

## Article

# Temperature Sensor Assisted Lifetime Enhancement of Satellite Embedded Systems via Multi-Core Task Mapping and DVFS

Beomsik Kim  and Hoesook Yang \* 

Department of Electrical and Computer Engineering, Ajou University, 206 Worldcup-ro, Yeongtong-gu, Suwon-si 16499, Korea; asd326541@ajou.ac.kr

\* Correspondence: hyang@ajou.ac.kr; Tel.: +82-32-219-2361

Received: 24 September 2019; Accepted: 6 November 2019; Published: 9 November 2019



**Abstract:** Recently, thanks to the miniaturization and high performance of commercial-off-the-shelf (COTS) computer systems, small satellites get popular. However, due to the very expensive launching cost, it is critical to reduce the physical size and weight of the satellite systems such as cube satellites (CubeSats), making it infeasible to install high capacity batteries or solar panels. Thus, the low-power design is one of the most critical issues in the design of such systems. In addition, as satellites make a periodic revolution around the Earth in a vacuum, their operating temperature varies greatly. For instance, in a low earth orbit (LEO) CubeSats, the temperatures vary from 30 to  $-30$  degrees Celsius, resulting in a big thermal cycle (TC) in the electronic parts that is known to be one of the most critical reliability threats. Moreover, such LEO CubeSats are not fully protected by active thermal control and thermal insulation due to the cost, volume, and weight problems. In this paper, we propose to utilize temperature sensors to maximize the lifetime reliability of the LEO satellite systems via multi-core mapping and dynamic voltage and frequency scaling (DVFS) under power constraint. As conventional reliability enhancement techniques primarily focus on reducing the temperature, it may cause enlarged TCs, making them even less reliable. On the contrary, we try to maintain the TC optimal in terms of reliability with respect to the given power constraint. Experimental evaluation shows that the proposed technique improves the expected lifetime of the satellite embedded systems by up to 8.03 times in the simulation of Nvidia's Jetson TK1.

**Keywords:** low earth orbit satellites; reliability; temperature sensors; real-time embedded systems; multi-core processor; dynamic voltage and frequency scaling (DVFS)

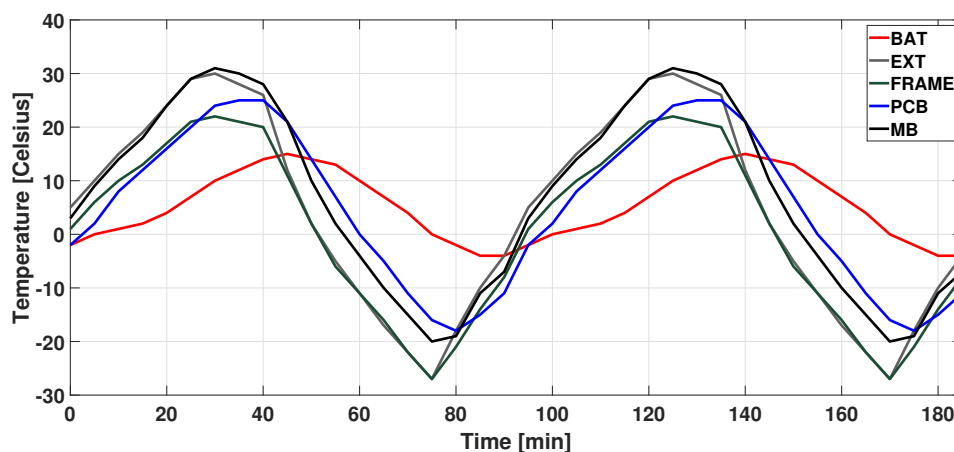
## 1. Introduction

The last decade has witnessed dramatic growth of space industry; From 2010 to 2015, the nano/microsatellite market has grown at an annual average of 39%, and it is expected that, from 2016 to 2022, it will further grow at an annual growth of 13% [1]. Especially, the demand for small satellites has increased significantly as the space industry has shifted from the government to the private market. In keeping with such an increasing need for small satellites or space missions, the CubeSat standard was initiated [2] for small satellites that weigh about a few kilograms (In the standard, 1 unit is a 10 cm cube ( $10 \times 10 \times 10 \text{ cm}^3$ ) with a mass of no more than 1.33 kg. A satellite may consist of a single (1U) or multiple cubes (3U, 6U, 12U and 27U).). While the CubeSats were originally developed for educational or demonstration purposes, their usages have been extended to more general and advanced missions, including scientific applications, deep space exploration, and so forth [3].

While the satellite systems are exposed to extreme conditions in terms of radiation and temperature, they are expected to operate for a long time without maintenance. Moreover, the missions imposed on such small satellites are getting more serious over time. In other words, the performance or reliability requirements of the onboard computer of satellite systems continue to increase. Thus, it is typical to design the satellite system with radiation-hardened processors [4] which generally have poorer performance than normal ones. In order to meet the reliability and performance requirements at the same time, the reconfigurable computing approach with field-programmable gate arrays (FPGAs) has been proposed, where various fault-tolerance techniques can be incorporated [5–7].

As CubeSats are subject to many physical constraints, including volume and weight, it is difficult to deploy large batteries or solar panels. Typical CubeSats with body-mounted solar panels generate less than 10 W, and state-of-the-art deployable solar panels produce 20–30 W. Batteries that are used in CubeSats typically store only 14–30 W·h [8]. In SwissCube [9], for instance, the average power generated from solar panels per orbit is only 1.5 W. Such a limited energy budget can restrict onboard computing performance. How to satisfy the increasing demand for performance and reliability within the given power budget is a challenge.

Reliability is one of the key design concerns in a satellite. Most space missions require a long lifetime. In general, low earth orbit (LEO) satellites tend to have shorter expected lifetimes (5–10 years) than that of geostationary orbit (GEO) satellites (15 years or more). Since the maintenance is physically impossible in the satellite systems, it is essential to design them to operate for a long lifetime without any failure in the first place. The thermal cycling (TC) effect that satellite experience in extreme temperature changes is one of the major reliability threats. In a LEO CubeSat (SwissCube), for instance, external temperature is reported to change from 30 °C to −30 °C as illustrated in Figure 1 [10]. In order to keep the system intact in severe external temperature changes, many physical protections, including thermal control, multi-layer insulation, sun shields, radiators, heat pipes and so forth, are applied in high-end satellites. For the small satellites such as CubeSats, however, it is difficult to fully have such physical protections due to the cost and physical constraints.



**Figure 1.** Temperature measurements in SwissCube [10] (BAT: battery, EXT: external, PCB: printed circuit board, and MB: motherboard).

In this paper, inspired by the fact that most small satellites are equipped with temperature sensors, we try to enhance the lifetime of the small satellite systems that are designed with multi-core processors without physical protections by adjusting the multi-core configuration in a temperature-aware manner. Thus far, most reliability enhancement techniques have tried to keep the operating temperature as low as possible [11–13] since it is well-known that high temperatures result in poor reliability. However, in satellite systems, this may not be the case as the external temperature varies greatly as shown in Figure 1. That is, in some cases, the artificial efforts to reduce the chip temperature may rather have an adverse effect of increasing the amplitude of TC.

To improve the lifetime reliability considering this TC effect, we propose to judiciously adjust the mapping of the software workload over the multiple cores and the operating frequency of the cores in a way that minimizes the amplitude of TC. In addition, we inject a virtual workload to the system if it is necessary to dissipate more power to improve the reliability. In doing so, the real-time schedulability of the satellites' mission and power constraints should still be satisfied.

Our contributions can be summarized as follows:

- we identify the lifetime anomaly, where lower temperatures result in an even worse expected lifetime in satellite systems;
- then in order to mitigate the TC effects, identified above, we propose a mapping/frequency assignment technique for multi-core satellite systems.

In the proposed technique, we target the satellite systems implemented on top of homogeneous multi-core system, where each core can have an independent frequency/voltage configuration (While many commercially available multi-core platforms only support cluster-level frequency and voltage modulation, there are such flexible systems [14] and other reliability enhancement techniques including [15] also target the same architecture.). As workloads, we assume that the satellites software is implemented as a set of periodically invoked real-time tasks. In order to enable fast yet accurate temperature evaluations, we assume that task execution time is long enough to reach the steady-state temperature. We believe this assumption is reasonable thanks to the satellite systems' low-power consumption. The inaccuracy that can be caused by this is analyzed in Section 4.4. To quantify the reliability of the system, we adopt the model proposed by Xiang et al. [16], where the lifetime is determined based on the temporal temperature profile. Considering the impact of spatial temperature gradients in the reliability remain as a future work.

