Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86

André Seznec, Yann Mével

▶ **To cite this version:**

André Seznec, Yann Mével. Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86. [Rapport de recherche] RR-2746, INRIA. 1995. inria-00073946

HAL Id: inria-00073946
https://inria.hal.science/inria-00073946

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.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86

André Seznec, Yann Mével

N° 2746

Décembre 1995

PROGRAMME 1

Rapport de recherche

ISSN 0249-6399

INRIA
RENNES

Évolution des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86

André Seznec, Yann Mével

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet CAPS

Rapport de recherche n° 2746 — Décembre 1995 — 121 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 circuit intégré ou au niveau de l'architecture, du logiciel ou de la fréquence d'horloge.

Ce rapport présente une vue de cette évolution à travers l'étude des gammes de processeurs MIPS, DEC Alpha, PowerPC, SPARC et xxx86. Il présente les principales caractéristiques de ces architectures et introduit les principaux intervenants : fondeurs, concepteurs de systèmes, etc...

Mots-clé : Veille technologique, architecture de microprocesseurs, MIPS, DEC, PowerPC, SPARC, xxx86.

(Abstract: pto)

Ce travail a été partiellement soutenu par la DRET (convention DRET-INRIA n° 94-2595 A/DRET), ainsi que par le GDR Architecture des Machines Nouvelles.

Unité de recherche INRIA Rennes

Evolution of the microprocessors MIPS, DEC Alpha, PowerPC, SPARC et xxx86

Abstract: In this report, we try to present in a synthetic form the evolution of different families of microprocessors. MIPS, DEC Alpha, PowerPC, SPARC and xxx86 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, xxx86.

Table des matières

Introduction **7**

1 Les microprocesseurs MIPS **9**
1.1 Les processeurs MIPS R3000, R4000 et R8000 9
1.1.1 Jeu d'instructions 10
1.1.2 Les pipelines 11
1.1.3 Les unités arithmétiques 14
1.1.4 Hiérarchie mémoire 15
1.1.5 Support des systèmes d'exploitation 17
1.1.6 Support multiprocesseur 19
1.2 Le développement de systèmes à base de processeurs MIPS 20
1.2.1 Processeurs dérivés du MIPS R3000 21
1.2.2 Processeurs dérivés du MIPS R4000 25
1.3 Perspectives 28
1.3.1 Le processeur MIPS R10000 28

2 Les microprocesseurs Alpha **31**
2.1 L'architecture Alpha 32
2.1.1 Caractéristiques générales 32
2.1.2 Le jeu d'instructions 32
2.2 Les processeurs Alpha AXP 33
2.2.1 Les processeurs 21064 et 21064A 34
2.2.2 Les processeurs DEC 21066 et 21068 39
2.2.3 Le processeur 21164 40

3 Les architectures POWER et PowerPC **43**
3.1 L'architecture POWER 43
3.1.1 Le jeu d'instructions POWER 44
3.1.2 L'architecture POWER 45
3.2 L'architecture PowerPC 48
3.2.1 Le jeu d'instructions PowerPC 49
3.2.2 Le PowerPC 601 50
3.2.3 Les processeurs PowerPC 603, 602 et 604 56
3.2.4 Les microcontrôleurs d'architecture PowerPC 59
3.2.5 Le PowerPC 620 60

4 Les microprocesseurs SPARC **63**
4.1 La norme SPARC-V9 . . . 63
4.1.1 Le jeu d'instructions . . . 64
4.1.2 Caractéristiques de mises en œuvre de l'architecture SPARC-V9 . . . 68
4.2 Le Sun SPARC . . . 71
4.3 Le SuperSPARC . . . 72
4.3.1 Le SuperSPARC+ . . . 74
4.3.2 Le SuperSPARC 2 . . . 74
4.4 Le MicroSPARC . . . 75
4.5 L'HyperSPARC . . . 77
4.6 L'UltraSPARC . . . 77
4.6.1 Implémentation matérielle de l'UltraSPARC . . . 78
4.6.2 Les ajouts à la norme SPARC-V9 . . . 82
4.7 Le Sparc64 . . . 82

5 Les microprocesseurs xxx86 **89**
5.1 L'architecture CISC d'INTEL . . . 90
5.2 Caractéristiques de l'architecture Intel . . . 91
5.2.1 Modes de fonctionnement . . . 91
5.2.2 Organisation de la mémoire . . . 91
5.2.3 Les registres . . . 93
5.3 Le jeu d'instructions CISC Intel . . . 95
5.3.1 Types de données . . . 95
5.3.2 Les modes d'adressage . . . 95
5.3.3 Format des instructions . . . 96
5.3.4 Les instructions . . . 98
5.4 Le processeur Pentium . . . 101
5.5 Les compatibles Pentium . . . 106
5.5.1 Le compatible Pentium de Cyrix : le M1 . . . 106
5.5.2 Le compatible Pentium d'AMD : le K5 . . . 109
5.6 Le processeur Intel P6 . . . 112

Remerciements **115**

Bibliographie **116**

Glossaire **118**

Table des figures

3.1 Relation entre les jeux d'instructions POWER et PowerPC 50

4.1 Exemple de fenêtres de registres sur l'architecture SPARC-V9 69
4.2 Mécanisme de traduction des pages selon la norme SPARC-V8 71
4.3 Pipeline d'exécution de l'UltraSPARC 78
4.4 Illustration du mécanisme de suivi des instructions sur le Sparc64 84
4.5 Pipeline du Sparc64 86

5.1 Mécanisme de traduction d'adresse du Pentium 93
5.2 Registres du Pentium 94
5.3 Calcul de l'adresse linéaire à partir d'un adressage basé indexé 96
5.4 Format des instructions du Pentium 97
5.5 Pipelines du processeur Cyrix M1 107
5.6 Pipeline d'exécution du P6 113

Liste des tableaux

1.1 Évolution du jeu d'instructions MIPS 12
1.2 Caractéristiques des principaux processeurs embarqués à base d'architecture MIPS R3000 (1992) 23

4.1 Définition du format des données selon la norme SPARC-V9 (en bits) 64
4.2 Valeur des déplacements relatifs au PC selon la norme SPARC-V9 68
4.3 Codage du numéro de registre flottant selon la norme SPARC-V9 70

Introduction

Les études comparatives sur les microprocesseurs, conduites jusqu'ici au sein de l'équipe CAPS[1] ([1], [2], [3], [4]) opposent deux ou trois processeurs d'architectures différentes mais de même génération. Ces études sont menées dans le cadre d'une activité de veille technologique et concernent des processeurs récents, c'est à dire juste disponibles ou bien même simplement annoncés.

Dans cette étude, au lieu de présenter des processeurs à performances comparables, nous étudions au contraire un ensemble de processeurs conformes à une architecture donnée, aux performances variables et de générations distinctes.

Au lieu de se focaliser sur un aspect du marché (les processeurs hautes performances), nous mesurons au contraire la pénétration d'une architecture (c'est à dire d'un ensemble de processeurs compatibles binairement) de manière plus générale. Effectivement, l'implémentation d'une architecture se « décline » pour couvrir l'ensemble du marché, du microcontrôleur visant l'application enfouie largement distribuée à la station de travail haut de gamme.

Cette étude n'est pas une étude exhaustive des processeurs du marché. Nous avons essayé, à partir de divers critères, de sélectionner les processeurs les plus représentatifs d'une architecture donnée que ce soit au niveau du succès de son implémentation qu'au niveau de l'évolution de l'architecture à travers diverses mises en œuvre.

En outre, cette étude vise à introduire les principaux acteurs (concepteurs, fondeurs, intégrateurs de systèmes, ...) qui gravitent autour de ces architectures et nous permet de prendre ainsi connaissance des différentes alliances qui régissent le marché des microprocesseurs.

Aussi, bien que cette étude s'adresse à des lecteurs « avertis » (nous ne revenons par sur les notions introduites dans [1], [2], [3], [4]), elle se veut être un ouvrage de vulgarisation et d'information générale sur les microprocesseurs à travers :

- la description d'une architecture ;
- son évolution ;
- ses diverses mises en œuvre ;
- ses principaux acteurs industriels.

Le choix des architectures étudiées est basé sur la disponibilité de l'information et nos connaissances de ces gammes. Cette étude devrait être complétée régulièrement et mise à jour dans les années à venir.

1. Compilation, Architectures Parallèles et Systèmes.

Les informations présentées dans ce document ont été principalement extraites de la revue *Microprocessor Report - The Insider's Guide To Microprocessor Hardware*, et des divers sites *World Wide Web* relatifs aux architectures étudiées.

Avertissement: les divers chapitres de cette étude peuvent être abordés indépendamment.

Chapitre 1

Les microprocesseurs MIPS

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

Le succès des processeurs MIPS est également du au marché des systèmes embarqués (*set-top boxes*, consoles de jeux, *Personal Digital Assistant*). La croissance de ce secteur et les enjeux économiques qu'il représente laissent augurer un avenir florissant pour MIPS.

Nous présenterons dans un premier temps les processeurs qui ont fait le succès de MIPS dans le domaine des processeurs RISC, à savoir les processeurs MIPS R3000, R4000 et R8000. Dans la mesure où ces processeurs ont déjà été le sujet de nos précédentes études (voir [1], [2], [4]), nous n'entrerons pas dans les détails mais nous présenterons les principales différences d'une génération de processeur à l'autre. Puis nous étudierons quelques uns des processeurs développés à partir des architectures MIPS R3000 et R4000. Cette partie nous permettra d'introduire la stratégie commerciale de MIPS à travers ses partenaires fondeurs. En effet, si MIPS a essentiellement une activité de développement de processeurs, la fonderie est assurée par des partenaires extérieurs. L'ensemble de ces fondeurs a d'autre part une activité d'innovation qui contribue au développement de systèmes à base de processeurs MIPS et qui accroît ainsi la pénétration de cette architecture sur le marché des systèmes embarqués.

1.1 Les processeurs MIPS R3000, R4000 et R8000

Annoncé en 1988 par la société *MIPS Computer Systems*, le MIPS R3000 fut le premier processeur à vocation industrielle. Il succède au MIPS R2000, dérivé des travaux de l'Université de Stanford (1985). L'une des principales caractéristiques de ce processeur est sa simplicité. En effet, la force des processeurs MIPS réside dans la relation très étroite entre l'architecture du processeur et le compilateur associé[1].

En 1991, Le MIPS R6000 fut commercialisé à la fréquence de 66 MHz. Développé par *Bipolar Integrated Technology*, ce processeur, reprenait pour l'essentielle 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

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

de traduction d'adresses séparée de l'unité entière et implémentée sur le cache secondaire, etc... Cependant, ce processeur réalisé en technologie *ECL*, ne connut pas le succès escompté. 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 processeur le plus connu de cette technologie.

Le MIPS R4000 est la troisième génération de processeurs MIPS (1992). Il est le premier à mettre en œuvre une architecture 64 bits qui reste cependant compatible avec les générations MIPS précédentes (un mode 32 bits assure la compatibilité). Afin d'offrir une cadence de séquencement très élevée (dès sa sortie, le R4000 est annoncé à la fréquence de 100 MHz), la technique du superpipeline a été utilisée.

Destiné au marché des stations de travail, ce processeur fut initialement prévu en trois versions : R4000PC (avec cache primaire seulement), R4000SC (avec cache secondaire) et R4000MC (version destinée aux systèmes multiprocesseurs).

Le MIPS R8000 est fondu par *Toshiba* (1994). Il vise clairement le marché des applications scientifiques et techniques (stations de travail graphiques 3-D), les serveurs de base de données, ainsi que le domaine des multiprocesseurs.

Ce processeur 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 processeur met en œuvre des mécanismes originaux afin d'accroître ses performances. Commercialisé initialement à la fréquence de 75 MHz, MIPS a annoncé tout dernièrement une version à 90 MHz.

Nous décrivons dans les chapitres suivants les principales caractéristiques de ces processeurs en comparant leurs jeux d'instructions, leurs pipelines, la structure des caches et les mécanismes de gestion de la mémoire.

1.1.1 Jeu d'instructions

La définition originale du jeu d'instructions MIPS a connu trois mises à jour, chacune étant un sur-ensemble du jeu d'instructions précédent. Nous présentons dans une première partie les concepts généraux de ce jeu d'instructions avant de détailler son évolution.

Le jeu d'instructions de base

Toutes les instructions sont codées sur 32 bits. Trois formats d'instructions sont couramment utilisés : immédiat (*I-type*), saut (*J-type*) et registre (*R-type*) (voir [1], [2], [4] pour plus de détails).

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*). Ces instructions sont de type *I-type*.

- Les instructions arithmétiques. Ces instructions concernent l'ensemble des calculs, les opérations logiques et de décalages. Elles sont de type *R-type* (les deux opérandes sont des registres) ou *I-type* (l'un des opérandes est alors une valeur immédiate de 16 bits).

- Les instructions de saut et de branchement. Ces instructions modifient le contrôle de flot du programme. Les instructions de saut peuvent être de type *J-type* (l'adresse est alors sur 26 bits) ou *R-type* (l'adresse est alors contenue dans un registre et a donc une largeur de 32 bits). Les instructions de branchement permettent des déplacements de 16 bits par rapport au compteur de programme (type *I-type*).

- 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 intégré sur le même composant que l'unité entière et gère le système de mémoire virtuelle, les exceptions ainsi que les transitions entre les modes *Noyau*, *Superviseur* (R4000 seulement) 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. Elles sont toujours de type *R-type*.

Évolution du jeu d'instructions MIPS

Le jeu d'instructions MIPS a connu trois mises à jour. Le R2000 implémente l'architecture originale MIPS I. Le R6000 et le R3000 mettent en œuvre les extensions apportées par le jeu d'instructions MIPS II, alors que le R4000 est le premier des processeurs MIPS à utiliser l'architecture MIPS III. La dernière mise à jour de ce jeu d'instructions, définie par l'architecture MIPS IV, est implémentée sur le MIPS R8000 et sera également utilisée pour le MIPS R10000. Le tableau 1.1 reprend les principales évolutions d'un jeu d'instructions à l'autre. Pour plus de détails sur le jeu d'instructions MIPS IV, le lecteur consultera [5].

1.1.2 Les pipelines

Pipeline entier

Les trois processeurs 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 processeur RISC traditionnel (chargement de l'instruction, décodage, exécution ou calcul de l'adresse, accès à la mémoire, mise à jour du banc de registres). La gestion des interblocages au sein du pipeline est effectuée par logiciel.

2. Ces instructions sont exécutées avec un retard de une instruction, l'instruction suivant le saut étant toujours exécutée avant le branchement.

	MIPS I	MIPS II	MIPS III	MIPS IV
Lecture/écriture		Transfert coprocesseur 64 bits. Mise à jour atomique de mots.	Extension 64 bits	Adressage *registre + registre* pour l'unité flottante.
Addition/soustraction	32 bits		64 bits	64 bits
Décalage	–	32 bits	64 bits	64 bits
Multiplication/division	32 bits		64 bits	64 bits
Instruction de multiplication-addition flottante	–	–	–	oui
Saut et branchement [2]	–	*Branch on likely*: l'instruction du *branch delay slot* est exécutée exclusivement si le branchement est pris.		Extension des codes conditions flottants (8).
Instructions d'exceptions	Appel au système et point d'arrêt.	Déclenchement d'une exception sur la comparaison de deux registres ou d'une valeur immédiate.		
Instructions de sérialisation		SYNC : synchronisation des accès à la mémoire partagée.		
Conditional Move	–	–	–	oui
Instruction de préchargement	–	–	–	oui

TAB. 1.1 - *Évolution du jeu d'instructions MIPS*

Le MIPS R4000 est le premier des processeurs RISC à mettre en œuvre la technique du super-pipeline. Les étages critiques ont été divisés en deux, voire en trois, pour permettre à ce processeur d'atteindre des fréquences élevées. Le pipeline a au total une profondeur de huit étages.

Contrairement au processeur MIPS R3000 (où la gestion des interblocages est effectuée par logiciel), ce processeur met en œuvre une gestion matérielle. L'acronyme *Microprocessor without Interlock Pipeline Stage* n'est donc plus respecté. Ceci s'explique par le besoin de compatibilité binaire entre des versions successives : gérer les interblocages uniquement par voie logicielle impose des contraintes sévères sur la structure des pipelines des futures versions. Par ailleurs le volume du code d'une application devient critique pour les performances : le cache est petit et le coût d'un défaut devient relativement élevé devant la fréquence de séquencement de ces processeurs de plus en plus rapides.

Le pipeline entier du MIPS R8000 diffère largement des structures de pipeline jusqu'ici employées par MIPS. Premier processeur 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, quatre instructions flottantes.

D'une profondeur de cinq étages, le pipeline du R8000 offre la particularité d'inverser l'étage 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 processeur met en œuvre un découplage au niveau de l'exécution des instructions entières et flottantes. Alors que les opérations entières et les générations d'adresse 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.

Par ailleurs, la dernière remarque que nous ferons sur la structure du pipeline entier concerne la mise en œuvre d'une prédiction de branchement. Cette prédiction est effectuée par le biais d'un cache d'adresse de branchement de 1024 entrées accédé au même moment que le cache d'instructions.

Pipeline flottant

Sur les trois microprocesseurs étudiés, les pipelines flottants sont sensiblement identiques aux pipelines entiers, les étages de chargement 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.

Le pipeline flottant du R8000 est totalement découplé du pipeline entier. Seuls les étages de chargement et de décodage sont communs. Comme on l'a dit précédemment, jusqu'à quatre instructions flottantes peuvent être émises par l'unité entière à chaque cycle vers la *Floating Point Queue*. Deux instructions au maximum, sont émises par cycle, dans l'ordre, à partir de la *FPQ* où elles ont attendues la disponibilité de leurs opérandes. Cette file fonctionne en synergie avec deux files de chargement de données (d'une capacité de 32 entrées) qui permettent d'accéder aux données

requises par les instructions tout en masquant la latence du cache secondaire externe (pour plus de détails se reporter à [4]).

1.1.3 Les unités arithmétiques

Unité arithmétique entière

Sur le processeur 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 (le registre *LO* contient le quotient et le registre *HI* le reste). 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 du fait de l'architecture superscalaire. Le banc de registres dispose toujours de 32 registres de 64 bits, mais les deux ports de lecture et le port d'écriture de l'architecture scalaire du MIPS R4000 ont été étendus à neuf ports en lecture et quatre ports en écriture afin de prendre en compte l'ensemble des conflits de contrôle d'accès au banc de registres.

Unité arithmétique flottante

L'architecture MIPS définit l'unité flottante comme le *coprocesseur 1* (*CP1*), qu'elle soit intégrée ou non sur la même puce que l'unité entière. Cette unité est responsable de l'ensemble des opérations arithmétiques flottantes. Elle est conforme aux exigences définies par la norme *ANSI/IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic.*

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*). Ces registres sont utilisés lors d'opérations sur des valeurs en double précision. Ce 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 processeurs embarqués, l'unité flottante et l'unité entière sont alors intégrées 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 processeur 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 à 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.

Le MIPS R8000 vise des performances flottantes élevées. L'architecture MIPS IV inclut à cet effet quatre nouvelles instructions de multiplication-addition flottantes à trois opérandes ($\pm(A \times B \pm C) \rightarrow D$). Ces instructions interviennent dans un grand nombre d'algorithmes scientifiques lors des calculs de sommes de produits. Elles devraient favoriser l'exécution des codes numériques vectorisables. De-même que l'opération *Fmadd* du Power1, l'opérateur n'est pas conforme à la norme IEEE 754 car il n'effectue pas d'arrondi entre la multiplication et l'addition. Ceci permet de diminuer la latence de l'opération et en augmente la précision.

On peut remarquer la différence d'implémentation entre les processeurs R3000 et R8000 qui génèrent une interruption externe vers l'unité entière alors que le R4000 utilise une exception.

1.1.4 Hiérarchie mémoire

L'architecture MIPS définit des processeurs 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 peut être éventuellement ajouté.

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 et à correspondances directes, physiquement adressés et testés. Leur capacité est variable de 4 à 64 Ko chacun. Il est à noter que le R3000 permet au moment de l'initialisation du système le choix de la politique d'écriture lors des défauts sur le cache en allouée ou non-allouée.

Sur le MIPS R4000, les deux caches de premier niveau (instructions et données) sont internes. Ils ont tous deux une taille de 8 Ko et sont à correspondances directes. Le cache de données met en œuvre une politique d'écriture différée. 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 au stockage des données flottantes.

Le cache d'instructions a une capacité de 16 Ko et est à correspondance directe. Il est structuré en lignes de quatre instructions de manière à alimenter en un cycle l'unité de chargement et de décodage des instructions. Ce cache est adressé virtuellement. Ce choix permet d'utiliser directement l'adresse venant du programme et simplifie le mécanisme de prédiction de branchement mis en œuvre sur ce processeur.

Comme on l'a dit précédemment, le cache de données n'est accédé qu'à travers les instructions de lecture/écriture entières. Ce cache a une capacité de 16 Ko 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 *direct-mapped*, 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 R8000 est le premier des processeurs (après le MIPS R6000) à introduire un identificateur de processus dans la ligne d'étiquettes du cache. Cet identificateur facilite la gestion des caches en évitant l'invalidation de plusieurs lignes de cache lors de changement de contexte. Codées sur huit bits (soit 256 processus actifs simultanément), deux valeurs distinctes d'identificateurs de processus sont utilisées pour les caches d'instructions et de données.

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

Deuxième niveau de cache

Les concepteurs du R3000 n'ont pas prévu l'ajout d'un deuxième niveau de cache sur ce processeur. C'est seulement à partir du MIPS R6000, que l'on a vu apparaître ce deuxième niveau de mémoire (cache secondaire externe d'instructions et de données).

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 Ko et 4 Mo. Ce cache est, quelle que soit sa configuration, à correspondance directe avec la mémoire principale, et utilise une politique d'écriture de type *write back* pour la mise à jour des données.

Il est relié au processeur 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 (*Error Correction Code*).

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 requiert en général un ensemble de travail (ou *working set*) volumineux. Aussi, la structure de son cache secondaire privilégie une large bande passante (1.2 Go) et une grande capacité (de 4 à 16 Mo) plutôt que le temps d'accès (accès pipeliné sur cinq cycles). Il est associatif par ensemble de 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 sur les performances du système. C'est pourquoi les concepteurs ont choisi de faire appel à une technologie récente de mémoires synchrones (c'est à dire utilisant une horloge) de 12 ns de temps de cycle, organisées en deux bancs entrelacés. À cause de son importante bande passante, ce cache est désigné par MIPS sous le nom de *Streaming Cache*.

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 identique 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.

Taille des pages

Le MIPS R3000 manipule des pages de taille constante de 4 Ko. Sur le MIPS R6000, la taille des pages a été étendue à 16 Ko. Il a fallu attendre le R4000 pour avoir des pages de taille variable de 4 Ko à 16 Mo (4, 16, 64, 256 Ko, 1 Mo ou 4 Mo). Chaque processus peut ainsi avoir une taille de page spécifique ce qui permet d'adapter la gestion de la mémoire au type de l'application.

Ce schéma a été repris sur le MIPS R8000. Par ailleurs, l'introduction d'un identificateur de processus distinct pour les instructions et les données permet d'affiner la gestion de la mémoire en allouant des tailles de pages différentes entre les instructions et les données.

Espace virtuel

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

L'espace physique du R4000 est de 64 Go (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 Go dans le premier cas et de 1 To dans le second (on ne tient compte que de 40 bits de l'adresse virtuelle).

Sur le MIPS R8000, les espaces d'adressages 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, les calculs d'adresse s'effectuant toujours dans l'un et l'autre cas sur 64 bits.

Cache de traduction d'adresses

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 (*Translation Lookaside Buffer*), conserve la traduction des pages récemment accédées. Suivant le même principe que les caches, le TLB est consulté à chaque fois que le processeur 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.

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 un algorithme aléatoire.

Le MIPS R4000 implémente un TLB à 48 entrées également commun aux instructions et aux données. Chacune de ces entrées est composée d'une adresse paire et impaire contiguës en mémoire. 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. Cependant, une possibilité de verrouillage de certaines entrées permet de conserver dans les TLB la traduction d'adresses de sections de codes ou de données fréquemment référencées.

Le R4000 possède également un TLB de deux entrées dédié spécifiquement 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 TLB instruction, 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. En effet, 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 [6]. Le regarnissage du TLB d'instructions à partir du TLB général ne coûte que quelques cycles. On peut remarquer par ailleurs que 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 ensemble à trois voies. L'indexage d'un TLB associatif par ensemble, où peuvent être rangés des descripteurs de page de différentes tailles, est un problème délicat (choix de l'index). Il est ici résolu simplement par l'utilisation pour chaque ASID [3] d'une seule taille de page; cette taille fait partie du contexte du processus. On rappelle par ailleurs que ce processeur met en œuvre des identificateurs de processus distincts pour les instructions et les données.

L'algorithme de remplacement du TLB est basé sur un algorithme de remplacement aléatoire. On notera le changement par rapport au R4000 où le remplacement est géré par logiciel. De la même manière que sur le R4000, certaines entrées du premier banc du TLB peuvent être verrouillées afin d'éviter que certaines pages fréquemment accédées ne soient vidées.

> Il est à remarquer que sur les processeurs R4000 et R8000, les lignes de cache d'instructions et de données sont indexées virtuellement mais sont vérifiées au moyen d'une étiquette physique contenant l'adresse physique correspondante ainsi que des informations sur l'état de la ligne. Ceci permet d'effectuer en parallèle l'accès au cache et la traduction d'adresse virtuelle en adresse physique. La comparaison des étiquettes s'effectuant au cycle suivant. Ceci permet de gagner un cycle en cas de succès et d'utilisation immédiate de la donnée.

Protection

Les processeurs disposent de plusieurs modes de fonctionnement. Ces modes protègent en écriture ou en lecture certaines zones mémoires et permettent à différentes tâches de s'exécuter sans interférences. Sur les processeurs MIPS, la protection est réalisée à chaque accès à la mémoire au

3. *Address Space Identifier*: identificateur de processus.

cours de la traduction d'adresse et s'effectue au niveau de la page.

Le MIPS R3000 implémente les deux modes de fonctionnement communément utilisés : le *mode noyau* et le *mode utilisateur*. La distinction entre ces modes s'effectue par les bits de poids fort des adresses. Comme on l'a dit, la mémoire virtuelle est partitionnée en deux zones de 2 Go, la zone d'adresse basse 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 *mode 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* lorsqu'une exception est détectée et y reste jusqu'à ce que le contexte dans lequel l'exception a été remarquée 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 soutenir 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 Go à 1 To. 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.
Un processus lancé par le système d'exploitation en *mode utilisateur* ne pourra générer des adresses que dans la zone mémoire allouée à ce mode, garantissant ainsi une protection du système.

Le mode est spécifié par les deux bits de poids fort de l'adresse virtuelle (bits [63 : 62]) nommés *bits de régions*.

1.1.6 Support multiprocesseur

Le degré de performance atteint par les microprocesseurs étudiés les rendent attrayant pour les systèmes multiprocesseurs. Les système multiprocesseurs envisagés sont des systèmes à mémoire partagée et à bus unique. Dans de tels systèmes, il est nécessaire de disposer de protocoles de cohérence de cache qui assurent l'exclusion mutuelle et la mise à jour des différents caches, afin que chaque processeur puisse lire des données cohérentes.

Le MIPS R4000 est réellement le premier des processeurs MIPS à inclure de telles fonctionnalités. Il permet les protocoles suivant :

- l'invalidation en écriture (*write invalidate*) : lorsqu'une donnée est modifiée par un processeur, 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*.

