Adaptive Coding in Networks-on-Chip: Transition Activity Reduction versus Power Overhead of the Codec Circuitry

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

[1] PPGC - II - UFRGS - Av. Bento Gonçalves, 9500, Porto Alegre, RS – Brazil
{jcspalma, reis}@inf.ufrgs.br
[2] MES – TU Darmstadt – Karlstr. 15, 64283 Darmstadt - Germany
{lsi, agarcia, glesner}@mes.tu-darmstadt.de
[3] PPGCC - FACIN – PUCRS - Av. Ipiranga, 6681, Porto Alegre, RS – Brazil
moraes@inf.pucrs.br

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

1 Introduction

The advances in fabrication technology allow designers to implement a whole system on a single chip, but the inherent design complexity of such systems makes it hard to fully explore the technology potential. Thus, the design of Systems-on-Chip (SoCs) is usually based on the reuse of pre-designed and pre-verified intellectual property core that are interconnected through special communication resources that must handle very tight performance and area constraints. In addition to those application-related constraints, deep sub-micron effects pose physical design challenges for long wires and global on-chip communication. A possible approach to overcome those challenges is to change from a fully synchronous design paradigm to a globally asynchronous, locally synchronous (GALS) design paradigm [1]. A Network-on-Chip (NoC) is an infrastructure essentially composed of routers interconnected by communication channels. It is suitable to support the GALS paradigm, since it provides asynchronous communication, scalability, reusability and reliability [2].

The growing market for portable battery-powered devices adds a third dimension (power) to the previously two-dimensional (speed, area) VLSI design space [3]. Power consumption is directly related to battery life as well as costly package and heatsink requirements for high-end devices [4]. In order to ensure the final system complies to the desired function, thermal and cost requirements, the power consumption issues must be addressed during the design of all subsystems in a SoC, including the interconnect structure. 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 consumes 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 we could 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. One way to reduce power consumption in NoCs, in both wires and logic, is to reduce the switching activity by means of coding schemes. [7]. Several schemes were proposed in the late 90's, all of them addressing bus-based communication architectures.

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

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

2 Coding 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 coding schemes have been previously proposed. Some of these schemes exploit spatial redundancy, increasing the number of bus lines, while others exploit temporal redundancy, increasing the number of bits transmitted in successive bus cycles. There are also few schemes that do not rely on spatial nor temporal redundancy [7].

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.

Bus-Invert [8] is an example of such schemes. It uses an extra control bit (an extra line) 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 [7] the authors propose an Adaptive Probability Encoding scheme which does not require any *a-priori* knowledge of the input traffic and its statistics. It is capable of on-line adaptation of the encoding to the stream statistic behavior. This scheme operates bit-wise, according to its actual and the last value.

The encoding is done with basis on approximate statistical information collected by observation of the bit stream over a window of fixed size S. The authors claim that a window size with S = 64 offers a good compromise between complexity and accuracy.

The statistical information concern the four joint probabilities for a single bit ($P_{0,0}$, $P_{0,1}$, $P_{1,0}$ and $P_{1,1}$). In order to deal with integer values and simplify the hardware, the authors use the occurrence frequencies N_{00}, N_{01}, N_{10} and N_{11} instead of probabilities. Since the sum of the four occurrence frequencies is known ($N_{00} + N_{01} + N_{10} + N_{11}$ = S -1), not all the four occurrences are actually required. Considering that N_{01} and N_{10} must be balanced over the observation window (they can differ by 1 at most), it is sufficient to consider only N_{00} and N_{11} joint probabilities, since their knowledge implies the other two. Considering $N_{01} = N_{10}$, the symbol N_T is used to denote them.

