Energy-efficient application mapping and scheduling for lifetime guaranteed MPSoCs

Liu, Weichen; Yi, Juan; Li, Mengquan; Chen, Peng; Yang, Lei

2019

Liu, W., Yi, J., Li, M., Chen, P., & Yang, L. (2019). Energy-Efficient Application Mapping and Scheduling for Lifetime Guaranteed MPSoCs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38(1), 1-14. doi:10.1109/TCAD.2018.2801242

<https://hdl.handle.net/10356/90103>

<https://doi.org/10.1109/TCAD.2018.2801242>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/TCAD.2018.2801242>.

Downloaded on 26 Apr 2023 13:32:17 SGT

Energy-Efficient Application Mapping and Scheduling for Lifetime Guaranteed MPSoCs

Weichen Liu, *Member, IEEE*, Juan Yi, Mengquan Li, Peng Chen, Lei Yang

Abstract—Energy optimization is one of the most critical objectives for the synthesis of multiprocessor system-on-chip (MPSoC). Besides, to ensure a long processor lifetime and to maintain a safe chip temperature are also important for multiprocessor manufactures under deep submicrometer process technologies. This paper presents a Mixed Integer Linear Programming (MILP) model to determine the mapping and scheduling of real-time applications onto embedded MPSoC platforms, such that the total energy consumption is minimized with the lifetime reliability constraint and the temperature threshold constraint satisfied. We develop a lightweight temperature model that can be integrated in the MILP model to predict the chip temperature accurately and efficiently. By exploiting the Dynamic Voltage and Frequency Scaling (DVFS) capability of modern processors, processor voltage/frequency assignment is also considered in our MILP model. Extensive performance evaluations on synthetic and real-world applications demonstrate the effectiveness of the proposed approach. Our MILP model achieves an average reduction of 19.09% and 28.53% total energy in comparison with two state-of-the-art techniques on the basis of guaranteeing the safe chip temperature and system lifetime reliability.

Index Terms—Multiprocessor system-on-chip, lifetime reliability, temperature model, real-time application mapping and scheduling, energy optimization.

I. INTRODUCTION

MultiProcessor System-on-Chip (MPSoC) is being widely used in various embedded application domains like telecommunication (*e.g.*, smart phones), transportation (*e.g.*, car electronics) and industrial automation (*e.g.*, process control). However, though have brought MPSoCs with enhanced functionality and improved performance, technology scaling and the ever-increasing demand for system performance have resulted in high power densities and increasing chip temperatures in current circuits. Both of them make failure mechanisms serious threats for reliable MPSoCs [1]. Chip temperature significantly affects the device lifespan and the circuit soft error rate, *e.g.*, a 10-15°C temperature increase can result in a 2× reduction in the lifespan of device [2], and a 216% increase in soft error rate is observed on c17 as the temperature climbs from 25°C to 125°C [3]. Moreover, the leakage power grows exponentially with the increasing temperature in the nanometer era [4], and high chip temperature also requires effective cooling and packaging solutions to ensure reliable runtime operation, resulting in a significant increase in cooling and packaging costs.

W. Liu is with School of Computer Science and Engineering, Nanyang Technological University, Singapore. Email: liu@ntu.edu.sg.

J. Yi is with Huawei Technologies CO., LTD, China. Email: wsiyjuan@hotmail.com. (Corresponding author: Juan Yi.)

M. Li, P. Chen, L. Yang are with College of Computer Science, Chongqing University, China.

To obtain the chip temperature profiles for thermal-aware managements on MPSoC platforms, numerous temperature estimation techniques have been proposed [5], [6], [7]. Most of them suffer from either inaccurate estimation precision or expensive computation cost (time consuming) during design space exploration. Inaccurate prediction (an underestimation or overestimation of chip temperature) would result in overheated processor cores and further affect the system reliability or reduced performance at runtime. Meanwhile, the temperature estimation process has to be computationally efficient.

In addition, ensuring a long processor lifetime by limiting wear-out related failures is also a critical requirement for all multiprocessor manufactures in deep submicrometer process technologies. As demonstrated in [8], the average Mean Time To Failure (MTTF) of a contemporary superscalar processor drops by about 4× from 180nm to 65nm technology nodes. The ITRS has identified lifetime reliability as a growing challenge for users and designers of all types of integrated circuits [9]. It is expected that, in the near future, the design cost and effort required for ensuring lifetime reliability will be as high as for classic design constraints, *e.g.*, timing and power consumption. Therefore, it is essential to consider the lifetime reliability issue in embedded system design.

Energy optimization has been a major factor to be considered for MPSoCs at system-level design. DVFS techniques are widely explored for energy optimization. Considering both the energy consumption and the system reliability in terms of lifetime reliability *or* chip temperature on MPSoCs, extensive researches have been proposed [10], [11], [12], [13], [14], [15]. However, existing works on energy-efficient application mapping and scheduling techniques only consider thermal control or lifetime reliability constraint separately. On-chip temperature profile and system lifetime reliability affect each other and complement each other. On the one hand, chip temperature is necessary to be considered to guarantee system lifetime reliability; many failure mechanisms are exponentially dependent on chip temperature. Considering chip temperature ensures that the chip temperature is always below safe threshold at any moment. Although temperature-aware methods [16], [15], [17] are effective in reducing temperature, they may not enhance and may even hamper the system lifetime reliability that is also related to other aging factors, such as current density and voltage. On the other hand, ensuring processors' lifetime cannot perfectly prevent chip temperature from violating the safe threshold, since the system lifetime reliability is estimated based on the whole scheduling period and is a result of the accumulation effect of multiple factors. The complicated relationship between chip temperature and

system lifetime reliability requires that both of them should be considered explicitly for the system design of energy-efficient techniques.

In this paper, we propose an effective Mixed Integer Linear Programming (MILP)-based approach to conduct energy-efficient application mapping and scheduling on MPSoC platforms, which explicitly takes both the transient impact caused by chip temperature and the duration impact resulted from lifetime reliability on MPSoCs into consideration. To estimate the chip temperature and determine the effect of temperature on chip's aging accurately and efficiently, we develop a lightweight temperature model that is integrated in the MILP model. Our MILP model minimizes the energy consumption of applications by judiciously determining their mapping and scheduling based on voltage/frequency scaling techniques. Chip temperature is guaranteed below the safe threshold and the chip lifetime would be at least the predefined target value. The accuracy and efficiency of the temperature model are verified by comparing with existing temperature estimation techniques; Compared to the HotSpot simulator that has been widely recognized for the architecture-level temperature modeling on MPSoCs, an average of over 99% accuracy is presented. The effectiveness and scalability of the proposed MILP-based technique are verified by a huge set of synthetic applications and real-world applications on different floorplans. The experimental results show that our approach reduces 24.3% reliability threshold violations on average and achieves 19.09% and 28.53% total energy savings in comparison with two state-of-the-art techniques.

The contributions of this paper are listed as follows:

- An accurate and efficient temperature model is developed, which achieves an average of over 99% temperature prediction accuracy with greatly reduced computation time, compared to a widely recognized temperature simulator.
- A systematical approach for energy-efficient application mapping and scheduling on MPSoC platforms is proposed, in which chip temperature profile and lifetime reliability issue are explicitly taken into consideration simultaneously and are well controlled. Performance evaluations on extensive synthetic and real-world applications prove the effectiveness of the proposed approach.

The rest of this paper is organized as follows: Section II discusses related work; Section III presents the system models and reliability modeling techniques; Section IV uses an example to illustrate the motivation of this paper, and then formulates the problem studied in this paper; Section V presents the lightweight temperature model, and Section VI presents the MILP-based energy optimization technique; Experimental results and concluding remarks are provided in Section VII and Section VIII, respectively.

II. RELATED WORK

A significant body of work has been published on DVFS-aware task allocation and scheduling techniques for thermal control or lifetime optimization. Chantem *et al.* [18], [14], [15], [19] tried to minimize the maximum temperature of MPSoCs by utilizing the DVFS technique, subjecting to the

real-time constraint of applications. Huang and Xu *et al.* [10] proposed a simulated annealing based algorithm to minimize energy for MPSoC considering the lifetime reliability constraint. Das *et al.* [11] tried to maximize system lifetime while maintaining the energy consumption under a given energy budget using DVFS and task replication. Hanumaiah *et al.* [12] maximized the overall energy efficiency yet maintained the safe chip temperature by considering DVFS and thread migration. Zhuravlev *et al.* [20] surveyed the related scheduling techniques performing energy-efficient computation. These works have mainly tackled one aspect of the complex scenario: by acting on DVFS and processor frequency assignment for optimizing the trade-off between performance and energy, or by accurately mapping the application to improve the system lifetime. However, none of these works takes both the chip temperature control and the system lifetime reliability into consideration explicitly.

