



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL**  
**FACULDADE DE INFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ESTIMATIVA E REDUÇÃO DA DISSIPACÃO  
DE POTÊNCIA EM REDES INTRA-CHIP  
COM CHAVEAMENTO POR PACOTES**

**GUILHERME MONTEZ GUINDANI**

Dissertação apresentada como  
requisito parcial à obtenção do grau  
de Mestre em Ciência da  
Computação.

Orientador: Prof. Dr. Fernando Gehm Moraes

Porto Alegre, Dezembro 2008.

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



## **Dados Internacionais de Catalogação na Publicação (CIP)**

G964e	Guindani, Guilherme Montez Estimativa e redução da dissipação de potência em redes intra-chip com chaveamento por pacotes / Guilherme Montez Guindani. – Porto Alegre, 2008. 95 f. Diss. (Mestrado) – Fac. de Informática, PUCRS. Orientador: Prof. Dr. Fernando Gehm Moraes. 1. Informática.            2. Redes de Computadores. 3. Arquitetura de Redes.    I. Moraes, Fernando Gehm. II. Título.
-------	---

**Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS**





Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Estimativa e Redução da Dissipação de Potência em Redes Intra-Chip com Chaveamento por Pacotes**", apresentada por Guilherme Montez Guindani, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Embarcados e Sistemas Digitais, aprovada em 12/01/09 pela Comissão Examinadora:

Prof. Dr. Fernando Gehm Moraes -  
Orientador

PPGCC/PUCRS

Prof. Dr. Ney Laert Vilar Calazans -

PPGCC/PUCRS

Prof. Dr. César Augusto Missio Marcon -

FACIN/PUCRS

Prof. Dr. Luigi Carro -

UFRGS

Homologada em 14/04/2009, conforme Ata No. 006... pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes  
Coordenador.

PUCRS

**Campus Central**

Av. Ipiranga, 6681 - P32 - sala 507 - CEP: 90619-900  
Fone: (51) 3320-3611 - Fax (51) 3320-3621  
E-mail: [ppgcc@pucrs.br](mailto:ppgcc@pucrs.br)  
[www.pucrs.br/facin/pos](http://www.pucrs.br/facin/pos)



## Agradecimentos

Gostaria de agradecer a diversas pessoas que contribuíram para a realização desta dissertação de mestrado, e não poderia deixar de manifestar meu profundo agradecimento a todas elas.

Agradeço a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo financiamento deste trabalho, provendo o apoio financeiro necessário para a realização deste trabalho e possibilitando a minha dedicação exclusiva ao mesmo. Aos funcionários, alunos e professores do PPGCC pela convivência, amizade e experiências compartilhadas ao longo do curso de mestrado. Especial abraço ao secretário Thiago Lingener que aturou meus diversos pedidos de documentos, assinatura de ponto e reclamações. Aos colegas e amigos do grupo de apoio ao projeto de hardware (GAPH) pelo ótimo ambiente de trabalho e ótima convivência, tanto os que já não se encontram no grupo quanto os que ainda estão trabalhando comigo. Agradeço aos amigos, Leandro Möller, Edson Moreno, Ewerson Carvalho, Leonel Tedesco e Luciano Ost pelos conselhos, discussões/sugestões técnicas e parceria nas publicações. E um agradecimento especial para os alunos Cezar Reinbrecht e Thiago Raupp, os quais assumiram comigo a responsabilidade de trabalhar com um tema nunca antes abordado no grupo de pesquisa, e sem eles este trabalho não seria realizado. Agradeço também ao pessoal dos projetos TETHA e X10GIGA, pelo ótimo ambiente de trabalho (quando eu por lá trabalhei) e pela parceria e amizade. Especialmente, a Eduardo Wächter, Samuel Marczak, Alzemiro Lucas e Taciano Rodolfo.

Ao Prof. Dr. Luigi Carro, professor da Universidade Federal do Rio Grande do Sul, que gentilmente aceitou participar e colaborar com este trabalho fazendo parte da banca de avaliação. Ao Prof. Dr. Ney Laert Vilar Calazans por seu apoio contínuo aos meus trabalhos, com comentários, críticas e sugestões que me ajudaram, desde meus primeiros artigos publicados como bolsista até esta dissertação de mestrado. Ao Prof. Dr. César Augusto Missio Marcon por estar contribuindo neste trabalho, com opiniões e críticas, desde o seu início, e que também aceitou gentilmente participar da banca de avaliação. Agradeço especialmente ao meu orientador e amigo, Prof. Dr. Fernando Gehm Moraes pelo constante incentivo, pela dedicação e paciência, por estar sempre disponível a esclarecer dúvidas e frustrações geradas durante meu trabalho. Agradeço principalmente, por depositar sua confiança em mim e no meu trabalho, e espero que não o tenha decepcionado.

Agradeço a todos os meus amigos pessoais, por me ajudar a esquecer um pouco o trabalho nos momentos certos, jogando um futebol, saindo para as festas. Momentos como estes foram a minha válvula de escape, e ajudaram a manter meu foco para conseguir esta importante vitória.

A toda a minha família que sempre me acreditou em mim, e sempre esteve ao meu lado. Ao meu irmão Gustavo Montez Guindani, que tudo o que faço tento passar-lhe como exemplo. Aos meus pais Sergio Dantas Guindani e Angelina de Melo Montez Guindani pela confiança, amor incondicional e apoio total em tudo o que proponho fazer. Amo todos vocês.

Sobre tudo, agradeço a Deus por sempre estar ao meu lado nos melhores e piores momentos da minha vida.

Muito Obrigado.





## Resumo

No passado, as maiores preocupações de um projetista VLSI compreendiam área de silício, desempenho, custo e confiabilidade. A dissipação de potência e a energia consumida pelo circuito ocupavam uma posição secundária no projeto do mesmo. A computação móvel mudou este cenário, trazendo a dissipação de potência e o consumo de energia de um circuito como preocupações primárias, visando principalmente aumentar a vida útil das baterias do dispositivo. A primeira medida para reduzir a potência do circuito é agregar diversas funções, que antes estariam em circuitos independentes, em um mesmo circuito, criando o conceito de SoC. Devido ao número de funções integradas nos SoCs modernos, as redes intra-chip (NoCs) estão sendo mais utilizadas devido a sua escalabilidade e paralelismo de comunicação. Os modelos atuais de estimativa da dissipação de potência nas NoCs utilizam como principal métrica o volume de dados transmitidos na rede. Este trabalho propõe um modelo para estimativa da dissipação de potência em uma rede intra-chip mais preciso, baseado nas taxas de recepção nos buffers de cada roteador da rede durante a transmissão do tráfego avaliado. Resultados mostram um erro de no máximo 10% em comparação com uma ferramenta comercial de estimativa de dissipação de potência, reduzindo o tempo de estimativa. Comparando o modelo proposto com modelos baseados em volume de dados, podemos verificar a imprecisão dos modelos de volume, mesmo que estes apresentem tempo de resposta melhores. O modelo proposto foi integrado ao framework ATLAS, permitindo que os projetistas avaliem a potência dissipada para as NoCs e tráfegos gerados neste framework. Além disto, este trabalho apresenta a avaliação da dissipação de potência da NoC Hermes, avaliando a aplicação de técnicas de redução de potência em circuitos CMOS sobre a rede.

**Palavras Chave:** Redes Intra-Chip (NoCs), Modelos de estimativa de potência, Técnicas de redução de potência, NoCs com baixa dissipação de potência, *Clock-gating*.

## **Abstract**

*The main cost functions in VLSI design during the 90's were silicon area, performance, cost and reliability. Power dissipation and energy consumption were a secondary concern. For example, some Intel processors could dissipate more than 120 watts. Mobile computing changes this scenario, making energy consumption and battery life a primary concern. A first move to reduce power is to integrate most of the system functions together in a single integrated circuit, leading to the SoC concept. Due to number of functions integrated in modern SoCs, networks on chip (NoCs) are becoming the preferred communication infrastructure due to their scalability and communication parallelism. Current power estimation models for NoCs capitalize mostly in the volume of information transmitted through the network. This work proposes a more precise NoC power estimation model, based in the buffer reception rates, according to the traffic scenario applied to the network. Results show the accuracy of the model compared to industrial power estimation tools, with an error inferior to 10%, with reduced execution time. Compared to a volume based estimation method, it was possible to demonstrate the weakness of such methods, even if they are faster to generate results. The proposed model is integrated into the ATLAS framework, providing designers a path to evaluate power and energy of automatically generated NoCs. Additionally, this work evaluates the dissipation of each Hermes internal component, with and without power reduction techniques.*

**Keywords:** *Networks on Chip (NoC), Power estimation models, Power reduction techniques, Low-power NoCs, Clock-gating.*

## Lista de Figuras

<i>Figura 1 – Diagrama de inversor CMOS.</i>	23
<i>Figura 2 – Modelo de pilha de protocolos para um SoC que utiliza uma NoC como estrutura de comunicação.</i>	29
<i>Figura 3 - Exemplos de CDG (a), CWG (b) e CRG (c).</i>	31
<i>Figura 4 – Modelagem de um MPSoC sobre a estrutura de um grafo; (a) Diagrama de blocos típico de uma arquitetura de microprocessador; (b) Representação do grafo correspondente.</i>	32
<i>Figura 5 - Arquitetura genérica de um roteador wormhole.</i>	33
<i>Figura 6 - Análise do efeito da transição dos bits na dissipação média de potência para diferentes profundidades de buffer e para uma lógica de controle centralizada em uma rede Hermes com flits de 8 bits [PAL07].</i>	35
<i>Figura 7 - Análise do efeito da transição dos bits na dissipação média de potência para diferentes profundidades de buffer e para uma lógica de controle centralizada em uma rede Hermes com flits de 16 bits [PAL07].</i>	35
<i>Figura 8 – Refinamentos para criar uma rede crossroad; (a) Grafo de comunicação entre os núcleos; (b) Posicionamento final dos núcleos na rede; (c) Posicionamento inicial dos núcleos; (d) Posicionamento otimizado dos núcleos.</i>	36
<i>Figura 9 – Estrutura de comunicação Crossroad; (a) Roteador CCB interconectando 4 núcleos, e (b) a interconexão entre CCBs formando uma rede.</i>	37
<i>Figura 10 - Arquitetura do roteador sensível ao caminho de dados.</i>	38
<i>Figura 11 - Mapeamento e roteamento de baixo consumo de energia.</i>	39
<i>Figura 12 – Projeto de MOCAs. (A) representa a entrada da primeira fase da MOCA. (B) e (C) representam a ação da técnica de refinamento baseado em árvore de cortes. (D) representa a segunda fase da do projeto de MOCAs.</i>	39
<i>Figura 13 – Gerente local de potência.</i>	40
<i>Figura 14 - Sinal de requisuição para o canal de comunicação Sul do roteador 1 (R significa roteador); (a) Canais de entrada na NoC; (b) Rede de requisuições utilizando portas lógicas do tipo OR.</i>	41
<i>Figura 15 – Estrutura da fila bi-síncrona adicionada à rede Hermes para sincronização dos dados [CUM02].</i>	42
<i>Figura 16 - Fluxo de aplicação do modelo de estimativa da dissipação de potência de uma NoC.</i>	49
<i>Figura 17 – Dissipação média de potência do buffer, crossbar e lógica de controle e roteamento, em função da taxa de recepção dos buffers.</i>	50
<i>Figura 18 – Fluxos de injeção utilizados nos estudos de caso, cada um possui 500 pacotes sendo transmitidos a 240 Mbps.</i>	52
<i>Figura 19 - Janela principal do módulo de estimativa de potência HEFESTUS.</i>	55
<i>Figura 20 - Perfil da dissipação de potência dos diversos roteadores; gráfico da potência em função da taxa de recepção nos buffers.</i>	56
<i>Figura 21 – Janela do ambiente de simulação de NoCs no framework ATLAS. Em destaque o campo que configura a janela de amostragem para a avaliação da dissipação de potência.</i>	57
<i>Figura 22 - Dissipação média de potência de uma NoC durante todo o tempo de simulação; gráfico da dissipação de potência em função do tempo.</i>	58
<i>Figura 23 - Dissipação média de potência de um roteador da NoC durante todo o tempo de simulação; gráfico da dissipação de potência em função do tempo.</i>	58
<i>Figura 24 - Distribuição da dissipação de potência nos diversos roteadores da NoC; gráfico 3D com a distribuição instantânea em um dado tempo de simulação.</i>	59
<i>Figura 25 - Resultados obtidos no módulo HEFESTUS exibidos na forma de um relatório.</i>	59
<i>Figura 26 - Tempos de liberação para um cenário de entrega de pacotes sem bloqueios na rede.</i>	60
<i>Figura 27 – Representação visual do PowerScope.</i>	61

<i>Figura 28 - Diferença na estimativa da dissipação média de potência entre as simulações baseada em atores (JOSELITO) e RTL (HEFESTUS).</i> .....	62
<i>Figura 29 - Dissipação de potência nos buffers em relação à dissipação total do roteador.</i> .....	65
<i>Figura 30 – Dissipação média de potência nos roteadores da NoC Hermes de referência, variando-se o tamanho de flit e a profundidade dos buffers de entrada.</i> .....	66
<i>Figura 31 – Diferença entre um circuito sintetizado sem a aplicação de clock-gating, e com a aplicação de clock-gating automático, no ambiente de síntese Design Compiler da Synopsys.</i> .....	67
<i>Figura 32 - Perfil da dissipação de potência dos roteadores da NoC, com e sem a técnica de clock-gating (CKED).</i> .....	68
<i>Figura 33 – Gráfico que apresenta a diferença nos valores de área, frequência e dissipação de potência em uma NoC com e sem a aplicação de clock-gating (CKED).</i> .....	69
<i>Figura 34 – Estrutura do buffer da NoC Hermes modificada, inserindo-se a técnica de clock-gating manual.</i> .....	70
<i>Figura 35 – Impacto da atividade de chaveamento nos flits transmitidos na dissipação média de potência para as NoCs Hermes de referência, com a utilização de clock-gating automático (CKED Auto) e utilizando clock-gating manual (CKED Manual).</i> .....	72
<i>Figura 36 – Impacto da atividade de chaveamento nos buffers de entrada na dissipação média de potência para as NoCs Hermes de referência, com a utilização de clock-gating automático (CKED Auto) e utilizando clock-gating manual (CKED Manual).</i> .....	73
<i>Figura 37 - Simulações da transmissão de 100, 1000 e 10000 pacotes de 16 flits (distribuição espacial aleatória e temporal normal) para diferentes algoritmos adaptativos mínimos.</i> .....	76
<i>Figura 38 - Simulações da transmissão de 100, 1000 e 10000 pacotes de 16 flits (distribuição espacial do tipo complemento e temporal uniforme) para diferentes algoritmos adaptativos mínimos.</i> .....	76
<i>Figura 39 – Estrutura de pastas para o processo de calibração.</i> .....	90
<i>Figura 40 – Interface principal do framework ATLAS.</i> .....	91
<i>Figura 41 – Interface de geração de NoCs, a ferramenta MAIA no framework ATLAS.</i> .....	91
<i>Figura 42 – Interface da ferramenta de geração de tráfego no framework ATLAS.</i> .....	92
<i>Figura 43 – Interface de configuração de tráfego de um roteador (roteador 00) na ferramenta de geração de tráfego.</i> .....	93
<i>Figura 44 – Fluxograma do funcionamento do programa desenvolvido para aplicar a técnica de ajuste à reta.</i> .....	99

## Lista de Tabelas

<i>Tabela 1 - Consumo de energia de cada um dos blocos básicos de um determinado módulo. ....</i>	<i>34</i>
<i>Tabela 2 – Tabela resumo dos trabalhos relacionados a modelos de estimativa da dissipação de potência em NoCs. ....</i>	<i>36</i>
<i>Tabela 3 - Tabela resumo dos trabalhos relacionados a técnicas de redução da dissipação de potência em NoCs. ....</i>	<i>44</i>
<i>Tabela 4 - Potência média estimada pelo PrimePower e no Modelo proposto nos diversos roteadores da NoC avaliada. ....</i>	<i>51</i>
<i>Tabela 5 – Impacto do aumento de fluxos injetados na rede sobre a potência média dissipada por uma NoC. Estimativas obtidas utilizando-se diferentes modelos de estimativa.....</i>	<i>53</i>
<i>Tabela 6 – Avaliações de erro entre as estimativas realizadas nos modelos avaliados e a ferramenta PrimePower. A última linha é a diferença entre os dois modelos.....</i>	<i>53</i>
<i>Tabela 7 – Avaliações de tempo de execução dos modelos avaliados em comparação com a ferramenta PrimePower. Na tabela os tempos estão em minutos, e a sigla “Inst.” demonstra que o tempo foi instantâneo. ....</i>	<i>54</i>
<i>Tabela 8 – Estimativas da dissipação de potência para cenários de tráfego mais intenso.....</i>	<i>54</i>
<i>Tabela 9 – Avaliação de erro entre os diferentes modelos quando aplicado cenários de tráfego mais intenso. ....</i>	<i>54</i>
<i>Tabela 10 - Avaliações de tempo de execução dos modelos avaliados em comparação com a ferramenta PrimePower para tráfegos com alta transmissão na rede. Na tabela os tempos estão em minutos, e a sigla “Inst.” demonstra que o tempo foi instantâneo. ....</i>	<i>55</i>
<i>Tabela 11 - Dissipação de potência nos roteadores centrais, a uma taxa de recepção de 10% da taxa máxima no enlace. As <u>diagonais</u> da tabela mostram regiões que possuem o mesmo tamanho de buffer (em bits), em consequência estas regiões possuem a mesma dissipação de potência. ....</i>	<i>67</i>
<i>Tabela 12 - Comparação da potência dissipada por NoCs com controle de fluxo do tipo handshake (HAN) e baseado em créditos (CRE), com e sem o uso de clock-gating (CKED). ....</i>	<i>69</i>
<i>Tabela 13 - Comparação da potência consumida por NoCs, com controle de fluxo baseado em créditos (CRE), variando-se a quantidade de canais virtuais (CV). Em todos os casos foi utilizada a técnica de clock-gating (CKED).....</i>	<i>69</i>
<i>Tabela 14 – Dissipação de potência para uma NoC sem clock-gating, com clock-gating automático e com clock-gating manual.....</i>	<i>71</i>
<i>Tabela 15 – Atividade de chaveamento nos pacotes gerados através do framework ATLAS, em função do tamanho do pacote em flits.....</i>	<i>72</i>
<i>Tabela 16 – Resultados das avaliações dos algoritmos adaptativos e XY para as simulações com 1000 pacotes de 16 flits. Os melhores resultados dentre os algoritmos avaliados estão destacados em negrito. ....</i>	<i>75</i>
<i>Tabela 17 - Roteadores origem e destino nos cenários de tráfego controlados. ....</i>	<i>93</i>



## Lista de Siglas

<b>ASIC</b>	<b>Application Specific Integrated Circuit</b>
<b>ASIP</b>	<b>Application Specific Instruction-Set Processor</b>
<b>ADC</b>	<b>Analog to Digital Converter</b>
<b>APH</b>	<b>Average Power per Hop</b>
<b>CAD</b>	<b>Computer Aided Design</b>
<b>CCB</b>	<b>Crossroad Communication Block</b>
<b>CDCG</b>	<b>Communication Dependence and Computation Graph</b>
<b>CWG</b>	<b>Communication Weighted Graph</b>
<b>DAC</b>	<b>Digital to Analog Converter</b>
<b>DoSP</b>	<b>Domain Specific Processor</b>
<b>DPM</b>	<b>Dynamic Power Management</b>
<b>DSP</b>	<b>Digital Signal Processor</b>
<b>DVFS</b>	<b>Dynamic Voltage/Frequency Scaling</b>
<b>E/S</b>	<b>Entrada e Saída</b>
<b>EMI</b>	<b>Electro-Magnetic Interference</b>
<b>FPGA</b>	<b>Field Programmable Gate Array</b>
<b>GALS</b>	<b>Globally Asynchronous Locally Synchronous</b>
<b>GAPH</b>	<b>Grupo de Apoio ao Projeto de Hardware</b>
<b>GPHR</b>	<b>Global Phase History Register</b>
<b>GPP</b>	<b>General Purpose Processor</b>
<b>HDL</b>	<b>Hardware Description Language</b>
<b>ILP</b>	<b>Instruction Level Parallelism</b>
<b>I/O</b>	<b>Input/Output</b>
<b>ITRS</b>	<b>International Technology Roadmap for Semiconductors</b>
<b>MOCA</b>	<b>Mesh based On-Chip interconnection Architectures</b>
<b>MPSoC</b>	<b>Multiprocessor System on a Chip</b>
<b>NE</b>	<b>North-East</b>
<b>NoC</b>	<b>Network on Chip</b>
<b>NW</b>	<b>North-West</b>
<b>PACP</b>	<b>Power-Aware Core Placement</b>
<b>PDA</b>	<b>Personal Digital Assistant</b>
<b>PE</b>	<b>Processor Element</b>
<b>PAT</b>	<b>Packet Abstraction Technique</b>
<b>PHT</b>	<b>Phase History Table</b>
<b>QoS</b>	<b>Quality of Service</b>
<b>RTL</b>	<b>Register Transfer Level</b>
<b>S.O.</b>	<b>Sistema Operacional</b>
<b>SoC</b>	<b>System on a Chip</b>
<b>VC</b>	<b>Virtual Channel</b>



<b>VHSIC</b>	<b>Very-High-Speed Integrated Circuit</b>
<b>VHDL</b>	<b>VHSIC Hardware Description Language</b>

# Sumário

<b>1</b>	<b><u>INTRODUÇÃO</u></b> .....	<b>19</b>
1.1	SISTEMAS DE COMPUTAÇÃO MÓVEL.....	19
1.2	OBJETIVOS.....	21
1.3	CONTRIBUIÇÕES DO PRESENTE TRABALHO.....	21
1.4	ORGANIZAÇÃO DO DOCUMENTO.....	21
<b>2</b>	<b><u>CONCEITOS BÁSICOS</u></b> .....	<b>23</b>
2.1	A DISSIPACÃO DE POTÊNCIA EM UM CIRCUITO INTEGRADO.....	23
2.1.1	Potência Dinâmica.....	23
2.1.2	Potência de Curto-Circuito.....	24
2.1.3	Potência Estática.....	25
2.2	CONSUMO DE ENERGIA.....	25
2.3	MODELOS DE ATRASO.....	26
2.4	<i>CLOCK-GATING</i> E <i>POWER-GATING</i> .....	26
<b>3</b>	<b><u>TRABALHOS RELACIONADOS</u></b> .....	<b>29</b>
3.1	DISSIPACÃO DE POTÊNCIA EM NOCS.....	29
3.2	MODELAGEM DA DISSIPACÃO DE POTÊNCIA E CONSUMO DE ENERGIA EM NOCS.....	31
3.3	NOCS DE BAIXA DISSIPACÃO DE POTÊNCIA.....	36
3.4	CONCLUSÃO.....	44
<b>4</b>	<b><u>MODELO DE ESTIMATIVA DA DISSIPACÃO DE POTÊNCIA EM REDES-INTRA CHIP</u></b>	<b>47</b>
4.1	DESCRIÇÃO DO MODELO PROPOSTO.....	47
4.2	COMPARAÇÃO COM UM MODELO BASEADO EM VOLUME DE DADOS.....	51
4.3	HEFESTUS – INTEGRAÇÃO DO MODELO PROPOSTO AO FRAMEWORK ATLAS.....	55
4.4	INTEGRAÇÃO DO MODELO PROPOSTO COM O MODELO JOSELITO.....	59
4.5	CONCLUSÃO.....	63
<b>5</b>	<b><u>DISSIPACÃO DE POTÊNCIA NA NOC HERMES</u></b> .....	<b>65</b>
5.1	DISSIPACÃO DE POTÊNCIA NA NOC HERMES DE REFERÊNCIA.....	65
5.2	DISSIPACÃO DE POTÊNCIA NA NOC HERMES <i>CLOCK-GATED</i> AUTOMÁTICO.....	67
5.3	DISSIPACÃO DE POTÊNCIA NA NOC HERMES <i>CLOCK-GATED</i> MANUAL.....	70
5.4	INFLUÊNCIA DA ATIVIDADE DE CHAVEAMENTO DOS DADOS.....	71
5.5	DISSIPACÃO DE POTÊNCIA DOS DIFERENTES ALGORITMOS DE ROTEAMENTO.....	74
5.6	CONCLUSÃO.....	76
<b>6</b>	<b><u>CONCLUSÕES E TRABALHOS FUTUROS</u></b> .....	<b>79</b>
6.1	CONCLUSÕES.....	79
6.2	TRABALHOS FUTUROS.....	80
	<b><u>REFERÊNCIAS BIBLIOGRÁFICAS</u></b> .....	<b>81</b>
	<b><u>APÊNDICE I – PUBLICAÇÕES ACEITAS E SUBMETIDAS</u></b> .....	<b>87</b>



# 1 INTRODUÇÃO

---

A crescente demanda por equipamentos eletrônicos cada vez menores, mais duradouros, mais rápidos e com mais funcionalidades no mesmo dispositivo, motiva a pesquisa por novos métodos de projeto destes. A busca por desempenho em microprocessadores seguiu duas estratégias simultâneas ao longo do tempo: aumento da frequência de relógio e paralelismo no nível de instrução (ILP – *Instruction Level Parallelism*) [NAI97]. O aumento da frequência está alcançando um limite, devido principalmente a problemas de dissipação de potência e complexidade de projeto. ILP seguiu o caminho dos pipelines superescalares com múltiplas unidades de execução (*multi-threading*), e também está chegando a um limite em sistemas mono-processados. Companhias como Intel, AMD, Sony, IBM e Toshiba vêm utilizando em seus microprocessadores o conceito de SoCs (*Systems on a chip*) multiprocessados (MPSoCs - *Multiprocessor Systems on Chip*), ou seja, múltiplos núcleos de processamento em um único chip. MPSoCs aumentam o desempenho, pois diversos processadores cooperam para executar as tarefas e reduzem a latência de comunicação entre os processadores, por estes estarem implementados em um único chip.

No entanto, a inclusão de diversos processadores em um único circuito integrado não implica necessariamente aumento de desempenho. A arquitetura de interconexão entre processadores e demais componentes do MPSoC é que definirá o desempenho real do sistema. MPSoCs atuais tendem a utilizar NoCs, devido às características de escalabilidade e desempenho [DAL01][WIN01][YE03]. Uma NoC é uma rede intra-chip composta por módulos de hardware conectados a roteadores, que por sua vez são conectados entre si através de canais de comunicação [BEN02].

MPSoCs vêm sendo utilizados em diversas aplicações eletrônicas, e as aplicações de computação móvel, em especial, necessitam de MPSoCs que consigam prover um alto desempenho computacional aliado a um baixo consumo de energia.

## 1.1 Sistemas de Computação Móvel

O termo computação móvel está associado ao aumento da capacidade de mover fisicamente serviços computacionais junto com o usuário dos mesmos, ou seja, os elementos computacionais tornam-se dispositivos sempre presentes, que expandem a capacidade do usuário de utilizar os serviços que estes oferecem, independentemente da sua localização. Combinada com a alta capacidade de acesso aos meios de comunicação, a computação móvel tem transformado a computação em serviços que podem ser acessados e utilizados em qualquer lugar.

No decorrer do tempo, a evolução dos sistemas de computação (processamento em lotes, compartilhamento de tempo, redes com sistemas compartilhados, sistemas distribuídos) diminuiu o acoplamento do usuário a recursos e a ambientes computacionais específicos. Para acessar uma página da internet, por exemplo, o usuário não precisa mais utilizar um computador pessoal que utiliza um sistema operacional de propósito geral, ele pode acessar esta página através de um

telefone celular. Contribui também a esta evolução o avanço das tecnologias de fabricação dos circuitos integrados, que permitiu criar dispositivos de hardware cada vez menores. Dentro desta progressão, sistemas de computação móvel representam a separação entre usuário e ambiente computacional. O usuário pode acessar os recursos computacionais (*hardware* e *software*) a qualquer tempo e lugar, desde que localizado dentro dos limites de alcance de uma infra-estrutura de comunicação.

A computação móvel não é um paradigma que lida exclusivamente com questões ligadas às áreas de sistemas distribuídos e redes de computadores. É um paradigma que trata de múltiplas áreas da Ciência da Computação. Por exemplo, o projeto de circuitos integrados para uso em computação móvel deve ser realizado considerando o consumo de energia; sistemas operacionais devem possuir outras características específicas para dispositivos móveis como memória reduzida e processamento de baixo ou médio desempenho; linguagens de programação e compiladores devem ser projetadas em função das características desse ambiente; bancos de dados devem considerar novos mecanismos de integridade e sincronização de dados; a engenharia de software deve propor novos princípios de desenvolvimento para a área como projeto de interface homem-máquina. Além disso, outras áreas, como Psicologia e Sociologia, têm um papel importante em computação móvel por definir formas alternativas de uso da tecnologia de processamento e comunicação de dados.

Os principais dispositivos de computação móveis disponíveis atualmente possuem uma arquitetura baseada em SoCs compostos de vários núcleos de propriedade intelectual [JER05]. Os núcleos presentes nestes SoCs são, na maior parte dos casos, um elemento de processamento, bloco(s) de memória, periféricos do processador (e.g. MAC Ethernet e conversores AD ou DA), e interfaces externas de comunicação para este elemento de processamento (e.g. portas de E/S). Ao se inserir mais de um elemento de processamento em um SoC, têm-se uma estrutura do tipo sistema multiprocessado em um único circuito integrado (do inglês, *multiprocessor system on chip*, ou MPSoC) [JER05].

Uma linguagem de programação muito difundida nos dispositivos de computação móvel é a linguagem Java [JAV08], adaptada a processadores mais simples (8 ou 16 bits). O desenvolvimento de SoCs que executem os *bytecodes* Java não é uma tarefa trivial, devido a restrições de memória e processamento. Diversos trabalhos tratam deste assunto, dentre eles o ambiente de projeto de aplicações Java embarcada SASHIMI [ITO01]. Este ambiente permite que ambos programadores Java e projetistas de SoCs interajam, refinando a aplicação para a execução em um ambiente restrito. O objetivo final do ambiente é a execução da aplicação Java em um microcontrolador batizado de *femtoJava* [ITO99][ITO01]. Este microcontrolador implementa a máquina virtual Java sobre uma pilha de 8 bits. O *femtoJava* é sintetizável e o código da aplicação pode ser executado no hardware sintetizado em FPGA.

Em dispositivos eletrônicos, desempenho e dissipação de potência são funções custo diretas, ou seja, alto desempenho implica alta dissipação de potência e vice-versa. O grande desafio nos dispositivos de computação móvel é obter o melhor desempenho com o menor consumo de energia.

Ao se projetar um MPSoC visando baixa dissipação de potência têm-se três frentes principais de trabalho: reduzir a dissipação de potência nos módulos de processamento [ISC06][EIS06], no meio de comunicação [BEN06][SIM02], nos meios de armazenamento e nos demais módulos de entrada/saída e periféricos.

## 1.2 Objetivos

Os objetivos do presente trabalho são a modelagem e a redução da potência dissipada no meio de comunicação de MPSoCs com interconexão por NoCs. A modelagem de potência dissipada em NoCs permite que os projetistas avaliem de forma rápida o impacto de mudanças na arquitetura da NoC na dissipação de potência e no consumo de energia. Já a redução na dissipação de potência em NoCs permite uma redução no consumo de energia desta, o que pode aumentar a vida útil da bateria que este dispositivo utiliza.

