DUMMY TSV-BASED TIMING OPTIMIZATION FOR 3D ON-CHIP MEMORY

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

By

Seyed Alireza Pourbakhsh

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

July 2016

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Dummy TSV-based Timing Optimization for 3D on-chip Memory

By

Seyed Alireza Pourbakhsh

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

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Jinhui Wang
Chair

Na Gong

Zhibin Lin

Approved:

7/10/2016
Date

Scott Smith
Department Chair

ABSTRACT

Design and fabrication of three-dimensional (3D) ICs is one the newest and hottest trends in semiconductor manufacturing industry. In 3D ICs, multiple 2D silicon dies are stacked vertically, and through silicon vias (TSVs) are used to transfer power and signals between different dies.

The electrical characteristic of TSVs can be modeled with equivalent circuits consisted of passive elements. In this thesis, we use “dummy” TSVs as electrical delay units in 3D SRAMs. Our results prove that dummy TSVs based delay units are as effective as conventional delay cells in performance, increase the operational frequency of SRAM up to 110%, reduce the usage of silicon area up to 88%, induce negligible power overhead, and improve robustness against voltage supply variation and fluctuation.

ACKNOWLEDGEMENTS

I am deeply grateful to Dr. Jinhui Wang, my advisor for providing me the opportunity to work under his supervision and for his invaluable support, enthusiasm, and guidance. I would also like to thank Dr. Na Gong and Dr. Zhibin Lin for having accepted to be members of my supervisory committee. I have used the suggestions and assistance provided by them numerous times.

I would specially like to thank my colleagues and fellow students Xiaowei Chen, Dongliang Chen, Xin Wang, Peng Gao, Yifu Gong, Ruisi Ge, and Jonathon Edstrom for their contributions in this study, and constant encouragement and support.

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

DEDICATION

Dedicated to:

My father and mother, Seyed Ali and Fatemeh,

My brothers and sister, Seyed Hadi, Seyed Hamed, and Zahra Sadat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D..Two Dimensional

3D..Three Dimensional

BEOL ....Back End of Line

CMOS ....Complementary Metal-Oxide-Semiconductor

CPU..Central Processing Unit

FEOL...Front End of Line

FinFET ....Fin Field Effect Transistor

IC..Integrated Circuit

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

NAND ....Negative AND

NCSU ....North Carolina State University

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

NOR ....Negative OR

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

PTM ....Predictive Technology Model

SOC...System on Chip

SRAM ....Static Random Access Memory

TSV ....Through Silicon Via

VLSI..Very Large Scale Integration

1. INTRODUCTION

1.1. Trends in Designing More Efficient Integrated Circuits

There are multiple desirable attributes in designing and manufacturing CMOS integrated circuits. These attributes include higher speed, lower power consumption, and smaller size. In the past decades, the most common way to achieve the goals in having better and improved CMOS integrated circuits is using smaller size transistors. In 1965, Gordon E. Moore predicted that the number of transistors in integrated circuits will double every two years. This prediction is known as Moore's Law. This doubling of the transistor count is the result of using smaller transistors.

Since 1970s, the size of transistors used in integrated circuits has continuously shrunk, and as a results, now we have faster, more power efficient integrated circuits. The transistor count in Intel CPUs from 1971 to 2015 is shown in Figure 1. The vertical axis in the graph is logarithmic, and the trend in data is linear. This means that the number of transistors are growing exponentially. The die size of Intel CPUs is illustrated in Figure 2. The plot in Figure 3 illustrates the exponential trend in transistor density. Density of transistors is the transistor count divided by the die size. As the size of transistors shrinks, the speed and performance of the processors also improve. Figure 4 illustrates the maximum clock frequency of Intel CPUs from 1970 to 2015.

It can be seen in Figure 4 that the maximum clock frequency in Intel CPUs has been stagnating in recent years. This is due to challenges the designers have to face when deep submicron technologies are used. In recent technology nodes, the transistors are faster than the previous technology nodes, but now the interconnects and metal layers are the limiting factor for reducing delay and increasing the frequency of integrated circuits. Because of these problems, new methods are needed in semiconductor industry. 3D IC is one of the methods that offers solutions for some of these problems.

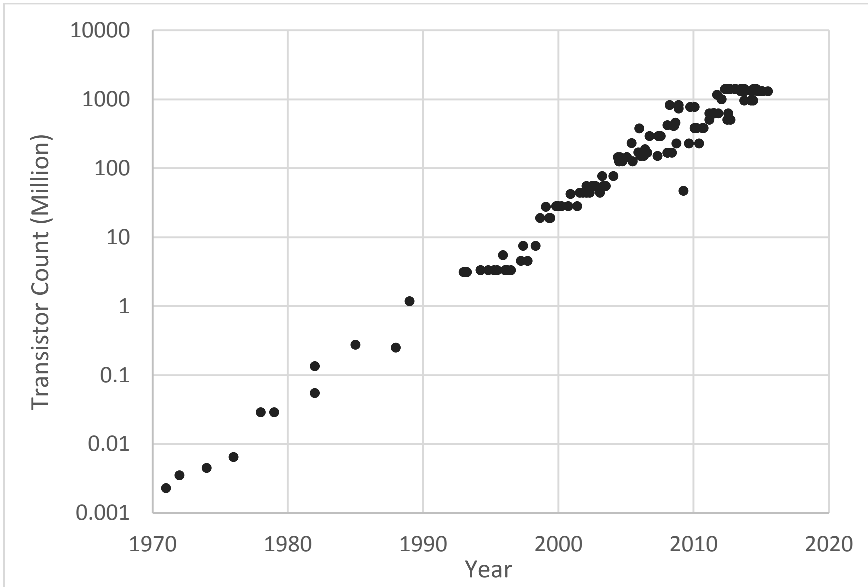


Figure 1. Number of transistor in Intel CPUs[1].

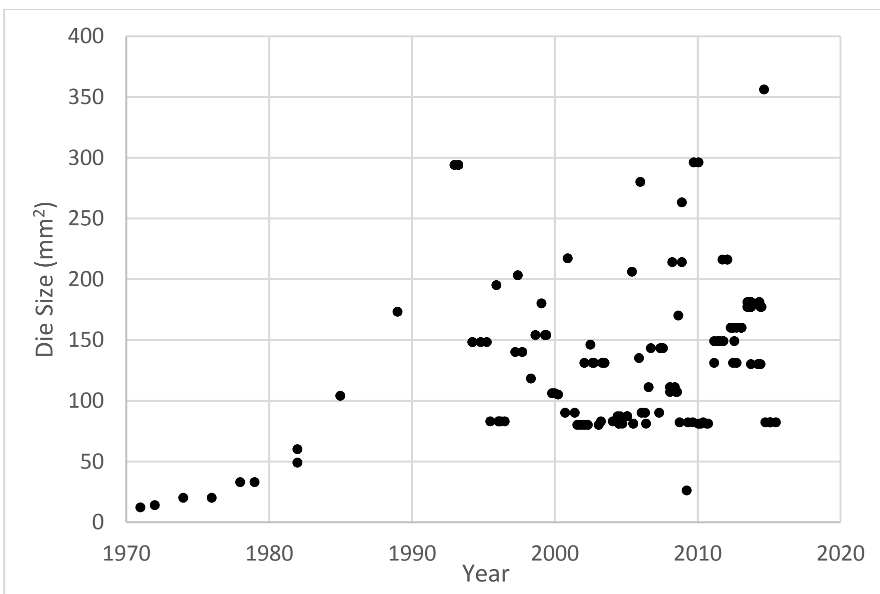


Figure 2. Die size in Intel CPUs[1].

[1] The data for this graph is extracted from http://www.techarp.com/

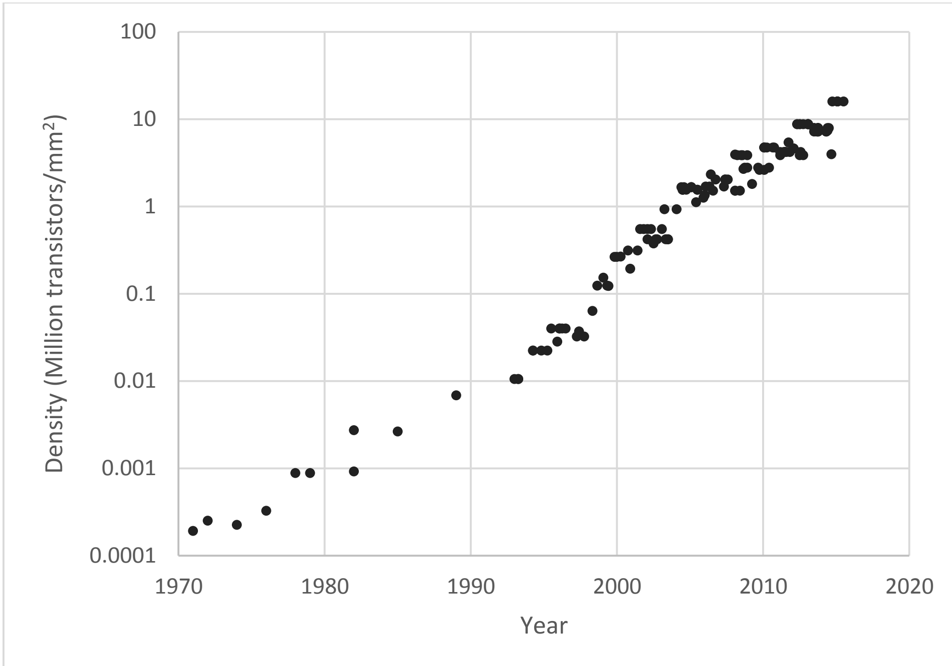


Figure 3. Density of transistors in Intel CPUs[2].

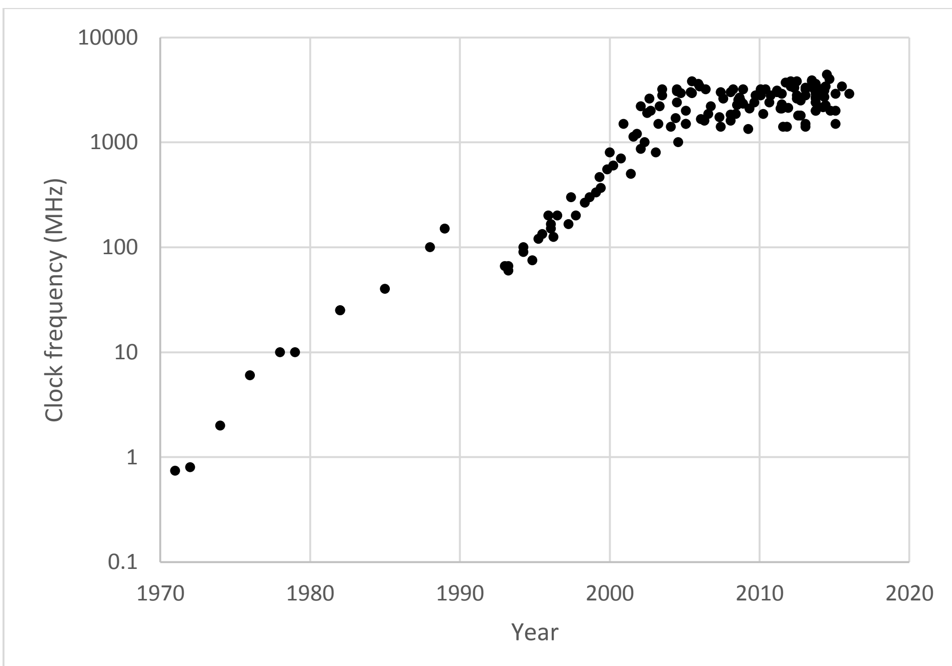


Figure 4. The maximum clock frequency in Intel CPUs[2].

[2] The data for this graph is extracted from http://www.techarp.com/

1.2. 3D IC

Integrated circuits are made by transistors arranged on a silicon die. These integrated circuits are known as 2D IC, since the transistors are located on the surface of the die. In 3D ICs, these 2D dies are stacked on top of each other, hence adding the vertical dimension to the IC, and turning the traditional 2D IC to 3D IC.

In a 3D IC, two or more silicon dies are stacked, and through silicon vias (TSV) are used to transfer signal between different dies. 3D ICs offer multiple advantages to the designers compared to traditional 2D ICs, that are discussed below.

1.2.1. Less Chips on Board

Having less chips on a board is a desirable attribute for electronic system. Reducing the number of chips makes the board level design easier. Also, eliminating board level connections and moving them to on-chip and package level can potentially increase the speed of the circuit, and decrease the power consumption, since package-level connections are generally shorter than board-level connections; shorter connections means less parasitic capacitive load, and less capacitive load mean shorter delay and less power wasted in the wire. With 3D ICs, it is possible to have more complex system on package (SoP). In a traditional SoP, designers put different dies in one package, and wire bonds are used to connect these different dies together, but in a 3D IC, two or more dies are stacked on each other, and they are connected to each other with TSVs, therefore eliminating the need for long wire bonds. Using TSVs instead of wire bonds can further reduce the parasitic effects of the interconnects.

1.2.2. Shorter On-Chip Interconnects

It is generally preferable to use shorter interconnects in chips. A shorter interconnect means lower parasitic resistance and capacitance. Lower resistance and capacitance means lower delay along the path of metal interconnect, and lower power consumption. Imagine a square shaped chip, with die area of A. In this case, each side is going to be. The longest imaginable path for metal interconnects is from one corner to opposite corner, along the sides. The length of this path is going to be $2\sqrt{A}$. Now imagine if this chip is designed in two separate dies, square shaped, and with the area of *A/2*. In this case the length of the longest path is $2\sqrt{A}/\sqrt{2}$. If the same chip is divided into 4 dies, the area of each die is going to be *A/4*, and the longest path is going to be as long as $\sqrt{A}$. We can see that breaking a big die into smaller dies, and stacking them vertically can potentially decrease the length of interconnects. Figure 5 shows how the interconnect length reduces in 3D ICs. Note that the dies in these diagrams are not to scale, and the length of vertical interconnects are negligible compared to the dimensions of the dies.

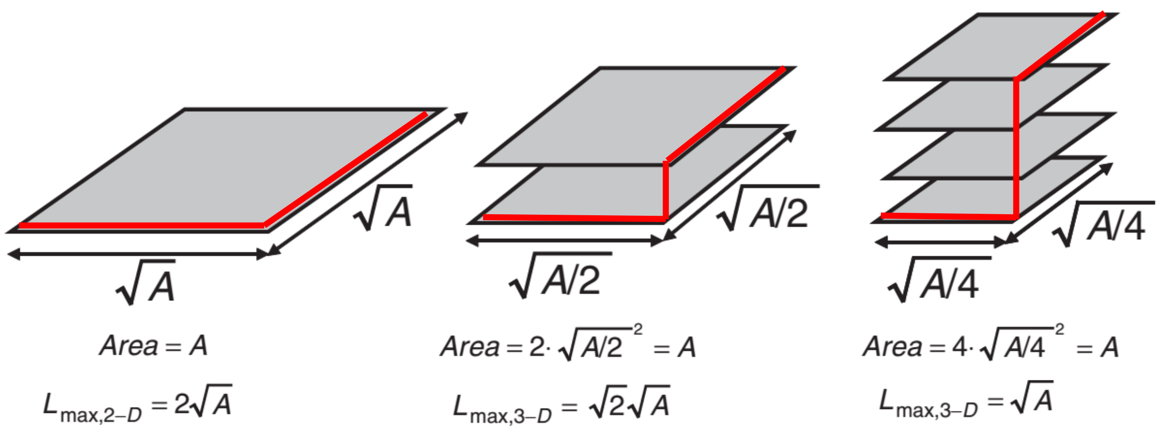


Figure 5. Reduction in interconnect length in 3D IC with multiple dies[3].

1.2.3. Heterogonous System Design

In most cases, it is not feasible to use different technologies in fabrication of one single die. When a system is designed in separate dies, different technologies and fabrication methods

---

[3] Source: Pavlidis, Vasilis F. and Friedman, Eby G. *Three-dimensional Integrated Circuit Design.* Burlington, MA : Morgan Kaufmann, 2009. 978-0-12-374343-5, page 10.

can be used for different dies. In this case it is feasible to use the fabrication process that yields the best result for each individual die, and optimize each die separately. This added degree of freedom in design process can lead to improvements in overall system performance, reduction in fabrication process complexity, or integrating systems that were previously not possible. An example of a chip including several dies with different applications is shown in Figure 6.

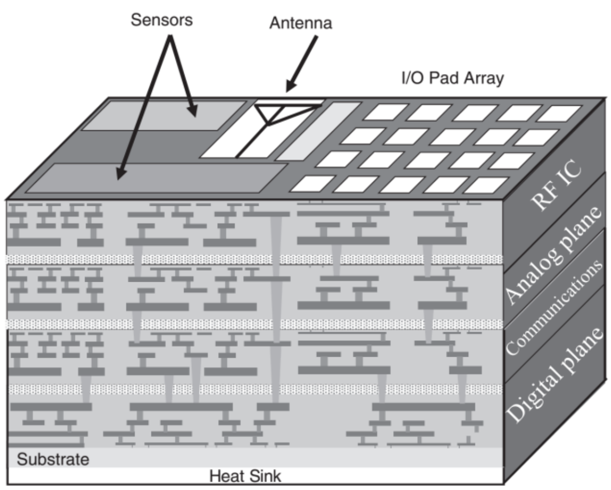


Figure 6. An example of a heterogeneous 3D IC. Notice that different dies have different functionalities[4].

1.2.4. Other Advantages of 3D IC

It is important in many circuits to hide the contents of chip, and protect the design of the circuits from being reverse engineered. One of the common methods is circuit obstruction, which involves adding additional metal interconnects, and transistors, and other obstacles to make it harder for others to identify the circuit. 3D ICs have better circuit security, because having multiple dies in one chip gives more opportunities for the designers to obstruct the circuit more.

[4] Source: Pavlidis, Vasilis F. and Friedman, Eby G. *Three-dimensional Integrated Circuit Design.* Burlington, MA : Morgan Kaufmann, 2009. 978-0-12-374343-5, page 11.

Also in 3D IC more transistors can be placed in one chip, therefore potentially increasing the performance of ICs.

1.3. TSV

Through Silicon Vias are used as connection between dies for carrying electrical signals in a 3D IC. In this section, we are going to discuss different attributes of TSVs.

1.3.1. Shape and Orientation of TSV

As the name suggests, TSVs are vias that pass through the silicon dies. These TSVs are usually shaped like a cylinder, or octagonal/hexagonal prism. The bonding of the dies can be done in two ways: face-to-back, and face-to-face. In face-to-back bonding, the front side (metal interconnects) of bottom die is attached to the back side (silicon substrate) of the top die. Figure 7 shows the different bondings. The wide tall blue metal connections are the TSVs.

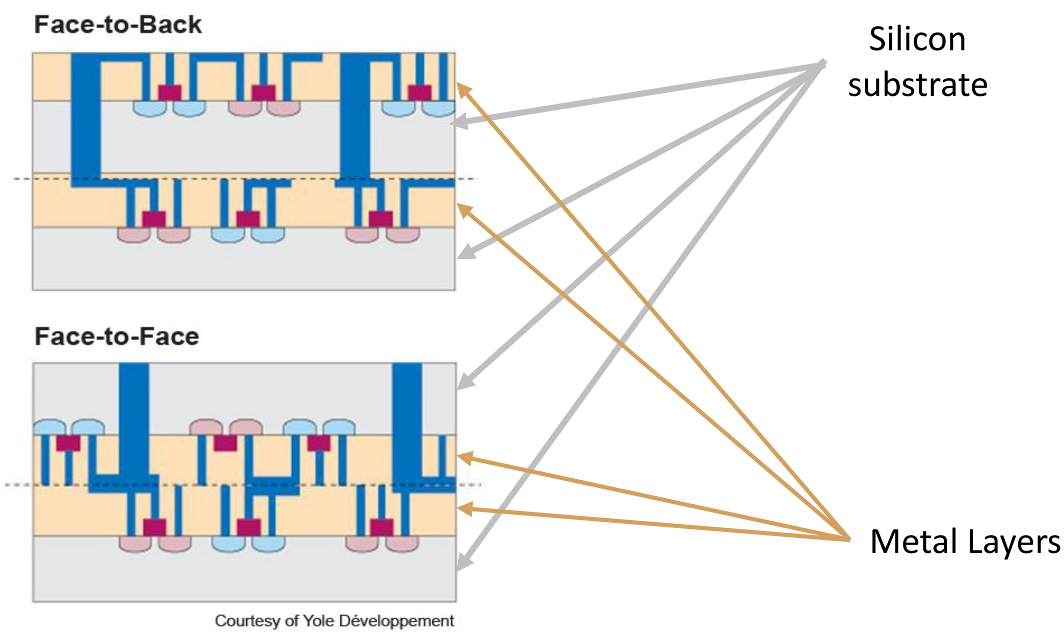


Figure 7. Different bonding schemes for dies in 3D IC.

1.3.2. Materials Used in TSV

Several materials have been considered to be used for TSVs, such as copper, tungsten, polysilicon, and nickel. These conductive materials are used as TSV filling. TSV fabrication process has two main steps: etching, and filling. In etching a hole is made in the silicon substrate, and in the filling process conductive materials are used to fill the hole. Silicon Oxide is used as an insulator between TSV and silicon substrate. This prevents any unintended conductive paths through the bulk of silicon.

1.3.3. Process Flow

The TSV process can be done at different stages in silicon fabrication process. In 'Via-first' process, the TSVs are placed in the die first, then the rest of fabrication process is done. In 'Via-middle' the etching and filling of TSVs is completed after FEOL; the fabrication of transistors; and before BEOL; the placement of metal layers. In 'Via-last' method, TSVs are inserted after BEOL, and before assembly and packaging.

1.3.4. Electrical Characteristics

The TSVs can be modeled with an equivalent circuit consisted of resistors, capacitors, and inductors. The equivalent circuits can help us understand how the signals pass through different dies. The parameters of the models depend on the process flow, and the materials used in TSV. The model used in this work is discussed in section 3.1.

1.4. 3D Memories

Memories are made used for data storage. They are made out of smaller blocks, each block stores a small amount of data, and these multiple copies of these blocks are used in the architecture of the memory. These individual blocks may be consisted of several sub-blocks, and at the end of the hierarchy there are cells. Each cell stores one bit of data. This means that a memory, is a structure that include countless small cells. This makes the circuitry of the memories very homogenous; the same blocks and patterns are repeated over and over.

