Published in IET Computers & Digital Techniques
Received on 31st March 2005
Revised on 2nd May 2007
doi: 10.1049/iet-cdt:20050060

ISSN 1751-8601

Low-power and error protection coding for network-on-chip traffic

A. Vitkovski[1] *A. Jantsch*[2] *R. Lauter*[2] *R. Haukilahti*[2] *E. Nilsson*[2]

[1]*Advanced Research Centre for Electronic Systems, University of Bologna, Viale Pepoli 3/2, 40123 Bologna, Italy*
[2]*Department of Microelectronics and Information Technology, Royal Institute of Technology, Electrum 229, SE-164 40 Kista, Sweden*
E-mail: arseni@kth.se

Abstract: The power consumption of the network-on-chip communication backbone is explored and the effectiveness of low-power encoding and error protection techniques is analysed. For the switch under the study, a Nostrum deflective routing switch, simulations and power analysis suggest that only a minor fraction of the power is dissipated in the logic blocks, whereas the major part is due to the interconnection wires. The authors have investigated a number of low-power and data protection mechanisms and studied their impact on power consumption of the whole network. The bus-invert encoding scheme and a limited set of Hamming data protection codes have been implemented on both data link and at the network layer. However, it turned out that all low-power data encoding schemes have little potential to decrease power consumption due to the significant overhead. On the other hand, error protection mechanisms have a significant potential to decrease power consumption because they allow to operate the network at a lower voltage. The authors' experiments show a 20% decrease of power consumption for a given error rate and for a given performance.

1 Introduction

Until today shared on-chip buses are the dominant way to implement on-chip communication. However, buses have limited scalability and constitute a serious bottleneck for SoCs with tens or hundreds of processing elements. One way to overcome this limitation is to separate computation from communication and establish a communication infrastructure. Computational resources of a system will communicate by means of global interconnection architecture, called network-on-chip (NoC) [1, 2]. We have chosen Nostrum NoC [3] as target architecture for our experiments. It is based on a regular $m \times n$ mesh of the resources, connected between each other by means of bidirectional network with the packet switching mechanism (Fig. 1).

Power consumption will be a key limitation factor in future NoCs, where the power is dissipated in two components: (i) the switches, located in the cross-points of the network and (ii) the interconnect wires. These wires consume significant part (up to 50% [4]) of the energy in an integrated circuit, and its share is expected to grow in future. For the NoC this parameter is even more critical because a major part of the communication network consists of wires.

There are two main sources of power consumption of on-chip interconnection wires. The first one is due to the coupling capacitance of the adjacent lines. Its value depends on the power supply voltage, intensity of the data streams in the wires, direction of data transmission and mutual correlation. The second source comes from capacitance between the wire and the ground substrate. It is also influenced by the power supply voltage as well as the order of bits transmitted on the line. For each source of power consumption, there is a series of appropriate low-power techniques. Such mechanisms as wire shielding by means of insertion of non-data line between two value-carrying lines, wire permutation etc. belong to the first class of the

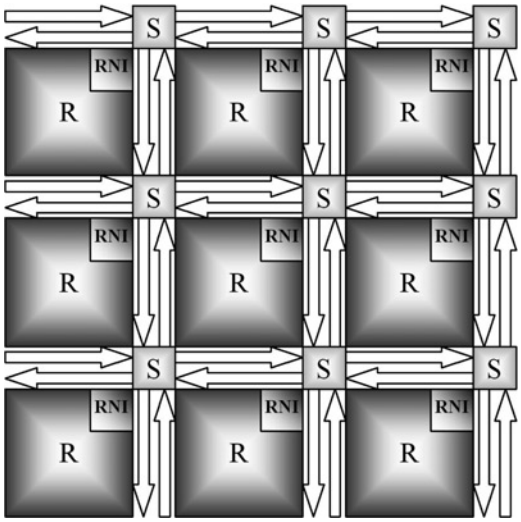

Figure 1 *3 × 3 network-on-chip*

low-power schemes. The second class employs special data encoding and decoding to reduce the bit-switching activity as well as the number of the transmitted bits on the interconnection line. There is a number of coding schemes presented in the literature such as bus-invert coding, codebook based coding, Gray code, working-zone coding etc. The Nostrum architecture already provides the shielding mechanism of the interconnection wires. However, this is not enough for having a reasonable value of power dissipation especially for large network sizes and the necessity of further improvement emerges.