Such scheme uses four different encoding functions *F(x(n), x(n-1))*. The function selection is done according to the JPD (*joint probability distribution*) of the current window. The decision is taken as follows:

a) y(n) = x(n)	when $N_{00} > N_T > N_{11}$
b) y(n) = x(n)'	when $N_{11} > N_T > N_{00}$
c) y(n) = x(n) xor x(n-1)	when $N_T < \{ N_{11}, N_{00}\}$
d) y(n) = x(n) nxor x(n-1)	when $N_T > \{ N_{11}, N_{00}\}$

When the most probable pair of symbols is 00, it is reasonable to leave the bits unchanged, since a zero in the stream will result in no transition. Similarly, when N_T is the most probable symbol, the transitions are eliminated by XOR-ing two consecutive bits. This yields a sequence of ones that are complemented before being sent through the communication channel.

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). In Hermes [5], the header is composed by two flits[1], 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[2] through the Network-on-Chip.

By this way, encoding and decoding operations must be done in the source and target cores only, converting the original data to the encoded (and transmitted) one and vice-versa. In this work the encoder and decoder modules are inserted in the local ports of the routers, that is, between the cores and the NoC interconnect structure. It is important to say that this method is general, since it does not change the NoC structure.

Figure 1 shows an IP core connected to a local port of a Hermes router. The incoming data packets pass through the decoder module, while outgoing data packets pass through the encoder module. The remaining ports of the router must be connected to other routers according to the network topology.

[1] Smallest data unit transmitted over the network-on-chip.

[2] Hop is the distance between two routers in a network-on-chip.

This figure also shows the internal modules of a Hermes router: a switch control (SC) responsible for arbitration and routing functions and FIFO buffers for the input channel of each port (North, South, East, West and Local). The input and output channels are simplified in the figure, since they are composed of groups of signals. For example, the input channel is composed of *data_in*, *tx* and *ack_tx* signals in a handshake flow control configuration, or *data_in*, *tx* and *credt_in* in a credit-based configuration.

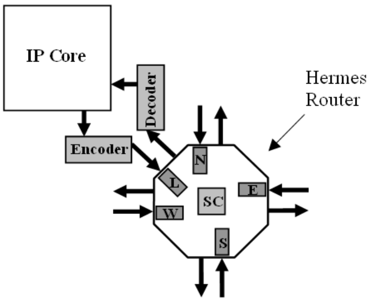


Figure 1 – Encoder and Decoder modules location.

4 NoC 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.

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.

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. However, until now in this work only the encoder and decoder configurations for 8-bit flit width were used.

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 [9]. Traffic input files are fed to the NoC through the router local channels, modeling local cores behavior. A VHDL simulator applies input signals to the NoC or to any NoC module, either a single router or 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 0.35µ. 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, the encoder and decoder modules were also developed in VHDL, synthesized to CMOS TSMC 0.35µ technology, converted to a SPICE netlist and simulated with different traffic patterns. In the case of Bus-Invert, it is necessary a new analysis of all NoC modules, since this scheme requires the insertion of a control bit in all modules, increasing the power consumption of them.

4.1 Model Definition

Average power per hop (*APH*) is used here to denote the average power consumption in a single hop of a packet transmitted over the NoC. *APH* can be split into three components: average power consumed by router (*APR*); average power consumed in a link between routers (*APL*); and average power consumed in 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.

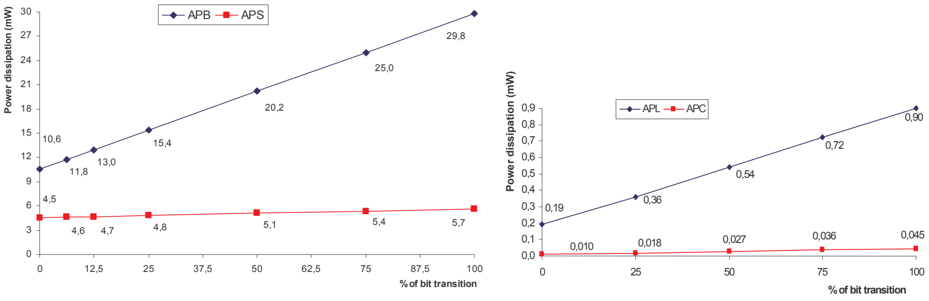
$$APH = APR + APL + APC \qquad (1)$$

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 2a illustrates this effect for a 16-word input buffer and for the centralized control logic of an 8-bit flit width Hermes router. The graph depicts power as a function of the amount of bit transitions in a 128-flit packet (100% = 127 bit transitions in each one of the 8 input bits). Clearly, power consumption increases linearly with the increase of bit transitions in a packet.