– la diffusion en écriture (*write update*) : lorsqu'une donnée est modifiée par un processeur, 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 par plusieurs bits dans le TLB. Chaque page possède un attribut, choisi parmi cinq, 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, le protocole étant choisi page par page.

Par ailleurs, le R4000 intègre des instructions spécifiques utilisées lors des accès à la mémoire partagée. Ces instructions 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'évènements. 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 des systèmes incluant des caches à écriture différée en mémoire.

Par ailleurs, l'un des éléments importants d'un système bâti autour du MIPS R8000, est *le contrôleur de cache*. Ce composant n'est pas proposé par MIPS, mais ses fonctionnalités sont spécifiées. Il est chargé de l'interface entre le MIPS R8000, la mémoire principale et le reste du système. 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 processeurs) ;

– gestion de la cohérence des deux niveaux de caches (cache secondaire externe et cache interne) ;

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

Sur les trois processeurs, la duplication des RAM d'étiquettes des caches ainsi que l'ajout d'un *contrôleur* réalisant les opérations spécifiques à un système multiprocesseur (espionnage de bus par exemple) sont nécessaires pour garantir des performances acceptables.

Les systèmes multiprocesseurs à base de R8000, commercialisés par SGI, accueillent jusqu'à 18 processeurs et peuvent atteindre des performances crêtes de 5.4 Gflops (système *Power Challenge XL*).

1.2 Le développement de systèmes à base de processeurs MIPS

Après avoir étudié les processeurs qui ont imposé MIPS sur le marché des RISC, nous abordons dans cette partie quelques uns des processeurs dérivés de ces architectures.

Les principaux systèmes sont bâtis autour des architectures R3000 et R4000, le R8000 étant trop récent pour avoir été déjà sujet à des modifications.

Initialement utilisée dans des stations de travail, des serveurs et des systèmes tolérants aux fautes, la famille de processeurs MIPS couvre également un large spectre d'applications. Principal marché : les applications embarquées. MIPS met en pratique une stratégie commerciale par le biais

de licences qui permettent à ses partenaires de développer des produits destinés à des applications enfouies. Voici une liste non exhaustive de fondeurs de processeurs d'architecture MIPS :

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

Il va de soi que ce chapitre ne constitue pas une liste cpmplète de l'ensemble des processeurs développés à partir de cette architecture, mais il permet de prendre conscience de la large utilisation des systèmes à base de processeurs MIPS dans un grand nombre d'applications enfouies.

1.2.1 Processeurs dérivés du MIPS R3000

L'architecture MIPS dispose d'atouts la rendant intéressante pour le marché des systèmes embarqués : de grands caches d'instructions et de données, des unités flottantes optionnelles et des tampons d'écritures. Par ailleurs, l'unité de gestion de la mémoire facilite la protection sur beaucoup de systèmes d'exploitation temps réels.

Le R3081

Bien que ce processeur soit relativement ancien, il connaît encore de nouvelles mises à jour qui lui permettent d'être encore présent sur le marché des systèmes embarqués.

Annoncé en janvier 1992 par *IDT*, ce processeur fait suite aux modèles R3051 et R3052 d'entrée de gamme commercialisés en 1991. Ce nouveau composant vise deux marchés distincts : les applications embarquées de hautes performances et les systèmes *ACE/ARC*[4] d'entrée de gamme. Il inclut une unité flottante, de grands caches et des caractéristiques qui facilitent son intégration dans les systèmes.

Configuration des caches. Le R3081 a un cache d'instructions de 16 Ko et un cache de données de 4 Ko. Les caches peuvent également être configurés de manière à accueillir chacun 8 Ko d'instructions et de données ce qui permet un partitionnement efficace en fonction du type d'applications. (Cette configuration peut également être changée dynamiquement, mais nécessite bien évidemment que les caches soient vidés).

Le chargement du cache de données est configurable de 1 à 4 mots et une parité a également été rajoutée. De même que le R3051, ce processeur inclut des tampons de lecture/écriture de 4 mots chacun.

4. Advanced Computing Environment/Advanced RISC Computing.

Caractéristiques technologiques. Ce processeur est proposé à la fréquence 20, 25, 33 et 40 MHz et une version à 50 MHz est également prévue. La fréquence interne d'entrée peut être configurée de manière à faciliter le remplacement du R3051 directement par ce composant.

Par ailleurs, comme la plupart des processeurs destinés au marché embarqué, il inclut des fonctionnalités qui lui permettent de réduire sa consommation (mode de *réduction de puissance* sous lequel la fréquence interne est divisée par 16. La consommation passe alors de 900 mA à 250 mA).

L'interface bus a été modifiée pour permettre à un bus externe maître d'invalider des lignes de cache sélectionnées. Cependant, la logique d'espionnage de bus (*bus snooping*) reste limitée (le composant externe doit préciser les lignes de caches à invalider).

Le tableau 1.2 reprend les principales caractéristiques des processeurs embarqués de la catégorie du R3081. Le concurrent direct de ce processeur est le *Performance Semiconductor PIPER* (1991) qui est le seul composant à inclure également une unité flottante.

R3071 et R3071E

Annoncés par *IDT* en juillet 1994, ces deux produits issus de la famille des processeurs R3000 sont typiquement des versions du R3081 avec les performances flottantes en moins. Ils sont spécialement destinés aux marchés des réseaux et autres applications nécessitant de grands caches (20 Ko au total) avec des performances flottantes moindres. Le *E* indique une version avec unité de gestion mémoire (MMU). Ces deux composants sont proposés aux fréquences de 33, 40 et 50 MHz, les prix s'échelonnant de \$75 à \$118[5].

Le R3900

Annoncé en février 1995, ce processeur constitue une interprétation de l'architecture MIPS par *Toshiba*. Appelé le R3900, pour indiquer la proximité de ses performances avec celles du R4000, il vise directement le marché des applications embarquées hautes performances. Ce processeur est conçu sur le jeux d'instructions MIPS I et sur les registres 32 bits du R3000 mais il en améliore les performances grâce à une gestion matérielle des interblocages (*register scoreboarding*) et un multiplieur rapide qui effectue une multiplication-addition (32x32) en 2 cycles.

Le jeu d'instructions du R3900 est en fait un hybride entre les définitions des jeux d'instructions MIPS I et MIPS II. Il implémente toutes les instructions standard du jeu d'instructions MIPS I à l'exception des opérations de lecture/écriture du coprocesseur, plus des améliorations vues pour la première fois sur le R4000, plus quelques ajouts effectués par *Toshiba*.

Ce processeur supporte une prédiction statique avec des instructions du style *Branch likely* définie dans l'architecture MIPS II. L'utilisation d'un mécanisme de *scoreboarding*, unique sur ce type de composant à base de R3000, évite des gels du pipeline quand il n'y a aucune dépendance entre des instructions consécutives.

Le cache d'instructions optionnel est à correspondance directe et a une taille de 4 Ko. Il est non verrouillable (bien que cette caractéristique soit quelquefois souhaitée par des programmeurs temps réels pour conserver certaines fonctions fréquemment sollicitées par l'application).

5. en quantité de 10000 en 07/94.

	IDT R3081	Perf.Semi. PIPER	IDT R3051/3052	LSI LR 33000	LSI LR 33020
FPU intégrée	oui	oui	non	non	non
Caractéristiques spéciales	bus optionnel rapide, mode faible consommation	*real-time tracing bus*	Version à faible coût	Interface logique	coprocesseur graphique vidéo, contrôleur d'E/S
Cache instructions	16 Ko ou 8 Ko	8 Ko ou 4 Ko	8 Ko (3052) 4 Ko (3051)	8 Ko	4 Ko
Taille d'une ligne (en mots)	4	4	4	4	4
Remplissage (en mots)	4	4	4	1,2,4,8 ou 16	1,2,4,8 ou 16
Cache données	4 Ko ou 8 Ko	2 Ko ou 4 Ko	2 Ko	1 Ko	1 Ko
Taille d'une ligne (en mots)	4	4	4	4	4
Remplissage (en mots)	4	4	4	1,2,4,8 ou 16	1,2,4,8 ou 16
Espionnage du bus	externe	non	non	oui	oui
TLB	oui (version E)	oui	oui (version E)	non	non
Bus de données et d'adresses	multipléxé	multipléxé	multipléxé	non multipléxé	non multipléxé
Fréquence (en MHz)	20, 25, 33, 40	33, 40	20, 25, 33, 40	25, 33, 40	25

TAB. 1.2 - *Caractéristiques des principaux processeurs embarqués à base d'architecture MIPS R3000 (1992)*

Le cache de données est associatif par ensemble à deux voies et supporte un verrouillage par ligne. Sa capacité est de 1 Ko et il est à écriture simultanée (choix logique pour la plupart des systèmes embarqués).

L'interface externe a été conçue de manière à faciliter son intégration dans les systèmes. Les bus de 32 bits d'adresses et de données sont séparés et une logique de contrôle pour maintenir la cohérence au sein du cache est également implémentée (*cache snoop control*). Le remplissage du cache peut se faire sous un *mode rafale* configurable de 16 à 128 octets.

Caractéristiques technologiques. Construit en technologie à 0.6 μm avec deux couches de métal, le cœur du processeur occupe une surface de 15 mm^2 sans les caches. Le fait de rajouter les 5 Ko de cache double presque la surface de la puce (26 mm^2). La tension d'alimentation varie de 1.8 à 3.3 V maximum avec des fréquences de fonctionnement respectives de 15 à 50 MHz. La consommation maximale est de 550 mW, et peut-être quasiment diminuée de moitié pour des fréquences de fonctionnement moindres.

Conditionnement commercial. *Toshiba* commercialise ce processeur dans une version destinée aux applications enfouies. Sous le nom de R3901, ou encore *Southern Cross*, il englobe le cœur du R3900 avec les caches et ajoute un tampon de lecture/écriture, une unité de gestion de la mémoire (MMU), une unité de traçage et de mise au point temps réel ainsi qu'une interface bus.

Les caractéristiques de ce processeur le désignent tout particulièrement au marché des *Personal Digital Assistant* avec des performances de l'ordre de 52 Dhrystone. Actuellement le R3091 est sur le point d'être produit, la prochaine étape pour *Toshiba* étant un composant R4000PC de 15 Watt.

Divers

Nous présentons dans cette dernière partie quelques uns des systèmes construits autour de processeur R3000.

On peut citer par exemple le *custom* développé par *LSI Logic* destiné aux consoles de jeux. Ce *custom* est bâti autour d'un processeur R3000 séquencé à 33 MHz et combiné avec un coprocesseur graphique, une logique de décompression vidéo et une interface facilitant son intégration dans des systèmes à faible coût. Ce *custom* fabriqué en technologie 0.5 μm contient environ un million de transistors.

L'année 1993 a également été particulièrement faste en annonce de processeurs embarqués à base de R3000. En mars, *NEC* a introduit le VR3800. Ce composant inclut des caches d'instructions et de données internes et un générateur d'horloge. Disponible à des fréquences pouvant aller de 16 à 25 MHz, son prix varie de \$40 à \$49.

En juin, *LSI* a annoncé une mise à jour de son LR33020 destiné aux terminaux-X. Le LR33120 combine un cœur R3000, un contrôleur vidéo et des entrées/sorties spécialisées.

En octobre, *Sony* et *HDL* ont détaillé leur système à base de R3000. Ils utilisent tous deux une approche différente. Le R3100 est un *custom* spécialement adapté aux besoins de *Sony*, alors que celui de *HDL* utilise un modèle de synthèse afin de créer un circuit adapté aux besoins spécifiques des clients et à divers procédés de fabrication. Seul le *noyau RISC* du R3000 est conservé (pipelines, registres et génération des adresses), le reste étant totalement configurable (présence de caches, capacité en mémoire, multiplieur/diviseur, multiplieur-accumulateur, etc...).

1.2.2 Processeurs dérivés du MIPS R4000

De même que dans le cas des processeurs à base de R3000, le MIPS R4000 est à l'origine de toute une gamme de processeurs destinés au marché des systèmes embarqués hautes performances, des stations de travail ou encore des PCs. Nous essayerons dans cette partie de donner un aperçu de cette gamme à travers la présentation de trois processeurs.

Le R4400

Commercialisé à partir du premier trimestre 1993, ce processeur est le successeur du R4000. Également disponible en trois versions : le R4400PC (cache primaire seulement), le R4400SC (avec interface cache secondaire) et le R4400MC (pour systèmes multiprocesseurs), ce processeur initialement proposé à une fréquence interne de 150 MHz (75 externe), atteignait 250 MHz en juin 1995.

Implémenté dans une technologie à 0.6 μm à deux couches de métal, sa consommation est de 20 W et 10 W pour des tensions d'alimentation respectives de 5 V et 3.3 V.

Du point de vue de son architecture, ce processeur est sensiblement identique au R4000, les principales différences étant l'ajout d'un tampon d'écriture supplémentaire dans le but d'améliorer les performances graphiques et le doublement de la taille des caches (32 Ko au total), en plus de l'augmentation de l'horloge interne.

Commercialisé par l'ensemble des partenaires de MIPS, le R4400 offre de 60 à 80% de performances supplémentaires

Le R4600

Ce processeur est issue d'un projet lancé à l'origine par *IDT*, puis repris par *Quantum Effect Design* (*QED*), une petite firme fondée par d'anciens employés de MIPS dont l'objectif était de développer des processeurs MIPS à faible coût pour des systèmes *desktop* bon marché, et plus particulièrement ceux destinés à *Windows NT*.

Ce composant a été échantillonné en 1993 et commercialisé en 1994. Également connu sous le nom de *Orion*, ce processeur est compatible avec le R4000PC au niveau logiciel et brochage. Cependant sa structure interne est différente. Ce processeur utilise un pipeline RISC à cinq étages semblable à celui du R3000 avec un minimum de pénalité. Cette simplicité a permis de réduire le temps de conception ainsi que la surface occupée. Cette simplicité se retrouve également au niveau de l'absence de support pour un cache secondaire ou encore pour un système multiprocesseur. Ce processeur inclut 32 Ko de caches internes (associatif à deux voies) et est conçu pour fonctionner sous 3.3 V. Initialement annoncé à une fréquence de fonctionnement de 100 MHz, il est successivement passé à 133 puis 150 MHz (juillet 1994).

L'une des caractéristiques de ce processeur est également la mise hors tension de certaines unités fonctionnelles lorsqu'elles ne sont pas utilisées. Cette gestion dynamique permet de réduire de manière non négligeable la consommation.

Pour atteindre une haute fréquence d'horloge avec un pipeline plus simple, le processeur *Orion* tire partie du processus de fabrication d'*IDT* à 0.6 μm avec trois couches de métal. Ce processeur a des performances équivalentes à celles d'un Pentium, et vise le marché des systèmes *Windows NT*. Le R4600 et son dérivé, le R4700, se sont également imposés dans les stations de travail bas de gamme de *SGI*.

Le R4300 et le R4200

Annoncé en mai 1995, le R4300, conçu par *NEC*, est un composant 64 bits bâti sur le design du R4200 avec un bus plus simple et des performances supérieures. Ce composant est destiné au domaine du jeu vidéo et a été choisi pour équiper les plate-formes *Nintendo*.

Une deuxième version de ce composant, nommé R4300i (*i* pour *interactive*) devrait voir le jour au cours du deuxième semestre 1995. Plus connue sous le nom de l'architecture multimédia de MIPS *Magic Carpet*, ce processeur a les performances brutes d'un Pentium à 60 MHz et est en outre optimisé pour le traitement d'image 3D et supporte la décompression MPEG2. Son prix de $35 le destine à une très large diffusion. *Silicon Graphics* va l'utiliser pour les *set top boxes* destinés à la télévision interactive, mais d'autres sociétés se sont montrées également très intéressées (*ATT, Samsung Electronics, Philips Electronics*).

Présentation générale. Ce nouveau composant n'est pas sans rappeler l'architecture du R4200. Ce dernier a été développé par *MIPS Technologies Inc.* (*MTI*) mais à la différence d'autres processeurs de MIPS, a été exclusivement fondu par *NEC* avec une période d'exclusivité de un an (qui a depuis expiré). Le R4300 a été également développé par *MTI* avec l'aide de *NEC* mais devrait théoriquement être disponible pour les autres licenciés de MIPS.

Les deux composants sont quasiment identiques à l'exception de leur interface bus externe, l'unité de multiplication et le procédé de fabrication.

Les chemins de données sont partagés entre les unités entière et flottante. Le R4200 a été le premier processeur à mettre en œuvre cette technique qui permet d'économiser énormément de place en logique et routage mais sacrifie les performances entières et flottantes (par exemple, tout le CPU est gelé pendant les opérations flottantes à longues latences, soit jusqu'à 50 cycles). *MTI* a modifié le multiplieur combiné entier/flottant pour éviter ce problème sur le R4300. La nouvelle unité de multiplication exécute une multiplication de 32x32 bits en cinq cycles, que ce soit en entier ou en flottant. Les opérations double précision ou autres opérations 64 bits ont une latence de huit cycles.

Le R4300, comme le R4200, met en œuvre un pipeline de cinq étages semblable à celui du R3000. Par ailleurs, la taille du cache d'instructions a été doublée par rapport au R4000 et est de 16 Ko, la taille du cache de données restant inchangée (8 Ko). Le R4200, comme le R4300, ne supporte pas de cache secondaire ni d'interface permettant son intégration dans des systèmes multiprocesseurs.

La structure du TLB reste inchangée, mais le nombre de ses entrées a été ramené de 48 (sur le R4000) à 32.

L'une des profondes modifications effectuées sur ce composant concerne le bus système externe. Dans le but de faciliter sa gestion, MIPS l'a divisé en deux bus de 32 bits et a ramené les signaux de contrôle à cinq bits. Cette nouvelle structure permet au R4300 de coexister avec d'autres maîtres, tels que des composants DMA.

Technologie. Le R4300 est fabriqué selon la même procédé que le R4100 et le R4400 à 200 MHz, en utilisant une technologie à 0.35 μm avec trois couches de métal. *NEC* teste actuellement le R4300 à 100 MHz avec une production prévue au mois de septembre 1995.

De même que l'ensemble des processeurs destinés au marché des systèmes enfouis, ce processeur dispose d'un mode de consommation réduite.

Le R4300 affiche des performances de 125 Dhrystone ce qui le place non loin de ses concurrents (*Intel 960*, et le *SH7708 d'Hitachi*). Il améliore les performances entières du R4200 de 10% et les performances flottantes de 50% du fait de l'amélioration du multiplieur.

Divers

Nous présentons dans cette dernière partie quelques uns des processeurs à base de R4000. Comme on l'a vue précédemment, l'architecture de ces processeurs peut être très différente du processeur original mais reste cependant conforme au jeu d'instructions MIPS III.

Annoncé en novembre 1994, le R4650 est un dérivé de l'architecture *Orion*. Produit par *IDT* et *NKK*, *Quantum Effect Design* est à l'origine de ce nouveau processeur. Ce composant vise le marché des systèmes embarqués nécessitant des calculs intensifs : routeurs de réseaux, applications vidéo de compression et décompression et consoles de jeux haut de gamme. Ce processeur améliore la latence de la multiplication entière d'un facteur de quatre par rapport au R4000 et R4600. Il inclut par ailleurs de nouvelles instructions telles que :

- une instruction de multiplication qui écrit un résultat de 32 bits directement dans un registre général (au lieu d'utiliser les registres spécialisés suivi d'une instruction de transfert) ;
- une instruction de multiplication-addition effectuée en trois cycles ;
- une instruction *Wait*. Cette instruction permet de réduire la consommation du processeur. Après une instruction *Wait*, la consommation passe à 200 mW, le processeur pouvant être réactivé par une interruption externe.

Il est à noter cependant que les opérations flottantes double-précision ont été supprimées afin de gagner de la place. Ces instructions déclenchent une exception et sont émulées logiciellement. La structure des caches est la même que sur le R4600 (caches d'instructions et de données séparés, associatif à deux voies) avec cependant une possibilité de verrouillage par logiciel d'un des ensembles.

Par ailleurs, ce processeur abandonne le concept de gestion de la mémoire, jugé complexe et coûteux pour un système enfoui, et inclut plutôt une protection de la mémoire plus rudimentaire par l'intermédiaire d'un ensemble de registres *frontières*. Deux registres servent de frontière supérieure pour les instructions et les données. Si une adresse virtuelle est supérieure ou égale à la valeur contenue dans l'un de ces registres, une exception est générée.

Un deuxième processeur, le R4700, a également été annoncé en octobre 1994 par *IDT*. Ce processeur constitue également une nouvelle version du R4600 destinée aux stations de travail d'entrée et de milieu de gamme. Ce nouveau composant améliore les performances flottantes d'environ 20% grâce à l'amélioration du multiplieur flottant. Il devrait délivrer 87.5 Mflops à 175 MHz grâce à l'utilisation d'une technologie CMOS à 0.6 μm au lieu des 0.64 μm utilisés pour le R4600.

Dernier processeur annoncé par *NEC* en mars 95, le R4100 est destiné au marché des *Personal Digital Assistant* grâce à sa faible consommation et à ses performances. Il inclut une instruction de multiplication-addition et délivre 45 Dhrystone à 40 MHz. Le R4100 sera vendu exclusivement par *NEC*.

Ce processeur est issu du R4200 avec des améliorations qui le prédestinent à des applications enfouies où les critères de consommations sont vitaux. Alimenté sous 3.3 V, sa consommation est

de 120 mW (à une fréquence de 40 MHz). Descendre sa tension à 2.2 V fait décroître sa fréquence à 20 MHz et sa consommation à 27 mW. Comme la plupart des processeurs embarqués, il dispose de plusieurs modes de fonctionnement en consommation réduite :

- le mode *standby* inhibe toute activité interne, l'interface bus restant opérationnelle ;
- le mode *suspend* suspend toute activité.

Ces deux modes permettent de réduire de 90 et 95% la consommation, une quelconque interruption externe permettant de rétablir un fonctionnement normal en quelques cycles. Un mode *hibernation* permet par ailleurs de mettre en veille le processeur, une intervention matérielle étant alors nécessaire pour le relancer (*hard reset*).

Ce processeur a une architecture 64 bits. Il supporte le jeu d'instructions MIPS III et inclut également une nouvelle instruction de multiplication-addition de type ($16*16+64 \rightarrow 64$ *bits*) utile pour des applications audio et vidéo.

Ce processeur inclut des caches d'instructions et de données de 2 Ko et de 1 Ko à correspondance directe et à écriture différée. Il devrait être réalisé en technologie CMOS à 0.5 μm avec trois couches de métal.

1.3 Perspectives

Selon les plans dévoilés par MIPS, SGI a demandé à *MTI* et *QED* de travailler ensemble sur le successeur du R4600, projet connu sous le nom de P4. Normalement disponible à partir du deuxième trimestre 96, ce processeur devrait atteindre 200 SPECint92, soit 50% de plus que l'*Orion* le plus rapide actuellement. Il devrait recevoir plus d'attention sur les performances flottantes qui devraient atteindre 300 SPECfp92. Ces performances requièrent un processeur superscalaire séquencé à 200 MHz. Le processeur *Orion*, destiné au marché *Windows NT*, a été limité par ses performances flottantes. Les spécifications du P4 au contraire le désignent comme un processeur de station de travail de milieu de gamme.

En 1998, le D2 devrait succéder au P4. Ce processeur devrait délivrer des performances du niveau du R10000 avec un coût équivalent à celui de l'*Orion*.

Dans le haut de gamme, le R8000 devrait voir sa fréquence de fonctionnement croître jusqu'à 100 MHz avec l'utilisation d'une technologie à 0.5 μm. Déjà les premiers systèmes à 90 MHz sont annoncés.

Cependant ce produit sera rendu obsolète par le R10000 que nous présentons au chapitre suivant. Initialement prévu avec une technologie CMOS à 0.5 μm et une fréquence interne de 200 MHz, une version à 0.35 μm est attendue pour fin 1996 et devrait permettre d'atteindre 300 MHz et jusqu'à 500 SPECint92. *MTI* travaille déjà sur son successeur, le H1, qui vise les 1000 SPECint92 pour le premier semestre 1998.

1.3.1 Le processeur MIPS R10000

Annoncé en octobre 1994, ce processeur est le tout dernier RISC 64 bits de MIPS. Il intégre les principales techniques évoluées d'architecture de processeurs telles que architecture superscalaire,

exécution dans le désordre, prédiction dynamique à deux bits, pile de branchement, renommage de registres, etc...

Ce processeur est annoncé à 200 MHz. Il met en œuvre une architecture découplée avec trois files d'instructions indépendantes [6] :

- une file d'instructions d'arithmétique entière ;
- une file d'instructions flottantes ;
- une file d'instructions de lecture/écriture.

À chaque cycle, quatre instructions peuvent être chargées à partir du cache et six peuvent être émises vers les unités d'exécution (dans le cas le plus favorable : deux instructions arithmétiques entières, deux instructions arithmétiques flottantes, une lecture/écriture et une instruction de branchement conditionnel flottant).

Le R10000 utilise un pipeline de cinq étages pour les opérations arithmétiques entières, six étages pour les accès mémoires et sept étages pour les opérations flottantes. Les deux premiers étages sont communs (accès au cache, décodage, renommage et détection de branchement s'effectuant au cycle suivant). Le MIPS R10000 implémente 64 registres physiques pour les instructions entières et flottantes, soit le double de registres logiques vus par le programmeur. Au fur et à mesure de leur décodage, toutes les instructions sont renommées avant d'être expédiées vers leurs files respectives. La table de renommage entière a douze ports de lecture et quatre ports d'écriture alors que celle concernant les instructions flottantes a seize ports de lecture et quatre ports d'écriture, ceci afin de soutenir un traitement de quatre instructions par cycle. Quand les instructions sont émises vers leur file respective, elles sont également entrées dans une liste d'instructions « actives ». Cette liste contient jusqu'à 32 instructions consécutives et garantit que l'exécution des instructions effectuée de manière spéculative ou dans le désordre est conforme à un modèle ordonné pour le reste du système. Une instruction peut être extraite de cette liste quand toutes les instructions qui la précède ont été effectuées. Jusqu'à quatre instructions peuvent être extraites de cette liste à chaque cycle (à raison cependant d'une seule écriture). Par ailleurs ce concept de liste d'instructions active est particulièrement utile pour la gestion des branchements ou pour traiter les exceptions. Effectivement, en cas d'exception, la liste est utilisée pour « défaire » les résultats des instructions exécutées dans le désordre. Ceci est effectué de manière simple à travers la table de renommage.
Dans le cas des branchements, la table de renommage est sauvegardée au moment de l'exécution spéculative de l'instruction de branchement et est restituée si une mauvaise prédiction est détectée. La structure de sauvegarde de cette table supporte jusqu'à quatre niveaux de branchement.

Le cache d'instructions a une capacité de 32 Ko et est associatif à deux voies. Les instructions sont partiellement décodées avant d'y être inscrites. Un *cache de reprise* à quatre entrées permet de réduire la pénalité d'annulation des branches mal prédites (deux cycles si les instructions sont présentes dans ce cache, trois autrement). Le cache de données primaire a une structure particulière avec une taille de 32 Ko organisé en deux bancs de 16 Ko entrelacés associatifs par ensemble à deux voies. Par ailleurs une interface pour un cache secondaire externe d'une capacité maximale de 16 Mo est implémentée. Un bus de données de 128 bits permet d'atteindre un taux de transfert

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

de 3.2 Go/sec. Appelé *bus avalanche* par MIPS, ce bus a la possibilité de maintenir jusqu'à huit transactions en attente d'exécution.

Le support multiprocesseur a été considérablement amélioré. Le R10000 supporte une configuration en grappe. Jusqu'à quatre processeurs peuvent être placés sur le même bus 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.

