

RICHARD REMPEL DE SOUZA

AVALIAÇÃO DE CACHES ESPECIALIZADAS PARA
SISTEMAS EMBARCADOS

Trabalho de Graduação apresentado ao curso de Ciência da Computação, Setor de Ciências Exatas, Universidade Federal do Paraná, como requisito parcial para a conclusão do curso.

Orientador: Prof. Dr. Roberto A Hexsel

CURITIBA

2009

Avaliação de Caches Especializadas para Sistemas Embarcados

Richard Rempel de Souza

Conteúdo

1	Introdução	1
2	Ambiente de Simulação	3
2.1	SimpleScalar	3
2.2	Modelagem dos processadores	3
2.3	Modelagem da hierarquia de memória	4
2.4	MediaBench	5
3	Modelos de Cache	8
3.1	MSHR	8
3.2	Cache de Vítimas	9
3.3	Stream Buffer	9
3.4	Pollution Control Cache	10
4	Resultados	12
4.1	CommBench – Caches Grandes × Pequenas	12
4.2	MediaBench	14
4.3	Stream Buffer – Barramento × Latência	15
5	Conclusão	19
	BIBLIOGRAFIA	21
A	Script Bash	22
A.1	Script Principal	22
A.2	Funções de execução	24
A.3	Funções de apoio	25
A.4	Linha de comando do Benchmark	26
B	Script para criação da base de dados	27
C	Script AWK	40

Lista de Tabelas

2.1	Modelos de CPU	4
2.2	Parametros da simulados L1	4
2.3	Características dos programas do MediaBench no SimpleScalar	6
2.4	Características dos programas do CommBench no SimpleScalar	7
4.1	Tempos de tranferência (ciclos) de um bloco entre L1 e memória	16

Lista de Figuras

3.1	Cache de Controle de Poluição (PCC).	10
3.2	Cache de vítima e Controle de Poluição (PCVC).	11
4.1	Média do IPC dos programas como simulados em [7], as simulações para as capacidades de 32 e 64Kbytes pertencem a um novo conjunto de testes. . . .	13
4.2	(a) Media do IPC vs capacidade, (b) Media da taxa de faltas vs capacidade.	13
4.3	(a) Media do IPC vs capacidade, (b) Media da taxa de faltas vs capacidade.	14
4.4	Comparativo Bloco x Associatividade do SB.	16
4.5	CPU larg=2, Bloco=16 Associatividade do SB=2.	17

Resumo

A disparidade entre a evolução da velocidade dos microprocessadores e a latência no acesso às memórias, torna o projeto da hierarquia de memória um importante fator no desempenho global dos sistemas com microprocessadores. Memórias cache é o item dessa hierarquia que permite diminuir o tempo médio de acesso à memória principal. Embora úteis, memórias cache podem ocasionar impactos negativos em parâmetros importantes para o projeto de sistemas embarcados, tais como consumo de energia, tamanho do circuito integrado, confiabilidade e custo.

A investigação deste trabalho parte de um conjunto de parâmetros e técnicas para projeto de caches de dados apresentado em [7]. Nosso objetivo é a busca por modelos de memória cache que apresentem os melhores resultados. Para tanto, os dados obtidos, a partir do trabalho base, para essa investigação, foram estendidos de duas maneiras: programas de teste que trabalham sobre um conjunto maior de dados e um refinamento do projeto do barramento entre memória e cache.

Os experimentos foram executados no simulador SimpleScalar com os programas de teste do pacote MediaBench. A Cache de Vítima com Controle de Poluição (Pollution Control Victim Cache – PCVC) é simples e tem desempenho superior para caches de capacidade entre 4 e 8Kbytes. Para capacidades maiores que 8Kbytes o *Stream Buffer* com o emprego de um barramento mais largo (16Bytes) resultou no melhor desempenho, contudo esta é a organização mais custosa investigada neste estudo.

Capítulo 1

Introdução

Sistemas embarcados são responsáveis por cerca de 98% da demanda do mercado de microprocessadores de 32 bits [4]. O estudo de técnicas de projeto de memórias cache visa reduzir o impacto causado pelas grandes latências da memória principal [8]. No entanto, a busca por um melhor desempenho não pode sacrificar requisitos chave na concepção de sistemas embarcados, como tamanho do circuito integrado e consumo de energia.

Um modelo de memórias cache que possui características compatíveis ao projeto de sistemas embarcados, é a Cache de vítima e Controle de Poluição (*Pollution Control Victim Cache* – PCVC) [6]. Este modelo foi concebido pela combinação da Cache de Controle de Poluição (*Pollution Control Cache* – PCC) e uma cache de vítimas, agregando a funcionalidade desses dois circuitos especializados.

Resultados de simulações de uma combinação entre PCVC e uma L1 de capacidade que varia de 1 a 16Kbytes, demonstrou resultados promissores pela simplicidade e desempenho superior em [7]. A PCVC foi simulada utilizando a suíte de simulações SimpleScalar e os programas de teste do pacote CommBench – núcleos de programas de processamento intensivo de aplicação comuns a processadores de rede [16].

O *Stream Buffer* (SB) provoca um aumento expressivo na taxa de ocupação do barramento, comportamento esse que reduz o desempenho global do sistema [7]. Buscamos informações sobre o impacto que variações nos recursos barramento entre memória e cache e latência de memória causam no desempenho do SB e se o aumento desses recursos torna o uso do SB vantajoso.

Objetivando projetar uma hierarquia de memória adequada para uma classe de aplicações de multimídia e de comunicações em sistemas embarcados, foram estendidos os estudos realizados e os resultados obtidos em [7]. O espectro de simulações foi modificado para: (i) conter programas que trabalhem sob um conjunto de dados maior e (ii) acompanhar o padrão de parâmetros para projeto de cache de processadores atuais, como a cache do ARM [1]. A primeira tarefa empregou um conjunto de programas de teste do pacote MediaBench [11], e a segunda uma cache de dados primária com tamanhos de 32 e 64Kbytes.

O texto está organizado como segue. No Capítulo 2 é descrito o ambiente de simulação, os programas de teste e as ferramentas e métodos utilizados no desenvolvimento da pesquisa. São apresentados no Capítulo 3 os modelos de cache e as técnicas investigadas. Os resultados sobre emprego de circuitos especializados são expostos e discutidos no Capítulo 4.

Capítulo 2

Ambiente de Simulação

Neste Capítulo são apresentadas as ferramentas para execução e tratamento dos resultados utilizados durante o desenvolvimento da pesquisa. Os parâmetros das simulações são definidos, explicitando as extensões aos parâmetros utilizados em [7].

2.1 SimpleScalar

O SimpleScalar (SS) é um conjunto de simuladores que permitem a criação de modelos de arquiteturas de processadores e hierarquia de memórias [2, 14]. Utilizamos a versão (3.0D) do SS com suporte de emulação do conjunto de instruções PISA (*portable instruction set architecture*).

Com o PISA - extensão do conjunto de instruções do MIPS - e o respectivo compilador baseado no GNU GCC, é possível fazer uma avaliação detalhada de desempenho, a partir da execução de programas de teste reais que não estão inclusos no pacote do SS [3].

O sim-outorder é o simulador que suporta a execução de instruções fora de ordem em uma CPU superescalar e a apresentação de estatísticas precisas sobre o desempenho. A hierarquia de memória que o sim-outorder modela será referenciada como “versão original”. Essa hierarquia e os modelos modificados são especificados no Capítulo 3.

2.2 Modelagem dos processadores

Para a modelagem dos processadores é feito um estudo sobre três parâmetros: primeiro, escalaridade e largura dos estágios de busca, decodificação, emissão e retirada de instruções do pipeline e número de unidades funcionais na execução. Segundo, uma fila que armazena as referências à memória, que estão em execução (*Load/Store Queue* – LSQ). Terceiro, um *buffer* de reordenação de instruções, responsável pela renomeação dos registradores e por garantir que as instruções sejam terminadas em ordem (*Register Update Unit* –

RUU). A partir da combinação dos parâmetros apresentados são definidos quatro modelos de processadores. A Tabela 2.1 apresenta os parâmetros dos quatro processadores. No caso das estruturas LSQ e RUU, os valores representam o numero de registros de cada estrutura.

MODELO DE CPU	1	2	4	8
Escalaridade	1	2	4	8
Tamanho da LSQ	4	8	16	32
Tamanho da RUU	8	16	32	64

Tabela 2.1: Modelos de CPU

Os parâmetros referentes aos modelos de CPU são herdados, sem modificações, de [7]. A escolha desses valores relativamente pequenos para LSQ e RUU foi feita levando-se em consideração dois aspectos: uma implementação de baixo custo para sistemas embarcados e o pequeno impacto na ocupação dos recursos, empregando-se estruturas maiores e portanto mais caras.

2.3 Modelagem da hierarquia de memória

A hierarquia de memória conta com uma cache de primeiro nível (L1) e a memória principal. Não é usada uma cache de segundo nível. O tamanho da palavra de dados e de instruções é de 32 bits. A memória principal é intercalada, com um barramento de dados de 8Bytes (duas palavras). A latência para acesso ao primeiro par de palavras é de 18 ciclos, mais 2 ciclos por par de palavras subsequente no bloco. A cache L1 é dividida em uma cache de instruções com uma capacidade fixa de 32Kbytes e uma cache de dados.

Para a execução dos estudos referentes ao *Stream Buffer*, a memória principal possui duas latências possíveis: 18 e 12 ciclos, e o barramento entre memória e cache possui largura de 8 ou 16Bytes. Nos outros casos a latencia é 18 ciclos e o barramento 8Bytes. A cache de instruções foi simulada sem modificações em relação ao trabalho de referência.

Os valores escolhidos para os parâmetros da cache de dados são relacionados na Tabela 2.2.

PARÂMETROS	Valores
Tamanho da cache (KB)	1, 2, 4, 8, 16, 32 e 64
Largura do bloco (bytes)	8, 16, 32 e 64
Associatividade da cache	1 e 2

Tabela 2.2: Parametros da simulados L1

A cache de dados possui os seguintes parâmetros variáveis: capacidade, tamanho do bloco e associatividade. O tamanho do bloco da cache varia entre 8 e 64Bytes, um bloco

possui de duas até oito palavras de dados. A capacidade da cache de dados varia entre 1Kbyte até 64Kbytes. Tanto para o tamanho do bloco como para a capacidade os valores intermediários são potências de dois. A decisão para o incremento de capacidade em relação ao mesmo parâmetro no trabalho de referência é baseada na importância que os processadores ARM possuem no mercado de sistemas embarcados [1]. A cache primária de dados com 32Kbytes de capacidade do ARM tornou-se padrão do mercado.

2.4 MediaBench

O Mediabench é um conjunto de programas de teste para sistemas de comunicações e multimedia [11]. Diferentemente de suítes de testes constituídas por núcleos de aplicação e codificados em assembly, o Mediabench é composto por programas inteiros e totalmente escritos em linguagem de alto nível. O Mediabench contém um total de 19 programas que refletem aplicações de processamento de imagens, comunicação e processamento digital de sinais (DSP).

As estatísticas para avaliação de desempenho foram obtidas através da execução dos programas listados a seguir.

- JPEG é um método de compressão para imagens coloridas e tons de cinza. Dois programas são derivados do código fonte do JPEG; *jpeg_menc* faz a compressão de imagens e *jpeg_mdec* a descompressão;
- MPEG2 é o padrão para transmissão de vídeos de alta qualidade. Os dois programas utilizados são *mpeg2_enc* e *mpeg2_dec* para codificação e decodificação respectivamente;
- GSM padrão europeu GSM 06.10 para codificação de voz. As variantes são *toast* para a codificação e *untoast* para decodificação;
- G.712 Implementação de referência do CCITT (*International Telegraph and Telephone Consultative Committee*) para compressão de voz. *g721_enc* faz a compressão e *g721_dec* a descompressão;
- PEGWIT é um programa para encriptação e autenticação usando chave pública. *pegwit_enc* faz a encriptação e *pegwit_dec* a decríptação;
- MESA Biblioteca grafica 3-D clone do OpenGL. Todas as funções de apresentação gráfica foram removidas da biblioteca e dos programas de demonstração. Os três programas utilizados são: *mipmap* executa o mapeamento de texturas, *osdemo* executa

um pipeline de renderização padrão, e *texgen* que cria uma versão texturizada da Chaleira de Utah;

- EPIC utilitário para compressão de imagens experimental. Os filtros foram projetados para permitir uma rápida decodificação sem hardware de ponto flutuante. *epic_enc* faz a compressão e *epic_dec* a descompressão;
- ADPCM (*Adaptive Differential Pulse Code Modulation*) é uma das formas mais simples de codificação de áudio. *adpcm_enc* faz a codificação e *adpcm_dec* a decodificação.

CARACTERÍSTICA	Nr.inst.	Nr.refs.	Refs.	Nr.loads	Entre dsv.
Programa / unid.	×1000	×1000	%	×1000	
ADPCM dec	5.460	521	9,55	370	3,36
ADMPC enc	6.689	522	7,80	444	3,48
EPIC dec	6.783	1.672	24,65	901	4,97
EPIC enc	52.772	7.536	14,28	6.764	6,79
JPEG mdec	6.142	1.880	30,60	1.199	10,82
JPEG menc	6.728	1.584	23,55	1.109	11,52
MPEG2 enc	1.134.249	326.866	28,81	304.987	5,87
MPEG2 dec	171.170	33.076	19,32	26.628	8,51
GSM toast	234.584	51.821	22,09	40.356	20,62
GSM untoast	75.664	8.308	10,98	5.342	5,75
G.721 enc	274.865	47.894	17,42	36.552	4,35
G.721 dec	519.091	95.387	18,37	71.675	4,39
PEGWIT enc	33.853	8.399	24,81	6.393	8,83
PEGWIT dec	18.941	5.131	27,08	3.960	9,31
MESA mipmap	47.044	15.838	33,66	10.296	7,71
MESA osdemo	8.575	2.921	34,06	1.944	9,04
MESA texgen	76.126	23.911	31,41	15.212	9,91

Tabela 2.3: Características dos programas do MediaBench no SimpleScalar

A Tabela 2.3 mostra o número de instruções executadas, o número de referências à memória, a taxa de referências, o número de *loads* e uma relação entre o número total de instruções executadas e o número de instruções de desvio – considerada como o número de instruções entre desvios, quando executados no simulador SimpleScalar. A Tabela 2.4 mostra as informações referentes aos programas do CommBench, um subconjunto composto por *Cast_dec*, *Cast_enc*, *DRR*, *FRAG*, *Jpeg_enc* e *Jpeg_dec* é descrito em [7, 6]

CARACTERÍSTICA	Nr.inst.	Nr.refs.	Refs.	Nr.loads	Entre dsv.
Programa / unid.	×1000	×1000	%	×1000	
CAST dec	137.728	41.307	29.99	28.644	11.45
CAST enc	137.728	41.307	29.99	28.644	11.45
DRR	212.806	126.091	59.25	104.323	4.83
FRAG	42.965	10.308	23.99	6.403	5.07
JPEG dec	219.476	67.253	30.64	42.142	10.01
JPEG enc	302.884	81.615	26.94	54.395	8.24
REED dec	1205.329	215.278	17.86	146.839	5.48
REED enc	622.825	134.359	21.57	100.736	4.27
RTR	810.909	303.978	37.48	245.925	4.35
ZIP dec	39.360	10.357	26.31	8.511	5.86
ZIP enc	220.315	60.830	27.61	45.726	4.94

Tabela 2.4: Características dos programas do CommBench no SimpleScalar

Capítulo 3

Modelos de Cache

Os modelos e técnicas descritos abaixo consideram uma hierarquia de memória que possui uma cache de dados primária (L1) de mapeamento direto, uma ou mais estruturas auxiliares e uma memória principal. Salvo algumas exceções explicitadas no texto, os modelos são os mesmos descritos e concebidos no trabalho de referência [7].

3.1 MSHR

A cache não-bloqueante foi apresentada por Kroft em 1981 [10]. Utiliza-se de registradores e circuitos de controle que mantêm o estado de uma referência que causa uma falta para que a cache possa continuar a atender as referências da CPU. Esses registradores, chamados de MSHR (*miss status holding register*), mantêm valores como endereço em memória que causou a falta, unidade de execução que aguarda pela palavra, e campos de controle como um indicador de válido ou ocupado. A análise das simulações da cache não bloqueante indica que a utilização de um único MSHR serializa o acesso à memória e degrada o desempenho em relação ao modelo original. O desempenho em um modelo com 4 MSHRs, em uma cache que possui poucos blocos (1Kbyte=16x64 bytes/bloco), obteve ganhos de cerca de 50% no IPC sobre o modelo original, com uma CPU de largura 8.

