Modeling the Traffic Effect for the Application Cores Mapping Problem onto NoCs

César A. M. Marcon, José C. S. Palma,
Altamiro A. Susin, Ricardo A. L. Reis

*PPGC - II - UFRGS - Av. Bento Gonçalves, 9500,
Porto Alegre, RS – Brazil*

{marcon, jcspalma, susin, reis}@inf.ufrgs.br

Ney L. V. Calazans,
Fernando G. Moraes

*PPGCC - FACIN – PUCRS - Av. Ipiranga, 6681,
Porto Alegre, RS – Brazil*

{calazans, moraes}@inf.pucrs.br

Abstract

This work addresses the problem of application mapping in networks-on-chip (NoCs) having as goal to minimize the total dynamic energy consumption of a complex system-on-a-chip (SoC). It explores the importance of characterizing network traffic to predict NoC energy consumption and of evaluating the error generated when the bit transitions influence on traffic is neglected. This error is proportional to the amount of bit transitions in transmitted packets. The paper proposes a high-level application model that captures the traffic effect. In order to evaluate the quality of the proposed model, a set of real and random applications were described using both, a previously proposed model (that does not capture the traffic effect), and the model proposed here. Each high-level application model was implemented inside a framework that enables the description of different applications and NoC topologies description. The goal of this environment is to achieve mappings that reduce some NoC cost function. Comparing the resulting mappings, those derived from the model proposed here showed an average improvement of 45% in energy saving with regard to the other model.

1. Introduction

New technologies allow the implementation of complex systems-on-chip (SoC) with hundreds of millions transistors integrated onto a single chip. These complex systems need special communication resources to cope with very tight design requirements. A NoC is suitable to deal with such requirements, since it provides high scalability, reusability and reliability [1].

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.

Assuming there are ***n*** equally-sized areas to where any of the ***n*** cores can be assigned, this mapping problem allows ***n!*** possible solutions. The cost of using exhaustive search algorithms to solve the mapping problem is obviously prohibitive for even moderately sized (e.g. 4x4 2D meshes) NoCs. Consequently, the search of an optimal implementation for such SoCs requires more efficient mapping strategies and sound application models. Some mapping strategies have been proposed. *Core graphs* [2] and *application characterization graphs* (APCGs) [3] are instances of a same generic model, called *communication weighted model* (CWM) [4], since both take into account only the amount of communication exchanged between pairs of cores. This kind of model abstracts important traffic information that affects dynamic energy consumption estimation, e.g. the separation between amount of bits and amount of bit transitions on communications.

When a physical wire changes its logic value from 0 to 1 or from 1 to 0 a *bit transition* occurs. Each bit transition consumes dynamic energy. However, traffic without bit transitions also affects the dynamic energy consumption. Experiments on the traffic behavior of some applications show that neglecting information on bit transition versus amount of bits during the transmission of a single packet may lead to an estimation error of more than 100% in dynamic energy consumption (see Section 3). For instance, considering a 16-word NoC router input buffer implemented with CMOS TSMC 0.35µm technology, the difference in dynamic energy consumption between minimum (zero) and maximum values of bit transitions (127) for a 128-flit packet is more than 180%. This discards the choice of an average value of bit energy consumption or to use only bit transition information as sound. Consequently, the effect of omitting the amount of bit transitions or the bit volume onto a NoC mapping 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, all conducted experiments showed improvement in dynamic energy consumption savings.

In the rest of this paper, Section 2 discusses related work, while Section 3 presents the dynamic energy consumption model for NoC, justifying its proposition.

Next, 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 the tools used to conduct the experiments and the associated results comparing distinct model mappings. Finally, Section 7 presents some conclusions.

2. Related Work

Ye, Benini and De Micheli [5] 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, which is applied to architectural exploration. Similar power dissipation models are presented in [2][3][4][6] and here.

Hu and Marculescu [2] 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 a kind of CWM model that captures the application core communication. Murali and De Micheli [3] proposed a similar solution, where the main contribution is an algorithm to map cores on 2D mesh NoC architectures with bandwidth constraints minimizing average communication delay.

