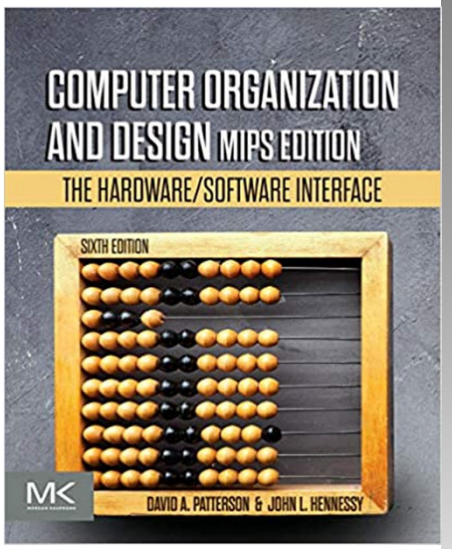
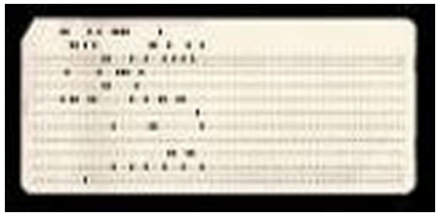


ESE 345 Computer Architecture

Memory Technology and Memory Systems

"memory technology" image created by SDXL text-to-image AI generative model 2023

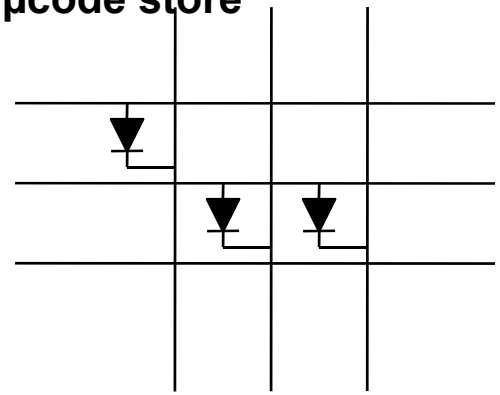
Early Read-Only Memory Technologies

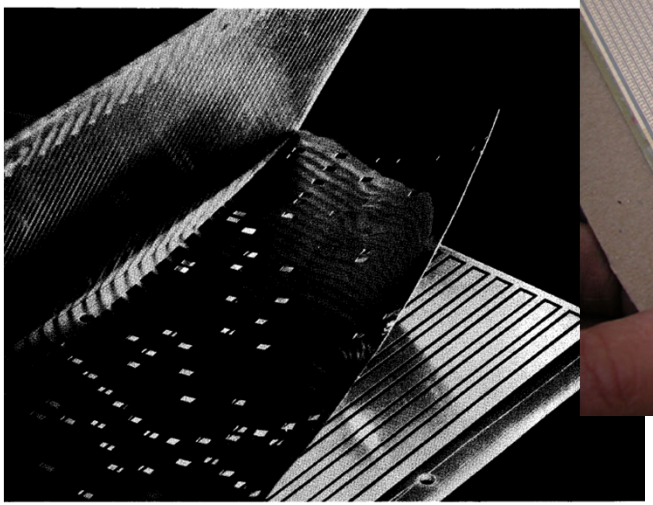


Punched cards, From early 1700s through Jaquard Loom, Babbage, and then IBM

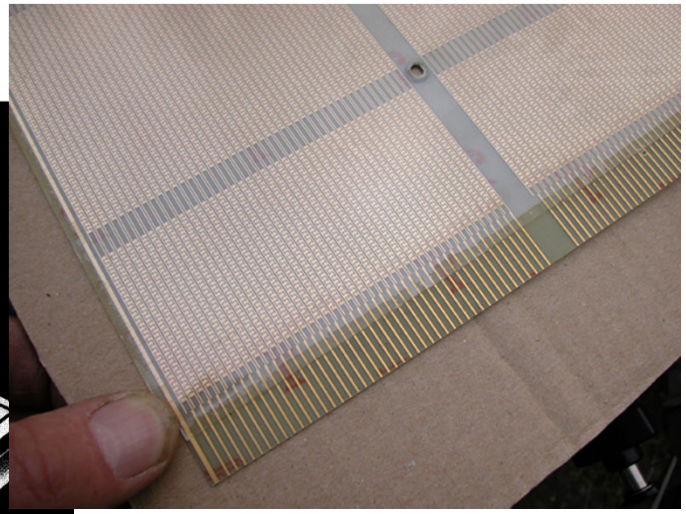
Punched paper tape, instruction stream in Harvard Mk 1

Diode Matrix, EDSAC-2 µcode store





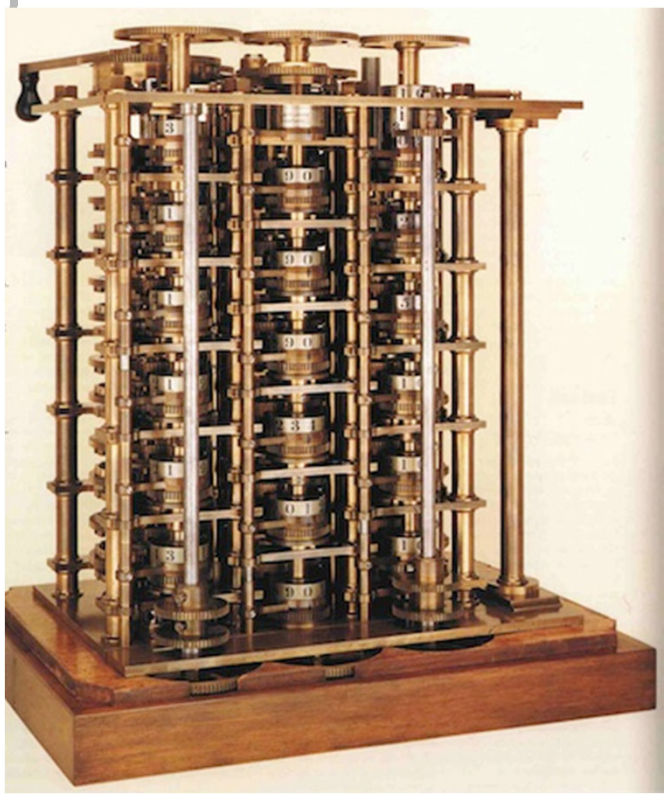
IBM Card Capacitor ROS

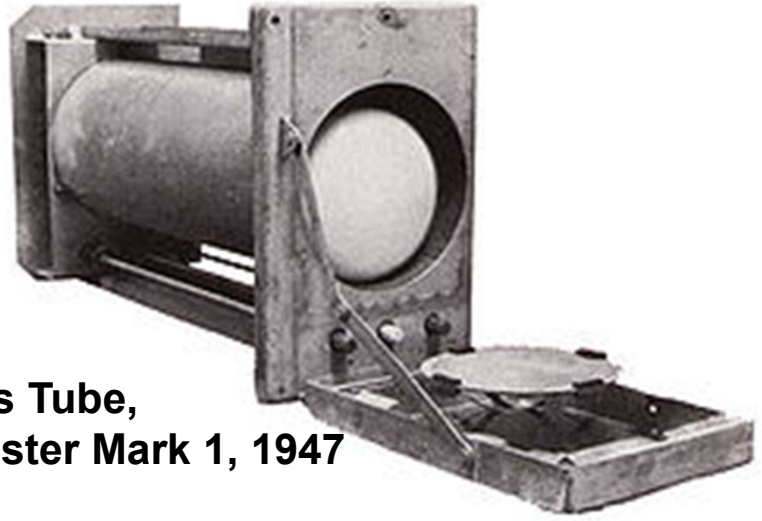


IBM Balanced Capacitor ROS

Early Read/Write Main Memory Technologies

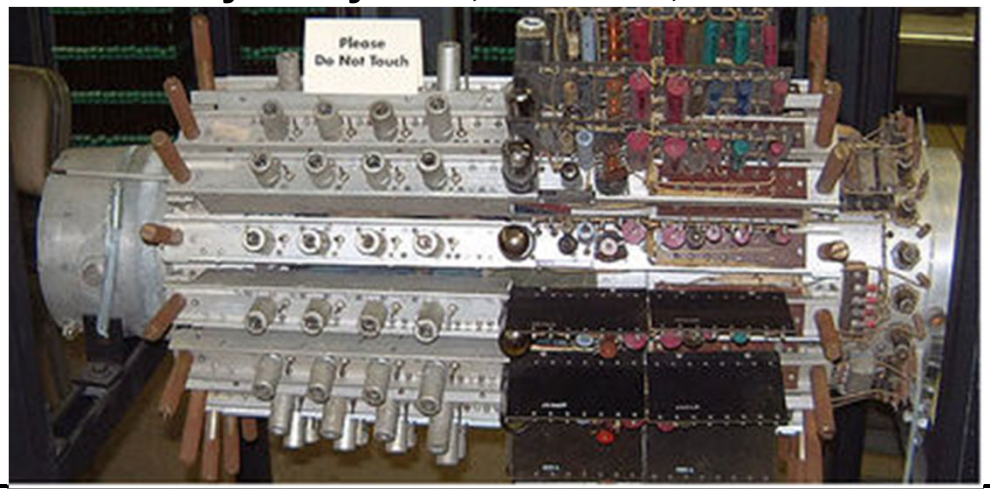
Babbage, 1800s: Digits stored on mechanical





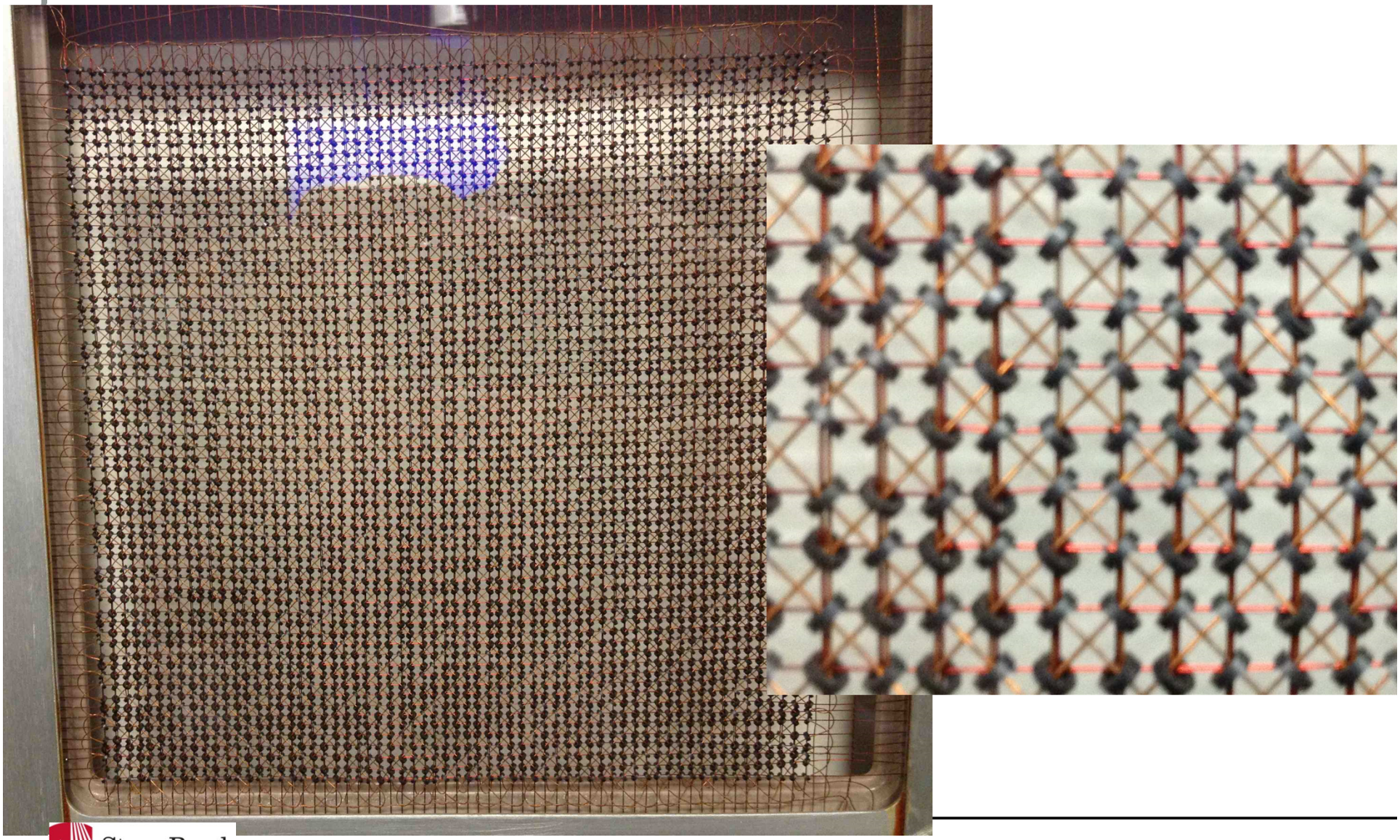
Williams Tube, Manchester Mark 1, 1947

Mercury Delay Line, Univac 1, 1951



Also, regenerative capacitor memory on Atanasoff-Berry computer, and rotating magnetic drum memory on IBM 650