Uma cache de mapeamento direto com a utilização do MSHR é o modelo base da pesquisa. A partir desse modelo, que oferece uma contabilização realista do custo das faltas na L1, serão apresentados os próximos modelos simulados. Diferentemente do trabalho de referência, estatísticas sobre o emprego do MSHR com uma cache de dados de associatividade 2 foram inseridas para que os custos – da adição de associatividade e outros modelos abordados – possam ser comparados.

3.2 Cache de Vítimas

A Cache de Vítimas (CV) é uma pequena cache, totalmente associativa, inserida entre a L1 e os níveis inferiores [9]. São carregados para a CV blocos que foram expurgados da cache principal. A CV adiciona um grau limitado de associatividade a uma cache de mapeamento direto, assim reduzindo as faltas por conflito. A política de troca de blocos na CV é LRU (*least recently used*). A referência é atendida simultaneamente na cache de dados e na CV. Caso o acerto ocorra na CV, haverá uma troca entre o bloco faltante na cache de dados e o bloco com acerto na CV, e esta operação tem um custo adicional de um ciclo. Os resultados da utilização de uma CV com 4 blocos mostram que a redução da taxa de faltas melhora o IPC, com ganhos de 2,6% em uma cache com capacidade 2Kbytes sobre o modelo base, com uma CPU de largura 2.

3.3 Stream Buffer

O *Stream Buffer* (SB) é uma estrutura que realiza a busca antecipada de blocos para a cache, muito eficiente na redução das faltas compulsórias e por capacidade [9]. Faltas compulsórias ocorrem quando um bloco é referenciado pela primeira vez. O comportamento das faltas compulsórias independe do grau de associatividade ou da capacidade da cache [8]. A partir do endereço de uma referência que causou uma falta na cache, o SB busca os blocos subseqüentes e os armazena em uma fila. A busca continua enquanto houver blocos livres na fila.

As referências são atendidas simultaneamente na cache e na fila do SB. Caso o acerto ocorra na fila, o bloco requisitado é entregue à CPU e é escrito na cache. Esta operação disponibiliza um ou mais blocos na fila permitindo a continuidade das buscas e tem custo de 1 ciclo adicional em caso de acerto. Na ocorrência de uma falta na cache e na fila, os blocos alocados na fila são expurgados e o processo de busca é reiniciado a partir do endereço da requisição faltante.

A busca especulativa de blocos resulta na redução da taxa de faltas na cache de dados L1 em até 25%, utilizando uma fila de 4 blocos, segundo dados apresentados em [9]. Devido ao severo impacto na ocupação do barramento, o tempo total de execução foi prolongado e o desempenho do SB se mostrou pior que o modelo base nos experimentos descritos em [7].

3.4 Pollution Control Cache

Apresentado por Walsh e Board, a PCC utiliza uma cache totalmente associativa, que opera em paralelo à cache L1 [15]. Diferentemente de outras técnicas apresentadas, os blocos que causaram uma falta são carregados inicialmente na PCC. Essa política é utilizada na prevenção da poluição da cache, que ocorre por blocos pouco referenciados, alocados no lugar de blocos frequentemente referenciados. A poluição de blocos gera duas faltas conjugadas, uma no acesso ao bloco (com referência infrequente) e outra na recarga do bloco original.

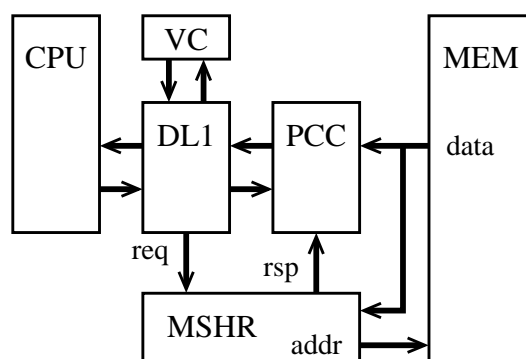


Figura 3.1: Cache de Controle de Poluição (PCC).

A PCC apresentada em [15] utiliza uma cache de dados e uma cache de controle de poluição, cada uma delas com um *Buffer* de Vítimas (versão modificada da cache de vítimas). Em [7] foram propostas duas implementações para a PCC. A PCC implementada segue parcialmente o modelo proposto por [15], e consiste em uma cache de vítima associada à L1, além da cache de controle de poluição. Um segundo modelo denominado Cache de Vítima e Controle de Poluição (*Pollution Control Victim Cache – PCVC*) utiliza a própria cache de controle de poluição como cache de vítima para a L1. Os diagramas da Figura 3.1 exibem as estruturas de dados e a organização dos modelos de PCC abordados neste estudo. A implementação da PCC implica que a estrutura DL1 representada na figura 3.1 possua três portas de leitura e escrita, para conexão com o processador, Cache de Vítimas e a PCC. A Figura 3.2 apresenta a organização da PCVC.

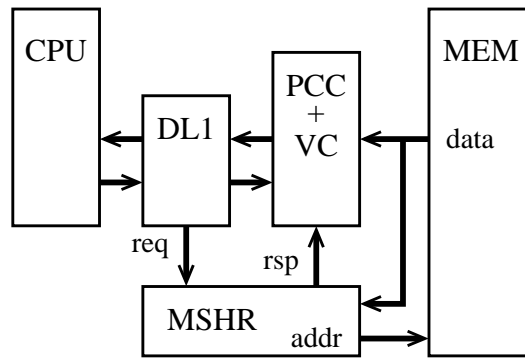


Figura 3.2: Cache de vítima e Controle de Poluição (PCVC).

As referências às palavras de dados são atendidas na L1 e na PCC paralelamente. Um acerto na PCC significa ser a segunda referência ao bloco, desta forma o bloco é transferido para a L1 liberando um bloco na PCC. O comportamento dos dois modelos propostos difere no momento da substituição entre o bloco transferido da PCC e o bloco expurgado da L1. A PCC escreve o bloco expurgado da L1 na cache de vítima associada. A PCVC aloca o bloco expurgado da L1 no espaço liberado pelo bloco que saiu da PCVC.

Capítulo 4

Resultados

Neste Capítulo são apresentados os resultados das simulações realizados para esta investigação. Na Seção 4.1 são apresentados os resultados referentes aos programas da suite CommBench. São comparados as simulações com caches de 32 e 64Kbytes com os resultados de [7]. O comportamento das duas implementações propostas para a PCC com a simulação dos programas do MediaBench é apresentado na Seção 4.2. O impacto que o barramento entre memória e cache, e a latência da memória causam na performance do *Stream Buffer* é discutido na Seção 4.3.

4.1 CommBench – Caches Grandes × Pequenas

São apresentados nesta seção os resultados das simulações dos onze programas de teste da Suíte CommBench. Os modelos estudados são cache de mapeamento direto com 4 MSHRs ($L1_{MD}$), cache com associatividade 2 e 4 MSHRs ($L1_{2w}$), PCC de 16 e 32 blocos com uma Cache de Vítimas com 4 registradores (PCC_{16} , PCC_{32}) e PCVC de 16 e 32 blocos ($PCVC_{16}$, $PCVC_{32}$). Todos os dados apresentados são resultantes da execução com um processador de largura 2 e blocos da cache de 32Bytes. Para todas as simulações com o PCC a Cache de Vítimas associada a L1 possui 4 blocos.

A Figura 4.1 apresenta a média do IPC dos programas simulados em [7]. São eles: *Cast_dec*, *Cast_enc*, *DRR*, *FRAG*, *Jpeg_enc* e *Jpeg_dec*. Esses programas são um subconjunto da suíte CommBench. Os dados referentes as caches com capacidade de até 16Kbytes são apresentados em [7], os dados para as maiores capacidades são novos. É possível observar que a PCVC é mais eficiente que outros modelos, principalmente em caches menores que 4Kbytes.

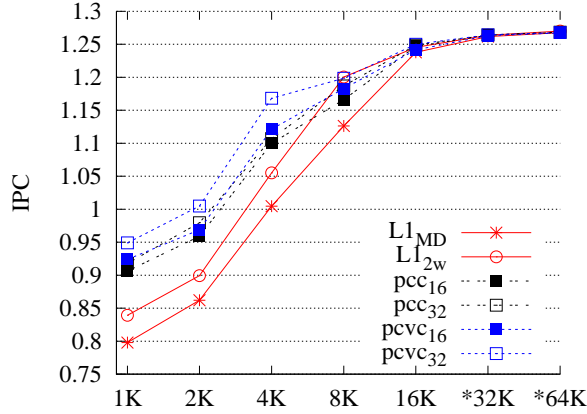


Figura 4.1: Média do IPC dos programas como simulados em [7], as simulações para as capacidades de 32 e 64Kbytes pertencem a um novo conjunto de testes.

A Figura 4.2 mostra os resultados médios para o IPC e para a taxa de faltas dos onze programas disponibilizados pelo CommBench. Com caches maiores que 16Kbytes é reduzida a diferença entre os modelos mais sofisticados que o mapeamento direto. Para caches com 32Kbytes o mapeamento direto tem desempenho 2% pior que a média dos modelos mais sofisticados. Esse desempenho inferior do mapeamento direto é devido ao maior número de faltas por conflito que ocorrem, mesmo que o conjunto de trabalho do programa seja menor que a capacidade da cache. A associatividade exibida pelo modelo $L1_{2w}$ e pelas estruturas auxiliares da PCC e PCVC reduz as faltas por conflito e consequentemente estes possuem um desempenho melhor.

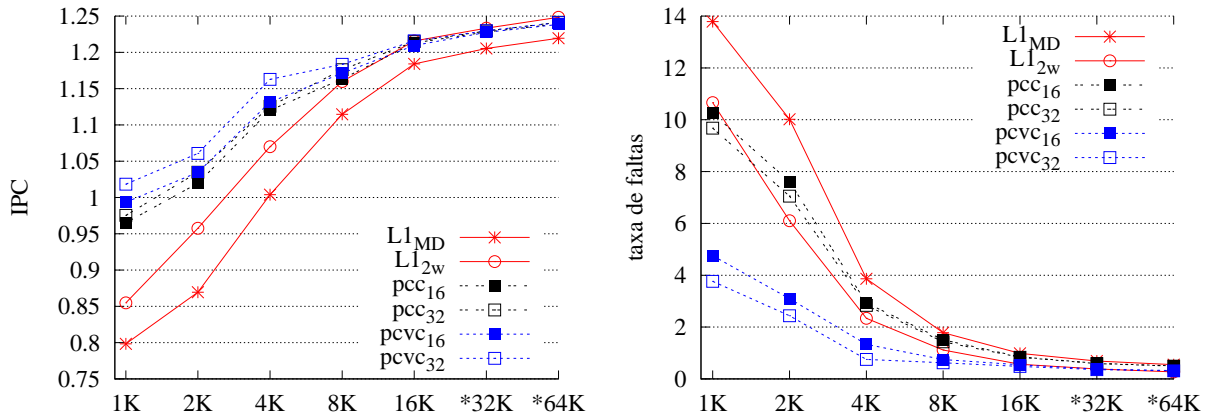


Figura 4.2: (a) Média do IPC vs capacidade, (b) Média da taxa de faltas vs capacidade.

O gráfico de taxa de faltas mostrado na Figura 4.2 apresenta um ponto de inflexão evidente em todos os modelos simulados para a capacidade de 4Kbytes. Os modelos PCC_{32} $PCVC_{32}$ atingem valores muito próximos do ótimo para a taxa de faltas com uma L1 de 4Kbytes. O modelo $L1_{2w}$ supera em desempenho (IPC) a PCC e a PCVC em caches maio-

res que 16Kbytes. O impacto negativo causado pela poluição de blocos pouco referenciados torna-se irrelevante para caches maiores de 16Kbytes devido ao conjunto de trabalho reduzido dos programas do CommBench. Esse comportamento, juntamente com os custos da movimentação de blocos entre as estruturas da PCC e PCVC explica a menor eficiência desses modelos para caches grandes.

4.2 MediaBench

Com o objetivo de ampliar o conjunto de dados para subsidiar a investigação de estruturas como a Cache de Controle de Poluição, foi simulado um conjunto novo de programas. Foram simulados todos programas listados na Seção 2.4 que fazem parte da suíte de testes MediaBench. Como na seção anterior, os dados apresentados são resultantes da execução com um processador de largura 2 e blocos da cache de 32Bytes. Os gráficos da Figura 4.3 comparam a taxa de faltas e o IPC com capacidades da L1 que variam de 1 a 64Kbytes. Os valores apresentados são a média do IPC e taxa de faltas dos programas do MediaBench simulados.

A taxa de faltas para os modelos $L1_{2w}$, PCC e PCVC são próximos entre si para a capacidade de 4Kbytes. Entretanto a análise do IPC para a mesma capacidade demonstra diferenças de desempenho significativas. Os modelos PCC_{32} e $PCVC_{32}$ superam o desempenho da $L1_{2w}$ em respectivamente, 1.97% e 3.82%. Em uma cache primária de 4Kbytes de capacidade, o modelo $PCVC_{32}$ supera a PCC_{32} em 1.82% no IPC. A PCVC é mais simples na sua concepção pois não possui a Cache de Vítimas perfazendo 128Bytes (4 blocos * 32Bytes) a menos que a PCC.

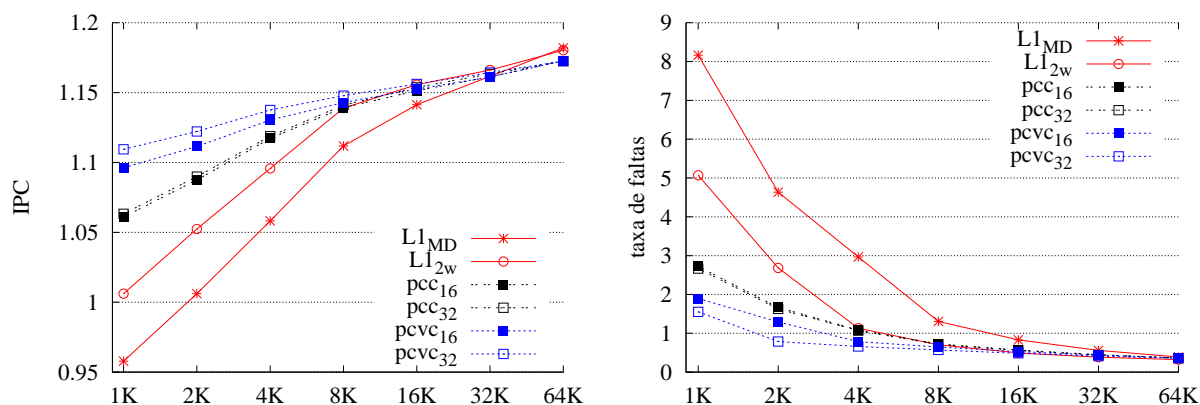


Figura 4.3: (a) Media do IPC vs capacidade, (b) Media da taxa de faltas vs capacidade.

Assim como os resultados do CommBench, para caches maiores que 16Kbytes, os benefícios oriundos da utilização de caches especializadas são decrescentes. Aquelas tem desempenho semelhante aos modelos mais simples, como uma cache com associatividade binária.

Analisando a taxa de faltas apresentadas pelas Figuras 4.2 e 4.3 é possível observar que o aumento do tamanho da cache primaria para capacidades maiores que 4Kbytes não é acompanhado pela redução correspondente na taxa de faltas nos modelos mais sofisticados.

O comportamento observado para as caches de maior capacidade ocorre devido a experimentos baseados em simulações de programas que executam de forma isolada; além de conjuntos de trabalho reduzidos, de ambas as suites de teste simuladas. Em sistemas reais ocorre a execução de mais de um programa; paralelamente aos tratadores de interrupção e drivers que realizam trabalho por um curto período de tempo, referenciando poucas palavras infrequentes. Este cenário impacta diretamente nas memórias cache, causando aumento da concorrência e poluição. Nessas condições esperamos que uma cache como a do ARM de 32Kbytes, para acomodar mais de um conjunto de trabalho, em conjunto com a PCVC, para manter referências infrequentes fora da cache, possua atributos compatíveis com o projeto de sistemas embarcados.

