Folded Banks: 3D-Stacked HBM Design for Fine-Grained Random-Access Bandwidth

Vignesh Adhinarayanan
Advanced Micro Devices (AMD), Inc.
AMD Research and Advanced Development
Austin, Texas, USA
Vignesh.Adhinarayanan@amd.com

Bradford M. Beckmann
Advanced Micro Devices (AMD), Inc.
AMD Research and Advanced Development
Bellevue, Washington, USA
Brad.Beckmann@amd.com

Wantong Li*
University of California, Riverside
Department of Electrical and Computer Engineering
Riverside, California, USA
wantong.li@ucr.edu

Mohammad Seyedzadeh†
Microsoft Corporation
Redmond, Washington, USA
sm.seyedzadeh@gmail.com

Sergey Blagodurov
Advanced Micro Devices (AMD), Inc.
AMD Research and Advanced Development
Bellevue, Washington, USA
Sergey.Blagodurov@amd.com

Derrick Aguren
Advanced Micro Devices (AMD), Inc.
AMD Research and Advanced Development
Austin, Texas, USA
Derrick.Aguren@amd.com

Hayden Hyungdong Lee
Advanced Micro Devices (AMD), Inc.
Central Engineering
Austin, Texas, USA
Hayden.Lee@amd.com

Abstract

Despite significant improvements in peak bandwidth, the HBM industry has neglected random-access (irregular) bandwidth, limiting performance in many real-world applications. Improving effective HBM bandwidth is challenging due to power-constrained activations and coarse-grained, long-distance data movement. Rather than addressing these issues directly, hardware vendors have opted for incremental changes, achieving a 6.4× increase in sequential access bandwidth over two generations while leaving the irregular bandwidth challenges unresolved.

To remedy this, we introduce **Folded Banks (FB-HBM)**, a novel 3D bank design that redistributes bank subarrays ("folds") across multiple dies and relocates command, control, and global sense amplifiers to an additional base layer. By implementing this logic in a new base layer, we eliminate the DRAM die overheads inherent in previous designs. This architecture enables vertical routing of intra-bank wires—column select lines (CSLs) and master data lines (MDLs)—through thin-pitch through-silicon vias (TSVs) and hybrid bonds, significantly reducing RC power losses. By employing self-timed sense amplifiers, we eliminate costly dummy subarrays previously required for reference voltages. Furthermore, our distributed TSV architecture minimizes inter-bank data movement and we reduce HBM row size and activation energy by selectively disabling memory arrays (MATs) within a bank.

Compared to a projected HBM4 design, FB-HBM achieves a 6.7× improvement in random-access performance with a conservative 5 µm TSV pitch. This architectural advantage translates to a 2.28× speedup across high-performance computing (HPC) and sparse machine learning applications.

ACM Reference Format:
Vignesh Adhinarayanan, Bradford M. Beckmann, Wantong Li, Mohammad Seyedzadeh, Sergey Blagodurov, Derrick Aguren, and Hayden Hyungdong Lee. 2025. Folded Banks: 3D-Stacked HBM Design for Fine-Grained Random-Access Bandwidth. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA '25), June 21–25, 2025, Tokyo, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3695053.3731111

1 Introduction

Recent generations of HBM have employed effective bandwidth-scaling techniques to improve bandwidth for regular memory accesses. However, these techniques have not improved HBM's random-access (or irregular) bandwidth. Projections from an analytical bandwidth model (explained in Section 2.5) show that the irregular bandwidth is becoming only a fraction of the peak-regular bandwidth (see Figure 1).

This trend is exemplified in the expected 4th generation of HBM where random 64B cache line accesses to a 2 TB/s HBM stack encounter 1.5× lower bandwidth than sequential accesses. Effective irregular bandwidth decreases even further with smaller accesses, with only 8.3% of peak bandwidth reached for random 8B writes.

*Work done while Wantong Li was an intern at AMD RAD.
†Work done while Mohammad Seyedzadeh was with AMD RAD.

ISCA '25, Tokyo, Japan

ACM ISBN 979-8-4007-1261-6/25/06
https://doi.org/10.1145/3695053.3731111

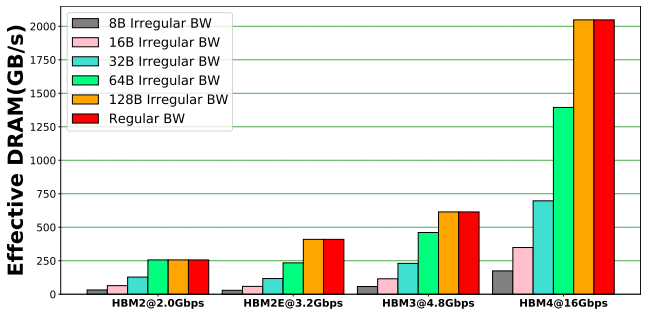

Figure 1: Regular vs. Irregular Bandwidth in HBM

Given the demands of graph processing and sparse machine learning (ML) workloads [3, 12], this paper aims to retarget the HBM architecture for this important class of memory behavior.

This paper proposes a new 3D-stacked memory organization, called Folded Banks High-Bandwidth Memory (FB-HBM), to improve irregular bandwidth. FB-HBM features a 3D bank organization that spans multiple DRAM dies and enables independent activations in each 3D slice within a bank. The banks are divided into smaller grains with reduced row sizes to enable efficient smaller DRAM accesses (also known as atom sizes). The design incorporates a novel self-referenced sense amplifier to reduce bank height, and uses distributed TSVs to reduce data movement and improve irregular bandwidth. Compared to past approaches, FB-HBM reduces power consumption for stream workloads by 2.8×, improves energy efficiency for GUPS by 4×, and achieves this without sacrificing DRAM density.

The contributions of this paper are the following:

- **Concurrent Row Activation Model.** We analyze current HBM3 technology and quantify concurrent row activations. Our findings reveal diminishing returns for further reducing row size and suggest future HBM designs should focus on minimizing data movement.
- **3D Folded Banks Architecture.** We propose a novel bank architecture, organizing a traditional 2D bank into a 3D "folded" structure. This design reduces data movement, increases activation parallelism by 8×, and results in 6.7× improvement in irregular bandwidth.
- **Area-efficient Folded Bank Design.** We identify dummy subarrays at the bank edges as a limiting design factor for folds and conduct a holistic evaluation of alternate designs. We show that introducing self-timed sense amplifiers at the bank edges eliminates the need for dummy subarrays and reduces area overhead from 25% to 8% in FB-HBM's folds.
- **Base Die Integration for Efficient Small Rows.** We implement row-segmentation circuits in the FB-HBM base die that reduce storage die area overhead from 3.2% to 1.54% while improving irregular bandwidth.
- **FB-HBM Application Analysis.** Using detailed cycle-level simulation, we showcase the efficacy of FB-HBM across various applications. Notably, our approach improves the performance of graph neural networks by 5× compared to a baseline HBM4 design.

2 HBM Background

This section describes the current HBM organization and operation in detail and highlights their limitations.

2.1 DRAM Organization

DRAM-based main memory is organized hierarchically, as shown in Figure 2. In DRAM, each bit of data is stored in a capacitor Ⓐ, and an access transistor Ⓑ connects the memory cell's capacitor to a bitline Ⓒ. A row of memory cells is connected to a wordline (WL) Ⓓ which drives the access transistors of each cell and enables the reading and writing of their values. Similarly, a bitline connected to a local sense amplifier (LSA) Ⓔ is shared by a column of memory cells, creating a 2-D array of cells known as the DRAM mat Ⓕ. The wordline architecture is organized hierarchically, and the wordline contained within a DRAM mat is referred to as the local wordline (LWL) Ⓗ.

The memory cells are also organized hierarchically as follows. A horizontal 1-D array of mats are organized into a subarray Ⓖ. Multiple subarrays then form a DRAM bank Ⓘ. The rows of cells across a subarray are connected by coarse-pitch metal wires called the master wordline (MWL) Ⓙ. The MWL feeds the local wordline drivers (LWD) Ⓚ which in turn drives the LWLs. To read or write data, the information is transferred from the memory cells to the local sense amplifiers (LSAs) and then to the global sense amplifiers (GSAs) Ⓛ at the bottom of a bank as shown in Figure 2. Wires known as the master data lines (MDLs) Ⓜ connect the LSAs and GSAs.

To decode row and column commands, DRAM uses row decoders Ⓠ and column decoders Ⓝ. The row decoder is responsible for driving the MWL during activation, while the column decoder drives the CSL (column select lines) Ⓟ for reading and writing data to and from the DRAM mat columns.

2.2 HBM Organization

In HBM, four or more DRAM dies are stacked vertically, interconnected by through-silicon vias (TSVs), as depicted in Figure 3. The memory controller retrieves data by dispatching DRAM commands and addresses via command and address (CA) pins Ⓤ, while data reception occurs through data (DQ) pins Ⓥ within the TSV stripe. The TSV density determines the number of CA and DQ pins in an HBM stack and determines its peak bandwidth. To improve bandwidth efficiency, HBM employs pseudo-channels (pCH) Ⓢ where pCHs within a channel share the CA pins, but divide the DQ pins equally as shown in red and pink in the bottom right corner of Figure 2.

Reading data from HBM incurs few additional steps compared to DDR DRAM as illustrated in Figure 3. First, a wide DRAM row is activated ❶. Then, a HBM atom-sized data (32B) is transferred to the central TSV stripe ❷. Next, the data is transferred vertically to the buffer and interposer dies via the TSVs ❸. Finally, the data is routed to the edge of the interposer for eventual transfer to the host processor ❹. Among these steps, data movement within the DRAM (❷) is the most energy intensive [33].

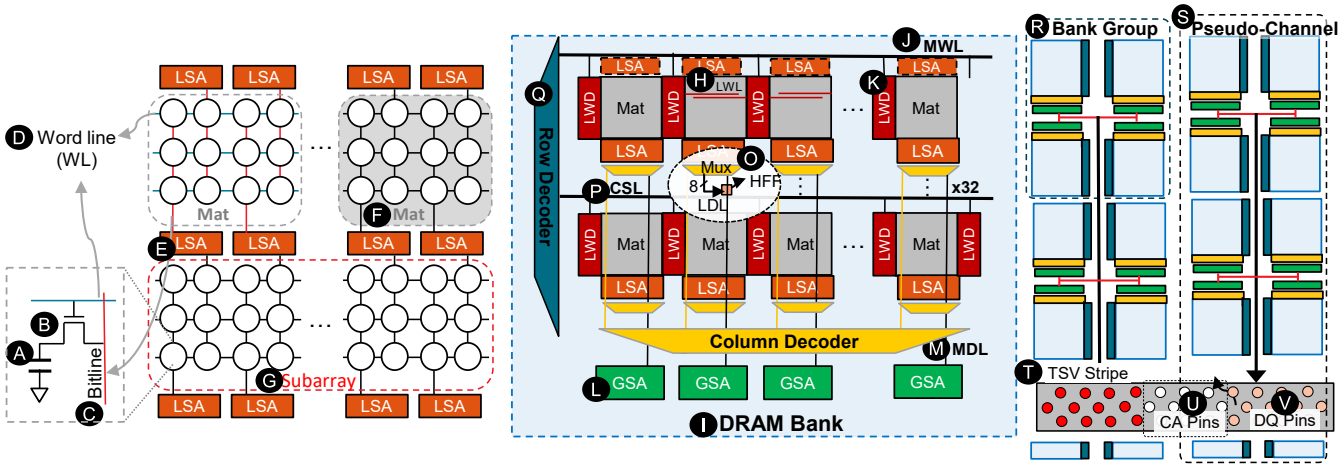

Figure 2: DRAM Organization

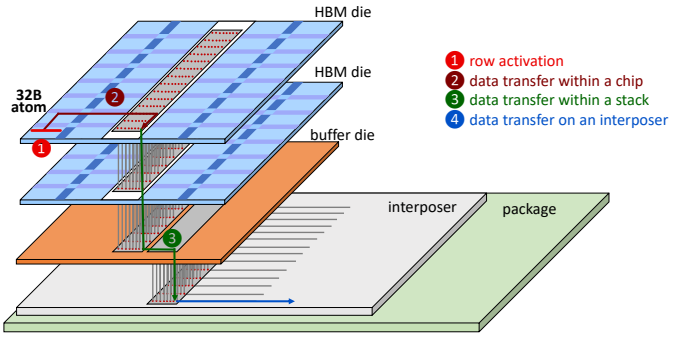

Figure 3: Baseline HBM Stack (from O'Connor et al. [33])

2.3 DRAM Operations

A DRAM access typically involves three fundamental operations: *row activation*, *column access*, and *precharge*.

