Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 784945, 15 pages
doi:10.1155/2012/784945

Research Article

A Systematic Methodology for Reliability Improvements on SoC-Based Software Defined Radio Systems

Dionysios Diamantopoulos, Kostas Siozios, Sotiris Xydis, and Dimitrios Soudris

Department of Computer Science, School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece

Correspondence should be addressed to Dionysios Diamantopoulos, diamantd@microlab.ntua.gr

Received 15 November 2011; Revised 29 February 2012; Accepted 3 April 2012

Academic Editor: Dionysios Reisis

Shrinking silicon technologies, increasing logic densities and clock frequencies, lead to a rapid elevation in power density. Increased power density results in higher onchip temperature, which creates numerous problems tightly firmed to reliability degradation. Since typical low-power design has been proved inefficient to tackle the temperature increment by itself, device architects are facing the challenge of developing new methodologies to guarantee timing, power, and thermal integrity of the chip. In this paper, we propose a thermal-aware exploration framework targeting temperature hotspots elimination through the efficient exploration of multiple microarchitecture selections over the temperature-area trade-off curve. By carefully planning at design time the resources of the initial microarchitecture that should be replicated, the proposed methodology optimizes the system's thermal profile and attens on-chip temperature under various design constraints. The introduced framework does not impose any architectural or compiler modification, whereas it is orthogonal to any other thermal-aware methodology. For evaluation purposes, we employ the software-defined radio executed onto a thermal-aware instance of LEON3 processor. Based on experimental results, we found that our methodology leads to an architecture that exhibits temperature reduction of 17 Kelvin degrees, which leads to improvement against aging phenomena about 14%, with a controllable overhead in silicon area about 15%, compared to the initial LEON3 instance.

1. Introduction

Communication has become one of the central uses of computing technology over the years. Architectures that facilitate communication, such as mobile phones and wireless networks, have been primary factors in driving the evolution of microprocessors and computer systems. With the evolution of wireless mobile communications, the problem emphasis has shifted to networking protocols and signal processing that are required to sustain the necessary bandwidth of these applications. In recent years, we have seen the emergence of an increasing number of wireless protocols (e.g., 2G, 3G, GPRS, WiFi, etc.) that are applicable to different types of networks.

Software-defined radio (SDR) technology was created to improve interoperability between different wireless networks, field radios, and devices [1]. SDR technology is comprised of both software and hardware that can be dynamically reconfigured to enable communication between a wide variety of changing communications standards, protocols, and radio links. With this latest SDR technology, system architects are able to create multimode, multiband, and multifunctional wireless devices and network equipment that can be dynamically reconfigured, enhanced, and upgraded through software updates and hardware reconfiguration.

Since the majority of these systems exhibit high-throughput, low-power requirements, and short time-to-market, previous studies proposed the usage of System-on-Chip (SoC) architectures to support the efficient implementation of SDR [2, 3]. Meeting the thermal constraints and reducing the temperature hotspots at these platforms are critical tasks in order to design reliable systems. Furthermore, since chip's temperature has significant impact on performance, reliability, power consumption, as well as cooling and packaging costs, it should be carefully optimized at design

time. Thermal-aware design is difficult, whereas designing a chip and package for the worst-case power-consumption scenario may be prohibitively expensive.

For this purpose, thermal management has recently received a lot of attention by design architects. The goal of thermal management is to meet maximum operating temperature constraints, while tracking timing specifications. Moreover, thermal management can also achieve further temperature reduction in order to improve the reliability degradation of SoCs.

Previous studies have shown that thermal stress is tightly firmed to reliability issues [4]. For instance, thermal cycling can be modeled with the Coffin-Manson relation, which relates in an exponential way the number of cycles to failure to the magnitude of thermal cycling [5]. Existing approaches aim to perform thermal management with techniques that come from the power reduction domain.

Typical instantiation of this solution is the usage of dynamic voltage and frequency scaling (DVFS) [6]. Due to the approximately quadratic relation between supply voltage and power consumption, DVFS-based techniques achieve to provide mentionable savings in power consumption, but they impose slower operation frequency. Moreover, these techniques cannot guarantee that temperature hotspots and/or temperature gradients will be reduced, since there are applied during runtime as a reaction to chip's thermal crisis.

Another technique for providing temperature hotspot elimination, especially at multicore architectures, is based on load balancing [7]. Even though these techniques have been studied for general purpose parallel computers, they targeted mainly the avoidance of performance bottlenecks rather than thermal issues.

A similar approach is discussed in [8] where tasks are swapped between hot and cool cores in order to control temperature values across the target architecture. However, this approach assumes that threads are transferred among different cores, which cannot provide focused thermal management (this would be feasible only if functionality transfer is also supported instead of transferring threads). Furthermore, there are available previous works [9, 10] that perform thermal management through compiler optimizations. The main limitation of this approach affects the difficulty to estimate with sufficient accuracy the temperature variations occurred due to revisions of source code. Furthermore, this technique is applicable only to a small percentage of SoCs, since the required architectural details are not always freely available. In order to alleviate these limitations, instruction-level accounting techniques that are based on empirical measures have been employed.

A common drawback among techniques discussed up to now is that they do not incorporate any mechanism for handling thermal history of the cores. This feature provides useful guidelines about the future behavior of the system and can be exploited to improve the results of the migration. In addition to that, existing approaches mainly provide thermal aware application mapping onto SoC devices based on exploration provided through simulation results. These approaches assume that target platform is fixed ignoring about potential improvements achieved through architecture-level optimizations [11]. In this paper, we propose a new methodology, as well as the software supporting framework, for performing architectural and physical design under constraints posed by temperature hotspots. In particular, the motivated idea introduced in this paper exploits the selective replication of hardware blocks that exhibit increased power densities. Then, by appropriately assigning tasks onto these replica blocks, it is feasible to alleviate the chip's thermal stress.

The proposed approach aims at temperature optimization, while it can be considered as a proactive strategy that alleviates thermal stress at run-time. The introduced framework does not impose any architectural or compiler modification, whereas it is orthogonal to any other thermal-aware methodology discussed above, since it is based on new architectural schemes to eliminate the consequences posed by temperature hotspots. Thus, existing work on thermal aware application mapping and dynamic thermal management can be used in a modular manner to extend the proposed methodology.

Specifically, we target at the development of an automated design space exploration framework that extracts and evaluates a large number of architectural solutions. Every solution exploits selective block replication. Based on the software-supported automatic exploration, we are able to compute higher thermal quality Pareto curves, in contrast to many similar existing optimization approaches that retrieve only a single architecture [8, 9]. Hence, architects can trade-off between the desired level of temperature reduction at hotspots and the resulting timing/area/power overheads. Furthermore, the supporting tool framework provides a considerable speedup at the exploration procedure.

Previous works introduced the usage of parallelism in order to achieve power savings, which in turn lead to temperature reduction [12]. A parallel implementation of a design essentially replicates component(s) of the design such that parallel branches process interleaved input samples. Therefore, the inputs coming into each parallel branch can be effectively downsampled. An output multiplexer is needed to recombine the outputs, and produce a single data stream.

The main differentiation of the proposed work, as compared to this approach, is that our solution does not assume that replica blocks of the same type are working in parallel. More specifically, in our methodology, only one of the available replica blocks is active at any time. The selection of this active block is based on its thermal condition, as it is described in upcoming sections. The contributions of this paper can be summarized as follows.

(i) We show the optimization potential regarding thermal aware exploration by exploiting selective replication of specific architectural blocks.

(ii) We introduce of a novel methodology targeting to provide: (i) elimination of thermal hotspots at SoCs targeting SDR architectures and (ii) alleviation to the temperature gradients.

(iii) Rather than providing only one architectural solution, our methodology retrieves a number of Pareto

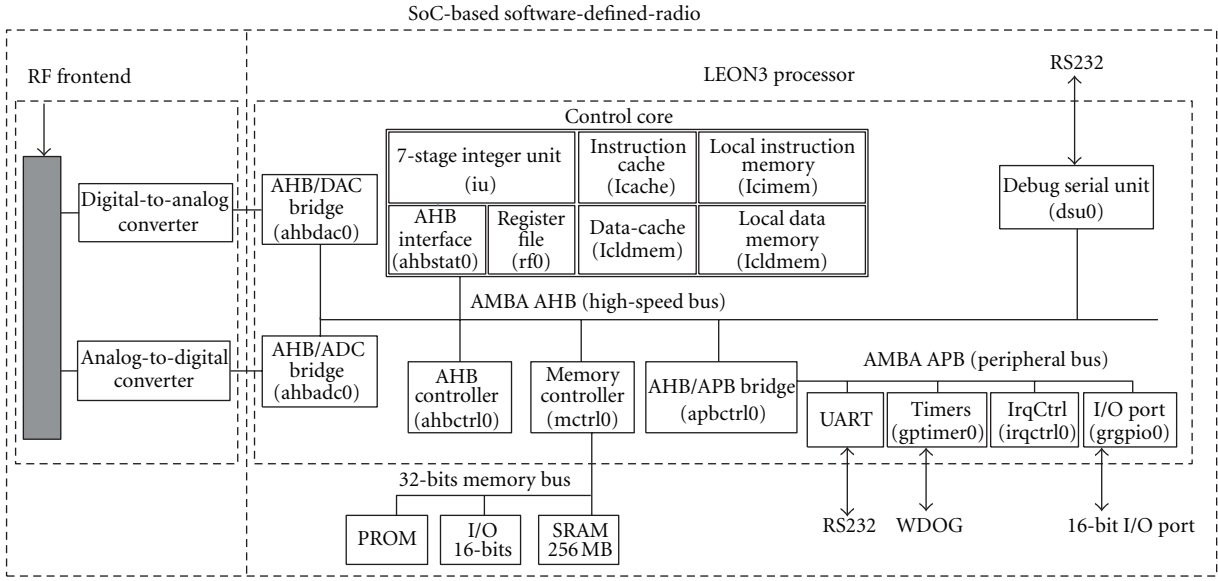

Figure 1: The block diagram of employed SoC-based SDR.

architectural solutions, each of which trades-off different design constraints/criteria.