As technology scales, on-chip interconnects become more sensitive to noise sources such as power supply noise, crosstalk and radiation-induced effects, that are likely to reduce the reliability of data [5]. To increase reliability, several solutions for data protection have been proposed [5, 6]. However, their overheads are still unexplored. Furthermore, employment of data protection mechanisms can save power, since stronger protection allows to decrease the power supply voltage. A lower voltage will result in some signalling errors, which can be tolerated due to the error protection [7]. Again a number of possible coding mechanisms for data protection have been investigated. Among the linear block codes such as parity code, Hamming code, Hsiao code, cyclic code etc., we have selected a Hamming code because it has relatively simple encoder/decoder logic suitable for hardware implementation and still a reasonable protection level. Different variations have been implemented for different parts of the transmitted data packet because head and payload have to be handled and protected differently.

Another important issue is the choice of the protocol layer where the technique is implemented. Realisations at the data link layer, the network layer or the transport layer will lead to very different benefit–cost trade-offs. For instance, data link layer protection requires that encoders/decoders are replicated at every link and the packet has to be encoded/decoded several times on its route through the network. In contrast network and transport layer protection require only one encoding at the source and one decoding at the destination. However, codes for end-to-end protection must be stronger because errors have more time to accumulate along the path. Consequently, we examined different variants of the techniques under study, performed a large number of simulations with various parameters, and made thorough analysis of the results in order to identify the optimal trade-off points.

The paper is organised as follows. Section 2 presents description of models and techniques used in the experiments. Section 3 lists the examined configurations and describes simulation and power estimation frameworks. Section 4 presents the setup for simulation experiments. Section 5 introduces experimental results and gives their analysis. Section 6 discusses the related work, and Section 7 concludes the paper.

2 Description of models and techniques

2.1 Nostrum NoC simulator

The Nostrum [3, 8] simulator is based on a regular $m \times n$ mesh of switches S (Fig. 1). Switches are the basic building blocks of the NoC backbone; they relay all traffic through the network. The switches themselves are bufferless and use hot-potato routing. Each of them is connected to the neighbour switches by means of 128 bidirectional data lines.

The signalling in the network is packet based. Each data packet consists of a 32 bit header and a 96 bit payload. The header holds the service information needed for switching packets over the network. The payload holds the actual data. The header includes the following data fields: virtual circuit bit, 12 bit destination address, 12 bit source address, empty bit and 6 bit hop counter. Since the size of the message can vary and a packet has fixed size, one or more packets are needed to transmit one message.

The Nostrum simulator models just the communication part of NoC. Resources R constitute the computation part. A resource can be any IP block like processor, DSP, FPGA, any kind of memory etc. Every resource is connected to one switch by means of a resource network interface (RNI).

In the original Nostrum simulator only the data link and the network layers were realised, but the physical layer was not modelled. Therefore we had to build the simplified physical model of network interconnections in the context of this work. In addition, the simplified behavioural model of the resources on the network abstraction layer has been developed which were charged with end-to-end communication.

doi: 10.1049/iet-cdt:20050060

2.2 Physical model of the interconnection wires

As has been mentioned in Section 1, we consider the two main sources of power consumption in the on-chip interconnections. They are: (i) the coupling capacitance between the adjacent wires and (ii) the capacitance between wire and ground. The effectiveness of wire shielding in the Nostrum architecture has been studied previously [9, 10].

The following equation describes the power, consumed on interconnection wires between two neighbour switches

$$P_{\text{Net}} = \frac{1}{2}\alpha C_{\text{Net}} V_{\text{DD}}^2 f \quad (1)$$

where C_{Net} is the capacitance of the interconnection wires, α the average switching activity on the wires V_{DD} the power supply voltage and f the clock frequency. The switching activity is defined as the ratio of the number of switched bits ($0 \rightarrow 1$ or $1 \rightarrow 0$) to the total number of bits, passed through a wire over a period of time.

Although the values of C_{Net}, V_{DD} and f depend on the technology process, the value of switching activity strongly depends on the data, transmitted through the wires. Linear dependency between power consumption and switching activity allows achieving power savings by reducing switching activity.

2.3 Network layer model of the resource

This model aims to simulate abstract resource behaviour as far as it contributes to the communication traffic. It enables data message exchange between resources through the network. In such way, we can fill the Nostrum network with data.

A resource divides the data to be communicated into messages. Then it determines the receiver for each message. After that, the message is split into smaller packets with predefined length and structure, which are sent to the consumer through the network. All decisions throughout this process are made taking into consideration the following parameters:

- Data type
- Resource mapping
- Packetisation and
- Emission probability

The data type determines which kind of data should be used by the resource to be sent. This parameter aims to simulate data streams of real applications. It can be an operating system, MP3, video, JPEG encoding/decoding etc. Thus, special text files as well as compressed audio and video streams are used as pseudo-real traffic inside the network. Random data were used to generate random traffic. It served as a reference point.

Before splitting the data message into packets, the sender resource chooses a receiver. In this way, we can simulate different resource mappings. It could be a single or multiprocessor system, with various memory structures, and any number of co-processors, controllers etc. Possible mappings are single sender–single receiver, single sender–multiple receiver, multiple sender–single receiver and multiple sender–multiple receiver. The sender and receiver resources may be predefined or random.

