Worst-Case Temperature Analysis for Different Resource Availabilities: A Case Study

Lars Schor, Hoeseok Yang, Iuliana Bacivarov, and Lothar Thiele

Computer Engineering and Networks Laboratory,
ETH Zurich, 8092 Zurich, Switzerland
firstname.lastname@tik.ee.ethz.ch

Abstract. With three-dimensional chip integration, the heat dissipation per unit area increases rapidly and may result in high on-chip temperatures. Real-time constraints cannot be guaranteed anymore as exceeding a certain threshold temperature can immediately reduce the systems reliability and performance. Dynamic thermal management methods are promising methods to prevent the system from overheating. However, when designing modern real-time systems that make use of such thermal management techniques, the designer has to be aware of their effect on the maximum possible temperature of the system. This paper proposes an analytic framework to calculate the worst-case temperature of a system with general resource availabilities. The event and resource model is based on real-time and network calculus so that the method is able to handle a broad range of uncertainties in terms of task arrivals and available computational power. In various case studies, the proposed framework is applied to an advanced multimedia system to analyze the impact of dynamic frequency scaling and thermal-aware scheduling techniques on the worst-case temperature of an embedded real-time system.

Keywords: Real-Time Systems, Worst-Case Peak Temperature, Thermal Analysis, Thermal Management.

1 Introduction

Three-dimensional stacked multi-processor systems are a promising approach to keep pace with the ever-increasing demand of computational performance by reducing the interconnect delay and effective chip footprint. However, the increase in performance imposes a major rise in heat dissipation per unit area, which in turn threats the reliability and performance of a system [7].

As modern embedded processors support several operation frequencies and deep power down (DPD) states for power efficient design, dynamic thermal management (DTM) based on dynamic voltage/frequency scaling is considered as a promising technique to prevent the system from overheating. Consequently, thermal management has been a hot topic in research in recent years including thermal-constraint scheduling to maximize the performance [2, 3, 9] or peak temperature reduction to meet performance constraints [1]. Specifically, in [3],

J.L. Ayala et al. (Eds.): PATMOS 2011, LNCS 6951, pp. 288–297, 2011.

the workload is maximized under thermal constraints for discrete DVS speeds and in [9], a convex optimization technique for temperature-aware frequency assignment is proposed. Bansal et al. [1] explore peak temperature reduction by adopting the on-line algorithm for energy efficiency proposed by Yao et al. [16].

Unfortunately, none of the previous work has studied the effect of general resource availabilities on the worst-case temperature of a system under all possible scenarios of task executions, even thought this knowledge is desirable for any modern real-time system. In [10], a system-level analytic technique has been proposed, which calculates the worst-case temperature of a real-time system. However, as the work assumes work-conserving systems with full service availability, the method is not able to analyze DTM techniques. Recently, the method has already been applied to DTM techniques, in particular leaky bucket shapers [5].

Consequently, we extend the base technique proposed in [10] to general resource availability. This enables us to study the effect of dynamic frequency scaling (DFS) and TDMA scheduling on the worst-case temperature of the system. Our framework considers general event arrivals modeled by arrival curves from real-time and network calculus [6, 12]. An arrival curve constraints an upper bound on the workload that might arrive to the system in any time interval. Similarly, the computational power provided by a processor in any time interval is also constrained by a service curve set. The contributions of this paper can be summarized as follows:

- The considered system is formally described by corresponding thermal and computational models.
- The technique proposed in [10] is extended to general resource availabilities based on service curves from real-time and network calculus [6, 12].
- We implemented the proposed technique in MPA [14] and used it in various experiments to study the effect of DFS and TDMA on the worst-case temperature of a real-time system.

2 System Model

This section introduces the models to analyze a single computing component for its worst-case temperature. The model proposed in [10] is extended to consider general resource availabilities, such as frequency modulation.

