Reducing Functional Unit Power Consumption and its Variation Using Leakage Sensors

Aviral Shrivastava, Deepa Kannan, Sarvesh Bhardwaj, and Sarma Vrudhula

***Abstract*—Energy reduction of functional units (FUs) is a very important concern for high-end superscalar processors, not only because FUs consume a significant portion of processor energy, but also because they are one of the most important hotspots in the processor. In addition, the high sensitivity of leakage on temperature and process variation result in very high variation in the FU power consumption in different processor dies. Such high process variation reduces the parametric yield of processors. Consequently, reducing the FU power consumption and its variation is an important problem. However, existing FU power reduction techniques assumes all the FUs are similar, and do not consider the sensitivity of leakage on temperature. Consequently, they are not very effective in reducing the variation of FU power consumption. The advent of extremely small, yet accurate leakage sensors allow us to develop leakage-aware microarchitectural techniques to reduce both the power consumption and its variation among processor dies. Our leakage-aware operation-to-FU binding mechanism (LA-OFBM) and leakage-aware power gating (LA-PG) mechanisms reduce the mean and standard deviation of the total arithmetic logic unit (ALU) power consumption of the ALPHA 21364 by 34% and 59%, respectively. At the processor level, this translates to a 13% reduction in the total processor energy consumption, with a 24 °C reduction in the maximum ALU temperature.**

***Index Terms*—Functional unit (FU) power reduction, leakage, process variations.**

I. Introduction

REDUCING the power consumption of the functional units (FUs) of high-end processors is important not only because they consume a significant percentage of processor energy, but also because they are one of the most important hotspots in the processor. Since FUs are one of the busiest components in a processor, they dissipate a lot of dynamic energy. This results in FUs becoming a hotspot in the processor. This is aggravated by the exponential dependence of leakage on the temperature, and FUs also become a site of high leakage. Thermal emergency, e.g., the temperature of an FU increasing beyond the safe temperature can significantly degrade processor performance. In addition, the high sensitivity of leakage on temperature and process variation result in very high variation in the FU power consumption in different processor dies. Such high process variation reduces the parametric yield of processors. Consequently, reducing the FU power consumption and its variation is an important problem.

Since, traditionally dynamic power was the main component of FU energy, early work on FU power reduction therefore focused on developing clock gating mechanisms to save energy during the idle cycles of the FUs [1], [2]. However, as shown by Rele *et al.* [3], for 65 nm and beyond, the leakage power of FUs is comparable to their dynamic powers. Consequently, several techniques including body bias control [4], input vector control [5], dual-threshold domino circuits [6], [7], and power gating [3], [4], [8]–[11] have been developed and researched to reduce the leakage of FUs. However, none of the previous FU leakage reduction mechanisms consider the differences in the FUs. FUs in the same die may have different power characteristics not only due to manufacturing variabilities, but also due to their current temperature.

The assumption of similar FUs is becoming increasingly inaccurate with incessant technology scaling. Leakage is extremely sensitive to process variations due to its exponential dependence on threshold voltage, which depends on the physical characteristics of transistor, including the gate length, gate width, doping level, etc. Process variability has grown in recent technologies due to random dopant effects in small devices, patterning of features smaller than the wavelength of the optical lithographic system and related trends. Also leakage power is very strongly related to the temperature in current technologies. In fact changing the temperature from 90 °C–130°C can result in 2× increase in the total power due to increase in the leakage currents [12]. Since the silicon substrate is a bad conductor of heat, there can be a temperature gradient of more than 30 °C even between the neighboring blocks on the chip [13].

The advent of extremely small (0.5 μW power) and highly accurate (< 1 °C inaccuracy) on-die leakage sensors, such as the ones developed by Kim *et al.* [14], now allow for runtime monitoring of the differences in FU power, and develop microarchitectural techniques to reduce the variation in the power consumption of FUs. This work is the first effort in this direction. In this paper, we: 1) model the impact of temperature and process variations on the leakage of FUs and 2) develop microarchitectural techniques to reduce not only the power consumption of FUs, but also reduce the variation in the FU power consumption among different dies. In particular, we propose: 1) a leakage-aware operation to functional unit binding mechanism (LA-OFBM), which decides the FUs to which the ready operations should be issued and 2) a leakage-aware power gating (LA-PG) mechanism, which decides which FUs to power gate, both aimed at reducing the variation in FU power consumption. Our experiments on the ALPHA DEC 21364 processor at the

Manuscript received September 01, 2008; revised December 29, 2008; accepted February 17, 2009. First published November 17, 2009; current version published May 26, 2010. This paper is an extended version of two papers titled, "Power Reduction of Functional Units considering Temperature and Process Variations" and "Temperature and Process Variations aware Power Gating of Functional Units", that appeared in the 21st International Conference on VLSI Design, 2008.

A. Shrivastava, D. Kannan, S. Vrudhula are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85048 USA (e-mail: aviral.shrivastava@asu.edu).

S. Bhardwaj is with Design Compiler Group, Synopsys Inc., Mountain View, CA 94043 USA.

Digital Object Identifier 10.1109/TVLSI.2009.2019082

45-nm technology node using the BSIM4 predictive models on benchmarks from Spec2000 and MiBench demonstrate that as compared to the traditional operation to functional unit binding mechanism, our scheme LA-OFBM reduces the mean and the standard deviation of the total ALU power consumption by 25% and 10%, respectively. Additionally, as compared to the traditional power gating scheme, our scheme, LA-PG reduces the mean and standard deviation of the total arithmetic logic unit (ALU) power consumption by 22% and 25%, respectively. Finally, by combining our two techniques, LA-OFBM + LA-PG reduces the mean and standard deviation of the total ALU power consumption by 34% and 59%, respectively. This translates into 13% reduction in the total processor energy consumption. In addition LA-OFBM + LA-PG result in 24 °C reduction in the maximum ALU temperature.

II. Related Work

Traditionally the leakage reduction of storage structures [15]–[19] has been research focus owing to their dominant transistor budget in a processor. However, due to high activity and therefore high temperature of FUs [11] and the exponential relationship of temperature on leakage [20], the power reduction of FUs has become an important concern. Since leakage of a transistor depends on the state of the circuit, during the standby mode, it's inputs can be set to low-leakage state. Such a technique for FUs was presented in [5]. Static energy is reduced in the integer functional units by leveraging the unique qualities of dual threshold voltage domino logic in [8]. However, by far, power gating is the most effective and researched microarchitectural techniques to reduce the leakage of FUs.

A. Previous Power Gating Techniques

Hu *et al.* [9] demonstrated that functional units are idle for more than 60% of total execution cycles across SPEC2K benchmarks for a superscalar processor. Duarte *et al.* [4] applied power gating techniques at the microarchitectural level on FUs. They explored various run-time power gating techniques and concluded that global power supply gating scheme, although effective, has high overheads, and is therefore not useful. They then proposed local power supply gating mechanism that achieves the same level of leakage reduction but with lower overheads. In addition, a power gating technique to gate the datapath logic and memory structure using a PLL circuit with a voltage follower has been proposed in [21]. An operand-value-based gating to turn off portions of arithmetic units that will be unused by narrow-width operations has been proposed and implemented in [22]. These techniques address the question of how to implement power gating.

The proposed solutions to perform power gating fall into two different categories: 1) compiler-based solutions and 2) microarchitectural solutions. Compiler-based FU leakage reduction techniques were studied in [3]. But these technique requires that the entire code be examined offline to identify suitable regions for turning the functional units off. Microarchitectural techniques [4] are idle-time-based power gating (IT-PG) techniques, which rely on predicting idle cycles of the FU and power gate the FU during that time. We compare our approach against IT-PG.