(iv) We propose a novel design methodology that is orthogonal to the existing approaches found in relevant literature [6–10, 13–16].

(v) We provide CAD support through developing a software supported thermal aware exploration framework, which is public available for additional extensions through [17].

(vi) We apply the proposed methodology to a real case SoC design consisting of a synthesized LEON3 processor [18].

Experimental results prove the efficiency of the proposed methodology, showing that the selected architecture leads to temperature reduction about 8% (from 380 Kelvin to 363 Kelvin), with a controllable silicon area increase of 15%. As we show latter, such a temperature reduction apart from reduction in cooling cost also achieves mentionable improvement to the consequences posed by aging phenomena about 14%.

The rest of the paper is organized as follows: Section Section 2 introduces the underline SDR architecture, whereas Section 3 discusses motivational observations that guide us to propose selective insertion of replica blocks. Section 4 describes in a brief manner the microarchitectural enhancements needed for applying selective block replication in existing microprocessor architectures. The proposed methodology is analyzed in detail in Section 5, while a number of evaluation results are discussed in Section 6. Finally, Section 7 concludes the paper.

2. Target SDR Architecture

During the last years, a number of different SDR-based architectures have been developed, whereas a typical instantiation is depicted in Figure 1. The front-end is responsible for converting the signal between the RF domain and an intermediate frequency, and the A/D and D/A components convert the signal between the analogue and the digital domain. In our analysis, the baseband functionality is carried out on software running on a system-on-chip based on LEON3-embedded processor [18].

LEON3 processor consists of the integer unit, the cache subsystem, the memory management system, and the AMBA interface. The instruction unit is fully compatible with the SPARC V8 instruction set, whereas the pipeline consists of 7 stages. The integer unit has configurable separate instruction and data cache (Harvard architecture), whereas the size for each of them is equals to 1 Kbyte. Furthermore, the integer unit includes a configurable register file with register window equals to 8. Regarding the L1 caches, they are managed by a cache controller which is interfaced to the system's AMBA AHB bus. The communication to LEON3 peripherals is performed with two bus controllers, referred as AHB (advanced high-performance bus) and APB (Advanced Peripheral Bus) controller, respectively. The first of these controllers (AHB controller) is used for the connection of high speed components (i.e., integer unit, memory controller, etc.), whereas the second one (APB controller) provides control to the low-speed peripherals (i.e., UARTs, I/Os, etc.). Finally, LEON3 processor contains a configurable separate local data (2 KByte) and instruction memory (2 KByte).

3. Motivation

In this section, we discuss the importance of different hardware blocks to be considered as critical for thermal stress.

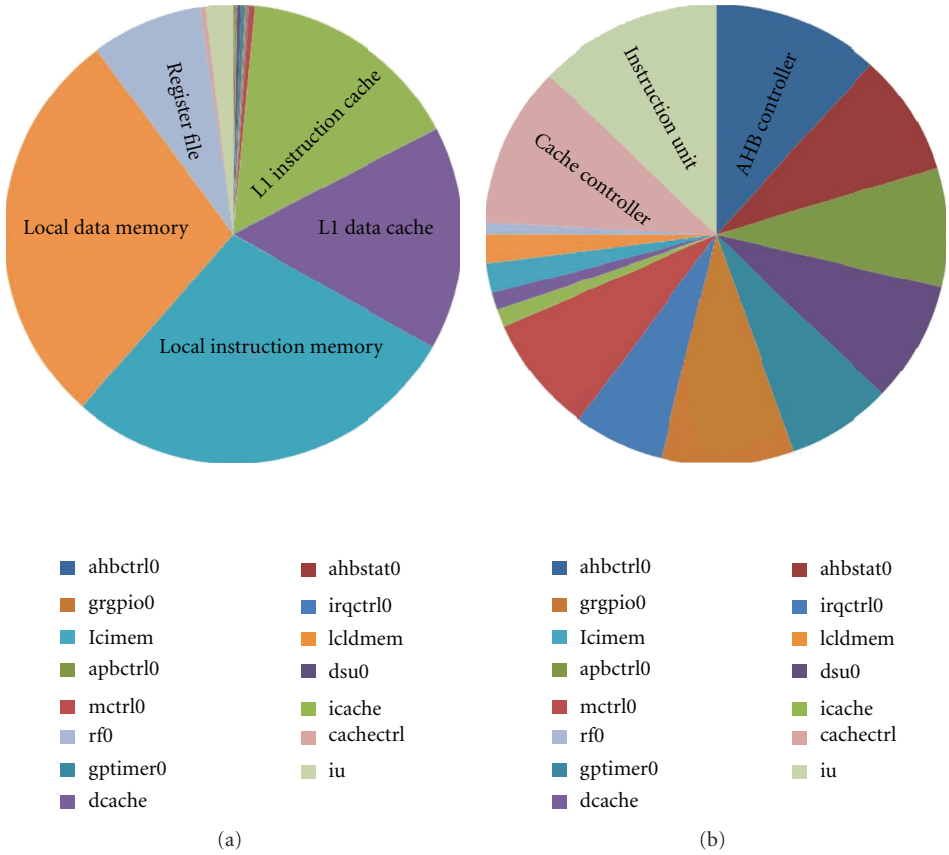

Figure 2: (a) Power consumption and (b) power density pies for LEON3 architecture.

This problem becomes even more important regarding either high-end processor architectures, that is, superscalar organizations, or multicore SoC designs, where multiple hardware components, each of which with different area and power values, are combined into a single device. Hence, one of the challenges that architects are facing today is to identify the hardware components that higher affect thermal stress.

In order to show how different hardware block thermal profiles affect the thermal stress of the entire IC, Figure 2(a) gives the power consumption for the components of the LEON3 processor, when the SDR system is executed.

We select such an embedded processor because it is widely used in numerous commercial and/or research products. However, apart from the selected target platform, the methodology we follow in this paper is also applicable to any other digital architecture.

Since embedded cores usually are designed with low power criterion, many researchers up to now pay effort to reduce maximal temperature values by identifying blocks that dissipate increased power budges. Regarding the LEON3 processor, the local data/instruction memories, the L1 data/instruction caches, as well as the register file are found to be the most power hungry blocks. More specifically, the average power consumption at these blocks, as compared to the total power dissipation, is 57%, 31%, and 8%, respectively.

Even though Figure 2(a) provides a first order metric about the components with increased power consumption, we show that it is not enough in order to retrieve conclusions about their importance regarding the thermal stress. This occurs since the power metric does not take into consideration the area of underline hardware block, which is especially crucial for thermal spreading. Hence, a more representative metric should be employed in order to evaluate the importance of each core into the chip's temperature values.

A candidate metric for this scope is power density, which denotes the ratio of power consumption for each hardware block per the area occupied by this block. Figure 2(b) gives the corresponding power density pie chart regarding the LEON3 architecture. As we can conclude from this figure, the components with increased power densities are not those identified as critical based solely on the power consumption criterion. More specifically, the power density denotes that AHB controller, instruction unit, and cache controller are the blocks with increased impact on thermal stress. These blocks contribute to the total power density about 12%, 13% and 11%, respectively, whereas the five blocks already identified based on power consumption correspond to 5% of total power density. This occurs mainly since the blocks with increased power consumption have also considerable increased area

