

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

César A. M. Marcon<sup>1</sup>, José C. S. Palma<sup>2</sup>, Ney L. V. Calazans<sup>1</sup>, Fernando G. Moraes<sup>1</sup>, Altamiro A. Susin<sup>2</sup>, Ricardo A. L. Reis<sup>2</sup>

1 PPGCC/FACIN/PUCRS - Av. Ipiranga, 6681, Porto Alegre, RS – Brazil  
{marcon, calazans, moraes}@inf.pucrs.br

2 PPGC/II/UFRGS - Av. B. Gonçalves, 9500, Porto Alegre, RS – Brazil  
{jcspalma, susin, reis}@inf.ufrgs.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 complex system-on-a-chips (SoCs). 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. In applications that present a large amount of packet exchanges the error is propagated, significantly affecting the mapping results. The paper proposes a high-level application model that captures the traffic effect, enabling to estimate the dynamic energy consumption. In order to evaluate the quality of the proposed model, a set of real and synthetic applications were described using both, a previously proposed model that does not capture the bit transition 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. Comparing the resulting mappings, the model proposed displays an average improvement of 45% in energy saving.

## 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 adequate communication resources to cope with very tight design requirements. In addition, deep sub-micron effects pose formidable physical design

challenges for long wires and global on-chip communication [1]. Many designers propose to change from the fully synchronous design paradigm to globally asynchronous, locally synchronous (GALS) design paradigm [2]. GALS design subdivides an application into sub-applications. Each sub-application is a synchronous design physically placed inside a usually rectangular area of the chip, called *tile*. Besides, the communication between tiles is provided by asynchronous communication resources. Problems with wiring scalability are causing a migration from the use of busses to more complex and more scalable intra chip communication infrastructure and architectures. A network-on-chip (NoC) is such an infrastructure, composed by routers interconnected by communication channels. NoCs are suitable to deal with the above mentioned tight requirements, since they can support asynchronous communication, high scalability, reusability and reliability [3].

*Intellectual property cores* or *IP cores* or simply *cores* are pre-designed and pre-verified complex hardware modules, which can be considered as key components in the development of SoCs. Consider a SoC implemented using the GALS paradigm, composed by  $n$  cores and employing a NoC as communication infrastructure. The *application mapping* problem or simply the *mapping* problem for this architecture consists in finding an association of each core to a tile (a *mapping*) for an SoC such that some cost function is minimized. Naturally, cost functions are derived from latency and/or throughput and/or power dissipation figures.

Assuming there are  $n$  equally-sized tiles to where any of the  $n$  cores can be assigned, the mapping problem allows  $n!$  distinct solutions. The cost of using exhaustive search algorithms to solve the mapping problem is obviously prohibitive for even moderately sized NoCs (e.g. 4x4 2D meshes). Consequently, the search of an optimal implementation for such SoCs requires efficient mapping strategies and sound application models. Some mapping strategies have already been proposed. *Core graphs* [4] and *application characterization graphs* (APCGs) [5] are instances of a same generic model supporting the solution of the mapping problem. This model is called here *communication weighted model* (CWM) [6], since it takes into account only the amount of communication exchanged between pairs of cores.

One important observation is that CWM models abstract at least one important traffic information that affects dynamic energy consumption estimation, namely 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 consumes dynamic energy. Experiments based on the traffic behavior for some applications showed that considering either bit transition or amount of bits may lead to estimation discrepancies of more than 100% in dynamic energy consumption (see Section 3). For instance, an implementation of a *16-word NoC router input buffer*<sup>1</sup> implemented with CMOS TSMC 0.35 $\mu$ m technology, showed a difference of more than 180% in dynamic energy consumption when comparing minimum (zero) and maximum values of bit transitions (127) for a 128-flit packet. This prevents the choice of an average value of bit energy consumption or the use of only bit transition information as sound. Consequently, the effect of omitting the amount of bit transitions or the bit volume onto a NoC traffic modeling

<sup>1</sup> The router input buffer is a subcircuit of the Hermes NoC [11].

