

Energy and Peak Power Efficiency Analysis for the Single Voltage Approximation (SVA) Scheme

Santiago Pagani, *Student Member, IEEE*, Jian-Jia Chen, *Member, IEEE*, and Jörg Henkel, *Fellow, IEEE*

Abstract—Energy-efficiency is an important issue in computing systems, and operating within a safe power budget is a necessary constraint. This paper presents a simple and practical solution both for energy minimization and peak power reduction, called *Single Voltage Approximation (SVA)* scheme, for periodic real-time tasks on multi-core systems with a shared supply voltage in a voltage island. SVA is inspired by the *Single Frequency Approximation (SFA)* scheme, in which all the cores in the island run at a single voltage and frequency such that all tasks can meet their deadlines. In SVA, all the cores in the island are also executed at the same single voltage as in SFA. However, the frequency of each core is individually chosen, such that the tasks in each core can meet their deadlines, but without running at unnecessarily high frequencies. Thus, all the cores are executing tasks all the time, and there is no need for any Dynamic Power Management (DPM) technique for reducing the energy consumption for idling. For task partitioning, SVA is combined with the *Double Largest Task First (DLTF)* partitioning scheme.

Most importantly, this paper provides comprehensive analysis for combining DLTF and SVA, deriving its worst-case behavior both for energy minimization and peak power reduction, compared against the optimal solutions. Our analysis shows that, depending on the hardware, the energy consumption by combining DLTF and SVA is at most 1.95 (2.21, 2.42, 2.59, respectively), compared to the optimal solutions, when the voltage island has up to 4 (8, 16, 32, respectively) cores, which outperforms the worst-case factors of SFA when the cores fail to sleep efficiently. For peak power reduction, due to running at slower frequencies, combining DLTF and SVA always outperforms SFA, both in average and corner cases. Finally, we extend our analysis considering multi-core systems with discrete voltage and frequency pairs, and multiple voltage islands.

Keywords—*Single Voltage Approximation (SVA)*, *Single Frequency Approximation (SFA)*, *Peak Power*, *Energy Efficiency*, *Task Partitioning*, *Multiple Voltage Islands*, *Power Management*

I. INTRODUCTION

ENERGY efficiency and low power design are major issues in modern computing systems, for example to prolong the battery lifetime of embedded systems or to reduce the power bills for servers. This was a main motivation for moving from single-core to multi-core platforms, mainly to balance the power consumption and computation performance. Furthermore, the trend of improvement of the power efficiency in transistors is not following the current trend for higher

integration. In the following years, this issue will result in dramatical power density increases in chips with every new process generation, deriving what is referred to as the *dark silicon* problem [8], [26], where big areas of a chip will need to remain *gray* (slowed down) or *dark* (turned off) at all times.

Reducing the peak power consumption is a major step towards dealing with dark silicon. For example, consider the *Thermal Dynamic Power (TDP)*, which is a commonly used abstraction that allows the system to guarantee that the thermal constraints are satisfied [13]. By keeping the peak power consumption below the TDP value, the system can safely run without further thermal considerations. Nevertheless, energy efficiency should only be sacrificed when there are no feasible solutions under the given power constraints.

The difference between energy and power can make existing energy minimization schemes unsuitable for peak power minimization. When referring to power consumption, the time instant in question is absolutely relevant, given that power is an instantaneous measurement. On the other hand, energy is the integration of power through time. This means that minimizing the energy consumption is generally equivalent to minimizing the average power consumption, while the peak power consumption values are ignored. An energy minimization strategy that ignores the power budgets could easily violate the dark silicon constraints, while consuming the same average power.

As shown in the literature, e.g. [15], the dynamic power consumption (mainly generated by switching activities) and the static power consumption (mainly generated by leakage currents) are the two major sources of power consumption in CMOS processors. Moreover, the power consumption in a CMOS core is mainly a convex function. Thus, it is usually better to execute at low frequencies both for energy and peak power minimization by using *Dynamic Voltage and Frequency Scaling (DVFS)* techniques. Nevertheless, for systems with considerable static power consumption, executing at low frequencies by recurring to greater parallelism might consume more energy and power than using less cores at higher frequencies, due to the simultaneous activation of several cores that consume high static power for long periods of time. Moreover, next-generation multi-core systems consider architectures with several voltage islands, in which the cores in an island share the same supply voltage, naturally consolidating a cluster [4], [11]. For example, Intel's Single-chip Cloud Computer (SCC) [12], [14] counts with such a feature.

Motivation: In multi-core systems with a shared supply voltage in a voltage island (or systems with a global supply voltage) and periodic real-time tasks, it is necessary to choose a policy for the task partitioning stage and to decide the voltage

Santiago Pagani is with the Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. E-mail: pagani@kit.edu

Jian-Jia Chen is with the Department of Informatics, TU Dortmund University, Dortmund, Germany. E-mail: jian-jia.chen@cs.uni-dortmund.de

Jörg Henkel is with the Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. E-mail: henkel@kit.edu

of the island and the frequencies of the cores for execution.

For periodic tasks, a practical strategy is to partition the tasks using the *Double Largest Task First* (DLTF) scheme [23], and then use a single voltage and frequency for execution. Particularly, the lowest voltage and frequency such that all tasks meet their deadlines, denoted as the *Single Frequency Approximation* (SFA) scheme [22], [23]. However, under SFA, cores with low computational requirements are forced to run at high frequencies in order to meet the deadlines of the highest loaded core. Another strategy is to use the same single voltage as in SFA, but to choose the frequencies of the cores individually, such that the tasks in all cores meet their deadlines, but without executing cores at unnecessarily high frequencies. We define such a strategy as the *Single Voltage Approximation* (SVA) scheme.

Combining DLTF with SVA (referred to as DTLF-SVA) is not the optimal DVFS scheduling strategy for energy minimization or peak power reduction. Nevertheless, it is a *practical* and *simple* solution, which significantly reduces the overhead for changing the supply voltage of the islands and frequencies of cores, because SVA does not require voltage or frequency changes at run time. Moreover, given that all the cores are executing tasks at all times, there is no need for any *Dynamic Power Management* (DPM) technique for reducing the energy consumption for idling. However, the worst-case performance of DTLF-SVA in terms of energy efficiency and peak power reduction is an open problem.

Objective: Motivated by the above discussions, the goal of this paper is to present and analyze a *simple* and *practical* solution, both for energy minimization and peak power reduction, which uses DLTF for task partitioning and SVA to decide the voltage of the islands and the frequencies of the cores. For the analysis, we compare such a scheme against the optimal energy and peak power solutions, especially for the state-of-the-art designs, that have a few number of cores per voltage island. Furthermore, we compare these results against the results of SFA [22], [23].

Our Contributions: For periodic real-time tasks in a system with multiple voltage islands, our contributions are:

- We present the SVA scheme, and propose combining DLTF and SVA for energy minimization and peak power reduction.
- We provide comprehensive analysis for the approximation factor of combining DLTF and SVA, both for energy minimization and peak power reduction, in a single voltage island with continuous voltages and frequencies.
- We compare the results of SVA against the results of SFA, extending the results in [22], [23] for SFA considering a more general power model for fair comparison.
- We extend our results to consider cores with a limited set of available voltages and frequencies.
- We extend our results to consider systems with multiple voltage islands, by applying a dynamic programming algorithm for mapping tasks sets to voltage islands [24].

II. RELATED WORK

Energy-efficient scheduling for homogeneous multi-core systems with per-core DVFS has been widely explored for

real-time systems, e.g., [1], [5], [6], [28]. The work in [6] shows that applying the *Largest-Task-First* (LTF) strategy for task mapping results in solutions with energy consumption approximation factors that depend on the hardware platforms. Particularly, by turning off cores to reduce the energy consumption, Xu et al. [28] and Chen et al. [6] propose polynomial-time algorithms to derive task mappings that try to execute at a *critical frequency*. Even though per-core DVFS is energy-efficient, based on VLSI circuit simulations, Herbert and Marculescu [11] suggest that it suffers from complicated design problems, making it costly for implementation.

For systems with a shared supply voltage, Yang et al. [29] use DVFS to minimize the energy consumption in systems with negligible static power consumption and frame-based real-time tasks. The assumptions in [29] are relaxed in [7], by considering periodic real-time tasks with non-negligible static and voltage-independent power consumptions and non-negligible overhead for sleeping. For sets of periodic real-time tasks, the analysis in [22] derives the worst-case approximation factor for the *Single Frequency Approximation* (SFA) scheme, in terms of energy consumption. The analysis is later extended in [23] to also consider the task partitioning stage, presenting the DLTF scheme for task partitioning.

The works mentioned above focus on the energy consumption, and disregard the effects of the instantaneous power consumption. In this aspect, there are many works in the literature that focus on improving performance under a given power budget [16], [18], [21]. However, these works do not address the issue of energy consumption and do not provide any timing guarantees, making them unsuited for real-time systems. To the best of our knowledge, the only related work that focuses on reducing the peak power consumption in real-time systems was recently conducted by Lee et al. [17]. In [17], without considering DVFS, authors propose a scheduling algorithm that finds a set of concurrent executable tasks, such that the design-time chip-level peak power consumption is minimized and all timing requirements are satisfied. Nevertheless, energy consumption is also not addressed in this work.

In conclusion, for real-time systems, due to the appearance of the *dark silicon* problem, solutions that focus on energy minimization without considering the peak power consumption might exceed the available power budgets, and should therefore be revisited considering additional constraints.

III. SYSTEM MODEL AND PROBLEM DEFINITION

A. Hardware Model

We focus in a *single voltage island* of a homogeneous multi-core system with multiple voltage islands, where all the M cores in the island have the same supply voltage at any given time point, and the frequencies of the cores can be chosen individually, e.g., one voltage island of SCC [12], [14]. The island can change the shared supply voltage and the frequencies of the cores by adopting DVFS. For a core to stably support a frequency, the supply voltage of the island has to be adjusted above a minimum value. This minimum voltage value depends on the frequency, and higher frequencies require higher minimum voltages. Particularly, in order to

consume less power and to save energy, we adjust the supply voltage of the island to the least available value such that the highest frequency among all cores in the island achieves stable execution, and thus, all cores in the island achieve stable execution. We define the highest frequency among all cores in the island as s_M . For the analysis in Sections VI and VII we consider continuous voltages and frequencies in the range of $[v_{\min}, v_{\max}]$ and $[s_{\min}, s_{\max}]$, respectively. We then extend these results in Section VIII, considering discrete voltage and frequency pairs $\{(v_1, f_1), (v_2, f_2), \dots, (v_F, f_F)\}$.

Given that the voltage of the island is set to the minimum value such that s_M can be stably achieved, we denote the power consumption of a core executing a certain task at frequency s as $P_{\text{core}}(s_M, s)$. Specifically, we use

$$P_{\text{core}}(s_M, s) = \alpha \cdot s_M^{\gamma-1} \cdot s + \beta \cdot s_M + \kappa, \quad (1)$$

where $\alpha > 0$ is a constant dependent on the effective switching capacitance, $\gamma > 1$ is related to the hardware, $\alpha \cdot s_M^{\gamma-1} \cdot s \geq 0$ represents the dynamic power consumption, $\beta \cdot s_M \geq 0$ represents the static power consumption, and $\kappa \geq 0$ represents the independent power consumption. In CMOS processors, γ is generally modeled equal to 3, e.g., [7], [22], [23], [29]. Because the voltage of the island depends on s_M , the dynamic power consumption of a core depends on its frequency and the voltage of the island to the power of $\gamma - 1$ (generally 2 in CMOS processors). Similarly, the static power consumption is linearly related to the voltage of the island. The independent power consumption is a constant value for keeping the core in execution mode. Thus, running at low voltages and frequencies reduces the power consumption. This power consumption function has been widely used in the literature, e.g., [1], [7], [28], and is more general than the model used in [22], [23], [29]. Moreover, in case a core runs at the frequency which decides the voltage of the island, the power consumption becomes

$$P_{\text{core}}(s) = \alpha \cdot s^\gamma + \beta \cdot s + \kappa. \quad (2)$$

For example, Fig. 1 presents power consumption values for a 22nm *out-of-order* Alpha 21264 core, based on simulations conducted on gem5 [3] and McPAT [19] for an H.264 video encoder from the Parsec benchmark suite [2]. As shown in Fig. 1, these experimental power values can be modeled by power parameters $\gamma = 3$, $\alpha = 0.27 \frac{\text{W}}{\text{GHz}^3}$, $\beta = 0.52 \frac{\text{W}}{\text{GHz}}$, and $\kappa = 0.5 \text{ W}$, resulting in a goodness of fit of: *Sum of Squares due to Error (SSE)* of 1.058, a *Square of the correlation between the response values and the predicted response values (R-square)* of 0.9992, an *Adjusted R-square* of 0.9992 and a *Root Mean Squared Error (RMSE)* of 0.1626.

