SCHEMES FOR REDUCING POWER

AND DELAY IN SRAMS

By

KATIE ANN BLOMSTER

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

AUGUST 2006

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of KATIE ANN BLOMSTER find it satisfactory and recommend that it be accepted.

____

Chair

____

____

Acknowledgment

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Funding has also been provided by the Boeing Centennial Chair in Computer Engineering, the School of Electrical Engineering and Computer Science at Washington State University, as well as the following scholarships and fellowships: Howard P. Curtis Scholarship and the Frank Harold EE Fellowship.

This research was conducted as part of the High Performance Computer Systems (HiPerCopS) group at Washington State University under the heading of Dr. José Delgado-Frias. Many thanks are extended to Dr. Delgado and to each of the members of the team, especially Mitchell Myjak, Jonathan Larson, and Suryanarayana Tadapudi.

The author would like to extend a special thank you to her parents, John and Sheree Blomster, as well as to her dear sister, Christie, and to her biggest supporter, Jerad Park. And, to the One without which none of this would have been possible, I give all the glory.

SCHEMES FOR REDUCING POWER

AND DELAY IN SRAMS

Abstract

by Katie Ann Blomster, M.S.
Washington State University
August 2006

Chair: José G. Delgado-Frias

Static random access memories (SRAMs) are used in a wide variety of applications ranging from ICs to embedded systems. As the demand for systems to reduce power consumption and increase speed continues to grow, these design requirements are passed on to a system's components, including SRAMs. In this thesis a number of novel schemes for reducing power and delay in SRAMs are presented. Memory access incorporates two different operations: the memory read and the memory write. To improve the performance of the memory write operation, seven different memory cell designs are proposed. Each of these designs has been extensively simulated in 180-nm CMOS technology for comparison with the standard six-transistor (6T) differential memory cell. The three cells performing the best in terms of energy consumption, delay and the energy-delay product demonstrate improvements of 27.6%, 12.3%, and 24.1% over the 6T cell, which uses 27.75 fJ and has an overwrite delay of 83.24 ps each write cycle.

The memory read circuitry is modified at the column level as well as the cell level. To save the power used in pre-charging and pulling down the bit-lines each read access, the pre-charge signal and its pull-up transistors are removed. A series of bit-line pull-up schemes that only switch the bit-lines when necessary are discussed and the most effective designs are

simulated thoroughly using 180-nm CMOS technology. In comparison with the standard reading design, the novel delayed bit-line capture pull-up (DBCP) scheme yield minimum energy savings of 44.3% and improves delay by at least 22.7%.

To ensure that the read logic works in conjunction with the memory write, four different test SRAMs containing the novel read designs are built in 90-nm technology. Each novel SRAM is compared with the standard SRAM implementation, which has a delay of 474.3 ps and dissipates an average of 521.3 fJ over a row of memory cells. The best results are achieved by the DBCP min style SRAM, with 28.6% power savings and a 39.6% improvement in delay. A discussion of the design tradeoffs when using a novel reading scheme is also included.

Table of Contents

Acknowledgment **iii**

Abstract **iv**

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 Static Random Access Memory 1

1.1.1 Features 2

1.1.2 Performance Metrics 4

1.2 Recent Work 6

1.3 Outline 7

2 Memory Write **8**

2.1 Memory Write Process 8

2.2 Performance Problems 10

2.2.1 Power Dissipation 10

2.2.2 Delay 12

2.3 Proposed Improvements 12

2.4 Design Implementation and Test 15

2.5 Measurement Techniques 18

2.6 Results 20

2.7 Analysis 22

2.8 Summary 25

3 Memory Read 27

3.1 Memory Read Process 27

3.2 Performance Problems 29

3.2.1 Power Dissipation 29

3.2.2 Delay 30

3.3 Prospective Bit-line Pull-up Designs 30

3.3.1 Two Stage Bit-line Capture 31

3.3.2 Single Stage Bit-line Capture 35

3.3.3 Equalized Bit-line Read 36

3.3.4 Delayed Bit-line Capture 38

3.3.5 Cross-Coupled Bit-line Pull-up 40

3.4 Measurement Techniques 42

3.5 Results 44

3.5.1 Conventional Reading Results 45

3.5.2 Delayed Bit-line Capture Results 46

3.5.3 Cross-Coupled Bit-line Pull-up Results 49

3.6 Analysis 54

3.7 Summary 59

4 Read/Write Combination **61**

4.1 Combined Memory Access Description 62

4.2 Measurement Techniques 64

4.3 Results 66

4.4 Analysis 69

4.5 Summary 76

5 Conclusion **78**

5.1 Contributions 79

5.2 Future Work 81

Bibliography **83**

List of Tables

Table 2.1: The 6T memory cell and its seven novel variations 16

Table 2.2: Simulation results for a 3 ns memory overwrite.. 21

Table 3.1: Standard read energy and delay data 45

Table 3.2: DBCP read energy and delay data 47

Table 3.3: CCBP (15λ) and $CCBP_{WEAK}$ (3λ) read energy and delay data 49

Table 3.4: Standard read delay (ps) 51

Table 3.5: $CCBP_{WEAK}$ read switching delay (ps).. 52

Table 3.6: Standard read switching energy (fJ) 52

Table 3.7: $CCBP_{WEAK}$ read switching energy (fJ) 52

Table 3.8: Standard read holding energy (fJ)... 53

Table 3.9: $CCBP_{WEAK}$ read holding energy (fJ)... 53

Table 3.10: Comparison of features for each reading technique 55

Table 4.1: Standard SRAM access energy and delay 67

Table 4.2: DBCP SRAM access energy and delay 68

Table 4.3: CCBP SRAM access energy and delay 69

Table 4.4: Total energy supplied to each SRAM (fJ) and average energy savings 72

List of Figures

Figure 1.1: Column of SRAM cells and external circuitry 3

Figure 1.2: Six-Transistor Memory Cell 4

Figure 1.3: Pipelining memory access—each stage is allowed excess time to execute 5

Figure 2.1: Basic memory latch for writing 9

Figure 2.2: V_{DD}-to-Zero dynamic power dissipation 11

Figure 2.3: One-to-GND dynamic power dissipation 11

Figure 2.4: Short-V_{DD}-GND dynamic power dissipation 12

Figure 2.5: Additional transistors and signals for improved writing 13

Figure 2.6: Timing diagram for memory write 14

Figure 2.7: Diagrams of the standard 6T memory (a.) and its seven variations: VG_N (b) VV_N (c) VGV_N (d) 8T (e) VG_C (f) VV_C (g) VGV_C (h) 17

Figure 2.8: Write simulation for a VG_C memory latch 21

Figure 2.9: Write simulation for a VV_N latch 23

Figure 2.10: Plot of delay versus energy for each novel memory latch 24

Figure 2.11: Plot of additional transistors versus energy-delay product for each novel memory cell design 24

Figure 3.1: Basic memory cell with pre-charge 28

Figure 3.2: Three stage memory access pipeline 30

Figure 3.3: SRAM column with two stage read logic and pull-up circuits PU_A-PU_D 33

Figure 3.4: SRAM column with two stage read logic and pull-down circuits PD_A-PD_C 34

Figure 3.5: Equalizing bit-line read circuit 37

Figure 3.6: Delayed bit-line capture pull-up circuitry 39

Figure 3.7: Cross-Coupled Bit-line Pull-up Circuits 41

Figure 3.8: Standard read with 60λ pre-charge pMOS drivers 46

Figure 3.9: DBCP read switch simulation 47

Figure 3.10: Standard and DBCP power comparison with the signal driver power included..... 48

Figure 3.11: Standard and DBCP power comparison without the signal driver power.............. 48

Figure 3.12: Simulation of a standard read memory access causing a bit-line switch 50

Figure 3.13: Simulation of a $CCBP_{WEAK}$ read memory access causing a bit-line switch............ 50

Figure 3.14: Comparison of standard and $CCBP_{WEAK}$ read instantaneous power....................... 54

Figure 3.15: Surface plot of $CCBP_{WEAK}$ read data *(data from Table 3.5)* 56

Figure 3.16: Energy used for different initial Bit- and NBit-line voltages during standard bit-line switches and holds 58

Figure 4.1: Comparison of supplied row energy 70

Figure 4.2: Comparison of supplied column energy 71

Figure 4.3: Energy dissipated through the memory cell to GND 73

Figure 4.4: Energy dissipated through the column bit-lines to GND 74

Figure 4.5: Delay comparison between the different SRAM designs 75

Dedication

This thesis is dedicated to Kyle and Andrew:

the two adorable and noisy twins upstairs.

Chapter 1

Introduction

The recent increase in mobile, hand-held, and battery operated devices as well as the increase in data transfer rates demands that these systems use less power and reduce operational delays. Since memory currently makes up a large part of systems, nearly fifty percent, reducing the power and delay in memories becomes an important issue. In fact, some systems, such as the reconfigurable hardware discussed in [1], which uses the memory latch as its most basic component, have an even larger percentage of their structure built with memory. Memories are also responsible for almost half of the total CPU dissipation. This has been shown to be true in some of the more power efficient designs with on-chip memories [2,15]. In cases like these it is necessary to determine the sources of power consumption and delay in memory blocks and cells so that they can be removed or reduced, allowing for better overall performance of the system.

1.1 Static Random Access Memory

Static random access memories (SRAMs) are used extensively in all kinds of systems and are found in almost every integrated circuit as an embedded component. They are known for their large storage density and small access latency [2,7]. This section discusses the features of the

SRAM block and the different metrics used to analyze and compare different SRAM implementations.

1.1.1 Features

A block of SRAM consists of the following features: a row decoder (and column decoders in larger memories), input buffers, bit-line conditioning circuitry, output sensing logic and buffers, and an array of memory cells (or latches). Fig. 1.1 shows a single column of SRAM cells with the bit-line conditioning circuitry and output sensing logic. Any number of memory cells can be placed on a column as long as the lines do not become too long or contain too much capacitance to operate properly. With similar limitations, a memory column can be joined up with as many other columns as desired to build a wide range of memory sizes. However, larger memories will experience larger delay. The word-lines are connect between columns to allow for single row access, as are the pre-charge signal, PRE, and the write enabling signal, WE. The row decoder controls each word-line in the memory. The word-line connected to an entire row is also known as the W/R signal since it determines (by controlling the access transistors) when a memory latch can be read from or written to.

The bit-line conditioning circuitry refers to either the logic used to write the input data onto the bit-lines or the pull-up logic used in pre-charging the bit-lines. Output sensing logic typically includes either sense amplifier circuits or output sensing inverters, which are used to determine more rapidly the direction of the bit-line swing and generate an output. The memory cell can be implemented in a number of ways as long as it can function with differential bit-lines. This means that accesses to memory expect one line to be a logic ‘1’ value when the other line is a logic ‘0’. The input and output circuits know how to interpret the values on the bit-lines to reflect the correct single bit of information to be stored to or retrieved from each memory cell.

The differential SRAM cell design used most commonly is the six-transistor (6T) latch, due to its stability, large noise margins, and relatively small size [5,8]. Other possible cells that could be used include the 12T and 4T latches [5]. Shown in Fig. 1.2 is the 6T SRAM cell, consisting of a pair of cross-coupled inverters and two access transistors, N1 and N2. Data is stored in the cell because of the feedback capabilities of the inverters. If node A is used to represent the stored bit, then holding V_{DD} on node A and GND on node B means that a logic '1' is stored in the cell. If GND is being held on node A and V_{DD} on node B, then a logic '0' is being stored in the memory.

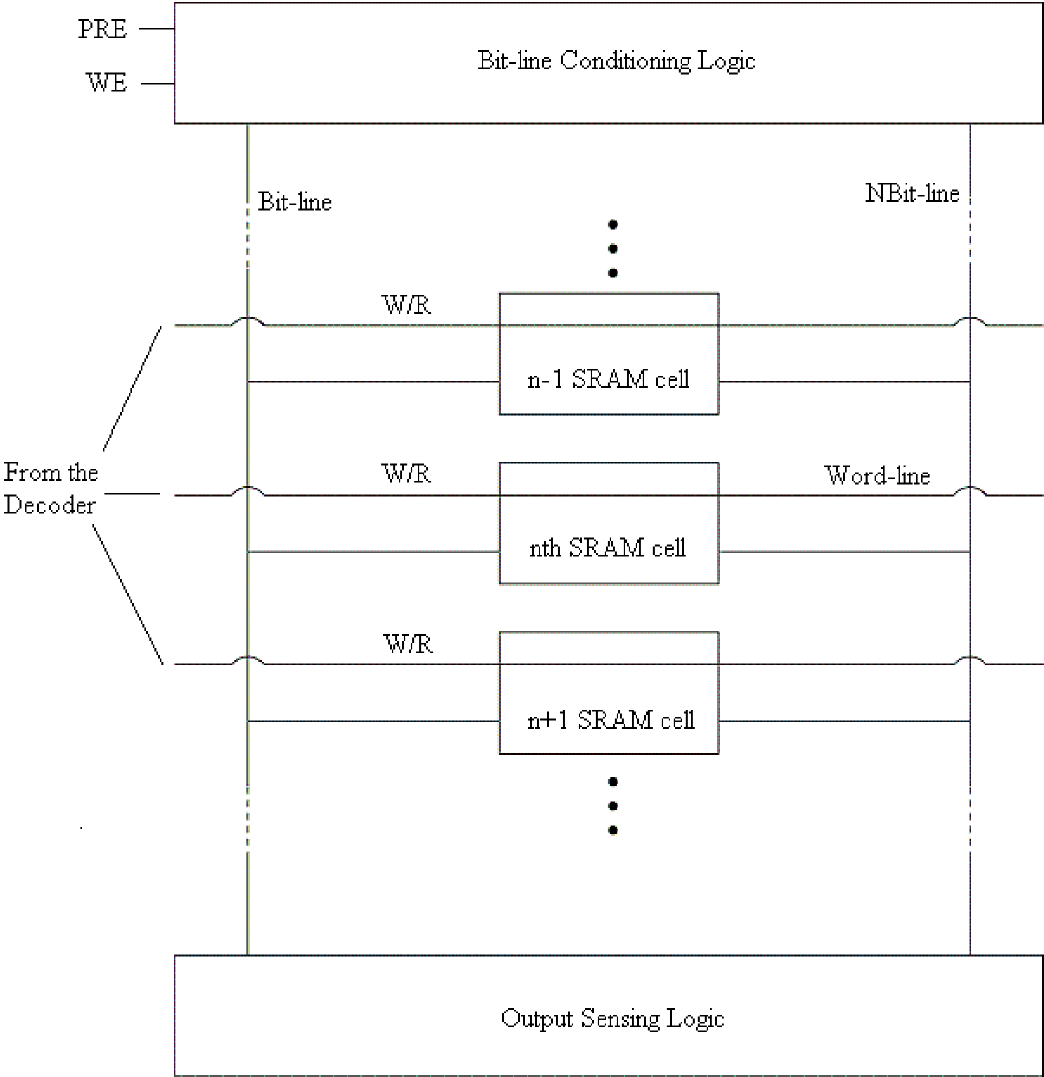

Figure 1.1: Column of SRAM cells and external circuitry

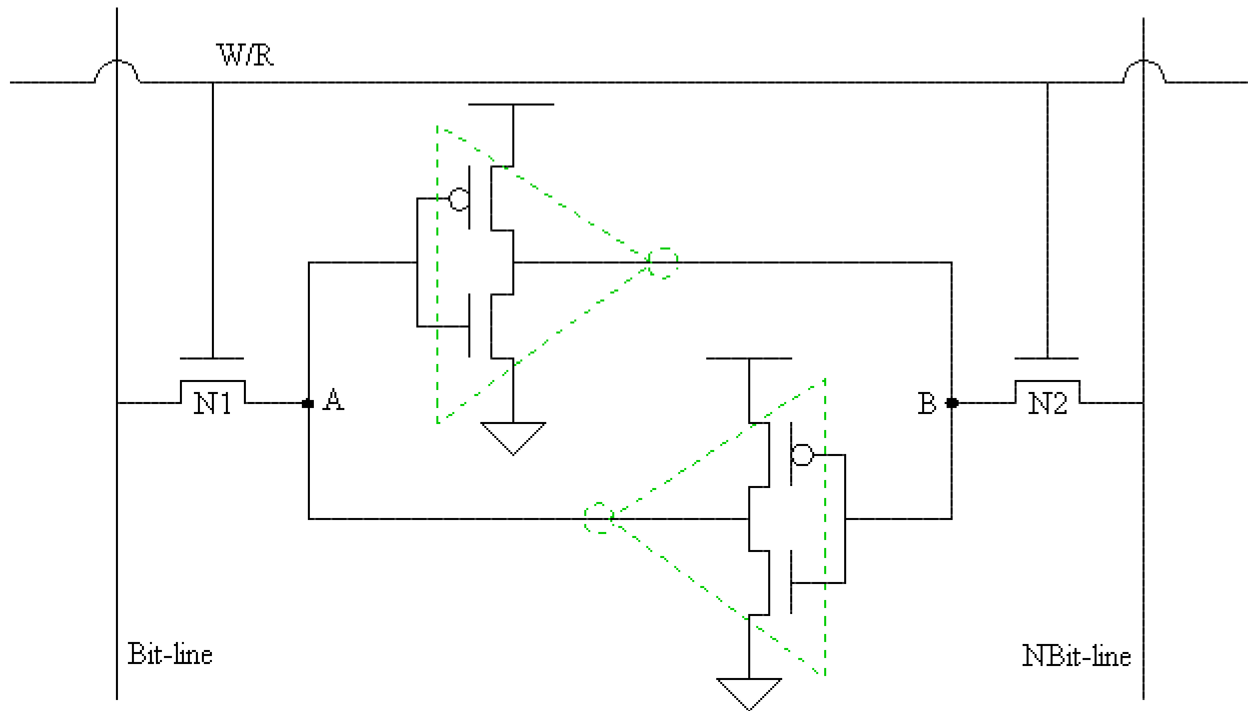

Figure 1.2: Six-Transistor Memory Cell

Writing to memory refers to storing a bit of data into a memory cell and reading from memory means that a stored bit of data is retrieved from a memory cell and placed on the bit-lines to be captured in the output latches. A memory access begins with the arrival of an address at the decoder, which then selects a word-line, followed by a memory read or write, and then with the generation of an output for the memory read. The process of accessing a block of SRAM for a memory write or read is complicated so no further explanation of memory access will be described here; however, Sections 2.1 and 3.1 will describe the write operation and read operation in great detail.

1.1.2 Performance Metrics

When measuring and comparing the performance of SRAMs, the following terms usually are involved:

- *Size:* One goal in the design of SRAMs is to keep them small and compact so that very large memories can be built on a very small amount of silicon.
- *Energy or power consumption:* Because of the density of SRAMs, power consumption can be a problem because of the heat and noise that result. Power consumption is also a primary concern because of the increasing number of mobile and battery operated devices on the market. Reducing the power in SRAMs is critical part of reducing the power in the systems they exist in.
- *Delay:* The meaning of delay changes depending on the situation. For non-pipelined memory access, delay simply refers to the *response time* or the total time between the appearance of the row address at the decoder and the arrival of the data read from memory at the output latches. The other approach is *throughput*, which is used in conjunction with pipelining. Throughput looks at the time it takes between processing each consecutive memory access, not the total time to complete each access. This is best for high performance systems which expect to receive frequent requests to access memory [6].

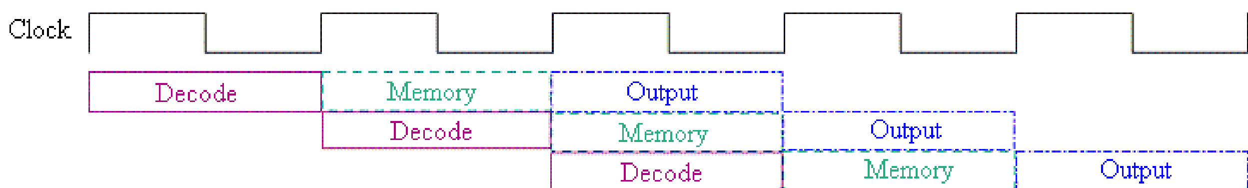

Figure 1.3: Pipelining memory access—each stage is allowed excess time to execute

By pipelining each read or write requests, a memory access can begin each clock cycle (assuming that the pipeline stage lasts only one clock period) instead of waiting for the previous memory access to attain its output before starting the next read or

write operation. Fig. 1.3 shows how pipelining is achieved in a system. First, the operation (memory access in this case) is broken into stages, preferably as similar in terms of execution time as possible. These stages can then be operated in parallel since they each take place in a different section of the hardware and because they all can be completed within the chosen pipeline cycle time. The problem with this implementation is that the longest stage in the pipeline determines the cycle length for every stage, causing the response time to be much larger than it would be without pipelining.

By using these different metrics in the study of SRAMs, useful comparisons can be made between different designs. Understanding the significance of each measurement is also important for providing a practical knowledge of what the most necessary areas of improvement should be during the design process.

1.2 Recent Work

A large amount of research focusing on low-power designs for memory cells has taken place recently. Many of the proposed designs take the approach of reducing the voltage swing levels or the capacitance on the bit-lines. This has been done by dividing the bit-lines into global and local lines. Sense amplifiers are places on the local lines to help with reading, and extra-wide high performance pMOS transistors are needed to supply the local lines with the values to be written. While this design yields great saving in terms of power and with little affect on the delay, there is a significant amount of additional circuitry needed for it implementation [3]. Another method involves hierarchically dividing the bit-lines so that they use their word-lines to allow access to the sub bit-lines [16]. This method performs well in 0.5 μm technology, but in more modern processes where interconnect capacitance has a much greater affect on delay.

A half-swing bit-line design and quiet bit-line architecture, which keeps the bit-lines near GND at all times, have also been created [3,7]. Divided word-line approaches have been taken as well to reduce the large line capacitance [3]. Other designs implement charge sharing, recycling, or recovering to reduce power [2,7]. The problem with many of these solutions is that by reducing power significantly, they tend to increase the delay of a memory access.

One other technique proposed to improve on the power to write logic ‘0’s to memory uses the single write, differential read (SWDR) cell. This design takes advantage of the fact that a majority of the data written to cache are logic ‘0’ values. Since the single write line will not discharge after being pre-charged unless it is writing a ‘1’, the bit-lines are not changed during a write. The SWDR cell requires an additional transistor between GND and one of the inverter nMOS transistors so that a write ‘1’ can be accomplished with only the single line. This technique results in a large reduction in cache write power without an affect on the delay [4]. Many of the designs mentioned here were influential in the work presented in this thesis.

1.3 Outline

This thesis is organized as follows. First, the memory write operation is studied in Chapter Two, and several methods to improve its performance are analyzed. Chapter Three discusses the memory read operation and presents several options for reducing its power and delay. Chapter Four combines the memory write and proposed read access circuitry into a single SRAM to analyze its performance and compare it with the standard SRAM design. Finally, some concluding remarks are made in Chapter Five.

Chapter 2

Memory Write

The SRAM block described in Chapter One consists of an array of memory cells, row and (if desired) column decoders, bit-line conditioning circuitry (for writing to and reading from a cell), and data input and output latches. Once the memory address to be accessed has been decoded and a word-line has been selected (by pulling its voltage level high), the stored value in each memory cell must either be overwritten with input data or read onto the bit-lines and into the output latches.

This chapter begins with a thorough study of the process of writing to memory and the sources of energy consumption and delay. Several SRAM cells designed to yield better performance or use less power are presented. A detailed discussion of the results reveals the benefits and tradeoffs that are attained with the proposed changes. Finally, each memory enhancement is evaluated.

2.1 Memory Write Process

When writing to a SRAM, only one row of memory latches can be accessed at any time. The process begins with a memory row address arriving at the SRAM's decoder. The decoder then

determines which word-line to select. Meanwhile, the data to be input into a row of memory arrives at the input latches. The bit-line conditioning circuitry then puts the correct voltage levels onto the bit-lines. Fig. 2.1 shows how by selecting a word-line (or pulling the W/R line high,) the two nMOS pass transistors, N1 and N2, are turned on. This allows current to flow between the conditioned bit-lines and the memory cell, and either charge or discharge the voltage levels stored at nodes A and B.

