



Le Microprocesseur Cell

Nicolas Morey-Chaisemartin <nicolas@morey-chaisemartin.com>

Table of Contents

Table des figures	2
I. Préface.....	3
II. Un peu d'histoire.....	3
III. Vue d'ensemble	4
III.1 PowerPC Processor Element	4
III.2 Synergistic Processor Element	5
III.3 Element Interconnect Bus.....	5
III.4 Memory Interface Controller	6
III.5 FlexIO Interface.....	6
IV. Les performances.....	7
V. Zoom sur quelques détails	8
V.1 Logique dynamique	8
V.2 Plus de programmation pour plus d'efficacité	11
VI. Les points faibles.....	13
VII. Perspectives d'avenir	14
VIII. Conclusion	15
IX. Bibliographies	16

Table des figures

Figure 1: Vue d'ensemble du Cell	4
Figure 2: Photo du Cell	4
Figure 3: Topologie de l'EIB	5
Figure 4: Système à 4 Cells	6
Figure 5: Porte NOR en CMOS.....	10
Figure 6: Porte NOR en pseudo-NMOS	10
Figure 7: Méthode d'accès au LS	11
Figure 8: Roadmap du Cell.....	14

I. Préface

Depuis quelques années maintenant, les architectures des processeurs varient peu. Nous avons vu l'arrivée des processeurs x86_64, de l'Itanium, de nombreuses fonctionnalités supplémentaires (gestion des consommations, de virtualisation), sans oublier les processeurs multi-coeurs mais aucun changement fondamental dans le paysage de l'électronique pour PC.

Les fréquences des processeurs commençant à stagner, les fondeurs se sont tournés vers le multicore et le 64 bits pour paralléliser les instructions et gagner en puissance de calcul, mais cela ne fait en pratique qu'empirer le phénomène de goulot déjà présent pour accéder aux données en mémoire centrale.

Le Cell Broadband Engine (couramment appelé Cell) prétend répondre à ses besoins. Nous allons donc nous plonger au coeur de ce nouveau microprocesseur pour en découvrir la composition et essayer de juger quels sont ses avantages et inconvénients.

II. Un peu d'histoire

(Source: [IBM 01])

Les discussions concernant le projet Cell ont débuté en été 2000, regroupant, à Tokyo, SCEI (Sony Computer Entertainment Incorporated), IBM et Toshiba pour un débat sur les choix architecturaux à faire. Il a été décidé que les architectures traditionnelles ne pouvaient plus suffire aux besoins de puissances (SCEI souhaitait un processeur 100 fois plus puissant que celui de la Playstation 2). Dès fin 2000, le concept était posé: une architecture Power 64bits, d'un contrôleur de flux mémoire et de processeurs « synergistic » afin d'obtenir une puissance de calcul suffisante.

En mars 2001 s'ouvrit le centre de recherche STI (Sony / Toshiba / IBM) au Texas.

L'essentiel du travail a été sur le besoin de performances du processeur, et surtout sur la manière de franchir les 3 barrières qui ralentissent aujourd'hui la progression de processeurs plus classiques:

- La latence mémoire
- La consommation élevée (qui amène des contraintes de refroidissement)
- La perte d'efficacité due à l'augmentation de la fréquence et de la profondeur des pipelines.

Mis en 2005 sur le marché, le processeur Cell fait beaucoup parlé de lui, en bien mais aussi ne mal.

Entrons maintenant dans le coeur de la bête.

III. Vue d'ensemble

(Sources: [IBM 02][WIKIPEDIA 01][ANANDTECH 01])