Row Activation. To load data from memory cells into LSAs for read and write operations, a row must be activated. By issuing an ACT command, the memory controller triggers row activation. The targeted row's wordline is asserted, causing cell capacitors to adjust bitline voltages for LSA detection.

Column Access. After row activation and once the LSAs reach the ready-to-access voltage level, the memory controller initiates a CAS command. The time span between row activation and CAS command is represented as t_{RCD} (see Figure 4).

Precharge. For data read from a different row within the same bank, bitlines require resetting using a PRE command. Upon receiving PRE, the bank de-asserts the wordline, disconnects the activated row from bitline, and resets the bitline. The interval from issuing PRE to precharge completion is labeled as t_{RP}. Importantly, PRE can only follow an ACT command by t_{RAS} duration for proper word cell preparation (Figure 4).

2.4 Timing Constraints

HBM's bandwidth is determined by several key timing parameters. In this section, we introduce these timing parameters which underpin our analytical bandwidth models.

Timing Parameters for Column Commands. As Figure 4 shows, back-to-back column commands to the same bank are separated by the t_{CCD_L} interval. This separation interval is largely due to the high capacitive load on the shared MDLs and CSLs within a bank. Remarkably, improvements in I/O signaling technology has compressed the time taken to transmit an atom-sized data over the DQ pins (t_{BURST}), surpassing t_{CCD_L}. Therefore, HBM arranges four banks into non-sharing bank groups Ⓡ, and separates successive accesses to different bank groups by the shorter t_{CCD_S} parameter. This arrangement allows for faster transfers on the DQ pins, matching the feed rate from multiple bank groups and enabling gapless transmission on the DQ interface [2].

Timing Parameters for Row Commands. As shown in Figure 4, bank-cycle time ($t_{RAS} + t_{RP} = t_{RC}$) governs the issuing of successive ACT bank commands. ACTs also impose burden on the power-delivery circuitry. Thus, consecutive ACT commands within a four-activate window (t_{FAW}) are capped at four per channel. Additionally, due to the long-signal traces for activations, ACTs to the same bank group are separated by t_{RRD_L}, while successive ACT commands to different bank groups conform to the shorter t_{RRD_S} parameter.

2.5 Bandwidth Model

In this section, we present our analytical bandwidth model, elucidating reasons for irregular bandwidth's limited scaling.

Modeling HBM Regular Bandwidth. Maximum bandwidth occurs for an access pattern where consecutive requests (read/write) are issued to an open row. For example, in HBM3, such requests can be issued 2.5 ns (t_{CCD_L}) apart. However, HBM3's I/O signaling is faster – sending a 32B atom over 32 DQ pins operating at 6.4 Gbps in just 1.25 ns (t_{BURST}). To harness I/O effectively, data is interleaved from two different bank groups, shifting the bottleneck from t_{CCD_L}

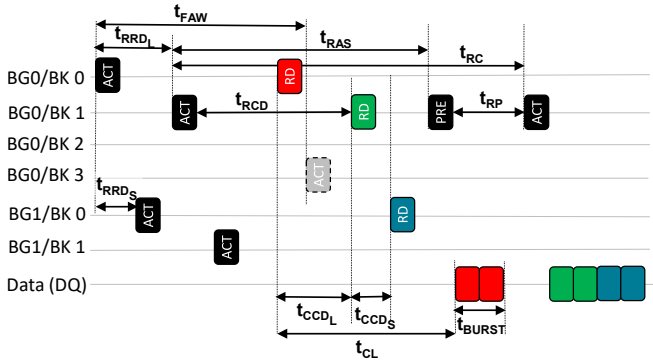

Figure 4: DRAM Timing Constraints

to either t_{CCD_S} or t_{BURST}. Thus, the peak regular bandwidth B_r is modeled as the minimum of t_{CCD_S}-induced or t_{BURST}-induced bandwidth.

$$B_r = min(\frac{S_{atom}}{t_{CCD_S}}, \frac{S_{atom}}{t_{BURST}}) \quad (1)$$

This model yields a channel[1] bandwidth of 25.6 GB/s for HBM3 and 32GB/s for a glide-path HBM4 design, referred to as Iso-HBM4 in this paper. While the HBM4 architecture is yet to be publicly defined, we assume DRAM core frequency, pin speed, channel count, and DRAM density scale at historical rates. The assumed configuration is summarized in Table 1.

Modeling HBM Irregular Bandwidth. Irregular bandwidth is governed by ACT command frequency — i.e., t_{RC}, t_{RRDS}, or t_{FAW} as explained in Section 2.4. Irregular bandwidth per channel (B_{ir}) is modeled as:

$$B_{ir} = min(\frac{n \times S_{atom}}{t_{RAS} + t_{RP}}, \frac{S_{atom}}{t_{RRD_S}}, \frac{\text{ACTs in } t_{FAW} \times S_{atom}}{t_{FAW}}) \quad (2)$$

where n is the number of banks in the channel and S_{atom} is the atom size.

Based on HBM3's timing constraints (Table 1), the irregular bandwidth for t_{RC}, t_{RRD}, and t_{FAW} is calculated as 11.38 GB/s, 16 GB/s, and 16 GB/s, respectively. Among these, t_{RC} limits HBM3's irregular bandwidth to 11.38 GB/s, creating a 2.25× gap compared to regular bandwidth. However, transitioning to a taller HBM stack trivially adds more physical banks and shifts the bottleneck to t_{FAW}, limiting irregular bandwidth to 16 GB/s (or 1.6× gap) in many configurations. In Iso-HBM4, row command frequencies remain unchanged, widening this bandwidth gap to 2×. Consequently, increasing row activations within the t_{FAW} window becomes crucial for improving irregular bandwidth.

3 Factors Limiting T_{FAW} and Irregular Bandwidth in HBM

HBM faces t_{FAW} limitations due to current spikes during multiple in-flight row activations. Excessive current draw during concurrent activations can cause erroneous sense amplifications. To mitigate this, memory vendors use the t_{FAW} timing parameter to restrict row activation frequency, effectively limiting current while also managing overall power consumption.

To develop strategies addressing these t_{FAW} restrictions, we propose a model for simultaneous (in-flight) row activations. This model identifies causes of low irregular bandwidth, offering insights to overcome these constraints.

[1]For brevity, we refer to pseudo-channel simply as a channel. When a distinction has to be made, we will use the term physical channel to refer to a traditional channel.

Table 1: Comparison of HBM Parameters

Type	*HBM3*	*Iso-HBM4*	*FG-DRAM*	*VFG-DRAM*	*FB-HBM*	*FB-G-HBM*
Channels	32	64	64	64	64	64
Banks	16	16	16	16	16	16
Grains	NA	NA	8	16	1	4
Subbanks/Grain	NA	NA	2	2	NA	NA
Folds/Bank	NA	NA	NA	NA	8	8
Row Size	1024	1024	256	128	1024	256
Sub Ch	32	64	512	1024	64	128
Sub Ch DQ pins	32	16	2	1	16	8
Sub Ch BW	25.6	32	4	2	32	16
Stack BW	819.2	2048	2048	2048	2048	2048
t_{RC}	45	45	45	45	45	45
t_{RCD}	18	16	16	16	16	16
t_{RP}	16	16	16	16	16	16
t_{RAS}	29	29	29	29	29	29
t_{CL}	16	16	16	16	16	16
t_{FAW}	16	16	16	16	16	16
t_{CCD_L}	2.5	2	8	16	1	1
t_{CCD_S}	1.25	1	8	16	1	1
t_{CCD_8B}	NA	NA	NA	NA	NA	0.25
t_{BURST}	1.25	1	8	16	1	1
t_{RRD}	2	2	2	2	2	2
t_{RRD_F}	NA	NA	NA	NA	NA	0.25
CA BW	1x	1x	2x	2x	2x	4x
ACTs in t_{FAW}	8	8	32	64	20	64
ACTs* in t_{FAW}	8	8	20	24	20	64
Atom Size (B)	32	32	32	32	8	8
TSV Stripes	1	1	2	2	16	16

3.1 Simultaneous Row Activation Model

JEDEC defines the IDD7 measurement loop to determine the current consumed during interleaved activate, read, auto-precharge operations [16]. In this section, we model IDD7 current, to determine the maximum number of simultaneous row activations achievable by various HBM designs.

IDD7 Current Model. Our IDD7 current model uses IDD7's predefined memory access pattern to determine the value for t_{FAW}. Our model expresses IDD7 current as the sum of two components. The first component is I_{act}, the frequency of activations within a t_{RC} window, multiplied by the current drawn for a row activation and precharge sequence. The second component is the bus utilization (bus_util) multiplied by IDD4R, the max current drawn for the intervening reads [24]. The following equation represents our model:

Table 2: IDD7 Workload Measurements (HBM3)

Parameter	Value	Parameter	Value
IDD4R	839 mA	tRC	45 ns
IDD7	958 mA	tRRD	2 ns
I_{act}	36 mA	Bus Util	44%
IMPT	4 CAS per ACT, 8.1 ACTs per tRC		
Workload	90% Bus Util, 990 mA		

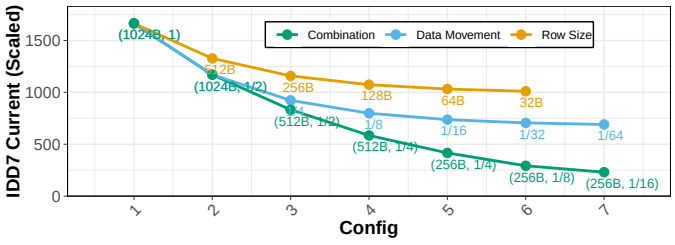

Figure 5: IDD7 (HBM4 estimates) vs. DRAM Config

$$\text{IDD7} = \frac{t_{RC}}{t_{RRD}} \cdot I_{act} + \text{bus\_util} \cdot \text{IDD4R} \quad (3)$$

By expressing IDD7 as the sum of two components, our model provides new insights into strategies for increasing simultaneous row activations. Traditionally, DRAM designs [33] focused on reducing row size to lower activation current (I_{act}), as IDD7 tests primarily stress the activation and precharge paths. However, our model shows that current also depends on data movement during reads (IDD4R). This dual dependency suggests that optimizing both components could reduce peak current and increase simultaneous activations.