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 amount 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 work, Section 2 discusses related work, while Section 3 presents the dynamic energy consumption model for NoC, justifying its proposition in more detail. Next, Section 4 defines the target architecture model and the application models. Section 5 shows how application models are applied on 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 [7] introduced a framework to estimate the energy consumption in a communication infrastructure considering routers, internal buffers, and interconnect wires. The framework includes a simulation facility 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 [4-6, 8] and here.

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 a model that captures the application core communication. Murali and De Micheli [5] proposed a similar model; both models are here classified as CWMs. The main contribution of their work is an algorithm to map cores on 2D mesh NoC architectures with bandwidth constraints minimizing average communication delay.

Marcon et al. [6] 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 [8], the same group proposes 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 obtained from design descriptions 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.

Peh and Easley [10] 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 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 phenomena 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 still accounts for the smallest part of the overall consumption [1]. 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 bits moving across the communication infrastructure. In NoCs dynamic power is dissipated in interconnect wires and inside each router. Several authors [4-9] have proposed to estimate NoC energy consumption by evaluating the effect of bit traffic and packet traffic on each component of the communication infrastructure. This work evaluates the dynamic energy consumption for 2D mesh NoCs with regular topology only. The choice of regular topologies facilitates the estimation of interconnect wires length, and consequently the accounting of their influence on dynamic energy consumption.

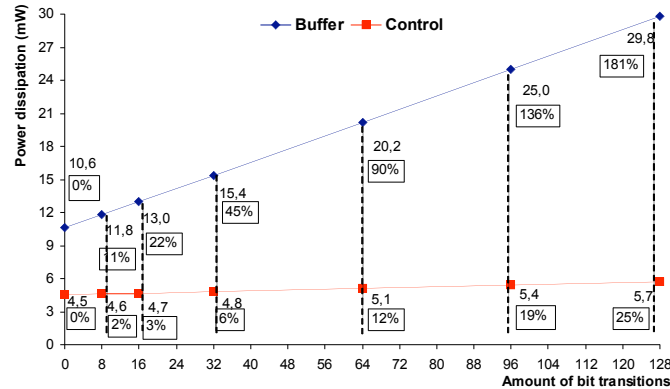
The bit energy notation  $E_{Bit}$  stands for an estimation of the dynamic energy consumption of each bit. It can be split into four components: bit dynamic energy, consumed into router buffers ( $E_{Bbit}$ ); bit dynamic energy, consumed into router control ( $E_{Sbit}$ ), and comprised by router wires and control logic gates; bit dynamic energy, consumed on links between tiles ( $E_{Lbit}$ ); and bit dynamic energy consumed on links between a router and the local core ( $E_{Cbit}$ ). The relationship between these quantities is expressed by Equation (1), which gives a way to estimate the dynamic energy consumption of a bit crossing a router, a local link and an inter-tile link.

$$E_{Bit} = E_{Bbit} + E_{Sbit} + E_{Lbit} + E_{Cbit} \quad (1)$$

This Section evaluates the effect of the above parameters on the computation of the overall dynamic energy consumption of a given SoC. Data were obtained from SPICE simulation of the Hermes NoC [11] synthesized for CMOS TSMC 0.35 $\mu$ m technology.

Even if the exact values of energy dissipation are subject to NoC implementation technology parameters, in general bit transitions affect much more the router buffers energy consumption than the consumption of router control circuits. This assertion is

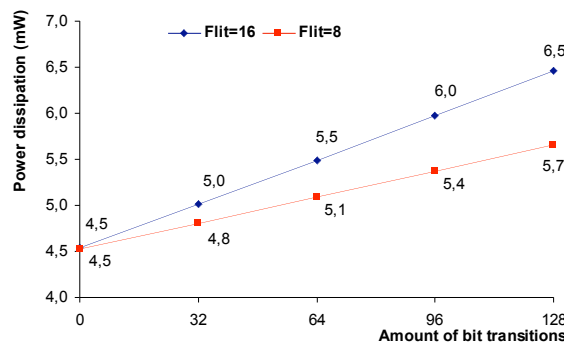
corroborated by **Fig. 1**, which illustrates this issue for different sizes of the Hermes router buffers with 8-bit flit width and centralized control logic. The graph depicts power dissipation of router buffers and router control circuits as a function of the amount of bit transitions in a 128-bit packet.



