

FERRAMENTA DE PROFILING PARA PROCESSADORES

SCHNEIDER, R. C.¹, NEVES, B. S.¹

¹Universidade Federal do Pampa (UNIPAMPA) – Bagé – RS – Brasil

RESUMO

O aumento da complexidade dos processadores e de suas etapas internas dificultam cada vez mais o entendimento do perfil das aplicações e a identificação de potenciais gargalos no hardware. Possibilitar ao projetista o acesso ao comportamento do hardware pode auxiliar na identificação das causas de perda no desempenho, subutilização ou sobrecarga de elementos. Neste sentido, evidencia-se a indisponibilidade de ferramentas open-source que possibilitem a obtenção de métricas para a análise de performance dos processadores. Neste trabalho, descrevemos uma unidade em hardware que permite coletar informações do comportamento das etapas de pipeline e arquitetura de memória cache de um processador, implementado sobre um FPGA, durante a execução de uma aplicação. Esta ferramenta, composta por registradores com conteúdo acessível através de endereços de memória, contabiliza os eventos através de estímulos originados diretamente a partir do processador utilizado. A consistência e funcionalidade do dispositivo foi preliminarmente verificada utilizando o software de simulação Mentor ModelSim. Após isto, buscou-se a validação final do hardware de medição através de experimentos com uso do processador Gaisler LEON3 e sintetizado para um dispositivo reconfigurável da Altera. Os resultados obtidos sugerem um elevado grau de confiabilidade das medições realizadas com a nova ferramenta, sustentando sua alta eficácia como instrumento para auxílio aos desenvolvedores na tarefa de obter maior conhecimento sobre o perfil de seus projetos.

Palavras-chave: Análise; Processadores; Desempenho.

1 INTRODUÇÃO

Em aplicações como sistemas embarcados é de interesse dos projetistas extrair a máxima eficiência do sistema em desenvolvimento, seja ela em termos de tempo de execução, performance computacional, consumo energético, etc. Neste sentido, utilizar ferramentas que permitam traçar o perfil da aplicação se mostram valiosas para dois propósitos distintos: identificar gargalos e guiar possíveis otimizações (DEAN *et al.*, 1997) ou auxiliar no particionamento do *hardware* versus *software* num contexto de co-projeto (ALDHAM *et al.*, 2011).

Neste aspecto, pode-se optar pela utilização de uma abordagem em *software*, mais conhecida como SBP (*Software Based Profiling*). Apesar do emprego deste tipo de ferramenta se mostrar mais popular e menos custoso, Tong e Khalid (2008) afirmam que as técnicas de amostragem usando este método podem levar a resultados de perfilamento imprecisos.

Por outro lado, é possível optar numa abordagem que se baseia no uso de recursos dedicados em *hardware* para realizar a avaliação de desempenho da aplicação. Apesar da alta complexidade desta técnica, ela apresenta maior precisão e velocidade quando comparado às ferramentas SBP. Adicionalmente, Aldham et al. (2011) citam que o perfilamento em *hardware* é geralmente uma necessidade em projetos de sistemas embarcados onde as restrições de tempo real requerem que a aplicação opere em conformidade com prazos rígidos de execução, o que torna o perfilamento através de *software* impraticável.

Recentemente, estudos como (CHEN *et al.*, 2009) e (KYUNG *et al.*, 2007) demonstram a implementação de unidades em *hardware* com a proposta de construir um perfil comportamental da aplicação durante a execução sobre o processador. Entretanto, no tocante a este ponto, observa-se a indisponibilidade deste tipo de ferramenta sob licença *open-source*.

Assim sendo, o objetivo deste trabalho é implementar uma ferramenta de livre acesso baseada em *hardware* para captura de dados de desempenho para a execução de dado programa, tendo como focos principais a análise do *pipeline* e da arquitetura de memória cache do processador alvo. Esta pesquisa está organizada da seguinte forma: a seção 2 apresenta a metodologia que foi utilizada para implementação do hardware proposto, a seção 3 apresenta uma discussão sobre os resultados obtidos e a seção 4 apresenta as conclusões do estudo.

2 METODOLOGIA (MATERIAIS E MÉTODOS)

A escolha do processador Gaisler LEON3, como plataforma para a implementação da ferramenta desenvolvida, baseia-se no livre acesso ao seu código HDL (*Hardware Description Language*), disponibilizado sob licença GNU GPL. Além disso, o processador LEON3 utiliza o protocolo aberto ARM AMBA nos barramentos de alta velocidade e baixa velocidade que facilita a implementação por ser um protocolo amplamente difundido e ter suas especificações disponibilizadas de forma gratuita.

