UNIVERSITY OF CALIFORNIA
Los Angeles

A Modeling Framework for Architecture-Level Exploration of High-Bandwidth Memory Systems

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Chanhee Lee

2026

© Copyright by
Chanhee Lee
2026

ABSTRACT OF THE THESIS

A Modeling Framework for
Architecture-Level Exploration of
High-Bandwidth Memory Systems

by

Chanhee Lee
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2026
Professor Puneet Gupta, Chair

High Bandwidth Memory (HBM) has become a critical technology for modern high-performance computing systems due to the rapidly increasing memory bandwidth demand in data-intensive workloads such as artificial intelligence and large-scale data processing. By vertically stacking multiple DRAM dies using through-silicon vias (TSVs), HBM significantly improves memory bandwidth and energy efficiency compared to conventional DRAM architectures. However, the increasing complexity of stacked memory structures introduces new challenges in early-stage design exploration, including accurate modeling of memory die organization, TSV allocation, thermal constraints, and power delivery.

In this thesis, we propose a flexible modeling framework for architecture-level exploration of HBM-based memory systems. The proposed framework models key physical and architectural components of HBM, including memory die structure, peripheral circuits, base die logic, and TSV interconnects. In addition, the framework incorporates thermal TSV allocation and energy-efficiency estimation to capture important design trade-offs in stacked memory architectures. By leveraging configurable architectural parameters, the model enables rapid exploration of design choices across different HBM generations.

Using the proposed framework, we analyze the architectural scaling trends of HBM2,

HBM3, and emerging HBM4 designs. We further conduct multiple case studies to evaluate the impact of thermal constraints, TSV allocation strategies, and power delivery requirements on stacked memory systems. Experimental results demonstrate that the proposed model can effectively capture the design trade-offs among area, power, thermal constraints, and memory bandwidth in modern HBM architectures.

The proposed framework provides a useful tool for early-stage design exploration of next-generation HBM systems and enables systematic evaluation of architectural trade-offs in future 3D-stacked memory technologies.

The thesis of Chanhee Lee is approved.

Nader Sehatbakhsh

Subramanian Srikanteswara Iyer

Puneet Gupta, Committee Chair

University of California, Los Angeles

2026

To my family, especially my parents and my wife

TABLE OF CONTENTS

1 Introduction **1**

1.1 Memory Bandwidth Bottleneck 1

1.2 Emergence of High-Bandwidth Memory (HBM) 2

1.3 Challenges in HBM Design 3

1.4 Motivation 4

1.5 Thesis Contributions 5

1.6 Thesis Organization 5

2 Background and Related Work **6**

2.1 DRAM and HBM Architecture 6

2.2 Through-Silicon Via Technology 7

2.3 Existing Memory Modeling Tools 7

2.4 Limitations of Existing Approaches 8

3 Proposed HBM Modeling Framework **10**

3.1 Modeling Overview 10

3.2 Memory Die Modeling 12

3.2.1 Cell Array 13

3.2.2 Peripheral Circuits 15

3.2.3 Latency 19

3.3 Base Die Modeling 21

3.4 TSV Modeling 24

3.5 Energy and Thermal Modeling 28

3.5.1 Energy Efficiency 28
3.5.2 Thermal Modeling 31

4 Model Comparison and Analysis 36
4.1 Experimental Setup 36
4.2 Cross-Generation Analysis of HBM2, HBM3, and HBM4 37
4.3 Comparison with Existing Models 38

5 Case Studies 42
5.1 Bandwidth Scaling, TSV Demand, and Die-Area Impact 42
5.2 TSV Pitch, TSV Allocation, and Area Trade-offs 45
5.3 Logic Stacking as a System-Level Case Study 46
5.4 Stack Height Scaling 49

6 Conclusion and Future Work 52
6.1 Summary 52
6.2 Future Work 54

References 57

LIST OF FIGURES

1.1 Comparison of conventional GDDR memory and HBM in terms of structure and interface width. Unlike GDDR, which relies on a narrower off-package interface, HBM achieves much higher bandwidth through an extremely wide I/O bus and vertically stacked DRAM dies interconnected by TSVs. Adapted from [1]. 2

1.2 Representative HBM architecture. Multiple DRAM dies are vertically stacked using TSVs on top of a logic base die, and the memory stack communicates with the processor through a silicon interposer, enabling a wide I/O interface and high memory bandwidth. 3

3.1 Overall flow of the proposed HBM modeling framework. User-defined architectural and technology parameters are provided as inputs to the memory-die, base-die, and TSV/thermal/energy models, whose outputs are then integrated to estimate the final HBM stack metrics. The framework can also incorporate optional stacked-logic loading through additional power and hotspot inputs. 10

3.2 Hierarchical organization of the HBM memory die assumed in the proposed model. Each HBM stack consists of multiple memory dies stacked on a base die, and each memory die is organized into channels, pseudo-channels, banks, mats, and subarrays. 12

3.3 Representative TSV array layouts: simple-grid and staggered patterns. 27

5.1 Bandwidth scaling versus TSV demand across HBM generations. 43

5.2 Impact of TSV scaling on memory die area for the HBM3 baseline configuration. 44

5.3 Maximum bandwidth that can be supported before the memory die area exceeds the current modeled base-die footprint. 45

5.4 Effect of TSV pitch on thermal TSV count, TSV array area, and memory die area for the HBM3 baseline. 46

5.5 Conceptual comparison of two logic-stacking scenarios. GPU-on-HBM primarily stresses power delivery through the memory stack, while HBM-on-GPU increases thermal resistance along the vertical heat-removal path. Adapted from [2]. 47

5.6 Power TSV and thermal TSV components under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4. 48

5.7 Total TSV demand under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4. 49

5.8 Minimum required memory die area under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4. The dashed line denotes the fixed base-die footprint limit. 50

5.9 Impact of stack height on TSV demand, memory die area, and worst-case latency for the HBM3 baseline configuration. 51

LIST OF TABLES

3.1 Parameters and symbols for the memory-die cell-array area model. 14

3.2 Parameters and symbols for the mat-level peripheral area model. The term F_{mem} is referenced from Table 3.1. 16

3.3 Parameters and symbols for the bank-level peripheral area model. The term $A_{\text{mat,peri}}$ is referenced from Table 3.2. 18

3.4 Parameters and symbols for the memory-die latency model. 20

3.5 Parameters and symbols for the energy and thermal model used in TSV budgeting. 30

3.6 Parameters and symbols for the TSV budgeting and footprint model. The term η_{mem} is referenced from Table 3.5. 35

4.1 Cross-generation summary of modeled HBM2, HBM3, and HBM4 results. 37

4.2 Validation against published reference data for HBM2 [3] and HBM3 [4], with DreamRAM results [5] included where directly comparable. HBM4 is excluded because its results are projection-based rather than directly validated against measured silicon. 39

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Professor Puneet Gupta, for his continuous guidance, support, and encouragement throughout my graduate studies. His insight and mentorship have greatly influenced my research and helped me develop as an independent researcher.

I would also like to thank Dr. Jooyeon Jeong for her valuable guidance and helpful discussions during the course of this research.

I am grateful to my labmates for creating a collaborative and supportive research environment. I would like to thank Alexander Graening, George Karfakis, Zhichao Chen, Dedeepyo Ray, Aaron Yen, Ernest Wang, Lime Yao, Tianrun Gou, Jialong Chen, and Binglu Chen for their discussions, support, and friendship during my time in the lab.

Finally, I would like to express my deepest gratitude to my family. I am especially thankful to my parents for their unconditional love, encouragement, and sacrifices that have supported me throughout my life. I am deeply grateful to my wife, Jiwon Min, for her unwavering support, patience, and encouragement during this journey. I also look forward to welcoming our daughter, Allyson Lee, who will soon join our family and bring even greater joy and motivation to our lives.

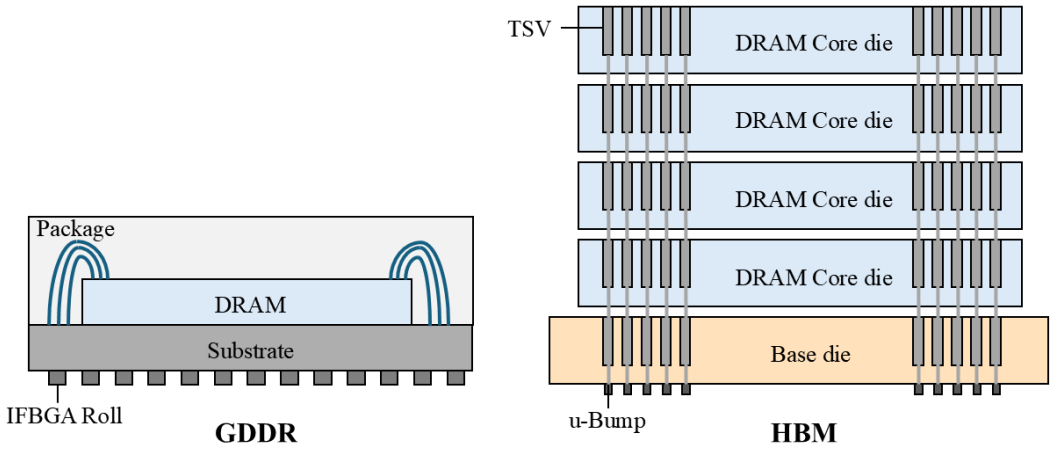
CHAPTER 1

Introduction

1.1 Memory Bandwidth Bottleneck

Modern computing systems increasingly face the memory bandwidth bottleneck, where the rate at which processors can access data from memory becomes a primary limitation to overall system performance. Over the past decades, processor performance has continued to improve significantly due to advances in transistor scaling and architectural innovations. However, the improvement in memory bandwidth has not kept pace with the growth in compute capability. As a result, many modern workloads are increasingly limited by memory access rather than computational throughput.

The demand for higher memory bandwidth has become particularly critical in emerging data-intensive applications such as artificial intelligence, machine learning, scientific computing, and large-scale data analytics. These applications require frequent access to large volumes of data, placing substantial pressure on the memory subsystem. Conventional memory architectures based on Dynamic Random-Access Memory (DRAM) struggle to provide sufficient bandwidth due to limitations in pin count, signaling frequency, and package-level interconnect constraints. Consequently, the gap between processor performance and memory bandwidth, often referred to as the *memory wall*, continues to widen. Figure 1.1 illustrates this limitation by comparing conventional GDDR memory with HBM. Unlike GDDR, which relies on a relatively narrow off-package bus, HBM achieves much higher aggregate bandwidth through an extremely wide I/O interface and vertically stacked memory dies.



Package	GDDR6X	HBM4
Data rate [Gbps]	24	8.0
Bus width [bits]	32	2048
Bandwidth [GB/s]	96	2048

Figure 1.1: Comparison of conventional GDDR memory and HBM in terms of structure and interface width. Unlike GDDR, which relies on a narrower off-package interface, HBM achieves much higher bandwidth through an extremely wide I/O bus and vertically stacked DRAM dies interconnected by TSVs. Adapted from [1].

1.2 Emergence of High-Bandwidth Memory (HBM)

High Bandwidth Memory (HBM) has emerged as a promising solution to address the memory bandwidth limitations of conventional memory technologies. Unlike traditional off-package DRAM modules, HBM adopts a three-dimensional (3D) stacked memory architecture in which multiple DRAM dies are vertically integrated using TSVs and placed on a logic base die [6]. As illustrated in Figure 1.2, the resulting memory stack communicates with the processor through a silicon interposer, enabling a very wide I/O interface and a much higher aggregate bandwidth than conventional DRAM organizations. The logic base die manages memory access and communication between the stacked DRAM layers and the external system [7].

By leveraging 3D integration and wide I/O interfaces [8], HBM can achieve significantly higher memory bandwidth compared to conventional DRAM architectures while maintain-

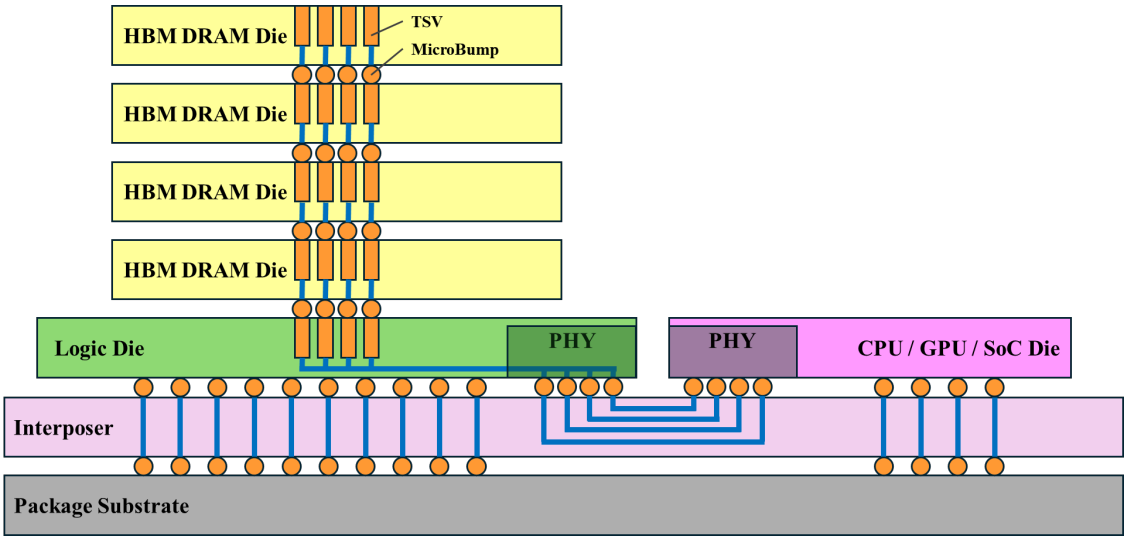


Figure 1.2: Representative HBM architecture. Multiple DRAM dies are vertically stacked using TSVs on top of a logic base die, and the memory stack communicates with the processor through a silicon interposer, enabling a wide I/O interface and high memory bandwidth.

ing improved energy efficiency. Furthermore, the close physical integration between memory stacks and compute devices enables shorter interconnect distances and reduced signaling overhead. The structural features of HBM, including vertical die stacking, TSV-based interconnects, and interposer-based integration, together explain why it can provide substantially greater bandwidth density than conventional memory organizations. As a result, HBM has become a key technology for high-performance computing systems, particularly in graphics processing units (GPUs), accelerators for artificial intelligence workloads, and data center processors. This architectural shift toward vertically stacked memory has enabled substantial improvements in memory bandwidth density while maintaining energy efficiency.

1.3 Challenges in HBM Design

Despite its advantages, HBM introduces several new challenges in system design due to its vertically stacked architecture and dense interconnect structures. The integration of multiple DRAM dies using TSVs creates complex interactions among thermal behavior,

power delivery, and signal integrity. For example, the increased power density in stacked memory dies can lead to thermal hotspots that degrade device reliability and performance. In addition, TSV resources must be carefully allocated across signal transmission, power delivery, and thermal management in order to achieve a balanced and efficient design.

Another major challenge lies in accurately modeling the architectural and physical characteristics of HBM systems during early-stage design exploration. Existing memory modeling tools provide useful foundations, but their support for HBM-specific physical analysis remains limited. For instance, CACTI-3DD models TSV-related overheads in terms of area, power, and timing, while DreamRAM incorporates TSVs as part of the inter-bank routing structure together with global buses and data paths [5, 9]. However, these approaches do not fully capture the broader physical design trade-offs required in HBM, such as die-level organization, explicit TSV budgeting, and thermal feasibility across different generations and design points.

As a result, it remains difficult to systematically analyze how architectural choices and physical constraints interact in modern HBM systems. This limitation motivates the need for a modeling framework that can jointly consider memory die organization, TSV allocation, power delivery, and thermal constraints in a unified manner.

