PowerSMT: Ferramenta para Análise de Consumo de Potência em Arquiteturas SMT[1]

Rogério Aparecido Gonçalves[2], Ronaldo Augusto de Lara Gonçalves
Universidade Estadual de Maringá - Departamento de Informática
Programa de Pós Graduação em Ciência da Computação
LECAD (Laboratório Experimental de Computação de Alto Desempenho)

Avenida Colombo 5790, Maringá, Paraná, Brasil
{rogerio, ronaldo}@din.uem.br

Resumo

As aplicações estão mais complexas, exigindo um processamento mais agressivo, tal como aquele obtido com as atuais arquiteturas SMT. Para avaliar o desempenho destas arquiteturas, ainda na fase de projeto, a ferramenta SS_SMT tem sido usada satisfatoriamente. Entretanto, como nestas arquiteturas as estruturas e os circuitos são maiores do que aqueles existentes nas arquiteturas superescalares convencionais, espera-se que o consumo de potência também seja maior. Nesse sentido, maiores estudos se fazem necessários para avaliar esta questão. Neste trabalho, a ferramenta SS_SMT foi adaptada para permitir a análise de consumo eficiente de potência, dando origem a nova ferramenta PowerSMT, a qual tem como base as plataformas SimpleScalar e CACTI. PowerSMT foi exaustivamente experimentada na análise da cache de instruções, estações de reserva e unidades funcionais. Os resultados sobre benchmarks do SPEC mostraram que a ferramenta é de fundamental importância na análise do consumo de potência em arquiteturas SMT.

1. Introdução

A evolução das aplicações faz com que ocorram também melhorias nos processadores de forma a responderem à carga de trabalho imposta pelos novos contextos e instruções. Assim, faz-se necessário o desenvolvimento de um processamento efetivo que atenda a demanda computacional criada, intensificando a utilização de recursos existentes e a proposição de novos modelos e estratégias para a obtenção de desempenho satisfatório.

Pode-se elencar alguns modelos arquiteturais que levaram à obtenção de melhorias no desempenho, sendo alguns deles: a arquitetura pipeline, arquiteturas superescalares, arquiteturas de múltiplas tarefas simultâneas (SMT), arquiteturas de múltiplos núcleos e de múltiplos núcleos com suporte a SMT, as quais estão sendo exploradas correntemente e com perspectivas de crescimento.

Essa complexidade crescente está diretamente relacionada ao aumento da complexidade do hardware, necessário para suportar a demanda da computação exigida com o melhor desempenho possível. Seja para aumentar a autonomia dos dispositivos móveis ou embarcados, reduzir o aquecimento, disponibilizar maior área do chip, prover a estabilidade dos componentes ou ainda para melhorar o desempenho, o controle do consumo de potência passa a ter uma importância fundamental nos projetos de novos processadores. É preciso alavancar as pesquisas nessa área e formular propostas que foquem a redução do consumo de potência em todos os níveis sistêmicos.

Este trabalho está inserido nesse contexto, tendo como foco a investigação do consumo de potência em arquiteturas SMT, em continuidade aos trabalhos iniciados em [28]. A preocupação está no fato das arquiteturas SMT terem uma utilização mais intensiva dos recursos e isto sugere um consumo de potência maior que precisa ser medido, estudado e avaliado. Com isso, será possível elaborar estratégias de controle e otimização para a utilização, a distribuição e o compartilhamento de recursos visando a redução do consumo de potência sem prejuízos ao desempenho

1Trabalho vinculado aos projetos PARANAE e PRÓ-ARQ II.
2 Mestrando PCC-DIN-UEM.

desejado. O presente artigo está organizado da seguinte forma. A próxima seção descreve sobre as arquiteturas SMT. A seção 3 aborda o consumo de potência. Os simuladores relacionados aparecem na seção 4. A seção 5 explica a proposta e a implementação da ferramenta PowerSMT. Os experimentos e as conclusões são mostrados nas seções 6 e 7. No final tem-se os agradecimentos e as referências utilizadas.

2. Arquiteturas SMT

A denominação *Simultaneous Multithreading* (SMT) teve origem em [20], embora o conceito tenha aparecido em trabalhos anteriores, como arquitetura Simultaneous Instruction Issuing [6] e Multistreamed Superscalar [21], ambos com a capacidade de executar instruções de diferentes fluxos simultaneamente.

As arquiteturas SMT têm como objetivo a execução simultânea de tarefas, remetendo à execução instruções de diferentes tarefas durante o mesmo ciclo do processador. Cada tarefa detém durante a execução o uso de alguns recursos que constituem o seu contexto de execução, compartilhando diversos outros com outras tarefas em execução. O contexto de cada tarefa é formado pelo conteúdo dos registradores, contador de programa e demais recursos que estão sob seu domínio.

Através do embaralhamento das instruções das diversas tarefas em execução simultânea, a arquitetura explora o paralelismo em nível de tarefas (TLP), em adição ao ILP, o qual é típico das arquiteturas superescalares convencionais. Esta característica proporciona um melhor aproveitamento dos recursos existentes, pois aqueles que estariam ociosos com a execução de uma única tarefa poderão ser aproveitados pela execução simultânea de outras tarefas [20, 14].