The homogeneity of memories, enables the designer to break apart the circuit of the memory into smaller identical blocks, and put each individual block in a separate die. This way, the memories can be fabricated as 3D memories. Figure 8 shows the cross section of an 3D DRAM. There are eight dies stacked on each other. Notice the uniformity of the dies, and how all the TSVs are aligned with each other.

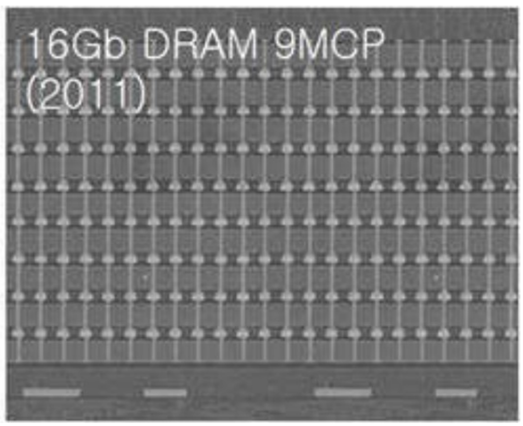


Figure 8. SK Hynix 16 Gb DRAM[5].

1.4.1. Commercial 3D Memories

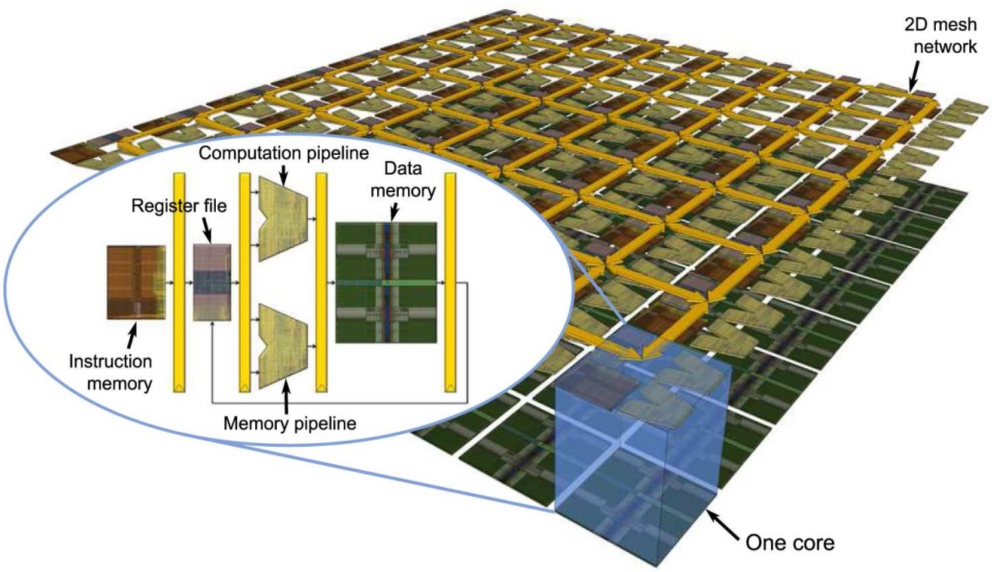
Because of the relative simplicity of designing 3D memories compared to other 3D systems, many companies have started using 3D integration in their memories. Samsung started to manufacture and fabricate 3D flash memories in 2013[6], and they released 3D DRAM for DDR4 memories in 2014[7].

---

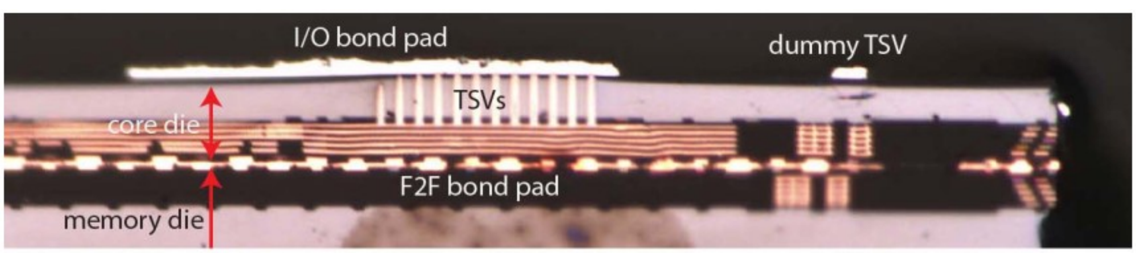
[5] Source: "Technology Roadmap of DRAM for Three Major manufacturers: Samsung, SK-Hynix and Micron" Technical Report, May 2013
[6] http://www.samsung.com/semiconductor/about-us/news/12990
[7] http://www.samsung.com/semiconductor/about-us/news/13602

1.4.2. 3D SRAM

SRAMs are integrated in many of the digital chips, such as processors, graphic processors, microcontrollers, and other system on chips. SRAMs can be integrated as a part of the circuit in a 3D IC. For example, in [1], a 64-core processor with on-chip SRAM is designed. The processor cores and memory blocks are on two separate dies, and 3D integration techniques were used to design this 3D IC with on-chip SRAM. Figure 9 shows the diagram of the multicore processor in [1].



(a)



(b)

Figure 9. 3D processor with 64 cores and 64 memory blocks. (a) Internal architecture, (b) Chip cross section.

1.5. Design Rules for TSV

Part of the process of designing a traditional 2D VLSI circuit is following the design rules. These design rules can be divided into two groups: first group makes sure the circuit can be fabricated, and second group of design rules are used to increase the yield of the fabricated circuits. The first group of design rules are developed based on the limitations of lithography, and other steps in the fabrication process. The second group are mostly based on electrical characteristics of semiconductor-based devices. These design rules apply to all aspects of circuit, such as transistors, vias, metal layers, their sizes, and their placements.

TSVs are no exception to the rules and when a 3D IC is designed, the design rules have to be followed. For example, there are rules for minimum distance between two TSVs, and the keep-out zone around TSV where metal layers should not be used.

One of these rules is the minimum density of TSVs; there has to be a minimum number of TSVs per area in the chip [2, 3]. In traditional 2D chip design, in order to satisfy the minimum density requirements for the metal layers, designers with the help of CAD tools, add 'fillers' to the chips. These fillers are pieces of metal, that are not connected to the circuit and do not have any electrical functionality. Similarly, in 3D ICs, in most chips, designers have to insert TSVs that that are not used in the circuit and are used only to satisfy DRC. These TSVs are called 'Dummy TSVs'.

The dummy TSVs are required for the thinning process. In thinning, the thickness of the die is reduced to match the length of TSV, and these TSVs help with the stability and reliability when the mechanical grinding and polishing is done for thinning [4]. Figure 10 illustrates different stages of the fabrication, including thinning. The thinning is a crucial and very sensitive step in the process of fabrication. For example, the thickness of a wafer with a diameter of 300 mm, is reduced from several hundred microns to 50 microns, and the variation in the thickness is less than one micron across the whole wafer. This level of precision makes the thinning process extremely difficult, and therefore it is necessary to follow the minimum density rules for TSVs to make thinning possible.

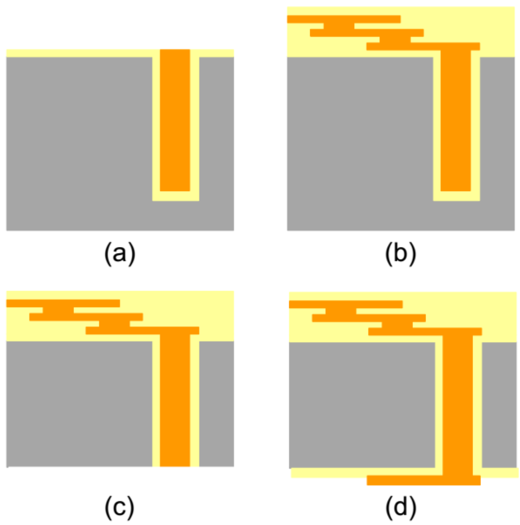


Figure 10. Different steps in the thinning process. (a) Before BEOL, (b) After BEOL, (c) After thinning the substrate, (d) after adding the back metal layer[8].

In many of the chips, the number of dummy TSVs is very large. For example, in 3D processor designed for [1], there are about 50,000 TSVs, and 3000 of them are dummy TSVs.

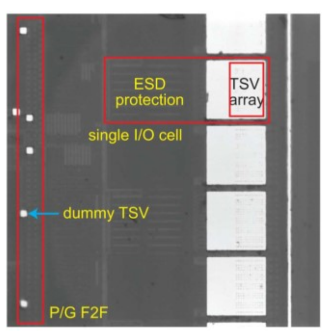


Figure 11. Dummy TSVs in 3D processor designed in [1].

[8]Source: "Overview of CMP for TSV Applications." 2016. Available at: http://www.entrepix.com/docs/papers-and-presentations/Rhoades-CMP-for-TSV-AVS-June2013-shareable.pdf. Accessed September 14, 2016.

1.6. Timing in SRAMs

When a large SRAM is designed, the SRAM is divided into multiple blocks, and depending on the size of the SRAM and the blocks, each block can be further divided into smaller arrays and so forth. At the lowest level some SRAM cells are connected to same output. The outputs of arrays are combined with each other, to create the output for the next level. At the last level, the outputs of all blocks generate the output of the whole SRAM. An example of the SRAM circuit with different levels is shown in Figure 12.

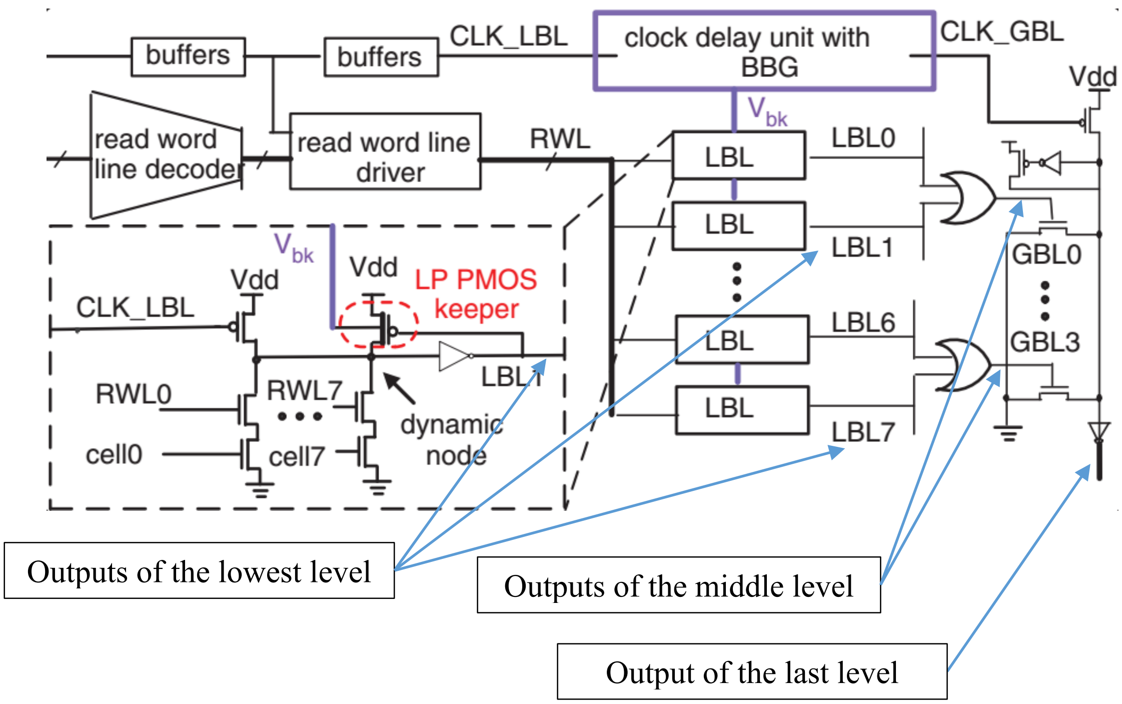


Figure 12. A block diagram of a SRAM with 3 levels[9].

The combining of the outputs of these levels is done by 'Dynamic Circuits.' In a dynamic circuit, the operation of the circuit is controlled by a clock signal. The dynamic circuit has two

---

[9] Source: N. Gong, J. Wang, S. Jiang and R. Sridhar, "Clock-biased local bit line for high performance register files," in *Electronics Letters*, vol. 48, no. 18, pp. 1104-1105, August 30 2012.

phases; the first phase is called 'pre-charge' and the second phase is 'evaluation' phase. The output of these circuits is connected to VDD through a PMOS, and is connected to GND through a 'pull-down network.' During the pre-charge phase, when clock signal is LOW, the output net is charged, and the voltage of this net reaches to VDD. During the evaluation phase, when clock signal is HIGH, depending on the function of the circuit and structure of the pull-down network, and the value of the inputs, the output net may discharge to GND. The circuit in Figure 13 shows the output nets, the PMOS transistors used for pre-charge, and the pull-down networks.

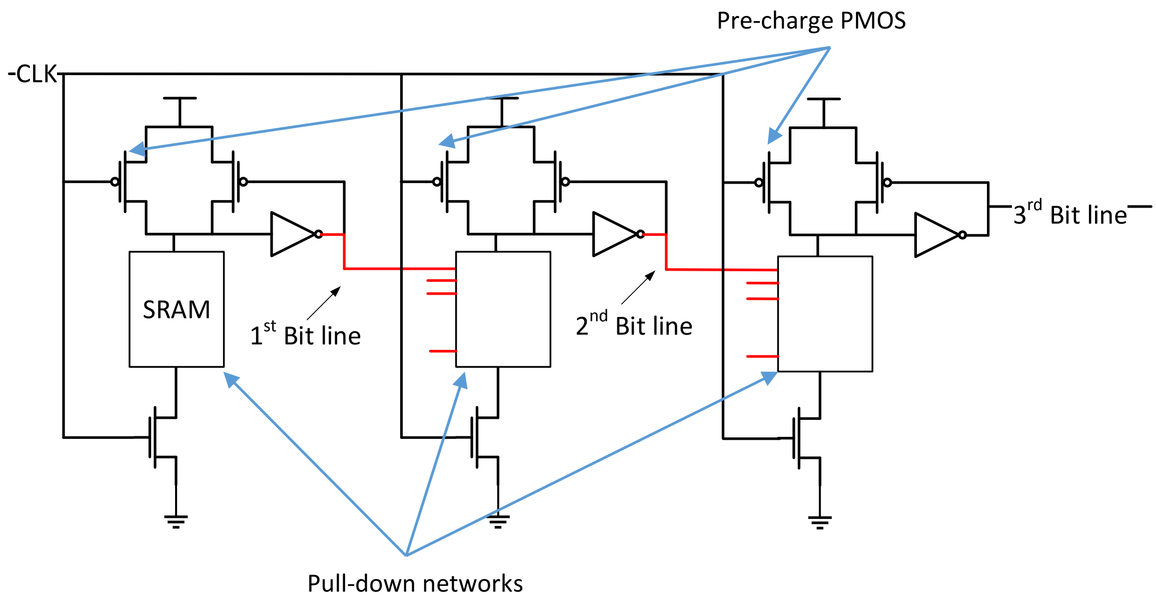


Figure 13. The circuit used for combining the output of different levels in SRAM.

Combining the inputs of previous stages by dynamic circuits requires some time to complete. This time is called the 'delay' of the circuit. A more specific definition of delay is the time between when the moments that evaluation starts and the time when the evaluation is done and the output of the circuit is ready. In Figure 13, the circuit that generate the output of the final level has the longest delay, because the inputs of this circuit have to wait until the outputs of all the previous circuits are ready. Therefore, the evaluation time of this final stage circuit is the delay of previous stage with the addition to the time needed for the circuits itself to generate its own output. The clock period has to be long enough that the evaluation time needed for the last stage covers the time needed for the last stage circuit. The output of the circuit in Figure 13 is shown in Figure 14.

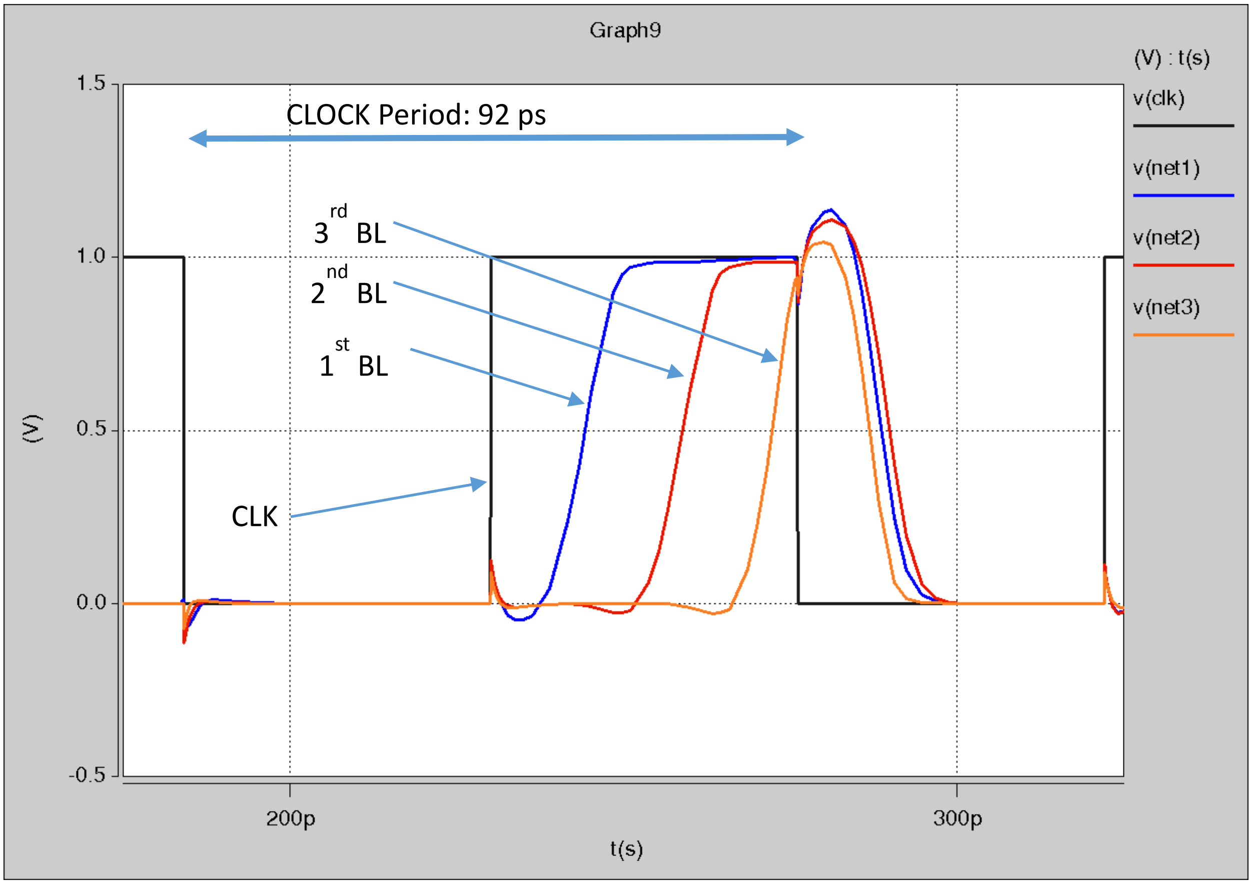


Figure 14. The output of different levels in the circuit shown in Figure 13.

The output of the first level is ready shortly after the evaluation phase starts, and then the outputs of second and third levels follow. The shortest period for the clock that allows all parts of circuits to function properly is 92 ps.

If we provide different clock signals to different levels, the performance of the circuit can be improved, the period of the clock can be reduced, and a higher clock frequency can be used. Adding a delay between the clock signals that goes to circuits in different stages helps reduce the clock period if the delay between clock signal matches the delay of circuits of the individual stages. The circuit below in Figure 15 shows the location of delay cells in the circuit. As it can be seen in Figure 16, the period of clock is reduced to 42 ps from 92 ps, and the frequency of the clock can be increased by 119%.

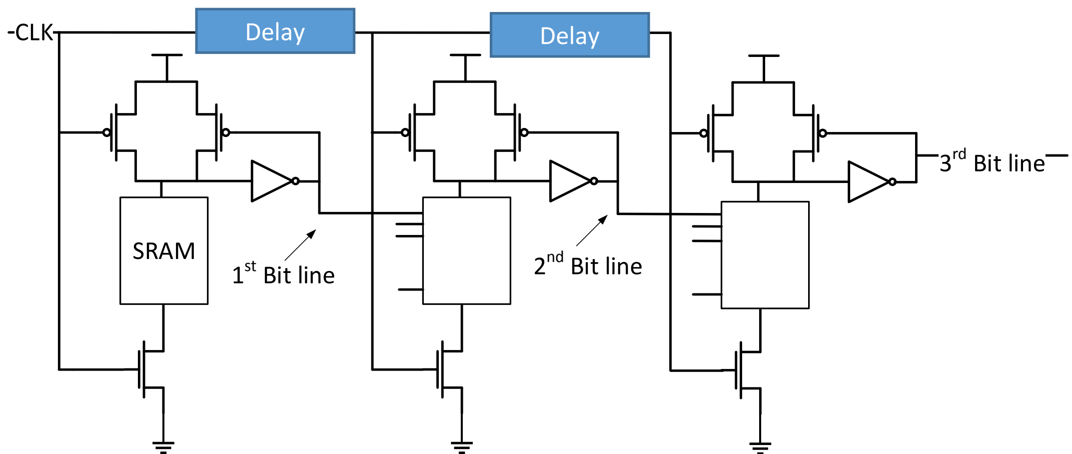


Figure 15. The dynamic circuits with added delay cells.

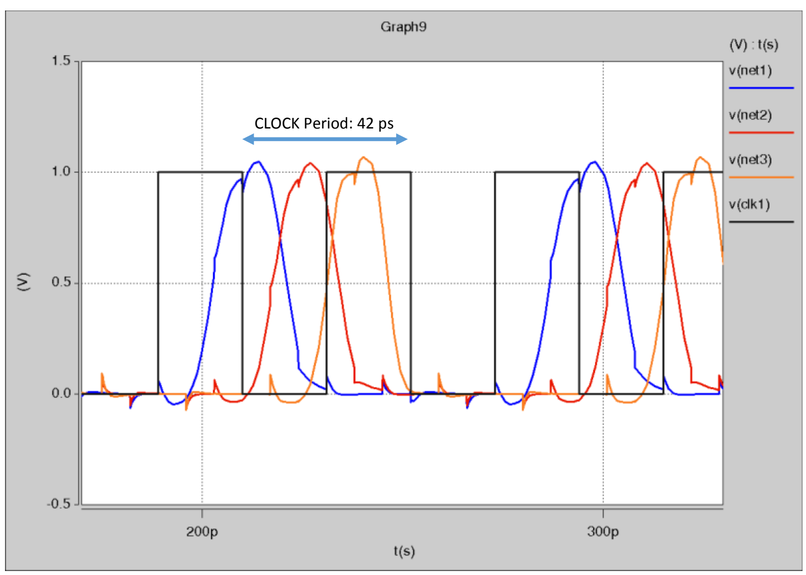


Figure 16. The output of different levels in Figure 15 circuit.