A Figura 1, mostra a estrutura do *hardware* de medição proposto já conectado ao processador LEON3. Este esquema, inspirado no trabalho de Chen et al. (2009), possibilita o acesso dos registradores de eventos através de endereços de memória previamente especificados. Aliado a isto, a forma como o hardware de medição se conecta ao processador usando o barramento de alta velocidade AMBA AHB (*Advanced High-Performance Bus*), permite reduzir o atraso para emissão de dos sinais de controle, enviado pelo processador ao dispositivo de medição, para início e término do processo de análise.

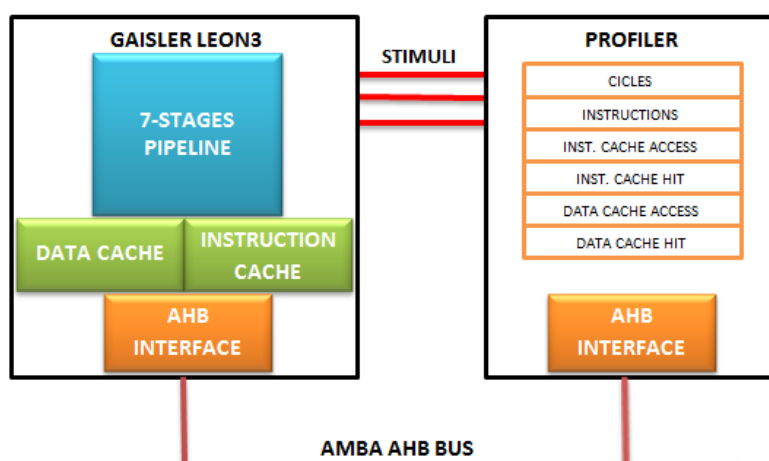


Figura 1. Esquemático do hardware de profiling e suas conexões com o processador Gaisler LEON3.

Como métricas a serem coletadas pela ferramenta proposta foram escolhidas para análise do *pipeline*: número de ciclos e de instruções. Estas métricas, além de fornecer dados relativos ao tempo de execução, permitem a derivação de outras

métricas como CPI (Ciclos Por Instrução). De maneira semelhante, foram selecionadas como métricas para a arquitetura cache a contagem de referências realizadas às memórias caches do processador e o número de acertos vinculado a estas referências. Desta forma, pode-se obter tanto a taxa de acerto quanto a taxa de falha para cada cache.

Sob estas especificações, a validação da ferramenta de medição foi realizada, usando resultados comparativos produzidos por duas outras ferramentas com propósitos similares. Primeiramente foi utilizada uma versão de teste (*trial*) do software simulador TSIM também fornecido pela Gaisler. Este simulador busca reproduzir o funcionamento em nível de instruções do LEON3 e retorna ao final de sua execução a quantidade de instruções executadas por este processador. Apesar de o TSIM fornecer outras informações como número de ciclos, estes resultados não foram considerados nesta primeira validação preliminar da nova ferramenta proposta, haja vista que não foram identificados, até o presente, documentos fornecidos pela Gaisler que descrevam detalhes da arquitetura de memória usada pelo TSIM.

Complementarmente, utilizou-se também para a validação da nova ferramenta o módulo L3STAT, disponibilizado gratuitamente pela Gaisler junto com a biblioteca GRLIB (Gaisler Research Library). Apesar deste módulo L3STAT coletar um número significativo de informações sobre o processador, ele não fornece dados que possibilitem obter direta ou indiretamente as taxas de acertos e falhas das memórias cache do LEON3. Além disso, este módulo está conectado ao barramento AMBA APB (ARM *Peripheral Bus*), sendo este um barramento de comunicação desenvolvido para elementos periféricos que operem a baixa velocidade. Complementarmente, cada registrador de evento precisa ser habilitado, desabilitado e configurado individualmente, o que pode levar a uma maior incidência de erros nos resultados finais produzidos pelo L3STAT dependendo da quantidade de registradores a serem utilizados e do *jitter* dos barramentos AHB e APB.

3 RESULTADOS E DISCUSSÃO

Para obtenção dos resultados foi utilizada a única configuração de arquitetura de memória cache disponível na ferramenta TSIM, na qual cada cache possui tamanho de 4KB, divididos em linhas de 16 Bytes, e utiliza mapeamento direto como política de substituição. Deste modo, para a obtenção dos resultados com a L3STAT e com a ferramenta proposta, foi sintetizado o processador LEON3 com estas configurações para um dispositivo reconfigurável da Altera. Como programas testes foram utilizados dois diferentes algoritmos de ordenação. As Tabelas 1 e 2 apresentam os resultados obtidos com as três ferramentas.

