

AVALIANDO O IMPACTO DE TÉCNICAS DE CODIFICAÇÃO DE DADOS SOBRE O CONSUMO DE POTÊNCIA EM NETWORKS-ON-CHIP

José C. S. Palma¹, Leandro Soares Indrusiak², Fernando G. Moraes³, Alberto Garcia Ortiz², Manfred Glesner², Ricardo A. L. Reis¹

¹ PPGC - II - UFRGS - Av. Bento Gonçalves, 9500, Porto Alegre, RS – Brazil
{jcspalma, reis}@inf.ufrgs.br

² MES – TU Darmstadt – Karlstr. 15, 64283 Darmstadt - Germany
{lsi, agarcia, glesner}@mes.tu-darmstadt.de

³ PPGCC - FACIN – PUCRS - Av. Ipiranga, 6681, Porto Alegre, RS – Brazil
moraes@inf.pucrs.br

ABSTRACT

This work investigates the reduction of power consumption in Networks-on-Chip (NoCs) through the reduction of transition activity using data coding schemes. The estimation of the NoC power consumption is performed with basis on macromodels which reproduce the power consumption on each internal NoC module according to the transition activity on its input ports. Such macromodels are embedded in a system model and a series of simulations are performed, aiming to analyze the trade-off between the power savings due to coding schemes versus the power consumption overhead due to the encoding and decoding modules.

1. INTRODUÇÃO

Networks-on-Chip (NoCs) são infra-estruturas essencialmente compostas por roteadores interconectados por canais de comunicação. As NoCs são adequadas para lidar com o paradigma GALS (GALS – *Globally Asynchronous, Locally Synchronous*) [1], pois oferecem uma comunicação assíncrona, além de outras vantagens como paralelismo, alta escalabilidade, reusabilidade e confiabilidade [2].

O crescimento do mercado para dispositivos portáteis alimentados por bateria acrescenta uma nova dimensão (potência) ao espaço de projeto VLSI, antes voltados para área, desempenho e testabilidade [3]. As NoCs resolvem o problema das altas capacidades encontradas nos barramentos, reduzindo a potência. Entretanto, o consumo de potência nos roteadores é significativo.

O consumo de potência em uma NoC cresce linearmente com a quantidade de transições de bit em pacotes transmitidos através da arquitetura de comunicação [4]. Utilizando a rede Hermes [5] como estudo de caso, este trabalho mostra que as transições de bit afetam o consumo de potência em mais de 370% nas linhas de interconexão, 180% nos buffers de entrada dos roteadores e 16% na lógica de controle dos roteadores.

Uma forma de reduzir o consumo de potência na

NoC, tanto na lógica quanto nas interconexões, é reduzindo a atividade de transição de bits, através da utilização de esquemas de codificação de dados [6]. Vários esquemas de codificação foram propostos no final dos anos 90, porém direcionados a arquiteturas de comunicação baseadas em barramentos.

A principal contribuição deste trabalho é a avaliação destes esquemas de codificação no contexto de sistemas baseados em NoCs, analisando o compromisso entre redução de potência obtida com a redução da transição de bit e o consumo extra do esquema de codificação.

This paper is organized as follows. Section 2 reviews coding schemes aiming to reduce power consumption in bus-based systems. Section 3 introduces the coding in NoCs. Section 4 presents the power consumption model for Networks-on-Chip. In Section 5 the analysis on power consumption is explained. Section 6 presents some experimental results and Section 7 presents the conclusions and future works.

2. ESQUEMAS DE CODIFICAÇÃO

Alguns esquemas de codificação requerem um conhecimento prévio dos parâmetros estatísticos do tráfego de entrada, o que nem sempre se faz disponível. Este trabalho enfatiza somente os esquemas de codificação que não requerem tal conhecimento, pois o objetivo aqui é aplicá-los a sistemas em geral baseados em Networks-on-Chip. Até o momento dois esquemas de codificação foram implementados e avaliados, no contexto deste trabalho, quanto ao seu efeito em um sistema onde a estrutura de interconexão é uma NoC: *Bus-Invert* [7] e *Adaptive Encoding* [6].

