A High Abstraction, High Accuracy Power Estimation Model for Networks-on-Chip

Luciano Ost[1] ,Guilherme Guindani[1], Leandro Soares Indrusiak[2] ,Cezar Reinbrecht[1],
Thiago Raupp[1], Fernando Moraes[1]

[1] FACIN-PUCRS - Av. Ipiranga 6681- Porto Alegre - 90619-900 – Brazil
{luciano.ost, guilherme.guindani, fernando.moraes}@pucrs.br

[2] Department of Computer Science - University of York YO10 5DD, York, United Kingdom
lsi@cs.york.ac.uk

ABSTRACT

Due to the vast number of alternatives in the design space of NoC-based MPSoCs, fast and accurate performance evaluation approaches can result in earlier - and often better - design decisions. Important design metrics for mobile embedded systems include power dissipation and energy consumption. To speed-up the evaluation of such metrics, state-of-the-art research proposes abstract models of the NoC interconnect, employing, for example, TLM SystemC, analytical descriptions and graph descriptions. Power parameters used at higher abstraction models (e.g. TLM) frequently rely upon data generated at lower abstraction levels (e.g. RTL). This paper presents an abstract model of a NoC coupled with a power estimation model, aiming to provide accurate estimations early on the design flow. Despite being abstract, this model considers typical NoC communication behavior such as congestion and burst transmissions, leading to accurate results compared to industrial tools. A proof-of-concept implementation using the Ptolemy II framework demonstrates the strength of this approach, showing that it is possible to use abstract models to estimate power and energy without incurring excessive accuracy loss. Other benefits of abstract modeling are increased system observability and simplicity of design space exploration. System observability is demonstrated with a graphic tool enabling the visualization of the power dissipation at run-time.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles – advanced technologies, VLSI (very large scale integration).

General Terms

Design, Experimentation, Performance, Verification.

Keywords

Power Modeling, Networks-on-Chip, High Abstraction Modeling.

SBCCI'09, August 31st - September 3rd, 2009, Natal, RN, Brazil.
Copyright 2009 ACM 978-1-60558-705-9 $5.00.

1. INTRODUCTION

Modelling at higher abstraction levels is a common practice to increase and simplify development and validation of complex systems as MPSoCs. The simulation speed, the improved observability, and debugging capabilities provided by higher-level models reduce design space exploration time. The accurate performance evaluation can be achieved by calibrating the high-level model using a reference design model, for instance an RTL implementation [1].

Mobile embedded systems have a limited power budget that must be efficiently used. As outlined in [2], one of the primary cost functions in design space exploration of MPSoCs is the power dissipation. Due to the large simulation time and amount of memory required by the power estimation tools, simple and accurate high level models became necessary to achieve acceptable results within the time-to-market frame of complex systems.

The research work reported in this paper covers the integration of a power estimation model into an abstract model of a NoC-based MPSoC, aiming to enable fast design space exploration and to provide an accurate estimation of the power dissipated by the NoC on each design alternative. The modeling and simulation framework that was chosen as the foundation of the proposed approach is actor-orientation, for its support to heterogeneous models of computation, which in turn enabled us to integrate the multiple abstraction levels that are required for power estimation. Another important contribution of this work is the increased NoC observability, enabling the visual analysis of the power dissipation during the simulation using a graphical interface.

This paper is organized as follows. Section 2 describes related works in high-level NoC power estimation models. An overview of the power estimation model is presented in Section 3. Section 4 presents the integration of the power estimation model into an actor-oriented model. Section 5 presents results, including model accuracy and simulation time. Finally, Section 6 points out conclusions and directions for future work.

