Optimizing Temperature Guardbands

Hussam Amrouch*, Behnam Khaleghi† and Jörg Henkel*
*Karlsruhe Institute of Technology, Chair for Embedded Systems (CES), Karlsruhe, Germany
† Sharif University of Technology, Tehran, Iran
{amrouch; henkel}@kit.edu

Abstract*—We introduce the first temperature guardbands optimization based on thermal-aware logic synthesis and thermal-aware timing analysis. The optimized guardbands are obtained solely due to using our so-called thermal-aware cell libraries together with existing tool flows and not due to sacrificing timing constraints (i.e. no trade-offs). We demonstrate that temperature guardbands can be optimized *at design time* through thermal-aware logic synthesis in which more resilient circuits against worst-case temperatures are obtained. Our *static guardband optimization* leads to 18% smaller guardbands on average. We also demonstrate that thermal-aware timing analysis enables designers to accurately estimate the required guardbands for a wide range of temperatures without over/under-estimations. Therefore, temperature guardbands can be optimized *at operation time* through employing the small, yet sufficient guardband that corresponds to the current temperature rather than employing throughout a conservative guardband that corresponds to the worst-case temperature. Our *adaptive guardband optimization* results, on average, in a 22% higher performance along with 9.2% less energy. Neither thermal-aware logic synthesis nor thermal-aware timing analysis would be possible without our thermal-aware cell libraries. They are compatible with use of existing commercial tools. *Hence, they allow designers, for the first time, to automatically consider thermal concerns within their design tool flows even if they were not designed for that purpose.

***Index Terms*—Temperature, Guardband, Performance, Timing Analysis, Logic Synthesis, Cell Library**

Download Software: This work is publicly available at [1] http://ces.itec.kit.edu/dependable-hardware.php

I. Introduction

The nano-CMOS era introduces thermal challenges due to the excessive on-chip power densities. To guarantee a reliable operation of circuits, designers need to employ a timing guardband that corresponds to the worst-case temperature in which circuits will always be clocked at a sustainable frequency at operation time. This is due to the fact that transistors become noticeably slower as temperature increases due to the reduction in their carriers mobility (i.e. the drift velocity of a particle in an electric field) [2]. In practice, a *temperature guardband* in this context is a time slack on top of the maximum delay of circuit (i.e. critical path) representing the projected delay increase when the temperature rises from the typical value considered at design time (e.g., 25°C) to the worst-case value that may occur at operation time (e.g., 85°C). While, on the one hand, such a solution guaranties a reliable operation of logic within the desired temperature range, it leads, on the other hand, to an considerable performance loss as the operating frequency will be conservatively restricted to the worst-case temperature even though the temperature might be lower during the majority of operation time. Furthermore, the worst-case temperature is typically determined based on the capability of the applied cooling/packaging. Due to the high cost of cooling, employing larger guardbands to compensate becomes necessary which exacerbates the incurred performance loss.

Traditionally, temperature effects are often considered as post-silicon guardbanding [3] in which the required guardband is determined through measuring the chip's delay after testing it under the worst-case temperature. However, this may not be always appropriate as designers may need to know at design time the accurate thermal behavior of their circuits to identify whether performance constraints will be met or not. To tackle this challenge, we propose to employ the available capabilities of existing EDA tool flows (e.g., Synopsys) in order to optimize temperature guardbands which are indispensable to overcome temperature-induced delay uncertainty.

To achieve that, we create thermal-aware cell libraries in which each library contains – for a specific temperature – the detailed delay and power information of every gate/cell. Building upon these libraries, we leverage mature optimization algorithms within commercial tool flows to a) design circuits that exhibit a better performance at the worst-case temperature and b) accurately estimate the required guardband at varied temperatures. In fact, creating such cell libraries is prerequisite because the same temperature rise *disproportionally* increases the gates delay. To demonstrate that, we show in Fig. 1 the distribution of delay increase of gates within a 45nm standard cell library when the temperature rises from 25°C to 65°C and 85°C. Furthermore, not only different gates are disproportionally influenced by the same temperature rise, but even a gate itself may exhibit different delay increases under the same temperature rise based on its $OPCs$' (i.e. the signal slew on its inputs and load capacitance on its output) as it will be investigated in Section III. Hence, creating thermal-aware cell libraries that contain the detailed gates behavior under different temperatures is essential. Otherwise, the overall impact of a temperature rise on increasing the paths delay of a circuit will be inaccurately estimated and therefore a temperature guardband, larger than what is actually required, would be necessary to cope with the uncertainty.