The rest of this paper is organized as follows: In the next section, we discuss the related works and why the existing techniques are inefficient in the satellite systems. Section 3 quantitatively defines the proposed problem with task, architecture, power/temperature, and reliability models. Section 4 describes our method to maximize the lifetime reliability of the LEO satellite multi-core embedded systems in three steps. The evaluations are performed in simulation in Section 5 to show how the proposed technique improves the lifetime reliability, followed by concluding remarks and future works in Section 6.

## 2. Related Work

In addition to the TC effect that we mainly consider in this paper, there are three other known causes of failures in CMOS integrated circuits (ICs): electromigration (EM), time-dependent dielectric breakdown (TDDB), and stress migration (SM) [17]. Each of these failure mechanisms is quantified by Mean Time To Failure (MTTF), which is the expected lifetime concerning the failure source [18]. Most existing works focused on each of the above-mentioned causes individually, for example, EM [18–20], TDDB [18,21], SM [18], and TC [18,22]. Since these causes physically coexist in the operation of ICs, it is important to consider them altogether at the same time. Srinivasan et al. [23] proposed the Reliability-Aware Microprocessor (RAMP) model with negative bias temperature instability (NBTI) in addition to the four causes mentioned above. The five different failure causes are quantified in terms of reliability using the sum-of-failure-rates (SOFR) model, in which each failure mechanism is assumed to be associated with a constant failure rate. Xiang et al. [16] proposed a system-level reliability model with EM, TDDB, SM, and TC based on the Monte Carlo simulations.

It is well-known that high temperatures result in degradations in the IC's lifetime [18,24,25]. So, based on a simple assumption that cooler ICs would always result in better reliability, many reliability enhancement techniques have been proposed to reduce the peak temperature without actually quantifying the expected reliability [11–13]. In these works, the actual reliability has not been quantitatively analyzed, but indirectly enhanced by reducing the temperature.

There are a handful of works that particularly focus on the TC effect. Ukhov et al. [26] proposed a multi-processor scheduling technique that maximizes the reliability considering the

TC effect. While they showed that the MTTF could be improved by considering the TC effect in mapping/scheduling, the other causes, that is, EM, TDDb, and SM, were ignored in the reliability quantification. Rosing et al. [27] proposed the modified SOFR model and showed that aggressive power managements may harm the system's reliability due to the TC effect. That is, dynamic power management (DPM) or dynamic voltage scaling (DVS) often causes temperature variations, and in some circumstances, these TC effects play crucial roles as the bottleneck in long-term reliability. Ma et al. [15] proposed an online framework that adjusts core frequencies and voltages in order to lower the peak temperature and balance the temperature differences between the cores in favor of the reduced TC effects. Chantem et al. [28] proposed a reliability-aware online task mapping/scheduling algorithm for homogeneous multi-core systems. They relied on a theoretical assumption that spatial and temporal load balancing would always improve the MTTF. However, to the best of our knowledge, none of the existing works takes the variable ambient temperatures into consideration except for Park et al. [29]. They proposed the dynamic thermal management (DTM) for networked embedded systems that consist of multiple vehicle electronic control units (ECUs) under high and variable ambient temperature. The proposed technique is different from their work in that it is focused on the TC effect caused by the repeating and highly varying ambient temperature of satellite systems. Further, the proposed technique tries to maximize the MTTF value directly, while Park et al. [29] indirectly enhance the reliability by reducing the peak temperature.

In this paper, we aim at maximizing the expected lifetime (MTTF) of satellite embedded systems that operate in the space environment where the temperature dynamically changes by an excessive amount as shown in Figure 1. We take this as a key technical challenge and propose a multi-core task mapping and dynamic voltage and frequency scaling (DVFS) technique that matches with such variable ambient temperature conditions using temperature sensors.

### 3. System Model

In this section, we describe the task, architecture, power/temperature, and reliability models, followed by the problem definition.

#### 3.1. Task-Architecture Model

We consider a homogeneous multi-core system that consists of  $M$  cores, that is,  $PE = \{pe_1, pe_2, \dots, pe_M\}$ , as the target architecture. Each core can be operated at one of  $L$  different frequency levels, that is,  $F = \{f_1, f_2, \dots, f_L\}$  and this frequency level can be modulated at runtime. Note that we assume that  $F$  is sorted in ascending order of frequency, that is,  $\forall i < j, f_i < f_j$ . The frequency selection of a core is defined as a function of  $fa : PE \rightarrow F$ . For instance, when  $pe_m$  is decided to be operated at  $f_1$ ,  $fa(pe_m) = f_1$ . Also, it is assumed that a temperature sensor is placed on every core, thus, one can keep track of the temperature profile of each core.

For workloads on the target multi-core system, we consider an independent multi-task set that is defined as  $W = \{\tau_1, \tau_2, \dots, \tau_N\}$ . Each task is periodically invoked with a deadline. That is,  $\tau_n$  is specified with a tuple  $(ex_n, p_n)$ , where  $ex_n$  and  $p_n$  denote the number of worst-case execution cycles and the invocation period, respectively. Tasks have implicit deadlines, that is, the relative deadline of each invocation of  $\tau_n$  is equal to  $p_n$ . In addition to the given workload  $W$ , we propose to inject a set of virtual tasks  $V$  if it is necessary to intentionally dissipate more heat in the system. Similarly, a virtual task  $v_i \in V$  is also characterized by a tuple of the number of worst-case execution cycles and invocation period, that is,  $(v_{ex_i}, v_{p_i})$ .

We adopt the partitioned scheduling policy where the task-to-core assignment is defined as a function of  $map : (W \cup V) \rightarrow (PE \cup \{0\})$ , that is,  $map(\tau_n) = pe_m$  implies that task  $\tau_n$  is executed on  $pe_m$ . If  $map(\tau_n) = 0$ , task  $\tau_n$  is not mapped on any core. In the proposed technique, the mapping decision is made in two separate steps: task-to-logical-core mapping and logical-to-physical-core mapping. In the task-to-logical-core mapping represented by a function  $map_l : (W \cup V) \rightarrow (LP \cup \{0\})$ , the workloads are mapped on  $LP = \{lp_1, \dots, lp_M\}$ , a proxy of the physical cores  $PE$ . Then,

the logical-to-physical-core mapping is determined by a function  $map_p : LP \rightarrow (PE \cup \{0\})$ . For instance, if we have  $map_l(\tau_i) = lp_j$  and  $map_p(lp_j) = pe_k$ ,  $map(\tau_i) = pe_k$ . The assigned frequency of a logical core is preserved in the physical core, that is,  $fa(lp_i) = fa(pe_j)$  if  $map_p(lp_i) = pe_j$ . Note that the execution time of a task is dependent upon the frequency assignment decision. If  $\tau_n$  is assigned to core  $pe_m$ , the worst-case execution time of one invocation of  $\tau_n$  is  $ex_n / fa(map(\tau_n))$ . Then, the initial utilization of core  $pe_m$  at the lowest frequency can be calculated as  $u_m = \sum_{\tau_i \text{ s.t. } map(\tau_i)=pe_m} \frac{ex_i}{p_i \cdot f_1}$ . With the frequency modulation,  $f_1$  in the equation can be replaced with  $fa(pe_m)$ . Once the mapping decision is made, the multiple tasks on each core are scheduled according to the preemptive earliest-deadline-first (EDF) policy.

### 3.2. Power-Temperature Model

The power consumption at time  $t$  can be characterized as follows:

$$\mathbf{P}(t) = \mathbf{U} \circ \mathbf{P}_{act}(\mathbf{f}) + \mathbf{P}_{oth}(\mathbf{f}) + \mathbf{P}_{leak}(\mathbf{T}(t)). \quad (1)$$

Note that we keep track of the temperature of  $M$  cores, thus,  $\mathbf{P}(t)$  is the  $M \times 1$  vector and so are  $\mathbf{P}_{act}(\mathbf{f})$ ,  $\mathbf{P}_{oth}(\mathbf{f})$ , and  $\mathbf{P}_{leak}(\mathbf{T}(t))$ .  $\mathbf{f}$  is the  $M \times 1$  frequency vector that indicates the current frequency assignment of each core, that is,  $\mathbf{f} = [fa(pe_1), fa(pe_2), \dots, fa(pe_M)]'$ . Likewise,  $\mathbf{U}$  is the utilization vector of  $M$  cores, that is,  $\mathbf{U} = [u_1, u_2, \dots, u_M]'$ .  $\mathbf{P}_{act}(\mathbf{f})$  is the active power consumption vector when all cores are utilized by 100% at the frequency assignment of  $\mathbf{f}$ , so, the element-wise multiplication  $\mathbf{U} \circ \mathbf{P}_{act}(\mathbf{f})$  accounts for the active power consumption of the system under the current load.  $\mathbf{P}_{oth}(\mathbf{f})$  is the utilization-independent dynamic power consumption vectors, while  $\mathbf{P}_{leak}(\mathbf{T}(t))$  is the temperature-dependent leakage power consumption vectors.

