Fast Multi-Design Thermal-Aware Placement Planning for Reconfigurable Systems using Power Budgeting

Shahin Golshan[1], Houman Homayoun[2] and Eli Bozorgzadeh[1]

[1] Center of Embedded Computer Systems
University of California, Irvine CA, 92697, USA
{ssgolsha, eli}@ics.uci.edu

[2] Department of Computer Science and Engineering
Univeristy of California, San Diego, CA, 9500, USA
hhomayou@eng.ucsd.edu

Abstract

In this paper, a novel multi-design thermal-aware dynamic placement planner for reconfigurable systems is presented, which targets transient temperature reduction. Rather than solving time-consuming differential equations to obtain the hotspots, we propose a fast and accurate heuristic model based on power budgeting to plan the dynamic placements of a stream of design statically, while considering the boundary conditions. We provide detailed mathematical analysis of our model for power budgeting which highlights the effectiveness of our methodology. Based on our heuristic model, we have developed a fast optimization technique to plan the dynamic placements at design time. Our results indicate that our technique is two orders of magnitude faster while the quality of the placements generated in terms of temperature and interconnection overhead is the same, if not better, compared to the thermal-aware placement techniques which perform thermal simulations inside the search engine.

Categories and Subject Descriptors:

J.6 [Computer Aided Design (CAD)]

General Terms:

Algorithms, Design, Reliability

Keywords:

Reconfigurable Systems, Temperature, Dynamic Reconfiguration, Placement, Computer Aided Design

1. Introduction

Reconfigurable System-on-Chip (RSoC) platforms are widely used in many application domains as they offer high performance, high flexibility in design and fast reconfiguration time. The ever-increasing demand for more computations in shorter times as well as technology scaling in these systems will increase the power density and consequently increase the operating temperature. High temperature heavily impacts the reliability reliability, performance and cost. The failure rate of the SoC depends exponentially upon operating temperature [1]. High operating temperature can drastically undermine the life span of the device as well [2]. High temperature also lowers the performance. Increases in interconnect delay due to high temperature degrade performance, which in turn lowers the operating frequencies, or in the worst case scenario, leads to timing failures in the design. Increases in leakage power caused by temperature rises can lead to thermal run-away. The cost of cooling solutions in chip packaging increases drastically with power density increases [3] and therefore such solutions might be prohibitively expensive in practice.

The work in this paper is partially supported by the NSF under the award CNS-CAREER #0846129.

In order to cope with the increasing demand for higher computation capacities in reconfigurable systems, several dynamic thermal management (DTM) schemes have been proposed [4,5,6]. In general, DTM techniques try to buy in more power saving by compromising the computation speed. Dynamic voltage/frequency scaling (DVFS) is a common example of DTM approaches which exploits the timing slacks in favor of power saving.

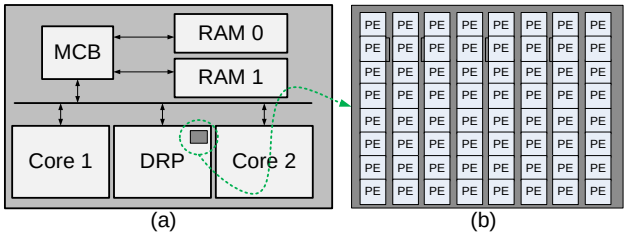


Figure 1: (a) Sample RSoC (b) Single tile of the DRP

Reconfiguration capabilities in RSoC provide a new and orthogonal dimension to all previous DTM approaches. Rather than throttling the execution speed to lower the power dissipation in RSoC, reconfiguration allows transporting the computation from hotspots to cooler locations on the chip.

In this paper, we propose to statically plan and place a stream of designs to be configured on the RSoC so that the peak temperature of the stream below the critical temperature. Our placement planner generates multiple versions of each design and reconfigures each placement on the dynamic reconfigurable system. Reconfigurable systems can be abstracted as arrays of processing elements (PE). A prominent example of such reconfigurable systems is Dynamically Reconfigurable Processor (DRP) [7,8]. The overhead of reconfiguration of the DRP is a single clock cycle. Therefore, rather than confining the implementation to a single version of a design, we exploit the reconfiguration capabilities of RSoC to reduce the peak transient temperature. At design time, new placements for the steam of the design are planned at certain checkpoints to be executed sequentially on the dynamic reconfigurable system. Since we are aiming at peak transient temperature reduction rather than steady-state temperature minimization, instantaneous changes in the power densities of the applications can be captured and handled through dynamic reconfiguration.

Given the layout of the processing elements on a reconfigurable system, temperature simulation tools can solve the differential equation of Fourier's heat flow model to obtain the temperature value based on the power consumptions over a period of time [9].

However, solving the complex finite difference equations is very time-consuming. Here we motivate an alternative methodology which delves into the problem taking an opposite approach. Given the temperature constraints, we compute the maximum power for each processing element that keeps the temperature of the processing element below the critical temperature. Then, based on the power budgets calculated, we can infer the hotspots of a placement by solely taking into account the power consumptions of the PEs rather than going through time-consuming differential equation solvers. As a result, the optimization tools would no longer need to perform time-consuming temperature simulations but rather to apply an efficient and yet sufficiently accurate (and thermally safe) technique to distribute the power densities to PEs. In order to achieve this, we propose thermal-aware *power budgeting*[1]. Based on the thermal model, we introduce the notion of *critical power* for each PE, which indicates the maximum safe power density permissible for each PE so that the temperature does not exceed a certain threshold. As long as the power densities of each operation on the PEs do not exceed this budget, the temperature does not exceed the threshold. In a similar fashion, we also introduce the concept of *minimal safe temperature*, which can be viewed as the minimal upper bound of the temperatures of all the elements in the reconfigurable system.

Our proposed solution not only considers the power densities of the PEs in the DRP architecture but also captures the varying power densities of surrounding hard IPs as shown in Figure 1. Hence, reconfiguration in DRP can be invoked when inflexible surrounding hard IPs are contributing to hotspots in DRP as well.

In our previous paper, we have introduced our thermal-aware placement planning framework for RSoC [10]. In this paper, we extend our temperature-power model to accurately capture the interaction between leakage power and temperature as well as the reconfiguration power. We also prove some interesting properties of the critical power through detailed mathematical analysis, which provide a solid foundation for our methodology. We also generalize our framework of thermal-aware placement planner to handle streams of multiple designs to be reconfigured on RSoC. We provide detailed studies on the efficiency of our proposed multi-design thermal-aware dynamic placement planner for RSoC. Our experimental results indicate that our methodology outperforms other techniques in terms of the execution time and also quality of solution.

2. Related Work

In general, temperature-aware techniques can be categorized into design time (static) or dynamic (online) optimizations. Design time techniques are applied at high level [11,12,13,14] or at physical level [15,16]. Researchers have only recently started work on temperature aware high-level synthesis [14] and design space exploration [11]. Thermal-aware layout generation for system on chip has been investigated in [15]. In [16], a new thermal-aware placement solution has been proposed for FPGA based on electrostatic charge model for temperature.

The objective of design time solutions is to optimize system performance subject to a peak temperature constraint. These works aim at improving the average power densities or leverage on lateral heat conduction to improve steady-state temperature. Due to their passive nature, these techniques cannot adapt to changes in the operating environment and therefore they do not perform well for transient temperature reduction.

Online DTM techniques apply several mechanisms (including voltage/frequency scaling [4], task migration [5,6] and processor resource adaptation [9,17,18]) to reduce temperature. Leveraging the redundancy in a processor pipeline, several techniques have been proposed for temperature reduction. In [9] the power density is controlled by balancing the utilization of several processor units including register file, issue queues, and functional units. Fetch throttling was also shown to be effective in reducing the temperature [9,18].