When a core finishes executing all its workload in the ready queue, it has to wait until a new task instance arrives. During this waiting interval, the core can stay idle, consuming $P_{\text{core}}^{\text{idle}}(s_M) = \beta \cdot s_M + \kappa$. Moreover, the core can also go to a low-power mode, e.g., sleep mode, that consumes $\kappa^{\text{sleep}} \geq 0$ power. As we can transfer the power consumption κ^{sleep} to the system for being active, without loss of generality, we can set $P_{\text{core}}(s)$ to $P_{\text{core}}(s) - \kappa^{\text{sleep}}$, and $P_{\text{core}}^{\text{idle}}(s_M)$ to $P_{\text{core}}^{\text{idle}}(s_M) - \kappa^{\text{sleep}}$, such that we can disregard the effect of

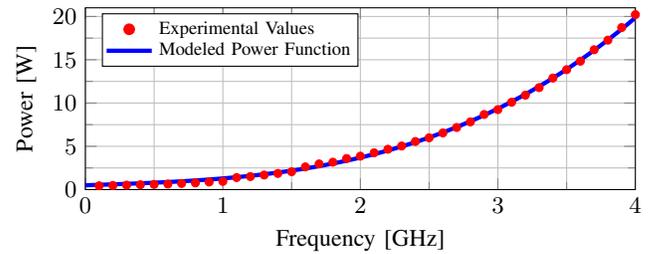


Fig. 1: Experimental results for a 22 nm *out-of-order* Alpha 21264 core, based on simulations conducted on gem5 [3] and McPAT [19] for an H.264 video encoder from the Parsec benchmark suite [2], and the derived power model $P_{\text{core}}(s)$ when $\gamma = 3$, $\alpha = 0.27 \frac{\text{W}}{\text{GHz}^3}$, $\beta = 0.52 \frac{\text{W}}{\text{GHz}}$, and $\kappa = 0.5 \text{ W}$.

the power consumption of a core in sleep mode. In this way, since not even the optimal solution can optimize for κ^{sleep} , we only focus on the *effective* optimization region and we avoid possible masking problems in systems with large κ^{sleep} values.

Energy is the integration of power through time, such that when a core consumes constant power during interval Δt , the energy consumed by the core is $E_{\text{core}}(s_M, s) = P_{\text{core}}(s_M, s) \cdot \Delta t$. During interval Δt , a core running at frequency s executes a certain amount Δc of core cycles, such that $\Delta t = \frac{\Delta c}{s}$. Hence,

$$E_{\text{core}}(s_M, s) = (\alpha \cdot s_M^{\gamma-1} \cdot s + \beta \cdot s_M + \kappa) \frac{\Delta c}{s}. \quad (3)$$

Furthermore, for the case in which a core runs at the frequency which decides the voltage of the island, i.e., $s = s_M$, the energy consumption on the core becomes

$$E_{\text{core}}(s) = (\alpha \cdot s^\gamma + \beta \cdot s + \kappa) \frac{\Delta c}{s}. \quad (4)$$

Equation (4) is a convex function with respect to s . Thus, by setting the first order derivative of Equation (4) with respect to s to zero, the minimum value for $E_{\text{core}}(s)$ is found when s is $\sqrt[\gamma]{\frac{\kappa}{(\gamma-1)\alpha}}$. In order to consider the case when such a value is smaller than s_{\min} , the *critical frequency* is defined as $s_{\text{crit}} = \max\left\{s_{\min}, \sqrt[\gamma]{\frac{\kappa}{(\gamma-1)\alpha}}\right\}$. The *critical frequency* represents the frequency that minimizes the energy consumption for execution, when the overhead for sleeping is considered negligible, as shown in [5], [15]. This happens because running at frequencies smaller than s_{crit} consumes excessive independent energy, which surpasses the saves in dynamic energy consumption. However, in Equation (3) for a given s_M and Δc , the dynamic energy is constant with respect to s , and the static and independent energy are decreasing with respect to s . This implies that running at frequencies slower than s_M increases the energy for execution by increasing the execution time, while reducing the energy consumption for idling. Fig. 2 shows examples of energy consumptions with respect to s for different values of s_M .

B. Task Model

This paper considers N periodic real-time tasks with implicit deadlines, i.e., $\{\tau_1, \tau_2, \dots, \tau_N\}$. We assume that there

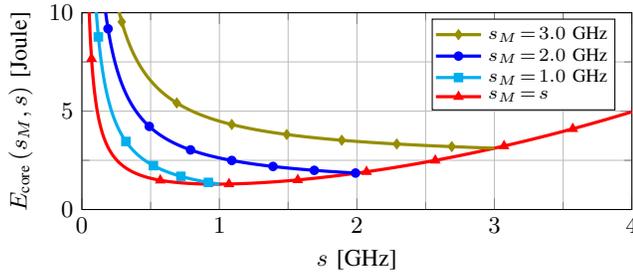


Fig. 2: $E_{\text{core}}(s_M, s)$ when $\gamma = 3$, $\alpha = 0.27 \frac{\text{W}}{\text{GHz}^3}$, $\beta = 0.52 \frac{\text{W}}{\text{GHz}}$, $\kappa = 0.5 \text{ W}$, and $\Delta c = 10^9$ cycles.

is no data dependency among tasks, i.e., we consider independent tasks. Every task τ_j releases an infinite number of task instances with period (and relative deadline) p_j and every instance has worst-case execution cycles e_j . We consider partitioned scheduling, in which each task is assigned onto a core, that is, when a task instance arrives to the system it is executed on the assigned core. Specifically, we use *Earliest-Deadline-First* (EDF) scheduling [20], where the task instance with the earliest absolute deadline on each core has the highest priority. The least common multiple among all periods of all tasks is called the *hyper-period*, denoted as L .

After the task partitioning is completed by using M cores, all N tasks are grouped into M task sets (we assume that if the partitioning strategy decides to group tasks into less than M task sets, dummy empty task sets are included in order to have M task sets). When talking about energy minimization, the optimal task partitioning is defined as a partitioning solution that results in the minimum energy consumption when using the optimal DVFS schedule for energy minimization. Similarly, when talking about peak power reduction, the optimal task partitioning is defined as a partitioning solution that results in the minimum peak power consumption when using the optimal DVFS schedule for peak power reduction. Later, in Section V-B, we show that both optimal task partitions are equivalent for the lower bounds of energy and peak power consumption. Therefore, for simplicity in presentation, we use a single notation for both cases and the optimal task partition is defined as $\{\mathbf{T}_1^*, \mathbf{T}_2^*, \dots, \mathbf{T}_M^*\}$. Without loss of generality, we assume that task set \mathbf{T}_i^* is assigned on core i , and we define its *cycle utilization* as $w_i^* = \sum_{\tau_j \in \mathbf{T}_i^*} \frac{e_j}{p_j}$, with unit *cycles per second*. By defining $w_0^* = 0$ for notational purposes and without loss of generality, we order the cores such that $0 = w_0^* \leq w_1^* \leq w_2^* \leq \dots \leq w_M^*$.

In our proposed scheme, the tasks are partitioned by using DLTF [23], which is based on the widely used *Largest-Task-First* (LTF) strategy [29]. LTF is a *Worst-Fit-Decreasing* heuristic algorithm, and is namely a reformulation of the *Longest-Processing-Time* (LPT) algorithm from [9] for the makespan problem. Task partitioning strategy DLTF considers LTF as an initial solution, and then regroups the tasks into less cores, such that as many cores as possible can be put to sleep (saving energy for idling) without increasing the energy consumption for execution. For completeness, the pseudo-code

Algorithm 1 Largest-Task-First (LTF) strategy

Input: Tasks $\{\tau_1, \tau_2, \dots, \tau_N\}$;
Output: Task sets $\{\mathbf{T}_1^{\text{LTF}}, \mathbf{T}_2^{\text{LTF}}, \dots, \mathbf{T}_M^{\text{LTF}}\}$;

- 1: Sort all tasks in a non-increasing order of their cycle utilizations;
- 2: **for** $i = 1$ **to** M **do**
- 3: $\mathbf{T}_i^{\text{LTF}} \leftarrow \emptyset$;
- 4: **end for**
- 5: **for** $j = 1$ **to** N **do**
- 6: Find the smallest w_i^{LTF} ;
- 7: $\mathbf{T}_i^{\text{LTF}} \leftarrow \mathbf{T}_i^{\text{LTF}} + \{\tau_j\}$;
- 8: **end for**
- 9: Re-order $\mathbf{T}_i^{\text{LTF}}$ by a non-decreasing order of their cycle utilization;
- 10: Return $\{\mathbf{T}_1^{\text{LTF}}, \mathbf{T}_2^{\text{LTF}}, \dots, \mathbf{T}_M^{\text{LTF}}\}$;

Algorithm 2 Double-Largest-Task-First (DLTF) strategy

Input: Tasks $\{\tau_1, \tau_2, \dots, \tau_N\}$;
Output: Task sets $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \dots, \mathbf{T}_M^{\text{DLTF}}\}$;

- 1: Execute LTF for $\{\tau_1, \tau_2, \dots, \tau_N\}$;
- 2: $w_{\max} \leftarrow \max\{s_{\text{crit}}, w_M^{\text{LTF}}\}$;
- 3: $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \dots, \mathbf{T}_M^{\text{DLTF}}\} \leftarrow \{\mathbf{T}_1^{\text{LTF}}, \mathbf{T}_2^{\text{LTF}}, \dots, \mathbf{T}_M^{\text{LTF}}\}$;
- 4: **for** $i = 1$ **to** $M - 1$ **do**
- 5: **for all** $\tau_k \in \mathbf{T}_i^{\text{DLTF}}$ **do**
- 6: **for** $j = M$ **to** $i + 1$ **do**
- 7: **if** $\frac{e_k}{p_k} + w_j^{\text{DLTF}} \leq w_{\max}$ **then**
- 8: $\mathbf{T}_j^{\text{DLTF}} \leftarrow \mathbf{T}_j^{\text{DLTF}} + \{\tau_k\}$;
- 9: Remove $\{\tau_k\}$ from $\mathbf{T}_i^{\text{DLTF}}$;
- 10: **break** for loop j ;
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: **end for**
- 15: Return $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \dots, \mathbf{T}_M^{\text{DLTF}}\}$;

for LTF and DLTF are presented in Algorithm 1 and Algorithm 2, respectively, with a worst-case time complexity of $O(MN + N \log N)$ and $O(M^2N + N \log N)$, respectively.

After partitioning tasks with DLTF, all N tasks are grouped into M task sets $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \dots, \mathbf{T}_M^{\text{DLTF}}\}$. Again, without loss of generality, task set $\mathbf{T}_i^{\text{DLTF}}$ is assigned on core i , and its *cycle utilization* is $w_i^{\text{DLTF}} = \sum_{\tau_j \in \mathbf{T}_i^{\text{DLTF}}} \frac{e_j}{p_j}$. The workload to be completed on core i during the hyper-period is $L \cdot w_i^{\text{DLTF}}$. The resulting number of cores with cycle utilization larger than 0 is defined as $M^{\neq 0}$. Defining $w_0^{\text{DLTF}} = 0$, without loss of generality, the cores are ordered such that $0 = w_0^{\text{DLTF}} \leq w_1^{\text{DLTF}} \leq w_2^{\text{DLTF}} \leq \dots \leq w_M^{\text{DLTF}}$. Furthermore, $\sum_{j=1}^N \frac{e_j}{p_j} = \sum_{i=1}^M w_i^{\text{DLTF}} = \sum_{i=1}^M w_i^*$ is denoted as the *total cycle utilization*, which does not depend on the task partitioning strategy.