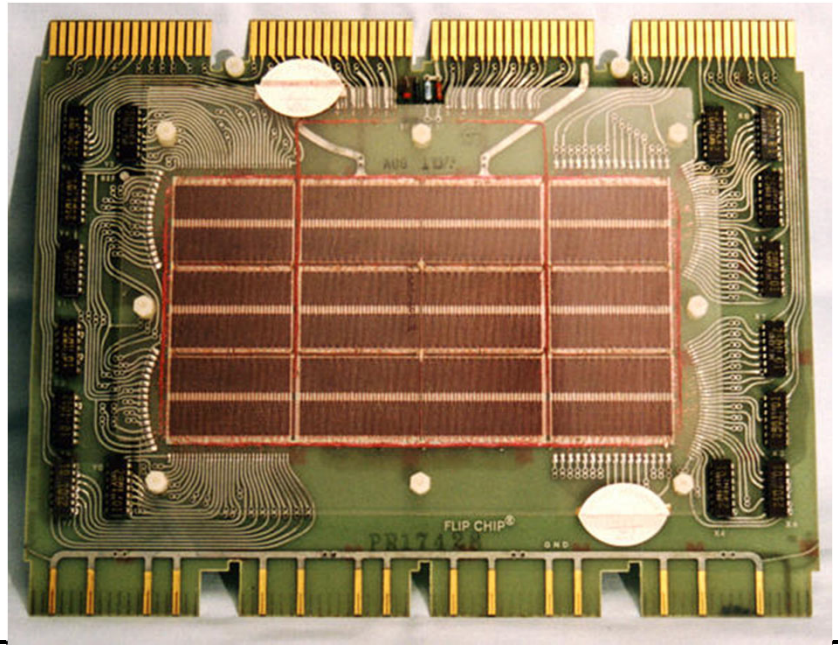
MIT Whirlwind Core Memory



Core Memory

- Core memory was first large scale reliable main memory
  - invented by Forrester in late 40s/early 50s at MIT for Whirlwind project
- Bits stored as magnetization polarity on small ferrite cores threaded onto two-dimensional grid of wires
- Coincident current pulses on X and Y wires would write cell and also sense original state (destructive reads)

- Robust, non-volatile storage
- Used on space shuttle computers
- Cores threaded onto wires by hand (25 billion a year at peak production)
- Core access time ~ 1ms



DEC PDP-8/E Board, 4K words x 12 bits, (1968)

Semiconductor Memory

- Semiconductor memory began to be competitive in early 1970s
  - Intel formed to exploit market for semiconductor memory
  - Early semiconductor memory was Static RAM (SRAM). SRAM cell internals similar to a latch (cross-coupled inverters).
- First commercial Dynamic RAM (DRAM) was Intel 1103
  - 1Kbit of storage on single chip
  - charge on a capacitor used to hold value
- Semiconductor memory quickly replaced core in '70s

Memory Hierarchy Technology

- Random Access:
  - "Random" is good: access time is the same for all locations
  - DRAM: Dynamic Random Access Memory
    - High density, low power, cheap, slow
    - Dynamic: need to be "refreshed" regularly
  - SRAM: Static Random Access Memory
    - Low density, high power, expensive, fast
    - Static: content will last "forever"(until lose power)
- "Non-so-random" Access Technology:
  - Access time varies from location to location and from time to time
  - Examples: Disk, CDROM, DRAM page-mode access
- Sequential Access Technology: access time linear in location (e.g.,Tape)
- Today's lecture will mostly concentrate on random access technology
  - The Main Memory: DRAMs + Caches: SRAMs

Memory Technology

- Static RAM (SRAM)
  - 0.5ns – 2.5ns, $500 – $1000 per GB
- Dynamic RAM (DRAM)
  - 50ns – 70ns, $3 – $6 per GB
- Magnetic disk
  - 5ms – 20ms, $0.01 – $0.02 per GB
- Ideal memory
  - Access time of SRAM
  - Capacity and cost/GB of disk

Main Memory Background

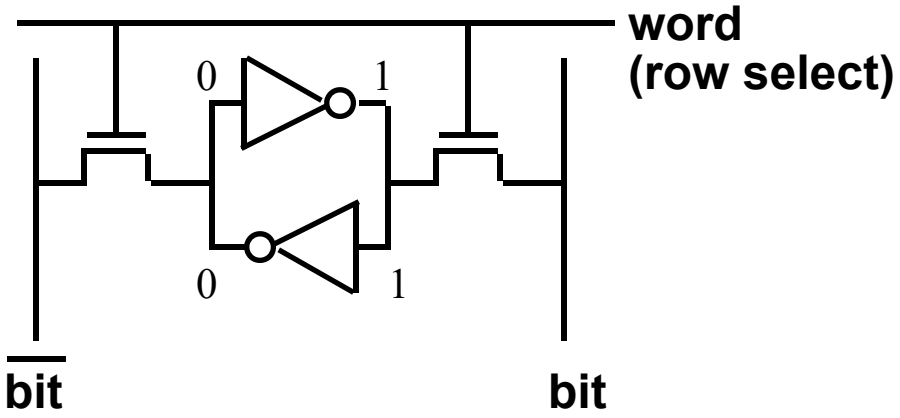
- **Performance of Main Memory:**
  - Latency: Cache Miss Penalty, e.g. in milli seconds
    - *Access Time*: time between request and word arrives
    - *Cycle Time*: time between requests
  - Bandwidth: Data read/write rate, e. g., in Mbytes/s
- **Main Memory is *DRAM :* Dynamic Random Access Memory**
  - Dynamic since needs to be refreshed periodically (8-64 ms)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - *RAS* or *Row Access Strobe*
    - *CAS* or *Column Access Strobe*
- **Cache uses *SRAM :* Static Random Access Memory**
  - No refresh (6 transistors/bit vs. 1 transistor)
    Size: DRAM/SRAM *4-8*
    Cost/Cycle time: SRAM/DRAM *8-16*

Random Access Memory (RAM) Technology

- Why do computer designers need to know about RAM technology?
  - Processor performance is usually limited by memory bandwidth
  - As IC densities increase, lots of memory will fit on processor chip
    - Tailor on-chip memory to specific needs
      - Instruction cache
      - Data cache
      - Write buffer
- What makes RAM different from a bunch of flip-flops?
  - Density: RAM is much denser

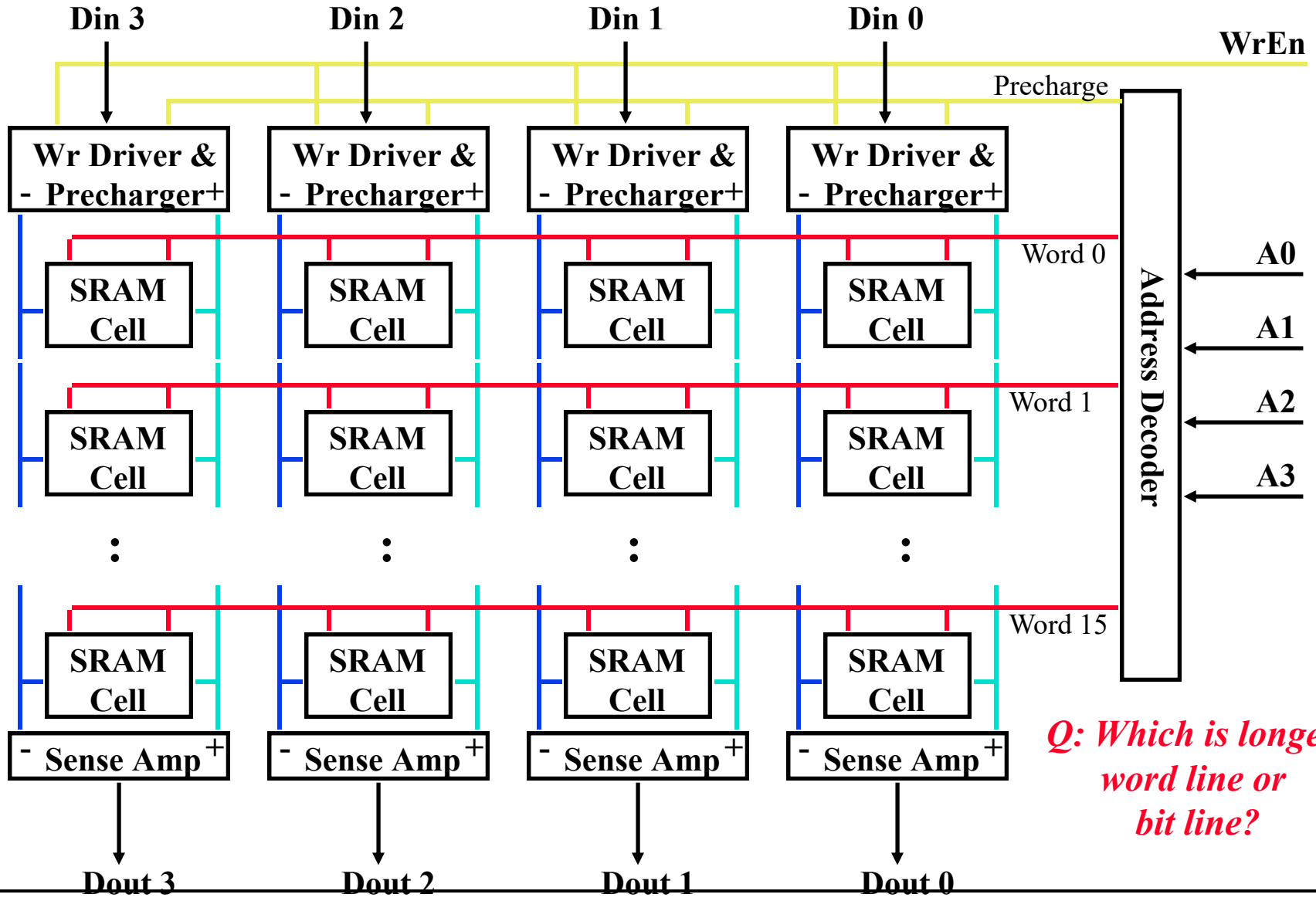
Static RAM Cell

6-Transistor SRAM Cell



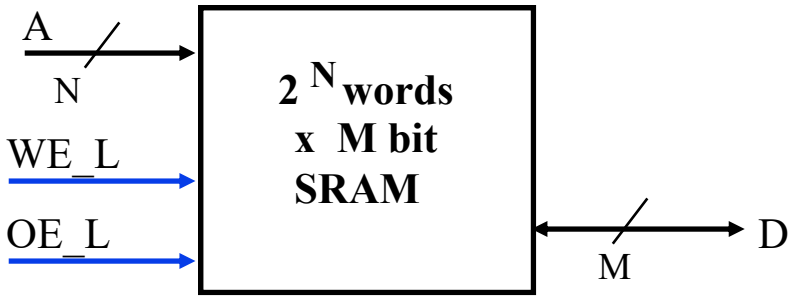
- **Write:**
  1. Drive bit lines (bit=1, bit=0)
  2. Select row
- **Read:**
  1. Precharge bit and $\overline{bit}$ to Vdd or Vdd/2 => make sure equal!
  2. Select row
  3. Cell pulls one line low
  4. Sense amp on column detects difference between bit and $\overline{bit}$

