# Exploiting Power Budgeting in Thermal-Aware Dynamic Placement for Reconfigurable Systems

Shahin Golshan[1], Eli Bozorgzadeh[1], Benjamin C. Schafer[2], Kazutoshi Wakabayashi[2], Houman Homayoun[1] and Alex Veidenbaum[1]

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

[2] NEC Corp., Central Research Laboratories
1753 Shimonumabe, Kanagawa, 211-8666 Japan
{schaderb, wakaba}@bl.jp.nec.com

## Abstract

In this paper, a novel 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 the design statically, while considering the boundary conditions. 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 of the system. 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 [1] and therefore such solutions might be prohibitively expensive in practice.

In order to cope with the increasing demand for higher computation capacities in reconfigurable systems, several dynamic thermal management (DTM) schemes have been proposed [2,3,4]. 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.

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


*ISLPED'10*, August 18–20, 2010, Austin, Texas, USA.


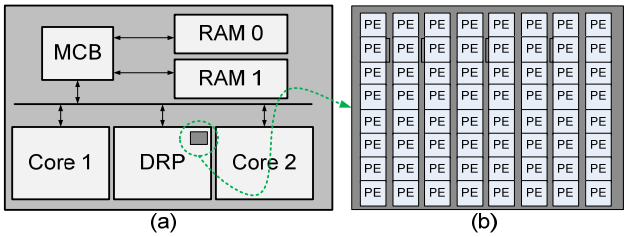


**Figure 1: (a) Sample RSoC (b) Single tile of the DRP**

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 multiple versions of the design and reconfigure each placement on the dynamic reconfigurable system. An interesting feature of coarse-grained reconfigurable systems is the low-overhead of dynamic reconfiguration as opposed to fine-grained reconfigurable systems (e.g. FPGA). Coarse-grained reconfigurable systems can be abstracted as arrays of processing elements (PE). A prominent example of such reconfigurable systems is Dynamically Reconfigurable Processor (DRP) [5,6]. The overhead of reconfiguration of the DRP is a single clock cycle. We exploit this feature to reduce the peak transient temperature through reconfiguration. At design time, new placements for the same 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 [7]. However, solving the complex finite difference equations is very time-consuming. Here we motivate an alternative methodology which infers 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. The main contributions of this paper are:

1. Development of a fast and accurate heuristic model based on power budgeting to plan dynamic placements of the design statically, considering the boundary conditions
2. Designing a fast placement technique based on the heuristic to plan the dynamic placements in design time.

Our experimental results indicate that our proposed technique 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 optimizations. Design time techniques are applied at high level [8,9,10] or at physical level [11,12]. 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 (voltage/frequency scaling [2] or task migration [3,4]) to reduce temperature. 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.

Task migration has been widely used as an efficient DTM scheme [13,14]. The work in [14] provides a simplistic thermal model in which the layout of the active cores are not taken into account. In [13], 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.

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

## 3. RSoC Architecture

An RSoC is a collection of a coarse-grained reconfigurable processers (i.e. DRP [6,5]), 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 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 $P_{ave}(i)$

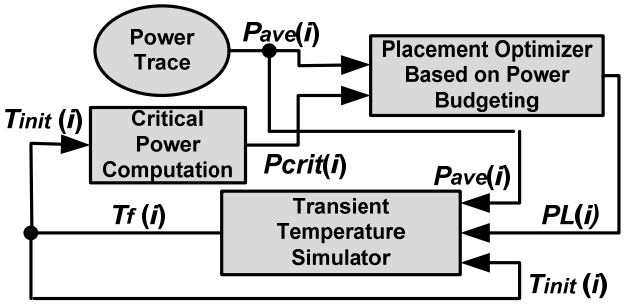


**Figure 2: Flow of our thermal-aware dynamic placement based on power budgeting**

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 [7] (Step 3).

## 4.1 Critical Power and its Application on Thermal-Aware Dynamic Placement for RSoC

It is well known that there is duality between electrical and thermal circuits. We have adopted a compact thermal model similar to [7], which models discrete heat flow using thermal resistance-capacitance (RC) network. In this model, for the die, spreader and sink layers, several nodes are assumed, where the neighboring nodes share lateral resistances to model heat diffusion within the layer. The vertical thermal resistance captures the heat across the different layers. 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 temperature vector, the thermal conductivity matrix and the thermal capacitance matrix respectively.

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.

We can simplify Equation (1) by merging the contribution of the initial temperature in Equation (1) to the left side:

$$T^{f} = G'^{-1} \cdot P' \quad (2)$$

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 [15].