2.1 Computational Model

The computational model of a component follows the idea of network and real-time calculus [6, 12]. Consequently, we suppose that in time interval $[s, t)$, events with a total workload of $R(s, t)$ time units arrive at processing component. The arrival curve α upper bounds all possible cumulative workloads:

$$R(s,t) \leq \alpha(t-s) \quad \forall s < t \tag{1}$$

with $\alpha(0) = 0$. Suppose that several independent workload functions R_k are individually bounded by arrival curves α_k. In case a component concurrently processes multiple workload functions, the accumulated workload can be bounded as $R(s,t) \leq \sum_{\forall k} \alpha_k(t-s) = \alpha(t-s)$.

A processing component is given $W(s,t)$ time units computing resources in time interval $[s,t)$. Events that have not yet been completed are queued in the component's ready queue while waiting for further resources. The resource availability W is upper and lower bounded using a service curve

$$\beta^l(t-s) \leq W(s,t) \leq \beta^u(t-s) \quad \forall s < t \tag{2}$$

with $\beta^l(0) = \beta^u(0) = 0$. The accumulated computing time $Q(s,t)$ describes the amount of time units that a component is spending to process an incoming workload of $R(s,t)$ time units. The accumulated computing time $Q(s,t)$ in time interval $[s,t)$ is

$$Q(s,t) = \inf_{s \leq u \leq t} \{W(s,t) - W(s,u) + R(s,u)\} \tag{3}$$

provided that there is no buffered workload in the ready queue at time s [12]. The upper bound on the accumulated computing time can be determined according to [13] as

$$Q(t-\Delta,t) \leq \gamma(\Delta) = \min\{(\alpha \otimes \beta^u) \oslash \beta^l, \beta^u\}, \tag{4}$$

where $(f \otimes g)(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{f(\Delta-\lambda)+g(\lambda)\}$ and $(f \oslash g)(\Delta) = \sup_{\lambda \geq 0}\{f(\Delta+\lambda)-g(\lambda)\}$.

Because of (3) and (4), the accumulated computing time $Q(s,t)$ for any fixed s and its upper bound $\gamma(t-s)$ are monotonically increasing. Consequently, the operating mode of a component can be expressed by the mode function

$$S(t) = \frac{\mathrm{d}Q(s,t)}{\mathrm{d}t} \in [0,1] \tag{5}$$

for any $s < t$. $S(t) = 1$ and $S(t) = 0$ denote that the processing component is fully utilized and idle, respectively.

2.2 Power Model

It is well known that the dynamic power consumption P_{dyn} growths quadratically with the supply voltage v and linearly with the operating frequency f:

$$P_{\mathrm{dyn}} \propto v^2 \cdot f. \tag{6}$$

For the sake of simplicity, the supply voltage is assumed to be given as fixed value, so that the model can be applied to DFS without loss of generality. In addition, we model the temperature dependency of leakage power by means of a linear approximation [8], i.e. $P_{\mathrm{leak}} = \phi \cdot T + \psi$. With fixed v, the operating frequency f is proportional to the mode l and the total power consumption can be derived as

$$P = \phi \cdot T + \rho \cdot l + \psi. \tag{7}$$

2.3 Thermal Model

A widely used duality to analyze the heat transfer of a modern VLSI system is to model the heat flow as current passing through a thermal resistance, to model the thermal difference as the corresponding voltage, and to model the heat capacity as an electrical capacity, see [4, 11, 15]. In particular, temperature will follow a first order linear differential equation of the form

$$C \cdot \frac{\mathrm{d}T}{\mathrm{d}t} = P - G \cdot (T - T_{\mathrm{amb}}) \tag{8}$$

where C, P, G and T_{amb} denote the thermal capacity, the generated power, the thermal conductance and the ambient temperature, respectively. Rewriting (8) with (7) leads us to

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -g \cdot T + h, \quad \text{with } g = \frac{G - \phi}{C}, \quad h = \frac{\rho \cdot l + \psi + G \cdot T_{\mathrm{amb}}}{C} \tag{9}$$

with time-dependent temperature T. A closed-form solution to the above yields

$$T(t) = T^{\infty} \cdot \left(1 - e^{-g \cdot (t - t_0)}\right) + T(t_0) \cdot e^{-g \cdot (t - t_0)} \tag{10}$$

where $T^{\infty} = h/g$ is the steady-state temperature in a single mode of slope l.

