Memory Hierarchy Design – Main Memory

Presentation Outline

- **Motivation**
- Random Access Memory and its Structure

Removing The Ideal Memory Assumption

- So far we have assumed that ideal memory is used for both instruction and data memory in all CPU designs considered:
  - Single Cycle, Multi-cycle, and Pipelined CPUs.
- Ideal memory is characterized by a short delay or memory access time (one cycle) comparable to other components in the datapath.
  - i.e 2ns which is similar to ALU delays.
- Real memory utilizing Dynamic Random Access Memory (DRAM) has a much higher access time than other datapath components (80ns or more).

Memory Access Time >> 1 CPU Cycle

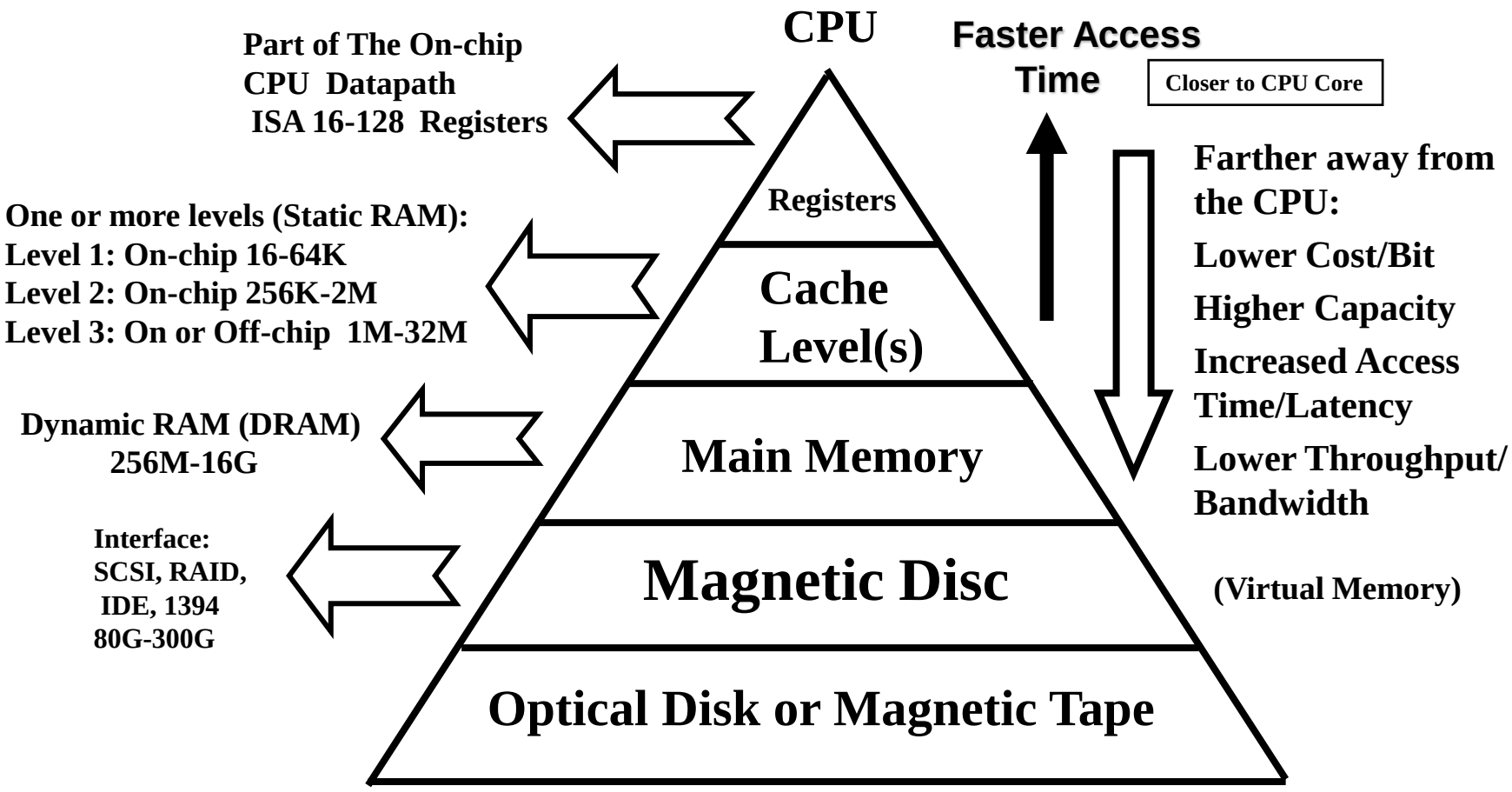
- Removing the ideal memory assumption in CPU designs leads to a large increase in clock cycle time and/or CPI greatly reducing CPU performance.

Ideal Memory Access Time ≤ 1 CPU Cycle
Real Memory Access Time >> 1 CPU cycle

Removing The Ideal Memory Assumption

- For example if we use real (non-ideal) memory with 80 ns access time (instead of 2ns) in our CPU designs then:
- Single Cycle CPU:
  - Loads will require 80ns + 1ns + 2ns + 80ns + 1ns = 164ns = C
  - The CPU clock cycle time C increases from 8ns to 164ns (125MHz to 6 MHz)
  - CPU is 20.5 times slower
- Multi Cycle CPU:
  - To maintain a CPU cycle of 2ns (500MHz) instruction fetch and data memory now take 80/2 = 40 cycles each resulting in the following CPIs
    - Arithmetic Instructions CPI = 40 + 3 = 43 cycles
    - Jump/Branch Instructions CPI = 40 + 2 = 42 cycles
    - Store Instructions CPI = 80 + 2 = 82 cycles
    - Load Instructions CPI = 80 + 3 = 83 cycles
    - Depending on instruction mix, CPU is 11-20 times slower++
- Pipelined CPU:
  - To maintain a CPU cycle of 2ns, a pipeline with 83 stages is needed.
  - Data/Structural hazards over instruction/data memory access may lead to 40 or 80 stall cycles per instruction.
  - Depending on instruction mix CPI increases from 1 to 41-81 and the CPU is 41-81 times slower!

Levels of The Memory Hierarchy



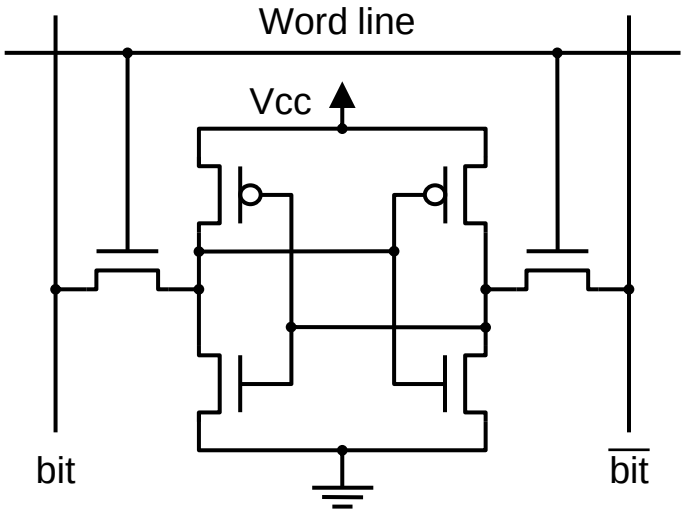
SRAM and DRAM

Two semiconductor memory technologies since the 1970's

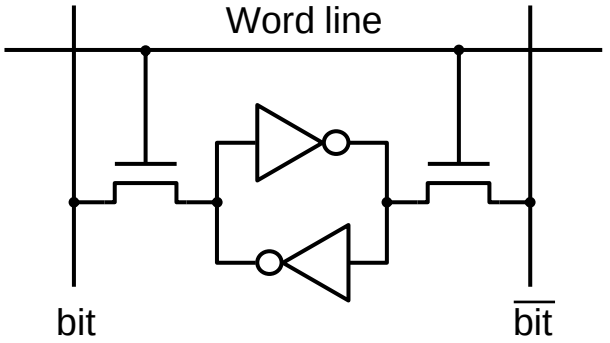
- SRAM = Static RAM
  - Invented by John Schmidt at Fairchild Semiconductor in 1964
  - 6-Transistor Cell in CMOS technology (low power to retain bit)
  - Faster than DRAM and easier to use ☺
  - SRAM is used for: cache memory, register files, tables, and buffers
- DRAM = Dynamic RAM
  - Invented by Robert Dennard in 1968
  - One Transistor + Capacitor per bit
  - More dense and cheaper than SRAM ☺
  - Must be refreshed periodically ☹
  - Typical choice for main memory

Static RAM Storage Cell

- 6-Transistor (6T) cell in CMOS
- Cell Implementation:
  - Cross-coupled inverters to store bit (4T)
  - Two access transistors (2T)
  - Does not require refreshing to maintain bit
  - Word line enables access to the cell
  - Two bit lines: bit and its complement $\overline{bit}$
  - Smaller and denser than a flip-flop
- Provides fast access time
- SRAM operation: standby, read, write

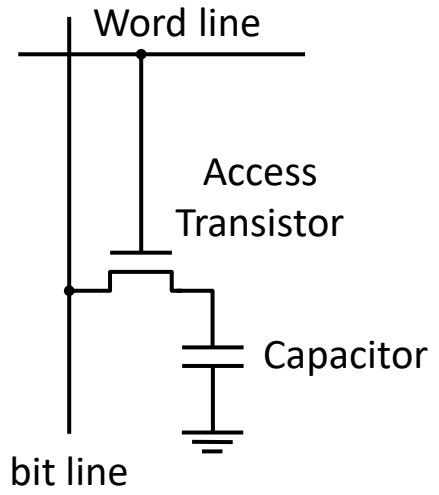


Typical SRAM cell



Dynamic RAM

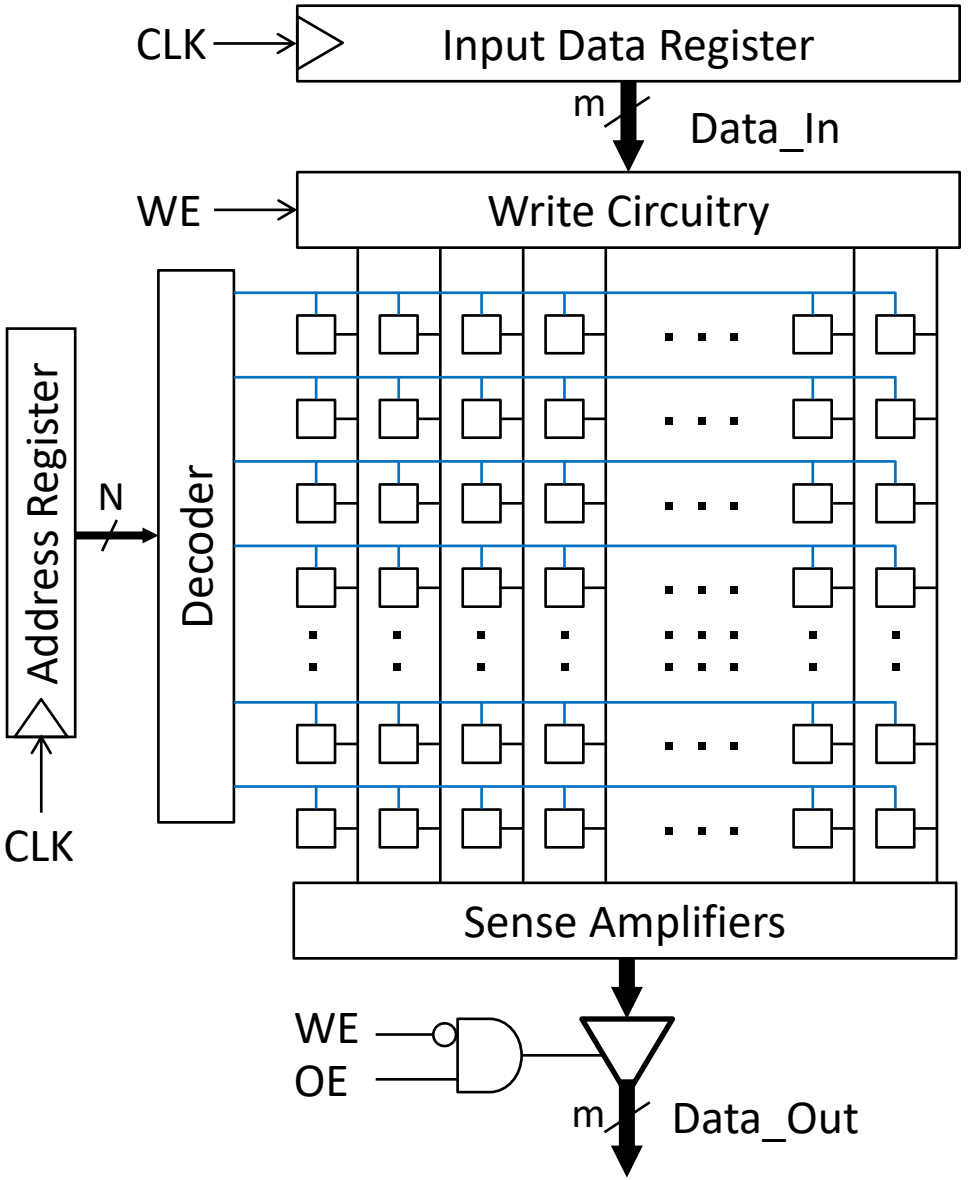
- Slower, Cheaper, and Denser memory than SRAM
- Typical choice for **main memory**
- Cell Implementation:
  - 1-Transistor (1T) Cell (access transistor)
  - Trench capacitor (stores bit)
  - Cell area is 10X to 20X smaller than SRAM
- Bit is stored as a **charge** on capacitor
- Must be **refreshed periodically**
  - Because of leakage of charge from tiny capacitor
- Refreshing for a memory row
  - Reading each row and writing it back to restore the charge