## 1.3 Contribuições do Presente Trabalho

As principais contribuições deste trabalho são um modelo de estimativa da potência média dissipada em uma rede intra-chip, e um estudo sobre a dissipação de potência da NoC Hermes.

O modelo desenvolvido foi inserido no framework ATLAS [ATL07], um ambiente de desenvolvimento de redes intra-chip. O modelo também foi adicionado ao modelo de simulação baseado na técnica PAT, JOSELITO [OST08]. O modelo JOSELITO foi implementado um ambiente de simulação Java, baseado em atores. Este modelo permite simulações e avaliações de potência mais rápidas que uma simulação RTL, mantendo-se a mesma precisão.

O estudo sobre a dissipação de potência da NoC Hermes permite avaliar o impacto da mudança de parâmetros da rede sobre a potência média dissipada por ela. Decisões de projeto tais como dimensionamento de *buffer*, tamanho de *flit* e escolha do algoritmo de roteamento afetam diretamente a dissipação média de potência de uma rede intra-chip.

## 1.4 Organização do Documento

O presente documento está organizado da seguinte forma. No Capítulo 2 são abordados conceitos que caracterizam a dissipação de potência e consumo de energia em redes intra-chip, e os modelos de atraso utilizados nas estimativas de potência e técnicas de redução da dissipação de potência utilizadas no projeto de circuitos integrados. No Capítulo 3 é realizado um estudo sobre a dissipação de potência em MPSoCs que utilizam uma rede intra-chip como meio de interconexão entre seus diversos núcleos, além de apresentar o estado da arte em modelos de estimativa da dissipação de potência em redes intra-chip, e redes intra-chip que utilizam técnicas de redução da dissipação de potência.

Os Capítulos 4 e 5 apresentam as contribuições deste trabalho. No Capítulo 4 é apresentado o modelo de estimativa da dissipação de potência desenvolvido durante o mestrado, bem como seus resultados, comparações com outros modelos e inserção no ambiente ATLAS. O

Capítulo 5 apresenta um estudo sobre a dissipação de potência da rede Hermes, além de avaliações da dissipação de potência desta rede utilizando diferentes técnicas de redução de potência e algoritmos de roteamento.

As conclusões e direcionamentos para trabalhos futuros são apresentados no Capítulo 6. O Apêndice I contém a lista das publicações realizadas no período do mestrado e o Apêndice II contém um tutorial de calibração do modelo proposto para uma rede intra-chip.

## 2 CONCEITOS BÁSICOS

---

Este Capítulo apresenta conceitos básicos relacionados ao desenvolvimento deste trabalho. A Seção 2.1 aborda os conceitos que caracterizam a dissipação de potência em circuitos integrados CMOS. A Seção 2.2 apresenta o conceito de consumo de energia em circuitos integrados CMOS, bem como sua influência no desempenho global de um sistema. A Seção 2.3 descreve o modelo de atraso utilizado nas estimativas de dissipação de potência. Os conceitos relacionados às técnicas de redução na dissipação de potência de um circuito integrado são apresentados na Seção 2.4.

### 2.1 A Dissipação de Potência em um Circuito Integrado

A dissipação de potência em um circuito elétrico é definida pela multiplicação entre os valores instantâneos da corrente e tensão. A dissipação de potência em uma porta lógica determina quanto calor este circuito dissipa e quanta energia está sendo consumida em um dado momento da operação deste circuito. Estes fatores influenciam decisões de projeto, tais como: encapsulamento, dimensionamento das linhas de alimentação, número de transistores que podem ser integrados em um mesmo chip, e sistema de arrefecimento [RAB96a].

A dissipação de potência em circuitos construídos com tecnologia CMOS pode ser dividida em três componentes principais [RAB96b]: dinâmica, de curto-circuito e estática.

#### 2.1.1 Potência Dinâmica

A potência dinâmica é o resultado da carga e descarga das capacitâncias do circuito, e ao se utilizar tecnologias de até 100 nm, é a parcela de dissipação de potência, das três citadas acima, que mais contribui para o consumo total de um circuito. Porém, para tecnologias mais modernas esta parcela de dissipação de potência pode ser menor que a potência estática. Para melhor definir a dissipação de potência dinâmica em circuitos CMOS, analisa-se o circuito mais simples: o inversor (Figura 1).

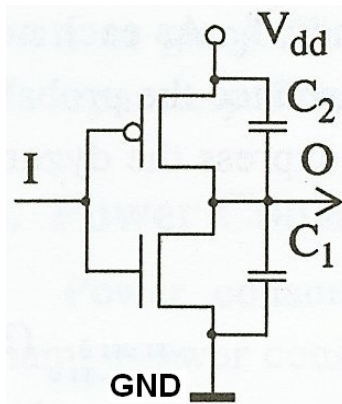


Figura 1 – Diagrama de inversor CMOS.

O inversor serve como um modelo geral de qualquer porta lógica CMOS. Assume-se que



todas as capacitâncias presentes no nodo de saída O, podem ser modeladas como uma capacitância ligada à massa (GND),  $C_1$  na Figura 1 e outra ligada à fonte de alimentação ( $V_{dd}$ ),  $C_2$  na Figura 1. Iniciando o circuito com a saída (O) em nível lógico alto e a entrada (I) em nível lógico baixo, altera-se que o sinal de entrada para o nível lógico alto. Esta ação leva a saída a um nível lógico baixo, e a uma troca nas cargas dos capacitores ligados ao nodo de saída. A carga no capacitor  $C_1$  irá trocar de  $C_1V_{dd}$  para 0, já que o transistor N deixará o capacitor  $C_1$  em curto-circuito. A carga no capacitor  $C_2$  irá trocar de 0 para  $C_2V_{dd}$ , sendo carregado através da fonte de alimentação  $V_{dd}$ . Agora alteramos novamente a entrada, de volta para o nível lógico baixo.  $C_2$  entrará em curto-circuito devido ao transistor P e  $C_1$  será recarregado para  $C_1V_{dd}$ , retirando carga da fonte de alimentação  $V_{dd}$ . Como resultado, durante um ciclo completo de troca nos valores lógicos da entrada, o circuito adquire uma carga total de  $(C_1+C_2)V_{dd}$  da fonte de alimentação ou uma energia total de  $(C_1+C_2)V_{dd}^2$ . Se este ciclo se repetir  $x$  vezes por segundo, o circuito irá dissipar uma potência equivalente à  $(C_1+C_2)xV_{dd}^2$ .

Em um circuito síncrono, a frequência com a qual as capacitâncias do circuito são carregadas e descarregadas reflete diretamente na potência dinâmica. Desta forma, a frequência de relógio aplicada no circuito é um fator que contribui majoritariamente para a potência dinâmica. A potência dinâmica ( $P_d$ ) pode ser calculada aplicando-se a Equação 1 [RAB96b].

$$P_d = \alpha \cdot f_c \cdot C_L \cdot V_{dd}^2 \quad (1)$$

Nesta Equação,  $\alpha$  corresponde à probabilidade de chaveamento do total de portas lógicas do circuito;  $f_c$  corresponde à frequência de relógio aplicada no circuito;  $C_L$  corresponde à carga capacitiva total do circuito;  $V_{dd}$  corresponde à tensão de alimentação aplicada.

### 2.1.2 Potência de Curto-Circuito

A potência de curto-circuito está associada a cada transição na entrada de uma porta lógica, quando todos os transistores (N e P) conduzem simultaneamente por um curto período de tempo. Durante este tempo, existe uma corrente de curto-circuito que flui direto da fonte de alimentação ( $V_{dd}$ ) para a massa (GND), não participando na carga ou descarga da capacitância do circuito ( $C_L$ ). A potência de curto-circuito gerada por esta corrente ( $P_{sc}$ ) pode ser calculada pela Equação 2 [RAB96b].

$$P_{sc} = \frac{\beta}{12} \cdot (V_{dd} - V_{thn} - V_{thp})^3 \cdot \frac{\tau}{T} \quad (2)$$

Nesta Equação,  $\beta$  corresponde à dimensão dos transistores;  $\tau$  corresponde ao tempo de subida, ou descida do sinal de entrada;  $T$  corresponde ao período de transição no sinal de entrada;  $V_{dd}$  corresponde à tensão de alimentação aplicada ao circuito.  $V_{thn}$  corresponde à tensão de *threshold* nos transistores N;  $V_{thp}$  corresponde à tensão de *threshold* nos transistores P.

A potência de curto-circuito depende do período de transição do sinal de entrada. Em circuitos bem projetados, a potência  $P_{sc}$  é, normalmente, em torno de 10% da potência dinâmica  $P_d$ . Desta forma, na maioria dos circuitos integrados, a potência  $P_{sc}$  é desprezada [RAB96b].

### 2.1.3 Potência Estática

Em circuitos CMOS, a potência estática é determinada pela corrente de fuga nos transistores e nas junções P-N (em inglês, *leakage current*). A potência estática ( $P_s$ ) pode ser calculada através da Equação 3 [RAB96b].

$$P_s = \frac{V_{dd} \cdot (I_{d0n} + I_{d1p})}{2} \quad (3)$$

Nesta Equação,  $V_{dd}$  corresponde à tensão de alimentação no circuito;  $I_{d0n}$  corresponde à corrente de fuga nos transistores N quando a porta de entrada do transistor está em nível lógico 0, e  $I_{d1p}$  é a corrente de fuga para nos transistores P quando a porta de entrada do transistor está em nível lógico 1.

Para tecnologias com largura de canal superior a 180 nm, a potência estática pode ser desprezada, pois é muito inferior à potência dinâmica. Nas tecnologias atuais, este fato não ocorre, pois a potência estática é inversamente proporcional ao tamanho dos transistores. Desta forma, a potência estática passa a ser uma componente fundamental na potência global do circuito.

## 2.2 Consumo de Energia

Matematicamente, o consumo de energia de um circuito é a integral da potência dissipada por este ao longo do tempo de operação (Equação 4 [RAB96b]). Desta forma, se um circuito dissipar menos potência durante um mesmo tempo de operação, estará consumindo menos energia naquele intervalo.

$$E_{total} = \int P \cdot dt \quad (4)$$

Tomemos como exemplo um processador executando um algoritmo de tratamento de imagem. Suponha que este processador utilizando todo o seu poder de processamento consegue executar sua aplicação em 1 minuto. Suponha que esta execução consumiu 480 mJ, e precisa-se reduzir este consumo para 240 mJ durante o tempo de execução para economizar bateria. Para que esta exigência seja cumprida o processador deverá dissipar a metade da potência que utilizou anteriormente, reduzindo seu poder de processamento. Ao final do mesmo tempo de execução anterior, este processador irá consumir menos energia, porém poderá não completar a execução do algoritmo de tratamento de imagem. Em resumo, ao se calcular a redução no consumo de energia de um circuito, deve-se observar o tempo total que este utilizou para realizar a tarefa para qual foi projetado.

A redução no consumo de energia afeta diretamente o tempo de vida da bateria do dispositivo. A portabilidade do dispositivo também é melhorada com a redução do consumo de energia, visto que o mesmo pode funcionar com baterias menores.

## 2.3 Modelos de Atraso

Baseado no modelo de atraso utilizado, a estimativa da potência dissipada por um circuito pode levar em conta as transições de estado, e/ou as transições espúrias (*hazards* e *glitches*) [RAB96a][PED96]. Como descrito em [BEN94], a parcela da dissipação de potência devido a transições espúrias na potência total varia significativamente (de 9% a 38%), e o valor médio desta taxa está entre 15-20%. A atividade espúria é bem maior em alguns tipos de módulos, como somadores e multiplicadores. Por exemplo, a potência devido a transições espúrias em um multiplicador de 32 bits com pipeline é pelo menos três vezes maior que a potência devido às transições de estado [DIN95].

Técnicas atuais de estimativa de potência podem utilizar modelos de porta lógicas com atraso nulo (do inglês, *zero-delay*) ou com atraso real (do inglês, *real-delay*). No primeiro modelo, assume-se que todas as mudanças nas entradas propagam-se instantaneamente para as saídas das portas lógicas. O segundo modelo emprega para cada porta lógica um atraso finito, levando em consideração as transições espúrias. A utilização do modelo de atraso real aumenta significativamente os requisitos de computação para se estimar a dissipação de potência, aumentando, contudo, a precisão das estimativas.

## 2.4 Clock-gating e Power-gating

A rede de distribuição do sinal de relógio é responsável por mais de 40% da dissipação de potência de um circuito CMOS típico [DON03], já que os sinais de relógio operam na maior frequência de chaveamento que qualquer outro sinal e possuem o maior *fanout*. O projeto da árvore de relógio do circuito é crítico não somente para o desempenho do circuito, mas também para a dissipação de potência deste.

O sinal de relógio é indispensável em circuitos síncronos. Porém, este nem sempre é necessário todo o tempo em todo o circuito. O sinal de relógio pode ser desconectado momentaneamente de partes inativas do circuito, diminuindo a dissipação de potência global do circuito. A técnica que desconecta o sinal de relógio em algumas partes do circuito é denominada de *clock-gating*, e é largamente utilizada em circuitos CMOS.

*Clock-gating* pode reduzir significativamente a atividade de chaveamento nos sinais de relógio e em outros sinais do circuito. Esta técnica é considerada a mais eficaz na redução da dissipação de potência nos níveis RTL e arquitetural. Contudo, se esta técnica for aplicada de forma indiscriminada poderá impactar de forma negativa a dissipação de potência nos sinais de relógio. Se os *flip-flops* que dividem a lógica de *clock-gating* em um mesmo domínio de relógio estão distribuídos em regiões distantes do circuito integrado, um acréscimo significativo de fios é inserido na árvore de distribuição de relógio, já que cada domínio de relógio deve ser distribuído independentemente, utilizando fiação dedicada. Como resultado, as fontes de relógio de cada domínio serão carregadas com uma capacitância maior e a dissipação de potência poderá aumentar, mesmo que a atividade de chaveamento diminua [OH01][GAR99].

Da mesma forma que o sinal de relógio pode ser desligado de partes do circuito que não estão sendo utilizadas, as próprias células que compõem estas partes podem ser desconectadas da alimentação. Este é o princípio da técnica de *power-gating*, que desconecta as linhas de alimentação de blocos não utilizados, ligando/desligando interruptores de alimentação inseridos entre as linhas de  $V_{dd}$  e os blocos ou entre as linhas de GND e os blocos. Este conceito vem sendo aplicado a blocos de circuito com diversas granularidades, como núcleos processadores [ISH05], unidades de execução de processadores [HU04][AGA07] e até portas lógicas primitivas [USA06].

É necessário compreender os impactos positivos e negativos ao se utilizar a técnica de *power-gating*. A transição entre os estados ativo (com o circuito ativado) e de espera (com o circuito desativado) acarreta uma perda de desempenho, e ao ligar/desligar interruptores de alimentação ocorre uma maior dissipação de potência. A utilização desta técnica deve levar em conta os fatores tempo de transição e potência dissipada na transição para definir quando utilizá-la com reais benefícios.



### 3 TRABALHOS RELACIONADOS

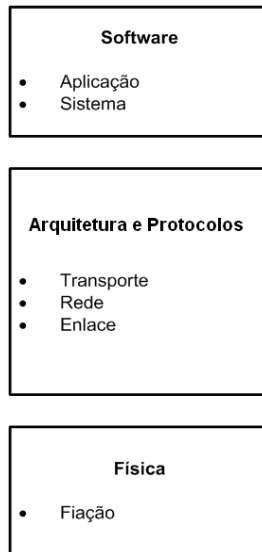
---

Este Capítulo apresenta uma revisão dos trabalhos relacionados à modelagem da dissipação de potência em redes intra-chip e arquiteturas de redes intra-chip que utilizam técnicas de redução da dissipação de potência.

A Seção 3.1 realiza um estudo sobre a dissipação de potência em SoCs que utilizam uma NoC como meio de interconexão de seus IPs. A Seção 3.2 descreve trabalhos relacionados à modelagem da dissipação de potência em NoCs, e a Seção 3.3 descreve trabalhos relacionados à NoCs que utilizam técnicas de redução da dissipação de potência.

#### 3.1 Dissipação de Potência em NoCs

A dissipação de potência em um SoC que utiliza uma NoC como seu meio de comunicação, pode ser descrita em três camadas principais: a física, a de arquitetura e protocolos de controle, e a de software [BEN01]. Cada camada é subdividida em subcamadas (como mostra a Figura 3).



**Figura 2 – Modelo de pilha de protocolos para um SoC que utiliza uma NoC como estrutura de comunicação.**

A camada física corresponde à implementação física dos canais de comunicação. Técnicas como transmissão via pares diferenciais [BON01] e codificação para redução de atividade de chaveamento nos canais [PAL07] são utilizadas para redução na dissipação de potência desta camada. A atividade de chaveamento entre *bits* de *flits* consecutivos influencia diretamente a dissipação de potência. Logo, técnicas de codificação que reduzem esta atividade de chaveamento, podem reduzir a potência dissipada. Outro fator importante nesta camada é a sincronização. A comunicação intra-chip tradicional é baseada na sincronização entre os diversos núcleos através de um sinal de sincronização global (e.g. o sinal de relógio). Sinais de relógio são ineficientes do ponto de vista do consumo de energia, tendo sido demonstrado que estes são responsáveis por uma fração

significativa da dissipação de potência em sistemas síncronos. Protocolos alternativos de sincronização intra-chip, que não necessitam a presença de um relógio global, foram propostos, por exemplo, em [ZHA99][BAI01][PON08], e reduzem a dissipação de potência em uma NoC.

Na segunda camada, a arquitetura especifica a topologia e a organização física da interconexão da rede, enquanto que os protocolos especificam como utilizar os recursos da rede durante a operação do sistema. A subcamada de enlace abstrai a camada física como um enlace digital não-confiável, onde a probabilidade de erro em um *bit* é não nula (crescendo à medida que a tecnologia evolui). Além disso, deve ser estabelecido um compromisso entre confiabilidade na transmissão e o consumo de energia [HEG00]. Uma boa forma de se lidar com os erros de comunicação na camada física é empacotar os dados. Se os dados são enviados por um canal não-confiável em pacotes, a correção e recuperação dos erros se tornam mais fáceis, pois o efeito dos erros estará contido nos limites do pacote e a recuperação dos erros pode ser realizada pacote a pacote ou mesmo flit a flit. Nesta subcamada, o desafio principal é atingir um nível desejado de confiabilidade na comunicação com a menor dissipação de potência possível. Códigos de correção de erros e códigos de detecção de erros com retransmissão são duas técnicas utilizadas para aumentar a confiança na comunicação. Ambas as técnicas são baseadas na transmissão de informação redundante no enlace de dados. Porém, a correção de erros é geralmente mais exigente em termos de redundância e complexidade na decodificação. Desta forma, espera-se que o consumo de energia na transmissão dos dados da técnica de correção de erros seja maior. Em contrapartida, quando ocorre um erro, a técnica de detecção de erros necessita da retransmissão dos dados corrompidos. Dependendo da arquitetura da rede, a retransmissão pode ser dispendiosa em termos do consumo de energia (e perda de desempenho).

Na subcamada de rede, a transmissão dos pacotes pode ser personalizada pela escolha de algoritmos de roteamento (determinísticos ou não-determinísticos) e modos de chaveamento (*store-and-forward*, *virtual cut-through* e *wormhole*). O chaveamento e o roteamento afetam diretamente o desempenho e o consumo de energia em uma NoC. Na subcamada de transporte, alguns algoritmos propõem a decomposição das mensagens em pacotes na origem, e a remontagem destes no destino. A granularidade do empacotamento é uma decisão crítica de projeto, pois o comportamento dos algoritmos de controle da rede é muito sensível ao tamanho dos pacotes. A subcamada de transporte se encarrega da otimização do uso dos recursos da rede e de prover requisitos de qualidade de serviço (*QoS*) vindos da camada de software.

A camada de software compreende o sistema operacional e as aplicações. O sistema operacional abstrai o hardware para a aplicação, e esta terá que executar operações de envio/recebimento de mensagens. O sistema operacional deve dar suporte a esquemas do tipo gerência dinâmica de potência (do inglês, *dynamic power management* ou DPM) além de gerência dinâmica do fluxo de informação (do inglês, *dynamic information-flow management*). Uma forma de gerência de potência bem conhecida é o ajuste dinâmico de tensão e frequência (do inglês, *dynamic voltage/frequency scaling* ou DVFS), muito utilizado em processadores modernos. Estas técnicas selecionam o estado de consumo de energia apropriado, de acordo com uma carga de

trabalho.

### 3.2 Modelagem da Dissipação de Potência e Consumo de Energia em NoCs

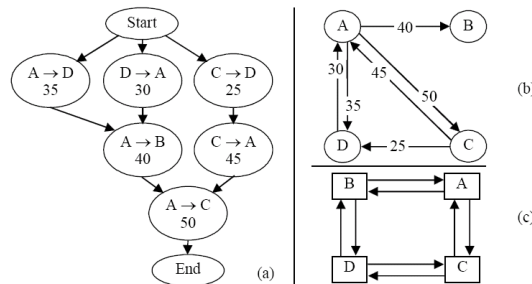
**Hu e Marculescu.** [HU03] propõem um modelo de potência para os macro-blocos de uma NoC (roteador e fios de conexão). Para este modelo, os Autores utilizam o conceito de energia de *bit* (*bit energy*) [YE02], que representa o gasto de energia quando um *bit* de dados é transportado através das interconexões e roteadores da NoC. Neste modelo, o consumo médio de energia no envio de um *bit* de dados, em uma transmissão fim-a-fim, entre dois pontos da NoC é dado pela Equação 5.

$$E_{bit}^{hops} = n_{hops} \times E_{S_{bit}} + (n_{hops} - 1) \times E_{L_{bit}} \quad (5)$$

Onde:  $E_{S_{bit}}$ ,  $E_{L_{bit}}$  e  $n_{hops}$  representam, respectivamente, o consumo de energia por 1 *bit* em um roteador, nos fios de interconexão e o número de roteadores por onde o *bit* passou.

**Marcon et al.** [MAR05], propõem um modelo baseado na dependência da comunicação e da computação entre os núcleos de uma rede intra-chip. Os Autores utilizam a proposição de Hu e Marculescu [HU03], que afirmam que usando algoritmos de mapeamento de aplicações pode-se reduzir em mais de 60% do consumo de energia quando comparado a mapeamentos aleatórios.

Os Autores de [MAR05] propõem um modelo da dependência da comunicação e computação baseado em um grafo, denominado de grafo da dependência da comunicação e computação, ou CDCG (do inglês, *communication dependence and computation graph*). Seja  $C$  o conjunto de núcleos de uma determinada aplicação, o CDCG ((a) na Figura 3) de uma aplicação é um grafo dirigido  $\langle P, D \rangle$ . Neste grafo, cada vértice de  $P$  representa um pacote trocado entre um dado par de núcleos que se comunica durante a aplicação. Existem também dois vértices especiais chamados de *Start* e *End*. O conjunto de arestas  $D$  contém todas as dependências de comunicação na aplicação. Os elementos de  $P$  são quádruplas com a forma  $P_{abq} = (c_a, c_b, t_{aq}, w_{abq})$ , onde  $c_a, c_b \in C$ , e  $p_{abq}$  é o  $q$ -ésimo pacote enviado de  $c_a$  para  $c_b$ . Este pacote contém  $w_{abq}$  *bits* e é transmitido após computado o tempo  $t_{aq}$  no núcleo de origem ( $c_a$ ). O conjunto de todos os pacotes enviados de  $c_a$  para  $c_b$  é denominado  $P_{ab}$ .



**Figura 3 - Exemplos de CDCG (a), CWG (b) e CRG (c).**

O CDCG representa a comunicação e a computação para uma aplicação composta por um

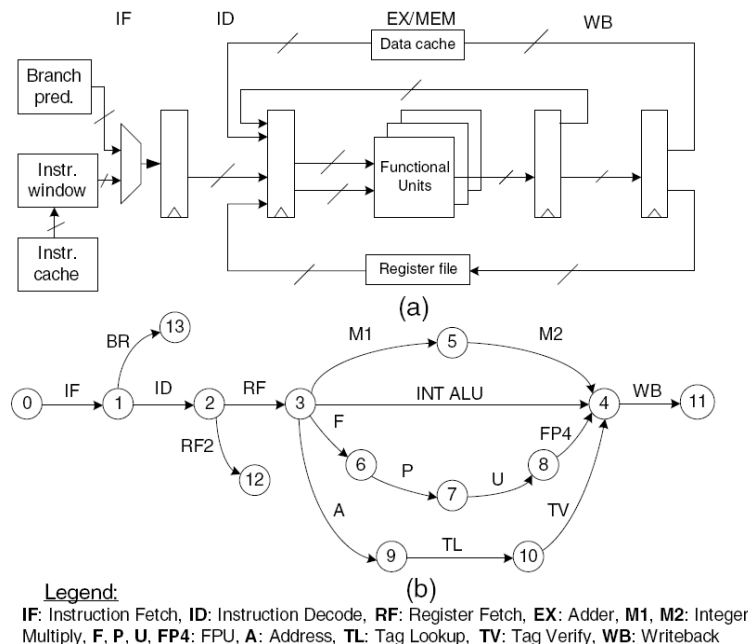


número arbitrário de núcleos. A direção das arestas neste grafo denota que o vértice de computação destino depende da computação do vértice da origem. Em outras palavras, o vértice destino apresenta uma dependência de comunicação com o vértice origem.

Os Autores de [MAR05] realizam uma comparação entre o mapeamento utilizando o modelo baseado em CDCG e o modelo de Hu e Marculescu, baseado em um grafo de comunicação com pesos, ou CWG (do inglês, *communication weighted graph*) [MAR05]. Este CWG ((b) na Figura 3) é um grafo dirigido  $\langle C, W \rangle$ , onde o conjunto de vértices  $C = (c_1, c_2, c_3, \dots, c_n)$  representa o conjunto de núcleos em uma aplicação. Assumindo que  $w_{ab}$  representa o número de *bits* em todos os pacotes enviados de um núcleo  $c_a$  para outro núcleo  $c_b$ , o conjunto de arestas  $W$  é  $\{(c_a, c_b) | c_a, c_b \in C \text{ e } w_{ab} \neq 0\}$ , e cada aresta é rotulada com o valor  $w_{ab}$ .  $W$  representa toda a comunicação entre os núcleos da aplicação, enquanto o CWG revela informações sobre o volume de comunicação relativa da aplicação.

Os Autores de [MAR05] realizam avaliações de consumo de energia e latência da aplicação utilizando os dois modelos de mapeamento da aplicação em diferentes núcleos. Foi verificado um ganho de até 20% de redução no consumo de energia na comparação entre os dois modelos avaliados.

**Eisley et al.** [EIS06] propõem um modelo para avaliar o consumo de energia em um MPSoC como um todo, medindo o consumo nos núcleos de processamento e no meio de comunicação simultaneamente. O objetivo deste modelo é criar perfis de consumo de energia em tempo de projeto. Os Autores fazem o mapeamento de um núcleo de processamento para um grafo. A seguir modelam as instruções deste núcleo de processamento como mensagens mapeadas neste grafo (como mostra a Figura 4).

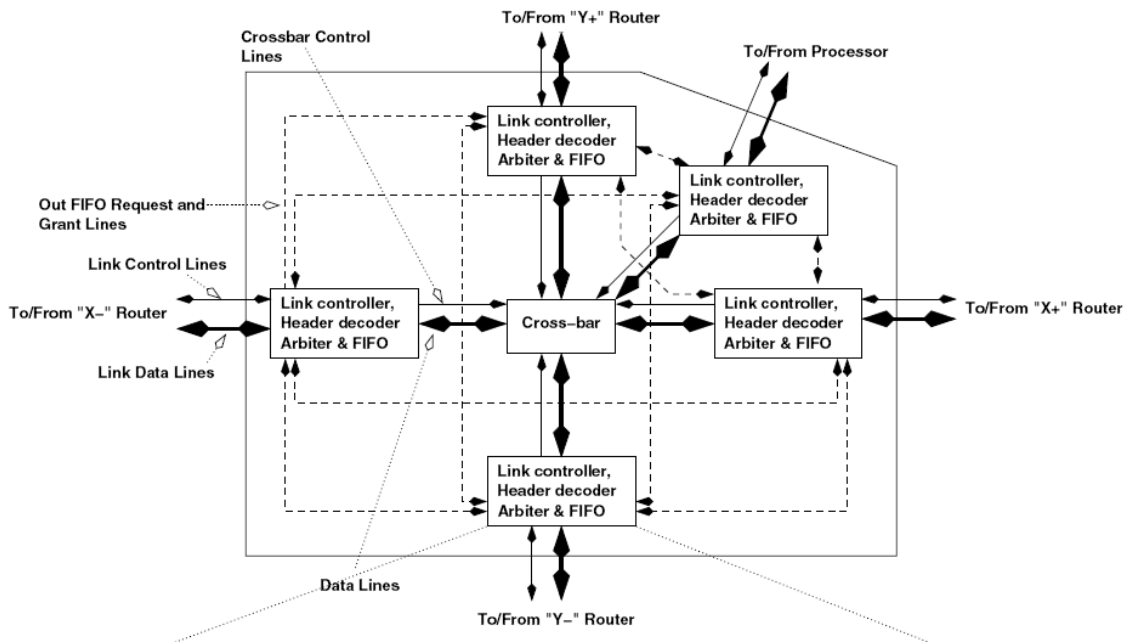


**Figura 4 – Modelagem de um MPSoC sobre a estrutura de um grafo; (a) Diagrama de blocos típico de uma arquitetura de microprocessador; (b) Representação do grafo correspondente.**

Os Autores conectam o grafo do elemento de processamento à estrutura do roteador (ao meio de interconexão propriamente dito). Com todos os processadores descritos como um grafo e conectados ao meio de conexão, que é uma rede intra-chip, o MPSoC foi completamente modelado como um grande grafo. A ferramenta LUNA [EIS04] é utilizada para determinar a dissipação de potência em um grafo, e ela é utilizada para avaliar o consumo deste MPSoC modelado como um grafo.

**Banerjee et al.** [BAN04] criam modelos da dissipação de potência para cada um dos elementos de uma NoC individualmente. Estes modelos partem da descrição RTL da NoC, contendo seus diversos roteadores e a interconexão entre eles. A fase seguinte corresponde à análise do *netlist* gerado após a síntese física. O consumo de energia de cada um dos fios que interconectam os roteadores pode ser gerado a partir do consumo de apenas um fio, sendo após expandido para  $n$  fios.

Em seguida são especificados modelos para cada um dos seus sub-módulos. Para esta avaliação foi usada uma estrutura de um roteador *wormhole* genérico, como mostra a Figura 5. Estes diversos sub-módulos foram reduzidos aos seus blocos básicos (como multiplexadores, registradores, somadores, comparadores e portas lógicas) para a avaliação de potência através de simulação elétrica (mostrada na Tabela 1). O que se obteve foi o consumo de energia em cada um destes diversos blocos. Somando-se todos estes consumos, obtém-se o consumo total deste sub-módulo e somando-se os consumos de todos os sub-módulos obtém-se o consumo de todo o roteador.



**Figura 5 - Arquitetura genérica de um roteador *wormhole*.**

Os Autores concluem o artigo afirmando que o modelo apresentado não contempla NoCs com detecção/correção de erros, roteamento adaptativo e esquemas de controle de fluxo. Além disto, afirmam que os resultados obtidos pelo modelo correspondem apenas à tecnologia 0,18  $\mu\text{m}$ .

