. Dans le cas où l'(ou les deux) instruction(s) x86 traduite(s) par cycle génère(nt) moins de quatre instructions RISC86 « utiles », des instructions RISC86 vides (NOP<sup>17</sup>) sont insérées<sup>18</sup>.

En moyenne, le K6 décode 1,9 instructions x86 par cycle sur du code 32 bits, et 1,8 instructions par cycle sur du code 16 bits.

Une fois les quatre instructions RISC86 produites, elles sont stockées dans un *tampon d'ordonnement (scheduler buffer)* qui centralise l'émission des instructions vers les unités d'exécution. C'est à ce niveau que se gèrent l'exécution dans le désordre et le renommage de registres (48 registres physiques). Le tampon d'ordonnement supporte jusqu'à 24 instructions RISC86, mais pas plus de 12 instructions x86 ne peuvent être actives (soit une moyenne de 15-18 instructions RISC86).

Jusqu'à six instructions peuvent être lancées en parallèle et dans le désordre, à chaque cycle, grâce à six unités d'exécution. Une vision ordonnée du système est assurée par le retrait des instructions de ce tampon dans l'ordre à raison de quatre maximum par cycle.

## Pipelines

Les pipelines du K6 sont relativement courts (FIG. 5.11) grâce en particulier au pré-décodage qui accélère le décodage.

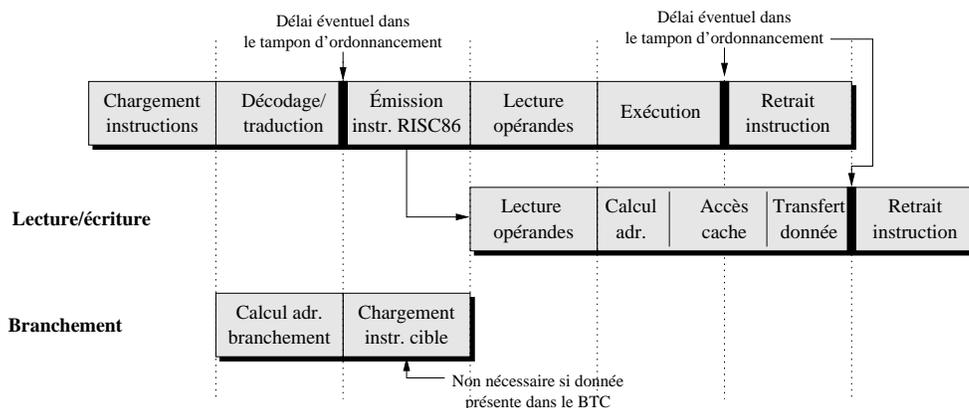


FIG. 5.11 – Les pipelines du K6.

17. *NO Operation*.

18. Ceci explique la possibilité, donnée aux décodeurs courts, de générer « zéro » instruction RISC86.

## Prédiction de branchement

L'algorithme de prédiction dynamique de branchement utilisé par le K6 est à deux niveaux, conforme à celui décrit par Yeh et Patt [23] avec indexage de type *GAs*<sup>19</sup>. Une table d'historiques de branchement de 8 192 entrées est indexée par neuf bits d'un registre d'historique global et quatre bits du PC.

Si le branchement est prédit pris, un cycle supplémentaire est nécessaire, en temps normal, pour charger les instructions cibles. Sur le K6, un cache d'instructions cibles, le *Branch Target Cache (BTC)* de 16 entrées de 16 octets chacune fournit l'instruction cible plus tôt. Ce cache est indexé par l'adresse de branchement et peut ainsi être accédé pendant la phase de décodage.

Ce dispositif de prédiction de branchement est complété par une pile d'adresses de retours de procédures de 16 entrées.

## Unités d'exécution

Les unités d'exécution sont au nombre de sept sur le K6 : une unité de lecture mémoire, une unité d'écriture mémoire, deux unités entières, une unité flottante, une unité MMX<sup>20</sup> et une unité de branchements. Chacune de ces unités n'exécute que du code RISC86 et peut travailler indépendamment des autres.

Le tableau 5.2 donne les fonctions réalisées par chaque unité d'exécution ainsi que la latence et le débit des opérations.

Unité d'exécution	Fonction	Latence	Débit
Écriture	LEA/PUSH, adresse *	2	1
	Écriture mémoire *	2	1
Lecture	Lecture mémoire *	2	1
Calcul entier X	ALU	1	1
	Multiplication	2-3	2-3
	Décalage	1	1
MMX	ALU	1	1
	Décalage, emballage, dépaquetage	1	1
	Multiplication	1	1
Calcul entier Y	ALU (opérandes 16 et 32 bits)	1	1
Branchements	Résolution des conditions de branchement	1	1
FPU	Addition, soustraction, multiplication	2	2

\*. Pipeliné.

TAB. 5.2 – Latences et débits d'exécution des unités du K6 en nombre de cycles.

On notera que la latence des opération flottantes n'est pas très élevée, mais que la FPU n'est pas pipelinée. D'autre part, toutes les opération MMX ont une latence d'un cycle sauf la multiplication-addition pour laquelle il faut deux cycles.

19. Utilisation d'un registre Global et adressage par ensembles (sets).

20. À l'origine, sur le Nx686, c'est une unité *multimédia* qui exécute du code multimédia propre à *NexGen*.

## Hiérarchie mémoire

Deux caches primaires de données et d'instructions séparés sont intégrés au K6. Le Nx686 ne prévoyait que 16 Koctets pour les instructions mais le K6 comporte 32 Koctets. Le cache de données a aussi une taille de 32 Koctets. Ces deux caches sont associatifs par ensembles à deux voies et supportent un protocole de cohérence de type MESI. Les écritures se font en différé. L'algorithme de remplacement sur le cache d'instruction est de type LRU, alors que sur le cache de données, l'algorithme utilisé est de type *LRA*<sup>21</sup>.

Au cache d'instructions est associé un TLB de 64 entrées. Un TLB de 128 entrées est quant à lui associé au cache de données.

Le Nx686 avait une interface pour un cache secondaire ; sur le K6, ce n'est plus le cas.

## Bus système

Le bus système particulier du Nx686 a été remplacé par un bus compatible Pentium.

## Technologie

Les 8,8 millions de transistors du K6 occupent 157 mm<sup>2</sup> et sont fabriqués en technologie CMOS à 0,35  $\mu\text{m}$ . La fréquence interne va jusqu'à 233 MHz.

## 5.4 Récapitulatif sur les microprocesseurs d'architecture x86

Le tableau 5.3 récapitule les principales caractéristiques des processeurs x86.

## 5.5 Les microprocesseurs x86 à venir

### 5.5.1 Le Pentium-II (Klamath) d'Intel

Officiellement nommé Pentium-II depuis mars 1997, le microprocesseur certainement plus connu sous le nom de Klamath est une version améliorée du PentiumPro qui devrait sortir après la mi-1997 à la fréquence de 266 MHz. Cette version vise en premier lieu le grand public et devrait se vendre, à terme, meilleur marché que le PentiumPro d'origine. L'augmentation de la latence d'accès au cache secondaire, comme nous allons le voir plus loin, écarte ce microprocesseur du marché des stations de travail et des serveurs qui requièrent de larges espaces de travail et donc de nombreux accès à ce cache. Par ailleurs, la configuration de quatre microprocesseurs sans ajout de logique restera une particularité du PentiumPro d'origine puisqu'elle n'est pas reprise sur le Pentium-II.

Le cœur du microprocesseur reste inchangé, mis à part quelques nouveautés, dont l'exécution des instructions MMX. Pour cela, les deux unités entières du PentiumPro se sont vues adjoindre une unité d'exécution d'instructions MMX chacune. L'unité fonctionnelle principale qui traite les opérations entières et flottantes (voir FIG. 5.7, p. 103) gagne une ALU et un multiplieur MMX. La seconde unité entière inclut une ALU MMX ainsi qu'un décaleur. Ainsi, presque toutes les combinaisons de deux instructions MMX peuvent être exécutées en même temps, excepté deux multiplications.

Une autre modification du cœur du PentiumPro concerne les écritures des registres de segments. Sur le PentiumPro, celles-ci ne peuvent pas avoir lieu de manière spéculative, elles doivent attendre que le pipeline soit vidé des instructions en cours d'exécution. Le Pentium-II renomme les registres de segments lors d'écritures, permettant à cette dernière d'avoir lieu à n'importe quel moment. Les instructions référant des valeurs de registres de segments différentes peuvent ainsi cohabiter. Cette

---

21. *Least Recently Allocated.*

	Intel				AMD		Cyrix		
	Pentium (P54CS)	Pentium/- MMX (P55C)	Pentium- Pro	Pentium- II	K5	K6	6x86 (M1)	M2	MediaGX
Interface	Pentium	Pentium	P6	P6 <sup>†</sup>	Pentium	Pentium	Pentium	Pentium	propre
MMX?	non	oui	non	oui	non	oui	non	oui	non
Cache primaire (I/D en Koc- tets)	8/8	16/16	8/8	16/16	16/8	32/32	16 <sup>U</sup>	64 <sup>U</sup>	16 <sup>U</sup>
Cache secondaire (en Koc-tets)	non	non	256-512	256-512	non	non	non	non	non
Vitesse bus cache secondaire*	—	—	1	1/2	—	—	—	—	—
Pipeline (nombre d'étages)	5	6	12	12	6	6	7	7	6
Exécution dans le désordre	non	non	oui	oui	oui	oui	limitée	limitée	non
Décodage en parallèle (nombre d'instruc- tions)	2	2	3 <sup>‡</sup>	3 <sup>‡</sup>	4	2	2	2	1
BTB (nombre d'entrées)	256	256	512	512	1024*	8192*	256	512	128
TLB (I/D, nombre d'en- trées)	32/64	32/64	32/64	32/64	128 <sup>U</sup>	128/64	128-8**	16-384**	32
Alimentation (en V)	3,3	2,8	3,3	2,8	3,5	2,9	3,3	2,5	3,3
Puissance typique dissi- pée (en W) <sup>◊</sup>	3,9-6,5	5,5-7,3	23-32,6	???	7,9-10,6	???	14,8-21,1	???	???
Processus de fabrication	BiCMOS 0,35 μm	CMOS 0,28 μm	BiCMOS 0,35 μm	CMOS 0,28 μm	CMOS 0,35 μm	CMOS 0,35 μm	CMOS 0,44 μm	CMOS 0,35 μm	CMOS 0,5 μm
Nombre de transistors (en millions)	3,3	4,5	5,5	7,5	4,3	8,8	3,0	6,0	1,9
Taille (en mm <sup>2</sup> )	90	140	196	203	177	157	169	197	120
Fréquences (en MHz)	100-200	166-200	150-200	233-266	75-100	180-233	100-150	180-225	120-133

\*. Fraction de la vitesse du CPU.

<sup>†</sup>. + bus cache secondaire.

<sup>U</sup>. Cache unifié.

<sup>‡</sup>. Deux simples plus une complexe.

\*. BHT: *Branch History Table*.

\*\* . Cache primaire - cache secondaire.

◊ . En correspondance avec les fréquences.

TAB. 5.3 – Les microprocesseurs d'architecture x86 actuels (1996-97).

particularité se répercute essentiellement sur l'exécution de code 16 bits, sur lequel le PentiumPro n'atteint pas de très hautes performances, et dans lequel les écritures de registres de segments sont monnaie courante (contrairement au code 32 bits).

La hiérarchie mémoire a été revue. Le cache secondaire du PentiumPro, intégré au boîtier du microprocesseur ne l'est plus sur le Pentium-II. En effet, cette particularité du PentiumPro limite son augmentation de fréquence car la mémoire utilisée pour ce cache est synchrone et cadencée à la même fréquence que le CPU. Sa taille se trouve aussi limitée à 512 Koctets. Pour réduire les coûts de fabrication et augmenter la fréquence interne, les concepteurs du Pentium-II ont opté pour un cache secondaire réellement externe, cadencé à la moitié (voire le tiers pour les systèmes à relatif bas prix) de la fréquence interne. D'autre part, sa taille n'est plus limitée mais elle reste plafonnée à 512 Koctets dans un premier temps. Le cache secondaire est interfacé au microprocesseur par un bus dédié de 64 bits de large pour les données (le bus système du Pentium-II reste le même que celui de son prédécesseur).

Pour compenser un cache secondaire plus lent, les caches primaires d'instructions et de données ont été doublés. Ils ont sur le Pentium-II une taille de 16 Koctets chacun.

La technologie de fabrication utilisée pour le Pentium-II sera la même que celle utilisé pour le P55C : CMOS à 0,28  $\mu\text{m}$ . Les 7,5 millions de transistors occuperont une surface de 203  $\text{mm}^2$  et opéreront à 233 et 266 MHz.

### 5.5.2 Autres

*Intel* prévoit un passage à la technologie de fabrication CMOS à 0,25  $\mu\text{m}$  à la fin de l'année 1997, et avec celle-ci une amélioration des processeurs existants. Le P55C (p. 95) amélioré technologiquement, est prévu pour fin 1997, il porte le nom de code de **Tillamook** et devrait être cadencé à 200 MHz, voire 233 MHz dès sa sortie. Le **Deschute**, version mise à jour du Pentium-II devrait apparaître aussi fin 1997 – début 1998.

Le **Merced** (P7) sera issu d'une architecture 64 bits toute nouvelle (IA-64) conçue par *Intel* et *HP*. Il verra le jour en 1998. La transition vers cette architecture 64 bits sera faite en douceur puisque le Merced assurera une compatibilité binaire pour les applications x86. Les applications PA-RISC devraient être, quant à elles, émulées.



## Chapitre 6

# L'architecture PA-RISC

Introduite en 1986 sous sa première forme par *Hewlett-Packard (HP)*, l'architecture PA-RISC<sup>1</sup> comporte un jeu d'instruction RISC. Elle a été réalisée pour remplacer les microprocesseurs CISC dans les systèmes commerciaux HP3000, les stations de travail et serveurs HP9000, et les contrôleurs temps-réel HP1000.

Les microprocesseurs d'architecture PA-RISC ne sont pas vendus en tant que tels, mais intégrés aux systèmes *Hewlett-Packard*. Ils ont contribué de par leurs hautes performances au succès commercial des stations de travail et des serveurs proposés par *Hewlett-Packard* (les systèmes basés sur l'architecture PA-RISC sont parmi les systèmes RISC les plus vendus).

D'autres compagnies, *Oki*, *Hitachi* et *Winbond*, se sont lancées dans la fabrication de microprocesseurs d'architecture PA-RISC. Ces microprocesseurs étaient avant tout destinés au marché embarqué, mais ils n'ont pas trouvé dans la presse d'écho retentissant, ce qui laisse douter de leur réussite tant technique que commerciale. Pour cela, nous n'en parlerons pas plus dans la suite de ce rapport, d'autant plus que ces tentatives datent de 1993 et 1994.

Ces trois fabricants font partie d'une association, la *PRO (Precision RISC Organization)*, regroupant des compagnies ayant pour but commun la fabrication de produits basés sur l'architecture PA-RISC. Créée en 1992, la PRO a pour membres (hormis ceux sus-cités et *HP*) : *Allegro Consulting*, *CS (France)*, *Convex Computer*, *CPI*, *Information Builders*, *Innovus*, *Interex*, *Kelly Computers*, *Mitsubishi Electric*, *NEC*, *Samsung Electronics*, *Stratus Computer*, *Wind River Systems* et *Yokogawa Electric*.