Both system-level lifetime design methods and thermal-aware design techniques require accurate thermal estimation techniques. A thermal emulation framework for field-programmable gate arrays (FPGAs) was introduced in [5], which allows designers to extract a number of critical statistics from processor cores, memories and interconnection systems. Another temperature model developed by Dhodapkar *et al.* [7], called TEMPEST, models the temperature directly using an equivalent RC circuit containing only a single RC pair for the entire chip. HotSpot [6] is an automated temperature model which uses the duality between heat flow and electrical circuit phenomena. Each core on the die is divided into several blocks, with the same division being applied to the Thermal Interface Material (TIM). The heat spreader and the heat sink are also divided into multiple blocks. For a processor with n cores and m functional units for each core, there will be $n * m$ block in the die, another $n * m$ blocks in the TIM and 14 blocks in the package according to the model. HotSpot has been widely recognized for the architecture-level temperature modeling on MPSoCs. However, the computation efficiency of HotSpot limits its application for quick temperature estimation during design space exploration.

III. PRELIMINARIES

In this section, we introduce the application model, system model, power model, and lifetime reliability model used in this paper in turn.

A. Application model

We use a Data Flow Graph (DFG) to model the application to be executed in the system. A DFG $G = (\mathcal{T}, \mathcal{E})$ is a *directed acyclic graph* (DAG), where $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{T} \times \mathcal{T}$ is the set of edges that defines the precedence relations among the nodes in $\mathcal{T}$. Each node $\tau_i \in \mathcal{T}$ represents a task to be executed. Each edge in the graph ($e_{ij} \in \mathcal{E}$) indicates that task τ_j is dependent on τ_i . A deadline L is associated with a DFG G , which indicates that the application G has to be completed before L . We assume the worst-case execution times (WCETs) of the tasks are known a priori.

B. System Model and Power Model

We consider a MPSoC architecture consisting of a set of processor cores, which is defined as $\mathcal{M}$. For each processor, we can assign different frequencies and voltages at any time. Unlike the static frequency assignment with fixed operating frequency at all time, the dynamic case that varies the frequencies and voltages gains a better performance and increases the energy efficiency. We assume that the processor cores are architecturally identical and the only source of heterogeneity is that the processors can operate at different voltage and frequency levels, as assumed in [21], [22].

Each processor has K discrete voltage settings $V = \{V_1, V_2, \dots, V_K\}$ (in descending order). Each voltage setting V_k can be associated with a computation speed v_k in terms of cycles/second. The energy consumption for executing any $\tau_i \in \mathcal{T}$ at speed v_k can be expressed as $E_{i,k} = Pw(V_k, v_k) \times t_{i,k}$, where $Pw(V_k, v_k)$ is the power consumption under voltage V_k and speed v_k , and $t_{i,k}$ is the execution time of task τ_i with speed v_k . Assuming the tasks execute up to their WCETs and $WCET_i$ is given as the execution time of task τ_i in the default (highest) processor frequency, the WCET for τ_i running at processor speed v_k is computed as: $t_{i,k} = WCET_i \times (v_1/v_k)$.

A processor core has two operating modes: an active mode, in which it is able to execute instructions, and a sleep mode, in which it ceases working and rests with minimized power consumption. The processor cores consume both dynamic power and leakage power while executing instructions, and only consume leakage power during idle periods. The leakage power in a sleep mode is 3% of the leakage power in the active mode running in the lowest frequency [23]. Dynamic power dominates the power consumption of processors in micrometer CMOS technology. While leakage power increases its percentage in total power consumption as chip density increases. Both dynamic power consumption and static power consumption need to be considered in overall energy optimization of the system. The power consumption under different voltage/frequency settings are similar to the model in [24], [25].

C. Lifetime reliability model

In this paper, we focus on two critical failure factors, i.e., the electromigration (EM) and time dependent dielectric breakdown (TDDB), in the lifetime reliability model. Various failure mechanisms have been extensively studied in the literature, such as interconnect shorts/opens during fabrication that can be weeded out during the manufacturing test and burn-in process, radiation effects that do not fundamentally damage the circuit and are not viewed as lifetime reliability threats, and the hard errors of intrinsic failures that are permanent once they manifest. The most representative hard errors include TDDB in the gate oxides and EM in the interconnects [1]. Both of them are widely recognized to decrease the reliability of chips (integrated circuits (ICs)) significantly.

Electromigration (EM) is the transport of material resulted from the gradual movement of the ions in the conducting path due to the momentum transfer between conducting electrons and diffusing metal atoms. It is a typical reason that will lead

to the permanent faults such as opening of metal lines/contacts, shortening between adjacent metal lines, etc. The MTTF of a wire, taking EM into consideration, is as follows [26]:

$$MTTF_{EM} \propto (J - J_{crit})^{-n} \times e^{\frac{E_a}{k \times T}}$$

where J is the current density in the interconnect, J_{crit} is the threshold current density, E_a is the activation energy for electromigration, k is Boltzmann's constant, and T is absolute temperature in Kelvin. n and E_a are constants that depend on the interconnect metal used. J tends to be much higher than J_{crit} in interconnects (nearly 2 orders of magnitude [27]). Hence, $J - J_{crit} \approx J$. J for an interconnect can be related to the switching probability of the line p [28]:

$$J = \frac{C \times V_{dd}}{W \times H} \times f \times p \quad (1)$$

where C , W , and H are the capacitance, width, and thickness of the line, respectively, f is the clock frequency, and V_{dd} is the supply voltage. We assume all interconnects in a structure to be similar, and do not differentiate interconnects on the basis of their C , W , and H .

Time Dependent Dielectric Breakdown (TDDB) is a wear-out mechanism of the gate dielectric, when the gate oxide breaks down as a result of long-time application of relatively low electric field (as opposite to immediate breakdown, which is caused by strong electric field). The breakdown is caused by formation of a conducting path through the gate oxide to substrate due to electron tunneling current, when MOSFETs are operated close to or beyond their specified operating voltages. With the wide application of low- k and ultralow- k dielectric materials at the 90-nm technology node, TDDB becomes a major system lifetime reliability threat and gets even worse with further scaling [29]. The MTTF affected by TDDB is based on recent experimental work in [30]:

$$MTTF_{TDDB} \propto \left(\frac{1}{V}\right)^{a-b \times T} \times e^{\frac{X+Y/T+Z \times T}{k \times T}}$$

where T is the absolute temperature in Kelvin, a, b, X, Y, Z are fitting parameters, and V is the voltage.

The models of the above failure mechanisms have been widely proposed [26], [31], [32]. It is shown that they are strongly related to the temperature and voltage applied to the circuit. Figure 1 shows the impact of voltage scaling on the chip lifetime under the EM and TDDB mechanisms. The parameters used in EM and TDDB are from [1] and [30], respectively. We can see that the chip lifetime under both EM and TDDB decrease as voltage goes up, and that under TDDB is highly sensitive to voltage. For this reason, we consider voltage selection for each task when evaluating a core's lifetime based on a schedule.

To estimate the combined lifetime of the platform, we adopt the lifetime estimation model proposed in [33]. The adopted model can avoid lifetime underestimation by distinguishing the impact of every individual mechanism compared to SOFR model [8]. The key fact ignored by SOFR model is that different failure mechanisms impact on different components of a system, for example, TDDB affects transistors, while

Fig. 1: The impact of voltage scaling on the chip lifetime under the failure mechanisms of EM and Tddb.

EM affects interconnects. In other words, the effects of failure mechanisms impacting on different components are independent from the spatial perspective, and thus should not be additive. The combined MTTF of a system component P_i , denoted by $MTTF(P_i)$, is determined by the minimum MTTF across various failure mechanisms [33]:

$$MTTF(P_i) = \min(MTTF_j(P_i)) \quad (2)$$

where $MTTF_j(*)$ is the independent MTTF result due to an individual failure mechanism (e.g., EM or Tddb).

