



ESTIMADOR E CARACTERIZADOR DE CONSUMO DE ENERGIA PARA SOFTWARE EMBARCADO

Francisco Coelho da Silva¹ /INDT

Thales Ruano Barros de Souza¹ /FIT

Luan da Silva Serrão²

Leonardo Afonso Reis Milward Junior³

Gledyson Cidade da Costa³

Maria José Castro Ferreira³ / ICCT

¹Mestre em Engenharia Elétrica, pela UFAM ;

²Discente do curso de Engenharia Elétrica, pelo UNINORTE;

³Engenheiro Eletricista, pelo UNINORTE.

RESUMO

Consumo de energia nos últimos anos tornou-se um aspecto importante em projetos de sistemas embarcados. A produção e utilização em larga escala dos dispositivos móveis tem imposto uma variedade de restrições a estes sistemas, como: peso, tamanho, tempo de vida útil da bateria e funcionalidades complexas. Dispositivos móveis operam sob uma fonte de energia limitada cuja autonomia e tempo de vida útil estão diretamente relacionados ao consumo de energia das aplicações. Estas questões contribuíram para incluir o



consumo de energia como métrica de qualidade no projeto de sistemas embarcados. Este trabalho tem como objetivo propor uma abordagem de medição, estimação e comparação do consumo de energia de código de programas escritos em linguagem ANSI-C, baseados em ensaios de códigos previamente escolhidos com características de consumo de energia e no tempo de execução. Para dar suporte à abordagem uma ferramenta de estimação chamada PESTI foi estendida para atender múltiplos cenários probabilísticos. Um programa escrito em linguagem ANSI-C é embarcado em um processador da família ARM7 é inserido neste programa *flags* de sinalização para *start* e *stop*, para medirmos o consumo de energia e o tempo de execução do código, um hardware chamado de caracterizador de consumo de energia auxiliará na medição em tempo real de execução do trecho do código. A ferramenta de estimação já estendida para a atribuição de múltiplos cenários probabilísticos, para um programa escrito em ANSI-C anotado e traduzido em um gráfico de fluxo de controle. Este modelo é simulado probabilisticamente utilizando o Método de Monte Carlo. Validamos a abordagem proposta executando um conjunto de experimentos para mostrar a viabilidade da extensão da ferramenta de estimação e o caracterizador, que em conjunto, tornarão as estimativas de consumo de energia viabilizável.

Palavras-chave: Sistemas Embarcados, Consumo de Energia, Tempo de Execução, Grafo de Fluxo de Controle, Múltiplos Cenários Probabilísticos, Método de Monte Carlo, Estimativas, ANSI C, Ferramenta PESTI, Caracterizador de Consumo de energia para software embarcado.

INTRODUÇÃO

A explosão do uso dos dispositivos móveis, levaram muitos grupos de pesquisa a busca de novas forma de diminuir o consumo de energia desses diversos sistema além de otimizarem este consumo com novos hardware e novos algoritmos nos projetos de **Sistemas Embarcados**.

Com a miniaturização de componentes eletrônicos e dispositivos cada vez menores e o aumento com certeza das linhas de códigos embarcados em processadores, percebe-se a necessidade de compreender melhor a variedade de restrições a que estes sistemas são submetidos, como: peso, tamanho, tempo de vida útil da bateria e funcionalidades complexas baseadas em código embarcado. Esses dispositivos portáteis trabalham sob uma fonte de energia limitada cuja autonomia e tempo de vida útil está diretamente relacionado ao **Consumo de Energia** das aplicações. Estes fatores mencionados anteriormente contribuem consideravelmente para o aumento do consumo de energia nos dispositivos embarcados, tornando este fato uma métrica de qualidade no projeto de sistemas embarcados.

O termo **Sistema Embarcado** pode ser utilizado para qualquer sistema digital que esteja incorporado a outro com o intuito de acrescentar ou aperfeiçoar funcionalidades. Com o avanço da tecnologia portátil e embarcada, principalmente da microeletrônica, e o barateamento dos sistemas microprocessados, fizeram com que ocorresse uma explosão na utilização deste tipo de sistema, nas mais diversas áreas da vida humana [3], como resultado desta revolução veio o desenvolvimento em larga escala de sistemas digitais com funcionalidade específica. Entretanto as técnicas e paradigmas existentes para desenvolvimento de aplicações não focavam essa nova classe de sistema, dessa forma, para desenvolver sistemas competitivos, ou seja, de baixo custo e boa qualidade, métodos de projeto precisaram ser criados ou adaptados. Devido a isso, grande parte das pesquisas, ainda hoje, visa à criação ou a melhoria de técnicas já desenvolvidas.

Atualmente, consumo de energia, particularmente em sistemas embarcados, é ponto de discussão em pesquisas em indústrias e em pesquisas acadêmicas. Essas discussões são respaldadas na massificação do uso de dispositivos móveis e a exigência cada vez maior do mercado para que os mesmos tenham grande potencial computacional e maior grau de autonomia.

Devido a essas tais exigências, uma das métricas de qualidade passou a ser o consumo de energia no contexto do desenvolvimento de projetos de sistemas embarcados. Além disso, o consumo de energia tem impacto na forma de encapsulamento do sistema, implicando no custo final do produto.

Um sistema embarcado tem como fontes de consumo três componentes: o *software*, o *hardware* e os barramentos. Entretanto, os maiores responsáveis pelo consumo de energia do sistema são o *software* e o *hardware*, pois suas descrições estão diretamente vinculadas ao padrão comportamental do sistema [3]. Neste projeto de pesquisa o interesse está concentrado em mecanismos de avaliação de consumo de energia estabelecidos pelo *software*, pois o mesmo determina um padrão de execução e, conseqüentemente, de consumo sob o processador no qual o executa [24], neste sentido a arquitetura ARM7 é uma das mais novas e pouco estudada. A Figura 1 mostra um diagrama de idealização do processador ARM7 em diversas aplicações, devido à sua grande gama de dispositivos incorporados a um único chip e suas respectivas aplicações dedicadas.

Neste trabalho iremos estender o trabalho desenvolvido por Carlos Mar[2], a ferramenta PESTI, que é algorítmico de consumo de energia e estimativa de desempenho para sistemas embarcados. A ferramenta é baseada em um modelo probabilístico que representa um gráfico de controle de fluxo do sistema, que durante a execução da sua simulação é executada através de sua entrada um código escrito em ANSI C anotado, onde seus resultados aumentaram consideravelmente a velocidade de simulação dos códigos executados [40].

Enfim a ferramenta estendida suportará múltiplos cenários probabilísticos, os quais somente eram possíveis realizar se fossem realizadas um a um.

ferramenta para fazer as análises necessárias das inferências para que se tenha a estimativa de consumo em potência.

