

ANÁLISE DO ACESSO A MEMÓRIA EM PROCESSADOR EMBARCADO COM VISTAS A UTILIZAÇÃO DE MEMÓRIAS HÍBRIDAS

Lisandro Luiz da Silva¹, Lisane Brisolara²; Júlio C. B. Mattos²

¹Programa de Pós-Graduação em Computação

²Centro de Desenvolvimento Tecnológico - UFPel

{lisandro, lisane, julius} @inf.ufpel.edu.br

1. INTRODUÇÃO

Atualmente, as aplicações dos sistemas embarcados tornam-se cada vez mais complexas, mesmo com o aumento da capacidade de processamento dos sistemas digitais modernos, contudo, o tempo para comercialização (*time-to-market*) torna-se cada vez menor.

De um modo geral a descrição dos requisitos funcionais não é suficiente para o projeto de um sistema embarcado, devendo ser considerados também requisitos não-funcionais, tais como desempenho, custo, consumo de energia, tamanho físico e peso. Em um sistema de computador moderno o gargalo dominante na obtenção de alto desempenho e eficiência energética é a distância tecnológica entre o desempenho do processador e a memória tradicional (Hennessy, 2011). Esta distância torna-se mais significativa em sistemas embarcados. Atualmente, sistema de memória é um dos principais fatores de desempenho e consumo energético, especialmente nos sistemas embarcados que utilizam bateria. Existem tecnologias emergentes para memórias em sistemas embarcados. Comparando uma memória tradicional SRAM, com as memórias emergentes STT-RAM e PCRAM, as memórias emergentes proporcionam um leakage menor e uma maior densidade. Além disso, STT-RAM pode ser melhor que a PCRAM, pois a STT-RAM possui uma menor latência de acesso e potência dinâmica, enquanto PCRAM possui maior densidade. Segundo Sun, 2008 e Wu, 2009 a memória STT-RAM é mais adequada para memórias de último de nível, enquanto que a PCRAM é promissora como uma alternativa para DRAM na memória principal (Lee, 2009). Novos modelos de STT-RAM foram desenvolvidos para diminuir os problemas envolvidos com as operações de escrita.

Este trabalho tem como objetivo investigar o consumo energético e o desempenho nos acessos a memória em processadores embarcados com vistas a utilização de um sistema híbrido de memória consistindo com memórias voláteis (VM, Volatile Memories) e memórias não voláteis (NVM, Non-Volatile Memories). Como estudo de caso é realizada uma comparação através da substituição de uma memória scratchpad SRAM por uma scratchpad STT-RAM, com a finalidade de reduzir o consumo energético e melhorar o desempenho. Uma scratchpad é um array de memória com a parte de lógica de decodificação (Banakar, 2002). A ideia por trás de uma scratchpad é manter objetos de memória mapeados pelo compilador no seu último estágio. A SPM ocupa uma parte do espaço de endereçamento e o restante é ocupado pela memória principal. É utilizado SPM no trabalho ao invés de uma memória cache, pois, as SPM's possuem a vantagem de não existir a necessidade da checagem da disponibilidade do dado ou instrução como as memórias caches, eliminando a questão dos rótulos (tags) e comparadores existentes nas caches. Isso contribui significativamente na redução da área e no consumo de energético.

2. METODOLOGIA

Foi utilizado um conjunto de benchmarks do MiBench, os benchmarks são divididos em seis categorias. Dentre todos os benchmarks do MiBench, foram escolhidos alguns aleatoriamente para este trabalho, pois existe, uma longa

demora na obtenção dos resultados (tempo de compilação). A preferência da escolha foi de pelo menos um benchmark por categoria. Os benchmarks escolhidos foram: basicmath, bitcount, crc32, dijkstra, FFT, ispell, jpeg, mad, patricia, qsort, rijndael, sha, stringsearch, susan, tiff2bw, tiff2rgba e typeset.

O simulador Simics suporta diversos tipos de arquiteturas de processadores, a arquitetura analisada neste trabalho foi a arquitetura ARM SA1110, responsável por modelar um sistema monoprocessoado com processador ARMv5 (Intel Strong ARM), ele oferece suporte nativo ao Linux, kernel 2.4.12, este modelo de processador ARMv5 é genérico.

O primeiro passo foi compilar todos os benchmarks selecionados para gerar binários através de um cross-compiler, para isso utilizamos o SIMICS. Para o segundo passo foi utilizado um script para a geração automática do trace de um determinado benchmark. O próximo passo foi a verificação de quantas instruções foram executadas no trace de instrução, qual o tipo de instrução, e o número de vezes que cada instrução foi executada. Sendo assim foi criado um script para a realização desta tarefa. Nesta etapa também foi realizada a verificação dos endereços, quais foram mais acessados, e a contagem desses acessos no trace de instruções e dados para cada benchmark selecionado no trabalho. Após é realizada a classificação das instruções de cada benchmark, a fim de saber o número exato de instruções que acessam a memória.