2. RELATED WORK

Hu et al. [3] presented an energy estimation model based on the traffic flow in the NoC's building blocks (routers and interconnection wires). The authors make use of the bit energy concept [4], which represents the amount of energy consumed in the transmission of a data bit throughout the NoC (in its routers and interconnection wires). This model evaluates the energy consumption in an end-to-end transmission only. Equation 1 describes the model energy consumption estimation, for a single data transmission between two points of the NoC.

$$E_{bit}^{hops} = n_{hops} \times E_{S_{bit}} + (n_{hops} - 1) \times E_{L_{bit}} \qquad (1)$$

In the equation above, E_{Sbit} is the energy consumption in one router; E_{Lbit} is the energy consumption of the interconnection wires; and n_{hops} is the number of routers used in the data bit transmission.

Lee et. al. [5] proposed a power estimation framework for SoCs, using power profiles to produce cycle accurate results. The SoC is divided in its building blocks (e.g. processors, memories, communication and peripherals) and the power estimation is based on the RTL analysis of each component. The authors validate the framework using an ARM926EJ-S CPU and the AMBA AXI 3.0 as NoC. Results have a maximum error of 10% compared with a gate-level power evaluation, and an average error of 5%. Speed-up compared to a gate level simulation is in average 100 times faster.

Beltrame et al. [6] developed a SoC power estimation method based on SystemC TLM modeling strategy. It adopts multi-accuracy models, supporting the switch between different models at run-time according to the desired accuracy level. The authors validate their model using the STBus NoC, and an analytical power model of this NoC. An MPEG4 application was tested, achieving up to 82% speed-up compared to TLM BCA (Bus-Cycle Accurate) simulation.

Koohi et. al. [7] presented a NoC power and performance analysis with different traffic models, using analytical models. The authors targeted a NoC with a mesh topology. The employed traffic models are: uniform, local, hot-spot and matrix transpose. Results were compared to Synopsys Power Compiler and Modelsim, showing an error of 2% for power estimation and 3% for throughput.

Atitallah et. al. [2] uses a stack of abstract models. The higher abstraction model, named Timed Programmer View (PVT), omits details related to the computation and communication resources. Such abstract model enables designers to select a set of solutions, to be explored at lower abstraction levels. The second model, CABA (Cycle-Accurate Bit-Accurate), is used for power estimation and platform configuration. Results present an error of 8% in power estimation compared to a physical measure and 17% of simulation speed-up.

Eisley et. al. [8] employ a framework that takes as input message flows, and derives a power profile of the network fabric. Authors map the CPU datapath as a graph, and the application as a set of messages that flow in this graph. Those mapped CPUs are connected into the network fabric, mapping the entire MPSoC as a network. The authors make use of a network power estimation tool, called LUNA, to evaluate the power dissipation of the entire MPSoC.

Most of those approaches calibrate the high-level model with parameters extracted from RTL implementations, just as the reference NoC power model used in this work and described in the next section. To the best of our knowledge, the present work is the first NoC power estimation model that allows accurate power analysis using a simplified NoC model based on actor-orientation. An advantage of using actor-oriented design is the possibility to jointly validate the NoC model with complex applications using multiple models of computation [9]. Another advantage of the present approach is the possibility of joint validation of applications (modeled as UML sequence diagrams) mapped onto the platform model, considering power constraints early at the design process [10].

3. RATE-BASED POWER MODEL ESTIMATION

Volume-based models estimate the average power as a function of the total transmitted data. Therefore, these models do not capture low-level effects, such as congestion and burstiness, being simple but inaccurate. At the other side, models derived from electrical simulation are accurate, but too complex to be integrated into abstract models. The adopted power model is a trade-off between such approaches: data volume is considered, but computed as a transmission rate inside a given sample period; and accuracy is guaranteed from a physical calibration step, which defines the power dissipation for each transmission rate.

Rate-based power estimation model, introduced in [11], comprises two steps: calibration and application. The calibration step defines the relevant model parameters. This step starts with the synthesis of one NoC router, generating a mapped HDL description that replaces the original router. This NoC description is then simulated with different traffic scenarios, each one with a fixed injection rate. The switching activity of each simulation is used as input for Synopsys PrimePower estimation tool, which computes the average power dissipation of each router element: (*i*) buffers, responsible for at least 80% of the average power dissipation [12]; (*ii*) internal crossbar and (*iii*) control logic.

