System level modeling of Networks-on-Chip for power estimation and design space exploration

Martin Gag, Tim Wegner, Philipp Gorski, Andreas Tockhorn, Dirk Timmermann
Universität Rostock, Institut für Angewandte Mikroelektronik und Datentechnik, Rostock
martin.gag@uni-rostock.de

Abstract

With a fast rising productivity and even faster rising integration densities, i.e., design-productivity-gap, energy and power dissipation are critical topics in high level system design more than ever. Thermal aware system design, reliable power delivery, and the overall energy dissipation are only few crucial design properties. In this work we present a framework based on SystemC, enabling the modeling and simulation of many-core systems reverting to Networks-on-Chip as their communicational infrastructure. The transaction level communication model is clock cycle accurate, yielding a fast yet concise functional simulation. The framework is enriched by parameters concerning technology node and floorplanning and by a thermal model of the eventual chip. Thereby, power estimation and on-chip temperature distribution can be evaluated in an early design phase. Furthermore, the framework is supplemented by extensions enabling an extensive and detailed design space exploration, namely proactive thermal management and thermal aware task mapping.

1. Introduction

Due to the design productivity gap it becomes more and more important to incorporate precise but fast system models. This applies for functional and transaction level simulation as well as power and energy estimation. As power densities are growing due to technology scaling further models including thermal parameters are needed in early design space exploration. In this paper we propose a model that combines a cycle accurate transaction model, detailed power backannotations, a model of thermal distribution and system management algorithms for system control and task mapping. Our model can be populated with synthetic task graphs that efficiently emulate real world applications. With a varying degree of parallelism in the task graph different algorithmic problems are modeled and can be evaluated utilizing the proposed model. This facilitates not only performance analysis, but also allows for exploration of power, energy efficiency and thermal design aspects.

In the following section we will give a brief description of the proposed cycle accurate transaction model and the many-core system we assume. We show exemplary results proving the speedup compared with register transfer level (RTL) simulation. The power model we integrated is described in Section 3. Here we report about the backannotation of NoC elements and the two power modes of the modeled IP-cores. Additionally we show exemplary results of router design space exploration and energy efficiency of different NoC sizes and task graphs. In Section 4 an approach to use our model for the investigation on power aware mapping strategies is shown. In the rest of this section we will briefly introduce exemplary simulation frameworks targeting a more or less holistic design space exploration for NoC-based systems.

In [KLPS12] Orion 2.0, a simulator focusing on power consumption and area requirements of architectural components of on-chip networks, is introduced with the objective of identifying tradeoffs between power, performance and area. The framework contains models for dynamic and leakage power for routers and links. Furthermore, it provides detailed modeling of microarchitectural router components and technological parameters as well. The most distinct differentiation compared to the work presented in this paper is the analytical approach compared to our more simulative strategy. Additionally, it has to be noted that Orion 2.0 is a pure power and area simulation framework. Thus, it lacks a functional simulation as well as consideration of consequences of power-oriented design decisions for performance and functional integrity.

Similarly, [EP04] presents an approach for power analysis of on-chip interconnection networks reverting to message flows in order to provide network power profiles. For this purpose, spatial variances across the network and temporal variance across application execution time are considered. The network traffic is modeled by synthetic message flows describing the link utilization between source and destination of a message. Besides that this approach does not include any performance analysis, it features only a rudimentary functional simulation of network traffic reverting to abstract description of message flows instead of application of more significant task graphs or communication patterns.

NIRGAM [NIR] is an extensible, discrete event, cycle accurate NoC simulator. It is aimed at the exploration of design space placing emphasis on topological, microarchitectural (i.e. switching, virtual channels, buffers, routing) and application-oriented (i.e. traffic patterns) issues. The analysis capabilities focus on performance providing data for different metrics (e.g. packet and flit latency, data throughput). Subsequently, [NIR] has been extended by the power model from [KLPS12] allowing for calculation of router power consumption. In spite of the integrated power model, no further consideration of power results, complementing this framework for functional simulation, is evident. There exist several similar NoC simulators [LTM$^+$05, NOX] primarily focusing on performance analysis.

