eCACTI: An Enhanced Power Estimation Model for On-chip Caches

Mahesh Mamidipaka
maheshmn@cecs.uci.edu

Nikil Dutt
dutt@cecs.uci.edu

Center for Embedded Computer Systems
Donald Dren School of Information and Computer Science
University of California, Irvine, CA 92697, USA

CECS Technical Report #04-28
Center for Embedded Computer Systems
University of California, Irvine, CA 92697, USA

Sept 14, 2004

Abstract

There is a growing need for accurate power models at the higher levels of design hierarchy. CACTI is a micro-architecture level tool widely used (i) to estimate power dissipation in caches and (ii) to determine the cache configuration that best meets the desired optimization criterion. However, we observed several limitations in CACTI that lead to inaccuracies in cache power estimates especially as we move to deep sub-micron (DSM) technologies: a) lack of models to account for leakage power, b) use of constant gate widths for most devices irrespective of its capacitive load, and c) lack of models to account for power dissipation in sub-blocks that are outside the time critical path. As a result, the cache configuration determined by CACTI may not be optimal because of these limitations. In this paper, we describe *eCACTI* (enhanced CACTI), a tool that addresses these limitations in CACTI thereby improving the accuracy of its power estimates. We validated *eCACTI* power estimates against SPICE based simulations on industrial designs. Furthermore, we show that for DSM technologies, CACTI does not generate power optimal cache configuration, which highlights the need for the enhancements we developed in *eCACTI*. Finally, we demonstrate the use of *eCACTI* to study the effects of (i) technology on cache leakage and total cache power, (ii) dual-V_{th} optimization on sub-block and total cache leakage power, (iii) effects of varying cache size, block size, and associativity for DSM technologies.

Contents

1 Introduction **4**

2. Related Work **6**

3. CACTI: An Overview **6**
3.1. Power Estimation Methodology in CACTI 7

4. Enhancements in *eCACTI* **10**
4.1 Leakage Power 10
4.2. Device Width Calculation 12
4.3. Non-time Critical Sub-blocks 12
4.4. Additional Enhancements 14
4.4.1 Considering Write Operation Power 14
4.4.2 Look-Up Table based Leakage Current Model 14
4.4.3 Evaluating the Effect of Low Power Techniques 15
4.4.4 Power Estimation of a Specific Sub-bank Configuration 15

5. Model evaluation **15**
5.1. Leakage Power Model Evaluation 16
5.2. Evaluation of Leakage and Dynamic Power Estimation in *eCACTI* 16

6. Comparison between CACTI and *eCACTI* **17**

7. Experiments **18**
7.1. Effect of Technology on Cache Power 18
7.2. Effect of Cache Parameters on Total Power Dissipation 19
7.3. Sub-block Power Dissipation Contributions to Total Cache Power 21
7.4. Effect of dual-V_{th} Optimization on Cache Leakage Power 21

8. Summary **23**

List of Figures

1 Impact of Design Decisions on Power Dissipation at Different Levels of Design Hierarchy [10] 5
2 Percentage Contribution of Sub-threshold Leakage with Decreasing Feature Size [4] 5
3 Typical Cache Structure 8
4 Typical Structure of a 6-T Memory Cell 11

5 Typical Implementation of Read Column Logic 13
6 Plot Showing the Accuracy of Leakage Current Model for a NMOS Device 16
7 Comparison of CACTI and *eCACTI* Power Estimates (Tech=0.07u) 18
8 Plot Showing Variation of Power Dissipation with Technology 19
9 Plot Showing Variation of Power Dissipation with Cache Size 20
10 Plot Showing Variation of Power Dissipation with Block Size 20
11 Plot Showing Variation of Power Dissipation with Associativity 21
12 Sub-block Power Contributions to Total Cache Dynamic Power 22
13 Sub-block Power Contributions to Total Cache Leakage Power 22
14 Leakage and Total Cache Power for Varying Cache Sizes 23
15 Sub-block Power Contributions to Total Cache Leakage Power (Dual Vt Optimization) 24

List of Tables

1 CACTI Input Parameters 9
2 Cache Organizational Parameters 9
3 Evaluation of *eCACTI* **Dynamic** Power Estimates 17
4 Evaluation of *eCACTI* **Leakage** Power Estimates 17

1 Introduction

Power dissipation has become a major design constraint in both portable devices and many system designs. System designers are often faced with the challenge of meeting the conflicting requirements of performance and power. Figure 1 shows the impact of design decisions on system power at various levels of design hierarchy [10]. It is seen that design decisions taken higher in design cycle have greater influence on the system power dissipation. To meet the stringent power and performance constraints in contemporary designs, system designers need tools to perform design space exploration at higher levels of design hierarchy.

Caches consume a significant portion of the total power dissipation in contemporary processors, as much as 40% [5, 8]. It is important to evaluate various possible cache configurations for power-delay-area trade-offs to meet the system requirements. CACTI [11, 18], an integrated cache timing and power model was proposed by Wilton and Jouppi to explore several possible cache configurations for power, delay, and area early in the design hierarchy. The tool is widely used by the research community (a) at the micro-architecture level to estimate the power dissipation in caches and (b) at the system level in design space exploration tools such as Wattch [5], SimplePower [15], and HotLeakage [19]. However, we observed that CACTI has the following limitations which become more prominent in deep sub-micron (DSM) technologies:

- Leakage power is increasing exponentially with process technology and projected to dominate the total power dissipation. Although there are other components contributing to static power dissipation, sub-threshold leakage power increases exponentially and contributes to a majority of the static power dissipation. Figure 2 shows the percentage contribution of sub-threshold leakage power to the total power dissipation with decreasing feature size. In 0.05μ technology, the sub-threshold leakage power contribution is projected to increase to almost half of the total power dissipation. But CACTI currently does not have models to account for leakage power in caches.

- Except for wordline drivers, the transistor widths of various devices are assumed to be constant in the analysis for power and delay. This assumption is incorrect because the transistor widths in actual cache designs change according to their capacitive load.

- A number of sub-blocks (such as read control logic, write control logic, and sense-amplifier logic) that do not fall in the time critical path, are not modeled in CACTI. However, for accurate power estimation, all the sub-blocks are critical and need to be considered.

We show in Section 6 that even with current feature sizes, these limitations lead to significant inaccuracies in the CACTI power estimates. This in turn leads to an error in determining the optimal cache configuration. Furthermore, these anamolies in CACTI will worsen with decreasing feature sizes in the future. In our proposed tool, *eCACTI* (enhanced CACTI), we improve the power estimation model in CACTI by explicitly addressing all the above mentioned limitations. This will enable system designers to do more accurate power-aware design space exploration at the micro-architecture

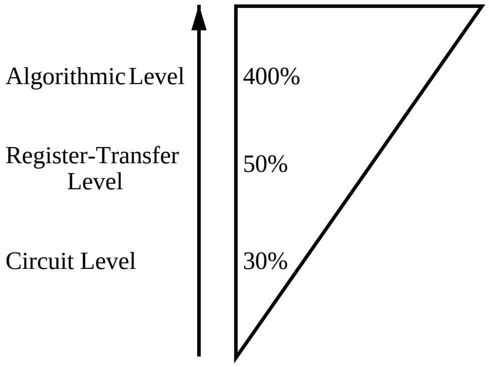


Figure 1. Impact of Design Decisions on Power Dissipation at Different Levels of Design Hierarchy [10]

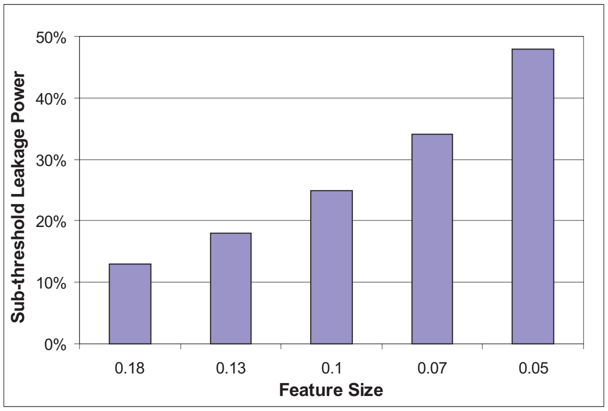


Figure 2. Percentage Contribution of Sub-threshold Leakage with Decreasing Feature Size [4]

level and at the system level as we move to more aggressive DSM technologies. In this paper, we use *eCACTI* to study the effect of process technology and various cache parameters on the cache power. We analyze the static power dissipation contributions of various sub-blocks to the total power dissipation. Finally we study the effect of dual threshold voltage (dual-V_{th}) technology on cache power dissipation.

