Variation Trained Drowsy Cache (VTD-Cache): A History Trained Variation Aware Drowsy Cache for Fine Grain Voltage Scaling

Avesta Sasan, *Member, IEEE*, Kiarash Amiri, *Student Member, IEEE*, Houman Homayoun, *Member, IEEE*,
Ahmed M. Eltawil, *Member, IEEE*, and Fadi J. Kurdahi, *Fellow, IEEE*

***Abstract*—In this paper we present the "Variation Trained Drowsy Cache" (VTD-Cache) architecture. VTD-Cache allows for a significant reduction in power consumption while addressing reliability issues raised by memory cell process variability. By managing voltage scaling at a very fine granularity, each cache way can be sourced at a different voltage where the selection of voltage levels depends on both the vulnerability of the memory cells in that cache way to process variation and the likelihood of access to that cache location. After a short training period, the proposed architecture will micro-tune the cache, allowing significant power reduction with negligible increase in the number of misses. In addition, the proposed architecture actively monitors the access pattern and reconfigures the supply voltage setting to adapt to the execution pattern of the program. The novel and modular architecture of the VTD-Cache and its associated controller makes it easy to be implemented in memory compilers with a small area and power overhead. In a case study, the SimpleScalar simulation of the proposed 32 kB cache architecture reports over 57% reduction in power consumption over standard SPEC2000 integer benchmarks while incurring an area overhead of less than 4% and an execution time penalty smaller than 1%.**

***Index Terms*—Cache, drowsy cache, fault tolerance, leakage, low power, manufacturing defects, power efficient, process variation, static random access memory (SRAM), technology scaling, voltage scaling.**

I. Introduction

RECENT studies [1]–[3] suggest that static power consumption is on the verge of dominating dynamic power consumption for CMOS-based circuits. This phenomenon is more pronounced in static random access memory (SRAM) structures such as processor caches due to the usage of minimum sized devices and their dense structural arrangement. Furthermore, caches account for a large portion of power consumption in modern processors. Voltage scaling has long been considered as a winning solution to reduce both dynamic and leakage power consumption super linearly. However its application to memory structures requires excessive care due to reliability concerns which are exacerbated by process variation especially when operating at lower voltages. The introduced process variation widens the distribution in electrical characteristics of fabricated devices and reduces the reliability and predictability of their behavior [4]–[7]. Thus, the write and access time of the cache (memory structure) can be modeled as a Gaussian distribution, where voltage scaling shifts the mean of access/write time as well as the standard deviation of the distribution [7].

To maintain a high yield, the memory cycle time at a given voltage level is defined as the maximum of mean access time and mean write time plus a large guard band thus creating a three dimensional voltage-frequency-yield design space tradeoff. Fig. 1, depicts the results of a Monte Carlo simulation for a 6T SRAM cell under process variation in 32-nm technology (with standard deviation of 34 mV for the threshold voltage [19]). Fig. 1 illustrates the exponential growth in the probability of cell failure with a reduction in supplied voltage. In obtaining this curve, cycle time is kept constant (to that of used in higher voltage). Depending on the choice of cycle time, different probabilities of failure curves can be obtained.

Recently, there has been several research efforts aimed at designing circuits that are either adaptive to the less predictable behavior of their underlying devices [8], [9], or are statistically designed to improve the reliability of the system at one, or across a range of voltages [10], [11]. The underlying consensus of most research efforts is that process variation has reached a severity where it is becoming essential to rethink the memory architecture and organization (addressing the fabrication imposed reliability issue at the architecture level) to avoid excessive margining which has a drastic impact on both yield and power consumption.

In this paper, we propose a drowsy cache architecture that is aware of process variability within the structure. By using a simple and low cost distributed supply voltage management unit (one for each cache way) our proposed architecture allows a majority of the memory cells to operate at a reduced voltage. In this design, the existence of a cell that is severely affected by

Manuscript received February 12, 2010; revised June 22, 2010 and November 21, 2010; accepted December 15, 2010. This work was supported in part by the Center for Pervasive Communications and Computing at the University of California, Irvine, Samsung Electronics Corporation under Grant SEC-43124, by the National Science Foundation under Grant ECCS-0955157, and by the National Institute of Justice, Department of Justice under Grant 2006-IJ-CX-K044.

A. Sasan, K. Amiri, A. M. Eltawil, and F. J. Kurdahi are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92617 USA (e-mail: mmakhzan@uci.edu; kamiri@uci.edu; aeltawil@uci.edu; kurdahi@uci.edu).

H. Homayoun is with the Department of Computer Science, University of California, Irvine, CA 92617 USA (e-mail: hhomayou@uci.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2011.2106523

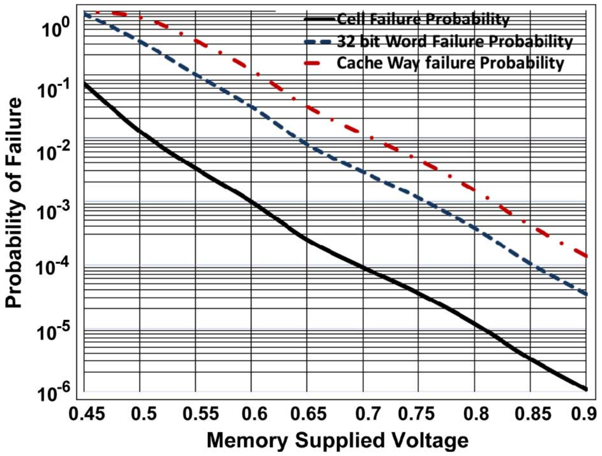


Fig. 1. Probability of cell, way, and cache failure in 32-nm technology. Way and cache figure are obtained for a case study, for a 32 kB four-way associative cache organizations.

process variation does not dictate a larger voltage to the entire cache but rather the higher voltage requirement is only mandated to the cache way(s) that contains the affected cell(s). This allows a majority of the cells in the cache to operate at a lower supply voltage while addressing the reliability concern raised by the "weaker" cells by shifting their statistics to a more favorable operating region via higher supply voltage. The proposed cache architecture not only guarantees safe and correct operation at lower voltages within the window of benchmark execution in the cache, but also explores the locality in the pattern of access to the cache in order to gain the largest improvement in energy saving by applying data retention voltage (DRV) to cold lines without considerably sacrificing the processor performance.

This paper is organized as follows. Section II explores prior work addressing design of low power and/or variation aware memory structures. Section III describes the proposed VTD-Cache. Section IV describes the simulation methodology and results. Section V identifies the major differences of VTD-Cache with Drowsy Cache [13]. Finally, Section VI concludes this paper.

II. Prior Work

As discussed previously, there has been a flurry of research activity to manage process variation and power consumption of memories in general and caches in particular. In [16] and [17], cache lines that are not recently accessed are power gated for reduction of static power consumption. When a gated line is accessed, it is charged back to nominal voltage, which requires charging all the internal capacitances of the memory cells in that cache line. Furthermore, the next level cache should be accessed to retrieve the information in those lines since by gating the supply voltage all the information in a gated line is lost.

In [8], adaptive body biasing with multi-threshold CMOS (ABB-MTCMOS) is used to change the threshold voltage of a cache line and reduce the static power consumption. This technique requires a more expensive processing step, as well as an increased area overhead for body biasing circuits. In [18], Sasan *et al.* presented the concept of the inquisitive defect cache (IDC), which is used as a small direct or associative cache that works in parallel with L1 cache and provides a defect free view of the cache for the processor in the current window of execution. This technique reduces the voltage on the entire cache and maps the faulty cache ways (parametric defects that are introduced after voltage scaling) which are recently accessed to a parallel small cache that operates at nominal voltage. An off chip defect map is also required when implementing this technique. Although the proposed architecture achieves considerable saving in power consumption, the associated area overhead is not negligible. A recent paper from Intel's microprocessor technology lab [19] suggests a form of fault tolerant mechanisms trading off the cache capacity and associativity in lower voltages versus masking process variation defects. The proposed approaches allow scaling the voltage from a nominal 0.9 V down to 0.5 V in a 65-nm technology, while the cache size is reduced by 75% or 50% depending on the fault tolerance mechanism used. This technique is used whenever the processor workload is low.

As will be described in this paper, the proposed VTD-Cache can be used both at nominal frequencies as well as at reduced workloads, while maintaining the maximum cache size at reduced power consumption. This is achieved via selective voltage supply for each cache way.

Finally, in [13] a Drowsy cache technique was proposed in which cache lines that have not recently been accessed (cold lines) are supplied with a low DRV in which the content of the cell stays intact, however the cell is not read/writeable. Such lines are referred to as drowsy lines. Upon accessing a drowsy line, the cache line has to wake-up before system could read or write it. The advantage of this technique compared to those in [16] and [17] is the fact that waking a drowsy line takes a much shorter time compared to accessing a lower level memory.

In this work, we extend the idea of drowsy cache and not only apply voltage scaling to cold lines by supplying them from a drowsy voltage; we also reduce the cache dynamic power consumption by selectively reducing voltage on recently accessed active lines. The voltage scaling of the recently accessed lines is performed with consideration of manufacturing process variability of memory cells in each cache way to assure correctness and yield.