A unidade de busca, entre outros recursos, compartilhada pelas diversas tarefas, permite que as instruções das tarefas sejam buscadas e inseridas no caminho de execução, onde a coexistência de vários contextos permite aproveitar melhor as oportunidades de execução enquanto as tarefas concorrentes alternam seus momentos de *cpu-bound* e *io-bound*.

O compartilhamento de recursos e estruturas internas do processador pode ser feito de diversas maneiras e da forma que melhor se adequar as especificidades do projeto arquitetural. Por exemplo, a implementação da Intel, denominada tecnologia *Hyper Threading*, implementa um pipeline para a execução simultânea de apenas duas tarefas, que é visto pelo SO como dois processadores físicos [14].

Com o surgimento de arquiteturas de múltiplos núcleos em um único processador físico ou Chip Multiprocessors (CMP) [15], houve certa redução nas pesquisas em SMT. Entretanto, devido aos limites de desempenho alcançado por esta nova abordagem, o desejo atual é que os processadores passem a combinar as duas estratégias em uma única arquitetura. Com isso, surgem as arquiteturas Chip Multi-Threaded (CMT) [18] como sendo processadores que suportam a execução simultânea de muitas tarefas via combinação do suporte a múltiplos núcleos com SMT. A aplicação de SMT em cada núcleo é feita para os mesmos fins que em processadores de núcleo único, melhor aproveitando os recursos e as oportunidades de execução durante a latência de operações de memória.

Alguns processadores podem ser classificados dentro desse novo conceito. O SPARC64 VI [12] da Fujitsu, possui dois núcleos com capacidade de executar duas tarefas simultâneas em cada um. O POWER-5 da IBM [9], que surgiu como uma melhoria no *dual-core* POWER-4 lançado em 2001, introduziu a capacidade de execução de tarefas simultâneas em dois núcleos. O Montecito da Intel [11] é também um processador com 2 núcleos que suporta a execução de 2 tarefas simultâneas. O CMT SPARC, UltraSPARC T1, codinome Niagara [10] da Sun, tem 8 núcleos com capacidade de executar 4 tarefas simultâneas cada, totalizando 32 tarefas. O Niagara-2 [5] traz 8 núcleos físicos com a capacidade de execução de 8 tarefas por núcleo, totalizando 64 tarefas em execução simultânea.

Esse cenário sugere a continuidade das pesquisas em arquiteturas SMT, haja vista que a tendência das pesquisas e do mercado são os processadores de múltiplos núcleos com a execução de múltiplas tarefas simultâneas. Desta forma, a preocupação com o consumo de potência deste tipo de processador é real e exige esforços de pesquisa.

3. Consumo de potência

O consumo de potência e as questões térmicas são cada vez mais preocupantes nos processadores modernos. Torna-se importante que informações referentes a potência/desempenho sejam visíveis aos projetistas e desenvolvedores em tempo de projeto, pois assim o equilíbrio entre o desempenho esperado e o consumo desejado pode ser equacionado.

Conceitos que a princípio parecem ser sinônimos, na realidade podem assumir diferentes significados conforme o contexto. Energia é comumente definida como a propriedade de um sistema capaz de realizar trabalho. Nesse sentido, energia e trabalho se associam a uma mesma grandeza. Como o trabalho é medido em *joules*, a energia também pode ser definida em *joules*, mas comumente é medida em Wh (Watts*hora).

Potência é a taxa de trabalho realizado ou de energia consumida por unidade de tempo, a qual é medida em Watts. Na computação, o trabalho envolve atividades associadas com a execução de programas (adição, subtração, operações de memória), que implica no fluxo de elétrons para o funcionamento do hardware. Nesse contexto, a potência é a taxa na qual o processador consome energia elétrica ou dissipa na forma de calor na realização dessas atividades.

Esta distinção entre os termos potência e energia é importante, pois as técnicas que reduzem o consumo de potência não necessariamente reduzem o consumo de energia no contexto global. A potência consumida pode ser reduzida pela alteração da freqüência de *clock*, mas se por conta disso o computador aumentar o tempo de execução dos mesmos programas, então o total de energia consumida será maior. Assim, a decisão entre reduzir potência ou energia dependente do contexto.

Em aplicações móveis, a redução de energia é freqüentemente mais importante para o aumento do tempo de vida da bateria, dando uma autonomia maior aos dispositivos que dela dependem. Já para sistemas baseados em servidores, a manutenção da temperatura dentro de limites aceitáveis é mais importante, sendo necessário reduzir a potência instantânea apesar do impacto na energia total. A CPU, caches, memória principal e barramentos compreende à região de tráfego mais intenso e normalmente é o responsável pela maior parte do consumo de potência.

Diversas técnicas são aplicadas nos diferentes níveis do sistema computacional, envolvendo componentes, arquitetura e aplicações, caracterizando o gerenciamento de potência como uma disciplina de grande abrangência e em expansão contínua.