Throughout this paper, we restrict our analysis to *proper* thermal models, in which the following reasonable conditions are satisfied:

- We have $g > 0$, i.e. $G > \phi$.
- The steady-state temperature is not smaller in a single mode of slope l than in a single mode of slope $l = 0$, i.e. we have $T_l^{\infty} \geq T_{l=0}^{\infty}$ or $l \geq 0$.

Moreover, according to the thermal model, the component has the *thermal monotonicity* property.

Lemma 1. *(Monotonicity) Suppose we consider two equal timed sequences of operation modes in a time interval from s to t. Then, the sequence with higher temperature at time s leads to a higher component temperature at time t.*

Proof. This lemma comes from (10) by taking h as a time-dependent function. □

2.4 Problem Definition

Now, we can formulate the worst-case peak temperature analysis problem:

> Given is the computing model in (3), the power model in (7), the distributed thermal model in (9). Then, the objective is to determine the peak temperature T^* for any cumulative workload R that complies with a given sub-additive arrival curve α and any resource availability W within service curve β.

3 Thermal Analysis

In this section, we will construct a worst-case accumulated computing time $Q^*(0,t)$ of the processing component that leads to an upper bound T^*. The upper bound T^* can later be obtained by simulating the system with this accumulated computing time $Q^*(0,t)$. As a first prerequisite, we will show in Lemma 2 that allowing more power later in time (closer to τ) will always increase the temperature $T(\tau)$.

Lemma 2. *(Shifting) Given is a proper system model according to (7) and (9) as well as some time instance τ. In addition, we consider two mode functions $S(t)$ and $\overline{S}(t)$ defined for $t \in [0,\tau)$ which are only different in a time interval $[\sigma, \sigma+2\delta)$ with $\sigma+2\delta < \tau$. In particular, we have $S(t) = l_1$ for all $t \in [\sigma, \sigma+\delta)$, $S(t) = l_2$ for all $t \in [\sigma+\delta, \sigma+2\delta)$. $\overline{S}(t)$ is $\overline{l_1}$ and $\overline{l_2}$ for $[\sigma, \sigma+\delta)$ and $[\sigma+\delta, \sigma+2\delta)$ respectively, where $\overline{l_1} = l_1 - \Delta$, $\overline{l_2} = l_2 + \Delta$, and $0 \leq \Delta \leq l_1$. In other words, we allow more power at the second time interval of $[\sigma, \sigma+2\delta)$ while keeping the total power consumption as the same. Then, if $\overline{T}(0) = T(0)$, we have $\overline{T}(\tau) \geq T(\tau)$, i.e. mode function $\overline{S}(\tau)$ results in a higher temperature at time τ when δ is small enough.*

Proof. As the mode functions satisfy $S(t) = \overline{S}(t)$ for all $t \in [0,\sigma)$ and $\overline{T}(0) = T(0)$, we clearly have $\overline{T}(\sigma) = T(\sigma)$. As $S(t) = l_1$ for $t \in [\sigma, \sigma+\delta)$ and $S(t) = l_2$ for $t \in [\sigma+\delta, \sigma+2\delta)$, we find

$$T(\sigma+2\delta) = T^{\infty}_{l=l_2} \cdot \left(1 - e^{-g\delta}\right) + T^{\infty}_{l=l_1} \cdot \left(1 - e^{-g\delta}\right) \cdot e^{-g\delta} + T(\sigma) \cdot e^{-2g\delta} \quad (11)$$

