MODI, KAUSHAL MANESH. Exploring different Architectures for an SRAM in 3DIC Technology. (Under the direction of Dr. William Rhett Davis).

The stacking of silicon wafers and processing of through-silicon vias, which is often called 3DIC Technology, has opened up the possibility of improving the performance of circuits that are wire-delay limited. This project explores the design of Static Random Access Memories (SRAM), an important building block in 3DICs. Detailed transistor-level and layout design-data are presented for bit-cells and peripheral circuitry. Two different architectures are tested and their parameters are compared. The thesis concludes with the decision of which architecture proved to be the better one for 1Kb SRAM.
Exploring different Architectures for an SRAM in 3DIC Technology

by
Kaushal Manesh Modi

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh, North Carolina

2008

APPROVED BY:

Dr. Winser Alexander Dr. Paul D. Franzon

Dr. William Rhett Davis
Chair of Advisory Committee
DEDICATION

To

MUMMI & PAPPA
BIOGRAPHY

Kaushal Manesh Modi was born on December 21, 1983 in Ahmedabad, a growing metropolitan city in the western part of India. He received his Bachelor of Engineering (B.E.) degree in Electronics and Communications Engineering from Nirma Institute of Technology (Gujarat University) in June 2005. He worked at the Space Applications Center, Indian Space Research Organization, Ahmedabad as a Digital Design Engineer for his final year project.

Kaushal has been a graduate student in the Electrical and Computer Engineering Department at North Carolina State University, Raleigh, NC since Fall 2005. He is a member of the Honor Society of Phi Kappa Phi. Since Spring 2006, he has been working with the MUSE group of Prof. Rhett Davis in the field of 3D-IC Technology. His research interests include Digital Circuit Designing and VLSI.
ACKNOWLEDGMENTS

I would not have reached where I am today, had it not been for the guidance and motivation of Manesh and Rita Modi, my parents. I thank them for making it possible for me to reach this major milestone in my life. I am where I once dreamed of being, before venturing into the field of electrical engineering, due to the added support of my grandparents, uncle and my loving sisters Raji and Rupal. Above this, I was continually guided by my spiritual guru Pujya Rajyogi Narendraji.

I sincerely acknowledge the efforts of Dr. Rhett Davis, my advisor, in providing guidance for the successful completion of this thesis. Dr. Davis provided me with the element of freedom in applying ideas to this project and also made available all possible resources required through the progress of my thesis. It was his vision to test the efficiency of an SRAM design in 3DIC technology. I am also grateful to the members of my thesis committee, Dr. Paul Franzon and Dr. Winser Alexander for devoting their time and providing useful feedback.

I am highly grateful to Ambarish Sule, Ravi Jenkal, Steve Lipa, Julie Oh and Christopher Mineo for their support during the course of my thesis. They helped me cross many roadblocks that I faced in my research, making me reach my goal in an easier manner.

Finally, I would like to thank Maitrik, Gunjan, Parth, Milan, Ushma, Ranjith, Ajit, Harsh, Aditya, Soha, Deep, Nayan and all the desi people at NC State who made my life at NC State a joyful and memorable experience. They all helped in making NC State, I cannot describe this better than the clichéd, ‘home away from home’.
# TABLE OF CONTENTS

## LIST OF FIGURES

- vii

## LIST OF TABLES

- viii

## 1 Introduction

1.1 Overview .......................................................... 1
1.2 Related Work ..................................................... 2
1.3 Why SRAM? ......................................................... 4
1.4 Contribution ....................................................... 5
1.5 Thesis Organization ................................................ 6

## 2 version 1.0

2.1 Design .............................................................. 7
2.2 Layout .............................................................. 8
2.3 Tapeout ............................................................ 10
2.4 Drawbacks ......................................................... 10
2.5 Summary of Results ............................................... 10

## 3 blsplit Architecture

3.1 Bit Cell ........................................................... 16
3.2 Row Decoder ....................................................... 17
3.3 Sense Amplifier .................................................. 19
3.4 Overview .......................................................... 22

## 4 wlsplit Architecture (1Kb)

4.1 Decoders ........................................................... 27
4.2 Sense Amplifier .................................................... 28
4.2.1 Data I/O Mesh .................................................. 29
4.2.2 SA Peripherals ................................................ 31
4.3 Overview .......................................................... 32
4.4 Results ............................................................. 32

## 5 blsplit Architecture (1Kb)

5.1 SRAM Components ................................................. 38
5.2 Results ............................................................. 39

## 6 Conclusion

................................................................. 43
LIST OF FIGURES

Figure 2.1 Schematic of the first version of bit cell ................................. 8
Figure 2.2 Layout of the first version of the bit cell ................................. 12
Figure 2.3 3d Visualization of the Design (v1) ................................. 13
Figure 2.4 Preliminary design layout with chip pads ......................... 14
Figure 2.5 Chip Photo .................................................................. 15

Figure 3.1 Bit cell used in blsplit and wlsplit architectures .................... 17
Figure 3.2 Layout of the second version of the bit cell .................. 18
Figure 3.3 4-input SCL NOR gate ........................................ 20
Figure 3.4 Examples of paginating a design across multiple tiers ........ 21
Figure 3.5 Sense Amplifier plus Multiplexer version 1 ......................... 24
Figure 3.6 3d Visualization of the blsplit architecture ....................... 26

Figure 4.1 Sense Amplifier plus Multiplexer version 2 ......................... 30
Figure 4.2 Data I/O Mesh ................................................................ 31
Figure 4.3 Sense Amplifier Peripheral Circuit ................................ 33
Figure 4.4 3d Visualization of 1Kb wlsplit SRAM ............................ 35
Figure 4.5 Final wlsplit Design Layout ........................................ 37

Figure 5.1 3d Visualization of 1Kb blsplit SRAM ............................ 41
Figure 5.2 Final blsplit Design Layout ........................................ 42
# LIST OF TABLES

Table 4.1 1Kb wlsplit SRAM performance figures .................................. 36

Table 5.1 1Kb blsplit SRAM performance figures ................................. 40

Table 6.1 Comparison of 1Kb SRAM parameters .................................. 45

Table 6.2 Comparison of equivalent bit line and word line lengths .......... 45
Chapter 1

Introduction

1.1 Overview

Gordon E Moore’s Law held quite true for the last few decades. Intel introduced Penryn, the world’s first high-volume 45nm processor in November 2007. On September 18, 2007, Intel had also advertised in a press release[1] that the world will be seeing 32nm processors in 2009. A few months later, on December 11, 2007, the Taiwan Semiconductor Manufacturing Company Limited (TSMC) reported[2] manufacturing a 2Mb SRAM test chip with the smallest bit-cell at the 32nm node. With the companies spitting out chips at 45nm and then at 32nm just a month after, the technology is actually moving at a much faster rate than the Moore’s Law (18/24 month version).