Uma das técnicas que tem sido citada em vários trabalhos é a Escala Dinâmica de Voltagem e Freqüência (DVFS) [3], que explora o fato da freqüência ser proporcional à voltagem, enquanto a quantidade de energia dinâmica requerida para dada carga de trabalho é proporcional ao quadrado da voltagem. Assim, reduzindo-se a voltagem, reduz-se a freqüência que conseqüentemente implicará em um aumento no tempo de execução. Alguns processadores adotam esta técnica na implementação de suas tecnologias, tais como a LongRun do Transmeta Crusoe, PowerNow nos processadores AMD e XScale em processadores da Intel.

A técnica de *gating* de circuitos [22] tem sido aplicada aos estágios do pipeline, permitindo que as partes do pipeline sejam ativadas de forma independente, restringindo a produção de atividades desnecessárias no chip. Esta técnica é tratada em alguns casos como banking [23]. O gating de *clock* é o controle da distribuição do sinal do *clock* pelas diversas regiões do chip, podendo reduzir o nível da voltagem o quanto for necessário até o impedimento por completo da passagem do sinal de *clock* para poupar o consumo em determinado componente que não esteja sendo utilizado em certo momento.

A técnica de *gating* [7] também é utilizada para reduzir o consumo de potência nas unidades funcionais, sendo utilizado um circuito para detectar as condições e decidir quando é apropriado bloquear o fornecimento de energia, que pode ser feito por meio de um único transistor.

Outras formas que abrangem o estudo do consumo de energia e potência estão relacionadas com o emprego de ferramentas computacionais para o projeto e testes de circuitos, e para simulação comportamental e de desempenho. Simuladores arquiteturais que, além da avaliação do desempenho, possibilitam o emprego de métricas relacionadas ao consumo de potência. A próxima seção apresenta os simuladores que estão diretamente relacionados a este trabalho e que serviram de base para a implementação do PowerSMT.

4. Simuladores relacionados

A ferramenta SimpleScalar [2] vem sendo amplamente utilizada por diversos grupos de pesquisa o projeto e avaliação de novas propostas de arquiteturas. O SimpleScalar é um pacote que agrupa simuladores de processadores superescalares baseados no MIPS, compiladores e depuradores, permitindo configurar as principais características arquiteturais. Dentre os simuladores do pacote, o sim-outorder é o mais completo para arquiteturas superescalares.

Como uma modificação do simulador sim-outorder da plataforma SimpleScalar, o simulador SS_SMT [4] foi desenvolvido para simular e avaliar arquiteturas SMT. O SS_SMT permite executar simultaneamente um conjunto de tarefas sobre um mesmo pipeline SMT contendo recursos privados e compartilhados. Todas as facilidades de configuração e execução, existentes no SimpleScalar, foram herdadas para o SS_SMT.

Em se tratando especificamente do consumo de potência, o simulador Wattch [1], também desenvolvido sobre a plataforma SimpleScalar, é uma das principais ferramentas de simulação em nível arquitetural que faz estimativas de consumo de potência. O *Wattch* permite medir o consumo de potência considerando 3 tipos diferentes de técnicas "*pipeline gating*", intituladas CC1, CC2 e CC3.

A técnica CC1 considera que qualquer acesso a um bloco arquitetural consome a potência do bloco todo, mesmo quando o bloco permite mais do que um acesso

simultâneo. A técnica CC2 considera que o consumo de potência em um bloco arquitetural que permite mais do que um acesso simultâneo é proporcional ao número de acessos. A técnica CC3 é uma variante de CC2, mas considera um consumo mínimo de 10% mesmo quando um bloco arquitetural não é acessado. Os autores afirmam que o tipo CC3 produz as estimativas mais próximas dos processadores reais. Estas mesmas técnicas foram herdadas pelo PowerSMT.

O Wattch é uma extensão do sim-outorder que utiliza a biblioteca CACTI [8] para estimar o consumo de potência. A CACTI é uma biblioteca que define e implementa um modelo baseado em formulações para a medição do consumo de potência. A versão inicial da CACTI 1.0 foi desenvolvida com o intuito de possibilitar a modelagem de caches SRAM, considerando que o consumo é calculado em função do tempo de acesso às estruturas da cache. Duas novas versões, a CACTI 2.0 [16] e CACTI 3.0 [17] adicionaram o modelo de área e o modelo de potência, considerando esses modelos na otimização.

A versão 4.0 [19] adicionou ao conjunto anterior o modelo de potência estática (*leakage power*) baseando-se na modificações da versão 3.0 proposta por outras implementações, tais como eCACTI [13] e Hotleakage [24], atualizando ainda a estrutura de circuitos básicos e parâmetros dos dispositivos para melhor refletir os avanços da tecnologia dos semicondutores.

Outros simuladores também são usados para a análise do consumo de potência, tais como o SimplePower [25], PowerTimer [26] e o SoftExplorer [27], mas não interagem com o presente trabalho.

