TSP: Thermal Safe Power - Efficient power budgeting for Many-Core Systems in Dark Silicon

Santiago Pagani*, Heba Khdr*, Waqaas Munawar*, Jian-Jia Chen†, Muhammad Shafique*, Minming Li‡, and Jörg Henkel*

*Chair for Embedded Systems (CES)
Karlsruhe Institute of Technology (KIT), Germany

†Department of Informatics
TU Dortmund University, Germany

‡Department of Computer Science
City University of Hong Kong, China

Corresponding Author: santiago.pagani@kit.edu

ABSTRACT

Chip manufacturers provide the Thermal Design Power (TDP) for a specific chip. The cooling solution is designed to dissipate this power level. But because TDP is not necessarily the maximum power that can be applied, chips are operated with Dynamic Thermal Management (DTM) techniques. To avoid excessive triggers of DTM, usually, system designers also use TDP as power constraint. However, using a *single* and *constant* value as power constraint, e.g., TDP, can result in big performance losses in many-core systems. Having better power budgeting techniques is a major step towards dealing with the dark silicon problem.

This paper presents a new power budget concept, called Thermal Safe Power (TSP), which is an abstraction that provides safe power constraint values as a function of the number of simultaneously operating cores. Executing cores at any power consumption below TSP ensures that DTM is not triggered. TSP can be computed offline for the worst cases, or online for a particular mapping of cores. Our simulations show that using TSP as power constraint results in 50.5% and 14.2% higher average performance, compared to using *constant* power budgets (both per-chip and per-core) and a boosting technique, respectively. Moreover, TSP results in dark silicon estimations which are more optimistic than estimations using constant power budgets.

1. INTRODUCTION

For a specific chip, the common industry practice is to provide the system designers with the Thermal Design Power (TDP). According to [10], TDP "is the highest expected sustainable power while running known power intensive real applications" and it should be a safe power level in which to run the system. Hence, the cooling solution should be designed to dissipate TDP, such that running at this target power level does not cause any thermal problems. However, TDP is not the maximum achievable power. To avoid the system from possible overheating, chips are provided with Dynamic Thermal Management (DTM) techniques. DTM can power down cores, gate their clocks, reduce their supply voltage and frequency, boost-up the fan speed, etc. By directly measuring the temperature, if the system heats up above a certain threshold, DTM is triggered such that the temperature is reduced [10].

Usually, system designers also use TDP as a power constraint in order to avoid excessive triggers of DTM. However, using a *single* and *constant* value as a power constraint for each core or for the entire chip, e.g., TDP, can result in tremendous performance losses for many-core systems. Some other techniques, e.g., Intel's Turbo Boost [10, 19] and AMD's Turbo CORE [16], allow for power consumptions above the TDP constraint during *short time intervals*. Nevertheless, once the boost time interval expires, the system must return to power consumption values below TDP, and some time is needed before a new violation of the power constraint is allowed. A more efficient technique would be to safely increase the power consumption to values in which the system can remain *indefinitely*, without the need of returning to a state with less power consumption. The following example gives some insight in both of these subjects.

ESWEEK'14, October 12-17 2014, New Delhi, India
Copyright 2014 ACM 978-1-4503-3051-0/14/10 ...$15.00.
http://dx.doi.org/10.1145/2656075.2656103

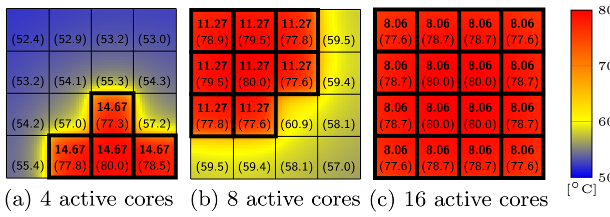


Figure 1: Example for a maximum temperature of 80°C. Top numbers are the power consumptions (in Watts) of each active core (boxed in black). Bottom numbers in parenthesis are the temperatures in the center of each core (in °C). Detailed temperatures are shown according to the color bar.

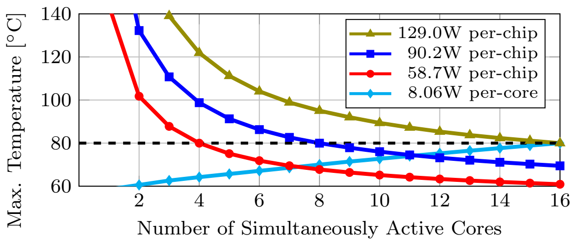


Figure 2: Maximum steady-state temperature (DTM deactivated) among all cores as a function of the number of active cores, for different per-chip and per-core power budgets.

Motivational Example: For simplicity of presentation, consider a hypothetical many-core system with 16 cores of size 2.31 × 2.31 mm[1], arranged in 4 rows and 4 columns. Assume a threshold temperature that triggers DTM of 80°C, and a cooling solution taken from the default configuration of HotSpot [9] (detailed in Section 8.1). To account for several *constant* power budgets, we consider four cases. Specifically, 58.7 Watts per-chip, 90.2 Watts per-chip, 129.0 Watts per-chip, and 8.06 Watts per-core.

Running simulations with HotSpot, Figure 1a shows the resulting steady-state temperatures on the chip when 4 cores

[1]Alpha 21264 core in 45nm, simulated with McPAT [14].

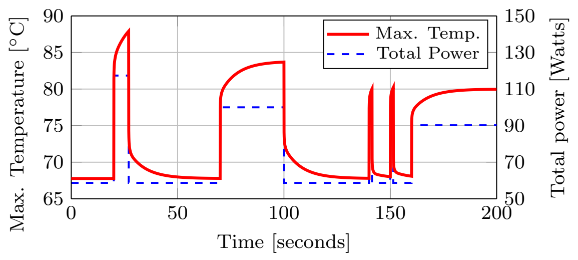


Figure 3: Motivational example for boosting techniques (the red line shows the maximum temperature among all cores).

consume 14.67 Watts each (58.7 Watts in total). Similarly, Figure 1b and Figure 1c show the resulting steady-state temperatures on the chip when 8 cores consume 11.27 Watts each (90.2 Watts in total) and when 16 cores consume 8.06 Watts each (129.0 Watts in total), respectively. For all cases, although different power budgets are considered, at least one core reaches 80°C in the steady-state. Furthermore, Figure 2 presents the maximum temperature among all cores (in the steady-state) as a function of the number of simultaneously active cores, in case that DTM is deactivated.

From Figures 1 and 2, it becomes clear that using a *single* and *constant* power budget can either be a pessimistic approach, or it can result in frequent triggers of DTM. For example, as seen in Figure 1c, when activating 16 cores each core could safely consume up to 8.06 Watts (129.0 Watts in total), which means that using a smaller power budget is pessimistic for such a case. On the other hand, for a power budget of 129.0 Watts, if the system consumes 129.0 Watts among any number of cores smaller than 16 the temperature on at least one core will exceed 80°C, triggering DTM. Moreover, Figure 2 shows that for this case using a power budget of 8.06 Watts per-core can be a good compromise, because the temperature never exceeds 80°C, but without being too far from it. However, there is still room for improvement, e.g., using different power budgets per-core depending on the number of active cores, such that the maximum steady-state temperature among all cores is 80°C for all cases. That is, according to Figure 1, a per-core power budget of 14.67 Watts when activating 4 cores, 11.27 Watts when activating 8 cores, 8.06 Watts when activating 16 cores, etc. *Such is the power budget concept which is presented in this paper.*

Regarding boosting technologies, consider the same 16 core system with DTM deactivated, and assume a given TDP of 58.7 Watts. Using HotSpot with its default configuration, Figure 3 presents the maximum temperature among all cores as a function of time, when there are 8 active cores according to Figure 1b, with different power consumptions for every time interval. During $t = [0s, 20s]$, $t = [27s, 70s]$, $t = [100s, 140s]$, $t = [141.4s, 150s]$, and $t = [151.4s, 160s]$, each one of these 8 active cores consumes 7.34 Watts, for a total power consumption equivalent to TDP. The maximum temperature in the steady-state among all cores during these intervals is 67.8°C, *as expected according to Figure 2*. At time $t = 20s$, the system is boosted by doubling the power in each core (117.4 Watts in total), resulting in an abrupt temperature increase, such that at least one core exceeds 80°C *almost instantly*. Similarly, at time $t = 70s$, the system is boosted by increasing the power of each active core to 12.5 Watts (100.0 Watts in total), such that the temperature of at least one core exceeds 80°C after *only 1.4 seconds*. Thus, a boosting technique would operate as shown during $t = [140s, 160s]$, where the system is boosted during 1.4 seconds to 12.5 Watts per core, and then returned to the original execution power of 7.34 Watts per core. Contrarily, as done during $t = [160s, 200s]$, the system is able to *indefinitely* consume 90.2 Watts (much more than TDP) by intelligently constraining the power boost to 11.27 Watts per core, as already seen in Figure 1b.

Objective: The objective of this paper is to present a new power budget concept, called Thermal Safe Power (TSP). TSP is an abstraction that provides safe power constraint values as a function of the number of simultaneously active cores. Executing cores at power consumptions below TSP results in maximum temperatures below the threshold level that triggers DTM. Based on the RC thermal network [9] of a specific chip, we focus on deriving a numerical method to compute TSP in an offline manner for the worst cases, as a function of the number of active cores. Moreover, the method should have polynomial-time complexity, such that TSP can also be computed online for a particular mapping of cores and ambient temperatures.