Using delay cells in SRAM arrays is a very common technique [5], but the usage of delay cells is not limited to SRAM arrays. Delay-locked loops also benefit from delay cells [6], phaselocked loops also use delay cells [7], and they are included in voltage controlled oscillators too [8].

1.7. Our Proposed Idea

Currently, designers use delay cells in all SRAMs to improve the performance of the memories. 3D SRAMs are not an exception either. As it was mentioned in Section 1.5, there are numerous unused TSVs in 3D ICs due to the design rules. These TSVs similar to all other wires and connections used in circuits, cause some delay in the transmission of the signals.

We propose a new method that uses the previously unused dummy TSVs to our benefit, and we design 'TSV-Based Delay Cells' (TBDC), that can be used in 3D SRAMs to improve their performance.

1.8. Previous Works

The existing research in this subject can be divided in three main groups.

1.8.1. Delay Models for TSVs

Numerous papers have been published about modeling the delay of TSVs for different conditions and different needs. In [9], the behavior of TSV in high frequencies is analyzed, and [10] offers a model similar to high speed RF transmission lines for TSVs. For lower frequencies, using a transmission line model is not necessary and [11] offers a lumped circuit model for TSV delay. Papers such as [12] and [13] study the effects of interference between multiple TSVs and provide a circuit to model the cross-interference. Reference [14] examines the relation between capacitance and the voltage carried by the TSV and how this capacitance changes the characteristics of TSV. Papers like [15] provide a closed form equation for finding the delay of TSV with as many as 13 variables. These variables are parameters such as the height of TSV, diameter of TSV, thickness of oxide used as insulator around TSV, TSV-to-TSV pitch, and number of stacked chips. In this work we use a RC model based on [16] and [17]. These two papers introduce a circuit with a resistor and a capacitor, which can accurately model the behavior and response of TSV in circuits such as SRAM arrays. Figure 17 shows the equivalent circuit in the RC based model for TSV.

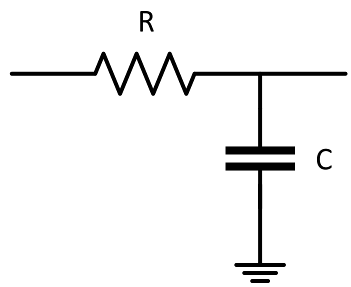


Figure 17. The equivalent circuit of a TSV.

1.8.2. TSVs in Clock Tree

When TSVs are used for transferring clock signal across multiple dies, it is very important to see how these TSVs can influence the clock tree. TSVs can influence the delay on the clock in the clock tree, or increase the power of the clock tree. In the 3D IC designed in [18], there is a 2D clock tree on each individual die and TSVs are used to connect these clock trees across different dies. This paper studies the method to increase the robustness of the 3D clock tree by adding multiple TSVs between dies that carry the clock signal to add redundancy to the circuit. Adding redundant TSVs ensures us that the clock tree will still be functional, even if some of the TSVs are faulty. In [19] the effect of number of TSVs used in the chip on the power and delay in clock tree is discussed, and methods for finding the optimum number of TSVs in a 3D IC are suggested.

1.8.3. Timing Assistance with Dummy TSVs

The papers mentioned in 1.8.1 and 1.8.2 do not discuss the effects of dummy TSVs. Our investigation in the existing literature shows that nobody has used the dummy TSVs in the electrical circuit of the 3D IC to improve the performance of the chip, but there are papers that use the mechanical and thermal properties of dummy TSVs to improve the stability of the clock tree. In [19] and [20], the authors propose a method to take advantage of the thermal and mechanical coupling of dummy TSVs, to reduce the thermal and mechanical stress on the 3D chip and improve the stability of clock in the clock tree in 3D IC. In these papers, the TSVs are not part of the electrical circuit, and are not in the structure of clock tree. They are only used as mechanical objects.

1.9. Our Contribution

Based on the extensive study that was done on the literature, it was concluded that nobody has ever tried to use TSVs as delay cells. In this work, we are going to implement this novel idea, and verify the effectiveness of this technique. Using dummy TSVs for delay cells has three major advantages over using traditional delay cells.

- Utilizing wasteful dummy TSVs: We are using the TSVs that are part of the chip, and our proposed design does not require inserting additional TSVs.
- Saving silicon area on the chip: Traditional delay cells use transistors in their circuitry. This means that more silicon area is needed for this kind of delay cells, because in our TSV-based delay cells, since the delay is caused by the TSVs, the added silicon area is much smaller than the traditional delay cells.

- More robustness against variations in circuit: Active circuits and transistors are very sensitive to changes in the circuit. Fluctuation in the voltage, increase in temperature, and supply noise can have negative effect on the performance of the active circuits and traditional delay cells. In TSV-based delay cell, since the behavior of the delay cell is similar to a passive RC circuit, the delay caused by the circuit is not influenced by the parameters such as supply noise or temperature.

The structure of 3D SRAM is discussed in chapter 2, chapter 3 includes the designs of TBDCs, and results of performance improvement of 3D SRAM with our proposed delay cells, and comparing them with traditional delay cells, and chapter 4 contains the conclusion and summary.

2. DESIGN USED FOR THE 3D SRAM ARRAY

One of the most attractive areas in 3D IC is designing 3D SRAM. SRAMs are widely used in many digital systems. SRAMs are consisted of arrays of SRAM cells, and for the same reason, it is easier to break the circuit into smaller blocks, and place different blocks on different dies and form a 3D SRAM. Also, SRAMs are used as cache memory in CPUs. In [21] and [22] multiple dies are used to design multi-core processors with cache. The homogeneity of the processor's cores and the cache memory blocks makes it very easy to separate different cores and parts, and design the processors in different dies as a 3D IC.

In this work, we are going to use the TBDC in 3D SRAMs. The SRAM is made out of multiple SRAM sub-arrays, and it is based on 8T SRAM cell design. We designed the memory in 4 different technology nodes: 180 nm CMOS based on IBM 7RF library, 45 nm CMOS based on NCSU FreePDK library, 16 nm high-K CMOS based on PTM model, and 7 nm CMOS based on PTM model. Testing the memory in these technology nodes helps us have a better understanding of the effectiveness of TBDCs in 3D SRAM.

We have designed three memories with different sizes and depth. Usage of different technology node and different sizes is for adding more generality to this research project, and to ensure that the findings of this research can be applied to other circuits and memories designed in other technology nodes or libraries.

2.1. Devices Used in Our 3D SRAM

The SRAMs designed for this work are designed with three different types of devices. In 180 nm and 45 nm SRAMs are designed with MOSFET devices. In smaller devices, the length of channel (L) and the thickness of gate oxide (t_{ox}) get smaller. Having a small t_{ox} deep sub-micron technology nodes drastically increase the leakage current. This is due to channeling effect in gate of MOSFET. In order to solve this problem multiple solutions have been used. One of these solutions is to use a different material with higher dielectric constant for the gate insulators instead of the traditional silicon dioxide. This is the approach used in High-K MOSFET devices that have a thicker gate insulator that reduces the channeling effect. The other solution is using a different geometrical configuration in the device. In FinFETs, the shape of gate is a thin fin which is different from the shape of gates in MOSFETs.

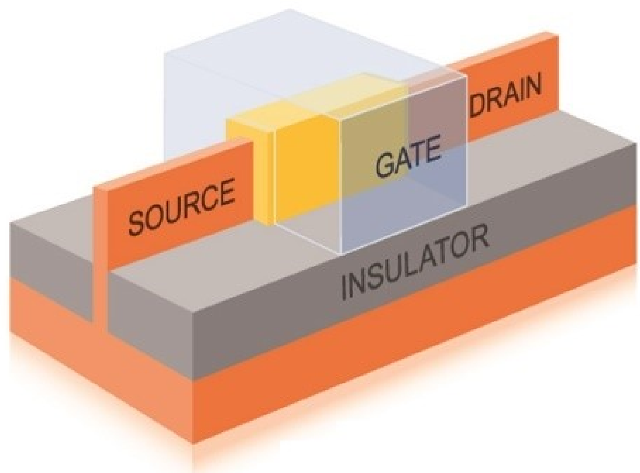


Figure 18. Diagram of a FinFET (Source: GLOBALFOUNDRIES).

2.2. 8T SRAM Cell

The circuit of an 8T SRAM cell is shown in Figure 19. There are several advantages in using 8T SRAM compared to more common 6T SRAM, the most important advantage comes from separating the write and read mechanisms, which has a significant influence in increasing the Read SNM (Signal-to-Noise Margin). However, these improvements come at a cost: 8T SRAM cell has 2 NMOS more than 6T SRAM cell, therefore the area of an 8T SRAM cell is larger than a 6T SRAM cell.

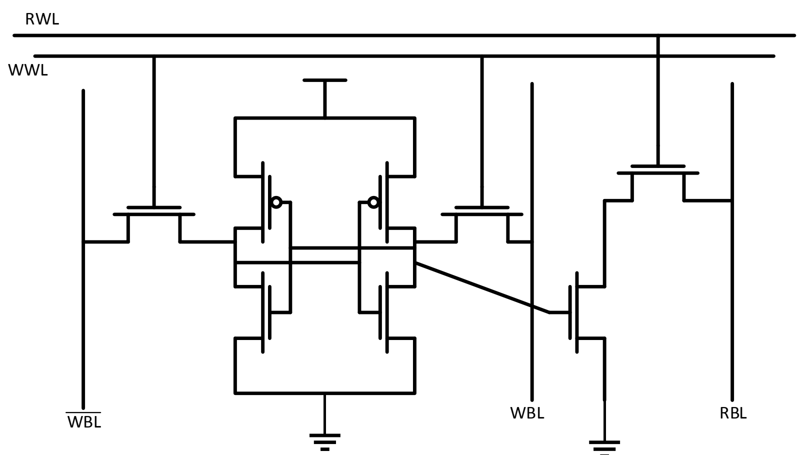


Figure 19. Schematic of 8T SRAM.

2.3. Structure of the 3D SRAM

The three memory architectures that were used are going to be called Mem.A, Mem.B, and Mem.C. The difference in this memories is number of bits in the memory, and number of address bits. The parameters for these memories are listed in Table 1.

Table 1. Parameters of three different memory architectures.

Parameters	**Mem.A**	**Mem.B**	**Mem.C**
Number of Address bits	15	12	9
Memory Size (kB)	128	16	2
Number of memory sub-arrays	32	16	8
Number of columns in each sub-array	32	16	8
Number of rows in each sub-array	32	16	8

2.3.1. Memory Sub-Array

The memory sub-arrays are consisted of 8T SRAM cells. This cells are arranged in equal number of rows and columns. The cells in the same row, share the Write Word Line (WWL) and Read Word Line (RWL) signals, while the cells in the column share Write Bit Line (WBL), inverted of Write Bit Line ($\overline{WBL}$), and Read Bit Line (RBL). The Word Lines are generated by address decoders. Figure 20 shows the SRAM cells and their connections inside an sub-array. The signals that are related to write operation (WBL, $\overline{WBL}$, WWL) are not shown in this figure.

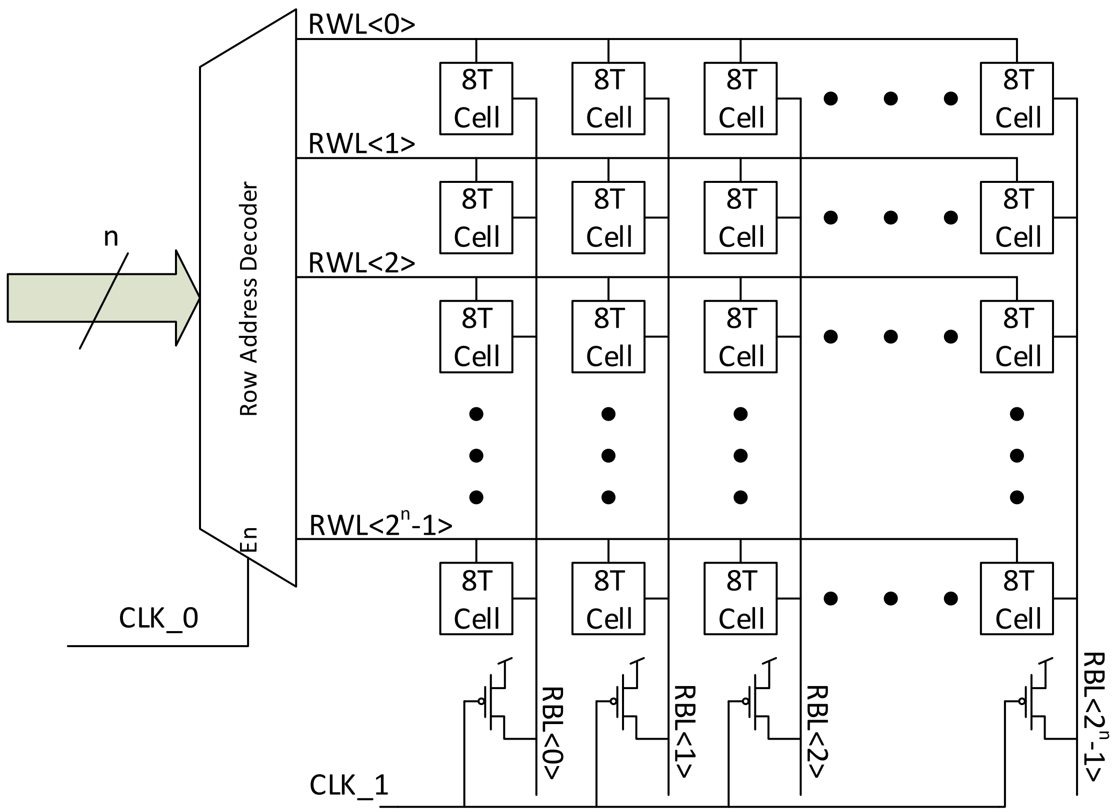


Figure 20. The SRAM cells inside a sub-array.

CLK_0 signal in Figure 20 is used to enable or disable the row decoder, and CLK_1 signal is used to enable or disable the PMOS transistors used for pre-charging the bitlines.

2.3.2. Dynamic Gate

Whenever a read operation is happening, only one of the Read Bit Lines is chosen, and this bit line is the output of sub-array. This output is determined by the value of one of the SRAM cell that was read during the read operation. To choose this bit line, a dynamic gate is used. The dynamic gate has two sets of inputs: the Read Bit Lines from the sub-array, and selector signals from the column address decoder. The input of the decoder determines which RBL is chosen. The SRAM cell which is read is chosen by the address input of the row address decoder, and column address decoder. The circuitry of the dynamic gate and connections are shown in Figure 21.

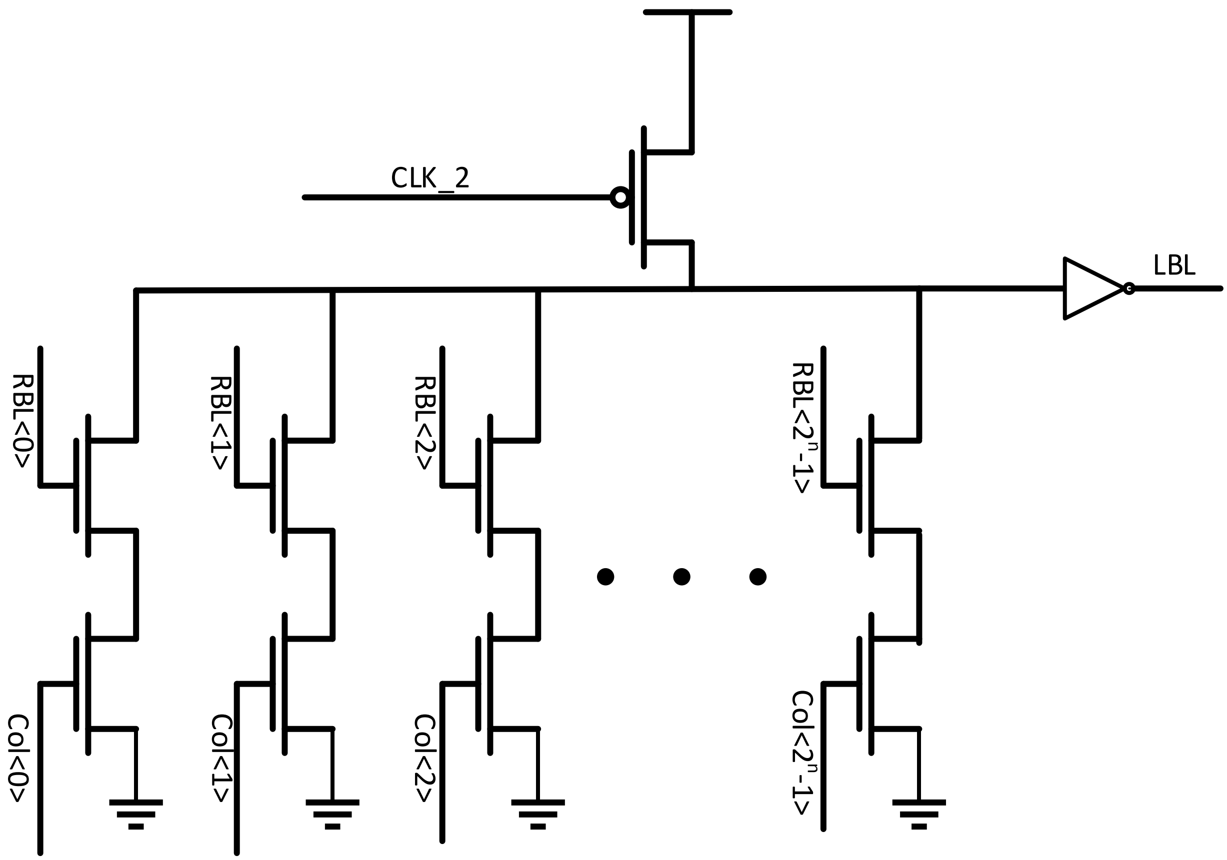


Figure 21. Structure of the dynamic gate.

CLK_2 signal controls the pre-charge and evaluation phases in the dynamic gate. The signals named Col<i> are the outputs of column decoder. Local Bit Line (LBL) is the output of the column decoder.

2.3.3. SRAM Array

In the top level, the output of the sub-arrays (LBLs) are connected to a dynamic gate, similar to the dynamic gate inside sub-array, and this dynamic gate gets its inputs from LBLs, and a sub-array decoder. The output of the sub-array decoder determines which LBL is chosen and sent to output. The pre-charge and evaluation in this dynamic gate is controlled by CLK_3.

The output of this dynamic gate is called Global Bit Line (GBL). The value of GBL is determined by the address bits, and the value stored in SRAM cells. If the *x*th output of row decoder, *y*th output of colomn decoder, and the *z*th output of the sub-array decoder are activated, then the value of GBL is determined by the value of SRAM cell in *z*th sub-array, which is connected to RBL<*y*> and RWL<*x*>. This means that the GBL reflects the value of only one bit of the memory. In practice, memories have 8,16,32,64 or more bits in the output. In order to have *m* bits in the output in our memory, we need to put *m* memory arrays in parallel.

2.4. Using Delay Cells in SRAM

The operation of the SRAM Array can be broken into few stages. For simplicity, we assume that all the signals that need to pre-charged are ready.

- The Row Decoder: The decoder has to generate RWL signals first.
- RBL: When the output of row decoder is ready, one row is chosen, and then the RBLs start to evaluate. At the end, some of these RBLs get dis-charged, and some maintain the pre-charged state.
- Sub-Array Dynamic Gate: When RBL signals are ready, the dynamic gate can generate the LBL signal.
- Array Dynamic Gate: The last stage is when the array dynamic gate generates the GBL signal based on the LBL signals.

All these stages are controlled with a separate clock signal.

- Row decoder is controlled with CLK_0
- The RBL signals are controlled by CLK_1
- The LBL signals and sub-array dynamic gates are controlled by CLK_2
- The GBL signal and array dynamic gate are controlled by CLK_3

If we use only one clock signal, then the period of the clock signal has to be long enough for the operation of all the stages of the circuit to be completed. The second stage cannot start its operation until the operation of first stage is finished, and the third stage has to wait for the output of the second stage and so forth.

Adding delay between different clock sources can speed up the process significantly. If the amount of delay needed between different clock signals is chosen correctly, then the whole circuit is as fast as the slowest stage. This technique is similar to pipelining technique used in computer architecture. We are going to use TBDC to add the desirable amount of delay between different stages to enhance the performance of the SRAM array.

2.5. Layout Configuration

The SRAM memory is made out of multiple instances of SRAM array, depending on the memory width, or number of data bits in memory output. Each array itself is divided into several sub-arrays. This structure of SRAM allows us to design an efficient 3D SRAM, since it is easy to break the circuit into smaller blocks, and put different blocks on different dies.

In SRAM array and sub-arrays, most of the layout area is occupied by the SRAM cells, meanwhile the area taken by the dynamic gates is pretty small, therefore if we are putting some SRAM cells on one die, and the circuitry for the dynamic gates on another die, we can use the opposite arrangement for another array or sub-array. This way, the area used in different dies by the 3D SRAM circuit is balanced.

The other thing that has to be considered when designing the layout is number of TSVs used in TBDCs. Number of TSVs and number of dies in 3D SRAM is going to limit the choices that we have for putting different parts of the circuit in different dies. TSVs exist between two different dies, and each TSV transfers the clock signal from one die to another, therefore it is necessary to use the number of TSVs that gives you the clock on the die that you desire.

3. RESULTS AND ANALYSIS

The 3D SRAM which was described in Section 2 is put to the test, to see how effective the TBDCs are. In order to find the results, the three memories mentioned in Section 2 (Mem.A, Mem.B, Mem.C) are designed in four different technology nodes (180 nm, 45nm, 16 nm, and 7 nm). The operation of these 12 different 3D SRAMs are tested in HSPICE software in two different conditions: with and without TBDCs, and the obtained results are compared with each other.