Apparently, there is a lack of simulation frameworks combining high-level functional simulation and power modeling of on-chip interconnection networks, while accounting for analysis of performance and power issues and temperature distribution as well as considering impact of design decisions on all these fields at the same time.

2. Model of Networks-on-Chip on system level

The proposed cycle accurate SystemC-based NoC model is functionally identical with a synthesizable RTL implementation. The synthesizable model serving as a reference is implemented by using only the synthesizable subset of SystemC. However, the cycle accurate model uses some functions and structures from the TLM-2.0 standard, which is part of SystemC.

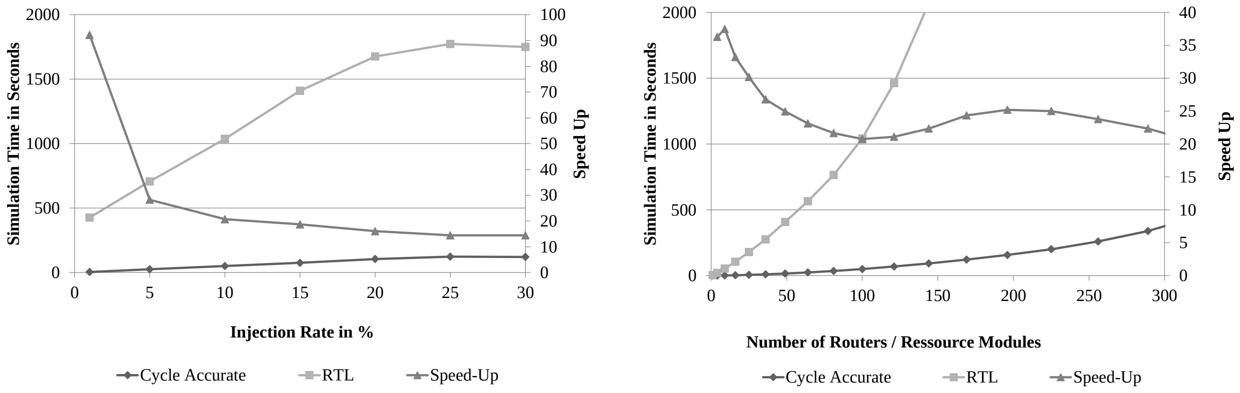
However, it abstracts to a functional description concerning functionality as well as communication, though preserving cycle accurate results. Therefore, this model uses different concepts. It abstracts signals from the RTL model to transactions. Furthermore, it simplifies state machines (FSMs) to processes, implementing these algorithms similar to software and replaces static sensitivities of the RTL implementation by dynamic sensitivities and wait statements implementing the advance of time.

A flit as the smallest unit, which influences the behavior of the NoC, is mapped to the concept of generic payload and is modeled by a transaction object. Since obtaining cycle accurate results is the goal, packets cannot be used as transaction objects, although this abstraction would increase simulation performance even further. As a result, it is much more convenient to add new options by using generic payload extensions.

To speed up management of transactions, a pool of transaction objects was designed. Whenever a new flit shall be generated, the corresponding source requests such an object from this pool. Eventually, it reaches its destination and is put back into the pool. In this way most of the constructor and destructor function calls are omitted. For special needs it is possible to configure the transaction pool in a way, to immediately release flits or to release unused flits after a certain time. Thus, memory usage of simulations can be reduced by the price of decreased simulation performance.

State machine implementations are abstracted to algorithmic descriptions. This applies especially to the FSMs necessary to implement the switching and routing algorithms.

Further acceleration of simulation was possible by omitting static sensitivities as well as a clock signal. Cycle accurate results are therefore obtained by inserting wait statements, which for example implement the delay caused by the processing of a routing algorithm or the time necessary to transmit a flit. Furthermore, processes are no more sensitive to a static list of signals, but to dynamic sensitivities on events or transactions instead. Consequently, processes are only invoked, when they really have to process relevant data. Especially the removed clock sensitivity saves processes of being invoked for many empty simulation cycles, just because the clock signal would have generated an event.