While the CMOS power consumption is usually modeled as a simple summation of active and static power consumption, we elaborate on the modeling of temperature-dependent static power, using  $\mathbf{P}_{leak}$ , as we target the satellite system where temperature-dependent leakage could be crucial. In fact, leakage power becomes increasingly significant in CMOS IC due to the technology scaling and it has been reported that it accounts for up to 40% of the power consumption of today's microprocessors [30]. We use the piece-wise linear (PWL) leakage model which is known to be fast and highly-accurate [30,31] as follows:

$$\mathbf{P}_{leak}(\mathbf{T}(t)) = \boldsymbol{\alpha} \cdot \mathbf{T}(t) + \boldsymbol{\beta}. \quad (2)$$

where  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  are  $M \times M$  and  $M \times 1$  fitting coefficient diagonal matrix/vector, respectively.

For temperature evaluations, we rely on the thermal RC-circuit model for multi-core systems [32], which is based on the duality between heat transfer and electrical phenomena. In that RC-circuit model, electrical current and capacitance corresponds to heat flow through the (thermal) resistance and the heat-absorbing capability of the component, respectively [33]. In that model, the temperature of a certain position can be easily obtained by evaluating the voltage in the circuit. To be more specific, we use the following equation for evaluating the  $M \times 1$  temperature vector  $\mathbf{T}(t)$ :

$$\mathbf{C} \cdot \frac{d\mathbf{T}(t)}{dt} = \mathbf{P}(t) + \mathbf{K} \cdot \mathbf{T}_{amb} - (\mathbf{G} + \mathbf{K}) \cdot \mathbf{T}(t). \quad (3)$$

In the above equation,  $\mathbf{C}$  denotes the thermal capacitance of the  $M$  cores, represented as  $M \times M$  diagonal matrix. Heat transfer between cores is taken into account by  $\mathbf{G}$ , while heat dissipation from cores to the outside is by  $\mathbf{K}$ . Note that both  $\mathbf{G}$  and  $\mathbf{K}$  are  $M \times M$  thermal conductance matrices, and  $\mathbf{K}$  is a diagonal matrix.  $\mathbf{T}_{amb}$  is the  $M \times 1$  temperature vector that denotes the ambient (environment) temperature and  $\mathbf{P}(t)$  is the power consumption vector formulated in Equation (1).

Equation (3) can be simplified as follows, using  $\mathbf{A} = \mathbf{C}^{-1} \cdot (\mathbf{G} + \mathbf{K} - \boldsymbol{\alpha})$ ,  $\mathbf{B} = \mathbf{C}^{-1} \cdot (\boldsymbol{\beta}' + \mathbf{K} \cdot \mathbf{T}_{amb})$ , and  $\boldsymbol{\beta}' = \mathbf{U} \circ \mathbf{P}_{act}(\mathbf{f}) + \mathbf{P}_{oth}(\mathbf{f}) + \boldsymbol{\beta}$ :

$$\frac{d\mathbf{T}(t)}{dt} = -\mathbf{A} \cdot \mathbf{T}(t) + \mathbf{B}. \quad (4)$$

When the system keeps the same configuration, that is, mapping and frequency assignment, for a long enough time, it will eventually reach a steady-state. The steady-state temperature can be easily obtained by having  $\frac{d\mathbf{T}(t)}{dt} = 0$  in Equation (4). That is, the steady-state temperature vector  $\mathbf{T}_{ss}$  can be formulated as follows

$$\mathbf{T}_{ss} = \mathbf{A}^{-1} \cdot \mathbf{B} = (\mathbf{G} + \mathbf{K} - \boldsymbol{\alpha})^{-1} \cdot (\boldsymbol{\beta}' + \mathbf{K} \cdot \mathbf{T}_{amb}). \quad (5)$$

Solving the differential Equation (4), we have the temperature vector

$$\mathbf{T}(t) = \mathbf{T}_{ss} + (\mathbf{T}(t_0) - \mathbf{T}_{ss}) \cdot e^{-\mathbf{A}(t-t_0)}. \quad (6)$$

where  $\mathbf{T}(t_0)$  is initial temperature vector.

### 3.3. Reliability model

Among the four failure mechanisms we consider in this work, EM, TDDB, and SM can be modeled as follows in Equations (7)–(9) [16]:

$$MTTF_{EM} = \frac{A_{EM}}{J^n} \cdot e^{\frac{E_{a,EM}}{k \cdot T}}, \quad (7)$$

$$MTTF_{TDDB} = A_{TDDB} \cdot \left(\frac{1}{V}\right)^{(a-bT)} \cdot e^{\frac{X+Y/T+ZT}{k \cdot T}}, \quad (8)$$

and

$$MTTF_{SM} = A_{SM} \cdot |T_o - T|^{-n} \cdot e^{\frac{E_{a,SM}}{k \cdot T}}. \quad (9)$$

What those three models have in common is that they are strongly dependent on the *absolute degree* of the temperature. That is, simply, the higher  $T$  they have, the smaller the MTTF values are. On the other hand, TC exhibits a different behavior as the wear in TC is mainly due to the difference in thermal expansion coefficients between adjacent material. This accumulated damage causes permanent failure in the package, solder, interconnects, and dielectric materials. Thus, in TC, how much *temporal temperature gradient* a system has is important. The MTTF due to TC can be modeled as follows [16]:

$$MTTF_{TC} = \frac{p}{\sum \frac{1}{N_{C_i}}} \quad (10)$$

with  $p$  equal to the period of temperature history.  $N_{C_i}$  is the effect of cycle  $i$  and can be quantified by the modified Coffin-Manson equation with the Arrhenius term [16] as follows:

$$N_{C_i} = A_{TC} \cdot (\Delta T - \Delta T_0)^{-q} \cdot e^{\frac{E_{a,TC}}{k \cdot T_{max}}} \quad (11)$$

where  $A_{TC}$  is fitting constant,  $\Delta T$  is cycle amplitude,  $\Delta T_0$  is the portion of the temperature range in the elastic region,  $E_{a,TC}$  is activation energy,  $k$  is Boltzmann's constant,  $T_{max}$  is the maximum temperature during the cycle, and  $q$  is Coffin-Manson exponent constant that depends on the material characteristic. Usually,  $q$  is set to 6–9 for brittle fracture (Si and dielectrics), to 3–5 for hard metal alloys/intermetallics (Al-Au), and to 1–3 for ductile metal (solder) [18]. It is worthwhile to mention that



in the TC mechanism, unlike all others, the negative impact of temperature gradient  $\Delta T$  is explicitly considered, which motivates our work.

### 3.4. Problem Definition

The problem we target to solve in this work can be summarized as follows:

**Input:** Given the LEO CubeSat PCB temperature history as exemplified in Figure 1, that is, ambient temperature ( $T_{amb}$ ), the periodic task set  $W$  as workloads, and the power-temperature models presented in Section 3.2,

**Constraints:** while respecting the timing constraints of the given task sets ( $p_i$  for each  $\tau_i \in W$ ) and not violating the given power budget  $P_{max}$ , that is,  $\forall t, \sum_{i=0}^M P(t)[i] \leq P_{max}$ ,

**Output:** determine the mapping decision  $map$  and the frequency assignment  $fa$ , and impose additional virtual workload  $V$  and determine its mapping if necessary,

**Objective:** in order to maximize the MTTF.

## 4. Proposed Mapping/DVFS Technique

In this section, we propose a mapping/DVFS technique for multi-core embedded systems, presented in Section 3, tailored to the reliability optimization of LEO satellites. The most noticeable property of the target system is that they are exposed to highly varying temperature environments as shown in Figure 1. Note that most conventional approaches typically try either to minimize the power consumption or to keep the temperature as low as possible. In the highly varying temperature condition, however, such approaches may result in considerable temperature fluctuations over time, which, in turn, can negatively affect the system as a reliability threat.

We argue that the temperature management decision should be judiciously made considering the outside temperature obtained through a temperature sensor. For instance, when the environment temperature is very low, it would be even better to have higher clock frequencies than necessary in order to intentionally heat up the cores. This over-clocking decision is helpful to reduce the amplitude of the TC. On the other hand, this is not always feasible or desirable. Firstly, the power budget may now allow wasteful over-clocking in some cases. Or, if the outside temperature is relatively high, this may cause even bigger thermal gradients. It is also important to consider the three other failure mechanisms as well as TC. Therefore, it is not trivial to make an optimal mapping/DVFS decision for the given condition.