B. Previous OFBMs

A resource allocation and binding approach for low leakage superscalar processors is proposed in [23]. Two temperature-aware resource allocation and binding algorithms are proposed in [24]. A compiler technique of changing the IPC and attempting to balance the tradeoff between performance and leakage energy is employed in [25]. At the microarchitecture level [11] evaluated fixed priority OFBM (FP-OFBM), in which the FUs are statically assigned priorities and in each cycle, an operation is issued to an FU only if an operation has been issued to all FUs with higher priority. FP-OFBM causes high priority FUs to heat up, and therefore to evenly distribute the heat over all the FUs, they proposed load balancing OFBM (LB-OFBM), in which the operations are distributed among FUs in a round robin fashion.

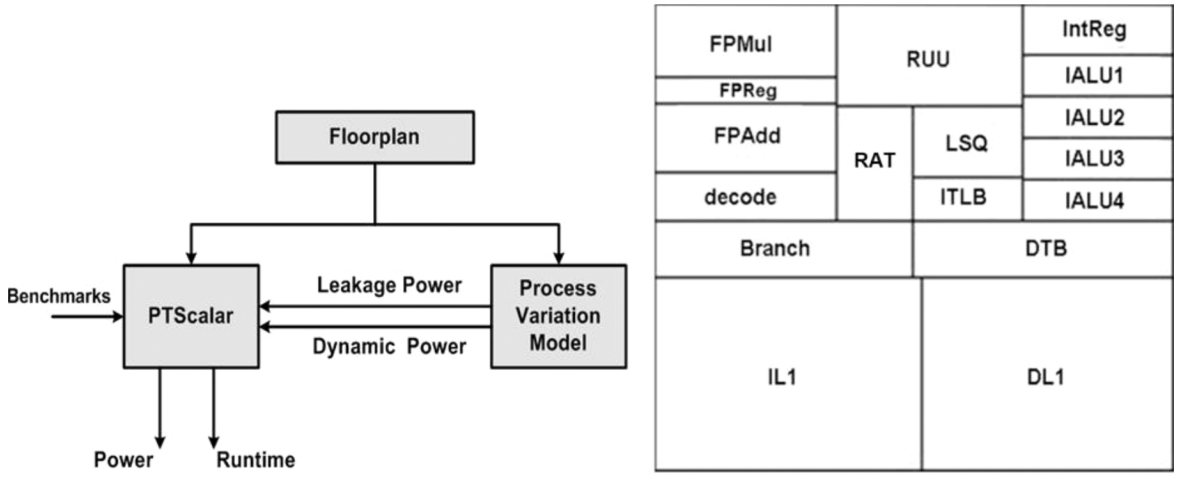
However, *none of the previous OFBM and power gating approaches consider the impact of temperature and process variations on leakage of FUs, and none of them attempt to reduce the variation in the leakage power of FUs among die samples.* In this paper, we propose power gating and OFBMs which aim at reducing both the leakage power, and the variation in the leakage power of FUs and compare our approach against existing OFBMs.

III. Experimental Setup

We have developed a processor power and performance simulation framework shown in Fig. 1(a) to implement our proposed operation to FU binding mechanisms and power gating techniques. The power, performance, and temperature modeling of alpha processor is done using a modified version of sim-outorder of the PTScalar toolset. PTScalar has been validated to model the power performance and thermal behavior of the DEC ALPHA 21364 processor correctly [12], [26]. The floorplan of the DEC ALPHA 21364 is shown in Fig. 1(b), and contains 4 ALUs contiguously located on the northeast corner of the die just below the integer register file. While keeping the package parameters the same, we scale the power models of processor components to 45-nm technology. This is because while the technology scaling has been exponential, more efficient cooling solutions are extremely expensive [27]. Our process variation model introduces random and correlated variations in the gate length (L) and the threshold voltage (V) of the transistors inside the FUs. To model the spatial correlations in device parameters, our process variation model takes the DEC ALPHA 21364 floorplan as an input. The overall dynamic and leakage power of all the 4 ALUs are then given to PTScalar as input. Using the floorplan, PTScalar performs a cycle-accurate simulation of benchmarks to estimate the power, performance and temperature of all the ALUs and the overall processor power and performance. We execute several benchmarks from the MiBench [28], and Spec 2000 [29] suite. We repeat these experiments over 1000 die samples.

IV. Modeling Process Variations

An important consequence of technology scaling is the decreasing control in the lithography as well as channel doping steps during the manufacturing of nano-scale circuits. This results in significant amount of variations in the characteristics of



(a) (b)

Fig. 1. Experimental setup. (a) Simulation framework. (b) Floorplan of ALPHA DEC 21364 processor.

manufactured devices [30]. A significant component of the variations in parameters can be contributed to intra-die variations [30], [31]. The intra-die variations can be divided into two main components: 1) spatially correlated random variations and 2) deterministic systematic variations. To model systematic variations, the parameters of a device are modeled as a deterministic function of the location of that device on the die. This function can be used to analyze the effect of systematic variations on some characteristic (e.g., leakage and delay) of the design. However, modeling spatially correlated random variations is not that straightforward.

The random parameter p_i of device i is modeled as

$$p_i = p_{o,i} + \sigma_{p,i} \cdot \zeta_{p,i} \tag{1}$$

where $p_{o,i}$ and $\sigma_{p,i}$ are the mean and standard deviation of the parameter, and $\zeta_{p,i}$ is a standard Gaussian random variable.

Recently, an accurate approach for modeling spatial correlations was proposed in [32]. The method is based on modeling the parameters of the devices as a quadratic mean continuous stochastic process $Z(x, y)$. This method has the advantage that the parameter of each device on the die can be modeled as a *unique* random variable. The process is described using the correlation function $C(x_1, y_1, x_2, y_2)$ between two points (x_1, y_1) and (x_2, y_2) on the die. The correlation function $C(x_1, y_1, x_2, y_2)$ is used to obtain a linear transformation that transforms the original set of correlated Gaussian random variables to a new set of independent Gaussian random variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_r)$. That is the parameter $\zeta_{p,i} = Z(x_i, y_i)$ of a device located at (x_i, y_i) can be written as

$$\zeta_{p,i} = Z(x_i, y_i) = \sum_{j=1}^{r} a_j(x_i, y_i) \cdot \xi_{p,j}. \tag{2}$$

The previous transformation is called Karhunen–Loève expansion [33]. Note that the coefficients $a_j(x_i, y_i)$ are a function of the location of the device. The number of terms r in the expansion are determined based on a predetermined error criteria such as the expansion accurately models the statistics of the original process [32]. Since the location of each device is different, the coefficients for each parameter will be different. Hence, all the parameters are modeled as different random variables. Also note that this mechanism of modeling spatial correlation, does not preclude the case of modeling completely uncorrelated variations, e.g., variation in the dopant concentration; they can be modeled as a special case, with $j = 1$.

In this work, we model the variations in the gate length of the devices as a stochastic process. We fit the measured experimental data for the CD variations given in [30] to an exponentially decreasing correlation function

$$C(x_1, y_1, x_2, y_2) = \exp\left(\rho_X |x_1 - x_2|\right) \cdot \exp\left(\rho_Y |y_1 - y_2|\right) \tag{3}$$

to obtain the parameters ρ_X and ρ_Y. This correlation function is used to compute the coefficients $a(x_i, y_i)$ in (2). Since the random variables $\xi_{p,j}$ corresponding to the parameter p are independent Gaussian random variables, we generate independent samples of Gaussian variables to simulate the values of the gate lengths in the process. Each sample of r independent random variables corresponds to one die sample. To generate samples for M dies, this process is repeated M times and the gate length of *all* the devices on the die are computed using (2).

