Power-Efficient Control of Thermoelectric Coolers Considering Distributed Hot Spots

Mohammad Javad Dousti and Massoud Pedram
Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA
{dousti,pedram}@usc.edu

***Abstract*—Thermoelectric coolers are compact devices that can target hot spots on a VLSI die. These devices are connected electrically in series and controlled together, i.e., all are ON or OFF at the same time. However, spatial and temporal distributions of hot spots on a VLSI die are non-uniform, and therefore, activating all of TECs to address one or a few localized hot spots is not economical. This traditional technique indeed leads to a significant power waste. This paper suggests that adjacent hot spots with the same thermal behavior can be grouped and controlled by a cluster of TECs. A bypass switch for each TEC cluster is added in order to allow selectively turning OFF some TEC clusters which are needed. More precisely, a clustering problem is formulated which aims to minimize the power waste due to excessive use of TECs. Due to the large number of variables in problems of interesting sizes, a greedy heuristic method for solving the problem is introduced. It is shown that the proposed heuristic can reduce the wasted power on average by 81% and also decrease the total TEC power consumption on average by 42%.**

I. Introduction

Thermoelectric coolers (TECs) are active devices that work based on the *Peltier effect*. This effect allows the device to absorb the heat from one side and release it to the other side when electrical current passes through it. The amount of cooling is linearly proportional to the amount of driving current. Notable features of TECs are the following. 1) Very high heat pumping rate—It has been shown that thin-film TECs can pump high heat fluxes as large as $\sim$1,300 W/cm^2 [1]. None of traditional cooling techniques has the ability to pump heat fluxes higher than 1,000 W/cm^2 [2]. 2) Compact size—TECs can be built as thin as tens of micrometers and their area can be smaller than 1 mm^2. These devices have the right size to exclusively cover typical hot spots on a chip. 3) Fast response time—Thin-film TECs have very fast response times in the order of a few milliseconds. 4) High reliability—These devices have no moving parts, and hence, can last longer than other active cooling solutions. Commercial TECs are expected to work for more than 11 years [3]. 5) High controllability—TECs can be controlled at the granularity of fractions of a degree of Celsius and can cool down a chip below the ambient temperature.

The unique features of TECs make them a perfect candidate for hot spot cooling. Unfortunately, Joule heating occurs as an adverse phenomenon during the cooling process by TECs, which causes them to dissipate heat when current flows through it. Due to this effect, the utilization of TECs should be limited to the time that their associated hot spot is active (i.e., the hot spot temperature is above the set point). One common approach is to selectively deploy TECs in order to only cover hot spots on the die (as opposed to covering the entire die) [4]. However, hot spots not only have spatially non-uniform distribution [5], but also have temporally non-uniform scattering throughout the die [6]. This temporal non-uniform distribution is due to the fact that each application (or execution phase of an application) utilizes different functional units on the die and hence exhibits different set of hot spots.

This paper aims to minimize the power consumption of TECs by identifying temporal and spatial distribution of hot spots in the chip, and subsequently, turning ON/OFF groups of TECs as needed. In traditional designs, TECs are connected electrically in series, which makes their selective control impossible [1], [2], [7]. We propose adding bypass switches in order to allow independent control of TECs. Note that bypass switches are not ideal and have an ON-resistance comparable to that of a single TEC. Hence, they consume power when current flows through them. Besides, excessive use of bypass switches increases the cost of cooling system. Thus, there is a trade-off between the number of switches and the power saving provided by this approach.

To address this challenge, adjacent hot spots with exactly same thermal behavior may be grouped and cooled by a few nearby TECs. We refer to these TECs as a *TEC group*. Consequently, each TEC group based on their distance from each other and the temporal behavior of these hot spots can be controlled by a single switch. More precisely, we formulate a clustering problem as an integer-quadratic program that minimizes the wasted power consumption of TECs and bypass switches. This problem has many variables which makes it impossible to be solved directly. Hence, we introduce a greedy heuristic to solve the clustering problem. This heuristic is executed during the design of the cooling system. Subsequently, during the runtime of the system, if at least a hot spot corresponding to a cluster is present, the bypass switch of that cluster should be off (does not bypass). Otherwise, the switch should be on (bypass mode). It is shown that the suggested heuristic can reduce the wasted power on average by 81% and also decrease the total TEC power consumption on average by 42%.

The rest of this paper is organized as follows. Section II overviews the thermoelectric cooling principles and surveys the related work. Section III introduces a control circuit for selective control of TEC clusters. Section IV formulates a clustering problem to minimize the wasted cooling power consumption. Then it presents a greedy heuristic for clustering TECs. Section V presents the experimental results. Finally, Section VI concludes the paper.

