Analytical Models for Leakage Power Estimation of Memory Array Structures *

Mahesh Mamidipaka † Kamal Khouri ‡ Nikil Dutt † Magdy Abadir ‡

† Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697, USA
maheshmn,dutt@cecs.uci.edu

‡ Global Strategy and Future Tools & Meth.
Freescale/Motorola Inc.
Austin, TX 78729 USA
kamal.khouri,m.abadir@freescale.com

ABSTRACT

There is a growing need for accurate power models at the system level. Memory structures such as caches, Branch Target Buffers (BTBs), and register files occupy significant area in contemporary SoC designs and are the main contributors to system leakage power dissipation. Existing models for leakage power estimation in array structures typically use coefficients derived from elaborate SPICE simulations. However, these methodologies are not applicable to array designs in a newer technology, that require power estimates early in the design cycle. In this paper, we propose analytical models for array structures that are based only on high level design parameters. Assuming typical circuit implementation styles, we identify the transistors that contribute to the leakage power in each array sub-circuit and develop models as a function of the operation (read/write/idle) on the array and organizational parameters of the array. The developed models are validated by comparing their estimates against the leakage power measured using SPICE simulations on industrial array designs belonging to the e500[1] processor core. The comparison shows that the models are accurate with an error margin of less than 21.5% and thus can be used in high-level power-performance exploration. Interestingly, in array designs with dual threshold voltage technology, we observed that contrary to the general expectation, the array memory core contributes to just 9% and the address decoder contributes to as much as 62% of the total leakage power.

Categories and Subject Descriptors: B.3 [Hardware]: Memory Strucutures

General Terms: Algorithms, Measurement

*This work was done in collaboration with Motorola Inc. and partially supported by NSF grants CCR 0203813 and CCR 0205712

[1]e500 is the Motorola processor core that is compliant with the PowerPC®Book E architecture. The "PowerPC" name is a trademark of IBM corporation and is used under license.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CODES+ISSS'04, September 8–10, 2004, Stockholm, Sweden.
Copyright 2004 ACM 1-58113-937-3/04/0009 ...$5.00.

Keywords: Leakage power, SRAMs, Estimation.

1. INTRODUCTION

Power dissipation which was previously considered an issue only in portable devices is rapidly becoming a significant design constraint in many system designs. Dynamic power has been a predominant source of power dissipation till recently. However, static power dissipation is becoming an significant fraction of the total power. The absolute and the relative contribution of leakage power to the total system power is expected to further increase in future technologies because of the exponential increase in leakage currents with technology scaling. The International Technology Roadmap for Semiconductors (ITRS) [9] predicts that leakage power would contribute to 50% of the total power in the next generation processors. It is known that design decisions taken higher in design cycle have greater influence on the system power dissipation. Therefore, it is important for system designers to get an early estimate of leakage power to meet the challenging power constraints.

Memory array structures such as caches (tag and data arrays), branch target buffers, reservation stations, etc. and occupy significant portion of the die area in contemporary designs. Expectedly, arrays contribute to majority of the leakage power in system designs. However, system designers currently do not have the ability to perform early estimation of such leakage power. A lot of research has been done on leakage power estimation at the gate level for combinational logic. But these methodologies cannot be applied to arrays because of their inherent transistor level design that cannot be represented at gate level. While there are estimation models for leakage power in memory structures, most of them require pre-characterized data from SPICE simulations. This pre-characterization is possible only for already existing designs and not applicable to new designs in a different technology that might require early estimates of power dissipation. We think, ours is the first work which proposes comprehensive analytical models for leakage power estimation in memory arrays without the need of any precharacterized data. We develop models parameterized in terms of the structure of the array (number of rows, columns, read multiplexer size, and write multiplexer size) and the operation on the array. Such models would greatly benefit to system designers in: (a) quantifying the static power early in the design cycle (b) performing power-performance trade-off analysis of different array configurations and (c) evaluating the dependencies of various micro-architecture level parameters on the static power dissipation.

For industrial designs, a comparison of the model estimates with SPICE simulation based estimates show that the models are accurate with an error margin of less than 21.5%. Also, we observed that in designs with aggressive dual threshold voltage (dual-V_{th}) technology [12], unlike typical arrays, memory core comprising of bit cells contributes to only 9% of the total array leakage power. The majority of leakage power in such designs is actually contributed by the address decoder, read, and write control logic sub-blocks. the leakage power.