A. Impact of Variations on FUs

Since the leakage power has an exponential dependence on the gate length, even small variations in the gate length can result in significant variations in the leakage power of the manufactured circuits. In this work, we model the sub-threshold leakage $I_{S,i}$ of a gate i as a function of gate length (L) and threshold voltage (V) based on the BSIM model [34]

$$I_{S,i} = I_{So} \cdot \frac{w_i}{L_{e,o}^k} \cdot \exp\left(\frac{-(\zeta_{V,i} + a_1 \cdot \zeta_{L,i} + a_2 \cdot \zeta_{p,i}^2)}{S}\right), \; k > 1. \tag{4}$$

The previous model was fitted to data from 45-nm technology with average error of less than 5% across 20% range of the gate length, centered about the nominal value of the gate length. Since we did not have a gate level model for the ALU, we estimate total leakage of the ALU using $I_{S,T} = N \cdot I_{s,i}$, where N is the number of transistors in the ALU.

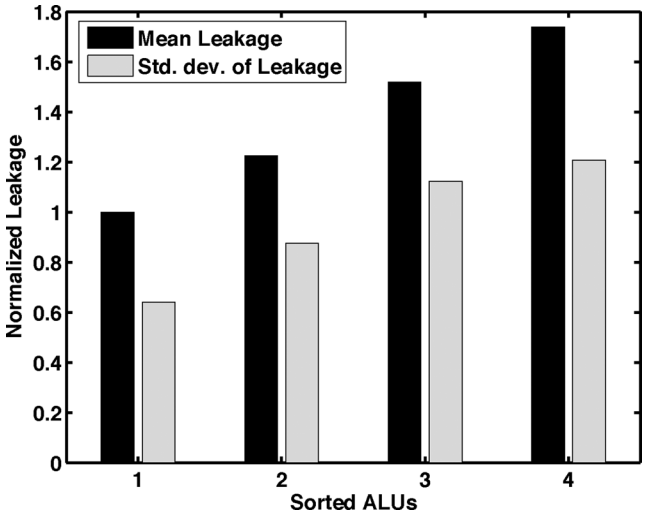


Fig. 2. Statistics of leakage power of ALUs.

Similarly, we model the dynamic power I_D of the ALU as

$$I_D = \alpha \cdot C_{\text{eff}} \cdot V_{\text{dd}}^2 \cdot f \tag{5}$$

where α is the switching factor, C_{eff} is the total effective capacitance, V_{dd} is the supply voltage, and f is the frequency of operation. We assume that the major source of variations in the dynamic power is due to the capacitance, which is a linear function of the gate length of the device.

Based on the previous power models and the process variation data of [30], we find both the leakage and dynamic power for ALUs in each sample die. In order to estimate the average difference in the leakage of ALUs, we sort the ALUs according to their leakage power and compute their statistics. For example, Fig. 2 shows the comparison in the mean and standard deviation of the leakage power of the four ALUs sorted according to the leakage power. As shown in the figure, on an average, there can be up to 80% difference in the leakage of two ALUs on the same die. This shows that there is a significant scope of power savings by considering intra-die variations.

V. Leakage Sensors

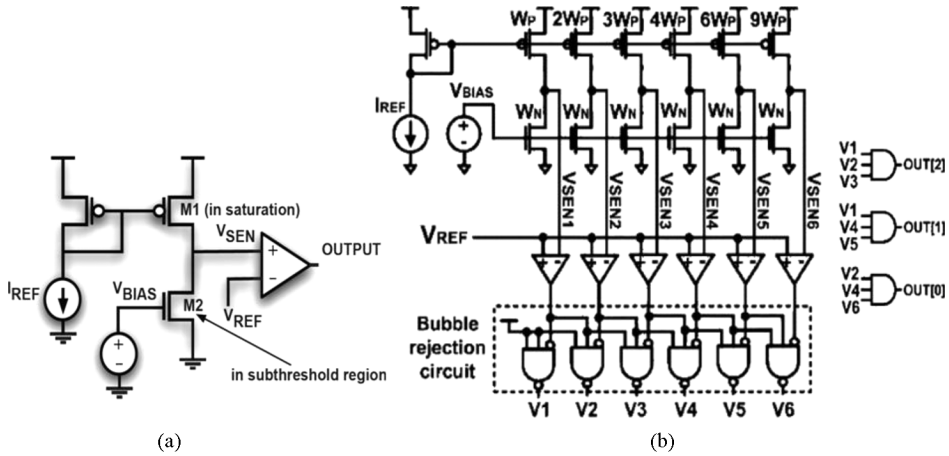
In the recent years, there has been an extensive research in the design and implementation of leakage current and temperature sensors. Kuroda *et al.* [35] were the first to propose a leakage current monitor that measured V_{th} variation with less than 1% error. But this design was limited by the complexity and power overhead of the biasing circuit. Griffin *et al.* [36], proposed a current sensor design to detect the variations in circuit operating conditions and to control the output slew rate. However, their design suffered from lower sensitivity and higher design complexity. Chen *et al.* [37] proposed an inter-die leakage variation tolerant, leakage canceling current sensor to improve the performance of self-timed circuits by tracking the local variations at the cost of a higher power overhead incurred in designing the circuit. Srivastava *et al.* [38] presented a built-in current sensor design for quiescent current (I_{DDQ}) testing of CMOS data converter circuits. The accuracy, however, is strongly related to the aspect ratio of the transistors forming the current mirror. Pertijs *et al.* [39] proposed a CMOS temperature sensor design that was accurate to within $+/-0.5\,^{\circ}\text{C}$ (3σ) over the range of $-50\,^{\circ}\text{C}$ to $120\,^{\circ}\text{C}$. Duarte *et al.* [40] presented various techniques on maximizing the accuracy of thermal sensors to within $+/-1\,^{\circ}\text{C}$.

We use the 6-channel leakage current sensor proposed by Kim *et al.* [14]. A single channel and six channel leakage sensors are shown in Fig. 3(a) and (b), respectively. The underlying principal on which the leakage sensor is based is the drain induced barrier lowering effect. Due to this effect, under constant bias currents the drain to source voltage V_{ds} is a monotonically increasing function of the device threshold voltage V_t[14]. Since the threshold voltage V_t is dependent on both the random dopant component (V_o) and the effective gate length (L) [34], the leakage sensor can measure variations in leakage power due to both V_o and L. V_{SEN} is the drain voltage of device M2 that indicates the leakage level. A comparator is used for the analog-to-digital (A/D) conversion of V_{SEN} giving an output of "1" when V_{SEN} is higher than V_{REF} and, otherwise a "0". In the 6-channel leakage sensor, by comparing V_{SEN1}-V_{SEN6} with an external V_{REF}, the leakage level can be determined. The bias circuits for generating I_{REF} and V_{BIAS} are process variation insensitive. M2 is the only transistor that is sensitive to the variation in leakage of the ALU due to the impact of temperature and process variations. The single channel and six channel leakage sensing circuits proposed, consisted of process variation insensitive bias circuits. The leakage sensor has a higher sensing gain when compared to [35] and [36], with an inaccuracy of less than 3% and minimal area and power overhead ($< 0.5\ \mu$W).

To measure a wide range of variations across different process corners, the authors in [14] use 6-channels (instead of single channel). Although 6-channels would correspond to 6-bit output resulting in a resolution of 2^6 different leakage values. However, due to the presence of intra-die variations the resulting *code* might not be correct. Hence a *bubble rejection circuit*, which reduces the resolution to 8 levels, is used to give correct output even in the presence of intra-die variations. Thus, the leakage sensor output is a 3-bit binary code sequence $\text{OUT}[2:0]$. The 1.2 V, 0.66 mW, 0.006 mm^2 6-channel leakage sensor is designed using transistors from the 45-nm CMOS technology and has 1.9–10.2 $\times$ higher sensing gain compared to previous leakage sensor designs [14]. The overhead of using leakage sensors accounts to around 3%–4% reduction in the total power savings obtained using our LA-OFBM. We take into account in all our experiments, the inaccuracy in converting the leakage sensor values into discrete output values.