Packetisation splits a message into packets, which are the elementary data unit of the network layer. Since the network packet has the fixed length, packetisation depends on the size of the whole message. This parameter allows simulating the length of communication burst between the resources.

Emission probability is the probability that the resource will start the transition process in the current clock cycle. This parameter directly influences the overall network load. Since in the real system in a given period of time some resources might consume more or less data than others, the emission probability defines how active the resource is.

2.4 Low-power encoding

Our low-power technique addresses the power, which is consumed due to the wire capacitance as described in Section 2.2. Consequently, the target of the low-power technique is to minimise the switching activity on the interconnection wires.

There are many schemes presented in the literature aimed to reduce switching activity, such as bus-invert coding, codebook-based coding, Gray code, working-zone coding etc. After a comparative analysis, which is beyond the scope of this paper, we decided to implement bus-invert code because of the following reasons:

- it is optimal for the compressed media data [11], which we assume to be an important data type;
- it has relatively simple implementation with small overhead and does not require large memory for the dictionary such as the codebook-based schemes;

The bus-invert scheme was implemented as a separate block of encoder and decoder. One block is connected to each of the four sides of the switch. The data are transmitted from a switch-sender to the appropriate encoder, goes through the interconnection link and then, after the decoder, the original signal comes to a switch-receiver. This implementation gives good reusability, allowing the exchange of the encoder/decoder blocks

without changing the switch functionality. On the other hand, it makes it possible to estimate overheads of each block individually.

The functionality of the bus-invert coding is the following. In the encoder, the current data packet is compared with the previous one. If the number of switched bits exceeds half of all bits of the packet then all the bits of the current packet are inverted and an additional bit is set high. In the decoder, the additional bit is checked. If it is high, the current packet is inverted; otherwise, it is transmitted to the switch as it is.

There might be several options in the implementation. First, this technique can be applied to just a part of the data packet. This will reduce some overheads concerning with multiplexing of additional bits. However, the effectiveness of the technique will decrease more or less depending on which part of the packet was chosen. Second, the packet can be split into several sections, which are encoded/decoded separately [partial bus-invert (PBI) encoding]. This option will increase the efficiency and the speed of the technique. On the other hand, we will have the overhead of additional lines because each section requires one additional bit. The more parts we have, the more wires we need to add, which finally leads to diminishing returns.

In our case, after performing a series of experiments, we decided to settle on two variants of bus-invert encoding scheme:

- encode the payload as a single section adding a single line – P1BI encoding;

- split the payload into two sections and encode them separately adding two lines – P2BI encoding;

Apart from building the low-power block, some changes were made inside the switch aiming to reduce its power consumption. In the packet header, the addressing type of the address fields was changed from two's complement to one's complement. This allowed reducing the switching activity of the address bits and, hence, slightly improving total power consumption.

2.5 Error protection encoding

Error protection mechanisms were implemented for an 8×8 network with links that consist of 80 payload bits, 8 address bits, 6 hop counter bits and 1 empty bit; in total 95 bits. For other network sizes, the address and hop counter fields are adjusted but in all scenarios the useful payload is 80 bits for easy comparison. We implement error protection techniques on a link layer and network layer in order to compare their efficiency.

Link layer protection uses a block code with 20 payload bits and 5 protection bits in each block and with double-error detection (DED) and single-error correction (SEC) capability [6]. The payload (address and hop counter) is also protected. The empty bit is not protected because the receiving switch must know immediately if the packet is valid or not and not only after the decoding stage. We assume that the empty bit is protected by other means, such as better shielding or stronger drivers. Thus, for an 8×8 network we use 20 bits for protecting the payload and 10 bits for header protection in addition to the 95 bits; 125 bits in total. This code has been chosen for an efficient hardware implementation that requires a few hundred gates only and incurs a low delay of <10 gates on the critical path [6]. A drawback is a higher number of inter-switch wires: 125 wires as compared with 95 without error protection and 119 with end-to-end protection.

In the end-to-end error protection, we can use stronger codes for the payload to compensate the accumulation of errors over multiple hops. For the header this is not possible, because if the address is corrupted, the packet will not be delivered correctly. Hence, we have to make sure that every packet is properly routed through the network, which requires an uncorrupted header after each hop. The payload protection code is a block code covering all 80 bits in a single block with 14 protecting bits. It is also an SEC/DED code, but it is stronger than the link layer code because it requires fewer redundancy bits. Altogether, 119 bits for both header and payload are required $(80 + 14 + 24 + 1)$. The implementation is much more expensive; it requires $\sim$10 000 gates and incurs a delay several times higher than the code in link layer. This is justified since the encoder/decoder exists only once per resource, whereas link layer implementation needs four encoder/decoder per switch. In addition, here a packet is delayed by the encoder/decoder only once and not once per hop as above. In this experiment we only take the power consumption of the links into account, but not the encoders, decoders and switches, because the links by far dominate the power consumption as observed in the previous configurations [7].