2.1. Codificação Bus-Invert

O método *Bus-Invert* [7] utiliza uma linha de controle a mais no canal de comunicação, chamada de *invert*, que indica quando o dado transmitido nas linhas de comunicação está ou não invertido. O dado é invertido quando a distância de *Hamming* (a quantidade de bits diferentes) entre este dado e o dado atual no canal

de comunicação (ou seja, o último dado que foi transmitido) for maior do que a metade do número de linhas no canal de comunicação.

2.2. Codificação Adaptive Probability Encoding

Em [6], os autores propõem uma arquitetura geral de codificação/decodificação capaz de adaptação online, de acordo com os dados transmitidos. Este esquema opera bit-a-bit, em vez de palavra-a-palavra, ignorando a correlação espacial entre bits de uma mesma palavra.

A codificação é feita com base em informações estatísticas aproximadas coletadas através da observação da seqüência de bits em uma janela de tamanho fixo. Estas informações estatísticas dizem respeito às probabilidades de ocorrência dos quatro possíveis pares de valores consecutivos de um único bit ($P_{0,0}$, $P_{0,1}$, $P_{1,0}$ e $P_{1,1}$). Este esquema utiliza quatro funções diferentes de codificação para determinar a palavra de saída. A seleção da função a ser utilizada a cada instante é feita de acordo com as frequências que estimam as probabilidades da janela de observação atual.

3. INSERINDO ESQUEMAS DE CODIFICAÇÃO DE DADOS EM NETWORKS-ON-CHIP

Em NoCs, os dados são transmitidos em pacotes, os quais são enviados através de roteadores, partindo de um núcleo de origem, até atingir um núcleo de destino. Estes pacotes são compostos por um cabeçalho (contendo informações de roteamento) e por um corpo (contendo os dados a serem transmitidos). No caso da rede Hermes [5], utilizada como estudo de caso neste trabalho, o cabeçalho é composto por dois *flits*, contendo o endereço de destino do pacote, bem como o tamanho do mesmo. Assim, em uma abordagem que combina esquemas de codificação e uma Network-on-Chip, não é vantajoso codificar o cabeçalho do pacote, já que a informação contida no mesmo deve ser lida pela lógica de roteamento de cada roteador, a cada *hop*¹ tomado pelo pacote, desde sua origem até seu destino.

Desta forma, as operações de codificação e decodificação podem ser executadas somente em sua origem, convertendo os dados originais em dados codificados, que são transmitidos através da NoC, e no seu destino, recuperando os dados originais. Sendo assim, os módulos de codificação e decodificação foram inseridos em *wrappers* que contém também o núcleo local e fazem a comunicação deste com as portas locais dos roteadores. Ou seja, comunicam o núcleo local com a estrutura de interconexão (NoC). Uma vantagem do *Adaptive Encoding* é que ele pode ser utilizado em qualquer NoC sem alteração da mesma. Já o *Bus-Invert*, requer alterações na NoC, de forma a inserir um bit de controle em todos os módulos internos e nos canais de comunicação.

A Figura 1 ilustra um *wrapper* conectado à porta local de um roteador Hermes. Os pacotes endereçados ao núcleo local, passam antes pelo módulo *decoder*, antes de serem entregues ao núcleo. Da mesma forma, os pacotes gerados pelo núcleo local são codificados no módulo *encoder* antes de serem transmitidos pela NoC. As portas restantes do roteador devem ser conectadas a outros roteadores da NoC (não mostrados na figura), de acordo com sua topologia.

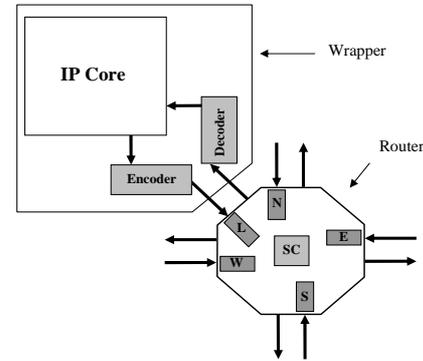


Figura 1: Localização dos módulos *encoder* e *decoder*.

