Reduction of Thermal Imbalances and Hot Spots in Networks-on-Chip Using Proactive Temperature Management

Tim Wegner, Martin Gag, Dirk Timmermann, Adelinde Uhrmacher
Institute of Applied Microelectronics and Computer Engineering, University of Rostock,
Richard-Wagner-Str. 31, 18119 Rostock-Warnemünde, Germany
tim.wegner@uni-rostock.de

Abstract

With the progress of deep submicron technology power consumption and temperature related issues have become two of the most critical aspects for chip design. Therefore, very large-scale integrated systems like Systems-on-Chip (SoCs) are exposed to an ever increasing thermal stress. This necessitates effective mechanisms for thermal management. In this paper we propose to precompute and proactively manage on-chip temperature of systems based on Networks-on-Chip (NoCs). Thereby, traditional reactive approaches, utilizing the NoC interconnect infrastructure to perform thermal management, can be replaced. For this purpose, an existing simulation environment for NoC-based systems is enhanced with mechanisms to predict and proactively manage on-chip temperature distribution. Simulations show that proactive thermal management achieves improvements of more than 95 % and 80 % regarding reduction of temperature imbalances as well as peak temperatures inside a 4×4 NoC compared to identical reactive approaches.

1 Introduction

The emergence of nanotechnology leads to increasing integration densities. Since this development is accompanied by increased power density and switching activity per unit area, especially complex and highly integrated systems like SoCs have to contend with well-known challenges, that are topical more than ever. An important aspect is that a huge amount of power is dissipated as heat, leading to high circuit temperatures and possibly strongly unbalanced on-chip temperature distributions. As a consequence, thermal stress and physical effects exponentially depending on temperature [1] threaten the integrity of Integrated Circuits (ICs) and have major influence on operability, lifetime and performance. A simplified but illustrative relationship between temperature and deterioration is given by the Arrhenius model [2], describing the influence of temperature on the velocity of chemical reactions. For this reason, monitoring and control of on-chip temperature distribution is an important task to secure system functionality and ensure high performance.

Traditionally, monitoring and control of on-chip temperature is performed by collecting temperature-related data by means of diodes integrated into the chip. In order to take actions this information has to be conveyed to a component responsible for thermal management. Regarding NoCs, commonly the NoC infrastructure is used for data transmission. At such a component temperature information is evaluated and a corresponding reaction is triggered. This can be referred to as reactive thermal management, since action is taken after a temperature-related event has been reported [3]. The main drawback of this approach is that the response time is extended due to transmission delay induced by the NoC. Therefore, prevention of temperature rises and inhibition of unbalanced temperature distributions is exacerbated. Hence, we propose to predict the on-chip temperature profile based on a model that will be realized as part of a thermal management unit (TMU) instead of using cost-intensive physical sensors. Such a TMU can be implemented in software running on a core of the SoC or it is an inherent part of a core implemented in hardware. By means of the made predictions, the TMU is able to immediately initiate execution of instructions for Dynamic Frequency Scaling (DFS) and thread relocation policies, which are modified to both reduce hot spots and balance the overall thermal profile. Thereby, response time is shortened and slow reactive countermeasures are replaced by proactive measures to prevent predicted peak temperatures or imbalances. Prerequisites are that predictions can be accomplished rather fast without inducing unreasonable calculation effort generating additional heat. Furthermore, we provide 3 different configurations of an equivalent reactive version for the purpose of comparison. To ascertain to which extent proactive management outperforms

reactive approaches, all configurations are compared to each other and to a setup without any thermal management. Comparisons focus on peak and average temperatures and uniformity of on-chip temperature distribution.
The remainder of this paper is organized as follows. In section 2 an overview over existing work regarding modeling of on-chip temperature and approaches for reactive and proactive management strategies is given. In section 3 the employed simulation environment including either proactive or reactive thermal management is introduced. In section 4 experiments for the implementations of proactive and reactive management are carried out and results are compared and discussed. Finally, in section 5 conclusions are drawn.