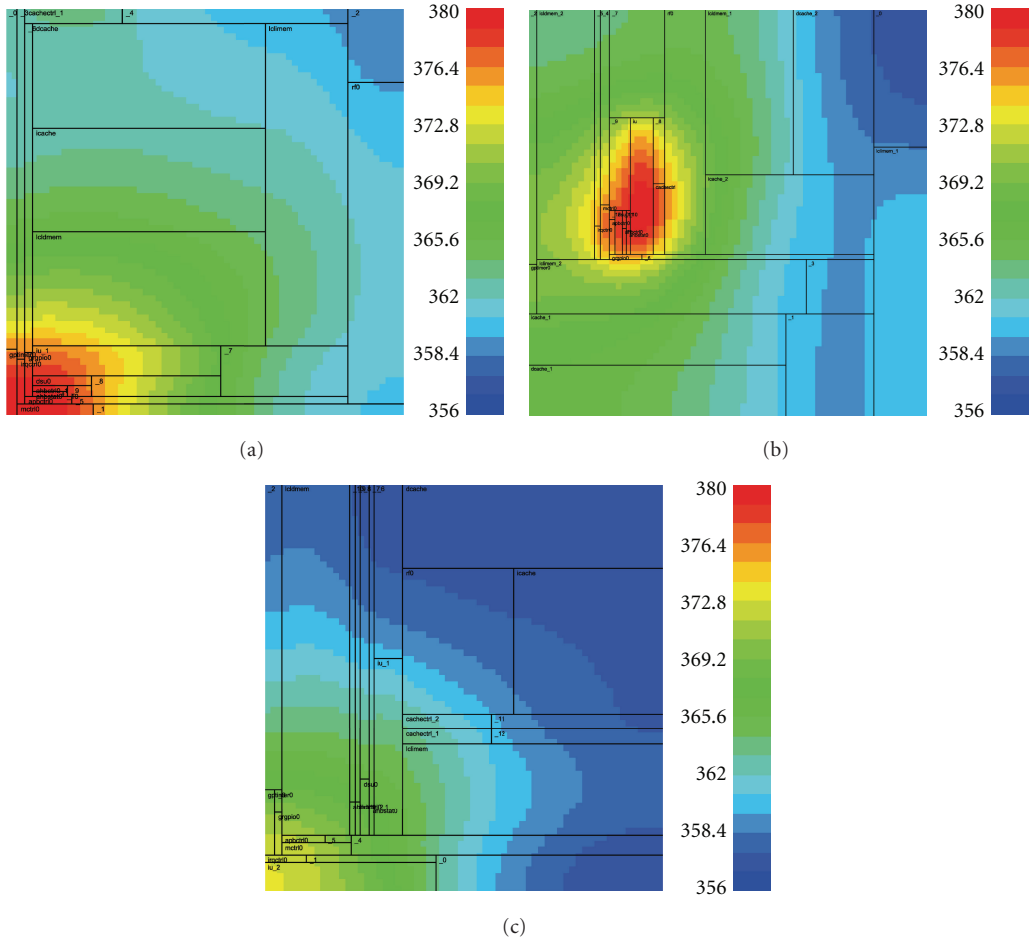

Figure 3: Thermal profile for LEON3: (a) without considering replica blocks, (b) with replica blocks (2 × local data/instruction memories, 2 × L1 data/instruction caches, 2 × register file), and (c) with replica blocks (2 × instruction unit, 2 × cache controller, 2 × AHB controller).

(about 91% of the total architecture's area), which in turn leads to almost negligible power density values for these blocks.

Next, we depict that the criterion of power density is much more important than the corresponding one about power consumption. For this purpose, Figures 3(b) and 3(c) give the thermal profiling as they derived with Hotspot tool [19] about a LEON3 processor running SDR applications (e.g., filters, encoding/decoding, etc.), when the five and three most critical components retrieved with the previously mentioned analysis, respectively, are replicated two times. In order to perform this replication of hardware blocks, we incorporate the methodology introduced in this paper. We note here that all of these floor-plans were retrieved with HotFloorplan tool [20].

As a reference point for this study, we use the thermal map for a LEON3-based SoC SDR architecture when no replica blocks are assumed. This map, shown in Figure 3(a), exhibits a temperature hotspot in region where the blocks with increased power densities (AHB controller, instruction unit and cache controller) are floor planned. This hotspot results in increased temperature value, as compared to the average onchip temperature, about 7%.

Similar to previous conclusion, the floor-plan that leads to architecture instantiation where components with increased power consumption are replicated (Figure 3(b)), also results in thermal stress. This occurs since the replicated components exhibit low power density, and hence they do not have considerable impact on thermal stress. However, we have to mention that this approach exhibits slightly reduced maximum temperature values, as compared to architecture without replica blocks, since it increases device area (the replicated blocks occupy about 91% of the total chip's area). On the other hand, the components with increased power densities still contribute to thermal stress, as it is shown at Figure 3(b).

Note that for the sake of completeness, the temperature scaling is constant among for all the thermal maps depicted in Figures 3(a), 3(b), and 3(c), in order to be clear that only careful replication of blocks with increased power densities can alleviate thermal stress.

4. Microarchitectural Considerations

For supporting selective block replication, the processor microarchitecture has to be properly enhanced. We mention

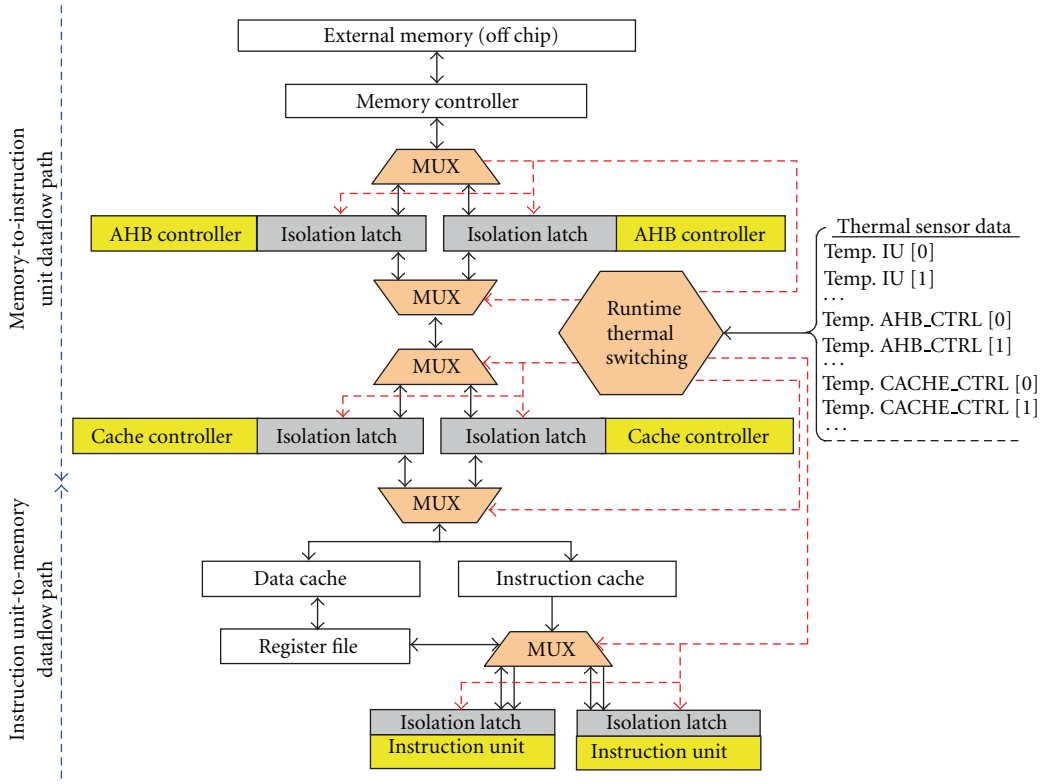

Figure 4: Proposed microarchitectural enhancement.

that in this paper, we focus mainly on the exploration methodology developed for the automatic evaluation of opportunities delivered through selective replication. Thus, in this section, we briefly introduce some micro-architectural considerations that enable the design of processor architectures with replicated components.

In the general case, the data flow and the control flow of the original processor architecture have to be modified towards two directions: (i) enabling mutual exclusiveness between the replicated units, and (ii) permitting run-time management of the replicated resources according to the run-time thermal state of the processor. We focus our analysis on the RISC microarchitecture of LEON3 [18] embedded processor. The block diagram of a SDR system based on LEON3 processor was already depicted in Figure 1.

We target on lightweight enhancements in original LEON3's datapath to avoid extensive area, organization, and control overheads in respect to the original datapath. For this purpose, we apply selective replication in a coarse grained manner, that is, replicating at the level of instruction unit, rather than at the ALU unit or the instruction fetch level. Furthermore, we avoid replication of the actual memory components (i.e., register file, data cache, etc.), since their replication will require proper control mechanisms to establish data coherency among the various replicas.

Although, selective replication in finer granularity than the proposed is a valid design option, we show that coarse-grained component replication can achieve significant temperature reduction and hotspot elimination, which in turn results among others in device improvement against aging phenomena. The proposed approach is not a restrictive one. As shown in Figure 2(b), except the register file which is excluded for replication, the rest of the maximum power densities inside a LEON3 processor are distributed among the replicated components, specifically the instruction unit (*IU*), the cache controller (*CACHE_CTRL*), and the AHB bus controller (*AHB_CTRL*).

According to previous analysis, we propose the adoption of a microarchitectural extension similar to the one depicted in Figure 4. In the examined case, we assume that each component has been replicated two times (this number is parametric in our methodology and its value is defined by the device architect). Each replicated module is enhanced with operand isolation latches [21] to lower redundant dynamic power by eliminating switching activity in time windows in which the component remains inactive. Leakage optimization techniques, that is, power gating, can also be applied in an orthogonal manner. However, in the context of this paper, we only account for dynamic power operand isolation techniques, thus assuming that the inactive component leaks power during its idle time window.

