

CARNEGIE MELLON

Department of Electrical and Computer Engineering

Exploring Low Power Memory Design

Michael Berty

1998

The Carnegie Mellon logo is located in the bottom right corner of the page. It consists of the words "Carnegie" and "Mellon" stacked vertically, enclosed within a tilted rectangular border.

Carnegie
Mellon

Exploring Low Power Memory Design

Michael Berty

**A thesis submitted to the graduate
school in partial fulfillment of the
requirements of the degree of**

**Master of Science
in
Electrical and Computer Engineering**

**Carnegie Mellon University
Pittsburgh, Pennsylvania 15213**

May 1, 1998

Copyright 1998 by Michael Berty

All rights reserved.

Abstract

In this thesis, a methodology for designing low power embedded SRAMs is explored. Starting from Cascade's automatically generated SRAM memories, designed with digital signal processing applications in mind, the overall power dissipation is reduced considerably without significant increases in access times using both circuit-level and architectural transformations. Special consideration is given to the feasibility of automating this entire process. Segmented memories are also explored. In this case the power dissipation can be reduced by over 2.6X when compared to Cascade's automatically generated memories. However, the drawback of this technique is that there is a considerable access time and area increase.

Table of Contents

Abstract iii

Table of Contents v

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Automating Design For Low Power 2

1.3 Thesis Overview 3

Chapter 2 Previous Research 5

2.1 Divided Word and Data Line Structure 5

2.2 Divided Power Line 7

2.3 Pulsed Operation of Word-Line Circuitry 8

2.4 Current Mode Operation 10

2.5 Voltage Down Converter 11

Table of Contents

Chapter 2 (continued)	
2.6 OuadRail	12
Chapter 3 Design For Low Power	13
3.1 Automatically Generated SRAMs	13
3.2 Where Is All the Power Going	15
3.3 Internal Circuits of the SRAM	15
3.4 Manipulating the Netlist	18
3.5 Synchronizing the SRAM	21
3.6 Manipulating the Layout	22
3.7 Memory Segmentation	24
Chapter 4 Conclusions	28
Bibliography	30

1 Introduction

Recently, the demand for portable communications has led to more and more low power ASIC designs. Thus, power consumption of digital systems is increasingly becoming one of the most important design parameters. Specifically, memory dominant applications such as speech recognition and video image processing are also being developed with portability in mind. Most portable applications incorporate embedded SRAMs that must meet critical speed and power constraints. It has been shown that power consumed during memory accesses accounts for a significant portion of the total power consumption in microprocessors [7],[9], thus minimization of memory power is an important area of concern for today's IC designers.

1.1 Motivation

In CMOS digital circuits there are two main contributors to power dissipation: dynamic and static. Static power dissipation is caused by leakage current drawn continuously from the supply. This is from the reverse bias leakage between diffusion regions and the substrate. Also, subthreshold currents can contribute to static power. In most of today's IC designs,

static power dissipation is orders of magnitude below that of dynamic (or switching) power dissipation.

Dynamic power dissipation is a result of the charging and discharging of load capacitances in addition to switching transient currents. Dynamic power dissipation also includes short circuit currents; However, we can generally ignore this short circuit current compared to the current due to charging of capacitors. Dynamic power dissipation of CMOS circuits is given by $P_{\text{dyn}} = \alpha C_L V_{\text{dd}}^2 f$, where α is the node transition activity factor, C_L is the load capacitance, V_{dd} is the supply voltage, and f is the clock frequency. Thus, the minimization of power dissipation in CMOS digital circuits is achieved by reducing the components of this equation. Since the dynamic power has a quadratic dependence on the supply voltage, V_{dd} , it is easily seen that the lowest supply voltage is desirable. However, lowering the supply voltage induces a cost of increased gate delays. Thus we must consider both power reduction along with increased delay when looking at low power design methodologies. Superior approaches are ones that provide more decrease in power for a given increase in delay.[2]

1.2 Automating Design For Low Power

As device sizes continue shrinking, more and more transistors are fitting on integrated circuits. This increased complexity is driving the industry towards more automated techniques to perform the actual layout of major blocks in their designs. The demand for designers to both get the design completed quickly along with the increased complexity of the design is pushing the industry to turn to tools that automate major bottlenecks in the design process.

Also, trends in portable communications are increasing the demand for power conscious methodologies. Low power design is quickly becoming a major criteria when developing tools to automate design. [3],[8]

1.3 Thesis Overview

The focus of this thesis is reducing the power dissipation of embedded SRAMs without incurring a large penalty in access times. An industry representative tool (Cascade's Epoch) is used to automatically generate SRAMs used as embedded memories in a typical Digital Signal Processor. Initial netlist simulations, generated by Epoch and tested in HSPICE, break down the SRAM designs into separate blocks in order to see the distribution of the total power dissipation and identify the areas to be modified. Following this initial breakdown, the problem areas are discussed and those circuits are improved to reduce the power dissipation without significantly increasing the access times. These new cells are hand designed in the Cadence Design Frameworks utilizing the .35 μ m HP process. The memories are extracted from Cadence and tested using HSPICE. Further exploration into the segmentation of SRAMs is also looked at next. Results indicate that if an increase in both area and speed can be tolerated, these segmented memories will drastically reduce the overall power dissipation.

Introduction

2 Previous Research

This section will provide some background into previous methods used to design low power RAMs. The techniques looked at are from both the architecture level and the circuit level. At the architecture level, the main concern is reducing the charging capacitance of long lines. Two partitioning approaches are discussed. Another approach that pulses the word line circuitry to reduce the active duty cycle is looked at. At the circuit level, the approaches discussed involve reducing the on-chip operating voltage, current mode operation, and limiting the voltage swing of large capacitance line (QuadRail).

2.1 *Divided Word and Data Line Structure*

Increasingly, the charging capacitance due to long word lines is becoming a major concern for power conscious memory designers. A typical example of the techniques used to minimize this word line capacitance is the partial activation of multi-divided word line approach proposed by M. Yoshimoto [12]. Figure 2.1 illustrates the two stage hierarchical row decoder structure. Memory cells are partitioned into memory sub-arrays, and selecting a word line for this sub-array requires two stages of decoding. The main decoder selects the

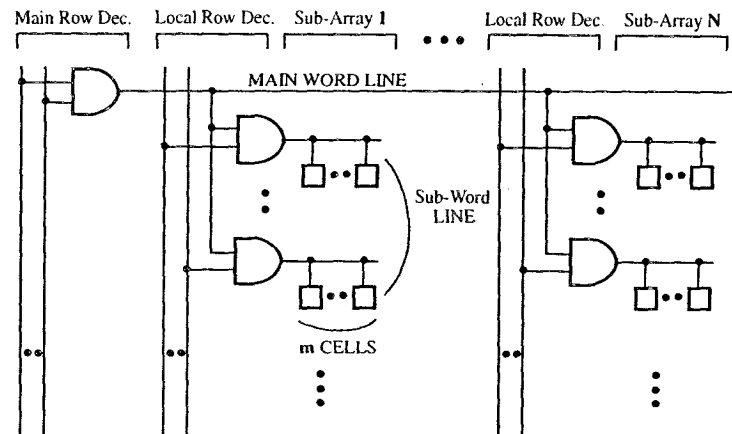


Figure 2.1 Divided Word Line (DWL) Structure [12]

main word line, then the local decoder selects the word line of the selected sub-array. Since only one sub-array is used during any read or write operation, the total charging capacitance in the memory is reduced.

