Closed-Loop Modeling of Power and Temperature Profiles of FPGAs

Kanupriya Gulati, Sunil P. Khatri and Peng Li
Department of Electrical and Computer Engineering,
Texas A&M University, College Station 77843

ABSTRACT

In recent times, the contribution of leakage power to the total power consumption of a chip has been increasing at an alarming rate. Leakage power is expected to exceed dynamic power in newer process technologies. Since leakage exhibits an exponential increase with temperature, it is possible that the high leakage of an IC causes a temperature increase, which in turn causes an increase in leakage, and so on, until the IC fails due to overheating. At the very least, this may cause the temperature and power consumption of the IC to be poorly estimated by traditional thermal or power modeling techniques. In this paper, we develop a framework to model this situation in an FPGA context. Our CAD framework accurately models the total power consumption of the design at a given temperature, finds the thermal profile of the IC under this power consumption, and then uses this new thermal information to update the power consumption. This is iterated until the temperature of the IC converges, or until the temperatures on the die exceed a safe value. The iterations are very fast, due to the use of accurate and compact mathematical macromodels for leakage and temperature computation in the inner loop. We have exhaustively verified the fidelity of all our leakage macromodels. They estimate the leakage, at any temperature, to within 3% of the values generated by SPICE, while providing greater than four orders of magnitude speedup over explicit SPICE runs. Our experiments show that this model helps avoid an incorrect estimation of chip temperature and total power consumption, and also helps detect the increase in device temperature beyond a safe value. The average (maximum) error of our temperature estimates has been found to be within 1% (2.5%) compared to a full-chip 3D temperature modeling tool.

1. INTRODUCTION

In recent times, the popularity of field programmable gate arrays [31] (FPGAs) as a digital circuit implementation methodology has been increasing. The main reason for this increase in popularity is the low cost associated with the FPGA approach for low-to-mid volume designs, and also the faster turn around time associated with FPGA based designs. Also, other factors like the increasing logic density and speed of recent FPGAs has contributed to this trend. It is conjectured that the reduction in the number of ASIC starts in recent years [27] (due to the increasing cost of generating IC fabrication masks), has also contributed significantly to the growing popularity of FPGAs.

However, there are potential problems that plague the FPGA based design approach. Due to shrinking feature sizes, power consumption and density of FPGAs broadly follow the trends of custom ASICs. Due to their reconfigurable nature, FPGAs are typically not the most optimal hardware implementation approach from a speed or power perspective. Additionally, in comparison to ASICs, FPGAs use more transistors and hence consume more power for a similar design. Given the trend that power is becoming a zeroth order design consideration in modern VLSI design, it becomes vital to study the power consumption trends of the FPGA.

The planning of the power budget for an FPGA design involves considering three types of power: startup power, dynamic power, and sub-threshold power. Upon startup, as Vdd ramps up to its final voltage, the unknown state of SRAM cells in an SRAM-based FPGA can cause a current spike known as *inrush current*. Also, SRAM based FPGAs draw current during the configuration process, as the routing and look-up table (LUT) configurations are read from memory into the FPGA device. The power consumption due to this current is referred to as the startup power. Ongoing research efforts like [26, 34] have addressed this component of the power. Our work does not address startup power. Once the FPGA begins operation after being configured, the other two kinds of power consumption – dynamic and static – are of relevance. Dynamic power for a CMOS circuit is linearly proportional to the switched capacitance, operating frequency and switching activity, while it is quadratically proportional to the supply voltage. The last component of FPGA power consumption is the static power. This power is due to leakage currents in the MOSFETs, and is predominantly comprised of the sub-threshold and gate leakage components.

The gate leakage values are temperature independent, while the sub-threshold leakage values have an exponential dependence on temperature. This is explained by the sub-threshold leakage equation [23, 33]:

$$I_{ds}^{sub} = \frac{W}{L} I_{D0} e^{\frac{V_{gs}-V_T-V_{off}}{nv_t}} [1 - e^{-\frac{V_{ds}}{v_t}}]$$

In the above equation, W and L are the device width and length. Also, I_{D0} is a constant while $v_t = \frac{kT}{q}$. Here k is the Boltzmann constant, and $v_t = 26mV$ at room temperature. n is the sub-threshold swing parameter (a constant). Finally, V_{off} is a constant, typically equal to -0.08V.

As the above equation indicates, with diminishing feature sizes and supply voltages, the sub-threshold leakage increases exponentially. Also, as the gate count on an FPGA goes up, the leakage increases linearly. Further, sub-threshold leakage is exponentially dependent on the junction temperature. With increased leakage, the associated increase in power dissipation results in more heat being generated, which in turn raises the junction temperatures even further, resulting in a yet further increase in leakage current. This could cause a non-convergence of the thermal and power estimates for the FPGA, a condition referred to as *thermal run-away*. Even if convergence is achieved, the temperature ↔ power inter-dependence, if ignored, could potentially result in a situation where the overheating of the FPGA occurs. We refer to this condition as *thermal breakdown*. Our work allows both these phenomena to be modeled.

This paper reports a CAD methodology to model the power consumption of an FPGA which is configured with a given design, compute the thermal profile of the FPGA under this power consumption, and use the new temperature information to update the (sub-threshold) leakage power consumption. This loop is iterated until convergence, or until the temperature of the FPGA rises past a critical level (taken to be 110°C for this work).

The key contributions of this work are:

- We propose an accurate method to efficiently and accurately estimate the FPGA temperature and power (dynamic and sub-threshold leakage) profile by accounting for the tight inter-dependence of temperature and leakage power. Accounting for temperature accurately is crucial since a higher temperature not only shortens the device life but also impacts timing and package design. The average (maximum) error in our temperature estimates is 1% (2.5%) compared to a full-chip 3D thermal modeling tool [15].

- Our thermal and leakage power estimates are self consistent, since the tight inter-dependence between temperature and leakage power is iterated to convergence.

