



**COPPE/UFRJ**

**SENSIBILIDADE DO MECANISMO DE REUSO DE TRAÇOS AOS  
SUBCONJUNTOS DE INSTRUÇÕES**

Sheila de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas da Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências em Engenharia de Sistemas da Computação.

Orientador: Felipe Maia Galvão França

Rio de Janeiro

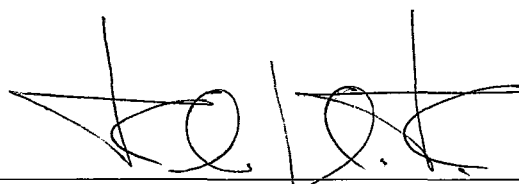
Março de 2009

SENSIBILIDADE DO MECANISMO DE REUSO DE TRAÇOS AOS  
SUBCONJUNTOS DE INSTRUÇÕES

Sheila de Oliveira

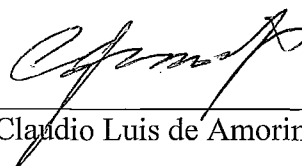
DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS DA COMPUTAÇÃO.

Aprovada por:



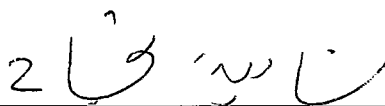
---

Prof. Felipe Maia Galvão França, Ph.D.



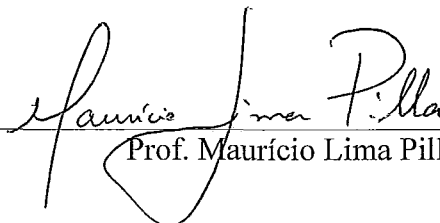
---

Prof. Claudio Luis de Amorim, Ph.D.



---

Profª Nadia Nedjah, Ph. D.



---

Prof. Mauricio Lima Pilla, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2009

Oliveira, Sheila de

Sensibilidade do Mecanismo de Reuso de Traços aos subconjuntos de instruções/Sheila de Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2009.

XII, 82 p.: il.; 29,7cm.

Orientador: Felipe Maia Galvão França

Dissertação (mestrado) – UFRJ / COPPE / Programa de Engenharia de Sistemas da Computação, 2009.

Referências Bibliográficas: p. 77-79.

1. Reuso dinâmico de traços 2. Arquitetura de Processador I. França, Felipe Maia Galvão II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas da Computação. III. Título

## **Agradecimentos**

A Deus, me dar a vida e fazer dela um eterno motivo de agradecimentos por todas as bênçãos que sempre recebi.

Aos meus pais, Rute e Timoteu, por serem meu alicerce em todos esses anos e terem a maior parcela de responsabilidade por eu ser a pessoa que sou e estar onde estou hoje.

Ao meu irmão Ozomatli, por toda a dedicação e torcida pra eu conquistasse mais essa etapa.

Ao meu amado Anderson, por todo o apoio, credibilidade, paciência, estímulo, compreensão, dedicação, amor, parceria e cumplicidade, nos vários momentos em que precisamos abdicar de muitas coisas para que este trabalho pudesse ser concluído.

Ao Programa de Engenharia de Sistemas e Computação da Universidade Federal do Rio de Janeiro, por ter aceitado o meu ingresso neste e permitido a conclusão de mais essa fase importantíssima de minha formação acadêmica.

Aos meus verdadeiros amigos, que me viram aos poucos me ausentar das ocasiões em que sempre estive presente, compreendendo que isto fazia parte de uma fase necessária e muito desejada da minha vida.

Ao Julio Cesar da Costa, que durante toda a minha permanência no programa teve a sensibilidade de compreender quão difícil é a tarefa de se obter um título de Mestre, cumprindo uma jornada de trabalho em paralelo e, em função disso, fazendo-me concessões que foram fundamentais para a conclusão deste trabalho de dissertação.

Ao colega Andrey Coppieters, pela boa vontade em dividir seus conhecimentos nos meus primeiros passos decisivos para o desenvolvimento deste trabalho.

Aos colegas da COPPE/Sistemas, mais especificamente do LAM, Bruno França, Leandro Marzullo e Lawrence Bandeira, pela atenção dispensada.

Ao meu orientador de fato, Felipe Maia Galvão França, por ter aceitado a parceria neste trabalho de dissertação e ter me tranquilizado e garantido que tudo daria certo, em todos os momentos em que eu demonstrei insegurança, acreditando nas possibilidades que este trabalho traria.