Overpartitioning will result in the total charging capacitance to actually increase. This is due to the fact that the length of the main word line increases as the number of local row decoders increases. When an array is overpartitioned, the capacitance of the main word line dominates.

Similar techniques have been adopted to reduce the total charging capacitance of data lines. Data lines have been segmented into several sections and only one section is active during a read or write operation. This technique utilizes a shared sense amplifier, shared I/O lines, and a decoder to choose which section to activate for any given operation.

2.2 Divided Power Line

Voltage scaling is one of the most common techniques to reduce the power dissipation in VLSI circuits. One major concern is that the dc current caused by subthreshold currents increases exponentially when V_T decreases along with a lowering of V_{DD} . Eventually this subthreshold current will dominate the active current of the entire chip. This subthreshold current is mainly attributed to inactive circuits in the memory array. To minimize this subthreshold current effect, K. Sakata [5] shows, through Figure 2.2, the approach to cut off the leakage path of inactive portions of the memory.

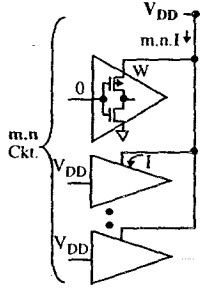
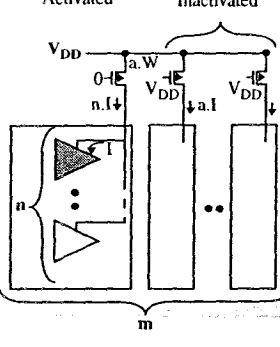
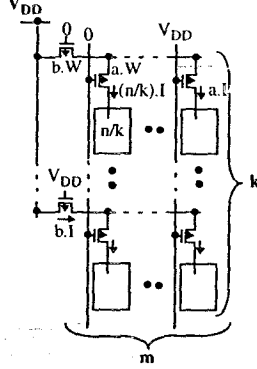
		Conventional	1-D Selection	2-D Selection
Configuration				
Active Subthreshold Current	Ideal	$m.n.I$	$n.I$	$(n/k).I$
	Actual	$m.n.I$	$n.I + (m-1).a.I$	$(n/k).I + (m-1).a.I + (k-1).b.I$

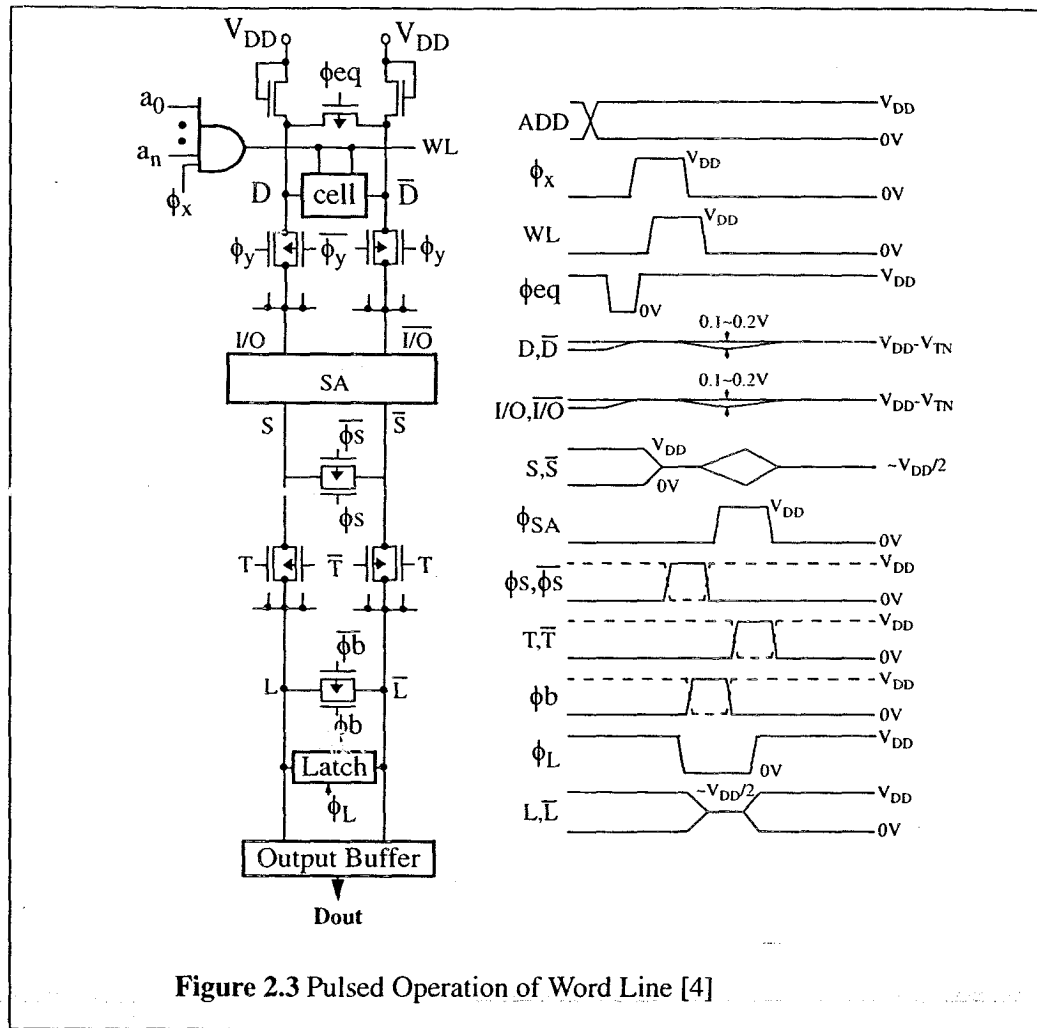
Figure 2.2 Multi-Divided Power Line [5]

This technique is described as the partial activation of multi-divided power lines. Figure 2.2 illustrates the comparison of the active subthreshold currents between the conventional method and 1-D and 2-D partitioning schemes.

In the conventional method, after one word line is selected, the additional word line drivers are all sources of subthreshold current, which can eventually become the dominating current of the entire chip. This active subthreshold current is reduced in the one-dimensional scheme by selectively cutting off the inactive components. All of these inactive components effectively have no subthreshold current. This reduction is further seen in the 2-D scheme where additional sub-arrays can be selectively turned off by PMOS switches. Ideally, the current drawn is only from the active blocks. Note that in this scheme, inactive blocks not only can not be read or written to, they do not store data either. Therefore, a memory controller that keeps track of memory used is also needed.

