

# Inserting Data Encoding Techniques into NoC-Based Systems

José C. S. Palma<sup>1</sup>, Leandro Soares Indrusiak<sup>2</sup>, Fernando G. Moraes<sup>3</sup>,  
Alberto Garcia Ortiz<sup>2</sup>, Manfred Glesner<sup>2</sup>, Ricardo A. L. Reis<sup>1</sup>

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

2. MES – TU Darmstadt –  
Karlstr. 15, 64283 Darmstadt -  
Germany

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

[jcspalma, reis]@inf.ufrgs.br, [lsi, agarcia, glesner]@mes.tu-darmstadt.de, Moraes@inf.pucrs.br

## Abstract

*This work investigates the reduction of power consumption in Networks-on-Chip through the reduction of transition activity using data coding schemes. Power macromodels for NoC and encoding modules were built, allowing the estimation of the power consumption as a function of the transition activity at each module input. Power macromodels are embedded in a system model and a set of simulations are performed, analyzing the trade-off between the power savings due to coding schemes versus the power consumption overhead due to the encoding and decoding modules.*

## 1. Introduction

Networks-on-Chip (NoCs) are infrastructures essentially composed of routers interconnected by communication channels. They are suitable to support the GALS (*Globally Asynchronous, Locally Synchronous*) paradigm [1], since they provide asynchronous communication, scalability, reusability and reliability [2].

The growing market for portable battery-powered devices adds a new dimension, power, to the VLSI design space, previously characterized by speed and area [3]. Power consumption is directly related to battery life as well as costly package and heatsink requirements for high-end devices [4]. One problem related to power consumption in busses is the capacitances induced by long wires. Such problem is minimized in NoCs, since point-to-point short wires are used between routers. However, NoCs consume power in routers, diminishing the apparent advantage in terms of power when compared to busses.

The power consumption in a NoC grows linearly with the amount of bit transitions in subsequent data packets sent through the interconnect architecture [5]. Using the Hermes NoC architecture [6] as case study, the experiments show that bit transitions affect the power consumption up to 370% for interconnect lines, 180% for router input buffers and 16% for router control logic.

A way to reduce power consumption in NoCs, in both wires and logic, is to reduce the switching activity by means of coding schemes. Several schemes were proposed in the late 90's, all of them addressing bus-based communication

architectures [7].

The main contribution of this work is the evaluation of such schemes in the context of NoC-based systems, analyzing the trade-off between power savings and power consumption overhead due to encoding additional circuitry.

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

## 2. Encoding Schemes

Based on the observation that it is possible to reduce the power dissipation on bus drivers by reducing the average number of signal transitions, several encoding schemes have been previously proposed. Some of these schemes require *a-priori* knowledge of the statistical parameters of the input traffic, but in this work we focus on schemes that do not require such knowledge as we intend to apply them on general-purpose NoC-based systems. Four encoding schemes were analyzed: Adaptive Encoding [7], Bus-Invert [8], Gray [9] and Transition [10].

### 2.1. Adaptive Encoding

In [7] the authors propose an Adaptive Probability Encoding scheme which is capable of on-line adaptation of the encoding, according to transmitted data. This scheme operates bit-wise, according to its actual and the last value.

The encoding is done based on statistical information gathered by observation of the bit stream over a window of fixed size. The statistical information concern the four joint probabilities for a single bit ( $P_{0,0}$ ,  $P_{0,1}$ ,  $P_{1,0}$  and  $P_{1,1}$ ). Such scheme uses four different encoding functions to determine the output data. The function selection is done according to the joint probabilities of the current window.

### 2.2. Bus-Invert

Bus-Invert [8] uses an extra control bit (side band signal)

called *invert*, which indicates when the data value in the communication lines are inverted or not. The data value is inverted when the Hamming distance (the number of bits in which they differ) between the present value and the next data value is bigger than half of the number of lines.

In busses with word size larger than 8 bits (for example 16 or 32), the authors claims that the method is more efficient splitting the word in clusters of 8 bits. In this case, each cluster has its own control bit. The Hamming distance and codification are performed separately in each cluster.

### 2.3. Gray

The Gray encoding method [9] is very efficient when applied to address busses. A sequence of consecutive numbers, when codified in Gray method, presents in each word only one bit different from the previous word.

