Leakage Power Estimation in SRAMs *

Mahesh Mamidipaka †
maheshmn@cecs.uci.edu

Kamal Khouri ‡
kamal.khouri@motorola.com

Nikil Dutt †
dutt@cecs.uci.edu

Magdy Abadir ‡
m.abadir@motorola.com

†Center for Embedded Computer Systems
Department of Information and Computer Science
University of California, Irvine, CA 92697, USA

‡High Performance PowerPC Platforms
Semiconductor Products Sector
Motorola Inc., Austin, TX 78729 USA

CECS Technical Report #03-32
Center for Embedded Computer Systems
University of California, Irvine, CA 92697, USA

Sept 1, 2003

Abstract

In this paper we propose analytical models for estimating the leakage power in CMOS based SRAM designs. We identify the transistors that contribute to the leakage power in each SRAM sub-circuit as a function of the operation (read/write/idle) on the SRAM and develop parameterized leakage power models in terms of the high level design parameters and transistor widths. The models take number of rows, number of columns, read column multiplexer size and write column multiplexer size of the SRAM along with the technology parameters as input to estimate the leakage power. The developed models are validated by comparing their estimates against the power measured using SPICE simulations on industrial SRAM designs belonging to the e500[1] *processor core. The comparison shows that the models are highly accurate with an error margin of less than 23.9%.*

[*] This work was done in collaboration with Motorola corporation

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

Contents

List of Figures

List of Tables

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. Static power is the power dissipated in a design in the absence of any switching activity and is defined as the product of supply voltage and leakage current. 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) [6] predicts that leakage power would contribute to 50% of the total power in the next generation processors. Therefore, it is important for system designers to get an early estimate of leakage power to meet the challenging power constraints.

SRAMs are widely used in high-performance processors in the form of caches (tag and data arrays), branch target buffers, reservation stations, etc. and occupy significant portion of the die area. In high-performance micro-processors, L1 and L2 caches alone occupy majority of the die area. Expectedly, SRAMs also contribute to majority of the leakage power in processors. However, system designers currently do not have the ability to perform early estimation of such leakage power. Although lot of research has been done on leakage power estimation, the focus has primarily been on estimation at gate level for combinational logic. These methodologies cannot be applied to SRAMs because of its inherent transistor level design that cannot be represented at gate level. In this paper, we propose analytical models for leakage power estimation in SRAMs as a function of the SRAM operation. The models are parameterized in terms of the structure of the SRAM (number of rows, columns, read multiplexer size, and write multiplexer size). Such models would greatly benefit to system designers in:

- quantifying the static power early in the design cycle
- performing power-performance trade-off analysis of different SRAM configurations
- evaluating the dependencies of various micro-architecture level parameters on the static power dissipation

The paper is organized as follows. Section 2 gives a brief description of the factors that influence leakage power in CMOS technology. Section 3 presents the details about the sub-blocks involved in the implementation of conventional SRAMs. Section 4 presents our analytical models for leakage power estimation in SRAMs. We illustrate the methodology used for transistor width determination in Section 5. Section 6 shows the accuracy of the proposed models by comparing their estimates against SPICE level simulation based estimates on industrial designs. Section 7 presents related work and Section 8 concludes this paper.

2 Leakage Power

Power dissipation in CMOS circuits can be categorized into two main components - dynamic and static power dissipation. Dynamic dissipation occurs due to switching transient current (referred to as short-circuit current) and charging and discharging of load capacitances (referred to as capacitive switching current). Static dissipation is due to leakage currents drawn continuosly from the power supply. There are various modes which contribute to leakage current, such as subthreshold leakage, reverse-biased PN junctions, drain-induced barrier lowering (DIBL), gate-induced drain leakage, punchthrough currents, gate oxide tunneling, and hot carrier effects[9]. However, the main contributor of leakage is the sub-threshold leakage current and is briefly discussed in this section.

$$I_{Dsub} = I_{s0} \cdot [1 - e^{\frac{-V_{ds}}{V_t}}] \cdot [e^{\frac{V_{gs}-V_T-V_{off}}{nV_t}}] \tag{1}$$