Reactive systems in general require the use of thermal sensors and actuators to detect a thermal emergency and activate the appropriate thermal management scheme, which can in turn increase the complexity of the design. A systematic approach for optimally allocating and placing thermal sensors within a microprocessor has been proposed in [19]. The placement algorithm used accurate thermal measurements for runtime temperature optimizations.

Task migration has been widely used as an efficient DTM scheme [20,21]. The work in [21] provides a simplistic thermal model in which the layout of the active cores are not taken into account. In [20], an elaborate model is used to replicate the hot modules in the design and then alternate the execution of the tasks evenly between the replicas. Since the power densities of the tasks might change during the execution of the tasks, evenly alternating the execution between the replicas is not always beneficial. The proposed approach is only limited to relocation of hotspot modules and duplications of such modules increases the area overhead. In this paper, we develop a new placement technique that utilizes all the processing elements on the reconfigurable system and re-maps the tasks to the processing elements according to their corresponding allocated power budgets.

3. RSoC Architecture

An RSoC is a collection of a coarse-grained reconfigurable processers (i.e. DRP [8,7]), and several other IP cores (e.g. DSP) on a single chip (Figure 1). In our example, we assume that the cores and the DRP communicate using a shared memory through dedicated bus communications.

DRP, as a coarse-grained reconfigurable processor, consists of several *Tiles*. DRP architectures are scalable in general and can have arbitrary number of tiles. In DRP-4 prototype, each tile is an 8x8 processing element (PE) grid with dedicated registers and a state transition controller (STC). Each tile contains a repository of configurations or contexts that are used by the STC to reconfigure the interconnections of the processing element to other elements.

While the current prototype can store up to 64 contexts in each PE, additional contexts can be loaded on-demand from DRAM. It is basically controlled by a uC (ARM/MIPS) on-chip. The configuration memory has two ports, one for executing and the other for loading. Therefore users do not need to stop the execution while loading the context with the DRP-4.

4. Thermal-Aware Dynamic Placement for Reconfigurable System on Chip

The flow of our thermal-aware dynamic placement based on power budgeting is depicted in Figure 2. We first divide the whole power trace into several equally-distanced check points. At the

[1] Throughout this paper, the terms maximum power budgeting and critical power are used interchangeably.

beginning of every check point, we decide the optimal placement to be applied on the reconfigurable system for the time interval bounded by the next check point. Note that the final temperatures calculated at one check point will be the initial temperatures of the next check point. The process continues until all the check points in the power trace are covered. Our flow includes three major steps:

1. Calculation of the maximum power budget ($P_{crit}(i)$) permissible for the time interval following the check point (i), given the layout of the reconfigurable system, the initial temperatures at the check point ($T_{init}(i)$), and the maximum permissible temperatures ($T_{crit}(i)$) for the processing elements
2. Optimization of the placement to be reconfigured for the time interval based on $P_{crit}(i)$ and the average powers $P_{ave}(i)$ of the processing elements
3. Calculation of the temperatures at the end of the interval ($T_f(i)$) based on the optimal placement ($PL(i)$), the initial temperatures ($T_{init}(i)$) and the average powers over the interval $P_{ave}(i)$

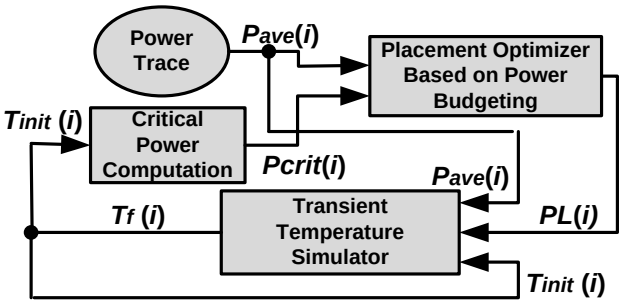


Figure 2: Flow of our thermal-aware dynamic placement based on power budgeting

In this section we first elaborate on the thermal model used in our dynamic placement technique for RSoC platforms. Based on the thermal model, we formally introduce and formulate the concepts of *critical power* and *minimal safe temperature* for each processing element in DRP (Step 1). Then we provide a simulated annealing algorithm to solve the problem of thermal-aware dynamic placement for DRP based on power budgeting (Step 2). In order to calculate the transient temperature for the optimal placement, we adopt the well-known thermal simulator, HotSpot [9] (Step 3).

4.1 Thermal model

It is well known that there is duality between electrical and thermal circuits (in terms of the differential equations governing the current-voltage or power-temperature relations). We have adopted a compact thermal model similar to [9], which models discreet heat flow using thermal resistance-capacitance (RC) network. We highlight the important aspects of this model in a simple example.

In Figure 3 (a), we present a SoC on which there is DRP with four processing elements and an IP-core. Figure 3 (b) presents the equivalent thermal RC network where only the IC die layer is considered. As Figure 3 (b) suggests, every two neighboring processing elements share a lateral resistance, which models the heat diffusion between the elements. The thermal capacitances model the transient behavior of temperature when there are variations in power consumption. The power dissipations of the active nodes are captured by the equivalent current sources. The voltages of the nodes capture the corresponding temperatures in the thermal RC-network. Ambient temperature in the model is captured by using a constant current source connected to the peripheral nodes in the network.

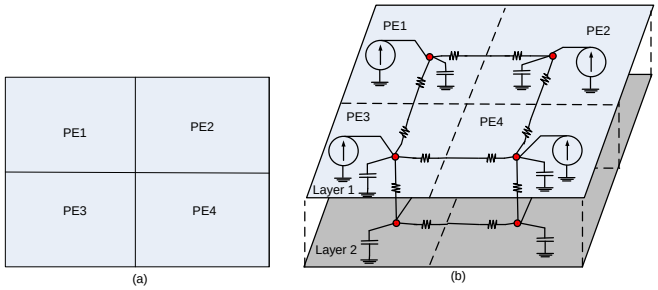


Figure 3: Thermal model for chip die, (a) A sample layout of the processing elements in DRP, (b) the equivalent multilayer thermal RC network

As chip packages may be placed against a heat spreader plate which is turn placed against a heat sink, multiple layers of thermal-RC networks should be used to capture the heat flow between the die and the heat spreader and the heat sink. Details of such extension have been well-explained in [9].

In this model, for the die, spreader and sink layers, the RC model consists of a vertical model and a horizontal model. For each layer, several nodes are assumed where neighboring nodes share lateral resistances to model heat diffusion within the layer. The vertical model captures the heat flow from the die to the heat spreader and from the heat spreader to the heat sink.

Given the thermal RC model of the chip, the temperature-power relationship for a short interval can be captured as stated in Equation (1)

$$P = \frac{C}{\Delta t} \cdot T^{f} - \frac{C}{\Delta t} \cdot T^{in} + G \cdot T^{in} \quad (1)$$

Where P, T, G, C are the power vector (the sum of leakage and dynamic power vectors), the temperature vector, the thermal conductivity matrix and the thermal capacitance matrix respectively. It has been observed that leakage power is a superlinear function of temperature and ignoring this interrelationship may result in underestimation of the chip peak temperature. It has been suggested that that leakage power can be approximated by a linear function in the close proximities of the initial operating ranges of with negligible (<5%) error margins [12,22]:

$$P = P_D + P_L \approx P_D + \alpha \cdot T + \beta \quad (2)$$

In Equation (2), α is a constant diagonal matrix in which the elements of the main diagonal are the thermal sensitivities of corresponding elements in the thermal RC network. β is a constant vector. α and β can be calculated using the leakage-temperature interdependence curves of the processing elements in the design.