(a) Influence of injection rates on simulation time

(b) Influence of network size on simulation time

Figure 1: Simulation results to show the speed of our cycle accurate transaction model vs. a RTL model

A 10x10 2D mesh using wormhole switching and XY-routing at 10 % injection rate serves as the reference for the following measurements. This applies to the proposed simulator as well as to RTL simulations.

As it can be seen in Figure 1a the injection rate has a significant influence on the simulation speed-up of the cycle accurate approach. For very low loads (i.e. low frequency at which new flits are generated and injected into the network) a speed-up of up to 90× is possible. For increasing loads a speed-up of at least 7× was observed. Injection rates greater than 25 % do not change the behavior, since NoC saturation occurs and therefore flit generation rates have no influence on NoC traffic. During these congestion dominated simulation runs the cycle accurate approach obtained its lowest overall performance benefit of 14×. From the moment the NoC is saturated continuously increasing memory consumption can be observed. Since more flits are generated than released, the transaction pool has to generate new generic payload instances and therefore increases memory usage of these simulation runs.

Figure 1b depicts an investigation concerning the NoC size. It shows the influence of the number of routing modules in the simulated system. When more routers and IP-cores are in the network, more flits are generated at a constant injection rate of 10 %. In addition the number of events occurring is growing especially in the RTL simulation, because it uses static sensitivities. For very small networks a speedup of 36× can be observed. However, for bigger networks the speedup is 20 to 25×.

Parameters of the payload of flits (i.e. flit width, number of flits per packet) have no major influence on speedups. The same is true for the overall simulation length, since both simulation methods scale well over time. Just for very short simulation times under 1000 clock cycles the speedup reduces significantly, because initialization phase the cycle accurate model seems to be longer.

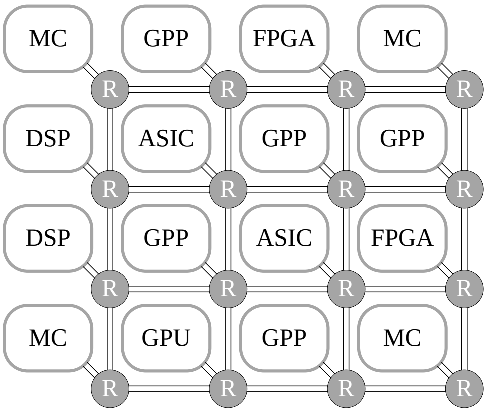


Figure 2: Example of the heterogeneous system model, containing general purpose processors (GPP), special purpose processors (DSP, GPU, ASIC), programmable hardware (FPGA) and Memory Controller (MC)

To build a model of many-core systems we added task graphs from a task graph generator, which is an extension of TGFF [DRW98]. This populates the NoC with different synthetic workload patterns. One simulation run contains a set of task graphs, each representing an application running on the system. It contains different tasks, that can be processed in parallel or depend on data from the preceding task to form serial processing. Tasks can be configured to be dependent on instructions or data from the memory controllers. The application can be configured to run one time or as a periodically repeating task graph. Each task has a specific amount of output data that is transferred via the NoC to the successors in the task graph. In that way different workload patterns can be configured, to match high or low data volume transactions.

The different tasks are processed on IP-cores connected to the NoC routers. To model a heterogeneous many-core system, different core types are defined (Figure 2). Our System-on-Chip can be configured to contain general purpose processors (GPP), application specific integrated circuits (ASICs), intermediate forms like signal or graphic processors (DSP, GPU), programmable hardware (FPGA) and Memory Controllers (MC). The tasks in the task graph can be restricted to run on a subset of the provided cores. However, the MCs are for providing the instructions or data needed by the tasks.

3. Power and thermal modeling

The power consumption of the system was modeled as a prerequisite for the evaluation of thermal distributions. Traditionally, most of the needed power of a chip is consumed by processing units (i.e. CPU, GPU). However, recently upcoming many-core systems show a huge contribution of communication subsystems [VHR$^+$08] to overall on-chip power consumption. To keep the model simple, power consumption of IP-cores was implemented by two power modes that represent active and idle states. During simulation the amount of time an IP-core is active, is calculated by the task graph annotations in combination with a mapping and scheduling algorithm. The remaining time it is assumed to be idle. These times are logged and multiplied by the average idle respective active power values. For the different types of cores (see Section 2) different power values are provided, however for now these are just estimates and need to be refined for a holistic system model approach.