1.4 Motivation

The increasing complexity of HBM architectures motivates the need for a flexible modeling framework that enables early-stage exploration of design trade-offs in stacked memory systems. Such a framework should capture key physical and architectural characteristics of HBM, including memory die organization, base die functionality, TSV interconnect structures, and thermal constraints.

A comprehensive modeling approach can provide valuable insights into the design space of next-generation HBM systems and support the evaluation of architectural decisions across different HBM generations. By enabling systematic exploration of design parameters, such

models can assist researchers and designers in understanding the interactions between area, power, thermal behavior, and memory bandwidth in 3D-stacked memory architectures.

1.5 Thesis Contributions

This thesis presents a modeling framework for architectural exploration of HBM memory systems. The proposed framework models key components of HBM architectures, including memory die structures, peripheral circuits, base die logic, and TSV interconnects. In addition, the framework incorporates thermal considerations and energy-efficiency estimation to capture important design trade-offs in stacked memory systems.

Using the proposed modeling framework, this work analyzes the architectural scaling trends of HBM2, HBM3, and emerging HBM4 designs. Multiple case studies are conducted to investigate the impact of TSV allocation strategies, thermal constraints, and power delivery considerations on stacked memory architectures. The results demonstrate how the proposed model can be used to systematically evaluate design trade-offs in modern HBM systems.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 introduces background information on DRAM and HBM architectures and reviews existing memory modeling tools. Chapter 3 presents the proposed HBM modeling framework and describes the modeling methodology in detail. Chapter 4 evaluates the proposed model and compares it with existing approaches while analyzing architectural trends across HBM generations. Chapter 5 presents several case studies that explore design trade-offs in stacked memory systems. Finally, Chapter 6 concludes the thesis and discusses directions for future work.

CHAPTER 2

Background and Related Work

2.1 DRAM and HBM Architecture

Dynamic Random Access Memory (DRAM) has long been the dominant technology used for main memory in computing systems due to its high density and relatively low cost. Conventional DRAM architectures organize memory cells into hierarchical structures consisting of channels, ranks, banks, and subarrays. Each DRAM cell stores a single bit of data using a capacitor and an access transistor, while peripheral circuits such as sense amplifiers, row decoders, and column decoders enable data access operations.

In traditional memory systems, DRAM devices are connected to processors through memory channels with relatively narrow I/O interfaces. While improvements in signaling speed and memory controller design have increased memory bandwidth over time, these approaches are fundamentally limited by the number of available pins and package-level interconnect constraints.

HBM addresses these limitations by adopting a three-dimensional stacked memory architecture. In HBM, multiple DRAM dies are vertically stacked using TSVs and connected to a base die that handles memory control and external communication. This architecture enables extremely wide memory interfaces and significantly higher bandwidth compared to conventional DRAM technologies.

HBM is typically organized into multiple channels within each memory stack, allowing parallel memory access across different memory partitions. This organization enables high aggregate memory bandwidth while maintaining improved energy efficiency compared to

traditional memory architectures.

2.2 Through-Silicon Via Technology

TSVs are vertical electrical interconnects that pass through silicon wafers or dies to enable three-dimensional integrated circuits [10]. In HBM architectures, TSVs provide high-density vertical connections between stacked DRAM dies and the base die.

TSVs can be categorized into several types depending on their functions within the stacked memory structure. Signal TSVs are used to transmit data and control signals between memory dies and the base die. Power and ground TSVs provide vertical power delivery to stacked memory layers. Thermal TSVs may also be introduced to improve heat dissipation in high-power stacked memory structures.

The allocation and placement of TSVs significantly affect the physical design and performance characteristics of HBM systems [10]. Increasing the number of signal TSVs can improve memory bandwidth, while sufficient power and ground TSVs are required to maintain reliable power delivery. Thermal TSVs can help mitigate temperature hotspots caused by high power density in stacked memory dies.

Because TSVs occupy valuable silicon area and introduce additional design constraints, careful modeling of TSV allocation is important when exploring design trade-offs in HBM architectures.

2.3 Existing Memory Modeling Tools

A variety of modeling tools have been developed to study the architectural and physical characteristics of DRAM-based memory systems. These tools can generally be categorized into three groups: controller-level DRAM simulators, memory array modeling tools, and structural DRAM modeling frameworks.

Controller-level DRAM simulators such as Ramulator, DRAMSim3, and DRAMSys pri-

marily focus on modeling memory timing behavior and memory controller policies [11, 12, 13]. These tools are widely used for evaluating system-level memory performance, including scheduling policies, memory access latency, and bandwidth utilization. However, they typically abstract away the physical structure of DRAM devices and therefore do not capture the detailed organization of memory arrays or the physical characteristics of stacked memory architectures.

Memory array modeling tools such as CACTI have been widely used to estimate the physical properties of memory arrays, including area, access latency, and power consumption [9]. Extensions such as CACTI-3DD attempt to incorporate three-dimensional memory structures by modeling stacked memory arrays and TSV-based interconnects. While these tools provide valuable insights into array-level design trade-offs, their modeling scope is primarily focused on memory arrays rather than the full architectural structure of modern stacked DRAM systems.

More recently, frameworks such as DreamRAM have been proposed to model the microarchitectural structure of stacked DRAM systems such as HBM [5]. DreamRAM models detailed DRAM die organizations including memory cell arrays, peripheral circuits, and bank-level structures, while also incorporating TSV-based vertical stacking across memory dies. The framework provides analytical estimates for area, power, and latency, enabling architectural exploration of stacked DRAM systems.

Despite these advances, existing modeling tools still face challenges when applied to modern HBM architectures, which introduce additional design complexity due to stacked dies, TSV interconnects, and thermal constraints.

2.4 Limitations of Existing Approaches

Although DreamRAM provides detailed modeling of stacked DRAM structures, it does not explicitly target architectural exploration across different HBM generations or capture system-level design considerations such as TSV allocation strategies, thermal TSV place-

ment, and design trade-offs in large-scale HBM systems [5].

Most system-level DRAM simulators primarily focus on memory timing behavior and scheduling policies rather than the physical organization of stacked memory architectures. As a result, they do not explicitly model structural aspects of HBM such as memory die organization, TSV allocation, or base die circuitry.

Similarly, traditional memory array modeling tools focus on planar memory structures and do not account for the additional design constraints introduced by three-dimensional stacking [9]. Important design factors such as TSV area overhead, power delivery through vertical interconnects, and thermal behavior of stacked memory dies are often not considered.

These limitations make it difficult to perform comprehensive design exploration of HBM architectures using existing tools. Therefore, a modeling framework that captures both architectural and physical characteristics of HBM systems is needed to enable systematic analysis of design trade-offs in modern stacked memory technologies.

CHAPTER 3

Proposed HBM Modeling Framework

3.1 Modeling Overview

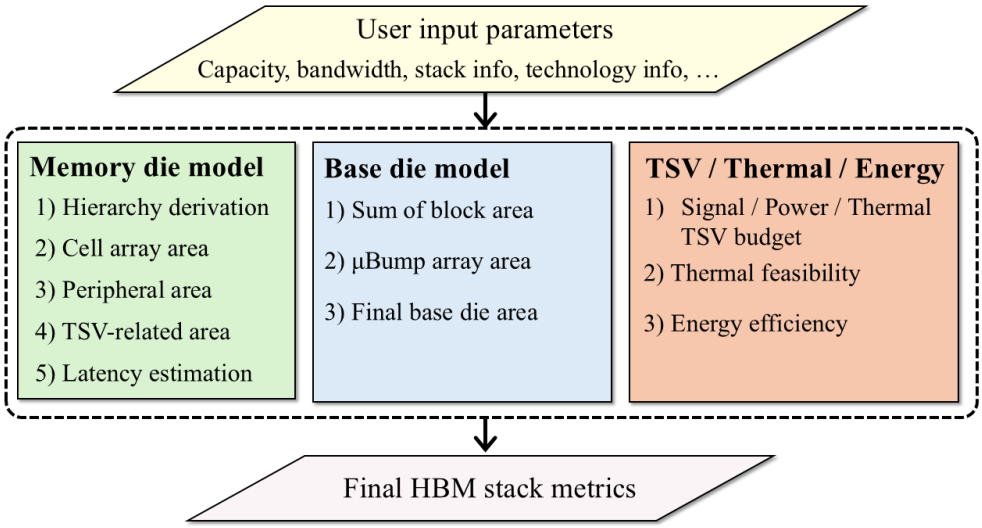


Figure 3.1: Overall flow of the proposed HBM modeling framework. User-defined architectural and technology parameters are provided as inputs to the memory-die, base-die, and TSV/thermal/energy models, whose outputs are then integrated to estimate the final HBM stack metrics. The framework can also incorporate optional stacked-logic loading through additional power and hotspot inputs.

This chapter presents the proposed modeling framework for architecture-level exploration of HBM systems. The framework is designed to capture the major physical and architectural characteristics of stacked memory structures while remaining flexible enough to support comparative studies across different HBM generations. Figure 3.1 illustrates the overall

modeling flow, showing how user-defined inputs are translated into separate memory-die, base-die, and TSV/thermal/energy models, and how these intermediate results are finally combined into stack-level metrics.

The framework takes user-defined architectural and technology parameters as inputs, including capacity, bandwidth, stack organization, and technology information. As summarized in Figure 3.1, these inputs propagate through three major modeling components. The first is the memory die model, which derives the internal hierarchy of the DRAM layer and estimates cell-array area, peripheral area, TSV-related area, and latency. The second is the base die model, which estimates the area contributions of logic and interface blocks, μbump arrays, and final base-die constraints. The third is the TSV, thermal, and energy model, which determines TSV budgeting, evaluates thermal feasibility, and estimates energy efficiency.

The outputs of these modules are combined to obtain final HBM stack metrics, including area, footprint, latency, and energy efficiency. This modular structure is important because HBM design is not determined by a single metric. Instead, the feasibility of a design point depends on the interaction among die organization, vertical interconnect resources, package-level constraints, and thermal limits.

In addition to the nominal HBM stack, the framework can also incorporate an abstract stacked-logic load coupled to the memory stack. In the current implementation, this is modeled through additional stack-level power, voltage, and hotspot terms that affect TSV budgeting and thermal feasibility. This optional path, also indicated in Figure 3.1, is intended to support heterogeneous integration studies in which logic may be placed above or below the memory stack while keeping its impact visible through changes in power-delivery demand, hotspot loading, and TSV allocation.

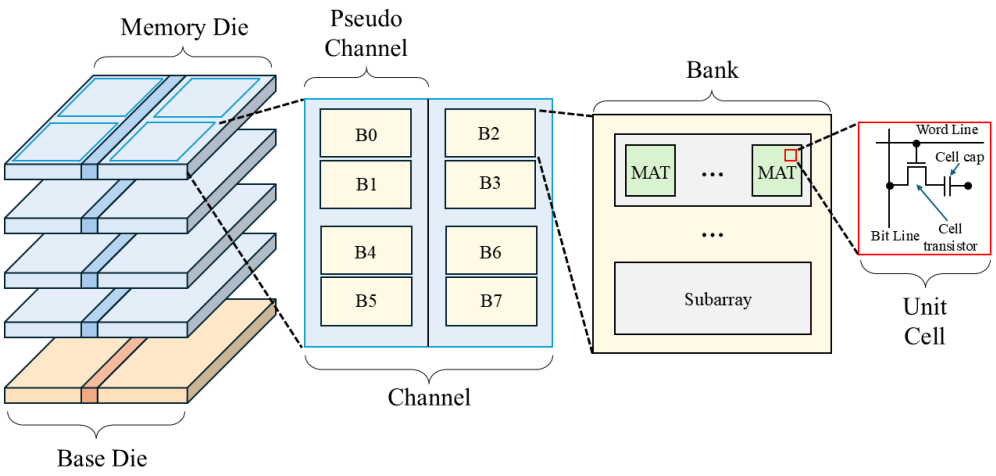


Figure 3.2: Hierarchical organization of the HBM memory die assumed in the proposed model. Each HBM stack consists of multiple memory dies stacked on a base die, and each memory die is organized into channels, pseudo-channels, banks, mats, and subarrays.

3.2 Memory Die Modeling

The memory die model is designed to capture, at a high level, how the physical organization of a DRAM layer in an HBM stack determines its area and timing behavior. As illustrated in Figure 3.2, the proposed model assumes a hierarchical memory-die organization in which each HBM stack consists of multiple memory dies placed on a base die, and each memory die is structured into channels, pseudo-channels, banks, mats, and subarrays. Rather than treating the memory die as a single lumped block, the proposed model decomposes it into three major parts: the cell array, the local mat-level peripheral structures, and the global bank-level peripheral structures. This hierarchy is important because the physical implementation cost emerges across multiple levels of organization, from the unit-cell and mat structures up to the bank and channel levels. These components are then combined with the TSV footprint and the latency model to produce the final memory-die results. This decomposition is useful because the physical cost of an HBM memory die depends not only on storage capacity, but also on how that capacity is organized and supported by peripheral circuitry.

3.2.1 Cell Array

The modeling flow begins from die-level storage capacity. For a given memory die, the user-visible capacity is determined by the configured channel density and the number of channels implemented per die:

$$Capacity_{\text{die}}\ [\text{bit}] = D_{\text{ch}}\ [\text{bit}] \cdot N_{\text{ch,die}}, \tag{3.1}$$

where D_{ch} is the channel density and $N_{\text{ch,die}}$ is the number of channels per die. This user capacity is then translated into the total number of stored bits. In addition to the user data bits, the model also includes error-correction and metadata overhead:

$$N_{\text{bit,total}}\ [\text{bit}] = N_{\text{user}}\ [\text{bit}] + N_{\text{ECC}}\ [\text{bit}] + N_{\text{meta}}\ [\text{bit}], \tag{3.2}$$

Here, Error-Correcting Code (ECC) bits are included because they occupy physical storage in the memory array, and metadata bits are included for the same reason. In modern memory systems, metadata is not merely an abstract control quantity; it is often stored together with or alongside the protected data block and therefore consumes real storage capacity. Prior work on compressed memory similarly shows that compression metadata can be integrated with ECC-related information inside the memory block itself, reinforcing the need to account for such bits in physical capacity estimation [14, 15, 16, 17].

After the total stored bit count is determined, the model estimates the cell-array area. In this work, the nominal DRAM unit-cell area is defined as $6F^2$, where F is the half pitch of the DRAM process [18]; in the generation-specific modeling, F is assumed to correspond to D1y for HBM2, D1z for HBM3, and D1c for HBM4. The corresponding core-array area is then written as

$$A_{\text{core}}\ [\text{mm}^2] = N_{\text{bit,total}}\ [\text{bit}] \cdot A_{\text{cell,eff}}\ [\text{nm}^2/\text{bit}] \cdot 10^{-12}, \tag{3.3}$$

where $A_{\text{cell,eff}}$ is the effective unit-cell area derived from the assumed DRAM pitch model. This formulation provides a direct link between storage capacity and array area while keeping the cell model simple and technology-aware. The key inputs and outputs of the cell-array model are summarized in Table 3.1.

Table 3.1: Parameters and symbols for the memory-die cell-array area model.