- Our methodology computes leakage estimates and temperature profiles using efficient pre-computed models, allowing each iteration to be performed in a fast and accurate manner. This is the main reason for extremely low cumulative runtimes across iterations. This is achieved by:

  - Leakage values are pre-characterized for the different blocks in the FPGA slice at 3 different temperatures, by using SPICE [22]. By curve-fitting this data, we derive a compact macro-model for the leakage of the FPGA slice as a function of temperature. This macromodel is used to estimate the leakage during any iteration of our algorithm, hence avoiding the need for running SPICE during any iteration of our algorithm. Our leakage macromodels have been exhaustively verified against SPICE. Our macromodels estimate the leakage, at any temperature, to within 3% of the values generated by SPICE (over all circuit components in the FPGA - LUTs, MUXes, DFFs, SRAMs and INVs), while providing greater than four orders of magnitude speedup over explicit SPICE runs.
  - The IC is discretized into $n \times n$ grids. The dependence of the temperature of any grid point on the IC as a function of the power at any other grid point is pre-characterized, and used during the iterations. This again avoids the need for running a full-blown thermal analysis during any iteration of our algorithm. The effect of heat dissipation due to heat sinks is included in this model.

- Our methodology is design and device[1] specific. The algorithm reported in this paper can be used in an FPGA design flow before FPGA configuration, to ensure that excessive temperatures are not encountered in the design.

The remainder of this paper is organized as follows: Some previous work in this area is described in Section 2. Section 3 details our approach for performing full-chip thermal and power modeling for the FPGA. In Section 4 we present some results from experiments which were conducted in order to validate our approach. We conclude in Section 5.

2. PREVIOUS WORK

Past work in the area of simultaneous power and thermal modeling and estimation for FPGAs is limited, although there are several efforts for power modeling and estimation. Power macro-modeling for FPGAs has been studied in [9, 6, 25]. In [28, 3, 13, 12], only dynamic power consumption has been modeled and studied for FPGAs. This paper, in contrast, studies both sub-threshold power and dynamic power consumption, and their effect on circuit temperature (including the interdependence of temperature and sub-threshold power). In [32], the authors perform the leakage analysis of a 90nm FPGA device. This analysis is performed only at two fixed temperatures (25°C and 85°C).

In [30], the authors attempt to characterize particular FPGA components with respect to temperature. Their aim is to study which component of the FPGA has the highest contribution to the temperature increase. On the modeling front, tools like Hotspot [29] and Hs3d [17] are used to perform an architectural level steady state temperature estimation.

A temperature-dependent leakage power model at the *micro-architectural level* was proposed in [16, 10]. However, their approach only uses a typical implementation for certain circuits (logic and memory based) in order to develop the relation between temperature and leakage current. The focus is micro-architectural design, in an ASIC implementation context. No references for the circuits considered in this exercise are provided. The constants in the leakage-temperature model can vary vastly across circuit designs and architectures. As a result, computing the temperature dependent leakage model by using an average over a few 'typical' circuits may result in significant temperature and power estimation errors. Further, they estimate the maximum leakage

[1] The results presented in this paper pertain to the Xilinx XC4VLX200 device. Running the algorithm for alternate FPGA devices is simply a matter of redoing the thermal and power pre-characterization steps.

current for a design by using a genetic algorithm. Our approach, in contrast, is a *design and device specific, circuit level approach for computing accurate power and temperature profiles for an FPGA*. Taking advantage of the structural regularity of an FPGA, we pre-characterize the basic circuit blocks in an FPGA slice, and use curve fitting to develop an accurate temperature dependent leakage macromodel for each of these circuit blocks in the FPGA slice. The authors of [16] report a 6-15% error in their leakage model when compared with SPICE. Our approach generates leakage values over all circuit components (LUTs, MUXes, DFFs, SRAMs and INVs) which are within 3% of those generated from SPICE, over all temperatures (27°C- 110°C). Our approach is design dependent, and thus computes leakage values specific to the configured circuit implemented in the FPGA. Also, the thermal modeling described in [16] is performed in two modes i) the individual mode (wherein different components of the processor are considered separately and it is assumed that there is no horizontal heat transfer across components) and ii) the universal mode (where the entire processor is considered as a single component). Our thermal modeling employs the technique of [14], wherein the circuit is discretized in both the horizontal (where the chip surface is discretized into $n \times n$ grid regions) and vertical (where the chip height is discretized into a total of k=15 layers for metal, dielectric and heat sink layers) directions. The heat transfer mechanism in both the horizontal and vertical directions are considered. A fast full-chip thermal simulator [14] is used to pre-compute the temperature increase in any of the grid regions with respect to a unit power consumption in each of the grid regions on the die. Our thermal model is therefore, more fine-grained and accurate in comparison to the thermal model used in [16]. In general, we can use any thermal macromodeling tool, such as [11], as long as its interface is similar to [14]. The thermal modeling tool used in [10] is Hotspot [29]. Yet another major difference between our approach and the approaches described in [16, 10] is that our approach aims at finding the consistent (converged) leakage and temperature values, after considering the interdependence of temperature and leakage power. In our approach, the iterations for computing leakage and temperature are continued until the change in temperature between successive iterations is less than a user-specified tolerance value ϵ, or until the temperature of the FPGA exceeds a safe value (resulting in our terminating the iterations and declaring that thermal breakdown has occurred). The approaches of [16, 10] do not aim at closing the temperature-leakage loop.

Thermal issues in FPGAs are relatively under-explored. There are research works which discuss the placement of temperature sensors on the FPGA [19], or using the frequency of a ring oscillator for estimating die temperature [18, 4]. None of the above works considers or models the strong interdependence between the sub-threshold leakage and chip temperature. This is imperative in modeling the thermal run-away and thermal breakdown conditions, and also to achieve a more accurate model of FPGA temperature profiles. The work of [20] discusses the relationship between power estimation and thermal budgeting for FPGAs. However the analysis is at a coarse-grained level, whereas our approach uses the specific LUT configuration information to perform the simultaneous modeling of leakage and temperature profiles in a design and device specific manner. Further, our approach computes the thermal profiles in a fine-grained manner, across different regions of the FPGA die, unlike the work of [20].