Furthermore, as $\overline{S}(t) = \overline{l}_1$ for $t \in [\sigma, \sigma+\delta)$ and $\overline{S}(t) = \overline{l}_2$ for $t \in [\sigma+\delta, \sigma+2\delta)$, we find

$$\overline{T}(\sigma+2\delta) = T^{\infty}_{l=\overline{l}_2} \cdot \left(1 - e^{-g\delta}\right) + T^{\infty}_{l=\overline{l}_1} \cdot \left(1 - e^{-g\delta}\right) \cdot e^{-g\delta} + \overline{T}(\sigma) \cdot e^{-2g\delta} \quad (12)$$

Then, for a proper thermal model, we have

$$\overline{T}(\sigma+2\delta) - T(\sigma+2\delta) = \frac{\rho}{G-\phi} \cdot \Delta \cdot \left(1 - e^{-g\delta}\right)^2 \geq 0 \quad (13)$$

where we used the fact that $\Delta \geq 0$. With the thermal monotonicity and $S(t) = \overline{S}(t)$ for all $t \in [\sigma+2\delta, \tau)$, the condition $\overline{T}(\sigma+2\delta) \geq T(\sigma+2\delta)$ also implies that $\overline{T}(\tau) \geq T(\tau)$. □

The next Lemma shows that we obtain a higher temperature at some time τ if in *any* interval ending at τ the component has larger accumulated computing time. This Lemma provides the foundation for the main theorem.

Lemma 3. *(Mode Functions Comparison) Given is a proper thermal model as well as some time instance τ. In addition, we consider two accumulated computing time functions Q and $\overline{Q}$ which satisfy*

$$\overline{Q}(\tau - \Delta, \tau) \geq Q(\tau - \Delta, \tau) \quad (14)$$

for all $0 \leq \Delta \leq \tau$. Then, if $\overline{T}(0) = T(0)$ we have $\overline{T}(\tau) \geq T(\tau)$, i.e. mode function $\overline{S}(\tau)$ results in a higher temperature at time τ.

Proof. Because of space limitations, only a sketch of the proof is provided. First note that because of (5), the condition of the Lemma translates equivalently to

$$\int_{\tau-\Delta}^{\tau} \overline{S}(t)\,\mathrm{d}t \geq \int_{\tau-\Delta}^{\tau} S(t)\,\mathrm{d}t. \tag{15}$$

In other words, in case of $\overline{Q}$, the component in *any* interval ending at τ has larger accumulated computing time.

Now, we will stepwise transform $S(t)$ into $\overline{S}(t)$ and in each step, the temperature will not decrease because of Lemma 2. In order to simplify the proof technicalities, we suppose discrete time, i.e. $S(t)$ and $\overline{S}(t)$ may change values only at multiples of δ. In other words, $S(t)$ and $\overline{S}(t)$ are constant for $t \in [k\delta, (k+1)\delta)$ for all $k \geq 0$. Let us define $\tau = k_{\max}\delta$. We now execute the following algorithm:

1. Determine the smallest $1 \leq k_1 \leq k_{\max}$ such that $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$. If there is no such k_1, then $S(t) = \overline{S}(t)$ for all $0 \leq t \leq \tau$ and therefore, $\overline{T}(\tau) = T(\tau)$ and the algorithm stops.
2. Determine the smallest k_2 with $k_1 < k_2 \leq k_{\max}$ such that $S(\tau - k_2\delta) \neq 0$. In case such a k_2 does not exist, it is trivial that $\overline{T}(\tau) \geq T(\tau)$ and the algorithm stops. Otherwise,
   (a) set $\Delta = \min\{S(\tau - k_2\delta), \overline{S}(\tau - k_1\delta) - S(\tau - k_1\delta)\}$,
   (b) add Δ to $S(t)$ for $t \in [\tau - k_1\delta, \tau - (k_1 - 1)\delta)$,
   (c) and subtract Δ from $S(t)$ for $t \in [\tau - k_2\delta, \tau - (k_2 - 1)\delta)$.
3. If $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$, continue with step 2. Otherwise, go to step 1.

Now, one can simply prove that after each iteration, $T(\tau)$ does not decrease until it reaches $\overline{T}(\tau)$ and therefore, the initial $T(\tau)$ was not larger than $\overline{T}(\tau)$. □

Based on the above Lemma we will show the main result of this section. The following Theorem provides a constructive method to determine the worst-case accumulated computing time Q^* for any processing component whose computational power is bounded by a service curve set.