Our novel contributions within this paper are as follows:
(1) We introduce thermal-aware cell libraries *that are publicly available at [1]*. Through solely using them with existing tool flows, optimized temperature guardbands can be obtained.
(2) We propose a *static guardband optimization* through bringing thermal awareness to logic synthesis to obtain circuits that exhibit a better performance at the worst-case temperature.
(3) We also develop an *adaptive guardband optimization* technique in which the guardband is tuned at operation time based on the current temperature rather than employing throughout a conservative guardband based on the worst-case temperature.

II. Related Work

Based on the contribution of our work, we categorize the related work into the following two categories:

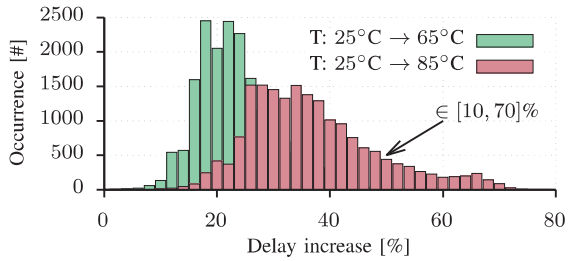


Fig. 1. Thermal analysis for a 45nm standard cell library showing how a temperature rise *disproportionally* increases the delay of gates/cells.

a) Minimizing guardbands: Works that aim at minimizing the peak temperature and hence shrinking the required guardband. There are plenty of them based on e.g., high-level synthesis, task scheduling, etc. *It is noteworthy that such techniques are orthogonal to our work and can be employed on top of our static and adaptive guardband optimization techniques.* [4] proposed an integer linear programming that takes thermal/power models into account for scheduling tasks on MPSoCs towards reducing the peak temperature. TAPHS [5] proposed high-level synthesis and floorplanning techniques to reduce the peak temperature through iterative rescheduling, resource sharing and thermal-aware floorplanning. [6] proposed floorplanning framework based on genetic algorithms to evenly distribute the temperature across the chip and thus avoiding hotspots. Note that combining any technique, which leads to a lower worst-case temperature, with our optimization provides further improvements due to the selection to even smaller guardbands either statically or adaptively. Recently, [10] proposed an aging-aware logic synthesis to minimize the required guardband of aging at design time using synthesis tool flows.
(b) Adapting guardbands: Works that aim at exploiting the relation between the temperature and circuit's delay to increase the frequency though adapting the guardband whenever the temperature is lower than the worst case. [7] proposed a voltage/frequency (V/F) scaling that relies on an off-line temperature-aware optimization in which the algorithm assumes an initial temperature for each task and then selects the corresponding V/F pair to accomplish the task within its deadline. Theoretically, the algorithm could also be used to employ higher frequencies at lower temperatures – even though this was not the focus of the paper. The key disadvantage is that the V/F might be underestimated at the beginning and thus errors due to timing violations temporarily occur at runtime. In fact, predicting the future temperature is nontrivial as it is subject to e.g., signal activities, ambient temperature, leakage feedback loop etc. Thus, erroneous V/F pairs may be selected due to temperature misspredictions. [8] increases the frequency by 3% per every 20°C reduction from the worst-case temperature. However, it imposes pessimistic guardbands and thus it does not exploit the prospect of increasing the frequency at finer-grained temperature steps due to the lack of accurate thermal analysis at such granularity. Razor [9] reduces the voltage of a pipelined design until an error is detected by a shadow flip-flop. Besides overheads, Razor increases the design complexity due to the replica flip-flops and detection logics in any path with the potential of becoming critical. Furthermore, such an approach might not be possible to be implemented in non-pipelined designs as, unlike the pipelined designs, occurring erroneous data cannot be recovered through stalling the stages.

Distinction from state of the art:

- We propose thermal-aware logic synthesis in which circuits are synthesized at the targeted worst-case temperature allowing for a smaller guardband. i.e. higher performance.
- We propose thermal-aware timing analysis to accurately determine the small, yet sufficient guardband at varied fine-grained temperature steps allowing for developing an adaptive guardband optimization.
- We create thermal-aware cell libraries in which they can be used within existing EDA flow tools. Without them, none of the aforementioned two objectives would be possible.