II. Background and Related Work

A. Principles of Thermoelectric Cooling

Thermoelectric coolers are compact devices, which are incorporated inside a cooling package in order to improve the

cooling efficiency. One common setup is shown in Fig. 1(a). TECs are made of pairs of N-type and P-type semiconductor pellets. When current flows through a P-type pellet (from the positive terminal to the negative terminal), heat flows in the same direction, i.e., heat is absorbed from the positive side, which is called *cold side*, and released to the negative side, which is called *hot side*. The heat flow direction in an N-type pellet is the reverse of that of the P-type pellet. The heat pumping direction can be changed by simply reversing the current direction. This feature makes TECs ideal for the applications where bi-directional cooling is required.

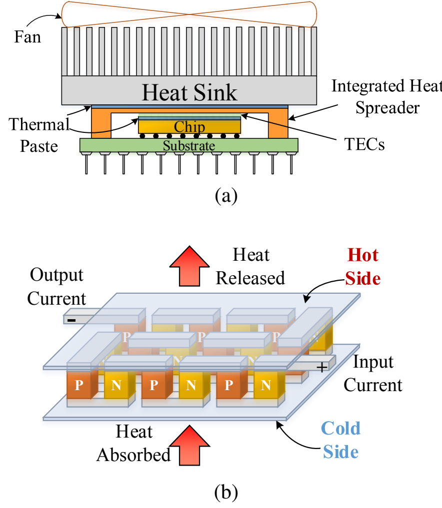

Fig. 1. (a) A chip cooling assembly equipped with TECs (b) A 3×3 array of TECs.

Usually several N-P pairs of pellets are connected electrically in series and thermally in parallel to increase the amount of heat rejection. Fig. 1(b) shows a 3×3 array of TECs (a total of 9 N-P pairs).

The heat absorbed and released per unit time from the cold side and to the hot side are denoted by $\dot{q_c}$ and $\dot{q_h}$, respectively. They can be calculated as

$$\dot{q_c} = N(\alpha T_c I - \frac{1}{2} R_{TEC} I^2 - K_{TEC} \Delta T), \text{and} \quad (1)$$

$$\dot{q_h} = N(\alpha T_c I + \frac{1}{2} R_{TEC} I^2 - K_{TEC} \Delta T), \quad (2)$$

where N is the number of TECs connected electrically in series, α is the *Seebeck coefficient*, T_c is the temperature of the cold side (in Kelvin), I is the current which flows through the TEC, ΔT is the temperature difference between the hot side and the cold side (i.e., $T_h - T_c$), and R_{TEC} and K_{TEC} are the electrical resistance and thermal conductance of the TEC, respectively. The first term in this equation captures the *Peltier effect* which represents the cooling phenomenon, the second term represents the *Joule heating effect*, and the last term signifies the heat conductivity from hot side to the cold side. The second and the last terms have adverse effects in the cooling applications and hence have a negative sign. Note that the Joule heating quadratically depends on the current whereas the Peltier effect linearly depends on it.

Power consumption of N TECs is the difference between $\dot{q_h}$ and $\dot{q_c}$ and may be written as follows. Note that the term corresponding to the Thomson effect is dropped due to its negligible effect.

$$P_{TEC,N} = \dot{q_h} - \dot{q_c} = N(P_{TEC}) = N(R_{TEC} I^2 + \alpha \Delta T I) \quad (3)$$

B. Related Work

Many studies have been conducted in the area of thermoelectric cooling. Most of them focus on improving the material that the device is made of and the manufacturing process. Reference [2] presents a comprehensive survey on TEC principles and the manufacturing advances in recent years.

Long *et al.* [4] formulates the selective deployment of TECs on top of a chip in order to achieve the maximum cooling (lowest temperature). The motivation is that excessive deployment of TECs adversely affects the temperature of the device because of lateral heating among TECs. Moreover, deploying unnecessary TECs increases the power consumption of the cooling solution. This work considers only the spatial distribution of TECs.

Same authors in [8] suggest independent control of TECs by multiple current sources. Adding several current sources leads to the addition of input pins to the chip which is costly. Consequently, authors show that with only three or four independent current sources, the temperature of the hottest spot on average is no higher than the case where infinite number of current sources are available (i.e., each TEC can be controlled independently) by 0.6 °C or 0.3 °C, respectively. Similar to their prior work [4], [8] has not considered the temporal hot spot distribution. Moreover, the target processor in these two articles is a single-core processor, which does not exhibit significant non-uniformity in temporal and spatial distributions of hot spots compared to a multi-core processor.

In our prior work [9], we consider power-aware controlling of a cooling system comprised of TECs and a fan. The optimization knobs are the fan rotation speed and TECs driving current. The technique presented in that paper is orthogonal to that of this paper and can be combined for further power saving.