Dans les sections suivantes, nous présentons l'architecture PA-RISC à travers les microprocesseurs qui en ont constitué l'implantation, après avoir donné les caractéristiques de ses spécifications, selon les différentes versions. Les microprocesseurs étudiés par la suite ne forment pas une liste exhaustive, mais donnent un aperçu de l'évolution des procédés et techniques mis en œuvre par *HP* au cours de la dernière décennie. Aussi, les premières implémentations de l'architecture PA-RISC ne seront pas développées par manque de documentation et parce qu'elles présentent un intérêt moindre.

### 6.1 Description de l'architecture

Depuis son introduction au milieu des années 80, l'architecture PA-RISC est restée relativement stable. Quelques changements ont eu lieu pour amener de plus hautes performances, en calcul flottant notamment.

La première version, PA-RISC 1.0, considérait le calcul flottant à travers un coprocesseur optionnel. En 1989, poussée par les besoins en performances des stations de travail techniques HP9000, la version PA-RISC 1.1 a été introduite, proposant un support complet pour le calcul flottant (nombreuses

---

1. *Precision Architecture, Reduced Instruction Set Computer*.

nouvelles instructions et 16 nouveaux registres flottants). Cette version a subi dans trois éditions successives, trois légères modifications qui concernent le système d'exploitation. Dans ce qui suit nous détaillons dans un premier temps la version PA-RISC 1.1 dans sa troisième édition [24] (le lecteur trouvera les deux autres éditions dans [25] et [26]).

La version PA-RISC 2.0 [27] que nous détaillons par la suite comporte pour la première fois des modifications visibles par l'utilisateur : le support pour les entiers sur 64 bits, les adresses physiques codées aussi sur 64 bits et de nouvelles instructions comme celles formant l'extension multimédia. Le premier microprocesseur à implémenter cette version d'architecture est le PA-8000, annoncé depuis la fin 1994.

### 6.1.1 Version 1.1

L'architecture PA-RISC 1.1 est une architecture RISC classique, à 32 bits et de type *load/store*. Toutes les instructions sont « cablées » (pas de microcode) et ont une taille fixe de 32 bits. Un petit nombre de modes d'adressage est supporté : basé (deux sortes de déplacements : court ou long) et indexé<sup>2</sup>.

#### Types de données

Les types de données supportés par l'architecture PA-RISC sont ceux du tableau 6.1.

	Octet (8 bits)	Demi-mot (16 bits)	Mot (32 bits)	Double mot (64 bits)	Quadruple mot (128 bits)
Caractères ASCII (codes 0 à 127)	×				
Caractères étendus « Roman-8 » (codes 128 à 255)	×				
Entiers signés		×	×		
Entiers non-signés		×	×	×	
Flottants IEEE-754			×	×	×

TAB. 6.1 – Les types de données de l'architecture PA-RISC.

À ces types de données viennent s'ajouter deux autres types : les décimaux *empaquetés* (*Binary Coded Decimal – BCD*) et non-empaquetés. Les décimaux empaquetés sont composés de 7, 15, 23 ou 31 chiffres décimaux (valeurs 0 à 9 codées sur quatre bits) suivis par quatre bits de signe<sup>3</sup>. La figure 6.1 représente un décimal empaqueté.



FIG. 6.1 – Décimal empaqueté de l'architecture PA-RISC.

Les décimaux non-empaquetés sont constitués de chiffres décimaux codés sur un octet chacun (ils n'occupent que quatre bits, les quatre autres étant inutilisés).

Le support de ces types de données est exceptionnel pour une architecture RISC, mais les raisons en sont certainement historiques : les systèmes *HP* étaient fortement introduits sur le marché des ordinateurs commerciaux qui utilisaient beaucoup le *Cobol*, langage gourmand en *BCD*.

2. Des options permettent cependant un mode qui modifie le registre de base (pré-décrémentation ou post-incrémentation).

3. Le signe standard pour un nombre positif est 0xC, mais toute autre valeur, exceptée 0xD, donne un signe positif.

## Registres

Les registres des microprocesseurs PA-RISC sont de plusieurs types, outre ceux contenant le PC :

- Trente-deux **registres généraux** (GR0 à GR31) de 32 bits permettent la plupart des calculs. GR0 est câblé à zéro ; GR1 et GR31 sont généraux, mais ont chacun une instruction donnée pour laquelle ils ont une signification particulière.
- Sept **registres cachés** (*shadow registers*, SHR0 à SHR6) de 32 bits prennent les valeurs de sept registres généraux lors de changements de contexte pour le traitement d'interruptions.
- Huit **registres d'espace d'adressage** (*Space Registers*, SR0 à SR7) de 0, 16, 24 ou 32 bits (selon le *niveau* d'architecture utilisé, voir plus loin) identifient l'espace d'adressage dans le cas où la mémoire est virtuelle. Par convention, tous les registres d'espace d'adressage n'ont pas la même utilité ni le même mode d'accès. Seuls les registres SR1 à SR3 peuvent être utilisé de manière générale et sont accédés directement.
- Le **mot d'état du processeur** (*Processor Status Word*, *PSW*) contient 32 bits divisés en de nombreux champs indiquant des initialisations système ou des masques (masque d'interruption par exemple).
- Vingt-cinq **registres de contrôle** de 32 bits (CR0 et CR8 à CR31) contiennent des paramètres de contrôle divers (masque d'interruptions externes, file d'adresses d'instructions de traitement d'interruptions, registres temporaires permettant au système de stocker des registres généraux lors de traitement d'interruptions, ...).
- Trente deux **registres flottants** de 64 bits sont utilisés pour les calculs flottants. Ils peuvent aussi prendre la forme de 64 registres de 32 bits ou 16 registres de 128 bits<sup>4</sup>. Seuls les registres 4 à 31 ont une utilité générale pour les opérandes. Les autres sont réservés à des états d'opérations et aux exceptions.

## Adressage de la mémoire et des entrées/sorties

### Adresse absolue

Les données de la mémoire principale et des entrées/sorties sont accédées à l'aide d'une *adresse absolue* (terminologie utilisée par *HP* pour l'adresse physique) de 32 bits. Les entrées/sorties correspondent à des adresses mémoire dont la relation avec la mémoire principale est illustrée en figure 6.2. Ainsi, d'une manière générale, la gestion des entrées/sorties se fait à travers des lectures/écritures mémoire classiques.

La plus petite donnée accessible est l'octet et les adresses doivent pointer sur des données alignées (les demi-mots commencent aux adresses paires, les mots aux adresses multiples de 4, etc.). Les octets sont rangés en mémoire dans l'ordre *little endian* ou *big endian* selon la valeur d'un champ du *mot d'état du processeur* (*Processor Status Word*)<sup>5</sup>.

### Niveaux d'architecture

Quatre *niveaux* d'architecture ont été définis : 0, 1, 1,5 et 2. Les systèmes de niveau 0 supportent uniquement un adressage absolu, ils n'implémentent donc pas de mémoire virtuelle. Les systèmes d'autres niveaux utilisent un adressage virtuel et leurs différences résident dans le nombre d'*espaces d'adressage* — une adresse virtuelle est composée d'un espace d'adressage et d'un *offset* dans cet espace. Les systèmes de niveau 1 ont  $2^{16}$  espaces d'adressage, ceux de niveau 1,5,  $2^{24}$  et ceux de niveau

4. Caractéristique amenée par la version 1.1. La version d'architecture précédente ne définissait en outre que 16 registres flottants.

5. Cette particularité date de la troisième édition de l'architecture PA-RISC 1.1. Auparavant, l'ordre donné aux octets était seulement de type *big endian*.

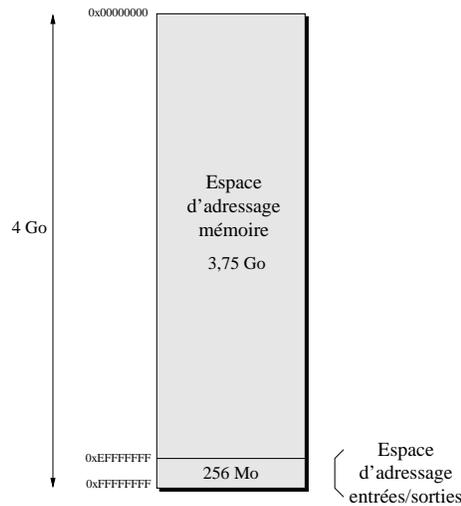


FIG. 6.2 – Adresses physiques de la mémoire et des entrées/sorties.

2,  $2^{32}$ . Ceci conditionne la taille des registres d'espaces d'adressage (*Space Registers*) en 16, 24 ou 32 bits.

### Structure de la mémoire virtuelle

Ce paragraphe concerne uniquement les systèmes de niveau strictement supérieurs à 0 : à adressage virtuel. L'architecture PA-RISC 1.1 définit une mémoire virtuelle composée d'*espaces virtuels* de  $2^{32} = 4$  Goctets chacun<sup>6</sup>.

Lors de la traduction d'adresse virtuelle en adresse physique, un espace d'adressage est sélectionné par un identificateur d'espaces d'adressage contenu dans un des registres d'espaces d'adressage (*Space Registers*) ; ceci constitue la « partie haute » de l'adresse virtuelle. La « partie basse » est constituée d'un décalage (*offset*) de 32 bits dans l'espace virtuel ; ce décalage est obtenu en faisant la somme des 32 bits du registre source de base avec les 32 bits du registre source d'index ou ceux d'un déplacement (valeur immédiate de 14 bits pour les longs ou 5 bits pour les courts, à signe étendu sur 32 bits). Un registre d'espace d'adressage est sélectionné parmi les sept présents grâce à un champ de deux bits de l'instruction (*s-field*) et/ou (selon la valeur du *s-field*) les deux bits de poids forts du décalage.

Pour une meilleure gestion de la mémoire, les espaces d'adressage sont subdivisés en *pages* d'une taille fixe de 4 Koctets<sup>7</sup> (les données sont accédées à l'intérieur des pages grâce aux 12 bits de poids faible de l'adresse virtuelle). L'émulation de pages de plus grande taille est possible : un *groupe de pages* est spécifié, il contient huit pages contiguës en mémoire et alignés sur 32 Koctets.

La figure 6.3 explicite la relation entre les espaces d'adressage, les décalages et les pages.

### Jeu d'instructions

Le jeu d'instructions PA-RISC 1.1 est relativement riche pour un jeu d'instructions RISC. Ainsi, de nombreuses instructions « de base » combinent deux opérations simples qui sont généralement séparées dans les jeux d'instructions RISC classiques. Aussi, le format des instructions ne suit pas un schéma général rigide malgré une taille fixe de 32 bits.

6. Ces espaces virtuels sont comparables à des *segments* mais sont disjoints.

7. La taille des pages est de 2 Koctets dans les systèmes d'architecture PA-RISC 1.0.

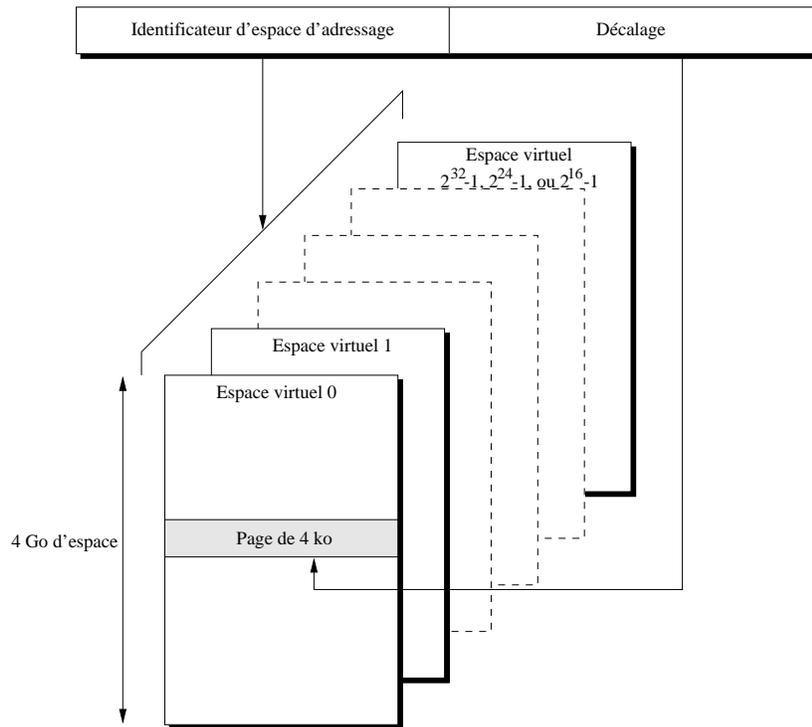


FIG. 6.3 – Adresse virtuelle de l'architecture PA-RISC 1.1.

Les différents types d'instructions définis par PA-RISC 1.1 sont les suivants :

- Instructions d'accès à la mémoire – Elles transfèrent des données entre la mémoire principale ou les entrées/sorties, et les registres généraux.
- Instructions arithmétiques et logiques – On trouve parmi elles les opérations entières classiques ainsi que des instructions de décalage-addition pour accélérer les multiplications par des constantes, des instructions d'« extraction » et de « dépôt » pour des manipulations de bits et des instructions opérant sur les formats de données décimaux. Une des améliorations de la version d'architecture 1.1 concerne le support de la multiplication entière. Celle-ci est implémentée dans l'unité flottante.
- Instructions de contrôle de flot – Elles gèrent les branchements conditionnels et les sauts inconditionnels. Tous les branchements sont « reportés » et un bit d'annulation permet de ne pas tenir compte de l'instruction suivante. Il n'y a pas de code condition ; la condition est déterminée dans l'instruction de branchement conditionnel.
- Instructions flottantes – Elles sont conformes aux spécifications de la norme IEEE-754. La version 1.1 de l'architecture complète le jeu d'instructions flottant avec une instruction de multiplication-addition et une instruction de multiplication-soustraction. Deux instructions sont nécessaires pour effectuer un branchement relatif à la comparaison de données flottantes, soit : évaluation de la condition, puis branchement conditionnel.
- Instructions de contrôle du système – Elles fournissent le support nécessaire à l'implémentation d'un système d'exploitation. Elles contrôlent en particulier les registres non-généraux et le(s) TLB.

### 6.1.2 Version 2.0

Nous ne détaillons dans cette partie que les améliorations apportées par la nouvelle version d'architecture, étant entendu que certaines caractéristiques demeurent inchangées.

Les plus grands changements apportés par l'architecture PA-RISC 2.0 concernent le support de mots de 64 bits pour les entiers et les adresses physiques, et l'extension multimédia. Ce sont ces deux points que nous développons dans les deux premières parties.

Outre ces caractéristiques, des améliorations au système mémoire et au(x) cache(s) sont introduites, ainsi que des perfectionnements au niveau de l'exécution de code flottant et à propos de la prédiction de branchement. Tout ceci est repris dans une dernière partie.

#### Extension 64 bits

Un système conforme à l'architecture PA-RISC 1.1 de niveau 2 comporte des adresses virtuelles codées sur 64 bits (32 bits d'espace d'adressage et 32 bits de décalage dans cet espace), et réalise des calculs flottants sur 64 bits. Cependant, tous les chemins de données ont 32 bits de large (de même que tous les registres non-flottants), et les adresses physiques restent codées sur 32 bits. Tout ceci est codé sur 64 bits par PA-RISC 2.0.

Ainsi, l'extension 64 bits amenée par PA-RISC 2.0 concerne deux domaines : les opérations non-flottantes et la formation des adresses physiques (accès mémoire).

##### Opérations non-flottantes

PA-RISC 2.0 fournit un jeu d'instructions dont les opérandes sont codés sur 64 bits. Cependant, une compatibilité binaire ascendante est assurée : le code 32 bits est supporté.