Marcon et al. [4] proposed a communication dependence model (CDM), which represents application cores describing both the dependence among messages and the amount of bits transmitted in each message. 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 [6], the same group 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. [7] 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 [2] and [3], extending it to the analysis of packet transmission phenomena.

Peh and Eisley [8] 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 knowledge of the authors, no model of energy consumption for application cores takes into account the bit transition effect of the inter-core traffic. This work shows the importance of this communication aspect, since abstracting bit transition considerations may lead to significant error in power dissipation estimation.

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 2D mesh NoC architectures.

Dynamic energy consumption is proportional to switching activity, and arises from packets moving across the NoC. Interconnect wires and routers dissipate dynamic power. Several authors [2][3][4][5][6][7] have proposed to estimate NoC energy consumption by evaluating the effect of bits/packets traffic on each component of the infrastructure. This work is no exception. It evaluates the dynamic energy consumption for regular 2D mesh NoCs.

Bit energy $EBit$ is used to estimate the dynamic energy consumption of each bit, when it flips polarity. $EBit$ can be split into four components: bit dynamic energy ($EBbit$) consumed into a router buffer; bit dynamic energy ($ESbit$) consumed into a router control comprised by router wires and logic gates; 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.

$$EBit = EBbit + ESbit + ELbit + ECbit \tag{1}$$

This Section evaluates the above parameters effect on dynamic energy consumption, with data obtained from SPICE simulation of the Hermes NoC [9] synthesized with CMOS TSMC 0.35μm technology.

Depending on the NoC technology implementation, the bit transition effect on energy consumption of the router control is too much different than its effect when considering router buffers. For instance, Figure 1 illustrates this effect in different sizes of Hermes router buffers with 8-bit flit width and centralized control logic. The graph depicts power dissipation of router buffers and router control as a function of the amount of bit transitions in a 128-bit packet. Energy consumption increases linearly and is proportional to the amount of bit transitions in a packet. However, the bit transition effect on energy consumption varies differently comparing buffers and control circuits. For instance, from 0% to 100% of bit transition, the energy consumption increases 181% on buffer against 25% on control circuits.

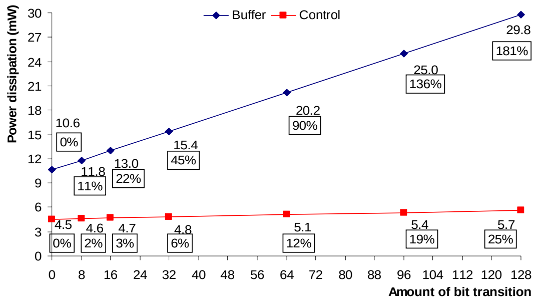


Figure 1 – Bit transition effect on dynamic energy consumption of buffers. Traffic is composed by 128-bit packet varying from 0% to 100% of transitions.

Figure 2 shows the same effect in Hermes control logic with 8-bit and 16-bit flit width. As we can observe here the amount of bit transition has low influence over the control logic energy consumption.

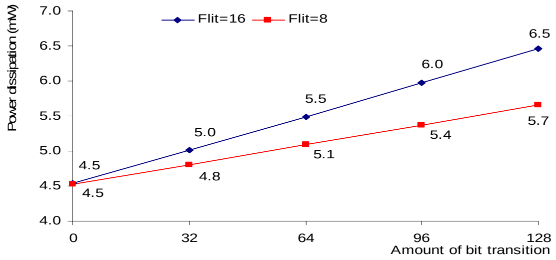


Figure 2 – Bit transition effect on dynamic energy of control logic with 8-bit and 16-bit flit width.

While control circuit does not depends on flit size, energy consumption on buffers increases linearly. This effect is illustrated in Figure 3.

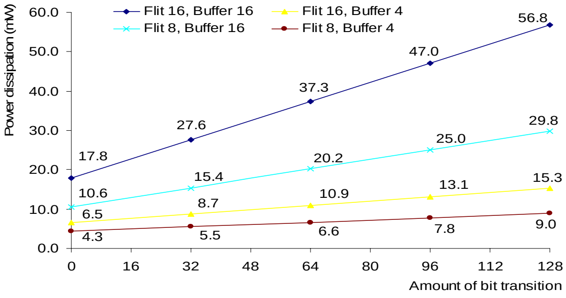


Figure 3 – Bit transition effect on dynamic energy consumption of buffers with 8 and 16-bit flit width and 2 buffer sizes (4 and 16 flit-positions).