III. Proposed Architecture: VTD-Cache

A. Concept

The VTD-Cache conceptually operates by allowing a fine grain control over the voltage of each cache way. Among available voltage levels the supplied voltage is chosen by a simple voltage selector that is implemented at each cache way. The voltage selector dynamically changes its state as the processor explores new segments of the running program and shifts and/or resizes its window of execution (WoE). In VTD-Cache each cache way can be supplied from one of three possible voltages. The lowest voltage level supplies DRV for cold lines which are determined by cache access pattern and are managed via the proposed architecture. Cache ways that are supplied with this voltage are referred to as "Cold Ways". The remaining two voltage levels are used in cache ways located within the Cache Window of Execution (CWoE). $V_{\mathrm{dd}}^{\mathrm{Low}}$ is supplied if cache way could operate correctly in that voltage, otherwise cache way is

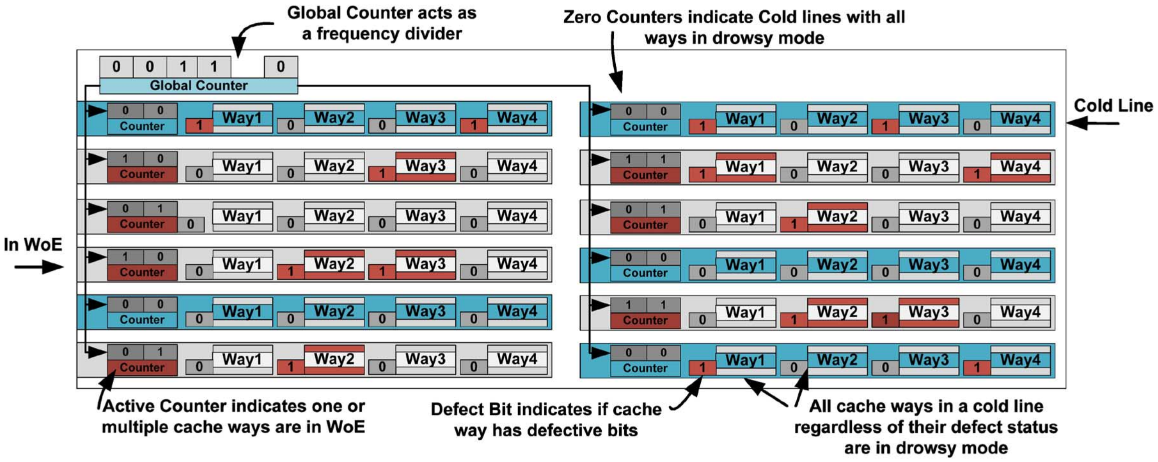

Global Counter acts as a frequency divider
Zero Counters indicate Cold lines with all ways in drowsy mode
Global Counter
Cold Line
In WoE
Counter
Way1
Way2
Way3
Way4
Active Counter indicates one or multiple cache ways are in WoE
Defect Bit indicates if cache way has defective bits
All cache ways in a cold line regardless of their defect status are in drowsy mode

Fig. 2. Top level view of the VTD-Cache, sets with non-zero set counters take part in WoE for which ways with defect bit set to 1 are on high voltage and the rest on low voltage.

supplied with $V_{\rm dd}^{\rm High}$. We refer to such cache ways as "Warm" and "Hot" Ways if they are supplied from $V_{\rm dd}^{\rm Low}$ or $V_{\rm dd}^{\rm High}$ accordingly. The decision to use which supply voltage is made based on a defect map that is generated using memory built-in self test (BIST). Section III-F further elaborates on how the Warm and Hot cache ways are redefined based on change in the operational temperature.

Fig. 2 illustrates the general idea of this architecture. In this Figure, each line represents a set that consists of four ways. Each set has its own set voltage selector (SVS) which contains a dedicated N bit counter (with $N = 1$ being the simplest realization and equal to what was proposed in the drowsy cache suggested in [13]). Upon access to a cache way in that set, the set is identified as being in the CWoE. This is being done with setting the countdown counter to a nonzero value. Each cache way has its own simple way voltage selector (WVS), which is linked to a defect bit that indicates whether that specific way contains defective bits or not. If the SVS counter reaches zero, all WVSs that are associated with that SVS automatically shift the state of their cache way to data retention (Drowsy) mode. Otherwise based on their internal defect map bit, they supply their associated cache way from either $V_{\rm dd}^{\rm High}$ or $V_{\rm dd}^{\rm Low}$. As suggested by Fig. 2, all cache ways within WoE (which are not cold/drowsy lines) and have their defect map bit set to one are sourced from higher voltage and those with defect map bit of zero are supplied from $V_{\rm dd}^{\rm Low}$. A Set enters the CwoE by an access to the set and exits the CWoE when its associated counter reaches 0. The set counter counts down when a count down signal (CDS) is send from global counter. The global counter acts as a cache access frequency divider and is shared by all sets. It is a cyclic counter that counts down and upon reaching 0 while being reset to its high value generates the CDS-signal that is fed to all SVSs.

B. Implementation

Fig. 3 suggests a simple implementation for the SVS. SVS contains an internal N bit counter which counts down every time CDS signal is triggered. Upon insertion of "Defect Update Mode" and "Wordline_in" the output signal "Defect Wordline" is set. This is done when the system intends to update the defect bit implemented at each WVS. On the other hand, if the defect update mode line is not set and only the input signal "Wordline_in" is set, the output "Wordline_out" signal will be inserted while the "Defect Wordline" output is kept low. The "Active Counter" signal is sent to each WVS giving a feedback on SVS counter status.

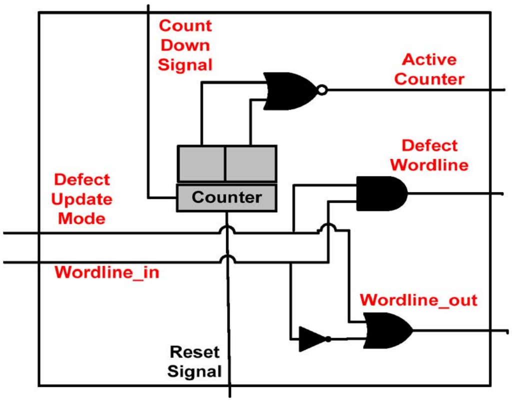


Fig. 3. Cache SVS.

WVS is shown in Fig. 4. It contains an internal memory bit referred to as fault tolerant bit (FT-Bit). FT-Bit is set if the cache way contains bit(s) that are severely affected by process variation which are unable to operate at $V_{\rm dd}^{\rm Low}$. The implemented FT-Bit is made more tolerant to process variation either by upsizing the basic 6T cell, or by using a Schmitt Trigger Cell [12] and is updated after running a BIST in low voltage and is written by using the same mechanism as we write to the other SRAM cells using dedicated "Defect Bitline" pair. Defect Bit wordline is set using the "Defect Wordline" input from SVS.

Combination of SVS's counter and WVS's FT-Bit create the state machine illustrated in Fig. 5 (a 2 bit SVS counter is assumed in this figure) which manages the choice of voltage that is supplied to the cells and the wordline in each cache way. In order to avoid bit stability issues when reading the cell content

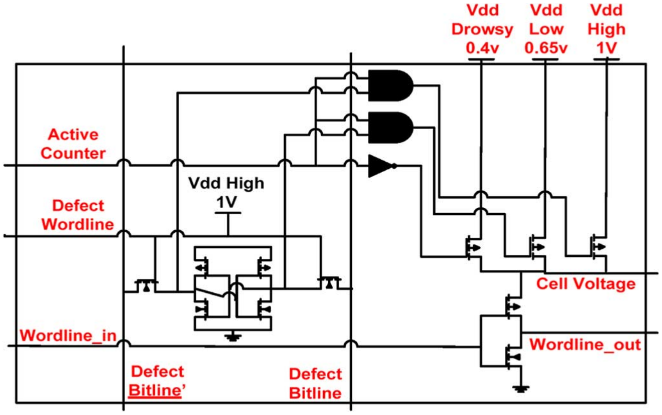


Fig. 4. Cache WVS.

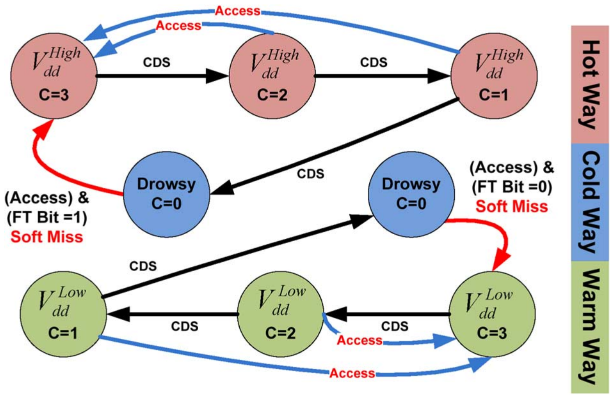


Fig. 5. State machine controlling the voltage at each cache way (CDS: Count Down Signal; C: Counter).

the wordline "Wordline_out" also uses the same voltage as that is being supplied to the cell.

Based on this state machine a cache way in any given set is kept in drowsy mode until an access to a cache way in that set is made. Subsequently based on FT-Bit state the voltage is switched to $V_{\mathrm{dd}}^{\mathrm{Low}}$ or $V_{\mathrm{dd}}^{\mathrm{High}}$. Each time the count-down-signal (CDS) from the global counter is sent, the state changes until finally the drowsy state is reached. An access to a cache way in that set resets the SVS counter and returns the state to "$C = 3$" state. If a set is accessed while it is drowsy/cold, the access cannot be granted in the same cycle. Based on the size of the cache ways, the size of the transistors that are feeding the cell voltage, and the state to which the cache way is switching ($V_{\mathrm{dd}}^{\mathrm{Low}}$ or $V_{\mathrm{dd}}^{\mathrm{High}}$) one or more transition cycles are required, subsequently the access is repeated. This is referred to as "soft-miss". The soft-miss penalty for transition to $V_{\mathrm{dd}}^{\mathrm{High}}$ and $V_{\mathrm{dd}}^{\mathrm{Low}}$ could be different since all cell capacitances need to charge up to different voltages, however this timing could also be controlled by sizing the pMOS transistor sourcing the $V_{\mathrm{dd}}^{\mathrm{High}}$ larger than the transistor sourcing the $V_{\mathrm{dd}}^{\mathrm{Low}}$.