3. OUR APPROACH

The goal of this paper is to compute an accurate estimate of the temperature and power profile of an FPGA, by iterating the computation of total power consumption (including the temperature dependent sub-threshold power and the interconnect component of dynamic power) and the temperature profile computation until convergence. It is important to note that in case convergence does not occur, we have a thermal run-away condition, which could result in the destruction of the FPGA device. Even without such a run-away condition, it is possible that the converged temperature values are large enough to affect the reliability of the FPGA device or thermal breakdown. Our algorithm models both these conditions. Our modeling is specific to the FPGA device under consideration, as well as the binary that has been configured on to the FPGA. The inner loop of our computation is extremely efficient on account of the use of accurate and compact precomputed models for the power and thermal behavior of the IC.

The top-level flow of our approach is shown in Figure 1. The different components of this flow are:

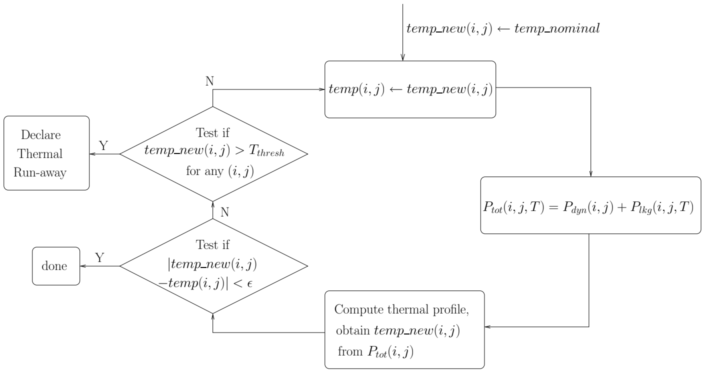

$temp\_new(i,j) \leftarrow temp\_nominal$
$temp(i,j) \leftarrow temp\_new(i,j)$
N
Declare Thermal Run-away
Y
Test if $temp\_new(i,j) > T_{thresh}$ for any (i,j)
N
$P_{tot}(i,j,T) = P_{dyn}(i,j) + P_{lkg}(i,j,T)$
done
Y
Test if $|temp\_new(i,j) - temp(i,j)| < \epsilon$
Compute thermal profile, obtain $temp\_new(i,j)$ from $P_{tot}(i,j)$

Figure 1: Flowchart of our Approach

- Given the current temperature, we estimate the total power consumption of the FPGA, which consists of dynamic power (largely temperature independent) and leakage power. Note that the interconnect component of dynamic power is modeled in our flow. Subthreshold leakage has an exponential dependence on temperature, so it is recomputed for each change in temperature, and hence we obtain the updated total power consumption for every iteration. This computation is done using a precomputed temperature-dependent leakage macromodel for each circuit block in a slice of the FPGA. The chip is discretized into an $n \times n$ grid for power computations (as well as thermal computations).

- Using the updated total power consumption, we compute the thermal profile over the FPGA die, after discretizing the die area into several rectangular regions.

This is done using the approach of [14], which precomputes an $n \times n$ matrix which can be used to determine the temperature of any of the n^2 grid regions, given the power of all the n^2 regions on the die. *This thermal model includes heat dissipation due to heat sinks as well.*

- If the maximum difference across all grid points of the newly computed thermal profile and the previously computed thermal profile is smaller than a user-specified quantity ϵ, we stop. Otherwise more iterations are performed.

- If the temperature anywhere on the FPGA die exceeds a threshold T_{thresh}, we declare non-convergence, and stop the iterations. We use a $T_{thresh} = 110°$C for our experiments. In case this occurs, we state that thermal breakdown has occurred.

In the remainder of this section, each of the above components of the flow are discussed in detail, after a brief discussion about the circuit model utilized in the rest of this paper.

3.1 Circuit Model Details

We first begin with a discussion of the circuit design for the FPGA. Our implemented design flow is targeted towards a Xilinx Virtex-4 XC4VLX200 FPGA device, which is one of the larger commercially available FPGA devices implemented in a 90nm process technology. Note that the XC4VLX200 die is rectangular and hence each of our $n \times n$ grids are rectangular regions. Our flow can be easily retargeted towards other FPGA devices, if their implementation details (such as process, slice logic, heat sink details etc.) are given. The circuit diagram for our FPGA logic block is shown in Figure 2. The Look-up Table (LUT) in this FPGA logic block is shown separately in Figure 3. It consists of a 16:1 MUX circuit, implemented using NMOS passgates. This is the typical circuit used for implementing LUTs [21, 8]. The circuit for the 16 SRAM configuration bits (labeled as ”S” in Figure 3 is shown in Figure 4. The DFF of Figure 2 is implemented using identical master and slave latches, each of which has a NMOS passgate connected to the clock, and a pair of inverters in a feedback configuration to implement the storage element. The feedback inverter uses long-channel devices. This figure is not shown for brevity.