DRAM cell

Memory Array Architecture

- 2D Memory Array
  - Good regularity, high density
- n-bit address ➔ 2^n rows
- Each row has m cells
- Capacity = $2^n \times m$ bits
- The row size (m) can be large
  - Cache: m can be 512 bits
  - Register File: m can be 64 bits
- Control signals:
  - WE = Write Enable
  - OE = Output Enable
  - CLK = Clock Signal



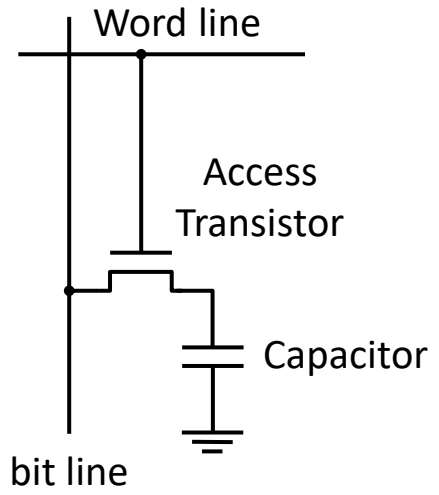
Presentation Outline

- Motivation

- **Random Access Memory and its Structure**

Dynamic RAM

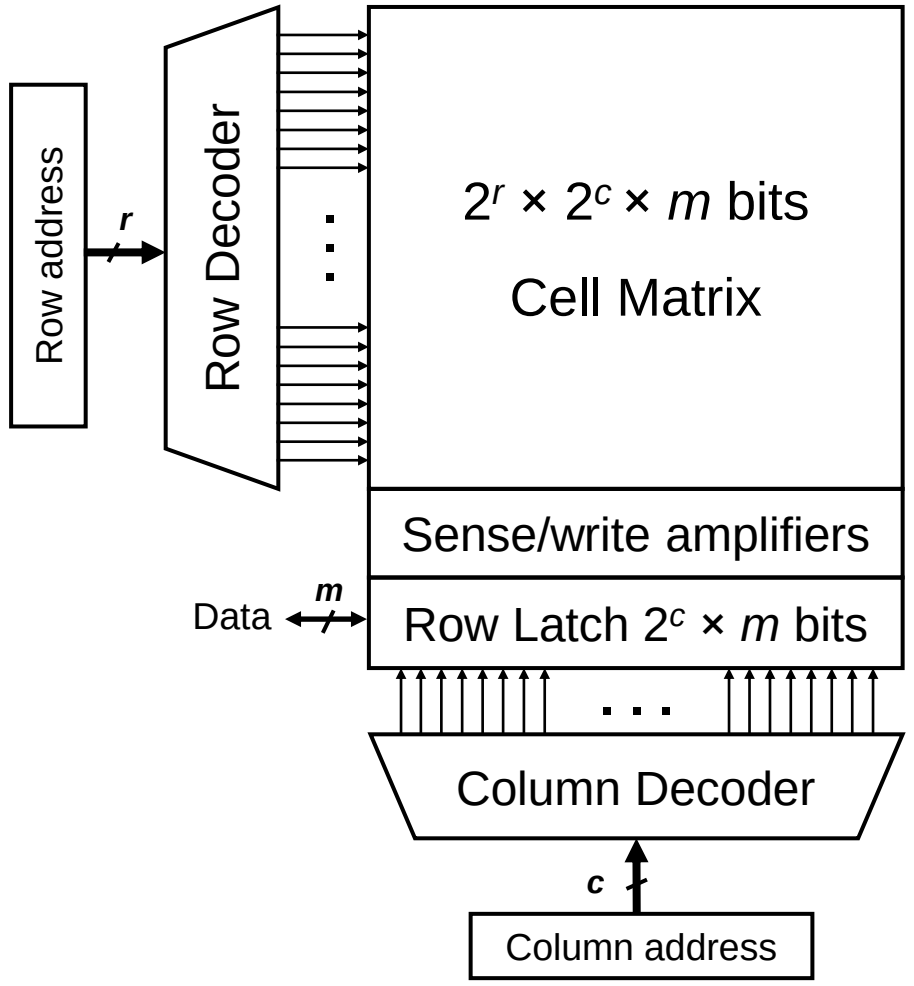
- Slower, Cheaper, and Denser memory than SRAM
- Typical choice for **main memory**
- Cell Implementation:
  - 1-Transistor (1T) Cell (access transistor)
  - Trench capacitor (stores bit)
  - Cell area is 10X to 20X smaller than SRAM
- Bit is stored as a **charge** on capacitor
- Must be **refreshed periodically**
  - Because of leakage of charge from tiny capacitor
- Refreshing for a memory row
  - Reading each row and writing it back to restore the charge



DRAM cell

DRAM Memory Structure

- 2D Memory Array of DRAM cells
  - *R* bits for the row address ➔ 2^R rows
  - *C* bits for the column address ➔ 2^C cols
  - Data bus width = *m* bits
- Capacity = $2^R \times 2^C \times m$ bits
- Row decoder: select row to open
  - Sense Amplifiers read data on bit lines
  - A complete row is read and latched
- Column decoder
  - Select column to read/write (within row)
  - Bidirectional (in/out) data bus = *m* bits
- Write Drivers
  - Write row back into the memory array



DRAM Operation

- Row Access (RAS)
  - Latch and decode row address to enable addressed row
  - Small change in voltage detected by sense amplifiers
  - Latch whole row of bits
  - Sense amplifiers drive bit lines to recharge storage cells
- Column Access (CAS) read and write operation
  - Latch and decode column address to select *m* bits
  - *m* = 4, 8, 16, or 32 bits depending on the DRAM package
  - On read, send latched bits out to chip pins
  - On write, charge storage cells to required value
  - Can perform multiple column accesses to same row (burst mode)

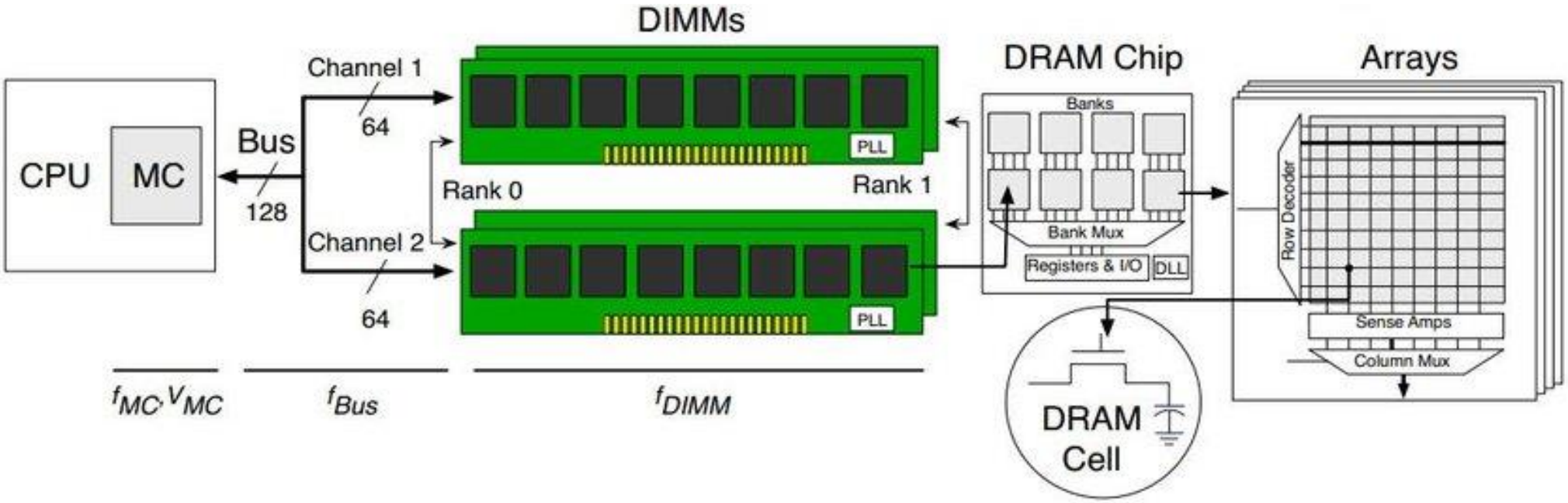
Burst Mode Operation

- Used for Block Transfer
  - Row address is latched and decoded
  - A read operation causes ALL cells in a selected row to be read
  - Selected row is latched internally inside the DRAM chip
  - Column address is latched and decoded
  - Selected column data is placed in the data output register
  - Column address is incremented automatically
  - Multiple data items are read depending on the block length
- Fast transfer of blocks between main memory and cache
- Fast transfer of pages between main memory and disk

SDRAM and DDR SDRAM

- Old DRAMs were asynchronous (no clock input)
- SDRAM is **Synchronous Dynamic RAM**
  - Added clock input to DRAM chip interface
- SDRAM is synchronous with the memory bus clock
  - As memory bus clock speed improved, SDRAM delivered higher performance than asynchronous DRAM
- DDR is **Double Data Rate** SDRAM
  - Like SDRAM, DDR is synchronous with the bus clock, but DDR transfers data on both the rising and falling edges of the clock

Organization of a modern memory subsystem



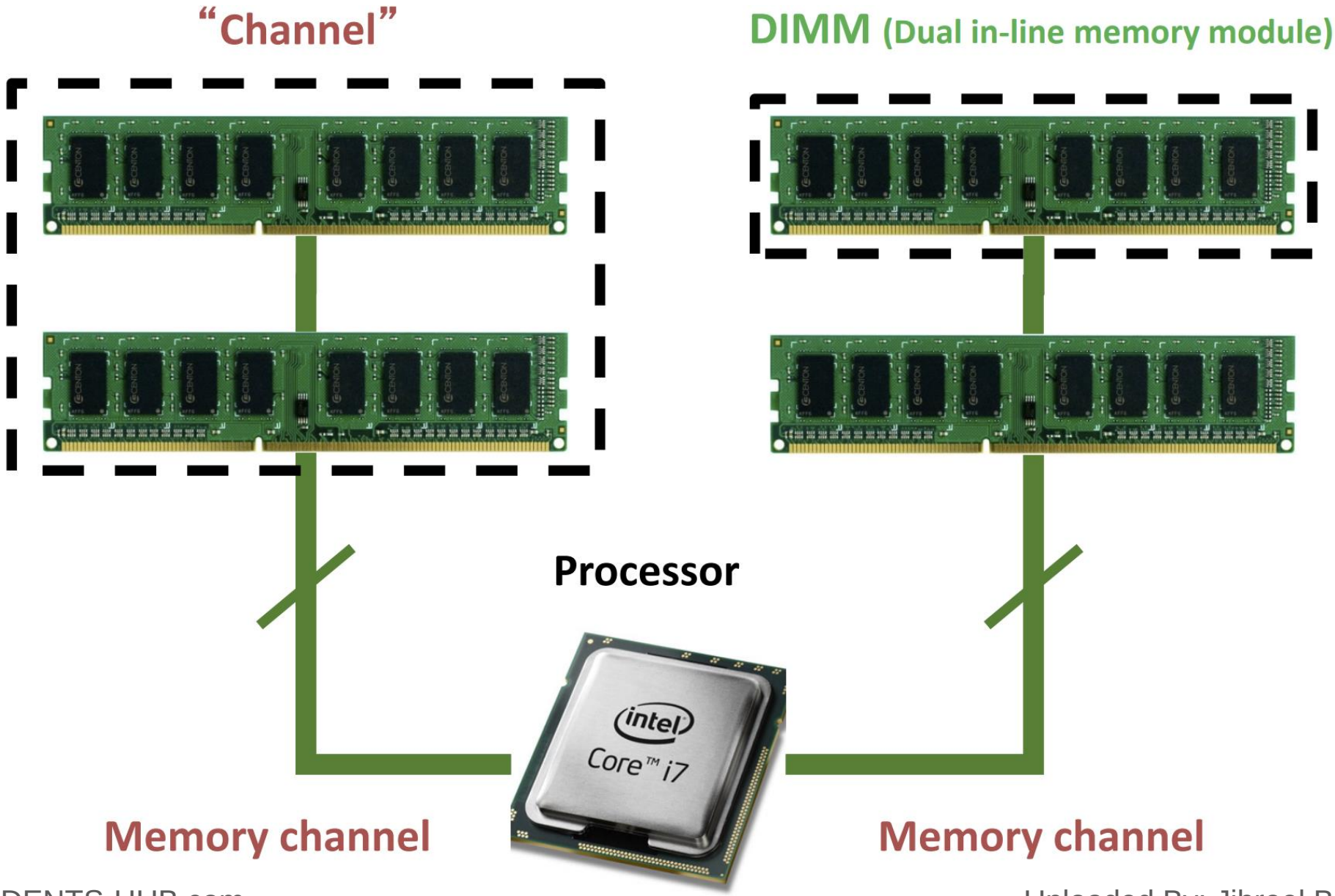
Memory Channels

- **Multi-channel memory architecture** is a technology that increases the data transfer rate between the DRAM memory and the memory controller by adding more channels of communication between them.
- The aim is to use the memory modules in pairs and combine the bandwidth, therefore, maximizing the system's capacity.
  - You should also use identical memory sticks in pairs (in terms of frequency, capacity, and from the same brand if possible).
- Theoretically, this multiplies the data rate by exactly the number of channels present.
- Modern high-end desktop and workstation processors such as the AMD Ryzen Threadripper series and the Intel Core i9 Extreme Edition lineup support quad-channel memory