After the calibration phase, a power dissipation table is generated for each injection rate and for each router element. Using linear approximation, an equation that gives the power dissipation as a function of the injection rate is obtained for each table.

In the application step, the NoC is simulated to obtain the reception rate at each buffer. This is measured with monitors inserted at each router buffer. The monitors count the amount of flits received in a parameterizable sample window. For each reception rate, the associated power dissipation (*Pbuffer*) is annotated, applying the equations obtained in the calibration step. The power dissipation of the control logic (*Pcontrol*) and the crossbar (*Pcrossbar*) are obtained using the average buffers reception rate. The power dissipation of a router is given by Equation (2), where *m* represents the number of buffers present in the router and *n* is the number of sampling periods.

$$P_{avg} = \sum_{k=1}^{m} \frac{\sum_{i=1}^{n} Pbuffer_{k_i}}{n} + Pcrossbar + Pcontrol \qquad (2)$$

Different traffic scenarios may be used without a new calibration, which is only repeated if some structural NoC parameter changes (e.g. buffer depth, clock frequency). Congestion and burst effects are implicitly taken into account, since such effects change reception rates. The rate-based model has an average error of 5% when compared to a commercial power estimation tool (Synopsys Prime Power). The model is faster than commercial tools, because it is not necessary to generate the switching activity for the entire NoC. For example, the CPU time to simulate a 4x4 NoC and evaluate its power is less than 10 minutes, while a commercial tool took more than 14 hours.

4. ACTOR-ORIENTED MODEL

Actor-orientation is a modeling and simulation framework supporting multiple models of computation (MoCs). While most simulators and simulation languages (e.g. Matlab/Simulink, VHDL, Spice, GPSS) follow a predefined MoC (e.g. discrete events, continuous time, untimed dataflow), actor-oriented tools allow the user/designer to specify the desired MoC for each part of the system that is being designed. This feature is particularly interesting for cross-abstraction models, as each abstraction can be modeled using the most adequate MoC.

In this work, we use an actor-oriented tool – UC Berkeley's Ptolemy II - to create an abstract model the NoC structure and the processing elements connected to it. The processing elements inject traffic into the network (it supports traffic obtained from an application model [10], from execution traces of a real application or synthetically generated traffic). The NoC structure, which we assume to have mesh topology and wormhole switching, is composed by buffers, arbiters, routers and wires, all modeled as actors, which communicate by exchanging tokens.

To increase abstraction – and thus accelerate simulation - we use a technique to reduce the number of communication tokens exchanged by the actors of the NoC model. Such tokens represent the NoC communication events, and usually represent the flit-by-flit transmission of data from one node of the NoC to the next one. In this work, we abstract away the flit-by-flit transmission of the packet payload by using the *Payload Abstraction Technique* (PAT), introduced in [13]. Such technique is based on: (*i*) packet abstraction, modeled as only a header and a trailer; (*ii*) buffer abstraction, modeled as a FIFO structure; (*iii*) analytical method to estimate the packet trailer release time. This abstraction eliminates the need to simulate the flit-by-flit payload transfer. Packets are forwarded from one NoC node to the next according to the packet trailer release time (*ptrt*), which is defined by Equation 3.

$$ptrt = hft + pcksize * ctf \qquad (3)$$

where:

hft:	*header forwarding time*
pcksize:	*packet size (number of flits)*
ctf:	*number of clock cycles to transmit one flit*

The example depicted in Figure 1 considers a 21-flit packet being transmitted between routers R1 and R5, without congestion, with *cft* equals to 1 (credit-based control flow) and *hft* equals to 7 (number of clock cycles required to execute arbitration and routing). As shown in the Figure, *hft* increases 7 clock cycles at each hop, and *ptrt* is updated according to Equation 3. The last *ptrt* value, 56, corresponds to the packet latency, and this value is the same as the RTL implementation.