Métrica	Ferramenta proposta	TSIM	L3STAT
Ciclos	32622232	33487304	-
Nº de Instruções	16883499	17361132	16883203
Nº de Acessos Cache de instruções	17593244	-	-
Nº de Falhas Cache de instruções	73448	-	34958
Nº de Acessos Cache de dados	4009955	-	-
Nº de Falhas Cache de dados	185714	-	183768

Tabela 1 – Resultados obtidos para o programa teste 1.

Observa-se que tanto os dados obtidos na ferramenta TSIM como na ferramenta L3STAT não permitem uma análise mais abrangente do comportamento do processador durante as execuções dos programas teste porque não é possível obter informações como taxas de acerto ou de falhas em cada memória cache, ou no caso específico da L3STAT não temos a medição precisa do tempo de execução dos programas teste. Dentre as informações que permitem comparação, verificamos que o número de ciclos, instruções e falhas no acesso à cache de dados apresentaram valores muito aproximados nas três ferramentas. Apesar do número de falhas no acesso à cache de instruções produzido pela ferramenta proposta não ser próximo ao do resultado produzido pelo módulo L3STAT, observou-se que a diferença segue um padrão linear de cerca de 2 vezes.

Métrica	Ferramenta proposta	TSIM	L3STAT
Ciclos	37338446	38763721	-
Nº de Instruções	20720840	21195587	20711890
Nº de Acessos Cache de instruções	20752452	-	-
Nº de Falhas Cache de instruções	88	-	40
Nº de Acessos Cache de dados	3782398	-	-
Nº de Falhas Cache de dados	276589	-	264275

Tabela 1 – Resultados obtidos para o programa teste 2.

Além disso, dentre as duas ferramentas pré-existentes à este estudo, somente a ferramenta TSIM possibilita obter uma métrica adicional, neste caso o CPI. Neste sentido, os resultados obtidos apontaram 1,932 ciclos/instrução com o uso da ferramenta proposta e 1,928 ciclos/instrução no TSIM, para o programa teste 1. Do mesmo modo, para o programa teste 2, foi obtido 1,80 ciclos/instrução na nova ferramenta e 1,82 ciclos/instrução no TSIM.

Considera-se importante salientar que o novo hardware dedicado proposto permite calcular o desempenho das memórias cache do processador, o que por falta de dados não é possível obter nas demais ferramentas testadas. Verifica-se, por exemplo, que para o programa teste 1 a cache de dados apresenta uma taxa de falha de cerca 4,63% dos acessos enquanto que no programa teste 2 este valor chega a 7,31%.

4 CONCLUSÃO

Neste trabalho foi apresentada a proposta de um novo *hardware* para avaliação de desempenho em processadores descritos em HDL. O dispositivo proposto foi implementado junto ao processador Gaisler LEON3, o qual possui seu código de descrição de *hardware* disponível sob licença aberta.

Com base na comparação com outras duas ferramentas, o simulador TSIM e o módulo L3STAT, verificou-se a consistência dos resultados obtidos com esta nova ferramenta. Observou-se desta maneira que a ferramenta proposta permite a obtenção de métricas que não estão disponíveis nas demais ferramentas que se utilizou para comparação como, por exemplo, maiores informações sobre o comportamento das memórias caches de dados e instruções.

Por fim, o *hardware* de *profiling* proposto nesta pesquisa foi disponibilizado na comunidade open-cores sob licença LGPL, podendo assim ser utilizado como fonte

de consulta ou base para novas ferramentas. Como trabalhos futuros, espera-se estender a instrumentação para coleta de métricas para unidades externas ao processador, como os barramentos de comunicação.

5 REFERÊNCIAS

ALDHAM, M. *et al.* Low-cost hardware profiling of run-time and energy in FPGA embedded processors. 2011, [S.l: s.n.], 2011. p. 61–68.

CHEN, P. H. *et al.* Multiprocessor system-on-chip profiling architecture: Design and implementation. 2009, [S.l: s.n.], 2009. p. 519–526.

DEAN, J. *et al.* ProfileMe: hardware support for instruction-level profiling on out-of-order processors. *Proceedings of 30th Annual International Symposium on Microarchitecture*, 1997.

KYUNG, H. *et al.* Performance monitor unit design for an AXI-based multi-core SoC platform. 2007, [S.l: s.n.], 2007. p. 1565–1572.

TONG, J. G.; KHALID, M. A. S. *Profiling tools for FPGA-based embedded systems: Survey and quantitative comparison. Journal of Computers.* [S.l: s.n.], 2008