The conversion from binary to Gray consists of repeating the most significant bit of the word to encode, and using *xor* operations between each pair of consecutive bits of such word. The conversion from Gray to binary is also performed repeating the most significant bit of the word to decode and using *xor* operations. However, each bit to be decoded depends on the previous decoded bit, which increases the critical path and complexity when compared to the encoding function.

### 2.4. Transition

The Transition method [10] consists of sending a logic signal '1' for each bit of the word to be encoded when such bit differs from its correspondent bit in the previously encoded word. In other words, '1' indicates a transition in that transmission wire and '0' indicates no transition. To perform this function, the previously encoded word must be temporarily stored in a buffer and compared to the current word to encode.

## 3. Adding Coding Modules into NoCs

In Networks-on-Chip the data is transmitted in packets, which are sent through routers, from sources to targets. These packets are composed by a header (containing routing information) and a payload (containing the data to be transmitted). Usually, the header is composed by two flits<sup>1</sup>, comprising the target address and the packet length. Thus, an approach merging coding schemes and a Network-on-Chip should not encode the packet header, since it must be used by routers in every hop<sup>2</sup> through the Network-on-Chip. Encoding and decoding operations must be done in the source and target cores only, converting the original data to the encoded (and transmitted) data and vice-versa.

In this work, encoder and decoder modules are inserted between the router local ports and IP cores ports, avoiding changes in the NoC structure. An exception is the Bus-Invert method, which requires the insertion of extra control bits in all NoC internal modules.

<sup>1</sup> Smallest data unit transmitted over the network-on-chip.

<sup>2</sup> Hop is the distance between two routers in a network-on-chip.

## 4. Dynamic Power Consumption Model

The power consumption in a system originates from the operation of the IP cores and the interconnection components between those cores. It is proportional to the switching activity arising from packets moving across the network. Interconnect wires and routers dissipate power. As shown in [5], several authors have proposed to estimate NoC power consumption by evaluating the effect of bits/packets traffic on each NoC component.

The router power consumption is estimated splitting it into the buffer power consumption and the control logic power consumption. It is also important to estimate the power consumption in the channels connecting a router to another one, as well as in the channels connecting a router to its local core. In this paper, wire lengths of 5 mm for inter-router links and 0.25 mm for local links were considered. These data correspond to the required area for a small 32-bit RISC processor in a 0.35  $\mu\text{m}$  technology.

The conducted experiments employ a mesh topology version of Hermes with six different configurations. The parameters were obtained varying flit width (8 and 16 bits), and input buffer 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 in each communication wire.

Considering an approach with data coding, it is also necessary to compute the power consumed in the encoder and decoder modules. These modules were also simulated with 128-flit packets with different traffic patterns.

The flow for obtaining power consumption data comprises three steps. The first step starts with the NoC VHDL description (without coding scheme) and traffic files, both obtained using a customized environment for NoC configuration and traffic generation [11]. Traffic input files are fed to the NoC through the router local ports, modeling local cores behavior. A VHDL simulator applies input signals to the NoC or to any NoC module, either to a single router or to a router inner module (input buffer or control logic). Simulation produces signal traces storing the logic values variations for each signal. These traces are converted to electrical stimuli and later used in SPICE simulation (third step).

In the second step, the module to be evaluated (e.g. an input buffer) is synthesized within LeonardoSpectrum using a technology-specific cell library, such as CMOS TSMC 0.35 $\mu\text{m}$ . The tool produces an HDL netlist, later converted to a SPICE netlist using a converter developed internally for the scope of this work.

The third step 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 power consumption parameters for a given traffic.

After that, this process was repeated with a new version of the NoC which uses a coding scheme. In the case of Adaptive Encoding, Gray and Transition methods, only the encoder and decoder modules were developed in VHDL, synthesized to CMOS TSMC 0.35 $\mu$  technology, converted to a SPICE netlist and simulated with different traffic patterns.

In the case of Bus-Invert, besides the encoder and decoder modules, it was also necessary a new analysis of all NoC modules, since this scheme requires the insertion of control bits in all modules, increasing the power consumption of them.