Pour ce faire, la solution retenue par *HP* a été de considérer tous les opérandes et tous les résultats comme des données 64 bits en les étendant. Certaines instructions ont néanmoins dû être dupliquées comme les décalages 32 bits et les comparaisons 32 bits pour les branchements conditionnels.

À noter que la multiplication entière sur 64 bits n'est pas supportée en tant que telle, elle peut être synthétisée à partir de l'ancienne multiplication 32 bits.

##### Adressage 64 bits

Dans le cas d'accès mémoire, et donc de génération d'adresse, le code 32 bits est toujours supporté, alors que les registres d'*identification d'espace d'adressage* et généraux (*offset* dans l'adresse virtuelle) ont 64 bits de large<sup>8</sup>, et les données 32 bits sont étendues à 64 bits.

Pour faire la distinction entre les deux types de code (32 bits ou 64 bits) pour la génération d'adresses, un bit de *mode d'adressage* est introduit. Il se trouve dans le registre *mot d'état du processeur*.

Dans le cas d'exécution de code 64 bits, le même modèle de génération d'adresse virtuelle est utilisé qu'auparavant quant à la sélection du registre d'identification d'espace d'adressage (utilisation des deux bits de poids fort du ou des registres sources selon la valeur du champ *s-field* de l'instruction). Cependant, la formation de l'adresse virtuelle n'est pas la même, les valeurs *espace d'adressage* et *décalage* se chevauchent selon le schéma 6.4.

Les parties se chevauchant sont les termes d'un OU logique dont le résultat est concaténé aux autres parties. Les deux bits de poids fort du *décalage* sont inutilisés ici.

D'un point de vue strictement matériel, n'importe quel *décalage* peut être associé à n'importe quel *espace d'adressage*, mais des conventions logicielles limitent ces combinaisons pour créer des régions disjointes de l'espace virtuel, et ainsi assurer une protection des données.

La taille des pages spécifié par l'architecture PA-RISC 2.0 varie de 4 Koctets à 64 Moctets par multiples de 4.

---

8. Les registres d'*identification d'espace d'adressage* peuvent être moins larges : sur le PA8000, ils ne font que 32 bits.

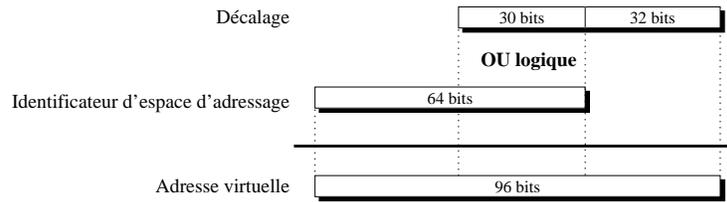


FIG. 6.4 – Modèle d'adressage virtuel de PA-RISC 2.0.

## Extension multimédia

*MAX*<sup>9</sup> est une extension au jeu d'instructions PA-RISC vouée à l'accélération des opérations sur les « données multimédias ». Ces données sont utilisées dans les traitements d'images, de sons, de graphiques 2D et 3D, etc., et ont une taille généralement plus petite (8, 12, 16 bits, voire 24 bits) que les données servant aux opérations classiques (64 bits). Il est donc intéressant de grouper plusieurs données multimédias dans un mot de 64 bits et de faire un traitement parallèle sur celles-ci (approche dite SIMD).

Les instructions dont il est question ici font partie de la deuxième version d'instructions MAX, MAX-2. MAX-1 a été implémentée sur le microprocesseur PA-7100LC, d'architecture PA-RISC 1.1 et constitue un sous-ensemble de MAX-2 opérant sur des données 32 bits (une compatibilité binaire ascendante est assurée pour le code MAX-1). MAX-2 (que nous appellerons MAX dans la suite) n'est pas une extension optionnelle, mais fait partie intégrante de la norme PA-RISC 2.0.

Les instructions MAX sont simples, de bas niveau et ainsi simples d'implémentation et d'exécution rapide. Elles opèrent en parallèle sur des données de 16 bits (compromis entre des données de 8 bits et des données de 32 bits<sup>10</sup>). Le tableau 6.2 en donne la liste.

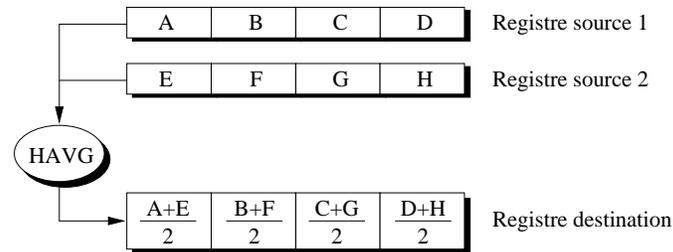


FIG. 6.5 – Moyenne parallèle sur les demi-mots dans l'extension MAX-2.

## Améliorations diverses

PA-RISC 2.0 amène diverses améliorations à la version d'architecture précédente. Elles concernent trois points : la hiérarchie mémoire, la prédiction statique de branchement et le calcul flottant.

### Hiérarchie mémoire

Deux nouvelles caractéristiques améliorent la synergie entre le compilateur et le processeur, en donnant plus de possibilités au compilateur pour qu'il puisse utiliser au mieux toutes les informations dont il dispose en ce qui concerne les accès mémoire :

9. *Multimedia Acceleration eXtension*.

10. On peut aussi se rappeler la compatibilité binaire ascendante qui permet de traiter des mots de 32 bits, mais de manière séquentielle.

Instruction	Option(s)	Description
Addition parallèle	saturation signée, saturation non signée ou sans dépassement de capacité.	Additionne chaque demi-mot d'un registre source avec le demi-mot correspondant du second registre source.
Soustraction parallèle	saturation signée, saturation non signée ou sans dépassement de capacité.	Soustrait chaque demi-mot d'un registre source au le demi-mot correspondant du second registre source.
Décalage parallèle à gauche et addition	saturation signée	Pour multiplier chaque demi-mot par une constante entière.
Décalage parallèle à droite et addition	saturation signée	Pour multiplier chaque demi-mot par une constante fractionnelle ( $\times \frac{1}{cte}$ ).
Moyenne parallèle		Moyenne arithmétique de chaque demi-mot des registres sources (FIG. 6.5).
Décalage parallèle à droite	signé ou non signé	Alignement de données, division entière signée ou non signée.
Décalage parallèle à gauche		Alignement de données.
Mixtion	bits de poids fort ou bits de poids faible	Intercala les demi-mots d'un registre source avec les demi-mots correspondants de l'autre registre source ; le résultat comporte 64 bits qui correspondent aux bits poids fort ou de poids faible des registres source, selon l'option.
Permutation		Réarrange les demi-mots à l'intérieur d'un registre source ; plusieurs schémas sont possibles.

TAB. 6.2 – Les instructions de l'extension MAX-2.

Pour ne pas polluer le ou les caches avec des données utilisées une seule fois, PA-RISC définit une **indication de localité** dans les instructions de lecture ou d'écriture mémoire. Cette indication permet de ne pas charger la donnée concernée dans le cache et de seulement la mettre dans un tampon. Ceci évite certains défauts de cache « sur conflits » qui surviennent lors de chargements dans le cache de lignes qui prennent la place d'autres lignes dont les données peuvent encore être utiles au CPU.

PA-RISC 2.0 définit aussi une **instruction de pré-chargement** de données dans le cache qui permet d'anticiper le chargement de la donnée. Cette instruction indique que la donnée devrait être chargée (pour une utilisation future), en indiquant l'état de la ligne de cache associée (*exclusif* ou *partagé*). Elle ignore en outre les exceptions de TLB : elle se réduit à une opération nulle (NOP) en cas de défaut de TLB.

Les accès mémoire sont définis par PA-RISC 2.0 comme étant *faiblement ordonnés* dans leur utilisation par défaut. **L'ordre des accès mémoire** peut être renforcé explicitement par les instructions de lecture ou d'écriture concernées. La compatibilité binaire ascendante est assurée par un bit du registre d'état du processeur (*PSW*).

### Prédiction statique de branchement

La prédiction statique définie par PA-RISC 1.x est la suivante : les sauts « en arrière » sont prédits pris et les sauts « en avant » sont prédits non pris. Ceci n'empêche bien sûr pas l'utilisation d'une prédiction dynamique de branchements.

L'idéal pour une prédiction statique de branchement serait que le compilateur puisse sélectionner le type de prédiction à utiliser selon le type de branchement. Ceci n'est pas possible avec les anciennes versions d'architecture, mais PA-RISC 2.0 fixe une convention logicielle qui utilise l'ordre des numéros des registres sources dans le cas d'un branchement conditionnel. En effet,

si  $R_5 \leq R_7$ , alors saut vers *cible*

est équivalent à :

si  $R_7 \geq R_5$ , alors saut vers *cible*

mais permet des indications différentes pour la prédiction de branchement. Ainsi PA-RISC 2.0 considère que si les numéros des registres sources de l'instruction de branchement conditionnel sont dans un ordre croissant, alors le branchement est prédit dans une direction. Sinon, il sera prédit dans l'autre direction. Dans le cas où l'un des opérandes est un immédiat, une instruction supplémentaire peut être générée pour mettre cette valeur dans un registre et ensuite utiliser le registre comme opérande source.

D'autre part, PA-RISC 2.0 permet l'utilisation de piles d'adresses de retours de procédures grâce à des indications (*hint*) pour les instructions réalisant de tels sauts (indications *push* et *pop*).

### Calcul flottant

Une nouvelle instruction vise l'amélioration des performances flottantes : c'est une multiplication-addition (*multiply-accumulate*) qui multiplie les deux premiers registres sources et les ajoute à un troisième avant de stocker le résultat dans un registre destination. Pour plus de précision, un seul arrondi est fait, ce qui met cette instruction hors de la norme IEEE-754.

Une autre caractéristique de l'unité flottante concerne le nombre de bits de condition<sup>11</sup> : ils sont désormais 8 (contre un pour la version précédente d'architecture).

---

11. Les bits de condition se trouvent dans le registre d'état (*Status Register*, registre flottant numéro 0) et sont mis à jour lors d'exécution d'instructions de comparaison. Ils sont ensuite utilisés par des instructions de test.

## 6.2 Les microprocesseurs d'architecture PA-RISC

Les premiers microprocesseurs d'architecture PA-RISC sont apparus en 1985 et étaient fabriqués en technologie TTL<sup>12</sup>. Depuis, *HP* a produit plusieurs implémentations PA-RISC en NMOS<sup>13</sup>, puis en CMOS. La volonté de *HP* d'introduire des stations de travail plus « populaires » (entre autres meilleur marché que les précédentes) l'a conduit à améliorer l'architecture (version 1.1) pour proposer de nouveaux microprocesseurs aux performances accrues et aux coûts moindres. Le premier exemple de ces microprocesseurs ne porte pas de nom, mais s'intègre dans les stations de travail de nom de code « **Snake** », appelées ensuite *HP Apollo 9000 Series 700*.

Dans ce qui suit, nous détaillons dans une première partie les microprocesseurs d'architecture PA-RISC 1.1, en commençant par celui que nous appellerons « Snake » (les autres possèdent un nom). Les microprocesseurs précédents dans la gamme PA-RISC ne seront pas présentés car, en ce qui les concerne, la documentation fait défaut (ils n'ont même pas de nom) et ils ne sont plus au goût du jour (antérieurs à 1991 et d'architecture PA-RISC 1.0).

Une seconde partie expose l'unique microprocesseur conforme à l'architecture PA-RISC 2.0 disponible actuellement, le PA-8000. D'autres microprocesseurs y feront suite dans un futur plus ou moins proche et sont présentés dans un sous-chapitre différent (section 6.3, p. 138).

### 6.2.1 Les microprocesseurs d'architecture PA-RISC 1.1

#### Le « Snake »

**Snake** est le nom de code des premières stations de travail *HP Apollo 9000 Series 700* sorties en 1991. Nous garderons ici ce nom pour désigner le microprocesseur qui en constitue le cœur.

Le snake est un processeur conforme à l'architecture PA-RISC 1.1, de niveau 1 : les registres d'espace d'adressage une taille de 16 bits — il y a donc  $2^{16} = 65\,536$  espaces d'adressage de 4 Goctets chacun. Il comporte deux composants séparés : un CPU fabriqué par *HP* (comportant une unité entière) et un FPC (*Floating Point Coprocessor*) fabriqué en collaboration par *HP* et *Texas Instruments* (*TI*). Les deux composants fonctionnent de manière synchronisée : le microprocesseur est scalaire. Les caches d'instructions et de données séparés sont externes et accédés par l'une ou l'autre des unités (FIG. 6.6).

#### Pipeline

Le pipeline du Snake comporte cinq étages, comme le montre la figure 6.7, divisés en deux phases d'horloge. Les opérations réalisées ne sont ainsi pas forcément calées sur les bornes des étages et peuvent les dépasser (voir FIG. 6.8, p. 127 à titre d'exemple).

Les blocages du pipeline dus aux branchements conditionnels sont réduits grâce à la prédiction de branchement statique définie par PA-RISC 1.1 : les branchements « en arrière » sont prédits pris, et ceux « en avant » sont prédits non pris. Un cycle est perdu dans le cas d'une mauvaise prédiction.

#### Unité entière

Le CPU comporte une unique unité entière qui gère les calculs arithmétiques et logiques, mais aussi les accès mémoire entiers et flottants. Les branchements sont aussi résolus dans cette unité<sup>14</sup>.

#### Unité flottante

L'unité flottante est synchrone avec le CPU et exécute les instructions qui la concernent, directement à partir du cache d'instructions externe : aucune instruction n'est passée entre le CPU et l'unité flottante. La latence des opérations flottantes est de trois cycles et leur débit est de deux cycles.

12. *Transistor-Transistor Logic*.

13. *Negative channel Metal-Oxide Semiconductor*.

14. On rappelle que les instructions de branchement conditionnel de l'architecture PA-RISC comportent en général une comparaison qui se résout par une addition ou une soustraction entière, puis un test sur le résultat.

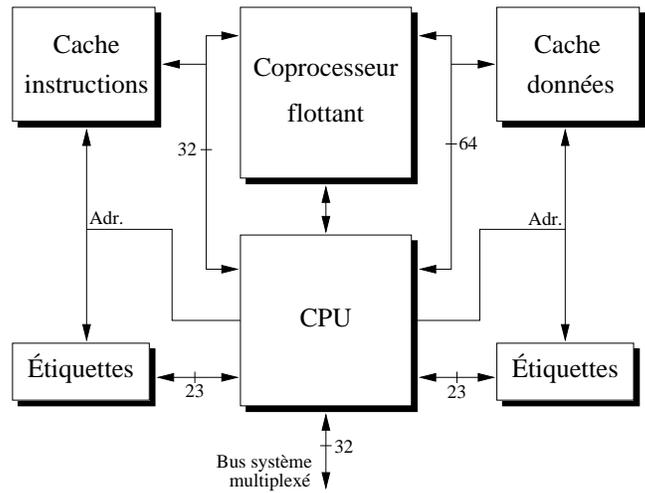


FIG. 6.6 – Le cœur du « Snake ».

<b>IR</b>	Lecture du cache d'instructions.
<b>OR</b>	Lecture des oprérandes.
<b>EX</b>	Exécution.
<b>DR</b>	Lecture du cache de données
<b>RW</b>	Écriture du registre destination.

FIG. 6.7 – Le pipeline du cœur du « Snake ».

### Hierarchie mémoire

Le CPU inclut un support pour deux caches séparés et externes pour les instructions et les données. Le fait de mettre en place des caches externes est une particularité que l'on retrouve dans la plupart des microprocesseurs PA-RISC. Ceci comporte deux avantages : leur taille peut être importante (réduction des défauts) et cela évite la mise en œuvre d'un cache secondaire (qui est souvent assez lent) avec tout le surcoût engendré. De la SRAM standard peut être utilisée pour ces caches à condition qu'elle soit assez rapide.