3 Methodology

3.1 Examined configurations of the Nostrum NoC simulator

To investigate an influence of the low-power technique, the following configurations were chosen for further examination. They all are based on the Nostrum NoC simulator. Additional design blocks are implemented in VHDL on the RTL level.

Configuration A. This is the baseline configuration. It consists of the switches S and four adjacent interconnection links (Fig. 2) that communicate with the neighbour switches. The grey colour marks the blocks belonged to the same configuration.

doi: 10.1049/iet-cdt:20050060

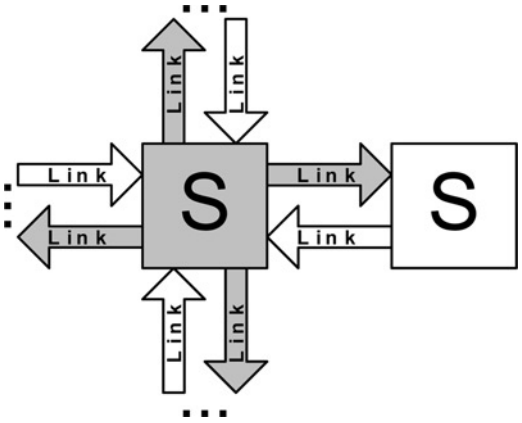

Figure 2 *Reference configuration*

Configuration B. This includes four low-power blocks with P1BI encoder/decoder (Section 2.4) at each side of the switch. They consist of the encoder Enc and the decoder Dec (Fig. 3). The data packet is transmitted from a switch-sender to the appropriate encoder, goes through the link in code and then, after the decoder, the original signal comes to a switch-receiver. Thus, in the link we have coded data with the switching activity different from the initial one.

Configuration C. This is similar to the previous one but here we use P2BI encoding/decoding mechanism as the low-power block (Section 2.4). In all other respects, the functionality is the same (Fig. 3).

For the data-protection mechanism, we compare the next three scenarios:

Scenario 1 is the reference with no error protection in place (Fig. 2). Again, here we use the pure Nostrum NoC model without any additional blocks as in Configuration A.

Scenario 2 uses error protection at the link layer, which also consists of encoder Enc and decoder Dec by each side of the switch (Fig. 3). The data protection mechanisms for Scenarios 2 and 3 are described in Section 2.5 in detail.

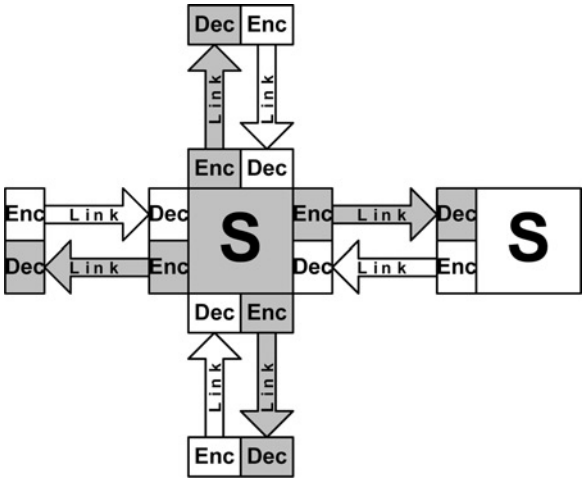

Figure 3 *Link level configuration*

Scenario 3 deploys end-to-end error protection for the payload (Fig. 4). Therefore the encoder/decoder block is placed inside RNI of the corresponding resource. However, the header is still protected on the link layer and encoded/decoded between two switches just as in Scenario 2.

3.2 Simulation experiments on data streams

To measure and compare the power consumption of the NoC for different configurations, the series of simulation experiments were performed. Fig. 5 shows dataflow of the simulation process, which consists of four stages.

During the first stage, according to the input parameters the resource model RM (Section 2.3) generates data messages, split them into the data packets and sends them to the Nostrum NoC simulator SM.

On the second stage, the NS (Section 2.1) performs simulation by sending prepared packets from their sources to the destinations and saves information about the data streams, transmitted through each interconnection bus in the output file – data streams in the network. On this stage, the researcher may set mesh size, simulation time and other characteristics of NS, which are described in Section 2.1. In addition, NS can have any examine configuration described in Section 3.1 depending on the technique is being simulated.