3.1. TSV Model Used in Delay Cells

In this work we use a simple RC model for the TSV. The model is based on [16, 17], which uses TSVs with the diameter of 10µm, oxide thickness of 1µm, and height of 75µm. The equivalent circuit is shown in Figure 22. The filling of TSV and the silicon substrate behave like a capacitor, and the silicon oxide acts like the dielectric insulator in this capacitor. The resistance is the result of finite conductance of TSV filling. For our TSVs, the value of C is 242 fF, and the value of R is 2 mΩ.

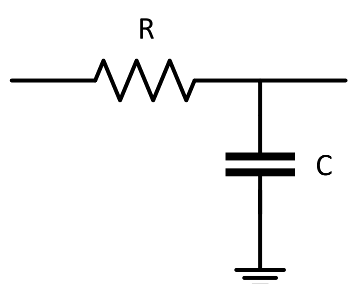


Figure 22. The equivalent circuit of a TSV.

3.2. Delay Needed by Stages

The first step for designing TBDC for the 3D SRAM is to determine how much delay is needed for each delay cell. We define three parameters: D_0 is the delay between CLK_0 and CLK_1 and it is determined by the propagation delay of row decoder. D_1 is the delay between

CLK_1 and CLK_2 and its value is equal to the amount of time needed for RBLs to discharge. D_2 is the delay between CLK_2 and CLK_3 and reflects the amount of time the sub array dynamic gate need for generating LBLs.

The 12 different memories are simulated in HSPICE, and the value D_0, D_1, and D_2 is found for each memory. These values are shown in Table 2.

Table 2. The delay time needed by different stages in SRAM.

Technology Node	Mem.A			Mem.B			Mem.C		
	D_0	D_1	D_2	D_0	D_1	D_2	D_0	D_1	D_2
180 nm (CMOS)	216 ps	178 ps	165 ps	186 ps	122 ps	101 ps	136 ps	52 ps	67 ps
45 nm (CMOS)	53 ps	80 ps	55 ps	48 ps	55 ps	30 ps	16 ps	40 ps	55 ps
16 nm (High-K CMOS)	34 ps	27 ps	47 ps	30 ps	19 ps	24 ps	11 ps	15 ps	14 ps
7 nm (FinFET)	14 ps	8 ps	7 ps	11 ps	5 ps	10 ps	4 ps	4 ps	6 ps

3.3. TBDC Design

Now that we now how much delay should be generated by the TBDC we can design them. It was mentioned in 3.1 that the resistance of a TSV is 2 mΩ and its capacitance is 242 fF. This resistance does not include the resistance of metal interconnects that carry the signal from and to TSVs. The resistance of this metal interconnect depends on the dimensions of it: length, width and thickness. The thickness is determined by the foundry and the process, and designers cannot change the thickness of the metal layers. The width and length of the metal interconnects are decided by the designers, and their values have to follow the design rules set by foundry. The resistance of this metal layer greatly influences the delay of the TSVs and TBDCs, and by tweaking these values, we can design TBDC with arbitrary delay values. In order to make the design stage easier and less complicated, we assume that all the metal interconnects have a fixed constant length, equal to TSV pitch. We also put boundaries on the width of the metal interconnects, the width has to be larger than the minimum width set by the DRC rules, and smaller than the TSV diameter, 10 μm. With these considerations, we can calculate the maximum and minimum value of the resistance of the metal interconnects between TSVs. In Table 3 R_{Total} is the sum of TSV resistance and metal interconnect resistance.

Table 3. The value R_{Total} in different technology nodes.

Technology Node	Max R_{Total} (Ω)	Min R_{Total} (Ω)
180 nm (CMOS)	3.5	0.084
45 nm (CMOS)	56	1.344
16 nm (High-K CMOS)	448	10.572
7 nm (FinFET)	1792	43

It can be seen that in smaller technology nodes, the value of resistance increases. This is caused by scaling in metal interconnects, whenever the transistors get smaller, the thickness and minimum width of the metal layer also shrink. The resistance of the metal layer is computed by (1).

$$R_{wire} = \rho \frac{Length}{Thickness \times Width} \tag{1}$$

In (1) ρ is the resistivity of the metal. In smaller technology nodes, the resistance can be calculated with (2). In this equation, s is the scaling factor, which is equal to the ratio of transistor size in two different technology node. The length is not changing, because we assumed the length is fixed and equal to the TSV pitch.

$$R_{wire} = \rho \frac{Length}{Thickness \times s \times Width \times s} \tag{2}$$

Now with the value of resistance and capacitance in TSV, and delay values in 3D SRAM, we can design TBDCs.

The TBDCs that are used in the memories are consisted of TSVs and buffers. The configuration of TSVs and buffer are described in following subsections.

3.3.1. 180 nm Technology

For each of the three memory architectures (Mem.A, Mem.B, and Mem.C) three TBDCs are used to generate the delay needed for three different stages. Figure 23 shows the connection between TSVs are buffers.

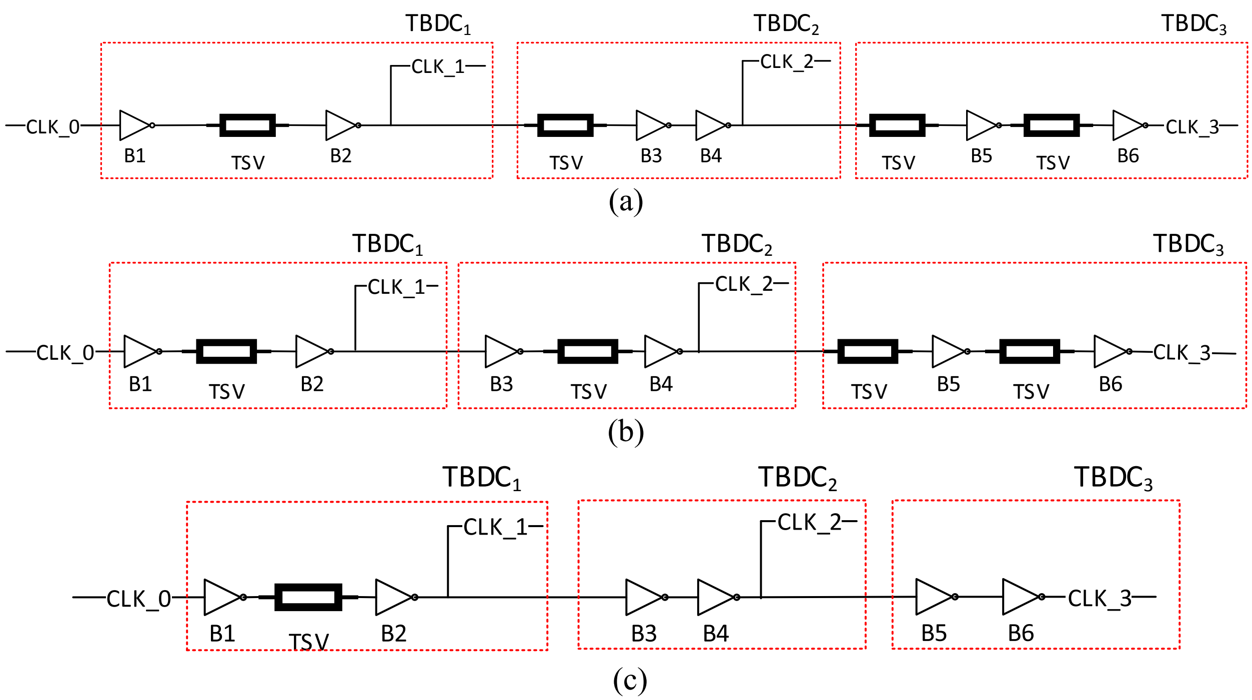


Figure 23. The TBDCs in 180 nm technology (a) Mem.A, (b) Mem.B, (c) Mem.C.

3.3.2. 45 nm Technology

For the 45 nm technology, similar to 180 nm, three TBDCs were used for each memory architecture. Figure 24 shows the connections between TSVs and buffers used in these TBDCs.

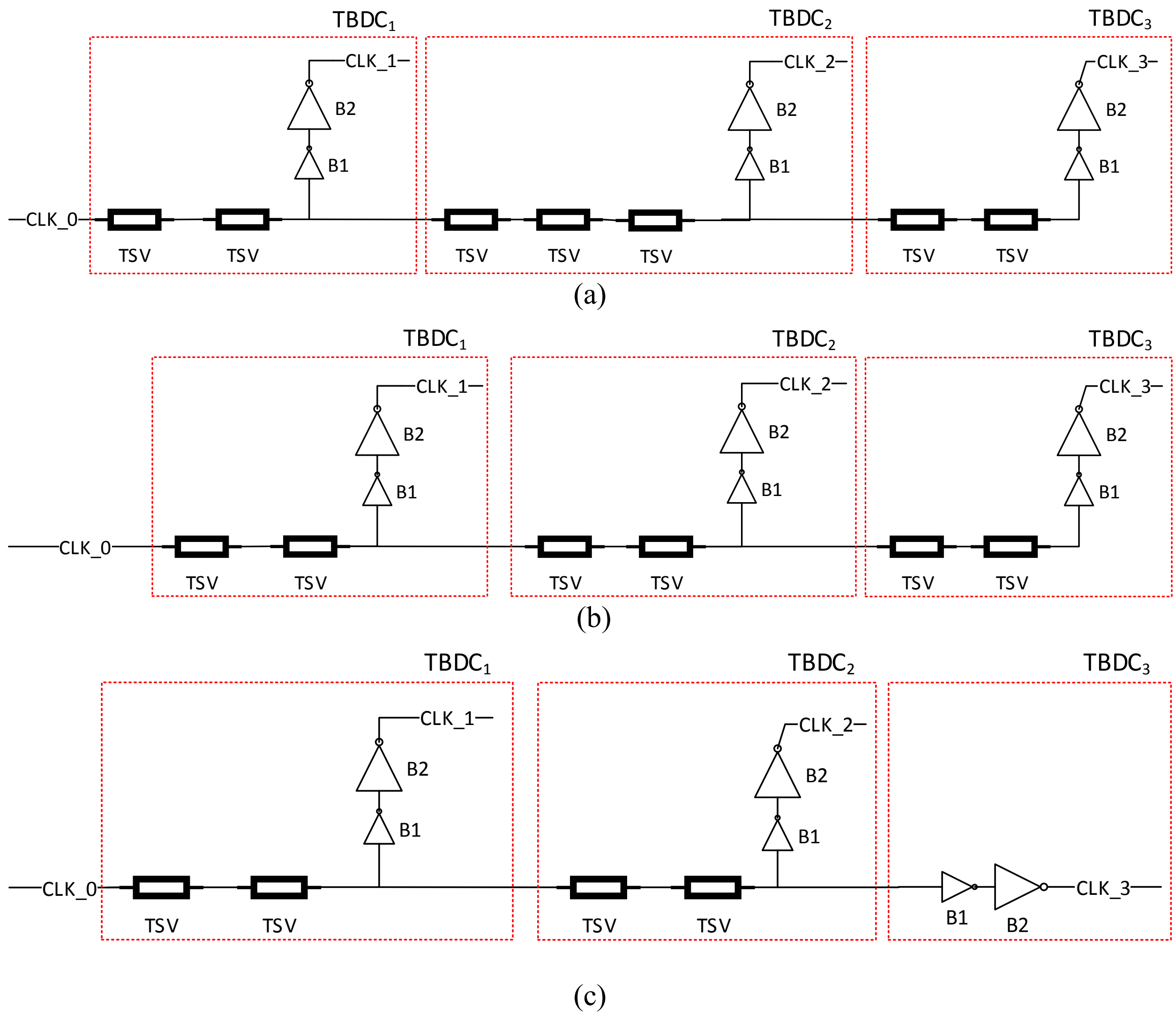


Figure 24. The TBDCs in 45 nm technology. (a) Mem.A, (b) Mem.B, (c) Mem.C.

3.3.3. 16 nm Technology

In this technology, no buffers are used. The reason for this is because the TSVs provide enough delay, and at the same time, the shape of the clock signal is not too distorted, therefore adding buffers to have a cleaner signal is not necessary. The TBDCs used in Mem.A, Mem.B, and Mem.C use the same configuration and connection, and this configuration is shown in Figure 25.

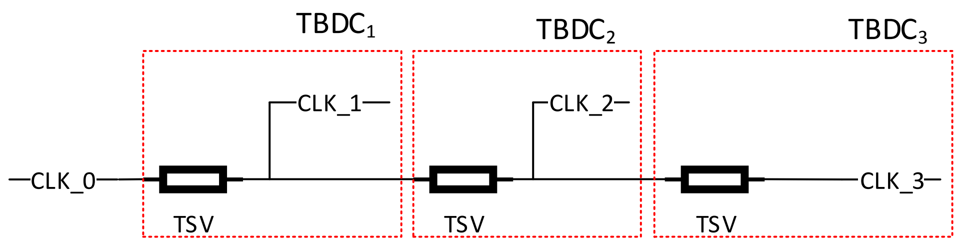


Figure 25. The TBDCs for 16 nm memories.

3.3.4. 7 nm Technology

The value of D_0, D_1, and D_2 in 7 nm memories are so small, that the delay one single TSV is way larger than D_0, D_1, or D_2. There were 6 different imaginable configurations for TSVs and buffers that could have been used. These 6 configurations for TBDC were tested, and at the end it was concluded that the configuration shown in Figure 26 yields the best performance. The same TBDCs were used in Mem.A, Mem.B, and Mem.C. Notice that the CLK_1 and CLK_2 are the same signals.

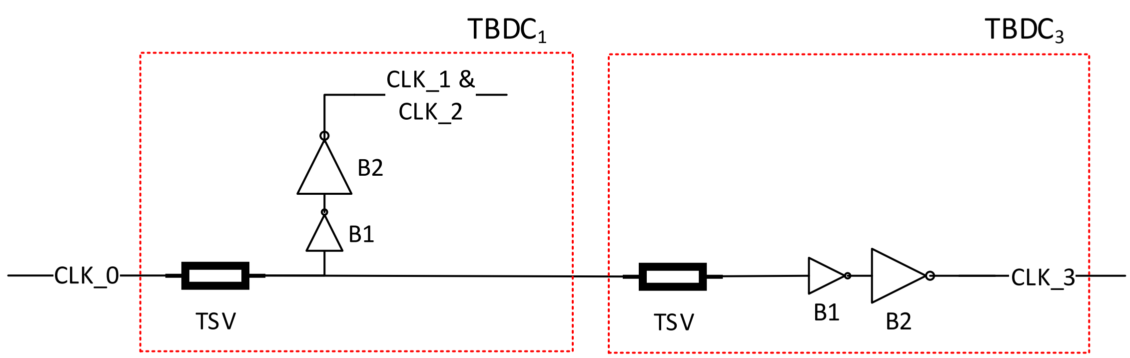


Figure 26. The TBDCs used in 7 nm memories.

3.3.5. Parameters of TBDCs

The TBDC circuits shown can have TSVs with different wire resistance, and the buffers can have transistors with different sizes. There parameters influence the delay of the TBDCs. The wire resistance and the number of the TSVs used in those TBDCs are listed in Table 4.

Table 4. The number and resistance of TSVs in TBDCs.

Technology Node		Mem.A			Mem.B			Mem.C		
		$TBDC_1$	$TBDC_2$	$TBDC_3$	$TBDC_1$	$TBDC_2$	$TBDC_3$	$TBDC_1$	$TBDC_2$	$TBDC_3$
180 nm (CMOS)	Number	2	1	2	1	1	1	1	0	0
	R_{Total} (Ω)	3.5	3.5	3.5	3.5	3.5	3.5	3.5	N/A	N/A
45 nm (CMOS)	Number	2	3	2	2	2	2	2	2	0
	R_{Total} (Ω)	14	25	56	20	34	38	33	35	N/A
16 nm (High-K)	Number	1	1	1	1	1	1	1	1	1
	R_{Total} (Ω)	100	45	140	85	31	78	38	27	50
7 nm (FinFET)	Number	1[a]		1	1[a]		1	1[a]		1
	R_{Total} (Ω)	43[a]		43	43[a]		43	43[a]		43

[a] In these memories, CLK_1 and CLK_2 are connected to each other, and sharing the same TSV.

For buffering the signals, ordinary inverters were used. In these inverters, $W_{PMOS}/W_{NMOS}=2$. The W_{NMOS} and number of buffers used in TBDCs are listed in Table 5. For the 7 nm memories, in addition to W_{NMOS}, number of 'fins' used in NFETs is also listed. In FinFET technologies, the width of the transistor are not arbitrary values. Each gate can have an integer number of 'fins' and the effective width of the transistor is the number of fins multiplied by the effective width of a single fin. In the library used in this work, the effective width for each fin is 42.5 nm.

Table 5. Number of buffers and their W_{NMOS} used in TBDCs.

<table>
<tr><th colspan="2" rowspan="2">Technology Node</th><th colspan="6">Mem.A</th><th colspan="6">Mem.B</th><th colspan="6">Mem.C</th></tr>
<tr><th colspan="2">$TBDC_1$</th><th colspan="2">$TBDC_2$</th><th colspan="2">$TBDC_3$</th><th colspan="2">$TBDC_1$</th><th colspan="2">$TBDC_2$</th><th colspan="2">$TBDC_3$</th><th colspan="2">$TBDC_1$</th><th colspan="2">$TBDC_2$</th><th colspan="2">$TBDC_3$</th></tr>
<tr><td rowspan="2">180 nm (CMOS)</td><td>Number</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td></tr>
<tr><td>W_{NMOS} (μm)</td><td>7.5</td><td>7.5</td><td>1.5</td><td>1.5</td><td>5.5</td><td>5.5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>4</td><td>4</td><td>7.5</td><td>7.5</td><td>1.5</td><td>1.5</td><td>5.5</td><td>5.5</td></tr>
<tr><td rowspan="2">45 nm (CMOS)</td><td>Number</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">2</td></tr>
<tr><td>W_{NMOS} (nm)</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td><td>110</td><td>350</td></tr>
<tr><td rowspan="2">16 nm (High-K)</td><td>Number</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td></tr>
<tr><td>W_{NMOS} (nm)</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td colspan="2">N/A</td></tr>
<tr><td rowspan="3">7 nm (FinFET)</td><td>Number</td><td colspan="4">2[a]</td><td colspan="2">2</td><td colspan="4">2[a]</td><td colspan="2">2</td><td colspan="4">2[a]</td><td colspan="2">2</td></tr>
<tr><td># of fins</td><td colspan="2">2</td><td colspan="2">4</td><td>2</td><td>4</td><td colspan="2">2</td><td colspan="2">4</td><td>2</td><td>4</td><td colspan="2">2</td><td colspan="2">4</td><td>2</td><td>4</td></tr>
<tr><td>W_{NMOS} (nm)</td><td colspan="2">85</td><td colspan="2">170</td><td>85</td><td>170</td><td colspan="2">85</td><td colspan="2">170</td><td>85</td><td>170</td><td colspan="2">85</td><td colspan="2">170</td><td>85</td><td>170</td></tr>
</table>

[a] In these memories, CLK_1 and CLK_2 are connected to each other, and sharing the same buffers. See Figure 26.

3.4. Improvement of Maximum Operating Frequency

In order to find maximum operating frequency, the memory, we have to define what condition is defined as operating condition. If the voltage of dynamic nodes (bit lines) in the memory get lower than 10% of V_{DD} during discharge, and higher than 90% of V_{DD} during pre-charge, we define this clock frequency as operating frequency. We did extensive simulations in HSPICE and found the value of the highest frequency in which the memory is still operating. This frequency is defined as the maximum operating frequency, f_{max}. This f_{max} is found for memories that use TBDCs and the memories that don't have any delay cells. By comparing this two values, we can find how much the memory with TBDCs is faster than the memory without them. This difference is the amount that maximum operating frequency has improved. The amount of improvement of maximum operating frequency is listed in Table 6.

Table 6. The f_{max} improvemevt caused by TBDCs.

Technology Node	Mem.A	Mem.B	Mem.C
180 nm (CMOS)	106.7%	81.8%	68.7%
45 nm (CMOS)	110.7%	59.5%	43.3%
16 nm (High-K)	50.1%	21.2%	25.0%
7 nm (FinFET)	97.2%	78.7%	68.0%

3.5. Power Consumption of SRAM and Power Overhead Analysis

Adding TBDCs to the circuit comes at a cost. The TSVs and buffers in the TBDCs are going to increase the power consumption of the memory. It is important to measure how much the power overhead is. The power overhead is computed for when the operating frequency is the f_{max} for the memory without TBDCs. The amount of power overhead is shown in Table 7.

Table 7. Power Overhead in memories with TBDCs compared to memories without TBDCs.

Technology Node	Mem.A	Mem.B	Mem.C
180 nm (CMOS)	0.14%	0.37%	0.61%
45 nm (CMOS)	0.23%	0.66%	1.78%
16 nm (High-K)	0.21%	1.18%	5.00%
7 nm (FinFET)	1.47%	1.02%	3.83%

In our measurements, we did not consider the power consumption of interconnects and clock network in the circuit. If the power consumption of the interconnects were considered in the measurements, the power overhead would be lower than the values in Table 7.

3.6. Area Overhead of TSV Based Delay Cells and Traditional Delay Cells

When we add the TBDCs to the circuit, we are adding buffers to the circuits. The TSVs in the TBDCs are not adding anything to the circuit, because we are using the dummy TSVs that already exist in the circuit. We calculated to total area of the circuit, and compared to area of the added buffers, and calculated the area overhead of the TBDCs. When calculating the area of memories, the area occupied by the interconnects is not considered. The circuit is broken down into basic elements, then the area of these building blocks are measured, and with these values, we estimate the total area of the circuit. The building blocks include 2-input NAND, 3-input NAND, Inverter, Latch, and SRAM cells.

For the 180 nm CMOS memories, the design rules of GLOBALFOUNDRIES 7RF were used in designing the layout of the gates. NCSU FreePDK45 library was used for the layout of the gates in 45 nm technology. The numbers obtained from 45 nm were scaled down and used for 16 nm. For 7 nm FinFET, the design rules in [23] were used, the values were modified to match the parameters in PTM 7nm FinFET model. The Table 8 lists the design rules used for drawing the layout of gates in 7 nm memories.

Table 8. Design rules used for 7 nm FinFET.