Inputs	**Symbol**	**Description**
Channel density	D_{ch}	User capacity per channel [bit]
Channels per die	$N_{\text{ch,die}}$	Number of channels implemented in one memory die
ECC ratio	r_{ECC}	ECC overhead ratio used to derive additional ECC bits
Metadata ratio	r_{meta}	Metadata overhead ratio; if $r_{\text{meta}} = 0$, no metadata bits are added
Half pitch	F_{mem}	Memory half pitch [nm]
Unit-cell factor	f_{cell}	Unit-cell area factor
Outputs	**Symbol**	**Formula**
User bits per die	N_{user}	$\lceil D_{\text{ch}} \cdot N_{\text{ch,die}} \rceil$ [bit]
ECC bits per die	N_{ECC}	$\lceil N_{\text{user}} / r_{\text{ECC}} \rceil$ [bit]
Metadata bits per die	N_{meta}	$\begin{cases} 0, & r_{\text{meta}} = 0 \\ \lceil N_{\text{user}} / r_{\text{meta}} \rceil, & r_{\text{meta}} > 0 \end{cases}$ [bit]
Total stored bits	$N_{\text{bit,total}}$	$N_{\text{user}} + N_{\text{ECC}} + N_{\text{meta}}$ [bit]
Effective cell area	$A_{\text{cell,eff}}$	$f_{\text{cell}} \left(F_{\text{mem}}\right)^2$ [nm^2]
Core array area	A_{core}	$N_{\text{bit,total}} \cdot A_{\text{cell,eff}}$ [mm^2]

Once the total bit count is known, the model derives the internal hierarchy of the memory die. A die is first partitioned into channels and banks, and each bank is then mapped into repeated mats and subarrays [19]. This hierarchy is important because it determines not only how much cell area is required, but also how many local and global support structures must be instantiated. In other words, two designs with the same total capacity can still have different physical area depending on how the hierarchy is organized.

3.2.2 Peripheral Circuits

Based on this hierarchy, the model separates peripheral circuitry into mat-level and bank-level structures. The mat-level peripheral area captures local structures that are replicated together with the array tiles, namely the bitline sense amplifier (BLSA) region and the local wordline driver (LWD) region. At this level, the BLSA is associated with the bitline direction, while the LWD is associated with the wordline direction. Accordingly, their total areas are determined from the number of instantiated mats together with the numbers of bitlines and wordlines in each mat, and are translated into physical area using the corresponding pitch dimensions and area-overhead assumptions. The corresponding mat-level peripheral area is approximated as the sum of the BLSA and LWD regions:

$$A_{\text{mat,peri}}\ [\text{mm}^2] \approx N_{\text{BLSA}} \cdot u_{\text{BLSA}}\ [\text{mm}^2] + N_{\text{LWD}} \cdot u_{\text{LWD}}\ [\text{mm}^2], \tag{3.4}$$

where N_{BLSA} and N_{LWD} denote the numbers of local sense-amplifier and local wordline-driver regions, and u_{BLSA} and u_{LWD} denote their corresponding unit areas. The main inputs and outputs of the mat-level peripheral model are summarized in Table 3.2.

The bank-level peripheral area captures more global support blocks, including row decoders, column decoders, global sense amplifiers, and multiplexing logic. In the implementation, the unit areas of these global peripheral blocks are first constructed from the basic logic required by each block and the standard-cell areas provided by the ASAP7 7 nm standard-cell library [20], which is a widely used academic open library [21, 22, 23, 24, 25]. The total area of each block is then obtained by multiplying these unit areas by the corresponding

Table 3.2: Parameters and symbols for the mat-level peripheral area model. The term F_{mem} is referenced from Table 3.1.

Inputs	**Symbol**	**Description**
WL pitch multiplier	λ_{WL}	Wordline pitch in units of F
BL pitch multiplier	λ_{BL}	Bitline pitch in units of F
BLSA height factor	h_{BLSA}	Bitline sense-amplifier stripe height in bitline-pitch units
LWD width factor	w_{LWD}	Local wordline-driver stripe width in wordline-pitch units
BLSA overhead	ξ_{BLSA}	Bitline sense-amplifier area overhead factor
LWD overhead	ξ_{LWD}	Local wordline-driver area overhead factor
Total MATs per die	$N_{\text{mat,die}}$	Number of instantiated MATs in one memory die
MAT rows	$N_{\text{row,mat}}$	Number of rows in one MAT
MAT columns	$N_{\text{col,mat}}$	Number of columns in one MAT
Outputs	**Symbol**	**Formula**
WL pitch	P_{WL}	$\lambda_{\text{WL}} F_{\text{mem}}$ [nm]
BL pitch	P_{BL}	$\lambda_{\text{BL}} F_{\text{mem}}$ [nm]
BLSA unit area	u_{BLSA}	$\xi_{\text{BLSA}}\, h_{\text{BLSA}}\, P_{\text{BL}}^2$ [mm^2]
LWD unit area	u_{LWD}	$\xi_{\text{LWD}}\, w_{\text{LWD}}\, P_{\text{WL}}^2$ [mm^2]
Number of BLSA slices	N_{BLSA}	$N_{\text{mat,die}} \cdot N_{\text{col,mat}}$
Number of LWD slices	N_{LWD}	$N_{\text{mat,die}} \cdot N_{\text{row,mat}}$
Mat-level peripheral area	$A_{\text{mat,peri}}$	$(N_{\text{BLSA}} u_{\text{BLSA}} + N_{\text{LWD}} u_{\text{LWD}})$ [mm^2]

hierarchy counts derived from the memory organization, such as the number of subarrays, the number of banks, and the number of selected global bitline paths. Finally, these areas are converted to the target technology node using the technology-scaling model introduced earlier [26]. At a high level, the bank-level peripheral area is expressed as

$$A_{\text{bank,peri}}\ [\text{mm}^2] \approx A_{\text{rowdec}}\ [\text{mm}^2] + A_{\text{coldec}}\ [\text{mm}^2] + A_{\text{GSA}}\ [\text{mm}^2] + A_{\text{mux}}\ [\text{mm}^2], \tag{3.5}$$

where

$$A_{\text{mux}}\ [\text{mm}^2] = A_{\text{colmux}}\ [\text{mm}^2] + A_{\text{outmux}}\ [\text{mm}^2]. \tag{3.6}$$

Here, A_{colmux} denotes the column-multiplexing logic associated with selecting data from the global bitline paths within the bank-level hierarchy, while A_{outmux} denotes the output-multiplexing logic associated with mapping the selected internal data paths to the final die-level channel outputs. Thus, each term reflects both the hierarchical replication of the corresponding block and the technology scaling applied to its unit implementation. This approach allows the model to remain simple while still capturing the fact that global peripheral cost changes with both memory hierarchy and technology assumptions. The bank-level peripheral model is summarized in Table 3.3.

The active memory-die area is then obtained by combining the core array and peripheral contributions:

$$A_{\text{active}}\ [\text{mm}^2] = A_{\text{core}}\ [\text{mm}^2] + A_{\text{mat,peri}}\ [\text{mm}^2] + A_{\text{bank,peri}}\ [\text{mm}^2]. \tag{3.7}$$

At this stage, the memory die is not yet complete because TSV-related area has not been incorporated. In the proposed framework, TSV allocation is modeled separately, as described in the TSV modeling section, and the resulting TSV footprint is then reflected back into the memory-die result. This separation is important because TSV demand is driven not only by local memory-die geometry, but also by stack-level bandwidth, power-delivery, and thermal considerations. After TSV budgeting and footprint estimation are completed, the TSV array area is integrated into the final memory-die area:

$$A_{\text{mem}}\ [\text{mm}^2] = A_{\text{active}}\ [\text{mm}^2] + A_{\text{TSV}}\ [\text{mm}^2], \tag{3.8}$$

Table 3.3: Parameters and symbols for the bank-level peripheral area model. The term $A_{\text{mat,peri}}$ is referenced from Table 3.2.

Inputs	**Symbol**	**Description**
Unit block areas	u_{row}, u_{col} u_{GSA}, u_{mux2}	Technology-scaled unit areas for row decoder, column decoder, global sense amplifier, and 2:1 mux blocks [mm^2]
Col-mux ratio	M_{col}	Column mux ratio
Col-mux fitting factor	k_{colmux}	Fitting factor for the simplified mux-tree model
Channel width	W_{IO}	Number of data outputs per channel [bit]
Outputs	**Symbol**	**Formula**
Rows per bank	$N_{\text{row,bk}}$	$N_{\text{sub,bk}} \cdot N_{\text{row,mat}}$
Selected cols per subarray	$N_{\text{sel/sub}}$	$\left\lceil \frac{N_{\text{mat/sub}} \cdot N_{\text{col,mat}}}{M_{\text{col}}} \right\rceil$
Global bitline paths	N_{GBL}	$N_{\text{bk}} \cdot N_{\text{sub,bk}} \cdot N_{\text{sel/sub}}$
Row-decoder area	A_{rowdec}	$N_{\text{bk}} \cdot N_{\text{row,bk}} \cdot u_{\text{row}}$ [mm^2]
Column-decoder area	A_{coldec}	$N_{\text{bk}} \cdot u_{\text{col}}$ [mm^2]
Global SA area	A_{GSA}	$N_{\text{GBL}} \cdot u_{\text{GSA}}$ [mm^2]
Col-mux depth	L_{colmux}	$\lceil \log_2 M_{\text{col}} \rceil$
Col-mux area	A_{colmux}	$N_{\text{GBL}} \cdot L_{\text{colmux}} \cdot k_{\text{colmux}} \cdot u_{\text{mux2}}$ [mm^2]
Data outputs per die	N_{out}	$W_{\text{IO}} \cdot N_{\text{ch,die}}$ [bit]
Output mux ratio	M_{out}	$\lceil N_{\text{GBL}}/N_{\text{out}} \rceil$
Output mux area	A_{outmux}	$N_{\text{out}} \cdot \lceil \log_2 M_{\text{out}} \rceil \cdot u_{\text{mux2}}$ [mm^2]
Bank-level peri. area	$A_{\text{bank,peri}}$	$A_{\text{rowdec}} + A_{\text{coldec}} + A_{\text{colmux}} + A_{\text{GSA}} + A_{\text{outmux}}$ [mm^2]
Total peripheral area	A_{peri}	$A_{\text{mat,peri}} + A_{\text{bank,peri}}$ [mm^2]

where A_{TSV} is the reserved TSV-array footprint. Thus, the final reported memory-die area includes both active circuitry and vertical-interconnect overhead. The main inputs and outputs of the TSV budgeting and footprint model are summarized in Table 3.6.

3.2.3 Latency

In addition to area estimation, the memory die model incorporates a standalone latency model. The latency analysis adopts a simplified analytical methodology based on DreamRAM [5], using baseline timing parameters, capacitance-related inputs, and key timing assumptions from that framework. The same memory organization derived for area modeling is reused here to evaluate the dominant timing quantities summarized in Table 3.4. The table presents the principal latency components considered in the analytical model rather than the detailed fitting parameters used in the full implementation. At a high level, the model computes

$$t_{CK}\ [\mathrm{ns}] \propto R_{\mathrm{WL}}\ [\Omega]\ C_{\mathrm{WL}}\ [\mathrm{F}] + R_{\mathrm{DL}}\ [\Omega]\ C_{\mathrm{DL}}\ [\mathrm{F}], \tag{3.9}$$

$$t_{CL}\ [\mathrm{ns}] \approx t_{\mathrm{array}} + t_{\mathrm{bus}} + t_{\mathrm{TSV}}, \tag{3.10}$$

$$t_{RCD}\ [\mathrm{ns}] \approx t_{\mathrm{WL}} + t_{\mathrm{BL}} + t_{\mathrm{SA}}, \qquad t_{RP}\ [\mathrm{ns}] \approx t_{\mathrm{BL,pre}} + t_{\mathrm{restore}}, \tag{3.11}$$

These expressions are intentionally written in simplified form to highlight the dominant timing components rather than low-level fitting factors. In this flow, t_{CK} is governed primarily by the core-array path, t_{CL} reflects the read-data path including die-level transport and TSV-related traversal, and t_{RCD}/t_{RP} capture the dominant activate and precharge timing behavior associated with bank geometry and capacitance scaling. The final worst-case latency is then summarized as

$$t_{\mathrm{worst}}\ [\mathrm{ns}] = t_{CL} + t_{RCD} + t_{RP}. \tag{3.12}$$

Taken together, this memory-die model provides a macroscopic view of how an HBM DRAM layer is constructed and evaluated in the proposed framework. The model starts from storage capacity, expands that capacity into a concrete hierarchy, translates the hierarchy into cell-array and peripheral area, incorporates TSV footprint from a separate vertical-interconnect

Table 3.4: Parameters and symbols for the memory-die latency model.

Inputs	**Symbol**	**Description**
Wordline resistance/capacitance	$R_{\mathrm{WL}}, C_{\mathrm{WL}}$	Dominant RC terms of the wordline path [Ω, F]
Data-line resistance/capacitance	$R_{\mathrm{DL}}, C_{\mathrm{DL}}$	Dominant RC terms of the internal data-line path [Ω, F]
Array access delay	t_{array}	Delay associated with internal array access in the read path [ns]
Bus delay	t_{bus}	Delay associated with on-die data transport through internal buses [ns]
TSV delay	t_{TSV}	Delay contribution of TSV traversal in the read path [ns]
Wordline activation delay	t_{WL}	Delay associated with wordline activation [ns]
Bitline sensing delay	t_{BL}	Delay associated with bitline development and sensing [ns]
Sense-amplifier delay	t_{SA}	Delay contribution of the sense-amplifier stage [ns]
Bitline precharge delay	$t_{\mathrm{BL,pre}}$	Delay associated with bitline precharge [ns]
Restore delay	t_{restore}	Delay associated with restoring the cell/bitline state [ns]
Outputs	**Symbol**	**Formula / Description**
Clock period	t_{CK}	$\propto R_{\mathrm{WL}}C_{\mathrm{WL}} + R_{\mathrm{DL}}C_{\mathrm{DL}}$ [ns]
CAS latency	t_{CL}	$\approx t_{\mathrm{array}} + t_{\mathrm{bus}} + t_{\mathrm{TSV}}$ [ns]
Row-to-column delay	t_{RCD}	$\approx t_{\mathrm{WL}} + t_{\mathrm{BL}} + t_{\mathrm{SA}}$ [ns]
Row precharge time	t_{RP}	$\approx t_{\mathrm{BL,pre}} + t_{\mathrm{restore}}$ [ns]
Worst-case latency	t_{worst}	$t_{CL} + t_{RCD} + t_{RP}$ [ns]

model, and finally evaluates latency using a DreamRAM-style timing flow tied to the same organizational assumptions. This integrated structure allows the reader to understand not only the final area and timing outputs, but also how those outputs arise from the interaction among capacity, hierarchy, technology assumptions, peripheral support, and TSV overhead.

3.3 Base Die Modeling

The base die model captures the physical constraints of the logic die located at the bottom of the HBM stack. Unlike the memory dies, which are dominated by dense cell arrays, the base die is shaped primarily by interface logic, TSV landing regions, μbump placement, and testing-related structures [7]. For this reason, the base die is modeled not as an array-dominated storage die, but as a constrained logic-and-interconnect die whose final footprint is determined by both internal block area and package-level placement limits.

The modeling flow begins from the PHY area. In the implementation, the PHY is represented using JEDEC-based interface templates for data-word and address-word structures. For a selected HBM generation, the model first determines the reference row and column dimensions of the PHY macro and then converts them into a physical footprint using the configured bump pitch, bump diameter, and bump-map pattern. This produces the unit PHY area for one interface block. The total PHY area is then obtained by scaling the data-word portion with the total number of 32-bit interface groups and the address-word portion with the total number of channels:

$$A_{\text{PHY}}\ [\text{mm}^2] = N_{\text{DW}} \cdot A_{\text{DW,unit}}\ [\text{mm}^2] + N_{\text{ch}} \cdot A_{\text{AW,unit}}\ [\text{mm}^2], \tag{3.13}$$

where $A_{\text{DW,unit}}$ and $A_{\text{AW,unit}}$ denote the unit areas of the data-word and address-word PHY blocks, respectively, N_{DW} is the number of 32-bit interface groups, and N_{ch} is the total number of HBM channels. This formulation reflects the fact that wider interfaces require proportionally more PHY area, while per-channel address/control support scales with channel count.

The geometric dimensions of each PHY unit depend on the bump-map organization. For

both the PHY macros and the global bump-map footprint, the framework supports simple-grid and staggered layouts. In the simple-grid case, horizontal and vertical spacing follow the full bump pitch, whereas in the staggered case the effective step in one direction is reduced by the half-pitch packing rule. As a result, the same logical bump count can produce different physical footprints depending on the assumed bump-map pattern. This is important because the base die is strongly constrained by package-facing interconnect geometry rather than only by transistor-level logic density.