Typical SRAM Organization: 16-word x 4-bit



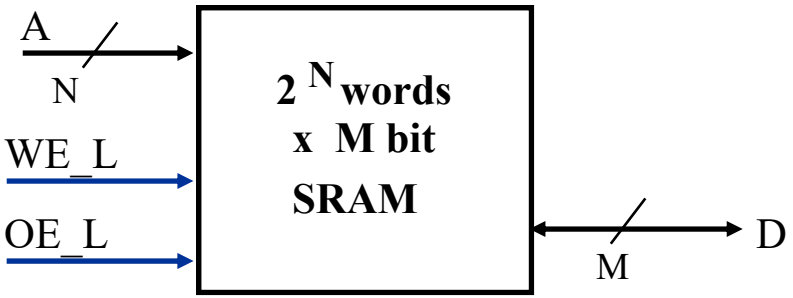
Q: Which is longer: word line or bit line?

Logic Diagram of a Typical SRAM



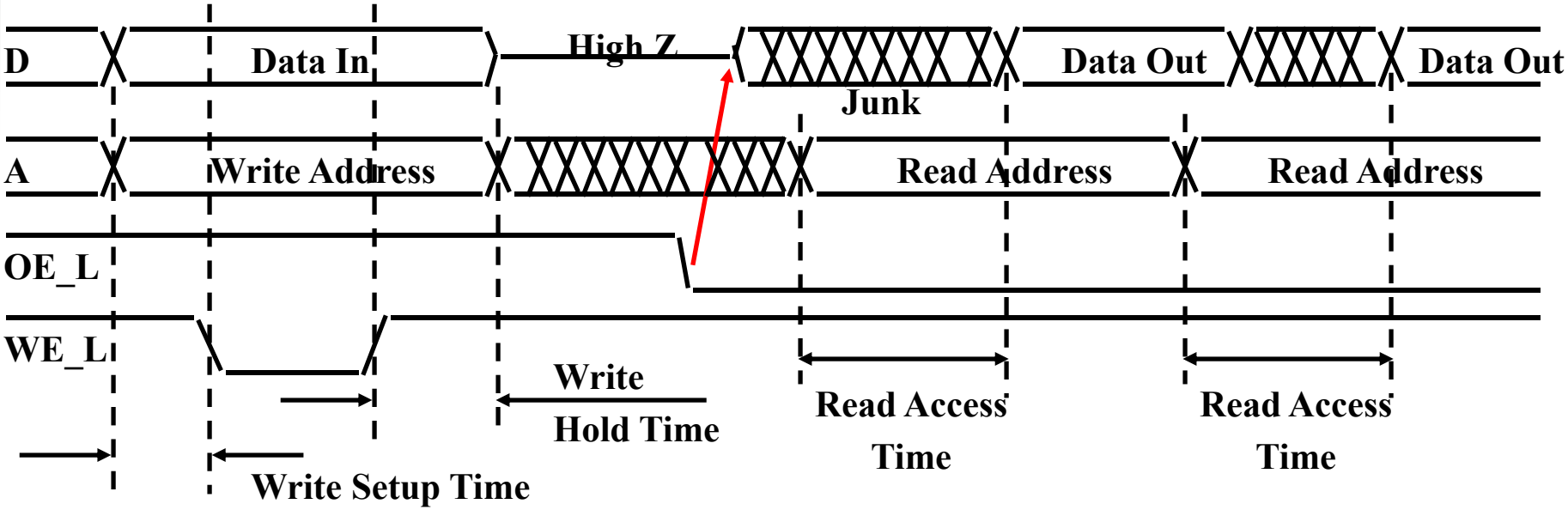
- Write Enable is usually active low (WE_L)
- Din and Dout are combined to save pins:
  - A new control signal, output enable (OE_L) is needed
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
  - Both WE_L and OE_L are asserted:
    - Result is unknown. Don't do that!!!

Typical SRAM Timing

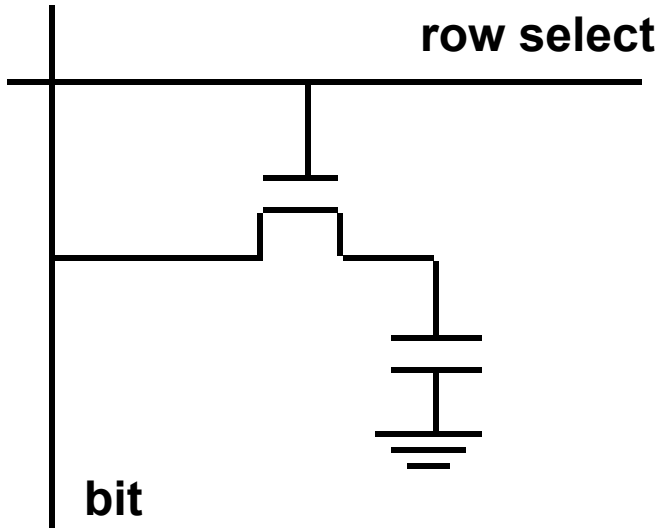


Write Timing:

Read Timing:

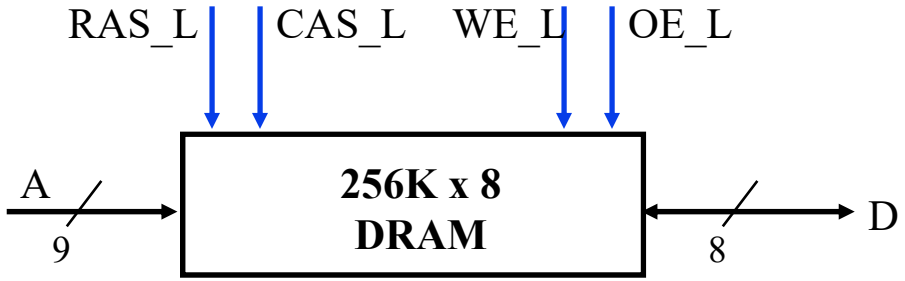


1-Transistor Memory Cell (DRAM)



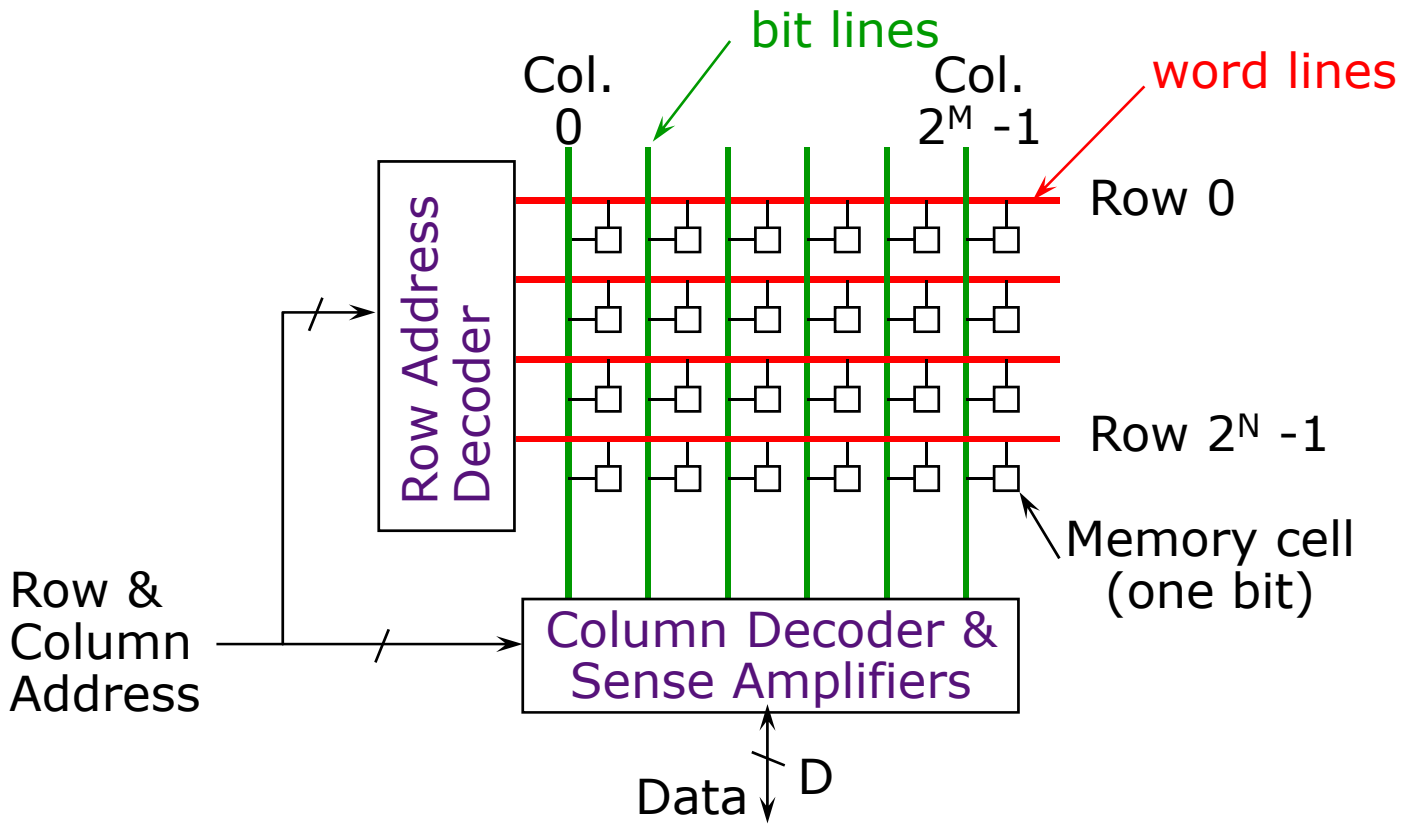
- **Write:**
  - 1. Drive bit line
  - 2. Select row
- **Read:**
  - 1. Precharge bit line to Vdd/2
  - 2. Select row
  - 3. Cell and bit line share charges
    - Very small voltage changes on the bit line
  - 4. Sense (fancy sense amp)
    - Can detect changes of ~1 million electrons
  - 5. Write: restore the value
- **Refresh**
  - 1. Just do a dummy read to every cell.