The paper is organized as follows. Section 2 presents related

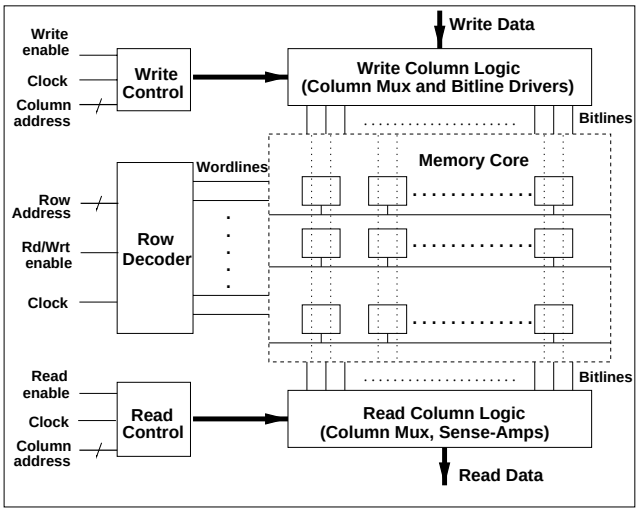


Figure 1: Typical Architecture of Array Structures

work and Section 3 presents the details about the sub-blocks involved in the implementation of conventional arrays. Section 4 presents our analytical models for leakage power estimation in arrays. We illustrate the methodology used for transistor width determination in Section 5. Section 6 shows the accuracy of the proposed models by comparing its estimates against SPICE level simulation based estimates on industrial designs and Section 7 concludes this paper.

2. RELATED WORK

Static power estimation has been an area of research interest for quite a long time. The focus however, was primarily on developing techniques at gate level [10, 4]. Bobba and Hajj [1] developed techniques for estimation of maximum leakage power in combinational logic based on simulation. Also leakage power estimation in ASIC design environment based on library characterization was proposed in [5]. However, these techniques are not applicable to memory array structures because of their inherent transistor level design.

Recently, more attention is being paid to leakage power estimation at higher level of design hierarchy. Butts and Sohi [2] propose a generic model for micro-architectural components. The model in this work is based on a key design parameter, K_{design}, captures device types(PMOS/NMOS), device geometries (W/L), and stacking factors and can be obtained based on simulations. Liao et al. [6] proposed models for leakage power estimation in memory arrays. The models were parameterized in terms on words, word sizes, and a set of coefficients derived using SPICE simulations. A methodology for estimation of leakage power for micro-architectural components in interconnection networks is proposed by Chen et al. [3]. The methodology is based on simulation of fundamental circuit components for various input states. Zhang et al. [15] develop an architectural model for sub-threshold and gate leakage that explicitly captures in temperature, voltage, gate leakage, and parameter variations. To the best of our knowledge, ours is the first attempt to develop comprehensive analytical models for leakage power estimation in memory arrays. Unlike many previous methods our models use high level design parameters as input and do not require any SPICE simulation based pre-characterization.

3. ARRAY STRUCTURES

Arrays contribute to a significant portion of the total system power dissipation. Caches, tag arrays, register files, branch table predictors, instruction windows, translation lookaside buffers are common array structures in contemporary micro-processors. Figure 1 shows a typical structure of an array. It is primarily composed of the following main sub-blocks: address decode logic, memory core, read column logic, write column logic, read control, and write control logic.

Arrays typically support read and write operations [13]. After a read/write has been performed, the bitlines are precharged to supply voltage (referred to as *precharge* phase) thereby getting ready for another read/write in the next cycle. Typically, in an array clock cycle, while read/write is performed in the first phase (referred to as *read/write* phase) of the clock cycle, precharge is performed in the second phase. Bitline precharge is done independent of the operation in the first phase of the clock cycle. If there is no operation being performed in a clock cycle, all the wordlines remain deactivated (logic LOW) and the bitlines stay precharged (logic HIGH). We refer to this no operation phase as *idle* phase.

The leakage current in arrays vary within a clock cycle depending on the phase of the operation being performed, since different transistors would be in off state during different operations. In the following section we propose analytical models for leakage current in arrays during each phase: read, write, precharge, and idle.

4. ANALYTICAL MODELS FOR MEMORY ARRAY LEAKAGE CURRENT ESTIMATION

In this section we develop leakage power models parameterized in terms of high level array design parameters (shown in Table 1) and technology parameters.

Table 1: Leakage Power Model Inputs

Parameter	Description
N_{rows}	Number of rows in the array
N_{cols}	Number of columns in the array
S_{rdMux}	Size of read column multiplexer
S_{wrtMux}	Size of write column multiplexer

As indicated in Section 3, array structures are primarily composed of 6 sub-blocks: memory-core, address decoder, read column circuit, write column circuit, read control and write control circuit. We consider the typical implementation styles of these sub-blocks and develop leakage power models for each sub-block and for each of its operational phase (read, write, precharge, and idle). The models for the write column circuit sub-block are similar to that of read column sub-block and are not given in this paper because of the space limitation, but are given in our technical report [7]. Note that similar analysis can be used to develop analytical models for other implementation styles of array sub-blocks. To simplify the analysis, we assume that the leakage current in a sub-block during a transient state is the same as the leakage current when it reaches a steady state. We show in Section 6 that the error introduced due to this approximation is reasonable. We first describe the MOSFET leakage current model used in our sub-block power models before going into the details of the sub-block power models.