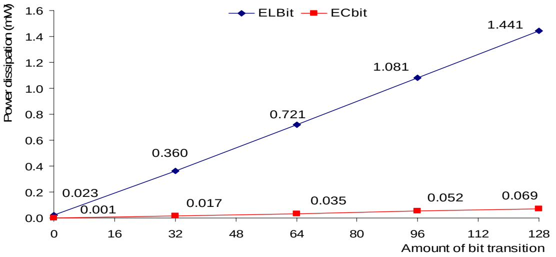


Figure 4 – Effect of bit transitions on dynamic energy consumption of local and inter-router links. Each tile has 5 mm x 5 mm of dimension, and uses 16-bit links.

In regular tile-based architectures, tile dimension is normally close to the average core dimension, and the core inputs/outputs are normally placed near the router local channel. As a consequence, *ECbit* is much smaller than *ELbit*. Figure 4 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 due to the fact that a physical link is equivalent to a large RC circuit if compared to local links.

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 *i* to tile *j*, where *η* corresponds to the number of routers through which the bit passes.

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

3.1. Model Parameters Acquisition

To acquire the above energy parameters (*EBbit*, *ESbit*, and *ELbit*), 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 bidirectional 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 127.

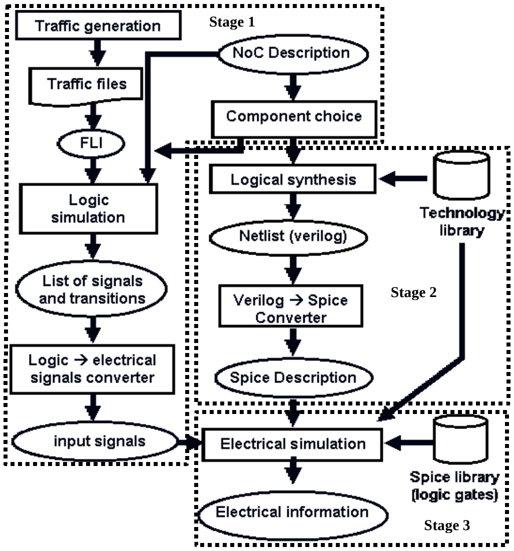


Figure 5 – Flow of model parameter acquisition.

The flow for obtaining dynamic energy consumption Figure 5 data comprises three stages. The first stage starts with the NoC VHDL description and traffic files, both obtained using an environment for NoC/NoC traffic generation [10]. 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 is used as input for a high-level energy consumption model of a NoC mesh topology.

4. Application Cores and NoC Models

Previous works [2][6] show that *ELbit*, *EBbit* and *ESbit* depend on the amount of bit traffic. On the other hand, Section 3 shows that the amount of bit transitions 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 [2] and [6]. 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 $<C, W>$. The set of vertices $C = \{c_1, c_2\ldots, c_n\}$ represents the set of application cores. Assuming w_{ab} is the number of bits of all packets sent from core a to core b, $W = \{(c_a, c_b, w_{ab}) \mid c_a, c_b \in C$ 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 $<C, T>$. The set of vertices $C = \{c_1, c_2\ldots, c_n\}$ 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 $\{(c_a, c_b, w_{ab}, t_{ab}) \mid c_a, c_b \in C, w_{ab} \in \mathbb{N}^*$ 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.

Figure 6 illustrates the above definitions using a hypothetical application with four IP cores exchanging a total of six packets and a 2×2 NoC. Figure 6 (a) shows a CWG where the set of vertices is C = {A, B, C, D}, and the set of edges is W = {(A, B, 100), (A, C, 120), (A, D, 60), (B, A, 80), (B, C, 80), (B, D, 80), (C, A, 90), (C, B, 120), (C, D, 90), (D, A, 100), (D, B, 50), (D, C, 80)}. Figure 6 (b) depicts an ECWG for the same hypothetical application and 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, 100, 0), (A, C, 120, 120), (A, D, 60, 30), (B, A, 80, 0), (B, C, 80, 40), (B, D, 80, 80), (C, A, 90, 90), (C, B, 120, 60), (C, D, 90, 0), (D, A, 100, 50), (D, B, 50, 50), (D, C, 80, 0)}.