Furthermore, for each replicated component proper pairs of multiplexing and demultiplexing logic are added to the original datapath, regarding the lightweight control and data flow extension. Specifically, the inputs of each replicated component are driven by the demultiplexer that properly guides the input data to the active module. Accordingly, the output signals from each replicated component are multiplexed in order to propagate to the next level. We recognized two data-flow paths inside the processor datapath, namely,

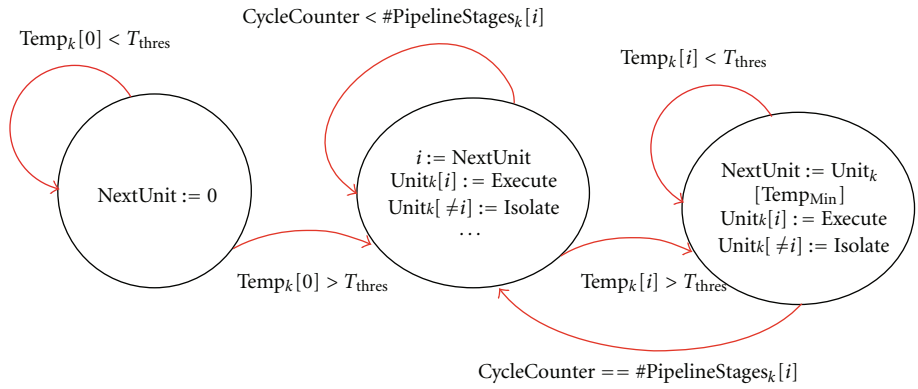

Figure 5: Employed thermal-aware runtime controller per replicated block.

the memory-to-instruction unit and the instruction unit-to-memory data-flow paths. Figure 4 depicts the combination of the two data-flow paths by traversing the architecture graph either in a top-down (memory-to-instruction unit) or in bottom-up (instruction unit-to-memory) manner.

The original LEON3 architecture is also enhanced with a thermal aware runtime controller module for distributing the workload to the available units during runtime. The thermal aware workload distribution is performed by properly issuing the selection signal to the added multiplexing and operand isolation logic. Actually, the same signal configures both the aforementioned components. Since only the selection signals to the extra logic are issued, the thermal aware controller works transparently from the control logic of the rest of the LEON3 architecture. The controller makes the decision which replicated unit to be turned on/off according to the thermal state of the processor. It is assumed that runtime thermal data are available, that is, through thermal sensors.

We consider a reactive scheme of the thermal aware controller. Thus, the controller alters its state whenever an upper temperature threshold, T_{thres}, is crossed. As pointed in [22], the runtime temperature threshold is set to lower value than the one estimated during design time exploration to guarantee proper functionality during execution.

The thermal controller reacts to the temperature readings provided by the onchip thermal sensing infrastructure. $Temp_k[i]$ refers to the temperature value read by the thermal sensor of the replica block i, $i \in \{0, \text{Max}_{\text{replicas}}\}$ that will be active for execution on the next clock cycle, regarding the unit type k, $k \in \{IU, CACHE\_CTRL, AHB\_CTRL\}$. The parameter $Unit_k[i]$ is a table of 1-bit registers. Each one of these registers latches the on/off—*Execute/Isolate* signal for the replica i of unit type k.

The state transitions of the thermal aware runtime controller for a single type of unit, that is, the instruction unit, are depicted in Figure 5. The same control logic is applied to every type of replicated block, k. Since each type of replicated block, k, is managed individually, the overall thermal controller is structured in a modular manner by several control paths like the one depicted in Figure 5, each one dedicated to a specific type of block type. If the monitored temperature is lower than the defined threshold, $Temp_k[i] < T_{thres}$, the controller remains in the same state. Each time the monitored temperature for the active replica block, that is, $Unit_k[i]$, crosses the maximum temperature threshold, $Temp_k[i] < T_{thres}$, the controller enters to an intermediate state. In this intermediate state, the controller maintains its previous configuration for a number of cycles that equals the pipeline stages, $\#PipelineStages_k[i]$ of the specific $Unit_k[i]$. Thus, the controller waits the pipeline of the component to complete its execution, in order to avoid data hazards. In addition, during the intermediate state, the coolest unit among the replicated ones of the same type is extracted using the monitored temperature data, $NextUnit_k := Unit_k[Temp_{\text{Min}}]$. When the pipeline completes its execution, the controller issues the steering signals to the extra multiplexors and operand isolators and reconfigures the control and the data flow of the processor architecture to be performed by the new selected unit (the coolest candidate at the specific time window).

5. Proposed Methodology

This section describes in detail the proposed methodology for reducing temperature hotspots through selective replication for some hardware modules of the target architecture. Note that throughout this methodology we do not aim at redesigning the whole microarchitecture, but we focus only to critical components. The goals of this methodology are: (i) to provide a proactive thermal-aware approach targeting at micro-architecture designs and (ii) to support the rapid exploration/evaluation of different architectural selections in term of thermal stress. Note that the architectural modifications applied with our methodology are transparent to the compilation flow (they do not affect existing tools), while they speed up the development of new products, since end-users (e.g., programmers) do not have to consider thermal issues).

This methodology is shown graphically in Figure 6. The inputs to this methodology are the description of target architecture in VHDL/Verilog, the technology constraints regarding the selected CMOS technology, as well as the operating conditions, and the affordable area overhand

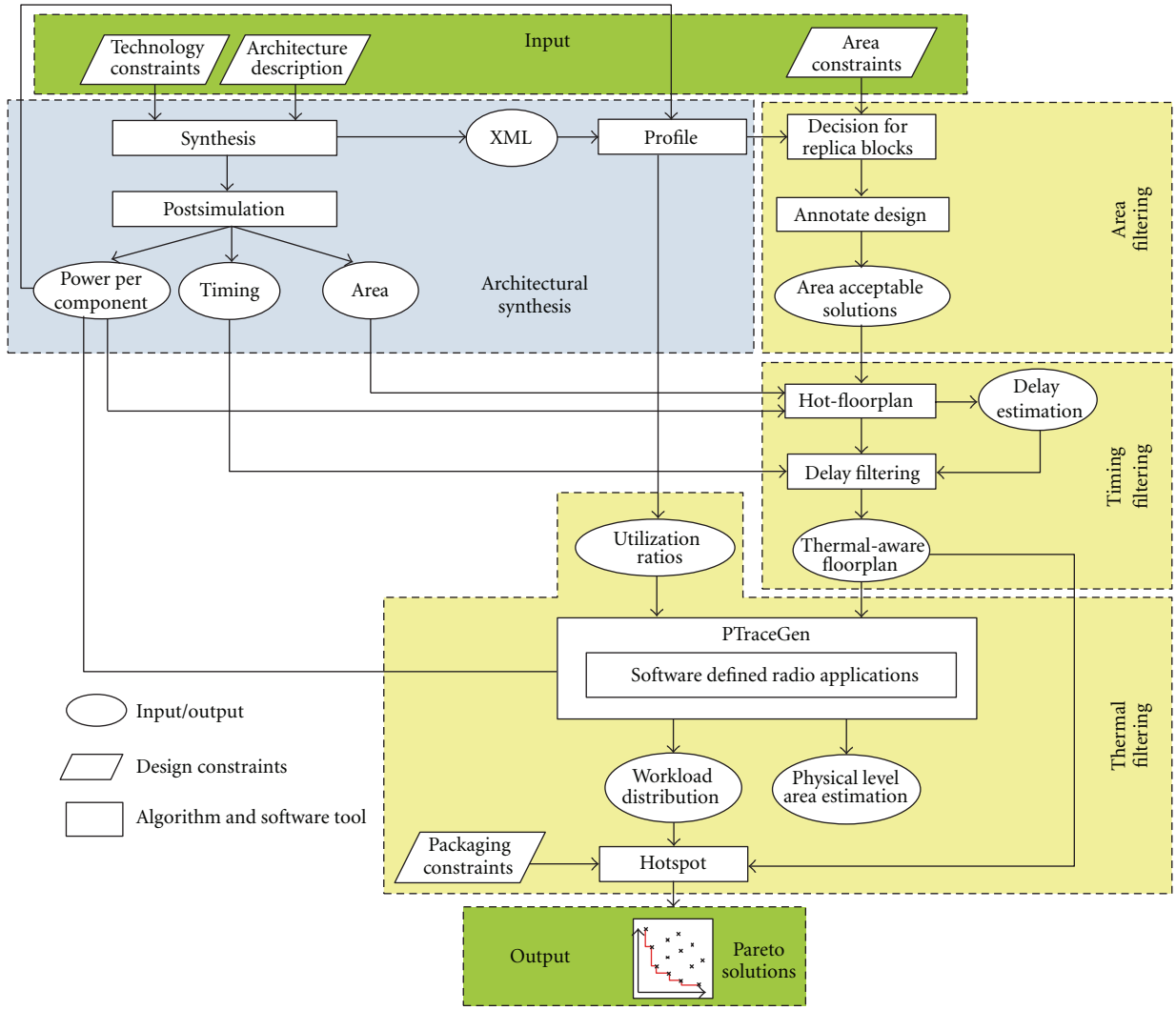

Figure 6: The proposed methodology for replication-aware thermal management.

(the higher area, the more replicas can be integrated into the design). Even though this methodology is applied at design time, the upcoming sections also evaluate the onchip temperature variations during application execution.

