

Mapping Embedded Systems onto NoCs - The Traffic Effect on Dynamic Energy Estimation

José Carlos S. Palma,
César Augusto M. Marcon,

Universidade Federal do Rio Grande do Sul – UFRGS/PPGC
Av. Bento Gonçalves, 9500 - Prédio 43412 / Bloco IV
CEP: 91501-970 - Porto Alegre - RS – BRAZIL

jcspalma@inf.ufrgs.br,
marcon@inf.ufrgs.br

Fernando G. Moraes,
Ney L. V. Calazans,

Pontifícia Universidade Católica do Rio Grande do Sul – FACIN
Av. Ipiranga, 6681 - Prédio 30 / Bloco 4
90619-900 - Porto Alegre - RS – BRAZIL

moraes@inf.pucrs.br,
calazans@inf.pucrs.br

Ricardo A. L. Reis,
Altamiro A. Susin

Universidade Federal do Rio Grande do Sul – UFRGS/PPGC
Av. Bento Gonçalves, 9500 - Prédio 43412 / Bloco IV
CEP: 91501-970 - Porto Alegre - RS – BRAZIL

reis@inf.ufrgs.br,
susin@eletro.ufrgs.br

ABSTRACT

This work addresses the problem of application mapping in networks-on-chip (NoCs). It explores the importance of characterizing network traffic to effectively predict NoC energy consumption. Traffic is seen as an important factor affecting the problem of mapping applications into NoCs having as goal to minimize the total dynamic energy consumption of a complex system-on-a-chip (SoC). Experiments showed that failing to consider the bit transitions influence on traffic inevitably leads to an energy estimation error. This error is proportional to the amount of bit transitions in transmitted packets. In applications that present a large number of packets exchange, the error is propagated, significantly affecting the mapping results. This paper proposes a high-level application model that captures the traffic effect and uses it to describe the behavior of applications. In order to evaluate the quality of the proposed model, a set of embedded systems were described using both, a previously proposed model (that does not capture the traffic effect), and the model proposed here. Comparing the resulting mappings, those derived from the proposed model showed improvements in energy savings with regard to the other model for all experiments.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – *advanced architectures, algorithms implemented in hardware, VLSI (very large scale integration)*.

General Terms

Design, Performance, Experimentation, Theory.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBCCI'05, September 4–7, 2005, Florianópolis, Brazil.
Copyright 2005 ACM 1-59593-174-0/05/0009...\$5.00.

Keywords

Networks-on-Chip, energy estimation, application mapping, traffic effect.

1. INTRODUCTION

Cores are complex pre-designed and pre-verified hardware modules, which can be considered as key component in the development of system-on-a-chip

New technologies allow the implementation of complex systems-on-chip (SoC) with millions of transistors integrated onto a single chip. These complex systems need special communication resources to handle very tight design requirements. In addition, deep sub-micron effects pose formidable physical design challenges for long wires and global on-chip communication. Many designers propose to change from the fully synchronous design paradigm to globally asynchronous, locally synchronous (GALS) design paradigm [1]. GALS design subdivides the application into sub-applications. Each sub-application is a synchronous design physically placed inside a tile, and the communication between tiles is provided by an asynchronous communication resource. A network-on-chip (NoC) is an infrastructure essentially composed by routers interconnected by communication channels. A NoC is suitable to deal with the GALS paradigm, since it provides asynchronous communication, high scalability, reusability, reliability, and efficient energy consumption [2].

The modeling of dynamic energy consumption for a system that uses a NoC as its communication infrastructure is important to guide both, the mapping of IPs cores in the NoC, and the choice of the dimension of the wires, which interconnect routers.

Consider a SoC implemented using the GALS paradigm, composed by n cores and employing a NoC as communication infrastructure. The application mapping problem for this architecture consists in finding an association of each core to a tile (a mapping) such that some cost function – like latency, throughput and power dissipation – is minimized.

