

O processador Cell

Flávio Augusto Wada de Oliveira Preto RA 032883

Luís Alfredo Harriss Maranesi RA 034231

Taísa Cristina Costa dos Santos RA 036065

Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Cidade Universitária “Zeferino Vaz” - Barão Geraldo - Campinas - SP - Brasil

{flavio.preto, luis.maranesi, taisha.santos}@ic.unicamp.br

Abstract. *Since the world's first processor, created in 1971, the processors evolved frighteningly. They were not just the components of the computers that most developed, but also the devices that had the fastest evolution through the History of Humanity. An example of this evolution is the Cell processor that, endowed with a new architecture, has a revolutionary nature. The objective of this paper is to introduce the Cell processor's main features in a concisely manner, describing and commenting information about how the Cell was created, its principal components and infrastructure, and finally, a comparison between its architecture and others contemporary microprocessors.*

Resumo. *Desde o primeiro processador do mundo, lançado em 1971, os processadores evoluíram assustadoramente. Eles não foram apenas os componentes dos computadores que mais evoluíram, mas sim os dispositivos que evoluíram mais rápido em toda a história da humanidade. Um exemplo dessa evolução é o processador Cell que, dotado de uma nova arquitetura, tem caráter revolucionário. O objetivo deste trabalho é apresentar, de forma sintética, as principais características do processador Cell, descrevendo e comentando as informações sobre como o Cell aconteceu, seus principais componentes e sua infraestrutura, e por fim, uma comparação de sua arquitetura com a de outros microprocessadores contemporâneos.*

1. Introdução

O processador Cell é uma arquitetura para computação distribuída de alta *performance*, desenvolvida por uma aliança entre a *Sony*, *Toshiba* e *IBM*, conhecida por *STI*. O projeto arquitetural e a primeira implementação foram realizados no *STI Design Center* durante um período de 4 anos iniciado em março de 2001, com um orçamento informado pela *IBM* de aproximadamente \$ 400 milhões.

Cell é uma abreviação para *Cell Broadband Engine Architecture*, comumente representada por CBEA ou Cell BE. Seu conceito foi originalmente pensado em 1999, quando Ken Kutaragi, da *Sony*, “o pai do PlayStation”, estava pensando sobre um computador que agisse como células num sistema biológico. Nesse contexto, o Cell foi criado de modo a combinar um núcleo *Power Architecture*, de propósitos gerais e de desempenho modesto, com elementos de coprocessamento ágil, o que acelera gigantesca-mente aplicações multimídia e de processamento vetorial, assim como outras formas de

computação dedicada. Além disso, como cada uma das três companhias produzem diferentes tipos de produtos, e estes possuem necessidades diferentes sobre a CPU, o projeto final do Cell incorpora características que satisfaz a todas essas necessidades.

A principal utilização comercial do Cell será no PlayStation 3, um *game console* da Sony, cujo lançamento está marcado para 17 novembro de 2006 nos Estados Unidos. A Mercury Computer Systems tem um servidor dual Cell, uma configuração *blade* dual Cell, um computador robusto e uma placa aceleradora PCI Express disponível em diferentes fases de produção. A Toshiba anunciou planos para incorporar o Cell em conjuntos de televisões de alta definição. Características exóticas, tais como o subsistema de memória XDR e a coerente interconexão EIB posicionam o Cell para aplicações futuras no espaço da supercomputação para explorar a habilidade do Cell em núcleos de ponto flutuante.

A arquitetura Cell é como nenhuma outra já existente em microprocessadores; ela é parecida, em projeto, com supercomputadores de multiprocessamento vetorial. Os desenvolvedores do Cell pegaram essa tecnologia e levaram, pela primeira vez, para a casa dos consumidores. O objetivo é produzir um sistema de baixo custo com um aumento massivo no desempenho computacional em relação aos sistemas existentes. Colocar tal arquitetura num único chip é um projeto árduo e complexo, que nenhum outro fabricante ousou tentar fazer até os dias de hoje.

2. A arquitetura Cell

A arquitetura Cell é compreendida por hardware e software Cells. Software Cells consiste de dados e programas (conhecidos por *jobs*); eles são enviados para o hardware Cells, onde são computados, e os resultados são então retornados.