During the third stage, the data streams analyser DSA estimates network load and switching activity using the information, saved in the previous stage. Network load is estimated for each clock cycle and defined as $L_{\text{Net}} = N/N_{\max}$, where N is the number of packets, available in the network in the current clock cycle and $N_{\max}$ is maximum possible number of packets in the network: $N_{\max} = 4 \times m \times n$. Data switching activity shows the ratio between number of switched bits and number of available bits: $\alpha = n_{\text{sw}}/n_{\text{pass}}$.

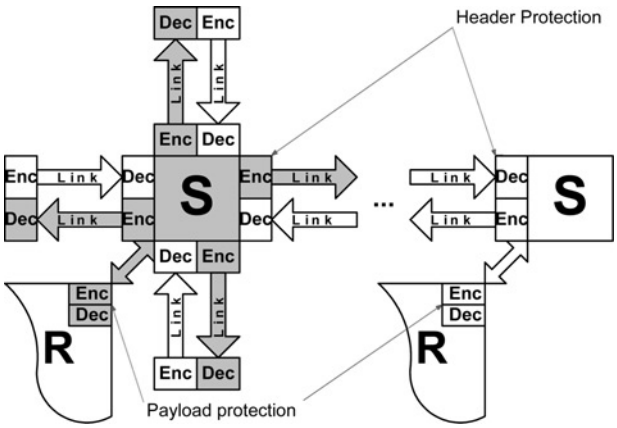

Figure 4 *End-to-end level configuration*

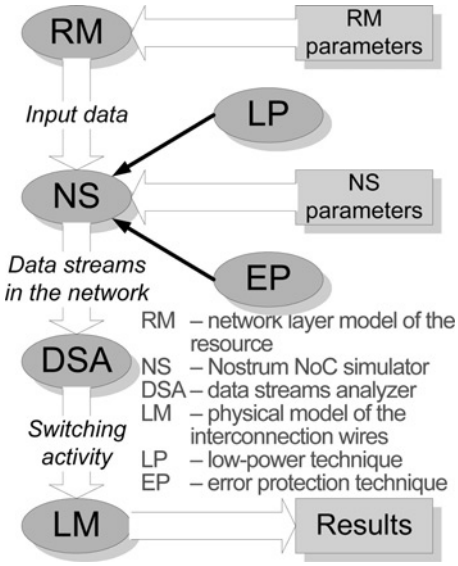

Figure 5 *Simulation dataflow*

This parameter is estimated as an average for the whole network and all cycles. The output results are presented both in numerical and visualised graphical formats.

Having the value of switching activity inside network interconnections, physical links model LM calculates power consumption using the methods from Section 2.2.

Thus, as the simulation result, we have values of the power, consumed on the interconnection wires. For different examined configurations, these values are different. However, they do not consider the overheads due to implementation of low-power and error protection techniques. The following chapter discusses further power estimations.

3.3 Power estimations

To estimate the overall power consumption, taking into account overheads, performance, technology process etc., the following framework was used.

1. The highest possible work frequency of the examined configuration is estimated in Synopsys Design Analyser.

2. Power consumptions of the design blocks of the examined configuration (switch, encoders and decoders) are estimated in Synopsys Power Compiler.

3. All values of the clock frequency and power (including power consumption of the design blocks and the links) are scaled to the 65 nm technology.

The overall power consumption equals to the sum of power consumed in the design blocks and network interconnections scaled from the 180 to 65 nm technology. The figures and scaling methodology was taken from International Technology Roadmap for Semiconductors [12].

Since the several different techniques with various operating frequencies will be examined, it is essential to normalise their values to the lowest one in order to have the same throughput. Furthermore, as the frequencies will be scaled to one value, the supply voltage should be scaled as well. In other words, all values were normalised in order to be able to compare the results. The dependency between the clock period t_{inv} and the power supply voltage V_{dd} is presented in [13]

$$t_{inv} = \frac{L_d \times K_6}{(V_{dd} - V_{th})^{\alpha}} \quad (2)$$

where L_d is logic depth of the path, K_6 a constant for a given process technology, α a measure of velocity saturation and V_{th} is the threshold voltage.

4 Experimental setup

There are two types of parameters used in simulation experiments. First type refers to the resource model RM and the second refers to Nostrum NoC simulator NS. RM parameters allow manipulating with data streams sent into the network. They are data type, resource mapping, packetisation and emission probability (Section 2.3). NS parameters characterise the NoC architecture modelled by Nostrum simulator (Section 2.1). The main ones of them are the simulated mesh size and simulation time in clock cycles. Table 1 presents their chosen values for the case study.

There are two data types: pseudo-real traffic and random traffic. In pseudo-real simulation, a compressed MP3 data is used as a payload of the data packet. For each resource, one fixed length MP3 file is associated. The information is read from the file sequentially, starting from the beginning of the file. For the random traffic, payload has a randomly distributed data. Performed experiments show that the behaviour of the pseudo-real traffic is very close to the random. That is why the following simulations will be preceded with the random traffic.