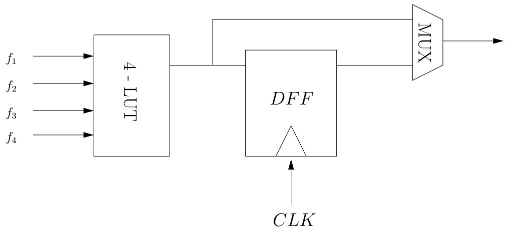


Figure 2: Logic Block in the FPGA

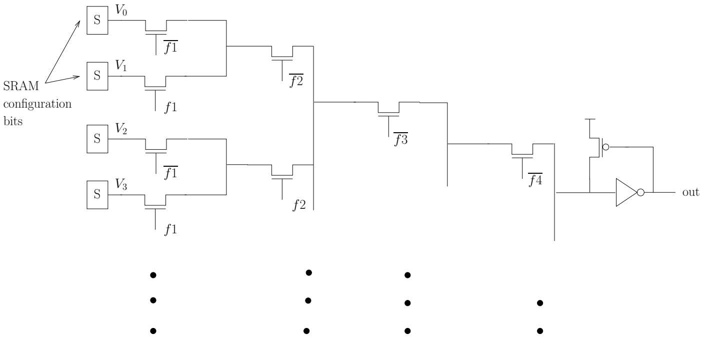


Figure 3: LUT implementation using a 16:1 MUX

3.2 Total Power Consumption

The first step of our algorithm requires the computation of the total power consumed by the FPGA design. This power number is calculated specifically for the design the FPGA is configured with. We compute the dynamic and static power components separately. Note that the static power (in particular, the sub-threshold leakage portion of the static power) is computed in a temperature-aware fashion. The computation of each of these components of the total power is described next.

We consider the entire FPGA to be partitioned into an $n \times n$ grid of rectangular regions for both power and temperature calculations. In particular we select $n = 16$, which was found to provide a good tradeoff between runtime and accuracy. For each of the 256 rectangular regions, we compute the different components of power, which are used in the subsequent step of temperature profile determination.

3.2.1 Dynamic Power Consumption

The dynamic power consumption of the configured FPGA design is computed using the XPower tool from Xilinx [1]. After synthesizing, placing and routing the design using the Xilinx ISE 8.2i tool suite, we compute the maximum operating frequency of the design f_{ckt}. The XPower tool reads the design data (the NCD file), and computes the activity estimate for the design. Using the node and wire capacitances, XPower computes the dynamic power of the circuit as

$P_{dyn} = C \cdot Vdd^2 \cdot f_{ckt} \cdot \alpha$

where α is the computed activity estimate for the design.

Note that this power is computed by the XPower tool as an aggregate number for the entire design. XPower also *includes the power consumption of the interconnects* in this aggregate number. In our approach, we partition this power into each of the n^2 grid regions of the FPGA based on the logic in the different LUTs in each region. Suppose there are

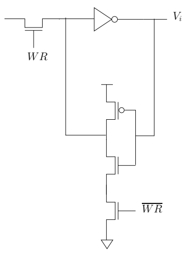


Figure 4: SRAM Configuration Bit Design

k LUTs per region[2]. For each of the k LUTs, we compute the probability of the output (see Figure 3) being a logic 1 as

$P_1 = (\Sigma V_i)/16$

Here V_i $(0 \leq i \leq 15)$ are the logic values stored in the 16 SRAM configuration bits of the LUT. Note that this allows us to account for the configuration data of each LUT, while estimating the dynamic power of the LUT. Now we compute the probability of switching of the LUT output as $P_{switch} = 2 \cdot P_1 \cdot (1 - P_1)$. By averaging the P_{switch} values for all k LUTs in any region, we find the average probability of switching for the region. For the grid region (i, j), let us call this quantity $P(i, j)$.

Now we distribute the total dynamic power P_{dyn} of the design into the n^2 regions, by assigning region (i, j) the dynamic power based on the following equation:

$P_{dyn}(i, j) = P_{dyn} \cdot \frac{P(i,j)}{\Sigma(P(i,j))}$

In this manner, we compute the dynamic power for each of the n^2 regions of the FPGA. Such a distribution of the *dynamic power also accounts for the interconnect power* distribution, since the higher the probability of switching of the LUT outputs in a grid region, the higher will be the power consumption by the interconnect. Note that the dynamic power is independent of temperature.

Note that XPower does compute quiescent power as well. The reason we do not use the XPower's quiescent power computation (but rather compute this on our own, as explained in the subsequent sections), is that we would like to compute the static power per region, based on the temperature of that region. This ability is not present in the XPower tool.

3.2.2 Leakage Power Consumption

We compute the leakage power of our design using a compact temperature-dependent leakage macromodel, which is derived using precomputed SPICE [22] runs. For SPICE simulations, we used a 90nm BPTM [5] process card.

We first discuss how the leakage of any LUT is computed. Recall that we know the values of each SRAM configuration bit of each LUT in the design. Using this information, we note that each of the 31 NMOS passgates shown in Figure 3 are in one of 3 states $(L_1, L_2$ or $L_3)$ shown in Table 1. In each of these states, the gate terminal is at 0V. Since an NMOS passgate passes a Vdd value with a signal degradation of V_T volts, the 3 states L_1, L_2 or L_3 represent the possible states of any leaking NMOS passgate in Figure 3. A special case is the state $L_2^{'}$, shown in Figure 5. In this case, the top passgate $N1$ is ON, but is connected to 2 leaking passgates ($N2$ and $N3$). This results in a lowering of the voltage at the common node of the 3 passgates. This leakage configuration for passgate $N3$ is referred to as $L_2^{'}$. There can be other hybrid leakage states like $L_2^{'}$ as well, but our experiments indicate that the error in our leakage estimate arising from neglecting such states is very low.

Our approach accounts for gate leakage as well. Gate leakage [24] increases with increasing V_{gs}, but decreases with increasing V_{ds}. Also, gate leakage has a weak temperature dependence [35].

For the 4 L-states listed above, the leakage is dominated by sub-threshold leakage (which has an exponential dependence on temperature). The gate leakage in these L-states is negligible in comparison, due to the low V_{gs} value in the L-states.

Gate leakage becomes consequential in the states listed in Table 2. In this table, the gate terminal is at Vdd (except for state K_4, in which the gate is connected to 0V). The states in Table 2 are referred to as K-states. These states have no sub-threshold leakage contribution (including state K_4, since $V_{ds} = 0$V for this state).

We pre-compute the leakage values of the L-states and the K-states by using SPICE. The gate leakage values are temperature independent, while the sub-threshold leakage values have an exponential dependence on temperature.