PAT allows simulating unblocked and blocked packet transmission scenarios. If no resource conflicts occur (unblocked scenario), latency and throughput of the NoC can be measured with no loss of accuracy. In a blocked scenario, when a header packet arrives in an input buffer, two blocking situations can occur: (*i*) the desired output port is reserved to another input port or (*ii*) the target neighbor input buffer is not able to receive a header or a trailer of the packet. In both cases the header blocks, and the trailer continues to follows its path, considering the buffer size. The use of PAT can considerably accelerate the simulation of NoC-based MPSoCs, and the results obtained by [13] show that the error for the worst-case throughput is 0.1% when compared to an RTL model simulating flit-by-flit transmission.

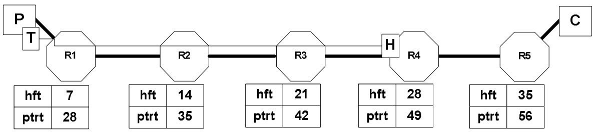

Figure 1 - Estimated release times regarding blocking-free delivery scenario.

To integrate the rate-based power estimation model into an actor-oriented model of a NoC using PAT, a number of updates were necessary on two of the actors that compose each node of the NoC (depicted in the lower left side of Figure 2): (*i*) input buffers to receive data; (*ii*) an arbiter, which is responsible for routing the incoming data packet through enabled channels.

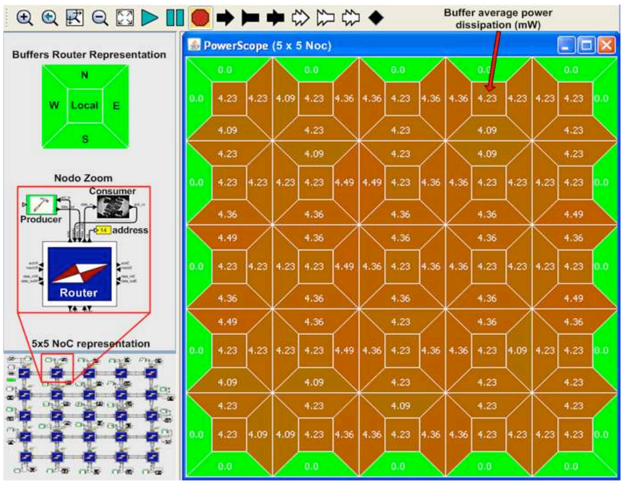

Figure 2 - Ptolemy II showing a 5x5 NoC and the PowerScope.

The three main improvements to the original actor-oriented model are:

1. A counter in each router buffer computes the number of received packets, within a predefined sample window period, named *recPkts*. Due to the absence of payload transmission, the counter is incremented when a packet trailer is received.
2. Each buffer computes its reception rate – *avbrr*, according to Equation 4, where: *phit*, is the phit size; T, the clock period; and *sw* the sample window, in clock cycles.

$$avbrr = \frac{recPkts \times pktSize \times phit}{T \times sw} \qquad (4)$$

3. A monitor collects the average reception rate of each buffer during the simulation. At the end of each sample period, the *avbrr* value is sent to PowerScope.

PowerScope is a parameterizable actor developed to display graphically the power dissipation. PowerScope is illustrated in the right side of the Figure 2, which shows during run-time the average dynamic power dissipation of each buffer, increasing system observability. This feature can help designers to detect power hotspots, enabling, for example, different application mapping targeting low-power budget. PowerScope generates a power report, with the maximum, minimum, average power per router and the energy consumption. PowerScope requires the following power parameters: (*i*) switch control base dissipation; (*ii*) switch control variable dissipation; (*iii*) buffer base dissipation; (*iv*) buffer variable dissipation. Besides the improved system observability, the proposed approach allows the designer to quickly analyze, early at the design process, different configurations for the NoC, aiming to satisfy the particular requirements of performance and power dissipation for a given application, as mentioned before.