In general, this mapping problem allows $n!$ possible solutions. Given a future SoC with hundreds of tiles [3], it is easy to conclude that an exhaustive search of the problem solutions space will become unviable. Consequently, the search of an optimal implementation for such SoCs requires efficient mapping strategies and representative application models. Some mapping strategies have been proposed. Core graph [4] and application characterization graph (APCG) [5] are instances of models that account for the overall communication volume of each pair of cores. These structures are implementations of a communication weighted model (CWM) [6], since both take into account only the weight of the communication activity between a pair of cores.

To achieve good application core mappings onto target architectures, several authors stress the importance of traffic modeling. However, previous application models, like those presented in [4] and [5], abstract important information that affects the dynamic energy consumption estimation.

When a physical communication wire changes its logic value from 0 to 1 or from 1 to 0 a bit transition occurs. Each bit transition consumes dynamic energy. Experiments on the traffic behavior of some applications, described on Section 3, show that neglecting the bit transition information during the transmission of a single packet may lead to an estimation error of more than 100% in dynamic energy consumption. For instance, considering a 16-word router input buffer implemented with CMOS TSMC 0.35 technology, the difference in dynamic energy consumption between minimum (zero) and maximum values of bit transition (128) for a 128-flit packet is more than 180%. This discards the choice of an average value of bit energy consumption as sound. Consequently, the effect of omitting the amount of bit transitions onto a NoC mapping, i.e. considering only the bit volume, will certainly lead to data poorly correlated to reality to be used for mapping estimation. To overcome this problem, this paper proposes an extended communication weighted model (ECWM), which captures both, the volume of communication and the bit transition rate in each communication channel. Comparing the mapping quality of applications modeled with ECWM versus CWM, we achieve for all experiments improvement in dynamic energy consumption savings.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents the dynamic energy consumption model for Networks-on-chip, justifying the proposed model. Section 4 defines the target architecture model and the application models. Section 5 shows how application models are applied over target architecture models to compute dynamic energy consumption. Section 6 presents experimental results comparing distinct model mappings. Finally, Section 7 presents some conclusions.

2. RELATED WORK

Ye, Benini and De Micheli [7] introduced a framework to estimate the energy consumption in a communication infrastructure considering routers, internal buffers, and interconnect wires. Inside the framework, they implemented a simulation platform to trace the dynamic energy consumption with bit-level accuracy. The simulation of NoCs under different traffic enabled them to propose a power dissipation model for NoC components. This model can be applied to the architectural exploration for low power high performance NoC designs. Similar

power dissipation models are presented in [4][5][6][8] as well as in the present work.

Hu and Marculescu [4] showed that by using mapping algorithms it is possible to reduce energy consumption by more than 60% when compared to random mapping solutions. The authors proposed the use of an *application characterization graph* (APCG), a way of capturing the application core communication by a CWM. Murali and De Micheli [5] proposed a solution similar to that in [4]. A structure called *core graph* represents the underlying CWM. The main goal of that work was to propose an algorithm to map cores on mesh NoCs under bandwidth constraints while minimizing average communication delay.

Marcon et al. [6] proposed a communication dependence model (CDM), which represents application cores describing both the dependence among messages exchanged in the network and the amount of bits transmitted between pairs of cores. They show that, compared to CWM, CDM allows obtaining mappings with 42% average reduction in the execution time, together with a 21% average reduction in the total energy consumption for state-of-the-art technologies. In [8], the same authors propose the communication dependence and computation model (CDCM), which is an improvement of CDM. However, for both models, to capture message dependence from an application is a hard, error prone and not easily automated task. The present work proposes another model that can be easily extracted by simulation, as occurs with CWM. In addition, this model improves CWM by the capture of bit transition quantities.

Ye et al. [9] analyzed different routing schemes for packetized on-chip communication on a mesh NoC architecture, describing the contention problem and the consequent performance reduction. In addition, they evaluate the packet energy consumption using the same energy model proposed in [4] and [5], extending it to the analysis of packet transmission phenomena.

Wang, Peh and Malik [10] showed the impact of the technology on networks topologies and provided the basis for predicting the most energy-efficient topology for future technologies based on this impact. They used average flit traversal energy as metric to evaluate network energy efficiency.