We evaluate the lifetime of each core for a given schedule based on (3) and (4). According to the EM model, the lifetime of core m is calculated based on the tasks assigned to it and the corresponding frequencies. In (3), $X_{i,m,f}$ is 1 if task i is mapped to core m with speed f , and 0 otherwise. J_f is the current density when the core runs under speed f , which is calculated by (1). $Task_m$ is the number of tasks mapped to core m . TH is the properly predefined temperature threshold.

$$MTTF_{EM}(m) = \sum_{i \in \mathcal{T}} \sum_{1 \leq f \leq K} \frac{X_{i,m,f} \times A_0}{(J_f - J_{crit})^n \times Task_m} \times e^{\frac{E_a}{k \times TH}} \quad (3)$$

For the Tddb model, the lifetime is calculated based on the tasks assigned to it and the corresponding voltage levels. V_f is the corresponding voltage level when the core runs with speed f .

$$MTTF_{TB}(m) = \sum_{i \in \mathcal{T}} \sum_{1 \leq f \leq K} \frac{X_{i,m,f} \times A_0}{V_f^{a-b \times TH} \times Task_m} \times e^{\frac{X+Y/TH+Z \times TH}{k \times TH}} \quad (4)$$

For moderate temperatures in 65nm technology, Srinivasan *et al.* [34] demonstrate that the contribution of electromigration, dielectric breakdown, and cycling to the overall failure rate are similar to each other. This allows us to calibrate the constants in each failure equation to give a system MTTF of 30 years at nominal temperature [35]. And we used the same constants for all the experiments. It is noteworthy that the relative impact of different failure mechanisms might change depending on the conditions.

IV. PROBLEM DEFINITION

There may be numerous ways in which an application can be mapped and scheduled to the hardware components, but different scheduling techniques would result in a great difference in the total energy consumption. Various scheduling solutions change the components' power density, voltage, and temperature profiles, and therefore allowing for a wide range of lifetime enhancement and energy optimization without requiring additional investment in hardware or affecting performance. It provides us a potential chance to improve the energy efficiency of MPSoCs by searching for the optimal application mapping and scheduling with optimized total energy consumption.

In this paper, we tackle the problem of energy-efficient application mapping and scheduling on MPSoC platforms considering both chip temperature and lifetime reliability constraints simultaneously.

A. Motivational Example

In this section, we use a simple example to illustrate how the chip temperature, system lifetime reliability, and total energy consumption affect one another.

Fig. 2: Motivational example.(a)Task graph;(b)Execution time

The example application is composed of 6 tasks whose execution times under different processor speeds are shown in Figure 2a. There are three voltage/frequency levels for each core: $k1, k2, k3$. $k1$ represents the original (highest) level and $k3$ represents the lowest one. A task requires different execution times under different voltage/frequency levels. For example, task τ_1 takes 7, 8, and 10 time units to complete under levels $k1, k2$, and $k3$, respectively. We run this application on a chip with two identical cores $P1, P2$, assuming the temperature threshold to be 70°C which is considered as a high temperature value for a silicon chip. Based on the analysis in Section III-C, we set the deadline for the application is 40 and the targeted service life of this chip is 10 years.

Figure 3 gives three different schedules for this application. Different voltage/frequency levels are distinguished by different patterns and colors. The tasks marked in dark gray are executed under level $k1$, and the tasks marked in light gray are executed under level $k3$.

In both schedules (a) and (b), each task runs under the same voltage/frequency level, and the chip temperatures are below 70°C . That means that both of them satisfy the thermal threshold and consume the same total energy. However, the schedule (a) leads to much shorter chip lifetime compared

Fig. 3: Three different schedules for the example task graph in Figure 2a.

to schedule (b). The chip can only survive 5 years based on our EM and TDDb models in accordance with schedule (a) because of the excessive stress on P1. Although it is able to keep the chip temperature under the thermal threshold, P1 always runs under the highest voltage/frequency level for each task, and ends up with a high temperature close to the thermal threshold and poor lifetime performance.

In fact, for a given system temperature threshold, there are significant variations on the achievable lifetime by different task mapping and scheduling decisions. From the results of the experiments on two benchmark sets including 20 task graphs, we find that not all the scheduling results that are generated by only considering the thermal constraints can meet the targeted chip service lifetime requirements. On average, the system lifetime is reduced by 17.3% compared to the results obtained by taking both thermal constraints and lifetime constraints into account. In addition, satisfying the targeted chip service lifetime requirement cannot ensure a safe on-chip temperature. The system lifetime reliability is estimated based on the whole scheduling period and is a result of the accumulation effect of multiple factors, such as temperature, current density, and voltage level. Therefore, it is important to include both lifetime reliability optimization and temperature management in designing task mapping and scheduling strategies for system reliability.

Schedule (c) also satisfies all the performance requirements including timing constraints and expected system lifetime. What's more, the schedule achieves minimized total energy consumption by further adjusting the voltage/frequency levels of certain processor cores, and 10.8% further energy reduction is observed compared to schedule (b). In fact, schedule (c) is the optimal solution to the problem, obtained by our MILP model (refer to Section VI).

B. Problem formulation

Based on the above analysis, the problem addressed in this paper is formulated as follows: *Given the floorplan of a MPSoC containing a set of processor cores $\mathcal{M}$, an application represented with a task graph $G = \langle \mathcal{T}, \mathcal{E} \rangle$, its associated deadline L , a thermal threshold TH , and a target system lifetime LF , determine a non-preemptive schedule of the application and the corresponding voltage/frequency level of the core running each task, such that the total energy consumption is minimized, under the constraints that tasks are finished before deadlines, the chip runs below thermal threshold TH , and the system lifetime is guaranteed to be no less than LF .*

V. LIGHTWEIGHT TEMPERATURE MODEL

In this section, we present a lightweight temperature model for on-chip temperature prediction.

The heating/cooling process of MPSoCs is complicated and dynamic. As the approaches in [15], [36], [13] show, inspired by the duality between heat flow and electrical circuit phenomenon, the heating/cooling process in an MPSoC can be approximately modeled as an equivalent first-order RC circuit by applying Fourier's Law. The thermal coefficients can be obtained in [37].

For a MPSoC platform with a set of processor cores $\mathcal{M}$, each core is modeled as a discrete thermal element. Heat sink and heat spreader layers are on top of the cores. We combine the heat sink layer and heat spreader layer as a whole and still call it heat sink layer for convenience, similar to the model in [6], [15]. Heat sink layer can also be modeled as multiple thermal elements whose partitioning corresponds to the layout of the cores. Define $\mathcal{H}$ as the set of heat sink elements in the MPSoC according to their partition.

The heat transfer among cores and heat sink elements is quite related to the power consumption of processor cores, the ambient temperature, and the temperature of neighboring thermal elements. When the power consumption of each core is the same, the percentage of energy converted to heat among cores is very low, around 3%. Theoretically, there is no heat transfer between the cores. However, in the situation where a core is idle and its neighbors are running tasks, a relatively high percentage of energy is converted to heat, and the core absorbs heat from its neighbors. A part of the heat is dissipated through heat sink, and the rest is absorbed by itself which leads to a rise of its temperature.

Let C_e be the thermal capacitance for any thermal element $e \in \mathcal{M} \cup \mathcal{H}$. Let $R_{m,l}$ be the lateral thermal resistance between core m and l for any $m, l \in \mathcal{M}$. Let $R_{h,g}$ be the thermal resistance between two heat sink elements h and g for any $h, g \in \mathcal{H}$. Let $R_{h,A}$ be the vertical resistance of a heat sink elements $h \in \mathcal{H}$ to the ambient. For each task-to-core assignment, $Pw(\tau, m)$ indicates the power consumption of core m when executing task τ . The temperature of thermal element $e \in \mathcal{M} \cup \mathcal{H}$ at time t is denoted by $T(t, e)$. The temperature of each thermal element can be expressed as a function of its power consumption, the ambient temperature, and the temperatures of neighboring thermal elements. The heating/cooling process by Fourier's Law can be formulated as follows:

$$\sum_{\tau \in T} \alpha(t, \tau, m) \times Pw(\tau, m) = \sum_{l \in N_m} \frac{T(t, m) - T(t, l)}{R_{m,l}} \quad (5)$$

$$+ C_m \times \frac{dT(t, m)}{dt} + fc \times \frac{T(t, m) - T(t, h)}{R_{m,h}}$$

$$fc \times \frac{T(t, m) - T(t, h)}{R_{m,h}} = \sum_{g \in N_h} \frac{T(t, h) - T(t, g)}{R_{h,g}} \quad (6)$$

$$+ C_h \times \frac{dT(t, h)}{dt} + fh \times \frac{T(t, h) - T_A}{R_{h,A}}$$

where $T(t, h)$ is the temperature of the heatsink element h at time t (h is vertically on top of core m), and $\alpha(t, \tau, m) = 1$ if task τ executes on active core m at time t . The thermal resistance parameters between any two thermal elements can be calculated easily as described in [37]. The steady-state temperature values can be obtained when the capacitor values C_m and C_h both equal zero in (5) and (6).