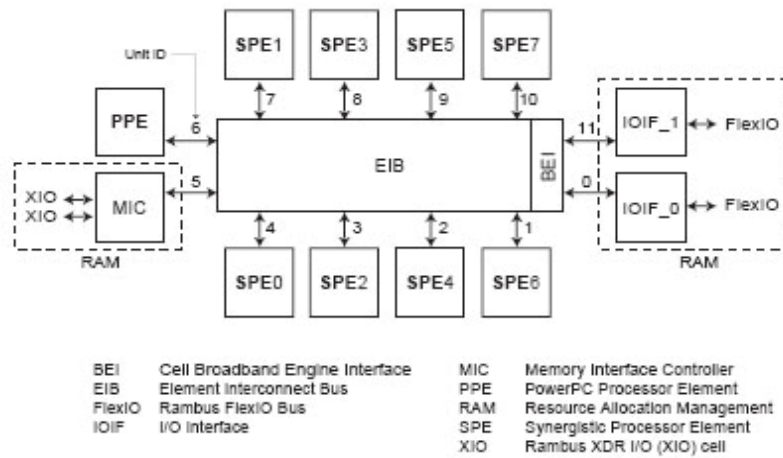


Figure 1: Vue d'ensemble du Cell

III.1 PowerPC Processor Element

Comme on peut le voir sur l'illustration 1, le Cell est composé de 9 processeurs (mais tous inclût dans la même puce).

Le processeur principal nommé PPE (PowerPC Processor Element) est un coeur PowerPC:

- 64 bits (supportant le set POWER d'instruction 32 bits),
- dual-thread (équivalent de la technologie Hyper Threading Intel, aussi appelé SMT)
- 32KB de cache L1 et 512KB de cache L2
- Support des instructions SIMD (Single Instruction Multiple Data) et VMX (alias AltiVec)

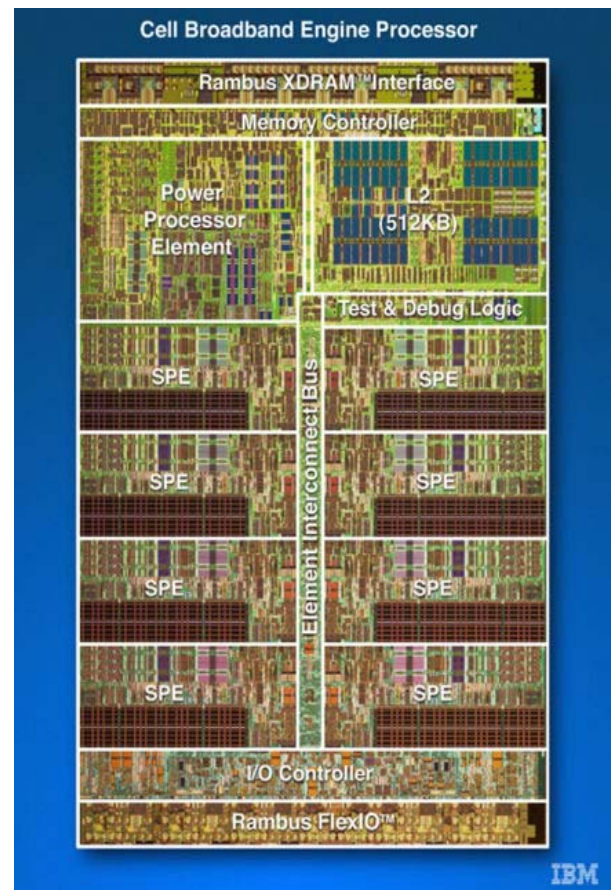


Figure 2: Photo du Cell

III.2 Synergistic Processor Element

A proximité se trouvent 8 SPE identiques (Synergistic Processor Element) optimisé pour des instructions SIMD. Ils disposent de 128 registres de 128 bits pour ces mêmes instructions.

Cependant, les SPE ne disposent pas de cache. Ils ont chacun muni d'une mémoire locale (Local Store, ou LS) entièrement gérée au niveau logiciel. Les SPE sont donc munis de commandes spéciales permettant de transférer des données depuis la MC vers les LS.