Pode ser acompanhado pelo site [12] o desenvolvimento de uma segunda versão do pacote Mediabench, que possui como principais modificações o uso de maiores conjuntos de dados e atuais programas com relevante participação no mercado. Este pacote não foi utilizado, pois os programas disponibilizados no momento são poucos e concentrados para aplicações de vídeo e processamento de imagens.

4.3 Stream Buffer – Barramento \times Latência

O *Stream Buffer* é muito eficiente na redução de faltas compulsórias, embora o desempenho global do sistema fique reduzido devido a ocupação do barramento ocasionado pela busca antecipada. Alterações no barramento entre memória e cache e na latência da memória têm impacto no comportamento do SB e no desempenho global do sistema. A largura do barramento indica o número de palavras que podem ser transferidas em um ciclo. São simulados sistemas com largura do barramento de 8 e 16 Bytes, respectivamente 2 e 4 palavras. Em um sistema com memória intercalada a latência de memória indica o tempo para acesso à primeira palavra, com um custo de 2 ciclos cada transferência de palavras subsequente. As latências utilizadas são 18 e 12 ciclos. Em todos os casos simulados é utilizado o SB com profundidade de 4 blocos. A Tabela 4.1 associa o número de ciclos necessários para a transferência de blocos com 16 e 32 Bytes, com cada modelo de barramento e latência. A partir deste ponto do texto, o barramento será referenciado como *bus* e latência como *lat*.

Foram realizados testes com o uso do SB com associatividade 2 e 4 . A Figura 4.4 compara a largura do barramento e latência com os parâmetros ‘bloco’ e ‘associatividade’ do SB. O valor apresentado é o IPC das combinações. Apesar dos valores menores para

MODELO / BLOCO	16Bytes	32Bytes
<i>lat18 bus08</i>	20	24
<i>lat18 bus16</i>	18	20
<i>lat12 bus08</i>	14	18
<i>lat12 bus16</i>	12	14

Tabela 4.1: Tempos de tranferência (ciclos) de um bloco entre L1 e memória

lat12 apresentados pela Tabela 4.1, o gráfico mostra que ambos modelos com barramento mais largo (16Bytes) tem melhor desempenho. Esta divergência pode ser explicada por dois fatores: (i) a partir de uma falta na cache e no SB, inicia-se uma sequência de buscas que só termina quando o SB não possui mais blocos livres, e (ii) durante o período em que o barramento esta ocupado com a transferência de palavras não é possível fazer requisições à memória. Assim, para a busca em um SB de 4 blocos de 32 Bytes temos com *lat18 bus16* um total de 32 ciclos ($18 + 7*2$) e com *lat12 bus08* 42 ciclos ($12 + 15*2$).

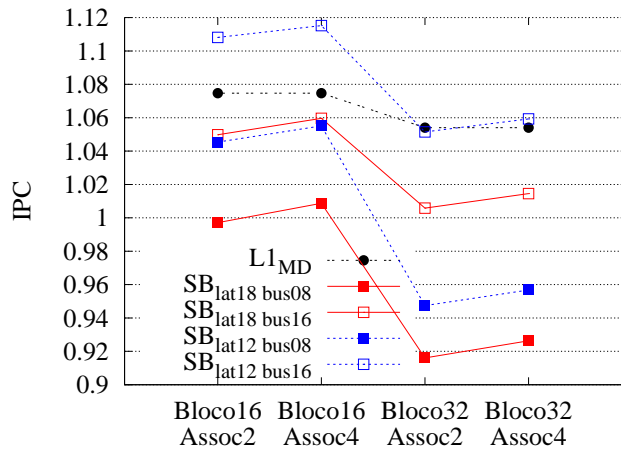


Figura 4.4: Comparativo Bloco x Associatividade do SB.

Tomando qualquer modelo de barramento e latência o desempenho do SB em uma cache com blocos de 32 Bytes é inferior comparado ao desempenho obtido pelo mesmo modelo em uma cache com blocos de 16 Bytes. O modelo *lat18 bus08* possui um custo de 32 ciclos para transferência de 4 blocos de 32 Bytes do SB. Considerando blocos de 16 Bytes o custo de transferência é 16 ciclos. Os custos de 32 e 16 ciclos para os blocos com tamanho respectivamente de 32 e 16 Bytes, significam um IPC 8,1% inferior para a cache com blocos de 32 Bytes no modelo *lat18 bus08*. A diferença de desempenho entre caches com blocos de 16 e 32 Bytes para o modelo *lat18 bus16* é de 4,2%. Por causa da utilização do SB, blocos maiores causam um severo impacto na ocupação do barramento, esse comportamento reflete diretamente no desempenho global do sistema. Pode-se minimizar o efeito que blocos maiores causam com o aumento da largura do barramento. Por exemplo, o IPC do modelo *lat18 bus08* com blocos de 16 Bytes e do modelo *lat18 bus16* com blocos de 32 Bytes são muito próximos, com leve vantagem para blocos de 32 Bytes por causa da localidade espacial. O SB com associatividade 4 causa um pequeno acréscimo no desempenho constante em todos os casos, comparado aos modelos com associatividade 2, porque permite a busca ao longo de mais de uma sequência de palavras.

A Figura 4.5 mostra o desempenho (IPC) dos quatro modelos de barramento e latência para as capacidades de cache que variam de 1 a 64Kbytes. Os valores são referentes a execução do MediaBench em processador de largura 2 com blocos de 16Bytes, a associatividade do SB é 2. Além das variantes do SB, o gráfico apresenta o desempenho dos modelos $L1_{MD}$ e $PCVC_{16}$. O modelo mais sofisticado de SB com *lat12 bus16* possui um bom desempenho, inclusive superando o modelo $PCVC_{16}$ para caches com capacidade maiores que 4Kbytes – em contrapartida *lat12 bus16* é o modelo mais caro. Com exceção do modelo mais simples, com *lat18 bus08*, para caches maiores que 8Kbytes o impacto negativo da elevada ocupação do barramento é minimizado.

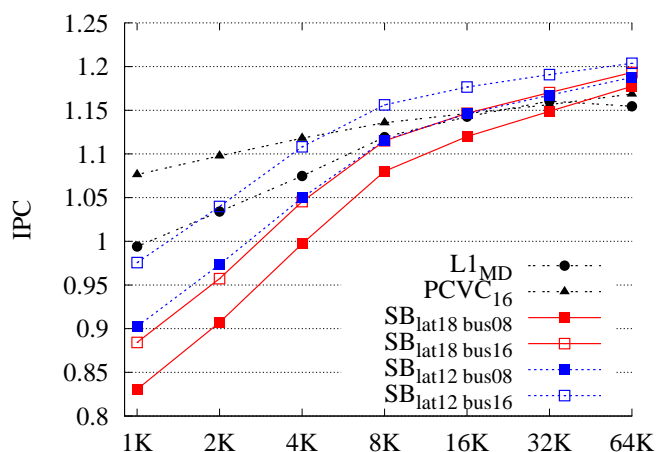


Figura 4.5: CPU larg=2, Bloco=16 Associatividade do SB=2.

Modificações no barramento e na latência da memória causam um significativo impacto no desempenho do SB, chegando a superar a $PCVC_{16}$. A política ativa do SB de buscas antecipadas mostra um bom desempenho com caches grandes, pois nessas caches as faltas por conflito e por capacidade tornam-se raras. Em caches menores que 8Kbytes o desempenho do SB fica prejudicado pela elevada ocupação do barramento provocada pelas faltas por conflito e por capacidade.

Capítulo 5

Conclusão

Investigamos o uso das caches de primeiro nível especializadas para sistemas embarcados, apresentando uma comparação de desempenho entre os modelos, cache de mapeamento direto ($L1_{MD}$), uma cache com associatividade binária ($L1_{2w}$), uma combinação de $L1_{MD}$ com uma cache de controle de poluição (PCC) e uma combinação de $L1_{MD}$ com uma cache de vítimas com controle de poluição (PCVC). Dois conjuntos de programas foram simulados, CommBench agregando simulações para caches de capacidade 32 e 64Kbytes aos resultados apresentados em [7], e MediaBench para aplicações multimídia. Os modelos de PCC e PCVC atingem um desempenho melhor que os outros modelos para a capacidade de 4Kbytes, o ganho de desempenho com o aumento da capacidade é pouco significativo. Para capacidades maiores que 4Kbytes, os modelos mais sofisticados que o mapeamento direto possuem resultados semelhantes. Estes resultados indicam que tanto o MediaBench como o CommBench apresentam na média pequenos conjuntos de trabalho.

Analizamos o impacto de recursos como o barramento entre memória e cache e a latência da memória com o emprego do *Stream Buffer* (SB). Uma barramento mais largo (16Bytes) apresentou melhores resultados e é menos sensível à queda de desempenho causada por blocos maiores (32Bytes). Os modelos de SB com barramento de 16Bytes superou a $PCVC_{16}$ para caches maiores que 8Kbytes porque para estas caches as faltas por conflito e capacidades são muito reduzidas e o SB efetua buscas eficientes.

Experimentos baseados em programas incluídos nos pacotes MediaBench ou CommBench executando isoladamente apresentam bom desempenho em caches pequenas. Em sistemas reais a concorrência entre programas e ferramentas do sistema operacional como os tratadores de interrupção e drivers de dispositivos exercem maior pressão sobre a hierarquia de memória. Esperamos que nesse cenário a utilização de uma cache de 32Kbytes como a cache do ARM seja mais relevante. Pretendemos analisar em trabalho futuro o potencial da PCVC de manter fora da cache referências infrequentes de processos que realizam trabalho por um curto intervalo de tempo, como os tratadores de interrupções.

Bibliografia

- [1] ARM. Advanced RISC Machines Ltd, 2009.
- [2] T. Austin, E. Larson, and D. Ernst. SimpleScalar: An infrastructure for computer system modeling. *Computer*, 35(2):59–67, 2002.
- [3] D Burger and T M Austin. The SimpleScalar Tool Set, Version 2.0. Technical report, University of Wisconsin-Madison and SimpleScalar LLC, 1997.
- [4] D Tennenhouse. Proactive computing. *Communications ACM*, 43(5):43–50, 2000.
- [5] Gnuplot - Open Source command-line driven interactive data and function plotting utility, nov 2008. <http://www.gnuplot.info/>.
- [6] G. C. Heck. Investigação de técnicas de projeto de cache de dados para sistemas embarcados. Dissertação de mestrado, Departamento de Informática, UFPR, Aug. 2008. <http://www.inf.ufpr.br/roberto/dissGiancarlo.pdf>.
- [7] G. C. Heck and R. A. Hexsel. The performance of pollution control victim cache for embedded systems. In *SBCCI '08: Proceedings of the 21st annual symposium on Integrated circuits and system design*, pages 46–51, New York, NY, USA, 2008. ACM.
- [8] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 4th edition, 2006. ISBN 0-12-370490-1.
- [9] N. P. Jouppi. Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. In *ISCA '90: 17th Intl Symp on Computer Architecture*, pages 364–373, 1990.
- [10] D. Kroft. Lockup-free instruction fetch/prefetch cache organization. In *ISCA '81: 8th Annual Symp on Computer Architecture*, pages 81–87, 1981.
- [11] C. Lee, M. Potkonjak, and W. H. Mangione-smith. MediaBench: A tool for evaluating and synthesizing multimedia and communications systems. In *In International Symposium on Microarchitecture*, pages 330–335, 1997.
- [12] MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems, apr 2009. <http://euler.slu.edu/~fritts/mediabench/>.
- [13] PostgreSQL - Open Source Relational Database System, nov 2008. <http://www.postgresql.org/>.
- [14] SimpleScalar LLC, out 2008. <http://www.simplescalar.com/>.
- [15] S. J. Walsh and J. A. Board. Pollution control caching. In *ICCD'95: Intl Conf on Computer Design*, page 300, 1995.

- [16] T. Wolf and M. A. Franklin. CommBench - a telecommunications benchmark for network processors. In *Proc IEEE Intl Symp on Performance Analysis of Systems and Software (ISPASS)*, pages 154–162, Apr. 2000.

Apêndice A

Script Bash

Neste apêndice são listadas os *scripts* Bash que auxiliam a execução dos experimentos.

A execução dos experimentos ocorreu com a assistência de *scripts* Bash. As combinações dos vários parâmetros estipulados gerou um número de 194110 resultados. Os *scripts* foram desenvolvidos para contemplar as seguintes funcionalidades: primeiro, gerar e cobrir todas possibilidades das combinações dos parâmetros; segundo, verificar a integridade e a completude dos arquivos de resultados; e terceiro, gerenciar a execução paralela dos experimentos.

A verificação de integridade e a execução paralela foram implementadas a partir de modificações nos *scrips* disponibilizados pelos autores de [7]. O operador “-s” da linguagem shell testa se um arquivo é vazio. Esse operador é utilizado como condição de disparo de uma simulação. É possível, por exemplo, testar o preenchimento de todos os arquivos executando novamente o script, sem disparar simulações desnecessárias. A execução paralela foi implementada utilizando o operador de execução não bloqueante “&”. Uma variável que armazena o pid dos processos disparados em conjunto com o comando “wait”, permite a criação uma barreira de processos, limitando dessa forma o número de simulações em execução.

A.1 Script Principal

```
#!/bin/bash
#set -x
sfx="_$(date +%d%b%y)"
#
# Benchmark adpcm_dec
#
# Simulador
# Arquivo
modelo=mshr_pcc2
```

```

SS=~/ss
sim=$SS/simuladores/sim-outorder_${modelo}
p_awk=$SS/scripts/gera_query.awk

simulacoes=$SS/simulacoes
cnf=$simulacoes/configs/config_${modelo}.cfg
res=$simulacoes/$(basename $(pwd))/resultados
progr=$(basename $(pwd))
dir=$SS/benchmarks/$progr

fnc=$simulacoes/configs/funcoes_${modelo}.sh
mnc=$simulacoes/configs/mnemonicos.sh
. $fnc
. $mnc

. bench.sh

if [ ! -d $res ] ; then mkdir -p $res ; fi
if [[ $1 = "-limpa" ]] ; then limpa ; fi

for MSHR in 1 2 4 8; do

for PCC in 1 2 4 8 16 32 ; do

# Largura do Processador 1, 2, 4 e 8
for LARG in 1 2 4 8 ; do
case $LARG in
1) RUU="8"   LSQ="4"   ;;
2) RUU="16"  LSQ="8"   ;;
4) RUU="32"  LSQ="16"  ;;
8) RUU="64"  LSQ="32"  ;;
esac

#64K
Waits=
for TEST in 64_64_1 64_32_1 64_16_1 64_08_1 ; do
map $TEST
run
done #FOR TEST
wait $Waits
for TEST in 64_64_1 64_32_1 64_16_1 64_08_1 ; do
if [ ! -s $(actfile).sql ] ; then
awk -f $p_awk $(actfile).out > $(actfile).sql &
fi
done #FOR awk

done #FOR LARG
done #FOR PCC
done #FOR MSHR

#sleep para terminar AWK
sleep 3s;
backup;