As reconfiguration of processing elements for a new placement consumes power, we also need to consider the reconfiguration power vector (P_{rec}) in our model as new placements are planned. The elements of this vector are either zero, if the corresponding processing element is not reconfigured, or equal to the reconfiguration power of that corresponding processing element, in case there is a reconfiguration on the processing element. We also add the reconfiguration power vector for all the processing elements which are reconfigured after replacement. We can now rephrase Equation (1) to reflect the interdependence of temperature and leakage, and the reconfiguration power:

$$P_D = \frac{C}{\Delta t} \cdot T^{f} - \frac{C}{\Delta t} \cdot T^{in} + G \cdot T^{in} - \alpha \cdot T^{in} - \beta - P_{rec} \quad (3)$$

4.2 Critical Power (Minimal Safe Temperature) and its Application on Thermal-Aware Dynamic Placement for RSoC

The critical power is formally defined as: The maximum average power consumption permissible for the processing elements over the interval $[t_{in}, t_f]$, where t_{in} and t_f are the start point and end point of the interval, provided that the initial temperatures of the processing elements are given (T^{in}), such that the temperatures at end of the period, T^f are lesser than the critical temperature T_{crit}.

Conversely, we define the minimal safe temperature (T_S) as: the minimal upper bound of the final temperatures of the processing elements reached at time t_f, provided that the initial temperatures of the processing elements (T^{in}) and the average power consumptions P over the interval $[t_{in}, t_f]$ are given. We can think of the minimal safe temperature as the minimum critical temperature that can be realized for given power and initial temperature vectors.

In the remainder of this section, we first provide the formulations used in our work to derive Critical Power (Minimal Safe Temperature) for RSoCs and then analyze the properties of Critical Power and Minimal Safe Temperature which highlight the advantages of the framework to be used in thermal-aware placement planners for reconfigurable systems.

4.2.1 Derivation of Critical Power (Minimal Safe Temperature) for RSoCs

It should be noted that as stated in Equation (3), the only portion of power independent of temperature is the dynamic power. Once the dynamic power vector is determined, the final temperature vector can be obtained using Equation (3). Therefore, in order to calculate the critical power, we set the final temperatures to the critical temperature and then derive the maximum dynamic power permissible for each processing element. Finally, in order to obtain the critical power, we sum the maximum dynamic power permissible with the leakage power (Equation (2)) and the reconfiguration power to obtain the critical power.

We can simplify Equation (3) by merging the contribution of the initial temperature in Equation (3) to the left side:

$$T^f = G'^{-1} \cdot P' \tag{4}$$

For a small interval, the capacitance matrix can be modeled as a conductivity matrix ($G'=C/\Delta t$). Since G' represent the conductivity matrix of the thermal network, it belongs to the class of positive-definite matrices, which implies that the matrix is invertible [23].

There is a challenge standing in the way of deriving the critical power for the processing elements in the RSoC. The number of thermal RC-network nodes is generally more than the number of processing elements in the RSoC due to chip packaging. Since we are only restricting the temperatures of the processing elements to be lesser than the critical temperature, the maximum temperature constraints for the internal nodes are unknown.

We have developed a smart way to derive the critical power vector P by rephrasing the linear equation stated above into two separate equations, the first of which calculates the final temperatures for all the nodes in the thermal network (T^f) and then the second calculates the critical powers based on T^f. In our formulation, we assume that the number of nodes representing processing elements and the total number of nodes in the RC-network are N and M respectively. The nodes related to the boundary cores are also included in M. The dimensions of the vectors and matrices in the equations are of M. the information (power and temperature) pertaining to the nodes representing the processing elements is positioned in the first N elements:

$$\begin{bmatrix} A_{N\times N} & C_{N\times(M-N)} \\ B_{(M-N)\times N} & D_{(M-N)\times(M-N)} \end{bmatrix} \times \begin{bmatrix} P_0 \\ . \\ P_{N-1} \\ . \\ P_{M-1} \end{bmatrix} = \begin{bmatrix} T_0 \\ . \\ T_{N-1} \\ . \\ T_{M-1} \end{bmatrix} \tag{5}$$

The matrices A, B, C and D are the sub blocks of G'^{-1}. Since G'^{-1} is positive definite, *Sylvester's criterion* holds [24], which states that the upper left sub blocks have positive determinants, hence it implies that the sub block A is nonsingular.

Since the internal nodes are passive nodes, they do not dissipate any power. As the final temperatures of the processing elements should be equal to T_{crit}, we rewrite the equation above as:

$$\begin{bmatrix} P_0 \\ . \\ P_{N-1} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} T_{crit} \\ . \\ T_{crit} \end{bmatrix} - A^{-1} \cdot C \cdot \begin{bmatrix} P_N \\ . \\ P_{M-1} \end{bmatrix} \tag{6}$$

The elements in the second term in Equation (6) are the average powers of the internal nodes and of the nodes pertaining to the boundary cores. The second term can be fused into the left side as:

$$\begin{bmatrix} P_0^{crit} \\ . \\ P_{N-1}^{crit} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} T_{crit} \\ . \\ T_{crit} \end{bmatrix} \tag{7}$$

4.2.2 Properties of Critical Power (Minimal Safe Temperature) for RSoCs

As mentioned previously, we are interested in obtaining the maximum power budget permissible to be dissipated by the processing elements while the temperature constraints are met. Accordingly, we have derived Critical Power. In order to elaborate on the significance of the critical power computation for temperature optimization in RSoC placement, we provide the mathematical foundation to demonstrate the interesting properties of the critical power.

***Definition* I.** An $n\times n$ real symmetric matrix M is *positive semidefinite* iff $X^T \cdot M \cdot X \geq 0$ for all vectors $X \in R^n$.

***Definition* II.** We define the ordering operator ($\leq$) between two vectors X and Y (e.g. temperature, power) as:

$$X \leq Y \Leftrightarrow \forall i : x_i \leq y_i \tag{8}$$

Note that based on the definition, we have the inequality relation defined above satisfies the transitivity property.

***Lemma* I.** All the elements of the main diagonal of a positive semidefinite matrix are nonnegative.

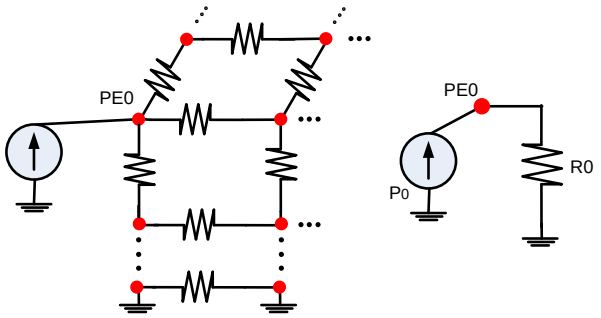
Proof. Plugging the vector Z for which only the i^{th} element is nonzero in the definition of semipositive matrices, we obtain:

$$Z^T \cdot M \cdot Z \geq 0 \rightarrow z_i^2 \cdot M_{i,i} \geq 0 \rightarrow M_{i,i} \geq 0 \tag{9}$$

As we change the index for the nonzero element of Z, we conclude that all the main diagonal elements must have nonnegative values □

***Lemma* II.** In case there is only 1 processing elements dissipating power (i.e. there is only one non-zero element in the power vector P.), the temperatures of all the nodes in the thermal network are nonnegative.