Subthreshold leakage is the current that flows from drain to source even when the transistor is off (gate voltage less than threshold voltage). Equation (1) shows the subthreshold drain current I_{Dsub} in BSIM3v3.2 MOSFET model. V_{off} is a emperically determined model parameter, $V_t = KT/q$ where K, q are physical constants and T is the absolute temperature, n is derived from a host of other model and device parameters, V_T is the threshold voltage, V_{gs} is the gate to source voltage, V_{ds} is the drain to source voltage, I_{s0} is the current dependent on the transistor geometry and may be written as $I'_{s0} \cdot \frac{W}{L}$. W and L being the channel width and length of the MOS device respectively. It can be noted from Equation (1) that subthreshold leakage increases exponentially with decreasing threshold voltage (V_T) and the continuous reduction of V_T with technology scaling is making the static power increasingly significant. As shown by Butts and Sohi [1], for a single device in off state, $V_{ds} = V_{cc}$ and $V_{gs} = 0$ and using the approximation $V_{ds} = V_{cc} >> V_t$ this equation can be reduced to:

$$I_{Dsub} = \frac{W}{L} \cdot {I_{s0}}' \cdot e^{\frac{-V_{off}}{nV_t}} \cdot e^{\frac{-V_T}{nV_t}} \tag{2}$$

$$= \frac{W}{L} \cdot K_{tech} \cdot 10^{\frac{-V_T}{S_t}} \tag{3}$$

$$= W \cdot I_{lkg}(T, V_T) = W \cdot I_l \tag{4}$$

where $K_{tech} = {I_{s0}}' \cdot e^{-V_{off}/nV_t}$ and $S_t = 2.303 \cdot n \cdot V_t$ referred to subthreshold slope. For all devices in a given design module, say SRAM, all the parameters in Equation (3) can typically be considered constant, for a given temperature and threshold voltage except for the width and length of the device. Since nearly every device is drawn with minimum L, W is the dimension which has to be accounted in a design for accurate estimation of leakage current as reflected by Equation (4). $I_{lkg}(T, V_T)$ is a constant that can be calculated for a given technology and given temperature and threshold voltage. The leakage characteristics of NMOS and PMOS transistors can be different from each other in a given technology. So to analytically estimate subthreshold current in a design, the leakage currents of the NMOS and PMOS transistors should be considered separately. Also, the above derivation is for an isolated transistor with an assumption that $V_{ds} = V_{cc}$. When there are stacks of transistors (transistors connected in series drain to source) in a design, V_{ds} could be less than V_{cc} thereby reducing the leakage current (from Equation (1)). It was observed in [4] that stacking four transistors reduces the leakage in a transistor by a factor of 20.

These observations form the basis for the leakage power models we develop for SRAMs. In the next section we illustrate the structure of SRAMs, their sub-circuit descriptions and briefly explain their behavior during SRAM operations.

3 SRAMs

SRAMs 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 examples of microprocessor modules in which SRAMs are used. Figure 1 shows a typical structure of a SRAM. It is primarily composed of the following sub-blocks: address decode logic, memory core, read column logic, write column logic, read control, and write control logic. While the generic structure of SRAMs is usually the same, SRAMs typically differ from each other in their size, organization of the memory core (in terms of number of rows and columns).

SRAMs usually support read and write operations [10]. For these operations, the row decoder selects the appropriate wordline corresponding to the input address thereby activating a row in the memory array. For a read operation, the precharged bitlines either retain charge or discharge depending on the data stored in the memory core cells selected by the wordline. The sense-amplifier in the read logic detects the changes in the voltage on the

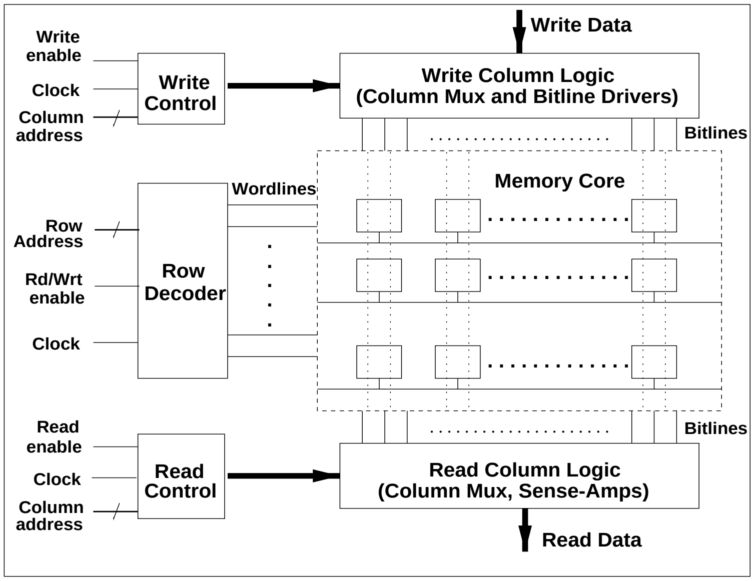


Figure 1. Typical Architecture of Array Structures

bitlines and the appropriate data is multiplexed to the data output. The read control logic controls the signals to the sense-amplifiers and bitline precharge logic. For a write operation, the sense-amplifiers are isolated and the write buffers in write logic drive the bitlines in accordance with the data to be written into the memory location corresponding to the write address. 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 a SRAM 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.