Peh and Eisley [11] proposed a framework for network energy consumption analysis that uses link utilization as the unit of abstraction for network utilization and energy consumption, capturing energy variations both spatially, across the network fabric, and temporally, across application execution time.

To the best of our knowledge, no energy consumption model for application cores takes into account the bit transition amount of the inter-core traffic. This work shows the importance of this communication aspect, since abstracting bit transition considerations may lead to errors in power dissipation estimation of more than 100%.

3. DYNAMIC ENERGY CONSUMPTION MODEL

Energy consumption originates from both IP cores operation and interconnection components between these cores. For most current CMOS technologies, static energy accounts for the smallest part of the overall consumption. Thus, this work focuses on NoC dynamic energy consumption only, using it as an

objective function to evaluate the quality of application cores mapping onto mesh NoC architectures.

Dynamic energy consumption is proportional to switching activity, arising from packets moving across the NoC. Interconnect wires and routers dissipate dynamic power. Several authors [4][5][6][7][8][9] have proposed to estimate NoC energy consumption by evaluating the effect of bits/packets traffic on each NoC component. This work is no exception. It evaluates the dynamic energy consumption for regular mesh NoCs.

Bit energy $EBit$ is used to estimate the dynamic energy consumption of each bit, when it flips polarity from a previous value. $EBit$ can be split into three components: bit dynamic energy ($ERbit$), consumed by a router comprised by buffers, router wires and logic gates for switching; bit dynamic energy ($ELbit$), consumed on a link between tiles; and bit dynamic energy ($ECbit$), consumed on a link between the router and the core of the tile. The relationship between these quantities is expressed by Equation (1), which gives the dynamic energy consumption of a bit passing through a router, a local link and a link between tiles.

$$(1) EBit = ERbit + ELbit + ECbit$$

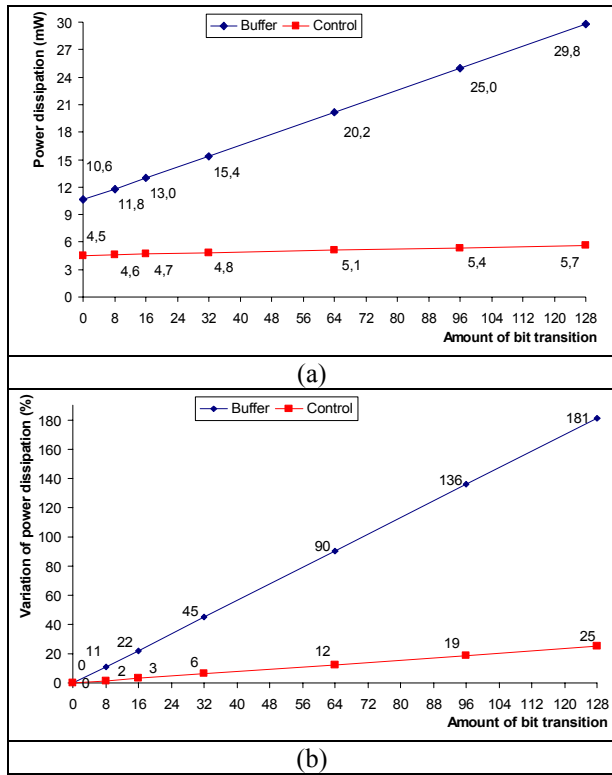


Figure 1 – Analysis of the bit transition effect on dynamic energy consumption for a Hermes router [12] with 16-word input buffers and centralized control logic. Data obtained from SPICE simulation (CMOS TSMC 0.35 technology). Results are for a single buffer. Percentage results are computed w.r.t. to the respective zero bit transition dissipation values.