5. PowerSMT

O PowerSMT foi desenvolvido sobre o simulador SS_SMT [4] por meio da adição do modelo de consumo de potência utilizado no Wattch [1], o qual é baseado na CACTI [8]. Esse modelo sofreu alguns ajustes e modificações para se adequar ao novo simulador. O modelo arquitetural implementado pelo simulador SS_SMT [4] permite a execução de *benchmarks* codificados para o conjunto de instruções do SimpleScalar (subconjunto do MIPS). Assim, no contexto do PowerSMT, as tarefas são de fato programas independentes.

Várias tarefas podem ser colocadas para a execução simultânea. Cada tarefa terá o seu próprio contexto de execução e produzirá os seus próprios resultados, muito embora todos estejam sujeitos as mesmas configurações arquiteturais. Ao final da execução de cada tarefa o resultado correto e a semântica do código original de cada tarefa são preservados.

O sistema de cache é estruturado por um conjunto de módulos que são acessados paralelamente, sendo que cada módulo possui um barramento separado e conjunto de bancos internos que são acessados de forma exclusiva entre si, conforme mostra a Figura 1. Uma ou mais tarefas podem compartilhar o mesmo banco, mas os experimentos mostram que o recomendável é que as tarefas estejam em bancos separados para reduzir conflitos significativamente.

A unidade de busca de instruções trabalha segundo estilo round-robin, buscando a cada ciclo instruções de todos os módulos paralelamente, mas de apenas uma das tarefas que estão dentro de cada módulo de cache, de forma a usar todo o barramento do respectivo módulo. As instruções são colocadas em filas privativas as tarefas em execução.

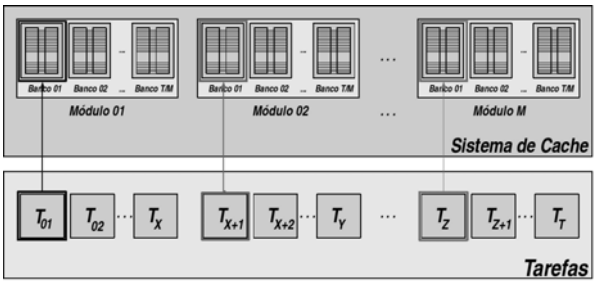

Figura 1: Uso dos bancos pelas tarefas

A decodificação verifica as extremidades de todas as filas de instruções e decodifica um mix de instruções de todas as tarefas simultaneamente, cujo número depende do tamanho do barramento de decodificação. As instruções decodificadas têm suas dependências tratadas e são despachadas às estações de reserva junto as unidades funcionais.

A unidade de remessa seleciona as instruções que estão prontas nas estações de reserva e as remete às unidades funcionais especializadas que estão livres e de acordo com o tipo de instrução. As instruções são executadas pelas unidades funcionais e os resultados alimentam as instruções que aguardam operandos nas estações de reserva para se tornarem prontas para execução. Uma fotografia do pipeline em funcionamento mostrará que a janela de instruções deste processador contém um mix de instruções de todas as tarefas simultaneamente.

O processo de desenvolvimento do PowerSMT deu-se por meio do estudo e mapeamento funcional das ferramentas que serviram de base: SimpleScalar, SS_SMT, CACTI e Wattch. Durante a fase de validação a ferramenta PowerSMT foi experimentada exaustivamente e os resultados da simulação foram

comparados com aqueles obtidos nas ferramentas originais de base.

A versão da biblioteca CACTI utilizada no Wattch é a CACTI 1.0. Esta versão de biblioteca permite somente otimizar os parâmetros orientados ao tempo de acesso às estruturas do pipeline. Os parâmetros repassados à CACTI para o cálculo do consumo de potência, referentes a tecnologia do processo de fabricação, são fixos na CACTI 1.0. Em função destas restrições, a biblioteca CACTI utilizada no PowerSMT foi atualizada para a versão 4.0.

Na versão CACTI 4.0, toda a interação com a biblioteca ocorre por meio de uma única chamada de função, que inclui diversos parâmetros, inclusive o da tecnologia do processo de fabricação. Além disso, os outros parâmetros são calculados dinamicamente na CACTI 4.0 em função do parâmetro de tecnologia. As duas versões foram depuradas e os modelos de área, de *leakage*, de tempo e de potência foram comparados para maior certificação de compatibilidade durante os cálculos feitos em tempo de simulação.

O simulador PowerSMT permite que as estações de reserva (RUU/LSQ) sejam organizadas de forma distribuída ou compartilhada. Outra característica adicionada é a possibilidade de configurar todas as unidades funcionais como sendo de três classes: heterogênea (hetero), homogênea (homo) ou compacta (compact). A organização arquitetural do PowerSMT pode ser vista na Figura 2.