O hardware do Cell foi especialmente projetado para fornecer dados suficientes para os elementos computacionais de modo a permitir alta performance. Essa é uma aproximação diferente da usual, que em geral tenta esconder as partes mais lentas do sistema. Todos os sistemas são retardados pelos componentes mais lentos. O Cell foi projetado para não ter componentes lentos.

Um hardware de Cell é formado de um grupo de *PowerPC Processor Elements* (PPEs), um grupo de *Synergistic Processor Elements* (SPEs), *Element Interconnect Bus* (EIB), *Direct Memory Access Controller* (DMAC), controladores de memória *Rambus XDR* e uma interface *Rambus FlexIO* (Input/Output). A figura 1 é um modelo representativo da arquitetura Cell.

A especificação final ainda não foi divulgada, mas pelo que foi estimado o Cell tem como características: capacidade de execução acima de 4 GHz; banda de memória: 25.6 GB/s; banda de I/O: 76.8 GB/s; 256 GFLOPS (precisão simples a 4 GHz); 256 GOPS (inteiro a 4 GHz); 25 GFLOPS (precisão dupla a 4 GHz); 235 mm²; e 235 milhões de transistores.

O consumo de energia foi estimado em 60 - 80W a 4 GHz para o protótipo, mas isso pode mudar na versão de produção. Além disso, a produção do chip é um processo complexo, e os chips que aparecem no final da linha de produção variam em capacidade e alguns possuem erros. Por esse motivo somado a razões econômicas e à dissipação de calor, o Cell do PS3 terá um clock de 3.2 GHz e terá apenas 7 SPEs. Já os eletrônicos de consumo terão 6 SPEs.

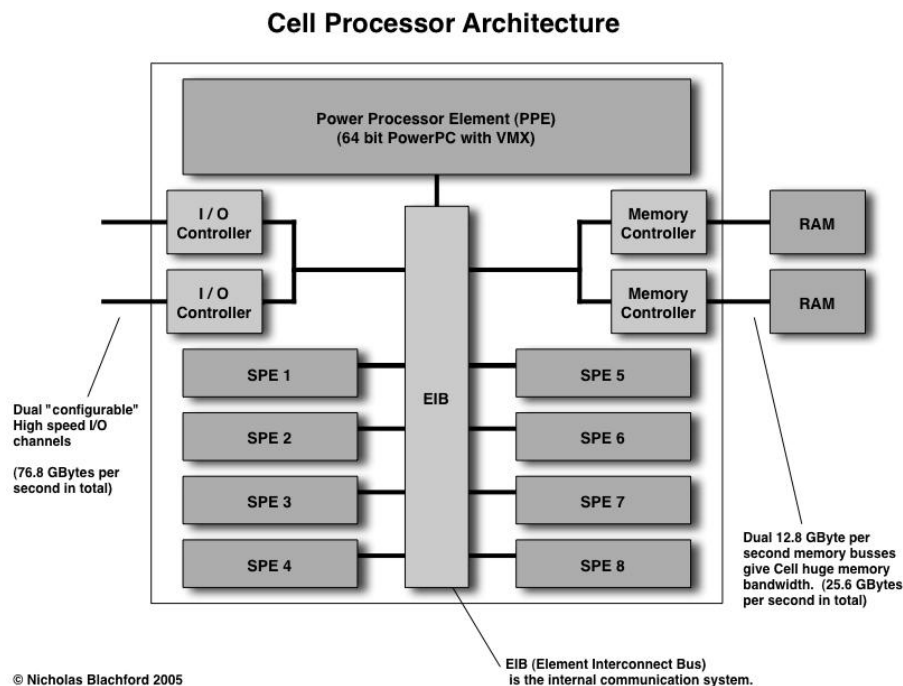


Figure 1. A arquitetura Cell

2.1. PowerPC Processor Elements (PPEs)

O *PowerPC Processor Element* (elemento de processamento PowerPC) ou PPE é constituída de uma unidade de processamento PowerPC de 64 bits, com caches L1 de 32KiB para instruções e 32KiB para dados e uma cache L2 de 512KiB. Apesar do PPE utilizar o conjunto de instruções do PowerPC, ele não é baseado em nenhum projeto do mercado atual, ou seja, não é baseado nos processadores existentes 970/65 ou POWER. No entanto o PPE é capaz de executar binários do POWER e PowerPC.

