

A High Level Power Model for the *Nostrum* NoC

Sandro Penolazzi and Axel Jantsch
Royal Institute of Technology (KTH)
164 40 Kista, Sweden
{sandrop, axel}@kth.se

Abstract

We propose a power model for the *Nostrum* NoC. For this purpose an empirical power model of links and switches has been formulated and validated with the Synopsys Power Compiler. The model, which from now on will be called *Nos-HPM* (*Nostrum* High-Level Power Model) allows a fast power analysis and is accurate within 5%. System simulations with *Nos-HPM* run up to 500 times faster than with Power Compiler for a 4 x 4 network. We find a maximum power consumption of 0.7 W for a 4 x 4 mesh and 3.5 W for an 8 x 8 mesh, both implemented in 0.18 μ m UPC CMOS technology. In the worst case the average energy per cycle for a 128-bit packet is 508 pJ, while it is 20 pJ for a payload byte. The power consumption of all the links is equivalent or slightly higher than the power consumption of all the switches. A comparison between our results and some related work is also presented.

1. Introduction

As speed and complexity of integrated circuits keep growing, new solutions need to be found to satisfy time-to-market constraints and get stable improvements in system performance. Next generation systems based on deep submicron technology must be based on platform reusability and high performance communication backbones.

In the latest years many researchers and developers have been trying to address such issues proposing the Network-on-Chip (NoC) concept as an evolution of the more conventional System-on-Chip (SoC) architecture [1], [2].

The present article aims at presenting a concrete analysis on power and energy consumption in *Nostrum*, which implements the NoC concept proposed at KTH [3]. A power model is therefore implemented, combining reasonable accuracy and computation

speed. This allows a high level analysis of the system at an acceptable simulation time.

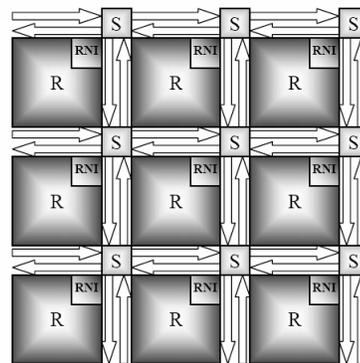


Figure 1. *Nostrum* NoC backbone

2. The *Nostrum* backbone

Nostrum has been conceived as a regular packet-switched network, implementing different conceptual layers according to the OSI model [4]. The topology chosen is an M x N-mesh, as shown in Figure 1, where each hot-potato switch is connected to one resource, through a Resource Network Interface (RNI), and to other four switches, through 128-bit-wide links. The choice of a deflective routing algorithm aims at keeping its area small and its power consumption low due to the absence of internal buffer queues. Together with the control unit, five multiplexers represent the core of a switch architecture in *Nostrum*. Being *Nostrum* a packet-switched NoC, Packet Data Units (PDUs) represent the means through which information is exchanged among the IP blocks in the network. A PDU is a 128-bit vector composed of several fields, according to the pattern shown in Figure 2. More detailed information on the VHDL model of *Nostrum* can be found in [3]. The distinction between the fields belonging to the header (VC, DA, SA, E, HC) and the field of the payload (PL) has shown to be relevant when dealing with power analyses.

1	12	12	1	6	96		
VC	DA		SA		E	HC	PL
127	126	115	114	103	102	101 96	95 0

Figure 2. PDU bit vector in *Nostrum* (VC: Virtual Circuit, DA: Destination Address, SA: Source Address, E: Empty bit, HC: Hop Counter, PL: PayLoad)

3. Link power model

Switches, Resource Network Interfaces (RNI) and Interconnection Links represent the elements mainly responsible for power consumption in a NoC [5]. In the present article though, the contribution given by the RNI will be neglected. The mathematical model used to study the power consumption behavior is reported in literature as the so called *Bit Energy* approach [5]. According to it, power is proportional to the switching activity (or switching probability) of the bits delivered into the network.