In regular tile[3]-based architectures (cores are placed inside a limited region, which is usually called tile), 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

[3] Cores are placed inside a limited region, which is usually called tile.

smaller than *APL*, as showed in Figure 2b.



(a) (b)

Figure 2 – (a) Analysis of the bit transition effect on the power consumption for a 16-word input buffer and for the centralized control logic of an 8-bit flit width Hermes router. (b) Analysis of the effect on local and inter-router links. Each tile has 5 mm x 5 mm of dimension, and uses 8-bit links. Data obtained from SPICE simulation (CMOS TSMC 0.35µ technology).

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 τi to tile τj, where η corresponds to the number of routers through which the packet passes.

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

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 3 illustrates the analysis of the effect of bit transitions on power consumption of the encoder and decoder modules, both with 8-bit width. As showed in this figure, power consumption also grows linearly in these modules as the amount of bit transitions is increased.

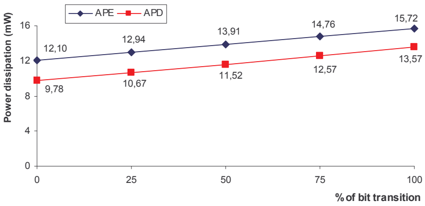


Figure 3 – Analysis of the effect of bit transitions on the power consumption of 8-bit encoder and decoder modules of Adaptive Encoding scheme.

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. For sake of simplicity, the macromodels estimate the average power consumption of a module without considering its internal state (power consumption is function of the transition activity on the module's inputs only). Such simplification incurs on error, but our experiments have shown that the error is relatively small. For instance, in the case of the encoder and decoder modules, the simplification leads to an error of 5% in the power estimation (about ±0,6 mW).

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 *Po* in the subsequent analysis (see Table 1). The remaining of the power consumption grows linearly with the bit transition rate (the slope is referred as *R*).

Table 1 – Macromodel for the NoC and Adaptive Encoding modules ($AP = Po + \%T * R$).

Module	***Po***	***R***
Buffer (*APB*)	10,61	19,19
Control (*APS*)	4,39	0,72
Encoder (*APE*)	12,1	3,62
Decoder (*APD*)	9,78	3,79
Inter-roter channel (*APL*)	0,19	0,71

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 bit of this scheme in all NoC modules (see Table 2).

Table 2 – Macromodels for the NoC and Bus-Invert modules ($AP = Po + \%T * R$).

Módulo	P_0	***R***
Buffer (*APB*)	11,49	22,13
Control (*APS*)	4,39	0,98
Encoder (*APE*)	1,16	3,88
Decoder (*APD*)	0,55	0,25
Inter-roter channel (*APL*)	0,19	0,8

Equation (3) computes the average router-to-router communication power dissipation, from tile τi to tile τj, passing through η routers and using a coding scheme.

$$CodedRRPij = APE + \eta \times (APB + APS) + (\eta - 1) \times APL + APD \qquad (3)$$

5 NoC Power Consumption Analysis

As stated in Section 4, the average power estimation depends on the communication infrastructure (here assumed to be a Hermes NoC) and on the application core traffic. 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. Some of these traffics are synthetic (generated by a traffic generation tool) and others 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 really large. 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 [10]. 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.