For the regrouping of tasks, DLTF considers auxiliary cycle utilization $w_{\max} = \max\{s_{\text{crit}}, w_M^{\text{LTF}}\}$, and the tasks are regrouped into less cores such that after regrouping the cycle utilization among all cores in the island is no more than w_{\max} . When $w_M^{\text{DLTF}} > s_{\text{crit}}$, if after the task partitioning there is *only one* task in $\mathbf{T}_M^{\text{DLTF}}$, then $w_M^{\text{DLTF}} \leq w_M^*$, as the cycle utilization of the task in $\mathbf{T}_M^{\text{DLTF}}$ is the lower bound of the maximum utilization in the optimal task partitioning. Moreover, when $w_M^{\text{DLTF}} > s_{\text{crit}}$, if after the task partitioning there are *at least two* tasks in $\mathbf{T}_M^{\text{DLTF}}$, according to [23], [29], it holds that

$$\frac{w_1^{\text{LTF}}}{w_M^{\text{LTF}}} \geq \frac{1}{2} \quad \text{and} \quad \frac{w_M^{\text{DLTF}}}{w_M^*} \leq \theta_{\text{LTF}} = \frac{4}{3} - \frac{1}{3M}, \quad (5)$$

where θ_{LTF} is the approximation factor of LTF, due to the approximation factor of LPT for the makespan problem [9].

It has been well studied, e.g., [20], that executing core i at frequencies higher than or equal to w_i^{DLTF} with EDF guarantees that all tasks assigned to core i will meet their timing constraints. Throughout the paper we implicitly consider that $w_M^{\text{DLTF}} \leq s_{\text{max}}$; otherwise, DLTF does not derive a feasible solution. If there exist a feasible solution for the optimal task partitioning under s_{max} , i.e., $w_M^* \leq s_{\text{max}}$, but there is no feasible solution for DLTF, this infeasibility can be resolved by a resource augmentation scheme to augment speed s_{max} to $(\frac{4}{3} - \frac{1}{3M}) s_{\text{max}}$, derived from the relation between w_M^{DLTF} and w_M^* in Equation (5).

C. Problem Definition

For N periodic tasks that are assigned into a voltage island, the objective of this paper is to present and analyze a *practical* solution, that assigns the N tasks onto the M cores in the island and then applies a DVFS schedule, such that the energy consumption in the voltage island is minimized and the peak power consumption is reduced. Specifically, we consider that the tasks are partitioned using the *Double-Largest-Task-First* (DLTF) strategy, and the *Single Voltage Approximation* (SVA) scheme is used as our DVFS schedule.

More importantly, we analyze the approximation factor (worst-case behavior) of such an approach both for energy minimization and for peak power reduction, against the optimal task partitioning and optimal DVFS solution for each case, defined $\text{AF}_{\text{DLTF-SVA}}^{\text{energy}}$ and $\text{AF}_{\text{DLTF-SVA}}^{\text{peak power}}$, respectively. The approximation factor for energy minimization is

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}} = \max \frac{E_{\text{SVA}}^{\text{DLTF}}}{E_{\text{OPT}}^*} \leq \max \frac{E_{\text{SVA}}^{\text{DLTF}}}{E_{\downarrow}^*}, \quad (6)$$

where E_{OPT}^* is the optimal energy consumption for the optimal DVFS schedule and optimal task partitioning during a hyper-period, $E_{\text{SVA}}^{\text{DLTF}}$ is the energy consumption for partitioning with DLTF under SVA during a hyper-period, and E_{\downarrow}^* is a lower bound for the optimal energy consumption for the optimal task partitioning and any feasible DVFS schedule during a hyper-period. Similarly, for peak power reduction we have

$$\text{AF}_{\text{DLTF-SVA}}^{\text{peak power}} = \max \frac{\hat{P}_{\text{SVA}}^{\text{DLTF}}}{\hat{P}_{\text{OPT}}^*} \leq \max \frac{\hat{P}_{\text{SVA}}^{\text{DLTF}}}{\hat{P}_{\downarrow}^*}, \quad (7)$$

where \hat{P}_{OPT}^* is the optimal peak power consumption for the optimal DVFS schedule and optimal task partitioning, $\hat{P}_{\text{SVA}}^{\text{DLTF}}$ is the peak power consumption for partitioning with DLTF under SVA, and \hat{P}_{\downarrow}^* is a lower bound for the optimal peak power consumption for the optimal task partitioning and any feasible DVFS schedule. Since, E_{OPT}^* and \hat{P}_{OPT}^* are not easily obtained, in the analysis we use lower bounds E_{\downarrow}^* and \hat{P}_{\downarrow}^* .

Note that SVA does not require any DVFS or DPM capabilities at run time. However, to explore the approximation factor we need E_{\downarrow}^* and \hat{P}_{\downarrow}^* , in which changing the supply voltage of the island and the frequencies of the cores is with negligible overhead and the available frequencies are continuous between

$(0, s_{\text{max}}]$. This approach results in a safe lower bound for the optimal energy and peak power consumptions.

Furthermore, we would like to compare the results of DLTF-SVA against those for combining DLTF and SFA (referred to as DLTF-SFA). For such a purpose, we extend the analysis in [22], [23] by considering a more general power model, and derive a *tighter* approximation factor for DLTF-SFA for energy minimization than in [23]. Moreover, we analyze the approximation factor of DLTF-SFA in terms of peak power reduction, which is not considered in [22], [23].

IV. SINGLE VOLTAGE APPROXIMATION (SVA) SCHEME

After the task partitioning, every core i has been assigned with the corresponding task set $\mathbf{T}_i^{\text{DLTF}}$, with cycle utilization w_i^{DLTF} . Thus, under SVA, to just meet the timing constraints using EDF [20], we set the frequency of core i to w_i^{DLTF} for all $i = 1, 2, \dots, M$. In this way, the highest frequency among all cores is $s_M = w_M^{\text{DLTF}}$. In order to consume less power and to save energy, the voltage of the island is set to the minimum available voltage such that frequency w_M^{DLTF} can be stably achieved. The time complexity of SVA is $O(M)$ to ensure the feasibility, where M is the number of cores in the island and this complexity comes from evaluating the highest cycle utilization and choosing the frequency of each core.

Under SVA, the voltage of the island and the frequencies of the cores do not change at run time. Furthermore, unlike in SFA, as shown in the example in Fig. 3b, *under SVA cores do not need to be put to sleep, because $s_i = w_i^{\text{DLTF}}$ for all $i = 1, 2, \dots, M$, and thus all cores are always busy such that they just meet their timing constraints*. This means that the power consumption on every core, and thus the power consumption in the entire voltage island, is constant through the entire hyper-period, resulting in a peak power consumption which is equivalent to the average power consumption. Hence, the peak power consumption in a voltage island of DLTF-SVA, which we define as $\hat{P}_{\text{SVA}}^{\text{DLTF}}$, is expressed as

$$\hat{P}_{\text{SVA}}^{\text{DLTF}} = \alpha \cdot w_M^{\text{DLTF} \gamma - 1} \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot w_M^{\text{DLTF}} + \kappa).$$

To consider the worst case, when $w_M^{\text{DLTF}} \leq s_{\text{crit}}$ we have to assume that after the regrouping with DLTF the voltage of the island can be set according up to s_{crit} . Moreover, if $\sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}}$ DLTF will partition all tasks into a single core, and this becomes a single core DVFS problem. For such a case, from Equation (4) and Fig. 2, we know that running at slow frequencies might consume excessive energy due to the static and independent power consumption. Therefore, unless running a single core at s_{crit} already exceeds the power budget (case in which we simply run at the maximum frequency allowed by the power budget), when $\sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}}$ the core is executed at frequency s_{crit} . Finally, $\hat{P}_{\text{SVA}}^{\text{DLTF}}$ becomes

$$\hat{P}_{\text{SVA}}^{\text{DLTF}} \leq \begin{cases} \alpha \cdot s_{\text{crit}}^{\gamma} + \beta \cdot s_{\text{crit}} + \kappa & \text{if } \sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}} \\ \alpha \cdot s_{\text{crit}}^{\gamma - 1} \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot s_{\text{crit}} + \kappa) & \text{if } w_M^{\text{DLTF}} \leq s_{\text{crit}} \\ \alpha \cdot w_M^{\text{DLTF} \gamma - 1} \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot w_M^{\text{DLTF}} + \kappa) & \text{otherwise.} \end{cases} \quad (8)$$

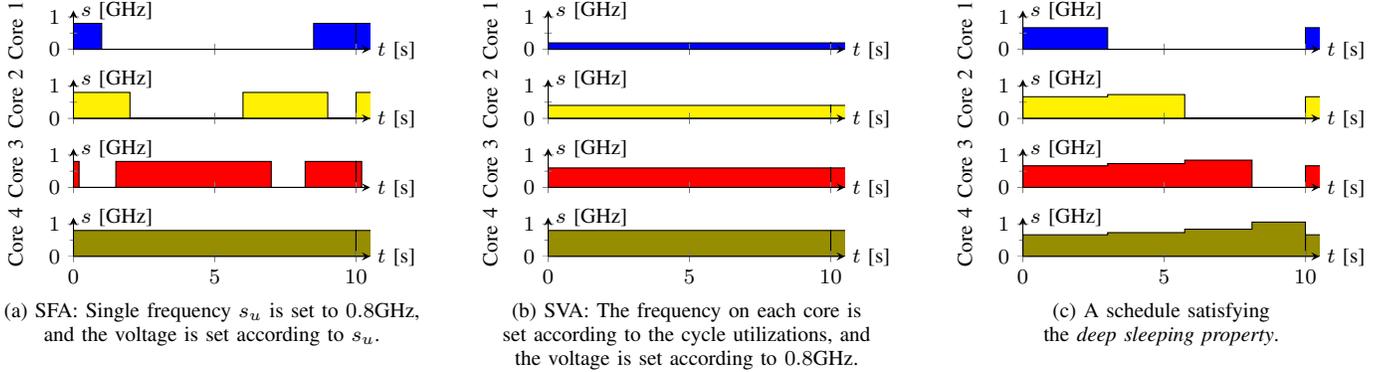


Fig. 3: Examples for 4 cores for SFA, SVA, and a schedule satisfying the *deep sleeping property*. The hyper-period is 10 seconds, and the cycle utilizations of the cores are 0.2 GHz, 0.4 GHz, 0.6 GHz, and 0.8 GHz. Under (a) SFA, cores go to sleep when they have no more workload on their ready queue. Under (b) SVA, cores just meet their timing constraints, and hence they are always busy. A speeding-up schedule like case (c) can result in the optimal solution when the voltage and frequencies are chosen such that there total power consumption is constant and the core with the highest cycle utilization is always busy.

Furthermore, considering that the workload to be completed on core i during the hyper-period is $L \cdot w_i^{\text{DLTF}}$, the total energy consumption on the island of DLTF-SVA, defined as $E_{\text{SVA}}^{\text{DLTF}}$, is

$$E_{\text{SVA}}^{\text{DLTF}} \leq \begin{cases} L \left(\alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa \right) \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{s_{\text{crit}}} & \text{if } \sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}} \\ L \left[\alpha \cdot s_{\text{crit}}^{\gamma-1} \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot s_{\text{crit}} + \kappa) \right] & \text{if } w_M^{\text{DLTF}} \leq s_{\text{crit}} \\ L \left[\alpha \cdot w_M^{\text{DLTF}\gamma-1} \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot w_M^{\text{DLTF}} + \kappa) \right] & \text{otherwise.} \end{cases} \quad (9)$$

V. LOWER BOUNDS

This section provides a lower bound for the energy consumption and for the peak power consumption for periodic real-time tasks, needed to obtain the approximation factors in Equation (6) and Equation (7). To obtain such lower bounds, we start by *unrolling* the periodic tasks in the hyper-period into frame-based real-time tasks, such that all tasks arrive at time 0 and have period and deadline equal to the hyper-period L . This is also a special case of periodic tasks, which implies that this approach is not necessarily pessimistic, as later shown in the evaluations in Section IX. Furthermore, as mentioned in Section III-C, we consider negligible overhead for changing the voltage and the frequencies, and continuous values for voltages and frequencies, which results in safe lower bounds.

A. Lower Bound for the Energy Consumption

For the special case of frame-based tasks, tasks can be scheduled according to the *deep sleeping property* [29] (every core is put to sleep after executing its workload), as shown in the example in Fig. 3c. For such a schedule, the period (also hyper-period in our case) is divided into M fragments.

During fragment i , there are $M - i + 1$ active cores (the rest of the cores are sleeping), all running at speed s_i during time t_i , with the minimum voltage such that s_i can be stably achieved. Moreover, each active core executes $c_i = L (w_i^* - w_{i-1}^*)$ core cycles during t_i , such that $t_i = \frac{c_i}{s_i}$. For a less flexible power model, the work in [22] proposes a lower bound for the energy consumption based on a schedule satisfying the *deep sleeping property*, by applying the Kuhn-Tucker conditions [25] under constraints $\sum_{i=1}^M t_i \leq L$ and $t_i \geq 0$ for $i = 1, 2, \dots, M$.