Um sistema em conformidade com a CBEA pode conter um ou mais PPEs, mas a primeira versão, tomada como base nesse trabalho e que será utilizada no PS3, contém apenas um. O PPE é um processador convencional que atribui tarefas aos SPEs. Ele executa o sistema operacional e a maioria das aplicações, mas as partes de computação intensiva são distribuídas aos SPEs.

A IBM também incluiu no PPE uma unidade VMX (Altivec), que é totalmente “pipelined” para pontos flutuantes de precisão dupla, e que, juntamente com os SPEs, que podem completar 2 operações de precisão dupla por ciclo de clock, gera uma taxa de 6,4 GFLOPS a 3,2GHz, ou 8 operações de precisão simples, produzindo 25,6 GFLOPS a 3,2Hz.

O PPE é um processador dual issue, dual threaded e in-order. Diferentemente dos processadores modernos, que necessitam de uma quantia maior de silício e energia para executar o máximo de instruções possíveis de uma vez, fora de ordem (out-of-order), os PPEs do Cell tem uma arquitetura de hardware no velho estilo RISC, ou seja, relativamente simples, e consomem bem menos energia que os dispositivos PowerPC, até mesmo a altas taxas de clock. Todavia, esse modelo possui *performance* irregular diante de aplicações carregadas de *branches*. Uma CPU tão simples precisa que o compilador

faça mais por ela quanto ao trabalho de sincronização, ou seja, um bom compilador é essencial. Entretanto, diante dessa desvantagem do Cell, um PowerPC de 4GHz parece um processador muito potente, até percebermos que os PPEs são meros controladores - a ação real está nos SPEs.

2.2. Synergistic Processor Elements (SPEs)

Como já foi dito anteriormente cada Cell possui, em seu projeto original, oito SPEs (*Synergistic Processing Elements*) que por sua vez são formados por um SPU (*Synergistic Processing Unit*) e um MFC (*Memory Flow Controller*) cada. É importante ressaltar que a arquitetura CBE (*Cell Broadband Engine Architecture*) pode suportar números diferentes de SPEs, não necessariamente oito como nessa primeira versão, e que esses elementos fazem um papel de “processadores dedicados”, análogo ao papel de um GPU (*Graphic Processing Unit*).

Os SPUs são basicamente processadores vetoriais (*Vector Processors*) de arquitetura RISC com instruções do tipo SIMD, capaz de efetuar cálculos com precisão simples ou dupla. Cada um deles possui 256 KB (apesar de ser possível o suporte a 4GB) de memória para dados locais (chamado em inglês de *local data memory area* ou *local store*), sendo essa memória visível também para o PPE e endereçável via software. Essa área de memória difere do cache de CPU convencionais, pois, além de não ser transparente para a camada de software, não possui nenhum tipo de estrutura que implemente “Branch Prediction”.

Os SPUs possuem bancos de registradores com 128 registradores de 128 bits cada. Dessa forma cada instrução pode operar com 16 inteiros de 8 bits, 8 inteiros de 16 bits, 4 inteiros de 32 bits, 4 valores de ponto flutuante de precisão simples ou ainda 2 de precisão dupla. Além disso possuem também capacidade de carregamento e armazenamento (*load-and-store*) à sua memória local e um canal de entrada e saída para controle do MFC.

Já o MFC serve de interface entre o sistema e o SPU, fornecendo os meios básicos para transferência de dados, proteção e sincronia entre a área de armazenamento principal e a área local ao SPU.

Como mencionado, uma das principais diferenças entre essa arquitetura e as demais que pode chamar a atenção é a falta de hardware que implemente o “Branch Prediction” e isso se deve ao compromisso de diminuir o *overhead* causado por essa técnica. Para compensar essa ausência foram projetadas instruções de previsão de *branch*, que são usadas pelo compilador, ou mesmo pelo próprio programador, para prevenir o hardware que haverá um salto, assim como seu endereço e seu alvo.