After the PHY area is determined, the model incorporates TSV-related area. The detailed TSV-array area calculation is presented later in the TSV modeling section. Once that TSV footprint is obtained, the same TSV-array area is reflected into the base-die model. This is because the base die and the memory dies are vertically stacked and therefore share the same TSV interface footprint. As a result, the TSV-array area calculated in the TSV model is applied consistently to both the memory-die and base-die area calculations.

The model then evaluates the package-facing μbump footprint. Given the total number of bumps, bump pitch, bump diameter, maximum allowable array width, and bump-map pattern, the framework packs bumps row by row until all required bumps are placed. This yields a bump-map-limited footprint area,

$$A_{\text{bump}}\ [\text{mm}^2] = W_{\text{bump}}\ [\text{mm}] \cdot H_{\text{bump}}\ [\text{mm}], \tag{3.14}$$

where W_{bump} and H_{bump} are the effective width and height of the packed bump array. This bump-map area is not simply another logic block; rather, it represents the physical placement constraint imposed by package interconnects and external connectivity.

Testing-related structures are modeled separately because they consume dedicated base-die area beyond the main PHY and TSV regions. In the implementation, the test-related area includes both design-for-test (DA) blocks and probe-pad regions:

$$A_{\text{test}}\ [\text{mm}^2] = A_{\text{DA}}\ [\text{mm}^2] + A_{\text{probe}}\ [\text{mm}^2]. \tag{3.15}$$

Within the bump map, dedicated regions are assigned to the DA block and the probe-pad area. The sizes of these regions are specified by their respective row and column counts,

and their physical footprints are then computed from the bump dimensions and bump pitch. In this way, A_{DA} and A_{probe} reflect explicitly assigned package-facing regions in the bump map rather than abstract logic-only estimates. This keeps the model consistent with the physical interpretation of the base die as a logic-and-infrastructure layer rather than only a data-transport layer.

The internal block-sum area is then formed by combining PHY, TSV, and testing contributions:

$$A_{\mathrm{blk}}\ [\mathrm{mm}^2] = A_{\mathrm{PHY}}\ [\mathrm{mm}^2] + A_{\mathrm{TSV}}\ [\mathrm{mm}^2] + A_{\mathrm{test}}\ [\mathrm{mm}^2]. \tag{3.16}$$

At this point, the framework applies a general base-die area margin to both the block-sum estimate and the bump-map estimate, and selects the larger one as the preliminary base-die result:

$$A_{\mathrm{blk}}^{(\mathrm{margined})}\ [\mathrm{mm}^2] = \frac{A_{\mathrm{blk}}\ [\mathrm{mm}^2]}{1 - m_{\mathrm{base}}}, \qquad A_{\mathrm{bump}}^{(\mathrm{margined})}\ [\mathrm{mm}^2] = \frac{A_{\mathrm{bump}}\ [\mathrm{mm}^2]}{1 - m_{\mathrm{base}}}, \tag{3.17}$$

$$A_{\mathrm{base,pre}}\ [\mathrm{mm}^2] = \max\left(A_{\mathrm{blk}}^{(\mathrm{margined})}\ [\mathrm{mm}^2], A_{\mathrm{bump}}^{(\mathrm{margined})}\ [\mathrm{mm}^2]\right), \tag{3.18}$$

where m_{base} is the configured base-die area margin. This decision step is important because different design points may be limited by different physical mechanisms. In some cases, the accumulated internal logic-and-TSV blocks dominate the base-die size; in other cases, the package-facing bump-map footprint becomes the limiting factor.

After the preliminary base-die area is obtained, a stack-level footprint harmonization step is applied in the top-level integration flow. The preliminary base-die area is compared against the final memory-die footprint. If the base die is already larger than the memory die, its size is retained. Otherwise, the base die is expanded so that the stack remains physically compatible during assembly:

$$A_{\mathrm{base,final}}\ [\mathrm{mm}^2] = \begin{cases} A_{\mathrm{base,pre}}\ [\mathrm{mm}^2], & \text{if } A_{\mathrm{base,pre}} \geq A_{\mathrm{mem}}, \\ A_{\mathrm{mem}}\ [\mathrm{mm}^2] \cdot \dfrac{1}{1 - m_{\mathrm{base}}}, & \text{if } A_{\mathrm{base,pre}} < A_{\mathrm{mem}}. \end{cases} \tag{3.19}$$

This post-processing step reflects the fact that the base die is part of a vertically assembled stack and therefore cannot be interpreted independently of the memory-die footprint.

Taken together, the base die model provides a macroscopic description of how the bottom logic die is sized in the proposed HBM framework. The model begins from generation-specific PHY structures, adds the TSV footprint produced by the separate TSV model, evaluates package-facing bump-map constraints, includes testing-related structures, and then selects the final area according to the dominant physical limit. This structure makes the base-die estimate more interpretable for HBM design exploration, because it reveals whether a given design point is limited primarily by logic accumulation, TSV landing demand, or package-interconnect footprint.

3.4 TSV Modeling

TSVs are modeled explicitly as a first-class design resource in the proposed framework. Rather than representing vertical interconnects by a single aggregate count, the model assigns separate budgets for power TSVs, ground TSVs, I/O TSVs, internal TSVs, and thermal TSVs. This separation is important because each class is driven by a different physical requirement: power and ground TSVs are determined by current-delivery constraints, I/O TSVs by bandwidth demand, internal TSVs by architectural organization, and thermal TSVs by heat-removal requirements. The total TSV demand is therefore written as

$$N_{\mathrm{TSV,total}} = N_{\mathrm{PWR}} + N_{\mathrm{GND}} + N_{\mathrm{IO}} + N_{\mathrm{internal}} + N_{\mathrm{thermal}}. \tag{3.20}$$

At a high level, the TSV budgeting flow combines three types of constraints: bandwidth delivery, power delivery, and thermal feasibility. The first two directly determine the signal and power-related TSV budgets, while the third determines whether additional thermal TSVs must be inserted. In this sense, the TSV model acts as the point where electrical and thermal requirements are unified into a single vertical-interconnect budget.

The budgeting process begins from the target stack bandwidth and the modeled memory energy efficiency. The framework first converts the energy-efficiency result into total HBM stack power:

$$P_{\mathrm{HBM}}\ [\mathrm{W}] = \eta_{\mathrm{mem}}\ [\mathrm{J/bit}] \cdot BW\ [\mathrm{bit/s}] \cdot U + P_{\mathrm{bg}}\ [\mathrm{W}], \tag{3.21}$$

where η_{mem} is the modeled memory energy efficiency, BW is the target stack bandwidth, U is the active-utilization factor, and P_{bg} is the background power. This stack power is then converted into total HBM current,

$$I_{\text{HBM}}\ [\text{A}] = \frac{P_{\text{HBM}}\ [\text{W}]}{V\ [\text{V}]}, \tag{3.22}$$

where V is the nominal supply voltage. The power TSV budget is obtained by comparing this current against the allowable current per TSV:

$$N_{\text{PWR}} = \left\lceil \frac{I_{\text{HBM}}\ [\text{A}]}{J_{\max}\ [\text{A/mm}^2] \cdot A_{\text{TSV}_1}\ [\text{mm}^2]} \right\rceil, \qquad A_{\text{TSV}_1}\ [\text{mm}^2] = \pi \left(\frac{D_{\text{TSV}}\ [\mu\text{m}]}{2} \right)^2 \cdot 10^{-6}, \tag{3.23}$$

where $J_{\max}$ is the TSV current-density limit and D_{TSV} is the TSV diameter. The ground TSV budget is then derived from the power-delivery budget so that return-current support is provisioned together with the power network:

$$N_{\text{GND}} \approx N_{\text{PWR}}. \tag{3.24}$$

That is, the ground TSV budget is provisioned together with the power TSV budget so that the return-current capability scales with the delivered current.

The I/O TSV budget is determined from interface throughput requirements. Let W_{IO} denote the per-channel interface width and R_{IO} the signaling rate per pin. The required number of stack-level channels is estimated as

$$N_{\text{ch,stack}} = \left\lceil \frac{BW\ [\text{bit/s}]}{W_{\text{IO}}\ [\text{bit}] \cdot R_{\text{IO}}\ [\text{bit/s}]} \right\rceil, \tag{3.25}$$

Based on this bandwidth requirement and the assumed channel organization, the model assigns the signal TSV budget as

$$N_{\text{IO}} = N_{\text{sig/ch}} \cdot N_{\text{ch,die}} + N_{\text{glob/die}}, \tag{3.26}$$

where $N_{\text{sig/ch}}$ denotes the number of signal TSVs allocated per channel and $N_{\text{glob/die}}$ denotes the global TSV overhead per die. This formulation captures the fact that higher bandwidth demand requires more vertical signal resources.

In addition to explicit signal TSVs, the model also includes internal TSVs to account for vertical interconnect overhead that is not directly exposed at the external I/O interface. At a high level, this term is written as

$$N_{\text{internal}} \propto N_{\text{IO}}. \tag{3.27}$$

This means that internal TSV demand scales with the overall signal TSV requirement. This abstraction allows the model to capture architecture-dependent internal routing overhead without introducing unnecessary circuit-level detail.

Thermal TSVs are then incorporated to account for heat-removal requirements. In the proposed framework, thermal TSV demand is not assigned independently of the other TSV classes. Instead, it is obtained from the thermal model described in the following section, which converts stack power into per-die power, hotspot area, required copper area, and finally additional thermal TSV demand. As a result, the TSV model receives

$$N_{\text{thermal}} \approx \left\lceil \frac{A_{\text{Cu,add}}\ [\text{mm}^2]}{A_{\text{TSV,Cu}}\ [\text{mm}^2]} \right\rceil, \tag{3.28}$$

where $A_{\text{Cu,add}}$ is the additional copper area required by the thermal model and $A_{\text{TSV,Cu}}$ is the copper cross-sectional area of one TSV. This formulation shows that thermal TSV demand is determined by the additional copper area needed for thermal feasibility after accounting for the copper already provided by existing TSV classes.

After all TSV classes are determined, the framework estimates the total TSV-array footprint. The TSVs are packed according to the assumed TSV pitch and pattern, and the resulting footprint is translated into a physical area:

$$A_{\text{TSV}}\ [\text{mm}^2] \approx W_{\text{TSV}}\ [\text{mm}] \cdot H_{\text{TSV}}\ [\text{mm}], \tag{3.29}$$

where W_{TSV} and H_{TSV} are the effective width and height of the packed TSV array determined by the total TSV count, TSV pitch, and packing pattern. Figure 3.3 illustrates representative array layouts using simple-grid and staggered patterns, highlighting that the assumed placement geometry can change the effective packing efficiency and, consequently,

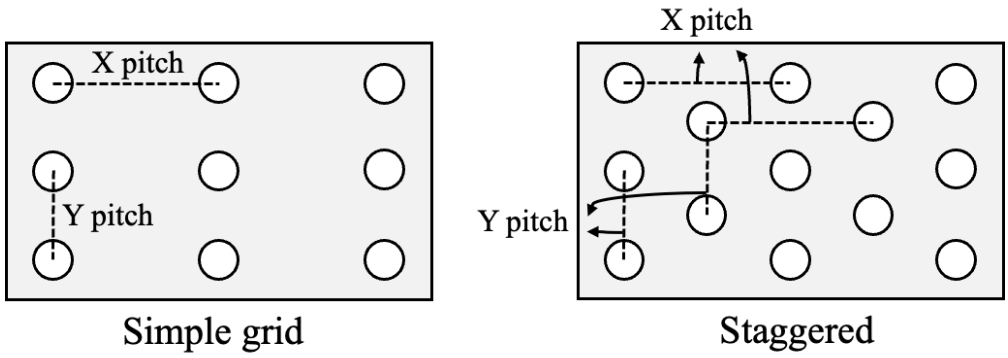


Figure 3.3: Representative TSV array layouts: simple-grid and staggered patterns.

the reserved TSV-array area even for a similar number of vertical interconnects. This step converts the abstract TSV budget into a physically reserved area that can be integrated into die-level modeling.

The resulting TSV-array area is then reused consistently across the framework. In the memory-die model, the TSV footprint is added to the active die area to obtain the final memory-die size. In the base-die model, the same TSV footprint is incorporated into the block-area calculation of the base die. This shared treatment ensures that the vertical-interconnect assumptions remain consistent across both the memory-die and base-die models.

Overall, the TSV model provides the bridge between stack-level requirements and die-level area impact. Starting from bandwidth, power-delivery, and thermal constraints, it determines the required vertical-interconnect budget, separates that budget into power, ground, signal, internal, and thermal TSV classes, and then translates the total TSV demand into a physical footprint. In this way, TSVs are modeled not as a fixed overhead, but as a design-dependent resource that reflects the electrical and thermal needs of the HBM stack.

3.5 Energy and Thermal Modeling

3.5.1 Energy Efficiency

The proposed framework first estimates memory energy efficiency before performing thermal analysis, because stack power is one of the key inputs to the thermal TSV model. In this work, energy efficiency is expressed as a first-order physical access-energy metric in pJ/bit rather than as a fixed vendor-reported number. This allows the model to connect memory organization and interface activity to stack-level power in a way that is consistent with the rest of the HBM modeling flow.

The energy model adopts a simplified analytical methodology based on DreamRAM [5]. It uses baseline capacitance assumptions and timing-related parameters derived from that framework, while focusing on the dominant energy components associated with a single memory access rather than reproducing every detailed circuit-level fitting term. At a high level, the total access energy is decomposed into read-transfer, activation, and precharge terms:

$$E_{\text{acc}}\ [\text{J}] = E_{\text{RD}}\ [\text{J}] + E_{\text{ACT}}\ [\text{J}] + E_{\text{PRE}}\ [\text{J}]. \tag{3.30}$$

The read-transfer energy captures the cost of moving data from the memory die toward the external interface. Here, the first term represents capacitive switching along the I/O and TSV path, while E_{term} represents the load- or termination-related dissipation during data transfer. At a high level, it is written as

$$E_{\text{RD}}\ [\text{J}] \approx \frac{1}{2}\left(C_{\text{DQ}}\ [\text{F}] + C_{\text{TSV,path}}\ [\text{F}]\right) V_{\text{IO}}^2\ [\text{V}^2] + E_{\text{term}}\ [\text{J}], \tag{3.31}$$

where C_{DQ} and $C_{\text{TSV,path}}$ denote the dominant capacitance terms in the read path, and E_{term} represents the load- or termination-related dissipation.

The activation energy represents the cost of opening a row and sensing data from the cell array. In the proposed model, this term is dominated by the energy associated with the wordline, bitline, and sense-amplifier activity:

$$E_{\text{ACT}}\ [\text{J}] = E_{\text{WL}}\ [\text{J}] + E_{\text{BL,act}}\ [\text{J}] + E_{\text{SA}}\ [\text{J}], \tag{3.32}$$

Similarly, the precharge energy captures the cost of restoring the bitline to its precharged state after the access:

$$E_{\mathrm{PRE}}\ [\mathrm{J}] \approx \frac{1}{2} C_{\mathrm{BL,tot}}\ [\mathrm{F}] V_{\mathrm{core}}^2\ [\mathrm{V}^2]. \tag{3.33}$$

Together, these terms account for the dominant internal array-side energy consumed during a memory transaction.