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:

$$\begin{bmatrix} A_{N\times N} & C_{N\times(M-N)} \\ B_{(M-N)\times N} & D_{(M-N)\times(M-N)} \end{bmatrix} \times \begin{bmatrix} P_0 \\ . \\ P_{N-1} \\ . \\ P_{M-1} \end{bmatrix} = \begin{bmatrix} T_0 \\ . \\ T_{N-1} \\ . \\ T_{M-1} \end{bmatrix} \quad (3)$$

The matrices $A$, $B$, $C$ and $D$ are the sub blocks of $G'^{-1}$. Since $G'^{-1}$ is positive definite, *Sylvester's criterion* holds [16], 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:

$$\begin{bmatrix} P_0 \\ . \\ P_{N-1} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} T_{crit} \\ . \\ T_{crit} \end{bmatrix} - A^{-1} \cdot C \cdot \begin{bmatrix} P_N \\ . \\ P_{M-1} \end{bmatrix} \quad (4)$$

The elements in the second term in Equation (4) 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:

$$\begin{bmatrix} P_0^{crit} \\ . \\ P_{N-1}^{crit} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} T_{crit} \\ . \\ T_{crit} \end{bmatrix} \quad (5)$$

The following Theorem expresses the significance of critical power computation for temperature optimization in RSoC placement.

***Theorem* I.** For short intervals (the condition under which Equation (2) holds), any average power consumption vector below the critical power vector will yield temperatures below the critical temperature.

***Proof***. For the sake of simplicity, we only consider the case where there is only one layer in the chip packaging. In this case, for each node in the thermal network, there is a vertical resistance connecting the node to the node with constant temperature (ambient temperature). Figure 3 depicts a sample thermal network. Since the critical power of each node is calculated for the case that all the nodes reach the critical temperature, we can state the critical power in terms of the critical temperature as ($T^{amb}$ is the ambient temperature):

$$P_i^{crit} = \frac{(T_{crit} - T_{amb})}{R_i} \quad (6)$$

Now we prove that for any power vector $P$ less than $P^{crit}$, if there exists a node with temperature higher than $T_{crit}$, then we reach a contradiction. Without loss of generality, we assume that the temperature at node 0 has exceeded the critical temperature. Applying Kirchhoff's current law we reach:

$$\sum_j \frac{T_0 - T_j}{R_{0,j}} + \frac{T_0 - T^{amb}}{R_i} = P_i \le P_i^{crit} \quad (7)$$

Note that we have summed up the heat diffusions of the nodes which share a lateral resistance. If $T_0$ becomes greater than $T^{crit}$ then there must be at least one other node, name it node 1 (The indices used here follow the order the high temperature nodes are visited), whose temperature $T_1$ is greater than $T_0$ or otherwise Equation (7) cannot be satisfied. Now we can repeat the process for node 1. It turns out that there must be another node, like node 2 with temperature higher than node 1 as so on. Eventually the last node is visited and since all the previous nodes have lower

temperatures than the last node ($T_{crit} < T_0 < \ldots < T_{N-1}$), therefore, in the thermal network, given that the power consumptions of the nodes are less than corresponding critical powers, the temperatures of all the nodes remain below the critical temperature □

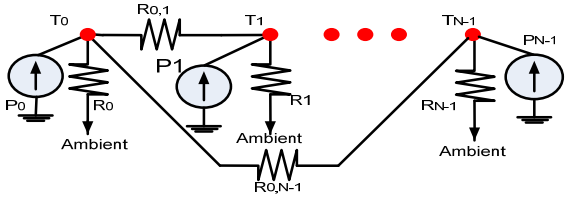


**Figure 3: Thermal network of single layer (Chip Die)**

When the temperature reaches the steady-state, the transient term of the right side of Equation (1) is zero (i.e. $P = G.T$). Hence, the steady-state relationship between the critical power and the critical temperature is analogous to the formulation provided in Equation (2). Hence, the same steps can be taken to derive the critical power for each processing element in the steady-state. The critical power for the steady-state is the maximum permissible average power of each block over the entire execution time of interest.

***Corollary.*** 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:

$$T_S = \max_{0 \le i < N} (P_i / \sum_j A^{-1}{}_{i,j}) \quad (8)$$

***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 (5). Rephrasing Equation (8) yields Equation (9), which implies the condition satisfying Equation (5) □

$$\forall\, 0 \le i < N - 1 : P_i \le T_S \cdot \sum_j A^{-1}{}_{i,j} \quad (9)$$

For the special cases where no lateral capacitance between neighboring blocks is assumed in the thermal model [7] ($C$ is a diagonal matrix), it can be shown that the matrices $A$ and $A^{-1}$ are also diagonal. Hence, Equation (8) can be simplified as:

$$T_S = \max_{0 \le i < N} (P_i / A^{-1}{}_{i,i}) \quad (10)$$

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 (8) are required. Such feature makes the heuristic formula desirable for optimization engines.

## 4.2 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. [7]). 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 [17]. We have used the following cost function inside our simulated annealing engine:

$$\Delta cost = \alpha \cdot \frac{\Delta T_S}{T_S^{PREV}} + (1 - \alpha) \frac{\Delta WL}{WL^{PREV}} \quad (11)$$

Where $\alpha$ 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 (11) uses the minimal safe temperature expressed in Equation (8) rather than the exact maximum temperature obtained by direct finite difference equation solving.

