Impact of Proactive Temperature Management on Performance of Networks-on-Chip

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

Abstract— **With the progress of deep submicron technology power consumption and temperature related issues have become dominant factors for chip design. Therefore, very large-scale integrated systems like Systems-on-Chip (SoCs) are exposed to an increasing thermal stress. On the one hand, this necessitates effective mechanisms for thermal management. On the other hand, appliance of thermal management is accompanied by disturbance of system integrity and degradation of system performance. 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 infrastructure to perform thermal management, can be replaced. This results not only in shorter response times for appliance of management measures and therefore in a reduction of temperature and thermal imbalances, but also in less impairment of system integrity and performance. Simulations show that proactive management achieves improvements of nearly 150 % regarding reduction of average temperature inside a 3×3 NoC compared to identical reactive approaches, while mitigating additional delay for packet transmission by more than 50 %.**

I. Introduction

The emergence of nanotechnology is accompanied by cumulative power densities and switching activities per unit area. Therefore, increasingly complex and highly integrated systems like SoCs have to contend with well-known challenges. Amongst others, this concerns heat dissipation, leading to high circuit temperatures and possibly strongly unbalanced on-chip temperature distributions. In the light of a growing number of transistors per chip, which are increasingly susceptible to environmental influences and deterioration, this issue is topical more than ever. 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. The relationship between temperature and deterioration is illustrated 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 are important tasks to secure system functionality and ensure high performance.

Typically, monitoring of on-chip temperature is performed by collecting temperature-related data (e.g. by using integrated diodes). In order to react to undesirable temperatures this data has to be transfered to a component responsible for data evaluation and determination of appropriate reactions (i.e. thermal management). Then instructions are sent to the concerned components. For NoC-based systems, commonly the NoC infrastructure is used for this communication. Despite the importance of thermal management reactive approaches impairs system performance, since the utilization of the NoC presents an intrusion into the system and the induced traffic curtails the availability of the NoC for regular communication. Another drawback is the comparatively long response time of thermal management caused by transmission delay, when using the NoC. Since two transmissions (i.e. reporting temperature and sending instructions) are necessary, an already highly congested NoC additionally exacerbates thermal management. Hence, we propose to predict the on-chip temperature profile based on a model that is realized as part of a Thermal Management Unit (TMU), instead of reverting to physical sensors. By means of the made predictions, the TMU is able to immediately initiate execution of instructions for thermal management. 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. Thereby, response time for thermal management is shortened by avoiding transmission of temperature-related data to the TMU and the traffic load of the NoC is reduced freeing up communication capacities for regular data traffic. Prerequisites are that predictions can be accomplished rather fast without inducing unreasonable calculation effort generating additional heat. To ascertain to which extent proactive thermal management influences system performance and on-chip temperature distribution, this approach is compared to a setup reverting to reactive management and to a setup without any thermal management.

The remainder of this paper is organized as follows. In section II an overview over existing work regarding modeling of on-chip temperature and approaches for reactive and proactive management strategies is given. In section III the environment for the simulation of proactive and reactive thermal management of NoC-based systems is introduced. In section IV experiments focusing on the impact of proactive and reactive management on system performance and temperature are conducted. Finally, in section V conclusions are drawn.

II. Related Work

Numerous investigations have already been conducted in the field of modeling thermal behavior [3]–[6] of ICs by exploiting the equivalence of electrical and thermal energy flows [7], since this approach implicates some worthwhile consequences

978-1-4577-0672-1/11/$26.00

regarding effort for thermal management. In [3], electrical RC-circuits are used to model the thermal behavior of an entire chip. Variability of modeling granularity allows for a trade-off between modeling accuracy and speed. Temperature of the functional blocks is computed by using values for average power dissipation. In [4] this approach is tailored to the simulation of 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. Temperature estimation is performed by capturing the network traffic, using these statistics for estimation of power consumption and computing the temperature profile. The creation of SPICE netlists consisting of RC-circuits in order to model on-chip thermal properties is proposed in [5], [6].