The paper is organized as follows. Section 2 presents related work with a brief overview of CACTI in Section 3. Section 4 presents the enhancements in *eCACTI* and the methodology used to incorporate them. Section 5 presents the methodology used to validate *eCACTI* power estimates. Section 6 compares the cache configuration results generated by CACTI with *eCACTI* and Section 7 presents experiments showing the applications of the tool. Finally, Section 8 summarizes this paper.

2. Related Work

Caches have long been recognized as a critical component in system designs. CACTI is a popular tool used by computer architects for estimation of power dissipation in caches and for determining the optimal cache configuration. It is also used in various system-level tools such as Wattch [5], SimplePower [15], and HotLeakage [19] for design space exploration. Before CACTI, analytical models were proposed for estimation of cache area by Mulder et al. [9] and models for cache access times were proposed by Wada et al. [16]. The access time models proposed by Wada had certain limitations which were accounted in CACTI 1.0 [18]. The next version, CACTI 2.0 [11], primarily included power models so as to evaluate power and access time trade-offs in different cache configurations. CACTI 3.0 [12] was later released along with area models to explore the cache configurations for area, power, and access times. In this paper, we refer to CACTI 3.0 as CACTI. A brief overview of CACTI is presented in Section 3. CACTI was developed and validated on cache designs using a 0.8μ technology. However, in today's DSM technologies, we observe that the power estimation models used in CACTI have significant limitations leading to inaccurate power estimates. In this paper, we describe *eCACTI* that addresses these limitations, thereby improving the accuracy of the cache power estimates.

3. CACTI: An Overview

CACTI takes high level cache parameters shown in Table 1 as input and outputs the cache configuration that best meets the desired optimization function. The basic structure of the cache considered in CACTI is shown in Figure 3. A cache configuration is described in terms of six organizational parameters: *Ndbl, Ndwl, Nspd, Ntwl, Ntbl, and Ntspd*. The description of these parameters are illustrated in Table 2. The parameters *Ndbl, Ndwl, and Nspd* correspond to the configuration of the data array, whereas *Ntwl, Ntbl, and Ntspd* refer to the configuration of the tag array. While the wordline and bitline segments (*Ndwl, Ndbl* for data array and *Ntwl, Ntbl* for tag array) define the number of sub-banks in the cache, the number of sets mapped to a wordline (*Ndsp, Ntspd*) controls the aspect ratio of each sub-bank. For a given set of cache parameters (cache size, block size, and associativity) the cache access time, power dissipation, and area are dependent on the values assigned to these configuration parameters. To determine optimal cache configuration for a desired objective function, CACTI exhaustively calculates the area, power, and access time for every possible configuration of cache and selects that configuration that best meets the optimization function. To calculate area, power, and access times, CACTI uses various analytical models. The pseudo code in CACTI that performs exhaustive exploration of all possible cache configurations is shown below.

```
// Initialize PowerDelayProd with a large number
PowerDelayProd = VeryLargeNumber;
// for all possible cache configurations
for (Nspd=1; Nspd<=MaxSpd; Nspd=Nspd*2)
 for (Ndwl=1; Ndwl<=MaxNdwl; Ndwl=Ndwl*2)
  for (Ndbl=1; Ndbl<=MaxNbl; Ndbl=Ndbl*2)
   for (Ntspd=1; Ntspd<=MAXSPD; Ntspd=Ntspd*2)
    for (Ntwl=1; Ntwl<=1; Ntwl=Ntwl*2)
     for (Ntbl=1;Ntbl<=MAXN;Ntbl=Ntbl*2) {

      params={C,B,A,Ndbl,Ndwl,Nspd,Ntwl,Ntbl,Ntspd};
      // find out if the configuration parameters
      // form a valid combination
      if (params_valid(params)) {
        // Estimate area
        area = cache_area(params);
        // Estimate power
        power = cache_power(params);
        // Estimate access time
        access_time = cache_access_time(params);

        // find configuration with minimal
        // power-delay product
        if (power*access_time < PowerDelayProd) {
          PowerDelayProd = power * access_time;
          // save this configuration
          save_cfg(Ndbl,Ndwl,Nspd,Ntwl,Ntbl,Ntspd);
        }
      }
    }
```

The pseudo code tries to find the cache configuration with minimal power-delay product and hence the area estimates are not used in the optimization function. While more details on configuration parameters and analytical models for area, access time can be found in [18, 16], the relevant power estimation modeling methodology in CACTI is briefly described below.

3.1. Power Estimation Methodology in CACTI

The basic model used for estimation of power dissipation is shown in Equation (1).