III. Motivation

As Fig. 1 shows, different gates exhibit distinct timing increase under the same temperature rise. While the delay of some gates at 85°C may increase by 70% compared to 25°C, the delay of others may merely increase by 10%. From such a considerable variation, the following key points can be raised:
(a) Critical Path Changeability: Because different gates/cells may disproportionally be influenced by a temperature rise, a path which was formerly non-critical at the typical temperature may become critical at a higher temperature and vice versa. This is demonstrated in Fig. 2 which shows how path_1 which was critical at 25°C ($103\,\text{ps} > 95\,\text{ps}$), became later non-critical at 85°C because the overall impact of temperature rise on path_2 is higher ($140\,\text{ps} > 135\,\text{ps}$). Therefore, analyzing the thermal behavior of the original critical path "CP" only is not sufficient. However, applying the worst-case delay increase (i.e. 70%) will indeed result in a pessimistic guardband and thus considerable performance loss. Hence, it is indispensable to analyze the timing of the entire circuit netlist – under a particular temperature rise – to accurately (i.e. without over/under-estimations) determine the required guardband.
(b) Operating Conditions Role: As some gates may marginally be influenced by a temperature rise, it can be inferred that using a specific collection of these gates can simply yield efficient circuits. However, this is not as straightforward as it might appear at the first glance because the delay of the same gate may disproportionally be increased under a temperature rise due to the major role that $OPCs$ can play. Fig 3 analyzes, as an example, the impact of a temperature rise (from 25°C to 85°C) on the FA_X1 gate and INV_X4 gate under different output load capacitances (determined by the number of gates are being driven) and different input signal slews (determined by the output transition of previous gate). Unlike FA_X1 in which the impact of a temperature rise on delay is almost independent of $OPCs$, INV_X4 exhibits a strong dependency. It is noteworthy that gates' delay is additionally subject to the rise and fall of their inputs. Thus, estimating accurately the impact of temperature on an entire circuit is more complex than it might be assumed.
(c) Guardband Estimation based on Delay Sensors: State-of-the-art approaches (e.g., [11], [12]) often propose to employ a delay sensor such as a Ring Oscillator (RO) to determine the required guardband through predicting the delay increase in the CP based on the RO's behavior. However due to the previous points (a and b), the ability of such RO-based estimations in protecting against temperature effects becomes questionable because the RO and CP may disproportionally be influenced by the same temperature rise. We investigate

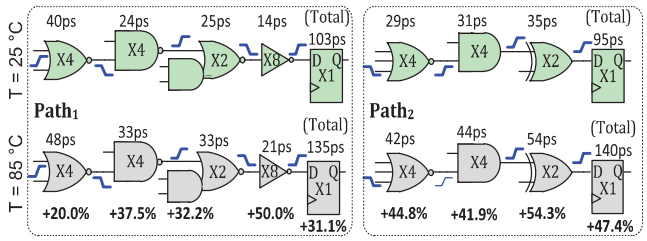


Fig. 2. The impact of temperature on circuit's paths is disproportional. Thus, the CP may change while the temperature rises/decreases at operation time.

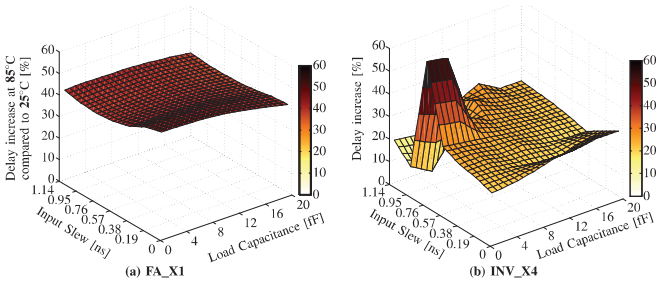


Fig. 3. Input signal slew and output load capacitance of a gate play a major role in determining the impact of temperature rise in INV_X4 unlike FA_X1.

in Fig. 4 whether the RO can accurately predict the delay increase of a CP when the temperature increases or not. As shown, when the temperature rises from e.g., 50°C to 85°C, a typical RO consisting of 15 stages [11] will predict a delay increase of 20.6%. However, the corresponding delay increase in two representative paths is different (26% and 32% in path$_1$ and path$_2$, respectively). Such noticeable underestimation leads to employing wrong temperature guardbands and thus timing errors. This will be evaluated further in Section V. **In Summary**, determining accurately the required temperature guardbands under varied temperature rises is elaborate as it entails investigating how gates/cells, which form the circuit, will jointly increase the overall delay of every individual path within the circuit at each particular temperature rise.

IV. Our Thermal-Aware Circuit Design

We create in this work thermal-aware cell libraries to employ them within existing EDA tools in order to leverage their mature algorithms. Our goals are a) performing thermal-aware logic synthesis to optimize circuits against temperature effects and b) performing thermal-aware timing analysis to accurately estimate the impact of temperature on circuits regardless of its complexity. In fact, the non-existence of such cell libraries (at varied fine-grained temperature steps) enforces designers to either interpolate between the two temperature corners (i.e. 25°C and 125°C) available in a standard cell library or employ a RO-based delay sensor to predict the impact of temperature rises. Both of the aforementioned alternative methods will evaluated in Section V to investigate whether we really need to create thermal-aware cell libraries or not.