```

A.2 Funções de execução

```
#!/bin/bash
#set -x
sfx="_$(date +%d%b%y)"
#
# Arquivo funcoes_pcc2.sh
# Funcoes Modelo: PCVC
# Execução do teste em background.
# Mapeamento de características: Tamanho, Bloco, Associatividade.

function run() {
echo -e "\n $(basename $(actfile)) \n"
if [ ! -s $(actfile).out ] ; then
bash -c "$sim -config $cnf \
        -cache:d1l $c1d \
        -cache:il1 $c1i \
        -ruu:size ${RUU} \
        -lsq:size ${LSQ} \
        -fetch:ifqsize ${LARG} \
        -decode:width ${LARG} \
        -issue:width ${LARG} \
        -commit:width ${LARG} \
        -res:ialu ${LARG} \
        -res:fpalu ${LARG} \
        -redir:sim $(actfile).out \
        $BENCH_EXE " &
Waits="$Waits $!"
fi
}

function map() {
case $1 in
00_00_1) c1d="none" c1i="il1:4096:8:1:1" ;;
64_64_1) c1d="d1l:1024:64:1:1" c1i="il1:512:64:1:1" ;;
64_32_1) c1d="d1l:2048:32:1:1" c1i="il1:1024:32:1:1" ;;
64_16_1) c1d="d1l:4096:16:1:1" c1i="il1:2048:16:1:1" ;;
64_08_1) c1d="d1l:8192:8:1:1" c1i="il1:4096:8:1:1" ;;
64_64_2) c1d="d1l:512:64:2:1" c1i="il1:512:64:1:1" ;;
64_32_2) c1d="d1l:1024:32:2:1" c1i="il1:1024:32:1:1" ;;
64_16_2) c1d="d1l:2048:16:2:1" c1i="il1:2048:16:1:1" ;;
64_08_2) c1d="d1l:4096:8:2:1" c1i="il1:4096:8:1:1" ;;
64_64_4) c1d="d1l:256:64:4:1" c1i="il1:512:64:1:1" ;;
64_32_4) c1d="d1l:512:32:4:1" c1i="il1:1024:32:1:1" ;;
64_16_4) c1d="d1l:1024:16:4:1" c1i="il1:2048:16:1:1" ;;
64_08_4) c1d="d1l:2048:8:4:1" c1i="il1:4096:8:1:1" ;;
64_64_8) c1d="d1l:128:64:8:1" c1i="il1:512:64:1:1" ;;
64_32_8) c1d="d1l:256:32:8:1" c1i="il1:1024:32:1:1" ;;
64_16_8) c1d="d1l:512:16:8:1" c1i="il1:2048:16:1:1" ;;
64_08_8) c1d="d1l:1024:8:8:1" c1i="il1:4096:8:1:1" ;;

*) echo -e "\n\n\n parametro errado $TEST \n\n\n" ; exit 1;
esac
}
```

A.3 Funções de apoio

```
#!/bin/bash
#set -x
sfx="_$(date +%d%b%y)"
#
#Arquivo mnemonicos.sh
#Script de definições de nomes, funcoes de backup e limpeza.

function actfile() {
case $modelo in
base) echo "$res/${TEST}_${modelo}_LARG-${LARG}" ;;
mshr) echo "$res/${TEST}_${modelo}-${MSHR}_LARG-${LARG}" ;;
mshr_cv) echo "$res/${TEST}_${modelo}-${MSHR}_CV-${CV}_LARG-${LARG}" ;;
mshr_sb) echo "$res/${TEST}_${modelo}-${MSHR}_SB-${SB}_ASSOC-${ASSOC}_LARG-${LARG}" ;;
mshr_sblat) echo "$res/${TEST}_${modelo}-${MSHR}_SB-${SB}_ASSOC-${ASSOC}_LARG-${LARG}_BARR-${BARR}_L"
mshr_pcc1) echo "$res/${TEST}_${modelo}-${MSHR}_PCC-${PCC}_CV-${CV}_LARG-${LARG}" ;;
mshr_pcc2) echo "$res/${TEST}_${modelo}-${MSHR}_PCC-${PCC}_LARG-${LARG}" ;;
*) echo -e "\n\n\n modelo: $modelo, de simulador errado\n\n\n" ; exit 1;
esac
}

function limpa() {
if [ $modelo != base ]
then
rm -rf $res/*_${modelo}-*
rm -rf $res/${modelo}-outs.tar.gz
rm -rf $res/${modelo}-sqls.tar.gz
else
rm -rf $res/*_${modelo}_*
rm -rf $res/${modelo}-outs.tar.gz
rm -rf $res/${modelo}-sqls.tar.gz
fi
}

function backup() {
if [ $modelo != base ]
then
cd $res
cat *_${modelo}-*.sql > insert_${modelo}.sql
tar -zcvf ${modelo}-outs.tar.gz *_${modelo}-*.out
tar -zcvf ${modelo}-sqls.tar.gz *_${modelo}-*.sql
else
cd $res
cat *_${modelo}_*.sql > insert_${modelo}.sql
tar -zcvf ${modelo}-outs.tar.gz *_${modelo}_*.out
tar -zcvf ${modelo}-sqls.tar.gz *_${modelo}_*.sql
fi
}
```

A.4 Linha de comando do Benchmark

```
#!/bin/bash
#set -x
# Arquivo bench.sh

BENCH_BIN=$dir/$progr
BENCH_OPT=$dir/words
BENCH_INP=$dir/${progr}.data
BENCH_OUT="/dev/null"
BENCH_EXE=" ${BENCH_BIN} < ${BENCH_INP} > ${BENCH_OUT} "
```


Apêndice B

Script para criação da base de dados

Devido a grande quantidade de arquivos com resultados provenientes das simulações é conveniente o uso de uma base de dados alocada no sistema gerenciador de base de dados PostgreSQL [13]. Através de consultas em linguagem SQL é possível apresentar os dados de forma a facilitar a análise dos resultados e a geração de gráficos pelo programa Gnuplot [5].

Com o *script* SQL disponibilizado pelos autores de [7], é possível definir e criar uma base de dados para acomodar os resultados das simulações.

```
-- Arquivo: cria_db.sql
--- Database: ss
-- DROP DATABASE ss;
CREATE DATABASE ss
  WITH OWNER = postgres
       ENCODING = 'LATIN1'
       TABLESPACE = pg_default;
GRANT ALL ON DATABASE ss TO public;
GRANT ALL ON DATABASE ss TO postgres;
-- Table: benchmarks
-- DROP TABLE benchmarks;

CREATE TABLE benchmarks
(
  bench_id smallint NOT NULL, -- Identificador do programa de teste.
  bench_nome character varying(100), -- Nome do programa de teste.
  bench_tx_descricao character varying(200), -- Descrição do programa de teste.
  CONSTRAINT pk_benchmarks PRIMARY KEY (bench_id)
)
WITHOUT OIDS;
--ALTER TABLE benchmarks OWNER TO postgres;
GRANT ALL ON TABLE benchmarks TO postgres;
GRANT ALL ON TABLE benchmarks TO public;
COMMENT ON TABLE benchmarks IS 'Tabela dos programas de teste.';
```

```
COMMENT ON COLUMN benchmarks.bench_id IS 'Identificador do programa de teste.';
COMMENT ON COLUMN benchmarks.bench_nome IS 'Nome do programa de teste.';
COMMENT ON COLUMN benchmarks.bench_tx_descricao IS 'Descrição do programa de teste.';
```

```
INSERT INTO benchmarks VALUES (1,'anagram');
INSERT INTO benchmarks VALUES (2,'bubble');
INSERT INTO benchmarks VALUES (3,'insertion');
INSERT INTO benchmarks VALUES (4,'merge');
INSERT INTO benchmarks VALUES (5,'quick');
INSERT INTO benchmarks VALUES (6,'selection');
INSERT INTO benchmarks VALUES (7,'shell');
INSERT INTO benchmarks VALUES (8,'cast_dec');
INSERT INTO benchmarks VALUES (9,'cast_enc');
INSERT INTO benchmarks VALUES (10,'drr');
INSERT INTO benchmarks VALUES (11,'frag');
INSERT INTO benchmarks VALUES (12,'jpeg_dec');
INSERT INTO benchmarks VALUES (13,'jpeg_enc');
INSERT INTO benchmarks VALUES (14,'reed_dec');
INSERT INTO benchmarks VALUES (15,'reed_enc');
INSERT INTO benchmarks VALUES (16,'rtr');
```

```
INSERT INTO benchmarks VALUES (17,'adpcm_dec');
INSERT INTO benchmarks VALUES (18,'adpcm_enc');
INSERT INTO benchmarks VALUES (19,'epic_dec');
INSERT INTO benchmarks VALUES (20,'epic_enc');
INSERT INTO benchmarks VALUES (21,'g721_dec');
INSERT INTO benchmarks VALUES (22,'g721_enc');
INSERT INTO benchmarks VALUES (23,'jpeg_mdec');
INSERT INTO benchmarks VALUES (24,'jpeg_menc');
INSERT INTO benchmarks VALUES (25,'mesa_mipmap');
INSERT INTO benchmarks VALUES (26,'mesa_osdemo');
INSERT INTO benchmarks VALUES (27,'mesa_osdemo');
INSERT INTO benchmarks VALUES (28,'mpeg2_dec');
INSERT INTO benchmarks VALUES (29,'mpeg2_enc');
INSERT INTO benchmarks VALUES (30,'pegwit_dec');
INSERT INTO benchmarks VALUES (31,'pegwit_enc');
INSERT INTO benchmarks VALUES (32,'toast');
INSERT INTO benchmarks VALUES (33,'untoast');
INSERT INTO benchmarks VALUES (34,'zip_dec');
INSERT INTO benchmarks VALUES (35,'zip_enc');
```

```
-- Table: cache_types
```

```
-- DROP TABLE cache_types;
```

```
CREATE TABLE cache_types
```

```
(
  caty_id smallint NOT NULL, -- Identificador do tipo de cache.
  caty_nm_cache character varying(100) NOT NULL, -- Nome do tipo de cache.
  caty_tx_descricao character varying(100), -- Descrição do nome da cache.
  CONSTRAINT pk_cache_types PRIMARY KEY (caty_id)
)
```

```
WITHOUT OIDS;
```

```
--ALTER TABLE cache_types OWNER TO postgres;
```

```
GRANT ALL ON TABLE cache_types TO postgres;
```

```
GRANT ALL ON TABLE cache_types TO public;
```

```

COMMENT ON TABLE cache_types IS 'Tabela dos tipos de memória cache.';
COMMENT ON COLUMN cache_types.caty_id IS 'Identificador do tipo de cache.';
COMMENT ON COLUMN cache_types.caty_nm_cache IS 'Nome do tipo de cache.';
COMMENT ON COLUMN cache_types.caty_tx_descricao IS 'Descrição do nome da cache.';

Insert Into cache_types values (1,'base','Modelo básico');
Insert Into cache_types values (2,'cv','Cache de Víctima');
Insert Into cache_types values (3,'sb','Stream Buffer');
Insert Into cache_types values (4,'mshr','Miss Status Holding Register');
Insert Into cache_types values (5,'mshr_cv','MSHR + CV');
Insert Into cache_types values (6,'mshr_sb','MSHR + SB');
Insert Into cache_types values (7,'mshr_ce1','MSHR + Cache de Escrita 1');
Insert Into cache_types values (8,'mshr_ce2','MSHR + Cache de Escrita 2');
Insert Into cache_types values (9,'mshr_pcc1','MSHR + Cache de Controle de Poluição + CV');
Insert Into cache_types values (10,'mshr_pcc2','MSHR + Cache de Controle de Poluição');
Insert Into cache_types values (11,'ncb','Buffer de Não-Críticos');
Insert Into cache_types values (12,'nts','Cache de Fluxo Não-Temporal');
Insert Into cache_types values (13,'lbic','Cache Intercalada Baseada na Localidade');

-- Table: sim_config

-- DROP TABLE sim_config;

CREATE TABLE sim_config
(
    sico_id smallint NOT NULL, -- Identificador da configuração do simulador.
    sico_tx_cfg character varying(200), -- Descrição da configuração do simulador.
    sico_nr_largura smallint NOT NULL, -- Largura do processador.
    sico_nr_lsq smallint NOT NULL, -- Número de registros na LSQ.
    sico_nr_ruu smallint NOT NULL, -- Número de registros no RUU.
    sico_mem_lat character varying(10), -- Latência da memória principal.
    sico_mem_width smallint NOT NULL, -- Largura do barramento de memória.
    sico_nr_ialu smallint NOT NULL, -- Número de ALUs de inteiros.
    sico_nr_fpalu smallint NOT NULL, -- Número de ALUs de ponto flutuante.
    CONSTRAINT pk_sim_config PRIMARY KEY (sico_id)
)
WITHOUT OIDS;
--ALTER TABLE sim_config OWNER TO postgres;
GRANT ALL ON TABLE sim_config TO postgres;
GRANT ALL ON TABLE sim_config TO public;
COMMENT ON TABLE sim_config IS 'Tabela de configurações do simulador.';
COMMENT ON COLUMN sim_config.sico_id IS 'Identificador da configuração do simulador.';
COMMENT ON COLUMN sim_config.sico_tx_cfg IS 'Descrição da configuração do simulador.';
COMMENT ON COLUMN sim_config.sico_nr_largura IS 'Largura do processador.';
COMMENT ON COLUMN sim_config.sico_nr_lsq IS 'Número de registros na LSQ.';
COMMENT ON COLUMN sim_config.sico_nr_ruu IS 'Número de registros no RUU.';
COMMENT ON COLUMN sim_config.sico_mem_lat IS 'Latência da memória principal.';
COMMENT ON COLUMN sim_config.sico_mem_width IS 'Largura do barramento de memória.';
COMMENT ON COLUMN sim_config.sico_nr_ialu IS 'Número de ALUs de inteiros.';
COMMENT ON COLUMN sim_config.sico_nr_fpalu IS 'Número de ALUs de ponto flutuante.';

INSERT INTO sim_config VALUES (1, 'ruu16', 2, 8, 16, '18_2', 8, 2, 2);

```

```

INSERT INTO sim_config VALUES (2, 'ruu32', 4, 16, 32, '18_2', 8, 4, 4);
INSERT INTO sim_config VALUES (3, 'ruu64', 8, 32, 64, '18_2', 8, 8, 8);
INSERT INTO sim_config VALUES (4, 'ruu8', 1, 4, 8, '18_2', 8, 1, 1);

```

```
-- Table: li_config
```

```
-- DROP TABLE li_config;
```

```
CREATE TABLE li_config
```

```
(
  lico_id smallint NOT NULL, -- Identificador da configuração da cache L1.
  lico_tx_cfg character varying(200), -- Descrição da configuração da cache L1.
  lico_tm_cache smallint NOT NULL, -- Tamanho da cache.
  lico_tm_set smallint NOT NULL, -- Tamaho do conjunto.
  lico_tm_bloco smallint NOT NULL, -- Tamanho do bloco.
  lico_nr_assoc smallint NOT NULL DEFAULT 0, -- Número da associatividade da cache.
  CONSTRAINT pk_li_config PRIMARY KEY (lico_id, lico_tm_cache, lico_tm_set, lico_tm_bloco, lico_nr_a
)
```

```
WITHOUT OIDS;
```

```
--ALTER TABLE li_config OWNER TO postgres;
```

```
GRANT ALL ON TABLE li_config TO postgres;
```

```
GRANT ALL ON TABLE li_config TO public;
```

```
COMMENT ON TABLE li_config IS 'Tabela de configurações da cache L1.';
```

```
COMMENT ON COLUMN li_config.lico_id IS 'Identificador da configuração da cache L1.';
```

```
COMMENT ON COLUMN li_config.lico_tx_cfg IS 'Descrição da configuração da cache L1.';
```

```
COMMENT ON COLUMN li_config.lico_tm_cache IS 'Tamanho da cache L1.';
```

```
COMMENT ON COLUMN li_config.lico_tm_set IS 'Tamanho do conjunto.';
```

```
COMMENT ON COLUMN li_config.lico_tm_bloco IS 'Tamanho do bloco.';
```

```
COMMENT ON COLUMN li_config.lico_nr_assoc IS 'Número da associatividade da cache L1.';
```

```

INSERT INTO li_config values (1, '00_00_1', 0, 0, 0, 1);
INSERT INTO li_config values (2, '01_08_1', 1, 128, 8, 1);
INSERT INTO li_config values (3, '01_16_1', 1, 64, 16, 1);
INSERT INTO li_config values (4, '01_32_1', 1, 32, 32, 1);
INSERT INTO li_config values (5, '01_64_1', 1, 16, 64, 1);
INSERT INTO li_config values (6, '02_08_1', 2, 256, 8, 1);
INSERT INTO li_config values (7, '02_16_1', 2, 128, 16, 1);
INSERT INTO li_config values (8, '02_32_1', 2, 64, 32, 1);
INSERT INTO li_config values (9, '02_64_1', 2, 32, 64, 1);
INSERT INTO li_config values (10, '04_08_1', 4, 512, 8, 1);
INSERT INTO li_config values (11, '04_16_1', 4, 256, 16, 1);
INSERT INTO li_config values (12, '04_32_1', 4, 128, 32, 1);
INSERT INTO li_config values (13, '04_64_1', 4, 64, 64, 1);
INSERT INTO li_config values (14, '08_08_1', 8, 1024, 8, 1);
INSERT INTO li_config values (15, '08_16_1', 8, 512, 16, 1);
INSERT INTO li_config values (16, '08_32_1', 8, 256, 32, 1);
INSERT INTO li_config values (17, '08_64_1', 8, 128, 64, 1);
INSERT INTO li_config values (18, '16_08_1', 16, 2048, 8, 1);
INSERT INTO li_config values (19, '16_16_1', 16, 1024, 16, 1);
INSERT INTO li_config values (20, '16_32_1', 16, 512, 32, 1);
INSERT INTO li_config values (21, '16_64_1', 16, 256, 64, 1);
INSERT INTO li_config values (22, '32_08_1', 32, 4096, 8, 1);
INSERT INTO li_config values (23, '32_16_1', 32, 2048, 16, 1);
INSERT INTO li_config values (24, '32_32_1', 32, 1024, 32, 1);