Considering our more general power model, the lower bound for the energy consumption in the voltage island during the hyper-period is

$$E_{\downarrow}^* = \sum_{i=1}^M (M - i + 1) \left(\alpha \cdot \frac{c_i^\gamma}{t_i^\gamma} + \beta \cdot \frac{c_i}{t_i} + \kappa \right) t_i. \quad (10)$$

Similar to [22], we apply the Kuhn-Tucker conditions [25] to Equation (10) under constraints $\sum_{i=1}^M t_i \leq L$ and $t_i \geq 0$ for $i = 1, 2, \dots, M$. Due to space constraints, the details are omitted. Once the Lagrangian is solved, the set of t_i that minimizes the energy consumption for the lower bound is

$$t_i = \sqrt[\gamma]{\frac{\alpha(\gamma-1)(M-i+1)}{(M-i+1)\kappa+\lambda}} c_i, \quad (11)$$

which coincides with Equation (7) in [22], and thus the same conclusions apply. Namely, when $\sum_{i=1}^M t_i < L$, then $\lambda = 0$, and all the active cores run at frequency s_{crit} , which is a feasible solution when $w_M^* \leq s_{\text{crit}}$. Furthermore, when $\sum_{i=1}^M t_i = L$, then $\lambda > 0$, and is no longer feasible to meet the timing constraints by running all cores at s_{crit} . Hence, from Equation (11), it holds that

$$\sum_{i=1}^M t_i = \sum_{i=1}^M \sqrt[\gamma]{\frac{\alpha(\gamma-1)(M-i+1)}{(M-i+1)\kappa+\lambda}} c_i = L, \quad (12)$$

in which for a specific case study, since the only unknown variable is λ and Equation (12) is strictly decreasing with respect to λ , one possibility to derive λ is to apply Newton's method. However, there is no explicit form to solve

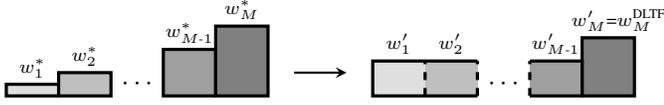


Fig. 4: Example of cycle utilization adjustment from $w_1^*, w_2^*, \dots, w_M^*$ to w'_1, w'_2, \dots, w'_M , when $w_M^* \geq w_M^{\text{DLTF}}$.

Equation (12) arithmetically. Therefore, we approximate E_{\downarrow}^* by defining an auxiliary frequency denoted as s_{dyn} , such that $s_{\text{crit}} < s_{\text{dyn}} < s_{\text{max}}$. When $w_M^* \leq s_{\text{dyn}}$, then E_{\downarrow}^* is approximated by considering that all cores run at s_{crit} . When $w_M^* > s_{\text{dyn}}$, then E_{\downarrow}^* is approximated by considering that $\kappa = 0$, such that Equation (12) can be solved arithmetically, resulting in

$$t_i = \frac{L(w_i^* - w_{i-1}^*) \sqrt[M-i+1]{M-i+1}}{\sum_{j=1}^M (w_j^* - w_{j-1}^*) \sqrt[M-j+1]{M-j+1}} \quad \text{and} \quad s_i = \frac{\sum_{j=1}^M (w_j^* - w_{j-1}^*) \sqrt[M-j+1]{M-j+1}}{\sqrt[M-i+1]{M-i+1}}. \quad (13)$$

Moreover, it holds that $\sum_{i=1}^M (w_i^* - w_{i-1}^*) (M - i + 1)$ is equal to $\sum_{i=1}^M w_i^*$. Thus, extending the results from [22] with our more general power model, the lower bound for the energy consumption is

$$E_{\downarrow}^* = \begin{cases} L(\alpha \cdot s_{\text{crit}}^{\gamma} + \beta \cdot s_{\text{crit}} + \kappa) \frac{\sum_{i=1}^M w_i^*}{s_{\text{crit}}} & \text{if } w_M^* \leq s_{\text{dyn}} \\ L\alpha \left[\sum_{i=1}^M (w_i^* - w_{i-1}^*) \sqrt[M-i+1]{M-i+1} \right]^{\gamma} + L\beta \sum_{i=1}^M w_i^* & \text{otherwise.} \end{cases} \quad (14)$$

Furthermore, for a given total cycle utilization, it was proven in [23] that the energy consumption $L\alpha \left[\sum_{i=1}^M (w_i^* - w_{i-1}^*) \sqrt[M-i+1]{M-i+1} \right]^{\gamma}$ can be reduced by an adjustment in the cycle utilizations of the optimal task partitioning. Specifically, starting from $w_1^*, w_2^*, \dots, w_M^*$, the energy consumption is further reduced by using w'_1, w'_2, \dots, w'_M , defined in Equation (15). Fig. 4 shows an example of this adjustment.

$$w'_1 = w'_2 = \dots = w'_{M-1} = \frac{\sum_{i=1}^M w_i^* - w'_M}{M-1} \quad (15)$$

$$w'_M = \begin{cases} w_M^{\text{DLTF}} & \text{if } w_M^* \geq w_M^{\text{DLTF}} \\ \max \left\{ \frac{w_M^{\text{DLTF}}}{\theta_{\text{LTF}}}, \frac{\sum_{i=1}^M w_i^*}{M} \right\} & \text{if } w_M^* < w_M^{\text{DLTF}} \end{cases}$$

For our lower bound in Equation (14), when $w_M^* \leq s_{\text{dyn}}$ using this adjustment does not change the energy consumption because the *total cycle utilization* is constant, that is, $\sum_{i=1}^M w_i^* = \sum_{i=1}^M w'_i$. Moreover, when $w_M^* > s_{\text{dyn}}$, given that in our lower bound in Equation (14) we have the same energy consumption expression that in [23] plus a factor depending on $\sum_{i=1}^M w_i^*$, this cycle utilization adjustment also reduces the lower bound of the energy consumption for the more general power model. Hence, E_{\downarrow}^* becomes

$$E_{\downarrow}^* = \begin{cases} L(\alpha \cdot s_{\text{crit}}^{\gamma} + \beta \cdot s_{\text{crit}} + \kappa) \frac{\sum_{i=1}^M w'_i}{s_{\text{crit}}} & \text{if } w_M^* \leq s_{\text{dyn}} \\ L\alpha \cdot w'_M{}^{\gamma} \left[1 + \frac{w'_1}{w'_M} (\sqrt[M]{M} - 1) \right]^{\gamma} + L\beta \sum_{i=1}^M w'_i & \text{otherwise,} \end{cases} \quad (16)$$

and Fig. 5 presents an example for E_{\downarrow}^* .

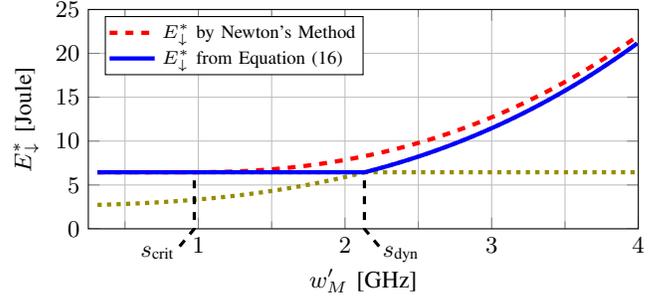


Fig. 5: E_{\downarrow}^* from Equation (16), and by using Newton's Method for Equation (12), with $\gamma = 3$, $\alpha = 0.27 \frac{\text{W}}{\text{GHz}^3}$, $\beta = 0.52 \frac{\text{W}}{\text{GHz}}$, $\kappa = 0.5 \text{ W}$, $M = 16$, $L = 1 \text{ sec.}$, constant $\sum_{i=1}^M w_i^* = 5 \cdot 10^9 \frac{\text{cycles}}{\text{second}}$, and $w'_1 = w'_2 = \dots = w'_{M-1} = \frac{\sum_{i=1}^M w_i^* - w'_M}{M-1}$.

We can derive an important conclusion from Fig. 5. That is, given that E_{\downarrow}^* is in the denominator of Equation (6), in order to derive an approximation factor without unnecessary pessimism, the value of s_{dyn} in Equation (16) should be chosen such that E_{\downarrow}^* becomes a continuous function. Hence, in Lemma 1 we choose s_{dyn} for this to hold.

Lemma 1: Equation (16) is a continuous function when

$$s_{\text{dyn}} = s_{\text{crit}} \cdot [\gamma \cdot h(\delta)]^{\frac{1}{\gamma-1}},$$

where $\delta = \frac{\sum_{i=1}^{M-1} w'_i}{w'_M (M-1)}$, $\delta^{\text{max}} = \frac{\gamma-1+M-\gamma \sqrt[M]{M}}{(\gamma-1)(M \sqrt[M]{M} - M - \sqrt[M]{M+1})}$, and

$$h(\delta) = \frac{1 - \delta + \delta M}{(1 - \delta + \delta \sqrt[M]{M})^{\gamma}} \leq h(\delta^{\text{max}}).$$

Proof: For Equation (16) to be continuous, we match both parts and find the w'_M for which the equality holds. That is,

$$\alpha \cdot w'_M{}^{\gamma} [1 - \delta(\sqrt[M]{M} - 1)]^{\gamma} + \beta \sum_{i=1}^M w'_i = (\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta) \sum_{i=1}^M w'_i$$

and then, since $\sum_{i=1}^M w'_i = w'_M [1 + \delta(M-1)]$, we have

$$w'_M{}^{\gamma-1} = \frac{\gamma \cdot s_{\text{crit}}^{\gamma-1} (1 - \delta + \delta M)}{[1 - \delta(\sqrt[M]{M} - 1)]^{\gamma}} = s_{\text{crit}}^{\gamma-1} \cdot \gamma \cdot h(\delta),$$

where this w'_M is s_{dyn} . Finally, since $h(\delta)$ is a convex function of δ when $\gamma > 1$, by taking the first order derivative of $h(\delta)$ with respect to δ , its maximum value happens when δ is δ^{max} . Thus, the lemma is proven. ■

B. Lower Bound for the Peak Power Consumption

Given that energy is the integration of power through time, when $w_M^{\text{DLTF}} \geq s_{\text{crit}}$, minimizing the energy consumption while satisfying the timing constraints is equivalent to minimizing the average power consumption while also satisfying the timing constraints. This means that the lower bound for the peak power consumption is found when the power consumption is constant for the entire hyper-period, and equivalent to the minimum average power consumption. Namely, when $w_M^{\text{DLTF}} \geq s_{\text{crit}}$, we have that $E_{\downarrow}^* = \hat{P}_{\downarrow}^* \cdot L$.

Nevertheless, the *critical frequency* is not involved when talking about power consumption, and running at slower voltages and frequencies (as long as the timing constraints allow it) always results in smaller power consumption values. Therefore, we only consider the part of Equation (16) that does not involve s_{crit} . Moreover, given that *at least one core* will be active, there is at least κ power consumption. Thus, \hat{P}_{\downarrow}^* is

$$\hat{P}_{\downarrow}^* = \alpha \cdot w'_M{}^\gamma \left[1 + \frac{w'_1}{w'_M} \left(\sqrt[M]{M} - 1 \right) \right]^\gamma + \beta \sum_{i=1}^M w'_i + \kappa. \quad (17)$$

VI. APPROXIMATION FACTOR ANALYSIS: DLTF-SVA

A. Energy Minimization Analysis for DLTF-SVA

This subsection presents the approximation factor analysis of DLTF-SVA in terms of energy consumption. That is, the worst-case behavior of such an approach for energy minimization, against the optimal task partitioning and optimal DVFS solution, defined as $\text{AF}_{\text{DLTF-SVA}}^{\text{energy}}$. For such a purpose, we first derive Lemma 2, in which we find an upper bound of $M^{\neq 0}$ when using DLTF with respect to the total cycle utilization.