Because the model reports energy efficiency in pJ/bit, the access-energy terms are normalized by the amount of useful transferred data. In addition, the framework includes a background-power contribution so that standby and non-access energy are also reflected in the final metric. The memory energy efficiency is therefore written as

$$\eta_{\mathrm{mem}}\ [\mathrm{J/bit}] = \frac{E_{\mathrm{ACT,atom}}\ [\mathrm{J}] + E_{\mathrm{PRE,atom}}\ [\mathrm{J}] + E_{\mathrm{RD,atom}}\ [\mathrm{J}]}{N_{\mathrm{atom}}\ [\mathrm{bit}]} + \frac{P_{\mathrm{bg}}\ [\mathrm{W}]}{BW_{\mathrm{useful}}\ [\mathrm{bit/s}]}, \tag{3.34}$$

where N_{atom} is the number of bits in one transfer atom, P_{bg} is the background power, and BW_{useful} is the useful delivered bandwidth. The first term captures the dynamic access energy per delivered bit, while the second term converts background power into an equivalent pJ/bit contribution.

This energy efficiency value serves as the bridge between memory-access behavior and stack-level physical analysis. Once η_{mem} is obtained, the framework converts it into total HBM power as a function of target bandwidth and utilization:

$$P_{\mathrm{HBM}}\ [\mathrm{W}] = \eta_{\mathrm{mem}}\ [\mathrm{J/bit}] \cdot BW\ [\mathrm{bit/s}] \cdot U + P_{\mathrm{bg}}\ [\mathrm{W}], \tag{3.35}$$

where BW is the target stack bandwidth and U is the active utilization factor. The resulting stack power is then used in the subsequent thermal model to estimate per-die power dissipation, hotspot area, and additional thermal TSV demand.

Overall, the energy-efficiency model provides a compact but physically meaningful way to estimate the power cost of an HBM design point. Instead of relying on a fixed energy number, it derives pJ/bit from the dominant read, activation, and precharge components, and then uses that result to drive the thermal feasibility analysis that follows. The grouped inputs and outputs of the energy and thermal models are summarized in Table 3.5.

Table 3.5: Parameters and symbols for the energy and thermal model used in TSV budgeting.

Inputs	**Symbol**	**Description**
Core / I/O voltage	$V_{\text{core}}, V_{\text{IO}}$	Voltage terms used in the access-energy model [V]
Capacitance terms	$\mathcal{C}_{\text{eng}}$	WL, BL, SA, DQ, TSV-path, and C_{load} parameters [F]
Atom size	N_{atom}	Number of bits in one transfer atom [bit]
Page size	B_{page}	Number of transferred bits per activated page [bit]
Useful bandwidth	BW_{useful}	Effective bandwidth used in energy normalization [bit/s]
Memory-die power	P_{cdie}	Average power dissipated by one memory die [W]
Reference die area	A_{die}	Memory-die area used in hotspot feasibility checking [mm^2]
Allowable heat-flux limit	q_{flux}	Local heat-flux limit used for hotspot-area estimation [W/mm^2]
Thermal conductivities	$k_{\text{bulk}}, k_{\text{BEOL}}, k_{\text{Cu}}$	Bulk, BEOL, and copper thermal conductivity terms [W/(m·K)]
Temperature limits	$T_{\text{j,max}}, T_{\text{amb}}$	Junction- and ambient-temperature assumptions [K]
Outputs	**Symbol**	**Formula / Description**
Read-transfer energy	E_{RD}	Read energy from I/O switching, TSV-path loading, and termination-related dissipation [J]
Activation energy	E_{ACT}	Wordline, bitline, and sense-amplifier activation energy [J]
Precharge energy	E_{PRE}	Bitline restore / precharge energy [J]
Energy efficiency	η_{mem}	$\dfrac{E_{\text{ACT,atom}} + E_{\text{PRE,atom}} + E_{\text{RD,atom}}}{N_{\text{atom}}} + \dfrac{P_{\text{bg}}}{BW_{\text{useful}}}$ [pJ/bit]
Required hotspot area	$A_{\text{hot,req}}$	$P_{\text{cdie}}/q_{\text{flux}}$ [mm^2]
Used hotspot area	$A_{\text{hot,used}}$	$\min(A_{\text{hot,req}}, A_{\text{die}})$ [mm^2]
Target R_{thermal}	R_{target}	Allowable thermal resistance derived from heat flux and temperature budget [K mm^2/W]
Required copper area	$A_{\text{Cu,req}}$	Additional Cu area needed to satisfy the R_{th} target [mm^2]
Existing hotspot Cu area	$A_{\text{Cu,exist}}$	Cu area already provided by existing non-thermal TSVs [mm^2]
Additional thermal TSVs	N_{thermal}	$\left\lceil \dfrac{\max(A_{\text{Cu,req}} - A_{\text{Cu,exist}}, 0)}{A_{\text{TSV,Cu}}} \right\rceil$

3.5.2 Thermal Modeling

After the memory energy efficiency is converted into total stack power, the framework performs thermal modeling to determine whether the resulting design point is thermally feasible and, if necessary, how many additional thermal TSVs are required. In this flow, thermal modeling is not treated as an isolated post-processing step. Instead, it is directly coupled to the power estimate produced by the energy-efficiency model, so that higher access energy and higher bandwidth demand naturally translate into stronger thermal constraints. The main inputs and outputs of the energy and thermal models are summarized in Table 3.5.

In this work, the hotspot area is not always assumed to be equal to the full die area. Using the full die area would implicitly assume spatially uniform heat spreading across the entire memory die, which can underestimate localized thermal stress. Instead, the model assumes a localized hotspot around the TSV-dense vertical-interconnect region, motivated by the observation that heat accumulation in 3D-integrated structures is often spatially non-uniform. To represent this behavior, the model uses an allowable local heat-flux range based on reported 3D IC heat-flux values, and interprets the hotspot as the effective region over which the local thermal load must be dissipated.

The thermal analysis begins by estimating the average power dissipated in one memory die. Starting from the total HBM stack power, the model derives the per-die power assigned to a memory die and uses it as the thermal load for the subsequent hotspot analysis. The first key quantity is the required hotspot area:

$$A_{\text{hot,req}}\ [\text{mm}^2] = \frac{P_{\text{cdie}}\ [\text{W}]}{q_{\text{flux}}\ [\text{W/mm}^2]}. \tag{3.36}$$

Here, P_{cdie} is the average power dissipated by one memory die, and q_{flux} is the allowable local heat-flux limit. In this work, the assumed range of q_{flux} is guided by prior 3D IC thermal studies that considered heat-flux conditions on the order of 1–2 W/mm^2 [27, 28]. This formulation reflects the basic thermal intuition that, for a fixed allowable local heat flux, a larger per-die power requires a larger effective hotspot area to remain thermally feasible.

Because the hotspot cannot exceed the physical die boundary, the model next defines the

used hotspot area as

$$A_{\text{hot,used}}\ [\text{mm}^2] = \min\big(A_{\text{hot,req}}\ [\text{mm}^2],\ A_{\text{die}}\ [\text{mm}^2]\big)\,, \tag{3.37}$$

where A_{die} is the physical memory-die area. If the required hotspot area exceeds the available die area, the design point becomes thermally difficult because even full-die spreading would not be sufficient to satisfy the assumed heat-flux limit. This step therefore provides an early indication of whether the design is likely to remain feasible before additional thermal TSVs are inserted.

Once the hotspot area is determined, the model estimates the thermal resistance that must be supported by the vertical heat-removal path. At a high level, the allowable thermal resistance is derived from the available temperature budget and the local hotspot heat flux:

$$q_{\text{hot}}\ [\text{W/mm}^2] = \frac{P_{\text{cdie}}\ [\text{W}]}{A_{\text{hot,used}}\ [\text{mm}^2]} \tag{3.38}$$

$$R_{\text{target}}\ [\text{K mm}^2/\text{W}] \approx \frac{T_{\text{j,max}} - T_{\text{amb}}}{q_{\text{hot}}}, \tag{3.39}$$

where $T_{\text{j,max}}$ is the junction-temperature limit and T_{amb} is the ambient temperature. In other words, the model determines how much vertical thermal resistance can be tolerated before the hotspot exceeds the allowed operating temperature.

The natural vertical conduction path of the die stack is then evaluated using the effective thermal properties of the bulk-silicon and BEOL regions. In the implementation, the BEOL region is represented by an effective composite thermal conductivity, and this is combined with the bulk-silicon path to form an effective vertical thermal conductivity:

$$k_{\text{eff}} = \frac{t_{\text{bulk}} + t_{\text{BEOL}}}{\dfrac{t_{\text{bulk}}}{k_{\text{bulk}}} + \dfrac{t_{\text{BEOL}}}{k_{\text{BEOL}}}}, \tag{3.40}$$

where k_{bulk} and k_{BEOL} denote the thermal conductivities of the bulk and BEOL regions, respectively. The thermal resistance of the natural conduction path is then compared with

the required target resistance. If the native path is insufficient, the model determines how much additional copper area must be introduced to reduce the effective thermal resistance.

$$R_{\text{native}} = \frac{L_{\text{tot}}}{k_{\text{eff}}}, \qquad R_{\text{Cu}} = \frac{L_{\text{tot}}}{k_{\text{Cu}}}, \tag{3.41}$$

where R_{native} is the thermal resistance of the native vertical conduction path and R_{Cu} is the corresponding resistance of an ideal copper-assisted path over the same conduction length L_{tot}.

This additional copper requirement is represented as

$$A_{\text{Cu,req}} = A_{\text{hot,used}} \cdot \frac{(1/R_{\text{target}}) - (1/R_{\text{native}})}{(1/R_{\text{Cu}}) - (1/R_{\text{native}})}, \tag{3.42}$$

where k_{Cu} is the thermal conductivity of copper. Physically, this step estimates how much copper-assisted vertical heat path is needed so that the hotspot can satisfy the target thermal resistance. However, some copper already exists in the hotspot due to previously allocated power, ground, I/O, and internal TSVs. The model therefore subtracts the copper area already provided by these non-thermal TSVs:

$$A_{\text{Cu,add}}\ [\text{mm}^2] = \max\big(A_{\text{Cu,req}}\ [\text{mm}^2] - A_{\text{Cu,exist}}\ [\text{mm}^2],\ 0\big)\,, \tag{3.43}$$

where $A_{\text{Cu,exist}}$ is the existing copper area available inside the hotspot region.

The remaining copper requirement is then translated into additional thermal TSVs:

$$N_{\text{thermal}} = \left\lceil \frac{A_{\text{Cu,add}}\ [\text{mm}^2]}{A_{\text{TSV,Cu}}\ [\text{mm}^2]} \right\rceil, \tag{3.44}$$

where $A_{\text{TSV,Cu}}$ is the copper cross-sectional area of one TSV. In this way, the thermal model directly converts a temperature-driven copper demand into a count of thermal TSVs that can be added to the TSV budget.

Finally, the framework evaluates whether the hotspot remains physically realistic after thermal TSV insertion. For each thermal-conductivity scenario, the model checks whether the required thermal TSVs can be physically accommodated within the final hotspot region. This is done by comparing the total TSV footprint inside the hotspot with the hotspot area

itself. Based on this occupancy, the design point is classified into a thermal status such as *OK*, *Congested*, or *Impossible*. This classification is useful in early-stage HBM exploration because it distinguishes between design points that are merely power-hungry and those that are physically unrealistic due to excessive thermal-TSV demand.

Overall, the thermal model provides the second half of the energy-to-TSV flow. The energy-efficiency model first determines how much power the HBM stack consumes, and the thermal model then determines whether that power can be supported within the die area and temperature budget. If not, the framework introduces additional thermal TSVs to enhance vertical heat removal. This coupling makes it possible to evaluate TSV demand not only from bandwidth and power-delivery requirements, but also from thermal feasibility.

Table 3.6: Parameters and symbols for the TSV budgeting and footprint model. The term η_{mem} is referenced from Table 3.5.

Inputs	**Symbol**	**Description**
Target bandwidth	BW	Target HBM stack bandwidth [bit/s]
Active utilization	U	Useful bandwidth utilization factor
Background power	P_{bg}	Stack-level background power [W]
Supply voltage	V	Nominal supply voltage used for current estimation [V]
TSV current-density limit	$J_{\max}$	Maximum allowable current density per TSV [A/mm^2]
TSV diameter	D_{TSV}	TSV diameter used to derive TSV cross-sectional area [μm]
Interface width	W_{IO}	I/O width per channel [bit]
Signaling rate	R_{IO}	I/O signaling rate per pin [bit/s]
Channels per die	$N_{\text{ch,die}}$	Number of channels implemented in one memory die
Signal TSVs per channel	$N_{\text{sig/ch}}$	Number of signal TSVs allocated per channel
Global TSVs per die	$N_{\text{glob/die}}$	Fixed global TSV overhead per die
TSV pitch / pattern	$P_{\text{TSV}}, \mathcal{P}_{\text{TSV}}$	TSV pitch [μm] and packing pattern used for footprint estimation
Outputs	**Symbol**	**Formula / Description**
Total HBM power	P_{HBM}	$\eta_{\text{mem}} \cdot BW \cdot U + P_{\text{bg}}$ [W]
HBM current	I_{HBM}	P_{HBM}/V [A]
Power TSVs	N_{PWR}	$\lceil I_{\text{HBM}}/(J_{\max} A_{\text{TSV}}) \rceil$
Ground TSVs	N_{GND}	Ground TSV count derived from the power TSV budget
I/O TSVs	N_{IO}	$N_{\text{sig/ch}} \cdot N_{\text{ch,die}} + N_{\text{glob/die}}$
Internal TSVs	N_{internal}	Additional TSVs used for internal vertical interconnect overhead
Thermal TSVs	N_{thermal}	Additional TSVs derived from the thermal model
Total TSV count	$N_{\text{TSV,total}}$	$N_{\text{PWR}} + N_{\text{GND}} + N_{\text{IO}} + N_{\text{internal}} + N_{\text{thermal}}$
TSV array area	A_{TSV}	TSV footprint derived from $N_{\text{TSV,total}}$, TSV pitch, and packing pattern [mm^2]

CHAPTER 4

Model Comparison and Analysis

4.1 Experimental Setup

This chapter evaluates the proposed HBM modeling framework across three generations: HBM2, HBM3, and HBM4. The same unified modeling flow is applied to all generations using generation-specific configuration files, while keeping the overall methodology consistent across memory die, base die, TSV, thermal, and latency analyses.

For HBM2, the model targets an 8 GB stack with 8 memory dies and 256 GB/s bandwidth at a 21.0 nm technology node and 1.2 V nominal voltage. The modeled energy efficiency is 4.6492 pJ/bit. The resulting memory die area is 52.8322 mm^2, with an estimated die dimension of 7.27×7.27 mm, while the base die area is 75.4746 mm^2. The total TSV count is 8,137, and the modeled worst-case latency is 59.5210 ns.

For HBM3, the model targets a 24 GB stack with 12 memory dies and 819 GB/s bandwidth at a 15.8 nm technology node and 1.1 V nominal voltage. The modeled energy efficiency is 2.9066 pJ/bit. The resulting memory die area is 102.9945 mm^2, with an estimated die dimension of 10.15×10.15 mm, while the base die area is 112.9342 mm^2. The total TSV count is 12,903, and the modeled worst-case latency is 60.4301 ns.

For HBM4, the model targets a 64 GB stack with 16 memory dies and 2048 GB/s bandwidth at an 11.2 nm technology node and 1.05 V nominal voltage. The modeled energy efficiency is 2.6502 pJ/bit. The resulting memory die area is 108.8597 mm^2, with an estimated die dimension of 10.43×10.43 mm, while the base die area is 153.0775 mm^2. The total TSV count is 20,898, and the modeled worst-case latency is 74.1885 ns. Since detailed

Table 4.1: Cross-generation summary of modeled HBM2, HBM3, and HBM4 results.