$$P_{diss} = C_L \cdot V_{dd}^2 \cdot P_{0\to 1} \cdot f \tag{1}$$

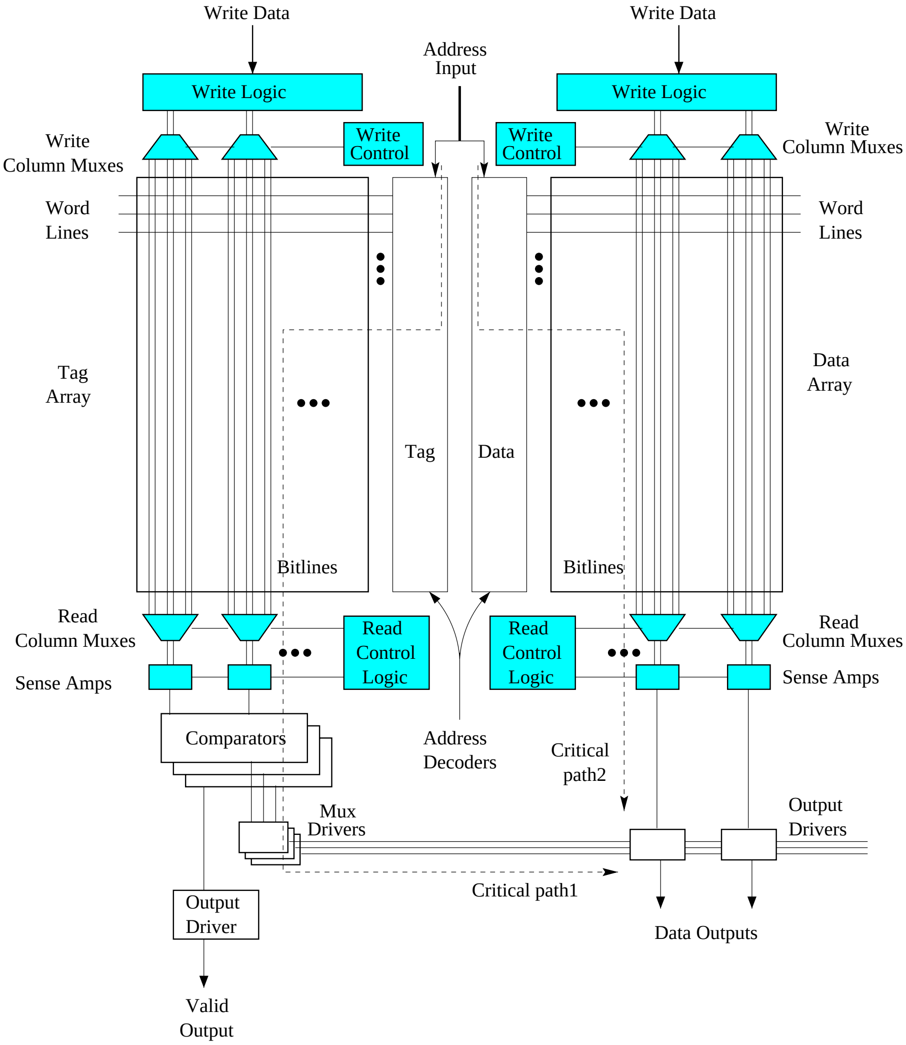


Figure 3. Typical Cache Structure

Table 1. CACTI Input Parameters

Parameter	Description
C	Cache size in bytes
B	Block size in bytes
A	Associativity
b_0	Data output width in bits
b_{addr}	Address bus width in bits

Table 2. Cache Organizational Parameters

Parameter	Description
Ndwl	# of wordline segments (data array)
Ndbl	# of bitline segments (data array)
Nspd	# of sets mapped to single wordline (data array)
Ntwl	# of wordline segments (tag array)
Ntbl	# of bitline segments (tag array)
Ntspd	# of sets mapped to single wordline (tag array)

where, C_L is the physical capacitance of a device, V_{dd} is the device supply voltage, $P_{0\to 1}$ is the probability of a transition at the capacitive load from '0' to '1' and f is the frequency of the cache operation. Since the supply voltage and frequency of operation is typically constant across the whole cache design, the crucial part of estimating power dissipation is to estimate the switching capacitance.

Typically an implementation of a cache has multiple sub-banks in both tag and data arrays. While the sub-banks in data arrays could have different sizes, aspect ratio, and organization compared to sub-banks in tag array, sub-banks within data and tag arrays are usually identical. The configuration parameters *Ndbl, Ndwl, Nspd, Ntwl, Ntbl, and Ntspd* define the size and organization of the sub-banks in the data and tag array of the cache. Each sub-bank is composed of six main sub-blocks: a local decoder to decode the input address, memory core containing the bit cells arranged in rows and columns, read logic containing read column multiplexers, differential sense amplifier and output data drivers, write logic containing write column multiplexers and logic to drive data on to the bitlines, read and write control logic to drive signals to control the write and read logic respectively. Modeling of power in CACTI is based on the assumption that these sub-blocks are implemented using typical circuit implementation styles. For example, it is assumed that the memory bit cell would be implemented using 6-transistor bit cell design. More details on the implementation styles considered for other sub-blocks can be obtained from [18]. Using the templates of these typical sub-block circuit implementation styles and organization specified by the configuration parameters, the capacitances on the switching nodes are calculated and are used to estimate power dissipation. For example, the derivation of the model for data wordline power is described in Equations 2, 3, and 4.

$$N_{cells} = \frac{8 \cdot B \cdot A \cdot N_{spd}}{Ndwl} \tag{2}$$

$$C_{wl} = N_{cells} \cdot (C_{bitCell} + C_{wire}) \tag{3}$$

$$P_{wl} = C_{wl} \cdot V_{dd}^2 \cdot f \tag{4}$$

where N_{cells} represents the number of memory bit cells per row in each data array sub-bank, C_{wl} represents the capacitance per wordline in each data array sub-bank, $C_{bitCell}$ is the capacitive load offered by a bit cell, C_{wire} is the wire capacitance per unit bit cell width, and P_{wl} is the wordline power dissipation in a data array sub-bank. Note that $C_{bitCell}$ and C_{wire} are user provided inputs.

4. Enhancements in *eCACTI*

In this section, we list the limitations in CACTI and illustrate how these limitations are addressed in our proposed tool, *eCACTI*[1].

4.1 Leakage Power

Leakage power has been increasing exponentially with technology and is expected to dominate the total power dissipation in future technologies. A major limitation in CACTI is the lack of models for leakage power estimation, which results in inaccurate power numbers for DSM technologies.

To account for leakage power in eCACTI, we use the transistor level model proposed by Zhang et al. [19] for estimating leakage current in a MOSFET. The analytical equation is shown in Equation (5). This model is shown to be accurate and also allows us to evaluate the effect of variations in temperature (T) and supply voltage (V_{dd}) which have exponential dependence on the leakage currents. For a given threshold voltage (V_{th}) and temperature (T), except for the device width (W) all the remaining terms are constant for all the transistors in a given design. So Equation (5) can be reduced to Equation (6), where I_l is the leakage of a unit width transistor at a given temperature and threshold voltage.

$$I_{lkg} = \mu_0 . C_{ox} . \frac{W}{L} \cdot e^{b(V_{dd} - V_{dd0})} \cdot v_t^2 \cdot (1 - e^{-\frac{V_{dd}}{v_t}}) \cdot e^{\frac{-V_{th} - V_{off}}{nv_t}} \tag{5}$$

$$I_{lkg} = W \cdot I_l(T, V_{th}) \tag{6}$$

In our earlier work, we proposed analytical models for leakage power estimation in SRAMs [1] in terms of MOSFET leakage currents and high level design parameters. Caches are SRAM based sub-banks organized in a regular manner to achieve the required functionality. We enhanced the SRAM leakage power models in [1] and integrated them into *eCACTI* for estimation of cache leakage power.

[1] In this paper, we focus on the limitations pertaining to the power estimation model and assume that the access time and area models are correct.

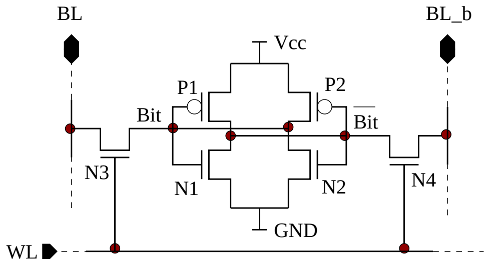


Figure 4. Typical Structure of a 6-T Memory Cell

A SRAM is typically composed of six sub-blocks: memory core, read column logic, read control logic, write column logic, write control logic and the address decoder. We developed analytical models for each SRAM sub-block in each of its operational states. For example, a memory cell sub-circuit in the memory core can be in a read, write, idle, or precharge operational state depending on the operation on the SRAM and phase of the clock. While details on the definition of an operational state and derivation of analytical models for various SRAM sub-blocks under each operational state are described in [1], the remainder of this section will illustrate this process by deriving the analytical model for the memory core in the read operational state.

Figure 4 shows a typical 6-transistor memory cell design. To maintain symmetry, in most memory cell designs, transistors (P1, P2) typically share the same characteristics and physical geometry and hence have same leakage in the off-state. Similarly transistors (N1, N2) and (N3, N4) also have the same characteristics. So $I_{lkg}(N1) = I_{lkg}(N2); I_{lkg}(N3) = I_{lkg}(N4); I_{lkg}(P1) = I_{lkg}(P2)$.