Em especial, agradeço ao Professor Maurício Lima Pilla, meu “orientador de fé”, por todas as horas de dedicação, auxílio, infindáveis explicações. Por dividir comigo seu conhecimento, que certamente me deu segurança e condições de continuar até o fim.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## SENSIBILIDADE DO MECANISMO DE REUSO DE TRAÇOS AOS SUBCONJUNTOS DE INSTRUÇÕES

Sheila de Oliveira

Março/2009

Orientador: Felipe Maia Galvão França

Programa: Engenharia de Sistemas da Computação

Este trabalho apresenta um estudo sobre a sensibilidade do mecanismo do reuso de traços aos subconjuntos característicos de instruções. Fragmentar o domínio de instruções reusáveis em subconjuntos torna possível perceber a parcela de contribuição que cada um destes possui dentro do contexto do reuso de traços. A partir da criação de um Índice de Eficiência (IE), que é resultante da aceleração obtida em cada subconjunto sobre o percentual de instruções executadas neste subconjunto, foi possível avaliar de forma mais precisa quão importante cada subconjunto é dentro do mecanismo de reuso. Subconjuntos de instruções lógicas e aritméticas, instruções de desvio e instruções de acesso à memória foram criados e simulações foram realizadas com estes. Concluiu-se que instruções lógicas e aritméticas têm grande relevância dentro do mecanismo de reuso, no momento em que atingem 92,8% da aceleração do reuso total, com apenas 42,8% do percentual de reuso. Instruções de desvios apresentam o melhor IE, uma vez que atingem 90,3% da aceleração com um reuso de 14,5%. Instruções de Acesso à Memória não participam do reuso de traços, apenas de instruções isoladas, mas mesmo assim conseguem manter mais do que o dobro da eficiência do Reuso Total, usando o índice criado como referencial comparativo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## SENSITIVITY OF THE TRACE REUSE MECHANISM TO INSTRUCTION SUBSETS

Sheila de Oliveira

March/2009

Advisor: Felipe Maia Galvão França

Department: Engenharia de Sistemas da Computação

This work presents a study about the sensitivity of the trace reuse mechanism to instruction subsets. Fragmenting the reusable instructions domain in subsets makes it possible to understand the contribution of each subset within the trace reuse context. The creation of an Efficiency Index (EI), which is obtained from a division of each subset speedup by the percentage of instructions executed in that subset, allows for a more precise evaluation of the importance of each subset within the reuse mechanism. Subsets of logical and arithmetic instructions, branch instructions and memory access instructions were created and simulations were performed with them. The conclusions are: logical and arithmetic instructions are very important in reuse mechanism, since they alone produces 92.8% of total reuse speedup, with only 42.8% of reusable rate; branch instructions have the best EI, since they alone produces 90.3% of total reuse speedup, with 14.5% of reusable rate. Memory access instructions do not participate in trace reuse, only single instructions, but still produce more than twice the total reuse efficiency, using the index created as a comparative benchmark.

## Sumário

1.	Introdução .....	1
2.	Trabalhos Relacionados .....	4
2.1.	Reuso de Instruções Dinâmicas .....	6
2.1.1.	Esquemas para o Reuso de Instruções .....	7
•	O Esquema Sv .....	8
•	O Esquema Sn .....	9
•	O Esquema Sn + d .....	10
2.1.2.	Micro arquitetura com um Reuse Buffer .....	11
2.2.	Result Cache .....	4
2.2.1.	A natureza redundante da computação .....	4
•	Memoização .....	5
•	Result Caching .....	5
2.3.	Remoção Dinâmica de Computações Redundantes .....	13
2.3.1.	O Redundant Computation Buffer .....	14
2.3.2.	Result Cache e Reuse Buffer Modificados .....	17
2.4.	Reuso de Blocos Básicos .....	18
2.4.1.	Implementação .....	19
2.5.	Reuso de Sub-Blocos .....	21
2.5.1.	Cortes dos blocos básicos .....	22
2.6.	Reuso em nível de traços .....	23
2.6.1.	O potencial de desempenho do reuso em nível de instruções e traços .....	26
2.7.	Reuso de Load .....	27
2.7.1.	A estrutura de ligação .....	28
2.7.2.	A estrutura de exploração .....	28
2.7.3.	O hardware .....	29
3.	DTM e RST .....	32
3.1.	DTM: Memorização dinâmica de traços .....	32
3.1.1.	A microarquitetura com DTM .....	36
•	Inclusão de instruções na Memo_Table_G .....	38
•	Identificação de instruções redundantes .....	38
•	Atualização do contexto do traço .....	38