2 Related Work

Thermal management is indispensable for provision of system integrity and high performance. Since in this regard proactivity is desirable, many investigations have already been conducted in the field of modeling thermal behavior [3]–[6] by exploiting the equivalence of electrical and thermal energy flows [7]. In [3], equivalent electrical RC-circuits are used to model the thermal behavior of an entire chip. The possibility to trade off modeling accuracy against speed is provided by variability of modeling granularity. Temperature of the functional blocks is computed by using values for average power dissipation. In [4] Shang et al. modify this approach to simulate the thermal behavior of on-chip networks. For this purpose, the model of equivalent RC-circuits is extended by the integration of heat spreading angles. During simulation temperature estimation is split into the three stages of capturing network traffic, using these statistics to estimate power consumption and finally computing the temperature profile. [5], [6] propose to create SPICE netlists based on RC-circuits to model on-chip thermal properties with different levels of granularity.
Research that can be related to reactive management strategies for on-chip networks is available abundantly. A general concept of an event-based runtime monitoring service for NoCs is proposed in [8]. This approach uses event-based hardware probes. These probes are attached to the NoC routers and observe the behavior of the NoC components during system runtime. In [9] this concept is examined with focus on the method of integration into an existing NoC. In [10] Guang et al. propose a hierarchical agent framework to realize monitoring services on parallel SoC systems. The main focus is on the interaction between agents of different hierarchy levels to provide reconfigurability and fault tolerance. An approach specified to reactive monitoring and control of temperature in NoC-based systems is provided by [11], where temperature sensors monitor component temperatures of a NoC-based Multi-Processor SoC (MPSoC) and use the NoC infrastructure to report temperature to a central unit responsible for thermal management.
In terms of thermal management, we define proactivity as precalculating or predicting temperature at runtime and taking appropriate actions instead of monitoring temperature and reacting to changes within the thermal profile. Assuming this, investigations in this field are available only sparsely. In [12] Coskun et al. use autoregressive moving average (ARMA) modeling to predict temperatures of MPSoCs by regressing previous measurements from thermal sensors. Predictions are utilized by a scheduler for thread allocation to balance temperature distribution. In [13] a regression-based thermal model exploiting hardware performance counters is used to predict temperature of a single processor. Predictions are utilized by a hybrid dynamic thermal management policy reverting to reactive hardware (e.g. clock gating) and proactive software (e.g. scheduling) mechanisms. An approach using a thermal model based on equivalent RC-circuits in order to apply both reactive and proactive measures for thermal management is introduced in [4].
A major drawback of the presented thermal models is that they rely on offline profiling to extract values for power consumption. Mostly, this is done by using external tools like Wattch [14]. This dependency renders these approaches more suitable for application during design phase (e.g. thermal-aware placement and mapping) than for dynamic thermal management. Although the introduced strategies for reactive management are undoubtedly capable of reducing hot spots and balancing the thermal profile to a certain extent, they still are reactive in nature and suffer from comparatively long response times. Moreover, most approaches for proactive thermal management show deficits, when they are deployed for NoC-based systems. Either they are not tailored to application in NoC-based systems and therefore are not suited for management of NoC components (i.e. routers, links, interfaces) [12], [13], or they focus on measures only partly applicable for NoCs (e.g. software-based thread allocation) [12], [13]. In some cases they partially still rely on physical sensors [12] or external tools for profiling [4]. Hence, we extend a NoC simulation environment by a thermal model, that (1) is based on equivalent RC-circuits, (2) does not depend on any external tools and allows for simultaneous system simulation and thermal modeling, (3) does not rely on thermal sensors. We use this model in conjunction with conventional measures for thermal management combining mechanisms applicable for NoC components (DFS) and IP cores (DFS, task relocation) to reduce hot spots and improve balance of temperature distribution. This facilitates efficient and cost-saving proactive thermal management for NoC-based systems and considerably decreases response time of adjustment measures.

3 Simulation Environment

The simulation environment, developed for simulating and evaluating reactive and proactive strategies for thermal management, is a cycle accurate simulator for functional simulation of NoCs based on a standard 2D mesh topology, wormhole packet switching and XY routing. It provides a high degree of parameterization allowing for specification of NoC size, link width and simulation duration. The system components are represented by Intellectual Property Cores (IPCs), which are individually configurable concerning generation frequency and length of packets, destination address and other parameters. Furthermore, the sample period ΔT for capturing statistics may be set. Captured statistics are examined regarding average and maximum temperature in the NoC. Average temperature variation between adjacent components and absolute maximum of temperature variation are analyzed to account for local and global balance of the thermal profile. Since the simulator is based on the SystemC [19] and SystemC Transaction-Level Modeling (TLM) [20] libraries, the thermal model was developed by using the SystemC Analog Mixed Signal (AMS) library [21]. This allows for both accurate temperature modeling and preservation of system integrity (i.e. independence from external tools for power tracing). Moreover, the dualism of electrical and thermal energy flows can be exploited for temperature modeling. For this purpose, the NoC infrastructure is mapped on a regular grid of RC-tiles [6]. The general flow of parallel functional and thermal simulation is depicted in **Fig. 1**. First the NoC topology is set up ① by analyzing information concerning NoC geometry, overall duration of simulation, deployed strategy for thermal management (i.e. proactive or reactive) and a set of other parameters. Additionally, the IPCs, represented as sending and receiving components, are configured. Subsequently, the equivalent RC-network is established ② according to parameters for chip geometry, desired modeling accuracy and others. Then, the actual simulation of the NoC, its thermal behavior and the employed strategy for thermal management is executed ③ as follows. After expiration of a predefined sample period ΔT the simulation is stalled, NoC component activity statistics are passed to the thermal model for current calculation, activity counters are reset and temperature output of the thermal model is logged. After this, the simulation is continued. While the simulation is running, the temperature output of the thermal model is updated every clock cycle. The electrical current I, corresponding to heat flow, which is fed into the equivalent RC-network, is calculated by:

$$I = \left(\sum Trans_{0\to1} * E_{Trans}\right) / \Delta T + P_{Static} \quad (1)$$