Parameter	Value	Comment
L_{FIN}	11 nm	Fin length
T_{SI}	6.5 nm	Fin width
H_{FIN}	18 nm	Fin height
P_{FIN}	17.5 nm	Fin pitch
t_{ox}	1.15 nm	Oxide thickness
W_C	16.5 nm	Minimum contact size
W_{M2M}	16.5 nm	Minimum space between metal layers
W_{G2C}	11 nm	Minimum gate to contact space

In a similar way, the area of buffers was calculated, and by dividing the total area of buffers by total area of memory, the area overhead is obtained. The area of gates and building blocks are listed in Table 9, and Table 10 includes the total area of memories and the area overhead of the buffers.

Table 9. The area of gates and blocks in different technology nodes.

Technology Node	Inverter	NAND2	NAND3	SRAM Cell	Latch
180 nm (CMOS)	8.55 μm^2	11.32 μm^2	16.43 μm^2	20.69 μm^2	55.65 μm^2
45 nm (CMOS)	.6156 μm^2	.8436 μm^2	1.260 μm^2	1.080 μm^2	4.330 μm^2
16 nm (High-K)	0.06307 μm^2	0.08639 μm^2	0.1290 μm^2	0.1105 μm^2	0.4434 μm^2
7 nm (FinFET)	0.01663 μm^2	0.03245 μm^2	0.05326 μm^2	0.02951 μm^2	0.08955 μm^2

Table 10. Area of memories, buffers and area overhead.

Technology Node	Mem.A			Mem.B			Mem.C		
	Memory Area (μm^2)	Buffer Area (μm^2)	Area Overhead (%)	Memory Area (μm^2)	Buffer Area (μm^2)	Area Overhead (%)	Memory Area (μm^2)	Buffer Area (μm^2)	Area Overhead (%)
180 nm (CMOS)	810635	157.14	0.019	116665	157.27	0.135	17688	157.14	0.888
45 nm (CMOS)	45333	3.672	0.0081	6828	3.672	0.0538	1092.4	3.672	0.336
16 nm (High-K)	4642	0	0	699.2	0	0	111.9	0	0
7 nm (FinFET)	1294	0.0927	0.00717	199.0	0.0927	0.0466	32.03	0.0927	0.290

The traditional delay cells use an active circuit with transistors to generate the delay needed in the circuit. In 45 nm and 7 nm technologies for Mem.A we designed a traditional delay cell using inverters, and then we compared the area of silicon used in traditional delay cells and TBDCs. Table 11 shows the comparison results between these two kinds of delay cells.

Table 11. Comparison of Silicon area between traditional delay cells and TBDCs.

Technology Node	TBDC Area (μm^2)	Traditional Delay Cell Area (μm^2)	Area Saving
45 nm (CMOS)	3.672	14.688	75%
7 nm (FinFET)	0.0927	0.7883	88.2%

3.7. Power Supply Variation

One of the advantages of using TSVs for delay cell is their robustness to variations and fluctuations in power supply. TSV acts like a passive electrical circuit, and its behavior and characteristics don't change if the voltage supply changes. On the other hand, the conventional delay cells, that are made out of active elements such as transistors can be very sensitive to the variations and fluctuations in the power supply.

We studied the effects of power supply variation in three different cases. These cases are described in details below.

3.7.1. Case I: DC Variation in 45 nm

The nominal voltage for power supply in 45 nm technology node is 1.0 V. In this case, we are going to study the influence of changing V_{DD} on the performance of TBDCs. The two circuits that are compared to each other is Mem.A in 45 nm, one of the is using TBDCs, and the other is using conventional delay cells consisted of inverters. We are going to measure the delay between CLK_0 and CLK_3 in these circuits when the value of V_{DD} is up to 20% lower or higher than the nominal value. After measuring the delay, the values are normalized to make it easier to compare the delay of two different delay cells. The graph in Figure 27 shows the normalized delay in two circuits. The delay of the inverter based delay changes up to 31%, while the delay caused by TBDC does not deviate more than 13%.

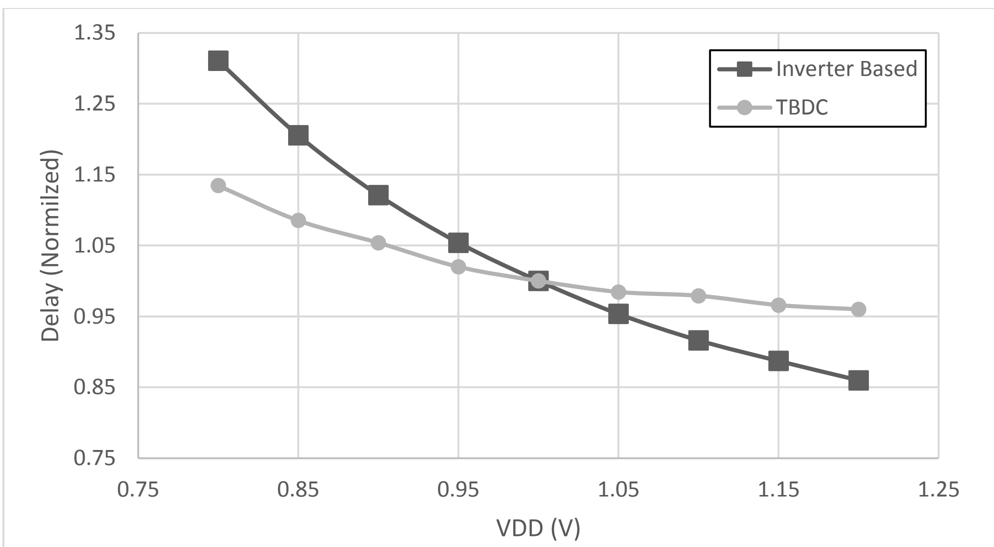


Figure 27. The delay of conventional and TSV based delay cells when V_{DD} changes.

3.7.2. Case II: Fluctuation Caused by Switching Noise in 45nm

The operation of the memory that we designed depends on a clock signal. Most transistors change their state from ON to OFF or OFF to ON on the rising and falling edges of clock. This causes a spike in current that memory draws from power supply. The graph in Figure

28 shows how current behaves relative to the CLK_0 signal in Mem.A in 45 nm technology node.

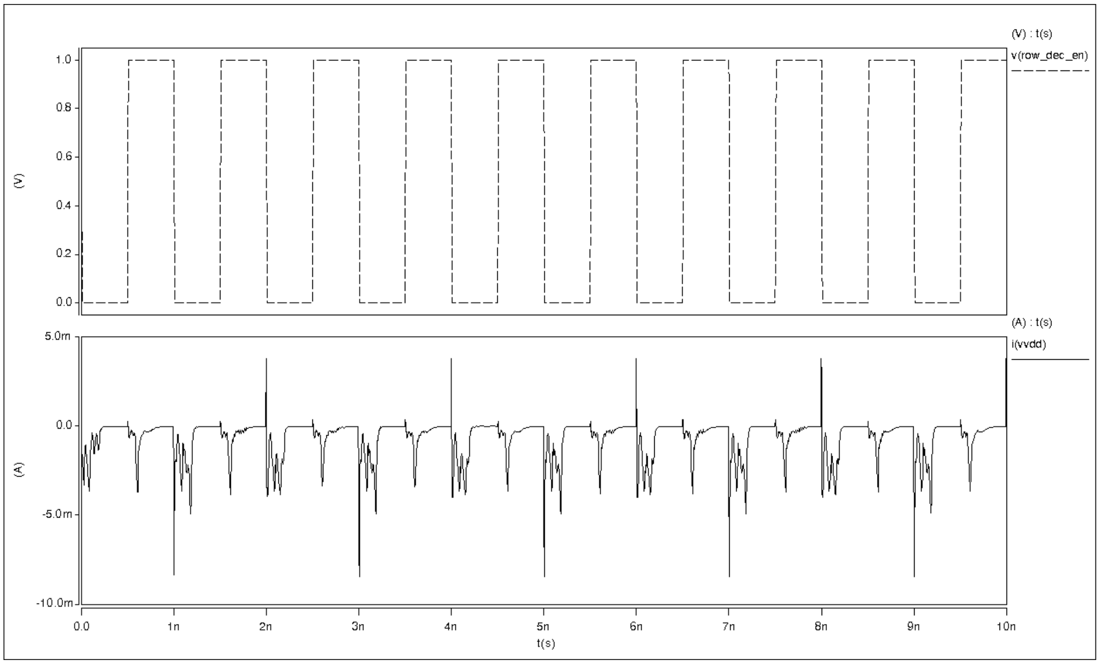


Figure 28. Signals in the SRAM, top: CLK_0 signal, bottom: Current from VDD.

In real circuits, the power and ground grid show the characteristics of an RLC circuit [24]. Similar to 3.7.1, in this case the Mem.A in 45 nm with TBDCs and inverter based delay cells are compared to each other. This time, an RLC circuit (Figure 29) is added on the way of power and ground signals. The RLC circuit causes the current and voltage to bounce, and the switching current adds fluctuation to the power supply. An example of these fluctuations is shown in Figure 30.

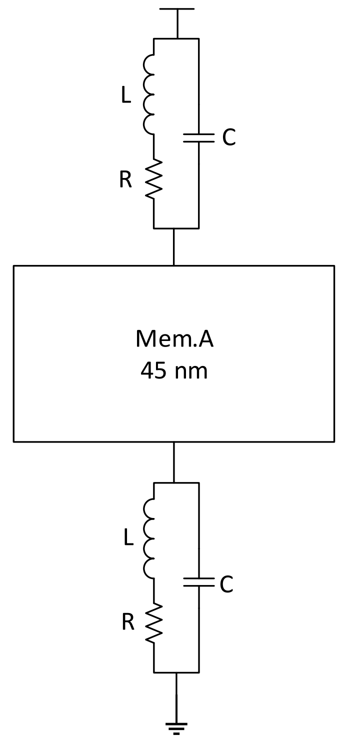


Figure 29. The RLC circuit on the way of VDD and GND.

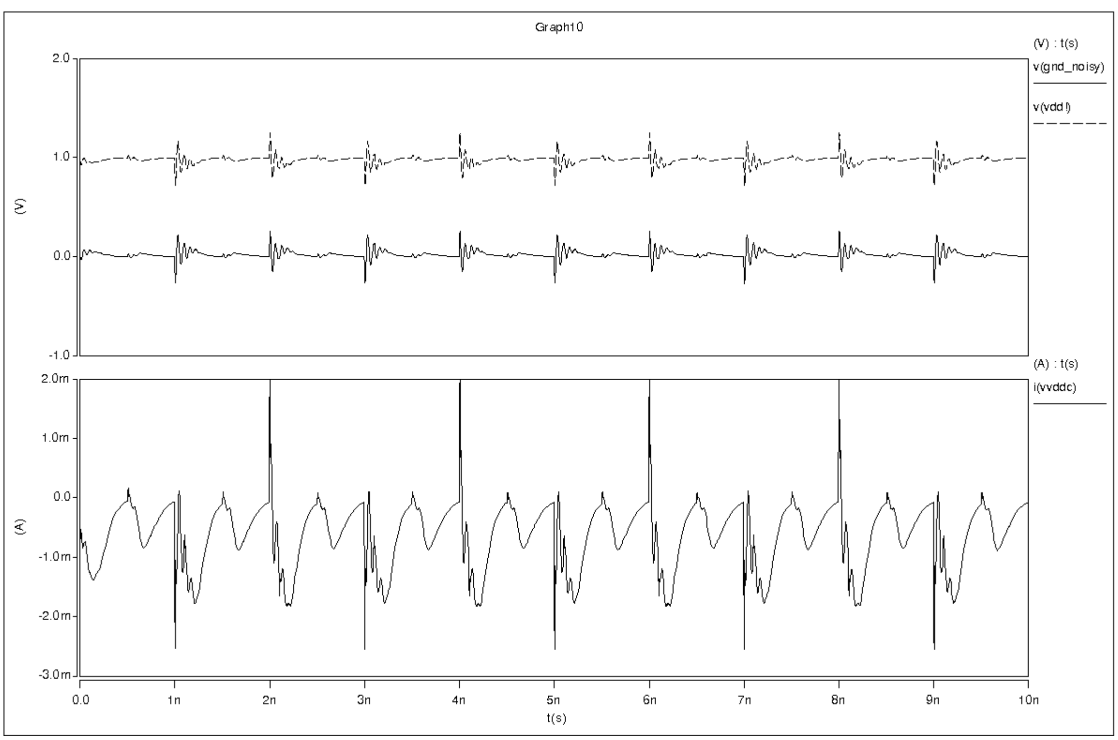


Figure 30. Signals in SRAM in presence of RLC power network, top: fluctuation in VDD and GND, bottom: current from VDD.

When the shape of current waveform in Figure 28 is compared to the waveform in Figure 30, it can be seen that in Figure 30 the amplitude of spikes is reduced, but an oscillating current is also present. This current passes through the RLC circuit, and causes a voltage drop across VDD and GND. Also, the waveform of VDD and GND has a underdamped oscillation.

Since the exact value of R, L, and C depend on the physical dimensions of metal interconnects, and wirebonds used in packaging, and these variables are unknown, a set of reasonable values for R, L, and C were used. The value of C is 0.05 pF, L changes from 0.5 nH to 1.5 nH, and R is in 10 Ω to 40 Ω range. Reference [24] offers values for R, L, C in 0.1 µm technology node. Those values are taken and scaled to 45 nm technology node. Similar to 4.6.1, the delay between CLK_0 and CLK_3 is compared. The results are shown in Figure 31.

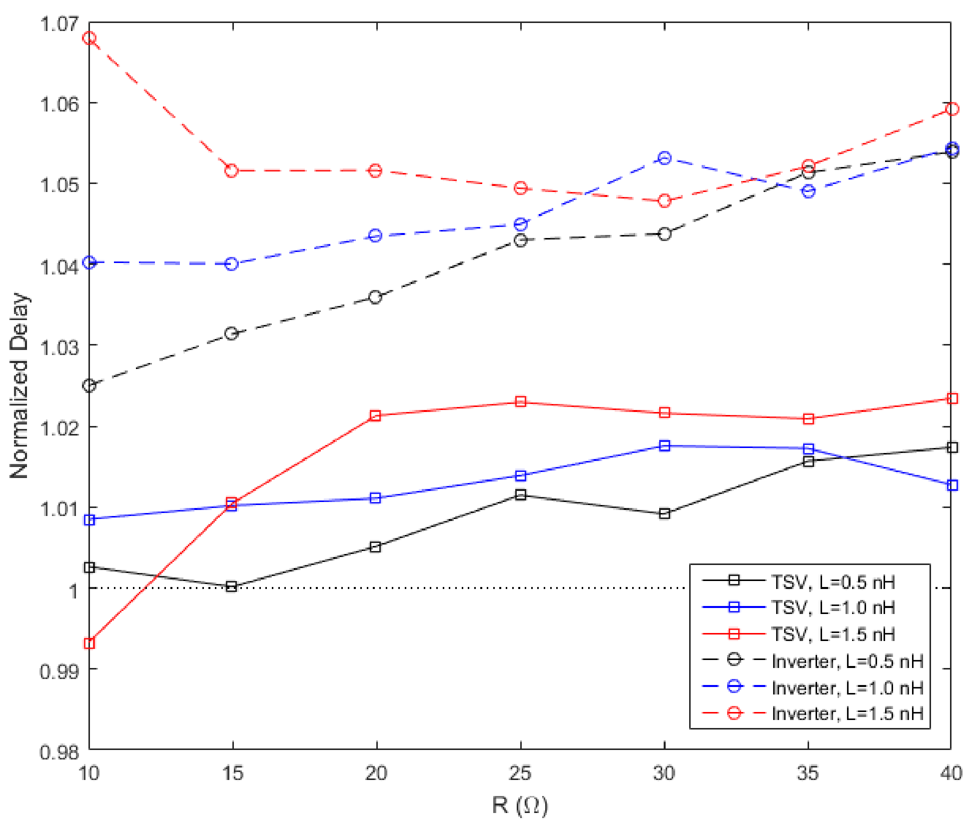


Figure 31. The delay of TSV and inverter based delay cell in the presence of RLC network.

In Figure 31, the closer the curve is to the horizontal dotted line at 1, the more robust the delay cell is against variations and fluctuations in power supply. In all cases the delay in circuits with TBDCs is more stable compared to the delay in circuits with inverter based delay cells.

3.7.3. Case III: Fluctuation Caused by Switching Noise in 7nm

When the size of transistors shrinks down, the capacitance and inductance of metal interconnects decrease [25]. The RLC network used in 3.7.2 can be simplified to a single R and be used in 7nm and still be accurate enough. The test conditions are similar to 3.7.2, but the measurement of the delay is done on Mem.A in 7nm technology node. The comparison results are shown in Figure 32. The normalized delay of TSV based delay cell shows a variation less than 0.2%, while the delay of inverter based delay cell changes up to 7.3%.

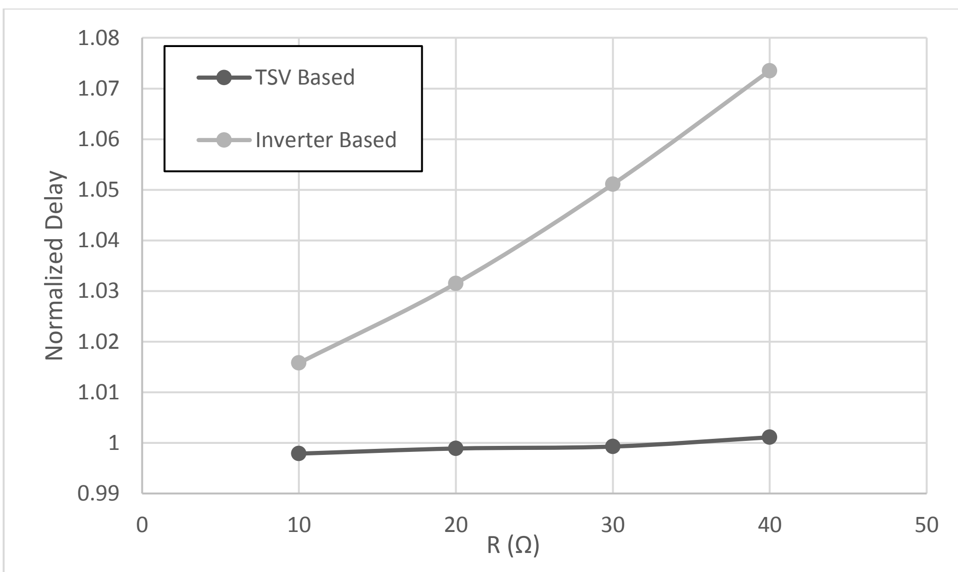


Figure 32. The delay of TSV and inverter based delay cell in presence of R in 7 nm Mem.A.

4. CONCLUSION

We proposed a novel method to use dummy TSVs as delay cells in 3D SRAM. These delay cells utilize the dummy TSVs that were previously unused, and the results show that this TSV based delay cells, can improve the performance of 3D SRAMs, increase the operating frequency up to 110%. The power overhead for using the TSVs for delay cells is negligible, and the silicon area used in less than the area used for traditional delay cells. Also, compared to conventional transistor based delay cells, the TSV based delay cells offer more robustness against variations and fluctuations in power supply.

4.1. Summary

3D ICs are becoming a more common technique used in VLSI chips. These chip are expected to be used for many applications, such as 3D memories and 3D SRAMs. Because of the design rules in 3D ICs, a large number of dummy TSVs are inserted that are not used for electrical purposes.

In this research, an equivalent circuit based on existing research was used to model TSVs. Then a series of delay cells with dummy TSVs and buffers were designed based on that model. These delay cells were incorporated in custom designed 3D SRAMs. Four group of measurements and comparisons were performed, to verify the effectiveness on the TSV based delay cells.

The first measurement, was maximum operating frequency. The circuit with delay cells can achieve a higher operating frequency, and this improvement can be as high as 110%.

The second measurement was the amount of power overhead. Adding TSVs and buffers to the memory can increase the power consumption of the memory, but our measurements show that this increase is very small and the power overhead is negligible (less than 5%).

Also, the area of the circuit was analyzed. The area added by buffers are compared to the total area of the circuit, and the amount of area overhead is calculated. This added area is negligible, and it is less than 0.9%. Also, we compared the area TSV based delay cells with conventional delay cells, and our measurements show that the TSV based delay cells use 88% less silicon area.

Lastly, the performance of TSV based delay cell was compared against the performance of conventional delay cells in the presence of voltage supply variation. Different scenarios were used in testing, and in all those scenarios, the TSV based delay cells were more robust than the conventional delay cells, and the delay generated by them was not influenced by variations and fluctuations in power supply.

4.2. Suggestions for Future Work

4.2.1. Reliability in Ultra Low-Voltage Circuits

Since the TSV based delay cells outperform the conventional delay cells in presence of voltage fluctuations, their effectiveness in ultra-low-voltage circuits can be studied. The conventional CMOS circuits have a very sensitive behavior in ultra-low-voltage settings, and using dummy TSVs can improve the performance and reliability in the circuit.

4.2.2. Thermal Analysis

Dummy TSVs are usually used for mechanical and thermal stability, but in this work they are included in the electrical circuit. Using dummy TSVs in the circuit can have negative or positive influence on the thermal dissipation and hot spots of the chip. These effects can be studied, and the dummy TSVs can be used for both purposes of improving the electrical performance of the chip, and thermal stability of the chip.

5. PUBLICATIONS

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

1- **Seyed Alireza Pourbakhsh**, Xiaowei Chen, Dongliang Chen, Xin Wang, Na Gong, Jinhui Wang, *"Sizing-Priority Based Low-Power Embedded Memory for Mobile Video Applications"*, 17th International Symposium on Quality Electronic Design, (ISQED) 2016, ***Nominated for Best Paper Award***.

2- Xiaowei Chen, **Seyed Alireza Pourbakhsh**, Ligang Hou, Na Gong, Jinhui Wang, *"Dummy TSV Based Bit-line Optimization in 3D On-chip Memory"*, 2016 International Electro/Information Technology Conference (IEEE EIT), ***Won Best Paper Award.***

3- Xiaowei Chen, **Seyed Alireza Pourbakhsh**, Ligang Hou, Na Gong, Jinhui Wang, *"Dummy TSV Based Delay Cell Optimization in 3D On-chip Memory"*, The 5th International Symposium on Next-Generation Electronics (ISNE 2016).