As shown in Fig. 6 and mentioned previously, the local counters are updated by a feedback from a global mechanism. The feedback in the proposed system is implemented using a global access counter (or frequency divider). The global counter logically extends the LSBs of all local counters. This mechanism allows VTD-Cache to estimate the window of execution with a much smaller overhead compared to implementing a full counter for each set. The length of the global counter determines the frequency of updates to the set counters. The global counter is a cyclic counter that sends a signal to local counters every time it reaches the 0 state.

In Fig. 7, a simple implementation of VTD-CacheTAG comparators is suggested. This comparator in addition to detecting cache hits and misses is capable of detecting soft-misses. It also generates the reset signal feedback that is sent to the SVS counters.

C. Exploring the Locality in the Cache

The mechanism that is illustrated so far provides strong support for temporal locality in the cache allowing the cache ways that have been recently accessed to be accessed again without incurring a soft-miss at the lowest readable dynamic power consumption. However, the cache could also be altered to add support for spatial locality. The support for spatial locality could be added by providing a mechanism by which, upon access to a cold set, the counter of the next set is also initialized. However, the next set counter value is not set to the max value, but to min value, making sure that if it is not accessed in the next few cycles, it changes state back to the drowsy mode. Adding spatial locality increments both dynamic and leakage power but reduces the number of soft-misses and therefore execution time. Support for spatial locality is useful for reducing the soft-misses that are encountered when the CWoE moves to a new region, therefore reducing the soft-misses sharply. Improvement in the program execution time by eliminating a large portion of the soft errors could vary widely depending on the benchmark. Since most programs have high locality, the number of soft-misses are usually low and adding this policy might even negatively impact the energy-delay product by increasing the total dynamic and leakage energy. On the other hand, adoption of this policy is useful if the expected program tends to have very low locality and quickly moves its WoE.

D. Accuracy of Prediction of Cache WoE

Upon access to a set, its SVS counter is set. At this time the global counter could have any value. Therefore the accuracy of the counter (with SVS counter at MSB and global counter at LSB) is only controlled by the initial value of the SVS counter while the global counter introduces uniform randomness in LSBs. Because it is shared, the global counter introduces a negligible area overhead while the SVS counters' area overhead (repeated for every set) could be significant. Choosing the right "split point" between the two slices is therefore a tradeoff between accurately estimating the CWoE and area overhead. In VTD-Cache this trade off is explored by carefully sizing the local and global counters. Having "m" local bits, the inaccuracy in determining the CWoE is obtained from $1/2^m$ meaning the starting point of the extended counter could range from $[2^{m+n} - 1, 2^{m+n} - 2^n - 1]$. For example in an architecture with a 2-bit local and 7-bit global counter the logical counter upon access could be set to a max value in the range of $[2^{2+7} - 1, 2^{2+7} - 2^7 - 1]$ or [511, 383].

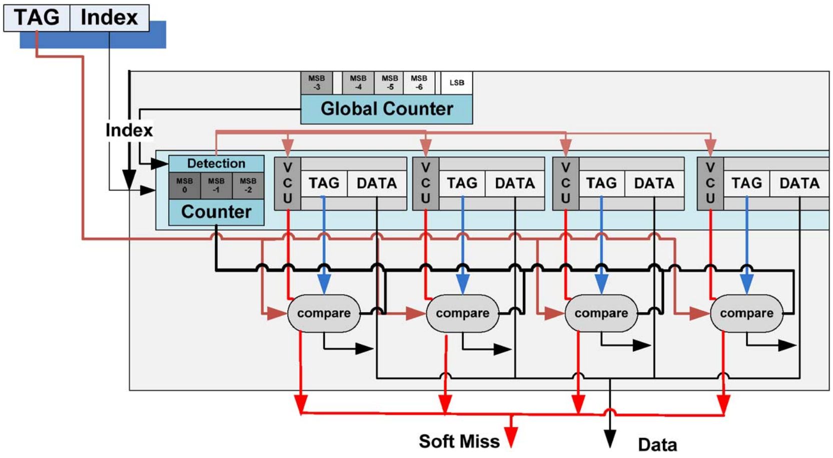


Fig. 6. Each comparator decision depends on local counter state which changes by a global counter CDS.

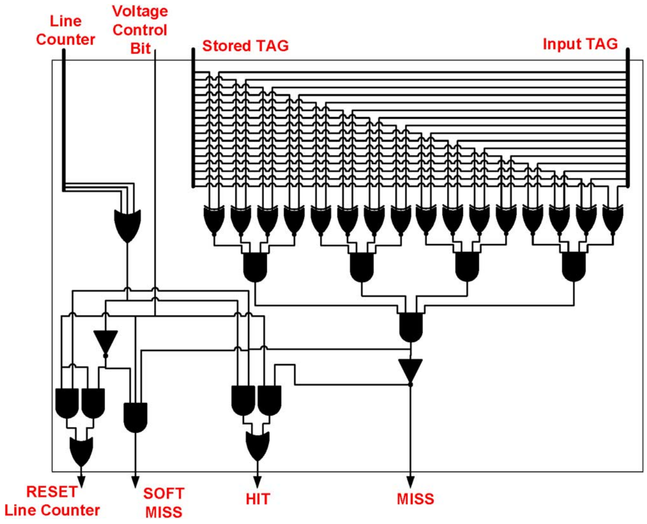


Fig. 7. Comparator structure in VTD-Cache with ability to detect soft-miss, hit, miss, and generating reset signal for set counter.

E. Finding the Suitable Width of Local and Global Counters

As mentioned previously the accuracy in the estimation of CWoE is attained by the size of the local counters. Since the local counters are repeated for every set, adopting a large local counter results in a larger area overhead. The more accurate the CWoE is estimated, the higher the energy savings are, albeit, at a cost in area.

The optimal sizes of the local and global counters are dependent on many factors such as mapping of the voltage to the probability of failure (which depends, among others on technology node and fabrication precision), the size of the cache, properties of executing benchmarks, organization of the cache, etc. Following is a case study for finding the choice of local and global counters for a 32 kB, four-way associative L1 data cache arranged in two banks that results in minimum energy consumption over SPEC2000 benchmarks. Each cache way contains four words. The mapping of voltage to probability of cell failures is given in Fig. 1. The final energy improvement is calculated based on (1), where each term used is defined in Table I. The dynamic and static energy consumption of the VTD-Cache and conventional cache are obtained from SPICE simulation post layout netlist of these caches. In addition, we also need information on type, number, and nature of accesses to the cache. This is obtained using SimpleScalar [14] simulation after we modified SimpleScalar to model the VTD-Cache. For simplicity in this model we assumed that the TAGs are supplied from $V_{\mathrm{dd}}^{\mathrm{High}}$.

The energy improvement metric is thus calculated as follows:

$$\%\ \text{Improvement} = \left[1 - \frac{E_{\mathrm{total}}^{\mathrm{VTD}}}{E_{\mathrm{Total}}^{\mathrm{Conv}}}\right] \times 100 \tag{1}$$

$$E_{\mathrm{total}}^{\mathrm{VTD}} = E_{\mathrm{dynamic}}^{\mathrm{VTD}} + E_{\mathrm{static}}^{\mathrm{VTD}} \tag{2}$$

$$E_{\mathrm{dynamic}}^{\mathrm{VTD}} = E_{\mathrm{Peripheral}}^{\mathrm{VTD-dyn}} + E_{\mathrm{Tags}}^{\mathrm{VTD-dyn}} + E_{\mathrm{Ways}}^{\mathrm{VTD-dyn}} \tag{3}$$

$$\begin{aligned} E_{\mathrm{Tags}}^{\mathrm{VTD-dyn}} = {} & E_{\mathrm{Tags-Read}}^{\mathrm{VTD-dyn}} \\ & .\left[N_{\mathrm{read}}^{\mathrm{High}} + N_{\mathrm{read}}^{\mathrm{Low}} + N_{\mathrm{Read}}^{\mathrm{Soft-Miss}}\right] \\ & + E_{\mathrm{Tags-Write}}^{\mathrm{VTD-dyn}} \\ & .\left[N_{\mathrm{Write}}^{\mathrm{High}} + N_{\mathrm{Write}}^{\mathrm{Low}} + N_{\mathrm{write}}^{\mathrm{Soft-Miss}}\right] \end{aligned} \tag{4}$$

$$\begin{aligned} E_{\mathrm{Ways}}^{\mathrm{VTD-dyn}} = {} & E_{\mathrm{Ways-Read}}^{\mathrm{VTD-dyn}} \\ & .\left[N_{\mathrm{read}}^{\mathrm{High}} + N_{\mathrm{read}}^{\mathrm{Low}} \times \left(\frac{\mathrm{Vdd_{low}}}{\mathrm{Vdd_{high}}}\right)^2\right] \\ & + E_{\mathrm{Ways-Write}}^{\mathrm{VTD-dyn}} \\ & .\left[N_{\mathrm{Write}}^{\mathrm{High}} + N_{\mathrm{Write}}^{\mathrm{Low}} \times \left(\frac{\mathrm{Vdd_{low}}}{\mathrm{Vdd_{high}}}\right)^2\right] \\ & + E_{\mathrm{Transition}}^{\mathrm{VTD-dyn}} \end{aligned} \tag{5}$$

$$\begin{aligned} E_{\mathrm{Transition}}^{\mathrm{VTD-dyn}} = {} & \mathrm{NT}_{\mathrm{Drowsy}}^{\mathrm{Low}} . E_{\mathrm{Tran\_to\_low\_voltage}}^{\mathrm{VTD-dyn}} \\ & + \mathrm{NT}_{\mathrm{Drowsy}}^{\mathrm{High}} . E_{\mathrm{Trans\_to\_low\_voltage}}^{\mathrm{VTD-dyn}} \end{aligned} \tag{6}$$