2.3. EIB e DMAC

O projeto inicial do Cell previa sua implementação baseada em um barramento de 1024 bits que ligasse todos seus elementos e um DMAC (*Data Memory Access Controller*) para controlar acessos de memória entre os elementos da arquitetura e implementar também um sistema de proteção à memória, que garantisse que programas não pudessem acessar espaços de memória indevidos ou de outros programas.

Com a evolução do projeto o DMAC foi desmembrado e, atualmente, só controla o acesso à memória. Sua antiga função de proteção à memória é exercida agora pelas

MMUs (*Memory Management Units*) que agora fazem parte do MFC (*Memory Flow Control*) de cada SPE. Além disso o barramento de 1024 bits foi substituído pelo chamado EIB (*Element Interconnect Bus*).

O EIB, ao contrário do projeto inicial, consiste em quatro anéis 16 bytes que funcionam com aproximadamente metade da velocidade total da CPU e pode suportar até 3 transferências simultâneas. Dessa forma a velocidade teórica de pico para o EIB é de 96 bytes por ciclo e a velocidade real prevista pela *IBM* é de quase dois terços dessa marca.

Apesar do processamento sequencial (do inglês, *stream process*) não depender de velocidade nem da banda do barramento, um dos motivos que garantem o alto desempenho do cell são essas características, providas pelo EIB e o DMAC.

2.4. Memória

Um dos grandes desafios dos processadores atuais é superar o problema da velocidade da memória, que é muito mais lenta que o processador. A arquitetura CELL sofre ainda mais do problema, já que ela tem vários núcleos de processamento que precisam fazer acesso a memória. Os projetistas então precisaram tomar algumas medidas para amenizar este problema.

O PPE trabalha em modo virtual de forma que a aplicação enxerga um espaço de endereçamento de até 64 bits atribuído pelo sistema operacional. Cada aplicação no PPE tem sua própria tabela de páginas que mapeia a memória virtual para a memória real.

Além disso, cada SPE possui uma área local de armazenamento, de 256KiB, que funciona como uma memória para o SPE. Desta forma a aplicação e os dados que estão em execução no SPE são primariamente armazenados na área local, que é endereçada pela aplicação como se fosse uma memória convencional. Qualquer acesso do SPE à memória real do sistema deve ser feita através de operações DMA. A grande vantagem obtida com esse esquema é que o número de acessos a memória real cai drasticamente, além do fato que o armazenamento local é feito da mesma tecnologia que o processador e portanto extremamente rápida.

As áreas de armazenamento locais dos SPE podem ser mapeadas pela memória virtual do sistema. Desta forma, pode-se escrever e ler dados da área local de um SPE pelo PPE ou através de DMA de outros dispositivos como processadores de vídeo. Embora reduza a sobrecarga do sistema liberando uma forma de acesso direto à área de armazenamento do SPE, esta operação é muito custosa e deve ser evitada, pois pode levar a uma queda drástica de performance.

Este sistema aliado a necessidade de grande velocidade na memória fez com que a *IBM* licenciasse as tecnologias da *Rambus* conhecidas por *Yellowstone* e *Redwood*. Desta forma foi possível construir tanto um sistema de I/O muito veloz como memórias mais rápidas dos que estão atualmente disponíveis no mercado.

3. O Cell e outros microprocessadores

Um dos maiores debates que ocorrem sobre o processador Cell é a respeito dele ser um processador de propósitos gerais (*general purpose processor*) ou um processador especializado (*specialized processor*). Apesar de ser otimizado para tarefas de alta

computação, o Cell é um processador de propósitos gerais, visto que os PPEs são processadores convencionais e agem como tal. A grande diferença está nos SPEs, pois esses foram projetados para acelerar tipos específicos códigos e serão notavelmente melhor em algumas áreas que outras. Contudo, até mesmo os SPE são *general purpose processors*. O Cell tem como essência a escolha por executar tudo em velocidade moderada pela habilidade de executar certos tipos de código em alta velocidade.