Mesh size of the NoC backbone determines the number of switches used in current simulation. The experiments were

Table 1 Experimental setup

Simulation parameters	Values							
Data type	Pseudo-real traffic (MP3)				Random traffic			
mesh size	5 × 5		8 × 8		5 × 5		8 × 8	
packetisaton	1	10	1	10	1	10	1	10
emission probability	five to seven values such that network load hits the range 20–70%							

doi: 10.1049/iet-cdt:20050060

performed with up to 8 × 8 meshes to provide generality of the results. Packetisation splits the message into packets. A packetisation of one means that the message consists of just one data packet. Typical messages consist of approximately ten packets in our case study.

Emission probability is the probability that the resource will start the transition process in the current clock cycle. The extreme values of emission probability fewer than 20% and over 70% were not considered since in these cases the network is either empty or overloaded. Five to seven values of emission probability within the range of 20–70% are enough to build a reasonably accurate trend.

The communication pattern stabilises after a few hundred cycles. Hence, 1000 cycles of simulation time is sufficient to observe steady-state behaviour. Network load and switching activity were calculated by DSA. They serve as intermediate values, which are used to characterise behaviour of data in the network and estimate its power consumption by the physical model of the links LM.

For each of the six examined configurations with low-power and error protection techniques, the full set of experiments was performed. It includes two data types, two mesh sizes for each data type, two values of packetisation for each mesh and five to seven values of emission probability for each packetisation value (Table 1). In total were 40–60 experiments for each configuration. This is apart from the simulations, which were performed to choose the right parameters. Thus, we pretend for generality of the received results.

5 Results and analysis

One of the main results of the performed simulation experiments is a distribution of switching activity through the data packet. To implement an efficient low-power technique, we should know the packet fields that have the primary impact on the total switching activity. Table 2 presents the part of total switching activity accounted to each packet field. The majority of the packet switching activity (88.1%) is due to the payload, while it occupies only 80.7% of the bus width. Thus, implementation of low-power techniques for the payload field should be the most efficient.

Table 3 presents the reduction of the switching activity in the network interconnections. The highest decrease that can be obtained after implementation of the bus-invert coding for the NoC interconnections is about 8.7%.

Table 4 shows the time delays in the switch, encoder, decoder and the link for the examined configurations as well as the common delay of the design. We assume that there are buffers on the switch and encoder inputs. Hence, the common delay of Configuration A equals to a switch delay. In Configurations B and C, a sum of delays in encoder, link and decoder dominates over time delay in the switch, and hence, the total delay equals to the sum of delays in encoder, link and decoder. The delay penalty due to encoding is very significant when compared with the delays on the switches and the links. Although the switching activity on the links can be reduced by 8–10%, the delay penalty is very significant. The values for the switch and network interconnection operating clock periods were presented in [10].

If we compare the three configurations with the normalised frequencies and voltages, the power consumption of Configuration A is the lowest. As can be seen in Table 5, the power consumption of Configuration A is in fact the lowest. The switching activity on the links can be reduced but only by a too small amount. As noted above, the traffic pattern has a significant influence on this result and consecutive packets with a strong correlation may result in much higher switching activity reduction.

Using encodings with more and smaller sub-buses (P3BI, P4Bi etc.) gives diminishing returns, as our experiment results support. We suspect that in general the optimal is around P2BI and P3BI.

Tables 2–5 show the result for a 5 × 5 network with 40% traffic load. We made experiments with up to 16 × 16 networks and load values between 0 and 100%, which all show the same trend. Hence, we conclude that our results are generally valid for moderately sized networks with the main limitation being the traffic pattern as discussed above.

Fig. 6 illustrates the error count per packet against the power per useful bit. For a low error rate, Scenario 2 is more power efficient than Scenario 1. However, as the error rate increases (as the voltage drops), the protecting

Table 2 Switching activity (SA) distribution

	Address	Empty bit	Hop counter	Payload	Whole packet
number of bits (fraction)	16 (13.5%)	1 (0.8%)	6 (5.0%)	96 (80.7%)	119 (100%)
average SA, %	11.0	39.5	10.1	21.4	18.2
normalised SA, %	7.6	1.7	2.6	88.1	100

Table 3 Reduction of the switching activity in 5 × 5 network

Configuration	Average switching activity, %	Average reduction, %
A	20.8	—
B	19.6	5.8
C	19.0	8.7

Table 4 Delay and frequency for three configurations

Conf.	Block	Delay, ns	Total delay/frequency
A	switch	0.42	0.42 ns/2.38 GHz
	link	0.08	
B	switch	0.42	2.52 ns/0.39 GHz
	enc	2.43	
	dec	0.01	
	link	0.08	
C	switch	0.42	1.37 ns/0.73 GHz
	enc	1.28	
	dec	0.01	
	link	0.08	

code of Scenario 2 is not strong enough to compensate and both scenarios approach each other. Scenario 3 is consistently more power efficient than the other scenarios over the entire reasonable voltage range.

