

Aluno: Anselmo Castelo Branco Ferreira RA: 023169
Artigo: A Novel Tag Access Scheme for Low Power L2 Cache
Autores: Hyunsun Park et al
Conferência: DATE 2011

No artigo é tratado o problema do consumo de energia em *caches* L2 em comparações de *tag*. É proposto um esquema de redução de energia composto de 3 passos: primeiramente, na chegada de um acesso à cada entrada da *cache*, um **filtro de bloom melhorado de tags parciais irá melhorar a previsão de um *cache miss***, reduzindo desta forma as comparações de *tag*. Esse filtro verifica quantos endereços há na entrada da *cache* onde o endereço seria mapeado por *hash*. Caso não haja endereço algum há um *per-way cache miss*. Caso haja apenas um endereço na entrada da *cache* (segundo os autores isso ocorre em 60% dos casos) a *tag* parcial do mesmo é comparada com a de entrada. Se as *tags* parciais não combinarem, há novamente um *per-way cache miss*.

Caso o filtro de *bloom* no passo anterior não previna o *cache miss*, continua-se a análise nas **linhas de *cache* quentes e frias**. Linhas de *cache* quentes ocorrem no período em que há mais acessos à *cache*, sendo o contrário para as linhas frias. A maioria dos *cache hits* ocorrem em linhas de *cache* quentes. Portanto, no segundo passo **busca-se a informação nas linhas de *cache* quentes**. Caso a informação não esteja lá, vamos ao terceiro passo que é **buscar a informação nas linhas de *cache* frias**. Em caso de *cache hits* em linhas de *cache* quentes, economiza-se comparações de *tag* com as linhas de *cache* frias. Caso a informação não esteja nem nas linhas de *cache* quentes nem nas frias temos um *cache miss*.

Para definir se uma linha de *cache* é quente ou fria equipam-se elas com um contador de vivacidade. Cada vez que uma linha de *cache* é referenciada, seu contador de vivacidade é inicializado em um valor *TH*. Esse contador é decrementado a cada *N* ($N = 16$ nos experimentos) referências à *cache*. Se o contador zera, sua correspondente linha de *cache* é considerada fria. O número de comparações de *tag* é em função de *TH*. Se ele puder ser setado ao valor que resulte em um número mínimo de comparações de *tag*, pode-se reduzir o consumo de energia. O relacionamento entre *TH* e o número de comparações de *tag* muda em diferentes programas e diferentes fases em um programa. Para resolver esse problema os autores determinam o *TH* ideal ao rastrear a dinâmica da mudança de comportamento de programas.

O consumo de energia no segundo e terceiro passo pode ser ainda mais reduzido ao **dividir a comparação de tags em 2 micro-passos**. Primeiro compara-se uma *tag* parcial do endereço de entrada com as *tags* parciais da *cache*. No caso de erro nas linhas de *cache* quentes procedemos as comparações parciais nas linhas de *cache* frias. No caso de *hit* parcial nas linhas de *cache* quentes ou frias, a parte remanescente da *tag* é comparada para dar o resultado completo da comparação. No caso de erro nas linhas de *cache* quentes procedemos as comparações parciais nas linhas de *cache* frias. No caso de erro nas linhas de *cache* frias, temos um *cache miss*. Esses 2 micro-passos de comparação de *tag* parcial e completa pode ser realizada em um ou 2 ciclos de clock, dependendo da frequência do *clock* considerada.

Nos experimentos foram executados 10 programas do SPEC2000 e SPEC2006 no simulador comercial da Tensilica LX2. As técnicas escolhidas para serem comparadas com a proposta foram os acessos seriais (nativo da *cache* L2) e outros métodos existentes de redução de comparações de *tag* (método de filtro de *bloom*, predição de caminho e decadência de *cache*). As três técnicas anteriores reduziram o consumo médio de energia na faixa de 11.98% a 16.55% se comparados ao acesso de *tag* seriais. Já a técnica proposta reduziu o consumo médio de energia em 22.45% se comparada ao mesmo acesso serial.