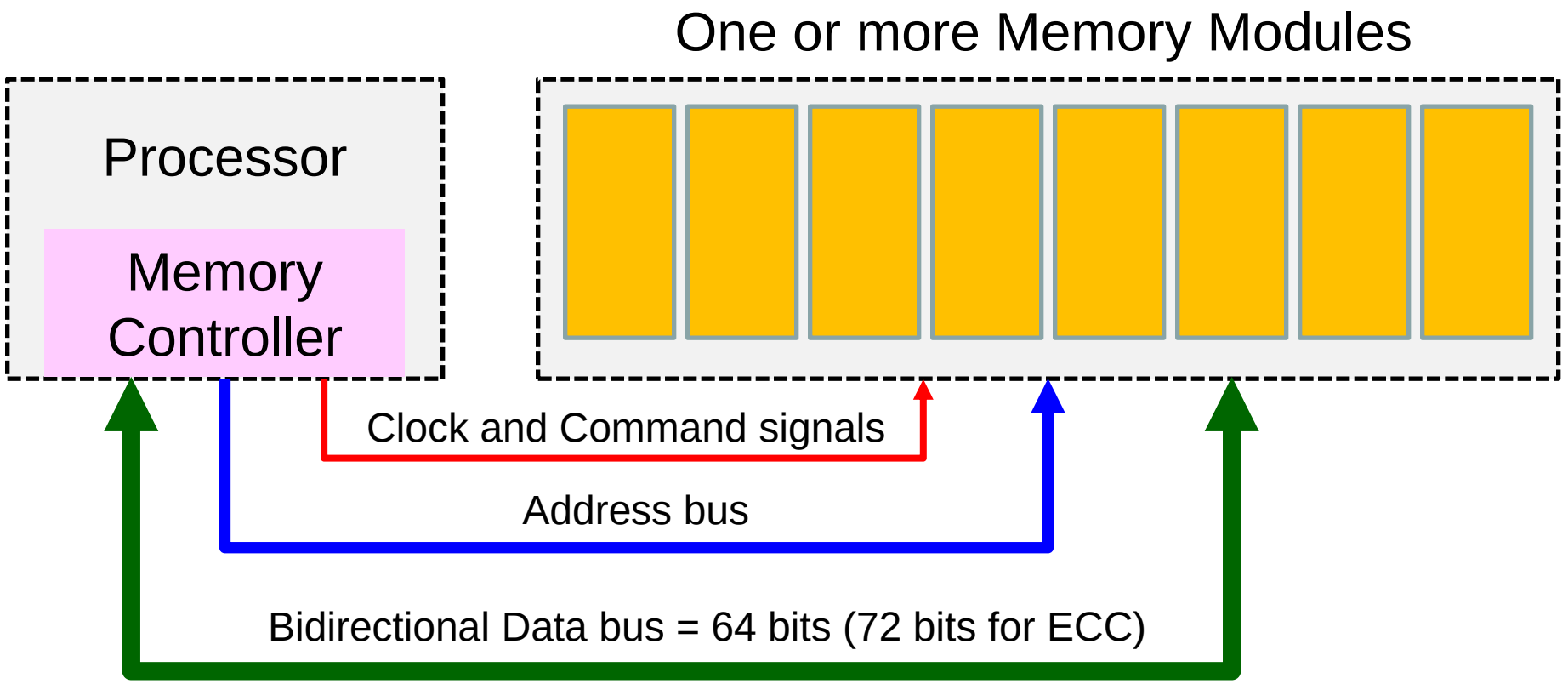
Memory Channels

- At a high level, each processor chip consists of one of more off-chip memory *channels*.
- Each memory channel consists of its own set of *command*, *address*, and *data* buses.
- Depending on the design of the processor, there can be either an independent memory controller for each memory channel or a single memory controller for all memory channels.
- All modules connected to a channel share the buses of the channel. Each module consists of many DRAM devices (or chips).

Dual-channel Memory



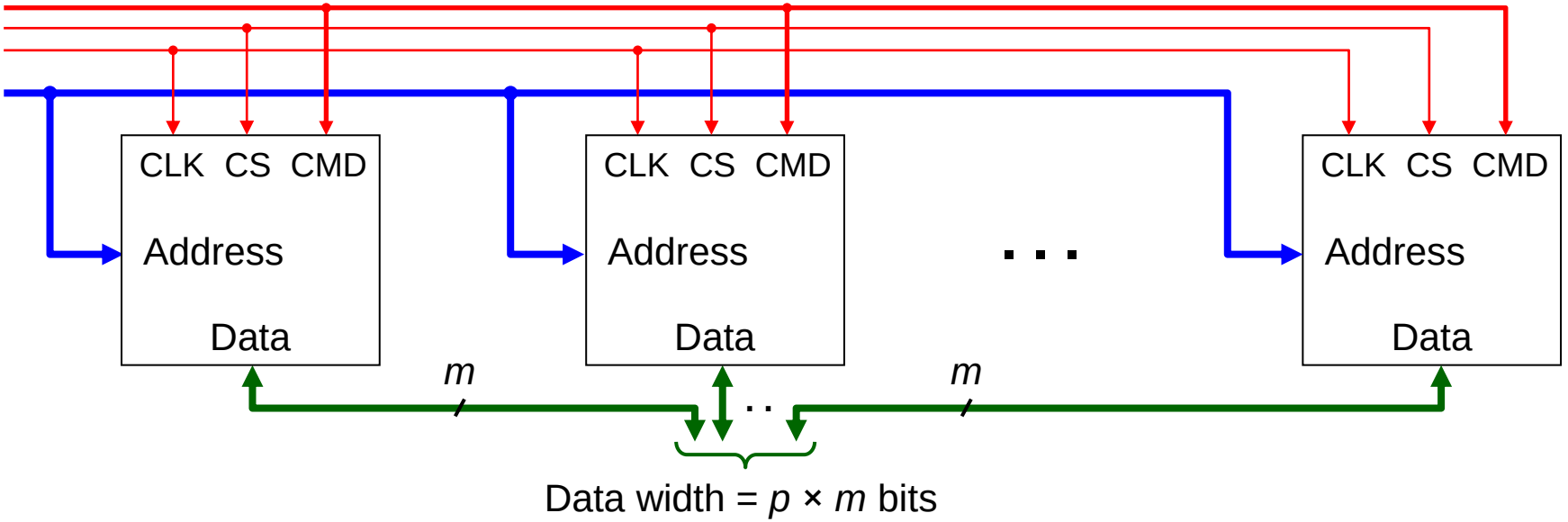
Main Memory Subsystem



- Memory controller is integrated on the processor chip
- Sends commands and memory addresses to the memory modules
- Sends and receives data on the bidirectional data bus

Memory Module

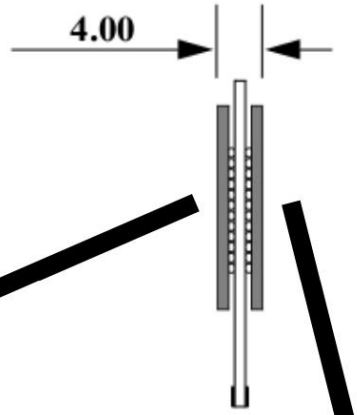
- Group of DRAM chips accessed in parallel on a small PCB
  - Same clock and command signals
  - Same address, but separate data lines
  - Increases memory capacity and bandwidth
  - Example: **Four x16** DRAM chips (p = 4, m = 16 bits)



DIMM (Dual in-line memory module)

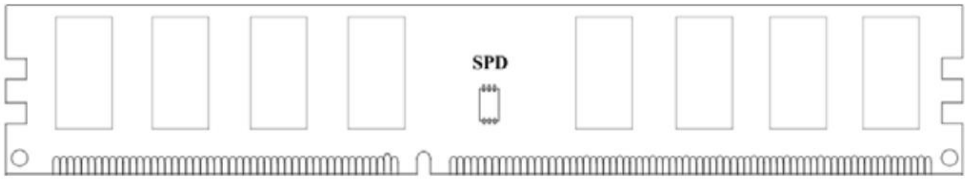
DIMM (Dual in-line memory module)

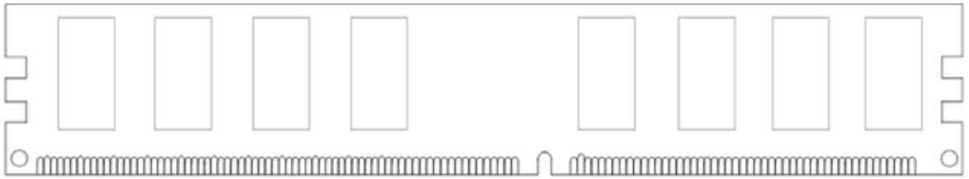
Side view



Front of DIMM

Back of DIMM





Module Characteristics

- Different forms: DIMM versus SO-DIMM (Small Outline)
- Different module interfaces for DDR, DDR2, 3, and 4
  - NOT backward compatible: different pins, signaling, and timing
- Chip density and data width
  - Size of each DRAM chip and width of data in bits: x4, x8, or x16
- ECC (Error Correcting Code) versus non-ECC module
  - Non-ECC module uses 64-bit data bus, while ECC uses 72-bit data bus
- Number of DRAM chips on a Module:
  - Four x16 chips, Eight x8 chips, or Sixteen x4 chips ➔ 64-bit data bus
  - Nine x8 chips or Eighteen x4 chips ➔ 72-bit data bus (ECC module)
- Registered (RDIMM) versus Unregistered (UDIMM):
  - Registered RDIMM add a register between the DRAMs and controller

Memory Module Cost and Capacity

- Example:
  - Suppose we use 16-Gbit DRAMs each costing $10, to build DIMMs
  - We have 3 choices for the data width: x4, x8, or x16 (16-bit data per chip)

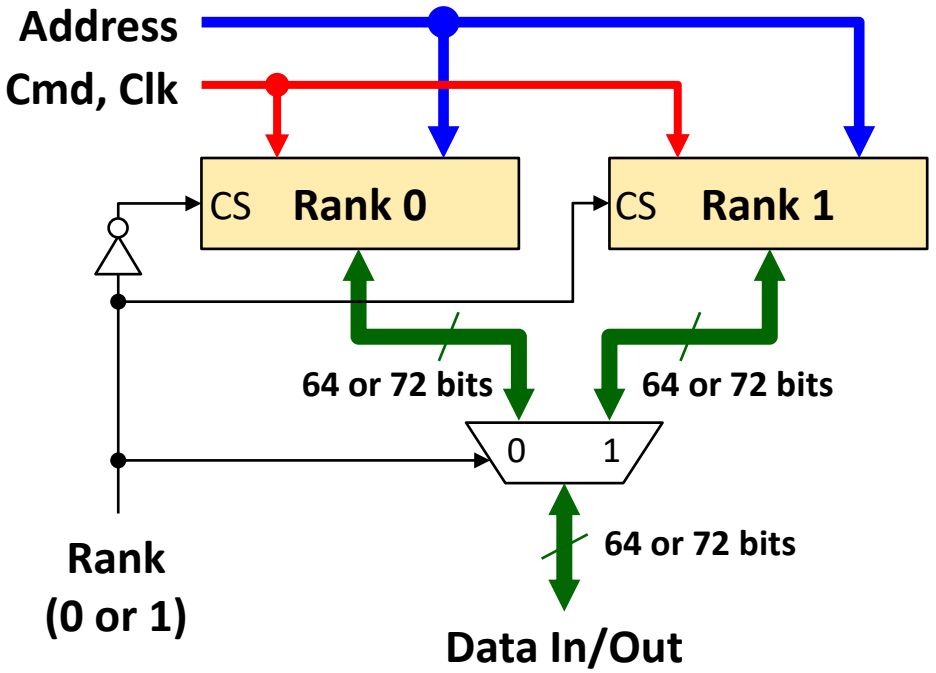
1. Which DRAM chip would you choose to build the lowest cost single-ranked non-ECC module, and what will be its capacity?
2. Which DRAM chip would you choose to built the highest capacity quad-ranked module with ECC, and what will be its cost?