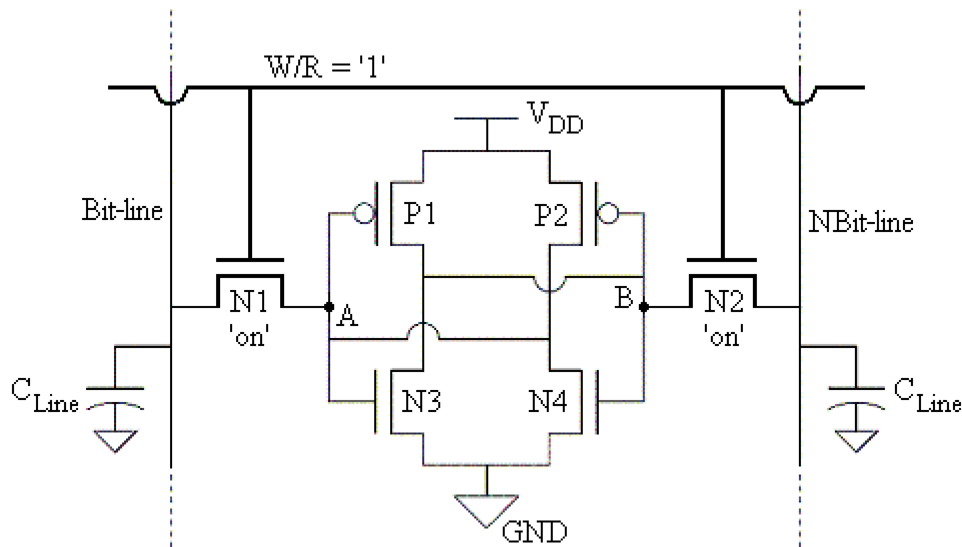

Figure 2.1: Basic memory latch for writing

Essentially, writing to memory is a two-step process. The first step involves the decoding of the memory address and conditioning the bit-lines to hold the proper values to be stored into memory. The second step includes turning on the access transistors and allowing the cell to be overwritten with the new data. The speed of a write access is determined by the time it takes to decode the address plus the time to pull the word-line high and pass the data from the bit-lines to the cell.

2.2 Performance Problems

There are several inherent problems in the design of the 6T SRAM. One flaw that becomes apparent during a 6T write operation is that significant power losses are allowed to occur. Delay is another aspect of the memory write that can be improved upon. It is, however, a secondary concern because typically the time to write to memory is already less than the time to read. This section is devoted to identifying the principle causes for power dissipation and delay.

2.2.1 Power Dissipation

Power is necessary for the proper functioning of an active memory latch: if the cell is to be overwritten during a write cycle, the capacitive loads at nodes A and B each have to either be charged or discharged. However, there are wasteful sources of dynamic power consumption that result from short circuits that exist when the memory cell is switching. There are three different short circuits: V_{DD}-to-Zero, One-to-GND, and Short-V_{DD}-GND, displayed in Fig. 2.2, 2.3, and 2.4, respectively. Each figure shows how the short circuits are created by switching nodes or transistors in the memory cell. V_{DD}-to-Zero exists between V_{DD} of the memory latch and the bit-line at logic '0'. One-to-GND is formed by the bit-line at logic '1' and GND of the cell. And, Short-V_{DD}-GND consists of the paths from V_{DD} to GND in each inverter in the memory cell since the inverter inputs do not have instantaneous transition times [5].

Other forms of wasteful power dissipation occur when the SRAM is inactive. These static power losses include sub-threshold conduction, junction leakage, and gate tunneling [5]. Leakage current and its consequences will not be analyzed in this thesis. Instead, dynamic power and delay are the main focal points.

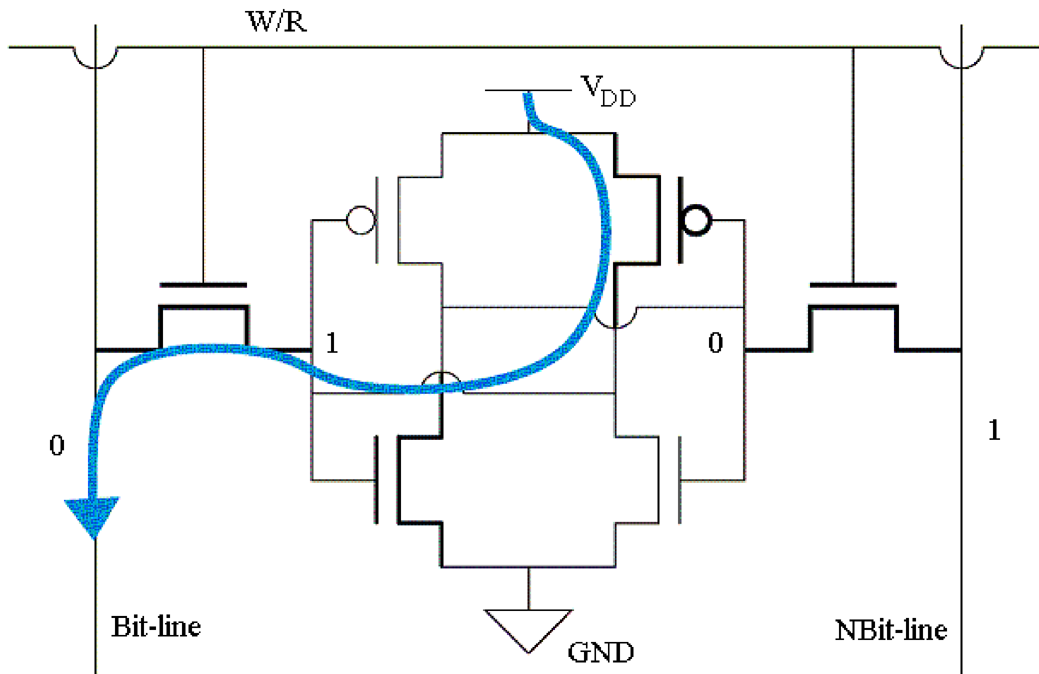

Figure 2.2: V_{DD}-to-Zero dynamic power dissipation

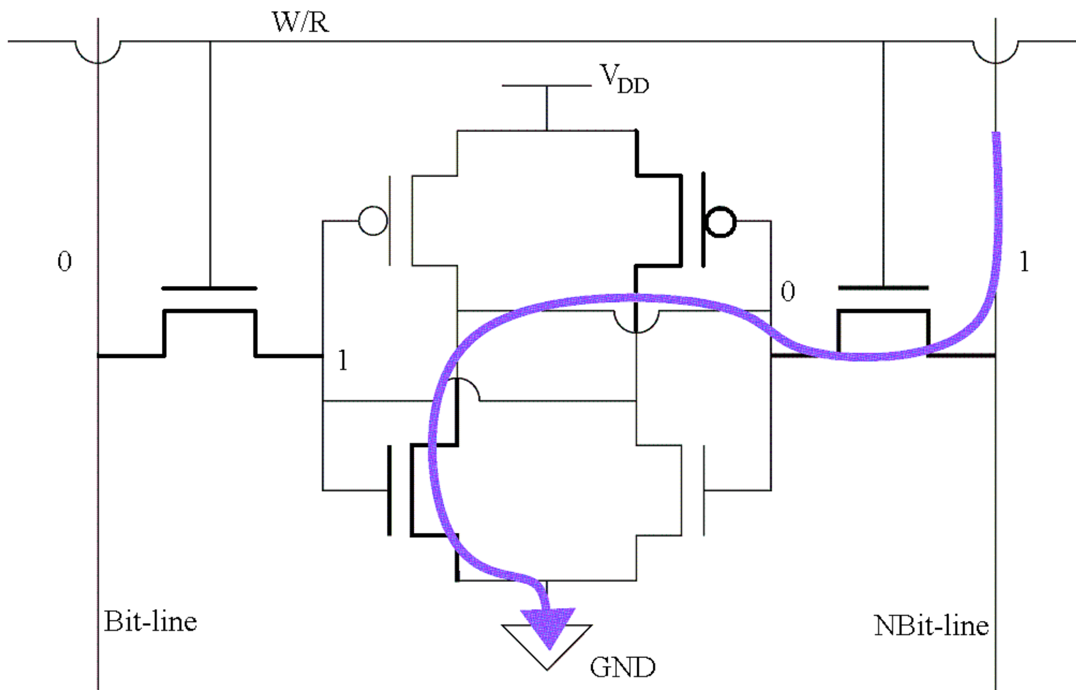

Figure 2.3: One-to-GND dynamic power dissipation

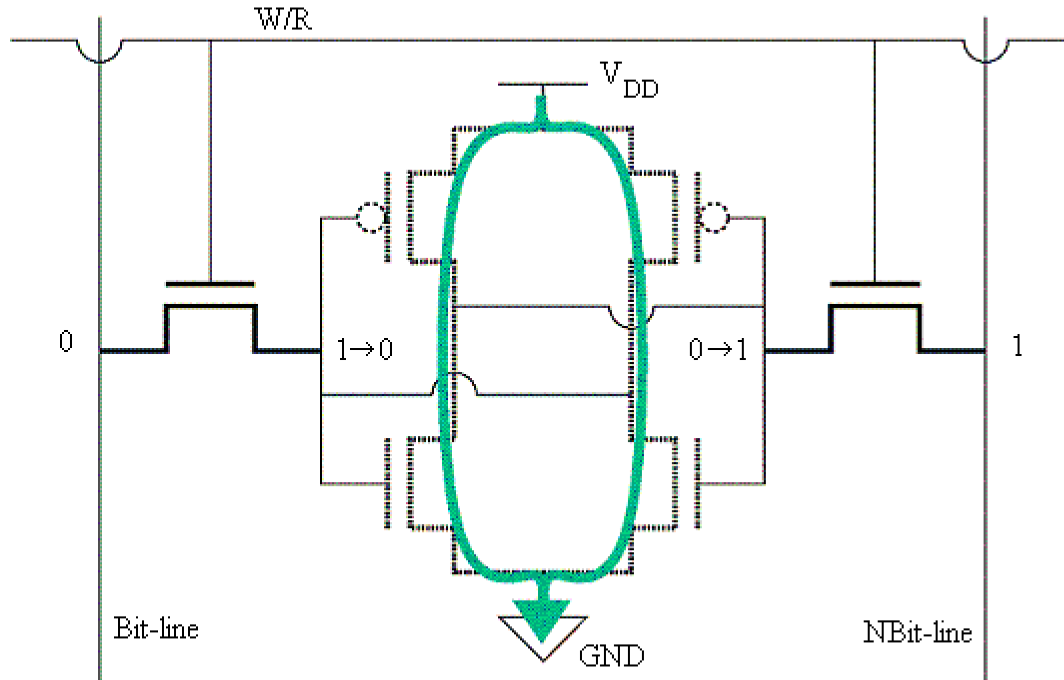

Figure 2.4: Short-V_{DD}-GND dynamic power dissipation

2.2.2 Delay

There are several causes for delay in the 6T SRAM. The first is due to the time it takes the input data to reach bit-lines, which depends on the switching time for both the external writing circuitry and the bit-lines. The others arise from the W/R word-line switching and N1 and N2 turning on, followed by the stored voltages in the memory latch being overwritten. All of these are directly related to the amount of capacitance on the lines or nodes being charged or discharged

2.3 Proposed Improvements

In the previous sections it was explained why power reduction is our primary goal in the design of SRAM cells focused on the write operation and why delay is only a secondary concern. Since short-circuits are responsible for much of the dynamic power loss, first we will look at methods

for removing each direct connection between V_{DD} and GND. Fig. 2.3 shows a transistor, T_{VG} inserted between GND and the source contacts of nMOS transistors N3 and N4. As two of the short-circuits terminate at the GND source contact in each memory latch, the addition of T_{VG} is an effective way of eliminating One-to-GND and Short-V_{DD}-GND simply by turning T_{VG} off. The same can be done with V_{DD} by adding a pMOS transistor between V_{DD} and the source contacts of pMOS transistors P1 and P2, although this time it is called T_{VV} and it is used to remove V_{DD}-to-Zero and Short-V_{DD}-GND. Including both T_{VG} and T_{VV} is redundant in terminating Short-V_{DD}-GND, but necessary if all three short circuits are to be removed.

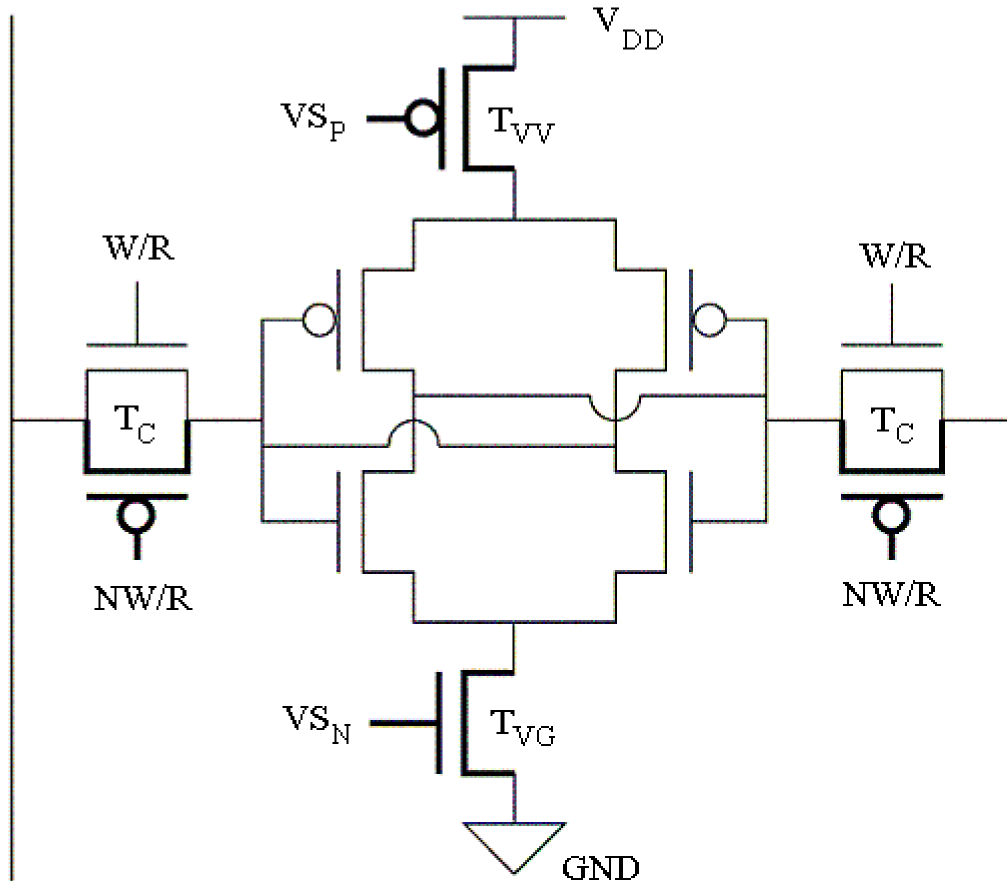

Figure 2.5: Additional transistors and signals for improved writing

T_{VG} and T_{VV} are called *virtual source transistors* as they are virtual suppliers of GND or V_{DD} to a memory cell (T_{VG} is Virtual GND and T_{VV} is a Virtual V_{DD} transistor). The signals that control T_{VG} and T_{VV} are called VS_N and VS_P. Both VS signals must be timed carefully so that T_{VV} or T_{VG} are off before the word-line of a memory is pulled high, which should not happen until the bit-lines have attained their proper voltage levels. The VS signals can be turned back on as soon as logic '0' has nearly been attained on the side of the memory latch that is discharging its load capacitance. A timing diagram of a memory write access with virtual source transistors is depicted in Fig. 2.6. Timing is important because dynamic power will only be reduced if the paths between V_{DD} and GND are cut off during the time that the memory cell is switching.

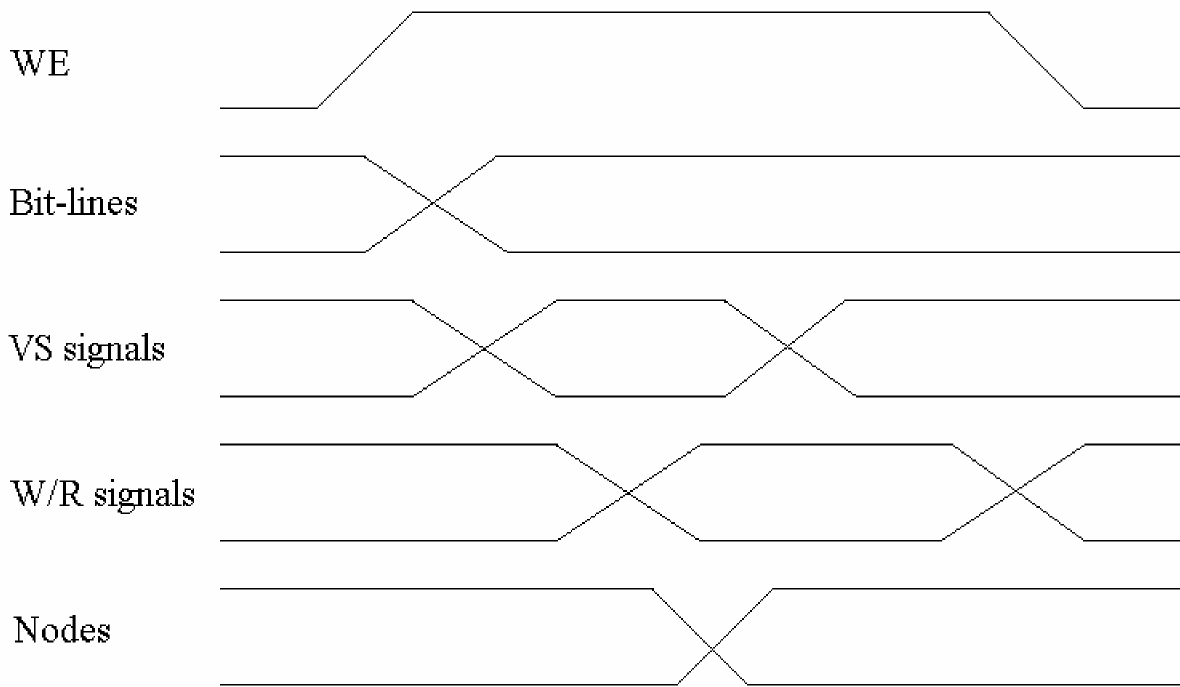

Figure 2.6: Timing diagram for memory write

Improving the speed of a memory write is the next step. To review, most of the causes of delay were external to the cell: input and bit-line conditioning circuitry and slow transitions of

the W/R word-line and access transistors. Within the memory latch, the causes of delay are the capacitive loads at nodes A and B that need to be charged or discharged and the nMOS access transistors which can each only pass a weak logic '1'. That means that instead of both nodes of the cell switching concurrently, one node (node A for example) slowly charges through N1 while node B quickly discharges through transistor N2. Node A cannot completely attain a logic '1' value until B fully discharges its capacitance and turns on transistor P2 to supply full V_{DD} to node A. A solution to this problem is to add two pMOS pass transistors to each memory latch. Once they are joined to each nMOS access transistor, as shown in Fig 2.5, they can be labeled as T_C, since each pair forms a CMOS transmission gate. Ideally, this will produce faster write times since both a strong '1' and '0' will simultaneously be written to the memory. This should lead to the added benefit of reduced dynamic power dissipation within the memory cell because of the shorter switching time for the cross-coupled inverters, which means less time for a short circuit to exist. However, it is important to remember that the additional pMOS transistors will also lead to greater capacitances on the bit-lines, and therefore greater power consumption and slower bit-lines switching speeds.

2.4 Design Implementation and Test

By using different combinations of virtual source transistors along with the CMOS transmission gates, seven distinct designs for SRAM cells can be created. Table 2.1 presents these variations of the 6T memory latch that we designed for improved writing performance. The standard 6T cell is also included for a comparison. The name of each cell describes its structure: for example, the CMOS Virtual GND/V_{DD} memory has CMOS transmission gates in place of the nMOS access transistors and both T_{VG} and T_{VV} are added as shown in Fig. 2.7h. Each design has an abbreviated name, which is listed in Table 2.1 as well. The subscripted N or C designates

which type of pass transistors the memory latch uses to access the bit-lines. Following the previous example, the name that is used from now on for the CMOS Virtual GND/V_{DD} memory is VGV_C.

Fig. 2.7 illustrates each of the novel memory cell designs. Once again, the 6T memory latch is included for reference. What should be noted in each diagram are the additional signals that need to be generated. Signals VS_N and VS_P are both produced in the decoder and are turned off only during write operations. The inverted W/R signal for controlling each T_Cp transistor is also supplied by the decoder. Intuitively, these signals will lead to greater power consumption and a slower operating speed for the decoder; however, it will be made apparent that the savings attained by each memory cell outweigh the losses due to the overhead circuitry.

Table 2.1: The 6T memory cell and its seven novel variations

Fig. 2.7	Memory Cell Name		T_{VG}	T_{VV}	T_C
a.	Standard Differential 6T	6T			
b.	nMOS Virtual GND	VG_N	X		
c.	nMOS Virtual V_{DD}	VV_N		X	
d.	nMOS Virtual GND/V_{DD}	VGV_N	X	X	
e.	CMOS 8T	8T			X
f.	CMOS Virtual GND	VG_C	X		X
g.	CMOS Virtual V_{DD}	VV_C		X	X
h.	CMOS Virtual GND/ V_{DD}	VGV_C	X	X	X

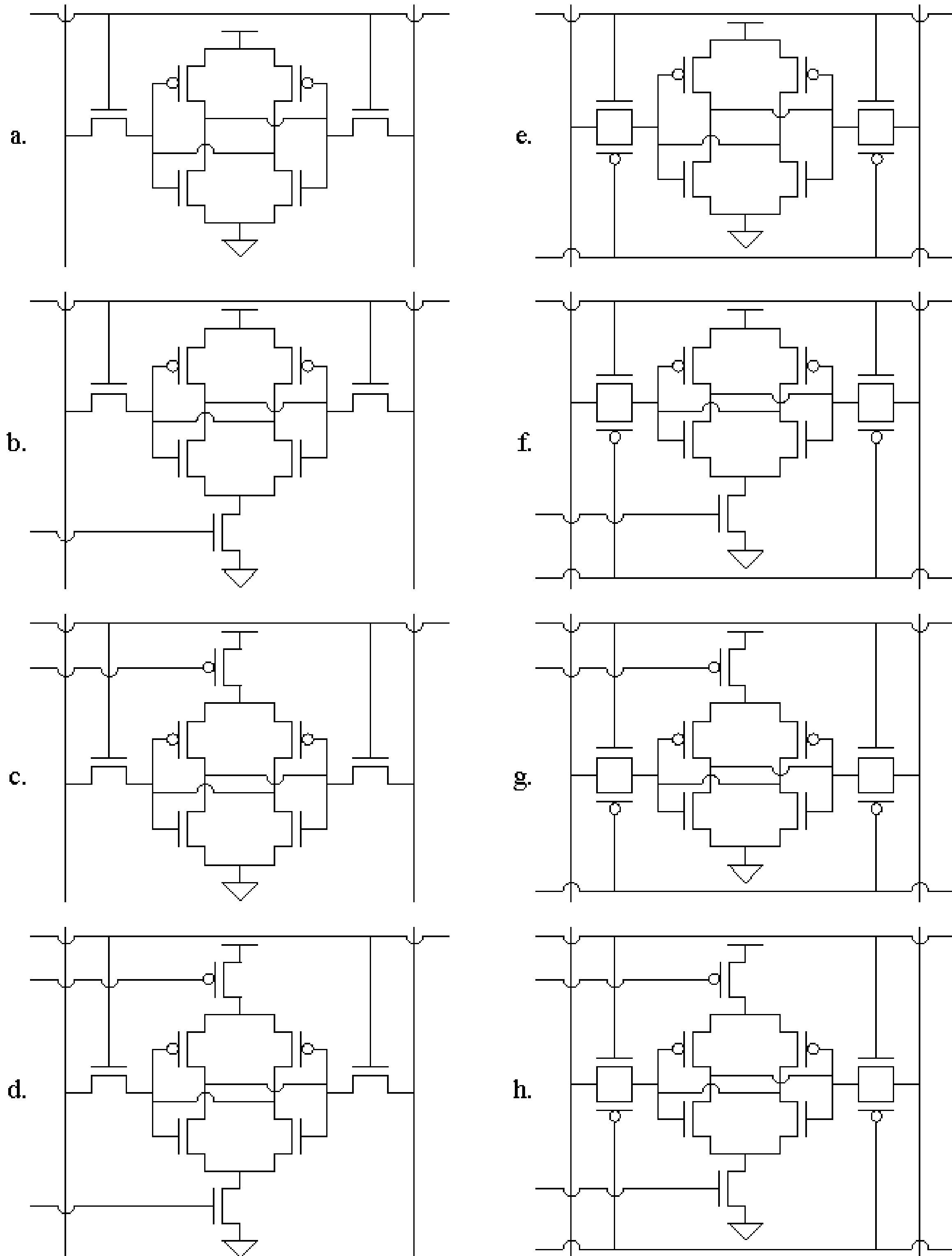

Figure 2.7: Diagrams of the standard 6T memory (a.) and its seven variations: VG_N (b) VV_N (c) VGV_N (d) 8T (e) VG_C (f) VV_C (g) VGV_C (h)

Transistor sizing within each memory latch is another matter that must be considered. For this reason it is necessary that each cell be tested over a range of transistor sizes in order to find the combination that yields the best performance for power or speed. Once the best performance has been determined for each design, an accurate and fair comparison can be made between the 6T latch and the seven new variations. Sizing of the T_C transistors is especially significant due to the large capacitance they add to the bit-lines. As was mentioned in Section 2.4, this causes greater power consumption when switching the bit-lines. If the bit-lines are switching at the same time as the decoding is taking place, the speed of the transition should not matter as much. For pipelining, this is not the case: the word-line switches at the same time as the bit-lines, so timing is much more important. All of these issues are taken into account in our simulations and will be analyzed further in the upcoming sections.