Our power model for the links is implemented by following the considerations in [6] and [7]. Each link in *Nostrum* is composed of 128 wires, connecting two neighboring switches (one link for each direction). We assume that no coupling capacitance exists between two adjacent wires, due to shielding. As a consequence, only the capacitance between a wire and ground is considered. The general relation used to model the power for a single wire is the following:

$$P_{wire} = \frac{1}{2} \cdot \alpha \cdot C_{wire} \cdot V_{DD}^2 \cdot f$$

where α is the switching activity of the wire ($0 \leq \alpha \leq 1$), C_{wire} is its capacitance, V_{DD} is the supply voltage, f is the network frequency. The numerical value for C_{wire} has been calculated from [6] and considered reliable for our NoC features and technology. The resulting capacitance for a wire is $C_{wire} = 1.4$ pF. The total link power for the entire NoC becomes therefore:

$$P_{NoC} = \sum_{i=1}^M \sum_{j=1}^N P_{wire}(i, j)$$

with M and N the number of links and of wires per link respectively.

4. Switch power analysis

Concerning power consumption inside a switch, three main elements can be distinguished which give a significant contribution: internal modules, such as

control units and multiplexers, internal interconnection wires, internal buffer queues. In the specific case no buffer queues are present. Since we aim at finding an empirical function that reports reliable power information for any possible combination of inputs, a deep analysis of the switch behavior is required.

The basic idea is to measure power consumption under a series of different input data patterns and, from the obtained results, elaborate a generally valid power model. We have used Synopsys Power Compiler [8] to measure the power consumption. The chosen technology library is 0.18 μm and the supply voltage is 1.8 V. In the analysis a few simplifications have been made: power has always been studied as depending on the contribution given by the clock signal and the five 128-bit input data vectors. Other input ports are present in the VHDL model of the switch, yet their value has permanently been set to 0, since their contribution to the power is considered negligible. Furthermore, even if we are dealing with 128-bit vectors, the bit corresponding to the VC is permanently set to 0, since the virtual circuit option has not been implemented in the switch under study. Therefore only the remaining 127 bits are being part of the analysis. The leakage power estimated for a switch unit is around 726 nW, value that is several orders of magnitude lower than the dynamic power. For this reason, leakage power has not been considered in the rest of the analysis.

4.1 variable inputs

Dynamic inputs represent the most interesting condition of analysis. To test this condition, provided a clock signal @ 100 MHz, all the input data bits have initially been set to a static condition. A series of simulations have been conducted by adding progressively one switching bit, until all the bits were switching. The results obtained with 0 and all 635 bits switching are 7.55 mW and 48.30 mW, respectively. Several different sequences of inputs have been tested, in order to get in the end a more reliable model. From the results, two important conclusions can be drawn: (a) Different static initial conditions affect differently the dynamic power; this behavior can be explained by the presence of a small amount of sequential logic in the switch architecture, which triggers different internal transitions depending on the static values 0 and 1 present on the data inputs. (b) The position of each bit inside the data vector has a tight relation with its contribution to the overall power. In particular, bits belonging to the *header* affect the power much more than those belonging to the *payload* because of the different amount of logic and circuitry associated to each single input bit.

5. Power model deduction and validation

Starting from the analysis performed with Power Compiler, an empirical function has been implemented with the purpose of modeling the switch behavior for any kind of input pattern. The relation found takes into account the considerations presented above and looks as follows:

$$\begin{aligned} \text{switch_pow} = & \text{min_pow} + \text{tr_pl} \cdot \left(p10 + p110 \cdot \frac{s1}{s_tot} \right) + \\ & + \text{tr_hc} \cdot \left(hc0 + hc10 \cdot \frac{s1}{s_tot} \right) + \text{tr_e} \cdot \left(e0 + e10 \cdot \frac{s1}{s_tot} \right) + \\ & + \text{tr_sa} \cdot \left(sa0 + sa10 \cdot \frac{s1}{s_tot} \right) + \text{tr_da} \cdot \left(da0 + da10 \cdot \frac{s1}{s_tot} \right) \end{aligned}$$

where the different elements have the following meaning:

- *min_pow*: reference power, calculated for static input data vectors and depending only on the clock signal. Its value is 7.55 mW
- *tr_pl, tr_hc, tr_e, tr_sa, tr_da*: number of total switching bits in the respective fields
- *p10, hc0, e0, sa0, da0*: contribution per bit according to the field it belongs to, supposing all the static bits set to 0
- *p110, hc10, e10, sa10, da10*: difference between the bit contribution calculated supposing all the static bits set to 1 and the bit contribution calculated supposing all the static bits set to 0
- *s1*: number of static 1
- *s_tot*: total number of static bits. When *s_tot* = 0 the expression does not apply and the power of 48.3 mW is assigned to the switch automatically

