



Acta Scientiarum. Technology

ISSN: 1806-2563

eduem@uem.br

Universidade Estadual de Maringá

Brasil

Cantos Faveri, Rodrigo Richard; de Lara Gonçalves, Ronaldo Augusto; Martini, João Angelo

Interface de apoio para projeto, simulação e avaliação de arquiteturas superescalares

Acta Scientiarum. Technology, vol. 31, núm. 2, 2009, pp. 133-140

Universidade Estadual de Maringá

Maringá, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=303226524001>

- ▶ Como citar este artigo
- ▶ Número completo
- ▶ Mais artigos
- ▶ Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal  
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

# Interface de apoio para projeto, simulação e avaliação de arquiteturas superescalares

Rodrigo Richard Cantos Faveri\*, Ronaldo Augusto de Lara Gonçalves e João Angelo Martini

Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá, Paraná, Brasil. \*Autor para correspondência.  
E-mail: rofaveri@din.uem.br

**RESUMO.** Este artigo apresenta uma interface gráfica para a ferramenta *SimpleScalar*, que auxilia o desenvolvimento de projeto, simulação e avaliação de desempenho de arquiteturas superescalares. A interface permite ao projetista configurar diversos parâmetros arquiteturais e disparar grande número de simulações variando até dois parâmetros. Ela facilita a análise e interpretação dos resultados obtidos das simulações realizadas pelo simulador sim-outorder, por meio de gráficos e tabelas.

**Palavras-chave:** SuperSim, SimpleScalar, sim-outorder, simulação, arquitetura superescalar.

**ABSTRACT. Support interface for superscalar architectures design, simulation and evaluation.** This article presents a graphical interface for the SimpleScalar tool that assists in project design, monitoring and simulation of superscalar architectures. The interface allows the designer to configure several architectural parameters and to execute a great number of simulations varying up to two parameters. It facilitates the analysis and interpretation of the simulations results executed by the sim-outorder simulator, through graphs and tables.

**Key words:** SuperSim, SimpleScalar, sim-outorder, simulation, simplescalar architecture.

## Introdução

Simulação é uma técnica de avaliação que representa o comportamento de um sistema no domínio do tempo por meio de um modelo (FERRARI, 1988). Os processadores modernos são dispositivos extremamente complexos de engenharia. Portanto, pesquisas e modelagens de processadores se tornaram também muito complexas, requerendo, assim, excelentes ferramentas de simulação. Construir protótipos de processadores em *hardware* tem custo muito elevado, além de consumir muito tempo, particularmente nas fases iniciais do projeto. A modelagem matemática de performance para as atuais gerações de processadores pode ser incalculável, pelo grande uso de *caches*, execuções fora de ordem e especulatividade presentes nos projetos destes processadores (WENISCH et al., 2006; YI; LILJA, 2006).

Para acelerar o desenvolvimento do *hardware*, projetistas frequentemente empregam a técnica de simulação do mesmo. Eles implementam em *software* o modelo do *hardware* a ser desenvolvido, usando linguagens de programação tradicionais. Os projetistas podem, então, executar programas de testes neste modelo, a fim de avaliar a performance e correção do projeto do *hardware* proposto.

Programadores podem usar modelos de *software* para desenvolver e testar um *hardware* antes mesmo de este se tornar real. Apesar de modelos em *software* serem mais lentos do que o *hardware* em si, eles podem ser construídos, testados e modificados mais rapidamente, se comparados à construção de um *hardware*. Sendo assim, a simulação torna-se um rápido mecanismo de projeto e teste, disponibilizando mais rapidamente o produto para o mercado com melhor qualidade (AUSTIN et al., 2002).

Para a modelagem, em *software*, de processadores superescalares, uma ferramenta muito utilizada é o *SimpleScalar*, que é um conjunto de ferramentas para análise de desempenho de processadores, desenvolvido e gratuitamente disponibilizado pelo Departamento de Computação da Universidade de Wisconsin-Madison. Esta ferramenta é composta por um conjunto de simuladores configuráveis para diferentes arquiteturas de computadores e oferece simulação detalhada e de alto desempenho para diferentes partes de um processador superescalar e sua hierarquia de memória. Suas principais características são: flexibilidade, portabilidade, extensibilidade e desempenho (BURGER; AUSTIN, 1997).

No entanto, a ferramenta *SimpleScalar* não apresenta uma interface amigável. Não há facilidades

para uso da ferramenta na execução de simulações, tampouco facilitades para coletar dados e resultados das simulações executadas. As simulações são executadas e configuradas por linha de comando, por meio do console do sistema operacional. E os resultados das simulações são apresentados no formato de texto, o que dificulta a análise dos resultados obtidos.