Estimativa por simulação (área cinza) - A simulação de circuitos modelados, em seguida, são realizadas análises dos padrões de formas de ondas geradas pelos códigos embarcados na plataforma trabalhada e por fim gerando a estimativa de consumo expressado em potência.

Estimativa de consumo por caracterização (área amarela) - é uma linha pouco explorada nos trabalhos relacionados, o que torna este trabalho uma contribuição a esta linha que pesquisa.

No entanto pode-se usar o modelo de consumo de energia no qual a ferramenta é baseada em duas formas. A primeira é por meio de um processo de caracterização energética do conjunto de instruções do processador que pode ser encontrado em [24] ou [3]. A segunda é através de *datasheets* fornecidos pelo fabricante da plataforma, quando estes disponibilizam tais informações. Este passo é executado apenas uma vez e está vinculado ao processador escolhido para a caracterização.

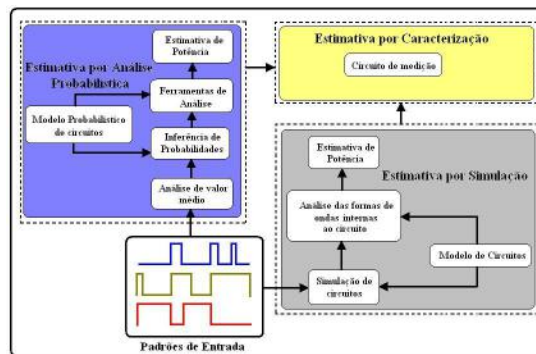


Figura 2 - Estimativas baseadas em Simulação, probabilidade e caracterização.

Com o intuito de garantir que o projeto proposto seja executado com êxito e os resultados esperados e metas definidas sejam alcançados, a seguinte metodologia será utilizada como forma de nortear sua execução.

PROJETO DO CARACTERIZADOR

O projeto do caracterizador está baseado em monitora a corrente elétrica consumida pelo processador ARM7 LPC2148 conforme a Figura 3 quando o mesmo executar o código adaptado e embarcado em seu núcleo de processamento, o processador está acoplado ao kit de desenvolvimento e posto em um arranjo de forma que a alimentação DC passasse por apenas um caminho, de acordo com a Figura 4.



Figura 3- Adaptador soquete com processador ARM7 LPC2148.



Figura 4 - Kit de desenvolvimento com processador ARM7 LCP2148 - fabricado pela NXP.

O circuito de monitoramento de corrente com características *Shunt*, é composto de um amplificador de instrumentação INA 138 e disposto conforme a Figura 5.

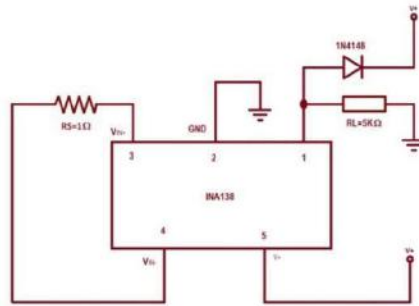


Figura 5 – Esquema de polarização do Amplificador Operacional de Instrumentação – INA138.

SOFTWARE DE DESENVOLVIMENTO DE FIRMWARE

A Keil Software desenvolveu a IDE MicroVision, que permite o desenvolvimento de aplicações embarcadas em C e Assembly no mesmo código fonte.

MicroVision 3 é a interface de desenvolvimento e compilador oferecido pela Keil Software, também fabricante do kit de desenvolvimento MCB2130, através dessa ferramenta é possível criar os projetos, subdividir os projetos em unidades e gerar o código HEXA para ser usado no processador ARM. A IDE foi desenvolvida para ser executada sobre o sistema operacional Windows e conta com um compilador C e macro assembler (KEIL, 2008). Nessa ferramenta é necessário configurar o hardware que está sendo utilizado, assim como as variáveis de compilação, enquanto que o restante dos seus itens se assemelha a outras ferramentas gráficas de desenvolvimento.

Visualizando da Figura 6, que ilustra a IDE MicroVision, é possível observar a sua divisão gráfica de tarefas. Na parte superior encontra-se o menu principal e logo abaixo é possível ter acesso aos botões de acesso rápido. No lado esquerdo tem-se como principal função manter aos olhos do programador a estrutura e organização do seu projeto. A parte central e maior é destinada a escrita do código fonte. Esse quesito possui vários recursos como: Minimizar e Maximizar as funções, adicionar pontos de parada do código fonte nas simulações, linhas de erros assinaladas.



Figura 6 – IDE de desenvolvimento Micro Vision da Keil Software.

SISTEMA DE MEDIÇÃO DE CORRENTE DE CONSUMO

O resistor foi polarizado em um ponto estratégico do circuito de medição e em série com o *jump* de alimentação no Kit de Desenvolvimento de modo a monitorar o consumo de corrente do processador.

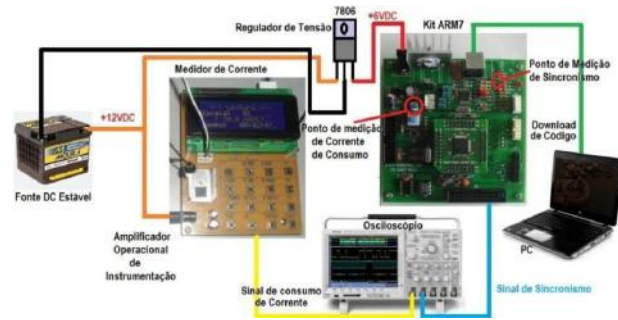


Figura 7 – Esquema do sistema de Caracterização e Medição de Consumo de Energia.

O sistema de caracterização esquematizado na Figura 7 mostra todos os elementos pertencentes ao processo de medição de consumo de energia.

- **Fonte Estável** – De origem química para garantir a ausência do ruído de 60Hz, espúrio e a estática, tem a função de alimentação DC para o conjunto de hardware (Amplificador e o Kit ARM);
- **Kit de medição Amplificador Operacional de Instrumentação** – Projetado com a sensibilidade de variação de corrente consumida pelo processador ARM7 LPC2148 durante a execução dos testes;
- **Regulador de Tensão 7806** – Este componente se fez necessário por conta de segurança do circuito de medição e proteção do Amplificador operacional de instrumentação;
- **Kit de desenvolvimento ARM7** – Desenvolvido de acordo com a necessidade deste trabalho, com uma entrada específica para uma resistência *shunt*;
- **Osciloscópio** – Equipamento utilizado neste trabalho com um poderoso processamento de sinais e uma gama excelente de funções para tratamento e exibição de sinais;
- **PC** – Necessário para a compilação dos códigos ANSI C: Bolha, CRC e QURT, assim como o embarque dos mesmos códigos no processador ARM7.