III.3 Element Interconnect Bus

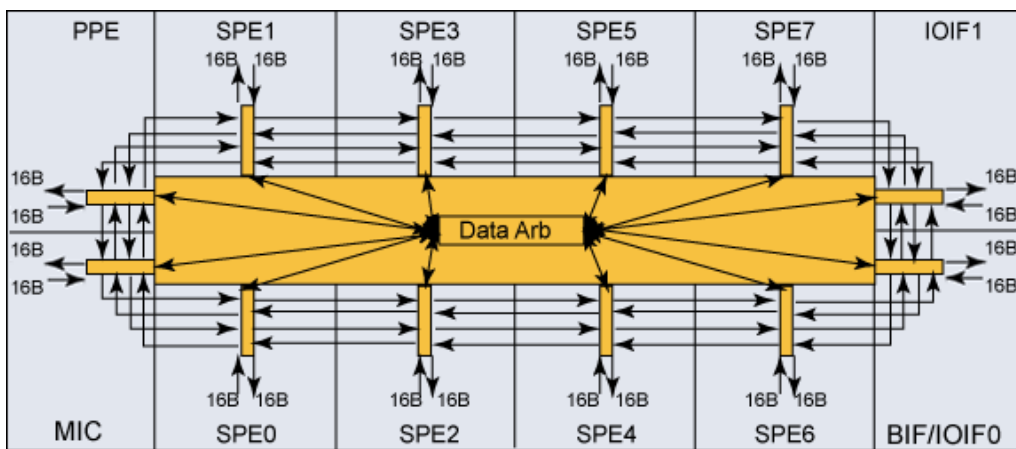


Figure 3: Topologie de l'EIB

Au centre de l'architecture, l'incontournable EIB (Element Interconnect Bus) par lequel passe tout les échanges d'un processeur à un autre ou vers la RAM et autres I/O.

Il est composé de 4 bus circulaire de 16 octets, transférant 128 octets à chaque fois (soit 8 beats). Chaque processeur est muni d'un bus d'entrée et d'un bus de sortie permettant ainsi de recevoir et d'émettre des données simultanément. Un arbitre au centre permet de gérer les accès concurrentiels au bus mais nous en reparlerons plus tard.

La fréquence théorique du bus annoncée par Intel est de 204.8Go/s et en pratique des pointes à 197Go/s ont été observées

III.4 Memory Interface Controller

Le choix de la technologie Rambus (plus précisément XDR RAM) a été fait ici. Cette technologie est connue depuis plusieurs années, mais a bénéficié d'une mauvaise image commerciale et a donc été un peu délaissée, mais elle revient de plus en plus au goût du jour avec les TV HD et même déjà la PS2.

La XDR, montée en dual channel, n'est cadencée ici qu'à 400MHz (peu comparé au 667MHz classique des DDR2) mais elle a l'avantage de pouvoir transférer 8 fois des données par cycle d'horloge. Le bus faisant 36bits de large (32 de données + 4 bits de contrôle), on obtient une bande passante de 25,6Go/s soit l'équivalent d'une carte vidéo et le double de la plupart des PC actuels.

III.5 FlexIO Interface

Le Cell dispose pour finir de 2 interfaces Rambus – FlexIO de 48 bits de large cadencé à 6,4GHz. L'avantage majeur de ces bus et qu'ils sont configurables dynamiquement par le Cell, permettant ainsi une utilisation dans un grand nombre de plate formes (points clé lors de la conception du processeur).

Ces bus donnent au Cell une bande passante de 76,8GB/s, soit 10 fois plus qu'un AMD Athlon 64

Il faut préciser que ce bus est de plus spécialement conçu pour permettre le chaînage de processeur Cell et permet donc d'obtenir une énorme bande passante pour une plate forme multi processeurs