5.1. Architecture Synthesis. The inputs to this replication-aware thermal management methodology are the architecture description (in VHDL/Verilog), as well as the selected CMOS technology. Initially, design is synthesized with synopsys design compiler [23] and then we perform post synthesis simulation (with cadence incisive simulator [24]) in order to extract some metrics about the design. Among others, these metrics include the area occupied by each component of the design, as well as its power consumption based on the switching activity of application executed onto the target architecture (the power consumption is retrieved with PrimeTime PX [25]). In addition to that, from the results of post synthesis simulation, it is feasible to have a first estimation regarding the timing of the design (maximum operation frequency) with the usage of Elmore delay model [26]. Note that, both for power and delay study, we employ worst case test vectors as workload to architecture, in order to guarantee that the derived results, as well as the consequence thermal stress, will not be violated for any applications during runtime.

The output of synthesis task is appropriately encoded into an XML format in order to be manipulated by the introduced tools of our framework. The granularity of system's description in this XML format is tunable, since higher detail means a more accurate thermal analysis, but it imposes the maximum computational effort. On the other hand, a more coarse grain approach leads to lower computational effort but it also imposes a penalty in term of hotspot elimination. Even though our methodology is applicable at design time, and hence there is almost no performance degradation due to additional computational complexity, however, the increased number of functionalities inside an SoC usually makes the selection of a fine grain description of target system a non desirable approach.

The derived system's description is profiled in order to compute the power density of the functionalities described in XML file. For this purpose, input regarding power and area

info, as it was already derived from post synthesis simulation, are employed.

5.2. Area Filtering. Based on this analysis, it is possible to make a decision regarding which of the architecture's hardware blocks have to be replicated. For this step, the power density for hardware blocks has to be measured. Note that the total number of replication blocks is limited by area constraints posed by designer.

Since we try to alleviate thermal stress mainly at hardware blocks with increased power densities, these are the blocks that should be replicated (as we have shown in Section 3). For this purpose, hardware blocks of the design are sorted in descending order based on their power density values. From our exhaustive exploration study, we found that only a few of the total blocks exhibit increased power densities. Hence, the architectures that contain all the possible combinations among replica blocks are evaluated. As a constraint to this procedure, we assume the maximum area overhead, as compared to the one retrieved when no replica blocks are assumed.

At this procedure, an extra architectural parameter needs to be defined. More specifically, apart from the blocks that need to be replicated, we also need to clarify the maximum number for each of these blocks that can be replicated. Since more replicas means better thermal management, in the expense of imposing overheads in area and delay, careful study should be applied. In this study, we evaluate solutions that correspond to maximum number of replica blocks up to five. This selection was based on our conclusion that architectures consisted of more replicas do not lead to additional temperature reduction (due to saturation effect). However, constraints posed by architecture specifications might reduce this number.

After defining the type of replica blocks, as well as how many times they should be replicated, the next tool in our framework performs automatically this task by annotating appropriately the design's description. Apart from the insertion of new (replica) blocks to the design, during this task we have to pay effort to provide the appropriate connectivity through routing infrastructure, as well as to insert the thermal-aware runtime controllers in XML format. Moreover, during this annotation we keep the same connectivity among hardware blocks, while we have also to preserve that all the connections to (and from) replica blocks should be also replicated. This is an important differentiation of proposed solution, as compared to similar approaches found in relevant references [6–10], since we do not aim at altering the functionality of underline architecture.

The outcome from this step is all the candidate architecture instantiations that meet area constraints. Such a criterion can eliminate from design space solutions that lead to unacceptable overheads in device area due to excessive number of replica blocks. Also, by allowing a designer-defined overhead in this metric, it is possible to explore and evaluate different architectural solutions. Regarding our exploration, we set this area overhead to 35%, since otherwise our methodology leads to excessive area penalties.

The employed criterion allows blocks with increased power densities to be replicated more times as compared to blocks with smaller values of power density. This occurs because the area occupied from these blocks is usually smaller, and hence more of them are fit into the given (affordable) percentage of area overhead. Since only one of the replica blocks is active at any time (based on approach discussed in Section 4), such an aggressive replication of blocks with increased power densities lead to (i) minimize maximum temperature hotspots and (ii) spread more uniformly over the entire architecture.

5.3. Timing Filtering. Next, we proceed to the second criterion for evaluating the efficiency of derived architecture instantiations that affects the timing constraint. For this purpose, the solutions derived from area filtering are floorplaned with the usage of a thermal-aware floor planner [20].

The optimization goal during this procedure is to reduce the thermal stress for each architecture in respect to the timing constraint. The alleviation of thermal stress is performed by spreading as much as possible the hardware blocks that contribute more to higher values of onchip temperature (e.g., modules with increased power densities). Similarly, by minimizing the perimeter of bounding box that surrounds all the modules that are connected with a single bus, it is possible to improve the delay of this bus. Hence, blocks that are connected through bus(ses) have to be floorplaned in spatially close locations.

Regarding our methodology we allow all the blocks of the chip to be "soft" blocks, that is, their aspect ratio can change (in a controlled manner) in each annealing movement of *Hot floor plan* tool, but their area is fixed.

The derived solutions are then evaluated in term of delay degradation, as compared to delay estimation retrieved from postsynthesis simulation. For this scope, we use the Elmore delay model [26]. Since we are primarily interested to retrieve a thermal-aware solution, we already knew that a penalty in architecture's performance is affordable. For our study, the timing overhead is assumed to be 14% in order to avoid any mentionable delay overheads.

The output of this step is all those thermal-aware floorplanned solutions which their timing degradation meets the design specification (as it was derived after postsynthesis simulation without considering yet any replica block).

5.4. Thermal Filtering. Finally, in the last task in the proposed methodology, the different architectural solutions are evaluated against thermal constraint. This task is automated with a new tool, named *PTraceGen*, that generates proper power traces onto the examined processor architecture regarding the statistical behavior of the targeted application domain. For this purpose, the outcome from this tool provides detailed information about architecture's units that describe (i) the amount of time that a unit is active, (ii) the amount of time that a unit is inactive, and (iii) the switching activity of these units. The first two parameters (i.e., amount of active/inactive time) refer to the number of execution cycles for a timing window, whereas the third parameter corresponds to the number of transitions between

operational states (active/inactive) during the architecture's execution phase.

Specifically, the statistical characterization is performed in a window-based manner (denotes the temporal granularity for performing thermal analysis) by computing the statistical mean values of the primitive operations executed by the processor, that is, number and type of ALU instructions, memory accesses, cache hits/misses, communication load, and so forth. For this purpose, power traces for all the replica units are generated. These traces describe the power activity for all the architecture's units (replicated or not) within a certain amount of time. Using these statistics, the utilization ration per component is extracted in each examined window. The per component utilization ratios are correlated with accurate postsynthesis power measurements to generate the power traces. For the examined power traces, the thermal filtering tool extracts the Pareto frontiers under various design objectives (i.e., power density of the chip, delay, area, max temperature, max thermal gradient, etc.).

In this paper, we study a uniform workload incorporating the key elements of base band processing domain, where these optimizations against to thermal stress can be applied. More specifically, the employed applications, obtained from [27] are summarized, as follows.

(i) Adaptive differential pulse-code modulation (ADPCM) is a variant of differential pulse-code modulation (DPCM) that varies the size of the quantization step, to allow further reduction of the required bandwidth for a given signal-to-noise ratio.

(ii) Cyclic redundancy check (CRC) is an error-detecting code designed to detect accidental changes to raw computer data, and is commonly used in digital networks.

(iii) Fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse.

(iv) GSM 06.10: GSM 06.10 is a digital speech coding standard used in the GSM digital mobile phone system.

These telecommunication applications exhibit increased demand for bandwidth requirements. Regarding our architecture, the components with higher power density values when such kind of applications is executed are the instruction unit, the cache and memory controller, the DSU (debug support unit), and the AMBA AHB/APB bus controllers.

The output of *PTraceGen* tool is a set of workload distributions based on employed applications, as well as accurate area estimation after floor plan. More specifically, the workload distribution is achieved by appropriately handling the power info per block of the target architecture (component of the selected level of granularity), as they were already derived from postsynthesis simulation. This approach guarantees that hardware blocks (replicated or not) dissipate power consumption only the time periods denoted by the application's functionality.

These time periods are retrieved by incorporating info from application's simulation. The employed utilization ratios are averaged over hardware components of LEON3 system in order to determine the active/idle time slots accurately. The output from utilization is fed as input to *PTraceGen* tool in order to redistribute the activity of each component to a timing trace of several time slots. This is achieved by setting each of these components either as active or idle for every time slot.

Table 1: Thermal characteristics of the employed processors.

Parameters	Model value
Sampling interval	20 ms
Die Thickness	0.15 mm
Core Area (no replication)	0.426213 mm^2
Cache area (L1 + Local I + D)	0.370561 mm^2
Convection Resistance	0.1 K/W
Convection Capacitance	140.4 J/K

Furthermore, the area derived in this stage may be different from the one computed during area filtering, due to additional free space inserted to design after floor plan (which is not occupied by any hardware block) in order to model the white space between hardware components.

