Self-Timed Thermally-Aware Circuits

David Fang, Filipp Akopyan, Rajit Manohar*
Computer Systems Laboratory
Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853, U.S.A.

Abstract

Thermal management is becoming increasingly important in circuit designs with high power density. Circuits that overheat beyond specified operating conditions may suffer timing failures, or become damaged for various reasons, including thermal runaway. Traditional power management in synchronous systems often involves transitions to different system states or modes, typically involving changes in clock frequencies or voltage levels. However, the self-timed nature of asynchronous circuits allows delays to vary continuously during operation, enabling stall-free performance-throttling. We present a novel application of a thermally sensitive circuit to automatically regulate the performance and power consumption of asynchronous circuits, with minimal implementation overhead.

1 Introduction

As the power density of modern integrated circuits continues to increase with shrinking feature size, power and temperature management become increasingly important and challenging [9]. Thermal profiling, along side analog simulation, is crucial to designing large and power-hungry circuits [3, 4]. Static thermal profile information can be used to place circuits on a die to maximize temperature uniformity, thereby reducing the peak temperature [12]. Likewise, dynamic temperature profiling can be used to direct operation, e.g. halting the system when the circuit is too hot, or switching to a lower power mode, as is common in many modern clocked processors [1, 2, 8]. In this paper we demonstrate a mechanism that regulates the performance of asynchronous circuits using a simple thermally-sensitive circuit. Our approach does not require temperature measurement, rather, we leverage the temperature response of subthreshold devices to construct a signal for controlling the speed of other circuits. In our work, we introduce explicit temperature sensitivity to amplify the dynamic range of a temperature-controlled delay element. A properly designed delay element can then be used to adjust the frequency of a local circuit.

The circuit we present is most easily applicable to asynchronous (or self-timed) circuits. Asynchronous circuits operate without any global clock, and use handshakes to move and communicate data. The data-driven nature of asynchronous circuits allows a circuit to idle with no switching activity when there is no work to be done. Asynchronous circuits are capable of operating correctly in the presence of continuous and dynamic changes in delays [6]. Sources of delay variation may include temperature, supply voltage, manufacturing, noise, radiation and other transient phenomena. The timing-robustness of asynchronous circuits facilitates the use of a thermally-sensitive delay element to automatically continuously modulate the speed of selected circuits without interrupting operation, thus imposing negligible implementation overhead. This approach can also be used in a GALS design, by local frequency scaling in a clock domain using a temperature-sensitive frequency synthesizer.

We discuss the basic operating principles of our thermally-sensitive circuit (Section 2), evaluate some simulations of examples where the circuit is applied (Section 3), and conclude our findings (Section 4).

2 Circuits

The frequency of an asynchronous pipeline is determined by the gate delays on the critical path rather than an external frequency source. As the circuit heats up, the gate delays increase and the frequency naturally drops, thereby reducing the circuit's dynamic power consumption and self-heat generation. However, the natural negative-feedback retardation of this self-heating rate is too weak to halt the increase in temperature [11]. By introducing an explicit temperature-sensitive delay element, we can design an asynchronous circuit that is thermally self-limiting, one that prevents self-overheating by reducing its own frequency sufficiently.

*E-mail: {fang,filipp,rajit}@csl.cornell.edu

2.1 Thermal Dependence

The operation of transistors in the subthreshold region is more sensitive to temperature than the above-threshold region. In the subthreshold region, the source-drain current is exponentially dependent on temperature [7]:

$$I_D = I_0 \cdot \exp\left(\frac{V_{gs} \cdot q}{\zeta \cdot k \cdot T}\right) \qquad (1)$$

where I_D is the drain-source current, I_0 depends on channel width, channel length, diffusion constant of carriers, carrier density and electron charge [13], ζ is a nonideality factor (greater than 1), and T is temperature in Kelvin.

We use the high temperature-sensitivity of the subthreshold transistors to construct a temperature-sensitive voltage source, shown boxed in Figure 1. The basic principle of operation is that transistors M1 and M2 are biased differently to have contrasting thermal sensitivities. For this paper, we have chosen M1 to operate in deep subthreshold (more temperature-sensitive), while M2 operates near-threshold. With different sensitivities, the two transistors will form a resistive voltage divider, producing a temperature-sensitive output voltage. The bias voltages and the sizes for transistors M1 and M2 are chosen to achieve the desired temperature response for a given technology. The bias voltages can be generated off-chip for post-fabrication tuning and runtime controlling. We found that this simple circuit was sufficient to meet our design goals. One generalization this circuit is to use different temperature-sensitive structures to produce arbitrary temperature-delay characteristics.