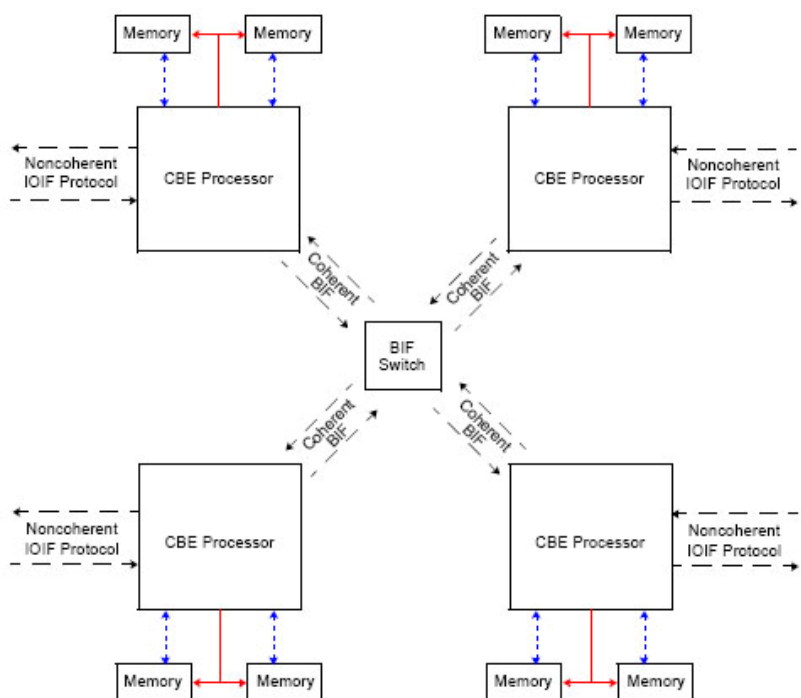


Figure 4: Système à 4 Cells

IV. Les performances

Nous venons de voir que les bus du Cell devançaient déjà largement la concurrence, mais qu'en est-il des performances de calcul?

Voici quelques chiffres :

Processeur	GFLOPS
PC Récent (Core 2 Duo)	~5
Itanium 2 à 1.5GHz	22,69
Cell	205

Il faut noter que ces chiffres sont ceux annoncés par les constructeurs (sauf pour les PC) et qu'ils ne sont pas forcément significatifs.

Un autre exemple basé sur les statistiques d'un logiciel de calcul scientifique distribué

([FOLDING 01]) Folding@Home:

Type de machine	Nombre de CPU	Puissance Totale (TFLOPS)	Puissance moyenne par CPU (GFLOPS)
PC / Windows	188431	179	0,95
PowerPC	8936	7	0,78
Mac (Intel)	6768	21	3,10
Playstation 3 (Cell)	38023	1148	30,20

On ne peut que constater ici la supériorité du Cell sur les autres processeurs. Il ne faut pas oublier que les machins sont complètement hétérogènes, et que le chiffre des PC par exemple est plus bas que la normale, les utilisateurs n'ayant pas tous un PC récent. A noter que les Mac Intel sont majoritairement composé de Core 2 Duo.

Le Cell est donc en pratique 10 à 20 fois plus performant qu'un processeur dual core x86_64 dernière génération pour du calcul scientifique (où le SIMD se rend très utile)

V. Zoom sur quelques détails

V.1 Logique dynamique

Comparons rapidement quelques chiffres de 2006:

Nom	Processeur Cell:	Processeur Intel Pentium D:
Gravure	90nm SOI manufacturing process	90nm strained silicon manufacturing process
Taille	221 mm ²	206 mm ²
Nombre de transistor	234M	230M
Fréquence d'horloge	> 4GHz	2,8 GHz - 3.2GHz
Puissance de pointe annoncée	205 GFLOPS	7.8 GFLOPS

Voici les nouveaux chiffre de 2008 :

Nom	Processeur Cell nouvelle génération:	Processeur Intel Core 2 Duo:
Gravure	65nm SOI manufacturing process	45nm strained silicon manufacturing process
Taille	140 mm ²	107 mm ²
Nombre de transistor	250M	291M
Fréquence d'horloge	4-6GHz	2,6 GHz - 3.2GHz
Puissance de pointe annoncée	384 GFLOPS	48 GFLOPS

Et les chiffres prévisionnels pour fin 2008

Nom	Processeur Cell:	Processeur Intel Quadcore (Yorkfield):
Gravure	45nm	45nm
Taille	~100 mm ²	214 mm ²
Nombre de transistor	234M	820M
Fréquence d'horloge	>6GHz	2,6 GHz - 3.2GHz
Puissance de pointe annoncée	>450 GFLOPS	~115 GFLOPS

Les chiffres sont donc quasi similaires concernant la taille, et le nombre de transistor (si on assimile le Cell à un unique core). Cependant, on constate que la fréquence observée est supérieur de près de 30% sur le Cell et augmente. On peut alors se demander comment le Cell peut tourner aussi vite alors que les fondeurs (Intel et AMD) semblent avoir de plus en plus de mal à faire grimper les vitesses d'horloges. Et se rabattent sur des architectures multicores qui ne font que renforcer le goulet qu'est déjà la mémoire.

A l'heure actuelle, la quasi-totalité des processeurs classiques utilisent les technologies CMOS et TTL pour les portes logiques.

Le CMOS bien que limité en rapidité et coûteux en transistors est toujours utilisé dans certains processeurs pour portable ayant besoin de basse consommation (une porte en CMOS non utilisée ne consomme rien ou presque).