Proof. In order to visualize the statement of Lemma I, without loss of generality, we assume that the first element of the power vector P' is the only nonnegative value, while the rest are all set to zero.



(a) (b)

Figure 4: (a) a sample case in which there is only 1 processing element dissipating power (b) the Thevenin equivalent circuit.

Note that since only PE0 is dissipating power, only one current source representing the processing element is depicted in the Figure. We first prove that the node representing the processing element has a temperature greater than the zero. In order to show the fact, we have depicted the Thevenin equivalent circuit [25] in Figure 4 (b). As mentioned earlier, since there is a one-to-one correspondence between the voltage-current relationship and the power-temperature relationship, therefore, in a similar fashion, we can realize the Thevenin equivalent circuit [25]. Since there is no active components (current sources, voltage sources), the equivalent circuit is modeled as a lumped resistance. Using Equation (4), we conclude that the value of this resistance is equal to $G'^{-1}_{0,0}$. Therefore, based on Lemma I, since $G'^{-1}_{0,0}$ and P_0 are both nonnegative, $T_0 = G'^{-1}_{0,0} \cdot P_0$ is also nonnegative.

So far we have proved that in case there is only 1 processing element dissipating power, there is a node whose temperature is nonnegative. Now we have to extend the proof to cover all the nodes in the thermal network. In order to do so we show that there would be a contradiction if we assumed that there is a node with negative temperature. Assume that such node exists (the voltage of X_0 in Figure 5 is negative). Applying Kirchhoff's current law ([25]), we infer that there has to exist another neighboring node (we call it X_1) whose voltage (temperature) is lesser that the voltage of X_0. Otherwise, if all the neighboring nodes' voltages were nonnegative, then sum of the currents into/from X_0 would be nonzero which contradicts Kirchhoff's current law. Note that the current always flows from a node with higher voltages to the neighboring nodes with lower voltage. Analogously, assuming X_1 has a negative voltage, we reach another node (X_2) that has a lower voltage and consequently a negative voltage. Since there is finite number of nodes in the thermal network and certain nodes have zero voltage (grounded) the sequence of nodes in the descending order of voltages will eventually reach those grounded nodes. The sequence of nodes' voltages in the descending order $\langle 0 > V(X_0) > V(X_1) > \cdots > V(X_{n-1}) = 0 \rangle$ cannot be realized. Therefore, we conclude that all the nodes in the thermal network must have nonnegative voltages (temperatures) □

***Theorem* I.** For short intervals (the condition under which Equation (4) holds), for any two average power consumption vectors X, Y with the relationship $X \le Y$, the corresponding temperatures at the end of the interval will also follow the relationship $T(X) \le T(Y)$.

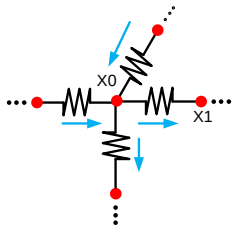


Figure 5: Example of the contradiction used in the proof of Lemma II. The arrows depict the direction of current flow.

Proof. In order to prove Lemma III, we take advantage of the conclusion of Lemma II. We define the sequence of vectors P_i as:

$$X = \begin{bmatrix} x_0 \\ x_1 \\ . \\ x_{n-1} \end{bmatrix}, Y = \begin{bmatrix} y_0 \\ y_1 \\ . \\ y_{n-1} \end{bmatrix}, P_i = \begin{bmatrix} y_0 \\ y_1 \\ . \\ y_{i-1} \\ x_i \\ x_{i+1} \\ . \\ x_{n-1} \end{bmatrix} \quad (10)$$

Obviously, based on the definition above we have $P_0 = P_x$ and $P_n = P_y$. We prove through induction that the sequence of temperature vectors corresponding to the power vectors can be ordered by the inequality operation defined in Definition II as:

$$T(P_x) = T(P_0) \le T(P_1) \le \ldots \le T(P_{n-1}) \le T(P_n) = T(P_y) \quad (9)$$

Statement: $T(P_0) \le T(P_i)$ for $0 \le i \le n$.

When i is 0, the case is true, since $T(P_0) \le T(P_0)$ based on Definition II.

Now, we prove that $T(P_0) \le T(P_{i+1})$ given $T(P_0) \le T(P_i)$. Since based on Definition II we have the transitivity property, it is sufficient to prove that $T(P_i) \le T(P_{i+1})$.

We define ΔP as the difference of P_i and P_{i+1}, which is a vector with only one nonzero element, based on the definition of P_i. The i^{th} element of ΔP is equal to $(y_i - x_i)$, which is nonnegative. Therefore, based on Lemma II, we can deduce the following inequalities:

$$\begin{aligned} & 0 \le G'^{-1} \cdot (P_{i+1} - P_i) \\ & G'^{-1} \cdot P_i \le G'^{-1} \cdot P_{i+1} \\ & T(P_i) \le T(P_{i+1}) \end{aligned} \quad (12)$$

The last inequality above can be extended to the case when i is equal to n -1. Therefore, we conclude that $T(X) \le T(Y)$ given $X \le Y$ for any generic vectors X and Y □

***Corollary* I.** For short intervals (condition in which Equation (5) holds), any average power consumption vector below the critical power vector will yield temperatures below the critical temperature.

***Corollary* II.** Given the average power and the initial temperature vectors, if the critical temperatures T^{crit} is realized, then so is any critical temperature $T^{crit'} > T^{crit}$.

The outcome of the aforementioned theorem can be viewed in the opposite direction to give a sense on how to calculate the minimal safe temperature:

***Theorem* II.** Given the average power and the initial temperature vectors, the minimal safe temperature can be computed as:

$$T_S = \max_{0 \le i < N} (P_i / \sum_j A^{-1}{}_{i,j}) \quad (13)$$

Proof. Based on Theorem I, any average power below the critical power will yield temperatures below the critical temperature. If we set the critical temperature to be equal to the minimal safe temperature, the corresponding critical powers can be used as constraints to figure out whether the processing elements reach the critical temperature or not. In order to obtain the minimal safe temperature, we need to obtain the minimum critical temperature that can be realized by the given power vector. In order to derive the minimal safe temperature, we need to find the minimal temperature that satisfies Equation (7). Rephrasing Equation (13) yields Equation (14), which implies the condition satisfying Equation (5) □

$$\forall 0 \le i < N-1 : P_i \le T_S \cdot \sum_j A^{-1}{}_{i,j} \quad (14)$$

For the special cases where no lateral capacitance between neighboring blocks is assumed in the thermal model [9] (C is a diagonal matrix), it can be shown that the matrices A and A^{-1} are also diagonal. Hence, Equation (13) can be simplified as:

$$T_S = \max_{0 \le i < N} (P_i / A^{-1}{}_{i,i}) \quad (15)$$

The significance of the equations above relies on the fact the computation of the equations takes $O(n)$ time, where n is the number of processing elements in DRP. All the matrices can be computed ahead of time and therefore only simple operations (divisions) expressed in Equation (13) are required. Such feature makes the heuristic formula desirable for optimization engines.

4.3 Thermal-Aware Dynamic Placement Planner (TADPP) for RSoC

The problem of thermal-aware dynamic placement for RSoC can be studied in two similar categories. While one tries to guarantee that the final temperatures of the processing elements are below the critical temperature, the other one tries to minimize the final temperatures. Since both problems are similar in nature, in this section we only state the problem of thermal-aware dynamic placement for RSoC for temperature minimization:

Given the average power vector and the initial temperature vector for the system at time t_0, map the data path resources onto the processing elements of the DRP such that the final temperature vector at time t_f of the system and the interconnection complexity between the processing elements is minimal.