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 NoC components using hardware probes is proposed in [8]. In [9] this concept is examined with focus on the integration into an existing NoC and the arising implications. In [10] Guang et al. propose a hierarchical agent framework to realize monitoring services on parallel SoC systems in order to provide for reconfigurability and fault tolerance. An approach specified to reactive monitoring and control of temperature in NoC-based systems is provided by [11], where sensors monitor the temperature of the system components and use the NoC infrastructure to report temperature to a central TMU.

Proactive thermal management can be defined as predicting temperature at runtime and taking appropriate actions instead of monitoring temperature and reacting to changes. Assuming this, investigations in this field are available more sparsely. In [12] autoregressive moving average (ARMA) modeling is used to predict temperature of SoCs by regressing previous measurements from thermal sensors. Predictions are employed for thread allocation in order to balance temperature distribution. An approach using a thermal model based on RC-circuits in order to apply reactive and proactive measures for thermal management is introduced in [4].

Our work is motivated by three issues. Firstly, thermal models eventually depend on offline profiling for the extraction of values for power consumption. This makes these models more suitable for tasks like thermal-aware placement and mapping than for the dynamic modeling of thermal properties of ICs. Secondly, due to their nature reactive strategies for thermal management suffer from long response times because the sending of instructions requires availability of monitoring data. The transmission of this data furthermore impairs system performance due to increased traffic. Thirdly, in many cases proactive approaches are not suited for management of NoC routers and links (e.g. they deploy measures like software-based thread allocation) or they partially still rely on physical sensors [12] and external tools for profiling [4]. For these reasons the main contributions of this work are as follows. We extend a NoC simulation environment by a thermal model, which is based on equivalent RC-circuits and therefore does not rely on thermal sensors, which does not depend on any external tools and which allows for simultaneous system simulation and thermal modeling. This model is used in conjunction with Dynamic Frequency Scaling (DFS) and task relocation in order to allow for simulation of proactive thermal management. To determine the impact on system performance and on-chip temperature distribution this setup is compared to an analog reactive implementation and a reference system without thermal management.

III. Simulation Environment

The simulation environment, developed for evaluation of reactive and proactive thermal management, allows for functional simulation of NoCs based on a 2D mesh topology, wormhole packet switching and XY routing. Amongst others, parameters like NoC size, link width and simulation duration can be specified. The system components, which are connected by the NoC, are represented by Intellectual Property Cores (IPCs). The IPCs are individually configurable concerning generation frequency, length and destination address of packets. The sample period T_S, determining the rate of capturing statistics, can be set, too. To preserve consistency the thermal model was developed by using the SystemC Analog Mixed Signal (AMS) library [20], since the simulation environment itself is based on the SystemC [18] and SystemC Transaction-Level Modeling (TLM) [19] libraries. By deploying the AMS library the dualism of electrical and thermal energy flows can be exploited for modeling, because models of all necessary electrical components are included. This allows for simultaneous system simulation and thermal modeling, while preserving system integrity (i.e. independence from external tools for power tracing). For modeling, 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 simulation parameters and the deployed strategy for thermal management and configuring the IPCs (represented as sending and receiving components). Subsequently, the equivalent RC-network is established ② according to the specified geometry and modeling parameters. Then, the simulation of the NoC, its thermal behavior and the employed strategy for thermal management is executed ③. Every time the specified sample period T_S expires, the simulation is stalled, NoC component activity statistics are passed to the thermal model for current calculation and temperature output of the thermal model is delivered to the thermal management system. After this, the simulation is continued. During simulation the output of the thermal model is updated every clock cycle. The electrical current I, corresponding to heat flow, which is fed into the RC-network, is calculated by (1).

$$I \mathrel{\widehat{=}} \left(\sum Trans_{0 \rightarrow 1} * E_{Trans} \right) / \Delta T + P_{Static} \qquad (1)$$

$Trans_{0 \rightarrow 1}$ is the number of bit transitions from 0 to 1 captured for a particular NoC component, E_{Trans} is the energy a single transition consumes, T_S is the sample period and P_{Static} is the value for static power consumption only relevant for active components. For routers E_{Trans} is set to 1,5 pJ due to energy

