HIGH PERFORMANCE STATIC RANDOM ACCESS MEMORY DESIGN FOR EMERGING APPLICATIONS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Xiaowei Chen

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

June 2018

Fargo, North Dakota

North Dakota State University
Graduate School

Title

High Performance SRAM Design for Emerging Applications

By

Xiaowei Chen

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Benjamin Braaten
Chair

Jacob Glower

Danling Wang

Joel Caton

Approved:

June 28, 2018
Date

Benjamin Braaten
Department Chair

ABSTRACT

Memory wall is becoming a more and more serious bottleneck of the processing speed of microprocessors. The mismatch between CPUs and memories has been increasing since three decades ago. SRAM was introduced as the bridge between the main memory and the CPU. SRAM is designed to be on the same die with CPU and stores temporary data and instructions that are to be processed by the CPU. Thus, the performance of SRAMs has a direct impact on the performance of CPUs.

With the application of mass amount data to be processed nowadays, there is a great need for high performance CPUs. Three dimensional CPUs and CPUs that are specifically designed for machine learning are gaining popularity. The objective of this work is to design high performance SRAM for these two emerging applications. Firstly, a novel delay cell based on dummy TSV is proposed to replace traditional delay cells for better timing control. Secondly, a unique SRAM with novel architecture is custom designed for a high-performance machine learning processor. Post-layout simulation shows that the SRAM works well with the processing core and its design is optimized to work well with machine learning processors based on convolutional neural networks. A prototype of the SRAM is also tapped out to further verify our design.

ACKNOWLEDGEMENTS

I am very appreciative of Dr. Benjamin Braaten who offered me great help in the process of looking for a job and getting ready for the defense. Without him, neither would be possible. I would also like to thank Dr. Jacob Glower, Dr. Danling Wang and Dr. Joel Caton for their willingness to serve on my supervisory committee. Their help and guidance are deeply appreciated.

I would especially like to thank Dr. Jinhui Wang, Dr. Na Gong, Seyed Alireza Pourbakhsh, Hritom Das, Jingyan Fu, and Yifu Gong, for their contributions in this study, and their constant encouragement and support.

This work was supported by the National Science Foundation, under Grant CCF-1514780.

DEDICATION

Dedicated to:

My parents for their support.

Dr. Mausbach for the unconditional help.

My Spiritual mother Liya Zhang for all the help.

My beloved wife Mei Li and my precious daughter Irena Yichu Chen,

whose emotional support keeps me working persistently toward my degree without giving up.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D.....Two Dimensional

3D.....Three Dimensional

BL .....Bitline

CDM .....Charge Device Model

CMOS .....Complementary Metal-Oxide-Semiconductor

CPU.....Central Processing Unit

DRAM.....Dynamic Random-Access Memory

DTDU .....Dummy TSV Delay Unit

EPROM.....Erasable Programmable Read Only Memory

ESD.....Electrostatic Discharge

HBM .....Human Body Model

IC.....Integrated Circuit

ITRS.....International Technology Roadmap for Semiconductors

MM .....Machine Model

MOSFET.....Metal-Oxide-Semiconductor Field-Effect Transistor

NAND.....Negative AND

NMOS.....n-type Metal-Oxide-Semiconductor Field-Effect Transistor

NOR.....Negative OR

Pre .....Precharge

PMOS.....p-type Metal-Oxide-Semiconductor Field-Effect Transistor

PROM .....Programmable Read Only Memory

PTM .....Predictive Technology Model

RAM .....Random Access Memory

ROM .....Read Only Memory

SA .....Sensing Amplifier

SRAM .....Static Random-Access Memory

TSV .....Through Silicon Via

VLSI.....Very Large-Scale Integration

WL .....Word Line

1. INTRODUCTION

Back in 1994, William Wulf and Sally McKee claimed that the memory will soon become the bottleneck of development of microprocessors. Because the processing speed of the microprocessors are steadily increasing at a rate much higher than the speed of memories. Two decades later, this claim has proven true and this has become the so-called problem of memory wall. The memory wall has become a more urgent problem especially because mass volume of information is digitized into big data due to advances in sensor technology, the Internet of things (IoT), social networking, wireless communications and huge collection of data from years and this dramatically changes the landscape of microprocessor workloads. Emerging big data calls for development of high computing ability microprocessors with memories that have high speeds. Thus, active research has been going on to solve the bottleneck of microprocessors caused by memories.

Memories include ROM and RAM. Two of the most common ROMs are PROM and EPROM. As the name implies, ROM is mainly used for reading purpose. Content of ROM is normally written into it before it is installed and cannot be modified after that. ROM is normally used to store the information that is used to power up the system. RAM refers to random access memory which includes DRAM and SRAM. DRAM is commonly used as the main memory of the computer. A basic DRAM cell includes one transistor and one capacitor. The advantage of DRAM is that it is fairly fast and takes small area. Thus, it can be used to to make large storage space that can be used as the main memory at an affordable price. However, DRAM needs periodic refresh to keep the data. Extra control circuit needs to be added to refresh a DRAM. The SRAM keeps the data as long as there is power supply. It does not need to be refreshed like the DRAM. An SRAM cell is comprised of six transistors, which means it takes a lot of silicon area

and is more expensive to fabricate compared to DRAM. SRAM is one of the fastest memories so far. It takes only one clock cycle to access an SRAM cell. Thus, SRAM is used in for applications which requires very high speed like microprocessors. In fact, SRAM is normally used as CPU cache memory, which requires very fast response speed to meet the data read and write requirement of a CPU.

Although there are many types of memories, SRAM is the memory that is used for CPU cache memory which is an extremely important part of modern processors. Cache was introduced to make up the gap between DRAM and CPU. Cache stores the data that is to be processed by computing cores. The ideal case is that whenever data is needed, the cache can be fast enough to feed the data to the computing cores. Otherwise, the computing cores have to stay idle for certain cycles to wait for the data from cache to be ready. Computing cores will not be able to work at their full speeds and a lot of processing power would be wasted. Although SRAM is generally fabricated on the same chip with computing cores to ensure the access speed and it is usually very expensive to fabricate compared to other types of memories, there is an increasing trend in the size and complexity of on-chip SRAM. For example, Motorola processor 68020 was the first processor released in 1984 that was designed with an on-chip cache. The cache on processor 68020 only has a cache of 256 bytes and it is a single level cache. After 34 years the Intel I7 8086k Processor has a three-level cache with a total size of 12 MB. The cache size has increased by 48,000 times in 34 years and it is still showing a steady increasing trend. Research on SRAM has shown that there is great potential to make breakthrough in high performance SRAM and will in turn boost the microprocessor speed. Thus, great emphasis has been placed on research of SRAM access speed.

This dissertation focuses on high-performance SRAM design for two emerging applications. The first emerging application is 3D microprocessor and the second application is a microprocessor for machine learning. ITRS predicts that the conventional 2D transistor density scaling will likely come to an end by 2021, and 3D will be a viable choice for further increasing the transistor density on a chip. So, 3D microprocessors have become an emerging research area in recent years. Also, with the development of big data and deep learning concepts and algorithms, machine learning is having more and more applications in the world today. The calling for high-performance microprocessors that are designed for machine learning is becoming urgent. The dissertation researches on possible approaches to design high performance SRAM for microprocessors that can be applied in two emerging applications. Chapter 2 discusses high-performance SRAM design for a 3D microprocessor. The research was performed from a timing control perspective. Delay cells are commonly used to control the timing of SRAMs on a 3D microprocessor. A novel delay cell based on dummy TSVs is proposed and it proves to be able to optimize the timing control compared to traditional delay cells. Chapter 3 proposes a new high-performance SRAM with novel architecture that is specially designed for machine learning purpose. The SRAM is a prototype, which is designed and taped out to verify the function of the SRAM. Future work will be needed to optimize the design for performance and power consumption.

2. A NOVEL DELAY CELL BASED ON DUMMY TSV FOR 3D MICROPROCESSOR

2.1. Background

Presently, on-chip memories, such as register files and caches in microprocessors, employ as high as 70% of the silicon area. This trend is still increasing and even threatening the development of the high-performance computing [1].

Three-dimensional (3D) on-chip memory saves fabrication costs, improves performance, and increases power efficiency [2] and is becoming more popular due to emergence of 3D microprocessors. The bit-lines have much influence on the 3D on-chip memory performance, because they go along the critical path both for data read/write and timing control [3]. To provide precise timing control/or satisfy timing constraints, delay units are widely used between different levels of bit-lines [4].

Traditional delay units are normally 2D and they are comprised of passive devices like resistors, capacitors, and inductors, or active devices like transistors and diodes [5]. Passive delay units usually are quite bulky and thus not suitable for high density application. Compared to passive delay units, active ones take smaller area, however, they are sensitive to supply voltage variations and process variations, and then tend to generate a less stable delay time. Active delay units based on CMOS technology include inverter chains, dual loop delay units [6], all-pass filters [7], decoders and pass transistors [8], and CMOS thyristors [9]. According to reference [10], as supply voltage scales down to lower than 3 volts, delay variations of CMOS delay units begin to increase rapidly. In above delay units, inverter chains are so far the most popular type [11]. As a result, when a large delay is needed between global and local bit-lines in memory, many inverters have to be cascaded as long buffer chains and occupy a significant portion of the memory circuit, causing high silicon area consumption [12]. In addition, in

nanoscale semiconductor technology, due to aggravated supply voltage and process variations, the traditional inverter chain based delay units are increasingly difficult to produce constant delay times, which will greatly degrade the performance of 3D on-chip memory. Thus, a novel delay cell is needed to deal with the above problems, and to enable 3D on-chip memory to well serve and support development of high performance microprocessors.

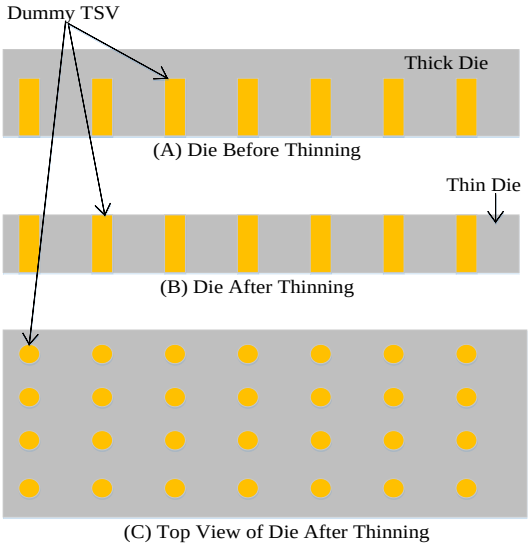


Figure 1. Die Thinning with Dummy TSV as Thinning Ruler. (A) Cross-sectional View of Die before Thinning. (B) Cross-sectional View of Die after Thinning. (C) Top View of Die after Thinning.

In 3D ICs, besides signal TSVs for signal and data transmission, a large number of dummy TSVs are required [13] - [15]. This is because the wafer thinning in 3D IC manufacture is achieved by mechanical grinding or polishing. It removes part of the bulk silicon and leaves an active circuit layer with a thinned substrate [16]. Typical benefits of thinned wafers include package miniaturization, denser stacking option, improved device reliability (due to more flexibility of dies), and thinner vertical stack [16]. Dummy TSVs, as thickness rulers, control a wafer which is shown in Figure 1 (A) to be thinned down to the desired thickness, as illustrated in Figure 1 (B) and (C). Furthermore, dummy TSVs, made by metal with higher thermal conductivity as compared to silicon [17], emphasize the objective of conveying heat rather than

providing signal communication for circuits located on different physical layers [18], and therefore they are often inserted to effectively optimize the thermal performance of 3D ICs. Accordingly, 3D IC physical design rule has included certain density requirements of such dummy TSVs, and even researchers have developed cooling techniques for 3D ICs by optimizing dummy TSV insertions [19], [20].

To sum up:

1) Delay units play a critical role in 3D on-chip memories. Conventional bit-line delay units are inverter chain based and sensitive to supply voltage and process variations as well as area costly.

2) Dummy TSVs have been placed in 3D memory due to thermal and thinning concerns. Dummy TSVs can also be "multifunctional" and play the role of delay units for bit-lines, while still serving their thermal and thinning purposes.

Therefore, the novel hybrid delay unit design using dummy TSVs is proposed to enhance performance, decrease area cost, and increase immunity to supply voltage and process variations for 3D on-chip memories.

As shown in Figure 2 (A), dummy TSV delay units (DTDU) are presented to replace traditional bit-line delay units, which will enlarge the design space and enhance design flexibility of 3D on-chip memories. Main contributions of this project include the following areas:

1) Novel hybrid DTDU structure is proposed to replace traditional delay units in bit-lines of memories with 180 nm [21], and 45 nm CMOS technologies [22].

2) Delay models are derived to estimate the buffer size within each DTDU based on the expected delay time.

3) Performance improvement, power, area, and delay time variation of DTDUs are thoroughly analyzed and compared with traditional inverter chains.

4) Layout of a 128 KB 3D on-chip memory with DTDUs is designed and area improvement of DTDUs is discussed.

5) Thermal analysis of the 3D on-chip memory is performed to verify that the proposed DTDUs enhance the designated ability for heat dissipation.

6) The architecture of 3D on-chip memory is analyzed to show the feasibility of the proposed design.

7) 3D design flow is proposed to implement the DTDUs in 3D on-chip memories.

The rest of the chapter is organized as follows: Section 2.2 reviews existing related works; Section 2.3 introduces research methodology which includes the circuit model of a dummy TSV, the proposed DTDU structure, and the derivation process of buffer size prediction models; Section 2.4 firstly presents the verification of models proposed in section 2.3, secondly compares the performance, area, power, and voltage and process variation of DTDUs with traditional inverter chains, thirdly analyzes the architecture and thermal performance of the 3D memory with DTDUs and proposes a 3D design flow, and finally compares the proposed work with the existing research; and section 2.5 concludes the chapter.

2.2. Related Work

Many researchers have studied on making use of dummy TSVs, but to our best knowledge, no existing research focus on using dummy TSVs for bit-line delay units. However, there are related research focusing on the timing characteristics of dummy TSVs. In this section, these related research works are reviewed.

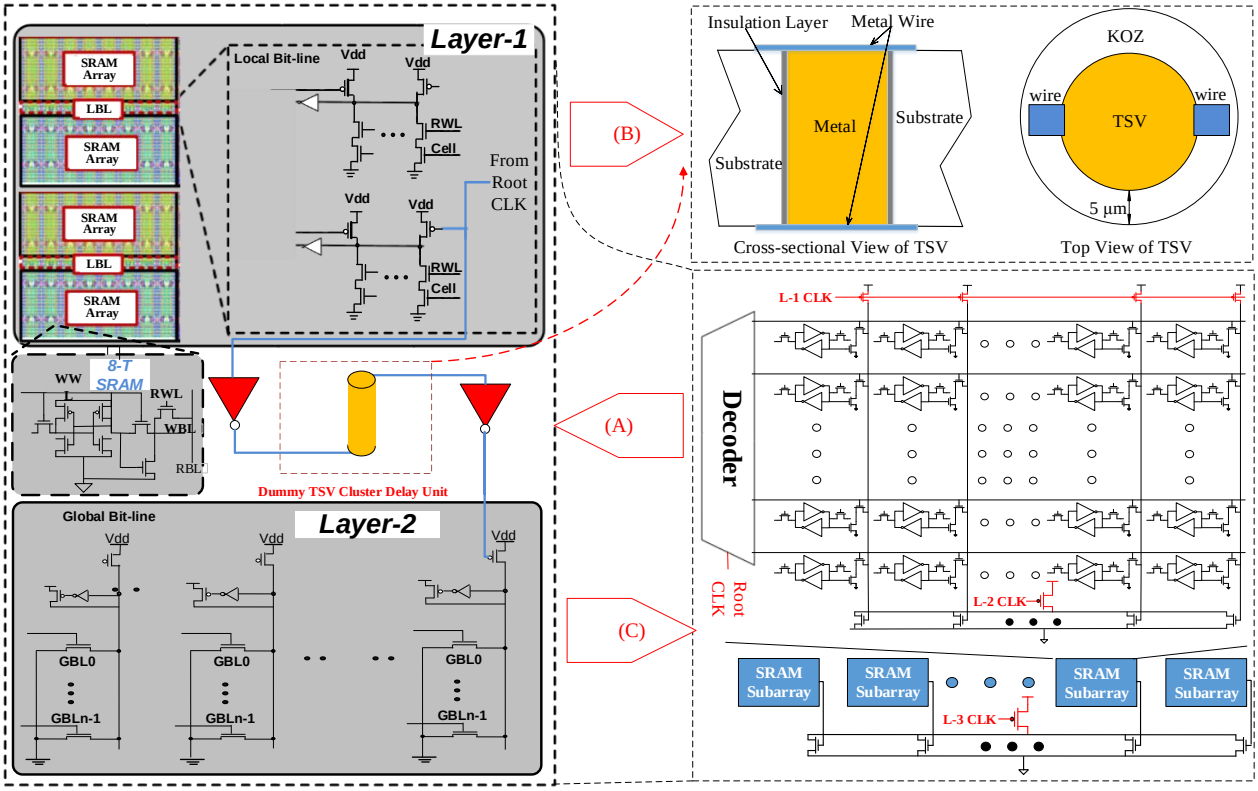


Figure 2. 128 KB Memory and DTDU (A) 3-D On-chip Memory with DTDU. (B) Dummy TSV Cross-sectional and Top Views. (C) Memory Architecture.

For the timing perspective for TSVs, researchers mainly focus on the following areas:

1) Delay models: Some research groups consider TSVs as resistor and capacitor networks and build delay models for the signal propagation. In [23], a delay model for evaluating vertical signal propagation delay in TSVs is developed, while parasitic capacitance is taken into consideration. In [24], the parametric analysis of TSV capacitance is performed based on several physical and material parameters, and an accurate electrical model of a single TSV considering metal-oxide-semiconductor (MOS) capacitance effect is developed. However, neither of them try to put their models into functional circuits nor use a TSV as a delay unit.

2) TSVs in clock trees: Some researchers use TSVs to build 3D clock trees in 3D ICs. For example, in [25], the impact of TSV count and TSV resistance-capacitance parasitic on low-power and low-slew clock network is analyzed. In [26], a white-space TSV arrangement

algorithm is designed for 3D clock-tree synthesis. Meanwhile, reference [27] attempts to cope with the reliability problem of clock TSVs, and presents a new circuit cell called slew-controlled TSV fault-tolerant unit. However, all of them focus on how to reduce delay time as signal TSVs, and none of them have paid attention to the potential of using existing TSVs and their delay characteristics as beneficial unit in clock trees.

3) Timing assistance using dummy TSVs: In [28], dummy TSVs for thermal reduction are allocated to decrease clock-skew under non-uniform temperature distribution and thus increase clock stability. In [29], the sensitivity analysis is performed for dummy TSV density to enable a reliable 3D clock tree for balancing the clock skew by reducing temperature and stress gradient. Unfortunately, both [30] and [31] utilize the thermal conductivity of dummy TSVs to optimize the clock trees in the whole 3D chip and dummy TSVs only keep inherent mechanical purpose without any electrical feature involved.

In this project, dummy TSVs are extended from mechanical function to electrical connection. Especially, recent research has indicated that dummy TSVs could be multi-functional as a timing path, thermal path, and thinning ruler simultaneously [30] [31]. The proposed technique in this project is to make use of dummy TSVs as bit-line clock delay control in 3D on-chip memory.

Compared with existing studies, it has the following advantages:

1) DTDUs can increase the maximum memory frequency as much as inverter chains based delay units can.

2) DTDUs save considerable silicon area compared with traditional inverter chains.

3) DTDUs are less sensitive to supply voltage and process variations and thus provide more constant delay time and enhance memory reliability compared to traditional inverter chains.

4) When the dummy TSVs are applied for delay units, they can simultaneously serve for thinning and thermal purposes.

2.3. Methodology

In this section, the RC model is used for a dummy TSV to study its delay characteristics; a 128 KB memory is built and its delay mechanism is discussed; a universal 3D DTDU structure is proposed and mathematical delay models for level-1 (L-1) DTDU, level-2 (L-2) DTDU, and level-3 (L-3) DTDU are derived to make a prediction on buffer sizes based on the given delay time.

2.3.1. TSV Model

TSVs, which are used to interconnect different layers in 3D ICs, play an important role in providing power, clock, and functional signals [32]. A typical structure of a TSV is a metal (usually copper or tungsten) cylinder surrounded by an insulator (silicon dioxide), which is used to prevent leakage current and capacitive coupling through the substrate [33], as shown in Figure 2 (B). To investigate the timing characteristics of a dummy TSV, it is modeled as a resistor-capacitor (RC) network. Each TSV has a parasitic capacitance (CTSV) and a resistance (RTSV), and their capacitance and resistance are determined based on the TSV dimension, material, and manufacturing process [34].

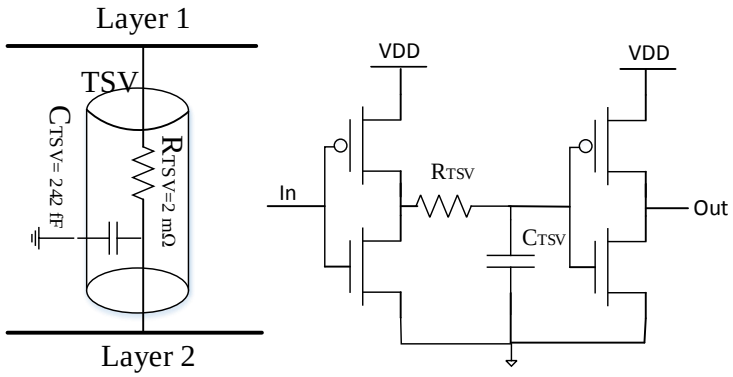
Various sizes of TSVs are used in academia and industry. One of the popular sizes is chosen based on the TSV model developed in [34] - [36]. The TSV has a diameter of 10 μm, an oxide thickness of 1 μm, and a height of 75 μm. Accordingly, the resistance (RTSV) and

capacitance (CTSV) are 2 mΩ and 242 fF, respectively [33]. Equivalent circuits of a TSV and a DTDU are shown in Figure 3. (A) and (B). TSV coupling effect can be ignored since the TSV pitches in this project is larger than 120 μm [37].

TSV fabrication causes tensile mechanical stress around them due to mismatch between coefficients of thermal expansion (CTE) of silicon and copper, thus damage can happen to devices around TSVs. So, it is necessary to have a keep out zone (KOZ) around TSVs where no transistors can be placed [38]. Size of KOZ used in this project is 5 μm.

To obtain the delay time generated by a dummy TSV, in addition to inherent resistance and capacitance of TSV, the wires, connecting TSVs in KOZ, are also taken into consideration as part of total resistance in the RC model R_{Total}, as expressed in (1):

$$R_{Total} = R_{TSV} + R_{wire} \tag{1}$$



(A) (B)

Figure 3. (A) TSV Model. (B) DTDU Equivalent Circuit.

The wire resistance is estimated based on wire length, width, and sheet resistance which is provided by foundries and previous studies [39]. Wire length is two times KOZ size as shown in Figure 2 (B).

The proposed DTDU comprises of buffers and one dummy TSV, as shown in Figure 2 (A). Buffers are necessary and serve as signal drivers in a DTDU. This is because clock signals

fail to reach enough high or low voltage after propagating through dummy TSVs without buffers driving when memory is operating at high frequencies. The delay time generated by DTDU in the circuit can be adjusted by changing buffer sizes. Vice versa, given an expected delay time of a DTDU, buffer sizes can be determined.

2.3.2. On-chip Memory Architecture

Figure 2 (A) and (C) show the DTDU based memory architecture. Taking a memory with a size of 128 KB as an example, each bank covers 32×32×32 bits conventional 8-T SRAM cells. One bank is divided into 32 arrays, each of which is divided into 32 sub-arrays. In each sub-array, 5-bit read and write address signals are decoded to generate 32 read and write select signals [40], and each reading bit line (RBL) connects to 32 cells in a column. And they are controlled by level-1 clock (L-1 CLK) as the first level local bit lines (L-1 LBL). All L-1 LBLs are then connected to an n-input dynamic gate controlled by level-2 clock (L-2 CLK) to combine into one second level local bit lines (L-2 LBL). Accordingly, each subarray comes out one L-2 LBL [41] [42].

Further, 32 second level LBLs converge into one global bit line (GBL) controlled by level-3 clock (L-3 CLK), which is 1-bit output.