Un bus de 32 bits est prévu pour communiquer avec le cache d'instructions et un bus de 64 bits assure les communications avec le cache de données — il permet le chargement de données flottantes double précision en un cycle. Les deux caches sont virtuellement indexés et physiquement étiquetés. Ils sont aussi tous deux à correspondance directe avec la mémoire. Le cache de données est, quant à lui, à recopie différée. La cohérence est assurée par matériel grâce à deux bits pour chaque ligne : *privé* (*private*) et *sale* (*dirty*).

Deux TLB séparés de 96 entrées chacun sont fournis pour les instructions et les données. Un algorithme de remplacement de type NLU (*Not Last Used*) est mis en place.

### Bus système

Le bus système, large de 32 bits, est multiplexé. Il permet l'interfaçage du CPU avec la mémoire et les entrées/sorties, mais nécessite le(s) contrôleur(s) approprié(s).

### Technologie

La technologie de fabrication utilisée est CMOS à 1  $\mu\text{m}$  pour le CPU et 0,8  $\mu\text{m}$  pour le FPC. Les 577 000 transistors du CPU sont cadencés à 50 MHz ou 66 MHz.

## **Le PA-7100**

Le PA-7100, sorti en 1992, est le direct descendant du « Snake » présenté précédemment. De nombreuses unités logiques ont été reprises de son prédécesseur (dont l'unité entière) avec des transistors plus petits (CMOS à 0,8  $\mu\text{m}$ ) permettant une fréquence plus haute (100 MHz). L'unité flottante fait désormais partie du processeur et permet une exécution superscalaire des instructions.

L'aspect général du microprocesseur reste conforme à la figure 6.6, p. 125 (l'unité flottante est toutefois intégrée sur le même composant que l'unité centrale) : les caches primaires sont externes. D'autre part, le PA-7100 est conforme à l'architecture PA-RISC 1.1

### Séquencement des instructions

Le PA-7100 est un microprocesseur superscalaire de degré deux : il peut séquencer une instruction entière et une instruction flottante à chaque cycle — les instructions de lecture/écriture mémoire sont considérées comme des instructions entières car elles sont traitées par la même unité (tout comme sur le Snake).

Le choix des unités d'exécution se fait dès le cache d'instructions dans lequel un bit de pré-décodage par instruction sélectionne le chemin de données entier ou flottant.

Toute combinaison d'une instruction entière et d'une instruction flottante est valable. Il n'y a aucune contrainte en ce qui concerne l'ordre ou l'alignement sur la paire d'instructions qui sont exécutées ensemble.

### Pipeline

Le pipeline d'exécution comporte comme son prédécesseur cinq étages subdivisés en deux phases. La figure 6.8 montre comment est agencée l'exécution des différents types d'instructions dans le pipeline.

Au contraire des processeurs d'architecture Alpha de DEC, par exemple, qui privilégient une haute fréquence d'horloge, le 7100 essaie de réduire au maximum les pénalités dues aux blocages du pipeline. Le tableau 6.3 montre le nombre de cycles perdus dans certains cas.

	Chargement instruction	Décodage	Exécution	Accès cache de données	Écriture registre cible
<b>Accès mémoire</b>	Chargement instruction	Décodage	Calc. adr.	Accès cache de données	Écr. reg.
<b>Opération entière</b>	Chargement instruction	Décodage	Exécution		Écr. reg.
<b>Opération flottante</b>	Chargement instruction	Décodage	Exécution		Écr. reg.
<b>Branchement correctement prédit</b>	Chargement instruction	Calc. adr.	Chargement instr. cible Évaluation cond.		
<b>Branchement mal prédit</b>	Chargement instruction	Décodage	Évaluation cond.	Chargement instr. cible	

FIG. 6.8 – Le pipeline du PA-7100.

Cause de la pénalité	Cycles max. perdus
Branchement	1
Chargement-utilisation donnée	1
Écriture mémoire	1
Blocage ALU entière	0
Blocage ALU flottante	1
Blocage multiplication flottante	1

TAB. 6.3 – Blocages du pipeline du PA-7100.

### Unité flottante

L'unité flottante est différente de celle du Snake. Toutes les opérations, exceptées la division et la racine carrée, sont pipelinées; le tableau 6.4 en donne les latences et les débits en nombre de cycles.

Opération	Simple précision	Double précision
	latence / débit	latence / débit
Add./Soustr.	2 / 1	2 / 1
Multiplication	2 / 1	2 / 1
Multiplication/add.	2 / 1	2 / 1
Division	8 / 8	15 / 15
Racine carrée	8 / 8	15 / 15

TAB. 6.4 – Latences et débits en nombre de cycles des opérations flottantes du PA-7100.

### Hierarchie mémoire

La configuration de deux caches primaires externes pour les instructions et les données du Snake se retrouve sur le PA-7100. La logique de contrôle de ces caches est presque identique à celle de son prédécesseur: outre le changement de technologie, le bus de communication pour les instructions est passé de 32 bits à 64 bits (séquençement superscalaire de degré deux au lieu du séquençement scalaire du Snake) et la fréquence est passée de 66 MHz à 100 MHz. Les deux caches sont toujours virtuellement indexés et physiquement testés. Ils sont aussi tous deux à correspondance directe avec la mémoire.

Le cache d'instructions peut avoir une taille comprise entre 4 Koctets et 1 Moctets. La taille du cache de données peut aller de 4 Koctets à 2 Moctets. On utilise pour ceux-ci de la SRAM standard en choisissant un bon temps de réponse car les contrôleurs de cache sont cadencés à la fréquence du microprocesseur (100 MHz pour la configuration haut de gamme).

Le cache de données est non bloquant: les données s'y trouvant peuvent être accédées alors qu'une résolution de défaut est en cours. D'autre part, afin de réduire la latence des écritures mémoire et de les découpler, un tampon d'écriture d'une entrée est mis en place. La valeur contenue dans ce tampon est écrite dans le cache lors de la lecture de l'étiquette de la donnée suivante à écrire<sup>15</sup>.

Le PA-7100 met en œuvre un TLB unifié de 120 entrées pour des pages de taille normale (4 Koctets) et de 16 entrées pour des blocs plus importants pouvant aller de 512 Koctets à 64 Moctets. Ce tampon est totalement associatif et certaines de ses entrées peuvent être verrouillées par logiciel pour ne pas être intempestivement écrasées.

Pour éviter les conflits lors d'accès simultanés pour les données et les instructions, un sous-TLB d'une entrée conserve la traduction d'une instruction.

### Bus système

Le bus système utilisé par le PA-7100 est le même que celui du Snake, le *P-bus*. Bien que sa largeur ne soit que de 32 bits et qu'il soit multiplexé, il peut être cadencé à la fréquence du processeur: les fractions de fréquences du bus admises par rapport au processeur sont  $\frac{1}{2}$ ,  $\frac{2}{3}$  et 1. Le PA-7100 peut ainsi remplacer le Snake sur ses propres cartes sans grandes modifications (la fraction  $\frac{2}{3}$  séquence le bus à 66 MHz quand le CPU est cadencé à 99 MHz).

Dix-sept bits supplémentaires permettent le support de nombreux protocoles et la cohérence des caches et du TLB dans le cas de systèmes multiprocesseurs à mémoire partagée.

15. Ceci explique le cycle maximum de pénalité lors d'écritures mémoire, que l'on trouve dans le tableau 6.3, p. 127.

### Support multiprocesseurs

Le PA-7100 peut être intégré à des systèmes multiprocesseurs à mémoire partagée. Deux configurations sont possibles en plus d'une configuration uniprocasseur « classique » (FIG. 6.9). Cette dernière utilise un composant précieux aux deux autres, le *Processor-Memory Interface* (**PMI**) qui connecte le P-bus à la mémoire principale et au sous-système d'entrées/sorties.

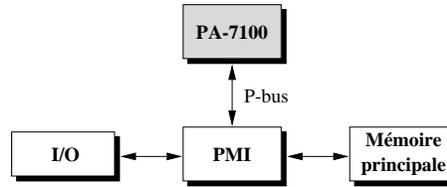


FIG. 6.9 – Configuration uniprocasseur avec le PA-7100.

Une configuration bi-processeurs « bon marché » est possible, ainsi que le montre la figure 6.10. Dans ce cas-ci, le P-bus est partagé entre les deux processeurs ; ceux-ci sont vus du PMI comme un seul PA-7100 très gourmand en bande passante. Les cohérences nécessaires à ce type de système sont gérées à l'intérieur de chaque processeur.

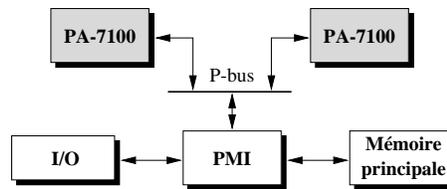


FIG. 6.10 – Configuration à deux PA-7100.

Enfin, une configuration multiprocesseurs haut de gamme est schématisée par la figure 6.11. Ici, chaque PMI « espionne » le bus partagé pour assurer la cohérence des données contenues dans chacun des processeurs.

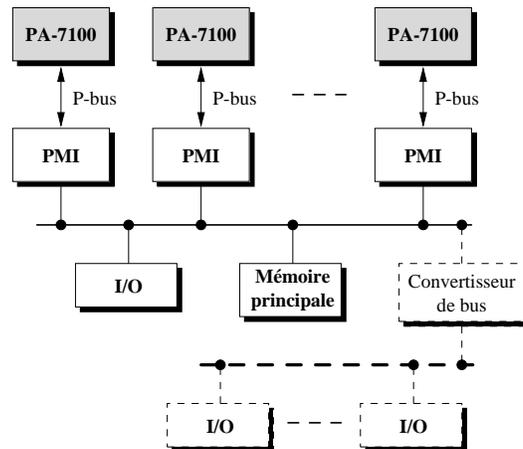


FIG. 6.11 – Configuration haut de gamme à plusieurs PA-7100.

## Technologie

Les 850 000 transistors CMOS à  $0,8 \mu\text{m}$  du PA-7100 occupent une place d'environ  $200 \text{ mm}^2$ . La fréquence d'horloge la plus haute est de 100 MHz (1992).

## Le PA-7100LC

Annoncé fin 1992, le PA-7100LC est destiné au marché des stations de travail uniprocasseur d'entrée de gamme (LC pour *Low Costs*) sur lequel HP n'était pas, à l'époque, bien implanté.

Afin de réduire les coûts des systèmes, le PA-7100LC intègre de nombreuses fonctions. C'est un microprocesseur superscalaire de degré deux, comportant deux unités entières, une unité flottante, une unité d'interface mémoire (MMU avec contrôleur de DRAM), un contrôleur de cache et une interface bus (FIG. 6.12).

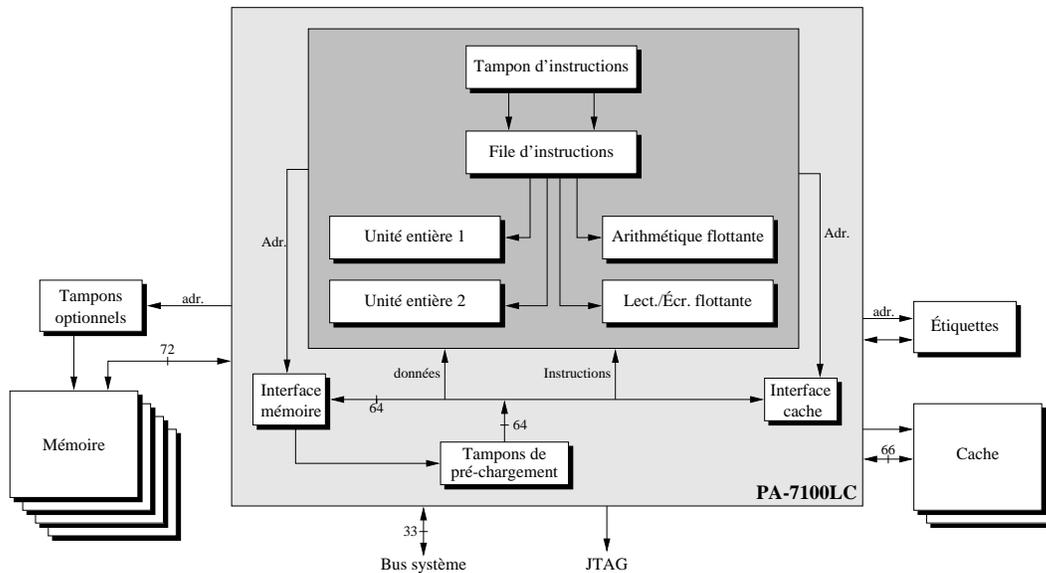


FIG. 6.12 – Synoptique du PA-7100LC

## Architecture et jeu d'instructions

Le PA-7100LC est conforme à l'architecture PA-RISC 1.1 (troisième édition) de niveau 1 (adresses virtuelles sur 48 bits, dont 16 d'espace d'adressage). Une des particularités intéressantes de ce processeur est qu'il supporte une extension au jeu d'instructions PA-RISC. Cette extension comporte des instructions destinées à accélérer les traitements de données multimédias : son, image, graphiques, ...

Les quelques instructions de l'extension portent le nom de MAX-1 (*Multimedia Acceleration eX-tension*) et précèdent celles de l'extension MAX-2 de PA-RISC 2.0 dont il est question p. 121. Ces instructions opèrent sur des demi-mots (16 bits) et permettent des traitements SIMD (les opérandes font  $32 \text{ bits} = 2 \times 16$ ).

Les instructions proposés par MAX-1 sont les suivantes :

- addition parallèle, soustraction parallèle ;
- décalage (droite ou gauche) et addition (pour multiplication ou division par une constante) ;
- moyenne arithmétique parallèle de deux valeurs.

Le tableau 6.2 (p. 122) détaille un peu plus ces instructions dans le cadre de MAX-2. Celles de MAX-1 ont la même signification à la différence qu'elles traitent des mots (32 bits) au lieu de double-mots.

Séquencement des instructions

Le PA-7100LC est superscalaire de degré deux et les instructions pouvant s'exécuter en parallèle sont détaillées dans le tableau 6.5.

Instructions provenant d'unités...	Commentaires
Flottante 1 – Flottante 2	Flottante 1: arithmétique flottante, Flottante 2: lect./écr. flottante.
Flottante 1 – Entière 1	Entière 1: ALU, décalages, branchements, lect./écr. entières.
Flottante 2 – Entière 1	Couplage possible sans lect./écr. entière.
Entière 1 – Entière 2	Deux lect./écr. doivent être du même type (lect. – lect. ou écr. – écr.) et les adresses doivent être consécutives et alignées.

TAB. 6.5 – Exécution superscalaire de degré deux sur le PA-7100LC.

Bien qu'elles effectuent toutes deux des opérations arithmétiques, les deux unités entières, qui font l'originalité de ce processeur, ne sont pas symétriques : seule l'unité 1 traite des branchements et des instructions de décalage. D'autre part, deux opérations dépendantes ne doivent pas être couplées.

Hierarchie mémoire

Le PA-7100LC comporte une interface pour un unique cache primaire externe. Ce cache est divisé en deux parties égales dont l'une est réservée aux instructions et l'autre aux données. Il est à correspondance directe et sa taille varie de 8 Koctets à 2 Moctets. D'autre part, il est non bloquant. Un bus séparé de 64 bits (+ 2 pour une protection des données par parité au niveau de chaque mot) permet le transit des données entre le CPU et le cache.