To quantify the relative impact of row size and data movement, we infer I_{act} from HBM3 current measurements. Table 2 summarizes the measured value, workload characteristics, and estimated I_{act} values. Using this data, we project IDD7 current for various HBM4 configurations by considering their row sizes and data movement distances during reads and writes.

Simultaneous Row Activations Analysis. We present our analysis on simultaneous row activations when adopting two different strategies: reducing row size and reducing data movement distance. Figure 5 illustrates these findings, showing the IDD7 current consumed across successive halving of both parameters for the referenced Iso-HBM4 device.

Row size reduction, the traditional approach for improving parallel row activations, shows diminishing returns. IDD7 current reduction plateaus at 39% of the baseline, allowing only a 1.65x increase in simultaneous row activations.

Reducing data movement distance is more effective, yielding a 59% reduction in IDD7 current from the baseline and enabling a 2.4x increase in activations. Combining both strategies offers the greatest benefit: after seven halvings of either parameter, IDD7 current decreases to 13.8% of baseline, allowing 7.2x more row activations, demonstrating the synergistic effect of the two approaches.

3.2 Factors Limiting Irregular Bandwidth in HBM4

In this section, we identify specific aspects of HBM4's design, layout, and architecture that affect IDD7 current and power consumption which limit irregular bandwidth.

Wide Rows → High Row-Activation Current. Modern HBM banks are relatively wide, spanning 16 mats. This leads to high activation current during ACT commands. Reducing row size by design is undesirable due to area costs from circuit replication across smaller mat groups. The challenge, therefore, is to *efficiently* segment existing DRAM rows.

Central TSVs → Inter-Bank Data-Movement Energy. Centralized TSV arrays in current HBM designs account for 49% of energy consumption due to long-distance data movement from LSAs to TSVs. This energy cost is expected to increase with faster DRAM bus speeds. Distributing TSV arrays across the chip could reduce this energy cost, but the necessary keep-out zones (KOZs) pose a significant area overhead challenge.

Tall Banks → Intra-Bank Data-Movement Energy. The heights of banks in current HBM designs require long-distance transfers from the LSA to the bank's edge via the MDL. This results in high RC wire load, leading to significant intra-bank data-movement energy costs. Reducing the bank's height could improve energy efficiency and bandwidth, but the additional peripheral circuits needed for more, smaller banks incurs a higher area cost.

Large DRAM Atom Size → Data-Movement Energy. HBM's 32B atom size leads to energy waste during data movement, especially in applications like GUPS which only consume 8B per atom. This inefficiency draws additional energy and consumes internal HBM bandwidth, affecting performance. Reducing the DRAM atom size could address these issues but would require significant architectural changes.

Limited Bank Parallelism. Once the t_{FAW} bottleneck is resolved, irregular bandwidth becomes limited by insufficient parallelism for activations within the bank cycle time (t_{RC}). Directly reducing t_{RC} would require intrusive changes to the mat organization. A more practical solution is introducing additional intra-bank parallelism to mitigate the impact of high t_{RC} on performance.

4 Proposed Folded Bank HBM

In this section, we present FB-HBM which addresses many of the limitations of the Iso-HBM4 design outlined in the previous section.

4.1 3D Bank Architecture

The crux of the FB-HBM design is a revolutionary change of the conventional two-dimensional bank into a three-dimensional structure as shown in Figure 6. This 3D design aggressively reduces data movement while maintaining the reduced activation energy achieved by state-of-the-art designs [33]. As a result, FB-HBM increases activation parallelism and improves irregular bandwidth.

FB-HBM introduces an additional die in the base layer of an HBM stack to implement the DRAM's command and control logic, and vertically stacks each *fraction* of the bank's subarrays across individual DRAM dies. We assume an 8-Hi stack, resulting in one-eighth of the bank's subarrays residing in each DRAM die. The bank partition contained within a DRAM die is referred to as a *fold*. The

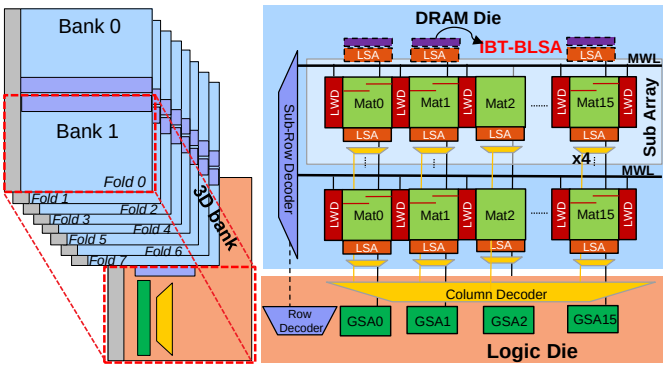

Figure 6: Folded Bank Design

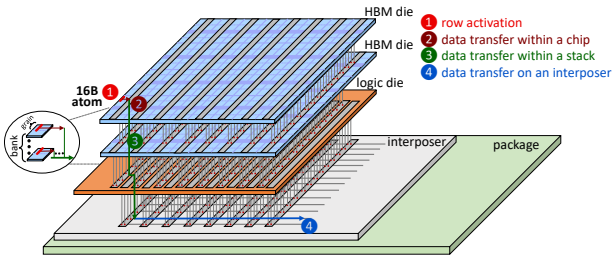

Figure 7: Proposed Folded Banks HBM stack

GSAs shared by all folds of a bank reside in the newly introduced base die.

4.2 Stack Organization in Folded Banks

FB-HBM introduces a novel stack organization that leverages distributed TSV arrays to reduce data movement by 16× within the DRAM die (Figure 7). Unlike traditional banks where TSVs only carry addresses and channel data, FB-HBM's TSV strips route bank-specific MDLs to GSAs and CSLs to column decoders, both located in the base die (Figure 6(a)). This innovative design enables GSAs to be shared across multiple bank-like partitions, while eliminating the need for signals to traverse the entire DRAM die.

FB-HBM utilizes hybrid bonding technology to meet the internal bandwidth requirements for the MDL and CSL routes and to minimize the KOZ area overhead. The wiring density requirements for routing MDLs and CSLs are 128 times higher than for DQ pins. Conservatively assuming a 5μm pitch for face-to-back bonding for stacked memory [11, 17, 22, 30], this technology achieves up to 40,000 vertical connections per mm^2, a 363-fold improvement over microbumps, and satisfies the MDLs and CSLs bandwidth requirements. Furthermore, the KOZ cost, which has typically been about 2.5 times the pitch size, is significantly reduced, thus saving area[2].

.

[2]Since hybrid bonding also reduces power and area costs in all HBM designs, we assume this technology is available for alternative designs to ensure a fair comparison (Section 5.3).

4.3 Design and Optimization of Folds for FB-HBM

FB-HBM's folded structure with just four sub-arrays, reduces effective bank height per die and enables simultaneous activations across multiple subarrays. This design improves performance and efficiency but introduces two key challenges: (1) providing reference voltage to the edge arrays while reducing bank height in an area efficient manner, and (2) maintaining multiple asserted MWLs for parallel row activations within a bank. This section explains these challenges in detail and presents our solutions.

Providing Reference Bitline Voltage. While FB-HBM's reduced bank height minimizes intra-bank data movement, traditional methods of providing a reference bitline voltage to edge arrays significantly increase area costs. When there are many subarrays within a bank, inserting dummy subarrays to provide reference voltages for the edge arrays has minimal area overhead. However, when there are few subarrays within a bank, such as the FB-HBM design, the area cost is prohibitive. Thus we considered multiple schemes for providing a reference voltage. First we considered a set of standard techniques such as using an open-bitline scheme [15, 23] with dummy subarrays at the bank's edges [31], a closed bitline scheme using adjacent bitlines for the reference[31], and a half-length mat scheme [37]. However we quickly dismissed their viability due to their respective area overhead of 25%, 33%, and 25% respectively. We also considered Bitline-Emulated CMOS Capacitors (BL-CAPs) because they have been validated in real DDR3 chips and only have an area overhead of 5%. However, BL-CAPs require individual post-fabrication fine-tuning, which is unlikely to scale to production methodologies.

Instead we use Imbalance Tolerant Sense Amplifiers (IBT-BLSA) [19] to provide the reference voltage because of its superior sensing and timing capability at an area competitive with BL-CAPs. A traditional bitline sense amplifier (BLSA) receives input from two bitlines (BLT and BLB) with equal capacitance. A small voltage difference between these lines is amplified to determine whether the value read is a '1' or a '0'. In contrast, an IBT-BLSA is designed for dummyless arrays and receives its second input from a much shorter BLB with lower capacitance than its first input. To accurately detect the voltage difference, an IBT-BLSA incorporates an extra circuit to boost the signal on the shorter BLB during a pre-amplification phase. Consequently, the two sides of the BLSA have separate control signals. Using precise control of timing and voltage levels, this circuit enables reliable data detection even with imbalanced bitline capacitances. After sensing, the IBT-BLSA equalizes the voltages to account for capacitance differences.

The IBT-BLSA employs an additional inverter for pre-amplification and uses distinct control paths for sensing. According to prior work [19], IBT-BLSA's height is 0.32 times that of a single subarray, resulting in an area overhead of only 8% for FB-HBM. SPICE simulations indicate a 1.2 ns reduction in equalization time and over a 100% increase in sensing voltage, demonstrating enhanced performance and reliability. Our evaluation conservatively presents performance results without factoring in the timing advantage.

Increasing Bank Parallelism. The second challenge in FB-HBM involves supporting simultaneous activations across subarrays within

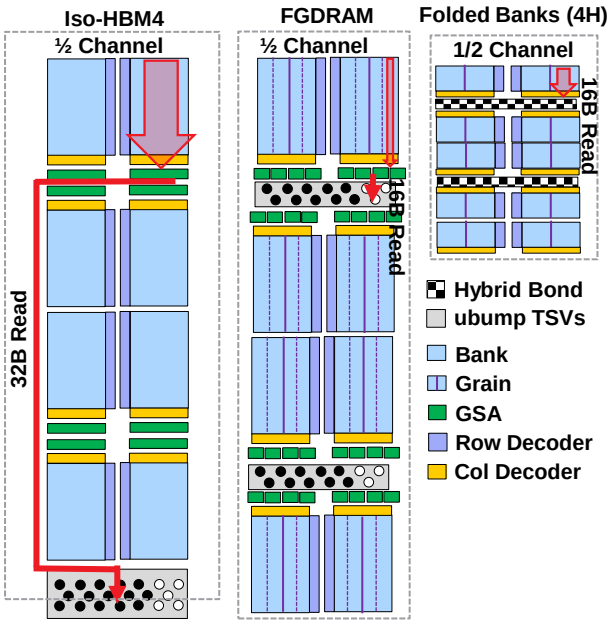

Figure 8: Data movement within bank and channel. FGDRAM layout is adjusted to use a 16 banks per channel configuration. FB-HBM layout shown for a 4-Hi stack for readability.

a bank. A DRAM row activation requires keeping a MWL and its associated LWLs asserted throughout the process. De-asserting these wordlines disconnects the row from the bitline, resetting it to the reference voltage. Increasing intra-bank parallelism becomes challenging because activating a second row before the first row is precharged causes premature de-assertion of the first row's wordlines.