$Trans_{0\to1}$ is the number of bit transitions from 0 to 1 captured for the corresponding NoC component, E_{Trans} is the energy such a transition consumes, ΔT is the sample period and P_{Static} is the value for static power consumption only relevant for active NoC components. For NoC links E_{Trans} is set to 11,62 fJ, assuming a wire length of at most 200 μm, random traffic patterns and a transition rate of 50 % [15]. For routers E_{Trans} is set to 1,5 pJ per transition due to energy consumption of 0,096 nJ, caused by a 64 bit wide flit crossing a router [16]. Since routers are active NoC elements, P_{Static} for input and output modules as well as FIFO buffers is considered, too [17]. The estimated value of 20 pJ for E_{Trans} of an IPC is unreferenced and only serves to reflect the proportion of IPC to router accounting for the variability of heat generation depending on IPC activity. P_{Static} for an IPC is estimated to be about 100 mW based upon power dissipation of an IBM PowerPC 405 [18] being suited for integration into a NoC.

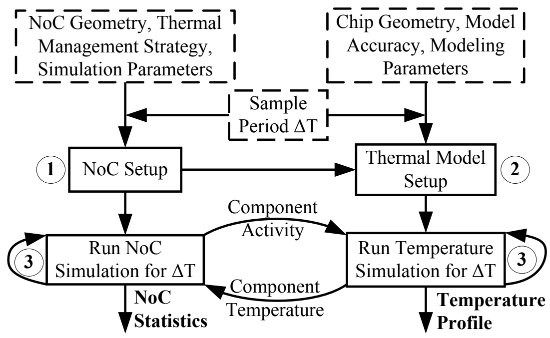


Fig. 1: Flow of simultaneous execution of NoC simulation and thermal modeling

3.1 Reactive Thermal Management

In order to determine to which extent the shortened response time of proactive thermal management influences the occurrence of hot spots and the balance of the thermal profile of NoC-based systems, it is necessary to compare this approach to the commonly applied reactive thermal management. Thus, we implemented a monitoring and control system for temperature, which reverts to event-based monitoring for NoCs [8]. According to this, a probe constantly monitoring temperature is attached to every NoC-tile. This probe is responsible for monitoring temperature of all components of its associated tile. In detail, this includes the IPC, the router and the 2 links from north to south as well as from east to west (see **Fig. 2**). In case an event (i.e. temperature change

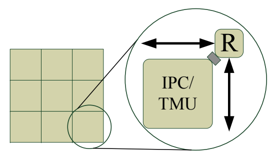


Fig. 2: Components of a NoC-tile a probe is responsible for

exceeding a predefined threshold) occurs for one or more components, the probe generates a packet containing the current temperature of the involved components. This packet is sent to the TMU via the NoC infrastructure. Reactivity of this approach and the associated extended response times are modeled by the fact that the TMU has to wait for temperature values provided by the probes in order to be able to update its own thermal profile of the NoC and to take action. This information is only available after a certain delay induced by transmission via the NoC. The overall scheme of the reactive TMU is illustrated in **Fig. 3**. All probe packets arriving consecutively are first stored in the FIFO and are processed in the sequence of their arrival. In parallel, the TMU periodically checks the FIFO for packets and stays in idle mode as long as the FIFO is empty. As soon as a packet is available, it is removed from the FIFO and analyzed regarding type (i.e. link, router, IPC), position and temperature value for the involved components. Thereupon, the TMU's internal thermal profile of the NoC is updated and an appropriate reaction is determined. Possible reactions are frequency adjustment (i.e. DFS) and task relocation between IPCs, in case an IPC is affected. Reaction policies follow previously defined values for step size of DFS as well as maximum and minimum frequency boundaries. Furthermore, an upper temperature limit and a lower limit for temperature variation between IPCs are defined both triggering IPC task relocation in a different way. The former serves to reduce hot spots by relocating a task to the IPC with the lowest temperature. The latter is used to balance the thermal profile by relocating a task to the IPC with the biggest temperature variation compared to the affected IPC. The TMU itself is not excluded from this process. Thus, every IPC is a potential TMU, since replacing a whole IPC by a TMU would induce unacceptable overhead. Hence, the TMU can be regarded as being implemented in software. While the TMU is in idle mode (i.e. no packets from probes available) or the TMU is currently not located in a particular IPC, this IPC switches to normal operation mode and regularly sends and receives data packets. After reaction is determined, a packet containing instructions for thermal management is generated and buffered in the output FIFO in order to be transmitted to the concerned NoC components. Since not only the strategy for thermal management but also its integration into the NoC impacts temperature distribution, 2 different approaches for probe integration into the NoC are investigated. In **Fig. 4(a)** the port of the router, normally connecting the IPC to the NoC, is shared among the probe and the IPC/TMU by using a Mux/Demux module. On the one hand, this approach is expected to minimize demand for additional logic necessary for integration and thus avoids immoderate growth of power consumption. On the other hand, probe and TMU packets are possibly slowed down due to lengthened communication paths or concurrent demand for communication resources. Integration of the probe by adding an extra port to the router, as depicted in **Fig. 4(b)**, does not introduce communication slowdown, but is expected to increase power consumption of the router more considerable and therefore additionally contributes to heat generation. This is due to additional FIFO buffers, input and output modules as well an increased complexity of the router crossbar.