Pour qu'il n'y ait pas de gel du pipeline à chaque accès mémoire pour les données, un petit cache d'instructions interne d'un Koctets est prévu. Celui-ci est alimenté dès qu'aucune instruction d'accès mémoire n'est en train de s'exécuter.

La mémoire principale est reliée au processeur à travers un bus spécifique de 64 + 8 bits pour les données (protection par *code correction d'erreur*, ECC).

Bus système

Le bus système du PA7100LC ne sert que pour les entrées/sorties et les graphiques et est multiplexé. Il peut être cadencé à la moitié ou au tiers de la vitesse du CPU.

Technologie

La technologie utilisée pour fabriquer les 800 000 transistors du PA-7100LC est CMOS à 0,8  $\mu\text{m}$ . Il mesure 196 mm<sup>2</sup>. La fréquence maximale atteinte est 100 MHz.

**Le PA-7150**

Sorti début 1994, le PA-7150 n'est pas réellement un nouveau processeur, mais une amélioration du PA-7100. En effet, le PA-7150 est fonctionnellement équivalent au PA-7100 et ses caractéristiques technologiques sont les mêmes, excepté la fréquence qui est poussée à 125 MHz sur le PA-7150. Cette fréquence a pu être atteinte en élargissant les chemins critiques du PA-7100 et confère au PA-7150 de meilleures performances. On notera tout de même que la SRAM qui fait les caches externes devra être encore plus rapide.

## Le PA-7200

Annoncé début 1994, le PA-7200 est un microprocesseur qui se veut bon marché pour les systèmes multiprocesseurs. Il constitue un autre descendant du PA-7100 et, comme lui, est superscalaire de degré deux. Il intègre en plus du PA-7100 une deuxième unité entière, un petit cache interne et une nouvelle interface bus qui permet de connecter quatre microprocesseurs sans logique supplémentaire. La fréquence initialement prévue était de 140 MHz, mais à sa sortie il n'a atteint que 100 MHz ; un an après la fréquence a pu être amenée à 120 MHz.

### Pipeline

Le pipeline classique de cinq étages du PA-7100 se retrouve sur le PA-7200. Les pénalités les plus fréquentes se rencontrent dans les cas suivants (un cycle est perdu à chaque fois) :

- enchaînement « chargement-utilisation » d'une donnée (*load-use*) ;
- branchement mal prédit ;
- écriture mémoire suivie par une lecture.

La pénalité due à deux écritures consécutives a été éliminée grâce aux progrès des SRAM utilisées pour les caches.

Pour faciliter le décodage de deux instructions en un cycle, six bits de pré-décodage ont été ajoutés par paire d'instructions.

### Unités arithmétiques et logiques

Les deux unités entières ne sont pas symétriques ; ce ne sont d'ailleurs pas les mêmes que celles du PA-7100LC. Une seule parmi les deux gère les accès mémoire et les décalages. Les autres traitements sont assurés par les deux unités.

L'unité flottante reste la même que celle du PA-7100 (voir tableau 6.4, p. 128).

### Hiérarchie mémoire

Comme son prédécesseur, le PA-7200 comporte deux interfaces pour des caches externes d'instruction et de données. Ceux-ci sont à correspondance directe et indexés virtuellement alors que testés physiquement.

Une des particularités du PA-7200 réside dans son petit cache interne, dit d'*assistance*, destiné à limiter les défauts de conflits du cache primaire de données, fréquents sur ce type de cache à correspondance directe<sup>16</sup>. Ce petit cache sert aussi de tampon entre le cache externe et la mémoire : toute donnée lue depuis la mémoire transite par le cache d'assistance avant de rejoindre le cache primaire externe selon une file FIFO<sup>17</sup>.

Le cache d'assistance est associatif et comporte 64 entrées de 32 octets (taille d'une ligne de cache). Celui-ci est physiquement adressé pour éliminer les problèmes dus à l'*aliasing* ; ceci réduit aussi la taille de la table d'étiquettes.

### Bus système et support multiprocesseurs

Le bus système du PA-7200 est différent de celui de son prédécesseur et possède une bande passante bien plus importante ; il porte le nom de *Runaway*. Il peut être cadencé à 120 MHz et multiplexe les adresses et les données sur 64 bits (les signaux de contrôle sont séparés). Il utilise le protocole des transactions imbriquées qui lui permet de soutenir jusqu'à six transactions actives (par processeur). Les fractions de fréquences de bus par rapport à celles du CPU sont  $\frac{2}{3}$ ,  $\frac{3}{4}$  et 1.

*Runaway* supporte jusqu'à quatre PA-7200 directement connectés, comme sur la figure 6.13. La cohérence des caches est assurée par chaque processeur qui examine chaque transaction du bus.

16. À ce titre, le cache d'assistance n'est pas considéré comme un cache primaire.

17. *First In, First Out*.

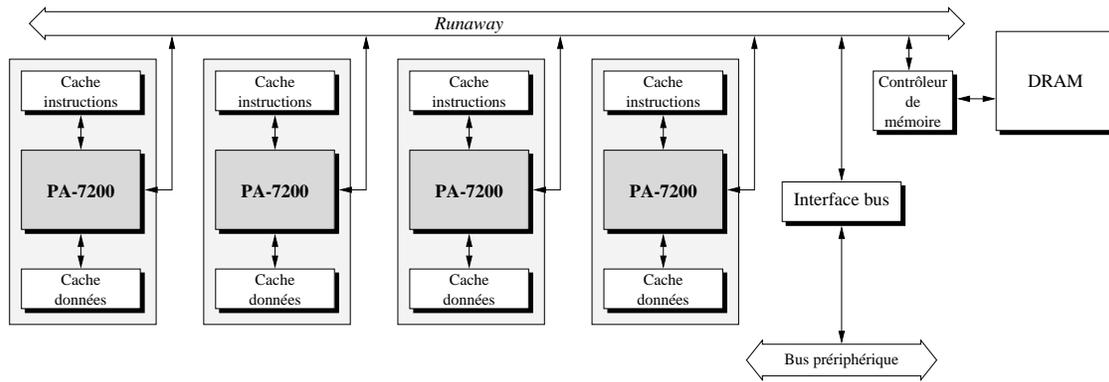


FIG. 6.13 – Configuration multiprocesseurs à base de PA-7200.

Technologie

Les fréquences de 100 MHz, puis 120 MHz sont atteintes grâce à un procédé de fabrication CMOS à 0,55  $\mu\text{m}$ . Les 1,26 millions de transistors occupent 210  $\text{mm}^2$  et sont alimentés en 4,4 V (les entrées/sorties sont alimentées en 3,3 V)

**Le PA-7300LC**

Apparu en octobre 1995, le PA-7300LC est le dernier processeur PA-RISC 32 bits. Ce microprocesseur a été conçu pour s'intégrer dans des systèmes d'entrée et moyen de gamme, comme le PA-7100LC dont il est issu. D'ailleurs le cœur du PA-7300LC est le même que celui du PA-7100LC. Sa particularité la plus importante est que les deux caches primaires sont internes (FIG. 6.14), ce qui est une première pour les microprocesseurs PA-RISC (le PA-7200 avec son petit cache interne étant exclu).

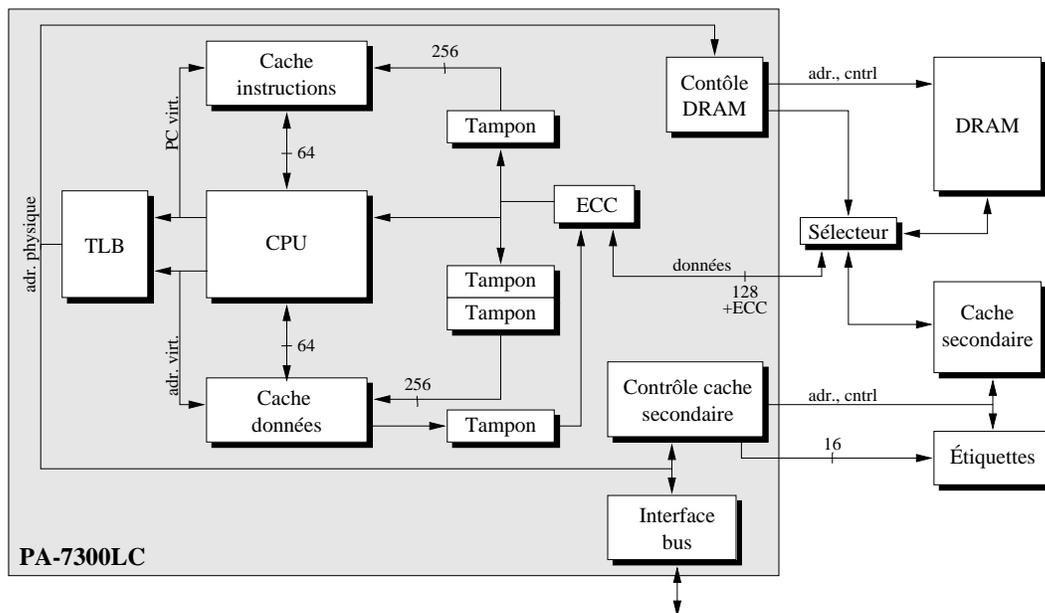


FIG. 6.14 – Le PA-7300LC : synoptique.

### Cœur du processeur

Le cœur du processeur est le même que celui du PA-7100LC : exécution superscalaire de degré deux grâce à une unité flottante et deux unités entières. Le jeu d'instructions est le même que celui du PA-7100LC, y compris l'extension multimédia. Pour plus de détails, nous invitons le lecteur à consulter la partie concernant le PA-7100LC en page 130.

### Hiérarchie mémoire

La principale distinction du PA-7300LC par rapport à ses prédécesseurs réside dans les deux caches primaires d'instructions et de données, intégrés au processeur. Ces deux caches ont une taille de 64 Koctets chacun, sont associatifs par ensembles à deux voies et sont virtuellement indexés. Le cache de données est non-bloquant : il peut servir des requêtes avec au plus deux défauts en cours de traitement ; ceci confère au processeur la possibilité d'une exécution partiellement dans le désordre.

Le PA-7300LC intègre un support pour un cache secondaire externe unifié, physiquement indexé et dont la taille peut aller de 512 Koctets à 8 Moctets.

Pour ce cache on utilise de la SRAM synchrone cadencée à la moitié de la fréquence interne (pour des systèmes à moindre coût, la fréquence de la SRAM synchrone est plus lente), ce qui permet de construire des systèmes à coût limité. La communication entre la SRAM et le microprocesseur se fait sur un bus de 128 bits pour les données (bus d'adresses séparé).

La mémoire principale partage le large bus du cache secondaire et est gérée par un contrôleur de DRAM interne. Ce contrôleur est souple puisqu'il supporte de nombreux types de DRAM, dont l'EDO DRAM<sup>18</sup>.

Pour les traductions d'adresses virtuelles en adresses physiques, on trouve un TLB associatif de 96 entrées similaire à celui du PA-7100LC.

### Bus système

Le bus système, séparé du bus mémoire, ne sert que pour les entrées/sorties. Il est multiplexé sur 32 bits et est identique à celui du PA-7100LC (appelé *GSC bus* par *HP*).

### Technologie

Le plus grand gain en performance pour le PA-7300LC provient du procédé de fabrication CMOS à 0,5  $\mu\text{m}$ . Les 9,2 millions de transistors (dont presque 8 servent pour les caches primaires) occupent 259 mm<sup>2</sup>. La fréquence interne du PA-7300LC est de 160 MHz.

## **6.2.2 Le microprocesseur d'architecture PA-RISC 2.0 (PA-8000)**

Annoncé dès fin 1994, le PA-8000 n'est réellement sorti que mi-1996. Dernier né de la famille PA-RISC, il est conforme à l'architecture 64 bits PA-RISC 2.0. Une architecture découplée et de nombreuses unités d'exécution (FIG. 6.15) lui permettent une exécution spéculative et dans le désordre de quatre instructions par cycle.

Ces caractéristiques font du PA-8000 le microprocesseur le plus puissant de la gamme de microprocesseurs d'architecture PA-RISC actuels, et c'est pourquoi il est destiné aux stations de travail moyen et haut de gamme, ainsi qu'aux serveurs.

### **Séquencement des instructions**

À chaque cycle, quatre instructions pré-décodées (par cinq bits) sont chargées depuis le cache d'instructions. Elles sont ensuite séparées en deux catégories, les instructions dites de calcul et les instructions d'accès mémoire, avant d'être stockées temporairement dans le cœur du PA-8000 : le tampon d'ordonnancement des instructions (*Instruction Reorder Buffer – IRB*). Ce tampon séparé

18. *Extended Data Out Dynamic Random Access Memory*.

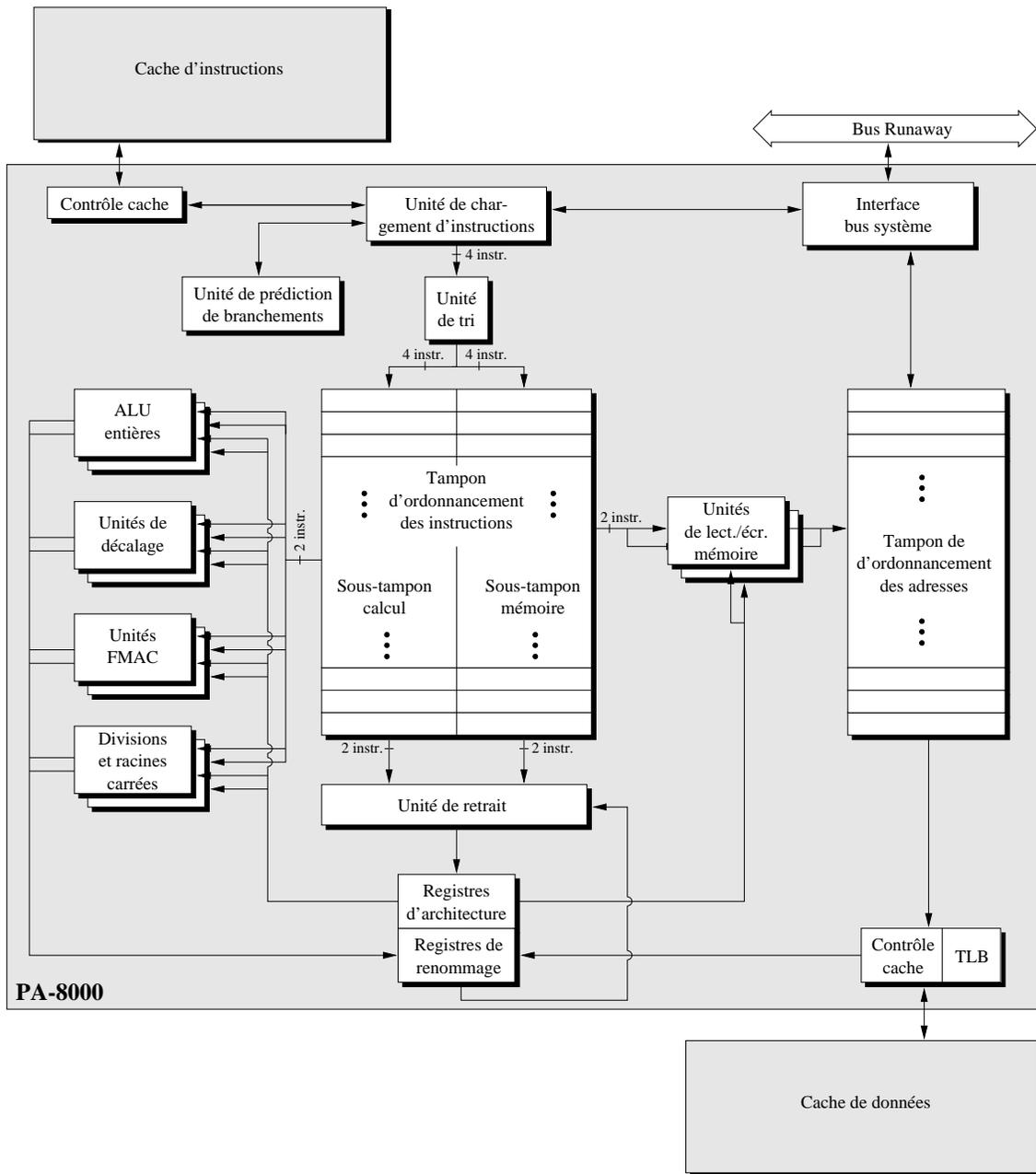


FIG. 6.15 – Architecture du PA-8000.

en deux parties égales et indépendantes (instructions de calcul et instructions d'accès mémoire) peut contenir jusqu'à 56 instructions au total. Les instructions sont ensuite lancées directement dans les unités d'exécution (pas de station de réservation) dès que leurs opérandes sont disponibles, à raison d'au plus deux instructions de calcul et deux instructions d'accès mémoire. Lorsque plus de deux instructions d'une catégorie donnée sont prêtes à être exécutées, les deux plus « anciennes » ont la priorité.

