Multi-Tier 3D SRAM Module Design: Targeting Bit-Line and Word-Line Folding

Aditya S. Iyer
Georgia Institute of Technology
Atlanta, Georgia, USA
aditya.iyer@gatech.edu

Daehyun Kim
Georgia Institute of Technology
Atlanta, Georgia, USA
daehyun.kim@gatech.edu

Saibal Mukhopadhyay
Georgia Institute of Technology
Atlanta, Georgia, USA
saibal.mukhopadhyay@ece.gatech.edu

Sung Kyu Lim
Georgia Institute of Technology
Atlanta, Georgia, USA
limsk@ece.gatech.edu

ABSTRACT

This study presents a novel approach to address the challenges of scaling limitations and high read/write latencies inherent in conventional 2D SRAM designs through the development of a 15nm FinFET-based standalone 3D SRAM Array. Utilizing Monolithic Intertier Vias, our two-tiered implementation effectively mitigates these limitations along both X and Y axes. We introduce two distinct design methodologies for 3D SRAM arrays - Wordline and Bitline folding. Through post-layout simulations conducted on arrays with capacities of 2kB (256WLx64BL) and 8kB (512WLx128BL), our 3D designs exhibit superior performance metrics. Notably, the 3D Wordline-Folded design achieves a remarkable 57.54% average reduction in footprint compared to the 2D baseline across both array sizes. Furthermore, the 3D Bitline-Folded configuration demonstrates consistent superiority in speed, with an average read latency improvement of 17.16% and a notable 54.2% enhancement in write latency. Conversely, the 3D Wordline-Folded array emerges as the most energy-efficient option, boasting an average reduction of 15.3% in read energy and over 21% in write energy compared to the 2D baseline.

1 INTRODUCTION

As the semiconductor industry progresses further into the single-digit nano-regime, SRAM technology lags behind. The proportion of SRAM modules in modern processors is only growing larger, while other components continue to shrink as the process technology advances. The transition to FinFETs represents a partial resolution to the challenge at hand. Although it enables a heightened memory cell density, resulting in expanded on-chip memory, this scaling capability is confined to two dimensions up to a certain threshold. Beyond this threshold, limitations arise due to the area and power overhead imposed by the peripheral logic. An excessively enlarged cell array in either the X or Y dimension confronts the detrimental effects of wire parasitics, inherent in lengthy bitlines or wordlines.

Recent attention has shifted towards 3D memory designs for potential performance gains. While architectural simulations in [1] illustrate the benefits, there's a lack of exploration into physical design aspects. Works like [2]-[3] delve into ultra-fine-grained memory design by partitioning SRAM bitcell transistors into two tiers, utilizing the Z-dimension for compact bitcell structures. Yet, they often overlook peripheral logic design considerations, leaving performance implications unanswered.

AMD's 3D V-Cache, introduced in [4], is a notable advancement, addressing the challenge of scaling memory vertically. However, it's designed as a last-level cache, offering benefits in scenarios with frequent DRAM accesses but effectiveness depends on access patterns and cache hierarchy, as discussed.

This paper introduces a novel design methodology for 3D SRAM Array Design, leveraging the 15nm FinFET open-source Process Design Kit (PDK) developed by NCSU [5] in conjunction with the standard cell library from Nangate [6]. The deliberate choice of utilizing an open-source PDK reflects on the broader applicability of the methodologies proposed herein, transcending specific technologies to encompass a wide array of commercial platforms. The key contributions of this research are as follows:

- Development of a comprehensive methodology for SRAM bitcell and peripheral design driven by layout constraints and simulation analysis.
- Addressing challenges posed by extended wordlines through the introduction of a folded 3D SRAM array design, which partitions the array along the wordline into a structured 2-tier configuration while consolidating peripherals within a single tier. Exploration of a similar architectural approach employing bitline folding.
- Demonstration, through physical layout and post-layout simulations, of the performance, power, and area benefits of the proposed 3D designs over a baseline 2D design. Utilization of a scaled RC model of an inter-tier via with a pitch of 0.12 μm to connect the two tiers.

To overcome scaling limitations inherent in 2D designs, this paper demonstrates that our 3D designs not only enhance performance, power efficiency, and area utilization compared to the 2D baseline, but also incorporate extensive peripheral reuse. Additionally, a rigorous comparison will be conducted with industry-standard state-of-the-art memory designs and existing literature to validate the efficacy and quality of our 2D designs.

2 DESIGN METHODOLOGY

2.1 Process Design Kit Used

Our 2D and 3D designs and simulations are performed using the 15nm FinFET open-source PDK [5]. This PDK comes with BSIM-CMG transistor models that were finely calibrated using the 14nm

industry-standard FinFET PDK. The fins and gate form the Front-End-Of-Line (FEOL) layers. The PDK also comes with Middle-Of-Line (MOL) layers that can be used for internal routing to reduce the number of contacts and Back-End-Of-Line (BEOL) metal layers that can be used for external routing. The two tiers are connected by Monolithic Intertier Vias (MIVs) of 0.12 μm pitch. The BEOL layers are divided into M1, followed by 5 intermediate metal layers, 5 semi-global metal layers and 2 global metal layers to give a total of 13 metal layers in the BEOL stack.

The number of fins and fin pitch play a huge role in reliability of the design. Unlike planar CMOS, where the "width" of the transistor defines how much current can flow through it, in FinFETs, an integer value defining the number of fins determines how much current can flow through them. As explained in [7], the driving current increases in pFETs and the threshold voltage degradation in nFETS decreases with a higher number of fins. While single-fin devices can offer very high layout density and marginally lower parasitic capacitance, it is recommended to use a higher number of fins, particularly in standard cell layouts. The Nangate 15nm Open Cell Library [6] comes with standard cells containing 8 fins. For custom cells, the process of deciding the number of fins is based on empirical analysis which will be explained in the following sections.

2.2 Overview of the Design Flow

Figure 1 illustrates the flowchart detailing our 2D and 3D design methodology. While the steps for both dimensions share similarities, notable differences arise in schematic and layout design. For the 3D MIV, a distributed RC model facilitates connecting the two tiers to form a unified schematic. Top and bottom tier layouts are individually created, followed by parasitic extraction and subsequent post-layout simulations using the extracted netlist. To assess the performance of our arrays, four key metrics—read/write latency and read/write energy —are measured from the post-layout simulations.