2.5 Measurement Techniques

This section explains the methods used to accurately test and measure each of the seven novel memory latch designs and to objectively compare the data with those for the 6T SRAM. Each memory cell is constructed using the Cadence Virtuoso Schematic Editor at 180-nm CMOS technology. Included in each schematic is a single latch, its external driving circuitry, and all of the capacitances and delays that would exist if the cell were part of a 32x32 bit memory. In setting up the simulations to be run on each memory, the bit-lines and internal nodes of the cell are set with initial conditions: each bit-line is given the opposite voltage level as that which is stored on the node it is connected though via the N1 or N2 transistors. Then, once the W/R word-line is turned on, a write *switch* takes place and the value stored in the memory is overwritten.

Proper timing of the simulations must be carefully composed. Recall from Fig. 2.4 that a memory write begins with the write enabling signal (WE) being turned on. Then the bit-lines start to attain their appropriate voltage levels as they are supplied with the proper input data. It is during this time that the VS signals are turned off. As was explained in Section 2.3, both the bit-line and VS signals must complete their transitions before W/R (and NW/R) can be turned on. If the VS signals are not turned off before the access transistors switch on, potential power savings that could be gained by including the virtual source transistors will be lost. By ensuring that each simulation is set up with the ideal timing, the true capabilities of the new designs will be demonstrated.

The timing regarding when the VS signals are switched back on depends on the cell being tested. Only in the memory latches VV_N and VGV_N is it important to turn VS_P on as soon as one of the internal nodes has nearly reached logic '0'. For all the other cells, it is simply required that the VS signals are turned back on before the writing sequence is completed. The trouble with leaving a VS signal on after a write access has been completed is that cell could easily be overwritten if its word-line was pulled high, which would yield faulty data when accessed later. To prevent that problem as well as guarantee that all of the tests are uniform, every VS signal begins to turn on after the discharging node has almost been pulled to logic '0'.

Before the simulations can be run, there needs to be a way to quantitatively measure and evaluate the success of each memory latch in reducing power dissipation and delay. For this reason, each memory latch schematic includes two power sources, one for the external circuitry and another for the memory cell itself. That way the dynamic power dissipated through short circuits can be measured separately from the power needed to supply the W/R or VS signals. The instantaneous power used within the latch is obtained by multiplying the current flowing

into the memory cell by the voltage supplied at that time. Energy is then calculated by computing the integral of the instantaneous power over a single write cycle.

Measuring the delay is less complicated. Writing propagation delay is calculated as the difference in time between when the transitioning W/R signal reaches 50% of V_{DD} and when the latter of the two switching internal nodes (A or B) of the memory cell reaches 50% of V_{DD}.

2.6 Results

Simulations of the 6T SRAM and its seven variations were run using the Cadence SpectreS simulator. This section reports the measured energy and delay values gathered from those simulations. The energy-delay product is also reported with the results since typically it is a more effective method of comparing data and not as easily manipulated by changing voltage levels or simulation times [5].

After some initial tests were performed, it was determined that memory cells containing minimum-sized transistors produced the results with the lowest power. The data presented in Table 2.2 were obtained from the simulations that were run on each minimum-sized memory latch. Every schematic contains a system of inverters and a NAND gate in order to generate and drive the W/R, NW/R, and VS signals. Since the number of signals to include as well as the amount of capacitance to drive varies between each memory implementation, some timing differences arise between individual tests. However, these differences only have a small impact on rise and fall times for the input signals so the transition times of the internal nodes A and B are minimally affected. Overall, the timings of the matching input signals are very similar between each simulation and they closely resemble those which appear in Fig 2.8. It should be noted that these tests do not include write enabling or bit-line transition times. The input data on the bit-lines is initialized and then held steady throughout the 550 ps simulations.

Table 2.2: Simulation results for a 3 ns memory overwrite

Configuration	Additional Hardware		Energy (fJ)	Delay (ps)	Product
	Transistors	Signals			
6T	0	0	27.75	83.24	2309.91
VG_N	1	1	22.66	77.42	1754.33
VV_N	1	1	22.19	103.00	2285.57
VGV_N	2	2	20.09	101.51	2039.33
8T	2	1	31.28	83.98	2626.89
VG_C	3	2	24.72	72.96	1803.57
VV_C	3	2	25.06	84.30	2112.55
VGV_C	4	3	22.40	80.07	1793.56

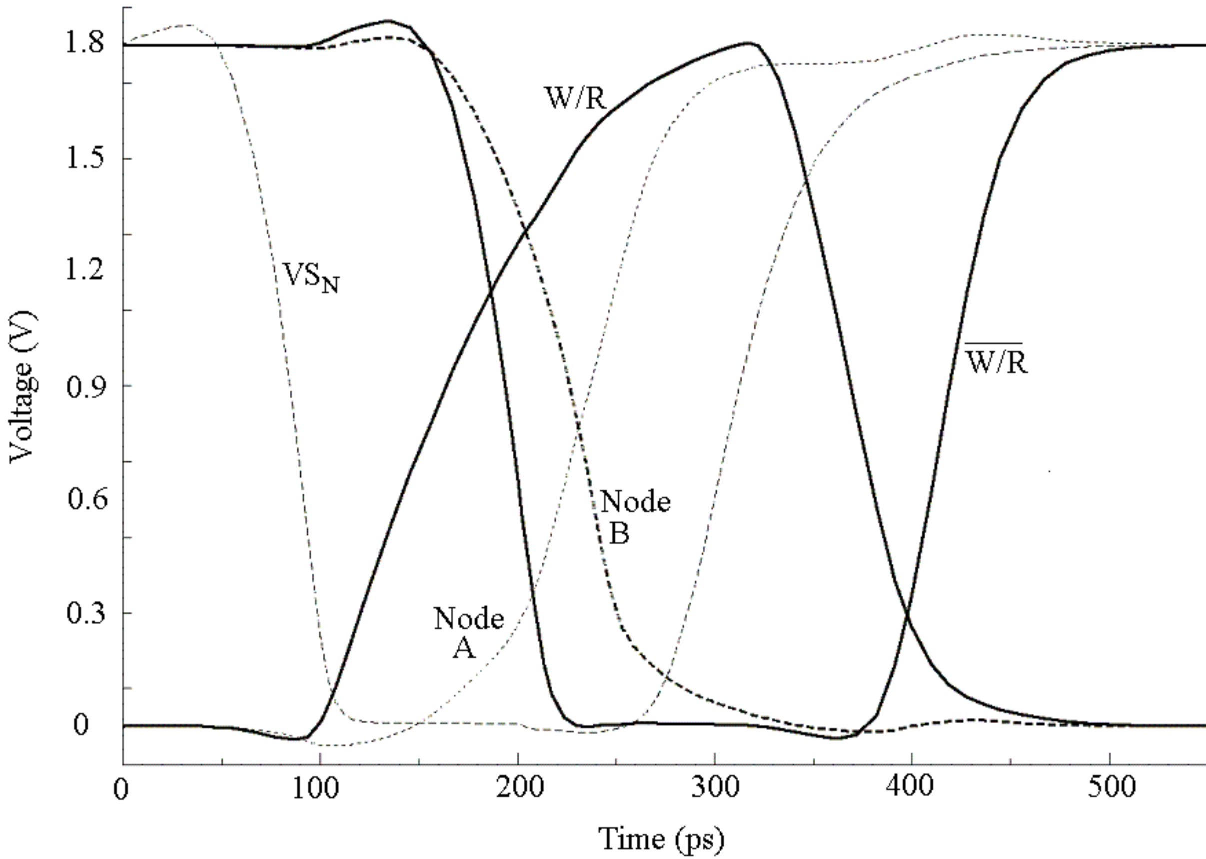

Figure 2.8: Write simulation for a VG_C memory latch

2.7 Analysis

At first glance, the results given in Table 2.2 reveal a different design that performs the best for each metric. Using the 6T SRAM as a standard for comparison, the VGV_N latch demonstrates a 27.6% decrease in energy dissipation. Delay is reduced by 12.3% when the VG_C cell is used in memory write operations. In terms of the energy-delay product, the VG_N SRAM exhibited 24.1% savings over the 6T cell. Based off of the discussion given in section 2.4, the memory cells performed as would be expected: the virtual source transistors helped to reduce power and including T_C in the memory cells decreased the speed to overwrite the stored value. An exception to this is the 8T memory. The remainder of this section will analyze the reason for this as well as discuss the tradeoffs in using any of the new latch designs and the situations in which each design would be most appropriate.

The first three memory cells with virtual source transistors, VG_N VV_N and VGV_N, were all successful in reducing the energy consumed in one write cycle; however, both of the cells with T_{VV} increased the delay by at least 21.9%. This is because node A cannot be charged as quickly while the source of V_{DD} is cut off from the P1 and P2 transistors: only a weak '1' is being passed through the access transistors. Then once T_{VV} is turned back on, V_{DD} starts to help the transition. The voltage-time curve for node A in Fig. 2.9 demonstrates this characteristic.

Adding T_C to those three memories to build VG_C VV_C and VGV_C proved to be an effective way to keep the power consumption low and at the same time reduce the writing delay for each latch. Unfortunately, for the VV_C memory, the delay is still larger than that of the 6T SRAM. The additional load capacitance in the cell because of T_C is enough to outweigh the benefits of a stronger '1' being provided to node A. The negative effects of the added capacitance can be seen even more clearly in the case of the 8T latch: not only is the delay

greater than the 6T cell, but it also consumes significantly more energy. Although a stronger '1' is provided through each T_C, the capacitive loads to charge and discharge are larger, making the cross-coupled inverters more resistant to change. Power dissipation is increased because the One-to-GND short circuit is stronger with the added p-type access transistor. Also, the longer transition delay allows Short-V_{DD}-GND to exist for an extended period of time.

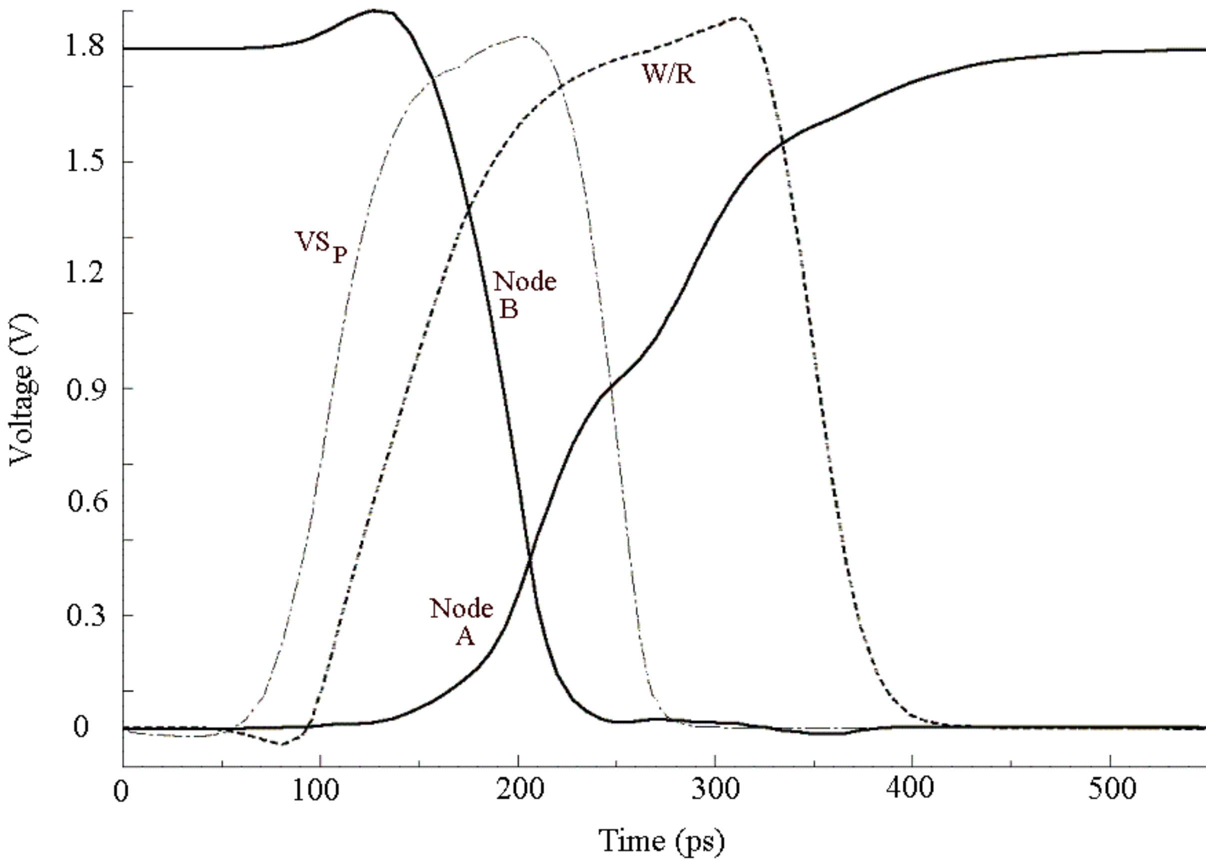

Figure 2.9: Write simulation for a VV_N latch

One method for improving on the problem of delay leading to larger power consumption is to alter the timing of the NW/R signal. Also, with larger T_C transistor sizes, speed can be increased even further, but not without increasing the energy dissipation as well.

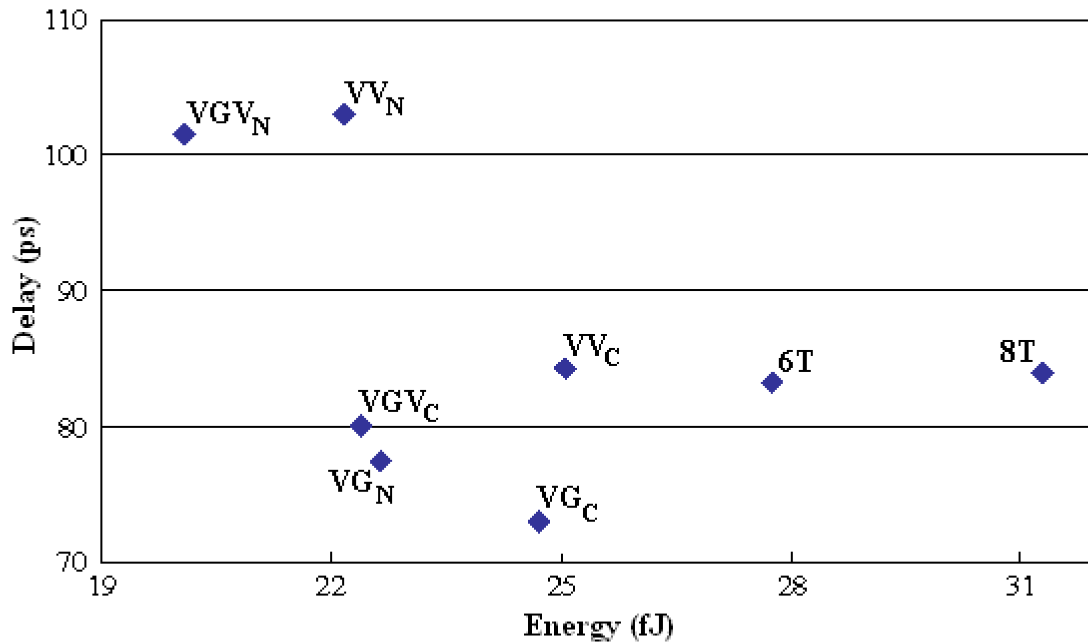

Figure 2.10: Plot of delay versus energy for each novel memory latch

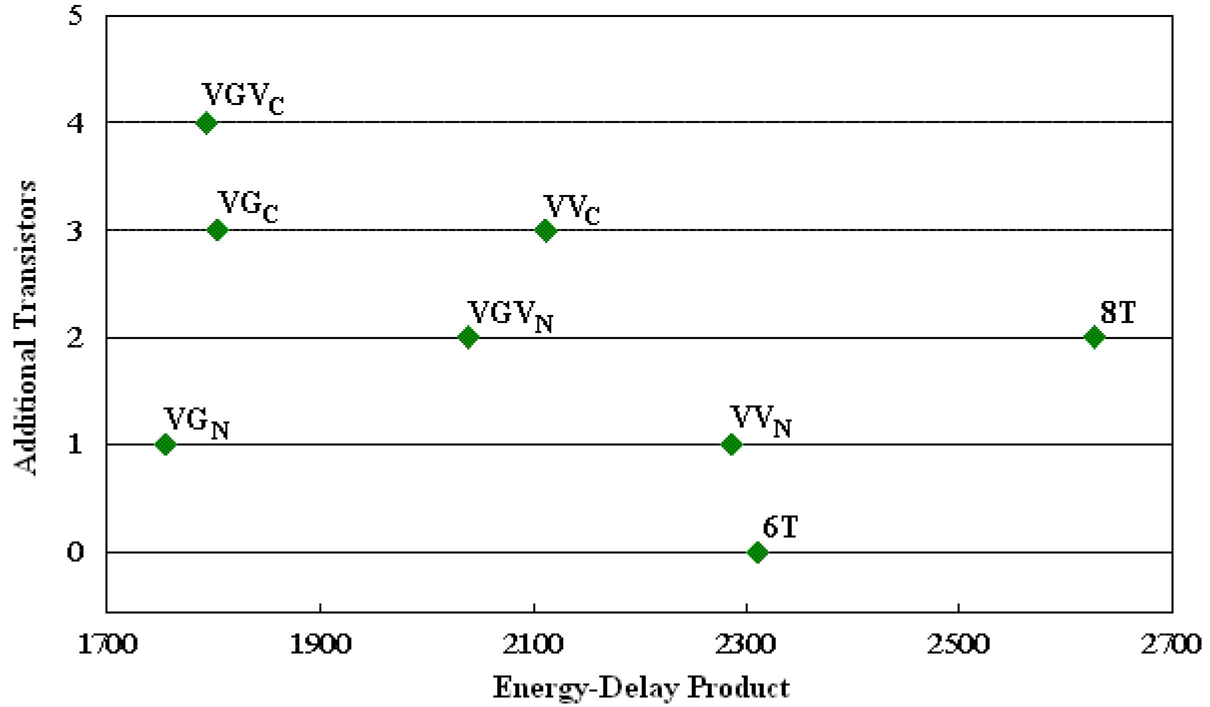

Figure 2.11: Plot of additional transistors versus energy-delay product for each novel memory cell design

The preceding discussion indicates that there are definite tradeoffs between power and delay in the implementation of memory cells. The plot in Fig 2.10 gives a more visual comparison of the results for each memory design. Including both T_{VG} and T_{VV} is an effective way to keep power consumption to a minimum, but those additional transistors do add to the size of the memory cell as well as the delay, VG_N excluded. If size is not a factor and reducing delay is the goal, it is worthwhile to add CMOS transmission gates along with the virtual source transistors to each SRAM cell. A more visual transistor count or size comparison with the energy-delay product is provided in Fig. 2.11 since low power, high speed, and size are usually all important considerations in memory design (as explained in Section 1.1.2). This plot shows that the VG_N latch is a good option for memory use since only one extra nMOS transistor is needed for each cell and it uses less power than the standard 6T memory. Adding transistors into the memory cell has an affect on the memory read operation as well. The next chapter will analyze the memory read.

2.8 Summary

Accessing SRAM involves either storing data to the memory cells (writing) or retrieving previously stored data (reading). This focus of this chapter was writing to memory. First, a description of the write process was given. The problems with its operation were then revealed: three short-circuits responsible for significant power losses were identified and causes for delay were mentioned. To remove the short-circuits, the addition of virtual source transistor(s) to the standard 6T differential memory cell was proposed. Delay was intended to be reduced by exchanging the two nMOS access transistors for two CMOS transmission gates. Seven different variations of the 6T memory cell were then designed using different combinations of virtual source transistors and transmission gates. Finally, each design was tested through simulation and

its performance was evaluated. The most successful designs in terms of energy consumption, delay, and the energy-delay product are reviewed below.

- *Lowest Energy Consumption:* In comparison with the 6T memory cell, VGV_N reduces the overall energy used during a write cycle by 27.6%. This confirms that virtual source transistors are effective in eliminating short-circuits that exist while switching the data stored in memory cells.
- *Shortest Delay:* Through the combined efforts of the virtual ground transistor and the CMOS transmission gates, VG_C is able to reduce the delay to store data by 12.3%.
- *Smallest Energy-Delay Product:* For the best combination of power and delay savings, VG_N should be used since it has the smallest energy-delay product, which is 24.1% lower than the standard 6T cell.

For a visual comparison of all seven novel memory cell designs, refer to Tables 2.10 and 2.11 in Section 2.7. These are provided as a reference for the designer who needs to know the tradeoffs of using one of the new cells to improve upon delay or power dissipation.

Chapter 3

Memory Read

When accessing memory for the purpose of recalling stored data, a memory read is being executed. Initially, this chapter will provide a brief review of the process used to read data from a memory block. Some of the deficiencies relating to energy consumption and speed, especially in terms of pipelining, will then be covered. Next, the sequence of attempts to improve the performance of the memory read will be disclosed. Measurement techniques and simulation results for the most successful design follow, and an evaluation and summary of the findings will conclude the chapter.

3.1 Memory Read Process

The SRAM read, like the write access, begins with a row address arriving at the decoder. However, instead of then setting the bit-lines with the proper input data, they are pre-charged to logic '1', a high voltage. Once the designated word-line (W/R) has been selected, each memory latch in a row will begin to discharge the capacitance stored on one of its two bit-lines. For example, if node A is holding logic '0', current will flow from Bit-line into the memory cell through transistors N1 and N4 to ground. (Refer to Fig 3.1 for transistor, node and bit-line

labels.) In order to prevent node A from being overwritten with logic ‘1’, N4 must be sized so that it is stronger than N1. Once a small difference in voltage levels develops on the bit-lines, a read sensing circuit, such as a sense amplifier, may be used to capture and magnify that difference into larger output voltages [5].