- **Solution:**

1. Choose **x16** DRAMs. Only **Four** x16 DRAMs are needed for 64-bit data. Cost = 4×$10 = **$40** (ignoring cost of board). Capacity = **8 GBytes**
2. Choose **x4** DRAMs. **Eighteen** x4 DRAMs are needed for 72-bit data bus (single rank with ECC). For a quad-ranked module, we need 4 × 18 = 72 DRAM chips. Cost = 72×$10 = **$720**. Capacity = 64×16 Gb = **128 GBytes**

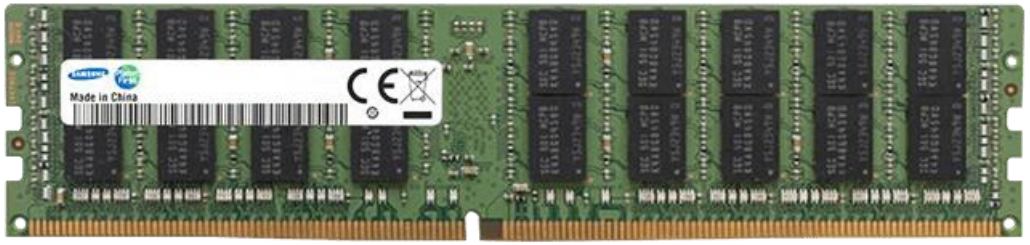
Multiple Memory Ranks

- A memory rank is a set of DRAM chips on a memory module
  - Connected to the **same chip select** and accessed simultaneously
  - Driven by the same command and address
- A memory module can have 1, 2, 4, or 8 ranks
  - Multiple ranks share the same 64-bit data bus (ECC adds 8 bits)



Given a Dual-Ranked Module

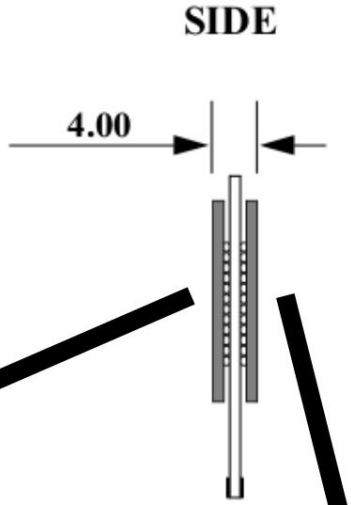
- If Rank input is 0, select Rank-0 chips
- If Rank input is 1, select Rank-1 chips



Breaking down a DIMM into Ranks

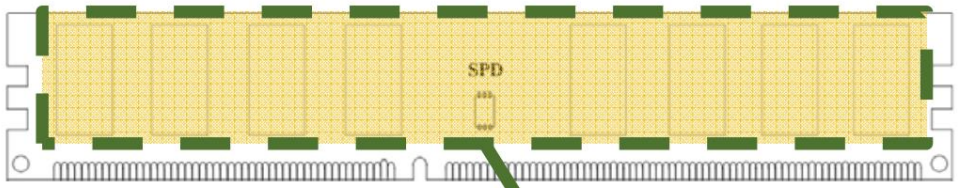
DIMM (Dual in-line memory module)