The analysis of the traffic effect reveals the need to split the computation of router dynamic energy consumption ($ERbit$) into buffer ($EBbit$) and control portions ($ESbit$). This arises because the

bit transition effect on energy consumption of the router control is much smaller than its effect on the energy consumption of the router buffer. Figure 1 illustrates this effect for a Hermes router [12] with a 16-word input buffers and centralized control logic. The graph depicts power as a function of the amount of bit transitions in a 128-bit packet. Clearly, energy consumption increases linearly and is proportional to the amount of bit transitions in a packet.

In regular tile-based architectures, tile dimension is close to the average core dimension, and the core inputs/outputs are placed near the router local channel. Therefore, $ECbit$ is much smaller than $ELbit$. Figure 2(a) corroborates this by comparing energy consumption for local and inter-tile links. A twenty-fold difference in energy consumption magnitude between $ELbit$ and $ECbit$ appears. It happens because a physical link is equivalent to a large RC circuit if compared to local links.

Figure 2 (b) depicts the same data of Figure 2 (a), but emphasizes the effect of the amount of bit transitions in the dynamic power dissipation. Minimum and maximum bit transitions values generate power dissipation values that differ by a factor of 62.

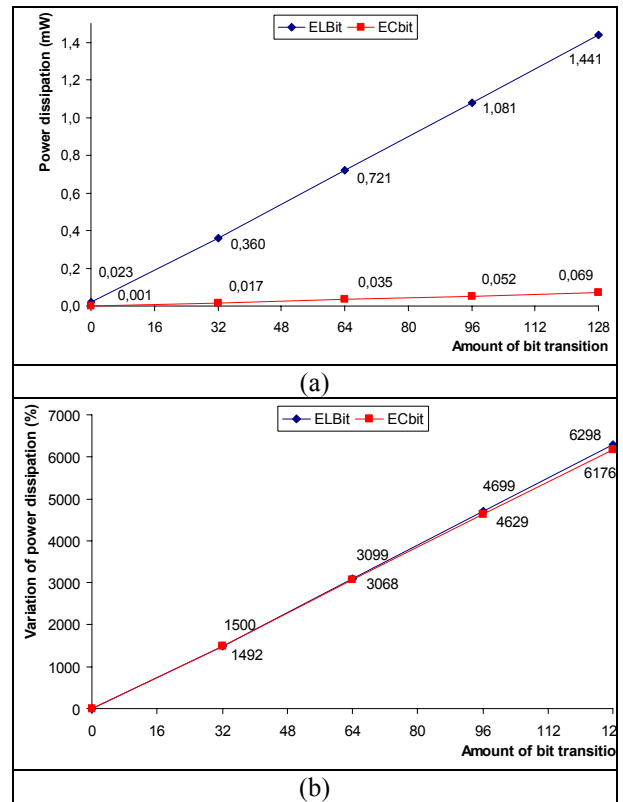


Figure 2 – Analysis of the effect of bit transitions on dynamic energy consumption of local and inter-router links. Data obtained from SPICE simulation for a NoC Hermes [12] (CMOS TSMC 0.35 technology). Each tile has 5 mm x 5 mm of dimension, and uses 16-bit links. Percentage results are computed w.r.t. to the respective zero bit transition dissipation values.

Considering these results, $ECbit$ may be safely neglected without significant errors in total energy dissipation. Therefore, Equation

(2) computes the dynamic energy consumed by a single bit traversing the NoC, from tile τ_i to tile τ_j , where η corresponds to the number of routers through which the bit passes.

$$(2) EBit_{ij} = \eta \times (EBbit + ESbit) + (\eta - 1) \times ELbit$$

3.1 Model Parameters Acquisition

To acquire the above energy parameters ($EBbit$, $ESbit$, $ELbit$ and $ECbit$), it suffices to evaluate the dynamic energy consumption of a communication infrastructure with different traffic patterns. For the Hermes NoC communication infrastructure, the basic element is a router with five bi-directional channels connecting to four other routers and to a local IP core. The router employs an XY routing algorithm, and uses input buffering only. The conducted experiments employ a mesh topology version of Hermes with six different configurations. These are obtained by varying flit width (either 8 or 16 bits), and input buffers depth (4, 8 and 16 flits). For each configuration, 128-flit packets enter the NoC, each with a distinct pattern of bit transitions in their structure, from 0 to 128.