4- Xiaowei Chen, **Seyed Alireza Pourbakhsh**, Na Gong, Jinhui Wang, *"Dummy TSV-cluster-based Bit-line Timing Optimization for 3D on-chip Memory"*, for submission to IEEE Transactions on Circuits and Systems I.

REFERENCES

[1] D. H. Kim and e. al., "Design and Analysis of 3D-MAPS (3D Massively Parallel Processor with Stacked Memory)," *Computers, IEEE Transactions on,* vol. 64, no. 1, pp. 112-125, Jan. 2015.

[2] T. Zhang, K. Wang, Y. Feng, Y. Chen, Q. Li, B. Shao, J. Xie, X. Song, L. Duan, Y. Xie, X. Cheng and Y.-L. Lin, "A 3D SoC design for H.264 application with on-chip DRAM stacking," in *3D Systems Integration Conference (3DIC), 2010 IEEE International*, Munich, 2010.

[3] S. K. Lim, Design for High Performance, Low Power, and Reliable 3D Integrated Circuits, New York: Springer, 2013.

[4] F. X. Che, "Dynamic Stress Modeling on Wafer Thinning Process and Reliability Analysis for TSV Wafer," *IEEE Transactions on Components, Packaging and Manufacturing Technology,* vol. 4, no. 9, pp. 1432 - 1440 , Sept. 2014.

[5] N. Gong, J. Wang, S. Jiang and R. Sridhar, "Clock-biased local bit line for high," *Electronic Letters,* vol. 48, no. 18, pp. 1104 - 1105, Aug. 2012.

[6] A. Ghaffari and A. Abrishamifar, "A Novel Wide-Range Delay Cell for DLLs," in *2006 International Conference on Electrical and Computer Engineering*, Dhaka, 19-21 Dec. 2006.

[7] J. Kim, D. G. Kam, P. J. Jun and J. Kim, "Spread spectrum clock Generator with delay cell array to reduce electromagnetic interference," *IEEE Transactions on Electromagnetic Compatibility,* vol. 47, no. 4, pp. 908 - 920, Nov. 2005.

[8] S. Choi, S. Yoo, Y. Lim and J. Choi, "A PVT-Robust and Low-Jitter Ring-VCO-Based Injection-Locked Clock Multiplier With a Continuous Frequency-Tracking Loop Using a Replica-Delay Cell and a Dual-Edge Phase Detector," *IEEE Journal of Solid-State Circuits,* vol. PP, no. 99, pp. 1 - 12, June 2016.

[9] H. Kim, J. Cho, M. Kim, K. Kim, J. Lee, H. Lee, K. Park, K. Choi, H.-C. Bae, J. Kim and J. Kim, "Measurement and Analysis of a High-Speed TSV Channel," *IEEE Transactions on Components, Packaging and Manufacturing Technology,* vol. 2, no. 10, pp. 1672 - 1685, Sept. 2012.

[10] P. Li, Y. Shang, H. Yu and M. Yu, "A wideband 150GHz antenna by 3D-TSV based Composite Right/Left Handed Transmission Line for Sub-THz biomedical imaging," in *Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO), 2013 IEEE MTT-S International*, Singapore, 9-11 Dec. 2013.

[11] Y.-J. Lee and S. K. Lim, "Fast delay estimation with buffer insertion for through-silicon-via-based 3D interconnects," in *Thirteenth International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, 19-21 March 2012.

[12] Z. Chen, Q. Wang, J. Xie, J. Tian, J. Jiang, Y. Li and W. Yin, "Modeling and analysis of signal transmission with Through Silicon Via (TSV) noise coupling," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, Beijing, 19-23 May 2013.

[13] L. Qinghai, M. Min, L. Zhensong, L. Qinghai and M. Min, "Study of Ground-Signal-Ground TSV in terms of transmission performance," in *Electronic Packaging Technology (ICEPT), 2014 15th International Conference on*, Chengdu, 12-15 Aug. 2014.

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

[15] M. Watanabe, N. Niioka, T. Kobayashi, R. Karel, M.-a. Fukase, M. Imai and A. Kurokawa, "An effective model for evaluating vertical propagation delay in TSV-based 3-D ICs," in *Sixteenth International Symposium on Quality Electronic Design*, Santa Clara, CA, 2-4 March 2015.

[16] G. Katti, M. Stucchi, K. D. Meyer and W. Dehaene, "Electrical Modeling and Characterization of Through Silicon via for Three-Dimensional ICs," *IEEE Transactions on Electron Devices,* vol. 57, no. 1, pp. 256 - 262, Dec. 2009.

[17] J.-W. You, S.-Y. Huang, D.-M. Kwai, Y.-F. Chou and C.-W. Wu, "Performance Characterization of TSV in 3D IC via Sensitivity Analysis," in *2010 19th IEEE Asian Test Symposium*, Shanghai, 1-4 Dec. 2010.

[18] 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, pp. 266 - 279, Feb. 2015.

[19] 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, pp. 247 - 259, Feb. 2011.

[20] M. P. D. Sai, H. Yu, Y. Shang, C. S. Tan and S. K. Lim, "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, pp. 1734 - 1747, Nov. 2013.

[21] K. Nomura, K. Abe, S. Fujita, Y. Kurosawa and A. Kageshima, "Performance analysis of 3D-IC for multi-core processors in sub-65nm CMOS technologies," in *Proceedings of*

2010 IEEE International Symposium on Circuits and Systems, Paris, May 30 2010-June 2 2010.

[22] S.-H. Ok, K.-r. Bae, S. K. Lim and B. Moon, "Design and analysis of 3D IC-based low power stereo matching processors," in *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, Beijing, 4-6 Sept. 2013.

[23] T. Cui, Q. Xie, Y. Wang, S. Nazarian and M. Pedram, "7nm FinFET standard cell layout characterization and power density prediction in near- and super-threshold voltage regimes," in *Green Computing Conference (IGCC), 2014 International*, Dallas, TX, 3-5 Nov. 2014.

[24] K. T. Tang and E. G. Friedman, "Delay uncertainty due to on-chip simultaneous switching noise in high performance CMOS integrated circuits," in *Signal Processing Systems, 2000. SiPS 2000. 2000 IEEE Workshop on*, Lafayette, LA, 11-13 Oct 2000.

[25] J. M. Rabaey, A. Chandrakasan and B. Nikolic, Digital Integrated Circuits: A Design Perspective, 2nd ed., Prentice Hall, 2003, p. 172.

[26] 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, pp. 510-523, April 2013.

[27] V. F. Pavlidis and E. G. Friedman, Three-dimensional Integrated Circuit Design, Burlington, MA: Morgan Kaufmann, 2009.

[28] Y. Shang, C. Zhang, H. Yu, C. S. Tan, X. Zhao and S. K. Lim, "Thermal-reliable 3D clock-tree synthesis considering nonlinear electrical-thermal-coupled TSV model," in *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, Yokohama, 22-25 Jan. 2013.

APPENDIX A. HSPICE NETLIST USED FOR 45 NM MEM.A

```
.GLOBAL vdd!
.PARAM d0=0 d1=0 d2=0 t=590p p5vonly=0 phires=0 VDD_VAL=1.0


.PROBE TRAN
+    I(v00)
+    I(vrc)
+    I(vck1)
+    I(vck2)
+    I(vck3)
.TRAN 1p 8e-9

.OP

.TEMP 25.0
.OPTION
+    ARTIST=2
+    INGOLD=2
+    PARHIER=LOCAL
+    PSF=2
+        LIST
+        NODE
+        POST

.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\PMOS_VTL.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\PMOS_THKOX.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\PMOS_VTG.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\PMOS_VTH.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\NMOS_VTL.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\NMOS_VTG.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\NMOS_VTH.inc"
.INCLUDE "C:\Users\SeyedAlireza\Documents\HSPICE\NCSU\NMOS_THKOX.inc"


.subckt D_Latch_1_45nm clk d q inh_gnd inh_vdd
m10 q net16 inh_vdd inh_vdd PMOS_VTG L=50e-9 W=600e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m11 net031 net16 inh_vdd inh_vdd PMOS_VTG L=50e-9 W=280e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m0 net16 net031 inh_vdd inh_vdd PMOS_VTG L=50e-9 W=190e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m9 net16 net10 net15 inh_vdd PMOS_VTG L=50e-9 W=900e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-
9 M=1
m8 net15 d inh_vdd inh_vdd PMOS_VTG L=50e-9 W=900e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m7 net10 clk inh_vdd inh_vdd PMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-
9 M=1
m6 net031 net16 inh_gnd inh_gnd NMOS_VTG L=50e-9 W=140e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m5 q net16 inh_gnd inh_gnd NMOS_VTG L=50e-9 W=480e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m4 net16 net031 inh_gnd inh_gnd NMOS_VTG L=50e-9 W=80e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m2 net16 clk net18 inh_gnd NMOS_VTG L=50e-9 W=450e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m3 net18 d inh_gnd inh_gnd NMOS_VTG L=50e-9 W=450e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m1 net10 clk inh_gnd inh_gnd NMOS_VTG L=50e-9 W=180e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-
9 M=1
.ends D_Latch_1_45nm


.subckt inverter_old_45nm in out
m0 out in 0 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 out in vdd! vdd! PMOS_VTG L=50e-9 W=670e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
.ends inverter_old_45nm
```

```
.subckt nand_3_45nm a b c z
m3 z c net13 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m5 net12 a 0 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m4 net13 b net12 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m2 z c vdd! vdd! PMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 z b vdd! vdd! PMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m0 z a vdd! vdd! PMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
.ends nand_3_45nm

.subckt dec2_4_45nm en in0 in1 y0 y1 y2 y3
xi14 net027 y3 inverter_old_45nm
xi13 net026 y2 inverter_old_45nm
xi12 net025 y1 inverter_old_45nm
xi11 net024 y0 inverter_old_45nm
xi0 in0 net15 inverter_old_45nm
xi1 in1 net16 inverter_old_45nm
xi6 net15 net16 en net024 nand_3_45nm
xi7 in0 net16 en net025 nand_3_45nm
xi9 in0 in1 en net027 nand_3_45nm
xi8 net15 in1 en net026 nand_3_45nm
.ends dec2_4_45nm

.subckt nand_2_45nm a b z
m2 z a net9 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m3 net9 b 0 0 NMOS_VTG L=50e-9 W=330e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 z a vdd! vdd! PMOS_VTG L=50e-9 W=500e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m0 z b vdd! vdd! PMOS_VTG L=50e-9 W=500e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
.ends nand_2_45nm

.subckt dec3_8_En_45nm a0 a1 a2 en y0 y1 y2 y3 y4 y5 y6 y7
xi93 net27 net25 net23 net039 nand_3_45nm
xi92 net27 net25 a0 net040 nand_3_45nm
xi91 net27 a1 net23 net043 nand_3_45nm
xi90 net27 a1 a0 net042 nand_3_45nm
xi89 a2 net25 net23 net044 nand_3_45nm
xi88 a2 net25 a0 net041 nand_3_45nm
xi87 a2 a1 net23 net046 nand_3_45nm
xi86 a2 a1 a0 net045 nand_3_45nm
xi101 net031 en net023 nand_2_45nm
xi100 net032 en net024 nand_2_45nm
xi99 net036 en net028 nand_2_45nm
xi98 net033 en net025 nand_2_45nm
xi97 net034 en net026 nand_2_45nm
xi96 net035 en net027 nand_2_45nm
xi95 net037 en net029 nand_2_45nm
xi94 net038 en net030 nand_2_45nm
xi85 net023 y0 inverter_old_45nm
xi84 net024 y1 inverter_old_45nm
xi83 net025 y2 inverter_old_45nm
xi82 net026 y3 inverter_old_45nm
xi81 net027 y4 inverter_old_45nm
xi80 net028 y5 inverter_old_45nm
xi79 net029 y6 inverter_old_45nm
xi78 net030 y7 inverter_old_45nm
xi77 net040 net032 inverter_old_45nm
xi76 net039 net031 inverter_old_45nm
xi75 net043 net033 inverter_old_45nm
xi74 net042 net034 inverter_old_45nm
xi73 net044 net035 inverter_old_45nm
xi72 net041 net036 inverter_old_45nm
xi71 net046 net037 inverter_old_45nm
xi70 net045 net038 inverter_old_45nm
xi69 a0 net23 inverter_old_45nm
xi68 a1 net25 inverter_old_45nm
xi67 a2 net27 inverter_old_45nm
.ends dec3_8_En_45nm

.subckt dec5_32_FF_45nm a0 a1 a2 a3 a4 en w0 w1 w10 w11 w12 w13 w14 w15 w16 w17 w18 w19 w2 w20
w21 w22 w23 w24 w25 w26 w27 w28 w29 w3 w30 w31 w4 w5 w6 w7 w8 w9 inh_gnd inh_vdd
xi26 en a4 a_ff<4> inh_gnd inh_vdd D_Latch_1_45nm
xi25 en a3 a_ff<3> inh_gnd inh_vdd D_Latch_1_45nm
```

```
xi24 en a2 a_ff<2> inh_gnd inh_vdd D_Latch_1_45nm
xi22 en a0 a_ff<0> inh_gnd inh_vdd D_Latch_1_45nm
xi23 en a1 a_ff<1> inh_gnd inh_vdd D_Latch_1_45nm
xi4 vdd! a_ff<3> a_ff<4> net9 net10 net8 net7 dec2_4_45nm
xi12 a_ff<0> a_ff<1> a_ff<2> net7 w24 w25 w26 w27 w28 w29 w30 w31 dec3_8_En_45nm
xi11 a_ff<0> a_ff<1> a_ff<2> net8 w16 w17 w18 w19 w20 w21 w22 w23 dec3_8_En_45nm
xi10 a_ff<0> a_ff<1> a_ff<2> net10 w8 w9 w10 w11 w12 w13 w14 w15 dec3_8_En_45nm
xi9 a_ff<0> a_ff<1> a_ff<2> net9 w0 w1 w2 w3 w4 w5 w6 w7 dec3_8_En_45nm
.ends dec5_32_FF_45nm

.subckt PreCharge_PMOS_45nm precharge_clk rbl<0> rbl<10> rbl<11> rbl<12> rbl<13> rbl<14> rbl<15>
rbl<16> rbl<17> rbl<18> rbl<19> rbl<1> rbl<20> rbl<21> rbl<22> rbl<23> rbl<24> rbl<25> rbl<26>
rbl<27> rbl<28> rbl<29> rbl<2> rbl<30> rbl<31> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7> rbl<8> rbl<9>
m31 rbl<0> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m30 rbl<1> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m29 rbl<2> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m28 rbl<3> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m27 rbl<4> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m26 rbl<5> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m25 rbl<6> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m24 rbl<7> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m23 rbl<8> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m22 rbl<9> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m21 rbl<10> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m20 rbl<11> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m19 rbl<12> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m18 rbl<13> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m17 rbl<14> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m16 rbl<15> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m15 rbl<16> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m14 rbl<17> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m13 rbl<18> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m12 rbl<19> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m11 rbl<20> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m10 rbl<21> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m9 rbl<22> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m8 rbl<23> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m7 rbl<24> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m6 rbl<25> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m5 rbl<26> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m4 rbl<27> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m3 rbl<28> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
```

```
m2 rbl<29> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m1 rbl<30> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
m0 rbl<31> precharge_clk vdd! vdd! PMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
.ends PreCharge_PMOS_45nm

.subckt SRAM_8T_45nm q rbl rwl wbl wblb wwl
m2 net28 q vdd! vdd! PMOS_VTG L=50e-9 W=100e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m0 q net28 vdd! vdd! PMOS_VTG L=50e-9 W=100e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m5 wbl wwl q 0 NMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m4 net28 wwl wblb 0 NMOS_VTG L=50e-9 W=200e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m3 net28 q 0 0 NMOS_VTG L=50e-9 W=100e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 q net28 0 0 NMOS_VTG L=50e-9 W=100e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m6 net34 q 0 0 NMOS_VTG L=50e-9 W=90e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m7 rbl rwl net34 0 NMOS_VTG L=50e-9 W=90e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
.ends SRAM_8T_45nm

.subckt SRAM_32x32_45nm rbl<0> rbl<10> rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17>
rbl<18> rbl<19> rbl<1> rbl<20> rbl<21> rbl<22> rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28>
rbl<29> rbl<2> rbl<30> rbl<31> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7> rbl<8> rbl<9> rwl<0> rwl<10>
rwl<11> rwl<12> rwl<13> rwl<14> rwl<15> rwl<16> rwl<17> rwl<18> rwl<19> rwl<1> rwl<20> rwl<21>
rwl<22> rwl<23> rwl<24> rwl<25> rwl<26> rwl<27> rwl<28> rwl<29> rwl<2> rwl<30> rwl<31> rwl<3>
rwl<4> rwl<5> rwl<6> rwl<7> rwl<8> rwl<9> wbl<0> wbl<10> wbl<11> wbl<12> wbl<13> wbl<14> wbl<15>
wbl<16> wbl<17> wbl<18> wbl<19> wbl<1> wbl<20> wbl<21> wbl<22> wbl<23> wbl<24> wbl<25> wbl<26>
wbl<27> wbl<28> wbl<29> wbl<2> wbl<30> wbl<31> wbl<3> wbl<4> wbl<5> wbl<6> wbl<7> wbl<8> wbl<9>
wblb<0> wblb<10> wblb<11> wblb<12> wblb<13> wblb<14> wblb<15> wblb<16> wblb<17> wblb<18> wblb<19>
wblb<1> wblb<20> wblb<21> wblb<22> wblb<23> wblb<24> wblb<25> wblb<26> wblb<27> wblb<28> wblb<29>
wblb<2> wblb<30> wblb<31> wblb<3> wblb<4> wblb<5> wblb<6> wblb<7>
+wblb<8> wblb<9> wwl<0> wwl<10> wwl<11> wwl<12> wwl<13> wwl<14> wwl<15> wwl<16> wwl<17> wwl<18>
wwl<19> wwl<1> wwl<20> wwl<21> wwl<22> wwl<23> wwl<24> wwl<25> wwl<26> wwl<27> wwl<28> wwl<29>
wwl<2> wwl<30> wwl<31> wwl<3> wwl<4> wwl<5> wwl<6> wwl<7> wwl<8> wwl<9>
xi1023 net321 rbl<31> rwl<31> wbl<31> wblb<31> wwl<31> SRAM_8T_45nm
xi1022 net322 rbl<31> rwl<30> wbl<31> wblb<31> wwl<30> SRAM_8T_45nm
xi1021 net323 rbl<31> rwl<29> wbl<31> wblb<31> wwl<29> SRAM_8T_45nm
xi1020 net324 rbl<31> rwl<28> wbl<31> wblb<31> wwl<28> SRAM_8T_45nm
xi1019 net325 rbl<31> rwl<27> wbl<31> wblb<31> wwl<27> SRAM_8T_45nm
xi1018 net326 rbl<31> rwl<26> wbl<31> wblb<31> wwl<26> SRAM_8T_45nm
**
** ALL 32X32 SRAM CELLS SHOULD BE INSERTED HERE
** THEY ARE DELETED TO MAKE THE APPENDIX SHORTER
**
**
xi4 net1340 rbl<0> rwl<4> wbl<0> wblb<0> wwl<4> SRAM_8T_45nm
xi3 net1341 rbl<0> rwl<3> wbl<0> wblb<0> wwl<3> SRAM_8T_45nm
xi2 net1342 rbl<0> rwl<2> wbl<0> wblb<0> wwl<2> SRAM_8T_45nm
xi1 net1343 rbl<0> rwl<1> wbl<0> wblb<0> wwl<1> SRAM_8T_45nm
xi0 net1344 rbl<0> rwl<0> wbl<0> wblb<0> wwl<0> SRAM_8T_45nm
.ends SRAM_32x32_45nm

.subckt AO_2x32_Dynamic_45nm a<0> a<10> a<11> a<12> a<13> a<14> a<15> a<16> a<17> a<18> a<19>
a<1> a<20> a<21> a<22> a<23> a<24> a<25> a<26> a<27> a<28> a<29> a<2> a<30> a<31> a<3> a<4> a<5>
a<6> a<7> a<8> a<9> b<0> b<10> b<11> b<12> b<13> b<14> b<15> b<16> b<17> b<18> b<19> b<1> b<20>
b<21> b<22> b<23> b<24> b<25> b<26> b<27> b<28> b<29> b<2> b<30> b<31> b<3> b<4> b<5> b<6> b<7>
b<8> b<9> clk output
m11 output a<26> net097 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m12 net097 b<26> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m9 output a<27> net098 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m10 net098 b<27> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m7 output a<28> net099 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m8 net099 b<28> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m6 net0100 b<29> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 output a<31> net167 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m13 output a<25> net096 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m14 net096 b<25> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
```

```
m15 output a<24> net095 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m16 net095 b<24> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m17 output a<23> net094 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m18 net094 b<23> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m19 output a<22> net093 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m20 net093 b<22> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m21 output a<21> net092 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m22 net092 b<21> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m23 output a<20> net091 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m24 net091 b<20> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m26 net090 b<19> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m25 output a<19> net090 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m28 net089 b<18> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m27 output a<18> net089 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m30 net088 b<17> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m29 output a<17> net088 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m31 output a<16> net087 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m32 net087 b<16> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m33 output a<15> net086 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m34 net086 b<15> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m35 output a<14> net085 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m36 net085 b<14> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m38 net084 b<13> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m37 output a<13> net084 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m40 net083 b<12> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m39 output a<12> net083 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m42 net082 b<11> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m41 output a<11> net082 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m44 net081 b<10> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m43 output a<10> net081 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m47 output a<8> net079 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m46 net080 b<9> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m45 output a<9> net080 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m48 net079 b<8> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m50 net078 b<7> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m49 output a<7> net078 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m52 net077 b<6> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m51 output a<6> net077 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m54 net076 b<5> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m53 output a<5> net076 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m56 net075 b<4> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m55 output a<4> net075 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m58 net074 b<3> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m57 output a<3> net074 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m60 net073 b<2> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m59 output a<2> net073 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m62 net072 b<1> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
```

```
m61 output a<1> net072 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m64 net071 b<0> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m63 output a<0> net071 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m5 output a<29> net0100 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m2 net167 b<31> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m4 net0101 b<30> 0 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m3 output a<30> net0101 0 NMOS_VTG L=50e-9 W=850e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
m0 output clk vdd! vdd! PMOS_VTG L=50e-9 W=8.5e-6 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
.ends AO_2x32_Dynamic_45nm

.subckt dec5_32_45nm a0 a1 a2 a3 a4 en w0 w1 w10 w11 w12 w13 w14 w15 w16 w17 w18 w19 w2 w20 w21
w22 w23 w24 w25 w26 w27 w28 w29 w3 w30 w31 w4 w5 w6 w7 w8 w9
xi12 a0 a1 a2 net7 w24 w25 w26 w27 w28 w29 w30 w31 dec3_8_En_45nm
xi11 a0 a1 a2 net8 w16 w17 w18 w19 w20 w21 w22 w23 dec3_8_En_45nm
xi10 a0 a1 a2 net10 w8 w9 w10 w11 w12 w13 w14 w15 dec3_8_En_45nm
xi9 a0 a1 a2 net9 w0 w1 w2 w3 w4 w5 w6 w7 dec3_8_En_45nm
xi4 en a3 a4 net9 net10 net8 net7 dec2_4_45nm
.ends dec5_32_45nm

.subckt SRAM_32x32_Block_NewDec_45nm col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4>
col_dec_en lvl1_clk lvl2_bitline lvl2_clk row_addr<0> row_addr<1> row_addr<2> row_addr<3>
row_addr<4> row_dec_en inh_gnd inh_vdd
xi1 lvl1_clk net047 net037 net036 net035 net034 net033 net032 net031 net030 net029 net028 net046
net027 net026 net025 net024 net023 net022 net021 net020 net019 net018 net045 net017 net016 net044
net043 net042 net041 net040 net039 net038 PreCharge_PMOS_45nm
xi39 net047 net226 inverter_old_45nm
xi38 net046 net2 inverter_old_45nm
xi37 net045 net3 inverter_old_45nm
xi36 net044 net164 inverter_old_45nm
xi35 net043 net165 inverter_old_45nm
xi34 net042 net231 inverter_old_45nm
xi33 net041 net7 inverter_old_45nm
xi32 net040 net8 inverter_old_45nm
xi31 net039 net169 inverter_old_45nm
xi30 net038 net170 inverter_old_45nm
xi29 net037 net171 inverter_old_45nm
xi28 net036 net237 inverter_old_45nm
xi27 net035 net238 inverter_old_45nm
xi26 net034 net174 inverter_old_45nm
xi25 net033 net240 inverter_old_45nm
xi24 net032 net16 inverter_old_45nm
xi23 net031 net17 inverter_old_45nm
xi22 net030 net243 inverter_old_45nm
xi21 net029 net19 inverter_old_45nm
xi20 net028 net180 inverter_old_45nm
xi19 net027 net246 inverter_old_45nm
xi18 net026 net22 inverter_old_45nm
xi17 net025 net183 inverter_old_45nm
xi16 net024 net249 inverter_old_45nm
xi15 net023 net250 inverter_old_45nm
xi14 net022 net186 inverter_old_45nm
xi13 net021 net27 inverter_old_45nm
xi12 net020 net188 inverter_old_45nm
xi11 net019 net29 inverter_old_45nm
xi10 net018 net190 inverter_old_45nm
xi8 net016 net32 inverter_old_45nm
xi9 net017 net31 inverter_old_45nm
m1 lvl2_bitline net084 vdd! vdd! PMOS_VTG L=50e-9 W=220e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9
PS=300e-9 M=1
xi0 net047 net037 net036 net035 net034 net033 net032 net031 net030 net029 net028 net046 net027
net026 net025 net024 net023 net022 net021 net020 net019 net018 net045 net017 net016 net044 net043
net042 net041 net040 net039 net038 net129 net139 net140 net141 net142 net143 net144 net145 net146
net147 net148 net130 net149 net150 net151 net152 net153 net154 net155 net156 net157 net158 net131
net159 net160 net132 net133 net134 net135 net136 net137 net138 net33 net43 net44 net45 net46
net47 net48 net49 net50 net51 net52 net34 net53 net54 net55 net56 net57 net58 net59 net60 net61
net62 net35 net63 net64 net36 net37 net38 net39 net40 net41 net42 net65 net75 net76 net77 net78
net79 net80 net81 net82 net83 net84 net66 net85 net86 net87 net88 net89 net90 net91 net92 net93
```

```
net94 net67 net95 net96 net68 net69 net70 net71 net72 net73 net74 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 SRAM_32x32_45nm
xi2 net32 net22 net246 net180 net19 net243 net17 net16 net240 net174 net238 net31 net237 net171
net170 net169 net8 net7 net231 net165 net164 net3 net190 net2 net226 net29 net188 net27 net186
net250 net249 net183 net0226 net215 net214 net213 net212 net0212 net210 net0210 net0209 net0208
net0207 net224 net205 net0205 net203 net0203 net201 net0201 net199 net0199 net197 net0197 net0224
net195 net194 net0223 net0222 net0221 net219 net0219 net217 net0217 lvl2_clk net084
AO_2x32_Dynamic_45nm
xi3 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en net129 net130 net139
net140 net141 net142 net143 net144 net145 net146 net147 net148 net131 net149 net150 net151 net152
net153 net154 net155 net156 net157 net158 net132 net159 net160 net133 net134 net135 net136 net137
net138 dec5_32_45nm
xi5 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en net194 net195 net0205
net205 net0207 net0208 net0209 net0210 net210 net0212 net212 net213 net0197 net214 net215 net0217
net217 net0219 net219 net0221 net0222 net0223 net0224 net197 net224 net0226 net0199 net199
net0201 net201 net0203 net203 inh_gnd inh_vdd dec5_32_FF_45nm
m0 lvl2_bitline net084 0 0 NMOS_VTG L=50e-9 W=110e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1
.ends SRAM_32x32_Block_NewDec_45nm

.subckt SRAM_32x32x32_NewDec_45nm block_addr<0> block_addr<1> block_addr<2> block_addr<3>
block_addr<4> block_dec_en col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en
clk1 clk2 lvl3_bitline lvl3_clk row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4>
row_dec_en inh_gnd inh_vdd
xi32 block_addr<0> block_addr<1> block_addr<2> block_addr<3> block_addr<4> block_dec_en net48
net47 net38 net37 net36 net35 net34 net33 net32 net31 net30 net29 net46 net28 net27 net26 net25
net24 net23 net22 net21 net20 net19 net45 net18 net17 net44 net43 net42 net41 net40 net39 inh_gnd
inh_vdd dec5_32_FF_45nm
xi64 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<28> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi63 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<24> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi62 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<20> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi61 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<16> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi60 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<25> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi59 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<29> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi58 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<21> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi57 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<17> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi56 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<26> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi55 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<30> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi54 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<22> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi53 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<18> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi52 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<31> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi51 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<27> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
```

```
xi50 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<23> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi49 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<19> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi48 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<8> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi47 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<9> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi46 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<12> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi45 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<13> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi44 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<10> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi43 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<11> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi42 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<15> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi41 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<14> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi40 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<4> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi39 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<7> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi38 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<6> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi37 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<5> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi36 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<2> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi34 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<1> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi35 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<3> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi31 col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1 bitline<0> clk2
row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en inh_gnd inh_vdd
SRAM_32x32_Block_NewDec_45nm
xi33 bitline<31> bitline<21> bitline<20> bitline<19> bitline<18> bitline<17> bitline<16>
bitline<15> bitline<14> bitline<13> bitline<12> bitline<30> bitline<11> bitline<10> bitline<9>
bitline<8> bitline<7> bitline<6> bitline<5> bitline<4> bitline<3> bitline<2> bitline<29>
bitline<1> bitline<0> bitline<28> bitline<27> bitline<26> bitline<25> bitline<24> bitline<23>
bitline<22> net17 net27 net28 net29 net30 net31 net32 net33 net34 net35 net36 net18 net37 net38
net39 net40 net41 net42 net43 net44 net45 net46 net19 net47 net48 net20 net21 net22 net23 net24
net25 net26 lvl3_clk lvl3_bitline AO_2x32_Dynamic_45nm
.ends SRAM_32x32x32_NewDec_45nm


m0 net015 net014 0 0 NMOS_VTG L=50e-9 W=110e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9 M=1
m1 net015 net014 vdd! vdd! PMOS_VTG L=50e-9 W=220e-9 AD=9.45e-15 AS=9.45e-15 PD=300e-9 PS=300e-9
M=1

xi0 block_addr<0> block_addr<1> block_addr<2> block_addr<3> block_addr<4> block_dec_en
col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en lvl1_clk lvl2_clk net014
```

```
lvl3_clk row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en 0 vdd!
SRAM_32x32x32_NewDec_45nm


v00 vdd! 0 DC=VDD_VAL
**     n+      n-   type      vhi   vlo  td           tr     tf       tsample
vr4 row_addr<4> 0      PAT    VDD_VAL 0       0p    'T/100' 'T/100'       'T'    B11001100
vr3 row_addr<3> 0      PAT    VDD_VAL 0       0p    'T/100' 'T/100'       'T'    B11001100
vr2 row_addr<2> 0      PAT    VDD_VAL 0       0p    'T/100' 'T/100'       'T'    B11001100
vr1 row_addr<1> 0      PAT    VDD_VAL 0       0p    'T/100' 'T/100'       'T'    B11001100
vr0 row_addr<0> 0      PAT    VDD_VAL 0       0p    'T/100' 'T/100'       'T'    B11001100


vc4 col_addr<4> 0      PAT           VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'
       B00110011
vc3 col_addr<3> 0      PAT           VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'
       B00110011
vc2 col_addr<2> 0      PAT           VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'
       B00110011
vc1 col_addr<1> 0      PAT           VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'
       B00110011
vc0 col_addr<0> 0      PAT           VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'
       B00110011


vb4 block_addr<4> 0    PAT    VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb3 block_addr<3> 0    PAT    VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb2 block_addr<2> 0    PAT    VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb1 block_addr<1> 0    PAT    VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb0 block_addr<0> 0    PAT    VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000

**      n+     n-    type     vhi   vlo  td      tr        tf     pw       per
vrc row_dec_en 0       PULSE  VDD_VAL 0        0     'T/100' 'T/100' '49*T/100'  'T'
*vcc col_dec_en 0      PULSE  VDD_VAL 0        0     'T/100' 'T/100' '49*T/100'  'T'
*vbc block_dec_en 0    PULSE  VDD_VAL 0        0     'T/100' 'T/100' '49*T/100'  'T'

RR_C row_dec_en col_dec_en 0
RR_B row_dec_en block_dec_en 0

*************************CLOCK SOURCES WITH IDEAL DELAY******************************
vck3 lvl3_clk 0        PULSE  VDD_VAL 0 'D2+D1+D0' 'T/100' 'T/100' '49*T/100'  'T'
vck2 lvl2_clk 0        PULSE  VDD_VAL 0  'D1+D0'     'T/100' 'T/100' '49*T/100'  'T'
vck1 lvl1_clk 0        PULSE  VDD_VAL 0    'D0'      'T/100' 'T/100' '49*T/100'  'T'

.END
```