#### 4.1. Model Definition

Average power per hop (*APH*), defined in [5], is used to denote the average dynamic power consumption in a single hop of a packet transmitted over the NoC. *APH* can be split into three components: average power consumed by a router comprised by buffers, router wires and logic gates for switching (*APR*); average power consumed on a link between routers (*APL*); and average power consumed on a link between the router and the system core attached directly to it (*APC*). Equation (1) gives the average power consumption of a packet transmitted through a router, a local link and a link between routers.

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

Moreover, the analysis presented in [5] shows that a better understanding of the average power consumption in the router (*APR*) can be achieved by dividing it into its buffer (*APB*) and control (*APS*) components. This is because the bit transition effect on power consumption at the router control is much smaller than its effect on the power consumption at the router buffer. Figure 1 and Figure 2 illustrate this effect for 4, 8 and 16-word input buffer and for the centralized control logic of an 8-bit and a 16-bit flit width Hermes router, respectively. The graphs depict power as a function of the amount of bit transitions in a 128-flit packet (100% = 127 bit transitions in each one of the input data wires). Clearly, power consumption increases linearly with the increase of bit transitions in a packet.

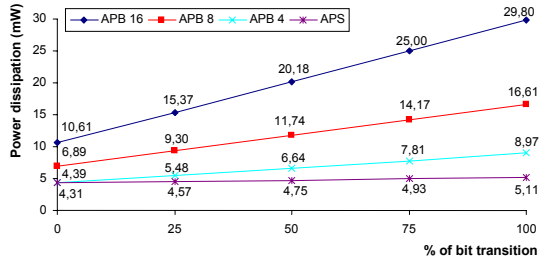


Figure 1: Analysis of the bit transition effect on the average power consumption for different buffer sizes and for the centralized control logic of an 8-bit flit width Hermes router.

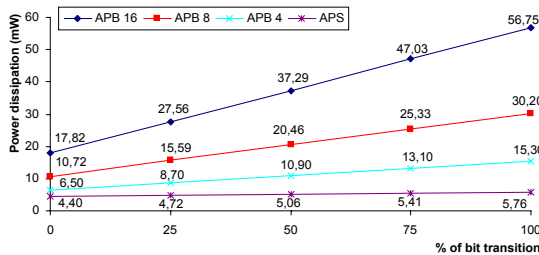


Figure 2: Analysis of the bit transition effect on the average power consumption for different buffer sizes and for the centralized control logic of a 16-bit flit width Hermes router.

In regular tile<sup>3</sup>-based architectures, tile dimension is close to the average core dimension, and the core inputs/outputs are placed near the router local channel. Therefore, *APC* is much smaller than *APL*, as showed in Figure 3.

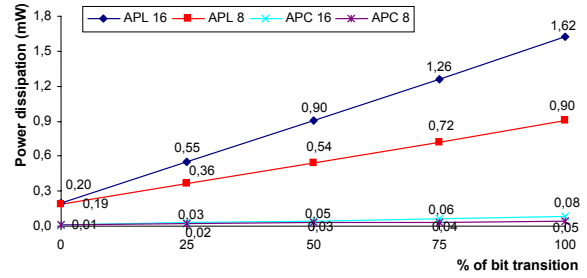


Figure 3: Analysis of the bit transition effect on the average power consumption of 8 and 16-bit flit width local and inter-router links. Each tile has 5 mm x 5 mm of dimension.

Based in these results, *APC* may be safely neglected without significant errors in total power dissipation. Therefore, Equation (2) computes the average router-to-router communication power dissipation, from tile  $\tau_i$  to tile  $\tau_j$ , where  $\eta$  corresponds to the number of routers through which the packet passes.

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

Considering now an approach with data coding in the NoC local ports, two new parameters may be introduced to Equation (2): *APE* and *APD* (encoder and decoder average power consumption, respectively).

Figure 4 and Figure 5 illustrate the analysis of the effect of bit transitions on power consumption of the encoder and decoder modules, respectively, both with 8-bit flit width.

Similarly, Figure 6 and Figure 7 show this effect on power consumption of the encoder and decoder modules with 16-bit flit width. As showed in these figures, power consumption also grows linearly in these modules as the amount of bit transitions is increased.