The flow for obtaining dynamic energy consumption data comprises three stages. The first stage starts with the NoC VHDL description and traffic files, both obtained using a customized environment for NoC/NoC traffic generation. Traffic input files enable to exercise the NoC through the router local channels, modeling local cores behavior. A VHDL simulator applies input signals lists to the NoC or for any NoC module, either a single router or a router inner module (input buffer or control logic). Simulation produces signal lists storing the logic values variations for each signal. These lists are converted to electric stimuli and used in SPICE simulation (in the third stage).

In the second stage, the module to be evaluated (e.g. an input buffer) is synthesized using a technology cell library, such as CMOS TSMC 0.35. Synthesis gives an HDL netlist, later converted to a SPICE netlist using a converter developed in the scope of this work.

The third stage consists in the SPICE simulation of the module under analysis. Here, it is necessary to integrate both, the SPICE netlist of the module, the electrical input signals and a library with logic gates described in SPICE. The resulting electric information allows the acquisition of NoC energy consumption parameters for a given traffic.

4. APPLICATION CORES AND NOC MODELS

Section 3 proposed to use $ELbit$, $EBbit$ and $ESbit$ as key components to represent bit dynamic energy consumption on a regular NoC with mesh topology. The analysis of the traffic effect on dynamic energy consumption shows that these three components depend on the amount of bit traffic [4][8]. The amount of bit transitions, on the other hand, affects mostly $ELbit$ and $EBbit$ and has small influence on $ESbit$. In addition, the effect of bit transitions on $EBbit$ and on $ESbit$ has a magnitude that is comparable to the effect obtained by varying the amount of bit traffic as described for example in [4] and [8]. Finally, $ELbit$ is basically influenced by bit transitions only. This analysis shows the importance of proposing a model considering both the amount of bits and the amount of bit transition for modeling communication using NoCs.

This section defines CWM, a model that captures only the amount of bits and proposes EWCM, an improvement of CWM that also captures the amount of bit transitions in communications. These models underlie the structures that enable to represent them (CWG and ECWG), as explained next.

Definition 1: A *communication weighted graph* (CWG) is a directed graph $\langle C, W \rangle$. The set of vertices $C = \{c_1, c_2, \dots, c_n\}$ represents the set of application cores. Assuming w_{ab} is the number of bits of all packets sent from core a to core b , $W = \{(c_a, c_b, w_{ab}) \mid c_a, c_b \in C \text{ and } w_{ab} \in \mathbb{N}^*\}$. The set of edges W represents all communications between application cores.

Definition 2: An *extended communication weighted graph* (ECWG) is a directed graph $\langle C, T \rangle$. The set of vertices $C = \{c_1, c_2, \dots, c_n\}$ represents the set of application cores. Assuming w_{ab} is the number of bits of all packets sent from core a to core b and that t_{ab} is the number of bit transitions occurred on all packets sent from core c_a to core c_b , the set of edges T is $\{(c_a, c_b, w_{ab}, t_{ab}) \mid c_a, c_b \in C, w_{ab} \in \mathbb{N}^* \text{ and } t_{ab} \in \mathbb{N}\}$. The set of edges T represents all communications between these cores, representing both, the amount of bits and the amount of bit transitions.

ECWG is very similar in structure to CWG. However, ECWG improves CWG, since it captures the number of bit transitions instead of only the number of bits transmitted from one core to another. While CWM and ECWM model application cores communication, the NoC is modeled by a graph that represents its physical components, i.e. routers and links. This graph is called CRG and its definition is stated below.

Definition 3: A *communication resource graph* is a directed graph $CRG = \langle R, L \rangle$, where the vertices set is the set of routers $R = \{r_1, r_2, \dots, r_n\}$, and the edge set $L = \{(r_i, r_j), \forall r_i, r \in R\}$ is the set of paths from router r_i to router r_j .