Figure 1 shows how the thermally-sensitive voltage source can be used to control the gate of a foot transistor in logic circuitry. The foot transistor will operate anywhere between fully-on and mostly-off to modulate the speed of the pull-down transition. As temperature increases, the foot transistor will conduct less current, until the point where it will just be off, halting the circuit. When the foot transistor is off, the pull-up half of the conditional staticizer will weakly retain the value of V_{int}. We are mostly interested in modeling the effective delay of the modified circuit as a function of temperature, for the digital-thermal simulation used in Section 3.

What follows is an example of the circuit with parameters tuned for a particular temperature response.

2.2 Design Example

We have simulated the above temperature-dependent circuit using TSMC 0.18 μm technology parameters, with 1.8 V nominal supply V_{dd}. In our example, we targeted a temperature-performance response that drops sharply above 100°C, boiling point. The following design parameters yielded the V_g-T response shown in Figure 2: for subthreshold M1, W = 5 μm, L = 400 nm; for the weak current source M2, W = 600 nm, L = 5 μm; the size of the foot transistor in the simulation is W = 720 nm, L = 360 nm. $V_{bias,n}$ on M1 is 200 mV, and $V_{bias,p}$ on M2 is 1.3 V. Figure 2 shows the V_g switching sharply near 100°C, and practically turning off the foot transistor by 120°C.

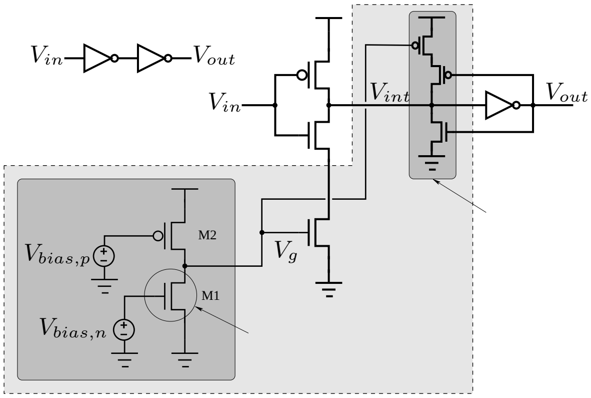

Figure 1. A temperature-sensitive delay element consisting of an inverter-pair modified with additional foot transistor controlled by a thermally-sensitive voltage source

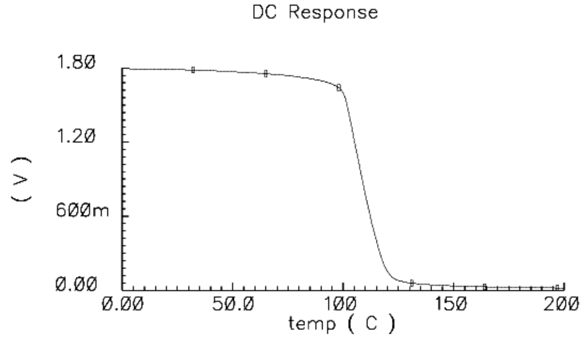

Figure 2. Foot Transistor V$_g$-T Dependence

In a simplified setup, where the drain of the foot transistor is connected directly to a V_{dd} voltage source, we also measure the current through the foot transistor. The results of the current measurement are shown in Figure 3. In the domain where the foot transistor is on, the initial slope (up to 90°C) is due to temperature-dependent properties in the above-threshold saturated region: the mobility of electrons/holes and threshold voltage V_{th} shifts [5]. Beyond 95°C, the conductivity drops sharply due to the transition in V_g. Figure 4 shows a close-up of the foot transistor current response above 90°C on a log-scale. Between 110 and 120°C, the current decays exponentially until a floor current is reached.