**Tabela 1 - Consumo de energia de cada um dos blocos básicos de um determinado módulo.**

Unit Full Adder		2-bit Comparator	
1-bit flip at the output	0.096 pJ	Output Transition	0.15 pJ
2-bit flip at the output	0.1608 pJ	Input Change but no output change	0.0708 pJ
Input Change but no output change	0.0552 pJ	Leakage	0.077 fJ
Leakage	0.00438 fJ		
2-1 Multiplexer		D Flip-Flop	
Output Transition	0.061 pJ	Output Transition	0.1089 pJ
Input Change but no output change	1.527 fJ	Input Change but no output change	0.014 pJ
Leakage	0.013 fJ	Leakage	0.034 fJ

**Palma et al.** [PAL07] utilizam um modelo da dissipação de potência baseado na atividade chaveamento na transmissão dos pacotes na rede. A dissipação de potência de um roteador é a soma das dissipações de potência nos *buffers*, no *crossbar* e na lógica de controle.

O fluxo utilizado pelos Autores para obter a dissipação de potência utiliza três passos. O primeiro inicia com a descrição VHDL da NoC (sem qualquer esquema de codificação da transmissão dos dados) e com arquivos de tráfego, ambos obtidos através de um ambiente personalizado de geração automática de NoCs e tráfego de rede [OST05]. Os arquivos de tráfego injetam pacotes na rede através das portas locais dos roteadores, simulando o comportamento dos núcleos locais. Um simulador VHDL aplica sinais de entrada sobre a NoC ou em qualquer módulo da NoC, seja este um roteador ou um módulo interno do roteador (*buffer* de entrada ou lógica de controle). A simulação produz registros dos sinais, armazenando as variações dos valores lógicos de cada sinal em um arquivo. Estes arquivos são convertidos em estímulos elétricos e depois utilizados em uma simulação elétrica (terceiro passo).

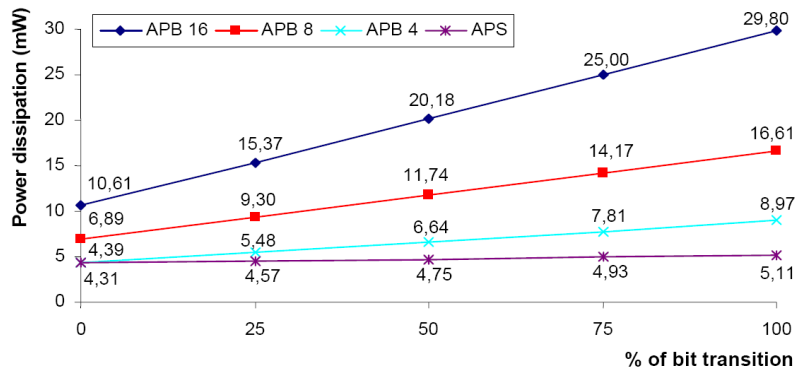
No segundo passo, o módulo a ser avaliado (e.g. um *buffer* de entrada) é sintetizado utilizando a ferramenta Leonardo Spectrum associada a uma tecnologia 0.35  $\mu\text{m}$ . Ao final da execução da ferramenta, um *netlist* HDL é gerado, e convertido para uma *netlist* SPICE, utilizando um conversor desenvolvido pelos Autores.

O terceiro passo consiste na simulação SPICE do módulo em análise. Neste passo, é necessário realizar a integração da *netlist* SPICE do módulo com a entrada dos sinais elétricos e da biblioteca de portas lógicas descrita pelo *netlist* SPICE. A informação elétrica resultante permite a aquisição dos parâmetros da dissipação de potência da NoC para um determinado tráfego.

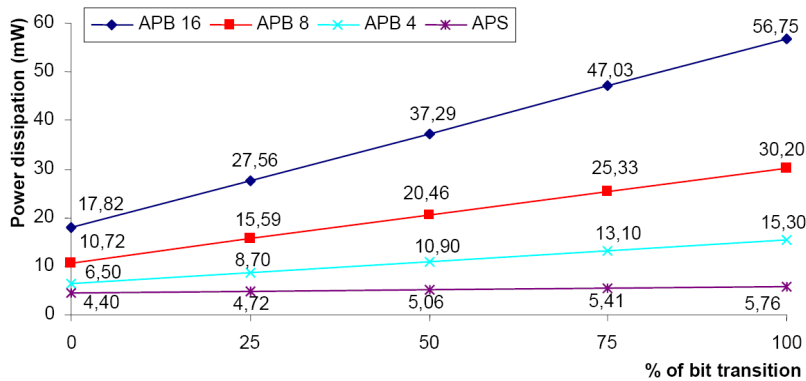
A dissipação de potência média por salto (do inglês, *Average Power per Hop*, ou APH), definido em [PAL05], é utilizada para denotar a média da dissipação de potência dinâmica em um único salto de um pacote transmitido através da NoC. O APH pode ser dividido em três componentes: potência média dissipada pelos *buffers*, pelos fios e pelo chaveamento das portas lógicas de um roteador (APR); potência média dissipada pelos fios entre os roteadores (APL); e potência média dissipada pelos fios que ligam o roteador e o seu núcleo local (APC). A Equação 6 mostra a potência média dissipada por um pacote transmitido através do roteador, conexão com o núcleo local e a conexão entre dois roteadores.

$$APH = APR + APL + APC \quad (6)$$

A análise apresentada em [PAL05] mostra que para melhor entender a dissipação de potência no roteador (APR), deve-se dividi-la nas dissipações de potência dos *buffers* (APB) e no controle (APS). Isto ocorre, pois a dissipação de potência devido à transição de *bits* é mais significativa nas estruturas de *buffer* do que na lógica de controle. As Figura 6 e Figura 7 mostram este efeito para *buffers* com 4, 8 e 16 *flits* de profundidade e para uma lógica de controle centralizada para *flits* de 8 e 16 bits respectivamente. Os gráficos apresentam a dissipação de potência em função da quantidade de transição dos *bits* em um pacote com 128 *flits* (100% = 127 transições de *bit* em cada um dos fios de entrada do roteador).



**Figura 6 - Análise do efeito da transição dos *bits* na dissipação média de potência para diferentes profundidades de *buffer* e para uma lógica de controle centralizada em uma rede Hermes com *flits* de 8 bits [PAL07].**



**Figura 7 - Análise do efeito da transição dos *bits* na dissipação média de potência para diferentes profundidades de *buffer* e para uma lógica de controle centralizada em uma rede Hermes com *flits* de 16 bits [PAL07].**

**Matsutani et al.** [MAT08] utilizam um modelo para avaliar a dissipação de potência em NoCs baseado em simulações e ferramentas comerciais. Os Autores sintetizam a NoC sob análise utilizando a ferramenta *Design Compiler* da *Synopsys*, a seguir projetam a árvore de distribuição do sinal de relógio do circuito através da ferramenta *Astro* da *Synopsys*. De posse da descrição *netlist* Verilog do circuito desenvolvido, os Autores realizam a simulação deste utilizando a ferramenta Verilog-XL da Cadence a fim de extrair a atividade de chaveamento do circuito. Ao final da simulação os Autores utilizam a ferramenta *Power Compiler* da *Synopsys* para realizar a estimativa da dissipação de potência do circuito em análise.

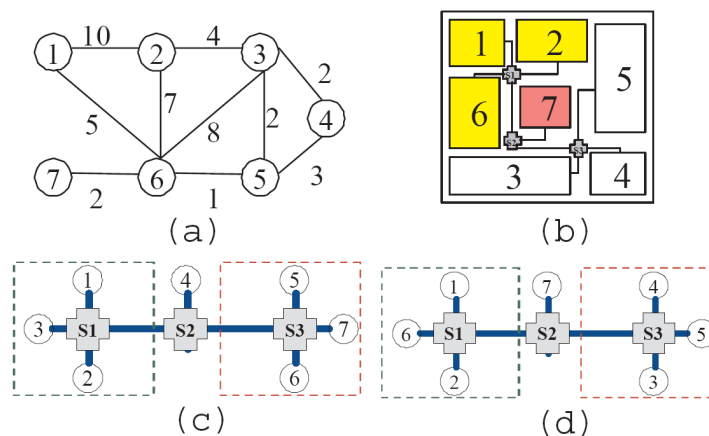
A Tabela 2 resume as características dos trabalhos apresentados nesta Seção.

**Tabela 2 – Tabela resumo dos trabalhos relacionados a modelos de estimativa da dissipação de potência em NoCs.**

Autor	Comentário dos modelos
Hu e Marculescu. [HU03]	Modelo de potência/energia pelo volume de dados transmitidos.
Marcon et al. [MAR05]	
Eisley et al. [EIS04]	Modelo da dissipação de potência em um MPSoC visto como um grafo.
Banerjee et al. [BAN04]	Modelagem da dissipação de potência no roteador baseado em simulação SPICE.
Palma et al. [PAL07]	
Matsutani et al. [MAT08]	Estimativa de potência baseada em fluxo comercial, utilizando ferramentas de diferentes empresas.

### 3.3 NoCs de Baixa Dissipação de Potência

Chang et al. [CHA05] apresentam uma estrutura de interconexão entre núcleos chamada de *crossroad*. Esta conexão é muito similar a uma estrutura de barramento segmentado, dividindo o meio de comunicação em regiões. As regiões contêm núcleos com alta densidade de comunicação. Os Autores partem de um grafo de comunicação entre processos (parte (a) na Figura 8), até chegar a uma topologia em que os processos que mais se comunicam estejam mais próximos uns dos outros (parte (d)). Os Autores desenvolveram uma ferramenta que analisa o grafo de comunicação, realiza um agrupamento (*clusters*) inicial dos processos na rede *crossroad* (parte (c)), e a seguir realizam otimizações para reduzir a comunicação entre processos montando a topologia final. Um exemplo do posicionamento físico dos núcleos na rede, segundo a topologia gerada, é exibido em (b).

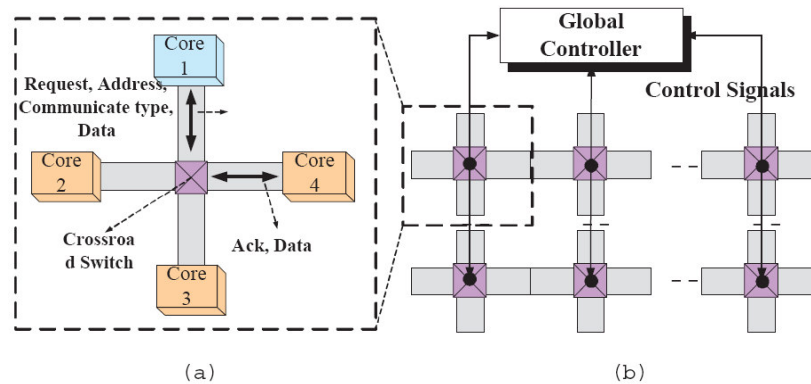


**Figura 8 – Refinamentos para criar uma rede *crossroad*; (a) Grafo de comunicação entre os núcleos; (b) Posicionamento final dos núcleos na rede; (c) Posicionamento inicial dos núcleos; (d) Posicionamento otimizado dos núcleos.**

O elemento que interconecta os núcleos entre si é denominado CCB (*crossroad communication block*). Este elemento é o roteador da rede, realizando o roteamento e a arbitragem das mensagens (parte (a) na Figura 9). Cada CCB pode se conectar a quatro dispositivos distintos, estes podendo ser núcleos ou outros CCBs. Um CCB pode transferir duas comunicações locais paralelas. Uma rede de CCBs é administrada através de um controlador global (parte (b)). Este controlador está conectado a todos os CCBs, e realiza políticas de controle e fluxo dos dados entre

CCBs tais como algoritmos de roteamento livres de *deadlock* e algoritmos anti-contenção. Os Autores utilizam o modelo de consumo de energia em NoCs proposto por Hu e Marculescu. [HU03].

Os Autores concluem o artigo afirmando que sua nova arquitetura é específica para aplicações onde haja muita comunicação envolvida entre diversos núcleos de processamento. A arquitetura proposta reduz a dissipação de potência, diminuindo o custo de comunicação entre os núcleos, agrupando-os em estruturas de *clusters*.

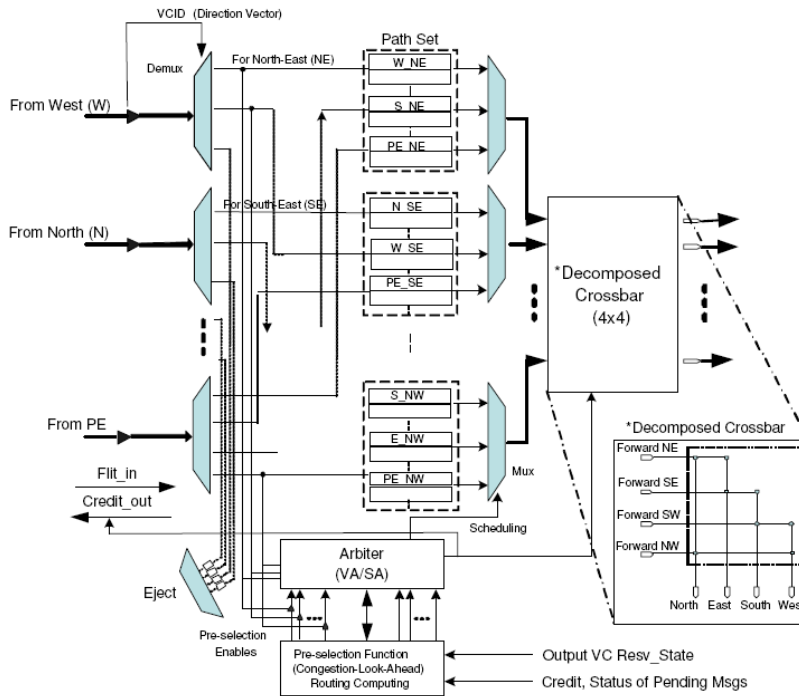


**Figura 9 – Estrutura de comunicação Crossroad; (a) Roteador CCB interconectando 4 núcleos, e (b) a interconexão entre CCBs formando uma rede.**

**Kim et al.** [KIM05] propõem uma arquitetura de roteador sensível ao caminho dos dados para aplicações que necessitam de baixa latência, além de um modelo de roteador baseado na teoria de filas para avaliar o desempenho e o consumo de energia de NoCs implementadas com este.

O roteador sensível ao caminho dos dados proposto pelos Autores emprega um roteamento do tipo *look-ahead* e alocação de recursos especulativa [DAL03]. Usando informações de tráfego dos roteadores vizinhos e o seu estado atual, o roteador seleciona a direção para o próximo roteador. Esta tarefa é garantida por uma unidade de pré-seleção presente no primeiro estágio do *pipeline* do roteador, o qual trabalha um ciclo adiante da chegada do primeiro *flit*.

As informações presentes no roteamento *look-ahead* permitem que o *flit* com destino à porta local (PE) seja repassado depois do demultiplexador (*demux*), ao invés de atravessar toda a lógica do roteador. Este repasse mais cedo economiza dois ciclos de relógio no nodo destino, ao evitar a alocação e os estágios transversais do roteador. A arquitetura do roteador sensível ao caminho pode ser visualizada na Figura 10.



**Figura 10 - Arquitetura do roteador sensível ao caminho de dados.**

Outra característica do roteador é o uso de uma estrutura de *crossbar* 4x4 decomposta, que possui metade das conexões de um *crossbar* completo 5x5. Esta arquitetura consegue explorar as vantagens do uso de topologias criadas visando o roteamento proposto.

Os Autores propõem um modelo matemático para o cálculo da dissipação de potência em um roteador, a partir da equação 7.

$$P_R = \sum_{r \in PE} (P_{r\_buf} + P_{r\_arbiter} + P_{r\_crossbar} + P_{r\_link}) \quad (7)$$

Onde:  $P_{r\_buf}$  é a dissipação média de potência dos *buffers*, incluindo as dissipações dinâmica e estática;  $P_{r\_arbiter}$  é a potência dos módulos VA e AS (árbitro);  $P_{r\_crossbar}$  é a potência média gasta no *crossbar*;  $P_{r\_link}$  é a potência média gasta na conexão entre dois roteadores vizinhos.

Os Autores concluem o artigo afirmando que foi desenvolvido um modelo analítico eficiente para avaliar os comportamentos de desempenho, tolerância a falhas e potência em NoCs. Este modelo é baseado na teoria de filas para redes do tipo malha de duas dimensões (2D), o qual consegue realizar avaliações de desempenho e potência na granularidade de sub-módulos de um roteador.

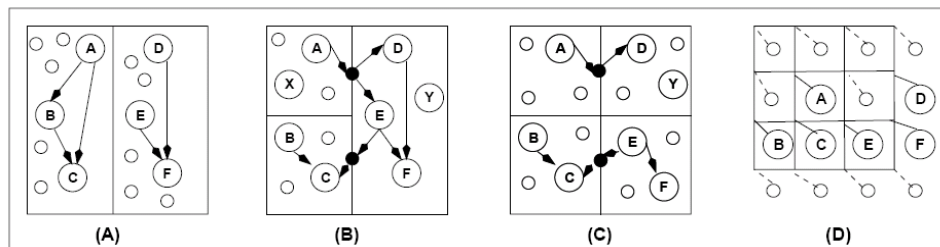
**Srinivasan e Chatha.** [SRI05] propõem uma ferramenta que aplica técnicas de mapeamento e roteamento dos núcleos de processamento a fim de reduzir a energia consumida nas comunicações entre núcleos. O problema do projeto de NoCs baseadas em uma interconexão de rede do tipo malha 2D é ilustrado na Figura 11. A entrada do problema é um grafo dirigido chamado de grafo de comunicação. Cada nodo neste grafo representa um núcleo de computação ou de armazenamento, e as arestas representam a comunicação entre estes núcleos. Cada aresta é descrita como  $C_m(B,L)$ , onde  $m$  representa um identificador de conexão,  $B$  representa a largura de

banda requerida, e  $L$  representa a restrição de latência. Os requisitos de largura de banda e latência de cada conexão podem ser facilmente obtidos a partir do desempenho desejável para a aplicação e da latência individual de cada tarefa de cada núcleo. A saída do problema é um mapeamento dos núcleos em diferentes roteadores, uma NoC com topologia malha correspondente, e uma rota de pacotes estática para cada conexão entre dois núcleos tal que a energia total de comunicação seja reduzida.



**Figura 11 - Mapeamento e roteamento de baixo consumo de energia.**

Os Autores propõem uma técnica para projetar arquiteturas de interconexão intra-chip baseadas em malha 2D (do inglês, *mesh based on-chip interconnection architectures* - MOCA) de baixo consumo de energia. O projeto de MOCA é realizado em duas fases. Na primeira fase, descrita em (A), (B) e (C) na Figura 12, é aplicado um particionamento do tipo *min-cut* sobre o grafo de comunicação. Este procedimento é realizado recursivamente até que se obtenha uma divisão deste grafo em regiões de altas taxas de comunicação. Na segunda fase ((D) na Figura 17) são gerados os caminhos de dados sobre a rede derivada na primeira fase. É realizado um roteamento hierárquico sobre as regiões descritas na primeira fase, além dos caminhos entre as diversas regiões.



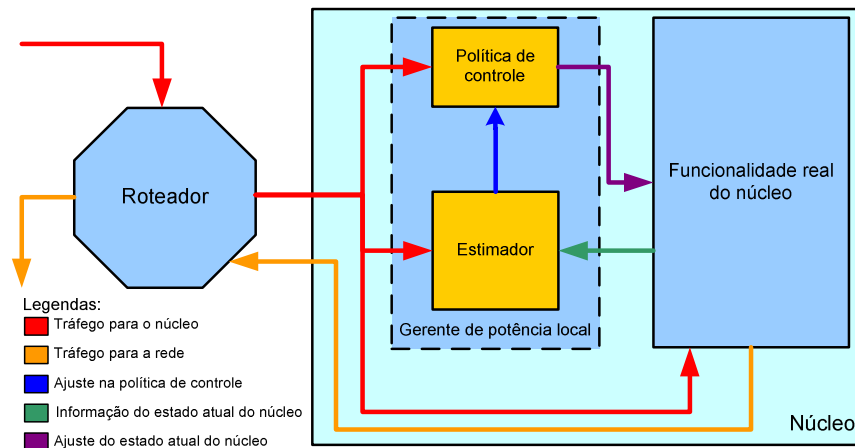
**Figura 12 – Projeto de MOCA. (A) representa a entrada da primeira fase da MOCA. (B) e (C) representam a ação da técnica de refinamento baseado em árvore de cortes. (D) representa a segunda fase da do projeto de MOCA.**

Os Autores concluem o artigo afirmando que a complexidade algorítmica do projeto de MOCA é menor que a do NMAP [BER05], outra técnica para o mapeamento de núcleos em uma NoC. O projeto de MOCA leva em consideração os requisitos de latência de cada roteador ao gerar um mapeamento, além de gerar uma topologia de rede válida para requisitos de latência estritos.

**Simunic e Stephen.** [SIM02] abordam uma política de gerência da dissipação de potência baseado em um modelo de malha fechada, utilizado em sistemas de controle. A cada núcleo de processamento conectado à porta local de um roteador de uma NoC é associado um gerente de



potência local (arquitetura apresentada na Figura 13). Este gerencia perfis de consumo pré-determinados na fase de análise de restrições de consumo do projeto durante a fase de execução.



**Figura 13 – Gerente local de potência.**

Cada gerente de potência local possui um estimador e um controlador. O estimador observa as mudanças no estado do núcleo local, monitora o tráfego que chega neste núcleo e pedidos de gerência de consumo de energia vindos de outros núcleos da rede. Baseado nestas mudanças o controlador recalcula o estado (de potência e desempenho) do núcleo local. Ao executar qualquer mudança, o estimador local envia seu novo estado ao gerente global, atualizando-o com a nova estimativa.

Os Autores concluem o artigo afirmando que os estimadores conseguiram uma boa taxa de acertos e que esta nova abordagem de gerência de consumo é viável para diversas implementações de NoCs.

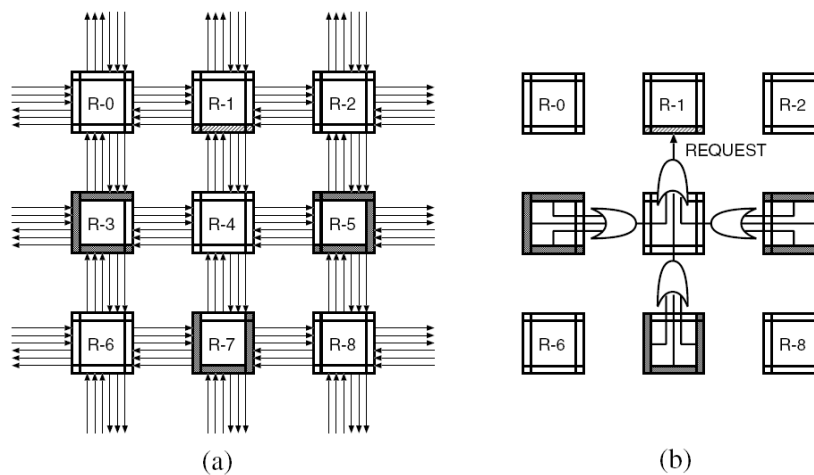
**Palma et al.** [PAL07] utilizam técnicas de codificação da carga útil dos pacotes da rede para reduzir o consumo de energia gerado pela atividade de chaveamento na transmissão. Foi utilizada a rede Hermes como estudo de caso. Experimentos realizados sobre esta rede mostraram que as transições dos *bits* afetam a dissipação de potência em até 370% para os fios de conexão entre os roteadores, 180% para os *buffers* de entrada dos roteadores e 16% para a lógica de controle do roteador.

Uma forma de reduzir a dissipação de potência em NoCs, nos fios de conexão e a lógica, é utilizar esquemas de codificação dos *bits* do pacote para se reduzir a atividade de chaveamento na transmissão. Os Autores concluem o artigo afirmando que a partir dos resultados obtidos pode-se perceber que a redução da potência dissipada nos canais entre os roteadores é muito menor que na lógica do roteador. Entretanto, em tecnologias mais recentes, a dissipação de potência nos canais é muito mais relevante [SYL01]. Neste cenário, a adoção de esquemas de codificação deve ser mais vantajosa, pois são desenvolvidos para atuar sobre os canais de comunicação.

**Matsutani et al.** [MAT08] introduzem o conceito de modo de espera (*sleeping mode*) nos roteadores. Este modo utiliza a técnica *power-gating*. Como os roteadores são os atores principais em uma comunicação entre núcleos de processamento ligados a uma NoC, eles devem estar sempre

aptos a receber pacotes mesmo que os núcleos de processamento ligados a estes estejam em um modo de espera, resultando em uma alta dissipação de potência dos roteadores em relação aos núcleos de processamento. A utilização de *power-gating* nos canais de comunicação do roteador em tempo de execução é uma boa solução para reduzir a dissipação de potência nos modos de espera, sem que isto afete a comunicação da NoC como um todo. Entretanto, a transição entre os estados de operação normal e de espera leva a uma perda de desempenho na NoC, e ao se ligar/desligar o roteador pode-se acabar por consumir mais energia, o que significa que em curto prazo o modo de espera no roteador pode acabar por aumentar a dissipação de potência da NoC.

A Figura 14 ilustra os sinais de controle utilizados no modo de espera para o canal de comunicação sul do roteador 1. Digamos que este canal de comunicação seja denominado de canal “destino”. O canal destino monitora os sinais de requisição dos canais de entrada dos roteadores 3, 5 e 7 para decidir se entra em modo de espera ou não. Note que o canal destino não necessita monitorar alguns canais de entrada vizinhos, já que algumas curvas do tipo “U” que introduzem *deadlocks* são geralmente proibidos mesmo quando sob alto tráfego na rede. Sendo assim, o canal destino somente deverá monitorar os canais de entrada descritos em níveis de cinza na Figura 14(a).



**Figura 14 - Sinal de requisição para o canal de comunicação Sul do roteador 1 (R significa roteador); (a) Canais de entrada na NoC; (b) Rede de requisições utilizando portas lógicas do tipo OR.**

Os canais de entrada destacados na Figura 14 ativam seus sinais de requisição para o roteador destino quando estes detectam um pacote destinado ao roteador 1, notificando que um pacote estará chegando ao canal destino em no mínimo cinco ciclos de relógio (Figura 14(b)). Estes sinais de requisição são duas vezes mais longos que a distância entre os roteadores. O atraso nos fios que contém os sinais de requisição irá aumentar na medida em que a distância entre os roteadores aumenta. Isto ocorre, mesmo que o caminho crítico do roteador destino não esteja relacionado aos controladores de modo de espera, pois ele está relacionado com a lógica de arbitragem nas portas de saída do roteador. Por outro lado, estes sinais de requisição podem ser “pipelineizados” e transferidos em dois ciclos de relógio, na tentativa de amortizar o atraso dos fios de requisição. Nestes casos cada canal de comunicação ainda pode detectar a chegada de pacotes com adiantamento de quatro ciclos de relógio.

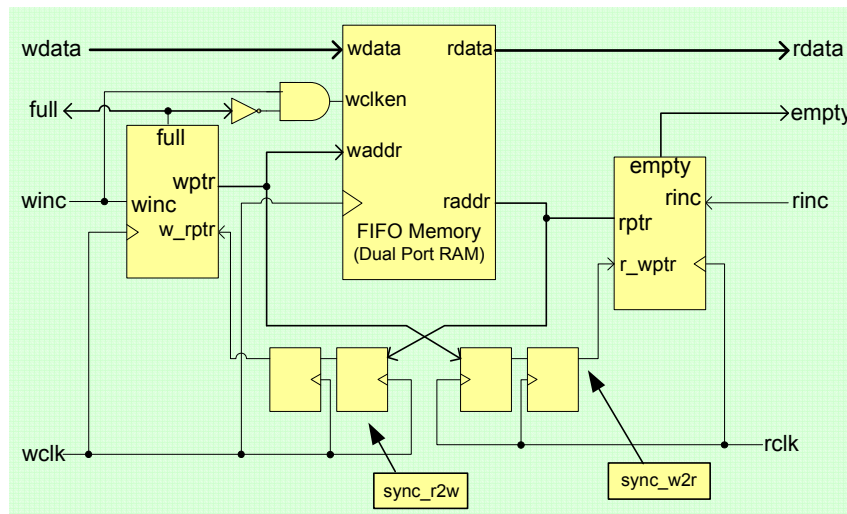
Os Autores propõem a utilização de um algoritmo de roteamento do tipo *look-ahead*, juntamente com a inserção de lógicas de *power-gating* nos roteadores. Os roteadores *look-ahead* necessitam de duas modificações em relação aos originais: (i) todos os canais de entrada necessitam ligar/desligar os sinais de requisição (*request*) de todos os canais de entrada que estão localizados em até dois *hops* distantes destes; (ii) todos os canais de entrada necessitam receber todos os sinais de requisição dos canais de entrada que estão localizados em até dois *hops* distantes destes. Os Autores afirmam que a quantidade de *hardware* adicional para implementar tais modificações é mínima, pois estas podem ser implementadas utilizando simples lógica combinacional, que é, em termos de área, muito menor que *buffers*.

Os Autores realizaram testes com algoritmos *look-ahead* determinísticos, porém uma série de extensões foram propostas para se utilizar algoritmos *look-ahead* adaptativos. Os Autores utilizaram como *benchmarks* um conjunto de tráfego de aplicações distribuídas reais contidas no NAS [BAI95].

Os Autores concluem afirmando que sua metodologia e arquitetura de NoC, reduz a dissipação de potência de estática (*leakage*). A redução desta dissipação de potência nas tecnologias mais modernas, 90nm e 65nm, permite reduzir o consumo de energia total da NoC.

**Pontes et al.** [PON08] utilizam o conceito de ilhas de comunicação síncrona conectadas entre si de forma assíncrona, ou GALS (do inglês, *Globally Asynchronous Locally Synchronous*), para desenvolver duas arquiteturas de roteadores de baixa dissipação de potência. Ambas as arquiteturas, Hermes-G e Hermes-GLP, são derivadas da NoC Hermes.

No desenvolvimento da primeira arquitetura, Hermes GALS ou Hermes-G, os Autores substituíram os *buffers* de entrada dos roteadores da Hermes original por filas bi-síncronas (Figura 15) [CUM02]. Estas permitem a comunicação entre módulos que operam em diferentes frequências de relógio e possuem a característica de realizar operações de escrita e leitura em apenas um ciclo de relógio.



**Figura 15 – Estrutura da fila bi-síncrona adicionada à rede Hermes para sincronização dos dados [CUM02].**

A NoC Hermes-G pode receber as filas bi-síncronas em cada porta de entrada dos seus roteadores, ou apenas nas portas de entrada que ligam um roteador a outro. Esta abordagem permite que cada roteador possua sua própria fonte de relógio ou que opere com o mesmo sinal de relógio que seu núcleo IP local. Para que o roteador possa operar com seu próprio sinal de relógio uma fila bi-síncrona deve ser inserida no núcleo IP, mais precisamente na interface de rede que realiza a conexão entre o núcleo IP e o roteador.

A segunda arquitetura, Hermes GALS *Low-Power* ou Hermes-GLP, utiliza os mesmos conceitos de sincronização aplicados aos roteadores da NoC Hermes-G, porém agrega duas técnicas de redução da dissipação de potência: *clock-gating* e variação dinâmica de frequência (DFS).

A Hermes-GLP aplica a técnica de *clock-gating* no nível do roteador. Um roteador está em estado de espera quando todas suas portas estão inativas, sendo assim não existem dados válidos a serem transmitidos por nenhuma uma destas. Para implementar o *clock-gating*, um módulo de controle de relógio foi adicionado ao projeto do roteador. Este módulo recebe um sinal de cada porta de entrada e atualiza o estado atual de cada porta. Quando não houver mais dados válidos a serem transmitidos por todas as portas, o módulo de controle de relógio desativa o sinal de relógio para o roteador. O roteador irá permanecer neste estado de espera até que uma operação de escrita seja realizada em uma das portas de entrada, atualizando esta porta para o estado de transmissão.

