# Inversão de Bit uma Alternativa na Redução do Desgaste das Memórias PCM

**Felipe L. Teixeira**[1*], **Maurício L. Pilla**[1], **André R. Du Bois**[1]

{flteixeira, pilla, dubois}@inf.ufpel.edu.br

[1]Universidade Federal de Pelotas (UFPEL)
Computação - CDTec
Caixa Postal 354 · CEP 96001-970 · Pelotas, RS - Brasil

***Resumo.*** *Memórias PCM e Memórias Transacionais são novas alternativas em suas áreas. Neste artigo, é analisado uma técnica de inversão de bits com a intensão de diminuir o número de bits alterados na memória PCM. Em geral os resultados mostraram que os* benchmarks *obtiveram uma redução de até 30% nos bits alterados.*

## 1. Introdução

A *Phase Change Memory* (PCM) [Lee et al. 2010] surgiu como uma alternativa para o consumo proibitivo de energia em memórias tradicionais, pois após o armazenamento de um bit não há consumo de energia para mantê-lo. O problema das memórias PCM está em suas escritas, que são lentas. Além das escritas serem lentas, elas desgastam o material devido à mudança de fase que fazem para armazenar os dados, diminuindo sua vida útil.

Memórias Transacionais ou *Software Transactional Memories* (STM) [Herlihy et al. 1993] foram desenvolvidas para diminuir as dificuldades e limitações de métodos de sicronização baseados em *locks*. O conceito por trás de STM é baseado em transações de banco de dados, assim tendo como uma vantagem sobre as sincronizações baseadas em *locks* a simplicidade de escrita de código.

Neste trabalho, será analisado uma técnica de inversão de bits, uma variação de [Stan and Burleson 1995] implementada em *software*, com a intensão de diminuir o número de bits alterados na memória PCM. Para isso utilizamos *benchmarks* de STM, visto que as STMs fazem parte de trabalhos futuros. Para tanto, a biblioteca de STM *TinySTM*, a ferramenta *PinTools* e o *benchmark* STAMP foram usados para gerar um ambiente de avaliação.

O artigo é dividido da seguinte forma: a Seção 2 apresenta a metodologia utilizada neste trabalho. Após, a Seção 3 mostra os resultados obtidos. Por fim, a Seção 4 discute as conclusões.

## 2. Metodologia

Para o desenvolvimento deste trabalho foi implementado uma instrumentação utilizando a ferramenta PinTools. Esta instrumentação faz uma avaliação, a cada escrita na memória, se é melhor inverter os bits ou não. Com isso pretende-se diminuir a mudanças de bits na memória PCM, fazendo com que diminua seu desgaste.

*Bolsista FAPERGS

Os testes foram feitos em uma máquina com arquitetura NUMA, AMD *Opteron* com 64 *cores*, 128GB de memória e Ubuntu 12.04.2 LTS de 64 bits. Os *benchmarks* foram executados 10 vezes com 1 *thread*, com a implementação feita e sem a implementação, isto para que fosse possível fazer uma comparação.

## 3. Resultados

A Figura 1 apresenta a porcentagem de redução de bits alterados, com a implementação, em relação à execução sem a implementação. Como podemos ver todos os *benchmarks* tiveram uma redução no número de bits alterados. Alguns mostraram uma porcentagem baixa, que foi o caso dos *benchmarks Genome* e *Intruder* que apresentaram uma redução de menos de 1% dos bits alterados. Isso se deve ao fato do algoritmo implementado ser um algoritmo guloso, assim ele busca a melhor opção no momento, mas essa pode não ser a melhor opção no decorrer da execução. Um *benchmark* que obteve uma redução considerável foi o *Labyrinth*. Ele obteve uma redução de 30% dos bits alterados.

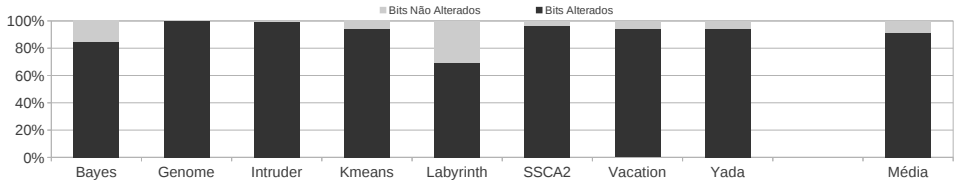


**Figura 1. Porcentagem de Redução dos Bits Alterados.**

Em geral os *benchmarks* obtiveram uma redução dos bits alterados de 8% em média. Com isso podemos observar que a inversão de bits é uma alternativa para a redução dos bits alterados. Isso visto que os *benchmarks* têm um alto número de bits alterados sendo o menor número de bits alterados de 986 milhões, apresentado pelo *benchmark Genome*.

## 4. Conclusões

Neste artigo foi abordado uma técnica de inversão de bits para diminuir o número de bits alterados em uma memória PCM. Os resultados mostraram que para alguns *benchmarks* a inversão de bits não mostrou uma redução, dos bits alterados, representativa. Isso devido ao algoritmo utilizado ser um algoritmo guloso. Mas para outros *benchmarks* a inversão de bits mostrou-se uma boa alternativa para reduzir os bits alterados em uma memória PCM. Como trabalhos futuros pretendemos avaliar o tempo inserido devido à inversão dos bits e também implementar a inversão de bits em uma biblioteca de STM.

## Referências

Herlihy, M., Eliot, J., and Moss, B. (1993). Transactional memory: Architectural support for lock-free data structures. In *in Proceedings of the 20th Annual International Symposium on Computer Architecture*, pages 289–300.

Lee, B. C., Zhou, P. Z. P., Yang, J. Y. J., Zhang, Y. Z. Y., Zhao, B. Z. B., Ipek, E., Mutlu, O., and Burger, D. (2010). Phase-change technology and the future of main memory. *IEEE Micro*, 30(1):131–141.

Stan, M. and Burleson, W. (1995). Bus-invert coding for low-power i/o. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 3(1):49–58.