5. RESULTS

This Section applies the techniques described in this paper to create an abstract model of the Hermes NoC using the rate-based analysis presented in Section 3 and the actor-oriented modeling approach covered in Section 4. The following results were analyzed: (*i*) comparison among Synopsys PrimePower, rate-based model (Section 3) and volume-based model (based on Hu's approach [3]); (*ii*) power estimation and energy consumption model accuracy of the actor-oriented PAT-based model when compared to a cycle-accurate RTL model; (*iii*) execution time of the PAT-based model.

5.1 Comparison of Power Estimation Models

The first evaluation scenario considers a set of traffic flows that generate congestion in the NoC channels. The experimental setup employs a 3x3 Hermes NoC with 16-bit flit width, 16-flit buffers, and 16-flit packets are injected into the network by every router. Routing follows the XY routing algorithm. Two different injection rates are applied: (*i*) 1000 injected packets per router, at 120 Mbps (15% of the maximum link injection rate); (*ii*) 5000 packets injected per router, at 400 Mbps (50% of the maximum link injection rate). Table 1 shows the power estimation values using Synopsys PrimePower, rate-based model and volume-based model. Considering Synopsys PrimePower as reference, Table 2 shows the evaluation error between the two models.

Table 1 - Average power dissipation results using a commercial power estimation tool (PrimePower), rate-based model, and volume-base model (NoC frequency: 50MHz).

Traffic	1000 packets @ 120 Mbps	5000 packets @ 400 Mbps
PrimePower	283,00 mW	288,00 mW
Rate-Based	299,30 mW	299,91 mW
Volume-Based	405,49 mW	442,60 mW

Table 2 - Error results comparing power estimation models to a commercial power estimation tool (PrimePower).

Traffic	1000 packets @ 120 Mbps	5000 packets @ 400 Mbps
Rate-Based Error	5,76%	4,14%
Volume-Based Error	43,28%	53,68%

This experiment shows that the error induced by the volume-based power estimation model can be superior to 50%, when compared to the Synopsys PrimePower tool. The rate-based model maintains the error below 6% in the same comparison. The rate-based power model presents such a small difference to the reference estimation because it considers blocked packets and burst transmissions, effects due to the congestion over the network.

The second evaluation scenario estimates the execution time of the power estimation models. A similar experimental setup is used, with each processing element transmitting 10,000 packets, random spatial packet distribution, with an injection rate equals to 25% of the available link rate. Total power estimation time was approximately 20 hours with PrimePower, and less than 20 minutes with the rate-based model (Intel Core2 Duo 2.4 GHz, 2GB RAM). For the same traffic scenario, using a 4x4 NoC, the power estimation with PrimePower becomes unfeasible. It is important to mention that the volume-based model computation time is almost zero, since it corresponds to the application of simple equations. However, volume-based model can only be applied in situations with a small number of collisions between packets (absence of congestion), an unrealistic scenario for NoCs.

5.2 Comparison between RTL and Actor-Oriented Models for Power/Energy Estimation

This section compares two implementations of the rate-based model, using two abstraction levels, RTL and actor oriented. The experimental setup uses a 4x4 2D-mesh NoC running at 50 MHz, with 16-bit flit width, 8-flit buffer depth, XY routing algorithm and handshake control flow. The maximum link rate is 800 Mbps. The following traffic parameters vary:

- packet size: 32 and 64 flits;
- temporal traffic distribution: uniform (200 Mbps), normal (minimal rate 150Mbps, maximal rate 250Mbps, and standard deviation 10Mbps), and Pareto on/off (200 Mbps, maximum number of bursts set to 10 packets);
- spatial traffic distribution: complement and random;
- number of packets: 100 (traffic T1), 1,000 (traffic T2), and 10,000 (traffic T3).