Considering for instance a synthetic traffic pattern with 2000 8-bit flits, the maximum amount of bit transition is 15992 (1999 x 8 = 100%). After the system simulation, the amount of bit transitions computed was 11994, corresponding to 80% of the maximum possible switching activity. Thus, the average power in the buffer and control logic must be calculated with base in the rate of 80% of bit transitions.

$$\begin{array}{lll} APB = Po + \%T * R & APS = Po + \%T * R & APL = Po + \%T * R \\ APB = 10{,}6 + 0{,}8 * 19{,}2 & APS = 4{,}39 + 0{,}8 * 0{,}72 & APL = 0{,}19 + 0{,}8 * 0{,}71 \\ APB = 25{,}96 \text{ mW} & APS = 4{,}97 \text{ mW} & APL = 0{,}76 \text{ mW} \end{array}$$

Using the Equation (2) it is possible to calculate the average power of a packet passing through only one router as follows:

$$RRP_{ij} = APB + APS + APL \quad \rightarrow \quad RRP_{ij} = 25{,}96 + 4{,}97 + 0{,}76 = 31{,}69 \text{ mW}$$

Considering now that using the Adaptive Encoding scheme the amount of bit transitions was reduced to 3998 (30%). In this case the energy consumption in the NoC is:

$$\begin{array}{lll} APB = 10{,}6 + 0{,}3 * 19{,}2 & APS = 4{,}39 + 0{,}3 * 0{,}72 & APL = 0{,}19 + 0{,}3 * 0{,}71 \\ APB = 16{,}36 \text{ mW} & APS = 4{,}6 \text{ mW} & APL = 0{,}4 \text{ mW} \end{array}$$

$$RRP_{ij} = 16{,}36 + 4{,}6 + 0{,}4 = 21{,}36 \text{ mW}$$

It is also necessary to include in this computation the power consumed by the encoder and decoder modules. In this example the original traffic pattern has 80% of bit transitions. In this case, the power consumption in the encoder module is:

$$APE = 12{,}1 + 0{,}8 * 3{,}62 \quad \rightarrow \quad APE = 14{,}99 \text{ mW}$$

However, the power consumption in the decoder module must be calculated with base in the encoded traffic with 30% of bit transition, as follows:

$$APD = 9{,}78 + 0{,}3 * 3{,}79 \quad \rightarrow \quad APD = 10{,}91 \text{ mW}$$

The power reduction in the Network-on-Chip in this example was 10,33 mW (31,69 - 21,36). On the other hand, the coding scheme consumes 25,9 mW (14,99 + 10,91). Considering a power save of 10,33 mW in each hop, we can assume that after 3 hops the power consumption of the coding scheme can be amortized, since the encoding and decoding occur only once in a transmission.

6 Experimental Results

This section presents experimental results obtained by system level simulation within Ptolemy II, using the macromodels described in Section 4, for a number of traffic patterns (both real application data and synthetically generated traffic).

Table 3 shows the results obtained with Adaptive Encoding scheme. The first column describes the type of traffic. The second column reproduces the reduction of transition activity reported by Benini et al. in [7] (when available) and the third column presents the reduction of transition activity found on our experiments by reproducing the coding techniques proposed in [7]. Both cases report the results in terms of reduction in the number of transitions with respect to the original data streams. The fourth column shows the power consumption (*APH*) without use of data coding techniques, while the fifth column shows the same measurement when coding techniques are used. Finally, the sixth and seventh 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.

The results presented in the second and third column show some differences between the results obtained in [7] and in the present work. This can be caused by the relation between consecutive flits sent through the NoC. In [7] the authors use a 32-bit flit channel. Thus, the transition activity is related to one byte and the fourth byte after it (sending 4 bytes in each flit). In this work we use 8-bit flit, and the transition activity is related to each byte and its consecutive one.

Experiments with the transmission of synthetically generated streams showed that the use of coding techniques can be advantageous if the average hop count for packages is larger than three (so the overhead of encoding and decoding is amortized). However, different examples like sound (.wav) streams showed that the encoding and decoding overhead can be amortized only after 21 hops (power reduction of 1,14 mW/hop). In some other cases, the encoder increases the number of transitions, increasing the power consumption of the system.