III. Selective Control of TECs

Different applications (or execution phases of an application) may stress different functional units of a multi-processor system-on-chip (MPSoC), such as register file, ALU, etc. This results in a non-uniform temporal distribution of hot spots on the chip. As it is explained previously, the common practice for deploying TECs is to only consider the spatial distribution of TECs by running a set of applications on the target chip and identifying hot spots accordingly. Next, one or several TECs will be assigned to each hot spot [4].

TECs are often connected in series. This requirement tends to result in a vast amount of power loss due to the fact that all TECs should remain ON as long as even one hot spot on the chip is present. The reason for serially connecting TECs is as follows. A TEC can tolerate at most 60 mV voltage difference between its two terminals while allowing 5 A or more current to pass through it [3]. Hence, connecting tens or hundreds of TECs in parallel requires hundreds to thousands of Ampers to be supplied by the current source while maintaining the voltage difference at tens of millivolts. Clearly, this is not possible, and thus, TECs are connected serially so that the overall current remains low and the controlling voltage across the input terminal of the first TEC in the chain and the output terminal of the last TEC in that chain is large enough so that it can be simply maintained.

Assume that N_{TEC}^{total} TECs are deployed and on average, N_{TEC}^{avg} of them are being used. Hence, power in the amount

of

$$P_{waste}^{avg} = (N_{TEC}^{total} - N_{TEC}^{avg})P_{TEC} \quad (4)$$

will be wasted on average. Our experiments revealed that the coefficient of P_{TEC} in this equation can be as large as $N_{TEC}^{total}/2$. In other words, half of the TECs' power consumption is wasted. Any technique for saving this wasted power can be simply evaluated by comparing the amount of saving it provides with respect to P_{waste}^{avg} (given in Eq. (4)).

In this paper, we propose selective control of TECs in order to eliminate the power consumption of inactive TECs. This can be achieved by using *bypass switches*, which are controlled through a chain of flip-flops (FFs), which determines the status of each switch. Using these FFs reduces the number of required control signals to three—an input to the first FF in the chain, a clock, and an enable signal. Fig. 2 shows the proposed control circuit. In this figure, when clock (*Clk*) is applied to the circuit, the configuration bits can be shifted in through *Config*. When all of configuration bits are scanned in, the enable signal (*EN*) is activated for one period of clock. This activates FFs in the second row, which causes the desired configuration to be loaded into them and consequently be applied to bypass switches.

We use PMOS switches in order to achieve perfect passing of the input signal. Note that the power consumption of FFs is negligible compared to that of TECs and bypass switches. Moreover, state-of-the-art switches are very fast and can be turned on and off in 663 µs and 2 µs, respectively [10]. These delay values are sufficiently smaller than the thermal constant of a chip which is in the order of seconds [11].

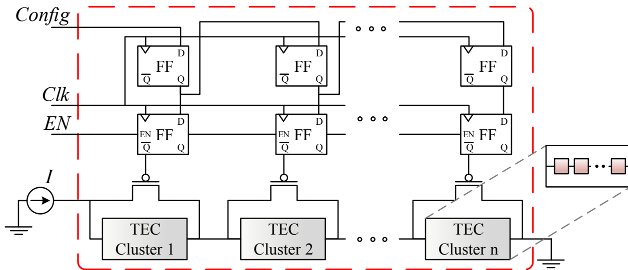

Fig. 2. The proposed circuit for selective control of TEC clusters

One tempting idea is to deploy a switch for every TEC in order to control them with fine granularity. This idea has two negative consequences. First, it increases the cost of the cooling system. Second, due to the fact that bypass switches are not ideal, aggressive deployment of them will not produce the best result. In other words, bypass switches exhibit a small resistance (in the order of a few mΩ [10]), which is comparable to the resistance of a TEC (about 5 mΩ [12]). Hence, in order to reduce the power loss in switches, we suggest *clustering* TEC groups. These TECs should correspond to adjacent hot spots that exhibit temporally similar thermal behavior. Placing two non-adjacent TECs inside a cluster makes the routing of the power supply line quite difficult and expensive. Moreover, the power line will exhibit a significant ohmic resistance and hence, wastes more power and produces heat due to the Joule heating. Clustering not only reduces the power loss in the switches, but also saves the cost required to add multiple switches and their corresponding control circuitry (e.g., FFs).

