Power analysis of link level and end-to-end data protection in networks on chip

Axel Jantsch, Robert Lauter, Arseni Vitkowski
Royal Institute of Technology, Stockholm, Sweden

***Abstract*— We provide a power analysis for the communication in the Nostrum NoC concluding that power consumption is dominated by the links between switches while the switches and network interfaces contribute with a mere few percent to the power consumption. Further we analyze link level low power encoding techniques with the conclusion, that they spend several times more power than no encoding at all, if normalized for the same performance, which is done by adjusting supply voltage and frequency. Data protection schemes also have the potential to reduce power if voltage levels can be reduced and certain faults can be tolerated. We experiment with link level and end-to-end data protection schemes. They only moderately increase power consumption and have indeed the potential to save power. However, this potential strongly depends on the application traffic patterns and fault models of future technology generations.**

I. Introduction

Network-on-Chip is an architecture template that targets very complex ICs with 10s or even a few 100s of resources. Predictably, power consumption will be one of the main bottlenecks, if not the single most important challenge. Thus, it must be tackled at all layers from the physical and technology level to the application and algorithmic level and every route for power reduction and power management must be pursued.

We study how link and network level bus encoding and error protection codes influence power consumption in the Nostrum NoC architecture. Nostrum is a regular 2-D mesh network with a deflective routing scheme, minimal switch internal buffering and very high link level bandwidth with 128 bit buses [1], [2], [3]. It aims at network sizes with roughly 100 resources and therefore offers the bandwidth necessary to realize on-chip networks of that size. We study two specific questions, first low power encoding of the switch to switch connections and then how error protection codes influence power consumption when considering that stronger codes have higher fault tolerance and thus allow for lower operating voltage.

A wide variety of low power bus encoding schemes have been developed; see [4], [5] for surveys. A recent analysis by Kretschmar et al. [6] concludes that bus lines must be 20mm or longer for encoding techniques to be energy efficient. Our results support this claim since we also conclude that for 2mm long bus lines low power encoding techniques will in effect increase power consumption for a next generation 65nm CMOS technology. In addition we report that the power consumed in the wires dominates the overall communication power consumption and can be as much as two orders of magnitude higher than the power consumed in the switches.

The question if forward error correction (FEC) with linear block codes is more or less power efficient than packet retransmission schemes for communication over a bus has been investigated by Bertozzi et a. [7]. For a fault model with statistically independent faults they conclude that retransmission schemes are more energy efficient in particular for long wires and high reliability constraints. In contrast, our study concerns multi-hop networks and investigates if error protection coding should be done at the link layer (from switch to switch) or at the network layer (end-to-end from resource to resource). We conclude that network layer error protection is more power efficient than both no protection and link layer protection.

II. Assumptions and Experimental Setup

We make the analysis for the Nostrum NoC which targets 65 nm technology and beyond. It is a regular two dimensional network, where each resource connects to exactly one switch. The switches perform hot-potato deflection routing with minimal switch internal buffering. The network implements a best effort communication service. Nostrum also offers guaranteed latency services based on virtual switches[3], but in our analysis here we only focus on the best effort service. The switches are connected over 128 bit buses which are $2mm$ long assuming each resource is a $2mm \times 2mm$ block. Nostrum is not limited to this rigid geometry but we use it here in our analysis.

The power consumption of all design blocks (switch, encoder, decoder, etc.) are analyzed with Synopsys Power compiler with the UMC18 (180nm) library. We scaled the result for an expected 65 nm technology using [8], [9]

$$P_{65} = P_{180} \times \frac{C_{id65} V_{dd65}^2}{t_{clk65}} \times \frac{t_{clk180}}{C_{id180} V_{dd180}^2}. \qquad (1)$$

Voltage scaling due to frequency changes was performed using [10]

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

where t_{inv} is the delay of one inverter, L_d, K_6 and α are technology constants and V_{th} is the threshold voltage.

The power consumption for the switch is taken from [11].

We calculate the power consumption for the links between the switches using

$$P_{Link} = \frac{1}{2}\alpha 128 C_W V_{dd}^2 f$$
$$C_w = \epsilon \frac{w_{Net} L_R}{t_{ox}}$$

where C_W is the switch capacitance for one wire, α is the switching probability of one wire, w_{Net} and L_R are the width and length of a wire, respectively, t_{ox} is the distance between wire and ground, V_{dd} is the supply voltage, and f is the frequency. The constants for 65nm technology are taken from the SIA technology roadmap, while L_R is assumed to be $2mm$.