Le TTL est quand à lui plus rapide mais consomme par contre beaucoup plus et reste malgré tout parmi les techniques les plus utilisées dans les processeurs.

STI a choisi d'utiliser pour le Cell un technique peu répandue (et donc coûteuse pour le design des nombreux opérateurs logiques nécessaires): le pseudo NMOS.

Cette technique a l'avantage :

- De fournir des portes logiques synchrones
- De pouvoir préparer l'opération en chargeant les transistors sans perdre le résultat de l'opération précédente. (utilisation de 2 horloges déphasées) et donc gagner en vitesse
- De diminuer le nombre de porte nécessaire par rapport à du CMOS (1 par entrée +2 pour les horloges)
- D'obtenir une faible consommation générale, inférieure à celle du CMOS (renforcé par l'utilisation du SOI qui tend à diminuer les fuites de courant dans les transistors)

Voici un exemple avec un NOR

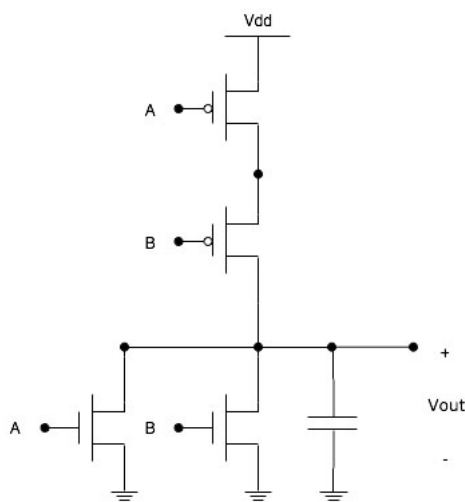


Figure 5: Porte NOR en CMOS

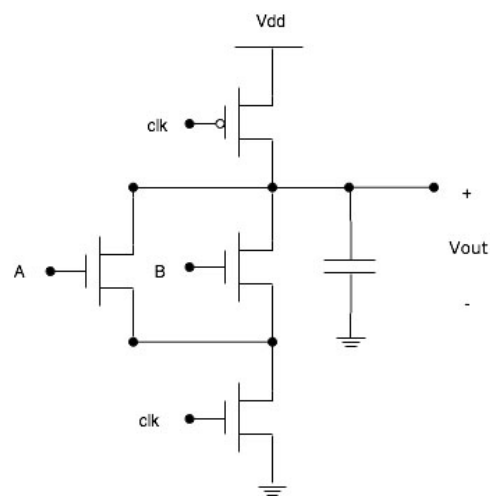


Figure 6: Porte NOR en pseudo-NMOS

STI a donc innové ici et trouvé une nouvelle solution gagnant à la fois en vitesse et en consommation. Les seuls points noirs de cette technique sont :

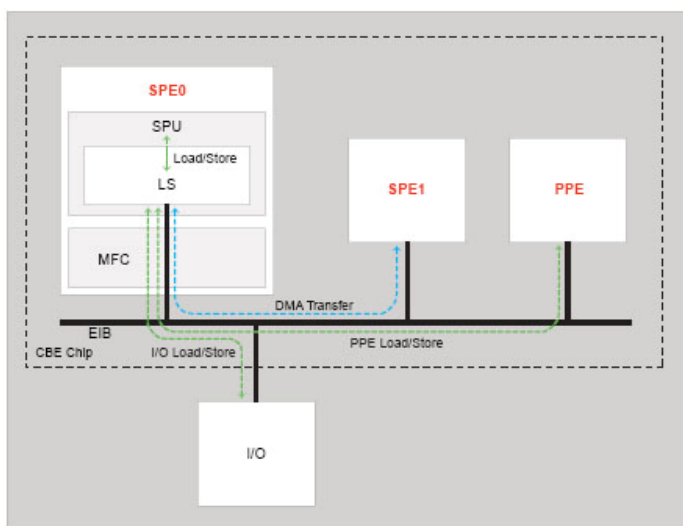
- La complexification de routines d'horloges puisque que chaque porte à 2 horloges déphasées, et dans le cas d'additionneur ou d'opérateur compliqué cela peut devenir extrêmement complexe mais STI l'a prouvé, ce n'est pas insurmontable.
- La nécessité de reprendre le design de toutes les portes logiques à 0 puisque qu'il n'existe aucune librairie aujourd'hui testée et éprouvée de portes en pseudo NMOS contrairement au librairie CMOS qui sont légions. Il faut pondérer cela avec le fait que pour les développements futurs de processeurs utilisant cette technologie STI aura déjà fait une grosse partie du travail de design.