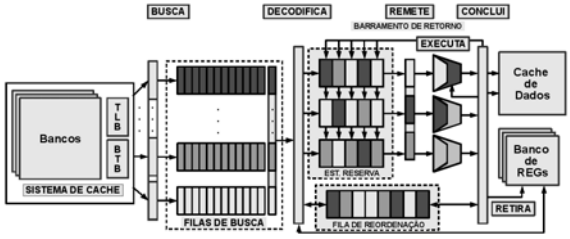

Figura 2: Organização do PowerSMT

Esta dinâmica que foi herdada pelo PowerSMT obrigou a ajustes do modelo de consumo de potência e algumas adaptações e modificações, principalmente nos cálculos dos valores de referência, para estimar cálculos corretos do consumo de potência e para que a atualização e a geração das estatísticas de execução pudessem ser feitas e reportadas também corretamente.

Outra necessidade de alteração no modelo de consumo foi a inclusão do cálculo de valores de referência e a atualização deles para os tipos variados de unidades funcionais do PowerSMT. Além disso, no modelo de consumo do Wattch não estava previsto o cálculo do consumo de potência para a fila de instruções da busca (IFQ) e para o Buffer de Reordenação (ROB), o qual foi corrigido no PowerSMT. Pelo fato da arquitetura SMT ter recursos privativos para cada tarefa, sendo que em várias partes da arquitetura é necessário selecionar qual estrutura utilizar, o cômputo do consumo destas lógicas de seleção (seletores) também foram considerados.

6. Avaliação de desempenho

Nesta seção são apresentados os experimentos que foram realizados com o PowerSMT com o intuito de avaliar algumas das principais características arquiteturais da ferramenta. Os experimentos utilizaram os *benchmarks* do SPEC2000 na composição dos cenários de execução denominados SMT-X, sendo que X indica o número de tarefas simultâneas, podendo ser 1, 2, 4, 8 e 16. Os *benchmarks* usados foram *gzip*, *vpr*, *cc*, *mcf*, *parser*, *vortex*, *bzip*, e *twolf* para aritmética de números inteiros e *wupwise*, *swim*, *mgrid*, *applu*, *mesa*, *art*, *equake* e *ammp* para aritmética de números em ponto-flutuante. Os agrupamentos foram feitos de forma a testar diferentes grupos de *benchmarks*, sendo a média entre eles usada como resultado. Por exemplo, a arquitetura SMT-4 foi testada com 4 grupos de 4 *benchmarks*. Com exceção dos parâmetros que foram variados, para todos os experimentos, o restante do *pipeline* foi configurado em grandes proporções para a obtenção de uma arquitetura ampla, a ponto de não ser ele o motivo de um possível baixo desempenho. Os resultados em termos de consumo de potência são apresentados segundo o estilo CC3.

O primeiro experimento realizado foi sobre à cache de instruções de nível 1 no qual foi variado o tamanho de cada banco da cache e a organização desses bancos em módulos. O tamanho de cada banco variou de 4KB até 4096KB. Foram executadas simulações com 1, 2, 4, 8 e 16 tarefas simultâneas.

Os resultados obtidos para o consumo da cache de instruções de nível 1 são apresentados na Figura 3 em termos de consumo médio de potência em Watts. As médias são calculadas sobre todas as simulações com grupos de *benchmarks* diferentes e sobre todas as possíveis configurações utilizadas usando diferentes números de módulos de cache. A objetivo é apresentar a ferramenta como uma opção flexível para a avaliação de configurações e estruturas arquiteturais, mostrando sua importância na avaliação de desempenho e consumo.

Podemos observar nesta figura que o consumo de potência se eleva menos do que o dobro na medida em que dobramos o tamanho da cache. De fato, olhando os arquivos de *log* das simulações, este aumento é em

média 87%, ou seja, na medida em que dobramos o tamanho da cache de instruções o consumo aumenta 87% na média de todas as configurações avaliadas. O maior pico que observamos é de 149% de aumento.

Sabe-se que existem várias questões associadas ao consumo da cache que não dependem somente do número de entradas. A lógica de decodificação de endereçamento tem um crescimento não linear. Por exemplo, decodificar 8 posições requer um único *bit* a mais no endereçamento do que decodificar 4 posições.

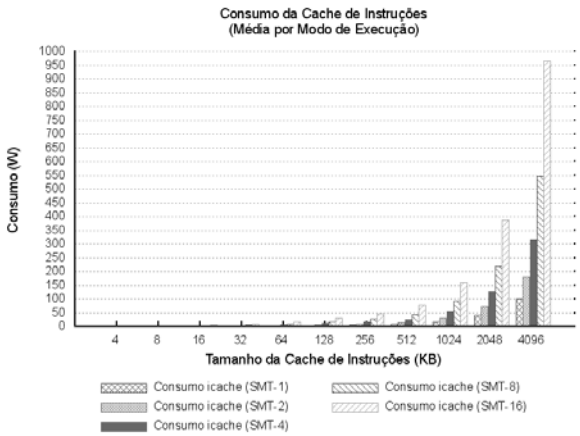

Figura 3: Consumo da i-cache nível 1

É interessante dizer que com o aumento do número de tarefas, ainda que usando o mesmo tamanho de cache, ocorre também um aumento no consumo de potência bastante significativo. Olhando os arquivos de log destas simulações este aumento atinge de 71%, quando dobramos de 8 para 16 o número de tarefas, a 81%, quando dobramos de 1 para 2 o número de tarefas, na média. A justificativa para este fato que a técnica de cálculo do consumo de potência CC3, derivado do Wattch, considera o consumo mediante o acesso as estruturas que estão sendo avaliadas.