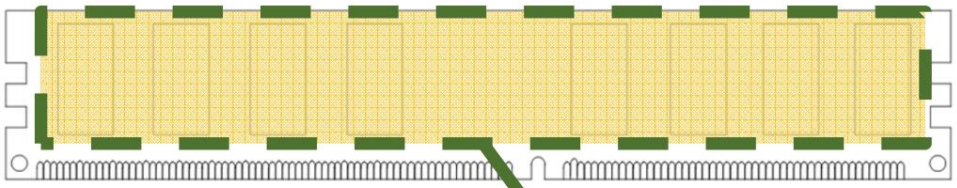
Side view



Front of DIMM

Back of DIMM

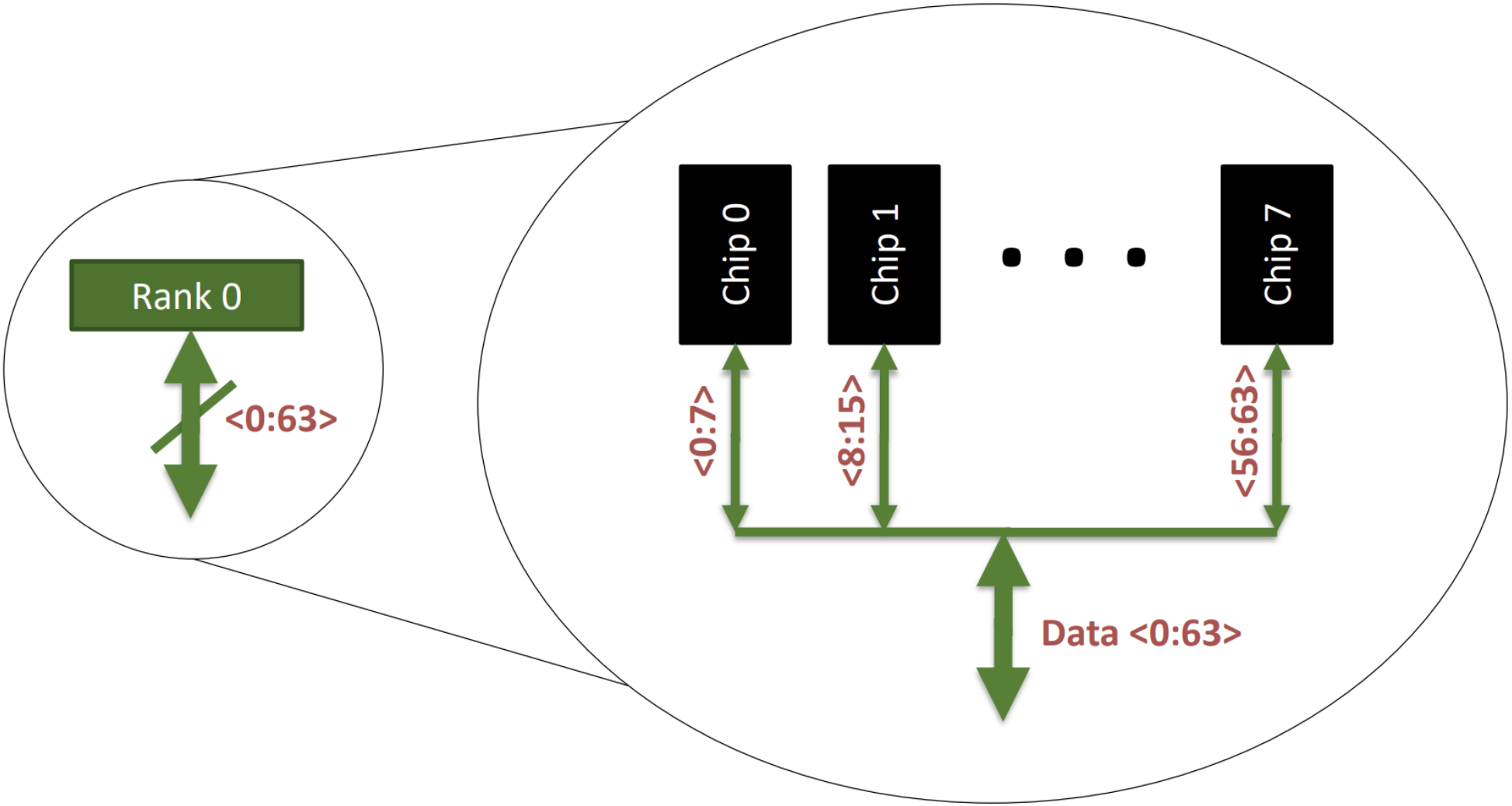




Rank 0: collection of 8 chips

Rank 1

Breaking down a Rank

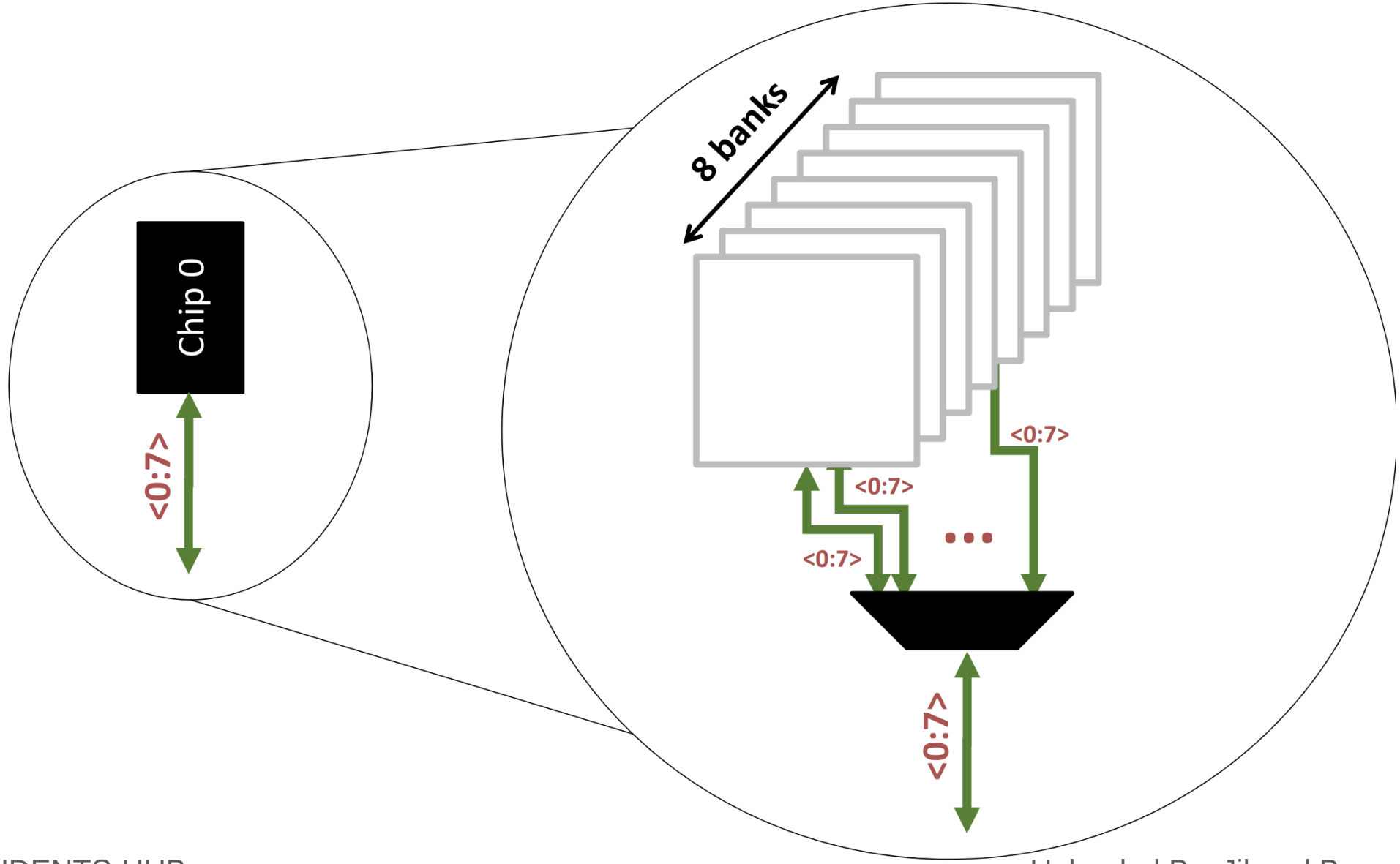


Memory Banks

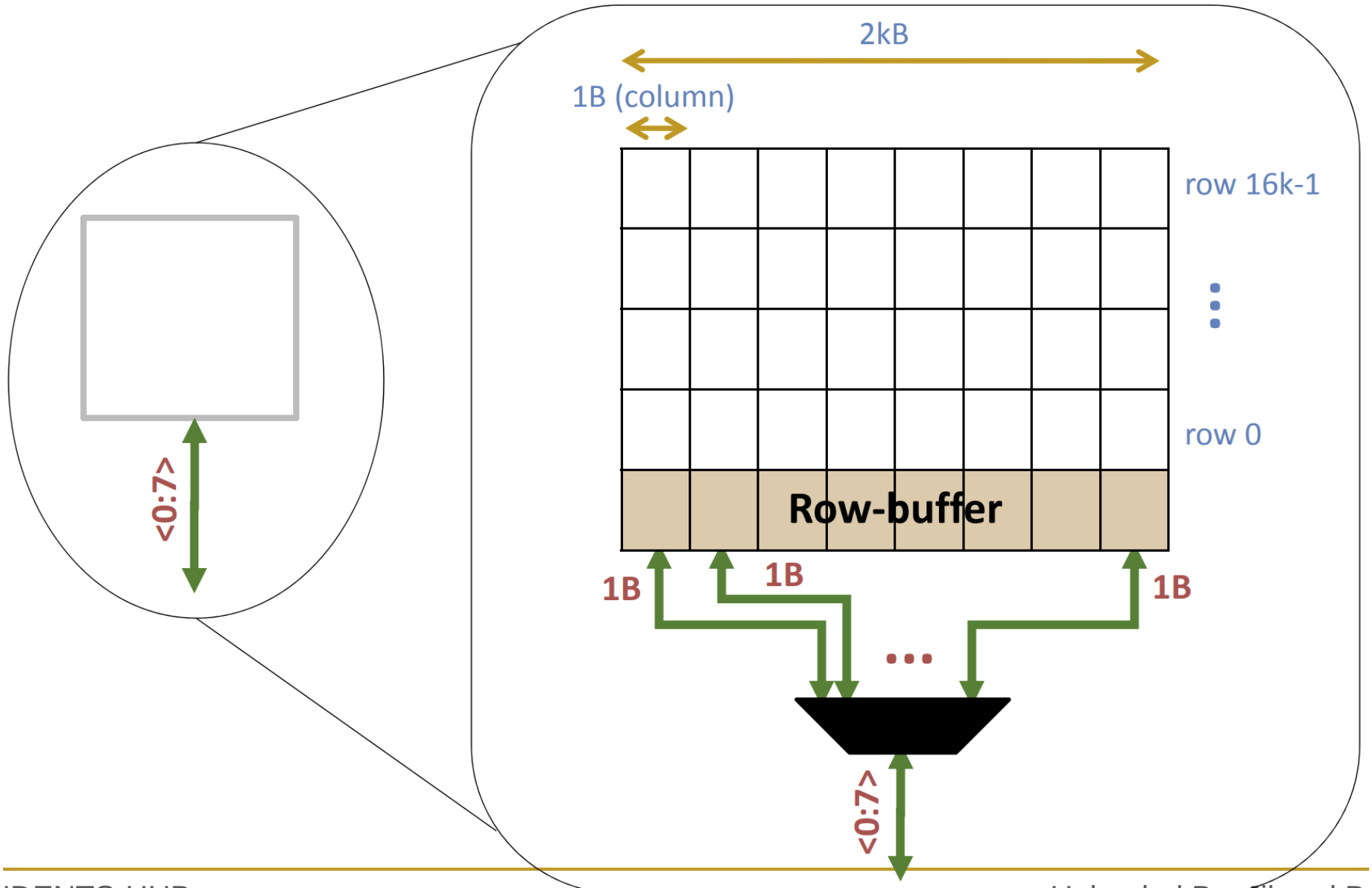
- When a memory array becomes large …
  - The wordlines and bitlines become long
  - Long lines have high capacitance, power consumption, and long delay
- Solution is to partition a large array into subarrays, called **banks**
- Memory banks can be accessed in parallel
- Memory address = 

  - Memory addresses are mapped to different banks at the **row level**
- Each bank must have the following:
  - A subarray of memory cells, row address, and decoder
  - Sense amplifiers and a row latch for storing an open row
  - A column decoder for read/write operations
  - Write drivers for writing the open row back and closing it

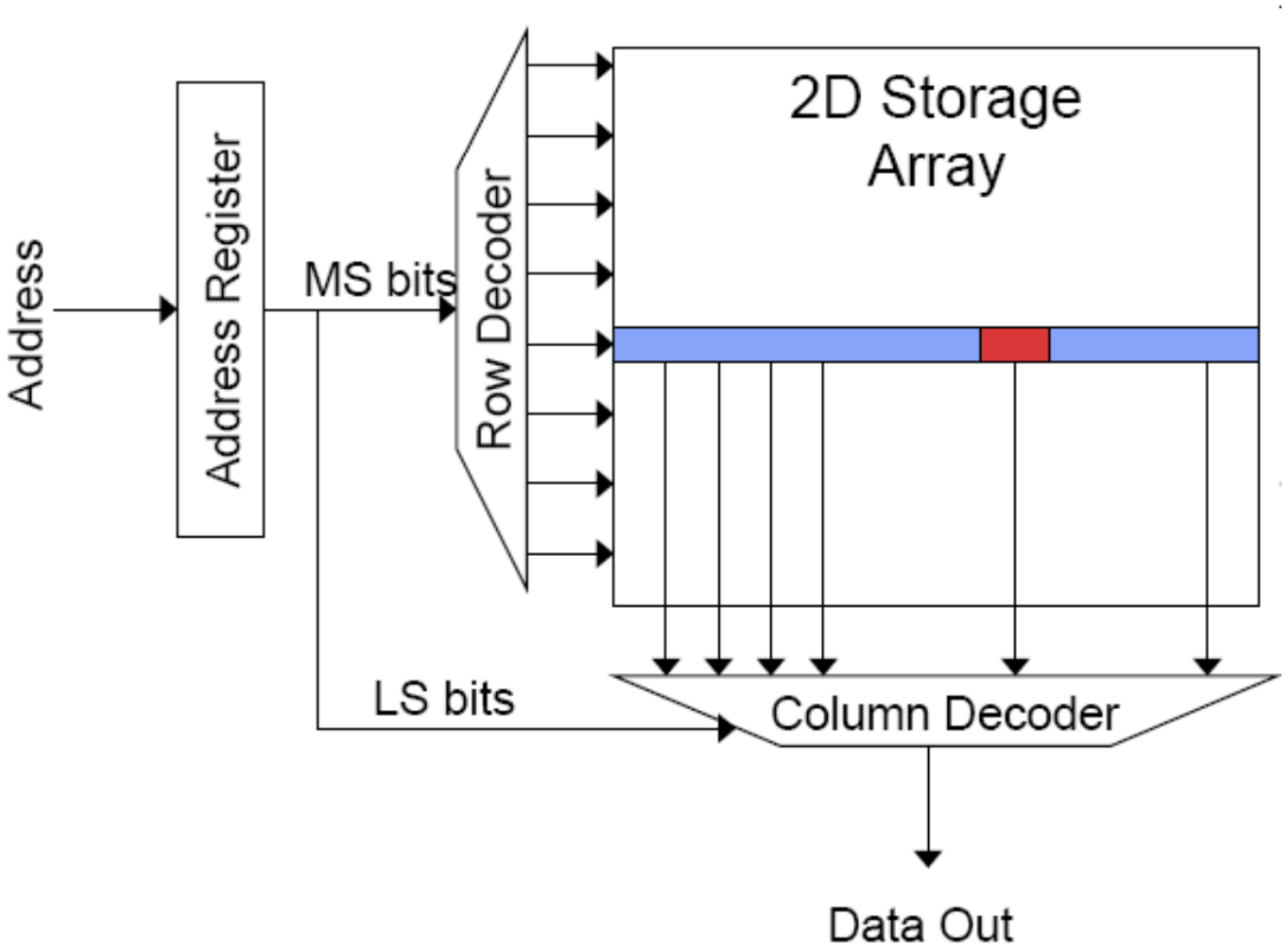
Breaking down a Chip



Breaking down a Bank



Memory Bank Organization and Operation



- Read access sequence:

1. Decode row address & drive word-lines

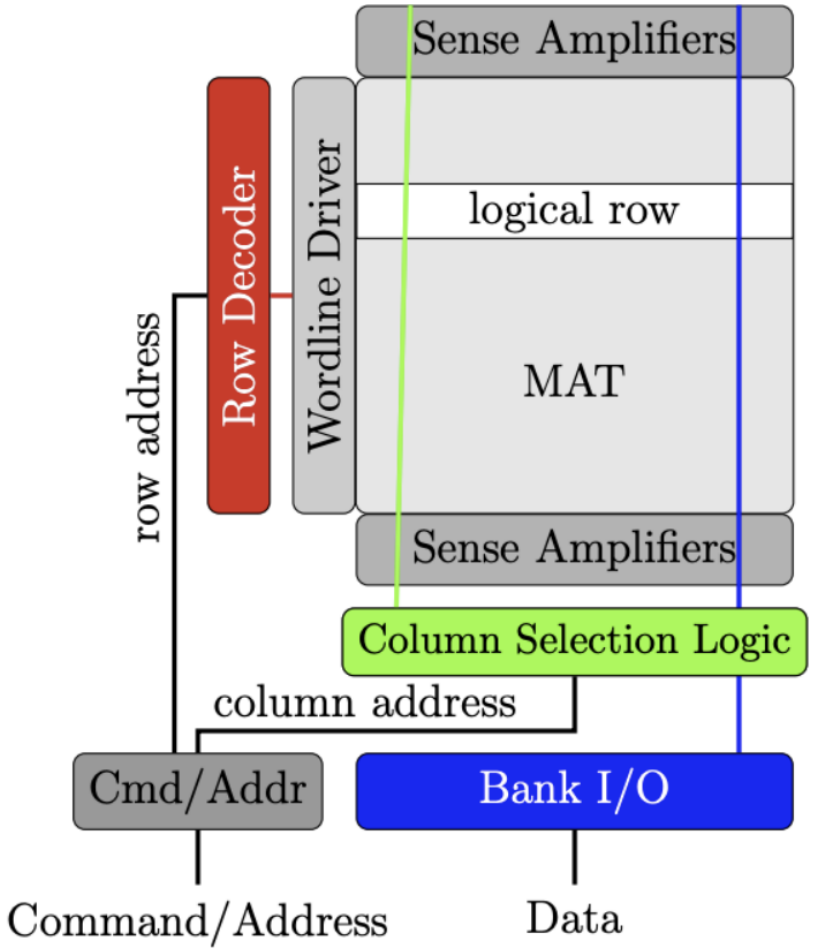
2. Selected bits drive bit-lines
   - Entire row read

3. Amplify row data

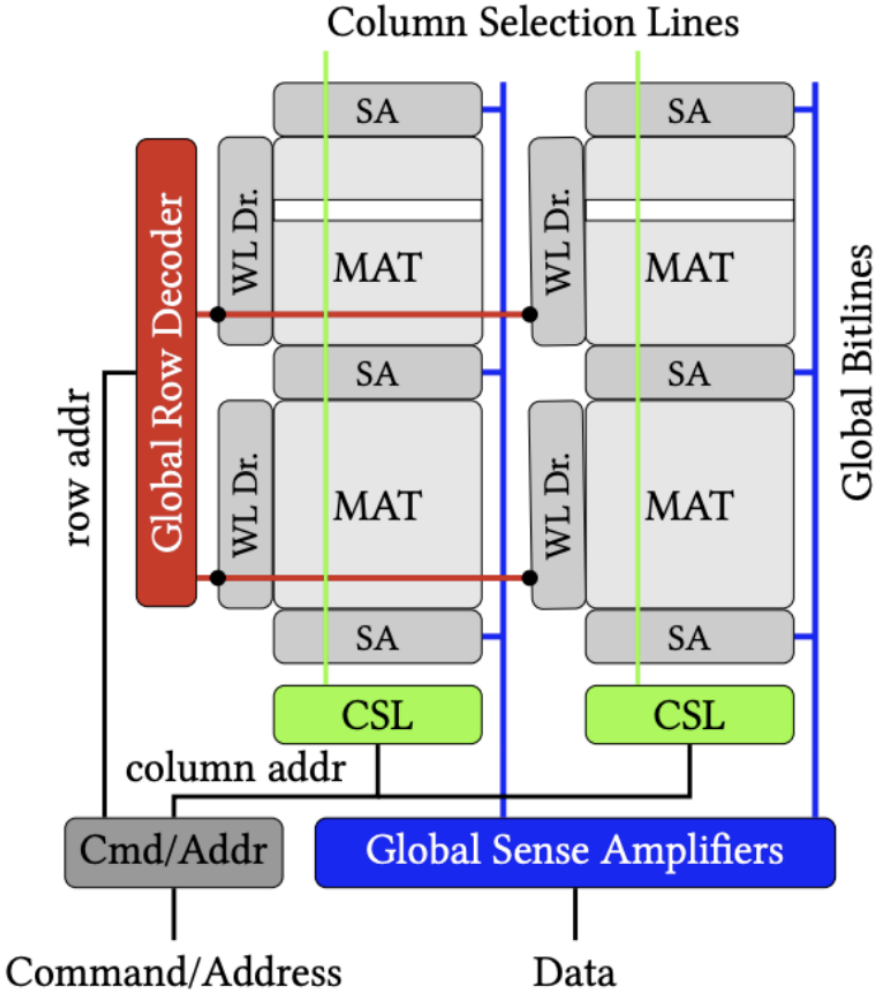
4. Decode column address & select subset of row
   - Send to output