2.3 Pulsed Operation of Word-Line Circuitry

Pulsing the word-line signal is another technique to reduce the overall power dissipation in SRAM memories. Figure 2.3 shows the overall method that O. Minato [4] has employed to shorten the active duty cycle during read and write operations. The word activation pulse ϕ_x is held high long enough to build up the data-line signal and latch the amplified signal by ϕ_L . Figure 2.3 shows this scheme with a pulsed sense amplifier and a latch circuit. Initially the data line signal (D, D_{bar}) is selected by ϕ_y and $\phi_{y\text{bar}}$ and transmitted to I/O and I/O_{bar} . The sense amplifier amplifies this signal to S and S_{bar} . This signal is selected by T and T_{bar} and transmitted to the latch. The latch holds the values after the sense amplifier and word line are deactivated. The power dissipated is reduced by the duty ratio of the pulse duration to the cycle time.



Exploring Low Power Memory Design
9

2.4 Current Mode Operation

The performance of the sense amplifier degrades considerably as operating voltages scale lower and lower. It becomes harder for the sense amplifier to sense the smaller voltage swings across bit lines. The current mode sense amplifier is one way to overcome this limi-

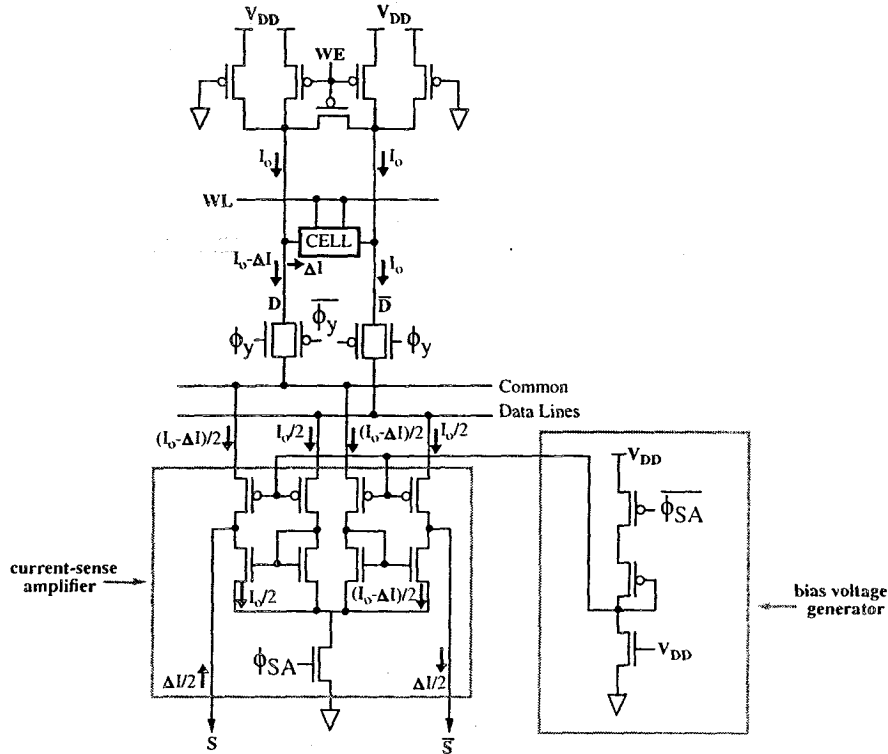


Figure 2.4 Current Sense Amplifier [6]

tation. Figure 2.4 shows K. Sasaki's [6] proposed current sense amplifier circuit. This current mode sense amplifier senses the difference in current between the bit, bit_{bar} lines as shown above. I_0 is initiated on both the bit and bit_{bar} lines, ΔI flows into either the bit or bit_{bar} input of the memory cell leaving $I_0 - \Delta I$ on one data line and I_0 on the other line. The sense amplifier utilizes PMOS data-line loads, and the current-mirror configuration produces the current $\Delta I/2$ which eventually charges and discharges the outputs S and S_{bar}. The

bias voltage generator increases the gain by operating the PMOS devices close to the saturation region. Current sensing is advantageous over conventional voltage sensing amplifiers since the required voltage swing on the bit lines is usually less than 30 mV. This eliminates the need for pulsed data-line equalization, allowing for fast sense times, with minimal power consumption.

2.5 Voltage Down Converter

Scaling the voltage on-chip of the memory devices is desirable to reduce the overall power consumption of VLSI chips. H Tanaka [11] shows, through Figure 2.5, a proposed VDC (Voltage Down Converter) that yields a stable and accurate output voltage, V_{DL} , under rapidly changing load currents. This VDC contains a current-mirror differential amplifier and a common source drive transistor. To minimize the output voltage drop, V_G is designed to respond quickly when the output voltage goes low, thus the VDC of Figure 2.5 is able to maintain a stable and accurate output voltage. The bias current, I_B , is used to clamp the output voltage when the load current approaches zero.

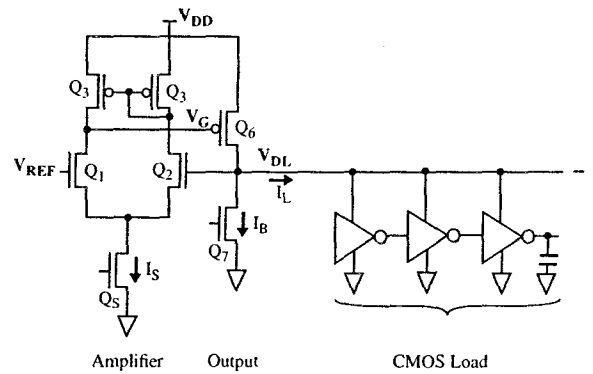


Figure 2.5 Voltage Down Converter [11]

2.6 QuadRail

Exploiting the fact that large capacitance lines driven to the rails dissipates a large amount of dynamic power, H. Sutioso [10] shows that a substantial savings of power over standard CMOS can be accomplished by using QuadRail incurring only minimal delay increases.