Our Contributions: Based on the above discussions,

- We present a polynomial-time algorithm that computes TSP for a particular mapping of active cores. This algorithm can be used online, thus accounting both for changes in the mapping decisions and in the ambient temperature.
- We present a polynomial-time algorithm to compute TSP for the worst-case mappings of active cores. That is, a mapping of active cores, for every possible number of simultaneously active cores, that results in the lowest TSP values. Considering such worst-case core mappings is the most pessimistic case, which results in TSP values that are safe for any other mapping scenario, and thus allows the system designers to abstract from core mapping decisions.

Open-Source Contributions: The algorithms to compute TSP are implemented as an open-source tool available for download at http://ces.itec.kit.edu/download.

Evaluations: Using gem5, McPAT, and HotSpot, we run simulations to compare the total system performance for using six different power constraints: TSP for given mappings, worst-case TSP, two constant power budgets per-chip, a constant power budget per-core, and boosting over a constant power budget per-chip [10, 19]. Our simulations show that using TSP for the worst-case mappings results in 50.5% and 14.2% higher average performance, compared to using constant power budgets and the selected boosting technique, respectively. We also show how TSP can be used to estimate the amount of dark silicon [3, 7, 20], and how such estimations are more optimistic than those for constant power budgets.

2. RELATED WORK

There are many works in the literature that focus on improving performance under a given (constant) power budget [12, 17, 18], including several that specifically use TDP as the power constraint [13, 15]. There is also work on energy efficiency [4] and on reliability [11] under TDP constraints, and on power budgeting based on reinforcement learning [6].

In [15], authors propose a control-based framework to obtain the optimal trade-off between power and performance for homogeneous multi-core systems under the TDP budget. The controllers throttle down the power if TDP is exceeded, and assign tasks onto cores to optimize the performance.

The work in [18] exploits process variations between cores in a homogeneous multi-core system to pick the more suitable cores for an application to improve its performance. Their results show that the performance efficiency can be increased along with the increase in the dark silicon area.

The work in [13] presents throughput analysis for the impact of applying per-core power gating and DVFS to thermal and power constrained multi-core processors, using TDP as the power constraint. The authors also exploit power and thermal headroom resulting from power-gated idle cores, allowing active cores to increase their frequency.

In [12], authors build a 32-core TFET-CMOS heterogeneous multi-core and propose a runtime scheme to improve the performance of applications running on such cores, operating under a fixed power budget. The scheme combines heterogeneous thread-to-core mapping, dynamic work partitioning, and dynamic power partitioning.

In [4], the authors evaluate the energy efficiency of different processors from Intel's i5 and i7 family, using selected benchmarks with variable core counts and vectorization techniques, while using TDP as the power constraint.

All of the above mentioned work uses a *single* and *constant* value as the power constraint to abstract from thermal problems. However, the motivational example in Section 1 shows that, depending on the amount of simultaneously active cores, different power consumptions result in the same maximum temperatures. This means that using a single value as the power constraint, e.g., TDP, can result in performance losses for multi- and many-core systems.

A few other technologies, like Intel's Turbo Boost [10, 19] and AMD's Turbo CORE [16], leverage this temperature headroom by increasing the power of some cores[2] in an online manner, favouring applications with very high workload. Such techniques violate the TDP constraint during *short intervals of time*[3] until reaching the threshold temperature that triggers DTM, to then return the system to power consumption values below TDP. This boosting is better suited after a period of low power consumption, achieving longer boost intervals for such cases. Unlike these techniques, in TSP the increases in power consumption are constrained (according to the number of cores) such that DTM is not activated, allowing the system to remain *indefinitely* in such power states, as shown in the motivational example.

The introduction of TSP as a new power budget concept may motivate researchers to revisit the related work, possibly resulting in extensions of the existing literature, that achieve a better system performance.

3. SYSTEM MODEL

3.1 Hardware Model

The computation of TSP is based on an RC thermal network modelled from the floorplan of any given architecture. The detail of the blocks that conform the floorplan can be freely chosen. For example, consider the 64 cores system presented in Figure 4, which is based on simulations conducted with gem5 [2] and McPAT [14] for *out-of-order* Alpha 21264 cores in 22 nm technology. Each core has an area of 9.6mm^2, and there is a shared L2 cache and a memory controller every 4 cores. The area of the L2 blocks together with the memory controllers is 4.7mm^2, which is comparable to the area of the cores. Therefore, it is reasonable to have independent blocks for the L2 cache and the memory controllers. For the rest of the chip, a practical approach is to consider blocks at a core level, and compute the TSP values for such a granularity, specifically, because the power consumptions of the internal blocks of the cores are tightly related with the execution frequency on each core.

For simplicity of presentation, throughout this paper we focus on a many-core system with M' cores. However, we consider that there are M blocks in the floorplan, such that $M - M'$ is the amount of blocks that correspond to other types of components, e.g., L2 caches and memory controllers. When computing TSP, we consider that each one of these $M - M'$ blocks always consumes its highest power.

We refer to a core being *active*, whenever the core is in its execution mode. To maintain a core active at its lowest speed, a minimum power consumption is needed, which we define as $P^{\text{core}}_{\text{min}}$. Contrarily, we refer to a core being *inactive*, when the core is in some low-power mode, e.g., sleep or turned off (power-gated). We define the power consumption of an inactive as $P^{\text{core}}_{\text{inact}}$. Moreover, when simultaneously activating m cores, there are $\binom{M'}{m}$ combinations of *which* specific cores in the floorplan to activate. Throughout the paper, we refer to a specific decision of *which* cores to activate as *core mapping* or *mapping of cores*.

[2]By increasing the voltage and frequency of the cores.

[3]Typically, in the range of 10s of seconds, up to 60 seconds.

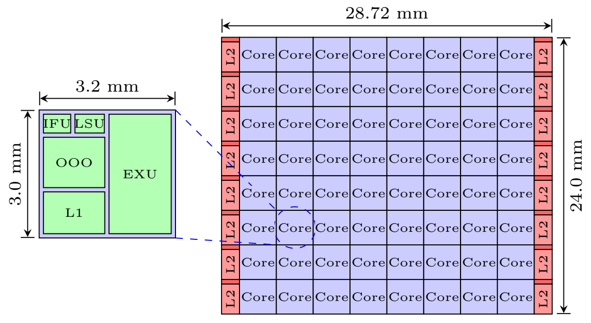


Figure 4: Floorplan for a 64 core system based on simulations in McPAT [14]. In McPAT, cores are composed by several units: an instruction fetch unit (IFU), an execution unit (EXU), a load and store unit (LSU), an out-of-order (OOO) issue/dispatch, and a private L1 cache.

Aside of having a power constraint as an abstraction from thermal problems, there can also exist a maximum chip power consumption that cannot be exceeded, which we denote P_{max}. This maximum power is not an abstraction, but an actual electrical constraint, e.g., from the power supply.

The power consumption in a CMOS core is the summation of its dynamic power consumption (mainly generated by switching activities) and its static power consumption (mainly generated by leakage currents). Part of the static power consumption depends on the temperature of the core, which means that higher temperatures cause higher power consumptions. In order to consider *safe margins for offline power profiles* of tasks and their scheduling decisions for safe operation, we assume that the power consumptions of cores are modelled at temperatures near the threshold level that triggers DTM, which we define as T_{DTM}. That is, when experimentally measuring the power consumption of a core executing a task at a specific voltage and frequency, this is done at temperatures near T_{DTM}. If not, we may have underestimated power models of tasks. When making online decisions, if cores are equipped with power meters, then no over- or underestimation occurs, because the system can measure the actual power consumption of each task (including leakage effects), and act accordingly.

3.2 Thermal Model

For our thermal model, we consider the well-known duality between thermal and electrical circuits, i.e., we use an RC thermal network [9]. Such a network is composed by N thermal nodes, where $N \geq M$. In an RC thermal network, thermal nodes are interconnected between each other through thermal conductances. Each thermal node also has a thermal capacitance associated to it, which accounts for the transient temperatures. The ambient temperature, denoted as T_{amb}, is considered to be constant, and thus there is no capacitance associated with it. The power consumptions of cores and other blocks correspond to heat sources. With these considerations, the temperature of every thermal node is a function of its power consumption, the temperatures of the neighbouring nodes, and the ambient temperature. Hence, for any RC thermal network with N thermal nodes, we can build a system of N differential equations associated with it, which can be expressed as

$$\mathbf{A}\mathbf{T}' + \mathbf{B}\mathbf{T} = \mathbf{P} + T_{\text{amb}}\mathbf{G},$$

where matrix $\mathbf{A} = [a_{i,j}]_{N\times N}$ contains the thermal capacitance values, matrix $\mathbf{B} = [b_{i,j}]_{N\times N}$ contains the thermal conductance values in $\left[\frac{Watt}{Kelvin}\right]$, column vector $\mathbf{T} = [T_i]_{N\times 1}$ represents the temperature on each node, column vector $\mathbf{T}' = [T_i']_{N\times 1}$ accounts for the first order derivative of the temperature on each node with respect to time, column vector $\mathbf{P} = [p_i]_{N\times 1}$ contains the power consumption on each node, and column vector $\mathbf{G} = [g_i]_{N\times 1}$ contains the thermal conductance between each node and the ambient. If node i is not in contact with the ambient temperature, e.g., the temperature of a core or an internal node, the value of g_i is set to zero. The thermal conductance values in matrix $\mathbf{B}$ include the thermal conductances between vertical and lateral neighbouring nodes. Matrix $\mathbf{A}$ is generally a diagonal matrix, since thermal capacitances are modelled to ground.