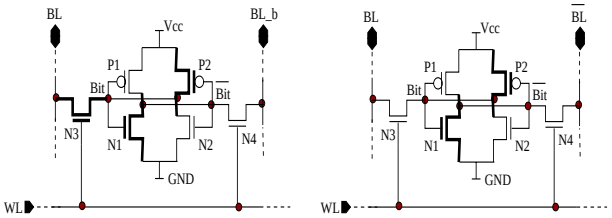
4.1 MOSFET Leakage Current Model

The basic methodology used to develop leakage power models is to identify the transistors contributing to leakage current and summing the leakage currents in each of these transistors. To calculate the leakage current in a MOSFET, we use the model proposed by Zhang et al. [15] and is shown in Equation (1).

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

$$I_{lkg} = W \cdot I_l(T, V_{th}) \quad (2)$$

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 (1) can be reduced to Equation (2), where I_l is the leakage of a unit



(a) Idle phase: WL=0, Bit=0, BL=1, BL_b=1 (b) Read phase: WL=1, Bit=0

Figure 2: Leaking memory cell transistors in various operational phases (leaking transistors in bold)

width transistor at a given temperature and threshold voltage. When there are stacks of transistors (transistors connected in series drain to source) in a design, it was observed in [4] leakage current reduces significantly. In such cases, to account for the reduction in leakage current, we use stacking factors derived based on the knowledge of previous designs. We now present the derivation of sub-block leakage power models.

4.2 Memory Core

The memory core is composed of memory cells that are arranged in rows and columns. Figures 2(a) and 2(b) show 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 $I_{Dsub}(N1) = I_{Dsub}(N2); I_{Dsub}(N3) = I_{Dsub}(N4); I_{Dsub}(P1) = I_{Dsub}(P2)$.

During an idle phase, the wordlines are deselected ($WL = 0$) and the bitlines are precharged ($BL = 1, BL\_b = 1$). Depending on the memory cell data, either transistors N4, P1, N2 (for Bit = 1) or N3, P2, N1 (for Bit = 0) will be in the off-state. Figure 2(a) shows the transistors in off-state in bold for Bit = 0. Because of the symmetry of the memory cell design, independent of the data in the memory cell, the leakage current of the memory cell in idle phase would be as shown in Equation (3). Equation (4) can be obtained by substituting Equation (2) in Equation (3) where W_{N3}, W_{P2}, W_{N1} are widths of N3, P2, and N1 respectively, and I_{lN}, I_{lP} are the leakage current per unit width for NMOS and PMOS transistors for a given threshold voltage and temperature. For a memory core with N_{rows} rows and N_{cols} columns (i.e., $N_{rows} \cdot N_{cols}$ memory cells), the total leakage of the memory core in the idle phase can thus be obtained as shown in Equation (5).

$$I_{memCellIdle} = I_{Dsub}(N1) + I_{Dsub}(N3) + I_{Dsub}(P2) \quad (3)$$

$$= (W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP} \quad (4)$$

$$I_{memCoreIdle} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad (5)$$

During the read phase, one of the wordlines is activated in accordance with the address and the remaining wordlines remain deactivated. Then corresponding to data in each memory cell of the selected row, one of the bitlines in all the bitline pairs (BL, BL_b), discharges partially. For simplicity of the analysis, we assume that the amount of discharge in the bitline is negligible and treat both BL and BL_b to be at V_{cc} during read phase as well. Considering the symmetry of the transistors, the leakage current in the memory cell in the two scenarios, $WL = 1$ and $WL = 0$, is shown in Equation (6). The transistors leaking during read phase with $WL = 1$ and $Bit = 0$ are shown in Figure 2(b). 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 read phase can be derived as shown in Equation (7).

$$I_{memCellRd} = \begin{cases} (W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP} & \text{for WL=0} \\ W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP} & \text{for WL=1} \end{cases} \quad (6)$$