Whilst the cycle of revolution of the LEO satellites is consistent, the maximum and minimum temperatures within the cycle vary depending on the season. Therefore, it is computationally intractable to precompute the mapping/DVFS solutions for all possible conditions. We propose a hybrid solution, that consists of two offline steps followed by an online step, as outlined in Figure 2. First, at the highest temperature of the revolution cycle, an initial mapping/DVFS decision is made in a way that minimizes the peak temperature ( $T_{top}$ ). How to obtain this initial mapping is presented in Section 4.1. Then, out of this initial condition, the minimum temperature of the TC that causes the largest MTTF value is derived as a temperature threshold ( $T_{th}$ ). This procedure is described in Section 4.2. At runtime, whenever this threshold is violated, that is, the temperature goes below the derived minimum, a new DVFS decision is made and a set of a virtual task is injected if necessary (Section 4.3).

### 4.1. Initial Mapping and Frequency Assignment

The main idea of the initial mapping and frequency assignment is to minimize the peak temperature as it is at the highest environment (PCB) temperature. In doing so, we rely on the worst-fit (WF) heuristic which has also been popularly used in the existing multi-core mapping approaches [28,34]. The initial mapping consists of three sub-procedures: (i) task-to-logical-core mapping; (ii) frequency modulation; and (iii) logical-to-physical-core mapping.

Algorithm 1 delineates the initial mapping procedure. Firstly, it determines the task-to-logical-core mapping in a way that the workloads are evenly distributed over the all constituent cores by means of

the WF bin-packing heuristic (lines 1–6). After sorting out the tasks in descending order of utilization (line 1), it maps the tasks one by one to the idle logical core (lines 4–5). Once the logical mapping is done, it scales up the operating frequencies of the cores as necessary (lines 8–16). That is, if a core is used by too many tasks exceeding its limit (line 9), its frequency is scaled up by one level until the utilization gets less than or equal to 1.0. With these frequency assignments done, we can calculate the power consumption of each core excluding the temperature-dependent part (line 17). At last, the logical-to-physical-core mapping decision is made in the third part (lines 19–34), in which the maximum steady-state temperature of the system is supposed to be minimized. In doing so, the logical core with the maximum power consumption that remains unmapped is chosen (line 21). Then, all possible physical core mapping candidates are investigated (lines 23–30) and the one that results in the lowest maximum steady-state temperature is chosen for mapping (line 31).

---

**Algorithm 1** Initial Mapping and Frequency Assignment
 

---

```

1: sort  $W$  in a descending order of  $\frac{ex_n}{p_n \cdot f_1}$ ;
2:  $\forall n$ , set  $map_l(\tau_n) = 0$ ; ▷ (i) Logical core mapping
3: for  $n = 1$  to  $N$  do
4:   find  $lp_m$  with the minimum  $u_m$  value;
5:   set  $map_l(\tau_n) = lp_m$  and update  $u_m$ ;
6: end for
7:
8: for  $m = 1$  to  $M$  do ▷ (ii) Frequency modulation
9:   while  $u_m > 1$  do
10:    if  $fa(lp_m) = f_L$  then ▷ Highest frequency
11:      return not schedulable;
12:    else ▷ Scaling up the frequency by one level
13:      when  $fa(lp_m) = f_l$ , adjust  $fa(lp_m)$  to  $f_{l+1}$ ;
14:    end if
15:  end while
16: end for
17: calculate  $\mathbf{P}$  with Equation (1) (w/o  $\mathbf{P}_{leak}$ )
18:
19:  $\forall i$ , set  $map_p(lp_i) = 0$ ; ▷ (iii) Physical core mapping
20: while  $LP \neq \phi$  do
21:   find  $lp_i \in LP$  with the maximum  $\mathbf{P}[i]$ ;
22:    $T_{curr\_min} \leftarrow \infty$ ;  $ind\_min \leftarrow 1$ ;
23:   for  $j = 1$  to  $M$  do ▷ Find the smallest temp
24:     set  $map_p(lp_i) = pe_j$ ; ▷ Try mapping  $lp_i$  on  $pe_j$ 
25:     evaluate the maximum temperature  $T_{max}$ ;
26:     if  $T_{max} < T_{curr\_min}$  then
27:        $T_{curr\_min} \leftarrow T_{max}$ ;  $ind\_min \leftarrow j$ ;
28:     end if
29:     set  $map_p(lp_i) = 0$ ; ▷ Restore the mapping
30:   end for
31:   set  $map_p(lp_i) = pe_{ind\_min}$ ;
32:    $LP \leftarrow LP - \{lp_i\}$ ;
33: end while
34: return schedulable;

```

---



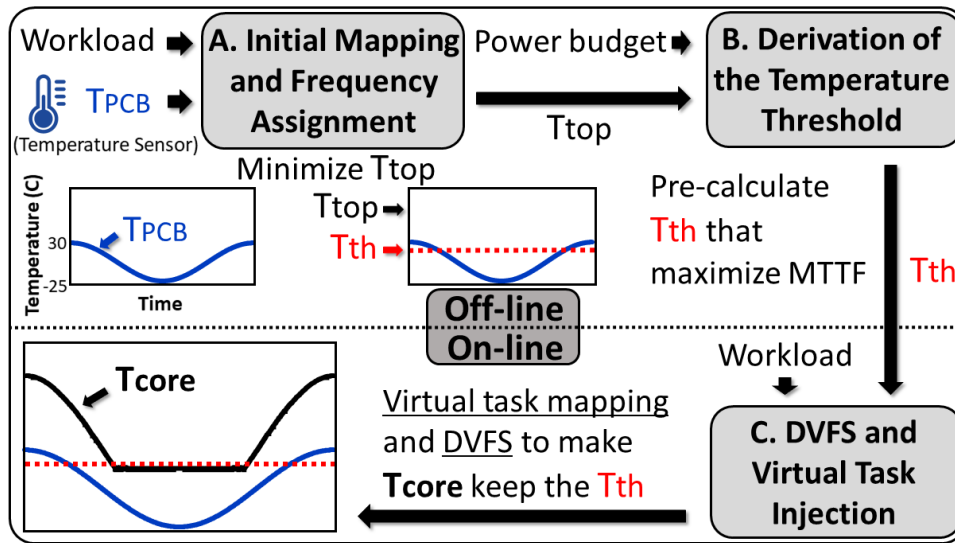


Figure 2. Overall procedure of the proposed mapping/DVFS technique.

#### 4.2. Derivation of the Temperature Threshold

Once the initial mapping/DVFS decision is fixed, we derive the temperature threshold vector  $T_{th}$ , above which the target system is always kept at runtime. For that, we first derive a model temperature profile  $T_{ev}$  whose minimum temperature is  $T_{th}$ . The main challenge in finding a good  $T_{ev}$  is to maximize the MTTF value considering TC and other three failure sources within the given power budget,  $P_{max}$ .

The procedure of finding  $T_{ev}$  is as follows. From the initial mapping, we can already determine the highest temperature vector, denoted as  $T_{top}$ , in a single revolution cycle. Basically, we iterate a number of candidates for the lowest temperature vector,  $T_{bot}$ , by means of binary search to find the one that results in the largest MTTF value. In order to limit the search range ( $T_{bot,min} \leq T_{bot} \leq T_{bot,max}$ ), we lower-bound  $T_{bot}$  by  $T_{bot,min}$  which can be obtained by invoking Algorithm 1 at the lowest ambient temperature ( $T_{PCB}$  in Figure 3). Similarly, we set the upper-bound of  $T_{bot}$  as  $T_{bot,max}$ , which can be obtained by assuming that every core is utilized by 100% at the highest frequency with the same mapping. Once those maximum and minimum temperatures,  $T_{top}$  and  $T_{bot}$  are fixed and the intermediate temperatures between the two can be interpolated using the cosine function as follows (Note that the temperature changes caused by highly varying ambient temperature are modelled by a cosinusoidal form based on the observation of the temperature measurement data from SwissCube [10]):

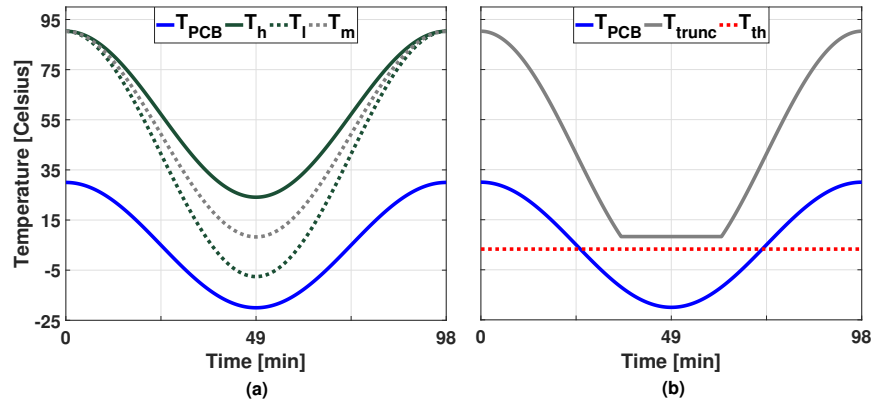
$$T_{ev}(t) = \frac{T_{top} - T_{bot}}{2} \cos\left(\frac{2\pi t}{t_p}\right) + \frac{T_{top} + T_{bot}}{2} \quad (12)$$

where  $t_p$  denotes the revolution period of the satellite.