Le modèle ordonné des instructions est maintenu par l'unité de retrait qui se charge de retirer dans l'ordre les instructions du tampon d'ordonnancement après leur exécution. L'unité de retrait recopie aussi les registres de renommage dans les registres d'architecture<sup>19</sup> : les registres de renommage servent à inscrire les résultats des instructions qui ne sont pas encore retirées (encore actives) ; lors du retrait, les registres d'architecture sont mis à jour dans l'ordre du programme. De plus, dans le cas d'instructions d'écriture mémoire, les valeurs à écrire sont insérées en ordre dans une file avant d'être écrites dans le cache.

## Pipeline et prédiction de branchement

Le découplage de l'architecture se traduit sur le PA-8000 par plusieurs pipelines. Ceux-ci comportent une partie commune, mais diffèrent selon le type d'instruction, comme le montre la figure 6.16.

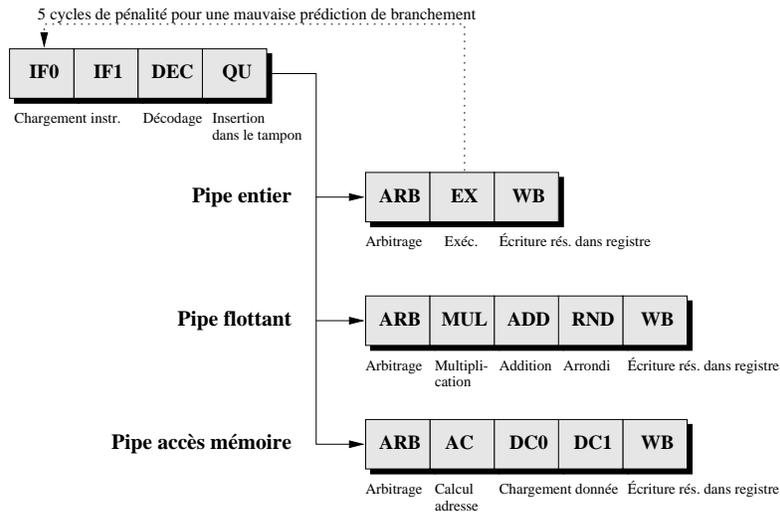


FIG. 6.16 – Les pipelines du PA-8000.

Comme sur la plupart des microprocesseurs d'architecture PA-RISC, les caches primaires d'instructions et de données sont externes. Leurs accès sont donc plus lents que s'ils étaient intégrés, ce qui explique les deux cycles d'accès pour les caches (chargement instruction et chargement donnée dans le pipeline mémoire) ; en effet, ceci permet de ne pas créer de chemin critique<sup>20</sup>.

La pénalité de mauvaise prédiction de branchement s'élève à 5 cycles minimum, ce qui n'est pas négligeable. Ainsi, une bonne prédiction est primordiale. Pour ce faire deux modes sont disponibles : dynamique et statique (chaque entrée du TLB contient un bit spécifiant ce mode).

En mode statique, seuls les sauts « en arrière » sont prédits pris. Le PA-8000 implémente aussi ce qui est spécifié par l'architecture PA-RISC 2.0 (voir p. 123).

19. Les « registres d'architecture » correspondent, ici, aux registres spécifiés par PA-RISC 2.0.

20. Le PA-8000 est le plus rapide des microprocesseurs PA-RISC puisqu'il est cadencé à 180 MHz. Ceci explique peut-être cette particularité.

En mode dynamique, l'unité de chargement des instructions consulte une table d'historique de branchements de 256 entrées qui contient les résultats des trois dernières occurrences de chaque branchement. La prédiction consiste à choisir la majorité (pris ou non-pris).

Pour éviter une pénalité supplémentaire lors de branchements pris, un tampon associatif d'adresses cibles (*Branch Target Address Cache*) contient 32 adresses cibles en rapport avec l'instruction de branchement. Toutes les entrées concernent des branchements pris : si le branchement est prédit non-pris, l'entrée est otée.

## Unités d'exécution

Toutes les unités d'exécution sont doubles pour favoriser un bon débit d'exécution (FIG. 6.15, p. 135). Chacune d'entre elles est alimentée par le tampon d'ordonnancement des instructions et fournit un résultat aux registres de renommage dans un premier temps.

### Unités entières

Pour les calculs entiers, deux ALU et deux unités de décalages sont disponibles. Toutes les opérations entières ont un unique cycle de latence.

Certaines opérations d'accès mémoire nécessitent l'intervention d'un calcul entier. De telles instructions sont coupées en deux dès le tampon d'ordonnancement en une instruction de calcul entier et une instruction d'accès mémoire. La sous-instruction de calcul entier est ensuite exécutée dans une des unités entières.

### Unités flottantes

Les quatre unités d'exécution flottantes se divisent en deux unités *FMAC* (*Floating Multiply-Accumulate*) et deux unités destinées aux divisions et racines carrées.

Les unités FMAC effectuent des multiplications-additions ( $A \times B + C \rightarrow C$ ). Ces opérations sont pipelinées et s'étalent sur trois cycles (FIG. 6.16, p. 136).

Les unités de division/racine carrée ne sont pas pipelinées. Ces opérations prennent 17 cycles en simple précision et 31 cycles en double précision.

### Unités d'accès mémoire

Les deux unités d'accès mémoire réalisent toutes les opérations de lecture/écriture mémoire et peuvent gérer à elles deux, deux accès mémoire par cycle.

L'intérêt de ces deux unités réside dans l'aspect double-port du cache de données. Deux accès peuvent avoir lieu simultanément s'ils concernent des bancs du cache de données différents.

## Hiérarchie mémoire

Hormis les registres, aucun élément de la hiérarchie mémoire ne se trouve sur le composant : les deux caches d'instructions et de données séparées sont externes, caractéristique se retrouvant sur presque tous les microprocesseurs PA-RISC. Les contrôleurs de caches associés sont internes et permettent d'implanter des caches à correspondance directe d'au plus 4 Moctets (1 Moctets si le PA-8000 est cadencé à son maximum 180 MHz) avec de la SRAM synchrone standard. Ces deux caches sont reliés au processeur par deux bus de 128 bits de large chacun<sup>21</sup>.

Le cache de données comporte deux ports et est structuré en deux bancs entrelacés. Ainsi deux accès peuvent avoir lieu par cycle à condition qu'ils s'adressent à des bancs différents. Afin de servir les requêtes de cette manière, les étiquettes sont dupliquées.

Pour soutenir les deux accès par cycle des unités mémoire, le PA-8000 comporte un TLB associatif de 96 entrées possédant deux ports. Un micro-TLB de quatre entrées est dédié aux instructions.

21. Pour les données le bus est partagé en  $2 \times 64$  bits permettant deux accès simultanés.

## Bus système et support multiprocesseurs

Le bus système supporté par le PA-8000 est le même que celui du PA-7200 (voir p. 132) : le bus *Runaway*. Celui-ci multiplexe les adresses et les données sur 64 bits et peut être cadencé jusqu'à 120 MHz. Le protocole des transactions imbriquées lui permet de soutenir jusqu'à six transactions actives (par processeur).

Comme avec le PA-7200, le bus *Runaway* supporte jusqu'à quatre processeurs directement connectés, dans une configuration multiprocesseurs à mémoire partagée (FIG. 6.13, p. 133).

## Technologie

Le PA-8000 est réalisé en CMOS à 0,5  $\mu\text{m}$  et comporte 3,8 millions de transistors. Le composant mesure 337,6 mm<sup>2</sup> et est alimenté sous 3,3 V. La fréquence maximum d'utilisation est de 180 MHz.

## 6.3 Microprocesseurs à venir

*Hewlett-Packard* a annoncé l'extension de la famille des microprocesseurs d'architecture PA-RISC 2.0 avec deux processeurs dérivés du PA-8000, le PA-8200 et le PA-8500.

### 6.3.1 Le PA-8200

Dévoilé depuis octobre 1996, le PA-8200 devrait équiper les systèmes *HP* aux alentours de la mi-1997. Ce microprocesseur maximise les améliorations de performances tout en minimisant les changements vis-à-vis de son prédécesseur. En effet, la taille du composant et son brochage restent les mêmes.

Des études sur le PA-8000 ont mis en exergue deux goulots d'étranglement qui sont éliminés sur le PA-8200 :

- Le taux de bonne prédiction de branchement est assez pauvre sur le PA-8000, et pour cela (la pénalité de mauvaise prédiction est de cinq cycles), la table d'historique de branchement a été agrandie de 256 à 1024 entrées. Un problème empêchant deux mises à jour par cycle de cette table d'historique a été résolu (le retrait de deux instructions de branchement dans un cycle est possible).
- Les pénalités de TLB sont très importants sur le PA-8000. Le TLB du PA-8200 comportera 120 entrées au lieu de 96.

Le procédé de fabrication du PA-8200 reste le même, CMOS à 0,5  $\mu\text{m}$ . Cependant, les améliorations concernant la technologie des SRAM synchrones (dont les deux caches externes sont composés) devraient permettre de cadencer le PA-8200 à 220 MHz. Ceci lui conférerait temporairement des performances au dessus de ses concurrents, malgré le procédé de fabrication vieillissant.

### 6.3.2 Le PA-8500

Le PA-8500 est un dérivé des PA-8000 et PA-8200. Annoncé en mars 1997, il devrait être produit en volume après la mi-1998. Quelques rares modifications ont été apportées au cœur du microprocesseur, mais l'originalité et la puissance du PA-8500 réside dans ses énormes caches internes (1,5 Moctets au total) qui occuperont les  $\frac{3}{4}$  de la taille du composant.

La seule modification notable du cœur du microprocesseur par rapport au PA-8200 concerne la prédiction de branchement. Le PA-8500 combine les avantages d'une prédiction statique et dynamique, alors que les PA-8000 et PA-8200 comportaient un mode dans lequel la prédiction était soit statique,

soit dynamique. La table d'historique de branchement du PA-8500 contient un compteur à deux bits par entrée. Ce compteur s'incrémente si la prédiction statique s'avère bonne et se décrémente sinon. La prédiction dynamique détermine un branchement en fonction du compteur : si celui-ci est à 0 ou 1, alors les indications de la prédiction statique sont suivies, sinon le contraire est décidé.

Le PA-8500 tire partie d'un procédé de fabrication CMOS à 0,25  $\mu\text{m}$  plus actuel. La réduction ainsi apportée de la taille du composant a été comblée par l'ajout de caches internes associatifs par ensembles à quatre voies. Pour les instructions, 0,5 Moctets sont prévus. Pour les données, le PA-8500 dispose de 1 Moctets. La taille importante de ces caches permet de s'abstenir de cache secondaire et élimine les coûts dus à l'ajout de caches externes primaires ou secondaires.

Une autre modification, plus légère concerne le TLB : celui-ci passe de 120 à 160 entrées.

La fréquence interne du PA-8500 n'a pas été donnée, mais elle pourrait être de l'ordre de 400 MHz.



	MIPS R10000	DEC 21164	IBM P2SC	PowerPC 604e	Sun UltraSPARC-II	Intel PentiumPro	AMD K6	HP PA-8000
Fréquence	275 MHz	500 MHz	135 MHz	233 MHz	250 MHz	200 MHz	233 MHz	180 MHz
Superscalaire	4 voies	4 voies	6 voies	4 voies	4 voies	3 voies*	2 voies	4 voies
Nb. étages du pipe	5 – 7	7 – 9	5 – 8	6	6 – 9	12 – 14	6 – 7	7 – 9
Exéc. dans le désordre	32 instr.	6 lect.	non	16 instr.	non	40 ROP	12 instr.	56 instr.
Renommage de registres	oui	non	oui	oui	non	oui	non	oui
Taille des caches (I/D en Koctets)	32/32	8/8/96	32/128	32/32	16/16	8/8	32/32	externe
TLB (nb. entrées, I/D)	64	48/64	64/64	128/128	64/64	32/64	128/64	96
Technologie (CMOS)	0,35 $\mu\text{m}$	0,35 $\mu\text{m}$	0,29 $\mu\text{m}$	0,27 $\mu\text{m}$	0,29 $\mu\text{m}$	0,35 $\mu\text{m}^{**}$	0,35 $\mu\text{m}$	0,5 $\mu\text{m}$
Nb. transistors (en millions)	6,7	9,3	15	5,1	3,8	5,5	8,8	3,8
Taille	298 mm <sup>2</sup>	209 mm <sup>2</sup>	335 mm <sup>2</sup>	96 mm <sup>2</sup>	149 mm <sup>2</sup>	196 mm <sup>2</sup>	157 mm <sup>2</sup>	337 mm <sup>2</sup>
SPEC95 (int/fp) <sup>†</sup>	10,7/17,4	<b>12,6/18,3</b>	5,9/15,4	9,0/8,5 <sup>‡</sup>	8,5/15	8,7/6,0*	???	10,8/ <b>18,3</b>

\*. Instructions x86.

\*\*.

†. Données valables en avril 1997 et susceptibles de changements rapides.

‡. Estimation de *MicroDesign Resources*.

\*. Version à 512 Koctets de cache secondaire.