$$\begin{aligned} E_{\mathrm{Peripheral}}^{\mathrm{VTD-dyn}} = {} & E_{\mathrm{Peripheral-read}}^{\mathrm{VTD-dyn}} \\ & \times \left(N_{\mathrm{read}}^{\mathrm{High}} + N_{\mathrm{read}}^{\mathrm{Low}} + \theta . N_{\mathrm{read}}^{\mathrm{Soft-Miss}}\right) \\ & + \left(N_{\mathrm{write}}^{\mathrm{High}} + N_{\mathrm{write}}^{\mathrm{Low}} + \gamma . N_{\mathrm{write}}^{\mathrm{Soft-Miss}}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} E_{\mathrm{Static}}^{\mathrm{VTD}} = {} & E_{\mathrm{Peripheral}}^{\mathrm{VTD-static}} + E_{\mathrm{Ways}}^{\mathrm{VTD-static}} \\ & + E_{\mathrm{Tags}}^{\mathrm{VTD-static}} \\ = {} & N_{\mathrm{execution}}^{\mathrm{Cycle}} . \frac{1}{f} \end{aligned}$$

$$\cdot \left[P_{\text{Peripheral}}^{\text{VTD}-\text{static}} + P_{\text{Ways}}^{\text{VTD}-\text{static}} + P_{\text{Tags}}^{\text{VTD}-\text{static}}\right] \tag{8}$$

$$\begin{aligned} P_{\text{Ways}}^{\text{VTD}-\text{static}} = {} & P_{\text{Ways}-\text{VDD}_{\text{High}}}^{\text{VTD}-\text{static}} \\ & \times [\text{AVG}_{\text{High}} + \text{AVG}_{\text{Drowy}} \\ & \times e^{b(\text{Vdd}_{\text{Drowsy}} - \text{Vdd}_{\text{high}})} \\ & + (N_{\text{wys}} - \text{AVG}_{\text{High}} + \text{AVG}_{\text{Drowy}}) \\ & \times e^{e(\text{Vdd}_{\text{low}} - \text{Vdd}_{\text{high}})}\Big] \end{aligned} \tag{9}$$

$$E_{\text{Total}}^{\text{Conv}} = E_{\text{dynamic}}^{\text{Conv}} + E_{\text{static}}^{\text{Conv}} \tag{10}$$

$$E_{\text{static}}^{\text{Conv}} = N_{\text{execution}}^{\text{Cycle}} \cdot \frac{1}{f} . P_{\text{static}}^{\text{Conv}} \tag{11}$$

$$\begin{aligned} E_{\text{dynamic}}^{\text{Conv}} = {} & N_{\text{Read}} . E_{\text{Read}}^{\text{Conv}-\text{dyn}} \\ & + N_{\text{Write}} . E_{\text{Write}}^{\text{Conv}-\text{dyn}} . \end{aligned} \tag{12}$$

In addition to the right split point of the local and global counters it is important to choose the $V_{\text{dd}}^{\text{Low}}$ such that the improvement in total energy consumption is maximized. $V_{\text{dd}}^{\text{Low}}$ should be chosen such that in that voltage most of the cache ways within the active lines (CWoE) are still readable. Choosing an inappropriately low $V_{\text{dd}}^{\text{Low}}$ results in the following three negative effects: 1) increase in the number of ways within the WoE that are supplied with higher voltage due to increase in the cell failure probability; 2) increase in energy required for transition of weak ways from drowsy to high voltage; 3) increase in the execution time due to an increase in the soft-misses associated with a slower transition from drowsy to high voltage modes. On the other hand, if $V_{\text{dd}}^{\text{Low}}$ is chosen to be inappropriately large, the cache consumes higher dynamic power. In this case, the number of cache way that are supplied from $V_{\text{dd}}^{\text{High}}$ is reduced, but we have to supply all the other healthy cache ways from a higher $V_{\text{dd}}^{\text{Low}}$.

F. Defect Map, BIST, and Temperature Variation

Generation and update of the defect map in VTD-Cache is essential for the proper operation of this non-conventional memory structure. With each operational setting (voltage, temperature, and frequency) the defect map changes changing the number of warm and hot cache ways. Although one could use a worse case defect map for safe operation of the VTD-Cache across all voltages (by running the BIST at low voltage and highest temperature), such approach results in waste of power during operation in regions where the operational setting is not close to the worse case by increasing the number of hot ways and overall power consumption.

Almost all modern processors today are equipped with digital temperature sensors (DTS). The temperature and voltage setting of the VTD-Cache is explicitly controlled by its wrapper structure. Usage of DTS reveals the value of the last remaining operational variable allowing usage of operational region dedicated defect map rather than a worse case defect map. The Generation, Update, and witching between defect maps with consideration for temperature variation is done as follows.

Step 1) After Manufacturing and during functional testing the cache is stress tested for the highest possible temperature. In this temperature BIST tests the memory cells when $V_{\text{DD}}^{\text{High}}$ is supplied to determine the manufacturing defects and process variation defects that still malfunction in the highest voltage and highest temperature. These defects are redirected to available redundancy.

Step 2) The stress test is repeated for $V_{\text{DD}}^{\text{Low}}$ and the worse case defect map for the VTD-Cache is generated. Since the number of weak cells in the lower temperatures super linearly reduces, the generated defect map is not a proper defect map for operation in lower voltages, but necessary in obtaining those; Although the generated defect map works in the $V_{\text{DD}}^{\text{Low}}$, its usage causes some cache ways capable of operation $V_{\text{DD}}^{\text{Low}}$ to be sourced from the $V_{\text{DD}}^{\text{High}}$. To reduce the chip cost, the previous two steps are the only required tests during the manufacturing.

Step 3) At the first boot of the system, VTD-Cache is loaded with the worse case defect map populated at manufacturing (Step 2) for the highest voltage and temperature.

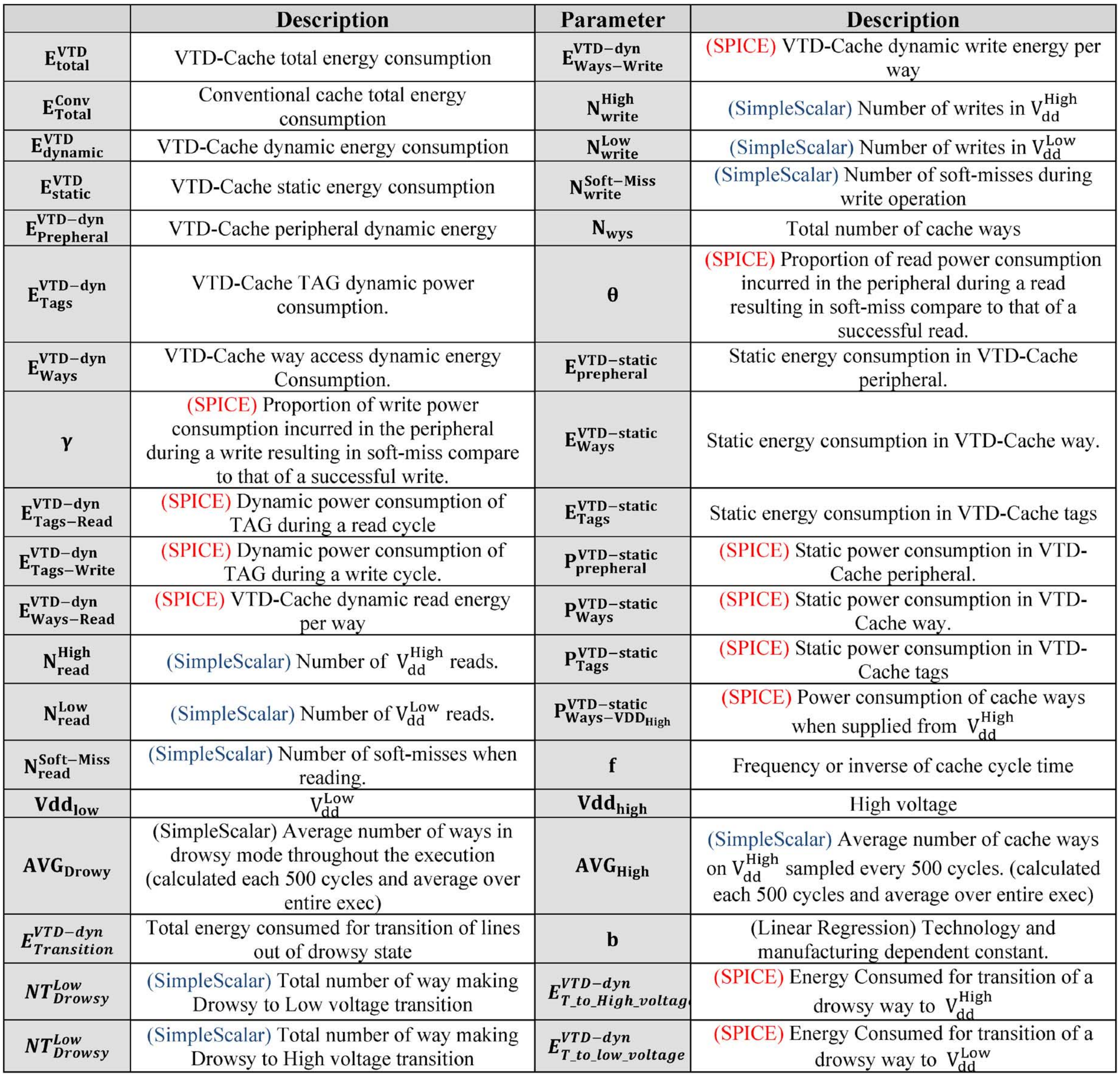
Step 4) The range of temperature variation is divided into different regions (each region covering a range of temperatures) and BIST will generate a defect map for each region. When temperature passes a region boundary that has not yet owns its dedicated defect map, the BIST is executed and a new defect map is generated. This strategy relaxes the defect map constrain such that instead of all temperatures worse case defect map, the worse defect map of each temperature regions (the one populated at the higher boundary) is used for operation in that region redefining some of the hot ways in the worse case defect map as warm ways and safely reducing their associated power consumption. The first time that the chip temperature reaches the region of the highest temperature, the VTD-Cache BIST has populated a defect map for every region of operation between the startup temperature and the chip worse case temperature.