**Fig. 1.** Bit transition effect on dynamic energy consumption of buffers and of control circuits. The traffic is one 128-bit packet with bit transitions varying from 0% to 100%. Percentage results are computed w.r.t. the respective zero bit transition dissipation values.

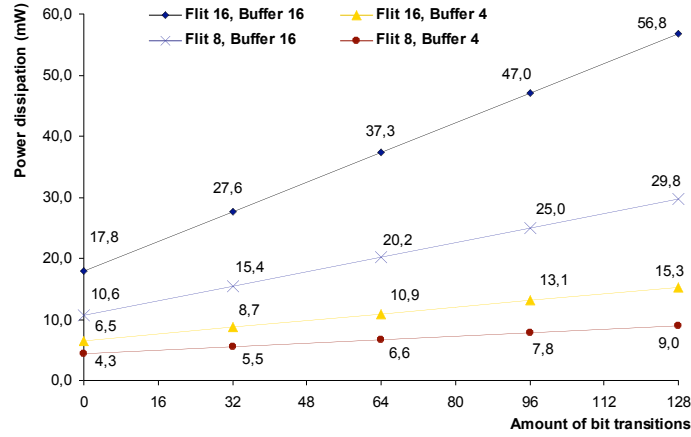
Clearly, energy consumption increases linearly and is directly proportional to the amount of bit transitions in the packet. However, the bit transition effect on the increase of dynamic energy consumption is around five times more pronounced for buffers than for control circuits. For instance, from 0% to 100% of bit transitions, the energy consumption increases from 0% to 181% on buffers against only 0% to 25% on control circuits.

**Fig. 2** compares the same bit transition effect in Hermes control logic circuit with 8-bit and 16-bit flit width. It is noticeable that the amount of bit transition has small influence over the control circuits’ energy consumption, even with significant increase in flit width.



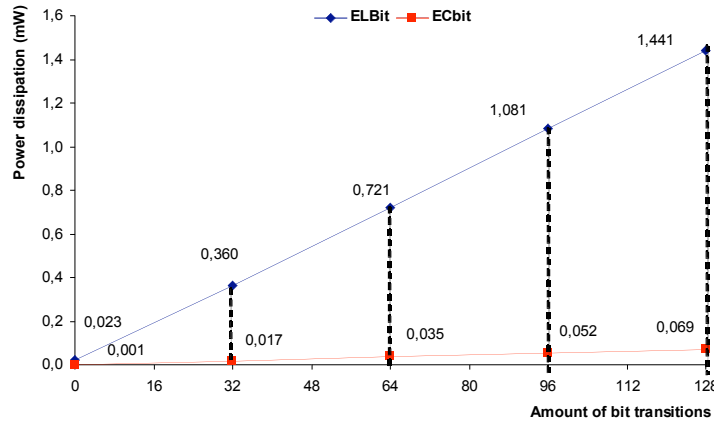
**Fig. 2.** Comparing bit transition effect on dynamic energy of control logic for 8- and 16-bit flit widths.

While the flit width has a small influence on dynamic energy consumption of control circuits, the same parameter cannot be neglected without consequences on the dynamic energy consumption of buffers, as illustrated by **Fig. 3**.



**Fig. 3.** Bit transition effect on dynamic energy consumption of buffers with 8 and 16-bit flit width and buffer depths of 4 and 16 flit positions.

In tile-based architectures with regular topology, the tile dimension is normally close to the average core dimension, and the core router interface is normally formed by small wires compared to inter-router wires. As a consequence,  $EC_{bit}$  is much smaller than  $EL_{bit}$ . **Fig. 4** corroborates this last statement, by comparing energy consumption for local and inter-tile links. A twenty-fold difference arises in the magnitude of energy consumption between  $EL_{bit}$  and  $EC_{bit}$ . This occurs because a link is an RC circuit, and inter-tile links are much longer than local ones.



**Fig. 4.** Analysis of bit transitions effect on dynamic energy consumption of local and inter-router links. Each tile is assumed to have a dimension of 5mm × 5mm, and uses 16-bit links of metal wires in CMOS TSMC 0.35μm technology.