```

```

INSERT INTO li_config values (25, '32_64_1', 32, 512, 64, 1);
INSERT INTO li_config values (26, '64_08_1', 64, 8192, 8, 1);
INSERT INTO li_config values (27, '64_16_1', 64, 4096, 16, 1);
INSERT INTO li_config values (28, '64_32_1', 64, 2048, 32, 1);
INSERT INTO li_config values (29, '64_64_1', 64, 1024, 64, 1);
--Associatividade 2
INSERT INTO li_config values (30, '01_16_2', 1, 32, 16, 2);
INSERT INTO li_config values (31, '01_32_2', 1, 16, 32, 2);
INSERT INTO li_config values (32, '02_16_2', 2, 64, 16, 2);
INSERT INTO li_config values (33, '02_32_2', 2, 32, 32, 2);
INSERT INTO li_config values (34, '04_16_2', 4, 128, 16, 2);
INSERT INTO li_config values (35, '04_32_2', 4, 64, 32, 2);
INSERT INTO li_config values (36, '08_16_2', 8, 256, 16, 2);
INSERT INTO li_config values (37, '08_32_2', 8, 128, 32, 2);
INSERT INTO li_config values (38, '16_16_2', 16, 512, 16, 2);
INSERT INTO li_config values (39, '16_32_2', 16, 256, 32, 2);
INSERT INTO li_config values (40, '32_16_2', 32, 1024, 16, 2);
INSERT INTO li_config values (41, '32_32_2', 32, 512, 32, 2);
INSERT INTO li_config values (42, '64_16_2', 64, 2048, 16, 2);
INSERT INTO li_config values (43, '64_32_2', 64, 1024, 32, 2);

```

```
-- Table: sim_statistics
```

```
-- ALTER TABLE sim_statistics DROP CONSTRAINT pk_sim_statistics;
-- DROP TABLE sim_statistics;
```

```
CREATE TABLE sim_statistics
```

```
(
```

```
-- Campos de índice
```

```

sim_id_sico smallint NOT NULL, -- Identificador da configuração do simulador.
sim_id_bench smallint NOT NULL, -- Identificador do programa de teste.
sim_id_caty smallint NOT NULL, -- Identificador do tipo de cache.
sim_id_lico smallint NOT NULL, -- Identificador da configuração da cache L1.
sim_nr_mshr smallint NOT NULL DEFAULT 0, -- Número de registros no MSHR.
sim_nr_cv smallint NOT NULL DEFAULT 0, -- Número de registros na CV.
sim_nr_sb smallint NOT NULL DEFAULT 0, -- Número de registros no SB.
sim_assoc_sb smallint NOT NULL DEFAULT 0, -- Número da associatividade do SB.
sim_is_ce smallint NOT NULL DEFAULT 0, -- Indicador de uso da CE.
sim_nr_pcc smallint NOT NULL DEFAULT 0, -- Número de registros na PCC.

```

```
-- Campos de resultado de simulação
```

```

sim_num_insn bigint, -- total number of instructions committed
sim_num_refs bigint, -- total number of loads and stores committed
sim_num_loads bigint, -- total number of loads committed
sim_num_stores bigint, -- total number of stores committed
sim_num_branches bigint, -- total number of branches committed
sim_elapsed_time integer, -- total simulation time in seconds
sim_inst_rate double precision, -- simulation speed (in insts/sec)
sim_total_insn bigint, -- total number of instructions executed

```

```

sim_total_refs bigint, -- total number of loads and stores executed
sim_total_loads bigint, -- total number of loads executed
sim_total_stores bigint, -- total number of stores executed
sim_total_branches bigint, -- total number of branches executed
sim_cycle bigint, -- total simulation time in cycles
"sim_IPC" real, -- instructions per cycle
"sim_CPI" real, -- cycles per instruction
"sim_exec_BW" real, -- total instructions (mis-spec + committed) per cycle
"sim_IPB" real, -- instruction per branch
"IFQ_count" bigint, -- cumulative IFQ occupancy
"IFQ_fcount" bigint, -- cumulative IFQ full count
ifq_occupancy real, -- avg IFQ occupancy (insn's)
ifq_rate real, -- avg IFQ dispatch rate (insn/cycle)
ifq_latency real, -- avg IFQ occupant latency (cycle's)
ifq_full real, -- fraction of time (cycle's) IFQ was full
"RUU_count" bigint, -- cumulative RUU occupancy
"RUU_fcount" bigint, -- cumulative RUU full count
ruu_occupancy real, -- avg RUU occupancy (insn's)
ruu_rate real, -- avg RUU dispatch rate (insn/cycle)
ruu_latency real, -- avg RUU occupant latency (cycle's)
ruu_full real, -- fraction of time (cycle's) RUU was full
"LSQ_count" bigint, -- cumulative LSQ occupancy
"LSQ_fcount" bigint, -- cumulative LSQ full count
lsq_occupancy real, -- avg LSQ occupancy (insn's)
lsq_rate real, -- avg LSQ dispatch rate (insn/cycle)
lsq_latency real, -- avg LSQ occupant latency (cycle's)
lsq_full real, -- fraction of time (cycle's) LSQ was full
sim_slip bigint, -- total number of slip cycles
avg_sim_slip real, -- the average slip between issue and retirement
bpred_bimod_lookups bigint, -- total number of bpred lookups
bpred_bimod_updates bigint, -- total number of updates
bpred_bimod_addr_hits bigint, -- total number of address-predicted hits
bpred_bimod_dir_hits bigint, -- total number of direction-predicted hits (includes addr-hits)
bpred_bimod_misses bigint, -- total number of misses
bpred_bimod_jr_hits bigint, -- total number of address-predicted hits for JR's
bpred_bimod_jr_seen bigint, -- total number of JR's seen
"bpred_bimod_jr_non_ras_hits_PP" bigint, -- total number of address-predicted hits for non-RAS JR's
"bpred_bimod_jr_non_ras_seen_PP" bigint, -- total number of non-RAS JR's seen
bpred_bimod_bpred_addr_rate real, -- branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod_bpred_dir_rate real, -- branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod_bpred_jr_rate real, -- JR address-prediction rate (i.e., JR addr-hits/JRs seen)
"bpred_bimod_bpred_jr_non_ras_rate_PP" real, -- non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs)
bpred_bimod_retstack_pushes bigint, -- total number of address pushed onto ret-addr stack
bpred_bimod_retstack_pops bigint, -- total number of address popped off of ret-addr stack
"bpred_bimod_used_ras_PP" bigint, -- total number of RAS predictions used
"bpred_bimod_ras_hits_PP" bigint, -- total number of RAS hits
"bpred_bimod_ras_rate_PP" real, -- RAS prediction rate (i.e., RAS hits/used RAS)

ill_accesses bigint, -- total number of accesses
ill_hits bigint, -- total number of hits
ill_misses bigint, -- total number of misses
ill_replacements bigint, -- total number of replacements
ill_writebacks bigint, -- total number of writebacks
ill_invalidations bigint, -- total number of invalidations
ill_miss_rate real, -- miss rate (i.e., misses/ref)

```

```

il1_repl_rate real, -- replacement rate (i.e., repls/ref)
il1_wb_rate real, -- writeback rate (i.e., wrbks/ref)
il1_inv_rate real, -- invalidation rate (i.e., invs/ref)

dl1_accesses bigint, -- total number of accesses
dl1_hits bigint, -- total number of hits
dl1_miss_hit_mshr bigint, -- total number of miss hit in mshrs
dl1_misses bigint, -- total number of misses
dl1_replacements bigint, -- total number of replacements
dl1_writebacks bigint, -- total number of writebacks
dl1_invalidations bigint, -- total number of invalidations
dl1_miss_rate real, -- miss rate (i.e., misses/ref)
dl1_repl_rate real, -- replacement rate (i.e., repls/ref)
dl1_wb_rate real, -- writeback rate (i.e., wrbks/ref)
dl1_inv_rate real, -- invalidation rate (i.e., invs/ref)

dl2_accesses bigint, -- total number of accesses
dl2_hits bigint, -- total number of hits
dl2_miss_hit_mshr bigint, -- total number of miss hit in mshrs
dl2_misses bigint, -- total number of misses
dl2_replacements bigint, -- total number of replacements
dl2_writebacks bigint, -- total number of writebacks
dl2_invalidations bigint, -- total number of invalidations
dl2_miss_rate real, -- miss rate (i.e., misses/ref)
dl2_repl_rate real, -- replacement rate (i.e., repls/ref)
dl2_wb_rate real, -- writeback rate (i.e., wrbks/ref)
dl2_inv_rate real, -- invalidation rate (i.e., invs/ref)

mshr_accesses bigint, -- total number of accesses
mshr_miss_hits bigint, -- total number of miss hits
mshr_misses bigint, -- total number of misses
mshr_full bigint, -- number of accesses in mshr full
mshr_hits_false bigint, -- number of hits falses

dcv_accesses bigint, -- total number of accesses
dcv_hits bigint, -- total number of hits
dcv_misses bigint, -- total number of misses
dcv_replacements bigint, -- total number of replacements
dcv_writebacks bigint, -- total number of writebacks
dcv_invalidations bigint, -- total number of invalidations
dcv_miss_rate real, -- miss rate (i.e., misses/ref)
dcv_repl_rate real, -- replacement rate (i.e., repls/ref)
dcv_wb_rate real, -- writeback rate (i.e., wrbks/ref)
dcv_inv_rate real, -- invalidation rate (i.e., invs/ref)

dsb_accesses bigint, -- total number of accesses
dsb_hits bigint, -- total number of hits
dsb_misses bigint, -- total number of misses
dsb_replacements bigint, -- total number of replacements
dsb_miss_rate real, -- miss rate (i.e., misses/ref)
dsb_repl_rate real, -- replacement rate (i.e., repls/ref)
dsb_bus_occupation bigint, -- total cycles of bus_occupation
dsb_reg1_hits bigint, -- total number of hits in registrators
dsb_reg2_hits bigint, -- total number of hits in registrators
dsb_reg3_hits bigint, -- total number of hits in registrators

```

```

dsb_reg4_hits bigint, -- total number of hits in registrators
dsb_reg5_hits bigint, -- total number of hits in registrators
dsb_reg6_hits bigint, -- total number of hits in registrators
dsb_reg7_hits bigint, -- total number of hits in registrators
dsb_reg8_hits bigint, -- total number of hits in registrators
dsb_reg9_hits bigint, -- total number of hits in registrators
dsb_reg10_hits bigint, -- total number of hits in registrators
dsb_reg11_hits bigint, -- total number of hits in registrators
dsb_reg12_hits bigint, -- total number of hits in registrators
dsb_reg13_hits bigint, -- total number of hits in registrators
dsb_reg14_hits bigint, -- total number of hits in registrators
dsb_reg15_hits bigint, -- total number of hits in registrators
dsb_reg16_hits bigint, -- total number of hits in registrators
dsb_reg17_hits bigint, -- total number of hits in registrators

dpcc_accesses bigint, -- total number of accesses
dpcc_hits bigint, -- total number of hits
dpcc_miss_hit_mshr bigint, -- total number of miss hit in mshrs
dpcc_misses bigint, -- total number of misses
dpcc_replacements bigint, -- total number of replacements
dpcc_writebacks bigint, -- total number of writebacks
dpcc_invalidations bigint, -- total number of invalidations
dpcc_miss_rate real, -- miss rate (i.e., misses/ref)
dpcc_repl_rate real, -- replacement rate (i.e., repls/ref)
dpcc_wb_rate real, -- writeback rate (i.e., wrbks/ref)
dpcc_inv_rate real, -- invalidation rate (i.e., invs/ref)

ce_accesses bigint, -- total number of accesses
ce_hits bigint, -- total number of hits
ce_misses bigint, -- total number of misses
ce_replacements bigint, -- total number of replacements
ce_miss_rate real, -- miss rate (i.e., misses/ref)
ce_repl_rate real, -- replacement rate (i.e., repls/ref)
ce_reg1_hits bigint, -- total number of hits in registrators
ce_reg2_hits bigint, -- total number of hits in registrators
ce_reg3_hits bigint, -- total number of hits in registrators
ce_reg4_hits bigint, -- total number of hits in registrators
ce_reg5_hits bigint, -- total number of hits in registrators
ce_reg6_hits bigint, -- total number of hits in registrators
ce_reg7_hits bigint, -- total number of hits in registrators
ce_reg8_hits bigint, -- total number of hits in registrators
ce_reg9_hits bigint, -- total number of hits in registrators

itlb_accesses bigint, -- total number of accesses
itlb_hits bigint, -- total number of hits
itlb_misses bigint, -- total number of misses
itlb_replacements bigint, -- total number of replacements
itlb_writebacks bigint, -- total number of writebacks
itlb_invalidations bigint, -- total number of invalidations
itlb_miss_rate real, -- miss rate (i.e., misses/ref)
itlb_repl_rate real, -- replacement rate (i.e., repls/ref)
itlb_wb_rate real, -- writeback rate (i.e., wrbks/ref)
itlb_inv_rate real, -- invalidation rate (i.e., invs/ref)
dtlb_accesses bigint, -- total number of accesses
dtlb_hits bigint, -- total number of hits

```



```

dtlb_misses bigint, -- total number of misses
dtlb_replacements bigint, -- total number of replacements
dtlb_writebacks bigint, -- total number of writebacks
dtlb_invalidations bigint, -- total number of invalidations
dtlb_miss_rate real, -- miss rate (i.e., misses/ref)
dtlb_repl_rate real, -- replacement rate (i.e., repls/ref)
dtlb_wb_rate real, -- writeback rate (i.e., wrbks/ref)
dtlb_inv_rate real, -- invalidation rate (i.e., invs/ref)
sim_invalid_addrs bigint, -- total non-speculative bogus addresses seen (debug var)
ld_text_base character varying(20), -- program text (code) segment base
ld_text_size bigint, -- program text (code) size in bytes
ld_data_base character varying(20), -- program initialized data segment base
ld_data_size bigint, -- program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base character varying(20), -- program stack segment base (highest address in stack)
ld_stack_size bigint, -- program initial stack size
ld_prog_entry character varying(20), -- program entry point (initial PC)
ld_envIRON_base character varying(20), -- program environment base address address
ld_target_big_endian bigint, -- target executable endian-ness, non-zero if big endian
mem_page_count bigint, -- total number of pages allocated
mem_page_mem character varying(20), -- total size of memory pages allocated
mem_ptab_misses bigint, -- total first level page table misses
mem_ptab_accesses bigint, -- total page table accesses
mem_ptab_miss_rate real, -- first level page table miss rate
CONSTRAINT pk_sim_statistics PRIMARY KEY(sim_id_sico, sim_id_bench, sim_id_caty, sim_id_lico, sim_nr
)
WITHOUT OIDS;

--ALTER TABLE sim_statistics OWNER TO postgres;
GRANT ALL ON TABLE sim_statistics TO postgres;
GRANT ALL ON TABLE sim_statistics TO public;

COMMENT ON TABLE sim_statistics IS 'Tabela com os resultados das simulações do SS.';
COMMENT ON COLUMN sim_statistics.sim_id_sico IS 'Identificador da configuração do simulador.';
COMMENT ON COLUMN sim_statistics.sim_id_bench IS 'Identificador do programa de teste.';
COMMENT ON COLUMN sim_statistics.sim_id_caty IS 'Identificador do tipo de cache.';
COMMENT ON COLUMN sim_statistics.sim_id_lico IS 'Identificador da configuração da cache L1.';
COMMENT ON COLUMN sim_statistics.sim_nr_mshr IS 'Número de registros no MSHR.';
COMMENT ON COLUMN sim_statistics.sim_nr_cv IS 'Número de registros na CV.';
COMMENT ON COLUMN sim_statistics.sim_nr_sb IS 'Número de registros no SB.';
COMMENT ON COLUMN sim_statistics.sim_assoc_sb IS 'Número da associatividade do SB.';
COMMENT ON COLUMN sim_statistics.sim_is_ce IS 'Indicador de uso da CE.';
COMMENT ON COLUMN sim_statistics.sim_nr_pcc IS 'Número de registros na PCC.';

COMMENT ON COLUMN sim_statistics.sim_num_insn IS 'total number of instructions committed';
COMMENT ON COLUMN sim_statistics.sim_num_refs IS 'total number of loads and stores committed';
COMMENT ON COLUMN sim_statistics.sim_num_loads IS 'total number of loads committed';
COMMENT ON COLUMN sim_statistics.sim_num_stores IS 'total number of stores committed';
COMMENT ON COLUMN sim_statistics.sim_num_branches IS 'total number of branches committed';
COMMENT ON COLUMN sim_statistics.sim_elapsed_time IS 'total simulation time in seconds';
COMMENT ON COLUMN sim_statistics.sim_inst_rate IS 'simulation speed (in insts/sec)';
COMMENT ON COLUMN sim_statistics.sim_total_insn IS 'total number of instructions executed';
COMMENT ON COLUMN sim_statistics.sim_total_refs IS 'total number of loads and stores executed';
COMMENT ON COLUMN sim_statistics.sim_total_loads IS 'total number of loads executed';
COMMENT ON COLUMN sim_statistics.sim_total_stores IS 'total number of stores executed';
COMMENT ON COLUMN sim_statistics.sim_total_branches IS 'total number of branches executed';