EXPERIMENTOS E RESULTADOS

Neste capítulo apresenta-se as devidas considerações para a montagem do ambiente real de testes, medidas e materiais além dos equipamentos usados na execução dos testes, considerações para se analisar os resultados obtidos no hardware e com a aplicação da ferramenta PESTI nas estimativas de consumo de energia e tempo de execução dos algoritmos de Ordenação em Bolha, CRC e QURT.

Em um cenário ideal estes experimentos foram executados sobre uma plataforma de hardware real através da execução de múltiplos cenários, da mesma forma que fora implementadas por Oliveira Junior et. al. [3], com a diferença que em seu trabalho Oliveira Junior [3] usa uma plataforma 8051. As métricas coletadas nos experimentos foram consumo de energia e tempo de execução.

Os objetivos destes experimentos foram:

- Validar a abordagem proposta
- Comparar os resultados e tempo de simulação entre a ferramenta PESTI e o caracterizador para a mesma plataforma de hardware (ARM7).

Os experimentos foram divididos em 3 passos:

- Simulação dos algoritmos na ferramenta PESTI,
- Caracterização do cenário probabilístico para o algoritmo bolha, CRC e QURT
- Caracterização dos códigos embarcados no processador ARM.

Os experimentos foram conduzidos sobre AMD Turion(tm) II Dual Core Mobile M500, 2.20GHz, 4GB de RAM SO Windows 7 32 bits e com SO Linux e no Kit de Desenvolvimento ARM o qual usa um processador ARM7 processador ARM7 LPC2148 Figura 3.

ADAPTANDO UM AMBIENTE CONTROLADO PARA TESTES

Para a realização de ensaios e uma coleta de dados confiáveis, foi necessário criar um ambiente um tanto controlado conforme a figura 8 , considerando certas variáveis como, ruído de 60Hz, a temperatura do ambiente de testes, humidade ar e cargas eletroestática, no entanto não foi possível controlar com mais eficácia, mas foi possível monitorá-la a todo momento durante a execução dos teste.

O ambiente ideal para ensaios do tipo realizado neste trabalho requer um investimento considerável, no entanto por falta de recursos, algumas considerações e recomendações foram levadas em conta para que fossem realizadas as medidas de consumo de energia no ARM7 com a presença mínima de variáveis elétricas que poderiam interferir nas e mascarar os resultados deste trabalho.

As mantas dissipativa, luvas e pulseiras antiestáticas também foram utilizado para prover resultados mais confiáveis, e a utilização de um termo higrômetro mostrado na figura 9 para o monitoramento da humidade do ambiente e a temperatura.

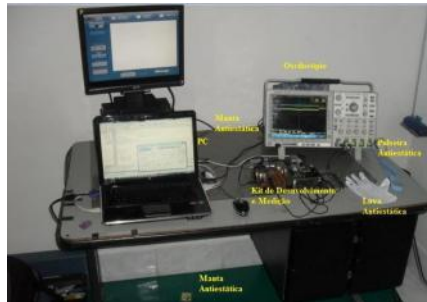


Figura 8 - Ambiente de Ensaios.



Figura 9 - Termo Higrômetro - monitora a temperatura e a humidade.

Fonte: Site Minipa

HARDWARE USADO PARA O PROCESSO DE MEDIÇÃO DE CONSUMO DE ENERGIA

O projeto de desenvolvido para o monitoramento de consumo de energia devido ao código embarcado em um processador, neste caso o ARM, está baseado na escolha dos componentes com características *shunt*, ou seja, dispositivos que trabalham com sensibilidade a passagem de corrente elétrica.

O INA 138 *High-Side Measurement CURRENT SHUNT MONITOR*, componente desenvolvido pelo fabricante *Burr-Brown Product from Texas Instruments*, possui características particulares e essenciais para sua escolha e utilização neste projeto, tendo aplicações nas mais diversas áreas de sistemas eletrônicos e particularmente em sistemas embarcados para monitoramento de corrente de baixo valor nominal.

Na Figura 10 temos um circuito elétrico do monitor de corrente elétrica que será usado no projeto do caracterizador para mensurar o consumo da plataforma escolhida, no caso ARM.

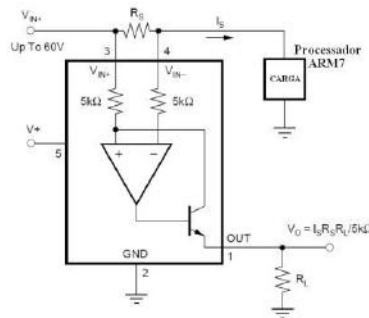


Figura 10 - Diagrama elétrico do monitor *Shunt* de corrente - caracterizador

PROCESSO DE ENSAIO PARA MEDIÇÃO DE CONSUMO DE ENERGIA

Em relação à técnica de medição, o sistema caracterizador teve que lidar com questões básicas:

- ☑ A definição da resistência *shunt* usado em conjunto com o amplificador de instrumentação,
- ☑ As questões do ambiente de testes,
- ☑ Adaptação do código C para serem embarcados no processador ARM7;
- ☑ A forma da taxa de amostragem do osciloscópio;

Em relação à resistência *shunt* sua definição foi feita com base na lei de Ohm e na polarização do amplificador de instrumentação, de forma que não interferisse na medição do consumo de corrente e de forma coerente.

O ambiente de teste foi no máximo do possível controlado, sendo utilizados artifícios de projeto e de execução de testes controlados: a substituição da tensão AC da rede por uma bateria de carro de valor nominal de 12V aproximadamente, isso fez com que reduzisse altamente a possibilidade de interferência espúrios e ruído de 60Hz nas medições e também o uso de forma estreitamente obrigatória de acessórios antiestáticos (Luvas e Mantas Dissipativa) para a realização dos ensaios e manuseio das placas e equipamento.

Sobre a adaptação do Código C para upload no processador, foi trabalhoso, pois o código que fora gerado para o estimador não é possível rodar no processador diretamente, então foi feito de forma criteriosa o isolamento do trecho do código que se quer medir.

O osciloscópio utilizado para a medição foi o DPO 4034 Figura 11 fabricado pela Tektronics, cujas principais características são: Larguras de banda de 1 GHz, 500 MHz e 350 MHz, 4 canais de medição, taxas de amostragem de até 5 GS/s em todos os canais analógicos, comprimento de registro de 10 Megapoints, taxa de exibição de 50.000 formas de onda/segundo e porta de dispositivo USB 2.0 para controle do osciloscópio com o protocolo USBTMC.



Figura 11 - Osciloscópio usado

ANÁLISE DOS DADOS COLETADOS NO PROCESSO DE MEDIÇÃO DE CONSUMO DE ENERGIA

A análise dos resultados é um dos passos mais importante de qualquer trabalho, em que os dados coletados devem ter: coerência, obedecer aos critérios estabelecidos e mostrar o comportamento do evento em questão.