The total wasted cooling power consumption is divided into two parts. First, the power wasted to keep clusters active while only a few TECs are actually necessary to cool the active hot spots on the chip. This is denoted by P_{on}. Second, the power wasted due to the non-ideality of switches, which is denoted by P_{off}. The power wasted in a single OFF cluster with n TECs is calculated as

$$\frac{nR_{DS,ON}}{nR_{TEC} + R_{DS,ON}}(R_{TEC}I^2 + \alpha\Delta TI) \quad (5)$$

where $R_{DS,ON}$ is the ON-resistance of a bypass switch. The first term inside brackets corresponds to the ohmic power dissipation. $\frac{nR_{TEC}R_{DS,ON}}{nR_{TEC}+R_{DS,ON}}$ is the equivalent resistance (R_{eq}) for a cluster of TECs paralleled with a bypass switch. The second term is due to the TEC power dissipation only (cf. Eq. (3)). P_{off} can be calculated as a summation over Eq. (5) written for every OFF cluster. Clearly, P_{off} is minimized when n is maximized, i.e., all of TECs are grouped into one cluster. On the other hand, increasing n reduces the controllability of TECs, and consequently, increases P_{on}. Hence, there should be an optimum clustering solution for which the wasted power consumption for driving TECs (i.e., $P_{on} + P_{off}$) is minimized. Next section will present an optimization problem that finds the optimum clustering to minimize the wasted power consumption.

IV. TEC Clustering

A. Problem Formulation

We assume that an exhaustive set of applications are executed on the target chip and their corresponding temperature maps are derived. This step, which is referred as the *benchmarking step*, is a common practice for determining best locations to place temperature sensors on the die (for instance see [6] and [13]). Based on the derived temperature map(s) for each application, a set of intervals for hot spots can be determined. Each interval depicts a hot spot which is present during the execution of an application (i.e., its temperature is higher than a certain threshold). Given this thermal information, one may find a near-optimum minimum set of TECs required for cooling down these hot spots as explained in [4].

Next, these TECs are grouped such that members of each group always behave similarly, i.e., they target the same hotspot. Due to the small size of TECs, we assume that each hot spot can be cooled down by one or several TECs. Finally, TEC groups are clustered and each cluster is controlled by a bypass switch in order to minimize the wasted power consumption. Note the difference between grouping and clustering. Grouping is done to target a single hot spot, whereas clustering targets multiple hot spots. A cluster contains one or several TEC groups.

The TEC clustering problem may be formulated as follows. In this optimization problem, $x_{i,k}$ and $y_{k,a}$ are optimization variables which are binary. $x_{i,k}$ is equal to 1 if TEC group i is assigned to cluster k and 0, otherwise. $y_{k,a}$ is equal to 1 if at least one TEC group assigned to cluster k remains ON during the execution of application a and 0, otherwise.

$$\min_{x_{i,k},y_{k,a}} \sum_{k=1}^{N_{clust}} \left(\frac{\sum_{a=1}^{N_{app}} \left(P_{on}^{k,a} + P_{off}^{k,a} \right)}{N_{app}} + P_{penalty}^{k} \right) \quad (6)$$

where:

$$P_{on}^{k,a} = \sum_{i=1}^{N_{grp}} x_{i,k}(y_{k,a} - A_{i,a})c_i P_{TEC} \quad (7)$$

$$P_{TEC} = R_{TEC}I^2 + \alpha \Delta T I \tag{8}$$

$$P_{off}^{k,a} = (1 - y_{k,a})\Big(\frac{n_k^{TEC} R_{DS,ON}}{n_k^{TEC} R_{TEC} + R_{DS,ON}} \times (R_{TEC}I^2 + \alpha \Delta T I)\Big) \tag{9}$$

$$n_k^{TEC} = \sum_{i=1}^{N_{grp}} c_i x_{i,k} \tag{10}$$

$$P_{penalty}^{k} = \sum_{i=1}^{N_{grp}} \sum_{j=i+1}^{N_{grp}} x_{i,k} x_{j,k} w_{i,j} \tag{11}$$

subject to:

$$\sum_{k=1}^{N_{clust}} x_{i,k} = 1, \quad \forall i \in \{1,..,N_{grp}\} \tag{12}$$

$$x_{i,k}(y_{k,a} - A_{i,a}) \geq 0, \quad \forall i \in \{1,..,N_{grp}\}, \forall a \in \{1,..,N_{app}\}, \forall k \in \{1,..,N_{clust}\} \tag{13}$$

$$x_{i,k} \in \{0,1\}, \forall i \in \{1,..,N_{grp}\}, \forall k \in \{1,..,N_{clust}\} \tag{14}$$

$$y_{k,a} \in \{0,1\}, \forall a \in \{1,..,N_{app}\}, \forall k \in \{1,..,N_{clust}\} \tag{15}$$