Table 3 presents the difference in the average power dissipation between the actor-oriented model (Section 4, referenced here as JOSELITO) and the RTL model (reference power model, Section 3). Note that the traffic scenarios induce network congestion, which implies in blocked packets. Even with injection rates near to the network saturation point (mesh networks saturate when injection rates between 20% and 30%) and collision between packets, both implementations of the rate-based model at the RTL and actor-oriented abstraction level present similar results. The clear advantage of the actor-oriented model is its faster construction, validation and debugging, enabling faster population and exploration of the design space. Such accurate power evaluation at higher abstraction level is possible because the estimation model is based on the buffer reception rates, sampled at fixed periods, and this can be properly captured at the actor-oriented model.

Table 3 - Average Power Dissipation difference between Model RTL and JOSELITO, using random (R) and complement (C) traffic distribution. T1, T2, T3 means 100, 1000 and 10,000 packets with 32 and 64 flits.

		Uniform Distribution			Normal Distribution			Pareto Distribution		
		Model RTL (mW)	JOSELITO (mW)	Difference (mW)	Model RTL (mW)	JOSELITO (mW)	Difference (mW)	Model RTL (mW)	JOSELITO (mW)	Difference (mW)
T1 (R)	32	302,35	302,35	3,22E-06	303,07	303,07	3,64E-06	288,77	288,77	1,11E-06
	64	303,23	303,23	5,02E-06	303,20	303,20	5,94E-06	289,05	289,05	3,20E-06
T1 (C)	32	311,25	311,25	5,06E-06	311,25	311,25	4,78E-06	292,23	292,23	3,26E-06
	64	311,86	311,86	5,65E-06	311,86	311,86	5,89E-06	292,64	292,64	4,50E-06
T2 (R)	32	303,94	303,94	2,81E-06	303,89	303,89	3,45E-06	289,02	289,02	1,85E-06
	64	303,86	303,86	4,69E-06	303,93	303,93	5,81E-06	289,65	289,65	3,95E-06
T2 (C)	32	312,88	312,88	5,02E-06	312,75	312,75	4,78E-06	293,70	293,70	4,57E-06
	64	312,95	312,95	5,37E-06	312,88	312,88	5,43E-06	293,64	293,64	5,17E-06
T3 (R)	32	303,99	303,99	3,06E-06	303,94	303,94	3,48E-06	288,99	288,99	1,89E-06
	64	303,99	303,99	4,78E-06	303,96	303,96	5,88E-06	289,61	289,61	3,91E-06
T3 (C)	32	313,00	313,00	5,03E-06	312,96	312,96	4,88E-06	293,05	293,05	4,56E-06
	64	312,17	312,17	5,26E-06	312,95	312,95	5,38E-06	293,25	293,25	5,09E-06

Table 4 - Average Energy Dissipation difference between Model RTL and JOSELITO, using random (R) and complement (C) traffic distribution. T1, T2, T3 means 100, 1000 and 10,000 packets with 32 and 64 flits.

		Uniform Distribution			Normal Distribution			Pareto Distribution		
		Model RTL (mJ)	JOSELITO (mJ)	Difference (mJ)	Model RTL (mJ)	JOSELITO (mJ)	Difference (mJ)	Model RTL (mJ)	JOSELITO (mJ)	Difference (mJ)
T1 (R)	32	163,27	163,27	1,74E-06	163,66	163,66	1,96E-06	381,17	381,17	1,47E-06
	64	321,42	321,42	5,32E-06	321,40	321,40	6,29E-06	745,76	745,76	8,27E-06
T1 (C)	32	168,07	168,07	2,73E-06	168,07	168,07	2,58E-06	385,75	385,75	4,31E-06
	64	330,57	330,57	5,99E-06	330,57	330,57	6,24E-06	749,18	749,18	1,15E-05
T2 (R)	32	1562,26	1562,26	1,44E-05	1568,10	1568,10	1,78E-05	3693,74	3693,74	2,36E-05
	64	3123,69	3123,69	4,82E-05	3124,46	3124,46	5,97E-05	6974,82	6974,82	9,52E-05
T2 (C)	32	1608,23	1608,23	2,58E-05	1613,81	1613,81	2,46E-05	3489,16	3489,16	5,43E-05
	64	3210,88	3210,88	5,51E-05	3216,46	3216,46	5,59E-05	6982,90	6982,90	0,000123
T3 (R)	32	15570,79	15570,79	0,000157	15586,36	15586,36	0,000178	37135,51	37135,51	0,000243
	64	31141,22	31141,22	0,000490	31187,08	31187,08	0,000603	70098,26	70098,26	0,000946
T3 (C)	32	16038,35	16038,35	0,000258	16048,84	16048,84	0,000250	36415,41	36415,41	0,000566
	64	32778,61	32778,61	0,000553	32108,84	32108,84	0,000552	71859,65	71859,65	0,001247