A Figura 12 mostra um dado coletado durante um dos ensaios de medição do código, o sinal em amarelo no canal 1 do osciloscópio é o valor da corrente que o processador ARM7 está consumindo durante a execução do código, já o sinal em azul no canal 2 do osciloscópio é o sincronismo e mostra o tempo de execução do código embarcado no ARM7.

As pontas de prova do osciloscópio foram fixadas de em pontos estratégicos do Kit ARM e da placa de medição de consumo de energia conforme a Figura 13.

Esse procedimento foi realizado para todos os ensaios realizados neste trabalho, sendo claro adequado a cada tipo de ensaio.

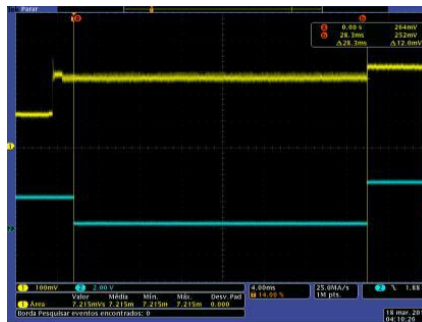


Figura 12- Dado coletado em um ensaios.



Figura 13 - Kits com as pontas de prova do osciloscópio fixadas em pontos estratégicos.

CÓDIGOS ANSI C USADOS NOS ENSAIOS

Foram realizados experimentos para validar o processo de caracterização proposto neste trabalho usando códigos a fim de se explorar os padrões comportamentais dos mesmos e os experimentos que ilustram casos típicos de códigos em sistemas embarcados e em todos eles é feita uma análise de estimativa de consumo de energia.

Os resultados são apresentados através de valores médios para cada cenário e os algoritmos utilizados foram os desenvolvidos pela Motorola e são chamados de *Power Stone Benchmark*[21], esse algoritmos possuem a particularidade de terem sido desenvolvidos especificamente com o objetivo de exercitar diferentes aspectos de consumo de energia em sistemas embarcados especificados em ANSI C.

Código BOLHA

O algoritmo *bubble sort*, ou ordenação por flutuação (mais conhecido como "bolha"),

é um algoritmo de ordenação dos mais simples. A ideia deste algoritmo é percorrer o vetor diversas vezes, a cada passagem fazendo flutuar para o topo o maior elemento da sequência. Essa

movimentação lembra a forma como as bolhas em um tanque de água procuram seu próprio nível, e disso vem o nome do algoritmo e mostrado no conjunto de Apêndice de trabalho.

Códigos CRC e QURT

CRC é um algoritmo para calcular *Cycle Redundant Checking*, ou seja, verificações do conteúdo de uma determinada série de dados são usados na verificação de dados em barramentos de comunicação influenciados por ruídos.

Já o QURT é um tipo de código para embaralhar dados em um *array* e, para isso, utiliza a função de raiz quadrada para calcular variações nesse *array*. Usando este código, os dados de entrada podem ser variados para se atingir diferentes valores de saída, pois isso funciona como um codificador de dados, usados em ambientes de transição de dados, usa vários *arrays* e operações aritméticas com apenas duas expressões condicionais.

Além disso, são algoritmos que resolvem problemas do mundo real. Estes experimentos seguiram o mesmo método de experimentação do algoritmo de Ordenação em Bolha.[2]. E é mostrado no conjunto de Apêndice de trabalho.

GRÁFICOS DOS RESULTADOS E COMPARATIVOS

Esta seção apresenta os resultados da abordagem proposta aplicada aos códigos bolha, CRC e QURT. Os algoritmos foram selecionados por tratar-se de algoritmos que resolvem problemas do mundo real.

Com os cenários probabilísticos determinados, seguimos o processo de estimação e caracterização.

- a) Numero mínimo de simulações: 1.000
- b) Intervalo de confiança: 0.95
- c) Erro máximo permitido: 0.03

Resultados – Código BOLHA – PESTI-1

Os códigos bolha embarcados e executados no processador ARM7 apresentaram um consumo de energia significativo, mas coerente em relação aos estimados na ferramenta PESTI.

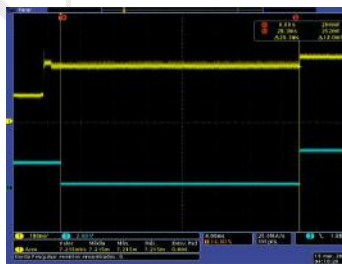


Figura 14 – Dado medido coletado - Bolha - Sinal característico.

Os gráficos a seguir ilustram o comportamento energético do código bolha estimado e medido. Figura 15 nota-se que o consumo do código no processador é na ordem de μJ enquanto que no estimador está na ordem de ηJ , também é possível notar que ocorre certa fidelidade nos dados apresentados.

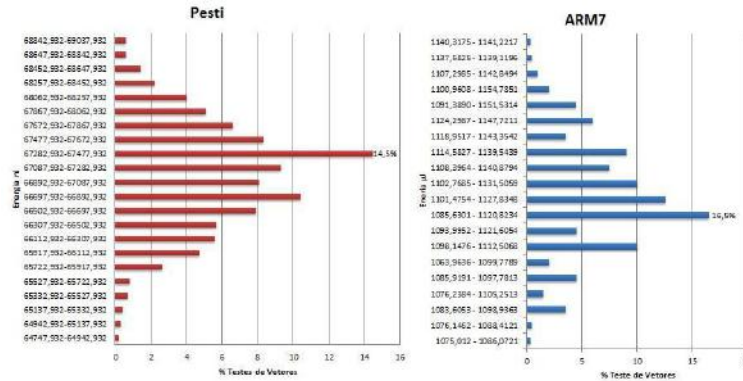


Figura 15 - Gráfico comportamental de consumo energético PESTI e ARM7- Bolha.

Esta diferença entre os valores medidos e estimados é observado na Figura 16 o gráfico comparativo de valores médio de consumo energético entre o ARM7 e a ferramenta PESTI, sendo o maiores valores mais frequente de consumo da PESTI ficaram entre 67287,932 e 67477,932 ηj e 14,5% vetores, e os valores mais frequentes do caracterizador ficaram entre 1085,6301 e 1120,8234 μj e 1,5% dos vetores.

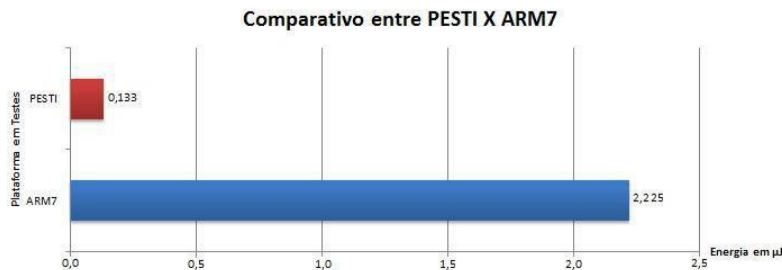


Figura 16 - Gráfico comparativo de consumo energético ARM7 X PESTI - Bolha.