The main memory cell is a 3-port (1 write, 2 reads) cell which is high speed and has the abil-

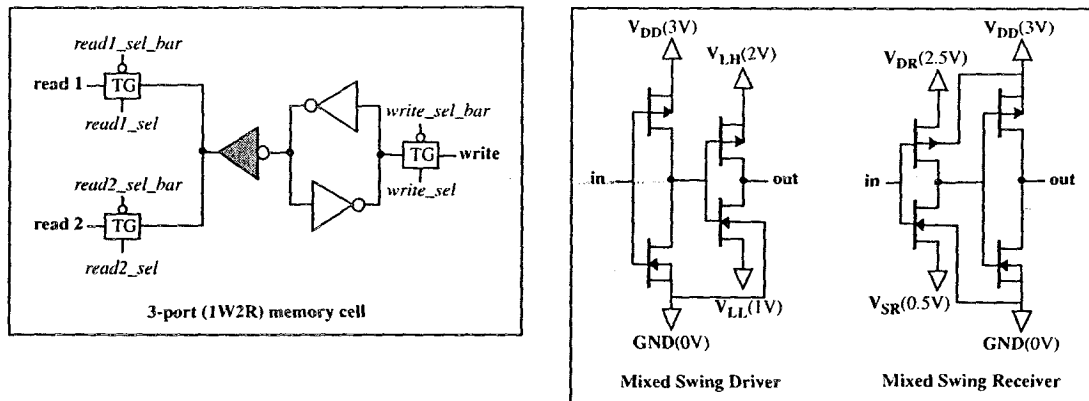


Figure 2.6 QuadRail Memory Cell Driver and Receiver [10]

ity to read and write at the same time. Figure 2.6 illustrates the architecture of this 3-port memory cell. The shaded buffer was implemented as a mixed swing (QuadRail) driver shown above. The basis behind QuadRail is that driving large capacitance lines should be done in low swing mode while logic functions should be high swing for speed considerations. The mixed swing driver and receiver shown above illustrate the high-swing to low-swing and low-swing to high swing conversions. Results indicate that for very long data lines, QuadRail provides a large reduction in power dissipation (about three times less power) while incurring minimal speed delays.

3 Design For Low Power

This section will explore the methodical reduction of power in automatically generated embedded SRAMs.

3.1 *Automatically Generated SRAMs*

For ASIC designers, the speed from the design concept to actual fabrication of the completed design is a very important criteria. More and more designers are turning to EDA (Electronic Design Automation) tools to ease the burden of designing millions of gates for a complete on-chip design. One such tool is Cascade's Epoch [1]. Our methodology is using Cascade's memory module generator, as a representative of current industry tools, to use as a building block for low power memories.

The memories that we used were Cascade's basic asynchronous single-port static RAMs. This memory was chosen because of fast access times, reduced power requirements, and high density as stated in Cascade's user manual. There are 4 separate parameters that Cascade uses to generate memories: the number of words, the number of bits/word, the number of address lines, and the number of bits per column (BPC). The BPC number must be one of

the set {1,2,4,8,16}. Each unique memory size is defined by the number of rows, number of columns, and the bits per column. ($Rows = Words/BPC$, $Columns = Bitwidth*BPC$). For our experiments, we used memories with the following specifications that are utilized by a digital signal processor: 512 words, 24 bits/word, and 4 bits/column. Figure 3.1 illustrates the actual partitioning of this specific memory by Cascade.

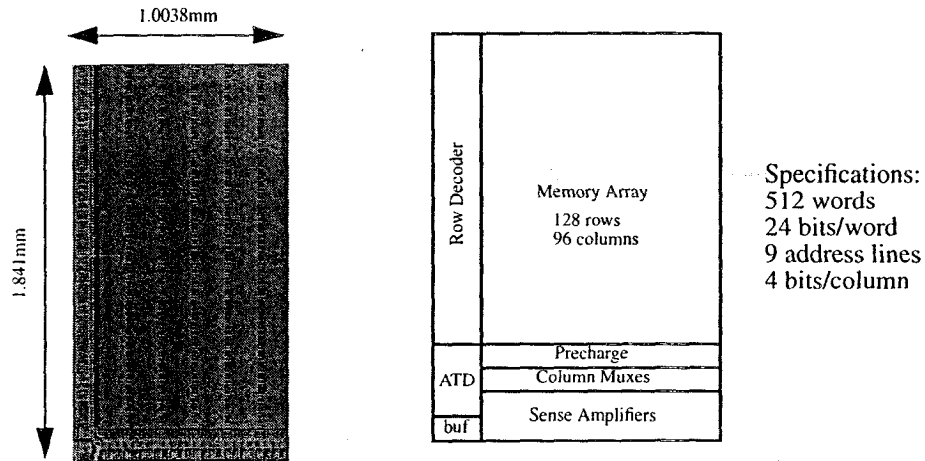


Figure 3.1 Cascade Generated SRAM

The SRAM is partitioned into 6 basic building blocks that complete the memory: the memory array, the precharge circuits, the row decoders, the ATD (address transition detection), the column muxes and the sense amplifiers. The ATD unit is used for asynchronous memories and detects a change in the address lines, thus initiating a memory access. Before we take an in-depth look at the circuits involved in each of these blocks, we will examine the different power dissipation numbers of these basic blocks and determine which blocks require the most attention.

3.2 Where Is All the Power Going

An initial survey of the distribution of the power in this memory was done by developing a hierarchical SPICE netlist, separating the various components as described in figure 3.1. These memories are designed to be used in association with a digital signal processor and are simulated changing the addresses at three separate clock speeds, 50MHz, 67MHz, and 100MHz. Our memory was simulated in its entirety as such; initial conditions were placed in each of the internal nodes of the memory array cells, addresses were specified, read and written to, verification of functional correctness was performed. Simulations were carried out using HSPICE. Figure 3.2 illustrates the fact that 69% of the power dissipated was due to the precharge circuitry. The next largest contributor was the sense amplifier block, with about 16% of the overall power dissipation.

SRAM block	period = 20ns	period = 15ns	period = 10ns
6T cells	0.607mW (1.41%)	0.669mW (1.40%)	0.800mW (1.28%)
Column Decode	0.911mW (2.12%)	1.100mW (2.30%)	1.661mW (2.66%)
Precharge	29.628mW (69.02%)	33.225mW (69.64%)	43.019mW (68.94%)
Sense Amps	7.675mW (17.88%)	7.749mW (16.24%)	9.535mW (15.28%)
Row Decode	3.014mW (7.02%)	3.609mW (7.56%)	5.397mW (8.65%)
Extra Buffers	1.089mW (2.54%)	1.355mW (2.84%)	1.992mW (3.19%)
Total	42.925mW	47.707mW	62.404mW

Figure 3.2 Original Power Distribution in Cascade SRAM

3.3 Internal Circuits of the SRAM

A general block diagram of our automatically generated SRAM is shown on Figure 3.1. This section will break down the SRAM into it's major components and look at the actual circuits involved.

3.3.1 Memory Cell

Figure 3.3 shows the core of the memory array, the six-transistor memory cell. This cell consists of two cross-coupled inverters and two n-channel pass gates. The inputs to this cell is the word select line and the bit and bit_{bar} lines. The outputs are the bit and bit_{bar} lines.