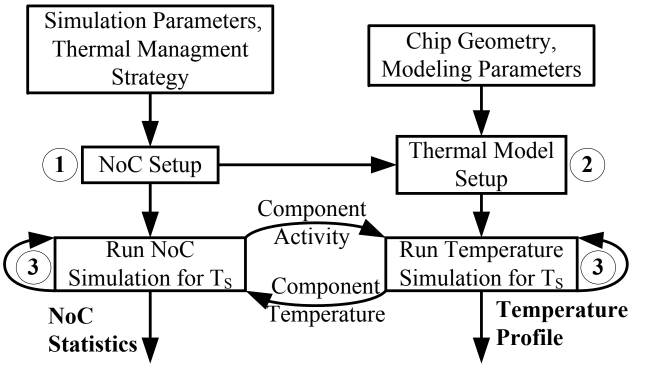


Fig. 1. Flow of simultaneous functional simulation and thermal modeling

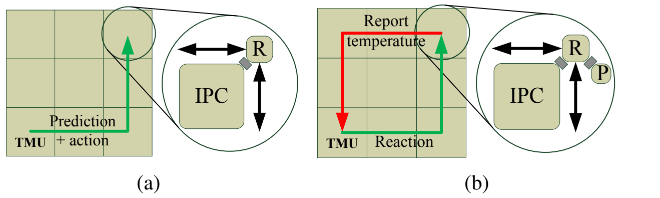


Fig. 2. Management flow and exemplary communication paths for thermal management in a 3×3 NoC: (a) proactive approach, (b) reactive approach (magnified area: components of a NoC-tile a probe is responsible for)

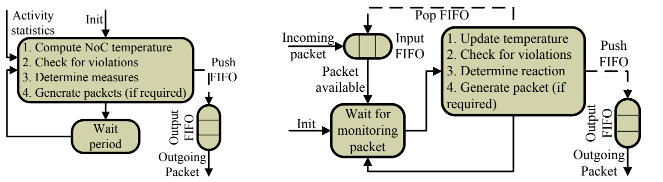
consumption of 0,096 nJ, caused by a 64 bit wide flit crossing a router [15]. Since routers are active elements, P_{Static} for input and output modules as well as FIFO buffers has to be considered [16]. The value of 20 pJ for E_{Trans} of an IPC is unreferenced and only serves to reflect the proportion 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 [17] being suitable for integration into a NoC. 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 % [14].

A. Reactive Thermal Management

Reactive thermal management is represented by an implementation of a monitoring and control system, which reverts to event-based monitoring for NoCs [8]. The general flow of reactive thermal management is depicted in Fig. 2 (b). A probe P is attached to every NoC-tile. This probe constantly monitors temperature of all components of its associated tile. This includes the IPC, the router and the 2 links from north to south as well as from east to west (see magnified area in Fig. 2 (b)). In case a temperature change exceeding a threshold T_{Thresh} is detected 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 where it is analyzed and, if necessary, appropriate instructions are sent back to the affected components. The overall scheme of the reactive TMU is illustrated in Fig. 3 (b). All arriving probe packets are stored in the input FIFO and are then processed in the sequence of their arrival. As long as the FIFO is empty the TMU stays in idle mode. In case a packet is available, it is removed from the FIFO and analyzed regarding type (i.e. link, router, IPC), position and temperature value of the involved components. Thereupon, the TMU's internal thermal profile of the NoC is updated and an appropriate reaction (i.e. DFS or task relocation) is determined. Then 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. Reaction policies follow specified values for step size of DFS as well as maximum and minimum frequency boundaries. Furthermore, an upper temperature limit T_{Bound} and a lower limit for temperature variation ΔT_{Max} between IPCs are defined both triggering IPC task relocation. T_{Bound} serves to reduce hot spots by relocating a task to the IPC with the lowest temperature. ΔT_{Max} 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 or it is currently not located in a particular IPC, this IPC switches to normal operation mode and regularly sends and receives data packets.