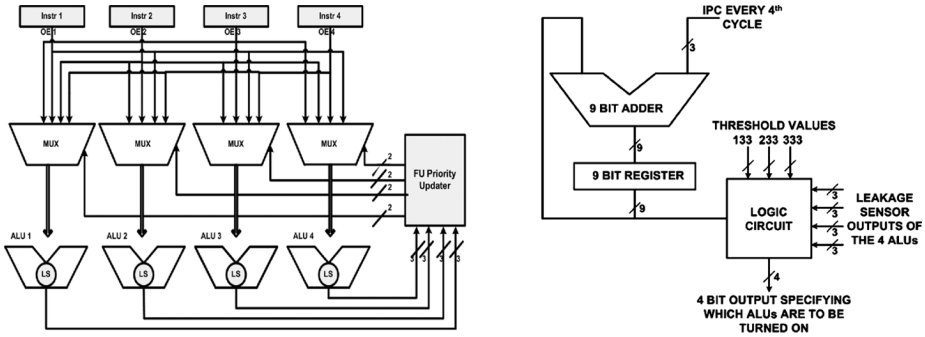
A. Leakage Sensor Placement

If the leakage measured by the sensor is not a good estimate of the total leakage of the FU, we might not get the correct ordering of the FUs in terms of their leakage power. This could potentially result in instructions being bound to a higher leakage FU instead to the lowest leakage FU, thus eliminating the power savings obtained using our approach. To find a good location for the leakage sensor, we compared the leakage of a device located at various locations (x_i, y_i) inside the FU, and the average leakage of the FU ($I_{av} = I_{S,T}/N$). We found that mean of the percentage difference between the average FU leakage and the leakage of a device located at the center of the FU for a sample of 1000 dies to be less than 1%. The maximum percentage error over the same set of samples was 7%. Thus a single leakage sensor located at the center of the FU can provide accurate estimation of the leakage power of the entire FU.



(a) (b)

Fig. 3. (a) Single and (b) 6-channel leakage sensor.



(a) (b)

Fig. 4. (a) Architecture model for LA-OFBM. (b) Microarchitectural enhancements for the IPC threshold-based power gating technique.

VI. Leakage-Aware OFBM (LA-OFBM)

In the LA-OFBM, whose architecture model is shown in Fig. 4(a), operations are issued to the FUs based on their leakage information. The sensors within the ALUs are used to accurately detect leakage and set the ALU priorities dynamically in the FU Priority Updater. The leakage sensor values are continuously read and the FU priorities are updated after every 10 000 cycles, to be in the decreasing order of increasing leakage. We introduce four 4-to-1 line multiplexors in the operation issue path, to select the ALUs to which the incoming instructions are to be issued. Since the temperature of the ALUs vary over time, the priorities assigned to the ALUs will also change dynamically. LA-OFBM is therefore both process and temperature variations aware OFBM.

A. Microarchitectural Overheads

We accurately model the impact of microarchitectural enhancements structurally in PTScalar. The leakage sensor is very small, only a few gates, and is not in the critical path of the ALU. Therefore there is no performance impact of the sensor. The multiplexors lie in the critical path of execution, they might cause some extra delay. However this is very small, and in our experiments, we observe that it can be accommodated in the cycle time slack. We synthesized the multiplexors and the FU Priority Update logic using Synopsys design compiler [41]. The energy overhead of multiplexors and the FU priority Updater is less than 0.75 μJ which is very small as compared to the 500 μJ energy of all the 4 ALUs. But we included both their leakage and dynamic powers in the power computation using PTscalar in all our simulations.

VII. Leakage Aware Power Gating (LA-PG)

Our first observation is that temperature increases are gradual, and like most previous works [12], [26], we assume that appreciable temperature changes occur only at 10 000 cycle granularity, and therefore it is reasonable to implement temperature dependent policies at this granularity. Our power gating mechanism is a two step process: we use the current IPC information to find out how many FUs to power gate, and then we use leakage sensor values to determine which FUs to power gate.

A. How Many FUs to Power Gate?

At each decision moment (i.e., every 10 000 cycles), we compute the *average IPC*, or the average number of instructions that are *ready to be issued* every cycle. Note that this is different from the regular definition of instructions per cycle (IPC), which is the number of instructions issued each cycle. However, due to it's close similarity to IPC, and since we do not use IPC otherwise in this paper, we call our approach as IPC-based technique. The number of FUs to power gate is determined by comparing our computed average with a *threshold*. For a n FU configuration, we have $n-1$ thresholds. For a 4-ALU system, there will be 3 thresholds, to turn "on" the ALUs. Suppose the thresholds are 1.1, 2.1, and 3.1, then we will turn on 2 ALUs if the average IPC is more than 1.1, and turn on 3 ALUs if the average IPC is more than 2.1, and we will turn on all the 4 ALUs if the computed average IPC is more than 3.1.

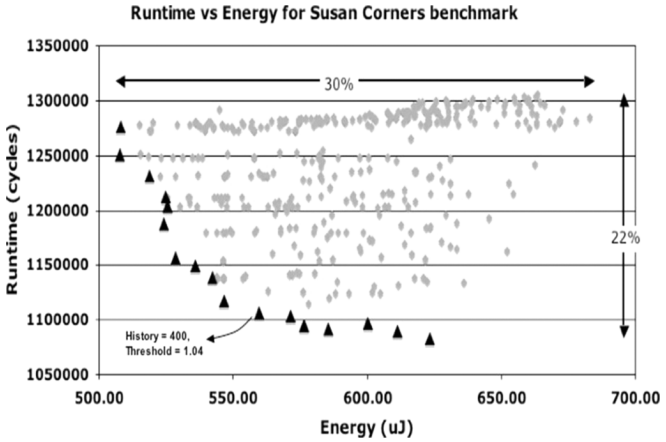


Fig. 5. Runtime versus energy showing the pareto-optimal points for susan corners benchmark.

The *average IPC* is computed as a average of IPCs of the last *history* number of cycles. The value of *history* determines the accuracy of our power gating technique. Therefore, the history and the thresholds are the two key parameters of our IPC threshold-based power gating technique. Designers can vary these parameters to trade off power, performance, and architectural complexity. To determine suitable values of history and thresholds, we simulated all the 10 benchmarks with IPC threshold-based power gating technique for several values of history and thresholds. We vary the history from 10 to 1000 cycles, and the threshold value for the case when a single ALU is in the active mode from 1.0 to 1.20 in steps of 0.01. The corresponding values for two and three ALUs to be in the active mode are varied from 2.0 to 2.20 and 3.0 to 3.20.

Fig. 5 shows the runtime vs energy plot for all history-threshold configurations for the representative *susan-corners* benchmark. The figure shows that a variation of 30% in the energy of the ALUs and a variation of 22% in the runtime is possible by power gating. We have identified and marked the pareto-optimal points by the dark triangles in the figure. A configuration is pareto-optimal if it is not worse than any other configuration in both power and performance. Designers can choose any of these pareto-optimal design points to tradeoff power and performance.

We varied the values of history and thresholds for all 10 benchmarks, and computed the energy-delay product for each history-threshold configuration. We then compute the summation of the energy-delay product for all benchmarks for each history-threshold configuration. We choose the configuration with the minimum sum as the best configuration for history and threshold. The optimal values of history and thresholds came out to be 400, 1.04, 2.04, and 3.04, respectively, and we use these values in estimating the effectiveness of our approach.

B. Which FUs to Power Gate?

In order to reduce the leakage, we want to power gate the FUs which have the highest leakage. We introduce the six channel leakage sensor proposed by Kim *et al.* [14] inside each FU and continuously measure the FU leakage during the chip operation. An FU may have high leakage either because of process variations, or because it's temperature is high. Thus, LA-PG is both temperature and process variation aware. Power gating the FU with the highest leakage, minimizes the FU power consumption; in addition it also reduces the variation in the leakages of FUs.