Theorem 4. *(Worst-case Computing Time) Given is a proper thermal model according to (7) and (9). The component has the computational model (3) with the operation modes and power defined in (5) and (7), respectively. Then the following holds:*

- *Suppose that the accumulated computing time function $Q^*(0, \Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ for all $0 \leq \Delta \leq \tau$ leads to temperature $T^*(\tau)$ at time τ. Then $T^*(\tau)$ is an upper bound on the highest temperature $T^*(\tau) \geq T(\tau)$ for all feasible workload traces (4) that are bounded by the arrival curve α and service curve β according to (2) and (3) respectively.*
- *If in addition $T(0) \leq T^{\infty}_{l=0}$ holds in general, where T^{∞}_{l} is a steady-state temperature of the operation mode $S(t) = l$, then for any feasible workload trace we find $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$.*

Proof. We apply Theorem 3 to the proof of Theorem 4 of [10]. □

Table 1. Parameters of the video conferencing application

	Video	Audio	Network
period	50 ms	30 ms	30 ms
jitter	[20, 90]ms	10 ms	10 ms
min. interarrival	1 ms	1 ms	1 ms
execution demand	6 ms	3 ms	2 ms
deadline	50 ms	30 ms	30 ms

Table 2. Thermal and power parameters of the considered embedded system architecture

G	C	ϕ	ρ	ψ
$0.3\,\frac{\mathrm{W}}{\mathrm{K}}$	$0.03\,\frac{\mathrm{J}}{\mathrm{K}}$	$0.1\,\frac{\mathrm{W}}{\mathrm{K}}$	14.0 W	−25.0 W

Note that the worst-case scenario for the temperature is often completely different from the conventional critical instance scenario that is used in real-time analysis in order to determine the worst-case timing behavior. For example, for periodic tasks with jitter, the worst-case peak temperature scenario is to first warm up the system with periodic arrivals and then heat up the system with burst arrivals and jitters.

4 Experimental Analysis

In this section, we analyze the effect of a reduced resource availability on the worst-case temperature and provide hints on how to design a system which is schedulable and meets temperature constraints at the same time.

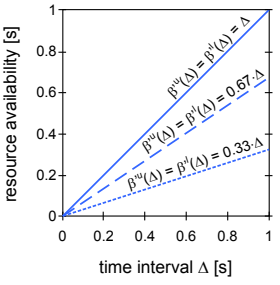
4.1 Experimental Setup

Two different example applications executing on a single-core embedded processor are considered in this section. The first example consists of a single process with a period of 200 ms and a execution demand of 50 ms. The second example is a multi-processing video-conferencing application, which includes a video code, an audio code and a network process for communication management. For illustration purpose, we use a period-jitter-delay model[1], with parameters summarized in Table 1. The system parameters are borrowed from [10] as summarized in Table 2 and in all our experiments, we start simulation with the initial temperature $T(0) = T^{\infty}_{l=0} = 325.00\,\mathrm{K}$, calculated from the parameters given in Table 2. The observation time τ is set to 1.0 s in all experiments.

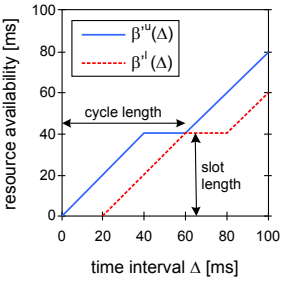
4.2 Dynamic Frequency Scaling

In the first experiment, we study the effect of frequency scaling on the peak temperature of the system. Figure 1(a) shows several services curves for different

[1] See [13] for the detail of all the parameters.

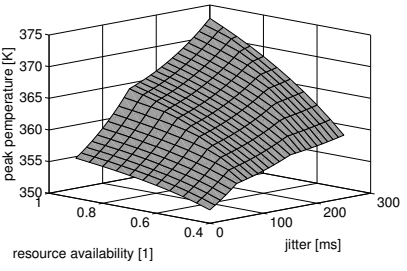


(a) Frequency modulated processor.