The populated defects maps are stored in non-volatile memory (H.D.D). Each time that the temperature enters a new boundary (passes boundary + some safety margin) the cache defect map is updated with that of the new region. Note that loading a new defect map, due to large latency of lower level memories, could introduce large latency. However the temperature is very slow changing variable and the rate of temperature dependent defect-map updates could be very infrequent. In addition, a processor temperature follows its workload; usually after change in the processor workload, due to larger or lower activity the temperature changes and then reaches the equilibrium and stays in that range. While no major change in the processor workload, no major temperature change is expected. This reduces the number of necessary defect map updates. In addition the designer could choose larger ranges of temperature for each defect map to limit the number of defect map updates due to temperature variation.

The VTD-Cache is also protected by single bit parity against soft errors. If cache way single bit parity signals a reoccurring

TABLE I

DEFINITION OF TERMS USED IN EQUATIONS (1)–(11). IN DESCRIPTION OF EACH PARAMETERS WE HAVE REFERRED TO THE SOURCE OF DATA WHETHER IT IS FROM SPICE NETLIST SIMULATION (SPICE) OR FROM SIMPLE SCALAR SIMULATION (SIMPLESCALAR)



	Description	Parameter	Description
E^{VTD}_{total}	VTD-Cache total energy consumption	$E^{VTD-dyn}_{Ways-Write}$	(SPICE) VTD-Cache dynamic write energy per way
E^{Conv}_{Total}	Conventional cache total energy consumption	N^{High}_{write}	(SimpleScalar) Number of writes in V^{High}_{dd}
$E^{VTD}_{dynamic}$	VTD-Cache dynamic energy consumption	N^{Low}_{write}	(SimpleScalar) Number of writes in V^{Low}_{dd}
E^{VTD}_{static}	VTD-Cache static energy consumption	$N^{Soft-Miss}_{write}$	(SimpleScalar) Number of soft-misses during write operation
$E^{VTD-dyn}_{Prepheral}$	VTD-Cache peripheral dynamic energy	N_{wys}	Total number of cache ways
$E^{VTD-dyn}_{Tags}$	VTD-Cache TAG dynamic power consumption.	θ	(SPICE) Proportion of read power consumption incurred in the peripheral during a read resulting in soft-miss compare to that of a successful read.
$E^{VTD-dyn}_{Ways}$	VTD-Cache way access dynamic energy Consumption.	$E^{VTD-static}_{prepheral}$	Static energy consumption in VTD-Cache peripheral.
γ	(SPICE) Proportion of write power consumption incurred in the peripheral during a write resulting in soft-miss compare to that of a successful write.	$E^{VTD-static}_{Ways}$	Static energy consumption in VTD-Cache way.
$E^{VTD-dyn}_{Tags-Read}$	(SPICE) Dynamic power consumption of TAG during a read cycle	$E^{VTD-static}_{Tags}$	Static energy consumption in VTD-Cache tags
$E^{VTD-dyn}_{Tags-Write}$	(SPICE) Dynamic power consumption of TAG during a write cycle.	$P^{VTD-static}_{prepheral}$	(SPICE) Static power consumption in VTD-Cache peripheral.
$E^{VTD-dyn}_{Ways-Read}$	(SPICE) VTD-Cache dynamic read energy per way	$P^{VTD-static}_{Ways}$	(SPICE) Static power consumption in VTD-Cache way.
N^{High}_{read}	(SimpleScalar) Number of V^{High}_{dd} reads.	$P^{VTD-static}_{Tags}$	(SPICE) Static power consumption in VTD-Cache tags
N^{Low}_{read}	(SimpleScalar) Number of V^{Low}_{dd} reads.	$P^{VTD-static}_{Ways-VDD_{High}}$	(SPICE) Power consumption of cache ways when supplied from V^{High}_{dd}
$N^{Soft-Miss}_{read}$	(SimpleScalar) Number of soft-misses when reading.	f	Frequency or inverse of cache cycle time
Vdd_{low}	V^{Low}_{dd}	Vdd_{high}	High voltage
AVG_{Drowy}	(SimpleScalar) Average number of ways in drowsy mode throughout the execution (calculated each 500 cycles and average over entire exec)	AVG_{High}	(SimpleScalar) Average number of cache ways on V^{High}_{dd} sampled every 500 cycles. (calculated each 500 cycles and average over entire exec)
$E^{VTD-dyn}_{Transition}$	Total energy consumed for transition of lines out of drowsy state	b	(Linear Regression) Technology and manufacturing dependent constant.
NT^{Low}_{Drowsy}	(SimpleScalar) Total number of way making Drowsy to Low voltage transition	$E^{VTD-dyn}_{T\_to\_High\_voltag}$	(SPICE) Energy Consumed for transition of a drowsy way to V^{High}_{dd}
NT^{Low}_{Drowsy}	(SimpleScalar) Total number of way making Drowsy to High voltage transition	$E^{VTD-dyn}_{T\_to\_low\_voltage}$	(SPICE) Energy Consumed for transition of a drowsy way to V^{Low}_{dd}

error (error does not go away so it is not a soft error) the BIST for that operational region is executed updating the defect map. The occurrence of an aging defect requires the update of all the other defect maps as well. An aging defect at a given temperature will exist in all higher temperatures. Therefore the defect map of those regions could be updated without the need of rerunning the BIST. However for the lower temperatures due to improved transistor characteristics at lower voltages, the aging defect might be corrected therefore the defect map is not updated until the temperature enters the lower regions and the memory is tested. When testing the BIST only needs to test the memory at the aged location (and possibly close locations for the coupling and neighborhood pattern errors) The operating system sets a flag and register the location of the aging defect so that whenever the temperature reduces and enters a new region, the BIST tests the memory location and updates the associated defect map.

IV. AREA OVERHEAD

Compared to a conventional Cache, VTD-Cache introduces the following four new entities each contributing to the overall area overhead:

1) WVS which is repeated for each cache-way;
2) SVS which is shared among cache ways in each set but repeated for each set;
3) comparators which is shared for each bitline;
4) global counter (GC) which is shared among all the sets in the cache.

The size of PULL up transistors in each way voltage control unit also plays a role in determining the area overhead of the VTD-Cache. The larger the size of this transistor the faster transition of a drowsy line to a high or low voltage occurs and therefore the execution time is less affected. However, a larger size increases the area overhead. In addition to the previous mechanisms, realization of multiple supply voltages also incurs extra routing overhead and complexity.

TABLE II
BREAKDOWN OF THE AREA OVERHEAD OF A 32 kB VTD CACHE, TWO BANK, FOUR WAYS, AND (1,2) TRANSITION PENALTY FROM DROWSY VOLTAGE TO $V_{\rm dd}^{\rm Low}$ AND $V_{\rm dd}^{\rm High}$, RESPECTIVELY

Entity	Contribution to area overhead
WVS	44%T
SVS	37%
Enhanced Comparator	8%
GC	4%
Enlarged Write Circuitry	7%

TABLE III
SIMPLESCALAR CONFIGURATION

Parameter	Value
ROB size	**256**
Register File Size	**256 FP, 256 INT**
Fetch/schedule/retire/width	**6/5/5**
Scheduling Window Size	**32FP, 32 Int, 32 Mem**
Memory Disambiguation	**Perfect**
Load/Store Buffer Size	**32/32**
Branch Predictor	**16KB**
Cache Line Size	**64 Byte**
L1 Data Cache Size	**32 KB, 4Way, 1 Cycles**
L1 Instruction Cache Size	**32 KB, 4Way, 1 Cycles**
Execution Length	**1B Fast Forward, 1B execution**
L2 Unified Cache	**2MB, 8Way, 6 Cycles**
Penalty of waking a drowsy line to V_{dd}^{Low}	**1 cycle**
Penalty of waking a drowsy line to V_{dd}^{Low}	**2 cycle**

To reduce the area overhead in designing the WVS of VTD-Cache the N-well of pull-up transistors are shared and is pinned to the highest voltage. As a negative side effect, the sharing of N-well reduces the drive power of pMOS transistors in the lower voltages. Our simulation shows negligible effect on read timing and failure probability. The write operation however due to higher dependency on pMOS transistor drive power is negatively affected. In order to offset the side effect, the write circuit drive power is increased by widening its size (∼10% increase). The increase in the area of the write circuit compared to reduction in the size of the cache by sharing the N-wells in WVSs is negligible. In the Drowsy mode, there is no read or write operation and therefore as far as the drive power of the pMOS is considerably greater than nMOS leakage the design is valid. Another consideration in the layout of the VTD cache is the number of metal layers available. Based on design methodology for supporting three voltage levels 1 (in case of interleaved power mesh grid) or 2 (in case of uniform power mesh grid) extra metal layers are required. Considering that the traditional cache could be realized in as little as four metal layers, VTD cache requires a design that allows usage of lower 5 to 6 metal layers for the cache (depending on usage of interleaved or uniform power grids). This is only possible in processes that offer large number of metal layers. Considering that almost all processors today are fabricated in fabs that offer up to 11 metal layers [29] in 32-nm technology assigning 5 or 6 metal layers to the cache structure is acceptable. Table II summarizes the breakdown of the area overhead of 32 kB VTD cache with 2 bit local and 7 bit global area overhead that realizes one cycle wake up latency from low drowsy to $V_{\rm dd}^{\rm Low}$ and two cycle latency when transitioning from $V_{\rm dd}^{\rm Drowsy}$ to $V_{\rm dd}^{\rm High}$. Note that the total area overhead for this realization is 3.98%.