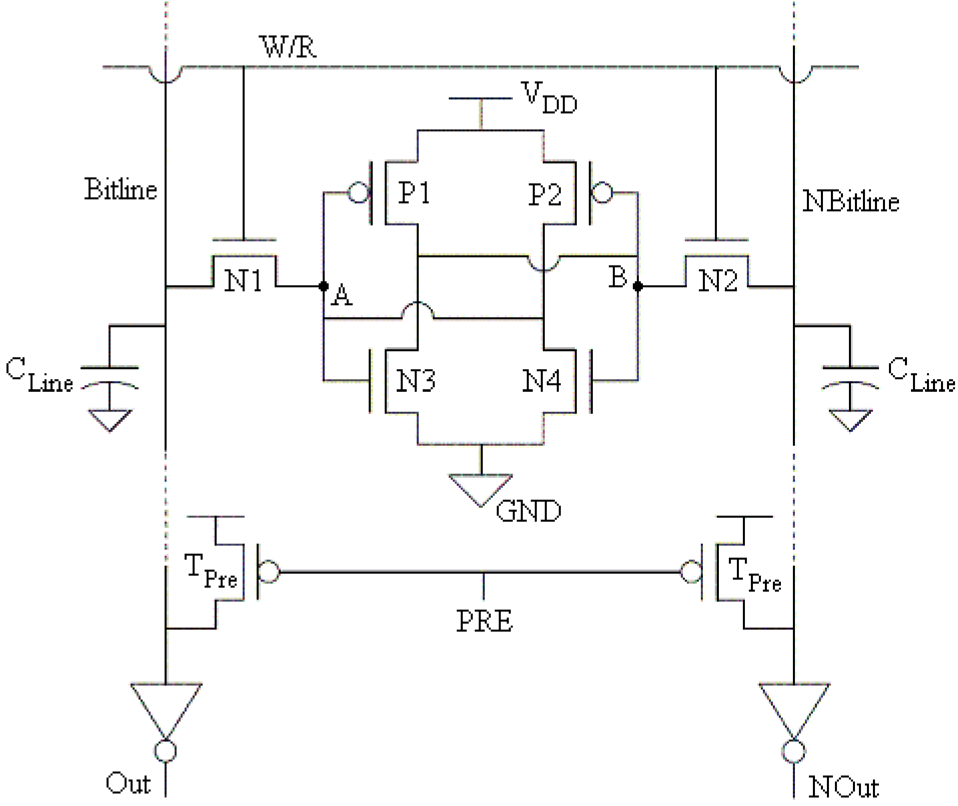

Figure 3.1: Basic memory cell with pre-charge

Reading is a two-step process. It has a pre-charge stage and a bit-line pull-down stage. In comparison with the write access, the timing of the second stage is quite different: instead of only needing the time to switch stored data within a single memory latch, the cell has to be strong enough to pull down an entire bit-line. In large memories, this can amount to very high capacitances because of the number of access transistors attached to the bit-line. For this reason, sense amplifiers are often used [5]. Since this thesis only analyzes smaller memories (32x32

bits), sense amplifiers will not be further discussed, although the output will be sensed by a pair of LO-skew inverters each containing a small load.

3.2 Performance Problems

Just as in the memory write operation, there are aspects of the 6T SRAM design that can be improved, especially in terms of making the memory more suitable for high performance systems. Energy consumption is once again an area of concern, as pre-charging leads to excess power loss. And, in terms of delay, the goal of pipelining brings forth some drawbacks in the current methodology for reading. Each of these issues will be analyzed in this section

3.2.1 Power Dissipation

By far, the largest waste of power occurs in the pre-charge phase. This is because in every read access, both bit-lines are pulled high and then one of them is pulled low. In the case that the bit-lines obtain the same value after reading as they already have before the pre-charge stage, the whole process is unnecessary. Instead, the bit-lines should *hold* their initial values. On the other hand, if a *switch* on the bit-lines is required, then it would make sense to charge the low line to high voltage and pull the high line down to GND. In terms of power usage, it would be much more efficient to only switch the bit-lines if the value to be read out is the opposite of the data presently on the lines. Another, though smaller, source of dynamic power loss is the short circuit that exists between GND of the memory and the bit-line that is being pulled from V_{DD} to GND. This takes place just after the bit-lines have been pre-charged and the access transistors of the latch are on.

Static power losses are very similar to those for the memory write. They result from sub-threshold conduction, gate tunneling, and reverse-biased diode leakage within many of the

transistors in the inactive rows of memory cells. Leakage currents that exist in SRAMs will not be discussed in this thesis.

3.2.2 Delay

In high performance systems, reducing memory access delay is another challenge. If pipelining is to be taken advantage of, the memory access has to be divided up into feasible stages that have similar execution times [6]. The necessity of having a pre-charge stage for a memory read makes this more difficult. The problem is, in a three stage memory pipeline, the second stage has to first pre-charge the bit-lines and then pull one of them down. This takes considerably longer to do than decoding or outputting data, which makes the read the critical stage (see Fig. 3.2). Throughput could increase significantly if the reading portion of the pipeline could be condensed and the stages made more equal in terms of operation time. Section 3.4 will present the methods used to accomplish this.

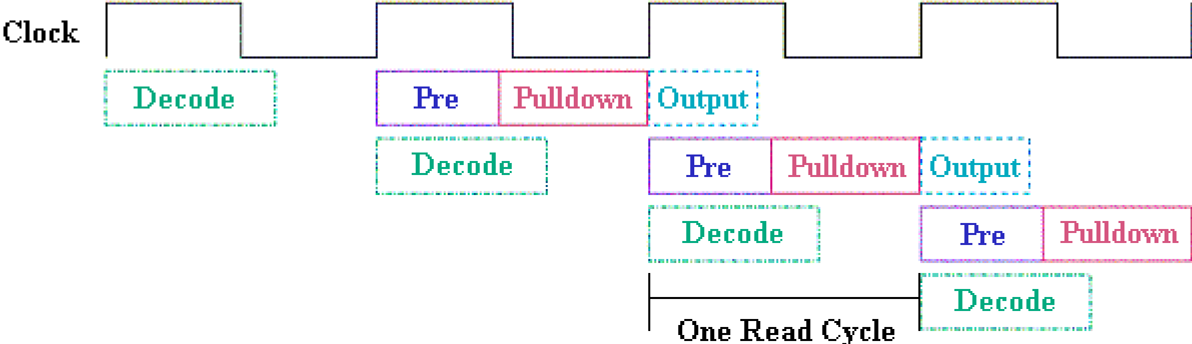

Figure 3.2: Three stage memory access pipeline

3.3 Prospective Bit-line Pull-up Designs

The primary goal we have in the development of a better scheme for reading from memory is to reduce the power consumed. By only switching the bit-lines when their voltage levels differ

from the value stored in memory, it is possible to attain substantial energy savings. The first step in accomplishing this involves removing the pre-charge circuitry and therefore the pre-charge stage in memory access. A memory write is not affected by this because the bit-lines are charged or discharged appropriately depending on the input data at the beginning of each write cycle. Without pre-charge, a method for supplying a strong V_{DD} to the bit-lines must be provided during a memory read access. The next five sub-sections discuss the design sequence followed in creating circuitry that would provide a strong source of V_{DD} to the bit-line transitioning from logic '0' to '1'.

3.3.1 Two Stage Bit-line Capture

The first low-power reading scheme proposed employs two stages: *capture* and *read*. In the capture stage, a latch constructed with minimum-sized cross-coupled inverters is used to store the initial logic level of each bit-line. The second stage uses the captured bit-line voltage levels to determine which line to assist in pulling it up to V_{DD}. The placement of the two stage read logic, along with four different pull-up (PU_A-PU_D) circuits, is shown in Fig. 3.3. PU_A is the first style of pull-up logic that was tested. If the line at logic '0' starts to rise when the W/R word-line is turned on, then V_{DD} is supplied to that bit-line through two series pMOS transistors. On the other hand, if the bit-line at logic '0' does not start to rise, V_{DD} will not be provided to either of the bit-lines—only the memory cell will be providing a weak logic '1'and a strong logic '0' to the bit-lines as they hold their original logic levels. Unfortunately, this manner of reading from memory increased the average power dissipated by 39.2%. Delay increased as well, by about 22%. There are two reasons for these failures. In order for V_{DD} to reach the rising bit-line, it first has to pass through two series pMOS transistors. Secondly, the nMOS transistor (N_X) responsible for providing GND to node X remains on as long as its bit-line is high. This makes it

more difficult for a bit-line at logic '1' to fall to logic '0'. If instead, one of the methods PU_B or PU_C is used along with a different timing for the NS signal, the energy consumed is reduced by at least 29.4%. Delay remains approximately the same as the conventional read method. Even with the large reduction in power consumption, these solutions are still not good options because the new NS timing puts an extra burden on the following memory access. Much of the savings that were attained using either scheme PU_B or PU_C are lost through short circuits between the capturing latch and the pull-up logic or bit-lines.

One common trait of these three pull-up techniques is that they are dependent on the rising bit-line to pass the threshold voltage of N_X before the logic can even begin to deliver V_{DD} to the bit-line. To counter this problem, a crisscrossed scheme can be implemented as depicted in PU_D. For example, if Bit is rising from logic '0' to '1', NBit will turn on the pull-up logic for Bit once NBit's voltage level falls past a certain value. Since a bit-line voltage level falls faster than it rises, this allows V_{DD} to begin flowing to the rising line sooner. While the crisscrossed technique is a good option, this method still is not completely functional. As soon as NS is turned on, charge begins to flow directly from V_{DD} to the bit-line at logic '0'. This then prevents the correct values from being written to the capture latch.

The next attempt at speeding up the memory read delay involves adding pull-down (PD) circuitry to each bit-line in hopes of speeding up the bit-line switch. Illustrated in Fig. 3.4 are several of the designs for pull-down logic, PD_A-PD_C. While the pull-down logic can improve the energy savings by 34.9% and the pull-down delay by 10.2%, the pull-up delay actually increases. The additional pull-down logic increases the capacitance on the bit-lines and introduces a strong source of GND for a rising bit-line to initially contend with. Also, twenty-two transistors are

needed for each column in the SRAM block for the largest design, which is a big overhead for smaller memories.

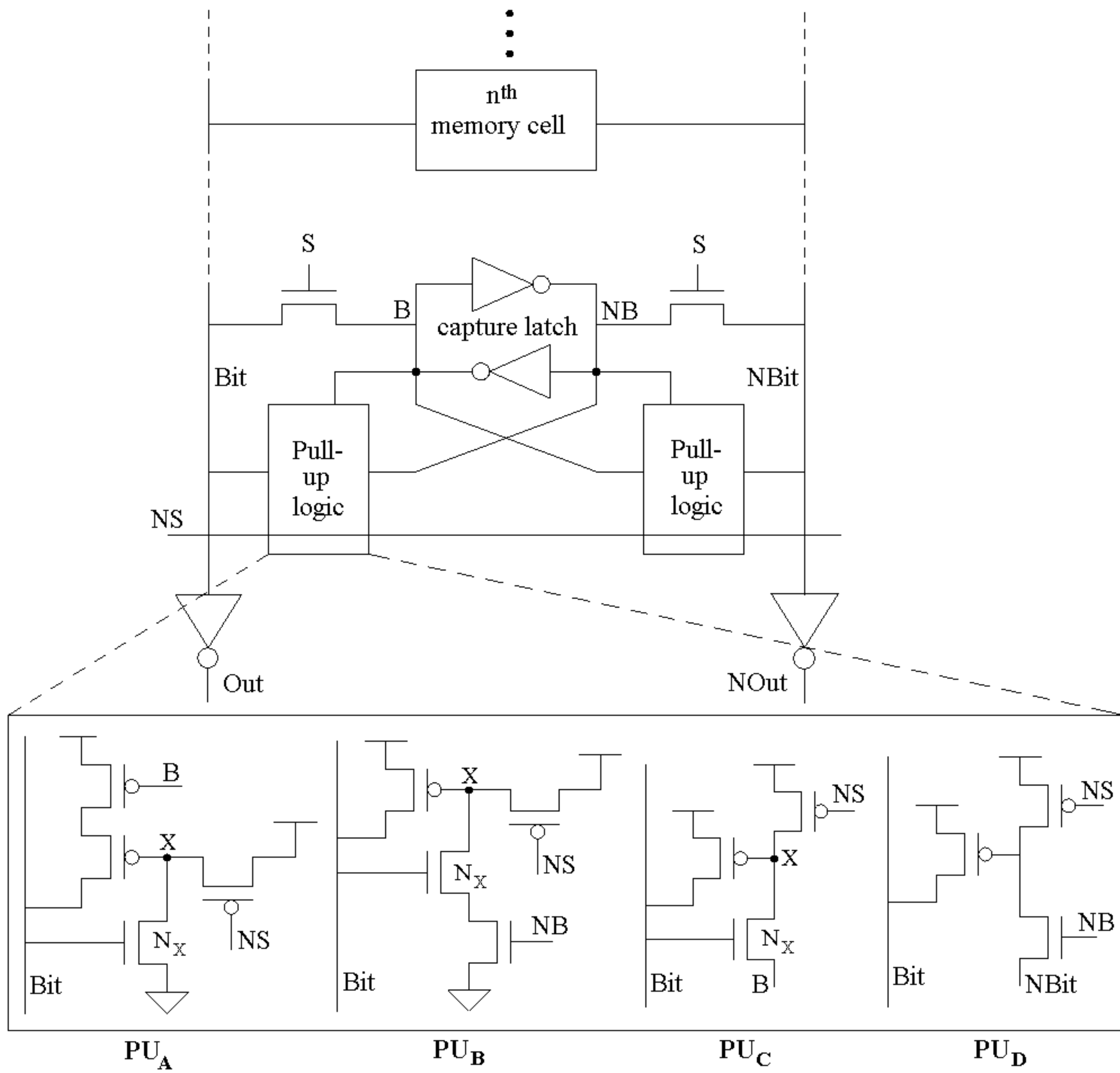

Figure 3.3: SRAM column with two stage read logic and pull-up circuits PU_A-PU_D

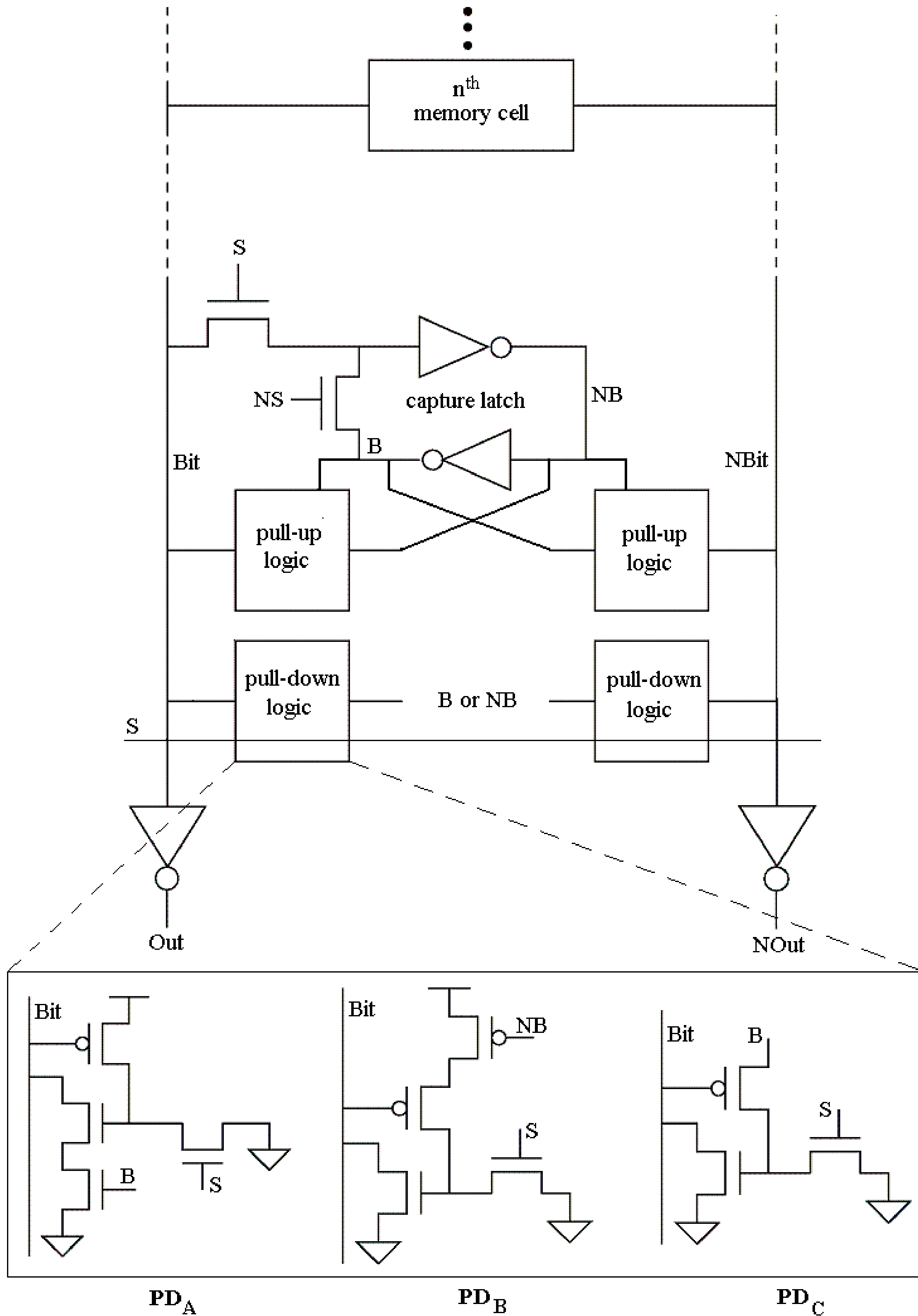

Figure 3.4: SRAM column with two stage read logic and pull-down circuits PD_A-PD_C

Several other problems with this methodology for reading become apparent after more testing. One issue concerns how to deal with bit-lines that are not holding full V_{DD} or GND. If a bit-line has just a high enough voltage level to be stored as a logic ‘1’ in the capture latch, then it cannot be pulled up to a strong V_{DD} even if the memory cell is passing it a logic ‘1’ voltage level. As mentioned earlier, timing control for the signals S and NS is another issue. When a memory read is not taking place, the question of what to do with those signals arises. If the bit-lines are changing due to a memory write or noise, a bit-line could be inappropriately pulled up or down. This is not only a waste of energy; it could also cause the bit-lines to attain the same voltage level, in which case the capture stage would be inaccurate and ineffective. Finally, while some power savings are attained using these schemes, delay only demonstrates an increase in time if the bit-lines are expected to swing their full range. In this way, the noise margin requirements are what dictate the success of these designs in terms of delay. If small noise margins are all that is required, the bit-lines can be allowed to operate at lower voltages and with a smaller swing in order to produce a faster output. Otherwise, speed performance deteriorates rapidly as the bit-lines are obligated to achieve higher voltage levels during pull-up. For all of these reasons, the two-stage bit-line capture will not be discussed any further as a method for decreasing power and delay.

3.3.2 Single Stage Bit-line Capture

This design style focuses on reducing delay by combining the capture and read phases, forcing them to take place simultaneously. This has the advantage of removing the time needed for the bit-line collection from the total reading delay. In order to implement the single stage capture, the latch used in the collection must be unidirectional (as is shown in Fig. 3.4); the feedback capabilities of the latch cannot be operating while it is connected to the bit-line. Since some of

the same pull-up logic is used in these designs as described in Fig. 3.3, the same basic problems with short circuits occur. There is still contention between the different sources to the bit-lines, and in a noisy system, the noise margins must be large enough to prevent faults. If the bit-lines do not swing their full range, the voltage levels captured in the latch during a read may not accurately represent the bit-line voltage levels and the rising bit-line may not receive assistance from the pull-up logic. Also, the control of the S and NS signals remains a complicated issue. Aside from those dilemmas, this method of reading from memory reduces delay and energy consumption considerably: 35% and 13.5% respectively. If system noise is insignificant, speed can be increased by 45.2% by taking the output from the two sensing inverters as soon as they switch. However, because of the numerous difficulties with implementation of this reading technique, the next three design styles will move away from the latched capture.

3.3.3 Equalized Bit-line Read

If the bit-lines can be operated with a small swing, a scheme called *equalized bit-line read* based off of the work in [7] can be used. This system uses a wide pMOS transistor called P_{EQ} to connect the two bit-lines. When P_{EQ} is on, charge is shared between the two lines. Then, as W/R is turned on, the memory cell begins to charge or discharge the bit-line capacitances. This technique has some wasted power if the bit-lines are equalized only to return to their original state, but, if the swing is small, this wasted power is small in comparison with the overall energy savings. The circuitry presented in Fig. 3.5 shows one method of implementing the equalized read. After the memory cell has created a small differential on the bit-lines, the S and NS signals can be turned on. Then the source-controlled inverters push the bit-lines in their appropriate directions.

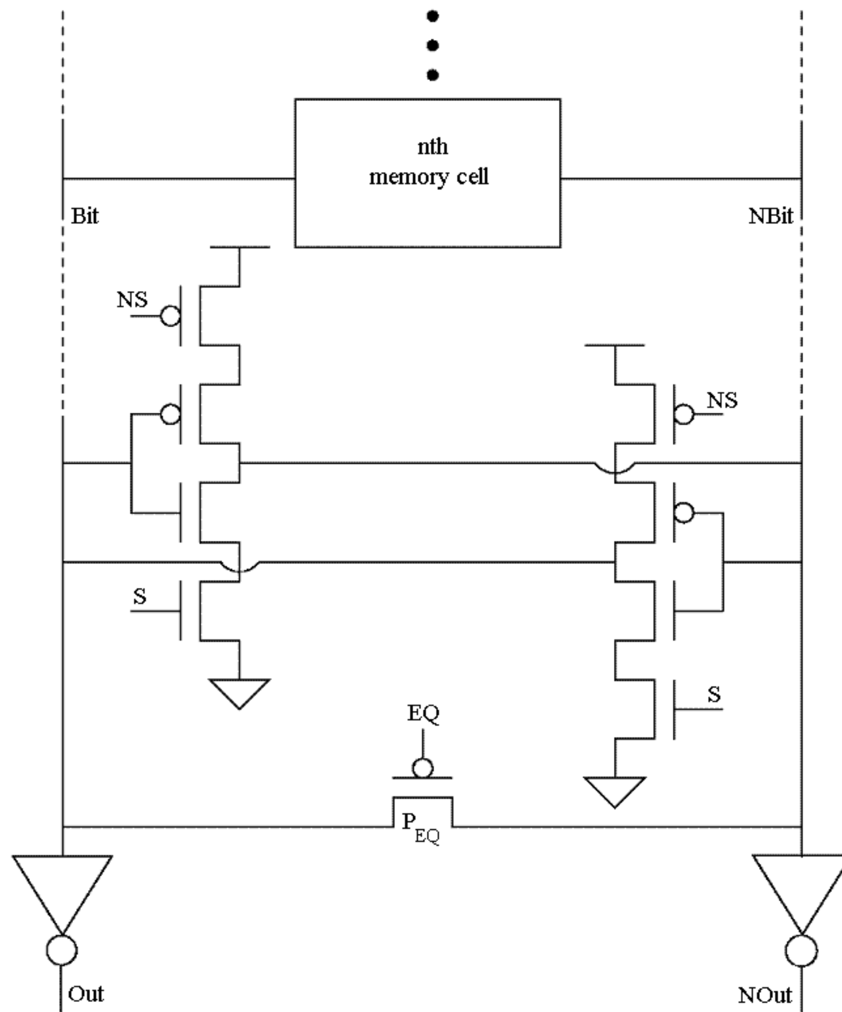

Figure 3.5: Equalizing bit-line read circuit

In the worst case, this design consumes 26.6% less energy than the standard read method, but with only a 4.9% increase in speed. If instead the bit-lines were operated with a small swing (assuming noise is not a factor), power savings could increase significantly. This design is a big improvement over the previous methods because the S and NS signal timings are greatly simplified and short circuits are no longer a big problem. Another benefit is that when a bit-line is transitioning, it does not have to fight with its own pull-up or pull-down logic. Also, the number of transistors used to implement the logic is only nine as compared to a range of twelve

to twenty-two for the previous techniques (although all five of the pMOS transistors are much larger than minimum width).

