

# Freescalé : les promesses de l'architecture Power – avec ou sans IBM...

**FTF 2007, Orlando** – IBM et Freescalé sont les deux pôles de la *Power Architecture*, mais si le premier règne sur le PowerPC avec pour cible les serveurs, le second s'active sur les systèmes embarqués (*embedded*).

Pour mieux comprendre cette architecture puissante et modulaire, un tour du propriétaire s'impose :



La technologie Power fait appel pour ses designs à trois référentiels de base : le **Book I**, *User Instruction Set Architecture* (UISA), pour les applications, le calcul, l'accélération, complété du **Book VLE** de Freescalé, pour l'encodage en longueur variable (*Variable Length Encoding*) ; le **Book II**, *Virtual Environment Architecture* (VEA), pour les bibliothèques et les modèles de programmation ; et le **Book III**, *Operating Environment Architecture* (OEA), pour les instructions et ressources du Superviseur/Hyperviseur.

Cette architecture est divisée en **catégories** ? *Base* (Book I et II), *Server* (Book III-S) et *Embedded* (Book III-E) – et chaque composant appartient à une catégorie. La création d'un processeur en architecture Power revient donc à implémenter un jeu de catégories. Par exemple, comme indiqué ci-dessus, les serveurs reprennent le Book III-S, tandis que les technologies embarquées (*embedded*) sont principalement basées sur le Book III-E.

Les catégories sont thésaurisées pour les applications des domaines recherchés. Ainsi la base reste commune, ce qui assure la compatibilité binaire.

A partir de cette base, les designers et les fondeurs peuvent apporter leur contribution. Les APU (*Auxiliary Processing Units*) de Freescalé, par exemple, viennent compléter les jeux de fonctionnalités, mais tout en continuant bien évidemment d'adresser des catégories, dont certaines sont hiérarchisées.



*L'apport des APUs de Freescalé au Book I*



*L'apport des APUs de Freescalé au Book III-E*

>>> Page 2 : *Power ISA, une architecture RISC*

## **Power ISA, une architecture RISC**

Power adopte une **architecture RISC** pour charger et conserver les données. La multiplication des

registres augmente en effet les performances. Elle permet également de disposer de registres spéciaux – Counter Register pour contrôler les *loops* (boucles), Link Register pour les liens des sous-routines, Time Base, Accumulator pour accumuler les opérations, Status registers pour diverses opérations, etc.). De plus, les instructions sont » triadiques« , c'est à dire que deux sources alimentent une destination.



### *Le mode registres du Power*

Les instructions ont une taille fixe, à l'exception de VLE de Freescale qui autorise la taille variable pour les applications qui demandent du code très dense. Le calcul est en virgule flottante et respecte la précision IEEE-754. Le support des instructions et caches de données est d'origine Harvard, ainsi que le cache unifié.

Le support comme l'adressage sont 64 bits, ainsi, que le traitement des données (*computing*). Power dispose d'un modèle de mémoire flexible, avec des instructions barrières pour organiser l'ordre des données pour le multi processeurs et le 'multi devices', pour le code multiprocessing et les pilotes des appareils.

Enfin, avec la version 2.06 attendue pour la fin 2008, les hyperviseurs, réservés aux serveurs, vont arriver sur l'embedded. Ils pourront être gérés dynamiquement et répartis sur les cœurs.



>>> Page 3 : *L'implémentation Power par Freescale*

## **L'implémentation Power par Freescale**

Freescale a défini une architecture spécifique, qui bien évidemment respecte les spécifications imposées par les Books. Elle est définie par *l'Embedded Implementation Standards (EIS)* et permet de définir de nouvelles implémentations pour l'exploitation des processeurs Power de Freescale.

La plupart des EIS reprennent les spécifications Power ISA du Book E. Les cœurs des processeurs reprennent donc les catégories du PowerPC, ainsi que de nouvelles catégories dédiées à l'embarqué (embedded), pour certaines optionnelles, qui étendent ses fonctionnalités. La gestion des interruptions et des exceptions, par exemple, supporte le synchrone et l'asynchrone.