Nesse sentido, o presente trabalho endereça este problema de interface da ferramenta *SimpleScalar* pela construção de uma ferramenta implementada e executada sobre a plataforma Java®, para o simulador sim-outorder do conjunto de ferramentas *SimpleScalar*. Assim, a execução e a configuração do simulador podem ser feitas por componentes de interface gráfica com o usuário, e os resultados obtidos das simulações exibidos em formato de tabelas e gráficos de linha. A ferramenta de interface, denominada SuperSim, viabiliza facilidades para projeto, simulação e avaliação de desempenho de processadores superescalares.

Alguns trabalhos relacionados são apresentados na obtenção de uma interface gráfica para utilização da ferramenta *SimpleScalar*.

Burger et al. (2004) desenvolveram uma ferramenta gráfica que permite ao usuário visualizar o desempenho de programas em pontos arbitrários dos *pipeline*. Essa ferramenta é baseada em Pearl/Tk e permite a visualização do *pipeline* de instruções, utilização de recursos e acessos às memórias.

Em seu trabalho, Moure et al. (2002) desenvolveram o KScalar, uma ferramenta gráfica para simulação de processadores. Essa ferramenta permite que estudantes e pesquisadores executem o comportamento de uma grande quantidade de arquiteturas de processadores: desde os mais simples de execução em ordem, *pipeline* escalar, até os mais detalhados com execuções fora de ordem, *pipeline* superescalar com caches não bloqueantes, execução especulativa. A ferramenta KScalar ainda pode ser utilizada para realização de grandes projetos que envolvam otimização de programas reais na interface de *software-hardware* ou a otimização da arquitetura do processador para uma determinada aplicação.

## Material e métodos

A ferramenta SuperSim foi desenvolvida utilizando-se a linguagem de programação Java®. Esta foi escolhida por ser uma linguagem orientada a objetos, que facilita a modularização e a especificação das classes do projeto e, principalmente, por ser uma linguagem multiplataforma, permitindo que a ferramenta seja utilizada em qualquer sistema

operacional para o qual esteja disponível o conjunto de ferramentas *SimpleScalar*.

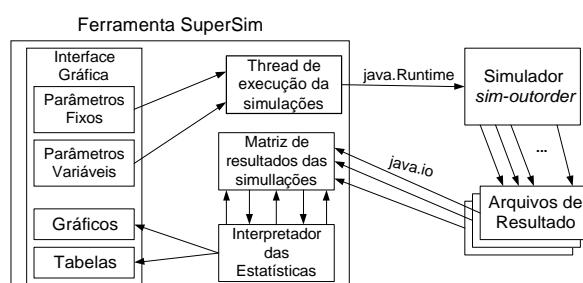
O ambiente de desenvolvimento, para a linguagem Java®, foi o Eclipse 3.1.1, por ser um ambiente de uso gratuito e por oferecer diversos recursos para a implementação e execução de testes da ferramenta. O sistema operacional no qual a ferramenta SuperSim foi implementada é o Linux, distribuição Slackware 10.

A parte mais importante da apresentação dos resultados das simulações pela ferramenta SuperSim são os gráficos, pois eles podem auxiliar a análise do desempenho da arquitetura sob determinada configuração, mostrando como alguns parâmetros (número de faltas da memória *cache*, número de instruções executadas por ciclo etc.) variam ao longo de diversas simulações. Para implementação dos gráficos na versão final da ferramenta, foi utilizada uma biblioteca para construção de gráficos em Java®, denominada JFreeChart, versão 0.9.11. Esta biblioteca foi escolhida porque, além da criação dos gráficos, permite que eles sejam impressos, salvos e exportados para outros formatos.

## Ambiente de simulação

O ambiente de simulação é composto basicamente de duas partes: a primeira é o simulador sim-outorder do conjunto de ferramentas *SimpleScalar*, que realiza as simulações e gera os resultados. A segunda parte é a ferramenta SuperSim, a qual realiza a configuração do sim-outorder, carrega o *benchmark* a ser executado pelo simulador e seus respectivos argumentos de execução, como também realiza a leitura e processamento dos resultados obtidos pelo simulador, apresentando-os graficamente.

A comunicação entre a ferramenta e o simulador, é realizada pelas chamadas do sistema operacional, ou seja, a ferramenta realiza a montagem da linha de comando, com a configuração captada pela interface gráfica, e a dispara no simulador sim-outorder, como está esquematizado na Figura 1.



**Figura 1.** Esquema de comunicação entre a ferramenta SuperSim e o simulador sim-outorder.

## Resultados e discussão

A ferramenta SuperSim foi desenvolvida com o intuito de prover facilidades de configuração para o simulador sim-outorder. Pelos componentes de interface gráfica com o usuário, a ferramenta permite executar diversas simulações com o mesmo *benchmark*, possibilitando a variação de até duas opções de configuração do simulador, sobre um intervalo predefinido. A ferramenta SuperSim converte os dados gerados pelo sim-outorder, da formatação do tipo texto para o formato de gráficos de linha e tabelas.