In the physical design phase, Synopsys Design Compiler generates the gate-level netlist. Block-level place-and-route (PnR) is conducted on the array .lef, exported from Cadence Virtuoso's in-built abstract generator, in Cadence Innovus to produce the timing .lib file for our memory macros. This file is then utilized in Synopsys Library Compiler to generate the hardmacro .db file. The entire Place-and-Route process, including stages such as placeopt, clockopt, and routeopt, is integral to our design flow, although omitted from Figure 1 for brevity.

Our design flow primarily operates in a 2D paradigm; the place-and-route process treats our 3D memory macros as conventional 2D macros. Leveraging the fact that all external I/O connections for our 3D memory macros reside in the bottom tier alone, we capitalize on the full capabilities of 2D PnR tools.

Post-layout simulations were performed using Cadence Spectre. Siemens Calibre was used to perform the parasitic extraction and design verification of our layouts.

To create the layouts of the digital modules such as the address decoders, we create custom verilog scripts that are first synthesized to get gate-level netlists. Post this, we create custom placement and routing scripts, that automatically decide the number of rows of in the layout based on the highest layout density achievable. The pin

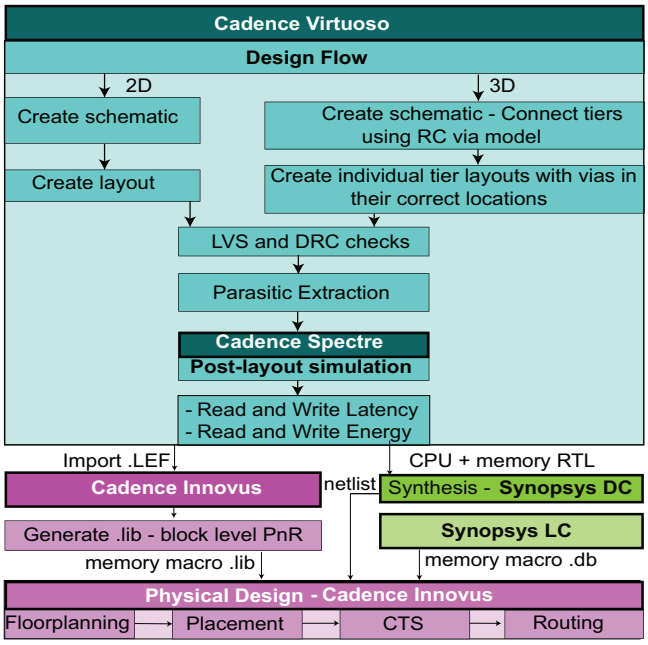

Figure 1: 2D and 3D SRAM Array Design Methodology.

spacing is decided based on the pitch of the bitlines and wordlines, to ensure that the decoders could directly interface with the SRAM cell array without any additional routing. Other digital modules are created using a similar methodology.

3 COMPONENT DESIGN

Creating custom layouts poses a formidable challenge owing to the intricate arrangement and interconnection demands among diverse components. To tackle this challenge head-on, our methodology is geared towards achieving harmonized pin pitches across all constituent elements, spanning both tiers and peripheral logic. This deliberate standardization facilitates seamless adjacency of these elements, thereby facilitating automatic connections and significantly curtailing the need for intricate routing.

The subsequent sections delineate the methodology employed for layout creation of the 6T SRAM bitcell alongside the pivotal peripherals utilized in our designs. For brevity's sake, certain standard designs such as multiplexers and registers will be cursorily mentioned without delving into exhaustive detail.

3.1 6T SRAM Bitcell

The ubiquitous and conventional 6T SRAM bitcell forms the cornerstone of our work due to its standardized design and reliable operation. In contrast to planar CMOS, where transistor width plays a pivotal role, FinFETs operate on the principle that the number of fins in each transistor determines access time and energy consumption. To ascertain the optimal choice for our work, experiments were conducted on 1:1:1, 1:2:2, and 2:2:2 bitcells, representing the number of fins in the inverter PFET, inverter NFET, and access transistors, respectively. Performance characterization based on read and write Static Noise Margin was employed to evaluate these cell

Table 1: Parameters of our SRAM bitcell

Parameter	Value
VDD (mV)	800
Height (μm)	0.768
Width (μm)	0.192
Area (μm^2)	0.147
Read Static Noise Margin (mV)	131
Write Margin (mV)	450

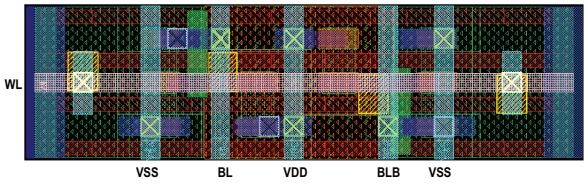

Figure 2: Layout of our 6T Bitcell in 15nm FinFET [5] (rotated anti-clockwise)

variants, with the 2:2:2 bitcell demonstrating superior performance among the three.

In devising the layout of the SRAM bitcell, design flexibility is limited in FinFETs due to the strict vertical orientation of the poly (gate) layer. This stands in stark contrast to planar CMOS PDKs, where gate orientation can be either horizontal or vertical, affording designers the freedom to tailor different bitcell layouts to prioritize density or speed. With FinFETs, owing to the inherent 3D fin structures perpendicular to the gate, gate orientation in both directions is unfeasible. Nonetheless, the layout depicted in Figure 2 adheres to standard thin cell rules following a similar design to the one in [8], achieving exceptional density without compromising speed. Notably, the vertical orientation of the Wordline and horizontal orientation of the bitlines, as depicted in Figure 2, play a pivotal role in the overarching architecture of our SRAM array as well as the design and placement of peripherals.

Table 1 shows the overall design parameters of the SRAM bitcell. The area of 0.147 μm^2 is very reasonable considering the fact that our PDK is open-source. As a reference, the closest commercial FinFET technology node, TSMC 16nm, reports a single port SRAM bitcell area of 0.0907 μm^2 [9].