For specific floorplans, the values for matrices and vector $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}$ can be computed through HotSpot [9], and these are the same matrices used internally by HotSpot.

When only considering the *steady-state*, we have that

$$\mathbf{B}\mathbf{T} = \mathbf{P} + T_{\text{amb}}\mathbf{G} \qquad \text{or} \qquad \mathbf{T} = \mathbf{B}^{-1}\mathbf{P} + T_{\text{amb}}\mathbf{B}^{-1}\mathbf{G},$$

where from $\mathbf{B}^{-1}$ we have that $b^{\text{-}1}{}_{i,j} \cdot p_j$ represents the amount of heat contributed by node j into the steady-state temperature of node i, i.e., T_i.

Regarding vector $\mathbf{P}$, we can divide it into three sub-vectors: $\mathbf{P}^{\text{cores}}$ for the power consumption on the cores; $\mathbf{P}^{\text{blocks}}$ for the power consumption on blocks of a different type, which we consider to be always active at their highest power consumption values, e.g., the L2 caches for the many-core system in Figure 4 as explained in Section 3.1; and $\mathbf{P}^{\text{int}}$ for internal nodes in which $p_i^{\text{int}} = 0$ for all i. It holds that $\mathbf{P} = \mathbf{P}^{\text{cores}} + \mathbf{P}^{\text{blocks}} + \mathbf{P}^{\text{int}}$. Decomposing $\mathbf{P}$, we have that

$$\mathbf{T} = \mathbf{B}^{-1}\mathbf{P}^{\text{cores}} + \mathbf{B}^{-1}\mathbf{P}^{\text{blocks}} + T_{\text{amb}}\mathbf{B}^{-1}\mathbf{G}, \qquad (1)$$

where by only focusing on one equation from the system of equations, the steady-state temperature on node i is

$$T_i = \sum_{j=1}^{N} b^{\text{-}1}{}_{i,j} \cdot p_j^{\text{cores}} + \sum_{j=1}^{N} b^{\text{-}1}{}_{i,j} \left(p_j^{\text{blocks}} + T_{\text{amb}} \cdot g_j\right). \quad (2)$$

For notational brevity, we define set $\mathbf{K} = \{k_1, k_2, \ldots, k_M\}$, such that the elements in $\mathbf{K}$ include all indices of the thermal nodes that correspond to blocks of the floorplan; as opposed to thermal nodes that represent the heat sink, internal nodes of the heat spreader, the thermal interface material, etc. Similarly, we define set $\mathbf{K}' = \{k_1', k_2', \ldots, k_{M'}'\}$, such that $\mathbf{K}'$ contains all indices of nodes that correspond to cores.

Furthermore, we also define column vector $\mathbf{Q} = [q_i]_{N\times 1}$ for a particular mapping of *active* cores. Vector $\mathbf{Q}$ is a binary vector: $q_i = 1$ means that node i corresponds to an *active* core; $q_i = 0$ means that node i corresponds to an *inactive* core, or that node i is not a core at all.

4. PROBLEM DEFINITION

This paper presents a new power budget concept, called Thermal Safe Power (TSP). TSP is an abstraction that provides power constraint values as a function of the number of simultaneously active cores, according to the definition of *active/inactive* from Section 3.1. The values of TSP vary according to the floorplan and which cores are simultaneously active. Some specific core mappings result in the lowest TSP values, and we define such core mappings as the *worst-case mappings*. Executing cores at power consumptions below TSP, for the corresponding mapping and number of active cores, results in maximum temperatures below T_{DTM}.

For a specific chip and its corresponding RC thermal network, the *first objective* of this paper is to provide a numerical method to compute TSP for a given mapping of cores. The mapping of cores is typically determined by an operating system/run-time system or by an offline system software. This method should have polynomial-time complexity, such that TSP can be computed online for a particular mapping of cores and ambient temperatures. Formally, for a given core mapping $\mathbf{Q}$, this means obtaining a uniform power constraint for the active cores, defined as $P_{\text{TSP}}(\mathbf{Q})$, such that $T_i \leq T_{\text{DTM}}$ for all $i \in \mathbf{K}$.

The *second objective* is to derive an algorithm to compute the most pessimistic TSP values for a given number of simultaneously active cores, i.e., for the worst-case mappings. Such TSP values can be used as safe power constraints for any possible mapping of cores, thus allowing the system designers to abstract from core mapping decisions. Formally, this means obtaining the most pessimistic uniform power constraint for any m active cores, defined as $P_{\text{TSP}}^{\text{worst}}(m)$, such that $T_i \leq T_{\text{DTM}}$ for all $i \in \mathbf{K}$.

The algorithms presented in Section 5 and Section 6 are derived considering the steady-state. Section 7 explains a method to further consider the transient temperatures.

5. TSP FOR A GIVEN CORE MAPPING

5.1 Different Power Constraints per Core

This subsection presents a solution to compute TSP for a particular core mapping and ambient temperature, which results in different TSP values per core. That is, cores are constrained to different power values, depending on their location and the adjacent active cores. The objective here is to maximize the total power consumption.

For a given $\mathbf{Q}$, $\mathbf{P}^{\text{blocks}}$, T_{amb}, T_{DTM}, $P_{\text{inact}}^{\text{core}}$, $P_{\max}$, and floorplan, the TSP computation for this case can be formulated as a linear programming. Note that this is not the main contribution of our paper, and we only present this solution for completeness. This formulation is expressed as

$$\text{Maximize} \quad \sum_{i=1}^{N} p_i^{\text{cores}}$$

such that:

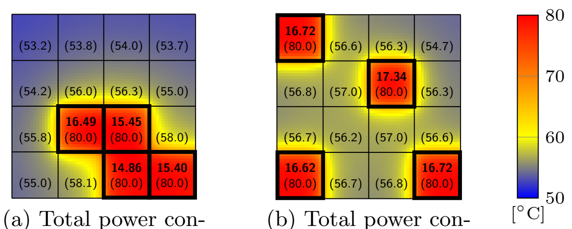
$$\begin{aligned}
&\mathbf{B}\mathbf{T} - \mathbf{P}^{\text{cores}} = \mathbf{P}^{\text{blocks}} + T_{\text{amb}}\mathbf{G} && \\
&\textstyle\sum_{i=1}^{N} p_i^{\text{cores}} \leq P_{\max} - \sum_{i=1}^{N} p_i^{\text{blocks}} && \\
&T_i \leq T_{\text{DTM}} && \text{for all } i \in \mathbf{K} \\
&T_i \geq 0 && \text{for all } i = 1, 2, \ldots, N \\
&p_i^{\text{cores}} = 0 && \text{for all } i \notin \mathbf{K}' \\
&p_i^{\text{cores}} = P_{\text{inact}}^{\text{core}} && \text{for all } i \in \mathbf{K}' \text{ and } q_i = 0 \\
&p_i^{\text{cores}} \geq P_{\min}^{\text{core}} && \text{for all } i \text{ in which } q_i = 1.
\end{aligned}$$

Any standard method, e.g., Simplex or an Interior-Point method [5], can be used to solve this programming, resulting in a vector with a different power constraint for each active core in vector $\mathbf{Q}$, which we define as $\mathbf{P}_{\text{TSP}}^{\text{cores}}(\mathbf{Q})$.

Due to the heat transfer among cores, for the same number of active cores different mappings result in different power constraints. This can be seen in the following example. Consider a many-core system of 16 cores with the same settings as in the motivational example from Section 1, with $P_{\text{inact}}^{\text{core}}$ of 0 Watts and $P_{\min}^{\text{core}}$ of 0.5 Watts. For a T_{amb} of 45°C and two different mappings $\mathbf{Q}$, both with 4 active cores, we compute $\mathbf{P}_{\text{TSP}}^{\text{cores}}(\mathbf{Q})$ and present the results in Figures 5a and 5b. The maximum total power consumption for the mapping in Figure 5a and 5b is 62.2 Watts and 67.4 Watts, respectively.