7.8.2 Resultados - CRC PESTI-1

Neste item os códigos embarcados no processador ARM7 são os CRC's, apresentaram um consumo de energia também significativo coerente com os estimados na ferramenta PESTI.

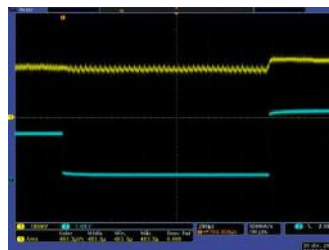


Figura 17 - Dado medido coletado - CRC - Sinal característico.

Os seguintes gráficos ilustram o comportamento e energético do código CRC estimado e medido Figura 18, sendo os maiores valores mais frequente de consumo da PESTI se dividiram em duas faixas de consumo entre 78990,334 - 82816,334 ηj e 48,5%, 2470,334 e 6296,334 ηj e 51,5% vetores, e os valores mais frequentes do caracterizador ficara entre 47,877 - 51,35 44,5% dos vetores e 46,225 - 47,872 μj e 55,5% dos vetores.

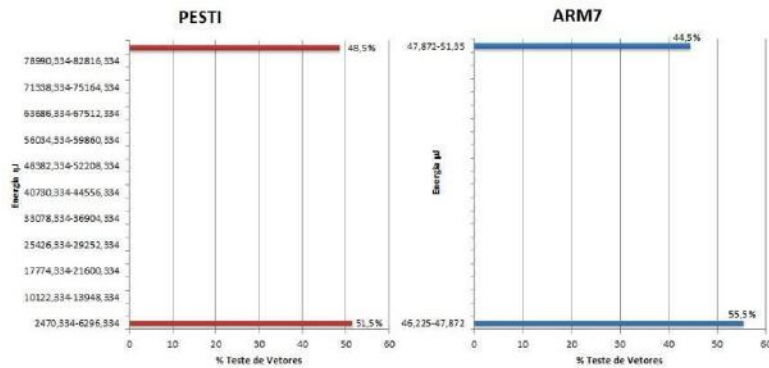


Figura 18 - Gráfico comportamental de consumo energético PESTI e ARM7- CRC.

E na Figura 19 o gráfico comparativo de valores médio de consumo energético entre o ARM7 e a ferramenta PESTI.

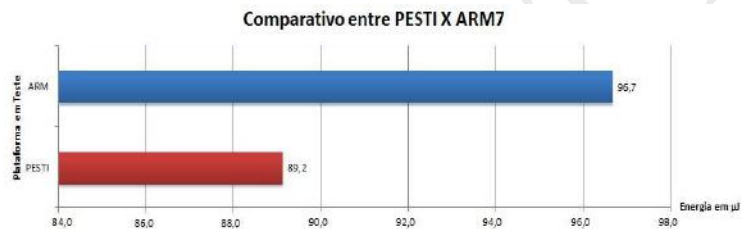


Figura 19 - Gráfico comparativo de consumo energético ARM7 X PESTI - CRC.

Resultados - QURT PESTI-1

E para este tópico os códigos embarcados no processador ARM7 são os QURT's, apresentaram um consumo de energia também significativo coerente com os estimados na ferramenta PESTI.

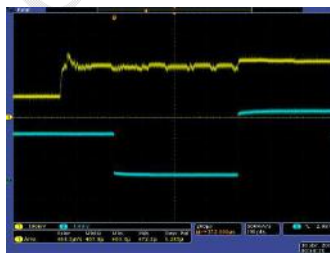


Figura 20 - Dado medido coletado - QURT - Sinal característico.

Os seguintes gráficos ilustram o comportamento energético do código QURT estimado e medido Figura 21 apresentando os maiores valores e mais frequente de consumo da PESTI: entre a faixa de 308,228 - 394,228 ηj e 44,54% e para o caracterizador ficaram entre 46,645 - 49,145 µj e 51,5% vetores, e os valores mais frequentes do caracterizador ficará entre 47,877 - 51,35 e 50,5% dos vetores.

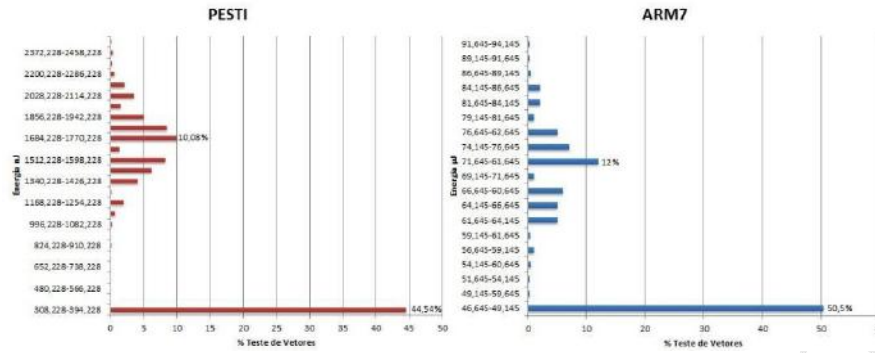


Figura 21 – Gráfico comportamental de consumo energético PESTI e ARM7- QURT.

Finalizando a Figura 22 mostra o gráfico que compara de valores médios de consumo energético entre o ARM7 e a ferramenta PESTI para os mesmos códigos do QURT

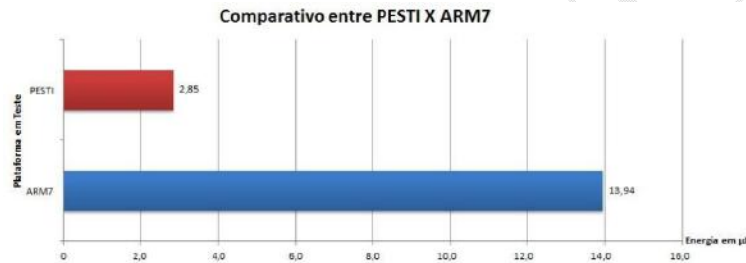


Figura 22- Gráfico comparativo de consumo energético ARM7 X PESTI – QURT.

Gráficos e Resultado – PESTI - Múltiplos

Nesta seção é apresentado os resultados da abordagem proposta aplicada aos códigos bolha, CRC e QURT, usando a ferramenta PESTI estendida de acordo com a proposta deste trabalho. Para os Múltiplos Cenários todos os códigos foram estimados numa faixa probabilística escolhida aleatoriamente, a faixa escolhida foi de 0.45 a 0.55 com um passo a passo de 0.01, perfazendo uma faixa com 11 cenários e 1000 iterações para a ferramenta PESTI num total de 11000 vetores testados.

Com os cenários probabilísticos determinados, seguimos o processo de estimação e caracterização.