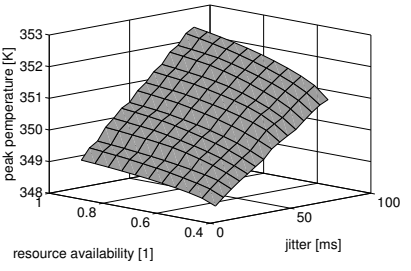


(b) TDMA.

Fig. 1. Service curves for various resource availabilities



(a) Single process system.

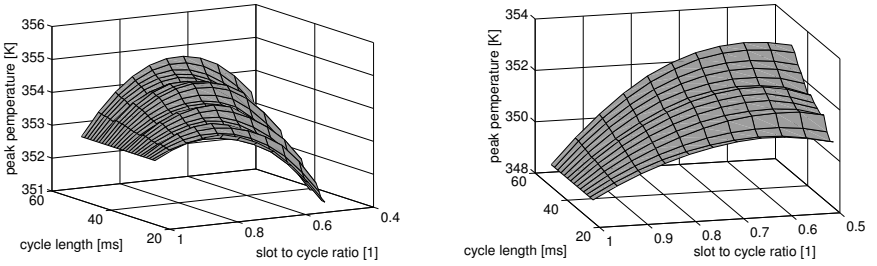


(b) Video-conferencing system.

Fig. 2. Worst-case temperature as a function of both resource availability and jitter

operation frequencies. The full service curve ($\beta(\Delta) = \Delta$) corresponds to full operation frequency while the two partially utilized service curves ($\beta(\Delta) = 0.67\Delta$ and $\beta(\Delta) = 0.33\Delta$) correspond to operation frequencies of 67 %, and 33 %, respectively.

Figures 2(a) and 2(b) outline the worst-case temperature as a function of resource availability and jitter for the single task application and the video conference application, respectively. For the single process application and a jitter of 50 ms, a reduction of the operation frequency by 50 % lowers the peak temperature by 4.23 K. Similarly, for a jitter of 300 ms, such a reduction of the operation frequency lowers the peak temperature by 14.50 K. As the power consumption and the inverse execution time of a process scale linearly with operation frequency, the steady-state temperature calculated by the average power consumption of an application is independent of the operation frequency. That is the reason why lowering the operation frequency has little influence on the peak temperature for relatively small jitters. As larger jitters cause the length of the burst to increase, the operation frequency has higher influence on the peak temperature for large jitters. During the burst, the component is continuously processing and the temperature increases towards the steady-state temperature T_l^∞ of slope l, which in turn depends on the operation frequency.



(a) Single process system. (b) Video-conferencing system.

Fig. 3. Worst-case temperature as a function of cycle length and slot to cycle ratio

4.3 Thermal-Aware Scheduling

A widely proposed method for temperature reduction is to place idle service intervals so that only a certain amount of time units are assigned to an application for computation. This service availability can be described by a TDMA schema, where the processor is always available for S consecutive time units during every cycle C. We call C the cycle length and S the slot length. Figure 1(b) outlines the service curves for a TDMA resource with $C = 60\,\text{ms}$ and $S = 40\,\text{ms}$.

Figures 3(a) and 3(b) outline the worst-case temperature as a function of cycle length and slot to cycle ratio. In both examples, the maximum jitter is 20 ms. Both figures show that longer idle intervals do not necessarily lead to a lower peak temperature, which is due to the buffering effect. In case the resource is not fully available, workload that arrives during idle intervals is buffered in the queue and cause the bursty shot later.

5 Conclusion

In this paper, we presented an analytical approach to determine the worst-case temperature for real-time systems with general resource availability. The accumulated workload arrival from all task invocations is characterized by an arrival curve, i.e. by an upper bound on the sum of task execution times arriving in any time interval. Similarly, we model the resource availability by service curves.