Considering these results,  $ECbit$  may be safely neglected without significant errors in total energy dissipation. Therefore, Equation (2) can be used to compute the dynamic energy consumed by a single bit traversing the NoC, from tile  $i$  to tile  $j$ , where  $\eta$  corresponds to the number of routers through which the bit passes.

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

In addition, **Fig. 4** also shows that the dynamic energy consumption of links becomes significant only in the presence of a relatively high percentage of bit transitions. The  $ELbit$  parameter is related only to the amount of bit transition while  $EBbit$  and  $ESbit$  are each sub-divided into two new parameters, corresponding to the effect of the amount of communication and the bit transition rate.

### 3.1 Model Parameters Acquisition

To estimate the above bit 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 typical element is a router with five bidirectional channels connecting to four other routers and to a local IP core. The router of Hermes employs an XY routing algorithm, and uses input buffering only. The conducted experiments used 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.

The flow for obtaining dynamic energy consumption data is depicted in **Fig. 5** and comprises three stages.

Stage 1 starts with the NoC VHDL description and traffic files, both obtained using Maia [12], an environment for automating NoC design capture and NoC traffic generation. Traffic input files enable to exercise the NoC through each router local channels. They model the communication behavior of local cores. A VHDL simulator applies input signals from traffic files to the NoC or to NoC modules (either a single router or a router inner module, i.e. input buffer or control logic). Traffic files and VHDL design files are connected using a Foreign Language Interface (FLI) method.

The simulation produces signal lists capturing the logic values variations for each signal. These lists are converted to electric stimuli and used in SPICE simulation (in Stage 3).

In Stage 2, the module to be evaluated (e.g. an input buffer) is synthesized using a technology cell library, such as CMOS TSMC 0.35, constraining the cells used in the synthesis tool to the ones available for electrical simulation. The synthesis process generates an HDL netlist, later translated to a SPICE netlist using a converter developed in the scope of this work.

Stage 3 consists in a SPICE simulation of the module under analysis. Here, it is necessary to integrate the SPICE netlist of the module, the electrical input signals and a library with logic gates described in SPICE. The resulting electrical

information is used to estimate  $EBbit$ ,  $ESbit$ , and  $ELbit$ , which is used as input to a high-level energy consumption model of a NoC mesh topology.

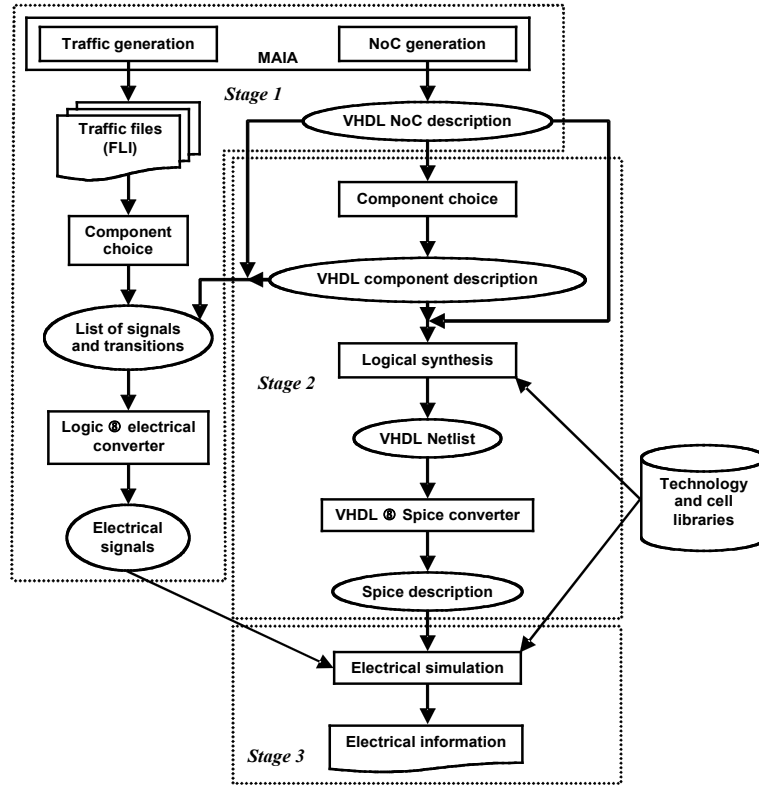


Fig. 5. Parameter acquisition flow for the high-level dynamic energy model.