Other experiments with various network sizes, traffic load and other parameter variations support the overall conclusion that Scenarios 1 and 2 are rather close to each other, whereas Scenario 3 is superior, even though the benefit can become rather small when the line error rate rises too sharply with the decrease of the voltage.

6 Related work

Various methodologies for reduction of bus power consumption has been presented in the literature [11, 14–16] and successfully applied to on- and off-chip communication. However, all these methods are well investigated and widely implemented; none of them has performed their feasibility study for NoCs. A need of this examination follows from the architectural and purpose differences between SoC buses and NoC communication backbone.

In [16, 17], several issues are described concerning applicability and inapplicability of low-power bus encodings for deep sub-micron buses. Hence, it is the focus of the research to show the adaptability of these methods for a new architecture for future integrated computer and telecommunication systems.

There are several works, describing methods to estimate power consumption both on combinational logics and interconnect wires. Benini *et al.* suggested the bit energy approach [18] to calculate power dissipation on switch fabrics in network routers. They used look-up table, pre-calculated from Synopsys Power Compiler simulation.

The error-control techniques for NoC architectures were investigated in [5, 6]. In these papers, several quality-of-service

Table 5 Power consumption normalised to the same performance

Configuration	Normalised voltage, V	Block	Switching activity, %	Power consumption, mW
A	0.51	*S*	20.8	2.26
		link (×4)	20.8	281.23
		total		283.49
B	0.90	*S*	20.8	7.03
		enc (×4)	20.8	7.98
		dec (×4)	19.6	0.86
		link (×4)	19.6	826.99
		total		842.86
C	0.70	*S*	20.8	4.26
		enc (×4)	20.8	3.81
		dec (×4)	19.0	0.49
		link (×4)	19.0	484.28
		total		492.84

doi: 10.1049/iet-cdt:20050060

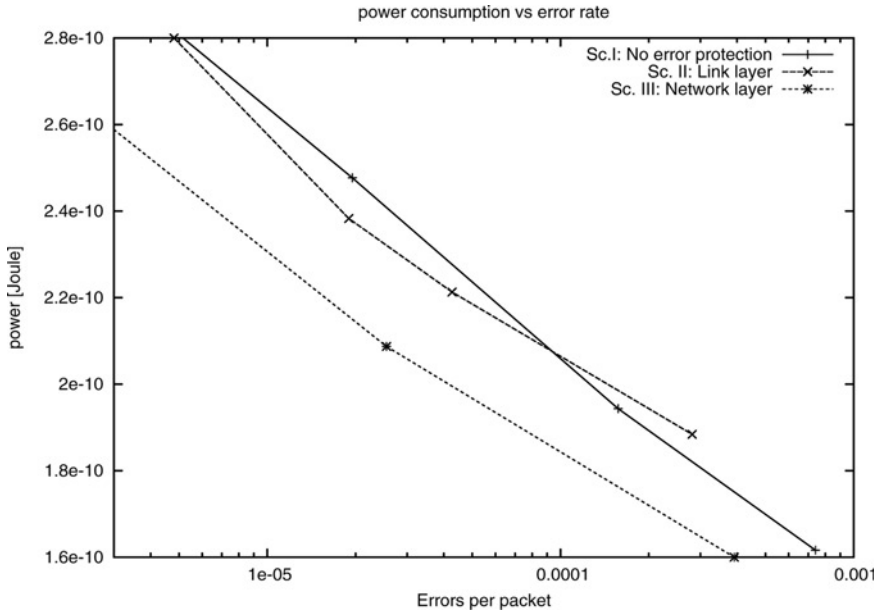

Figure 6 *Power consumption per communicated useful bit*

schemes with different levels of data protection were presented. Four of them were implemented in System C in [6]. In the contents of current research, several of these schemes were implemented in VHDL. Its power consumption, time and area overheads were estimated with Synopsys Power Compiler.

7 Conclusion

In the context of the presented research, the most significant low-power techniques, which are presented in literature, were analysed and compared. From this analysis, we conclude that a bus-invert encoding technique is the most suitable mechanism for the Nostrum NoC architecture. To implement it in the most efficient way, the part of the data packet with the largest value of switching activity became subject to the bus-invert mechanism, which was implemented on a data-link layer. In addition, the error protection technique was implemented both on data-link and network layers. We performed a series of simulation experiments, taking into account various parameters and conditions. Several models were specially developed to support the experiments. The received results were then analysed in detail and a number of conclusions were presented.