We also measure the delay through the modified circuit as a function of temperature in Figure 5, on a linear scale, and in Figure 6, zoomed in on log-scale. The delay is

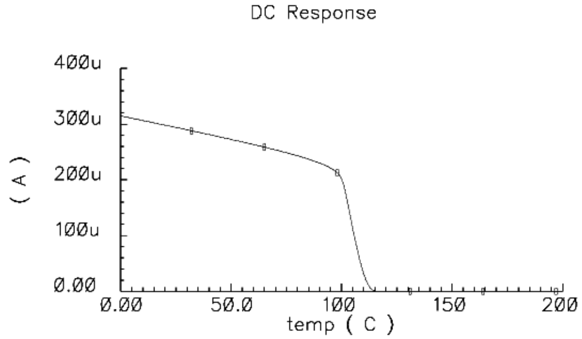

Figure 3. Foot Transistor I-T Dependence

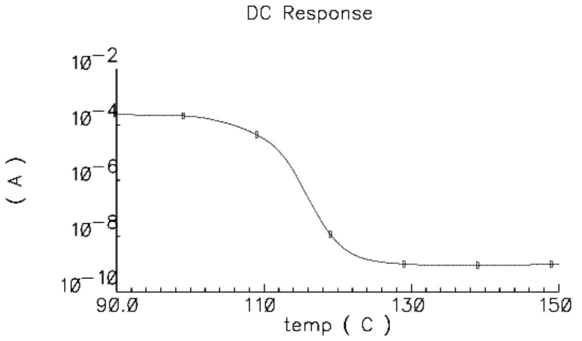

Figure 4. Foot Transistor I-T Zoomed In

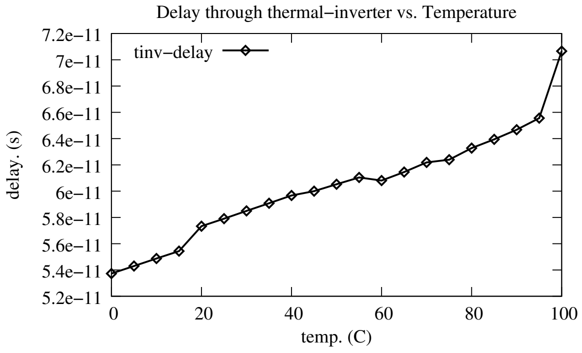

Figure 5. Delay through modified inverter

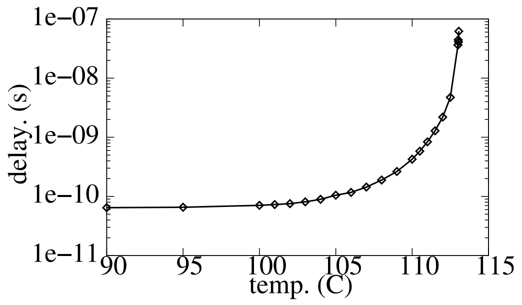

Figure 6. Delay through modified inverter

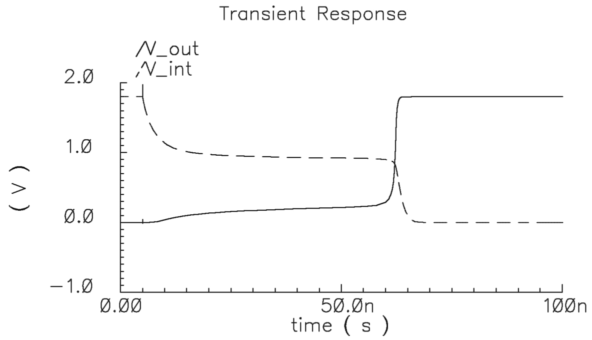

Figure 7. Waveforms of thermally-sensitive inverter signals at 113°C

approximately linear up to around 95°C. Figure 7 shows the waveform of cross-coupled nodes V_{int} and V_{out} from Figure 1 switching at 113°C. The internal V_{int} is slowly pulled to the switching threshold while being opposed by the staticizer. After 60 ns, a long delay, V_{out} finally switches cleanly, and stabilizes V_{int} through weak positive feedback. The intermediate logic level of V_{int} is internal to this subcircuit and never used elsewhere. The output V_{out} is cleanly digital in all situations, only ever delayed arbitrarily, which is acceptable to asynchronous circuits.

Above 113°C, V_{out} never switches because the current through the pull-down logic is insufficient to overpower the pull-up staticizer, so the delay is infinite. This means the circuit will naturally halt above a critical temperature, and resume once the temperature drops sufficiently. We approximate the delay-derating factor with a piecewise-continuous function of temperature in our digital-thermal simulator (Section 3).

The static power consumption of the temperature-dependent voltage is due to the current through the subthreshold, weakly-sized transistors M1 and M2. Figure 8 plots the current through transistors M1 and M2 as a function of temperature; using the sizes and parameters we have chosen in our example, the peak current is on the order of tens of nA, comparable to leakage current for this technology. Thus, the power consumption of the thermally-sensitive voltage source is negligible compared to the power in rest of the system.