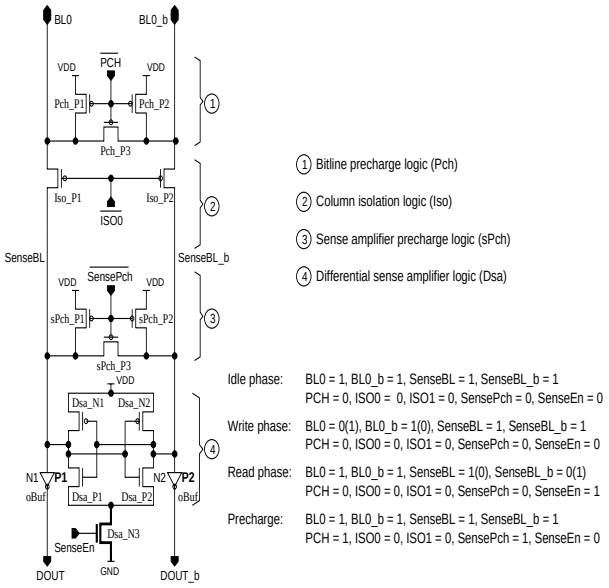


Figure 3: Schematic of a differential read column circuit

$$I_{memCoreRd} = N_{rows} \cdot N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) + (N_{rows} - 1) \cdot N_{cols} \cdot W_{N3} \cdot I_{lN} \quad (7)$$

The analytical equations for leakage in the memory core for precharge and write phases can similarly be derived and are seen to be same as the models for idle and read phases respectively as shown in Equations 8 and 9. Because of the space limitation the detailed analysis is omitted here but is given in our technical report [7]. Since the product of rows and columns in the memory core ($N_{rows} \cdot N_{cols}$) is much greater than number of columns (N_{cols}), Equation (9) can be reduced to Equation (10). This means that the leakage current in the memory core can be considered independent of the array operational phase.

$$I_{memCore} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \text{for idle or precharge phase} \quad (8)$$

$$I_{memCore} = N_{rows} \cdot N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) + (N_{rows} - 1) \cdot N_{cols} \cdot W_{N3} \cdot I_{lN} \quad (9)$$

$$I_{memCore} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \text{for read or write phase} \quad (10)$$

$$I_{memCore} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \text{for read or write or idle or precharge phase} \quad (11)$$

4.3 Read Column Circuit

The read column circuit is typically composed of bitline precharge logic, isolation logic, differential sense amplifier, precharge logic for sense bitlines and buffers driving the data output. Figure 3 shows the schematic of a differential sense amplifier based read column logic.

In the idle phase, the bitlines, sense bitlines are precharged and the sense enable, sense precharge, precharge, and isolation signals are deselected (logic LOW). The leakage current in the idle phase is contributed by the sense enable transistor and PMOS transistors in the output buffers as highlighted in Figure 3. The signal values in various phases of sub-block operation are shown in the right bottom corner of Figure 3. Note that in a read phase, the

isolation transistors are active for a small period of time so that the differential sense amplifier samples the bitline voltages. In a read phase, as indicated in previous subsection, we make an approximation that both bitlines are at logic HIGH although one of the bitlines discharges partially. Analyzing the basic schematic under these conditions, and using Equation (2), the leakage current in idle, write, and read phases and for the whole read column sub-block are shown in Equations 12, 13, 14, and 15 respectively. S_{rdMux} and S_{wrtMux} indicate the size of the read column multiplexer and write column multiplexer respectively.

$$\begin{aligned} I_{rdColIdle} &= I_{rdColPch} \\ &= I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1} \\ &= W_{Dsa\_N3} \cdot I_{lN} + 2 \cdot W_{oBuf\_P1} \cdot I_{lP} \end{aligned} \quad (12)$$

$$\begin{aligned} I_{rdColWrite} \\ &= 2 \cdot I_{Pch\_P1} + I_{Iso\_P1} + I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1} \\ &= I_{lP} \cdot (2 \cdot W_{Pch\_P1} + 2 \cdot W_{oBuf\_P1} + W_{Iso\_P1}) \\ &\quad + I_{lN} \cdot W_{Dsa\_N3} \end{aligned} \quad (13)$$

$$\begin{aligned} I_{rdColRead} \\ &= 2 \cdot I_{sPch\_P1} + I_{Iso\_P1} + I_{Dsa\_N1} + I_{Dsa\_P2} \\ &\quad + I_{oBuf\_P1} + I_{oBuf\_N2} \\ &= I_{lP} \cdot (2 \cdot W_{sPch\_P1} + W_{Dsa\_P2} + W_{oBuf\_P1} + I_{Iso\_P1}) \\ &\quad + I_{lN} \cdot (I_{Dsa\_N1} + I_{oBuf\_N2}) \end{aligned} \quad (14)$$

$$\begin{aligned} I_{rdCol} \\ &= \frac{N_{cols}}{S_{rdMux}} \cdot I_{rdColIdle} \qquad \text{for idle or precharge phase} \\ &= 2 \cdot N_{cols} \cdot I_{Pch\_P1} + \frac{N_{cols}}{S_{wrtMux}} I_{Iso\_P1} \\ &\quad + \frac{N_{cols}}{S_{rdMux}} (I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1}) \quad \text{for write phase} \\ &= \frac{N_{rows}}{S_{rdMux}} \cdot I_{rdColRead} \qquad \text{for read phase} \end{aligned} \quad (15)$$