Assim, este aumento está diretamente relacionado ao número de acessos a cache quando aumentamos o número de tarefas compartilhando o sistema de cache. A variação do número de IPC (instruções executadas por ciclo) para a mesma simulação é apresentada na Figura 4.

Este gráfico mostra um ponto de saturação no aumento do desempenho pois o desempenho sofre pouco aumento quando dobramos o tamanho da cache, alcançando a média de 6% apenas nas 2 caches menores e nas demais caches não sofre nenhum incremento considerável. Agora, quando dobramos o número de tarefas simultâneas, o desempenho aumenta significativamente independente do tamanho da cache de instruções nível 1, alcançando a média de 78% no aumento do IPC.

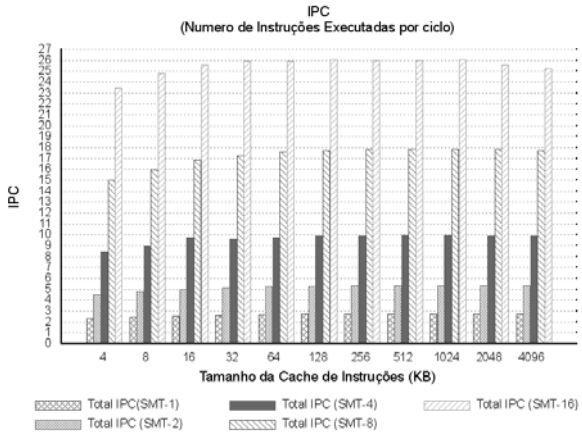

Figura 4: Desempenho da i-cache nível 1

O segundo experimento realizado foi relacionado à organização heterogênea das unidades funcionais. Variou-se a quantidade de unidades funcionais de 1 até 10, para cada um dos 4 tipos contabilizados (IALU, IMultDiv, FPALU e FPMultDiv) simultaneamente. As unidades funcionais de acesso a memória não foram contabilizadas. As Figuras de 5 a 9 mostram os resultados desta simulação para os modelos SMT-X.

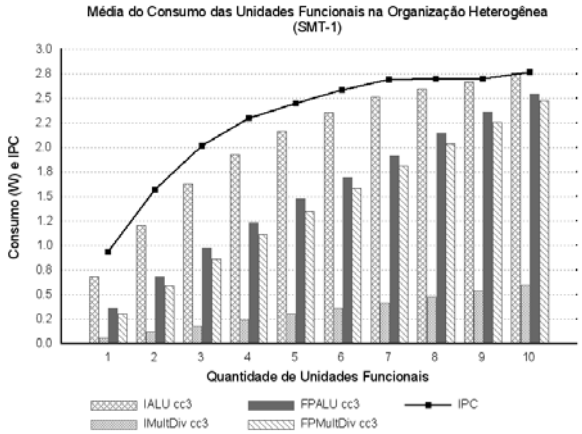

Figura 5: Consumo das UFs no SMT-1

Nestas figuras as colunas indicam o consumo de potência em Watts por tipo de unidade funcional e a curva o IPC global da arquitetura, ambos valores médios entre todas as possíveis configurações. Observe que o consumo total para dada quantidade de unidades funcionais é obtido pela somatória dos respectivas dos valores das colunas. Note também que a curva do IPC está na escala correta, em cada gráfico, possibilitada por uma proximidade coincidente entre os valores do IPC e do consumo de potência obtido.

Entretanto, os gráficos possuem escalas diferentes pois seria difícil visualizá-los em uma mesma escala.

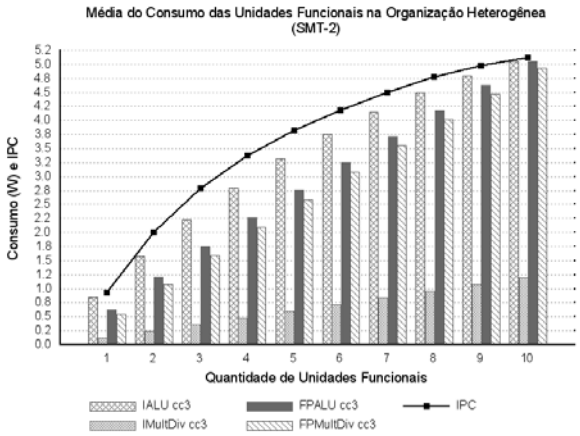

Figura 6: Consumo das UFs no SMT-2

De uma forma geral, percebe-se em todas as situações que o consumo de potência possui um crescimento razoavelmente incremental (linear) para os 4 tipos de unidades funcionais, com alguma distorção no caso da IALU na SMT-1. Percebe-se também que o desempenho obtido cresce mais próximo ao consumo de potência quanto menor for o número de tarefas. Nas arquiteturas com maior número de tarefas, por exemplo SMT-16, o desempenho possui uma taxa de crescimento menor do que a taxa de crescimento do consumo de potência. Obviamente, o aumento do hardware certamente aumenta o consumo de potência, mas não necessariamente aumenta o desempenho haja visto a possibilidade de saturação dos demais recursos.