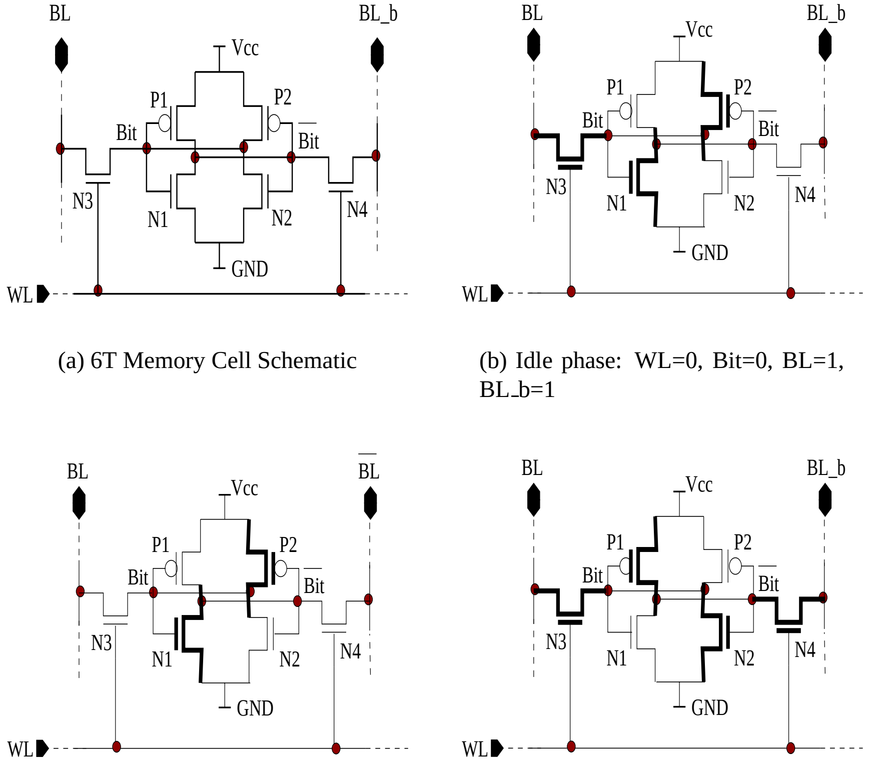
The leakage current in SRAMs vary within a clock cycle depending on the phase of the operation being performed, since different transistors would be in off state during different operations. In the following section we propose analytical models for leakage current in SRAMs during each phase: read, write,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

4 Analytical Models for SRAM Leakage Current

The objective of this work is to develop models parameterized in terms of high level design parameters. As indicated in Section 3, SRAMs 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 in each of its operational phase (read, write, precharge, and idle). To simplify the analysis, we assume that the leakage current in a sub-block during a transient state is same as the leakage current when it reaches a steady state. Although this approximation might introduce some error, we show in Section 6 that the error margin is reasonable.

4.1 Memory Core

Memory core is composed of memory cells that are arranged in rows and columns. Figure 2(a) shows the typical 6-transistor memory cell design. To maintain symmetry, in most memory cell designs, transistors P1, P2 typi-



(a) 6T Memory Cell Schematic

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

(c) Read phase: WL=1, Bit=0

(d) Write phase: WL=0, Bit=1, BL=0, BL_b=1

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

cally 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 idle phase, the wordlines are deselected ($WL = 0$) and the bitlines are precharged ($BL = 1, BL\_b = 1$). So depending on the memory cell data, either transistors N3, P1, N2 (for Bit = 1) or N4, P2, N1 (for Bit = 0) will be in the off-state. Figure 2(b) 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 (5). Equation (6) can be obtained by substituting Equation (4) in Equation (5) where W_{N4}, W_{P2}, W_{N1} are widths of N4, 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 temparature. 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 using Equation (7).

$$I_{memCellIdle} = I_{Dsub}(N1) + I_{Dsub}(N4) + I_{Dsub}(P2) \qquad (5)$$

$$= (W_{N1} + W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP} \qquad (6)$$

$$I_{memCoreIdle} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N4} \cdot I_{lN} + W_{P2} \cdot I_{lP}] \qquad (7)$$

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 (8). The transistors leaking during read phase with $WL = 1$ and $Bit = 0$ are shown in Figure 2(c). 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 (9).