To test the effectiveness of our formula, a set of simulations has been performed and power has been estimated both by using the formula and Power Compiler. The results show that *Nos-HPM* tends to underestimate the actual power found by Synopsys. The explanation is that power depends on the distribution of dynamic bits over the 5 inputs. For our switch power analysis, the bit vectors have always been filled with dynamic bits in an ordered sequence: bit after bit, and input after input. In a real simulation environment instead, the distribution of dynamic bits is much more random. It seems therefore that, given the same amount of bits switching, power consumption is higher if these bits are randomly spread over all the inputs than if they are stored sequentially in the same input. To solve this problem, a corrective function has been introduced,

which is simply added to the original one. It has been derived empirically:

$$\text{correction} = \text{rand} \cdot \left(4.764 \cdot \sqrt[3]{\text{tr_tot}} - 0.064 \cdot \text{tr_tot} \right)$$

The factor *rand* has the purpose to vary the effect of the corrective function according to how much the switching bits are spread over the 5 inputs: the more they are spread, the higher *rand* is. Its value is limited between 0 and 1.

Figure 3 shows a comparison between calculations made with Power Compiler and with our new formula. The two functions plotted look quite similar to each other. In fact there is a slight difference in the initial phase of the simulation, where our function seems to overestimate the actual power value. However, when the network has reached a steady state, this difference almost disappears. In the whole simulation an average difference of +3.6% for our function has been calculated, mainly due to the initial part. Sporadic peaks of +40% and -35% are present, but they do not affect the global behavior of *Nos-HPM*.

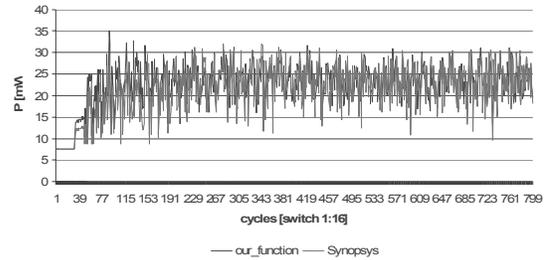


Figure 3. Synopsys vs. *Nos-HPM*

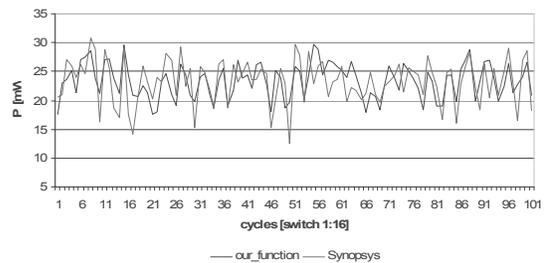


Figure 4. Synopsys vs. *Nos-HPM* (zoom)

Figure 4 reports a closer view of the two functions. From that, it is also possible to notice that their trend is very similar. Further simulations, with different environmental parameters, have been made, to validate the switch power model. The results have confirmed its reliability, being the average difference with respect to Power Compiler within 5%.

6. Power estimations in *Nostrum* and related work

Using *Nos-HPM*, a high number of simulations has been run, and the most significant results are here reported. Different network sizes have been tested, from 4 x 4 up to 8 x 8. Uniform random traffic has been assumed, with all the resources emitting one packet each simulation cycle. 100-cycle simulations have been run. Table 1 and 2 show a list of results, obtained assuming a payload (PL) that contains randomly generated values.