- Número mínimo de simulações: 1.000
- Intervalo de confiança: 0.95
- Erro máximo permitido: 0.03

Para cada cenário a ferramenta gera um conjunto de dados no formato CSV, ou seja, um conjunto de dados de 11 arquivos CSV, preparado para a geração de gráficos e todos esses dados foram plotados e são mostrados nos tópicos a seguir.

Resultados – BOLHA - PESTI Múltiplos

O código bolha quando submetido ao processo de estimação em múltiplos cenários se comporta energeticamente conforme o gráfico da Figura 23.

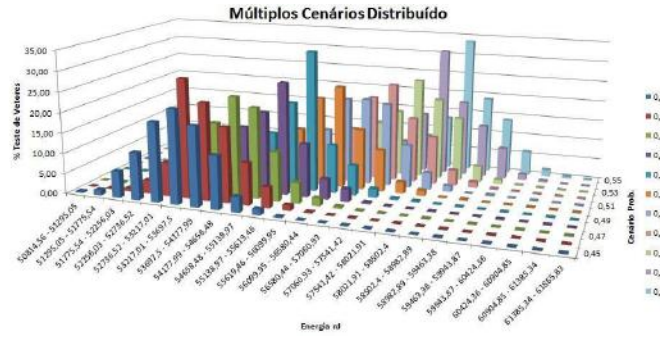


Figura 23 – Gráfico Múltiplo Cenário probabilístico Distribuído – BOLHA.

Para um melhor entendimento os dados da Figura 26, foram tratados e mostrados na sua forma de média e é mostrado na Figura 24.

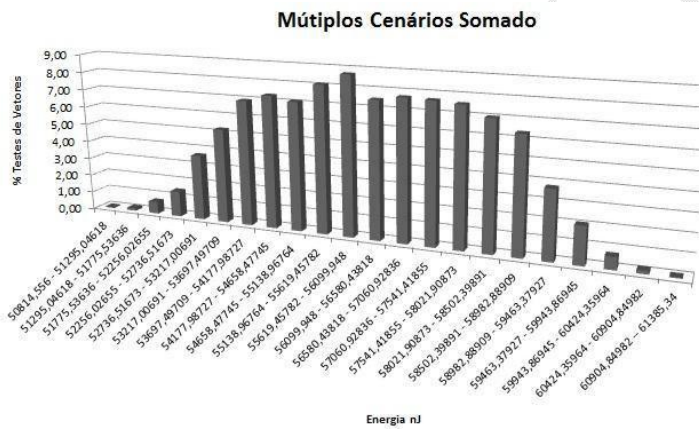


Figura 24 – Gráfico Múltiplo Cenário probabilístico Somado- BOLHA.

Resultados – CRC - PESTI Múltiplos

O código CRC tem seu comportamento mostrado no gráfico da Figura 25, quando submetido aos múltiplos cenários da ferramenta PESTI.

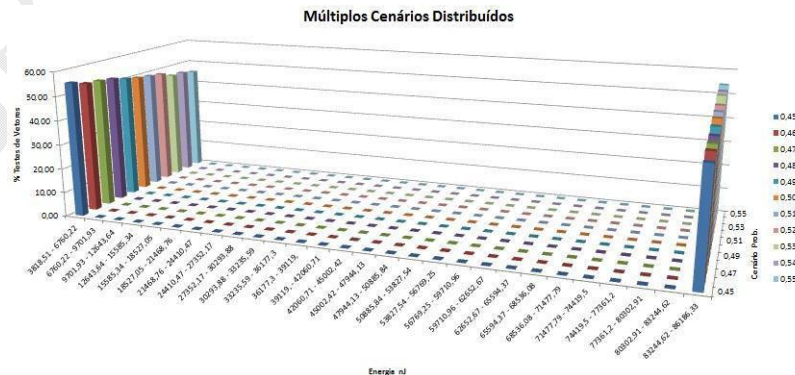


Figura 25 – Gráfico Múltiplo Cenário probabilístico Distribuído – CRC.

É tratado na forma de média na Figura 26 para uma melhor compreensão do seu comportamento.

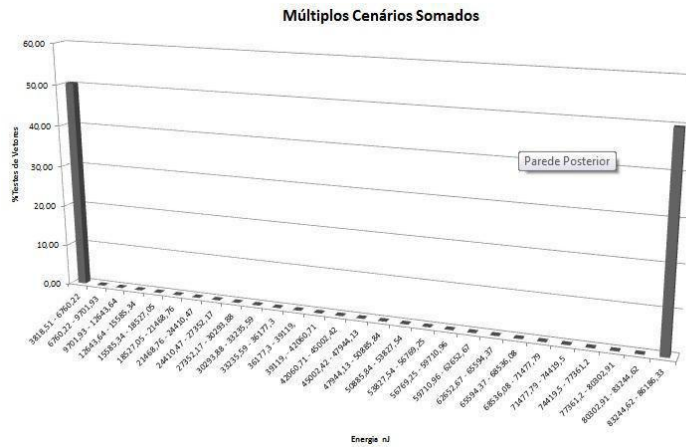
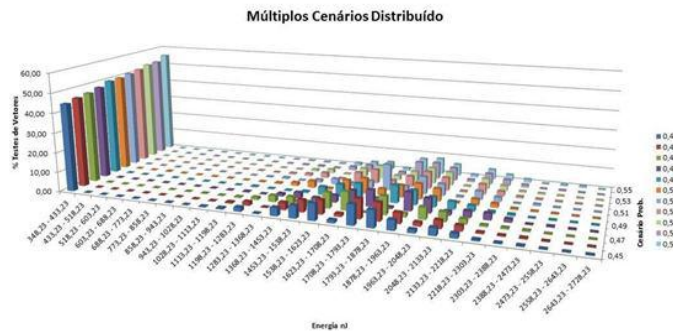


Figura 26 – Gráfico Múltiplo Cenário probabilístico Somado- CRC

Resultados – QURT - PESTI Múltiplos

E finalmente o código QURT sendo submetido aos múltiplos cenários da ferramenta PESTI seu comportamento pode ser visualizado na Figura 27.



E da mesma forma que foi feito para o bolha e CRC seus dados foram tratados na forma de média e são mostrados no gráfico da Figura 28 para uma melhor entendimento do seu comportamento energético.