4.4 Address Decoder

We assume that the decoder architecture is composed of 3 stages as shown in Figure 4. The first stage is a set of 3-8 decoders acting as a predecoder to the row address. If the number of address bits is not divisible by three, then a 2-4 decoder can be used to make up the difference. The second stage combines the outputs of the 3-8 decoders using NOR gates to generate an output corresponding to each row of the memory array. Note that each NOR gate must take an input from each 3-8 decoder; thus having N_{3to8} inputs (where N_{3to8} is the number of required 3-8 decoders). The final stage of the address decoder is an inverter that drives each wordline driver. This architecture is similar to the address decoder architecture considered in CACTI [14] for cache delay estimation. For a given number of rows (N_{rows}) in the memory core, the number of address bits (N_{addr}), the number of 3-8 decoders (N_{3to8}), the number NOR gates (N_{nor}), the number of inverters (N_{inv}) and the number of wordline drivers (N_{wlDrv}) required for the implementation of address decoder are given in equations 17 and 18 respectively. To make the analysis simpler we assume that N_{addr} is divisible by 3.

$$N_{addr} = log_2^{N_{rows}} \quad (16)$$

$$N_{3to8} = \frac{N_{addr}}{3} \quad (17)$$

$$N_{nors} = N_{inv} = N_{wlDrv} = N_{rows} \quad (18)$$

Each 3-8 decoder is typically implemented using eight NAND gates and three inverters to complement the address inputs. The enable signal is activated during the read/write phases of the array operation thereby triggering the 3-8 decoder outputs. The NOR gate outputs is active low during idle and precharge phases and during read/write phases one of the NOR gates is pulled active HIGH. Similarly the inverter and wordline drivers in the decoder third stage have outputs active HIGH and active LOW

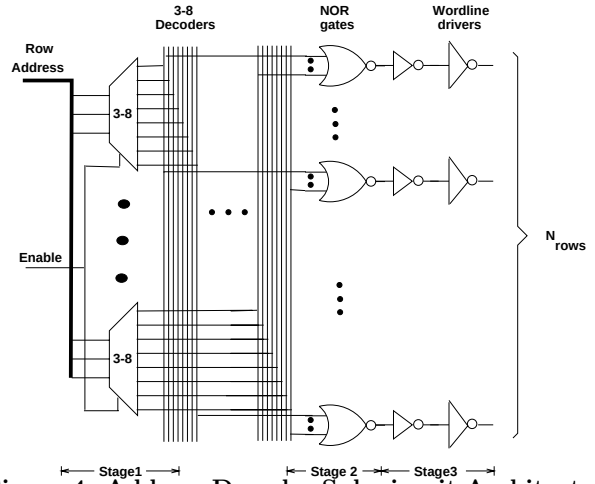


Figure 4: Address Decoder Sub-circuit Architecture

respectively during idle and precharge phases; during read/write phase, one of the inverter and wordline driver switch to active LOW and active HIGH respectively.

The leakage current in the third stage of the decoder can be modeled as shown in Equation (19), where W_{inv_N} and W_{inv_P} are widths of inverter NMOS and PMOS transistors, W_{wlDrv_N} and W_{wlDrv_P} are the widths of wordline driver NMOS and PMOS transistors. However, the leakage in second and first stages of the decoder cannot be accurately determined because of the random nature of the address to the decoder. We use probabilistic approaches to model the leakage currents in second and first stages corresponding to equations 20 and 21 respectively. We assume that all possible inputs to these gates have equal probability. Hence the leakage current in each gate is calculated as the average of the leakage current due to each possible input.

$$\begin{aligned} I_{stage3} &= N_{rows} \cdot (I_{inv} + I_{wlDrv}) \\ &= (N_{rows} - 1) \cdot (W_{inv\_N} \cdot I_{lN} + W_{wlDrv\_P} \cdot I_{lP}) + \\ &\quad (W_{inv\_P} \cdot I_{lP} + W_{wlDrv\_N} \cdot I_{lN}) \\ &\qquad \text{for write and read phases} \\ &= N_{rows} \cdot (I_{lN} \cdot W_{inv\_N} + I_{lP} \cdot W_{wlDrv\_N}) \\ &\qquad \text{for precharge and idle phases} \end{aligned} \quad (19)$$

$$I_{stage2} = N_{rows} * (I_{nor}) \quad (20)$$

$$I_{stage1} = N_{3to8} * (3 * I_{addrInv} + 8 * I_{nand}) \quad (21)$$

$$I_{dec} = I_{stage1} + I_{stage2} + I_{stage3} \quad (22)$$

Finally, the leakage current for the whole decoder can be obtained by summing the leakage currents of the three stages as shown in Equation (22).