5.2 Uniform Power Constraint for all Cores

This subsection presents a polynomial-time algorithm to compute TSP in an online manner for a particular core mapping and ambient temperature, which results in a uniform



(a) Total power consumption of 62.2 Watts

(b) Total power consumption of 67.4 Watts

Figure 5: Example of TSP with different power constraints. Top numbers are the power consumptions (in Watts) of each active core (boxed in black). Bottom numbers in parenthesis are the temperatures in the center of each core (in °C). Detailed temperatures are shown according to the color bar.

value of TSP per-core, for all active cores in $\mathbf{Q}$. That is, one power constraint value for each active core in the specified mapping, that results in a maximum temperature (in the steady-state) among all cores which does not exceed T_{DTM}. We define such a power constraint as $P_{\mathrm{TSP}}(\mathbf{Q})$. *Note that this does not mean that all active cores consume the same power, as this would be an unrealistic assumption. In fact, this means that each active core can consume any amount of power, which can be different for each core, as long as the power consumption of each core is no more than $P_{TSP}(\mathbf{Q})$.*

The algorithm that computes $P_{\mathrm{TSP}}(\mathbf{Q})$ is presented in Theorem 1, which is based on Lemma 1. Lemma 1 derives a uniform power constraint for all active cores in mapping $\mathbf{Q}$ such that the maximum temperature in the steady-state among all blocks does not exceed T_{DTM}, by taking into account the floorplan, the power consumption on other blocks, the ambient temperature, and the power consumption of inactive cores. This power constraint is defined as $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ and it does not take into consideration the maximum chip power $P_{\max}$. Theorem 1 verifies that the maximum power $P_{\max}$ is not violated. We define auxiliary function $\mathrm{R}(m)$ as

$$\mathrm{R}(m) = P^{\mathrm{core}}_{\mathrm{inact}} + \frac{P_{\max} - \sum_{i=1}^{N} p_i^{\mathrm{blocks}} - P^{\mathrm{core}}_{\mathrm{inact}} M'}{m}, \tag{3}$$

where m represents the number of active cores. If consuming $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ in all active cores violates $P_{\max}$, then Theorem 1 sets $P_{\mathrm{TSP}}(\mathbf{Q})$ to a smaller value, specifically, to $\mathrm{R}\left(\sum_{i=1}^{N} q_i\right)$, such that $P_{\max}$ is satisfied. Moreover, the pseudo-code for computing $P_{\mathrm{TSP}}(\mathbf{Q})$ is presented in Algorithm 1.

LEMMA 1. *For a given $\mathbf{Q}$, $\mathbf{P}^{blocks}$, T_{amb}, T_{DTM}, P^{core}_{inact}, and floorplan, we define $P^{\star}_{TSP}(\mathbf{Q})$ as a power constraint for each active core in the specified mapping, that results in a maximum temperature (in the steady-state) among all cores which does not exceed T_{DTM}. $P^{\star}_{TSP}(\mathbf{Q})$ is computed as*

$$P^{\star}_{TSP}(\mathbf{Q}) = \min_{\forall i \in \mathbf{K}} \left\{ \frac{T_{DTM} - P^{core}_{inact} \cdot \sum_{\forall j \in \mathbf{K}'} b^{-1}{}_{i,j}(1-q_j)}{\sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j} + \frac{-\sum_{j=1}^{N} b^{-1}{}_{i,j}\left(p_j^{blocks} + T_{amb} \cdot g_j\right)}{\sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j} \right\}. \tag{4}$$

PROOF. Considering vector $\mathbf{Q}$, and set $\mathbf{K}'$, if all inactive cores consume $P^{\mathrm{core}}_{\mathrm{inact}}$ and all active cores consume equal power P_{equal} at a given time, we have that for this time instant, Equation (2) can be rewritten as

$$T_i = P_{\mathrm{equal}} \cdot \sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j + P^{\mathrm{core}}_{\mathrm{inact}} \cdot \sum_{\forall j \in \mathbf{K}'} b^{-1}{}_{i,j}(1-q_j) + \sum_{j=1}^{N} b^{-1}{}_{i,j}\left(p_j^{\mathrm{blocks}} + T_{\mathrm{amb}} \cdot g_j\right). \tag{5}$$

For a given $\mathbf{Q}$, $\mathbf{P}^{\mathrm{blocks}}$, T_{amb}, and floorplan, the only variables in Equation (5) are T_i and P_{equal}. By setting T_i to T_{DTM}, we can then compute, for every i that belongs in $\mathbf{K}$, what value of P_{equal} would make T_i reach T_{DTM}. The most pessimistic value of P_{equal} is a safe power constraint for all cores in mapping $\mathbf{Q}$, and is therefore the resulting $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ for the given $\mathbf{P}^{\mathrm{blocks}}$, T_{amb}, and floorplan, which is expressed in Equation (4). The value of i that results in $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ corresponds to the block with the highest temperature for such a case. Thus, the lemma is proven. □

Algorithm 1 Uniform TSP for a given mapping

```
Input: $\mathbf{Q}$, $\mathbf{P}^{\mathrm{blocks}}$, $T_{\mathrm{amb}}$, $T_{\mathrm{DTM}}$, $P^{\mathrm{core}}_{\mathrm{inact}}$, $P_{\max}$, and floorplan;
Output: Uniform TSP power constraint for mapping $\mathbf{Q}$;
    {First compute $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ according to Equation (4)}
 1: $P^{\star}_{\mathrm{TSP}}(\mathbf{Q}) \leftarrow \infty$;
 2: for all $i \in \mathbf{K}$ do
 3:     $auxP \leftarrow T_{\mathrm{DTM}} - P^{\mathrm{core}}_{\mathrm{inact}} \cdot \sum_{\forall j \in \mathbf{K}'} b^{-1}{}_{i,j}(1-q_j)$;
 4:     $auxP \leftarrow auxP - \sum_{j=1}^{N} b^{-1}{}_{i,j}\left(p_j^{\mathrm{blocks}} + T_{\mathrm{amb}} \cdot g_j\right)$;
 5:     $auxP \leftarrow \frac{auxP}{\sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j}$;
 6:     if $auxP < P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ then
 7:         $P^{\star}_{\mathrm{TSP}}(\mathbf{Q}) \leftarrow auxP$;
 8:     end if
 9: end for
    {Then compute $P_{\mathrm{TSP}}(\mathbf{Q})$ according to Equation (6)}
10: if $P^{\star}_{\mathrm{TSP}}(\mathbf{Q}) \leq \mathrm{R}\left(\sum_{i=1}^{N} q_i\right)$ then
11:     $P_{\mathrm{TSP}}(\mathbf{Q}) \leftarrow P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$;
12: else
13:     $P_{\mathrm{TSP}}(\mathbf{Q}) \leftarrow \mathrm{R}\left(\sum_{i=1}^{N} q_i\right)$;
14: end if
15: return $P_{\mathrm{TSP}}(\mathbf{Q})$;
```

THEOREM 1. *For a given $\mathbf{Q}$, $\mathbf{P}^{blocks}$, T_{amb}, T_{DTM}, P^{core}_{inact}, P_{max}, and floorplan, we define $P_{TSP}(\mathbf{Q})$ as a power constraint for each active core in the specified mapping, that results in a maximum temperature (in the steady-state) among all cores which does not exceed T_{DTM}, and that will not violate P_{max}. The value of $P_{TSP}(\mathbf{Q})$ is computed as*

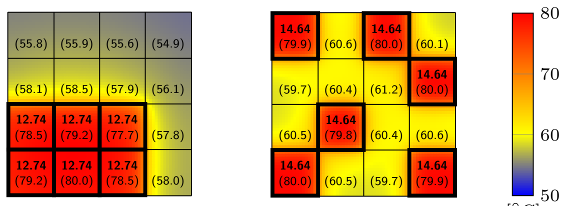
$$P_{TSP}(\mathbf{Q}) = \begin{cases} P^{\star}_{TSP}(\mathbf{Q}) & \text{if } P^{\star}_{TSP}(\mathbf{Q}) \leq R\left(\sum_{i=1}^{N} q_i\right) \\ R\left(\sum_{i=1}^{N} q_i\right) & \text{otherwise.} \end{cases} \tag{6}$$

PROOF. The first case is based on Lemma 1. As for the second case, the summation of the elements in $\mathbf{Q}$ is equal to the number of active cores in the mapping. Thus, it should hold that $P_{\mathrm{TSP}}(\mathbf{Q}) \cdot \sum_{i=1}^{N} q_i + P^{\mathrm{core}}_{\mathrm{inact}}\left(M' - \sum_{i=1}^{N} q_i\right) \leq P_{\max}$. If $P^{\star}_{\mathrm{TSP}}(\mathbf{Q})$ is larger than $\mathrm{R}\left(\sum_{i=1}^{N} q_i\right)$, then we simply set $P_{\mathrm{TSP}}(\mathbf{Q})$ to $\mathrm{R}\left(\sum_{i=1}^{N} q_i\right)$, such that $P_{\max}$ is not violated. Thus, the theorem is proven. □

The total time complexity for computing $P_{\mathrm{TSP}}(\mathbf{Q})$ for a given $\mathbf{P}^{\mathrm{blocks}}$, T_{amb}, T_{DTM}, $P_{\max}$, and floorplan, is $O(MN)$.

6. TSP FOR WORST-CASE MAPPINGS

This section presents a polynomial-time algorithm to compute TSP for the worst-case core mappings, for m active cores. The algorithm results in a uniform value of TSP per-core for all active cores. That is, one power constraint value for each active core in *any* possible core mapping with m simultaneously active cores, that results in a maximum temperature (in the steady-state) among all cores which does not exceed T_{DTM}. We define such a power constraint as $P^{\mathrm{worst}}_{\mathrm{TSP}}(m)$. *As in Section 5.2, note that this does not mean that all active cores consume the same power, as this would be an unrealistic assumption. It means that each active core, for any given mapping of m active cores, can consume any amount of power, which can be different for each core, as long as the power consumption of each core is no more than $P^{worst}_{TSP}(m)$.* The purpose for doing this is to allow system



(a) Worst-case for 6 cores (b) Best-case for 6 cores

Figure 6: Example of worst-case and best-case mappings. Top numbers are the power consumptions (in Watts) of each active core (boxed in black). Bottom numbers in parenthesis are the temperatures in the center of each core (in °C). Detailed temperatures are shown according to the color bar.

designers to abstract themselves from mapping decisions, as opposed to the TSP computations from Section 5.