Table 4 presents the experimental results with 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 1 more bit in all modules).

Table 3 – Experimental results with Adaptive Encoding.

Stream	Bit transition reduction (reported in [7])	Bit transition reduction (simulated within the scope of this work)	NoC Average Power consumption without coding (*APB+APS+APL*)	NoC Average Power consumption with coding (*APB+APS+APL*)	Coding modules power consumption (*APE + APD*)	# of hops
HTML	9,3 %	5,5 %	18,26 mW	18 mW	23 mW	88
GZIP	16,3 %	8%	20,34 mW	19,7 mW	23,65 mW	37
GCC	15,6 %	4 %	19,93 mW	19,7 mW	23,5 mW	102
Bytecode	-	9 %	19,32 mW	18,93 mW	23,29 mW	61
WAV	2,1 %	21,95 %	20,36 mW	19,22 mW	23,52 mW	21
MP3	-	- 2,17 %	20 mW	20,11 mW	23,63 mW	-
RAW	- 3,4 %	- 10,98 %	18,7 mW	19,1 mW	23,2 mW	-
BMP	-	- 0,5 %	20,24 mW	20,27 mW	23,7 mW	-
Synthetic 1	-	75 %	22,62 mW	17,02 mW	23,52 mW	4
Synthetic 2	-	-27 %	18,45mW	19,34 mW	23,21 mW	-

Table 4 – Experimental results with Bus-Invert.

Stream	Bit transition reduction (simulated within the scope of this work)	NoC Average Power consumption without coding (*APB+APS+APL*)	NoC Average Power consumption with coding (*APB+APS+APL*)	Coding modules power consumption (*APE + APD*)	# of hops
HTML	6,2 %	18,26 mW	19,41 mW	2,32 mW	-
GZIP	18,7%	20,34 mW	20,93 mW	2,73 mW	-
GCC	17,9%	19,93 mW	20,68 mW	2,65 mW	-
Bytecode	12 %	19,32 mW	20,28 mW	2,5 mW	-
WAV	18,8 %	20,36 mW	20,93 mW	2,7 mW	-
MP3	17,4 %	20 mW	20,68 mW	2,66 mW	-
RAW	14,6 %	18,7 mW	19,56 mW	2,4 mW	-
BMP	18,2%	20,25 mW	20,86 mW	2,71 mW	-
Synthetic 1	50,7%	22,62 mW	20,3 mW	3,15 mW	1
Synthetic 2	2,45 %	18,45 mW	19,76 mW	2,36 mW	-

Table 4 shows that the power consumption in the NoC increases with the Bus-Invert scheme applied over real traffic, regardless of the fact that the bit transition was reduced. This is due to the inclusion of the extra bit in all NoC modules, increasing their power consumption. Even so, this scheme was efficient with one of the synthetic traffics. It is important to point out that these results concern the NoC configuration used in this work, using 0.35μ technology.

In both 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 [11]. In that scenario, the encoding schemes may be advantageous, since they were developed to communication channels.

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.

Our experiments have shown that the effectiveness of the coding is dependent of the transition activity patterns. By synthesizing a variety of traffic patterns, we could see that the power savings with Adaptive Encoding range from -1 to 15 mW/hop. However, the coding modules of this scheme present high power consumption. Bus-invert modules are economic in power consumption, but are not efficient in an 8-bit *flit* NoC.

Our 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, sending encoded packets to neighbor cores won't pay off). Encoded packets could carry an identification bit in their header.

Future work includes the evaluation of coding schemes in NoCs implemented using state-of-the-art technologies, where the power consumption in channels becomes more relevant due to relative increase in wire capacitance.

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] D. Singh, J. M. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal and T. J. Mozdzen. "Power conscious cad tool and methodologies: A perspective". Proc. IEEE, vol.83, pp. 570-594, Apr. 1995.

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

[10] 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.

[11] 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.