APPENDIX B. HSPICE NETLIST USED FOR 7NM FINFET BASED MEM.A

```
.lib 'C:\Users\SeyedAlireza\Documents\HSPICE\PTM-MG\models' ptm7hp

.PARAM L1 = lg
.PARAM VDD_VAL=vdd
.PARAM T=142p

.GLOBAL vdd!


.subckt D_Latch_1_7nm clk d q

x1      clk_i        clk          0       0       nfet          L='L1' nfin=1
x2      clk_i        clk          vdd!    vdd!    pfet          L='L1' nfin=2

xNMOS   d            clk          Q_unbuff 0      nfet          L='L1' nfin=1
xPMOS   d            clk_i        Q_unbuff vdd!   pfet          L='L1' nfin=2

x10     q_i          Q_unbuff     vdd!    vdd!    pfet          L='L1' nfin=2
x5      q_i          Q_unbuff     0       0       nfet          L='L1' nfin=1

x9      Q_unbuff     q_i          vdd!    vdd!    pfet          L='L1' nfin=2
x8      Q_unbuff     q_i          0       0       nfet          L='L1' nfin=1

x7      q            q_i          vdd!    vdd!    pfet          L='L1' nfin=2
x6      q            q_i          0       0       nfet          L='L1' nfin=1

.ends D_Latch_1_7nm

.subckt inverter_old_7nm in out
x0 out in 0    0       nfet L='L1' nfin=1
x1 out in vdd! vdd! pfet L='L1' nfin=2
.ends inverter_old_7nm


.subckt nand_3_7nm a b c z
x3 z c net13 0               nfet    L='L1' nfin=3
x5 net12 a 0 0               nfet    L='L1' nfin=3
x4 net13 b net12 0    nfet    L='L1' nfin=3
x2 z c vdd! vdd!      pfet    L='L1' nfin=2
x1 z b vdd! vdd!      pfet    L='L1' nfin=2
x0 z a vdd! vdd!      pfet    L='L1' nfin=2
.ends nand_3_7nm


.subckt dec2_4_7nm en in0 in1 y0 y1 y2 y3
xi14 y3_i y3   inverter_old_7nm
xi13 y2_i y2   inverter_old_7nm
xi12 y1_i y1   inverter_old_7nm
xi11 y0_i y0   inverter_old_7nm
xi0 in0  in0_i        inverter_old_7nm
xi1 in1  in1_i        inverter_old_7nm
xi9 in0   in1   en y3_i       nand_3_7nm
xi8 in0_i in1   en y2_i       nand_3_7nm
xi7 in0   in1_i en y1_i       nand_3_7nm
xi6 in0_i in1_i en y0_i       nand_3_7nm
.ends dec2_4_7nm


.subckt nand_2_7nm a b z
x2 z a        net9   0      nfet   L='L1' nfin=2
x3 net9 b     0             0      nfet   L='L1' nfin=2
x1 z a        vdd! vdd!     pfet   L='L1' nfin=2
x0 z b        vdd! vdd!     pfet   L='L1' nfin=2
.ends nand_2_7nm

.subckt dec3_8_En_7nm a0 a1 a2 en y0 y1 y2 y3 y4 y5 y6 y7
xi69   a0 a0_i inverter_old_7nm
xi68   a1 a1_i inverter_old_7nm
```

```
xi67   a2 a2_i inverter_old_7nm
xi93   a2_i a1_i a0_i z0_i nand_3_7nm
xi92   a2_i a1_i a0   z1_i nand_3_7nm
xi91   a2_i a1   a0_i z2_i nand_3_7nm
xi90   a2_i a1   a0   z3_i nand_3_7nm
xi89   a2       a1_i a0_i z4_i nand_3_7nm
xi88   a2       a1_i a0   z5_i nand_3_7nm
xi87   a2       a1   a0_i z6_i nand_3_7nm
xi86   a2       a1   a0   z7_i nand_3_7nm
xi76   z0_i z0 inverter_old_7nm
xi77   z1_i z1 inverter_old_7nm
xi75   z2_i z2 inverter_old_7nm
xi74   z3_i z3 inverter_old_7nm
xi73   z4_i z4 inverter_old_7nm
xi72   z5_i z5 inverter_old_7nm
xi71   z6_i z6 inverter_old_7nm
xi70   z7_i z7 inverter_old_7nm
xi101  z0 en y0_i nand_2_7nm
xi100  z1 en y1_i nand_2_7nm
xi98   z2 en y2_i nand_2_7nm
xi97   z3 en y3_i nand_2_7nm
xi96   z4 en y4_i nand_2_7nm
xi99   z5 en y5_i nand_2_7nm
xi95   z6 en y6_i nand_2_7nm
xi94   z7 en y7_i nand_2_7nm
xi85   y0_i y0 inverter_old_7nm
xi84   y1_i y1 inverter_old_7nm
xi83   y2_i y2 inverter_old_7nm
xi82   y3_i y3 inverter_old_7nm
xi81   y4_i y4 inverter_old_7nm
xi80   y5_i y5 inverter_old_7nm
xi79   y6_i y6 inverter_old_7nm
xi78   y7_i y7 inverter_old_7nm

.ends dec3_8_En_7nm


.subckt dec5_32_FF_7nm a0 a1 a2 a3 a4 en w0 w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12 w13 w14 w15
w16 w17 w18 w19 w20 w21 w22 w23 w24 w25 w26 w27 w28 w29 w30 w31
xLatch4 en a4 a_ff<4> D_Latch_1_7nm
xLatch3 en a3 a_ff<3> D_Latch_1_7nm
xLatch2 en a2 a_ff<2> D_Latch_1_7nm
xLatch1 en a1 a_ff<1> D_Latch_1_7nm
xLatch0 en a0 a_ff<0> D_Latch_1_7nm
xEnDec vdd! a_ff<3> a_ff<4> En00_07 En08_15 En16_23 En24_31 dec2_4_7nm
xDec24_31 a_ff<0> a_ff<1> a_ff<2> En24_31 w24 w25 w26 w27 w28 w29 w30 w31 dec3_8_En_7nm
xDec16_23 a_ff<0> a_ff<1> a_ff<2> En16_23 w16 w17 w18 w19 w20 w21 w22 w23 dec3_8_En_7nm
xDec08_15 a_ff<0> a_ff<1> a_ff<2> En08_15 w8 w9 w10 w11 w12 w13 w14 w15 dec3_8_En_7nm
xDec00_07 a_ff<0> a_ff<1> a_ff<2> En00_07 w0 w1 w2 w3 w4 w5 w6 w7 dec3_8_En_7nm
.ends dec5_32_FF_7nm


.subckt PreCharge_PMOS_7nm precharge_clk rbl<0> rbl<1> rbl<2> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7>
rbl<8> rbl<9> rbl<10> rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19>
rbl<20> rbl<21> rbl<22> rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31>
rbl_i<0> rbl_i<1> rbl_i<2> rbl_i<3> rbl_i<4> rbl_i<5> rbl_i<6> rbl_i<7> rbl_i<8> rbl_i<9>
rbl_i<10> rbl_i<11> rbl_i<12> rbl_i<13> rbl_i<14> rbl_i<15> rbl_i<16> rbl_i<17> rbl_i<18>
rbl_i<19> rbl_i<20> rbl_i<21> rbl_i<22> rbl_i<23> rbl_i<24> rbl_i<25> rbl_i<26> rbl_i<27>
rbl_i<28> rbl_i<29> rbl_i<30> rbl_i<31>
xInv_Arr rbl<0>  rbl<1>  rbl<2>  rbl<3>  rbl<4>  rbl<5>  rbl<6>  rbl<7>  rbl<8>  rbl<9>  rbl<10>
rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19> rbl<20> rbl<21> rbl<22>
rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31> rbl_i<0> rbl_i<1>
rbl_i<2> rbl_i<3> rbl_i<4> rbl_i<5> rbl_i<6> rbl_i<7> rbl_i<8> rbl_i<9> rbl_i<10> rbl_i<11>
rbl_i<12> rbl_i<13> rbl_i<14> rbl_i<15> rbl_i<16> rbl_i<17> rbl_i<18> rbl_i<19> rbl_i<20>
rbl_i<21> rbl_i<22> rbl_i<23> rbl_i<24> rbl_i<25> rbl_i<26> rbl_i<27> rbl_i<28> rbl_i<29>
rbl_i<30> rbl_i<31> Inverter_Array
x00 rbl<0> precharge_clk vdd! vdd!   pfet  L='L1' nfin=1
x01 rbl<1> precharge_clk vdd! vdd!   pfet  L='L1' nfin=1
x02 rbl<2> precharge_clk vdd! vdd!   pfet  L='L1' nfin=1
x03 rbl<3> precharge_clk vdd! vdd!   pfet  L='L1' nfin=1
x04 rbl<4> precharge_clk vdd! vdd!   pfet  L='L1' nfin=1
```

```
x05 rbl<5> precharge_clk vdd! vdd!    pfet  L='L1' nfin=1
x06 rbl<6> precharge_clk vdd! vdd!    pfet  L='L1' nfin=1
x07 rbl<7> precharge_clk vdd! vdd!    pfet  L='L1' nfin=1
x08 rbl<8> precharge_clk vdd! vdd!    pfet  L='L1' nfin=1
x09 rbl<9> precharge_clk vdd! vdd!    pfet  L='L1' nfin=1
x10 rbl<10> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x11 rbl<11> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x12 rbl<12> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x13 rbl<13> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x14 rbl<14> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x15 rbl<15> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x16 rbl<16> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x17 rbl<17> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x18 rbl<18> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x19 rbl<19> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x20 rbl<20> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x21 rbl<21> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x22 rbl<22> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x23 rbl<23> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x24 rbl<24> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x25 rbl<25> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x26 rbl<26> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x27 rbl<27> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x28 rbl<28> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x29 rbl<29> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x30 rbl<30> precharge_clk vdd! vdd! pfet  L='L1' nfin=1
x31 rbl<31> precharge_clk vdd! vdd! pfet  L='L1' nfin=1

.ends PreCharge_PMOS_7nm

.subckt SRAM_8T_7nm q rbl rwl wbl wblb wwl

x2 net28 q vdd! vdd! pfet     L='L1' nfin=1
x3 net28 q 0 0 nfet           L='L1' nfin=2

x0 q net28 vdd! vdd! pfet     L='L1' nfin=1
x1 q net28 0 0 nfet           L='L1' nfin=2

x5 wbl wwl q 0 nfet           L='L1' nfin=2
x4 wblb wwl net28  0 nfet     L='L1' nfin=2

x6 net34 q 0 0 nfet           L='L1' nfin=2
x7 rbl rwl net34 0 nfet       L='L1' nfin=2

.ends SRAM_8T_7nm

.subckt SRAM_32x32_7nm rbl<0> rbl<1> rbl<2> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7> rbl<8> rbl<9>
rbl<10> rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19> rbl<20> rbl<21>
rbl<22> rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31>
+rwl<0> rwl<1> rwl<2> rwl<3> rwl<4> rwl<5> rwl<6> rwl<7> rwl<8> rwl<9> rwl<10> rwl<11> rwl<12>
rwl<13> rwl<14> rwl<15> rwl<16> rwl<17> rwl<18> rwl<19> rwl<20> rwl<21> rwl<22> rwl<23> rwl<24>
rwl<25> rwl<26> rwl<27> rwl<28> rwl<29> rwl<30> rwl<31>
+wbl<0> wbl<1> wbl<2> wbl<3> wbl<4> wbl<5> wbl<6> wbl<7> wbl<8> wbl<9> wbl<10> wbl<11> wbl<12>
wbl<13> wbl<14> wbl<15> wbl<16> wbl<17> wbl<18> wbl<19> wbl<20> wbl<21> wbl<22> wbl<23> wbl<24>
wbl<25> wbl<26> wbl<27> wbl<28> wbl<29> wbl<30> wbl<31>
+wblb<0> wblb<1> wblb<2> wblb<3> wblb<4> wblb<5> wblb<6> wblb<7> wblb<8> wblb<9> wblb<10>
wblb<11> wblb<12> wblb<13> wblb<14> wblb<15> wblb<16> wblb<17> wblb<18> wblb<19> wblb<20>
wblb<21> wblb<22> wblb<23> wblb<24> wblb<25> wblb<26> wblb<27> wblb<28> wblb<29> wblb<30>
wblb<31>
+wwl<0> wwl<1> wwl<2> wwl<3> wwl<4> wwl<5> wwl<6> wwl<7> wwl<8> wwl<9> wwl<10> wwl<11> wwl<12>
wwl<13> wwl<14> wwl<15> wwl<16> wwl<17> wwl<18> wwl<19> wwl<20> wwl<21> wwl<22> wwl<23> wwl<24>
wwl<25> wwl<26> wwl<27> wwl<28> wwl<29> wwl<30> wwl<31>
x31x31  q31x31  rbl<31> rwl<31> wbl<31> wblb<31> wwl<31> SRAM_8T_7nm
x31x30  q31x30  rbl<31> rwl<30> wbl<31> wblb<31> wwl<30> SRAM_8T_7nm
x31x29  q31x29  rbl<31> rwl<29> wbl<31> wblb<31> wwl<29> SRAM_8T_7nm
x31x28  q31x28  rbl<31> rwl<28> wbl<31> wblb<31> wwl<28> SRAM_8T_7nm
**
**
** ALL OF THE 32x32 SRAM CELLS ARE HERE.
** THEY ARE DELETED TO MAKE THE APPENDIX SHORTER
**
```

```
x00x03  q00x03  rbl<0> rwl<3> wbl<0> wblb<0> wwl<3> SRAM_8T_7nm
x00x02  q00x02  rbl<0> rwl<2> wbl<0> wblb<0> wwl<2> SRAM_8T_7nm
x00x01  q00x01  rbl<0> rwl<1> wbl<0> wblb<0> wwl<1> SRAM_8T_7nm
x00x00  q00x00  rbl<0> rwl<0> wbl<0> wblb<0> wwl<0> SRAM_8T_7nm

.ends SRAM_32x32_7nm

.subckt AO_2x32_Dynamic_7nm a<0> a<1> a<2> a<3> a<4> a<5> a<6> a<7> a<8> a<9> a<10> a<11> a<12>
a<13> a<14> a<15> a<16> a<17> a<18> a<19> a<20> a<21> a<22> a<23> a<24> a<25> a<26> a<27> a<28>
a<29> a<30> a<31> b<0> b<1> b<2> b<3> b<4> b<5> b<6> b<7> b<8> b<9> b<10> b<11> b<12> b<13> b<14>
b<15> b<16> b<17> b<18> b<19> b<20> b<21> b<22> b<23> b<24> b<25> b<26> b<27> b<28> b<29> b<30>
b<31> clk output output_inverted
x1 output a<31> net31 0       nfet L='L1' nfin=5
x2 net31 b<31> 0 0            nfet L='L1' nfin=5
x3 output a<30> net30 0       nfet L='L1' nfin=5
x4 net30 b<30> 0 0            nfet L='L1' nfin=5
x5 output a<29> net29 0       nfet L='L1' nfin=5
x6 net29 b<29> 0 0            nfet L='L1' nfin=5
x7 output a<28> net28 0       nfet L='L1' nfin=5
x8 net28 b<28> 0 0            nfet L='L1' nfin=5
x9 output a<27> net27 0       nfet L='L1' nfin=5
x10 net27 b<27> 0 0           nfet L='L1' nfin=5
x11 output a<26> net26 0      nfet L='L1' nfin=5
x12 net26 b<26> 0 0           nfet L='L1' nfin=5
x13 output a<25> net25 0      nfet L='L1' nfin=5
x14 net25 b<25> 0 0           nfet L='L1' nfin=5
x15 output a<24> net24 0      nfet L='L1' nfin=5
x16 net24 b<24> 0 0           nfet L='L1' nfin=5
x17 output a<23> net23 0      nfet L='L1' nfin=5
x18 net23 b<23> 0 0           nfet L='L1' nfin=5
x19 output a<22> net22 0      nfet L='L1' nfin=5
x20 net22 b<22> 0 0           nfet L='L1' nfin=5
x21 output a<21> net21 0      nfet L='L1' nfin=5
x22 net21 b<21> 0 0           nfet L='L1' nfin=5
x23 output a<20> net20 0      nfet L='L1' nfin=5
x24 net20 b<20> 0 0           nfet L='L1' nfin=5
x25 output a<19> net19 0      nfet L='L1' nfin=5
x26 net19 b<19> 0 0           nfet L='L1' nfin=5
x27 output a<18> net18 0      nfet L='L1' nfin=5
x28 net18 b<18> 0 0           nfet L='L1' nfin=5
x29 output a<17> net17 0      nfet L='L1' nfin=5
x30 net17 b<17> 0 0           nfet L='L1' nfin=5
x31 output a<16> net16 0      nfet L='L1' nfin=5
x32 net16 b<16> 0 0           nfet L='L1' nfin=5
x33 output a<15> net15 0      nfet L='L1' nfin=5
x34 net15 b<15> 0 0           nfet L='L1' nfin=5
x35 output a<14> net14 0      nfet L='L1' nfin=5
x36 net14 b<14> 0 0           nfet L='L1' nfin=5
x37 net13 b<13> 0 0           nfet L='L1' nfin=5
x38 output a<13> net13 0      nfet L='L1' nfin=5
x39 net12 b<12> 0 0           nfet L='L1' nfin=5
x40 output a<12> net12 0      nfet L='L1' nfin=5
x41 net11 b<11> 0 0           nfet L='L1' nfin=5
x42 output a<11> net11 0      nfet L='L1' nfin=5
x43 net10 b<10> 0 0           nfet L='L1' nfin=5
x44 output a<10> net10 0      nfet L='L1' nfin=5
x45 net09 b<9> 0 0            nfet L='L1' nfin=5
x46 output a<9> net09 0       nfet L='L1' nfin=5
x47 net08 b<8> 0 0            nfet L='L1' nfin=5
x48 output a<8> net08 0       nfet L='L1' nfin=5
x49 net07 b<7> 0 0            nfet L='L1' nfin=5
x50 output a<7> net07 0       nfet L='L1' nfin=5
x51 net06 b<6> 0 0            nfet L='L1' nfin=5
x52 output a<6> net06 0       nfet L='L1' nfin=5
x53 net05 b<5> 0 0            nfet L='L1' nfin=5
x54 output a<5> net05 0       nfet L='L1' nfin=5
x55 net04 b<4> 0 0            nfet L='L1' nfin=5
x56 output a<4> net04 0       nfet L='L1' nfin=5
x57 net03 b<3> 0 0            nfet L='L1' nfin=5
x58 output a<3> net03 0       nfet L='L1' nfin=5
x59 net02 b<2> 0 0            nfet L='L1' nfin=5
```

```
x60 output a<2> net02 0       nfet L='L1' nfin=5
x61 net01 b<1> 0 0            nfet L='L1' nfin=5
x62 output a<1> net01 0       nfet L='L1' nfin=5
x64 net00 b<0> 0 0            nfet L='L1' nfin=5
x63 output a<0> net00 0       nfet L='L1' nfin=5