The difference in the average energy consumption between JOSELITO and RTL model was evaluated as well, as reported in Table 4. Applying packets with 32 and 64 flits through the NoC, the worst-case difference is presented when 10000 packets (Pareto on-off distribution), are sent per producer over the NoC. In this case, the difference on the average energy consumption to deliver all packets is only 0,001247 mJ between both models (in practice, an error of 0%). Because JOSELITO abstracts the buffer size, increasing the packet size (considering the same buffer depth) leads to a larger error on the average energy consumption. For example, considering Pareto on-off traffic distribution, 8-flit buffer depth and a packet size of 32 and 64 flits, the worst-case difference increases 0,000681 mJ to deliver 160,000 64-flits packets (16 routers delivering each one 10,000 packets). The difference can also be considered insignificant, taking into account that even assuming a packet size 8 times bigger than the buffer depth the error is still in practice 0%.

5.3 Simulation Time

The last set of results shows the speed-up obtained using the actor-oriented model, w.r.t the RTL model considering traffic with small number of packets (e.g. T1, 100 packets per producer), as demonstrated in Table 5. In these simulated scenarios, the actor-oriented model was faster than RTL model.

For the traffic scenarios T2 and T3, the simulation of the RTL model is, in average, 2.6 times faster than the actor-oriented model. The number of necessary simulation events to deliver all packets (assuming that one Ptolemy II simulation cycle corresponds to a RTL clock cycle) explains the increased simulation time. This is probably because of the fact that our actor-oriented model is implemented on top of PtolemyII, which is a Java application running on interpreted mode on top of a virtual machine with restricted heap space and managing memory using garbage collection. All such implementation aspects contribute to the simulation slowdown (the simulation server has to spend much of its processing capacity on memory management rather than on the simulation process itself). This is acceptable as a *proof-of-concept*, but additional work on compiling PtolemyII to native code and reducing the memory management overhead is needed to make the proposed technique competitive in simulation time, in comparison with current commercial tools. Another alternative to solve the memory management problem would be to reimplement this approach using simulation frameworks based on C++ (such the multi-MoC extensions for SystemC done by Patel and Shukla [14]).

Table 5 - Speed up of actor-oriented power model in comparison to RTL power model for 3 traffic distributions with 100 packets.

Power model\Traffic	Uniform	Normal	Pareto
RTL	45 sec.	45 sec.	49 sec.
Actor-oriented	26 sec.	27 sec.	28 sec.
Speed-up Factor	1,7307	1,6666	1,75

6. CONCLUSIONS AND FUTURE WORKS

The most promising technique to explore the complex design space of NoC-based MPSoC platforms is to build simpler, more abstract models of the platform components, and to evaluate the different compositions and alternatives regarding performance and power consumption. The accuracy and speed of such evaluation must be high, and the effort to build and compose such models must be very low, so that they can provide meaningful results early on the design flow.

This paper addressed an import issue in this scenario: the accuracy of power estimation. It was shown that using simple volume-based models to estimate power in NoC-based systems can lead to substantial error as those models that abstract away the congestion on the network. By employing abstract models that consider the actual behavior of packet transmission over multi-hop networks, it is possible to obtain accurate results which are comparable to those obtained using commercial RTL power evaluation tools.