To address this issue, previous designs employed address-decoupling latches for MWLs and CSLs [5, 33]. These latches preserve wordline assertions during parallel row activations. While effective for achieving intra-bank parallelism, this method incurs additional area costs due to the latch circuitry.

To reduce the latch costs, FB-HBM employs a split address decoding scheme, using a global decoder in the logic die and sub-row decoders in the DRAM dies (Figure 6). Partially pre-decoded addresses from the global decoder are sent to sub-row decoders based on fold IDs. Each sub-row decoder maintains one MWL assertion per fold, allowing multiple simultaneous row activations without extra address-decoupling circuitry. Notably, the sub-row decoders are smaller than the global decoder, leading to a net reduction in DRAM die area.

Notably, these pre-decoded addresses are transmitted via distinct TSVs to the folds, enabling a command delivery rate faster than t_{RRD_S} and t_{CCD_S}, provided the commands are sent to different folds.

4.4 Implementing the DRAM Grain Architecture

The grain architecture represents a significant innovation in DRAM bank design, aimed at reducing row size and enhancing parallelism [33]. This approach segments the MWL, dividing the DRAM bank vertically into two independent grains and creating parallel data paths within the bank. Each MWL segment is controlled independently, leading to a reduced DRAM row size. However, practical implementation complexities and increased area overhead pose challenges for its commercial viability.

To understand the difficulties in reducing DRAM row size, we analyze the key components involved in memory cell selection during row activation. The process begins with control circuitry driving a MWL associated with a specific address. In HBM3, each MWL connects to four rows of LWLs through LWDs. To activate a specific row, LWLSel wires are routed beneath each bank and an AND operation between MWL and LWLSel within each LWD activates the targeted row as shown in Figure 9(a). Extending this approach to the 4th generation of HBM and activating smaller rows of the grain architecture introduces two main challenges:

Pitch-Matched LWD *and* MWL. The LWD layout, including its drivers and circuitry, is pitch-matched with the MWLs to ensure a compact design. Modifications to the LWD circuitry can disrupt this alignment, increasing area overhead. This issue becomes particularly critical when implementing smaller rows, as additional circuitry is required to reduce the number of LWDs and LWLs driven by each MWL.

LWD *Proliferation.* Smaller DRAM rows require more LWDs, especially at bank edges. Typically, each LWD has left and right LWL arms extending to mats on either side, except for edge LWDs, which have only one arm. Consequently, 17 LWDs are needed to drive LWLs in 16 mats. Halving the DRAM row size doubles the number of edge LWDs, resulting in a proportional area increase.

4.5 Folded Banks + Grain (FB-G-HBM)

To address the challenges of implementing smaller rows, we propose the Folded Banks + Grain (FB-G-HBM) architecture, an extension of FB-HBM designed to incorporate small rows. This section begins by evaluating the limitations of three potential approaches to the design, before concluding by describing our solution to incorporate the grain logic in the base die.

Grain logic within the LWD. Implementing small rows requires activating subsets of LWDs within a row, necessitating additional control signals (e.g., grain selectors, GrSel) and three-input AND operations (MWL, LWLSel, GrSel). Embedding this logic within the LWD as shown in Figure 9(c) requires 4-6 transistors per LWL, at least doubling the transistor count within the existing LWD. This increase causes pitch mismatch between LWD and MWL and increases wiring complexity as further shown in Figure 9(c). The resulting layout inefficiency approximately doubles the mat size, rendering this approach impractical.

Segmenting Grain Logic. Optimizing the MWL segmentation logic by inverting LWLSel and GrSel signals before they reach the DRAM bank is one potential approach to reduce the area overhead. This approach replaces one of the two AND operations with a single-transistor NAND operation within the LWD (Figure 10(b)), reducing the total transistor count from 4-6 to 3 per LWL. While this improves

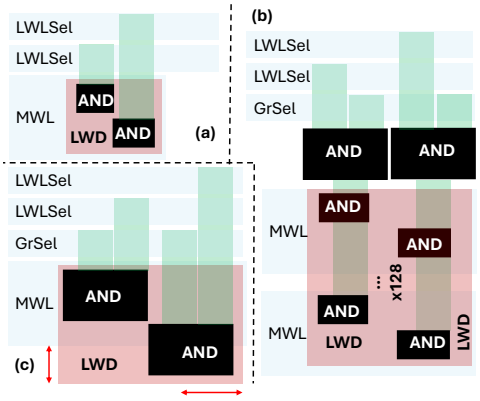

Figure 9: FG-DRAM Area Overhead.

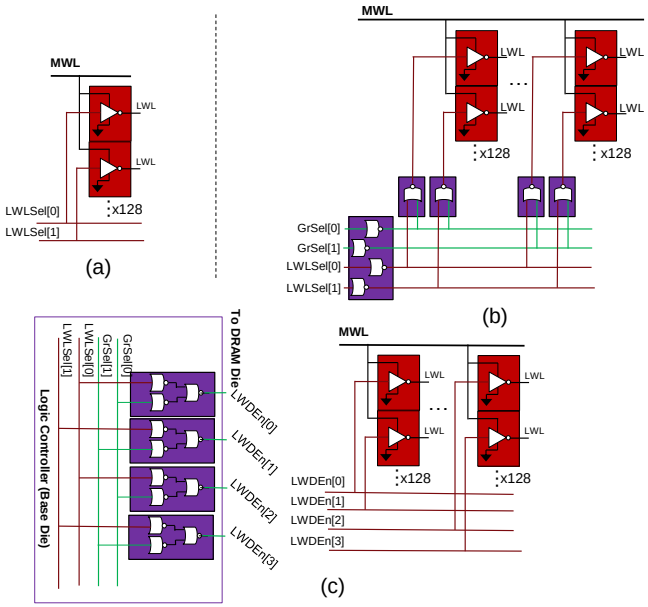

Figure 10: Circuit-level comparison of local wordline drivers: (a) HBM3, (b) FG-DRAM, and (c) FB-G-HBM.

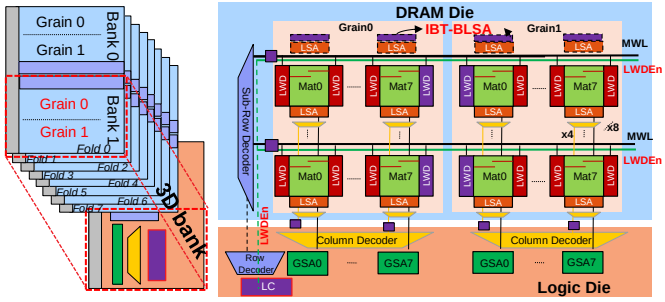

Figure 11: Folded Banks with Small Rows

area efficiency, a challenge remains: the extra transistor in the LWD still creates pitch mismatch with MWL, albeit with a reduced overhead of 24%.

Placing Grain Logic in the LSA-LWD Corner. FGDRAM proposed implementing small rows by adding the new AND gate (combining LWLSel and GrSel) in the corner space between the LWD and LSA. This gate's output would feed into the existing AND gate in the LWD, which combines it with the MWL signal. However, recent work reveals that this corner space is already occupied by critical components like IO switches [13]. In addition, while the logic in the corner amortizes the area costs across multiple MWLs within a mat as shown in Figure 9(c), the implementation disrupts the existing grid layout of mats, rendering the FGDRAM impractical. Specifically, the undesired layout would increase the subarray area by 3.5% for 4 grains due to additional circuitry and wiring.

Grain logic in Base Die. Our FB-G-HBM design addresses the limitations of the three previously described approaches by implementing MWL segmentation logic in the base die. This design performs the logical AND between GrSel and LWLSel signals in the bottom logic die, producing an LWDEn signal. LWDEn is then vertically transmitted through TSVs to the target DRAM die, where it activates the LWDs within the targeted grain (Figure 10(c)).

The vertical transmission is highly efficient due to hybrid bonding, which increases interconnect density by 363x, and allows FB-G-HBM design to support four grains per bank. Normally, there is an area cost associated with routing the LWDEn signal under a bank, which increases linearly with the number of grain partitions. However, memory vendors project increasing the number of metal layers in the HBM core die for future devices [35], ensuring that the additional wires required by our approach do not incur extra area costs. Although this approach potentially allows us to increase in the number of grains beyond four, we limit the grain count to four because it meets the energy targets for peak GUPS and avoids the additional LWD area costs associated with smaller grains.

In summary, we implement the grain logic in the base die so that the grain area overhead has minimal impact on the DRAM die and can best leverage the benefits of hybrid bonding.

4.6 Optimizing Base Die Data Movement in FB-G-HBM

Despite efforts to reduce IDD7 current by minimizing data movement (Sections 4.2 and 4.3) and row size (Sections 4.5), activation parallelism remains capped at 4.4x due to the energy costs associated with base-die data movement. Unlike other dies in a 2.5D stacked configuration, the physical distance of data movement on the base die cannot be reduced. To address this limitation, we developed alternative strategies to mitigate the associated energy cost.

We propose an 8B RD/WR option in FB-HBM to reduce the data volume transferred per column access for irregular access patterns, such as GUPS. This approach leverages the grain architecture's existing capability to fetch fewer bits per column select signal. However, it introduces three additional costs: (i) implementing a new SM_RD/SM_WR command, (ii) routing two extra address wires

Table 3: Workloads Summary

Class	Workload
Micro-benchmarks	STREAM, GUPS
Academic Workloads	NW, StreamCluster
Graph Analytics	GNN
HPC	LULESH, Havoq, Kripke, Laghos, LAMMPs, PENNANT, QMCPACK, QuickSilver, HPGMG, Mini AMR, MCB (CORAL) [1]
Language Models	BERT GEMM (Google) [8]
Recommendation	DLRM (Facebook) [29]

Table 4: Modeled GPU Parameters

GPU Parameters	Value
# of Compute Units (CUs)	80
# SIMD Units per CU	4
Max # Wavefronts per SIMD Unit	10
Memory Hierarchy	**Value**
GPU L1 D-Cache per CU	16 KB, 128B line, 32B sectored
GPU L2 Cache (all CUs)	4 MB, 128B line, 32B sectored
Latency (L1/L2/Mem)	50/125/225 cycles

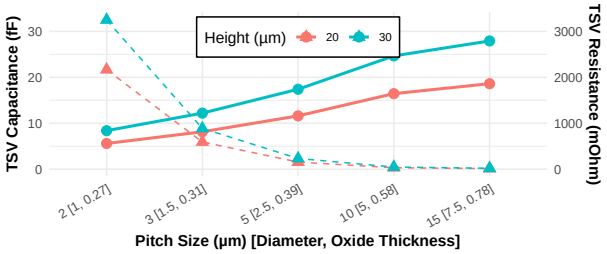

Figure 12: TSV Resistance and Capacitance Sensitivity

and decoding these additional bits, and (iii) a modest increase in PHY area to support the new command and address wires.