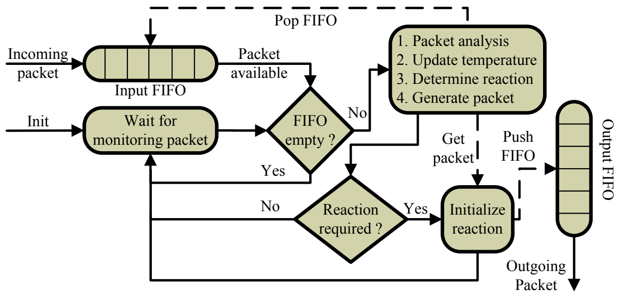


Fig. 3: Thermal Management Unit (TMU) for reactive management strategy

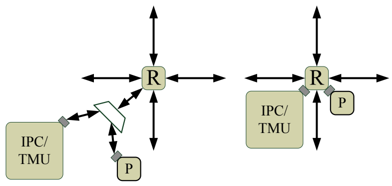


Fig. 4: Integration of a probe into the NoC: (a) using the IPC router port, (b) using an extra router port

3.2 Proactive Thermal Management

Proactive thermal management does not require cost-intensive probes for temperature monitoring. This minimizes costs regarding hardware resources and energy consumption, eliminates measuring inaccuracies induced by temperature sensors [3] and above all accelerates thermal management by reducing response times. Furthermore, packet transmission from probes to TMU is excluded as an error source (e.g. packet loss or data corruption) and a set of momentous design decisions (e.g. number and placement of probes) become dispensable. Instead of relying on sensors, the thermal profile of the NoC is directly modeled by the TMU avoiding additional delay, that would be induced by transmission via the NoC. The main challenge to enable a TMU to model the thermal profile of a NoC, is to provide the TMU with activity statistics of all network components. For this work it is assumed that this task can be accomplished by using system software running on the IPCs, since thermal management might be realized in software, too. The scheme of the proactive TMU is shown in **Fig. 5**. The parts of the figure colored in gray belong to a decentralized variation, in which

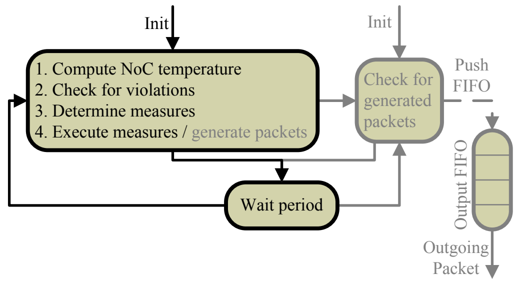


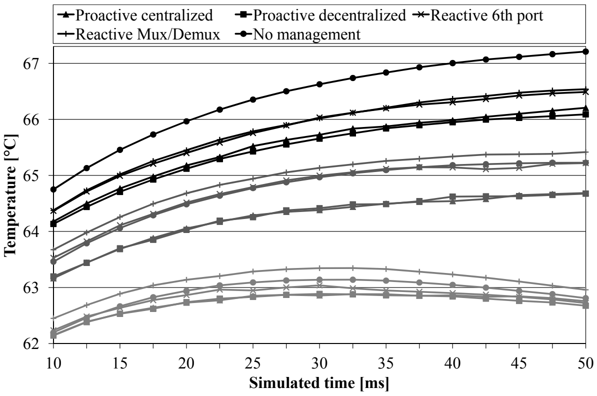
Fig. 5: TMU for proactive management strategy (gray colored parts belong to decentralized approach)