```

COMMENT ON COLUMN sim_statistics.sim_cycle IS 'total simulation time in cycles';
COMMENT ON COLUMN sim_statistics."sim_IPC" IS 'instructions per cycle';
COMMENT ON COLUMN sim_statistics."sim_CPI" IS 'cycles per instruction';
COMMENT ON COLUMN sim_statistics."sim_exec_BW" IS 'total instructions (mis-spec + committed) per cycle';
COMMENT ON COLUMN sim_statistics."sim_IPB" IS 'instruction per branch';
COMMENT ON COLUMN sim_statistics."IFQ_count" IS 'cumulative IFQ occupancy';
COMMENT ON COLUMN sim_statistics."IFQ_fcount" IS 'cumulative IFQ full count';
COMMENT ON COLUMN sim_statistics.ifq_occupancy IS 'avg IFQ occupancy (insn's)';
COMMENT ON COLUMN sim_statistics.ifq_rate IS 'avg IFQ dispatch rate (insn/cycle)';
COMMENT ON COLUMN sim_statistics.ifq_latency IS 'avg IFQ occupant latency (cycle's)';
COMMENT ON COLUMN sim_statistics.ifq_full IS 'fraction of time (cycle's) IFQ was full';
COMMENT ON COLUMN sim_statistics."RUU_count" IS 'cumulative RUU occupancy';
COMMENT ON COLUMN sim_statistics."RUU_fcount" IS 'cumulative RUU full count';
COMMENT ON COLUMN sim_statistics.ruu_occupancy IS 'avg RUU occupancy (insn's)';
COMMENT ON COLUMN sim_statistics.ruu_rate IS 'avg RUU dispatch rate (insn/cycle)';
COMMENT ON COLUMN sim_statistics.ruu_latency IS 'avg RUU occupant latency (cycle's)';
COMMENT ON COLUMN sim_statistics.ruu_full IS 'fraction of time (cycle's) RUU was full';
COMMENT ON COLUMN sim_statistics."LSQ_count" IS 'cumulative LSQ occupancy';
COMMENT ON COLUMN sim_statistics."LSQ_fcount" IS 'cumulative LSQ full count';
COMMENT ON COLUMN sim_statistics.lsq_occupancy IS 'avg LSQ occupancy (insn's)';
COMMENT ON COLUMN sim_statistics.lsq_rate IS 'avg LSQ dispatch rate (insn/cycle)';
COMMENT ON COLUMN sim_statistics.lsq_latency IS 'avg LSQ occupant latency (cycle's)';
COMMENT ON COLUMN sim_statistics.lsq_full IS 'fraction of time (cycle's) LSQ was full';
COMMENT ON COLUMN sim_statistics.sim_slip IS 'total number of slip cycles';
COMMENT ON COLUMN sim_statistics.avg_sim_slip IS 'the average slip between issue and retirement';
COMMENT ON COLUMN sim_statistics.bpred_bimod_lookups IS 'total number of bpred lookups';
COMMENT ON COLUMN sim_statistics.bpred_bimod_updates IS 'total number of updates';
COMMENT ON COLUMN sim_statistics.bpred_bimod_addr_hits IS 'total number of address-predicted hits';
COMMENT ON COLUMN sim_statistics.bpred_bimod_dir_hits IS 'total number of direction-predicted hits';
COMMENT ON COLUMN sim_statistics.bpred_bimod_misses IS 'total number of misses';
COMMENT ON COLUMN sim_statistics.bpred_bimod_jr_hits IS 'total number of address-predicted hits for JR';
COMMENT ON COLUMN sim_statistics.bpred_bimod_jr_seen IS 'total number of JR's seen';
COMMENT ON COLUMN sim_statistics."bpred_bimod_jr_non_ras_hits_PP" IS 'total number of address-predicted hits for JR that were not RAS hits';
COMMENT ON COLUMN sim_statistics."bpred_bimod_jr_non_ras_seen_PP" IS 'total number of non-RAS JR's seen';
COMMENT ON COLUMN sim_statistics.bpred_bimod_bpred_addr_rate IS 'branch address-prediction rate (i.e., hits/lookups)';
COMMENT ON COLUMN sim_statistics.bpred_bimod_bpred_dir_rate IS 'branch direction-prediction rate (i.e., hits/lookups)';
COMMENT ON COLUMN sim_statistics.bpred_bimod_bpred_jr_rate IS 'JR address-prediction rate (i.e., JR hits/lookups)';
COMMENT ON COLUMN sim_statistics."bpred_bimod_bpred_jr_non_ras_rate_PP" IS 'non-RAS JR addr-pred rate (i.e., JR hits/lookups)';
COMMENT ON COLUMN sim_statistics.bpred_bimod_retstack_pushes IS 'total number of address pushed onto retstack';
COMMENT ON COLUMN sim_statistics.bpred_bimod_retstack_pops IS 'total number of address popped off of retstack';
COMMENT ON COLUMN sim_statistics."bpred_bimod_used_ras_PP" IS 'total number of RAS predictions used';
COMMENT ON COLUMN sim_statistics."bpred_bimod_ras_hits_PP" IS 'total number of RAS hits';
COMMENT ON COLUMN sim_statistics."bpred_bimod_ras_rate_PP" IS 'RAS prediction rate (i.e., RAS hits/lookups)';

COMMENT ON COLUMN sim_statistics.il1_accesses IS 'total number of accesses';
COMMENT ON COLUMN sim_statistics.il1_hits IS 'total number of hits';
COMMENT ON COLUMN sim_statistics.il1_misses IS 'total number of misses';
COMMENT ON COLUMN sim_statistics.il1_replacements IS 'total number of replacements';
COMMENT ON COLUMN sim_statistics.il1_writebacks IS 'total number of writebacks';
COMMENT ON COLUMN sim_statistics.il1_invalidations IS 'total number of invalidations';
COMMENT ON COLUMN sim_statistics.il1_miss_rate IS 'miss rate (i.e., misses/ref)';
COMMENT ON COLUMN sim_statistics.il1_repl_rate IS 'replacement rate (i.e., repls/ref)';
COMMENT ON COLUMN sim_statistics.il1_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
COMMENT ON COLUMN sim_statistics.il1_inv_rate IS 'invalidation rate (i.e., invs/ref)';

COMMENT ON COLUMN sim_statistics.dl1_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dl1_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dl1_miss_hit_mshr IS 'total number of miss hit in mshrs';
 COMMENT ON COLUMN sim_statistics.dl1_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dl1_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dl1_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.dl1_invalidations IS 'total number of invalidations';
 COMMENT ON COLUMN sim_statistics.dl1_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.dl1_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.dl1_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
 COMMENT ON COLUMN sim_statistics.dl1_inv_rate IS 'invalidation rate (i.e., invs/ref)';

COMMENT ON COLUMN sim_statistics.dl2_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dl2_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dl2_miss_hit_mshr IS 'total number of miss hit in mshrs';
 COMMENT ON COLUMN sim_statistics.dl2_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dl2_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dl2_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.dl2_invalidations IS 'total number of invalidations';
 COMMENT ON COLUMN sim_statistics.dl2_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.dl2_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.dl2_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
 COMMENT ON COLUMN sim_statistics.dl2_inv_rate IS 'invalidation rate (i.e., invs/ref)';

COMMENT ON COLUMN sim_statistics.mshr_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.mshr_miss_hits IS 'total number of miss hits';
 COMMENT ON COLUMN sim_statistics.mshr_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.mshr_full IS 'number of accesses in mshr full';
 COMMENT ON COLUMN sim_statistics.mshr_hits_false IS 'number of hits false';

COMMENT ON COLUMN sim_statistics.dcv_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dcv_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dcv_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dcv_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dcv_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.dcv_invalidations IS 'total number of invalidations';
 COMMENT ON COLUMN sim_statistics.dcv_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.dcv_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.dcv_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
 COMMENT ON COLUMN sim_statistics.dcv_inv_rate IS 'invalidation rate (i.e., invs/ref)';

COMMENT ON COLUMN sim_statistics.dsb_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dsb_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dsb_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dsb_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dsb_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.dsb_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.dsb_bus_occupation IS 'total cicles of bus_occupation';
 COMMENT ON COLUMN sim_statistics.dsb_reg1_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg2_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg3_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg4_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg5_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg6_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg7_hits IS 'total number of hits in registrators';

COMMENT ON COLUMN sim_statistics.dsb_reg8_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg9_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg10_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg11_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg12_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg13_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg14_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg15_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg16_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.dsb_reg17_hits IS 'total number of hits in registrators';

COMMENT ON COLUMN sim_statistics.dpcc_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dpcc_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dpcc_miss_hit_mshr IS 'total number of miss hit in mshrs';
 COMMENT ON COLUMN sim_statistics.dpcc_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dpcc_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dpcc_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.dpcc_invalidations IS 'total number of invalidations';
 COMMENT ON COLUMN sim_statistics.dpcc_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.dpcc_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.dpcc_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
 COMMENT ON COLUMN sim_statistics.dpcc_inv_rate IS 'invalidation rate (i.e., invs/ref)';

COMMENT ON COLUMN sim_statistics.ce_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.ce_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.ce_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.ce_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.ce_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.ce_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.ce_reg1_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg2_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg3_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg4_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg5_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg6_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg7_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg8_hits IS 'total number of hits in registrators';
 COMMENT ON COLUMN sim_statistics.ce_reg9_hits IS 'total number of hits in registrators';

COMMENT ON COLUMN sim_statistics.itlb_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.itlb_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.itlb_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.itlb_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.itlb_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.itlb_invalidations IS 'total number of invalidations';
 COMMENT ON COLUMN sim_statistics.itlb_miss_rate IS 'miss rate (i.e., misses/ref)';
 COMMENT ON COLUMN sim_statistics.itlb_repl_rate IS 'replacement rate (i.e., repls/ref)';
 COMMENT ON COLUMN sim_statistics.itlb_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
 COMMENT ON COLUMN sim_statistics.itlb_inv_rate IS 'invalidation rate (i.e., invs/ref)';
 COMMENT ON COLUMN sim_statistics.dtlb_accesses IS 'total number of accesses';
 COMMENT ON COLUMN sim_statistics.dtlb_hits IS 'total number of hits';
 COMMENT ON COLUMN sim_statistics.dtlb_misses IS 'total number of misses';
 COMMENT ON COLUMN sim_statistics.dtlb_replacements IS 'total number of replacements';
 COMMENT ON COLUMN sim_statistics.dtlb_writebacks IS 'total number of writebacks';
 COMMENT ON COLUMN sim_statistics.dtlb_invalidations IS 'total number of invalidations';

```

COMMENT ON COLUMN sim_statistics.dtlb_miss_rate IS 'miss rate (i.e., misses/ref)';
COMMENT ON COLUMN sim_statistics.dtlb_repl_rate IS 'replacement rate (i.e., repls/ref)';
COMMENT ON COLUMN sim_statistics.dtlb_wb_rate IS 'writeback rate (i.e., wrbks/ref)';
COMMENT ON COLUMN sim_statistics.dtlb_inv_rate IS 'invalidation rate (i.e., invs/ref)';
COMMENT ON COLUMN sim_statistics.sim_invalid_adrrs IS 'total non-speculative bogus addresses seen (d
COMMENT ON COLUMN sim_statistics.ld_text_base IS 'program text (code) segment base';
COMMENT ON COLUMN sim_statistics.ld_text_size IS 'program text (code) size in bytes';
COMMENT ON COLUMN sim_statistics.ld_data_base IS 'program initialized data segment base';
COMMENT ON COLUMN sim_statistics.ld_data_size IS 'program init''ed '.data'' and uninit''ed '.bss'' s
COMMENT ON COLUMN sim_statistics.ld_stack_base IS 'program stack segment base (highest address in st
COMMENT ON COLUMN sim_statistics.ld_stack_size IS 'program initial stack size';
COMMENT ON COLUMN sim_statistics.ld_prog_entry IS 'program entry point (initial PC)';
COMMENT ON COLUMN sim_statistics.ld_environ_base IS 'program environment base address address';
COMMENT ON COLUMN sim_statistics.ld_target_big_endian IS 'target executable endian-ness, non-zero if
COMMENT ON COLUMN sim_statistics.mem_page_count IS 'total number of pages allocated';
COMMENT ON COLUMN sim_statistics.mem_page_mem IS 'total size of memory pages allocated';
COMMENT ON COLUMN sim_statistics.mem_ptab_misses IS 'total first level page table misses';
COMMENT ON COLUMN sim_statistics.mem_ptab_accesses IS 'total page table accesses';
COMMENT ON COLUMN sim_statistics.mem_ptab_miss_rate IS 'first level page table miss rate';

```

```

CREATE OR REPLACE FUNCTION mult(real, real)
  RETURNS real AS
  $BODY$
DECLARE

n1 ALIAS FOR $1;
n2 ALIAS FOR $2;

BEGIN
return n1 * n2;
END;$BODY$
  LANGUAGE 'plpgsql' VOLATILE;
ALTER FUNCTION mult(real,real) OWNER TO postgres;
GRANT EXECUTE ON FUNCTION mult(real,real) TO public;
GRANT EXECUTE ON FUNCTION mult(real,real) TO postgres;
COMMENT ON FUNCTION mult(real,real) IS 'Função de multiplicação de dois números';

```

```
--select media_geom(distinct(lico_tm_cache)) from li_config;
```

```
--DROP AGGREGATE media_geom(float4);
```

```

CREATE AGGREGATE media_geom(
  BASETYPE=float4,
  SFUNC=mult,
  STYPE=float4,
-- FINALFUNC =,
  INITCOND=1
);
ALTER AGGREGATE media_geom(float4) OWNER TO postgres;