## Interface de configuração

Todos os 54 argumentos de configuração do simulador sim-outorder podem ser configurados pela interface da ferramenta SuperSim. No total são quatro menus, e cada um configura um subconjunto das opções existentes no simulador, como descrito a seguir: *Menu General* traz as opções gerais do simulador, como redirecionamento de saída dos *benchmarks*, restrições no número de instruções executadas e outras opções; *Menu Core* apresenta as opções que especificam o núcleo do simulador, como a largura de busca, largura do LSQ e também do RUU, o número de instruções *commit*, além da opção de marcar ou não o modo *backward-compatible bugs*; *Menu Branch* permite especificar qual dos seis tipos de previsor de desvios será utilizado nas simulações, assim como seus subsequentes argumentos; *Menu Memory* especifica a hierarquia de memória do simulador, por meio dos itens de *Cache* de Dados, *Cache* de Instruções, TLB e Opções Gerais, cada qual com seus respectivos argumentos de configuração; *About* contém as informações sobre os autores da ferramenta. A Figura 2 apresenta essa interface principal.

Cada um dos itens de menu abre uma caixa de diálogo capaz de configurar um determinado conjunto de parâmetros de um item arquitetônico do simulador.



Figura 2. Interface principal da ferramenta.

A Figura 3 apresenta a caixa de diálogo que configura as opções referentes ao *cache* de dados do simulador sim-outorder.

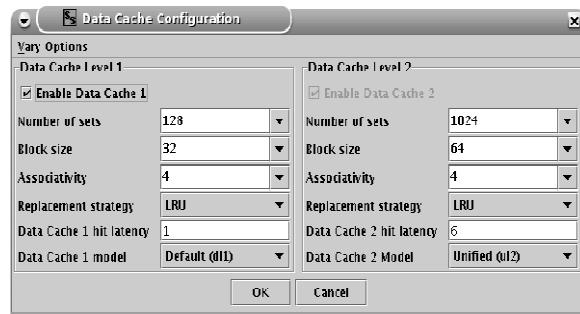


Figura 3. Caixa de diálogo para configuração do cache de dados.

Como pode se visualizar pela Figura 2, a interface da ferramenta SuperSim possui outros componentes gráficos que facilitam o seu uso, como: *Barra de Ferramentas*, que dá acesso às principais funcionalidades da ferramenta, como mostrar o comportamento de um determinado parâmetro no gráfico e as tabelas com todos os resultados; *Configurações*, que informa três parâmetros definidos para o simulador, como o número de simulações que serão realizadas, as tabelas de resultados que serão mostradas e as opções variáveis escolhidas; *Barra de Status*, que informa o tempo de simulação decorrido e o estado do simulador (*Stopped*, *Ready*, *Simulating*, *Terminate* e *Loaded*).

## Apresentação de resultados na interface

A interpretação de resultados obtidos por um grande número de simulações é uma atividade complexa, principalmente se os dados forem apresentados no formato de texto, como ocorre com a ferramenta *SimpleScalar*. Isto torna extremamente difícil a análise e o aprendizado sobre esses dados. Portanto, uma das mais importantes funcionalidades da ferramenta SuperSim é permitir que os resultados obtidos pela execução de várias simulações sejam dispostos de forma clara e objetiva por meio de gráficos de linha e tabelas, para qualquer um dos parâmetros resultantes da simulação, mostrando sua variação ao longo das simulações.

A Figura 4 ilustra um exemplo de gráfico de faltas no *cache* (*cache misses*), obtido após a execução de várias simulações, variando o número de conjuntos (*sets*) de 128 a 2048 e a associatividade do *cache* de 1 a 4. Pelo gráfico, pode-se analisar o comportamento da taxa de faltas no *cache* em função do número de conjuntos e da associatividade do *cache*. Essa apresentação gráfica facilita muito a análise dos resultados.

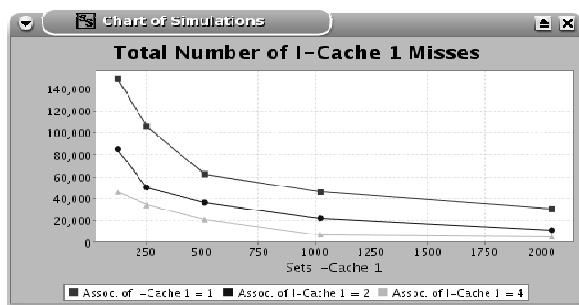


Figura 4. Gráfico de resultados (Conjuntos do I-Cache 1 x Número total de faltas do I-Cache 1).

### Simulações com SuperSim

Para testar as funcionalidades da ferramenta SuperSim, foram executadas diversas simulações que utilizam alguns *benchmarks* do pacote Spec2000-little (HENNING, 2000), trabalhando a variação de até dois parâmetros durante as simulações e exercitando o máximo possível dos 54 argumentos configuráveis do simulador sim-outorder. Os *benchmarks* utilizados para os testes foram: applu, art, bzip, cc1, gzip, quake, mcf, parser, mesa, mgrid, twolf, swim e vpr.