4. MODELO DE CONSUMO DE POTÊNCIA EM NOCS

O consumo de potência em um sistema origina-se da operação dos núcleos e dos componentes de interconexão entre estes núcleos. O consumo de potência é proporcional à atividade de chaveamento originada pelos pacotes que se movem através da NoC. Canais de comunicação e roteadores dissipam potência. Vários autores [4] propõem estimar o consumo de potência na NoC através da avaliação do efeito do tráfego em cada componente da mesma.

O consumo de potência do roteador é estimado separando-se o consumo de potência no buffer do consumo de potência na lógica de controle. Também se deve estimar o consumo de potência nos canais de comunicação entre roteadores e entre roteador e núcleo local. Neste trabalho foram considerados comprimentos de 5mm para os canais entre roteadores e 0.25 mm para os canais locais.

Nos experimentos conduzidos neste trabalho foi utilizada uma rede Hermes de topologia grelha com seis configurações diferentes. Os parâmetros de consumo de potência nos componentes da NoC foram obtidos através da variação da largura de *flit* (8 e 16 bits) e da variação da profundidade dos buffers (4, 8 e 16 *flits*). Para cada configuração foram simulados envios de diferentes pacotes de 128 *flits* através da NoC, com diferentes padrões de transições de sinais em sua estrutura, variando de 0 a 127 transições em cada linha de comunicação.

Considerando uma abordagem com codificação de dados, é também necessário estimar o consumo de potência nos módulos de codificação e decodificação.

¹ Um *hop* é a distância entre dois roteadores vizinhos em uma NoC.

Estes módulos também foram simulados com os mesmos padrões de tráfego utilizados na simulação da NoC. Entretanto, até o momento os módulos de codificação e decodificação somente foram simulados com a configuração para largura de *flit* igual a 8 bits.

O fluxo utilizado para a aquisição dos parâmetros de consumo de potência consiste de três etapas. A primeira etapa parte da descrição VHDL da NoC (sem esquemas de codificação) e de arquivos de tráfego, ambos obtidos através da ferramenta MAIA [8]. Os arquivos de tráfego de entrada alimentam a NoC através das portas locais dos roteadores, modelando o comportamento dos núcleos locais. Um simulador VHDL aplica sinais de entrada à NoC ou à qualquer componente da NoC, como por exemplo um roteador ou um de seus componentes internos (buffers ou lógica de controle), de acordo com os arquivos de tráfego. A simulação produz listas de sinais, armazenando suas transições ao longo do tempo. Estas listas são convertidas para estímulos elétricos em formato PWL que são usados posteriormente na simulação SPICE (terceira etapa). Esta conversão é feita através de um programa desenvolvido no escopo deste trabalho.

Na segunda etapa, o módulo a ser avaliado (como por exemplo num buffer) é sintetizado através da ferramenta *LeonardoSpectrum* utilizando uma biblioteca de células para uma tecnologia específica, tal como TSMC 0.35. A ferramenta produz uma *netlist* HDL que é, em seguida, convertida para uma *netlist* SPICE através de um outro conversor também desenvolvido no escopo deste trabalho.

A terceira etapa consiste na simulação SPICE do módulo em avaliação. Nesta etapa o simulador recebe como entradas a *netlist* SPICE do módulo, os sinais elétricos de entrada produzidos na primeira etapa e uma biblioteca com portas lógicas descritas em SPICE. As informações elétricas resultante desta etapa são utilizadas na construção de um macromodelo de consumo de potência do componente avaliado, utilizando um determinado tráfego.

Posteriormente, este processo é repetido para uma nova versão da NoC, agora utilizando um esquema de codificação. No caso do *Adaptive Encoding*, somente os módulos *encoder* e *decoder* precisam ser agora avaliados. Já no caso do *Bus-Invert*, é necessário analisar novamente todos os módulos da NoC, pois este esquema requer a inserção do bit de controle, o que altera os parâmetros de potência dos mesmos.