Logic Diagram of a Typical DRAM



- Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low
- Din and Dout are combined (D):
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
- Row and column addresses share the same pins (A)
  - RAS_L goes low: Pins A are latched in as row address
  - CAS_L goes low: Pins A are latched in as column address
  - RAS/CAS edge-sensitive

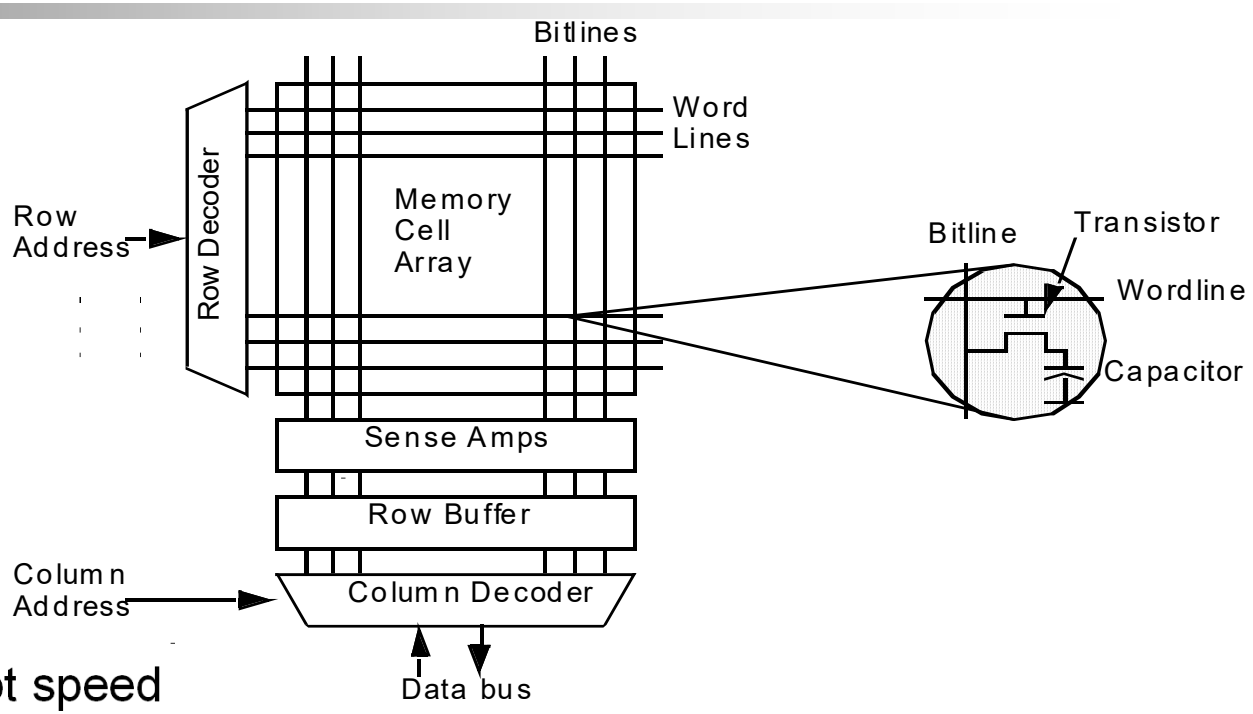
DRAM Architecture



- Row and Column Addresses are sent through the same input pins
- Bits stored in 2-dimensional arrays on chip
- Modern chips have around 4-8 logical banks on each chip
  - **each logical bank physically implemented as many smaller arrays**

DRAM Chip Organization

- Address pins are time-multiplexed
  - Row address strobe (RAS)
  - Column address strobe (CAS)



- Optimized for density, not speed
- Data stored as charge in capacitor
- Discharge on reads => destructive reads
- Cycle time roughly twice access time
- Need to precharge bitlines before access
- Charge leaks over time
  - refresh every 64ms

- DRAM in 2014
  - 8Gbit @25nm
  - 266 MHz synchronous interface
  - Data clock 2 x (1066MHz), double-data rate so 2133 MT/s

Advanced DRAM Organization

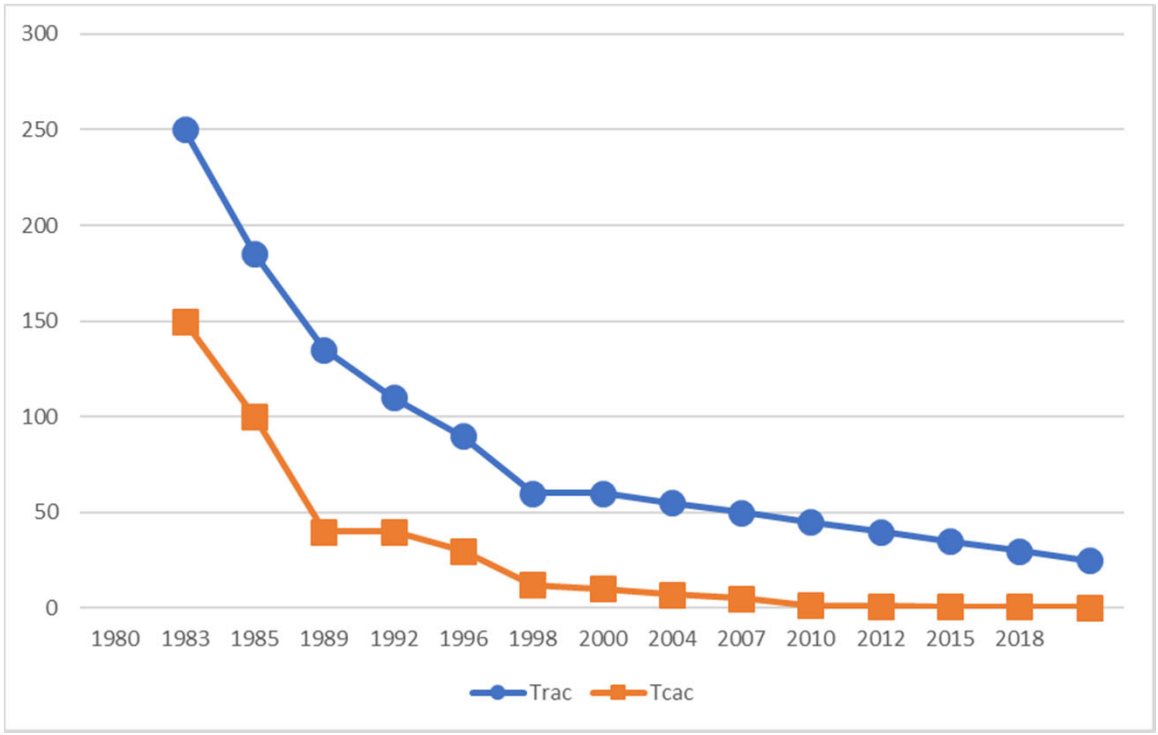
- Bits in a DRAM are organized as a rectangular array
  - DRAM accesses an entire row
  - Burst mode: supply successive words from a row with reduced latency
- Double data rate (DDR) DRAM
  - Transfer on rising and falling clock edges
- Quad data rate (QDR) DRAM
  - Separate DDR inputs and outputs

DRAM Generations

Year	Capacity	$/GB
1980	64 Kibibit	$6,480,000
1983	256 Kibibit	$1,980,000
1985	1 Mebibit	$720,000
1989	4 Mebibit	$128,000
1992	16 Mebibit	$30,000
1996	64 Mebibit	$9,000
1998	128 Mebibit	$900
2000	256 Mebibit	$840
2004	512 Mebibit	$150
2007	1 Gibibit	$40
2010	2 Gibibit	$13
2012	4 Gibibit	$5
2015	8 Gibibit	$7
2018	16 Gibibit	$6

The cost per GB is not adjusted for inflation.

Row and Column Access times, ns



In 2018, Row and Column Access times are 25 ns, and 0.4 ns, respectively.

DRAM Performance Factors

- Row buffer
  - Allows several words to be read and refreshed in parallel
- Synchronous DRAM
  - Allows for consecutive accesses in bursts without needing to send each address
  - Improves bandwidth
- DRAM banking
  - Allows simultaneous access to multiple DRAMs
  - Improves bandwidth

DRAM name based on Peak Chip Transfers / Sec
DIMM name based on Peak DIMM MBytes / Sec

Stan-dard	Clock Rate (MHz)	M transfers / second	DRAM Name	Mbytes/s/ DIMM	DIMM Name
DDR	133	266	DDR266	2128	PC2100
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10664	PC10700
DDR3	800	1600	DDR3-1600	12800	PC12800
DDR4	1600	3200	DDR4-3200	25600	PC 25600
DDR5	3200	6400	DDR5-6400	51200	PC5-51200

x 2

x 8

DRAM Operation

Three steps in read/write access to a given bank

- **Row access (RAS)**
  - decode row address, enable addressed row (often multiple Kb in row)
  - bitlines share charge with storage cell
  - small change in voltage detected by sense amplifiers which latch whole row of bits
  - sense amplifiers drive bitlines full rail to recharge storage cells
- **Column access (CAS)**
  - decode column address to select small number of sense amplifier latches (4, 8, 16, or 32 bits depending on DRAM package)
  - on read, send latched bits out to chip pins
  - on write, change sense amplifier latches which then charge storage cells to required value
  - can perform multiple column accesses on same row without another row access (burst mode)
- **Precharge**
  - charges bit lines to known value, required before next row access

Each step has a latency of around 15-20ns in modern DRAMs

Various DRAM standards (DDR, RDRAM) have different ways of encoding the signals for transmission to the DRAM, but all share same core architecture

