# Reducing the Power Consumption in Networks-on-Chip through Data Coding Schemes

José Carlos S. Palma[1], Leandro S. Indrusiak[2], Fernando G. Moraes[3], Ricardo Reis[1], Manfred Glesner[2]

[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, glesner]@mes.tu-darmstadt.de, 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 developed for bus-based systems and proposes a new coding scheme suitable for NoC-based systems. The estimation of the NoC power consumption was 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 were embedded in a system model and a series of simulations were performed, aiming to analyze the trade-off between the power savings due to coding techniques versus the power consumption overhead due to the encoding and decoding modules. The proposed coding scheme presented the best results for all of the simulated traffic patterns, when compared to other coding schemes found in the literature.**

## I. 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. 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 [1].

The growing market for portable battery-powered devices adds a new dimension, power, to the VLSI design space, previously characterized by speed and area. Power consumption is directly related to battery life as well as costly package and heatsink requirements for high-end devices [2]. 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 [3]. One 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.

The contribution of this work are two: (*i*) the evaluation of coding schemes in the context of NoC-based systems and the trade-off analysis of the power savings obtained by the application of such coding schemes versus the power consumption overhead due to the additional encoding and decoding circuitry, and (*ii*) the proposal of a new coding scheme suitable for NoC-based systems.

This paper is organized as follows. Section 2 reviews coding schemes aiming to reduce power consumption in bus-based systems and presents T-Bus-invert, a contribution of this work. Section 3 introduces the coding into NoC based systems. 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.

## II. 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(ex: Bus-Invert [4]), while others exploit temporal redundancy, increasing the number of bits transmitted in successive bus cycles (ex: T-Bus-Inv, proposed in this paper). There are also few schemes that do not rely on spatial nor temporal redundancy (ex: Adaptive Encoding [5], Gray [6] and Transition [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.

A problem common to many if not most works related to coding schemes is the lack of analysis concerning the power consumed by the coding modules. Also, these works do not present the relationship between the power consumed by the coding modules and the power consumed by the communication infrastructure. The proposed work contributes in this arena, presenting the power consumed by the coding modules, and the impact of power of these in the NoC consumption.

### A. *T-Bus-Invert*

T-Bus-Invert is a coding scheme that uses the basic principles of Bus-Invert coding, that is, inverting the data when the Hamming distance between the present value and the next data value is bigger than half of the number of lines. However, T-Bus-Invert does not require the insertion of an extra line in the communication channel. Such scheme uses the most significant bit as the control bit and sends, in each flit, *n*-1 valid bits considering an original flit with *n* bits. For example, using an 8 bits flit width NoC, this scheme sends, in the first flit, the 7 least significant bits from the original data, buffering the most significant bit. In the second encoded flit are sent the buffered bit concatenated with the 6 least significant bits from the original data, while the 2 most significant bits are buffered. This process continues until the seventh flit, where seven bits are buffered. In this moment, the encoder module must stop receiving new data during 1 clock cycle, while sends the buffered bits as a new encoded flit.

T-Bus-Invert transforms the spatial redundancy of Bus-Invert in a temporal redundancy (Temporal Bus-Invert). One possible disadvantage of the T-Bus-Invert scheme is the reduction of the maximum throughput at the IP side, since a latency of one clock cycle after each group of seven consecutive flits is inserted. Actually, this is not a drawback, since IP transmission rates are inferior to the channel rate. For example, in an 8-bit 50 MHz router, the available bandwidth per channel is 376 Mbps. Using T-Bus-Invert, this bandwidth decreases to 344 Mbps. In contrast, the rate of an application requiring a large amount of bandwidth, such as an HDTV stream (MPEG2), is 15 Mbps.

## III. Adding Coding Modules into NoCs

In Networks-on-Chip, the data is transmitted as packets which are sent through routers, from one source to one target core. These packets are composed of a header (containing routing information) and a payload (containing the data to be transmitted). In the case of Hermes [8], used here as a case study, the header is composed of two flits[1], comprising the target address and the packet length. Thus, such approach merging encoding schemes and a Network-on-Chip does not encode the packet header, since it must be used by routers in every hop[2] through the Network-on-Chip.

Thus, encoding and decoding operations must be done in the source and target cores only, so to convert the original data to the encoded (and transmitted) one and vice-versa. In this work the encoder and decoder modules were inserted in the local ports of the routers, that is, between the integrated cores and the NoC interconnect structure.

## IV. 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.

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. Considering that, in regular tile-based architectures, tile dimension is close to the average core dimension, and the core inputs/outputs are placed near the router local channel, the power consumed in the channels connecting a router to its local core may be safely neglected without significant errors in total power dissipation.