#### 4. Application Cores and NoC Models

Previous works [4, 8] have showed that estimating  $EBbit$ ,  $ESbit$  and  $ELbit$  parameters requires the knowledge of amount of bit traffic. On the other hand, Section 3 showed 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 comparable to the effect obtained by varying the amount of bit traffic as described, for example, in [4] and [8]. Finally,  $ELbit$  is mostly 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 enhancement of CWM that also captures the amount of bit



transitions in a given communication. These models underlie the structures that enable to represent them (CWG and ECWG), as defined below.

**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 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.

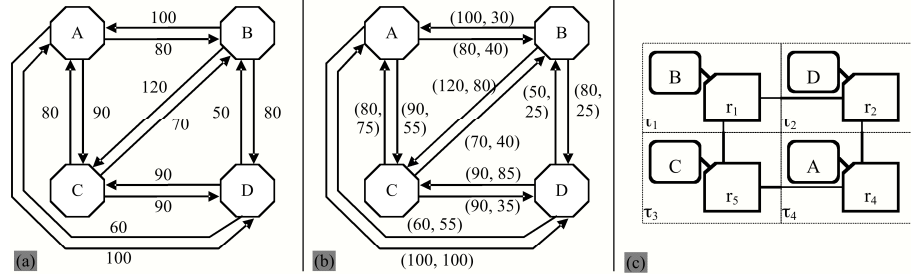
While CWM and ECWM model application cores communication, NoCs are modeled by a graph that represents their physical components, i.e. routers and links. This graph, called CRG, is defined next.

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

Value  $n$  is the total number of routers and is equal to the product of the two NoC dimensions in 2D mesh topologies. 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 2D mesh topology NoC using wormhole switching and deterministic XY routing algorithm. Nevertheless, other NoC topologies can be similarly considered, just changing the CRG formulation.

**Fig. 6** illustrates the above definitions using a synthetic application with four IP cores exchanging a total of six packets in a  $2 \times 2$  NoC. **Fig. 6 (a)** shows a CWG where the set of vertices is  $C = \{A, B, C, D\}$ , and the set of edges is  $W = \{(A, B, 80), (A, C, 90), (A, D, 100), (B, A, 100), (B, C, 120), (B, D, 80), (C, A, 80), (C, B, 70), (C, D, 90), (D, A, 60), (D, B, 50), (D, C, 90)\}$ . **Fig. 6 (b)** depicts an ECWG for the same synthetic application and the same set of vertices. However, each edge also contains the amount of bit transitions of the communication. The set of edges is  $T = \{(A, B, 80, 40), (A, C, 90, 55), (A, D, 100, 100), (B, A, 100, 30), (B, C, 120, 80), (B, D, 80, 25), (C, A, 80, 75), (C, B, 70, 40), (C, D, 90, 35), (D, A, 60, 55), (D, B, 50, 25), (D, C, 90, 85)\}$ . **Fig. 6 (c)** depicts an arbitrary mapping of  $C$  onto a NoC mesh  $2 \times 2$ , 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  $L = \{(r_1, B), (r_2, D), (r_3, C), (r_4, A)\}$ .



**Fig. 6.** (a) CWG and (b) ECWG of a synthetic application, and (c) core mapping onto a NoC mesh 2x2.

## 5. NoC Energy Consumption with CWM and ECWM Application Cores Models

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

Let  $\tau_i$  and  $\tau_j$  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, Equation (3) shows how CWM computes the dynamic energy consumed on this communication by associating  $w_{ab}$  with Equation (2).

$$\begin{aligned} ECommunication_{ab} &= w_{ab} \times EBit_{ij} \\ &= w_{ab} \times (\eta \times (EBbit + ESbit) + (\eta - 1) \times ELbit) \end{aligned} \quad (3)$$