The definition of parameters in this problem follows. $P_{on}^{k,a}$ and $P_{off}^{k,a}$ are the wasted power consumption values during the execution of application a when cluster k is ON (i.e., its corresponding switch is open) and OFF (i.e., its corresponding switch is closed), respectively. $P_{penalty}^{k}$ is the penalty value imposed due to the distance of TEC groups assigned to cluster k. N_{clust} is the maximum number of clusters that are allowed to be formed. This value is determined by the cooling package budget. N_{grp} is the total number of TEC groups. N_{app} is the number of representative applications executed on the VLSI chip during the benchmarking step. c_i is the number of TECs inside TEC group i. $A_{i,a}$ determines if TEC group i is active during the execution of application a or not. The value of this parameter may be found from the set of intervals derived earlier. $w_{i,j}$ is a monotonically increasing function of the distance of TEC groups i and j which placed in the same cluster. $R_{DS,ON}$, α, ΔT, R_{TEC}, and I are defined as before.

The objective function (shown in Eq. (6)) tries to minimize the average total power waste among clusters when they are unnecessarily ON ($P_{on}^{k,a}$), the average total power waste due to the non-ideality of bypass switches for each cluster ($P_{off}^{k,a}$), and a penalty value ($P_{penalty}^{k}$) imposed due to the non-locality of TEC groups inside a cluster.

Eq. (7) defines $P_{on}^{k,a}$. Basically, a TEC group wastes power when it is OFF (i.e., $A_{i,a} = 1$), but its corresponding cluster k is ON during the execution of application a (i.e., $y_{k,a} = 1$.) P_{TEC} is defined in Eq. (8) similar to Eq. (3).

Next, Eq. (9) defines $P_{off}^{k,a}$. Note that $P_{off}^{k,a}$ has a non-zero value only if cluster k is OFF during the execution of application a ($y_{k,a} = 0$.) This equation is written similar to Eq. (5); however, instead of n, n_k^{TEC} is used which is defined as the total number of TECs in cluster k (cf. Eq. (10)).

Eq. (11) defines $P_{penalty}^{k}$. This value is the summation of penalty values among every pair of TEC groups assigned to cluster k. $w_{i,j}$ is a function of the distance between TEC groups i and j. In this paper, we define it as follows. If TEC groups i and j are not *adjacent*, i.e., farther apart than distance r from each other, $w_{i,j}$ is set to a large positive number; otherwise, it is zero. With this definition we make sure that TEC groups with the distance of more than r from each other are not assigned to same the cluster, unless N_{clust} is set to a very small number.

Constraint (12) ensures that all of TEC groups are assigned to a cluster. Constraint (13) assures that cluster k is active during the execution of application a, if TEC group i is active during the same period and it is assigned to cluster k (i.e., $x_{i,k} = 1$). Finally, constraints (14) and (15) ensure $x_{i,k}$ and $y_{k,a}$ are binary variables.

B. Proposed Solution

The problem formulation presented above is not a standard 0-1 *integer-quadratic program* (0-1 IQP). The only non-quadratic term is the definition of P_{off}^{k} in Eq. (9). Due to the nature of the problem, the total number of TECs in a cluster is large and hence $n_k^{TEC} R_{TEC} \gg R_{DS,ON}$. Thus,

$$P_{off}^{k,a} \approx (1 - y_{k,a})\left(R_{DS,ON}I^2 + \frac{R_{DS,ON}}{R_{TEC}}\alpha \Delta T I\right). \tag{16}$$

Using the aforementioned simplification, the problem formulation becomes a standard 0-1 IQP. Unfortunately, the number of optimization variables are too high for problems of interesting sizes. More precisely, this problem has a total of $N_{clust}(N_{app} + N_{grp})$ binary variables. Clearly, it cannot be optimally solved for reasonable values of N_{clust}, N_{app}, and N_{grp}.

Hence, we propose a greedy heuristic to solve the clustering problem. The pseudocode of this heuristic is listed in Algorithm 1. As it will be shown in the experimental results, this heuristic allows us to save substantial amount of wasted power by generating high-quality clustering solutions. Note that the heuristic is executed during the design of the cooling system and hence, have no impact on the performance and power consumption of the system. Subsequently, during the runtime of the system, if at least a hot spot corresponding to a cluster is present, the bypass switch of that cluster should be off (does not bypass). Otherwise, the switch should be on (bypass mode).

First, (lines 1–5), z_i is calculated for each TEC group which represents how many times any TEC in group i is active during the execution of N_{app} applications. In this heuristic, TEC groups which have higher z_i are processed first, since they are more *critical*. This is done through sorting $\mathbb{G}$ based on calculated z_i values (line 6).