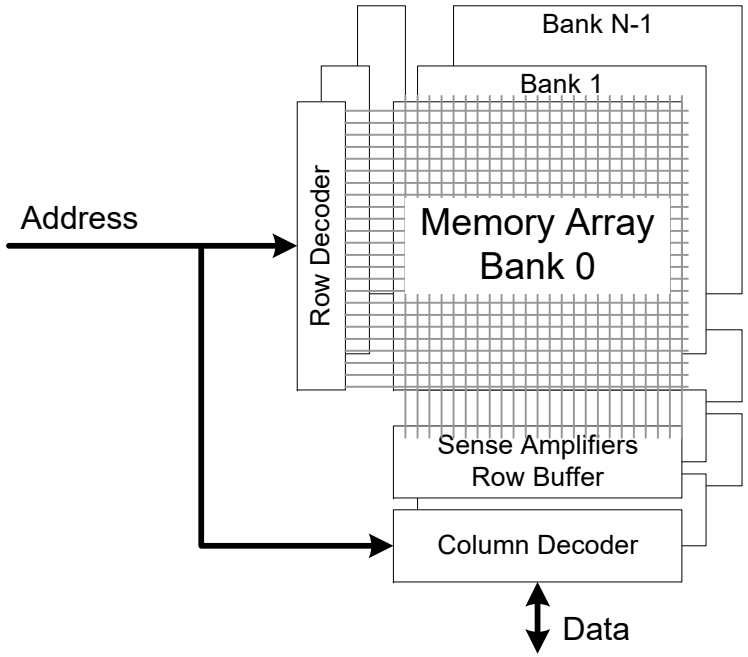
DDR SDRAM Control

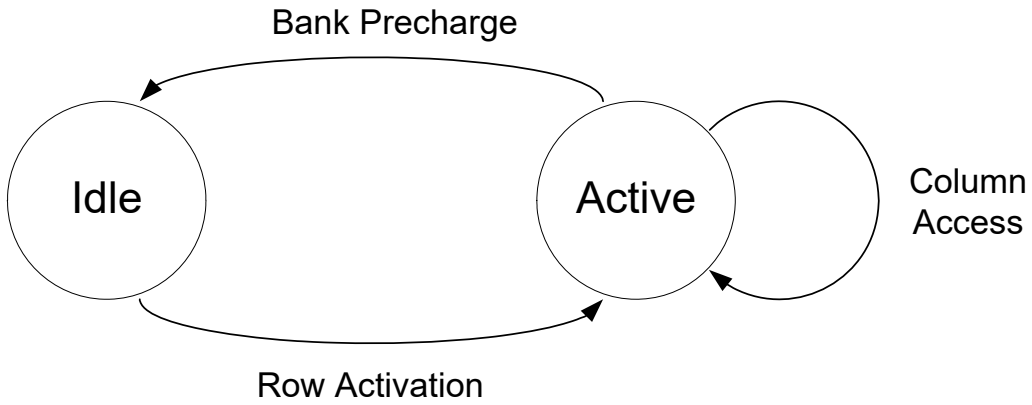
- **Commands**
  - **Activate row**

    Read row into row buffer
  - **Column access**

    Read data from addressed row
  - **Bank Precharge**

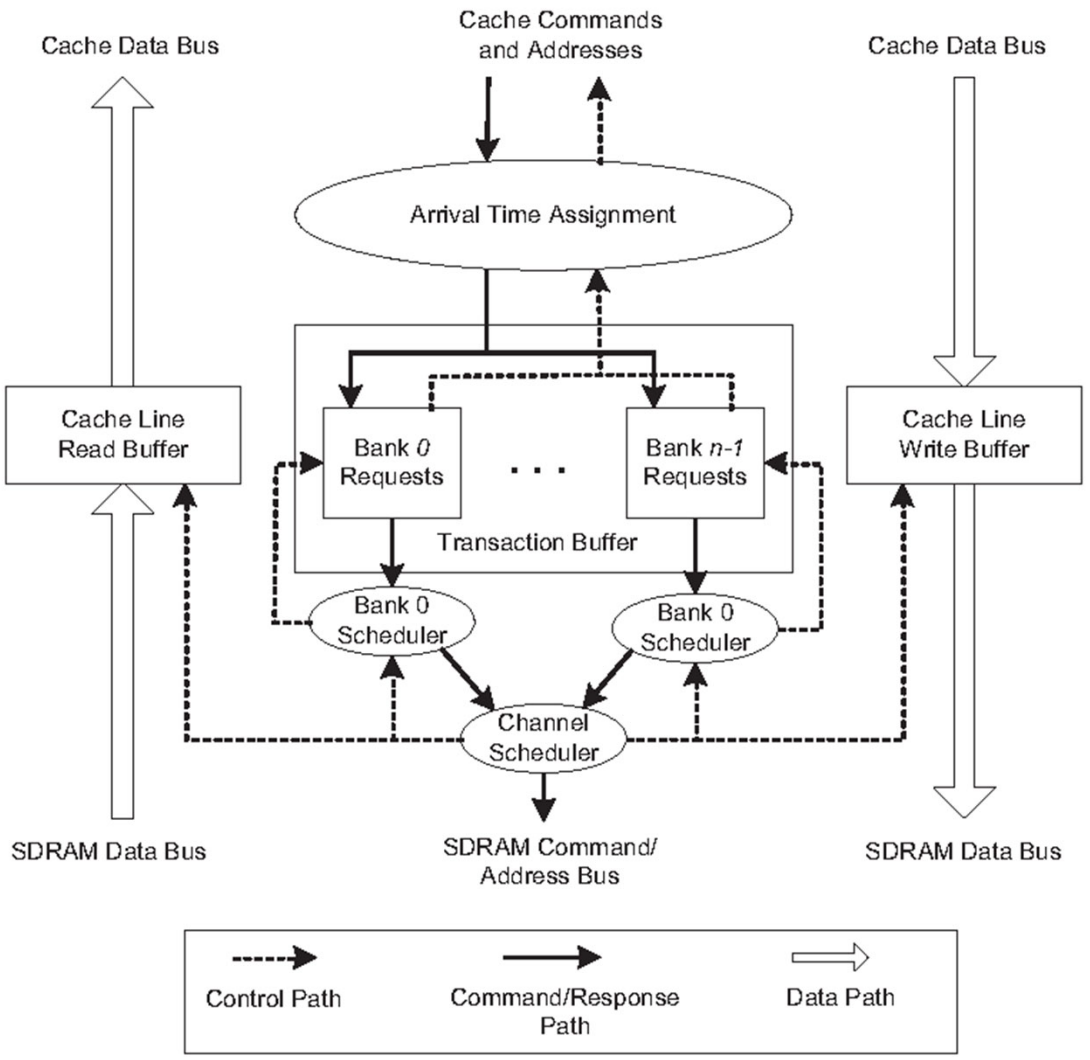
    Get ready for new row access



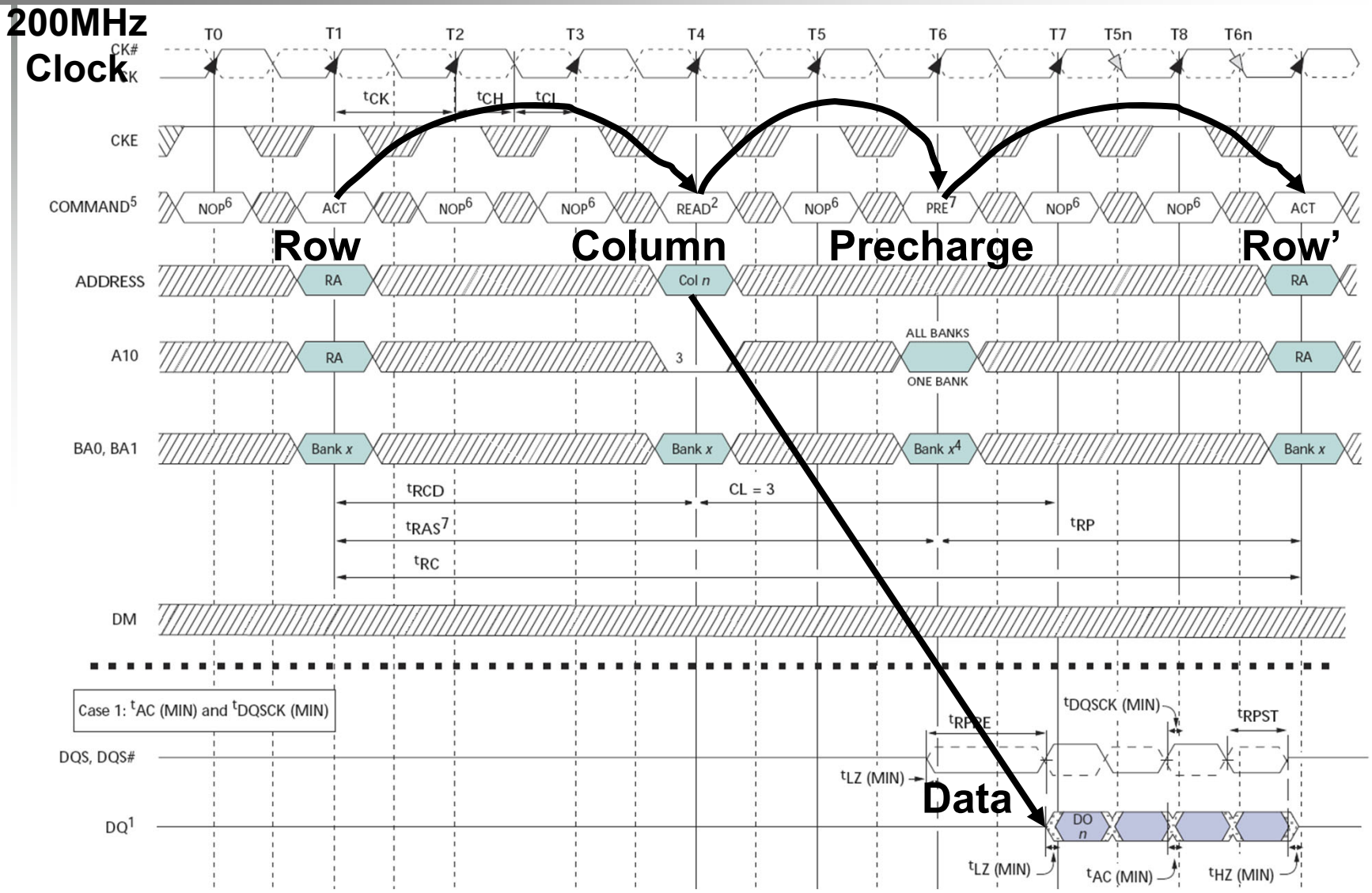


SDRAM Memory Controller

- Interface between a cache hierarchy and main memory)
- Translates read and write requests into sequences of SDRAM commands
- Memory scheduler
  - keeps track of the state of memory banks,
  - reorders and interleaves memory requests to optimize memory latency and bandwidth utilization



Double-Data Rate (DDR2) DRAM

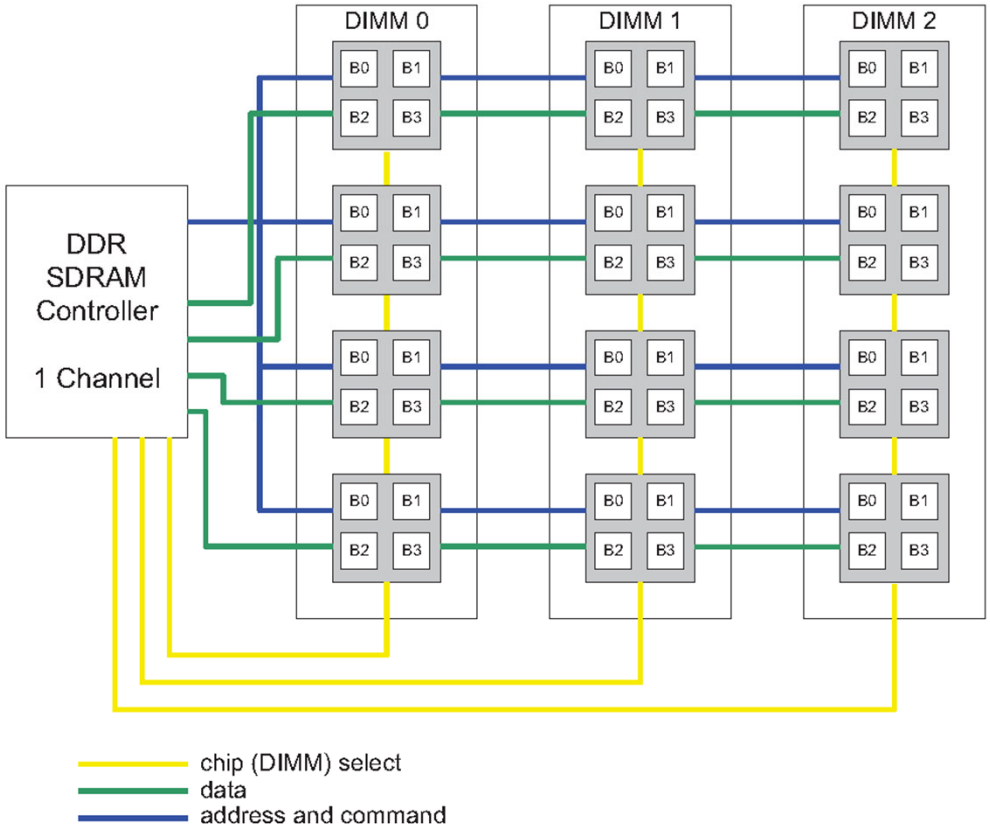


[Micron, 256Mb DDR2 SDRAM datasheet]

Constructing a Memory System

- Combine chips in parallel to increase access width
  - E.g. 4 16-bit wide DRAMs for a 64-bit parallel access
  - DIMM – Dual Inline Memory Module
- Combine DIMMs to form multiple ***ranks***
- Attach a number to DIMMs to a memory channel
  - Memory Controller manages a channel (or two lock-step channels)
- Interleave patterns:
  - Rank, Row, Bank, Column, [byte]
  - Row, Rank, Bank, Column, [byte]
    - Better dispersion of addresses
    - Works better with power-of-two ranks