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

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

During the write phase, one of the wordlines is active as per the address. Also in steady state, depending on the write data, one of the bitlines in all bitline pairs is discharged completely to logic '0' ($BL = \tilde{\ } BL\_b$). The transistors that will be in the off-state will be different depending on the wordline selection (WL), data in the memory cell ($bit, \overline{bit}$) and write data (BL, BL_b). The transistors leaking in the memory cell for the case, $WL = 0, Bit = 1, BL = 0$, and $BL\_b = 1$ is shown in Figure 2(d). Taking into account the symmetry of the memory cell, the leakages for different scenarios is shown in Equation (11). Since the data in the all the memory cells cannot be determined apriori, we assume that the probability of 0.5 for $BL == Bit$ and 0.5 for $BL \neq Bit$. The leakage current in write phase for the memory core can then be derived as shown in Equation (13).

$$\begin{aligned} I_{memCellWrt} &= (W_{N1} + W_{N4} + W_{N3}) \cdot I_{lN} + W_{P2} \cdot I_{lP} \\ &= (W_{N1} + 2 \cdot W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP} \quad \textbf{for} \ (WL = 0 \text{ and } Bit \neq BL) & (10) \\ &= W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP} \quad \textbf{for} \ (WL = 1) \text{ or } (WL = 0 \text{ and } Bit == BL) & (11) \\ I_{memCoreWrt} &= N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) + (N_{rows} - 1) \cdot N_{cols} \cdot \\ & [0.5 \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP} + 2 \cdot W_{N4} \cdot I_{lN}) + 0.5 \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP})] & (12) \\ &= N_{rows} \cdot N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) + (N_{rows} - 1) \cdot N_{cols} \cdot W_{N4} \cdot I_{lN} & (13) \end{aligned}$$

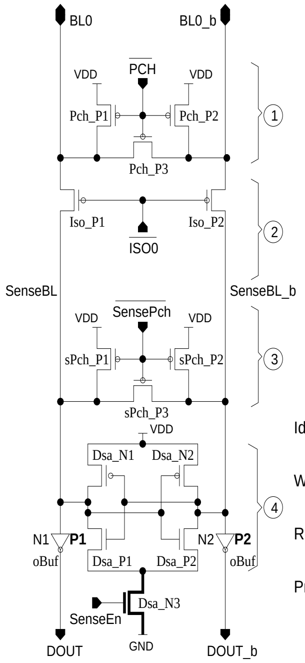
During the precharge phase, the wordlines are usually deselected and the bitline pairs are charged to V_{cc}. The precharge time is significant only when the precharge phase precedes a write phase since in idle and read phases there is either no or partial bitline discharge of bitline. In steady state, as both BL and BL_b are both equal to V_{cc}, for memory core, the leakage current in precharge phase is equal to leakage current in idle phase. Equation (14) and Equation (15) show the leakage currents in memory core for different operational phases. Using the approximation, $N_{rows} \cdot N_{cols} >> N_{cols}$, Equation (15) can be reduced to Equation (16). This means that the leakage current in the memory core can be considered independent of the SRAM operational phase as shown in Equation (17).

$$
\begin{aligned}
I_{memCore} &= N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \textbf{for} \text{ idle or precharge phase} && (14)\\
&= N_{rows} \cdot N_{cols} \cdot (W_{N1} \cdot I_{lN} + W_{P2} \cdot I_{lP}) + (N_{rows} - 1) \cdot N_{cols} \cdot W_{N4} \cdot I_{lN} && (15)\\
&= N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \textbf{for} \text{ read or write phase} && (16)
\end{aligned}
$$

$$
I_{memCore} = N_{rows} \cdot N_{cols} \cdot [(W_{N1} + W_{N4}) \cdot I_{lN} + W_{P2} \cdot I_{lP}] \quad \textbf{for} \text{ read or write or idle or precharge phase} \quad (17)
$$

4.2 Read Column Circuit

Read column circuit is composed of bitline precharge logic, isolation logic, differential sense amplifier, and 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.



1 Bitline precharge logic (Pch)

2 Column isolation logic (Iso)

3 Sense amplifier precharge logic (sPch)

4 Differential sense amplifier logic (Dsa)

Idle phase:	BL0 = 1, BL0_b = 1, SenseBL = 1, SenseBL_b = 1
	PCH = 0, ISO0 = 0, ISO1 = 0, SensePch = 0, SenseEn = 0

Write phase:	BL0 = 0(1), BL0_b = 1(0), SenseBL = 1, SenseBL_b = 1
	PCH = 0, ISO0 = 0, ISO1 = 0, SensePch = 0, SenseEn = 0

Read phase:	BL0 = 1, BL0_b = 1, SenseBL = 1(0), SenseBL_b = 0(1)
	PCH = 0, ISO0 = 0, ISO1 = 0, SensePch = 0, SenseEn = 1

Precharge:	BL0 = 1, BL0_b = 1, SenseBL = 1, SenseBL_b = 1
	PCH = 1, ISO0 = 0, ISO1 = 0, SensePch = 1, SenseEn = 0