B. Proactive Thermal Management

Proactive thermal management does not require probes for temperature monitoring. This accelerates thermal management by reducing response times, excludes packet transmission from probes to the TMU as an error source (e.g. packet loss or data corruption) and redundantizes a set of momentous design decisions (e.g. number and placement of probes). The flow of proactive thermal management is illustrated in Fig. 2 (a). The stage of data transmission from a probe to the TMU is omitted, since the thermal profile of the NoC is directly modeled by the TMU. This avoids additional delay, that would be induced by transmission of monitoring data via the NoC. The detailed scheme of the proactive TMU is shown in Fig. 3 (a). Generally, the TMU periodically updates its internal thermal model of the NoC by analyzing NoC activity statistics, computing the temperature of all components and checking for temperature violations. This corresponds to event-based temperature monitoring executed by the probes of the reactive approach using an identical threshold T_{Thresh}. In case violations are detected, according measures are determined. 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. Once measures are scheduled, instruction packets are generated and buffered in the output FIFO until they are transmitted. Of course this process contributes to heat generation due to additional core activity. As long as no instruction packets are generated, the IPC, in which the TMU is currently located, performs normal operation (i.e. sending and receiving of data



(a) (b)

Fig. 3. Thermal Management Unit (TMU): (a) proactive approach, (b) reactive approach

packets). The main challenge to enable a TMU to model the thermal profile of a practical 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 without occupying the NoC by reverting to system software running on the IPCs, since thermal management might be realized in software, too. Another conceivable approach could be to exploit structures and mechanisms integrated for the purpose of testability (i.e. Built-In-Self-Test, Design for Test) [13]. Admittedly, in case activity statistics have to be transported to the TMU using the NoC, the performance advance of proactive management clearly diminishes. Besides reduced response times and lower traffic load, application of proactive thermal management additionally implies two possible advantages compared to reactive approaches, provided that temperature can be influenced positively. Either, thermal stress and peak temperatures are reduced, when applying identical adjustment measures, leading to increased reliability and lifetime. Or, to achieve identical results, for the proactive approach less effort regarding adjustment measures has to be put in, resulting in lower detraction of overall system performance.

IV. Experiments and Results

In this section the impact of proactive thermal management on temperature and system performance of NoC-based systems is investigated with reference to reactive management. Investigations focus on average temperature, uniformity of temperature distribution, net data throughput $Data_{Net}$, router delay D_R (i.e. time a flit needs to cross a router), delay of packet delivery D_P and the number of delivered packets P_{Trans}. All parameters related to performance refer to values for user data (i.e. traffic for thermal management is excluded). For this purpose, simulations for NoC sizes of 2×2, 3×3 and 4×4 are executed using configuration C1 (see Table I). Moreover, different configurations varying T_{Thresh}, T_{Bound} and ΔT_{Max} are applied to a 3×3 NoC. 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 chip temperature and the ambient temperature are set to 60 °C [4] and 45 °C [3]. For the purpose of comparison, results for the reference system without thermal management are depicted in Table II.

TABLE I

Configurations for simulation varying thresholds for detection of temperature changes T_{Thresh}, temperature limit of IPCs T_{Bound} and temperature variation between IPCs ΔT_{Max}

Config	T_{Thresh} $[°C]$	T_{Bound} $[°C]$	ΔT_{Max} $[°C]$
C1	0,2	60,5	0,2
C2	0,2	66,0	0,2
C3	0,2	66,0	2,0
C4	2,0	66,0	2,0

TABLE II

Router delay D_R, delay of packet delivery D_P, net data throughput $Data_{Net}$, the number of delivered packets P_{Trans}, average temperature T_{Avg} and temperature difference ΔT for the reference system (configuration: C1)

NoC size	D_R $[cycles]$	D_P $[cycles]$	$Data_{Net}$ $[bits/cycle]$	P_{Trans}	T_{Avg} $[°C]$	ΔT $[°C]$
2×2	5,6	45,9	17,9	≈ 1,64m	62,7	4,9
3×3	5,7	50,6	38,7	≈ 3,58m	64,4	13,1
4×4	6,8	60,1	70,8	≈ 6,55m	66	18