2.3 Delay Model

For normal (not thermally-sensitive) transistors, we use the following model for delay-derating as a function of voltage and temperature (here denoted as θ) [5]. The derating function can be factored into three components: a voltage-dependent factor f, a temperature-dependent factor g, and a mixed threshold voltage term h.

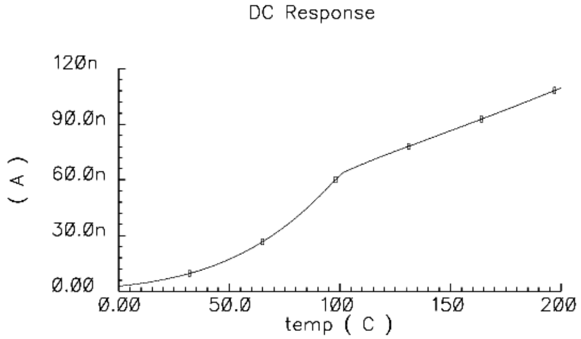

Figure 8. Subthreshold Transistor I-T Relation

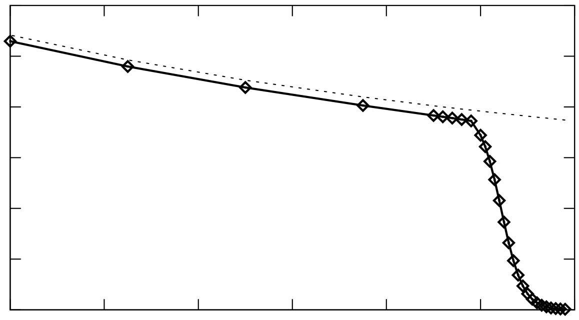

Figure 9. Frequency of normal (dashed) and thermally-sensitive (solid) ring oscillator vs. circuit temperature

$$\frac{d(V_{dd},\theta)}{d_{nom}} = f(V_{dd}) \cdot g(\theta) \cdot h(V_{dd},\theta) \quad (2)$$

$$f(V_{dd}) = \left(\frac{V_{dd}}{V_{dd,0}}\right)^{1-\alpha} \quad (3)$$

$$g(\theta) = \left(\frac{\theta}{\theta_0}\right)^{\theta_k} \quad (4)$$

$$h(V_{dd},\theta) = \left[\frac{1-\frac{V_{T0}(\theta_0)}{V_{dd,0}}}{1-\frac{V_{T0}(\theta_0)}{V_{dd}}+\frac{\sigma\Delta\theta}{V_{dd}}}\right]^{\alpha} \quad (5)$$

Table 1 lists the parameters used for our simulations, using a TSMC .18 μm technology. Some parameters have separate values for NFETs and PFETs, denoted by additional subscripts N and P. The coefficients were determined through empirically fitting to `hspice` data.

Table 1. Simulation parameters

parameter, symbol	value
nom. supply voltage, $V_{dd,0}$	1.8 V
nom. threshold voltage, V_{T0}	0.4 V
nom. temperature, θ_0	300 K
velocity saturation index, α_N, α_P	1.5, 1.7
threshold coefficient, σ_N, σ_P	2e-3, 1.5e-3 V/K
temp. mobility index, $\theta_{k,N}$, $\theta_{k,P}$	1.65, 1.65

3 Evaluation

We present the results of simulations or circuits that use our proposed thermally-sensitive transistors in this section. As shown below, in each situation we take the original digital circuit and then replace a few selected inverters with temperature-sensitive delay elements. The result is a circuit that regulates its speed based on the local temperature with minimal hardware overhead. To demonstrate the effectiveness of our application of thermally-sensitive circuits we present examples of varying complexity.

The simulator. The simulator we used for our experiments is an event-driven digital simulator, extended to capture the transient effects of temperature and supply voltage on delay. The input to the simulator is a sized netlist including event rules describing the logic. Event rules are tagged with a voltage domain (all the same in these cases), a thermal region corresponding to physical placement, and a flag for thermal-sensitive performance response. Delays and capacitances are based on the logical effort model and calibrated against a TSMC .18 μm technology (Section 2.3). The simulator accounts for gate and parasitic output capacitances in computing switching energy and delay, but does not account for internal capacitances in transistor stacks. Digital simulation is coupled with a finite-element thermal simulator, where switching circuits inject heat into their respectively mapped thermal regions. For the results in this paper, we modeled our system as a silicon die mounted on an aluminum heat sink in contact with constant temperature air. Our simple simulator has the advantage of the fast digital simulation with the added realism of transient thermal effects.