4.1. Definição do Modelo

O parâmetro *APH* (*Average Power per Hop*) é utilizado para indicar o consumo médio de potência dinâmica em um único *Hop* percorrido por um pacote transmitido através da NoC. O *APH* pode ser dividido em três componentes: (i) potência média consumida em um roteador composto de buffers e lógica interna para roteamento e chaveamento (*APR* – *Average Power per*

Router); (ii) potência média consumida no canal de comunicação entre roteadores (*APL*); e (iii) potência média consumida no canal de comunicação entre um roteador e seu núcleo local (*APC*). A Equação 1 representa a potência média consumida por um pacote transmitido através de um roteador, um canal de comunicação entre roteadores e um canal de comunicação local.

$$APH = APR + APL + APC \quad (1)$$

Além disso, os resultados obtidos mostram que uma avaliação melhor da potência média consumida no roteador pode ser feita dividindo-a em: (i) potência média do buffer (*APB*) e (ii) potência média do módulo de controle (*APS*). Isto se deve ao fato de que o efeito das transições de sinais sobre a potência consumida no controle é muito menor do que o efeito sobre a potência consumida no buffer. A Figura 2 ilustra este efeito em um buffer com profundidade de 16 palavras e na lógica de controle de um roteador Hermes com largura de *flit* igual a 8 bits. O gráfico mostra a potência média (em mW) consumida em função da quantidade de transições de sinais em um pacote com 128 *flits* (100% = 127 transições em cada um dos 8 fios do canal de comunicação). Na figura é possível observar que o consumo de potência aumenta linearmente com o aumento das transições.

Em arquiteturas regulares baseadas em tiles, as dimensões do tile são próximas das dimensões médias dos núcleos. Além disso, as portas de entrada e saída do núcleo são posicionadas próximas à porta local do roteador (assumindo que em cada tile há um núcleo local, e não um subsistema com mais núcleos), fazendo com que o canal de comunicação local seja bem menor do que o canal de comunicação entre roteadores. Sendo assim, *APC* é bem menor que *APL*, apresentando um consumo médio de potência insignificante, mesmo no pior caso, como mostra a Figura 3. Com base nestes resultados, *APC* pode ser desconsiderado sem causar erros significantes no cálculo da potência dissipada.

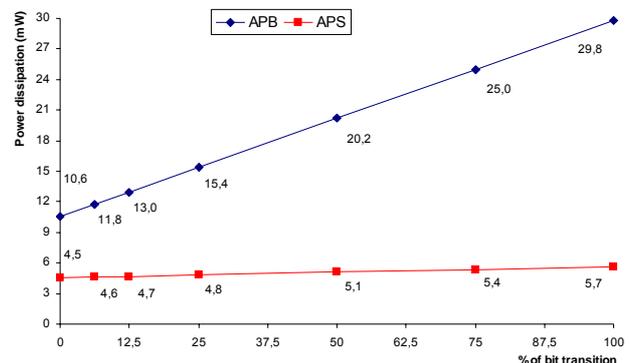


Figura 2: Análise do efeito da transição de sinais sobre a potência média consumida em um buffer de 16 palavras e na lógica de controle de um roteador Hermes.

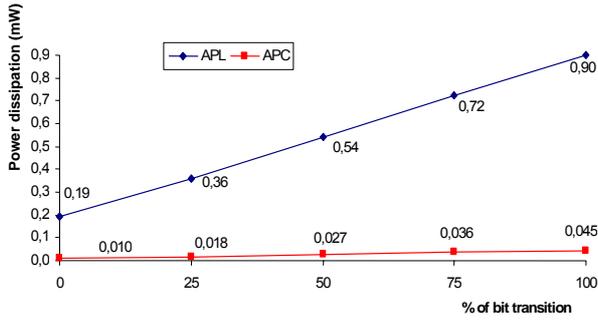


Figura 3: Análise do efeito das transições de sinais sobre a potência média consumida nos canais de comunicação locais e entre roteadores. Cada tile tem dimensões de 5 mm x 5 mm e os canais de comunicação possuem largura de flit igual a 8 bits.

A Equação 2 computa a potência média consumida na comunicação entre um roteador i (no *tile* τ_i) e um roteador j (no *tile* τ_j), onde η corresponde ao número de roteadores que o pacote deve percorrer.

$$RRP_{ij} = \eta \times (APB + APS) + (\eta - 1) \times APL \quad (2)$$

Considerando uma abordagem que utilize um esquema de codificação de dados, dois novos parâmetros devem ser adicionados à Equação 2: *APE* e *APD* (potência média consumida nos módulos de codificação e decodificação, respectivamente). O primeiro esquema de codificação analisado neste trabalho foi o Adaptive Encoding. A Figura 4 mostra a análise do efeito das transições de sinais sobre a potência média consumida nos módulos de codificação e decodificação deste esquema de codificação, ambos com largura de *flit* igual a 8 bits. Como se pode observar na figura, o consumo de potência também cresce linearmente nestes módulos, de acordo com o aumento da quantidade de transições de sinais.