However, it is unclear that scaling below 32nm will be possible with current lithography methods. The wavelength of light used to pattern high-volume production chips is 193 nm, which is nearly 6 times as large as the smallest feature size in a 32nm technology. The methods of optical proximity correction and phase-shift masks, which
have allowed 193nm-wavelength lithography to be useful for these small feature sizes, may not work for technologies smaller than 32nm. In addition to the issue of the feasibility of lithography of smaller processes, another major issue would be the cost of development of lithography processes for smaller processes. An alternative to continued scaling of transistors is to turn to methods of “equivalent scaling”, that improve other performance aspects of a system, without changing the transistors themselves. This thesis deals with an equivalent scaling method known as 3DIC technology, which refers to the stacking of chips and the processing of through-silicon-vias between the chips in the stack.

1.2 Related Work

3DIC is a very viable technology. Since last few years, research has undergone to make 3DIC chips a large scale commodity of the consumer market.

Tezzaron® Semiconductor specializes in high-speed memory products, 3D wafer stacking processes and 3D memory elements. A technical document on their FaStack™ Technology[9] briefly describes a process in which the conventional 2D wafers can be stacked above each other to form a pseudo-3D architecture.

Massachusetts Institute of Technology Lincoln Laboratory (MITLL) has developed a through-wafer-via, 180-nm 3DIC FDSOI[4] process that offers three tiers and the highest-density vertical interconnect available. The advantages of this process over the conventional chip-stacking are:

- Lesser die thickness → The thickness of a FaStack process die is 33.5µ[9] without the overglass and silicon substrate thickness, whereas the same thickness is
22.45\(\mu\)[4] for the MITLL FDSOI process.

- No need for precise wafer-wafer alignment.
- No issue of thermo-mechanical stress that is induced in post-device processing.

The MITLL FDSOI process is still being actively researched; its advantages still need to be explored. This fact motivated me to work in 3DIC.

Ignoring the inter-tier vias, the architectures that are designed to take the advantage of the vertical interconnects of this technology should experience a considerable reduction in the average wire length. Here’s an example of a system level benefit of the 3DIC Technology. In a paper[8] published in the 24th International VLSI/ULSI Multilevel Interconnection Conference, the authors talk about the benefit of using a 3-bank 1-port FIFO in 3D over the conventional 2-bank 1-port FIFO in 2D. The ability to split the three banks of the FIFO into three separate tiers allows the FIFO to be a much less complex and less area consuming single ported, compared to the dual-ported design. The conclusion was that the 3D design was 8% faster and 6% more energy efficient.

Above was an example of 3DIC at system level. This thesis is an example of architecture exploration at circuit design level that can help improve the desired design parameters. It was decided to design an SRAM in different architectures and analyze the pros and cons of both architectures.
1.3 Why SRAM?

Static random access memory (SRAM), being the fastest of the currently high-volume manufactured memory families, continues to be a critical component for a multitude of micro-electronic applications. In earlier days, different components of a system used to be manufactured separately and then connected together on a PCB\(^1\). The disadvantages of this process were plenty: large real estate, cumulative higher cost of manufacturing the different packages used, delay caused due to the physical connections between different packages especially at high frequency operation, noise interference, high power consumption.

One of the greatest limiting factors in the systems-on-chip (SoC) used in mobile products and general purpose microprocessors used in personal computers is the need for off-chip memory accesses. To remove the bottleneck of external memory accesses, on-chip caches were introduced on the processors and now the current research is in making these caches faster. All of this boils down to the cache architecture which inherently is an SRAM.

The current MITLL 3DIC FDSOI process has three tiers. A possible future application of this technology can be a CPU in which the processor unit is designed on one tier just as in a conventional 2D 180nm process and the other two tiers hold the on-chip 3D cache. Depending on the application of the chip, it could be desired to be a low-power consuming device for handheld gadgets or a high-speed device for non-mobile products like desktop processors.

\(^1\)printed circuit board
In this thesis, I am exploring and experimenting on different SRAM topologies using this 3DIC process. The design parameters of 1Kb SRAM are compared side-by-side for two different 3D topologies.

1.4 Contribution

In the case of designing 2D layouts, there are ways in which you can minimize the use of metal layers and minimize the wire lengths by using stick diagrams and Euler paths[6]. But the layout optimization techniques are still uncharted for 3DIC technology. For 3DIC technology, we have to consider the wire lengths in horizontal dimension as well as the wire (via) length in the vertical dimension. There is a need to analyze how the design parameters differ for different layout architectures in 3DIC for the same design.

The MITLL FDSOI 3DIC process is used for the exploration of different architectures to design an SRAM in 3D. While the main goal of the thesis is to compare the layout architectures, effort is also made to make the designs as optimum as possible. The optimization process starts from the point when the schematics of bit-cells and other components are designed. After that, the next level of optimization is introduced in the manner in which custom layouts are created for each component so that the overall 3D layout can look like a homogenous compact planned design, rather than a patchwork of discrete layouts.

Different layouts of 1Kb SRAM are designed by splitting the bitlines or wordlines across the three tiers. Also a note is made on the careful placement of the 3D vias so as to minimize the parasitic capacitance, thus minimizing the charge/discharge delay
of the bitlines. In the final comparison, one architecture results in a high performance design and the other architecture results in a low power design.