4.5 Read and Write Control Circuits

Unlike regular structures (such as the memory core, read column and write column circuits), control circuits do not have a basic block which is replicated. For these blocks, we analyzed the critical contributors of leakage power through simulations on existing designs to develop their analytical models.

The read and write control blocks drive the signals that go across the read column and write column circuitry respectively. For example, the read control logic drives the signals controlling the precharge, differential sense-amplifier logic in the read logic for each column of the memory core. It is observed that the main contribution of leakage in these blocks comes from the buffers driving these long signal lines traversing the width of the memory core. Moreover, leakage power estimates using SPICE simulations on these blocks for 7 different array designs showed that leakage of the whole block is 1.3-1.5 times the leakage of

the circuit output drivers. This observation is not completely unexpected because the size of most logic gates in all these control circuits is driven by the size of the output drivers. The additional logic that these circuits may have, contribute to an insignificant amount of leakage power in these circuits. So the leakage power for these circuits can be obtained as shown in Equation (23), where 1.4 is the empirical value calculated as the average of all the measurements using SPICE simulations. Although this is an emperically derived value, this will not change with technology because it is dependent on the typical nature of the design (transistor sizing) rather than on the process parameters. So the same emperical value can be used for future technologies and new array designs without doing any simulations.

$$I_{cntlLkg} = 1.4 \cdot \sum_{i} I_{oBuf_i} \quad (23)$$

The output drivers for read control includes sense enable driver (senseEnDrv), precharge driver (PchDrv), sense precharge driver (sPchDrv), and isolation drivers (isoDrv). The number of isolation drivers correspond to the size of the read column multiplexer (S_{rdMux}). The write control logic compromises of write multiplexer drivers and its associated combinational logic. The detailed analytical models for these read and write control blocks are given in our technical report [7].

Using the sub-block analytical models, the total array leakage power in each phase can be computed as the sum of the leakage power of sub-blocks as shown in Equation (24). Since each array operational cycle is composed of an operational phase (opPh) and a precharge phase (pchPh) the average leakage current in an operational cycle can be calculated as shown in Equation (25). The leakage power for an array operation can then be computed as shown in Equation (26).

$$\begin{aligned} I_{array} &= I_{memCore} + I_{rdCol} + I_{wrtCol} \\ &\quad + I_{dec} + I_{rdCntl} + I_{wrtCntl} \quad (24) \\ I_{arrayOp} &= 0.5 \cdot (I_{opPh} + I_{pchPh}) \quad (25) \\ P_{arrayOp} &= I_{arrayOp} \cdot V_{dd} \quad (26) \end{aligned}$$

5. DEVICE WIDTH CALCULATION

As can be noted from the previous section, the analytical models for leakage power in arrays depend on the device widths. Hence, for early estimation of leakage power, it is necessary to determine the transistor widths using high level design parameters. In this section, we present a methodology that can be used for calculating the device widths based on high level design parameters. The methodology is similar to the one used for dynamic power estimation of array structures in [8] and for delay estimation of caches in CACTI [14]. Similar to these works, the methodology makes the following assumptions for determining the device widths: (a) The effective size of PMOS transistor in a logic gate is assumed to be twice the effective size of NMOS transistors. (b) We assume that the size of devices in a memory cell and the dimensions of the memory cell are known a priori. It is very often the case that the memory cells are designed much earlier than the design of the memory array. (c) The technology dependent parameters and clock frequency of array operations are assumed to be provided by the user.

Figure 5 shows the flow used for capacitive width calculation, leading to leakage power estimation. Since the size of the devices depend on the capacitive loads driven by them, the methodology starts by calculating the capacitive loads on these devices. Then the methodology uses the a set of analytical models for determining the device sizes. Since the capacitive load determination of some nodes might require the knowledge of width of certain transistors, the device width and capacitive load calculation is an iterative process that continues till all the required transistor widths are determined. For example, for calculation of the width of the bitline precharge logic, the capacitive load on the bitline needs to be calculated as shown in Equation (27), where C_{metal} indicates the metal capacitance per unit micron, $H_{memCell}$ indicates the height of the memory cell in microns, C_{drain} indicates the drain capacitance per unit micron. The width of the PMOS precharge transistor (W_{pmos}) transistor can then be calculated