$ECommunication_{ab}$  is computed differently for ECWM, since  $ELbit$ ,  $EBbit$  and  $ESbit$  have different values for the amount of bit traffic and for the amount of bit transitions. Let 1 be the index representing the fraction of  $EBit_{ij}$  due to the amount of bit traffic only ( $EBit_{ij1}$ ) and let 2 be the index representing the fraction of  $EBit_{ij}$  due to the amount of bit transitions only ( $EBit_{ij2}$ ). Then, Equation (4) relates these amounts and Equation (5) expands Equation (4). As stated in Section 3.1,  $ELbit_2$  is not significant, which allows simplifying Equation (5).

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

$$\begin{aligned} ECommunication_{ab} &= \eta \times (w_{ab} \times (EBbit_1 + ESbit_1) + t_{ab} \times (EBbit_2 + ESbit_2)) + \\ &\quad (\eta - 1) \times t_{ab} \times ELbit_2 \end{aligned} \quad (5)$$

For both models, Equation (6) computes the total amount of *NoC dynamic energy consumption* ( $EDyNoC$ ), i. e. the summation of dynamic energy consumption for all communications between application cores. Let  $D$  be the set of edges in the model graphs, i.e. either  $W$  for CWG or  $T$  for ECWG. Then,  $EDyNoC$  represents the

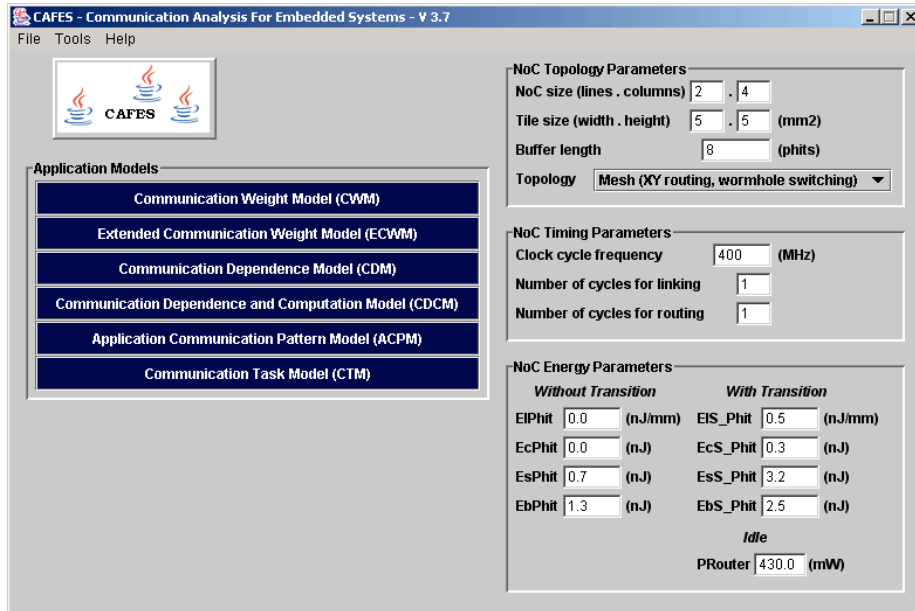
objective function for the NoC mapping problem considering the use of CWM and ECWM models.