A técnica de DFS foi implementada utilizando um módulo assíncrono simples de troca dinâmica entre dois ou mais sinais de relógio, livre de *glitches*. A utilização de DFS permite que os roteadores realizem a escolha de um sinal de relógio adequado, baseado nos requisitos de comunicação a cada instante. Roteadores envolvidos em uma comunicação de alta prioridade devem operar em uma frequência de relógio maior que outros envolvidos em uma comunicação de mais baixa prioridade. Para permitir a configuração dinâmica do sinal de relógio, cada pacote leva a sua prioridade consigo através de um sinal adjacente (*sideband*). Cada roteador recebe um conjunto de  $n$  sinais de prioridade, onde  $n$  é o número total de portas em cada roteador, realiza a computação da prioridade mais elevada e seleciona a melhor frequência de operação baseado neste resultado. Desta forma, as restrições de latência para todos os pacotes são atingidas mais facilmente, enquanto que o mecanismo de DFS produz um pequeno aumento de área para o roteador.

Os Autores utilizam como métrica de avaliação da dissipação de potência a taxa de ativação dos roteadores, já que esta é uma das parcelas que contribuem para o cálculo da dissipação de potência dinâmica da NoC. Os Autores concluem o artigo afirmando que a Hermes-GLP foi desenvolvida para reduzir a dissipação de potência em SoCs que utilizam NoCs. Ela é baseada em um mecanismo de DFS simples que permite uma redução significativa na dissipação de potência da NoC, aliado a uma pequena perda na latência na entrega dos pacotes.

A Tabela 3 resume as características dos trabalhos apresentados nesta Seção.

**Tabela 3 - Tabela resumo dos trabalhos relacionados a técnicas de redução da dissipação de potência em NoCs.**

<b>Autor</b>	<b>Técnica Utilizada</b>	<b>Comentário</b>
Simunic e Stephen [SIM02]	DVFS	Redução da dissipação de potência na NoC através da inserção de módulos "gerentes" que controlam a potência em função do volume de tráfego
Kim et al. [KIM05]	Algoritmo <i>look-ahead</i>	Redução da dissipação de potência na NoC através da utilização de um algoritmo de roteamento.
Srinivasan e Chatha. [SRI05]	Mapeamento de IPs	Redução da dissipação de potência na NoC através do mapeamento de IPs na rede.
Marcon et al. [MAR05]		
Chang et al. [CHA05]	Topologia dedicada	Definição da topologia de acordo com o tráfego.
Palma et al. [PAL07]	Codificação dos <i>bits</i> transmitidos no pacote	Redução do consumo de potência na NoC através da redução da atividade de chaveamento nos enlaces.
Matsutani et al. [MAT08]	Sleep-Mode ( <i>power-gating</i> )	Redução da dissipação de potência na NoC através da utilização de algoritmo de roteamento do tipo <i>look-ahead</i> e <i>power-gating</i> .
Pontes et al. [PON08]	GALS, <i>Clock-Gating</i> e DFS	Redução da dissipação de potência na NoC Hermes através de filas bi-síncronas, <i>clock-gating</i> e variação dinâmica de frequência (DFS).

### 3.4 Conclusão

Os artigos avaliados sobre a modelagem da dissipação de potência em redes intra-chip exibem dois níveis distintos de estimativas: elétrico e baseado em volume de dados nas simulações de mais alto nível de abstração (RTL ou SystemC). As estimativas geradas no nível elétrico possuem uma melhor precisão em relação a volume de dados, porém demandam mais recursos computacionais e tempo para serem computadas. Já as estimativas baseadas em volume de dados possuem menor precisão, e demandam menos tempo e recursos computacionais, porém estão mais propensas a erros nas simulações e desconsideram problemas que possam ocorrer durante as simulações (como o congestionamento da rede).

O modelo de estimativa da dissipação de potência em NoCs, descrito no Capítulo 4, propõe uma alternativa aos dois modelos citados acima. O modelo proposto relaciona as taxas de recepção nos *buffers* de entrada dos roteadores com a potência dissipada nos roteadores, e conseqüentemente pela NoC. Esta nova abordagem é mais precisa que o modelo baseado em volume de dados, como será demonstrado no Capítulo 4, e tempo de processamento inferior à simulação elétrica.

Os artigos avaliados sobre redes intra-chip com baixa dissipação de potência mostram as aplicações de diversas técnicas de redução de potência em circuitos CMOS no projeto de NoCs. Técnicas como *clock-gating*, *power-gating*, codificação dos canais de transmissão de dados, GALS, DFS e DVFS são aplicados ao projeto de NoCs, demonstrando resultados expressivos na redução da dissipação de potência e consumo de energia em NoCs. Além disso, utilizam-se mudanças em alguns parâmetros da rede e mapeamento de aplicações sobre a rede a fim de reduzir o consumo de energia. Parâmetros como profundidade dos *buffers* de entrada, algoritmos de roteamento e

topologia da rede podem afetar o consumo de energia da NoC. O objetivo do trabalho não é propor uma nova arquitetura de roteador para baixa dissipação de potência, mas é estudar a influência da mudança dos parâmetros da NoC na dissipação de potência e consumo de energia, descrito no Capítulo 5.



## 4 MODELO DE ESTIMATIVA DA DISSIPACÃO DE POTÊNCIA EM REDES-INTRA CHIP

---

Este Capítulo descreve a primeira contribuição do trabalho, um modelo para avaliação da dissipação de potência em NoCs. Diferentemente de outros modelos baseados no volume de dados transmitidos na NoC, este avalia a dissipação de potência em função das taxas de recepção nos *buffers* de cada porta de entrada dos roteadores da NoC. A consideração das taxas de recepção nos *buffers* permite capturar a flutuação das taxas de injeção devido a colisões entre os pacotes, e assim obter uma estimativa mais precisa que o modelo baseado apenas em volume.

A Seção 4.1 descreve o modelo, apresentando resultados sobre a NoC Hermes. A Seção 4.2 exibe uma comparação do modelo de taxas em relação a um modelo baseado no volume de dados transmitidos na rede. As Seções 4.3 e 4.4 exibem duas aplicações distintas do modelo de taxas proposto, uma no nível RTL de abstração e outra no nível TLM temporal (*timed*). No nível RTL, o modelo foi inserido no framework ATLAS [ATL07], capaz de gerar automaticamente NoCs e tráfegos para estas. Já no nível TLM temporal, o modelo proposto foi adicionado ao modelo JOSELITO [OST08].

### 4.1 Descrição do Modelo Proposto

Modelos de estimativa da dissipação de potência em NoCs como propostos em [PAL07], [CHA05], [SRI05] e [MAR05] utilizam o volume de tráfego transmitido pelos roteadores como principal métrica de estimativa. Esta abordagem é imprecisa, pois não considera os atrasos causados pelas colisões entre pacotes nos roteadores.

Como demonstrado em trabalhos anteriores [DAL03][PAL07][PAL05], a dissipação de potência de um roteador da NoC pode ser dividida em três componentes: (1) dissipação nos *buffers*; (2) dissipação na lógica de controle e roteamento; (3) dissipação no *crossbar* interno ao roteador. Como analisado em [PAL05], a maior dissipação de potência do roteador ocorre nos *buffers*, e análises posteriores, mostraram que estes dissipam cerca de 90% da potência total nos roteadores da NoC Hermes. Desta forma, a dissipação de potência geral do roteador está fortemente ligada à dissipação de cada um dos seus *buffers*. Caso a NoC possua uma arquitetura sem *buffers*, com roteamento do tipo “batata quente” (*hot-potato*) [IL05], as dissipações de potência do *crossbar* e lógica de controle e roteamento poderão ser uma função da taxa média de transmissão, assim como a taxa de transmissão afeta a dissipação de potência nos *buffers*.

A dissipação de potência em um *buffer* de entrada deve-se a dois componentes principais: o chaveamento no sinal de relógio e o chaveamento do valor armazenado. O sinal de relógio é de vital importância em um circuito síncrono. A transição periódica do sinal de relógio do *buffer* proporciona a dissipação de potência base do mesmo. Soma-se a esta a dissipação gerada pela troca do valor lógico que este *buffer* armazena. Sempre que houver uma mudança de pelo menos um *bit* no valor armazenado pelo *buffer*, ocorrerá uma maior dissipação do que a potência de base. Esta



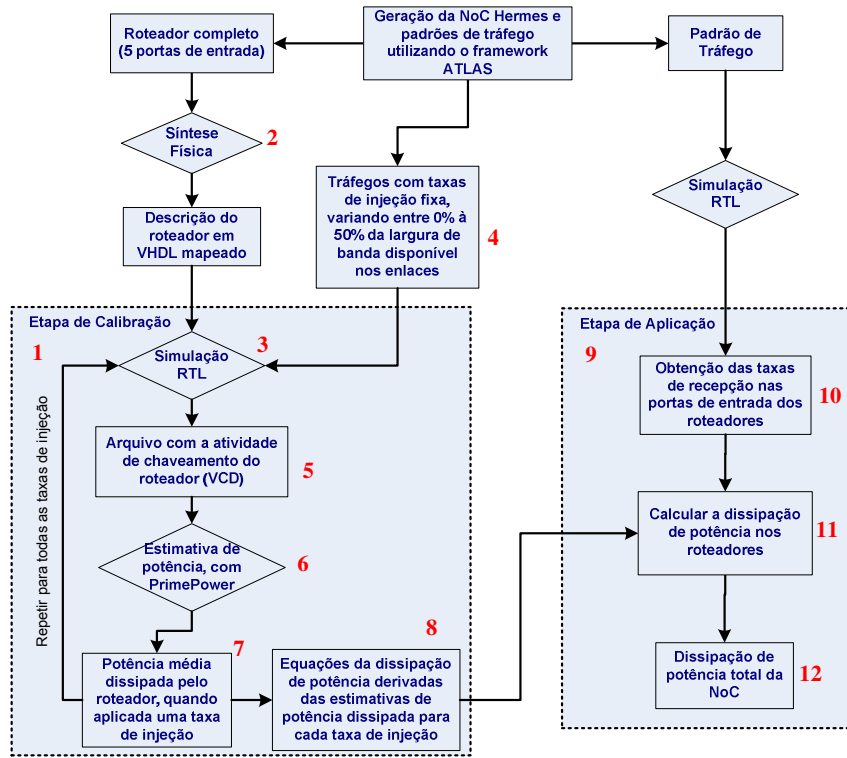
dissipação, adicionada à dissipação de base, irá aumentar na medida em que mais *bits* do valor armazenado troquem de valor lógico [PAL07]. A utilização da técnica de *clock-gating* [OH01] nos *buffers* pode reduzir significativamente a dissipação de potência dos roteadores.

A troca dos valores lógicos nos *buffers* está ligada à taxa de recepção destes. Quanto mais rápido o *buffer* receber valores, maior será a probabilidade de que algum *bit* armazenado mude de valor lógico, ocasionando assim uma dissipação de potência maior do que a de base. Desta forma, a taxa de recepção de um *buffer* irá refletir diretamente em alterações na dissipação de potência deste. O modelo aqui proposto para a estimativa da dissipação de potência em NoCs opera em duas etapas: calibração e aplicação.

Na primeira etapa do método proposto, a calibração (1 na Figura 16), ocorre à definição da parte do modelo que considera as características dos roteadores da NoC. Este procedimento consiste em sintetizar o roteador com o maior número de portas de entrada presente na NoC avaliada (2 na Figura 16). Utilizaremos a NoC Hermes como exemplo. Nesta NoC o roteador do tipo central é o roteador com o maior número de portas, 5 portas de entrada cada uma contendo um *buffer* de entrada. Este roteador foi sintetizado utilizando a ferramenta *Design Compiler* da Synosys [SYN08] aplicado a tecnologia XL035 (0,35  $\mu\text{m}$ ) da X-FAB [XFA08]. Ao final da síntese do roteador, obtêm-se dois arquivos principais: o *netlist* mapeado às células da tecnologia (arquivo “.db”, formato utilizado em ferramentas de posicionamento e roteamento) e um *netlist* VHDL para simulações.

A seguir deve-se simular a rede (3 na Figura 16) retirando o roteador central descrito puramente em VHDL, pelo *netlist* VHDL gerado anteriormente. Esta simulação deverá possuir um cenário de teste com tráfego de rede controlado, fazendo com que existam cinco fluxos não-concorrentes atravessando o roteador sintetizado a uma determinada taxa de transmissão (4 na Figura 16). Serão executadas sete simulações com este cenário de teste, cada uma com uma taxa de transmissão variando entre 0% e 50% da taxa máxima do enlace da NoC em análise. Para cada uma destas simulações é gerado um arquivo com as transições lógicas (VCD) dos sinais contidos no *netlist* VHDL do roteador avaliado (5 na Figura 16).

A seguir, avalia-se a dissipação de potência do roteador sob análise, utilizando a ferramenta *PrimePower* da Synosys [SYN07], (6 na Figura 16) obtendo-se um relatório de potência média para uma dada simulação. Esta ferramenta utiliza o *netlist* mapeado em células do roteador sintetizado e o arquivo VCD gerado nas simulações do *netlist* VHDL, para estimar a potência média dissipada no roteador e em cada um de seus sub-módulos. Para cada taxa de injeção simulada e avaliada obtêm-se uma potência média para os *buffers*, *crossbar* e lógica de controle e roteamento (7 na Figura 16). Esta etapa resulta em uma tabela com as potências médias, dada uma taxa de recepção no *buffer* avaliado. Aplica-se uma técnica de ajuste linear para obter uma equação que retorne a dissipação média de potência em função de uma taxa de recepção. Utiliza-se a mesma técnica para obter funções para o *crossbar* e a lógica de controle e roteamento do roteador (8 na Figura 16) quando aplicada uma dada taxa média de recepção nos *buffers* do roteador avaliado.



**Figura 16 - Fluxo de aplicação do modelo de estimativa da dissipação de potência de uma NoC.**

A etapa de calibração é executada apenas uma vez para a rede, e uma vez concluída pode-se estimar a dissipação de potência para diferentes cenários na etapa de aplicação do modelo. Notar que o método é genérico, pois utiliza informações da síntese para relacionar a dissipação de potência no roteador com suas taxas de recepção. O modelo então, pode ser aplicado a NoCs com diferentes características. Um tutorial sobre a calibração de uma NoC, descrevendo cada passo e ferramenta envolvida no processo, encontra-se no Apêndice II.

Na segunda fase do modelo proposto, aplicação do modelo (9 na Figura 16), deve-se calcular a quantidade de *flits* recebidos, em um *buffer*, em uma dada janela de tempo via simulação, transformando este valor em uma taxa de recepção (10 na Figura 16). A seguir, deverão ser aplicadas as equações geradas na etapa de calibração para transformar a taxa de recepção do *buffer* em um valor de potência média para uma dada janela de tempo. Este valor é denominado  $P_{buffer}$ , e está descrito na Equação 8 (11 na Figura 16), onde  $D_b$  é a dissipação de potência base do *buffer*,  $D_v$  é a *dissipação* de potência variável e  $tx$  é a taxa de recepção do *buffer*.

$$P_{buffer} = D_b + (tx \times D_v) \quad (8)$$

As potências médias do *crossbar* e lógica de controle e roteamento são obtidas calculando-se as médias das taxas de recepção em todos os *buffers*, e aplicando-se as equações geradas na fase de calibração. A potência média dissipada pelo roteador é dada pela Equação 9.

$$P_{avg} = \sum_{k=1}^m \frac{\sum_{i=1}^n P_{buffer_{k_i}}}{n} + P_{crossbar} + P_{control} \quad (9)$$

Onde:  $m$  corresponde ao número de *buffers* do roteador;  $n$  corresponde ao número de intervalos de amostragem das taxas de recepção utilizados durante toda a simulação. A seguir somam-se as dissipações de potência de todos os roteadores para se obter a estimativa da dissipação de potência de toda a NoC (12 na Figura 16).

Um dos fatores que contribuem para a precisão do modelo proposto é o intervalo de amostragem das taxas de recepção. Para intervalos muito pequenos o valor capturado seria a dissipação de potência instantânea em cada ciclo de relógio, e na prática, estaríamos operando como as ferramentas comerciais de estimativa de potência, consumindo muita memória e tempo de processamento. Para intervalos muito grandes, cresce o erro em relação à avaliação gerada por uma ferramenta comercial de estimativa de potência. Isto ocorre porque não seriam amostradas as potências médias intermediárias, as quais contribuem para uma melhor aproximação da estimativa potência média dissipada na rede.

O modelo proposto foi testado e avaliado sobre a NoC Hermes. Esta possui uma topologia do tipo malha 2D, chaveamento por pacotes do tipo *wormhole* e algoritmo de roteamento do tipo XY. A rede avaliada possui uma dimensão de 3x3, tamanho de *flit* igual a 8 *bits*, 8 *flits* de profundidade de *buffer* e controle de fluxo baseado em créditos. A avaliação do modelo sobre esta rede permitiu a visualização do perfil da dissipação de potência em seus roteadores, definindo as curvas de dissipação de potência em função das taxas de recepção nos *buffers*.

Na etapa de calibração foi estimada, através de uma ferramenta comercial, a dissipação média de potência em um *buffer*, no *crossbar* e na lógica de controle e roteamento do roteador completo (com 5 portas) (Figura 17). Estes gráficos exibem um crescimento linear da dissipação média de potência em função da taxa de recepção nos *buffers*.

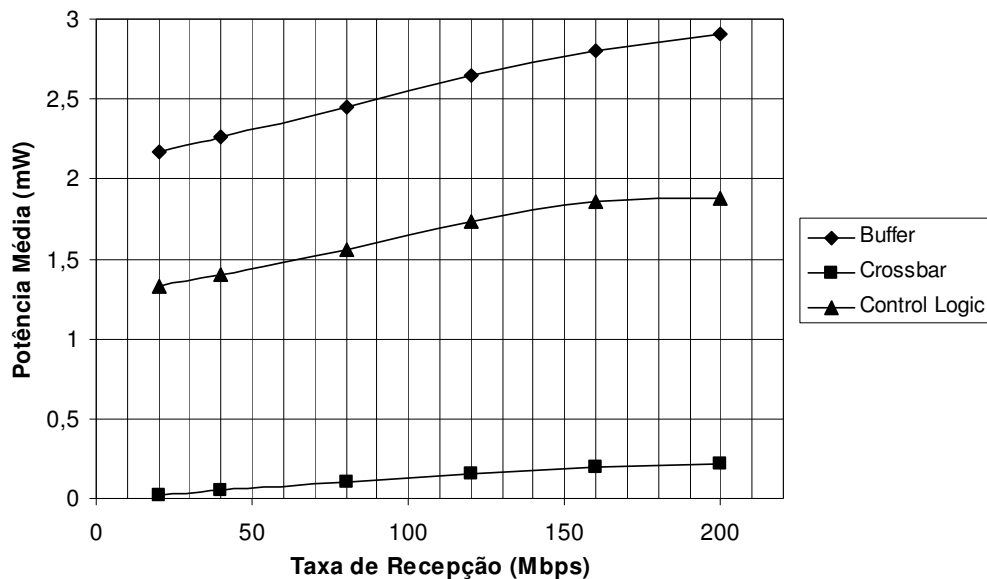


Figura 17 – Dissipação média de potência do *buffer*, *crossbar* e lógica de controle e roteamento, em função da taxa de recepção dos *buffers*.

A Tabela 4 mostra a dissipação de potência nos roteadores com 5 portas (ChaveCC), 4 portas (ChaveBC) e 2 portas (ChaveBL). A primeira linha da tabela mostra a dissipação estimada pelo modelo proposto, à segunda linha mostra a dissipação de potência avaliada pela ferramenta *PrimePower* e a terceira linha exibe o erro entre as duas avaliações. O erro entre a potência estimada pelo modelo e o avaliado pela ferramenta comercial será menor à medida que se retiram *buffers* de entrada da estrutura do roteador. Isto ocorre, pois se reduz o erro na avaliação da taxa média de recepção no roteador, a qual deve ser aplicada nas equações de dissipação de potência da lógica de controle e roteamento e no *crossbar*.

**Tabela 4 - Potência média estimada pelo *PrimePower* e no Modelo proposto nos diversos roteadores da NoC avaliada.**

Roteador	ChaveCC	ChaveBC	ChaveBL
<i>PrimePower</i>	12,483 mW	9,969 mW	7,828 mW
Modelo Taxas	13,2 mW	10 mW	7,84 mW
Erro	5,4%	0,31%	0,16%

## 4.2 Comparação com um Modelo Baseado em Volume de Dados

Modelos de estimativa da dissipação de potência, e/ou consumo de energia, em uma rede intra-chip baseado no volume de dados transmitidos são utilizados como opção a ferramentas comerciais de estimativa, ao se avaliar de forma rápida o impacto de mudanças estruturais ou lógicas em uma NoC. Estes modelos calculam quanto irá custar, em termos de potência/energia, a transmissão de um pacote desde sua origem até seu destino, passando por  $n$  roteadores intermediários. Este método permite uma boa precisão na estimativa, porém não leva em consideração possíveis problemas referentes ao tráfego se propagando na rede (e.g. congestionamento de rede).

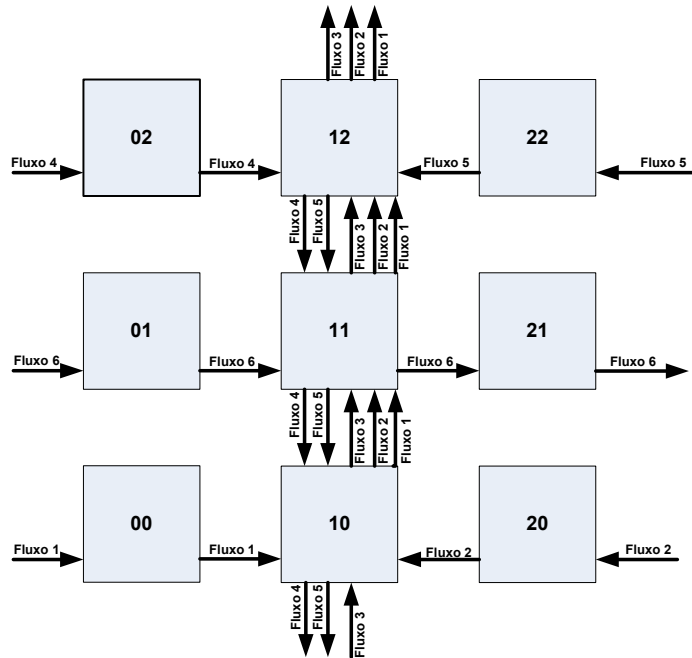
O modelo proposto, baseado em taxas, permite uma melhor estimativa da dissipação de potência de uma NoC, pois os problemas ocasionados pelo fluxo do tráfego na rede afetam as taxas de recepção nos *buffers* de cada roteador. Desta forma, o modelo baseado em taxas irá prover uma estimativa mais correta, quando comparada com o modelo baseado em volume de dados transmitidos. Para avaliar a correteza desta afirmação foram realizadas avaliações comparando os modelos citados, todos aplicados sobre a NoC Hermes.

Nestas avaliações a Equação 10 foi utilizada como modelo de estimativa baseada no volume de dados transmitidos na rede. Esta equação, definida no escopo do presente trabalho, calcula a potência dissipada pela rede utilizando como informação principal o caminho que os fluxos injetados na rede percorrem. Esta equação compreende o somatório de três parcelas (exibidas através de colchetes na Equação 10): a potência dissipada pelos fluxos injetados na rede, a potência dissipada nos *buffers* não utilizados dos roteadores que participam dos fluxos injetados, e a potência dissipada pelos roteadores que não participam de nenhum fluxo.

$$P_{NoC} = \left[ \sum_1^f \sum_1^{n+1} P_{buffer}(tx) + P_{crossbar} \left( \frac{tx}{p} \right) + P_{control} \left( \frac{tx}{p} \right) \right] + \left[ \sum_0^{B_f} P_{buffer}(0) \right] + \left[ \sum_0^{R_{inativos}} P_{buffer}(0) + P_{crossbar}(0) + P_{control}(0) \right] \quad (10)$$

Na equação acima,  $f$  representa a quantidade de fluxos injetados na rede;  $n$  é o número de saltos (*hops*) presentes em um fluxo;  $tx$  representa a taxa (em Mbps) utilizada em um fluxo injetado na rede;  $p$  é o número de portas presentes no roteador avaliado;  $B_f$  representa a quantidade de *buffers* não utilizados presentes nos roteadores que participam nos fluxos;  $R_{inativos}$  é a quantidade de roteadores que não participam em nenhum fluxo e  $P_{buffer}$ ,  $P_{crossbar}$ , e  $P_{control}$  são as potências dissipadas por um *buffer*, no *crossbar*, e na lógica de controle e roteamento a uma determinada taxa de transmissão, respectivamente.

O primeiro estudo de caso avalia a precisão de ambos os modelos, comparados com uma estimativa gerada através da ferramenta *PrimePower*, quando se aumenta a quantidade de fluxos injetados sobre rede (Figura 18). A NoC avaliada possui dimensão 3x3, com *flits* de 16 bits, 16 *flits* de profundidade de *buffer* e controle de fluxo baseado em créditos. Os tráfegos utilizados nestas avaliações possuem 500 pacotes de 16 *flits*, sendo transmitidos a 240 Mbps (30% da taxa máxima de injeção).



**Figura 18 – Fluxos de injeção utilizados nos estudos de caso, cada um possui 500 pacotes sendo transmitidos a 240 Mbps.**

Foram realizadas cinco avaliações, nas quais se aumenta gradativamente a quantidade de fluxos injetados na rede. O objetivo é avaliar o impacto do congestionamento nos modelos de estimativa de potência dissipada. Para apenas um fluxo injetado, é esperado que os modelos possuam a mesma precisão. Entretanto, na medida em que o número de colisões entre pacotes

aumente, ocorrerá flutuação na carga injetada, e por conseqüência o erro do modelo baseado em volume deverá crescer.

A Tabela 5 exibe o resultado desta avaliação. A primeira linha mostra a quantidade de fluxos injetados na rede, a segunda linha mostra a estimativa da potência média através da ferramenta *PrimePower*, a terceira linha exibe a estimativa da potência média utilizando o modelo de taxas proposto e a quarta linha mostra a potência média utilizando um modelo baseado em volume de dados transmitidos.

**Tabela 5 – Impacto do aumento de fluxos injetados na rede sobre a potência média dissipada por uma NoC. Estimativas obtidas utilizando-se diferentes modelos de estimativa.**

Fluxos:	1	1 e 2	1, 2 e 3	1, 2, 4 e 5	1, 2, 4, 5 e 6
<b>PrimePower:</b>	281,00 mW	282,00 mW	278,00 mW	280,00 mW	281,00 mW
<b>Modelo Taxas:</b>	287,94 mW	292,43 mW	296,93 mW	301,43 mW	304,78 mW
<b>Modelo Volume:</b>	287,99 mW	312,78 mW	336,41 mW	345,85 mW	350,49 mW

A Tabela 6 exibe as avaliações de erro entre o modelo de estimativa de potência proposto baseado em taxas e o modelo baseado em volume de dados transmitidos em comparação a ferramenta *PrimePower*. A última linha da tabela se refere à diferença existente entre os dois modelos. A tabela mostra que para tráfegos com poucos fluxos injetados na rede, o modelo de estimativa baseado em volume de dados apresenta um erro de até 10% em comparação com a estimativa de uma ferramenta comercial. Porém para três ou mais fluxos o erro deste modelo aumenta. Isto ocorre devido à colisão entre os pacotes na rede. Já o modelo de estimativa baseado em taxas se mantém abaixo dos 10% de erro, mesmo com mais fluxos sendo aplicados na rede. Isto ocorre, pois a inserção de mais fluxos de transmissão na rede afeta as taxas de recepção dos *buffers* de cada um dos roteadores, modificando o valor da estimativa de potência dissipada na NoC.

**Tabela 6 – Avaliações de erro entre as estimativas realizadas nos modelos avaliados e a ferramenta PrimePower. A última linha é a diferença entre os dois modelos.**

Fluxos:	1	1 e 2	1, 2 e 3	1, 2, 4 e 5	1, 2, 4, 5 e 6
<b>Erro Taxas:</b>	2,47%	3,70%	6,81%	7,65%	8,47%
<b>Erro Volume:</b>	2,49%	10,42%	21,01%	23,52%	24,73%
<b>Diferença:</b>	0,02%	6,96%	13,30%	17,54%	15,00%

A Tabela 7 exibe os tempos de execução de cada fluxo para a obtenção da estimativa da potência dissipada na NoC. Note que o modelo baseado em volume de dados obteve tempos de execução *instantâneos*, isto ocorre, pois o modelo somente avalia a potência baseado nos arquivos de tráfego disponíveis. O modelo baseado em taxas obteve tempos de execução menores que uma avaliação utilizando ferramentas comerciais.

**Tabela 7 – Avaliações de tempo de execução dos modelos avaliados em comparação com a ferramenta *PrimePower*. Na tabela os tempos estão em minutos, e a sigla “*Inst.*” demonstra que o tempo foi instantâneo.**

<b>Fluxos:</b>	1	1 e 2	1, 2 e 3	1, 2, 4 e 5	1, 2, 4, 5 e 6
<b><i>PrimePower</i>:</b>	15 min.	15 min.	25 min.	15 min.	15 min.
<b>Modelo Taxas:</b>	2 min.	2 min.	2 min.	2 min.	2 min.
<b>Modelo Volume:</b>	<i>Inst.</i>	<i>Inst.</i>	<i>Inst.</i>	<i>Inst.</i>	<i>Inst.</i>

O próximo estudo de caso avalia um cenário com um maior número de iniciadores de tráfego. Nesta simulação todos os roteadores da rede transmitem pacotes para os demais roteadores, congestionando a rede. Foram simulados dois cenários de tráfegos: (i) 1000 pacotes de 16 *flits* por iniciador de tráfego, sendo transmitidos a 120 Mbps (15% da taxa máxima de injeção); (ii) 5000 pacotes de 16 *flits* por iniciador de tráfego, sendo transmitidos a 400 Mbps (50% da taxa máxima de injeção). A mesma NoC do estudo de caso anterior foi utilizada. As Tabela 8 e Tabela 9 exibem os valores das estimativas e a avaliação de erro entre elas, respectivamente.

**Tabela 8 – Estimativas da dissipação de potência para cenários de tráfego mais intenso.**

<b>Tráfego:</b>	1000 pacotes a 120 Mbps	5000 pacotes a 120 Mbps
<b><i>PrimePower</i>:</b>	283,00 mW	288,00 mW
<b>Modelo Taxas:</b>	299,30 mW	299,91 mW
<b>Modelo Volume:</b>	405,49 mW	442,60 mW

**Tabela 9 – Avaliação de erro entre os diferentes modelos quando aplicado cenários de tráfego mais intenso.**

<b>Tráfego:</b>	1000 pacotes a 120 Mbps	5000 pacotes a 120 Mbps
<b>Erro Taxas:</b>	5,76%	4,14%
<b>Erro Volume:</b>	43,28%	53,68%
<b>Diferença:</b>	35,48%	47,57%

Os resultados apresentados mostram que em simulações com tráfego mais intenso, o modelo baseado em taxas ainda mantém um erro inferior a 10% em relação ao *PrimePower*, enquanto que no modelo baseado em volume de dados este erro ultrapassa de 50% em relação à ferramenta comercial de estimativa.