The temperature dependence of the sub-threshold leakage is accounted for by computing the leakage values of the L-states at 3 different temperatures, and fitting an exponential to the resulting values. The result is a temperature-dependent leakage macromodel for all L-states. This resulted in a very high fidelity estimate for the temperature dependence of the sub-threshold leakage at an arbitrary temperature and fixed input values, with a maximum error of 1% (compared to SPICE).

The leakage estimate for any LUT is computed by first traversing the LUT netlist, and determining the state of each NMOS passgate (from the 8 states above). Now the temperature-dependent leakage values of all the devices with L-states are computed using the temperature dependent leakage macromodel, and added up. The result is added to the leakage values of all the devices with K-states, as well as the leakage of the output inverter and keeper device (shown in Figure 3), to yield the leakage of the LUT. Note that the output inverter/keeper can be in one of two states, whose temperature-dependent leakage is pre-characterized and expressed in a separate macromodel. This macromodel is used to compute the temperature-dependent leakage of the output inverter and keeper during any iteration.

We also computed the leakage of the SRAM configuration bits of Figure 4, the D-flipflop and MUX of Figure 2. After computing 3 values of the leakage of these circuits at 3 different temperatures, we fitted an exponential to these leakage values, thus again constructing a compact temperature-dependent leakage macromodel for these circuits, which is used to compute the leakage of these circuits at any temperature.

The values of the LUT configuration bits (V_i in Figure 3) are known to us, and used in leakage estimation. However, the values of the 4 inputs of the LUT (f_1, f_2, f_3 and f_4 in Figure 2 and 3) are not known to us. Hence, we compute the leakage of the LUT to be the average value of leakage, averaged over all 16 combinations of f_1, f_2, f_3 and f_4. From these numbers of leakage power for each logic block, we compute the total leakage for the region (i, j) at its current temperature $temp(i, j)$. The total leakage power for region (i, j) at any temperature T is denoted by $P_{lkg}(i, j, T)$. Finally, the total power for the region (i, j) at any temperature T is computed as $P_{tot}(i, j, T) = P_{dyn}(i, j) + P_{lkg}(i, j, T)$.

Note that value of the leakage of the LUT, as computed above, was also compared to that computed by SPICE by performing exhaustive SPICE simulations over all possible values of the SRAM configuration bits and the values of f_1, f_2, f_3 and f_4. The results of this exercise showed that the maximum error in our leakage computation methodology was below 3%, while providing a speedup of greater than

[2] k is computed as N / n^2, where N is the total number of LUTs in the FPGA

Case	V_d	V_s	V_g
L_1	Vdd	0	0
L_2	$Vdd - V_T$	0	0
L_3	Vdd	$Vdd - V_T$	0
$L_2^{'}$	$(Vdd - V_T)^{'}$	0	0

Table 1: NMOS Passgate Sub-threshold Leakage States

Case	V_d	V_s	V_g
K_1	0	0	Vdd
K_2	Vdd	$Vdd - V_T$	Vdd
K_3	$Vdd - V_T$	$Vdd - V_T$	Vdd
K_4	$Vdd - V_T$	$Vdd - V_T$	0

Table 2: NMOS Passgate Gate Leakage States

four orders of magnitude when compared to explicit SPICE runs.

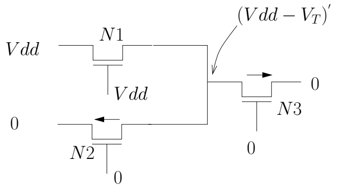


Figure 5: $L_2^{'}$ Leakage

3.3 FPGA Thermal Profile Computation

We use the approach of [14] to compute the thermal profile within the FPGA. Note that this pre-characterization results in n^2 power-to-temperature tables, each with n^2 values. The table Z_{ij} for the $(i,j)^{th}$ grid point indicates the contribution to the temperature of the $(i,j)^{th}$ grid point by a 1 Watt power consumption in each of the n^2 grid points. Since we know the total power consumption of each grid point, we can find the new temperature of the $(i,j)^{th}$ grid point, by superposition of the temperature contribution from all the n^2 grid points.

$T_{new}(i,j) = \Sigma_{k,l}(P_{tot}(k,l,T) \cdot Z_{ij}(k,l)$

where Z_{ij} is the power-to-temperature table for the $(i,j)^{th}$ grid point. Note that k and l are each iterated from 1 to 16.

To compute the power-to-temperature tables, we need to solve the steady-state heat conduction equation numerically, subject to proper boundary conditions [7]

$\nabla \cdot (k(\overrightarrow{r},t)\nabla T(\overrightarrow{r},t)) + g(\overrightarrow{r},t) = 0$

Here T is the temperature in degrees Celsius, $\overrightarrow{r}$ is the location in the 3D space, k is the thermal conductivity of the material in $W/m^2 \cdot^\circ C$, and g is the power density of the heat sources in W/m^3.

The circuit is discretized into $n \times n$ grid regions. This formulation computes how a power source applied in each discretized circuit region impacts the temperature distribution at the surface of the silicon substrate. The chip is also discretized in the vertical direction, resulting in a large discretized thermal analysis problem. For each circuit block, we distribute one unit of power uniformly within the block and adopt a fast full-chip thermal simulator to compute the resulting temperature increase in each bin at the silicon substrate surface.

We choose the value of n to be 16. This value of n was found to provide a good compromise between runtime (∼1 sec/iteration) and accuracy (∼1% average error compared to a full-chip 3D temperature modeling tool [15]). 15 layers of materials are modeled in the thermal simulation. For each metal/dielectric layer, a metallization percentage is assumed. Based on the percentage of metal, the thermal conductivities of metal and dielectric are averaged to produce an average thermal conductivity for each layer. This thermal model includes heat dissipation due to heat sinks as well. The output of the thermal profile generator is the temperature increase in each of the n^2 bins, as a result of applying power within each region. By applying superposition, we obtain the final temperature increase in each region (by adding the contribution to the temperature increase of the region, from each of the remaining regions). In this way, we obtain the new temperature $temp\_new(i,j)$ for each region, given $P_{tot}(i,j,T)$ and $temp(i,j)$ for each of these regions as input.