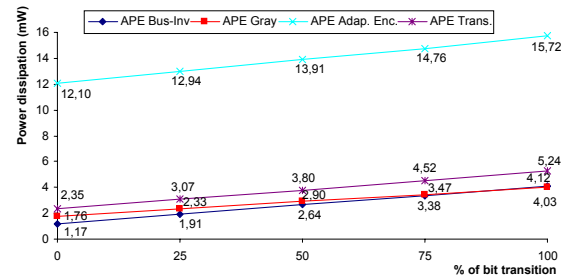


Figure 4: Analysis of the effect of bit transitions on the average power consumption of 8-bit encoder modules of different encoding schemes.

Observe that Adaptive Encoding was implemented only with the 8-bit NoC, since its encoder and decoder modules consume too much power (See Figure 4 and Figure 5). Note that Bus-Invert with 2 clusters was added in the 16-bit NoC, shown in Figure 6 and Figure 7.

<sup>3</sup> Cores are placed inside a limited region, which is usually called tile.

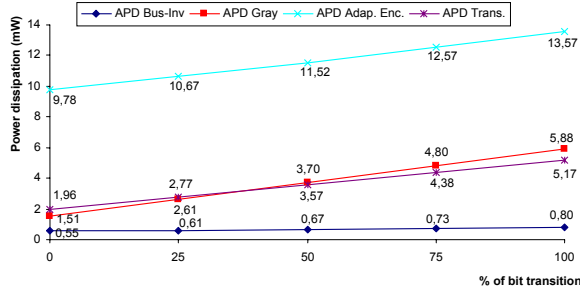


Figure 5: Analysis of the effect of bit transitions on the average power consumption of 8-bit decoder modules of different encoding schemes.

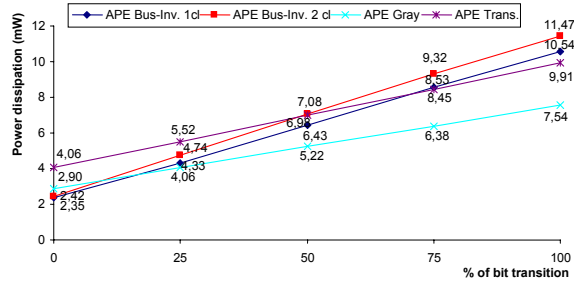


Figure 6: Analysis of the effect of bit transitions on the average power consumption of 16-bit encoder modules of different encoding schemes.

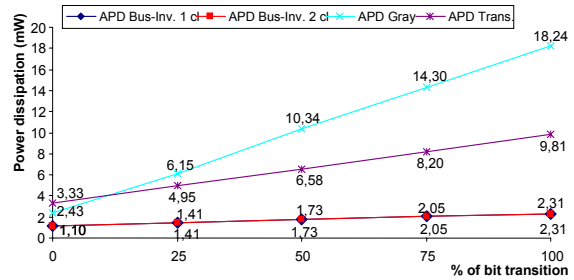


Figure 7: Analysis of the effect of bit transitions on the average power consumption of 16-bit decoder modules of different encoding schemes.

Based on the described analysis, it was possible to build macromodels for the several parts of the proposed model – *APB*, *APS*, *APL*, *APE* and *APD* - representing the power consumption in the different modules of a NoC.

Note that all modules (buffer, control, encoder and decoder) and also the communication channels consume power even with 0% of bit transitions on the data. This occurs due to static power consumption, switching of internal signals, internal state update and clock. In communication channels, this power consumption is due to clock and flow control signal activity. We refer it as  $P_o$  in the subsequent analysis. The remaining of the power consumption grows linearly with the bit transition rate (the slope is referred as  $R$ ). Table 1 shows, as an example, the macromodel for the NoC and Adaptive Encoding modules.

Considering a NoC with the Bus-Invert coding scheme, the power consumption must be calculated with basis on a new macromodel, which takes into account the extra bits of this scheme in all NoC modules. Table 2 shows the

macromodel for a NoC with flit width equal to 9 bits (8 data bits + 1 control bit).

Table 1: Macromodel for an 8-bit flit width NoC and Adaptive Encoding modules ( $AP = P_o + \%T \cdot R$ ).

Module	$P_o$	$R$
Buffer ( <i>APB</i> )	10.61	19.19
Control ( <i>APS</i> )	4.39	0.72
Encoder ( <i>APE</i> )	12.1	3.62
Decoder ( <i>APD</i> )	9.78	3.79
Inter-router channel ( <i>APL</i> )	0.19	0.71

Table 2: Macromodels for a 9-bit flit width NoC and Bus-Invert modules ( $AP = P_o + \%T \cdot R$ ).