Metric	HBM2 [15]	HBM3 [16]	HBM4 [17]
Total Capacity (GB)	8.0	24.0	64.0
Stack Height (Memory Dies)	8	12	16
Target Bandwidth (GB/s)	256.0	819.0	2048.0
Nominal Voltage (V)	1.20	1.10	1.05
Target Node (nm)	D1y	D1z	D1c
Energy Efficiency (pJ/bit)	4.6492	2.9066	2.6502
Memory Die Area (mm^2)	52.8322	102.9945	108.8597
Memory Die Dimension (mm)	7.27×7.27	10.15×10.15	10.43×10.43
TSV Array Area (mm^2)	7.1276	11.4672	18.6398
Total TSV Count	8137	12903	20898
Base Die Area (mm^2)	75.4746	112.9342	153.0775
Worst Latency (ns)	59.5210	60.4301	74.1885

silicon validation data for HBM4 are not publicly available, the HBM4 results in this chapter should be interpreted as projected estimates derived from public specifications and architectural assumptions. The reported base-die area values correspond to the final integrated footprint after comparing the preliminary base-die result against the modeled memory-die footprint in the top-level flow.

4.2 Cross-Generation Analysis of HBM2, HBM3, and HBM4

Table 4.1 summarizes the key outputs of the proposed framework across HBM2, HBM3, and HBM4. As expected, both memory die and base die area increase substantially from HBM2 to HBM4 as stack capacity, bandwidth, interface width, and vertical interconnect demand increase. The memory die area grows from $52.8322\,mm^2$ in HBM2 to $102.9945\,mm^2$ in

HBM3 and 108.8597 mm^2 in HBM4. Similarly, the base die area increases from 75.4746 mm^2 to 112.9342 mm^2 and 153.0775 mm^2, respectively.

The TSV budget also scales strongly with generation. Total TSV count increases from 8,137 in HBM2 to 12,903 in HBM3 and 20,898 in HBM4, while TSV array area increases from 7.1276 mm^2 to 11.4672 mm^2 and 18.6398 mm^2, respectively. This reflects the combined effect of wider interfaces, increasing internal TSV demand, and stronger thermal-management requirements in later generations.

The energy-efficiency trend improves across generations. The modeled energy efficiency decreases from 4.6492 pJ/bit in HBM2 to 2.9066 pJ/bit in HBM3 and 2.6502 pJ/bit in HBM4, indicating that later generations deliver higher bandwidth at lower energy per transferred bit.

The latency trend is more nuanced. The modeled worst-case latency is 59.5210 ns for HBM2, 60.4301 ns for HBM3, and 74.1885 ns for HBM4. While HBM3 benefits from improved energy efficiency and aggressive scaling, the larger organization and increased structural complexity of HBM4 lead to a higher latency estimate. This result suggests that generational scaling does not uniformly improve all metrics, and that bandwidth scaling must be evaluated together with timing, TSV demand, and physical feasibility.

Thermal feasibility also becomes more challenging with generation scaling. HBM2 selects a conservative thermal TSV scenario with 949 additional thermal TSVs, while HBM3 selects an optimistic scenario with 1,169 additional thermal TSVs. In contrast, all modeled thermal scenarios for HBM4 become impossible under the given assumptions, and the model falls back to a packed-to-100% occupancy case with 1,668 thermal TSVs. This highlights the growing importance of thermal constraints in future HBM stacks.

4.3 Comparison with Existing Models

Table 4.2 compares the proposed framework against published reference data for HBM2 [3] and HBM3 [4]. The results show that the proposed model provides reasonable first-order

Table 4.2: Validation against published reference data for HBM2 [3] and HBM3 [4], with DreamRAM results [5] included where directly comparable. HBM4 is excluded because its results are projection-based rather than directly validated against measured silicon.

Generation	Metric	Reference	DreamRAM	This Work
HBM2	Base die area (mm^2)	71.64	–	75.47
	TSV array area (mm^2)	7.77	–	7.13
	Memory die area (mm^2)	54.44	–	52.83
HBM3	Base die area (mm^2)	121.0	–	112.93
	TSV array area (mm^2)	12.46	–	11.4672
	Memory die area (mm^2)	102.01	111.0	102.9945
	Reported energy (pJ/bit)	–	0.98–3.01	2.9066
	Reported latency (ns)	–	64.2	60.4301

Note: DreamRAM reports HBM3 results for a different design point (16 GB, 1024 GB/s), whereas this work uses a 24 GB, 819 GB/s HBM3 configuration. Accordingly, the DreamRAM energy and latency values are included only as a qualitative reference.

agreement with reported physical dimensions while offering a more structured interpretation of the HBM stack.

For HBM2, the proposed framework estimates the base die area, TSV array area, and memory die area as $75.47\,mm^2$, $7.13\,mm^2$, and $52.83\,mm^2$, respectively, compared with reference values of $71.64\,mm^2$, $7.77\,mm^2$, and $54.44\,mm^2$. For HBM3, the proposed framework estimates the base die area, TSV array size, and memory die area as $112.93\,mm^2$, $11.47\,mm^2$, and $102.9945\,mm^2$, respectively, compared with reference values of $121.0\,mm^2$, $12.46\,mm^2$, and $102.01\,mm^2$. Overall, the model shows good agreement with the reported die-level dimensions while explicitly separating the physical contributions of the base die, TSV region, and memory die.

The reference values in Table 4.2 were reconstructed from published die photographs and reported die dimensions for HBM2 and HBM3 rather than taken directly from a single

tabulated source. In both cases, prior publications provide the overall base-die footprint together with annotated die-shot images that indicate the relative locations of the PHY and TSV regions. Based on these images, the TSV band was identified as a distinct physical block within the die. To obtain a comparable memory-die reference, this work assumes that the TSV region occupies approximately the same area on the base die and the memory die, since the stacked dies must share a vertically aligned TSV interface. Under this assumption, the memory-die area was back-calculated from the published die dimensions by subtracting the estimated TSV region from the reported base-die footprint. Although this reconstruction is approximate, it yields a more physically interpretable reference for HBM stack decomposition.

This explicit reconstruction process is also an important distinction from DreamRAM. In the DreamRAM validation flow, the HBM3 stack was validated using the reported 11 mm $\times$ 11 mm micro-pillar grid area (MPGA) from Samsung Electronics, and this MPGA size was treated as the primary die-area reference [5]. As a result, DreamRAM compared its modeled die area against an overall stack-level footprint without explicitly separating the logic base die, TSV band, and memory die contributions. This approach is reasonable as a first-order validation target, but it does not provide a direct breakdown of where the physical area is consumed within the stack.

In contrast, the proposed framework models the memory die and base die separately and further identifies the contribution of the TSV array. This makes the resulting area estimates more interpretable for HBM-specific architectural exploration. Instead of comparing a single lumped die metric against the reported MPGA, the proposed model allows the designer to identify whether the dominant overhead comes from the memory array, peripheral circuitry, TSV footprint, or base-die-related constraints such as PHY and bump-map requirements.

At the same time, the proposed framework maintains a modeling scope that is comparable to prior analytical frameworks in terms of timing and efficiency metrics. Similar to DreamRAM, the model estimates latency-related quantities such as worst-case access latency and captures energy efficiency in pJ/bit [5]. Therefore, the proposed framework preserves

the ability to study performance- and efficiency-related trends while improving the physical interpretability of area modeling for HBM stacks.

Compared with existing frameworks such as CACTI-3DD and DreamRAM, the proposed model is more explicitly oriented toward HBM-specific architectural exploration. CACTI-3DD provides useful array-level estimates for stacked-memory structures, while DreamRAM models detailed DRAM hierarchy, timing, and efficiency [5, 9]. However, the proposed framework extends this style of analysis by jointly treating memory die hierarchy, base die constraints, TSV allocation, thermal feasibility, and energy efficiency within a unified HBM-focused modeling flow. This makes the framework particularly suitable for both cross-generation comparison and forward-looking estimation of emerging standards such as HBM4.

CHAPTER 5

Case Studies

This chapter presents several case studies conducted using the proposed HBM modeling framework. The purpose of these studies is to examine how bandwidth scaling, TSV design parameters, heterogeneous integration scenarios, and stack organization influence the physical feasibility of future HBM systems. While Chapter 4 focused on validating the baseline models of HBM2, HBM3, and HBM4, the studies in this chapter use those validated baselines as starting points for design-space exploration. Together, these experiments illustrate how TSV allocation, die area, power delivery, and thermal feasibility evolve as stacked-memory architectures are pushed beyond their current operating points.

5.1 Bandwidth Scaling, TSV Demand, and Die-Area Impact

Memory bandwidth continues to increase in modern computing systems driven by data-intensive workloads such as artificial intelligence, large-scale analytics, and high-performance accelerators. In HBM architectures, higher bandwidth is typically achieved through wider interfaces, more channels, and higher effective signaling throughput. However, bandwidth scaling also increases vertical interconnect requirements, stack power, and thermal burden. As a result, bandwidth growth must be evaluated not only from an interface perspective but also from the viewpoint of TSV demand and physical die footprint.

To study this effect, the target bandwidth was swept while other architectural and technology parameters were kept fixed for each HBM generation. Figure 5.1 shows the resulting TSV demand for HBM2, HBM3, and HBM4 as bandwidth increases.

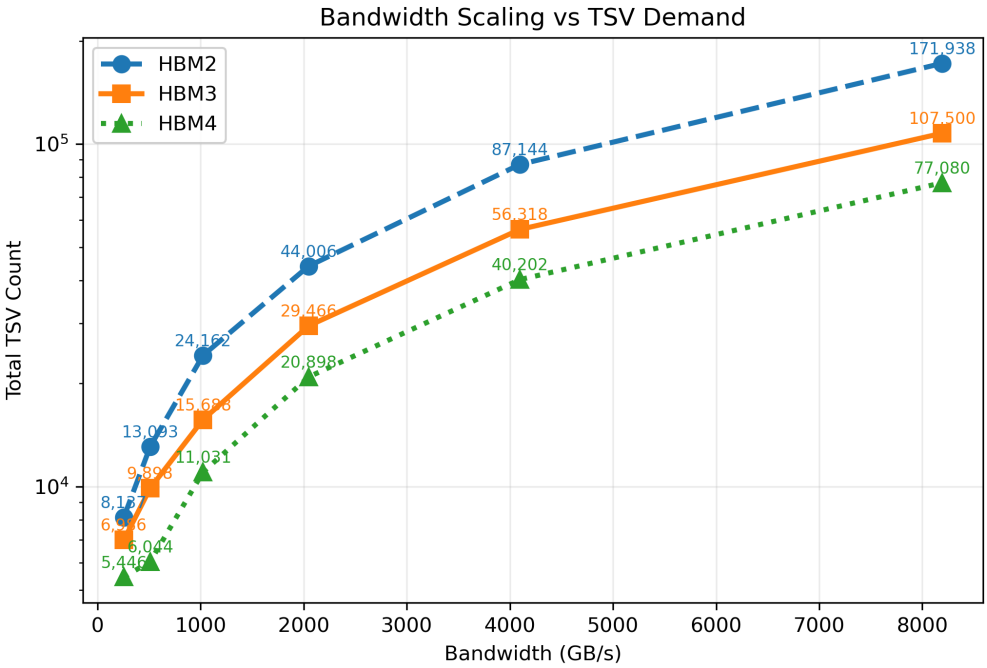


Figure 5.1: Bandwidth scaling versus TSV demand across HBM generations.

The results show that TSV demand increases rapidly as bandwidth scales upward. Although signal TSVs grow with the required interface width and channel activity, the total TSV count grows more strongly because higher bandwidth also increases power-delivery TSV and thermal TSV requirements. This effect becomes more pronounced in later generations, where the baseline bandwidth and stack power are already high. Therefore, TSV resources emerge as a primary physical scaling constraint as HBM bandwidth increases.

The increase in TSV demand also affects the memory-die footprint. Figure 5.2 shows the relationship between total TSV count and memory die area for the HBM3 baseline under bandwidth scaling.

As bandwidth increases, the memory die area also increases because the TSV array occupies a larger fraction of the die footprint. Compared with the rapid growth in TSV count, the increase in die area is more gradual, but it is still significant because it directly influences whether the memory die can remain within the footprint budget imposed by the base die.

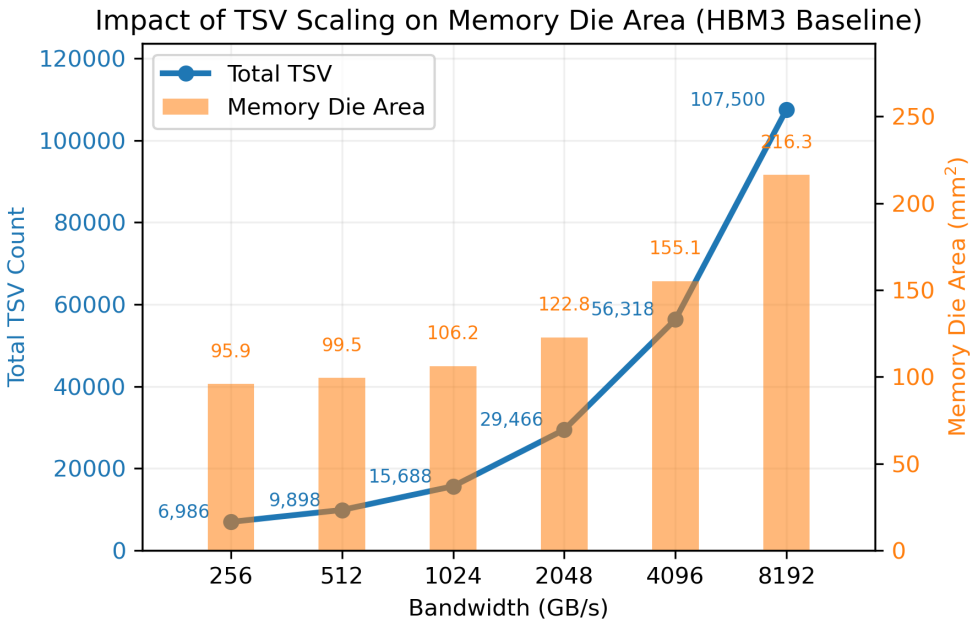


Figure 5.2: Impact of TSV scaling on memory die area for the HBM3 baseline configuration.

To further quantify this limit, the current modeled base-die area of each HBM generation was treated as a fixed footprint constraint, and the maximum bandwidth that could be supported before the memory die area exceeded this limit was estimated. Figure 5.3 summarizes the resulting bandwidth headroom under the current base-die footprint for HBM2, HBM3, and HBM4.

The result shows that the current HBM2 and HBM3 configurations still have additional bandwidth headroom before the memory die area reaches the modeled base-die footprint limit. In contrast, the HBM4 baseline already operates much closer to a high-bandwidth regime, although it still retains substantial headroom under the assumed footprint. This experiment highlights an important design implication: bandwidth scaling is not only limited by interface and TSV demand, but also by the extent to which TSV-driven memory-die area can remain compatible with the base-die footprint.

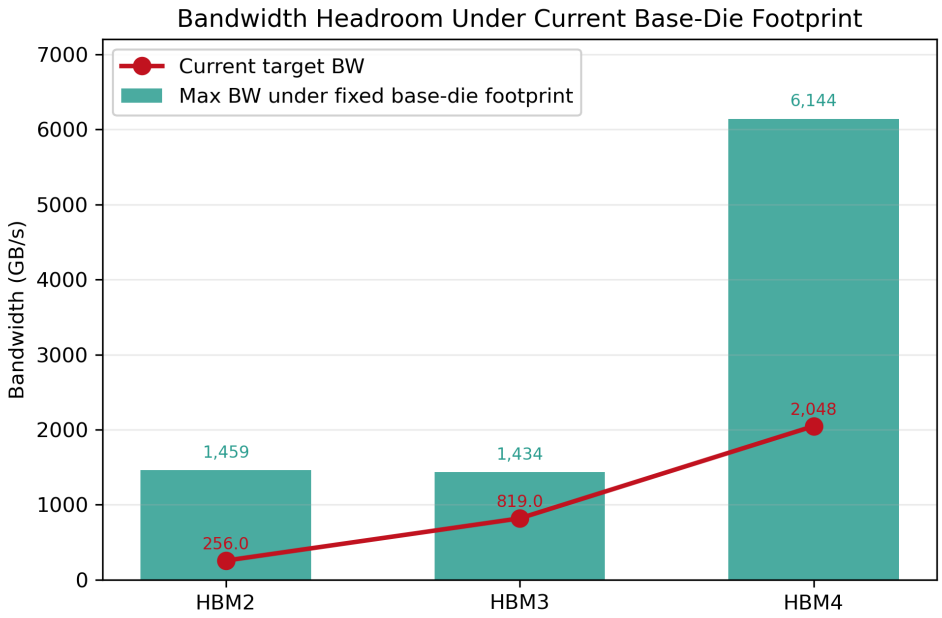


Figure 5.3: Maximum bandwidth that can be supported before the memory die area exceeds the current modeled base-die footprint.