1.5 Thesis Organization

This thesis is organized into the following sections:

Chapter 2 is “version 1.0”. Here I discuss the preliminary SRAM design without any peripheral circuits (which are required for large practical sized SRAMs). That chapter concludes with the tapeout of that design.

Chapter 3 is “blsplit Architecture” which is a major upgrade to version 1.0. This design has the bit-cell re-designed, the whole layout topology re-thought and the row decoding and sense amplifier modules added.

Chapter 4 is “wlsplit Architecture”. Here the SRAM is designed with a different look, with a different idea of sharing the word-lines instead of the bit-lines across the three tiers. The sense amplifiers required re-designing to enable them to be used in larger SRAM designs.

In Chapter 5, an SRAM is designed, using the blsplit architecture, of the same size as the one designed using wlsplit architecture in Chapter 4.

Finally Chapter 6 concludes this thesis by comparing the performance figures of both the topologies and determining the better one.
Chapter 2

version 1.0

This chapter begins with the general discussion of how an SRAM bit-cell is designed considering the transistor sizing issues that can cause an improper read or write from/to the bit cell. Following that, the layout is designed for a single bit-cell that is replicated on all three tiers to create a 3D multi-bit SRAM.

The first version of the SRAM design was kept as simple as possible to see if the bit cells were designed properly and if the inter-tier via connections were made properly. This design comprised of the precharge transistors, the bit cells and column multiplexers. The inclusion of row/column decoders and sense amplifiers was avoided in this first phase of designing to eliminate the chances of design errors due to these advanced modules. The first goal was to having a working version of bit cells in 3D.

2.1 Design

The bit cell was designed following its sizing rules[6] to avoid Read Upset. According to these rules, the cell ratio should be greater than 1.2 and the pull-up ratio
should be less than 1.8. The cell ratio (CR) and the pull-up ratio (PR) are defined in the following equations:

\[
CR = \frac{W_3/L_3}{W_0/L_0} \tag{2.1}
\]

\[
PR = \frac{W_5/L_5}{W_1/L_1} \tag{2.2}
\]

Figure 2.1 shows the basic schematic of a bit cell and the sizes of the transistors used for this design.

![Figure 2.1: Schematic of the first version of bit cell](image)

2.2 Layout

The layout of the bit-cell design in Figure 2.1 is shown in Figure 2.2. The dimensions of this version are 5.675μ(wd) × 8.175μ(ht).

This was a 6-byte RAM, with 2 bytes on each tier. As the decoder was not designed for this version of the design, the wordline selects for each row (2 rows on each tier) were given as 6 separate inputs to the design.
Even though the SRAM is designed in 3D, it has a single set of inputs and outputs connecting the design through the top tier. The same i/o lines need to be transported to the bit cells in all the tiers. The i/o’s are differential, that is, data is written into each bit cell through din(bl) and dinbar(blbar); and data is read out through dout(bl) and dinbar(blbar). The column multiplexers control whether the bl/blbar lines should be used as input or output.

To facilitate this, the bitlines (bls and blbars) for each byte were shorted across the three tiers. The resultant set of bitlines for 8 discrete bits formed an input/output port to the set of 8 column multiplexers.

The bit cell layout (Figure 2.2) dimensions were 5.675µ(wd) × 8.175µ(ht) while that of the corresponding column multiplexer were 5.85µ(wd) × 12.15µ(ht). As the column multiplexers were designed wider than the bit cells, it was difficult to align the bitlines of the bit-cell block with the bitlines of the column multiplexer block, if all 8 column multiplexers were laid out side-by-side on the same tier. In order to accommodate all 8 column muxes in the width of 8 bit cells, the layout of column multiplexers was split between two different tiers. The column muxes for even bits were laid out on the top-most tier (C) and the ones for the odd bits were laid out on the middle tier (B). Figure 2.3 will help give a better idea of this logic arrangement in 3d.

This preliminary version of the SRAM has just two bytes of memory on each tier and four column muxes each on tiers C and B. Inter-tier vias are required to transport the data i/o signals to tier (B). Putting another set of muxes on tier (C) instead of on A eliminates the need of similar vias from tier (C) to A.
2.3 Tapeout

This design was included in the new chip to be taped out using the MITLL 3DIC 180nm Technology in October of 2006. Figure 2.4 shows the layout of the design with the chip pads and Figure 2.5 shows the photo of the whole chip. The SRAM logic and I/O pads can be seen on the left side in the chip photo.

2.4 Drawbacks

- This design being small and basic did not have the required peripheral circuits of an SRAM like the row and column decoders, input buffers, input signal generation circuits and sense amplifiers. So the resulting SRAM parameters weren’t dependable for a practical use.

- For each tier, the bl lines for all bits were shorted at the top of the bit cell block, while all the blbar lines were shorted at the bottom of the block. For such a small design, this asymmetry did not cause any problem, but for larger designs this could create major timing differences between the bl and blbar lines of the same bit, as the propagation delay of the signals from the inputs of the circuit to the bl and blbar nodes on bit cells will be different.

2.5 Summary of Results

The experience gained from designing this first version of the SRAM helped me build a better and more practical design as will be seen in the final chapters.

- Because of improper planning, the layout widths of the bit-cells and the column multiplexers were caused to be different which introduced design complexity like
interleaving the column multiplexers between top two tiers. It was learned that the bit-cells should be designed so that its dimensions match with the modules it is being connected to.

• Input buffers were not used in this design for the same purpose that decoders and sense amplifiers were not used. But eliminating these buffers made the SRAM work in ideal fashion, and the effects of gate capacitance loading got neglected in the designing process. Buffers should be used for all input signals to create more practical simulation results and make the design implementable.