Normally, implementing an 8B atom with a 4x increase in command rate would require a proportional increase in CA/RA pins to accommodate the row and column addresses. However, HBM3 overprovisions these pins by a factor slightly greater than two. By reassigning a subset of CA pins originally intended for column addresses to row addresses, we further reduce the PHY area overhead. Our analysis indicates that increasing the total pins per channel from 72 to 76 is sufficient to handle the increased command rate and encode the new smaller access commands.

5 Methodology

In this section, we summarize our methodology to estimate performance, power, and area for the proposed FB-HBM design and its variants.

5.1 Simulation Setup

Trace Collection. We collect virtual address traces for the workloads summarized in Table 3 using a state-of-the-art GPU dynamic binary instrumentation tool. In addition, we also collect metadata associated with the addresses such as their timestamp, compute unit ID, and wavefront ID. We post process the collected trace and order the virtual addresses by time stamp. In round-robin fashion, we then re-distribute the requests by their work-group IDs to the modeled GPU's compute units. The GPU's configuration is summarized in Table 4.

GPU Model. We simulate the cache hierarchy of an AMD Instinct MI200 Series GPU (Table 4) and the caches are assumed to be sectored (32B) to make use of the small-element accesses. For address translation, we assume a random virtual-to-physical address mapping. The last-level cache misses are collected and the resulting DRAM trace is provided as input to our detailed memory simulator.

Memory Simulation. We modified Ramulator [21] to accurately simulate HBM in detail. We tuned the simulator for HBM1 and HBM2, achieving bandwidth estimates within 9% of real-silicon measurements for GUPS and STREAM.

We selected an address mapping policy that eliminates most bank and channel conflicts for common access patterns via experimentation. We use an FR-FCFS scheduler with write drain mode and a closed-page policy for irregular applications. A single-bank refresh scheme reduces stalling during refresh operations. For FGDRAM and FB-HBM, we implemented additional hierarchical elements — grains and folds — along with their associated timing constraints. For FB-G-HBM, we also modified the simulator to accept 8-16B requests.

Modeling Timing Parameters. Accurate timing parameters are essential for evaluating, optimizing, and assessing the feasibility of HBM designs. Therefore, we obtained reference timing parameters for HBM3 from memory vendors (Table 1), which served as a baseline for modeling FB-HBM and Iso-HBM4. These architectures involve modifications to CSLs and MDLs, including reduced wire lengths and partial vertical routing, necessitating the use of DRAMSpec [28] for accurate modeling.

We tuned and validated DRAMSpec to align with HBM3 timing data. The resulting parameters, shown in Table 5, provided the basis for HBM4 projections. Wire resistance and capacitance were extrapolated using NeuroSim [6] based on feature sizes, while TSV parameters were modeled independently [18, 43].

t_{CCD_L} and t_{CL}. Signal path capacitance and resistance directly influence t_{CCD_L}, as expressed by:

$$t_{CCD_L} = t_{CSL} + t_{SSA_P} + t_{MDL} - t_{Dr} \quad (4)$$

$$t_{CSL} = t_{Dr} + t_{90} \cdot R_{CSLDr} \cdot C_{CSL} + t_{63} \cdot R_{CSL} \cdot C_{CSL} \quad (5)$$

$$t_{MDL} = t_{Dr} + t_{90} \cdot R_{MDLDr} \cdot C_{MDL} + t_{63} \cdot R_{MDL} \cdot C_{MDL} \quad (6)$$

The resistance and capacitance of the CSL and MDL lines dominate these calculations. FB-HBM reduces wire capacitance and resistance but introduces up to eight TSVs in the signal path. TSV capacitance and resistance depend on pitch size, TSV diameter, die height, and oxide thickness, as shown in Figure 12. We conservatively select a 5 μm pitch, yielding 11.6 fF capacitance and 154.9 mΩ resistance. Future designs may reduce TSV overhead further.

Crossing eight TSVs adds 92.8 fF capacitance and 1.2 Ω resistance. However, shorter wire lengths in the 2D bank reduce capacitance and resistance, resulting in a net reduction of 31 fF in capacitance

Table 5: DRAMSpec Calibrated Timing Parameters

Parameter	HBM3	Iso-HBM4	FB-HBM
Technology Node (nm)	16	9	9
Bank Height (μm)	918.4	635.5	95.8
Driver Enable Delay (ns)	0.2	0.2	0.2
CSL Driver Resistance (Ω)	250	300	300
CSL Load Capacitance (fF)	8	8	8
SSA Pre Delay (ns)	0.2	0.2	0.2
Wire Resistance (Ω/mm)	2670	4000	4000
Wire Capacitance (fF/mm)	180	180	180
MDL Driver Resistance (Ω)	200	300	300
TSV Resistance ($m\Omega$/TSV)	–	–	154.9
TSV Capacitance (fF/TSV)	–	–	11.6
CSL Resistance (Ω)	2450	2550	381.8
CSL Capacitance (fF)	173.1	122.7	91.6
MDL Resistance (Ω)	2450	2550	381.8
MDL Capacitance (fF)	165.1	114.7	83.6
t_{CCD} (ns)	2.55	2.07	1.044
t_{CL} (ns)	15.8	13.4	10.04

Table 6: Component-Wise Energy Cost (pJ/b)

Component	Iso-HBM4	FGDRAM	VFGDRAM	FB-G-HBM
Data Movement Energy				
↪ Intra-Bank	0.21	0.21	0.21	0.03
↪ Inter-Bank	2.01	1.17	1.17	0.22
↪ Thin-pitch TSV*	*0.15*	*0.15*	*0.15*	0.15
↪ Interposer	0.18	0.18	0.18	0.18
Data Movement Cost	2.55	1.71	1.71	0.58
Implied Cost for GUPS	10.21	6.82	6.82	0.58
Row Activation Energy				
Stream ACT Cost	0.05	0.05	0.05	0.05
Implied Cost for GUPS	5.80	1.45	0.73	1.45
Total Energy				
Stream Energy	2.60	1.75	1.75	0.63
GUPS Energy	**16.02**	**8.27**	**7.55**	**2.03**

*Revised values assume benefits of thin-pitch TSVs, even if not planned. Original cost was 0.4 pJ/b.

and 2.1 kΩ in resistance. These optimizations enable FB-HBM to achieve a t_{CCD_L} of 1 ns, compared to 2.5 ns for HBM3 and 2 ns for Iso-HBM4. Similarly, t_{CL} reduces from 13.4 ns to 10 ns.

Our methodology to estimate t_{FAW} is elaborated in Section 6.1. The remaining HBM parameters, as summarized in Table 1, were projected based on trends from previous generations and anticipated performance requirements.

5.2 Points of Comparison

We evaluate our proposed designs against the Iso-HBM4 and FGDRAM designs, as well as an extended FGDRAM variant. For a fair comparison, all designs assume 2 TB/s per stack bandwidth and we increase FGDRAM banks per channel [33] to 16 to align its density with other designs. Furthermore, we assume hybrid bonding and thin-pitch TSVs are available for all designs, reducing energy and area costs (Tables 6 and 7).

FGDRAM Modifications. The increased bank count requires LSA-GSA data movement in FGDRAM to span two banks (Figure 8), although this does not significantly affect data movement distance. Each grain's DQ pins and associated GSA are now shared by four slices across four physical banks. The sharing of GSAs requires a higher t_{CCD} for back-to-back accesses to banks within the same grain, similar to bank grouping. However, the t_{CCD_L} cost is already increased to match t_{BURST} and thus does not require further increases.

Similar to the original FGDRAM proposal, we split grains into two partitions with 256B rows and refer to the partitions as pseudo-banks. The original FGDRAM design, featured 16 pseudo-banks per channel and exposed a tRC bottleneck. Because our evaluated FGDRAM design targets a denser technology generation, we increase the pseudo-banks per channel to 64, which better utilizes the extra ACTs available in the t_{FAW} window. To support this improvement and double FGDRAM grain's bandwidth from 2 GB/s to 4 GB/s, we assume additional command pins are available. These changes provide an optimistic baseline of FGDRAM performance.

For both FGDRAM variants, we refine our area cost estimates, accounting for the previously described modifications and the unoverlapped circuit shown in Figure 10(c).

VFGDRAM. We introduce a variant called VFGDRAM, which extends FGDRAM's strategy by further reducing row size to 128B. This variant demonstrates that shrinking the row size below 256B reduces the activation energy further, thus increasing the number of ACTs possible in the t_{FAW} window. Table 1 summarizes the configuration and timing parameters for all evaluated variants.

5.3 Power and Area Modeling

In this section, we describe our methodology to determine the power and area for FB-HBM and the other points of comparison.

Power Model. We model power consumption by scaling HBM3 microbenchmark measurements from power rails to HBM4 technology. Row activation energy is estimated by scaling HBM3 IDD0 measurements to HBM4 following the linear regression trends across HBM generations. Read and write power are first measured from IDDR4R and IDDR4W tests on HBM3 then the Pre-GSA and post-GSA energy components are determined by applying the same ratios as prior HBM2 studies [33].

Data movement distances are measured from HBM2 die shots. For our designs, pre-GSA and post-GSA energy is scaled based on bank height and the proposed HBM layout (Figure 8). I/O energy is calculated and scaled to future technology nodes based on estimates from Chatterjee et al. [5]. We present a summary of our power model in Table 6.

Area Model. We derive the baseline DRAM macro dimensions from O'Connor's detailed 20 nm HBM2 die-shot analysis [31] and scale them to a 9 nm node for Iso-HBM4. The estimated area for the HBM2 components are summarized in Table 7, and our scaled numbers appear in the Iso-HBM4 column.

For FG-DRAM and FB-G-HBM, we calculate the incremental and reduced area costs by identifying which macrolevel components are added, removed, or replaced in the DRAM die relative to the Iso-HBM4 design and then aggregate those changes.

Table 7: Component-Level Area Breakdown for DRAM Die*

Hierarchy	HBM2 (20 nm)	Iso-HBM4 (9 nm)	FG-DRAM (9 nm)	FB-G-HBM (9 nm)
Subarray (μm^2)				
↪ Mats	14,468 (18 × 803.8)	2,927 (18 × 162.63)	3,021 (18 × 167.83)	3,003 (18 × 166.86)
↪ LWD Stripe	1,833.5 (19 × 96.5)	369 (19 × 19.4)	427 (22 × 19.4)	427 (22 × 19.4)
↪ Local SAs	3,715.2 (18 × 206.4)	747 (18 × 41.5)	747 (18 × 41.5)	747 (18 × 41.5)
Subtotal	**20,017**	**4,043**	**4,195**	**4,177**
Bank (μm^2)				
↪ Subarrays	680,578 (34 × 20,017)	137,459 (34 × 4,043)	142,624 (34 × 4,195)	133,672 (32 × 4,177)
↪ IBT-BLSA	N/A	N/A	N/A	21,388 (16 × 1,337)
↪ Column Decoder	46,586 (1 × 46,586)	9,434 (1 × 9,434)	9,434 (1 × 9,434)	9,434 (1 × 9,434)
↪ Row Decoder	82,173 (1 × 82,173)	16,640 (1 × 16,640)	16,640 (1 × 16,640)	0 (Base Die)
↪ Sub-Row Dec.	N/A	N/A	N/A	13,074 (1 × 13,074)
↪ Driver + Other	51,578 (1 × 51,578)	10,482 (1 × 10,482)	10,482 (1 × 11,000)	10,482 (1 × 9,000)
↪ GSA	26,205 (1 × 26,205)	5,307 (1 × 5,307)	5,307 (1 × 5,307)	0 (Base Die)
Subtotal	**887,120**	**179,321**	**185,004**	**186,568**
Channel (mm^2)				
↪ # Banks	4	16	16	16
↪ Channel Area	3.55	2.87	2.96	2.99
DRAM Die (mm^2)				
↪ Core Array	56.78	45.91	47.36	47.76
↪ TSV Block	15.2	3.84	4.65	4.92
↪ TSV Block (HB)[3]	15.2	1.13	1.27	4.92
Total Die Area[2]	**72.0**	**49.75**	**52.01**	**52.68**
Total Die Area (HB).3[2]	**N/A**	**47.04**	**48.63**	**52.68**

*HBM2 is the 20 nm baseline; Iso-HBM4 is the 9 nm reference. Subarray/Bank areas in μm^2. Channel/Die areas in mm^2.
**Includes TSV overhead.
***HB refers to thin-pitch hybrid bonding variant.