3.2 Address Decoders

The address decoders, serving both the row and column selection functions, play a pivotal role in determining the activation of wordlines and bitlines in each cycle. To ensure seamless operation, the pitch of the output pins in both row and column decoders must align with the pitch of the wordlines and bitlines. In our design, we implement two distinct decoders for columns and wordlines. The first is a standard k-to-2^k decoder, while the second is a k-to-2^{k-1} decoder equipped with built-in tier selection logic. This tier selection logic utilizes the Most Significant Bit (MSB), i.e., the k^{th} bit of the address, to determine the active tier's wordlines. Consequently,

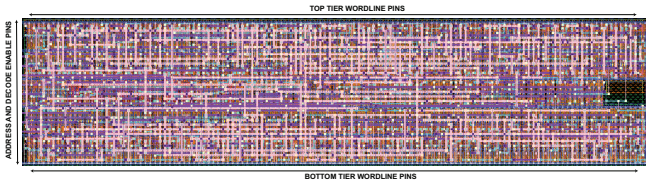

Figure 3: 8-to-128 decoder used in OUR 3D SRAM Arrays, with in-built tier selection logic

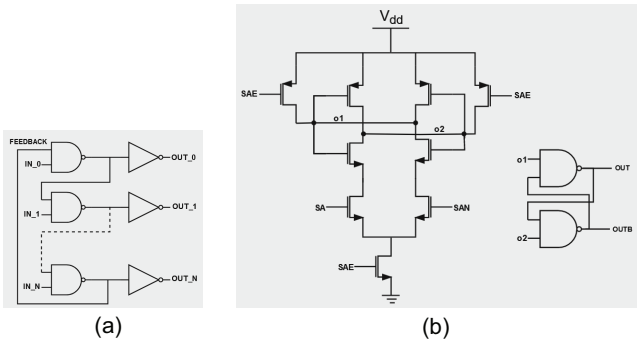

Figure 4: Schematic. (a) Feedback based N-bit wordline driver, (b) current-latch based differential sense amplifier.

two sets of 2^{k-1} output pins are generated. Figure 3 illustrates the layout of the latter type of decoder, wherein 8 address bits decode to either 128 wordlines in the bottom tier (bottom pins) or 128 wordlines in the top tier (top pins).

3.3 Wordline Driver

The wordline driver is an optional circuit that utilizes a feedback loop to drive the values from the inputs to the outputs. A simple schematic of it is shown in Figure 4(a). The feedback loop is used to ensure that the output remains stable and robust under different operating conditions.

3.4 Sense Amplifier

The sense amplifier is one of the most important peripheral modules required in the design of SRAM arrays. Every read operation of the array uses the sense amplifier to interpret the value of the selected bitlines. There are several sense amplifier designs available, each with its pros and cons. We design a current-latch-based differential sense amplifier, with our choice being substantiated by prior investigations detailed in [10], which affirms the superiority of differential sense amplifiers for FinFET-based SRAMs due to enhanced electrostatic integrity. Notably, our sense amplifier design yields a robust sense margin of 50 mV for a VDD of 800mV. The schematic of our sense amplifier is depicted in Figure 4(b). It is imperative to underscore the temporal aspect of sense amplifier operation, which significantly influences the operational frequency of the SRAM array. Upon satisfying the setup time constraints of the address registers, the predominant portion of time expended during the read operation stems from the duration taken by the sense amplifiers to decode the values on the bitlines.

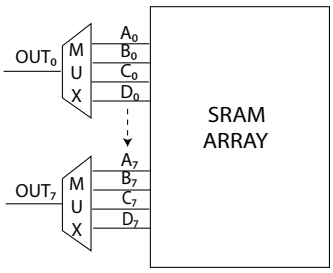

Figure 5: Bitline interleaving shown for an array containing 32 bitlines. Bitlines are divided into 8 groups of 4 bits each where these 4 bits are multiplexed to give a single output, with the top bits of the user-entered address used for the select signals.

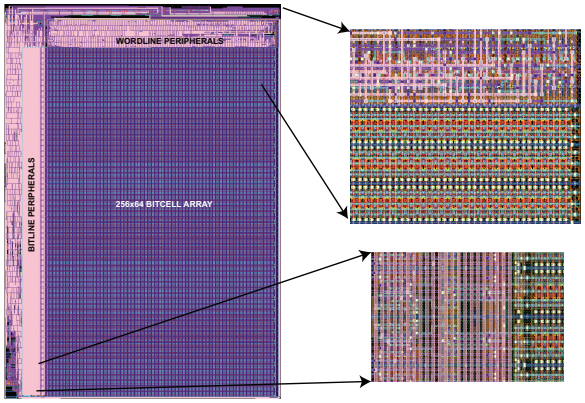

Figure 6: Layout of our 2D 256WL x 64BL SRAM array.

4 BASELINE 2D ARRAY DESIGN

This chapter will discuss the architecture and layout creation of the baseline 2D SRAM array and provide some key insights into the layout creation of 2D arrays before moving on to 3D arrays.

4.1 Architecture of the Array

As depicted in Figure 2, the orientation of the wordline (WL) is vertical, while the bitline (BL) is horizontal. This necessitates the placement of wordline peripherals atop the array and bitline peripherals on the sides. The array architecture, illustrated in Figure 7, features the address decoder and optional wordline driver positioned at the top, facilitating connection to the wordlines. On the left side, the write driver, precharge/select circuit, and sense amplifier for the bitlines are situated. For brevity, discussions regarding bitline multiplexers and registers are omitted.

Control logic orchestrates various functions, including providing addresses to the address decoder, activating precharge, bitline select, sense amplifier enable SAE, as well as managing write data inputs and read data outputs. Operating at an 8-bit precision, each read and write operation targets 8 "interleaved" bits within a single cycle. Consequently, the number of bits in a wordline must be a multiple of 8, and the number of 8-bit words in a wordline determines the additional address bits required for correct word decoding.