Módulo	$P_o$	$R$
Buffer ( <i>APB</i> )	11.49	22.13
Control ( <i>APS</i> )	4.39	0.98
Encoder ( <i>APE</i> )	1.17	3.88
Decoder ( <i>APD</i> )	0.55	0.25
Inter-router channel ( <i>APL</i> )	0.19	0.8

The impact of bits insertion, required by Bus-Invert scheme is better clarified in Figure 8. This figure shows the average power consumption of a 16-word input buffer with 16-bit (normal), 17-bit (1 cluster) and 18-bit (2 clusters) flit width.

Equation (3) computes the average router-to-router communication power dissipation, from tile  $\tau_i$  to tile  $\tau_j$ , passing through  $\eta$  routers and using a coding scheme.

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

*APD*

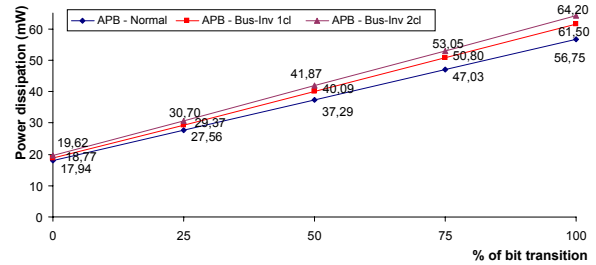


Figure 8: Average power consumption of a 16-word input buffer with different flit widths: normal (without control bits), 1 cluster and 2 clusters.

## 5. Dynamic Power Consumption Analysis

As stated in Section 4, the average power estimation depends on the communication infrastructure and on the application core traffic. The Hermes NoC is used as the communication infrastructure in all experiments, since its internal architecture share common features with most NoCs [12]: 2D mesh topology, wormhole packet switching, deterministic distributed routing, input buffering.

This Section shows the method to compute the average power consumption in a NoC with and without coding schemes. The average power consumption analysis is done with different traffic patterns. Such traffic patterns are real application data which was packetized and transmitted over the NoC.

In order to fully evaluate the traffic resulting from real application data, experiments must be performed with realistic amounts of data. However, the simulation times for hundreds of packets in a SPICE simulator are unfeasible. So, a better alternative was taken, exploring the possibility to

embed the macromodels into a higher abstraction model, which was simulated within the PtolemyII environment [13]. Such abstract model include a model of an encoder, a section of the NoC interconnect and a decoder. These models are used to track the percentage of bit transitions in each traffic pattern and, based on the energy macromodels, calculate the average power consumption for such traffic patterns.

## 6. Experimental Results

This section presents the experimental results obtained by system level simulation within Ptolemy II, using the macromodels described in Section 4.1, for different real traffic patterns.

In all tables, the first column describes the type of traffic. The second column presents the reduction of transition activity found on our experiments. The results are reported in terms of reduction in the number of transitions with respect to the original data streams. The third column shows the power consumption (*APH*) without use of data coding techniques, while the fourth column shows the same measurement when coding techniques are used. In the case of Bus-Invert, the third column is calculated with basis on the macromodel of the normal NoC (without coding scheme), while de fourth column is calculated with basis on the macromodel using Bus-Invert (with extra bits in all modules). Finally, the fifth and sixth columns present the power consumption overhead due to the encoder and decoder modules (*APE* + *APD*) and the number of hops which are needed to amortize this overhead. When the encoding scheme increases the transition activity it is not possible to amortize the overhead. In this case, the field in the sixth column is marked as a trace.

Table 3 presents the results obtained with Adaptive Encoding in an 8-bit flit width Hermes NoC. With most of the simulated traffics, the Adaptive Encoding is not effective. In some cases the encoded data increases the transition activity compared to the normal traffic. In other cases, even reducing the transition activity, it is necessary too many hops to amortize the power consumption of encoding and decoding modules. The best case is with the WAV stream, where the encoding power overhead can be amortized after 11 hops.