We have developed a simulated annealing search engine which simultaneously optimizes both the wire length and the final temperatures of the placement for each checkpoint. In general, simulated annealing based approaches tend to minimize a cost function during random moves of blocks. Therefore, the run-time of the algorithm is significantly affected by the time complexity of the cost function as it needs to be executed in every move. Traditionally, simulated annealing based thermal-aware placers have used temperature simulations directly to calculate the maximum temperature (e.g. [9]). Such techniques lead to a slow algorithm because of the need to solve finite difference equations governing the temperature-power relationship upon every move in the placement (iteration). In contrast to the traditional techniques, our power budgeting based heuristic efficiently captures temperature inside the simulated annealing engine, therefore drastically reduces the run time of the annealing engine.

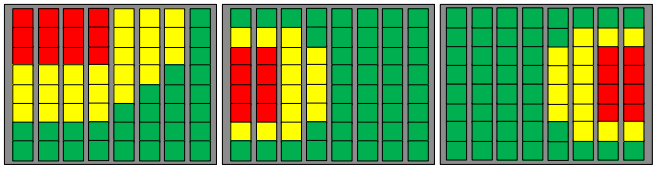
We have adopted the adaptive annealing schedule of VPR, the state-of-art FPGA placement tool. Features of VPR adaptive annealing schedule are explained in details in [26]. We have used the following cost function inside our simulated annealing engine:

$$\Delta cost = \alpha \cdot \frac{\Delta T_S}{T_S^{PREV}} + (1-\alpha)\frac{\Delta WL}{WL^{PREV}} \quad (16)$$

Where α is the coefficient denoting the tradeoff between wire length and temperature. Half perimeter metric is used to estimate the total wire length of the placement. The cost function in Equation (16) uses the minimal safe temperature expressed in Equation (13) rather than the exact maximum temperature obtained by direct finite difference equation solving.

5. Multi-design Thermal-aware dynamic placement planner (MDTAPP) for RSoC

So far in this work, we have introduced a novel thermal-aware dynamic placement planner for reconfigurable systems for transient temperature reduction. In this section, we will introduce the generalization of the problem of thermal-aware dynamic placement planner for RSoC.

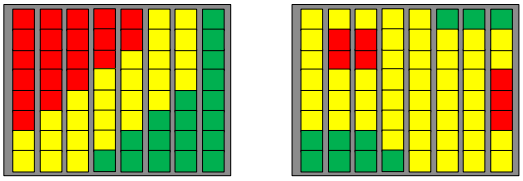


(a) Design 1 (b) Design 2 (c) Design 2'

Figure 6: Steady-state temperature profiles of two designs on RSoC.

When multiple designs are to be implemented on reconfigurable systems, their joint contribution to temperature has to be considered simultaneously. The main reason is that temperature-awareness in placement requires the knowledge of the initial temperatures of the processing elements. Thermal–aware placement tools try to optimize the steady-state temperature by managing the lateral heat conduction of each processing element in the reconfigurable system. However, as the instantaneous temperature profiles are dependent on the initial conditions (temperatures) of the processing elements of the reconfigurable system, placement planning for multiple designs has to be performed jointly, considering the history of the previous designs configured earlier. In order to clarify the concept, we consider to designs which are individually optimized for steady-state temperature and wire lengths. The temperature profiles of the optimized placements for the two designs are depicted in Figure 6. Note that Figure 6 (b) and Figure 6 (c) are in fact the placements of the same design. A 180° rotation of one placement generates the other placement. In our example, we are assuming that Design 2 is going to be reconfigured on the RSoC after once Design 1 finishes execution. In our example we are assuming that the execution time

of each design is comparable to the thermal time constant. Otherwise, if the thermal time constant is much smaller than the execution times of the designs, the thermal profiles of the designs will not influence each other. Using the placements depicted Figure 6, the resultant temperature profiles for the transient temperature are depicted in Figure 7. As shown in Figure 7, the hot spots in a sequence of designs can be mitigated when placement planning for the sequence of designs is performed collectively subject to the transient temperatures of individual designs within the sequence.



(a) **(b)**

Figure 7: Transient temperature profiles (a) when Design 2 is executed after Design 1, (b) when Design 2' is executed after Design 1.

We define the problem of thermal-aware multi-design placement planner for RSoC as: given the layout of the RSoC and a sequence of designs $\{D_1, D_2,\ldots, D_n\}$ to be executed one after another on the RSoC with execution times $\{E_1, E_2,\ldots, E_n\}$ and dynamic power dissipations of $\{P_1, P_2,\ldots, P_n\}$ (P_i is the power vector corresponding to the power dissipation of the individual processing elements in the design), plan the placement of each design on the RSoC so that the maximum temperature of the resources of each design is minimized , in addition to the conventional objectives (i.e. wire length and area).

Multi-Design Thermal Aware Dynamic Placement Planner for RSoC (MDTADPP)

Inputs: RSoC architecture (Layout, leakage power libraries, thermal RC network properties), sequence of designs $\{D_1, D_2, \ldots, D_n\}$, sequence of dynamic power vectors $\{P_1, P_2,\ldots, P_n\}$, sequence of execution times$\{E_1, E_2,\ldots, E_n\}$, initial temperature vector T^{init}, index of the current design: i

Outputs: sequence of placements$\{PL_1, PL_2,\ldots, PL_n\}$, final temperature vector $\{T^f_1, T^f_2, \ldots, T^f_n\}$

[1] if (i = 1) then
[2] $\{PL_1\} \leftarrow$ TADPP(T^{init}, D_1, P_1, E_1)
[3] $\{T^f_1\} \leftarrow$ Transient Temperature Simulator (T^{init}, D_1,P_1, E_1, PL_1)
[4] else
[5] $[\{PL_1, \ldots, PL_{i-1}\}, \{T^f_1, T^f_2, \ldots, T^f_{i-1}\}] \leftarrow$ MDTADPP(T^{init}, $\{D_1, \ldots, D_{i-1}\}$, $\{P_1, \ldots, P_{i-1}\}$, $\{E_1, \ldots, E_{i-1}\}$)
[6] $\{PL_i\} \leftarrow$ TADPP(T^f_i, D_i, P_i, E_i)
[7] $T^f_i \leftarrow$ Transient Temperature Simulator (T^f_i, D_i, P_i, E_i, PL_i)

Figure 8: Multi-design thermal aware dynamic placement planner

In order to solve the problem of thermal-aware multi-design placement planner for RSoC, we use an inductive approach similar to the one depicted in Figure 2. Our proposed solution is based on thermal-aware dynamic placement planner (TADPP) for RSoC (Section 4.3). Figure 8 outlines our proposed solution. As the basis of our solution, we first optimize the placement of the first design D_1 using minimal safe temperature (Equation (16)). Then, we use the transient temperature simulator to obtain final temperatures for the design for the duration E_1. Then, as the inductive step, for each design D_i, given than all the designs $\{D_1, D_2, \ldots, D_{i-1}\}$ are optimized in a similar fashion, we optimize the placement for the design D_i using the minimal safe temperature (Equation (16)). We then use the transient temperature simulator to find the transient temperatures at the end of the execution time E_i.

6. Experimental Results

In this section we first describe our experimental flow and then we will present our results.

6.1 Experimental Setup

Figure 9 depicts our experimental flow. The benchmarks used in our experiments are DSP and multimedia applications, widely used in high-level synthesis community [27]. The operations used in the data flow graphs are adders, multipliers and logic operations. 8-bit ALUs and 8-bit multipliers are used in the data path.