V.2 Plus de programmation pour plus d'efficacité

La plupart des processeurs aujourd'hui sont des processeurs CISC et OoO (Out of Order). Cela signifie qu'en plus d'un jeu d'instructions très complet (et donc aussi plus lourd), le processeur est capable dynamiquement de réorganiser les lignes de codes pour optimiser son exécution. Le programmeur et le compilateur sont donc moins exploités et délèguent le travail au processeur qui essaye de travailler au mieux de ses capacités (typiquement si on a un cache miss, on peut exécuter une instruction qui n'a pas besoin de cette zone mémoire en attendant). Ce choix a été fait entre autre par Intel et AMD dans tous leurs processeurs.

Le problème de ce choix est le surcoût qu'il entraîne. En effet le décodage et la permutation d'instruction ne se fait pas sans mal et cela entraîne de lourde complexifications au niveau architectural. Pourquoi STI ont-ils fait un tel choix?

Le gros problème de l'utilisation d'un processeur OoO excepté son surcoût en transistor vient du fait qu'il est impossible de prévoir quand une instruction sera effectuée. Sur un ordinateur classique type x86 cela ne pose pas de problème. L'utilisateur ne verra pas la différence si son instruction est exécutée à au temps t ou à $t+10\mu s$. Mais se pose les problèmes des systèmes temps réel, de synchronisations.



DMA	Direct Memory Access	MFC	Memory Flow Controller
EIB	Element Interconnect Bus	PPE	PowerPC Processor Element
LS	Local Store	SPE	Synergistic Processor Element

Figure 7: Méthode d'accès au LS

STI a répondu à ce problème en utilisant des processeurs RISC (set d'instruction réduit) et In Order (qui exécutent dans l'ordre). L'effet immédiat et la simplification des SPE et permet donc d'en mettre 8 sur une puce de même taille qu'un dual core actuel, ce qui est encore impossible avec les processeurs AMD ou Intel.

D'autre part le « In Order » va permettre de faire du Cell un processeur aussi capable de faire du temps réel en supprimant de l'équation la seule inconnue: le cache. En effet lors d'une exécution normale il est impossible de savoir dans quel état est le cache, il dépend de l'algorithme utilisé, de sa taille, des différents programmes qui ont tournés avant.

Il est donc impossible de prévoir si l'instruction pourra être exécutée directement, ou si elle devra attendre qu'une nouvelle valeur soit chargée depuis la mémoire. STI a donc introduit ici une mémoire locale LS dont nous avons abordé le sujet un peu plus tôt.

Les SPE du Cell sont incapable de récupérer une instruction ou des données depuis la mémoire directement dans leur registre, comme le fait le PPE. Pour accéder à la mémoire, chaque SPE est muni d'un MMU (Memory Management Unit) fonctionnant comme un DMA. Le SPE dispose d'une file d'attente de requête mémoire (lecture ou écriture des données depuis la mémoire centrale/périphérique ou LS d'un autre SPE vers son propre LS). Ces LOAD et STORE sont en plus asynchrones. Il est donc possible pour le programmeur de connaître le contenu exact de sa mémoire, de planifier à l'avance le besoin de certaines données et de lancer le transfert pendant que le programme effectue autre chose.

STI a donc fait le choix ici de compliquer la tâche du programmeur et du développeur (nécessité de préfetcher les instructions avant de pouvoir y accéder par exemple) pour pouvoir gagner en performance derrière mais aussi en adaptabilité puisque le Cell peut aussi servir de processeur temps réel. Ce choix, couplé à la performance de l'EIB et aux fonctions SIMD permet d'excellentes performances sur des applications multimédias, les données nécessaires pouvant être « streamées » depuis la MC et les opérations vectorielles effectuées très rapidement.