Nesta etapa também foi realizada a separação dos endereços mais acessados, dos traces de dados e de instruções de todos os benchmarks analisados. Ainda no trace de dados foi separado os endereços de leitura e os de escrita. Sabendo que cada endereço possui 4 bytes realizamos a comparação de memórias de 4 bytes até 8 Kbytes que possuem os endereços mais acessados em ambos os traces analisados, e a comparação de memórias de 8 bytes até 8 Kbytes dos endereços mais acessados de leitura e escrita do trace de dados. A ideia por trás desta avaliação é identificar a quantidade de memória necessária para os acessos mais realizados, tanto para instruções como dados.

Sendo estes os procedimentos utilizados para a obtenção dos resultados de cada benchmark para arquitetura ARM. Para os cálculos de desempenho e consumo energético foi utilizada a ferramenta CACTI na versão 6.0 para as memórias voláteis e a ferramenta NVSIM para as memórias não voláteis.

3. RESULTADOS E DISCUSSÃO

A Figura 1 representa a porcentagem média dos endereços mais acessados dos traces de instrução, juntamente com a maior e menor porcentagem de cada tamanho de memória. Observa-se que na média para um programa de 8K bytes é responsável por 64,41% dos acessos a memória.

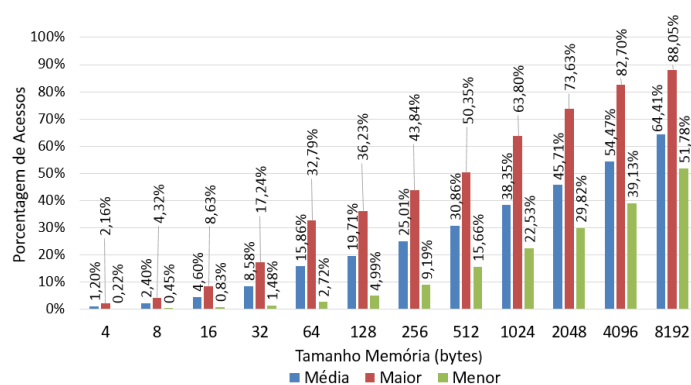
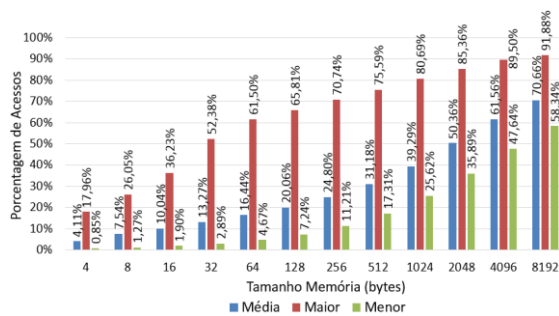
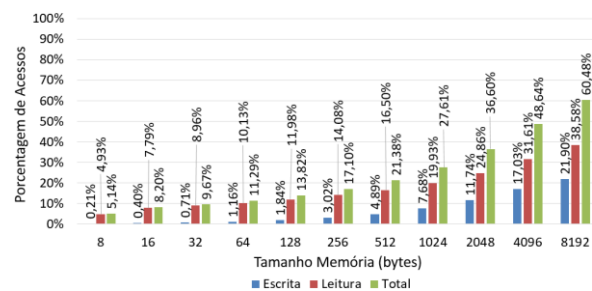


Figura 1 – Trace de Instruções

A Figura 2 (a) apresenta a porcentagem média dos endereços mais acessados dos trazes de dados (tanto leituras e escritas), juntamente com a maior e menor porcentagem de cada tamanho de memória. Pode observar que na média existe uma grande localidade temporal dos dados, pois 8K bytes dos dados são responsáveis por 70,66% dos acessos à memória. A Figura 2 (b) apresenta os acessos a memória de dados, classificando em leitura e escrita, em relação ao total de acessos da quantidade de dados. Observa-se que leitura, como já esperado, é responsável pela maior parte dos acessos a memória de dados.