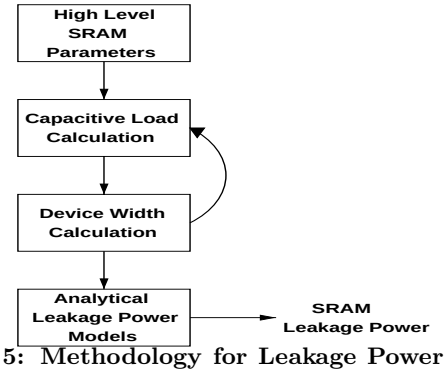


Figure 5: Methodology for Leakage Power Estimation in Array Structures

as a function of bitline capacitance (C_{BL}) and precharge time ($T_{precharge}$). $T_{precharge}$ is derived as a fraction of the frequency of operation. The precharge transistor width is then used for deriving the capacitive load on the precharge driver in the read control logic for calculation of its device sizes.

$$C_{BL} = N_{rows}.(C_{memCell} + C_{metal}.H_{memCell}) + 3.C_{drain} \quad (27)$$

$$W_{pmos} = f(C_{BL}, T_{precharge}) \quad (28)$$

Once all the required transistor widths are calculated, these are used in the leakage power analytical models illustrated in Section 4 for obtaining leakage power estimates in memory array structures.

6. MODEL EVALUATION

In this section we show the results of the evaluation of the analytical power estimates with those based on SPICE simulations. Although we showed the analytical models for typical sub-block implementation styles in this paper, we developed models for various other standard sub-block implementation styles and present their evaluation in this section. The SPICE simulations are done on a transistor-level netlist with RC back annotation obtained from layout. The leakage power values are calculated as the average power for a large number of input stimulus. This stimulus was obtained from the benchmarks: dhrystone, goke_fft, and 6 Motorola internal benchmarks.

Table 2 shows the comparison across different industrial array designs used in a e500 processor core based on 0.13μ technology. 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 model estimates and SPICE. Column 2 indicates the size of the array in terms of the number of bit cells, Columns 3, 4, and 5 indicate the percentage error in the model estimates for idle, read, and write operation cycles respectively. The percentage error is calculated as ($model\_value - actual\_value$)/$actual\_value$ where, the $actual\_value$ is the value obtained from SPICE. These arrays differ from each other in size, row/column organization, number of memory bit-cell ports (single read/write, multiple read/write, and dedicated read/write), memory bit- cell dimensions, read logic styles, write logic styles, and self-timed read logic styles. For example, Arrays 1 and 2 have separate read and write ports for simultaneous read and write accesses. While the write operation was implemented using single ended bitline and static inverter based write logic, the read operation was implemented using double ended bitline and inverter based sense-amplifier. Arrays 4-7 mostly correspond to the typical implementation styles illustrated in the Section 4. From Table 2 the error margin varies from -21.5% to +17.7%. The reasons for variation were due to:

- error in the leakage current model of a MOSFET. To evaluate the possible error due to the MOSFET model, we ran simulations for a NMOS device for varying gate widths and compared their leakage currents against SPICE based estimates. Figure 6 shows this comparison. It was observed

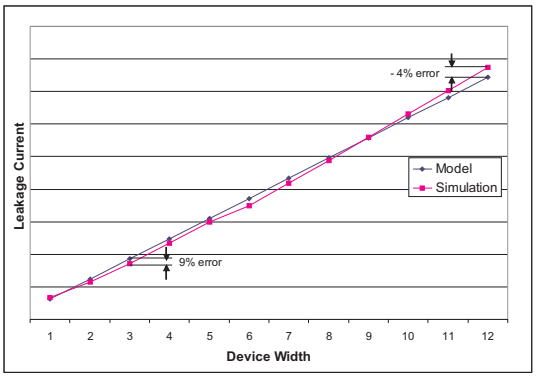


Figure 6: Plot Showing the Accuracy of Leakage Current Model for a NMOS Device

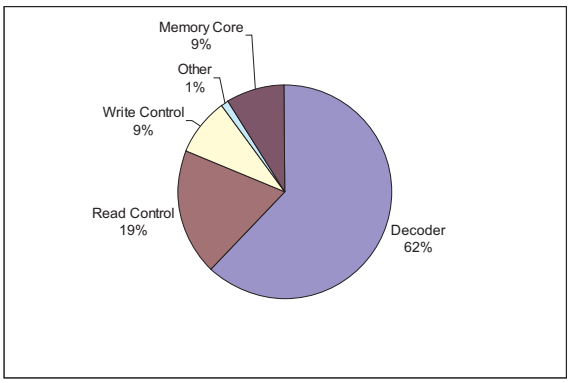


Figure 7: Percentage Contributions of a 64x80 Array with dual-V_{th} Technology

that the maximum error in this model is 9% and the average error is observed to be 2%.

- mismatch in the calculated device widths and the actual device widths.
- various approximations used for simplifying the analytical models.
- various custom design optimizations for speed which are not accounted for in the model. For example, gate skewing [11] in designs leads to reduced node capacitances.