Emphasis is placed on the modeling of the communication infrastructure. The power dissipation of the NoC is characterized by routers and interconnections. The former consist out of buffers, routing logic, and crossbar, while interconnections comprise wires and drivers.

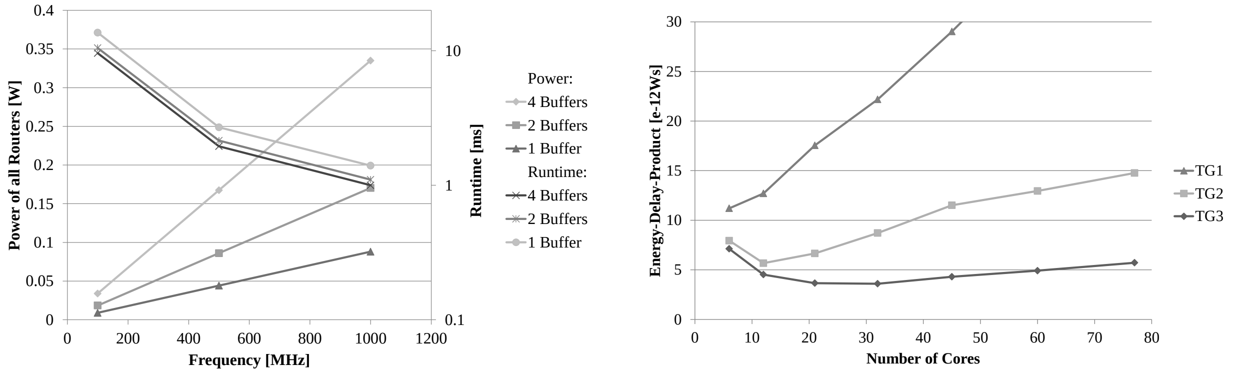
For the power consumption model of the NoC a RTL design of the router is used. It is optimized for xy-wormhole switching, meaning that east and west ports cannot be accessed by north and south port. The design is parametrized and synthesized utilizing an automated design flow for three different technology nodes. Adjustable parameters are link width, buffer depth, and frequency (see Table 1). Because of a highly automated design flow, the individual designs are not fully optimized but the flow is fast instead and comparable results for design space exploration are obtained. The power per transmitted flit of a single router is automatically estimated on gate level by simulation as is the idle power. Header and body flits produce slightly different power values, since the routing logic is active, when a header is transmitted. When a body flit is transmitted the wormhole is already open, consequently there are less state registers to switch and the power consumption is lower. Because of ports optimized for xy-routing, the results for different ports slightly differ. The resulting power values were backannotated into the cycle accurate system model.

Table 1: Router characterization parameters

parameter	values
frequency	100 . . . 1000 MHz
buffer depth	1, 2, 4
link width	32, 64, 128 bit
technology nodes	65 nm, 45 nm, 32 nm

The results show that power dissipation of this design is buffer dominated in each case. A consideration of data correlation between the flits is discarded.

The power model of interconnections between the routers are modeled by a similar link model like it is used by ORION 2 [KLPS12]. It includes technology dependent capacitance estimation and a driver insertion algorithm estimating the dissipated power of the interconnections.



(a) Power of the communication subsystem and runtime of a task graph set for different buffer depth of NoC routers

(b) Energy delay product of various sized NoCs for three workloads differing in degree of parallelism

Figure 3: Exemplary simulation results concerning power and energy of NoCs

In Figure 3a results from power estimation are shown for router designs differing in the number of buffer stages. The power dissipation is corresponding to the number of buffer stages and to the working frequency. Dynamic power and the number of buffers are dominating the power breakdown of the router design. The tested task graph was configured to mimic a high traffic situation where the mean runtime of a task was only 0.5 us and each task provided 2 kB of output data on average. As a result it can be seen in Figure 3a that the runtime of the application not only depends on the working frequency but also on the number of buffer stages. As more buffers are provided, the packets traveling the NoC are blocking each other more infrequently and can be transferred faster.