**Table 3: Results using Adaptive Probability Encoding in an 8-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	-1.4 %	22.24 mW	22.34 mW	24.1 mW	-
GZIP	1.03%	25.5 mW	25.4 mW	25.6 mW	240
GCC	0.91 %	24.69 mW	24.6 mW	25.27 mW	292
Bytecode	9.3 %	23.45 mW	22.68 mW	24.7 mW	32
WAV	21.95 %	25.5 mW	23.25 mW	25.17 mW	11
MP3	-2.41 %	24.8 mW	25 mW	25.38 mW	-
RAW	-10.98 %	22.24 mW	23 mW	24.5 mW	-
BMP	-0.5 %	25.3 mW	25.36 mW	25.5 mW	-
JPG	0.8 %	25.46 mW	25.38 mW	25.5 mW	327
TIFF	-1.16 %	25.34 mW	25.46 mW	25.55 mW	-
PDF	6.61 %	26.06 mW	25.34 mW	25.65 mW	36

Table 4 shows that the power consumption in the 8-bit flit width NoC increases with the Bus-Invert scheme, regardless of the fact that the bit transition was reduced with all traffic patterns. This is due to the inclusion of the extra bit in all NoC modules, increasing their power consumption, as presented in Figure 8. Only with the PDF stream the power consumption was reduced and amortized after 13 hops. This is possible because of a significant reduction of transition

activity (23,15%).

**Table 4: Results using 1-cluster Bus-Invert Encoding in an 8-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	6.2 %	21.33 mW	22.75 mW	2.93 mW	-
GZIP	18.7%	25.5 mW	25.79 mW	3.75 mW	-
GCC	17.9%	18.46 mW	19.18 mW	2.36 mW	-
Bytecode	12 %	23.45 mW	24.49 mW	3.35 mW	-
WAV	18.8 %	25.52 mW	25.6 mW	3.75 mW	-
MP3	18.48 %	25.3 mW	25.63 mW	3.7 mW	-
RAW	14.6 %	22.24 mW	23 mW	3.1 mW	-
BMP	18.2%	25.3 mW	25.66 mW	3.71 mW	-
JPG	19.5 %	25.46 mW	25.65 mW	3.74 mW	-
TIFF	18.3 %	25.25 mW	25.6 mW	3.7 mW	-
PDF	23.15 %	26 mW	25.76 mW	3.85 mW	13

Similar to Adaptive Encoding, Gray and Transition schemes are effective in an 8-bit flit width Hermes NoC with some traffic patterns, as presented in Table 5 and Table 6. The best case for power overhead amortization is 7 hops with Gray and 6 with Transition.

**Table 5: Results using Gray Encoding in an 8-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	- 11.36 %	22.24 mW	23.04 mW	5.71 mW	-
GZIP	- 0.49 %	25.5 mW	25.55 mW	6.6 mW	-
GCC	- 5.57 %	18.47 mW	18.65 mW	4.36 mW	-
Bytecode	0.21 %	23.56 mW	23.54 mW	5.96 mW	331
WAV	- 0.01 %	25.51 mW	25.51 mW	6.59 mW	-
MP3	- 0.06 %	25.31 mW	25.32 mW	6.53 mW	-
RAW	4.11 %	22.24 mW	21.95 mW	5.48 mW	19
BMP	11.82 %	21.22 mW	20.5 mW	5.06 mW	7
JPG	3.42 %	24.55 mW	24.23 mW	6.22 mW	19
TIFF	- 1.36 %	24.65 mW	24.78 mW	6.34 mW	-
PDF	3.5 %	26.06 mW	25.68 mW	6.69 mW	18

**Table 6: Results using Transition Encoding in an 8-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	- 2.97 %	22.24 mW	22.45 mW	6.42 mW	-
GZIP	1.17 %	25.5 mW	25.38 mW	7.34 mW	61
GCC	2.38 %	24.68 mW	24.46 mW	7.07 mW	31
Bytecode	9.22 %	23.18 mW	22.44 mW	6.55 mW	9
WAV	12.22 %	24.45 mW	23.32 mW	6.87 mW	6
MP3	- 0.21 %	25.3 mW	25.33 mW	7.3 mW	-
RAW	- 12.44 %	22.24 mW	23.12 mW	6.53 mW	-
BMP	- 0.02 %	25.3 mW	25.3 mW	7.3 mW	-
JPG	- 0.6 %	24.55 mW	24.61 mW	7.08 mW	-
TIFF	- 1.16 %	25.25 mW	25.37 mW	7.3 mW	-
PDF	7.52 %	26.06 mW	25.24 mW	7.39 mW	9