In our experiments we have assumed that only a single tile with 64 processing elements (ALUs) and 8 multipliers is available for our thermal-aware dynamic placement planner. We have performed list scheduling under resource constraints and then Left-Edge-Algorithm (LEA) to generate the RTL description of the data path for each benchmark.

We have applied 10000 randomly generated input vectors to each data path and recorded the activities using ModelSim. We assume that the same pattern will be repeated in the power traces throughout our experiments. The switching activities are then passed to PowerTheater in order to obtain the dynamic power traces of the data path components. PowerTheater simulations were carried out for 130nm technology and 500MHz of operation. We have applied the same technique as reported in [12,28] to modify the power densities of the component as technology size is scaled down from 130 nm to 45 nm.

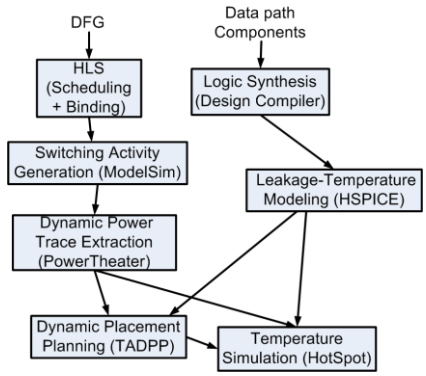


Figure 9: Experimental Flow

In order to model the interdependence of temperature and leakage power, we first logic synthesized each data path component using Design Compiler. For each standard gate we performed leakage calculations for 3 different temperatures 25˚C, 85˚C and 110˚C using HSPICE for 45 nm. The sum of the leakage powers of the individual gates is considered as the leakage of the component. A second degree polynomial is used to approximate the behavior of temperature-dependant leakage power. We have assumed the same chip packaging configuration as modeled by HotSpot [9]. Ambient temperature is set to be 45°C. Different HotSpot parameters are shown in Table 1.

Table 1. Thermal packaging parameters

$C_{Convection}$ = 140.4 J/K	$R_{Convection}$ = 0.1 K/W
Heat Sink Side = 60mm	Heat Sink Thickness = 69mm
Spreader Side = 30mm	Spreader Thickness = 1mm
Chip Thickness = 0.15mm	Sampling Interval = 20 µs

The power traces, chip packaging information, DRP layout and the number of checkpoints are passed to our dynamic placement planner tools to obtain the dynamic placements. Our placement planner tools use Equation Finally, HotSpot-5 simulator is used to obtain the transient temperature of the DRP over time. The

experiments are carried out on a 2.99 GHz Pentium IV machine with 1 GB of RAM running Microsoft Windows XP.

6.2 Experimental Results

6.2.1 Experimental Results on TADPP

In order to evaluate the significance of our developed formulation for thermal-aware dynamic placement, we have compared our heuristic formulation (Equation (6)) with two other formulations that can be used to constrain the peak temperature during dynamic placement planning. The first formulation is to use the finite difference equation solving used in temperature simulators (e.g. HotSpot), which gives the most accurate temperature measurement. The second formulation is adopted from [15], which uses temperature-weighted-distance (TWD) as a tradeoff between execution time and final temperature values:

$$TempDist = \sum_i \sum_j \frac{T_i \times T_j}{d_{i,j}} \qquad (17)$$

In this scheme the temperature update was modified to occur once out of every n annealing moves. During the period between temperature updates the previously simulated temperature obtained by HotSpot is used for calculating the TWD.

We have performed two sets of experiments. The first set of experiments examines the performance of critical power consideration in dynamic placement and the second set explores temperature reduction when minimal safe temperature heuristic is applied in dynamic placement. In all our experiments, we have used power traces of ten million cycles and the initial temperatures of the processing elements are assumed to 85°C.

For the first set of experiments we have performed dynamic placement planner with 100 random moves performed per each check point. We have used 1000 checkpoints to recalculate the placement of the design. The optimization objective in this experiment is to minimize the wire length of the design for all the placements while the maximum temperature of the processing elements does not exceed 100°C. We perform temperature simulations for the entire power trace based on the dynamic placements to get the transient temperatures over the execution of the design. The results in Figure 10 are averaged over the benchmarks.

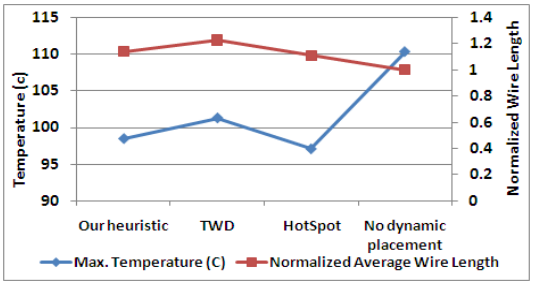


Figure 10: Comparison of placements obtained for 1000 checkpoints using different temperature metrics

As a reference case, we have performed steady-state temperature plus wire length optimization to obtain the fixed placement with the minimum wire length. In the other cases, we have used the metrics explained earlier to find the placement with the minimal wire length for each checkpoint subject to the critical temperature of 100°C. TWD metric was updated every 20 moves. As shown in Figure 10, on average, the maximum temperatures acquired for TWD and fixed placement exceed 100°C. HotSpot and our heuristic metrics maintained the maximum temperatures below 100°C. While TWD reached 21% increase in the wire length on average, the overhead for our heuristic and HotSpot were 15% and 11% respectively. The performance of HotSpot formulation and our heuristic are close in terms of temperature and wire length. However, our technique outperforms HotSpot temperature calculation in execution time. The execution times for our heuristic, TWD, HotSpot and the fixed placement are 8.1, 52.9, 882.7 and 6.2 seconds. The runtime of HotSpot is prohibitively high, which is the main reason to adopt critical power metric inside the search engine to acquire the optimal placement.

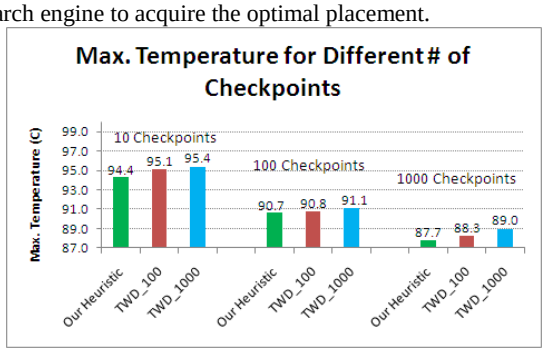


Figure 11: Max. temperature, comparing our heuristic to TWD [15] **for different no. of checkpoints**

In the second set of experiments, we have compared the performance of our thermal-aware dynamic placement planner, when the cost function in Equation (16) is used to minimize the temperature and total wire length, with TWD. We have implemented the adaptive annealing schedule of VPR for both techniques. We have gathered the average results for three cases: 10, 100 and 1000 checkpoints in the same execution traces. The maximum temperatures in the execution traces are presented in Figure 11. In Figure 12, the wire lengths are normalized based on the case where steady-state temperature and wire length minimization are used as the cost function to optimize. The notation TWD_100(1000) represents the number of moves performed before the accurate temperatures are obtained through HotSpot. We have set the number of moves between updates to be 100(1000).