To make the temperature model more robust to different configurations, we add two scaling factors, fc and fh , in the model, as (5) and (6) suggest. Both fc and fh are applied for adjusting vertical thermal resistances. The vertical thermal resistance captures heat flow from one layer to the next, moving from the die through the package and eventually into the air. For example, $R_{m,h}$ counts the amount of heat flowing from core m into the heat sink elements h that is directly above the core m . The value of the vertical thermal resistance of a specific architecture generally indicates the heat dissipation rate. The lateral thermal resistance captures the heat diffusion between adjacent thermal elements within a layer. It indicates the temperature difference between the adjacent thermal elements. It is observed that the temperature value estimated by [15] is lower than that obtained from HotSpot. This is because the heat spreader layer is omitted and the divided equivalent circuit is not accurate enough, though the temperature differences among adjacent thermal elements are well presented. To offset the accuracy loss from imprecise vertical thermal resistance, two scaling factors are used to adjust the vertical thermal resistance empirically in an efficient manner. The two scaling factors can be calculated easily as described in [4].

To testify the prediction accuracy of the temperature model, we compare the predicted temperature values from the temperature model with the simulated results from HotSpot simulator on a 2×2 -core mesh-based MPSoC. Every core is simulated as out-of-order Alpha 21364 in 32nm technology. We execute 10 different sets of synthetic applications. The power consumption traces of every set of applications are generated by McPAT [38] and are used as the input to both the temperature model and HotSpot to obtain the respective estimated temperature values. Detailed experimental setup refers to Section VII. Figure 4 shows the steady-state temperature results of each core under different power consumption traces. The unit of temperature is uniformly Celsius degree in this paper. Since the temperature of every processor core in each set of synthetic applications is distinctly different, we draw the normalized values to clearly illustrate the temperature differences between

the values obtained from our temperature model and those from HotSpot. The normalization is performed for the 4 cores individually. For each core under every power consumption trace, the temperature value obtained by HotSpot simulator is normalized to be 1, as the ground truth value; the prediction value obtained by our temperature model is normalized with respect to the HotSpot value. Each curve represents the normalized temperature values of a processor core for various power consumption traces. The temperature values obtained from the proposed model and HotSpot simulator only differ in 0.3% on average. Besides, the running time of our model can be reduced by over two orders of magnitude compared to HotSpot simulator-based technique. The execution time of the temperature model is less than 60 μ s, whereas HotSpot takes about 0.2s for temperature simulation.

Fig. 4: The steady-state temperature estimation on different processor cores.

Figure 5 shows the transient temperature behavior variations of one processor core when executing different tasks under different speeds. The red curve and the blue curve represent transient thermal results generated by using the temperature model and HotSpot, respectively. The light blue curve is the steady state temperatures curve for the core. We can see that the transient temperature results of our model is consistent with that generated from HotSpot. This observation once again clearly testifies that the temperature model is accurate enough to be used for chip temperature prediction.

Our temperature model provides accurate temperature prediction with reduced computation time in high-level modeling. HotSpot-based simulation has been widely applied in architecture level thermal modeling on MPSoCs. It achieves high accuracy while incurs heavy computation overhead [39]. The thermal model proposed in [15] simplifies the high-order model of HotSpot and models the heat dissipation process into an equivalent first-order RC circuit based on the Fourier's Law. It can obtain the chip temperature distribution by solving a set of equivalent linear equations in polynomial time. However, the accuracy of the simple RC thermal model is not perfectly promised due to the extreme uncertainty and complexity of heat dissipation process. Our temperature model is also a linearized formal method, similar to the model in [15]. While we have added two scaling factors, fc and fh ,

Fig. 5: The transient temperature variation of a processor core.

to adjust the dissipation rate of the heat transfer process, consequently increasing the accuracy of the temperature model in an efficient manner. The two scaling factors are empirically trained with respect to the HotSpot values at design time. Our temperature model is suitable to be combined with high-level optimization techniques for real-time application mapping and scheduling thanks to the high accuracy and efficiency.

VI. MILP-BASED APPLICATION MAPPING AND SCHEDULING APPROACH

In this section, we formally propose a MILP model to solve the problem formulated in Section IV-B.

A. Inputs and Decision

For our MILP formulation, we provide several input variables that represent the power consumption and characteristics of the target applications and platform, as shown in Table I. The $t_{i,k}$ and $Pw(V_k, v_k)$ parameters are calculated based on worst case execution time (WCET) of every task node for every frequency level supported by the processor (as discussed in Section III-B).

TABLE I: Input variables for the MILP formulation.

Inputs	Description
$t_{i,k}$	worst-case execution time of task i when running at speed v_k .
$\mathcal{T}$	the set of tasks.
$\mathcal{M}$	the set of cores, and the number of thermal elements is $2 * \mathcal{M} $.
$\mathcal{H}$	the set of heat sink elements.
K	the set of discrete voltage settings of each core.
L	given timing constraint of the application (deadline).
LF	targeted chip lifetime.
V_k/v_k	the k -th voltage/speed level.
$Pw(V_k, v_k)$	the power consumption of the cores running with speed v_k .
Pw_{sl}	the power consumption when core is with the sleep mode.

There are two major requirements for decision variables in our MILP problem: firstly, they must form a complete

representation of a feasible execution schedule; secondly, they should make it possible to represent all constraints and objectives as linear formulations. The decision variables used in our formulation are shown in Table II. The variables $X_{i,m,k}$, s_i and f_i are used to construct the objective function. Binary variable $O_{i,j}$ is used to ensure that no task overlaps with another task on the same core. $TS_{i,m}$ and $\beta_{i,j,m,k}$ are used to model chip temperature distribution and lifetime value.

TABLE II: Decision variables used in the MILP model.

Variables	Description
s_i	the start time of task i .
f_i	the finish time of task i .
$X_{i,m,k}$	binary variable, equals to 1 if task i executes on core m with speed k .
$O_{i,j}$	binary variable, equals to 1 if and only if task j finishes after task i begins.
$TS_{i,m}$	the temperature of thermal element m when task i starts to execution.
$\beta_{i,j,m,k}$	binary variable, equals to 1 if task j executes on core m with speed k , precedes and overlaps with task i .

B. Optimization Objective

Our objective is to minimize total energy consumption, including the dissipated energy when processor executing tasks and the energy when the core being idle. $[L * |\mathcal{M}| - \sum_{i \in \mathcal{T}} (f_i - s_i)]$ is the total time of all cores staying in the sleep mode, and Pw_{sl} is the corresponding power consumption in the sleep mode. $\sum_{i \in \mathcal{T}} \sum_{m \in \mathcal{M}} \sum_{k \in K} X_{i,m,k} * t_{i,k} * Pw(V_k, v_k)$ is the total energy consumption for executing all the tasks.

$$\begin{aligned} & \min \{ [L * |\mathcal{M}| - \sum_{i \in \mathcal{T}} (f_i - s_i)] * Pw_{sl} \\ & + \sum_{i \in \mathcal{T}} \sum_{m \in \mathcal{M}} \sum_{k \in K} X_{i,m,k} * t_{i,k} * Pw(V_k, v_k) \} \end{aligned}$$

C. Constraints

To satisfy the real-time requirements of applications on a target MPSoC platform within safe chip temperature and lifetime reliability demands, the constraints are listed as follows.

Constraint for target platform: each task is assigned to exactly one core with an assigned execution speed.

$$\sum_{m \in \mathcal{M}} \sum_{k \in K} X_{i,m,k} = 1 \quad \forall i \in \mathcal{T} \quad (7)$$

Timing constraints: all the tasks must be finished before L to meet the deadline constraint, as shown in (8). In addition, each task must follow the precedence relation defined by the DFG. (9) model this dependency by forcing destination task nodes to start only after their predecessor nodes have finished.

$$\begin{aligned} f_i &= s_i + \sum_{m \in \mathcal{M}} \sum_{k \in K} X_{i,m,k} \times t_{i,k} \quad \forall i \in \mathcal{T} \\ f_i &\leq L \quad \forall i \in \mathcal{T} \end{aligned} \quad (8)$$

$$f_i \leq s_j \quad \forall (i, j) \in \mathcal{E} \quad (9)$$

Mutual exclusion: Mutual exclusion ensures that no task overlaps with another task on the same core. Variable O_{ij} is used to guarantee that. O_{ij} is 1 iff task j finishes after task i begins as shown in (10) and (11). Z is a very large constant number that no less than the maximum finish time.

$$f_j - s_i \leq Z \times O_{ij} \quad \forall i, j \in \mathcal{T} \quad (10)$$

$$\epsilon + s_i - f_j \leq Z \times (1 - O_{ij}) \quad \forall i, j \in \mathcal{T} \quad (11)$$

Based on the definition of O_{ij} , $O_{ij} + O_{ji} = 2$ if and only if the executions of the two tasks i and j overlap. According to (12), $\sum_{k \in K} X_{i,m,k}$ and $\sum_{k \in K} X_{j,m,k}$ cannot be 1 at the same time if $O_{ij} + O_{ji} = 2$, indicating if i and j overlap, they will not execute on the same core. In contrast, $O_{ij} + O_{ji}$ would be less than 2 if $\sum_{k \in K} X_{i,m,k}$ and $\sum_{k \in K} X_{j,m,k}$ are both 1, indicating their executions must be serialized if they are mapped to the same core.

$$O_{ij} + O_{ji} + \sum_{k \in K} X_{i,m,k} + \sum_{k \in K} X_{j,m,k} \leq 3 \quad (12)$$

$$\forall i, j \in \mathcal{T}, i \neq j, \forall m \in \mathcal{M}$$

Thermal threshold: We can use (5) and (6) to compute the temperature for each core at any given time corresponding to dynamic or transient thermal analysis.