execution of management measures is shifted from the TMU to the affected NoC components. Generally, the TMU periodically updates its internal thermal model of the NoC by computing temperature of all components and checking for possible temperature violations. This corresponds to event-based temperature monitoring executed by the probes of the reactive approach using identical thresholds. In case violations are detected, according measures are determined. Admittedly, this process contributes to heat generation due to additional core activity. The measures follow the same policies described for reactive management, again including the TMU itself for possible relocation and therefore turning every IPC into a potential TMU. For a pure centralized solution scheduled measures are directly executed by the TMU avoiding delay induced by transmission of instruction packets to the concerned components. For decentralized management instruction packets are generated and buffered in the output FIFO until they are transmitted. This permits to assess to which extent thermal conditions benefit from centralized proactive thermal management, because that way a possible feedback loop of additional traffic in the NoC, leading to heat generation, triggering further adjustment measures and increasing traffic in turn, can be avoided. Since the centralized proactive TMU does not produce any packets, normal operation (i.e. sending and receiving of data packets) is constantly performed in parallel. This also applies to the decentralized proactive TMU as long as no instruction packets are available. The aspect of integration into the NoC can be omitted due to the absence of additional components leading to reduction of power consumption, complexity and transmission slowdown. Application of proactive thermal management additionally implies two possible advantages in comparison to reactive approaches, provided that temperature can be influenced positively. Either, thermal stress and peak temperatures are reduced, when adjustment measures of reactive management (i.e. DFS and task relocation) are exactly adopted leading to increased reliability, lifetime and performance. Or, to achieve identical thermal profiles, for the proactive approach less effort regarding adjustment measures has to be put in resulting in lower detraction of overall system performance. Since in this work adjustment policies for proactive management are completely adopted from reactive management, we expect the former case to occur.

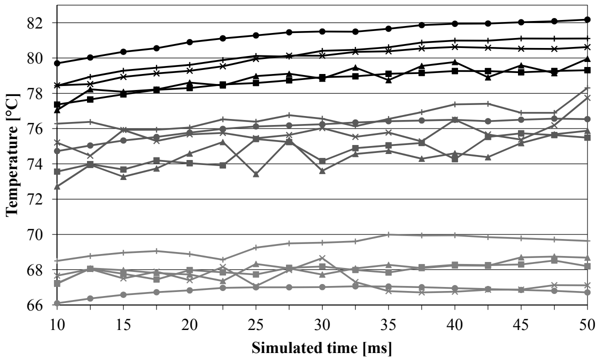
4 Experiments and Results

In this section the impact of proactive thermal management on the characteristics of thermal hot spots as well as stability of temperature distribution is investigated with reference to reactive management. For this purpose, simulations for NoC sizes of 2×2 up to 4×4 are executed to analyze to which extent both approaches impact heat generation and temperature distribution. Simulations are executed for reactive thermal management using the integration methods introduced in section 3.1, decentralized and centralized proactive management introduced in section 3.2 and a corresponding implementation without any thermal management. Since currently simulation of practical periods of time (e.g. a couple of minutes) turns out to be very time consuming, a single run is restricted to 50 ms. In return, to allow for illuminative analysis, all currents injected into the equivalent RC-network are amplified in order to accelerate the occurrence of noteworthy temperature variations. The initial temperature of the NoC components and ambient temperature are set to 60 °C [4] and 45 °C [3]. Threshold for detection of temperature changes is set to 0,2 °C resulting in a sufficient number of temperature-related events. In case DFS is scheduled, adjustment is allowed within the range of 1 GHz down to 500 MHz using step sizes of 10 %. Task relocation is triggered either by temperature exceeding 60,5 °C or by detection of temperature differences of at least 0,2 °C. Although these settings are not very common, they allow for testing thermal management at maximum capacity by frequent creation of alarming temperature profiles. Expectations are that temperature distribution will be considerably more balanced and peak temperatures can be reduced by means of proactive management. Furthermore, the NoC infrastructure should be relieved, since temperature-related information no longer has to be transported through the NoC resulting in lower power consumption and thus less heat generation. Especially for growing NoC sizes deviations should become apparent.

In **Fig. 6(a)** the average temperature of all on-chip components (i.e. routers, links and IPCs) over a simulation period of 50 ms is illustrated for NoC sizes of 2×2, 3×3 and 4×4. First, it can be seen that average temperature rises with increasing NoC sizes. This is due to the fact that an increased number of components generally results in a higher amount of generated heat and increased effort necessary for thermal management. Furthermore, it turns out that with increasing NoC sizes deviations between proactive and reactive thermal management become increasingly



(a) Average temperature of NoC components for reactive and proactive thermal management strategies as well as no management



(b) Peak temperature in a NoC for reactive and proactive thermal management strategies as well as no management

Fig. 6: Upper set of curves: 4×4 NoC; middle set of curves: 3×3 NoC; lower set of curves: 2×2 NoC (simulation period: 50 ms; data width: 32 bit)

TABLE 1: Improvements for avg and peak temperature, avg temperature deviation between adjacent components and max temperature difference compared to the reference system (abs value/%)