Figure 3. Schematic of a differential read column circuit

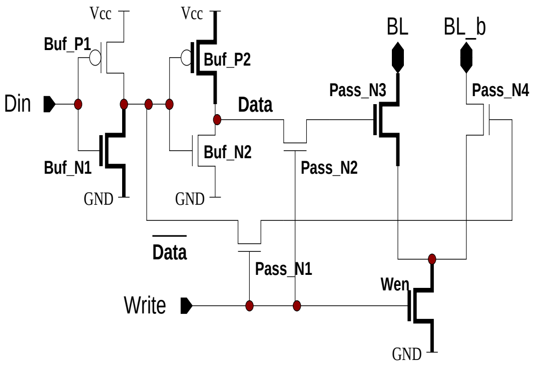
In the idle phase, the bitlines, sense bitlines are precharged and the sense enable, sense precharge, precharge, and isolation signals are deselected (logic LOW). The leakage current in the idle phase is contributed by the sense enable transistor and PMOS transistors in the output buffers as highlighted 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 the in read phase, the isolation transistors are active for a small period of time so that the differential sense amplifier samples the bitline voltages. Also in 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. Analysing the basic schematic under these conditions, and using Equation (4), the leakage current in idle, precharge, read and write phases and for the whole read column sub-block can be derived as shown in equations 18, 19, 20, and 21 repectively.

$$\begin{aligned} I_{rdColIdle} &= I_{rdColPch} = I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1} \\ &= W_{Dsa\_N3} \cdot I_{lN} + 2 \cdot W_{oBuf\_P1} \cdot I_{lP} \qquad (18) \\ I_{rdColWrt} &= 2 \cdot I_{Pch\_P1} + I_{Iso\_P1} + I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1} \\ &= 2 \cdot I_{lP} \cdot (W_{Pch\_P1} + W_{oBuf\_P1} + W_{Iso\_P1}) + I_{lN} \cdot W_{Dsa\_N3} \qquad (19) \\ I_{rdColRead} &= 2 \cdot I_{sPch\_P1} + I_{Iso\_P1} + I_{Dsa\_N1} + I_{Dsa\_P2} + I_{oBuf\_P1} + I_{oBuf\_N2} \\ &= I_{lP} \cdot (2 \cdot W_{sPch\_P1} + W_{Dsa\_P2} + W_{oBuf\_P1} + I_{Iso\_P1}) + I_{lN} \cdot (I_{Dsa\_N1} + I_{oBuf\_N2}) (20) \end{aligned}$$

$$\begin{aligned} I_{rdCol} &= \frac{N_{cols}}{S_{rdMux}} \cdot I_{rdColIdle} && \textbf{for} \text{ idle or precharge phase} \\ &= 2 \cdot N_{cols} \cdot I_{Pch\_P1} + \frac{N_{cols}}{S_{wrtMux}} I_{Iso\_P1} + \frac{N_{cols}}{S_{rdMux}} (I_{Dsa\_N3} + 2 \cdot I_{oBuf\_P1}) \\ & && \textbf{for} \text{ write phase} \\ &= \frac{N_{rows}}{S_{rdMux}} \cdot I_{rdColRead} && \textbf{for} \text{ read phase} \qquad (21) \end{aligned}$$

4.3 Write Column Circuit

The write circuit is a simple differential stage that is driven to saturation by $Data$ and $\overline{Data}$. Two pass transistors and the current source for the differential amplifier is controlled by the Write signal. Figure 4 shows the schematic of the write circuit with the transistors leaking during idle phase in bold.



Idle or Read or Precharge Phase: BL = 1, BL_b = 1, Write = 0, Din = 0

Write Phase: BL = 1(0), BL_b = 0(1), Write = 1, Din = 0

Figure 4. Schematic of a typical write column circuit

In the idle, precharge, and read phase, the bitlines are precharged and the write enable signal is disabled. The transistors leaking in this phase for $Din = 0$ are shown in bold in Figure 4. The leakage current in the sub-circuit for these phases is shown in Equation (22). Note that transistors $pass\_N3$ and Wen are in series and hence because of the stacking effect the leakage current would be considerably less than the leakage of single device in stack. This into account in the leakage power model by the stacking factor (S_2). The stacking factor can be computed by methods described in [3]. Assuming that there is a 0.5 probability of Din being 1 or 0, and since

$pass\_N3$ and $pass\_N4$ would share the same characteristics, Equation (22) can be reduced to Equation (23). Similarly, leakage current for write column circuit in write phase can be derived as shown in Equation (24). The leakage currents for various operational phases for the whole write column logic can thus be calculated as shown in Equation (24).