Table 7 and Table 8 present two scenarios using Bus-Invert scheme in a 16-bit flit width Hermes NoC. The first one splits the flit in two clusters, inserting 2 control bits in all modules. The second one uses one single cluster, inserting 1 control bit in all modules. As asserted in [8], the first approach, with clusters of 8 bits, is more efficient with respect of transition activity reduction. Nevertheless, the second approach is more effective in terms of power reduction. This is due to the fact of the power overhead of inserting 2 control bits is significant and not compensate by its transition activity reduction. With most of the simulated traffics, the number of hops to amortize the power overhead is between 4 and 6.

**Table 7: Results using 2-cluster Bus-Invert in a 16-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	11.73 %	37.16 mW	39.17 mW	7 mW	-
GZIP	22.83 %	43.17 mW	42.67 mW	8.47 mW	17
GCC	20 %	41.26 mW	41.57 mW	8.02 mW	-
Bytecode	17.08 %	37.9 mW	38.98 mW	7.23 mW	-
WAV	29.04 %	38.93 mW	37.7 mW	7.42 mW	6
MP3	23.14 %	43.02 mW	42.47 mW	8.43 mW	15
RAW	21.41 %	38.51 mW	38.77 mW	7.36 mW	-
BMP	22.81 %	42.94 mW	42.47 mW	8.41 mW	18
JPG	23.27 %	42.78 mW	42.22 mW	8.37 mW	15
TIFF	23.33 %	42.62 mW	42.06 mW	8.34 mW	15
PDF	23.99 %	42.82 mW	42.09 mW	8.38 mW	11



**Table 8: Results using 1-cluster Bus-Invert in a 16-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	10.57 %	37.16 mW	37.9 mW	6.6 mW	-
GZIP	19.59 %	43.17 mW	41.81 mW	7.84 mW	6
GCC	16.86 %	41.26 mW	40.67 mW	7.45 mW	13
Bytecode	12.89 %	37.9 mW	38.25 mW	6.75 mW	-
WAV	22.69 %	38.93 mW	37.46 mW	6.92 mW	5
MP3	19.9 %	43.02 mW	41.61 mW	7.81 mW	6
RAW	17.2 %	38.51 mW	38.08 mW	6.86 mW	16
BMP	19.19 %	42.94 mW	41.7 mW	7.8 mW	6
JPG	19.99 %	42.78 mW	41.38 mW	7.76 mW	6
TIFF	20.75 %	42.62 mW	41.06 mW	7.72 mW	5
PDF	21.52 %	42.82 mW	41.07 mW	7.76 mW	4

As show in Table 9 and Table 10, Gray and Transition schemes are effective in a 16-bit flit width Hermes NoC only for some kinds of traffic. Moreover, the number of hops necessary for amortization is high. The best case with Gray is 10 hops for WAV traffic and, with Transition, 4 hops for WAV and 6 hops for GCC traffic.

**Table 9: Results using Gray Encoding in a 16-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	- 32.07 %	37.16 mW	41.84 mW	14.29 mW	-
GZIP	- 0.45 %	43.16 mW	43.25 mW	15.5 mW	-
GCC	- 4.09 %	29.7 mW	29.99 mW	8.96 mW	-
Bytecode	- 13.42 %	37.9 mW	39.96 mW	13.66 mW	-
WAV	7.91 %	38.93 mW	37.63 mW	12.89 mW	10
MP3	- 0.01 %	43.02 mW	43.02 mW	15.4 mW	-
RAW	0.43 %	38.52 mW	38.45 mW	13.15 mW	192
BMP	4.95 %	35.11 mW	34.49 mW	11.27 mW	18
JPG	1.03 %	41.54 mW	41.34 mW	14.59 mW	74
TIFF	0.88 %	42.62 mW	42.45 mW	15.13 mW	85
PDF	- 1.16 %	42.82 mW	43.06 mW	15.39 mW	-