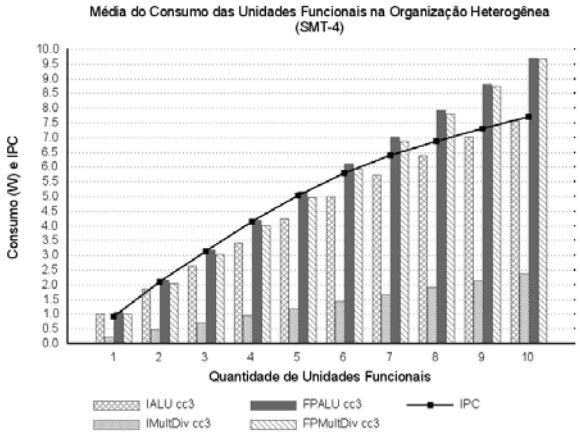

Figura 7: Consumo das UFs no SMT-4

Este crescimento indesejável do consumo de potência nas arquiteturas com muitas tarefas pode ser explicado, em parte, pela forma como o experimento foi realizado, que fixou para todos os tipos de unidades funcionais a mesma quantidade. Entretanto, existem tipos de unidades funcionais que não precisam ter uma quantidade tão grande, até mesmo pelo tipo de instruções que executam e pela quantidade de instruções desses tipos presentes nos códigos dos *benchmarks*. Os valores de consumo que excedem a curva de desempenho poderiam ser melhor ajustados com um número menor de unidades funcionais para estes tipos de instruções com menor ocorrência, já que não influenciam tanto para a melhoria do desempenho, acarretando somente um aumento no consumo.

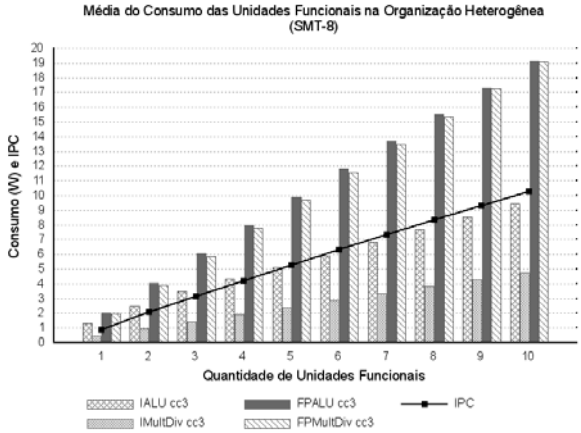

Figura 8: Consumo das UFs no SMT-8

Os experimentos aqui apresentados abordam apenas duas das muitas questões relacionadas às arquiteturas SMT e assim os resultados são parciais. Maiores experimentos são necessários para levar a conclusões mais efetivas sobre o destino deste tipo de arquitetura no referente ao consumo eficiente de energia.

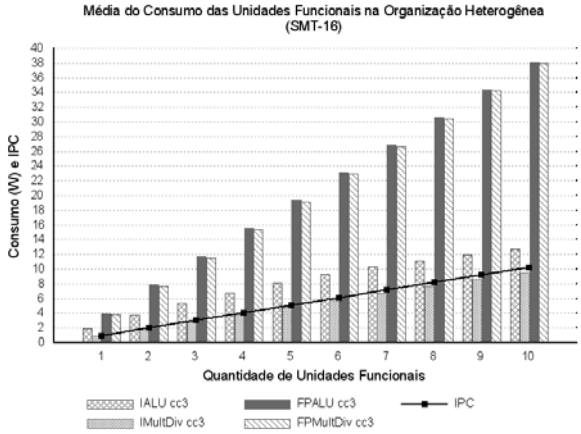

Figura 9: Consumo das UFs no SMT-16

7. Conclusões e trabalhos futuros

A ferramenta PowerSMT permite estudar e avaliar as arquiteturas SMT, ainda na fase de projeto, considerando estimativas de desempenho e de consumo de potência. A ferramenta permite avaliar diferentes modelos arquiteturais e assim buscar maximizar a relação custo/benefício. PowerSMT proporciona a obtenção de resultados de forma rápida e eficiente para arquiteturas SMT, atendendo assim a necessidade de ferramentas para o auxílio nas pesquisas da área.

Os resultados experimentais apresentados neste trabalho e a análise feita sobre os mesmos confirmam a utilidade e importância da ferramenta PowerSMT. Os próximos passos objetivam a inclusão de melhorias tais como o compartilhamento de estruturas que atualmente são privativas as tarefas. Além disso, a clusterização do pipeline de forma a permitir o uso de técnicas de desligamento e ligamento automático de áreas não usadas são de grande interesse na continuidade deste.

