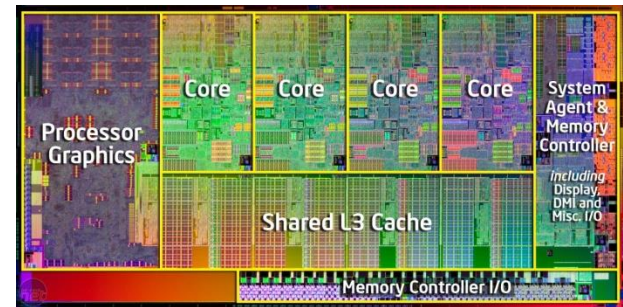


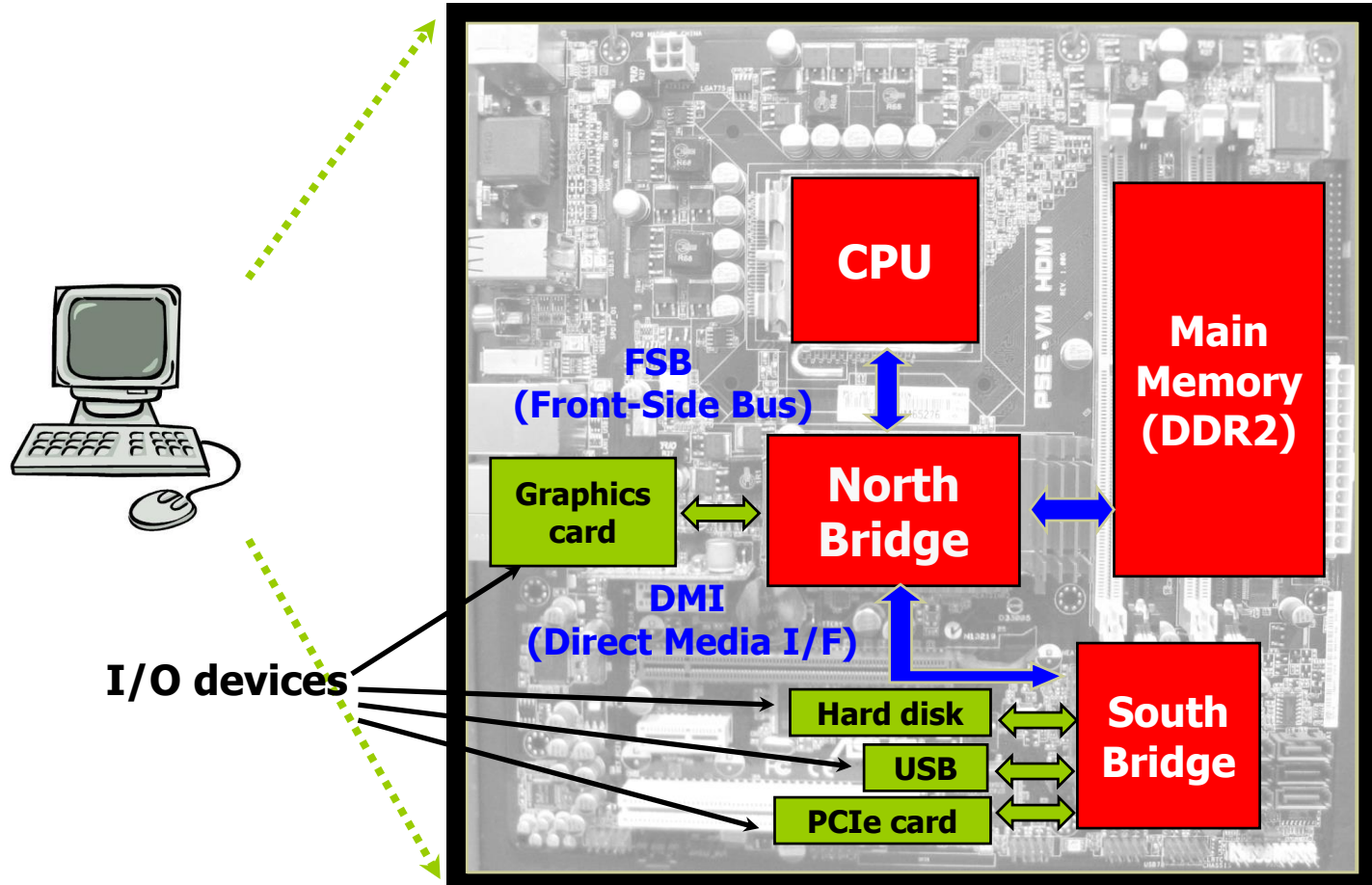


COSE222, COMP212 Computer Architecture

Lecture 8. Storage and I/O

Prof. Taeweon Suh
Computer Science & Engineering
Korea University

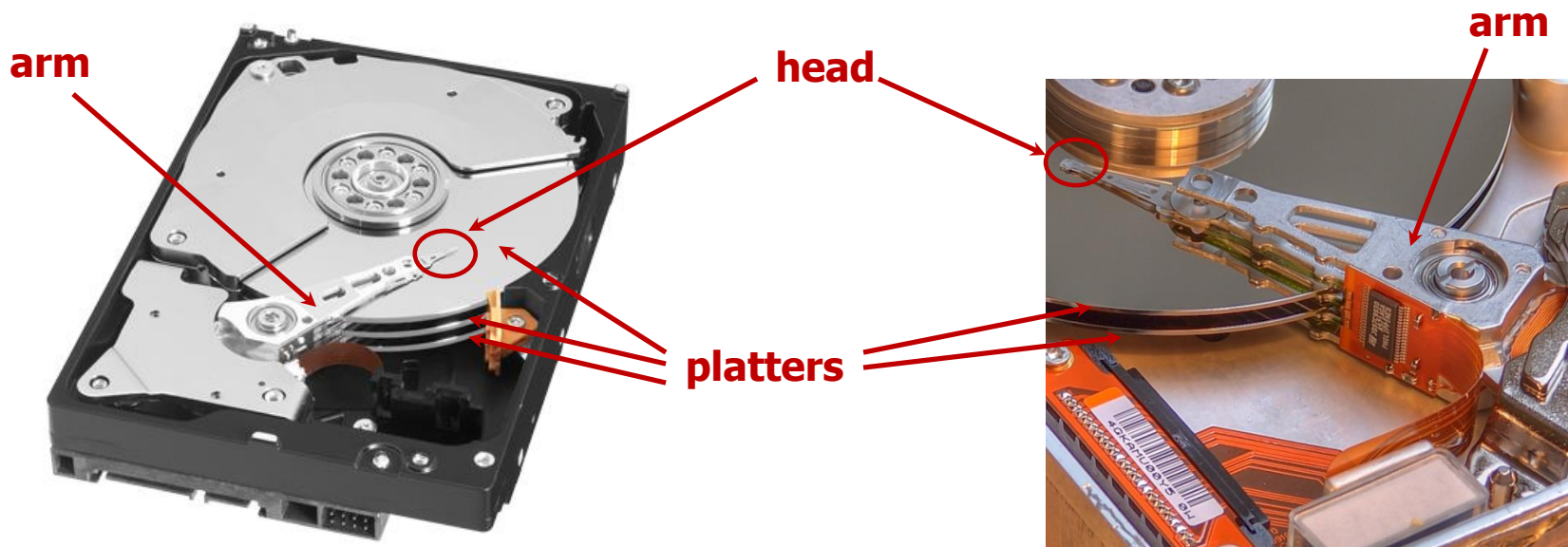
A Computer System



But, don't forget the big picture!