Due to the heat transfer among cores, the mapping of cores, i.e., *which* cores are active and *which* cores are inactive, plays a major role in the computation of the maximum temperatures. This can be seen in the following example. Consider a many-core system of 16 cores with the same settings as in the motivational example from Section 1. Figure 6 shows two possible mappings when simultaneously activating 6 cores. For Figure 6a, the maximum temperature among all cores reaches 80°C when each core consumes 12.74 Watts. However for Figure 6b, this happens when each core consumes 14.64 Watts. We have a total of 11.4 Watts difference between these two core mappings, but with the same maximum temperature.

According to Equation (5), if all the active cores in the system run at the same power consumption, one or more of the active cores will heat up the most with respect to the rest of the cores. For the same value of P_{equal}, different $\mathbf{Q}$ mappings result in different maximum temperatures. Dually, for the same maximum temperature among all cores, different mappings will result in different power consumption values to produce such a temperature.

Generally, as shown in Figure 6, for the same maximum temperature values with $p_i^{\text{blocks}} = 0$ for all i, activating cores together in a corner of the chip results in lower P_{equal} values, compared to dispersing them throughout the chip. *This happens because* active *cores have higher chances of transferring heat to* inactive *cores when they are dispersed.* The worst-case mappings for TSP are those that produce the lowest power constraints, while no block in the floorplan exceeds (in the steady-state) the threshold temperature that triggers DTM, i.e., $T_i \leq T_{\text{DTM}}$ for all $i \in \mathbf{K}$.

By computing TSP for such cases, system designers can abstract themselves from core mapping decisions. This happens because such worst-case core mappings are the most pessimistic cases. When having m active cores, executing cores at power consumptions below $P_{\text{TSP}}^{\text{worst}}(m)$ will result in maximum temperatures (in the steady-state), among all blocks in the floorplan, below the threshold level that triggers DTM, for any possible mapping of m active cores.

For a given m, our algorithm is presented in Theorem 2, which is based on auxiliary matrix $\mathbf{H}$, Lemma 2, Lemma 3, and Lemma 4. Lemma 4 derives the worst-case uniform power constraint for m active cores such that the maximum temperature in the steady-state among all blocks does not exceed T_{DTM}, by taking into account the floorplan, the power consumption on other blocks, the ambient temperature, and the power consumption of inactive cores. This power constraint is defined as $P_{\text{TSP}}^{\star\text{worst}}(m)$ and it does not take into consideration the maximum power $P_{\max}$. Theorem 2 verifies that the maximum power $P_{\max}$ is not violated. If consuming $P_{\text{TSP}}^{\star\text{worst}}(m)$ in m cores violates $P_{\max}$, then we set $P_{\text{TSP}}^{\text{worst}}(m)$ to a smaller value such that $P_{\max}$ is satisfied.

Auxiliary matrix $\mathbf{H} = [h_{i,j}]_{M \times M'}$ is used in Lemma 4 to compute the maximum amount of heat that any m cores can contribute to the temperature on node i. Matrix $\mathbf{H}$ is built by making a partial copy of matrix $\mathbf{B}^{-1}$, such that $h_{i,j} = b^{-1}{}_{k_i,k'_j}$ for all $i = 1, 2, \ldots, M$ and for all $j = 1, 2, \ldots, M'$, and then reordering each row of $\mathbf{H}$ in a decreasing manner. Matrix $\mathbf{H}$ needs to be build and ordered only one time, which has a time complexity $O(MM' \log M')$. Lemma 2 and Lemma 3 prove that using matrix $\mathbf{H} = [h_{i,j}]_{M \times M'}$ results in the most pessimistic power constraints.

Lemma 2. *If all active cores consume equal power P_{equal}, the m highest values for $P_{equal} \cdot b^{-1}{}_{i,j}$ such that $j \in \mathbf{K}'$, correspond to the amount of heat contributed to temperature T_i by the m cores that contribute more heat into T_i.*

Proof. From the definition of matrix $\mathbf{B}^{-1}$, we know that $b^{-1}{}_{i,j} \cdot p_j$ represents the heat contributed by node j into the steady-state temperature of node i, i.e., T_i. Particularly, if we focus on $j \in \mathbf{K}'$, we are referring to the heat contributed by each core j into the steady-state temperature T_i.

From Equation (5), we know that there is one or more mappings $\mathbf{Q}$ that result in the maximum T_i for a given P_{equal}. Given that $\mathbf{B}^{-1}$, $\mathbf{G}$, $\mathbf{P}^{\text{blocks}}$, and T_{amb} are constant, it is clear that, for a given P_{equal}, T_i is maximized when $\sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j$ is also maximized. Moreover, it holds that the summation of the elements in $\mathbf{Q}$ is equal to the number of active cores in the mapping, that is, $\sum_{j=1}^{N} q_j = m$. Hence, for row i, we have that $\sum_{j=1}^{N} b^{-1}{}_{i,j} \cdot q_j$ is maximized when mapping $\mathbf{Q}$ activates the m cores with the highest $b^{-1}{}_{i,j}$, for row i, such that $j \in \mathbf{K}'$. Thus, the lemma is proven. □

Lemma 3. *If all active cores consume equal power P_{equal}, multiplying the power consumption on each core with the summation of the first m elements in row i of auxiliary matrix $\mathbf{H}$, that is, computing $P_{equal} \cdot \sum_{j=1}^{m} h_{i,j}$, results in the maximum amount of heat that any m cores can contribute to the steady-state temperature on node k_i, that is, T_{k_i}.*

Proof. Based on the definition of auxiliary matrix $\mathbf{H}$ and Lemma 2, the first m elements in row i of matrix $\mathbf{H}$ correspond to the m highest $b^{-1}{}_{k_i,j}$ values, for node k_i, such that $j \in \mathbf{K}'$. In turn, multiplied by P_{equal}, this is the amount of heat contributed to temperature T_i by the m cores that contribute more heat into T_i. Thus, the lemma is proven. □

Lemma 4. *For a given $\mathbf{P}^{blocks}$, T_{amb}, T_{DTM}, P_{inact}^{core}, and floorplan, we define $P_{TSP}^{\star worst}(m)$ as a power constraint for each active core in* any *possible core mapping with m simultaneously active cores, that results in a maximum temperature (in the steady-state) among all cores which does not exceed T_{DTM}. The value of $P_{TSP}^{\star worst}(m)$ is computed as*

$$P_{TSP}^{\star worst}(m) = \min_{1 \leq i \leq M} \left\{ \frac{T_{DTM} - P_{inact}^{core} \cdot \sum_{j=m+1}^{M'} h_{i,j}}{\sum_{j=1}^{m} h_{i,j}} \right.$$
$$\left. + \frac{-\sum_{j=1}^{N} b^{-1}{}_{k_i,j} \left(p_j^{blocks} + T_{amb} \cdot g_j\right)}{\sum_{j=1}^{m} h_{i,j}} \right\}. \tag{7}$$

Proof. Considering matrix $\mathbf{H}$ and Lemma 3, Equation (5) becomes

$$T_{k_i} \leq P_{\text{equal}} \cdot \sum_{j=1}^{m} h_{i,j} + P_{\text{inact}}^{\text{core}} \cdot \sum_{j=m+1}^{M'} h_{i,j}$$
$$+ \sum_{j=1}^{N} b^{-1}{}_{k_i,j} \left(p_j^{\text{blocks}} + T_{\text{amb}} \cdot g_j\right).$$

Similar to Lemma 1, by setting T_{k_i} to T_{DTM}, we can compute, for every $i = 1, 2, \ldots, M$, what value of P_{equal} would make T_{k_i} reach T_{DTM}. The most pessimistic value of P_{equal} is a safe power constraint for the m active cores, and is therefore the resulting $P_{\text{TSP}}^{\star\text{worst}}(m)$ for the given $\mathbf{P}^{\text{blocks}}$, T_{amb}, and floorplan, which is expressed in Equation (7). The value of i that results in $P_{\text{TSP}}^{\star\text{worst}}(m)$ corresponds to the block with the highest temperature in the worst-case mapping. Thus, the lemma is proven. □

THEOREM 2. *For a given* $\mathbf{P}^{blocks}$, T_{amb}, T_{DTM}, P_{inact}^{core}, P_{max}, *and floorplan, we define* $P_{TSP}^{worst}(m)$ *as a power constraint for each active core in* any *possible core mapping with* m *simultaneously active cores, that results in a maximum temperature (in the steady-state) among all cores which does not exceed* T_{DTM}*, and that will not violate* P_{max}*. The value of* $P_{TSP}^{worst}(m)$ *is computed as*

$$P_{TSP}^{worst}(m) = \begin{cases} P_{TSP}^{\star worst}(m) & \textit{if } P_{TSP}^{\star worst}(m) \leq R(m) \\ R(m) & \textit{otherwise.} \end{cases} \tag{8}$$

PROOF. The first case is based on Lemma 4, and the second case is based on the proof of Theorem 1. Thus, the theorem is proven. □

Given that matrix $\mathbf{H}$ is only build once, we have that the total time complexity for computing $P_{\text{TSP}}^{\text{worst}}(m)$ for a given m is $O(M(N + M' \log M'))$, and for computing $P_{\text{TSP}}^{\text{worst}}(m)$ for all $m = 1, 2, \ldots, M'$ is $O(MM'(N + \log M'))$. The corresponding pseudo-code is presented in Algorithm 2. Both Algorithm 1 and Algorithm 2 are implemented as an open-source tool available at http://ces.itec.kit.edu/download.