A. Thermal-Aware Cell Libraries

Standard cell libraries provide the EDA flow tools (e.g., synthesis and static timing analysis tools) with the accurate delay and leakage/dynamic power information for every cell/gate. Because the $OPCs$ play a major role in determining the

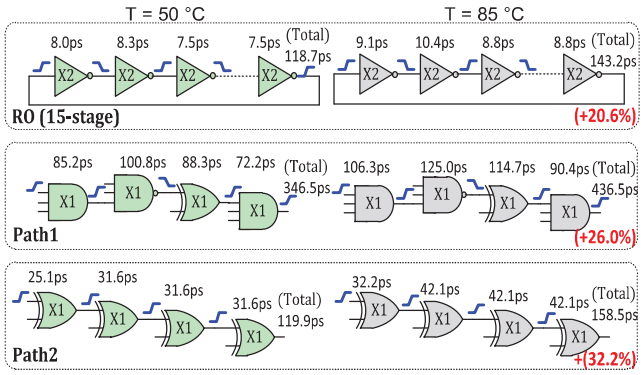


Fig. 4. Evaluation of how an RO-based delay sensor may underestimate the impact that a temperature rise may actually have on the CP's delay.

delay of gates/cells, any standard cell library must contain this information for every gate/cell under a range of input signal slews along with a range of output load capacitances. This is a prerequisite for any synthesis tool to optimize the circuit's netlist in which the most suitable gate/cell is always being selected according the existing $OPCs$. In addition, the detailed delay information within a cell library is also a prerequisite for the static timing analysis in which the overall delay of a circuit can be precisely estimated based on the exact delay of every individual gate/cell within the netlist.

A cell library is often created for three corners, namely: *typical* (i.e. normal V_{dd} and normal T), *fast* (i.e. high V_{dd} and low T) and *slow* (i.e. low V_{dd} and high T) [13]. Some cell libraries may additionally provide another corner of (normal V_{dd} and T_{worst}) like the 28nm Synopsys. Nevertheless, these limited temperature cases do not enable designers to tackle thermal concerns within the standard design flow. For instance, T_{worst} depends on the end-user scenario, i.e. while 125°C might be an appropriate T_{worst} for high-end CPUs, mobile CPUs might not have T_{worst} higher than 65°C. In addition, as the worst-case temperature itself is subject to the cooling/package, designers need to investigate the required guardband of their circuits under different worst-case temperatures towards identifying trade-offs between performance and cost. *Therefore, to allow design-space exploration with respect to temperature effects, the impact of temperature needs to be automatically captured within the standard design flow.*

To achieve that, we create a set of cell libraries under different temperatures $[T_{min}, T_{worst}]$ with the granularity of T_{step}. For each created each cell library, SPICE netlists of a variety of combinational and sequential gates/cells are simulated in an automatic process under a specific temperature. It is noteworthy that we employ SPICE netlists containing the accurate parasitics information (i.e. resistors and capacitors) obtained from gates/cells' layout based on the NanGate 45 nm open cell library [13]. For every gate/cell, we measure using HSPICE its rise/fall delay, output slope (transition), and dynamic power under a range of N input signal slews $[S_{min}, S_{max}]$ along with a range of M output load capacitances $[C_{min}, C_{max}]$. This is essential to properly consider the role of $OPCs$. In addition, the leakage power is also measured to take the correlation between temperature and leakage power into account. Then, all the measured values are stored within a $N \times M$ tables based on the standard "*liberty*" format which is typically

used in commercial libraries. This what makes our created thermal-aware cell libraries compatible with existing EDA tool flows. Note that the required time to create our thermal-aware cell libraries is spent only once (offline) and the performed simulations can be easily parallelized to speed up.

Implementation Details: To model the pMOS and nMOS transistors, we employ the high-performance 45 nm Predictive Technology Model (PTM) [14]. However, our work is not limited to a specific technology node and we selected the 45 nm to be compatible to the post-silicon 45 nm SPICE netlist of gates [13] which we employ. To model the electrical characteristics of MOSFETs under the impact that temperature has, we employ the BSIM compact modeling from Berkeley [15]. We consider 7 input slews ranging from 0.005 ns to 0.867 ns along with 7 load capacitances ranging from 0.5 fF to 20 fF. It is noteworthy that considering $OPCs$ of 7×7 is typical in both academic and commercial standard cell libraries. We select the temperature range to be [25°C to 125°C] which should be sufficient for most commercial applications. We choose a T_{step} of 1°C as a smaller step would not be helpful because the accuracy of thermal diodes needed for in our guardbands adaptation is typically > 1°C (details in Section IV-C)

B. Thermal-Aware Logic Synthesis

Designers add a guardband on top of the maximum delay of a circuit to maintain a reliable operation when temperature rises. Such a temperature guardband corresponds to the caused timing slack at T_{worst}. Hence, if the temperature becomes higher at operation time than what it was considered at design time (i.e. during synthesis) the guardband will compensate the induced delay increase. Thus, the circuit will be always clocked with a sustainable frequency ($f_{T_{worst}}$).