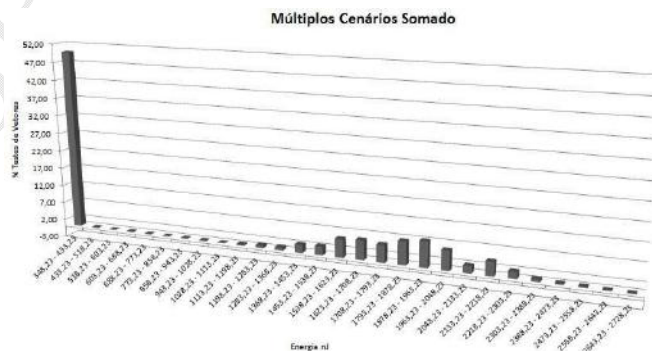


Figura 28 – Gráfico Múltiplo Cenário probabilístico Somado- QURT

Tabela 1 – Resultados Comparativos entre Estimado e Executado no ARM

Código ANSI C	Estimador - PESTI - Energia em μJ	Caracterizador - Energia em μJ	Estimador - PESTI - Múltiplos Somado- Energia em μJ
Bolha	0,133	2,225	112,199
CRC	89,2	96,7	90,004
QURT	2,85	13,94	3,76

Portanto é fato que ambas ferramentas PESTI e o CARACTERIZADOR possuem coerência em seus valores adquiridos em seu funcionamento, porém com fidelidade de valores para menos da PESTI em relação ao CARACTERIZADOR.

CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma abordagem para fase de planejamento e desenvolvimento de sistemas embarcados, possibilitando a análise de aspectos de consumo de energia.

Descrever uma forma de se estimar e caracterizar o consumo de energia e tempo de execução de sistemas embarcados desenvolvidos em linguagem ANSI C. A abordagem conta com o suporte da ferramenta PESTI modificada para atender simples cenários ou múltiplos cenários e validada aplicando através de estudos de casos diferentes de algoritmos:

- De ordenação em bolha (*bubblesort*);
- De redundância cíclica (*CRC*);
- De cálculo de raiz quadrada (*QURT*).

E da mesma forma os códigos usados foram embarcados em um processador ARM7 executados e medidos de forma coerente em um ambiente real.

O primeiro estudo de caso trata-se de um código um tanto conhecido e que possui um elemento de natureza probabilística que define cenários probabilísticos usando uma matriz de elementos classificados em ordem direta e os dois últimos estudos de caso foram extraídos do *benchmark Powerstone* [32]. O *Powerstone* é um benchmark especificamente desenvolvido para avaliação de consumo de energia por meio da exploração de diferentes aspectos do sistema como já foi dito anteriormente.

No caracterizador através da medição do consumo de corrente do processador, comparação dos resultados mostra que existe uma diferença significativas entre estimativas produzidas pela ferramenta PESTI e pelo caracterizador. Esta diferença por hipótese o consumo energético dos blocos internos do processador ARM7, que trata aritmeticamente os valores de variáveis nos códigos usado neste trabalho, isto ocorreu para os três códigos bolha, CRC e QURT usado nos experimentos.

Acreditamos que as evidências produzidas pelos estudos de casos, que nossa abordagem pode ser empregada no projeto de aplicações de pesquisas e desenvolvimento de sistemas embarcados que utilizam códigos escritos em ANSI C em produtos portáteis comerciais, desde que existam ferramentas que suportem o modelo proposto, automatizem o processo de tomada de estimativas e forneçam resultados confiáveis em tempo de projeto, representa uma importante tarefa no projeto de sistemas embarcados, pois nas fases iniciais do desenvolvimento do sistema é que o projetista deve tomar a decisão sobre quais algoritmos e plataformas de hardwares (processador) deverão ser utilizadas. Nossa ferramenta permite que o desenvolvedor estime diversos cenários de execução para o mesmo algoritmo sobre um determinado processador, não só ARM, mas outros processadores. Portanto, o desenvolvedor é capaz de comparar os resultados obtidos, não só estimativas de consumo de energia, mas também tempo de execução, para vários tipos de algoritmos e outros processadores desejados.

CONTRIBUIÇÕES

A proposta desta dissertação de mestrado tem como objetivo definir uma abordagem de medição e comparação para estimar consumo de energia e caso necessário o Tempo de Execução de Códigos para sistemas embarcados desenvolvidos em linguagem ANSI C.

A abordagem definida e apresentada neste trabalho teve como base as idéias de Oliveira Junior [3] e Carlos Mar [2]. As principais diferenças entre ambos os trabalhos são os modelos utilizados para representar o fluxo de controle de um programa, a linguagem de programação utilizada para codificar os programas, além de que ambos utilizaram plataforma distintas (8051) Oliveira Junior [3] e (ARM7, mas não usada, e é substituída pelo simulador *Sim-PAalyzer*) Carlos Mar [2].

Neste trabalho as principais contribuições foram:

- A definição formal é um modelo caracterização real para códigos embarcados em um processador ARM7 sugeridos muitos autores de trabalhos relacionados utilizados nesta dissertação;
- A definição de um novo padrão de anotação de código para representação de múltiplos cenários probabilísticos;
- A extensão da ferramenta PESTI para estimar o consumo de energia em sistemas embarcados, para múltiplos cenários.

Avaliando sistemas que possuam códigos escritos em linguagem ANSI C, percebe-se que a abordagem proposta tenha a possibilidade de ser empregada em grande parte de aplicações de sistemas embarcados, pois essa linguagem de programação é bastante difundida no desenvolvimento de aplicações de linha de pesquisa de sistemas embarcados e de produtos portáteis. O padrão de anotação proposto neste trabalho permite ao desenvolvedor e projetista definam desde que simples cenários até múltiplos cenários probabilísticos de execução de seus sistemas e também permitir o desenvolvimento de outras ferramentas automáticas de reconhecimento de tal notação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] [Barreto 2005] Barreto, R. (2005). **A Time Petri Net-Based Methodology for Embedded Hard Real-Time Software Synthesis**. PhD thesis, Cin/UFPE.
- [2] [Araújo Mar 2009] Araújo Mar, R. (2009). **Uma Abordagem Probabilística baseada em Grafo de Fluxo de Controle para Estimar o Consumo de Energia**. M.Sc. thesis, UFAM.
- [3] [Oliveira Junior 2006] Oliveira Junior, M. N. (2006). **Estimativa do Consumo de Energia Devido ao Software: Uma Abordagem Baseada em Redes de Petri Coloridas**. PhD thesis, Cin/UFPE.
- [4] [Callou 2006] Oliveira Junior, M. N. (2006). **Um Método Formal Baseado em Redes de Petri Coloridas para Estimar o Tempo de Execução e o Consumo de Energia em Sistemas Embarcados**. PhD thesis, Cin/UFPE.
- [5] [Pires Ribeiro 2007] Pires Ribeiro, A. (2009). **Estimativa do Consumo de Energia de Código ANSI-C para sistemas embarcados: Uma abordagem baseada em simulação estocástica**. M.Sc. thesis, UFPE.
- [6] [Bona et al. 2002] Bona, A., Sami, M., Sciuto, D., Zaccaria, V., Silvano, C., and Energy, R. Z. (2002). **Estimation and optimization of embedded vliw processors based on instruction clustering**. In In Proceedings of the 39th conference on Design automation, pages 886 - 891. ACM Press.
- [7] [Brooks et al. 2000] Brooks, D., Tiwari, V., and Martonosi, M. (2000). **Wattch: a framework for architectural-level power analysis and optimizations**. In Proceedings of 27th annual international symposium on Computer architecture, pages 83 - 94. ACM Press.
- [8] [Burch et al. 1993] Burch, R., Najm, F., Yang, P., and Trick, T. (1993). **A monte carlo approach for power estimation**. In IEEE Transactions on VLSI Systems.
- [9] [Costa et al. 1999] Costa, J., Monteiro, J., Silveira, L., and Devadas, S. (1999). **A probabilistic approach for rt-level power modeling**. In 6th IEEE International Conference on Electronics, Circuits and Systems.