Since leakage power is contributed only by the transistors in off-state, we first identify the transistors in off-state. During the read phase, one of the wordlines is activated by the address and the remaining wordlines remain deactivated. Then, corresponding to the data in each memory cell of the selected row, one of the bitlines in all the bitline pairs (BL, BL_b), discharges partially (typically 15% of V_{dd}). For simplicity of the analysis, we assume that the amount of discharge in a bitline is negligible and treat both BL and BL_b to be at V_{dd}. For example, for memory cells with $WL = 1$ and $Bit = 0$, transistors $N1$ and $P2$ would be in the off-state contributing to leakage current. Considering the symmetry of the transistors irrespective of the data in the memory cells, the leakage current in the memory cell in the two scenarios, $WL = 1$ and $WL = 0$ are shown in Equation (7).

$$I_{memCellRd} = \begin{cases} I_{N1} + I_{N4} + I_{P2} & \textbf{for } \text{WL=0} \\ I_{N1} + I_{P2} & \textbf{for } \text{WL=1} \end{cases} \tag{7}$$

$$I_{memCellRd} = \begin{cases} (W_{N1} + W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP} & \textbf{for } \text{WL=0} \\ W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP} & \textbf{for } \text{WL=1} \end{cases} \tag{8}$$

$$\begin{aligned} I_{memCoreRd} = & N_{rows} \cdot N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) \\ & + (N_{rows} - 1) \cdot N_{cols} \cdot W_{N4} \cdot I_{lN} \end{aligned} \tag{9}$$

where, I_{N1}, I_{N4}, and I_{P2} are the leakages of transistors $N1$, $N4$, and $P2$ respectively. Using Equation (6), Equation (7) can be reduced to Equation (8). W_{N1}, W_{N4}, and W_{P2} are the widths of transistors $N1$, $N4$, and $P2$ respectively. I_{lN} and I_{lP} are the leakage currents of unit width NMOS and PMOS transistors respectively for a given technology and process parameters. Since there are N_{cols} cells for which $WL = 1$ and $(N_{rows} - 1) \cdot N_{cols}$ cells for which $WL = 0$, the memory core leakage in the read phase can be derived as shown in Equation (9).

4.2. Device Width Calculation

Device width is a primary factor that influences both leakage and dynamic power dissipation. Device width affects the gate capacitance linearly, thereby influencing dynamic power dissipation. It also affects sub-threshold leakage linearly as can noted from Equation (5). The width of a device in a cache design is determined based on the capacitive load driven by the device. For example, the width of the decoder output driver that drives the wordline driver, is determined according to the capacitive load offered by the wordline driver. However, in CACTI, except for the device width of the wordline driver, *all the other device widths assume a constant value for all the cache configurations,* leading to inaccuracies.

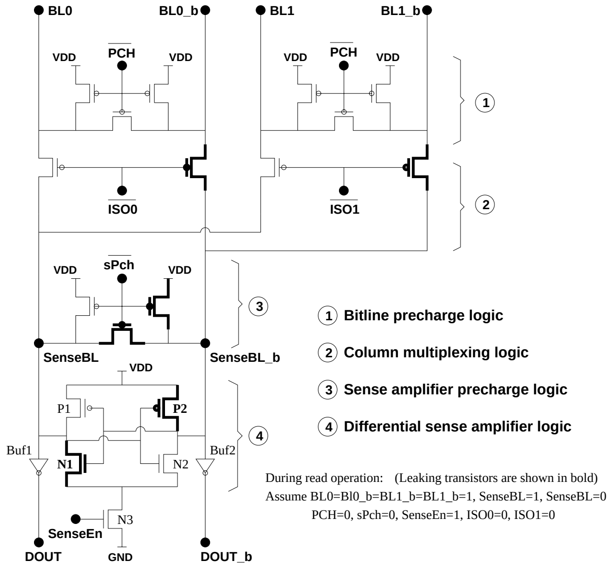
In *eCACTI*, we calculate the device widths in accordance to its capacitive load. While various strategies can be used to determine transistor size based on its capacitive load, we use the principles of logical effort discussed in [13]. The assumption being that the cache design would be optimized for minimal delay. Assuming that the template of each sub-block in caches is known, the following procedure is used in *eCACTI* to determine the transistor sizes.

- Compute the path effort: $F = GBH$
- Estimate the best number of stages: $\hat{N} \approx log_4 F$
- Compute the stage effort: $\hat{f} = F^{1/N}$
- Starting from the end, work backward to find the transistor sizes: $C_{in_i} = \frac{C_{Out_i} \cdot g_i}{\hat{f}}$

where, f is the path effort, G is the path logical effort, H is the path electrical effort, B is the path branching effort, N is the best number of stages, g_i is the logic effort of stage i, C_{in_i} and C_{out_i} are the input and output capacitances for stage i. More details on this optimization procedure can be obtained from [13].

4.3. Non-time Critical Sub-blocks

A typical implementation of a cache sub-bank consists of six main sub-blocks: an address decoder, memory core, read column logic, read control logic, write column logic and write control logic. However, CACTI models only the sub-blocks that fall in the potential time critical paths. The potential time critical paths in caches, *critical path1* and *critical path2*, are shown in Figure 3. Accordingly,


BL0
BL0_b
BL1
BL1_b
VDD
PCH
VDD
VDD
PCH
VDD
1
ISO0
ISO1
2
VDD
sPch
VDD
3
SenseBL
SenseBL_b
VDD
P1
P2
4
Buf1
N1
N2
Buf2
N3
SenseEn
DOUT
GND
DOUT_b
1 Bitline precharge logic
2 Column multiplexing logic
3 Sense amplifier precharge logic
4 Differential sense amplifier logic

During read operation: (Leaking transistors are shown in bold)
Assume BL0=Bl0_b=BL1_b=BL1_b=1, SenseBL=1, SenseBL=0
PCH=0, sPch=0, SenseEn=1, ISO0=0, ISO1=0

Figure 5. Typical Implementation of Read Column Logic

CACTI models mainly the address decoder, memory core, and read column logic blocks in both data and tag arrays in addition to comparators and some multiplexer logic. Furthermore, CACTI assumes a constant power dissipation value for sense-amplifier based read column logic. However, for accurate estimation of power dissipation, *all the logic blocks* need to be modeled. In *eCACTI*, we model the power dissipation in read control logic, write column logic, and write control logic in both the data and tag arrays. These sub-blocks block are shaded in in Figure 3.

As an example, we detail the derivation of the dynamic and leakage power models for read column logic; the leakage power models for the remaining sub-blocks can be found in [2]. We assume that the circuit level implementation of the read column logic is based on differential sense-amplifier and uses self-timed [3] logic for low power. A typical structure of a differential sense-amplifier based read logic with a 2:1 column multiplexer is shown in Figure 5. It is primarily composed of bitline precharge logic, column multiplexing and isolation logic, sense precharge logic, and the differential sense-amplifiers. During a read operation, while a bitline in each of the bitline pairs discharges partially, one of the sense bitline among each sense bitline pair discharges completely. Also, since the sense bitlines are initially precharged, a transition would occur at the output of the data out buffers as well. So the dynamic power dissipation in read column logic for a read operation can be written as shown in Equation (10).

$$P_{dyn} =(C_{bl} \cdot \delta + C_{SenseBl} + C_{Dout}) \cdot V_{dd}^2 \cdot f \tag{10}$$

where, C_{bl} is the capacitive load on the bitlines, δ is the percentage bitline discharge during a read operation, $C_{senseBl}$ is the capacitive load on the sense bitlines, C_{Dout} is the capacitive load on the data out buffers, V_{dd} is the supply voltage and f being the frequency of operation. Figure 5 shows the leaking transistors in bold during a read operation assuming that $SenseBL = 1$ and $SenseBL\_b = 0$. Although one of the bitlines in each column discharges partially, for the sake of simplicity we assume that this percentage discharge is negligible and consider the voltage at bitlines to be V_{dd}. The leakage power in read column logic during read operation can then be derived as shown in Equation (12).

$$\begin{aligned} P_{lkg} =& 2 \cdot I_{iso} + 2 \cdot I_{sPch} + I_{P2} + I_{N1} + I_{Buf1\_P} + I_{Buf2\_N} && (11) \\ =& (2 \cdot W_{iso} + 2 \cdot W_{sPch} + W_{P2} + W_{Buf\_P}) \cdot I_{lP} + (W_{N1} + W_{Buf2\_N}) \cdot I_{lN} && (12) \end{aligned}$$

where, W_{iso}, W_{sPch}, W_{P2}, W_{N1}, W_{Buf1_P}, and W_{Buf2_N} are the widths of isolation, sense precharge, sense-amplifier and data out buffer transistors yielding leakage currents I_{iso}, I_{sPch}, I_{P2}, I_{N1}, I_{Buf1_P}, and I_{Buf2_N} respectively.