3.1 Ring-oscillator

The simplest circuit we show is a 31-stage ring oscillator with one stage modified to a thermally-sensitive inverter. The frequencies are plotted against temperature in Figure 9. As expected, the frequency quickly degrades past around 100°C, beyond which the delay grows exponentially. Before 95°C, the difference between the curves shows what performance overhead one might expect from adding a thermally-sensitive inverter to a critical path.

3.2 FPGA

We simulated a 5x5 asynchronous FPGA running a function-block-intensive benchmark to demonstrate a circuit transiently reaching a self-heating equilibrium. The design of our particular FPGA is described in [10]. We modified the original design by adding thermally-sensitive inverters to the handshake acknowledges coming from the input buffers of each logic block, a total of four modified inverters per FPGA tile[1]. Since the original FPGA design operated at such low power, we artificially amplified the switching energy by a factor of 100 in this simulation to emulate self-overheating.

Table 2. Normalized throughput of thermal-aware FPGA at different temperatures

Temperature °C	normalized throughput
25	1.00
45	0.85
80	0.70
89	0.64
94	0.60
97	0.49
100	0.27

The asynchronous FPGA benchmark is a finely-pipelined feed-forward computation and, thus, its performance is expected to be limited by the slowest handshake in the forward path. Since we expect the critical cycle to be in the hottest thermal region, we report the normalized throughput against the peak surface temperature in Table 2. The thermally-aware FPGA's average surface temperature stabilizes at around 100°C after 1 ms of simulated time at an operating frequency of 27% of the room-temperature throughput. In the same scenario, the same FPGA without our thermal-aware modifications continues to heat itself to destructive temperatures.

This FPGA example demonstrates an application where global performance is determined by the hottest spots on the die surface. The data-driven nature of asynchronous designs makes it extremely easy to thermally regulate the performance of the entire system by modifying very few points with thermally-sensitive circuits.

3.3 Dynamic Resource Scheduling

Our final example demonstrates how our thermally-sensitive circuits can be used to dynamically schedule activity away from hot-spots. Chip dies naturally have non-uniform thermal signatures depending on physical design and dynamic operation characteristics. While synchronous circuits may benefit from a more uniform thermal profile, asynchronous circuits have an additional benefit where high performance may be sustained by scheduling work to cooler units. The entire system need not slow down on account of a single hot-spot.

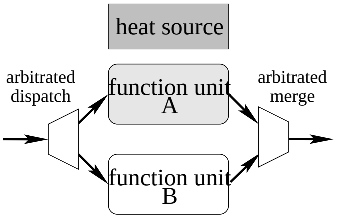

Figure 10. A thermally-sensitive dispatch

We set up a asynchronous circuit arranged as shown in Figure 10, where an arbitrated dispatcher can forward data to one of two twin function units. Function unit 'A' abuts a heat source, mimicking the heat signature from power-hungry circuits. Unit 'B' is placed further away from the heat source. Each function unit is modified with one additional thermally-sensitive inverter generating the input acknowledge that indicates which unit is ready to receive and process data. The arbiter in the dispatcher chooses between these acknowledges[2]. This represents a simple mechanism by which an asynchronous dispatch unit might schedule instructions in a dynamically scheduled asynchronous microprocessor. The dispatch circuit would, under normal circumstances, simply schedule an equal number of instructions to the two function units, thereby increasing execution throughput.

Table 3. Arbitrations of thermally-sensitive function units over a fixed window of time

Temp.-A	A cycles	Temp.-B	B cycles
40	46	40	46
60	43	42	44
80	35	45	36
90	31	49	32
95	28	57	34
98	24	64	36
99	19	76	38
99.8	14	83.5	29
100	12	45	49
101	8	45	48
102	5	45	49
103	4	46	48
104	2	46	48

[1] In practice, all four modified sites could even share the same thermally-sensitive voltage source from Figure 1.

[2] The simulated arbiter is *fair*; no input will remain ready and unserviced for more than one iteration of the other input.

One simulation run used a heat source with negative-thermal-feedback, so the temperature never exceeded 100°C. Table 3 summarizes the correlation between function unit temperatures (Temp. A and B) and the corresponding number of arbitrated iterations (A and B cycles) for a fixed-size window of time. As unit A approaches the threshold temperature, its acknowledges to the arbiter arrive less frequently. In steady-state (at 99.8°C), the arbitration ratio is approximately 2:1.