V. RESULTS

In order to investigate the right split point of the local and global counters and optimal $V_{\rm dd}^{\rm Low}$ we simulated the VTD-Cache architecture for different voltages (probability of failures) and for different combination of local and global counters. The local counter was varied from 1 to 3 bits and the global counter from 4 to 13 bits and finally the voltage is varied from 1.1 to 0.6 V. In this simulation based on the probability of memory failure in Fig. 1, at each voltage point a cache way failure probability was obtained. Based on that probability, defective cache ways were uniformly and randomly distributed in the cache. The simulation as well as SimpleScalar configuration used is shown in Table III. Integer benchmarks were executed to extract the parameters needed for (1)–(12) for each benchmark from over one billion executed instructions after one billion fast forwarding. In order to make sure that we are not running to special corner cases, the simulation was repeated 30 times for each benchmark, each time using a different seed for distribution of the faulty cache ways (thus generating different defect maps). At each voltage the simulation is repeated for different choices of global and local counters. Based on the parameters that are extracted from each run, improvement in total energy [based on (1)] was obtained. Then the improvement index for each pair of local and global counter setting was averaged over all benchmarks and all runs. Fig. 8 illustrates the obtained average improvement index for various combinations of local and global counters at each voltage.

In Fig. 8 at each voltage point for each combination of local and global counter, a bar represents the total energy if that setting is realized. Each bar is divided into a “Dual Voltage” and a “Triple Voltage” segments. The Dual-Voltage segment indicate the extent of power saving if only two voltage levels $(V_{\rm dd}^{\rm Drowsy}, V_{\rm dd}^{\rm High})$ where to be used and the “Tripple-Voltage” segment illustrates how VTD cache increases the power saving by selectively allowing some ways to operate in the lower voltage and therefore reducing the dynamic power consumption.

Fig. 8 suggests that for the given cache organization, at 0.7 V the power saving is maximized. Based on Fig. 8 if a 5 bit global counter and 3 bit local counters are used, the maximum power saving is obtained.

Results illustrated in Fig. 8 are averaged across different benchmarks over long execution period. Our studies show that some benchmarks could achieve a larger amount of power saving if their logical counter was defined differently. In addition, each program during execution time goes trough different phases changing the size of its WoE and the access pattern to instruction and data caches. This suggests the possibility of dynamically reconfiguring the number of global bits and the initial value of the local bit for maximizing the power savings. These extended features will be addressed in future work.

In order to further quantify the power savings of VTD-Cache, a case study was constructed for a 32 kB data-cache. The energy

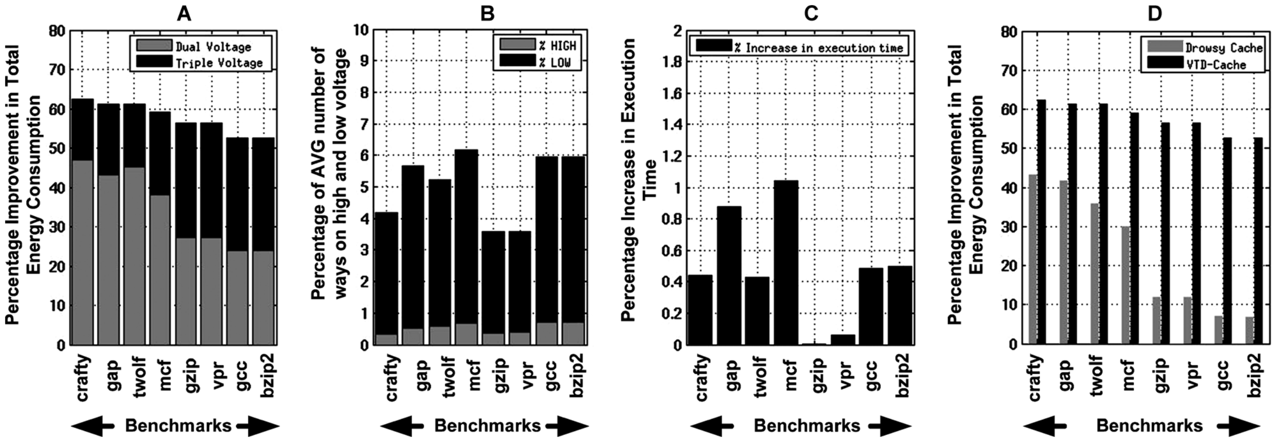


Fig. 9. (a) Improvement in energy consumption of selected SPEC2000 benchmarks. (b) Percentage of drowsy, High, and Low voltage ways over simulation sampled every 10 000 cycles. (c) Increase in the execution time due to soft-misses. (d) Comparing the Improvement in total energy consumption when compared with Drowsy cache in [13].

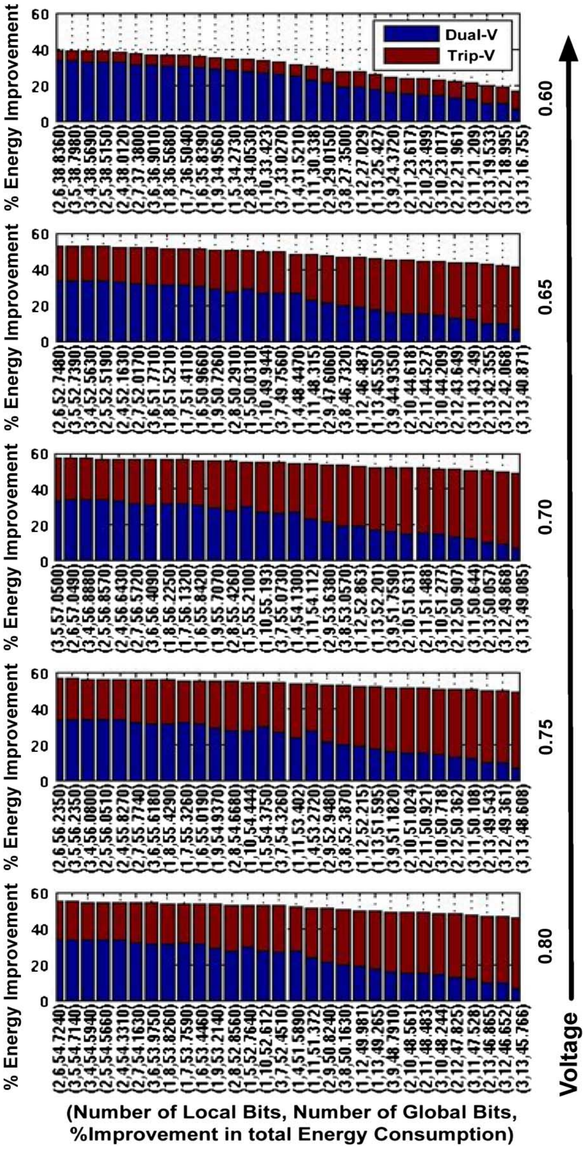


Fig. 8. Percentage improvement in total energy consumption over the program execution for different setting of local and global counters.

consumption of different access types (write, read, soft-miss, miss, write miss) was extracted from a SPICE simulation of the VTD-Cache post layout netlist, and access pattern information from SimpleScalar simulation of SPEC2000 benchmarks. The purpose of the simulation is to obtain an energy improvement metric according to (1) for simulated benchmarks. SimpleScalar was setup as suggested by Table III.

Voltage scaling could be achieved using a wide range of policies that map each voltage to a frequency. In this paper, we purposely selected an aggressive model of voltage scaling in which in order to keep the peak performance the frequency is kept constant while voltage is scaled. This model is referred to as fixed frequency voltage scaling (FFVS). Adopting FFVS results in an exponential increase in the number of failures as voltage is scaled.

Voltage scaling is applied when the processor workload is not high, and performance degradation is not an issue. Although VTD-Cache could also be used this way, but by adopting FFVS policy we intend to show that VTD-Cache could be used when near peak performance is expected. Note that by adopting a different voltage scaling policy (in which frequency is scaled along with voltage it is possible to reduce both $V_{\mathrm{dd}}^{\mathrm{Low}}$ and $V_{\mathrm{dd}}^{\mathrm{High}}$ to lower values.

Fig. 9 illustrates the simulation results of the VTD-Data-Cache for selected SPEC2000 benchmarks. Chosen benchmarks are selected carefully to represent different behavior of data access by SPEC2000 benchmarks. In generation of Fig. 9 the setting of 2 bit local and 6 bit global counter is used. Based on Fig. 8 this setting (2,6) closely estimate the best setting (3,5) with negligible lost in energy savings, however the smaller local counter has a considerably lower area overhead. The transition penalties of 1 and 2 cycles for waking a drowsy line to $V_{\mathrm{dd}}^{\mathrm{Low}}$ and $V_{\mathrm{dd}}^{\mathrm{High}}$ were accordingly considered.

Fig. 9(a) quantifies the percentage improvement in total energy consumption for selected benchmarks. Each bar contains a “Dual Voltage” and a “Triple Voltage” segment. The “Dual Voltage” segment refers to the power saving if only $V_{\mathrm{dd}}^{\mathrm{Drowsy}}$ and $V_{\mathrm{dd}}^{\mathrm{High}}$ were to be used, meaning that every line inside the CWoE was set to high voltage and every cold (Drowsy) line

TABLE IV
AREA OVERHEAD, PERCENTAGE IMPROVEMENT IN TOTAL ENERGY CONSUMPTION, AND PERCENTAGE INCREASE IN EXECUTION TIME FOR BZIP2 BENCHMARK FOR DIFFERENT TIME PENALTIES OF CHARGING A DROWSY LINE READABLE HIGH AND LOW VOLTAGE

Transition Time (Ro Low, To High)	**(1,1)**	**(1,2)**	**(1,3)**	**(1,4)**	**(1,5)**	**(2,2)**	**(2,3)**	**(2,4)**	**(2,5)**	**(3,3)**	**(3,4)**	**(3,5)**
Area overhead Percentage	**4.02**	**3.93**	**3.75**	**3.69**	**3.63**	**3.58**	**3.54**	**3.51**	**3.49**	**3.45**	**3.41**	**3.37**
Percentage improvement in total Energy	**53.75**	**52.81**	**52.04**	**51.35**	**50.15**	**49.62**	**49.26**	**48.91**	**48.04**	**47.66**	**47.41**	**47.18**
Percentage increase in execution time	**0.47**	**0.51**	**0.55**	**0.59**	**0.64**	**0.70**	**0.73**	**0.76**	**0.80**	**0.86**	**0.89**	**0.93**

outside CWoE was sourced with drowsy voltage. The “Triple Voltage” segment illustrates the extra improvement obtained by using triple source voltage, thus enabling the cache ways to also use$V_{\rm dd}^{\rm Low}$. The extra improvement is obtained due to:

1) lower dynamic power consumption when accessing a line sourced with $V_{\rm dd}^{\rm Low}$;
2) lower static power consumption of all the healthy ways within CWoE due to exponential relation of leakage to voltage level;
3) reduction in the amount of energy used to power the drowsy lines (charge up the capacitances) to a readable voltage since most of the lines within the CWoE are readable at $V_{\rm dd}^{\rm Low}$.