1) Microarchitectural Optimizations: A naive implementation of the circuit to estimate the average IPC can have high power, and performance overheads. Microarchitectural enhancements are needed to efficiently estimate the average IPC, and implement the logic of how many and which FUs to power gate. The first optimization we do is that we define IPC to be only from 0 to n_{issue}, for a n-*issue* superscalar processor, instead of 0 to n_{reorder} for a superscalar processor, with a reorder buffer of size n_{reorder}. The size of reorder buffer can be quite large in out-of-order superscalar processors. If the number of instructions that are ready is more than n_{issue}, we still express it as n_{issue}, or in other words, IPC saturates at n_{issue}. This reduces the microarchitectural overhead tremendously. In addition, instead of keeping a history of 400 IPC values, we collect IPCs every fourth cycle for a period of 512 cycles. This essentially increases the span of history, but reduces the sampling rate. This results in 128 samples of IPC over 512 cycles. On a 4-issue superscalar, the maximum value of the sum of the IPC over the entire sampling period will not exceed $128 \times 4 = 512$. Hence, a 9 bit adder is sufficient for this purpose. Since, this is only a 9-bit adder, it can be implemented as very low-power *ripple carry adder*, and still meet the timing constraint. This reduces the power consumption of the architectural overhead. Fig. 4(b) describes our overall design.

The IPC sum is accumulated in the 9 bit register, and the sum of the 128 samples over a range of 512 cycles is given as an input to the power gating logic circuitry. The combinational logic circuit determines how many ALUs to power gate based on the IPC sum and the threshold values and which ALUs to power gate based on the leakage sensor output values. The average IPC threshold values were computed as 1.04, 2.04, and 3.04 for a history for 400. With this new optimized implementation, we performed the exploration again, and found out that optimal values of the running totals of 128 IPC values of 512 cycles should be 133, 233, and 333, respectively. In addition to this, 3 bit output of the leakage sensor placed in each of the 4 ALUs also comes to the combinational logic. The circuit compares the IPC sum with the threshold values and determines the number of ALUs to power gate. It then processes the leakage sensor outputs according to the LA-PG technique and outputs a 4 bit value that determines which ALUs will be power gated in the next 10 000 cycles. To better explain the process, consider for instance that the IPC sum computed after the first 10 000 cycles is less than 233. It means that only 2 ALUs are required to be turned on in the next 10 000 cycles and the other 2 ALUs can be power gated. The leakage sensor output of the 4 ALUs are then processed according to the LA-PG technique and suppose the fourth and third ALUs are chosen to be power gated because of their higher leakage. The logic circuit then gives as output a 4 bit value 1100, where the first bit, 1 represents the MSB or ALU4 and the last bit, 0 represents the LSB or ALU1, indicating that ALU4 and ALU 3 are to be power gated.

We synthesized this logic using Synopsys Design Compiler and implemented it in Cadence Spectre toolset (Virtuoso schematic editor) using TSMC 0.25-μm CMOS deep submicron process, and scaled the numbers to 45 nm. We also synthesized the logic for the IT-PG technique for comparison purposes. This logic has an area overhead of 3% and energy

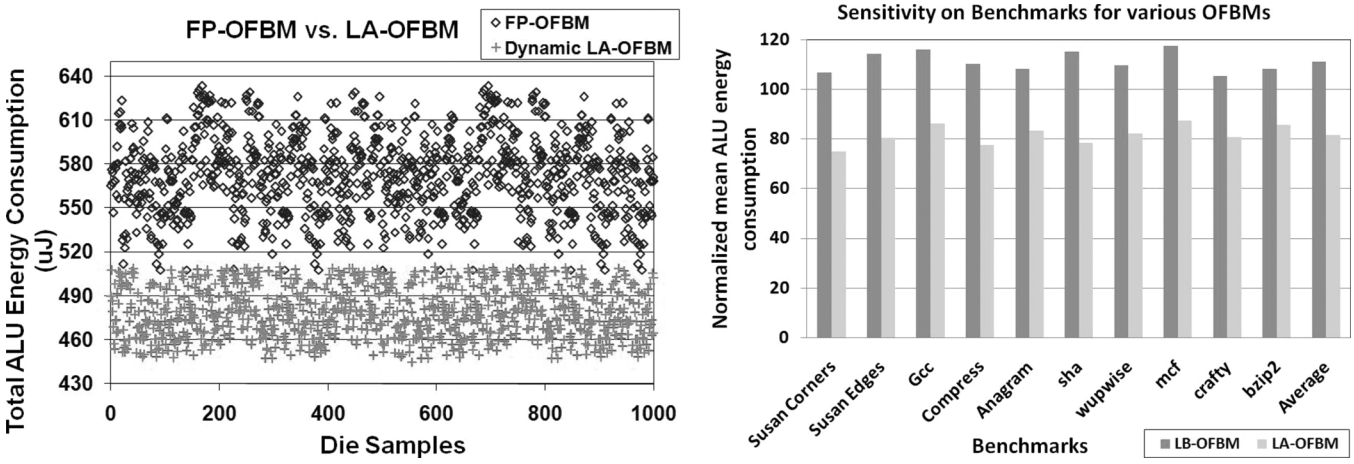


Fig. 6. LA-OFBM reduces the total ALU energy consumption.

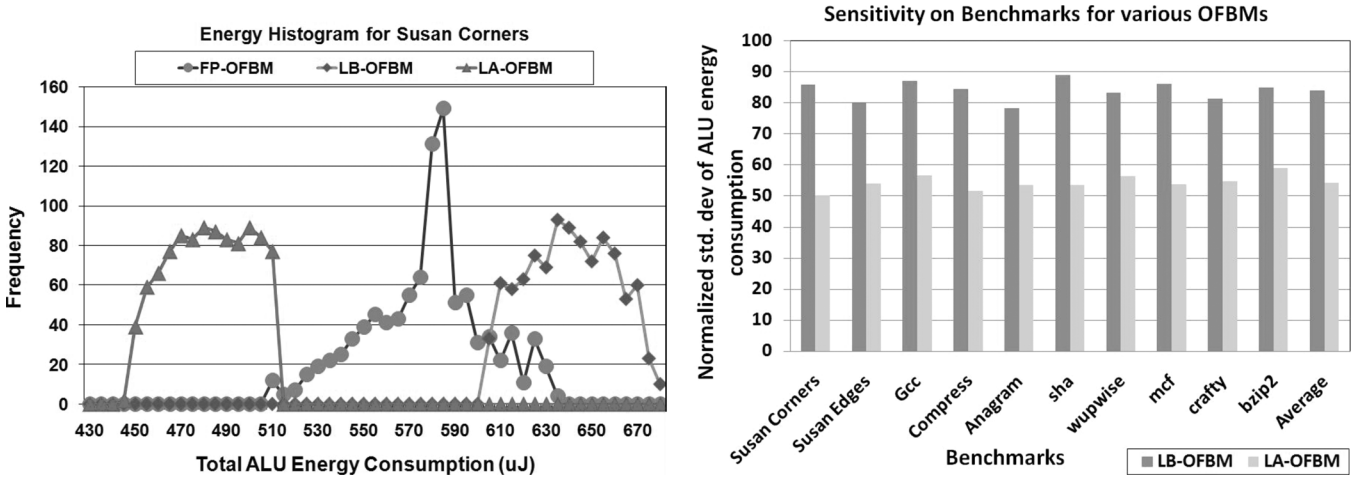


Fig. 7. LA-OFBM reduces the variation in the total ALU energy consumption.

overhead of $< 0.15\%$, as compared to the architectural overhead of idle-time-based technique. Furthermore, since this logic does not become a part of timing critical path in the processor, it does not cause an increase in the cycle time.