```

Apêndice C

Script AWK

Modificações no *script* foram necessárias para inserção de dados referentes às simulações de um novo conjunto de programas e diferentes parâmetros, como o modelo MSHR com associatividade 2. O *Script* AWK é aplicado sobre os arquivos de saída do simulador para preparar os resultados à serem inseridos na base de dados. O *Script* AWK também foi modificado pelos mesmos motivos. Este apêndice apresenta a versão final do *script* AWK utilizado para tratar os arquivos de resultados do SimpleScalar

```
#!/bin/awk
#Arquivo geraquery.awk
#Richard R. Souza

BEGIN {
    printf "INSERT INTO sim_statistics (\n"
    campos = "sim_id_caty, sim_id_lico, sim_id_bench"
    valores = " "

# valores = "1, 1"
# caty = ""
# cfg = ""
# bench = ""
}

# Valor do sim_id_caty
/^sim-outorder_base:/ { valores = "1"}
/^sim-outorder_cv:/ { valores = "2"}
/^sim-outorder_sb:/ { valores = "3"}
/^sim-outorder_mshr:/ { valores = "4"}
/^sim-outorder_mshr_cv:/ { valores = "5"}
/^sim-outorder_mshr_sb:/ { valores = "6"}
/^sim-outorder_mshr_cel:/ { valores = "7"}
```

```

/^sim-outorder_mshr_ce2:/ { valores = "8"}
/^sim-outorder_mshr_pcc1:/ { valores = "9"}
/^sim-outorder_mshr_pcc2:/ { valores = "10"}
/^sim-outorder_ncb:/ { valores = "11"}
/^sim-outorder_nts:/ { valores = "12"}
/^sim-outorder_lbic:/ { valores = "13"}
/^sim-outorder_mshr_sblat:/ { valores = "14"}

```

```
# Valor do sim_id_lico
```

```

/^sim:%%/00_00_1_/{ valores = valores ", 1" }
/^sim:%%/01_08_1_/{ valores = valores ", 2" }
/^sim:%%/01_16_1_/{ valores = valores ", 3" }
/^sim:%%/01_32_1_/{ valores = valores ", 4" }
/^sim:%%/01_64_1_/{ valores = valores ", 5" }
/^sim:%%/02_08_1_/{ valores = valores ", 6" }
/^sim:%%/02_16_1_/{ valores = valores ", 7" }
/^sim:%%/02_32_1_/{ valores = valores ", 8" }
/^sim:%%/02_64_1_/{ valores = valores ", 9" }
/^sim:%%/04_08_1_/{ valores = valores ", 10" }
/^sim:%%/04_16_1_/{ valores = valores ", 11" }
/^sim:%%/04_32_1_/{ valores = valores ", 12" }
/^sim:%%/04_64_1_/{ valores = valores ", 13" }
/^sim:%%/08_08_1_/{ valores = valores ", 14" }
/^sim:%%/08_16_1_/{ valores = valores ", 15" }
/^sim:%%/08_32_1_/{ valores = valores ", 16" }
/^sim:%%/08_64_1_/{ valores = valores ", 17" }
/^sim:%%/16_08_1_/{ valores = valores ", 18" }
/^sim:%%/16_16_1_/{ valores = valores ", 19" }
/^sim:%%/16_32_1_/{ valores = valores ", 20" }
/^sim:%%/16_64_1_/{ valores = valores ", 21" }
/^sim:%%/32_08_1_/{ valores = valores ", 22" }
/^sim:%%/32_16_1_/{ valores = valores ", 23" }
/^sim:%%/32_32_1_/{ valores = valores ", 24" }
/^sim:%%/32_64_1_/{ valores = valores ", 25" }
/^sim:%%/64_08_1_/{ valores = valores ", 26" }
/^sim:%%/64_16_1_/{ valores = valores ", 27" }
/^sim:%%/64_32_1_/{ valores = valores ", 28" }
/^sim:%%/64_64_1_/{ valores = valores ", 29" }
/^sim:%%/01_16_2_/{ valores = valores ", 30" }
/^sim:%%/01_32_2_/{ valores = valores ", 31" }
/^sim:%%/02_16_2_/{ valores = valores ", 32" }
/^sim:%%/02_32_2_/{ valores = valores ", 33" }
/^sim:%%/04_16_2_/{ valores = valores ", 34" }
/^sim:%%/04_32_2_/{ valores = valores ", 35" }
/^sim:%%/08_16_2_/{ valores = valores ", 36" }
/^sim:%%/08_32_2_/{ valores = valores ", 37" }
/^sim:%%/16_16_2_/{ valores = valores ", 38" }
/^sim:%%/16_32_2_/{ valores = valores ", 39" }
/^sim:%%/32_16_2_/{ valores = valores ", 40" }
/^sim:%%/32_32_2_/{ valores = valores ", 41" }
/^sim:%%/64_16_2_/{ valores = valores ", 42" }
/^sim:%%/64_32_2_/{ valores = valores ", 43" }

```

```
# Valor do sim_id_bench
```

```

/\anagram / { valores = valores ", 1" }

```

```

/\bubble /      { valores = valores ", 2" }
/>\insertion /   { valores = valores ", 3" }
/>\merge /      { valores = valores ", 4" }
/>\quick /      { valores = valores ", 5" }
/>\selection /   { valores = valores ", 6" }
/>\shell /      { valores = valores ", 7" }
/>\cast_dec /    { valores = valores ", 8" }
/>\cast_enc /    { valores = valores ", 9" }
/>\drr / { valores = valores ", 10" }
/>\frag /      { valores = valores ", 11" }
/>\jpeg_dec /    { valores = valores ", 12" }
/>\jpeg_enc /    { valores = valores ", 13" }
/>\reed_dec /    { valores = valores ", 14" }
/>\reed_enc /    { valores = valores ", 15" }
/>\rtr / { valores = valores ", 16" }

/>\adpcm_dec /   { valores = valores ", 17" }
/>\adpcm_enc / { valores = valores ", 18" }
/>\epic_dec /    { valores = valores ", 19" }
/>\epic_enc / { valores = valores ", 20" }
/>\g721_dec / { valores = valores ", 21" }
/>\g721_enc /    { valores = valores ", 22" }
/>\jpeg_mdec / { valores = valores ", 23" }
/>\jpeg_menc /   { valores = valores ", 24" }
/>\mesa_mipmap / { valores = valores ", 25" }
/>\mesa_osdemo / { valores = valores ", 26" }
/>\mesa_texgen / { valores = valores ", 27" }
/>\mpeg2_dec /   { valores = valores ", 28" }
/>\mpeg2_enc /   { valores = valores ", 29" }
/>\pegwit_dec /  { valores = valores ", 30" }
/>\pegwit_enc /  { valores = valores ", 31" }
/>\toast / { valores = valores ", 32" }
/>\untoast / { valores = valores ", 33" }
/>\zip_dec / { valores = valores ", 34" }
/>\zip_enc / { valores = valores ", 35" }

# Identificador da configuração do simulador
/^-fetch:ifqsize /&&/1 #/{ ifqsize = 1 }
/^-fetch:ifqsize /&&/2 #/{ ifqsize = 2 }
/^-fetch:ifqsize /&&/4 #/{ ifqsize = 4 }
/^-fetch:ifqsize /&&/8 #/{ ifqsize = 8 }
/^-mem:lat .*/ {latencia = ($2);}
/^-mem:width .*/ {bus = ($2);}
campos = campos ", sim_id_sico"
if ( ifqsize == 2 ) {
if( latencia == 18 ){
if (bus == 8){ valores = valores ", 1" }
else { valores = valores ", 6" } }
else if (bus == 8){ valores = valores ", 5" }
else { valores = valores ", 7" } }
if ( ifqsize == 1 ){ valores = valores ", 4" }
if ( ifqsize == 4 ){ valores = valores ", 2" }
if ( ifqsize == 8 ){ valores = valores ", 3" }
}

```



```

# Configurações específicas das caches
# sim_nr_mshr
/^-cache:mshr_dl1 /&&/mshr:1 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 1"}
/^-cache:mshr_dl1 /&&/mshr:2 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 2"}
/^-cache:mshr_dl1 /&&/mshr:4 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 4"}
/^-cache:mshr_dl1 /&&/mshr:8 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 8"}
/^-cache:mshr_dl1 /&&/mshr:16 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 16"}
/^-cache:mshr_dl1 /&&/mshr:32 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 32"}
/^-cache:mshr_dl1 /&&/mshr:64 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 64"}
/^-cache:mshr_dl1 /&&/mshr:128 #/{ campos = campos ", sim_nr_mshr"
valores = valores ", 128"}

# sim_nr_cv
/^-cache:dcv /&&/:1:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 1"}
/^-cache:dcv /&&/:2:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 2"}
/^-cache:dcv /&&/:4:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 4"}
/^-cache:dcv /&&/:8:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 8"}
/^-cache:dcv /&&/:16:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 16"}
/^-cache:dcv /&&/:32:1 #/{ campos = campos ", sim_nr_cv"
valores = valores ", 32"}

# sim_nr_sb
/^-streambuf:dsb /&&/dsb:1:/{ campos = campos ", sim_nr_sb"
valores = valores ", 1"}
/^-streambuf:dsb /&&/dsb:2:/{ campos = campos ", sim_nr_sb"
valores = valores ", 2"}
/^-streambuf:dsb /&&/dsb:4:/{ campos = campos ", sim_nr_sb"
valores = valores ", 4"}
/^-streambuf:dsb /&&/dsb:8:/{ campos = campos ", sim_nr_sb"
valores = valores ", 8"}
/^-streambuf:dsb /&&/dsb:16:/{ campos = campos ", sim_nr_sb"
valores = valores ", 16"}

# sim_assoc_sb
/^-streambuf:dsb /&&/:1 #/{ campos = campos ", sim_assoc_sb"
valores = valores ", 1"}
/^-streambuf:dsb /&&/:2 #/{ campos = campos ", sim_assoc_sb"
valores = valores ", 2"}
/^-streambuf:dsb /&&/:4 #/{ campos = campos ", sim_assoc_sb"
valores = valores ", 4"}
/^-streambuf:dsb /&&/:8 #/{ campos = campos ", sim_assoc_sb"
valores = valores ", 8"}

```

```

# sim_is_ce
/^# -redir:sim /&&/_ce/{ campos = campos ", sim_is_ce"
valores = valores ", 1" }

# sim_nr_pcc
/^-cache:dpcc /&&/:1:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 1"}
/^-cache:dpcc /&&/:2:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 2"}
/^-cache:dpcc /&&/:4:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 4"}
/^-cache:dpcc /&&/:8:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 8"}
/^-cache:dpcc /&&/:16:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 16"}
/^-cache:dpcc /&&/:32:1 #{ campos = campos ", sim_nr_pcc"
valores = valores ", 32"}

# Resultados das simulações
/^sim_num_insn .*/{ campos = campos ", sim_num_insn"
valores = valores ", " ($2)}
/^sim_num_refs .*/{ campos = campos ", sim_num_refs"
valores = valores ", " ($2)}
/^sim_num_loads .*/{ campos = campos ", sim_num_loads"
valores = valores ", " ($2)}
/^sim_num_stores .*/{ campos = campos ", sim_num_stores"
valores = valores ", " ($2)}
/^sim_num_branches .*/{ campos = campos ", sim_num_branches"
valores = valores ", " ($2)}
/^sim_elapsed_time .*/{ campos = campos ", sim_elapsed_time"
valores = valores ", " ($2)}
/^sim_inst_rate .*/{ campos = campos ", sim_inst_rate"
valores = valores ", " ($2)}
/^sim_total_insn .*/{ campos = campos ", sim_total_insn"
valores = valores ", " ($2)}
/^sim_total_refs .*/{ campos = campos ", sim_total_refs"
valores = valores ", " ($2)}
/^sim_total_loads .*/{ campos = campos ", sim_total_loads"
valores = valores ", " ($2)}
/^sim_total_stores .*/{ campos = campos ", sim_total_stores"
valores = valores ", " ($2)}
/^sim_total_branches .*/{ campos = campos ", sim_total_branches"
valores = valores ", " ($2)}
/^sim_cycle .*/{ campos = campos ", sim_cycle"
valores = valores ", " ($2)}
/^sim_IPC .*/{ campos = campos ", \"sim_IPC\""}
valores = valores ", " ($2)}
/^sim_CPI .*/{ campos = campos ", \"sim_CPI\""}
valores = valores ", " ($2)}
/^sim_exec_BW .*/{ campos = campos ", \"sim_exec_BW\""}
valores = valores ", " ($2)}
/^sim_IPB .*/{ campos = campos ", \"sim_IPB\""}
valores = valores ", " ($2)}

```

```

/^IFQ_count .*/{ campos = campos ", \"IFQ_count\""
    valores = valores ", " ($2)}
/^IFQ_fcount .*/{ campos = campos ", \"IFQ_fcount\""
    valores = valores ", " ($2)}
/^ifq_occupancy .*/{ campos = campos ", ifq_occupancy"
    valores = valores ", " ($2)}
/^ifq_rate .*/{ campos = campos ", ifq_rate"
    valores = valores ", " ($2)}
/^ifq_latency .*/{ campos = campos ", ifq_latency"
    valores = valores ", " ($2)}
/^ifq_full .*/{ campos = campos ", ifq_full"
    valores = valores ", " ($2)}
/^RUU_count .*/{ campos = campos ", \"RUU_count\""
    valores = valores ", " ($2)}
/^RUU_fcount .*/{ campos = campos ", \"RUU_fcount\""
    valores = valores ", " ($2)}
/^ruu_occupancy .*/{ campos = campos ", ruu_occupancy"
    valores = valores ", " ($2)}
/^ruu_rate .*/{ campos = campos ", ruu_rate"
    valores = valores ", " ($2)}
/^ruu_latency .*/{ campos = campos ", ruu_latency"
    valores = valores ", " ($2)}
/^ruu_full .*/{ campos = campos ", ruu_full"
    valores = valores ", " ($2)}
/^LSQ_count .*/{ campos = campos ", \"LSQ_count\""
    valores = valores ", " ($2)}
/^LSQ_fcount .*/{ campos = campos ", \"LSQ_fcount\""
    valores = valores ", " ($2)}
/^lsq_occupancy .*/{ campos = campos ", lsq_occupancy"
    valores = valores ", " ($2)}
/^lsq_rate .*/{ campos = campos ", lsq_rate"
    valores = valores ", " ($2)}
/^lsq_latency .*/{ campos = campos ", lsq_latency"
    valores = valores ", " ($2)}
/^lsq_full .*/{ campos = campos ", lsq_full"
    valores = valores ", " ($2)}
/^sim_slip .*/{ campos = campos ", sim_slip"
    valores = valores ", " ($2)}
/^avg_sim_slip .*/{ campos = campos ", avg_sim_slip"
    valores = valores ", " ($2)}
/^bpred_bimod.lookups .*/{ campos = campos ", bpred_bimod_lookups"
    valores = valores ", " ($2)}
/^bpred_bimod.updates .*/{ campos = campos ", bpred_bimod_updates"
    valores = valores ", " ($2)}
/^bpred_bimod.addr_hits .*/{ campos = campos ", bpred_bimod_addr_hits"
    valores = valores ", " ($2)}
/^bpred_bimod.dir_hits .*/{ campos = campos ", bpred_bimod_dir_hits"
    valores = valores ", " ($2)}
/^bpred_bimod.misses .*/{ campos = campos ", bpred_bimod_misses"
    valores = valores ", " ($2)}
/^bpred_bimod.jr_hits .*/{ campos = campos ", bpred_bimod_jr_hits"
    valores = valores ", " ($2)}
/^bpred_bimod.jr_seen .*/{ campos = campos ", bpred_bimod_jr_seen"
    valores = valores ", " ($2)}
/^bpred_bimod.jr_non_ras_hits.PP .*/{ campos = campos ", \"bpred_bimod_jr_non_ras_hits_PP\""