Com base nas análises acima descritas, é possível construir macromodelos para cada um dos parâmetros do modelo proposto – *APB*, *APS*, *APE*, *APD* e *APL* – representando a potência média consumida nos diferentes módulos da NoC. Por razões de simplicidade, os macromodelos estimam a potência média consumida em um determinado módulo sem considerar seu estado interno, ou seja, a potência é calculada em função na atividade de transições nos sinais de entrada do módulo. Esta simplificação implica em um erro relativamente pequeno, segundo experimentos conduzidos neste trabalho. Por exemplo, no caso dos módulos *encoder* e *decoder*, esta simplificação implica em um erro de 5% na estimativa de consumo de potência, o que corresponde a $\pm 0,6$ mW.

Observe que todos os módulos (*buffer*, *control*, *encoder* e *decoder*), bem como os canais de comunicação, apresentam um consumo de potência mesmo quando a taxa de transição de sinais é de 0%. Isto acontece, nos módulos, devido aos chaveamentos de sinais internos e atualizações de estado, ativados pelo *clock*. Da mesma forma, nos canais de comunicação,

mesmo quando se transmite somente ‘0’s nos dados do pacote, os sinais de *clock* e de controle de fluxo permanecem chaveando, consumindo potência. Este consumo é referido como P_0 na análise subsequente. O restante da potência consumida cresce linearmente com a taxa de transição (o topo da rampa é referido como R). A Tabela 1 apresenta o macromodelo linear para os componentes da NoC com o esquema *Adaptive Encoding*.

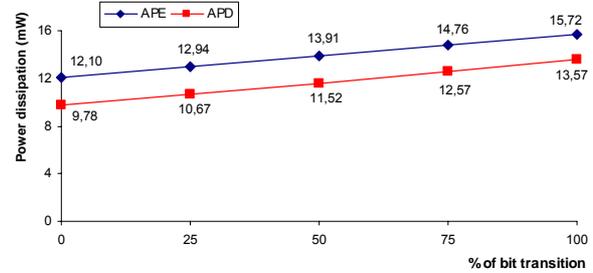


Figura 4: Análise do efeito das transições de sinais sobre a potência média consumida nos módulos de codificação e decodificação do Adaptive Encoding (largura de flit = 8 bits).

Tabela 1: Macromodelo linear para os módulos da NoC com o esquema Adaptive Encoding ($AP = P_0 + \%T * R$).

Módulo	P_0	R
Buffer (<i>APB</i>)	10,61	19,19
Control (<i>APS</i>)	4,39	0,72
Encoder (<i>APE</i>)	12,1	3,62
Decoder (<i>APD</i>)	9,78	3,79
Canal entre roteadores (<i>APL</i>)	0,19	0,71

Considerando uma NoC com o esquema Bus-Invert, o consumo de potência deve ser calculado com base em um novo macromodelo, o qual leva em consideração o bit extra de controle em todos os módulos (ver Tabela 2).

Tabela 2: Macromodelo linear para os módulos da NoC com o esquema Bus-Invert ($AP = P_0 + \%T * R$).

Módulo	P_0	R
Buffer (<i>APB</i>)	11,49	22,13
Control (<i>APS</i>)	4,39	0,98
Encoder (<i>APE</i>)	1,16	3,88
Decoder (<i>APD</i>)	0,55	0,25
Canal entre roteadores (<i>APL</i>)	0,19	0,8

A Equação 3 computa a potência média consumida na comunicação entre o núcleo local do roteador localizado no *tile* τ_i e o núcleo local do roteador localizado no *tile* τ_j , passando por η roteadores e utilizando o esquema de codificação de dados.

$$CodedRRP_{ij} = APE + \eta \times (APB + APS) + (\eta - 1) \times APL + APD \quad (3)$$

5. ANÁLISE DO CONSUMO DE POTÊNCIA

A análise da potência média consumida nos módulos foi feita com diferentes padrões de tráfego. Alguns destes tráfegos são sintéticos, ou seja, gerados por uma ferramenta de geração de tráfego, enquanto que outros são dados de aplicações reais inseridas em pacotes e transmitidos através da NoC.