• This design layout had asymmetrical design for bl and blbar lines. For larger sized memories, this asymmetry can result in significant differences in the charge/discharge times for both the lines.
Figure 2.2: Layout of the first version of the bit cell
Figure 2.3: 3d Visualization of the Design (v1)
Figure 2.4: Preliminary design layout with chip pads
Figure 2.5: Chip Photo
Chapter 3

blsplit Architecture

The design of a 48-bit SRAM was demonstrated in Chapter 2. It concluded with the design tapeout and changes to make the design more practical. In this chapter, those changes, namely: matching the layout dimensions of the bit cell and the column multiplexor, making the layouts symmetrical and including input signal buffers, are implemented.

In addition to those changes, the row decoder and sense amplifier components are added to the SRAM architecture. The resultant SRAM has $16 \times 3$ rows (16 rows on each tier) and one 8-bit column; 384 bits versus the earlier 48 bit version.

blsplit is an acronym for ‘bit line split’. In this SRAM architecture, the bit lines (bl and blbar) and split and distributed across all tiers. Inter-tier vias are used to short the bit lines across the tiers. The three split bit lines shorted at one end using inter-tier vias can be visualized at a bit line three times the length in two dimensions. This architecture is explained in detail in this chapter.
This chapter walks through the redesigning of the bit-cell schematic and layout to implement the matching of bit-cell and sense-amplifier layout widths. The row decoder design follows after that. Later in this chapter, the design of the sense amplifier is explained, which also incorporates the column multiplexer. The chapter concludes with the explanation of the SRAM architecture that uses these new and redesigned components.

### 3.1 Bit Cell

The re-designing started from the bit cell. In the earlier version, the largest transistor was 4$\mu m$. In order to reduce the overall area and still keep the bit cells functioning properly, after some testing and iterations, the largest transistor was shrunk down to 1.35$\mu m$. The updated bit cell design is shown in Figure 3.1.

![Bit cell schematic](image)

Figure 3.1: Bit cell used in blsplit and wlsplit architectures

The overall impact is seen in the layout. The area of the older bit cell was 5.675$\mu m \times 8.175\mu m$ and that was reduced to 6.2$\mu m \times 6.175\mu m$ in this version. Another major difference is that the earlier layout design was very asymmetric: the WL
line had many corners, was placed closer to the $V_{DD}$ rail, and BL and $\overline{BL}$ were unequal in length which could create unwanted charging/discharging delays between the two lines for large sized memories. All these issues were solved in the second version, making the layout symmetrical, horizontally as well as vertically. This can be compared between Figures 2.2 and 3.2.

Figure 3.2: Layout of the second version of the bit cell
3.2 Row Decoder

This block has four main sections: the input buffers for address and clock/precharge, NOR gates, WL drivers and page selector.

The input buffers generate the $ADDR_n$ and $\overline{ADDR_n}$ signals for the SCL\(^1\) NOR gates. The source of the clock given to these gates is same as the source of the precharge signal used for pulling up the bit lines. The clock for SCL gates and the precharge signal are buffered separately as required.

The main constraint while designing of the decoders is that the height of the decoder for each row must be less than or equal to that of the bit cell (6.175 $\mu$) to reduce the designing and wire routing complexities. Even a 4-to-16 sub-decoder will result in large pmos stacks if CMOS logic is used. The resultant design would be too large to fit in the given space constraint. The larger pmos stacks will also cause this design to be slower. This problem is solved if SCL NOR gates[5] are used.

The advantages of source-coupled logic are:

- The p-net will have a maximum stack of 1 pmos while the n-net will have a maximum stack of 2 nmoses.
- The design is faster than the complementary MOS logic.
- Complementary outputs are available simultaneously without any inverter delay.

Figure 3.3 shows the schematic of the gate used in all row decoders.

\(^1\)source-coupled logic
Considering the height constraint imposed due to bit cell height, it was possible to have an SCL NOR gate with a maximum of four inputs. So the maximum size of one stage of the decoder can be 4-to-16. For adequate sized memories, this size is not adequate. That problem can be solved by using Divided Word Line (DWL)[10] architecture. We can have a global 4-to-16 row decoder and 4-to-16 or smaller row sub-decoders. Larger memories can be created similarly by nesting each sub-decoder with another set of sub-decoders. That would lead to the Hierarchical Word Decoding (HWD)[3] architecture.

The 3DIC technology leads to a natural thinking of folding up a conventional 2D layout in three folds (as currently only three tiers are available). Now this folding can either occur vertically: the bit lines are split across tiers, or horizontally: the word lines are split across tiers. This chapter demonstrates the SRAM with vertical
folding. The design with horizontal folding is demonstrated in Chapter 4.

Considering that ‘n’ tiers are available for the design, the SRAM bit-cell block can be divided into ‘n’ pages, with each page on a separate tier. In general, if we keep the length of word lines constant and if the original length of bit lines were ‘len_bl’, then this technique reduces that length to ‘len_bl/n’. Depending on the type of design being worked on, and considering that 3 tiers are available, Figure 3.4 shows two examples of how pagination can be implemented: it could be a design with a processor in one tier and memory in other two tiers; or it could be a whole memory chip by itself.

![Figure 3.4: Examples of paginating a design across multiple tiers](image)

Each of these three pages can be enabled individually using decoding structure which is a modification of the DWL\(^2\). The DWL uses only one global decoder. In this case of 3d structure, I have the liberty of placing clones of global decoders on each tier in the manner shown in Figure 3.6. All the clones are fed the same address lines as inputs. The advantages of doing so are,

- The word lines on each tier have their own set of drivers. Keeping the drivers only on one tier can degrade the word line pulse shapes when they are transferred to other two tiers through inter-tier vias. Having drivers only on one tier

\(^2\)Divided Word Line structure
generates the requirement of including extra buffers, which eventually leads to more area.

- In case the word line drivers are placed on only one tier, the same area on other two tiers remains empty. A better option would be to utilize the same space to drive the word lines on the corresponding tiers.

A separate 2-input NOR gate on each tier creates a tier-select signal that enables the outputs of the global decoder on that tier. After this first stage of tier selection, the DWL or HWD structure can be implemented as required for the design size.