Since the power consumptions for each core only change at the beginning and end of a task's execution, and the steady-state temperature of a core only experiences a rapid change when the power consumption of at least one core on the chip changes. We can reduce the amount of computation for obtaining the temperatures during tasks execution. Specifically, we only need to evaluate the temperature of each core immediately after a task starts or finishes on any core of the MPSoC. Besides, the tasks run for a duration of at least the thermal time constant of chip die (in a few milliseconds). The thermal time constant is defined as the time for the chip peak temperature to reach 63% of the steady-state after turning on the power. For the Alpha 21364 circuit in 32nm, the largest time constant for any chip block was 10 ms [40]. Thus the thermal die capacitances are normally saturated during tasks execution. For this reason, we set the capacitance values in (5) and (6) to zero to obtain the steady-state temperature when predicting temperatures in our MILP formulation. In addition, the differential part is eliminated, which greatly enhance the computation efficiency during design space exploration.

Variable $\beta_{i,j,m,k}$ is used to model the temperature estimation process. $\beta_{i,j,m,k} = 1$ if and only if task j executes on core m with speed k , precedes and overlaps with task i , that is, $\beta(i, j, m, f) = 1$ iff $O_{ij}, O_{ji}, X_{j,m,k}$ all equals to 1 and $s_j \leq s_i$.

$\forall i, j \in \mathcal{T}, \forall m \in \mathcal{M}$, we have:

$$3 \times \beta_{i,j,m,k} \leq O_{ij} + O_{ji} + X_{j,m,k}$$

$$s_j \leq s_i + (1 - \beta_{i,j,m,k}) * Z$$

$$s_i + \epsilon - (3 - (O_{ij} + O_{ji} + X_{j,m,k})) \times Z \leq s_j + \beta_{i,j,m,k} * Z$$

(13) and (14) calculate the temperatures for each thermal element based on the proposed temperature model when any core begins to execute a new task. (15) states that the steady-state temperature (TS) of each core must be no more than TH when a new task begins to execute on a core.

$$Pw_{sl} + \sum_{j \in \mathcal{T}} \sum_{1 \leq k \leq K} \beta_{i,j,m,k} \times (Pw(V_k, v_k) - Pw_{sl})$$

$$= \sum_{l \in N_m} \frac{TS_{i,m} - TS_{i,l}}{R_{m,l}} + fc \times \frac{TS_{i,m} - TS_{i,h}}{R_{m,h}} \quad (13)$$

$$fc \cdot \frac{TS_{i,m} - TS_{i,h}}{R_{m,h}} =$$

$$\sum_{g \in N_h} \frac{TS_{i,h} - TS_{i,g}}{R_{h,g}} + fh \times \frac{TS_{i,h} - T_A}{R_{h,A}} \quad (14)$$

$$TS_{i,m} \leq TH \quad \forall i \in \mathcal{T}, \forall m \in \mathcal{M} \quad (15)$$

The temperature of each core when a task finishes its execution can be modeled in the same way. Due to space limitations, this part is omitted here.

Lifetime constraint: Based on the model in (2), we only need to guarantee that the predicted lifetime of each core under EM or TDDb is no less than LF , as shown in (16) and (17). It is based on our lifetime reliability model (refer to (3) and (4)). For EM, we take the physical parameters like current density into account. For TDDb, the different voltage/frequency levels lead to different predicted lifetime values, which is handled in (17). By applying the predefined temperature threshold TH , (16) and (17) linearize the lifetime estimation model. The expected lifetime of one core will be at least LF , since the actual temperature is less than or equal to TH .

$\forall m \in \mathcal{M}$, we have:

$$\sum_{i \in \mathcal{T}} \sum_{k \leq K} \frac{X_{i,m,k} \times A_0}{(J_f - J_{crit})^n} \times e^{\frac{E_a}{k \times TH}} \geq LF \times \sum_{i \in \mathcal{T}} \sum_{k \in K} X_{i,m,k} \quad (16)$$

$$\sum_{i \in \mathcal{T}} \sum_{1 \leq f \leq K} \frac{X_{i,m,f} \times A_0}{V_f^{a-b \times TH}} \times e^{\frac{x + \frac{Y}{TH} + Z \times TH}{k \times TH}}$$

$$\geq LF \times \sum_{i \in \mathcal{T}} \sum_{k \in K} X_{i,m,k} \quad (17)$$

Based on the defined input and decision variables, optimization objective, and formulated constraints, we employ a MILP solver to systematically explore all the application mapping and scheduling alternatives to obtain a globally optimal solution with minimum total energy consumption. We allow every task of an application to map to any a processor core with any voltage/frequency level. For the tasks that map to the

same core, we provide a total scheduling order by employing order-based scheduling (run-to-completion scheduling). Each task must follow the precedence relation and be finished before the deadline. We also consider both chip temperature and system lifetime reliability constraints. Equation (13)–(15) estimate the steady-state temperature profiles and limit the chip temperature under the safe threshold. Equation (16)–(17) limit the obtained solution satisfying the target demand of system lifetime reliability.

Our MILP technique will be conducted at design time in the warehouse to obtain the optimal application mapping and scheduling solution with minimum energy consumption. The obtained solution will be applied at runtime efficiently with very little overhead. The design-time optimization (e.g., on high-performance workstation) and runtime application (e.g., on MPSoC) can be executed on different platforms. Moreover, our MILP technique gains in energy optimization based on the guaranteed optimality.

VII. PERFORMANCE EVALUATION

In this section, we first verify the accuracy and efficiency of the proposed temperature model by comparing it with existing temperature models. We then conduct extensive experiments on different MPSoC floorplans with a large number of synthetic and real-world applications to verify the effectiveness and scalability of the proposed MILP-based application mapping and scheduling approach by comparing it with two state-of-the-art optimization techniques.

In the experiments, we consider mesh-based MPSoC platforms. Every processor core of the MPSoC platforms is simulated as out-of-order Alpha 21364 core in 32nm technology. We assume each processor core can be adjusted at four different voltage and frequency levels for DVFS: 1.210V at 2.1GHz, 1.200V at 2.0GHz, 1.187V at 1.9GHz, and 1.06V at 1.7GHz. The power consumption and area cost of the processor cores are simulated by McPAT simulation. According to our simulation results, each core has an area of 9.12mm^2 .

All the experiments are performed on a Dell Precision T5610 Tower Workstation with Red Hat Enterprise Linux Workstation release 6.4, based on Intel[®] Xeon[®] E5-2650 at 2.6GHz and 64GB memory.

A. Evaluation of the temperature model

We compare our temperature model with a widely used temperature model presented in [15] and a HotSpot simulator for accuracy verification on chip temperature prediction. The resulting temperature values of the HotSpot v5.0.2 are used as reference values. The parameter settings used in HotSpot and the temperature models are listed in Table III.¹

As shown in Figure 4 and 5, the temperature model well estimates the temperature behavior of every individual core, no matter the steady-state temperature or the transient one. In this section, we further verify the accuracy of the temperature model on MPSoCs of different sizes. We randomly generate

TABLE III: Parameters in HotSpot and temperature models.

Parameter	Value
Silicon thermal conductivity [W/m·K]	100
Silicon specific heat [J/m ³ ·K]	$1.75 \cdot 10^6$
Thickness of the chip [mm]	3.5
Convection resistance [K/W]	2
Heatsink thickness [mm]	2
Heatsink thermal conductivity [W/m·K]	400
Heatsink specific heat [J/m ³ ·K]	$3.55 \cdot 10^6$
Ambient temperature [K]	318.15

10 sets of synthetic applications for every floorplan of MPSoC, 2×2 , 3×3 , 4×4 in total. For each of the sets of applications, we randomly generate the number of applications and obtain the corresponding task graphs by TGFF [41]. The power consumption traces of every set of applications are generated by McPAT based on random voltage/frequency-level assignment and used as the input to the temperature models and HotSpot to obtain the respective estimated average temperature over all the processor cores, i.e., the average chip temperature. Figure 6 shows the comparison of the normalized steady-state chip average temperature predicted by different temperature models on different floorplans. The results from HotSpot simulator on each floorplan is always 1 in the figure. The results show that the average difference between the temperatures obtained from our model and HotSpot is less than 0.5% on the three different floorplans. While the average difference between our model and the model in [15] is as high as 46% on 2×2 mesh-based floorplan, 11.0% on 3×3 mesh-based floorplan, and 15.8% on 4×4 mesh-based floorplan. The results convincingly verify that the proposed temperature model is much more accurate than the existing technique.