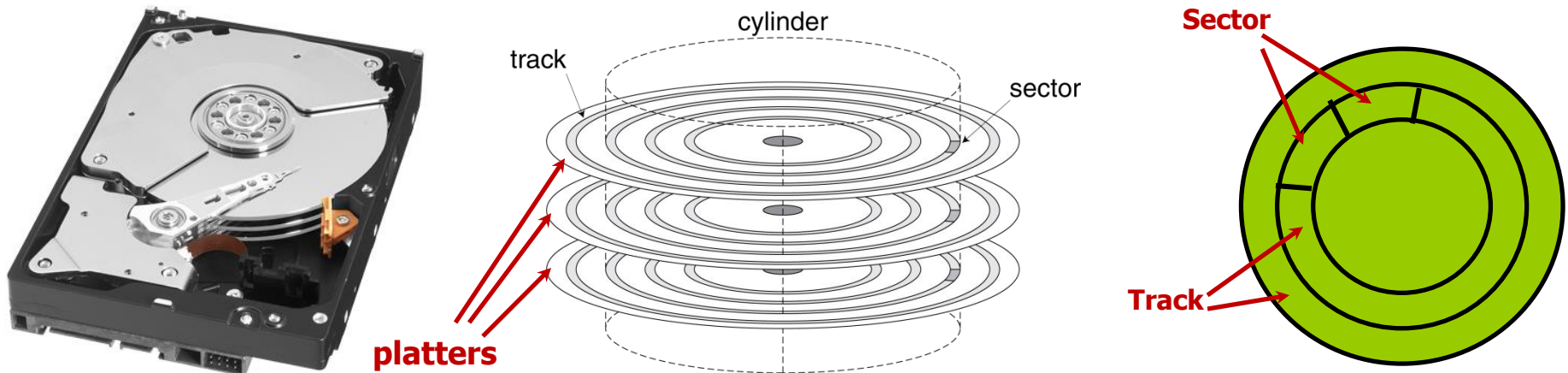
Disk Storage

- Disk (typically HDD) provides a non-volatile, long-term storage
 - It is located at the lowest level in the memory hierarchy
- Magnetic disk
 - A rotating platter is coated with magnetic material
 - Use a movable head to access the disk for read/write