We use random traffic and assume there is no correlation between the data in two consecutive packets over the same link. This is a critical assumption that depends heavily on the traffic scenario in the network. It is violated if a few connections dominate the traffic and monopolize several links over a long period and if only weak data compression is used for these connections. We make this assumption because larger network with more diverse applications and traffic will exceedingly approach this scenario. Also, we assume high bandwidth data streams will be aggressively compressed in the future in order to minimize the traffic and to save power. Thus, in a sense we decouple end-to-end more application specific data compression efforts from techniques that can be employed in the network by network services. This avoids the danger that we report seemingly efficient data encoding on the link level, which are in effect rendered valueless and even counter productive when end-to-end data encoding is employed or other, more random traffic patterns emerge.

We only consider faults on the 2mm long wires between switches, because these long wires will be a subject to various potentially harmful deep submicron effects. To obtain a complete understanding of the occurrence of faults and the power consumption of the communication network, the switches, encoders, decoders and network interfaces need to by analyzed as well in a complementary study.

III. Link Encoding

The requirements for link encoding are very tight. Since an encoder and decoder has to be implemented on each of the four links of every switch, the area overhead has to be tiny and, obviously, these blocks should be power efficient and must not harm performance.

After a comprehensive review of low power bus encoding techniques [5] we selected two for our experiments that have the highest potential for power saving while still represent a feasible implementation on every link. Thus we distinguish three configurations:

- Configuration A is the reference scenario with no encoders and decoders on the links.
- Configuration B uses partial bus invert (P1BI) coding technique [12], [5].
- Configuration C uses the same bus inverting technique but splits the bus in two equal parts and encodes them separately (P2BI). The advantage over P1BI is that each part can be encoded more efficiently and faster but we pay in terms of an additional line that signals if a bus part is inverted or not.

TABLE I

Delay and clock period for the three scenarios.

Conf.	Block	delay [ns]	total delay	frequency
A	Switch	$0.42ns$	0.42 ns	2.38 GHz
	Link	$0.08ns$		
B	Switch	$0.42ns$	2.52 ns	0.39 GHz
	Enc.	$2.43ns$		
	Dec.	$0.01ns$		
	Link	$0.08ns$		
C	Switch	$0.42ns$	1.37 ns	0.73 GHz
	Enc.	$1.28ns$		
	Dec.	$0.01ns$		
	Link	$0.08ns$		

TABLE II

Power consumption normalized to the same performance.

Conf.	V_{dd}	Block	Switching activity %	Power consumption
A	0.51	Switch	20.8	2.26
		4 links	20.8	281.23
		Total		283.49
B	0.90	Switch	20.8	7.03
		4 Enc.	20.8	7.98
		4 Dec	19.6	0.86
		4 links	19.6	826.99
		Total		842.86
C	0.70	Switch	20.8	4.26
		4 Enc.	20.8	3.81
		4 Dec	19.0	0.49
		4 links	19.0	484.28
		Total		492.84

Table I shows the effect of the encoding on the delay. The delay penalty due to encoding is very significant when compared to 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. If we compare the three configurations with the same frequency, we can operate configuration A at a lower voltage, which results in a much lower power consumption. As can be seen in table II, 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 I and II 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.

IV. End-to-end data protection

Next we ask the question if power can be saved by employing data protection mechanisms, because then the network can operate with a lower voltage. A lower voltage will result in some signaling errors, which can be tolerated due to the error protection. Again we compare three scenarios:

- Scenario I is the reference with no error protection in place. For an 8×8 network we have 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.
- Scenario II uses error protection at the link layer. It 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. In this scenario also the payload (address and hop counter) is 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 the empty bit is protected by other meas, 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 of scenario I; 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 less than 10 gates on the critical path[13]. The price that we pay is a higher number of inter switch wires: 125 wires as compared to 95 in scenario I and 119 in scenario III.
- Scenario III deploys end-to-end error protection for the payload. However, the header is still protected on the link layer and encoded/decoded between two switches just as in scenario II. For the payload we can use stronger codes to compensate for the accumulation of errors over multiple hops. For the header this is not possible, because if the address gets corrupted, the packet will not be delivered correctly. Hence, we have to make sure that every packet gets 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 a SEC/DED code, but it is stronger than the code of scenario II 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 roughly 10000 gates and incurs a delay several times higher than the code in scenario II. This is justified since the encoder/decoder exists only once per resource while scenario II needs 4 encoder/decoder per switch. Also, 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 section.