Table 1. Expected delay time of memories in two technologies (ps)

Technology node	Memory 1		
	L-1 DU	L-2 DU	L-3 DU
180 nm	216	178	165
45 nm	59	25	23

2.3.3. Uniform Structure of DTDU

As shown in Figure 4, the decoder enables signal (EN) is activated by root clock (RTCLK) at first, and as soon as word line is active, L-1 CLK is turned on to start the reading evaluation process on L-1 LBL. When reading evaluation process is finished on the L-1 LBL, the second level dynamic gate is turned on by L-2 CLK. Finally, L-3 CLK is turned on to start the evaluation process on GBL after evaluation is completed on L-2 LBL. To optimize the timing control, L-1 DTDU, L-2 DTDU, and L-3 DTDU are respectively inserted between EN and L-1 CLK, L-1 CLK and L-2 CLK, L-2 CLK and L-3 CLK, as shown in Figure 4 (A). Accordingly, the expected delay time for each delay unit is determined as follows. For L-1 DTDU, the expected delay time is from EN is ready to L-1 LBL is ready for evaluation. For L-2 DTDU, the expected delay time is from the L-1 LBL is ready for evaluation to the L-2 LBL is ready for evaluation. For L-3 DTDU, the expected delay time is from L-2 LBL is ready to GBL is ready for evaluation. Memories have the maximum speed with such expected delay times. Either shorter or longer delay time would decrease the maximum speed of the memory. As a result, delay times needed for L-1 DTDU, L-2 DTDU, and L-3 DTDU have been obtained.

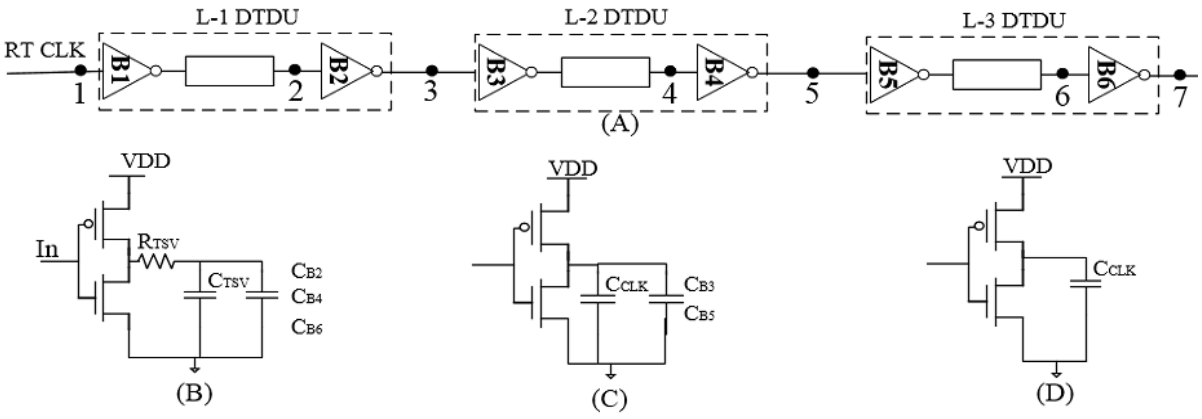


Figure 4. DTDU Structure and Equivalent Circuit of Buffers. (A) Uniform DTDU Structure. (B) Equivalent Circuit of B1, B3, and B5. (C) Equivalent Circuit of B2 and B4. (D) Equivalent Circuit of B6.

DTDUs, which are used for L-1 DTDU, L-2 DTDU, and L-3 DTDU, are networks of buffers and dummy TSVs. Figure 4 (A) shows DTDU structures used for memories. The width of buffers is the variable that is used to control the generated delay time.

To verify the proposed DTDU scheme, memories using 180 nm and 45 nm technology nodes are developed.

Table 2. Buffer size in each DTDU for memories of two technology nodes.

Technology nodes		B1 and B2 Size	B3 and B4 Size	B5 and B6 Size
180 nm	NMOS Width (μm)	4.5	5.8	4.3
45 nm	NMOS Width (μm)	4.2	4.7	3.6

2.3.4. Delay Time and Buffer Size Prediction Model

The expected delay time required by L-1 DTDU, L-2 DTDU, and L-3 DTDU for memories at two technology nodes are shown in Table I. The technology node and memory size have a positive relationship with delay time. This is because larger technology node has larger loading capacitances, which take longer time to charge and discharge. The longest delay time is 216 ps which is delay time of L-1 DTDU for180 nm technology and the shortest delay time is 23 ps which is delay time of L-3 DTDU for 45 nm technology.

Buffer sizes used in this chapter refer to width of the NMOS used in the buffer, and width of PMOS is always twice the width of the NMOS in the same buffer. Lengths of all transistors in DTDU buffers for 180 nm and 45 nm are the minimum sizes allowable in design rules, respectively.

Buffer size prediction models and their derivation process can be found in Appendix A.

2.4. Implementation Results

2.4.1. Buffer Size Prediction

Memories using 180 nm and 45 nm are used to verify the accuracy of those models. Each simulation result is compared with the corresponding value calculated from L-1 DTDU, L-2 DTDU, and L-3 DTDU delay models.

As shown in Figure 5, the max errors between the simulated delay time and calculation from delay time models are 8.9% and 8.7% respectively. Therefore, these models can be used to calculate buffer sizes of B1 to B6 for technologies from 180 nm to 45 nm with the accepted accuracy. Buffer sizes of these two memories are listed in Table II.

2.4.2. Maximum Frequency Improvement

Figure 6 is a simple timing diagram of root CLK and L-1 CLK during a read operation that illustrates how delay units can help improve the max frequency in a memory. Suppose the root CLK has a pulse width of P and a period of 2P. When root CLK sends an enable signal, decoder begins to work and wordline will be activated and data will be available on L-1 bitline from T1 to T2 which equals P. When the required delay time is added between root CLK and L-1 CLK, L-1 CLK is activated from T1 to T2. Data evaluation starts when data is available and L-1 CLK is enabled, thus effective evaluation time Te1=T1-T2=P. If inadequate delay time is added, L-1 CLK is activated before data is ready on the bitline and deactivated while data is still available on bitline. The effective evaluation time Te2 is smaller than P. If over delay time is added, L-1 CLK gets activated after the data is ready on bitline and gets deactivated after the data becomes unavailable, and the effective evaluation time Te3 is also smaller than P. Since there is a minimum time requirement for data evaluation called Te_min, the minimum value for

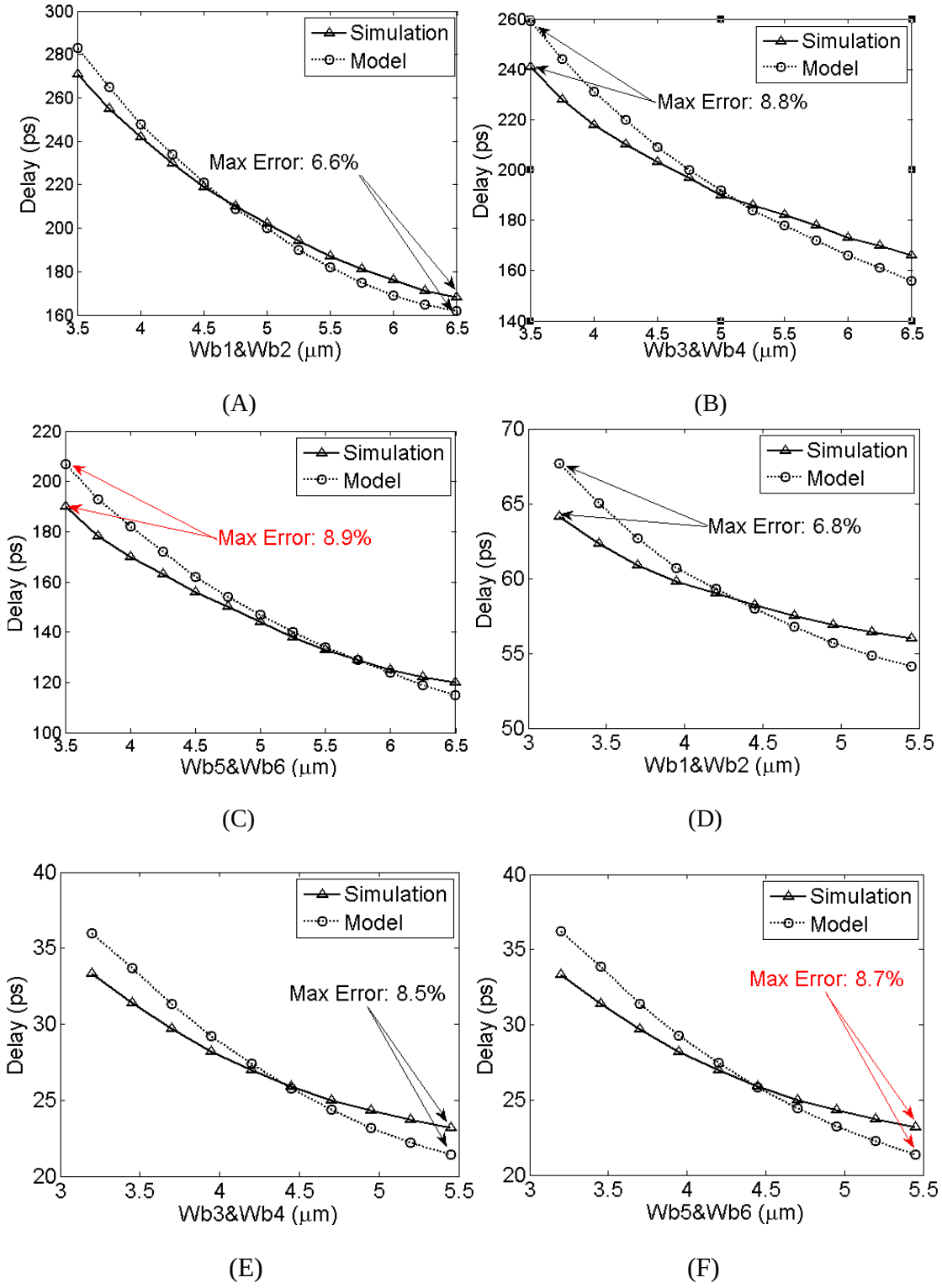


Figure 5. Comparison between simulation and model in 180 nm and 45 nm Technology. (A) Simulation vs L-1 Delay Model in 180 nm. (B). Simulation vs L-2 Delay Model in 180 nm. (C). Simulation vs L-3 Delay Model 180 nm. (D) Simulation vs L-1 Delay Model in 45 nm. (E). Simulation vs L-2 Delay Model in 45 nm. (F). Simulation vs L-3 Delay Model 45 nm.

P is P_min=Te_min when the required delay time is added. For example, assume P_min=Te_min=1ns, and the period would be 2ns and max frequency would be 500 MHZ. If inadequate delay time is added, suppose L-1 CLK is activated 500 ps ahead of T1, since evaluation has to be on for at least 1 ns, L-1 CLK cannot be deactivated till T1+1 ns. In this case, P_min=1.5 ns and period would be 3 ns, which means max frequency can only be 333 MHZ.

Simulation results show that frequency improvement of the memory with inverter delay units is 110.1% and with DTDUs is 110.7%. DTDUs is comparable as inverter delay cells to increase the maximum frequency in a memory, which indicates it is reliable to replace traditional inverter delay chains.

2.4.3. Delay Time Variation

Delay time variations caused by both supply voltage variations and process variations are discussed in this section.

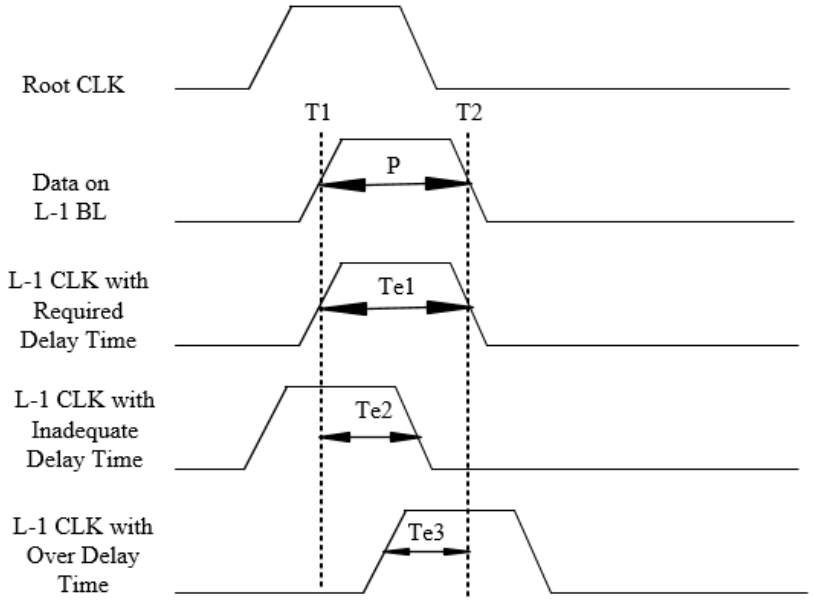


Figure 6. Timing Diagram of Mechanism of Delay Units.

The supply voltage variations in the memory result from a large number of parasitic elements in circuits, which can be modeled as a RLC model between the core circuit and supply

sources [45]. In synchronous circuits controlled by a system clock signal, a big number of transistors switch on/off at the same time. This results in an instantaneous large current in the substrate and interconnection, which causes fluctuations in VDD and GND, when such current pass through parasitic RLC circuits. Subsequently, these fluctuations change the delay time produced by traditional inverter chains.

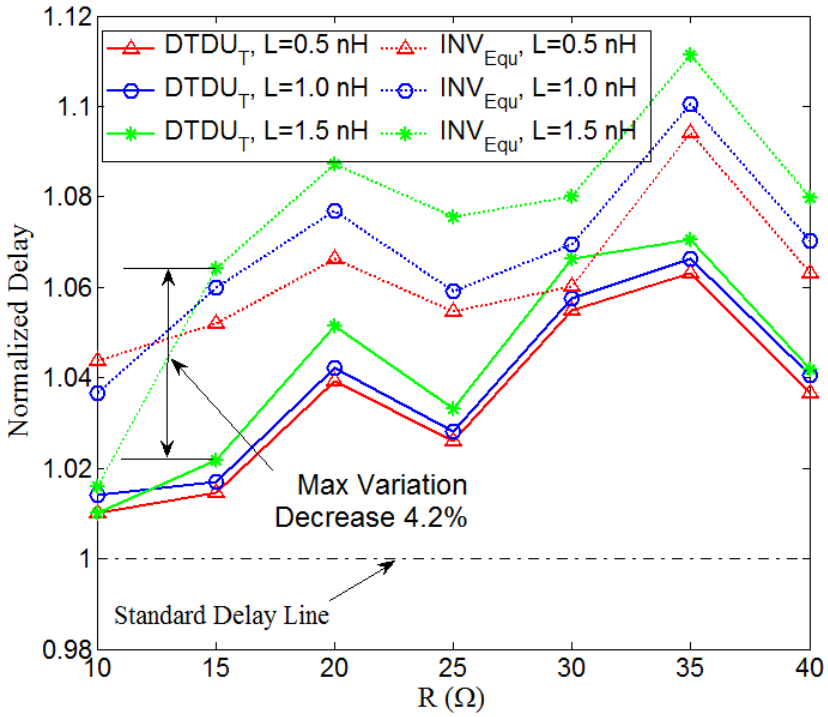
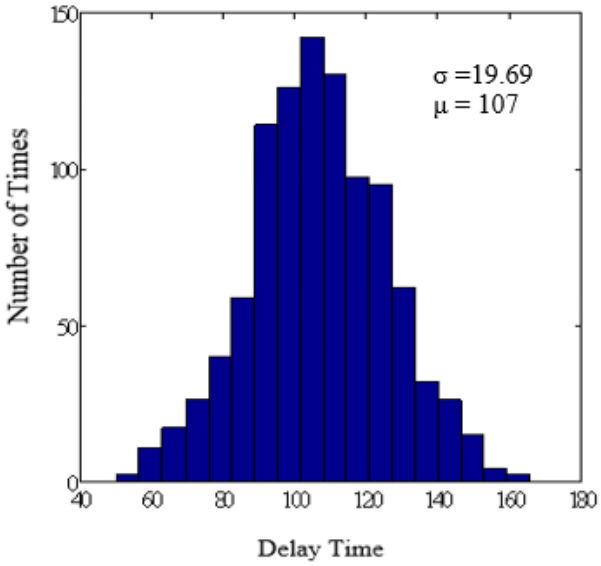


Figure 7. Time variation comparison between inverter chain and DTDU. Parasitic capacitance is set to be 1 pF.

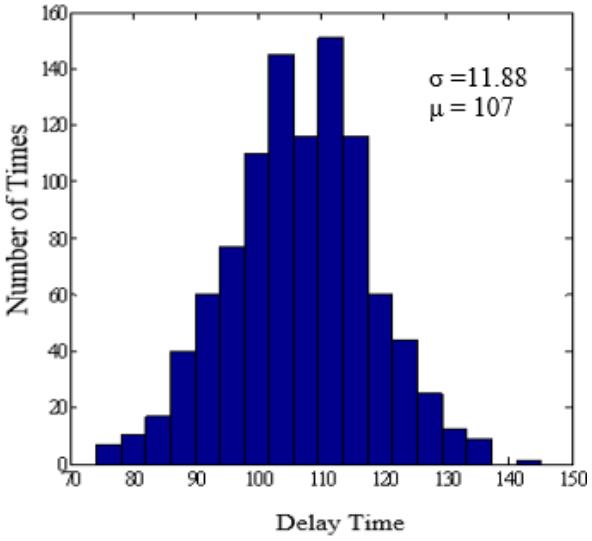
As shown in Figure 7, an inverter chain that has the same delay time with a DTDU is designed, and total delay time generated by DTDUs in a memory is DTDUT, whose equivalent inverter chain is INV_{Equ}. They are compared under same VDD and GND voltage fluctuation. Three sets of data are obtained while parasitic inductance L is 0.5 nH, 1.0 nH, and 1.5 nH [45]. Parasitic capacitance is set to be 0.1 pF. As parasitic resistance increases, three DTDUT are always smaller than their counterpart INV_{Equ}, and the maximum difference is 4.2%.

As technology scales down, process variation is becoming a much more critical issue that affects circuit reliability and performance. There are various types of process variations. For 2D inverter delay cells, delay variations result from intra-die variations and open circuit failures. According to [46], for intra-die process variations, transistor length and oxide thickness variation are two main variation sources and they demonstrate a normal distribution with variations of 1.7%. CMOS circuits have an open circuit failure of one in a billion [47], which can be ignored for the inverter delay cells. Finally, 1000-time Monte Carlo analysis for process variations are performed for 2D inverter chains and the result is, shown in Figure 8 (A), delay variation (standard deviation (σ) /mean value (μ)) is 18.4%.

For 3D structure DTDUs, the worst case is that buffers on different layers are fabricated separately with different wafers. As a result, buffers in DTDUs suffer both intra-die variations and inter-die variations. Transistor length and oxide thickness show a normal distribution with 1.7% and 0.5% for inter-die and intra-die variations [46]. Process variations of dummy TSVs are also taken into consideration. Dummy TSV process variations originate from two sources [48]: 1) During fabrication, TSVs may even have voids, and this results in an average of 8.6% process variations; 2) TSVs suffer from open circuit and short circuit failure, which happens when the metal filling is inaccurate or misalignment occurs between TSVs and metal wires [49]. According to [50], TSVs open circuit and short circuit failure rate is lower than 0.1%. Then, 1000-time Monte Carlo analysis, taking account of buffer inter-die and intra-die variations, TSV process variations, and open circuit and short circuit failure variations, is performed and the result is shown in Figure 8 (B). Delay variation (standard deviation (σ) /mean value (μ)) is 11.1%. DTDUs have a delay variation 39.7% lower than inverter delay units.



(A)



(B)

Figure 8. (A). Distribution of Delay Time Variation of Inverter Delay Cells. (B). Distribution of Delay Time Variation of DTDUs.

According to the Monte Carlo simulation result, inverter delay cells have more delay variations than DTDUs. The CMOS inverter fabrication process involves oxidation, ion implantation, lithography and chemical mechanical planarization, and Optical Proximity Correction. The limitations of the control of each of these processes will cause variations in

transistor parameters like device carrier mobility (μ), threshold voltage (V_{th}), gate oxide capacitance (C_{ox}), and transistor dimension (W and L). Variations in these device parameters will result in large distribution of delay time. What is more, inverter delay cells are cascaded into many stages, and the delay variation propagates through each stage and accumulates as the number of stages increase. While TSV fabrication mainly involves etching and metal deposition, which has fewer processes and fewer sources of variations of TSV than that of inverters, resulting in less variations which are mainly as TSV diameter and length. Also, DTDUs has much fewer delay stages compared with inverter delay cells. As a result, DTDU whose delay time is dominated by one TSV has fewer delay variations than longer inverter delay cells.

2.4.4. Layout, Area, and Power

For the layout, area, and power of 3D memories, the 45nm technology is taken as an example for analysis. In memories, word-lines are evenly partitioned onto layer-1 and layer-2, so there are 64 KB memory cells on each layer. Row decoder is on layer-1, L-1 CLK is on layer-2, L-2 CLK is on layer-1, and L-3 CLK is on layer-2.

Dummy TSVs are placed at convenient locations for DTDUs by custom design and the area overhead of DTDUs is caused by inserted buffers. Figure 9 demonstrates layouts of layer 1, 2, 3, and DTDUs that are used in each bank, which include 6 buffers (B1 to B6). Parts of L-1 DTDU is also shown in the figure. All DTDUs have an area of 65.17 μm^2, while inverter chains have the same drivability and delay time take an area of 116.0 μm^2. So DTDUs have an area saving of 43.8% compared to traditional inverter chains.

Since modern SoCs tend to have distributed memory of various sizes. Besides the 128 KB memory, two smaller memories of 4 KB and 128 B are also discussed. The 4 KB memory has 32 blocks which has 32×32 bits while the 128 B memory has one block. As a result, two

stages of DTDUs are needed for the 4 KB memory and one DTDU is needed for the 128 B memory. The expected delay time of L-1 DTDU and L-2 DTDU in the 4 KB memory and L-1 DTDU in the 128 B memory are 56 ps, 23 ps, and 54 ps respectively. Simulation results show that compared to inverter delay units, layout benefit of the 4 KB memory and the 128 B memory are 40.8% and 39.1% respectively. Therefore, as memory size goes down, DTDUs nearly keep the same area saving as compared to inverter delay units.

Due to the additional power needed for driving TSVs, a 128 KB memory with DTDUs shows a power overhead as little as 3.6% as compared to that with inverter chains, accordingly it is almost negligible.

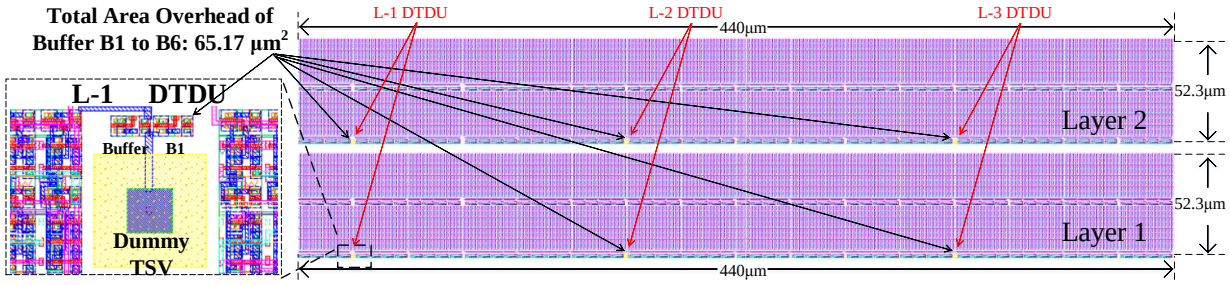


Figure 9. Layout of One Bank of the 3D Memory with Three DTDUs.

2.4.5. Architecture

The proposed 3D on-chip memory includes two layers, the 128 KB 3D on-chip memory is divided into 32 banks as shown in Figure 10 (A), and each bank is then divided into 32 arrays as shown in Figure 10 (B). Then each array is comprised of 32 sub-arrays (Sub-A) as shown in Figure10 (C), and each Sub-A includes 32×32 SRAM cells. Three DTDUs are employed in each bank. Wordline partition technique is applied and 16×32 cells are on layer 1 while the other 16×32 cells are on layer 2. For example, for wordline 1, SRAM cell 1 (SRAM 1-1) to SRAM cell 16 (SRAM 1-16) are on layer 1, and SRAM cell 17 (SRAM 1-17) to SRAM cell 32 (SRAM 1-32) are on layer 2, as shown in Figure 10 (D). 3D wordline partition strategy is used to divide

and map wordlines onto different layers. In this way, capacitive loads on partitioned wordlines are smaller and the time to drive the wordlines decreases. Also, the delay from decoder to SRAM cell wordlines which is the bottle-neck of high performance of on-chip memories is greatly reduced.