The rest of the algorithm is straightforward. It picks a TEC group from $\mathbb{G}$ and tries every cluster that is already formed in $\mathbb{S}$. It chooses the cluster which has the lowest overhead (i.e., $p_{on} + p_{off} + p_{penalty}$). Next, it compares this value with the overhead of creating a new cluster and decides which one increases the power waste minimally (line 10). Note that this line avoids the excessive deployment of switches when N_{clust} is set high to a large number. Subsequently, it either adds the selected TEC group in a previously created cluster or create a new one and add the TEC group to it. This process continues until all of TEC groups are clustered.

Note that P_{on} and P_{off} are functions of the driving current (I) and P_{on} is a function of the temperature difference (ΔT) as well. In our greedy solution, we assume that these values are constant and we study the dependence of the final solution on them through sensitivity analysis in the next section.

Algorithm 1 A greedy heuristic for clustering TEC groups

Input: Set of TEC groups $\mathbb{G}$, set of intervals $\{A_{i,a}\}$, penalty function $w_{i,j}$, TEC related parameters (I, α, R_{TEC}, ΔT), N_{clust}, N_{grp}, N_{app}, c_i, and $R_{DS,ON}$
Output: Clustering of TEC groups ($\mathbb{S}$)

1: **foreach** $g_i \in \mathbb{G}$ **do**
2: $\quad z_i \leftarrow 0$
3: $\quad$ **for** $a = 1$ **to** N_{app} **do**
4: $\quad\quad$ **if** $A_{i,a} = 1$ **then**
5: $\quad\quad\quad z_i \leftarrow z_i + c_i$
6: Sort $\mathbb{G}$ based on the respective z_i's in descending order
7: Initialize $\mathbb{S}$ with an empty cluster
8: **foreach** $g_i \in \mathbb{G}$ **do**
9: $\quad [cluster, p_{waste}] \leftarrow$ Find the cluster in $\mathbb{S}$ that adding g_i to it minimally increases the power waste
10: $\quad$ **if** $|\mathbb{S}| < N_{clust}$ **and** (power waste of adding g_i to a new cluster) $< p_{waste}$ **then**
11: $\quad\quad new_cluster \leftarrow \{g_i\}$
12: $\quad\quad \mathbb{S} \leftarrow \mathbb{S} \cup \{new_cluster\}$
13: $\quad$ **else**
14: $\quad\quad cluster \leftarrow cluster \cup \{g_i\}$
15: **return** $\mathbb{S}$

V. Experimental Results

A. Simulation Setup

It is expected that a multi-core processor exhibit more non-uniformly distributed hot spots compared to a single-core processor. Hence, we selected a quad-core Intel Xeon X5550 running at 3.06GHz [14] as the target chip. *SniperSim 5.3* [15] is used as the performance simulator. It utilizes *McPat 1.0* [16] to generate area and power information for a given processor and a benchmark. Using the area and power information that McPat provides, we employ *HotFloorplan* [17] in order to generate a temperature-aware floorplan for the Xeon X5550 processor. The aspect ratio of this floorplan is 19.1 mm×10.2 mm. PARSEC benchmark suite [18], which consists of multi-threaded applications, is chosen to be executed on the processor.

For thermal simulations, we use *Teculator* which is a thermal simulator capable of simulating TECs [12]. Teculator and some other prior art such as [19] assume that leakage power depends only on the temperature and area. We improved this simple model by utilizing McPat leakage models for various functional units. This captures the dependence of the leakage current to the circuit implementation of each functional unit. For instance, the leakage current of ALU differs from that of instruction cache (ICache).

A cooling package similar to Fig. 1(a) is adopted for the simulations. Main component characteristics are taken from [12]. A fan with RPM of 1000 is also used for improving the cooling efficiency of TECs. TEC properties are taken form [1], which is the state-of-the-art in the thermoelectric cooling technology.

Moreover, we select Texas Instrument's TPS22920L [10] as the bypass switch, which has $R_{DS,ON}$=5 mΩ and a very large $R_{DS,OFF}$. We also set ΔT=10 K, and r=4.775 mm (i.e., a quarter of the floorplan width) unless otherwise is mentioned. Furthermore, I is set to 3 A, which is close to their optimal operation point in our setup.

The proposed heuristic is implemented in Java and executed on a machine with Intel Core-i7 3770 running at 3.4 GHz with 8GB RAM. The runtime of the program is less than one second, which is spent during the design time.

B. Simulation Results

First, we simulated the thermal behavior of the Xeon processor by running PARSEC benchmarks. These simulations provide a temperature map for each benchmark. As it is expected, some benchmarks exhibit quite different temperature maps and some quite similar maps compared to the others. Using these temperature maps, a list of hot spots for each benchmark is made. We consider a spot as hot if its temperature exceeds 85 °C. Next, we find the minimum number of TECs that are sufficient to cool down these spots. This provides a baseline which only considers the spatial distribution of hot spots.