It can be noted that because of the reasons illustrated above, the models yield to an over-estimate of leakage power in some array 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 margin is reasonable and can used for various exploration purposes. Also we observed that the leakage power values during various modes of operation (read/write/idle) vary by as much as 32%, highlighting the need for models for each operational state.

Table 2: Leakage Power Models Vs SPICE

	Array Size (# of cells)	Error		
		IDLE	READ	WRITE
ARRAY1	352	12.20%	-6.22%	-3.15%
ARRAY2	704	15.47%	9.20%	-0.11%
ARRAY3	1024	11.03%	4.23%	-16.61%
ARRAY4	1536	-3.21%	-10.17%	3.62%
ARRAY5	5120	-16.31%	-11.57%	-21.50%
ARRAY6	5888	-18.61%	-8.59%	-16.52%
ARRAY7	9504	-0.23%	17.72%	-3.06%

Figure 7 shows the sub-block leakage power contributions for an array with 64 rows and 80 columns and sub-blocks implemented using typical styles described in Section 4. The results are shown for a read operation but the contributions are similar for write and idle operations as well. To reduce leakage power dissipation this array design was optimized using aggressive dual-V_{th} technology. To meet the stringent access time constraints, low threshold voltage devices were used in the time critical paths. High threshold voltage devices were used in memory bit cells and non-time critical paths of the array. In these arrays, the leakage power dissipation component to the total power in various operation states varied between 12% to 32%. More analysis on the leakage and dynamic power components in relation to the size and implementation styles of the arrays is given in [7]. The leakage current in write and read column logic is negligible ($< 1\%$) because of the fewer number of transistors in off-state. Although the memory core consumes the most area of the array, the leakage power contribution of the memory core is only 9%. This is because of the being the use of high thereshold devices in the memory bit cells. The main contributor of leakage power is the decoder with 62% of the total array leakage power because of the low threshold voltage devices used for the huge wordlines drivers in the decoder.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented analytical models for leakage power estimation of memory array structures early in the design cycle. The models are based on the high level array parameters such as number of rows, number of columns, read column multiplexer size and write column multiplexer size of the array along with the technology parameters. The analytical models were evaluated by comparing against detailed SPICE simulations on leading industrial designs. The error margin is seen to be less than 21.5%. Since the models give the leakage power contributions of each sub-block, they can be used by system architects to identify the sub-blocks with most leakage power for use of optimization techniques. Future work includes enhancing the analytical models to increase the accuracy of the estimates. Also, we plan to extend these models so as to estimate leakage power in caches for a given configuration.

8. REFERENCES

[1] S. Bobba et al. Maximum Leakage Power Estimation for CMOS Circuits. In *IEEE Workshop on Low-Power Design*, 1999.

[2] J. A. Butts and G. S. Sohi. A Static Power Model for Architects. In *Intl. Symposium on Microarchitecture*, 2000.

[3] X. Chen and L. Peh. Leakage Power Modeling and Optimization in Interconnection Networks. In *ISLPED*, 2003.

[4] M. Johnson et al. Models and Algorithms for Bounds on Leakage in CMOS Circuits. *IEEE TCAD* , 1999.

[5] R. Kumar et al. Leakage Power Estimation for Deep Submicron Circuits in an ASIC Design Environment In *ASP-DAC*, 2002.

[6] W. Liao et al. Microarchitecture Level Power and Thermal Simulation Considering Temperature Dependent Leakage Model In *ISLPED*, 2003.

[7] M.Mamidipaka et al. Leakage Power Estimation in SRAMs. CECS Technical report, TR 03-32, UC Irvine, Oct. 2003.

[8] M. Mamidipaka et al. IDAP: A Tool for High-level Power Estimation of Custom Array Structures. In *ICCAD*, 2003.

[9] SIA. International Technology Roadmap for Semiconductors (ITRS). Technical report, http://public.itrs.net/.

[10] S. Sirichotiyakul, et al. Duet: An Accurate Leakage Estimation and Optimization Tool for dual-Vt Circuits. *IEEE TVLSI*, 2002.

[11] T. Thorp, G. Yee, and C. Sechen. Design and Synthesis of Monotonic Circuits. In *ICCD* , 1999.

[12] L. Wei et al. Design and Optimization of Dual Threshold Circuits for Low Voltage Low Power Applications. *IEEE TVLSI*, pages 16–24, 1999.

[13] N. Weste and K. Eshragian. *Principles of CMOS VLSI Design, A Systems Perspective*. Addison-Wesley, 1998.

[14] S. Wilton et al. An Enhanced Access and Cycle Time Model for On-chip Caches. WRL Research Report 93/5, June, 1994.

[15] Y. Zhang et al. Hotleakage: A Temperature-aware Model of Subthreshold and Gate Leakage for Architects. Technical Report CS-2003-05, Univ. of Virginia, March 2003.