In this work, we propose to bring thermal awareness to the logic synthesis tool. Based on the targeted worst-case temperature (T_{worst}) the designer selects (from our created thermal-aware cell libraries in Section IV-A) the cell library that corresponds to that particular T_{worst} and provides it to the synthesis tool allowing for optimizing the circuit's netlist against T_{worst}. Thus, a better performance (i.e. higher $f_{T_{worst}}$) can be achieved and the required guardband will inherently included. This is because that a temperature rise does not influence all gates/cells proportionally as Fig. 1 demonstrated. Hence, there is a prospect that the mature algorithms of the synthesis tool to select the most suitable set of gates/cells that exhibit a smaller delay at that particular T_{worst}. This leads to circuits that operate at faster $f_{T_{worst}}$ compared to the traditional thermal-unaware synthesis. The complexity of our thermal-aware synthesis is the same as in the original synthesis because the cell library size is not increasing. Therefore, neither the required time nor the efficiency of the algorithms of the synthesis tool will be affected when any of our thermal-aware cell libraries is employed instead of the original one. T_{worst} is the maximum allowed on-chip temperature is determined depending on the end-user scenario (e.g., mobile/server CPU, thermal management policy, cooling, etc.). Thus, T_{worst} may vary from a scenario to another. Hence, creating thermal-aware cell libraries at various temperatures is prerequisite.

C. Thermal-Aware Timing Analysis

As a matter of fact, when the temperature at operation time becomes lower than T_{worst}, there is the prospect of indeed shrinking the employed guardband (i.e. temporarily *boosting* the frequency) without causing any timing errors. Therefore, we propose to estimate the exact circuit's delay at different temperatures to build a look-up table (LUT) that links each temperature step with the corresponding small, yet sufficient guardband that is actually needed. We first synthesize the circuit at T_{min}. Then, we perform a timing analysis for the obtained netlist using our thermal-aware cell libraries (see Section IV-A) for the range of [T_{min}, T_{worst}]. Our proposed thermal-aware timing analysis allows for accurately estimating the circuit's delay without concerning whether the CP may change or not at each different temperature step (see Section III(a) and Fig. 2) because the static timing analysis tool is designed to overcome that. Note that the synthesis is performed only once and then we repeatedly do thermal-aware timing analysis. The latter is quite fast and hence building the LUT can be done in the order of minutes.

Proposed Adaptive Guardband Optimization: Once the required LUT is built, it can be employed at operation time for guardband adaptation (i.e. boosting frequency whenever $T < T_{worst}$) and thus avoiding the employment of a conservative guardband that corresponds to T_{worst} throughout. We periodically read the current temperature from a thermal diode which is often available in chips to capture its maximum temperature. Then, we select from the LUT the corresponding frequency. Note that the rate of our adaptation depends on the reading's rate of thermal diode (i.e. the sensing granularity Δt_{read}). For a safe operation, it is crucial (when adapting the guardband) to add on top of the current temperature reading the temporal thermal gradient ΔT_{grd} (i.e. the maximum possible increase in temperature until the next reading of the thermal diode comes). Otherwise, timing errors might occur due to the fast pace of temperature rise compared to the rate of adaptation. ΔT_{grd} is mainly subject to the cooling/package and one can determine it at design time based on an offline analysis of the worst-case activities in which the maximum temperature rise per Δt_{read} can be captured. Algorithm 1 summarizes our proposed technique. Note that the reverse temperature dependency does not influence our technique as it appears at lower voltage than one we target (1.2V) [2].

Algorithm 1 Our Adaptive Guardband Optimization (Hardware)

Require: Δt_{read}, ΔT_{grd}, *thermal diode, our thermal-aware LUT*
1: **for** every Δt_{read} **do**
2: **Read** $T_{current}$ ▷ temperature monitoring
3: $T_{safe} \leftarrow T_{current} + \Delta T_{grd}$ ▷ safety margin adding
4: **Get** $guardband$ at T_{save} ▷ LUT selection
5: **Set** frequency f_{clock} ▷ frequency adaptation
6: **end for**

V. Evaluation and Comparison

We evaluate our work with respect to three criteria: a) *static optimization* in which the effectiveness of thermal-aware synthesis (see Section IV-B) is investigated, b) *adaptive optimization* in which the effectiveness of guardband adaptation (see Section IV-C) is investigated and c) *necessity of thermal-aware timing analysis* in which we investigate if other alternative methods like the interpolation between two temperature concerns and the employment of a *RO*-based delay sensor are able to accurately estimate temperature guardbands.