Fig. 9(b) provides statistical information for each benchmark on the average number of ways that are sourced from each of the three available voltage supplies. In order to obtain this graph, every 10 000 cycles, a snapshot of the cache state is taken to calculate the number of ways sourced from each of $(V_{\rm dd}^{\rm Drowsy}, V_{\rm dd}^{\rm High}, V_{\rm dd}^{\rm Low})$. Benchmarks with better locality in access pattern to data statistically have a larger portion of their ways in drowsy mode. Note that, by choosing a 2 bit local and 6 bit global counter according to Section III-D the Cache Window of Execution is somewhere in [191 255] cycles. Fig. 9(b) suggests that in every [191 to 256] cycles the recently accessed data is stored in 4% to 6% of cache physical locations. Furthermore, note that Fig. 9(b) is an average over the entire execution time. In real time, the average number of ways that are not in drowsy state will vary depending on benchmark properties at that execution window, typically during phase changes, the CWoE is the largest.

Fig. 9(b) presents a breakdown of the execution time penalty for each benchmark. Percentage increase in the execution time is related to many factors such as:

1) *Penalty for a soft-miss due to transition between* $V_{\rm dd}^{\rm High}$ *and* $V_{\rm dd}^{\rm Low}$: The larger the associated penalties the larger the execution time.
2) *Locality of Access to Data and Instruction:* Higher locality reduces the chances of soft-miss and reduces the number of transitions.
3) *Miss Rate:* Since voltage of tags are not scaled in this architecture (scaling the voltage of tags will be considered in future work) upon a miss on a drowsy line, conditioned that the cache line is not accessed during access to L2 cache (for a non blocking cache), the cold line have the entire duration of L2 access to charge up to the writable voltage level without affecting the execution time.

In addition, since the penalty of soft-miss compared to cache miss is small, having a lot of cache misses reduces the contribution of soft-misses to the execution time percentage wise.

As suggested previously by changing the size of the pMOS transistors in WVSs that charge the memory cells in each cache way to $V_{\rm dd}^{\rm High}$ or $V_{\rm dd}^{\rm Low}$, the number of cycles that a drowsy line takes to charge could be changed. Choosing a larger pMOS makes the transition faster, however since each cache way uses its own pMOS pull up transistor the area overhead quickly increases. On the other hand, choosing a smaller pMOS pull up transistor increases the charge up latency and possibly the number of cycle penalties for a soft-miss affecting the execution time and therefore the circuit leaks for a longer time. This increases the static power consumption of the memory (and entire system) and reduces the total energy savings.

Fig. 9(d) compares the energy savings of a VTD-Cache with Drowsy cache described in [13] for different benchmarks. Drowsy cache is a subset of VTD-Cache in which the $V_{\rm dd}^{\rm Low}$ net is tied to $V_{\rm dd}^{\rm High}$. To obtain the setting of drowsy cache as suggested in [13] the local counter is set to one bit and the global counter to 11 bits (for ∼2000 cycle interval between transition to drowsy) Since the CWoE is not optimized for the maximum saving (changes with the technology model card) and also since now every transition out of drowsy state needs to charge the drowsy line all the way to $V_{\rm dd}^{\rm High}$ the energy saving is significantly reduced. Note that the captured data could change when layout of the cache, or the technology model card, as well as the ratio of leakage energy to dynamic energy. However VTD-Cache always outperforms the conventional Drowsy-cache in terms of energy saving.

As mentioned previously, one of factors that affect the final energy saving is the time penalty of transition of a drowsy line to low or high voltage after a soft-miss has occurred. To better illustrate this, a set of simulations were performed, in which the charge up latencies for transition of drowsy lines to high and low voltage is varied. Table IV quantifies the extent of power saving for “bzip2” benchmark across different combinations of transition latencies. As illustrated the smaller the charge up latencies (by paying larger area overhead) the larger the improvement in total energy saving. Table IV further quantifies the area overhead and increase in the execution time of the cache for each combination. Note that the area overhead includes the area overhead incurred by pMOS pull up transistor at each WVS, the area overhead of WVSs, SVSs with 2 bit counter and modified comparator structures. Increasing the size of SVS counter to 3 bit roughly adds an addition 0.9% to the total area overhead.

TABLE V
BEST CHOICES OF INITIAL VALUE 2 AND 3 bit LOCAL COUNTER, WIDTH OF GLOBAL COUNTER, AND VOLTAGE FOR MAXIMUM IMPROVEMENT IN TOTAL ENERGY CONSUMPTION OF DIFFERENT BENCHMARKS

SVS Counter Size	2 bits				3 bits			
Bench mark	**Global Bits**	**Local Counter Initial Value**	V_{dd}^{Low}	**Percentage improvement in total Energy**	**Global Bits**	**Local Counter Initial Value**	V_{dd}^{Low}	**Percentage improvement` in total Energy**
bzip2	6	3	0.70	52.468	5	6	0.70	52.6%
Gcc	6	3	0.70	53.365	5	7	0.70	53.5%
Vpr	6	3	0.70	56.640	5	6	0.70	56.9%
Gzip	6	3	0.70	56.773	5	5	0.70	56.9%
Mcf	4	2	0.70	58.721	4	4	0.70	58.8%
Twolf	7	3	0.75	62.570	6	5	0.75	62.7%
Gap	4	2	0.70	62.415	4	4	0.70	62.5%
Crafty	5	3	0.70	63.198	4	7	0.70	63.2%

As suggested previously the size of the local and global counter and $V_{\text{dd}}^{\text{Low}}$ at which a benchmark could obtain its highest energy improvement might be different from the setting chosen across all benchmarks (as obtained from Fig. 7). Upon accessing a drowsy line the local counter could also be loaded to a value smaller than the maximum number. Table V summarizes the best global counter settings, initial value of the local counter for choices of 2 and 3 bit local counter, and $V_{\text{dd}}^{\text{Low}}$ such that the total energy consumption improvement is maximized. For example, the setting of 3 bit local counter (initialized to maximum value) and 5 bit global counter that was suggested by Fig. 7 is the ideal setting for benchmark "gcc". However, as suggested by Table V other benchmark could gains higher power savings if the initial value of the counter and size of global counters are changed. In addition throughout the execution as the benchmark goes through different phases, its behavior and cache access pattern (CWoE) changes, suggesting that dynamically reconfiguring the VTD-Cache settings could potentially result in higher energy savings.

VI. MAJOR DIFFERENCES WITH DROWSY CACHE

Our proposed architecture improves the drowsy cache in multiple ways: 1) it provides better chances of power saving by allowing voltage scaling in non-drowsy lines while providing a mechanism for tolerating process variation by enabling the WVSs to supply the cache way from a higher voltage; 2) for most transitions (drowsy to low voltage) it reduces the energy penalty of waking a drowsy line; and 3) allows a new and enhanced drowsy state control mechanism.

Drowsy cache reduces the power consumption by putting the cold lines, which are out of the window of execution, in data retention voltage and supplies the nominal voltage to non drowsy lines or sets. In our approach not only are the cold cache lines sourced with DRV voltage but also the cache has the ability to control the voltage in the CWoE by sourcing each cache way from an appropriate voltage. This mechanism allows the cache to tune its window of execution voltage setting such that it reduces its read and write power while taking into account process variation effects. Since voltage control is distributed over smaller slices of memory (a cache way compare to an entire line or cache) existence of a bit which is severely affected by process variation only causes the smaller memory slice to be sourced from a higher voltage and not the entire set or cache.

When a cold line is accessed in a drowsy cache, a soft-miss is incurred and then the entire cache line is connected to the $V_{\text{dd}}^{\text{High}}$. When changing state each memory cell has to charge or discharge its internal capacitances. By comparison, in VTD-Cache during a wake up process only cache ways that contain weak bits will be charged to the $V_{\text{dd}}^{\text{High}}$ and others only have to charge their internal capacitances to the $V_{\text{dd}}^{\text{Low}}$ thus reducing the energy required for state transition. Considering that $V_{\text{dd}}^{\text{Low}}$ is chosen such that the majority of the cache ways are readable in that voltage, in VTD-Cache the transition to $V_{\text{dd}}^{\text{High}}$ happens with much lower frequency.

VTD-Cache provides a far more accurate control over the CWoE by having a logical dedicated counter for each cache set. Compared to other policies [13], [15], this prevents a burst of soft-misses from happening, thus reducing the execution time of a system using VTD-Cache. In addition VTD cache could be enhanced by allowing it to load different reset values into local counters to adapt to phase changes in the program execution. In fact, the VTD-cache can be considered as a generalization of the Drowsy Cache presented in [13].