Table III shows the penalties for the performance parameters caused by reactive and proactive management for NoC sizes of 2×2 up to 4×4 using configuration C1. Additionally, improvements induced by the proactive approach are shown. Generally, both reactive and proactive management decrease overall performance for all NoC sizes with degradation growing with larger NoC sizes. For D_R reactive and proactive management cause impairments of at least 223 % (2×2 NoC) exceeding triplication of D_R compared to the reference system. Maximum improvement induced by proactive management is achieved for a 3×3 NoC (29 %). Basically, this also applies to D_P, since larger NoCs are entailed with increased traffic for management due to a higher number of components requiring management. The disproportional degradations for larger NoCs indicate that the NoC precociously gets congested due to additional management traffic. As it can be seen, proactive management, in contrast to reactive management, starts to cause heavy congestion only for larger NoCs, since no additional packets for monitoring data are generated. The observed advances clarify that proactive management is able to considerably relieve the NoC infrastructure. Regarding $Data_{Net}$ and P_{Trans} the proactive approach continuously performs worse than reactive management, although the NoC should have more capacities for regular traffic. This phenomenon can be explained by referring to the design of the TMU. As stated before, the TMU behaves like a regular IPC (i.e. receiving and sending data packets), as long as no management has to be performed. Apparently, this case occurs more often for reactive than for proactive management because in a congested NoC monitoring packets arrive comparatively infrequent. This leads to the reactive TMU being in normal operation mode for most of the time abetting $Data_{Net}$ and P_{Trans} but in turn contributing to congestion, while the proactive TMU primarily stays in management mode.

The above findings lead to the conclusion that for a more

TABLE III

Penalties for avg values of router delay D_R, delay of packet delivery D_P, net data throughput $Data_{Net}$ and the number of delivered packets P_{Trans} for different NoC sizes compared to the reference system and improvements of proactive management compared to its reactive counterpart (configuration: C1; management traffic excluded; abs value / %)

	$D_R[cycles]$			$D_P[cycles]$			$Data_{Net}[bits/cycle]$			P_{Trans}		
	2×2	**3×3**	**4×4**	**2×2**	**3×3**	**4×4**	**2×2**	**3×3**	**4×4**	**2×2**	**3×3**	**4×4**
Reactive	12,6 / 224	19,2 / 335	27,2 / 398	77,3 / 168	1626 / 3212	≈ 95k / ≈ 159k	1,8 / 10	2 / 5	9 / 13	≈ 140k / 8,6	164k / 4,6	≈ 803k / 12
Proactive	12,5 / 223	13,7 / 238	24,2 / 353	69,8 / 152	101 / 200	≈ 15k / ≈ 26k	2,4 / 13,3	5,9 /15	12 / 17	≈ 205k / 12,5	≈ 513k / 14	≈ 1,1m / 17
Improve-ment	< 0,1 / < 1	5,5 / 29	3 / 11	7,5 / 10	1525 / 94	≈ 80k / 84	– 0,6 / – 32	– 3,9 / – 194	– 3 / – 30	≈ – 65k / – 46	≈ – 349k / – 212	≈ – 320k / – 40

TABLE IV

Penalties for avg values of router delay D_R, delay of packet delivery D_P, net data throughput $Data_{Net}$ and the number of delivered packets P_{Trans} in a 3×3 NoC for different configurations compared to the reference system and improvements of proactive management compared to reactive management (management traffic excluded; abs value / %)