5.2 TSV Pitch, TSV Allocation, and Area Trade-offs

The physical design parameters of TSVs strongly affect the feasibility of stacked memory systems. Among these parameters, TSV pitch is particularly important because it determines how densely TSVs can be placed within a given region. TSV pitch therefore influences not only TSV array footprint but also the number of thermal TSVs required to maintain thermal feasibility.

To evaluate this trade-off, the TSV pitch was swept for the HBM3 baseline configuration. The study tracks how TSV pitch affects thermal TSV count, TSV array area, and total memory die area. The results are shown in Figure 5.4.

As TSV pitch increases, the available TSV density decreases, which makes it more difficult to place enough TSVs within the hotspot region. Consequently, the required number of thermal TSVs changes significantly with pitch. At the same time, the TSV array area grows as pitch increases because each TSV occupies more lateral footprint. This area increase

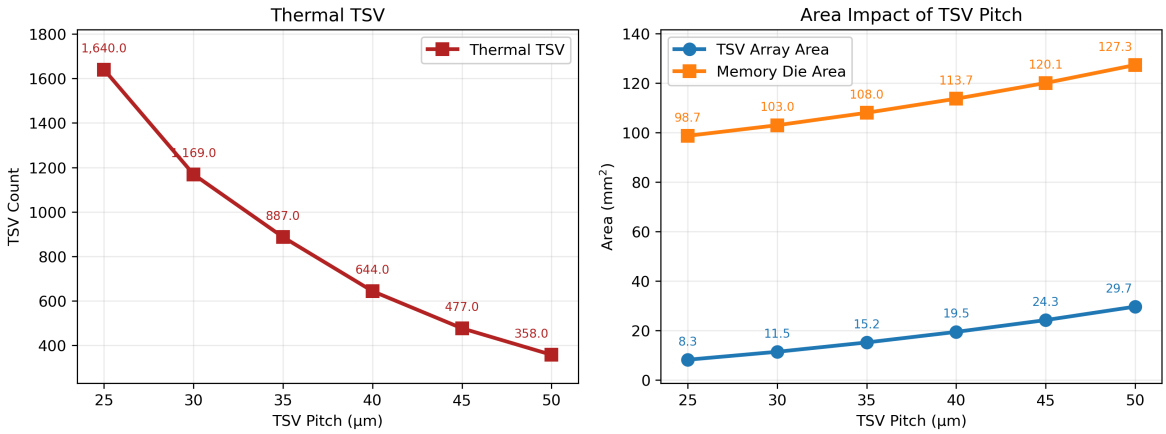


Figure 5.4: Effect of TSV pitch on thermal TSV count, TSV array area, and memory die area for the HBM3 baseline.

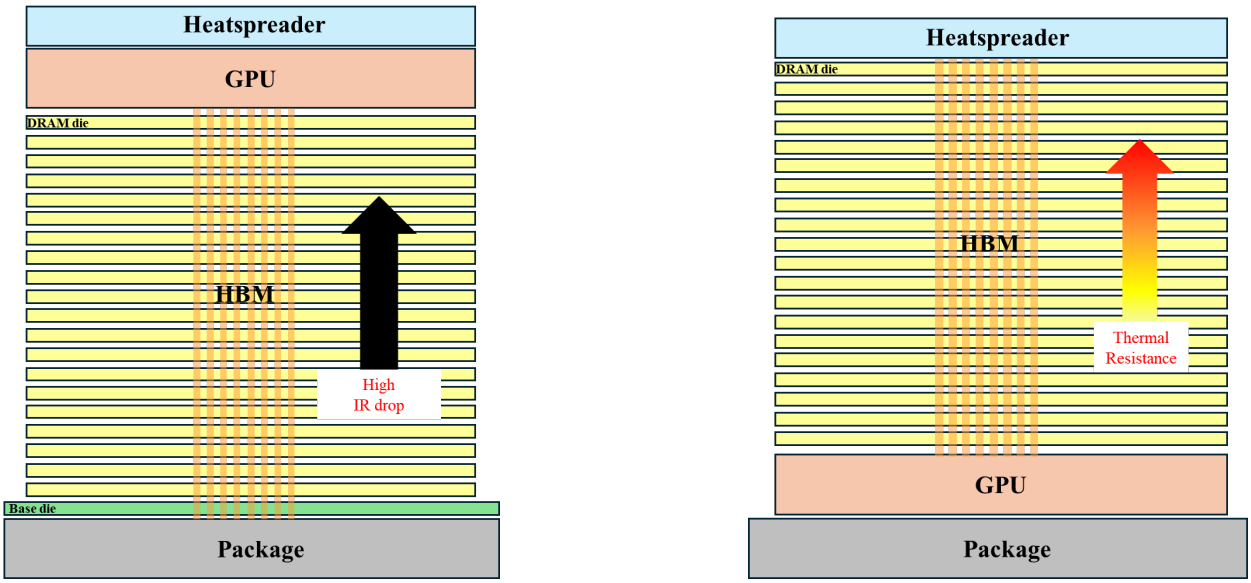
propagates directly into the final memory die area.

These results highlight that TSV pitch must be chosen through a joint physical and thermal design trade-off. Smaller pitches improve vertical interconnect density and reduce footprint, but may introduce fabrication complexity. Larger pitches simplify manufacturing assumptions, but increase TSV array footprint and can make thermal TSV allocation more difficult. Therefore, TSV pitch is not merely a process parameter, but an architectural design choice that directly affects HBM feasibility.

5.3 Logic Stacking as a System-Level Case Study

Recent work has explored heterogeneous integration of stacked memory and compute logic to improve bandwidth efficiency and reduce communication distance in AI and accelerator systems. In particular, two integration styles are receiving significant attention: placing the GPU above the HBM stack (GPU-on-HBM) and placing the HBM stack above the GPU (HBM-on-GPU). These two configurations create fundamentally different physical bottlenecks.

Prior studies have shown that these two arrangements create distinct thermal and elec-



(a) GPU-on-HBM

(b) HBM-on-GPU

Figure 5.5: Conceptual comparison of two logic-stacking scenarios. GPU-on-HBM primarily stresses power delivery through the memory stack, while HBM-on-GPU increases thermal resistance along the vertical heat-removal path. Adapted from [2].

trical challenges. In GPU-on-HBM integration, delivering GPU power through the memory stack can increase IR drop and power-delivery TSV requirements. In HBM-on-GPU integration, the memory stack sits farther from the cooling boundary, which increases thermal resistance and can substantially raise the required thermal TSV count [2, 29]. The physical difference between these two cases is illustrated in Figure 5.5.

To quantify these effects, the stacked logic power, referring to the power of the stacked logic die such as a graphics processing unit (GPU) or central processing unit (CPU) rather than the HBM stack itself, was swept for both HBM3 and HBM4 under the two logic-stacking assumptions. Figure 5.6 shows the resulting power TSV and thermal TSV components. Figure 5.7 shows the resulting total TSV demand, and Figure 5.8 reports the corresponding minimum required memory die area.

The results show a clear asymmetry between the two integration styles. In the GPU-on-HBM case, power TSV demand increases strongly with stacked logic power because the

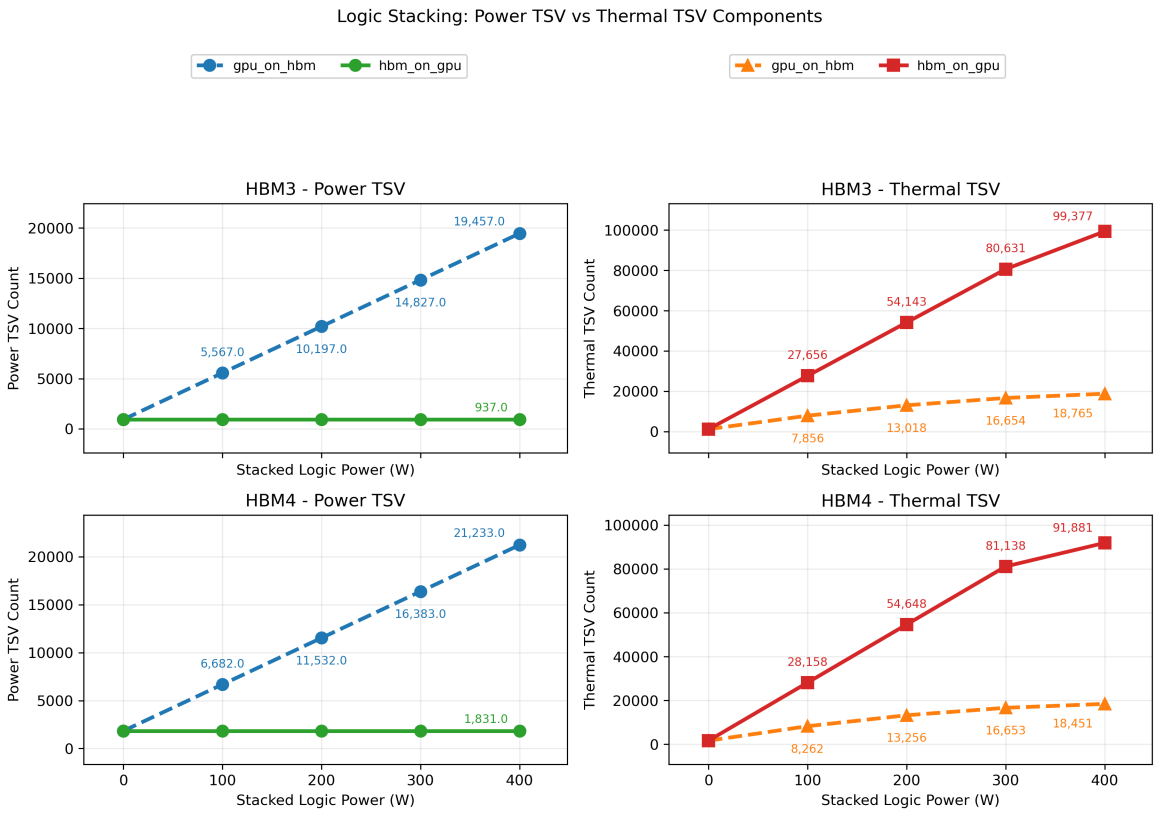


Figure 5.6: Power TSV and thermal TSV components under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4.

additional logic power must be delivered through the HBM stack. In contrast, the HBM-on-GPU case produces a much larger increase in thermal TSV demand because the thermal path becomes more resistive when the memory stack is placed farther from the heat-spreading boundary. In both HBM3 and HBM4, the thermal TSV count in the HBM-on-GPU case grows much more aggressively than in the GPU-on-HBM case.

These differences are also reflected in the total TSV demand shown in Figure 5.7. As stacked logic power increases, both scenarios require additional TSV resources, but the underlying causes differ. GPU-on-HBM is primarily power-delivery-limited, whereas HBM-on-GPU is primarily thermal-limited.

This difference directly affects physical footprint, as shown in Figure 5.8. In both HBM3 and HBM4, the HBM-on-GPU case reaches a larger minimum required die area because

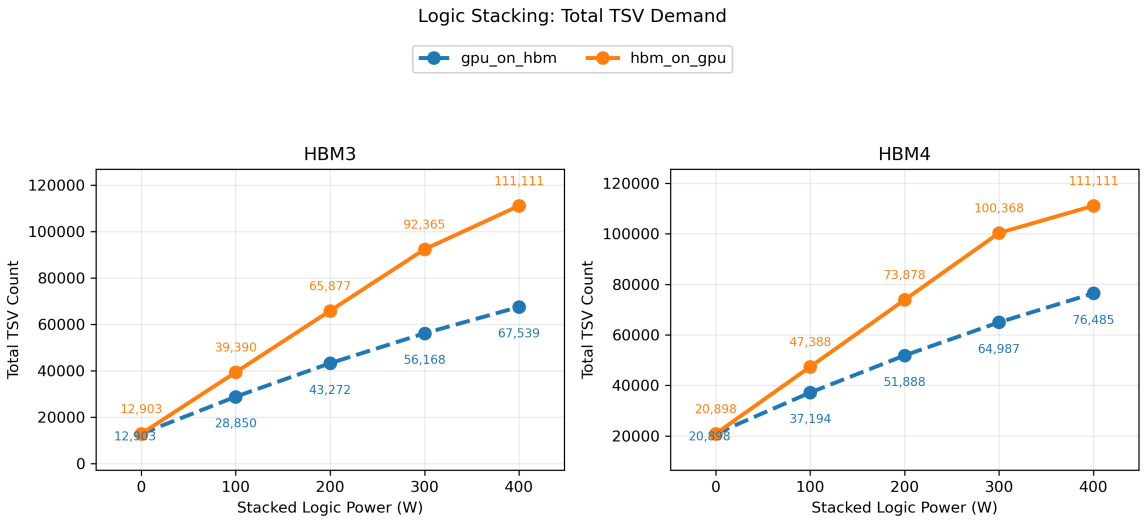


Figure 5.7: Total TSV demand under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4.

of the rapid growth in thermal TSV demand. The dashed fixed base-die limits further indicate when the required memory-die footprint would exceed the currently modeled base-die footprint.

This system-level case study shows that logic stacking cannot be evaluated solely from the perspective of bandwidth improvement. The placement of the compute die relative to the memory stack changes the dominant physical bottleneck and therefore changes the required TSV budget and minimum die footprint. These results demonstrate the importance of jointly considering power delivery and thermal management in future heterogeneous HBM integration.

5.4 Stack Height Scaling

Another important architectural parameter in HBM design is stack height. Increasing the number of stacked memory dies improves storage density and can make better use of package footprint. However, larger stack height also increases the accumulated stack power and the vertical path through which both current and heat must travel.

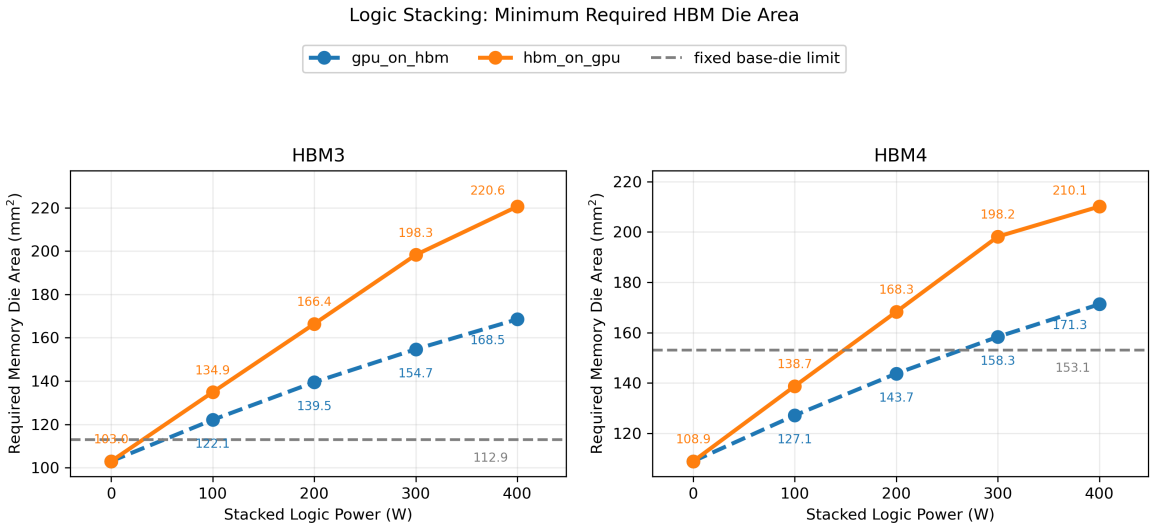


Figure 5.8: Minimum required memory die area under GPU-on-HBM and HBM-on-GPU stacking scenarios for HBM3 and HBM4. The dashed line denotes the fixed base-die footprint limit.