One of the major drawbacks to using this reading method is that it has a large overhead in terms of power, since it has three different signals to drive. Each of these signals requires a special driving circuit to ensure that it is properly timed. In fact, the extra overhead for the equalizing transistor in terms of power and speed is large enough that using EQ cannot be justified. If P_{EQ} is extra-wide to speed up the read, the delay (and power needed) to turn on P_{EQ} increases and cancels out the savings in speed. By keeping P_{EQ} small, power and delay to turn on signal EQ is minimized, but then P_{EQ} is of little or no help because a memory cell could begin to equalize the bit-lines almost as fast during a read switch. That is why the use of P_{EQ} and the EQ signal will not be analyzed any further in this thesis. The next two techniques to be discussed are simpler in terms of driving circuitry and they demonstrate better savings both in terms of power and delay.

3.3.4 Delayed Bit-line Capture

Capturing the initial value of the bit-lines is not very effective if they do not have a very large differential between them. The delayed bit-line capture pull-up (DBCP) reading technique depicted in Fig 3.6 focuses on where the bit-lines will transition to. Once W/R turns on, either a bit-line switch or hold will be forced by the memory cell. The sample signal, S, can be turned on after enough time has passed for the bit-lines to transition past their switching point. When S is a logic ‘1’, current begins flowing through the N_C transistors. The two P_C transistors then push the bit-line with the higher voltage to V_{DD} so that its P_V transistor is turned off. Then the opposite P_V transistor begins supplying a strong source of V_{DD} to the rising bit-line. If the bit-lines are not switching, then once the sample signal turns on, the bit-line holding a logic ’1’ will be provided

with V_{DD} while the bit-line retaining '0' will be held low by the memory cell. This circuitry resolves the problem with noise margins since it supplies V_{DD} to the high bit-line even when it is holding. While S is logic '0', the P_S transistors are on and preventing the two P_V transistors from passing V_{DD} to either bit-line. Only one signal is needed to control the read circuitry and its line has much less capacitance than the pre-charge line. One downside of this design is that eight transistors are needed. Although six of them are minimum sized, the two P_V transistors are five times the minimum width. This design yields an average of 45.9% power savings over the conventional read and a around a 22.7% reduction in delay. Sections 3.5 and 3.6 will continue the discussion of this reading scheme as it is a very successful solution.

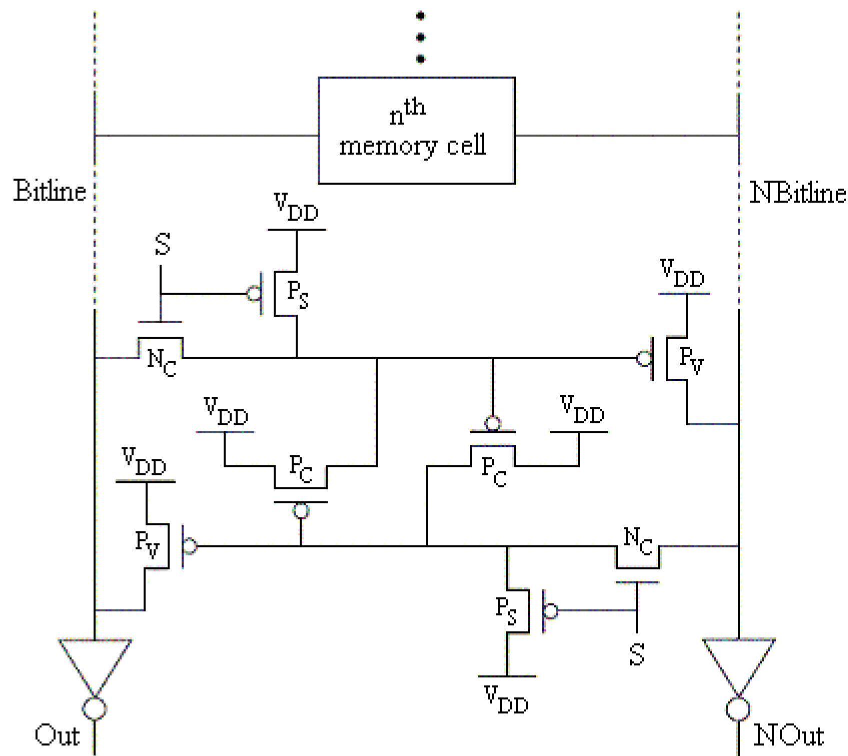

Figure 3.6: Delayed bit-line capture pull-up circuitry

3.3.5 Cross-Coupled Bit-line Pull-up

In Section 3.3.3, two source-controlled inverters were used to boost the speed of the read bit-line switch. Those inverters incorporated both a pull-up and a pull-down. Earlier work showed how the pull-down actually only creates a more complicated system and does not assist much in speeding up the read access. By removing the pull-down from the source controlled inverters, not only is the number of transistors reduced to four, but only one signal, the delayed read (DR), can be used to control the flow of the V_{DD} source to the bit-lines. The resulting circuit shown in Fig 3.7a is very simple: it is dependent on the falling bit-line to turn on the T_B transistor that has its drain attached to the rising bit-line. Once the DR signal falls, the T_V transistors and the correct T_B transistor are all on, and V_{DD} can begin to flow to the rising bit-line. The timing of DR is important: it should not turn on the T_V transistors until the bit-lines have reached their switching point. For pMOS transistors sized at five times the minimum width, this circuit demonstrates 33.1% savings in energy and a small 5.9% improvement in delay. This read logic outperforms the source-controlled inverter method in all areas: size, simplicity, energy consumption, and delay.

If the metric to improve is the delay, the second circuit shown in Fig 3.7b is an option since V_{DD} can begin to flow to the rising bit-line as soon as DR turns on. When the T_V transistors are sized at 25λ and the T_B transistors are at 20λ, this method yields a 16.5% reduction in delay and still uses 22.7% less power than the conventional read.

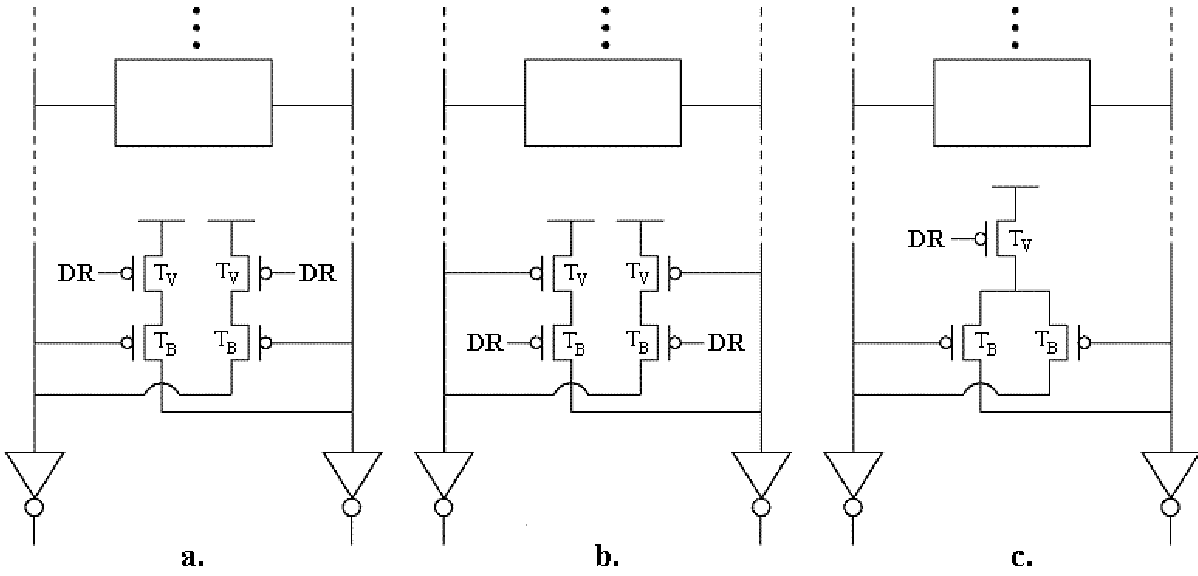

Figure 3.7: Cross-Coupled Bit-line Pull-up Circuits

The final option, another rendition of the cross-coupled bit-line pull-up (CCBP) read logic is illustrated in Fig 3.7c. This scheme removes one of the T_V transistors completely and each remaining transistor has a width of 15λ. The benefits of doing this are revealed in the fact that this method uses 42.9% less power than the standard read. Since DR only has to drive one transistor per column instead of two, its line capacitance and driving power decrease by nearly half—the degree of this improvement is dependent on the transistor widths being driven. Delay, like with the first cross-coupled reading technique, is not improved much, only a 5.7% reduction occurs.

It should be expected that none of these CCBP designs would improve delay significantly because they all have two series transistors between the source of V_{DD} and the bit-lines. The power savings for the three-transistor pull-up circuit, however, are comparable to those that result from the use of the DBCP circuitry. Therefore, if a low-power SRAM is being designed that requires a smaller size over a decrease in delay, the three-transistor CCBP would be the best

choice. When all factors are considered, neither of the other two CCBP schemes perform as well as the DBCP logic; for that reason, only the three-transistor method will be subjected to additional study. From now on, when the acronym CCBP is used, it will refer only to the three-transistor style of reading logic. Other issues to consider when determining which pull-up scheme to implement will be discussed in the results and analysis sections, especially noise margins and the required bit-line voltage levels for different SRAMs as the $CCBP_{WEAK}$ circuitry is well suited for such systems. $CCBP_{WEAK}$ is a 3λ (weak pull-up) implementation of the CCBP novel read style.

3.4 Measurement Techniques

In the last section, a number of different designs were mentioned as possible solutions to the problems of delay and power dissipation that each occur during a memory read operation. The remainder of this chapter narrows the selection down to the two schemes which were very successful in one of more of the following areas: low-power, high-speed (taking noise margin requirements into account), and small in size or simple in design. These designs will be more thoroughly tested and analyzed using the measurement techniques and proper signal timings that are described in this section.

For the purpose of comparison, each of the circuits proposed in Section 3.3 is constructed using the Cadence Virtuoso Schematic Editor at 180-nm CMOS technology. Each schematic consists of one memory cell, its two bit-lines, a word-line, LO-skewed inverters for sensing the output, and the appropriate bit-line and word-line capacitances. The standard 6T SRAM includes pre-charge transistors like those shown in Fig. 3.1, whereas each of the read circuits include their own pull-up logic. Two sets of tests are run on each system using the SpectreS simulator: one to determine the proper timing for its signals and therefore overall read delay,

and the other to measure energy dissipation. Simulations run with the goal of determining signal timing and delay include large inverter drivers for each signal in the system, such as EQ or S, since they take into account the capacitance for the entire row of memory cells. When energy consumption is the test subject, the inverter drivers are not included; instead correct rise and fall times are approximated and only the power used in the cell and its word-line is measured.

The timing of the sampling signals, S, NS, and DR is also very important. This is because it has a big effect on the power dissipation and performance of both the CCBP and DBCP implementations. During a bit-line switch, if the any of those signals arrive too soon the bit-line transitions will be slowed and excess power consumed (due to one of the pull-up transistors fighting to hold its bit-line at a logic '1'). If the sampling signals transition too late, operation will be slowed because of the lack of a strong source of V_{DD}. When the timing is correct, savings in terms of delay speed-up are acquired since no pre-charge stage exists. And, with the cycle time reduction of the reading stage of the memory access pipeline comes both a reduced response time and an increased throughput.

As was the case in the memory simulations in Chapter Two, the bit-lines and nodes of the memory latches are initialized with either logic '1' or '0' to reproduce the conditions that would exist if a bit-line switch or a bit-line hold was to occur. Normal operation for a memory read would only require that a memory cell pull one of the bit-lines down. However, if a bit-line switch is taking place, the memory cell is then confronted with fully charged bit-lines carrying the opposite voltages of the memory's internal nodes. If the sizes of the pMOS transistors are not increased, the memory cell will be overwritten; so, in this chapter, P1 and P2 are twice the width of the standard 6T memory pMOS transistors and they are at minimum length instead of 3λ.

Energy consumption is measured in much the same way as in the previous chapter: the total energy dissipated is determined by monitoring the instantaneous voltage and current supplied to the system and then evaluating the integral over entire read cycle. Separating the power needed to drive the read access from that dissipated on the bit-lines and in the read logic is necessary for the most accurate comparison of power saving between the standard read and the novel read designs. The methods used for doing that will be explained in the analysis section of this chapter. Since the $CCBP_{WEAK}$ operates with small noise margins, extra power analysis must be performed on its system to ensure that it will operate correctly when used in noisy systems. This will also be talked about in Section 3.6.

The delay for reading from each system is measured from the time the read access begins (in many systems that is when the W/R signal begins to be driven high) until the latter of the two bit-lines reaches 90% of its destination. In the case of the $CCBP_{WEAK}$, the delay is between the 50% points of the transitioning W/R and Out signals. The conventional reading method adds the pre-charge delay to this sum as well: just for the power comparison with the $CCBP_{WEAK}$ system, the conventional 6T read also has its delay recorded between the 50% of V_{DD} points of its PRE and Out signals.

3.5 Results

This section presents the results of the simulations run on a standard SRAM with pre-charge and on two of the novel reading schemes: the DBCP and CCBP. Thorough simulations of the other designs discussed in Section 3.3 were not taken, so no further data on those schemes is available past what is presented in that section.

3.5.1 Conventional Reading Results

The data provided in Table 3.1 shows the delay and energy dissipation of the conventional read method with pre-charge. Since driving an entire row of pre-charge transistors uses a large amount of energy, simulations were taken with and without the pre-charge driver. Energy dissipation is divided into two columns: the first column shows the energy consumed during a normal read operation, and the second column shows the energy used just to drive the word-line. Delay is given in two forms as well. The first column gives the time to switch the output through the HI-skew output inverters. The delay for the bit-line to achieve 90% of its final voltage is also shown.

Table 3.1: Standard read energy and delay data

pMOS Width	Pre-charge Driver	Energy (fJ)		Delay (ps)	
		Total	Word-line	50% Out	90% Bit-line
30λ	Included	1374	300.5	1301	1441
	Not Included	511.8	300.5	1311	1460
60λ	Included	1571	300.9	1143	1283
	Not Included	513.7	300.9	1145	1284

The pre-charge driver plays a significant role in determining the speed of the standard read. Table 3.1 contains the results from a read column with a pMOS driver at 30λ and at 60λ. The first memory is designed for low-power and the second is designed for less delay. The simulation in Figure 3.7 was run on a schematic with a pMOS driver at 60λ, which improves the 30λ pre-charge pull-up by almost 160 ps. The tradeoff is power consumption. Since this is the general trend for all signal drivers in the read circuitries tested, the remaining simulation results presented in this section reflect driver circuits that balance power-consumption and delay.

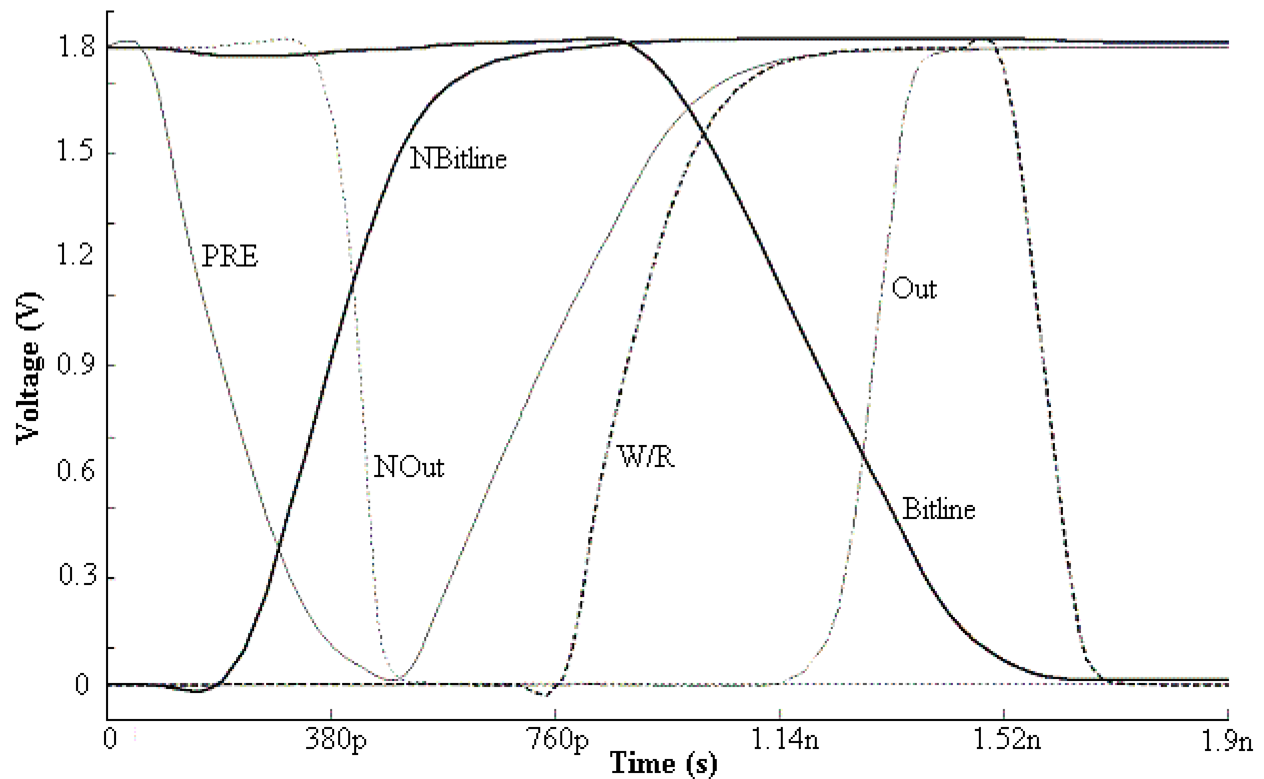

Figure 3.8: Standard read with 60λ pre-charge pMOS drivers

3.5.2 Delayed Bit-line Capture Results

In Table 3.2, the results for a DBCP read are displayed. In this table, energy is given in terms of the amount used to switch the bit-lines from logic ‘0’ to logic ‘1’ and visa versa, as well as the energy needed to hold the bit-lines at their current voltage levels. The simulation in Fig. 3.9 was taken with the sample signal (S) driver included and it demonstrates how the S signal should be delayed more than 200 ps after the W/R signal because this yields the best NBitline pull-up time.

Table 3.2: DBCP read energy and delay data

S Driver	Energy (fJ)		Delay (ps)	
	Bit-line Switch	Bit-line Hold	50% Out	90% Bit-line
Included	834.2	631.5	680.5	991.7
Not Included	521.0	318.5	679.7	978.6

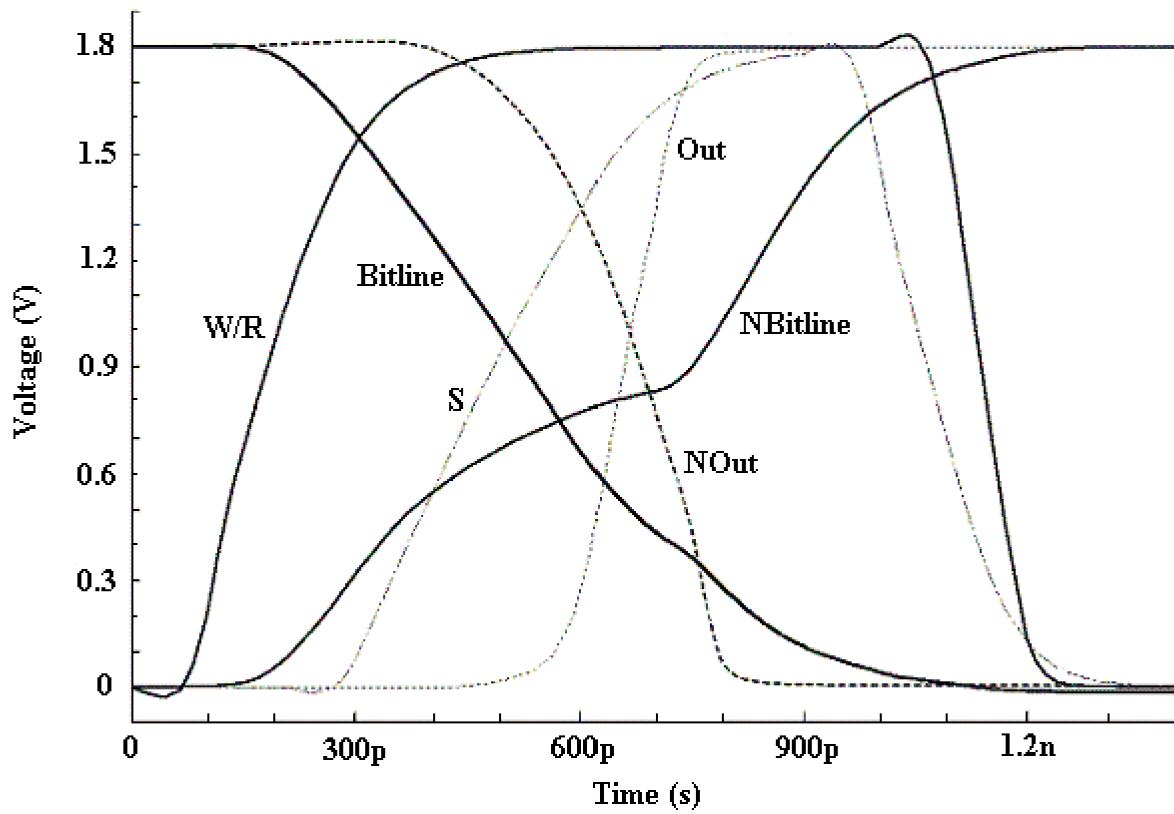

Figure 3.9: DBCP read switch simulation

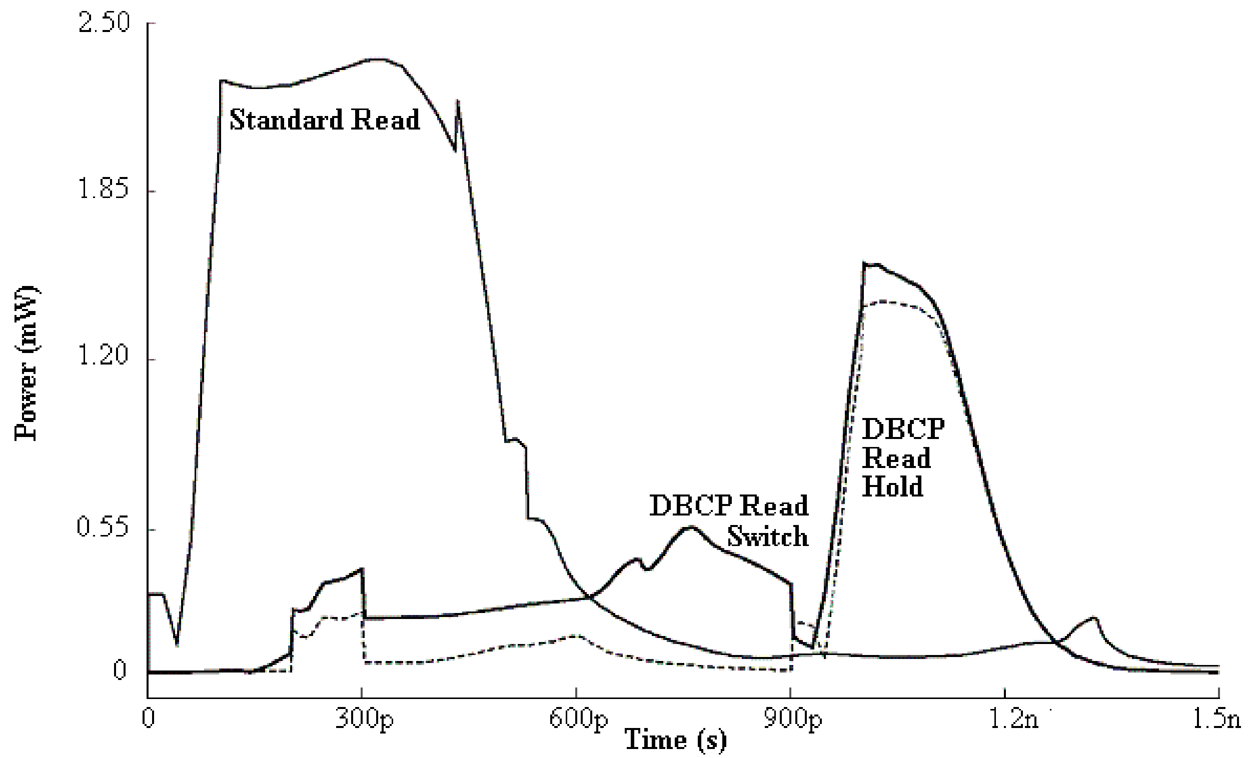

Figure 3.10: Standard and DBCP power comparison with the signal driver power included