The first half of Algorithm 2 (lines 1–23) illustrates this binary search procedure. Note that we borrow the Monte-Carlo simulation framework from Xiang et al.’s work [16] for the evaluation of MTTF for a temperature profile  $T$  and this is denoted as  $SIM(T)$  in the pseudocode. It is worthwhile to mention that the temperature profile used during the binary search is not the exact one. It is an approximated one that is temporarily used for quantifying the effect of candidate TCs. Figure 3a demonstrates exemplary temperature profiles that are compared in the binary search.

Note in Equation (11) that TC is only dependent on the peak and bottom temperatures of the cycle, not on the intermediate temperatures in-between. Thus, regarding the TC effect, just keeping the system temperature above  $T_{th}$  is fine. Considering the other three effects, it is desirable to minimize power consumption to reduce the temperature. So, we keep the mapping/DVFS decision obtained by Algorithm 1 as long as the temperature threshold is not violated. On the other hand, if this threshold is too high, it would not be feasible to heat up the system within the given power budget. So, in the

second half of the algorithm (lines 25–30),  $T_{th}$  is adjusted to confirm that the threshold is always maintainable with the given power budget. The expected power consumption is calculated using the modified Equation (5):  $P(t) = (G + K) \cdot T(t) - K \cdot T_{amb}$ . That is, we calculate back the required power  $P(t)$  that results in the steady-state temperature of  $T(t)$ . If it violates the following power constraint,  $\forall t, \sum_{i=0}^M P_{ev}(t)[i] = P_{ev}^{sum} \leq P_{max}$ , it repeatedly reduces the temperature threshold (line 29) until satisfied.



**Figure 3.** (a) Example temperature traces compared in the binary search and (b) the truncated envelope temperature and the temperature threshold.

---

#### Algorithm 2 Derivation of the Temperature Threshold

---

```

1:  $T_{h\_bot} \leftarrow T_{bot,max};$  ▷ Initialization
2:  $T_{l\_bot} \leftarrow T_{bot,min};$ 
3:  $T_{th} \leftarrow T_{l\_bot};$ 
4:  $T_h(t) \leftarrow \text{Equation (12) using } T_{bot} = T_{h\_bot};$ 
5:  $T_l(t) \leftarrow \text{Equation (12) using } T_{bot} = T_{l\_bot};$ 
6:  $MTTF_h \leftarrow \text{SIM}(T_h), MTTF_l \leftarrow \text{SIM}(T_l);$ 
7:
8: while true do ▷ (i) Binary search
9:    $T_{m\_bot} \leftarrow (T_{h\_bot} + T_{l\_bot})/2;$ 
10:   $T_m(t) \leftarrow \text{Equation (12) using } T_{bot} = T_{m\_bot};$ 
11:   $MTTF_m \leftarrow \text{SIM}(T_m);$ 
12:  if  $MTTF_h \geq MTTF_l$  then ▷  $T_{bot}$  to be increased
13:     $T_{l\_bot} \leftarrow T_{m\_bot}, MTTF_l \leftarrow MTTF_m;$ 
14:    if  $T_{h\_bot} \simeq T_{m\_bot}$  then
15:       $T_{th} \leftarrow T_{h\_bot};$  break;
16:    end if
17:  else ▷  $T_{bot}$  to be decreased
18:     $T_{h\_bot} \leftarrow T_{m\_bot}, MTTF_h \leftarrow MTTF_m;$ 
19:    if  $T_{l\_bot} \simeq T_{m\_bot}$  then
20:       $T_{th} \leftarrow T_{l\_bot};$  break;
21:    end if
22:  end if
23: end while
24:
25:  $T_{trunc}(t) \leftarrow \text{Equation (12) using } T_{bot} = T_{bot,min};$ 
26: repeat ▷ (ii) Power constraint
27:    $\forall t \text{ s.t. } T_{trunc}(t) < T_{th}, T_{trunc}(t) \leftarrow T_{th};$ 
28:   calculate  $P_{ev}^{sum}$  with  $T_{trunc}(t);$ 
29:    $T_{th} \leftarrow T_{th} - [1, \dots, 1]';$ 
30: until  $P_{max} \geq P_{ev}^{sum};$ 
31: return  $T_{th};$ 

```

---

### 4.3. DVFS and Virtual Task Injection

At runtime, the initial decision made by Algorithm 1 is preserved unless the temperature threshold is violated. In case of the violation, the system needs to artificially heat up properly. On the occasion of the  $T_{th}$  violation, Algorithm 3 is invoked. It first reads the current temperatures  $T_{cur}$  from the temperature sensors (line 1) and calculates the current power vector  $P_{cur}$  (line 2). At line 3, the power consumption needed to reach the threshold temperature  $T_{th}$  is calculated. Then, for each core (line 5), it tries to scale up the frequency repeatedly (lines 7–8) until it either reaches to the maximum level (line 7) or the target power consumption is reached (line 6). When a core frequency is scaled up maximally and the target power consumption is not reached yet (line 9), a virtual task  $v_i$  is injected on the core (lines 11–12). Note that the execution cycle and period of the virtual task is fixed a priori.

As invoked at runtime, it is important for Algorithm 3 to be light-weight to be executed without causing considerable overheads in CPU. The time complexity of Algorithm 3 is  $O(|M| \cdot L)$ , where  $L$  denotes the maximum number of iterations of the while loop in lines 6–19. We believe that both  $|M|$  and  $L$  are manageably small for the following reasons. First, the number of cores installed in the small satellite,  $|M|$ , is typically small. And, the maximum number of the while loop iterations,  $L$ , is also not too big due to the triggering condition of Algorithm 3. Note that Algorithm 3 is invoked *each* time the  $T_{th}$  violation is detected. Thus, the current power that causes the temperature violation ( $P_{th}[m]$ ) is not so far away from ( $P_{cur}[m]$ ). Thus, the actual number of loop iterations is not usually big. From our empirical evaluations, that will be presented in Section 5, the maximum number loop iterations caused by the frequency scaling (lines 7–8) and virtual task injection (lines 9–17) were only 2 and 5, respectively.

---

#### Algorithm 3 DVFS and Virtual Task Injection

---

```

1:  $T_{cur} \leftarrow$  current temperature sensor values;
2: Calculate  $P_{cur}$  with  $T_{cur}$  using Equation (1);
3:  $P_{th} = (G + K) \cdot T_{th} - K \cdot T_{amb}$ ;
4:  $i \leftarrow 0$ ;
5: for  $m = 1$  to  $M$  do ▷ For each core
6:   while  $P_{th}[m] > P_{cur}[m]$  do
7:     if  $fa(pe_m) \neq f_L$  then ▷ Frequency scaling
8:       when  $fa(pe_m) = f_l, fa(pe_m)$  to  $f_{l+1}$ ;
9:     else ▷ Virtual task injection
10:      if  $u_m + \frac{v_{ex}}{v_{p \cdot f_L}} \leq 1$  then
11:         $V \leftarrow V \cup \{v_i\}$ ;
12:        set  $map(v_i) = pe_m$  and update  $u_m$ ;
13:         $i \leftarrow i + 1$ ;
14:      else
15:        break;
16:      end if
17:    end if
18:    update  $P_{cur}[m]$ ;
19:  end while
20: end for

```

---

### 4.4. Inaccuracy of Temperature Analysis

In Algorithm 2 and 3, the temperature is not exactly evaluated, but approximately assuming a steady-state. In some cases, task execution time is not long enough to reach the steady-state temperature. The exact transient temperature can only be calculated by Equation (6), which is computationally too expensive. Therefore, we adopt the steady-state temperature approximation (line 28 in Algorithm 2 and line 3 of Algorithm 3). To compensate for the inaccuracy caused by the approximation, we introduce a margin  $P_\Delta$  in the temperature comparison. That is, the inequality at line 3 of Algorithm 3 can be replaced with  $P_{th}[m] + P_\Delta > P_{cur}[m]$ . It is also worthwhile to mention that

there could be a lot of small TCs in an actual schedule as the task execution bursts and idle times are interleaved in a complicated pattern. However, we decide to ignore the effect of such small cycles by approximating the temperature evaluation as it has been reported that they do not cause any serious long-term reliability problems [27,35].