Using these data, a set of intervals is obtained as shown in Fig. 3. Each TEC group is named using the processing core and the functional unit it targets. The number of TECs inside each group is determined in parenthesis. A green block shows that a TEC group should be active during a benchmark in order to maintain the temperature of the hottest spot on the die below 85 °C. For instance, TECs in group *Core1-Mem* are required to be active only during the execution of benchmark *raytrace*.

The set of intervals allows us to calculate N_{TEC}^{total} and N_{TEC}^{avg} and consequently use Eq. (4) to derive an upper bound for saving the wasted power (i.e., P_{waste}^{avg}). Hence, $N_{TEC}^{total} = 76$ and $N_{TEC}^{avg} = 36.8$. This means on average, 39 TECs are unnecessarily ON. Assuming TEC parameters given in [1], the total cooling power consumed by TECs is equal to 3.76 W and P_{waste}^{avg} =1.94 W, which is 52 % of the total TEC power consumption. In this section, we report the power saving normalized to this value (i.e., 1.94 W). The power saving compared to the overall TEC power consumption is roughly half of the reported values (due to the 52 % derived above).

TEC Groups
Core 0: IS (8), RU (1)
Core 1: IS (7), RBB (6), RU (1), Mem (2), ICache (6)
Core 2: IS (8), RBB (6), RU (1), Mem (2), ICache (6)
Core 3: IS (6), RBB (6), RU (2), Mem (2), ICache (6)
blackscholes, bodytrack, canneal, dedup, facesim, fluidanimate, freqmine, raytrace, streamcluster, swaptions, vips, x264

Fig. 3. A set of intervals drawn for PARSEC benchmarks executed on Xeon X5550

Fig. 4(a) shows the power saving achieved by utilizing at most N_{clust} clusters. Note that $N_{clust} \leq N_{grp}$, since the cluster count cannot be larger than the number of TEC groups. The power saving saturates at almost 81% for N_{clust}>13. Also, the power saving advantage slows down as N_{clust} increases.

Fig. 4(b) shows how power saving shown in Fig. 4(a) is achieved. This figure depicts the break down of the wasted power into two elements: P_{off} and P_{on}. As can be seen, for

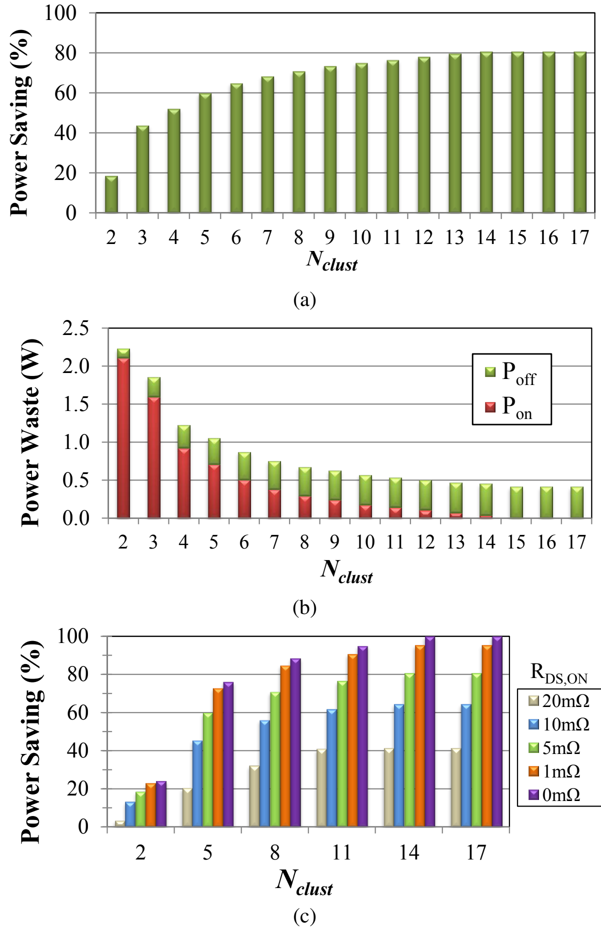

Fig. 4. (a) The power saving percentage achieved by clustering TECs using the proposed heuristic (b) Power waste break down (c) Sensitivity analysis of power saving with respect to $R_{DS,ON}$

small values of N_{clust}, P_{on} is the main portion of the wasted power. However, as N_{clust} increases, TECs can be selectively controlled. Hence, P_{on} decreases. On the other hand, P_{off} increases, since more bypass switches are employed. With $N_{clust} > 14$, P_{off} is the only contributor to power waste.