$$\begin{aligned}
I_{wrtColIdle} &= I_{wrtColRead} = I_{wrtColPch} \\
&= I_{Buf\_N1} + I_{Buf\_P2} + S_2 \cdot I_{pass\_N3+Wen} \quad \textbf{for}\ \text{Din = 0} \\
&= I_{Buf\_P1} + I_{Buf\_N2} + S_2 \cdot I_{pass\_N4+Wen} \quad \textbf{for}\ \text{Din = 1} \qquad (22) \\
\\
&= 0.5 \cdot (I_{Buf\_N1} + I_{Buf\_P2} + I_{Buf\_P1} + I_{Buf\_N2}) + S_2 \cdot I_{pass\_N4+Wen} \\
&= 0.5 \cdot I_{lN} \cdot (W_{Buf\_N1} + W_{Buf\_N2} + S_2 \cdot W_{pass\_N4+Wen}) + 0.5 \cdot I_{lP} \cdot (W_{Buf\_P1} + W_{Buf\_P2})
\end{aligned}$$

$$\begin{aligned}
I_{wrtColWrite} &= I_{Buf\_N1} + I_{Buf\_P2} \quad \textbf{for}\ \text{Din = 0} \\
&= I_{Buf\_P1} + I_{Buf\_N2} \quad \textbf{for}\ \text{Din = 1} \qquad (23) \\
\\
&= 0.5 \cdot I_{lN} \cdot (W_{Buf\_N1} + W_{Buf\_N2}) + 0.5 \cdot I_{lP} \cdot (W_{Buf\_P1} + W_{Buf\_P2})
\end{aligned}$$

$$\begin{aligned}
I_{wrtCol} &= \frac{N_{cols}}{S_{wrtMux}} \cdot I_{wrtColWrite} \quad \textbf{for}\ \text{write phase} \\
&= \frac{0.5 \cdot N_{cols}}{S_{wrtMux}} \cdot (I_{Buf\_N1} + I_{Buf\_P2} + I_{Buf\_P1} + I_{Buf\_N2}) + N_{cols} \cdot S_2 \cdot I_{pass\_N4+Wen} \\
&\qquad \textbf{for}\ \text{idle or precharge or read phase} \qquad (24)
\end{aligned}$$

4.4 Address Decoder, 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 stucture and the critical contributors of leakage power to develop their analytical models.

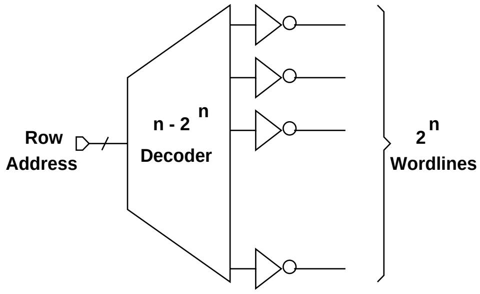


Figure 5. Organization of Address Decoder Sub-circuit

The address decoder, read and write control blocks drive the signals that go across the memory core, read column and write column circuitry respectively. For example, the address decoder drives the wordlines which traverses through all the memory cells in each row of the memory core. Similarly, 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 6 different SRAM designs showed that leakage of the whole block is 1.3-1.6 times the leakage of the circuit output drivers. For example, the leakage power of the address decoder was observed to be 1.4-1.6 times the leakage of the wordline drivers alone in various SRAM designs. This was valid for all phases of SRAM operation (read, write, precharge, and idle). Figure 5 shows the organization of a typical address decoder. 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 (25), where 1.45 is the empirical value calculated as the average of all the measurements using SPICE simulations.

$$I_{cntlLkg} = 1.45 \cdot \sum_i I_{oBuf_i} \tag{25}$$

In the case of address decoders, since the output buffers are wordline drivers, the leakage for the address decoder can be derived as shown in Equation (26) where I_{wlDrv} is the leakage of single wordline driver. Note that the number of wordline drivers in the circuit are equal to number of rows in the memory core. During a read or write operation, since only one of the wordline drivers will be active, the leakage current in the decoder circuit for various phases can be derived as shown in Equation (27) and Equation (28).

$$\begin{aligned}
I_{dec} &= 1.45 \cdot \sum_i I_{wlDrv} = 1.45 \cdot N_{rows} \cdot I_{wlDrv} && (26)\\
&= 1.45[W_{wlDrv\_N} \cdot I_{lN} + (N_{rows} - 1) \cdot W_{wlDrv\_P} \cdot I_{lP}] \quad \textbf{for} \text{ read and write phases} && (27)\\
&= 1.45 \cdot N_{rows} \cdot W_{wlDrv\_P} \cdot I_{lP} \quad \textbf{for} \text{ precharge and idle phases} && (28)
\end{aligned}$$