TAB. 6.6 – *Caractéristiques principales des microprocesseurs hautes performances.*



# Bibliographie

- [1] André Seznec et Yann Mével. Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86. Rapport de recherche 2746, INRIA, décembre 1995.
- [2] André Seznec et Patrice Laporte. Une étude comparative des microprocesseurs MIPS R3000, Sun SPARC Version 7 et IBM POWER: Architectures et performances. Rapport de recherche, IRISA, février 1992.
- [3] André Seznec, Anne-Marie Kermarrec et Thierry Vaulton. Étude comparée des architectures des microprocesseurs MIPS R4000, DEC 21064 et T.I. SUPERSPARC. Rapport de recherche 1836, INRIA, janvier 1993.
- [4] André Seznec et Thierry Vaulton. Étude comparative des architectures des microprocesseurs Intel Pentium et PowerPC 601. Rapport de recherche 835, IRISA, juin 1994.
- [5] André Seznec et Yann Mével. Étude des architectures des microprocesseurs DEC 21164, IBM POWER2 et MIPS R8000. Rapport de recherche 2553, INRIA, Juin 1995.
- [6] André Seznec et Fabien Lloansi. Étude des architectures des microprocesseurs MIPS R10000, UltraSPARC et PentiumPro. Rapport de recherche 2893, INRIA, mai 1996.
- [7] Charles Price. *MIPS IV Instruction Set*. MIPS Technologies, Inc., janvier 1995.
- [8] L. Gwennap. Digital, MIPS Add Multimedia Extensions. *Microprocessor Report*, pages 24 – 28, novembre 1996.
- [9] *MIPS V Specification*, 1996.
- [10] J. Turley. LSI's TinyRisc Core Shrinks Code Size. *Microprocessor Report*, pages 40 – 44, octobre 1996.
- [11] Norman P. Jouppi J. Bradley Chen, Anita Borg. A Simulation Based Study of TLB Performance. In *The 19th Annual International Symposium on Computer Architecture*, volume 20, pages 114 – 123, Mai 1992.
- [12] Steve Rhodes. MIPS R5000 processor preliminary software interface specification. Technical report, Silicon Graphics Computer Systems, novembre 1995.
- [13] Steve Rhodes. MIPS R5000 processor preliminary bus interface specification. Technical report, Silicon Graphics Computer Systems, novembre 1995.
- [14] Digital Equipment Corporation. *Alpha Architecture Handbook*. Maynard, Massachusetts, 3<sup>e</sup> édition, octobre 1996.
- [15] Digital Equipment Corporation. *DECchip 21064-AA Microprocessor - Hardware Reference Manual*, first edition, octobre 1992.
- [16] L. Gwennap. Digital 21264 Sets New Standard. *Microprocessor Report*, 10(14):11 – 16, octobre 1996.

- [17] Scott Mc Farling. Combining Branch Predictors. In *WRL – Technical Note TN-36*. Western Research Laboratory, juin 1993. <http://www.research.digital.com/wrl/techreports/abstracts/TN-36.html>.
- [18] J. Turley. Alpha Runs x86 Code with FX!32. *Microprocessor Report*, 10(3):11 – 13, mars 1996.
- [19] S. W. White and S. Dhawan. POWER2: Next Generation of the RISC System/6000 Family, 1996. <http://www.rs6000.ibm.com/resource/technology/POWER2.2.html>.
- [20] L. Gwennap. Exponential’s PowerPC Blazes. *Microprocessor Report*, 10(14), octobre 1996.
- [21] TEMIC Semiconductors. *SPARClet TSC701 Communication Controller – User’s manual*, 1996.
- [22] *Intel Architecture Software Developer’s Manual, Volume 1: Basic Architecture*, 1997. <http://developer.intel.com/design/pentium/manuals/24319001.pdf>.
- [23] Tse-Yu Yeh and Yale Patt. Two-level adaptive training branch prediction. In *24th International Symposium on Microarchitecture*, pages 51 – 61, 1991.
- [24] Hewlett-Packard. *PA-RISC 1.1 Architecture and Instruction Set Reference Manual*, 3rd. edition, February 1994.
- [25] Hewlett-Packard. *PA-RISC 1.1 Architecture and Instruction Set Reference Manual*, 1st. edition, November 1990.
- [26] Hewlett-Packard. *PA-RISC 1.1 Architecture and Instruction Set Reference Manual*, 2nd. edition, November 1992.
- [27] Kane G. *PA-RISC 2.0 Architecture*. Prentice Hall, 1995.
- [28] Bernard Ourghanlian. *Les microprocesseurs ALPHA*. InterÉditions, 1995.
- [29] J. Smith. A study of branch prediction strategies. In *8th International Symposium on Computer Architecture*, pages 135 – 148, mai 1981.

# Index

## — A —

AIX  
  Advanced Interactive eXecutive ..... 58  
ALU  
  Arithmetic and Logical Unit ..... 24  
AMD  
  Advanced Micro Devices ..... 87  
ANDES  
  Architecture with Non-sequential Dynamic  
  Execution ..... 10  
APIC  
  Advanced Priority Interrupt Controller 94  
ASE  
  Application Specific Extension ..... 8  
ASIC  
  Application Specific Integrated Circuit 66  
ASID  
  Address Space IDentifier ..... 15  
ATM  
  Asynchronous Transfert Mode ..... 84

## — B —

BCD  
  Binary-Coded Decimal ..... 87  
BHT  
  Branch History Table ..... 108  
BiCMOS  
  Bipolar-CMOS ..... 62  
BTB  
  Branch Target Buffer ..... 93  
BTC  
  Branch Target Cache ..... 108

## — C —

CISC  
  Complex Instruction Set Computer ... 50  
CMOS  
  Complementary Metal-Oxyde Semiconduc-  
  tor ..... 4  
CPU

Central Processing Unit ..... 16

## — D —

DCU  
  Data Cache Unit ..... 51  
DEC  
  Digital Equipment Corporation ..... 31  
DMA  
  Direct Memory Access ..... 26  
DRAM  
  Dynamic Random Access Memory .... 55  
DSP  
  Digital Signal Processor ..... 84

## — E —

ECC  
  Error Correction Code. .... 14  
ECL  
  Emitter Coupled Logic ..... 4  
EDO DRAM  
  Extended Data Out Dynamic Random Ac-  
  cess Memory ..... 134

## — F —

FIFO  
  First In, First Out ..... 84  
FMAC  
  Floating Multiply-ACcumulate ..... 137  
FPC  
  Floating Point Coprocessor ..... 124  
FPQ  
  Floating-Point Queue ..... 10  
FPU  
  Floating Point Unit ..... 12  
FXU  
  FiXed point Unit ..... 51

## — H —

HDLC	
High-level Data Link Control .....	84

## — I —

I <sup>2</sup> C	
Inter-Integrated Circuit .....	65
IBM	
International Business Machines .....	49
ICU	
Instruction Cache Unit .....	51
IRB	
Instruction Reorder Buffer .....	134
IU	
Integer Unit .....	75

## — J —

JTAG	
Join Test Action Group .....	66

## — L —

LRA	
Least Recently Allocated .....	111
LRU	
Least Recently Used .....	14

## — M —

MAX	
Multimedia Acceleration eXtension ..	121
MBus	
Module Bus .....	74
MCM	
MultiChip Module .....	55
MDMX	
MIPS Digital Media eXtension .....	8
MESI	
Modified, Exclusive, Shared, Invalid ..	17
MIPS	
Microprocessor without Interlocked Pipe-	
line Stages .....	3
MMU	
Memory Management Unit .....	58
MMX .....	89
MOB	
Memory Order Buffer .....	105

## MOESI

Modified, Owned, Exclusive, Shared, Inva-	
lid .....	78

## MPEG

Moving Pictures Experts Group .....	33
-------------------------------------	----

## MROM

Microcode Read-Only Memory .....	99
----------------------------------	----

## MRU

Most Recently Used .....	93
--------------------------	----

## MSI

Modified, Shared, Invalid .....	78
---------------------------------	----

## MTI

MIPS Technology Incorporated .....	3
------------------------------------	---

## MVI

Motion-Video Instructions .....	33
---------------------------------	----

## — N —

## NLU

Not Last Used .....	40
---------------------	----

## NMOS

Negative channel Metal-Oxide Semiconduc-	
tor .....	124

## NOP

NO Operation .....	109
--------------------	-----

## — P —

## PA-RISC

Precision Architecture, Reduced Instruction	
Set Computer .....	115

## PALcode

Privileged Architecture Library code ..	32
---	----

## PC

Personal Computer .....	3
Program Counter .....	32

## PCI bus

Peripheral Connect Interface bus .....	47
--	----

## PCM

Pulse Code Modulation .....	84
-----------------------------	----

## PCMCIA

Personal Computer Memory Card Interna-	
tional Association .....	28

## PDA

Personal Digital Assistant .....	3
----------------------------------	---

## PIA

Peripheral Interface Adapter .....	84
------------------------------------	----

## PLL

Phase-Locked Loop .....	94
-------------------------	----

- PMI  
Processor Memory Interface ..... 129
- POWER  
Performance Optimized With Enhanced RISC 49
- PRO  
Precision RISC Organization ..... 115
- PSW  
Processor Status Word ..... 117
- Q —
- QED  
Quantum Effect Design ..... 19
- R —
- RAM  
Random Access Memory ..... 47
- RAMDAC  
Random Access Memory Digital-to-Analog  
Converter ..... 102
- RISC  
Reduced Instruction Set Computer .... 3
- ROB  
ReOrder Buffer ..... 104
- ROP  
RISC Operations ..... 99
- RS/6000  
RISC System 6000 ..... 49
- RSC  
RISC Single Chip ..... 49
- S —
- SCC  
Serial Communication Controller ..... 65
- SCU  
Storage Control Unit ..... 51
- SDRAM  
Synchronous Dynamic Random Access Me-  
mory ..... 84
- SIMD  
Single Instruction, Multiple Data ..... 8
- SP  
Scalable POWERparallel ..... 55
- SPARC  
Scalable Processor ARChitecture ..... 69
- SPI  
Serial Peripheral Interface ..... 65
- RR n° 3188
- SRAM  
Static Random Access Memory ..... 22
- T —
- TLB  
Translation-Lookaside Buffer ..... 15
- TTL  
Transistor-Transistor Logic ..... 124
- U —
- UART  
Unidirectional Asynchronous Receiver and  
Transmitter ..... 26
- UDB  
UltraSPARC Data Buffer ..... 79
- UMA  
Unified Memory Architecture ..... 102
- UPA  
UltraSPARC Port Architecture ..... 80
- USART  
Universal Synchronous/Asynchronous Re-  
ceiver and Transmitter ..... 84
- V —
- VAX  
Virtual Address eXtension ..... 31
- VIS  
Visual Instruction Set ..... 69, 73
- VLSI  
Very Large Scale Integration ..... 3
- VSA  
Virtual System Architecture ..... 102
- W —
- Windows CE  
Windows Consumer Electronics ..... 28



# Glossaire

## ACE/ARC systems

Les systèmes de type *ARC* (*Advanced RISC Computing*) obéissent aux spécifications mises en œuvre par MIPS et ses partenaires à travers l'initiative *ACE* (*Advanced Computing Environment*) (1992). Celles-ci s'adressent au marché des *desktop computers* à base de processeurs MIPS et visent à établir une architecture cible pour le développement de logiciels. L'un des buts essentiels de cette initiative est de concurrencer le marché à base de processeur *x86* en développant une architecture compatible.

## Alpha AXP

Il semblerait que les trois lettres *AXP* accolées à Alpha pour désigner le nom commercial de l'architecture soient l'acronyme de *Almost eXactly Prism* (selon [28]). *Prism* désigne un projet d'architecture RISC présenté lors de la conférence *ISSCC*<sup>22</sup> en 1988. Il apparaît en effet clairement que les deux architectures partagent un nombre de similitudes importantes comme par exemple l'existence d'un mode opératoire particulier appelé *PALcode* sur l'Alpha ou *Epicode* sur *Prism*.

## Cache

Voir mémoire cache.

## Dhrystone

Le Dhrystone est l'un des *benchmarks* synthétiques. Ce type de *benchmark* calcule un nombre moyen d'opérations et de données correspondant à l'exécution d'une vaste gamme de programmes. Les *benchmarks* synthétiques sont créés artificiellement de manière à correspondre à un certain profil d'exécution. Ils sont donc très éloignés de la réalité. Le *Whetstone* et le *Drhystone* sont deux *benchmarks* populaires de cette catégorie.

## DMA

### Direct Memory Access

Un transfert de type *DMA* est effectué entre un composant d'entrées/sorties et la mémoire principale sans l'intervention du CPU (d'où le terme direct).

Un contrôleur *DMA* désigne la logique qui contrôle un transfert de type *DMA*.

## Écriture attribuée – Écriture non attribuée

Lorsque l'UC veut écrire une donnée absente du cache, une des deux options suivantes est utilisée :

- écriture attribuée ou *write allocate* (ou encore *fetch on write*) : la ligne correspondante à la donnée est chargée dans le cache avant d'être modifiée ;
- écriture non attribuée ou *no write allocate* (ou *write around*) : la ligne est modifiée directement dans le niveau inférieur de la hiérarchie mémoire sans être chargée dans le cache.

## EISA Bus

### Extended Industry Standard Architecture Bus

Introduit en 1988, il constitue une extension 32 bits du bus *ISA*, spécialement adaptée aux processeurs de type 386 et 486. Ses principales caractéristiques sont :

- une totale compatibilité avec le bus *ISA* ;
- un chemin de données 32 bits ;
- un taux de transfert de 33 Mo/sec ;
- supporte plusieurs maîtres du bus ;
- supporte le partage des interruptions ;
- supporte jusqu'à 4 Go de mémoire ;
- configuration automatique des cartes.

Le bus *EISA* est totalement compatible avec son prédécesseur. Il en améliore les performances avec une fréquence de fonctionnement de 10 MHz et un bus d'adresses de 32 bits.

Les interruptions peuvent être sensibles au niveau ou au front. Quand les cartes *EISA* sont configurées sur des niveaux d'interruption, plusieurs cartes peuvent alors partager la même ligne d'interruption.

Par ailleurs l'une de ses caractéristiques est son auto-configuration. Avec un système *ISA*, cette configuration se faisait manuellement par des *switchs* ou des *jumpers*.

## Exécution « dans le désordre »

Les microprocesseurs superscalaires actuels doivent gérer un flot d'instructions important (exécution de 4 à 6 instructions par cycle) qui peut être ralenti par

<sup>22</sup> *International Solid State Circuit Conference*.

des branchements, des dépendances de données, des défauts de cache. Pour pallier d'éventuels blocages du pipeline, certains (de plus en plus) microprocesseurs mettent en œuvre un mécanisme d'exécution des instructions « dans le désordre » qui permet d'utiliser les unités fonctionnelles qui restent alors disponibles, avec des instructions indépendantes de celle qui est bloquée.

### Exécution spéculative

Dans les processeurs à pipeline, les branchements peuvent induire des cycles de latence. De plus ils représentent entre 15 et 30 % des instructions ce qui est considérable. Pour éviter dans certains cas ces cycles d'attente, une *prédiction de branchement* est mise en œuvre. Grâce à elle, inutile d'attendre le résultat de l'instruction de branchement pour charger les instructions suivantes, la direction du branchement et sa cible sont prédites. Les instructions chargées après un branchement, et avant sa résolution sont dites *spéculatives* car elles sont simplement initialisées mais peuvent être avortées en cas de mauvaise prédiction (voir *prédiction de Smith*, *prédiction de Yeh and Patt*).

### ISA Bus

#### Industry Standard Architecture Bus

Développé à l'origine pour les PC d'*IBM* (8086, bus de huit bits), il a été étendu à 16 bits en 1984 au moment de l'introduction du 80286 AT. Ses principales caractéristiques sont les suivantes :

- un maître unique du bus ;
- supporte jusqu'à 16 Mo de mémoire ;
- une largeur des données de 8 ou 16 bits ;
- 15 interruptions (sans partage) ;
- 7 canaux DMA ;
- une fréquence de fonctionnement de 8 MHz pour une bande passante de 5.3 Mo/sec.

L'arrivée des systèmes d'exploitation 32 bits tels que OS/2 et Windows NT a rendu obsolète l'utilisation de ce bus, ce dernier n'étant pas capable de gérer de telles quantités de données.

### LRU

#### Least Recently Used

Désigne l'une des stratégies de remplacement des lignes appliquées au sein d'un cache associatif. La ligne à remplacer est celle qui a été la plus anciennement référencée (« la moins récemment utilisée »). Cette stratégie donne en général de bons résultats, mais elle devient rapidement coûteuse lorsque l'associativité croît.