# 5. Experimental Results

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

## 5.1 Experimental Setup

Figure 4 depicts our experimental flow. The benchmarks used in our experiments are DSP and multimedia applications, widely used in high-level synthesis community [18]. 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 [9,19] to modify the power densities of the component as technology size is scaled down from 130 nm to 45 nm.

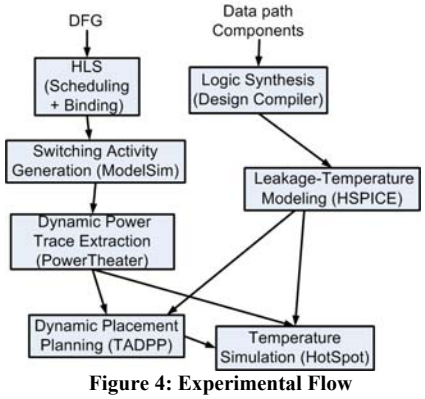


**Figure 4: Experimental Flow**

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 [7]. 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 fed to TADPP tool to obtain the dynamic placements for the check points. 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.

## 5.2 Experimental Results

In order to evaluate the significance of our developed formulation for thermal-aware dynamic placement, we have compared our heuristic formulation (Equation (4)) 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 [11], which uses temperature-weighted-distance (TWD) as a tradeoff between execution time and final temperature values:

$$TempDist = \sum_i \sum_j \frac{T_i \times T_j}{d_{i,j}} \qquad (12)$$

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 5 are averaged over the benchmarks.

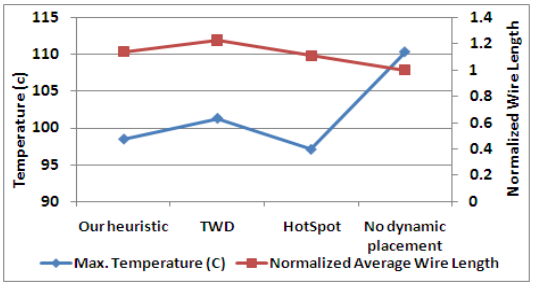


**Figure 5: Comparison of placements obtained for 1000 checkpoints using different temperature metrics**

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 5, 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.

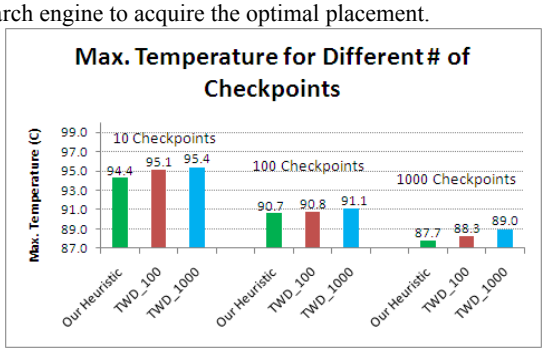


**Figure 6: Max. temperature, comparing our heuristic to TWD** [11] **for different no. of checkpoints**

In the second set of experiments, we have compared the performance of our thermal-aware dynamic placement planner,

when the cost function in Equation (11) 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 6. In Figure 7, 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).

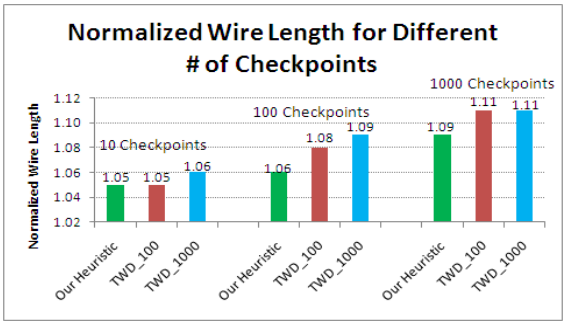


**Figure 7: Normalized wire length, comparing our heuristic to TWD [11] for different no. of checkpoints**

As a general trend, the maximum temperature reached reduces as the number of checkpoints goes higher (Figure 6). 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.

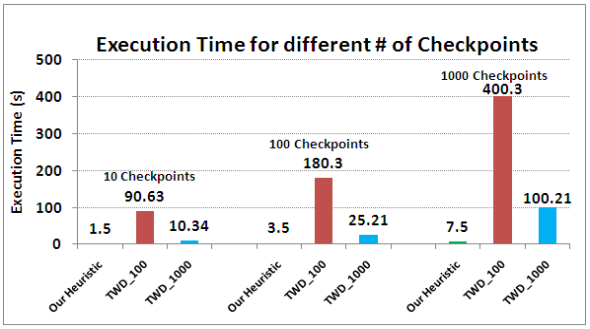


**Figure 8: Average execution times, comparing our heuristic to TWD [11] for different number of checkpoints**

In Figure 8 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.

## 6. Conclusion

In this paper, a novel 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. We have designed a fast placement technique to be reconfigured dynamically during design time. 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.

## References


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

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

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

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

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

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

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

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

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

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

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

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

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

[14] S. Heo, et al., "Reducing power density through activity migration," in *ISLPED*, 2003.

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

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

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

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

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