

# Avaliação em FPGA do mecanismo de memorização de traços dinâmicos

Elias Luiz da Silva Júnior

CEFET-MG

- Demanda crescente por computação.
- Em parte atendida pela Lei de Moore.
- Avanços tecnológicos tem permitido:
  - Aumento exponencial no número de transistores em um circuito integrado;
  - Redução exponencial no custo de circuito integrado;
  - Redução no consumo elétrico de sistemas de computação.

- Isso se manteve verdadeiro por cerca de 50 anos.
- Limites físicos próximos de serem atingidos (MACK, 2011).
- Necessárias alternativas para criação de melhorias a nível de arquitetura.

- Uma alternativa bastante utilizada é explorar o paralelismo tanto a nível de instrução quanto a nível de tarefa.
- Porém não é uma solução completa:
  - Incapaz de paralelizar instruções com dependências;
  - Depende do programador e do compilador para ser melhor explorado (HENNESSY; PATTERSON, 2011).

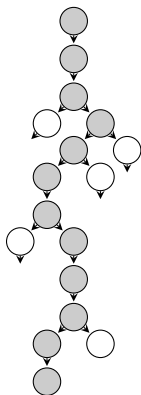
- Técnica que produz ganhos de desempenho em várias aplicações.
- Lida muito bem com instruções que possuem dependências.
- Permite a execução de múltiplas instruções de uma vez (COSTA, 2001).

- Instruções redundantes: não possuem efeitos colaterais e já foram executadas anteriormente com os mesmos valores de entrada.
- Traço: sequência de instruções redundantes.

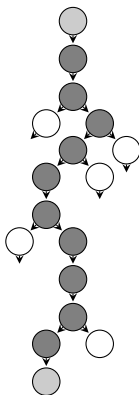
- A memorização dinâmica de traços (DTM) armazena traços e instruções redundantes.
- Em execuções posteriores identifica oportunidades de reuso e recupera os resultados anteriores.

# Memorização dinâmica de traços

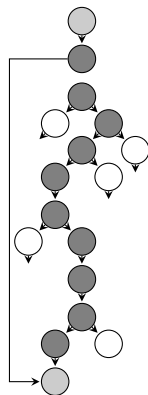
**Figura:** Exemplo do processo de DTM: (a) Instruções executadas; (b) Instruções redundantes; (c) Fluxo de controle quando ocorre reuso.



(a)



(b)



(c)

Fonte: elaborada pelo autor



- Os valores utilizados em um traço produzidos externamente a ele compõe o chamado “contexto de entrada”.
- Os valores produzidos em um traço e disponíveis para uso externo a ele compõe o chamado “contexto de saída”.
- Ao reusar um traço, é feito um desvio para a próxima instrução a ser executada.

Figura: Exemplo de pseudocódigo.

```
 $c \leftarrow c + a$   
 $x \leftarrow c + b$   
se  $x \leq 5$  então  
     $y \leftarrow x * 2$   
senão  
     $x \leftarrow x + 1$   
fim se
```

Fonte: elaborada pelo autor

Nesse código, os valores  $a, b, c$  compõe o contexto de entrada. Os valores  $c, x$  e, dependendo do fluxo de execução  $y$ , compõe o contexto de saída.

# Memorização dinâmica de traços

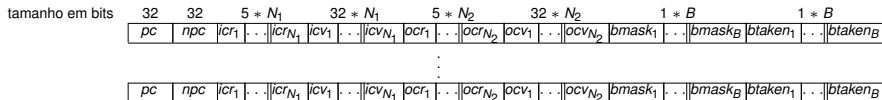
**Figura:** Representação da tabela *Memo\_Table\_G*, utilizada para recuperação de instruções individuais.

tamanho em bits	32	32	32	32	1	1	1
	<i>pc</i>	<i>sv1</i>	<i>sv2</i>	<i>res/targ</i>	<i>jmp</i>	<i>brc</i>	<i>btaken</i>
	⋮						
	<i>pc</i>	<i>sv1</i>	<i>sv2</i>	<i>res/targ</i>	<i>jmp</i>	<i>brc</i>	<i>btaken</i>

Fonte: elaborada pelo autor

# Memorização dinâmica de traços

**Figura:** Representação da tabela *Memo\_Table\_T*, utilizada para recuperação de traços.



Fonte: elaborada pelo autor

Diversos trabalhos, entre eles Costa (2001), Pilla et al. (2003), Silva (2006) e Laurino (2007), testaram a DTM em diferentes formas e todas produziram ganhos de desempenho para diversas aplicações.

Além da mudança no desempenho do sistema, incluir a DTM em um processador também pode alterar:

- área de chip;
- potência consumida;
- latência de ciclo;

- Trabalhos anteriores que analisavam a DTM utilizaram apenas modelos teóricos ou simulações computacionais.
- Impossível ou extremamente difícil de prever tais alterações dessa forma.

## Proposta do trabalho:

- Implementar o mecanismo de DTM em uma arquitetura de processador;
- Sintetizar o circuito e gravar em uma placa de FPGA;
- Executar os programas de *benchmark* do SPEC CPU 2006;
- Comparar medições realizadas no processador original com as do processador alterado.



- A arquitetura de processador utilizada é o LEON3, da Cobham Gaisler.
- O LEON3 implementa a ISA SPARC V8.
- A placa de FPGA utilizada é a Altera Cyclone II EP2C35F672C6.

Objetivos do trabalho:

- Definir os *trade-offs* de implementar a DTM;
- Saber para quais aplicações práticas a técnica é viável.

COSTA, A. T. da. *Explorando dinamicamente o reuso de traces em nível de arquitetura de processador*. 2001. Tese (Doutorado) —

COPPE/UFRJ, Rio de Janeiro, 2001.

HENNESSY, J. L.; PATTERSON, D. A. *Computer architecture: a quantitative approach*. [S.l.]: Elsevier, 2011.

LAURINO, L. S. *Reuso especulativo de traços com instruções de acesso à memória*. 2007. Dissertação (Mestrado), 2007.

MACK, C. A. Fifty years of moore's law. *IEEE Transactions on semiconductor manufacturing*, IEEE, v. 24, n. 2, p. 202–207, 2011.

PILLA, M. L. et al. The limits of speculative trace reuse on deeply pipelined processors. In: IEEE. *Computer Architecture and High Performance Computing, 2003. Proceedings. 15th Symposium on*. [S.l.], 2003. p. 36–44.

SILVA, B. R. *Memorização e reuso dinâmico de traços em uma arquitetura de processador Java*. 2006. Dissertação (Mestrado) — COPPE/UFRJ, Rio de Janeiro, 2006.