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

## 6. Experimental Results

### 6.1 Estimation Tool

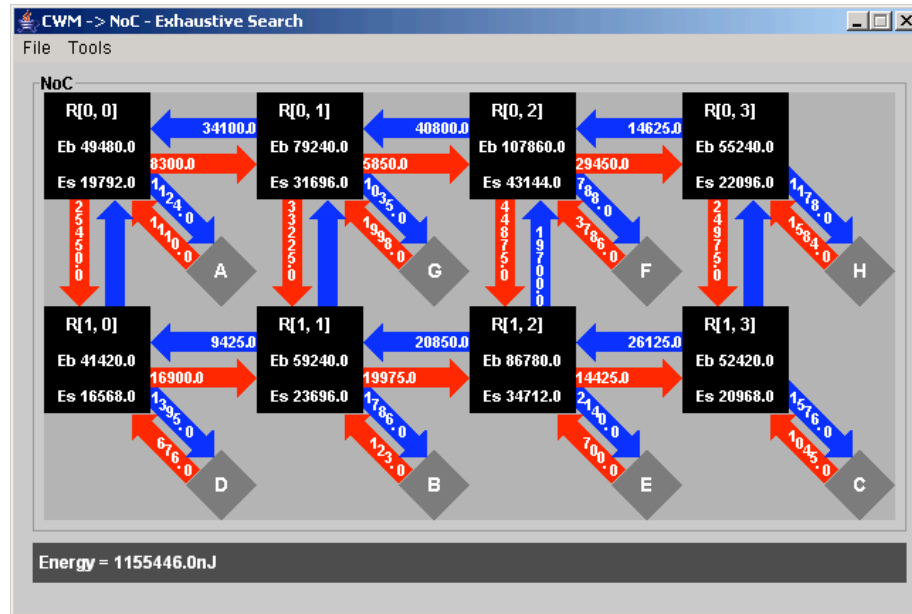
A framework called CAFES (Communication Analysis For Embedded Systems), developed in the context of this work, supports the generation of experimental results based on the equations developed before. CAFES enables to evaluate mappings of application cores into NoCs. The behavior of an application can be described with models that consider different aspects, with respect to computation and communication. **Fig. 7** shows the starting window of CAFES graphical user interface (GUI). Here, the user can choose one of six application models, and also describe some of the NoC parameters.



**Fig. 7.** Starting window for the CAFES framework GUI. It displays application model choices and supports the specification of NoC topology and NoC energy consumption parameters.

According to the choice of NoC topology, NoC energy parameters and application model, CAFES estimate the energy consumption of different mappings for each application. It also helps finding mappings to reduce communication latency.

**Fig. 8** shows a 2D mesh NoC with a mapping obtained after algorithm execution. All application resources are annotated with the computed dynamic energy consumption caused by the bit traffic.



**Fig. 8.** A mapping of application cores onto a 2x4 NoC mesh topology. Each link and router is marked with the computed dynamic energy consumption

CAFES implements algorithms mixing simulated annealing and simulated evolution approaches for both, CWM and ECWM models. The only difference between these algorithms is the employed mapping objective functions. Each function considers different NoC energy parameters. Comparing the results achieved with CWM and ECWM algorithms, it is possible to evaluate the impact of traffic on mappings. When compared to CWM, experimental results showed that the increased detailing of ECWM does not significantly affect the algorithm complexity neither in terms of memory usage nor in CPU time.

## 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 synthetic applications generated by a proprietary system similar to TGFF [13]. **Table 1**

summarizes applications features and required NoC size.

**Table 1.** Application features. Embedded applications are Video Object Plane Decoder (V) [14], MPEG4 decoder (M) [14], Fast Fourier Transform (F) [15], distributed Romberg integration (R) [16], object recognition and image encoding (O).

Application	NoC size	Number of cores	Total amount (in Mbits)	
			Transmitted	Bit transitions
Embedded	3 × 4 (V)	12	4,268	815
	4 × 5 (M)	17	3,780	720
	6 × 6 (F)	33	343	170
	7 × 7 (R)	49	219	175
	8 × 8 (O)	64	65,555	20,934
Synthetic	5 × 5	22	120	[0, 120]
	7 × 9	60	450	[0, 450]
	8 × 8	62	2,390	[0, 2,390]
	10 × 8	77	3,456	[0, 3,456]
	10 × 11	107	567,777	[0, 567,777]
	10 × 12	115	23,432	[0, 23,432]

\* [minimum, maximum]: synthetic applications exhaustively explore the full range of bit transition values.

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/Transmitted* column reflects the number of bits transmitted during application execution, and is used on both models, while the *Total amount/Bit transitions* 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 synthetic applications the column represents minimum and maximum limits for bit transitions. Here, both limits are explored to evaluate the difference from minimum to maximum energy consumption.

**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 × 4	2.47	2.09	18.18
4 × 5	2.53	2.23	13.45
6 × 6	0.65	0.63	3.17
7 × 7	0.33	0.25	32.00
8 × 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 bit transition consumption to compute the values for bit energy parameters. To do so, *E<sub>Bit</sub>* values were estimated according to 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 3.** Comparison of dynamic energy consumption for synthetic applications after applying CWM and ECWM mappings algorithms.