5. Precharge bit-lines
   - For next access

Another View of a DRAM Bank

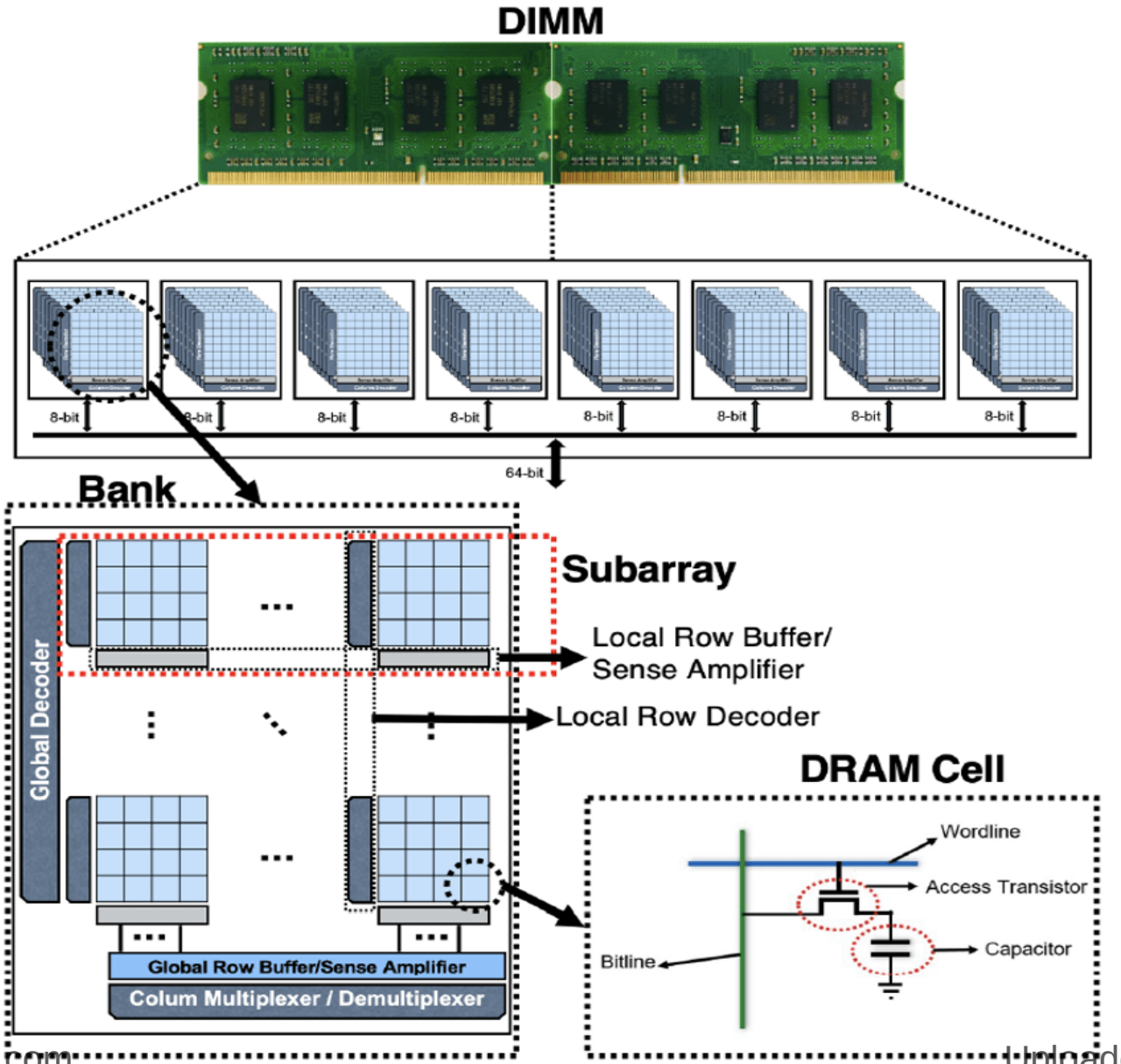


Logical Abstraction

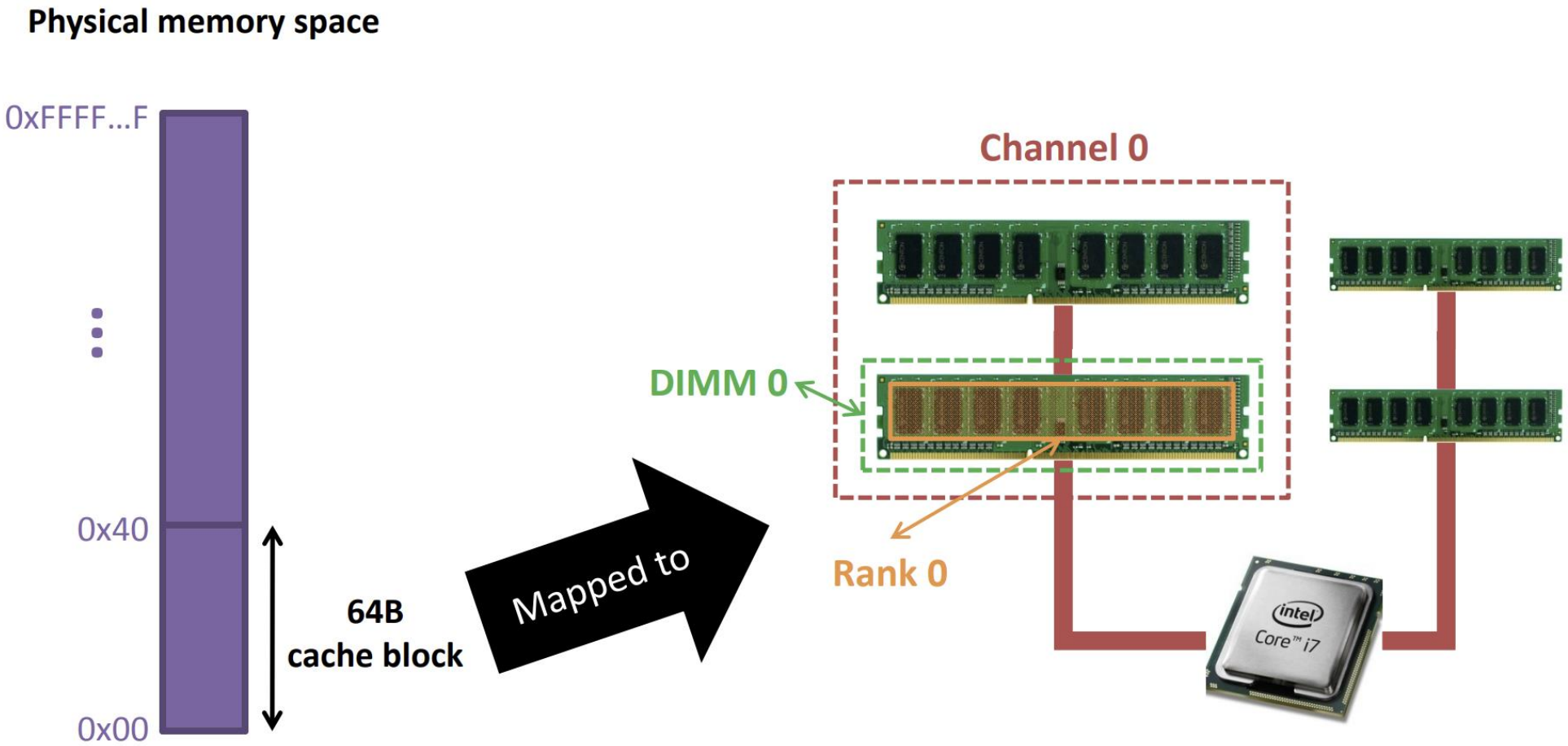


Physical View

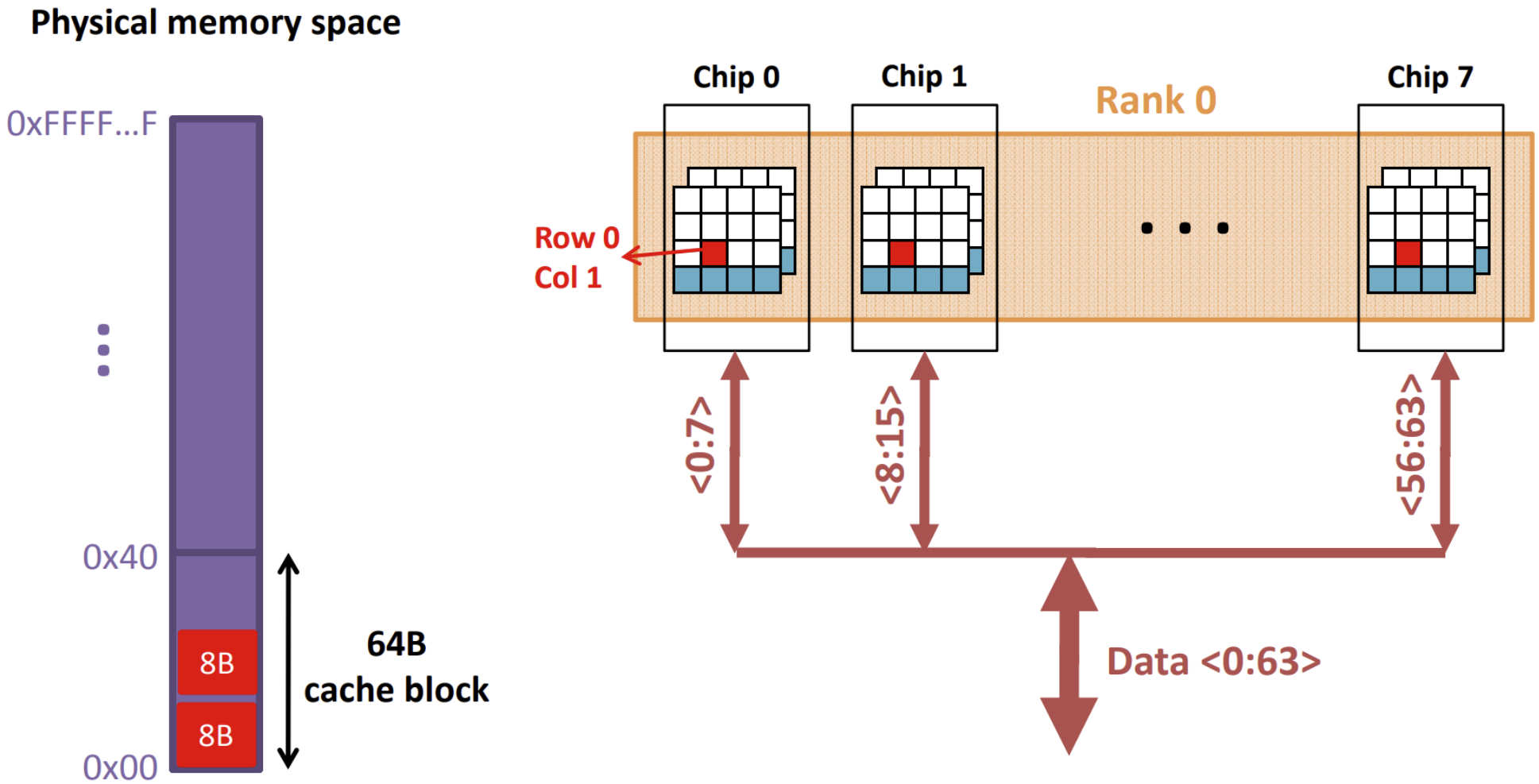
The Hole Picture



Example: Transferring a cache block



Example: Transferring a cache block



A 64B cache block takes 8 I/O cycles to transfer.

During the process, 8 columns are read sequentially.