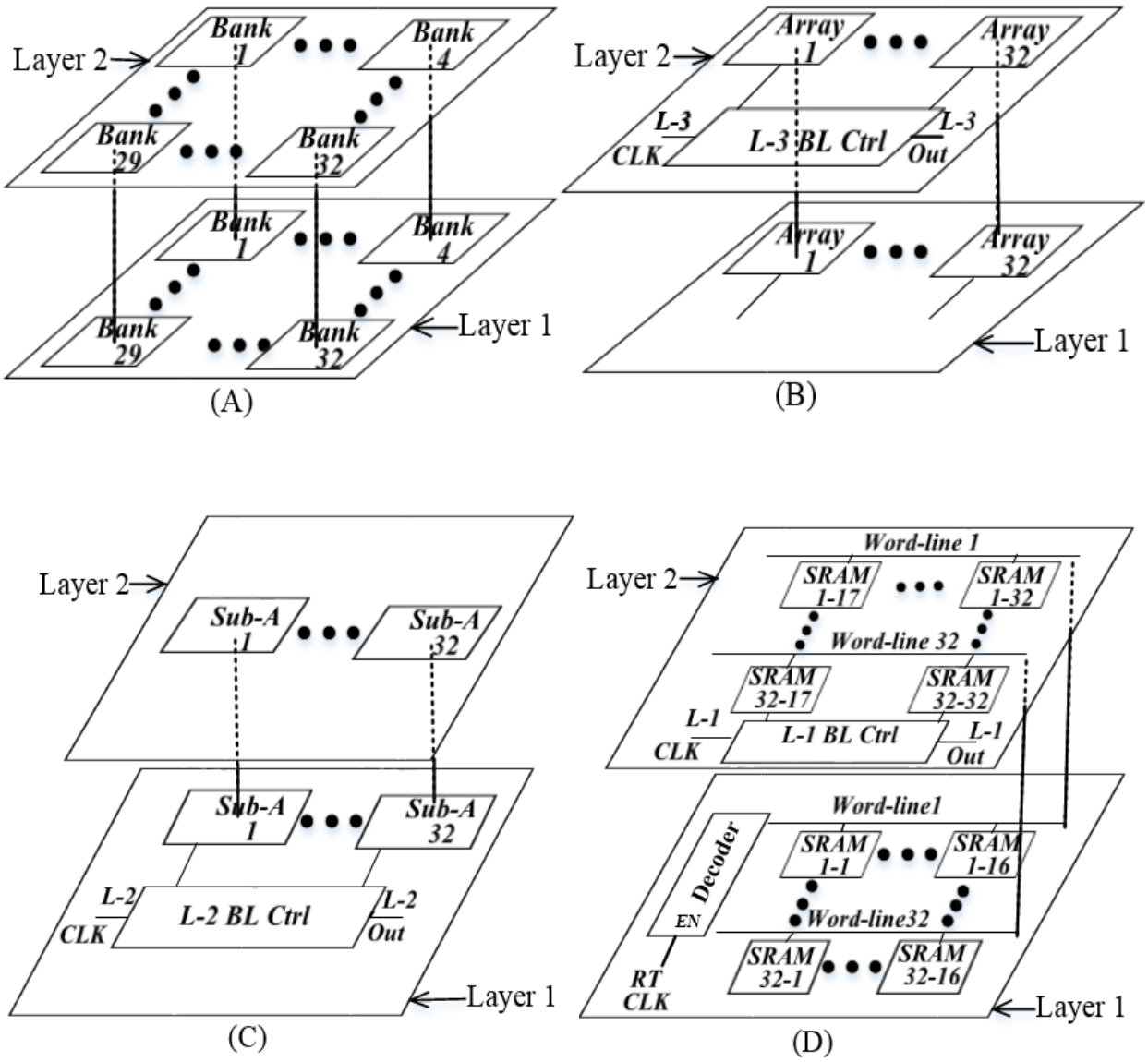
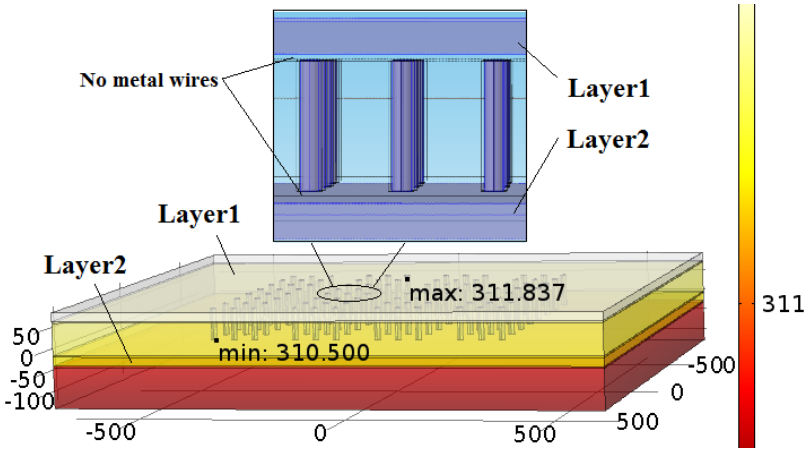
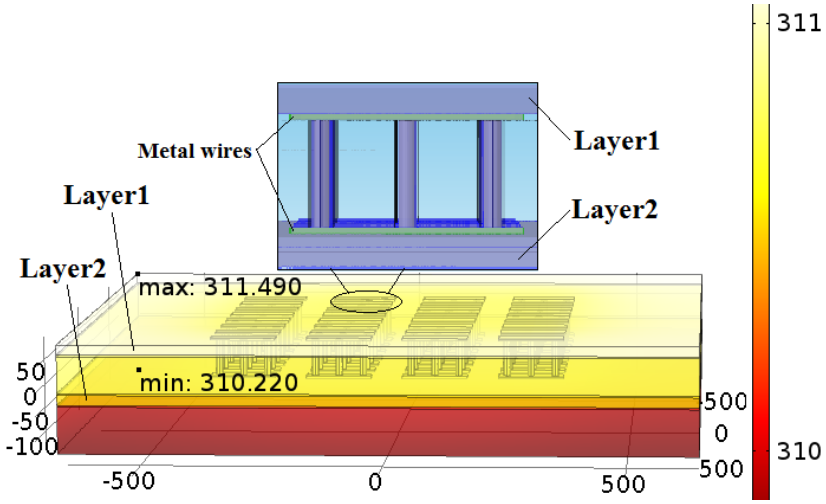


Figure 10. Architecture of 128 KB 3D Memory. (A) Architecture of the whole memory. (B) Architecture of a Bank. (C) Architecture of an Array. (D) Architecture of a Sub-array.

2.4.6. Thermal Analysis



(A)



(B)

Figure 11. Thermal condition of 3D Chips when dummy TSVs are not as DTDUs and when dummy TSVs are used as DTDUs. (A) When dummy TSVs are not used for DTDUs, max temperature on chip is higher. (B) When dummy TSVs are used for DTDUs, max temperature on chip is lower.

To verify the dummy TSVs in DTDUs do not degrade the thermal performance of the proposed 3D on-chip memory, thermal analysis is performed using COMSOL Multiphysics. The proposed 3D on-chip memory is modeled in terms of size ratio, structure, and material. In the 3D thermal model, 96 dummy TSVs are used in the 128 KB 3D on-chip memory, all of them connected by metal wires. Simulation is performed at a power density of 10 Watts/cm2 for the

chip where dummy TSVs are not used as delay units [51], as shown in Figure 11 (A). The maximum temperature of dummy TSVs when they are used for thermal purpose is 311.837 K.

Since dummy TSVs incur a power overhead of 3.6% when used as delay units, power density for the chip where dummy TSVs are used as delay units is 10× (1+3.6%) Watts/cm2, as shown in Figure 11 (B). The maximum temperature of dummy TSVs is 311.49 K. So, applying dummy TSVs for DTDUs positively affects the heat dissipation of an 3D IC. This is because when dummy TSVs are used in DTDUs they are connected by metal wires which perform as heat sinks. Metal wire connections help dissipate part of the heat from dummy TSVs and thus reduce the temperature. In the proposed 3D on-chip memory, 96 dummy TSVs are connected in DTDUs to reduce the maximum temperature from 311.837 K to 311.49 K. While modern on-chip memories are usually much bigger in sizes, so many more dummy TSVs could be used, which could further reduce the temperature.

2.4.7. 3D Design Flow

According to [52] and [53], there are few design tools specially developed to synthesis TSV-based 3D circuits for public use. This is because as reference [54] points out, 3D circuits synthesis is suffering from uneven distribution of TSVs, where there are some areas with extremely high TSV density and some areas with very low TSV density. This may cause reliability problems and thermal problems. An alternative way is to use 2D design tools to synthesis each layer separately or some prototype 3D design tools to synthesis some blocks of the 3D circuit and then make extra efforts to combine different layers or different blocks [52] [53]. For example, the logic elements and controller are synthesized by 2D design tools while its memory part is designed in a full custom way [53]. This design approach is taken for two reasons: 1) Memory blocks have comparatively low TSV density, thus it is easy for TSV

placement and it simplifies the work by custom design; 2) Custom design ensures TSVs can be placed at desired locations and this optimizes the performance of memory blocks.

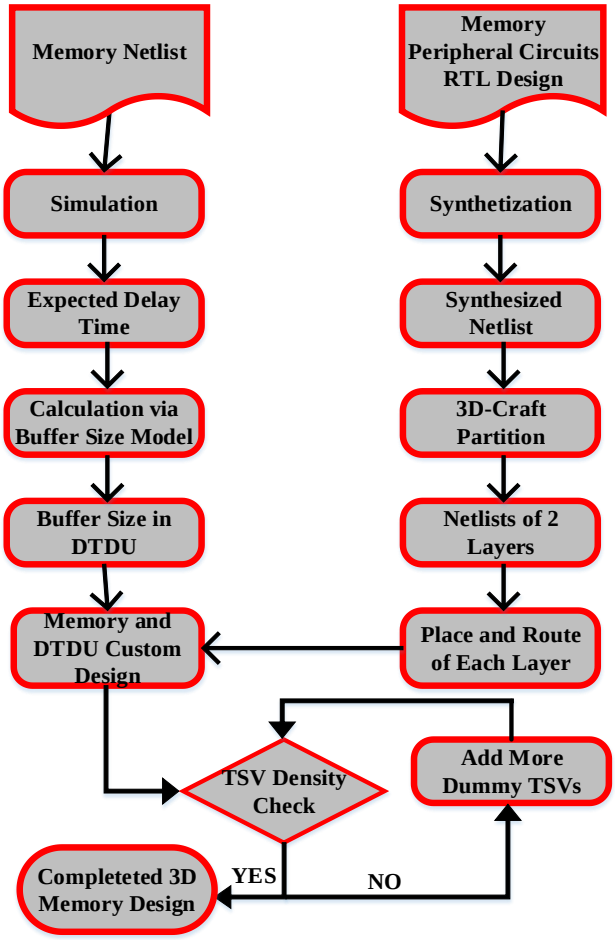


Figure 12. 3D Design Flow for 3D Memory with DTDUs.

A similar design flow is adopted for the proposed 3D memory with DTDUs, as shown in Figure 12. The design flow starts with memory transistor level netlist and a RTL description of memory peripheral circuits, delay time needed for different levels of bit-lines is obtained through simulation of the memory netlist. Sizes of buffers driving TSVs in DTDUs could be calculated from the closed form models proposed in Section 2.3.4. Then the RTL file will be synthesized and the partition of the circuit among layers is completed by 3D-Craft [55]. The following work is that placing and routing will be completed by 2D design tools. Memories and DTDUs will be

designed in a full custom. In this way, dummy TSVs can be placed in a convenient location for DTDUs and little extra wire routing is necessary to connect DTDUs. The last step is to check TSV density to satisfy density requirements. Dummy TSV density should be at least 0.09% according to NCSU, for mechanical reliability concern. But it is recommended to be 0.5% for thermal concerns [56]. If density rules are not satisfied, more dummy TSVs need to be added through custom design. Otherwise, the 3D design flow is finished.

2.4.8. Comparison with the Existing Research

In this section, the proposed DTDU is compared with 2 passive delay units and 2 active delay units and 1 hybrid delay units, with results shown in Table III. Passive delay units are too costly to be integrated into 3D memory chips. As shown in Table III, reference [57] proposes a passive delay cell with inductor and capacitor which achieves a delay time variation smaller than 6.6%. Reference [58] proposes another delay cell of inductors and capacitors which can keep the delay variation as low as 5%. However, the area cost is several hundred times larger than the area of DTDUs. Active delay units are much smaller in size. However, they are more sensitive to supply voltage and process variations. Reference [59] proposes an active delay unit using transmission gates and reference [60] proposes a cascaded inverter delay. They have similar power consumption like DTDUs, but they suffer delay time variation as high as 30.3% which is much higher than delay time variation of DTDUs. Reference [61] also proposes a hybrid delay unit composed of CMOS and capacitors. It has delay time variations as low as 1.8%, however, the area cost of it is about 100 times of the DTDU. Finally, all 5 delay units are designed for 2D circuits while our work has a 3D structure and the existing dummy TSVs are employed to benefit 3D memory. And, the proposed DTDU achieve a trade-off between passive and active delay units and is a good solution for 3D on-chip memory.

Table 3. Comparison between our work and existing delay unit.

	Passive		Active		Hybrid	
Reference	[57]	[58]	[59]	[60]	[61]	Our work
Delay cell	Inductor and Capacitor	Inductor and Capacitor	Transmission Gate	Cascaded Inverters	CMOS and Capacitor	Inverter and dummy TSV
Area (mm^2)	0.82	0.76	N/A	N/A	0.15	0.000065
Delay Variation	<6.6%	5%	15.6%	30.3%	1.8%	11.1%
Power/Delay (μw/ns)	330	N/A	167.9	121.4	N/A	160.7
Technology node	250 nm	130 nm	22 nm	16 nm	140 nm	45 nm
Dimension	2D	2D	2D	2D	2D	3D

As technology scales down, delay time needed between bitlines gets smaller and fewer stages of cascaded inverters will be needed. As a result, inverter delay units would take less area. Similar for DTDUs, dummy TSVs sizes are also scaled down with technology development, so sizes of buffers needed to drive dummy TSVs decrease, as listed in Table II. Thus, there is not expectation for a significant change in DTDU area and power over inverter delay units. However, because delay variation of active delay units increases rapidly as feature size become smaller [14], the hybrid delay unit - DTDUs will have much more stable delay time [58].

2.5. Summary

In this project, a 3D hybrid DTDU is presented for 3D memory bit-line timing optimization. Our technique offers an innovative way to extend dummy TSVs from mechanical function to electrical connection. DTDU is mathematically modeled to find the appropriate

buffer size. In addition, thermal analysis of the 3D on-chip memory is performed to verify that the proposed DTDUs enhance the designated ability for heat dissipation; the architecture of 3D on-chip memory is analyzed to show the feasibility of the proposed design; a 3D design flow is discussed for the 3D memory so that dummy TSVs can be planned at convenient locations. Finally, comparison between our proposed work and traditional delay units are performed and results show that:

1) DTDUs save as much as 43.8 % silicon area compared with traditional active delay units and take an area several orders smaller than passive delay units.

2) DTDUs improve the maximum frequency that a memory can run as much as inverter delay units.

3) DTDUs can achieve a delay variation that is as low as 1/3 of traditional active delay units.

4) DTDU have slightly better heat dissipation ability than traditional delay units.

To sum up, the proposed hybrid DTDU have both advantages of passive delay units and active delay units, and it proves to be a good trade-off of area, power, and immunity to supply voltage variations and process variations. Overall, the proposed 3D hybrid DTDU is a good solution for next generation 3D on-chip memory design.

3. HIGH PERFORMANCE ASYNCHRONOUS SRAM DESIGN FOR MACHINE LEARNING

3.1. Why SRAM is Necessary for Machine Learning Processors

SRAM is normally used as Cache memory in a microprocessor and generally takes more than half of the total area. Figure 13 shows the layout of Intel Core i7-3960X processor and gives us a general information of how much area cache memory takes.

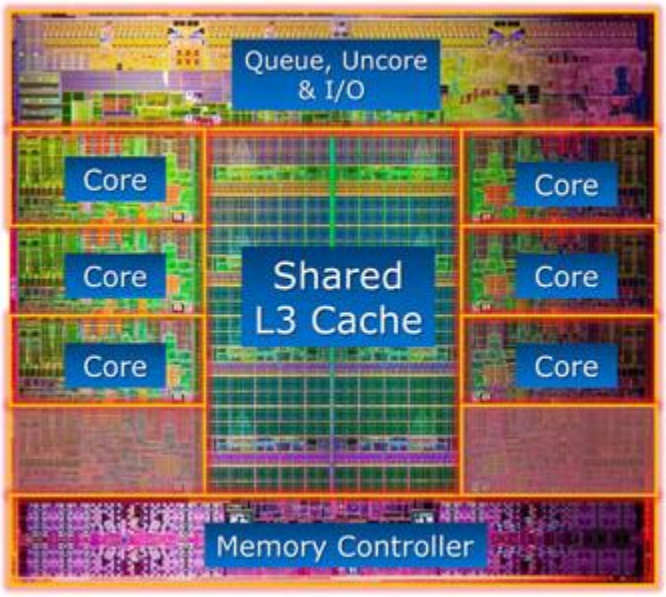
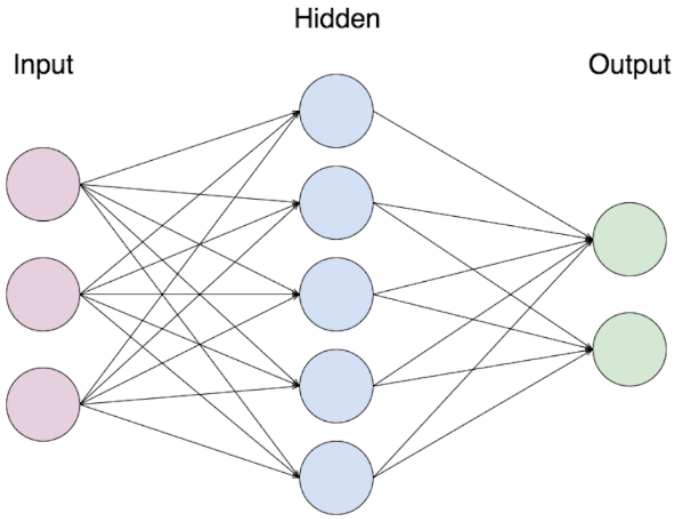


Figure 13. Intel Core i7-3960X Processor[1].

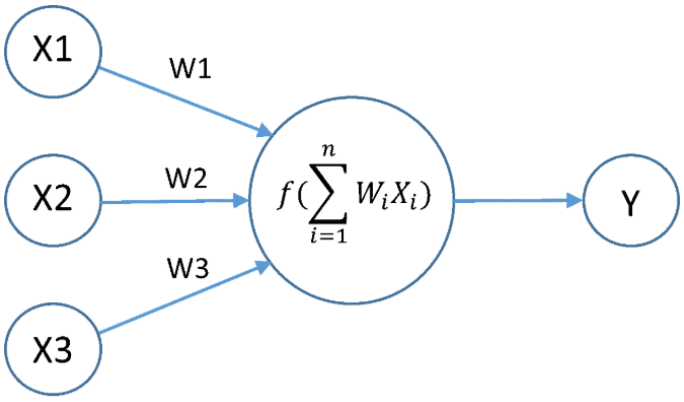
No matter whether we realize it or not, neural networks and deep learning are gradually changing our life. They are widely used in many applications like business, search engine, social media, and image recognition. Machine learning is based on computational statistics, which allows computers to make progressive predictions on data sets. And this progressive prediction is normally guided by algorithms based on neural network. A neural network is biologically inspired and can be composed of three parts: input layer, hidden layer, and output layer, and each

[1] Source: https://www.extremetech.com/computing/104835-intel-end-sandy-bridge-e-3960x-review

layer may include multiple neurons, as shown in Figure 14 A. As can be seen from Figure 14 B, a neuron is basically a mathematical activation function which takes several numbers as inputs. A hidden layer neuron calculates the convolution of previous layer neurons' output and their respective weights (w_1, w_2, w_3) and then pass the result on to output neurons through an activation function. Output of the neural network may not be the expected value; however, the



(A)



(B)

Figure 14. (A) Neuro Network. (B) Activation Function[2].

[2] Source: https://blog.webkid.io/neural-networks-in-javascript/

goal of the machine learning is to minimize the difference between the output and the expected value. To do this, algorithms like backpropagation are adopted to trace back the neural network and update the weights between neurons. This process goes back and forth many times till an accepted difference between the neural network output and the expected value is achieved. This is called neural network training. During this training process, all the weights should be updated multiple times. To update a weight, it is necessary to first read the weight out from SRAM and then write the new value into the SRAM. So neural network training requires frequent SRAM read and write access and SRAM read and write speed plays a vital role in determining the speed of machine learning microprocessors. As a matter of fact, SRAM read and write speed has become the bottleneck of the development of microprocessors. Thus, it is urgent to research on solutions to increase SRAM read speed. In this dissertation, it is focused on SRAM read speed.

In this project, we are working with a research team from University of Arkansas and Otureo Technologies (Beijing) Co. Ltd on designing a microprocessor for image recognition. We take full responsibility of the SRAM design while research teams from University of Arkansas and Otrueo work on the rest of the microprocessor design which includes the logic core and peripheral circuits. Specifications of SRAM are determined by the research team from Otureo Technologies (Beijing) Co. Ltd and University of Arkansas who is the designer of the logic core. The SRAM is custom made to work with the logic core. Figure 15 shows a brief block diagram of the data flow within the machine learning processor. The microprocessor takes a 32×32-pixel image as an input. Then the image is disassembled into many small arrays. These arrays go through convolutional calculations three times to get the final output. SRAM is designed to store these arrays so that the convolutional calculations can be performed. For each convolutional

calculation, these arrays need to be read from the SRAM. Results of the array convolutional calculation are new arrays which will be written back to the SRAM.

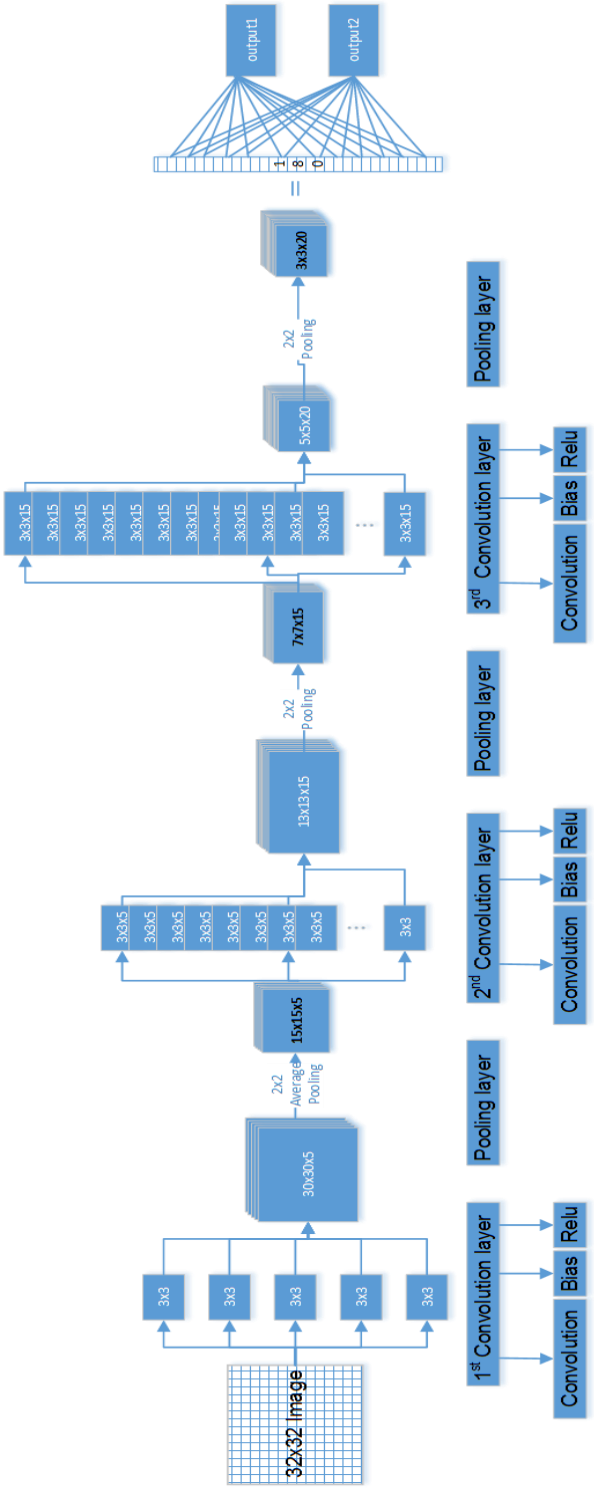


Figure 15. Data Flow of the Machine Learning Processor.