The main contribution of this paper is the identification of the elements of a NoC which can be abstracted away (and those that cannot) so that a rate-based power estimation analysis can still be performed. The major deliverable is the set of modeling techniques that can be used to create simplified models of NoCs that are easier to design, setup, debug and visualize results (when compared to RTL or even TLM), and can produce accurate figures for performance (as shown in [13]) and for power consumption (as shown in this paper). By using such techniques, we could greatly accelerate the power estimation of large NoC models (from hours to minutes, if comparing with RTL models) without any loss of accuracy. Still comparing against RTL models, it must be said that the observed simulation speed-up of the proposed models is not significant, since we are comparing a proof-of-concept tool executed over a virtual machine with commercial tools that are compiled to native code and that have been exhaustively optimized over the past decades. In any case, the obtained results point to promising future work on optimizing memory management of the simulator and the further reduction of simulation events (for instance, by abstracting away negative *acks* on the flow control, or the internals of the arbitration algorithm). Additional work will also be done on extending the power estimation model so that it considers also the power dissipation due to the switching activity over the links connecting routers, and in the router buffers.

7. ACKNOWLEDGMENTS

This research was supported partially by CNPq (Brazilian Research Agency), projects 142049/2008-5 and 471134/2007-4.

8. REFERENCES

[1] Pimentel, A. D.; et. al. **Calibration of abstract performance models for system-level design space exploration**. J. of Signal Processing Systems, 50(2), 2008.

[2] Atitallah, R. B.; et. al. **MPSoC power estimation framework at transaction level modeling**. In: Int. Conference on Microelectronics (ICM'07), 2007.

[3] Hu, J.; Marculescu, R. **Energy-aware mapping for tile-based NoC architectures under performance constraints**. In: Asia South Pacific Design Automation Conference (ASP-DAC'03), 2003.

[4] Ye, T.; et. al. **Analysis of Power Consumption on Switch Fabrics in Network Routers**. In: Design Automation and Conference (DAC'02), 2002.

[5] Lee I.; et. al. **PowerViP: SoC power estimation framework at transaction level**. In: Asia and South Pacific Conference on Design Automation (ASPDAC'06), 2006.

[6] Beltrame, G.; et. al. **Multi-Accuracy Power and Performance Transaction-Level Modeling**. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 26(**10**), 2007.

[7] Koohi, S.; et. al. **High-Level Modeling Approach for Analyzing the Effects of Traffic Models on Power and Throughput in Mesh-Based NoCs**. In: International Conference on VLSI Design (VLSID'08), 2008.

[8] Eisley, N.; et. al. **Low power: High-level power analysis for multi-core chips**. In: Int. Conference on Compilers, Architecture and Synthesis for Embedded Systems, 2006.

[9] Lee, E. A.; et. al. **Actor-Oriented Design of Embedded Hardware and Software**. Systems, Journal of Circuits, Systems, and Computers, 12 (**3**), 2003.

[10] Määttä, S.; et. al. **Validation of Executable Application Models Mapped onto Network-on-Chip Platforms**. In: IEEE Symposium on Industrial Embedded Systems (SIES'08), 2008.

[11] Guindani, G. M.; et. al. **NoC Power Estimation at the RTL Abstraction Level**. In: Annual Symposium on VLSI Design (ISVLSI'08), 2008.

[12] Palma J.; et al. **Inserting Data Encoding Techniques into NoC-Based Systems**. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI'07), 2007.

[13] Ost, L; et. al. M. **A Simplified Executable Model to Evaluate Latency and Throughput of Networks-on-Chip**. In: Symposium on Integrated Circuits and Systems Design (SBCCI'08), 2008.

[14] Patel, H. D., et. al. **SystemC Kernel Extensions for Heterogeneous System Modeling**. Kluwer Academic Publishers, 2004.