Table 1. NoC power ($f = 100$ MHz)

Size	Tot. NoC Power [W]	Tot. Switch Power [W]	Tot. Link Power [W]	Av. Switch Power [mW]	Av. Link Power [mW]	Link Power / Switch Power
4x4	0.733	0.356	0.377	22.25	5.89	1.06
5x5	1.230	0.583	0.647	23.32	6.47	1.11
6x6	1.860	0.870	0.990	24.17	6.88	1.14
7x7	2.600	1.200	1.400	24.49	7.14	1.17
8x8	3.470	1.600	1.870	25.00	7.30	1.17

Table 2. Packet power ($f = 100$ MHz)

Size	Power per packet [mW]	Power per packet per hop [mW]	Power per payload byte [mW]	Power per payload byte per hop [mW]
4x4	45.8	15.6	1.85	0.63
5x5	49.1	15.7	1.92	0.61
6x6	51.7	15.9	1.98	0.61
7x7	53.0	15.9	2.03	0.61
8x8	54.3	15.9	2.09	0.61

The results show an average switch power around 24 mW and an average link power around 6.7 mW. Note the strong impact that links have in affecting the overall power. The ratio Link Power / Switch Power is even higher than 1. Simulations have been run also with constant payload, giving power savings up to 17% for the switches and 73% for the links. Such percentages confirm the low impact of payload on the total switch power consumption.

Se-Joong Lee et al. in [9] analyzed area and energy performance for three different NoCs, discussing architectural decisions, approaches and solutions. Concrete area and energy figures are also reported for the building blocks used, implemented in 0.18 μm technology. A uniformly distributed network traffic is assumed, with all resources transferring 80-bit packets at the same rate. Typical energy values for an 80-bit 1-mm metal link suggest 47.8 pJ, against an average of 67.4 pJ for our 128-bit 4-mm link in case of random payload. The maximum theoretical value for the link energy in *Nostrum* is 290.3 pJ (with $\alpha = 1$). Therefore, the average value indicates an experimental switching activity α around 0.23. In the same way, energy results

for an 80-bit 1 x 1 switch in [9] are reported, and quantified to 36.16 pJ. For a 128-bit 1 x 1 switch in *Nostrum*, an average value of $(236 / 5) = 47.2$ pJ is measured, which is around 1.3 times higher. However, it should be observed that the packet size we are dealing with in *Nostrum* is 1/3 higher, and that we do not know the value of switching activity related to the simulations in [9].

7. Conclusion

We have presented a study on power consumption in *Nostrum* based on a power model for links and switches. *Nos-HPM* has been validated with Synopsys Power Compiler and has then been integrated in the *Nostrum* SystemC based simulator. The accuracy of *Nos-HPM* and its insignificant simulation overhead allow fast simulations and various kinds of analysis. The experimental results show the strong impact that interconnection links have on the overall NoC power consumption with a ratio of switch power to link power around 1. The simulations show that the total power consumption of a 4 x 4 *Nostrum* network is approximately 0.7W and of a 8 x 8 network is around 3.5W in 180nm UPC CMOS technology.

References

- [1] L. Benini and G. De Micheli. Networks on Chips: A New SoC Paradigm. *IEEE Computer*, 35(1):70-78, January 2002.
- [2] A. Jantsch and H. Tenhunen. *Networks on Chip*. Kluwer Academic Publishers, 2003.
- [3] E. Nilsson. Design and Implementation of a Hot-potato Switch in a Network on Chip. Master's thesis, Department of Microelectronics and Information Technology, Royal Institute of Technology, IMIT/LECS 2002-11, Stockholm, Sweden, June 2002.
- [4] W. Stallings. *Data and Computer Communications*. Prentice Hall International Editions, 4 edition, 1994.
- [5] T. T. Ye, L. Benini, and G. De Micheli. Analysis of Power Consumption on Switch Fabrics in Network Routers. In *Proceedings of Design Automation Conference*, 524-529, June 2002.
- [6] Z. Lu, L. Tong, B. Yin, and A. Jantsch. A Power Efficient Flit-admission Scheme for Wormhole-switched Networks on Chip. In *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*, July 2005.
- [7] A. Vitkovski. A Study on Power Consumption in the Nostrum Communication Network. Master's thesis, Institute of Microelectronics and Information Technology, Royal Institute of Technology (KTH), IMIT/LECS 2004-22, Stockholm, Sweden, April 2004.
- [8] *Synopsys Power Compiler User Guide*, Release V-2004.06, June 2004.
- [9] S.-J. Lee, K. Lee, and H. Yoo. Analysis and Implementation of Practical Cost-Effective Network-on-Chips, Sep-Oct 2005.