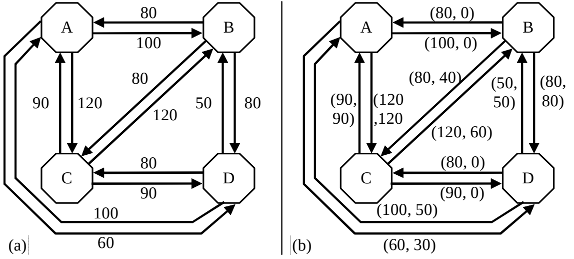


Figure 6 – (a) CWG and (b) ECWG.

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, called CRG, has its definition stated below.

Definition 3: A *communication resource graph* is a directed graph $CRG = <R, L>$, where the vertices set is the set of routers $R = \{r_1, r_2, \ldots, r_n\}$, 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.

ELbit, *EBbit* and *ESbit* parameters are used to represent the energy consumption of each one of the communication resources, when a bit passes through.

5. NoC Energy Consumption with CWM and ECWM Application Cores 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).

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

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).

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

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

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.

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

6. Experimental Results

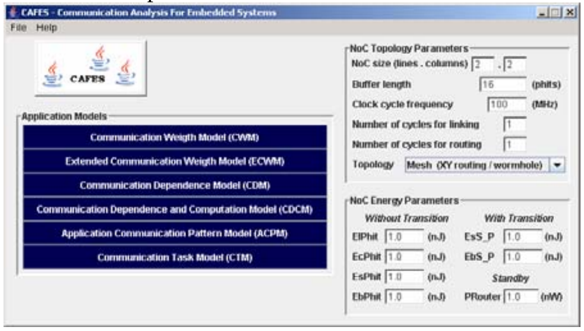
6.1. Estimation Tool

To estimate the experimental results we implemented a framework called CAFES (Communication Analysis for Embedded Systems).

CAFES provides the means to evaluate mappings, whose application may be described with models that consider different application aspects, like computation quantity and communication dependence.

For CWM and ECWM models, this work implements similar algorithms that mix simulated annealing and simulated evolution approaches. The main difference consists in the mapping objective functions that compute different NoC energy parameters.

Figure 7 shows an interface of CAFES, where the user can chose one of six application models, and also describe NoC parameters.


CAFES - Communication Analysis For Embedded Systems
File
Help
CAFES
Application Models
Communication Weigth Model (CWM)
Extended Communication Weigth Model (ECWM)
Communication Dependence Model (CDM)
Communication Dependence and Computation Model (CDCM)
Application Communication Pattern Model (ACPM)
Communication Task Model (CTM)
NoC Topology Parameters
NoC Energy Parameters
Without Transition
With Transition
Standby

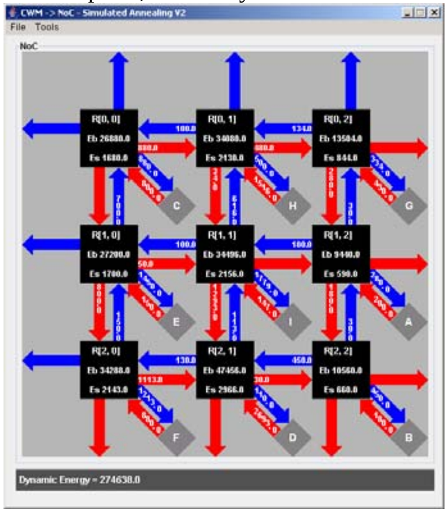
NoC size (lines . columns) 2 . 2
Buffer length 16 (phits)
Clock cycle frequency 100 (MHz)
Number of cycles for linking 1
Number of cycles for routing 1
Topology Mesh (XY routing / wormhole)

ElPhit 1.0 (nJ) EsS_P 1.0 (nJ)
EcPhit 1.0 (nJ) EbS_P 1.0 (nJ)
EsPhit 1.0 (nJ)
EbPhit 1.0 (nJ) PRouter 1.0 (nW)

Figure 7 – An interface of CAFES framework containing parameters of application models, NoC topology and NoC dynamic energy.