Equation (1) 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. APB, APS and APL correspond to the power consumed in Buffers, Control logic and inter-router Links, respectively.

$$RRPij = \eta \times (APB + APS) + (\eta - 1) \times APL \quad (1)$$

Considering now an approach with data coding in the NoC local ports, two new parameters may be introduced to Equation (1): APE and APD (encoder and decoder average power consumption, respectively), producing the Equation (2).

$$\begin{aligned} CodedRRPij = APE + \eta \times (APB + APS) + \\ (\eta - 1) \times APL + APD \end{aligned} \quad (2)$$

Based on the described analysis, it is possible to build macromodels for the several parts of the proposed model –

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

[2] hop is the distance between two adjacent routers in a network-on-chip

APB, APS, APL, APE and APD – representing the power consumption in the different modules of a NoC, as shown in Table 1. 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. The parameters of the macromodel were acquired after the SPICE simulation of the communication infrastructure and encoding modules with different traffic patterns.

TABLE 1: MACROMODEL FOR AN 8-BIT FLIT WIDTH 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 |

## V. 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: 2D mesh topology, wormhole packet switching, XY deterministic routing, input buffering.

In order to fully evaluate the traffic resulting from real application data, experiments must be performed with realistic amounts of data. For example, to simulate the transmission of 5 seconds of sound in "wav" format it is necessary to send 980.000 8-bit flits over the NoC. 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 [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.

## VI. 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. Table 2 presents the results obtained with T-Bus-Invert coding scheme in an 8-bit flit width Hermes NoC. 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 in a single hop (APB + APS + APL) without use of data coding techniques, while the fourth column shows the same measurement when T-Bus-Invert coding technique is used. 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.

As presented in Table 2, T-Bus-Invert is effective with all of the simulated traffics. The coding power overhead can be amortized after 3 hops in the best case and after 12 hops in the worst case.

Table 3 shows the results obtained with five different coding schemes in an 8-bit flit width Hermes NoC: Adaptive Encoding, Bus-Invert, Gray, Transition and T-Bus-Invert. The results correspond to the reduction of transition activity and the number of hops needed to amortize the coding power overhead. 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 coding power overhead can be amortized after 11 hops.

TABLE 2 – RESULTS USING T-BUS-INV CODING IN AN 8-BIT FLIT WIDTH HERMES NOC.

| Stream | Transition Reduction | A.P. NoC no Cod. | A.P. NoC Cod. | A.P. Cod. Modules | # of hops |
|---|---|---|---|---|---|
| HTML | 9,8 % | 22,24 mW | 21,55 mW | 8,27 mW | 12 |
| GZIP | 26,89 % | 25,5 mW | 22,73 mW | 9,45 mW | 3 |
| GCC | 26,35 % | 24,68 mW | 22,18 mW | 9,11 mW | 4 |
| Bytecode | 20,88 % | 23,56 mW | 21,81 mW | 8,72 mW | 5 |
| WAV | 28,19 % | 24,45 mW | 21,84 mW | 8,99 mW | 3 |
| MP3 | 27,09 % | 25,3 mW | 22,57 mW | 9,36 mW | 3 |
| RAW | 14,5 % | 22,24 mW | 21,22 mW | 8,22 mW | 8 |
| BMP | 26,38 % | 25,3 mW | 22,63 mW | 9,37 mW | 4 |
| JPG | 26,26 % | 25,06 mW | 22,47 mW | 9,28 mW | 4 |
| TIFF | 27,68 % | 25,26 mW | 22,47 mW | 9.34 mW | 3 |
| PDF | 30,57 % | 26,06 mW | 22,74 mW | 9,62 mW | 3 |

With Bus-Invert the power consumption increases, 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. 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%). Similar to Adaptive Encoding, Gray and Transition schemes are effective with some traffic patterns. The best case for power overhead amortization is 7 hops with Gray and 6 with Transition.

The best results were obtained with T-Bus-Invert. Such scheme uses the advantage of the ample transition reduction provided by Bus-Invert scheme without inserting an extra control bit. As a result, the power reduction is significant. The number of hops needed to amortize power coding overhead is only 3 with several traffic patterns.

Similar to Table 3, Table 4 presents the results obtained with different coding schemes in a Hermes NoC, but now using 16-bit flit width. Observe that Adaptive Encoding was implemented only with the 8-bit NoC, since its encoder and decoder modules consume too much power when compared to other coding schemes and compared to NoC power consumption. Bus-Invert with 2 clusters was added in the 16-bit NoC, instead of Adaptive Encoding. Columns from 2 to 5 show 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 [4], the first approach, with clusters of 8 bits, is more efficient with respect of transition activity reduction. Nevertheless, in NoC based systems, 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 compensated by its transition activity reduction.

Both scenarios using Bus-Invert are not efficient for some traffic patters. With most of the simulated traffics, the number of hops to amortize the power overhead is between 15 and 18 in the first scenario and between 4 and 6 in the second scenario.

As in the approach with 8-bit flit width, Gray and Transition schemes are effective only with some traffic patterns, but the number of hops necessary to amortize the coding power overhead is high.