- [10] [Klass et al. 1998] Klass, B., D.Thomas, Schmit, H., and Nagle, D. (1998). **Modeling inter-instruction energy effects in a digital signal processor**. in power-driven microarchitecture workshop. In conjunction with ISCA98.
- [11] [Laurent et al. 2004] Laurent, J., Julien, N., Senn, E., and Martin, E. (2004). **Functional level power analysis: An efficient approach for modeling the power consumption of complex processors**. In IEEE DATE.
- [12] [Lee et al. 1997] Lee, M., Tiwari, V., Malik, S., and Fujita, M. (1997). **Power analysis and minimization for embedded DSP software**. In IEEE Trans. VLSI Systems, pages 123 - 135.
- [13] [Nikolaidis et al. 2002] Nikolaidis, S., Kavvadias, N., and Neofotistos, P. (2002). **Instruction-level power measurement methodology**. Technical report, Electronics Lab., Physics Dept., Aristotle University of Thessalonik.
- [14] [Ravi et al.] Ravi, S., Raghunatha, A., and Chakradhar, S. **Efficient RTL power estimation for large designs**. Computer and Communication Research Labs, NEC USA, Princeton, NJ 08540.
- [15] [Leon-Garcia et al. 2008] LEON-GARCIA, A. **Probability and Random Processes for Electrical Engineering**. 3rd Edition. Prentice Hall.
- [16] [Allen, A. O. et al. 1990] ALLEN, A. O. **Probability, Statistics, and Queueing Theory with Computer Science Applications**. Second Edition. Academic Press.
- [17] [Ball, Stuart et al. 2005] Ball, Stuart. - **Embedded Microprocessor Systems: Real World Design**, 3rd edition, Editora: MCPros, EUA.
- [18] - [Reis, Claiton et al. 2004] Reis, Claiton - **Sistemas Operacionais para Sistemas Embarcados**, Tutorial, Editora: EDUFBA, BRASIL.
- [19] - [Cunha, Alessandro et al. 2007] Cunha, Alessandro - **Sistemas Embarcados**, Revista Saber Eletrônica, 414, Editora: Saber, BRASIL.
- [20] - Prof. Rajesh K. Gupta - **Introduction to Embedded Systems**, Website, Acessado em: 10/12/2010. UCLA, EUA, 2002. <http://www.ics.uci.edu/~rgupta/ics212/w2002/intro.pdf>
- [21] - SOUSA, Daniel Rodrigues. **Microcontroladores ARM7**. São Paulo: Érica. 2006
- [22] - RENALDI, Felipe. Genos - **Protótipo De Um Montador De Sistemas Operacionais para Sistemas Embarcados**. Santa Catarina: Monografia FURB, 2006. Disponível em: http://www.bc.furb.br/docs/MO/2006/307628_1_
- [23] - DAVID, J. Lilja, **Measuring Computer Performance: A Practitioner's Guide**. Cambridge University Press, 2000.
- [23] - OSCILOCOPIO TEKTRONICS, **Manual de Operação DPO 4034**.
- [24] V. Tiwari, S. Malik, and A. Wolf. **Power analysis of embedded software: a first step towards software power minimization**. In IEEE Trans. VLSI Systems, volume 2, pages 437-445. December 1994.
- [25] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. **The design and use of simplepower: A cycle-accurate energy estimation tool**. In *Proceedings of 37th conference on Design automation*, pages 340-345. ACM Press, 2000.
- [26] E. Senn, J. Laurent, N. Julien, and E. Martin. Softexplorer: **estimation, characterization and optimization of the power and energy consumption at the algorithmic level**. In IEEE PATMOS. September 2004.
- [27] V. Tiwari and M. Lee. **Power analysis of a 32-bit embedded microcontroller**. In VLSI Design Journal, page 7(3). 1998.
- [28] V. Tiwari, S. Malik, A. Wolf, and M. Lee. **Instruction level power analysis and optimization of software**. In J. VLSI Signal Process. Syst., pages 13(2-3):223-238. 1996.
- [29] Vivek Tiwari. **Logic and System Design for Low Power Consumption**. PhD thesis, Princeton University, November 1996.
- [30] J. Gross. **Graph Theory and its applications**. CRC, Boca Raton, 1999.
- [31] Sim-Panalyzer, **The SimpleScalar-Arm Power Modeling Project**. Disponível em <http://www.eecs.umich.edu/panalyzer/>, 2008.

- [32] Afzal Malik, Bill Moyer, and Dan Cermak. **A low power unified cache architecture providing power and performance flexibility (poster session)**. In ISLPED '00: Proceedings of the 2000 international symposium on Low power electronics and design, pages 241-243, New York, NY, USA, 2000. ACM.
- [33] **Synopsys products and solutions**. Disponível em <http://www.synopsys.com/products/solutions/galaxy/power/power.html>, 2008.
- [34] **Introduction to Embedded Systems**. Disponível em <http://www.ics.uci.edu/~rgupta/ics212/w2002/intro.pdf>, 2009.
- [35] **The Lex and Yacc Page**. Disponível em <http://dinosaur.compilertools.net/>, 2009.
- [36] **Python Programming Language** - Oficial Website. Disponível em <http://www.python.org/>, 2009.
- [37] **Sparse - a Semantic Parser for C**. Disponível em <http://www.kernel.org/pub/software/devel/sparse/>, 2009.
- [38] **Manta Antiestática**. Disponível em <http://www.clangsm.com.br/vb/esquemas-de-hardware-repara%E7%F5es-jumpers-em-geral/7222-manta-dissipativa.html>.
- [39] **Pulseira Antiestática**. Disponível em http://pt.wikipedia.org/wiki/Pulseira_antiest%C3%A1tica.
- [40] Carlos Mar, Raimundo Barreto, Meuse Oliveira Jr e Edward Moreno Ordonez. **PESTI:A Probabilistic Tool for Energy Consumption and Execution-Time Estimation for ANSIC Embedded Systems**. IEEE International Conference on Industrial Technology (ICIT 2009). 10 - 13 February, 2009. Churchill, Victoria, Australia.