SDRAM Chip Interface

CLK: Memory Bus Clock input

CKE: Clock Enable

BA0 – BA1: Bank Address

A0 – A12: Address lines

DQ0 – DQ15: Data in/out

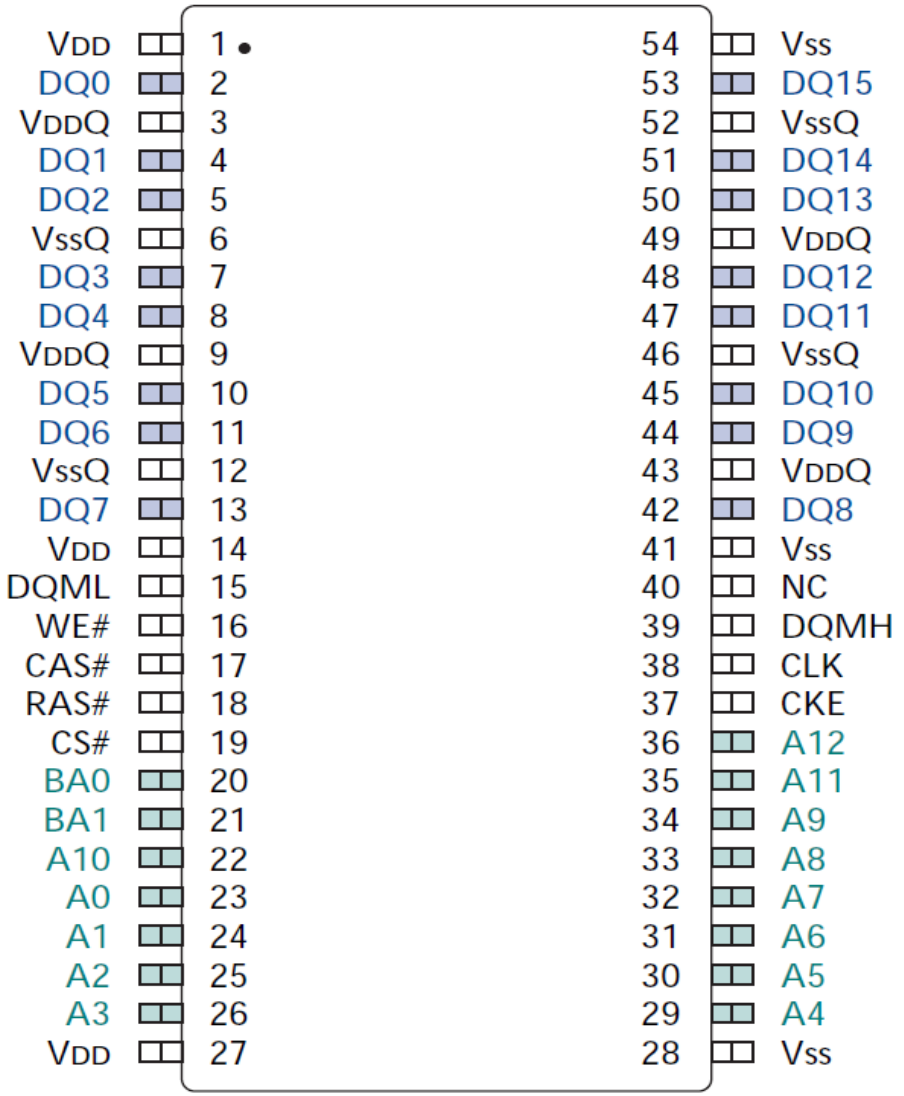
DQMH, DQML: Data Mask

CS: Chip Select

RAS, CAS, WE: Control Command

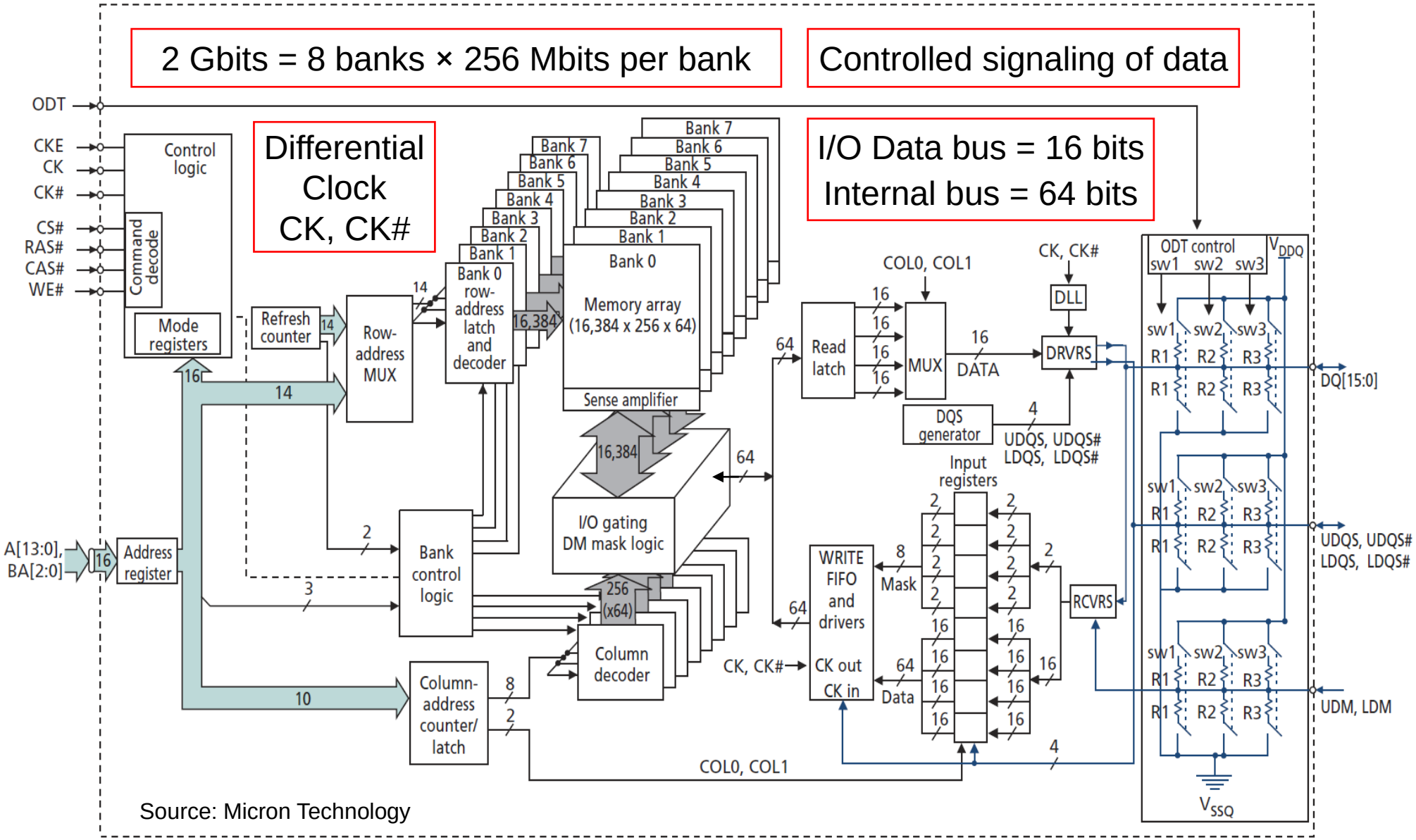
VDD, VDDQ: Power supply

VSS, VSSQ: Ground



Source: Micron Technology

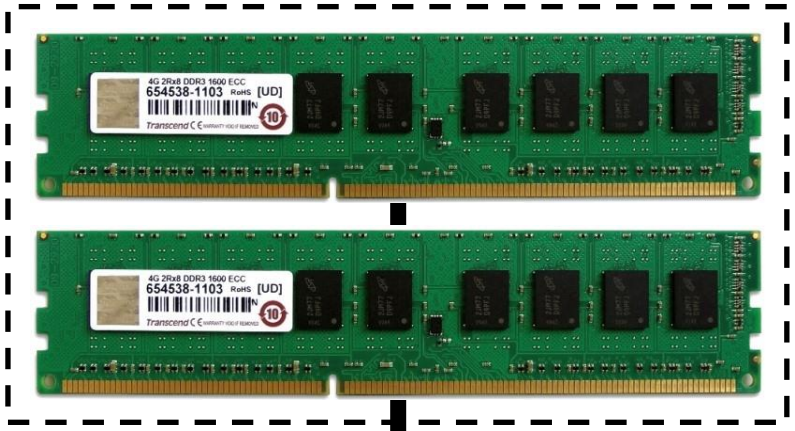
DDR2 SDRAM Chip (Eight Banks)



Source: Micron Technology

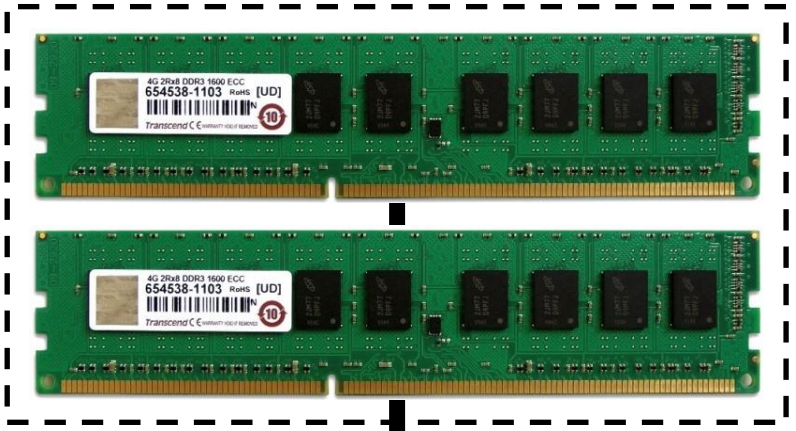
Memory Channels, Ranks, and Banks in i7

Multiple DIMMs per Channel
Multiple Ranks per DIMM
Multiple Banks per Rank
The Banks exist inside the chips



Memory Channel 0

Multiple DIMMs per Channel
Multiple Ranks per DIMM
Multiple Banks per Rank
The Banks exist inside the chips



Memory Channel 1

Address Mapping

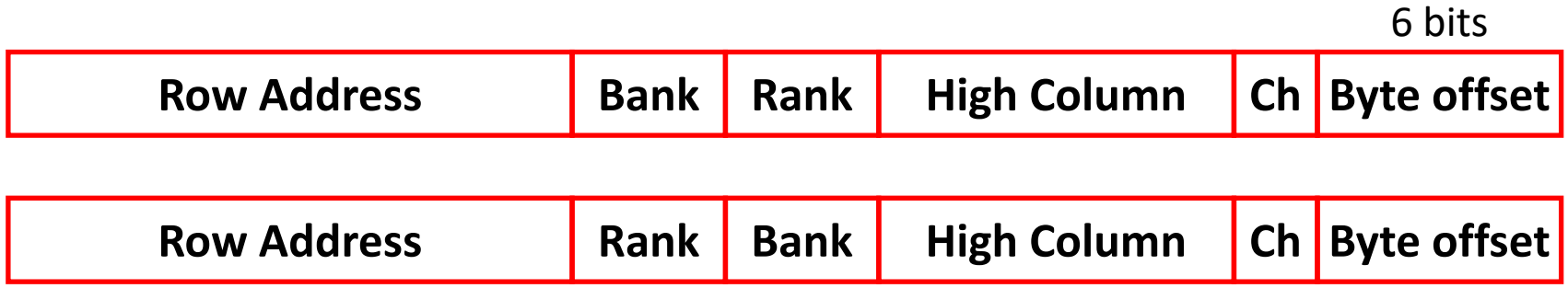
- Many ways of mapping addresses to channels, ranks, and banks
- Row (or Page) Interleaving ➔ favors **row locality**
  - Consecutive memory addresses are mapped to rows ➔ **row hit**
  - Then rows are mapped to channels, ranks, and banks
  - Different ways to map rows to channels, ranks, and banks
  - Byte offset is 3 bits because the data bus is 8-byte wide

3 bits

Row Address	**Bank**	**Rank**	**Ch**	**Column Address**	**Byte**
Row Address	**Rank**	**Bank**	**Ch**	**Column Address**	**Byte**
Row Address	**Rank**	**Ch**	**Bank**	**Column Address**	**Byte**
Row Address	**Bank**	**Ch**	**Rank**	**Column Address**	**Byte**
Row Address	**Ch**	**Rank**	**Bank**	**Column Address**	**Byte**
Row Address	**Ch**	**Bank**	**Rank**	**Column Address**	**Byte**

Address Mapping (cont'd)