Le R10000 implémente le jeu d'instructions MIPS IV mis en œuvre sur le R8000. Réalisé en technologie CMOS 0.5 μm à quatre couches de métal, les performances annoncées de ce processeur sont de l'ordre de 300 SPECint92 et 600 SPECfp92.

Chapitre 2

Les microprocesseurs Alpha

L'architecture Alpha AXP est issue de la volonté d'un groupe de travail organisé en 1988 d'explorer les diverses voies qui permettraient de préserver les clients VAX à travers les années 1990. Définie comme l'architecture des 25 prochaines années, elle est censée répondre aux exigences suivantes :

- hautes performances ;
- longévité ;
- capacité d'exécuter les systèmes d'exploitation VMS et UNIX ;
- migration facile à partir des architectures VAX et MIPS.

Au contraire de l'architecture VAX, cette architecture est ouverte ; c'est à dire que tout fondeur 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. Il n'existe cependant pas à notre connaissance de processeur Alpha développé par d'autres sociétés que DEC.

L'architecture Alpha étant relativement récente, elle ne compte qu'un nombre limité de processeurs.

Annoncé en 1992, le DEC 21064 est le premier processeur à utiliser cette architecture ([2], [7]). Destiné au marché des serveurs et des stations haut de gamme, ce processeur d'abord sorti à la fréquence de 150 MHz, atteint maintenant 200 MHz. Bénéficiant de l'avancée technologique, une deuxième version de cette architecture a été annoncée en 1993 à la fréquence de 225 et 275 MHz (DEC 21064A).

Basée sur la même architecture, 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.

Le DEC 21164 est le dernier processeur de cette famille. Annoncé en 1994, il constitue la deuxième implémentation de l'architecture Alpha AXP. Ce processeur vise essentiellement le marché des serveurs et des stations de travail haut de gamme.

Nous développerons brièvement dans une première partie quelques caractéristiques de l'architecture Alpha avant d'aborder son implémentation à travers l'étude de ces processeurs. Pour plus de détails sur l'architecture Alpha, le lecteur pourra se reporter à [8].

2.1 L'architecture Alpha

2.1.1 Caractéristiques générales

L'architecture Alpha AXP est une architecture RISC de type *load/store*, 64 bits. L'accent a été mis sur sa haute fréquence d'horloge, l'émission simultanée de plusieurs instructions, l'aspect multiprocesseur et le support de plusieurs systèmes d'exploitation.

N'ayant aucune compatibilité binaire à assurer, une architecture 64 bits a été définie. Tous les registres ont une taille de 64 bits (32 registres flottants et entiers) et toutes les opérations sont effectuées entre des registres de 64 bits. Toutes les instructions ont une taille fixe de 32 bits. Les seules opérations d'accès à la mémoire sont des opérations de lecture/écriture. Toutes les manipulations de données sont faites entre registres.

L'architecture Alpha supporte les types de données suivants :

- les entiers sur 8 (octet), 16 (mot)[1], 32 (double-mot) et 64 bits (quadruple-mot) ;
- les formats flottants *IEEE-754* 32 et 64 bits ;
- les formats flottants VAX 32 et 64 bits.

L'architecture Alpha utilise un ensemble de routines, appelé *Privileged Architecture Library code* (*PALcode*), spécifique à un système d'exploitation et à une implémentation matérielle. Ces routines sont utilisées lors des changements de contexte, pour les interruptions et les exceptions ainsi que pour la gestion de la mémoire. Elles peuvent être invoquées soit par matériel, soit directement par l'intermédiaire de l'instruction *Call_Pall*.

L'une des particularités de l'architecture Alpha est qu'elle n'implémente aucune instruction de lecture/écriture d'octet ou de mot de 16 bits. Ces formats de données sont accédés par combinaison d'instructions de décalage, de masquage et d'extraction.

L'ordre des octets en mémoire est conforme au schéma *little-endian*. Le format *big-endian* est également supporté mais nécessite une instruction supplémentaire[2].

2.1.2 Le jeu d'instructions

L'architecture Alpha définit cinq formats d'instructions :

- format des instructions mémoire ;
- format des instructions de branchement : branchement relatif au PC avec un déplacement de 23 bits (codé sur 21 bits les deux bits de poids faible étant toujours nuls) ;

1. 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.

2. L'endianness n'affecte seulement que les instructions susceptibles de manipuler des données de format inférieur à 64 bits.

- 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.

Trois modes d'adressage sont supportés : absolu, indirect et basé (le registre R31 est cablé à zéro). L'architecture Alpha n'implémente pas de branchement retardé (comme sur le MIPS). Ce type de branchement rend plus complexe la gestion d'un séquencement superscalaire.

Comme on l'a dit précédemment, l'architecture Alpha ne permet des accès à la mémoire que pour des mots de 32 bits. Ce choix a pour conséquence une pénalité sur l'ensemble des programmes manipulant des données de 8 ou 16 bits. DEC affirme que moins de 8% des références mémoires VAX/VMS concernent des quantités inférieures à 32 bits, et que beaucoup concernent des manipulations de chaînes de caractères qui peuvent être effectuées sur des double ou quadruple-mots (présence d'instructions de manipulation d'octets). Par contre, les données de 32 bits ne sont pas forcément alignées sur des frontières de 32 bits. Le jeu d'instructions comprend donc des instructions qui permettent d'accéder à ces données en ignorant les trois bits de poids faible. La gestion de ce type de données est particulièrement importante pour DEC amené à opérer sur des programmes VAX où des structures de données non alignées peuvent subsister malgré la recompilation.
Cette caractéristique a également un impact important sur les caches à écriture différée. DEC implémente un mécanisme de correction d'erreur de type ECC. Pour implémenter un cache à écriture différée accessible par octet avec une protection ECC, des bits de corrections et de tests doivent être associés à chaque octet. Ceci est relativement coûteux et pourrait être évité en abandonnant ce type de données.

Le jeu arithmétique entier définit par l'architecture Alpha est traditionnel. Les opérations d'addition, de soustraction et de multiplication sont fournies en version 32 et 64 bits. La division entière par une constante est effectuée en multipliant par l'inverse, la division par une variable est quant à elle émulée logiciellement (table de *look-up* et algorithme itératif).

Les instructions de déplacement conditionnel sont spécifiques à l'architecture Alpha mais commencent à se trouver sur d'autres jeu d'instructions (SPARC-V9 par exemple). Ces instructions recopient un registre source vers un registre destination si un troisième registre vérifie une condition. Cette condition est testée par rapport à zéro. Ces instructions permettent d'implémenter de manière élégante un cas fréquent sans provoquer de rupture dans le pipeline.

Nous n'expliciterons pas plus longuement les particularités de ce jeu d'instructions. nous renvoyons le lecteur à [4] et [8] pour plus de détails à ce sujet.

2.2 Les processeurs Alpha AXP

Nous avons séparé l'étude des processeurs Alpha en trois parties afin de faire ressortir la démarche menée par DEC. Dans une première partie, nous étudions les processeurs DEC 21064 et DEC 21064A qui nous permettent d'introduire les concepts de base de l'implémentation de cette architecture. Puis nous étendons cette étude au domaine des processeurs enfouis, avant d'aborder dans une dernière partie l'étude du tout dernier processeur DEC Alpha, le DEC 21164.

2.2.1 Les processeurs 21064 et 21064A

Annoncé en 1992, le DEC 21064 est la première mise en œuvre matérielle de l'architecture RISC Alpha. Outre sa mise en œuvre superpipeline, il utilise également la technique du superscalaire (jusqu'à deux instructions par cycle). Tout a été mis en œuvre dans sa conception pour obtenir une fréquence d'horloge élevée : 200 MHz en 1992 !

Bénéficiant de l'avancée technologique, DEC a annoncé en 1993 le 21064A qui reprend à peu près la même implémentation. C'est pourquoi nous aborderons dans une même partie ces deux processeurs en précisant au moment voulu les quelques différences qui les distinguent.

Le DEC 21064 inclut trois unités fonctionnelles d'exécution entièrement pipelinées :

- 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 comme son nom l'indique gère les accès à la mémoire et effectue les calculs d'adresses.

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

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.

Émission et exécution des instructions

Émission des instructions. Il nous parait intéressant de développer les mécanismes d'émission d'instructions de cette architecture dans la mesure où le DEC 21064 est l'un des premiers processeurs (avec l'IBM POWER) à avoir été confronté aux problèmes inhérents à une architecture superscalaire. Pour plus de détails, le lecteur pourra se reporter à [2].

Le chargement des instructions à partir du cache, et leur lancement dans les diverses unités d'exécution est assuré par l'IBOX. Cette unité dispose en outre d'un tampon d'instructions à une seule entrée qui lui permet d'anticiper le chargement des instructions afin de réduire au maximum la pénalité en cas de défaut de cache.

L'IBOX décode deux instructions en parallèle. Si les ressources demandées sont disponibles, les deux instructions sont émises simultanément, sinon, seule la première est exécutée, le séquencement étant alors arrêté en attendant la résolution de l'interblocage. Le séquencement est toujours fait dans l'ordre. Les combinaisons d'instructions possibles sont :

- opération entière + opération flottante ;
- opération flottante + branchement flottant ;
- opération entière + branchement entier ;
- opération de lecture/écriture + toute autre opération. Deux combinaisons sont cependant impossibles dans ce dernier cas :
  - écriture entière + opération flottante ;

– écriture flottante + opération entière.

L'IBOX contient également une unité logique de prédiction de branchement. La prédiction de branchement mise en œuvre a été améliorée entre les deux versions de processeurs DEC 21064 et 21064A.

Le DEC 21064 met en œuvre deux stratégies de prédiction dynamique[3] : la première consiste à enregistrer le résultat des instructions de branchement dans un bit de la ligne du cache d'instructions. Ce bit indique si lors de la dernière exécution de l'instruction, le branchement a été pris ou non. Ce comportement est enregistré afin de pouvoir prévoir le comportement de l'instruction de branchement lors de la prochaine exécution. La deuxième stratégie de prédiction est basée sur le signe du déplacement du branchement :

– si le déplacement est négatif, c'est-à-dire si la cible du branchement concerne une adresse antérieure à celle de l'instruction de branchement (saut en arrière), le branchement est prédit pris (normal, cas d'une boucle par exemple) ;

– si le déplacement est, au contraire, positif (saut en avant), le branchement conditionnel ne sera pas pris.

Cette seconde stratégie est généralement utilisée lors de la première exécution d'une instruction de branchement, car le bit d'historique n'a pas encore pu être positionné (ou alors lorsque la table d'historique des branchements est invalidée).

Le DEC 21064A utilise quant à lui une technique de prédiction de branchement totalement différente, reprise également sur le DEC 21164. Cette stratégie fait appel à une table de prédiction de branchement basée sur une prédiction à deux bits selon l'algorithme de Smith. Cette table est accédée en parallèle avec le cache d'instructions au premier étage du pipeline d'instructions. Deux bits sont associés à chaque instruction. Cette nouvelle stratégie (identique à celle du Pentium) devrait améliorer la prédiction de 5 à 10%.

Par ailleurs, les deux processeurs implémentent une pile de quatre entrées destinée à conserver les adresses de retour de procédures. Cette pile est sollicitée lors du décodage du code opération de l'instruction et apporte une grande sûreté dans la prédiction de l'adresse de retour.

Exécution des instructions. Comme on l'a dit précédemment, le DEC 21064 utilise la technique du superpipeline avec un pipeline de sept étages. Les quatre premiers servent au chargement des instructions, à leur décodage, à la prédiction de branchement ainsi qu'à la résolution des conflits sur un processeur superscalaire de degré deux. Ces quatre premiers étages sont dit *statiques* dans la mesure où les instructions peuvent attendre à ce niveau la résolution des conflits de ressources, par opposition aux étages d'exécution dits *dynamiques*, et dans lesquels la phase d'exécution des instructions ne peut être interrompue. Ces quatre premiers étages sont donc communs aux pipelines entier et flottant. Le pipeline flottant du DEC 21064 a une profondeur de dix étages.

Les exceptions et interruptions. La gestion précise des exceptions est délicate à réaliser sur un processeur superscalaire, c'est pourquoi, le choix d'une gestion imprécise a été retenue dans la définition de l'architecture Alpha. Ces exceptions sont gérées par les routines de la librairie PALcode.

3. Le DEC 21064 était le premier processeur à mettre en œuvre une prédiction dynamique des branchements.

Le 21064 supporte trois sources d'interruption :

- matérielle, avec six niveaux d'interruption ;
- logicielle, avec 15 niveaux d'interruption configurés par un registre interne ;
- interruptions asynchrones, au nombre de quatre configurées également par un registre interne spécifique.

Les unités arithmétiques

L'unité arithmétique entière. Désignée sous le nom de EBOX, cette unité comprend les opérateurs classiques de calcul arithmétique (additionneur, décaleur, etc...). Les 32 registres généraux implantés pour l'unité entière sont de 64 bits (R31 étant cablé à zéro). Il est à noter que la mise en œuvre superscalaire du DEC 21064 rend nécessaire l'extension des ports d'accès au banc de registres.

L'unité arithmétique flottante. Le banc de registres flottants du DEC Alpha est entièrement indépendant du banc de registres entiers : cette mise en œuvre permet un accès en parallèle aux registres entiers et flottants qui est indispensable dans le cas d'une architecture superscalaire autorisant le lancement simultané d'une opération flottante et d'une opération entière. Cependant, l'absence de chemins de données entre les registres flottants et entiers oblige un passage par la mémoire pour chaque transfert de données entre les deux bancs de registres. Le banc de registres flottants est composé de 32 registres de 64 bits et d'un registre de contrôle.

En plus des deux formats standards de la norme *IEEE 754*, il existe sur le DEC Alpha un sous-ensemble de formats flottant VAX qui permet d'assurer la compatibilité de résultats pour des applications précédemment développées sous ce format.

Toutes les opérations, mise à part la division, sont entièrement pipelinées et peuvent donc être lancées au taux d'une instruction par cycle.

La latence de la division flottante est de 31 cycles pour des opérations simple précision et de 61 cycles pour les opérations double précision. Toutefois, toutes les opérations indépendantes peuvent continuer de s'exécuter pendant que des instructions de longues latences s'effectuent. Toutes les autres opérations flottantes ont une latence d'au moins six cycles : en effet, étant donné la haute fréquence d'horloge utilisée, il devient difficile de mettre en œuvre un mécanisme de chaînage.

Sur les deux processeurs, les unités sont identiques à l'exception de la prise en compte de certains *flags*, améliorée sur le 21064A.

Hiérarchie mémoire

Les processeurs DEC 21064 et 21064A mettent en œuvre un premier niveau de caches d'instructions et de données séparés. Ces caches ont des caractéristiques communes : ils sont à correspondance directe, physiquement adressés. La taille relativement petite de 8 Ko des caches du DEC 21064 a été étendue à 16 Ko sur le 21064A.

Il est à noter que sur les deux processeurs, le cache d'instructions inclut un identificateur de processus de 6 bits.

Seul le DEC 21064A inclut une protection par parité de ses caches primaires d'instructions et de données.

Le cache de données est à écriture simultanée. Un tampon d'écriture de quatre lignes de 32 octets permet de regrouper les accès vers une même ligne mémoire. Pour être réellement efficace le DEC 21064 est conçu pour être utilisé avec un cache secondaire.

Le cache secondaire du DEC 21064 est un cache externe optionnel dont la taille varie de 128 Ko à 16 Mo. Il est à écriture différée. Le DEC 21064 fournit toute la logique de contrôle nécessaire à la mise en œuvre de ce cache. On peut remarquer ici que les données n'ont pas à traverser le processeur pour être inscrites dans ce cache. La largeur du bus de données est de 128 bits. Ce bus est utilisé aussi bien entre le processeur et le cache secondaire, qu'entre le cache secondaire et la mémoire principale.

Support des systèmes d'exploitation

L'architecture Alpha gére 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, OSF, 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 définit une taille des pages pouvant varier de 8 Ko à 4 Mo. Cependant, le DEC 21064 met en œuvre des pages de taille constante égale à 8 Ko.

L'adresse virtuelle est étendue par un identificateur de processus nommé *ASN* sur le DEC 21064 (pour *Address Space Identifier*). Cet identificateur est codé sur six bits permettant ainsi à 64 processus d'être simultanément résidents en mémoire.

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.

Caches de traduction d'adresses. Le DEC 21064 possède deux TLB distincts : un TLB d'instructions et un TLB de données. Le choix de TLB distincts provient de la nécessité de disposer de l'adresse physique à chaque accès aux caches (instructions et données). Le TLB d'instructions possède 12 entrées :

- 8 entrées qui manipulent l'adresse de pages de 8 Ko,
- 4 entrées qui manipulent l'adresse de pages de 4 Mo (peut en particulier être utilisé pour mapper le système)

Le TLB de données possède 32 entrées et peut manipuler les adresses de pages dont la taille est de 8 Ko pour le processeur 21064 mais qui pourra être de 64 Ko, 512 Ko et 4 Mo dans les futures mises en œuvre du DEC Alpha. Ces deux caches de traduction d'adresses sont entièrement associatifs et utilisent la stratégie de remplacement de cache de type NLU (*Not Last Used*).

Ces deux TLBs sont gérés logiciellement grâce à des fonctions du PALcode. La taille des TLBs (en particulier instructions) est particulièrement faible pour un processeur haut de gamme. Ceci explique en grande partie les performances parfois décevantes du DEC 21064.

Protection. Sur le DEC Alpha, les appels au mode noyau sont implantés via une librairie, le PALcode (*Privileged Architecture Library*), servant d'intermédiaire entre le système d'exploitation et le matériel. Le PALcode a l'avantage d'avoir des fonctions pouvant être utilisées dans plusieurs mise en œuvres. Ces fonctions remplacent celles microcodées que l'on peut trouver dans les processeurs CISC. Elles permettent d'assister le matériel dans un certain nombre d'opérations :

- remplissage du TLB en cas de défaut ;
- routines de traitement des exceptions ;
- primitives de synchronisation.

Le Palcode fonctionne dans un environnement particulier dans lequel les interruptions sont masquées afin de pouvoir exécuter des séquences de code de manière atomique. L'utilisation des instructions réservées au PALcode n'est autorisée que dans cet environnement. Dans le cas contraire, une exception d'instruction illégale est déclenchée.

Les fonctions PALcode sont appelées à travers l'instruction CALL_PAL, qui provoque une exception dans le matériel. Un mécanisme masque alors les interruptions, autorise le mode privilégié pour les routines du PALcode et sauve l'état courant de la machine. La transition inverse est ensuite réalisée pour retrouver l'environnement initial.

Le DEC 21064 supporte les quatre modes classiques de fonctionnement (*noyau, superviseur, exécution, utilisateur*).

Support multiprocesseur

Destinée à être l'architecture des 25 prochaines années, les concepteurs de l'architecture Alpha ont bien entendu privilégié l'intégration de leur processeurs 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 DEC Alpha définit également l'ordre des accès à la mémoire dans un système à mémoire partagée.

Il est à noter que *Cray* a fait le choix de cette architecture pour la mise en œuvre de ses systèmes parallèles de type *Cray T3D*.

Les compteurs 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 processeurs DEC 21064 et 21064A 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.

Caractéristiques technologiques

Quasiment deux années séparent la commercialisation de ces processeurs. Les technologies employées pour chacun d'eux sont représentatives de l'évolution effectuée.

Le DEC 21064 est implémenté dans une technologie CMOS à 0.65 μm à trois couches de métal. Il est disponible en version 150, 166 et 200 MHz.

Le DEC 21064A utilise quant à lui une technologie CMOS à 0.5 μm à quatre couches de métal. Il est séquencé à 225 et 275 MHz.

2.2.2 Les processeurs DEC 21066 et 21068

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. L'intérêt de ces processeurs est d'intégrer sur la puce un contrôleur mémoire, un contrôleur de cache ainsi qu'une interface *PCI*[4] réduisant ainsi le coût et la complexité de l'intégration de ces processeurs dans des systèmes. Selon le même principe que le 21064A, une deuxième version du DEC 21066 (le DEC 21066A) a été annoncée en novembre 1994.

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 Ko à 2 Mo).

En fait, les principales différences résultent de l'adaptation de ce processeur aux marchés des systèmes d'entrée de gamme (réduction du coût) et des processeurs enfouis (contrôleur d'entrées/sorties).

Différences avec le DEC 21064

Les bus. Afin de réduire le coût des processeurs destinés à une plus large diffusion, des efforts important ont été faits pour réduire leur nombre de broches. Cette réduction se ressent au niveau de la largeur des bus de données avec l'extérieur. Les processeurs DEC 21066 et 21068 mettent en œuvre un bus de 64 bits vers le cache externe au lieu du bus de 128 bits du DEC 21064 (le remplissage d'une ligne de cache se fait donc en quatre cycles au lieu de deux).

Par ailleurs, le bus d'adresses a été totalement supprimé, celui-ci étant désormais multiplexé sur le bus des index du cache. De-même, les processeurs DEC 21066 et 21068 calculent une valeur ECC sur 64 bits alors que le 21064 maintenait un ECC pour chacune des sections 32 bits de son bus 128 bits. Les signaux de contrôle du cache et de la mémoire sont simplifiés. Les microprocesseurs 21066 et 21068 n'implémentent par ailleurs aucun support multiprocesseur.

Cette politique de réduction du nombre de broches a permis d'économiser jusqu'à 144 broches entre le DEC 21064 et les processeurs DEC 21066 et 21068, et ce malgré l'introduction d'un bus PCI. Cependant, cela n'est pas sans conséquences sur les performances, puisque selon [9] la réduction du bus vers la mémoire devrait dégrader les performances de 25%.

Le contrôleur mémoire. Le contrôleur mémoire contrôle la mémoire externe primaire du système (jusqu'à 512 Mo). Il a également la possibilité de lire et d'écrire le cache secondaire. Il supporte également de nombreuses opérations graphiques.

Le contrôleur d'entrées/sorties. Son rôle est de gérer les communications entre le CPU, la mémoire et l'ensemble des périphériques externes. Tous les périphériques d'un système sont connectés au CPU et à la mémoire par l'intermédiaire de ce contrôleur. Un composant extérieur est cependant nécessaire pour l'arbitrage du bus PCI entre les divers composants.

4. *Peripheral Component Interconnect Bus.*

Le contrôleur d'entrées/sorties dispose d'une logique de type *scatter-gather* qui effectue la traduction des adresses 32 bits générées par le maître du bus PCI, vers les adresses 64 bits du processeur. Afin de permettre une traduction rapide, le contrôleur implémente une table de traduction d'adresses de huit entrées.

Le contrôleur joue le rôle d'un maître du bus PCI quand le CPU exécute une instruction qui adresse un périphérique. De même, quand le contrôleur répond à une transaction émise par un périphérique externe ou un contrôleur DMA maître du bus PCI, il est alors esclave. Les microprocesseurs DEC 21066 et 21068 sont conçus pour assurer la cohérence de la mémoire dans tous les cas de transferts.

PLL - *Phase-Locked Loop.* Les processeurs DEC 21066 et 21068 incluent un multiplicateur de fréquence permettant d'accepter une horloge plus basse, facilitant ainsi leur intégration sur des systèmes bon marché. La valeur du multiplicateur est programmable (par tout nombre entier compris entre 2 et 9. Une utilisation directe concerne l'horloge du bus PCI à 33,3 MHz qui peut être utilisée en positionnant le PLL pour une valeur multiplicative de cinq). Ceci permet d'éliminer par ailleurs une entrée de signaux d'horloge à haute fréquence (sur le DEC 21064, comme sur le 21164, il est nécessaire d'amener une fréquence d'horloge double de celle utilisée, soit respectivement 400 MHz et 600 MHz. Ceci a pour effet de rebuter certains constructeurs de systèmes).

Caractéristiques technologiques. Ces deux microprocesseurs sont réalisés en technologie CMOS à 0.65 μm. Le DEC 21066 est séquencé à la fréquence de 166 MHz. Le DEC 21068 atteint quant à lui 66 MHz.

2.2.3 Le processeur 21164

Le DEC 21164 a été annoncé le 7 septembre 1994, et la commercialisation de systèmes bâtis autour de ce processeur a débuté en avril 1995. Ce processeur est destiné aux systèmes scientifiques, aux serveurs haut de gamme et au marché des stations de travail.

Deuxième implémentation de l'architecture Alpha, le DEC 21164 est représentatif des processeurs annoncés fin 1994 à travers une mise en œuvre superscalaire de degré quatre et une haute fréquence d'horloge (266 MHz et 300 MHz, une récente annonce faisant même état d'un processeur à 333 MHz). Par ailleurs, les capacités d'intégration atteintes lui permettent d'inclure sur le même composant un deuxième niveau de cache de 96 Ko.

Ce processeur améliore l'ensemble des caractéristiques développées dans le chapitre relatif au DEC 21064. Le jeu d'instructions n'a subi aucune modification entre les deux générations de processeurs.

Du fait de la haute fréquence d'horloge, la technique du superpipeline a été utilisée, le pipeline entier compte sept étages, le pipeline flottant neuf. 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 d'instructions qui permet d'anticiper le chargement a été élargi à quatre entrées. Pour pouvoir pleinement exécuter ces quatre instructions, l'architecture du DEC 21164 inclut quatre unités fonctionnelles (deux unités d'exécution entières et flottantes). L'émission des instructions dans les diverses unités fonctionnelles est synchrone et se fait toujours

dans l'ordre. Le mécanisme de prédiction de branchement est identique à celui mis en œuvre sur le DEC 21064A. Cependant la profondeur de la pile d'adresses de retour utilisée a été amenée à 12 entrées.

La plupart des latences d'opérations entières et flottantes ont été améliorées. L'exécution des instructions de branchement conditionnel a été retardée de un cycle dans le but d'optimiser le débit du pipeline. En effet, cette astuce permet de séquencer au même cycle une opération et l'instruction de branchement directement dépendante. Le résultat de la prédiction de branchement est retardé d'un cycle mais l'importante table de prédiction (2 K entrées) devrait compenser ce retard.

La hiérarchie mémoire mise en œuvre sur ce processeur est impressionnante avec les deux niveaux de caches internes, plus un troisième niveau optionnel externe. Le DEC 21164 est le premier processeur à inclure sur la même puce deux niveaux de hiérarchie mémoire !

Le premier niveau de mémoire est constitué d'un cache d'instructions et de données séparés. Ils ont les mêmes caractéristiques que ceux de son prédécesseur mis à part leur adressage virtuel et l'extension du champ d'identification de processus à sept bits (au lieu de six). L'une des différences majeures avec le DEC 21064 est la présence d'un deuxième niveau de cache interne, d'une capacité de 96 Ko, associatif par ensemble à trois voies, à écriture différée. Ce cache constitue une solution au problème de l'adressage d'un cache primaire à 300 MHz qui impose une petite capacité pour conserver un temps d'accès faible. L'associativité du deuxième niveau de cache compense les problèmes de conflits du niveau primaire.

Le DEC 21164 fournit également le contrôle nécessaire à la mise en œuvre d'un troisième niveau de cache externe. Sa taille est configurable de 1 à 64 Mo. Ce cache est à correspondance directe, avec une politique d'écriture différée.

À la fréquence de 300 MHz, le DEC 21164 est confronté 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 a été amené à quatre entrées. Par ailleurs 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 au large bus mis en œuvre (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.

Le support aux systèmes d'exploitation a été considérablement amélioré. L'espace d'adressage linéaire physique a été étendu à 40 bits (soit un rapport de 64 avec celui du DEC 21064). Les tables de traduction d'adresses pour les instructions et les données ont été respectivement étendues à 48 et 64 entrées. Ces tables sont totalement associatives et implémentent un algorithme de remplacement de type *NLU*. Le champ d'identification de processus a été amené à sept bits. Les quatre modes d'exploitation sont restés identiques.