	$D_R[cycles]$				$D_P[cycles]$				$Data_{Net}[bits/cycle]$				P_{Trans}			
	C1	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**
Reactive	19,2 / 335	12,8 / 224	5,6 / 98	13,7 / 239	1626 / 3212	112 / 221	47 / 92	133 / 262	2 / 5	7,3 / 19	17,1 / 44	0,9 / 2	≈ 164k / 4,6	≈ 643k / 18	≈ 1,5m / 42,3	≈ 62k / 2
Proactive	13,7 / 238	6,6 / 115	6,1 / 107	10,8 / 188	101 / 200	52 / 103	50 / 99	77 / 152	5,9 / 15	14,9 / 38	16,7 / 43	3 / 8	≈ 513k / 14	≈ 1,35m / 38	≈ 1,5m / 42,3	≈ 260k / 7
Improve-ment	5,5 / 29	6,2 / 49	– 0,5 / – 10	– 2,9 / –21	1525 / 94	60 / 53	– 3 / – 7	56 / 42	– 3,9 / – 194	– 7,6 / – 104	0,4 / 2	– 2,1 / – 239	≈ – 349k / – 212	≈ – 707k / – 110	– 7 / < 0,1	≈ – 198k / – 321

effective management the parameters need to be modified in order to reduce impairment of system performance, while still sustaining positive impact on temperature distribution. For this purpose, different configurations varying T_{Thresh}, T_{Bound} and ΔT_{Max} (see Table I) are applied for proactive and reactive management of a 3×3 NoC. The impact of the different configurations on average temperature and peak temperature difference is illustrated in Fig. 4, while the influence on D_R, D_P, $Data_{Net}$ and P_{Trans} is depicted in Table IV. For configuration C2 T_{Bound} is raised to 66 °C reducing the number of instruction packets for relocating a task from an IPC violating this boundary to the IPC with the lowest temperature. This measure considerably decreases the negative impact on D_R and D_P for both reactive and proactive management, indicating a relaxation of traffic load, with proactive management still outperforming the reactive approach. However, $Data_{Net}$ and P_{Trans} are additionally impaired, since due to relaxed traffic conditions the remaining monitoring and instruction packets reach their destinations much faster. This means that response times are shortened and adjustment measures can be applied more promptly leading to the observed effect and noticeably reduced temperatures as it is depicted in Fig. 4. In detail, proactive management achieves improvements of 150 % and 134 % for average temperature and peak difference compared to reactive management, corresponding to absolute values of nearly 1 °C and 3 °C. For configuration C3 ΔT_{Max} is raised to 2,0 °C. Again, this reduces D_R and D_P for both approaches. The moderate results for proactive management denote that a saturation seems to be reached, because due to the absence of monitoring packets reduced traffic load only affects response times. In contrast, reactive management benefits much more from this measure, since the reduction of traffic load not only influences response times but also transmission times of monitoring packets. Thereby, the time the TMU is in normal operation mode is shortened more drastically for reactive management than for the proactive approach. As it can be seen in Table IV, this leads to a distinct impairment of $Data_{Net}$ and P_{Trans} for reactive management, while proactive management exhibits only slight degradation. These circumstances are also reflected in the results for average temperature and peak temperature differences of the NoC. While reactive management achieves noticeable reductions (see Fig. 4), improvements for proactive management are only marginal. For configuration C4 T_{Thresh} is raised to 2,0 °C. This results in fewer monitoring packets for reactive management and therefore also in fewer instruction packets congesting the NoC, while for proactive management naturally only the latter are reduced. On the one hand, relaxed traffic conditions lead to increased values for $Data_{Net}$ and P_{Trans} partly almost achieving the level of the reference design. This can be attributed to the fact that both TMUs are in normal operation more frequently, improving data throughput and overall number of regular data packets. On the other hand, the increased amount of regular data crossing the NoC leads to exacerbation of D_R and D_P, especially as the remaining management instructions (e.g. DFS) still decrease performance and the routers' operating frequency has a major influence on the delay. Due to a lower sensitivity to temperature changes both approaches achieve the worst results for average temperature and peak difference using configuration C4, partly even acting counterproductive (i.e. temperature rise compared to reference). For both approaches this can be attributed to reactions to temperature changes,