3.2. SRAM Design Specification

3.2.1. SRAM Pin Information

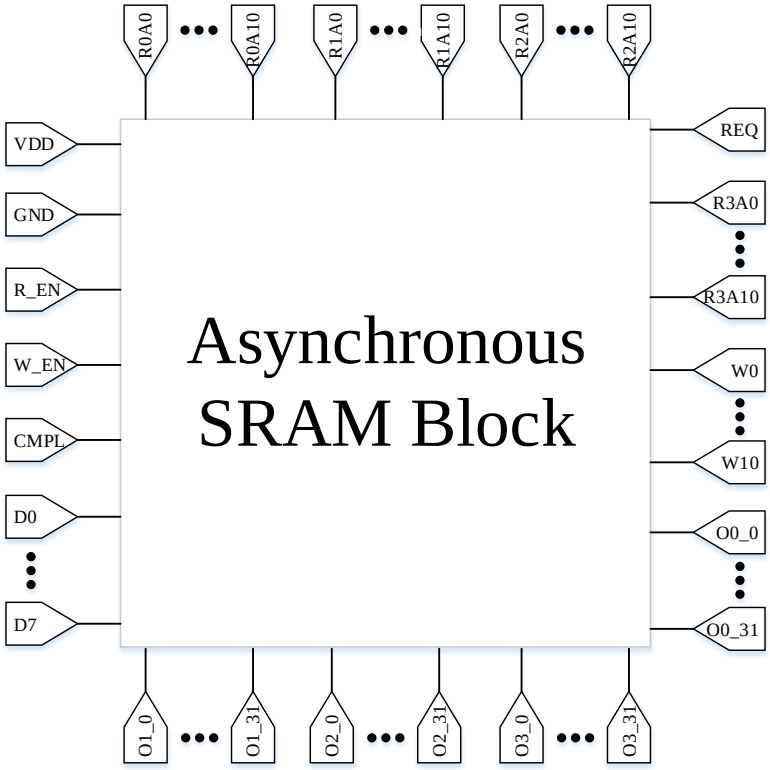


Figure 16. Asynchronous SRAM Schematic Pins.

As shown in Figure 16 and Table 4, the memory has totally 186 pins, which include VDD, GND, read enable (R_EN), write enable (W_EN), completion signal(CMPL), four 11-bit read addresses, four 32-bit output data, a 11-bit write address, and an 8-bit data to write in.

3.2.2. SRAM Memory Operation

The SRAM memory is asynchronous, read and write operation of the memory is not controlled by clock signals, instead, they are controlled by R_EN and W_EN signals. When neither R_EN nor W_EN is asserted, the memory goes to reset mode, when reset is completed, the request for data signal REQ will go high which means the memory is ready to read and write. As shown in Table 4, if R_EN is asserted, the memory goes to read mode, and REQ signal will go low and CMPL signal will be asserted when read is completed. When CMPL signal is

asserted, the memory is ready to go to reset mode by de-asserting R_EN and W_EN. CMPL signal will be de-asserted once reset mode starts, during which REQ signal will be asserted once reset is completed. If W_EN is asserted, the memory goes to write mode and de-asserts the REQ signal. When write is finished, CMPL signal will be on and memory is ready to go to reset mode by de-asserting R_EN and W_EN. R_EN and W_EN should never be asserted at the same time.

Table 4. Asynchronous sram memory pins and descriptions.

Pin Names	Description
VDD	Power supply, 1.2 V
GND	Ground
R_EN	Read enable signal
W_EN	Write enable signal
CMPL	Completion signal
REQ	Request for input data
R0A0-R0A10	Read address 0, 11 bits
R1A0-R1A10	Read address 1, 11 bits
R2A0-R2A10	Read address 2, 11 bits
R3A0-R3A10	Read address 3, 11 bits
O0_0-O1_31	Output data for read address 0, 32 bits
O1_0-O1_31	Output data for read address 1, 32 bits
O2_0-O2_31	Output data for read address 2, 32 bits
O3_0-O3_31	Output data for read address 3, 32 bits
W0-W10	Write address, 11 bits
D0-D7	Data to write in, 8 bits

The memory needs to be reset before each read and write operation. To reset the memory, W_EN and R_EN need to be held low till REQ signal is activated.

Each read address is 11 bits and matches 1-byte (8 bits) data. For each read operation, four addresses (address_0, address_1, address_2, and address_3) will be given and this is called one address pattern. For each address within an address pattern, data stored in the byte which matches that address is selected and read out, besides, data stored in the next 3 following addresses are also selected and read out. For example, suppose read address pattern {0, 20, 30, 40} is given, then addresses {0, 1, 2, 3}, {20, 21, 22, 23}, {30, 31, 32, 33}, and {40, 41, 42, 43} will all be selected to read out. For address_0, 4 bytes will be selected which means 32-bit data will be read out to pins O0_0 to O0_31. For address_1, data will be read out to pins O1_0 to O1_31. For address_2, data will be read out to pins O2_0 to O2_31. For address_3, data will be read out to pins O3_0 to O3_31. Within each address pattern, minimum gap between two addresses is 3 bytes to ensure that there is no overlap; and maximum gap between address_1 and address_4 can be set to 60 bytes, according to the address patterns provided by Dr. Di's team. REQ signal has to be asserted before a read cycle can begin. To start a read cycle, R_EN has to be asserted, and then 4 addresses needs to be ready on pins R0A0-R0A10, R1A0-R1A10, R2A0-R2A10, and R3A0-R3A10, and then 4 32-bit data will be read out and ready on pins O0_0 to O0_31, O1_0 to O1_31, O_2 to O2_31, and O3_0 to O3_31 respectively. Both R_EN and 4 read addresses need to be held till CMPL signal is asserted.

As shown in Table 5, for each write operation, 8-bit data is written into the SRAM memory. REQ signal has to be asserted before a write cycle can begin. To start a write cycle, W_EN signal needs to be asserted, 11-bit address need to be ready on pins W0-W10, and one 8-

bit data to be stored in the memory needs to be ready on pins D0 to D7. All W_EN and address signals need to be held till CMPL signal is asserted.

Figure 17 demonstrates the status of all pins during one read operation and one write operation of the asynchronous memory.

Table 5. State diagram of sram pins.

	Read	Write	Reset	Never Happen
R_EN	1	0	0	1
W_EN	0	1	0	1
CMPL	Asserted when read is completed	Asserted when write is finished	De-asserted	Do not care
REQ	De-asserted	De-asserted	Asserted when reset is completed	Do not care
R0A0-R0A10	Ready	Do not care	Do not care	Do not care
R1A0-R1A10	Ready	Do not care	Do not care	Do not care
R2A0-R2A10	Ready	Do not care	Do not care	Do not care
R3A0-R3A10	Ready	Do not care	Do not care	Do not care
O0_0-O0_31	Ready	Do not care	Do not care	Do not care
O1_0-O2_31	Ready	Do not care	Do not care	Do not care
O2_0-O2_31	Ready	Do not care	Do not care	Do not care
O3_0-O3_31	Ready	Do not care	Do not care	Do not care
W0-W10	Do not care	Ready	Do not care	Do not care
D0-D7	Do not care	Ready	Do not care	Do not care

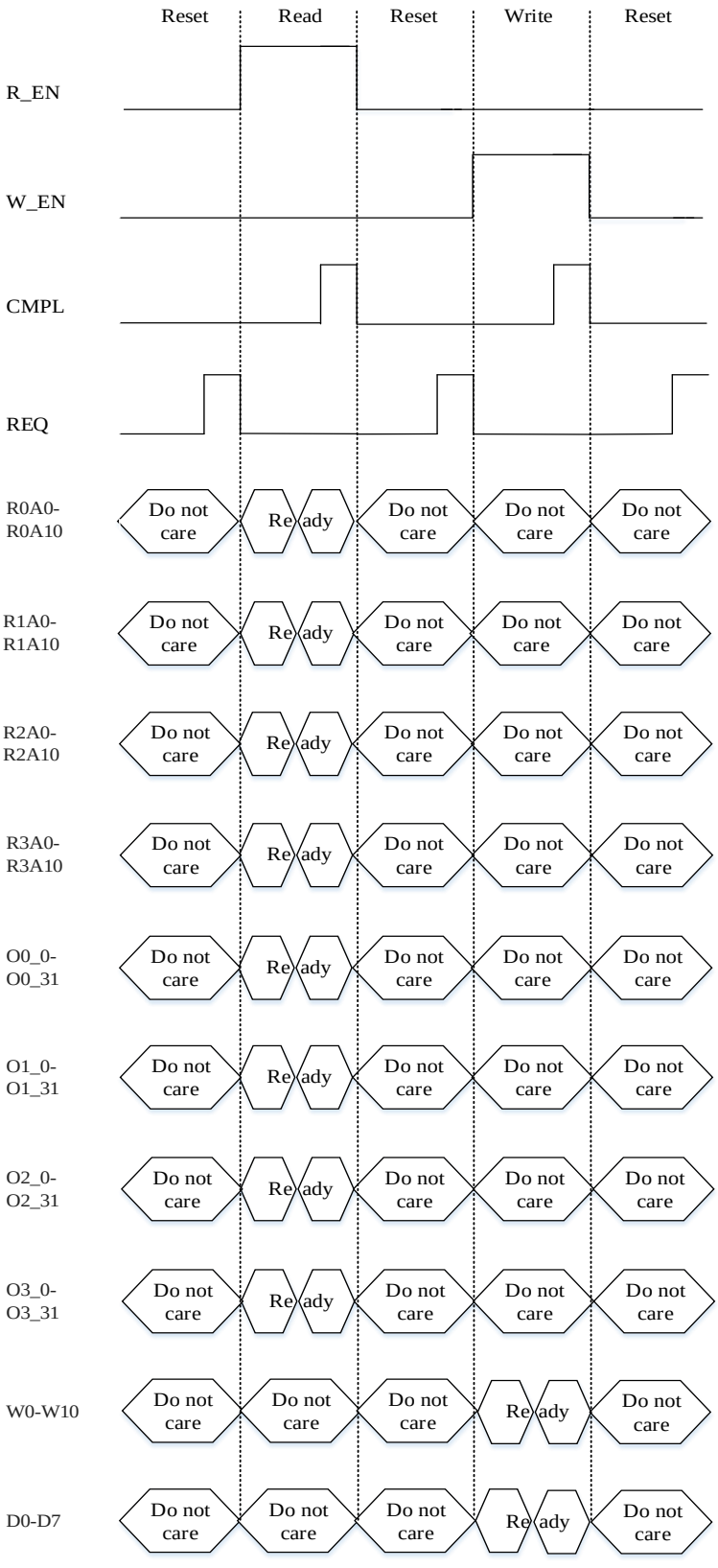


Figure 17. Timing Diagram of All Pins during Read and Write Operation.

3.2.3. VHDL Behavioral Model

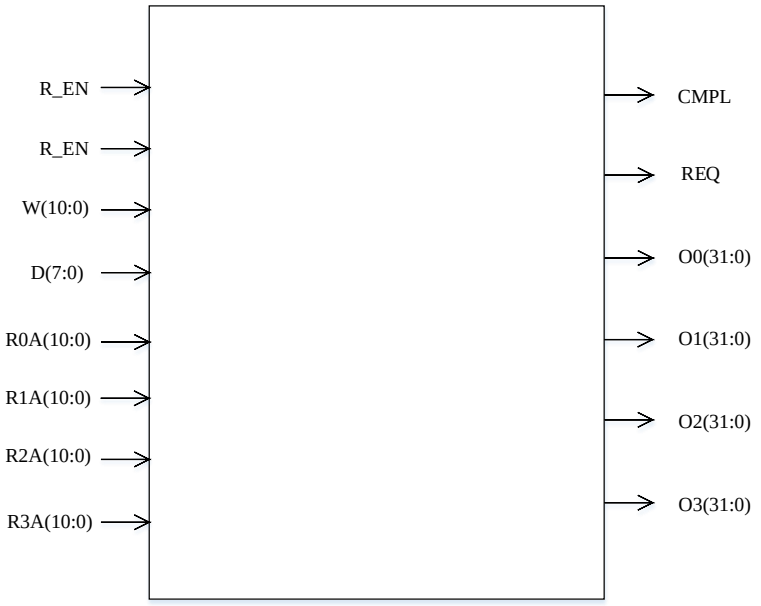


Figure 18. Asynchronous SRAM Inputs and Outputs for VHDL Behavioral Model.

Behavioral model of the asynchronous SRAM is completed using VHDL and inputs and outputs of the behavioral model are shown in Figure 18. Eight inputs are read enable (R_EN), write enable (W_EN), 11-bit write address (W), 8-bit write data (D), 11-bit read address 0 (R0A), 11-bit read address 1(R1A), 11-bit read address 2 (R2A), and 11-bit read address 3 (R3A). Six outputs are completion signal (CMPL), request for data signal (REQ), 32-bit read output data 0 (O0), 32-bit read output data 1 (O1), 32-bit read output data 2 (O2), and 32-bit read output data 3 (O3).

The address pattern samples can be seen in Appendix B.

3.2.4. Difference Between the Proposed SRAM and Traditional SRAM

Based on the specifications, this application specific SRAM are different from traditional ones in several ways:

1. This SRAM is asynchronous, which means there is no system clock controlling the operation of the SRAM. Instead, the SRAM is controlled by write enable W_EN, read

enable R_EN, and hand shaking signals like request for data REQ, and completion CMPL. W_EN needs to be asserted before any write operation can happen, and similarly, R_EN needs to be asserted before any read operation can happen. When bitlines are precharged and SRAM is either ready to read or write, REQ signal will be asserted. CMPL signal goes high only when either read or write operation is completed. So, compared to conventional SRAM, extra control circuits need to be designed to properly control W_EN, R_EN, REQ, and CMPL signals.

2. The write operation of SRAM is same with the conventional ones. One write address is given at a time, and one word (normally 8 bits, 16 bits, 32 bits, or 64 bits, depending on the word size of the processor) will be written to the corresponding address. However, the read operation of SRAM is unique compared to conventional ones. Instead of taking one address at a time, the SRAM takes four addresses at a time. Instead of reading out one word for each address, it reads out four consecutive words for each address. So, for each read operation, 16 words will be read out. What is worth to mention is that the four reading addresses given at each read operation are not completely random, and they follow two address patterns. Within the first address pattern, it is ascending order from the first address to the fourth one and the difference between two adjacent ones is 15. Within the second address pattern, it is also in ascending order except the difference between two adjacent ones is 7.

3.3. SRAM Circuit Design

3.3.1. Hand Shaking Signal Control Circuit

From previous discussion it is known that when either SRAM is ready to read or write, REQ will be asserted. Both BLs and BLbs will be precharged high before a read or write

operation happens. In other words, when both BLs and BLbs are high, SRAMs are ready to request for data and signal REQ goes high. So, an AND gate is used to evaluate a set of bitlines. We want to make sure that REQ is asserted on time to eliminate the delay between REQ signal and the read/write operation, so two sets of bitlines and connect them via an OR gate. When either set of bitlines is precharged, REQ would be asserted. The schematic of the REQ control circuit is shown in Figure 19.

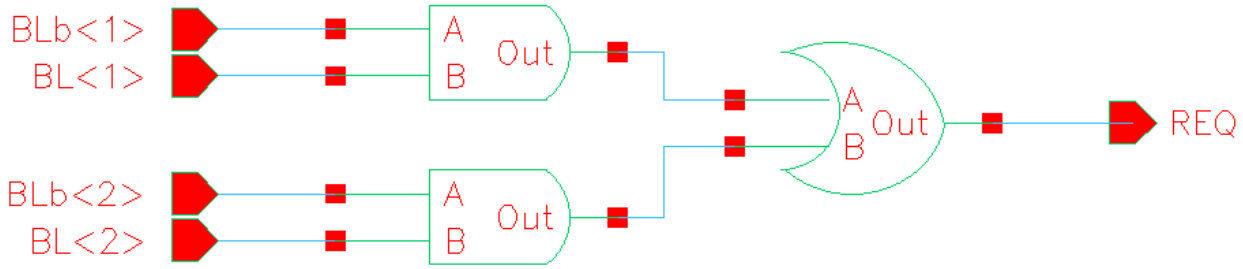


Figure 19. Schematic of REQ Circuit.

In this proposed SRAM, both read and write operation starts with one of the bitlines in a set stays high and the other one gets discharged. For example, if a "1" is to be written into an SRAM cell, BLb of that cell will stay high and BL will be discharged. If the condition that one bitline is "0" and the other one is "1" within a bitline set, then the read/write operation is finished. So XOR gate is used to realize this control logic, as shown in Figure 20. Since delay propagation exists in the process of bitline precharge, BLs and BLbs of different SRAM cells are precharged to high at different times, to make sure the bitline with the longest delay time can finish the read/write operation before CMPL signal goes high, two sets of BL and BLb with the longest propagation delay are connected with a two-input OR gate and delay cells are added. Only when read/write operations are all completed in those three bitline sets can CMPL signal be asserted.

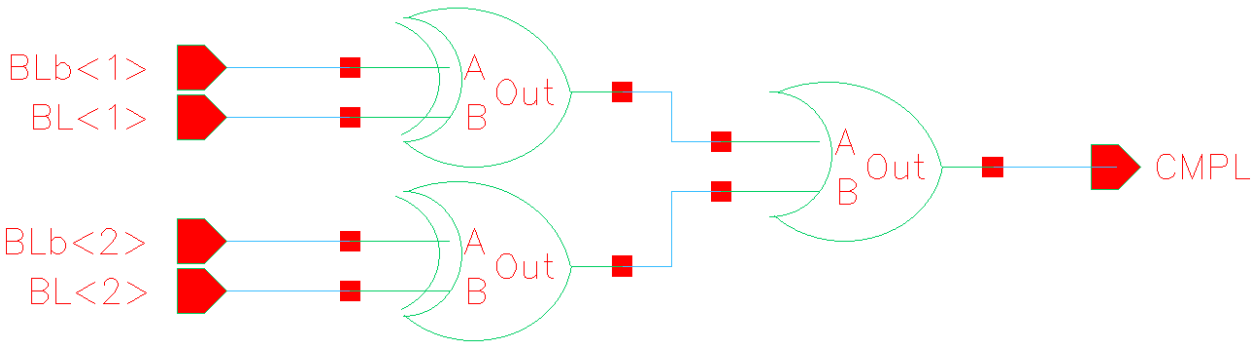


Figure 20. Schematic of CMPL.

3.3.2. Transmission Gate

A transmission gate has three inputs, namely they are In, Enable, and $\overline{\text{Enable}}$, and one output named Out. As can be seen from Figure 21, a transmission gate is a combination of a pair of paralleled connected NMOS and PMOS. The gate voltage applied to these two transistors are complementary signals. When Enable signal is high, $\overline{\text{Enable}}$ will be low, so both the NMOS and the PMOS are turned on and there will be a low impedance current path from node In to node Out. If Enable signal is low, $\overline{\text{Enable}}$ will be high, both NMOS and PMOS will be turned off. The path between node In and node Out will be high impedance. So, the transmission gate works as a bidirectional switch between node In and node Out and the switch is controlled by Enable signal. Either a single NMOS or a single PMOS can work as switch controlled by a digital signal, however, both NMOS and PMOS have their limitations as a switch. The maximum high voltage an NMOS can pass is not VDD but VDD minus the NMOS threshold voltage. The minimum low voltage an PMOS can pass is not zero but the PMOS threshold voltage. The transmission gate overcomes these limitations by connecting an NMOS and a PMOS. It can pass a voltage as high as VDD and as low as zero. Transmission gates are widely used as a switch in the proposed SRAM design and its symbol is shown in Figure 21.

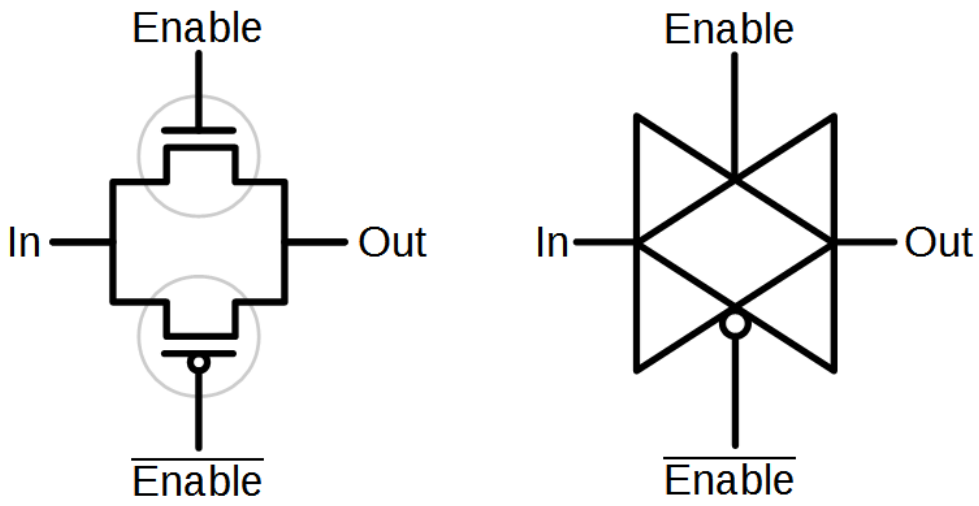


Figure 21. Schematic of a Transmission Gate.

3.3.3. Write Driver

Both bitlines in a pair will be charged to VDD before a write operation. Figure 22 shows the schematic of the proposed write driver. When the write enable signal (W_EN) goes low and W_ENb goes high, both transmission gates are turned off and no data can be written into SRAM through bitlines. A write operation starts when the W_EN goes high. In this case, both transmission gates connecting to BL and BLb are turned on to be a low impedance path. W_data is the input data. If the input is a "1", the NMOS on the left side will be turned on and the one on the right side will be turned off. This will keep BLb to stay high and discharge BL to low. Since the wordline of an SRAM is turned on before a write operation, high BLb and low BL will be able to write a "1" into the SRAM cell. On the contrary, if the input is a "0", the BL will go high and BLb will be discharged, and a "0" will be written into the SRAM cell. Note that the NMOS sizes have a negative relationship with the discharge time of bitlines from high to low in a write operation. PMOS sizes determines the drivability a write driver should keep the bitlines high during a write operation. These transistors shall be large enough to be able to flip the cross-coupled inverters in an SRAM cell, and the larger their sizes are, the smaller the write latency is.

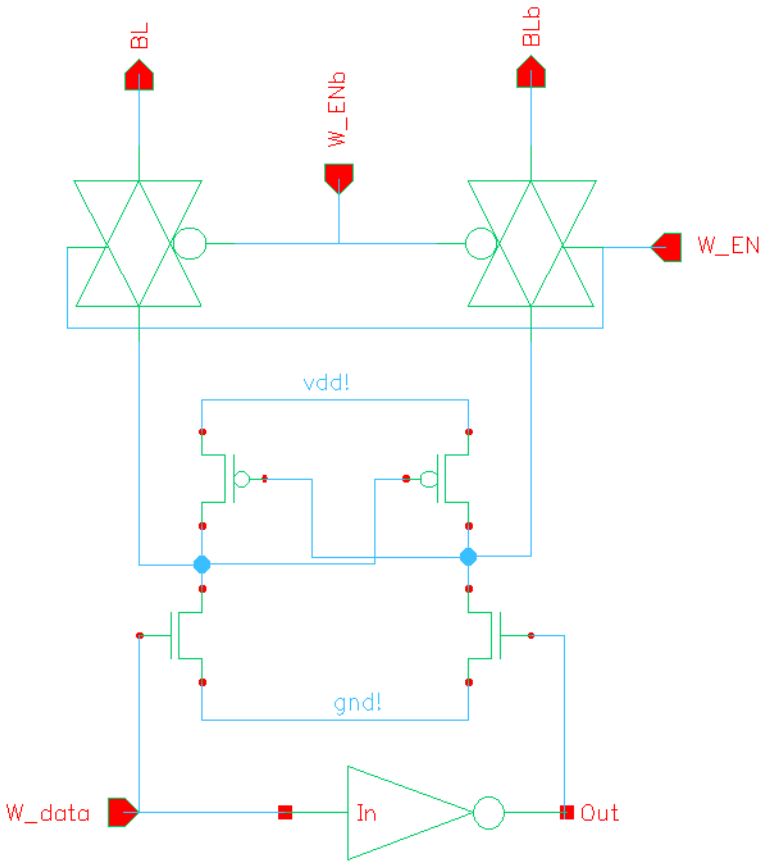


Figure 22. Write Driver Schematic.

3.3.4. Sensing Amplifier

When the evaluation process starts during an SRAM read operation, either BL stays high and BLb starts to discharge or BLb stays high and BL starts to discharge, depending on whether it's a "0" or "1" stored in the SRAM. To sense the voltage difference between a pair of bitlines and change this information into the correct binary output value, sensing amplifier is needed. The sensing amplifier proposed in this design is a differential amplifier, as shown in Figure 23. A differential amplifier is an amplifier that amplifies the difference between two voltages and rejects the average or common mode value of the two voltages. As shown in Figure 23, the SA used in the proposed SRAM is a differential amplifier with current mirror load. PMOS T0 and T1 are the current mirror load. When sensing amplifier enable signal (SE) is low, NMOS T4 is

turned off and output (Out) is invalid. During a read operation, SE will go high and NMOS T4 will be turned on. In case of reading a “0”, BL will stay high and BLb will be discharged and go low. If BLb goes low T3 will be turned off. Since BL stays high, T2 will stay on. Drain voltage of T2 will go low and drain voltate of T3 will go high, and the output will be a “0”. On the contrary, if it is a “1” stored in the SRAM, BL will go low and BLb will stay high. This will cause T2 to be off and T3 to be on. Then the drain voltage of T3 will go low and the output will be a “1”.