A Tabela 10 exhibe os tempos de execução para os testes acima, note que mesmo para tráfegos com muitos pacotes, o modelo baseado em volume gera resultados *instantaneamente*. Por outro lado, o modelo baseado em taxas reduziu significativamente o tempo de obtenção de uma estimativa, obtendo uma alta precisão nas estimativas.

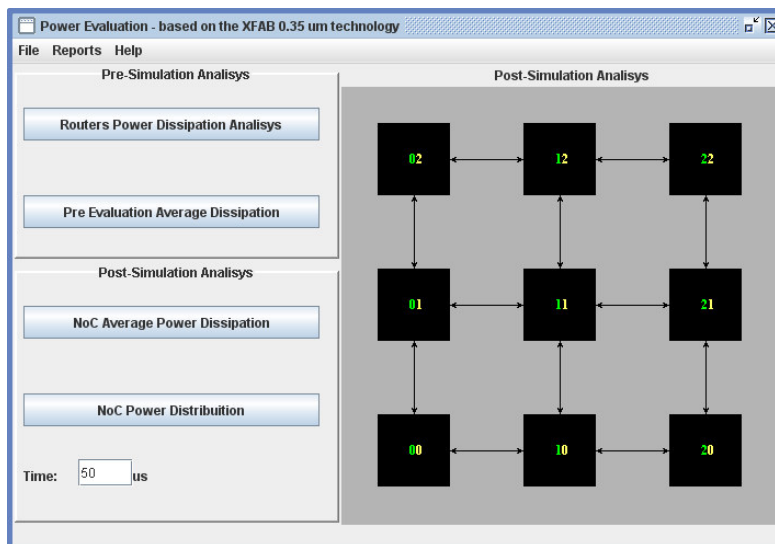
**Tabela 10 - Avaliações de tempo de execução dos modelos avaliados em comparação com a ferramenta *PrimePower* para tráfegos com alta transmissão na rede. Na tabela os tempos estão em minutos, e a sigla “*Inst.*” demonstra que o tempo foi instantâneo.**

Tráfego:	1000 pacotes a 120 Mbps	5000 pacotes a 120 Mbps
<i>PrimePower:</i>	45 min.	140 min.
Modelo Taxas:	10 min.	13 min.
Modelo Volume:	<i>Inst.</i>	<i>Inst.</i>

Apesar de modelo baseado em volume gerar uma estimativa de consumo muito rápida, o projetista deve ter em mente que o mapeamento das tarefas, e os conseqüentes fluxos de dados, podem gerar variações muito grandes na potência em função das colisões dos pacotes. Por esta razão, o modelo baseado em taxas permite uma estimativa precisa, em um nível de abstração que ainda permite a obtenção de resultados em um tempo aceitável.

### 4.3 HEFESTUS – Integração do Modelo Proposto ao Framework ATLAS

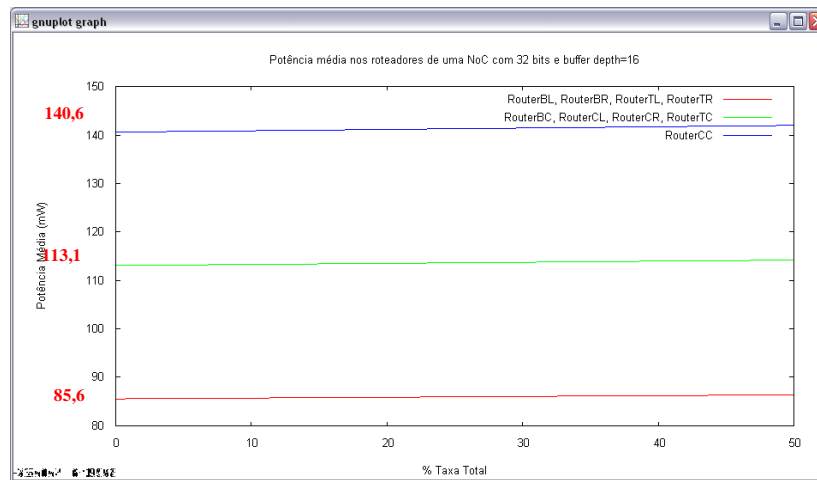
O módulo HEFESTUS, integrado ao framework ATLAS, automatiza a etapa de aplicação do modelo proposto de estimativa da dissipação de potência em NoCs. Este módulo contém uma série de programas, transparentes ao usuário, que coletam, processam e exibem a dissipação de potência de uma NoC Hermes quando aplicado um padrão de tráfego. A Figura 19 mostra a janela principal do módulo HEFESTUS. O HEFESTUS possui quatro botões principais: análise da dissipação de potência nos roteadores (*Routers Power Dissipation Analysis*), estimativa da dissipação de potência na rede baseada em um modelo de volume de dados transmitidos (*Pre Evaluation Average Dissipation*), estimativa da dissipação de potência na rede baseada em um modelo de taxas (*NoC Average Power Dissipation*), e análise da distribuição de potência sobre a rede (*NoC Power Distribution*). Na Figura 19, o campo “*time*” refere-se à janela de tempo para análise da distribuição de potência.



**Figura 19 - Janela principal do módulo de estimativa de potência HEFESTUS.**

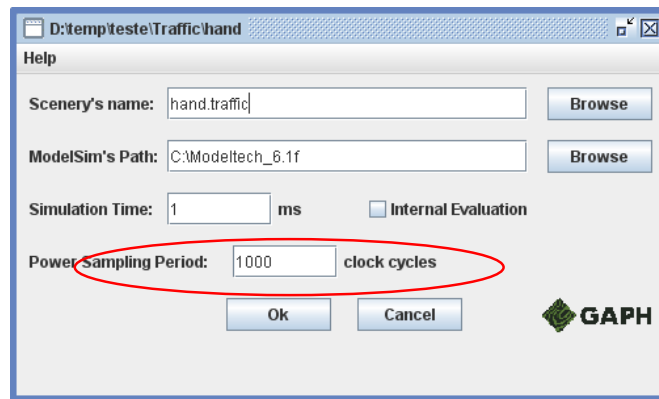


O módulo HEFESTUS pode realizar estimativas de dissipação de potência antes e após a simulação da rede gerada no framework ATLAS. Antes da simulação, no espaço denominado de “*Pre-Simulation Analysis*” na Figura 19, o usuário pode visualizar os perfis de dissipação de potência dos roteadores da rede clicando no botão “*Routers Power Dissipation Analysis*” (Figura 19). Será exibido um gráfico da dissipação de potência de cada um dos tipos de roteadores presente na NoC gerada, como mostra a Figura 20. Observar nesta figura que a dissipação de potência nos diferentes roteadores é praticamente constante para esta rede. Devido à alta atividade de chaveamento do sinal de relógio nos *buffers*, a dissipação de potência torna-se praticamente independente da taxa de recepção dos dados. Este fato aponta para potenciais vantagens de utilização de interfaces assíncronas [PON08] e técnicas de *clock-gating*. Ainda no espaço “*Pré-Simulation Analysis*”, o usuário poderá realizar uma rápida estimativa da dissipação de potência da rede quando submetida ao tráfego gerado na ATLAS, clicando no botão “*Pre Evaluation Average Dissipation*” (Figura 19). Esta estimativa utiliza o modelo baseado no volume de dados transmitidos na rede, descrito na Seção 4.2.



**Figura 20 - Perfil da dissipação de potência dos diversos roteadores; gráfico da potência em função da taxa de recepção nos buffers.**

As estimativas de potência realizadas após a simulação da rede estão presentes no espaço “*Post-Simulation Analysis*” na Figura 19, e são realizadas aplicando o modelo proposto de dissipação de potência baseado em taxas. A etapa de calibração do modelo às diferentes NoCs Hermes (geradas no framework ATLAS) foi executada previamente, e todas as equações de potência foram armazenadas no módulo HEFESTUS. Para automatizar o processo de aplicação do modelo, modificou-se o ambiente de simulação de uma NoC no framework ATLAS. Foi inserida uma segunda simulação, esta possuindo monitores instanciados em todas as portas de entrada dos roteadores da NoC. Estes monitores repassam a quantidade de *flits* recebidos, em uma dada janela de tempo (configurada diretamente na janela de simulação da ferramenta ATLAS, e exibida na Figura 21), à ferramenta *Modelsim* e esta armazena estes dados em uma lista.

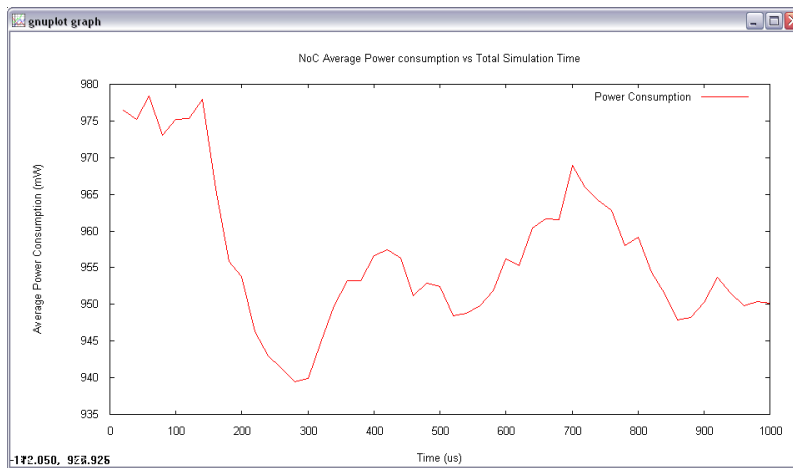


**Figura 21 – Janela do ambiente de simulação de NoCs no framework ATLAS. Em destaque o campo que configura a janela de amostragem para a avaliação da dissipação de potência.**

O módulo HEFESTUS, então, realiza a estimativa de potência em uma NoC Hermes, seguindo os seguintes passos:

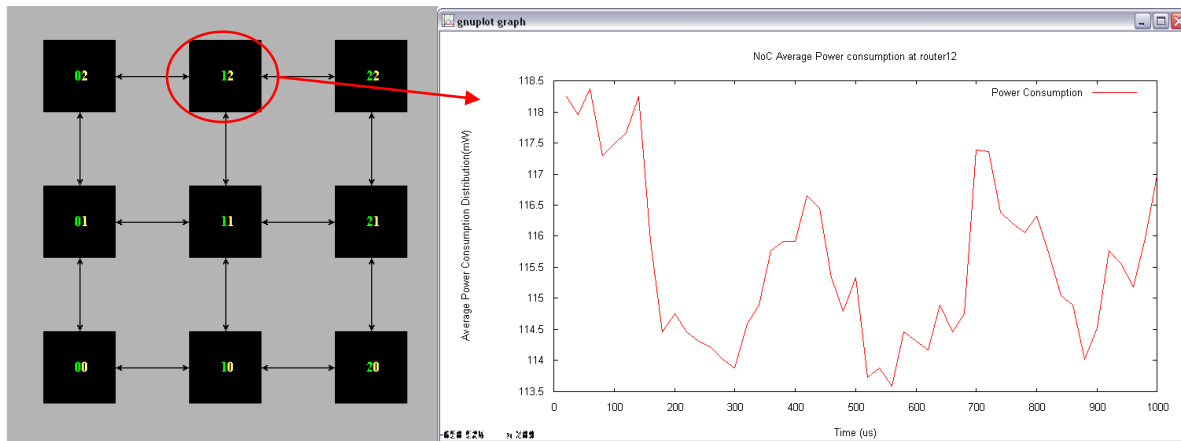
1. O módulo HEFESTUS processa a lista de *flits* recebidos gerada pela ferramenta *Modelsim*, calculando a taxa de recepção em cada uma das portas de cada roteador presente na NoC simulada;
2. As equações de potência, armazenadas no módulo HEFESTUS e específicas para cada NoC Hermes, são aplicadas e a potência média dissipada (em cada janela de tempo) por cada um dos roteadores é extraída e armazenada em arquivos (um para cada roteador);
3. É extraída a potência média dissipada pela NoC inteira (em cada janela de tempo) com base nos arquivos dos roteadores. Estes valores são armazenados em um arquivo;
4. Os resultados são apresentados na forma de gráficos, utilizando a ferramenta Gnuplot [GNU08].

Este processo é realizado uma vez ao abrir a janela principal do HEFESTUS, caso haja um registro de uma simulação da rede aplicada a um padrão de tráfego no framework ATLAS. Se outra simulação for executada (mudanças no tempo de simulação ou padrão de tráfego), a avaliação será executada novamente ao abrir a janela principal do HEFESTUS. Ao final da análise pode-se visualizar o gráfico da dissipação de potência média da NoC durante todo o tempo de simulação, clicando no botão “*NoC Average Power Dissipation*” (Figura 19). A Figura 22 mostra este gráfico. Note que o valor inferior da dissipação de potência para esta NoC é de aproximadamente 940 mW, o qual corresponde aproximadamente aos valores de  $4 \cdot 85,6$  (roteadores com 3 portas operando sem qualquer tráfego, valor obtido na Figura 20) +  $4 \cdot 113,1$  (roteadores com 4 portas operando sem qualquer tráfego, Figura 20) + 140,6 (roteador com 5 portas operando sem qualquer tráfego, Figura 20).



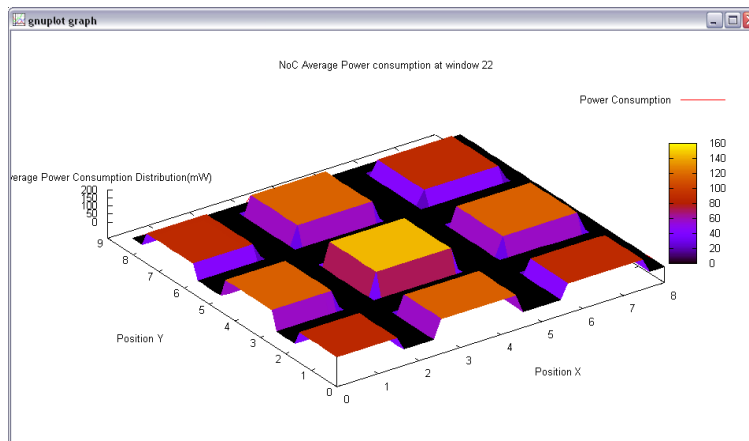
**Figura 22 - Dissipação média de potência de uma NoC durante todo o tempo de simulação; gráfico da dissipação de potência em função do tempo.**

Ao lado direito da janela principal do módulo HEFESTUS é apresentada a NoC graficamente. Ao se clicar em algum dos roteadores presentes nesta representação gráfica, o usuário obtêm o gráfico da dissipação de potência média deste roteador durante todo o tempo de simulação. A Figura 23 exhibe este gráfico, que somente é obtido após a simulação da NoC no framework ATLAS. Estes gráficos podem ser utilizados para direcionar, por exemplo, decisões de mapeamento de tarefas ou roteamento adaptativo na rede.



**Figura 23 - Dissipação média de potência de um roteador da NoC durante todo o tempo de simulação; gráfico da dissipação de potência em função do tempo.**

Um gráfico 3D com a distribuição da dissipação de potência nos diferentes roteadores da NoC pode ser visualizado inserindo-se o período tempo desejado no campo “time”, e clicando no botão “NoC Power Distribution” (Figura 19). A Figura 24 exhibe este gráfico para um tempo de 440 us. Este tipo de análise é fundamental para avaliar como a dissipação de potência está distribuída na rede, ilustrando-se graficamente os pontos de maior dissipação de potência quando aplicado um determinado tráfego.



**Figura 24 - Distribuição da dissipação de potência nos diversos roteadores da NoC; gráfico 3D com a distribuição instantânea em um dado tempo de simulação.**

Além da apresentação gráfica dos resultados, o módulo HEFESTUS permite a visualização dos resultados na forma de relatórios ou tabelas de valores (Figura 25). Estes relatórios podem ser exportados para outras ferramentas (e.g. Microsoft Excel) para cálculo da energia consumida pela rede ou a energia consumida na transmissão de um *flit*.

noc.dat		AveragePowerConsumption.txt	
20.0	976.4645	NOC:	976.031877
40.0	975.28864	Router00:	89.4228999
60.0	978.3889	Router10:	117.992462
80.0	973.0578	Router20:	89.5928894
100.0	975.29504	Router01:	117.924689
120.0	975.3615	Router11:	146.353558
140.0	978.00214	Router21:	117.889204
160.0	965.7412	Router02:	89.4802847
180.0	955.8442	Router12:	117.929157
200.0	953.7859	Router22:	89.4667240
220.0	946.22424		
240.0	942.9323		
260.0	941.19464		
280.0	939.41455		
300.0	939.88434		
320.0	944.8244		
340.0	949.79816		
360.0	953.21423		
380.0	953.1991		
400.0	956.6509		
420.0	957.41345		
440.0	956.34		

**Figura 25 - Resultados obtidos no módulo HEFESTUS exibidos na forma de um relatório.**

O módulo HEFESTUS somente se aplica às NoCs Hermes geradas através do framework ATLAS, pois somente estas NoCs foram calibradas. Porém, a inclusão dos demais tipos de NoCs (geradas através da ATLAS) pode ser facilmente realizada, bastando calibrar a nova NoC e inserir suas equações de dissipação de potência no módulo HEFESTUS.

#### 4.4 Integração do modelo proposto com o modelo JOSELITO

JOSELITO é um modelo orientado a atores, implementado no ambiente *PtolemyII* [PTO08], baseado na infra-estrutura da NoC HERMES. A orientação a atores foi, inicialmente, proposta como um modelo matemático de computação concorrente [HEW77] e, posteriormente, como um modelo de concorrência. Os principais componentes dessa abordagem são atores, que são elementos concorrentes que se comunicam entre si através da troca de fichas (do inglês, *tokens*) por meio de canais (do inglês, *channel*) [LIU04][LEE03]. As fichas transmitidas entre os atores podem

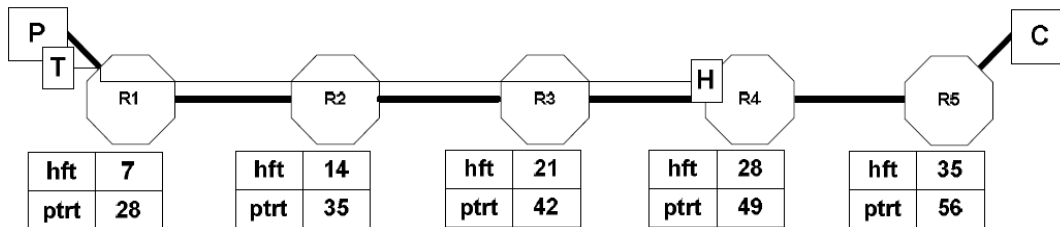
encapsular um valor simples (e.g. tipo inteiro) ou um conjunto de valores de diferentes tipos identificados por um nome (denominados de registro de *tokens*) [LEE03].

JOSELITO é baseado na técnica de abstração de pacotes (do inglês, *Packet Abstraction Technique* ou PAT) [OST08]. A principal idéia do PAT [OST08] é diminuir o tempo total de simulação através da redução no número de eventos de comunicação, que ocorre devido à abstração da carga útil dos pacotes transmitidos. A PAT consiste em: (i) abstração dos pacotes, contendo apenas cabeçalho e terminador; (ii) abstração dos *buffers*, modelados como uma estrutura de FIFO; (iii) um método analítico para estimar o tempo de liberação do terminador de um pacote. Estas abstrações eliminam a transferência, *flit-a-flit*, da carga útil dos pacotes. Os pacotes são encaminhados na NoC de acordo com o tempo de liberação do terminador do pacote (do inglês, *packet trailer release time* ou *ptrt*), conforme definido na Equação 11,

$$ptrt = hft + pcksize * ctf \quad (11)$$

onde: *ptrt* é o tempo de liberação do terminador do pacote, *hft* é o tempo de encaminhamento do cabeçalho do pacote (do inglês, *header forwarding time*), *pcksize* é o tamanho em *flits* do pacote, e *ctf* é o número de ciclos de relógio para transmitir um *flit* (do inglês, *clock cycles to transmit one flit*).

Para melhor exemplificar a técnica, considere um pacote de 21 *flits* sendo transmitido entre os roteadores R1 e R5 de uma NoC Hermes (Figura 26), sem congestionamento na rede, o *ctf* desta rede é igual a 1 (pois possui controle de fluxo baseado em créditos), e seu *hft* é igual a 7 (que é a quantidade de ciclos de relógio necessária para executar a arbitragem e roteamento). Na Figura 26, o *hft* aumenta em 7 ciclos de relógio a cada salto (*hop*), e o *ptrt* é atualizado de acordo com a Equação 11. O último valor de *ptrt*, 56, corresponde à latência do pacote, e este valor é o mesmo que a implementação RTL da NoC.



**Figura 26 - Tempos de liberação para um cenário de entrega de pacotes sem bloqueios na rede.**

O PAT permite a simulação de cenários de transmissão de pacotes com ou sem bloqueios na rede. Se não houver conflito de recursos da rede (cenário sem bloqueios), os valores de latência e vazão da NoC podem ser avaliados sem perdas de precisão. Em um cenário com bloqueios, quando o cabeçalho de um pacote chega ao *buffer* de entrada, duas situações de bloqueio poderão ocorrer: (i) a porta de saída destino está reservada para outra porta de entrada ou (ii) a porta de entrada de um roteador vizinho não está apta a receber o cabeçalho ou o terminador deste pacote. Em ambos os casos o cabeçalho fica bloqueado, e o terminador continua a seguir seu caminho de acordo com a profundidade do *buffer*.

O JOSELITO possui dois tipos de atores principais: (i) *buffers* de entrada para receber os

dados do pacote, e (ii) um árbitro, responsável pelo roteamento dos dados do pacote recebido através dos canais ativos. O JOSELITO foi avaliado sob diferentes condições de tráfego, e os resultados mostram que o pior caso de vazão no JOSELITO possui 0.1% de erro quando comparado com o nível RTL [OST08]. A integração do modelo proposto de estimativa da potência média dissipada em uma NoC, ao JOSELITO necessitou três mudanças principais no ambiente original:

1. Um contador foi inserido em cada *buffer* de cada roteador, e estes calculam o número de pacotes recebidos dentro de um período de tempo denominado de *recPkts*. Devido à ausência de transmissão da carga útil, este contador é incrementado quando um terminador de pacote é recebido.
2. Cada *buffer* calcula sua taxa média de recepção (*avbrr*), de acordo com a Equação 12, onde: *pktSize*: é o tamanho do pacote em *flits*; *phit*: é o tamanho, em *bits*, do canal físico de transmissão de dados entre dois roteadores; *T*: é o período de relógio; e *sw* é a janela de amostragem, em ciclos de relógio.

$$avbrr = \frac{recPkts \times pktSize \times phit}{T \times sw} \quad (12)$$

3. Um monitor recolhe a taxa média de recepção de cada *buffer* durante a simulação. Ao final de cada período de amostragem, o valor de *avbrr* é enviado ao *PowerScope*.

O *PowerScope* é um ator parametrizável, ilustrado ao lado direito da Figura 27. Este ator exibe em tempo de execução a potência média dissipada em cada *buffer* de toda a NoC, aumentando a observabilidade do sistema. Esta característica pode ajudar os projetistas na detecção de pontos de alta dissipação de potência, permitindo, por exemplo, a avaliação de diferentes mapeamentos de aplicações na rede visando à baixa dissipação de potência.

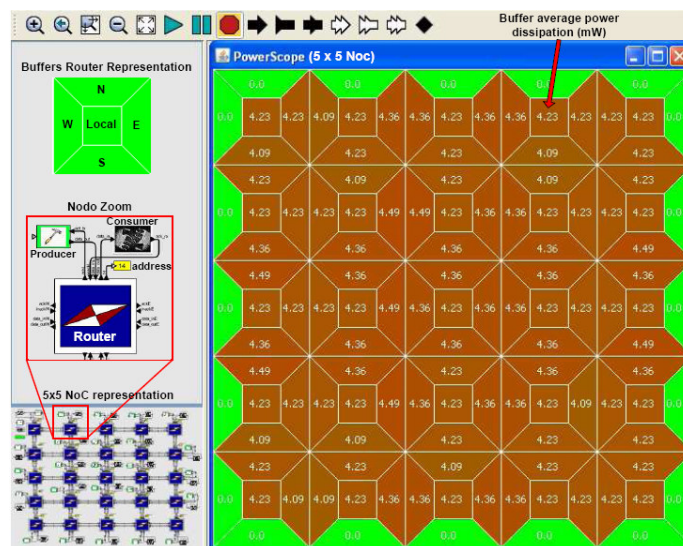


Figura 27 – Representação visual do *PowerScope*.

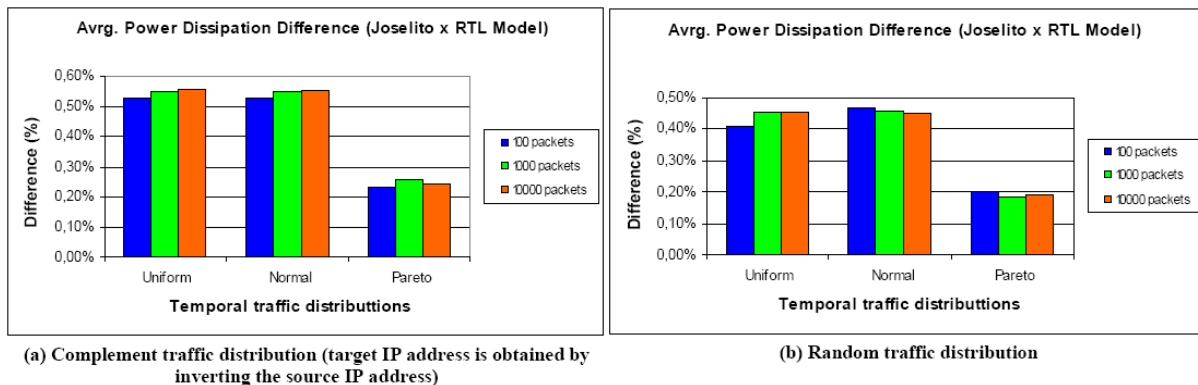
O *PowerScope* gera um relatório da dissipação de potência, com os valores máximo, mínimo e médio de potência dissipada por roteador. O *PowerScope* necessita dos seguintes parâmetros de potência do usuário: (i) dissipação base da lógica de controle e roteamento; (ii)

dissipação variável da lógica de controle e roteamento; (iii) dissipação base do *buffer* de entrada; (iv) dissipação variável do *buffer* de entrada. Estes parâmetros são adquiridos na etapa de calibração do modelo proposto. Note que as dissipações de potência do *crossbar* não estão citadas nos parâmetros acima, pois na etapa de calibração da NoC Hermes com controle de fluxo do tipo *handshake* (utilizada nesta versão do JOSELITO), as dissipações de potência do *crossbar* foram computadas juntamente com a lógica de controle e roteamento.

Foi realizada uma série de testes utilizando o modelo sobre uma simulação RTL (utilizando o módulo HEFESTUS no framework ATLAS) e uma simulação baseada em atores (JOSELITO no PtolemyII). A realização destes testes ocorreu sobre uma NoC Hermes 4x4 com topologia malha 2D, 16 *bits* de tamanho de *flit*, 8 *flits* de profundidade de *buffer*, e algoritmo de roteamento XY. A taxa máxima nos enlaces desta rede é de 800 Mbps. Foram utilizados tráfegos com pacotes de tamanho fixo, de 32 *flits*, e os demais parâmetros variam entre:

- Distribuição temporal do tráfego: uniforme (a 200 Mbps), normal (com taxa mínima de 150 Mbps, máxima de 250 Mbps e desvio padrão de 10 Mbps), e *Pareto on/off* (a 200 Mbps com número máximo de rajadas igual a 10);
- Distribuição espacial do tráfego: complemento e aleatório.
- Quantidade de pacotes transmitidos: 100, 1000 e 10000.

A Figura 28 apresenta a diferença da estimativa da potência dissipada entre as simulações realizadas no PtolemyII (JOSELITO) e no HEFESTUS. Note que os cenários de tráfego induzem congestionamento na rede, o que implica em pacotes bloqueados. Mesmo com taxas de injeção próximas ao ponto de saturação da rede (redes com topologia malha saturam quando aplicadas taxas de injeção entre 20% e 30% do máximo permitido em seus enlaces), e admitindo colisões entre pacotes, a simulação baseada em atores mostrou resultados similares aos da simulação RTL. A diferença de pior caso (0,55%) é observada quando um número pequeno de pacotes é inserido na rede. Quando o número de pacotes inserido na rede aumenta, a diferença entre as duas simulações diminui.



**Figura 28 - Diferença na estimativa da dissipação média de potência entre as simulações baseada em atores (JOSELITO) e RTL (HEFESTUS).**

## 4.5 Conclusão

Ferramentas comerciais de estimativa da potência dissipada em circuitos integrados permitem uma avaliação precisa da potência dissipada, porém demandam alto poder de recursos computacionais e tempo de avaliação. O modelo de estimativa da dissipação em NoCs baseados no volume de dados transmitidos na rede permite uma rápida estimativa da potência dissipada, porém são imprecisos em comparação com as ferramentas comerciais. O modelo proposto baseado em taxas de recepção nos *buffers* de entrada dos roteadores permite uma avaliação mais precisa que o modelo baseado em volume de dados transmitidos, e é mais rápido que ferramentas comerciais.

Estimativas de dissipação de potência podem ser geradas em diversos níveis de abstração, do RTL ao TLM temporal. O módulo de estimativa da dissipação de potência desenvolvido, HEFESTUS, tem o objetivo de suprir esta necessidade no framework ATLAS, provendo condições para que o projetista avalie de forma rápida uma NoC gerada pelo framework ATLAS. O modelo JOSELITO permite gerar estimativas de potência de forma mais rápida, pois este modelo utiliza a técnica PAT para acelerar as simulações.





## 5 DISSIPAÇÃO DE POTÊNCIA NA NOC HERMES

Este Capítulo descreve a segunda contribuição do trabalho, os resultados de avaliações da dissipação de potência na NoC Hermes utilizando o módulo HEFESTUS. Foram realizadas avaliações de potência sobre a NoC Hermes de referência (sem a utilização de qualquer técnica de redução da dissipação de potência), com a utilização de *clock-gating* em todo o roteador, automaticamente via ferramenta de síntese e com *clock-gating* manual, inserido apenas nos *buffers* de entrada. Além destas, realizou-se um estudo sobre a influência da atividade de chaveamento dos dados nos pacotes transmitidos na potência média dissipada pela NoC, e uma avaliação da dissipação de potência dos diversos algoritmos de roteamento adaptativos disponíveis na NoC Hermes.

As Seções 5.1, 5.2 e 5.3 apresentam o estudo da dissipação de potência na NoC Hermes de referência, com a aplicação de *clock-gating* automático e *clock-gating* manual, respectivamente. A Seção 5.4 exibe um estudo sobre a influência da atividade de chaveamento dos dados nos pacotes transmitidos, e a Seção 5.5 descreve o impacto dos algoritmos adaptativos na dissipação de potência da NoC Hermes.

### 5.1 Dissipação de Potência na NoC Hermes de Referência

Na rede Hermes de referência, a dissipação de potência dos *buffers* é dominante devido à atividade de chaveamento do sinal de relógio. A Figura 29 exibe a avaliação do percentual de contribuição dos *buffers* na dissipação de potência total dos roteadores. Estes dados foram extraídos de uma NoC Hermes com controle de fluxo baseado em créditos, e variando-se o tamanho de *flit* e profundidade dos *buffers*.