Fig. 6: Accuracy comparison between different temperature models and HotSpot on different platforms.

Comparing with HotSpot simulator, our temperature model achieves high temperature prediction accuracy with reduced running time. We obtain the running time of our model and HotSpot when estimating the temperature of every set of synthetic applications. The computation time of the model in [15] is almost the same as our model's and is omitted for simplicity. Figure 7 shows the accumulated time for multiple

¹Our temperature model can adapt to different parameter settings corresponding to different chip physical properties.

runs of our temperature model and HotSpot. As discussed in Section V, the running time of our model can be reduced by over two orders of magnitude compared to HotSpot simulator-based technique. The execution time of our temperature model is less than 60 μ s. Such a big gap between the running time of our temperature model and that of HotSpot indicates that our model is of satisfactory efficiency.

Fig. 7: Efficiency comparison between our temperature model and HotSpot on different platforms.

An accurate temperature model is vital for application mapping and scheduling techniques. To demonstrate this, we conduct a large number of experiments on randomly generated synthetic applications. We generate 1000 sets of synthetic applications, and then, assign and schedule each of them to a 4×4 MPSoC. Our temperature model and the model in [15] are used separately to test the chip peak temperature for every case. Since the temperature obtained from HotSpot simulator is almost the same with that obtained from our model, we use the temperature obtained from our model as reference value for simplicity. A temperature threshold violation occurs when the real chip temperature exceeds the safe threshold. It may eventually cause the damage of the physical system. Underestimation of chip peak temperature that result from inaccurate temperature estimation techniques would increase the risk of threshold violations. As shown in Figure 8, 24.3% of temperature threshold violation cases resulted from inappropriate application mapping and scheduling solutions are avoided by applying our temperature model, which enhances the system reliability to some extent.

Due to the feature of high accuracy and efficiency, our model is applied in the proposed MILP-based application mapping and scheduling approach to estimate the temperature of every processor core and to determine the impact on its aging, and hence, its reliability.

B. Evaluation of the MILP-based Approach

We conduct extensive performance evaluations on both synthetic and real-world benchmarks to demonstrate the effectiveness of the proposed MILP-based application mapping and scheduling approach.

In this paper, the temperature threshold is set to 70°C, and the targeted chip lifetime is set to 10 years. We calibrate the constants in each failure equation to give a system MTTF of 30

Fig. 8: Temperature threshold violation resulted from inaccurate temperature model.

years at nominal temperature and we use the same constants for all the experiments. The parameters of the electromigration failure model are set as follows: the cross-sectional area of conductor $A_c = 6.4 \times 10^{-8} \text{ cm}^2$, the current density for the original voltage level $J = 1.5 \times 10^6 \text{ A/cm}^2$, and the activation energy $E_a = 0.48 \text{ eV}$ [1]. The parameters of the TDDB failure model are set as follows: $a = 78$, $b = -0.081$, $X = 0.759 \text{ ev}$, $Y = -66.8 \text{ ev/K}$ and $Z = -8.37 \text{ e} - 4 \text{ ev/K}$ [30].

To illustrate the effectiveness and performance of our MILP-based approach, a large number of experiments are performed by comparing our approach with a simulated annealing-based technique [1] and a heuristic algorithm *ThermalSched* [15]. Simulated annealing-based algorithm is an effective technique which can be devised to solve application mapping and scheduling problems. Many researchers devise SA-based algorithm to obtain the results effectively and efficiently [42].

The MILP model is solved by Gurobi v5.60 [43] with CVX 2.1 in the Matlab, which is designed to be the fastest, most powerful solver available for MILP problems. We set the running time limit of the MILP model to two hours and ten hours for the synthetic benchmarks and real-world benchmarks, respectively. Considering the heavy time overhead of this formal method, we can relax its optimality to obtain a sub-optimal yet highly acceptable application mapping and scheduling solution more efficiently. In this paper, a sub-optimal result is also generated from the MILP solver by setting the current relative MIP (Mixed Integer Programming) optimality gap to 5%. The gap is computed as $(ObjBound - ObjVal) / |ObjVal|$, where $ObjBound$ and $ObjVal$ are the MIP objective bound and incumbent solution objective, respectively. The Gurobi solver will terminate (with an optimal/sub-optimal result) when the relative gap between the lower and upper objective bound is less than the gap times the upper bound.

Both the simulated annealing algorithm and the *ThermalSched* algorithm are run in the Matlab. The parameters for the simulated annealing approach are set as follows: initial temperature = 100, cooling rate = 0.99, and ending temperature = 10^{-5} . The *ThermalSched* algorithm [15] is extended accordingly to be applied to the proposed problem. Specifically, the earliest start time and the latest start time of a given task are computed using a middle voltage level

Fig. 9: Comparison of different techniques on total energy consumption.

between the highest and lowest voltage levels since the latest start time of a task is set according to the deadline of that task. Using the middle voltage to calculate the latest start time makes the problem resilient. Furthermore, the first available processor core is selected to schedule the current ready task, since the processor cores have identical configurations. Then, a proper speed for running each task is identified such that all the requirements can be met and the total energy can be minimized.

We first apply the proposed MILP-based approach to sets of *synthetic benchmarks*. The synthetic benchmarks are generated by TGFF using the sample input files that come with the software package. The maximum in-degree and out-degree of graph nodes for the TGFF input files are set among interval [2, 9]. For the synthetic benchmarks, ten synthetic task graphs are generated for each benchmark category, beginning from task graphs each of whom contains 6 tasks, and increasing the task number by 3 for tasks in each benchmark category.

Figures 9a, 9b, and 9c compare the normalized energy consumption of the three techniques on the 2x2, 3x3, and 4x4 floorplans, respectively. The X-axis is the synthetic task graph sets with different number of tasks. The Y-axis shows the normalized energy consumption by each technique. The energy consumption is in unit of microjoule, μJ , in this paper. The optimal results generated by the MILP-based approach are normalized to be 1 and are set as the baseline. Several key observations can be derived from Figure 9. Firstly, for all task graph sets on every floorplan, our technique outperforms both simulated annealing algorithm and *ThermalSched* algorithm. Specifically, for 2x2 floorplan, 26.4% and 22.6% lower energy consumption are achieved on average compared to the *ThermalSched* algorithm and SA-based algorithm, respectively. For 3x3 floorplan, the *ThermalSched* algorithm and SA-based algorithm deviate from the optimal solution by 20.3% and 8.7% on average, respectively. For 4x4 floorplan, an average of 25.6% and 9.5% lower energy consumption are achieved compared to the *ThermalSched* algorithm and SA-based algorithm, respectively. Secondly, the sub-optimal results obtained by our model are very close to the optimal results: 1.43% more energy consumption on average compared to the optimal approach, which indicates that the sub-optimal results also

achieve significant energy saving compared to the other two algorithms. Finally, our approach is able to obtain the optimal results for the applications with 33 tasks on average, and it is scalable to as many as 51 tasks for 2x2 floorplans. This strongly verifies the scalability of the proposed technique.

Figure 10 shows the average running time of the three techniques on different floorplans within the predetermined limited time (two hours). As the problem is NP-hard, the running time of the MILP-based approach increases exponentially with the increasing sizes of task graphs and platforms. It is also highly related to the shape of the task graph according to our experimental results. There are some task graphs in the task graph sets that the MILP approach cannot obtain the optimal results for them within two hours. These task graphs are not included in Figure 10 since the running time of the MILP technique is beyond the predetermined limited time in the experiments. Comparing to the MILP technique, both the *ThermalSched* and SA-based algorithm take a short period of time to find the results. It takes less than 1 second for the *ThermalSched* to obtain a solution, and the running time of the SA-based algorithm finding a feasible solution ranges from less than 1 second to 17 seconds. *ThermalSched* heuristic algorithm is scalable and efficient, which can speedily find a solution. The SA-based algorithm performs stochastic search in the design space and delivers better results with increased search time. The MILP approach, as a formal method instead, can perform systematic exploration in the design space and provide a theoretically guaranteed optimal solution. In addition, for the full set of task graphs with 6 tasks, our MILP technique normally takes less than one second to obtain the optimal solution; for the full set of task graphs with 33 tasks, it still takes only 20 minutes on average. The sub-optimal results can be obtained within one hour for all the task graphs. In most cases, the time needed for finding a sub-optimal result is much lower than that for generating a globally optimal result which takes much more time to cover the entire design space for the provable optimality. As shown in Figure 9, the average difference between the optimal and sub-optimal results is small and the sub-optimal result outperforms the other two techniques outstandingly. This indicates that a sub-optimal solution is good alternative when finding the optimal

Fig. 10: Comparison of different techniques on running time.

solution experiences a long running time.