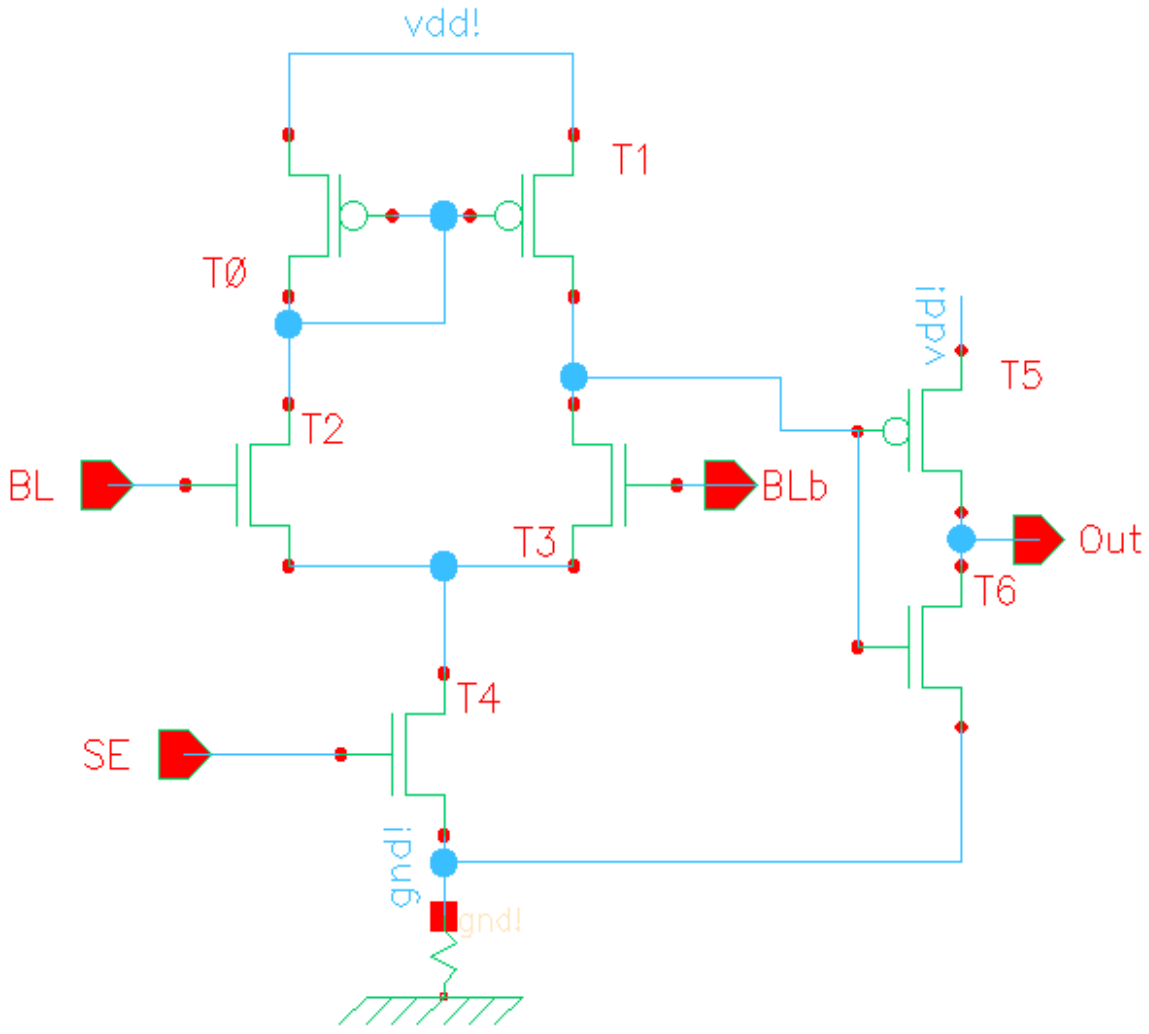


Figure 23. Sensing Amplifier Schematic.

3.3.5. Precharge Circuit

The precharge circuit is very simple. It includes three PMOS, two of which are used to generate the charging current and one is used as an equalizer. In many designs, an equalizer PMOS is used to reduce power consumption and for faster precharge [62]. Figure 24 shows the precharge circuit. The Pre signal is low active. When Pre signal goes low, all three PMOS will be turned on, both BL and BLb will be pulled to high voltage and this is the charging process. Pre signal stays high during read and write process.

Pre signal is controlled by W_EN and R_EN. Since SRAM is supposed to stop precharging during both read and write operation and starts to precharge when there is neither read nor write operation. Pre is "1" when either W_EN or R_EN is "1" and is "0" when both W_EN and R_EN are "0". W_EN and R_EN should never be "1" at the same time. So, Pre is controlled by W_EN and R_EN via an XOR gate.

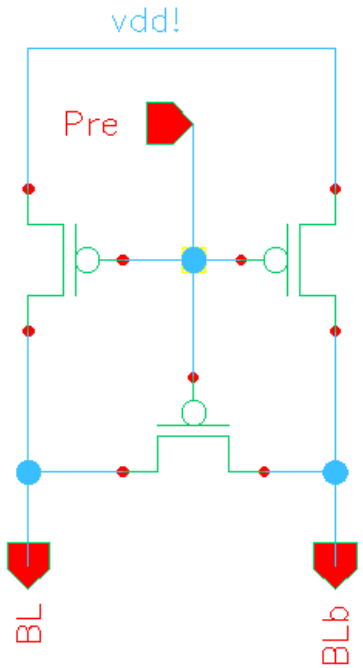


Figure 24. Precharge Circuit.

3.3.6. Column Circuitry

Figure 25 shows the block diagram of an SRAM column circuitry. It gives a general idea of how SRAM cells are connected with precharge circuit, write driver, sensing amplifier, and column selection circuit. Transmission gates are used in the column selection circuitry. Word line (WL) performs the function of row selection. It is common that there are multiple columns and rows. To write to or read from a certain SRAM cell, both column selection circuits and row selection circuits are necessary. Column decoders are used to select a column and row decoders are used to select a row. If a certain column is selected a “high” signal generated by the decoder is applied to the column selection transmission gates and this will turn both transmission gates on. BL and BLb will be ready to read or write in this case. However, when a certain column is not selected, column decoder will keep the column selection signal low and transmission gates will be at high impedance status. Neither read nor write operation can be operated to this column. Similarly, if a row is selected to read or write, a “high” signal generated by a row decoder is applied to the WL of that row so that SRAM are ready to read or write.

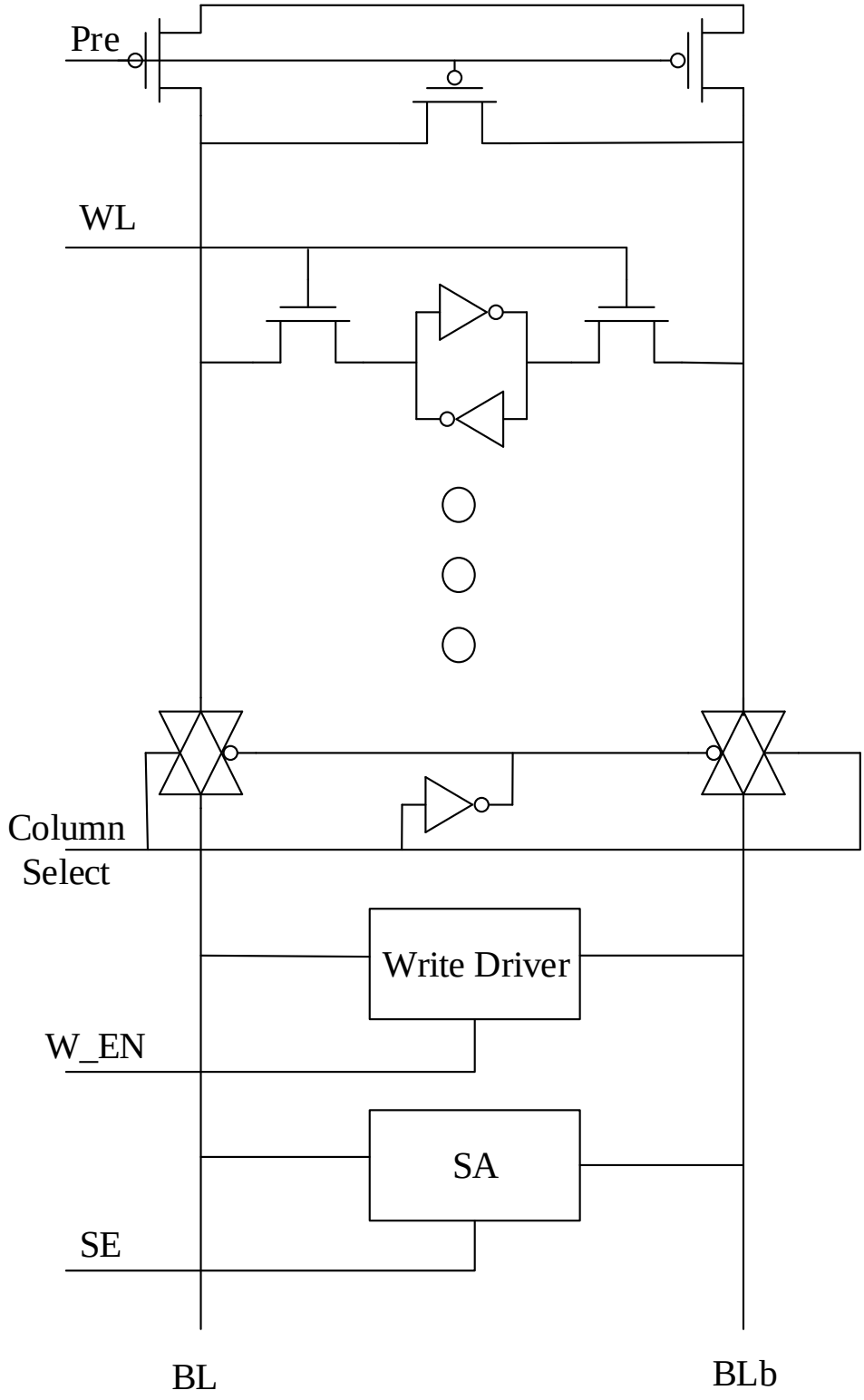


Figure 25. Column Circuitry.

3.3.7. Address Decoder

Address decoder is a circuitry that takes a binary input and decodes the the address by activating the corresponding outputs. Take a 2-to-4 address decoder for example, suppose the input is “10”, the decoder will decode it as the third address and activate the third output. In SRAM, address decoders are needed to select a certain row (by activating WL) or column (by activating column select). Figure 26 and Figure 27 demonstrate a 2-to-4 address decoder and a 3-to-8 address decoder respectively. Both address decoders have an EN which is the enable signal. The EN signal must be asserted before a decoder works and is deserted when a decoder is not working to avoid activating a row or column mistakenly. Three types of address decoders are used in the proposed SRAM design: 4-to-16 address decoder, 5-to-32 address decoder, and a 6-to-64 address decoder. All these three types of address decoders are made from 2-to-4 address decoders and 3-to-8 address decoders. The 4-to-16 address decoder circuitry is composed of four 2-to-4 address decoders; the 5-to-32 address decoder circuit includes one 2-to-4 address decoder and three 3-to-8 address decoders; the 6-to-64 decoder circuit is built up with nine 3-to-8 address decoders.

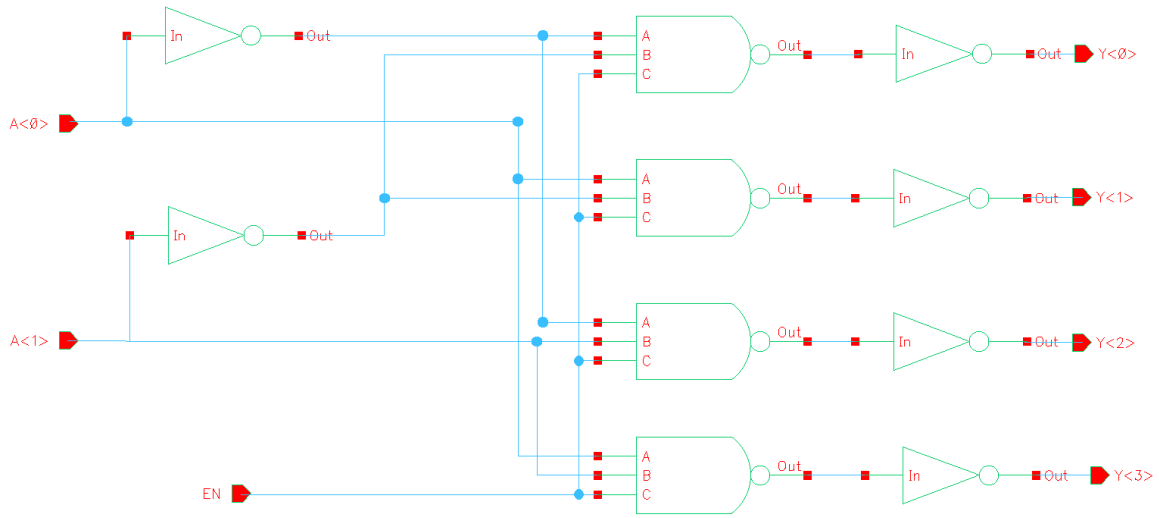


Figure 26. Schematic of 2-to-4 Decoder.

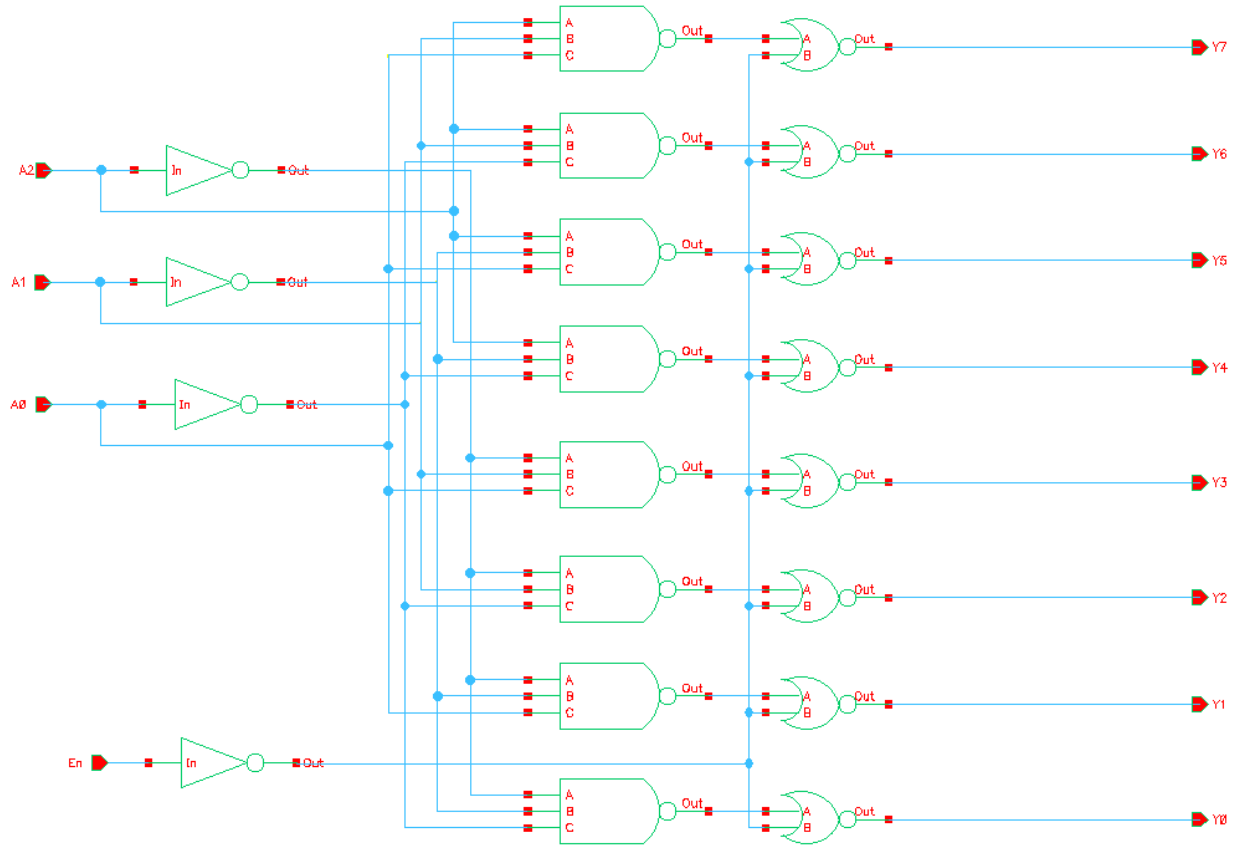


Figure 27. Schematic of 3-to-8 Decoder.

3.3.8. SRAM Block

The cells of SRAM are arranged in the form of two dimensional blocks. Each block has 32 rows and there are eight SRAM cells in each row. The whole SRAM includes 64 identical blocks. The SRAM has a different access order compared with traditional SRAMs. Traditional ones start from row zero to row 31 at block zero first, and then same order at block 1, and goes on till the last block. However, the proposed SRAM accesses from row zero at block zero to row zero at block one and goes on till row zero at the last block, then it starts from row one at block zero to row one at the last block. It keeps this access order till the last row at the last block. The SRAM is designed this way due to the unique reading requirements. It is easy to conclude from the address pattern that the 16 read addresses have a span of either 49 or 25 in decimal. Take

address pattern “1215 1216 1217 1218, 1230 1231 1232 1233, 1245 1246 1247 1248, 1260 1261 1262 1263” for example, from the first address “1215” to the last address “1263”, the addresses have a span of 49. Take address pattern “51 52 53 54, 58 59 60 61, 65 66 67 68, 72 73 74 75” for another example, the addresses have a span from “51” to “75” which is 25. Each read address corresponds to a row in a block where there are eight bits or one byte. For instance, address “0” refers to the first byte in the SRAM which are the eight SRAM cells connecting to WL0. To read from all the SRAM cells that are corresponding to an address pattern within one read cycle, none of these 16 bytes should be in the same block. Because if there are two bytes in the same block, they would share same bitlines. Like the byte connecting WL0 and and the byte connecting WL1 in the same column use the same pair of bitlines to evaluate the value stored in SRAM cells. There would be bitline conflict if the byte connecting to WL0 and the byte connecting to WL1 evaluate at the same time. Since the read address patterns have a maximum span of 49, there should be at least 49 blocks like the one in Figure 28. With the total size of the SRAM being 2 KB and the number of blocks in the SRAM being 2^n (n is an integer), 64 which is 2^6 is the smallest number that is greater than 49. This is the reason why the memory is designed into 64 blocks where there are 32 bytes in each of them.

Within each SRAM block, there are eight columns of SRAM cells and each column has a pair of bitlines (BL and BLb). Each column is connected to a sensing amplifier which outputs a binary data. So, there are 8 SAs in each SRAM block. The 2 KB SRAM has 512 SAs in total. All the SAs are controlled by the same SE signal. For each read operation, every sensing amplifiers evaluates the bitlines and outputs a one-bit data. Figure 29 illustrates the architecture of the SRAM.

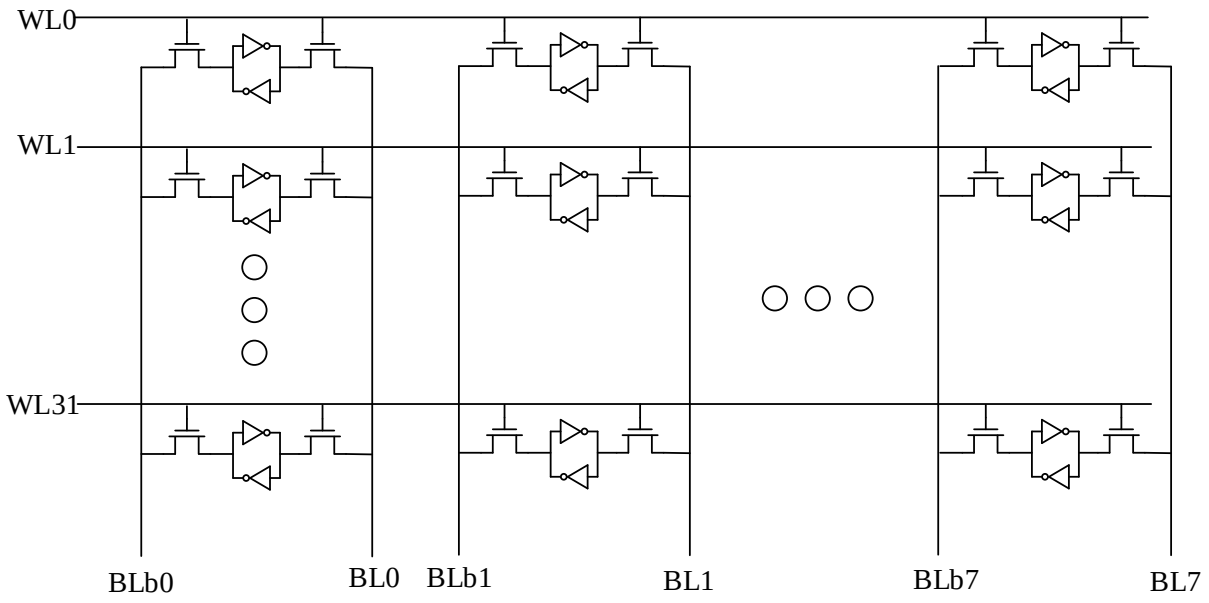


Figure 28. Single SRAM Block.

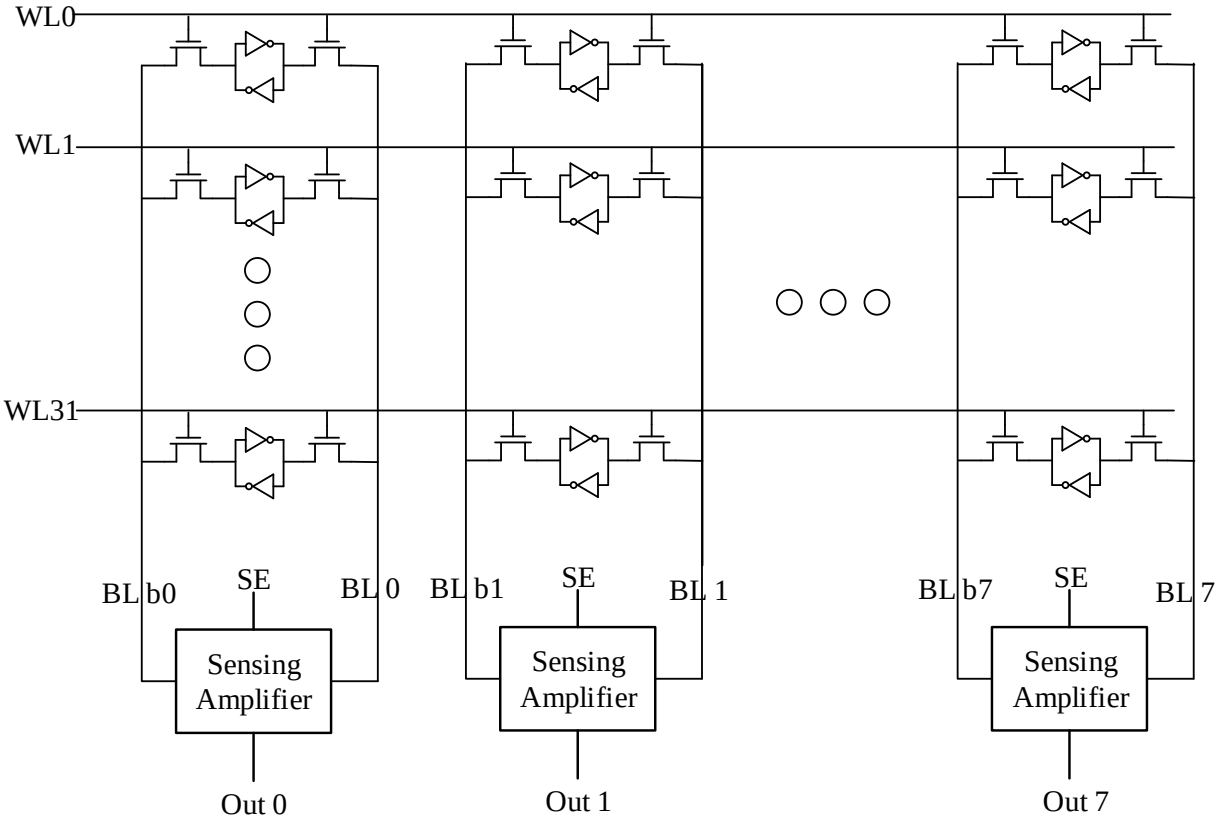


Figure 29. SRAM Block with Sensing Amplifier.