Le résultat sont présent puisque des tests de Raycasting/Terrain Rendering ont été effectués (génération de terrain en 3D en 720p, application d'effets lumineux, compression MPEG au vol et envoie par réseau). Le Cell obtient un résultat de 36 fps (frame per second) contre 1 pour un bi-G5.

Les gros points noirs de ce choix mais qui touche plus à la programmation qu'à l'architecture est que ce choix qui fait la force du Cell est aussi une de ses plus grandes faiblesses. La différence entre l'exécution sur un le PPE et sur un SPE est flagrante, et à l'heure actuelle les compilateurs exploitent mal cette différence. Les programmes non spécifiquement développés sur le Cell n'utilisant parfois que le PPE même lorsqu'ils sont multithreadés.

VI. Les points faibles

Évidemment malgré les nombreuses innovations, il existe des points faibles.

Il n'y a pas grand choses à reprocher aux SPE si ce n'est le manque actuel de compilateur capable de les exploiter au mieux à partir d'un code non optimisé.

(Source [PRESPEC 01])

Le gros point faible du Cell provient du PPE. Le retour vers des processeurs plus simple n'est pas une nouvelle, Intel a déjà fait de même avec l'Itanium mais en cherchant à augmenter l'ILP (Instruction Level Parallelism) où STI cherche à améliorer le TLP (Thread Level Parallelism).

Le PPE manquant de puissance et la norme Linux imposant à tout les processus d'avoir leur thread père tournant sur le PPE, ce dernier se retrouve souvent surchargé. Il aurait été préférable de garder l'optique RISC/In Order sur les SPE mais de préférer au moins du « OoO » sur le PPE afin de permettre une meilleure intégration des logiciels actuels.

Un autre défaut du Cell provient de l'EIB. Son débit est certes énorme en interne, mais 9 processeurs y accèdent en quasi permanence et comme sur les processeurs classiques on retrouve un goulot d'étranglement. Bien que beaucoup moins critique dans le Cell, le bus peut se retrouver rapidement saturé dû aux priorités. En effet le SPE numéro 1 est prioritaire sur le 2, etc... Ainsi si une application nécessitant beaucoup de données mémoires tournant sur le processus 1 va potentiellement empêcher les SPE suivant d'accéder en mémoire.

Le Hardware ne prenant pas en compte la surcharge du bus, bien que permettant de la détecter, ce problème trouvera sa solution au niveau software (encore inexistante, mais en cours de développement dans le cadre du projet de spécialité) mais cela implique de gros échanges mémoires pour migrer un processus d'un SPE vers un autre, et donc une perte de performances importantes

VII. Perspectives d'avenir

(Source [PPCNUX 01])

Il est une chose importante à préciser: Le Cell n'est pas qu'un processeur, c'est un nouveau type d'architecture comme l'est le x86 ou le PPC. Le Cell actuel n'est donc pas un produit final mais simplement le premier d'une, on l'espère, longue lignée de processeurs dans le même esprit.

Et comme tout projet, le Cell a bien sur une roadmap qui donne les grands axes de son développement

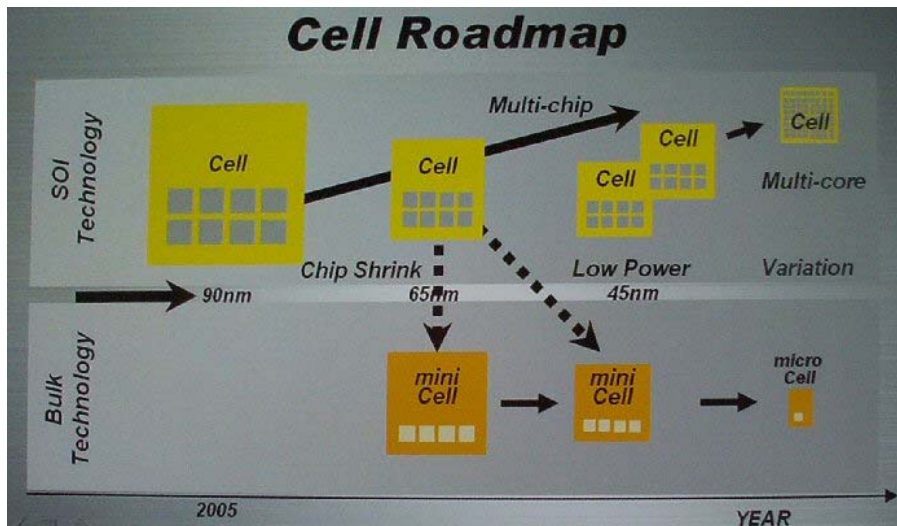


Figure 8: Roadmap du Cell

D'ici fin 2007:

- passage en gravure à 65nm et donc probablement augmentation de la fréquence d'horloge
- Sortie d'une nouvelle version « Enhanced Cell Broadcast Engine ». Peu d'information officielle sur cette version si ce n'est une amélioration des unités de calculs en flottant. Mais aux dire des forums et suite à certaines discussions avec des employés IBM, un nouveau PPE renforcé devrait compléter le tableau.

Dès 2008:

- Réalisation d'un mid-class Cell (muni d'uniquement 4 SPE) puis d'un micro-Cell très basse consommation (STI espère descendre en dessous de 10W) pour les systèmes embarqués, et tout les appareils mobiles (PDA, Téléphone, etc.)

En pratique la version 65nm est déjà sortie et disponible dans les dernières PS3. Cette version 65nm a en effet été légèrement modifiée pour accélérer les échanges intra-cores.

Pas de nouvelles concernant un Cell alléger pour le moment.

Prévu pour 2010:

- Passage en 45nm
- Nouveau processeur Cell 3 avec 2PPE et 32 SPE soit une puissance estimée d'1 TFLOPS

STI reste assez discret sur le développement du Cell, ne donnant que de petits indices sur les changements fondamentaux, et publiant la classique courses à la puissance (tel Intel qui prévoit les processeurs à plus de 80 coeur). Cependant les rumeurs sur le Cell 3 semble se vérifier et devrait se trouver dans la Playstation 4.

VIII. Conclusion

Malgré quelques défauts au niveau du PPE, le Cell reste une grande réussite. Sa puissance éprouvée par déjà de nombreuses entreprises et au travers de la Playstation 3 montre la puissance de cette architecture encore toute neuve.

On est en droit d'espérer après le chemin parcouru par STI depuis l'été 2000 qu'ils sauront tenir compte des remarques des nombreux développeurs, et renforcer leur PPE pour la prochaine version.

Il est important de préciser que le Cell n'est pas le premier processeur de ce type, Texas Instrument avait développé un processeur assez similaire, le MVP 320C8X, et le processeur de la Playstation 2 n'était pas si différent.

Grande innovation ou pas, le Cell a l'avantage de proposer une solution alternative aux x86 qui commencent à stagner, et même si les résultats sont très satisfaisants, il reste du chemin à parcourir avant d'obtenir la plate forme de demain qui saura s'installer dans toutes les demeures.

Espérons qu'il saura tenir ses promesses et surtout tenir têtes à Intel et AMD pour s'imposer sur un marché bien établi. Mais il semble que STI a les moyens et l'expérience pour faire aboutir ce projet autant sur le plan technique, que commercial.

IX. Bibliographies

** **IBM 01:** J.A. Kahle, Introduction to the Cell microprocessor, 2005

<http://www.research.ibm.com/journal/rd/494/kahle.html>

** **IBM 02:** IBM, Cell Broadband Engine Programming handbook v1,0, 2005

<http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/9F820A5FFA3ECE8C8725716A0062585F>

** **WIKIPEDIA 01:** Wikipedia US, Cell microprocessor, 2007

http://en.wikipedia.org/wiki/Cell_microprocessor

** **ANANDTECH 01:** Anand Lal Shimpi, Understanding the Cell Microprocessor, 2005

<http://www.anandtech.com/cpuchipsets/showdoc.aspx?i=2379>

** **FOLDING 01:** Folding@Home, Systems Statistics, 2007

<http://fah-web.stanford.edu/cgi-bin/main.py?qttype=osstats>

** **PRESPEC 01:** Fedy Aby Chahla, Le processeur Cell, 2005

<http://www.presence-pc.com/tests/Le-processeur-Cell-366/>

** **PPCNUX 01:** Bericht, The Cell Roadmap, 2007

<http://www.ppcnux.com/modules.php?name=News&file=article&sid=6666>