To perform these calculations, we identify any new structures introduced by these designs (e.g., control logic, LWDs, specialized sense amplifiers) and assign their overhead to either front-end-of-line (FEOL) components composed of transistors and capacitors, or back-end-of-line (BEOL) components consisting of routing wires across multiple metal layers. In some cases, newly added transistors can be "hidden" beneath existing wires, partially offsetting the footprint cost. For replaced blocks, we remove the original footprint and add a new block footprint. When TSV strips are introduced, we account for routing density and KOZs. Tracking each element in its appropriate layer ensures a precise assessment of layout constraints and area impacts. The exact area estimates of these modified components are elaborated in Section 6.5.

6 Results and Analysis

We compare our proposal FB-G-HBM against three baselines: Iso-HBM4, FG-DRAM, and VFG-DRAM. We present both peak irregular bandwidth projections as well as real application performance, along with power and area. We also highlight which design enhancements (folding, grain partitioning, small-atom support, distributed TSVs) contribute to certain performance gains, and indicate where further optimization may yield the greatest impact.

Table 8: Estimated ACTs per tFAW for HBM designs

Design	row fact.	dist fact.	I_{act} (mA)	IDD4 (mA)	bus util	ACTs (cap.)	IDD7 (mA)
Iso-HBM4	1	1	26	585	0.5	8 (8)	500
FG-DRAM	4	1.99	6.5	294	1.0	20 (16)	398
VFG-DRAM	8	1.99	3.25	294	1.0	24 (16)	346
FB-HBM	1	8.07	26	72.5	1.0	20 (16)	488
FB-G-HBM	4	13.4	6.5	33.2	1.0	71 (64)	449

6.1 tFAWs in ACT Window

We estimate the number of simultaneous row activations (ACTs per tFAW) supported by the proposed designs by measuring IDD7 current on HBM3 and then applying scaling factors to the measurements. To compute IDD7 for the new designs, we use the equation:

$$I_{\mathrm{IDD7(ch)}} = \left(\#\mathrm{ACT} \times \frac{I_{\mathrm{act(ch)}}}{\mathrm{row\_factor}}\right) + \left(\mathrm{bus\_util} \times \frac{\mathrm{IDD4(ch)}}{\mathrm{dist\_factor}}\right) \quad (7)$$

In this equation, row_factor represents the ratio of row sizes between the evaluated design and Iso-HBM4. Meanwhile, dist_factor represents the ratio of communicate bits times distance traveled between the evaluated design and Iso-HBM4. The final number of activations per $t_{FAW}$is determined by applying the constraint that each channel must remain within a 500 mA IDD7 limit, reflecting modern HBM per-channel power budgets.

For Iso-HBM4, prior measurements on HBM3 showed an activation current of 34 mA and an I/O current of 770 mA per channel. With technology scaling, these values are estimated to be 26 mA and 585 mA, respectively. For FB-G-HBM, reducing row size by a factor of four reduces activation current to 6.5 mA per ACT command. The intra-bank and inter-bank distances are each reduced 8×, the former by using smaller banks and shorter bitlines, and the latter by distributing TSVs across 16 strips instead of a single TSV block. The interposer remains unchanged, but data movement on the interposer consumes 4.24× less energy than data movement on the DRAM die. These reductions yield a 3.42× improvement in total data movement distance. Combined with a 4× reduction in data volume due to FB-G-HBM's 8B atom size, the final distance factor is 13.7×.

For a 500 mA budget, FB-G-HBM can support 71 ACTs per tFAW before reaching the IDD7 limit. However, since 64 ACTs are sufficient to fully utilize the 32 GB/s channel bandwidth, the final concurrency is capped at 64 ACTs. The tFAW values and reduction factors for the remaining designs are summarized in Table 8.

Even though VFG-DRAM reduces IDD7 current compared to FG-DRAM, it does not offer a higher number of ACTs per tFAW because its channel bandwidth is already saturated. In contrast, FB-G-HBM, which employs an 8B atom size, not only provides additional means to reduce IDD7 current but also enables an effective way to fully exploit the peak channel bandwidth.

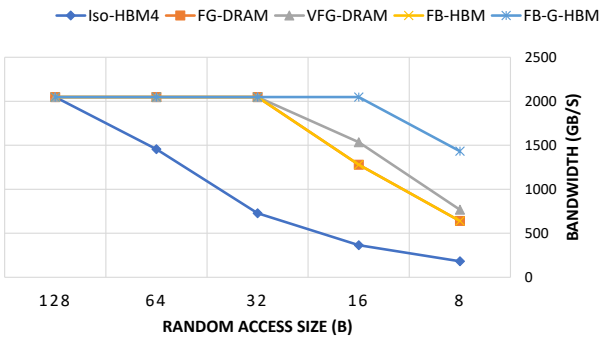

Figure 13: Effective irregular bandwidth for proposed FB-HBM and FB-G-HBM compared against Iso-HBM4, FG-DRAM, and VFG-DRAM for different random access sizes.

6.2 Bandwidth Projections

Figure 13 compares the effective irregular bandwidth for the proposed FB-HBM and FB-G-HBM designs to the baseline designs across different random access sizes (128B to 8B).

Iso-HBM4 maintains peak bandwidth for large random accesses (128B) but exhibits a steady performance decline as access granularity decreases. For 8B accesses, Iso-HBM4 achieves only 201 GB/s, which is 10× lower than its peak provisioned bandwidth (2 TB/s). This degradation stems from its low ACTs per tFAW constraint, preventing it from efficiently handling fine-grained accesses.

FG-DRAM mitigates some Iso-HBM4 inefficiencies by reducing row size, allowing for more activations within the tFAW limit. This results in improved bandwidth at smaller access sizes. Specifically at 8B accesses, FG-DRAM reaches 640 GB/s, providing a 3.2× improvement over Iso-HBM but still 3.2× lower than its peak.

VFG-DRAM follows FG-DRAM closely since both architectures allow the same number of activations per tFAW. However, VFG-DRAM introduces additional parallel partitions, enabling better scheduling flexibility. As a result, it slightly increases bandwidth to 1.13× over FG-DRAM for small accesses.

FB-HBM achieves bandwidth parity with FG-DRAM across all access sizes. Unlike FG-DRAM, which reduces row size, FB-HBM employs a folded bank design to mitigate activation overheads while maintaining Iso-HBM4's larger row size. This demonstrates that FB-HBM can achieve competitive performance with FG-DRAM by reducing data movement distance rather than row size.

FB-G-HBM achieves the highest bandwidth across all access sizes, reaching 1350 GB/s at 8B accesses—6.7× higher than Iso-HBM4. This is enabled by sustaining 64 ACTs per t_{FAW} and increasing parallelism within each channel. Conventional DRAM is constrained by t_{RRD} due to long signaling paths and shared bus circuitry. FB-G-HBM overcomes this by introducing t_{RRD_F}, a significantly smaller timing parameter, enabled by independent TSVs, additional I/O pins, and separate buses that deliver row commands independently to folds without shared circuitry. These enhancements allow FB-G-HBM to fully exploit t_{FAW} while bypassing t_{RRD} constraints, delivering superior performance for fine-grain random accesses.

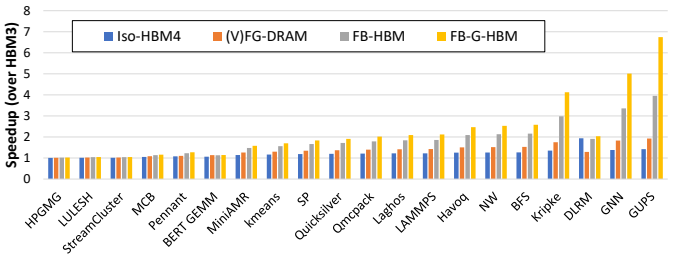

Figure 14: Application speedup over HBM3 for FB-HBM and FB-G-HBM compared against Iso-HBM4 and (V)FG-DRAM.

6.3 Application Speedup

Figure 14 compares the application-level performance of FB-G-HBM and other HBM4 designs relative to HBM3.

FB-G-HBM achieves a 6.74× speedup in the GUPS benchmark—2.53× over FG-DRAM and 2.24× over VFG-DRAM. FG-DRAM itself achieves a speedup of 1.87× over Iso-HBM4.

Thus, 27% of the overall speedup achieved by FB-G-HBM can be attributed to the grain feature adopted from FG-DRAM. To breakdown FB-G-HBM's performance benefits further, we analyzed GUPS using our previously introduced equations and determined that the newly introduced folded banks design and small-atom support is responsible for the remaining 73% of the achieved speedup.

The CORAL-2 benchmark suite, which includes a mix of structured and irregular scientific computing applications, experiences performance gains in the range of 2.0× to 2.5× with FB-G-HBM. Notably, Havoq (a parallel graph analytics algorithm), Laghos (an Euler equation solver), LAMMPS (a molecular dynamics simulator), QMCPACK (a Monte Carlo solver), and QuickSilver (a particle transport solver with heavy indexed lookups) all benefit from FB-G-HBM's irregular bandwidth enhancements. PENNANT, an unstructured mesh solver, achieves a 1.27× speedup, slightly lower than other irregular workloads. Kripke, a structured yet high-dimensional solver, exhibits a 4.13× speedup, leveraging FB-G-HBM's ability to efficiently manage multidimensional array accesses.

DLRM represents a different class of irregular workloads, where embedding bag operations result in coarse-grained but irregular memory accesses. While its irregular access pattern makes it a candidate for FB-G-HBM's optimizations, its accesses remain wider compared to GUPS or CORAL-2 applications.

As a result, FB-G-HBM achieves a speedup of 2×, which is comparable to FG-DRAM and VFG-DRAM. GNNs present an even more compelling use case, with FB-G-HBM achieves a 5.02× speedup for GNNs, compared to 2.42× with FG-DRAM and 2.69× with VFG-DRAM.

Finally, even regular workloads benefit from FB-G-HBM's optimizations. A representative GEMM kernel from the BERT deep learning model achieves a 1.14× performance improvement over Iso-HBM4. This relatively modest gain arises due to the interleaving of regular requests from massively parallel cores, which makes the application behave similarly to an irregular workload from the memory controller's perspective. A similar trend is observed for STREAM.