The value n is the total number of routers and is equal to the product of the two NoC dimensions. CRG edges and vertices represent physical links and routers, respectively, and each router is connected to an application core.

CWG and ECWG represent the communication of an application composed by an arbitrary number of cores. These graphs are evaluated here on a mesh topology NoC using wormhole and deterministic XY routing algorithm. Nevertheless, other NoC topologies can be equally treated, just changing the CRG formulation.

Figure 3 illustrates the above definitions using a hypothetical application with 4 IP cores exchanging, 6 packets and a 2x2 NoC.

Figure 3(a) shows a CWG where the set of vertices is $C = \{A, B, E, F\}$, and the set of edges is $W = \{(A, B, 80), (A, E, 90), (A, F, 100), (B, A, 100), (B, E, 120), (B, F, 80), (E, A, 80), (E, B, 70), (E, F, 90), (F, A, 60), (F, B, 50), (F, E, 90)\}$. Figure 3(b) depicts an ECWG for the same hypothetical application. This graph has the same set of vertices. However, each edge also contains the amount of bit transitions of the communication. The set of edge is $T = \{(A, B, 80, 40), (A, E, 90, 55), (A, F, 100, 100), (B, A, 100, 30), (B, E, 120, 80), (B, F, 80, 25), (E, A, 80, 75), (E, B, 70, 40), (E, F, 90, 35), (F, A, 60, 55), (F, B, 50, 25), (F, E, 90, 85)\}$. Figure 3(c) depicts an arbitrary mapping of C onto the NoC, corresponding to a CRG

where the set of vertices is $R = \{r_1, r_2, r_3, r_4\}$, and the set of edges is $L = \{(r_1, B), (r_2, F), (r_3, E), (r_4, A)\}$.

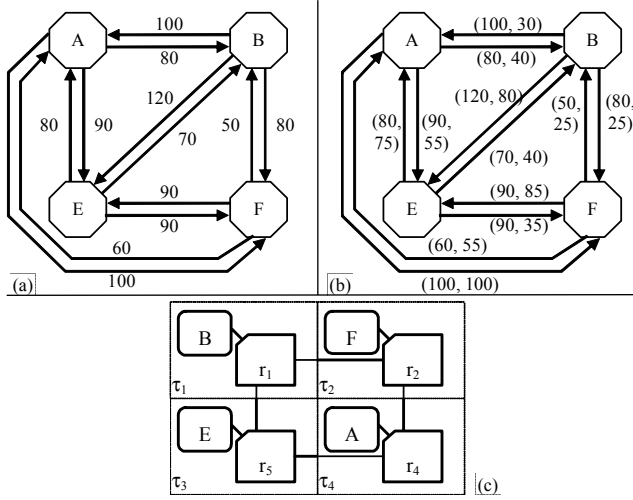


Figure 3 – (a) CWG; (b) ECWG; (c) Core mapping onto a NoC mesh 2x2.

5. ENERGY CONSUMPTION WITH APPLICATION CORES MODELS

As stated in Section 3, the dynamic energy estimation depends on the communication infrastructure (here assumed to be a NoC) and on the application core traffic (modeled by CWM and ECWM). This Section shows how to compute the dynamic energy consumption in a NoC for both models.

Let τ_i and τ_j be the tiles to which cores c_a and c_b , are respectively mapped, and w_{ab} be the amount of bits transmitted from core c_a to core c_b . Then, CWM computes the dynamic energy consumed on this communication by Equation (3).

$$(3) \ ECommunication_{ab} = w_{ab} \times EBit_{ij}$$

The same $ECommunication_{ab}$ is differently computed on ECWM, since $ELbit$, $EBbit$ and $ESbit$ have different values for the amount of bit and for the amount of bit transitions. Let 1 be an index representing $EBit_{ij}$, which regards only the amount of bits ($EBit_{ij1}$) and let 2 be an index representing $EBit_{ij}$ that considers only the amount of bit transitions ($EBit_{ij2}$). Equation (4) relates these amounts and Equation (5) expands Equation (4).