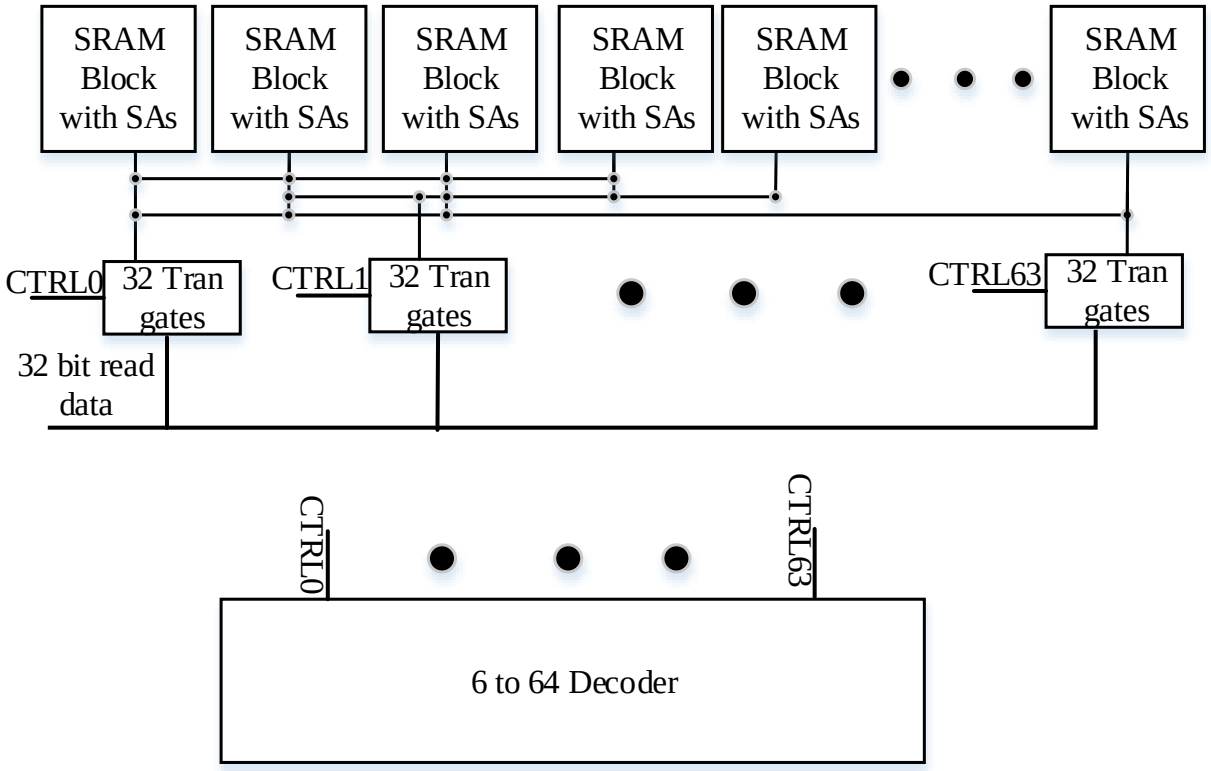


Figure 30. Transmission Gate Circle.

3.3.9. Control Circuit to Implement Four Byte Read Each Cycle

According to the design specification, four bytes (32 bits) next to each other need to be read for each read address. In order to meet the design requirement, a unique circuit is proposed, as shown in Figure 30. A 32-input transmission gate made of 32 single-input transmission gates is designed. All the 32 transmission gates are controlled by one high active CTRL signal. When "1" is applied to the CTRL, all 32 transmission gates act like closed switches and signals can pass through it. On the contrary, when "0" is applied to the CTRL, all 32 transmission gates act like open switch and signals will be blocked. A circuit called transmission circuit is designed with 64 32-input transmission gates. Each of the inputs of a 32-input transmission gate is connected with eight outputs from sensing amplifiers of the corresponding SRAM block and output from 24 amplifiers of the next three SRAM blocks. For example, as shown in Figure 30, the first transmission gate is connected with block 0, block 1, block 2, and block 3. The 62nd 32-

input transmission gate is connected with block 61, block 62, block 63, and block 0. This is because the first 64 bytes are in the first row of the 64 SRAM blocks from block 0 to block 63. The 65^{th} byte to 128^{th} byte are located in the second row from block 0 to block 63. Specifically speaking, the 65^{th} byte is in second row of block 0, 66^{th} byte is in second row of block 1, etc. The way how these 32-input transmission gates are connected looks like a circle, and this is how the circuit is named.

These 64 transmission gates are controlled by a 6-to-64 decoder, with each output of the decoder controlling one 32-input transmission. These 64 transmission gates perform the function of column selection circuits. Suppose the read address input of the 6-to-64 decoder is “000000”, CTRL0 in Figure 30 will be activated. CTRL0 controls the first 32-input transmission gate, which connects with the first four blocks. The four-byte data from block 0 to block 3 will be read out.

Four sets of the read circuits shown in Figure 30 which includes the transmission gate circle and a 6-to-64 decoder are employed in the design. Each set of the circuits can read one address in and outputs 4 bytes next to each other. These four sets of circuits can work at the same read cycle and take 4 read address and read the data out. For instance, the read address set which includes four addresses is {0, 7, 14, 21}, these read address will be applied to 4 different 6-to-64 decoders in the four sets of the read circuits: 0 applied to the first decoder, 7 applied to the second decoder, 14 applied to the third decoder, and 21 applied to the fourth decoder. The first read circuit will read the data out from block 0 to block 3, the second read circuit read the data from block 7 to block 10, the third read circuit read from block 14 to block 17, and the fourth read circuit read from block 21 to block 24.

3.3.10. Row Selection Control Circuit

It is easy to understand how the read process works if the four read addresses are all within 0 to 64. However, since there are 2K bytes in the SRAM, there will always be cases that the read addresses are out of this range. For example, one of the read address pattern is {51, 58, 65, 72}. Address 65 and 72 are at second row of block 1and block 8 respectively. Row decoders are used to control which row to read. If a decoder activates a certain row, WL for that row will go high and data in that row can be evaluated through bitlines. Since there are 32 rows in the SRAM, row decoders are designed to be a 5 input to 32 output decoders. The read address should be 11 bits to be able to assign a unique address to each byte within a 2 KB SRAM (2^{11}=2048). Suppose there is one 11-bit reading address A_{10} to A_0, the first 5-bit address A_{10} to A_6 goes to the row decoder and the last 6-bit A_5 to A_0 goes to the column decoder.

There are cases that the four addresses within a read address set locate at two different rows. Which means different WLs need to be activated during the same read cycle. For traditional SRAMs, there is only one row decoder and all WLs are controlled by that row decoder. There can only be one WL activated within the same read cycle. What's more, there would be bitline conflict if two WLs in the same column are turned on in the same cycle. So new row selection circuit must be designed to activate different WLs in one read cycle.

Instead of using one row decoder to control all bitlines, we used 16 decoders with each one controlling four SRAM blocks. Decoder 0 controls SRAM block 0 to block 3, decoder 1 controls SRAM block 4 to block 7, etc. Since there are 16 row decoders, there can potentially be a WL control select circuit that controls some of the 16 row decoders to activate the first row and some of them to activate the next row. As mentioned in the previous discussion, for a set of read addresses {address 0, address 1, address 2, address 3}, the decimal value of these four addresses

are in an ascending order. Address 0 is always the smallest and address 3 is the largest. The basic design concept to control which decoders should activate the first row and which ones should activate the next row is as follows:

Location of address 0 is decided by the A0 to A5 of address 0. If address 0 is in one of the four blocks controlled by row decoder 0, since the largest span of the read addresses is 49, it is for sure that all the four addresses will be smaller than 64 and will all be in the first row. For example, if A5 to A0 is “000011”, it means that address 0 is in block 3. According to the address pattern, the decimal value of A5 to A0 of the next three addresses could either be {10, 17, 24} or {18, 33, 48}. For both cases, these addresses are all in the same row. If address 0 is in the one of the four blocks controlled by row decoder n (n is an integer and greater than 0), row decoder 0 to row decoder n-1 are designed to activate the next row or WL. But designing it this way, it is secure that in case some of the read addresses locates in the next row, the WL of the next row is always activated and ready. For instance, A5 to A0 of the read address 0 is “011111”, address 0 is in block 31, which is controlled by row decoder 7. So, row decoder 7 to decoder 15 will activate the first WL and row decoder 0 to decoder 6 will activate the next WL. The next three addresses could either be {36, 41,46} or {46, 61, 76}. 76 would be the next row in SRAM block 12. Since row decoder 0 to decoder 6 activates the next WL which means SRAM block 0 to SRAM block 27 activates the next WL. As a result, data in address 76 can be read out correctly.

3.3.11. Row Decoder Control Circuit

A unique row decoder control circuit is proposed to control which row decoders should activate the first row and which row decoders should activate the next row. The row decoder control circuit can be divided into two parts.

The first part of the control circuit solves the problem how a decoder can either activate the first WL or the next WL. As shown in Figure 31, an adder circuit is used to control whether the row decoder shall activate the first WL or the next WL. Address A6 to A10 does not control the row decoder directly. Instead, it controls an adder circuit and the 5-bit output of the adder controls the decoder. If the addend input of the adder is 0, 5-bit output of the adder will be same with A6 to A10, the row decoder activates the first WL. However, if the addend is 1, the output of the adder would be greater than A6 to A10 by 1and the row decoder would activate the next WL. So by controlling whether a 0 or 1 signal sent to addend input of the adder, activation of the first WL or the next WL could be controlled. For example, if the reading address is 1035, the binary form would be “10000001011”. A6 to A10 would be “10000”. When addend is 0, WL16 would be activated. However, when addend is 1, WL17 would be activated.

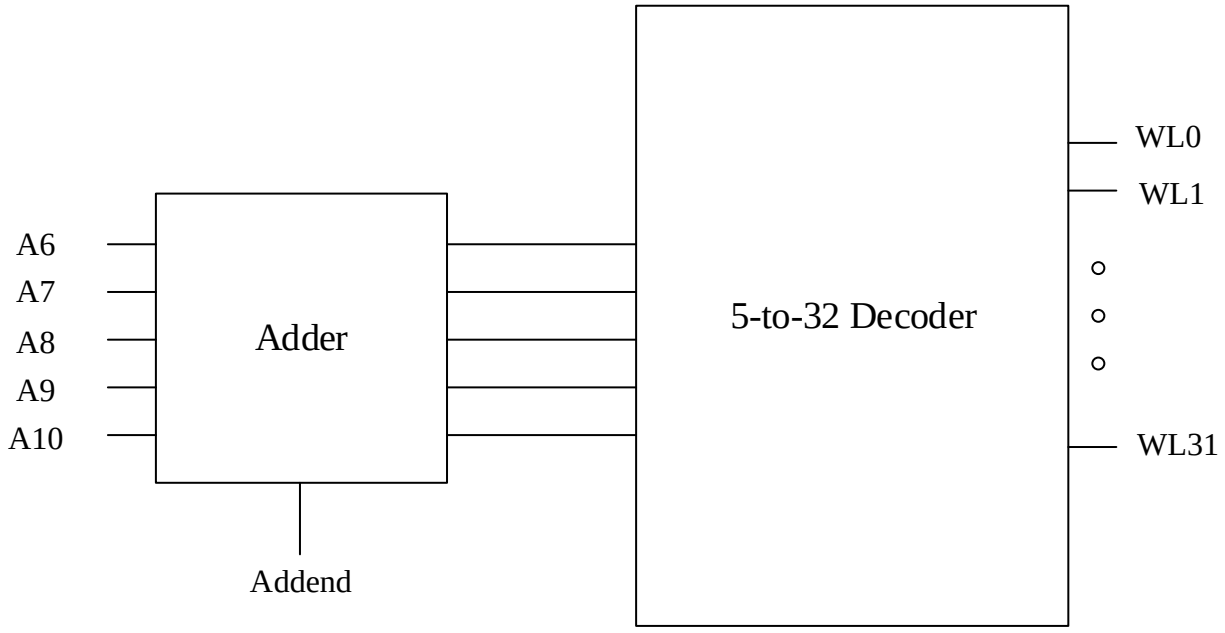


Figure 31. Row Decoder Control Circuit.

The second part of the row decoder control circuit is to solve the problem of addends of which decoders should be 0 and which decoders should be 1. It is called addend controller. For a

read address set {address0, address1, address2, address3}, any decoders that control SRAM blocks which are in front of the SRAM block where address0 is shall have 1 as the addend, and the rest shall have 0 as the addend. It is obvious that address0 decides whether the addend is 1 or 0. Specifically speaking, the SRAM block group number that address0 is in decides which addends are 1 and which addends are 0. With the 11-bit read address, A0 to A5 decides which SRAM block the address is in. Each SRAM block group has four SRAM blocks, so the higher four bits of A0 to A5 which is A2 to A5 decides which SRAM block group the address is in. For instance, suppose the read address is 1035, which is "10000001011" in binary. A0 to A5 is "001011", so the address is in SRAM block 11. A2 to A5 is "0010", so the address is in SRAM block group 2, which is the third group. Figure 32 shows the block diagram of the addend controller. Input of the addend controller is A2 to A4, and 15outputs are addend 0 to add14. To design the control logic from input to output, the truth table is shown in Table.

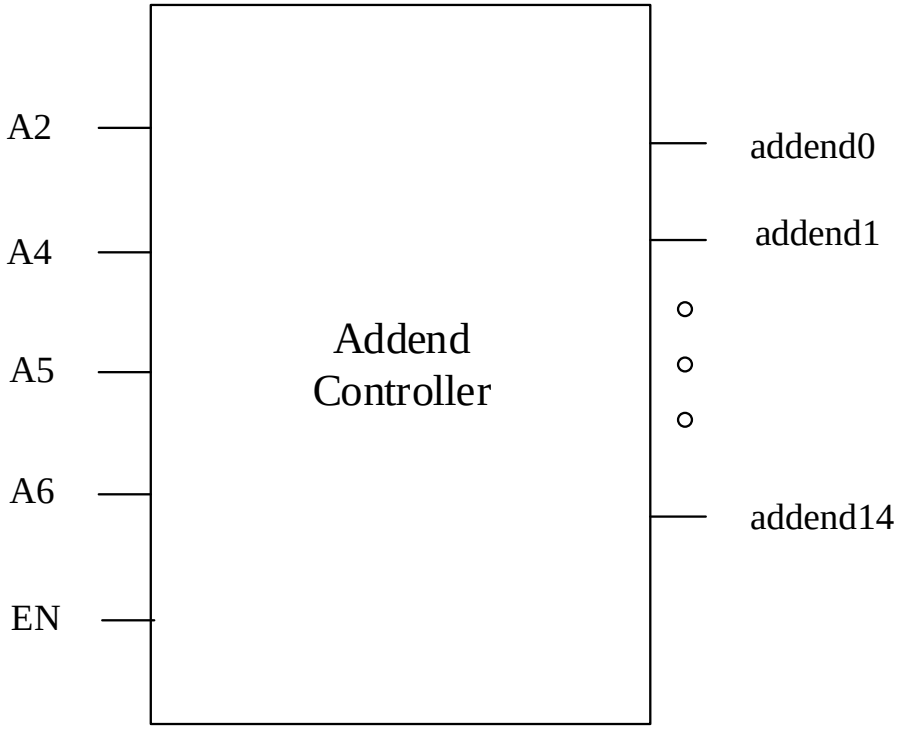


Figure 32. Addend Controller Block Diagram

Table 6. State Diagram of the Addend Controller

A5	A4	A3	A2	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The addend controller also has an enable control input EN. The truth table above is true when EN is high. When EN is low, all outputs of the controller are zero. From the truth table, K-maps of E0 to E14 can be plotted and Boolean expressions of E0 to E14 can be developed from the K-maps. To design logic circuits directly from Boolean expressions developed from K-maps, it may need various logic gates like OR gate, AND gate, NOT gate, etc. However, in practice it is preferable to use NOR gates and NAND gates to replace OR gates and AND gates. This is

because compared to an AND gate or an OR gate which has six transistors and two stages of transistors, a NAND gate or a NOR gate only has four transistors and one stage of transistors. So NAND and NOR gates tend to take smaller silicon area and have less propagation delay. Thus, we further simplify these Boolean expressions after developing from K-maps so that the circuit can be optimized with NAND and NOR gates. Boolean expressions of E0 to E14 and their Nand and NOR gate versions are listed from Equation 2 to Equation 16.

$$E0= EN\cdot A2+EN\cdot A3+EN\cdot A4+EN\cdot A5$$

$$= \overline{\overline{\overline{\overline{A3 + A4 + A5 + \overline{A2}}}} \cdot EN} \quad (2)$$

$$E1=EN\cdot A3+EN\cdot A4+ENA5$$

$$= \overline{\overline{A5 + A4 + A3} + \overline{EN}} \quad (3)$$

$$E2=EN\cdot A4+EN\cdot A5+EN\cdot A3\cdot A2$$

$$= \overline{\overline{\overline{A2 \cdot A3 \cdot A4 \cdot A5} \cdot EN}} \quad (4)$$

$$E3=EN\cdot A4+EN\cdot A5$$

$$=\overline{\overline{A4 + A5} + \overline{EN}} \quad (5)$$

$$E4=EN\cdot A5+EN\cdot A4\cdot A2+EN\cdot A4\cdot A3$$

$$=\overline{\overline{A2 + A3 + A5} + \overline{A4 + A5} + \overline{EN}} \quad (6)$$

$$E5=EN\cdot A5+EN\cdot A4\cdot A3$$

$$=\overline{\overline{EN \cdot A3 \cdot A4} \cdot \overline{EN \cdot A5}} \quad (7)$$

$$E6=EN\cdot A5+EN\cdot A4\cdot A3\cdot A2$$

$$=\overline{\overline{\overline{A2 \cdot A3 \cdot A4} \cdot \overline{A5} \cdot EN}} \quad (8)$$

$$E7=EN\cdot A5$$

$$=\overline{\overline{EN} + \overline{A5}} \quad (9)$$

$$E8=EN\cdot A5\cdot A2+EN\cdot A5\cdot A3+EN\cdot A5\cdot A4$$

$$=\overline{\overline{A2}+\overline{A3}+\overline{A4}+\overline{A5}+\overline{EN}} \quad (10)$$

$$E9=EN\cdot A5\cdot A3+EN\cdot A5\cdot A4$$

$$=\overline{\overline{A3}+\overline{A4}+\overline{EN}+\overline{A5}} \quad (11)$$

$$E10=EN\cdot A5\cdot A3+EN\cdot A5\cdot A3\cdot A2$$

$$=\overline{\overline{\overline{A2}+\overline{A3}}+A4+\overline{EN}+\overline{A5}} \quad (12)$$

$$E11=EN\cdot A5\cdot A4$$

$$=\overline{\overline{EN}+\overline{A5}+\overline{A4}} \quad (13)$$

$$E12=EN\cdot A5\cdot A4\cdot A2+EN\cdot A5\cdot A4\cdot A3$$

$$=\overline{\overline{A4\cdot A5\cdot EN}+\overline{A2+A3}} \quad (14)$$

$$E13=EN\cdot A5\cdot A4\cdot A3$$

$$=\overline{\overline{A3\cdot A4}+\overline{EN\cdot A5}} \quad (15)$$

$$E14=EN\cdot A5\cdot A4\cdot A3\cdot A2$$

$$=\overline{\overline{A2\cdot A3\cdot A4}+\overline{A5\cdot EN}} \quad (16)$$

According to the Boolean expressions of E0 to E14, the addend controller circuit can be designed with NAND, NOR, and NOT gates. With the addend controller circuit, a read operation can be performed successfully given a set of four read addresses, no matter whether these addresses are on the same WL or on two different WLs.

3.3.12. Write Circuit

The write operation of the proposed SRAM is same with write operation of traditional SRAM. One write address is given at a time, and 8 bit data is written to the SRAM cells that match the write address. The write circuit is similar to write circuits of traditional SRAMs. To write the 8-bit data into SRAM cells, the WL of the eight SRAM cells should be asserted by the

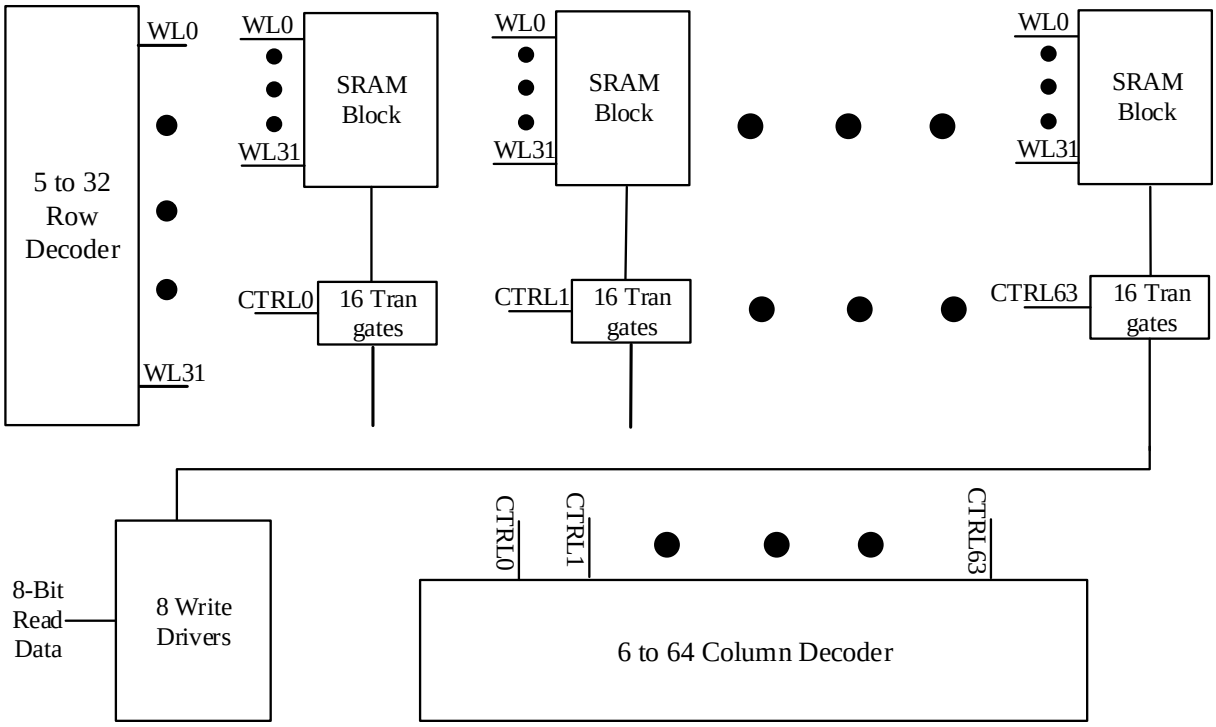


Figure 33. Write Circuit Block Diagram.

5-to-32 row decoder. The write operation shares exactly the same row decoders with read operations. All the addends of the row decoder adders are set to be 0 during write operation. The 8-bit data also should be ready when WL is asserted. The eight write drivers will be enabled once the 8-bit data is ready. Each write driver takes 1-bit data and outputs two signals which goes to both BL and BLb. Then the 6-to-64 column decoder will activate the column selection circuit which is composed of 16 transmission gates. Write drivers will be able to write to the column whose column select circuit is activated. For example, if data “01010101” needs to be written into address “00110011001”, the most significant five bits is the row address and the least significant bits is the column address. The 8-bit data will be the input of the eight write drivers which output to eight pair of bitlines. The row address is “00110”, WL6 will be activated and the eight SRAM cells that are connected with WL6 will be ready to write. The column address is “011001”, CTRL 25 will be asserted and the 16 transmission gates will be on. Outputs from the

the eight write drivers will be passed on to bitlines of cells whose bitlines are connected with these transmission gates. The 8-bit data will be written to these eight SRAM cells.

3.4. SRAM Layout

3.4.1. Electrostatic Discharge Protection Circuit

ESD protection circuits need to be designed to protect the SRAM from electrostatic discharge damage. This is highly recommended by the foundry if the layout design is going to be taped out.

Static charge is created by insulator surfaces rubbing together or pulling apart. In this condition, one surface gains electrons and the other loses electrons. This results in an unbalanced electrical condition known as static charge [63]. When two surfaces with different static charge potentials touch each other, electrons move from one surface to the other, and this is called ESD.

Electrostatic Discharge phenomenon can be observed a lot in our daily lives. For example, people sometimes get a quick and mild shock when they grab the handle of a door in dry weathers, this is because of the static electricity got discharged the moment we touch the door handle. However, if this same amount of ESD stress is injected into a tiny memory chip, the instant large current from ESD can destroy a lot of circuits on the chip. In fact, a substantial number of IC failures are related with ESD events during the fabrication, test, and packaging process, which accounts for one third of the failures of IC [64]. ESD has become one of the major concerns for chip reliability. Thus, it is very important to have on-chip ESD protection circuits.