Le support multiprocesseur a également été amélioré. Deux types de protocoles sont disponibles en fonction du coût d'intégration désiré de ce processeur dans le système (protocole de type MESI[5] et par éviction de bloc dans le cas de systèmes d'entrée de gamme).

5. D'après les quatre états possibles d'une ligne de cache : *Modified, Exclusive, Shared, Invalid.*

Ce processeur inclut également comme son prédécesseur un support à l'analyse de performances par le biais de trois compteurs configurables par logiciel pour suivre certains évènements.

Le bus système vers la mémoire a une largeur de 128 bits. Des composants externes d'interfaces sont nécessaires pour communiquer avec des systèmes de type PCI ou ISA[6] . Les mécanismes de correction et de détection d'erreur restent inchangés (ECC et parité).

Ce processeur est réalisé en technologie CMOS à 0.5 μm. Il intègre 9.3 millions de transistors et il sera disponible en deux versions : 266 et 300 MHz. Les performances annoncées sont de l'ordre de 330 SPECint92 et 500 SPECfp92 pour la version à 300 MHZ.

6. *Industry Standard Architecture.*

Chapitre 3

Les architectures POWER et PowerPC

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

La première de ces architectures a donné naissance à deux processeurs que nous étudierons dans une première partie. La deuxième quant à elle est issue d'une alliance scellée en 1991, entre *Apple*, *IBM* et *Motorola*. Ces trois sociétés se sont unies pour définir et mettre au point une nouvelle famille de processeurs RISC fabriqués à la fois par *Motorola* et par *IBM* et destinés aux ordinateurs *Apple* et *IBM* dans un premier temps. Comme nous le verrons au cours de cette étude, cette alliance s'est très rapidement étendue à d'autres constructeurs.

3.1 L'architecture POWER

IBM a introduit le premier système RISC 6000 (RS/6000) en février 1990. L'ensemble de ces produits était basé sur une implémentation *multichip* de l'architecture POWER[1], qui comme son nom l'indique regroupe les principales caractéristiques des processeurs RISC : instructions de taille constante, architecture *load/store*, ensemble de registres généraux entiers et flottants, etc...

L'architecture POWER est issue d'une orientation différente des constructeurs de processeurs 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 du 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 processeur à bénéficier de cette approche (voir [1]).

Deux ans plus tard, bénéficiant de l'avancée technologique dans le domaine de l'intégration qui permis de regrouper sur le même composant l'ensemble des unités fonctionnelles du POWER, un second processeur visant les systèmes bas de gamme fut introduit (processeur *RSC* pour *RISC Single Chip*).

Le POWER2 constitue la version la plus élaborée de cette architecture grâce à un séquencement superscalaire de degré six ainsi qu'un doublement de la largeur des bus et des unités fonctionnelles par rapport au POWER1. Il en marque également l'apogée car bien qu'un POWER3 ait été initialement prévu, le projet est maintenant abandonné pour évoluer vers l'architecture PowerPC. Aussi,

1. POWER est l'acronyme de *Performance Optimization With Enhanced RISC.*

l'avenir de cette architecture semble désormais figé, l'évolution du POWER2 devant essentiellement se limiter à l'augmentation de sa fréquence d'horloge (jusqu'à 133 MHz). Les systèmes IBM seront maintenant bâtis autour de l'architecture PowerPC.

3.1.1 Le jeu d'instructions POWER

Il nous semble particulièrement intéressant de détailler quelques caractéristiques de ce jeu d'instructions qui encore maintenant (bien que certains constructeurs aient adopté certaines de ces instructions) le différencie des autres processeurs RISC.

Le reproche classique adressé au RISC vis à vis du CISC est qu'il lui faut davantage d'instructions pour exécuter une même tâche, ce qui tend à accroître la longueur des programmes et leur occupation mémoire. Pour répondre à cette objection, *IBM* a imaginé des instructions composées assurant une extension du code mais ne constituant nullement un retour au CISC.

Par exemple, on constate que l'appel d'une procédure et le retour au programme appelant, avec la sauvegarde des registres et leur restauration, sont à l'origine d'une forte expansion de code. C'est pourquoi *IBM* a introduit des *load* et *store multiple*, permettant de déplacer le contenu de plusieurs registres simultanément vers ou à partir de la mémoire et ce, en une seule instruction.

Un autre exemple est la mise à jour automatique du registre d'adresse de base sur un *load* ou un *store*. Une instruction combinée permet de supprimer les instructions d'incrémentation de l'index lors du déplacement des chaînes.
Il s'ajoute à ces instructions combinées :

- un jeu extensif d'instructions manipulant les bits ;
- des instructions combinées de multiplication-addition en virgule flottante ;
- des lectures et des écritures pour des instructions de chaînes.

On notera l'absence d'instructions combinées branchement et exécution (*conditional move*) car une nouvelle technique a été mise en œuvre grâce à l'implémentation d'un registre de condition amélioré :

- Dans le code opération de chaque instruction un bit assure une mise à jour optionnelle du registre de condition (CR : *Condition Register*). Cette mise en œuvre permet au compilateur de réorganiser le code.
- Huit registres de conditions sont disponibles. Ils forment un vaste espace de registres conditionnels, ce qui accroît d'autant les ressources.

Cet espace conditionnel est inclus dans l'unité de branchement avec les registres d'adresse de branchement. Il n'est donc pas nécessaire d'accéder aux registres généraux pour évaluer un branchement, ce qui permet de les mettre en œuvre sans retard. Par ailleurs, un registre de comptage 32 bits, le *CTR* (*Count Register*), peut servir à compter des tours de boucle par décrémentation mais il permet également de fournir une adresse cible de branchement lorsqu'il est utilisé à travers l'instruction de branchement conditionnel avec *CTR*.

Le jeu d'instructions POWER compte cinq formats principaux : des instructions de type *registre-registre*, *registre-immédiat*, branchement, saut et appel de procédure et enfin le format arithmétique

en virgule flottante. Il inclut les quatre formes courantes d'adressage (absolu, indirect, basé et indexé) plus une forme d'adressage avec pré-incrémentation (voir [1] et [4] pour plus de détails).

3.1.2 L'architecture POWER

Nous décrirons dans cette partie les processeurs POWER1 et POWER2, dans la mesure où leurs architectures sont sensiblement identiques et seuls le nombre d'instructions prises en compte, la largeur des chemins de données et le nombre des unités fonctionnelles diffèrent. Le processeur *RSC* constitue quant à lui une implémentation monocomposant du POWER1, nous ferons donc peu de remarques à ce sujet.

Généralités

D'un point de vue générale, les processeurs POWER sont multicomposants et se répartissent respectivement sur huit et neuf composants (une unité entière, une unité flottante, une unité de chargement et de séquencement des instructions, une unité d'interface avec la mémoire et quatre composants constituant le cache de données. Dans le cas du POWER1, il faut également ajouter un contrôleur d'entrées/sorties qui a été fusionné sur l'unité d'interface avec la mémoire dans le cas du POWER2).

Les pipelines

L'architecture POWER est une architecture résolument superscalaire. Le POWER1 pouvait traiter simultanément quatre instructions par cycle, le POWER2 quant à lui en traite six et en inspecte deux de plus en vue d'un éventuel branchement.

Le POWER2 a conservé les mêmes pipelines que ceux implémentés sur son prédécesseur : cinq étages pour les opérations entières, six lors d'opérations sur la mémoire et huit pour les opérations flottantes. Cependant le nombre d'unités d'exécution ajoutées à cette architecture lui permet de traiter un plus grand nombre d'instructions. Effectivement, alors que le POWER1 traite quatre instructions à chaque cycle, le POWER2 en séquence six. Ceci a bien entendu des conséquences sur la largeur des chemins de données (élargi à 256 bits dans le cas de ce dernier).

Le choix d'une gestion décentralisée des conflits a été fait, afin d'exploiter au maximum le parallélisme entre les instructions. Aussi, l'ICU (pour *Instruction Cache Unit*) 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 destiné au chargement des instructions du flot séquentiel (8 entrées sur le POWER1, 16 sur le POWER2) et l'autre destiné à recevoir les instructions issues du chemin prédit (4 entrées sur le POWER1 et 8 sur le POWER2). 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.

L'une des améliorations apportée sur le POWER2 concerne la prise en compte des instructions sur le bus. Les unités entière et flottante ne prennent sur l'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 on va le voir au chapitre suivant.

Les unités arithmétiques

Unité arithmétique entière. Sur l'architecture POWER, cette unité a un triple rôle: elle est responsable de l'ensemble des calculs arithmétiques entiers, effectue les calculs d'adresse, et 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.

La gestion des conflits de contrôle est effectuée au niveau de cette unité. Elle dispose de 32 registres de 32 bits pour effectuer l'ensemble des opérations. Le fichier de registres dispose d'un nombre supérieur de ports de lecture/écriture du fait du jeu d'instructions (adressage avec mise à jour). Ceci est particulièrement net sur le POWER2, qui du fait de son architecture superscalaire nécessite sept ports en lecture et quatre ports en écriture.

L'unité arithmétique de l'architecture POWER a été considérablement améliorée entre ces deux processeurs. 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).

Par ailleurs, il faut mentionner les unités présentes sur l'ICU responsables de l'exécution des opérations sur le registre de condition et une unité de branchement.

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 dépendances de données de type *écriture après lecture* et *écriture après écriture* (*WAR* et *WAW*). Sur les deux processeurs, 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 augmentée 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 permettant 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 mémoire mise en œuvre sur le POWER1 est restée sensiblement identique sur le POWER2 aux capacités des caches près. Le cache d'instructions du POWER2 a une taille de

32 Ko et il est directement implémenté sur l'ICU (sur le POWER1, ce cache est externe et a une taille de 8 Ko). Il est associatif par ensemble à deux voies.

Le cache de données a quant à lui une taille de 256 Ko, 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 et le test de ce cache sont physique.

Il est à noter que bien que les instructions transitent par le cache de données, elles ne font que traverser cette unité. Ceci permet de bénéficier des mécanismes de détection et de correction d'erreur dispensés par cette unité tout en minimisant le nombre de broches vers l'extérieur.

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 plus 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 Ko.

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 Mo 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.

Mécanisme de traduction d'adresses. Dans le cas du POWER2, une adresse effective (adresse issue du programme) est d'abord traduite en une adresse virtuelle puis en une adresse physique.

Les 32 bits d'adresses effectives sont convertis en une adresse virtuelle de 52 bits comme suit :

1. Utilisation des 4 bits de poids forts de l'adresse effective pour sélectionner l'un des 16 registres de segments.

2. Concaténation des 24 bits d'identification du segment fournis par le registre de segment accédé, avec les 28 bits restants de l'adresse effective.

On obtient alors une adresse virtuelle sur 52 bits. Celle-ci est alors convertie en une adresse physique de 32 bits en utilisant une table de pages indexée par l'intermédiaire de deux fonctions de hachage (ce mécanisme est décrit dans [3]).

Caches de traduction d'adresses. Sur les deux processeurs, deux caches de traduction d'adresses instructions et données ont été implémentés ; seule différence entre les deux processeurs, la capacité de ces caches. Sur le POWER2, les TLB instructions et données ont une taille respective de 128 et 512 entrées, soit le quadruple de la taille des caches du POWER1. Ils sont tous deux associatif par ensemble à deux voies. La mise à jour des TLB lors de défaut est gérée par un algorithme de type *LRU*.

Support multiprocesseur. L'architecture POWER n'offre aucun support matériel à la mise en œuvre d'une mémoire partagée (à la différence de l'architecture PowerPC). Les processeurs sont cependant utilisés dans les systèmes *POWERparallel* tels que le *SP1* ou *SP2*[2] qui lui succède. Ces systèmes sont utilisés dans le domaine scientifique ainsi que dans des architectures de type *client/serveur*.

Le SP2 a été annoncé en avril 1994 et comporte de 2 à 128 processeurs en standard et jusqu'à 512 sur demande spéciale. Des processeurs POWER2 remplacent les POWER1 précédemment utilisés.

Les systèmes *POWERparallel* sont organisés selon le principe du parallélisme à mémoire distribuée avec échange de messages et réseau d'interconnexion.

3.2 L'architecture PowerPC

L'architecture PowerPC est issue de l'alliance scellée en septembre 1991 entre *IBM*, *Apple* et *Motorola*. Cette alliance comprend cinq axes principaux de recherche :

- la technologie orientée objet (création de *Taligent* par *Apple* et *IBM*, une société indépendante chargée de développer et de diffuser un noyau de système d'exploitation orientée objet, ou *OOOS* pour *Object Oriented Operating System*) ;
- le multimédia (création de la filiale *Kaleida* par *Apple* et *IBM*, dont l'objectif est de créer et de diffuser des technologies multimédia pour les applications grands public) ;
- l'interconnexion et les réseaux (*Apple* et *IBM* travaillent en commun pour assurer l'intégration des *Macintosh* dans les systèmes et réseaux *IBM*) ;
- l'environnement de systèmes ouverts (*PowerOpen Environment*, développement d'un Unix pour systèmes ouverts qui supportera à la fois les applications *Macintosh* et *AIX* de *IBM* et de *Apple*) ;
- le développement de microprocesseurs.

2. *Scalable POWERparallel.*

Ce dernier point est à l'origine de l'architecture PowerPC, qui selon le souhait des intervenants devait s'imposer mondialement et s'opposer à la suprématie d'*Intel*.

Issue 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 contrôleur (403 GA pour *IBM* et MPC 505 pour *Motorola*) au processeur haut de gamme (PowerPC 620) ;
- être doté d'une architecture suffisamment simple pour supporter une haute fréquence de fonctionnement ;
- minimiser les effets qui empêchent la conception d'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.

Nous introduirons dans une première partie quelques caractéristiques de mise en œuvre de cette architecture à travers l'étude du PowerPC 601. Puis nous aborderons l'étude des processeurs PowerPC 603, 604 et le 602 récemment annoncé. Enfin, nous présenterons brièvement les contrôleurs destinés aux applications enfouies avant d'aborder dans un ultime chapitre de perspectives le processeur PowerPC 620.

3.2.1 Le jeu d'instructions PowerPC

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 des processeurs POWER à l'exception de celles exécutées peu fréquemment ou jugées trop complexes. À titre d'exemple, nous détaillons quelques unes de ces instructions afin que le lecteur puisse prendre conscience de la complexité de mise en œuvre de certaines instructions sur les processeurs POWER. On peut citer :

- l'instruction de *lecture avec mise à jour*. Cette instruction charge un opérande dans un registres cible *Rt* et met à jour l'adresse dans le registre *Ra*. Que se passe-t-il quand *Ra* est égal à *Rt*?
  L'architecture POWER résout ce conflit en imposant que la donnée chargée dans le registre soit l'opérande lu dans la mémoire. L'une des conséquences, est que le matériel doit tester à chaque fois si *Ra* et *Rt* sont les mêmes, et, dans ce cas, gérer la situation.
  Dans l'architecture PowerPC, cette nécessité a été supprimée. Le compilateur est utilisé pour prévenir les cas où $Ra = Rt$ en n'utilisant pas de lecture avec mise à jour lorsqu'il y a égalité.

- certaines instructions de l'architecture POWER nécessitent trois ports en lecture (exemples : *rlmi : Rotate Left then Mask Insert; rrib : Rotate Right and Insert Bit; maskdir : Mask Insert from Register*). La nécessité de disposer d'un opérande supplémentaire exige la mise en œuvre d'un port de lecture additionnel qui accroit la taille du banc de registres et complexifie le mécanisme de renommage.

– de même pour des instructions de manipulation de chaînes de caractères ou encore les instructions arithmétiques qui manipulent le registre *MQ* (*Multiplier and Quotient*). Effectivement, dans une séquence de multiplication et de division, ce registre unique, provoque un goulot d'étranglement.

Au total, 36 instructions ont été éliminées de l'architecture POWER afin « d'alléger » la complexité de cette architecture. D'un autre coté, le jeu d'instructions PowerPC a été étendu pour qu'il réponde à de nouvelles fonctionnalités :

– introduction d'instructions de calcul sur 64 bits ;

– introduction d'instructions de calculs arithmétiques flottants simple précision ;

– extension des instructions de lecture/écriture ;

– etc...

La figure 3.1 schématise l'évolution des jeux d'instructions entre les architectures POWER et PowerPC.

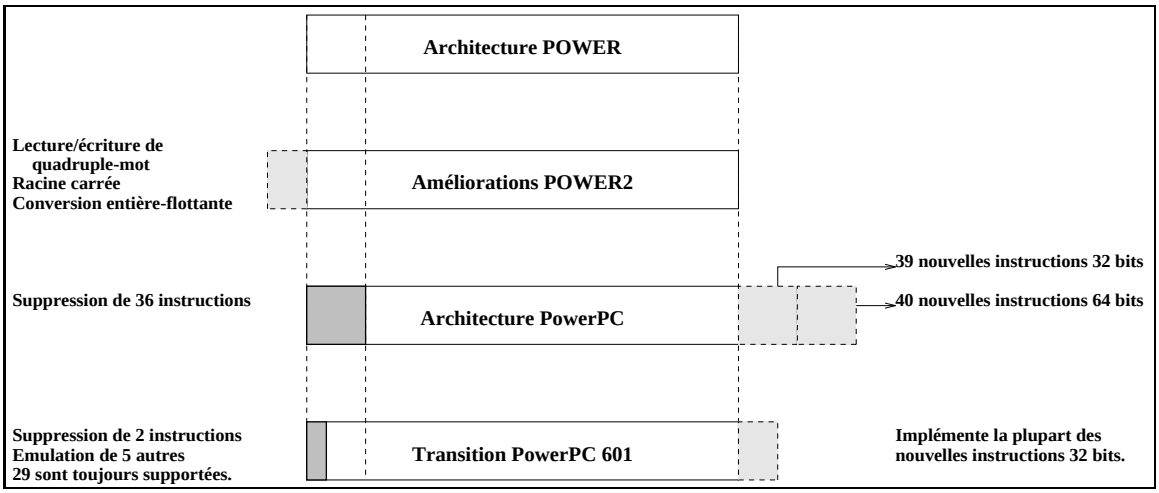

FIG. 3.1 - *Relation entre les jeux d'instructions POWER et PowerPC*

3.2.2 Le PowerPC 601

Ce processeur est le premier à mettre en œuvre l'architecture PowerPC (consulter [3] pour plus de détails). Fortement inspiré des processeurs POWER, ce processeur constitue une passerelle entre ces deux architectures. Sur les 36 instructions devant être éliminées, 29 sont toujours supportées par matériel, cinq sont émulées par logiciel et deux ne sont plus supportées du tout. Les applications existantes sur le POWER peuvent s'exécuter directement sur le PowerPC 601 sans être recompilées. Ce processeur assure ainsi une période de transition aux utilisateurs avant que les futures générations (pour lesquelles la recompilation des applications sera nécessaire pour éviter l'émulation des instructions éliminées) soient disponibles.

Ce processeur n'implémente pas la totalité des instructions définies par la norme PowerPC. On peut citer à titre d'exemple les instructions manipulant des données sur 64 bits destinées à être implémentées sur le PowerPC 620, quelques instructions manipulant des opérandes sur 32 bits telles que la racine carrée flottante, jugée coûteuse en silicium sur un processeur destiné à un marché d'entrée de gamme (voir la figure 3.1 qui illustre l'évolution du jeu d'instructions PowerPC).

Architecture générale

On a privilégié le traitement simultané de plusieurs instructions. Le PowerPC 601 est donc un processeur superscalaire de degré trois. Il supporte par ailleurs des mécanismes de séquencement des instructions partiellement dans le désordre.

La structure superscalaire de ce processeur s'organise autour de l'unité de lancement des instructions. Cette unité contrôle trois unités fonctionnelles indépendantes :

- une unité de branchement ;
- une unité arithmétique entière ;
- une unité arithmétique flottante.

Les instructions sont distribuées simultanément au sein de ces unités et peuvent être exécutées en parallèle. Les unités d'exécution disposent de leurs propres ressources et sont synchronisées de façon transparente pour le logiciel.

L'unité de lancement des instructions contient une file d'instructions, une unité de *dispatch* et une unité de branchement. Elle gère le flot des instructions et répartit les instructions dans les trois unités d'exécution. Il est à noter la possibilité d'émettre les instructions dans un ordre différent de celui du flot d'instructions. Cependant, l'ordre de chacun des trois flots d'instructions séquentiels sur les trois unités fonctionnelles est respecté[3].

La file d'instructions. Comme dans les processeurs POWER, l'unité de lancement des instructions précharge les instructions à partir du cache dans la file d'instructions. Ce préchargement est fait par blocs d'au maximum huit mots, ce qui correspond à la granularité minimale de stockage du cache (une ligne). Les instructions vont du haut de la file (Q7) vers le fond (Q0) par décalages successifs. Alors que la partie amont de la file d'instructions (Q4-Q7) est une simple FIFO (*First In First Out*), les instructions peuvent être retirées des quatre éléments de la partie aval de la file (Q0-Q3).

Unité de branchement (BPU). Contrairement aux autres unités fonctionnelles du PowerPC 601 qui sont relativement semblables à celles de l'IBM POWER, l'unité de branchement possède une architecture totalement nouvelle. Son rôle est de chercher dans la partie avale de la file d'instructions (Q0-Q3) les instructions de branchement et de les traiter le plus tôt possible dans le but de ne pas retarder le séquencement des instructions de calcul.

3. Le terme *out-of-order* utilisé dans de nombreuses documentation de processeur est à relativiser. Effectivement, sur le PowerPC 601 l'émission en désordre des instructions se fait au niveau de l'étage de décodage et non à celui d'exécution, comme c'est le cas sur le PowerPC 604.

La BPU contient un additionneur qui lui permet de calculer l'adresse cible d'un branchement et les valeurs de trois registres, LR (*Link register*) utilisé lors des appels de procédures, CTR (*Count Target Register*) qui contient l'adresse cible et CR (*Count Register*). La présence de cet additionneur supplémentaire comble une lacune du POWER et permet au PowerPC 601 d'exécuter simultanément un calcul d'adresse de branchement et une opération entière; sur le POWER, l'absence de cet additionneur retardait le séquencement de certaines instructions de branchement.

Le PowerPC 601 met en œuvre la technique dite de *Branch Folding* qui consiste à retirer physiquement l'instruction de branchement de la file d'instructions au moment de son émission vers son unité d'exécution. Si le branchement est pris, les instructions en séquence sont évacuées de la file tandis que de nouvelles instructions du groupe cible sont chargées. Si le branchement n'est pas pris, le flot d'instructions continue son exécution en ignorant l'instruction de branchement.

Lors d'un branchement conditionnel, si la condition de branchement ne peut être résolue, une prédiction statique est utilisée afin de permettre le préchargement des instructions à partir de l'adresse cible.

Les instructions issues de la prédiction sont émises vers les unités fonctionnelles mais ne sont achevées qu'après la résolution du branchement. Une instruction devant être exécutée dans l'unité de branchement ne sera exécutée qu'après la résolution du branchement prédit. Par contre, une instruction devant être exécutée par l'unité entière ou par l'unité flottante peut progresser dans le pipeline jusqu'à l'étage d'écriture du registre résultat et ne modifie ainsi pas de façon erronée l'état de la machine.

La prédiction statique de branchement permet, en cas de réussite, de ne pas interrompre le flot d'instructions à chaque branchement. En cas de prédiction erronée, la file d'instructions est vidée et les instructions amorcées annulées, puis le flot d'instructions reprend à la bonne adresse.

La prédiction statique mise en œuvre sur le PowerPC 601 est faite uniquement sur la direction du branchement : le branchement est prédit *pris* si son adresse cible est située en arrière dans le programme par rapport à l'adresse du branchement. Afin d'améliorer les performances, l'utilisateur a la possibilité d'inverser cette stratégie au niveau d'une application. Cette prédiction est à un niveau (c'est à dire qu'un branchement ne peut être prédit que si le précédent est résolu). Par ailleurs si l'adresse cible doit être calculée juste avant le branchement, aucune prédiction n'est possible.

Il est à noter qu'en découplant l'exécution des branchements du reste du flot d'instructions, le PowerPC 601 dépend moins de la pertinence de la prédiction de branchement que les microprocesseurs exécutant leurs instructions dans l'ordre; dans de nombreux cas, le branchement peut être résolu très en avant dans le pipeline (codes conditions connus, boucles, etc...).

Unités d'exécution

Les trois unités d'exécution sont quasi-indépendantes, ce qui permet au processeur de lancer les exécutions dans le désordre et d'effectuer en parallèle plusieurs opérations comme le calcul de l'adresse cible d'un branchement en même temps qu'une instruction entière et une instruction flottante. Il est à noter que les instructions sont exécutées dans l'ordre sur chacune des unités.

Unité entière. Son rôle est d'exécuter toutes les instructions entières ainsi que tous les accès à la mémoire. Toutes les instructions entières sont pipelinées et une instruction peut être émise tous

les cycles (exceptions : multiplications et divisions entières, lectures et écritures multiples). Cette unité contient un banc de 32 registres entiers à usage général.

L'unité entière fait appel à l'unité de gestion de la mémoire (*MMU*) pour traduire l'adresse virtuelle en adresse physique lors des accès à la mémoire.

Unité flottante. Le PowerPC 601 est destiné aux systèmes de bas ou de milieu de gamme, son unité flottante est plus simple et moins efficace que celle de l'IBM POWER.

Elle supporte les formats flottants *IEEE 754* en simple et double précision. Le banc de registres est constitué de 32 entrées de 64 bits.

Comme dans les processeurs POWER, l'unité flottante du PowerPC 601 intègre la primitive de multiplication-addition [4].

Les pipelines d'exécution des instructions entières et flottantes ont une profondeur respective de cinq et six étages. Le pipeline entier est identique à celui du POWER à ceci près que l'étage de prédécodage des instructions a disparu. Le pipeline flottant a un étage commun avec le pipeline entier (préchargement des instructions). Par ailleurs un étage tampon par lequel transitent toutes les instructions flottantes ajoute un cycle de latence mais permet lorsque l'opération prend plus d'un cycle de ne pas interrompre le flot d'émission des instructions.

De la même façon que l'unité de branchement, l'unité flottante peut accéder aux instructions dans la partie basse de la file d'instructions ; ceci permet d'exécuter des instructions flottantes en avance par rapport à leur place dans le flot d'instructions séquentiel si elles ne dépendent pas d'instructions en cours.

Il est à noter que le PowerPC 601 n'intègre pas de mécanisme de renommage de registres.

Hiérarchie mémoire

Le PowerPC 601 tranche avec la plupart des autres processeurs étudiés jusqu'à maintenant en intégrant un cache de premier niveau unifié d'une capacité de 32 Ko et associatif par ensemble de huit. Ce degré d'associativité est particulièrement élevé pour un processeur. La stratégie de remplacement des lignes d'un ensemble est de type *LRU*. Ce cache ne permet qu'un seul accès par cycle. Du fait des conflits d'accès entre les instructions et les données un système de priorité fixe a été instauré. Ainsi, quand l'unité de branchement génère le chargement de nouvelles instructions, ce dernier n'est réalisé que si aucun accès à une donnée par l'unité entière et aucun accès à la mémoire par le cache n'est suspendu ou en cours.

De même, lorsqu'une ligne d'un secteur est chargée à partir de la mémoire, le cache essaye de charger, avec une priorité faible, la seconde ligne du secteur.

Cependant, deux ports sur les étiquettes permettent d'effectuer un cycle de recherche en même temps qu'un accès à un élément du cache.