A fim de avaliar completamente o efeito do tráfego

resultante de uma aplicação real, os experimentos devem ser executados com quantidades realistas de dados, com até centenas de pacotes. Entretanto, o tempo de simulação para estas quantidades de pacotes em um simulador SPICE é impraticável. Sendo assim, uma melhor alternativa foi tomada, explorando a possibilidade de inserir os macromodelos em um modelo com mais alto nível de abstração, o qual foi simulado com o ambiente *Ptolemy II* [9]. Este, por sua vez, inclui um modelo do codificador, dos componentes da NoC e do decodificador. Estes modelos são utilizados para calcular a percentagem de transições de sinais em cada padrão de tráfego e, com base nos macromodelos de potência dos módulos, estimar a potência média consumida na comunicação, de acordo com o tráfego.

6. RESULTADOS PRELIMINARES

Esta seção apresenta os resultados preliminares obtidos através de simulações em nível de sistema executadas no ambiente *Ptolemy II*, utilizando os macromodelos descritos na seção anterior. Para isto, o sistema compreendendo o modelo da NoC e do esquema de codificação foi simulado com diferentes tipos de padrão de tráfego, a maioria deles tráfego de aplicações reais, e também com tráfegos gerados sinteticamente.

A Tabela 3 mostra os resultados obtidos com o esquema *Adaptive Encoding*. A primeira coluna descreve o tipo de tráfego utilizado. A segunda coluna reproduz a redução na atividade de transição de bit relatado por Benini et al. em [6] (quando disponível), enquanto que a terceira coluna mostra a redução na atividade de transição de bit obtidos nos experimentos executados no escopo deste trabalho, reproduzindo o esquema de codificação proposto em [6]. Ambos os casos apresentam os resultados em termos de redução na quantidade de transições do tráfego codificado com respeito ao tráfego original. A quarta coluna mostra a média de potência consumida (*APH*) sem o uso de técnicas de codificação (tráfego original). A quinta coluna mostra a mesma medida (*APH*) com o uso do *Adaptive Encoding*. Por fim, a sexta coluna mostra o consumo extra de potência devido aos módulos de codificação e decodificação. A sétima e última coluna diz respeito à quantidade de *hops* necessários para amortizar o custo extra da codificação.

Os resultados mostrados na segunda e terceira colunas apresentam algumas diferenças entre os resultados obtidos em [6] e no presente trabalho. Isto pode ser causado pela relação entre *flits* consecutivos enviados através da NoC. Em [6] os autores utilizaram um canal de comunicação de 32 bits (*flit* de 32 bits). Desta forma, a atividade de transição de bit é computada entre os bytes de mesma posição de *flits* consecutivos (enviando 4 bytes em cada *flit*). No exemplo apresentado na Tabela 3 foi utilizado um canal de comunicação com largura de *flit* igual a 8 bits. Assim, a atividade de transição de bit é computada entre os bytes consecutivos. Porém, o objetivo aqui não é obter os mesmos resultados obtidos em [6], mas sim avaliar o efeito de diferentes técnicas de codificação em diferentes configurações da NoC.

Experimentos com tráfegos gerados sinteticamente mostram que o uso do *Adaptive Encoding* pode ser vantajoso quando a média de *hops* percorridos por pacotes é maior do que 3. Ou seja, a partir de 3 *hops* o custo da codificação é amortizado. Entretanto, com tráfegos reais como, por exemplo, som no formato “wav”, o custo da codificação só é amortizado após 11 *hops* (redução de 2,25 mW/roteador). Em outros casos, o esquema de codificação aumenta a atividade de transição, aumentando o consumo de potência no sistema, sendo desvantajoso.