8. Agradecimentos

Agradecemos à Fundação Araucária e à CAPES pelo apoio financeiro.

9. Referências

[1] BROOKS, D. et al. Wattch: A framework for architectural-level power analysis and optimizations. In PROC. OF 27TH ANN. INT'L SYMP. COMPUTER ARCHITECTURE, pages 83-94. IEEE Computer Society Process, Los Alamitos, USA, 2000.

[2] BURGER, D., AUSTIN, T. M.. The SimpleScalar Tool Set, Version 2.0. University of Wisconsin-Madison Computer Sciences Department. TR#1342, Jun. 1997.

[3] FLAUTNER K., and MUDGE, T. Vertigo: automatic performance-setting for Linux. SIGOPS OPER. SYST. REV. 36, SI (DEC. 2002), 105-116.

[4] GONÇALVES, R. A. L. et al. A Simulator for SMT Architecture: Evaluating Instruction Cache Topologies, SBAC-PAD, São Pedro, 2000.

[5] GROHOSKI, G., Niagara-2: A Highly Threaded Server-on-a-Chip, in Hot Chips 2006.

[6] HIRATA, H. et al. An Elementary Processor Architecture with Simultaneous Instruction Issuing from Multiple Threads. Proceedings of the 19th ISCA.

[7] HOMAYOUN, H. et al. "Functional units power gating in SMT processors," Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on , vol., no., pp. 125-128, 24-26 Aug. 2005.

[8] JOUPPI, N. and WILTON, S.. An enhanced access and cycle time model for on-chip caches. Technical Report TR-93-5, Compaq WRL, July 1994.

[9] KALLA, R. et al. "IBM Power5 chip: a dual-core multithreaded processor," Micro, IEEE , vol.24, no.2, pp. 40-47, Mar-Apr 2004.

[10] KONGETIRA, P. et al. "Niagara: a 32-way multithreaded Sparc processor," Micro, IEEE , vol.25, no.2, pp. 21-29, March-April 2005.

[11] MCNAIRY, C.; BHATIA, R., "Montecito: a dual-core, dual-thread Itanium processor," Micro, IEEE , vol.25, no.2, pp. 10-20, March-April 2005.

[12] MARUYAMA, T., "SPARC64 VI: Fujitsu's Next Generation Processor," in Microprocessor Forum, 2003.

[13] MAMIDIPAKA, M., DUTT, N. eCACTI: An Enhanced Power Estimation Model for On-chip Caches. University of California Irvine Center for Embedded Computer Systems Technical Report TR-04-28, Sept. 2004.

[14] MARR, D. T. et al. Hyper-Threading Technology Architecture and Microarchitecture. Intel Technology Journal, 6(1), February 2002.

[15] OLUKOTUN, K. et al. The case for a single-chip multiprocessor. SIGPLAN N. 31, 9 , 2-11, Sep. 1996.

[16] REINMAN, G., and JOUPPI, N.. "CACTI 2.0: An integrated cache timing and power model," WRL Research Report 2000/7, Feb.2000.

[17] SHIVAKUMAR, P. e JOUPPI, N. P.. CACTI 3.0: An Integrated Cache Timing, Power, and Area Model., TR Compaq Computer Corporation August, 2001.

[18] SPRACKLEN, L.; ABRAHAM, S.G., "Chip multithreading: opportunities and challenges," High-Performance Computer Architecture, HPCA, Feb. 2005.

[19] TARJAN, D. et al. CACTI 4.0, HP Laboratories Palo Alto, HPL-2006-86, June 2, 2006.

[20] TULLSEN, D. M. et al. Simultaneous Multithreading: Maximizing On-Chip Parallelism, ISCA, Italy. 1995.

[21] YAMAMOTO, W.; NEMIROVSKY, M. Performance Estimation of Multistreamed, Superscalar Processors, Proceedings of the Hawaii International Conference on System Sciences, January, 1994.

[22] MANNE, S. et al. Pipeline gating: Speculation control for energy reduction. ISCA, Barcelona, 1998.

[23] PARIKH, D. et al. Power issues related to branch prediction. Proceedings of the 8th International Symposium on High-Performance Computer Architecture, Feb. 2002.

[24] ZHANG, Y. et al. HotLeakage: A temperatureaware model of subthreshold and gate leakage for architects. TR-CS-2003-05, University of Virginia, Dept of Computer Science, March 2003.

[25] YE, W. et al. The Design and Use of SimplePower: A cycle-accurate energy estimation tool. Proceedings of the Design Automation Conference, June 2000.

[26] BROOKS, D. et al. Microarchitectural-Level Power-Performance Analysis: The PowerTimer Approach. IBM Journal of Research and Development, V.47, Nov. 2003.

[27] LAURENT, J. et al. Functional level power analysis: an efficient approach for modeling the power consumption of complex processors. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Feb. 2004.

[28] MINHOLI, Diego H. et al. Correlacionando Desempenho e Potência em Processadores Superescalares: Analisando Cache e Janela de Instruções In: II Encontro Paranaense de Computação, Cascavel. Unioeste, Ago. 2007.