Figure 3b shows an evaluation of three different task graph sets representing various application mixes. The three sets (TG1, TG2, TG3) differ in their degree of parallelism, where TG1 is the most sequential and TG3 is the most parallel task mix. To evaluate the efficiency of the system the energy delay product was used as a metric. While the most parallel task graph set was the fastest in functional simulation on all different NoC sizes, the energy delay product shows, that for varying application mixes different NoC sizes are optimal. The most parallel mix TG3 benefits the most when core number is around 30. The most sequential one TG1 is most efficient on a 3×3 NoC (the smallest tested), while TG2 is most efficient on a 4×4 NoC.

The origin of power dissipation and the location of thermal problems are tightly connected. However, heat can distribute over the system and become a global problem that needs to be monitored and managed. To model this distribution of temperature through the whole system we implemented a simple RC-Circuit in SystemC AMS representing the duality of electricity and temperature [WCG^{+}10]. This model is integrated into the SystemC functional NoC simulation (Section 2).

4. Application of the model: thermal aware mapping

In this chapter we will introduce an example application providing runtime thermal aware task mapping. For this purpose, the functional NoC simulation from Section 2 is combined with the TGFF-based task-graph model and the temperature model from Section 3 to allow for temperature-focused proactive task mapping.

The objective is to evaluate to which extent a thermal aware task mapping, which is capable of near-term prediction of temperature changes due to mapping of single tasks, is able to improve the temperature distribution of NoC-based many-core systems at runtime. The enhancement of a high-level NoC simulator (see Section 2) with TGFF-based task graph modeling enables emulation of runtime task mapping and its effects on system behavior. Furthermore, simulation of realistic workload and communication patterns is facilitated due to modeling of:

- Task dependencies (i.e. tasks rely on completion and/or input data from pending tasks)
- Amount of traffic caused by inter-task communication
- Processing time, overall runtime and time slices of single tasks
- Effort for management and coordination of task execution and inter-task communication (e.g. delay caused by resumption of paused tasks or data request from memory)

Finally, the inclusion of the thermal model (see Section 3) serves to predict the impact of runtime task mapping on the on-chip temperature profile. The input for temperature prediction is provided by the profiles of single tasks including task runtime, amount of processed data and traffic and the involved IP-cores, which are used to comprise the resulting traffic routes. The input data is delivered to the temperature model of the system, which is part of the proactive thermal aware mapping unit. This unit refers to the predicted temperature profile in order to identify the presumably coldest IP-core to which the next pending task has to be mapped. Subsequently, the mapping decision is forwarded to the actual functional simulation continuously providing activity statistics to another temperature profile. For an overview over the framework containing proactive thermal aware runtime task mapping see figure Figure 4. For the assessment of effectivity this approach

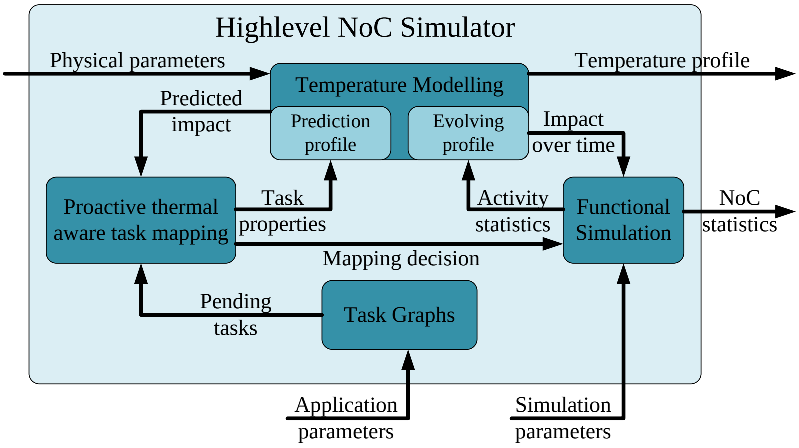


Figure 4: Scheme of high level NoC simulator with proactive thermal aware task mapping