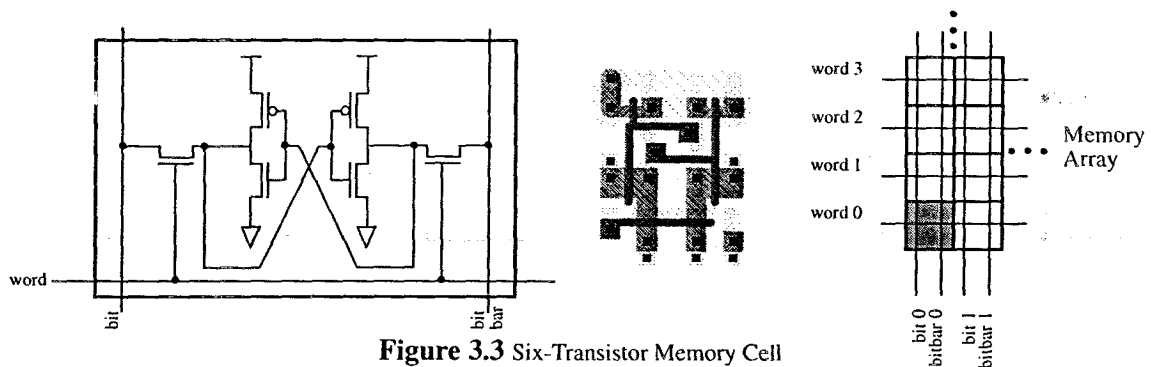


Figure 3.3 Six-Transistor Memory Cell

When a *read* or *write* operation occurs, the word line is set high allowing current to flow either in or out of the cell. The layout of these cells is illustrated above also. They are connected together and oriented as shown in the Figure. This cell has minimal static power dissipation because of its CMOS characteristics.

3.3.2 Precharge Circuit

The main contributor to power dissipation in our SRAMs is the precharge circuit. It consists of about 69% of the overall power loss as presented in the previous section. Figure 3.4 shows us what this circuit looks like and its orientation with the memory array cells situated above the precharge cells. The circuit consists of two n-channel transistors that are always on and always pulling the bit and bit_{bar} lines up. Two more n-channel transistors are used to further pull up the bit and bit_{bar} lines when the pre signal is high. A p-channel transistor is

used to tie the two lines together during a precharge cycle. Since the bit and bit_{bar} lines are relatively long, they have a large internal capacitance associated with them. This cell is used to precharge the bit and bit_{bar} lines high during a *read* operation enabling a fast sense of the difference in voltages of the lines by the sense amplifier.

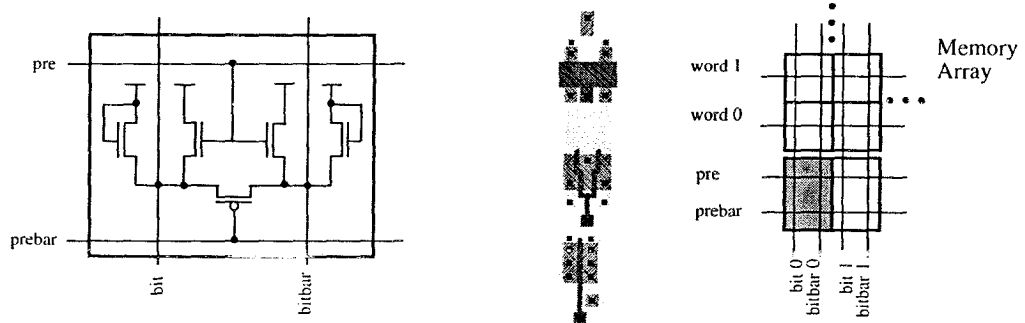


Figure 3.4 Precharge Circuit

3.3.3 Sense Amplifier

The sense amplifier employed by the Cascade generated SRAMs is shown in Figure 3.5. Differential amplifiers are used to amplify the small difference between the bit and bit_{bar} lines from the memory array. The amplifiers feed into a cross coupled latch that holds the values before they are sent to the outputs (Dout). The signal, donebar, tells the sense amplifier when it can begin performing the sensing operation. This signal is sent by a dummy column relaying information on when the information is prepared on the bit and bit_{bar} lines. During a *read* operation the WR signal is set low and the sensing occurs when the donebar signal is set low. The *write* operation occurs when the WR signal is set high, effectively allowing the Din and Din_{bar} signals to propagate through to the latch. The orientation of the sense amplifier is shown also on Figure 3.5. Our SRAMs contain 4 columns to every one

senseamp, thus a column mux is necessary to select the one set of lines to propagate to the sense amplifier. The differential amplifier is a large contributor to the power dissipation of this cell.

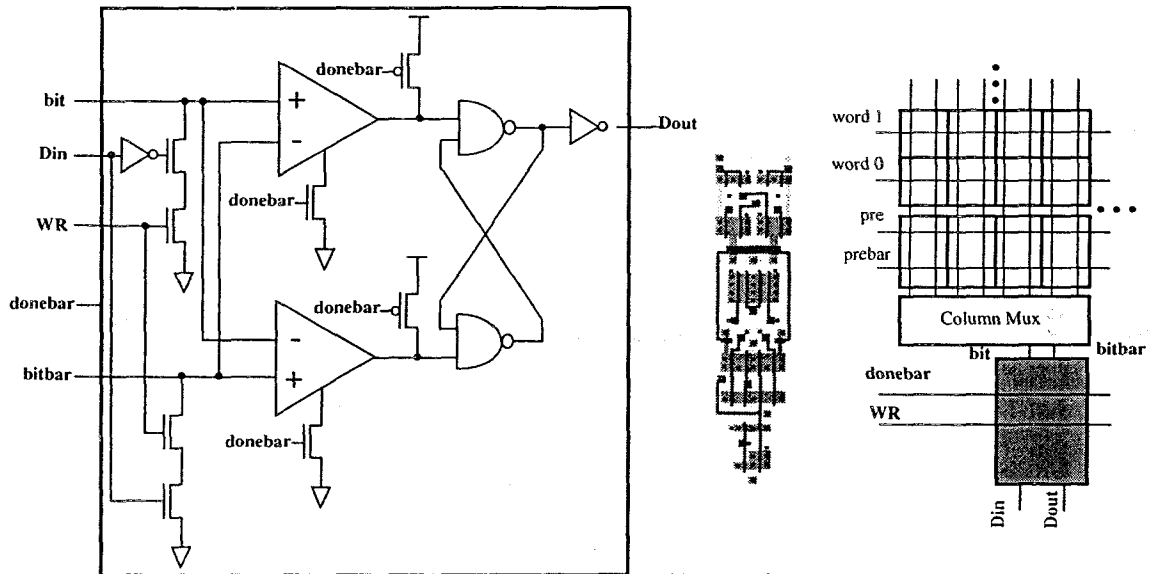


Figure 3.5 Sense Amplifier

3.4 Manipulating the Netlist

After the initial distribution of power dissipation was determined, the next step involved the manipulation of the circuits of the SRAM to reduce the overall power in this SRAM. The main contributors to this power dissipation were the precharge and sense amplifier cells. Both cells were dissipating a large amount of static power.

3.4.1 New Precharge

The new precharge cell is shown in Figure 3.6 and is void of the two n-channel transistors with gates and sources connected to V_{DD} . This eliminated the static current drawn from the

power sources to the relatively long lines bit and bit_{bar}. The drawback of this modification is that the bit and bit_{bar} lines will not be charged to their full potential as fast, increasing access times.