### 3.3 Sense Amplifier

The bitlines have very large capacitance. So the complete rise and fall times are very long. As a result, when the bit cells are read, it is not practical to use the signals directly from the bit lines as the final outputs. The sense amplifiers sense the change of voltage on the bit lines and amplify it to the full line voltage at the output. Thus they make the SRAM reads faster, and also the bit lines are not required to have a full line voltage swing, thus saving power.

The basic design of the sense amplifier is borrowed from a relatively recent paper published on high-performance and low-voltage SA\textsuperscript{3} techniques\textsuperscript{[7]}. The design presented in this paper was not fit to be readily adapted in the blsplit architecture. So modifications were made to this design to optimize the area for this architecture. Further changes are made in this design for the 1Kb designs demonstrated in Chapters 4 and 5.

\textsuperscript{3}sense amplifier
The column multiplexer controls whether new data is required to be written on the bit lines or whether data is to be read from the sense amplifier outputs. Thus the bit lines are bi-directional: they act as inputs to the sense amplifier and as outputs to the column multiplexer.

Having column multiplexers as separate modules posed a space constraint. As the bit cell width is 6.2\( \mu \)m, the column multiplexer as well as the sense amplifier were required to be placed in the same width per bit-column. Routing the bl and blbar lines from the bitcell block to the column multiplexer as well as the sense amplifier becomes very difficult if it is tried to place them separately in this constrained width. As a solution, the column multiplexer is fused with the sense amplifier design (Figure 3.5). Among all the custom layout designs in this project, the layout of the sense amplifier was the most creative and challenging one because 28 transistors had to be placed in an available 6.2\( \mu \)m wide rectangular area.

3.4 Overview

The blsplit architecture is tested using a 3-tiered SRAM module. Considering a step-by-step verification of various modules in increments as an efficient method for creating larger designs, in this second version of SRAM, only the row decoders and sense amplifiers have been included in the peripheral circuits. The column decoders are included in the wlsplit architecture in Chapter 4.

In a conventional 2d SRAM module, the bit lines have precharge transistors at one end and column multiplexer contacts at the other end. But in this architecture, as the bit lines are split across three tiers, they would form ‘E’-shaped bl and blbar
Figure 3.5: Sense Amplifier plus Multiplexer version 1

lines in the vertical plane. The end where the bit lines are shorted is where the sense amplifier and column multiplexer circuits are required. On the other end, the three bit lines are open ended to which precharge transistors are connected.

An SRAM module of 16 rows of 1 byte each on all three tiers was tested using this architecture. As the same bit line is being routed to all three tiers, virtually a $3 \times N$ row 2d SRAM is generated where $N$ is the number of rows on one tier.

The unique points of blsplit architecture are:
• As the bit lines are shorted on one end, a separate set of sense amplifiers and column multiplexers is not required for each tier. We get the liberty of placing only one such set on the topmost tier, avoiding the need of 3 sense amplifiers for 3 tiers.

• As the bit lines are shorted, inter-tier vias are required only for the propagation of bl and blbar signals to all the tiers. If the bit lines were not shorted, sense amplifiers would have been required on all tiers and inter-tier vias would have been required to propagate data-in (Din and Dinbar) and data-out (Dout and Doutbar) signals to all tiers.

• Each tier will have its own row decoding block corresponding to the bit cell block on that tier. So only the input address lines are required to be ported to all tiers, rather than porting the outputs of the row decoders to all the tiers. If we take the example of the current design, 4-to-16 row decoders are used. As row decoders are instantiated on each tier, inter-tier vias are required only for the 4 input lines, not for the 16 output lines.

• Memories can be created of sizes $M \times 2^N$ (where M is the number of tiers, N is an integer) instead of the traditional $2^N$ sizes without much real estate overhead. A small 2-input decoder is put on each tier that selects only one tier according to the combination of the most significant 2 bits of the address bus. For instance, in this design, $A_5A_4 = 00$ will select a row from Tier C, $A_5A_4 = 01$ will select a row from Tier B and $A_5A_4 = 10$ will select a row from Tier A.

Figure 3.6 gives a visualization of how the architecture looks and how much area it utilizes on each tier.
Figure 3.6: 3d Visualization of the blsplit architecture
Chapter 4

**wlsplit Architecture (1Kb)**

This architecture is a yet further upgrade to the blsplit architecture. The sense amplifier was redesigned as the earlier version in Chapter 3 was not ready for an SRAM design with multiple columns. That version lacked tri-state buffers at the sense amplifier outputs. A combinational circuit design was made for the generation of sense amplifier inputs, namely write, SA enable and precharge. Timing them manually through an input vector is quite impractical. Major re-routing of wires leading into and out of the sense amplifiers was also required.

As the name *wlsplit* suggests, in this architecture, the word lines are split and distributed across all three tiers. Inter-tier vias are used to create this split. As the bit lines are not split across the tiers, each column on each tier gets its own sense amplifier. As a result, the bit lines do not have to be transported to the other two tiers. However, a set of inter-tier vias and intra-tier metal layers are required for shorting each of the Data In (din’s and dinbar’s) and Data Out (dout’s and doutbar’s) on all tiers so that a single I/O interface can be set up on the topmost tier.
Another important improvement in this design is that none of the input signals is given directly to the design. In earlier designs, inputs like SA enable (sense amplifier enable) and SA precharge had to be timed manually in the vector input files. The importance of timing matching can be explained with an example of a bit cell read. Before the values from a bit cell appear on the bl and blbar inputs of the sense amplifier, there is a considerable delay on the data-path from the input address lines, through the input buffers, through the row decoders, and through the length of the bit lines. All this delay should be incorporated in the input datapath of the the sense amplifier so that those inputs and the inputs from bl and blbar arrive at the same time at the sense amplifier. Similarly, another set of buffers are used to buffer the input data that generate Din and Dinbar signals for the sense amplifiers.

4.1 Decoders

In the blsplit architecture, each tier had its own set of row decoders. But in this architecture, the outputs from a single set of row decoders is sent to all the tiers. As a result, the wordline drivers are designed to drive a stronger capacitive load in this version.