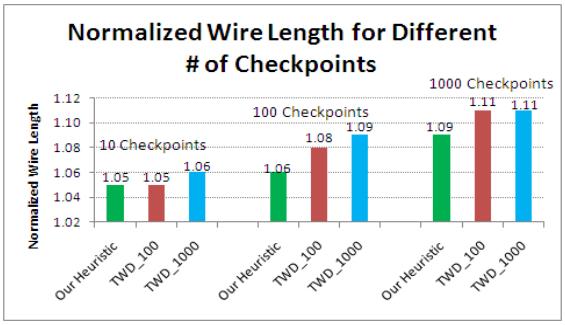


Figure 12: Normalized wire length, comparing our heuristic to TWD [15] **for different no. of checkpoints**

As a general trend, the maximum temperature reached reduces as the number of checkpoints goes higher (Figure 11). Also, for TWD technique, as the frequency of temperature updates goes higher, the average wire length and the maximum temperature improves. This

is due to the fact that the accuracy of temperature calculations in TWD depends on the frequency of invocation of HotSpot. Such inaccuracy impacts both wire length and temperature. Our cost function used in the simulated annealing engine on the other hand combines both parameters. The results suggest that our heuristic outperforms TWD technique in both temperature and wire length.

In Figure 13 we report the average execution times for the different techniques and different number of checkpoints. As shown in the Figure, the execution time of TWD technique deteriorates as the number of check points goes higher. The same trend is seen as the number of moves between temperature updates are reduced (TWD_100). Compared to both versions of TWD, our technique performs dynamic placement planning very fast. The main reason is that our technique does not need to call time-consuming differential equation solving inside the simulated annealing engine. In fact, the formula provided in the previous section requires simple operations that can be performed very fast.

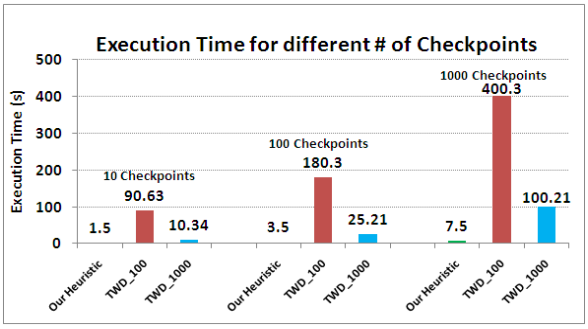


Figure 13: Average execution times, comparing our heuristic to TWD [15] **for different number of checkpoints**

6.2.2 Experimental Results on MDTADPP

In order to highlight the advantages of our formulation for thermal-aware dynamic placement and also the solution used to minimize the operating temperatures as well as the interconnection complexity (wire length) of the implemented designs, we have compared our solution (Figure 8) which uses our heuristic formulation (Equation (6)), with another variant of our solution which use TWD [15]. TADPP simulated annealing scheme in our MDTADPP framework (Line 6 in Figure 8) has been tuned to be same as the one explained in Sec. 6.2.1. Also, as the reference case, we use the weighted sum of the steady-state temperature and the wire length as the cost function in our simulated annealing engine. We obtain the placements for each individual benchmark which are jointly optimized for wire length and steady-state temperature. Note that in this case, we do not take into account the initial conditions imposed by the designs previously placed on the RSoC.

The test cases used in our multi-design thermal-aware dynamic placement planning are expressed in Table 2. The designs used in the sequences in Table 2 are in fact the 20 benchmarks from MediaBench suite [27]. The first test case is sorted based on the average power consumption of the designs. Test case 2(3) uses the 10 benchmarks with the lowest(highest) average overall power consumption. The sequence in test case 4 alternates between low-average power consuming designs and high-average power consuming designs. We have divided the total execution time of each test case (10 million cycles) equally between the comprising benchmarks in the sequence. In other words, for test cases 1 and 4, we have used power traces of 500 thousand cycles for each benchmark and for test cases 2 and 3, the power traces have been collected for 1 million cycles of execution. The initial temperatures of the processing elements are assumed to 85°C.

Table 2: Test cases used in our experiments.

Test Case	**Sequence of benchmarks (from** [27]**),** **Implemented sequentially from left to right**
1	HAL, HORNER, FIR1, ARF, MOTION, EWF, FIR2, FEEDBACK, COSINE1, COSINE2, DOWNSAMPLE, BMP, COLLAPSE, INTERPOLATE, MATMUL, JPEG-FDCT, JPEG-IDCT, IDCTCOL, SMOOTH, INVERT
2	HAL, HORNER, FIR1, ARF, MOTION, EWF, FIR2, FEEDBACK, COSINE1, COSINE2
3	DOWNSAMPLE, BMP, COLLAPSE, INTERPOLATE, MATMUL, JPEG-FDCT, JPEG-IDCT, IDCTCOL, SMOOTH, INVERT
4	HAL, INVERT, HORNER, SMOOTH, FIR1, IDCTCOL, ARF, JPEG-IDCT, MOTION, JPEG-FDCT, EWF, MATMUL, FIR2, INTERPOLATE, FEEDBACK, BMP, COSINE1, COLLAPSE, COSINE2, DOWNSAMPLE

We have compared the performance of our solution for MDTADPP (Figure 8) with the one that uses TWD, in terms of wire length, peak temperature and run-time. In our experiments, TWD metric was updated every 100 moves in the simulated annealing engine. We also report the results obtained for the cases in which only the placements are individually optimized for the steady-state temperature and the wire length.

The maximum temperatures reached for the test cases are shown in Figure 14. As a general trend, our heuristic outperforms both counterparts in the maximum temperature. In fact, merely optimizing the placements for the steady-state temperature and wire length results in poor thermal management. The results indicate that the initial temperatures have to be considered when placements are planned. There is also an interesting correspondence between the properties of the test case sequences and the maximum temperatures obtained in the experiments. The sequence in test case 2(3) has the lowest(highest) maximum temperature among the test cases. The main reason is that the benchmarks used in test case 2(3) are the ones with the lowest(highest) average power consumption. We also observe that the maximum temperature reached in test case 1 is more than the one of test case 4. Since the sequence of test case 4 alternates between the high-power consuming designs and low-power consuming ones, the resultant peak temperature observed is lesser than the other sequence that executes the designs in the ascending order of overall average power consumptions. The fact is also in accordance with the observations made in [29,30] for a single processing element, in which task sequences with alternating power consumptions result in lower peak temperatures.

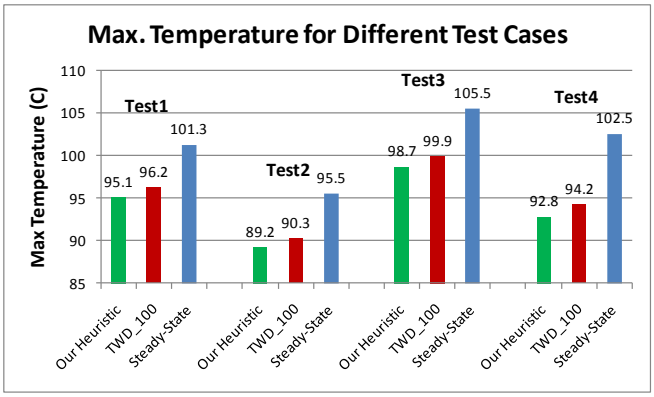


Figure 14: Max. temperature, comparing our heuristic with TWD[11] and steady-state temperature + wire length optimization for different test cases

In Figure 15, we report the average wire length increases in the benchmarks in the sequence, comparing our heuristic with TWD. As the reference case, we find the placements for each benchmark which are optimized for the combination of steady-state temperature and wire length. As shown in Figure 15, our heuristic technique results in lower wire lengths on average, which indicates the benefit of our proposed model for combined temperature and interconnection optimization (Equation (16)).