4.4. Additional Enhancements

Additionally, *eCACTI* has certain other enhancements which improve its accuracy and applicability in a variety of cache designs, as discussed below.

4.4.1 Considering Write Operation Power

In CACTI the power dissipation is estimated only for a read operation. However, we observed that in some configurations of caches, power dissipation for a write operation is more than that of a read operation. To address this issue, *eCACTI* estimates power dissipation for both read and write operations and the maximum of the two estimates is considered in determining the optimal cache configuration.

4.4.2 Look-Up Table based Leakage Current Model

We observed based on SPICE simulations that the transistor level model for leakage current (shown in Equation (5)), used in *eCACTI* can have an error margin of as much as 9%. The results of these experiments are shown in Section 5. While this model can be further enhanced to decrease the error margin, we propose a technique to eliminate the error due to leakage current model. The idea is to follow a look-up table (LUT) based approach. Since the SPICE model for MOSFETs are typically available very early in the design cycle, leakage current can be found for various transistor sizes and process corners (best, typical, and worst case) using SPICE simulations. These values can then be stored in a LUT and used in *eCACTI* for leakage current estimation. Since the leakage current for devices would then be based on SPICE simulations, the error due to leakage current model is eliminated. In *eCACTI*, we have an option for the user to either provide a LUT of leakage currents for various device widths or to use the default analytical leakage current model.

4.4.3 Evaluating the Effect of Low Power Techniques

In contemporary cache designs, designers use various optimizations to reduce power dissipation and/or access times. For example, to reduce leakage power dissipation, drowsy caches [6], gated-Vdd [7] techniques are used. *eCACTI* has the flexibility to explore the cache configurations in the presence of these optimization techniques. These optimization techniques can be specified in a parameter file which acts as an input to *eCACTI*. An example command line usage for exploring a cache design at 0.13μ technology of size 4K bytes, block size of 16 bytes, associativity of 2, and with an input parameter file, paramFile, is shown below.

```
eCACTI 4096 16 2 0.13 paramFile
```

4.4.4 Power Estimation of a Specific Sub-bank Configuration

Cache designers typically focus on the design of a single tag array sub-bank and single a data array sub-bank. These sub-banks are then replicated in data and tag arrays for the design of the whole cache. While CACTI suggests the configuration that best meets the desired optimization function, there is no flexibility to find the power dissipation in an already existing cache with a specific tag and data sub-bank configurations. For example, if a designer wishes to find the power dissipation in a cache with data array sub-banks of size 128 rows and 128 columns and tag array sub-banks of size 64 rows and 128 columns, CACTI will not be able to yield power estimates. This feature is particularly useful to get early estimates of power dissipation for an already existing cache design being ported into a new technology. In *eCACTI*, we provide this flexibility to estimate power dissipation of any user-specified cache configuration.

5. Model evaluation

The ideal way to evaluate *eCACTI* is to compare its estimates with SPICE simulation based estimates for caches designed for various configurations. However, cache design and analysis is typically done on a single bank of data and tag arrays, and are replicated in the final layout. This procedure is usually employed because of the impractical run times associated with RC extraction and SPICE level simulations on the whole cache design. So we evaluate eCACTI with SPICE level simulations for cache sub-bank designs instead of the whole cache designs. Considering the regularity of the sub-banks in caches, we project that similar accuracy will hold for the whole cache design. We follow a three step methodology to validate our power estimation models as described below.

- Evaluate the analytical model for leakage current for a MOSFET shown in Equation (5).
- Evaluate the models in *eCACTI* for dynamic power dissipation.
- Evaluate the models in *eCACTI* for leakage power dissipation.

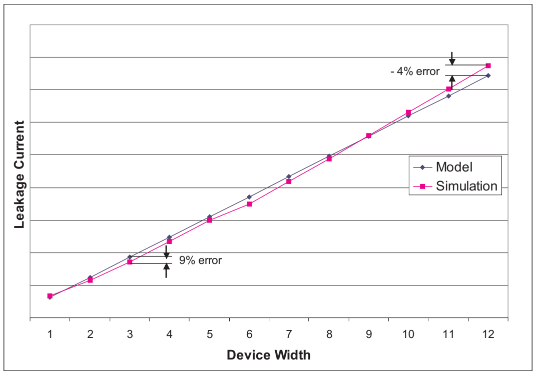


Figure 6. Plot Showing the Accuracy of Leakage Current Model for a NMOS Device

5.1. Leakage Power Model Evaluation

Figure 6 shows the evaluation of NMOS leakage current model for varying transistor widths. The model estimates are compared against simulation based estimates for 0.13μ technology transistor models from Motorola. The actual values and the process parameters cannot be published because they are Motorola proprietary data. The analytical model used for leakage current estimation for a MOSFET is accurate with an error margin of less than 9%. To eliminate this error, we proposed a LUT based leakage current estimation technique as described in Section 4.4.2.

5.2. Evaluation of Leakage and Dynamic Power Estimation in *eCACTI*

Tables 3 and 4 show the comparison of *eCACTI* based leakage and dynamic power estimates against SPICE level simulations on actual industrial designs. The designs are from the e500[2] processor core based on 0.13μ technology from Motorola. The sizes of the sub-bank design are expressed in terms of the number of memory cells, in the Column 2 of the tables. Columns 3 and 4 show the comparison of power dissipation estimates during a read and write operation respectively. The actual leakage power numbers and the names of the array designs are not shown because they are Motorola proprietary data and cannot be published. Instead, we show the percentage error between the *eCACTI* estimates and SPICE based estimates. The eCACTI dynamic power estimates were seen to have an error margin of -16.1% to +14.4% and the leakage power estimates are seen to have an error margin of -21.5% to +17.7%.

The reasons for these variations were due to:

- mismatch in the calculated device widths and the actual device widths.
- various approximations used for simplifying the analytical models.

[2] e500 is the Motorola processor core that is compliant with the PowerPC Book E architecture

Table 3. Evaluation of *eCACTI* Dynamic Power Estimates

	Design Size	Error	
	(# bit cells)	READ	WRITE
Design1	1024	+14.4%	-16.1%
Design2	5120	+0.4%	-3.1%
Design3	5888	+4.6%	+1.2%
Design4	9504	-10.5%	-6.7%

Table 4. Evaluation of *eCACTI* Leakage Power Estimates

	Design Size	Error	
	(# bit cells)	READ	WRITE
Design1	1024	4.23%	-16.61%
Design2	5120	-11.57%	-21.50%
Design3	5888	-8.59%	-16.52%
Design4	9504	17.72%	-3.06%

- various custom design optimizations for speed which are not accounted for in the model. For example, gate skewing [14] in designs leads to reduced node capacitances which affects the device width calculations leading to discrepancies in leakage and dynamic power estimates.