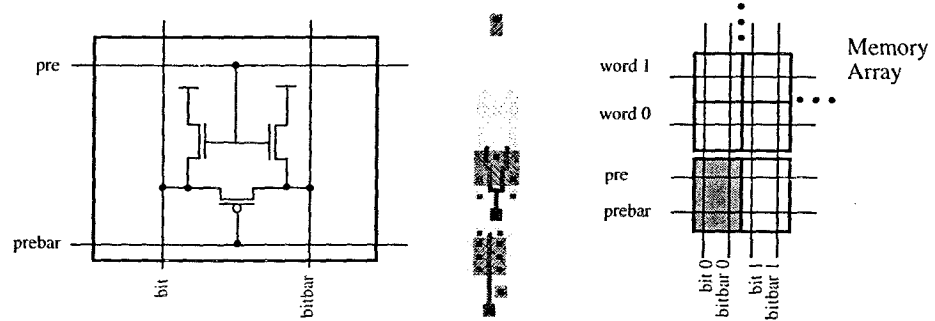


Figure 3.6 Modified Precharge Circuit

3.4.2 New Sense Amplifier

It was seen through simulations that the bit and bit_{bar} lines were driven high enough to pass the threshold requirements of the sense amplifier latch without aid of the differential ampli-

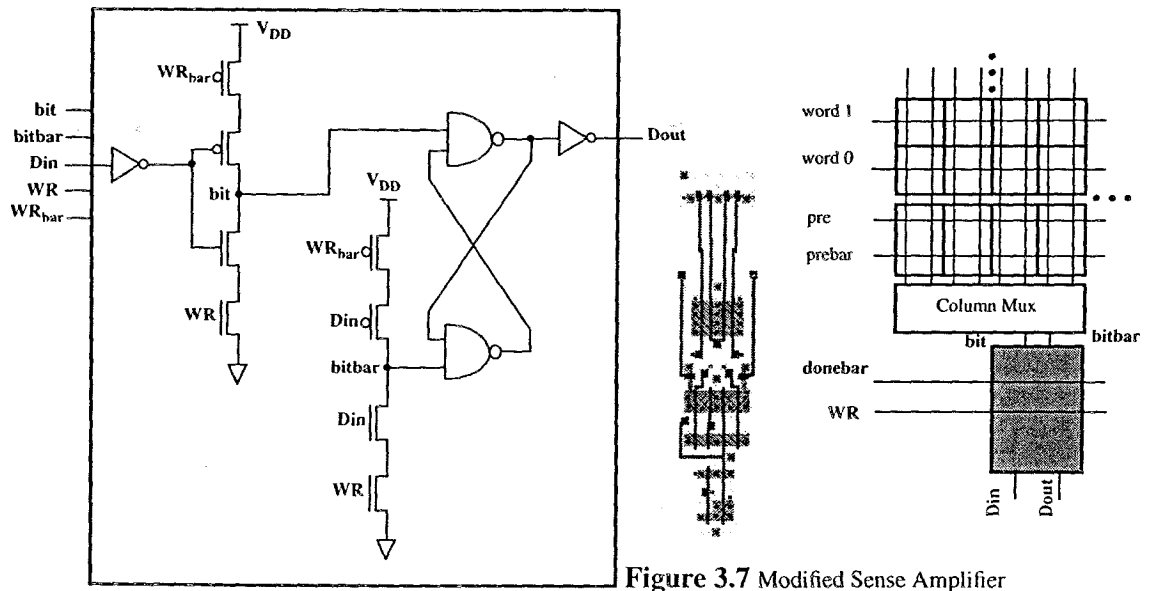


Figure 3.7 Modified Sense Amplifier

fier. Thus the elimination of the differential amplifier and the static power involved with that circuit was the next step. In order to pull up the correct values for D_{in} and $D_{in_{bar}}$ two additional PMOS transistors were required on both the bit and bit_{bar} lines since the original pull up transistors of the precharge cell were taken away. Thus on a *write* operation, the correct values for both D_{in} and $D_{in_{bar}}$ can now be propagated to the latch correctly.

3.4.3 Results of Netlist Modifications

SRAM block	period = 20ns	period = 15ns	period = 10ns
6T cells	0.743mW (2.80%)	0.802mW (2.49%)	0.930mW (2.00%)
Column Decode	0.903mW (3.40%)	1.089mW (3.38%)	1.695mW (3.65%)
Precharge	15.679mW (59.08%)	19.759mW (61.30%)	29.916mW (64.49%)
Sense Amps	5.080mW (19.14%)	5.589mW (17.34%)	6.441mW (13.89%)
Row Decode	3.019mW (7.02%)	3.615mW (11.22%)	5.401mW (11.64%)
Extra Buffers	1.116mW (4.20%)	1.377mW (4.27%)	2.035mW (4.39%)
Total	26.540mW	32.231mW	46.383mW

Figure 3.8 Power distribution of SRAM after modifications of precharge and sense amplifier

Figure 3.8 illustrates the results of the SPICE Netlist modifications on the precharge cell and the sense amplifier cell. Power has been reduced in both the Precharge and Sense Amp distribution of the SRAM along with a substantial savings of overall power. Note that this simulation does not include extracted parasitic capacitances from actual layout cells. The table above shows an improvement of: Precharge power savings = {47%,40%,30%}, Sense Amp power savings = {33%,28%,32%} and Overall power savings = {38%, 32%,26%} for the periods = {20ns,15ns,10ns}. The precharge is still by far the dominating power dissipation factor in this embedded SRAM with an average of over 60% of the total power due to the

precharge. The next section explores further methods of reducing the power due to the precharge circuitry.

3.5 Synchronizing the SRAM

Memories generated by Cascade are asynchronous and rely on the Address Transition Detection Unit to know when to set the precharge in order to perform a correct *read* operation. More power due to the precharge can be saved if the memory is changed to a synchronous SRAM. Precharging would then only occur when you have a *read* operation. Figure 3.9 provides insight into how to clock the precharge signal. The precharge pulse would be clocked if we had access to a Precharge Pulse (PrePulse) that was synchronous. This signal, along with the *read* signal (WR_{bar}), determines the actual signal that activates the precharge circuits.

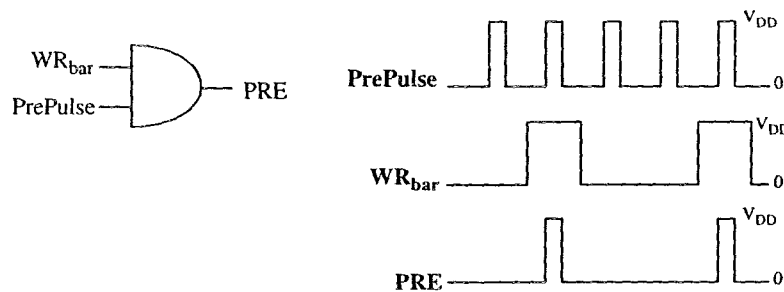


Figure 3.9 Clocking the SRAM

3.5.1 Netlist Simulations

Synchronizing our SRAM involved modifying the ATD cell logic. This involved eliminating all of the combinatorial logic dealing with determining when an address transition

occurs. The basic pulsed Precharge enabling signal as described by figure 3.9 was implemented. This modification was again made to the SPICE netlist and simulated using HSPICE. Results of the synchronous SRAM memory can be seen in Figure 3.10. Notice the amount of power savings in the precharge division of our SRAM. The extra buffers also received some power savings due to the reduction in logic of the ATD which was measured in that category.