Figure 5 offers a visual representation of bitline interleaving in our design, assuming 32 bitlines for simplicity. The 4 8-bit words - A, B, C, and D - from each wordline are interleaved from $A_0B_0C_0D_0$ to $A_7B_7C_7D_7$. Row multiplexers employ two bit select lines to choose one of four bits in each of the 8 row multiplexers, thereby forming the final 8-bit word. This strategy is mirrored in write operations as well.

4.2 2D Array Layout

For the physical design of our arrays, we opt for two array sizes: a 2kB 256WL x 64BL array and an 8kB 512WL x 128BL array. These selections aim to showcase our 3D design methodology across varying macro sizes, encompassing both smaller and larger configurations. Furthermore, the rationale behind choosing these sizes stems from the desire to maintain an equitable aspect ratio for the 2D arrays, a point elucidated in subsequent area analyses. Figure 6 illustrates

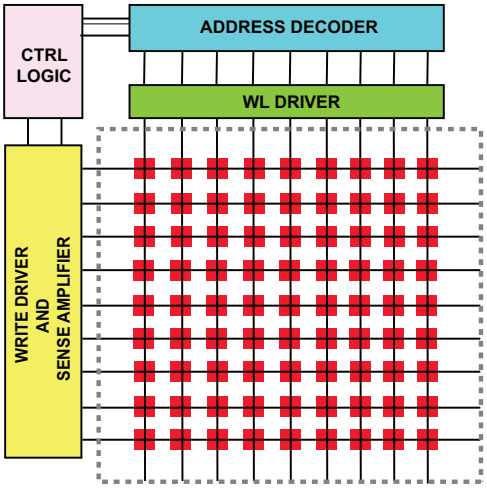

Figure 7: Architecture of our 2D SRAM array.

the layout of the 256WL x 64BL 2D baseline SRAM Array, with the architecture depicted in Figure 7.

Despite the larger size of the bitcell compared to the 16nm counterpart, by horizontally flipping alternate bitcells within a row, we achieve an exceptionally high layout density. This approach effectively reduces the effective area of our SRAM bitcell to a mere 0.1 μm^2. The pitch of the wordlines is set at 128nm, equivalent to two track spacings, a parameter of significance for our forthcoming discussion on 3D array design methodology. To facilitate integration with bitline peripherals, a custom cell is devised to transform the triple power rail SRAM bitcell into the standard dual power rail design. Within our PDK, a standard cell measures 768 nm in height, while the pitch of the wordlines remains at 128nm. This alignment explains our ability to maintain an even aspect ratio of the 2D array despite accommodating four times more wordlines than bitlines.

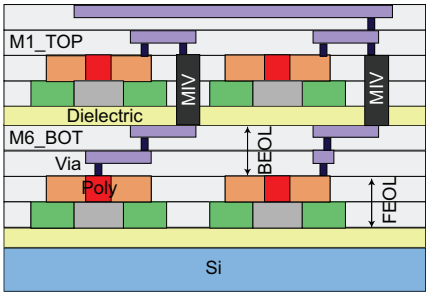

Figure 8: Monolithic 3D stack with monolithic inter-tier vias (MIVs) connecting the two tiers.

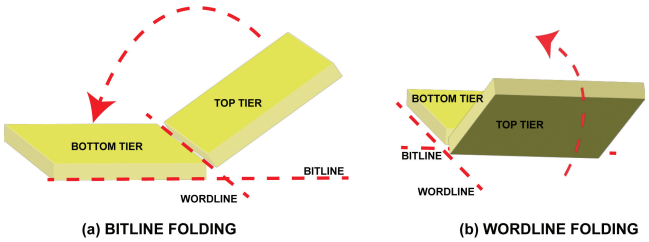

Figure 9: Folding the array along the (a) bitlines, (b) wordlines.

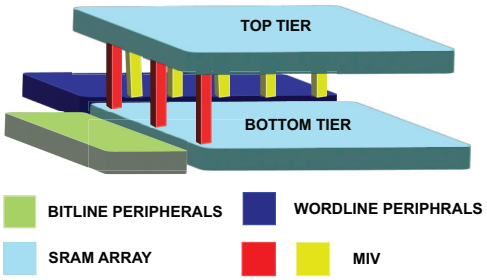

Figure 10: 3D view showing the location of MIVs in our 3D array designs.

5 3D FOLDED SRAM DESIGNS

5.1 3D Integration Technology Used

We leverage the Z-dimension to reduce interconnect latencies and minimize the footprint of our 2D memories. While previous studies explored various 3D SRAM array approaches, recent advancements have primarily focused on integrating memory and logic on separate tiers [4, 8]. Despite offering reduced interconnect lengths, such designs often necessitate advanced cooling solutions and are limited to specific use cases. In contrast, we propose a standalone 3D SRAM Array suitable for integration into any existing full chip design, offering connectivity similar to 2D arrays.

The integration methodology choice depends largely on the pitch of the required 3D vias. Given our design's smallest pitch, the wordline pitch of 128nm, traditional Through Silicon Vias (TSV) are impractical due to their larger size [11]. Although hybrid bonding using CuPads offers nanometer-scale sizes and pitches, it still falls short of our requirements. Monolithic 3D IC (M3D) presents a promising alternative, involving sequential device fabrication, connected using Monolithic Intertier Vias (MIV). MIV-based 3D tier integration allows the creation of very high-density 3D vias with sizes less than 100nm [12–14]. Consequently, our design adopts MIV-based 3D tier integration, sequentially fabricating devices on top of each other in two tiers, as depicted in Figure 8.

The pitch of the Monolithic 3D ICs (MIVs) in our design is well-defined. Utilizing the CoolCube process detailed in [15], we achieve a 3D MIV pitch of 110nm, accompanied by a significantly reduced thermal budget, and a via diameter of merely 50nm. To incorporate a model of this via into our work, it is imperative to comprehend the RC effects of this structure within our 15nm technology, subsequently facilitating its utilization in post-layout simulations. Based on the via employed in [2], we assign a resistance of 100 Ω. For simulation purposes, we construct an RC delay model of this via utilizing the M3D model described in [16], attributing a via capacitance of 1 fF. Through our simulations of the two tiers, we observe a worst-case delay imposed by our 3D Via to be just 1 picosecond.

