**2.1 Basic DRAM Structure & Architecture Overview**

To meet the demands of modern deep learning (DL) workloads, on-chip memory stacking has emerged as a promising solution. Several innovative 3D DRAM architectures have been developed to demonstrate the viability of stacked memory designs. These architectures leverage three-dimensional (3D) integrated circuit technology, utilizing through-silicon vias (TSVs) to vertically interconnect multiple layers. This vertical stacking significantly reduces memory access latency and increases bandwidth by overcoming the limitations of conventional I/O pins. In the following, we explore a range of 3D-DRAM architectures and examine the critical role of TSV technology in their implementation.

2.1.1 DRAM Architecture

A typical DRAM chip is organized hierarchically into multiple **banks**, each of which contains several **subarrays** and shares a central **I/O peripheral** interface for data access. As shown in Figure 2.1.1, four banks are connected to a shared I/O peripheral that handles communication over a common bus. Within each bank, **subarrays** are stacked vertically and managed by peripheral logic such as local decoders and sense amplifiers.

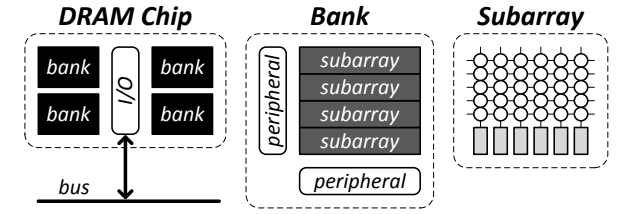
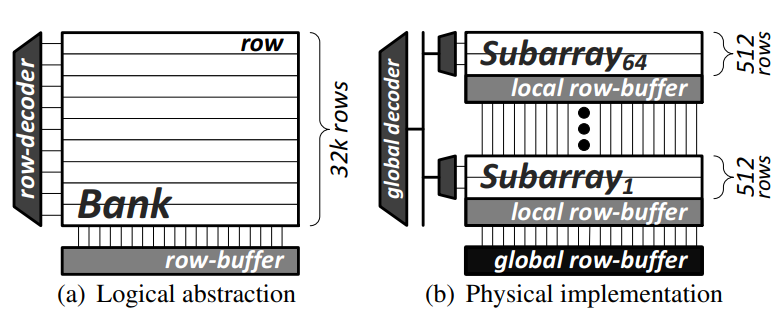


Figure 2.1.2 provides both a **logical abstraction** and a **physical implementation** of a DRAM bank. Logically, a bank is composed of thousands of rows (e.g., 32K rows), which are addressed through a row decoder and accessed via a global row buffer. Physically, these rows are distributed across multiple **subarrays** (e.g., 64 subarrays with 512 rows each). Each subarray contains a **local row buffer** that temporarily holds the accessed row data. During an **activation operation**, data from the local row buffer is transferred to the **global row buffer**, enabling subsequent read or write operations. This hierarchical buffering improves both area efficiency and access latency.



Each subarray is further divided into **mats**, where every mat contains its own set of sense amplifiers and wordlines, forming the basic units of data storage and access.

一張含有 文字, 圖表, 行, 方案 的圖片

自動產生的描述

2.1.2 DRAM Command, Operations

The following figure explain the basic DRAM access operations

**1. Row Activation (ACTIVATE)**  
Before accessing a row, the DRAM bank must be in the **precharged state**, where all bitlines are equilibrated to **1/2VDD**. Upon receiving an **ACTIVATE** command with a specific row address, the corresponding **wordline** is elevated to **VPP**, connecting the memory cells of the target row to the bitlines. This connection causes a slight voltage disturbance on the bitlines, depending on whether the cells are charged (1) or discharged (0). The **sense amplifiers** in the **row buffer** detect this perturbation and amplify it to full logic levels, latching the row data. During this sensing phase, cell contents are temporarily undefined. Once amplification completes, the original data is restored to the cells. The total duration of this operation is defined by the timing parameter **tRAS** (Row Access Strobe), typically around **35 ns**.

**2. Column Access (READ/WRITE)**  
After activation, the **memory controller** issues a **READ** or **WRITE** command with a column address to access specific data within the activated row. A timing delay, denoted as **tRCD** (Row to Column Delay), typically **~15 ns**, must elapse between the ACTIVATE and column command. This delay allows the row buffer to fully capture and stabilize the data. If a subsequent access targets the same row, it can proceed immediately with a READ/WRITE command, avoiding a new activation and thus reducing latency.

**3. Bank Precharge (PRECHARGE)**  
To access a different row within the same bank, the current row must first be **closed** by returning the bank to the **precharged state**. This process involves two steps: the active **wordline is lowered** to disconnect the memory cells, and the **bitlines are re-equalized** to **1/2VDD**. The time required for this operation is captured by the **tRP** (Row Precharge) parameter, typically around **15 ns**.

一張含有 文字, 字型, 圖表, 行 的圖片

自動產生的描述

**4. Refresh(REF)**

DRAM requires periodic refresh operations to preserve data integrity, as each cell gradually leaks charge over time. These refreshes momentarily block access to the corresponding banks or rows, making it essential for the memory controller to schedule them efficiently without significantly degrading performance.

**2.1.3 DRAM Timing Constraints**

The execution of DRAM operations is strictly governed by a set of timing constraints, which are essential for maintaining correct and reliable memory behavior. These constraints are determined by the internal architecture of the DRAM device and are specified by the DRAM manufacturer based on worst-case conditions, ensuring robust operation under all supported scenarios.

As shown in Figure X, each timing parameter defines the minimum delay between specific command sequences—such as tRCD (ACTIVATE to READ), tRTP (READ to PRECHARGE), or tWR (WRITE to PRECHARGE)—and must be respected by the memory controller to prevent data corruption and maintain signal integrity across banks, rows, and I/O channels.

A memory controller is responsible for enforcing all of these timing constraints while simultaneously striving to maximize performance, typically through techniques such as request reordering, bank-level parallelism, and row-buffer locality optimization.

The table below summarizes additional critical timing constraints, including **tRC** (row-to-row activation delay), **tRP** (precharge-to-activate delay), **tRTW** and **tWTR** (read/write turnaround times), as well as **CL** (CAS latency) and **CWL** (CAS write latency), which represent the delay from column commands to data transfer.一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

**2.2 3D-DRAM key architecture TSV technology and state-of-art DRAM**

**一張含有 文字, 螢幕擷取畫面, 平行, 設計 的圖片

自動產生的描述**

**FGDRAM**

**一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述**

**一張含有 圖表, 圖畫, 文字, 寫生 的圖片

自動產生的描述**

**FGR,HalfDRAM**

**一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 圖表, 地圖 的圖片

自動產生的描述**

**一張含有 螢幕擷取畫面, 圖表, 設計, 圖解 的圖片

自動產生的描述** 一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

**2.3 Modern 3D-DRAM Controller**