The output drivers for read control include 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 multiplexer (S_{rdMux}). Equation (29) shows the leakage in read control logic. During read operation, one of the isolation signals is active during small period in read phase so as to enable the sense amplifier to sample the bitline voltage drop. We assume that the isolation driver is in the active state for half the read phase and in the inactive state for the remaining half. The leakage currents in the read control logic block for various phases can then be derived as shown in Equation (30) and Equation (31). Similarly, the leakage in write control logic which comprimises of write multiplexer drivers and some associated logic can be derived as shown in Equations 32-34, where, S_{wrtMux} is the size of the write multiplexer.

$$\begin{aligned}
I_{rdCntl} &= 1.45 \cdot (I_{sEnDrv} + I_{pchDrv} + I_{sPchDrv} + S_{rdMux} \cdot I_{isoDrv}) && (29)\\
&= 1.45 \cdot [I_{lN} \cdot (W_{sEnDrv\_N} + W_{pchDrv\_N} + W_{sPchDrv\_N} + 0.5 \cdot W_{isoDrv\_N}) &&\\
&\qquad\qquad + (S_{rdMux} - 0.5) \cdot W_{isoDrv\_P} \cdot I_{lP}] \quad \textbf{for} \text{ read phase} && (30)\\
&= 1.45 \cdot I_{lP} \cdot (W_{sEnDrv\_P} + W_{pchDrv\_P} + W_{sPchDrv\_P} + S_{rdMux} \cdot W_{isoDrv\_P}) &&\\
&\qquad\qquad \textbf{for} \text{ write, precharge or idle phases} && (31)
\end{aligned}$$

$$\begin{aligned}
I_{wrtCntl} &= 1.45 \cdot S_{wrtMux} \cdot I_{wrtDrv} && (32)\\
&= 1.45 \cdot [I_{lN} \cdot W_{wrtDrv\_N} + (S_{wrtMux} - 1) \cdot I_{lP} \cdot W_{wrtDrv\_P}] \quad \textbf{for} \text{ write phase} && (33)\\
&= 1.45 \cdot I_{lP} \cdot W_{wrtDrv\_P} \cdot S_{wrtMux} \quad \textbf{for} \text{ read, precharge or idle phases} && (34)
\end{aligned}$$

Using the sub-block analytical models, the total SRAM leakage power in each phase can be computed as the sum of the leakage power of sub-blocks as shown in Equation (35).

$$I_{sram} = I_{memCore} + I_{rdCol} + I_{wrtCol} + I_{dec} + I_{rdCntl} + I_{wrtCntl} \tag{35}$$

5 Device Width Calculation

As can be noted from the previous section, the analytical models for leakage power in SRAMs 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 in SRAMs in [5] and for delay estimation of caches in CACTI [11]. Similar to these works, the methodology makes the following assumptions for determining the device widths:

- The effective size of PMOS transistor in a logic gate is assumed to be twice the effective size of NMOS transistors.
- We assume that the size of devices in a memory cell and the dimensions of the memory cell are known apriori. It is very often the case that the memory cells are design much earlier than the design of the SRAM.
- The technology dependent parameters and frequency of operation of SRAM are assumed to be provided by the user.

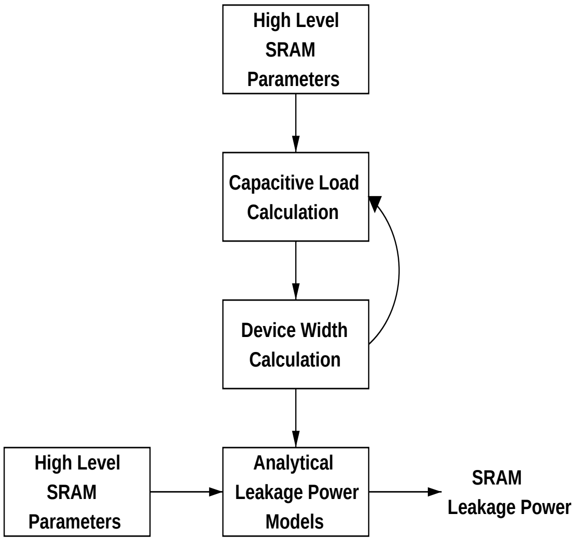

High Level SRAM Parameters
Capacitive Load Calculation
Device Width Calculation
High Level SRAM Parameters
Analytical Leakage Power Models
SRAM Leakage Power

Figure 6. Methodology for Leakage Power Estimation in SRAMs

Figure 6 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 aims to start 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 might require the 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 (36), 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 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.

$$C_{BL} = N_{rows}.(C_{memCell} + C_{metal}.H_{memCell}) + 3.C_{drain} \tag{36}$$

$$W_{pmos} = f(C_{BL}, T_{precharge}) \tag{37}$$

Once all the required transistor widths are derived, these are used in the leakage power analytical models illustrated in Section 4 for obtaining leakage power estimates in SRAMs.