5.2 Overview of the Approach

We introduce two distinct 3D configurations of the SRAM array: bitline folded and wordline folded. As elucidated previously, we partition the cell array into two tiers along either the bitlines or the wordlines. The creation of these configurations aims to individually analyze the impact of shortened bitlines and wordlines on various performance metrics of the SRAM array, as well as to assess the area saved through Z-dimension utilization. The conceptual representation of this approach is illustrated in Figure 9. For both our bitline and wordline folded configurations, the location of the MIVs are fixed, as shown in Figure 10. Subsequent sections are dedicated to delving into the architecture and other intricacies of these two configurations.

5.3 3D Bitline Folded Array Design

The rationale behind the 3D Bitline Folded (3DBL) configuration is rooted in an in-depth analysis of the impact of long bitlines on read/write latency and energy consumption. With elongated bitlines, there is a substantial increase in the energy required to drive signals through them. Various approaches, including additional drivers and circuits employing read/write assist techniques, have been proposed and extensively studied to mitigate the RC parasitics of long bitlines. However, while these methods do alleviate certain aspects of the issue, they exacerbate others—namely, they significantly elevate energy consumption and introduce a considerable area overhead to the SRAM array. Given that memory macros already occupy the largest area in a chip, this exacerbation compounds the problem.

The unique solution we propose aims to address both of these challenges by partitioning the bitcell array into two tiers and folding the long bitlines into two equal halves. In this configuration, both tiers share the same set of bitline and wordline peripherals, which are situated in the bottom tier. This strategic decision offers several advantages. Firstly, it consolidates all I/O pins on one tier,

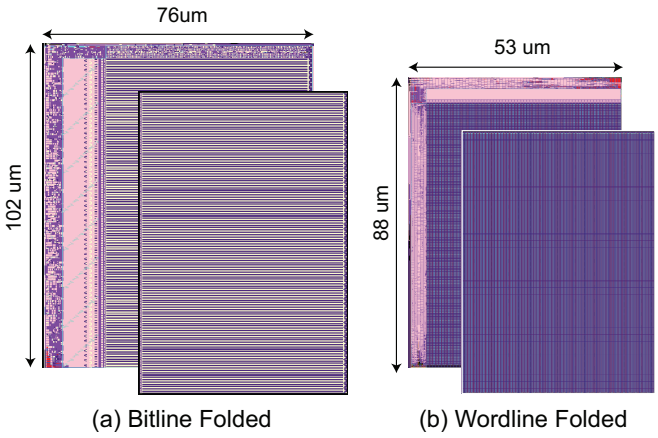

Figure 11: GDS layouts of 512WL x 128BL SRAM array. (a) 3D bitline folded, (b) 3D wordline folded.

facilitating easier connectivity to memory controllers or other modules located on the bottom tier. Secondly, it reduces the number of components used in the array. While we employ a redesigned wordline decoder capable of serving both tiers, the bitline peripherals remain unchanged and can be reused from the 2D arrays. Consequently, no additional control circuitry is required to access the words in the top tier, except for the wordline decoder. During any given read or write cycle, only the bitlines corresponding to the active wordline on either tier will be active. A visual representation of the bitline folded array is depicted in Figure 9 (a).

The EDA tools we use are all 2D, therefore, the layouts of the two tiers will be created separately and they will be connected together using the 3D integration strategy that was discussed in Figure 8. In our 3DBL 256WL x 64BL array configuration, we adopt a two-tier approach featuring 128 wordlines (WL) and 64 bitlines (BL) in each tier. Regarding the peripherals, we employ an 8-to-128 decoder for the wordline decoding process, as detailed in Section Figure 3.2, with its layout illustrated in Figure 3. The most significant bit (MSB) of the address signal determines whether the address pertains to the wordlines on the top or bottom tier. However, it's noteworthy that each tier maintains an equal number of bitlines per wordline, resulting in identical specifications for the row multiplexers and the bitline signal select decoders across both tiers. Consequently, the left side of the array remains unchanged between the 2D and 3D Bitline Folded (3DBL) designs. For brevity, the layout of this configuration is not shown here.

In our 3DBL 512WL x 128BL array configuration, we have two tiers of 512WL and 128 BL each. In terms of the peripherals, the wordline decoder used is the 9-to-256 decoder. Figure 11(a) offers a visual representation of the two tiers comprising the 3DBL array, wherein the top tier exclusively comprises SRAM bitcells, while the entire peripheral logic is located in the bottom tier.

5.4 3D Wordline Folded Array Design

In the 3DWL array configuration, we fold the array vertically along the wordline by tier segmentation. Similar to previous designs, the bottom tier contains peripherals and half of the bitcells, while the top tier holds the remaining bitcells. This configuration reduces both the width and number of bitline peripherals. Halving the wordline width decreases the number of 8-bit words per wordline, necessitating fewer row multiplexers and control bits, resulting in downsized bitline peripherals. We investigate the impact of shortening wordlines on read/write latency and energy efficiency. Reducing the word width (i.e., decreasing the number of bitlines per wordline) leads to a wordline length approaching that of the bitline. Extended wordlines require additional drivers to maintain signal integrity, especially for deeper bitcells.

While wordline peripherals undergo slight modifications, we choose to adapt the address decoder described in Section 3.2 to serve both tiers' wordlines. This decision considers the inherent behavior of SRAM cells, where simultaneous activation of wordlines in both tiers risks triggering destructive operations on bitlines associated with both wordlines. Employing bidirectional control logic in bitline peripherals would incur significant area overhead and complicate pin pitch matching. Thus, modifying the wordline decoder offers a more pragmatic solution. The visual representation of wordline folding is depicted in Figure 9 (b).

In our 3DWL 256WL x 64BL array configuration, we have 256 wordlines, each with a width of 64 bits or 8 8-bit words. We use a 9-bit address, ADDR, for the 9-to-256 decoder, where ADDR[7:0] is used to decode the exact wordline out of the 256, while the MSB, ADDR[8], is used as the tier select bit. The important thing to note here is that each tier, only has 4 8-bit words instead of 8 in the 2D counterpart, and as described previously, this will shrink the bitline peripherals by half. In our 3DWL 512WL x 128BL array configuration, we have 512 wordlines, each with a width of 128 bits or 16 8-bit words. We use a 10-bit address, ADDR, for the 10-to-512 decoder, where ADDR[8:0] is used to decode the exact wordline out of the 512, while the MSB, ADDR[9], is used as the tier select bit. Figure 11(b) shows the layouts of the top and bottom tiers.