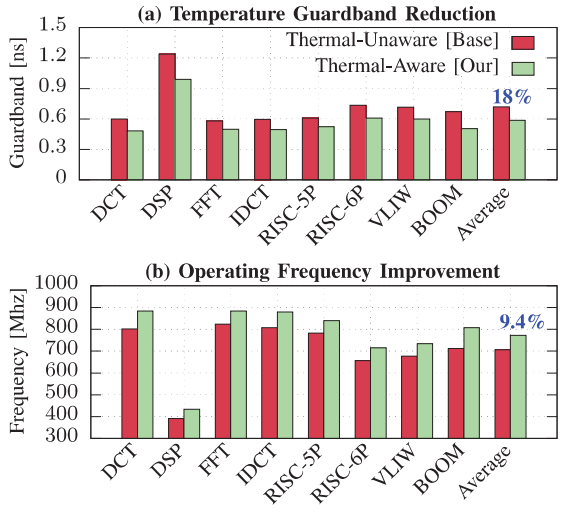


Fig. 5. Evaluating the effectiveness of our ***static*** guardband optimization.

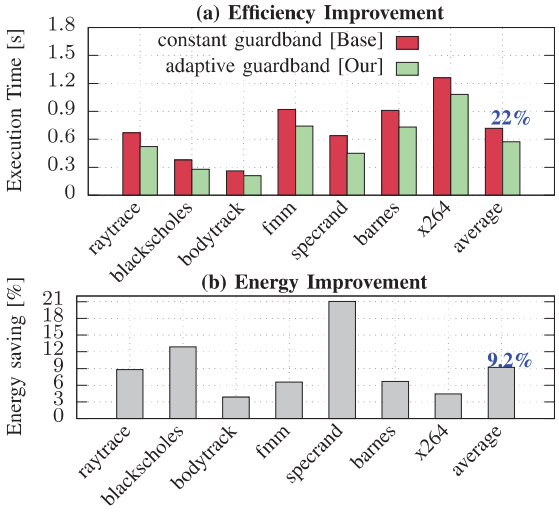


Fig. 6. Evaluating the effectiveness of our ***dynamic*** guardband optimization.

Benchmark Circuits: We employed the Synopsys Processor Designer tool to generate RTL for 5 processors: VLIW, RISC (with 5 and 6-pipeline stages), FFT and DSP. We also employed DCT and IDCT circuits often used in image processing. Additionally, we also employed the Berkeley Out-of-Order Machine (BOOM) [16]. In our evaluations, we employ the Design Compiler Synthesis and the Static Timing Analysis tools from Synopsys. To achieve the highest optimization efforts, we used the "compile_ultra" while synthesizing circuits.

(a) Static Guardband Optimization: As explained in Section IV-B, providing the synthesis tool with the thermal-aware cell library at the worst-case temperature (e.g., 125°C) enables its optimization algorithms to consider the thermal effects and hence generate more resilient circuits that exhibit better performance. Figs. 5 (a and b) present direct comparisons between our thermal-aware synthesis and the traditional thermal-unaware synthesis with respect to the achieved reduction in guardbands and the corresponding achieved increase in frequencies, respectively. For the case of thermal-unaware synthesis, the cell library at 25°C was used in synthesis and then the guardband that corresponds to the incurred time slack when the temperature reaches 125°C was calculated. As shown, our optimized circuits always exhibit smaller temperature guardbands and the reduction reaches on average 18% leading to 9.4% improvement in the frequency and up to 13.3% in the case of the BOOM processor. Our area analysis showed that the optimized circuits come with merely 2% overhead.

(b) Adaptive Guardband Optimization: To evaluate the effectiveness of our adaptive guardband technique compared to employing throughout a constant guardband that corresponds to T_{worst}, we performed our proposed thermal-aware timing analysis for the BOOM processor to build the required LUT as explained in Section IV-C. Then, we built a toolchain as follows: The instruction-set simulator (gem5 [17]) for an Out-of-Order processor is used to extract the activities of running applications. Then, the run-time on-chip temperatures are estimated based on the Hotspot thermal simulator [18] in conjunction with the McPAT power simulator [19]. The feedback loop between the temperature and leakage has also been considered. Finally, we implemented our Algorithm 1 to adaptively tune the operating frequency based on the maximum on-chip temperature at every $\Delta T_{read} = 1$ms. The ΔT_{grd} has been estimated to be < 2°C/ms after analyzing the temperature transient profiles of all studied applications. To investigate diverse activities and hence thermal behaviors, we employed different applications from the PARSEC [20] and SPEC2006 [21] benchmark suites. We considered a T_{worst} of 85°C similar to Intel Xeon. Note that the possible feedback loop caused by increasing power/temperature when a smaller guardband is employed (akin to the higher frequency) is implemented and considered within our evaluations.