is compared to three conventional mapping strategies. The first one is a reactive thermal aware task mapping considering the current temperature profile (i.e. the evolving profile from Figure 4), basically comparable to the thermal aware design-time approach from [KyCBB06]. The second one is a simple dynamic load balancing approach placing tasks on an appropriate IP-core with the lowest number of assigned tasks (also known as "first free" algorithm) like it is exemplified in [CRWG08]. The last one is a distance-oriented approach shortening the communication paths between interdependent tasks by placing them as close together as possible (also known as "nearest neighbor" algorithm) like it is done in [KSS11]. For these approaches the prediction profile of the temperature model not required. Note that before individual mapping directives take effect, the SoC is checked for appropriate idle IP-cores in order to utilize them first. All four mapping strategies are evaluated by consulting three different types of task graphs and NoC sizes ranging from 3×3 to 6×6. The three sets of task graphs are composed as follows:

- C1: large number of tasks, small amount of data and traffic, short execution times
- C2: small number of tasks, large amount of data and traffic, long execution time
- C3: average values for all parameters

Investigations focus on values for average temperature as well as maximum temperature difference measured across the chip during each time step of the simulation lasting 1 s of system runtime. In Table 2 the average on-chip temperature as well as the maximum temperature difference between all chip components for NoC sizes from 3×3 to 6×6 are depicted. In every case proactive task mapping outperforms at least one of the conventional approaches for average temperature or the maximum temperature difference, indicating a more homogenous temperature distribution. However, this advantage is achieved by a more or less acceptable difference between performances (i.e. higher or lower number of completed tasks) leading to average variations of -13 %, -12 % and 11 % for C1, -4 %, 3 % and 5 % for C2 as well as 0 %, 1 % and 7 % for C3 compared to "Dist", "# Tasks" and "Reactive"' (note that these values are averaged over the simulated NoC sizes). This means "Proactive" performs better than "Reactive" and worse than or equal to "Dist" and "# Tasks" mapping approaches. Task graphs with setups similar to that of C1 (i.e. short execution times and small amounts of traffic and data) exhibit worst results for proactive task mapping, since they do not allow for a sufficient prediction. This especially applies to maximum temperature difference and performance. Another interesting observation is that deviations between the approaches regarding average on-chip temperature (including potential advances of proactive thermal aware task mapping) diminish with growing NoC sizes. Further investigations regarding on-chip temperature imbalances concern the temperature difference between the router in the middle and on the edge of the NoC. Analysis of configurations C1, C2 and C3 for a 5×5 confirm that proactive task mapping is only appropriate for configurations with long execution time and large amounts of traffic and data, since for C1 average and maximum differences can only slightly be reduced compared to "Dist". For C2 and C3 the average reduction of temperature difference compared to the conventional approaches amounts to 0.2 °C and 1.0 °C, while the maximum reduction amounts to 0.7 °C and 2.5 °C respectively (note that these values are averaged over all approaches). Generally, proactive thermal aware task mapping is more suitable for task graphs exhibiting long execution times as well as large amounts of traffic and computational data and mainly can serve to reduce temperature imbalances while reducing possible performance penalties to a minimum. Since conventional approaches partially still yield better results or cause less performance impairment, further modifications of proactive thermal aware task mapping considering information regarding workload and communication paths need to be accomplished.

Table 2: Average on-chip temperature and maximum temperature difference for the four mapping strategies considering three different sets of task graphs as well as NoC sizes from 3×3 to 6×6