	2×2 NoC				3×3 NoC				4×4 NoC			
	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}
Proactive centralized	< 0,1/ 0,1	− 1,2/ − 1,8	− 0,24/ − 17	− 1,3/ − 26	0,4/0,7	1,5/2	0,24/7	1,2/9	0,8/1	2,3/3	0,52/10	2,2/12
Proactive decentralized	0,2/0,3	− 1,15/ − 1,7	− 0,23/ − 16	− 1,2/ − 25	0,4/0,7	1,2/1,6	0,21/6	1/7,6	0,9/1,3	2,5/3,1	0,55/ 10,5	2,3/13
Reactive Mux/Demux	− 0,2/ − 0,3	− 2,5/ − 4	− 0,54/ − 37	− 2,4/ − 50	− 0,2/ − 0,3	− 0,8/ − 1,0	− 0,37/ − 11	− 0,7/ − 5,5	0,5/0,8	1,1/1	0,25/5	0,9/5
Reactive extra port	< 0,1/ 0,1	− 0,7/ − 1	− 0,17/ − 12	− 0,8/ − 16	0/0	0,2/0,3	− 0,12/ − 3	0,2/2	0,6/1	1,4/2	0,36/7	1,2/6,5

TABLE 2: Improvements for avg and peak temperature, avg temperature deviation between adjacent components and max temperature difference compared to reactive management (extra port) (abs value/%)

	2×2 NoC				3×3 NoC				4×4 NoC			
	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}	T_{Avg}	T_{Max}	ΔT_{Adj}	ΔT_{Max}
Proactive centralized	< 0,1/4	− 0,5/ − 72	< − 0,1/ − 40,5	− 0,5/ − 66	0,47/ 2860	1,3/526	0,4/302	1/454	0,2/43	0,9/68	0,15/43	1/85
Proactive decentralized	< 0,1/ 146	− 0,4/ − 62	< − 0,1/ − 33	− 0,46/ − 60	0,46/ 2800	1/422	0,3/277	0,8/349	0,3/54	1,1/81	0,2/52	1,1/96

apparent, since the advantages of proactivity take effect. Improvements of all implemented approaches compared to the reference system are depicted in **Table 1**. As it can be seen, reduction of the average on-chip temperature is only marginal (or even negative) for the simulated NoC sizes. However, proactive approaches edge out reactive approaches due to shortened response times and reduced traffic. These improvements increase for growing NoC sizes with proactive decentralized management showing the best result of 1,3 % for a 4×4 NoC. Reactive management using a Mux/Demux unit continuously performs worst, since thermal management is not able to compensate additional heat generation induced by the Mux/Demux unit as well as monitoring and instruction flits crossing the NoC. Improvements induced by the proactive approaches in relation to reactive management providing the best results are illustrated in **Table 2**. Simulations show that both approaches are able to achieve only minor advancements of at most half a degree Celsius. Although the presented results do not reveal significant benefits for proactive thermal management compared to reactive strategies, it is assumed that proactive management will extend its lead compared to reactive management with increasing NoC sizes and run times. Moreover, these results refer to average values for the whole simulation time. As it can be seen in Fig. 6(a) peak deviations are more distinctive.

Fig. 6(b) depicts the maximum temperature measured among all on-chip components during simulation for NoC sizes of 2×2 up to 4×4. Basically, results resemble findings for average temperature. However, for the 2×2 NoC all approaches induce a raise of maximum temperature of at least 1 % (or 0,7 °C) compared to the reference system (see Table 1). Potential causes are the applied policy parameters for thermal management carrying adjustment to excess and thus acting counterproductive. In contrast, for a 3×3 NoC all approaches, except reactive management (Mux/Demux), achieve positive results. As expected, proactive thermal management outperforms the reactive approach using an extra port. The most distinct deviations between proactive and reactive management arise from simulation of a 4×4 NoC with proactive decentralized management reducing peak temperature by 3,1 % (corresponds to 2,5 °C), while reactive management using an extra port only yields a reduction of approximately 2 %. Comparisons between proactive management and the reactive approach performing best (see Table 2) show that, except for a 2×2 NoC, proactive management yields improvements of up to 1,3 °C for a 3×3 NoC, for example. When considering the results at hand, it stands to reason that larger NoCs (e.g. 10×10) will benefit from proactive thermal management, since the introduced proactive approaches apparently only tap their full potential with increasing NoC sizes. In this case as well, results represent average values for the overall simulation. For particular cases, temperature reduction induced by proactive management is more considerable (see Fig. 6(b)).