La politique d'écriture mise en œuvre sur ce cache est configurable (simultanée ou différée) au niveau de chaque page. Afin d'améliorer les performances du système, deux files (lecture et écriture) servent de tampon avec la mémoire.

Il est à noter que le PowerPC 601 n'intègre pas le contrôle nécessaire à l'ajout d'un cache secondaire externe. Celui-ci est donc effectué par un contrôleur externe au processeur. Afin de diminuer au mieux la latence des transferts, le PowerPC 601 possède un mode de transferts groupés

4. À noter que comme dans le cas du POWER, cet opérateur n'est pas conforme à la norme IEEE 754 du fait de la suppression d'une étape de calcul d'arrondi entre les deux opérations.

(*Burst*) qui permet d'effectuer le transfert d'une seule adresse pour tous les transferts de données d'une ligne de cache primaire ou de le pipeliner (adresse + donnée).

Bien que la définition de l'architecture PowerPC soit de type 64 bits, le PowerPC 601 est une implémentation 32 bits. Il a un bus externe de données de 64 bits et des bus internes de largeur variant de 32 à 256 bits.

Support des systèmes d'exploitation

L'architecture PowerPC conserve le même format mémoire que celui mis en œuvre sur les processeurs POWER.

Le PowerPC 601 utilise donc un espace d'adressage segmenté paginé. L'espace d'adressage virtuel est de quatre peta-octets (2^{52}) et la taille des pages est de 4 Ko. Cependant, l'architecture PowerPC 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 Ko à 8 Mo.

Le mécanisme de traduction d'adresse utilisé sur le PowerPC 601 est globalement identique à celui mis en œuvre sur l'architecture POWER (sélection d'un registre de segment, adresse logique sur 52 bits, recherche dans la table des pages à partir de deux fonctions de hachage, obtention d'un numéro de page physique sur 32 bits). Cependant, l'introduction du concept de bloc qui permet de disposer d'un large espace mémoire contigu nécessite un mécanisme de traduction d'adresse particulier (voir [3]). En effet, les descripteurs de blocs sont dans des registres au lieu d'être dans une table en mémoire. Ces registres BAT (*Block Address Translation registers*) sont regroupés en paires, la partie haute contenant l'adresse virtuelle du bloc et la partie basse son adresse physique. Ce mécanisme est prioritaire sur la pagination classique. Si l'adresse logique correspond au contenu d'une des quatre paires de registres BAT, le mécanisme de traduction d'adresse de bloc est enclenché[5]. De ce fait, la segmentation est alors inactive.

Chaque paire de registres BAT[6] contient les adresses virtuelle et physique du début de bloc ainsi que sa taille. Ils sont accessibles uniquement en *mode superviseur* et ne sont pas vidés lors de l'exécution d'instructions d'invalidation de TLBs. Leur manipulation est à la charge du système d'exploitation. Ces mécanismes sont hérités des processeurs *Motorola 88110*.

Caches de traduction d'adresses. Le PowerPC 601 possède trois caches de descripteurs de pages ou de blocs :

- un UTLB de 256 entrées, contenant des descripteurs de page de données et d'instructions, et associatif par ensembles de deux voies. Un algorithme de type *LRU* est utilisé pour le remplacement de ces éléments.
- un petit ITLB de quatre éléments qui ne peut contenir que des descripteurs de pages d'instructions. La présence de ce deuxième TLB permet d'effectuer en parallèle la traduction d'une page d'instructions et d'une page de données. En cas de défaut de ce ITLB, il est rechargé à partir de l'UTLB.
- le troisième TLB du PowerPC 601 est le BTLB (Bloc TLB). Il est constitué des registres BAT. Sa gestion est totalement différente de celle des deux autres. Les registres BAT font partie intégrante du contexte d'exécution d'un processus.

5. Cette priorité n'est pas observée lorsque le segment adressé appartient à l'espace des entrées/sorties.
6. Le nombre et le format des registres BAT du PowerPC 601 ne constituent pas une norme PowerPC.

Sécurité. Les concepts de sécurité sont identiques à ceux définis sur l'architecture POWER (contrôle des accès au niveau des segments et des pages). La séparation *superviseur / utilisateur* se fait au niveau des segments ou des blocs. Chaque processus utilisateur dispose de 16 segments dont deux sont réservés au noyau (accessible en *mode superviseur*). À noter cependant qu'il est possible d'effectuer des accès à la mémoire qui ne supportent aucun contrôle. Il suffit pour cela de désactiver tous les mécanismes de traduction d'adresses.

Support multiprocesseur

L'une des différences essentielles avec l'architecture POWER est l'introduction d'un support aux systèmes multiprocesseurs à mémoire partagée. Nous développons quelques uns des concepts implémentés dans la mesure où ils constituent un apport important de cette architecture.

Dans un système multiprocesseur à mémoire partagée, il est nécessaire de disposer de protocoles de cohérence de caches qui assurent l'exclusion mutuelle en écriture et la mise à jour des différents caches (primaires et secondaires), afin que chaque processeur puisse lire des données cohérentes.

Le PowerPC 601 dispose d'un support matériel pour l'implémentation de protocoles de cohérences de caches ainsi que de primitives de synchronisation qui permettent l'exécution, de façon atomique, d'accès à la mémoire. Le protocole de cohérence de cache employé est le protocole à invalidation sur écriture nommé MESI.

Accès à la mémoire partagée. Le PowerPC 601 dispose d'instructions spécifiques pour accéder à des variables partagées. L'instruction *LWARX* accède à la donnée et verrouille la ligne de cache qui ne sera à nouveau accessible qu'après son écriture par l'instruction *STWCX*. La combinaison des deux instructions permet ainsi d'effectuer une lecture/écriture atomique.

Ordre des lectures/écritures. La mise en œuvre de multiprocesseurs à mémoire partagée pose le problème de l'ordre des accès à la mémoire par des processeurs distincts.

Il n'est pas donné de modèle pour ordonner les accès à la mémoire pour le PowerPC 601. Cependant, ce processeur offre la possibilité d'implémenter un modèle d'accès à forte rigidité, grâce au mode *Write Through* d'écriture en mémoire, ainsi que les trois modes proposés dans la norme Sparc version 8 (voir chapitre 4.1.1 page 66).

Quand le modèle d'ordre partiel des écritures est choisi, le PowerPC 601 peut spécifier, à l'aide des deux instructions *SYNC* et *EIEIO* quand l'ordre des accès mémoire doit être maintenu. L'instruction *SYNC* attend que toutes les instructions qui la précèdent soient achevées avant de commencer son exécution alors que l'instruction *EIEIO* n'assure une forte rigidité que sur les instructions d'accès à la mémoire.

Caractéristiques technologiques

Commercialisé dans un premier temps aux fréquences de 50 et 66 MHz, ce processeur atteint désormais 80 MHz. Les dernières annonces concernent un PowerPC 601+. Ce processeur, identique au PowerPC 601, résulte d'un passage à une technologie CMOS à 0.5 μm (au lieu de 0.6 μm) qui devrait lui permettre d'atteindre une fréquence de 100 MHz. Les performances s'accroissent de 25% et la surface de silicium est réduite du tiers pour une consommation divisée par deux (elle passe de 8 Watt à 4 Watt).

3.2.3 Les processeurs PowerPC 603, 602 et 604

Après avoir survolé les caractéristiques de l'architecture PowerPC à travers l'étude du PowerPC 601, nous présentons maintenant diverses implémentations de cette architecture.

Le PowerPC 603

Annoncé en 1993, il est destiné au marché des systèmes portables (*note-books, palmtops*). L'effort a essentiellement porté sur la faible consommation de ce processeur, et la mise en veille de certains de ses organes (processeur, disque, unités d'exécution). Ce processeur a cinq unités d'exécution (entière, flottante, branchement, lecture/écriture et système). Il peut séquencer trois instructions par cycle (deux instructions plus un 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 processeur 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 taille 8 Ko. Ces deux caches sont associatifs par ensemble à deux voies, physiquement adressés et testés. L'algorithme de remplacement des lignes est de type *LRU*.

Bien que le PowerPC 603 implémente un protocole de cohérence à trois états, sous-ensemble du protocole MESI, il ne peut pas être utilisé dans des systèmes multiprocesseurs symétriques. Le protocole à trois états permet de maintenir la cohérence lors de transfert de type DMA.

Le PowerPC 603 implémente le schéma de gestion de la mémoire virtuelle précédemment détaillé. Ce schéma de traduction fait appel à deux caches de traduction d'adresses de 64 entrées, associatifs à deux voies pour les instructions et les données et à un TLB de quatre entrées totalement associatif destiné à la traduction des adresses de blocs.

La largeur des bus sur ce processeur est de 32 ou 64 bits et sa consommation typique est de 2.5 W. En plus de l'inhibition de certaines unités, ce processeur dispose de trois modes de fonctionnement configurables par logiciel et qui permettent de mieux gérer sa consommation.

Réalisé en technologie CMOS à 0.65 μm avec quatre couches de métal, ce processeur est séquencé à la fréquence de 80 MHz. Une évolution de ce composant a été récemment annoncée : le PowerPC 603e. Ce dernier conserve la même architecture avec cependant un cache de taille double (16 Ko), associatif par ensemble à quatre voies. Séquencé à la fréquence de 100 MHz, il contient 2.6 millions de transistors (1.6 millions pour le 603). Les performances attendues sont de 120 SPECint92 et 105 SPECfp92 (75 SPECint92 et 85 SPECfp92 pour le PowerPC 603 à 80 MHz).

Le PowerPC 602

Annoncé tout récemment, ce processeur est un dérivé du PowerPC 603 et devrait être disponible d'ici la fin de l'année 1995. Il vise le marché des processeurs embarqués haut de gamme tels que les consoles de jeux, les *set-top boxes* et les *Personal Digital Assistant*. Il constitue un compromis entre performances et coût de fabrication (divisé par trois par rapport au PowerPC 603 avec une consommation réduite de moitié). Pour atteindre un tel résultat, les changements par rapport au 603 concernent la réduction de la taille des caches, la suppression du séquencement superscalaire et la simplification des bus systèmes. Cependant, bien que beaucoup de ces changements aient des conséquences sur les performances, les caches ont été modifiés pour gérer plus rapidement les écritures et la latence des multiplications entières a également été améliorée. Ces changements

devraient permettre a ce processeur de s'imposer dans un environnement de calcul graphique intensif (le concepteur de jeux *3DO* a déjà choisi ce processeur pour équiper sa prochaine génération d'accélérateur graphique).

Comme on l'a dit précédemment, le PowerPC 602 est limité à une exécution scalaire des instructions. Ceci simplifie énormément les mécanismes associés au séquencement des instructions, à la prédiction des branchements et à l'exécution des instructions (réduction de moitié du nombre de ces composants). Cependant, sous certaines conditions, ce processeur peut exécuter deux instructions grâce à la technique du *Branch folding* conservée du 603 et du 601.

D'autres simplifications ont été effectuées au niveau de certaines unités d'exécution :

- l'unité flottante ne calcule que des valeurs en simple précision (les calculs double-précision sont émulés par logiciel) ;
- l'unité de lecture/écriture a été simplifiée en supprimant le support matériel aux instructions complexes de manipulation de chaînes de caractères.

Il est à noter cependant que l'unité entière a été améliorée par rapport à celle du 603. Cette amélioration vise directement les applications multimédia où de nombreuses multiplications sont effectuées.

La hiérarchie mémoire a elle aussi été considérablement simplifiée à travers la suppression des tampons de lecture/écriture entre les caches et le sous-système mémoire, ainsi que la réduction de la capacité des caches (4 Ko pour les instructions et les données). Dans le domaine d'application visé, cette réduction de capacité devrait avoir des effets limités. Du fait de la suppression des tampons de lecture/écriture, il est déterminant que les caches mis en œuvre soient non bloquants.

Le protocole de cohérence sur les caches est semblable à celui du PowerPC 603. Ce processeur ne peut donc pas être utilisé dans des systèmes multiprocesseurs.

Les caches de traduction d'adresses d'instructions et de données (TLB) ont chacun une taille de 32 entrées. Chacun de ces TLB inclut quatre entrées de blocs capables de mapper un espace mémoire contigu de 128 Ko à 256 Mo.

Le PowerPC 602 implémente un bus multiplexé de 32 bits pour les adresses et de 64 bits pour les données, ceci afin de réduire la connectivité et la taille de la puce.

La dernière caractéristique héritée du PowerPC 603, concerne la gestion de l'alimentation de certaines unités qui peuvent être inhibées quand elles ne sont pas utilisées. Trois modes de fonctionnement économiques sont également disponibles.

Réalisé dans la même technologie que le PowerPC 603, ce processeur est séquencé à 66 MHz. Il devrait fournir des performances de l'ordre de 65 Dhrystone.

Le PowerPC 604

Annoncé au premier trimestre 1994, les premiers systèmes à base de PowerPC 604 ont vu le jour en juin 1995 (système *Power Macintosh 9500*). De nombreux aspects du 603 et du 601 auxquels il succède ont été améliorés afin d'augmenter les performances de ce processeur. Ils concernent :

- le séquencement superscalaire de quatre instructions par cycle ;
- l'intégration d'une seconde unité arithmétique entière ;

- l'utilisation de méthodes modernes de gestion du séquencement (renommage de registres, exécution dans le désordre) ;
- la structure des caches ;
- et enfin l'utilisation d'une technologie qui permet d'atteindre une fréquence de 100 MHz.

Ce processeur vise le marché des stations de travail (IBM) ainsi que les PC. Il équipe également la seconde génération de Power Macs d'*Apple* et s'inscrit comme un concurrent direct du Pentium avec des performances attendues de 160 SPECint92 et 165 SPECfp92 (à 100 MHz).

Du point de vue du séquencement des instructions, le 603 et le 604 ont des caractéristiques similaires dans la mesure où ils ont été développés en parallèle. Le PowerPC 604 inclut une unité arithmétique entière supplémentaire et combine le multiplieur et le diviseur dans une même unité entière complexe. Ceci permet de disposer de deux unités pour les opérations classiques (en un cycle) et d'une unité pour des calculs à latence plus longue (multiplication/division).

Les instructions sont chargées à partir du cache, au rythme de quatre instructions par cycle, dans un tampon de prédécodage de quatre entrées. Puis elles transitent dans un tampon de séquencement où elles subissent un renommage des registres. Cette étape supplémentaire particulièrement importante pour la suppression des conflits (de type *WAR* et *WAW*), ajoute un étage au pipeline classique. Le mécanisme de renommage inclut 12 registres entiers et 8 registres flottants supplémentaires. À la suite de cette étape, jusqu'à quatre instructions peuvent être envoyées vers chacune des unités fonctionnelles. Les instructions non émises le sont au cycle suivant si les conflits le permettent. Chacune des unités fonctionnelles est équipée d'une station de réservation de deux entrées. Les stations des unités relatives aux branchements, aux opérations de lecture/écriture de la mémoire ou aux opérations flottantes sont de type FIFO. En ce qui concerne les unités entières, ces stations supportent l'émission dans le désordre des instructions.

Le PowerPC 604 est le premier de cette famille à mettre en œuvre une prédiction de branchement dynamique. Cette prédiction s'appuie sur deux structures :

- un cache d'adresse de branchement cible de 64 entrées, totalement associatif qui conserve les adresses cibles de branchements récents prédits pris.
- une table d'historique des branchements de 512 entrées. Cette table utilise un algorithme à deux bits (algorithme de Smith) pour prédire le résultat du branchement.

Par ailleurs, l'architecture du PowerPC 604 supporte l'exécution spéculative des instructions. Leurs résultats sont soit stockés dans les registres de renommage soit dans la file d'écriture vers le cache. Les données seront réellement mises à jour dans le cache au moment où le résultat de la prédiction sera déterminé.

Le PowerPC 604 met également en œuvre une unité flottante double-précision. Cette unité joue un rôle capital dans l'obtention de performances sur des applications manipulant ce type de données. Cependant, dans la mesure où ces données sont stockées sous forme double précision en mémoire, un cycle d'accès supplémentaire est nécessaire en lecture pour tester le type et si besoin convertir la donnée au format approprié au moment de son écriture dans le cache.

L'organisation des caches a été adaptée au niveau de performances de cette architecture. Les caches d'instructions et de données ont chacun une taille de 16 Ko et sont associatifs par ensemble à quatre voies et physiquement adressés. Ils implémentent une politique de remplacement des lignes de type *LRU* et une protection par parité.

Un mécanisme de files permet de découpler les accès vers la mémoire en lecture (le cache reste ainsi accessible) et en écriture (exécution spéculative et mise à jour du cache différée). À la différence du 603 (ou du 602), ce processeur peut être intégré dans des systèmes multiprocesseurs. Il implémente pour cela un protocole de cohérence de cache de type *MESI* et un double-port d'accès aux étiquettes de manière à résoudre le problème des conflits lors des cycles d'espionnage.

Le 604 inclut deux tables de traduction d'adresses séparées pour les instructions et les données, associatives à deux voies et chacune de 128 entrées.

Réalisé en technologie 0.65 μm à cinq couches de métal, ce processeur contient 3.6 millions de transistors. Il est annoncé en trois versions 100, 120 et 133 MHz.

3.2.4 Les microcontrôleurs d'architecture PowerPC

C'est au printemps 1994 que *Motorola* a introduit sa première famille de microcontrôleur RISC, sur 32 bits, appelée *RMCU 505* (*RISC Microcontroler Central Unit*). Cette famille est fondée sur l'architecture PowerPC, et, en raison de ses performances élevées, est destinée aux applications enfouies haut de gamme.

Le premier élément de cette famille est le MPC 505 dont la performance est de 25 Mips pour un circuit séquencé à 25 MHz (et de 46.4 Dhrystone). Elle devrait être portée à 40 Mips fin 95 avec un circuit à 40 MHz (à titre comparatif, le 68300, le plus puissant des microcontrôleur de *Motorola* atteint 5 Mips.)

IBM a développé également de son coté un microcontrôleur d'architecture PowerPC, le *PPC 403 GA*, séquencé à 33 MHz et destiné au même marché (performance de l'ordre de 39 Dhrystone).

Ces deux composants marquent l'introduction de l'architecture PowerPC dans le domaine des systèmes embarqués et illustrent la politique commerciale de ce consortium décidé à être présent sur tous les secteurs de marché.

Les deux composants ont des caractéristiques communes et correspondent à une simplification de l'architecture PowerPC afin de diminuer les coûts de production. À titre comparatif, le PowerPC 601 contient 2.8 millions de transistors, le 603, 1.6 millions, le RMCU 505 en contient 585 000. Bien que ces deux processeurs puissent exécuter une instruction entière et un branchement par cycle, ils ne peuvent en fait émettre qu'une instruction à chaque cycle. Tous deux disposent d'une unité arithmétique entière et d'une unité de branchement, le 505 inclut par ailleurs une unité flottante par opposition au 403 qui émule ces instructions.

Le RMCU 505 dispose d'un cache d'instructions et de données de 4 Ko. Le 403 implémente quant à lui un cache d'instructions de 2 Ko et un cache de données de 1 Ko. La faible capacité de ces caches est compensée par un degré d'associativité de deux.

Par souci de simplification de l'architecture, aucun de ces deux composants n'intègrent d'unité de gestion de la mémoire. Par contre, ils incluent des caractéristiques spécifiques au domaine des processeurs 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 leur intégration dans les systèmes.

Par ailleurs, le consortium PowerPC met l'accent sur l'environnement de développement de logiciels autour de ces processeurs. Ils intègrent des capacités étendues de traçage de code en temps réel, des broches supplémentaires permettent de suivre les changements d'adresses, d'autres options incluent le contrôle du processeur (point d'arrêt, démarrage, etc...), des accès aux registres et à la mémoire ainsi qu'un outil de mise au point.

IBM a prévu au moins deux autres processeurs de ce type : le 401 destiné aux systèmes à très bas coûts et le 405 avec des unités d'exécution supplémentaires et des performances supérieures au 403.

3.2.5 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 processeur 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. On retrouve au sein de ce processeur des caractéristiques déjà mises en œuvre sur le PowerPC 604. Cependant, l'accroissement de certaines fonctionnalités telles que unités fonctionnelles, hiérarchie mémoire, largeur des bus et autres contribuent à l'augmentation des performances.

Le PowerPC 620 a une architecture superscalaire. Le chargement des instructions se fait au rythme de quatre instructions par cycle. Jusqu'à quatre instructions peuvent être émises vers les six unités fonctionnelles implémentées. L'émission des instructions est faite dans l'ordre, leur exécution pouvant être effectuée dans le désordre grâce aux files d'instructions précédant chacune des unités (structure identique au PowerPC 604). Ces files d'instructions permettent d'émettre les instructions avant que leurs opérandes soient disponibles. Elles contribuent ainsi à réduire le taux de rupture du pipeline. Les trois unités d'exécution entière (deux pour les opérations simples et une pour les opérations complexes) ont chacune une file d'instructions à deux entrées. L'unité de lecture/écriture dispose d'une file de trois entrées, l'unité de branchement et l'unité flottante sont respectivement précédées par des files de quatre et deux entrées. Par souci de simplification de l'architecture, les instructions au sein des files des unités de branchement et arithmétiques flottantes sont exécutées dans l'ordre.

Le PowerPC 620 utilise un pipeline classique de cinq étages. Pour prédire les branchements, il s'appuie sur quatre techniques :

- un cache d'adresse de branchement cible de 256 entrées, associatif à deux voies. Si ce dernier contient l'adresse de chargement, le branchement est alors prédit pris et son contenu est utilisé pour charger le flot d'instructions au cycle suivant.

- lors d'un branchement conditionnel relatif au registre de condition, la prédiction est faite sur la valeur de ce registre si elle est déjà disponible. Il est à noter qu'un mécanisme de renommage est aussi mis en œuvre sur ce registre.

- pour prédire l'adresse de retour de procédure, une pile de huit entrées est implémentée.

- la dernière prédiction repose sur une table d'historique de 2 K entrées à correspondance directe. Cette table s'appuie sur une prédiction à deux bits et est mise à jour lorsque l'instruction de branchement est exécutée. Cette table contrôle par ailleurs la mise à jour du cache d'adresse cible : si le branchement est prédit pris lors de sa prochaine occurrence, il est ajouté au cache, dans le cas contraire il est supprimé.

Une possibilité existe d'invalider la prédiction dynamique, une prédiction statique est alors utilisée pour maintenir le cache d'adresse cible selon la direction du branchement.

Par ailleurs, le PowerPC 620 permet l'exécution spéculative des instructions en supportant jusqu'à quatre niveaux de branchements non résolus. Les résultats temporaires issus de ces exécutions sont conservés dans les registres de renommage ou dans les tampons de réordonnancement en attendant la confirmation de la prédiction.

Au niveau de l'exécution des instructions, le PowerPC 620 dispose de deux unités entières arithmétiques responsables des instructions courantes (exécution en un cycle). Pour les opérations plus longues (multiplication et division), une unité arithmétique complexe à trois étages est implémentée.

L'unité flottante a également un pipeline d'exécution de trois étages et est conforme à la norme *IEEE 754*. Il est à noter qu'un mode non IEEE a également été implémenté dans le cadre d'un support aux opérations graphiques.

Au niveau de l'étage de mise à jour des registres, un tampon de réordonnancement de 16 entrées permet de conserver l'ordre et l'état des instructions. Par ailleurs, comme on peut s'y attendre sur ce type de processeur, une structure de files permet de découpler les accès vers le cache de données à partir de l'unité de lecture/écriture.

Le PowerPC 620 met en œuvre deux caches d'instructions et de données séparés, de 32 Ko, associatifs par ensemble 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).

La sélection d'un ensemble parmi huit pose un problème de vitesse et de consommation. Aussi, une partie de l'adresse virtuelle est utilisée pour pré-sélectionner une ligne de l'ensemble.

Les caches sont protégés par parité au niveau de l'octet pour les données et les instructions. Pour le cache d'instructions, sept bits de prédécodage sont associés à chaque instruction.

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 Mo. Le bus de données vers ce cache a une largeur de 128 bits. Un mécanisme de correction d'erreur simple et de détection d'erreur double est utilisé. Comme le premier niveau de cache, il implémente un protocole de cohérence de type MESI en vue de son implémentation dans un système multiprocesseur.

L'une des innovations de ce processeur est son architecture 64 bits. Le 620 implémente une unité de gestion de la mémoire à deux niveaux. Le premier niveau s'appuie sur une table de traduction d'adresse virtuelle vers des adresses physique de 64 entrées, totalement associative pour les instructions et les données (*ERAT* pour *Effective to Real Address Translation*). Le second niveau utilise une table de traduction de 128 entrées associative à deux voies (*TLB*). Une table de 20 entrées, totalement associative, est également implémentée et est dédiée à la traduction des segments (*SLB*). Les seize premières entrées servent également à implémenter les registres de segments pour l'exécution de codes compilés pour des microprocesseurs PowerPC d'architecture 32 bits.

La gestion de l'ensemble des tables est entièrement faite par matériel et respecte le système de mémoire virtuelle paginée segmentée définie par l'architecture PowerPC. L'unité de gestion de la mémoire du PowerPC 620 supporte jusqu'à 2^{80} octets de mémoire virtuelle et 2^{40} octets de mémoire physique.

Par ailleurs, le PowerPC 620 implémente les registres de traduction d'adresse de blocs pour les instructions et les données qui facilitent la traduction des adresses pour de vastes espaces de mémoire contigus (de 128 Ko à 256 Mo).

L'unité d'interface système implémente une nouvelle architecture de bus spécifiquement adaptée aux besoins de systèmes multiprocesseurs (jusqu'à huit PowerPC 620). Ce bus est constitué d'un

bus d'adresse physique de 40 bits (au format *big-endian*) et d'un bus de données configurable lors de l'initialisation en mot double ou quadruple. Ses performances devraient permettre d'atteindre des transactions de 33 millions d'adresses par seconde pour un débit en données de 1 Go/sec.

Ce processeur dispose par ailleurs de fonctionnalités de test des éléments mémoire. Un contrôleur de performances est également implémenté par l'intermédiaire de deux compteurs de 32 bits. Ces compteurs permettent de suivre deux évènements sélectionnables sur un ensemble de 68.

Séquencé à la fréquence de 133 MHz, le PowerPC 620 implémente un contrôle dynamique de sa consommation (idem que le PowerPC 604 qui dispose d'un mode de veille). Il inclut par ailleurs un *Phase Looked Loop* qui facilite son intégration dans les systèmes. Réalisé en technologie CMOS à 0.5 μm et quatre couches de métal, il comporte environ 7 millions de transistors. Ses performances attendues sont respectivement de 222 SPECint92 et 300 SPECfp92.

Chapitre 4

Les microprocesseurs SPARC

SPARC Technology Business est une division de *Sun Microsystems, Inc.*. Cette architecture issue des travaux menés à l'Université de Berkeley entre 1984 et 1987 est basée sur les concepts RISC. Elle privilégie une architecture pipelinée, un jeu d'instructions simple accompagné d'un compilateur efficace. SPARC est un acronyme pour *Scalable Processor ARChitecture*.

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 l'ensemble des développeurs ou vendeurs souhaitant influer sur l'évolution du standard de cette architecture. On y retrouve en dehors de *Sun*, *Fujitsu Ltd.* et *Texas Instruments*, tous deux acteurs des développements de l'architecture SPARC. *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 processeurs d'architecture SPARC offrent une grande diversité. Nous essayerons à travers l'étude de quelques uns de ces processeurs de dresser un panel représentatif des diverses implémentations de cette architecture.