Lemma 2: For a given w_M^{DLTF} and total cycle utilization $\sum_{i=1}^M w_i^{\text{DLTF}}$, an upper bound of $M^{\neq 0}$ with DLTF is

$$M^{\neq 0} \leq \begin{cases} \min \left\{ M, 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{s_{\text{crit}}} \right\} & \text{if } w_M^{\text{DLTF}} \leq s_{\text{crit}} \\ \min \left\{ M, 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{w_M^{\text{DLTF}}} - 1 \right\} & \text{if } w_M^{\text{DLTF}} > s_{\text{crit}}. \end{cases}$$

Proof: Algorithm DLTF is mainly a first-fit bin packing algorithm, in which, from the initial solution of LTF, we try to pack the tasks into the minimum amount of bins of size $w_{\text{max}} = \max \{s_{\text{crit}}, w_M^{\text{LTF}}\}$. This implies that after the regrouping, if in the initial solution of LTF $w_M^{\text{LTF}} > s_{\text{crit}}$, then task set M is unchanged, such that $\mathbf{T}_M^{\text{DLTF}} = \mathbf{T}_M^{\text{LTF}}$ and $w_M^{\text{DLTF}} = w_M^{\text{LTF}} > s_{\text{crit}}$. Moreover, if $w_M^{\text{LTF}} \leq s_{\text{crit}}$, then most likely $\mathbf{T}_M^{\text{DLTF}} \neq \mathbf{T}_M^{\text{LTF}}$, but it will hold that $w_M^{\text{DLTF}} \leq s_{\text{crit}}$. From the properties of the first-fit bin packing algorithm [27], it holds that $M^{\neq 0} \leq \min \left\{ M, 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{s_{\text{crit}}} \right\}$ if $w_M^{\text{DLTF}} \leq s_{\text{crit}}$, and $M^{\neq 0} \leq \min \left\{ M, 1 + 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}} - w_M^{\text{DLTF}}}{w_M^{\text{DLTF}}} \right\}$ if $w_M^{\text{DLTF}} > s_{\text{crit}}$. Thus, the lemma is proven. ■

From Equation (6), Equation (9), and Equation (16), we can express $\text{AF}_{\text{DLTF-SVA}}^{\text{energy}}$ as

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}} \leq \max \left\{ \begin{array}{l} 1 \\ \frac{\alpha \cdot s_{\text{crit}}^{\gamma-1} \cdot \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot s_{\text{crit}} + \kappa)}{(\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta) \sum_{i=1}^M w'_i} \quad \text{if } \sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}} \\ \frac{\alpha \cdot w_M^{\text{DLTF} \gamma-1} \cdot \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot w_M^{\text{DLTF}} + \kappa)}{(\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta) \sum_{i=1}^M w'_i} \quad \text{if } w_M^{\text{DLTF}} > s_{\text{crit}} \\ \frac{\alpha \cdot w_M^{\text{DLTF} \gamma-1} \cdot \sum_{i=1}^M w_i^{\text{DLTF}} + M^{\neq 0} (\beta \cdot w_M^{\text{DLTF}} + \kappa)}{\alpha \cdot w'_M{}^\gamma \left[1 + \frac{w'_1}{w'_M} \left(\sqrt[M]{M} - 1 \right) \right]^\gamma + \beta \sum_{i=1}^M w'_i} \quad \text{otherwise.} \end{array} \right. \quad (18)$$

Considering Lemma 1, and Lemma 2, the following lemmas present the worst-cases of Equation (18) for the different conditions. Theorem 1 then summarizes these lemmas to present the final approximation factor.

Lemma 3: The approximation factor for energy minimization of DLTF-SVA, when $w_M^{\text{DLTF}} \leq s_{\text{crit}}$, $w_M^* \leq s_{\text{dyn}}$, and $\sum_{i=1}^M w_i^{\text{DLTF}} > s_{\text{crit}}$, is expressed as

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*1} \leq \frac{\alpha \cdot s_{\text{crit}}^{\gamma-1} + \min \{M, 2\} \left(\beta + \frac{\kappa}{s_{\text{crit}}} \right)}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta}$$

Proof: For this case, replacing $M^{\neq 0}$ from Lemma 2 in Equation (18), we have

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*1} \leq \frac{\alpha \cdot s_{\text{crit}}^{\gamma-1} + \min \left\{ \frac{M}{\sum_{i=1}^M w_i^{\text{DLTF}}}, \frac{2}{s_{\text{crit}}} \right\} (\beta \cdot s_{\text{crit}} + \kappa)}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta},$$

which given that α , β , κ , γ , and s_{crit} are constants, it is maximized if and only if the *min* function is maximized. Clearly, this happens when $\sum_{i=1}^M w_i^{\text{DLTF}} \rightarrow s_{\text{crit}}$, reaching the expression in the statement of the lemma. ■

Lemma 4: The approximation factor for energy minimization of DLTF-SVA, when $w_M^{\text{DLTF}} > s_{\text{crit}}$, $w_M^* \leq s_{\text{dyn}}$, and $w_M^* \geq w_M^{\text{DLTF}}$, is expressed as

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*2} \leq \frac{\alpha \cdot \gamma \cdot h(\delta) \cdot s_{\text{crit}}^{\gamma-1} + \frac{\min \{M, 1 + 2(M-1)\delta\}}{1 - \delta + \delta M} \left(\beta + \frac{\kappa}{s_{\text{crit}} \cdot [\gamma \cdot h(\delta)]^{\frac{1}{\gamma-1}}} \right)}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta}.$$

For every M , we compute the maximum $\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*2}$ among all δ , such that $0 \leq \delta \leq 1$.

Proof: For this case, inside Equation (18), we replace $M^{\neq 0}$ by $\min \left\{ M, 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{w_M^{\text{DLTF}}} - 1 \right\}$ from Lemma 2, w_M^{DLTF} by w'_M from Equation (15), and $\sum_{i=1}^M w_i^{\text{DLTF}}$ by $w'_M (1 - \delta + \delta M)$. Moreover, we replace w'_M by s_{dyn} from Lemma 1, such that E_{\downarrow}^* becomes a continuous function. Thus, we reach the expression in the statement of the lemma. ■

Lemma 5: The approximation factor for energy minimization of DLTF-SVA, when $w_M^{\text{DLTF}} > s_{\text{crit}}$, $w_M^* \leq s_{\text{dyn}}$, and $w_M^* < w_M^{\text{DLTF}}$, is expressed as

$$\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*3} \leq \frac{\alpha \cdot \gamma \cdot h(\delta) \cdot (\theta_{\text{LTF}} \cdot s_{\text{crit}})^{\gamma-1} + \frac{M \cdot \beta \cdot \theta_{\text{LTF}} + \frac{M \cdot \kappa}{1 - \delta + \delta M}}{s_{\text{crit}} \cdot [\gamma \cdot h(\delta)]^{\frac{1}{\gamma-1}}}}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta},$$

For every M , we compute the maximum $\text{AF}_{\text{DLTF-SVA}}^{\text{energy}^*3}$ among all δ , such that $\frac{4M+1}{6M} \leq \delta \leq 1$.

Proof: Similar to the proof of Lemma 4, inside Equation (18), we replace $\sum_{i=1}^M w_i^{\text{DLTF}}$ by $w'_M (1 - \delta + \delta M)$, and we replace w'_M by s_{dyn} from Lemma 1 such that E_{\downarrow}^* becomes a continuous function. From Equation (5) we have that $M \cdot w_M^{\text{DLTF}} \geq \sum_{i=1}^M w_i^{\text{DLTF}} \geq \frac{M+1}{2} \cdot w_M^{\text{DLTF}}$. Hence, it holds that $M \leq 2 \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{w_M^{\text{DLTF}}} - 1$ for all $M \geq 1$, and $M^{\neq 0}$ is set to M . Furthermore, from Equation (15) we have that $w'_M = \max \left\{ \frac{w_M^{\text{DLTF}}}{\theta_{\text{LTF}}}, \frac{\sum_{i=1}^M w_i^*}{M} \right\}$, in which case $\frac{\sum_{i=1}^M w_i^*}{M}$ is only considered if the cycle utilization adjustment in from Equation (15) does not reach $\frac{w_M^{\text{DLTF}}}{\theta_{\text{LTF}}}$ without reducing w'_M below the

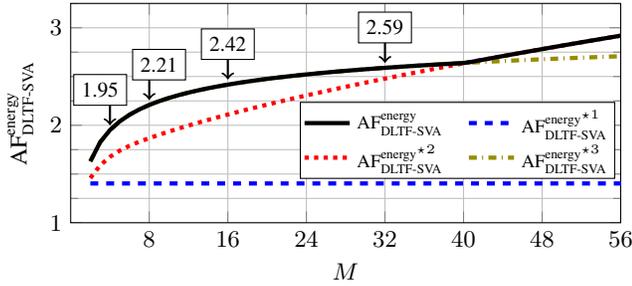


Fig. 6: Example of $AF_{DLTF-SVA}^{energy}$ when $\gamma = 3$, $\alpha = 0.27 \frac{W}{GHz^3}$, $\beta = 0.52 \frac{W}{GHz}$, and $\kappa = 0.5 W$.

average cycle utilization. This means that $w'_M = \frac{w_M^{DLTF}}{\theta_{LTF}}$ is the worst case for the relation between w'_M and w_M^{DLTF} , and the other case is not considered. Finally, from Lemma 5 in [23], it holds that $\delta = \frac{\sum_{i=1}^{M-1} w'_i}{w'_M(M-1)} \geq \frac{4M+1}{6M}$. Thus, the lemma is proven. ■

Theorem 1: The approximation factor for energy minimization of DLTF-SVA, against the optimal task partitioning and optimal DVFS solution, is expressed as

$$AF_{DLTF-SVA}^{energy} \leq \max \left\{ AF_{DLTF-SVA}^{energy*1}, AF_{DLTF-SVA}^{energy*2}, AF_{DLTF-SVA}^{energy*3} \right\},$$

according to the definitions in Lemma 3, 4, and 5.

Proof: This comes just from taking the maximum among all cases from Lemma 3, Lemma 4, and Lemma 5. ■

Fig. 6 shows an example of $AF_{DLTF-SVA}^{energy}$, for a power function modeled from simulations conducted on gem5 [3] and McPAT [19] for a 22 nm *out-of-order* Alpha 21264 core, which results in power parameters $\gamma = 3$, $\alpha = 0.27 \frac{W}{GHz^3}$, $\beta = 0.52 \frac{W}{GHz}$, and $\kappa = 0.5 W$ (details in Section III-A). For the given hardware parameters, Fig. 6 shows that the approximation factor in terms of energy consumption of DLTF-SVA is at most 1.95 (2.21, 2.42, 2.59, respectively), when the voltage island has up to 4 (8, 16, 32, respectively) cores.

B. Peak Power Reduction Analysis for DLTF-SVA

This subsection presents the approximation factor analysis for DLTF-SVA in terms of peak power reduction. That is, the worst-case behavior of such an approach for peak power reduction, against the optimal task partitioning and optimal DVFS solution. From Equation (7), Equation (8), and Equation (17), this is

$$AF_{DLTF-SVA}^{peak power} \leq \max \left\{ \begin{array}{l} \frac{\alpha \cdot s_{crit}^\gamma + \beta \cdot s_{crit} + \kappa}{\alpha \cdot w'_M{}^\gamma \left[1 + \frac{w'_M}{w'_M} (\sqrt[M]{M} - 1) \right]^\gamma + \beta \sum_{i=1}^M w'_i + \kappa} \quad \text{if } \sum_{i=1}^M w_i^{DLTF} \leq s_{crit} \\ \frac{\alpha \cdot s_{crit}^{\gamma-1} \sum_{i=1}^M w_i^{DLTF} + M \neq 0 (\beta \cdot s_{crit} + \kappa)}{\alpha \cdot w'_M{}^\gamma \left[1 + \frac{w'_M}{w'_M} (\sqrt[M]{M} - 1) \right]^\gamma + \beta \sum_{i=1}^M w'_i + \kappa} \quad \text{if } w_M^{DLTF} \leq s_{crit} \\ \frac{\alpha \cdot w_M^{DLTF \gamma-1} \sum_{i=1}^M w_i^{DLTF} + M \neq 0 (\beta \cdot w_M^{DLTF} + \kappa)}{\alpha \cdot w'_M{}^\gamma \left[1 + \frac{w'_M}{w'_M} (\sqrt[M]{M} - 1) \right]^\gamma + \beta \sum_{i=1}^M w'_i + \kappa} \quad \text{otherwise.} \end{array} \right. \quad (19)$$

The following lemmas present the worst cases of Equation (19) for the different conditions. Theorem 2 then summarizes these lemmas to present the final approximation factor.