•	Seleção das instruções e traços redundantes candidatos.....	38
•	Identificação de uma instrução ou traço redundante .....	39
•	Reuso de uma instrução ou traço .....	39
3.1.2.	Implementação do Mecanismo DTM .....	39
3.2.	RST: Reuso de traços através da especulação de valores.....	40
3.2.1.	O RST e sua integração com o DTM.....	41
3.2.2.	Reuso e construção de traços no RST.....	42
3.2.3.	Teste de predição incorreta e recuperação .....	43
3.2.4.	A arquitetura RST: Diferenças entre o pipeline RST e o pipeline DTM.....	44
•	Os estágios do pipeline RST.....	46
4.	Base Experimental, Resultados e Avaliações .....	50
4.1.	Base Experimental .....	50
4.1.1.	Ambiente de Simulação .....	50
4.1.2.	Benchmarks .....	51
4.1.3.	Parâmetros Arquiteturais do Processador Simulado.....	52
4.1.4.	Parâmetros Arquiteturais do Mecanismo DTM e RST.....	53
4.2.	Metodologia Experimental .....	54
4.2.1.	Motivação .....	54
4.3.	Resultados .....	56
4.3.1.	Medidas.....	56
4.3.2.	Aceleração .....	57
4.3.3.	Taxa de Reuso.....	60
4.3.4.	Índice de Eficiência .....	63
5.	Conclusões .....	73
6.	Referências Bibliográficas.....	77
7.	ANEXO A.....	80



## Lista de Figuras

Figura 2. 1 – Reuso Buffer Genérico.....	7
Figura 2. 2 - Estrutura da entrada RB para o Esquema Sv .....	8
Figura 2. 3 - Estrutura da entrada RB para o Esquema Sn .....	9
Figura 2. 4 – Estrutura da entrada RB para o Esquema Sn+d .....	11
Figura 2. 5 - Microarquitetura genérica com um Reuse Buffer.....	12
Figura 2. 6 - Exemplos de Quase-Invariantes (a) e Subexpressões Quase Comuns (b). 13	
Figura 2. 7 - Estrutura do Redundant Computation Buffer .....	14
Figura 2. 8 - Estrutura de uma entrada do Block History Buffer .....	19
Figura 2. 9 - Modelo do processador que avalia o potencial de desempenho do reuso de blocos.....	20
Figura 2. 10 - Estrutura de campos de uma entrada RTM.....	24
Figura 2. 11 – Reuso de traços durante as fases do pipeline .....	25
Figura 2. 12 - Estrutura da Load Table.....	30
Figura 2. 13 – Estrutura da Recent Store Queue .....	31
Figura 3. 1 - Estrutura das entradas da Memo_Table_G.....	33
Figura 3. 2 – Estruturas para o armazenamento de traços no DTM .....	34
Figura 3. 3 – Estrutura das entradas da Memo_Table_T.....	35
Figura 3. 4 – Microarquitetura do DTM.....	37
Figura 3. 5 – Pipeline DTM.....	45
Figura 3. 6 – Pipeline RST .....	46
Figura 3. 7 – Exemplo de um registro da Tabela de Recuperação(RT) .....	48
Figura 3. 8 – Integração entre os estágios do pipeline RST .....	49
Figura 4. 1 – Aceleração Add Sub.....	58
Figura 4. 2 – Aceleração Desvios.....	59
Figura 4. 3 – Aceleração Load Store .....	60
Figura 4. 4 – Add e Sub: Taxa de Reuso .....	61
Figura 4. 5 – Desvios: Taxa de Reuso .....	62
Figura 4. 6 – Load e Store: Taxa de Reuso .....	63
Figura 4. 7 – Índice de Eficiência Add e Sub.....	64
Figura 4. 8 – Índice de Eficiência Desvios.....	65
Figura 4. 9 – Índice de Eficiência Load e Store .....	66

Figura 4. 10 – Média Harmônica da Aceleração .....	67
Figura 4. 11 – Entrada da Memo_Table_G sem reuso de desvios .....	69
Figura 4. 12 – Entrada da Memo_Table_T sem reuso de desvios.....	70
Figura 4. 13 - Média Harmônica das Taxas de Reuso.....	70
Figura 4. 14 – Média Harmônica dos Índices de Eficiência.....	72

## Lista de Tabelas

Tabela 4. 1 - Benchmarks usados nos experimentos .....	51
Tabela 4. 2 – Distribuição das instruções executadas por tipo de instrução .....	52
Tabela 4. 3 – Configurações arquiteturais do processador utilizado nas simulações.....	52
Tabela 4. 4 – Parâmetros dos mecanismos DTM e RST .....	53
Tabela 4. 5 – Domínio de instruções reusáveis .....	55
Tabela 4. 6 – Memo_Table_G original e sem reuso de desvios.....	68
Tabela 4. 7 – Memo_Table_T original e sem reuso de desvios .....	69

## Lista de Abreviaturas

BHB	Block History Buffer
DTM	Dynamic Trace Memoization
IE	Índice de Eficiência
LLT	Load Linking Table
LT	Load Table
RB	Reuse Buffer
RCB	Redundant Computation Buffer
RLQ	Recent Load Queue
RSQ	Recent Store Queue
RST	Reuse Through Speculation on Traces
RST	Register Source Table
RT	Recovery Table
RTM	Reuse Trace Memory























































































































































