6 EXPERIMENTAL RESULTS

6.1 Simulation Methodology

Parasitic extraction of both the 2D designs and the separate tiers of the 3D designs is conducted using Siemens Calibre, yielding comprehensive parasitic netlists. These netlists are subsequently employed in our post-layout simulations using Cadence Spectre. Notably, in the case of our 3D designs, inter-tier connectivity is facilitated through the utilization of the MIV RC delay model. All simulations are executed at a clock frequency of **1 GHz** to ensure consistency and enable accurate assessment of performance metrics across different design configurations. Our simulation methodology adheres to a straightforward yet robust strategy, where we write to the last word of the last wordline and after one idle cycle, read back the value to verify its correctness. The comprehensive summary of our post-layout simulation results is presented in Table 2. A comparative analysis with the commercial 16nm technology memory designs serves as a benchmark for evaluating our work.

6.2 Performance Metrics Used

Our simulations utilize a 50% input-output signal threshold for metric measurement. While read latency is pivotal for SRAM macros, a comprehensive evaluation requires consideration of all relevant

metrics, especially for full-chip designers optimizing specific pathways. In addition to read latency (which is the only performance benchmark in [8]), write latency is crucial for timing insights across applications. Energy efficiency priorities may also require understanding both read and write energies. Hence, our analysis covers all four vital metrics from 2D and 3D simulations: read and write latency, and read and write energy consumption.

The **read latency** is one of the most important metrics to determine memory performance, and it is what we are trying to reduce through our 3D folding strategies. We don't consider the setup times of the address and data registers in our simulations. We calculate read latency as the time taken from DEN $1 \rightarrow 0$ to $RD_x < 7:0>$ to go from $0 \rightarrow 1$ or vice-versa.

The **write latency** is measured from the time DEN is active to the time either the BL or the BLB signal gets discharged depending on the value being written to the cell. The time for the internal bitcell nodes to trip is measured in a separate simulation and that latency is added to the previous value to give the overall write latency. Given that the majority of the write operation is dependent on the bitline parasitics, it is unsurprising to see that the 3DBL design is faster in both the 256WL x 64BL and the 512WL x 128BL configurations by **61.21%** and **47.32%** respectively compared to the 2D baseline.

The **read/write energy** is calculated per read/write half cycle, when DEN is low so as to neglect leakage power and consider only dynamic power. For both read and write energy, in our designs as well as the metrics reported for commercial technology, we use a common equation to calculate the dynamic read and write energy, shown in (1), where T is the time period and 'n' is the cycle:

$$\left[\int_{nT}^{\frac{(2n+1)T}{2}} I_{\mathrm{VDD}}\, dt\right] \times V_{\mathrm{DD}} \tag{1}$$

6.3 Footprint Analysis

Table 3 illustrates the footprint comparison between our 2D and 3D SRAM arrays and 2D arrays fabricated using commercial 16nm and 28nm technologies. Employing folding techniques along wordlines, as delineated in Figure 9, results in a significant reduction in the size of bitline peripherals by half, leading to footprints **56.84%** and **58.25%** smaller for the 3DWL arrays compared to their 256WLx64BL and 512WLx128BL 2D counterparts, respectively. This reduction is primarily attributed to the decreased number of multiplexers required in the bitline peripherals and the halved size of the row decoder. Conversely, folding along bitlines only reduces the bitcell array's area by half, while the bitline peripherals remain unchanged.

To establish a baseline for our 15nm 2D arrays, we conducted a comparative analysis of their area against arrays of identical sizes fabricated using commercial 16nm technology nodes. Our findings indicate closely comparable areas between the two, with any observed discrepancies likely stemming from the incorporation of additional control circuitry and peripherals in the commercial memory arrays.

Finally, for the 256WLx64BL array, a maximum area imbalance between the top and bottom tiers of 58.6% is observed in the 3DBL design, while in the 512WLx128BL array, it stands at 57%. Conversely, the area imbalance in the 3DWL designs for both array sizes is 30%.

Table 2: Post-Layout simulation results of our 2D and 3D SRAM arrays, compared with the 2D array metrics from a commercial 16nm technology.

	Our work			Com 16nm
	2D	**3DBL**	**3DWL**	**2D**
256WL x 64BL				
Read Latency (ps)	102.5	86.3	93	165.6
Write Latency (ps)	59.8	23.2	38.9	114.2
Read Energy (fJ)	120.8	107	102	300.7
Write Energy (fJ)	111.6	98.6	91.8	193.6
512WL x 128BL				
Read Latency (ps)	179.4	146.2	145.9	184.7
Write Latency (ps)	75.2	39.6	67.3	121
Read Energy (fJ)	277.6	254.6	235.6	628
Write Energy (fJ)	303.5	253.3	229.2	401

Table 3: Footprint analysis of our 2D and 3D SRAM Arrays, compared with 2D arrays designed using 16 and 28nm commercial technologies. For 3D arrays, the dimensions of the bottom tier are used to calculate the area as it is larger. Bitcell density is reported as the total number of bits divided by the total area of the array, and is rounded to the nearest integer.

	Our work			Com 16nm
	2D	**3DBL**	**3DWL**	**2D**
256WL x 64BL				
X (μm)	55	38	44	41
Y (μm)	53	53	28	85
Area (μm^2)	2915	2014	1232	3485
Bitcell Density ($/\mu m^2$)	6	8	13	5
512WL x 128BL				
X (μm)	109	76	88	40
Y (μm)	102	102	53	278
Area (μm^2)	11118	7752	4664	11120
Bitcell Density ($/\mu m^2$)	6	8	14	6

6.4 Analysis of Simulation Results

Analysis of Table 2 underscores the superior performance of our 3D arrays compared to their 2D counterparts. Notably, the 3DBL configuration emerges as consistently the fastest across both array sizes, exhibiting an average read latency enhancement of 17.16% and a notable 54.2% improvement in write latency. In the case of the larger array size, the 3DWL configuration demonstrates competitive read latency performance, attributed to reduced wordline lengths and significant reduction in bitline peripherals. However, this trend is not mirrored in write latency due to the continued presence of longer bitlines necessitating discharge.