VII. CONCLUSION

In this paper, we presented the VTD-Cache a novel solution for obtaining low power cache for high performance processors while addressing the reliability issues raised by process variability. We explored the design space of VTD-Cache architecture and its components. We demonstrated how the VTD-Cache setting (number of local bits, global bits and $V_{\text{dd}}^{\text{Low}}$) is chosen to maximize the improvement in total energy savings. Our simulation results indicate a significant improvement in total energy consumption across simulated benchmarks. We consider VTD-Cache as a logical extension to drowsy cache, further improving its dynamic power consumption. While taking into account "weak cells," the VTD-Cache reduces dynamic power consumption of accessing most of the cache ways within CWoE while reducing the static power consumption of cache ways supplied from low voltage between accesses. In future work, we intend to address the problem of enforcing triple voltage supply policy to tag section of the cache as well as dynamic reconfiguration policies and design issues to further improve energy consumption for adapting with changes in the phase of each benchmark execution.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23–29, Jul.–Aug. 1999.

[2] Y.-F. Tsai, D. Duarte, N. Vijaykrishnan, and M. J. Irwin, "Implications of technology scaling on leakage reduction techniques," in *Proc. Des. Autom. Conf.*, Jun. 2003, pp. 187–190.

[3] M. Anis, "Subthreshold leakage current: Challenges and solutions," in *Proc. 15th Int. Conf. Microelectron. (ICM)*, Dec. 2003, pp. 77–80.

[4] A. Agarwal, B. C. Paul, S. Mukhopadhyay, and K. Roy, "Process variation in embedded memories: Failure analysis and variation aware architecture," *IEEE Trans. Solid-State Circuits*, vol. 40, no. 9, pp. 1804–1814, Sep. 2005.

[5] S. R. Nassif, "Modeling and analysis of manufacturing variation," in *Proc. CICC*, 2001, pp. 223–228.

[6] Bhavnagarwala, X. Tang, and J. D. Meindl, "The impact of intrinsic device fluctuation on CMOS SRAM cell stability," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 658–665, Apr. 2001.

[7] A. Sasan, H. Homayoun, A. Eltawil, and F. J. Kurdahi, "Process variation aware SRAM/cache for aggressive voltage-frequency scaling," in *Proc. DATE*, pp. 911–916.

[8] A. Sasan (Mohammad A Makhzan), A. Khajeh, A. Eltawil, and F. Kurdahi, "Limits of voltage scaling for caches utilizing fault tolerant techniques," in *Proc. ICCD*, pp. 488–495.

[9] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S. L. Lu, "Trading off cache capacity for reliability to enable low voltage operation," in *Proc. ISCA*, 2008, pp. 203–214.

[10] H. Mahmoodi, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nano-scaled cmos," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1859–1880, Dec. 2003.

[11] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Statistical design and optimization of SRAM cell for yield enhancement," in *Proc. Int. Conf. Comput.-Aided Des. (ICACD)*, 2004, pp. 10–13.

[12] J. P. Kulkarni, K. Kim, and K. Roy, "A 160 mV robust schmitt trigger based subthreshold SRAM," *IEEE J. Solid-State Circuits*, vol. 42, no. 10, pp. 2303–2313, Oct. 2007.

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

[14] SimpleScalar LLC, Ann Arbor, MI, "SimpleScalarTM simulator," 2003. [Online]. Available: http://www.simplescalar.com/

[15] J. Zushi, G. Zeng, H. Tomiyama, H. Takada, and K. Inoue, "Improved policies for drowsy caches in embedded processors," in *Proc. 4th IEEE Int. Symp. Electron. Des., Test, Appl. (DELTA)*, Jan. 2008, pp. 362–367.

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

[17] H. Zhou, M. C. Toburen, E. Rotenberg, and T. M. Conte, "Adaptive mode-control: A static-power-efficient cache design," in *Proc. Int. Conf. Parallel Arch. Compilation Techn.*, 2001, pp. 61–70.

[18] A. Sasan (Mohammad A Makhzan), A. Khajeh, A. Eltawil, and F. Kurdahi, "Limits of voltage scaling for caches utilizing fault tolerant techniques," in *Proc. ICCD*, 2007, pp. 488–495.

[19] K. Ishimaru, "45 nm/32 nm CMOS 'challenge and perspective'," in *Proc. 37th Euro. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2007, pp. 32–35.

[20] A. Agarwal, K. Roy, and T. N. Vijaykumar, "Exploring high bandwidth pipelined cache architecture for scaled technology," in *Proc. Des., Autom. Test Eur. Conf. Exhibition*, 2003, pp. 778–783.

[21] M. A. Makhzan (A. Sasan), A. Khajeh, A. Eltawil, and F. J. Kurdahi, "A low power JPEG2000 encoder with iterative and fault tolerant error concealment," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 827–837, Jun. 2009.

[22] K. Flautner, K. Nam Sung, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *Proc. 29th Annu. Int. Symp. Comput. Arch.* , 2002, pp. 148–157.

[23] H. Homayoun, A. Sasan (M. A. Makhzan), and A. V. Veidenbaum, "Multiple sleep mode leakage control for cache peripheral circuits in embedded processors," in *Proc. CASES*, pp. 197–206.

[24] M. A. Lucente, C. H. Harris, and R. M. Muir, "Memory system reliability improvement through associative cache redundancy," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1990, pp. 19.6/1–19.6/4.

[25] Z. Bo, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[26] X. Tang, V. K. De, and J. D. Meindl, "Intrinsic MOSFET parameter fluctuations due to random dopant placement," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 4, pp. 369–376, Dec. 1997.

[27] K. Ishimaru, "45 nm/32 nm CMOS 'challenge and perspective'," in *Proc. 37th Eur. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2007, pp. 32–35.

[28] R. K. Krishnarnurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70 nm microprocessor circuits," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2002, pp. 125–128.

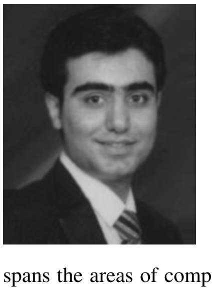
[29] K. Kuhn, "32nm SOC process, getting many things right at once," Mar. 26, 2010. [Online]. Available: http://blogs.intel.com/technology/2010/03/32nm_soc_process_an_analogy_wi.php

Avesta Sasan (Mohammad A. Makhzan) (M'05) received the B.S. degree (*summa cum laude*) in computer engineering and the M.S. and Ph.D. degrees in electrical engineering from University of California Irvine, Irvine, in 2005, 2006, and 2010, respectively.

He is currently with Broadcom Corporation. His research interests include low power design, process variation aware architectures, fault tolerant computing systems, nano-electronic power and device modeling, VLSI signal processing, processor power and reliability optimization and logic-architecture-device co-design. His latest publication and research updates can be found on http://www.avestasasan.com.

Kiarash Amiri (S'10) received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2003, and the M.Sc. degree in electrical engineering from University of Southern California, Los Angeles, in 2006. He is currently pursuing the Ph.D. degree in electrical engineering from the Department of Electrical Engineering and Computer Science, University of California, Irvine.

His research interests include multimedia data compression, low power video and image coding, process variation aware system design, and fault tolerant computing.



Houman Homayoun (M'04) received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2003, the M.S. degree in computer engineering from University of Victoria, Canada, in 2005, and the Ph.D. degree in computer science from University of California, Irvine, Irvine, in 2010.

Dr. Homayoun was the recipient of the four-years UCI/ICS chair fellowship. His research interests include power-temperature and reliability-aware memory and processor design optimizations and spans the areas of computer architecture and VLSI circuit design. The results of his research were published in top-rated conferences including ISLPED, DAC, DATE, HiPEAC, CASES, ICCD, CF, and LCTES.

Ahmed M. Eltawil (M'97) received the Doctorate degree from the University of California, Los Angeles, in 2003 and the M.Sc. and B.Sc. degrees (with honors) from Cairo University, Giza, Egypt, in 1999 and 1997, respectively.

He is an Associate Professor with the University of California, Irvine, where he has been with the Department of Electrical Engineering and Computer Science, University of California, Irvine, since 2005. He is the founder and director of the Wireless Systems and Circuits Laboratory (http://newport.eecs.uci.edu/~aeltawil/), a member laboratory of the Center for Pervasive Communications and Computing (CPCC). He also holds a visiting professorship with King Saud University, Saudi Arabia. His current research interests include low power digital circuit and signal processing architectures for wireless communication systems with a focus on physical layer design where he has published over 60 technical papers on the subject, including four book chapters.

Dr. Eltawil has been on the technical program committees for numerous workshops, symposia, and conferences in the area of VLSI and communication system design. He has received several distinguished awards, including the NSF CAREER Award in 2010 supporting his research in low power systems, as well as the Best Paper Award in 2006 at ISQED. Since 2006, he has been a member of the Association of Public Safety Communications Officials (APCO) and has been actively involved in efforts towards integrating cognitive and software defined radio technology in critical first responder communication networks.

Fadi J. Kurdahi (M'87-SM'03-F'05) received the Ph.D. degree from the University of Southern California, Los Angeles, in 1987.

Since then, he has been a faculty member with the Department of Electrical and Computer Engineering, University of California, Irvine (UCI), where he conducts research in the areas of computer-aided design of VLSI circuits, high-level synthesis, and design methodology of large scale systems, and serves as the Associate Director for the Center for Embedded Computer Systems (CECS).

Dr. Kurdahi was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, Area Editor in *IEEE Design and Test* for reconfigurable computing, and served as program chair, general chair, or on program committees of several workshops, symposia, and conferences in the area of CAD, VLSI, and system design. He was a recipient of the Best Paper Award for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS in 2002, the Best Paper Award in 2006 at ISQED, and four other Distinguished Paper Awards at DAC, EuroDAC, ASP- DAC, and ISQED. He also received the Distinguished Alumnus Award from this Alma Mater, the American University of Beirut in 2008. He is a fellow of the AAAS.