The workload distribution, in conjunction to the packaging constraints, is fed to the *hotspot* tool [19] to compute thermal profile of target architecture. For increased accuracy, we used the steady-state temperature of each hardware block as the initial temperature values. The default characteristics of *hotspot* version 5 were used for package, whereas the analytic model parameters are summarized in Table 1.

Based on the derived thermal profile, it is possible to evaluate the architecture instantiation in terms of different criteria tightly firmed to onchip temperature. For the scope of this paper, all the solutions that do not meet the selected thermal constraints (maximum temperature and the temperature gradient) are eliminated from exploration space, whereas typical packaging for embedded processors is assumed [28]. Thus, the output of the proposed methodology contains only the instantiations that correspond to architectural solutions that meet all the three constraints, namely, area, timing. and thermal.

6. Experimental Results

This section provides a number of experimental results derived from the proposed exploration methodology that prove the efficiency of our approach in term of reducing the consequences posed by thermal stress. For this purpose, a LEON3-based design is employed [18], and the introduced methodology is applied to identify the blocks that lead to higher temperature reduction, and hence they should be selectively replicated, as it was already discussed in Section 5.

We have to mention that functionality of the underlined LEON3 processor is not affected by the additional (replica) blocks.

The majority of aging phenomena are tightly firmed to onchip temperature values. Hence, higher maximum temperatures lead among others to devices having increased

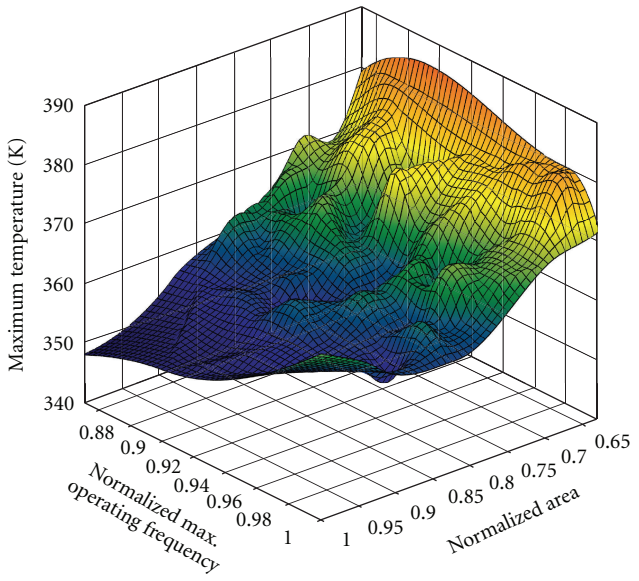

Figure 7: Temperature variation for different instantiations of target architectural.

failure rates. For this reason, the first criterion employed in our methodology for selecting the architecture of target platform involves to study how temperature values are spatially distributed over the target device.

6.1. Impact of Selective Replication on Temperature. Figure 7 depicts the variation of maximum temperature (in Kelvin) when different instantiations of the target architecture are considered. The axes of this figure denote the normalized operation frequency and the architecture's area, as compared to the corresponding maximum values found among all the candidate architectures, whereas the vertical axis gives the onchip temperature values. Based on Figure 7, it is evident that temperature values vary considerable among architectures with different selection of replica blocks. More specifically, regarding the LEON3 architecture, temperature variations from 354 to 382 Kelvin's, were reported.

The following conclusions can be derived from this figure. More specifically, as we increase the area of target architecture, the temperature values are reduced (almost monotonically). However, this temperature reduction in not constant since just replication of blocks does not guarantee alleviation of thermal stress (as we have already depicted in Figure 3(c)). Hence, apart from the number of replica blocks, their properties (e.g., power density, area, power consumption, etc.) are also crucial for designing an efficient architecture.

Apart from area, the maximum operation frequency also affects the onchip temperature values. Based on Figure 7, the alternative architectures lead to performance variations up to 14%, which mainly occur due to (i) additional replica blocks inserted into the design, (ii) the consequent different floorplans, and (iii) the increased wire-length for connecting these blocks.

Another interesting conclusion might be derived from Figure 7. Even thought higher operation frequencies usually result in higher temperatures, this seems to be alleviated when architectures with increased area are assumed. Regarding the LEON3 architecture, the additional area is dominated by blocks with low power density values (as we have already mentioned in Section 2). Hence, by introducing more replica blocks, it is possible to improve thermal spreading. However, since temperature values seem to be more agnostic about the maximum operation frequency, as compared to area overhead, we cannot make safely any conclusion about this.

Based on this figure it is possible to select an architecture that better trades-off design constrains. A balanced design solution under the aforementioned criteria (area, maximum temperature, and delay) is the one that replicates four AHB controllers, three integer units, and two cache controllers. This architectural instantiation, mentioned as "selected architecture" in upcoming figures, belongs to solutions marked as valid during the area, timing, and thermal filtering. The selection of this architecture for further evaluation is performed since it belongs to the Pareto front for reliability improvement, as it is discussed in more details in Section 6.2.

More specifically, the area and delay overheads for our selected replication-aware LEON3 design are 15% and 7%, respectively, as compared to initial implementation (without considering any replica blocks). Even though these penalties are not negligible for ASIC designs, we have to mention that they comes with considerable gains in term of maximum temperature value (about 17 Kelvin or 8%), which in turn leads to higher reliability improvements. Furthermore, the proposed methodology for selectively replication of blocks with increased power densities can also be applied to -core architectures, where the performance degradation is more affordable.

Apart from the selected architecture, any other architecture instantiation can be chosen without affecting the efficiency of our proposed methodology, if different constraints are applied. Note that our framework mainly indents to enable the thermal improvement of architectures through inserting replica blocks.

In order to show the importance of proper identification for hardware blocks that have to be replicated, Figure 8 plots the temperature variation (in Kelvin) versus the power density (W/cm^2) for each instantiation of the selected architecture with replica blocks. We choose to evaluate such a criterion, because power density for existing and upcoming devices becomes a major issue for architects. Researchers have already identified this problem, whereas based on projections it is expected that power density regarding 14 nm nodes will be higher than 100 W/cm^2 [29].

In order to plot this graph, architectures are grouped into three categories based on their power densities, as follows.

(i) those with area smaller than the 33% of maximum area among all the solutions;

(ii) those with area ranging between 33% and 66% of the maximum area among all the solutions;

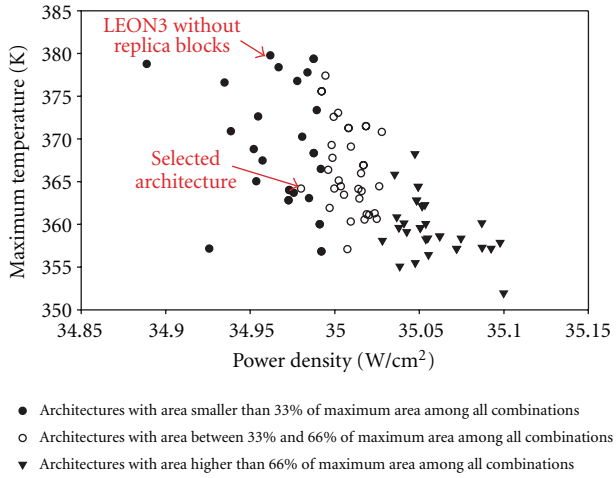

FIGURE 8: Results about power density versus maximum temperature.

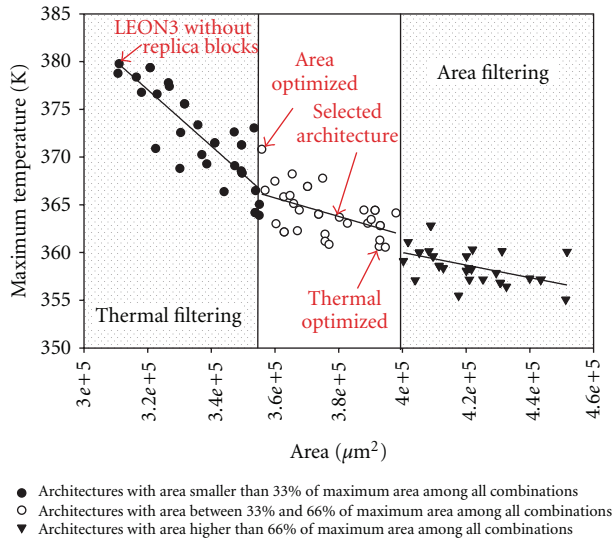

FIGURE 9: Results about area versus maximum temperature.

(iii) those with area higher than 66% of the maximum area among all the solutions.

Note that this classification with respect to area occupied by different architectures is also applied in upcoming figures (Figures 9 and 10), since it can provide qualitative comparisons about the importance, as well as the efficiency, of the proposed methodology in term of alleviating thermal stress.