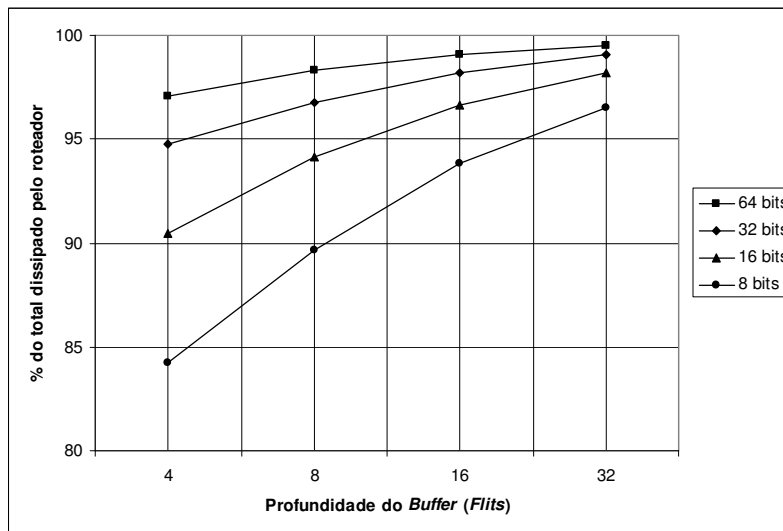
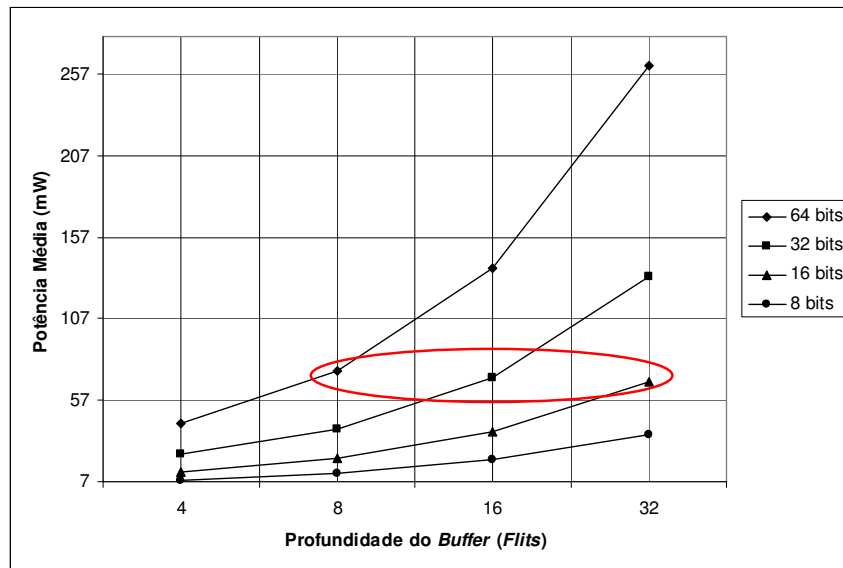


Figura 29 - Dissipação de potência nos *buffers* em relação à dissipação total do roteador.

A Figura 30 exibe a dissipação de potência dos roteadores da NoC Hermes de referência, variando-se o tamanho de *flit*, de 8 a 64 bits, e profundidade dos *buffers* de entrada, de 4 a 32 flits.

Note que nesta figura existem roteadores que possuem diferentes tamanhos de *flit* e profundidade de *buffer*, porém possuem dissipação de potência similar. Isto ocorre, pois os roteadores podem ser divididos em regiões que possuem o mesmo dimensionamento total de *buffer*, este dimensionamento pode ser expresso pela multiplicação da profundidade de *buffer* pelo tamanho de *flit*. Por exemplo, um roteador com *flits* de 16 bits e 8 posições no *buffer*, possui um dimensionamento total igual a 128 *bits*.



**Figura 30 – Dissipação média de potência nos roteadores da NoC Hermes de referência, variando-se o tamanho de *flit* e a profundidade dos *buffers* de entrada.**

O gráfico da Figura 30 pode ser condensado em uma tabela (Tabela 11), exibindo em valores a divisão da potência dissipada nos roteadores em regiões. A Tabela 11 mostra a dissipação de potência para uma NoC com controle de fluxo baseada em créditos, para cada um dos possíveis tamanhos de *buffer*. Esta tabela mostra que, roteadores de diferentes tamanhos de *flit* e profundidade de *buffer*, os quais possuem diferentes características como vazão e latência, podem dissipar a mesma potência média. Por exemplo, uma NoC Hermes com controle de fluxo baseado em créditos, com *flits* de 16 *bits* e 16 posições no *buffer* (tamanho do *buffer* igual a 256 *bits*), irá dissipar a mesma potência que uma NoC Hermes com controle de fluxo baseado em créditos, com *flits* de 64 *bits* e 4 posições no *buffer*.

Esta avaliação permite explorar NoCs de diferentes características, porém com a mesma dissipação de potência. Tomando o exemplo acima citado, para um projeto que necessita de uma alta vazão na rede utiliza-se a primeira NoC citada (16 *bits* de tamanho de *flit* e 16 posições no *buffer*), porém se o projeto utiliza o enlace de forma aperiódica e transmitindo grandes quantidades de dados, pode-se utilizar à segunda NoC citada (64 *bits* de tamanho de *flit* e 4 posições no *buffer*). Estas avaliações têm como único compromisso a dissipação média de potência pela rede, outros compromissos como área e desempenho não foram considerados nestas avaliações.

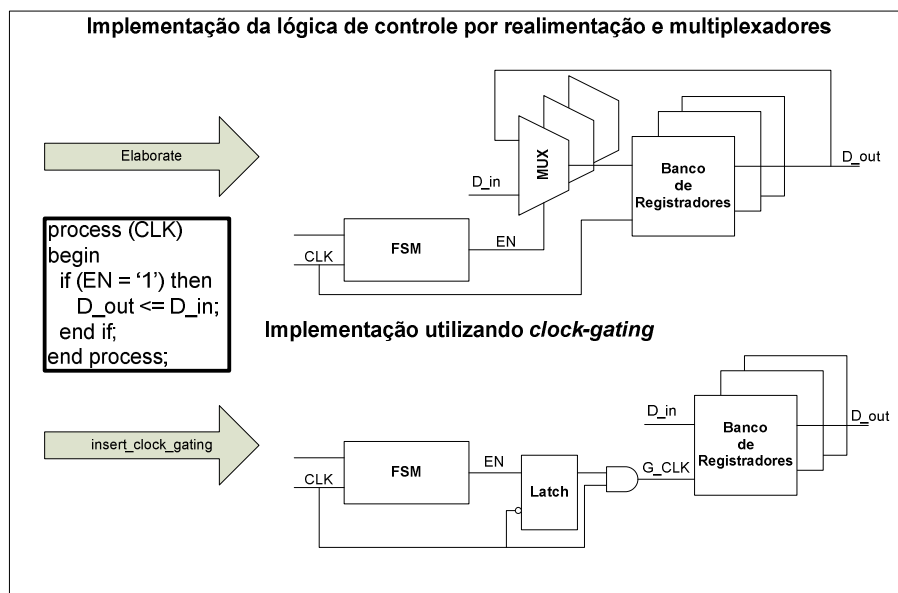
**Tabela 11 - Dissipação de potência nos roteadores centrais, a uma taxa de recepção de 10% da taxa máxima no enlace. As diagonais da tabela mostram regiões que possuem o mesmo tamanho de *buffer* (em *bits*), em consequência estas regiões possuem a mesma dissipação de potência.**

Tam. Flit / Prof. Buffer	4	8	16	32
8	7,92	12,08	20,20	35,96
16	13,20	21,45	37,50	68,62
32	24,10	39,24	70,97	133,21
64	43,04	74,85	137,98	262,22

## 5.2 Dissipação de Potência na NoC Hermes *Clock-Gated* Automático

O modelo de estimativa de potência demonstrou que a maior parcela da potência dissipada deve-se aos *buffers*, justificando o estudo de técnicas para reduzir esta dissipação de potência. Dentre as técnicas utilizadas para redução de potência, destaca-se o *clock-gating* [OH01][GAR99]. A técnica de *clock-gating* está largamente difundida nas ferramentas comerciais de síntese, podendo ser aplicada automaticamente sobre o projeto.

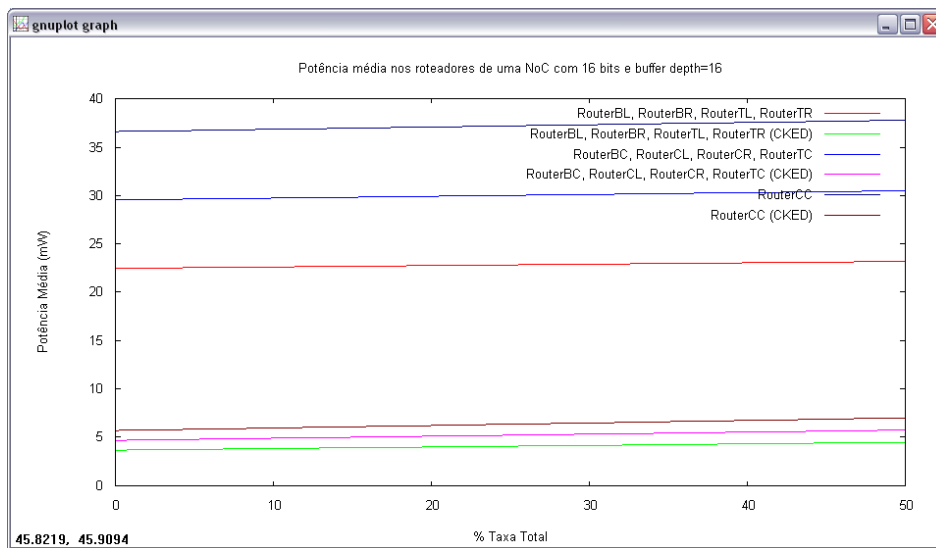
No fluxo de projeto da *Synopsys*, utilizado no escopo deste trabalho, a técnica de *clock-gating* pode ser aplicada automaticamente através da diretiva “*insert\_clock\_gating*” na ferramenta *Design Compiler*. A ferramenta de síntese, então, insere uma alternativa à lógica de controle tradicional dos registradores. A síntese passa de uma implementação utilizando lógica de controle (sinal de *enable*) por realimentação e multiplexadores, para uma implementação na qual o sinal de controle dos registradores (*enable*) é embutido no sinal de relógio (Figura 31). Desta forma quando o sinal de *enable* estiver desativado, o sinal de relógio não é repassado adiante e o banco de registradores é desativado.



**Figura 31 – Diferença entre um circuito sintetizado sem a aplicação de *clock-gating*, e com a aplicação de *clock-gating* automático, no ambiente de síntese *Design Compiler* da *Synopsys*.**

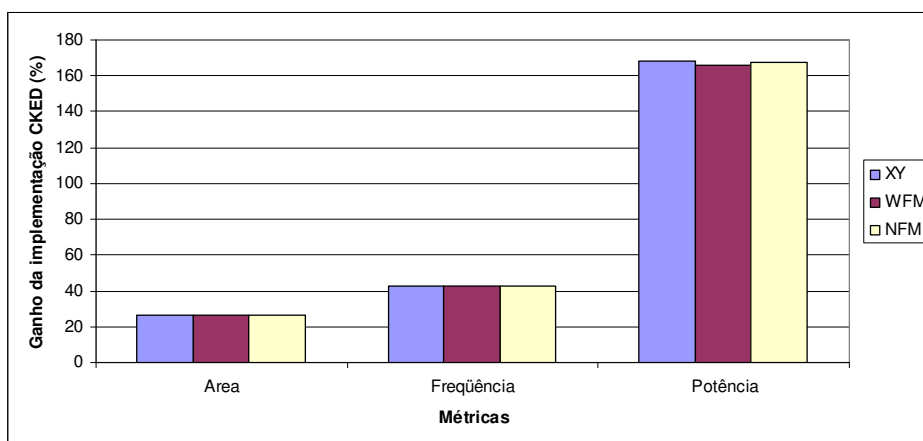
Esta técnica permite a redução da dissipação de potência na árvore de distribuição de relógio, nos registradores e na lógica combinacional (multiplexadores). Além da redução na dissipação de potência, o *clock-gating* pode reduzir a área do projeto (reduzindo a lógica combinacional, retirando-se multiplexadores) e aumentar a frequência máxima de operação [SYN08]. Espera-se que com a utilização de *clock-gating*, via diretiva na ferramenta de síntese, a dissipação de potência da NoC Hermes seja reduzida. Para este experimento foi utilizada uma NoC Hermes 3x3, com 16 *bits* de tamanho de *flit*, *buffer* com profundidade de 16 posições, e com controle de fluxo variando entre *handshake*, baseado em créditos sem canais virtuais, baseado em créditos com dois e quatro canais virtuais.

Executou-se novamente a rotina de calibração do modelo para esta nova NoC, e foram obtidas as equações de dissipação de potência para os roteadores. Observou-se uma redução significativa na dissipação de potência dos roteadores, e por consequência, na dissipação da NoC. A Figura 32 exhibe graficamente esta redução para cada um dos roteadores presentes em uma NoC Hermes com controle de fluxo baseado em créditos sem canais virtuais. A potência dissipada nos roteadores centrais (com cinco portas) caiu de 37,8 mW para 7,04 mW, a uma taxa de recepção nos *buffers* de 50% do total de banda disponível. Na Figura 32 os roteadores que utilizam a técnica de *clock-gating* estão descritos como CKED.



**Figura 32 - Perfil da dissipação de potência dos roteadores da NoC, com e sem a técnica de *clock-gating* (CKED).**

Além da redução na potência dissipada nos roteadores da NoC, observou-se uma redução na área total dos roteadores e um aumento na sua frequência de operação. Foram realizados testes utilizando a NoC citada acima, variando-se os algoritmos de roteamento entre *XY*, *west-first* mínimo (*WF*) e *negative-first* mínimo (*NF*). A Figura 33 mostra graficamente a redução, em porcentagem, na dissipação de potência total, área consumida e o aumento da frequência máxima de operação na NoC com e sem a aplicação de *clock-gating*.



**Figura 33 – Gráfico que apresenta a diferença nos valores de área, frequência e dissipação de potência em uma NoC com e sem a aplicação de *clock-gating* (CKED).**

A seguir fixou-se uma NoC 3x3, com 8 *bits* de tamanho de *flit* e 8 posições no *buffer*, variando-se o controle de fluxo entre *handshake* e baseado em créditos. Mediu-se a dissipação de potência para estas NoCs com e sem utilização de *clock-gating*. Aplicou-se o mesmo tráfego em ambas as medições, a uma frequência de 50 MHz e um tempo de simulação de 1 ms. A dissipação de potência da NoC, baseada em créditos, por exemplo, caiu de 246 mW médios para 53 mW médios quando aplicada a técnica de *clock-gating*. A Tabela 12 exibe a dissipação média de potência para o teste acima. Nota-se a redução da dissipação de potência quando utilizada a técnica de *clock-gating*.

**Tabela 12 - Comparação da potência dissipada por NoCs com controle de fluxo do tipo *handshake* (HAN) e baseado em créditos (CRE), com e sem o uso de *clock-gating* (CKED).**

NoC	Potência Média (mW)	Energia Média (mJ)
HAN	243,58	12,42
CRE	246,22	12,55
HAN-CKED	25,42	1,29
CRE-CKED	53,16	2,71

A seguir, utilizando a mesma NoC do exemplo anterior, mediu-se a contribuição dos canais virtuais na dissipação de potência. Todas as simulações possuem o mesmo tráfego, mesma frequência e tempo de simulação igual a 1 ms. Pode-se notar na Tabela 13 que a diferença na dissipação de potência entre as NoCs sem canais virtuais e com dois canais virtuais é menor que o dobro, porém esta diferença entre as NoCs com dois e quatro canais virtuais é praticamente o dobro.

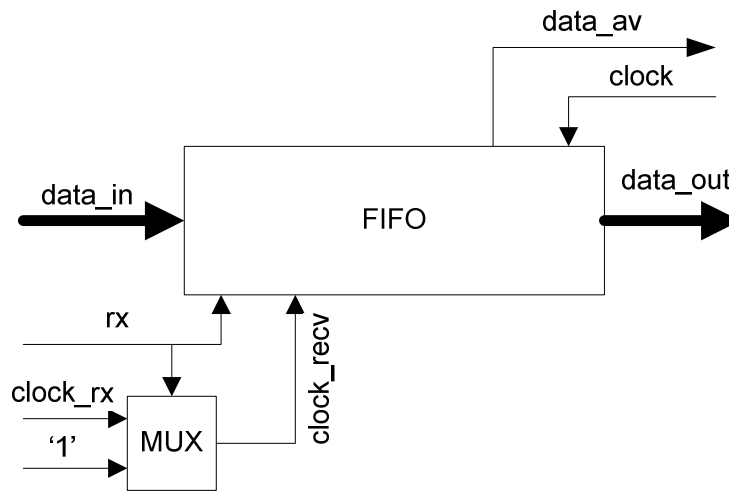
**Tabela 13 - Comparação da potência consumida por NoCs, com controle de fluxo baseado em créditos (CRE), variando-se a quantidade de canais virtuais (CV). Em todos os casos foi utilizada a técnica de *clock-gating* (CKED).**

NoC	Potência Média (mW)	Energia Média (mJ)
CRE 1CV CKED	53,16	2,71
CRE 2CV CKED	84,97	4,33
CRE 4CV CKED	167,23	8,52

### 5.3 Dissipação de Potência na NoC Hermes *Clock-Gated Manual*

Os experimentos da Seção anterior confirmaram a importância da utilização de técnicas de *clock-gating* no projeto de NoCs de baixa dissipação de potência. Porém estes experimentos utilizaram a inserção automática de *clock-gating* pela ferramenta de síntese. Se esta opção não estiver disponível para o projetista, como em ferramentas de síntese para FPGAs, alternativa como a inserção de *clock-gating* manual deve ser avaliada.

O principal objetivo do *clock-gating* manual é a aplicação da técnica de *clock-gating* apenas nos *buffers* de entrada, através de uma alteração no código VHDL destes. Na NoC Hermes, com controle de fluxo baseado em crédito, esta técnica pode ser aplicada modificando-se o código VHDL dos *buffers* de entrada para somente repassar o sinal de relógio aos *buffers* de entrada quando houverem dados a serem transmitidos para estes *buffers*. Desta forma, o sinal de relógio enviado aos *buffers* de entrada (“*clock\_rx*” na Figura 34) estará sobre o controle do sinal de transmissão (“*rx*” na Figura 34) através da utilização de um multiplexador.



**Figura 34 – Estrutura do *buffer* da NoC Hermes modificada, inserindo-se a técnica de *clock-gating* manual.**

Foram realizados testes com uma NoC Hermes 3x3, com 8 *bits* de tamanho de *flit*, profundidade de *buffer* de 8 posições, controle de fluxo baseado em créditos e roteamento do tipo XY. A NoC com *clock-gating* manual foi submetida à calibração do modelo à tecnologia XL035 da X-FAB, resultando em equações de dissipação de potência para os seus roteadores. Foram simulados três tráfegos com distribuições temporais diferentes: Uniforme, Normal e *Pareto On/Off*, cada um com 1000 pacotes de 16 *flits* transmitidos a uma taxa de 240 Mbps. A Tabela 14 mostra os resultados desta avaliação, comparados com avaliações com *clock-gating* automático e sem *clock-gating* (Hermes de referência).

**Tabela 14 – Dissipação de potência para uma NoC sem *clock-gating*, com *clock-gating* automático e com *clock-gating* manual.**

NoC / Tráfego	Uniforme	Normal	Pareto On/Off
Referência	275,51 mW	275,62 mW	259,19 mW
<i>Clock-gated</i> automática	73,57 mW	73,69 mW	55,28 mW
<i>Clock-gated</i> manual	185,58 mW	186,13 mW	105,33 mW

A partir dos dados da Tabela 14 pode-se notar uma redução de 48% na potência dissipada pela NoC com *clock-gating* manual, nos tráfegos Uniforme e Normal, e 146% no tráfego Pareto On/Off. Diferentemente do *clock-gating* automático, a aplicação do *clock-gating* manual aumenta em 3,33% a área total do circuito, referente à lógica de controle (multiplexadores) adicionada aos *buffers* de entrada dos roteadores. O *clock-gating* manual mantém a mesma frequência máxima de operação da rede Hermes de referência, 195,31 MHz. Estes resultados, porém, foram obtidos apenas para NoC mapeadas na tecnologia XL035 da XFAB (ASIC), e futuros estudos sobre a eficiência desta técnica na redução da dissipação de potência em NoCs prototipadas em FPGAs deverão ser realizados.

## 5.4 Influência da Atividade de Chaveamento dos Dados

A atividade de chaveamento do circuito é um dos fatores principais no cálculo da dissipação de potência dinâmica deste. A atividade de chaveamento em uma NoC, ocorre em ambos os enlaces de transmissão dos dados (*links*), quanto na lógica que compõe os roteadores. O modelo de estimativa da dissipação de potência em NoCs baseado em taxas desenvolvido neste trabalho, não avalia a dissipação de potência nos enlaces que conectam os roteadores. A adição da avaliação da dissipação de potência nos enlaces será realizada futuramente, aumentando a precisão do modelo baseado em taxas.

Um enlace é um conjunto de fios que interconecta dois roteadores vizinhos, e a avaliação da potência dissipada nos enlaces está diretamente associada à atividade de chaveamento nos fios que os compõem [PAL07]. A atividade de chaveamento em um enlace é a variação dos *bits* de um *flit* do pacote transmitido para outro, a porcentagem da atividade de chaveamento no enlace aumenta na medida em que mais *bits* variam de um *flit* para outro. A atividade de chaveamento nos roteadores é composta pela atividade de chaveamento em cada um dos elementos que os compõem (*buffers* de entrada, lógica de controle e roteamento e *crossbar*). Dentre estes elementos formadores do roteador, destacam-se os *buffers* de entrada, responsáveis por cerca de 90% da dissipação total dos roteadores, desta forma a influência da atividade de chaveamento nos *buffers* de entrada deverá afetar diretamente a dissipação de potência do roteador. É necessário, porém, salientar que a atividade de chaveamento nos enlaces pode (ou não) afetar a atividade de chaveamento nos *buffers* de entrada, desta forma existe a necessidade de se avaliar independentemente a influência da atividade de chaveamento em cada estrutura.

Primeiramente, será avaliado o impacto da atividade de chaveamento nos enlaces da NoC Hermes. A Tabela 15 mostra a atividade de chaveamento dos pacotes gerados pelo framework

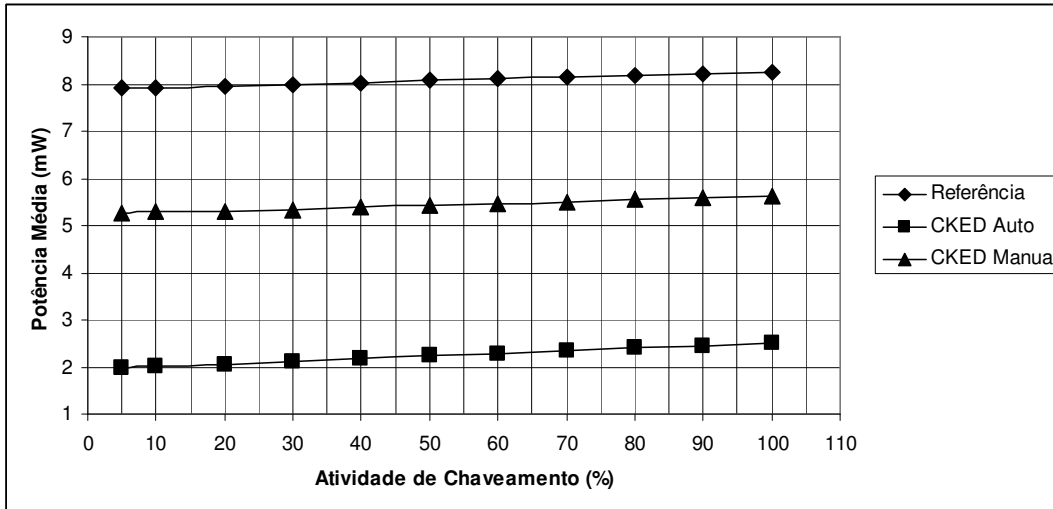


ATLAS nos enlaces. Note que na medida em que se aumenta o tamanho do pacote (em *flits*) a atividade de chaveamento diminui. Isto ocorre, pois os primeiros *flits* possuem variação elevada de conteúdo, enquanto que os demais *flits* apenas poucos *bits* mudam de estado (contador de seqüência em cada flit).

**Tabela 15 – Atividade de chaveamento nos pacotes gerados através do framework ATLAS, em função do tamanho do pacote em *flits*.**

Tamanho do Pacote	Atividade de Chaveamento
15 <i>flits</i>	22,2 %
25 <i>flits</i>	15,8 %
100 <i>flits</i>	13,55 %
1000 <i>flits</i>	12,64 %

A seguir, foi realizada uma avaliação da atividade de chaveamento nos enlaces na NoC Hermes, utilizando pacotes controlados que possuem de 5% a 100% de variação na atividade de chaveamento. Esta avaliação foi realizada sobre uma NoC Hermes 3x3, com 16 *bits* de tamanho de *flit*, 16 *flits* de profundidade de *buffer* e controle de fluxo baseado em créditos. O tráfego possui pacotes com tamanho de 16 *flits*, transmitidos a 200 Mbps (50% da taxa máxima de transmissão nos enlaces). Foram avaliadas além da rede Hermes de referência, as redes que utilizam roteadores com a aplicação das técnicas de *clock-gating* automática (CKED Auto) e manual (CKED Manual). A Figura 35 exibe de forma gráfica os resultados desta avaliação.

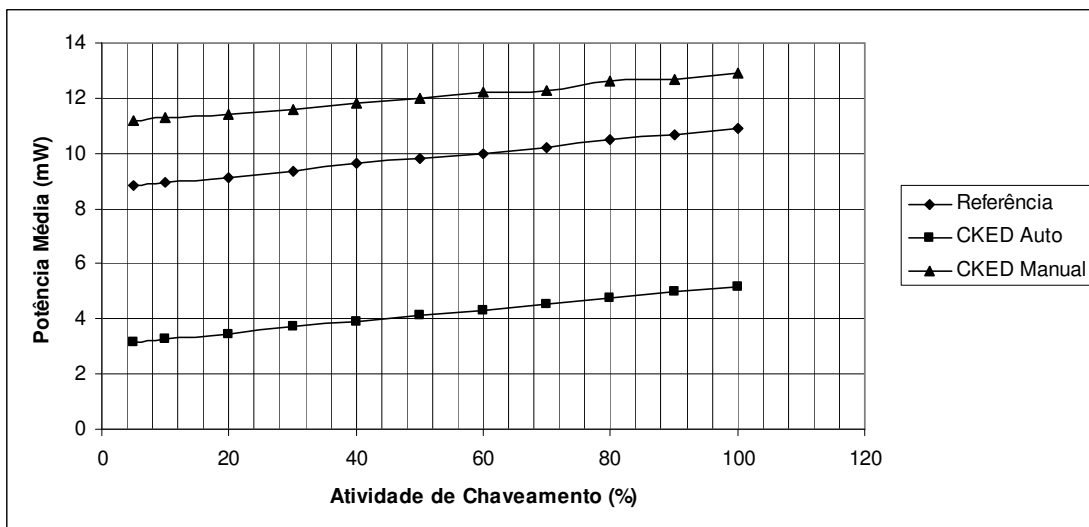


**Figura 35 – Impacto da atividade de chaveamento nos *flits* transmitidos na dissipação média de potência para as NoCs Hermes de referência, com a utilização de *clock-gating* automático (CKED Auto) e utilizando *clock-gating* manual (CKED Manual).**

Nesta avaliação observou-se uma variação de 4,42% (0,35 mW) na potência dissipada pela NoC Hermes de referência, 26,13% (0,52 mW) na rede que utiliza *clock-gating* automático e 7,33% (0,36 mW) na rede que utiliza *clock-gating* manual. Nesta avaliação ocorreu pouco impacto da atividade de chaveamento na potência dissipada nas redes avaliadas. Isto ocorre, pois nestas simulações a relação entre o tamanho dos pacotes e o tamanho dos *buffers* de entrada proporcionou

uma baixa atividade de chaveamento nos *buffers*, mesmo que exista uma alta atividade de chaveamento nos enlaces. Estas simulações foram específicas para avaliar o impacto da atividade de chaveamento nos enlaces, e não nos *buffers* de entrada.

O objetivo da segunda avaliação é observar o impacto da atividade de chaveamento nos *flits* dos pacotes transmitidos na dissipação de potência dos *buffers* de entrada dos roteadores. A mesma rede da avaliação anterior foi utilizada, e foram desenvolvidos tráfegos controlados que forcem a escrita em uma mesma posição dos *buffers* de entrada, variando os *bits* destas posições. Novamente foram utilizados pacotes com tamanho de 16 *flits*, porém foi utilizada uma taxa de transmissão de 800 Mbps (100% da taxa máxima permitida nos enlaces). Esta taxa é necessária, pois para obter uma atividade de chaveamento de até 100% dos bits das posições do *buffer*, um único pacote controlado é enviado. Este pacote controlado permite variar a atividade de chaveamento nos *buffers* entre 5% e 100%. Foram realizadas avaliações com a rede Hermes de referência, e as redes que utilizam roteadores com a aplicação das técnicas de *clock-gating* automática (CKED Auto) e manual (CKED Manual). A Figura 36 exibe graficamente os resultados desta avaliação.



**Figura 36 – Impacto da atividade de chaveamento nos *buffers* de entrada na dissipação média de potência para as NoCs Hermes de referência, com a utilização de *clock-gating* automático (CKED Auto) e utilizando *clock-gating* manual (CKED Manual).**

Nesta avaliação observou-se uma variação de 23,58% (2,08 mW) na potência dissipada pela NoC Hermes de referência, 64,44% (2,03 mW) na rede que utiliza *clock-gating* automático e 15,18% (1,7 mW) na rede que utiliza *clock-gating* manual. Note que a potência dissipada na NoC com a aplicação de *clock-gating* manual é maior que a NoC de referência. Isto ocorre, pois quando se aplica a taxa máxima de injeção da rede (taxa apenas teórica, porém necessária na realização desta avaliação) elimina-se a atuação da técnica de *clock-gating*, pois os *buffers* sempre estarão recebendo dados (desta forma operando com seu sinal de relógio normalmente). A aplicação do *clock-gating* manual aumenta a área do circuito (como visto na Seção 5.3), desta forma a dissipação de potência da NoC com *clock-gating* manual será maior que a NoC de referência. Isto não ocorre com a aplicação de *clock-gating* automático, pois existe uma redução na área do circuito em

comparação com a NoC de referência (como descrito na Seção 5.2).

Observa-se nos resultados desta avaliação um alto impacto da atividade de chaveamento nos *buffer* de entrada na dissipação de potência dos roteadores, principalmente na NoC com a aplicação da técnica de *clock-gating* automático. Esta avaliação mostra que as equações do modelo proposto deverão ser ajustadas para considerar ambas as variações de dissipação de potência devido à atividade de chaveamento nos enlaces e nos *buffers* de entrada dos roteadores. Um exemplo de tal ajuste seria a modificação da equação da dissipação de potência nos *buffers* (Equação 8, no Capítulo 4), para um novo formato (Equação 13). Onde  $\beta$  é o acréscimo de potência devido à atividade de chaveamento nos *buffers*,  $D_b$  é a dissipação de potência base do *buffer*,  $D_v$  é a dissipação de potência variável e  $tx$  é a taxa de recepção do *buffer*.

$$P_{buffer} = \beta \times D_b + (tx \times D_v) \quad (13)$$

## 5.5 Dissipação de Potência dos Diferentes Algoritmos de Roteamento

Esta Seção realiza um estudo sobre a dissipação de potência de diferentes algoritmos de roteamento adaptativos aplicados na rede Hermes. A exploração de diferentes algoritmos de roteamento permite ao projetista avaliar um melhor compromisso entre grandezas como vazão e latência e a dissipação de potência da rede. Nesta Seção foram abordadas apenas avaliações de dissipação média de potência dos algoritmos adaptativos, e não se objetivou avaliar grandezas como vazão e latência média.

