

Verification of the cell broadband engine™ processor

K. Shimizu, S. Gupta, T. Koyama, T. Omizo, J. Abdulhafiz, L. McConville, T. Swanson
ACM IEEE Design Automation Conference 2006, pp 338-343.

Autor: Bruno Teles - ra 042348

Este resumo descreve como a equipe principal de verificação formada por profissionais da aliança STI produziu no primeiro silício um microprocessador Cell BE funcional o bastante para iniciar um sistema operacional e executar aplicações. Vale ressaltar que o Cell BE possui uma série de características nunca antes vistas em outros projetos deste tipo. Muitos dos desafios de verificação que foram encontrados e as estratégias para abordar estes desafios são destacados neste resumo: um enfoque será dado em técnicas específicas e tópicos que tiveram um significativo impacto tais como verificação hierárquica, verificação aleatória e verificação arquitetural *trace-based*.

Antes de apresentar as técnicas de verificação anteriormente citadas, é feita uma breve apresentação dos principais componentes desta arquitetura heterogênea. Seus componentes primários são: oito SPEs; o *PowerPC Processor Element* (PPE), cuja função primária é o gerenciamento e a alocação de tarefas para a SPEs; o *Element Interconnect Bus* (EIB), um barramento com coerência de memória conectando os núcleos dos processadores entre si e com os controladores de interface para a memória principal e outros dispositivos; um controlador de interface de memória (*Memory Interface Controller* - MIC); e um controlador de interface de barramento (*Bus Interface Controller* - BIC).

É importante ressaltar que vários dos novos aspectos deste chip são voltados para a SPE e suas interações com o resto do chip. A SPE é requisitada para tarefas que requerem alta densidade computacional e foi feita como uma ponte entre processadores de propósito geral e hardwares dedicados. Cada SPE inclui um *Synergistic Processor Unit* (SPU); o *Memory Flow Controller* (MFC), que é um mecanismo de transferência de dados; e o *Local Storage* (LS), que tem 256K de memória embutidos no chip.

Uma razão chave para o sucesso da equipe pode ser atribuído ao excelente uso da verificação hierárquica. Primeiro, a equipe atingiu o objetivo de encontrar a maioria dos bugs com os ambientes menores e mais rápidos. Isso leva a uma alta taxa de bugs encontrados por tempo de cada engenheiro, por unidade de tempo de simulação, e libera a parte da carga que testes de ambientes maiores teriam que suportar para completar sua verdadeira missão. Esta métrica de eficiência é especialmente crítica para um chip maior com múltiplos núcleos como o Cell BE, e em geral, à medida que os projetos ficam maiores e ter múltiplos núcleos se torna uma característica comum a arquiteturas usuais, isto pode se tornar um ponto chave para a indústria. Além disso, é crucial compreender que ambientes de verificação de diferentes tamanhos, apresentam uma classe diferente de bugs. À medida que a complexidade do chip cresce e o período para fazer um projeto encurta, esta abordagem paralela hierárquica se torna cada vez mais necessária.

Uma outra característica da abordagem adotada pela equipe do Cell foi a ênfase em geração de estímulo randômico. Um ambiente de verificação Cell típico inclui um estímulo randômico na entrada e um verificador que observa e checka as ações continuamente na saída. A utilização de verificação aleatória pela equipe do Cell não foi apenas abrangendo todo o chip, mas também se deu em vários níveis de granularidade. Aleatoriedade é usada no nível de sinal; é usada no nível de transação; e é também usada no nível de instrução. Este uso abrangente de estímulo aleatório e geração de cenários em vários níveis diferentes foi um fator chave na produção de uma lógica clara, iniciada do zero.

Estes casos de testes para verificação no nível do chip são gerados pela utilização do gerador de testes da IBM X-Gen. O X-Gen é um gerador de testes aleatório para transações do sistema. Ele cria a priori seqüências de instruções do processador baseada em restrições especificadas pelo usuário. Uma importante característica funcional deste gerador de testes é que ele é capaz de direcionar o teste com autonomia em vez apenas usar parâmetros aleatórios não-especificados. Isto permite um alto grau de automação e produtividade do usuário. Além disso, ele é crucial na geração de testes capciosos que atingem os casos-limite em vez de desperdiçar preciosos ciclos de simulação em testes ineficazes e repetitivos. Dessa maneira, o X-Gen foi um fator chave na verificação da inovadora arquitetura multiprocessador PPE/SPE dentro do tempo previsto. Enfim, ele expôs vários bugs, entre eles problemas de coerência de memória.

Por último, a verificação *trace-based* foi feita com o intuito de garantir que o projeto do arquiteto para esta nova arquitetura foi implementado corretamente. Esta abordagem visa verificar o funcionamento do chip como um todo. A ferramenta utilizada para tal tarefa foi a *Coherency monitor Late* (CML). Devido à sua natureza global, verificadores CML são eficazes em encontrar bugs que não são detectados por verificadores de unidade, seja em ambientes de unidade ou no ambiente do chip. Com este foco na arquitetura global, verificadores CML são capazes de detectar bugs que podem passar despercebidos por verificadores de unidade. CML foi uma ferramenta indispensável para garantir que a arquitetura Cell BE fosse de fato estabelecida.

Este resumo do artigo é um breve estudo das técnicas que foram especialmente eficazes ou até mesmo críticas para o sucesso do primeiro chip Cell BE. A intenção do artigo não foi fazer uma análise completa da metodologia de verificação ou das etapas de projeto de verificação utilizado. Na verdade, pode-se considerar que se uma técnica não foi citada no artigo, esta teve apenas um impacto limitado. Em particular, técnicas baseadas em simulação foram muito mais eficazes do que técnicas formais (estas últimas foram responsáveis por achar apenas 0.2% dos bugs encontrados).