To study this effect, the number of stacked memory dies was varied for the HBM3 baseline configuration while keeping the remaining technology assumptions fixed. Figure 5.9 shows the resulting total TSV demand.

As stack height increases, the total TSV count also increases. The increase is strongest when moving from a short stack to a more realistic multi-die HBM organization, after which the TSV count continues to rise more gradually. This trend indicates that vertical scaling introduces additional TSV overhead even when the bandwidth target and the process assumptions remain fixed.

The same figure also shows the impact of stack height on memory die area and worst-case latency. As the stack becomes taller, the memory die area increases modestly, reflecting the additional TSV burden that must be integrated into the die footprint. The worst-case latency also increases, although the change is relatively small compared with the change in TSV demand. This indicates that stack-height scaling primarily stresses the physical interconnect budget before it severely degrades timing.

Overall, the stack-height study shows that increasing stack depth cannot be evaluated

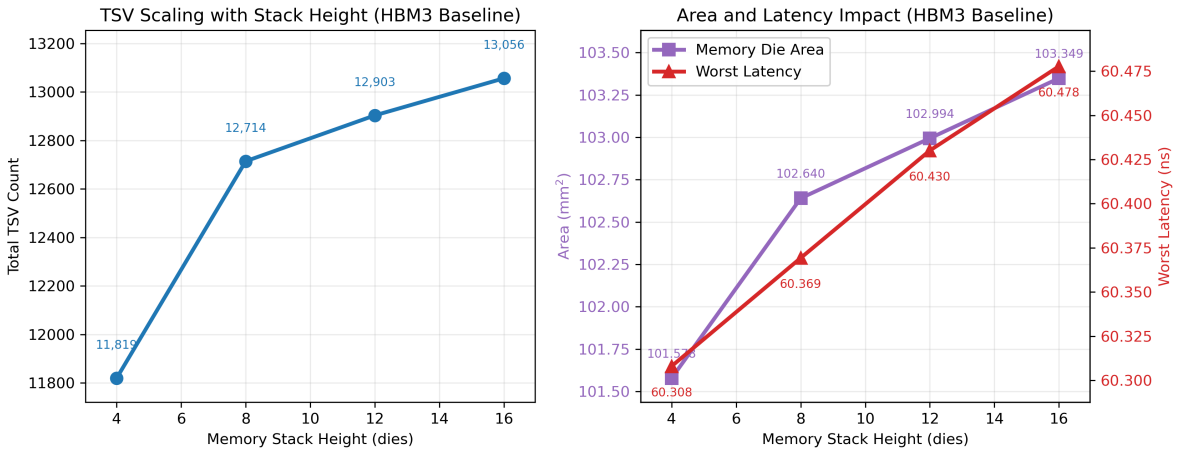


Figure 5.9: Impact of stack height on TSV demand, memory die area, and worst-case latency for the HBM3 baseline configuration.

solely from a capacity perspective. Taller stacks improve density, but they also require more TSV resources and modestly increase die area and latency. Therefore, stack height should be treated as a coupled architectural and physical-design parameter in future HBM systems.

CHAPTER 6

Conclusion and Future Work

6.1 Summary

This thesis presented a modeling framework for architecture-level exploration of High-Bandwidth Memory (HBM) systems. The framework was motivated by the growing need for systematic analysis of stacked memory architectures as memory bandwidth demand, stack complexity, and physical design constraints continue to increase in modern computing systems. While existing DRAM modeling approaches often focus primarily on timing behavior, controller-level performance, or array-level estimation, modern HBM design requires a more integrated view that jointly considers memory organization, vertical interconnect resources, power delivery, thermal feasibility, and die footprint.

To address this need, the proposed framework models the major physical and architectural components of HBM within a unified flow. The memory die model captures the internal hierarchy of stacked DRAM layers, including channels, pseudo-channels, banks, mats, and subarrays, and estimates both core-array and peripheral-circuit area together with worst-case latency. The base die model captures PHY area, testing-related structures, TSV landing regions, and µbump footprint constraints, allowing the final base-die size to reflect both logic-area accumulation and package-level placement limits. The TSV model explicitly budgets power, ground, I/O, internal, and thermal TSVs rather than treating TSV overhead as a single lumped quantity. In addition, the thermal and energy models extend the framework beyond geometric estimation by incorporating stack-level power consumption, thermal TSV requirements, and first-order energy-efficiency analysis.

The framework was evaluated across HBM2, HBM3, and HBM4 configurations. For HBM2 and HBM3, the modeled results showed reasonable first-order agreement with published reference data in terms of memory die area, TSV array area, and base die dimensions. For HBM4, where direct silicon validation data are not yet publicly available, the framework was used as a forward-looking projection tool based on public specifications and architectural assumptions. The cross-generation analysis showed that newer HBM generations require substantially larger TSV budgets and larger die footprints even as energy efficiency improves. The results also showed that generational scaling does not uniformly improve all metrics: while bandwidth and efficiency improve, latency and thermal feasibility become increasingly challenging as stack organization and interconnect demand grow.

The case studies in this thesis further demonstrated the usefulness of the framework for design-space exploration. First, the bandwidth-scaling study showed that increasing bandwidth rapidly increases total TSV demand and gradually increases memory-die area, making TSV budgeting and die footprint central constraints in future high-bandwidth HBM designs. The same study also quantified the bandwidth headroom available under the currently modeled base-die footprint, showing that TSV-driven area growth can eventually limit further bandwidth scaling even when interface-level scaling remains possible.

Second, the TSV-pitch study showed that TSV pitch is not merely a layout parameter but a coupled architectural and physical-design variable. Larger pitch increases TSV array footprint and changes the thermal TSV requirement, while smaller pitch improves density but may imply tighter manufacturing assumptions. This result highlights the importance of jointly considering interconnect density, thermal feasibility, and die area when selecting TSV geometry.

Third, the logic-stacking case study showed that heterogeneous integration changes the dominant physical bottleneck depending on relative die placement. In the GPU-on-HBM case, stacked logic primarily increases power-delivery TSV demand because the additional logic power must be supplied through the memory stack. In the HBM-on-GPU case, the dominant burden shifts to thermal TSV demand because the memory stack becomes farther

from the cooling boundary. These different TSV trends also translate into different minimum required memory-die areas, demonstrating that logic stacking must be evaluated not only from the viewpoint of bandwidth benefit but also from the perspective of power delivery and thermal management.

Finally, the stack-height study showed that increasing the number of stacked memory dies cannot be evaluated solely as a capacity scaling mechanism. Taller stacks increase TSV demand and modestly increase memory-die area and worst-case latency even when the bandwidth target and process assumptions are held fixed. This indicates that vertical scaling introduces non-negligible interconnect and physical-design overheads that must be considered early in HBM architecture exploration.

Taken together, the results of this thesis show that HBM design is governed by strongly coupled interactions among bandwidth scaling, TSV allocation, power delivery, thermal feasibility, and die footprint. The proposed modeling framework makes these interactions explicit and provides an interpretable tool for studying both current and emerging HBM architectures. As future memory systems move toward even higher bandwidth, taller stacks, and tighter integration with compute logic, such physically informed modeling will become increasingly important for early-stage architecture exploration.

6.2 Future Work

There are several directions in which this work can be extended.

First, the timing model can be refined further and calibrated more tightly against measured silicon data and detailed public timing specifications. Although the current framework provides architecture-aware latency estimation, future work could incorporate more explicit modeling of row-decoder delay, local and global bitline propagation, sense-amplifier behavior, TSV loading along different paths, and I/O serialization effects. Such refinements would improve the interpretability of timing trends across generations and under non-baseline organizations.

Second, the power-delivery model can be extended beyond TSV counting to include a more detailed electrical analysis of the full vertical power-distribution network. In the current framework, power and ground TSV budgets are estimated from stack-level current demand, but future work could incorporate IR-drop analysis, current-density distribution, voltage-noise estimation, and transient load behavior across the stack. This would be particularly important for analyzing heterogeneous integration scenarios in which high-power logic dies are placed above or below HBM stacks.

Third, the thermal model can be improved by incorporating more detailed spatial and workload-aware analysis. The current implementation uses a compact hotspot-oriented feasibility model, which is suitable for early-stage exploration, but future work could add spatial thermal maps, layer-by-layer temperature gradients, non-uniform activity factors, and dynamic interactions between memory and stacked logic. Such an extension would enable more realistic study of localized thermal bottlenecks in advanced 3D-integrated systems.

Fourth, the framework can be expanded toward reliability-aware modeling. As HBM systems continue to scale in bandwidth, stack height, and heterogeneous integration complexity, reliability challenges such as TSV failure, electromigration, thermal stress, process variation, and long-term degradation are likely to become more significant. Integrating these factors would allow the framework to evaluate not only nominal feasibility, but also design robustness over time.

Fifth, the current methodology can be extended toward broader package- and system-level co-design. Although this thesis focused on HBM stack modeling, the same principles can be applied to systems involving interposers, chiplets, logic-on-memory integration, and other advanced 2.5D and 3D packaging schemes. Extending the framework to jointly model memory stacks, compute dies, package interconnects, and cooling boundaries would enable more comprehensive exploration of future heterogeneous computing platforms.

Finally, future versions of the framework could incorporate automated design-space exploration and optimization. By integrating parameter sweeps, sensitivity analysis, and multi-objective optimization, the framework could identify Pareto-efficient design points under

simultaneous constraints on bandwidth, latency, area, TSV allocation, and thermal feasibility. Such an extension would further improve the usefulness of the framework as a practical tool for next-generation HBM architecture research.

REFERENCES

[1] Digital Foundry, "Scaling the Memory Wall: The Rise and Roadmap of HBM." `https://newsletter.semianalysis.com/p/scaling-the-memory-wall-the-rise-and-roadmap-of-hbm`, 2025. [Online; accessed 2025-12-20].

[2] J. Sharda, M. Manley, J. Kwak, C. Park, M. Bakir, and S. Yu, "3d stacked hbm and compute accelerators for llm: Optimizing thermal management and power delivery efficiency," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. PP, pp. 1–1, 01 2025.

[3] J. C. Lee, J. Kim, K. W. Kim, Y. J. Ku, D. S. Kim, C. Jeong, T. S. Yun, *et al.*, "High bandwidth memory (hbm) with tsv technique," in *International SoC Design Conference (ISOCC)*, pp. 181–182, IEEE, 2016.

[4] Y. Ryu, S.-G. Ahn, J. H. Lee, J. Park, Y. K. Kim, H. Kim, Y. G. Song, *et al.*, "A 16 gb 1024 gb/s hbm3 dram with source-synchronized bus design and on-die error control scheme for enhanced ras features," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 4, pp. 1051–1061, 2023.

[5] V. Cai, J. Zhou, H. Do, D. Brooks, and G.-Y. Wei, "Dreamram: A fine-grained configurable design space modeling tool for custom 3d die-stacked dram," *arXiv preprint arXiv:2512.12106*, 2025.

[6] H. Park, K. Chang, J. Jeong, J. Ahn, K.-S. Chung, and T. Kim, "Challenges on dtco methodology towards deep submicron interconnect technology." 2021 18th International SoC Design Conference (ISOCC), 2021.

[7] H. Jun, J. Cho, K. Lee, H.-Y. Son, K. Kim, H. Jin, and K. Kim, "Hbm (high bandwidth memory) dram technology and architecture," *Institute of Electrical and Electronics Engineers (IEEE)*, pp. 1–4, 2017.

[8] M. O'Connor, N. Chatterjee, D. Lee, J. D. Wilson, A. A. Agrawal, S. W. Keckler, and W. J. Dally, "Fine-grained dram: Energy-efficient dram for extreme bandwidth systems," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 41–54, 2017.

[9] K. Chen, S. Li, N. Muralimanohar, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 33–38, IEEE, 2012.

[10] C. Weis, N. Wehn, L. Igor, and L. Benini, "Design space exploration for 3d-stacked drams," in *2011 Design, Automation & Test in Europe*, pp. 1–6, 2011.

[11] Y. Kim, W. Yang, and O. Mutlu, "Ramulator: A fast and extensible DRAM simulator," *IEEE Computer Architecture Letters*, vol. 15, no. 1, pp. 45–49, 2016.

[12] S. Li, Z. Yang, D. Reddy, A. Srivastava, and B. Jacob, "DRAMsim3: A Cycle-Accurate, Thermal-Capable DRAM Simulator," *IEEE Computer Architecture Letters*, 2020.

[13] L. Steiner, M. Jung, F. S. Prado, K. Bykov, and N. Wehn, "Dramsys4.0: An open-source simulation framework for in-depth dram analyses," *International Journal of Parallel Programming*, vol. 50, no. 2, pp. 217–242, 2022.

[14] A. Deb, P. Faraboschi, A. Shafiee, N. Muralimanohar, R. Balasubramonian, and R. Schreiber, "Enabling technologies for memory compression: Metadata, mapping, and prediction," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pp. 17–24, 2016.

[15] JEDEC Solid State Technology Association, "High bandwidth memory (hbm) dram specification." JESD235D, 2020.

[16] JEDEC Solid State Technology Association, "High bandwidth memory (hbm) dram specification." JESD238B, 2024.

[17] JEDEC Solid State Technology Association, "High bandwidth memory (hbm) dram specification." JESD270-4, 2025.

[18] T. TAKAHASHI, "A multi-gigabit dram technology with 6f2 open-bit-line cell distributed over-driven sensing and stacked-flash fuse," *ISSCC Dig. of Tech. Papers, Feb. 2001*, 2001.

[19] N. Chatterjee, M. O'Connor, D. Lee, D. R. Johnson, S. W. Keckler, M. Rhu, and W. J. Dally, "Architecting an energy-efficient dram system for gpus," in *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 73–84, IEEE, 2017.

[20] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 predictive design kit development and cell design technology co-optimization," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 992–998, IEEE, Nov. 2017.

[21] J. Jeong and T. Kim, "Utilizing middle-of-line resource in filler cells for fixing routing failures," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 228–231, IEEE, 2021.

[22] J. Jeong, S. Chung, K. Jo, and T. Kim, "Synthesis and utilization of standard cells amenable to gear ratio of gate-metal pitches for improving pin accessibility," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2023.

[23] J. Jeong and T. Kim, "Placement legalization for heterogeneous cells of non-integer multiple-heights," *Integration*, vol. 97, p. 102177, 2024.

[24] J. Jeong and T. Kim, "Binding multi-bit flip-flop cells through design and technology co-optimization," in *Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, ACM/IEEE, 2024.

[25] J. Jeong and T. Kim, "Machine learning driven early clustering for multi-bit flip-flop allocation," in *2025 ACM/IEEE 7th Symposium on Machine Learning for CAD (ML-CAD)*, pp. 1–7, IEEE, 2025.

[26] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of cmos device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, 2017.

[27] S. S. Salvi and A. Jain, "A review of recent research on heat transfer in three-dimensional integrated circuits (3-d ics)," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 5, pp. 802–821, 2021.

[28] S. G. Kandlikar and A. Ganguly, *Fundamentals of Heat Dissipation in 3D IC Packaging and Thermal-Aware Design*, pp. 369–395. Singapore: Springer Singapore, 2021.

[29] Y. Chen, M. Lofrano, D. Moolchandani, H. Oprins, G. Van Der Plas, J. Ryckaert, D. Biswas, and J. Myers, "Breaking thermal bottleneck in 3d hbm-on-gpu integration via system-technology co-optimization," in *2025 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, 2025.