Lemma 6: The approximation factor for peak power reduction of DLTF-SVA, when $\sum_{i=1}^M w_i^{DLTF} \leq s_{crit}$, is expressed as

$$AF_{DLTF-SVA}^{peak power*1} \leq \frac{\alpha \cdot s_{crit}^\gamma + \beta \cdot s_{crit} + \kappa}{\kappa}.$$

Proof: For this case, since in Equation (19) $\alpha \cdot s_{crit}^\gamma + \beta \cdot s_{crit} + \kappa$ is constant, the worst case happens when $\sum_{i=1}^M w_i^{DLTF} \rightarrow 0$, such that there is only independent power consumption in the lower bound, and the lemma is proven. ■

Lemma 7: The approximation factor for peak power reduction of DLTF-SVA, when $\sum_{i=1}^M w_i^{DLTF} > s_{crit}$ and $w_M^{DLTF} \leq s_{crit}$, is expressed as

$$AF_{DLTF-SVA}^{peak power*2} \leq \frac{\alpha \cdot s_{crit}^{\gamma-1} + \min \left\{ \frac{M}{w'_M}, \frac{2(1-\delta+\delta M)}{s_{crit}} \right\} \frac{\beta \cdot s_{crit} + \kappa}{1-\delta+\delta M}}{\alpha \cdot w'_M{}^{\gamma-1} \cdot \frac{1}{h(\delta)} + \beta + \frac{\kappa}{w'_M(1-\delta+\delta M)}}.$$

For every M , we compute the maximum $AF_{DLTF-SVA}^{peak power*2}$ among all δ and all w'_M , such that $0 \leq \delta \leq 1$ and $\frac{s_{crit}}{1-\delta+\delta M} < w'_M \leq s_{crit}$.

Proof: For this case, inside Equation (19), we replace $\sum_{i=1}^M w_i^{DLTF}$ with $w'_M(1-\delta+\delta M)$, $M \neq 0$ from Lemma 2, and δ and $h(\delta)$ from Lemma 1. Thus, we reach the expression in the statement of the lemma and the lemma is proven. ■

Lemma 8: The approximation factor for peak power reduction of DLTF-SVA, when $w_M^{DLTF} > s_{crit}$ and $w_M^* \geq w_M^{DLTF}$, is expressed as

$$AF_{DLTF-SVA}^{peak power*3} \leq \frac{\alpha \cdot w'_M{}^\gamma + \min \{ M, 1 + 2(M-1)\delta \} \frac{\beta \cdot w'_M + \kappa}{1-\delta+\delta M}}{\alpha \cdot w'_M{}^\gamma \cdot \frac{1}{h(\delta)} + \beta \cdot w'_M + \frac{\kappa}{1-\delta+\delta M}}.$$

For every M , we compute the maximum $AF_{DLTF-SVA}^{peak power*3}$ among all δ and all w'_M , such that $0 \leq \delta \leq 1$ and $s_{crit} < w'_M \leq s_{max}$.

Proof: For this case, inside Equation (19), we replace $\sum_{i=1}^M w_i^{DLTF}$ with $w'_M(1-\delta+\delta M)$, $M \neq 0$ from Lemma 2, and δ and $h(\delta)$ from Lemma 1. Furthermore, from Equation (15) we have that $w'_M = w_M^{DLTF}$. Thus, we reach the expression in the statement of the lemma. ■

Lemma 9: The approximation factor for peak power reduction of DLTF-SVA, when $w_M^{DLTF} > s_{crit}$ and $w_M^* < w_M^{DLTF}$, is expressed as

$$AF_{DLTF-SVA}^{peak power*4} \leq \frac{\alpha \cdot \theta_{LTF}^{\gamma-1} \cdot w'_M{}^\gamma + M \cdot \frac{\beta \cdot \theta_{LTF} \cdot w'_M + \kappa}{1-\delta+\delta M}}{\alpha \cdot w'_M{}^\gamma \cdot \frac{1}{h(\delta)} + \beta \cdot w'_M + \frac{\kappa}{1-\delta+\delta M}}.$$

For every M , we compute the maximum $AF_{DLTF-SVA}^{peak power*4}$ among all δ and w'_M , such that $\frac{4M+1}{6M} \leq \delta \leq 1$ and $\frac{s_{crit}}{\theta_{LTF}} < w'_M \leq s_{max}$.

Proof: For this case, inside Equation (19), we replace $\sum_{i=1}^M w_i^{DLTF}$ with $w'_M(1-\delta+\delta M)$, $M \neq 0$ from Lemma 2, and δ and $h(\delta)$ from Lemma 1. Moreover, from Equation (15) and Lemma 5 in [23], we have that $w'_M = \max \left\{ \frac{w_M^{DLTF}}{\theta_{LTF}}, \frac{\sum_{i=1}^M w_i}{M} \right\}$, and $\delta = \frac{\sum_{i=1}^{M-1} w'_i}{w'_M(M-1)} \geq \frac{4M+1}{6M}$. As shown in Lemma 5, for this case it holds that $M \leq 2 \frac{\sum_{i=1}^M w_i^{DLTF}}{w_M^{DLTF}} - 1$ for

all $M \geq 1$, and $w'_M = \frac{w_M^{\text{DLTF}}}{\theta_{\text{LTF}}}$ is the worst case for the relation between w'_M and w_M^{DLTF} . Thus, the lemma is proven. ■

Theorem 2: The approximation factor for peak power reduction of DLTF-SVA, against the optimal task partitioning and optimal DVFS solution, is expressed as

$$\text{AF}_{\text{DLTF-SVA}}^{\text{peak power}} \leq \max \left\{ \text{AF}_{\text{DLTF-SVA}}^{\text{peak power}*1}, \text{AF}_{\text{DLTF-SVA}}^{\text{peak power}*2}, \right. \\ \left. \text{AF}_{\text{DLTF-SVA}}^{\text{peak power}*3}, \text{AF}_{\text{DLTF-SVA}}^{\text{peak power}*4} \right\},$$

according to the definitions in Lemma 6, 7, 8, and 9.

Proof: This comes from taking the maximum among all cases from Lemma 6, Lemma 7, Lemma 8, and Lemma 9. ■

VII. COMPARING DLTF-SVA AGAINST DLTF-SFA

This section compares the worst-case efficiency of DLTF-SVA against the worst-case efficiency of DLTF-SFA. Given that in this paper we use a more general power model than the one used in [22], [23], in order to have a fair comparison we must first extend the analysis from [22], [23]. Without such an extension, it would be possible to draw wrong conclusions, due to comparing approximation factors for different power models. Furthermore, [22], [23] do not include the analysis of peak power reduction for SFA.

The frequencies used by SFA are (1) s_{crit} if $w_M^{\text{DLTF}} \leq s_{\text{crit}}$, and (2) w_M^{DLTF} otherwise. Thus, the energy consumption in the island for combining DLTF and SFA during a hyper-period is

$$E_{\text{SFA}}^{\text{DLTF}} = \begin{cases} L(\alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa) \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{s_{\text{crit}}} & \text{if } w_M^{\text{DLTF}} \leq s_{\text{crit}} \\ L(\alpha \cdot w_M^{\text{DLTF}\gamma} + \beta \cdot w_M^{\text{DLTF}} + \kappa) \frac{\sum_{i=1}^M w_i^{\text{DLTF}}}{w_M^{\text{DLTF}}} & \text{otherwise.} \end{cases}$$

The peak power consumption in the island occurs when all cores execute tasks simultaneously, e.g., at the beginning of every hyper-period when at least one task in each core has arrival time zero, as shown in Fig. 3a. Thus, the peak power consumption in the island for combining DLTF and SFA is

$$\hat{P}_{\text{SFA}}^{\text{DLTF}} = \begin{cases} \alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa & \text{if } \sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}} \\ M \neq 0 (\alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa) & \text{if } w_M^{\text{DLTF}} \leq s_{\text{crit}} \\ M \neq 0 (\alpha \cdot w_M^{\text{DLTF}\gamma} + \beta \cdot w_M^{\text{DLTF}} + \kappa) & \text{otherwise.} \end{cases}$$

A. Energy Minimization Analysis for DLTF-SFA

The approximation factor for combining DLTF and SFA is presented in Theorem 3 and Theorem 4.

Theorem 3: The approximation factor for energy minimization of DLTF-SFA, when there is *negligible overhead for sleeping*, against the optimal task partitioning and optimal DVFS solution, is expressed as

$$\text{AF}_{\text{DLTF-SFA}}^{\text{energy}} \leq \max \left\{ \frac{\alpha \cdot \gamma \cdot h(\delta^{\max}) \cdot s_{\text{crit}}^{\gamma-1} + \beta + \frac{\kappa}{s_{\text{crit}}[\gamma \cdot h(\delta^{\max})]^{\gamma-1}}}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta}, \right. \\ \left. \frac{\alpha \cdot \gamma \cdot h\left(\frac{4M+1}{6M}\right) \cdot (\theta_{\text{LTF}} \cdot s_{\text{crit}})^{\gamma-1} + \beta + \frac{\kappa}{\theta_{\text{LTF}} \cdot s_{\text{crit}} \left[\gamma \cdot h\left(\frac{4M+1}{6M}\right)\right]^{\gamma-1}}}{\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta} \right\}.$$

Proof: We divide $E_{\text{SFA}}^{\text{DLTF}}$ by E_{\downarrow}^* from Equation (16) to express the energy consumption ratio. Then, similar to SVA, we replace w_M^{DLTF} according to the corresponding relation with w'_M from Equation (15), and we replace w'_M by s_{dyn} from Lemma 1. Given that $(\alpha \cdot \gamma \cdot s_{\text{crit}}^{\gamma-1} + \beta) \sum_{i=1}^M w'_i$ in Equation (16) is constant for a given $\sum_{i=1}^M w'_i$, then the worst case for the approximation factor happens when s_{dyn} is maximized. For the case that $w_M^* < w_M^{\text{DLTF}}$, from Lemma 5 in [23] it holds that $\delta = \frac{\sum_{i=1}^{M-1} w'_i}{w'_M(M-1)} \geq \frac{4M+1}{6M}$. Thus, the theorem is proven. ■

Theorem 4: The approximation factor for energy minimization of DLTF-SFA, when there is *non-negligible overhead for sleeping*, against the optimal task partitioning and optimal DVFS solution, is expressed as

$$\text{AF}_{\text{DLTF-SFA}}^{\text{energy}} \leq \text{AF}_{\text{DLTF-SFA}}^{\text{energy}} + \frac{\gamma - 1}{\gamma}.$$

(no overheads)

Proof: When there is *negligible overhead for sleeping*, if $\beta = 0$, the result from Theorem 3 coincides with that of Theorem 4 in [23]. This happens because $\beta = 0$ is actually the worst case. Hence, when $\beta > 0$ Theorem 5 of [23] still holds, proving the theorem. ■

B. Peak Power Reduction Analysis for DLTF-SFA

The analysis is similar to that for SVA. We first divide $\hat{P}_{\text{SFA}}^{\text{DLTF}}$ by \hat{P}_{\downarrow}^* from Equation (17), and then consider all cases. The following lemmas present the worst cases for the different conditions. Theorem 5 then summarizes these lemmas.

Lemma 10: The approximation factor $\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}}$, when $\sum_{i=1}^M w_i^{\text{DLTF}} \leq s_{\text{crit}}$, is expressed as

$$\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}*1} \leq \frac{\alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa}{\kappa}.$$

Proof: This comes from the proof of Lemma 6. ■

Lemma 11: The approximation factor $\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}}$, when $\sum_{i=1}^M w_i^{\text{DLTF}} > s_{\text{crit}}$ and $w_M^{\text{DLTF}} \leq s_{\text{crit}}$, is expressed as

$$\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}*2} \leq \frac{\min \left\{ \frac{M}{w'_M}, \frac{2(1-\delta+\delta M)}{s_{\text{crit}}} \right\} \frac{\alpha \cdot s_{\text{crit}}^\gamma + \beta \cdot s_{\text{crit}} + \kappa}{1-\delta+\delta M}}{\alpha \cdot w'_M{}^{\gamma-1} \cdot \frac{1}{h(\delta)} + \beta + \frac{\kappa}{w'_M(1-\delta+\delta M)}}.$$

For every M , we compute the maximum $\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}*2}$ among all δ and all w'_M , such that $0 \leq \delta \leq 1$ and $\frac{s_{\text{crit}}}{1-\delta+\delta M} < w'_M \leq s_{\text{crit}}$.