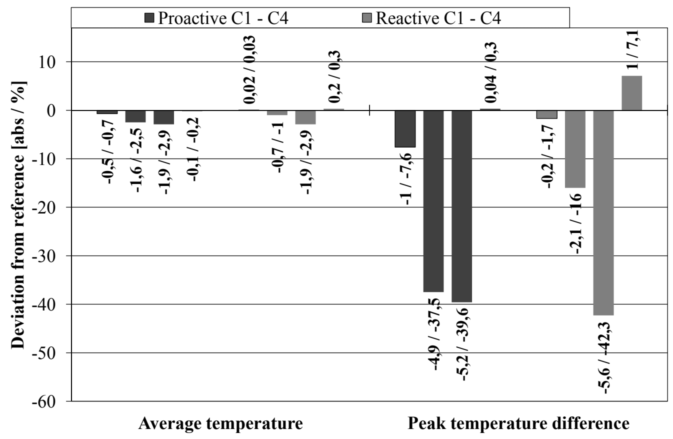


Fig. 4. Reduction of average temperature and peak temperature difference for different configurations normalized to the reference (abs values / %)

which apparently are scheduled too late and therefore apply instructions that are not suitable for the current situation.

The simulations conducted in this section show that parameters for thermal management, system performance, temperature distribution and NoC size are strongly correlated. Results clarify that management parameters have to be individually adapted to different NoC sizes in order to guarantee a certain level of system performance. This applies to both reactive and proactive management. Furthermore, it turns out that reactive management is applicable to large NoCs only to a limited extent, since the management traffic heavily congests the NoC and leads to unacceptable delays. Modification of management parameters facilitates adaptation of thermal management to the performance needs of the underlying system. Generally, results show that for both reactive and proactive management positive effects on temperature distribution can be traded off against performance. In case temperatures and a impairment of delay (D_R, D_P) are required to be as low as possible, parameters for task relocation need to be relaxed. In contrast, if high data throughput ($Data_{Net}$, P_{Trans}) is preferred, thresholds for the detection of temperature violations have to be increased. However, since proactive management exhibits shorter response times and dispenses with monitoring data provided by thermal sensors or probes, the impact of modifications can be predicted more accurately. Therefore, it stands to reason that proactive management reduces the number and the impact of side effects and interdependencies (e.g. negative of effect of raised T_{Thresh} on D_R and D_P), requires less effort for management (i.e. no monitoring data) and needs less fine adjustment of parameters.

V. Conclusions

In this paper a proactive approach for thermal management of NoCs is proposed. For this purpose, the NoC infrastructure is mapped on a network of RC-tiles. Thereby, the dualism of electrical and thermal energy flows can be exploited in order to model the thermal behavior of a NoC. The RC-model is used to simulate the proactive thermal management of NoC-based systems executed by a central Thermal Management Unit (TMU). The TMU uses the temperature model to predict the temperature distribution and triggers appropriate measures, instead of relying on temperature values, which are transmitted by thermal sensors using the NoC. This contributes to the reduction of response times for thermal management and to the decrease of additional traffic congesting the NoC. Comparisons between proactive thermal management and an equivalent reactive implementation show improvements of nearly 134 % and 150 % regarding reduction of temperature imbalances and average temperature inside a 3×3 NoC, while lowering additional routing latency as well as packet transmission delay by more than 48 % and 53 % at the same time. Nevertheless, results for different configurations of management parameters show that in order to achieve practical advancements for on-chip temperature distribution, performance decreases have to be accepted. Furthermore, these parameters have to be individually adapted to particular NoC sizes to sustain sufficient system performance while applying thermal management.

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] Tran, Xuan-Tu, et al.: "A DFT Architecture for Asynchronous Networks-on-Chip", in Proc. of ETS 2006

[14] 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

[15] 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

[16] Cornelius, C.: "Design of complex integrated systems based on networks-on-chip - Trading off performance, power and reliability", Dissertation, 2011

[17] IBM PowerPC 405 CPU Core Product Overview, https://www-01.ibm.com/chips/tech\-lib/tech\-lib.nsf/products/Power\-PC$_$405$_$Embedded$_$Cores

[18] OSCI Core SystemC Language 2.2, http://www.systemc.org/

[19] OSCI SystemC Transaction-Level Modeling Library 2.0.1

[20] OSCI SystemC Analog Mixed Signal Extensions 1.0