We assume an exponential dependence of the wire fault probability on the voltage. The wire fault probability is the probability that one bit gets flipped when transmitted from one switch to the next. Unfortunately we do not have reliable data for future technology generations on this relation. We use a similar model as Bertozzi et al. [7] but scale it to lower voltages and checked the result for consistency with recently published data, e.g. from [14], [15]. We consider the range between 0.6V to 1.5V and assume wire fault probabilities of 10^{-7} and 10^{-15}, respectively. We acknowledge that this is a weak link in our argumentation and not well supported by a convincing quantitative fault model. However, we experimented with variations of this model with the conclusion that our overall results are not very sinsitive with respect to the details of this model. In next generation CMOS technologies of 65nm and below there will be a voltage range in which the fault probability will sharply rise with decreasing voltage. Our model assumes a fairly sharp rise and is thus conservative. If the rise is even steeper and the corresponding voltage range is narrower, there may be practical problems in realizing the potential power saving, e.g. the limitation on voltage tuning.

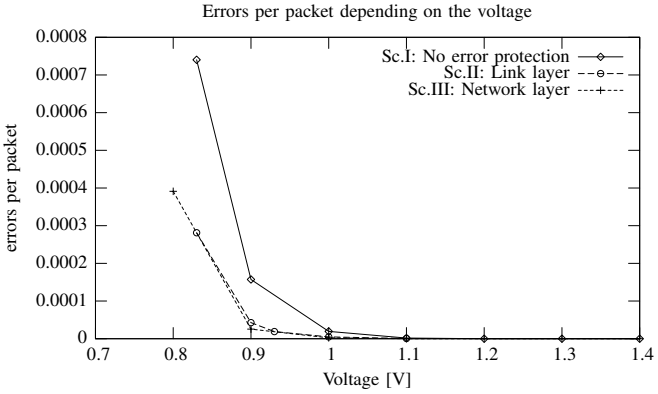

Fig. 1. The fault probability grows exponentially with decreasing voltage.

Figure 1 shows how the average number of error increases sharply as the voltage decreases. It increases less sharply for the scenarios II and III due to error correction.

Figure 2 shows how the power consumption per useful bit (excluding header and redundancy bits) decreases with falling voltage for all three scenarios. It also shows the overhead due to the data protection mechanism which is greatest for scenario II due to the larger number of lines between switches.

While figures 1 and 2 show the negative and the positive effects, respectively, of voltage decrease, figure 3 illustrates their combination by plotting the error count per packet versus the power per useful bit. For a low error rate, scenario II is more power efficient than scenario I. However, as the error rate increases (as the voltage drops) the protecting code of scenario II is not strong enough to compensate and both scenarios approach each other. Scenario III is consistently more power efficient than the other scenarios over the entire reasonable

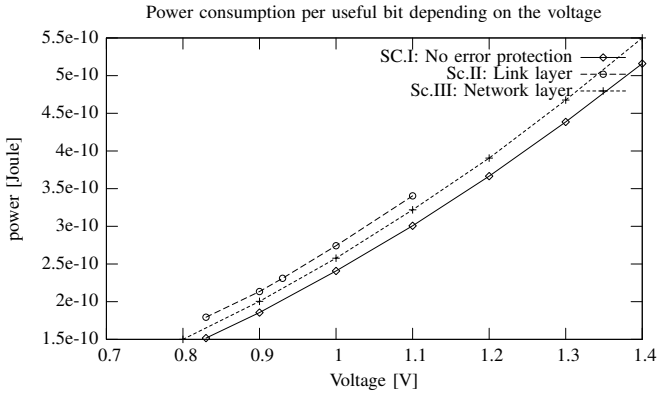

Fig. 2. Power consumption per communicated useful bit.

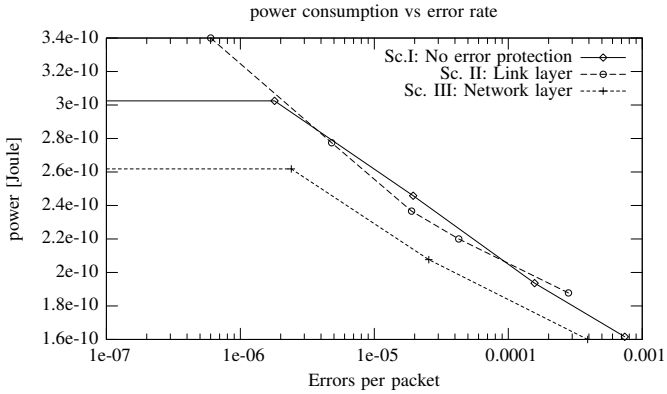

Fig. 3. A low error rate requires more power.

voltage range.