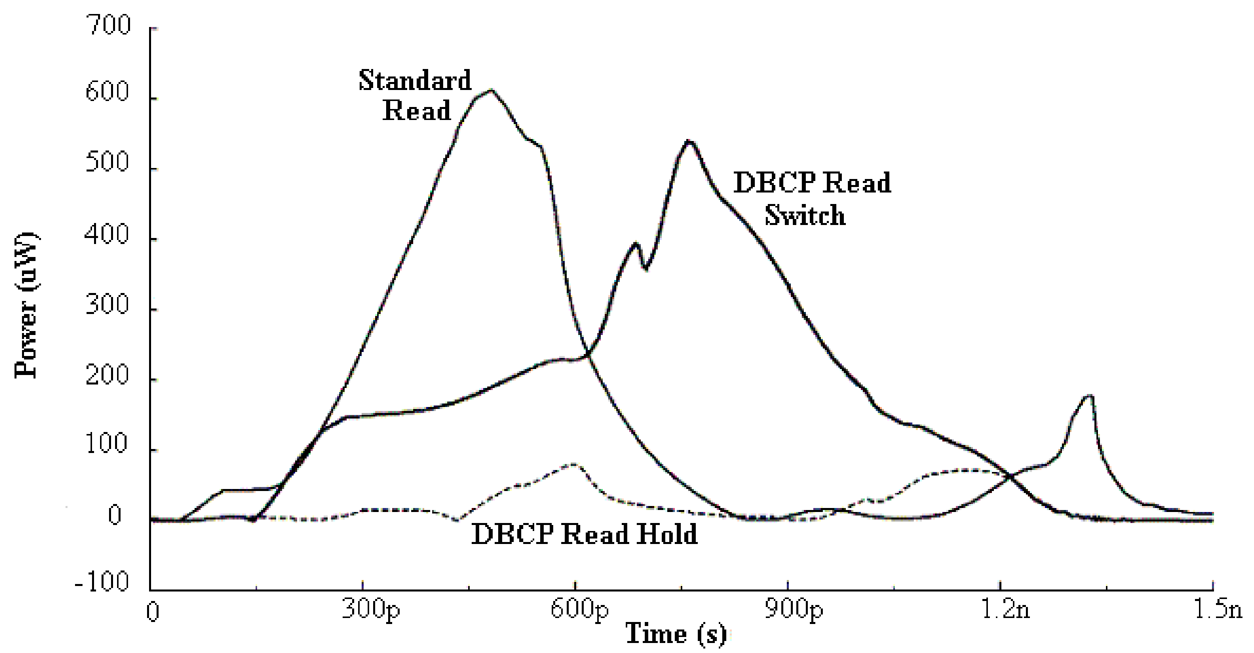

Figure 3.11: Standard and DBCP power comparison without the signal driver power

The simulations in Fig. 3.10 and 3.11 compare the power dissipated during one read cycle. Fig 3.10 includes the power to drive pre-charge and S for an entire row of 32 memory cells, whereas Fig. 3.11 only gives the energy for reading from one memory cell in a single column. In the analysis section, the total savings for the DBCP scheme will be calculated using the energy results from both simulations.

3.5.3 Cross-Coupled Bit-line Pull-up Results

The following results are taken from simulations of the CCBP reading scheme. Table 3.3 shows how this reading style performs when expected to function in systems that require large noise margins. The data is divided into two groups based off of the size of the pMOS transistors used in the pull-up circuitry. The 15λ CCBP circuit is designed to be another option for replacing the pre-charged read logic. The 3λ, or $CCBP_{WEAK}$, design is better suited for systems that only require small noise margins. The remainder of the results in this sub-section will focus on the $CCBP_{WEAK}$ performance when the bit-lines are allowed more freedom regarding their initial voltage level.

Table 3.3: CCBP (15λ) and $CCBP_{WEAK}$ (3λ) read energy and delay data

pMOS Widths	DR Driver	Energy (fJ)		Delay (ps)	
		Bit-line Switch	Bit-line Hold	50% Out	90% Bit-line
15λ	Included	924.2	676.8	757.9	1208.5
	Not Included	542.6	308.4	755.3	1205.6
3λ	Included	664.2	457.3	684.0	1624.8
	Not Included	502.6	303.0	676.5	--

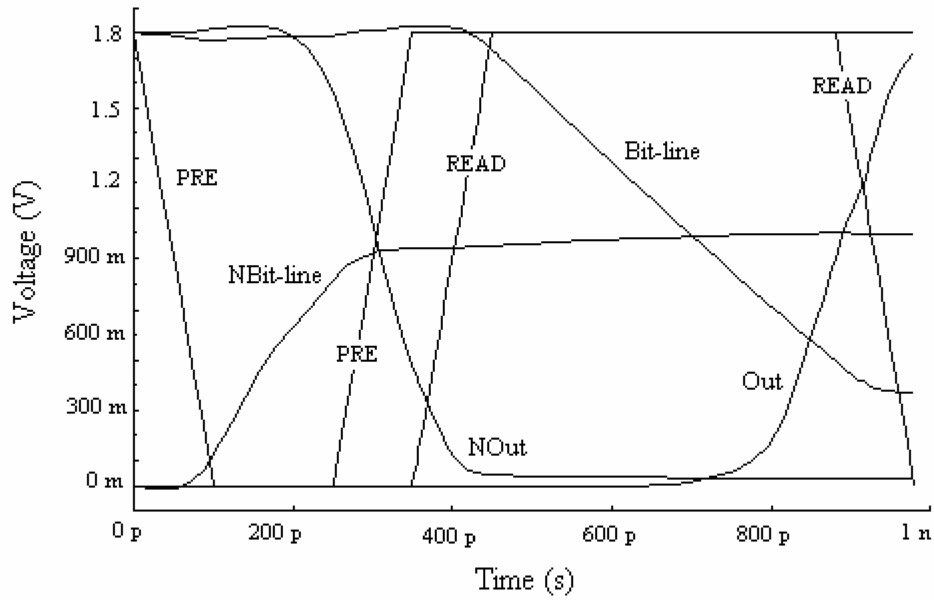

Figure 3.12: Simulation of a standard read memory access causing a bit-line switch

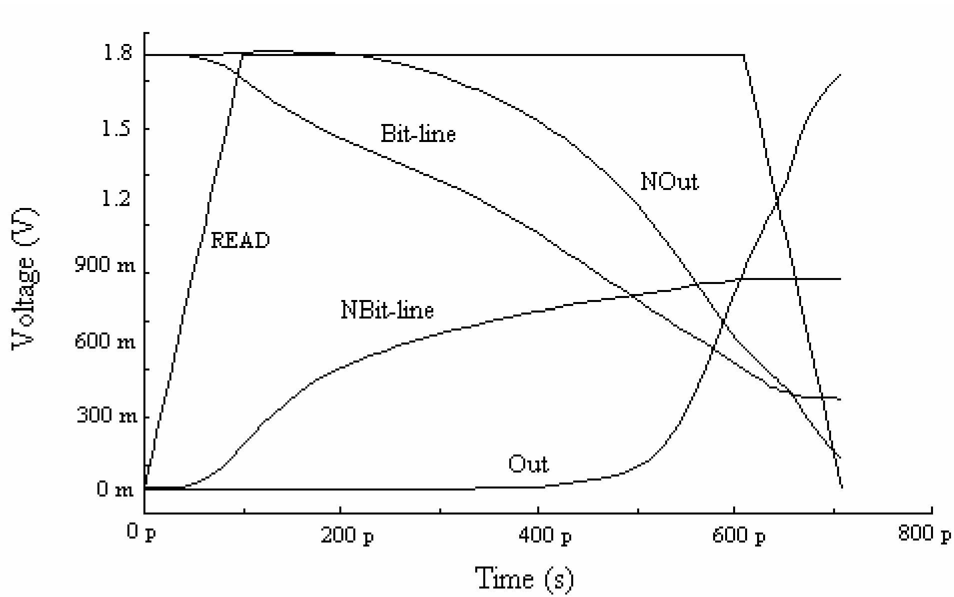

Figure 3.13: Simulation of a $CCBP_{WEAK}$ read memory access causing a bit-line switch

When smaller noise margins are allowed, the bit-lines do not have to swing their full range to generate the correct output. If it is only necessary that they swing half of their range,

pre-charge and the $CCBP_{WEAK}$ only need to pull the low voltage bit-line up to 900 mV. The simulations in Figures 3.12 and 3.13 illustrate how each reading scheme functions when operated under those conditions. It should be noted that for these simulations, the READ and PRE signals are given set rise and fall times; signal drivers are not included in order to keep the power dissipation analysis simple. Also, these graphs give the worst case delay for each system since the bit-lines are at their full voltage levels and have further to swing to give the correct output.

The measured delays for both the conventional and $CCBP_{WEAK}$ read are shown in Table 3.4 and 3.5, respectively. The values in Table 3.4 are the summation of the pre-charge and pull-down delays. Since only one bit-line is pulled down in a standard read for both a switch and a hold, Table 3.4 only gives the delay for one bit-line's initial value. Table 3.5 shows the delay of the $CCBP_{WEAK}$ read switches for any combination of bit-line voltages ranging from 0 to 400 mV and 0.9 to 1.8 V. These ranges were selected due to the nature of both the standard and $CCBP_{WEAK}$ read circuits. In the time allowed for a bit-line to switch or hold, a falling bit-line will always achieve at least 400 mV and a rising bit-line will always reach at least 900 mV. The pre-charge stage does not last long enough in the standard read for both bit-lines to reach their full voltage, which causes energy and delay to vary based on the initial bit-line voltages. A table for $CCBP_{WEAK}$ read holding delay is not included because in that case, the delay is negligible since neither bit-line is switching its value.

Table 3.4: Standard read delay (ps)

NBit-line (V)	Bitline Initial Voltages (V)									
	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
0	673	690	706	720	733	744	755	764	772	780

Table 3.5: CCBP$_{WEAK}$ read switching delay (ps)

		Bit-line Initial Voltages (V)									
		0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
NBit-line (V)	0	458	464	470	476	481	487	491	496	523	559
	0.1	440	445	449	454	459	463	467	471	475	508
	0.2	422	427	431	436	440	444	448	451	464	497
	0.3	402	406	411	415	419	423	426	430	457	490
	0.4	378	382	386	390	394	398	401	419	452	485

Table 3.6: Standard read switching energy (fJ)

		Bit-line Initial Voltages (V)									
		0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
NBit-line (V)	0	500	483	466	447	428	407	385	363	339	315
	0.1	485	469	452	433	414	393	371	349	325	301
	0.2	475	459	441	423	403	382	361	338	315	290
	0.3	467	451	434	415	396	375	353	331	307	283
	0.4	462	445	428	409	390	369	348	325	302	277

Table 3.7: CCBP$_{WEAK}$ read switching energy (fJ)

		Bit-line Initial Voltages (V)									
		0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
NBit-line (V)	0	295	295	296	296	295	294	292	291	289	286
	0.1	272	272	272	271	270	269	268	266	264	261
	0.2	253	253	252	252	251	250	248	247	245	242
	0.3	234	234	234	233	232	231	230	228	227	224
	0.4	216	216	216	215	214	213	212	211	209	207

Table 3.8: Standard read holding energy (fJ)

		Bit-line Initial Voltages (V)									
		0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
NBit-line (V)	0	469	452	435	416	396	376	354	331	308	284
	0.1	466	450	432	413	394	373	351	329	305	281
	0.2	463	447	429	411	391	370	349	326	303	279
	0.3	460	444	427	408	388	368	346	323	300	276
	0.4	457	441	423	405	385	364	343	320	297	272

Table 3.9: $CCBP_{WEAK}$ read holding energy (fJ)

		Bit-line Initial Voltages (V)									
		0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
NBit-line (V)	0	80	61	50	43	37	31	25	19	12	4
	0.1	80	61	50	43	37	31	25	19	12	5
	0.2	80	61	50	43	37	32	26	19	12	5
	0.3	80	61	50	43	37	32	26	19	12	5
	0.4	80	61	50	43	37	32	26	20	13	5

The data presented in Tables 3.6–3.9 are measurements of the energy consumption in femtojoules (fJ) for the given range of initial voltages on the bit-lines. Tables 3.6 and 3.7 give the energy for switching reads and Tables 3.8 and 3.9 show the energy consumed while holding the bit-line values. A power consumption comparison between the standard and $CCBP_{WEAK}$ schemes is depicted in Figure 3.14. This graph does not include any power used to drive the input signals.

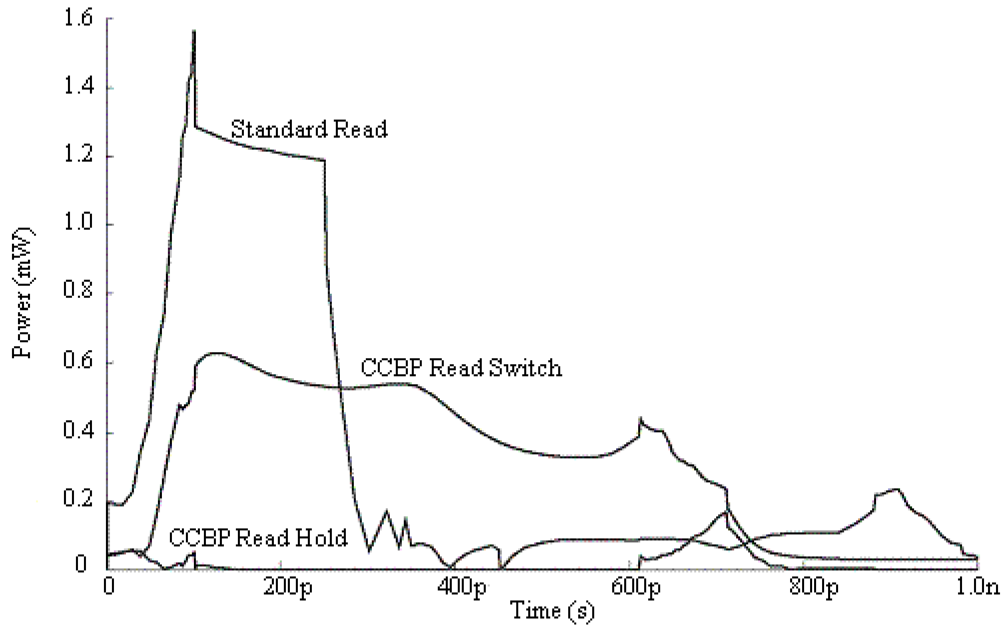

Figure 3.14: Comparison of standard and $CCBP_{WEAK}$ read instantaneous power

3.6 Analysis

An accurate comparison of the different memory read techniques is key to the selection of the best suited style for the targeted memory application. By knowing the power used to drive the read signals and the energy dissipated on the bit-lines and in the read circuitry, the total energy consumed during one read cycle for one 32 bit row of memory can be calculated. These energy totals can then be compared to yield the percentage savings each method obtains in comparison to the conventional reading scheme. To determine the power needed just to drive the pre-charge signal, the total energy for simulation without the driver included is subtracted from the energy total that includes the driver signal. Using the 60λ data in Table 3.1, this yields 1571 - 513.7 = 1057.3 fJ. The power to drive the entire row's word-line is then added to that total 1057.3 + 300.9 = 1358.2 fJ. Next, the energy used in one column is obtained by subtracting the word-line power from the energy total not including the pre-charge signal driver 513.7 – 300.9 = 212.8 fJ.

Since the 32x32 bit memory has 32 columns, the single column total is multiplied by 32: 212.8 * 32 = 6809.6 fJ and then it is added to the pre-charge and word-line driver total 6809.6 + 1358.2 = 8167.8 fJ. This same routine is used to evaluate the total row energy consumed for the 30λ data. Analysis of the DCBP and CCBP reading techniques is a little more complicated since both bit-line switches and holds need to be accounted for. If the assumption is made that 50% of memory reads are holds and 50% are switches, then the values presented in Table 3.10 reflect the adjusted computations for each of the memory read schemes shown.

Table 3.10: Comparison of features for each reading technique

Design	Delay (ps)	Row Energy (fJ)	Read Hardware		Noise Tolerant
			Capacitance	Transistors	
Standard 60λ	1283	8167.8	Very Large	2	Yes
Standard 30λ	1441	7924.3	Large	2	Yes
DBCP	991.7	4417.2	Small	8	Yes
CCBP	1209	4663.1	Small	3	Yes
$CCBP_{WEAK}$	684.0	<3719.7	Very small	3	No

Once the row energy has been determined for each read method, the percentage comparisons that were given in Section 3.3 can be calculated. For the DBCP design, energy savings range from 44.3% to 45.9% and delay improvements range between 22.7% and 31.2% when compared with both the 30λ and 60λ conventional read energy and delay totals. Improvements in speed for the CCBP read vary between 5.8% and 16.1% and energy savings range from 41.2% to 42.9%. In terms of both energy and delay, the DBCP method performs better, although it does require eight transistors to implement it whereas the CCBP scheme only needs three.

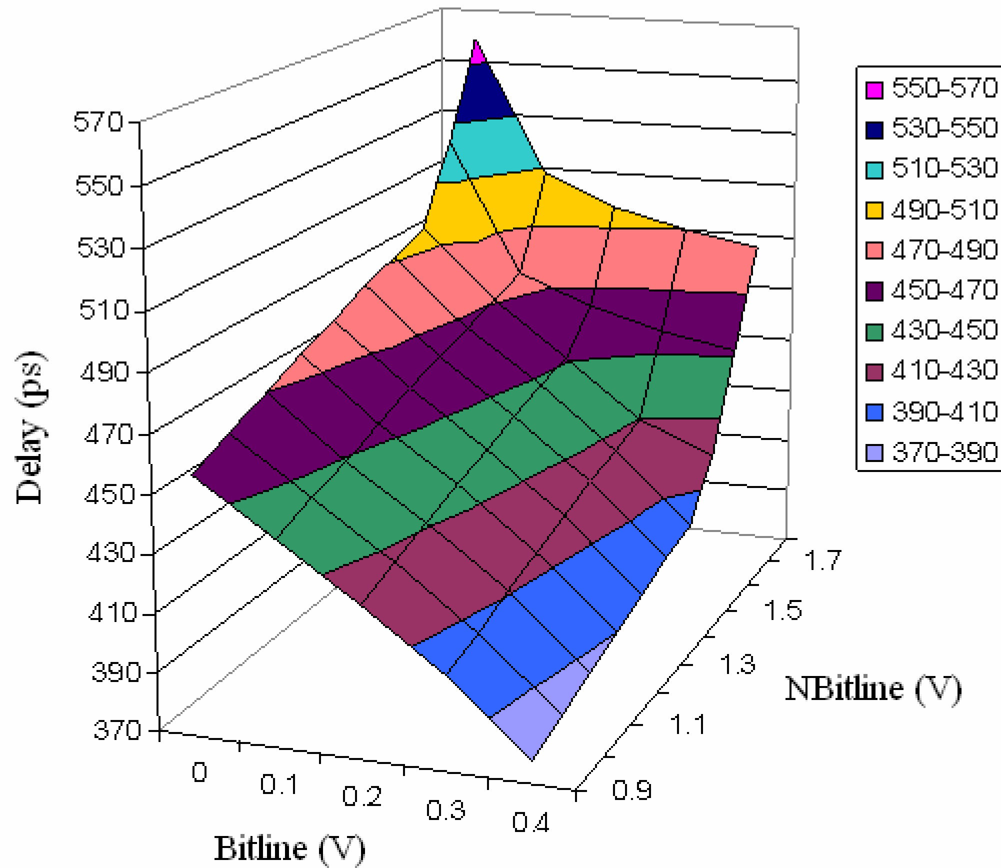

Figure 3.15: Surface plot of $CCBP_{WEAK}$ read data *(data from Table 3.5)*

The best use of the CCBP circuitry is in systems with little noise or that only require small noise margins. For these memories, the $CCBP_{WEAK}$ logic can be used to greatly decrease the delay and power of a read access. Two different sets of delay comparisons can be made between the standard read and the $CCBP_{WEAK}$ logic. The first uses the data in Tables 3.1 and 3.3. Since the output is captured as soon as the second transition at Out or NOut takes place, the delay for $CCBP_{WEAK}$ is 684.0 ps. The standard read has delays of 1301 ps and 1143 ps for the 30λ and 60λ reads respectively to generate the correct 50% voltage level on both Out and NOut. Therefore, the $CCBP_{WEAK}$ read style yields 40.2% to 47.4% savings in delay. However, that

comparison cannot be completely justified since the standard read is expected to pre-charge its bit-lines to full V_{DD} each read.

The second delay comparison can be made using Tables 3.4 and 3.5. This method implies that delay savings for the $CCBP_{WEAK}$ method are instead only 28.3%. Fig. 3.15 shows the decrease in delay that occurs over the varied initial bit-line logic levels. These values reflect the data gathered from the simpler reading simulations that do not include signal driving time. When the driving time is incorporated, the delay for both the standard read and the $CCBP_{WEAK}$ increase, but since the increase is much larger for the standard read, the savings in delay are actually larger than those implied by the data in Tables 3.4 and 3.5, and instead range somewhere between 28.3% and the 47.4% improvement calculated above.

This is especially beneficial in the area of high performance computing where pipelining of memory accesses is practiced. Assuming that the read stage of a memory access is the determining factor for the length of the pipeline cycle time, this cycle time could be reduced to three-fifths of its original length by using the $CCBP_{WEAK}$ read method. Another point to notice in Fig 3.15 is that as the bit-line voltages approach each other, the delay decreases significantly. If the bit-lines were certain to never reach their full voltages, then it would be safe to reduce the read access and pipeline cycle time even further resulting in greater savings.

Energy consumption is the other area in which the $CCBP_{WEAK}$ reading scheme shows vast improvements. It is partly reduced because the DR signal driver is smaller than pre-charge and it does not have to charge or discharge as much capacitance on its line during each memory read as the pre-charge signal. If the switching energy for each initial bit-line voltage is compared between the two reading methods, the smallest ratio occurs when one bit-line is at full V_{DD} while the other is at ground. In that worse case scenario, the result is a 9.2% reduction in energy for

the $CCBP_{WEAK}$ read. For the more likely case, where one bit-line is at 1.0 V while the other is at 0 V, savings of 38.9% would arise. During a bit-line hold, even bigger savings can be realized. When one of the bit-lines is already at 0 V, even if the second line is at 900 mV, 82.9% of the energy can be saved by using the proposed $CCBP_{WEAK}$ read method. And, when the second line is at full V_{DD} instead of 900 mV, a 98.6% reduction in energy consumption will result.