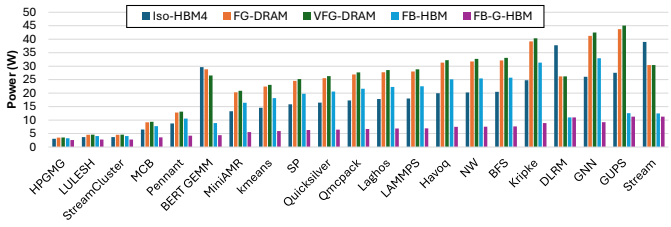

Figure 15: Power for FB-HBM and FB-G-HBM compared against Iso-HBM4, FG-DRAM, and VFG-DRAM.

6.4 Power Projections

Figure 15 reports total stack power for eighteen benchmarks; we use the two end-points—STREAM (fully regular) and GUPS (fully irregular)—to illustrate the key trends.

Iso-HBM4 sustains its advertised 2 TB/s on STREAM but consumes 39 W, exceeding the ∼30 W thermal envelope of an HBM stack. This highlights the limitations of conventional "wider-bus, faster-clock" scaling: even regular traffic saturates the thermal budget due to high data-movement costs. FG-DRAM lowers power to 30.4 W via distributed TSVs but still approaches the upper thermal bound. FB-HBM significantly reduces power to 12.5 W by minimizing data-movement energy, and FB-G-HBM pushes it further down to 11.3 W. Both fall comfortably within thermal limits while sustaining peak bandwidth.

For GUPS, Iso-HBM4 consumes only 27.6 W, but this is not due to efficiency—it stems from enforcing a strict cap of 8 ACTs per t_{FAW}, which throttles concurrency and limits bandwidth to just 201 GB/s. As a result, energy efficiency remains poor at 0.137 J/GB.

FG-DRAM improves bandwidth to 640 GB/s (3.2×) but increases power to 43.8 W, exceeding the stack's thermal budget. VFG-DRAM pushes bandwidth further to 723.2 GB/s, but power rises to 45.1 W, yielding only marginal efficiency gains (0.062 J/GB).

FB-HBM relaxes the t_{FAW} constraint through folded banks, raising GUPS bandwidth by 3.92× relative to Iso-HBM4. While its bandwidth is comparable to FG-DRAM, it lowers the energy cost of data movement, achieving the result with only 12.55 W. FB-G-HBM combines folded banks with an 8-byte atom and grain architecture, enabling higher 8B random bandwidth while maintaining a power level comparable to FB-HBM.

6.5 Area Projections

Both FG-DRAM and FB-G-HBM adopt a grain-based subarray architecture that increases the number of LWDs from 19 (including those for ECC mats) to 22 for each set of four grains, with each LWD occupying roughly 19.4 μm^2. As a result, the subarray area grows from 4042.94 μm^2 in Iso-HBM4 to 4101.14 μm^2, a 1.4% overhead, as summarized in Table 7.

Because subarrays constitute about 60% of the DRAM die, this translates to approximately 0.8% at the full-die level. A variant with eight grains, denoted as VFG-DRAM, pushes the LWD count even higher (to 26), raising the subarray area to 4178.74 μm^2, or 3.4% above Iso-HBM4—equivalent to about 2.0% overhead at the die level.

FG-DRAM adds three new transistors per subarray to enable grain selection, introducing a 0.6% transistor overhead, plus four additional `GrSel` wires on top of the existing 128 `MWL`, four `LWLSel`, and 16 `LDL` lines, resulting in a 2.7% bank-height overhead. By contrast, FB-G-HBM places the grain-selection logic on a separate base die, thus avoiding the 0.6% transistor overhead. Although it replaces four `LWLSel` lines with eight `LWDEn` lines, the net addition is still four new wires, retaining the same 2.7% wiring overhead. This yields an overall 0.6% lower area cost for FB-G-HBM compared to FG-DRAM. For eight-grain configurations, such as VFG-DRAM or FB-G-HBM with eight grains, the wire overhead doubles to eight new lines (5.4%), while the transistor overhead remains 0.6% if located in the main DRAM die or zero if moved off-die.

Furthermore, FB-G-HBM replaces the two dummy subarrays per bank with two IBT-BLSAs to reduce intra-bank data movement by 8× and mitigate the performance penalty associated with global signaling. While each IBT-BLSA occupies about 1293 μm^2, replicating these across eight dies, twice per die, yields a total IBT-BLSA die footprint of $8 \times 2 \times 1293$ μm^2 = 20,688 μm^2. In contrast, FG-DRAM and Iso-HBM4 both retain two full dummy subarrays, costing 2×4042 μm^2 = 8084 μm^2 per bank.

FB-H-HBM relocates multiple logic components to the base die, thereby reducing the area overhead on the main DRAM die. Specifically, FB-G-HBM moves its GSAs off the DRAM die, saving 5,307 μm^2. In addition, it further modifies the row decoder by introducing a sub-row decoder and migrates the remaining decode circuitry to the base die, yielding another 3,657 μm^2 in area savings.

TSV structures are updated with changes to count, dimensions and distribution. Vertical wiring requires 168 connections per bank (128 `MDLs`, 16 `CSLs`, 16 CAs, 8 `LWDEns`), totaling 172,032 connections. At 40,000 connections per mm^2 [11, 30], this needs $4.92mm^2$, including the costs of the KOZ. The corresponding TSV areas for Iso-HBM4 and FGDRAM are $3.84mm^2$ and $4.65mm^2$ respectively assuming nominal TSV pitch scaling. While these designs also benefit from hybrid bonding and thin-pitch TSVs, scaling the pitch beyond 18um is infeasible as these designs need to transmit information at 16Gbps instead of FB-G-HBM's 1Gbps. Nevertheless, adopting hybrid bonding and thin-pitch TSVs reduce this cost to $1.13mm^2$ and $1.27mm^2$. Table 7 summarizes our area estimates.

I/O and PHY. Normally, implementing an 8B atom with a 4x increase in command rate would require a proportional increase in CA/RA pins to accommodate the row and column addresses. However, HBM3 overprovisions these pins by a factor slightly greater than two. By reassigning a subset of CA pins originally intended for column addresses to row addresses, we further reduce the PHY area overhead. For instance, HBM3 uses 120 pins per channel. We assume the baseline HBM4 design reduces this to 72 pins, with 32 for data, 8 for column command/address, 10 for row command/address, and 22 for auxiliary functions. The FB-G-HBM architecture requires additional signaling to support small read/write operations: two address bits each for row and column grain selection, and one bit for the encoded SM_RD or SM_WR command. To accommodate the 4-fold increase in row command frequency and support parallel access across four grains, the design reallocates one column address pin and adds four row address pins. As a result the enhanced functionality of the final FB-G-HBM design requires 76 pins per channel – a modest increase of only four extra pins.

7 Related work

Multiple approaches attempt to scale down DRAM rows and activation power. Half-DRAM [42] divides each wordline into two halves and activates one half in odd mats and the other half in even mats. Half Page Row [13] modifies the wordline connections within a bank to halve the number of cells fetched to the row buffer to save energy. Cooper-Balis et al. [7] use the posted CAS command to enable a finer-grained selection when activating a portion of the DRAM array. Selective bitline activation [38] waits for both row and column address signals to arrive before activating only the bitlines of the requested cache line. The partial row activation [25] scheme minimizes row activation granularity for memory writes while retaining the read bandwidth of the conventional DRAM. Sectored DRAM [32] is a DRAM substrate that enables fine-grained access and activation by predicting the words in a cache block to be accessed during cache residency and transferring only the predicted words on the memory channel. It also activates a smaller set of cells that contain the predicted words.

Among the techniques to improve irregular memory parallelism, the DRAM channel partitioning [9] allows for more fine-grained irregular accesses. However, recent tests on irregular applications [14] showed that performance of this scheme is limited by data layout and workload imbalance issues. The Subarray-Level Parallelism [20] helps mitigate the negative impact of bank serialization by overlapping bank access latencies of multiple requests that go to different subarrays within the same bank. Our approach instead focuses on increasing the total number of banks by vertically spanning them across DRAM dies, with independent activates across each die. We go beyond the 3D-Stacked Memory Architecture [27] approach and propose spanning banks across the memory dies in HBM. FIGARO [41] enables small, frequently-accessed portions of DRAM rows to be cached in a designated region of DRAM. We instead target irregular, poorly-cached applications.

Several papers focused on improving performance of irregular codes from compute [26, 34, 39, 40], interconnect [10], memory controller [36], and data reorganization [4] perspective.

In contrast, our approach focuses on reducing data movement power. We propose spanning banks across memory dies in HBM and going beyond existing 3D-stacked approaches to increase the total number of banks independently activated within each die. Additionally, to reduce DRAM row size, we creatively leverage the base die to implement many DRAM circuits.

8 Conclusion

Random, irregular bandwidth is emerging as a bottleneck for many data-centric workloads. While continued shrinking of HBM row size is helpful, it is insufficient on its own. The full benefit comes only when row-size reduction is paired with architectural techniques that shorten the physical distance that data must travel.

This paper presented **Folded-Bank HBM (FB-HBM)**, a family of 3-D-stacked memory that couples fine-grained activation with folded banks connected through hybrid bonding. The proposed organization increases independent row activations by 8× and boosts GUPS bandwidth by 6.74× versus a traditionally-scaled HBM4 baseline. In detailed cycle-level simulations of graph neural-network kernels, these gains translate to a 5× performance uplift while driving per-bit energy for regular AI traffic below the 1 pJ threshold. Area-efficient refinements keep the cost of these improvements modest. Replacing dummy edge subarrays with self-timed sense amplifiers reduces fold overhead from 25% to 8%, and relocating row-segmentation logic to the base die trims storage-die area from 3.2% to 1.5%.

Although the benefits are substantial, several issues must be resolved before FB-HBM can be deployed at scale. Our design requires thinned dies <30 µm or stronger on-die drivers. Maintaining yield in such thinned without negating the area savings remains an open manufacturing challenge. In addition, small-atom activation and folded-bank command support are absent from current JEDEC HBM specifications, and the rest of the SoC (e.g., caches, on-chip interconnects) lacks the support for the finer access granularity. Our design also requires a custom base die, but the cost can be justified by the energy-efficiency gains on AI workloads.

We encourage the JEDEC HBM task groups to standardise sub-row activation primitives and folded-bank commands (`SM_RD/SM_WR`), and invite vendors and foundries to prototype yield-robust DRAM- and base-die options. Addressing these practical hurdles will allow the industry to fully capitalize on FB-HBM's ability to deliver high random bandwidth at low energy, meeting the needs of next-generation AI and graph analytics workloads.