7. TRANSIENT STATE CONSIDERATIONS

Due to the effect of transient temperatures, if TSP is computed for T_{DTM}, the temperatures of some cores may exceed the value of T_{DTM} during short time intervals when there are changes or increases in power that reach TSP. When this happens, DTM is triggered, resulting in some performance losses. *When using a constant power constraint, e.g., TDP, these effects due to transient temperatures are also present.*

The following example uses HotSpot [9] with its default configuration (detailed in Section 8.1) to show this effect for systems constrained both by (1) TDP and (2) TSP. The same example applies for both cases. Consider a many-core system of 16 cores with the same settings as in the motivational example from Section 1, with $P_{\text{inact}}^{\text{core}}$ of 0 Watts, and (1) TDP of 90.2 Watts per-chip, or (2) $P_{\text{TSP}}^{\text{worst}}(4) = 14.67\text{W}$ and $P_{\text{TSP}}^{\text{worst}}(8) = 11.27\text{W}$, i.e., TSP of 14.67 Watts and 11.27 Watts per-core when activating 4 and 8 cores, respectively. Figure 7 presents simulations in which during $t = [0s, 0.5s]$ there are 8 active cores according to Figure 1b, consuming 11.27 Watts each (90.2 Watts in total). During $t = [0.5s, 1s]$, these cores are shut-down and we activate 4 cores according to Figure 1a, consuming 14.67 Watts each (58.7 Watts in total). That is, if constrained by (1) TDP, the system consumes TDP during $t = [0s, 0.5s]$, and less than TDP during $t = [0.5s, 1s]$. Moreover, if constrained by (2) TSP, the system consumes the corresponding TSP according to the number of active cores all the time. Nevertheless, although for both cases the power budgets are not violated, Figure 7 shows that during $t = [0.5s, 1s]$ the temperature of at least one core exceeds the 80°C threshold for triggering DTM.

If the frequency of the changes in power that produce this effect is too high, then DTM could be triggered frequently, and such performance losses would be noticeable. For such cases, one strategy is to quantify the maximum values that these temperatures can actually reach during the transient state. The difference between such maximum transient temperatures and T_{DTM} is denoted as $\Delta T_{\text{transient}}^{\max}$, with value 3.08°C for the example in Figure 7. Therefore, by computing TSP for $T_{\text{DTM}} - \Delta T_{\text{transient}}^{\max}$, we can make sure that the transient temperatures never reach T_{DTM}. Nevertheless, depending on the floorplan and the resulting thermal capacitances, it can happen that the transient temperatures for

Algorithm 2 Uniform TSP for all worst-case mappings

```
Input: $\mathbf{P}^{\text{blocks}}$, $T_{\text{amb}}$, $T_{\text{DTM}}$, $P_{\text{inact}}^{\text{core}}$, $P_{\max}$, and floorplan;
Output: Uniform TSP for all worst-case mappings;
    {First build auxiliary matrix $\mathbf{H} = [h_{i,j}]_{M \times M'}$}
 1: for $i = 1$ to $M$ do
 2:     for $j = 1$ to $M'$ do
 3:         $h_{i,j} \leftarrow {b^{\text{-}1}}_{k_i, k'_j}$;
 4:     end for
 5:     Re-order row $i$ of matrix $\mathbf{H}$ in a decreasing manner;
 6: end for
 7: for $m = 1$ to $M'$ do
        {Then compute $P_{\text{TSP}}^{\star\text{worst}}(m)$ from Equation (7)}
 8:     $P_{\text{TSP}}^{\star\text{worst}}(m) \leftarrow \infty$;
 9:     for $i = 1$ to $M$ do
10:         $auxP \leftarrow T_{\text{DTM}} - P_{\text{inact}}^{\text{core}} \cdot \sum_{j=m+1}^{M'} h_{i,j}$;
11:         $auxP \leftarrow auxP - \sum_{j=1}^{N} {b^{\text{-}1}}_{k_i,j} \left(p_j^{\text{blocks}} + T_{\text{amb}} \cdot g_j\right)$;
12:         $auxP \leftarrow \frac{auxP}{\sum_{j=1}^{m} h_{i,j}}$;
13:         if $auxP < P_{\text{TSP}}^{\star\text{worst}}(m)$ then
14:             $P_{\text{TSP}}^{\star\text{worst}}(m) \leftarrow auxP$;
15:         end if
16:     end for
        {Finally compute $P_{\text{TSP}}^{\text{worst}}(m)$ according to Equation (8)}
17:     if $P_{\text{TSP}}^{\star\text{worst}}(m) \leq \mathrm{R}(m)$ then
18:         $P_{\text{TSP}}^{\text{worst}}(m) \leftarrow P_{\text{TSP}}^{\star\text{worst}}(m)$;
19:     else
20:         $P_{\text{TSP}}^{\text{worst}}(m) \leftarrow \mathrm{R}(m)$;
21:     end if
22: end for
23: return $P_{\text{TSP}}^{\text{worst}}(m)$ for all $m = 1, 2, \ldots, M'$;
```

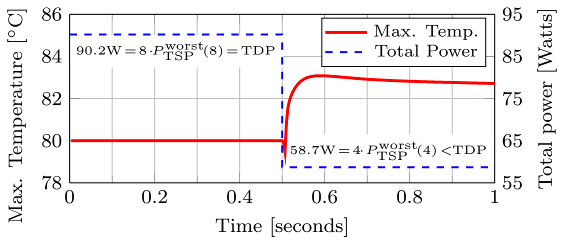


Figure 7: Transient example for both TSP and TDP (the red line shows the maximum temperature among all cores). During $t = [0s, 0.5s]$ there are 8 active cores according to Figure 1b, consuming 11.27 Watts each. During $t = [0.5s, 1s]$, these cores are shut-down and we activate 4 cores according to Figure 1a, consuming 14.67 Watts each.

this new TSP are too pessimistic compared to T_{DTM}. For such cases, with just a few iterations, a near optimal value for which to compute TSP can be achieved. This method should be applied offline, due to the overheads for obtaining $\Delta T_{\text{transient}}^{\max}$ for each case. *A similar method should be adopted for systems that use constant power budgets, e.g., TDP.*

Procedure Example: Consider a many-core system of 16 cores with the same settings as in the motivational example from Section 1, and an ambient temperature of 45°C. For the power consumptions, consider a scenario in which the power of the cores changes every 0.1 seconds. The changes in power are for a random number of active cores. The mapping and power values adopted in all cases are those of TSP for the worst-case, computed through Algorithm 2, according to the number of active cores at each time instant. We run simulations with HotSpot for such a case, and present the results in Figure 8a. Figure 8a presents the temperature on each core with a different color, and the maximum temperature among all cores at any given time is highlighted by the bold-red curve. We can observe that for this floorplan the thermal capacitances are not negligible, which results in long transient state periods. Therefore, for such a case, TSP should be recomputed for some temperature below T_{DTM}.

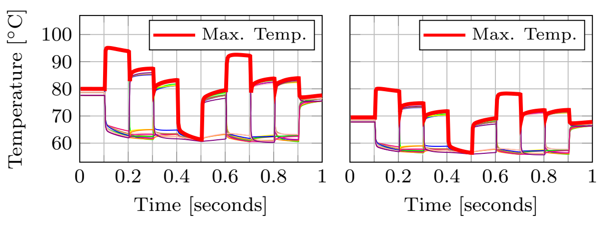


Figure 8: Transient example for 16 cores. The number of active cores and their power consumption changes every 0.1 seconds. The mapping and power values adopted are those of worst-case TSP computed for (a) 80°C and (b) 69.5°C.

Looking at Figure 8a, we can quantify $\Delta T^{\max}_{\text{transient}}$, which we set to 15°C, and then recompute TSP for 65°C. Given that this results in a maximum transient temperature of 74°C, which is too pessimistic, we need a higher value. Thus, we iterate computing TSP and running transient temperature simulations. After just 5 iterations, specifically, computing TSP for 80°C, 65°C, 71°C, 69°C, and 69.5°C, we reach a near optimal value for which to compute TSP, which for this floorplan is 69.5°C. Clearly, the new TSP values are smaller than those for a TSP computed for 80°C.

Finally, we run the transient simulations with the same settings, but with power states according to the new TSP values. The results are presented in Figure 8b, which shows that when computing TSP for 69.5°C, the temperature is always below T_{DTM}. When computing TSP online for particular mapping scenarios, the target temperature should also be 69.5°C, and not the original 80°C.

If, *unlike* Figure 8, the changes in power are *not* too frequent, such that DTM is triggered with low frequency during short time intervals, then computing TSP for T_{DTM} is still a better approach that results in higher total performance.