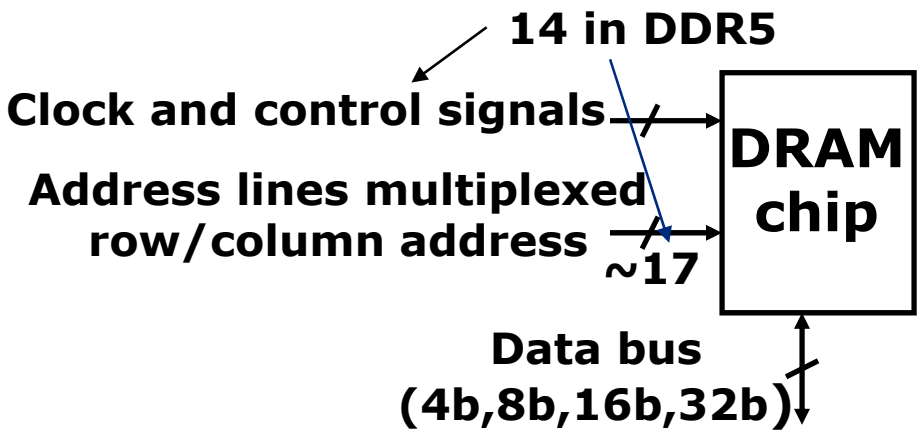
DDR SDRAM Memory System Example



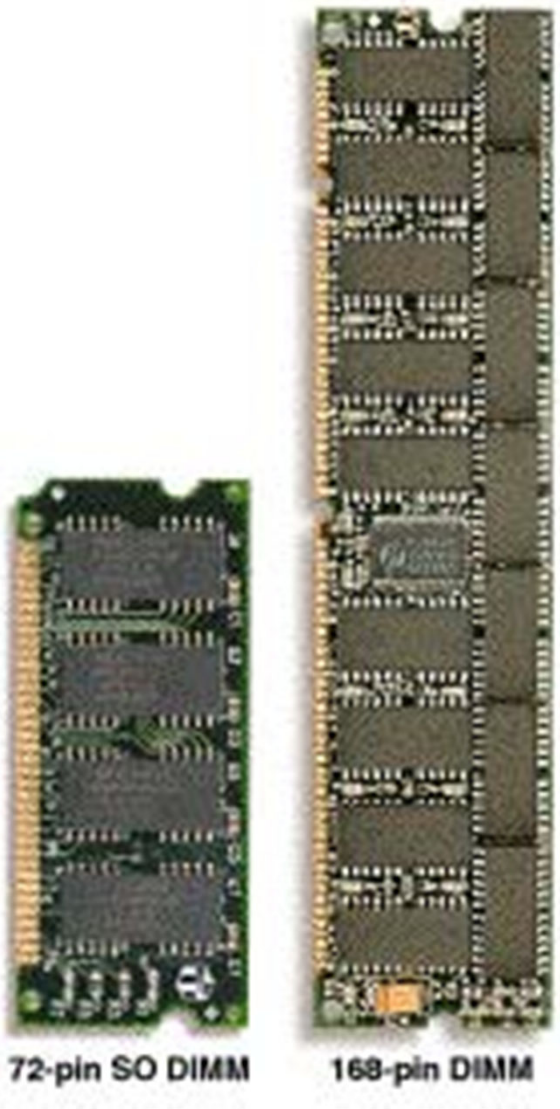
- 3 ranks = 3 DIMMs (Dual Inline Memory Modules)
- 4 16-bit SDRAM DDR chips per DIMM (64 bits per DIMM access)
- 4 memory banks (B0-3) per chip on DIMM
- Real memory address partitioning: rank|row|bank|column[byte]

DRAM Packaging

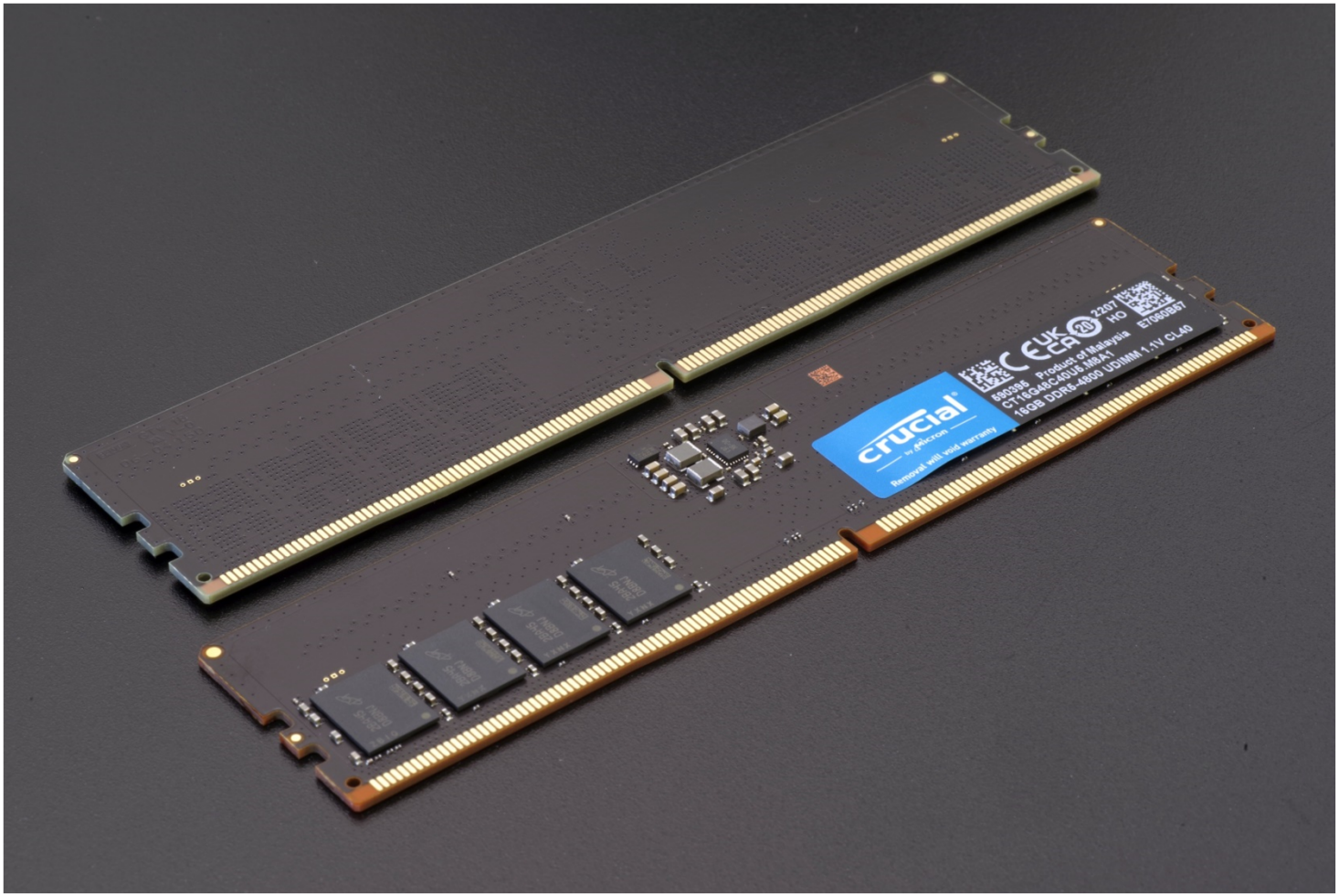
(Laptops/Desktops/Servers)



- DIMM (Dual Inline Memory Module) contains multiple chips with clock/control/address signals connected in parallel (sometimes need buffers to drive signals to all chips)
- Data pins work together to return wide word (e.g., 64-bit data bus using 16x4-bit parts)



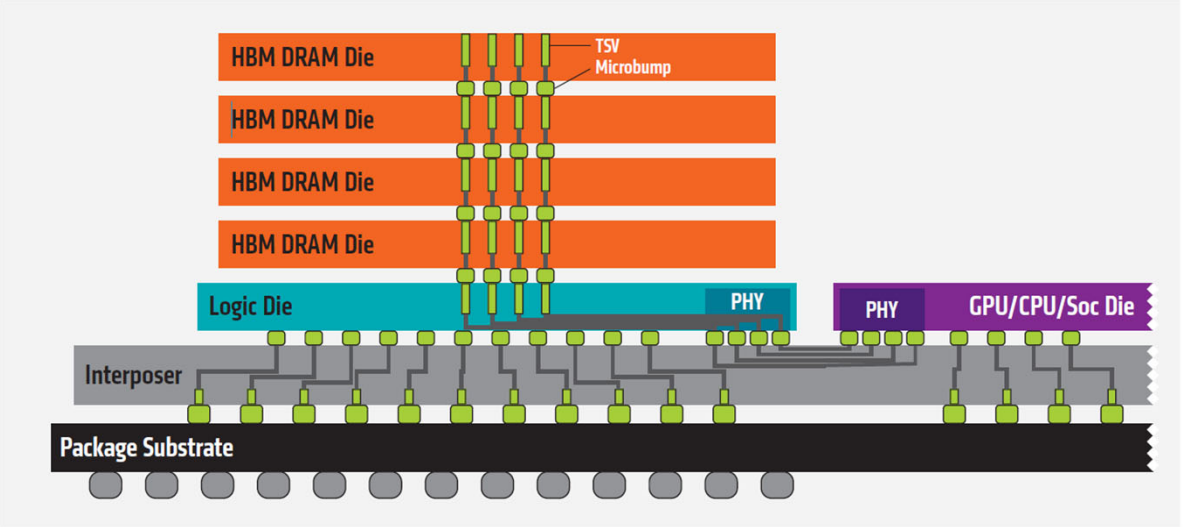
16 GB DDR5-4800 1.1V 288 Pin UDIMM (2020)



3D DRAM Stacking Technologies

Recent enabling technology: 3D stacking of DRAM chips

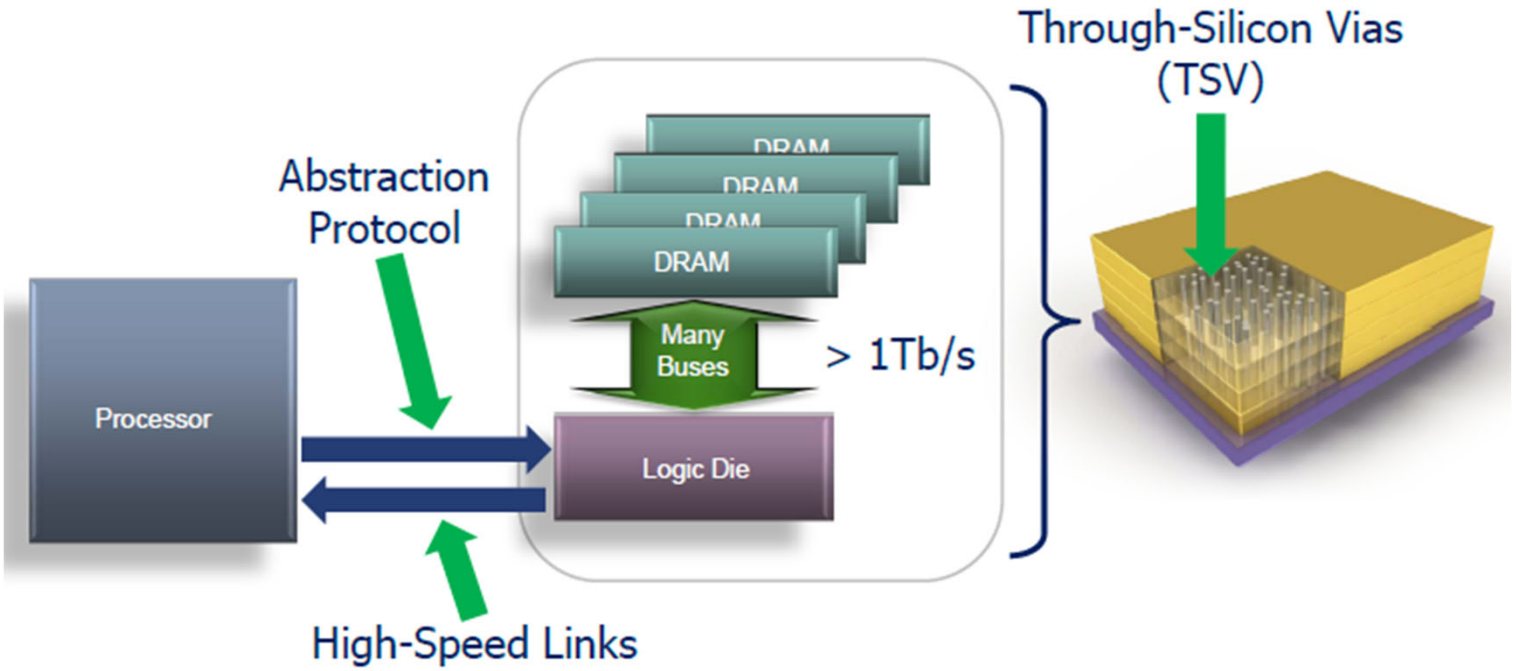
- DRAMs connected via through-silicon-vias (TSVs) that run through the chips
  - TSVs provide highly parallel connection between logic layer and DRAMs
- Base layer of stack "logic layer" is memory controller, manages requests from processor
- Silicon "interposer" serves as high-BW interconnect between DRAM stack and processor