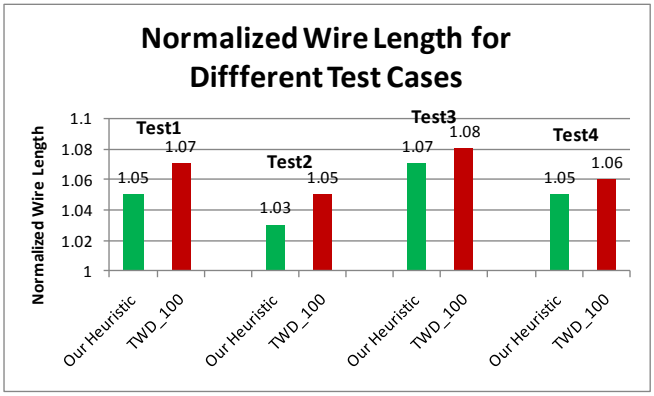


Figure 15: Normalized Wire length, comparing our heuristic with TWD [11] for different test cases

We have compared the execution time of our solution for MDTADPP (Figure 8) with the one that uses TWD and the solution that uses steady-state temperature and wire length as the optimization criteria. As shown in Figure 16, the execution times of the placement planner when TWD is used are two orders of magnitude higher than our solution. For the case of combined steady-state temperature and wire length optimization, the placement of each individual benchmark can be independently determined (regardless of the sequence of benchmarks), since the optimization of such objectives is independent of the transient temperatures at the time the current benchmark is be reconfigured on the RSoC. Therefore, it can be argued that placement planning for each benchmark can be done once. However, such advantage diminishes as the peak temperatures reached indicate, when a sequence of designs is implemented on an RSoC the joint contribution to temperature has to be considered simultaneously.

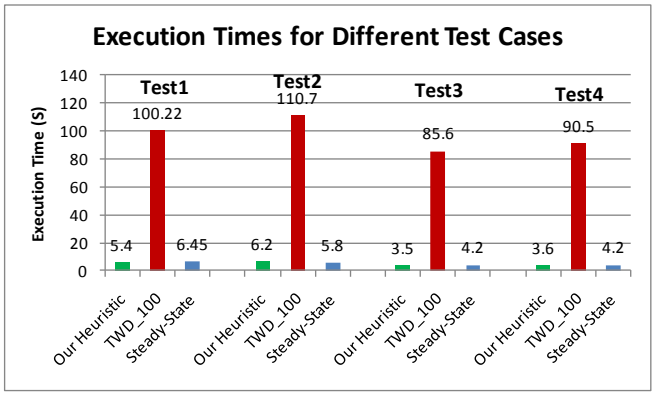


Figure 16: Execution times, comparing our heuristic with TWD [11] and steady-state temperature + wire length optimization for different test cases

7. Conclusion

In this paper, a novel multi-design thermal-aware dynamic placement planner for reconfigurable systems for transient temperature reduction is presented. We have introduced and modeled the concepts of critical power and minimal safe temperature which directly relates the power densities of the elements in the reconfigurable system to hotspots of the chip. Our model includes the interdependency of temperature and leakage power as well as the reconfiguration power to give a more accurate estimation of the critical power and the minimal safe temperature. We have provided in-depth mathematical analysis of the aforementioned concepts in order to provide a more detailed understanding on their applications.

We have designed a fast placement technique to be reconfigured dynamically during design time. We have plugged our placement technique in TADPP and MDTADPP framework.

Our results indicate that our technique performs significantly faster (two orders of magnitude) while the quality of the placements generated in terms of temperature and interconnection overhead is the same, if not better, compared to the thermal-aware placement techniques which perform thermal simulations inside the search engine. We also show that in order to reduce the peak temperature in a multiple designs (multiple versions of the same design or distinctive designs), the joint contribution of the all the designs have to considered simultaneously in the placement planner. The results indicate that we can drastically reduce the peak temperature in the RSoC by using our methodology.

References

[1] J. Srinivasan, et al., "Exploiting structural duplication for lifetime reliability enhancement," in *Int. Symp. Computer Architecture*, 2005, p. 520–531.

[2] R. Viswanath, et al., "Thermal performance challenges from silicon to systems," *Intel Technology Journal*, vol. 4, no. 3, pp. 1-16, Aug. 2000.

[3] S. H Gunther, et al., "Managing the impact of increasing microprocessor power consumption," *Intel Technology journal*, vol. 5, pp. 1-9, Feb. 2001.

[4] R Rao, et al., "Efficient online computation of core speeds to maximize the throughput of thermally constrained multi-core processors," in *ICCAD*, 2008.

[5] F. Mulas, et al., "Thermal Balancing Policy for Streaming Computing on multiprocessor architectures," in *DATE*, 2008.

[6] A. K. Coskum, et al., "Temperature management in multiprocessor SoCs using online learning," in *DAC*, 2008.

[7] NEC electronics, "Dynamically Reconfigurable Processor (DRP) – What is DRP?," 2004.

[8] N. Suzuki, et al, "Implementing and Evaluating Stream Applications on DRP," in *FCCM'04*.

[9] K. Skadron, et al., "Temperature–aware microarchitecture," in *ISCA*, 2003.

[10] S. Golshan, et al., "Exploiting power budgeting in thermal-aware dynamic placement for reconfigurable systems," in *ISLPED*, 2010.

[11] P. Lim, et al., "Thermal-aware high-level synthesis based on network flow method," in *CODES+ISSS*, 2006.

[12] T. Chantem, et al., "Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs," in *DATE*, 2008.

[13] Z. gu, et al., "TAPHS: Thermal-Aware Unified Physical-Level and High-Level Synthesis," in *ASPDAC*, 2006.

[14] R. Mukherjee, et al., "Temperature-aware resource allocation and binding in high-level synthesis," in *DAC*, 2005.

[15] M. Healy, et al., "Thermal Optimization in Multi-Granularity Multi-Core Floorplanning," in *ASPDAC*, 2009.

[16] J. Jaffari, et al., "Thermal-Aware Placement for FPGAs using Electrostatic Charge Model," in *ISQED*, 2007.

[17] M. Powell, et al., "Balancing resource utilization to mitigate power density in processor pipelines," in *International Symposium on Microarchitecture*, 2005.

[18] K. Skadron, et al., "Hybrid architectural dynamic thermal management," in *DATE*, 2004.

[19] R. Mukhrejee, et al., "Systematic temperature sensor allocation and placement for microprocessors," in *DAC*, 2006.

[20] H. D. Mogal, et al., "Thermal-Aware Floorplanning for Task Migration Enabled Active," in *ICCAD*, 2008.

[21] S. Heo, et al., "Reducing Power Density through Activity Migration," in *ISLPED*, 2003.

[22] Y. Liu, et al., "Accurate Temperature-Dependent Integrated Circuit Leakage Power Estimation is Easy," in *Proc. Design, Automation & Test in Europe Conf.*, 2007, p. 204–209.

[23] S. C. Chapra, *Numerical Methods for Engineers, 4th Ed.* McGraw Hill, 2002.

[24] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000.

[25] C. M. Alexander, et al., *Fundamentals of electrical circuits*, 4th ed. McGraw-Hill, 2008.

[26] Vaughn Betz, et al., *Architecture and CAD for Deep-Submicron FPGAs*. KLUWER Academic Publishers, 1999.

[27] Express group at UCSB:, "http://express.ece.ucsb.edu/benchmark/".

[28] G. Link, et al., "Thermal trends in emerging technologies," in *ISQED*, 2006.

[29] S. Zhang, et al.,, "Thermal aware task sequencing on embedded processors∗," in *Proceedings of DAC*, 2010.

[30] R. Jayaseelan, et al., "Temperature aware Task Sequencing and Voltage Scaling," in *Proceedings of ICCAD*, 2008.