x0 output clk vdd! vdd!       pfet L='L1' nfin=10

xi1 output_inverted output vdd! vdd! pfet L='L1' nfin=2
xi0 output_inverted output 0 0 nfet L='L1' nfin=1

.ends AO_2x32_Dynamic_7nm


.subckt dec5_32_7nm a0 a1 a2 a3 a4 en w0 w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12 w13 w14 w15 w16
w17 w18 w19 w20 w21 w22 w23 w24 w25 w26 w27 w28 w29 w30 w31
xi12 a0 a1 a2 y3 w24 w25 w26 w27 w28 w29 w30 w31 dec3_8_En_7nm
xi11 a0 a1 a2 y2 w16 w17 w18 w19 w20 w21 w22 w23 dec3_8_En_7nm
xi10 a0 a1 a2 y1 w8 w9 w10 w11 w12 w13 w14 w15 dec3_8_En_7nm
xi9 a0 a1 a2 y0 w0 w1 w2 w3 w4 w5 w6 w7 dec3_8_En_7nm
xi4 en a3 a4 y0 y1 y2 y3 dec2_4_7nm
.ends dec5_32_7nm


.subckt Inverter_Array rbl<0>  rbl<1>  rbl<2>  rbl<3>  rbl<4>  rbl<5>  rbl<6>  rbl<7>  rbl<8>
rbl<9>  rbl<10> rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19> rbl<20>
rbl<21> rbl<22> rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31> rbl_i<0>
rbl_i<1> rbl_i<2> rbl_i<3> rbl_i<4> rbl_i<5> rbl_i<6> rbl_i<7> rbl_i<8> rbl_i<9> rbl_i<10>
rbl_i<11> rbl_i<12> rbl_i<13> rbl_i<14> rbl_i<15> rbl_i<16> rbl_i<17> rbl_i<18> rbl_i<19>
rbl_i<20> rbl_i<21> rbl_i<22> rbl_i<23> rbl_i<24> rbl_i<25> rbl_i<26> rbl_i<27> rbl_i<28>
rbl_i<29> rbl_i<30> rbl_i<31>
xi39 rbl<0>  rbl_i<0>  inverter_old_7nm
xi38 rbl<1>  rbl_i<1>  inverter_old_7nm
xi37 rbl<2>  rbl_i<2>  inverter_old_7nm
xi36 rbl<3>  rbl_i<3>  inverter_old_7nm
xi35 rbl<4>  rbl_i<4>  inverter_old_7nm
xi34 rbl<5>  rbl_i<5>  inverter_old_7nm
xi33 rbl<6>  rbl_i<6>  inverter_old_7nm
xi32 rbl<7>  rbl_i<7>  inverter_old_7nm
xi31 rbl<8>  rbl_i<8>  inverter_old_7nm
xi30 rbl<9>  rbl_i<9>  inverter_old_7nm
xi29 rbl<10> rbl_i<10> inverter_old_7nm
xi28 rbl<11> rbl_i<11> inverter_old_7nm
xi27 rbl<12> rbl_i<12> inverter_old_7nm
xi26 rbl<13> rbl_i<13> inverter_old_7nm
xi25 rbl<14> rbl_i<14> inverter_old_7nm
xi24 rbl<15> rbl_i<15> inverter_old_7nm
xi23 rbl<16> rbl_i<16> inverter_old_7nm
xi22 rbl<17> rbl_i<17> inverter_old_7nm
xi21 rbl<18> rbl_i<18> inverter_old_7nm
xi20 rbl<19> rbl_i<19> inverter_old_7nm
xi19 rbl<20> rbl_i<20> inverter_old_7nm
xi18 rbl<21> rbl_i<21> inverter_old_7nm
xi17 rbl<22> rbl_i<22> inverter_old_7nm
xi16 rbl<23> rbl_i<23> inverter_old_7nm
xi15 rbl<24> rbl_i<24> inverter_old_7nm
xi14 rbl<25> rbl_i<25> inverter_old_7nm
xi13 rbl<26> rbl_i<26> inverter_old_7nm
xi12 rbl<27> rbl_i<27> inverter_old_7nm
xi11 rbl<28> rbl_i<28> inverter_old_7nm
xi10 rbl<29> rbl_i<29> inverter_old_7nm
xi8  rbl<30> rbl_i<30> inverter_old_7nm
xi9  rbl<31> rbl_i<31> inverter_old_7nm
.ends Inverter_Array



.subckt SRAM_32x32_Block_NewDec_7nm col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4>
col_dec_en lvl1_clk lvl2_bitline lvl2_clk row_addr<0> row_addr<1> row_addr<2> row_addr<3>
row_addr<4> row_dec_en
xi1 lvl1_clk rbl<0> rbl<1> rbl<2> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7> rbl<8> rbl<9> rbl<10>
rbl<11> rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19> rbl<20> rbl<21> rbl<22>
```

```
rbl<23> rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31> rbl_i<0> rbl_i<1>
rbl_i<2> rbl_i<3> rbl_i<4> rbl_i<5> rbl_i<6> rbl_i<7> rbl_i<8> rbl_i<9> rbl_i<10> rbl_i<11>
rbl_i<12> rbl_i<13> rbl_i<14> rbl_i<15> rbl_i<16> rbl_i<17> rbl_i<18> rbl_i<19> rbl_i<20>
rbl_i<21> rbl_i<22> rbl_i<23> rbl_i<24> rbl_i<25> rbl_i<26> rbl_i<27> rbl_i<28> rbl_i<29>
rbl_i<30> rbl_i<31> PreCharge_PMOS_7nm
xSRAM_Cells rbl<0> rbl<1> rbl<2> rbl<3> rbl<4> rbl<5> rbl<6> rbl<7> rbl<8> rbl<9> rbl<10> rbl<11>
rbl<12> rbl<13> rbl<14> rbl<15> rbl<16> rbl<17> rbl<18> rbl<19> rbl<20> rbl<21> rbl<22> rbl<23>
rbl<24> rbl<25> rbl<26> rbl<27> rbl<28> rbl<29> rbl<30> rbl<31>
+rwl<0> rwl<1> rwl<2> rwl<3> rwl<4> rwl<5> rwl<6> rwl<7> rwl<8> rwl<9> rwl<10> rwl<11> rwl<12>
rwl<13> rwl<14> rwl<15> rwl<16> rwl<17> rwl<18> rwl<19> rwl<20> rwl<21> rwl<22> rwl<23> rwl<24>
rwl<25> rwl<26> rwl<27> rwl<28> rwl<29> rwl<30> rwl<31>
+wbl<0> wbl<1> wbl<2> wbl<3> wbl<4> wbl<5> wbl<6> wbl<7> wbl<8> wbl<9> wbl<10> wbl<11> wbl<12>
wbl<13> wbl<14> wbl<15> wbl<16> wbl<17> wbl<18> wbl<19> wbl<20> wbl<21> wbl<22> wbl<23> wbl<24>
wbl<25> wbl<26> wbl<27> wbl<28> wbl<29> wbl<30> wbl<31>
+wblb<0> wblb<1> wblb<2> wblb<3> wblb<4> wblb<5> wblb<6> wblb<7> wblb<8> wblb<9> wblb<10>
wblb<11> wblb<12> wblb<13> wblb<14> wblb<15> wblb<16> wblb<17> wblb<18> wblb<19> wblb<20>
wblb<21> wblb<22> wblb<23> wblb<24> wblb<25> wblb<26> wblb<27> wblb<28> wblb<29> wblb<30>
wblb<31>
+0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 SRAM_32x32_7nm
xDy_Gate rbl_i<0>  rbl_i<1>  rbl_i<2>  rbl_i<3>  rbl_i<4>  rbl_i<5>  rbl_i<6>  rbl_i<7>  rbl_i<8>
rbl_i<9>  rbl_i<10> rbl_i<11> rbl_i<12> rbl_i<13> rbl_i<14> rbl_i<15> rbl_i<16> rbl_i<17>
rbl_i<18> rbl_i<19> rbl_i<20> rbl_i<21> rbl_i<22> rbl_i<23> rbl_i<24> rbl_i<25> rbl_i<26>
rbl_i<27> rbl_i<28> rbl_i<29> rbl_i<30> rbl_i<31>  col<0> col<1> col<2> col<3> col<4> col<5>
col<6> col<7> col<8> col<9> col<10> col<11> col<12> col<13> col<14> col<15> col<16> col<17>
col<18> col<19> col<20> col<21> col<22> col<23> col<24> col<25> col<26> col<27> col<28> col<29>
col<30> col<31> lvl2_clk lvl2_Dy_Node lvl2_bitline AO_2x32_Dynamic_7nm_Keeper
xRow_Dec row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en rwl<0> rwl<1>
rwl<2> rwl<3> rwl<4> rwl<5> rwl<6> rwl<7> rwl<8> rwl<9> rwl<10> rwl<11> rwl<12> rwl<13> rwl<14>
rwl<15> rwl<16> rwl<17> rwl<18> rwl<19> rwl<20> rwl<21> rwl<22> rwl<23> rwl<24> rwl<25> rwl<26>
rwl<27> rwl<28> rwl<29> rwl<30> rwl<31> dec5_32_7nm
xCol_Dec col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en_i col<0> col<1>
col<2> col<3> col<4> col<5> col<6> col<7> col<8> col<9> col<10> col<11> col<12> col<13> col<14>
col<15> col<16> col<17> col<18> col<19> col<20> col<21> col<22> col<23> col<24> col<25> col<26>
col<27> col<28> col<29> col<30> col<31>  dec5_32_FF_7nm
xCol_CLK_inv col_dec_en col_dec_en_i inverter_old_7nm
.ends SRAM_32x32_Block_NewDec_7nm

.subckt SRAM_32x32x32_Simplified_NewDec_7nm block_addr<0> block_addr<1> block_addr<2>
block_addr<3> block_addr<4> block_dec_en col_addr<0> col_addr<1> col_addr<2> col_addr<3>
col_addr<4> col_dec_en clk1 clk2 lvl3_Dy_Node lvl3_bitline lvl3_clk row_addr<0> row_addr<1>
row_addr<2> row_addr<3> row_addr<4> row_dec_en
xBlo_Dec block_addr<0> block_addr<1> block_addr<2> block_addr<3> block_addr<4> block_dec_en_i
blo<0> blo<1> blo<2> blo<3> blo<4> blo<5> blo<6> blo<7> blo<8> blo<9> blo<10> blo<11> blo<12>
blo<13> blo<14> blo<15> blo<16> blo<17> blo<18> blo<19> blo<20> blo<21> blo<22> blo<23> blo<24>
blo<25> blo<26> blo<27> blo<28> blo<29> blo<30> blo<31> dec5_32_FF_7nm
xBlo_CLK_inv block_dec_en block_dec_en_i inverter_old_7nm
xSRAM_2Level<00> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<0>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<01> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<1>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<02> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<2>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<03> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<3>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<04> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<4>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<05> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<5>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<06> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<6>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<07> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<7>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
```

```
xSRAM_2Level<08> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<8>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<09> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<9>  clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<10> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<10> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<11> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<11> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<12> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<12> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<13> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<13> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<14> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<14> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<15> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<15> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<16> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<16> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<17> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<17> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<18> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<18> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<19> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<19> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<20> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<20> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<21> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<21> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<22> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<22> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<23> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<23> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<24> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<24> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<25> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<25> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<26> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<26> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<27> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<27> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<28> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<28> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<29> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<29> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
xSRAM_2Level<30> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<30> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm
```

```
xSRAM_2Level<31> col_addr<0> col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en clk1
bitline<31> clk2 row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32_Block_NewDec_7nm

xDyGate bitline<0> bitline<1> bitline<2> bitline<3> bitline<4> bitline<5> bitline<6> bitline<7>
bitline<8> bitline<9> bitline<10> bitline<11> bitline<12> bitline<13> bitline<14> bitline<15>
bitline<16> bitline<17> bitline<18> bitline<19> bitline<20> bitline<21> bitline<22> bitline<23>
bitline<24> bitline<25> bitline<26> bitline<27> bitline<28> bitline<29> bitline<30> bitline<31>
blo<0> blo<1> blo<2> blo<3> blo<4> blo<5> blo<6> blo<7> blo<8> blo<9> blo<10> blo<11> blo<12>
blo<13> blo<14> blo<15> blo<16> blo<17> blo<18> blo<19> blo<20> blo<21> blo<22> blo<23> blo<24>
blo<25> blo<26> blo<27> blo<28> blo<29> blo<30> blo<31> lvl3_clk lvl3_Dy_Node lvl3_bitline
AO_2x32_Dynamic_7nm_Keeper
.ends SRAM_32x32x32_Simplified_NewDec_7nm

xSRAM_3level blo_addr<0> blo_addr<1> blo_addr<2> blo_addr<3> blo_addr<4> blo_dec_en col_addr<0>
col_addr<1> col_addr<2> col_addr<3> col_addr<4> col_dec_en lvl1_clk lvl2_clk lvl3_Dy_node
lvl3_Bitline lvl3_clk row_addr<0> row_addr<1> row_addr<2> row_addr<3> row_addr<4> row_dec_en
SRAM_32x32x32_Simplified_NewDec_7nm

v00 vdd! 0 DC=VDD_VAL
**     n+      n-   type      vhi   vlo  td           tr     tf       tsample
vr4 row_addr<4> 0      PAT     VDD_VAL 0       0p     'T/100' 'T/100'       'T'    B1100
vr3 row_addr<3> 0      PAT     VDD_VAL 0       0p     'T/100' 'T/100'       'T'    B1100
vr2 row_addr<2> 0      PAT     VDD_VAL 0       0p     'T/100' 'T/100'       'T'    B1100
vr1 row_addr<1> 0      PAT     VDD_VAL 0       0p     'T/100' 'T/100'       'T'    B1100
vr0 row_addr<0> 0      PAT     VDD_VAL 0       0p     'T/100' 'T/100'       'T'    B1100

vc4 col_addr<4> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0011
vc3 col_addr<3> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0011
vc2 col_addr<2> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0011
vc1 col_addr<1> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0011
vc0 col_addr<0> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0011

vb4 blo_addr<4> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb3 blo_addr<3> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb2 blo_addr<2> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb1 blo_addr<1> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000
vb0 blo_addr<0> 0      PAT     VDD_VAL 0      'T/10'  'T/100' 'T/100'       'T'    B0000

**      n+     n-    type     vhi   vlo     td      tr        tf     pw       per
vrc row_dec_en 0      PULSE   VDD_VAL 0      0       'T/100' 'T/100' '49*T/100'  'T'
vcc col_dec_en 0      PULSE   VDD_VAL 0      0       'T/100' 'T/100' '49*T/100'  'T'
vbc blo_dec_en 0      PULSE   VDD_VAL 0      0       'T/100' 'T/100' '49*T/100'  'T'

vck3 lvl3_clk 0       PULSE   VDD_VAL 0   0e-12      'T/100' 'T/100' '49*T/100'  'T'
vck2 lvl2_clk 0       PULSE   VDD_VAL 0   0e-12      'T/100' 'T/100' '49*T/100'  'T'
vck1 lvl1_clk 0       PULSE   VDD_VAL 0   0e-12      'T/100' 'T/100' '49*T/100'  'T'

.OPTION LIST NODE POST
.TRAN 0.1p '10*T' START=0

.END
```