Technologies:

- Micron/Intel's Hybrid Memory Cube (HMC)
- AMD's High-Bandwidth Memory (HBM): 1024 bit interface to stack

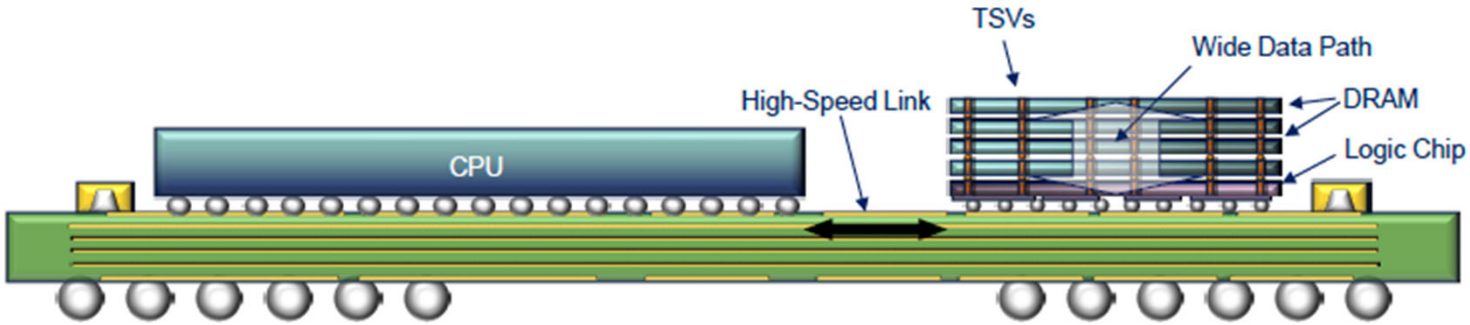
Hybrid Memory Cube (HMC)



Notes: Tb/s = Terabits / second
HMC height is exaggerated

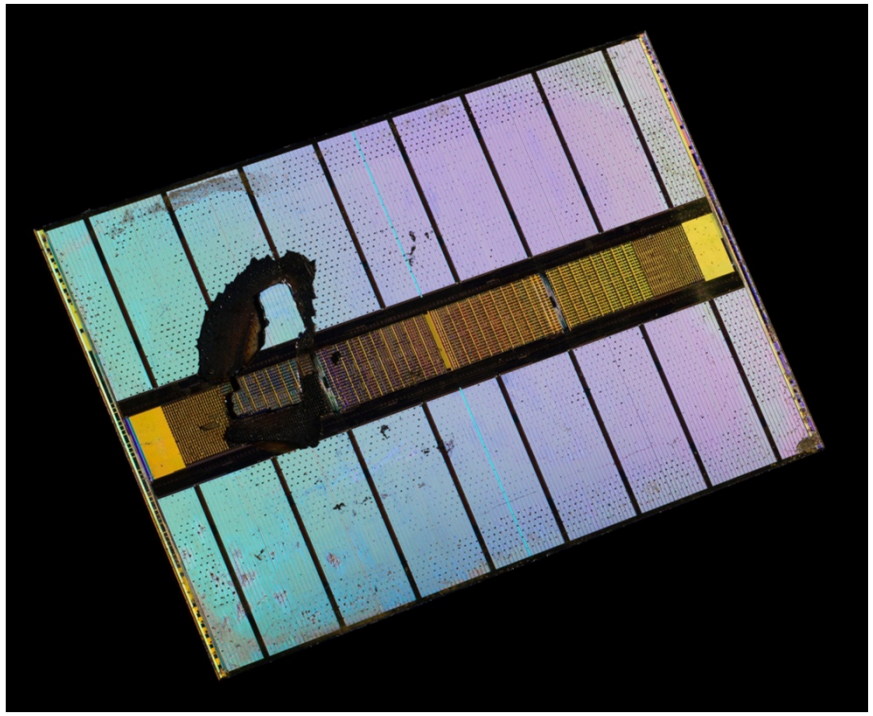
- Micron proposal [Pawlowski, Hot Chips 11]
  - www.hybridmemorycube.org

Hybrid Memory Cube MCM

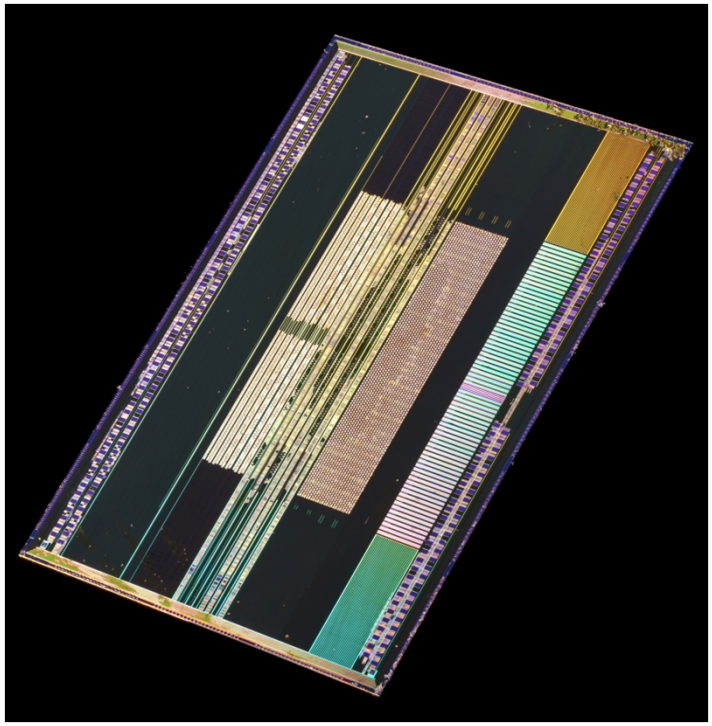


Notes: MCM = multi-chip module
Illustrative purposes only; height is exaggerated

High Bandwidth Memory (HBM)



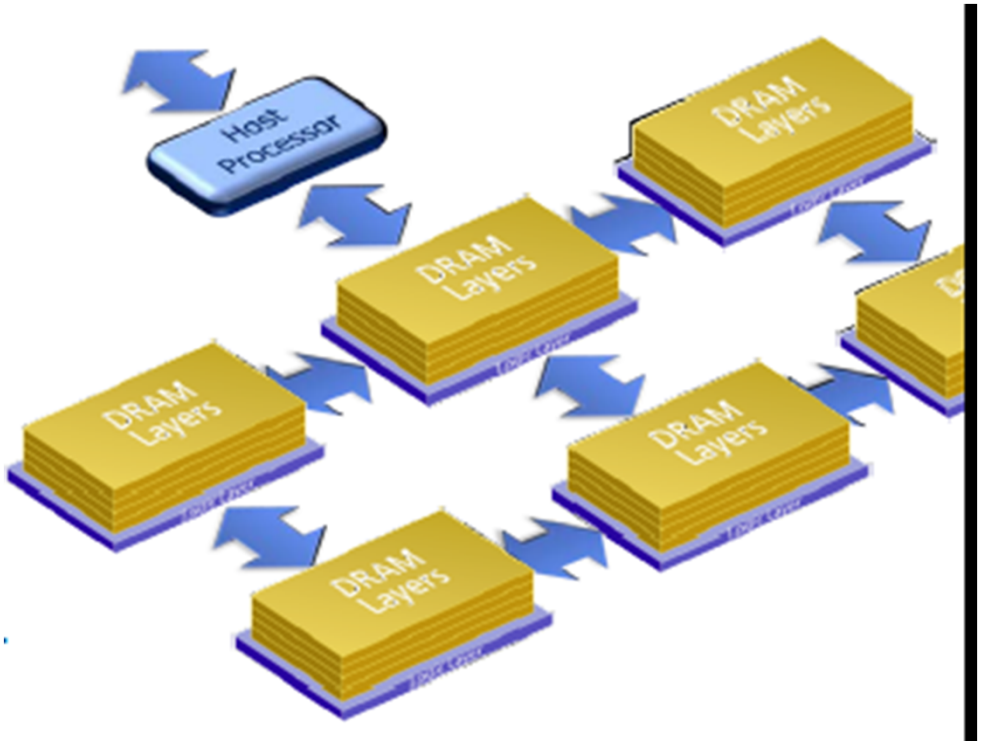
HBM2 DRAM die



HBM2 controller die

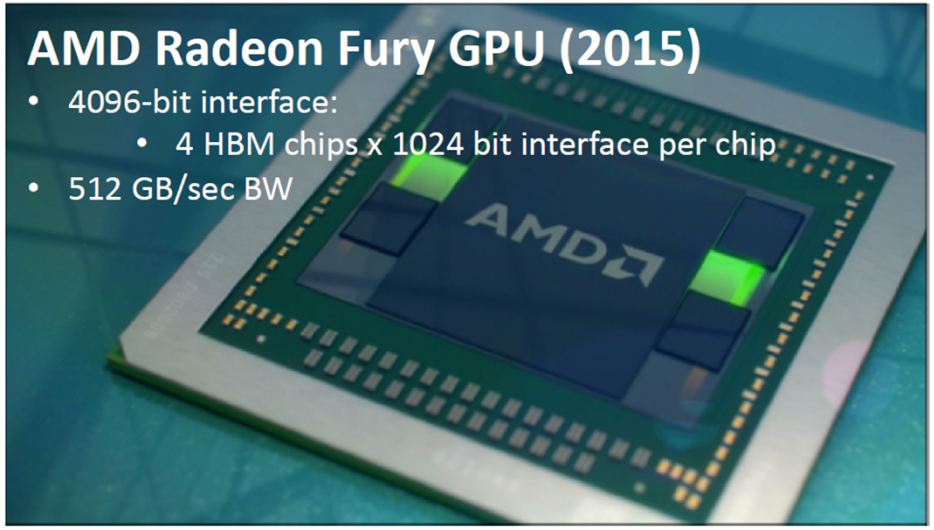
- Up to 8 dies per stack
- Up to 8 GB per stack
- With 1024-bit wide access, HBM2 has 256 GB/s memory BW / package.
- A new HBM3 standard with up to 24GB/stack was announced in 2022.