Since different architectures consist of different replicated blocks, their power densities also vary. As we can conclude from Figure 9, there is not a straightforward correlation among the occupied area, the overall power density and maximum temperature. More specifically, regarding architectures shown in Figure 9 that correspond to increased area (more than 66% of the maximum area), they seem to exhibit the maximal power densities but the lower temperature values. On the other hand, the architectures with smaller area (less than 66% of the maximum area) exhibit reduced

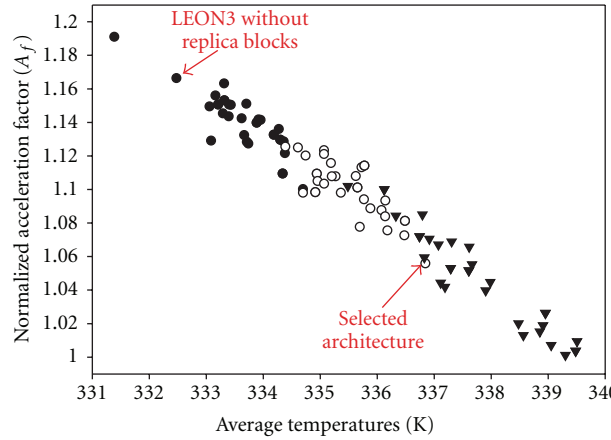

FIGURE 10: Evaluation in term of A_f parameter for architectures with different average temperatures.

power densities and increased temperatures values. This rather strange result is justified by the fact that the type of the replicated blocks is not considered in the previous analysis. Thus, we can conclude that the type of the replication block has a great impact on the thermal behavior of the silicon.

The last conclusion is very interesting since it shows that even architectures with increased power densities can achieve considerable onchip temperature reduction. This point verifies the argument discussed in Section 3, that nonoptimal replication of blocks leads to similar (or higher) thermal profiles, as compared to initial architecture implementation (without replica blocks).

In Figure 9, we have also marked the solutions which correspond to previously mentioned selected architecture, as well as to the original (without replica blocks) LEON3. Based on this, both those two solutions exhibit comparable power densities, but the selected one achieves to reduce maximum temperature value about 17 Kelvin, or 8%.

In order to study the correlation between areas occupied by target architectures and the maximum temperature values, Figure 9 plots the corresponding diagram of these parameters. For the sake of completeness, we cluster alternative architectures into three groups based on their area (similar to previous figure), while for demonstration purposes we also compute the temperature gradient for each of these clusters.

Based on Figure 9, the maximum temperature gradient occurs for architectures with smaller areas, whereas for architectures with area more than 66% of the maximum area is almost constant. If we take into consideration also the smaller delay overhead posed by devices consisted of fewer replica blocks (as it was already shown in Figure 7), an additional filtering of derived architectural instantiations can be performed. More specifically, without affecting the generality of the proposed methodology, solutions that correspond to area overheads higher than 66% of the maximum area are

assumed not to be desirable (due to additional delay), and hence they are eliminated from exploration space.

Furthermore, the temperature values for architectures with few replicas (less than 33% area overhead) are about 3x higher, as compared to the remaining solutions. Since our methodology tries to alleviate the hotspots, such high-temperature variations usually result in increased cost for packaging and cooling, and hence they are not desirable. These solutions can also be eliminated from exploration space.

This conclusion is very important in order to find the amount of blocks that have to be replicated. In other words, based on Figure 9, it is clear that only a few replicas of the blocks with increased power densities should be incorporated, in order to achieve the desired balance between temperature reduction and the consequences area and delay overheads.

6.2. Impact of Temperature on Reliability. Reliability is defined as the probability that a device will perform its required function under stated conditions for a specific period of time. Predicting with some degree of confidence, strongly depends on defining a number of parameters.

Accelerated life testing employs a variety of high-stress test methods that shorten the life of a product, or quicken the degradation of the products performance. The goal of such testing is to efficiently obtain performance data that, when properly analyzed, provide reasonable estimates of the products life or performance under normal conditions. This induces early failures that would sometimes manifest themselves in the early years of a products life, and also allows issues related to design tolerances to be discovered before volume manufacturing. Both the type of stressor and the time under test are used to determine the normal lifetime. Regarding SoC designs, usually the majority of these aging degradation mechanisms are tightly firmed to onchip temperature values.

The effect of these stressors can be mathematically determined. Next we model aging acceleration due to thermal stress with the usage of Arrhenius equation (1). More specifically, this equation models how the age of a product is increased when it operates under higher temperature values, as compared to its normal operating temperature. Figure 10 plots how this parameter varies for different architectural instantiations discussed in this paper. More specifically, the horizontal axis in this figure gives the average temperature, whereas the vertical axis shows how the onchip temperature affects aging degradation:

$$A_f \propto \exp\left[\left(\frac{E_a}{k}\right) \times \left(\frac{1}{T_u} - \frac{1}{T_t}\right)\right], \quad (1)$$

where A_f is the acceleration factor, E_a is the activation energy in electron volts (its value is 0.5 eV for silicon defects), and k is Boltzmann's constant, whereas T_u and T_t are the reference (Kelvin) and the operation temperature during testing.

Based on the values depicted in this figure, we can conclude that the selected architectural instantiation achieves almost the minimum value for A_f parameter among all the candidate solutions. Furthermore, the conventional approach for designing LEON3 architecture exhibits about 14% higher value for this parameter. This mainly occurs due to additional thermal stress introduced by architecture's components with increased power density. However, we have to mention that even this no-replica aware architecture is not the one with the maximum aging degradation, since there exist solutions that correspond to A_f value up to 20% of the selected solution (this occurs since non all the possible combinations of blocks lead to alleviate the thermal stress).

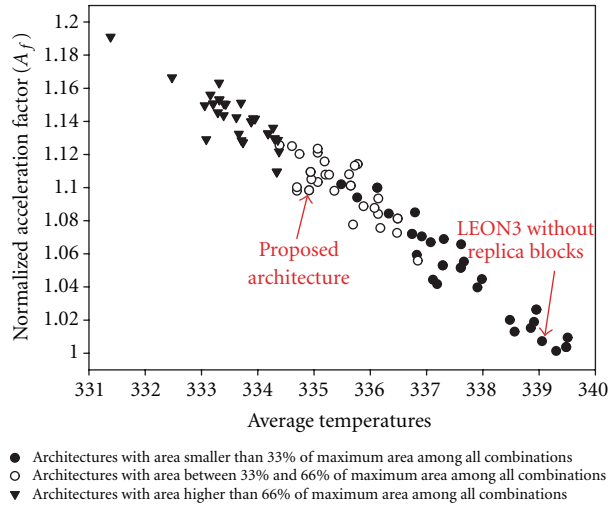

FIGURE 11: Evaluation of different architectures under TDDB.

Apart from Arrhenius equation, we also evaluate the different architectures derived during our exploration, under the time-depended dielectric breakdown (TDDB) [30]. Since oxide breakdown has already been of serious reliability concern in the semiconductor industry because of the continuous trek towards smaller device sizes, such kind of aging phenomenon should be carefully studied. Defects occurred due to TDDB are primarily caused due to the trapping of charges in the oxide that create an electric field, followed by charge flow through the oxide, resulting in a breakdown after sometime. The MTTF due to TDDB phenomenon is described by

$$\text{MTTF} = A_0 \times \exp(-\lambda \times E_{\text{ox}}) \times \exp\left(\frac{E_a}{k \times T}\right), \quad (2)$$

where λ is a field acceleration parameter, which is temperature dependent.

Figure 11 plots the variation of mean time to failure parameter for the different architectures, as they derived from our exploration framework. For demonstration purposes, the values at vertical axis were plotted in normalized manner over the corresponding MTTF for the initial architecture instantiation of LEON3 processor (without replica blocks).

Based on Figure 11, a number of conclusions might be derived. Among others, as we increase the maximum onchip temperature values, the MTTF also increases. This is explained due to the tight correlation between aging phenomena and temperature values. Additionally, the selected

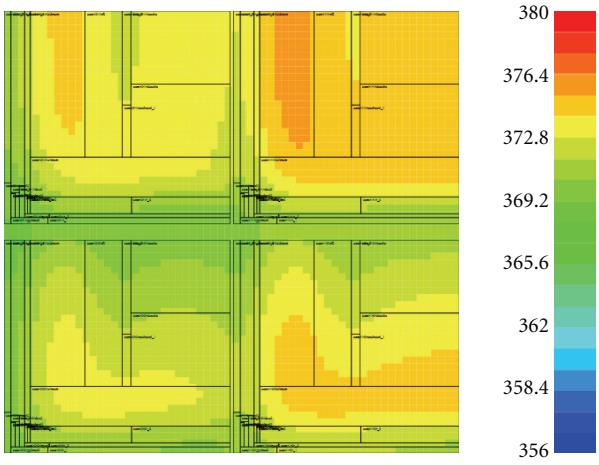

Figure 12: Thermal profile for 2 × 2 CMP LEON3-based architecture.

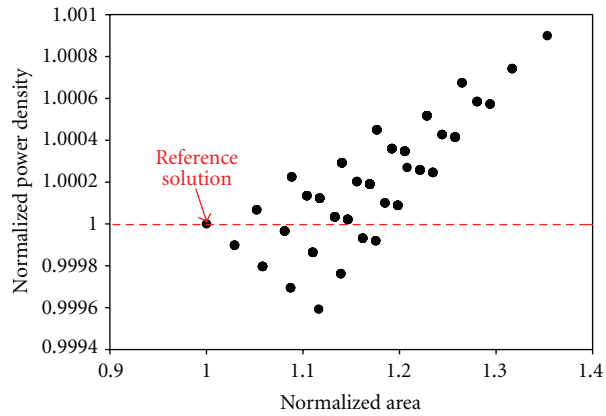

Figure 13: Normalized power density versus area overhead for multiprocessor LEON3.