**Table 10: Results using Transition Encoding in a 16-bit flit width Hermes NoC.**

Stream	Reduction	A.P. NoC no Cod.	A.P. NoC Cod.	A.P. Cod. Modules	# of hops
HTML	- 2.22 %	37.15 mW	37.48 mW	11.77 mW	-
GZIP	- 0.4 %	43.16 mW	43.24 mW	13.51 mW	-
GCC	11.78 %	41.26 mW	39.05 mW	12.59 mW	6
Bytecode	- 2.42 %	37.9 mW	38.27 mW	11.99 mW	-
WAV	19.67 %	38.93 mW	35.7 mW	11.74 mW	4
MP3	0.05 %	42.98 mW	42.97 mW	13.44 mW	1183
RAW	- 11.41 %	38.52 mW	40.34 mW	12.4 mW	-
BMP	0.08 %	42.94 mW	42.92 mW	13.43 mW	826
JPG	- 0.72 %	42.78 mW	42.92 mW	13.4 mW	-
TIFF	1.8 %	42.62 mW	42.26 mW	13.28 mW	37
PDF	1.8 %	40.95 mW	40.62 mW	12.79 mW	39

## 7. Conclusions

This work investigated the reduction of power consumption in Networks-on-Chip through the reduction of signal transition activity using data coding techniques. Power macromodels for various NoC modules were built and embedded in a system-level model, which was simulated with a series of real and synthetic traffic.

Experiments have shown that the effectiveness of the coding is dependent of the transition activity patterns. The presented results point the direction for further research addressing the use of multiple coding schemes to better match the transition activity patterns, and the use of NoC configuration to help the decision whether a packet should be encoded or not. For instance, packets sent to neighbor cores must not be encoded. Also, encoded packets could carry an identification bit in their header.

It is important to point out that these results concern the NoC configuration used in this work, using 0.35μ technology. In all schemes, the power savings in inter-router channels are much smaller than in the router logic. However, in new technologies the power consumption in channels will be more relevant [14]. In that scenario, the encoding schemes may be advantageous, since they were developed to communication channels. Future work includes the

evaluation of coding schemes in NoCs implemented using state-of-the-art technologies, as well the proposal of a new encoding scheme more efficient for NoCs.

## 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] Burd. T and Brodersen, R. "Energy Efficient Microprocessor Design". Kluwer Academic Publishers, 2002. Pages: 376.
- [4] M. Pedram. "Power minimization in IC design". ACM Trans. Design Automat. Electron. Syst., vol. 1, no. 1, Jan. 1996.
- [5] J.C. Palma, C.A. Marcon, F. Moraes, N.L. Calazans, R.A. Reis, A.A. Susin. "Mapping Embedded Systems onto NoCs - The Traffic Effect on Dynamic Energy Estimation". In: 18th Symposium on Integrated Circuits and Systems Design - SBCCI 2005. New York: ACM Press, 2005. pp. 196-201.
- [6] 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, vol. 38, issue 1, pp. 69-93, October 2004.
- [7] L. Benini, A. Macii, E. Macii, M. Poncino, R. Scarsi. "Architecture and Synthesis Algorithms for Power-Efficient Bus Interfaces". Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 19, Issue 9, Sept. 2000 Page(s):969-980.
- [8] M. R. Stan, W. P. Burleson. "Bus-Invert Coding for Low-Power I/O". VLSI Systems, IEEE Transactions on Volume 3, Issue 1, March 1995 Page(s):49-58.
- [9] H. Mehta, R. M. Owens, M. J. Irwin. "Some Issues in Gray Code Addressing". GLS-VLSI-96, pp. 178-180, Mar. 1996.
- [10] Ramos, P.; Oliveira, A. „Low Overhead Encodings for Reduced Activity in Data and Address Buses". Em Proceedings of the International Symposium on Signals, Circuits and Systems, pp. 21-24. Julho, 1999.
- [11] L. Ost, A. Mello; J. Palma, F. Moraes, N. Calazans. "MAIA - A Framework for Networks on Chip Generation and Verification". ASP-DAC, Jan. 2005.
- [12] Bjerregaard, T.; Mahadevan, S. "A survey of research and practices of Network-on-chip". ACM Computing Surveys, v.38(1), 2006, pp. 1-51.
- [13] C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng. "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II,)" Technical Memorandum UCB/ERL M05/21, University of California, Berkeley, CA USA 94720, July 15, 2005.
- [14] Sylvester, D.; Chenming Wu; "Analytical modeling and characterization of deep-submicrometer interconnect". Proceedings of the IEEE. Volume 89, Issue 5, May 2001 Page(s):634 – 664.