The following example helps visualize the architecture difference between blsplit and wlsplit. Consider three word lines: wl1, wl2 and wl3, one above the other in tiers C, B and A respectively. In the blsplit architecture, each of these word lines got selected differently for a different input address. In wlsplit architecture, as the word lines are shorted vertically, wl1, wl2 and wl3, all will be selected. So a mechanism was required for address disambiguation. As a result, a tier selector was designed that selected one tier at a time on basis of the input address.
The current design has 16 rows and 8 columns, 8-bit wide. The wlsplit nature is demonstrated by distributing the 8 columns among 3 tiers: 3 columns on the topmost tier (C), 3 on tier B and 2 on tier A. So the column decoder was introduced in this design to select a column on each tier.

The column decoders also use the SCL NOR gate design used in the row decoders (Section 3.2). A clock buffer was used to supply clock to the row decoders and bitline precharge transistors. A similar buffer is used to supply clock to the column decoders and the sense amplifiers.

### 4.2 Sense Amplifier

The sense amplifiers required massive redesigning to suit this architecture. As the earlier design had only one column (8 bits wide), it was not necessary to route the data in/out and column select lines to the base of the sense amplifier. This design changed that.

In this design, as mentioned earlier, tiers C and B have 3 columns each, and tier A has 2 columns. A set (8 bits) of sense amplifiers is required for each of these 8 columns. As the data being input could be an input for any one of the eight columns, it is required that the data in lines of all the sense amplifiers be shorted. Similarly, it is required that the data out lines be shorted. So the data I/Os of all the columns need to be connected together in inter-tier as well as intra-tier fashion.

The sense amplifier design required to be modified. The pull-down network of the sense amplifiers was resized to get faster 1 to 0 response. The original sense amplifier design[7] had non-tri-state inverters at the output stage. That had to be changed for
this design because the outputs of the sense amplifiers of all columns were required to be connected together. To make that possible, without creating a tied-AND net, the inverters were replaced with tri-state inverters. The resultant schematic of the sense amplifier after the required modifications in shown in Figure 4.1. Now the outputs of the sense amplifiers of all columns remain at high-Z except for the column that is selected.

Figure 4.1: Sense Amplifier plus Multiplexer version 2
4.2.1 Data I/O Mesh

This module consists purely of interconnects, without any active components. It connects together the data and databar in/outs of the columns stacked above each other on the three tiers. It also connects each column with its column select line from the column-select bus running below this mesh. Figure 4.2 shows the layout of this data I/O mesh for 8 bits. For each bit, five lines can be seen running from top to bottom for the following signals: column_select, data_in, data_in_bar, data_out and data_out_bar. The same I/O mesh can also be seen in the final wsplit SRAM layout, at the bottom of the sense amplifier block, in Figure 4.5.

![Figure 4.2: Data I/O Mesh](image_url)
4.2.2 SA Peripherals

Figure 4.3 shows the schematic of this module. This module serves the following purposes:

- Sync the rising edge of write signal with the falling edge of the column select (active low) generated by the column decoders.

- Eliminate the need of an extra input signal for reads by generating an internal read signal using the write signal and precharge clock.

- Generate the Sense Amplifier Enable (SAen) signal with the required delay of its rising edge after the rising edge of precharge (active low).

- Synchronize the falling edge of SAen with that of the precharge signal.

Keeping the synchronization of the write/read cycles with the precharge cycles of the sense amplifiers is very important. A slight mismatch in sync can lead to drastic differences in rise/fall times of Data Out.

Each tier has its own peripheral circuits module for two purposes. One is to distribute the total driving load of precharge, SAen and write signals among separate tiers. Another reason is that the load faced by these three signals can be different on different tiers. Having these signals’ generation circuits discrete for each tier give the designer a provision to time/buffer each signal differently as required for each tier.

4.3 Overview

Figure 4.4 shows the 3d visualization of the 1Kb SRAM in wsplit architecture with all the required peripheral components, which were incrementally added to design
Figure 4.3: Sense Amplifier Peripheral Circuit

Normally in SRAMs, the bitlines are much longer than the wordlines. In blsplit architecture, there is only one set of sense amplifiers to load data (as the sense amplifier blocks also incorporate the column muxes to drive data onto the bit lines during writes.) into the bit lines on all the three tiers. For larger sized SRAMs for practical applications, this may not be very efficient due to increased capacitive load. In wlsplit architecture, each tier has a set of sense amplifiers of its own for the bit lines on it. The job of the word lines is to turn on/off the pass transistors M0 and M1 (Figure 3.1). So to compensate with the increase in length of the wordlines, adding buffers in the wordline drivers will suffice. On the other hand, modifying the bit line lengths requires modification of the sense amplifiers, because the change in the bit line capacitive load will need reconfiguration of the timing of the SA Enable signal to the sense amplifier. Also the drivers driving data onto the bit lines will need to
be resized. Thus, wlsplit architecture will be more suited where the high number of sense amplifiers does not cause the design to consume more power than the maximum power requirement limit specified in the design specs.

4.4 Results

The SRAM is first tested by writing 0xa5 to a random address and reading back from that location. A similar write/read cycle is repeated for an inverted data value 0x5a. This process gives the average power consumed in writing/reading 0’s and 1’s; and also helps calculate the write/read delays separately for 0’s and 1’s.

An extracted netlist is created from the layout which includes the smallest parasitic capacitances. In other words, the calculated delays reflect the effects of lumped capacitances created in layout extraction. Different delays are calculated, namely: the input buffer delays, the delay from the output of wordline drivers to the output of read data on the bit lines, the delay from bit lines to the internal nodes in the sense amplifiers, and output buffer delay. These delays are listed in Table 4.1.