To study the effect of $R_{DS,ON}$ on the power saving, we find the power saving achieved for different values of $R_{DS,ON}$ and N_{clust}. Fig. 4(c) shows the result. As it is expected, smaller values of $R_{DS,ON}$ provides higher power saving. Especially, in the case of $R_{DS,ON} = 0\,\Omega$, $100\,\%$ power saving can be achieved with 14 clusters.

We also varied I and ΔT and found the power saving percentage. However, no change is seen. Note that the actual power saving changes but the power saving percentage which is normalized to P^{avg}_{waste} does not vary. This is due to the fact that P^{avg}_{waste} is a function of P_{TEC}, which itself is a function of I and ΔT (cf. Eq. (4)). As a result, one can select an arbitrary but reasonable value (non-zero and not too high) for I and ΔT and use the proposed heuristic to find a good clustering solution. Also, we would like to point out that due to the way our method is constructed, always the thermal constraint of chip was met.

VI. Conclusion

This paper considered the non-uniform spatial and temporal distributions of hot spots on a VLSI die as one of the key sources of power waste in TECs. It proposed that adjacent hot spots with the same thermal behavior can be grouped and controlled by a cluster of TECs. A bypass switch for each TEC cluster is added in order to allow selectively turning OFF some TEC clusters which are needed. More precisely, a clustering problem was formulated as an integer-quadratic program which aims to minimize the power waste due to excessive use of TECs. Due to the large number of variables in problems of interesting sizes, a greedy heuristic method for solving the problem was introduced. It was shown that the proposed heuristic can reduce the wasted power on average by 81% and also decrease the total TEC power consumption on average by 42%.

Acknowledgment

This project is funded in part by the Computer Systems program of the Division of Computer and Network Systems of the NSF CISE.

References

[1] I. Chowdhury *et al.*, "On-chip cooling by superlattice-based thin-film thermoelectrics," *Nature Nanotechnology*, vol. 4, no. 4, pp. 235–238, 2009.

[2] A. Bar-Cohen and P. Wang, "On-chip thermal management and hot-spot remediation," in *Nano-Bio- Electronic, Photonic and MEMS Packaging*. Springer, 2010.

[3] "Tellurex - An introduction to thermoelectrics," http://www.tellurex.com/technology/design-manual.php.

[4] J. Long *et al.*, "Optimization of an on-chip active cooling system based on thin-film thermoelectric coolers," in *DATE*, 2010.

[5] A. Watwe and R. Viswanath, "Thermal implications of non-uniform die power and CPU performance," in *InterPACK*, 2003.

[6] S. H. Gunther *et al.*, "Managing the impact of increasing microprocessor power consumption," *Intel Technology Journal*, vol. 5, no. 1, pp. 37–45, 2001.

[7] D. M. Rowe, *CRC Handbook of Thermoelectrics*, 1st ed. CRC-Press, Sep. 1995.

[8] J. Long and S. Memik, "A framework for optimizing thermoelectric active cooling systems," in *DAC*, 2010.

[9] M. J. Dousti and M. Pedram, "Power-aware deployment and control of forced-convection and thermoelectric coolers," in *Design Automation Conference*, Jun. 2014.

[10] "TPS22920L - Texas Instruments," http://www.ti.com/product/tps22920l.

[11] V. Tiwari *et al.*, "Reducing power in high-performance microprocessors," in *DAC*, 1998.

[12] M. J. Dousti and M. Pedram, "Platform-dependent, leakage-aware control of the driving current of embedded thermoelectric coolers," in *ISLPED*, 2013.

[13] J. Long *et al.*, "Thermal monitoring mechanisms for chip multiprocessors," *ACM Trans. Archit. Code Optim.*, vol. 5, no. 2, pp. 9:1–9:33, Sep. 2008.

[14] "ARK | Intel® Xeon® Processor X5550," http://ark.intel.com/products/37106/Intel-Xeon-Processor-X5550-8M-Cache-2_66-GHz-6_40-GTs-Intel-QPI.

[15] T. E. Carlson *et al.*, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *SC*, 2011.

[16] S. Li *et al.*, "The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 1, pp. 5:1–5:29, Apr. 2013.

[17] M. R. Stan *et al.*, "HotSpot: a dynamic compact thermal model at the processor-architecture level," *Microelectronics Journal*, vol. 34, no. 12, pp. 1153–1165, Dec. 2003.

[18] C. Bienia *et al.*, "The PARSEC benchmark suite: Characterization and architectural implications," in *PACT*, 2008.

[19] S. Biswas *et al.*, "Fighting fire with fire: modeling the datacenter-scale effects of targeted superlattice thermal management," in *ISCA*, 2011.