TABLE IV: Benchmark description.

Benchmark	α	ε	IN/OUT	Description
audiobeam	22	35	15/15	Audio beamformer
autocor	12	18	8/8	Produce auto-correlation series
fmradio	31	37	4/4	FM radio with equalizer
tde_pp	29	28	1/1	Time-delay equalization
h.264	51	71	11/12	Video compression standard
channelvocoder	55	70	17/17	Channel voice coder

¹ α and ε are the number of actors and edges in the benchmark, respectively.

² IN/OUT represents the maximum indegree/outdegree of the benchmark.

In order to test the practicability and scalability of the proposed approach to *real-world applications*, we conduct case studies on several different real-world benchmarks that are shown in Table IV. The benchmarks are collected from two sources. The first source is the StreamIt benchmark [44] which contributes 5 streaming applications (*audiobeam*, *autocor*, *fmradio*, *tde_pp*, *channelvocoder*). The actors execution times of the StreamIt benchmark are specified by its authors in clock cycles measured on MIT RAW architecture. The second source is the *H.264* HDTV decoder. In [45], the functional parallelization of *H.264* decoding is parsed into macro blocks (MB). We derive a prototype task graph with 51 task nodes from *H.264* decoder. WCETs of tasks are obtained by profiling. In total, 6 applications are considered.

The experiment results are shown in Table V. We can see that the optimal result obtained with MILP model reduces energy by 19.09% compared to SA-based algorithm, and 28.53% compared to *ThermalSched*. The time needed to obtain an optimal result spans from less than 1 minute (for *autocor* and *tde_pp*) to several hours. The optimal results cannot be obtained within 10 hours for *H.264* decoder under the 3×3 and 4×4 mesh-based MPSoC. For the suboptimal results obtained with MILP formulation, it is quite close to the optimal results. Specifically, an average of 16.29% and 27%

energy reduction is achieved compared to SA-based algorithm and *ThermalSched*, respectively. Moreover, the time needed for suboptimal results is much shorter than it for optimal results, ranging from less than 1 minute to at most 45 minutes. Note that the time needed for the MILP solver depends on a number of factors, and it cannot simply be measured by the size of the chip floorplan or the application. In the case of the “fmradio”, the time needed for the 3×3 mesh-based floorplan is much shorter than it needed for the 2×2 mesh-based floorplan. This is normal phenomenon in the case of finding optimal solution using MILP model.

In summary, our approach exhibits high effectiveness and good performance compared to the state-of-the-art techniques, which achieves energy minimization for real-time application mapping and scheduling at design time, with the safe chip temperature and system lifetime reliability constraints satisfied.

VIII. CONCLUSION

This paper has proposed a high-level energy optimization framework, in which a MILP formulation is proposed for mapping and scheduling real-time applications onto MPSoCs. Combining DVFS techniques, the MILP formulation minimizes the total energy consumption with both thermal threshold constraint and system lifetime reliability constraint satisfied. A lightweight temperature model has also been employed to efficiently yet accurately estimate the thermal distribution of MPSoCs in the MILP formulation. The accuracy and efficiency of the temperature model have been verified by comparing with existing temperature models. Its prediction accuracy is improved by as much as 46% than the existing temperature model, and an average of over 99% accuracy is provided with greatly reduced computation time comparing with the architecture-level HotSpot simulator. The effectiveness and scalability of the MILP-based technique have been further verified. We have conducted extensive experiments on a large number of synthetic and real-world applications on different floorplans. The results show that our approach reduces on average 19.09% and 28.53% total energy consumption in comparison with two state-of-the-art techniques, with the tem-

TABLE V: Energy consumption for different real-world applications obtained by using different techniques.

Floorplan	Benchmark	SA	ThermalSched	MILP sub-optimal				MILP optimal			
				Energy	Time	%(SA)	%(TS)	Energy	Time	%(SA)	%(TS)
2×2	audiobeam	4.54	4.63	3.63	5.86	20.04%	21.60%	3.58	247	21.15%	22.68%
	autocor	15.36	19.35	13.32	0.2	13.28%	31.16%	12.75	0.31	16.99%	34.11%
	fmradio	26.07	29.56	20.59	11.56	21.02%	30.35%	20.05	180.3	23.09%	32.17%
	tde_pp	2814.02	3322.4	2441.56	0.18	13.24%	26.51%	2394.2	0.25	14.92%	27.94%
	channelvocoder	7330.14	8825.56	6147.17	9.85	16.14%	30.35%	6025.87	259.76	17.79%	31.72%
	h.264	901	1059	807	6	10.43%	23.80%	800	356.4	11.21%	24.46%
3×3	audiobeam	5.35	5.85	4.26	10.1	20.37%	27.18%	4.07	156.12	23.93%	30.43%
	autocor	16.56	20.41	14.11	0.22	14.79%	30.87%	13.51	0.21	18.42%	33.81%
	fmradio	29.61	29.22	23.05	5.55	22.15%	21.12%	22.82	13.48	22.93%	21.90%
	tde_pp	3634.25	4256.8	3020.5	0.23	16.89%	29.04%	3004.6	0.62	17.33%	29.42%
	channelvocoder	7647.48	9215.28	6514.81	14.57	14.81%	29.30%	6479.91	472.34	15.27%	29.68%
	h.264	1044	1253	946	16.2	9.39%	24.50%	-	-	-	-
4×4	audiobeam	7.18	7.46	5.77	7	19.64%	22.65%	5.53	126.38	22.98%	25.87%
	autocor	21.03	23.44	18.14	0.55	13.74%	22.61%	17.52	0.56	16.69%	25.26%
	fmradio	37.01	38.89	27.07	7.72	26.86%	30.39%	26.9	268.7	27.32%	30.83%
	tde_pp	4107.37	4684.3	3405.8	0.22	17.08%	27.29%	3309.1	0.58	19.44%	29.36%
	channelvocoder	8478.63	9748.25	7463.28	14.79	11.98%	23.44%	7128.64	574.22	15.92%	26.87%
	h.264	1266	1696	1122	45	11.37%	33.84%	-	-	-	-
Average						16.29%	27.00%			19.09%	28.53%

¹ The energy consumption is in unit of microjoule, μJ .

² The time for MILP formulation is measured in minutes.

perature threshold and system lifetime reliability constraints satisfied.

ACKNOWLEDGMENT

This work is partially supported by NTU SUG M4082087.020, Singapore, and NSFC No. 61772094, Chongqing High-Tech Research Program cstc2017jcyjA1430, and the Fundamental Research Funds for the Central Universities No. 106112017CDJQJ188829, China.