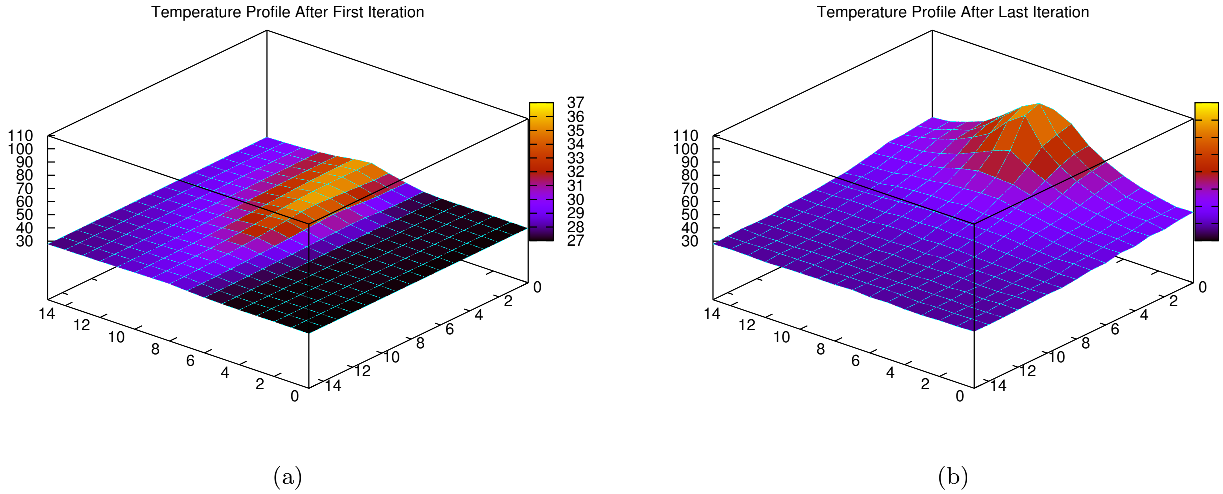
3.4 Endgame

We find the absolute difference of the updated temperature matrix $temp\_new(i,j)$ and the old temperature matrix $temp(i,j)$. We declare convergence when the maximum difference across all grid points is less than $0.001^\circ C$. In case convergence does not occur (the maximum value of $temp\_new(i,j)$ is greater than $T_{thresh} = 110^\circ C$), we declare that the FPGA has encountered thermal breakdown.

4. EXPERIMENTAL RESULTS AND COMPARISONS

We implemented our entire flow using a script written in the *perl* programming language. Initially, the FPGA is assumed to be operating at $27^\circ C$, with no power in any of the 16×16 regions of the FPGA. The design is synthesized, placed and routed in the Xilinx ISE 8.2i environment, and the maximum operating frequency f_{ckt} is determined. We next run XPower, with the input frequency assigned to f_{ckt}. The output database of the Xilinx ISE tool used to obtain the configuration information of each LUT, and the region it belongs to. Based on this information, we assign the dynamic power (obtained from XPower) to different regions as described in Section 3.2.1. The thermal modeling tool described in [14] is run up-front, and the pre-computed results are stored in a table. All the above steps are performed before any iterations are started. During the iterations, we compute the leakage power as described in Section 3.2.2. The total power is now used to compute the new temperature $temp\_new(i,j)$ for each region, as described in Section 3.3. The new temperature is again used to compute the modified power of each region (only the sub-threshold leakage portion of the power is modified due to the temperature change), and the process is repeated until convergence.

Table 3 describes the results we obtained for 10 designs. These designs are taken from the IWLS benchmarks suite [2]. In this table, the first column lists the example under consideration, while the second column lists the frequency of the design f_{ckt}. The third column reports the number of CLBs in this design. Column 4 lists the initial dynamic power. Column 5 reports the number of iterations required for convergence. The maximum percentage change in temperature, between the first and the final iteration, is reported in Column 6. The highest temperature across all grid regions, after the final iteration, is reported in Column 7. Columns



(a)

(b)

Figure 6: Our Temperature Profile for Circuit DMA Operating at 450 MHz.

Example	Freq	# CLB	DynPower (W)	# Itrs	Max % Temp Chg	High Temp ($^{\circ}C$)	Max % LkgPower Chg	Runtime (sec)
ac97_ctrl	300	1859	3.49	6	46.38	42.57	20.61	3
pci	160	1813	4.51	5	45.51	41.63	21.18	2
ethernet	290	3061	9.71	7	91.35	61.04	49.54	4
usb_funct	200	3057	5.37	5	65.09	48.06	32.81	2
wb_dma	350	939	13.16	8	165.97	98.73	121.95	4
RISC	200	6478	8.95	5	102.20	65.50	64.33	7
DMA	165	5548	6.43	6	76.90	54.29	44.49	7
s38584	210	3014	7.42	5	95.70	64.76	59.44	4
s38417	150	2488	3.74	5	43.49	40.43	20.82	4
s35932	320	3055	5.65	6	67.96	49.75	33.63	5

Table 3: Experimental Results for Our Approach

6 and 7 indicate that the temperature of the device can be incorrectly estimated if the tight inter-dependence of sub-threshold leakage and temperature is not considered. The maximum percentage change in the leakage power for any grid region (between the first and the final iteration) is listed in Column 8. The runtime for the *entire flow* is reported in the last column.

Table 4 reports similar data, computed under the assumption that the design is heavily pipelined, so as to allow operation at 450MHz, which is the specified frequency of the Virtex-4 XC4VLX200 FPGA.

Based on Tables 3 and 4, we note that without the simultaneous modeling of leakage and temperature, the estimates of temperature and leakage of the design can be vastly incorrect. This trend is observed across all the designs we studied. We observe a thermal breakdown condition for circuit *wb_dma* in Table 4. In other words, for this design, the temperature of the device increased beyond T_{thresh} ($110^{\circ}C$), causing the iterations to be terminated.