4.1 La norme SPARC-V9

La norme SPARC-V9, comme sa prédécesseur la version 8, a été définie par *SPARC International*. SPARC-V8 définit une architecture de type 32 bits. Cette nouvelle architecture, tout en maintenant une compatibilité binaire introduit les caractéristiques suivantes :

- calcul entier des adresses sur 64 bits ;
- 16 registres flottants double-précision supplémentaires ;
- codes conditions pour des données 32 et 64 bits ;
- quatre ensembles de codes conditions flottants au lieu de un ;
- exécution spéculative d'instructions de lecture ;

- ordre des octets en mémoire configurable en *little* ou *big-endian* ;
- validation logicielle de la prédiction de branchement ;
- nouvelles instructions de déplacement conditionnel, de branchement, de synchronisation, de multiplication/division, de préchargement, de calcul sur des données de 64 bits, etc... ;
- gestion plus aisée des fenêtres de registres ;
- gestion plus rapide des exceptions grâce à un nouvel ensemble de registres ;
- nouvelles instructions de synchronisation et d'accès à la mémoire.

Cette norme offre aux utilisateurs un espace d'adressage et de calcul de type 64 bits et des instructions spécifiques à un environnement multiprocesseur.
Nous présentons dans ce chapitre cette nouvelle norme à travers l'étude du jeu d'instructions et quelques caractéristiques de son implémentation.

4.1.1 Le jeu d'instructions

Types de données

L'architecture SPARC-V9 reconnaît les types fondamentaux suivants (tableau 4.1) :

- les entiers signés (8, 16, 32 et 64 bits) ;
- les entiers non signés (8, 16, 32 et 64 bits) ;
- les données flottantes (32, 64 et 128 bits).

octet	demi-mot	mot	mot étendu	mot étiqueté	double-mot	quadruple-mot
8	16	32	64	32 [a]	64	128

TAB. 4.1 - *Définition du format des données selon la norme SPARC-V9 (en bits)*

[a] 30 bits de valeur plus 2 bits d'étiquette.

Modes d'adressage

L'architecture SPARC-V9 met en œuvre les quatre modes d'adressage classiques déjà mis en œuvre dans la norme SPARC-V8, à savoir les modes d'adressage absolu, indirect, basé et indexé. La valeur des immédiats est signée et codée sur 13 bits.

Le jeu d'instructions

Les instructions définies par cette architecture se répartissent dans les types de base suivant :

- les instructions d'accès à la mémoire ;
- les instructions arithmétiques entières ;
- les instructions arithmétiques flottantes ;
- les instructions de transferts de contrôle ;
- les instructions de déplacement conditionnel ;
- et enfin des instructions de contrôle du système et de gestion de registres.

Les instructions d'accès à la mémoire. L'architecture SPARC étant une architecture *load/store*, les seules instructions d'accès à la mémoire sont des instructions de lecture ou d'écriture.

Le jeu d'instructions SPARC inclut des instructions atomiques d'accès à la mémoire. Il est à noter que les données en mémoire doivent être alignées sur des frontières respectant le format de la donnée accédée. Par défaut, l'ordre des données en mémoire est *big-endian*. Cependant, la norme SPARC-V9 introduit la possibilité de passer à un format *little-endian* grâce à un bit de configuration.

Le jeu d'instructions défini par l'architecture SPARC-V9 introduit un nouveau type d'instructions de lecture qui permet au compilateur de déplacer une instruction de lecture classique en amont de la structure de contrôle qui garantit son usage. La sémantique de cette instruction est la même que les autres si ce n'est qu'elle ne provoque par d'exception lorsqu'une erreur est détectée. Une valeur nulle est alors renvoyée (d'où le nom de cette instruction appelée *non-faulting load*). L'exemple 1 illustre son usage. Cette optimisation est particulièrement utile dans un contexte superscalaire.

Exemple 1 : Rôle des instructions de lecture non génératrice d'exception

Fragment de programme C :

```
if (p != NULL) x = *p + y;
```

*Avec cette nouvelle instruction de lecture, la lecture de (*p) peut être placée par le compilateur avant le test de (p != NULL). L'effet est alors équivalent à cette transformation :*

```
temp_register = *p;
if (p != NULL) x = temp_register + y;
```

Une autre particularité du jeu d'instructions définie par la norme SPARC-V9 concerne l'introduction d'instructions de préchargement qui se déclinent selon plusieurs versions :

- préchargement pour une lecture ;
- préchargement pour plusieurs lectures ;

- préchargement pour une écriture ;
- préchargement pour plusieurs écritures ;
- préchargement d'une page ;
- préchargement dépendant de l'implémentation.

Ces instructions permettent d'anticiper le chargement des données et de procéder à la résolution des défauts de cache ou de TLB si il y a lieu. Le fait de disposer de plusieurs variantes permet de mêler logiciel et matériel afin d'améliorer les performances. Effectivement, ces instructions fournissent au matériel une indication sur le mode d'utilisation de la donnée préchargée. Ainsi, un préchargement pour plusieurs lectures (cas le plus important) sera initié de manière à ce que la donnée soit fournie avec des efforts « raisonnables » [1]. Un préchargement pour une lecture (la donnée ne sera lue qu'une fois puis non réutilisée en lecture ou écriture), ne sera exécuté que si il perturbe le moins possible le cache (pas de réécriture de données dans le cache).

Il est à noter que ces diverses variantes ne sont pas utilisées pour le préchargement des instructions. Ce cas est traité à travers l'instruction de branchement *BPN* (*Branch Never with Prediction*) qui est interprété par le processeur comme le préchargement de la cible jamais prise !

La norme SPARC-V9 reprend et améliore le modèle de mémoire précédemment introduit dans la norme SPARC-V8. Ce modèle spécifie le comportement de la mémoire dans des systèmes monoprocesseurs ou multiprocesseurs à mémoire partagée. La norme SPARC-V8 introduisait trois modes de fonctionnement :

Strong Consistency - Ce mode à forte rigidité assure que l'ordre de tous les accès mémoire en provenance d'un même processeur vu par n'importe lequel des processeurs est rigoureusement l'ordre spécifié par les instructions.

Total Store Ordering (TSO) - Ordre total des écritures en mémoire. Même chose que précédemment mais seulement sur les écritures.

Partial Store Order (PSO) - Ordre partiel des écritures. Les écritures sur une même case mémoire sont ordonnées.

La norme SPARC-V9 introduit un nouveau modèle de mémoire appelé *Relaxed Memory Order (RMO)*. Ce modèle de mémoire impose peu de contraintes et permet de séquencer les accès à la mémoire (lecture ou écriture) dans n'importe quel ordre tant que le résultat reste juste. Il constitue donc le mode de fonctionnement le plus performant. Lorsqu'une nécessité d'ordonnancement est requise, elle doit être explicitement formulée dans le programme par l'instruction *MEMBAR*.

L'instruction *MEMBAR* pour *memory Barrier* sert deux fonctions distinctes :

- une fonction explicite de synchronisation lors des accès à la mémoire (instruction similaire à celle du DEC Alpha) ;
- une fonction de contrôle sur le séquencement des accès à la mémoire.

1. Par exemple, un défaut de TLB sera résolu, une erreur multiple ECC provoquera l'abandon de cette instruction.

Cette deuxième fonctionnalité est nécessaire lorsque l'on souhaite que l'effet de cette instruction soit globalement visible au lieu de s'appliquer localement. Elle introduit alors un ordre dans le flot d'exécution des instructions en fonction de la commande passée en paramètre. Ce paramètre est codé dans le champ de l'instruction. Ainsi :

- un champ 01_{16} (*#LoadLoad*) impose que toutes les instructions de lecture précédant l'instruction MEMBAR soit effectuées avant toutes celles qui la suivront. Les opérations d'écriture ne subissent quant à elles aucune contrainte.
- un champ 08_{16} (*#StoreStore*) est l'équivalent pour les instructions d'écriture (précédemment instruction STBAR de la norme SPARC-V8) ;

Un panachage des masques est également possible, un paramètre 09_{16} par exemple (*#LoadLoad — #StoreStore*) ordonne les lectures, les écritures mais n'introduit aucune contrainte entre les lectures et les écritures et vice-versa.

Le jeu d'instructions défini par la norme SPARC-V9 inclut également des instructions d'accès à la mémoire spécifiques à un système multiprocesseur. Ces instructions permettent de mettre en œuvre des primitives de synchronisation utilisées par les mécanismes d'exclusion mutuelle. On peut citer :

- l'instruction *Compare and Swap*, qui compare la valeur d'un registre avec une valeur contenue en mémoire, et, si elles sont égales effectue l'échange entre la valeur contenue en mémoire et celle d'un second registre.
- l'instruction *swap* qui n'effectue que la dernière partie de l'instruction précédente.
- l'instruction classique de *test-and-set*.
- et enfin l'instruction *Flush* qui assure que le double-mot spécifié par l'adresse est cohérent à travers toute la hiérarchie mémoire et éventuellement dans un système multiprocesseur, dans tous les caches locaux.

Les instructions arithmétiques entières. Ces instructions correspondent aux instructions classiques de calcul entier, ainsi qu'aux opérations de test et de décalage.

L'instruction de multiplication entière prend deux opérandes 64 bits pour fournir un résultat sur 64 bits, de même que la division. Par ailleurs, pour des raisons de compatibilité avec la norme SPARC-V8, des instructions de type (32 x 32 $\rightarrow$ 64 bits) et (64 $\div$ 32 $\rightarrow$ 32 bits) sont incluses.

La norme SPARC-V9 introduit en outre l'ensemble des instructions relatives au calcul sur 64 bits.

L'architecture SPARC fournit des instructions d'addition et de soustraction « étiquetées ». Ces instructions sont utiles pour les langages *Lisp* et *Smalltalk*. L'étiquette concerne les deux bits de poids faible des opérandes. Si l'une ou l'autre de ces étiquettes est non nulle ou si un débordement de capacité survient alors un *overflow* est signalé.

Les instructions arithmétiques flottantes. Ces instructions effectuent l'ensemble des calculs arithmétiques flottants conformes à la norme *IEEE 754-1985* (en ce qui concerne les données 32 et 64 bits. Pour les données 128 bits la norme en vigueur est alors la *IEEE Standard for Shared Data Formats 1596.5-1993*).

La norme SPARC-V9 définit 32 registres flottants simple-précision (32 bits), 32 registres double-précision (64 bits) et 16 registres flottants de 128 bits. Nous reviendrons au chapitre 4.1.2 sur le détail de leur implémentation.

Les instructions de transfert de contrôle. De même que MIPS, SPARC a fait le choix de mettre en œuvre des branchements retardés. Un bit de l'instruction de transfert permet d'annuler cette instruction. Il est à noter que dans le cas de la norme SPARC-V8, cette instruction était chargée à partir du cache même en cas d'annulation. La norme SPARC-V9 améliore cet aspect en ne chargeant l'instruction que si elle est exécutée. Ceci, permet d'éviter le traitement des défauts de caches sur instruction annulée.

Les instructions de transfert de contrôle sont relatives au PC. Ces déplacements sont variables et sont explicités au tableau 4.2. Il est à noter que dans le cas des instructions de *jump and link*, l'adresse de retour est conservée dans un registre spécifié par l'instruction.

branchement sur codes conditions sans prédiction	$\pm 8Mo$
branchement sur codes conditions avec prédiction	$\pm 1Mo$
branchement sur comparaison du contenu d'un registre	$\pm 128Ko$
déplacement selon l'instruction CALL (mot de 30 bits)	$\pm 2Go$

TAB. 4.2 - *Valeur des déplacements relatifs au PC selon la norme SPARC-V9*

Les instructions de déplacement conditionnel. Ces instructions sont similaires aux instructions mises en œuvre sur l'architecture DEC. Elles effectuent la copie d'une registre source vers un registre destination selon la valeur d'un code condition entier ou flottant ou selon le contenu d'un registre entier. Ces instructions augmentent les performances en réduisant le nombre de branchement.

Pour plus de détails sur l'évolution de l'architecture SPARC, nous renvoyons le lecteur à [10].

4.1.2 Caractéristiques de mises en œuvre de l'architecture SPARC-V9

La configuration des registres entiers

L'architecture SPARC met en œuvre des fenêtres de registres. Ce choix a été motivé par des études qui ont montré (selon [11]) que la sauvegarde des registres lors des appels de procédures et leur restitutions à leur retour peut coûter entre 5% à 40% des références mémoires. Une alternative est d'utiliser plusieurs bancs de registres et d'en allouer un nouveau à chaque nouvel appel de procédure. Pour éviter d'être limité dans ces appels, les bancs de registres sont organisés sous la forme d'une file circulaire. Cette technique est appelée *fenêtre de registres*.

L'une des difficultés est de passer les paramètres. Si chaque procédure à son propre ensemble de registres, aucune donnée ne peut être partagée. L'un des moyens de résoudre cet inconvénient est de permettre un recouvrement des fenêtres de registres de manière à disposer d'une zone commune pour passer les paramètres. C'est cette technique qui a été mise en œuvre sur l'architecture SPARC.

L'unité entière peut contenir, selon la norme SPARC-V9, de 64 à 528 registres généraux de 64 bits. Ils sont partitionnés en huit registres globaux, huit registres « d'alternance » globaux destinés

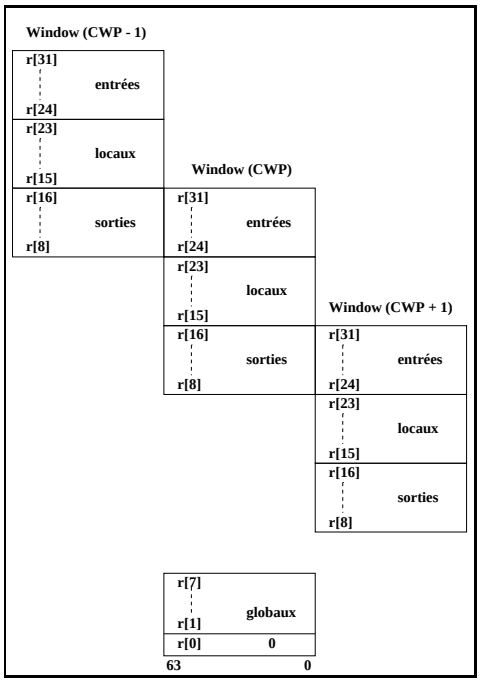

FIG. 4.1 - *Exemple de fenêtres de registres sur l'architecture SPARC-V9*

au *mode privilégié*[2], plus un nombre d'ensemble de 16 registres dépendant de l'implémentation matérielle (ce nombre étant compris entre 3 et 32).

Une fenêtre de registres consiste en huit registres d'entrée, huit registres locaux et huit registres de sortie.

À n'importe quel moment, une instruction peut accéder aux huit registres globaux et aux 24 registres de la fenêtre courante. La fenêtre de registres comprend les huit registres d'entrées et locaux d'un ensemble particulier, plus les huit registres d'entrées d'un ensemble adjacent, adressés à partir de la fenêtre courante comme des registres de sortie. La figure 4.1 illustre cette mise en œuvre. La fenêtre courante est désignée par un pointeur (nommé *CWP* sur la figure pour *Current Window Pointer*).

L'adressage des registres flottants

Comme on l'a dit précédemment, la norme SPARC-V9 impose :

- 32 registres flottants simple précision (nommés *f[0], f[1], ..., f[31]*) ;
- 32 registres flottants double-précision (nommés *f[0], f[2], ..., f[62]*) ;
- 16 registres flottants quadruple-précision (nommés *f[0], f[4], ..., f[60]*).

À la différence des fenêtres de registres entiers, l'ensemble des registres flottants est accessible à tout moment. Ces registres sont lus et écrits par l'ensemble des instructions de calcul arithmétiques

2. Seul l'un des ensembles de registres globaux est accessible à un instant donné. Par ailleurs, le registre global *R0* est toujours lu à zéro.

flottantes ainsi que par les instructions de lecture/écriture. Il existe cependant une astuce au niveau du numéro de codage de ces registres en fonction du type de données accédées. Le tableau 4.3 explicite cette mise en correspondance.

type de donnée	Codage du numéro de registre dans l'instruction					Adressage effectif du registre					
simple	b[4]	b[3]	b[2]	b[1]	b[0]	0	b[4]	b[3]	b[2]	b[1]	b[0]
double	b[4]	b[3]	b[2]	b[1]	b[5]	b[5]	b[4]	b[3]	b[2]	b[1]	0
quadruple	b[4]	b[3]	b[2]	0	b[5]	b[5]	b[4]	b[3]	b[2]	0	0

TAB. 4.3 - *Codage du numéro de registre flottant selon la norme SPARC-V9*

Former un mot quadruple par concaténation de registres simple précision n'est pas une manipulation qui est considérée comme importante.

La gestion des exceptions

Sur les processeurs d'architecture antérieure à la norme SPARC-V9, les exceptions sont gérées en allouant une nouvelle fenêtre de registres et en sauvant le PC ainsi que le futur PC dans deux registres de cette fenêtre, les bits d'états critiques étant sauvegardés dans le registre PSR (*Processor State Register*).

Pour la gestion des exceptions, la norme SPARC-V9 implémente une pile de registres pour sauvegarder l'état courant du processeur, rendant inutile le changement de fenêtres. Cette pile de registres permet de sauvegarder les deux PC, le type de l'exception et l'état courant du processeur (codes conditions, registre d'identification de l'espace d'adressage, registre d'état du processeur, pointeur sur la fenêtre courante de registres). Quatre niveaux de pile constituent le minimum spécifié par la norme SPARC-V9 qui peut aller jusqu'à sept niveaux d'interruption.

On peut également remarquer que sur la norme SPARC-V8, les routines d'exceptions étaient quittées par l'instruction *RETT* exécutée dans le *delay slot* de l'instructions de branchement qui la précédait. Cette instruction est désormais remplacée par les instructions *DONE* et *RETRY* qui restitue l'état du processeur de façon atomique.

Mécanismes de traduction d'adresses

Dans la norme SPARC-V8, l'unité de gestion de la mémoire traduit les 32 bits d'adresses virtuelles et les 16 bits de numéro de contexte en une adresse physique de 36 bits, en accédant à quatre niveaux de tables de page. De manière générale, la traduction de ces adresses est conservée dans la TLB, inhibant ainsi le besoin de passer par ces tables. Mais quand un défaut de TLB est détecté, ces quatre niveaux de tables permettent de retrouver l'adresse de la page physique. La figure 4.2 récapitule ce schéma de traduction.

La taille des pages gérée par la MMU à travers son mécanisme de traduction d'adresse varie de 4 Ko à un contexte de 4 Go, en passant par les tailles intermédiaires de 256 Ko et 16 Mo. Dans ce

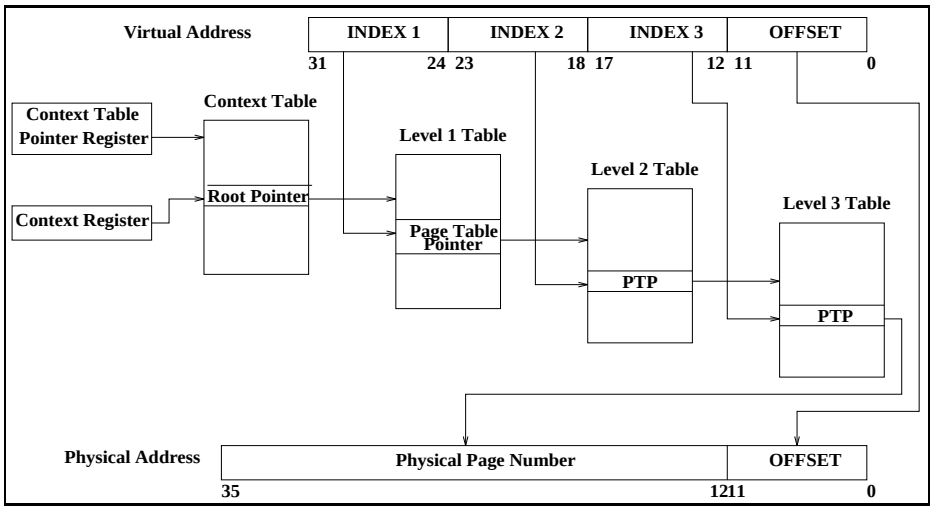

FIG. 4.2 - *Mécanisme de traduction des pages selon la norme SPARC-V8*

cas, le nombre de tables de pages adressées pour obtenir la traduction de l'adresse diminue. Ainsi, dans le cas d'une page de 256 Ko, le champ *INDEX 3* de la figure 4.2 est joint au champ *OFFSET* et la table de niveau 3 n'est pas adressée.

La norme SPARC-V9 reprend le même mécanisme de traduction d'adresses. Elle définit des adresses virtuelles sur 44 bits et un espace d'adressage physique sur 41 bits. Par ailleurs, quatre tailles de pages sont supportées par l'unité de gestion de la mémoire : 8 Ko, 64 Ko, 512 Ko et 4 Mo.

Après avoir expliqué quelques caractéristiques de cette architecture, nous allons maintenant pouvoir aborder son implémentation à travers l'étude des microprocesseurs.

4.2 Le Sun SPARC

Ce processeur est maintenant obsolète du fait des nouvelles versions mais présente un intérêt historique.

Le Sun SPARC est le premier microprocesseur à caractère industriel issu des travaux de l'Université de Berkeley sur les architectures RISC et commercialisé en 1987. Ce processeur, scalaire, implémente la norme SPARC-V7 et est organisé autour d'une unité entière, d'une unité flottante externe, d'un cache unifié d'instructions et de données ainsi que d'une unité de gestion de la mémoire (voir [1]).

Pour les instructions entières, un pipeline simple de quatre étages est utilisé (chargement, décodage, exécution, mise à jour). Les instructions de lecture/écriture nécessitent un cycle supplémentaire (étage d'accès au cache). Les opérations flottantes nécessitent au minimum 5 cycles.

Les unités flottantes et entière sont totalement synchronisées. L'unité entière lit une instruction dans le cache à chaque cycle. Celle-ci est décodée en parallèle dans les deux unités. Lorsque l'unité entière reconnaît qu'il s'agit d'une instruction flottante, elle l'abandonne. Le temps d'exécution d'une instruction flottante varie de quatre cycles à 63 cycles (cas de la racine carrée double préci-

sion). Aussi, l'unité flottante comprend une pile qui permet à l'instruction d'attendre la fin de son exécution. Lorsque cette pile est pleine, il se produit un gel des pipelines entier et flottant.

L'unité entière implémente 136 registres de 32 bits utilisés à travers le mécanisme de fenêtrage (soit huit fenêtres de registres). Les opérations de multiplication/division sont émulées par logiciel.

L'unité flottante quant à elle comprend un additionneur, un multiplieur et 32 registres de 32 bits ou 16 registres de 64 bits.

Comme on l'a dit précédemment, le Sun SPARC regroupe sur un même cache externe les instructions et les données. Ce cache a une capacité maximale de 64 Ko et est à correspondance directe. Il est virtuellement adressé. La politique d'écriture est configurable en simultanée ou différée. Le temps d'accès au cache est de un cycle en cas de succès, 18 autrement. Une protection par parité est implémentée. Un deuxième niveau de cache peut également être ajouté.

L'unité de gestion mémoire est également externe. L'architecture SPARC définit un espace d'adressage linéaire paginé. La taille des pages est constante (4 Ko). L'espace virtuel alloué à un processus est un contexte. L'architecture supporte jusqu'à 256 contextes simultanément (un champ d'identification d'espace d'adressage de huit bits est émis avec les 32 bits d'adresse virtuelle). Le mécanisme de traduction d'adresse utilise plusieurs niveaux d'indirection pour convertir l'adresse virtuelle en adresse physique (à partir du contexte, adressage de la table des contextes, puis table de pointeurs de niveau 1, table de pointeurs de niveau 2, puis table de pages de niveau 3. Voir [1] pour plus de détails). L'adresse virtuelle est codée sur 32 bits, l'adresse physique compte quant à elle 36 bits[3].

Par ailleurs, le Sun SPARC inclut une table de traduction d'adresses, totalement associative de 64 entrées. Un algorithme aléatoire est utilisé pour le renouvellement des lignes de cette table.

Ce processeur comprend également des fonctionnalités permettant sont intégration dans des systèmes multiprocesseurs : jeu d'instructions, protocole de cohérence de type MOESI. Ce processeur traduit une différence d'approche avec le MIPS R3000 qui nécessite d'être initialisé dans un mode spécial à un fonctionnement multiprocesseur. Dans le cas du Sun SPARC, tout est réalisé par matériel avec des composants spécifiques. Le microprocesseur ignore alors qu'il fonctionne dans un environnement multiprocesseur.

Comme le MIPS R3000, le Sun SPARC a donné lieu à des déclinaisons de son architecture dans le milieu des microcontrôleurs embarqués. Le concept de fenêtres de registres favorise cette architecture en offrant des changements de contexte extrêmement rapide (pas de sauvegarde de registres). On citera à titre d'exemple le *CY7C611* de *Cypress Technology* et le *MB86930* de *Fujitsu*.

4.3 Le SuperSPARC

Annoncé par *Texas Instruments* et *Sun* en août 1991, le SuperSPARC est le premier processeur superscalaire d'architecture SPARC. Destiné au marché des stations de travail, il comprend une unité entière intégrant deux ALUs, une unité flottante, une unité de gestion de la mémoire ainsi qu'un cache d'instructions et de données séparés. Par ailleurs, pour compléter l'unité de gestion de la mémoire, un TLB de 64 entrées est mis en œuvre pour les instructions et les données. Ce microprocesseur est conforme à l'architecture SPARC-V8.

3. Il est intéressant de remarquer que ce processeur était le seul à mettre en œuvre une adresse virtuelle inférieure à l'adresse physique.

Commercialement, ce processeur n'a pas remporté un franc succès du fait des nombreux déboires survenus. Annoncé dans un premier temps à 50 MHz en 1991, les premiers processeurs seront livrés à 33 MHz du fait de problèmes de *timing* internes. Il faudra attendre début 1993 avant d'atteindre les 40 MHz, puis le SuperSPARC+ commercialisé au deuxième semestre 1993 pour atteindre les 50 MHz. Les versions actuelles de SuperSPARC sont séquencées à 60 MHz, mais restent encore bien en dessous du niveau de performances atteint par d'autres architecture RISC. L'origine de ces difficultés peut être attribuée à la technique employée sur le banc de registres entiers (ainsi que sur la table de traduction d'adresses), véritable goulot d'étranglement du processeur au moment de l'augmentation de la fréquence d'horloge (technologie *double-pump*). Le concept de fenêtres de registres nécessite d'implémenter un grand nombre de registres. La mise en œuvre d'un fichier de 136 registres avec huit vrais ports d'accès occupe une place énorme. Aussi, le banc de registres est en fait un faux double-port, le même port étant sollicité deux fois dans le même cycle afin de réduire la surface nécessaire.

Quatre instructions sont chargées directement à partir du cache vers deux files d'instructions. 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). La profondeur de chacune de ces files est de six entrées. Le SuperSPARC précharge les deux chemins d'exécution. Quand le branchement est réellement 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, cependant, un effort important a été fourni pour minimiser les gels du pipeline :

- Du fait de la présence des deux ALUs, deux instructions entières peuvent être émises simultanément, ou une instruction entière et une instruction de lecture. Le banc de registres entiers comprend donc deux ports d'écriture destinés à recevoir les résultats de ces opérations. Il est à noter cependant que dans le cas d'une lecture de donnée de 64 bits, les deux ports sont simultanément utilisés, empêchant l'exécution d'une instruction entière au même cycle. Deux instructions entières dépendantes peuvent également être séquencées au même cycle. Ceci est possible dans la mesure où chacune des ALU effectue un calcul en un demi-cycle, et que les deux ALUs peuvent être utilisées en cascade dans le même étage de pipeline.

- Une instruction de branchement peut être simultanément exécutée avec les deux instructions entières.