VIII. Experiments

A. LA-OFBM Reduces Both the Total ALU Energy Consumption and the Variation in the Total ALU Energy Consumption

Fig. 6(a) plots the total energy consumption of the ALUs for the FP-OFBM (baseline) and LA-OFBM techniques for 1000 die samples, for the representative *susan-corners* benchmark from MiBench suite. The first observation we make from this figure is that all the LA-OFBM points are lower than the FP-OFBM points. As compared to the FP-OFBM, LA-OFBM reduces the total energy consumption of the ALUs by 18%. In terms of leakage energy alone, the LA-OFBM decreases the total leakage energy of all the ALUs by 44% as compared to FP-OFBM. Fig. 6(b) further bolsters our claim by demonstrating that LA-OFBM consistently reduces the energy consumption across the benchmark spectrum. Another observation that we make from Fig. 6(a) is that the width of the vertical band in which points of LA-OFBM lie is lesser than the width of the band in which the points of FP-OFBM lie.

Fig. 7(a) plots another view of the same data. It plots the energy histogram for each of the OFBMs for *susan-corners* benchmark for 1000 die samples. The second curve from the right corresponds to the energy distribution for FP-OFBM. The rightmost curve is the energy histogram of LB-OFBM, which increases the mean energy consumption but reduces the standard deviation in the energy consumption by 17%. Finally LA-OFBM shown by its energy histogram depicted by the leftmost curve, reduces the energy consumption by 18% and reduces the standard deviation by 46% as compared to FP-OFBM. Fig. 7(b) shows that the reduction in variation in total ALU energy, by LA-OFBM is consistent over all our benchmarks. Thus our LA-OFBM reduces the total energy as well as the variation in total energy in the presence of both temperature and process variations.

B. LA-PG Reduces Both the Total ALU Energy Consumption and the Variation in the Total ALU Energy Consumption

Fig. 8(b) plots the mean of the ALU energy consumption computed over 1000 sample dies, normalized to IT-PG, for LA-PG, for all the 10 benchmarks. The figure shows that for architectures that have leakage sensors in FUs, LA-PG technique

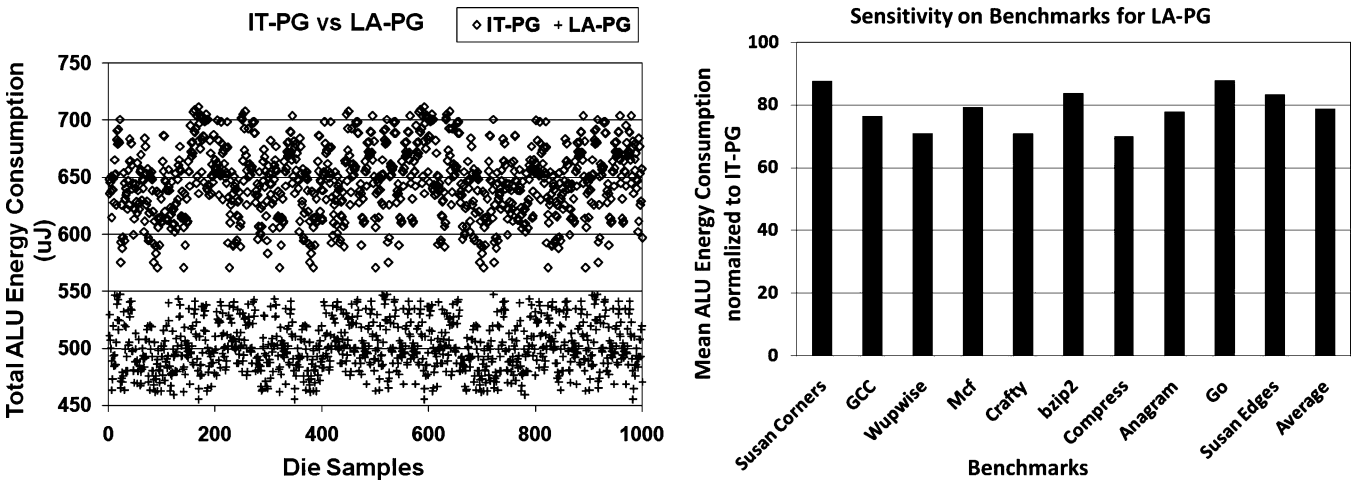


Fig. 8. LA-PG reduces the total ALU energy consumption.

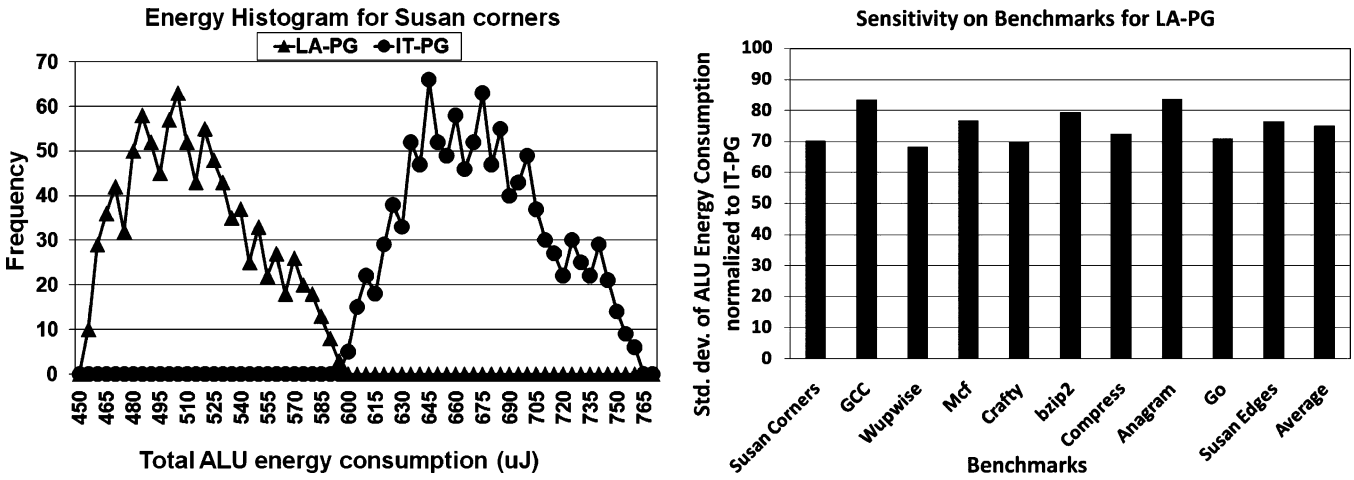


Fig. 9. LA-PG reduces the variation in the total ALU energy consumption.

decreases the average energy consumption by 22% as compared to the IT-PG. Hence, it can be observed that the effectiveness of our technique is consistent through the benchmark spectrum. The performance penalty of applying our IPC threshold-based power gating techniques is less than 2%. This performance loss is lesser than the performance loss of IT-PG, which is 2.2%, as compared to the case with no power gating.

Fig. 9(a) plots the energy histogram for IT-PG and LA-PG techniques, for *susan-corners* benchmark for 1000 die samples. The rightmost curve corresponds to the energy distribution for IT-PG. As compared to IT-PG, LA-PG as shown by its energy histogram depicted by the leftmost curve reduces the energy consumption by 22% and reduces the standard deviation by 25%. Fig. 9(b) further shows that the reduction in standard deviation by LA-PG is consistent over benchmarks.

Fig. 8(a) plots another view of comparison between the ALU power consumption for IT-PG and LA-PG in 1000 die samples for the *susan-corners* benchmark. The same two observations can be made from this graph are that all the LA-PG points are lower than the IT-PG points, and that the width of the vertical band in which points of LA-PG lie is lesser than the width of the band in which the points of IT-PG lie.