Again, the most efficient coding scheme is T-Bus-Invert, reducing the power consumed in the NoC with all traffic patterns. Such scheme uses the advantage of the ample transition reduction provided by Bus-Invert scheme without inserting an extra control bit. As a result, the power reduction is significant. With most of the simulated traffic patterns, the power overhead is amortized after 3 or 4 hops.

TABLE 3 – RESULTS OBTAINED WITH DIFFERENT CODING SCHEMES IN AN 8-BIT FLIT WIDTH HERMES NOC.

| Stream | Adaptive Encoding | | Bus-Invert | | Gray | | Transition | | T-Bus-Invert | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops |
| HTML | -1,4 % | - | 6,2 % | - | - 11,36 % | - | - 2,97 % | - | 9,8 % | 12 |
| GZIP | 1,03% | 240 | 18,7% | - | - 0,49 % | - | 1,17 % | 61 | 26,89 % | 3 |
| GCC | 0,91 % | 292 | 17,9% | - | - 5,57 % | - | 2,38 % | 31 | 26,35 % | 4 |
| Bytecode | 9,3 % | 32 | 12 % | - | 0,21 % | 331 | 9,22 % | 9 | 20,88 % | 5 |
| WAV | 21,95 % | 11 | 18,8 % | - | - 0,01 % | - | 12,22 % | 6 | 28,19 % | 3 |
| MP3 | - 2,41 % | - | 18,48 % | - | - 0,06 % | - | - 0,21 % | - | 27,09 % | 3 |
| RAW | - 10,98 % | - | 14,6 % | - | 4,11 % | 19 | - 12,44 % | - | 14,5 % | 8 |
| BMP | - 0,5 % | - | 18,2% | - | 11,82 % | 7 | - 0,02 % | - | 26,38 % | 4 |
| JPG | 0,8 % | 327 | 19,5 % | - | 3,42 % | 19 | - 0,6 % | - | 26,26 % | 4 |
| TIFF | -1,16 % | - | 18,3 % | - | - 1,36 % | - | - 1,16 % | - | 27,68 % | 3 |
| PDF | 6,61 % | 36 | 23,15 % | 13 | 3,5 % | 18 | 7,52 % | 9 | 30,57 % | 3 |

TABLE 4 – RESULTS OBTAINED WITH DIFFERENT CODING SCHEMES IN A 16-BIT FLIT WIDTH HERMES NOC.

| Stream | Bus-Invert 2 clusters | | Bus-Invert 1 cluster | | Gray | | Transition | | T-Bus-Invert | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops | Tr. Reduction | # of hops |
| HTML | 11,73 % | - | 10,57 % | - | - 32,07 % | - | - 2,22 % | - | 18,47 % | 6 |
| GZIP | 22,83 % | 17 | 19,59 % | 6 | - 0,45 % | - | - 0,4 % | - | 32,17 % | 3 |
| GCC | 20 % | - | 16,86 % | 13 | - 4,09 % | - | 11,78 % | 6 | 29,48 % | 3 |
| Bytecode | 17,08 % | - | 12,89 % | - | - 13,42 % | - | - 2,42 % | - | 25,39 % | 4 |
| WAV | 29,04 % | 6 | 22,69 % | 5 | 7,91 % | 10 | 19,67 % | 4 | 35,7 % | 3 |
| MP3 | 23,14 % | 15 | 19,9 % | 6 | - 0,01 % | - | 0,05 % | 1183 | 31,83 % | 3 |
| RAW | 21,41 % | - | 17,2 % | 16 | 0,43 % | 192 | - 11,41 % | - | 25,18 % | 4 |
| BMP | 22,81 % | 18 | 19,19 % | 6 | 4,95 % | 18 | 0,08 % | 826 | 31,53 % | 3 |
| JPG | 23,27 % | 15 | 19,99 % | 6 | 1,03 % | 74 | - 0,72 % | - | 31,13 % | 3 |
| TIFF | 23,33 % | 15 | 20,75 % | 5 | 0,88 % | 85 | 1,8 % | 37 | 32,68 % | 3 |
| PDF | 23,99 % | 11 | 21,52 % | 4 | - 1,16 % | - | 1,8 % | 39 | 32,38 % | 3 |

## VII. CONCLUSIONS

Experiments have shown that the effectiveness of the coding is dependent of the transition activity patterns. The coding scheme proposed in this work presented best results for all traffic patterns, when compared to other coding schemes found in the literature.

The presented results point the direction for further research addressing 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. In that scenario, the encoding schemes may be advantageous, since they were developed to communication channels.

## 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] Burd. T and Brodersen, R. “Energy Efficient Microprocessor Design”. Kluwer Academic Publishers, 2002. Pages: 376.

[3] Omitted for blind review.

[4] 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.

[5] 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.

[6] H. Mehta, R. M. Owens, M. J. Irwin. “Some Issues in Gray Code Addressing". GLS-VLSI-96, pp. 178-180, Mar. 1996.

[7] 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.

[8] 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.

[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.