```

```

    valores = valores ", " ($2)}
/^bpred_bimod.jr_non_ras_seen.PP .*/{ campos = campos ", \"bpred_bimod_jr_non_ras_seen_PP\""}
    valores = valores ", " ($2)}
/^bpred_bimod.bpred_addr_rate .*/{ campos = campos ", bpred_bimod_bpred_addr_rate"}
    valores = valores ", " ($2)}
/^bpred_bimod.bpred_dir_rate .*/{ campos = campos ", bpred_bimod_bpred_dir_rate"}
    valores = valores ", " ($2)}
/^bpred_bimod.bpred_jr_rate .*/{ campos = campos ", bpred_bimod_bpred_jr_rate"}
    valores = valores ", " ($2)}
/^bpred_bimod.bpred_jr_non_ras_rate.PP .*/{
if ($2 ~ /<error:/){
campos = campos ", \"bpred_bimod_bpred_jr_non_ras_rate_PP\""}
valores = valores ", 0"}
}
else {
campos = campos ", \"bpred_bimod_bpred_jr_non_ras_rate_PP\""}
valores = valores ", " ($2)}
}
}
/^bpred_bimod.retstack_pushes .*/{ campos = campos ", bpred_bimod_retstack_pushes"}
    valores = valores ", " ($2)}
/^bpred_bimod.retstack_pops .*/{ campos = campos ", bpred_bimod_retstack_pops"}
    valores = valores ", " ($2)}
/^bpred_bimod.used_ras.PP .*/{ campos = campos ", \"bpred_bimod_used_ras_PP\""}
    valores = valores ", " ($2)}
/^bpred_bimod.ras_hits.PP .*/{ campos = campos ", \"bpred_bimod_ras_hits_PP\""}
    valores = valores ", " ($2)}
/^bpred_bimod.ras_rate.PP .*/{ campos = campos ", \"bpred_bimod_ras_rate_PP\""}
    valores = valores ", " ($2)}

/^il1.accesses .*/{ campos = campos ", il1_accesses"}
    valores = valores ", " ($2)}
/^il1.hits .*/{ campos = campos ", il1_hits"}
    valores = valores ", " ($2)}
/^il1.misses .*/{ campos = campos ", il1_misses"}
    valores = valores ", " ($2)}
/^il1.replacements .*/{ campos = campos ", il1_replacements"}
    valores = valores ", " ($2)}
/^il1.writebacks .*/{ campos = campos ", il1_writebacks"}
    valores = valores ", " ($2)}
/^il1.invalidations .*/{ campos = campos ", il1_invalidations"}
    valores = valores ", " ($2)}
/^il1.miss_rate .*/{ campos = campos ", il1_miss_rate"}
    valores = valores ", " ($2)}
/^il1.repl_rate .*/{ campos = campos ", il1_repl_rate"}
    valores = valores ", " ($2)}
/^il1.wb_rate .*/{ campos = campos ", il1_wb_rate"}
    valores = valores ", " ($2)}
/^il1.inv_rate .*/{ campos = campos ", il1_inv_rate"}
    valores = valores ", " ($2)}

/^dl1.accesses .*/{ campos = campos ", dl1_accesses"}
    valores = valores ", " ($2)}
/^dl1.hits .*/{ campos = campos ", dl1_hits"}
    valores = valores ", " ($2)}

```

```

/^dl1.miss_hit_mshr .*/{ campos = campos ", dl1_miss_hit_mshr"
    valores = valores ", " ($2)}
/^dl1.misses .*/{ campos = campos ", dl1_misses"
    valores = valores ", " ($2)}
/^dl1.replacements .*/{ campos = campos ", dl1_replacements"
    valores = valores ", " ($2)}
/^dl1.writebacks .*/{ campos = campos ", dl1_writebacks"
    valores = valores ", " ($2)}
/^dl1.invalidations .*/{ campos = campos ", dl1_invalidations"
    valores = valores ", " ($2)}
/^dl1.miss_rate .*/{ campos = campos ", dl1_miss_rate"
    valores = valores ", " ($2)}
/^dl1.repl_rate .*/{ campos = campos ", dl1_repl_rate"
    valores = valores ", " ($2)}
/^dl1.wb_rate .*/{ campos = campos ", dl1_wb_rate"
    valores = valores ", " ($2)}
/^dl1.inv_rate .*/{ campos = campos ", dl1_inv_rate"
    valores = valores ", " ($2)}

/^dl2.accesses .*/{ campos = campos ", dl2_accesses"
    valores = valores ", " ($2)}
/^dl2.hits .*/{ campos = campos ", dl2_hits"
    valores = valores ", " ($2)}
/^dl2.miss_hit_mshr .*/{ campos = campos ", dl2_miss_hit_mshr"
    valores = valores ", " ($2)}
/^dl2.misses .*/{ campos = campos ", dl2_misses"
    valores = valores ", " ($2)}
/^dl2.replacements .*/{ campos = campos ", dl2_replacements"
    valores = valores ", " ($2)}
/^dl2.writebacks .*/{ campos = campos ", dl2_writebacks"
    valores = valores ", " ($2)}
/^dl2.invalidations .*/{ campos = campos ", dl2_invalidations"
    valores = valores ", " ($2)}
/^dl2.miss_rate .*/{ campos = campos ", dl2_miss_rate"
    valores = valores ", " ($2)}
/^dl2.repl_rate .*/{ campos = campos ", dl2_repl_rate"
    valores = valores ", " ($2)}
/^dl2.wb_rate .*/{ campos = campos ", dl2_wb_rate"
    valores = valores ", " ($2)}
/^dl2.inv_rate .*/{ campos = campos ", dl2_inv_rate"
    valores = valores ", " ($2)}

/^mshr.accesses .*/{ campos = campos ", mshr_accesses"
    valores = valores ", " ($2)}
/^mshr.miss_hits .*/{ campos = campos ", mshr_miss_hits"
    valores = valores ", " ($2)}
/^mshr.misses .*/{ campos = campos ", mshr_misses"
    valores = valores ", " ($2)}
/^mshr.full .*/{ campos = campos ", mshr_full"
    valores = valores ", " ($2)}
/^mshr.hits_false .*/{ campos = campos ", mshr_hits_false"
    valores = valores ", " ($2)}

/^dcv.accesses .*/{ campos = campos ", dcv_accesses"
    valores = valores ", " ($2)}

```

```

/^dcv.hits .*/{ campos = campos ", dcv_hits"
    valores = valores ", " ($2)}
/^dcv.misses .*/{ campos = campos ", dcv_misses"
    valores = valores ", " ($2)}
/^dcv.replacements .*/{ campos = campos ", dcv_replacements"
    valores = valores ", " ($2)}
/^dcv.writebacks .*/{ campos = campos ", dcv_writebacks"
    valores = valores ", " ($2)}
/^dcv.invalidations .*/{ campos = campos ", dcv_invalidations"
    valores = valores ", " ($2)}
/^dcv.miss_rate .*/{ campos = campos ", dcv_miss_rate"
    valores = valores ", " ($2)}
/^dcv.repl_rate .*/{ campos = campos ", dcv_repl_rate"
    valores = valores ", " ($2)}
/^dcv.wb_rate .*/{ campos = campos ", dcv_wb_rate"
    valores = valores ", " ($2)}
/^dcv.inv_rate .*/{ campos = campos ", dcv_inv_rate"
    valores = valores ", " ($2)}

/^dsb.accesses .*/{ campos = campos ", dsb_accesses"
    valores = valores ", " ($2)}
/^dsb.hits .*/{ campos = campos ", dsb_hits"
    valores = valores ", " ($2)}
/^dsb.misses .*/{ campos = campos ", dsb_misses"
    valores = valores ", " ($2)}
/^dsb.replacements .*/{ campos = campos ", dsb_replacements"
    valores = valores ", " ($2)}
/^dsb.miss_rate .*/{ campos = campos ", dsb_miss_rate"
    valores = valores ", " ($2)}
/^dsb.repl_rate .*/{ campos = campos ", dsb_repl_rate"
    valores = valores ", " ($2)}
/^dsb.bus_occupation .*/{ campos = campos ", dsb_bus_occupation"
    valores = valores ", " ($2)}
/^dsb.reg1_hits .*/{ campos = campos ", dsb_reg1_hits"
    valores = valores ", " ($2)}
/^dsb.reg2_hits .*/{ campos = campos ", dsb_reg2_hits"
    valores = valores ", " ($2)}
/^dsb.reg3_hits .*/{ campos = campos ", dsb_reg3_hits"
    valores = valores ", " ($2)}
/^dsb.reg4_hits .*/{ campos = campos ", dsb_reg4_hits"
    valores = valores ", " ($2)}
/^dsb.reg5_hits .*/{ campos = campos ", dsb_reg5_hits"
    valores = valores ", " ($2)}
/^dsb.reg6_hits .*/{ campos = campos ", dsb_reg6_hits"
    valores = valores ", " ($2)}
/^dsb.reg7_hits .*/{ campos = campos ", dsb_reg7_hits"
    valores = valores ", " ($2)}
/^dsb.reg8_hits .*/{ campos = campos ", dsb_reg8_hits"
    valores = valores ", " ($2)}
/^dsb.reg9_hits .*/{ campos = campos ", dsb_reg9_hits"
    valores = valores ", " ($2)}
/^dsb.reg10_hits .*/{ campos = campos ", dsb_reg10_hits"
    valores = valores ", " ($2)}
/^dsb.reg11_hits .*/{ campos = campos ", dsb_reg11_hits"
    valores = valores ", " ($2)}

```

```

/^dsb.reg12_hits .*/{ campos = campos ", dsb_reg12_hits"
    valores = valores ", " ($2)}
/^dsb.reg13_hits .*/{ campos = campos ", dsb_reg13_hits"
    valores = valores ", " ($2)}
/^dsb.reg14_hits .*/{ campos = campos ", dsb_reg14_hits"
    valores = valores ", " ($2)}
/^dsb.reg15_hits .*/{ campos = campos ", dsb_reg15_hits"
    valores = valores ", " ($2)}
/^dsb.reg16_hits .*/{ campos = campos ", dsb_reg16_hits"
    valores = valores ", " ($2)}
/^dsb.reg17_hits .*/{ campos = campos ", dsb_reg17_hits"
    valores = valores ", " ($2)}

/^dpcc.accesses .*/{ campos = campos ", dpcc_accesses"
    valores = valores ", " ($2)}
/^dpcc.hits .*/{ campos = campos ", dpcc_hits"
    valores = valores ", " ($2)}
/^dpcc.miss_hit_mshr .*/{ campos = campos ", dpcc_miss_hit_mshr"
    valores = valores ", " ($2)}
/^dpcc.misses .*/{ campos = campos ", dpcc_misses"
    valores = valores ", " ($2)}
/^dpcc.replacements .*/{ campos = campos ", dpcc_replacements"
    valores = valores ", " ($2)}
/^dpcc.writebacks .*/{ campos = campos ", dpcc_writebacks"
    valores = valores ", " ($2)}
/^dpcc.invalidations .*/{ campos = campos ", dpcc_invalidations"
    valores = valores ", " ($2)}
/^dpcc.miss_rate .*/{ campos = campos ", dpcc_miss_rate"
    valores = valores ", " ($2)}
/^dpcc.repl_rate .*/{ campos = campos ", dpcc_repl_rate"
    valores = valores ", " ($2)}
/^dpcc.wb_rate .*/{ campos = campos ", dpcc_wb_rate"
    valores = valores ", " ($2)}
/^dpcc.inv_rate .*/{ campos = campos ", dpcc_inv_rate"
    valores = valores ", " ($2)}

#/^ce.accesses .*/{ campos = campos ", ce_accesses"
#    valores = valores ", " ($2)}
#/^ce.hits .*/{ campos = campos ", ce_hits"
#    valores = valores ", " ($2)}
#/^ce.misses .*/{ campos = campos ", ce_misses"
#    valores = valores ", " ($2)}
#/^ce.replacements .*/{ campos = campos ", ce_replacements"
#    valores = valores ", " ($2)}
#/^ce.miss_rate .*/{ campos = campos ", ce_miss_rate"
#    valores = valores ", " ($2)}
#/^ce.repl_rate .*/{ campos = campos ", ce_repl_rate"
#    valores = valores ", " ($2)}
#/^ce.reg1_hits .*/{ campos = campos ", ce_reg1_hits"
#    valores = valores ", " ($2)}
#/^ce.reg2_hits .*/{ campos = campos ", ce_reg2_hits"
#    valores = valores ", " ($2)}
#/^ce.reg3_hits .*/{ campos = campos ", ce_reg3_hits"
#    valores = valores ", " ($2)}
#/^ce.reg4_hits .*/{ campos = campos ", ce_reg4_hits"

```

```

#     valores = valores ", " ($2)}
#/^ce.reg5_hits .*/{ campos = campos ", ce_reg5_hits"
#     valores = valores ", " ($2)}
#/^ce.reg6_hits .*/{ campos = campos ", ce_reg6_hits"
#     valores = valores ", " ($2)}
#/^ce.reg7_hits .*/{ campos = campos ", ce_reg7_hits"
#     valores = valores ", " ($2)}
#/^ce.reg8_hits .*/{ campos = campos ", ce_reg8_hits"
#     valores = valores ", " ($2)}
#/^ce.reg9_hits .*/{ campos = campos ", ce_reg9_hits"
#     valores = valores ", " ($2)}

/^itlb.accesses .*/{ campos = campos ", itlb_accesses"
    valores = valores ", " ($2)}
/^itlb.hits .*/{ campos = campos ", itlb_hits"
    valores = valores ", " ($2)}
/^itlb.misses .*/{ campos = campos ", itlb_misses"
    valores = valores ", " ($2)}
/^itlb.replacements .*/{ campos = campos ", itlb_replacements"
    valores = valores ", " ($2)}
/^itlb.writebacks .*/{ campos = campos ", itlb_writebacks"
    valores = valores ", " ($2)}
/^itlb.invalidations .*/{ campos = campos ", itlb_invalidations"
    valores = valores ", " ($2)}
/^itlb.miss_rate .*/{ campos = campos ", itlb_miss_rate"
    valores = valores ", " ($2)}
/^itlb.repl_rate .*/{ campos = campos ", itlb_repl_rate"
    valores = valores ", " ($2)}
/^itlb.wb_rate .*/{ campos = campos ", itlb_wb_rate"
    valores = valores ", " ($2)}
/^itlb.inv_rate .*/{ campos = campos ", itlb_inv_rate"
    valores = valores ", " ($2)}
/^dtlb.accesses .*/{ campos = campos ", dtlb_accesses"
    valores = valores ", " ($2)}
/^dtlb.hits .*/{ campos = campos ", dtlb_hits"
    valores = valores ", " ($2)}
/^dtlb.misses .*/{ campos = campos ", dtlb_misses"
    valores = valores ", " ($2)}
/^dtlb.replacements .*/{ campos = campos ", dtlb_replacements"
    valores = valores ", " ($2)}
/^dtlb.writebacks .*/{ campos = campos ", dtlb_writebacks"
    valores = valores ", " ($2)}
/^dtlb.invalidations .*/{ campos = campos ", dtlb_invalidations"
    valores = valores ", " ($2)}
/^dtlb.miss_rate .*/{ campos = campos ", dtlb_miss_rate"
    valores = valores ", " ($2)}
/^dtlb.repl_rate .*/{ campos = campos ", dtlb_repl_rate"
    valores = valores ", " ($2)}
/^dtlb.wb_rate .*/{ campos = campos ", dtlb_wb_rate"
    valores = valores ", " ($2)}
/^dtlb.inv_rate .*/{ campos = campos ", dtlb_inv_rate"
    valores = valores ", " ($2)}
/^sim_invalid_addr .*/{ campos = campos ", sim_invalid_addr"
    valores = valores ", " ($2)}
/^ld_text_base .*/{ campos = campos ", ld_text_base"

```



```

    valores = valores ", '" ($2) '"'}
/^ld_text_size .*/{ campos = campos " ld_text_size"
    valores = valores ", " ($2)}
/^ld_data_base .*/{ campos = campos " ld_data_base"
    valores = valores ", '" ($2) '"'}
/^ld_data_size .*/{ campos = campos " ld_data_size"
    valores = valores ", " ($2)}
/^ld_stack_base .*/{ campos = campos " ld_stack_base"
    valores = valores ", '" ($2) '"'}
/^ld_stack_size .*/{ campos = campos " ld_stack_size"
    valores = valores ", " ($2)}
/^ld_prog_entry .*/{ campos = campos " ld_prog_entry"
    valores = valores ", '" ($2) '"'}
/^ld_environ_base .*/{ campos = campos " ld_environ_base"
    valores = valores ", '" ($2) '"'}
/^ld_target_big_endian .*/{ campos = campos " ld_target_big_endian"
    valores = valores ", " ($2)}
/^mem.page_count .*/{ campos = campos " mem_page_count"
    valores = valores ", " ($2)}
/^mem.page_mem .*/{ campos = campos " mem_page_mem"
    valores = valores ", '" ($2) '"'}
/^mem.ptab_misses .*/{ campos = campos " mem_ptab_misses"
    valores = valores ", " ($2)}
/^mem.ptab_accesses .*/{ campos = campos " mem_ptab_accesses"
    valores = valores ", " ($2)}
/^mem.ptab_miss_rate .*/{ campos = campos " mem_ptab_miss_rate"
    valores = valores ", " ($2)}

END{
# campos +=
# valores +=
    printf campos "\n) VALUES (\n"
    printf valores "\n);\n"
}

```