In summary, Algorithm 1 minimizes the peak temperature of the core at the highest PCB temperature to reduce TC amplitude. Algorithm 2 derives the minimum temperature (threshold), above which the system's temperature should be maintained, considering the power budget. Algorithm 3 artificially heats up the processor by means of DVFS and virtual task injections, whenever the temperature goes below the threshold.

## 5. Experiments

### 5.1. Evaluation Environment and Parameters

For evaluation, we choose Nvidia's Jetson TK1 as the target architecture, which has quad-core ARM Cortex-A15 CPU and supports 12 different frequencies from 1.24 to 2.32 GHz, that is,  $PE = \{pe_1, pe_2, pe_3, pe_4\}$  and  $F = \{1.24, 1.33, 1.43, 1.53, 1.63, 1.73, 1.84, 1.94, 2.01, 2.12, 2.22, 2.32\}$ . Though all cores operate at the same frequency in the actual setup of the target architecture, we assume that each core can have an independent frequency level in the simulation. (Ma et al. [15] also had the same assumption in their evaluations.)

We borrow the power and temperature parameters of Nvidia's Jetson TK1 from Ma et al.'s work [15] to characterize the power and temperature behaviors. For the temperature-dependent leakage power, we extend their power model using the PWL approximation [30]. And, the model is calibrated to have the same power consumption reported in Reference [15] at the ambient temperature of 20 °C using the leakage power estimation, presented in Reference [31]. The resultant power model and its parameters are as follows:  $P_{act}(f) = 0.8031 \cdot f^2 - 2.046 \cdot f + 1.481$ ,  $P_{oth}(f) = -0.08089 \cdot f^2 + 0.3841 \cdot f$ , and  $P_{leak}(T(t)) = \alpha \cdot T(t) + \beta$ , where  $(\alpha, \beta)$  is (0.001796, 0.1098) if  $T(t) < 0$ , (0.00393, 0.1079) if  $0 \leq T(t) < 40$ , (0.006781, -0.0080065) if  $40 \leq T(t) < 80$ , and (0.01035, -0.2955) if  $T(t) \geq 80$ . For all cores, thermal capacitance  $C$  and thermal ground conductance  $K$  is set to 2.34 J/°C and 0.098 W/°C, respectively. The thermal conductance  $G$  to the adjacent cores and to the core are set to -0.03 W/°C and -0.0075 W/°C, respectively, for all cores. To obtain the temperature profile with respect to the given power profile, the architecture-level thermal RC-circuit with the above parameters is evaluated. In doing so, Equation (6) is calculated with the given parameters for each time step in MATLAB.

In the reliability model, since we consider the failure of the brittle materials [18,36,37], we set the Coffin-Manson exponent  $q$  to 6 in the Monte Carlo simulator [16]. All other parameters of the four mechanisms were set to the default values.

We modeled a set of periodic tasks using actual satellite workload periodic profiles such as executive, attitude determination and control, thermal management, and power management software, that is,  $W = \{\tau_1, \tau_2, \dots, \tau_9\}$ , the tuple denoted as  $(\frac{ex_n}{f_1}, p_n)$ , that is,  $\tau_1 = (0.06, 0.1)$ ,  $\tau_2 = (0.06, 0.1)$ ,  $\tau_3 = (0.3, 0.5)$ ,  $\tau_4 = (0.3, 0.5)$ ,  $\tau_5 = (0.8, 1)$ ,  $\tau_6 = (0.8, 1)$ ,  $\tau_7 = (2, 8)$ ,  $\tau_8 = (3, 8)$ , and  $\tau_9 = (3, 8)$ . For the virtual task, we use the parameter of  $v_i = (1, 8)$ , that is, its execution time at the lowest frequency and the invocation period is 1 and 8 s, respectively.

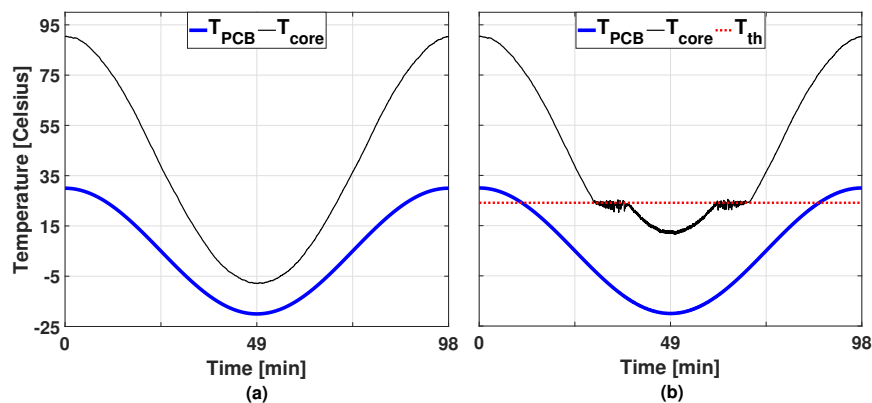
### 5.2. Simulation Results

#### 5.2.1. Comparison to the Conventional Low-Power Mapping

We first compare the proposed technique with the conventional low-power mapping approach. We choose Xian et al.'s work [34] as a comparable target which is based on the WF bin packing heuristic that balances the load to minimize the total energy consumption. The mapping decision is made in each hyper-period, that is 8 seconds and the EDF scheduling policy is adopted for the scheduling of the multiple workloads assigned to the same core. Figure 4a depicts the resultant temperature

profile of Xian et al. [34], which causes a large TC with an amplitude of about 98 degrees. In this case, the average power consumption is 3.79 W.

We first apply the proposed technique without virtual task injections (lines 10–13 of Algorithm 3) and the resultant temperature trace is shown in Figure 4b. As can be seen in the figure, applying DVFS solely is not enough to prevent the temperature threshold violations from happening at the lower ambient temperatures. However, despite that, the normalized MTTF has been improved by 5.48 times as summarized in Table 1. Due to the artificial heat up procedure (Algorithm 3), the average power consumption has increased to 4.59 W.



**Figure 4.** Temperature profiles of  $T_{core}[0]$  caused (a) by Xian et al. [34] and (b) by the proposed technique without virtual task injections.

### 5.2.2. Different Power Budgets

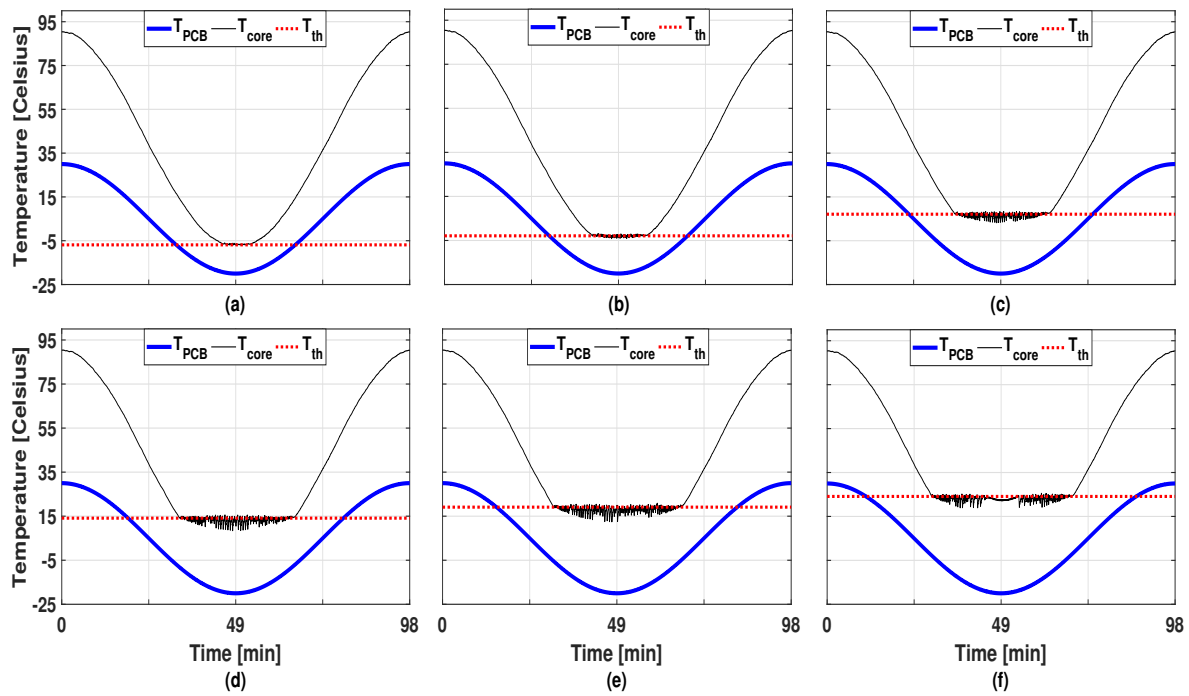
In order to investigate the effects of the power budget in the proposed technique, we try six different power budgets (4.02 W, 4.08 W, 4.34 W, 4.59 W, 4.81 W, and 5.00 W), and, in this case, the virtual task injection is enabled. Figure 5 shows the temperature profiles caused by each case and Table 1 summarizes the threshold temperature, the average power consumption, and the normalized MTTF. Due to space limitation, only the temperature profiles for one core ( $T_{core}[0]$ ) are displayed in Figure 5 and Table 1. It is clearly noticeable that a bigger power budget allows for a higher threshold temperature. Thanks to the higher threshold, the amplitude of the TC could be effectively reduced, bringing the gain of enlarged MTTF values of up to 8.03 times. In all cases, the average power consumption was kept below the imposed power budget as summarized in Table 1.