- Cache Block Interleaving
  - Consecutive cache blocks are mapped to different channels
  - Cache block size = 64 bytes ➔ Byte offset = 6 bits
  - Burst Length = 8 transfers = 64 bytes
  - The lower 3 bits of the column address are used in the byte offset
  - High column = Column address excluding the lower 3 bits
  - Favors **Parallelism** ➔ Channel **Ch** appears next to **Byte offset**

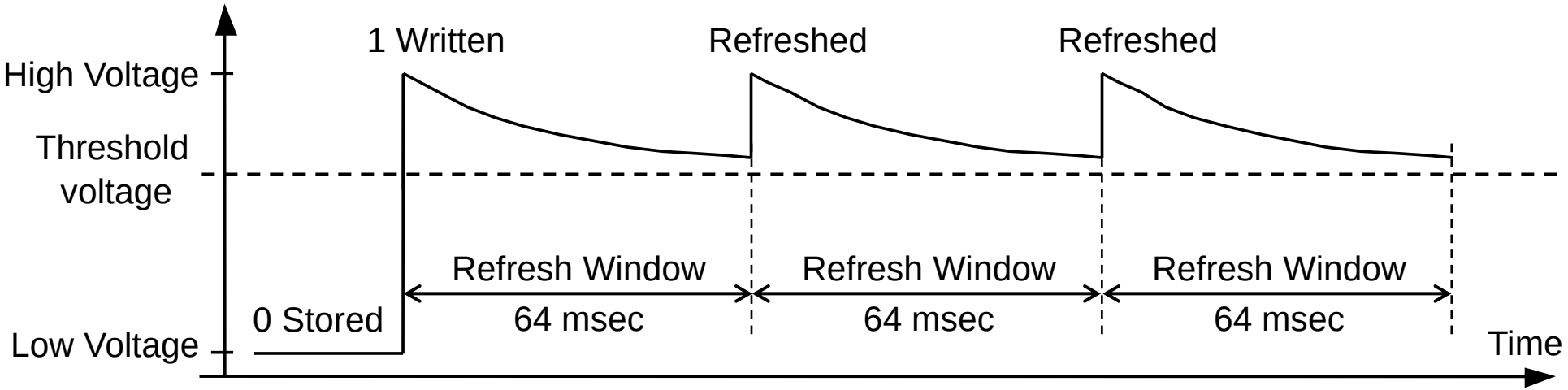


Memory Latency versus Bandwidth

- Memory Latency
  - Elapsed time between sending address and receiving data
  - Measured in nanoseconds
  - The total latency to a new row/column is the time between opening a new row of memory and accessing a column within it.
  - Reduced from 60 ns to 35 ns (between 1996 and 2016)
  - Improvement in memory latency is less than 2X (1996 to 2016)
- Memory Bandwidth
  - Rate at which data is transferred between memory and CPU
  - Bandwidth is measured as millions of Bytes per second
  - Increased from 800 to 25600 MBytes/sec (between 1996 and 2016)
  - Improvement in memory bandwidth is 32X (1996 to 2016)

Refresh Window

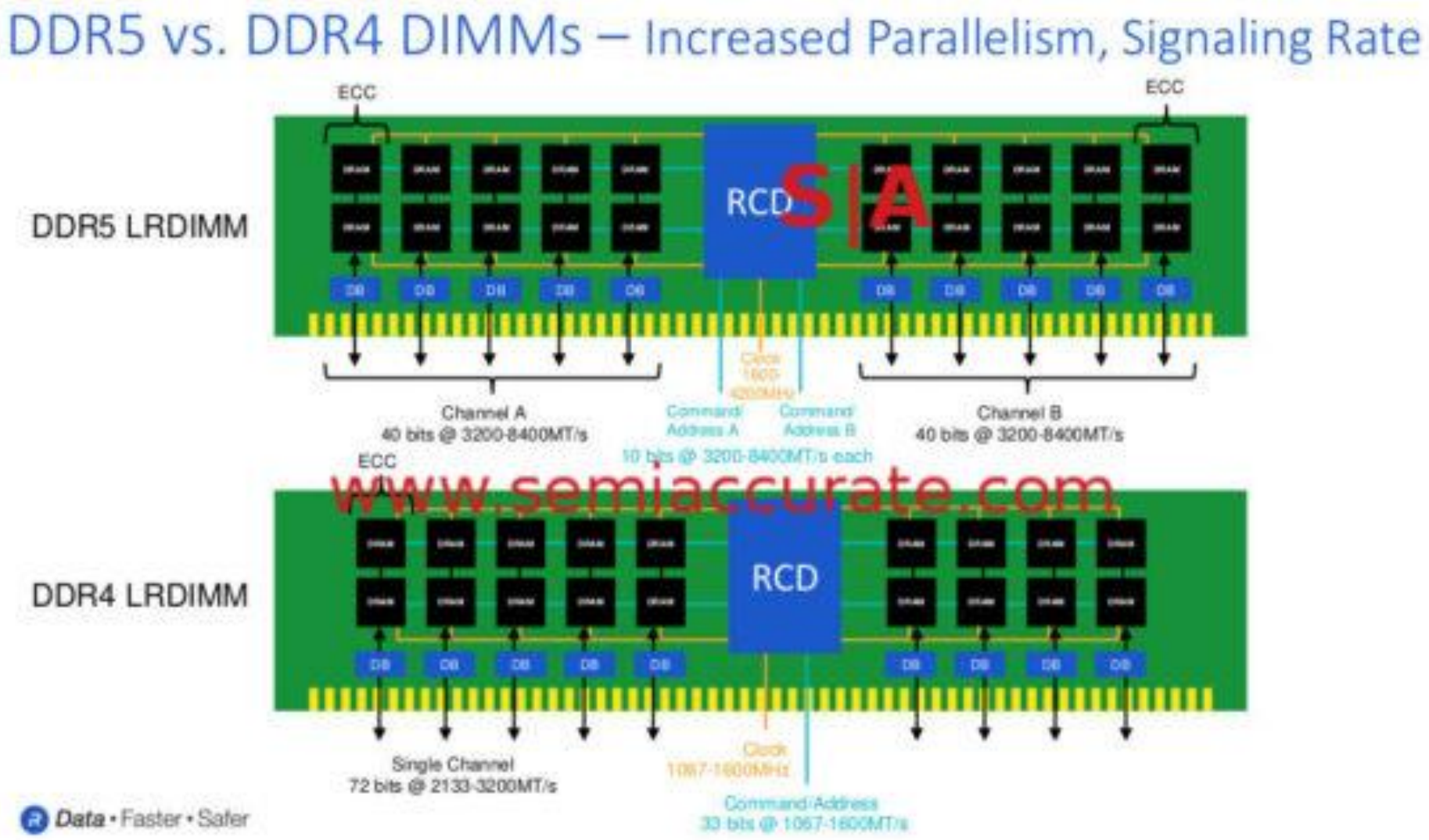
- The capacitors in a DRAM are not perfect and leak their charge
- A capacitor's voltage drops over time, but should not be allowed to drop below a threshold voltage to avoid the loss of bits
- **Refresh Window:** time interval in which **all rows** are refreshed
  - Refresh window is typically 64 msec, according to recent standards
  - **All rows** must be **read and rewritten** (refreshed) at least once every 64 ms



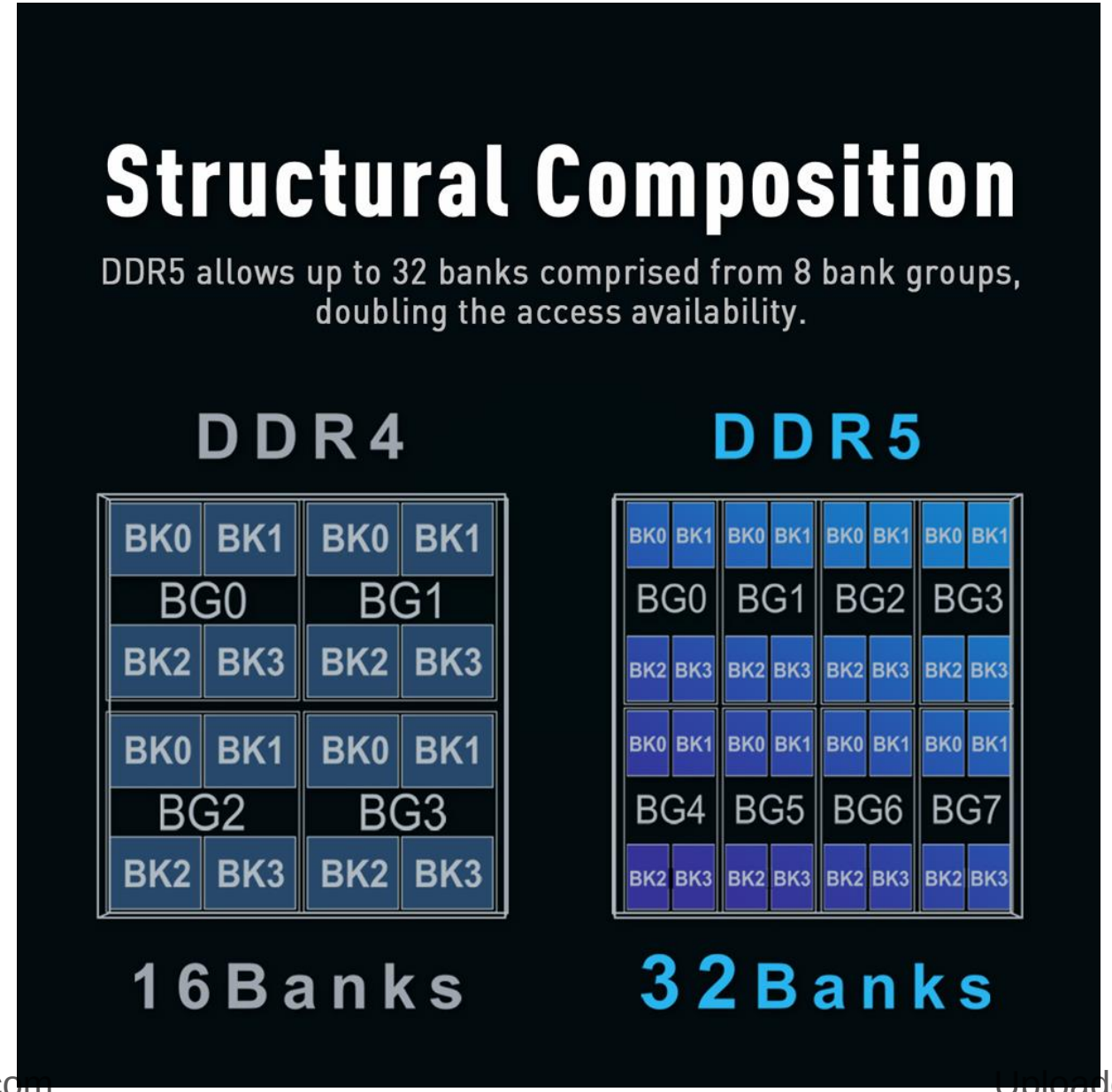
Loss of Bandwidth to Refresh Operations

- During a REFRESH operation, all banks must be **idle**
- Some memory bandwidth is lost to REFRESH operations
- Example:
  - 2 Gbit DRAM organized as: 8 banks, 8K rows × 2K columns × 16 bits
  - Refresh window = 64 msec (specified by standard)
  - Refresh operation takes 40 ns (tRFC = 40 ns)
  - What fraction of the memory bandwidth is lost to refresh operations?
  - If refresh commands are distributed, what is the average refresh interval?
- **Solution:**
  - Refreshing all 8K rows takes: 8 × 1024 × 40 ns = 327680 ns
  - Loss of 327680 ns every 64 ms
  - Fraction of lost memory bandwidth = 0.32768 / 64 = 0.512%
  - Average refresh interval = 64 msec / 8192 rows = 7.8125 μsec

DDR5 vs DDR4



DDR4 vs DDR5



Trends in DRAM

Year	Memory Standard	Chip Capacity (Mibit)	Bus Clock (MHz)	Data Rate (MT/s)	Peak Bandwidth (MB/s)	Total latency to a new row / column
1996	SDRAM	64-128	100-166	100-166	800-1333	60 ns
2000	DDR	256-512	100-200	200-400	1600-3200	55 ns
2004	DDR2	512-2048	200-400	400-800	3200-6400	50 ns
2010	DDR3	2048-8192	400-800	800-1600	6400-12800	40 ns
2014	DDR4	8192-32768	800-1600	1600-3200	12800-25600	35 ns

- Memory chip capacity: 1 Mibit = 2^{20} bits, 1 Gibit = 2^{30} bits
- Data Rate = Millions of Transfers per second (MT/s)
- Data Rate = 2 × Bus Clock for DDR, DDR2, DDR3, DDR4
- 1 Transfer = 8 bytes of data ➔ Bandwidth = MT/s × 8 bytes