As shown in Figure 34, there are various kinds of ESD events that are different in voltage, current, and time. Generally, all ESD events are classified into three models for

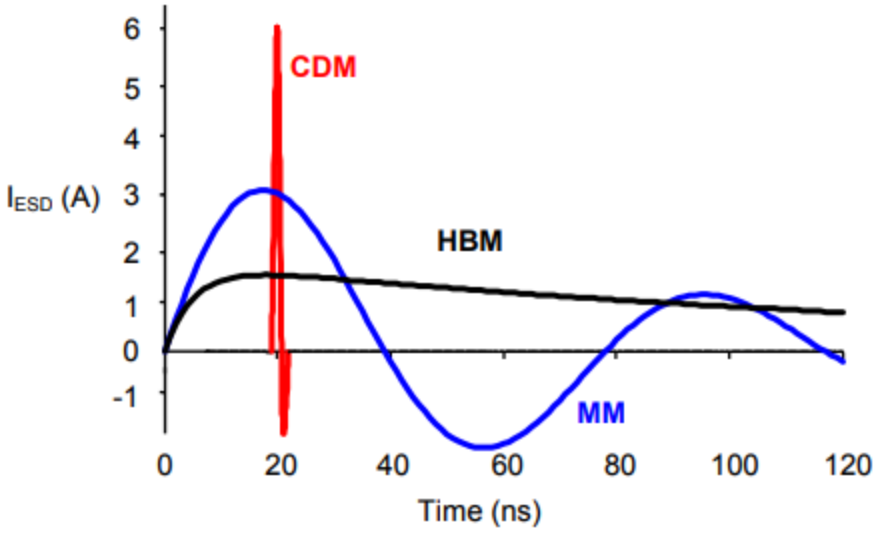


Figure 34. Current waveforms for HBM, MM, and CDM ESD events[3].

simulation and test. Namely, they are the human body model (HBM), the charge device model (CDM), and the machine model (MM). HBM simulates ESD events that occur when a charged human body contacts an electronic device and then the accumulated static charge discharges through the device to ground. HBM models the ESD event with series RC network with a 1500 Ω resistor and a 100 pF capacitor. Figure 34 shows the discharging current of the HBM event. CDM models ESD events that cannot be modeled by HBM. It models some ungrounded electronic devices that are self-charged during the fabrication or packaging process and these devices can discharge through ground pins upon contact. Figure 34 shows the discharging current of CDM event. MM is designed to simulate a charged machine discharges when touching IC during automatic testing. MM circuit model is similar to HBM model, only with different resistance and capacitance. The capacitance of the capacitor is 200pF while the resistor has a resistance close to zero. Figure 34 shows the discharge current of MM event.

[3] Source: http://www-tcad.stanford.edu/tcad/pubs/theses/chun.pdf

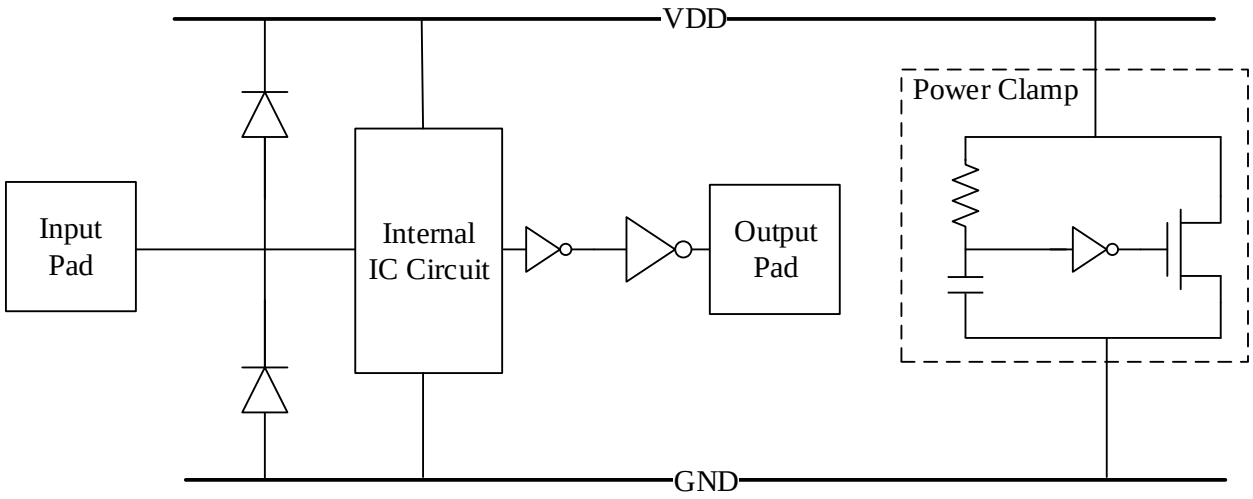


Figure 35. Chip ESD Protection Scheme.

ESD events can potentially bring hazards to transistors on a chip. If ESD discharge current flows through the internal IC circuit, it could possibly destroy the transistors in the internal circuit. For example, the MOSFET gate oxide breakdown voltage is normally much lower than the voltage generated during an ESD event and are very likely to be destroyed. So the main concept for ESD protection is to provide a low impedance path so that the large current generated in case of ESD event will not discharge through the internal IC circuit [65]. On this SRAM chip, ESD protection circuits can be divided into two parts:

1. Two large diodes which are commonly reverse-connected to input pads are added to protect the circuit from HBM ESD events. If there is a large positive voltage applied to input pad, the upper diode will be turned on the current will flow through the upper diode to VDD power rail. In case of a large negative voltage applied on the input pad, the lower diode will be turned on and current will discharge through the lower diode to ground. So, in either cases, the discharge current in HBM ESD events will not go through the internal IC circuit and destroy it.

2. Power clamp circuit is designed to handle the discharge current between VDD and ground power rails during various ESD events. NMOS in the power clamp is designed large enough to discharge large surge currents in different ESD events. During normal operation input of the inverter is disconnected from ground and connected to VDD through a resistor. So, it stays high and thus the voltage applied to gate of the NMOS stays low and the NMOS is cut off. There will be abrupt change on voltage and current between VDD and GND in case of ESD events and input of the inverter will be shorted to grounds and NMOS will be triggered to "on" state very quickly, as shown in Figure 35.

Layout of the SRAM is designed, the final layout is shown in Figure 36. It has a length of 1.1 mm and width of 0.97 mm.

3.5. Simulation Results

Post-layout simulation is performed. Due to limited resources, it is not possible to do an exhaustive simulation to read and write to all the cells. The worst case for read and write operation is when the SRAM critical path is the longest, which means it has the longest delay and most parasitics. From the timing control perspective, the cells that have the longest propagation delay face the highest read or write failure rate due to lack of sufficient read or write time. From the drivability perspective, the cells that have the largest parasitics have the high read and write failure rate due to insufficient drivability in the read or write path. So our test plan is to test the read and write for the worst case SRAM cells. If read and write in these cells are successful, read and write to the rest of cells will be successful too. To make sure that this is true, we also randomly select three other groups of SRAM cells to perform the read and write operations and check the results.

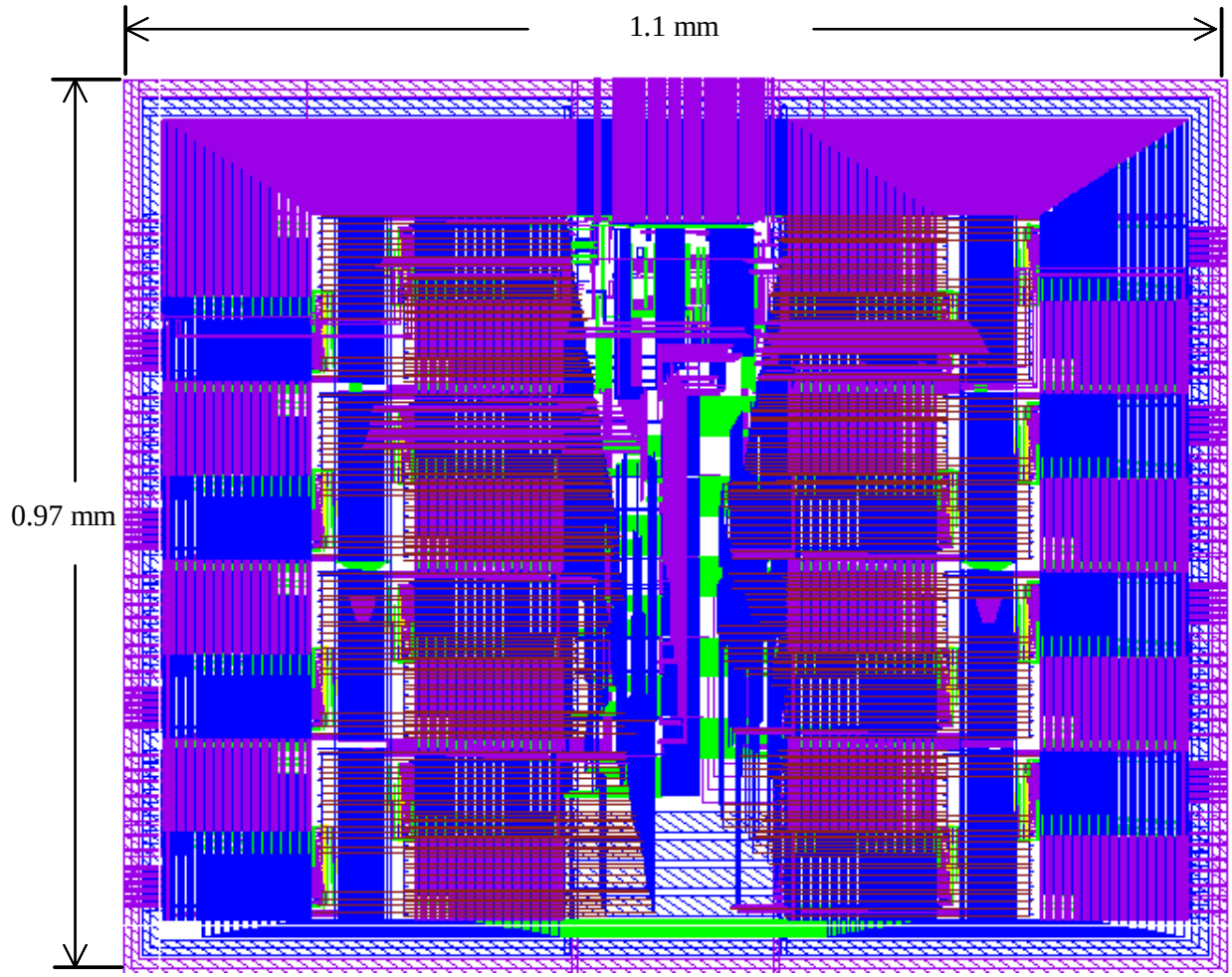


Figure 36. SRAM Layout.

Three groups of SRAM cells are tested for read and write. Each test includes first the write test and then the read test. To perform the write test, 16 8-bit data which is all set to be "01010101" is written to 16 address locations which will be read out later. The the read operations takes one of the read address set from the address patterns and read the 16 bytes out from those 16 locations which were just written to. The write and read is considered to be a success if the written data is exactly the same with the read data. Since the worst case among all SRAM blocks is block 15 according to the layout design. The worst-case test should include cells in this block. Address set {1472, 1487, 1502, 1517} meets this requirement and is chosen as the address set for the first test. Data "01010101" is written 16 times to address 1472, 1473,

1474, 1475, 1487, 1488, 1489, 1490, 1502, 1503, 1504, 1505, 1517, 1518, 1519, and 1520. Then the read operation starts and takes in the address set {1472, 1487, 1502, 1517}, 16 bytes of data will be read out. According to the post-layout simulation, the 16 bytes data read out from the SRAM is the same with the data that was previously written to the SRAM. Three more tests are also completed at address sets {776, 791, 806, 821}, {357, 364, 371, 378}, and {14, 21, 28, 35}. All the 16 byte read data turn out to be the same with the write data for these SRAM cells. According to post-layout simulations, the asynchronous SRAM functions correctly and meets all the design specifications.

4. CONCLUSION

This dissertation focuses on designing high-performance SRAM for emerging applications like 3D CPUs and CPUs for machine learning purpose. SRAM plays a critical role in the performance CPUs and optimizes the performance of SRAM and directly increases the processing speed of CPUs.

The dissertation introduces two research projects that have the same design goal, that is to boost the performance of SRAM for future microprocessors. Chapter two proposes a novel delay cell based on dummy TSVs. By using the dummy TSVs which are only used for heat dissipation purpose, the novel delay cells outperform traditional delay cells in areas like delay time stability, area, and heat dissipation. High performance SRAM is very sensitive to delay flucation due to very short clock cycle. Delay cells that can generate more stable delay time will be necessary for next generation high performance SRAM. Chapter three talks about a custom SRAM design for a machine learning processor. Machine learning is quickly gaining popularity and requires a huge number of processors that can support machining learning algorithms. However, processors at present time are designed for "general purpose" which is not very efficient for heavy convolutional calculation for machine learning purpose. Processors that are custom made are still at the early stage of their development. An attempt to design a custom designed processor for convolutional network for machine learning purpose is made by University or Arkansans, Otureo Technologies and our research group from NDSU. We as the team at NDSU lead the design of a custom designed SRAM that works well with the asynchronous processing cores. The SRAM and the processing cores work together as an asynchronous processor. And the processor is designed based on the convolutional neural network. Both the processing cores and the SRAM are optimized for machine learning purposes.

The processor is sent for fabrication and will be tested in the future to verify our design concepts as well as measure the power and the delay.

5. PUBLICATIONS

This work of research resulted in to multiple publications listed below:

Conference Publications

1. S. Nelson, C. Sherrill, J. Di, **X. Chen**, J. Wang, G. Sun, and A. Jia, "An Asynchronous Convolutional Neural Network Implementation for IoT Applications", 24th IEEE International Symposium on Asynchronous Circuits and Systems, Vinna, Austria, 2018.
2. J. Fu, **X. Chen**, M. Li, Z. Lin, L. Hou, F. Haring, N. Gong, J. Wang, "A Thermal Constraint Method Based on PCM for Integrated Circuits", Elsevier's Microelectronic. Engineering, 2017.
3. **X. Chen**, S. A. Pourbakhsh, N. Gong and J. Wang, "Closed-Form Delay Models for Buffer-driven TSVs in 3D On-chip Memory", 60th IEEE International Midwest Symposium on Circuits and Systems, Boston, MA, 2017.
4. J. Fu, S. A. Pourbakhsh, **X. Chen**, "On-Chip Thermal Management Method Based on Phase Change Material", 60th IEEE International Midwest Symposium on Circuits and Systems, Boston, MA, 2017.
5. **X. Chen**, S. A. Pourbakhsh, L. Hou, N. Gong and J. Wang, "3D memory design based on through silicon vias enabled timing optimization", 2016 5th International Symposium on Next-Generation Electronics (ISNE), Hsinchu, 2016.
6. **X. Chen**, S. A. Pourbakhsh, L. Hou, N. Gong and J. Wang, "Dummy TSV based bit-line optimization in 3D on-chip memory", 2016 IEEE International Conference on Electro Information Technology (EIT), Grand Forks, ND, 2016.

7. R. Ge, H. Pan, Z. Lin, N. Gong, J. Wang, and **X. Chen**, "RF-powered battery-less Wireless Sensor Network in structural monitoring", 2016 IEEE International Conference on Electro Information Technology (EIT), Grand Forks, ND, 2016.
8. S. A. Pourbakhsh, **X. Chen**, Dongliang Chen, Xin Wang, N. Gong and J. Wang, "Sizing-priority based low-power embedded memory for mobile video applications", 2016 17th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2016.
9. D. Chen, J. Edstrom, **X. Chen**, W. Xin, J. Wang, and N. Gong, "Data-Driven Low-Cost On-Chip Memory with Adaptive Power-Quality Trade-off for Mobile Video Streaming," International Symposium on Low Power Electronics and Design(ISLPED), San Francisco, CA, 2016.

Journal Publications

1. **X. Chen**, S. A. Pourbakhsh, J. Fu, N. Gong and J. Wang, "A Novel Hybrid Delay Unit based on Dummy TSVs for 3D On-chip Memory", IEEE Transactions on Very Large Scale Integration Systems, 2018, accepted.
2. N. Gong, S. A. Pourbakhsh, **X. Chen**, X. Wang, D. Chen, and J. Wang, "SPIDER: Sizing-Priority Based Application-Driven Memory for Mobile Video Applications", IEEE Transaction on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 9, pp. 2625-2634, Sept. 2017.

REFERENCES

[1] N. Gong, J. Wang, S. Jiang, et al., "TM-RF: Aging-Aware Power-Efficient Register File Design for Modern Microprocessors," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 7, July 2015, pp. 1196-1209.

[2] X. Dong, Y. Xie, N. Muralimanohar, et al., "Simple but Effective Heterogeneous Main Memory with On-Chip Memory Controller Support," International Conference for High Performance Computing, Networking, Storage and Analysis, November 2010, pp. 1-11.

[3] N. Gong, J. Wang, S. Jiang, et al., "Clock-biased local bit line for high performance register files," Electronics Letters, vol. 48, no. 18, August 2012, pp. 1104-1105.

[4] A. Singhvi, M. T. Moreira, R. N. Tadros, N. L. V. Calazans and P. A. Beerel, "A Fine-Grained, Uniform, Energy-Efficient Delay Element for FD-SOI Technologies," 2015 IEEE Computer Society Annual Symposium on VLSI, Montpellier, 2015, pp. 27-32.

[5] B. I. Abdulrazzaq, I. A. Halin, S. Kawahito, et al. “A Review on High-Resolution CMOS Delay Lines: Towards Sub-Picosecond Jitter Performance.” SpringerPlus, April 2016.

[6] M. Gharib and A. Abrishamifar, "A novel low-power and high-performance dual-loop DLL with linear delay element," 2008 51st Midwest Symposium on Circuits and Systems, Knoxville, TN, 2008, pp. 763-766.

[7] I. Mondal and N. Krishnapura, "A 2-GHz Bandwidth, 0.25–1.7 ns True-Time-Delay Element Using a Variable-Order All-Pass Filter Architecture in 0.13 um CMOS," IEEE Journal of Solid-State Circuits, vol. 52, no. 8, August 2017, pp. 2180-2193.

[8] P. Sharma and A. Gupta, "Decoder and pass transistor based digitally controlled linear delay element," 2014 International Conference on Advances in Computing, Communications and Informatics, New Delhi, September 2014, pp. 613-619.

[9] B. Saft, E. Schäfer, A. Jäger, et al., "An improved low-power CMOS thyristor-based micro-to-millisecond delay element," ESSCIRC 2014 - 40th European Solid-State Circuits Conference, Venice Lido, September 2014, pp. 123-126.

[10] Masakazu Shoji, Theory of CMOS Digital Circuits and Circuit Failures, 2014, Princeton Legacy Library, pp 407.

[11] STMicroelectronics, 'Timing Analysis in Presence of Supply Voltage and Temperature Variations', [Online]. Available: http://www.ispd.cc/slides/slides_backup/ispd06/1-2.pdf.

[12] M. M. D. Savio and M. A. Mary, "A clock tree based gated driver in low power delay buffer," 2016 International Conference on Electrical, Electronics, and Optimization Techniques, Chennai, March 2016, pp. 4767-4772.

[13] S. Lim, Design for High Performance, Low Power And Reliable 3D Integrated Circuits, Springer, 2012, pp. 553.

[14] T. Zhang, K. Wang, Y. Feng, et al., "A 3D SoC Design for H.264 Application with On-Chip DRAM Stacking," IEEE International 3D Systems Integration Conference, November 2010, pp. 1-6.

[15] V. F. Pavlidis and E. G. Friedman, Three-Dimensional Integrated Circuit Design, Morgan Kaufmann, 2008, pp. 180.

[16] K. Salah, Y. Ismail, and A. EI-Rouby, Arbitrary Modeling of TSVs for 3D Integrated Circuits, Springer, 2015, pp. 135.

[17] G. Luo, Y. Shi, and J. Cong, "An Analytical Placement Framework for 3-D ICs and Its Extension on Thermal Awareness," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 4, April 2013, pp. 510-523.

[18] V. F. Pavlidis and E. G. Friedman, Three-Dimensional Integrated Circuit Design, Morgan Kaufmann, 2008, pp. 193.

[19] Y. Tsai, Y. Xie, N. Vijaykrishnan, et al., “Three-dimensional cache design using 3DCacti,” International Conference on Computer Design and VLSI in Computers and Processors, October 2005, pp. 519-524.

[20] R. Qian, X. Yuan, and Zhibin Lin, Solid and Structural Analysis with the Finite Element Method, Nanjing: Southeast University Press, 2012, pp. 117.

[21] The MOSIS Service, ‘7HV CMOS Process’, [Online]. Available: https://www.mosis.com/vendors/view/global-foundries/7hv.

[22] NCSU EDA, ‘FreePDK45’, [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Contents.

[23] M. Watanabe, N. Niioka, T. Kobayashi, et al., "An effective model for evaluating vertical propagation delay in TSV-based 3-D ICs," International Symposium on Quality Electronic Design, March 2015, pp. 519-523.

[24] T. Bandyopadhyay, K. J. Han, D. Chung, et al., "Rigorous Electrical Modeling of Through Silicon Vias (TSVs) With MOS Capacitance Effects," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 1, no. 6, June 2011, pp. 893-903.

[25] X. Zhao, J. Minz, and S. K. Lim, "Low-Power and Reliable Clock Network Design for Through-Silicon Via (TSV) Based 3D ICs," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 1, no. 2, February 2011, pp. 247-259.

[26] W. Liu, Y. Wang, G. Chen, et al., "Whitespace-Aware TSV Arrangement in 3-D Clock Tree Synthesis," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 9, September, 2015, pp. 1842-1853.

[27] H. Park and T. Kim, "Synthesis of TSV Fault-Tolerant 3-D Clock Trees," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 2, February 2015, pp. 266-279.

[28] Y. Shang, C. Zhang, H. Yu, et al., "Thermal-reliable 3D clock-tree synthesis considering nonlinear electrical-thermal-coupled TSV model," Asia and South Pacific Design Automation Conference (ASP-DAC), Jaunary 2013, pp. 693-698.

[29] M. P. D. Sai, H. Yu, Y. Shang, et al., "Reliable 3-D Clock-Tree Synthesis Considering Nonlinear Capacitive TSV Model With Electrical–Thermal–Mechanical Coupling," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 11, November 2013, pp. 1734-1747.

[30] X. Chen, S. A. Pourbakhsh, L. Hou, et al., “Dummy TSV Based Bit-line Optimization in 3D On-chip Memory,” International Electro/Information Technology Conference, May 2016.

[31] A.W. Topol, J.D. C.L. Tulipe, L. Shi, et al., Three-dimensional integrated circuits. IBM J. Res. Dev. 50, 491-506(2006).

[32] H. Sung, K. Cho, K. Yoon, et al., "A Delay Test Architecture for TSV With Resistive Open Defects in 3-D Stacked Memories," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 11, November 2014, pp. 2380-2387.

[33] K. Salah, A. El-Rouby, Y. Ismail, et al., "Compact TSV Modeling for Low Power Application," International Conference on Energy Aware Computing (ICEAC), December 2010, pp. 1-2.

[34] Y. J. Lee and S. K. Lim, "Fast Delay Estimation with Buffer Insertion for Through-Silicon-Via-Based 3D Interconnects," International Symposium on Quality Electronic Design (ISQED), March 2012, pp. 228-335.

[35] K. Athikulwongse, A. Chakraborty, J. S. Yang, et al., "Stress-driven 3D-IC placement with TSV keep-out zone and regularity study," 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), November 2010, pp. 669-674.

[36] G. Katti, M. Stucchi, K. De Meyer, et al., "Electrical Modeling and Characterization of Through Silicon Via for Three-Dimensional ICs," IEEE Transaction on Electron Devices, vol. 57, no. 1, January 2010, pp. 256–262.

[37] J. Kim et al., "High-Frequency Scalable Modeling and Analysis of a Differential Signal Through-Silicon Via," in IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 4, no. 4, pp. 697-707, April 2014.

[38] J. W. You, S. Y. Huang, D. M. Kwai, et al., "Performance Characterization of TSV in 3D IC Via Sensitivity Analysis," Asian Test Symposium, December 2010, pp. 389–394.

[39] VlSI System Design, 'Interconnect Scaling Trends', [Online]. Available: http://vlsisystemdesign.com/scaling.php.

[40] J. M. Rabaey, "Electrical wire models," Digital Integrated Circuit: A Design Perspective, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2003, pp. 150–159.

[41] J. Wang, N. Gong, and E. G. Friedman, "PNS-FCR: Flexible Charge Recycling Dynamic Circuit Technique for Low-Power Microprocessors," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, February 2016, pp. 613-624.

[42] N. Gong, J. Wang, and R. Sridhar, "Variation Aware Sleep Vector Selection in Dual Vt Dynamic OR Circuits for Low Leakage Register File Design," IEEE Transactions on Circuits and Systems I, vol. 61, no. 7, July 2014, pp. 1970-1983.

[43] V. Adler and E. G. Friedman, "Delay and power expressions for a CMOS inverter driving a resistive-capacitive load," Analog Integrated Circuits for Signal Processing, vol. 14, no. 1/2, September 1997, pp. 29–40.

[44] Chalmers, 'Inverter Speed and Propagation Delay', [Online]. Available:http://pingpong.chalmers.se/public/pp/public_courses/course01705/published/1349354335911/resourceId/947092/content/Chapter%204%20preliminary.pdf.

[45] K. T. Tang and E. G. Friedman, "Delay uncertainty due to on-chip simultaneous switching noise in high performance CMOS integrated circuits," IEEE Workshop on Signal Processing Systems, October 2000, pp. 633-642.