$$(4) \ ECommunication_{ab} = w_{ab} \times EBit_{ij1} + t_{ab} \times EBit_{ij2}$$

$$(5) \ ECommunication_{ab} = \eta \times (w_{ab} \times (EBbit_1 + ESbit_1) + t_{ab} \times (EBbit_2 + ESbit_2)) + (\eta - 1) \times (w_{ab} \times ELbit_1 + t_{ab} \times ELbit_2)$$

For both models, Equation (6) gives the total amount of NoC dynamic energy consumption ($EDyNoC$), computing this for all communications between application cores. Let D be the set of edges in the model graph, i.e. either W for CWG or T for ECWG. Then, $EDyNoC$ represents the objective function for NoC mapping problem with CWM and ECWM models.

$$(6) \ EDyNoC = \sum_{i \in D} ECommunication_{ab}(i)$$

6. EXPERIMENTAL RESULTS

For CWM and ECWM, this work implements similar algorithms that mix simulated annealing and simulated evolution approaches. The main difference consists in the mapping objective function used for each model.

This section presents experimental results of estimating dynamic energy consumption for 11 applications. There are 5 embedded applications and 6 random applications generated by a proprietary system similar to TGFF [16]. Table 1 summarizes applications features and required NoC size.

Table 1 – Application features. Embedded applications are Video Object Plane Decoder (V) [13], MPEG4 decoder (M) [13], Fast Fourier Transform (F) [14], distributed Romberg integration (R) [15], and object recognition and image encoding (O).

Application	NoC size	Number of cores	Amount of bits (M bits)	Amount of bits transition (M bits)	
Embedded	3 x 4 (V)	12	4,268	815	
	4 x 5 (M)	17	3,780	720	
	6 x 6 (F)	33	343	170	
	7 x 7 (R)	49	219	175	
	8 x 8 (O)	64	65,555	20,934	
Random	5 x 5	22	120	0	120
	7 x 9	60	450	0	450
	8 x 8	62	2,390	0	2,390
	10 x 8	77	3,456	0	3,456
	10 x 11	107	567,77	0	567,77
	10 x 12	115	23,432	0	23,432

The NoC size is the number of CRG vertices and the number of cores corresponds to the number of CWG or ECWG vertices. The total amount of bits column reflects the number of bits transmitted during application execution, and is used on both models, while the total amount of bits transition column is used only on the ECWM mapping algorithm. This last column represents typical values of bit transitions for each embedded application, which can be easily extracted from functional simulation. For random applications the column represents minimum and maximum limits for bit transitions.

Table 2 – Dynamic energy consumption of embedded applications with mappings obtained with CWM and ECWM mappings algorithms.

NoC size	CWM (mJ)	ECWM (mJ)	CWM / ECWM (%)
3 x 4	2.47	2.09	18.18
4 x 5	2.53	2.23	13.45
6 x 6	0.65	0.63	3.17
7 x 7	0.33	0.25	32.00
8 x 8	35.98	31.40	14.59
Average	8.39	7.32	16.28

For each application, the best mapping achieved with the CWM algorithm is compared to the best mapping achieved with the ECWM algorithm. As CWM does not consider the bit transition effect, to minimize the error of using this model this work proposes to employ the average consumption of bit transition to compute the values for bit energy parameters, i.e. $Ebit$ values

were estimated according the average case. Even with this measure, the CWM mapping algorithm still does not lead to best mappings competitive with the results of the ECWM algorithm. Table 2 and Table 3 compare the results for both algorithms.

Table 3 – Dynamic energy consumption of hypothetical applications with mapping obtained with CWM and ECWM mappings algorithms.