Proof: The proof is similar to the proof of Lemma 7. ■

Lemma 12: The approximation factor $\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}}$, when $w_M^{\text{DLTF}} > s_{\text{crit}}$ and $w_M^* \geq w_M^{\text{DLTF}}$, is expressed as

$$\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}*3} \leq \frac{\min \{M, 1 + 2(M-1)\delta\} \frac{\alpha \cdot w'_M{}^\gamma + \beta \cdot w'_M + \kappa}{1-\delta+\delta M}}{\alpha \cdot w'_M{}^\gamma \cdot \frac{1}{h(\delta)} + \beta \cdot w'_M + \frac{\kappa}{1-\delta+\delta M}}.$$

For every M , we compute the maximum $\text{AF}_{\text{DLTF-SFA}}^{\text{peak power}*3}$ among all δ and all w'_M , such that $0 \leq \delta \leq 1$ and $s_{\text{crit}} < w'_M \leq s_{\text{max}}$.

Proof: The proof is similar to the proof of Lemma 8. ■

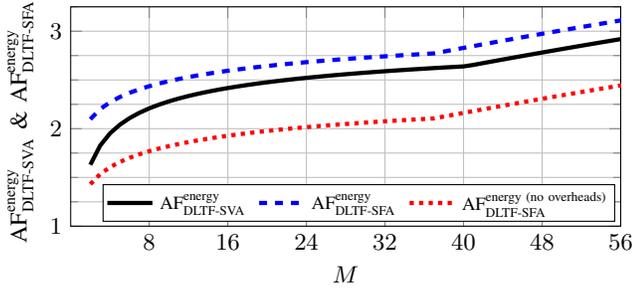


Fig. 7: Example of $AF_{DLTF-SVA}^{energy}$ and $AF_{DLTF-SFA}^{energy}$ when $\gamma = 3$, $\alpha = 0.27 \frac{W}{GHz^3}$, $\beta = 0.52 \frac{W}{GHz}$, and $\kappa = 0.5 W$.

Lemma 13: The approximation factor $AF_{DLTF-SFA}^{peak power}$, when $w_M^{DLTF} > s_{crit}$ and $w_M^* < w_M^{DLTF}$, is expressed as

$$AF_{DLTF-SFA}^{peak power*4} \leq \frac{M \cdot \alpha \cdot (\theta_{LTF} \cdot w'_M)^\gamma + M \cdot \beta \cdot \theta_{LTF} \cdot w'_M + M \cdot \kappa}{\alpha \cdot w'_M{}^\gamma \cdot \frac{1-\delta+\delta M}{h(\delta)} + \beta \cdot w'_M (1-\delta+\delta M) + \kappa}.$$

For every M , we compute the maximum $AF_{DLTF-SVA}^{peak power*4}$ among all δ and w'_M , such that $\frac{4M+1}{6M} \leq \delta \leq 1$ and $\frac{s_{crit}}{\theta_{LTF}} < w'_M \leq s_{max}$.

Proof: The proof is similar to the proof of Lemma 9. ■

Theorem 5: The approximation factor for peak power reduction of DLTF-SVA, against the optimal task partitioning and optimal DVFS solution, is expressed as

$$AF_{DLTF-SFA}^{peak power} \leq \max \left\{ AF_{DLTF-SFA}^{peak power*1}, AF_{DLTF-SFA}^{peak power*2}, AF_{DLTF-SFA}^{peak power*3}, AF_{DLTF-SFA}^{peak power*4} \right\},$$

according to the definitions in Lemma 10, 11, 12, and 13.

Proof: The proof is similar to the proof of Lemma 9. ■

C. Numerical Results: DLTF-SVA against DLTF-SFA

Now that we have closed-form expressions for all approximation factors, we can compare the worst-case efficiency of DLTF-SVA, against the worst-case efficiency of DLTF-SFA, for an example power function modeled from simulations conducted on gem5 [3] and McPAT [19] for a 22 nm *out-of-order* Alpha 21264 core running an H.264 video encoder from the Parsec benchmark suite [2].

The comparison is presented in Fig. 7 and Fig. 8. With respect to peak power reduction, Fig. 8 shows that SVA always outperforms SFA, as expected. Furthermore, Fig. 7 shows that in the worst-cases for energy minimization SVA outperforms SFA when there is *non-negligible overhead for sleeping*, and the opposite happens when there is *negligible overhead for sleeping*. In practical terms, this means that if SFA manages to sleep efficiently, it will save more energy than SVA at the cost of having a higher peak power consumption. Nevertheless, if SFA fails to sleep efficiently (that is, cores mostly remain idle because there is not enough time for them to enter the sleep mode and return to execution mode), then SVA will save more energy than SFA.

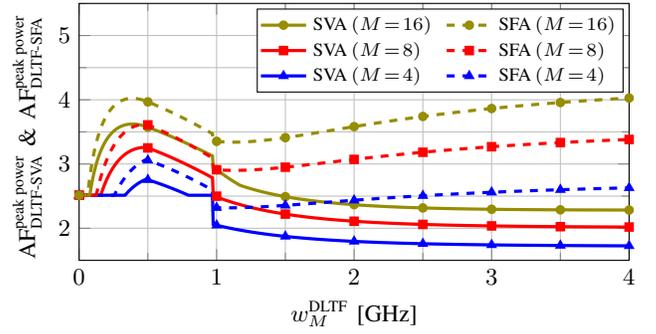


Fig. 8: $AF_{DLTF-SVA}^{peak power}$ and $AF_{DLTF-SFA}^{peak power}$ example for several M values, when $\gamma = 3$, $\alpha = 0.27 \frac{W}{GHz^3}$, $\beta = 0.52 \frac{W}{GHz}$, and $\kappa = 0.5 W$.

VIII. DISCRETE VOLTAGE AND FREQUENCY PAIRS

When the system has discrete voltage and frequency pairs $\{(v_1, f_1), (v_2, f_2), \dots, (v_F, f_F)\}$, the voltage of the island is set according to f_m such that $f_{m-1} < w_m^{DLTF} \leq f_m$, and the frequency of core i is set to f_j such that $f_{j-1} < w_i^{DLTF} \leq f_j$. Because now we consider that the cores execute at frequencies slightly higher than their cycle utilization, it no longer holds that all cores are always busy, and thus some cores will be kept idle during a short time. Therefore, by keeping the cores idle when they finish all their workload in the ready queue, the worst-case energy consumption and peak power consumption ratio for considering discrete voltage and frequency pairs against the continuous cases is $\rho_{SVA}^{max} = \max \left\{ \frac{\alpha \cdot f_m^{\gamma-1} \cdot \sum_{i=1}^M w_i^{DLTF} + M \neq 0 (\beta \cdot f_m + \kappa)}{\alpha \cdot w_M^{DLTF \gamma-1} \cdot \sum_{i=1}^M w_i^{DLTF} + M \neq 0 (\beta \cdot w_M^{DLTF} + \kappa)} \right\}$.

There are two extreme cases for ρ_{SVA}^{max} : (1) the total cycle utilization is really small, such that the static and independent energy/power consumptions dominate; or (2) the total cycle utilization is high, for which the highest total cycle utilization is $M \neq 0 \cdot w_M^{DLTF}$. Moreover, the worst case for ρ_{SVA}^{max} happens when $w_M^{DLTF} \rightarrow f_{m-1}$ and $w_i^{DLTF} \rightarrow f_{j-1}$ for all $i = 1, 2, \dots, M-1$. Therefore, we have that $\rho_{SVA}^{max} = \max_{1 < i \leq F} \left\{ \max \left\{ \frac{\beta \cdot f_i + \kappa}{\beta \cdot f_{i-1} + \kappa}, \frac{\alpha \cdot f_i^{\gamma-1} \cdot f_{i-1} + \beta \cdot f_i + \kappa}{\alpha \cdot f_{i-1}^{\gamma-1} + \beta \cdot f_{i-1} + \kappa} \right\} \right\}$. Finally, the approximation factors become $\rho_{SVA}^{max} \cdot AF_{DLTF-SVA}^{energy}$ and $\rho_{SVA}^{max} \cdot AF_{DLTF-SVA}^{peak power}$. For example, for a system with $\gamma = 3$, $\alpha = 0.27 \frac{W}{GHz^3}$, $\beta = 0.52 \frac{W}{GHz}$, $\kappa = 0.5 W$, and frequencies $\{0.1 GHz, 0.2 GHz, \dots, 4.0 GHz\}$, the value of ρ_{SVA}^{max} is 1.096.

IX. EXPERIMENTAL EVALUATIONS

This section presents experimental evaluations conducted with gem5 [3] and McPAT [19]. We compare the power and energy efficiency of DLTF-SVA and DLTF-SFA against the peak power and energy lower bounds.

A. Setup

We run our simulations for a single voltage island, for which we consider two different cases for the number of cores in the island: 8 cores and 16 cores. We consider 22 nm

out-of-order Alpha 21264 cores, with available frequencies $\{0.1 \text{ GHz}, 0.2 \text{ GHz}, \dots, 4.0 \text{ GHz}\}$. The minimum voltage for each frequency is taken from the work in [10]. 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.

For benchmarks, we consider two applications from the Parsec benchmark suite [2]: an H.264 video encoder, and a body track application. Each application is executed as a single independent thread in gem5 [3] (for all different frequencies) to obtain performance statistics, and then in McPAT [19] (for all different voltages for each frequency) to obtain the corresponding power consumptions. Thus, a task is an instance of one of these two types of applications, and by selecting different deadlines/periods we can consider different cycle utilizations. Furthermore, we test $5 \cdot 10^6$ different combinations of tasks, with a random number of tasks, random periods and random cycle utilizations, as well as tasks that result in cycle utilization distributions according to the analyzed worst cases.

We partition the tasks with DLTF, and we schedule them in each individual core according to EDF. For each core in SVA and for the island in SFA, the execution frequency is chosen as the closest available frequency that is higher than or equal to the required cycle utilization. The voltage for the island is chosen as the minimum voltage for stable execution for the core running at the highest frequency. As mentioned in Section VIII, since cores execute at frequencies higher than their cycle utilization, cores will be idle during short intervals for SVA, and during longer intervals for SFA. We assume that the time overhead of a core for entering the sleep mode and returning to execution mode is 100 ms. Under SVA, when a core has no more workload on its ready queue, the core can be simply kept idle¹. Under SFA, if the idle time in a core is less than 100 ms, then the core is kept idle (consuming idle power); and if it is larger, then the core is put to sleep (taking it 100 ms to enter the sleep mode and returning to execution mode), consuming idle power only during 100 ms.

To evaluate our previous analysis, we need to also compare against the peak power and energy lower bounds, for which the power values from McPAT can be modeled by power parameters $\gamma = 3$, $\alpha = 0.27 \frac{\text{W}}{\text{GHz}^3}$, $\beta = 0.52 \frac{\text{W}}{\text{GHz}}$, and $\kappa = 0.5 \text{ W}$ (details in Section III-A). Furthermore, we use Newton's Method to solve Equation (12) with 200 iterations, since this is possible for concrete cases and it reduces the pessimism.

B. Results

Fig. 9 and Fig. 10 present the results for energy minimization and peak power reduction, respectively, for DLTF-SVA and DLTF-SFA against the lower bounds, for $M = 8$ and $M = 16$. The horizontal axis represents the cycle utilization of the highest loaded core after partitioning with DLTF, i.e., w_M^{DLTF} . Among all the tested cases for each frequency, we show the maximum experimental ratio between DLTF-SVA or DLTF-SFA and the corresponding lower bound. The *saw-tooth* shapes observed in the figures are due to the regrouping

¹There is no restriction to further combine SVA with a DPM technique. We only assume that cores are kept idle in order to account for the worst cases.

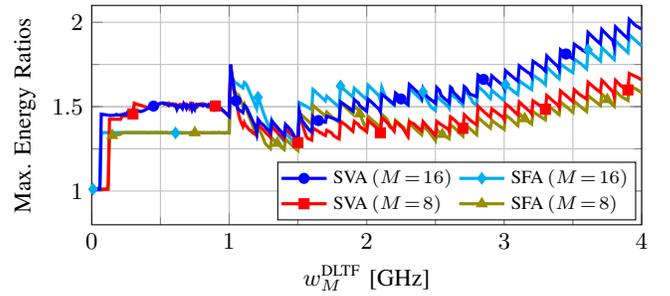


Fig. 9: Experimental results of energy minimization for DLTF-SVA and DLTF-SFA, against the energy lower bound.