In **Fig. 7** the average deviation of temperature between adjacent components, illustrating the degree of local temperature balance, is shown for all approaches in relation to the reference system. For a 2×2 NoC both reactive and proactive management negatively affect local temperature balancing with the reactive approach using a Mux/Demux unit repeatedly performing worst (see Table 1). For a 3×3 NoC proactive approaches are able to increase thermal balance by slightly reducing deviations while reactive approaches still impair balance. Results for a 4×4 NoC reveal that all approaches are capable of positively influencing local temperature balance. Proactive management achieves the best results by reducing local imbalances by 10,5 % (0,55 °C) compared to the reference system. Comparisons of centralized as well as decentralized proactive management with reactive management using an extra port (see Table 2) exhibit slight improvements (up to 0,4 °C for a 3×3 NoC), which are restricted to NoC sizes larger than 2×2. Summing up, all approaches only provide practical improvements with growing NoC sizes, whereupon proactive management outperforms reactive approaches.

Results for maximum temperature difference between all on-chip components (i.e. global temperature balance) are depicted in **Fig. 8**. As it can be seen, for the 2×2 NoC none of the approaches is capable of improving global balance. This conforms to previous results. In this case however, proactive approaches perform even worse than reactive management using an extra port (see Table 1). For larger NoCs of 3×3 and 4×4 this relation is reversed. Reduction of global temperature difference using proactive management amounts to at most 13 % (2,3 °C) when compared to

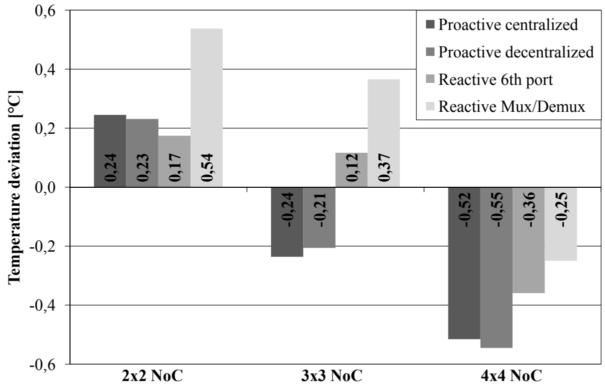


Fig. 7: Average deviation of temperature between adjacent on-chip components over a simulation period of 50 ms in relation to the reference system

the reference system, while advancements induced by reactive approaches are limited to 6,5 % (1,2 °C). Compared to reactive management using an extra port, maximum advancements of proactive management amount to 1,1 °C or 96 % for a 4×4 NoC, while for a 2×2 NoC reactive management performs better by up to 0,5 °C or 66 % (see Table 2). Conclusions are that for growing NoC sizes all approaches are able to increasingly improve global temperature balance. Furthermore, larger NoCs benefit most from proactive approaches, which can be mainly attributed to reduced response times.

The experiments conducted in this section show that neither proactive nor reactive thermal management are applicable for small NoCs, since average and peak temperatures as well as local and global temperature balances cannot be influenced positively to a worthwhile extent. In some cases thermal management even has a negative effect. Potential causes are that the applied policy parameters for thermal management carry adjustment to excess and that options for adjustment measures are not adequate for application to such small networks. Nevertheless, with growing NoC sizes both reactive and proactive management increasingly reveal their capability of reducing hot spots and contributing to balancing on-chip temperature distribution. Results show that proactive management generally outperforms reactive approaches and especially is preferable regarding reduction of temperature imbalances. First of all, this can be attributed to calculating on-chip temperature distribution at runtime, instead of relying on slow physical sensors. This provides proactive management with a time advantage due to reduced response times. Furthermore, absence of probe-generated packets and reduction of the number of control packets result in less additional activity and heat generation. Moreover, integration of reactive management necessitates additional hardware components, which consume power and generate heat on their own. Against expectations, integration

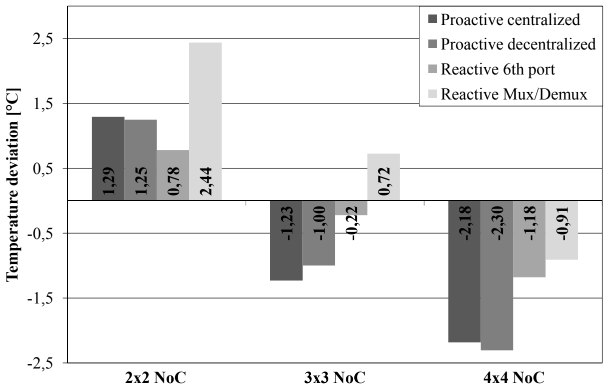


Fig. 8: Peak value of temperature difference between all on-chip components over a simulation period of 50 ms in relation to the reference system

of reactive management using a Mux/Demux unit requires more effort (i.e. combinational logic and buffers) than the approach using an extra port. Thus, exaltation of power consumption for this approach is higher yielding worse results. Hence, it stands to reason that proactive approaches are preferable, since thermal management can be accomplished more economical and more effective at the same time. Unexpectedly, results for centralized and decentralized proactive management differ only slightly, although decentralized management reverts to transmission of instructions via packets. In some cases, decentralized management even performs better. This allows for the assumption that adjustment is executed too frequently, since from a temperature-oriented point of view a certain extent of transmission delay has a promotive impact. Hence, in order to curtail frequency, modification of the adjustment policies is necessary. Moreover, the comparatively small improvements indicate that the thermal model and the adjustment measures, employed by proactive thermal management, require further calibration. For this purpose, a limited number of thermal sensors may be used in order to calibrate the according parameters and, if necessary, to correct the computed temperatures.