[46] X. Wang, M. Tehranipoor, and R. Datta, "Path-RO: A novel on-chip critical path delay measurement under process variations," 2008 IEEE/ACM International Conference on Computer-Aided Design, November 2008, pp. 640-646.

[47] S. Biswas, G. Wang, R. Kastner, et al., "Combining static and dynamic defect-tolerance techniques for nanoscale memory systems," IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, November 2007, pp. 773-778.

[48] S. K. Lim, Design for High Performance, Low Power, and Reliable 3D Integrated Circuits, Springer, 2013, pp 145.

[49] S. Piersanti, F. de Paulis, and C. Olivieri, "Localization of Short and Open Defects in Multilayer Through Silicon Vias (TSV) Daisy-Chain Structures," IEEE Transactions on Electromagnetic Compatibility, vol. 59, no. 5, October 2017, pp. 1558-1564.

[50] Y. Chen, D. Niu, Y. Xie, et al., "Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis," 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, November 2010, pp. 471-476.

[51] M. Zahran, "Graphics Processing Units (GPUs): Architecture and Programming", [Online].Available:http://cs.nyu.edu/courses/spring12/CSCIGA.3033012/lecture12.pdf.

[52] H. Kalargaris, Y. C. Chen, and V. F. Pavlidis, "STA compatible backend design flow for TSV-based 3-D ICs," 18th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, March 2017, pp. 186-190.

[53] T. Thorolfsson, K. Gonsalves, and P. D. Franzon, "Design automation for a 3DIC FFT processor for synthetic aperture radar: A case study," 2009 46th ACM/IEEE Design Automation Conference, San Francisco, CA, November 2009, pp. 51-56.

[54] Wulong Liu et al., "TSV-aware topology generation for 3D Clock Tree Synthesis," International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, March 2013, pp. 300-307.

[55] J. Cong and G. Luo, "A Multilevel Analytical Placement for 3D ICs," IEEE Asia and South Pacific Design Automation Conference, November 2009, pp. 361–366.

[56] P. Budhathoki, A. Henschel, and I. A. M. Elfadel, "Thermal-driven 3D floorplanning using localized TSV placement," 2014 IEEE International Conference on IC Design & Technology, Austin, TX, May 2014, pp. 1-4.

[57] Q. Ma, D. Leenaerts, and R. Mahmoudi, "A 12ps true-time-delay phase shifter with 6.6% delay variation at 20–40GHz," 2013 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), Seattle, WA, 2013, pp. 61-64.

[58] A. Ahmed, S. M. S. Rashid and A. B. M. H. Rashid, "A 200PS differential CMOS delay element for 10 GHz ultra wideband wireless receivers," 2012 7th International Conference on Electrical and Computer Engineering, Dhaka, 2012, pp. 713-716.

[59] V. K. Agarwal, M. Guduri and A. Islam, "Power and variability analysis of CMOS logic families @ 22-nm technology node," Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization, Noida, 2014, pp. 1-6.

[60] S. Tyagi, A. K. Dwivedi, B. B. Pal and A. Islam, "Revisiting performance of various delay elements to realize a trigger pulse generator," 2015 International Conference on Smart Sensors and Systems (IC-SSS), Bangalore, 2015, pp. 1-6.

[61] S. K. Garakoui, E. A. M. Klumperink, B. Nauta and F. E. et al., "Compact Cascadable g m -C All-Pass True Time Delay Cell with Reduced Delay Variation Over Frequency," IEEE Journal of Solid-State Circuits, vol. 50, no. 3, March 2015, pp. 693-703.

[62] L. Fuller, “Static Random Access Memory”, [Online]. Available: https://people.rit.edu/lffeee/SRAM.pdf.

[63] Texas Instrument, “Electrostatic Discharge”, [Online]. Available: http://www.ti.com/lit/an/ssya010a/ssya010a.pdf.

[64] J. Chun, “ESD Protection Circuits for Advanced CMOS Technologies”, Ph.D. Dissertation, Standford University, June 2006.

[65] A.Z Wang, H.G Feng, K Gong, R.Y Zhan, J Stine, On-chip ESD protection design for integrated circuits: an overview for IC designers, Microelectronics Journal, Volume 32, Issue 9, 2001, Pages 733-747, ISSN 0026-2692.

APPENDIX A. BUFFER SIZE PREDICTION MODEL

DTDUs have a uniform structure as shown in Figure 4 (A). Size of dummy TSVs is given in Section 2.3.1. However, sizes of 6 buffers in three DTDUs have yet to be decided. Therefore, the mathematical models are proposed to predict their sizes. As shown in Figure 4, two buffers used in the same DTDU have the same size.

Delay of L-1 DTDU TL-1 can be divided into two parts:

$$T_{L-1} = T_{1-2} + T_{2-3} \tag{A1}$$

As shown in Figure 4 (A), T_{1-2} and T_{2-3} are respectively the delay time from point 1 to point 2 and from point 2 to point 3. Equivalent circuit from point 1 to point 2 can be derived as an inverter driving a dummy TSV, which is an RC load and B2 gate capacitance C_{B2}, as shown in Figure 4 (B). Equivalent circuit from point 2 to point 3 can be derived as an inverter driving two capacitive loads - B3 gate capacitance C_{B3} and L-1 CLK capacitance C_{CLK}, as shown in Figure 4 (C).

According to [43], the propagation delay T_{1-2} is calculated as

$$T_{1-2} = 0.693 \frac{C_{Load1} + \frac{I_{S1}}{V_{DD}} R_{Total} C_{TSV}}{\frac{I_{S1}}{V_{DD}}} \tag{A2}$$

where C_{Load1} is the capacitive load of B1, R_{Total} is the total resistance of a dummy TSV and the wire connecting to it.

$$C_{Load1} = C_{TSV} + C_{B2} \tag{A3}$$

$$C_{B2} = C_{ox} \times W_{B2} \times L_{B2} \tag{A4}$$

where CB2, WB2, and L_{B2} are gate capacitance, width, and length of B2 respectively.

$$I_{S1} = 0.5 \mu_n C_{ox} \frac{W_{B1}}{L_{B1}} (V_{DD} - V_{th})^2 \tag{A5}$$

where I_{S1} represents the saturation current of buffer B1, and V_{DD} represents the drain-to-source voltage at saturation. W_{B1} and L_{B1} are width and length of buffer B1 respectively. μ_n is the electron mobility of NMOS, C_{ox} is the oxide capacitance per unit, and V_{th} is the threshold voltage of an NMOS.

According to [44], delay from point 2 to point 3 can also be written as:

$$T_{2-3} = T_{B2} + slope \times T_{1-2} \tag{A6}$$

$$T_{B2} = 0.7 \times \frac{C_{load2} \times V_{DD}}{I_{S1}} \tag{A7}$$

where *slope* is a constant. C_{load2} is capacitive load of B2, which includes C_{B3} and gate capacitances of 32 PMOS connecting to L-1 CLK - C_{P1}, as shown in Figure 2 (C). T_{B2} is the delay of B2.

$$C_{B3} = C_{ox} \times W_{B3} \times L_{B3} \tag{A8}$$

where C_{B3}, W_{B3}, and L_{B3} are gate capacitance, width, and length of B3 respectively. Width and length of each PMOS are W_{P1} and L_{P1} and gate capacitance of 32 PMOS is

$$C_{P1} = 32 \times C_{ox} \times W_{P1} \times L_{P1} \tag{A9}$$

$$C_{load2} = C_{B3} + C_{P1} \tag{A10}$$

Delay of L-2 DTDU can also be divided into two parts:

$$T_{L-2} = T_{3-4} + T_{4-5} \tag{A11}$$

where T_{3-4} and T_{4-5} are respectively the delay time from point 3 to point 4 and from point 4 to point 5.

$$T_{3-4} = T_{Inv\_RC2} + slope \times T_{2-3} \tag{A12}$$

$$T_{4-5} = T_{B4} + slope \times T_{3-4} \tag{A13}$$

T_{Inv_RC2} is same as $T_{1\text{-}2}$ and calculated from (A3). T_{B4}, which is delay of B6 and same as T_{B2}, can be calculated from (A8).

Delay of L-3 DTDU can also be divided into two parts:

$$T_{L-3} = T_{5-6} + T_{6-7} \tag{A14}$$

where $T_{5\text{-}6}$ and $T_{6\text{-}7}$ are respectively the delay time from point 5 to point 6 and from point 6 to point 7.

$$T_{5-6} = T_{Inv\_RC3} + slope \times T_{4-5} \tag{A15}$$

$$T_{6-7} = T_{B6} + slope \times T_{5-6} \tag{A16}$$

T_{Inv_RC3} is same as $T_{1\text{-}2}$ and calculated from (A3). T_{B6}, which is delay of B6 and same as T_{B2}, can be calculated from (A8).

Equations (A1) to (A10), (A11) to (A13), and (A14) to (A16) mathematically model the relations between L-1 DTDU delay time and buffer B1 and B2 size, between L-2 DTDU delay time and buffer B3 and B4 size, and between L-3 DTDU delay time and buffer B5 and B6 size. The expected delay time of L-1 DTDU, L-2 DTDU, and L-3 DTDU determine sizes of B1-B6.

APPENDIX B. ADDRESS PATTERN (SAMPLE)

Access #	Write Addr	Read Addr 0	Read Addr 1	Read Addr 2	Read Addr 3
1125		735	750	765	780
1126		960	975	990	1005
1127		1185	1200	1215	1230
1128		1410	1425	1440	1455
1129		1635	1650	1665	1680
1130	0				
1131		735	750	765	780
1132		960	975	990	1005
1133		1185	1200	1215	1230
1134		1410	1425	1440	1455
1135		1635	1650	1665	1680
1136	49				
1137		735	750	765	780
1138		960	975	990	1005
1139		1185	1200	1215	1230
1140		1410	1425	1440	1455
1141		1635	1650	1665	1680
1142	98				
1143		735	750	765	780
1144		960	975	990	1005
1145		1185	1200	1215	1230
1146		1410	1425	1440	1455
1147		1635	1650	1665	1680
1148	147				
1149		735	750	765	780
1150		960	975	990	1005
1151		1185	1200	1215	1230
1152		1410	1425	1440	1455
1153		1635	1650	1665	1680
1154	196				
1155		735	750	765	780
1156		960	975	990	1005
1157		1185	1200	1215	1230
1158		1410	1425	1440	1455
1159		1635	1650	1665	1680
1160	245				
1161		735	750	765	780
1162		960	975	990	1005
1163		1185	1200	1215	1230
1164		1410	1425	1440	1455
1165		1635	1650	1665	1680
1166	294				
1167		735	750	765	780
1168		960	975	990	1005
1169		1185	1200	1215	1230
1170		1410	1425	1440	1455

1171		1635	1650	1665	1680
1172	343				
1173		735	750	765	780
1174		960	975	990	1005
1175		1185	1200	1215	1230
1176		1410	1425	1440	1455
1177		1635	1650	1665	1680
1178	392				
1179		735	750	765	780
1180		960	975	990	1005
1181		1185	1200	1215	1230
1182		1410	1425	1440	1455
1183		1635	1650	1665	1680
1184	441				
1185		735	750	765	780
1186		960	975	990	1005
1187		1185	1200	1215	1230
1188		1410	1425	1440	1455
1189		1635	1650	1665	1680
1190	490				
1191		735	750	765	780
1192		960	975	990	1005
1193		1185	1200	1215	1230
1194		1410	1425	1440	1455
1195		1635	1650	1665	1680
1196	539				
1197		735	750	765	780
1198		960	975	990	1005
1199		1185	1200	1215	1230
1200		1410	1425	1440	1455
1201		1635	1650	1665	1680
1202	588				
1203		735	750	765	780
1204		960	975	990	1005
1205		1185	1200	1215	1230
1206		1410	1425	1440	1455
1207		1635	1650	1665	1680
1208	637				
1209		735	750	765	780
1210		960	975	990	1005
1211		1185	1200	1215	1230
1212		1410	1425	1440	1455
1213		1635	1650	1665	1680
1214	686				
1215		737	752	767	782
1216		962	977	992	1007
1217		1187	1202	1217	1232
1218		1412	1427	1442	1457
1219		1637	1652	1667	1682
1220	1				

1221	0	737	752	767	782
1222	0	962	977	992	1007
1223	0	1187	1202	1217	1232
1224	0	1412	1427	1442	1457
1225	0	1637	1652	1667	1682
1226	50				
1227	0	737	752	767	782
1228	0	962	977	992	1007
1229	0	1187	1202	1217	1232
1230	0	1412	1427	1442	1457
1231	0	1637	1652	1667	1682
1232	99				
1233	0	737	752	767	782
1234	0	962	977	992	1007
1235	0	1187	1202	1217	1232
1236	0	1412	1427	1442	1457
1237	0	1637	1652	1667	1682
1238	148				
1239	0	737	752	767	782
1240	0	962	977	992	1007
1241	0	1187	1202	1217	1232
1242	0	1412	1427	1442	1457
1243	0	1637	1652	1667	1682
1244	197				
1245	0	737	752	767	782
1246	0	962	977	992	1007
1247	0	1187	1202	1217	1232
1248	0	1412	1427	1442	1457
1249	0	1637	1652	1667	1682
1250	246				
1251	0	737	752	767	782
1252	0	962	977	992	1007
1253	0	1187	1202	1217	1232
1254	0	1412	1427	1442	1457
1255	0	1637	1652	1667	1682
1256	295				
1257	0	737	752	767	782
1258	0	962	977	992	1007
1259	0	1187	1202	1217	1232
1260	0	1412	1427	1442	1457
1261	0	1637	1652	1667	1682
1262	344				
1263	0	737	752	767	782
1264	0	962	977	992	1007
1265	0	1187	1202	1217	1232
1266	0	1412	1427	1442	1457
1267	0	1637	1652	1667	1682
1268	393				
1269	0	737	752	767	782
1270	0	962	977	992	1007

1271	0	1187	1202	1217	1232
1272	0	1412	1427	1442	1457
1273	0	1637	1652	1667	1682
1274	442				
1275	0	737	752	767	782
1276	0	962	977	992	1007
1277	0	1187	1202	1217	1232
1278	0	1412	1427	1442	1457
1279	0	1637	1652	1667	1682
1280	491				
1281	0	737	752	767	782
1282	0	962	977	992	1007
1283	0	1187	1202	1217	1232
1284	0	1412	1427	1442	1457
1285	0	1637	1652	1667	1682
1286	540				
1287	0	737	752	767	782
1288	0	962	977	992	1007
1289	0	1187	1202	1217	1232
1290	0	1412	1427	1442	1457
1291	0	1637	1652	1667	1682
1292	589				
1293	0	737	752	767	782
1294	0	962	977	992	1007
1295	0	1187	1202	1217	1232
1296	0	1412	1427	1442	1457
1297	0	1637	1652	1667	1682
1298	638				
1299	0	737	752	767	782
1300	0	962	977	992	1007
1301	0	1187	1202	1217	1232
1302	0	1412	1427	1442	1457
1303	0	1637	1652	1667	1682
1304	687				
1305	0	739	754	769	784
1306	0	964	979	994	1009
1307	0	1189	1204	1219	1234
1308	0	1414	1429	1444	1459
1309	0	1639	1654	1669	1684
1310	2				
1311	0	739	754	769	784
1312	0	964	979	994	1009
1313	0	1189	1204	1219	1234
1314	0	1414	1429	1444	1459
1315	0	1639	1654	1669	1684
1316	51				
1317	0	739	754	769	784
1318	0	964	979	994	1009
1319	0	1189	1204	1219	1234
1320	0	1414	1429	1444	1459

1321	0	1639	1654	1669	1684
1322	100				
1323	0	739	754	769	784
1324	0	964	979	994	1009
1325	0	1189	1204	1219	1234
1326	0	1414	1429	1444	1459
1327	0	1639	1654	1669	1684
1328	149				
1329	0	739	754	769	784
1330	0	964	979	994	1009
1331	0	1189	1204	1219	1234
1332	0	1414	1429	1444	1459
1333	0	1639	1654	1669	1684
1334	198				
1335	0	739	754	769	784
1336	0	964	979	994	1009
1337	0	1189	1204	1219	1234
1338	0	1414	1429	1444	1459
1339	0	1639	1654	1669	1684
1340	247				
1341	0	739	754	769	784
1342	0	964	979	994	1009
1343	0	1189	1204	1219	1234
1344	0	1414	1429	1444	1459
1345	0	1639	1654	1669	1684
1346	296				
1347	0	739	754	769	784
1348	0	964	979	994	1009
1349	0	1189	1204	1219	1234
1350	0	1414	1429	1444	1459
1351	0	1639	1654	1669	1684
1352	345				
1353	0	739	754	769	784
1354	0	964	979	994	1009
1355	0	1189	1204	1219	1234
1356	0	1414	1429	1444	1459
1357	0	1639	1654	1669	1684
1358	394				
1359	0	739	754	769	784
1360	0	964	979	994	1009
1361	0	1189	1204	1219	1234
1362	0	1414	1429	1444	1459
1363	0	1639	1654	1669	1684
1364	443				
1365	0	739	754	769	784
1366	0	964	979	994	1009
1367	0	1189	1204	1219	1234
1368	0	1414	1429	1444	1459
1369	0	1639	1654	1669	1684
1370	492				

6000		2	9	16	23
6001		51	58	65	72
6002		100	107	114	121
6003		149	156	163	170
6004		198	205	212	219
6005		247	254	261	268
6006		296	303	310	317
6007		345	352	359	366
6008		394	401	408	415
6009		443	450	457	464
6010		492	499	506	513
6011		541	548	555	562
6012		590	597	604	611
6013		639	646	653	660
6014		688	695	702	709
6015		2	9	16	23
6016		51	58	65	72
6017		100	107	114	121
6018		149	156	163	170
6019		198	205	212	219
6020		247	254	261	268
6021		296	303	310	317
6022		345	352	359	366
6023		394	401	408	415
6024		443	450	457	464
6025		492	499	506	513
6026		541	548	555	562
6027		590	597	604	611
6028		639	646	653	660
6029		688	695	702	709
6030		2	9	16	23
6031		51	58	65	72
6032		100	107	114	121
6033		149	156	163	170
6034		198	205	212	219
6035		247	254	261	268
6036		296	303	310	317
6037		345	352	359	366
6038		394	401	408	415
6039		443	450	457	464
6040		492	499	506	513
6041		541	548	555	562
6042		590	597	604	611
6043		639	646	653	660
6044		688	695	702	709
6045		2	9	16	23
6046		51	58	65	72
6047		100	107	114	121
6048		149	156	163	170
6049		198	205	212	219
6050		247	254	261	268

6051		296	303	310	317
6052		345	352	359	366
6053		394	401	408	415
6054		443	450	457	464
6055		492	499	506	513
6056		541	548	555	562
6057		590	597	604	611
6058		639	646	653	660
6059		688	695	702	709
6060		2	9	16	23
6061		51	58	65	72
6062		100	107	114	121
6063		149	156	163	170
6064		198	205	212	219
6065		247	254	261	268
6066		296	303	310	317
6067		345	352	359	366
6068		394	401	408	415
6069		443	450	457	464
6070		492	499	506	513
6071		541	548	555	562
6072		590	597	604	611
6073		639	646	653	660
6074		688	695	702	709
6075		2	9	16	23
6076		51	58	65	72
6077		100	107	114	121
6078		149	156	163	170
6079		198	205	212	219
6080		247	254	261	268
6081		296	303	310	317
6082		345	352	359	366
6083		394	401	408	415
6084		443	450	457	464
6085		492	499	506	513
6086		541	548	555	562
6087		590	597	604	611
6088		639	646	653	660
6089		688	695	702	709
6090		2	9	16	23
6091		51	58	65	72
6092		100	107	114	121
6093		149	156	163	170
6094		198	205	212	219
6095		247	254	261	268
6096		296	303	310	317
6097		345	352	359	366
6098		394	401	408	415
6099		443	450	457	464
6100		492	499	506	513

6101		541	548	555	562
6102		590	597	604	611
6103		639	646	653	660
6104		688	695	702	709
6105		2	9	16	23
6106		51	58	65	72
6107		100	107	114	121
6108		149	156	163	170
6109		198	205	212	219
6110		247	254	261	268
6111		296	303	310	317
6112		345	352	359	366
6113		394	401	408	415
6114		443	450	457	464
6115		492	499	506	513
6116		541	548	555	562
6117		590	597	604	611
6118		639	646	653	660
6119		688	695	702	709
6120		2	9	16	23
6121		51	58	65	72
6122		100	107	114	121
6123		149	156	163	170
6124		198	205	212	219
6125		247	254	261	268
6126		296	303	310	317
6127		345	352	359	366
6128		394	401	408	415
6129		443	450	457	464
6130		492	499	506	513
6131		541	548	555	562
6132		590	597	604	611
6133		639	646	653	660
6134		688	695	702	709
6135		3	10	17	24
6136		52	59	66	73
6137		101	108	115	122
6138		150	157	164	171
6139		199	206	213	220
6140		248	255	262	269
6141		297	304	311	318
6142		346	353	360	367
6143		395	402	409	416
6144		444	451	458	465
6145		493	500	507	514
6146		542	549	556	563
6147		591	598	605	612
6148		640	647	654	661
6149		689	696	703	710
6150		3	10	17	24

6151		52	59	66	73
6152		101	108	115	122
6153		150	157	164	171
6154		199	206	213	220
6155		248	255	262	269
6156		297	304	311	318
6157		346	353	360	367
6158		395	402	409	416
6159		444	451	458	465
6160		493	500	507	514
6161		542	549	556	563
6162		591	598	605	612
6163		640	647	654	661
6164		689	696	703	710
6165		3	10	17	24
6166		52	59	66	73
6167		101	108	115	122
6168		150	157	164	171
6169		199	206	213	220
6170		248	255	262	269
6171		297	304	311	318
6172		346	353	360	367
6173		395	402	409	416
6174		444	451	458	465
6175		493	500	507	514
6176		542	549	556	563
6177		591	598	605	612
6178		640	647	654	661
6179		689	696	703	710
6180		3	10	17	24
6181		52	59	66	73
6182		101	108	115	122
6183		150	157	164	171
6184		199	206	213	220
6185		248	255	262	269
6186		297	304	311	318
6187		346	353	360	367
6188		395	402	409	416
6189		444	451	458	465
6190		493	500	507	514
6191		542	549	556	563
6192		591	598	605	612
6193		640	647	654	661
6194		689	696	703	710
6195		3	10	17	24
6196		52	59	66	73
6197		101	108	115	122
6198		150	157	164	171
6199		199	206	213	220
6200		248	255	262	269

6201		297	304	311	318
6202		346	353	360	367
6203		395	402	409	416
6204		444	451	458	465
6205		493	500	507	514
6206		542	549	556	563
6207		591	598	605	612
6208		640	647	654	661
6209		689	696	703	710
6210		3	10	17	24
6211		52	59	66	73
6212		101	108	115	122
6213		150	157	164	171
6214		199	206	213	220
6215		248	255	262	269
6216		297	304	311	318
6217		346	353	360	367
6218		395	402	409	416
6219		444	451	458	465
6220		493	500	507	514
6221		542	549	556	563
6222		591	598	605	612
6223		640	647	654	661
6224		689	696	703	710
6225		3	10	17	24
6226		52	59	66	73
6227		101	108	115	122
6228		150	157	164	171
6229		199	206	213	220
6230		248	255	262	269
6231		297	304	311	318
6232		346	353	360	367
6233		395	402	409	416
6234		444	451	458	465
6235		493	500	507	514
6236		542	549	556	563
6237		591	598	605	612
6238		640	647	654	661
6239		689	696	703	710
6240		3	10	17	24
6241		52	59	66	73
6242		101	108	115	122
6243		150	157	164	171
6244		199	206	213	220
6245		248	255	262	269
6246		297	304	311	318
6247		346	353	360	367
6248		395	402	409	416
6249		444	451	458	465
6250		493	500	507	514

6251		542	549	556	563
6252		591	598	605	612
6253		640	647	654	661
6254		689	696	703	710
6255		3	10	17	24
6256		52	59	66	73
6257		101	108	115	122
6258		150	157	164	171
6259		199	206	213	220
6260		248	255	262	269
6261		297	304	311	318
6262		346	353	360	367
6263		395	402	409	416
6264		444	451	458	465
6265		493	500	507	514
6266		542	549	556	563
6267		591	598	605	612
6268		640	647	654	661
6269		689	696	703	710
6270		3	10	17	24
6271		52	59	66	73
6272		101	108	115	122
6273		150	157	164	171
6274		199	206	213	220
6275		248	255	262	269
6276		297	304	311	318
6277		346	353	360	367
6278		395	402	409	416
6279		444	451	458	465
6280		493	500	507	514
6281		542	549	556	563
6282		591	598	605	612
6283		640	647	654	661
6284		689	696	703	710
6285		3	10	17	24
6286		52	59	66	73
6287		101	108	115	122
6288		150	157	164	171
6289		199	206	213	220
6290		248	255	262	269
6291		297	304	311	318
6292		346	353	360	367
6293		395	402	409	416
6294		444	451	458	465
6295		493	500	507	514
6296		542	549	556	563
6297		591	598	605	612
6298		640	647	654	661
6299		689	696	703	710
6300		3	10	17	24