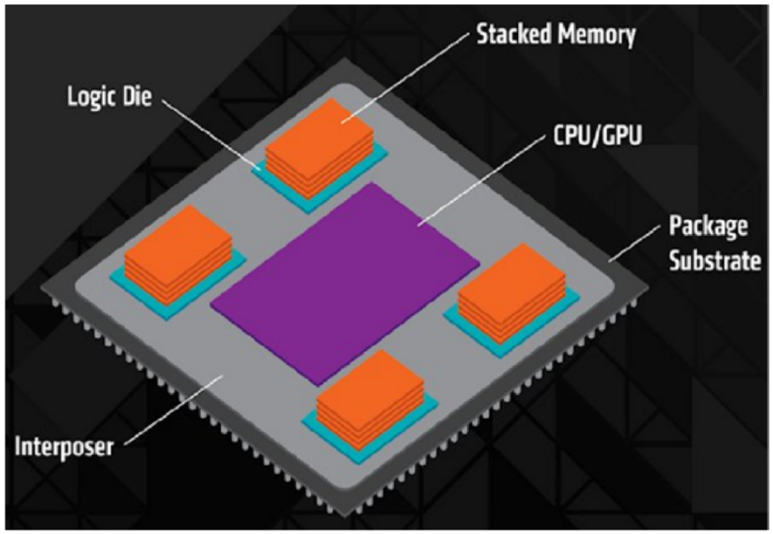
Network of DRAM

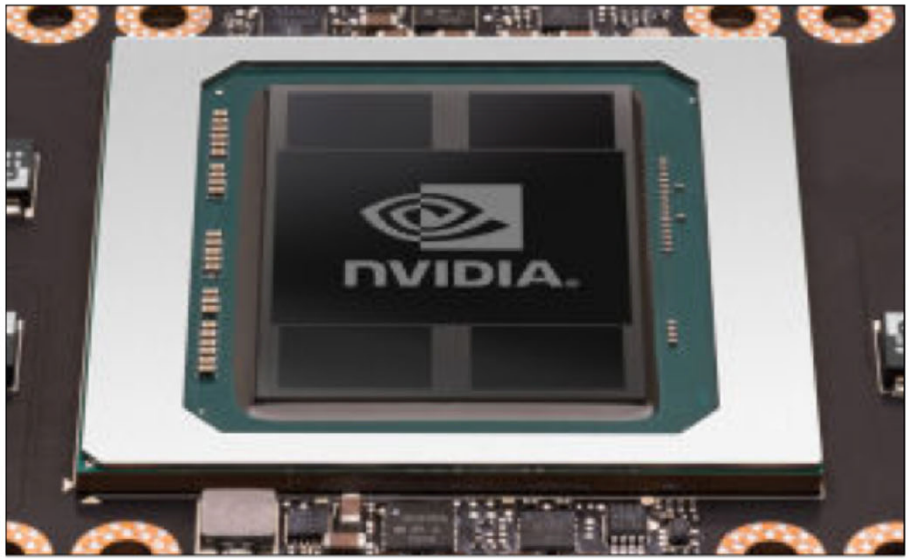


- Traditional DRAM: star topology
- HMC: mesh, etc. are feasible

3D DRAM Stacking in GPUs





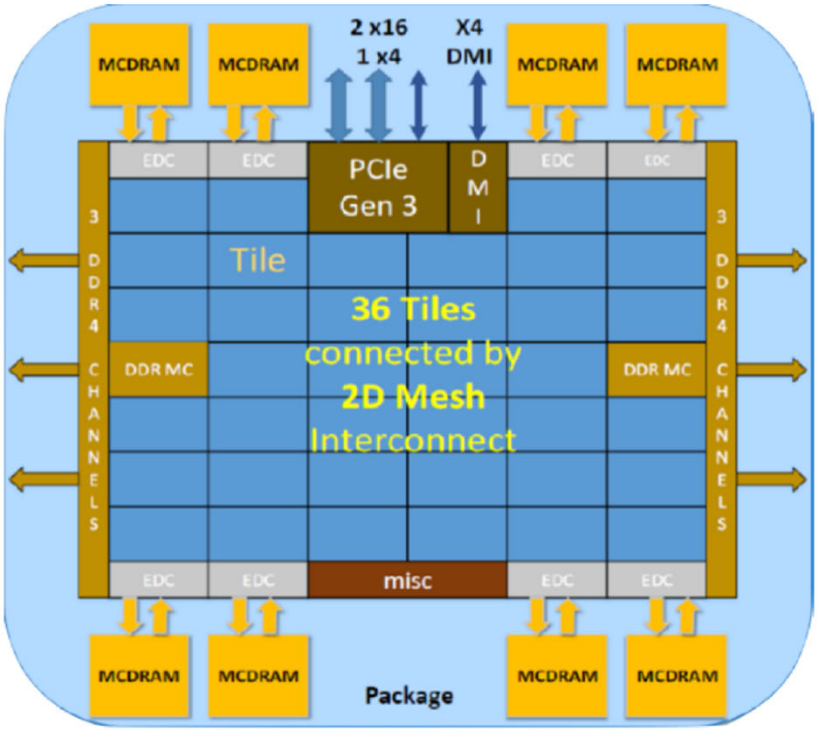


NVIDIA P100 GPU (2016)

- 4096-bit interface:
  - 4 HBM2 chips x 1024 bit interface per chip
- 720 GB/sec peak BW
- 4 x 4 GB = 16 GB capacity

Xeon Phi MCDRAM

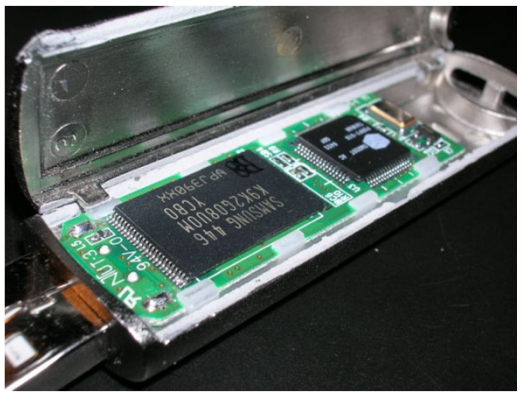
- 16 GB in package stacked DRAM
- Can be configured as either:
  - 16 GB last level cache
  - 16 GB separate address space
    - aka "flat mode"
- Intel's claims:
  - ~ same latency at DDR4
  - ~5x bandwidth of DDR4
  - ~5x less energy cost per bit transferred




```
// allocate buffer in MCDRAM ("high bandwidth" memory malloc)
float* foo = hbw_malloc(sizeof(float) * 1024);
```

Flash Storage

- Nonvolatile semiconductor storage
  - 100× – 1000× faster than disk
  - Smaller, lower power, more robust
  - But more $/GB (between disk and DRAM)



Flash Types

- Type of EEPROM
- NOR flash (faster): bit cell like a NOR gate
  - Random read/write access
  - Used for instruction memory in embedded systems
- NAND flash (denser): bit cell like a NAND gate
  - Denser (bits/area), but block-at-a-time access
  - Cheaper per GB
  - Used for USB keys, media storage, …
- Flash bits wears out after 1000's of accesses
  - Not suitable for direct RAM or disk replacement
  - Wear leveling: remap data to less used blocks

NAND Flash Memory

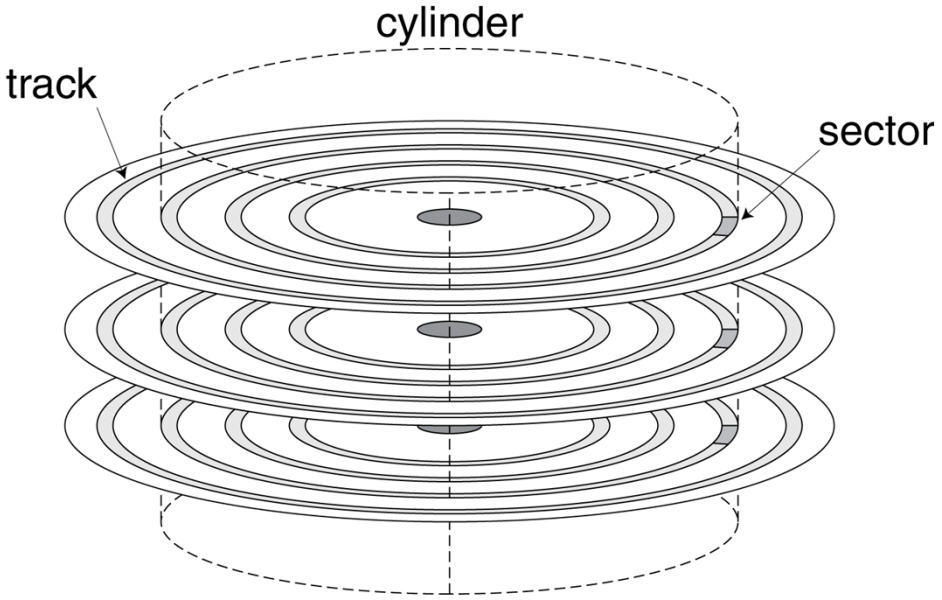
- Reads are sequential, reads entire page (.5 to 4 KB)
- 25 µs for first byte, 40 MB/s for subsequent bytes
- SDRAM: 40 ns for first byte, 4.8 GB/s for subsequent bytes
- 2 KB transfer: 75 µS vs 500 ns for SDRAM, NAND 150X slower than SDRAM
- 300 to 500X faster than magnetic disk
- Must be erased (in blocks) before being overwritten
- Nonvolatile, can use as little as zero power
- Limited number of write cycles (~100,000)
- $2/GB, compared to $20-40/GB for SDRAM and $0.09 GB for magnetic disk

Memory Dependability

- Memory is susceptible to
  - electrically charged particles trapped in the earth's magnetosphere and
  - solar and cosmic rays with ionizing electromagnetic radiation (such as X-rays and gamma rays)
- Heavy ions and other high energy particles can cause:
  - *Soft errors (Single Event)* : dynamic errors
    - Detected and fixed by error correcting codes (ECC)
  - *Hard errors*: permanent errors
    - Use spare rows to replace defective rows

Disk Storage

- Nonvolatile, rotating magnetic storage



Disk Sectors and Access

- Each sector records
  - Sector ID
  - Data (512 bytes, 4096 bytes proposed)
  - Error correcting code (ECC)
    - Used to hide defects and recording errors
  - Synchronization fields and gaps
- Access to a sector involves
  - Queuing delay if other accesses are pending
  - Seek: move the heads
  - Rotational latency
  - Data transfer
  - Controller overhead

Disk Access Example

- Given
  - 512B sector, 15,000rpm, 4ms average seek time, 100MB/s transfer rate, 0.2ms controller overhead, idle disk
- Average read time
  - 4ms seek time
    + ½ / (15,000/60) = 2ms rotational latency
    + 512 / 100MB/s = 0.005ms transfer time
    + 0.2ms controller delay
    = 6.2ms
- If actual average seek time is 1ms
  - Average read time = 3.2ms

Disk Performance Issues

- Manufacturers quote average seek time
  - Based on all possible seeks
  - Locality and OS scheduling lead to smaller actual average seek times
- Smart disk controller allocates physical sectors on disk
  - Present logical sector interface to host
  - SCSI, ATA, SATA
- Disk drives include caches
  - Prefetch sectors in anticipation of access
  - Avoid seek and rotational delay

Summary

- SRAM is fast but expensive and not very dense:
  - Good choice for providing the user FAST access time.
- DRAM is slow but cheap and dense:
  - Good choice for presenting the user with a BIG memory system
- New Stacked DRAM to replace traditional DRAM
  - in high performance systems first

Acknowledgements

- These slides contain material developed and copyright by:
  - Morgan Kauffmann (Elsevier, Inc.)
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)