Pour répondre aux attentes de *l'embedded*, Freescale a également soigné les problématiques de 'timing', proposant plusieurs solutions d'incrémentation, ou plutôt de décrémentation, la sélection des intervalles, des options d'auto (re)lancement et de surveillance active qui peut déclencher des interruptions critiques.

Les fonctionnalités de 'debug' sont également critiques dans l'embarqué. Un registre est dédié à ces opérations, *Debug Control Registers (DCCR0/1)*, qui si un événement intervient le poste dans *Debug Status Register (DBSR)*. Le debug externe est également proposé, ce qui permet de contrôler les cœurs

de l'extérieur.

Evoquons également la technologie AltiVec, issue de la catégorie Vector. Elle offre 32 registres 128 bits dédiés très rapidement accessibles qui peuvent être « mixés » avec les exécutions intégrées ou en virgules flottantes, mais qui permettent d'exécuter des opérations logicielles plus sophistiquées, comme de disposer de plus d'espace, de traiter plus de données en parallèle, et de séparer les registres.



On peut ainsi disposer de 16 voies parallèles en 8 bits à 4 voies parallèles en 32 bits, permuter les unités, cumuler des données de deux registres vers un registre cible unique, etc. C'est fonctionnalités ne sont cependant disponibles que sur le Power e600.



Relevons encore quelques briques importantes de la technologie Power: le moteur de traitement des signaux SPE (*Signal Processing Engine*), le calcul en virgule flottante embarqué, la densité du code, ou encore la technologie VLE (*Variable Length Encoding*) spécifique à Freescale...

Le site du fondeur ([www.freescale.com](http://www.freescale.com)) dispose d'une documentation très riche. La démonstration est faite, en revanche, de la richesse de l'architecture Power et de son intérêt pour le monde de l'embarqué (*embedded*).

Au delà des processeurs pour PC et serveurs, il existe un monde riche, celui de l'électronique embarqué sur les objets de l'industrie ou de notre quotidien, dont les performances sont ou seront certainement plus riches que nos ordinateurs. On comprend mieux pourquoi des acteurs comme IBM ou Intel s'y intéressent?



>>> Page 4 : *Un peu d'histoire et de futur sur le Power...*

### **Un peu d'histoire et de futur sur le Power...**

La technologie Power a été développée à l'origine par IBM, dès 1991, puis reprise en partenariat avec Motorola pour le PowerPC 'Classic' dans ses versions 1.0, puis 1.10, sur la base des Books 1, 2 et 3, des référentiels qui définissent les architectures et permettent de développer des fonctionnalités spécifiquement destinées à l'embarqué.

En 1999, le PowerPC devient Power ISA 64 bits et les référentiels sont fusionnés sur le Book E. Les deux technologies, PowerPC et Power, continuent cependant d'évoluer en parallèle. L'architecture PowerPC 2.02 est annoncée en 2003 par IBM, qui adopte les Books 1, 2 et 3. Mais elle reste très proche des précédentes versions.



Pour révolutionner le système, il faut attendre 2005 et le PowerPC Book E 1.9, qui paradoxalement

n'a pas été publié. En réalité, cette version va servir de base pour l'architecture Power ISA 2.03. A partir d'ici, les technologies Power signées IBM et Freescale vont pouvoir commencer à converger.

De son côté, après la création de VLE, Freescale confirme donc en 2006 la convergence avec Power 2.03, une technologie basée sur la version 1.10 (nous l'avons évoqué, les versions 2 restent proches des v1), qui exploite enfin la définition Book E complétée des APU et du Book VLE de Freescale.



Cette version 2.03 n'a pas de lien avec le PowerPC 2.02 d'IBM destiné au marché des serveurs. D'ailleurs, la version Power 2.04 annoncée en mai 2007 reste orientée serveurs, de même que le sera la future version 2.05.

Il faudra certainement attendre la version Power 2.06, qui apportera le partitioning et la virtualisation à l'embarqué, pour que l'on assiste réellement à la convergence entre les technologies Power d'IBM et de Freescale...