		Avg. Temp.				**Max. Temp. Delta**			
		Dist	# Tasks	Proactive	Reactive	Dist	# Tasks	Proactive	Reactive
C1	3×3	97.6	97.6	89.5	79.7	15.5	15.0	20.3	21.0
	4×4	83.7	84.2	79.7	75.3	11.6	9.7	21.7	21.7
	5×5	76.5	76.8	74.9	74.4	7.5	5.3	22.0	21.8
	6×6	72.2	72.4	71.8	72.3	6.4	3.6	21.8	21.1
	Avg	**82.5**	**82.8**	**79.0**	**75.4**	**10.2**	**8.4**	**21.4**	**21.4**
C2	3×3	96.8	85.0	89.6	83.5	18.7	13.9	16.0	21.2
	4×4	83.2	83.2	83.3	81.8	14.8	12.0	12.4	21.7
	5×5	76.1	76.1	76.1	76.2	11.0	8.5	8.5	15.5
	6×6	71.9	71.9	71.9	71.9	9.0	7.2	7.1	11.2
	Avg	**82.0**	**79.0**	**80.2**	**78.3**	**13.4**	**10.4**	**11.0**	**17.4**
C3	3×3	100.1	97.5	99.9	88.2	19.8	17.3	20.8	21.0
	4×4	85.4	85.6	85.7	84.2	17.4	11.4	17.0	21.5
	5×5	77.7	77.8	77.8	77.3	13.7	7.3	11.0	20.3
	6×6	73.1	73.1	73.2	73.0	11.6	5.4	7.5	19.2
	Avg	**84.1**	**83.5**	**84.1**	**80.6**	**15.6**	**10.3**	**14.1**	**20.5**

5. Conclusion

In this paper we introduced our work on modeling many-core systems with emphasis on their communication infrastructure. As not only performance but energy, power and thermal aspects have a major influence on the system design constraints a conclusive simulation framework was proposed and evaluated. We showed exemplary results demonstrating speed of simulation, power estimation for design space exploration, energy efficiency for different scenarios and as a sophisticated application of the model two thermal aware mapping approaches.

As an outlook it is to investigate if a feedback loop can be applied, where thermal distribution is influencing the power dissipation. This only makes sense if the power breakdown shows a major portion of leakage power, because it is an exponential function of temperature. Further, a verification method of the estimation results is needed. For this task crosschecking against different simulators and estimators or verification by lower level simulations come to mind.

References

[CRWG08] Coskun, Ayse Kivilcim, Tajana Simunic Rosing, Keith A. Whisnant, and Kenny C. Gross: *Temperature-aware MPSoC scheduling for reducing hot spots and gradients*. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, ASP-DAC, 2008.

[DRW98] Dick, R.P., D.L. Rhodes, and W. Wolf: *TGFF: task graphs for free*. In *Proceedings of the Sixth International Workshop on Hardware/Software Codesign. (CODES/CASHE)*, March 1998.

[EP04] Eisley, Noel and Li Shiuan Peh: *High-level power analysis for on-chip networks*. In *Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*, CASES, 2004.

[KLPS12] Kahng, Andrew B., Bin Li, Li Shiuan Peh, and Kambiz Samadi: *Orion 2.0: A power-area simulator for interconnection networks*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20, 2012.

[KSS11] Kaushik, Samarth, Amit Kumar Singh, and Thambipillai Srikanthan: *Preprocessing-based run-time mapping of applications on NoC-based MPSoCs*. IEEE Computer Society Annual Symposium on VLSI, 2011.

[KyCBB06] Kursun, Eren, Chen yong Cher, Alper Buyuktosunoglu, and Pradip Bose: *Investigating the effects of task scheduling on thermal behavior*. In *Third Workshop on Temperature-Aware Computer Systems (TACS)*, 2006.

[LTM$^+$05] Lu, Zhonghai, Rikard Thid, Mikael Millberg, Erland Nilsson, and Axel Jantsch: *NNSE: nostrum network-on-chip simulation environment*. In *Proceedings of Swedish System-on-Chip Conference, SSoC*, 2005.

[NIR] *NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling*. http://www.nirgam.ecs.soton.ac.uk/.

[NOX] *NOXIM*. http://sourceforge.net/projects/noxim/.

[VHR$^+$08] Vangal, S.R., J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar: *An 80-tile sub-100-w teraflops processor in 65-nm cmos*. IEEE Journal of Solid-State Circuits, 2008.

[WCG$^+$10] Wegner, Tim, Claas Cornelius, Martin Gag, Andreas Tockhorn, and Adelinde Uhrmacher: *Simulation of thermal behavior for networks-on-chip*. In *28th NORCHIP Conference*. NORCHIP, 2010.