As avaliações foram realizadas sobre uma NoC com dimensão de 3x3, *flits* de 16 *bits*, 16 *flits* de profundidade do *buffer* e controle de fluxo baseado em créditos. Foram avaliados os algoritmos adaptativos **West-First** (WF), **North-Last** (NL) e **Negative-First** (NF), em suas versões **mínimas** (M) e **não-mínimas** (NM), todos comparados com o algoritmo **XY** (que é o algoritmo de roteamento padrão na rede Hermes). Foram avaliadas duas distribuições espaciais do tipo: **aleatória** (*Random*) e **complemento** (*Complement*), sobre cada distribuição espacial foram utilizados três distribuições temporais: (i) **uniforme** a uma taxa de 200 Mbps (25 % da taxa máxima permitida nos enlaces), (ii) **normal** com taxa mínima de 150 Mbps, máxima de 250 Mbps e média de 200 Mbps, e (iii) **Pareto On/Off** com taxa de 200 Mbps e rajadas de 10 *flits*. Sobre cada distribuição temporal foram simulados tráfegos que transmitiam **100**, **1000** e **10000** pacotes com tamanho de 16 *flits*.

A Tabela 16 exibe os resultados de todas as avaliações para simulações com 1000 pacotes de 16 *flits* por roteador sendo transmitidos na rede. Os melhores resultados de cada simulação estão destacados em negrito. Avaliando-se a tabela abaixo, verificou-se que ocorre pouca variação na potência média dissipada entre os algoritmos de roteamento. A maior diferença é uma redução de 2,24% (19,64 mW) observada na simulação presente na sétima linha da Tabela 16.

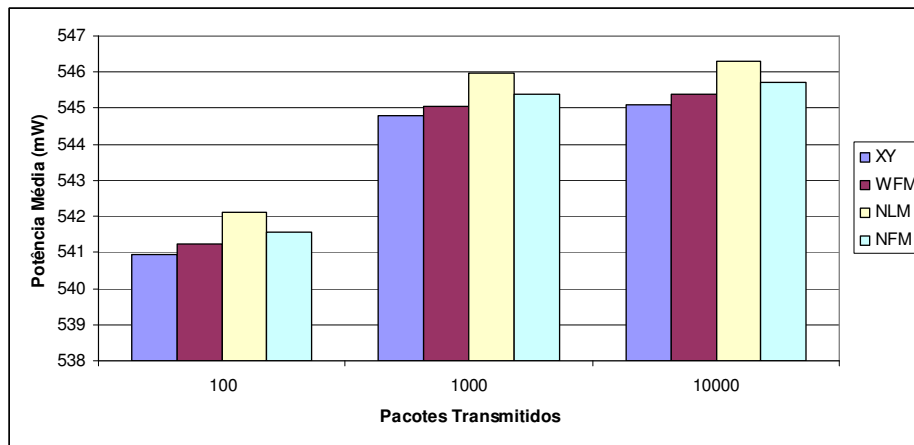
A análise da Tabela 16 mostra que para padrões de tráfego que apresentam distribuição espacial do tipo aleatória (onde existe pouca colisão entre pacotes transmitidos), o algoritmo XY é a melhor opção dentre todos os algoritmos avaliados. Dentre os algoritmos adaptativos, a melhor opção de escolha para a NoC Hermes é o algoritmo **West-First** (WF) mínimo. Já para padrões de

tráfego que apresentam distribuição espacial do tipo complemento (onde existe um alto índice de colisão dos pacotes transmitidos), o algoritmo *Negative-First* (NF) mínimo é melhor que todos os demais algoritmos de roteamento, a exceção de tráfegos com distribuição temporal do tipo *Pareto On/Off*. Nestes tráfegos algoritmo XY apresenta melhores resultados em comparação com os demais algoritmos de roteamento.

A Figura 37 exibe graficamente os resultados dos algoritmos adaptativos mínimos para as simulações com tráfego que apresentam distribuição espacial do tipo aleatória e distribuição temporal normal, para 100, 1000 e 10000 pacotes com 16 *flits* sendo transmitidos na rede. Note neste gráfico que a dissipação de potência de todos os algoritmos de roteamento é similar, variando em no máximo 1,2 mW em relação à dissipação de potência de melhor caso (algoritmo XY) para o pior caso (algoritmo *North-Last*).

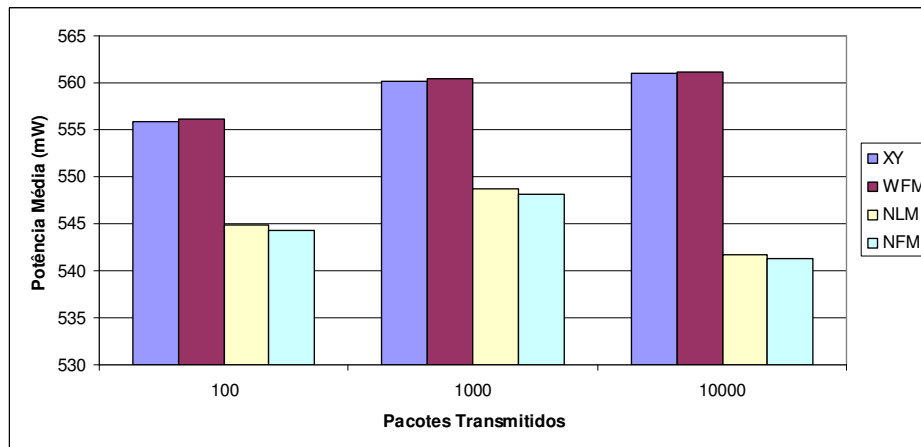
**Tabela 16 – Resultados das avaliações dos algoritmos adaptativos e XY para as simulações com 1000 pacotes de 16 *flits*. Os melhores resultados dentre os algoritmos avaliados estão destacados em negrito.**

Dist. Espacial	Dist. Temporal	Minimalidade	XY	West-First	North-Last	Negative-First
Aleatória	Uniforme	Mínimo	<b>544,77 mW</b>	545,04 mW	545,96 mW	545,36 mW
Aleatória	Normal	Mínimo	<b>544,78 mW</b>	545,06 mW	545,98 mW	545,38 mW
Aleatória	<i>Pareto On/Off</i>	Mínimo	<b>519,11 mW</b>	519,85 mW	520,1 mW	519,84 mW
Aleatória	Uniforme	Não-mínimo	<b>544,77 mW</b>	550,31 mW	547,03 mW	546,65 mW
Aleatória	Normal	Não-mínimo	<b>544,78 mW</b>	559,41 mW	547,05 mW	546,65 mW
Aleatória	<i>Pareto On/Off</i>	Não-mínimo	<b>519,11 mW</b>	525,55 mW	520,43 mW	520,23 mW
Complemento	Uniforme	Mínimo	560,11 mW	560,36 mW	548,73 mW	<b>548,1 mW</b>
Complemento	Normal	Mínimo	560,56 mW	559,41 mW	549,6 mW	<b>548,67 mW</b>
Complemento	<i>Pareto On/Off</i>	Mínimo	<b>525,21 mW</b>	525,55 mW	526,23 mW	525,89 mW
Complemento	Uniforme	Não-mínimo	560,11 mW	564 mW	549,85 mW	<b>549,46 mW</b>
Complemento	Normal	Não-mínimo	560,56 mW	569,8 mW	550,44 mW	<b>550,04 mW</b>
Complemento	<i>Pareto On/Off</i>	Não-mínimo	<b>525,21 mW</b>	527,06 mW	526,73 mW	526,48 mW



**Figura 37 - Simulações da transmissão de 100, 1000 e 10000 pacotes de 16 *flits* (distribuição espacial aleatória e temporal normal) para diferentes algoritmos adaptativos mínimos.**

A Figura 38 exibe graficamente os resultados das simulações dos algoritmos adaptativos mínimos para a distribuição espacial do tipo complemento e distribuição temporal uniforme. Note que para este tipo de tráfego, o algoritmo adaptativo que apresenta melhores resultados é o *Negative-First* (NF), chegando a obter uma redução na dissipação média de potência de até 19,64 mW. Nestas avaliações o algoritmo *North-Last* (NL) também reduziu a dissipação de potência na rede em até 19,16 mW. O algoritmo *West-First* (WF) se manteve similar ao algoritmo XY.



**Figura 38 - Simulações da transmissão de 100, 1000 e 10000 pacotes de 16 *flits* (distribuição espacial do tipo complemento e temporal uniforme) para diferentes algoritmos adaptativos mínimos.**

## 5.6 Conclusão

Uma boa estimativa da dissipação de potência de uma NoC na fase de projeto permite explorar diversas arquiteturas de rede (com diferentes vazões e latências) que possuam a mesma dissipação de potência. Observou-se que no projeto de uma NoC, a utilização da técnica de *clock-gating* de forma automática, via diretivas de síntese, é fundamental na redução de ambos a dissipação de potência e o consumo de energia. Se esta opção não estiver disponível ao projetista, como no projeto direcionado a FPGAs, alternativas de *clock-gating* manual podem ser aplicadas, reduzindo a dissipação de potência e o consumo de energia.

Verificou-se a importância da atividade de chaveamento em ambos os enlaces e *buffers* de entrada, direcionando futuras modificações no modelo proposto para que esta atividade de chaveamento seja considerada. Ao se projetar uma NoC, um dos pontos de avaliação é o algoritmo de roteamento a ser empregado. Ao se analisar apenas a dissipação de potência de todos os algoritmos de roteamento empregados na NoC Hermes, verificou-se que a melhor opção para tráfegos com distribuição espacial uniforme é o algoritmo XY. Porém, para tráfegos com distribuição espacial do tipo complemento, a melhor opção de algoritmo de roteamento é o *Negative-First* (NF) mínimo.



## 6 CONCLUSÕES E TRABALHOS FUTUROS

---

### 6.1 Conclusões

No presente trabalho foi apresentado um modelo para estimativa da potência dissipada em NoCs, baseado nas taxas de recepção nos *buffers* dos roteadores. Adicionalmente ao modelo proposto, foi desenvolvido um fluxo genérico de aplicação do modelo, calibrando-o para a rede a ser avaliada.

Diferentemente dos modelos propostos na literatura, os quais estimam a potência a partir do grafo da aplicação e do volume de dados transmitido, este modelo requer a simulação RTL da NoC em análise para a obtenção das taxas de recepção nos *buffers* dos roteadores. O modelo proposto tem a vantagem de ser mais preciso que os modelos baseados em volume de comunicação, pois as taxas medidas nos *buffers* consideram os efeitos do congestionamento na rede. Entretanto, dependendo do tráfego a ser analisado a simulação RTL poderá apresentar um custo elevado de tempo de simulação. Para acelerar o processo de estimativa, o modelo pode ser implementado em níveis de abstração mais elevados.

O modelo foi adicionado ao framework ATLAS sob a forma do módulo HEFESTUS, o qual automatiza o processo de aplicação do modelo sobre as redes Hermes geradas no framework ATLAS. Para avaliar a aceleração do modelo implementado em um nível RTL (HEFESTUS) em relação à implementação em um nível mais elevado de abstração, o modelo proposto foi integrado ao modelo JOSELITO, o qual utiliza a técnica PAT e orientação a atores para reduzir o tempo de simulação da NoC Hermes. Verificou-se nesta comparação precisão similar nas duas implementações, com uma redução significativa no tempo de geração da estimativa devido à simulação da rede.

Este trabalho também apresentou um estudo da dissipação de potência da NoC Hermes, utilizando o módulo HEFESTUS. Avaliações mostraram que na rede Hermes de referência (sem a aplicação de qualquer técnica de redução da dissipação de potência), a dissipação de potência é definida pela atividade dos *buffers* de entrada nos roteadores (cerca de 90% da dissipação total). Observou-se a existência de grupos de dissipação de potência similar, divididos pelo tamanho total dos *buffers* nos roteadores. Foi avaliada uma NoC que utiliza a técnica de *clock-gating* de forma automática, via diretivas da ferramenta de síntese. Nesta avaliação observou-se uma grande redução na potência dissipada na rede, além de uma redução na área do circuito (cerca de 20%) e um aumento na frequência máxima de operação (cerca de 40%). Foi avaliada uma NoC Hermes que utiliza a técnica de *clock-gating* inserida manualmente no seu código VHDL. Esta opção é utilizada em ferramentas de síntese que não possuam a aplicação da técnica de *clock-gating* de forma automática. Resultados apontam uma redução na potência dissipada pela rede (cerca de 50%), porém observou-se um aumento na área do circuito (cerca de 3%) e não houve alteração na frequência máxima de operação.



Foram realizados testes para a verificação da dissipação de potência dos diversos algoritmos de roteamento disponibilizados pelo framework ATLAS. Resultados indicam que para tráfego com baixa colisão entre pacotes (exemplo, distribuição espacial aleatória) o melhor algoritmo é o XY, porém para tráfegos com muita colisão entre pacotes transmitidos (exemplo, distribuição espacial do tipo complemento) o melhor algoritmo de roteamento é o *Negative-First*.

## 6.2 Trabalhos Futuros

Citam-se como trabalhos futuro deste trabalho: (i) o refinamento do modelo proposto inserindo-se as contribuições na dissipação de potência da rede gerada pela atividade de chaveamento nos enlaces e nos *buffers* de entrada; (ii) comparação do modelo com outros modelos de estimativa da potência dissipada em NoCs; (iii) aplicação do modelo proposto sobre a estrutura de um MPSoC; (iv) desenvolvimento de uma arquitetura de roteador para baixa dissipação de potência.

O refinamento do modelo proposto inserindo-se as parcelas de dissipação de potência referentes à atividade de chaveamento nos enlaces e *buffers* de entrada é a seqüência natural deste trabalho. A variação da atividade de chaveamento nos enlaces é responsável por até 27% de erro na estimativa da dissipação de potência em um roteador. Já a variação da atividade de chaveamento no buffer de entrada é responsável por até 65% de erro na estimativa. O modelo proposto deverá ser refinado a fim de compensar estas fontes de erro na sua estimativa, tornando-o mais preciso.

Após o refinamento do modelo, deve-se avaliar o modelo proposto em relação aos demais modelos de estimativa da potência dissipada em NoCs. A maioria dos modelos de estimativa disponíveis é baseado no volume de dados transmitidos. Desta forma, espera-se que o modelo proposto seja mais preciso em relação aos demais modelos.

Outro trabalho futuro compreende a estimativa de potência dissipada em um MPSoC completo, incluindo processadores, memórias e a NoC. Pode-se utilizar estimativa de potência em processadores a partir do conjunto de instruções executadas e a estimativa em memória a partir no número de transações de leitura/escrita realizadas. Assim poder-se-ia integrar estes três modelos em uma plataforma real, por exemplo, a plataforma HEMPS [WOS07], para realizar estimativas da dissipação de potência desta plataforma ao executar diferentes conjuntos de aplicações.

O quarto trabalho futuro compreende o desenvolvimento de uma arquitetura de roteador para baixa dissipação de potência. Neste trabalho observou-se que a utilização da técnica de *clock-gating* de forma automática reduz, significativamente, a potência de uma rede intra-chip. A utilização desta técnica juntamente com o emprego do algoritmo de roteamento que apresenta melhores resultados de dissipação de potência poderá gerar uma arquitetura de roteador que apresenta dissipação de potência reduzida. Outras técnicas de redução como, *power-gating*, deverão ser avaliadas e inseridas no roteador da rede Hermes a fim de desenvolver uma arquitetura de roteador que apresente a menor dissipação de potência, mantendo-se bons resultados de vazão e latência média.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- [AGA07] Agarwal, N; Dimopoulos, J. “Automated Power Gating of Registers using CoDeL and FSM Branch Prediction”. In: Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS), 2007. pp. 294-303.
- [ALT02] ALTERA. Avalon Bus Specification. Reference Manual. Document Version 1.2, 2002.
- [AMB00] ARM Corp. AMBA 2.0 Specification. Capturado em: [http://www.arm.com/armtech/AMBA\\_Spec](http://www.arm.com/armtech/AMBA_Spec), 03/10/2000.
- [ATL07] Atlas - An Environment for NoC Generation and Evaluation. Capturado em: [http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex\\_us.html](http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex_us.html), 12/12/2007.
- [BAI01] Bainbridge, W; Furber, S. “Delay insensitive system-on-chip interconnect using 1-of-4 data encoding”. In: International Symposium on Asynchronous Circuits and Systems (ASYNC), 2001, pp.118-126.
- [BAI95] Bailey, D. “The NAS Parallel Benchmarks 2.0”. NAS Technical Report, 1995.
- [BAN04] Banerjee, N; Vellanki, P; Chatha, K. “A Power and Performance Model for Network-on-Chip Architectures”. In: Design, Automation & Test in Europe (DATE), 2004, pp.1250-1255.
- [BEN01] Benini, L; De Micheli, G. "Powering Networks on Chip", In: International Symposium on System Synthesis (ISSS), 2001, pp. 33-38.
- [BEN02] Benini, L; De Micheli, G. "Networks on Chips: a New SoC Paradigm". Computer, Vol.35, No.1, pp.70-78, 2002.
- [BEN06] Benini, L. “Application Specific NoC Design”. In: Design, Automation & Test in Europe (DATE), 2006, pp.491-495.
- [BEN94] Benini, L; Favalli, M; Ricco, B. “Analysis of hazard contribution to power dissipation in CMOS IC’s”. In: International Workshop on Low Power Design (IWLDP), 1994. pp. 27-32.
- [BER05] Bertozzi, D; Jalabert, A; Murail, S; Tamhankar, R; et al. "NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip", IEEE Transactions On Parallel and Distributed Systems, Vol. 16, No. 2, pp.113 – 129, 2005.
- [BON01] Boni, A; Pierazzi, A; Vecchi, D., "LVDS I/O interface for Gb/s-per-pin operation in 0.35- $\mu$ m CMOS ," IEEE Journal of Solid-State Circuits, Vol.36, No.4, pp.706-711, 2001.
- [BUY06] Buyuktosunoglu, A; Isci, C; Cher, C; Bose, P; Martonosi, M. “An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget”. In: International Symposium on Microarchitecture (MICRO), 2006, pp.347-358.
- [CHA05] Chang, K; Shen, J; Chen, T. “A Low-Power Crossroad Switch Architecture and Its Core Placement for Network-On-Chip”. In: International Symposium on Low Power Electronics and Design (ISLPED), 2005, pp.375-380.

- [CUM02] Cummings, C. "Simulation and Synthesis Techniques for Asynchronous FIFO Design". In: Synopsys Users Group Conference (SNUG), 2002, pp. 1-21.
- [DAL01] Dally, W; Towles, B. "Route Packets, not Wires: On-Chip Interconnection Networks". In: Design Automation Conference (DAC), 2001, pp.684-689.
- [DAL03] Dally, W; Towles, B. "Principles and Practices of Interconnection Networks". Morgan Kaufmann Inc., San Francisco – USA, 2003, 550 p.
- [DIN95] Ding, C-S; Pedram, M. "Tagged probabilistic simulation provides accurate and efficient power estimates at the gate level". In: Symposium on Low Power Eletronics, 1995. pp. 42-43.
- [DON03] Donno, M; Ivaldi, A; Benini, L; Macii, E. "Clock-tree power optimization based on RTL clock-gating". In: Design Automation Conference (DAC), 2003. pp.622-627.
- [DUA02] Duarte, D; Vijaykrishnan, N; Irwin, M; Tsai, Y-F. "Impact of technology scaling and packaging on dynamic voltage scaling techniques". In: ASIC/SOC Conference (ASIC/SOC), 2002. pp. 244- 248.
- [EIS04] Eisley, N; Peh, L.-S. "High-Level Power Analysis for on-Chip Networks". In: International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2004, pp. 104-115.
- [EIS06] Eisley, N; Soteriou, V; Peh, L. "Low power: High-level power analysis for multi-core chips". In: International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2006, pp. 389-400.
- [GAR99] Garret, D; Stan, M; Dean, A. "Challenges in Clock Gating for Low Power ASIC Methodology". In: International Symposium on Low Power Electronics and Design (ISLPED), 1999. pp. 176-181.
- [GNU08] Ferramenta para criar gráficos Gnuplot, Capturado em: <http://www.gnuplot.info/>, 2008.
- [GUI08] Guindani, G; Reinbrecht, C; Da Rosa, T; Calazans, N; Moraes, F. "NoC Power Estimation at the RTL Abstraction Level". In: Annual Symposium on VLSI (ISVLSI), 2008, pp. 475-478.
- [HEG00] Hegde, R; Shanbhag, N. "Toward achieving energy efficiency in presence of deep submicron noise". IEEE Transactions on VLSI Systems, Vol.8, No.4, pp. 379-391, 2000.
- [HEW77] Hewitt, C. "Viewing control structures as patterns of passing messages". Journal of Artificial Intelligence, Vol.8, No.3, pp. 323-363, 1977.
- [HU03] Hu, J; Marculescu, R. "Energy-aware mapping for tile-based NoC architectures under performance constraints". In: Asia and South Pacific Design Automation Conference (ASP-DAC), 2003, pp. 233-239.
- [HU04] Hu, Z; Buyuktosunoglu, A; Srinivasan, V; Zyuban, V; Jacobson, H; Bose, P. "Microarchitectural Techniques for Power Gating of Execution Units". In: International Workshop on Low Power Design (IWLPD), 2004. pp. 32-37.
- [IBM02] IBM. The CoreConnect™ Bus Architecture. Capturado em: [http://www.3ibm.com/chips/techlib/techlib.nsf/productfamilies/CoreConnect\\_Bus\\_Architecture](http://www.3ibm.com/chips/techlib/techlib.nsf/productfamilies/CoreConnect_Bus_Architecture), 24/11/2002.

- [IL05] Il-Gu L; Jin L; Sin-Chong P. "Adaptive routing scheme for NoC communication architecture". In: International Conference on Advanced Communication Technology (ICACT), 2005. pp.1180-1184.
- [ISC06] Isci, C; Contreras, G; Martonosi, M. "Live, Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management". In: International Symposium on Microarchitecture (MICRO), 2006, pp. 359-370.
- [ISH05] Ishikawa, M. "A 4500 MIPS/W, 86 $\mu$ A Resume-Standby, 11 $\mu$ A Ultra-Standby Application Processor for 3G Cellular Phones". IEICE Transactions on Electronics, Vol.88-C, No.4, 2005. pp. 528-535.
- [ITO99] Ito, S.A; Carro, L; Jacobi, R.P. "Designing a Java microcontroller to specific applications", In: Symposium on Integrated Circuits and System Design (SBCCI), 1999. pp. 12-15.
- [ITO01] Ito, S.A; Carro, L; Jacobi, R.P. "Making Java work for microcontroller applications", IEEE Design & Test of Computers, Vol. 18, No. 5, 2001. pp. 100-110.
- [JAV08] Linguagem de programação Java. Capturado em <http://www.java.com/>, 15/12/2008.
- [JER05] Jerraya, A; Wolf, W. "Multiprocessor Systems-on-Chips". Morgan Kaufmann Publishers Inc., San Francisco – USA, 2005, 608 p.
- [KIM05] Kim, J; Park, D; Nicopoulos, C; Vijaykrishnan, N; Das, C. "Design and Analysis of an NoC Architecture from Performance, Reliability and Energy Perspective". In: Symposium On Architecture For Networking And Communications Systems (ANCS), 2005, pp.173-182.
- [KIS06] Kistler, M.; Perrone, M.; Petrini, F. "Cell Multiprocessor Communication Network: Built for Speed". IEEE Micro, Vol. 23, No. 6, pp. 10-23, 2006.
- [LEE03] Lee, E.A; Neuendorffer, S; Wirthlin, M. J. "Actor-Oriented Design of Embedded Hardware and Software". In: Journal of Circuits, Systems, and Computers, Vol. 12, No. 3, pp. 231-260, 2003.
- [LIU04] Liu, J; Eker, J; Janneck, J.W; Liu, X.J; Lee, E A. "Actor-oriented control system design: A responsible framework perspective". IEEE Transactions on Control Systems Technology, Vol. 12, No. 2, pp. 250-262, 2004.
- [MAR05] Marcon, C; Calazans, N; Moraes, F; Susin, A; Reis, I; Hessel, F. "Exploring NoC Mapping Strategies: An Energy and Timing Aware Technique". In: Design, Automation & Test in Europe (DATE), 2005, pp.502-507.
- [MAT08] Matsutani, H; Koibuchi, M; Wang, D. "Run-Time Power Gating of On-Chip Routers Using Look-Ahead Routing". In: Asia and South Pacific Design Automation Conference (ASP-DAC), 2008. pp. 55-60.
- [MIC06] De Micheli, G; Benini, L. "Networks on Chips". Morgan Kaufmann Publishers Inc., San Francisco – USA, 2006, 408 p.
- [MOR04] Moraes, F; Calazans, N; Mello, A; Möller, L; Ost, L. "Hermes: An infrastructure for low area overhead packet-switching networks on chip". In: The VLSI Integration, Vol. 38, No. 1, pp. 69-93, 2004.

- [NAI97] Nair, R; Hopkins, M. E. "Exploiting instruction level parallelism in processors by caching scheduled groups". SIGARCH Computer Architectures News, Vol. 25, No. 2, pp 13-25, 1997.
- [OH01] Oh, J; Pedram, M. "Gated Clock Routing for Low-Power Microprocessor Design". In: IEEE Transactions on CAD/ICAS, Vol. 20, No. 6, pp. 715-722, 2001.
- [OST04] Ost, L; Moraes, F. "Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware". PPGCC-PUCRS, 2004.
- [OST05] Ost, L; Mello, A; Palma, J; Moraes, F; Calazans, N. "MAIA - A Framework for Networks on Chip Generation and Verification". In: Asia and South Pacific Design Automation Conference (ASP-DAC), 2005, pp. 49-52.
- [OST08] Ost, L; Möller, L; Indrusiak, L; Moraes, F; Määttä, S; Nurmi, J. e Glesner, M. "A Simplified Executable Model to Evaluate Latency and Throughput of Networks-on-Chip". In: Symposium on Integrated Circuits and System Design (SBCCI), 2008, pp. 170-175.
- [PAL05] Palma, J; Marcon, C; Moraes, F; Calazans, N; Reis, R; Susin, A; "Mapping Embedded Systems onto NoCs – The Traffic Effect on Dynamic Energy Estimation". In: Symposium on Integrated Circuits and System Design (SBCCI), 2005, pp. 196-201.
- [PAL07] Palma J; Indrusiak, L; Moraes, F; Ortiz, A; Glesner, M; Reis, R. "Inserting Data Encoding Techniques into NoC-Based Systems". In: Annual Symposium on VLSI (ISVLSI), 2007, pp. 299-304.
- [PED96] Pedram, M. "Power minimization in IC design: principles and applications". In: ACM Transactions on Design and Automation of Electronic Systems, Vol. 1, No. 1, pp. 3-56, 1996.
- [PON08] Pontes, J; Moreira, M; Soares, R; Calazans, N. "Hermes-GLP: A GALS Network on Chip Router with Power Control Techniques". In: Annual Symposium on VLSI (ISVLSI), 2008. pp. 347-352.
- [PTO08] "Ptolemy II - Heterogenous Modeling and Design". Capturado em: <http://ptolemy.berkeley.edu/ptolemyII/>, 17/04/2008.
- [RAB96a] Rabaey, J. M. "Digital Integrated Circuits: A design perspective". Prentice-Hall, New Jersey – USA, 1996, 702 p.
- [RAB96b] Rabaey, J; Pedram, M. "Low Power Design Methodologies", Kluwer Academic Publishers, Hingham – USA, 1996, 367p.
- [SIM02] Simunic, T; Stephen, B. "Managing Power Consumption in Networks on Chip". In: Design, Automation & Test in Europe (DATE), 2002, pp. 110-116.
- [SRI05] Srinivasan, K; Chatha, K. "A technique for low energy mapping and routing in network-on-chip architectures". In: International Symposium on Low Power Electronics and Design (ISLPED), 2005, pp. 387-392.
- [SYL01] Sylvester, D; Chenming, W. "Analytical modeling and characterization of deep-submicrometer interconnect". In: Proceedings of the IEEE, Vol.89, No.5, pp.634-664, 2001.

- [SYN07] The PrimeTime® Static Timing Analysis (STA). Capturado em: [http://www.synopsys.com/products/analysis/primetime\\_ds.html](http://www.synopsys.com/products/analysis/primetime_ds.html), 24/06/2007.
- [SYN08] Design Compiler Ultra® RTL synthesis. Capturado em: [http://www.synopsys.com/products/logic/dc\\_ultra\\_ds.html](http://www.synopsys.com/products/logic/dc_ultra_ds.html), 26/11/2008.
- [TIL07] The TILE64 Processor Product Brief. Captured at: [www.tilera.com](http://www.tilera.com), 06/09/2007.
- [USA06] Usami, K; Ohkubo, N; "A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals". In: International Conference on Computer Design (ICCD), 2006.
- [VAN07] Vangali, S; Howard, J; Ruhi, G; Dighe, S. "An 80-Tile 1.28 TFLOPS Network-on-Chip in 65nm CMOS". In: International Solid State Circuits Conference (ISSCC), 2007, pp 98-99.
- [WIN01] Wingard, D. "MicroNetwork-Based Integration for SoCs". In: Design Automation Conference (DAC), 2001, pp.673-677.
- [WOS07] Woszezenki, C; Moraes, F. "Alocação de Tarefas e Comunicação entre Tarefas em MPSoCS". PPGCC-PUCRS, 2007, 121 p.
- [XFA08] 0.35 Micron Modular Low Voltage / Non Volatile Mixed Signal Technology. Capturado em: <http://www.xfab.com/xfab/frontend/index.php?itid=305>, 2008.
- [YE02] Ye, T; Benini, L; De Micheli, G. "Analysis of Power Consumption on Switch Fabrics in Network Routers". In: Design Automation Conference (DAC), 2002, pp.524-529.
- [YE03] Ye, T; Benini, L; De Micheli, G. "Packetized On-Chip Interconnect Communication Analysis for MPSoC". In: Design, Automation & Test in Europe (DATE), 2003, pp.344-349.
- [ZHA99] Zhang, H; Wan, M; George, V; Rabaey, J. "Interconnect architecture exploration for low-energy configurable single-chip DSPs". In: Computer Society Workshop on VLSI (VLSI), 1999, pp.2-8.



## APÊNDICE I – PUBLICAÇÕES ACEITAS E SUBMETIDAS

---

As publicações realizadas no período do mestrado demonstram a aceitação do presente trabalho na comunidade científica, nacional e internacional. As publicações aceitas e submetidas são listadas abaixo.

Guindani, G.; Reinbrecht, C.; Raupp, T.; Calazans, N.; Moraes, F. “*NoC Power Estimation at the RTL Abstraction Level*”. In: IEEE Computer Society Annual Symposium on VLSI Design (ISVLSI'08), 2008. pp. 475-478.