It can be noted that because of the reasons illustrated above, our models yield an over-estimate of leakage power in some designs and an under-estimate in some designs, depending on its implementation. However, considering that these models are based on high level design parameters with very little knowledge of the actual design, we think these error margins are acceptable and can used for early architectural exploration.

6. Comparison between CACTI and *eCACTI*

We now compare CACTI with *eCACTI* and highlight the necessity of *eCACTI* for determining the optimal cache configuration. Figure 7 shows the plot of the total cache power based on CACTI and *eCACTI* for varying cache sizes. The experiments are done assuming a direct-mapped cache with block size of 16 bytes in 0.07μ technology. The leakage and dynamic power components of the *eCACTI* power estimates are also shown in the figure. Because of the limitations and inaccuracies pointed out in CACTI, the power dissipation values are rather inaccurate for DSM technologies. For *eCACTI* power estimates, we assume that the caches use dual-V_{th} technology to reduce leakage power dissipation. Note that inspite of using dual-V_{th} technology, leakage power still dominates the total power dissipation. Apart from inaccuracies due to the lack of leakage power models in CACTI, the error due to the lack of a technique to change the device widths according to its capacitive load is observed to be significant. CACTI uses a constant for device widths, leading to constant node capacitances. Hence the dynamic power estimates are over-amplified in caches whose devices require

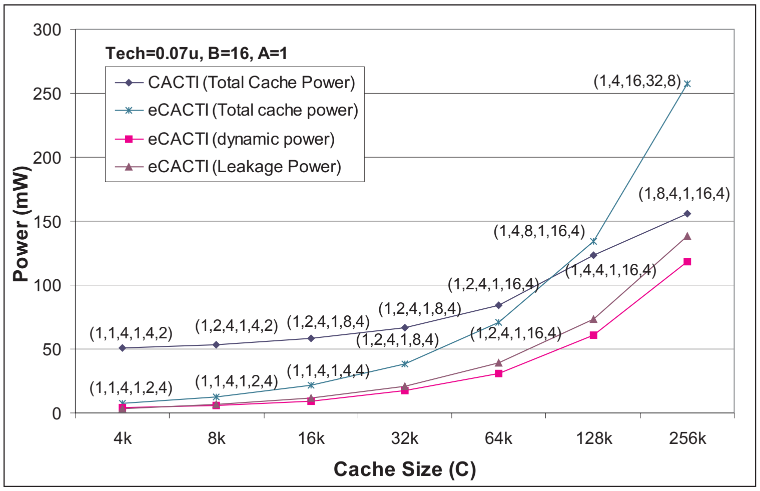


Figure 7. Comparison of CACTI and *eCACTI* Power Estimates (Tech=0.07u)

smaller gate widths than the assumed constant values. For the same reason, CACTI has very high estimates of power dissipation for smaller cache sizes and then gradually converges for larger caches. The value in the parenthesis corresponding to each node in the plot for total cache power, indicates the cache configuration that leads to lowest power in the cache. The sextuplets correspond to parameters *Ndbl, Ndwl, Nspd, Ntwl, Ntbl, and Ntspd* respectively. Note that due to inaccuracies in CACTI power estimates, *CACTI leads to an incorrect configuration for a desired optimization function*, thus highlighting the need to address the limitations in CACTI.

7. Experiments

We now present the results of various experiments conducted using our proposed tool, *eCACTI*. The experimental results include analyzing the effect of technology on leakage and dynamic power dissipation of caches, analyzing the effect of different cache parameters — cache size (C), block size (B), and associativity (A) on cache power, and analyzing the sub-block power contributions to the total cache power. For a given set of cache parameters, C, B, and A, the cache configuration yielding the lowest power is considered in all our experiments.

7.1. Effect of Technology on Cache Power

Figure 8 shows the effect of technology on cache power using *eCACTI*. The variation in leakage, dynamic, and total power dissipation are plotted for a direct-mapped cache of size 16 KB, block size of 16 bytes. As can be seen the contribution of leakage power increases exponentially with technology while the dynamic component is decreasing at a similar rate. While dynamic power dominates the total power dissipation till 0.10μ technology, leakage power is projected to dominate in technologies beyond 0.07μ. For the cache under consideration, the total power decreases till 0.10μ technology and

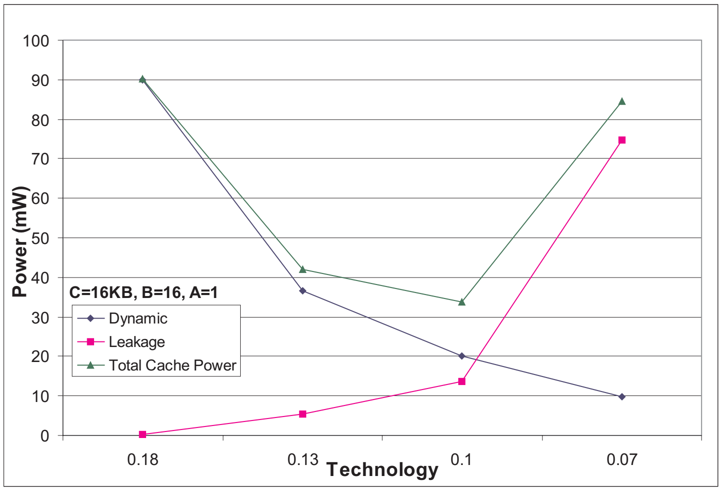


Figure 8. Plot Showing Variation of Power Dissipation with Technology

increases beyond that primarily because of the exponential increase in leakage power dissipation. So it is important for designers to devise techniques that reduce leakage power so as to decrease total power dissipation and meet the design requirements.

7.2. Effect of Cache Parameters on Total Power Dissipation

Figures 9, 10, and 11 show the effect of varying cache size, block size and associativity on power dissipation respectively. The estimates were obtained for 0.10μ technology.

Figure 9 shows the effect of cache size on the total power dissipation (block size and the associativity are kept constant). We observe that with increasing size, both the dynamic and leakage power increase at a similar rate. While the leakage power increases mainly because of the increased number of memory cells, dynamic power increases due to the increased bitline lengths in the cache sub-banks. Increased bitline lengths leads to increased bitline switching capacitance during a read/write operation thereby leading to a proportionate increase in the dynamic power dissipation.

Figure 10 shows the effect of varying block size on cache power. We keep the cache size and associativity constant for all the experimental nodes. The block size is varied from 8 bytes to 128 bytes. Since the size of the cache is the same, increasing block size leads to reduced number of sets in the cache. This leads to a decrease in leakage and dynamic power dissipation in the address decoder and wordline driver sub-blocks; and increased power dissipation in the data out buffers due to increased number of memory cells in each data array row. Overall this leads to only a slight change in the total power dissipation for increasing block size as is reflected in Figure 10. However, the decrease in leakage power with increasing block size is more prominent. This is because of the reduced tag array leakage power with increasing block size. Increasing block size has a cascading effect leading to reduced number of cache sets, reduced number of tags in the tag array and reduced number of memory cells in the tag array eventually leading to reduced leakage power in the tag array.