Esta seção apresenta dois exemplos de simulação utilizando a ferramenta SuperSim, para melhor visualização e compreensão de suas funcionalidades.

O primeiro experimento de simulação consiste em alterar o tipo do previsor de desvios para *Taken* e variar o número de conjuntos e da associatividade do *cache* de instruções de nível 1. Logo após o término das simulações, é analisado o comportamento do *cache* em função do número de conjuntos e da associatividade. A sequência do experimento de simulação é descrita a seguir.

Inicialmente, deve-se selecionar o *benchmark* que será utilizado durante o processo de simulação, por meio do botão *Open the benchmark to simulate*, na barra de ferramentas. Para o exemplo foi selecionado o *benchmark* *mgrid.ss*. Para fins ilustrativos, o contador de instruções foi ativado para 500 milhões de instruções para o *benchmark*. Isso permite que o *benchmark* seja executado até atingir 500 milhões de instruções. Em seguida, devem-se configurar os parâmetros que se manterão fixos durante a simulação. Para esta simulação, foi alterado o tipo do previsor de desvios de *Bimod* (padrão da ferramenta) para *Taken*, como ilustra a Figura 5. O próximo passo é configurar os parâmetros que variarão durante o processo de simulação. Para o exemplo, variou-se o número de conjuntos, de 128 a 2048 (Figura 6), a associatividade, de 1 a 4 vias (Figura 7), o tamanho de bloco do *cache* de instruções de primeiro nível foi alterado para 16. A Figura 8

apresenta como ficou a caixa de diálogo de configuração do *cache* de instruções após a escolha dos parâmetros variáveis.

Quando se escolhem dois parâmetros variáveis, o número total de simulações é  $n_1 \times n_2$ , em que  $n_1$  é o número de simulações com o primeiro parâmetro e  $n_2$  é o número de simulações com o segundo parâmetro. Para o exemplo,  $n_1 = 5$  (número de conjuntos 128, 256, 512, 1024 e 2048) e  $n_2 = 3$  (associatividade: 1, 2 e 4), o que totaliza 15 simulações.

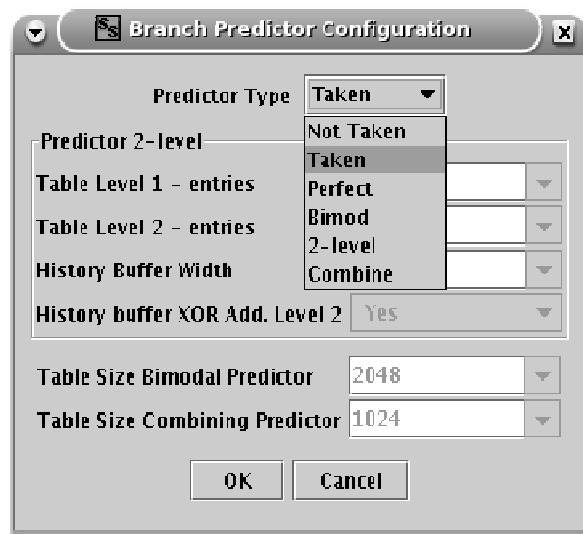


Figura 5. Tela de configuração do previsor de desvios.

A simulação é iniciada ativando-se *start benchmark simulation*. Então, a ferramenta exibe uma caixa de diálogo requisitando a entrada de argumentos, se estes forem necessários para a execução do *benchmark*. Para o exemplo, o argumento é a entrada do fluxo do arquivo *mgrid.in*. Feito isso, clica-se em *OK* para iniciar a simulação, assim o relógio da barra de *status* começa a contar o tempo de simulação e a barra mostra a quantidade de simulações realizadas, conforme a Figura 9.