A Tabela 4 apresenta os resultados experimentais obtidos com o esquema *Bus-Invert*. A terceira coluna é calculada com base no macromodelo da NoC original, sem esquemas de codificação, enquanto que a quarta coluna é calculada com base no macromodelo utilizando *Bus-Invert* (com 1 bit a mais em todos os módulos). Esta tabela mostra que o consumo de potência médio da NoC aumentou com a utilização do esquema *Bus-Invert* submetido a tipos de tráfego real, mesmo com a redução da atividade de transição de bit. A única exceção foi o tráfego PDF. Isto ocorre devido à inclusão de 1 bit a mais em todos os módulos da NoC, o que aumentou o consumo de potência nos mesmos. Além disso, o esquema foi bastante eficiente com um dos tráfegos gerados sinteticamente.

É importante salientar que estes resultados dizem respeito somente à configuração da NoC simulada até o momento, e para tecnologia 0.35 μ .

Tabela 3: Resultados preliminares com Adaptive Encoding.

Stream	Redução na transição de bit (apresentada em [6])	Redução na transição de bit (simulada no escopo deste trabalho)	Potência média consumida na NoC sem codificação (APB+APS+APL)	Potência média consumida na NoC com codificação (APB+APS+APL)	Potência média consumida pelos módulos de codificação (APE + APD)	# of hops
HTML	9,3 %	-1,4 %	22,24 mW	22,34 mW ↑	24,1 mW	-
GZIP	16,3 %	1,03%	25,5 mW	25,4 mW	25,6 mW	240
GCC	15,6 %	0,91 %	24,69 mW	24,6 mW	25,27 mW	292
Bytecode	-	9,3 %	23,45 mW	22,68 mW	24,7 mW	32
WAV	2,1 %	21,95 %	25,5 mW	23,25 mW	25,17 mW	11
MP3	-	- 2,41 %	24,8 mW	25 mW ↑	25,38 mW	-
RAW	- 3,4 %	- 10,98 %	22,24 mW	23 mW ↑	24,5 mW	-
BMP	-	- 0,5 %	25,3 mW	25,36 mW ↑	25,5 mW	-
JPG	-	0,8 %	25,46 mW	25,38 mW	25,5 mW	327
TIFF	-	-1,16 %	25,34 mW	25,46 mW ↑	25,55 mW	-
PDF	-	6,61 %	26,06 mW	25,34 mW	25,65 mW	36
Synthetic 1	-	75 %	30 mW	18,86 mW	25,16 mW	2
Synthetic 2	-	-27 %	21,71 mW	23,5 mW ↑	24,55 mW	-

Tabela 4: Resultados preliminares com Bus-Invert.

Stream	Redução na transição de bit (simulada no escopo deste trabalho)	Potência média consumida na NoC sem codificação (APB+APS+APL)	Potência média consumida na NoC com codificação (APB+APS+APL)	Potência média consumida pelos módulos de codificação (APE + APD)	# of hops
HTML	6,2 %	21,33 mW	22,75 mW ↑	2,93 mW	-
GZIP	18,7%	25,5 mW	25,79 mW ↑	3,75 mW	-
GCC	17,9%	18,46 mW	19,18 mW ↑	2,36 mW	-
Bytecode	12 %	23,45 mW	24,49 mW ↑	3,35 mW	-
WAV	18,8 %	25,52 mW	25,8 mW ↑	3,75 mW	-
MP3	18,48 %	25,3 mW	25,63 mW ↑	3,7 mW	-
RAW	14,6 %	22,24 mW	23 mW ↑	3,1 mW	-
BMP	18,2%	25,3 mW	25,66 mW ↑	3,71 mW	-
JPG	19,5 %	25,46 mW	25,65 mW ↑	3,74 mW	-
TIFF	18,3 %	25,25 mW	25,6 mW ↑	3,7 mW	-
PDF	23,15 %	26 mW	25,76 mW	3,85 mW	13
Synthetic 1	50,7%	30 mW	24,53 mW	4,6 mW	1
Synthetic 2	2,45 %	21,7 mW	23,45 mW ↑	3 mW	-

Em ambos os esquemas utilizados, a redução do consumo de potência nos canais entre roteadores foi muito menor do que na lógica do roteador. Entretanto, em novas tecnologias, o consumo de potência nos canais será mais relevante, devido ao crescimento relativo das capacitâncias nas linhas de comunicação em comparação com a lógica [10]. Neste cenário tais esquemas de codificação podem ser mais vantajosos, já que eles foram desenvolvidos para canais de comunicação.

7. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho investigou a redução do consumo de potência em NoCs através da redução da atividade de transição de sinais utilizando esquemas de codificação de dados. Macromodelos de consumo de potência para os módulos da NoC, bem como para os módulos de codificação e decodificação foram desenvolvidos e embarcados em um modelo de mais alto nível, o qual foi simulado com tipos de tráfego real e com tráfegos gerados sinteticamente.

Os resultados preliminares mostram que a eficiência do esquema de codificação é dependente do padrão de dados transmitido. Experimentos com uma variedade de padrões de tráfego mostram que a redução no consumo de potência com *Adaptive Encoding* vai de -1 a 15 mW/hop. Entretanto, os módulos de codificação e decodificação deste esquema apresentaram um consumo alto de potência. Já os módulos do esquema *Bus-Invert* são econômicos quanto ao consumo de potência, mas não são eficientes em uma NoC com largura de *flit* igual a 8 bits.

Os resultados obtidos até agora apontam a direção para futuras pesquisas endereçando o uso de múltiplos esquemas de codificação para melhorar a eficiência em diferentes padrões de tráfego, bem como o uso de diferentes configurações da NoC, para ajudar na decisão de quando um pacote deve ser ou não codificado. Por exemplo, não é vantajoso codificar pacotes enviados a núcleos vizinhos na NoC. Uma possível abordagem é utilizar um bit de sinalização no cabeçalho do pacote, indicando se ele é ou não codificado.

Os trabalhos futuros incluem a avaliação dos esquemas de codificação em NoC implementadas utilizando tecnologias do estado-da-arte, onde o

consumo de potência nos canais de comunicação se tornam mais relevantes devido ao aumento relativo das capacitâncias nas linhas. Além disso, pretende-se avaliar a utilização de outros esquemas de codificação, tais como Transition e Gray.

Bibliografia

- [1] A. Iyer and D. Marculescu. "Power and performance evaluation of globally asynchronous locally synchronous processors". 29th Annual International Symposium on Computer Architecture (ISCA), pp. 158-168, May 2002.
- [2] W. Dally and B. Towles. "Route packets, not wires: on-chip interconnection networks". Design Automation Conference (DAC), pp. 684-689, June 2001.
- [3] D. Singh, J. M. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal and T. J. Mozdzen. "Power conscious cad tool and methodologies: A perspective". Proc. IEEE, vol.83, pp. 570-594, Apr. 1995.
- [4] J.C. Palma, C.A. Marcon, F. Moraes, N.L. Calazans, R.A. Reis, A.A. Susin. "Mapping Embedded Systems onto NoCs - The Traffic Effect on Dynamic Energy Estimation". In: 18th Symposium on Integrated Circuits and Systems Design - SBCCI 2005. New York: ACM Press, 2005. pp. 196-201.
- [5] F. Moraes, N. Calazans, A. Mello, L. Möller and L. Ost. "HERMES: an infrastructure for low area overhead packet-switching networks on chip". The VLSI Journal Integration (VJI), vol. 38, issue 1, pp. 69-93, October 2004.
- [6] L. Benini, A. Macii, E. Macii, M. Poncino, R. Scarsi. "Architecture and Synthesis Algorithms for Power-Efficient Bus Interfaces". Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 19, Issue 9, Sept. 2000 Page(s):969-980.
- [7] M. R. Stan, W. P. Burleson. "Bus-Invert Coding for Low-Power I/O". VLSI Systems, IEEE Transactions on Volume 3, Issue 1, March 1995 Page(s):49-58.
- [8] L. Ost, A. Mello; J. Palma, F. Moraes, N. Calazans. "MAIA - A Framework for Networks on Chip Generation and Verification". ASP-DAC, Jan. 2005.
- [9] C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng. "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II,)" Technical Memorandum UCB/ERL M05/21, University of California, Berkeley, CA USA 94720, July 15, 2005.
- [10] Sylvester, D.; Chenming Wu; "Analytical modeling and characterization of deep-submicrometer interconnect". Proceedings of the IEEE. Volume 89, Issue 5, May 2001 Page(s):634 - 664.