The wlsplit architecture is easily distinguished by the Row Select vias and the I/O Mesh seen in its layout (Figure 4.5).
Figure 4.4: 3d Visualization of 1Kb wsplit SRAM
## Table 4.1: 1Kb wsplt SRAM performance figures

<table>
<thead>
<tr>
<th>Delay</th>
<th>Explanation</th>
<th>ps</th>
</tr>
</thead>
<tbody>
<tr>
<td>delay_addr_buf</td>
<td>Address Buffer</td>
<td>86.7</td>
</tr>
<tr>
<td>delay_write_buf</td>
<td>Write Buffer</td>
<td>228.5</td>
</tr>
<tr>
<td>delay_pch_buf</td>
<td>Precharge Clock Buffer</td>
<td>160.3</td>
</tr>
<tr>
<td>delay_rdec</td>
<td>Row Decoder</td>
<td>163.2</td>
</tr>
<tr>
<td>delay_cdec</td>
<td>Column Decoder</td>
<td>169.1</td>
</tr>
<tr>
<td>delay_pch_read</td>
<td>Pchbar→Read</td>
<td>135.4</td>
</tr>
<tr>
<td>delay_pch_pchsa</td>
<td>Pchbar→Pchbar_SA</td>
<td>343.8</td>
</tr>
<tr>
<td>delay_rd_saen</td>
<td>Read→SAen</td>
<td>584.8</td>
</tr>
</tbody>
</table>

### Write Delays

| d_sa_w1_wl_int         | Column Select→Data(1) transfer to bl        | 241.7 |
| d_sa_w0_wl_int         | Column Select→Data(0) transfer to bl        | 94.2  |
| d_bc_w1_wl_int         | Row Select→Data(1) transfer to bit-cell     | 242   |
| d_bc_w0_wl_int         | Row Select→Data(0) transfer to bit-cell     | 155.7 |

### Net Write Access Time

242

### Read Delays

| d_bc_r1_wl_int         | Row Select→Data(1) transfer to bit-line     | 101.5 |
| d_bc_r0_wl_int         | Row Select→Data(0) transfer to bit-line     | 58.04 |
| d_sa_r1_saen_saint     | Sense Enable→Data(1) transfer to SA internal node | 83.51 |
| d_sa_r0_saen_saint     | Sense Enable→Data(0) transfer to SA internal node | 143  |
| d_sa_r1_saint_dout     | Buffering Data(1)                           | 57.41 |
| d_sa_r0_saint_dout     | Buffering Data(0)                           | 105.7 |

### Net Read Access Time

248.7

<table>
<thead>
<tr>
<th>Power</th>
<th>mW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Kb SRAM</td>
<td>24.16</td>
</tr>
</tbody>
</table>
Figure 4.5: Final wlsplit Design Layout
Chapter 5

blsplit Architecture (1Kb)

Chapter 3 discussed the design of a 384-bit SRAM in blsplit architecture and Chapter 4 discussed the design of a 1-Kb SRAM in wlsplit architecture. In order to compare these two architectures on the same grounds, in this chapter, the blsplit SRAM design in Chapter 3 is upgraded to the size of 1Kb.

This chapter explains the differences in the layout design of 1Kb SRAM in blsplit architecture and wlsplit architecture and how the final design modules are used and rearranged from the wlsplit design to build this blsplit design. At the end, the power and delay parameters are calculated for the blsplit 1Kb design.

5.1 SRAM Components

In order to achieve a fair comparison, this second version of blsplit architecture is designed by just fetching the components used for the wlsplit architecture and connecting them as this architecture requires.
Similarities are kept in the number of inputs, the sequence and timing of input signals given to the design, the bit-cell, the input/output buffers, row/column decoder designs, SA input signal generation circuit and the sense-amplifier.

Differences observed are solely due to the architectural differences:

- In blspit, most of the inter-tier vias are used to connect together the bl/blbar lines on all tiers. In wlsplit, most of these vias were used to connect the data input/output and row-select lines on all tiers.

- In blsplit, each tier has its own set of row decoders. wlsplit had a set of sense amplifiers for each column on each tier.

- blsplit requires row decoders and tier-select decoders on each tier as well as a column decoder on the top-most tier. The wlsplit architecture had a row decoder only on the top-most tier and a column decoder that was split across all three tiers.

Figures 5.1 and 5.2 show the 3d visualization and the layout, respectively, of the 1Kb blsplit SRAM.

5.2 Results

The design parameters for the blsplit SRAM are calculated in the same way done for the wlsplit SRAM. The write and read access times for 0’s and 1’s is calculated by writing/reading 0x5a and 0xa5 to a random address.

The layout extraction step creates the design netlist with the lumped capacitances of the metal layers. So all the signal rises and falls reflect the path delay created by
these capacitances. Just as in Chapter 4, various buffer delays and important path delays are calculated using hspice .measure statements and the results are shown in Table 5.1.

The blsplit architecture is easily distinguished by the bl and blbar vias seen in its layout (Figure 5.2).