C. LA-OFBM + LA-PG Reduces Both the Total ALU Energy Consumption and the Variation in the Total ALU Energy Consumption

In order to obtain maximum reduction in the total ALU energy consumption and the variation in the total energy consumption, we simulate all the 10 benchmarks with all combinations of the two proposed techniques (OFBM + PG). The combination of FP-OFBM + IT-PG is considered as the basecase for these experiments. Fig. 10(a) plots the mean and standard deviation of ALU energy consumption, for all combinations of OFBM + PG normalized to the basecase (FP-OFBM + IT-PG), for susan corners benchmark. Out of all the combinations explored, LA-OFBM + LA-PG combination was observed to give the maximum reduction in both the mean and the standard deviation of the total ALU energy consumption. Our experimental results show that the combination of LA-OFBM + LA-PG reduces the mean and standard deviation of the total ALU energy consumption by 34% and 59%, respectively, as compared to the baseline combination of FP-OFBM + IT-PG. Fig. 10(b) shows that the results obtained for the LA-OFBM + LA-PG combination are consistent across all our benchmarks.

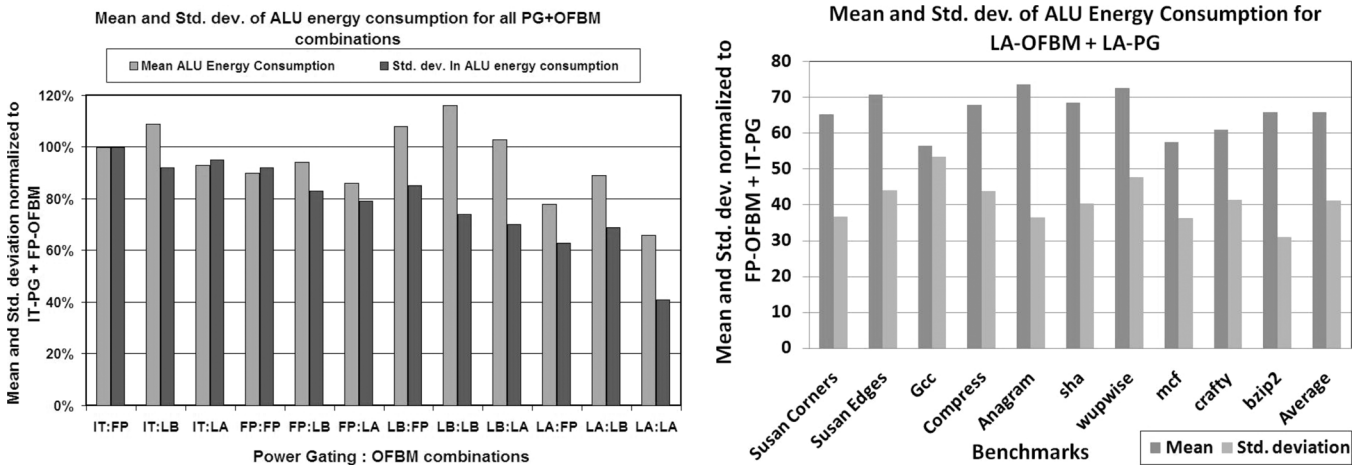


Fig. 10. LA-PG + LA-OFBM reduces both the total ALU energy consumption and its variation.

The reduction in the total ALU energy consumption using our LA-OFBM + LA-PG combination also translates to a 13% reduction in the total processor energy consumption. Our leakage aware OFBM + PG combination, as a result of reducing the total ALU energy consumption, also reduces the maximum ALU temperature by 24 °C. Thus, a combination of our leakage aware OFBM and leakage aware Power Gating techniques result in a significant reduction in the ALU energy consumption as well as the variation in ALU energy consumption.

IX. Summary

The FUs in a processor not consume significant amount of processor energy, but are also one of the most important hotspots in the processor. In addition, due to extreme sensitivity of leakage on temperature and process variations, they cause large variations in the power consumption of the processor dies, resulting low parametric yield. Prior work on FU power reduction do not consider the impact of temperature and process variations, and therefore are unable to reduce the FU power consumption and its variation. In this paper, we exploit recently developed, small, yet accurate leakage sensors, to develop microarchitectural techniques to reduce the FU power consumption and its variation. In specific, we develop a leakage-aware operation-to-functional unit binding mechanism (LA-OFBM), and leakage-aware power gating (LA-PG) schemes. Our experiments on the ALPHA 21364 processor shows that by combining these two leakage-aware techniques, we achieve 34% and 54% reduction in the FU power consumption, and in its variation respectively. This translates into 13% reduction in the total processor energy, and a 24 °C decrease in the maximum FU temperature.

References

[1] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," 1992. [Online]. Available: citeseer.ist.psu.edu/chandrakasan95low.html

[2] V. Tiwari, R. Donnelly, S. Malik, and R. Gonzalez, "Dynamic power management for microprocessors: A case study," in *Proc. 10th Int. Conf. VLSI Des.: VLSI Multimedia Appl. (VLSID)*, Washington, DC, 1997, p. 185.

[3] S. Rele, S. Pande, S. Onder, and R. Gupta, "Optimizing static power dissipation by functional units in superscalar processors," in *Proc. Computational Complexity*, 2002, pp. 261–275.

[4] D. Duarte, Y.-F. Tsai, N. Vijaykrishnan, and M. J. Irwin, "Evaluating run-time techniques for leakage power reduction," in *Proc. VLSI Des.*, 2002, pp. 31–38.

[5] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," 1997. [Online]. Available: citeseer.ist.psu.edu/halter97gatelevel.html

[6] K. Roy, S. Prasad, J. Claude, and E. Roy, *Low-Power CMOS VLSI Circuit Design*. New York: Wiley-Interscience, 2000.

[7] J. Casmira and D. Grunwald, "Dynamic instruction scheduling slack," presented at the Kool Chips Workshop, Dec. 2000.

[8] S. Dropsho, V. Kursun, D. Albonesi, S. Dwarkadas, and E. Friedman, "Managing static leakage energy in microprocessor functional units," in *Proc. 35th Ann. IEEE/ACM Int. Symp. Microarch.*, 2002, pp. 321–332.

[9] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, "Microarchitectural techniques for power gating of execution units," in *Proc. ISLPED*, New York, 2004, pp. 32–37.

[10] W. Zhang, Y.-F. Tsai, D. Duarte, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Reducing dynamic and leakage energy in vliw architectures," *Trans. Embed. Comput. Syst.*, vol. 5, no. 1, pp. 1–28, 2006.

[11] M. Mutyam, F. Li, V. Narayanan, M. Kandemir, and M. J. Irwin, "Compiler-directed thermal management for vliw functional units," *SIGPLAN Not.*, vol. 41, no. 7, pp. 163–172, 2006.

[12] W. Liao, F. Li, and L. He, "Microarchitecture level power and thermal simulation considering temperature dependent leakage model," in *Proc. ISLPED*, New York, 2003, pp. 211–216.

[13] A. Gupta, N. D. Dutt, F. J. Kurdahi, K. S. Khouri, and M. S. Abadir, "Stefal: A system level temperature- and floorplan-aware leakage power estimator for socs," in *Proc. 20th Int. Conf. VLSI Des.*, 2007, pp. 559–564.

[14] C. H. Kim, K. Roy, S. Hsu, R. Krishnamurthy, and S. Borkar, "A process variation compensating technique with an on-die leakage current sensor for nanometer scale dynamic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 6, pp. 646–649, Jun. 2006.

[15] S.-H. Yang, B. Falsafi, M. D. Powell, K. Roy, and T. N. Vijaykumar, "An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance i-caches," in *Proc. HPCA*, Washington, DC, 2001, p. 147.

[16] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: Exploiting generational behavior to reduce cache leakage power," in *Proc. ISLPED*, 2001, pp. 240–251.