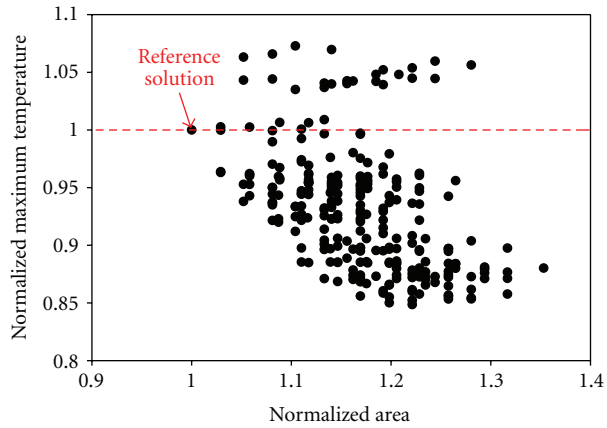

Figure 14: Normalized maximum temperature versus area overhead for multiprocessor LEON3.

architecture achieves to improve the MTTF parameter, as compared to initial instantiation of LEON3 processor (without considering any replica blocks), about 14%. We have to notice that in case we increase the architecture's area, through inserting more replica blocks, the MTTF parameter can be further increased. This is due to the fact that architectures with more replica blocks usually occupy more area, which in turn improve thermal spreading. However, as we have already mentioned, such an improvement in MTTF parameter (from 0.86 to 0.81) leads to architectures with unacceptable area penalties (they have eliminated with area filtering as depicted in Figure 9).

6.3. On Designing Chip Multiprocessors for SDR. This subsection describes the results retrieved of applying the proposed methodology for designing chip multiprocessor architectures. For demonstration purposes, the target multiprocessor consists of four instances of LEON3 (this number is parametric to our framework and can be appropriately tuned based on designer's requirements), while the replica modules among LEON3 processors for a given instantiation of multiprocessor, are the same. As a reference to this study we employ a multiprocessor architecture consisted of the LEON3 which was marked as "selected" in the previous figures. For the following figures, this solution is denoted as "reference solution."

Figure 12 gives the thermal profiles for this architecture, as it was retrieved from Hotspot tool. Based on this figure, we can conclude that a number of modules per LEON3 exhibit increased temperature values. Note that these hotspots differ from those reported previously at Figure 3, due to (i) the thermal diffusion effect, and (ii) the different floorplans for each LEON3 processor.

Next, we will quantify the efficiency of the above solution when this is used in multiple instantiations of a multiprocessor architecture.

Figure 13 shows the variation of power density as we select solutions with higher area overheads. Based on this figure, we can conclude that power density seems that it does not depend on silicon area. We have to mention that, during this analysis, the additional area mostly occurs due to the whitespace between hardware components, rather than the area of these replica components.

In contrast to our conclusion about power density, area has a great impact on the maximum onchip temperature values. This is also depicted at Figure 14.

More specifically, based on this figure we can conclude that a controllable area overhead (e.g., 20% increase as compared to the multiprocessor solution composed of LEON3 components selected previously) leads to the reduction of the maximum temperature by almost 0.85x of the previous corresponding value.

Notice that architecture instantiations of the same area exhibit temperature variations due to the different components that are replicated.

7. Conclusions

In this paper, we propose the adoption of selective replication techniques in order to optimize the thermal behavior of

the synthesized microprocessor systems targeting at an SDR system. We developed an automated exploration methodology that permits the thermal aware evaluation of various micro-architectural instantiations.

We show that by using selective replication, we can deliver optimized architectural solutions with minimum thermal stress, affordable delay, and user-constrained area overheads. Experimental results have shown a significant reduction of the maximum operating frequency, by 8%, which in turn leads to improvement at maximum on-chip temperature values. Moreover, they have shown that our approach improves by 14% the aging phenomena. These two results show that our approach compares favorably to the conventional design techniques for SoC-based SDR architectures.

References

[1] T. Ulversoy, "Software defined radio: challenges and opportunities," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 4, pp. 531–550, 2010.

[2] J. Glossner, E. Hokenek, and M. Moudgill, "The sandbridge sandblaster communications processor," in *Proceedings of the 3rd Workshop on Application Specic Processors (WASP '04)*, vol. 38, pp. 53–58, Stockholm, Sweden, September 2004.

[3] Y. Lin, H. Lee, M. Woh et al., "SODA: a low-power architecture for software radio," in *Proceedings of the 33rd International Symposium on Computer Architecture (ISCA '06)*, pp. 89–100, Boston, Mass, USA, June 2006.

[4] A. Coskun, T. Rosing, K. Mihic, G. Micheli, and Y. Leblebici, "Analysis and optimization of mpsoc reliability," *Journal of Low Power Electronics*, vol. 2, no. 1, pp. 56–69, 2006.

[5] V. Gektin, A. Bar-Cohen, and J. Ames, "Coffin-manson fatigue model of underfilled flip-chips," *IEEE Transactions on Components Packaging and Manufacturing Technology A*, vol. 20, no. 3, pp. 317–326, 1997.

[6] Y. Liu, H. Yang, R. P. Dick, H. Wang, and L. Shang, "Thermal vs energy optimization for DVFS-enabled processors in embedded systems," in *Proceedings of the 8th International Symposium on Quality Electronic Design (ISQED '07)*, pp. 204–209, San Jose, Calif, March 2007.

[7] M. Harchol-Balter and A. B. Downey, "Exploiting process lifetime distributions for dynamic load balancing," *ACM Transactions on Computer Systems*, vol. 15, no. 3, pp. 253–285, 1997.

[8] J. Yang, X. Zhou, M. Chrobak, Y. Zhang, and L. Jin, "Dynamic thermal management through task scheduling," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS '08)*, pp. 191–201, Austin, Tex, USA, April 2008.

[9] D. Atienza, V. Del, G. Pablo et al., "A fast hw/sw fpga-based thermal emulation framework for multi-processor system-on-chip," in *Proceedings of the 43rd annual Design Automation Conference (DAC '06)*, pp. 618–623, San Francisco, Calif, USA, July 2006.

[10] M. M. Sabry, J. L. Ayala, and D. Atienza, "Thermal-aware compilation for system-on-chip processing architectures," in *Proceedings of the 20th ACM Great Lakes Symposium on VLS (GLSVLSI '10)*, pp. 221–226, May 2010.

[11] M. Monchiero, R. Canal, and A. Gonzalez, "Power/performance/thermal design-space exploration for multicore architectures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 666–681, 2008.

[12] J. M. Rabaey, *Low Power Design Essentials*, Springer, New York, NY, USA, 2009.

[13] H. Li, P. Liu, Z. Qi et al., "Efficient thermal simulation for run-time temperature tracking and management," in *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '05)*, pp. 130–133, San Jose, Calif, USA, October 2005.

[14] W. Huang, S. Velusamy, K. Sankaranarayanan, K. Skadron, M. Stan, and A. Tarjanet, "Temperature-aware microarchitecture: extended discussion and results," Tech. Rep. CS-2003-08, University of Virginia, Computer Science Department, Charlottesville, Va, USA, 2003.

[15] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *Proceedings of the the 27th Annual International Symposium on Computer Architecture (ISCA '00)*, pp. 83–94, June 2000.

[16] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA '01)*, pp. 171–182, Monterrey, Mexico, January 2001.

[17] http://proteas.microlab.ntua.gr/ksiop/software.

[18] Gaisler, Leon3, http://www.gaisler.com/.

[19] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, "Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1277–1288, 2008.

[20] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *Journal of Instruction-Level Parallelism*, vol. 7, pp. 1–16, 2005.

[21] A. Raghunathan, N. K. Jha, and S. Dey, *High-Level Power Analysis and Optimization*, Kluwer Academic, New York, NY, USA, 1998.

[22] F. Zanini, D. Atienza, and G. De Micheli, "A control theory approach for thermal balancing of MPSoC," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '09)*, pp. 37–42, Yokohama, Japan, January 2009.

[23] Synopsys Incorporation, http://www.synopsys.com/.

[24] Cadence Incorporation, http://www.cadence.com/.

[25] Prime Time PX, http://www.synopsys.com/tools/implementation/signoff/pages/primetime.aspx.

[26] S. Sapatnekar, "Rc interconnect optimization under the elmore delay model," Tech. Rep. ISU-CPRE-94-SS03, 1994.

[27] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: a free, commercially representative embedded benchmark suite," in *Proceedings of the 4th IEEE International Workshop on Workload Characterization (WWC '01)*, pp. 3–14, December 2001.

[28] S. Canumalla and P. Viswanadham, *Portable Consumer Electronics: Packaging, Material, and Reliability*, PennWell Corporation, Tulsa, Okla, USA, 2010.

[29] ITRS, International technology roadmap for semiconductors. 2009.

[30] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "Lifetime reliability: toward an architectural solution," *IEEE Micro*, vol. 25, no. 3, pp. 70–80, 2005.