8. EVALUATIONS

This section presents simulations conducted with gem5 [2], McPAT [14], and HotSpot [9]. We compare the total system performance for using six different power budget techniques: TSP for given mappings, worst-case TSP, two constant power budgets per-chip, a constant power budget per-core, and boosting over a constant power budget per-chip like Intel's Turbo Boost [10,19] or AMD's Turbo CORE [16].

8.1 Setup

For our hardware platform, we consider the 64 cores system presented in Figure 4, which is based on simulations conducted on gem5 [2] and McPAT [14] for *out-of-order* Alpha 21264 cores in 22 nm technology. The cores are composed by several units: an instruction fetch unit (IFU), an execution unit (EXU), a load and store unit (LSU), an out-of-order (OOO) issue/dispatch, and a private L1 cache. According to our simulations, as shown in Figure 4, each core has an area of 9.6mm^2. There is a shared L2 cache of 2 MB and a memory controller every 4 cores, with a combined area of 4.7mm^2. For the floorplan, we consider one thermal block for each core, and independent blocks for the L2 caches and the memory controllers.

For such a floorplan, we obtain the values for $\mathbf{B}$, $\mathbf{B}^{-1}$, and $\mathbf{G}$, by using HotSpot [9] v5.02 with its default configuration in the block model mode. That is, chip thickness of 0.15mm, silicon thermal conductivity of $100\frac{W}{m\cdot K}$, silicon specific heat of $1.75\cdot 10^6\frac{J}{m^3\cdot K}$, a heat sink of 6×6 cm and 6.9 mm thick, heat sink convection capacitance of $140.4\frac{J}{K}$, heat sink convection resistance of $0.1\frac{K}{W}$, heat sink and heat spreader thermal conductivity of $400\frac{W}{m\cdot K}$, heat sink and heat spreader specific heat of $3.55\cdot 10^6\frac{J}{m^3\cdot K}$, a heat spreader of 3×3 cm and 1 mm thick, interface material thickness of $20um$, interface material thermal conductivity of $4\frac{W}{m\cdot K}$, and interface material specific heat of $4\cdot 10^6\frac{J}{m^3\cdot K}$.

We consider an ambient temperature of 45°C, and a threshold temperature that triggers DTM, T_{DTM}, of 80°C. For all the evaluated power constraints, except when a boosting technique is being applied, we consider that DTM is triggered whenever the temperature of at least one core exceeds the T_{DTM} of 80°C. We assume a DTM technique, similar to that in [10], which gates 67.5% of the clock cycles during a control period. Furthermore, we assume a $P_{\max}$ of 250 Watts, which is a common value seen in the literature [10]. The value of $P^{\text{core}}_{\text{inact}}$ is taken from the simulations with McPAT, and set to 0.14 Watts. We assume available frequencies from 100 MHz to 4.0 GHz, in steps of 100 MHz. The voltage settings for each frequency are taken from the work in [8].

As benchmarks, we use applications from the Parsec benchmark suite [1]. Specifically, an x264 application (H.264 video encoder), a body track application, an option pricing application with Black-Scholes Partial Differential Equation, and a pricing application of a portfolio of swaptions. All applications can run $1, 2, \ldots, 4$ parallel dependent threads.

8.2 Power Constraints

In this subsection we compute the power constraints used for our simulations. Using Algorithm 2, we compute TSP for the worst-case mappings, i.e., $P^{\text{worst}}_{\text{TSP}}(m)$ for all $m = 1, 2, \ldots, M'$. Figure 9 presents the computed TSP values per-core as TSP$_{\text{worst}}$, resulting in a decreasing function with respect to the number of active cores. For presentation purposes, Figure 10 shows TSP estimations at a chip level, by multiplying the TSP$_{\text{worst}}$ values from Figure 9 with the number of active cores for each case. In Figure 10, TSP results in a non-decreasing function with respect to the number of active cores. Moreover, to compare TSP for different core mappings, we consider $64\cdot 10^5$ random mappings (10^5 for each number of active cores), compute TSP using Algorithm 1 for every given mapping, and present the highest resulting TSP values as TSP$_{\text{best}}$ in Figures 9 and 10. On a desktop computer with a 64-bit quad-core Intel Sandybridge i5-2400 CPU running at 3.10GHz, we measure the execution time required by Algorithm 1 to compute TSP for every random mapping, resulting in an execution time of at most 5.47 ms among all measurements, *which is suitable for online usage.*

For the constant per-chip power constraints, since this is a simulated platform, we cannot simply consider TDP because we have no datasheet with that information. Therefore, we consider two different per-chip power budgets, which coincide with $m\cdot P^{\text{worst}}_{\text{TSP}}(m)$ for $m = 16$, and $m = 64$. Specifically, these power budgets are 150.0 Watts per-chip, and 225.0 Watts per-chip. These constant per-chip budgets are also representative TDP values for current technologies [10], and are shown as horizontal lines in Figure 10. In Figure 9, we estimate the maximum power per core when these constraints are equally distributed among the active cores. For the constant per-core power constraint, we consider it to be equal to TSP when simultaneously activating all cores, i.e., $P^{\text{worst}}_{\text{TSP}}(m)$ for $m = 64$. This results in a constant power budget of 3.61 Watts per-core, which is represented by a horizontal line in Figure 9 and by an increasing linear function in Figure 10. *Note that representing TSP and the constant per-core power constraint in Figure 10 does not mean that either constraint should be considered as a per-chip constraint. Both budgets should be strictly considered at a per-core level. We include them in Figure 10 only to compare their resulting total system power to the per-chip power budgets. Similarly, the opposite applies when representing the constant per-chip budgets in Figure 9.*

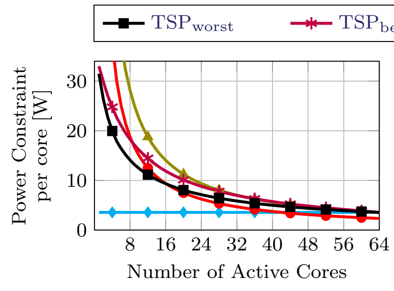


Figure 9: Worst- and best-case TSP for the floorplan in Figure 4, compared to a constant per-core budget, and estimations of constant per-chip budgets equally distributed among active cores.

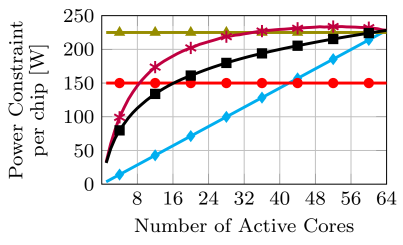


Figure 10: Constant per-chip budgets, compared to multiplying the number of active cores by a constant per-core budget, and the worst- and best-case TSP for the floorplan from Figure 4.

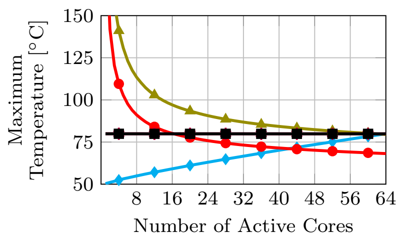


Figure 11: Maximum steady-state temperatures among all blocks (DTM deactivated), when using TSP, a constant per-core budget, and equally distributed constant per-chip budgets.

With regards to temperature, Figure 11 presents simulations conducted in HotSpot that show the maximum temperature (in the steady-state) among all blocks as a function of the number of simultaneously active cores, for the discussed power budgets, in case that DTM is deactivated. As expected, when consuming TSP in all active cores, the maximum temperature among all blocks is 80°C. Moreover, from Figure 11 we can conclude that whenever TSP is greater than any constant power budget, if such a power budget is used as the power constraint, the system could actually be consuming more power without reaching T_{DTM}, which accounts for performance losses. Contrarily, when a constant power budget exceeds TSP, this means that if the cores consume more power than TSP, most likely DTM will be triggered almost the entire time.

8.3 Dark Silicon Estimations

Through Figure 9, we can easily estimate the amount of dark silicon. For example, we assume that the minimum power consumption for activating a core at its lowest speed is 4 Watts (*this is not a real practical setting, but we choose such a value for presentation purposes*). Then, considering the values of TSP, no more than 54 cores could be active simultaneously, which results in 15.63% of the chip being dark at all times. If a constant per-chip power budget is used for the same example estimations, for example 150.0 Watts per-chip, then only 37 cores could be activated simultaneously, resulting in 42.19% of the chip being dark, which is much higher than the estimations for TSP.

8.4 Performance Simulations

By using gem5, we run simulations for our benchmark applications, for the corresponding number of threads, and for all available frequencies. From the results of gem5 we obtain the total instructions per second (IPS) for each one of the simulated cases. The traces from gem5 are then injected into McPAT to obtain the corresponding power consumption values for the described architecture.

With the results from gem5 and McPAT, by considering the power constraints described in Section 8.2 we can then derive Figure 12 and Figure 13. Figure 12 presents the average total cycles per second and Figure 13 presents the average total IPS count, for considering each one of the different power budgets and different numbers of active cores. We assume that every active core runs at the highest possible frequency such that the power budget under consideration is not exceeded. We conduct additional simulations with HotSpot to evaluate the temperature behaviour of the cores, such that we can apply the DTM technique detailed in Section 8.1 when building Figure 12 and Figure 13.