**Table 1.** Comparisons of  $T_{th}$ , average power consumption, and normalized MTTF.

| Figure                             | 4(a) | 4(b)  | 5(a)  | 5(b)  | 5(c) | 5(d)  | 5(e)  | 5(f)  |
|------------------------------------|------|-------|-------|-------|------|-------|-------|-------|
| $T_{th}[0]$ ( $^{\circ}\text{C}$ ) | —    | 24.11 | −6.89 | −2.89 | 7.11 | 14.11 | 19.11 | 24.11 |
| Average Power (W)                  | 3.79 | 4.59  | 3.81  | 3.89  | 4.11 | 4.35  | 4.53  | 4.81  |
| Normalized MTTF                    | 1    | 5.48  | 1.02  | 1.32  | 2.24 | 3.65  | 4.89  | 8.03  |

### 5.2.3. Effect of Frequency Ranges

If the hardware supports a wider frequency range, it has more room for lifetime enhancement. In the setup we used in the previous experiments when the frequency scaling is possible up to 2.52 GHz, the normalized MTTF is further enhanced to 22.15 with the average power consumption of 5.56 W. In modern satellite systems, however, it is not common to have such high frequency. In the case of narrower ranges, the proposed technique could considerably maximize the MTTF. When the maximum frequency is set to 1.73 GHz, the normalized MTTF is 1.21 with the average power consumption of 3.87 W.



**Figure 5.** Temperature profiles asb of  $T_{core}[0]$  with the proposed technique using virtual tasks and DVFS under different power constraint: (a) 4.02 W, (b) 4.08 W, (c) 4.34 W, (d) 4.59 W, (e) 4.81 W, and (f) 5.00 W.

#### 5.2.4. Effect of Coffin-Manson Exponent

Lastly, we show how sensitive the proposed technique to the parameters of the Coffin-Manson equation. As mentioned in Section 3.3, the Coffin-Manson exponent  $q$  is material dependent and the effect of TC is significantly dependent on the  $q$  value. In order to quantify the effect of this value, we compare the two TCs presented in Figure 3, varying  $q$  from 7 to 2. The MTTF gains of the cycle with a smaller amplitude ( $[T_{top}, T_{bot,max}]$ ) to the other one ( $[T_{top}, T_{bot,min}]$ ) were 17.50, 11.42, 7.50, 4.91, 2.21, and 0.52, respectively for  $q = 7$ ,  $q = 6$ ,  $q = 5$ ,  $q = 4$ ,  $q = 3$ , and  $q = 2$ . When  $q$  is relatively big, the MTTF gain is more significant. Conversely, a very small  $q$  value,  $q = 2$  in this case, the reduced amplitude may result in a negative effect in MTTF. This is because that the adverse effects in EM, Tddb, and SM, due to the increased temperature, nullify the gain in TC. However, we could observe a considerable gain in MTTF even with a considerably small  $q$  value, that is, 2.21 X gain in MTTF with  $q = 3$ . The typical choice of  $q$  is 6–9 for brittle fracture (Si and dielectrics), 3–5 for hard metal alloys/intermetallics (Al-Au), and 1–3 for ductile metal (solder) [18]. Therefore, we believe that the proposed technique is effective for the most materials used in the IC and package of the multi-core microprocessors.

## 6. Conclusions and Future Works

In this paper, we presented a lifetime enhancement technique in multi-core satellite embedded systems using virtual tasks and DVFS under power constraints. In LEO CubeSats, the temperature changes greatly and a large TC occurs in the electronic parts, which is a well-known lifetime reliability threat. Conventional reliability enhancement techniques focus on lowering the operating temperature whenever possible, but in LEO CubeSats, the TC can be large and the MTTF may deteriorate. The proposed technique focuses on reducing a large TC amplitude by intentionally consuming more power while considering other high temperature-dependent failure mechanisms. The proposed technique manages the system temperature in three steps. First, it minimizes the peak temperature at the highest PCB temperature to reduce TC amplitude. Second, it derives the minimum temperature (threshold), above which the system's temperature should be maintained, considering the power



budget. Lastly, at runtime, whenever the temperature goes below the threshold, it artificially heats up the processor by means of DVFS and virtual task injections. Experimental results show that the proposed technique improves the MTTF up to 8.03 times in the Nvidia's Jetson TK1 board with a real workload of a small satellite system. In the extension of the proposed technique of no virtual tasks, different frequency range and different Coffin-Manson exponent, we show that our proposed technique is effective in improving the MTTF.

The proposed technique can be further extended as follows to be better utilized in the small satellite systems. Firstly, it has been reported that spatial thermal gradients in the satellite system would result in negative effects in reliability [38]. As the reliability model that we used in this work only focuses on the temporal gradient, it is necessary to investigate how the spatial gradients, for example, between cores, can be analyzed and mitigated in mapping and DVFS. Another future work that can be done is to co-optimize the reliability and power consumption at the same time. Whilst the power budget was given as input and the lifetime is just to be maximized in this work, the mission lifetime can be different from one mission to another in reality. Thus, how much power to be sacrificed can be effectively balanced with the target lifetime during the optimization, which also remains as a future work.

**Author Contributions:** Conceptualization, H.Y. and B.K.; methodology, H.Y. and B.K.; software, B.K.; validation, B.K.; investigation, H.Y. and B.K.; writing—original draft preparation, H.Y. and B.K.; writing—review and editing, H.Y. and B.K.; supervision, H.Y.; project administration, H.Y.; funding acquisition, H.Y.

**Funding:** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2018-0-01424) supervised by the IITP (Institute for Information & communications Technology Promotion).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|          |                                       |
|----------|---------------------------------------|
| COTS     | Commercial-off-the-shelf              |
| CubeSats | Cube satellites                       |
| LEO      | Low earth orbit                       |
| TC       | Thermal cycling                       |
| BAT      | Battery                               |
| EXT      | External                              |
| PCB      | Printed circuit board                 |
| MB       | Motherboard                           |
| FPGAs    | Field-programmable gate arrays        |
| GEO      | Geostationary orbit                   |
| ICs      | Integrated circuits                   |
| EM       | Electromigration                      |
| TDDDB    | Time-dependent dielectric breakdown   |
| SM       | Stress migration                      |
| MTTF     | Mean time to failure                  |
| RAMP     | Reliability-aware microprocessor      |
| NBTI     | Negative bias temperature instability |
| SOFR     | Sum-of-failure-rates                  |
| DPM      | Dynamic power management              |
| DVS      | Dynamic voltage scaling               |
| DTM      | Dynamic thermal management            |
| ECUs     | Electronic control units              |
| DVFS     | Dynamic voltage and frequency scaling |
| EDF      | Earliest-deadline-first               |
| PWL      | Piece-wise linear                     |
| WF       | Worst-fit                             |