5 Conclusions

In this paper we propose an approach for proactive thermal management of NoCs. For this purpose, the NoC infrastructure is mapped on a network of RC-tiles exploiting the dualism of electrical and thermal energy flows. This model is used to simulate the prediction and proactive management of on-chip temperature distribution executed by a central Thermal Management Unit (TMU). Thereby, comparatively slow and possibly imprecise physical temperature probes inducing additional hardware and energy costs become dispensable. Additionally, design decisions regarding number and placement of probes for example, have not to be taken anymore. Comparisons between proactive thermal management and an equivalent reactive implementation show that the former reveals improvements of up to 54 % and 81 % concerning the reduction of average on-chip temperature and peak temperatures. Regarding local and global temperature balance, improvements of up to 52 % and 96 % are achieved. Although absolute values for improvement do not exceed 1,1 °C, results allow for the assumption that in the long term especially large NoCs will benefit most from application of proactive thermal management. Future work regarding proactive thermal management will focus on investigations concerning implementability and impact on NoC performance parameters like bandwidth, latency and network load.

References

[1] Failure Mechanisms and Models for Semiconductor Devices, JEDEC publication JEP122F, March 2009
[2] Srinivasan, J., et al.: "RAMP: A Model for Reliability Aware Microprocessor Design", IBM Research Report, RC23048, 2003
[3] Skadron, K., et al.: "Temperature-Aware Microarchitecture, in Proc. of ISCA 2003
[4] Shang, L., et al.: "Thermal Modeling, Characterization and Management of On-Chip Networks", in Proc. of MICRO 2004
[5] Liu, W., et al.: "On-Chip Thermal Modeling Based on SPICE Simulation", in Proc. of PATMOS 2009
[6] Tockhorn, A., et al.: "Modeling Temperature Distribution in Networks-on-Chip using RC-Circuits", in Proc. of DDECS 2010
[7] Krum, A.,: "Thermal Management", in Keith, F., editor, The CRC handbook of thermal engineering, CRC Press, Boca Raton, FL, 2000
[8] Ciordas, C., et al.: "An Event-based Monitoring Service for Networks on Chip", in ACM TOADES 2005, vol. 10, no. 4, pp. 702-723
[9] Ciordas, C., et al.: "NoC Monitoring: Impact on the Design Flow", in Proc. of IEEE ISCAS 2006
[10] Guang, L., et al.: "Hierarchical Agent Monitoring Design Approach towards Self-Aware Parallel Systems-on-Chip", in ACM TECS 2010, vol. 9, no. 3
[11] Atienza, D., Martinez, E.: "Inducing Thermal-Awareness in Multicore Systems Using Networks-on-Chip", in Proc. of IEEE Computer Society Annual Symposium on VLSI 2009
[12] Coskun, A. K., et al.: "Proactive Temperature Balancing for Low Cost Thermal Management in MPSoCs", in Proc. of ICCAD 2008, pp. 250-257
[13] Kumar, A., et al.; "HybDTM: A Coordinated Hardware-Software Approach for Dynamic Thermal Management", in Proc. of DAC 2006, pp. 548-553
[14] Brooks, D., et al.: "Wattch: A framework for architectural-level power analysis and optimizations", in Proc. of ISCA 2000
[15] Gag, M., et al.: "System Level Power Estimation of System-on-Chip Interconnects in Consideration of Transition Activity and Crosstalk", in Proc. Of PATMOS 2010, pp. 21-30
[16] Ye, T.T., et al.: "Packetization and routing analysis of on-chip multiprocessor networks", Journal of Systems Architecture, vol. 50, issue 2-3, February 2004, pp. 81-104
[17] Cornelius, C.: "Design of complex integrated systems based on networks-on-chip - Trading off performance, power and reliability", Dissertation, 2011
[18] IBM PowerPC 405 CPU Core Product Overview, https://www-01.ibm.com/chips/techlib/techlib.nsf/products/PowerPC_405_Embedded_Cores
[19] OSCI Core SystemC Language 2.2, http://www.systemc.org/downloads/standards/
[20] OSCI SystemC Transaction-Level Modeling Library 2.0.1, http://www.systemc.org/downloads/standards/tlm20/
[21] OSCI SystemC Analog Mixed Signal Extensions 1.0, http://www.systemc.org/downloads/standards/ams10/