(a)



(b)

Figura 2 – Trace de Dados

Para um estudo de caso foi realizada a comparação de desempenho e energia entre uma memória scratchpad SRAM com uma memória scratchpad STT-RAM. Na Figura 3, percebe-se que, o único caso que se obteve melhoria no parâmetro latência de leitura, foi o de 64 KB, onde se obteve 12,54% de melhoria. No parâmetro, latência de escrita, percebe-se que, não se obteve nenhum ganho, devido ao fato que a latência para uma SRAM é inferior, quando comparada com uma STT-RAM do mesmo tamanho. Ainda na Figura 3, percebe-se que, o consumo energético para a operação de leitura, obteve ganhos em todos os tamanhos comparados. Esses ganhos estão entre 65,19% e 92,31%. Para uma operação de escrita, percebe-se pela Figura 3, que se obteve ganhos para memórias a partir de 16K bytes. Esses ganhos estão entre 34,10% e 83,91%. A Tabela 1 apresenta a porcentagem dos resultados obtidos na comparação entre NVM e VM. A memória SRAM foi normalizada para a obtenção desses resultados.

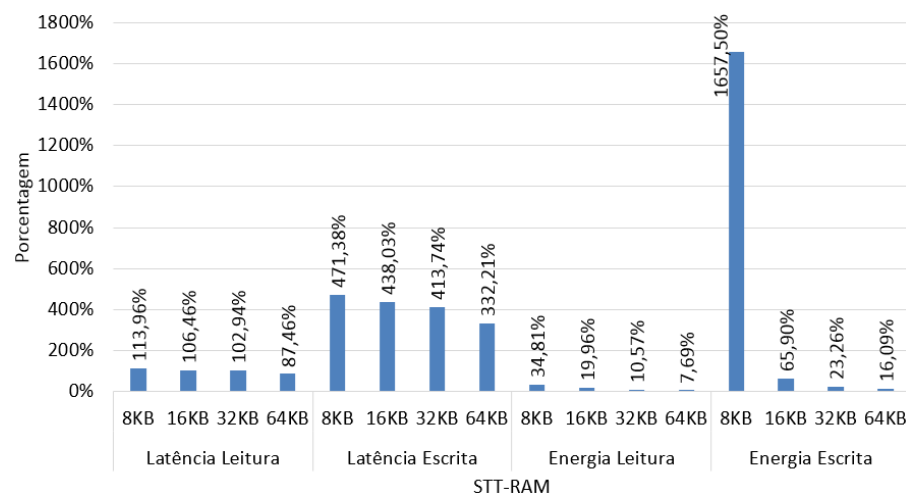


Figura 3 – STT-RAM vs SRAM (normalizada em 100%)

TABELA 1 . RESULTADOS ENCONTRADOS

32nm	STT-RAM			
	8 Bytes	16 Bytes	32 Bytes	64 Bytes
Latência Leitura	+13,96%	+6,46%	+2,94%	-12,54%
Latência Escrita	+371,38%	+338,03%	+313,74%	+232,21%
Energia Leitura	-65,19%	-80,04%	-89,43%	-92,31%
Energia Escrita	+1557,50%	-34,10%	-76,74%	-83,91%

4. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho realizou uma análise do impacto dos acessos à memória em *softwares* embarcados executados em processadores ARM. Foram estudadas diversas técnicas de otimizações de memória na literatura para os sistemas embarcados e atualmente o trabalho encontra-se na fase de modelar uma arquitetura de memória híbrida. Será utilizado duas memórias, uma memória volátil e outra memória não volátil. A memória volátil possui o tempo de acesso para uma operação de escrita inferior ao da memória não volátil, entretanto a memória não volátil possui o tempo de acesso inferior para uma operação de leitura, ou seja, os endereços que possuírem uma frequência de acessos maior para as operações de escrita serão armazenados na memória volátil, enquanto os endereços que possuírem uma maior frequência de acessos para as operações de leitura serão armazenados na memória não volátil. Assim, conclui-se que a memória é um dos principais fatores que influenciam o desempenho e o gasto energético em sistemas embarcados, graças a isso a mesma possui um grande potencial para as futuras pesquisas relacionadas a esse fato, como demonstrado nesse trabalho.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- J. L. Hennessy; D. A. Patterson. **Computer Architecture – A quantitative approach**. 5ª. Ed. San Francisco: Morgan Kaufmann, 2011.
- G. Sun; X. Dong; Y. Xie; J. Li & Yiran Chen; 2008. **A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs**. in Proc. HPCA, pp. 239 - 249.
- X. Wu, et al., 2009. **Hybrid Cache Memory Architecture with Disparate Technologies**. in Proc. BAIT, pp. 34 - 45.
- B. Lee; E. Ipek; O. Mutlu & D. Burger. 2009; **Architecting Phase Change Memory as a Scalable DRAM Alternative**. s.l., s.n., pp. 2 - 13.
- R. Banakar, et. al., 2002. **Scratchpad Memory: a design alternative for cache on-chip memory in embedded systems**. IEEE International Symposium on Hardware/Software Codesign, pp. 73-78.