# APÊNDICE II – TUTORIAL DE CALIBRAÇÃO DO MODELO DE ESTIMATIVA DA DISSIPACÃO DE POTÊNCIA EM REDES INTRA-CHIP

---

## Introdução

O processo de calibração de NoCs consiste em estimular a rede às taxas pré-definidas para diferentes tempos de simulação, a fim de se obter informações que, a partir de um método matemático, resultem em uma equação que relacione potência dissipada em função da taxa de recepção dos *buffers*. Esse processo possui sete etapas: Organização da Área de Trabalho, Geração da Rede, Geração dos Cenários, Síntese Lógica, Simulação, Avaliação da Potência e Geração das Equações.

A organização da área de trabalho é necessária para criar a estrutura de pastas utilizada pelos scripts/ferramentas que automatizam o processo de calibração.

A geração da rede é feita através do framework Atlas, e caso a rede já esteja disponível, é possível ir diretamente para a etapa de criação dos cenários de tráfego. Na Atlas, pode-se escolher entre diversos parâmetros para a geração da rede, porém é necessário um tamanho de rede de no mínimo 3x3. Esse requisito se deve ao fato de que a calibração é feita apenas sobre roteadores que possuam todas as portas de entrada possíveis disponíveis, ou seja, apenas redes com dimensão mínima 3x3 possuem tais roteadores.

A terceira etapa é a geração de cenários de tráfego, que consiste em criar diversos tráfegos controlados para estimular a rede durante as simulações. Deve-se gerar 7 cenários de tráfego, cada um terá uma taxa de injeção de pacotes diferente. Os tráfegos devem apresentar as seguintes taxas de injeção: 0%, 5%, 10%, 20%, 30%, 40% e 50% da capacidade máxima de injeção na rede. Cada cenário deve estimular todas as 5 portas de entrada e saída do roteador alvo, na mesma taxa de injeção, de forma que não ocorra colisão.

Com a rede e os cenários de tráfego gerados, o próximo passo é a síntese lógica do roteador que será utilizado na calibração. A ferramenta utilizada na síntese é o DC\_SHELL, da Synopsys. Esse tutorial disponibiliza um script para sintetizar o roteador, sendo necessárias apenas algumas modificações.

O quarto passo é a simulação da rede para os 7 cenários de tráfego. Com esse passo são obtidos todos os arquivos necessários para o passo de avaliação da dissipação de potência. Cada cenário é simulado 5 vezes, para 20, 40, 60, 80 e 100 microssegundos (us). Para a realização das simulações são disponibilizados scripts para a ferramenta Modelsim e também um script TCL/TK para a repetição das simulações. São necessárias algumas modificações nos scripts, para a compilação dos arquivos corretos e para a captura dos sinais do módulo sintetizado.

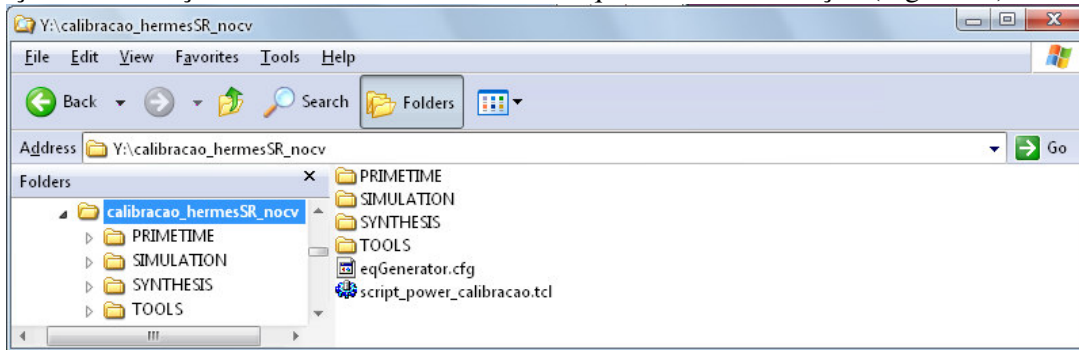
O penúltimo passo é a geração dos relatórios de dissipação de potência. “Para isso é necessário que já estejam na pasta de avaliação um arquivo “.db”, gerado no momento da síntese, e

os arquivos “.vcd” gerados em cada simulação. São disponibilizados scripts para a execução na ferramenta *PrimeTime* da Synopsys, sendo necessárias modificações, que serão detalhadas no decorrer do tutorial.

Após a conclusão de todas as avaliações uma tabela é gerada, e a técnica de ajuste à reta é utilizada para a gerar uma equação relacionando potência e taxa de recepção dos *buffers*. As equações podem ser calculadas manualmente, porém existe um programa que realiza esse cálculo. Esse programa é analisado ao final do tutorial, juntamente com as mudanças que devem ser realizadas no programa para que este funcione corretamente sobre qualquer rede.

### 1. Organização da Área de Trabalho

Para iniciar o processo de calibração é necessário extrair os arquivos compactados que estão no arquivo “*calibracao.zip*”. Após a extração, toda a estrutura de pastas necessária para a execução da calibração fica montada no diretório em que foi feita a extração (Figura 39).



**Figura 39 – Estrutura de pastas para o processo de calibração.**

Na pasta *PRIMETIME* estão disponíveis os arquivos necessários para que a ferramenta *PrimePower* possa realizar a avaliação de dissipação de potência, além dos scripts que devem ser utilizados pela ferramenta. Os arquivos gerados pela ferramenta ficarão nessa pasta, porém deverão ser copiados para a pasta *TOOLS/eqGenerator* para posterior utilização.

Na pasta *SIMULATION* se encontram os arquivos que irão ser utilizados no momento das simulações, como scripts, arquivos da NoC gerada na ATLAS, arquivos de tráfego, etc. Os arquivos “.vcd” que serão utilizados no momento da avaliação de dissipação de potência são gerados nessa pasta, mas devem ser copiados para a pasta *PRIMETIME*.

Na pasta *SYNTHESYS* estão disponíveis os arquivos necessários para a realização da síntese lógica. Após a síntese lógica, são gerados dois arquivos *netlist*, um “.vhd” e um “.db” nesta pasta, mas deverão ser copiados para outras pastas. O arquivo “.vhd” deve ser modificado e copiado para a pasta *SIMULATION/vsim* e o arquivo “.db” deve ser copiado para a pasta *PRIMETIME*.

A pasta *TOOLS* contém as ferramentas desenvolvidas no escopo desse trabalho para a realização de algumas modificações que são necessárias nos arquivos gerados pelo ModelSim e DC\_SHELL. Se a calibração não for realizada através do uso dos scripts fornecidos, estas modificações podem ser feitas manualmente e não é necessária a utilização das ferramentas que

estão nessa pasta. Porém, caso contrário estas ferramentas modificam os arquivos gerados, e se faz necessário a verificação de possíveis modificações nas ferramentas e sua recompilação. As ferramentas foram desenvolvidas na linguagem C++ e podem ser compiladas no ambiente Windows ou Unix. No código fonte de cada ferramenta existem comentários que explicam o funcionamento das mesmas.

Além disso, caso as ferramentas da pasta *TOOLS* sejam utilizadas, é necessário configurar o arquivo *eqGenerator.cfg*. Este arquivo possui diversas informações da NoC avaliada, que são utilizadas pelas ferramentas, como o tamanho da rede, o tamanho de flit e a frequência de operação da rede.

## 2. Criação da NoC

A geração da NoC é realizada através do framework Atlas, desenvolvida pelo grupo GAPH (Grupo de Apoio ao Projeto de Hardware). A Figura 40 ilustra a interface da ATLAS.



Figura 40 – Interface principal do framework ATLAS

Para iniciar o processo de geração da NoC é necessário clicar em *Projects->New Project*, e escolher o nome do projeto e o local onde ficarão os arquivos. Após a criação do projeto, ao clicar no botão *NoC Generation* é aberta a ferramenta Maia, onde podemos definir os parâmetros da rede a ser criada. A Figura 41 mostra a interface da ferramenta Maia.

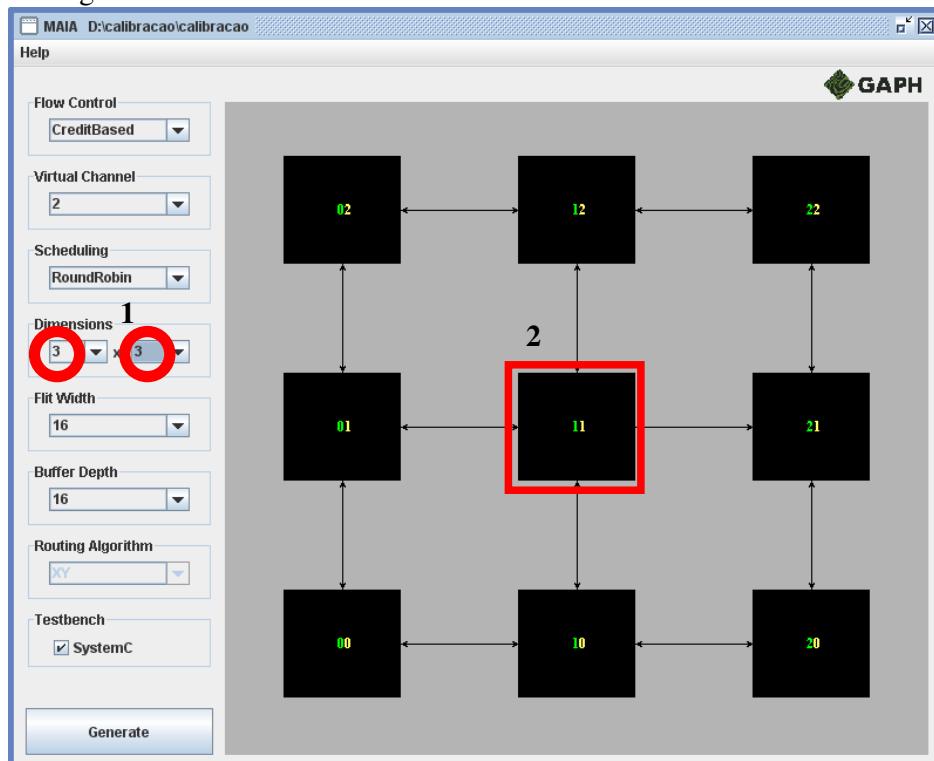


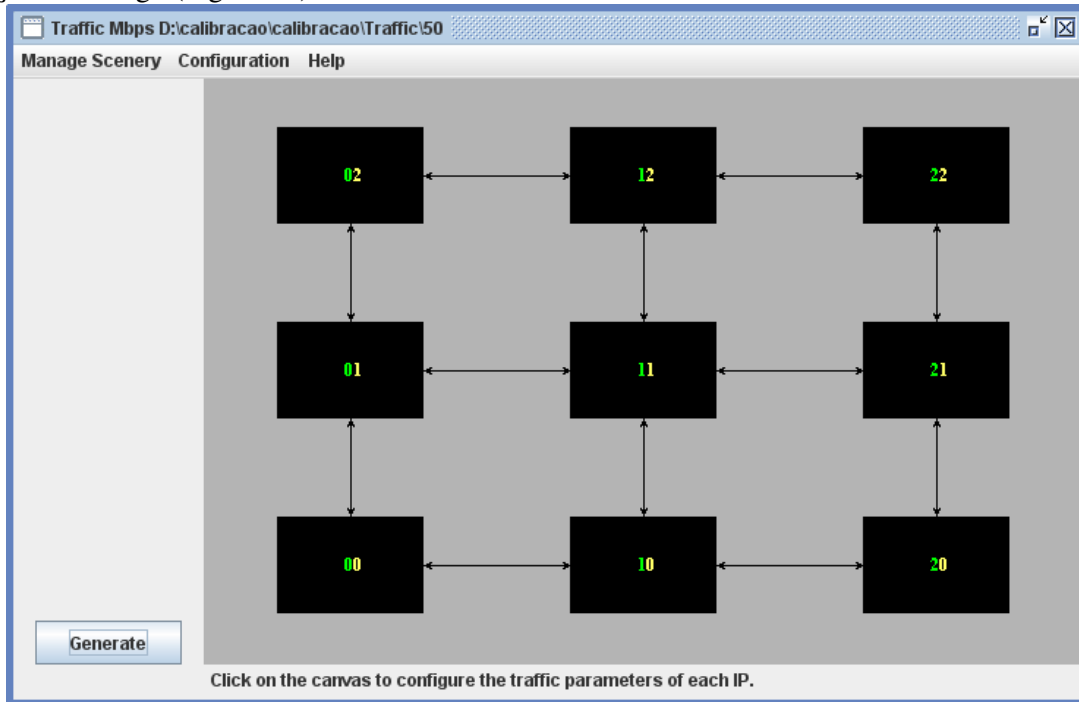
Figura 41 – Interface de geração de NoCs, a ferramenta MAIA no framework ATLAS.

Como dito anteriormente a única restrição nesse passo é o tamanho da rede, que deve ser

no mínimo 3x3 (1 na Figura 41), visto que é necessário que pelo menos um dos roteadores contenha as cinco portas de entrada (2, na Figura 41). Ao se clicar em “Generate” a NoC é gerada automaticamente.

### 3. Geração dos Cenários de Tráfego

Ao clicar-se no botão *Traffic Generation* do framework ATLAS, é aberta a interface de geração de tráfego (Figura 42).



**Figura 42 – Interface da ferramenta de geração de tráfego no framework ATLAS.**

Os cenários devem ser gerados de modo que as taxas de injeção atinjam 0%, 5%, 10%, 20%, 30%, 40% e 50% da capacidade máxima da rede. Os nomes dos cenários também devem seguir o mesmo padrão das taxas de injeção, ou seja, para o cenário com a taxa de injeção de 0%, o nome do cenário deverá ser 0, para o de 5% o nome do cenário deverá ser 5, e assim sucessivamente. O cálculo da taxa máxima de injeção da rede é descrito pela Equação 14.

$$TaxaMáxima = \frac{tamflit}{clockperiod} \quad (14)$$

Onde *tamflit* é o tamanho do *flit* em *bits*, definido no momento da criação da rede e *clockperiod* é a frequência de relógio do roteador da rede. Na Atlas a frequência de relógio padrão é 50 Mhz. Pegamos como exemplo uma NoC com tamanho de *flit* igual a 16 *bits* e frequência de operação de 50 Mhz. Aplicando a Equação 14 obtemos uma taxa máxima de injeção igual a 800 Mbps. Neste exemplo, as taxas utilizadas nos cenários de tráfego devem ser respectivamente: 0Mbps, 40Mbps, 80Mbps, 160Mbps, 240Mbps, 320Mbps e 400Mbps. Como a ferramenta Atlas não permite a criação de tráfego com 0Mbps, sugere-se a criação do tráfego com a taxa 1Mbps, que é a menor possível. Dessa forma na janela utilizada na calibração, apenas 1 *flit* será inserido na rede.

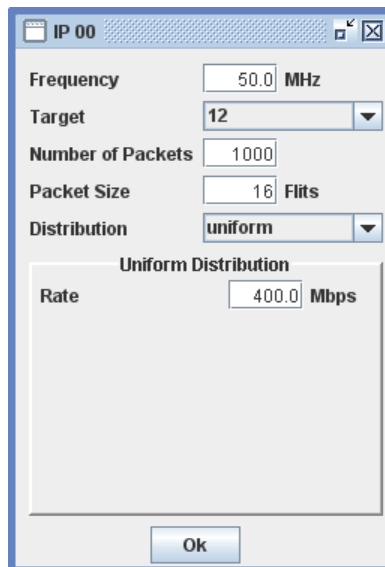
A distribuição espacial de todos os cenários são as mesmas, diferenciando-se apenas na

taxa de injeção. A Tabela 17 exibe a relação dos roteadores origem e destino.

**Tabela 17 - Roteadores origem e destino nos cenários de tráfego controlados.**

Origem	Destino
Roteador 00	Roteador 12
Roteador 01	Roteador 21
Roteador 11	Roteador 01
Roteador 21	Roteador 11
Roteador 02	Roteador 10

Nestes cenários de tráfego o tamanho dos *flits* é de 16 *bits*, e o número de pacotes deve ser suficiente para que haja tráfego na rede até o tempo de 100 microssegundos. Utilizou-se 1000 pacotes para cada tráfego, a frequência de 50Mhz e a distribuição temporal uniforme. Para gerar um tráfego diferente para cada roteador é necessário clicar em cima do roteador desejado e configurá-lo corretamente. A Figura 43 exibe a janela de configuração de tráfego para o roteador 00.



**Figura 43 – Interface de configuração de tráfego de um roteador (roteador 00) na ferramenta de geração de tráfego.**

Após configurar os roteadores de cada cenário de tráfego, basta clicar no botão “*Generate*” para que os tráfegos daquele cenário sejam criados. Finalizada esta etapa, basta copiar as pastas *NOC*, *Traffic*, *SC\_NoC* e os arquivos *simulate.do* e *topNoC.vhd*, para a pasta *SIMULATION/vsim*, na área de trabalho da calibração. Após a geração dos 7 cenários de tráfego, pode-se passar adiante para a etapa de síntese lógica do roteador da NoC a ser calibrada.

#### 4. Síntese Lógica

A ferramenta utilizada para a realização da síntese lógica é o *DC\_Shell*, da Synopsys. É utilizado um script para automatizar o processo de síntese. Os arquivos necessários estão disponíveis na pasta *SYNTHESIS*. O comando “*dc\_shell -db\_mode -dcsh\_mode -x "include synthesis.scr"*” é utilizado para a execução da ferramenta no sistema operacional Unix.

O script “*synthesis.scr*” contém os seguintes comandos:

```

analyze -f vhdl Hermes_package.vhd
analyze -f vhdl Hermes_crossbar_pwr.vhd
analyze -f vhdl Hermes_switchcontrol_pwr.vhd
analyze -f vhdl Hermes_buffer_pwr.vhd
analyze -f vhdl RouterCC_pwr.vhd
elaborate RouterCC
create_clock -name "clock" -period 20 -waveform {0.0 10.0}
set_output_delay 0.5 -clock [get_clocks clock] [all_outputs]
set_input_delay 0.5 -clock [get_clocks clock] [all_inputs]
check_design
compile -map_effort high -incremental_mapping
change_names -rules vhdl
vhdout_architecture_name = "RouterCC"

vhdout_use_packages = {
"IEEE.std_logic_1164",
"IEEE.std_logic_arith.all",
"IEEE.std_logic_textio.all",
"D_CELLS_33.all"}

write -f vhdl -hierarchy -output "RouterCC.vhd"
write -f db -hierarchy -output "RouterCC.db"
report_timing >> timing.txt
report -area >> area.txt
exit

```

Onde é necessário apenas mudar o nome dos arquivos que serão utilizados na síntese e o nome da entidade de mais alto nível hierárquico do projeto (*top\_level*), especificada após o comando *elaborate* (neste exemplo é RouterCC). Ao final também se pode escolher os nomes dos arquivos de saída que são gerados na síntese. Recomenda-se dar aos arquivos o mesmo nome da entidade *top\_level* sintetizada.

É recomendável também, alterar o nome das entidades que serão utilizadas na síntese, visto que no momento da simulação, caso o nome da entidade sintetizada seja o mesmo da entidade apenas simulada, o simulador irá sobrescrever a entidade não sintetizada e irá utilizar a última entidade compilada. Abaixo um exemplo de alteração feita para o arquivo do *buffer* do roteador.

```

Antes:
entity Hermes_buffer is
...
...
end Hermes_buffer;
...
architecture Hermes_buffer of Hermes_buffer is
...
...
end Hermes_buffer;

Depois:
entity Hermes_buffer_pwr is
...
...
end Hermes_buffer_pwr;
...

```

```
architecture Hermes_buffer_pwr of Hermes_buffer_pwr is
...
...
end Hermes_buffer_pwr;
```

Na NoC Hermes, se faz necessário a realização de tais mudanças para os arquivos de *buffer*, lógica de controle e *crossbar*. Também é necessária a modificação do arquivo do roteador, onde deve ser declarado o valor do endereço antes da síntese.

```
generic( address: regmetadeflit := ADDRESSN0101);
```

Na NoC Hermes, o nome da constante de endereço pode variar dependendo do tamanho do *flit*. É necessário verificar qual valor é atribuído ao roteador central no arquivo *vhd* da NoC. É importante também que cada arquivo tenha apenas uma arquitetura declarada. No caso do arquivo de *Switch Control*, que contém 6 arquiteturas declaradas, é necessário apagar todas as arquiteturas que não são utilizadas, antes de realizar a síntese. Um programa C++, denominado *ParserPWR.cpp*, realiza estas modificações, e este encontra-se na pasta *TOOLS/ParserPWR*.

Logo após a conclusão da síntese, é importante verificar no *netlist* gerado, se algum tipo de dados é redefinido em um package dentro do *netlist*. Isso geralmente acontece, pois a ferramenta de síntese captura os sinais utilizados pelo roteador e seus módulos, e os redefine em um package dentro do *netlist*. Então é necessário retirar esse package do *netlist* gerado, e utilizar as definições originais, que no caso da NoC Hermes estão definidas no arquivo *HermesPackage.vhd*.

Também é recomendado trocar o nome da entidade do roteador, para que caso exista mais de um roteador central, apenas um destes seja o roteador sintetizado. Também existe um programa que realiza tais modificações, sendo necessários apenas alguns ajustes no código. Este programa também cria o novo arquivo para a NoC, que instancia o roteador sintetizado dentro da rede. O nome do arquivo do programa C++ é *ParserVHDL.cpp* e pode ser encontrado na pasta *TOOLS/ParserVHDL*.

Depois de realizar todas as mudanças é necessário copiar o arquivo “.db” para a pasta *PRIMETIME* e o arquivo “.vhd” modificado para a pasta *SIMULATION/vsim*.

## 5. Simulação

Nesta etapa serão realizadas as simulações lógicas que permitem a geração dos arquivos VCD utilizados na próxima etapa. Nesta etapa serão simulados todos os cenários de tráfego gerados anteriormente, 5 vezes, sendo realizadas no total 35 simulações. Em cada cenário de tráfego são simuladas 5 janelas de tempo total de simulação, sendo que o tamanho default da janela de amostragem é de 1000 ciclos de relógio.

As simulações deverão ser executadas sobre o arquivo que contém a descrição da NoC, substituindo um dos roteadores centrais originais pelo roteador sintetizado. A ferramenta *ParserVHDL* realiza esta modificação gerando o arquivo *NoC\_pwr.vhd*, utilizado nas simulações.

Para tornar as equações geradas na calibração mais precisas, um monitor de tráfego é inserido na ultima simulação de cada cenário de tráfego. Este monitor conta a quantidade de *flits*



recebidos em cada janela de tempo, observando a taxa exata de recepção do *buffer*. Para a execução desta simulação é necessária a modificação do arquivo *NoC\_pwr.vhd*, adicionando o monitor.

A instanciação do monitor e a criação dos sinais que o conectam com o roteador, mudarão conforme a arquitetura da rede. No arquivo VHDL do monitor fornecido junto com o “zip” da calibração exista a descrição, sobre a forma de comentários, de como este monitor deverá ser instanciado na rede. Para o correto funcionamento dos scripts de simulação, o arquivo da NoC modificado, o qual instancia o monitor, deve ser *NOC\_monitores.vhd*, e o nome da instância do monitor no arquivo da NoC deverá ser *Monitor\_Router0101*.

Para gerar o arquivo VCD com a transição dos sinais do roteador sintetizado, os scripts de execução da ferramenta Modelsim (arquivos *simulateXX.do*) deverão conter o caminho correto do roteador sintetizado (na hierarquia de simulação), no exemplo da NoC Hermes este caminho é descrito na linha “*vcd add -r topNoC/NOC/Router0101/\**”

Os arquivos *simulateXX.do* tem praticamente a mesma estrutura do arquivo *simulate.do* (Gerado automaticamente pela ATLAS), porém algumas mudanças deverão ser realizadas nos arquivos. Ao total são 5 arquivos *simulateXX.do*, um para cada tempo de simulação 20, 40, 60, 80 e 100 us. As modificações são as seguintes:

Substituir linha:

```
vcom -work work -93 -explicit NOC/NOC.vhd
```

Pelas linhas:

```
vcom -work work -93 -explicit RouterCC_pwr.vhd  
vcom -work work -93 -explicit NOC_pwr.vhd
```

Inserir as seguintes linhas após o comando *set StdArithNoWarnings 1*:

```
vcd file routercc.vcd  
vcd on  
vcd add -r topNoC/NOC/Router0101/*
```

Substituir o tempo após o comando *run* pelo tempo correspondente do script em questão.

Inserir a seguinte linha após o comando *run*:

```
vcd flush
```

O arquivo *simulate100.do*, que corresponde a ultima simulação de cada cenário de tráfego, possui duas linhas a mais que os demais arquivos. Caso este script seja executado manualmente as seguintes linhas devem ser adicionadas:

```
Entre os comandos quit -sim e quit -f:  
do simulate_monitor.do  
quit -sim
```

O script que simula a rede juntamente com o monitor e escreve os flits recebidos em cada janela numa lista deve estar na mesma pasta dos scripts *simulateXX.do*. O nome deste script deve ser *simulate\_monitores.do*.

Como dito anteriormente, ao final de cada simulação é gerado um arquivo *.vcd*, que lista o chaveamento de todos os sinais de um módulo e seus sub-módulos. Como apenas o roteador central foi sintetizado, para que a ferramenta de avaliação de potência analise corretamente esse arquivo, é

necessário realizar uma modificação nestes arquivos. Os módulos que estão acima na hierarquia, do roteador sintetizado, devem ser retirados do vcd.

Exemplo - Caso o caminho seja topNoC/NOC/Router0101, o arquivo .vcd será da seguinte forma:

```
$scope module topnoc $end
$scope module noc $end
$scope module router0101 $end
$var wire 1 ! clock $end
...
$var wire 1 A q $end
$upscope $end
$upscope $end
$upscope $end
$upscope $end
$enddefinitions $end
```

As linhas *scope module topnoc \$end* e *scope module noc \$end* devem ser apagadas, e como 2 linhas de declaração de módulos foram apagadas, 2 linhas de *\$upscope \$end* imediatamente antes de *\$enddefinitions \$end* também deverão ser apagadas. Caso existam 3 módulos acima do módulo sintetizado, 3 *\$upscope \$end* devem ser apagados. Com isso o arquivo está pronto para ser utilizado pela ferramenta que avalia potência. Para NoCs que apresentem a estrutura de módulos do exemplo, topNoC, NOC e RouterXX, um programa pode ser utilizado para a realização destas modificações de forma automática. O nome do arquivo fonte do programa é *ParserVCD\_linux.cpp* e encontra-se na pasta *TOOLS/ParserVCD\_linux*.

Após a simulação de cada cenário obtêm-se 5 arquivos VCD, estes arquivos devem ser modificados (como descrito no parágrafo anterior) e copiados para a pasta *PRIMETIME*. O próximo passo, a avaliação de dissipação de potência é repetido para todos os cenários de tráfego.

#### 6. Avaliação de Dissipação de Potência

A avaliação de dissipação de potência é feita pela ferramenta *PrimeTime* da Synopsys e os arquivos necessários se encontram na pasta *PRIMETIME*. Para realizar as avaliações, já existem scripts que executam todos os comandos necessários, sendo necessário apenas alterar alguns parâmetros que variam de um projeto para outro. Existe um script para cada *vcd*, estes arquivos encontram-se na pasta *PRIMETIME*. Abaixo a linha de comando que deve ser digitada para disparar a ferramenta utilizando o script, no sistema operacional Unix:

```
pt_shell -file prime_timeXX.scr >> pt_shell_XX.output
Onde XX é o tempo de simulação que foi inserido no vcd.
```

Ao final de todas as avaliações, os relatórios gerados devem ser copiados para a pasta *TOOLS/eqGenerator/prYY*. Onde *YY* corresponde à taxa do cenário que está sendo avaliado.

```
0% - pr00
5% - pr20
10% - pr40
20% - pr80
30% - pr120
40% - pr160
```

50% - pr200

As linhas que necessitam ser verificadas nos scripts são as seguintes:

```
...  
read_db <nome do arquivo .db>  
current_design <nome da instancia do módulo top>  
...  
read_vcd -strip_path /<nome da entidade top do vcd> <nome do arquivo vcd>  
...  
create_power_waveforms -output pw<XX>  
report_power -hierarchy >> power_report<XX>.txt
```

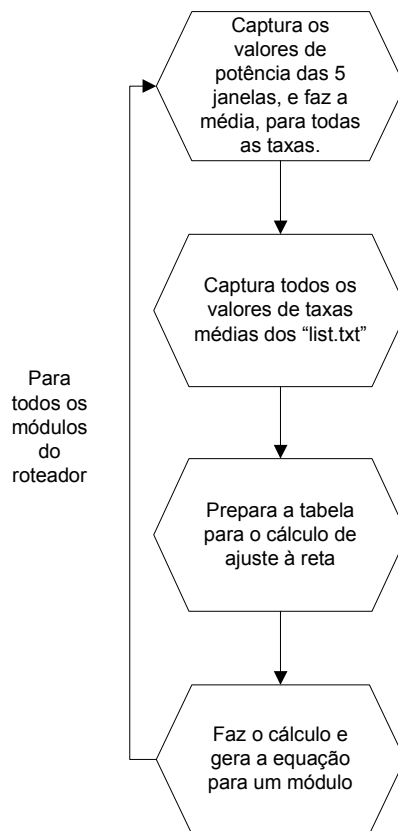
Onde *XX* é o tempo de simulação que foi inserido no vcd.

Exemplo:

```
...  
read_db RouterCC.db  
current_design RouterCC  
...  
read_vcd -strip_path /routercc routercc20.vcd  
...  
create_power_waveforms -output pw20  
report_power -hierarchy >> power_report20.txt
```

#### 7. Ajuste à Reta e Script de Automatização

Depois de todas as simulações e avaliações é possível obter-se uma equação que relacione potência e taxa de injeção. Essas equações são obtidas através da técnica de ajuste à reta. Este cálculo pode ser feito manualmente, ou utilizando uma ferramenta que realiza esse cálculo. Esta ferramenta, o eqGenerator encontra-se na pasta TOOLS/eqGenerator. Esta ferramenta captura todos os relatórios que estão nas pastas *prXX*, e realiza os cálculos necessários para obter as equações de dissipação de potência. O arquivo *eqGenerator.cpp*, que é o código fonte da ferramenta, e possui em seus comentários a descrição completa do cálculo. É possível que seja necessária alguma mudança ao se variar o projeto sob análise, mas estas possíveis alterações também estão detalhadas no código e há explicações de como realizá-las. Ao executar a ferramenta, é gerado um arquivo chamado *Equations.txt* que contém uma equação para cada módulo do roteador. A Figura 44 exhibe um fluxograma detalhando o funcionamento do programa.



**Figura 44 – Fluxograma do funcionamento do programa desenvolvido para aplicar a técnica de ajuste à reta.**

Com isso, finaliza-se o processo de calibração da NoC. Outro recurso utilizado na calibração, mas que não é obrigatório é o uso de scripts *tcl*, que automatizam esse processo. O arquivo *.zip* de calibração contém um script chamado *script\_power\_calibracao.tcl*, que executa todo o fluxo do processo de calibração, para NoCs que tenham mesma estrutura da NoC Hermes, gerada pela ferramenta Atlas. Para que esse script seja executado corretamente, é necessário usar um arquivo de configuração, chamado *eqGenerator.cfg*, onde deve ser informada a frequência de relógio utilizada na NoC, tamanho de flit, tamanho da janela em ciclos de relógio e dimensões da rede. Ao final do script, as equações de dissipação de potência estarão disponíveis na pasta *TOOLS/eqGenerator*, no arquivo *Equations.txt*.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)