Table 4: Differences between our memory simulation metrics with commercial memory compiler.

Our Work	Commercial
No register file used	Register file used
Short control path	Longer control path
Basic wire parasitics	Advanced wire parasitics

Regarding energy efficiency, the 3DWL array outperforms, boasting an average reduction of 15.3% in read energy consumption and over 21% in write energy compared to the 2D baseline across both array sizes.

Comparatively, state-of-the-art 16nm memory designs exhibit higher metric values, attributed partly to technological differences and complexities in design rules and parasitic definitions. Additionally, the utilization of more sophisticated control circuitry and incorporation of register setup and hold times contribute to these discrepancies, as explained in Table 4. Furthermore, some performance metrics such as the write latency are inferred from the datasheet as the hold time for the write data input and are not derived from post-layout simulations as we do not have the necessary files (due to the confidential nature of the IP) available to do so. Nonetheless, for benchmarking purposes, our robust 2D baseline simulation results affirm the competitive performance of our designs, aligning closely with state-of-the-art memories for the same array configuration.

6.5 Full-Chip Design with Folded Memory

To validate the applicability of our SRAM arrays in System-on-Chip (SoC) designs, we execute the entire RTL-to-GDS flow for the RISC-V based Rocket core [17] open-source processor. The design incorporates six memory modules, utilized by both the data and instruction caches, along with their corresponding tag arrays. We employ the larger of our two array sizes, specifically the 8kB 512WLx128BL array in the design. The RTL representation of our Rocket core corresponds to a single-tile quad-core configuration, featuring individual Floating-Point Units (FPUs) per core and shared L1 caches. Following RTL synthesis, incorporating our custom memory modules, we derive the synthesized gate-level netlist. During floorplanning, macro placement prioritizes external I/O connections. Subsequently, an in-house custom Place-and-Route flow is employed to obtain the final GDS.

Post-route analysis reveals timing and area parameters, detailed in Table 5. Notably, the design utilizing the 3DWL memory modules achieves a maximum area reduction of 36.38% and demonstrates the highest effective frequency of 1.36 GHz. Regarding power efficiency, the configuration employing the 3DBL setup exhibits the lowest total power consumption among the three designs. Upon further analysis, we observe that the 3D Via usage is the highest in the design using the 3DWL memory macros, which results in the marginally higher power consumption as compared to the design using the 3DBL memory macros. Furthermore, the memory macro area in designs utilizing our 3D memory modules notably surpasses that of the 2D module-based design. The 3D via pitch remains the same in both the 3D designs as the pitch of the wordlines remains constant. The GDS of the three designs are shown in Figures 12, 13 and 14. We don't show the GDS of the top tier as this is not a true 3D design with two tiers, rather, it is a 2D design that uses standalone 3D memory macros, designed using the methodology described in detail in the previous sections.

Table 5: Comparison of Rocket core designs using three different configurations of the 8kB 512WL x 128BL SRAM array.

Metric	2D	3DBL	3DWL
VDD (mV)		800	
Target Frequency (MHz)		1000	
Effective Frequency (MHz)	1007	1111	1360
WNS (ps)	7	100	265
Area ($10^{-3}mm^2$)	88.5	60.4	56.3
Memory Macro Area (%)	55.9	44.9	42.2
Power (mW)	48.3	47.1	47.6
3D Via Usage		3072	3840
3D Via Pitch (nm)		128	

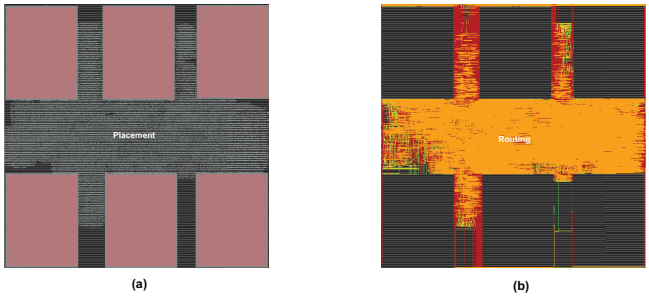

Figure 12: Final GDS view of Rocket Core (a) Placement (b) Routing using 2D Memory macros

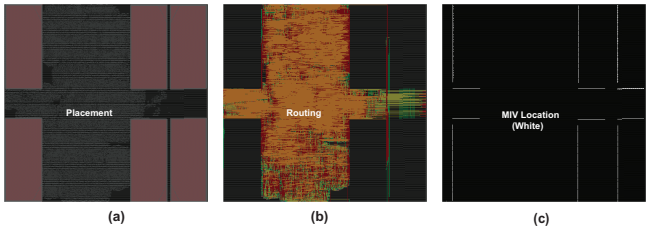

Figure 13: Final GDS View of Rocket core (a) Placement, (b) Routing, (c) Location of MIVs, using 3DBL memory macros. Only the bottom tier placement are routing are shown. Top tier contains the folded cell array of the bitline-folded cell array, containing half the total number of bitcells.

7 CONCLUSION

This work presents a novel methodology for constructing 3D SRAM arrays by "folding" 2D arrays along wordlines or bitlines. This technique enhances both read and write speeds while also improving energy efficiency. Physical layout comparisons demonstrate significant size reduction and performance improvement relative to 2D arrays. Post-layout simulations results prove this, with the 3DBL

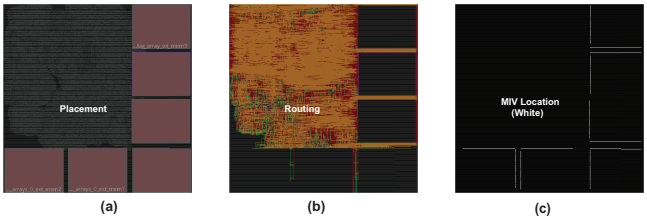

Figure 14: Final GDS View of Rocket core (a) Placement, (b) Routing, (c) Location of MIVs, using 3DWL memory macros.

design emerging as the fastest option, while the 3DWL design emerges as the most energy-efficient option. The 3DWL design exhibits a reduction of over 50% in area compared to the 2D baseline. This provides users with the option of high density, low power or high speed memory modules. Integration into a full-chip design illustrates real-world applicability and emphasizes the performance, power, and area advantages.