NoC size	CWM(mJ)	Minimum bit transitions		Maximum bit transitions	
		ECWM(mJ)	CWM/ECWM(%)	ECWM(mJ)	CWM/ECWM(%)
5 × 5	0.47	0.35	33.33	0.34	38.89
7 × 9	0.76	0.52	44.93	0.53	42.86
8 × 8	2.22	1.49	49.25	1.40	58.73
10 × 8	2.36	1.70	38.89	1.77	33.33
10 × 11	275.10	178.82	53.85	184.32	49.25
10 × 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 respectively 16.3% and 45.6% on dynamic energy savings, when comparing ECWM and CWM mappings. The second value is the average between minimum and maximum bit transition improvements. Synthetic applications differ more than embedded ones. This is due to the fact that for synthetic applications the minimum and maximum bit transition amount values are used 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 addressed the problem of mapping applications cores onto tiles of 2D NoC mesh topologies. It emphasized the importance of bit transition on traffic modeling for 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 showed the importance of considering the bit transition amount together with the amount of bits transmitted between application cores to achieve quality mappings. Often, solutions to the mapping problem aim at minimizing dynamic energy consumption in the communication infrastructure. It has been shown here that dynamic energy consumption grows linearly with the amount of bit transitions. In the conducted 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 the proposition of the Extended Communication

Weighted Model (ECWM), which builds on a previously proposed model, called Communication Weighted Model (CWM). While CWM only captures the amount of bit traffic, ECWM also contemplates the amount of bit transition. The conducted experiments showed that ECWM obtains significant energy consumption savings when compared to CWM in all cases.

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

## References

1. R. Ho and K. Mai and M. A. Horowitz. The future of wires. *Proceedings of the IEEE*, vol. 89 no. 4, pp. 490–504, Apr. 2001.
2. A. Iyer and D. Marculescu. Power and performance evaluation of globally asynchronous locally synchronous processors. In: *29th Annual International Symposium on Computer Architecture (ISCA)*, pp. 158-168, May 2002.
3. W. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In: *Design Automation Conference (DAC)*, pp. 684–689, Jun. 2001.
4. J. Hu and R. Marculescu. Energy-aware mapping for tile-based NoC architectures under performance constraints. In: *Asia 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. In: *Design, Automation and Test in Europe (DATE)*, pp. 896-901, Feb. 2004.
6. C. Marcon, A. Borin, A. Susin, L. Carro and F. Wagner. Time and Energy Efficient Mapping of Embedded Applications onto NoCs. In: *Asia Pacific Design Automation Conference (ASP-DAC)*, pp. 33-38, Jan. 2005.
7. T. Ye; L. Benini and G. De Micheli. Analysis of power consumption on switch fabrics in network routers. *DAC*, pp.524-529, Jun. 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. In: *Design, Automation and Test in Europe (DATE)*, pp. 502-507, Mar. 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, Feb. 2004.
10. N. Eislely; L. Peh. High-Level Power Analysis of On-Chip Networks. In: *7th International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, Sep. 2004.

16 César A. M. Marcon, José C. S. Palma, Ney L. V. Calazans, Fernando G. Moraes, Altamiro A. Susin, Ricardo A. L. Reis

11. 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.

12. L. Ost, A. Mello; J. Palma, F. Moraes, N. Calazans. MAIA - A Framework for Networks on Chip Generation and Verification. In: *Asia Pacific Design Automation Conference (ASP-DAC)*, pp. 49-52, Jan. 2005.

13. R. Dick, D. Rhodes and W. Wolf. TGFF: task graphs for free. In: *6th International Workshop on Hardware/Software Co-Design (CODES/CASHE)*, pp.97-101, Mar. 1998.

14. E. Van der Tol and E. Jaspers. Mapping of MPEG-4 Decoding on a Flexible Architecture Platform. In: *Proceedings of the International Society for Optical Engineering (SPIE)*, Vol 4674, pp. 1-13, Jan, 2002.

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

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