A second run used a heat source without thermal-feedback to demonstrate the effect at even higher temperatures, shown in the bottom part of Table 3. Beyond 100°C, unit A practically stops operating, yielding almost all of the computation work to unit B. Once unit A cools down, it will begin to request data from the dispatcher more frequently.

We emphasize that the scheme we presented uses no direct temperature-sensing in arbitration, and is thus, very simple and efficient. We do not need any explicit temperature monitoring and scheduling logic to be incorporated into the dispatcher, as might be required if conventional temperature sensors were used to control the scheduling [1, 2, 8]. Proper application of a thermally-sensitive circuit can achieve a simple form temperature-aware resource-scheduling with minimal modifications to an existing asynchronous circuit.

4 Conclusion

In this paper, we have presented a simple thermally-sensitive circuit that can be used to regulate gate delays in digital circuits. We began by characterizing the temperature response of the circuit, using one particular set of bias voltages and transistor sizes. We constructed a temperature-sensitive voltage source that can be used to throttle the speed of selected logic gates. We showed that the output of a thermally-sensitive delay element is suitable for digital asynchronous circuits [6]. We have demonstrated a few applications of the thermally-sensitive transistors in self-regulating the performance of asynchronous circuits in temperature-critical situations.

Perhaps the greatest asset of the circuits presented is the simplicity of the technique. With only local and minimal modifications to an existing asynchronous circuit, a design can be made thermally-aware and slow itself down without interruption of operation to prevent self-overheating. An asynchronous circuit may even halt and resume operation correctly at an arbitrarily later time later with no additional control circuitry, which makes performance-regulating thermal circuits trivial to apply. The timing robustness alone mitigates the need for additional control circuitry and re-design effort. A small number of strategically placed thermally-sensitive gates is sufficient to regulate the global operating performance across an entire asynchronous system. By significantly slowing down the operation of circuits in hot-spots, we have also shown that the same circuits may be used to dynamically steer work to cooler units, thereby localizing performance regulation and achieving better temperature uniformity.

References

[1] Intel(R) Pentium(R) 4 processor in the 423-pin package at 1.30, 1.40, 1.50, 1.60, 1.70, 1.80, 1.90 and 2 GHz datasheet.

[2] David Brooks and Margaret Martonosi. Dynamic thermal management for high-performance microprocessors. In *HPCA '01: Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, page 171, Washington, DC, USA, 2001. IEEE Computer Society.

[3] Yi-Kan Cheng and Sung-Mo Kang. A temperature-aware simulation environment for reliable ULSI chip design. *IEEE TCAD*, 19(8):1211–1220, October 2000.

[4] Yi-Kan Cheng, Prasun Raha, Chin-Chi Teng, Elyse Rosebaum, and Sung-Mo Kang. ILLIADS-T: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of CMOS VLSI chips. *IEEE TCAD*, 17(8):668–681, August 1998.

[5] J. M. Daga, E. Ottaviano, and D. Auvergne. Temperature effect on delay for low voltage applications. In *Proc. DATE*, pages 680–685, 1998.

[6] Alain J. Martin. The limitations to delay-insensitivity in asynchronous circuits. In William J. Dally, editor, *Proc. ARVLSI*, pages 263–278. Massachusetts Institute of Technology, 1990.

[7] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. Tata McGraw-Hill, 2004.

[8] H. Sanchez, B. Kuttanna, T. Olson, M. Alexander, G. Gerosa, R. Philip, and J. Alvarez. Thermal management system for high performance PowerPC(TM) microprocessors. In *COMPCON '97: Proceedings of the 42nd IEEE International Computer Conference*, page 325, Washington, DC, USA, 1997. IEEE Computer Society.

[9] Dennis Sylvester and Himanshu Kaul. Future performance challenges in nanometer design. In *Proc. DAC*, pages 3–8, New York, NY, USA, 2001. ACM Press.

[10] J. Teifel and R. Manohar. Highly pipelined asynchronous FPGAs. In *Proc. FPGA*, February 2004.

[11] J. A. Tierno. *An Energy-Complexity model for VLSI computations*. PhD thesis, California Institute of Technology, 1995.

[12] Ching-Han Tsai and Sung-Mo Kang. Cell-level placement for improving substrate thermal distribution. *IEEE TCAD*, 19(2):253–266, February 2000.

[13] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 2005.