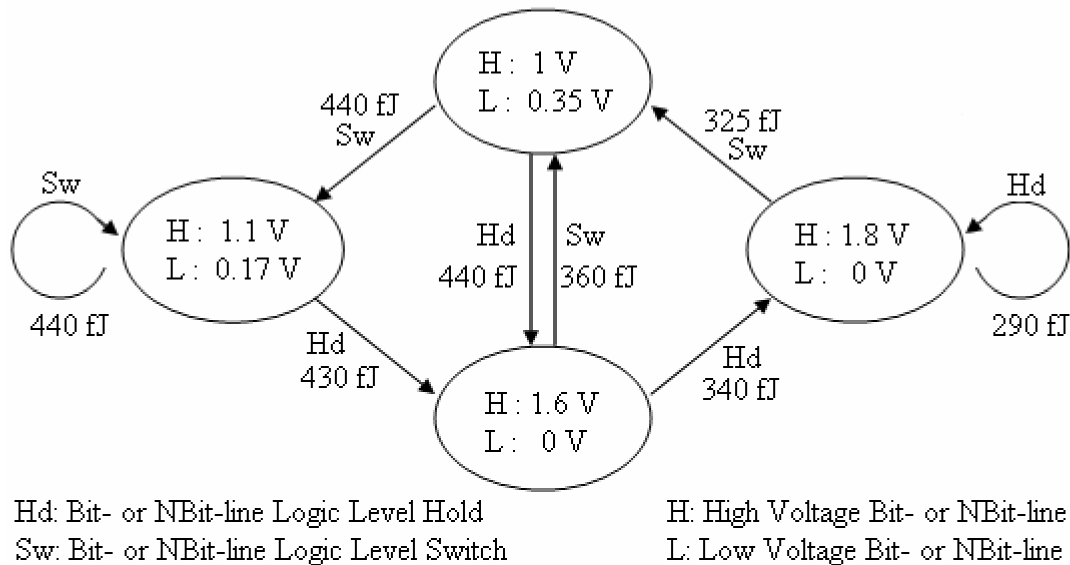

Figure 3.16: Energy used for different initial Bit- and NBit-line voltages during standard bit-line switches and holds

In order to best compare the two methods while taking both bit-line holds and switches into account, a state diagram has been derived for the standard read. Fig. 3.16 shows the four states that the bit-line voltages will usually fall within over a series of reads. If the voltages do not fall within one of these states, after several cycles they will eventually find their way among them and remain there as long as the high capacitances on the bit-lines prevent the bit-line voltages from changing much in between read accesses. The two values within each oval represent each bit-line voltage (± 70 mV) before the standard read takes place. An arrow labeled

Hd signifies a bit-line hold and the label *Sw* represents a switch. The numbers next to the Sw or Hd for each arrow give the approximate energy used (± 15 fJ) for that operation. The diagram explains how energy consumption is quite large for both standard read holds and switches, whereas for the $CCBP_{WEAK}$ reading scheme, every time a bit-line holds its value, it is expending at least 63.0% less energy than if it were to switch its values.

3.7 Summary

This chapter focused on the second aspect of SRAM access: the memory read. To begin with, the procedure for reading from a memory cell was given. The pre-charge stage was identified as a problem with the read operation because it consumes significant power and increases the overall delay, which is especially detrimental to pipelined memory access. Five different styles of read logic to replace the pre-charge circuitry were discussed along with performance comparisons between each scheme and the standard reading technique. The final two schemes: the Delayed Bit-line Capture Pull-up (DBCP) and Cross-Coupled Bit-line Pull-up (CCBP) demonstrated improvements in delay and notable savings in terms of power dissipation. These two design styles were then subjected to further analysis including a small noise margin study on the $CCBP_{WEAK}$ design. Following is a summary of the characteristics and simulation results for each of the improved reading techniques.

- The novel DBCP method uses eight transistors (two of which have widths of 10λ, the other six are minimum sized) and one delayed sampling signal to assist a memory cell in switching the bit-line voltage levels. Once the read access starts, the memory cell can either cause the bit-lines to switch or hold their voltage levels. After enough time has passed for the bit-lines to reach their switching point, the sampling signal is

activated and the bit-line capture occurs in the read logic. The read logic then causes a rapid voltage transition on the bit-lines to their final levels. Energy consumption for this reading scheme ranges between 44.3% to 45.9%, while delay improvements vary from 22.7% and 31.2% when compared with both the 30λ and 60λ implementations of the conventional read.

- The CCBP reading scheme has three transistors, each 10λ wide. When a read access begins and the bit-lines transition past their switching point, the delayed read signal (DR) activates, turning on the read logic. Two of the pMOS transistors then help to quickly pull the rising bit-line to V_{DD}. Improvements in energy consumption are between 41.2% and 42.9%, and speed is increased by 5.8% to 16.1% over the 30λ and 60λ standard reading circuits.
- $CCBP_{WEAK}$ is the 3λ implementation of the CCBP read logic. It is designed for operation in systems only requiring small noise margins. Bit-line swing is nearly halved and therefore, significant savings in power and delay result. Depending on the initial bit-line voltage level and whether the bit-lines are switching or holding, energy reduction ranges between 9.2% and 98.6%. Delay is decreased by at least 28.3% and at most 47.4%.

Table 3.10 shows the tradeoffs of each novel design as well as the conventional reading method. It is important to remember that these measurements are only taken for memory reads, no power savings are provided during memory write accesses. The proposed reading techniques are not the best solution for every memory system. They will be most appropriately applied to memories which experience frequent read accesses, especially reads where the bit-lines would hold their initial voltage levels.

Chapter 4

Read/Write Combination

Static Random Access Memories are designed to be both read from and written to so it is incomplete to only analyze the cells from a single direction. Each operation has its own requirements for fast or low-power performance. This means that when the memory is expected to both read and write, there are certain tradeoffs that have to be examined. The goal becomes finding a combination of improvements that achieve the desired performance for the specific SRAM being designed. The question is not which implementation is the best, but which is the most appropriate for the given situation.

This chapter first explores the functionality of the SRAM: the aspects that are necessary to successfully read and write and the issues in performance that exist as a result of the read/write combination. Several SRAM implementations using the improvements from the last chapter are built and each is simulated to see how the improvements affect the memory write operation and how the overall performance compares with the other SRAM designs. A brief discussion on the differences in performance between 90-nm and 180-nm CMOS technology is also included.

4.1 Combined Memory Access Description

The design requirements for a memory cell that reads and writes quickly while using little power are somewhat different than for a latch that focuses on one operation over the other. This is because to a certain extent, the goals of the two operations compete. Desired transistor sizes disagree and bit-line pre-charge is unnecessary for writing. Power needs to be continually supplied to the bit-lines during a write access, but when reading, a single memory cell drives both bit-lines to reflect the stored value in the cell. Creating a functional memory is a consequence of finding a balance of these opposing values that allows reading and writing to occur in an efficient manner.

The objective when writing to memory is to first set the bit-lines to the correct voltage levels and then to propagate those voltage levels into the selected latch, overwriting the values stored on the cross-coupled inverters. The smaller those inverters are, the more easily they can be overwritten. Wider access transistors also speed up the time needed to overwrite the inverters, but they increase bit-line capacitance so that switching the bit-lines will then take longer and consume more power.

Reading is more demanding in two areas. It requires more time to have control over the bit-lines since they first need to be pre-charged and then must take on the stored voltage levels of the selected memory cell. And, the cross-coupled inverters must be larger so they can more strongly drive the bit-lines. It is also important to ensure that memory will never be overwritten during a read. Keeping bit-line capacitance small as well as having good size ratios between the access transistors and the n- and p-type transistors will help with that.

The majority of the conflict between reading and writing is in the bit-line conditioning circuitry and the memory cell transistor sizes. Finding a good compromise begins with

determining memory cell transistor sizes that allow for functional reads and writes. For the standard read method with pre-charge, it is important that the P1 and P2 transistors are weaker than the N1 and N2 transistors. This ensures that the memory cell can be overwritten. Typical sizes for the pMOS transistors are 3λ wide and a length of 3λ, while the access transistors are 4λ wide and 2λ long. To guarantee readability, the N3 and N4 transistors should be stronger than the access transistors; so, if N1 and N2 have a width to length λ ratio of 4/2, the N3 and N4 should have a λ ratio of 8/2 [5].

Timing and overall cycle time is also very different between the two operations. Delay in particular is a major issue for high performance systems, so pipelining is recommended. In a pipelined system with a decode stage, a read or write stage, and an output stage, the read and write operations are given the same amount of time to complete their task. But, the standard method of reading takes more time than the write operation. In a system where there are significantly more write accesses than reads, much time is wasted during each cycle, especially since the read/write cycle is the time limiting stage in comparison with the decode and output stages.

The purpose of this chapter is to compare the performance of the standard 6T SRAM with several SRAMs which are built using the different enhancements proposed in the previous two chapters. This includes the removal of pre-charge from the bit-line conditioning circuitry, which is instead replaced with either the DBCP or CCBP read logic. Without pre-charge, the requirements on the memory cell change for the memory read, and those new transistor sizes have an effect on the write operation. Using the techniques described in the next section, each of the SRAMs will be measured for power consumption and delay to determine if the previously proposed designs actually demonstrate improvements over the standard SRAM design.

4.2 Measurement Techniques

This section explains the methods used to measure and compare the standard 6T SRAM design for the combined operations of read and write with the DBCP and CCBP SRAMs. Schematics are created for each SRAM implementation using the Virtuoso Schematic Editor at 90-nm CMOS technology. The standard SRAM schematic includes the following features: a single memory cell with the size ratios given in Section 4.2, bit-lines and a word-line with capacitances equivalent to those which would exist in a 32x32 bit memory, bit-line conditioning circuitry, appropriately sized inverters to drive the control signals, and output sensing inverters. Bit-line conditioning circuitry consists of the logic needed to set the bit-lines with input data for writing or to pre-charge the lines for reading. For the standard SRAM, three input signals must be provided: WE (which allows the bit-lines to be set with input data during a write), W/R (which controls the word-line), and PRE (which pre-charges the bit-lines with V_{DD}). The schematics constructed for the DBCP and CCBP SRAMs are very different from the standard SRAM in that they have different sized memory cells and neither of them have pre-charge logic (or the PRE signal). Instead, they implement their own style of read logic that is driven by either the S or DR signal.

To test each of the SRAMs, simulations are created and run using the Spectre simulator. A standard read operation starts with both lines pre-charged—set with initial conditions to 1 V. When the W/R signal rises, one of the bit-lines is pulled low. Then pre-charge turns on and the bit-lines are pulled back up to V_{DD}. The standard write operation first activates WE to set the bit-lines with the input data, and then turns on the word-line so that the memory cell can be overwritten with the new data. Both simulations are built for the fastest operation. That means that as soon as the line has been pulled to GND for a read, the pre-charge circuitry turns on and

begins to restore the bit-lines to full V_{DD}. Writing is also set up so that as soon as the bit-lines reach their correct input values, the word-line turns on. This is the same for the simulations of the DBCP and CCBP memories, although they have different timing requirements for their reading schemes.

Read timing for the DBCP and CCBP SRAMs are quite different from the standard reading method. A read begins by pulling W/R high so that the memory cell can begin to switch the bit-lines. Once the bit-lines have reached their switching point, the S or DR signals can be activated appropriately so that the read logic turns on and helps pull the bit-lines to their final destination. Depending on the goal of the design, the timing of the S and DR signals can be modified to achieve more or less power or delay. The simulations in this thesis are balanced between those two tradeoffs so that neither metric is favored. As for the DBCP and CCBP write timings, they are very similar to the standard write, the only difference being the overwriting delay and for the minimum sized memories, the bit-line writing time. These variations arise because the DBCP and CCBP memory latch sizes are not the same as in the standard 6T SRAM.

If either the DBCP or CCBP methods for reading were implemented with the standard memory cell transistor sizes, the latch would be overwritten. Therefore, it is necessary to adjust the memory cell sizes so that when the bit-lines are at the opposite voltage levels of the memory cell, they will not switch the cell's level, but instead, the cell can begin to switch the bit-lines themselves. This can be done in two ways: the inverters in the latch can be strengthened so that the p-type transistors have λ ratios of 8/2; and the second method is to adjust the access transistors so that they are minimum sized. Both of these size schemes were implemented and tested, and the next section gives the results that the simulations generated.

The measurement of delay and power for each SRAM design is done in much the same way as in the previous two chapters. Delay is measured from the time the memory access starts until the bit-line (for a read) or memory latch node (for a write) reaches ninety percent of its final voltage level. Power consumption data is collected once again by monitoring the current through the supplies of V_{DD} or to GND. To determine how much power is used by each active row of memory and each column, the power supplied to each of these areas is measured separately. The energy dissipated through the memory cell or bit-lines to GND is also monitored so that power consumption can be compared between the different memory implementations. The next two sections will discuss more about the different energy measurements taken and what they imply about the SRAM design being tested.

4.3 Results

This section presents all the data attained from the series of simulations run on each type of SRAM. For the standard memory, only three simulations were necessary: one for a read, one for a write where the bit-lines are switched by the write circuitry, and one for a write where the bit-lines hold their current voltage levels. The standard read has no switch or hold since the bit-lines are always pre-charged after one of the lines has been discharged. Table 4.1 displays all of the results from the three tests.

During each simulation, power consumption is measured on each row and column. Row energy consists of the power supplied to each word-line driver to charge the word-line as well as the driver energy for WE and PRE since their signals are supplied to the entire row of bit-line conditioning circuitry. Column energy includes all the power supplied to pre-charge or write to the bit-lines. Dissipated energy is measured in terms of the energy that is pulled to GND through the memory cell and that which is grounded through the writing circuitry. This energy is part of

the column energy in terms of total energy consumption and is only measured separately so that the differences between the three memory styles might be seen more clearly. The last column in the table is the delay. The delay for a write hold is always the same as the write switch since the cell is given the same amount of time to set the bit-lines whether they need to be switched or not.

Table 4.1: Standard SRAM access energy and delay

Access Type	Supplied Energy (fJ)		Dissipated Energy (fJ)		Delay (ps)
	Row	Column	Memory	Bit-lines	
Read	80.2	14.2	12.0	0.4	474.3
Write Switch	75.1	19.5	1.0	15.0	306.6
Write Hold	73.6	7.6	1.0	3.2	--

The results in the Table 4.2 are gathered from eight different simulations run on two implementations of the DCBP memory design. The first four tests were run on the DBCP 4λ implementation, named so for the widths of its access transistors. As mentioned in the previous section, this method has extra-wide cross-coupled inverters within its latch. The next four simulations were run on the second implementation, which is called DBCP min because it has access transistors with the minimum width of 1.2 μm. The purpose in testing two different memory cell sizes with each design is to ensure that the performance of the SRAM is improved because of the modified reading style and not simply because the memory cell size and bit-line capacitances have been altered.

Since pre-charge is not included in the DBCP design, a different simulation needs to be run for a read where the bit-lines are switched and for a read where the bit-lines hold. As with the standard SRAM, the supplied energy is measured for an entire row of memory. However, for the DBCP scheme, this includes the power supplied to the S driver and its signal line instead of

the PRE signal. The remaining energy measurements are conducted in the same way as with the standard SRAM design.

Table 4.2: DBCP SRAM access energy and delay

Access Type		Supplied Energy (fJ)		Dissipated Energy (fJ)		Delay (ps)
		Row	Column	Memory	Bit-lines	
4λ	Read Switch	41.6	13.8	14.0	0.4	315.3
	Read Hold	39.3	2.1	1.9	0	--
	Write Switch	74.8	23.0	2.5	16.5	278.5
	Write Hold	73.1	10.5	2.5	4.7	--
min	Read Switch	36.3	10.6	10.5	0.4	290.9
	Read Hold	34.2	2.1	1.7	0.1	--
	Write Switch	70.8	18.3	1.9	12.8	272.3
	Write Hold	69.1	8.0	1.9	4.0	--

Delay is only recorded for the read and write switch operations. This is because a read hold delay is negligible since it already has the correct output and a write hold yields the same delay as a write switch for the same reason mentioned for the standard write hold.

Shown in Table 4.3 are the results for the CCBP style SRAM. Just as with the DBCP SRAM, the two different memory cell sizes are tested for this design. All of the energy and delay measurements are taken in the same way as for the DBCP design with the exception of the row energy since the CCBP SRAM drives the DR signal instead of the S signal. These results and those for the DBCP and standard memories will be studied in depth and compared in the next section.

Table 4.3: CCBP SRAM access energy and delay

Access Type		Supplied Energy (fJ)		Dissipated Energy (fJ)		Delay (ps)
		Row	Column	Memory	Bit-lines	
4λ	Read Switch	41.4	15.1	14.5	0.4	379.8
	Read Hold	39.1	1.3	0.4	0	--
	Write Switch	74.8	24.6	2.5	18.2	303.0
	Write Hold	73.1	10.6	2.5	4.8	--
min	Read Switch	36.0	11.5	11.0	0.4	333.7
	Read Hold	33.7	1.1	0.3	0	--
	Write Switch	70.8	19.8	1.9	14.0	271.8
	Write Hold	69.1	9.2	1.9	4.0	--

4.4 Analysis

In this section, a thorough analysis of the causes and implications of the results presented in Section 4.3 will be given. A comparison of the energy consumption and delay for the different SRAM designs will also be discussed. In Fig. 4.1, a comparison of the power supplied to one row of each of the five different SRAM designs is shown. The bar graph makes it clear that driving a pre-charge signal for an entire row of memory is a very expensive operation in terms of energy usage. This can be seen by the fact that the standard read operation uses about twice the energy to drive its signals as the novel SRAMs do to drive their sampling signals: S and DR. Because the different designs all use the same writing circuitry and signal drivers, the energy consumption measurements for the memory write are all very similar.

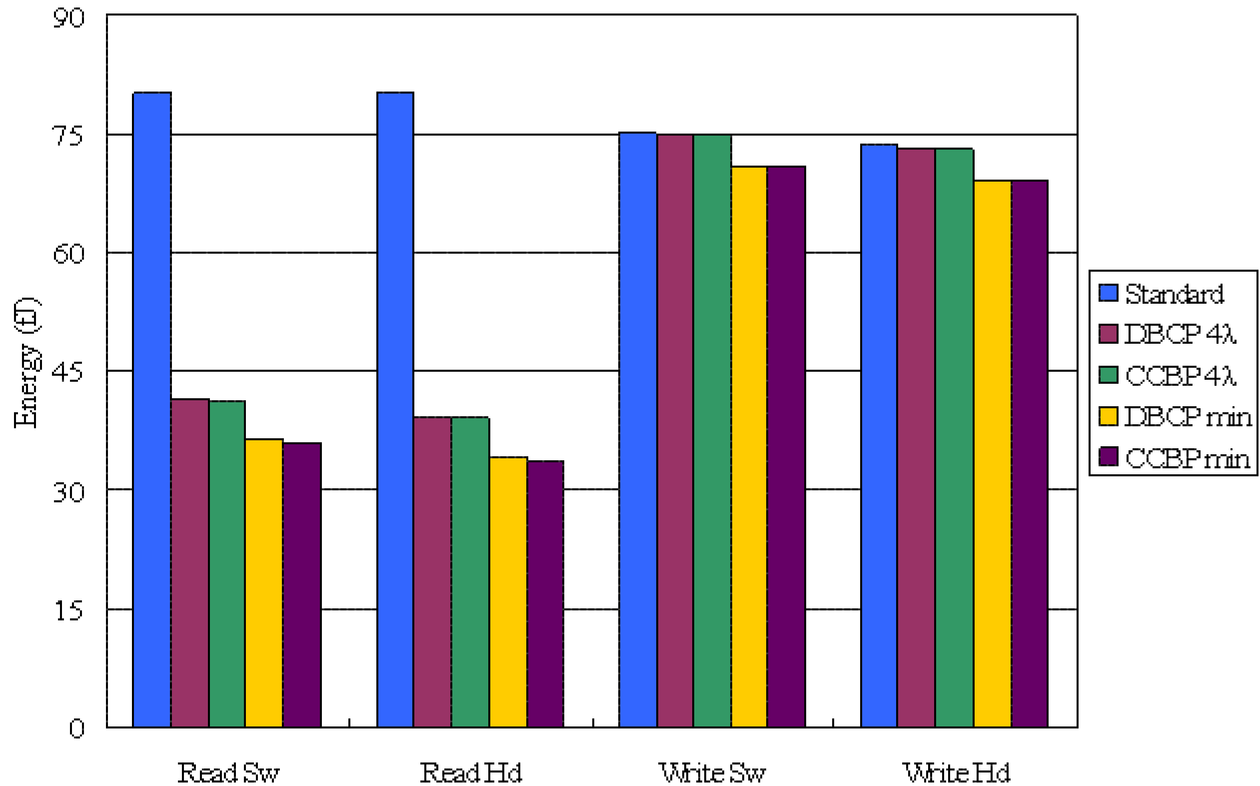

Figure 4.1: Comparison of supplied row energy

Power is also supplied to each column during a memory read or write. The bar graph shown in Fig. 4.2 reveals how much energy is supplied to a column of SRAM for each type of memory access. The first concept to notice is that reading from memory is much more efficient for read holds when any of the new SRAM designs are used than for when the standard SRAM and pre-charge are used. This is because the bit-lines can simply retain their values instead of charging and discharging one of the lines. Typically the two minimum sized SRAM implementations use less power than the 4λ designs since they have less bit-line capacitance and less delay. However, for the read hold, the two DBCP memories are supplied more power. This can be explained by the fact that at the end of the read, the DBCP logic has to restore its internal nodes to V_{DD} to prevent the pull-up transistors from supplying logic '1' values to the bit-lines.

Both the write switch and hold require a significant source of power. The added read circuitry and larger memory latch inverters for the novel designs allow for short-circuits when a memory is being overwritten. Even when the bit-lines are holding their voltage levels a significant amount of energy is used because of the short-circuits between the bit-lines and the memory cell supplies.

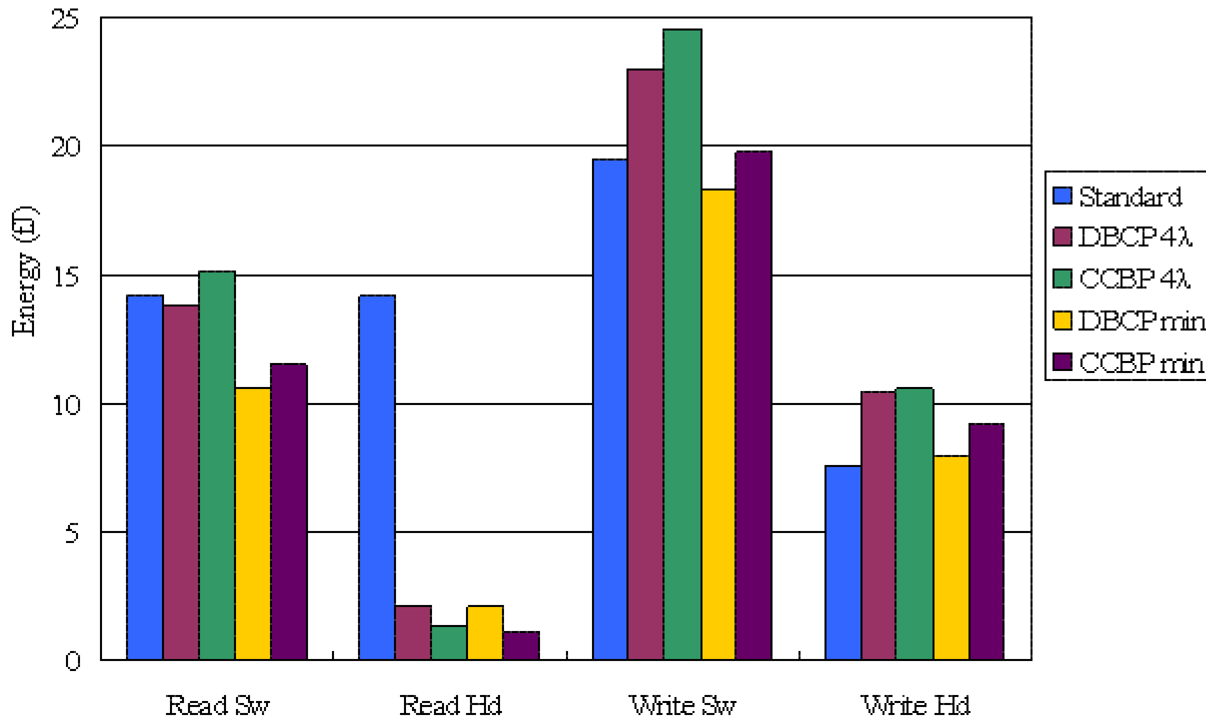

Figure 4.2: Comparison of supplied column energy

Using the column and row energy totals for each type of memory access, the overall power consumption can be calculated for each SRAM implementation. This is accomplished by multiplying the column power by 32 and adding that to the energy needed to drive a row and the total row energy results. For the sake of comparison, we assume that each type of access happens with the same frequency of each of the other access types. Under that assumption, the total energy consumed for each type of memory design can be calculated to yield the results

shown in Table 4.4. When these values are used to compute the overall energy savings of each SRAM, we find that each novel design yields the following savings when compared to the standard implementation: 11.1% for DBCP 4λ, 8.3% for CCBP 4λ, 28.6% for DBCP min and 24.0% for CCBP min. Because of the great amount of power consumed during the write operations, the novel SRAMs do not demonstrate as large of savings overall as they did for the memory read access alone. It is therefore important to consider what kind of memory accesses will be occurring most commonly prior to using one of the novel SRAM designs for a new memory.