References

[1] [n. d.]. CORAL-2 Benchmarks. https://asc.llnl.gov/coral-2-benchmarks

[2] [n. d.]. High Bandwidth Memory DRAM (HBM3). https://www.jedec.org/system/files/docs/JESD238.pdf

[3] Sriram Aananthakrishnan, Nesreen K. Ahmed, Vincent Cave, Marcelo Cintra, Yigit Demir, Kristof Du Bois, Stijn Eyerman, Joshua B. Fryman, Ivan Ganev, Wim Heirman, Hans-Christian Hoppe, Jason Howard, Ibrahim Hur, MidhunChandra Kodiyath, Samkit Jain, Daniel S. Klowden, Marek M. Landowski, Laurent Montigny, Ankit More, Przemyslaw Ossowski, Robert Pawlowski, Nick Pepperling, Fabrizio Petrini, Mariusz Sikora, Balasubramanian Seshasayee, Shaden Smith, Sebastian Szkoda, Sanjaya Tayal, Jesmin Jahan Tithi, Yves Vandriessche, and Izajasz P. Wrosz. 2020. PIUMA: Programmable Integrated Unified Memory Architecture. https://arxiv.org/abs/2010.06277

[4] Berkin Akin, Franz Franchetti, and James C. Hoe. 2015. Data Reorganization in Memory Using 3D-Stacked DRAM. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture*.

[5] Niladrish Chatterjee, Mike O'Connor, Donghyuk Lee, Daniel R. Johnson, Stephen W. Keckler, Minsoo Rhu, and William J. Dally. [n. d.]. Architecting an Energy-Efficient DRAM System for GPUs. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*.

[6] Pai-Yu Chen, Xiaochen Peng, and Shimeng Yu. 2018. NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 12 (2018), 3067–3080.

[7] Elliott Cooper-Balis and Bruce Jacob. 2010. Fine-Grained Activation for Power Reduction in DRAM. *IEEE Micro* 30 (2010).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bi-directional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Timothy Dysart, Peter Kogge, Martin Deneroff, Eric Bovell, Preston Briggs, Jay Brockman, Kenneth Jacobsen, Yujen Juan, Shannon Kuntz, Richard Lethin, Janice McMahon, Chandra Pawar, Martin Perrigo, Sarah Rucker, John Ruttenberg, Max Ruttenberg, and Steve Stein. [n. d.]. Highly Scalable Near Memory Processing with Migrating Threads on the Emu System Architecture. In *2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3)*.

[10] Marjan Fariborz, Mahyar Samani, Pouya Fotouhi, Roberto Proietti, Il-Min Yi, Venkatesh Akella, Jason Lowe-Power, Samuel Palermo, and S. J. Ben Yoo. 2022. LLM: Realizing Low-Latency Memory by Exploiting Embedded Silicon Photonics for Irregular Workloads. In *High Performance Computing: 37th International Conference, ISC High Performance 2022, Hamburg, Germany, May 29 – June 2, 2022, Proceedings*. 44–64.

[11] Bai Fujun, Jiang Xiping, Wang Song, Yu Bing, Tan Jie, Zuo Fengguo, Wang Chunjuan, Wang Fan, Long Xiaodong, Yu Guoqing, et al. 2020. A Stacked Embedded DRAM Array for LPDDR4/4X Using Hybrid Bonding 3D Integration with 34GB/s/1Gb 0.88 pJ/b Logic-to-Memory Interface. In *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 6–6.

[12] Oded Green, James Fox, Jeffrey Young, Jun Shirako, and David Bader. 2019. Performance Impact of Memory Channels on Sparse and Irregular Algorithms.

[13] Heonjae Ha, Ardavan Pedram, Stephen Richardson, Shahar Kvatinsky, and Mark Horowitz. 2016. Improving Energy Efficiency of DRAM by Exploiting Half Page Row Access. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[14] Eric Hein, Srinivas Eswar, Abdurrahman Yasar, Jiajia Li, Jeffrey S. Young, Thomas M. Conte, Umit V. Catalyurek, Rich Vuduc, Jason Riedy, and Bora Ucar. 2019. Programming Strategies for Irregular Algorithms on the Emu Chick.

[15] Michihiro Inoue, TOSHIO Yamada, HISAKAZU Kotani, HIROYUKI Yamauchi, ATSUSHI Fujiwara, JUNK Matsushima, HIRONORI Akamatsu, MASANORI Fukumoto, MASAFUMI Kubota, ICHIRO Nakao, et al. 1988. A 16-Mbit DRAM with a Relaxed Sense-Amplifier-Pitch Open-Bit-Line Architecture. *IEEE journal of solid-state circuits* 23, 5 (1988), 1104–1112.

[16] JEDEC Solid State Technology Association. 2023. *High Bandwidth Memory DRAM (HBM3)*. JEDEC Standard JESD238A. JEDEC Solid State Technology Association. Revision of JESD238, January 2022.

[17] Yoshihisa Kagawa, Takumi Kamibayashi, Yuriko Yamano, Kenya Nishio, Akihisa Sakamoto, Taichi Yamada, Kan Shimizu, Tomoyuki Hirano, and Hayato Iwamoto. 2022. Development of Face-to-Face and Face-to-Back Ultra-Fine Pitch Cu-Cu Hybrid Bonding. In *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*. IEEE, 306–311.

[18] Joohee Kim, Jun So Pak, Jonghyun Cho, Eakhwan Song, Jeonghyeon Cho, Heegon Kim, Taigon Song, Junho Lee, Hyungdong Lee, Kunwoo Park, et al. 2011. High-frequency scalable electrical model and analysis of a through silicon via (TSV). *IEEE Transactions on Components, Packaging and Manufacturing Technology* 1, 2 (2011), 181–195.

[19] Suk Min Kim, Byungkyu Song, and Seong-Ook Jung. 2021. Imbalance-Tolerant Bit-Line Sense Amplifier for Dummy-less Open Bit-line Scheme in DRAM. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68, 6 (2021), 2546–2554.

[20] Yoongu Kim, Vivek Seshadri, Donghyuk Lee, Jamie Liu, and Onur Mutlu. 2012. A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 368–379.

[21] Yoongu Kim, Weikun Yang, and Onur Mutlu. 2015. Ramulator: A Fast and Extensible DRAM Simulator. *IEEE Computer architecture letters* 15, 1 (2015), 45–49.

[22] John H Lau. 2022. Recent Advances and Trends in Advanced Packaging. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 12, 2 (2022), 228–252.

[23] Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri, and Onur Mutlu. 2016. Understanding and Exploiting Design-Induced Latency Variation in Modern DRAM Chips. *arXiv preprint arXiv:1610.09604* (2016).

[24] Dong Uk Lee, Kang Seol Lee, Yongwoo Lee, Kyung Whan Kim, Jong Ho Kang, Jaejin Lee, and Jun Hyun Chun. 2015. Design Considerations of HBM Stacked DRAM and the Memory Architecture Extension. In *2015 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 1–8.

[25] Yebin Lee, Hyeonggyu Kim, Seokin Hong, and Soontae Kim. 2017. Partial Row Activation for Low-Power DRAM System. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*.

[26] Yuxi Liu, Xia Zhao, Magnus Jahre, Zhenlin Wang, Xiaolin Wang, Yingwei Luo, and Lieven Eeckhout. [n. d.]. Get Out of the Valley: Power-Efficient Address Mapping for GPUs. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*.

[27] Gabriel H. Loh. [n. d.]. 3D-Stacked Memory Architectures for Multi-core Processors. In *2008 International Symposium on Computer Architecture*.

[28] O. Naji, C. Weis, M. Jung, N. Wehn, and A. Hansson. 2015. A High-Level DRAM Timing, Power and Area Exploration Tool. In *2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. 149–156. doi:10.1109/SAMOS.2015.7363670

[29] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *arXiv preprint arXiv:1906.00091* (2019).

[30] Dimin Niu, Shuangchen Li, Yuhao Wang, Wei Han, Zhe Zhang, Yijin Guan, Tianchan Guan, Fei Sun, Fei Xue, Lide Duan, et al. 2022. 184QPS/W 64Mb/mm^2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65. IEEE, 1–3.

[31] James Michael O'Connor. 2021. *Energy Efficient High Bandwidth DRAM for Throughput Processors*. Ph. D. Dissertation.

[32] Ataberk Olgun, F. Nisa Bostanci, Geraldo F. Oliveira, Yahya Can Tugrul, Rahul Bera, A. Giray Yaglikci, Hasan Hassan, Oguz Ergin, and Onur Mutlu. 2022. Sectored DRAM: An Energy-Efficient High-Throughput and Practical Fine-Grained DRAM Architecture.

[33] Mike O'Connor, Niladrish Chatterjee, Donghyuk Lee, John Wilson, Aditya Agrawal, Stephen W. Keckler, and William J. Dally. 2017. Fine-Grained DRAM: Energy-Efficient DRAM for Extreme Bandwidth Systems. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[34] Subhankar Pal, Swagath Venkataramani, Viji Srinivasan, and Kailash Gopalakrishnan. 2022. OnSRAM: Efficient Inter-Node On-Chip Scratchpad Management in Deep Learning Accelerators. *ACM Trans. Embed. Comput. Syst.*, Article 86 (oct 2022).

[35] Myeong-Jae Park, Jinhyung Lee, Kyungjun Cho, Jihwan Park, Junil Moon, Sung-Hak Lee, Tae-Kyun Kim, Sanghoon Oh, Seokwoo Choi, Yongsuk Choi, et al. 2022. A 192-Gb 12-high 896-GB/s HBM3 DRAM with a TSV auto-calibration scheme and machine-learning-based layout optimization. *IEEE Journal of Solid-State Circuits* 58, 1 (2022), 256–269.

[36] Vivek Seshadri, Thomas Mullins, Amirali Boroumand, Onur Mutlu, Phillip B Gibbons, Michael A. Kozuch, and Todd C Mowry. 2015. Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-Unit Strided Accesses. In *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[37] Tsugio Takahashi, Tomonori Sekiguchi, Riichiro Takemura, Seiji Narui, Hiroki Fujisawa, Shinichi Miyatake, Makoto Morino, Koji Arai, Satoru Yamada, Shoji Shukuri, et al. 2001. A Multigigabit DRAM Technology with $6F^2$ Open-Bitline Cell, Distributed Overdriven Sensing, and Stacked-Flash Fuse. *IEEE Journal of Solid-State Circuits* 36, 11 (2001), 1721–1727.

[38] Aniruddha N. Udipi, Naveen Muralimanohar, Niladrish Chatterjee, Rajeev Balasubramonian, Al Davis, and Norman P. Jouppi. 2010. Rethinking DRAM Design and Organization for Energy-Constrained Multi-Cores. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*.

[39] Haonan Wang and Adwait Jog. [n. d.]. Exploiting Latency and Error Tolerance of GPGPU Applications for an Energy-Efficient DRAM. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.

[40] Xi Wang, Antonino Tumeo, John D. Leidel, Jie Li, and Yong Chen. 2019. MAC: Memory Access Coalescer for 3D-Stacked Memory. In *Proceedings of the 48th International Conference on Parallel Processing*.

[41] Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu. 2020. FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching.

[42] Tao Zhang, Ke Chen, Cong Xu, Guangyu Sun, Tao Wang, and Yuan Xie. [n. d.]. Half-DRAM: A High-Bandwidth and Low-Power DRAM Architecture from the Rethinking of Fine-Grained Activation. In *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*.

[43] Ting Zheng and Muhannad S Bakir. 2022. Benchmarking frequency-dependent parasitics of fine-pitch off-chip I/Os for 2.5 D and 3D heterogeneous integration. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 12, 12 (2022), 2002–2012.