- Le banc de registres flottants comprend également deux ports d'écriture 64 bits, permettant ainsi l'exécution simultanée d'un *load* et d'une opération.

Le SuperSPARC utilise un pipeline entier de quatre cycles. Chaque cycle a deux phases. Le premier cycle (*F0, F1*) charge quatre instructions à partir du cache dans l'une des files. Puis, trois phases de décodage sont utilisées (*D0, D1, D2*) et enfin deux phases servent à l'exécution (*E0, E1*) avant la mise à jour du banc de registres (*WB*).

En accord avec la norme SPARC version 8, ce processeur inclut des instructions de multiplication et de division entières. Ces calculs sont exécutés par l'unité flottante (quatre cycle de latence pour la multiplication et 15 cycles pour la division).

Le SuperSPARC comprend deux caches d'instructions et de données séparés, tous deux physiquement adressés et testés. Le cache d'instructions a une taille de 20 Ko, il est associatif par

ensemble à cinq voies. Un degré d'associativité minimal de cinq permet de bénéficier d'ensembles de taille inférieure ou égale à la taille des pages, facilitant ainsi l'adressage des éléments en utilisant les bits d'adresse de poids faible.

Le cache de données a quant à lui une taille de 16 Ko et est associatif par ensemble à quatre voies (ce degré d'associativité s'explique pour la même raison que le cache d'instructions). La politique de mise à jour de la mémoire est configurable en simultanée ou en différée.

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

Ce processeur est à rapprocher du R4000 dont l'architecture traduit une approche différente (superpipeline versus superscalaire, voir [2]). Le SuperSPARC comprend 3.1 millions de transistors et est réalisé en technologie BiCMOS à 0.6 μm avec trois couches de métal. Ses performances sont de l'ordre de 89 SPECint92 et 103 SPECfp92 pour une fréquence de 60 MHz.

4.3.1 Le SuperSPARC+

Ce processeur a été annoncé au premier semestre 1993 par *Texas Instrument*. Séquencé à la fréquence de 50 MHz, il est fonctionnellement équivalent au SuperSPARC si ce n'est qu'il permet l'exécution d'opérations dans un système multiprocesseur à travers l'interface du MBus (le SuperSPARC nécessite un contrôleur de cache externe pour de telles opérations).

Le SuperSPARC+ ne supprime cependant pas la totalité des composants nécessaire à la réalisation d'un système multiprocesseur. Effectivement, l'unité centrale doit fonctionner à la fréquence du bus externe, les opérations au-dessus de 40 MHz, limite du MBus, nécessitent un contrôleur de cache externe qui fournit une interface asynchrone entre le processeur et le MBus. Ce composant est également sollicité par le deuxième niveau de cache.

Réalisé en technologie BiCMOS à 0.7 μm, les performances annoncées de ce processeur sont de l'ordre de 65.2 SPECint92 et 83 SPECfp92

4.3.2 Le SuperSPARC 2

Annoncé fin 1994 par *Sun's SPARC Technology Business*, ce processeur vient combler le vide laissé entre l'arrivée de l'UltraSPARC et le SuperSPARC. Le design de ce dernier a subi de profondes modifications qui lui permettent d'atteindre une fréquence de 90 MHz et des performances de l'ordre de 135 SPECint92 et 145 SPECfp92.

Il est intéressant de remarquer les modifications survenues entre ces deux générations qui soulignent les points faibles du SuperSPARC.

Registres et pipeline

La technique du *double-pump* sur le banc de registres entier a été abandonnée pour être remplacée par un ensemble de ports de lecture. Le fait d'éliminer ce goulot d'étranglement permet d'anticiper la lecture des registres d'où l'élimination d'une phase de décodage dans le pipeline entier. Le pipeline du SuperSPARC 2 reste cependant sur quatre cycles.

La capacité des files destinées à la réception des instructions issues des flots d'instructions cibles et séquentielles (douze entrées) a été doublée.

L'une des améliorations majeures concerne le cache de traduction d'adresses unifié, de 64 entrées, également accédé deux fois par cycle. Le SuperSPARC 2 inclut deux TLB d'instructions et de données distincts. Pour maintenir un taux de succès au moins équivalent au design précédent, les concepteurs ont conservé la même capacité pour les données (64 entrées) et ont introduit un cache de traduction d'adresses pour les instructions de 16 entrées.

Un troisième problème détecté sur le SuperSPARC concerne le multiplieur flottant qui sert également aux divisions et aux racines carrées (algorithme itératif). Pour permettre une fréquence d'horloge plus élevée, le SuperSPARC 2 simplifie cette unité en ajoutant une unité distincte pour les divisions et les racines carrées. Cette nouvelle organisation permet d'avoir une latence de trois cycles pour les multiplications mais allonge la latence des opérations de division et de racine carrée de 50%. L'augmentation de la fréquence d'horloge compense cependant cet allongement (gain de 40% entre les deux versions).

Les autres caractéristiques du SuperSPARC ont été peu modifiées. Le même procédé de fabrication a été utilisé que pour le SuperSPARC à 60 MHz (technologie BiCMOS à 0.6 μm et à trois couches de métal).

4.4 Le MicroSPARC

Annoncé en octobre 1992 par *Texas Instrument* et *Sun Microsystems*, ce processeur (connu également sous le nom de *TMS 390S10*) marque une évolution dans le design des processeurs SPARC en incluant sur le même composant l'essentiel des fonctionnalités lui permettant d'être intégré dans un système.

Destiné aux stations de travail d'entrée de gamme et au marché des machines portables, ce composant comprend le cœur du processeur plus toute la logique d'interface vers la DRAM et le SBus. Il est séquencé à 50 MHz.

Ce processeur est conforme à l'architecture SPARC-V8. Il est constitué d'une unité entière, d'une unité flottante, d'un cache d'instructions et de données séparés, d'une unité de gestion de la mémoire ainsi que de toute la logique d'interface vers l'extérieur.

L'unité entière utilise les cinq étages classiques d'un processeur RISC. Conformément à la norme SPARC version 8, les multiplications et divisions entières sont effectuées au sein de cette unité. L'écriture et la lecture de mots de 32 bits en mémoire prennent deux cycles, de même que la lecture de mot de 64 bits, alors que dans le cas des écritures ces derniers nécessitent un cycle supplémentaire.

Le MicroSPARC implémente sept fenêtres de registres pour un total de 120 registres.

L'unité flottante utilise un cœur développé par la *Meiko Ltd.*. Cette unité est non pipelinée. Les latences des opérations classiques sont de :

- 5 cycles pour les additions (simple et double précision) ;
- respectivement 5 et 9 cycles pour les multiplications simple et double précision ;
- de même 20 et 35 cycles pour les divisions ;
- et 37-65 cycles pour la racine carrée.

Un premier niveau de cache est implémenté sous la forme de deux caches à correspondance directe. Leur capacité est de 4 Ko pour le cache d'instructions et de 2 Ko pour le cache de données (à écriture simultanée). Ils sont tous deux physiquement adressés. La faible capacité de ces caches est compensée par une interface directe avec la DRAM. Cependant, du fait de cet accès, il est impossible de rajouter un second niveau de cache. Les défauts de cache sont satisfaits en quatre cycles si la page est présente dans la ligne de « cache » de la DRAM ou neuf cycles dans le cas contraire.

L'unité de gestion de la mémoire s'appuie sur un TLB complètement associatif de 32 entrées. Le MicroSPARC utilise un *micro-TLB Instruction* de une entrée accédé uniquement lorsque le TLB principal est inaccessible. Ce *micro-TLB* est mis à jour lors de chaque accès au TLB principal de manière à augmenter la probabilité d'y trouver la bonne traduction. Lors d'un défaut, la pénalité est alors de trois cycles.

La taille des pages gérée par l'unité de gestion de la mémoire est paramétrable et est de 4 Ko, 256 Ko ou 16 Mo.

L'une des caractéristiques innovantes de ce processeur est la réduction du nombre de composants nécessaires à la conception d'une station de travail SPARC. Comparé aux systèmes SPARC existants (autour du SPARC-V7), les caches internes ainsi que la MMU réduisent de manière non négligeable le nombre de composants, sans compter l'interface de contrôle de la DRAM ainsi que celle du Sbus.

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.

Pour compléter ce processeur, un ensemble de deux composants est fourni. Ils viennent directement s'interfacer sur le Sbus dont ils occupent une adresse logique. L'un de ces composants (le 89C100) fonctionne en maître/esclave alors que le deuxième est exclusivement un esclave (89C105). Le premier implémente un contrôleur DMA et SCSI ainsi qu'une interface Ethernet et un port parallèle. Le deuxième quant à lui fournit des organes auxiliaires de communication (interface bus huit bits, ports séries, contrôleur de disque, etc...).

Ce processeur se substitue au *Fujitsu 86903* séquencé à 40 MHZ sur lequel est basé la station de travail *SPARC IPX*. L'équivalent des 29 composants constituant ce système sont remplacés par le MicroSPARC séquencé à 50 MHz. Réalisé en technologie 0.8 μm avec un procédé CMOS à deux couches de métal, le MicroSPARC est issu de l'utilisation d'outils de conception automatique. Aussi, bien que le degré d'intégration soit moins dense, l'utilisation de ces outils a permis à *Sun* la commercialisation de ce processeur en 18 mois.

L'architecture de ce processeur est toujours d'actualité puisqu'un MicroSPARC 2 a été annoncé en mars 94 à la fréquence de 100 MHz avec des performances de l'ordre de 70 SPECint92 et 61 SPECfp92.

Par ailleurs, un MicroSPARC 3 est attendu pour la fin de l'année 96. Initialement prévu comme une simple mise à jour du MicroSPARC 2, les performances attendues étaient de l'ordre de 100 SPECint92. Cependant, la menace représentée par le Pentium d'Intel a obligé *Sun* à concevoir une nouvelle architecture qui devrait atteindre des performances doubles et rendre obsolète les architectures SuperSPARC et SuperSPARC 2. Peu de détails sont actuellement disponibles sur ce processeur. Il devrait privilégier une architecture superscalaire avec une fréquence de fonctionnement de l'ordre de 200 MHz. De-même que ses prédécesseurs, le MicroSPARC 3 inclura une interface SBus et *Sun* devrait ajouter également une interface PCI pour concurrencer les PC d'entrée de gamme basés sur des processeurs Pentium.

4.5 L'HyperSPARC

Annoncé en avril 1992, l'HyperSPARC (ou *CYM6222K*) constitue en fait une implémentation *multichip* de l'architecture SPARC-V8 conçue par *Ross Technology* filiale de *Fujitsu*. Il est constitué de deux composants (*CY7C620*) regroupés sous le même conditionnement et reliés au reste du système par l'intermédiaire du MBus.

Le *CY7C620* est un processeur multicomposant constitué d'un processeur central (composant *CY7C620*), d'un contrôleur de cache (*CY7C625*) qui regroupe une unité de gestion de la mémoire et les étiquettes du cache secondaire (*CMTU*) et de deux (128 Ko) ou quatre (256 Ko) composants *CY7C627* constituant un deuxième niveau de cache. L'ensemble de ces composants peut être configuré comme un système monoprocesseur (niveau 1 du MBus) ou multiprocesseur (niveau 2 du MBus).

Le *CY7C620* est un processeur superscalaire. 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 Ko associatif par ensemble 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 *CY7C620* utilise un pipeline classique de cinq étages. L'unité entière inclut 136 registres configurés en 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 est à correspondance directe. Il est virtuellement adressé mais physiquement testé, facilitant ainsi le maintient de la cohérence dans un environnement multiprocesseur. La politique d'écriture mise en œuvre est configurable (différée ou simultanée). Ce second niveau de cache est commun aux instructions et aux données.

Le contrôle de ce cache se fait par l'intermédiaire du contrôleur de cache qui comprend également une unité de gestion de la mémoire responsable des traductions d'adresses virtuelles (32 bits) en physique (36 bits). Cette unité inclut pour cela une table de traduction d'adresse totalement associative de 64 entrées susceptible de supporter jusqu'à 4096 processus.

Ce processeur est réalisé dans une technologie CMOS à 0.65 μm à deux couches de métal et dispose de plusieurs fréquences de fonctionnement : 55 MHz, 66 MHz ou 80 MHz. La fréquence du MBus reste cependant limitée à 40 MHz. Les performances mesurées pour un système à 66 MHz sont de l'ordre de 64.6 SPECint92 et 85.5 SPECfp92.

En novembre 1994, *Ross* a annoncé un *HyperSPARC* à une fréquence de 100 MHz. Ce processeur a été choisi par *Sun* pour équiper ses SPARCstation 20 modèle HS11.

4.6 L'UltraSPARC

L'UltraSPARC est le premier processeur de *Sun Microsystems SPARC Technology Business* à implémenter la nouvelle architecture 64 bits SPARC-V9. Annoncé en septembre 1994, sa commercialisation devrait prendre effet fin 1995. Ce processeur superscalaire de degré quatre, totalement compatible avec les applications 32 bits déjà existantes est destiné au marché des applications hautes performances. Il privilégie une haute fréquence d'horloge (167 MHz) et dispose en outre des caractéristiques additionnelles suivantes :

- un ensemble d'instructions spécifiques au traitement vidéo (compression et décompression MPEG-2, vidéo-conférence, traitement d'images, etc...) ;

– des instructions de lecture/écriture par bloc ;

– des compteurs de performances ;

– une instruction d'extinction du système.

Nous détaillerons dans une première partie l'implémentation matérielle de ce processeur avant de revenir sur les caractéristiques de ce nouvel ensemble d'instructions qui, on le rappelle, constitue un ajout à la norme SPARC-V9.

4.6.1 Implémentation matérielle de l'UltraSPARC

Comme on l'a dit précédemment, l'UltraSPARC est un processeur superscalaire de degré quatre. Les quatre instructions émises à un cycle donné appartiennent aux combinaisons suivantes :

– deux instructions entières ;

– deux instructions flottantes ;

– une instruction de lecture/écriture ;

– une instruction de branchement.

Les instructions sont émises dans l'ordre mais peuvent être exécutées dans le désordre. L'UltraSPARC met en œuvre la technique du superpipeline avec neuf étages d'exécution. La figure 4.3 détaille les diverses phases d'exécution des instructions.

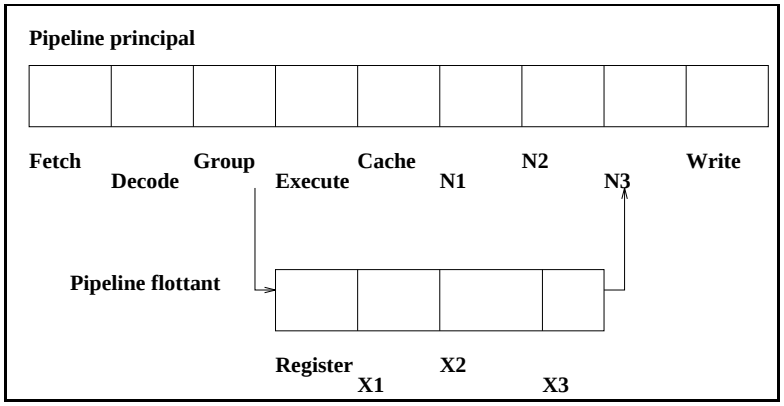

FIG. 4.3 - *Pipeline d'exécution de l'UltraSPARC*

Fetch - À cette étape, quatre instructions sont lues à partir du cache d'instructions ainsi que des informations de prédiction de branchement.

Decode - Les instructions sont décodées et placées dans un tampon d'instructions de 12 entrées qui permet de découpler l'exécution des instructions de leur chargement.

Group - Le banc de registres entiers est accédé en même temps que les quatre instructions les plus anciennes. Les conflits de dépendances et de ressources sont résolus à cet étage. Les instructions non consommées sont reconsidérées au cycle suivant.

Les pipelines entier, flottant et graphique diffèrent à partir de ce niveau. L'étage suivant dans le pipeline entier est un étage d'exécution où les instructions entières sont exécutées et les adresses virtuelles calculées. Les deux pipelines d'exécution entier ne sont pas identiques (en particulier les opérations de décalage ainsi que celles ayant trait aux codes conditions ne peuvent être exécutées que dans un pipeline).

Cache - Cet étage marque l'accès au cache de données ainsi qu'au TLB (accès à partir de l'adresse virtuelle). Les branchements sont également résolus au niveau de cet étage.

N1, N2, N3 - Durant l'étage *N1*, les défauts sur le cache de données sont détectés et sont envoyés vers le tampon de lecture (capacité de neuf entrées), en cas de succès la donnée est disponible. L'étage *N2* constitue un étage d'attente pour le pipeline entier vis à vis du pipeline flottant. Durant l'étage N3, les deux pipelines convergent et toutes les exceptions sont résolues à ce niveau.

Write - L'étage d'écriture marque la mise à jour des deux bancs de registres.

En ce qui concerne le pipeline d'exécution flottant, les instructions sont émises à la fin de l'étage de groupement, puis :

Register - Cet étage marque la lecture des registres flottants. Un décodage complémentaire des instructions reçues est également effectué.

X1, X2, X3 - Ces trois étages constituent les phases d'exécution des instructions flottantes. Toutes les opérations flottantes et graphiques, à l'exception de la division et de la racine carrée, sont pipelinées et ont une latence de une à trois cycles.

.

Comme on peut le constater, le pipeline entier contient trois étages supplémentaires de synchronisation avec le pipeline flottant. Cette approche élimine, selon les concepteurs de l'UltraSPARC, le besoin d'implémenter une file pour les instructions flottantes et facilite par ailleurs la gestion des exceptions effectuée de manière précise pour toutes les instructions entières et flottantes.

Les instructions sont chargées à partir d'un cache d'instructions de 16 Ko associatif par ensemble à deux voies. La politique de remplacement est basée sur un algorithme de type LRU. Il est à noter que ce cache est légèrement différent des caches traditionnels dans la mesure où un mécanisme de prédiction 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. En cas d'erreur, deux cycles de pénalités sont insérés. Ce cache est physiquement adressé et testé. Un *micro-TLB* de une entrée permet de conserver la traduction de la dernière adresse accédée.

Comme tous les processeurs destinés à de hautes performances, l'UltraSPARC met en œuvre une prédiction dynamique de branchement. Ce mécanisme prend ici la forme d'un cache d'adresse de branchement [4] double-port de 2 Ko. Ce cache est lui aussi légèrement différent de ceux habituellement utilisés. Effectivement, en plus des bits d'état de la prédiction, et de l'adresse cible, il

4. Appelé dans la terminologie de l'UltraSPARC : *Next Field Ram*.

contient également les bits de prédiction de l'ensemble ainsi que les bits LRU. La prédiction utilisée sur l'UltraSPARC s'appuie sur l'algorithme de Smith à deux bits. La *Next Field Ram* contient un champ de prédiction à deux bits pour tout bloc de deux instructions et un champ d'adresse cible pour quatre instructions.

L'UltraSPARC contient neuf unités d'exécution concurrentes :

- deux pipelines d'exécution entiers (non symétriques) ;
- deux pipelines de lecture/écriture (mais seule l'une de ces instructions peut être séquencée à chaque cycle) ;
- trois pipelines d'exécution flottants (addition, multiplication, division) ;
- deux pipelines graphiques (addition et multiplications). Ils utilisent le même banc de registres que les instructions flottantes.

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 pour le contrôle des pipelines (validation des chemins de *bypass*, écriture des résultats, etc...).
Le banc de registres inclut huit fenêtres de registres (soit un total de 144 registres entiers selon la norme SPARC-V9) avec sept ports de lecture (quatre pour les instructions entières, deux pour l'unité de calcul d'adresses et un dédié aux instructions d'écriture vers la mémoire) et trois ports d'écriture.

Les instructions entières sont généralement effectuées en un cycle (étage *execute* de la figure 4.3). Les résultats sont alors inscrits dans une unité indépendante (la *completion unit*) qui permet d'attendre la fin d'exécution des instructions susceptibles de provoquer une exception. Si les instructions finissent leur exécution normalement le banc de registres est alors mis à jour. Des circuits de *bypass* permettent cependant de prélever les opérandes directement dans la *completion unit*.

L'unité flottante et graphique. L'unité flottante et graphique (*FGU*) comprend cinq unités fonctionnelles et un banc de registres. À chaque cycle, deux opérations flottantes ou graphiques ainsi qu'une opération de lecture/écriture flottante peuvent être exécutées. Toutes les opérations, à l'exception de la division et de la racine carrée, sont pipelinées. De même que dans le cas de l'unité entière, les résultats issus des opérations sont inscrits dans la *completion unit* avant la mise à jour du banc de registres. 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*.

> À noter : la présence des deux unités graphiques d'addition et de multiplication responsable de l'exécution des instructions du *Visual Instruction Set* (voir le chapitre 4.6.2). Ces deux unités sont pipelinées et effectuent des opérations sur des pixels de 8 ou 16 bits.

L'unité de lecture/écriture. Comme son nom l'indique, cette unité est responsable des transferts de données entre les deux bancs de registres et la hiérarchie mémoire. Elle inclut pour cela un tampon de lecture de douze entrées et un tampon d'écriture de huit entrées. Ce dernier dispose d'une faculté de compression des requêtes lorsque celles-ci adressent le même bloc de 16 octets.

Le cache de données est à correspondance directe et a une capacité de 16 Ko. Il est virtuellement indexé et physiquement testé. Ce cache n'a qu'un seul port d'accès mais la RAM 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.

Un deuxième niveau de cache est obligatoire. Ce cache externe est utilisé pour les instructions et les données. Sa taille varie de 512 Ko à 4 Mo. Il est physiquement adressé et testé. Il implémente une protection par parité au niveau de chaque octet et supporte le protocole de cohérence de type *MOESI* pour les cinq états possibles ainsi que tout autre sous-ensemble (MESI, MSI). La pénalité d'accès à ce cache est de six cycles. À noter l'utilisation de *Synchronous SRAM* avec le même temps de cycle que le processeur.

L'unité de gestion mémoire. L'UltraSPARC implémente deux unités de gestion de la mémoire indépendantes pour les instructions et les données. Chacune de ces MMU implémente une table de traduction d'adresses totalement associative de 64 entrées qui fournit la traduction d'une adresse virtuelle vers une adresse physique à chaque cycle. L'adresse virtuelle est codée sur 44 bits, l'adresse physique est quant à elle codée sur 41 bits. Il est à noter que pour accélérer le mécanisme de traduction d'adresse particulièrement vital pour les instructions, l'UltraSPARC implémente en plus un micro-TLB de une entrée pour les instructions.

Ces deux unités offrent une protection des données au niveau de chaque page. Quatre tailles de pages sont supportées : 8 Ko, 16 Ko, 512 Ko et 4 Mo. La gestion des TLB est assurée par logiciel.

l'UltraSPARC dispose de deux 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. On peut remarquer l'introduction des huit registres *Alternate global registers* qui servent aux opérations effectuées dans ce mode et qui accélèrent la prise en compte des exceptions.

De plus l'UltraSPARC introduit un troisième mode de fonctionnement appelé *RED_state* (pour *Reset, Error* et *Debug*). Ce mode est spécifique aux états précédemment cités ou en cas d'erreur grave. Dans ce mode, la traduction d'adresse pour les instructions est invalidée (les données restent gérées normalement).

Tests et mise au point. Les aspects de tests et de mise au point d'un microprocesseur sont rarement abordés. L'UltraSPARC est conforme à la norme *IEEE 1149.1* et dispose d'un port d'accès de test et d'un contrôleur. Une interface standard à cinq broches est implémentée. Un registre d'instruction huit bits supporte cinq instructions *publiques* et 70 instructions *privées*. Les instructions *publiques* concernent principalement l'identification du composant, les tests d'interconnexion, etc... Les instructions *privées* sont principalement utilisées pour accéder à des caractéristiques de test de fabrication, d'analyse de défauts, tests des éléments mémoire, etc...

Par ailleurs, l'UltraSPARC inclut des caractéristiques de mise au point concernant le fonctionnement du composant telles que :

- validation logicielle de points de surveillance au niveau des adresses (virtuelle ou physique) ;
- bus de sortie de 15 bits permettant de suivre la plupart des signaux internes ;

– contrôle de l'horloge.

Nous n'expliciterons pas plus ces diverses fonctionnalités dans la mesure où elles s'adressent principalement à des experts de conception matérielle.

L'UltraSPARC intégre 5.2 millions de transistors. Il est réalisé en technologie CMOS à 0.5 μm à quatre couches de métal. Séquencé à la fréquence de 167 MHz, les performances attendues sont de l'ordre de 250 SPECint92 et de 300 SPECfp92.

4.6.2 Les ajouts à la norme SPARC-V9

Comme nous l'avons dit précédemment, l'UltraSPARC comprend certaines caractéristiques n'appartenant pas à la norme SPARC-V9 :

– un ensemble d'instructions spécifiques au traitement vidéo ;

– des instructions de lecture/écriture par bloc ;

– des compteurs de performances ;

– une instruction d'extinction du système.

L'instruction d'extinction du système concerne la mise en veille du composant durant des périodes de moindre activité. Cette instruction s'affiche dans une politique d'économie d'énergie mise en œuvre par les USA [5] et permet de diminuer la consommation de 30 Watt à 20 mWatt.

En plus du registre *TICK* défini par la norme SPARC-V9 (registre destiné à compter les cycles d'horloge), l'UltraSPARC fournit deux registres 32 bits accessibles par logiciel destinés à suivre certains évènements (défaut de cache, mauvaise prédiction de branchement, etc...)

Les instructions de lecture/écriture par bloc concernent le transfert de données vers ou à partir de la mémoire par bloc de 64 octets. Ceci permet à l'unité centrale de soutenir une bande passante de 600 Mo/sec.

La dernière caractéristique que nous aborderons concerne l'introduction d'instructions spécifiques au traitement vidéo (nommées *Visual Instruction Set* selon la terminologie utilisée par l'UltraSPARC). Ces instructions, au nombre de trente, visent directement le domaine des applications multimedia. Elles sont toutes pipelinées et opèrent sur plusieurs données de 8, 16 et 32 bits en parallèle. Elles permettent des traitements d'images 2-D et 3-D, des compressions et des décompressions, etc...

Son successeur l'UltraSPARC II est annoncé pour le deuxième trimestre 96.

4.7 Le Sparc64

Développé par la société *Hal Computer Systems*, ce processeur a été annoncé en mars 1995 et devrait être commercialisé d'ici la fin de cette année. Il vise le marché des stations de travail haut de

5. *Environmental Protection Agency's Energy-Star Program.*

gamme et des serveurs et se place donc sur le même plan que l'UltraSPARC avec des performances comparables : 256 SPECint92 et 330 SPECfp92.

Créé en 1990 à l'initiative de *Andrew Heller*[6], *Hal* a rejoint le groupe *Fujitsu* début 1991. Cette fusion s'inscrit dans une politique globale de la part de *Fujitsu* d'être présent sur le marché des processeurs 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 processeur ne devrait pas être distribué sur l'ensemble du marché. Celui-ci sera plutôt utilisé dans un premier temps pour des besoins internes, au sein des systèmes commercialisés par *Fujitsu* ainsi que dans les deux autres compagnies où elle intervient : *Amdhal* et *ICL*.

Ce processeur met en œuvre une architecture découplée conforme à la norme SPARC-V9. Des caractéristiques originales de sa mise en œuvre prédestinent ce processeur aux applications hautes performances utilisant de vastes ensembles de travail ou une grande fiabilité (cas des serveurs d'entreprises).

Description générale

Le Sparc64 est une implémentation multicomposant (conditionnement *MCM*). Il est constitué de deux composants logiques et de quatre composants mémoire responsables des fonctionnalités suivantes :

- une unité centrale qui contient le cœur du séquencement superscalaire, un cache d'instructions de 4 Ko 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 Ko, chacun utilisant deux composants mémoire de 64 Ko.
- 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.