The simulation results show that the payload switching activity has the main part in total packet switching activity and approximately equals to 88%. Therefore the implemented low-power techniques should target the payload. The switching activity is linearly dependent on network load. Implemented low-power techniques on the link level give ~8–10% reduction of the switching activity in the network interconnection. Since in case of random data that have uniformly distributed switching activity, bus-invert coding is optimal [19]; other encoding techniques will perform worse. Thus, low-power bus encoding on the switch-to-switch links has limited value and most likely increases overall power consumption.

Error protection codes have a potential to decrease power consumption because they would allow operating the network with a lower voltage. End-to-end encoding schemes are superior to link-level encoding schemes. All experiments have compared the different cases with respect to the same performance. The additional delay of encoders and decoders has been accounted for by adjusting the frequency and the voltage.

In summary, we believe our conclusions are valid for all regular, mesh-based multi-hop communication networks, and for a wide range of routing, switching and buffering schemes. The major difference in other networks will come from the relative weight of switches and network interfaces compared with the links. In Nostrum, the switches and interfaces are lightweight without buffers and consume very little power compared with the links. In other networks, this ratio is different and hence the overall trade-off will change. Critical assumptions that may change the overall picture are the traffic patterns in the network and the relation of voltage and wire line error probability. We have argued for the sensibility of our choice but these have to be revisited when more data about applications and technology are available.

8 References

[1] JANTSCH A., TENHUNEN H.: 'Networks on chip' (Kluwer Academic Publishers, 2003)

[2] BENINI L., DEMICHELI G.: 'Networks on chip: a new SoC paradigm', *IEEE Comput.*, 2002, **35**, (1), pp. 70–78

[3] NILSSON E.: 'Design and implementation of a hot-potato switch in a network on chip'. Master Thesis, IMIT/LECS 2002-11

[4] LIU D.L., SVENSSON C.: 'Power consumption estimation in CMOS VLSI chips', *IEEE J. Solid State Circuits*, 1994, **29**, (6), pp. 663–670

[5] VELLANKI P., BANERJEE N., CHATHA K.S.: 'Quality-of-service and error control techniques for network-on-chip architectures'. GLSVLSI'04, 26–28 April 2004, Boston, MA, USA

[6] ZIMMER H., JANTSCH A.: 'A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip'. CODES + ISSS'03, 1–3 October 2003, Newport Beach, CA, USA

[7] JANTSCH A., LAUTER R., VITKOVSKI A.: 'Power analysis of link level and end-to-end data protection in networks on chip'. Proc. IEEE Int. Symp. Circuits and Systems, May 2005

[8] KUMAR S., JANTSCH A., SOININEN J.-P., *ET AL.*: 'A network on chip architecture and design methodology'. ISVLSI'02, 2002

[9] PAMUNUWA D.: 'Modelling and analysis of interconnects for deep submicron systems-on-chip'. Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden, 2003

[10] NILSSON E., ÖBERG J.: 'Reducing power and latency in 2-D mesh NoCs globally pseudochronous locally synchronous clocking'. Hardware/Software Codesign and System Synthesis, CODES + ISSS, 2004

[11] RAGHUNATHAN V., SRIVASTAVA M.B., GUPTA R.K.: 'A survey of techniques for energy efficient on-chip communication'. DAC 2003, 2–6 June 2003, Anaheim, CA, USA

[12] International Technology Roadmap for Semiconductors, 2003 Edn., http://public.itrs.net/Files/2003ITRS/Home2003.htm

[13] JEJURIKAR R., PEREIRA C., GUPTA R.K.: 'Leakage aware dynamic voltage scaling for real time embedded systems', CECS Technical Report #03–35 Report, November 2003

[14] MAMIDIPAKA M., HIRSCHBERG D., DUTT N.: 'Low power address encoding using self-organizing lists'. ISLPED'01, 6–7 August 2001, Huntington Beach, CA, USA

[15] BENINI L., DE MICHELLI G., MACII E., PONCINO M.: 'Power optimization of core-based systems by address bus encoding', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 1998

[16] WONG S.-K., TSUI C.-Y.: 'Re-configurable bus encoding scheme for reducing power consumption of the cross coupling capacitance for deep sub-micron instruction bus' (IEEE, 2004)

[17] KRETZSCHMAR C., NIEUWLAND A.K., MULLER D.: 'Why transition coding for power minimization of on-chip buses does not work' (IEEE, 2004)

[18] TAO YE T., BENINI L., DE MICHELI G.: 'Analysis of power consumption on switch fabrics in network routers'. DAC 2002, 10–14 June 2002, New Orleans, LA, USA

[19] SIEGMUND R., KRETZSCHMAR C., MULLER D.: 'Adaptive partial businvert encoding for power efficient data transfer over wide system buses'. SBCCI 2000, Manaus, Brasil

doi: 10.1049/iet-cdt:20050060