Furthermore, in order to also compare with a boosting technique, in Figures 12 and 13 we also consider boosting over one of the constant per-chip power budgets described in Section 8.2. Specifically, we boost to 1.5x over 150.0 Watts per-chip. Similar to [10], we set the boosting interval to 0.3 seconds, followed by 1 second of operation under the nominal power budget. If under nominal operation the temperature of at least one core also exceeds T_{DTM}, we assume that cores cannot be boosted for this condition and the normal DTM technique is applied.

Figures 12 and 13 show that using TSP as power constraint results in a higher total performance for almost all cases. Moreover, we obtain the *average percentage increase in performance* for using TSP for the worst-case mappings. This is done by computing the ratios between the total IPS for each power budget and the total IPS for TSP$_{\text{worst}}$ from Figure 13, for all number of active cores and all applications, and computing the average among these ratios. Specifically, using TSP for the worst-case mappings results in 50.5% and 14.2% higher average total IPS compared to all constant power budgets and the boosting technique, respectively.

When using just a few cores, TSP and the per-chip budgets behave in the same way, given that cores do not consume so much power as indicated by each budget for any of the tested benchmarks. Then, as expected from Figure 11, using the entire per-chip budget can derive in very frequent triggers of DTM, considerably reducing the performance.

The small drops in the performance observed at some points when using an additional core for all budgets, including TSP, are due to the available frequency steps. That is, generally, when using an additional core, the frequency of all cores has to be reduced in order to meet the power constraints. Depending on the available frequencies and the number of active cores, the resulting system performance might be bigger when using one less core.

Under just a few conditions, the boosting technique or the 225 Watts per-chip budget result in higher performance than TSP. Nevertheless, this case is not due to inefficiency of TSP as a power constraint, but due to the available frequency steps in our platform. That is, for example when activating all 64 cores, the value of TSP is 3.56 Watts. In the case of application x264, we can run at 2.3 GHz consuming 2.90 Watts per-core under TSP, because running at 2.4 GHz consumes 3.68 Watts per-core. On the other hand, for the case of the 225 Watts per-chip budget, some cores can further run at 2.4 GHz, still satisfying the power budget and resulting in a slightly better performance. However, if there would exist an intermediate frequency step for which cores consume a power closer to that of TSP, then TSP would result in a better performance. Moreover, this only happens in a few cases, while in general using 225 Watts per-chip triggers DTM almost the entire time.

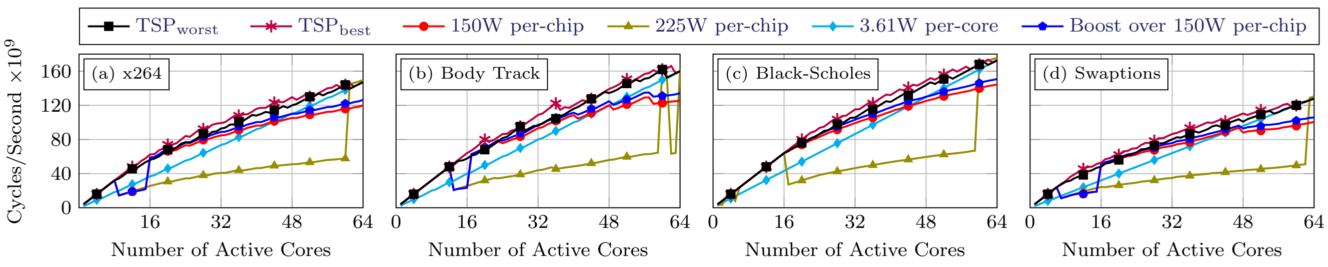


Figure 12: Simulation results comparing the *average total cycles per second* when using different power budgets.

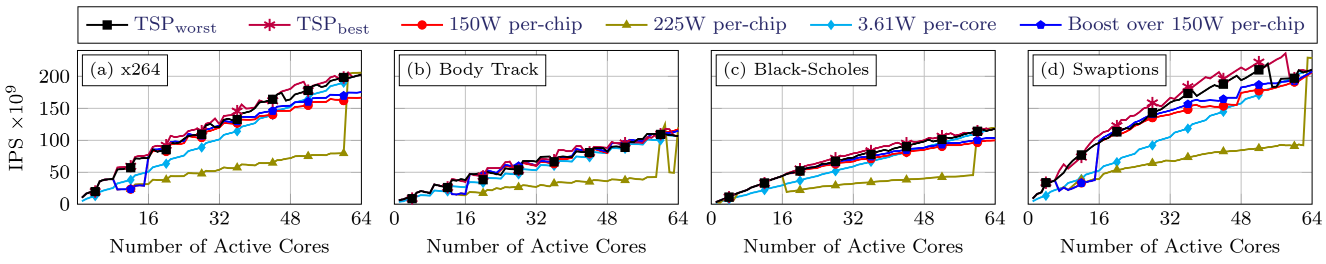


Figure 13: Simulation results for the *average total system performance* when using different power budgets. From Figure 12, x264 and Swaptions have a high instruction level parallelism (ILP), while Body Track and Black-Scholes are with lower ILP.

9. CONCLUSIONS

Using a *single* and *constant* power constraint, for example, TDP, is a pessimistic approach for many-core systems. Boosting techniques do not deal with this issue properly, because the system can only be boosted during short time intervals, and some cool-down time is needed between these intervals. Thus, this paper presented a new power budget concept, called Thermal Safe Power (TSP), which results in a higher total system performance, while the maximum temperature among all cores remains below the threshold level that triggers DTM. TSP is a fundamental new step and advancement towards dealing with the dark silicon problem as it alleviates the pessimistic bounds of TDP and enables system designers and architects to explore new avenues for performance improvements in the dark silicon era.

For a specific floorplan and ambient temperature, TSP can be computed offline, in order to obtain safe power constraints for the worst cases, allowing the system designers to abstract from mapping decisions. Moreover, TSP can also be computed online, for a particular mapping of active cores and ambient temperature. The simulations show the validity of our arguments, comparing the total performance of using TSP, several constant power constraints, and a boosting technique. TSP can also be used to estimate the amount of dark silicon, which results in more optimistic estimations than those using constant power budgets.

10. ACKNOWLEDGEMENTS

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre *Invasive Computing* [SFB/TR 89], by Baden Württemberg MWK Juniorprofessoren-Programms, and by the Research Grants Council of the Hong Kong Special Administrative Region, China [Project CityU 117913].

11. REFERENCES

[1] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC benchmark suite: Characterization and architectural implications. In *PACT*, pages 72–81, 2008.
[2] N. Binkert, B. Beckmann, and others. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.
[3] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran. darkNoC: Designing energy-efficient network-on-chip with multi-vt cells for dark silicon. In *DAC*, pages 161:1–161:6, 2014.
[4] J. Cebrian, L. Natvig, and J. Meyer. Improving energy efficiency through parallelization and vectorization on intel core i5 and i7 processors. In *SC Companion*, pages 675–684, 2012.
[5] G. Dantzig and M. Thapa. *Linear Programming 2: Theory and Extensions*. Springer-Verlag, 2003.
[6] T. Ebi, D. Kramer, W. Karl, and J. Henkel. Economic learning for thermal-aware power budgeting in many-core architectures. In *CODES+ISSS*, pages 189–196, 2011.
[7] H. Esmaeilzadeh, E. Blem, R. St.Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *ISCA*, pages 365–376, 2011.
[8] A. Grenat, S. Pant, R. Rachala, and S. Naffziger. 5.6 adaptive clocking system for improved power efficiency in a 28nm x86-64 microprocessor. In *ISSCC*, pages 106–107, 2014.
[9] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Transactions on VLSI Systems*, 14(5):501–513, May 2006.
[10] Intel Corporation. Dual-core intel xeon processor 5100 series datasheet, revision 003, August 2007.
[11] F. Kriebel, S. Rehman, D. Sun, M. Shafique, and J. Henkel. ASER: Adaptive soft error resilience for reliability-heterogeneous processors in the dark silicon era. In *DAC*, pages 12:1–12:6, 2014.
[12] E. Kultursay, K. Swaminathan, V. Saripalli, V. Narayanan, M. T. Kandemir, and S. Datta. Performance enhancement under power constraints using heterogeneous CMOS-TFET multicores. In *CODES+ISSS*, pages 245–254, 2012.
[13] J. Lee and N. S. Kim. Optimizing throughput of power- and thermal-constrained multicore processors using DVFS and per-core power-gating. In *DAC*, pages 47–50, 2009.
[14] S. Li, J.-H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *MICRO-42*, pages 469–480, 2009.
[15] T. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin. Hierarchical power management for asymmetric multi-core in dark silicon era. In *DAC*, pages 174:1–174:9, 2013.
[16] S. Nussbaum. AMD trinity APU. In *Hot Chips*, 2012.
[17] B. Raghunathan and S. Garg. Job arrival rate aware scheduling for asymmetric multi-core servers in the dark silicon era. In *CODES+ISSS*, 2014.
[18] B. Raghunathan, Y. Turakhia, S. Garg, and D. Marculescu. Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors. In *DATE*, pages 39–44, 2013.
[19] E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan, and E. Weissmann. Power-management architecture of the intel microarchitecture code-named sandy bridge. *Micro, IEEE*, 32(2):20–27, March 2012.
[20] M. Shafique, S. Garg, J. Henkel, and D. Marculescu. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *DAC*, pages 185:1–185:6, 2014.