Table 5.1: 1Kb blsplit SRAM performance figures

<table>
<thead>
<tr>
<th>Delay</th>
<th>Explanation</th>
<th>ps</th>
</tr>
</thead>
<tbody>
<tr>
<td>delay_addr_buf</td>
<td>Address Buffer</td>
<td>86.41</td>
</tr>
<tr>
<td>delay_write_buf</td>
<td>Write Buffer</td>
<td>228.6</td>
</tr>
<tr>
<td>delay_pch_buf</td>
<td>Precharge Clock Buffer</td>
<td>148.2</td>
</tr>
<tr>
<td>delay_rdec</td>
<td>Row Decoder</td>
<td>182.2</td>
</tr>
<tr>
<td>delay_cdec</td>
<td>Column Decoder</td>
<td>166.7</td>
</tr>
<tr>
<td>delay_pch_read</td>
<td>Pchbar→Read</td>
<td>125.3</td>
</tr>
<tr>
<td>delay_pch_pchsa</td>
<td>Pchbar→Pchbar_SA</td>
<td>310.5</td>
</tr>
<tr>
<td>delay_rd_saen</td>
<td>Read→SAen</td>
<td>584.2</td>
</tr>
<tr>
<td>Write Delays</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d_sa_w1_cs_bl</td>
<td>Column Select→Data(1) transfer to bl</td>
<td>134.3</td>
</tr>
<tr>
<td>d_sa_w0_cs_bl</td>
<td>Column Select→Data(0) transfer to bl</td>
<td>151.0</td>
</tr>
<tr>
<td>d_bc_w1_wl_int</td>
<td>Row Select→Data(1) transfer to bit-cell</td>
<td>288.2</td>
</tr>
<tr>
<td>d_bc_w0_wl_int</td>
<td>Row Select→Data(0) transfer to bit-cell</td>
<td>168.8</td>
</tr>
<tr>
<td></td>
<td><strong>Net Write Access Time</strong></td>
<td>288.2</td>
</tr>
<tr>
<td>Read Delays</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d_bc_r1_wl_bl</td>
<td>Row Select→Data(1) transfer to bit-line</td>
<td>68.9</td>
</tr>
<tr>
<td>d_bc_r0_wl_bl</td>
<td>Row Select→Data(0) transfer to bit-line</td>
<td>77.93</td>
</tr>
<tr>
<td>d_sa_r1_saen_saint</td>
<td>Sense Enable→Data(1) transfer to SA internal node</td>
<td>80.47</td>
</tr>
<tr>
<td>d_sa_r0_saen_saint</td>
<td>Sense Enable→Data(0) transfer to SA internal node</td>
<td>142.6</td>
</tr>
<tr>
<td>d_sa_r1_saint_dout</td>
<td>Buffering Data(1)</td>
<td>78.51</td>
</tr>
<tr>
<td>d_sa_r0_saint_dout</td>
<td>Buffering Data(0)</td>
<td>54.35</td>
</tr>
<tr>
<td></td>
<td><strong>Net Read Access Time</strong></td>
<td>196.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power</th>
<th>mW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Kb SRAM</td>
<td>17.77</td>
</tr>
</tbody>
</table>
Figure 5.1: 3d Visualization of 1Kb bsplit SRAM
Figure 5.2: Final bsplit Design Layout
Chapter 6

Conclusion

The development of a 1Kb SRAM in 3DIC and exploration of its two different layout architectures have been documented in this work. It was seen that given a required size of SRAM, the design could be implemented in different architectures, giving different performance parameter values.

The circuit design was done in incremental fashion, testing and debugging each new component added to the design. The design process started with designing a basic 48-bit SRAM without any decoder or sense amplifier. A 6-byte memory was split into three parts along the bit lines and placed among three tiers. This design was improved in the next phase by making the bl and blbar symmetrical in the bit cell layout and correcting the placement of vias so that bl and blbar lines do not have different load capacitances. In this next design phase, a 16-byte memory was placed on each tier. This required the design of row decoders and sense amplifiers. The decoding structure followed the Hierarchical Wordline Decoding\cite{3} topology: consider the decoder that selects one of the three tiers as the main decoder and the decoder on each tier as a sub-decoder. In this design again, the design partitioning was done by splitting the
Once the basic components were designed and tested, a new architecture was designed using the same components; also modifying them if required. In the blsplit architecture, row decoders were placed on each tier and a single set of sense amplifiers was used for all columns in the same vertical dimension. Now a different approach was used in which sense amplifiers were placed on each tier and a single set of row decoders was placed on the top tier; effectively splitting the whole SRAM into three parts along the word lines. This was the wlsplit architecture. One more component, an important one, was added during this architecture redesign phase: the Sense Amplifier signal generation circuit. Earlier the SA Enable, read/write and precharge signals were required to be toggled from the input vector file for simulation. Now that part was made more practical by generating these signals with proper timings using just the input Precharge clock (common for bitline precharge transistors, SCL logic in the decoders and sense amplifiers) and input write signal. This SRAM was 1Kb in size.

Finally, for a valid design comparison, the blsplit architecture SRAM was redesigned for a 1Kb size using all the components used for 1Kb wlsplit SRAM. The performance parameters of both architectures are compared in Table 6.1.

In the current design, the length of bit lines is equivalent to 16 times the bit cell height and the length of word lines is equivalent to 24 times the bit line width. As the bit cell height and width are almost same, consider both to be equal to some unit. So in the blsplit architecture, the maximum equivalent bit line and word line lengths are 48 units (16 × 3) and 24 units (8 × 3, because of maximum of 3 columns per tier).
Table 6.1: Comparison of 1Kb SRAM parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>wlsplit</th>
<th>bsplit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Access Time</td>
<td>242ps</td>
<td>288.2ps</td>
</tr>
<tr>
<td>Read Access Time</td>
<td>248.7ps</td>
<td>196.95ps</td>
</tr>
<tr>
<td>Power</td>
<td>24.16mW</td>
<td>17.77mW</td>
</tr>
<tr>
<td>Power-Delay-Product (PDP)</td>
<td>6e-12W-s</td>
<td>5.12e-12W-s</td>
</tr>
<tr>
<td>Energy-Delay-Product (EDP)</td>
<td>1.49e-21J-s</td>
<td>1.48e-21J-s</td>
</tr>
</tbody>
</table>

In the wlsplit architecture, the maximum equivalent bit line and word line lengths are 16 units and 64 units (8 $\times$ 8, because of 8 8-bit columns) respectively. This is summarized in Table 6.2.

Table 6.2: Comparison of equivalent bit line and word line lengths

<table>
<thead>
<tr>
<th>Parameter</th>
<th>wlsplit</th>
<th>bsplit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Equivalent bit line length</td>
<td>16</td>
<td>48</td>
</tr>
<tr>
<td>Max Equivalent word line length</td>
<td>64</td>
<td>24</td>
</tr>
</tbody>
</table>

Thus, for the current configuration of word line and bit line lengths, we see that the bsplit SRAM is slower during writes as its column muxes (designed within sense amplifiers) have to drive approximately 3 times the load compared to wlsplit. It is also evident that as wlsplit has a smaller number of sense amplifiers, it consumes much lesser power when compared to bsplit. For this configuration, by looking at the PDP and EDP numbers from Table 6.1, the bsplit architecture is better.

But this is not true for all combinations of different number of rows and columns. For any given SRAM size requirement, it is recommended to experiment on the variation in the number of rows and columns, till the best EDP/PDP number is achieved for either of the two architectures.
Bibliography