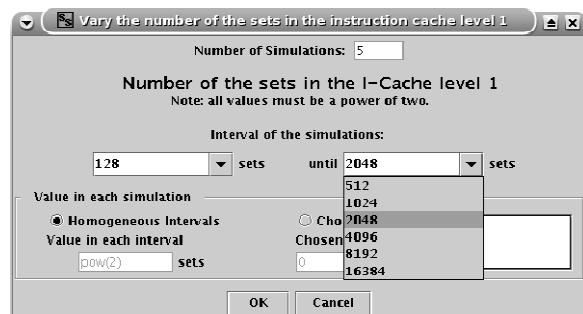
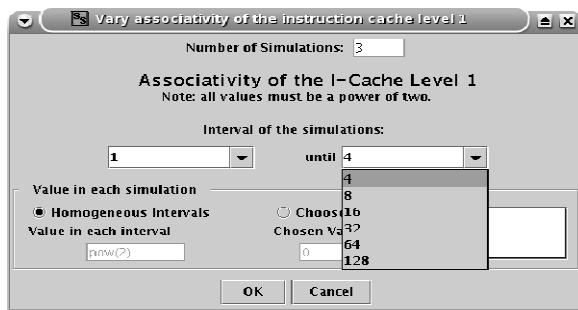
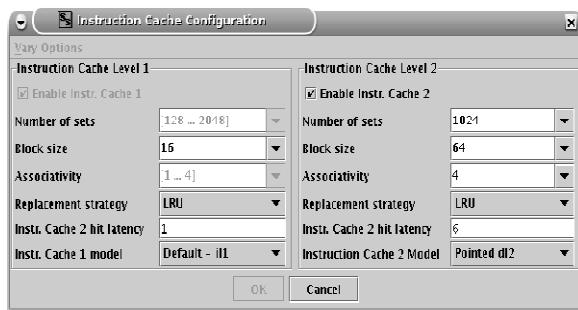
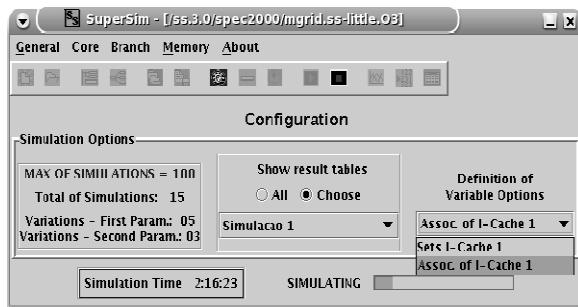


Figura 6. Tela de configuração do número de conjuntos do *cache*.

Figura 7. Tela de configuração da associatividade do *cache*.Figura 8. Configuração do *cache* de instruções (*instruction cache*).Figura 9. Interface principal exibindo as opções variáveis e o status da simulação do *benchmark*.

No segundo experimento de simulação, é analisado o comportamento da arquitetura sobre a variação do número de busca de instruções e a variação do número de instruções decodificadas. Para este exemplo, foi selecionado o *benchmark* de ponto flutuante *applu.ss*. O contador de instruções foi ativado para 500 milhões de instruções para o *benchmark*.

Em seguida, devem-se configurar os parâmetros que permanecerão fixos durante a simulação; neste caso, todas as unidades funcionais foram alteradas para seis, como ilustra a Figura 10. O próximo passo é configurar os parâmetros que irão variar durante a simulação, são eles: o número de busca de instruções (*Instruction Fetch*), variando de 2 a 32, vide Figura 11, e número de instruções decodificadas (*Instruction Decode*), variando de 2 a 64. O novo estado da caixa de diálogo das opções de busca e decodificação é apresentado na Figura 12.

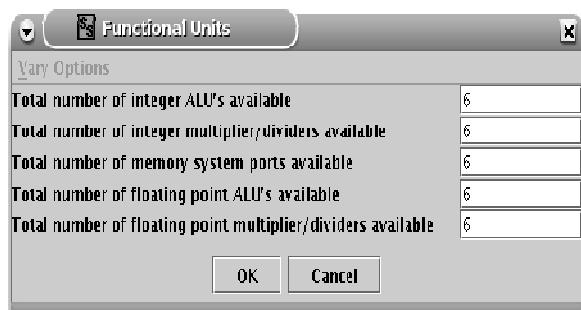


Figura 10. Caixa de diálogo de unidades funcionais com os novos valores.

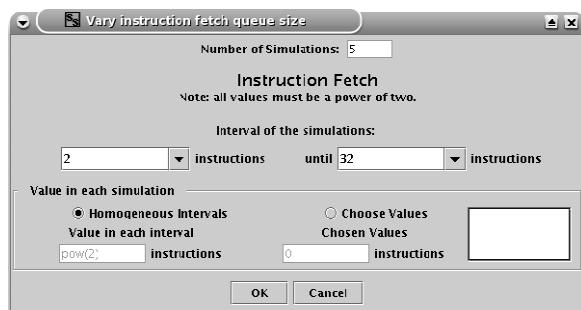


Figura 11. Caixa de diálogo de configuração para busca de instruções.

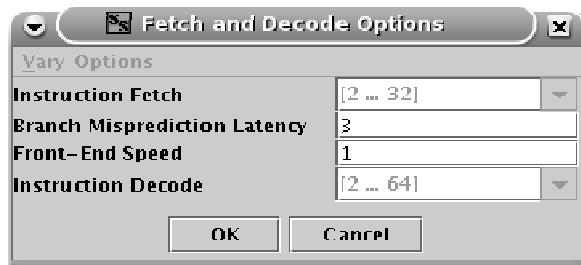


Figura 12. Caixa de diálogo com a nova configuração das opções de busca e decodificação de instruções.

## Resultados das simulações com SuperSim

Terminado o processo de simulação, é possível verificar o comportamento de qualquer um dos parâmetros gerados como resultado. Eles podem ser visualizados de diversas formas.