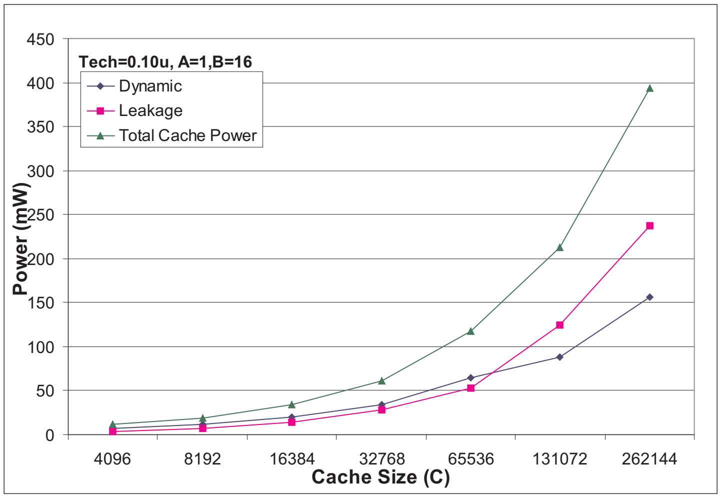


Figure 9. Plot Showing Variation of Power Dissipation with Cache Size

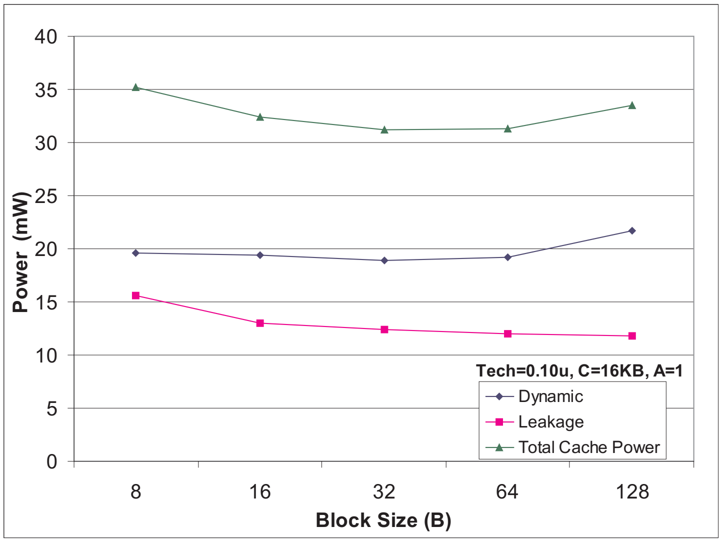


Figure 10. Plot Showing Variation of Power Dissipation with Block Size

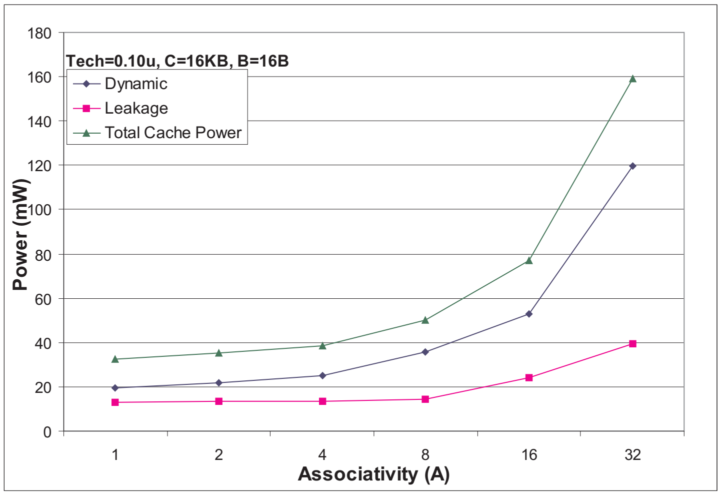


Figure 11. Plot Showing Variation of Power Dissipation with Associativity

Figure 11 shows the effect of increasing associativity for a fixed cache size and fixed block size. The associativity is varied from 1 to 32 in factors of 2. As can be noted, both the dynamic and leakage power increase significantly with increasing associativity. This was mainly because of the increased power dissipation (both leakage and dynamic) in tag comparison sub-blocks and the multiplexer select drivers corresponding to the data out multiplexers in the data array.

7.3. Sub-block Power Dissipation Contributions to Total Cache Power

Figures 12 and 13 show the sub-block power contributions to the total cache dynamic and leakage power respectively. As expected bitline sub-block power (includes the bit cell power and bitline switching power) contributes to a majority of the total power dissipation: as much as 70% of the total dynamic power and 90% of the total leakage power dissipation. However, for the dynamic power dissipation, while the transitions on the high capacitive bitlines are the major contributors, the memory bit cells contribute to a majority of the leakage power dissipation.

7.4. Effect of dual-V_{th} Optimization on Cache Leakage Power

The memory core leakage power is projected to increasingly dominate the total cache power in future technologies because of its percentage contribution to the total leakage power and exponential relationship of leakage power with technology. So researchers have proposed many techniques to reduce the leakage power dissipation in memory cores such as drowsy caches, dual-V_{th}, and reverse body biased technique. We evaluated the effect of dual-V_{th} technology on the cache power. Figure 14 shows the effect of dual-V_{th} for varying cache sizes in 0.10μ technology. Plots are shown for leakage and total cache power with and without dual-V_{th} optimization. With the use of dual-V_{th} technique, the total cache power decreases because of significant reduction in leakage power. For a 32K byte