According to NoC topology, NoC energy parameters and the application model, CAFES estimate the dynamic energy consumption of different mapping for each application. Comparing the results achieved with CWM and ECWM algorithms, we evaluate the impact of traffic on mappings.

Figure 8 shows a mesh NoC with an arbitrary mapping achieve after the mapping algorithm execution. All application resources are noted with the dynamic energy consumption, caused by the bit traffic.



Dynamic Energy = 274630.0

Figure 8 – A mapping of application cores onto a 3x3 NoC mesh topology. Each link and router is marked with the total dynamic energy consumption.

6.2. Benchmarks and Results

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 [11]. Table 1 summarizes applications features and required NoC size.

Table 1 – Application features. Embedded applications are Video Object Plane Decoder (V) [12], MPEG4 decoder (M) [12], Fast Fourier Transform (F) [13], distributed Romberg integration (R) [14], object recognition and image encoding (O).

Application	NoC size	Number of cores	Total amount of (M bits)	
			Bits	Bits transition
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	567,777	[0, 567,777]
	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 model. This last column represents typical values of bit transitions for each embedded application, which can be extracted from functional simulation. For random applications the column represents minimum and maximum limits for bit transitions.

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 bit transition consumption 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 mapping algorithm. Table 2 and Table 3 compare the results for both algorithms.

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

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	32.90	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 due to the fact that for random applications it is used the minimum and maximum bit transitions amount and not a typical bit transition. 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 NoC mesh topologies 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. In our experiments, 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.

References

[1] W. Dally and B. Towles. *Route packets, not wires: on-chip interconnection networks*. **DAC**, pp. 684–689, Jun. 2001.

[2] J. Hu and R. Marculescu. *Energy-aware mapping for tile-based NoC architectures under performance constraints*. **ASP-DAC**, pp. 233-239, Jan. 2003.

[3] S. Murali and G. De Micheli. *Bandwidth-constrained mapping of cores onto NoC architectures*. **DATE**, pp. 896-901, Feb. 2004.

[4] C. Marcon, A. Borin, A. Susin, L. Carro and F. Wagner. *Time and Energy Efficient Mapping of Embedded Applications onto NoCs*. **ASP-DAC**, Jan. 2005.

[5] T. Ye; L. Benini and G. De Micheli. *Analysis of power consumption on switch fabrics in network routers*. **DAC**, pp.524-529, Jun. 2002.

[6] C. Marcon; N. Calazans, F. Moraes; A. Susin L. Reis and F. Hessel. *Exploring NoC Mapping Strategies: An Energy and Timing Aware Technique*. **DATE**, pp. 502-507, Mar. 2005.

[7] T. Ye; L. Benini and G. De Micheli. *Packetization and routing analysis of on-chip multiprocessor networks*. **JSA**, vol. 50, issues 2-3, pp. 81-104, Feb. 2004.

[8] N. Eisley; L. Peh. *High-Level Power Analysis of On-Chip Networks*. **CASES**, Sep. 2004.

[9] F. Moraes, N. Calazans, A. Mello, L. Möller and L. Ost. *HERMES: an infrastructure for low area overhead packet-switching networks on chip*. **VLSI the Integration Journal**, vol. 38, issue 1, pp. 69-93, Oct. 2004.

[10] L. Ost, A. Mello; J. Palma, F. Moraes, N. Calazans. *MAIA - A Framework for Networks on Chip Generation and Verification*. **ASP-DAC**, Jan. 2005.

[11] R. Dick, D. Rhodes and W. Wolf. *TGFF: task graphs for free*. **CODES/CASHE**, pp.97–101, Mar. 1998.

[12] E. Van der Tol and E. Jaspers. *Mapping of MPEG-4 Decoding on a Flexible Architecture Platform*. **SPIE** pp. 1-13, Jan, 2002.

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

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