Magnetic Disk Structure

- Magnetic disk has a collection of platters
 - Each platter has 2 recordable disk surfaces
 - The stack of platters is rotated at 5400 to 15,000 RPM (Revolutions Per Minute)
 - The diameter of platters is from 1-inch to just over 3.6 inches
- Each platter surface is divided into concentric circles, called tracks
 - There are typically 10,000 ~ 50,000 tracks per surface
- Each track is, in turn, divided into sectors that contain information
 - Each track may have 100 ~ 500 sectors
 - Each sector stores typically 512 bytes



Magnetic Disk Characteristic

- The disk access (read or write) involves the following steps
 1. **Seek**: position head over the proper track
 - **Seek time**: time to position the head over the proper track
 - Average seek time is usually advertized as 3ms ~ 13ms
 2. **Rotation**: rotate platter to position the desired section under the head
 - **Rotation latency**: time to position the head over the desired sector
 - Average rotation latency = $(0.5 \text{ rotation} / 5,400\text{RPM} = 5.6\text{ms}) \sim (0.5 \text{ rotation} / 15,000\text{RPM} = 2.0\text{ms})$



Magnetic Disk Characteristic (Cont)

3. Transfer: transfer a block of bits (1 or more sectors) to the disk controller's cache

- 70 to 125 MB/s are typical disk transfer rates in 2008
- The disk controller's "cache" takes advantage of spatial locality in disk accesses
 - Cache transfer rates are much faster (e.g., 375 MB/s)

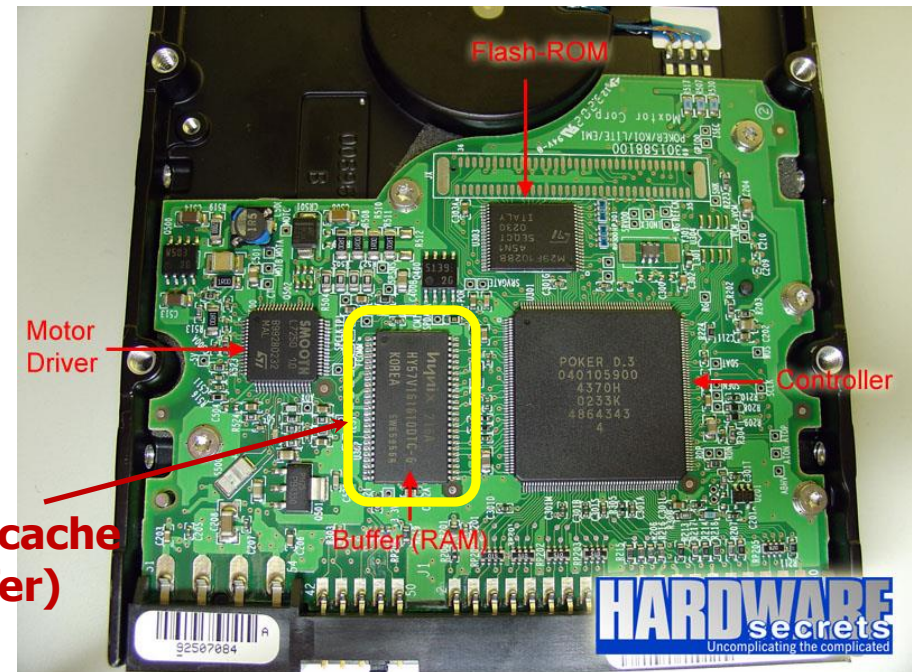
4. Controller overhead: the overhead the disk controller imposes in performing a disk I/O access

- **Disk access time**

= seek time + rotation latency +
transfer time + controller overhead

  	  
Western Digital Scorpio Blue 320GB 2.5" SATA 3.0Gb/s Internal Notebook Hard Drive ...	Seagate Barracuda LP SATA 3.0Gb/s Hard Drive - Bare Drive
• 320GB • 5400 RPM 8MB Cache • SATA 3.0Gb/s	• 2TB • 5900 RPM 32MB Cache • SATA 3.0Gb/s