## References

- Doncaster, B.; Shulman, J.; Bradford, J.; Olds, J. SpaceWorks' 2016 Nano/Microsatellite Market Forecast. Available online: <https://digitalcommons.usu.edu/smallsat/2016/TS2Launch/1/> (accessed on 9 November 2019).
- Heidt, H.; Puig-Suari, J.; Moore, A.; Nakasuka, S.; Twiggs, R. CubeSat: A New Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation. Available online: <https://digitalcommons.usu.edu/smallsat/2000/All2000/32/> (accessed on 9 November 2019).
- Poghosyan, A.; Golkar, A. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. *Prog. Aerosp. Sci.* **2017**, *88*, 59–83. [CrossRef]
- George, A.D.; Wilson, C.M. Onboard processing with hybrid and reconfigurable computing on small satellites. *Proc. IEEE* **2018**, *106*, 458–470. [CrossRef]
- Vladimirova, T.; Fayyaz, M. Fault-Tolerant Distributed Attitude and Orbit Control System for Space Applications. In Proceedings of the IEEE 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Edinburgh, UK, 6–9 August 2018; pp. 43–50.
- Wirthlin, M. High-reliability FPGA-based systems: Space, high-energy physics, and beyond. *Proc. IEEE* **2015**, *103*, 379–389. [CrossRef]
- Jacobs, A.; Cieslewski, G.; George, A.D.; Gordon-Ross, A.; Lam, H. Reconfigurable fault tolerance: A comprehensive framework for reliable and adaptive FPGA-based space computing. *ACM Trans. Reconfig. Technol. Syst. (TRETS)* **2012**, *5*, 21. [CrossRef]
- Spangelo, S.; Cutler, J. Analytical modeling framework and applications for space communication networks. *J. Aerosp. Inf. Syst.* **2013**, *10*, 452–466. [CrossRef]
- Noca, M.; Jordan, F.; Steiner, N.; Choueiri, T.; George, F.; Roethlisberger, G.; Scheidegger, N.; Peter-Contesse, H.; Borgeaud, M.; Krpoun, R.; et al. Lessons Learned from the First Swiss Pico-Satellite: SwissCube. Available online: <https://digitalcommons.usu.edu/smallsat/2009/all2009/84/> (accessed on 9 November 2019).
- Rossi, S.; Ivanov, A.; Richards, M.; Gass, V. The SwissCube's technologies results after four years of flight. In Proceedings of the International Astronautical Conference, Beijing, China, 23–27 September 2013.
- Kumar, P.; Thiele, L. Thermally optimal stop-go scheduling of task graphs with real-time constraints. In Proceedings of the IEEE 16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011), Yokohama, Japan, 25–28 January 2011; pp. 123–128.
- Chantem, T.; Hu, X.S.; Dick, R.P. Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *19*, 1884–1897. [CrossRef]
- Fisher, N.; Chen, J.J.; Wang, S.; Thiele, L. Thermal-aware global real-time scheduling on multicore systems. In Proceedings of the IEEE 2009 15th IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco, CA, USA, 13–16 April 2009; pp. 131–140.
- Pagani, S. Power, Energy, and Thermal Management for Clustered Manycores. Available online: <https://publikationen.bibliothek.kit.edu/1000063307> (accessed on 8 November 2019).
- Ma, Y.; Chantem, T.; Dick, R.P.; Hu, X.S. Improving system-level lifetime reliability of multicore soft real-time systems. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 1895–1905. [CrossRef]
- Xiang, Y.; Chantem, T.; Dick, R.P.; Hu, X.S.; Shang, L. System-level reliability modeling for MPSoCs. In Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Scottsdale, AZ, USA, 24–29 October 2010; pp. 297–306.
- Srinivasan, J.; Adve, S.V.; Bose, P.; Rivers, J.A. The case for lifetime reliability-aware microprocessors. In Proceedings of the 2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Scottsdale, AZ, USA, 24–29 October 2010; Volume 32, p. 276.
- Failure Mechanisms and Models for Semiconductor Devices. Available online: <https://standards.globalspec.com/std/10047309/jedec-jep-122> (accessed on 9 November 2019).
- Black, J.R. Electromigration—A brief survey and some recent results. *IEEE Trans. Electron Devices* **1969**, *16*, 338–347. [CrossRef]
- Wang, S.; Chen, J.J. Thermal-aware lifetime reliability in multicore systems. In Proceedings of the IEEE 2010 11th International Symposium on Quality Electronic Design (ISQED), San Jose, CA, USA, 22–24 March 2010; pp. 399–405.

21. Srinivasan, J.; Adve, S.V.; Bose, P.; Rivers, J.A. Exploiting structural duplication for lifetime reliability enhancement. In Proceedings of the 32nd International Symposium on Computer Architecture (ISCA'05), Madison, WI, USA, 4–8 June 2005; Volume 33, pp. 520–531.
22. Ciappa, M.; Carbognani, F.; Cova, P.; Fichtner, W. Lifetime prediction and design of reliability tests for high-power devices in automotive applications. *IEEE Trans. Device Mater. Reliab.* **2003**, *3*, 523–528. [\[CrossRef\]](#)
23. Srinivasan, J.; Adve, S.V.; Bose, P.; Rivers, J.A. Lifetime reliability: Toward an architectural solution. *IEEE Micro* **2005**, *25*, 70–80. [\[CrossRef\]](#)
24. Srinivasan, J.; Adve, S.V.; Bose, P.; Rivers, J.A. The impact of technology scaling on lifetime reliability. In Proceedings of the International Conference on Dependable Systems and Networks, Florence, Italy, 28 June–1 July 2004; pp. 177–186.
25. Viswanath, R.; Wakharkar, V.; Watwe, A.; Lebonheur, V. Thermal performance challenges from silicon to systems. *Intel Technol. J.* **2000**, *4*, 1–16.
26. Ukhov, I.; Bao, M.; Eles, P.; Peng, Z. Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems. In Proceedings of the 49th Annual Design Automation Conference, San Francisco, CA, USA, 3–7 June 2012; pp. 197–204.
27. Rosing, T.S.; Mihic, K.; De Micheli, G. Power and reliability management of SoCs. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2007**, *15*, 391–403. [\[CrossRef\]](#)
28. Chantem, T.; Xiang, Y.; Hu, X.S.; Dick, R.P. Enhancing multicore reliability through wear compensation in online assignment and scheduling. In Proceedings of the 2013 IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 1373–1378.
29. Park, S.; Chen, J.J.; Shin, D.; Kim, Y.; Yang, C.L.; Chang, N. Dynamic thermal management for networked embedded systems under harsh ambient temperature variation. In Proceedings of the 2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), Austin, TX, USA, 18–20 August 2010; pp. 289–294.
30. Liu, Y.; Dick, R.P.; Shang, L.; Yang, H. Accurate temperature-dependent integrated circuit leakage power estimation is easy. In Proceedings of the 2007 IEEE Design, Automation & Test in Europe Conference & Exhibition, Nice, France, 16–20 April 2007; pp. 1–6.
31. Biswas, S.; Tiwari, M.; Sherwood, T.; Theogarajan, L.; Chong, F.T. Fighting fire with fire: Modeling the datacenter-scale effects of targeted superlattice thermal management. In Proceedings of the ACM SIGARCH Computer Architecture News, San Jose, CA, USA, 4–8 June 2011; Volume 39, pp. 331–340.
32. Schor, L.; Bacivarov, I.; Yang, H.; Thiele, L. Efficient worst-case temperature evaluation for thermal-aware assignment of real-time applications on MPSoCs. *J. Electron. Test.* **2013**, *29*, 521–535. [\[CrossRef\]](#)
33. Chhabra, R.P. *CRC Handbook of Thermal Engineering*; CRC Press: Boca Raton, FL, USA, 2017.
34. Xian, C.; Lu, Y.H.; Li, Z. Energy-aware scheduling for real-time multiprocessor systems with uncertain task execution time. In Proceedings of the 2007 IEEE 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 664–669.
35. Coskun, A.K.; Ayala, J.L.; Atienza, D.; Rosing, T.S.; Leblebici, Y. Dynamic thermal management in 3D multicore architectures. In Proceedings of the Conference on Design, Automation and Test in Europe, Nice, France, 20–24 April 2009; European Design and Automation Association: Grenoble, France, 2009; pp. 1410–1415.
36. Huang, M.; Suo, Z.; Ma, Q.; Fujimoto, H. Thin film cracking and ratcheting caused by temperature cycling. *J. Mater. Res.* **2000**, *15*, 1239–1242. [\[CrossRef\]](#)
37. Nguyen, V.H. *Multilevel Interconnect Reliability on the Effects of Electro-Thermomechanical Stresses*; University of Twente: Enschede, The Netherlands, 2004.
38. Kirischian, L.; Kirischian, V.; Sharma, D. Mitigation of Thermo-cycling effects in Flip-chip FPGA-based Space-borne Systems by Cyclic On-chip Task Relocation. In Proceedings of the 2018 IEEE NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Edinburgh, UK, 6–9 August 2018; pp. 17–24.