3.1. Ordem de execução

Tanto os PPEs quanto os SPEs são *in-order processors*. Uma justificativa para isso é o fato de as CPUs *out-of-orders* (OOO) serem projetadas especificamente para reduzir a taxa de instruções executadas por ciclo de clock. Entretanto, em códigos com muitas dependências, a prática OOO torna-se praticamente inútil. Ou seja, processadores OOO, além de consumir muita energia, aumentam o desempenho em casos específicos, enquanto que o aumento de clock, por exemplo, aumenta a performance de tudo.

3.2. “Branch Prediction”

Assim como o OOO, a utilidade de um *branch predictor* depende do tipo de código que está sendo executado. Por esse motivo, os PPEs possuem hardware de *branch prediction*, porém esse é bem limitado, diferentemente dos processadores convencionais de *desktops*.

3.3. Dual Threading

Quanto a característica de *dual threading*, apenas os PPEs a possuem, sendo os SPEs *stream processors*. Ambas as *threads* rodam alternadamente, mas quando uma entra em espera, a outra toma o comando. Duas instruções podem ser executadas por ciclo de clock (4 são buscadas a cada ciclo). O hardware de *multithreading* é semelhante ao *Hyper-threading* da Intel, já presente no POWER5. Tal característica recompensa parcialmente a falta da prática de OOO e de um *branch predictor* mais robusto, já que a segunda *thread* pode utilizar todos os recursos de execução enquanto a primeira está em espera.

3.4. Desempenho × Energia

Ao longo dos anos verificou-se que, conforme eram criadas novas técnicas e tecnologias para melhorar a *performance* das CPU, como por exemplo, o aumento do número de bits dos processadores, a criação da cache, da MMU, do *pipeline*, da execução superescalar, da execução OOO e etc, o aumento que essas tecnologias proporcionaram era cada vez menor. Infelizmente, o oposto acontece com o consumo de energia, uma vez que cada melhoria parece ficar cada vez mais complexa.

No Cell, a escolha foi entre um projeto OOO com amplos *branch predictors* ou um projeto mais simples, menor e com maior velocidade de clock. Uma vez que as características mais complexas como OOO proporcionam um aumento de desempenho proporcionalmente maior que seu aumento de consumo de energia, o Cell não as possui. Além disso, o uso de um conjunto de registradores de tamanho razoável exige um bom compilador que sincronize as instruções de modo que as dependências causem pouco impacto, gerando o efeito de um hardware OOO.

Portanto, o resultado de tal simplicidade do Cell terá impacto em algumas áreas, mas a combinação de um clock de alta velocidade, uma nova tecnologia de compiladores e uma seleção de algoritmos eficazes poderá reduzir esses problemas enormemente.

4. Conclusão

Apesar de, no mundo dos PCs, a evolução ser preferida a revolução, o Cell é uma revolução. É um microprocessador de arquitetura completamente nova, e promete um enorme aumento de desempenho sobre os sistemas atuais. Além disso, sendo produzido em alta escala, haja vista sua utilidade não só para PCs mas também para o mundo dos sistemas embarcados, o Cell será um processador barato.

Por ser uma arquitetura nova, o Cell pode soar estranho para muitos, acostumados com projetos para *desktops* convencionais. Além disso, não podemos ignorar o quanto do potencial do Cell pode ser alcançado e como é difícil extraí-lo. Portanto, os programadores cujo intuito seja utilizar adequadamente o processador, deverão encarar novos problemas e elaborar caminhos para solucioná-los.

Muitas pessoas não gostam de mudanças. Para elas, o Cell representa um desafio. Para os outros, uma oportunidade, incluindo os fabricantes de outras CPUs, que terão a chance de dar sua resposta.

5. References

References

- [1] Sony Computer Entertainment Inc., *Cell Broadband Engine™ Architecture*, Version 1.0, August 8, 2005.
- [2] Computing Frontiers 2006, *Chip Multiprocessing and the Cell Broadband Engin*, May 2006.
- [3] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy, *Introduction to the Cell multiprocessor*, Volume 49, Number 4/5, 2005.