Table 4.4: Total energy supplied to each SRAM (fJ) and average energy savings

	Standard	DBCP 4λ	CCBP 4λ	DBCP min	CCBP min
Read Sw	534.6	483.2	524.6	375.5	404
Read Hd	534.6	106.5	80.7	101.4	68.9
Write Sw	699.1	810.8	862	656.4	704.4
Write Hd	316.8	409.1	412.3	325.1	363.5
Savings	--	11.1%	8.3%	28.6%	24.0%

The next bar graph reveals which cases the energy supplied to a column is dissipated through the memory cell to GND. During a standard memory read, the memory cell is responsible for pulling down one of the bit-lines for both a switch and a hold. The other memory designs only discharge a bit-line during the read switch. This is why the power dissipation is so much larger for the standard read than for the other methods. For the write accesses, the power used is a combination of the power discharged from one of the memory cell nodes and power resulting from short-circuits. That is also why the novel designs dissipate more energy through

the memory cell—they have larger inverter transistors, which strengthen the dynamic short-circuits that temporarily exist during memory overwrites.

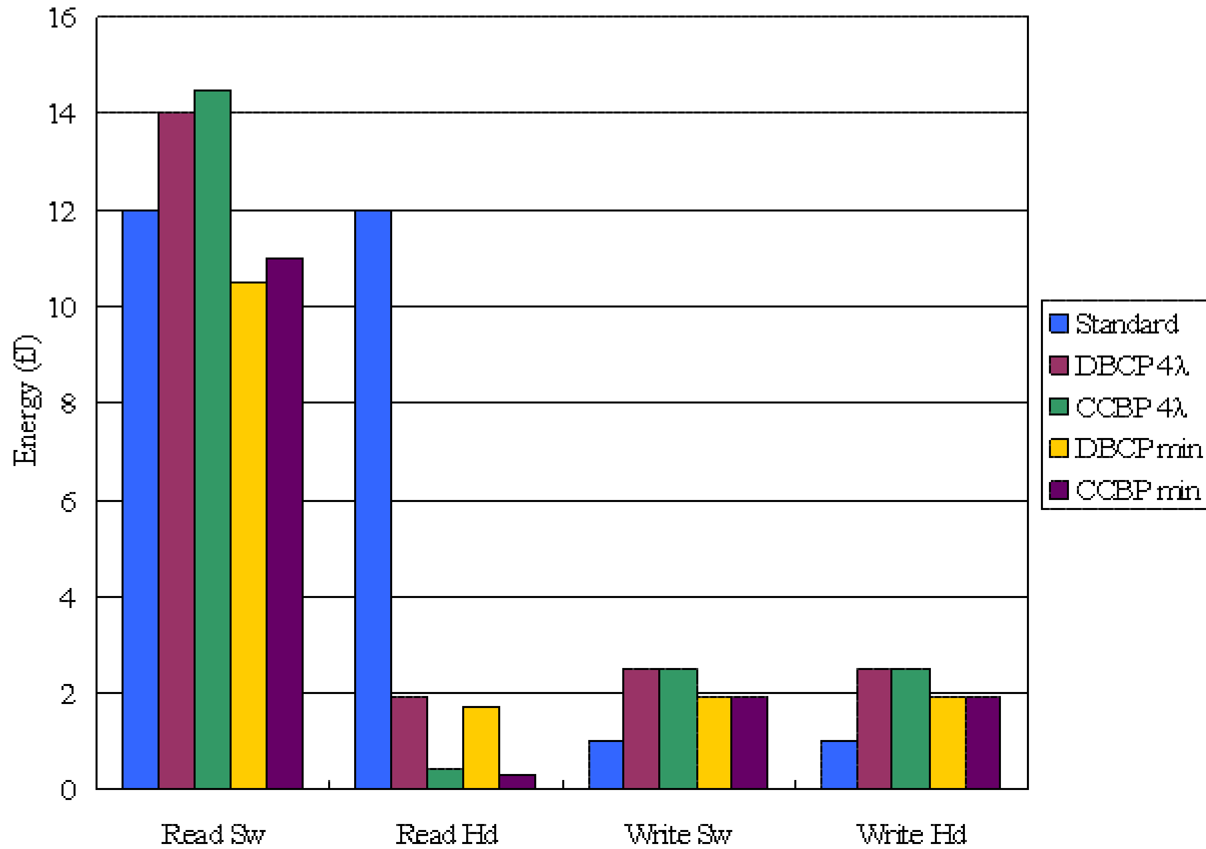

Figure 4.3: Energy dissipated through the memory cell to GND

Fig. 4.4 reveals that GND in the bit-line writing circuitry is the second destination for the current flow on a column of SRAM cells. Since the write logic is turned off during a memory read, the only power dissipated to the bit-line GND is the result of leakage currents. For memory write operation, however, power is supplied by the bit-line GND to help overwrite the memory cell. Also, for a switch, a large amount of power is dissipated on the bit-lines since the bit-lines must reverse their voltage levels.

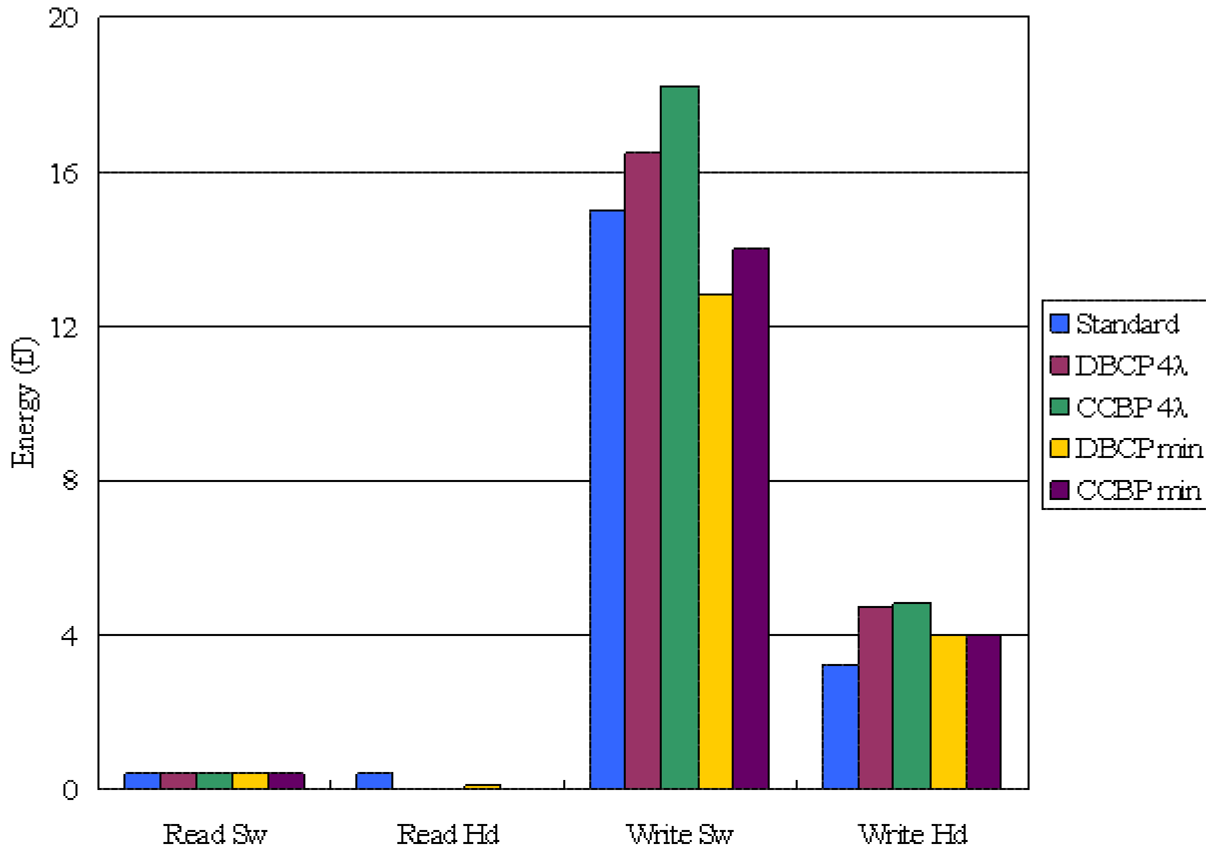

Figure 4.4: Energy dissipated through the column bit-lines to GND

The next diagram (Fig. 4.5) shows the delay of each of the SRAM designs for both a read and a write operation. The fastest design is the DBCP min SRAM which is39.6% faster than the standard memory read. The DBCP 4λ is the next fastest at 33.5%, and then the CCBP min and CCBP 4λ designs yield 29.7% and 19.9% increases in speed over the standard read. Write performance is also improved by using the new SRAM designs, however, since pipelined performance only depends on the slowest cycle time, the focus remains on the speed of the read accesses.

The results reveal that the DDBC and CCBP designs are successful in speeding up the read access time. This is true for both the 4λ and the minimum sized design. If pipelining is not

necessary in a particular system, the novel SRAM designs are still positive improvements overall since they reduce the delay of the write accesses as well.

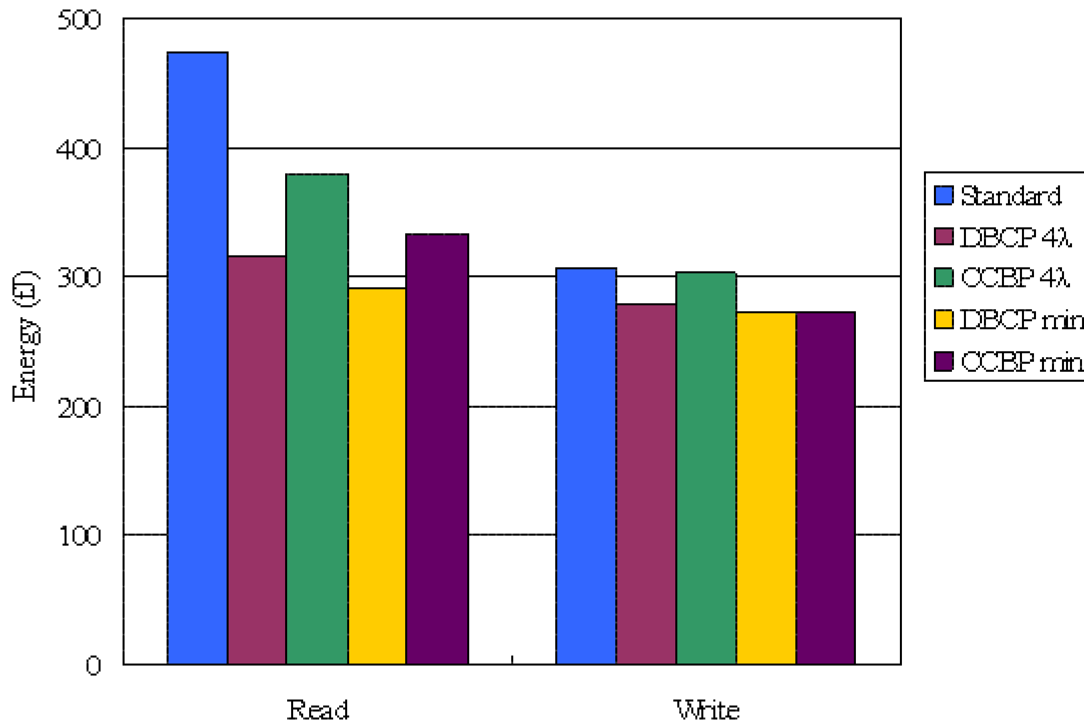

Figure 4.5: Delay comparison between the different SRAM designs

One other topic to discuss is the difference between the performances of the read circuit designs in 180-nm CMOS technology to their performance in 90-nm. A straightforward comparison can be made using the energy data in Table 4.4 and Table 3.10. These results show how the power consumed in 180-nm is almost ten time as large as that for the 90-nm read. Delay is also drastically different: a read access that took nearly one nanosecond in 180-nm technology only takes around 300 ps in the 90-nm. To a certain extent, these differences will vary based on the timing of the driving signals, but in general, the improvements in delay and energy dissipation can be attributed to the reduced voltage level and increased transistor switching speed that exists in the 90-nm technology.

4.5 Summary

SRAMs are designed to perform both read and write operations. In the previous two chapters, the memory write and memory read access were each analyzed separately. The purpose of this chapter was to analyze the read and write operations together and to test out the functionality of the two reading schemes proposed in Chapter Three. Four novel SRAMs were developed to be compared with the standard read and write methods of memory access. Each of the novel designs contains one of the two reading methods (DBCP and CCBP) and also a memory cell with access transistor widths of either 4λ or the minimum. Writing circuitry is added to each SRAM as well. The SRAMs were then subjected to a series of read and write switch and hold simulations to determine if the novel read circuits were functional and to see how the write operations were affected by the modifications to the read architecture on the column. A brief review over the results of interest follows.

- *Energy Consumption*: Assuming reads and writes occur with the same frequency, the DBCP SRAM design yields power savings between 11.1% and 28.6%, depending on the size of the memory cell accessed. The CCBP method also demonstrates savings ranging from 8.3% to 24.0%. These improvements are not as large as was shown for the read operations on their own, so it is very important to take into consideration what kind of memory access will most regularly occur before selecting which type of SRAM design to use.
- *Delay*: By using any of the four proposed SRAMs, delay for pipelined access is reduced. To achieve an increase in speed of between 33.5% and 39.6% over the standard method, the DBCP scheme should be selected. The CCBP SRAM decreases delay by either 19.9% or 29.7%. In a non-pipelined memory scheme, delay will still

be decreased overall (although by smaller percentages) since the DBCP and CCBP memories also improve the delay in write accesses.

The results presented in this chapter confirm that the novel read methods are good options for improving power and delay when the memory has a high enough percentage of read accesses or when pipelining is implemented. The size of the memory cells being accessed did play a significant role in determining the success of the reading scheme because of short-circuits through the larger memory cells. Delay was also affected by the memory cell size because of the change in bit-line capacitances. Therefore, it is necessary to carefully analyze the features of the SRAM in question before determining whether to replace it with one of the novel SRAM designs.

Chapter 5

Conclusion

In this thesis novel schemes for reducing power and delay in static random access memories (SRAMs) have been presented. Accessing memory consists of writing to and reading from memory cells, and each of those operations use power and require time to complete. An analysis of the write operation revealed several sources of dynamic power loss through short-circuits. Causes for delay were also noted. Seven novel memory cell designs were then proposed and tested as possible solutions for the delay and power problems. Three designs in particular, VGV_N, VG_C, and VG_N demonstrated notable savings of 27.6%, 12.3%, and 24.1% in the areas of energy, delay, and the energy-delay product respectively.

The memory read access was thoroughly studied as well. For high performance memory access, pipelining is a necessity. Since the memory read operation takes more time than the memory write, reduction of delay became a priority in the design of read circuitry. The standard method for reading from SRAMs is a two stage process. One stage pre-charges the bit-lines and the second stage pull bit-line down. If the bit-lines were pulled to the same voltage levels they had prior to the read, then the whole process was an unnecessary power expense. Instead, the bit-lines should only be switched when the stored value in the memory cell being read is opposite

that as the value represented on the bit-lines. To do this, two novel schemes for memory read accesses were proposed: the Bit-line Capture Pull-up (DBCP) and the Cross-Coupled Bit-line Pull-up (CCBP). By removing the pre-charge stage and its pull-up logic and replacing it with one of the novel read designs, both power and delay were successfully reduced. Energy savings for the DBCP design came to be between 44.3% and 45.9%, while its delay improvements ranged from 22.7% to 31.2% over the standard read. The CCBP reduced power by at least 41.2% and delay by between 5.8% and 16.1%. And, for smaller noise margin designs, the $CCBP_{WEAK}$ reading scheme gives energy savings of 9.2% to 98.6% and reduces delay by between 28.3% and 47.4%.

Analysis of the novel read circuits would not be complete if they were not also tested with the memory write operation. Several SRAMs with the DBCP and CCBP read circuitry were implemented and compared with the standard SRAM. The functionality and feasibility of using each design in real applications was assessed and it was shown that in cases where read accesses occur at least as frequently as write accesses, energy and delay savings are attainable. In particular, when reading at least fifty percent of the time, energy can be reduced by 8.3% to 24.0% for the CCBP design and between 11.1% and 28.6% for the DBCP SRAM. When pipelining, the CCBP design reduces delay by 19.9% to 29.7%, while the DBCP method decreases the reading delay by 33.5% to39.6%.

5.1 Contributions

This research proposes several novel schemes for reducing power or delay in SRAMs as well as the methods used to test the performance of these designs.

- *Virtual source transistors*: To prevent short-circuits between the bit-lines and the 6T memory cell as well as the short-circuits that occur in the cell during a write switch, Virtual Ground (T_{VG}) and Virtual V_{DD} (T_{VV}) can be added to the memory. T_{VG} is an nMOS transistor that is placed between GND and the inverter nMOS transistors, and can be turned off during the write access. T_{VV} is the same except that it is a pMOS transistor located between V_{DD} and the inverters' pMOS transistors. The VS control signals are generated in the decoder [9].
- *CMOS transmission gates*: The replacement of the nMOS access transistors with CMOS transmission gates allows for decreased delays in writing because a strong logic '1' can reach the internal memory cell nodes from the bit-lines [9].
- *Novel memory cells for writing*: Seven new memory cells were designed using different combinations of the virtual source transistors and CMOS transmission gates. Each of these was simulated and with the exception of the 8T memory cell, all of the cells demonstrated savings either in terms of the delay or energy consumed [9].
- *DBCP reading scheme*: For improved speed (especially when pipelining memory) and decreased energy usage during the memory read access, a novel eight-transistor bit-line capture that can help pull-up the rising bit-line was implemented in place of the pre-charge circuitry that is typically used in memory read accesses.
- *CCBP reading scheme*: Using only three transistors, this is a smaller scheme for reducing power consumption and delay during a memory read. It is slower than the DBCP design, but if size is a bigger concern than delay, then this is a good option to use for reading from SRAM without pre-charge.

- *CCBP$_{WEAK}$ reading scheme*: For systems where having large noise margins is not important, the CCBP$_{\text{WEAK}}$ novel reading scheme is ideal for low-power, high speed reads. The added circuitry and signal driver are both very small, and little power is used on the bit-lines because of the reduced swing. The reduced bit-line swing also enables fast transition times and therefore faster reads [10].
- *Combined write and read analysis*: By testing out the novel read techniques along with write operations and circuitry, it was shown in which cases savings could be best attained by using the new reading designs. If a system expects to read more frequently than write, the use of either the DBCP or CCBP schemes would be very beneficial.

5.2 Future Work

There remains more research to be done within this topic. Due to time constraints, the scope of the present research work was limited to dynamic power. However, it is recognized that an in-depth analysis of the static power consumption including sub-threshold currents and gate leakage would also help in better understanding where unnecessary losses of power are occurring. Some of the leakage current analysis discussed in [11]-[14] is very applicable to the memory write analysis with the virtual ground, so it would be worthwhile to see if the additional transistors used to prevent leakage could also be used to prevent dynamic short-circuits.

Decoding is another area where further work could be done. Decoders are responsible for a significant amount of power consumption; it would be incomplete to leave it out of a discussion on SRAM energy usage. Some research has already been performed on pipelined decoding schemes that could be used to control the memory accesses for the discussed pipelined

read and write operations. However, that work is not discussed in this thesis. The decoder would also be responsible for generating the virtual source signals for the memory write cell designs and it could also be used to activate the sampling or delayed read signals for the novel read circuitries proposed in Chapter Three.

A full system design which includes any number of the methods for improvement at the cell level, column level, or the decoding level, could also be implemented. A comparison between the low-power schemes and the current memory system would be a very revealing as to the success of the designs.

Finally, by developing a compiler or some kind of software to control the order that data is stored in memory and how it is accessed, the percentage of memory read operations that lead to a bit-line holding situation can be intentionally maximized. This would lead to much larger savings in terms of energy since it is the read bit-line hold and not the read bit-line switch condition, which generates the greatest reduction of power dissipation.

Bibliography

[1] M. J. Myjak and J. G. Delgado-Frias, "A two-level reconfigurable architecture for digital signal processing," *The 2003 International Conference on VLSI (VLSI'03),* Las Vegas, Nevada, June 23-26, 2003.

[2] J. Kim, C.H. Ziesler, and M. C. Papaefthymiou, “Energy recovering static memory,” *Proc. of the 2002 International Symposium on Low Power Electronics and Design*, IEEE, pp. 92-97, 2002.

[3] R. E. Aly, M. A. Bayoumi, and M. Elgamel, “Dual sense amplified bit lines (DSABL) architecture for low-power SRAM design,” *Proc. of IEEE International Symposium on Circuits and Systems*, pp. 1650-1653, May 2005.

[4] Y. J. Chang, C. L. Yang, and F. Lai, “A power-aware SWDR cell for reducing cache write power,” *Proc. of the 2003 Int’l Symposium on Low Power Electronics and Design*, IEEE, pp. 14-17, August 2003.

[5] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed., San Francisco, CA: Pearson Addison Wesley, 2005.

[6] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed., San Francisco, CA: Morgan Kaufmann Publishers, 2003.

[7] S. Cheng and S. Huang, “A low-power SRAM design using quiet-bitline architecture,” *Proc. of IEEE Int’l Workshop on Memory Technology Design and Testing*, pp. 135-139, August 2005.

[8] J. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed.,Upper Saddle River, NJ: Pearson Education, 2003.

[9] K. Blomster and J. G. Delgado-Frias, “Reducing power and delay in memory cells using virtual source transistors,” *48th IEEE Int. Midwest Symposium on Circuits and Systems,* pp. 299-302, Aug. 2005.

[10] K. Blomster and J. G. Delgado-Frias, “High performance memory read using cross-coupled pull-up circuitry,” *49th IEEE Int. Midwest Symposium on Circuits and Systems*, Aug. 2006.

[11] M. D. Powell, S. H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, “Gated-V_{DD}: A circuit technique to reduce leakage in cache memories,” in *Proc. of the 2000 Int. Symposium on Low Power Electronics and Design (ISLPED)*, ACM Press, pp. 90–95, July 2000.

[12] A. Agarwal, H. Li, and K. Roy, “A single-V_t low-leakage gated-ground cache for deep submicron” *IEEE Journal of Solid-state Circuits*, 2003.

[13] A. Agarwal, H. Li, and K. Roy, “DRG-Cache: A data retention gated-ground cache for low power” *Proc. of the 39th Design Automation Conference*, 2002, pp. 473-478, June 10-14, 2002.

[14] A. Agarwal and K. Roy, “A noise tolerant cache design to reduce gate and sub-threshold leakage in the nanometer regime” *Proc. of the 2003 Int. Symposium on Low Power Electronics and Design (ISLPED)*, pp. 18-21, Aug. 2003.

[15] M. Margala, “Low-power SRAM circuit design,” *Proc. of IEEE Int. Workshop on Memory Technology Design and Testing*, pp. 115-122, August 1999.

[16] A. Karandikar and K. K. Parhi, “Low power SRAM design using hierarchical divided bit-line approach,” *Proc of the Int. Conf. on Computer Design: VLSI in Computers and Processors*, IEEE, pp. 82-88, 1998.