Primeiramente seleciona-se o parâmetro a ser visualizado, por meio do botão *select chart option* da barra de ferramentas, localizada na interface principal da ferramenta. Uma caixa de diálogo é aberta, exibindo todos os parâmetros que podem ser visualizados por meio de gráficos e tabelas. Dependendo do modelo de memória *cache* e previsor de desvios, alguns parâmetros podem não estar disponíveis para visualização.

Para o experimento 1, como parâmetros variáveis foram utilizados o número de conjuntos e a associatividade do *cache* de instruções. Neste caso, os

parâmetros que são visualizados de melhor forma são aqueles referentes à memória *cache* de instruções, pois estes sofreram maior variação ao longo das simulações. Para o exemplo, foi requisitado o gráfico *Total Number of Cache 1 Misses*, ou seja, o gráfico sobre o número de faltas no *cache* de instruções. Uma caixa de diálogo com o gráfico requisitado é aberta, como ilustra a Figura 13.

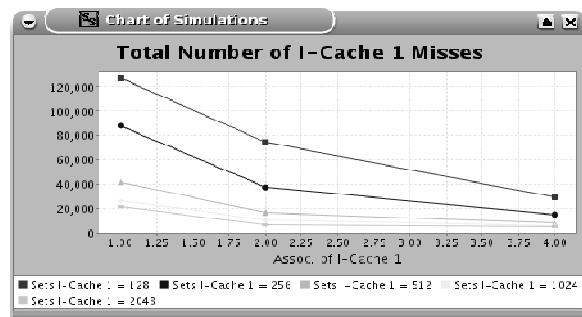


Figura 13. Assoc. of I-Cache 1 x I-Cache 1 Misses.

Para o gráfico da Figura 13, foi selecionada a variável *Assoc. of I-Cache 1*, para o eixo *x*, e cada linha representa o número de faltas no *cache* em função do número de conjuntos da memória *cache*. Mas esta representação pode ser alterada para a outra opção de parâmetro variável, pela interface principal no componente *Definition of Variable Table*, selecionando o item *Sets of I-Cache 1*, o qual será exibido no eixo *x*; cada linha representará o número de faltas no *cache* em função da associatividade do *cache*, como apresenta a Figura 14.

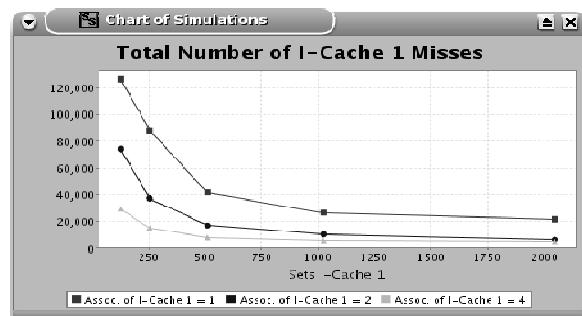


Figura 14. Sets of I-Cache 1 x I-Cache 1 Misses.

Outra forma de representação de resultados na ferramenta SuperSim é mediante tabelas. Ativando-se o botão *Show result tables*, da barra de ferramentas, podem-se exibir as tabelas com todas as estatísticas geradas pelo simulador. Pela interface principal é possível escolher se serão exibidas todas as tabelas de resultados de uma única vez, como na Figura 15, ou individualmente. Pela opção *Show results of chart option*, da barra de ferramentas, é possível visualizar o

valor da opção selecionada no gráfico, para cada uma das simulações.

Esses gráficos e tabelas possibilitam a avaliação de desempenho para determinada configuração de processador. Por exemplo, o gráfico da Figura 13 permite analisar o comportamento da taxa de faltas no *cache* de instruções de nível 1, em função da variação da associatividade e do número de conjuntos. Esses resultados viabilizam a escolha da configuração do *cache* com melhor desempenho e eficiência para a aplicação executada na simulação. É possível, por exemplo, executar um intenso experimento de simulação para uma determinada classe de aplicação e definir a melhor configuração do processador para esta classe de aplicação, com base na avaliação de desempenho extraída por meio dos resultados apresentados pela ferramenta SuperSim.

Edit		Simulation Results - mgrid.ss-little.O3														
		Simul. 11	Simul. 12	Simul. 13	Simul. 14	Simul. 15	Simul. 1	Simul. 2	Simul. 3	Simul. 4	Simul. 10					
<b>Simul. 1</b>		Simul. 5	Simul. 6	Simul. 7	Simul. 8	Simul. 9	<b>Assoc. of I-Cache 1 : 128</b>									
<b>Assoc. of I-Cache 1 : 1</b>																
Option		Results														
Number of instructions committed		380051545														
Number of loads and stores committed		14596069														
Number of loads committed		124762704														
Number of stores committed		20813365.0000														
Number of branches committed		13382473														
Simulation time in seconds		25.87														
Simulation speed (in insts/sec)		146908.2122														
Number of instructions executed		500000001														
Number of loads and stores executed		202281347														
Number of loads executed		154980562														
Number of stores executed		48200785.0000														
Number of branches executed		28333948														
Total simulation time in cycles		287097365														
Instructions per cycle		1.3382														
Cycles per instruction		0.7472														
Total instructions (mis-spec + committed) per cycle		1.7606														
Instruction per branch		28.3992														
Cumulative IFQ occupancy		377020097														
Cumulative IFQ full count		244224236														
Average IFQ occupancy (instn's)		3.4403														
Average IFQ dispach rate (instn/cycle)		1.7606														
Average IFQ occupant latency (cycle's)		1.9540														
Fraction of time (cycles) IFQ was full		0.8600														
Cumulative RUU occupancy		3518776322														
Cumulative RUU full count		147298868														