[17] H. Hanson, M. Hrishikesh, V. Agarwal, S. Keckler, and D. Burger, "Static energy reduction techniques for microprocessor caches," presented at the Int. Conf. Comput. Des., [Online]. Available: citeseer.ist.psu.edu/hanson01static.html

[18] P. Li, Y. Deng, and L. T. Pileggi, "Temperature-dependent optimization of cache leakage power dissipation," in *Proc. ICCD*, Washington, DC, 2005, pp. 7–12.

[19] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *Proc. 29th Ann. Int. Symp. Comput. Arch. (ISCA)*, Washington, DC, 2002, pp. 148–157.

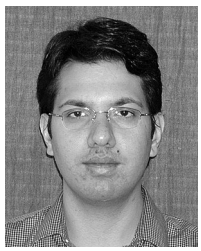
[20] J. A. Butts and G. S. Sohi, "A static power model for architects," in *Proc. Micro33*, 2000, pp. 191–201.
[21] Y. Tsai, D. Duarte, N. Vijaykrishnan, and M. Irwin, "Implications of technology scaling on leakage reduction techniques," 2004. [Online]. Available: citeseer.ist.psu.edu/tsai04implications.html
[22] D. Brooks and M. Martonosi, "Value-based clock gating and operation packing: dynamic strategies for improving processor power and performance," *ACM Trans. Comput. Syst.*, vol. 18, no. 2, pp. 89–126, 2000.
[23] C. Gopalakrishnan and S. Katkoori, "Resource allocation and binding approach for low leakage power," in *Proc. VLSID*, Washington, DC, 2003, p. 297.
[24] R. Mukherjee, S. O. Memik, and G. Memik, "Temperature-aware resource allocation and binding in high-level synthesis," in *Proc. DAC*, New York, 2005, pp. 196–201.
[25] H. S. Kim, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Adapting instruction level parallelism for optimizing leakage in vliw architectures," *SIGPLAN Not.*, vol. 38, no. 7, pp. 275–283, 2003.
[26] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Arch. Code Opt.*, vol. 1, no. 1, pp. 94–125, 2004.
[27] S. H. Gunther, F. Binns, D. M. Carmean, and J. C. Hall, "Managing the impact of increasing microprocessor power consumption," *Intel Technol. J.*, vol. 5, no. 1, Feb. 2001 [Online]. Available: http://www.intel.com/technology/itj/q12001/pdf/art_4.pdf
[28] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Proc. IEEE Workshop Workload Characterization*, 2001, pp. 3–14.
[29] "SPEC2000 benchmarks," 2000. [Online]. Available: www.spec.org/benchmarks/html
[30] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization," in *Proc. ISQED*, 2005, pp. 516–521.
[31] B. E. Stine, D. S. Boning, and J. E. Chung, "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE Trans. Semicond. Manuf.*, vol. 10, pp. 24–41, Feb. 1997.
[32] S. Bhardwaj, S. Vrudhula, P. Ghanta, and Y. Cao, "Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits," in *Proc. IEEE/ACM Des. Autom. Conf.*, 2006, pp. 791–796.
[33] M. Loève, *Probability Theory*. New York: Van Nostrand, 1960.
[34] Univ. California Berkeley Device Group, Berkeley, "BSIM 4.2.1 MOSFET model—User's manual," 2004.
[35] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, F. Sano, M. Norishima, and T. Sakurai, "A 0.9 V 150 MHz 10 row 4RAM2 2-D discrete cosine transform core processor with variable-thresholdvoltage scheme," *IEEE J. Solid-State Circuits*, vol. 31, no. 2, pp. 166–167, Feb. 1996.
[36] M. M. Griffin, J. Zerbe, G. Tsang, M. Ching, and C. L. Portmann, "A process-independent, 800-mb/s, dram byte-wide interface featuringcommand interleaving and concurrent memory operation," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1741–1751, Nov. 1998.
[37] Q. Chen, S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Process variation tolerant online current monitor for fault immune systems," in *Proc. Int. On-Line Test. Symp. (IOLTS)*, Jul. 2005.
[38] A. Srivastava, S. Aluri, and A. K. Chamakura, "A simple built-in current sensor for iddq testing of cmos data converters," *Integr., VLSI J.*, vol. 38, no. 4, pp. 579–596, 2005.
[39] M. Pertijs, A. Niederkorn, X. Ma, B. Mckillop, A. Bakker, and J. Huijsing, "A CMOS temperature sensor with a 3/spl sigma/ inaccuracy of /spl plusmn/0.1/spl deg/c from $-55/\mathrm{spl}\ \mathrm{deg}/\mathrm{c}$ to 125/spl deg/c," in *IEEE Int. Solid-State Circuits Conf. Dig. Techn. Papers ISSCC*, 2005, pp. 238–239.
[40] D. E. Duarte, G. Taylor, K. L. Wong, U. Mughal, and G. Geannopoulos, "Advanced thermal sensing circuit and test techniques used in a high performance 65 nm processor," in *Proc. Int. Symp. Low Power Electron. Des. (ISLPED)*, New York, 2007, pp. 304–309.
[41] Synopsys, Inc., "Synopsys design compiler," [Online]. Available: www.synopsys.com/products/logic/design-compiler.html

Aviral Shrivastava received the B.S. degree in computer science and engineering from Indian Institute of Technology, Delhi, India, and the M.S. and Ph.D. degrees in computer science and engineering from the University of California, Irvine.

He is an Assistant Professor with the Department of Computer Science and Engineering, Arizona State University, Tempe. His areas of interest lie at the intersection of compilers, computer architecture, and VLSI computer-aided design, with a particular focus on compiler, microarchitectural, and compiler microarchitecture hybrid techniques for improving power, performance, temperature, code size, reliability, and robustness of embedded and multicore processor systems.

Deepa Kannan received the B.E. (honors) in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India, in 2006 and the M.S. degree in electrical engineering from Arizona State University, Tempe, in 2008.

She is a CAD Engineer with the Technology and Manufacturing Group (TMG), Intel Corporation, Hillsboro, OR. She develops EDA tools that enables efficient analog circuit design for next generation Intel processors. Her research interests include low power analog circuit design, leakage power reduction in circuits, and process variation aware power reduction techniques for microprocessors.



Sarvesh Bhardwaj received the B.Tech. in electrical engineering from the Indian Institute of Technology, Delhi, India, in 2000, the M.S. degree in electrical and computer engineering and the Ph.D. degree in electrical engineering from the University of Arizona, Tucson, in 2003 and 2006, respectively.

He is a Senior Research and Development Engineer with the PrimeTime Group, Synopsys, Inc., Mountain View, CA. He has previously worked in various R&D roles in MindTree Technologies, Aplion Networks, SUN Microsystems, and Stratosphere Solutions. His research interests include statistical analysis and optimization of integrated circuits in the presence of process variations, logic synthesis and design for manufacturability.

Sarma Vrudhula is a Professor with the Department of Computer Science and Engineering, Arizona State University, Tempe. During 1985-1992, he was on the faculty of the EE-Systems Department, University of Southern California. From 1992 to 2005, he was a Professor with the Electrical and Computer Engineering Department, University of Arizona, Tucson, and the Director of the NSF UA/ASU Center for Low Power Electronics. During 2000-2001, he was a Visiting Scientist with the Advanced Design Tools group in Motorola, Austin. His work spans several areas in design automation and computer-aided design for digital integrated circuit and systems. These include logic synthesis, optimization and verification, power and performance optimization including analysis of process variations, low power design and thermal and energy management. He has served on the technical program committees of many national and international conferences in VLSI design automation and CAD, and on government review panels.

Dr. Vrudhula served as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and is presently an Associate Editor for the IEEE TRANSACTION ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the *ACM Transactions on Design Automation of Electronic Systems (TODAES)*.