As shown in Fig. 6(a), our adaptive guardband optimization results, on average, in 22% less execution time due to boosting the frequency whenever the temperature $T < T_{worst}$. Because faster frequency leads to a higher dynamic power consumption, it is necessary to also evaluate the energy of each application before and after implementing our technique. As shown Fig. 6(b), our technique also provides energy saving which reaches 9.2% on average due to the shorter execution time despite the slightly-higher dynamic powers due to the higher frequency. For a fair evaluation, we also analyzed the average temperature traces (obtained before and after employing our technique) to quantify the potential temperature increase caused by the higher frequencies. Our analysis showed an increase of merely 3°C on average. We also evaluated the impact of such a small temperature increase on long-term reliability w.r.t to aging using our physics-based aging model which considers both BTI and HCI mechanisms jointly [22], [23]. Our adaptive optimization has a negligible impact on reliability. The 3°C temperature increase leads to a threshold voltage shift in pMOS by just $\sim$0.7mV on average.

(c) Necessity of Thermal-Aware Timing Analysis: To investigate the necessity of creating thermal-aware cell libraries and employing them to perform thermal-aware timing analysis (as our focus), we first explore whether a linear interpolation between two temperature corners (25°C and 125°C) would be sufficient. To achieve that, we compare the gates delays at 65°C (estimated by the linear interpolation) with the actual gates delay measured by HSPICE as explained in Section IV-A. Fig. 7 reports that the error in estimations is considerable $\in$[-6, 10]%. In fact, such errors results in under- or over-estimating the required temperature guardband which

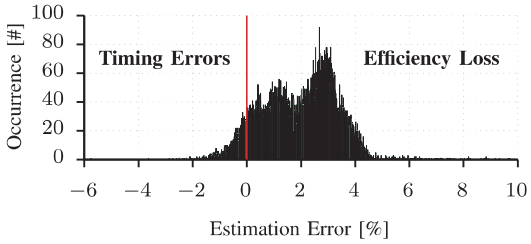
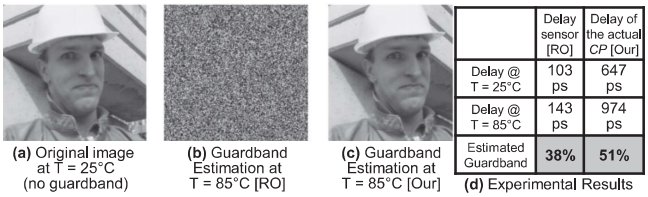


Fig. 7. Distribution of errors when estimating gates delay at 65°C based on the interpolation between two temperature corners (25°C and 125°C).



	Delay sensor [RO]	Delay of the actual *CP* [Our]
Delay @ T = 25°C	103 ps	647 ps
Delay @ T = 85°C	143 ps	974 ps
Estimated Guardband	38%	51%

(d) Experimental Results

Fig. 8. Impact of guardband estimation on the output image of DCT-IDCT circuits. [*RO*]: Estimating the guardband through an *RO*-based delay sensor consisting of 15-stage [11]. [Our]: Estimating the guardband through our thermal-aware timing analysis presented in Section IV-C.

leads to either timing errors or efficiency loss, respectively.

We additionally evaluate the ability of a typical (15-stage) *RO*-based delay sensor [11] to correctly estimate the required temperature guardbands. To achieve that, we study DCT-IDCT image processing chain to explore how the output image (obtained from gate-level Modelsim simulations) will be affected. Fig. 8(a) shows the output image when the circuit is clocked with the maximum frequency obtained during synthesis (i.e. no guardband). Then, we show the resulting image after adding a guardband – that corresponds to a temperature rise from 25°C to 85°C – estimated first using a *RO*-based delay sensor (see Fig. 8(b)) and then using our thermal-aware analysis (see Fig. 8(c)). The analysis shows that if the *RO* is employed to predict the required guardband of the DCT-IDCT, the guardband will be noticeably underestimated (see Fig. 8(d)). This is because the *RO* does not accurately represent the impact that such a temperature rise has on the actual CP as Fig. 4 earlier motivated. In turn, such an underestimation can have a severe impact of the image quality (see Fig. 8(b)) as a non-sustainable clock will be provided to the DCT-IDCT circuits resulting in timing errors in abundant paths. On the other hand, the image quality remains the same as at $T = 25°C$ when our thermal-aware timing analysis is employed to estimate the required guardband (see Figs. 8(a, c)) due to the correct provided clock. To explore whether our technique overestimates the guardband and thereby the image's quality is sustained, we slightly reduced our estimated guardband by just 0.01ns and repeated the DCT-IDC simulations. We observed that such a tiny reduction directly leads to a noisy output image which ensures that our obtained guardband was as small as required.