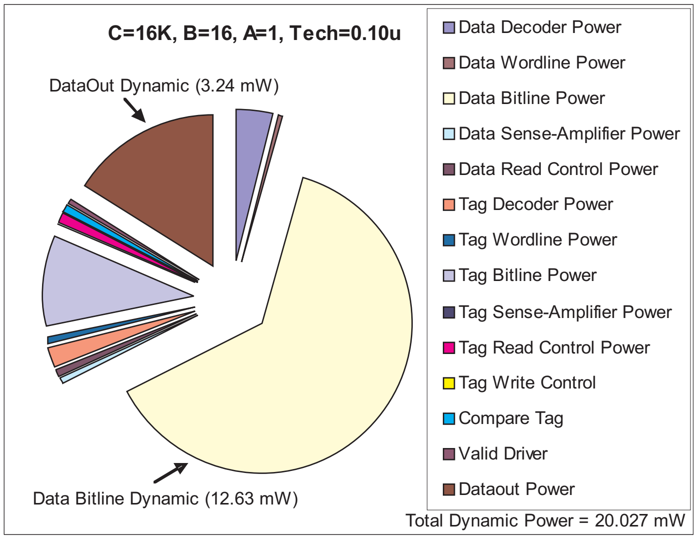


Figure 12. Sub-block Power Contributions to Total Cache Dynamic Power

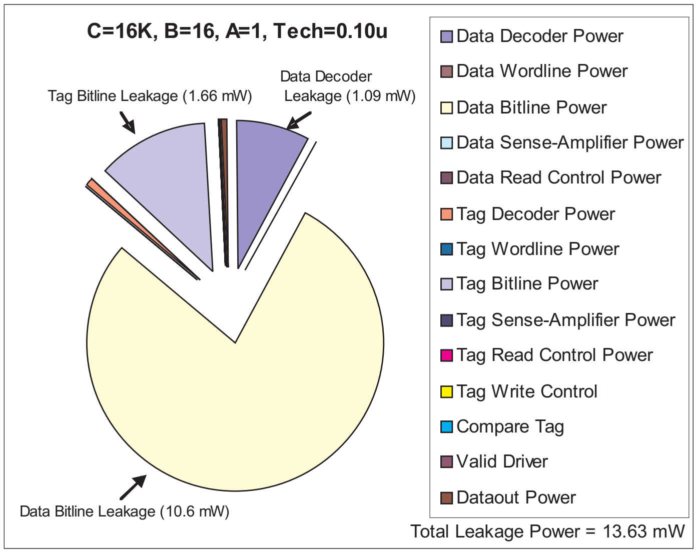


Figure 13. Sub-block Power Contributions to Total Cache Leakage Power

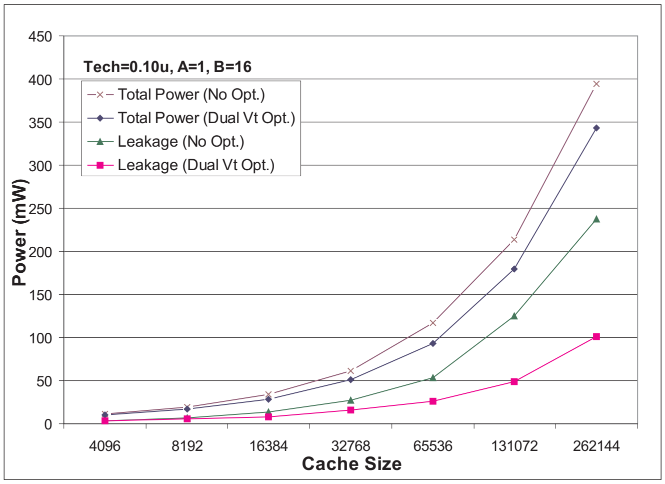


Figure 14. Leakage and Total Cache Power for Varying Cache Sizes

size cache, the leakage power reduced by 43%. This reduction is more prominent for larger cache sizes and is projected to have a dominant effect in future technologies as well. Figure 15 shows the sub-block contributions to the total leakage power considering the dual-V_{th} optimization. Comparing it to sub-block contributions without dual-V_{th} optimization (Figure 13), it was observed that the main reductions come in the tag and data memory core. This is because of using a higher threshold voltage devices in the memory core[3]. However, in cache designs based on dual-V_{th} optimizations, typically all the devices in the time critical path other than the memory cells are fabricated using low threshold voltage devices to meet the access time constraints. This can be observed in the case of the address decoder, for which leakage power increases from 1.09 mW to 2.79 mW by using dual-V_{th} optimization.

8. Summary

Caches consume a significant portion of the total system power and the leakage power component is increasing exponentially with technology. CACTI is a popular tool used by micro-architects to perform power-delay-area trade-offs. However, CACTI has a number of limitations, primary one being the lack of models to account for leakage power. In this paper, we propose a tool, *eCACTI*, which addresses all the known limitations for modeling power. We showed that CACTI does not generate the optimal cache configuration for DSM technologies, and that our enhancements in *eCACTI* is able to generate these desired optimal configurations. We also evaluated the accuracy of the power models and demonstrated the use of *eCACTI* to study the effects of (i) technology on cache leakage and total cache power, (ii) dual-V_{th} optimization on sub-block and total cache leakage power, (iii) effects of varying cache size, block size, and associativity for DSM technologies. We observed that

[3] memory core leakage is included in the bitline sub-block leakage in Figures 13 and 15.

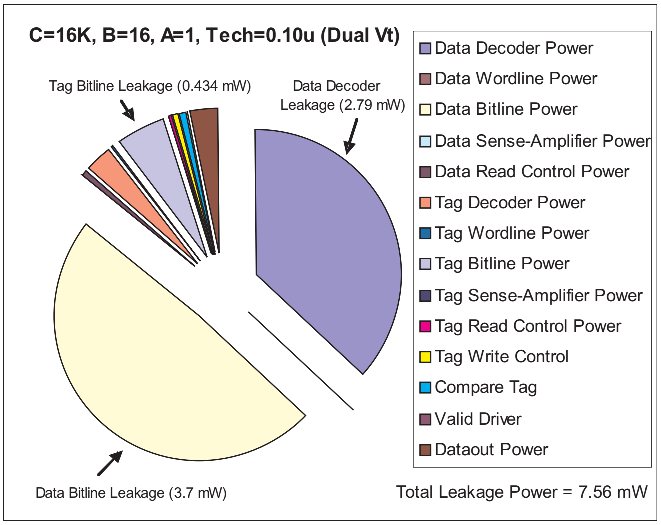


Figure 15. Sub-block Power Contributions to Total Cache Leakage Power (Dual Vt Optimization)

in 0.10μ technology, for a 32 KB cache, dual-V_{th} optimization leads to as much as 43% reduction in cache leakage power. We plan to do a public release of the *eCACTI* tool so that it can used in the research community to do better design space exploration and to evaluate various system level optimizations. We are currently working on developing power models for fully associative caches. We hope to include this feature in the upcoming public release.

References

[1] M. Mamidipaka, K. Khouri, N. Dutt, and M. Abadir Analytical Models for Leakage Power Estimation of Memory Array Structures In *International Conference on Hardware/Software and Co-design and System Synthesis (CODES+ISSS)*, 2004

[2] M. Mamidipaka, K. Khouri, N. Dutt, and M. Abadir Leakage Power Estimation in SRAMs CECS Technical Report TR 03-32, University of California, Irvine, Oct. 2003.

[3] B. S. Amrutur and M. Horowitz. A Replica Technique for Wordline and Sense Control in Low Power SRAMs. In *IEEE Journal on Solid State Circuits*, pages 1208–1219, Vol. 33, Aug. 1998.

[4] S. Borkar. Low Power Design Challenges for the Decade (Invited Talk) In *ASPDAC*, pages 293–296, 2001.

[5] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural Level Power Analysis and Optimizations. In *International Symposium on Computer Architecture*, pages 83–94, 2000.

[6] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy Caches: Simple Techniques for Reducing Leakage Power. In *ISCA*, pages 147–157, May 2002.

[7] S. Kaxiras, Z. Hu, and M. Martonosi. Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power. In *ISCA*, July 2001.

[8] J. Montanaro. et al. A 160-MHz, 32b, 0.5-W CMOS RISC microprocessor. In *IEEE JSSC*, pages 1703–1712, Nov. 1996.

[9] J. M. Mulder, N. T. Quach, and M. J. Flynn. An Area Model for On-chip Memories and its Application In *IEEE Journal of Solid-State Circuits*, pages 98–106, Vol. 26, Feb. 1991.

[10] A. Raghunathan, N. K. Niraj, and S. Dey. *High-Level Power Analysis and Optimization*. Kluwer Academic Publishers, 1998.

[11] G. Reinman and N. Jouppi. *CACTI 2.0: An Integrated Cache Timing and Power Model*. WRL Research Report 2000/7, Feb. 2000.

[12] P. Shivakumar and N. Jouppi. *CACTI 3.0: An Integrated Cache Timing, Power, and Area Model*. WRL Research Report 2001/2, Aug. 2001.

[13] I. Sutherland, R. Sproull, D. Harris, and R. Sproull. *Logical Effort : Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.

[14] T. Thorp, G. Yee, and C. Sechen. Design and Synthesis of Monotonic Circuits. In *International Conference on Computer Design*, 1999.

[15] N. Vijaykrishnan et al. Energy-Driven Integrated Hardware-Software Optimizations using Simplepower. In *ISCA*, pages 95–106, 2000.

[16] T. Wada, S. Rajan, S. A. Przybylski. An Analytical Access Time Model for On-Chip Cache Memories. In *IEEE Journal of Solid-State Circuits*, pages 1147–1156, Vol. 27, Aug. 1992.

[17] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley, 1985.

[18] S. Wilton and N. Jouppi. *An Enhanced Access and Cycle Time Model for On-chip Caches*. WRL Research Report 93/5, Jun. 1994.

[19] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. Hotleakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. TR-CS-2003-05, Univ. of Virginia, Dept. of Computer Science, Mar. 2003.