Figures 6 (a) and 6 (b) describe the thermal profile of the FPGA for the DMA design. The temperature profile is shown after the first iteration (Figure 6 (a)), as well as after the final iteration (Figure 6 (b)). We note the large inaccuracy that would be incurred in temperature estimation, if the simultaneous modeling of temperature and leakage were not conducted.

We verified the final temperature values calculated by our approach at each of the grid regions, against a full-chip 3D thermal modeling and simulation approach discussed in [15], for the DMA design. The inputs to the model of [15] were the power values (dynamic power plus leakage power for all grid regions) which were obtained from the final iteration in our approach, and the characteristics of the dielectric, metal and heat sink layers. We used the same metallization percentages, thermal conductivities, heat sink dimensions and materials, etc. for the full-chip 3D modeling tool as we had used for our method. We then carried out the full chip thermal analysis using [15]. The discretization was selected to be 64 × 64. The resultant 64 × 64 temperature values obtained from this analysis were averaged over 4 × 4 regions to obtain 16 × 16 temperature values. These were compared to the 16 × 16 temperature values obtained from our approach when we declared convergence (final temperature) for the DMA design. The maximum (average) error obtained was 2.52% (1.05%). Also, note that the maximum absolute difference between the temperature of any 4 × 4 region (obtained using [15], without averaging) and the temperature of the corresponding grid point (on the 16 × 16 grid computed using our approach) was $4.76^{\circ}C$. The temperature plot obtained from [15] for the DMA design is shown in Figure 7.

[15] is a highly efficient multigrid 3D fullchip thermal simulator with significant improvement in runtime and memory over several existing works. The time required by one run of [15] is heavily dependent on the discretization. With a

Example	# CLB	DynPower (W)	# Itrs	Max % Temp Chg	High Temp ($^{\circ}C$)	Max % LkgPower Chg	Runtime (sec)
ac97_ctrl	1859	5.16	6	65.79	49.83	31.80	3
pci	1813	10.91	6	101.55	62.20	51.45	2
ethernet	3061	13.48	7	119.47	74.16	72.02	4
usb_funct	3057	12.54	6	138.16	75.95	75.24	4
wb_dma	939	16.26	> 8	> 192.65	> 110 (*breakdown*)	> 154.94	4
RISC	6478	18.67	5	180.39	107.14	136.54	8
DMA	5548	16.15	7	162.90	95.22	111.99	7
s38584	3014	14.92	6	161.82	102.61	122.12	4
s38417	2488	10.70	7	114.87	65.03	56.18	6
s35932	3055	8.04	6	93.03	59.30	46.49	5

Table 4: Experimental Results for Our Approach (Circuits Operating at 450 MHz)

discretization of 64 × 64, [15] took ∼30 seconds for a single iteration. In contrast our approach takes ∼0.75 seconds.

To the best of our knowledge, no other work reports the converged temperature or power numbers for FPGA designs after closing the dependence loop between leakage and temperature. We therefore, validate for correctness in fullchip temperature calculation alone.

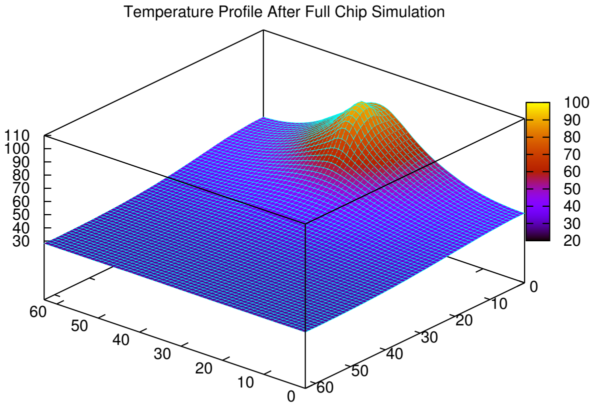


Figure 7: Temperature Profile for Circuit DMA Operating at 450 MHz, obtained by [15].

5. CONCLUSIONS

In this paper, we have developed a technique to simultaneously model leakage power and temperature, in an FPGA context. Our CAD framework models the temperature-dependent total power consumption of the design at a given temperature (including its dynamic and static components) using compact macromodels. We divide the FPGA into a grid of 16 × 16 regions, and find this power consumption for each region. From this information, we find the thermal profile of the IC under this power consumption, using a pre-computed power-to-temperature matrix. The new thermal information is used to update the power consumption. These steps are iterated until the temperature of the IC converges, or exceeds a safe value. The iterations are very fast, due to the use of accurate and fast macromodels in the inner loop. Our experiments show that our approach helps avoid an incorrect estimation of chip temperature and power consumption, and identify thermal breakdown scenarios. The error of our temperature dependent leakage computation macromodel (over all circuit components in the FPGA - LUTs, MUXes, DFFs, SRAMs and INVs) is less than 3% compared to SPICE values, with a speedup of over 4 orders of magnitude over SPICE. Also, our final converged temperatures have a worst-case (average) error of 2.5% (1%) compared to a full-chip 3D temperature estimation tool.

6. REFERENCES

[1] www.xilinx.com.

[2] IWLS 2005 Benchmarks. `http://www.iwls.org/iwls2005/benchmarks.html`.

[3] J. H. Anderson and F. N. Najm. Power estimation techniques for FPGAs. *IEEE Trans. Very Large Scale Integr. Syst.*, 12(10):1015–1027, 2004.

[4] E. I. Boemo and S. López-Buedo. Thermal monitoring on FPGAs using ring-oscillators. In *FPL '97: Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications*, pages 69–78, London, UK, 1997. Springer-Verlag.

[5] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit design. In *Proc. of IEEE Custom Integrated Circuit Conference*, pages 201–204, Jun 2000. http://www-device.eecs.berkeley.edu/ ptm.