VI. Conclusion

We introduced *thermal-aware cell libraries* to bring thermal awareness to existing commercial EDA tools. Hence, designers can automatically use them together with existing tool flows to address thermal concerns. Our libraries are publicly available allowing other researchers to employ them without requiring any modifications. Based on our libraries, we proposed *static* and *adaptive* optimization techniques for temperature guardbands. We demonstrated how the efficiency of circuits can be increased through employing small, yet sufficient guardbands at operation time rather than employing a conservative guardband throughout (which is the consequence of the non-existence of thermal-aware cell libraries).

Acknowledgments

This work is supported in parts by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" (SPP 1500 - spp1500.itec.kit.edu). Authors would like thank Andreas Gerstlauer and Ku He from University of Texas at Austin for sharing DCT/IDCT circuits.

References

[1] "Thermal-Aware Cell Libraries, V1.0," http://ces.itec.kit.edu/dependable-hardware.php
[2] D. Wolpert and P. Ampadu, "Temperature effects in semiconductors," in *Managing Temperature Effects in Nanoscale Adaptive Systems*, 2012, pp. 15–33.
[3] J. Keane and C. H. Kim, "An odometer for cpus," *IEEE Spectrum*, vol. 48, no. 5, pp. 28–33, 2011.
[4] T. Chantem, X. S. Hu, and R. P. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on mpsocs," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 10, pp. 1884–1897, 2011.
[5] Z. P. Gu, Y. Yang, J. Wang *et al.*, "TAPHS: thermal-aware unified physical-level and high-level synthesis," in *ASPDAC*, 2006, pp. 879–885.
[6] W.-L. Hung, Y. Xie, N. Vijaykrishnan *et al.*, "Thermal-aware floorplanning using genetic algorithms," in *ISQED*, 2005, pp. 634–639.
[7] M. Bao, A. Andrei, P. Eles *et al.*, "On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration," in *DAC*, 2009, pp. 490–495.
[8] J. Tschanz, N. S. Kim, S. Dighe *et al.*, "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," in *ISSCC*, 2007, pp. 292–604.
[9] D. Ernst, N. S. Kim, S. Das *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, 2003, pp. 7–18.
[10] H. Amrouch, B. Khaleghi, A. Gerstlauerz *et al.*, "Reliability-aware design to suppress aging," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, p. 12.
[11] G.-Y. Wei and M. Horowitz, "A fully digital, energy-efficient, adaptive power-supply regulator," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 4, pp. 520–528, 1999.
[12] C. R. Lefurgy, A. J. Drake, M. S. Floyd *et al.*, "Active Management of Timing Guardband to Save Energy in POWER7," in *MICRO*, 2011.
[13] "Nangate, Open Cell Library," http://www.nangate.com/.
[14] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *Electron Devices, IEEE Transactions on*, vol. 53, no. 11, pp. 2816–2823, 2006.
[15] Y. Chauhan, S. Venugopalan, M. Karim *et al.*, "BSIM - Industry standard compact MOSFET models," in *ESSCIRC, Proceedings of the*, 2012.
[16] C. Celio, D. A. Patterson, and K. Asanovi, "The Berkeley Out-of-Order Machine (BOOM): An Industry-Competitive, Synthesizable, Parameterized RISC-V Processor," Berkeley, Tech. Rep., Jun 2015.
[17] N. Binkert, B. Beckmann, G. Black *et al.*, "The Gem5 simulator," *SIGARCH Comput. Archit. News*, 2011.
[18] M. R. Stan, K. Skadron, M. Barcella *et al.*, "Hotspot: a dynamic compact thermal model at the processorarchitecture level," *Microelectronics Journal*, 2003.
[19] S. Li, J. H. Ahn, R. D. Strong *et al.*, "The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. Archit. Code Optim.*, 2013.
[20] C. Bienia, S. Kumar, J. P. Singh *et al.*, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th PACT*, 2008, pp. 72–81.
[21] J. L. Henning, "SPEC cpu2006 benchmark descriptions," *SIGARCH Comput. Archit. News*, 2006.
[22] H. Amrouch, V. van Santen, T. Ebi *et al.*, "Towards interdependencies of aging mechanisms," in *Computer-Aided Design (ICCAD), IEEE/ACM International Conference on*, 2014, pp. 478–485.
[23] H. Amrouch, V. M. van Santen, and J. Henkel, "Interdependencies of degradation effects and their impact on computing," *IEEE Design & Test Magazine*, 2016.