Figura 15. Tabela de resultados das simulações realizadas.

Para o experimento 2, como parâmetros variáveis foram utilizados o número de busca de instruções e o número de instruções decodificadas por ciclo. Os parâmetros que são visualizados de melhor forma são aqueles referentes ao núcleo (core), pois estes sofreram maior variação ao longo das simulações. Para o exemplo, foi requisitado o gráfico *Instructions per Cycle (IPC)*, ou seja, o gráfico sobre o número de instruções executadas por ciclo. Uma caixa de diálogo com o gráfico requisitado é aberta, como apresenta a Figura 16. Para o gráfico da Figura 16, foi selecionada a variável *Instruction Fetch*, para o eixo *x*, e cada linha representa o número de instruções decodificadas por ciclo. Mas esta representação pode ser alterada para a outra opção de parâmetro variável, pela interface principal, selecionando o item

*Instruction Decode*, o qual será exibido no eixo  $x$ , e cada linha representará IPC em função da busca de instruções, como ilustra a Figura 17.

Pela opção *Show results of chart option*, da barra de ferramentas, é possível visualizar o valor da opção selecionada no gráfico, para cada uma das simulações, como apresenta a Figura 18. Pelos gráficos das Figuras 16 e 17, pode-se, por exemplo, definir qual a melhor configuração das unidades de busca e decodificação de instruções, para se obter melhor desempenho.

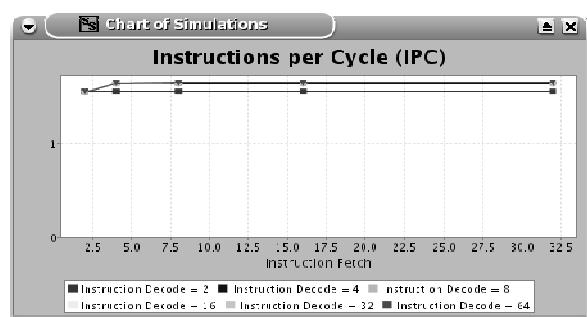


Figura 16. Instructions per Cycle (IPC) x Instructions Fetch.

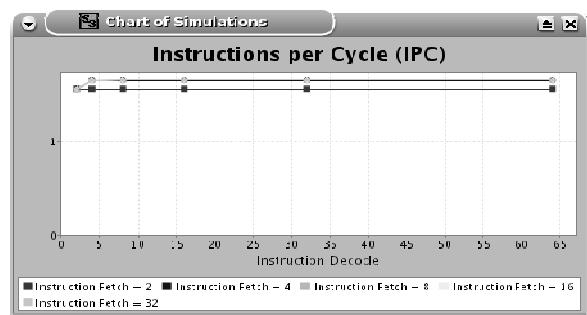


Figura 17. Instructions per Cycle (IPC) x Instructions Decode.