NoC size	CWM (mJ)	minimum bit transition		maximum bit transition	
		ECWM (mJ)	CWM / ECWM (%)	ECWM (mJ)	CWM / ECWM (%)
5 x 5	0.47	0.35	33.33	0.34	38.89
7 x 9	0.76	0.52	44.93	0.53	42.86
8 x 8	2.22	1.49	49.25	1.40	58.73
10 x 8	2.36	1.70	38.89	1.77	33.33
10 x 11	275.10	178.82	53.85	184.32	49.25
10 x 12	13.11	8.26	58.73	9.05	44.93
Average	49.00	31.86	46.50	32.9	44.67

Table 2 and Table 3 show an improvement of 16% and 45.6% on dynamic energy savings, respectively, when comparing ECWM and CWM mappings. Random applications differ more than embedded ones. This is because for random applications minimum or maximum bit transitions amount are used to generate ECWM mappings that compare with CWM mappings produced for an average bit transitions amount. The objective here is not obtaining precise estimations, but to show how the bit transition effect can influence mapping results.

7. CONCLUSIONS

This paper addresses the problem of mapping applications onto Networks-on-chip and emphasizes the importance of traffic modeling on dynamic energy consumption estimation.

The first contribution is the dynamic energy consumption analysis with different traffic patterns and its effect in different NoC modules, i.e. router input buffer, router control logic and links. The analysis shows the importance of bit transitions and the net amount of bits transmitted between application cores to solving the mapping problem. Often, this problem aims at minimizing dynamic energy consumption in the communication infrastructure. Dynamic energy consumption grows linearly with the amount of bit transitions. Bit transitions affect the dynamic energy consumption by as much as 6400% for links, 180% for router input buffers and 20% for router control logic.

The second contribution is a model that contemplates the amount of bits and its transitions. Experiments conducted showed that ECWM obtains energy consumption savings when compared to CWM in all cases.

Data to build CWM and ECWM are easily extracted from simulation, even for large systems. In addition, the experiments show that ECWM is more accurate for dynamic energy consumption estimation with low extra computational effort when compared to CWM.

8. REFERENCES

[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] S. Kumar et al. A network on chip architecture and design methodology. IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 105-112, April 2002.

[4] J. Hu and R. Marculescu. Energy-aware mapping for tile-based NoC architectures under performance constraints. Asia and South Pacific-Design Automation Conference (ASP-DAC), pp. 233-239, Jan 2003.

[5] S. Murali and G. De Micheli. Bandwidth-constrained mapping of cores onto NoC architectures. Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 896-901, February 2004.

[6] C. Marcon, A. Borin, A. Susin, L. Carro and F. Wagner. Time and Energy Efficient Mapping of Embedded Applications onto NoCs. Asia and South Pacific-Design Automation Conference (ASP-DAC), January 2005.

[7] T. Ye; L. Benini and G. De Micheli. Analysis of power consumption on switch fabrics in network routers. Design Automation Conference (DAC), pp.524-529, June 2002.

[8] C. Marcon; N. Calazans, F. Moraes; A. Susin L. Reis and F. Hessel. Exploring NoC Mapping Strategies: An Energy and Timing Aware Technique. Design, Automation and Test in Europe (DATE), pp. 502-507, March 2005.

[9] T. Ye; L. Benini and G. De Micheli. Packetization and routing analysis of on-chip multiprocessor networks. Journal of Systems Architecture (JSA), vol. 50, issues 2-3, pp. 81-104, February 2004.

[10] H. Wang; L. Peh; S. Malik. A Technology-aware and Energy-oriented Topology Exploration for On-chip Networks. Design, Automation and Test in Europe (DATE), pp. 1238-1243, March 2005.

[11] N. Easley; L. Peh. High-Level Power Analysis of On-Chip Networks. International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), September 2004.

[12] 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.

[13] E. Van der Tol and E. Jaspers. Mapping of MPEG-4 Decoding on a Flexible Architecture Platform. SPIE Conference on Visualization and Data Analysis. pp. 1-13, January 2002.

[14] M. Quinn. Parallel Computing- Theory and Practice, McGraw-Hill, New-York, 1994.

[15] R. Burden and J. D. Faires. Study Guide for Numerical Analysis, McGraw-Hill, New-York, 2001.

[16] R. Dick, D. Rhodes and W. Wolf. TGFF: task graphs for free. International Workshop on Hardware / Software Co-Design (CODES/CASHE), pp.97-101, March 1998.