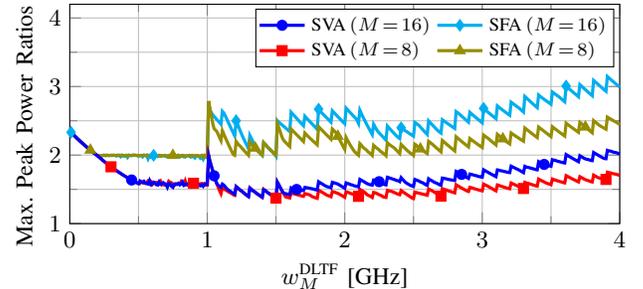


Fig. 10: Experimental results of peak power for DLTF-SVA and DLTF-SFA, against the peak power lower bound.

done by DLTF, which changes the resulting number of cores with cycle utilizations larger than 0 when possible, i.e., $M \neq 0$. Moreover, Fig. 11 and Fig. 12 directly compare DLTF-SVA to DLTF-SFA. The figures show the maximum and average experimental ratios between DLTF-SVA and DLTF-SFA, and vice-versa.

In Fig. 9, for the same number of cores, we can observe that DLTF-SFA generally behaves better than DLTF-SVA for the worst-cases with respect to energy minimization. However, there are several cases in which DLTF-SVA is more efficient. As explained in Section VII-C, this depends on how efficiently DLTF-SFA manages to bring cores into sleep mode. When DLTF-SFA fails to sleep efficiently, DLTF-SVA will save more energy. This can also be observed in Fig. 11. Contrarily, Fig. 10 and Fig. 12 show that DLTF-SVA always consumes less peak power than DLTF-SFA, both in average and for the worst-cases.

X. SYSTEMS WITH MULTIPLE VOLTAGE ISLANDS

For systems with multiple voltage islands that use SFA as DVFS schedule in individual islands, the work in [24] presents the Dynamic Voltage Island Assignment (DYVIA) algorithm for mapping task sets to islands. After the task partitioning, DYVIA results in the optimal task set mapping under SFA. Due to the similarities between SFA and SVA, the proofs from [24] can be trivially extended to show that DYVIA also results in the optimal task set mapping for SVA under our power

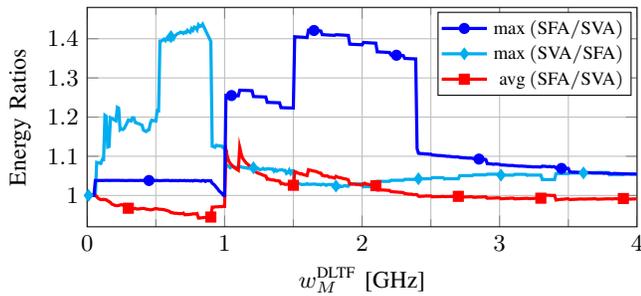


Fig. 11: Experimental results of energy minimization comparing maximum and average ratios for DLTF-SVA against DLTF-SFA, for an island with $M = 16$ cores.

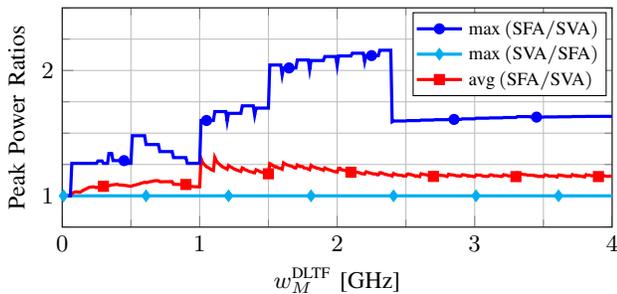


Fig. 12: Experimental results of peak power reduction comparing maximum and average ratios for DLTF-SVA against DLTF-SFA, for an island with $M = 16$ cores.

model (details are omitted due to space constraints). Moreover, [24] extends the analysis of SFA for energy minimization in a single voltage island, to consider combining DYVIA and SFA in multiple voltage islands. Such an analysis can also be trivially extended to derive the *approximation factor for combining DYVIA and SVA in multiple voltage islands under a given task partition*, resulting in the same approximation factors as those for a single voltage island.

XI. CONCLUSIONS

In this paper we have presented the *Single Voltage Approximation* (SVA) scheme and we have provided comprehensive analysis for the worst-case behavior of DLTF-SVA, both in terms of energy minimization and peak power reduction. The analysis and the experimental evaluations show that the efficiency for **energy minimization** of DLTF-SVA outperforms that of DLTF-SFA when the latter fails to efficiently bring cores to low-power modes. Nevertheless, when DLTF-SFA manages to sleep efficiently, DLTF-SVA results in higher energy consumptions. With respect to **peak power reduction**, *DLTF-SVA always consumes less peak power than DLTF-SFA*. This happens because the static and independent power consumption are equivalent for DLTF-SVA and DLTF-SFA in most cases (the peak power consumption of DLTF-SFA is bigger with low cycle utilizations), but the dynamic power consumption of DLTF-SFA can be much bigger than that

of DLTF-SVA, due to running cores at higher frequencies. Therefore, *DLTF-SVA is much more efficient than DLTF-SFA for peak power reduction in all cases, both average and corner cases*. Because of this reason, DLTF-SVA can potentially satisfy the chip's power budget for many more cases than DLTF-SFA would. Furthermore, this can be achieved without unnecessary sacrifices in terms of energy consumption.

ACKNOWLEDGMENT

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre *Invasive Computing* [SFB/TR 89], and by Baden Württemberg MWK Juniorprofessoren-Programms.

REFERENCES

- [1] H. Aydin and Q. Yang, "Energy-aware partitioning for multiprocessor real-time systems," in *Proceedings of 17th International Parallel and Distributed Processing Symposium (IPDPS)*, 2003, pp. 113 – 121.
- [2] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2008, pp. 72–81.
- [3] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.
- [4] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th Design Automation Conference (DAC)*, 2007, pp. 746–749.
- [5] J.-J. Chen, H.-R. Hsu, and T.-W. Kuo, "Leakage-aware energy-efficient scheduling of real-time tasks in multiprocessor systems," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2006, pp. 408–417.
- [6] J.-J. Chen and L. Thiele, "Energy-efficient scheduling on homogeneous multiprocessor platforms," in *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC)*, 2010, pp. 542–549.
- [7] V. Devadas and H. Aydin, "Coordinated power management of periodic real-time tasks on chip multiprocessors," in *Proceedings of the International Conference on Green Computing (GREENCOMP)*, 2010, pp. 61–72.
- [8] H. Esmailzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *38th Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 365–376.
- [9] R. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, pp. 263–269, 1969.
- [10] 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 *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 106–107.
- [11] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *Proceedings of the 2007 International Symposium on Low Power Electronics and Design (ISLPED)*, 2007, pp. 38–43.
- [12] J. Howard, S. Dighe, S. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, and R. Van Der Wijngaart, "A 48-core ia-32 processor in 45 nm cmos using on-die message-passing and dvfs for performance and power scaling," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, 2011.
- [13] Intel Corporation, "Dual-core intel xeon processor 5100 series thermal/mechanical design guide, revision 001," June 2006.

- [14] Intel Corporation, "Single-chip cloud computer (SCC)," 2009. [Online]. Available: <http://www.intel.com/content/www/us/en/research/intel-labs-single-chip-cloud-overview-paper.html>
- [15] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *Proceedings of the 41st Design Automation Conference (DAC)*, 2004, pp. 275–280.
- [16] 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 *Proceedings of the 8th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2012, pp. 245–254.
- [17] J. Lee, B. Yun, and K. Shin, "Reducing peak power consumption in multi-core systems without violating real-time constraints," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2013.
- [18] J. Lee and N. S. Kim, "Optimizing throughput of power- and thermal-constrained multicore processors using dvfs and per-core power-gating," in *Proceedings of the 46th Annual Design Automation Conference (DAC)*, 2009, pp. 47–50.
- [19] 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 *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009, pp. 469–480.
- [20] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.
- [21] T. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multicore in dark silicon era," in *Proceedings of the 50th Annual Design Automation Conference (DAC)*, 2013, pp. 174:1–174:9.
- [22] S. Pagani and J.-J. Chen, "Energy efficiency analysis for the single frequency approximation (SFA) scheme," in *Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2013, pp. 82–91.
- [23] S. Pagani and J.-J. Chen, "Energy efficient task partitioning based on the single frequency approximation scheme," in *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS)*, 2013, pp. 308–318.
- [24] S. Pagani, J.-J. Chen, and M. Li, "Energy efficiency on multi-core architectures with multiple voltage islands," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2014.
- [25] R. L. Rardin, *Optimization in Operations Research*. Prentice Hall, 1998.
- [26] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, 2014, pp. 185:1–185:6.
- [27] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.
- [28] R. Xu, D. Zhu, C. Rusu, R. Melhem, and D. Mossé, "Energy-efficient policies for embedded clusters," in *Proceedings of the 2005 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, 2005, pp. 1–10.
- [29] C.-Y. Yang, J.-J. Chen, and T.-W. Kuo, "An approximation algorithm for energy-efficient scheduling on a chip multiprocessor," in *Proceedings of the 2005 Conference on Design, Automation, and Test in Europe (DATE)*, 2005, pp. 468–473.



Santiago Pagani is a Ph.D. student and part of the research staff at the Chair for Embedded Systems (CES) in Karlsruhe Institute of Technology (KIT) in Germany. He received his Diploma in Electronics Engineering from the Department of Electronics, National Technological University (UTN), Argentina in 2010. From 2003 until 2012, he worked as a hardware and software developer in the industry sector for several companies in Argentina. He joined KIT and started his doctoral research in March 2012. His research interests include embedded systems, real-time systems, energy-efficient scheduling, power-aware designs and temperature-aware scheduling. He received Best Paper Awards from IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) in 2013, and from IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) in 2014.



Jian-Jia Chen is a Professor in the Department of Informatics in TU Dortmund University in Germany. He was a Juniorprofessor in the Department of Informatics in Karlsruhe Institute of Technology (KIT) in Germany from May 2010 to March 2014. He received his Ph.D. degree from Department of Computer Science and Information Engineering, National Taiwan University, Taiwan in 2006. He received his B.S. degree from the Department of Chemistry at National Taiwan University 2001. Between Jan. 2008 and April 2010, he was a postdoc researcher at Computer Engineering and Networks Laboratory (TIK) in ETH Zurich, Switzerland. His research interests include real-time systems, embedded systems, energy-efficient scheduling, power-aware designs, temperature-aware scheduling, and distributed computing. He received Best Paper Awards from ACM Symposium on Applied Computing (SAC) in 2009, IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) in 2005 and 2013, and IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) in 2014.



Jörg Henkel is currently with Karlsruhe Institute of Technology (KIT), Germany, where he is directing the Chair for Embedded Systems (CES). Before, he was with NEC Laboratories in Princeton, NJ. His current research is focused on design and architectures for embedded systems with focus on low power and reliability. Prof. Henkel has organized various embedded systems and low power ACM/IEEE conferences/symposia as General Chair and Program Chair and was a Guest Editor on these topics in various Journals like the IEEE Computer Magazine. He was/is Program Chair of CODES'01, RSP'02, ISLPED'06, SIPS'08, CASES'09, Estimedia'11, VLSI Design'12, ICCAD'12, PATMOS/VARI'13 and NOCS'14 and served as General Chair for CODES'02, ISLPED'09, Estimedia'12 and ICCAD'13. He is/has been a steering committee member of major conferences in the embedded systems field like at ICCAD, ISLPED, CODES+ISSS, CASES and is/has been an editorial board member of various journals like the IEEE TVLSI, IEEE TCAD, JOLPE, etc. He has given full/half-day tutorials at leading conferences including DAC, ICCAD, DATE, etc and has delivered six keynotes at CAD Conferences. Prof. Henkel received the 2008 DATE Best Paper Award, the 2009 IEEE/ACM William J. Mc Calla ICCAD Best Paper Award, the CODES+ISSS 2011 Best Paper Award, the MaXentric Technologies AHS 2011 Best Paper Award, and the CODES+ISSS 2014 Best Paper Award. He is the Chairman of the IEEE Computer Society, Germany Section, and was the Editor-in-Chief of the ACM Transactions on Embedded Computing Systems (ACM TECS) for six years. He is an elected board member of the DFG board on Technical Computer Science. He is an initiator and the coordinator of the German Research Foundation's (DFG) program SPP1500 on "Dependable Embedded Systems" and the site coordinator (Karlsruhe site) of the Three-University collaborative research center DFG TR89 "Invasive Computing". He holds ten US patents.