REFERENCES

- [1] L. Huang, F. Yuan, and Q. Xu, "On task allocation and scheduling for lifetime extension of platform-based mpso designs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 2088–2099, 2011.
- [2] R. Viswanath, V. Wakharkar, A. Watwe, V. Lebonheur, M. Group, and I. Corp, "Thermal performance challenges from silicon to systems," *Intel Corporation Tech Rep*, no. 3, 2000.
- [3] S. S.-Y. Hsueh, R. H.-M. Huang, and C. H.-P. Wen, "Tasser: A temperature-aware statistical soft-error-rate analysis framework for combinational circuits," in *Quality Electronic Design (ISQED), 2014 15th International Symposium on*. IEEE, 2014, pp. 529–534.
- [4] M. Li, W. Liu, L. Yang, P. Chen, and C. Chen, "Chip temperature optimization for dark silicon many-core systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.
- [5] D. Atienza, P. G. Del Valle, G. Paci, F. Poletti, L. Benini, G. De Micheli, and J. M. Mendias, "A fast hw/sw fpga-based thermal emulation framework for multi-processor system-on-chip," in *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006, pp. 618–623.
- [6] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *ACM SIGARCH Computer Architecture News*, vol. 31, no. 2. ACM, 2003, pp. 2–13.
- [7] A. Dhodapkar, C. H. Lim, G. Cai, and W. R. Daasch, "Tem2p2est: A thermal enabled multi-model power/performance estimator," in *International Workshop on Power-Aware Computer Systems*. Springer, 2000, pp. 112–125.
- [8] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "Lifetime reliability: Toward an architectural solution," *IEEE Micro*, vol. 25, no. 3, pp. 70–80, 2005.
- [9] "International technology roadmap for semiconductors, 2011 edition: Process integration, devices, and structures." 2011.
- [10] L. Huang and Q. Xu, "Energy-efficient task allocation and scheduling for multi-mode mpsoes under lifetime reliability constraint," in *Proceedings of the conference on Design, automation and test in Europe*. European Design and Automation Association, 2010, pp. 1584–1589.
- [11] A. Das, A. Kumar, B. Veeravalli, C. Bolchini, and A. Miele, "Combined dvfs and mapping exploration for lifetime and soft-error susceptibility improvement in mpsoes," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. IEEE, 2014, pp. 1–6.
- [12] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by dvfs, task migration, and active cooling," *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 349–360, 2014.
- [13] M. Li, W. Liu, L. Yang, P. Chen, and C. Chen, "Chip temperature optimization for dark silicon many-core systems," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.
- [14] Y. Liu, H. Yang, R. P. Dick, H. Wang, and L. Shang, "Thermal vs energy optimization for dvfs-enabled processors in embedded systems," in *Quality Electronic Design, 2007. ISQED'07. 8th International Symposium on*. IEEE, 2007, pp. 204–209.
- [15] T. Chantem, X. S. Hu, and R. P. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on mpsoes," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 10, pp. 1884–1897, 2011.
- [16] N. Fisher, J.-J. Chen, S. Wang, and L. Thiele, "Thermal-aware global real-time scheduling on multicore systems," in *Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE*. IEEE, 2009, pp. 131–140.
- [17] H. Huang, V. Chaturvedi, G. Quan, J. Fan, and M. Qiu, "Throughput maximization for periodic real-time systems under the maximal temperature constraint," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 2s, p. 70, 2014.

- [18] V. Hanumaiah and S. Vrudhula, "Temperature-aware dvfs for hard real-time applications on multicore processors," *IEEE Transactions on Computers*, vol. 61, no. 10, pp. 1484–1494, 2012.
- [19] M. Li, J. Yi, W. Liu, W. Zhang, L. Yang, and E. H.-M. Sha, "An efficient technique for chip temperature optimization of multiprocessor systems in the dark silicon era," in *International Conference on High Performance Computing and Communications (HPCC)*. IEEE, 2015, pp. 688–693.
- [20] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, "Survey of energy-cognizant scheduling techniques," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1447–1464, 2013.
- [21] C. Ravishankar, S. Ananthanarayanan, S. Garg, and A. Kennings, "Analysis and evaluation of greedy thread swapping based dynamic power management for mpsoe platforms," in *Quality Electronic Design (ISQED), 2012 13th International Symposium on*. IEEE, 2012, pp. 617–624.
- [22] K. K. Rangan, G.-Y. Wei, and D. Brooks, "Thread motion: fine-grained power management for multi-core systems," in *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3. ACM, 2009, pp. 302–313.
- [23] M. Fleischmann, "Longrun power management," *White Paper of Transmeta Corporation*, 2001.
- [24] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *Design Automation Conference, 41st Conference on (DAC'04)*. ACM, 2004, pp. 275–280.
- [25] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*. ACM, 2002, pp. 721–725.
- [26] J. R. Black, "Electromigration: a brief survey and some recent results," *Electron Devices, IEEE Transactions on*, vol. 16, no. 4, pp. 338–347, 1969.
- [27] "Failure mechanisms and models for semiconductor devices," in *JEDEC Publication JEP122-C*, 2006.
- [28] A. Dasgupta and R. Karri, "Electromigration reliability enhancement via bus activity distribution," in *Design Automation Conference Proceedings 1996, 33rd, Jun 1996*, pp. 353–356.
- [29] F. Chen and M. Shinosky, "Addressing cu/low-dielectric tddb-reliability challenges for advanced cmos technologies," *Electron Devices, IEEE Transactions on*, vol. 56, no. 1, pp. 2–12, 2009.
- [30] E. Wu, J. Sune, W. Lai, E. Nowak, J. McKenna, A. Vayshenker, and D. Harmon, "Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides," *Solid-State Electronics*, vol. 46, no. 11, pp. 1787–1798, 2002.
- [31] S. Zafar, A. Kumar, E. Gusev, and E. Cartier, "Threshold voltage instabilities in high- κ gate dielectric stacks," *IEEE Transactions on Device & Materials Reliability*, vol. 5, no. 1, pp. 45–64, 2005.
- [32] Z. Lu, W. Huang, M. R. Stan, K. Skadron, and J. Lach, "Interconnect lifetime prediction for reliability-aware systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 159–172, 2007.
- [33] K.-C. Wu, M.-C. Lee, D. Marculescu, and S.-C. Chang, "Mitigating lifetime underestimation: A system-level approach considering temperature variations and correlations between failure mechanisms," in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2012, pp. 1269–1274.
- [34] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The impact of technology scaling on lifetime reliability," in *Dependable Systems and Networks, 2004 International Conference on*. IEEE, 2004, pp. 177–186.
- [35] Z. Gu, C. Zhu, L. Shang, and R. P. Dick, "Application-specific mpsoe reliability optimization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 5, pp. 603–608, 2008.
- [36] S. Murali, A. Mutapic, D. Atienza, R. Gupta, S. Boyd, and G. De Micheli, "Temperature-aware processor frequency assignment for mpsoes using convex optimization," in *Hardware/Software Codesign and System Synthesis (CODES+ ISSS), 2007 5th IEEE/ACM/IFIP International Conference on*. IEEE, 2007, pp. 111–116.
- [37] R. A. Serway and J. W. Jewett Jr, "Physics for scientists and engineers with modern physics, volume 2," 2008.
- [38] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 469–480.
- [39] A. Ziabari, J.-H. Park, E. K. Ardestani, J. Renau, S.-M. Kang, and A. Shakouri, "Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, pp. 2366–2379, 2014.

- [40] R. Rao, S. Vrudhula, and C. Chakrabarti, "Throughput of multi-core processors under thermal constraints," in *Proceedings of the 2007 international symposium on Low power electronics and design*. ACM, 2007, pp. 201–206.
- [41] R. P. Dick, D. L. Rhodes, and W. Wolf, "Tgff: task graphs for free," in *Proceedings of the 6th international workshop on Hardware/software codesign*. IEEE Computer Society, 1998, pp. 97–101.
- [42] J. Yi, Q. Zhang, Y. Tian, T. Wang, W. Liu, E. H.-M. Sha, and Q. Xu, "Approxmap: On task allocation and scheduling for resilient applications," in *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*. IEEE, 2016, pp. 318–323.
- [43] "Gurobi optimizer," <http://www.gurobi.com/>.
- [44] W. Thies and S. Amarasinghe, "An empirical characterization of stream programs and its implications for language and compiler design," in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '10. New York, NY, USA: ACM, 2010, pp. 365–376.
- [45] J. Chong, N. Satish, B. C. Catanzaro, K. Ravindran, and K. Keutzer, "Efficient parallelization of h. 264 decoding with macro block level scheduling," in *ICME*, 2007, pp. 1874–1877.

Dr. Weichen Liu (S'07-M'11) received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong.

He is an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include embedded and real-time systems, multiprocessor systems, and fault-tolerant systems. He has authored and co-authored over 70 research papers in peer-reviewed journals, conferences, and books.

Juan Yi received the B.E. degree from the School of Software Engineering in Chongqing University, Chongqing, China, in 2010 and Ph.D. degree in the Department of Computer Science at the same university in June 2016. She is now a senior engineer at Huawei Technologies CO., LTD.

Her current research interests include embedded and real-time systems, big data technology, and multicore/manycore system.

Mengquan Li received the B.E. degree from the School of Computer Science at Chongqing University, Chongqing, China, in 2015 and is currently pursuing the Ph.D. degree from the Department of Computer Science at Chongqing University, Chongqing, China.

Her current research interests include temperature modeling and optimization, optical network-on-chip optimization, and high-performance parallel computing.

Peng Chen received the B.E. degree from the School of Software Engineering in Chongqing University, Chongqing, in 2015 and is currently pursuing the Ph.D. degree from the Department of Computer Science at Chongqing University, Chongqing, China.

His current research interests include embedded and real-time systems, multicore scheduling.

Lei Yang received the B.E. degree from the School of Computer Science at Chongqing University, Chongqing, China, in 2013 and is currently pursuing the Ph.D. degree from the Department of Computer Science at Chongqing University, Chongqing, China.

Her current research interests include Network-on-chip optimization, multi-core architecture optimization and high-performance parallel computing.