By applying the proposed method to various resource availability scenarios, we could show that frequency modulation is only a viable method for peak temperature reduction if the system has a large non-determinism. Analogously, placing idle service intervals does not always result in a lower peak temperature, as shown in the experiments. Consequently, our analysis methods can be used by system designer to choose proper resource parameters that reduce the peak temperature and guarantee real-time constraints.

Acknowledgments. This work was supported by EU FP7 projects EURETILE and PRO3D, under grant numbers 247846 and 249776.

References

[1] Bansal, N., Pruhs, K.: Speed Scaling to Manage Temperature. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 460–471. Springer, Heidelberg (2005)

[2] Chantem, T., Dick, R.P., Hu, X.S.: Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs. In: Proc. Design, Automation and Test in Europe (DATE), pp. 288–293 (2008)

[3] Chantem, T., Hu, X.S., Dick, R.P.: Online Work Maximization under a Peak Temperature Constraint. In: Proc. Int'l Symposium on Low Power Electronics and Design (ISLPED), pp. 105–110 (2009)

[4] Krum, A.: Thermal Management. In: Kreith, F. (ed.) The CRC Handbook of Thermal Engineering, pp. 1–92. CRC Press, Boca Raton (2000)

[5] Kumar, P., Thiele, L.: Cool Shapers: Shaping Real-Time Tasks for Improved Thermal Guarantees. In: Proc. of Design Automation Conference, DAC (2011)

[6] Thiran, P., Le Boudec, J.-Y.: Network Calculus. LNCS, vol. 2050. Springer, Heidelberg (2001)

[7] Li, F., Nicopoulos, C., Richardson, T., Xie, Y., Narayanan, V., Kandemir, M.: Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. In: Proc. Int'l Symposium on Computer Architecture (ISCA), pp. 130–141 (2006)

[8] Liu, Y., et al.: Accurate Temperature-Dependent Integrated Circuit Leakage Power Estimation is Easy. In: Proc. Design, Automation and Test in Europe, DATE (2007)

[9] Murali, S., Mutapcic, A., Atienza, D., Gupta, R., Boyd, S., De Micheli, G.: Temperature-Aware Processor Frequency Assignment for MPSoCs using Convex Optimization. In: Proc. Int'l Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS), pp. 111–116 (2007)

[10] Rai, D., Yang, H., Bacivarov, I., Chen, J.-J., Thiele, L.: Worst-Case Temperature Analysis for Real-Time Systems. In: Proc. Design, Automation and Test in Europe, DATE (March 2011)

[11] Skadron, K., Stan, M.R., Sankaranarayanan, K., Huang, W., Velusamy, S., Tarjan, D.: Temperature-Aware Microarchitecture: Modeling and Implementation. ACM T. Arch. and Code Opt. 1(1), 94–125 (2004)

[12] Thiele, L., Chakraborty, S., Naedele, M.: Real-Time Calculus for Scheduling Hard Real-Time Systems. In: Proc. IEEE Int'l Symposium on Circuits and Systems (ISCAS), vol. 4, pp. 101–104 (2000)

[13] Wandeler, E., Maxiaguine, A., Thiele, L.: Performance Analysis of Greedy Shapers in Real-Time Systems. In: Proc. Design, Automation and Test in Europe (DATE), pp. 444–449 (2006)

[14] Wandeler, E., Thiele, L., Verhoef, M., Lieverse, P.: System Architecture Evaluation using Modular Performance Analysis: A Case Study. Int'l J. on Software Tools for Technology Transfer (STTT) 8, 649–667 (2006)

[15] Wang, S., Bettati, R.: Reactive Speed Control in Temperature-Constrained Real-Time Systems. Real-Time Systems 39, 73–95 (2008)

[16] Yao, F., Demers, A., Shenker, S.: A Scheduling Model for Reduced CPU Energy. In: Hájek, P., Wiedermann, J. (eds.) MFCS 1995. LNCS, vol. 969. Springer, Heidelberg (1995)