MO401 – Trabalho 1 – Resumo de Artigo

Multiprocessor System-on-Chip Designs with Active Memory Processors for Higher Memory Efficiency. Junhee Yoo, Sungjoo Yoo, Kiyoung Choi. Proceedings of the 46th Annual Design Automation Conference, July 26-31, 2009, San Francisco, California.

Autora: Flávia de Oliveira Santos. **RA: 100594**

A latência de acesso a memória e operações relacionadas são freqüentemente o gargalo de desempenho em aplicações paralelas. No artigo em questão, é apresentado o conceito de *active memory operations* (operações de memória ativa) que é uma transação de rede *on-chip* que opera baseada no micro código provido pelo engenheiro de *software*. A abordagem proposta tem por objetivo tentar reduzir o número de transações na rede *on-chip*, ao invés de reduzir a latência em si. Em resumo, o método proposto combina muitas operações de leitura/escrita simples em uma operação de alto nível chamada de operação de memória ativa.

O artigo apresenta também a implementação de um processador chamado AMP (*Active Memory Processor* - Processador de Memória Ativa) que está localizado próximo a memória e executa as operações de memória ativa. O AMP é um típico processador VLIW *in-order-issue*. Seu conjunto de instruções VLIW é de 64 bits e pode decodificar, em cada ciclo, um acesso a memória, uma computação de dados, uma computação de endereço, um desvio e uma operação de leitura/escrita na rede.

A arquitetura utilizada nos estudos de caso para validação da abordagem é composta de 36 blocos com 32 *cores* e 4 memórias. Cada *core* é um elemento de processamento (PE – *Processing Element*) e contém um processador RISC e numa memória local. Cada memória contém um processador de memória ativa.

Foram realizados estudos de caso utilizando três aplicações reais: *Fast Fourier Transform* (FFT), codificador JPEG paralelizado e indexação de textos para mineração de dados. Em cada estudo de caso foi verificado o desempenho para um caso base que utiliza apenas os PEs e um caso melhorado que utiliza o AMP.

No exemplo do FFT, quando o número de processadores é 32, o tempo de execução do caso melhorado diminui 25,6% comparado ao caso base, o que traduz para um aumento de desempenho de 34,3%. Isso é devido à diminuição de transações de acesso de dados que caíram 30,6% reduzindo assim o tráfego na rede. Houve, entretanto, um aumento no número de transações para carregar a *cache* de instruções dos PEs de 37% devido ao código adicional para utilização do AMP.

No exemplo do codificador JPEG, para o caso base, houve uma saturação na melhora do desempenho por volta de 12 processadores devido à presença de um *lock* no *buffer* de armazenamento. O caso melhorado não mostra essa saturação, visto que o tempo de execução no AMP é bem menor comparado ao caso base. Como resultado, houve uma melhoria de 198,7% no desempenho do caso melhorado para o caso de 32 processadores.

Já no exemplo da indexação de textos, o caso base requer no mínimo 10 acessos à memória para inserir um valor na tabela da aplicação enquanto que o caso melhorado faz o mesmo em apenas uma transação. Em termos de número de palavras processadas por segundo, o caso base tem seu desempenho saturado por volta de 1.000K palavras por segundo enquanto que o caso melhorado é capaz de processar até 7.000K palavras por segundo quando todos os 32 processadores são utilizados. Isso resultou em uma melhoria de 618%.

O artigo em questão apresentou o conceito de operações de memória ativa e um processador de memória ativa como implementação. Resultados experimentais mostraram que a abordagem proposta pode melhorar o desempenho em cerca de 34,3% a 618% para aplicações reais com o custo de um esforço adicional de *design* e um aumento de área moderado na interface de rede do bloco de memória.