Other experiments with various network sizes, traffic load and other parameter variations support the overall conclusion, that scenarios I and II are rather close to each other while scenario III is superior, even though the benefit can become rather small when the line error rate rises too sharply with the decrease of the voltage.

V. Conclusion

We have investigated techniques for lowering power consumption of communication in Networks-on-Chip. We have specifically done a study on the Nostrum NoC. 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 to operate the network with a lower voltage. End-to-end encoding schemes are superior over 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.

It remains to be seen if our findings are valid for other networks with different topologies and routing policies. We expect major differences in the trade-off between power and data protection in other networks will come from the relative weight of switches and network interfaces compared to the links. In Nostrum, the switches and interfaces are lightweight without buffers and consume very little power compared to 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 date about applications and technology are available.

References

[1] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrjä, and A. Hemani, "A network on chip architecture and design methodology," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, April 2002. [Online]. Available: http://www.imit.kth.se/ axel/papers/2002/ISVLSI.pdf

[2] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, "The Nostrum backbone - a communication protocol stack for networks on chip," in *Proceedings of the VLSI Design Conference*, Mumbai, India, January 2004. [Online]. Available: http://www.imit.kth.se/ axel/papers/2004/VLSI-Millberg.pdf

[3] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," in *Proceedings of the Design Automation and Test Europe Conference (DATE)*, February 2004. [Online]. Available: http://www.imit.kth.se/ axel/papers/2004/DATE-Millberg.pdf

[4] V. Raghunathan, M. B. Srivastava, and R. K. Gupta, "A survey of techniques for energy efficient on-chip communication," in *Proceedings of the Design Automation Conference*, June 2003, pp. 900–905.

[5] A. Vitkowski, "A study on power consumption in the nostrum communication network," Master's thesis, Institute of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, April 2004. [Online]. Available: http://www.imit.kth.se/ axel/papers/2004/MSc-arseni-vitkowski.pdf

[6] C. Kretzschmar, A. K. Nieuwland, and D. Muller, "Why transition coding for power minimization of on-chip buses does not work," in *Proceedings of the Design Automation and Test Europe Conference (DATE)*, February 2004.

[7] D. Bertozzi, L. benini, and G. D. Micheli, "Low power error resilient encoding for on-chip data buses," in *Proceedings of the Design Automation and Test Europe Conference (DATE)*, March 2002.

[8] W. J. Dally and J. W. Poulton, *Digitial Systems Engineering*. New York: CUP, 1998.

[9] D. Pamunuwa, J. Öberg, L.-R. Zheng, M. Millberg, A. Jantsch, and H. Tenhunen, "A study on the implementation of 2-D mesh based networks on chip in the nanoregime," *Integration - The VLSI Journal*, vol. 38, no. 1, pp. 3–17, October 2004.

[10] R. Jejurikar, C. Pereira, and R. K. Gupta, "Leakage aware dynamic voltage scaling for real time embedded systems," CECS, Tech. Rep. 03-35, 2003.

[11] E. Nilsson and J. Öberg, "Reducing peak power and latency in 2-d mesh nocs using globally pseudochronous locally synchronous clocking," in *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis*, September 2004. [Online]. Available: http://www.imit.kth.se/ axel/papers/2004/CODES+ISSS-Erland.pdf

[12] N. Chang, K. Kim, and J. Cho, "Bus encoding for low-power high performance memory systems," in *Design Automation Conference*, Los Angelos, June 2000.

[13] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip," in *Proceedings of the CODES-ISSS Conference*, October 2003. [Online]. Available: http://www.imit.kth.se/ axel/papers/2003/Codes+ISSS-Zimmer.pdf

[14] M. Nakai, S. Akui, K. Seno, T. Meguro, T. Seki, T. K. abd A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura, "Dynamic voltage and frequency management for a low power embedded microprocessor," in *ISSCC*, 2004.

[15] K. Nowka, G. Carpenter, E. M. Donald, H. Ngo, B. Brock, K. Ishii, T. Nguyen, and J. Burns, "A 0.9V to 1.95V dynamic voltage scalable and frequency scalable 32b power PC processor," in *ISSCC*, 2002.