### MBus

#### Module Bus

Développé par SPARC (1991), le *Mbus* est un bus synchrone destiné à relier le processeur à la mémoire et

optimisé pour des communications ultra-rapides entre un faible nombre de résidents (jusqu'à 16). Il implémente un bus d'adresses et de données de 64 bits multiplexé. Les adresses ont une largeur de 36 bits, les 28 bits restants étant utilisés au cours du transfert comme information de contrôle. Il fonctionne à une fréquence de 40 MHz.

Ses implémentations peuvent être conformes à l'une ou l'autre des spécifications de « niveau-1 » et de « niveau-2 ». Le « niveau-1 », destiné aux systèmes monoprocesseurs ne supporte que deux transactions : lecture et écriture.

Le « niveau-2 » est adapté aux systèmes multiprocesseurs et ajoute quatre transactions supplémentaires ainsi que deux signaux utiles à la cohérence des caches. Le protocole de cohérence utilisé est de type *MOESI* pour les cinq états possibles (*Modified, Owned, Exclusive, Shared, Invalid*).

### Mémoire cache

Mémoire relativement petite (et donc très rapide) se trouvant en haut de la hiérarchie mémoire (au plus près de l'UC, souvent sur le même circuit intégré), et destinée à garder des données ou des instructions qui ont été récemment référencées. Ces données ou instructions ont de fortes chances d'être accédées à nouveau dans un futur proche, d'après le principe de localisation temporelle des données (au sens large). Ainsi, elles seront accédées bien plus vite que si elles se trouvaient dans la mémoire principale (mémoire de grosse capacité de stockage, mais à temps d'accès plus important).

La mémoire cache est constituée de *lignes* de plusieurs octets se trouvant côte à côte en mémoire (principe de localité spatiale<sup>23</sup>). Ces lignes peuvent être organisées de trois façons différentes :

- en correspondance directe avec la mémoire (*direct mapped*). Chaque bloc de la mémoire principale ne peut être chargé qu'à un seul endroit du cache.
- de manière totalement associative (*fully associative*). Chaque bloc de la mémoire principale peut être chargé à n'importe quel endroit du cache.
- de manière associative par ensembles (*set associative*). Le cache est divisé en ensembles d'emplacements, et chaque bloc de la mémoire principale ne peut être chargé qu'à l'intérieur d'un des ensembles. Cette organisation est un compromis entre les deux précédentes.

Plusieurs niveaux de cache peuvent être mis en œuvre. Les niveaux les plus bas dans la hiérarchie mémoire (les plus près de la mémoire principale) sont ceux

23. Si un objet est référencé, alors un objet proche de celui-ci en mémoire a de fortes chances d'être bientôt référencé à son tour.

qui ont la plus grosse capacité (ils sont aussi plus lents), et sont généralement des sur-ensembles des niveaux plus hauts dans la hiérarchie. Actuellement, jusqu'à trois niveaux de cache sont implémentés, dont parfois deux faisant physiquement partie du microprocesseur.

### Micro Channel Architecture

Développé par *IBM* pour ses machines *Personal System/2* (1987), le *Micro Channel Bus* est depuis utilisé dans tous les systèmes *IBM*. Les caractéristiques de ce bus sont sensiblement similaires à celles du bus *EISA* avec cependant des performances supérieures :

- transferts de données sur 8/16/32/64 bits ;
- bande passante de 160 Mo/sec ;
- supporte plusieurs maîtres du bus ;
- supporte le partage des interruptions (sensibles au niveau) ;
- supporte jusqu'à 4 Go de mémoire ;
- configuration automatique des cartes.

Le cœur de l'architecture *Micro Channel* est un bus d'adresses et de données séparé de 32 bits. Un protocole permet au bus d'adresse d'être multiplexé (mis en œuvre dans certains systèmes RS/6000 haut de gamme) permettant alors des transferts de données sur 64 bits.

### MIPS

MIPS est un acronyme pour *Microprocessor without Interlock Pipeline Stage*. Seuls les processeurs R3000 et R6000 furent fidèles à cette appellation. À partir du MIPS R4000, la gestion des interblocages au sein du pipeline est effectuée par matériel.

### NLU

#### Not Last Used

Stratégie de remplacement des lignes appliquée au sein d'un cache associatif. Les adresse des lignes récemment accédées sont mémorisées, la ligne à remplacer étant choisie de manière aléatoire parmi les lignes restantes.

Cette stratégie est plus fiable que la stratégie de choix aléatoire, car elle évite certains problèmes liés aux rejets trop prématurés.

### PC

#### Program Counter

Ce registre particulier contient l'adresse de la prochaine instruction à charger pour l'exécution.

### PCI Bus

#### Peripheral Connect Interface Bus

Le bus PCI a été développé par *Intel* avec le soutien de plusieurs constructeurs d'ordinateurs tels que *IBM*, *NEC* et *Compac*.

RR n° 3188

Il supporte jusqu'à dix périphériques et est légèrement plus lent que le *Vesa Local Bus* mais trois fois plus rapide que les bus *EISA* et *MCA*. Il fonctionne à une fréquence de 33 MHz et permet une bande passante de 132 Mo/sec.

Le bus PCI supporte une configuration automatique des composants. Il conserve un inventaire des ressources utilisables et attribuera une ressource libre lorsqu'une nouvelle carte PCI sera insérée.

Trois types d'espace d'adressage sont supportés :

- mémoire ;
- entrées/sorties ;
- configuration. Chaque composant PCI dispose d'un tampon de 256 octets qui contient des informations sur son type (carte réseau, contrôleur de disque, etc...), ainsi que d'autres détails tels que registres de contrôle et de statut, adresses de cartes, etc... Au démarrage du système, cette configuration sera testée pour chaque composant PCI et chacun de ces composants se verra attribuer une unique adresse de base ainsi qu'un numéro d'interruption.

Les adresses et les données sont multiplexées. Le bus PCI utilise des lectures et écritures groupées lors des transferts de données. Chacun de ces transferts a une longueur indéfinie et dure tant que le maître ou l'esclave le souhaite. Une valeur de *timer* par composant PCI permet cependant de paramétrer la durée d'occupation du bus.

### Pipeline

Le principe du pipeline est comparable à celui du travail à la chaîne : il consiste à décomposer l'exécution des instructions en phases élémentaires de longueur fixe, égale au temps de cycle (la phase la plus lente détermine le temps de cycle). Les phases du pipeline sont appelées étages, et chaque étage est géré par une unité fonctionnelle donnée. Ainsi, les exécutions de plusieurs instructions se recouvrent et le débit d'exécution est plus important.

Le pipeline type se compose de 5 étages (cas d'un processeur *RISC*) :

- **IF (Instruction Fetch)** : chargement de l'instruction.
- **ID (Instruction decode and register fetch)** : décodage de l'instruction et lecture des registres sources ; incrémentation du PC.
- **EX (EXecution and effective address calculation)** : exécution de l'instruction avec les opérandes (celles-ci dépendent du type d'instruction : référence mémoire, instruction arithmétique, branchement/saut, ...).

- **MEM (MEMory access/branch)** : accès mémoire si besoin (lecture ou écriture), ou remplacement du PC dans le cas d'un branchement.
- **WB (Write Back)** : écriture du résultat dans le registre destination.

De nombreux mécanismes doivent être mis en œuvre pour assurer le maintien d'un flot important d'instructions, car des conflits peuvent survenir. Trois types de conflits gênent la bonne marche d'un pipeline :

- les conflits de ressource: lorsque plusieurs instructions nécessitent un accès à une ressource n'ayant qu'un port d'entrée.
- les conflits de dépendance des données: une instruction nécessite l'accès au résultat d'une autre non encore terminée. Ces conflits se subdivisent en trois cas :
  - conflit de type RAW (Read After Write) ;
  - conflit de type WAR (Write After Read) ;
  - conflit de type WAW (Write After Write) ;
- les conflits de contrôle: lors de sauts ou de branchements conditionnels (changement du PC).

Pour éviter ou gérer au mieux ces conflits, chaque processeur a sa technique.

### Prédiction de Smith [29] (algorithme)

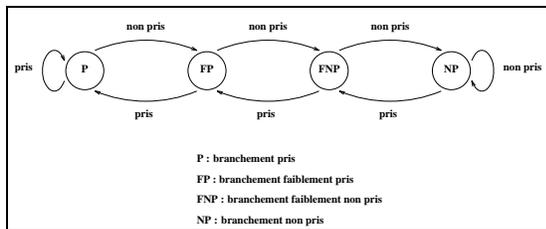


FIG. 6.2 – Prédiction de branchement à deux bits selon l'algorithme de Smith

Également appelé prédiction à deux bits, la prédiction du branchement est faite selon l'état courant associé à l'instruction de branchement. Les deux bits utilisés définissent quatre états possibles conformément à la figure ci-contre (plusieurs mises en œuvre sont possibles).

Il est à remarquer que deux mauvaises prédictions de branchement consécutives sont nécessaires pour changer de prédiction.

### Prédiction de Yeh and Patt (algorithme)

Cet algorithme est basé sur un schéma à deux niveaux. Un registre « d'historique » de largeur K (global ou pour chaque instruction de branchement) conserve les K dernières occurrences de tous les branchements.

La séquence contenue dans ce registre permet d'indexer une table (de  $2^K$  entrées) qui fournit une prédiction par l'intermédiaire d'un compteur. Une fois le résultat du branchement connu, le compteur est alors mis à jour. Diverses variations de cet algorithme sont utilisables selon que l'on utilise un registre d'historique par branchement ou une table de prédiction par branchement.

### Processeurs embarqués ou enfouis

Ces deux termes désignent un processeur utilisé pour une application spécifique dans un système et qui est invisible de l'utilisateur.

Traditionnellement, un processeur embarqué (ou microcontrôleur) est un microprocesseur qui a des capacités significatives en entrées/sorties telles que des ports parallèles et séries, des interfaces mémoires pré-décodées, des ports analogiques, etc... Les processeurs embarqués sont définis par leur application, plutôt que par le design même du processeur, et n'importe quel processeur peut être embarqué. De ce fait, un processeur embarqué est généralement un processeur dédié à l'exécution d'un simple programme et qui n'est pas reprogrammable par l'utilisateur. Les aspects les plus importants du marché des processeurs embarqués sont les contraintes de coût et de consommation.

**Protocole MESI** Protocole de cohérence de cache où chaque ligne de cache est marquée par un des quatre états : *Modified*, *Exclusive*, *Shared*, *Invalid* (d'où *MESI*). Le diagramme d'état finis suivi est celui de la figure FIG. 6.3.

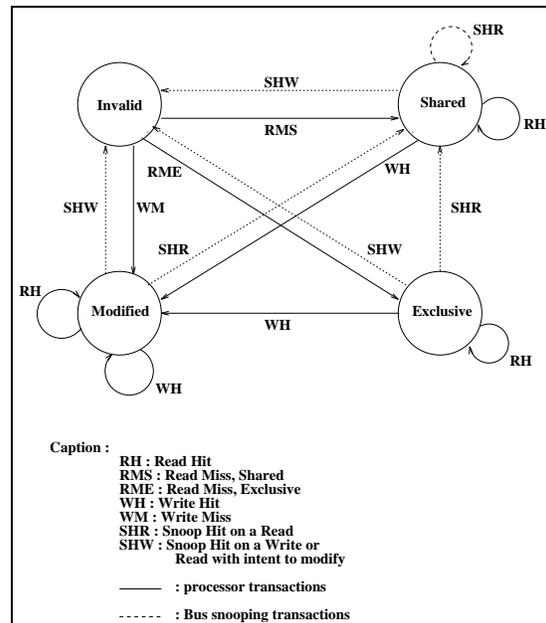


FIG. 6.3 – Diagramme d'états finis du protocole MESI.

Ce protocole est très utilisé dans des systèmes incluant des caches à écriture différée. Lorsqu'une donnée

est modifiée par un processeur, tous les caches possédant une copie de cette donnée invalident la ligne qui la contient.

### Random

Stratégie de remplacement des lignes appliquée au sein d'un cache associatif. La ligne à remplacer est choisie de manière aléatoire parmi les lignes possibles. Cette stratégie est peu coûteuse mais également peu fiable du point de vue des performances.

### Renommage de registres

Technique qui vise à éviter certaines dépendances de données entre les instructions (et leurs conséquences fâcheuses pour les performances) en mettant à disposition du CPU plus de registres physiques que l'utilisateur (en général le compilateur) ne peut en voir. De cette façon, le CPU peut exécuter dans le désordre des instructions qui *a priori* étaient séquentielles.

## RISC

### Reduced Instruction Set Computing

La conception de tout système RISC emploie deux concepts architecturaux de base :

- utiliser un jeu d'instructions simple pouvant être facilement implémenté sous la forme d'une machine pipelinée ;
- utiliser une technologie de compilateur sophistiquée permettant de générer un code optimal afin de tirer partie de cette machine pipeline.

Les critères plus classiques concernent :

- des instructions de taille constante, naturellement alignées sur des frontières de mot et réalisant des opérations dites « simples » ;
- un faible nombre de formats d'instructions ;
- un grand nombre de registres ;
- une architecture de type *load/store* : toute opération est exécutée entre les registres et écrit son résultat dans un registre. Aucune opération d'accès mémoire autre que des instructions de « lecture » et « d'écriture » n'est permise par le jeu d'instructions.

## SCSI

### Small Computer Systems Interface Bus

Supportant un vaste ensemble de médias : CDROM, support optique, scanner, ..., il est principalement utilisé par les périphériques de sauvegarde. Chaque adaptateur peut supporter jusqu'à sept périphériques, chacun possédant ses propres paramètres en interne (adresse, contrôleur). Sa bande passante est relativement lente (4 Mo/sec).

RR n° 3188

### Superpipeline

Un superpipeline est un pipeline où les étages typiques (IF, ID, EX, MEM, WB) sont décomposés en d'autres étages. Par exemple on trouve dans certains microprocesseurs l'étage ID (décodage) divisé en ID1 et ID2. Cette notion a été introduite par *MIPS* avec le R4000.

### Superscalaire

Une architecture superscalaire est composée d'un monoprocesseur comportant un pipeline qui peut exécuter plusieurs opérations scalaires en un cycle. Pour ce faire certaines unités fonctionnelles sont dupliquées.

Les processeurs actuels exécutent jusqu'à 4 instructions par cycle, voire 5 pour le MIPS R10000.

### TLB

#### Translation-Lookaside Buffer

Les mécanismes de traduction d'adresses virtuelles en adresses physiques sont généralement très lourds (sélection de la table des processus, adressage de la table des pages du processus à partir de l'adresse virtuelle, accès au numéro de page physique). Aussi, pour atteindre des performances correctes, un cache de traduction d'adresses, généralement appelé le *TLB*, conserve la traduction des pages récemment accédées. Suivant le même principe que les caches, le TLB est consulté à chaque fois que le microprocesseur produit une adresse virtuelle. Ainsi, si la donnée ne se trouve pas dans le cache, l'adresse physique est immédiatement disponible pour adresser le sous-système mémoire.

### VLIW

#### Very Long Instruction Word

Une architecture dite VLIW traite des mots d'instructions contenant plusieurs opérations. Ces mots sont alors longs, voire très longs, d'où le nom qui leur est donné.



---

Unit e de recherche INRIA Lorraine, Technop le de Nancy-Brabois,  
Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS L S NANCY  
Unit e de recherche INRIA Rennes, Irista, Campus universitaire de  
Beaulieu, 35042 RENNES Cedex  
Unit e de recherche INRIA Rh ne-Alpes, 655, avenue de l'Europe, 38330  
MONTBONNOT ST MARTIN  
Unit e de recherche INRIA Rocquencourt, Domaine de Voluceau,  
Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unit e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles,  
BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

 diteur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105,  
78153 LE CHESNAY Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399