SRAM block	period = 20ns	period = 15ns	period = 10ns
6T cells	1.689mW (8.68%)	2.206mW (10.35%)	2.998mW (9.64%)
Column Decode	0.895mW (4.60%)	1.077mW (5.05%)	1.642mW (5.28%)
Precharge	8.582mW (44.13%)	8.662mW (40.64%)	14.344mW (46.11%)
Sense Amps	5.023mW (25.83%)	5.526mW (25.92%)	6.324mW (20.33%)
Row Decode	2.529mW (13.00%)	3.036mW (14.24%)	4.533mW (14.57%)
Extra Buffers	0.731mW (3.76%)	0.810mW (3.80%)	1.269mW (4.08%)
Total	19.448mW	21.316mW	31.110mW

Figure 3.10 Power distribution of SRAM after synchronizing the SRAM

The table illustrates drastic improvements over the original Cascade generated SRAM. Figure 3.10 shows an improvement of: Precharge power savings = {71%,74%,67%}, and an Overall power savings = {55%, 55%,50%} for the periods = {20ns,15ns,10ns}.

3.6 Manipulating the Layout

Netlist simulations were shown to have some very promising results. The next step is to incorporate these modifications in the actual layout of the memories. The Cadence Design Frameworks is the tool that enabled changes in layout corresponding to the modifications discussed in the previous sections. The layouts of the modified cells are shown also in the previous section. An important design constraint for these cells was to leave the inputs and

outputs the same is the nominal case and to make sure the sizes of these cells were exactly the same as the nominal case. These two design constraints allow for these cells to be easily incorporated into a general design methodology utilizing automated tools.

3.6.1 Results

Simulation of the extracted netlist from Cadence was performed by HSPICE. The same vectors that were used on the original netlists of the previous section were also utilized here for comparison purposes. Figure 3.11 illustrates simulation results showing a typical outputs of the nominal SRAM, the modified version (senseamp and precharge modifications), and the synchronous SRAM. This particular simulation was at a cycle time of 15ns. This Figure shows that there is minimal delay increases in the generation of the output signal (Out3) in either the modified SRAM or the Synchronous SRAM.

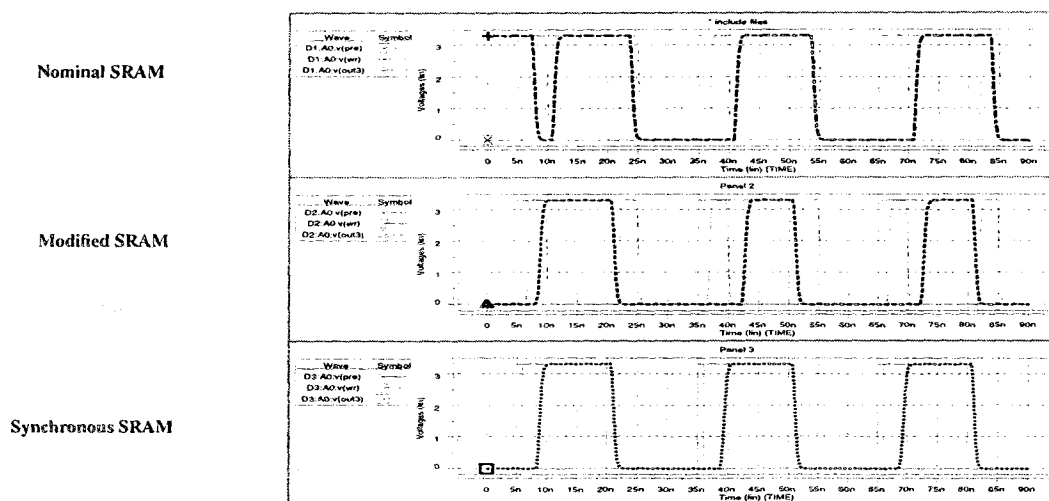


Figure 3.11 Comparison of outputs of SRAMs

Figure 3.12 shows the substantial power savings realized in the modified and synchronous SRAMs. For a clock cycle of 15ns, the modified version of the SRAM saves us over 34% in power dissipation from the original Cascade generated SRAM. Using the modified SRAM and changing it to a synchronous SRAM saves over 50% in overall power.

Simulation Period	nominal SRAM	modified SRAM	synchronous SRAM
period = 20ns	63.311mW	41.121mW (35.05%)	28.478mW (55.02%)
period = 15ns	70.354mW	45.912mW (34.74%)	31.296mW (55.52%)
period = 10ns	92.041mW	63.968mW (30.50%)	43.894mW (52.31%)
Average Savings		33.43% savings	54.283% savings

Figure 3.12 Power savings of modified SRAMs

3.7 Memory Segmentation

Although the savings from performing the methods stated in the previous sections was substantial, there still is more room for improvement. The precharge circuitry is still by far the dominating factor in power numbers due to the large amount of capacitance inherent in the bit and bit_{bar} lines. Segmentation is one method used to reduce the effective capacitance of those lines by precharging only a small portion of the total array. Figure 3.13 illustrates our technique to segment the 512 word memory. The left picture shows that we are dividing the entire SRAM into 16 smaller “segments”. Each of these segments contains 32 rows and 24 columns (1 BPC). The idea is to selectively precharge only one active segment at a time, effectively reducing the power dissipation substantially.