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. Also the memory cell devices used in SRAMs were different from the devices in rest of SRAM sub-blocks to reduce the leakage power. The memory cell devices are primarily high-threshold voltage devices customized to reduce the overall SRAM leakage power. So different I_{lN} and I_{lP} were calculated for leakage power estimations in memory core and other sub-blocks. 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.

	Array Size	Error		
	(# of cells)	IDLE	READ	WRITE
SRAM1	352	19.50%	-8.22%	-5.17%
SRAM2	704	16.97%	10.70%	-0.11%
SRAM3	1024	14.23%	4.23%	-16.61%
SRAM4	1536	-3.21%	-10.27%	3.62%
SRAM5	5120	-19.31%	-15.35%	-23.95%
SRAM6	5888	-19.61%	-8.59%	-17.95%
SRAM7	9504	-0.23%	19.78%	-3.08%

Table 1. Comparison of the Leakage Power Models with SPICE

Table 1 shows the comparison across different SRAMs used in an industrial e500 processor core design. The actual leakage power numbers and the names of the array 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 SRAM in terms of the number of bit cells, Columns 3, 4, 5, and 6 indicate the percentage error in the model estimates for read, write, precharge and idle operational phases 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, SRAMs 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. SRAMs 3-7 mostly correspond to the typical implementation styles illustrated in the Section 4. From Table 1 the error margin varies from -23.9% to +19.5%. The reasons for variation were due to:

- mismatch in the calculated device widths and the actual device widths
- various approximations used for simplyfying the analytical models.
- various custom design optimizations for speed which are not accounted for in the model. For example, gate skewing [8] 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 power in some SRAM designs and an under-estimate in some arrays depending on its implementation. Hence a variation between -23.9% to +19.5% in error is seen between the model estimates and the actual power based on SPICE simulations.

7 Related Work

Static power estimation has been an area of research interest for quite a long time. The focus however, was primarily on estimation at gate level [7, 4]. Recently, more attention is being paid to leakage power estimation at higher level of design hierarchy. Butts and Sohi[1] propose a generic model for microarchitectural 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. A methodology for estimation of leakage power for micro-architectural components in interconnection networks is proposed by Chen et al.[2]. The methodology is based on simulation of fundamental circuit components for various input states. Zhang et al.[12] develop an architectural model for subthreshold and gate leakage that explicitly captures in temperature, voltage, gate leakage, and parameter variations. To the best of our knowledge, this is the first attempt to estimate leakage power in SRAMs based on analytical models parameterized in terms of high level design parameters.

8 Conclusions and Future Work

In this paper, we presented analytical models for leakage power estimation of SRAMs early in the design cycle. The models are based the high level SRAM parameters such as number of rows, number of columns, read column multiplexer size and write column multiplexer size of the SRAM 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 23.9%. Since the models give the leakage power contributions of each sub-block, they can be used to identify the sub-blocks with most leakage power for use of optimization techniques. We plan to extend these models so as to estimate leakage power in caches for a given configuration.

References

[1] J. A. Butts and G. S. Sohi. A static power model for architects. In *International Symposium on Microarchitecture*, pages 191–201, 2000.

[2] X. Chen and L. Peh. Leakage power modeling and optimization in interconnection networks. In *International Symposium on Low Power Electronics and Design*, 2003.

[3] W. Jiang, V. Tiwari, E. Iglesia, and A. Sinha. Topological analysis for leakage prediction of digital circuits. In *VLSI Design 2002*, pages 39–44, 2002.

[4] M. Johnson, D. Somasekhar, and K. Roy. Models and algorithms for bounds on leakage in cmos circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 714–725, 1999.

[5] M. Mamidipaka, K. Khouri, N. Dutt, and M. Abadir. Idap: A tool for high level power estimation of custom array structures. In *International Conference on Computer Aided Design*, 2003 (to appear).

[6] SIA. International technology roadmap for semiconductors. Technical report, http://public.itrs.net/.

[7] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw. Duet: an accurate leakage estimation and optimization tool for dual-vt circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 79–90, 2002.

[8] T. Thorp, G. Yee, and C. Sechen. Design and synthesis of monotonic circuits. In *International Conference on Computer Design*, 1999.

[9] Y. P. Tsividis. *Operation and Modeling of the MOS Transistor*. McGraw-Hill Book Company, 1988.

[10] N. Weste and K. Eshragian. *Principles of CMOS VLSI Design, A Systems Perspective*. Addison-Wesley Publishing Company, Reading, CA, 1998.

[11] S. Wilton and N. Jouppi. An enhanced access and cycle time model for on-chip caches. Technical report, WRL Research Report 93/5, June, 1994.

[12] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects. Technical Report CS-2003-05, Univ. of Virginia, March 2003.