Simulation	Variable option 1	Variable option 2	Result
Simulation 1	Instruction Fetch = 2	Instruction Decode = 2	1.5544
Simulation 2	Instruction Fetch = 4	Instruction Decode = 2	1.5555
Simulation 3	Instruction Fetch = 8	Instruction Decode = 2	1.5555
Simulation 4	Instruction Fetch = 16	Instruction Decode = 2	1.5555
Simulation 5	Instruction Fetch = 32	Instruction Decode = 2	1.5555
Simulation 6	Instruction Fetch = 2	Instruction Decode = 4	1.5544
Simulation 7	Instruction Fetch = 4	Instruction Decode = 4	1.6428
Simulation 8	Instruction Fetch = 8	Instruction Decode = 4	1.6428
Simulation 9	Instruction Fetch = 16	Instruction Decode = 4	1.6429
Simulation 10	Instruction Fetch = 32	Instruction Decode = 4	1.6429
Simulation 11	Instruction Fetch = 2	Instruction Decode = 8	1.5544
Simulation 12	Instruction Fetch = 4	Instruction Decode = 8	1.6428
Simulation 13	Instruction Fetch = 8	Instruction Decode = 8	1.6441
Simulation 14	Instruction Fetch = 16	Instruction Decode = 8	1.6441
Simulation 15	Instruction Fetch = 32	Instruction Decode = 8	1.6441
Simulation 16	Instruction Fetch = 2	Instruction Decode = 16	1.5544
Simulation 17	Instruction Fetch = 4	Instruction Decode = 16	1.6428
Simulation 18	Instruction Fetch = 8	Instruction Decode = 16	1.6441
Simulation 19	Instruction Fetch = 16	Instruction Decode = 16	1.6441
Simulation 20	Instruction Fetch = 32	Instruction Decode = 16	1.6442
Simulation 21	Instruction Fetch = 2	Instruction Decode = 32	1.5544
Simulation 22	Instruction Fetch = 4	Instruction Decode = 32	1.6428
Simulation 23	Instruction Fetch = 8	Instruction Decode = 32	1.6441
Simulation 24	Instruction Fetch = 16	Instruction Decode = 32	1.6441
Simulation 25	Instruction Fetch = 32	Instruction Decode = 32	1.6442
Simulation 26	Instruction Fetch = 2	Instruction Decode = 64	1.5544
Simulation 27	Instruction Fetch = 4	Instruction Decode = 64	1.6428
Simulation 28	Instruction Fetch = 8	Instruction Decode = 64	1.6441
Simulation 29	Instruction Fetch = 16	Instruction Decode = 64	1.6441
Simulation 30	Instruction Fetch = 32	Instruction Decode = 64	1.6442

Figura 18. Tabela indicando o valor de IPC em cada uma das simulações.

## Conclusão

A técnica de simulação e o uso de excelentes ferramentas de simulação se tornaram essenciais na área de Arquitetura de Computadores, uma vez que as pesquisas e modelagens de processadores modernos se tornaram muito complexas. Entre as vantagens do uso de técnicas de simulação estão: a avaliação do desempenho de uma arquitetura antes de sua implementação, detecção de falhas de projeto e viabilização de correções a um menor custo, identificação da viabilidade do projeto em uma arquitetura em função de vários parâmetros como desempenho, consumo, encapsulamento etc.

A ferramenta SuperSim é capaz de realizar a integração do simulador sim-outorder, que é amplamente utilizado por diversos grupos de pesquisas mundialmente reconhecidos, em um ambiente em que o simulador pode ser configurado por componentes gráficos e os resultados gerados pelas simulações podem ser apresentados em formato de gráficos e tabelas, o que facilita muito a avaliação do comportamento e do desempenho do processador durante o projeto.

Entre as principais funcionalidades da ferramenta estão: a possibilidade de realizar diversas simulações, variando até dois parâmetros, e a possibilidade de visualizar os resultados gerados pela variação dos parâmetros ao longo das simulações. A ferramenta SuperSim tornou muito mais fácil projetar, simular e avaliar o comportamento de uma arquitetura em diferentes configurações.

Pelos seus diversos recursos, a ferramenta SuperSim pode ser utilizada, no meio acadêmico, como ferramenta de apoio ao ensino da disciplina de Arquitetura de Computadores, tanto na graduação quanto na pós-graduação. Além disso, pode ser usada como ferramenta de simulação para grupos de pesquisas na área de Arquiteturas de Alto Desempenho.

## Referências

- AUSTIN, T. M.; ERNST, D.; LARSON, E. SimpleScalar: an infrastructure for computer system modeling. **IEEE Computer**, v. 35, n. 2, p. 59-67, 2002.
- BURGER, D. C.; AUSTIN, T. M. The SimpleScalar Tool Set: version 2.0. **ACM SIGARCH Computer Architecture News**, v. 25, n. 3, p. 13-25, 1997.
- BURGER, D. C.; AUSTIN, T.; KECKLER, S. Recent extensions to the SimpleScalar tool suite. **ACM SIGMETRICS Performance Evaluation Review**, v. 31, n. 4, p. 4-7, 2004.
- FERRARI, D. **Computer systems performance evaluation**. Englewood Cliffs: Prentice Hall, 1988.

- HENNING, J. L. SPEC CPU2000: measuring CPU performance in the new millennium. **IEEE Computer**, v. 33, n. 7, p. 28-35, 2000.
- MOURE, J. C.; REXACHS, D. I.; LUQUE, E. The KScalar Simulator. **ACM Journal on Educational Resources in Computing**, v. 2, n. 1, p. 73-116, 2002.
- WENISCH, T. F.; WUNDWELICH, R. E.; FERDMAN, M.; AILAMAKI, A.; FASAFI, B.; HOE, J. C. Simplex: statistical sampling of computer system simulation. **IEEE Micro**, v. 26, n. 4, p. 18-31, 2006.
- YI, J. J.; LILJA, D. J. Simulation of computer architecture:

simulations, benchmarks, methodologies and recommendations. **IEEE Transactions on Computers**, v. 55, n. 3, p. 268-280, 2006.

Received on November 23, 2007.

Accepted on August 12, 2008.

License information: This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.