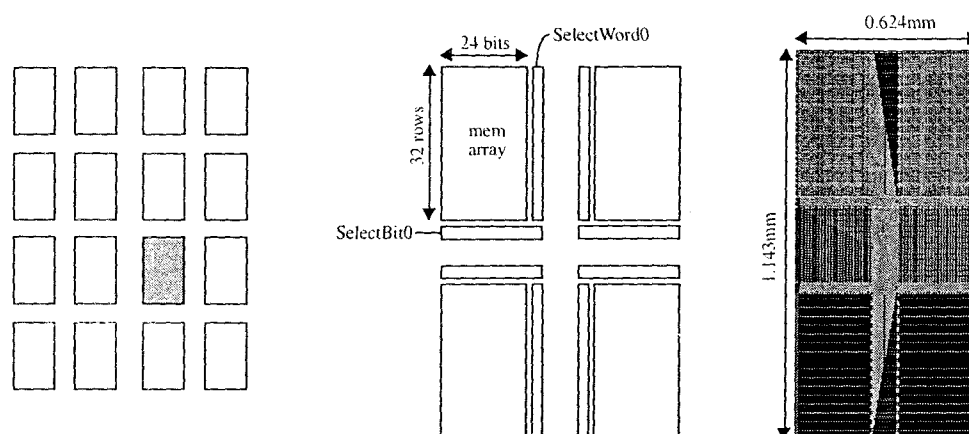


Figure 3.13 Segmentation of SRAM

3.7.1 Architecture of Segmented SRAM

The memory array was split up into 16 equal parts. These segments utilized the same 6-T memory cell of Figure 3.3 on page 16. A word enable and bit enable switch was placed at the periphery of the basic 32 row by 24 column SRAM array segment. This switch consists of an n-channel pass transistor and an enable line generated by control logic. The architecture exploits a shared Sense Amplifier and a decoder that chooses one row out of 32. Figure 3.14 shows the actual implementation of the word decoder. The decoder uses a 2-input NAND gate and a 3-input NAND gate inputted into a 2-input NOR gate. A large buffer drives the word activation line. There are 32 of these structures, only one active for a given address. Data lines have also been segmented into several sections and only one section is active during a *read* or *write* operation. A shared sense amplifier similar to the senseamp described in Figure 3.5 was used. This sense amplifier senses small changes in data lines

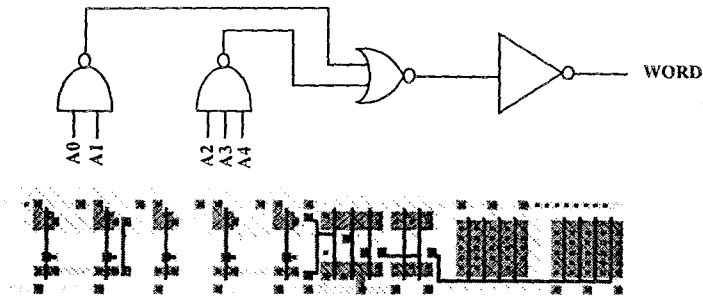


Figure 3.14 Word Decoder

and amplifies that change as an input into a latch. This helps reduce the delay increases inherent in this design. This design utilizes similar techniques to reduce the operating capacitance of long word and data lines as the proposal by [12].

3.7.2 Results

A comparison between the segmented SRAM and the SRAM of the previous section was accomplished. Specific areas of concern were the total area, access time, and measured power dissipated. The power and speed measurements were achieved through simulations in HSPICE. Netlists were extracted from the Cadence Design Frameworks.

The area of the proposed segmented SRAM was measured at 2.286mm X 1.302mm producing a total area of 2.976mm². The SRAMs described in the previous section had an area measured as 1.841mm X 1.0038mm producing a total area of 1.848mm². This is more than a 61% increase in the total area.

The access times of the memories explored in the previous sections were on the order of 5ns or better. The access times of the segmented SRAM are considerably slower. We measured an average access time of less than 15ns for the segmented SRAM. The large increase in delay was due to the fact that logic that determines which partition to activate dominates the time required to gain access to a memory cell. The measurements indicate a 200% increase in access times.

Power reduction was our main concern when partitioning the memories of this section. The segmented SRAMs were compared to those memories of the previous section running at a cycle time of 20ns. Results indicate that the segmented SRAMs generate only 17.8mW of total power as compared to the 28.478mW of the previous section. This is an improvement of 37% over the previous section's memory.

Thus, results indicate that if an area and delay increase can be incorporated into a design, the segmented memory can drastically reduce the power dissipated by the SRAM.

4 Conclusions

We have presented a methodology for low power design of SRAM memories. This method can be applied to electronic design automation tools to ease the burden on industry designers. We first examined some of the previous methods in designing low power RAMs. We looked at the architecture level issues such as partial activation of divided word and data lines and partial activation of divided power lines. Also we explored the pulsed operation of word line circuitry. Next we looked at some circuit issues such as current mode operation and an on-chip voltage down converter. Lastly, we reviewed how QuadRail techniques can be applied to SRAM design. Chapter 3 explored the design of low power memories. Starting with automatically generated memories, we reduced the power by over 50% without serious affects of access times. Finally we looked at the feasibility of partitioning memories into several smaller “segments”.

Several modifications to the segmented array can be done to improve the large increase in access times incurred. Specifically, a new decoder which does not use the n-channel pass transistors to partition the segments from the main data and word lines could improve both access times and lower the power dissipated even further. The charging of the main data

Conclusion

and word lines was the dominating factor in power dissipation numbers. A new method of partitioning these lines could be attempted. Current mode sensing and writing could also be promising when looking into these partitioned memories. This might speed up the access times even more. Further research into the partitioning of these memories looks promising.

Bibliography

- [1] *Epoch User's Manual*. Cascade Design Automation Corporation, Bellevue, WA, 1996.
- [2] A. Chandrakasan, R. Broderon. "Minimizing Power Consumption in Digital CMAS Circuits," *Proceedings of the IEEE*, pp. 498-523, April 1995.
- [3] P. Landman, R. Mehra, J. Rabaey. "An Integrated CAD Environment for Low-Power Design," *IEEE Design and Test of Computers*, pp. 72-82, Summer 1996.
- [4] O. Minato, et al., "A 20 ns 64K CMOS RAM," *ISSCC Dig. Tech. Papers*, pp. 222-223, Feb. 1984.
- [5] T. Sakata, et al., "Subthreshold-current reduction circuits for multi-gigabit DRAM's," *Symp. VLSI Circuits Dig. Tech. Papers*, pp. 45-46, May 1993.
- [6] K. Sasaki, et al., "A 7 ns 140 mW/Mb CMOS SRAM with current sense amplifier," *ISSCC Dig. Tech. Papers*, pp. 208-209, Feb. 1992.
- [7] Y. Shimazaki, et al., "An automatic-power-save cache memory for low-power RISC processors," *IEEE Symposium on Low Power Electronics*, pp. 58-59, 1995.
- [8] D. Singh, J. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal, T. Mozdzen. "Power-conscious CAD tools and methodologies: a perspective," *Proceedings of the IEEE*, pp.570-594, April 1995.

-
- [9] C. Su and A. Despain., "Cache design trade-offs for power and performance optimization: A case study," *IEEE Symposium on Low Power Electronics*, pp. 63-68, 1995.
 - [10] H. Sutioso., *Low Power Memory IC Design*, M.S. Thesis, Carnegie Mellon University, 1996.
 - [11] H. Tanaka et al., "Stabilization of voltage limiter circuit for high-density DRAM's using pole-zero compensation," *IEICE Trans. Electron.*, vol. E75-C, no. 11, pp. 1333-1343, Nov. 1992.
 - [12] M. Yoshimoto, et al., "A 64Kb CMOS RAM with Divided Word Line Structure," *ISSCC Dig. Tech. Papers*, pp. 58-59, Feb. 1983.