ACKNOWLEDGEMENT

This work was supported by the Logic and Memory Devices program of Semiconductor Research Corporation under Task 3142.001 and the National Science Foundation under CNS-2235398.

REFERENCES

[1] K. Puttaswamy *et al.*, "3d-integrated sram components for high-performance microprocessors," *IEEE Transactions on Computers*, pp. 1369–1381, 2009.

[2] M. Brocard, R. Boumchedda, J. P. Noel, K. C. Akyel, B. Giraud, E. Beigne, D. Turgis, S. Thuries, G. Berhault, and O. Billoint, "High density sram bitcell architecture in 3d sequential coolcube™ 14nm technology," in *2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, pp. 1–3, 2016.

[3] D. Bhattacharya and N. K. Jha, "Ultra-high density monolithic 3-d finfet sram with enhanced read stability," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1176–1187, 2016.

[4] J. Wuu, Agarwal, *et al.*, "3d v-cache: the implementation of a hybrid-bonded 64mb stacked cache for a 7nm x86-64 cpu," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, pp. 428–429, 2022.

[5] K. Bhanushali *et al.*, "Freepdk15: An open-source predictive process design kit for 15nm finfet technology," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, ISPD '15, (New York, NY, USA), p. 165–170, Association for Computing Machinery, 2015.

[6] M. Martins, J. M. Matos, R. Ribas, A. Reis, G. Schlinker, L. Rech, and J. Michelsen, "Open cell library in 15nm freepdk technology," 04 2015.

[7] W.-K. Yeh, Zhang, *et al.*, "The impact of fin number on device performance and reliability for multi-fin tri-gate n- and p-type finfet," *IEEE Transactions on Device and Materials Reliability*, vol. 18, no. 4, pp. 555–560, 2018.

[8] R. Chen, P. Weckx, S. M. Salahuddin, S.-W. Kim, G. Sisto, G. Van der Plas, M. Stucchi, R. Baert, P. Debacker, M. Na, J. Ryckaert, D. Milojevic, and E. Beyne, "3d-optimized sram macro design and application to memory-on-logic 3d-ic at advanced nodes," in *2020 IEEE International Electron Devices Meeting (IEDM)*, pp. 15.2.1–15.2.4, 2020.

[9] Y.-H. Chen, W.-M. Chan, W.-C. Wu, H.-J. Liao, K.-H. Pan, J.-J. Liaw, T.-H. Chung, Q. Li, G. H. Chang, C.-Y. Lin, M.-C. Chiang, S.-Y. Wu, S. Natarajan, and J. Chang, "13.5 a 16nm 128mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-vmin applications," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 238–239, 2014.

[10] M.-L. Fan, Hu, *et al.*, "Variability analysis of sense amplifier for finfet subthreshold sram applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 12, pp. 878–882, 2012.

[11] S. K. Samal *et al.*, "Monolithic 3d ic vs. tsv-based 3d ic in 14nm finfet technology," in *2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, pp. 1–2, 2016.

[12] B. Rajendran, R. S. Shenoy, D. J. Witte, N. S. Chokshi, R. L. DeLeon, G. S. Tompa, and R. F. W. Pease, "Low thermal budget processing for sequential 3-d ic fabrication," *IEEE Transactions on Electron Devices*, vol. 54, no. 4, pp. 707–714, 2007.

[13] M.-C. Nguyen, J. Yoon, A. H.-T. Nguyen, Y. Seok, N. Kim, H. Kim, S. Kim, and R. Choi, "Low-temperature deep ultraviolet laser polycrystallization of amorphous silicon for monolithic 3-dimension integration," *IEEE Electron Device Letters*, vol. 42, no. 6, pp. 784–787, 2021.

[14] Y. Park, J. Jeong, S. Noh, H. Kim, S. Kim, K. Kim, D. Kim, M. J. Kim, and B. J. Cho, "Application of pulsed green laser activation to top-tier mosfet fabrication for monolithic 3-d integration," *IEEE Transactions on Electron Devices*, vol. 71, no. 1, pp. 890–895, 2024.

[15] P. Batude, C. Fenouillet-Beranger, L. Pasini, V. Lu, F. Deprat, L. Brunet, B. Sklenard, F. Piegas-Luce, M. Cassé, B. Mathieu, O. Billoint, G. Cibrario, O. Turkyilmaz, H. Sarhan, S. Thuries, L. Hutin, S. Sollier, J. Widiez, L. Hortemel, C. Tabone, M.-P. Samson, B. Previtali, N. Rambal, F. Ponthenier, J. Mazurier, R. Beneyton, M. Bidaud, E. Josse, E. Petitprez, O. Rozeau, M. Rivoire, C. Euvard-Colnat, A. Seignard, F. Fournel, L. Benaissa, P. Coudrain, P. Leduc, J.-M. Hartmann, P. Besson, S. Kerdiles, C. Bout, F. Nemouchi, A. Royer, C. Agraffeil, G. Ghibaudo, T. Signamarcheix, M. Haond, F. Clermidy, O. Faynot, and M. Vinet, "3dvlsi with coolcube process: An alternative path to scaling," in *2015 Symposium on VLSI Technology (VLSI Technology)*, pp. T48–T49, 2015.

[16] S. Srinivasa, X. Li, M.-F. Chang, J. Sampson, S. K. Gupta, and V. Narayanan, "Compact 3-d-sram memory with concurrent row and column data access capability using sequential monolithic 3-d integration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 671–683, 2018.

[17] K. Asanović, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. A. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, and A. Waterman, "The rocket chip generator," Tech. Rep. UCB/EECS-2016-17, Apr 2016.