Chargement et séquencement des instructions

L'accès aux deux caches externes est pipeliné et a une latence de trois cycles. Ces caches sont tous deux virtuellement adressés et testés. Chaque ligne de cache (d'une capacité de 128 octets) se répartit sur les deux composants créant ainsi deux bancs par cache. Chacun des bancs peut être adressé individuellement permettant ainsi de servir deux requêtes par cycle. Les caches sont non bloquants et peuvent fonctionner avec deux défauts de cache en attente. Bien que les deux caches soient identiques, leur utilisation est cependant différente.

Les 128 Ko de cache d'instructions constituent un cache secondaire après les 4 Ko mis en œuvre sur l'unité centrale[7]. Ce plus petit cache a une latence de un cycle. Les deux caches sont accédés à chaque cycle. Le but du cache primaire est de répondre rapidement (en cas d'urgence, cas d'une mauvaise prédiction de branchement par exemple), tandis que le cache secondaire répond plus lentement, mais grâce à son taux de défaut peu élevé, permet de soutenir un débit effectif important. Cette approche est valide dès lors que le débit en instructions du cache secondaire est largement supérieur à la demande réelle du processeur.

6. Architecte ayant participé au développement de systèmes IBM RS/6000.

7. *Hal* appelle ce cache le niveau zéro. Nous conserverons cependant notre terminologie cache primaire et cache secondaire.

L'unité de chargement mise en œuvre sur le Sparc64 est relativement complexe. Elle fournit à l'unité de séquencement quatre instructions à chaque cycle.

L'unité de séquencement lit les opérandes sources, renomme les registres destinations et émet jusqu'à quatre instructions à chaque cycle vers les stations de réservation associées à chaque unité d'exécution. Cette unité peut gérer jusqu'à deux opérations flottantes, deux instructions de lecture/écriture et un branchement. Elle peut également traiter jusqu'à quatre opérations entières si deux au moins restent des calculs simples (pas de décalage ni de multiplication et division).

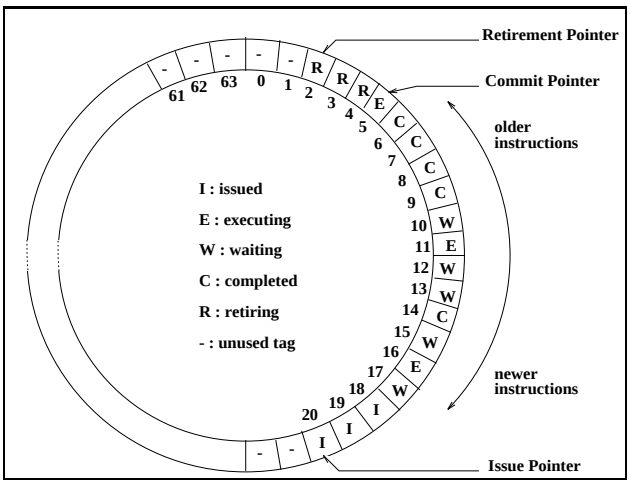

FIG. 4.4 - *Illustration du mécanisme de suivi des instructions sur le Sparc64*

Le Sparc64 est le premier processeur de cette architecture à mettre en œuvre un renommage de registres (gestion rendue plus complexe du fait des fenêtres). Il implémente pour cela 38 registres entiers et 24 registres flottants supplémentaires. Le renommage est une technique particulièrement efficace pour supprimer les fausses dépendances, situation fréquente dans les processeurs exécutant les instructions dans le désordre. Pour suivre la trace de l'exécution des instructions, une étiquette de six bits est assignée à chaque instructions. Cette mise en œuvre diffère du traditionnel tampon de réordonnancement utilisé sur la plupart des microprocesseurs (AMD K5, PowerPC 620, UltraSPARC). La largeur de l'étiquette permet d'avoir jusqu'à 64 instructions actives en même temps. La figure 4.4 explicite ce mécanisme. Sur cette figure, les instructions actives les plus anciennes sont prêtes à être retirées de la même manière que les instructions qui les précédaient. L'instruction *(5)* est toujours en cours d'exécution, empêchant les instructions *(6-9)* d'être retirées bien que leur exécution soit achevée. Les instructions suivantes sont dans diverses phases d'exécution, les instructions les plus récentes *(18-20)* sont en cours d'émission. Sur cet exemple seules 19 instructions sont actives. Trois pointeurs sont utilisés pour contrôler ce procédé :

- un pointeur d'émission qui désigne la dernière instruction émise (*Issue Pointer*). ;
- un deuxième pointeur désigne la dernière instruction en cours d'exécution (*Commit Pointer*) ;
- et enfin un dernier pointeur désigne l'instruction active la plus ancienne (*Retirement Pointer*).

Cette méthode est plus simple que le tampon de réordonnancement où toutes les instructions actives sont sauvegardées avec leur résultat dans un vaste fichiers de registres mais nécessite de

disposer d'un mécanisme de renommage pour conserver les résultats des instructions exécutées de manière spéculative.

Les unités d'exécution

Après leur renommage, les instructions sont émises dans l'une des quatre stations de réservation ou dans l'unité de branchement (architecture découplée). Ces stations ont des capacités variables : huit entrées pour les unités entière, flottante et de calcul d'adresses, 12 entrées pour les opérations d'accès à la mémoire. Chacune de ces stations peut recevoir deux instructions par cycle. Les instructions attendent au sein de ces tampons la disponibilité de leurs opérandes avant d'être émises vers les unités d'exécution. Si plus d'une instruction peut être émise, la plus ancienne est alors choisie.

En ce qui concerne les instructions entières, deux opérations peuvent être émises par cycle. Deux unités d'exécution non symétriques sont implémentées (seule une ALU dispose d'un opérateur de multiplication et division). Deux calculs d'adresse peuvent également être émis vers deux autres unités d'exécution. Ces deux unités peuvent également accepter des opérations entières. Par conséquent jusqu'à quatre opérations entières peuvent être exécutées à chaque cycle.

Deux instructions flottantes peuvent être émises vers une unité flottante générale ou vers une unité de division flottante. L'unité flottante comprend un pipeline de quatre étages de multiplication-addition.

Pour répondre aux exigences de cette architecture, le banc de registres entiers dispose de dix ports de lecture et de quatre ports d'écriture. Le banc de registres flottants est quant à lui équipé de six ports de lecture et trois ports d'écriture. Un tel nombre de ports est difficilement compatible avec les spécifications de l'architecture SPARC qui impose un nombre de registres important. L'UltraSPARC évite ce problème grâce à une méthode de compression du banc de registres (banc de registres à trois dimensions), mais *Hal* na pas pu réutiliser cette technique. Aussi, seules quatre fenêtres de registres sont mises en œuvre au lieu des sept ou huit des processeurs SPARC classiques. La taille du banc de registres reste ainsi acceptable, la conséquence étant un transfert de données plus important avec la mémoire en cas de débordement de fenêtres de registres (après quatre appels de procédures emboîtées).

La gestion des branchements est toujours délicate sur une architecture découplée superscalaire où une mauvaise prédiction peut causer l'abandon d'un nombre important d'instructions.

Le Sparc64 implémente pour cela une table d'historique de branchement (*BHT*) de 1024 entrées basée sur une prédiction à deux bits (algorithme de Smith). Au moment du chargement des instructions, leurs adresses sont testées dans la BHT. Si un branchement est prédis pris, l'instruction contient suffisamment d'informations sur l'adresse cible (grâce à l'étape de prédécodage) pour initialiser un chargement à partir du cache d'instructions. Si l'adresse cible est dans le cache primaire d'instructions, aucun cycle n'est perdu si la prédiction est juste. La présence de *delay slot* contribue à augmenter la complexité de gestion des branchements.

Pour prédire les adresses de retour de procédures, le Sparc64 implémente une table d'adresses de retour de quatre entrées indexée par le pointeur de la fenêtre courante.

Par ailleurs, des points de reprises (ou *checkpoint* selon la terminologie employée) sont effectués avant l'émission d'instructions qui peuvent altérer l'état visible de la machine. Il n'est pas nécessaire de sauvegarder directement le contenu des registres, seule la table de correspondance utilisée par

le renommage est conservée. Le Sparc64 supporte 16 niveaux de points de reprises. La mise en œuvre d'un tel mécanisme permet la restitution de l'état de la machine en un cycle. La table de mise en correspondance des registres est restitué à sa valeur précédente provoquant ainsi l'abandon des valeurs calculées de manière spéculative. L'unité de gestion des branchements maintient actives toutes les branches et s'assure que le résultat de l'unité fonctionnelle chargée d'évaluer la condition du branchement correspond bien à la valeur prédite. Si une mauvaise prédiction est détectée, l'unité de branchement la signale à l'unité responsable des points de reprises qui génère alors le procédé de restauration du système à un état correct.

Ce même mécanisme est utilisé pour la mise en œuvre d'une gestion précise des exceptions. Dans ce dernier cas, la restauration de l'état de la machine peut prendre plusieurs cycles. Le processeur commence par restaurer l'état fourni par le premier point de reprise situé après l'instruction fautive (ceci prend un cycle comme dans le cas des branchements). Puis, le CPU peut « défaire » quatre instructions tous les deux cycles jusqu'à atteindre l'instruction génératrice de l'exception (schéma similaire à celui du MIPS R10000 si ce n'est qu'il traite quatre instructions à chaque cycle).

La figure 4.5 présente le pipeline du Sparc64. Les quatre étages de base sont représentés en grisés. Les trois étages à la fin du pipeline schématisent la gestion du retrait des instructions (mais n'altèrent pas les performances) alors que les trois premiers représentent le préchargement des instructions à partir du cache secondaire. Ces trois étages sont transparents pour le CPU tant que le préchargement est bon, sinon ils introduisent une pénalité dans le cas d'un branchement absent du cache primaire.

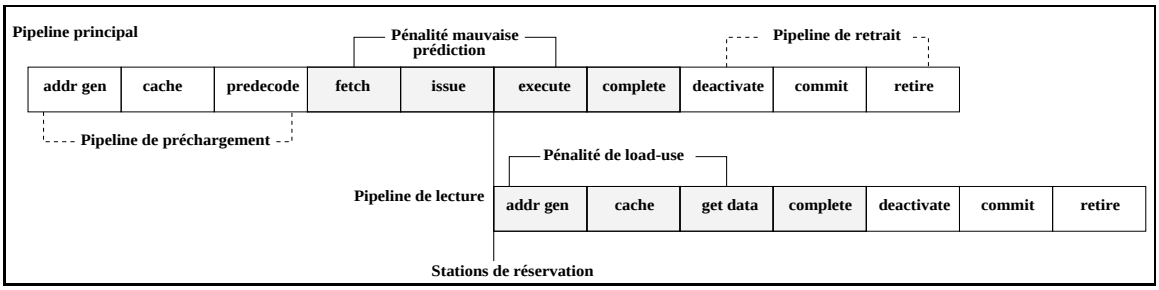

FIG. 4.5 - *Pipeline du Sparc64*

Le pipeline des instructions de lecture de la mémoire ajoute également trois cycles de pénalité si elles ne peuvent pas être anticipées.

Il est à noter que le Sparc64 supporte deux des modes mémoire de la norme SPARC-V9 (voir chapitre 4.1.1). Le mode *Total Store Ordering* et le mode *Relaxed Memory ordering*. Les opérations de lecture/écriture attendent dans un premier temps le calcul des adresses avant d'être émises au rythme de deux instructions à chaque cycle. Du fait de la division du cache de données en deux bancs, une unité est implémentée pour chacun des bancs pair et impair.

L'unité de gestion de la mémoire

L'unité de gestion de la mémoire gère les communications entre le CPU, les caches et le reste du système. De plus, elle contrôle le bus d'interface avec la mémoire, de 128 bits, et le bus d'entrées/sorties, de 64 bits. Ce partitionnement permet au CPU et aux caches de ne fonctionner qu'avec

un espace d'adressage virtuel, alors que la MMU maintient la cohérence de ces adresses avec le reste du système.

À la différence des autres processeurs, le Sparc64 utilise la totalité de l'adressage 64 bits et supporte trois niveaux d'adressage. Les adresses virtuelles sont dans un premier temps mappées vers des adresses logiques qui sont utilisées par le sous-système d'entrées/sorties ainsi que par la mémoire distribuée dans un environnement multiprocesseur. Ces adresses logiques sont alors converties en adresses physiques pour le système de mémoire local. Les adresses logiques fournissent une méthode pratique pour mettre en correspondance et partager les composants d'entrées/sorties et la mémoire (cette technique traduit une approche différente de celle d'IBM qui utilise une mémoire segmentée).

Pour assurer rapidement ces traductions d'adresses, le Sparc64 implémente deux types de tables de traduction communes pour les instructions et les données :

- une table de traduction assurant la mise en correspondance des adresses virtuelles et logiques de 128 entrées, totalement associative (la *VLB* pour *View Lookaside Buffer*) ;
- une table de traduction classique (*TLB*) de 1024 entrées, associative par ensemble à quatre voies et qui fournit la conversion de l'adresse logique en adresse physique.

Le fait d'avoir séparé le CPU de l'unité de gestion de la mémoire permet d'augmenter la capacité de ces tables et améliore les performances sur les applications nécessitant de larges espaces de travail.

Par ailleurs, dans la mesure où le CPU peut générer trois adresses à chaque cycle (une pour le cache d'instructions et une pour chacun des bancs du cache de données), l'unité de gestion de la mémoire inclut un *micro-TLB* de huit entrées à trois ports. Ce *micro-TLB* effectue directement la traduction de l'adresse virtuelle en adresse physique en même temps que l'accès au cache. Si la traduction recherchée n'est pas présente elle est alors fournie par la combinaison VLB/TLB. Les défauts de TLB/VLB sont gérés par matériel à travers le mécanisme de traduction défini par l'architecture SPARC.

Un contrôleur de mémoire externe est nécessaire pour générer les signaux de contrôle pour la DRAM. Lors d'un défaut de cache primaire, la pénalité est de l'ordre de 21 cycles selon l'implémentation.

Un bus de 64 bits (le *Hal I/O*) est utilisé pour relier le processeur aux composants d'entrées/sorties. Ce bus multiplexé fonctionne à la fréquence de 50 MHz. *Hal* développe actuellement des passerelles vers les bus SBus et SCSI.

Des caractéristiques de fiabilité

Le Sparc64 intégre de nombreuses caractéristiques permettant la détection et la correction d'erreur. Le cache primaire est protégé par parité, les données erronées étant relues à partir du cache secondaire. Le cache de données (ainsi que la mémoire principale) est protégé par ECC à raison de huit bits par mot de 64 bits. Ces huit bits permettent la correction d'erreur simple et la détection d'erreur double.

Le VLB et le TLB, ainsi que tous les signaux qui traversent les circuits sont également protégés par parité. Les erreurs pouvant être corrigées sont enregistrées et signalées selon un faible niveau de priorité. Les erreurs non recouvrables par matériel provoquent une interruption à haute priorité et font passer le CPU en mode *RED_state* selon la norme SPARC-V9. À partir de cet état, le Sparc64 peut charger des instructions de diagnostic à travers un bus série géré par la MMU. Ces instructions sont exécutées séquentiellement. Cette procédure, bien que lente, permet le diagnostic d'erreur dans

le cache, dans la mémoire principale ou dans le CPU. Dans la plupart des cas, ce diagnostic peut invalider l'unité défectueuse, abandonner les processus affectés et redémarrer le système.

Hal espère que ces caractéristiques de fiabilité offriront au Sparc64 un avantage sur les autres processeurs dans le domaine des applications nécessitant une haute intégrité des données. Seuls les systèmes à base de processeur IBM RS/6000 atteignent un niveau équivalent de fiabilité. Ceci n'est pas sans rappeler que le fondateur de *Hal* a participé au développement de ces systèmes

Caractéristiques technologiques

Nous passerons rapidement sur les détails d'implémentation physique de ce processeur. Il est implémenté sous la forme d'un conditionnement *MCM* céramique basé sur la technologie des *mainframe* de *Fujitsu*. Il est conçu selon une technologie CMOS à 0.4 μm avec quatre couches de métal et contient au total 21.9 millions de transistors, l'essentiel étant constitué par la mémoire (2.7 millions pour le CPU). Séquencé à la fréquence de 154 MHz, il dissipe environ 60 Watt.

Chapitre 5

Les microprocesseurs xxx86

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*[1]) développeurs de clones ainsi que ses récentes « mésaventures » avec le diviseur flottant du microprocesseur Pentium ont contribué à faire connaître au grand public le nom de cette architecture.

Ce qui est peut-être moins connu, c'est que Intel a également su s'imposer sur le marché des systèmes embarqués et que les performances de ses puces (le Pentium et ses successeurs), lui permettent même désormais de viser le marché des stations de travail. Le futur P6 (annoncé sous le nom de *Pentium Pro*), processeur aux performances encore supérieures devrait permettre à cette architecture de conserver sa suprématie jusqu'à l'avènement du P7, issue d'une collaboration avec *Hewlett Packard*, qui devrait marquer la fin de l'architecture CISC xxx86.

Nous étudierons dans une première partie les caractéristiques de l'architecture xxx86 à travers l'exemple du Pentium avant d'aborder différents « clones » de cette architecture à travers les processeurs d'*AMD* et de *Cyrix* principaux concurrents. Ce chapitre nous permettra de voir les diverses solutions apportées au problème de la compatibilité avec l'architecture CISC d'Intel. Puis, dans un chapitre de perspectives, nous aborderons l'étude du processeur Intel P6 qui sera annoncé en novembre 1995.

Le premier processeur d'Intel date de 1971. Environ 75% des ordinateurs personnels utilisés à travers le monde aujourd'hui sont à base de microprocesseurs d'architecture Intel[2]. Intel fournit à cette industrie les composants, les cartes, les systèmes et logiciels associés.

Les activités d'Intel s'articulent autour de trois axes :

- la production de processeurs. Ceci concerne la conception et le développement des processeurs ainsi que celle des périphériques associés à leur fonctionnement (cartes mères, contrôleur de cache, RAM, etc...).
- les produits de communication et réseaux.
- le marché des semiconducteurs (les mémoires DRAM, les mémoires Flash, les contrôleurs, etc...).

1. *Advanced Micro Devices.*
2. Source Intel.

5.1 L'architecture CISC d'INTEL

Il serait vain de parler de la totalité des processeurs développés par Intel. L'un des premier processeurs monocomposants fut le *Intel 4004* (1971), utilisé comme calculateur. Les données étaient codées sur quatre bits, les instructions (au nombre de 46) avaient huit bits de long. Les instructions et les données étaient séparées en mémoire (1 K de données et 4 K de programme, soit un PC sur douze bits). Ce processeur implémentait également 16 registres généraux de quatre bits (ou huit de huit bits).

Le 8080 fut le successeur du 8008 (contrôleur de terminal, similaire au 4040 version améliorée du 4004). Ce processeur avait un espace d'adressage sur 16 bits et un bus de données de 8 bits. Il disposait également de sept registres de huit bits (dont six pouvaient être combinés en trois de 16 bits) et d'un pointeur de pile de 16 bits vers la mémoire.

L'Intel 8086 (1978) est basé sur le design des processeurs 8080/8085, avec un jeu d'instructions similaire mais étendu à 16 bits. L'unité d'interface du bus permettait de charger les instructions à travers une file d'instructions de six octets. L'exécution et le chargement étaient donc concurrentes sous la forme d'un pipeline primitif. La taille des instructions variait de un à huit octets. La série des processeurs 8086 fut choisie par IBM pour équiper ses systèmes. Bien que d'autres alternatives s'offraient aux concepteurs (tel le 68000 de Motorola utilisé plus tard dans les *IBM Instruments 9000 Laboratory Computer*) le choix de ce composant s'explique par un choix plus vaste de composants déjà disponibles et par le fait qu'IBM en possédait déjà les droits de production.

Après le 80286, dernier processeur Intel 16 bits, le 80386 (1985) a étendu cette architecture aux 32 bits, a ajouté une MMU et de nouveaux codes opérations. Le 80486 a ajouté des pipelines, un cache interne ainsi qu'une unité flottante intégrée et le doublement de l'horloge. Le Pentium (fin 1993) est un processeur superscalaire (jusqu'à deux instructions à chaque cycle dans les deux unités entières et l'unité flottante). « Pentium » fut le nom donné par Intel au processeur 80586 car il ne pouvait pas légalement protégé le nom « 586 » pour prévenir d'autres compagnies de l'utiliser, et de ce fait, le compatible Pentium développé par *NexGen* est appelé le Nx586.

Le P6, dernier processeur de la famille xxx86 devrait être officiellement lancé cet automne. Il constitue un nouveau seuil de performances pour cette architecture et met en œuvre des techniques avancées d'architecture de microprocesseurs.

Intel, en collaboration avec *Hewlett-Packard*, a commencé le développement d'une nouvelle génération de processeurs (compatible avec la série 80x86, vraisemblablement à travers un traducteur d'instructions). Ce futur processeur (successeur du P6) reposera sur la technologie VLIW[3]. Grâce à cette technologie, le P7 (ou PA-9000) devrait exécuter jusqu'à un milliard d'instructions par seconde. Ce processeur devrait être capable d'exécuter les codes Intel et ceux issus de l'architecture PA-RISC sans recompilation.

L'approche VLIW marque un retour à la simplicité des premières architectures RISC par rapport aux architectures de ces dernières années. Effectivement, celles-ci sont devenues de plus en plus complexes du fait de l'augmentation de la vitesse d'exécution et de la nécessité d'utiliser des techniques de prédiction de branchement de plus en plus sophistiquées. L'approche VLIW simplifie le design de l'architecture en déplaçant la complexité de l'ordonnancement des instructions vers le compilateur.

Il semblerait que HP prévoit d'utiliser deux niveaux de compilateur qui produiraient un code intermédiaire (ou *meta code*). Le code compilé serait optimisé pour un composant particulier. Un

3. *Very Long Instruction Word.*

nouveau compilateur serait donc introduit avec chaque nouveau composant. Ceci pose le problème de la compatibilité binaire !

5.2 Caractéristiques de l'architecture Intel

Avant d'étudier le jeu d'instructions, il nous paraît intéressant de détailler quelques caractéristiques de cette architecture afin de mieux comprendre le sens de certaines instructions de gestion de la mémoire ou autre.

Dans la mesure où l'ensemble des processeurs Intel assurent tous une totale compatibilité binaire, nous avons décidé d'étudier cette architecture à travers celle du processeur Pentium qui représente l'état de l'art de cette architecture (la documentation du P6 n'étant pas encore disponible).

5.2.1 Modes de fonctionnement

Le processeur Pentium a deux modes d'exécution primaires et un mode de gestion du système (dédié au contrôle de la consommation ainsi qu'à certaines caractéristiques des OEM[4]). Ces modes sont :

- *Le mode protégé.* C'est le mode natif de ce processeur qui fournit les meilleures performances. Dans ce mode toutes les instructions et les fonctionnalités de l'architecture sont disponibles. Des applications de type *real-address mode 8086* peuvent également être directement exécutées en mode protégé sous un environnement multitâches. Cette caractéristique est connue sous le nom de « mode virtuel 8086 », bien qu'il ne s'agisse pas d'un mode du processeur mais d'un attribut qui est validé pour l'exécution d'une application spécifique.

- *Real-address mode* (également appelé *Real mode*). Ce mode est équivalent à l'environnement de programmation du processeur Intel 8086 avec juste quelques extensions. Lors de l'initialisation, le processeur entre dans ce mode avant qu'une instruction le fasse basculer en mode protégé.

5.2.2 Organisation de la mémoire

L'organisation de la mémoire concerne la segmentation et la pagination. L'un ou l'autre de ces mécanismes, voire les deux, peuvent être utilisés simultanément par les processeurs Intel.

Une adresse issue du programme est appelée une « adresse logique ». Les mécanismes matériels liés à la segmentation traduisent cette adresse logique en une adresse vers un espace d'adressage non segmenté, continu, appelé « adresse linéaire ». Puis les mécanismes de pagination convertissent cette adresse linéaire en une adresse physique.

La segmentation

Afin de bien comprendre ce qui va suivre, il est intéressant de revenir sur le concept de segmentation. L'adresse logique est vue comme un déplacement à l'intérieur du segment. À chaque segment est associé un descripteur qui contient son adresse de base, sa taille limite ainsi que des informations de protection. Si le déplacement ne dépasse pas la taille limite, et si aucune autre

4. *Original Equipment Manufacturer.*

condition n'existe qui pourrait empêcher la lecture du segment, alors le déplacement et l'adresse de base sont ajoutés pour former une adresse linéaire.

Deux types de modèles de mémoire peuvent être utilisés :

- un modèle de mémoire uniforme ou non segmenté. Ce modèle de mémoire est le plus simple. Bien que le mécanisme de segmentation ne puisse pas être invalidé, le même effet peut être obtenu en conférant à tous les segments la même adresse de base. Ceci a pour conséquence de faire pointer vers le même espace d'adressage l'ensemble des opérations mémoire.

- le modèle de mémoire segmenté. Avec ce type d'organisation, l'espace d'adressage est découpé en 16 K segments pouvant contenir jusqu'à 4 Go de mémoire chacun, ce qui représente un total de 2^{46} octets. Le processeur mappe alors les 64 To vers l'espace d'adressage linéaire de 2^{32} octets selon un mécanisme de traduction d'adresse transparent pour le programmeur : adresse linéaire = adr. base du segment + adr.virtuelle.

L'intérêt d'un modèle de mémoire segmenté est que les déplacements dans l'espace d'adressage sont testés individuellement et que l'accès à chaque segment peut être individuellement controlé. Un pointeur vers un espace d'adressage segmenté est donc constitué de deux parties :

- un sélecteur de segment, qui est un champ de 16 bits qui identifie le segment;

- un déplacement dans le segment codé sur 32 bits.

La pagination

L'adresse linéaire fournie par le mécanisme de segmentation peut être directement utilisée pour pointer vers un vaste espace d'adressage physique, ou alors la pagination peut être utilisée pour simuler cet espace à partir d'une petite quantité de RAM et d'espaces disques. 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.

La pagination est différente de la segmentation dans la mesure où elle utilise des tailles fixes de pages (sur l'architecture Intel), contrairement au mécanisme de segmentation qui adapte la taille du segment à la structure de données qu'il contient. Ainsi, si seule la segmentation est utilisée, la structure de données qui est présente en mémoire physique sera entièrement en mémoire. Si la pagination est utilisée, une structure de données peut être partiellement en mémoire et partiellement sur le disque.

La figure 5.1 décrit le mécanisme de pagination du Pentium. Deux niveaux de tables sont utilisés pour adresser une page en mémoire physique. Chaque table est un tableau d'entrées de 32 bits qui est contenu, lui-même, dans une page de 4 Ko. Les dix bits de poids fort de l'adresse linéaire (c'est-à-dire l'adresse obtenue après segmentation) servent à indexer le répertoire de tables. L'entrée indexée contient l'adresse de la table de pages utile à la traduction d'adresse. Cette dernière est indexée par la partie *table* de l'adresse linéaire (10 bits). L'entrée désignée pointe sur l'adresse de base de la page à laquelle il faut ajouter le déplacement (les 12 bits de poids faible) de l'adresse pour obtenir l'adresse physique. Notons que ce mécanisme ne permet d'adresser que des pages de taille fixe (4 Ko). Il est possible d'adresser des pages de plus grande taille (jusqu'à 4 Mo), mais, pour des raisons commerciales, le mécanisme utilisé pour l'adressage de telles pages n'est pas diffusé par Intel. On peut toutefois remarquer qu'un répertoire de tables peut adresser un espace linéaire