[6] Z. Chen and K. Roy. A power macromodeling technique based on power sensitivity. In *DAC '98: Proceedings of the 35th annual conference on Design automation*, pages 678–683, New York, NY, USA, 1998. ACM Press.

[7] Y. Cheng, C. Tsai, C. Teng, and S. Kang. *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.

[8] P. Chow, S. Seo, J. Rose, K. Chung, G. Paez-Monzon, and I. Rahardja. The design of a SRAM-based field-programmable gate array - part II : Circuit design and layout. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(3):321–330, Sept 1999.

[9] V. Degalahal and T. Tuan. Methodology for high level estimation of FPGA power consumption. In *ASP-DAC '05: Proceedings of the 2005 conference on Asia South Pacific design automation*, pages 657–660, New York, NY, USA, 2005. ACM Press.

[10] L. He, W. Liao, and M. R. Stan. System level leakage reduction considering the interdependence of temperature and leakage. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 12–17, New York, NY, USA, 2004. ACM.

[11] S.-M. S. Kang. On-chip thermal engineering for peta-scale integration. In *ISPD '02: Proceedings of the 2002 international symposium on Physical design*, pages 76–76, New York, NY, USA, 2002. ACM.

[12] H. G. Lee, K. Lee, Y. Choi, and N. Chang. Cycle-accurate energy measurement and characterization of FPGAs. *Analog Integr. Circuits Signal Process.*, 42(3):239–251, 2005.

[13] H. Li, W.-K. Mak, and S. Katkoori. LUT-based FPGA technology mapping for power minimization with optimal depth. In *WVLSI '01: Proceedings of the IEEE Computer Society Workshop on VLSI 2001*, page 123, Washington, DC, USA, 2001. IEEE Computer Society.

[14] P. Li. Critical path analysis considering temperature, power supply variations and temperature induced leakage. In *International Symposium on Quality Electronic Design (ISQED)*, Mar 2006.

[15] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra. Efficient full-chip thermal modeling and analysis. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 319–326, Washington, DC, USA, 2004. IEEE Computer Society.

[16] W. Liao, F. Li, and L. He. Microarchitecture level power and thermal simulation considering temperature dependent leakage model. In *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*, pages 211–216, New York, NY, USA, 2003. ACM Press.

[17] G. M. Link and N. Vijaykrishnan. Thermal trends in emerging technologies. In *ISQED '06: Proceedings of the 7th International Symposium on Quality Electronic Design*, pages 625–632, Washington, DC, USA, 2006. IEEE Computer Society.

[18] S. Lopez-Buedo and E. Boemo. Making visible the thermal behaviour of embedded microprocessors on FPGAs: a progress report. In *FPGA '04: Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, pages 79–86, New York, NY, USA, 2004. ACM.

[19] S. Lopez-Buedo, J. Garrido, and E. Boemo. Dynamically inserting, operating and eliminating thermal sensors of FPGA-based systems. *IEEE Transactions on Components and Packaging Technologies (CPM)*, 25(4):561–566, Dec 2002.

[20] H. Lui, C. Lee, and R. Patel. Power estimation and thermal budgeting methodology for FPGAs. In *IEEE Custom Integrated Circuit Conference (CICC)*, pages 32–1–1 to 32–1–4, 2004.

[21] P. Mal, J. Cantin, and F. Beyette. The circuit designs of an SRAM based look-up table for high performance FPGA architecture. In *45th Midwest Symposium on Circuits and Systems (MWCAS)*, volume III, pages 227–230, Aug 2002.

[22] L. Nagel. Spice: A computer program to simulate computer circuits. In *University of California, Berkeley UCB/ERL Memo M520*, May 1995.

[23] J. Rabaey. *Digital Integrated Circuits - a design perspective.* Prentice Hall, 1996.

[24] R. Rao, R. Brown, K. Nowka, and J. Burns. Analysis and mitigation of CMOS gate leakage. In *Proceedings of the Fifth Annual Austin Center for Advanced Studies Conference*, pages 7–11, Austin, TX, Feb 2004.

[25] A. Reimer, A. Schulz, and W. Nebel. Modelling macromodules for high-level dynamic power estimation of FPGA-based digital designs. In *ISLPED '06: Proceedings of the 2006 international symposium on Low power electronics and design*, pages 151–154, New York, NY, USA, 2006. ACM Press.

[26] I. Robertson and J. Irvine. A design flow for partially reconfigurable hardware. *Trans. on Embedded Computing Sys.*, 3(2):257–283, 2004.

[27] A. Sangiovanni-Vincentelli. The tides of EDA. Keynote Talk, Design Automation Conference, June 2003.

[28] L. Shang, A. S. Kaviani, and K. Bathala. Dynamic power consumption in Virtex-II FPGA family. In *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages 157–164, New York, NY, USA, 2002. ACM Press.

[29] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, 2004.

[30] P. Sundararajan, A. Gayasen, N. Vijaykrishnan, and T. Tuan. Thermal characterization and optimization in platform FPGAs. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 443–447, New York, NY, USA, 2006. ACM Press.

[31] S. Trimberger, editor. *Field-Programmable Gate Array Technology.* Kluwer Academic Publishers Group, Netherlands, 1994. ISBN: 9780792394198 (0792394194).

[32] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *Proceedings of IEEE Custom Integrated Circuits Conference*, pages 57 – 60, Sep 2003.

[33] N. Weste and K. Eshraghian. *Principles of CMOS VLSI design - a systems perspective.* Addison-Wesley, 1988.

[34] G.-M. Wu, J.-M. Lin, and Y.-W. Chang. Performance-driven placement for dynamically reconfigurable FPGAs. *ACM Trans. Des. Autom. Electron. Syst.*, 7(4):628–642, 2002.

[35] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A temperature-aware model of subthreshold and gate leakage for architects. Technical Report CS-2003-05, University of Virginia, Department of Computer Science, 2003.