**Hard-disk cache
(or buffer)**



Example

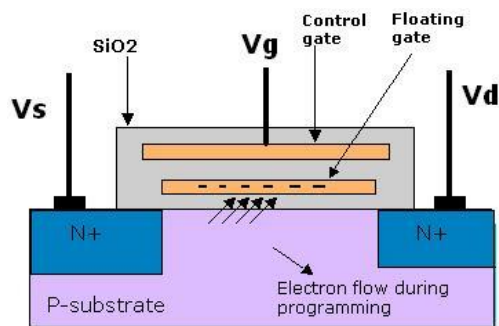
- What is the average time to read or write a 512B sector for a typical disk rotating at 15,000RPM?
 - Avg. seek time = 4ms
 - Transfer rate = 100MB/sec
 - Controller overhead = 0.2ms
- **Disk access time**
= avg. seek time + avg. rotation latency + transfer time + controller overhead
 - Avg. seek time = 4 ms
 - Avg. rotation latency = $(0.5 \text{ rotation} / 15,000\text{RPM}) = 2\text{ms}$
 - Transfer time = $512\text{B} / (100\text{MB/sec}) = 0.005 \text{ ms}$
 - Controller overhead = 0.2 ms
- **Disk access time**
= $4\text{ms} + 2\text{ms} + 0.005\text{ms} + 0.2 \text{ ms} = 6.2 \text{ ms}$
- Rotation latency is often the largest component of access time!

Magnetic Disk Examples

Feature	Seagate ST31000340NS	Seagate ST973451SS	Seagate ST9160821AS
Disk diameter (inches)	3.5	2.5	2.5
Capacity (GB)	1000	73	160
# of surfaces (heads)	4	2	2
Rotation speed (RPM)	7,200	15,000	5,400
Transfer rate (MB/sec)	105	79-112	44
Minimum seek (ms)	0.8r-1.0w	0.2r-0.4w	1.5r-2.0w
Average seek (ms)	8.5r-9.5w	2.9r-3.3w	12.5r-13.0w
MTTF (hours@25°C)	1,200,000	1,600,000	??
Dim (inches), Weight (lbs)	1x4x5.8, 1.4	0.6x2.8x3.9, 0.5	0.4x2.8x3.9, 0.2
GB/cu.inch, GB/watt	43, 91	11, 9	37, 84
Power: op/idle/sb (watts)	11/8/1	8/5.8/-	1.9/0.6/0.2
Price in 2008, \$/GB	~\$0.3/GB	~\$5/GB	~\$0.6/GB

Flash Storage

- Flash memory is the first credible challenger to disks
- Flash memory is **semiconductor memory** (like DRAM and SRAM) that is **nonvolatile** like disks
- Flash memory has latency 100 to 1000 times faster than disk and is smaller, more power efficient, and more shock resistant
- In 2008, the price of flash is \$4 to \$10 per GB
 - About 2 to 10 times higher than disk
 - About 5 to 10 times lower than DRAM



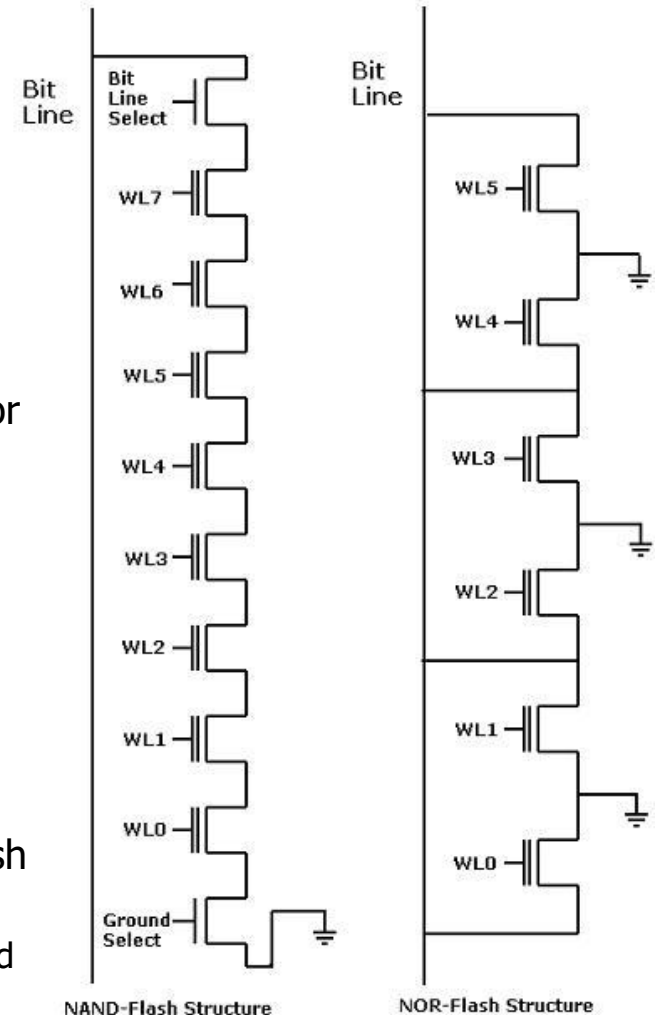
floating gate transistor



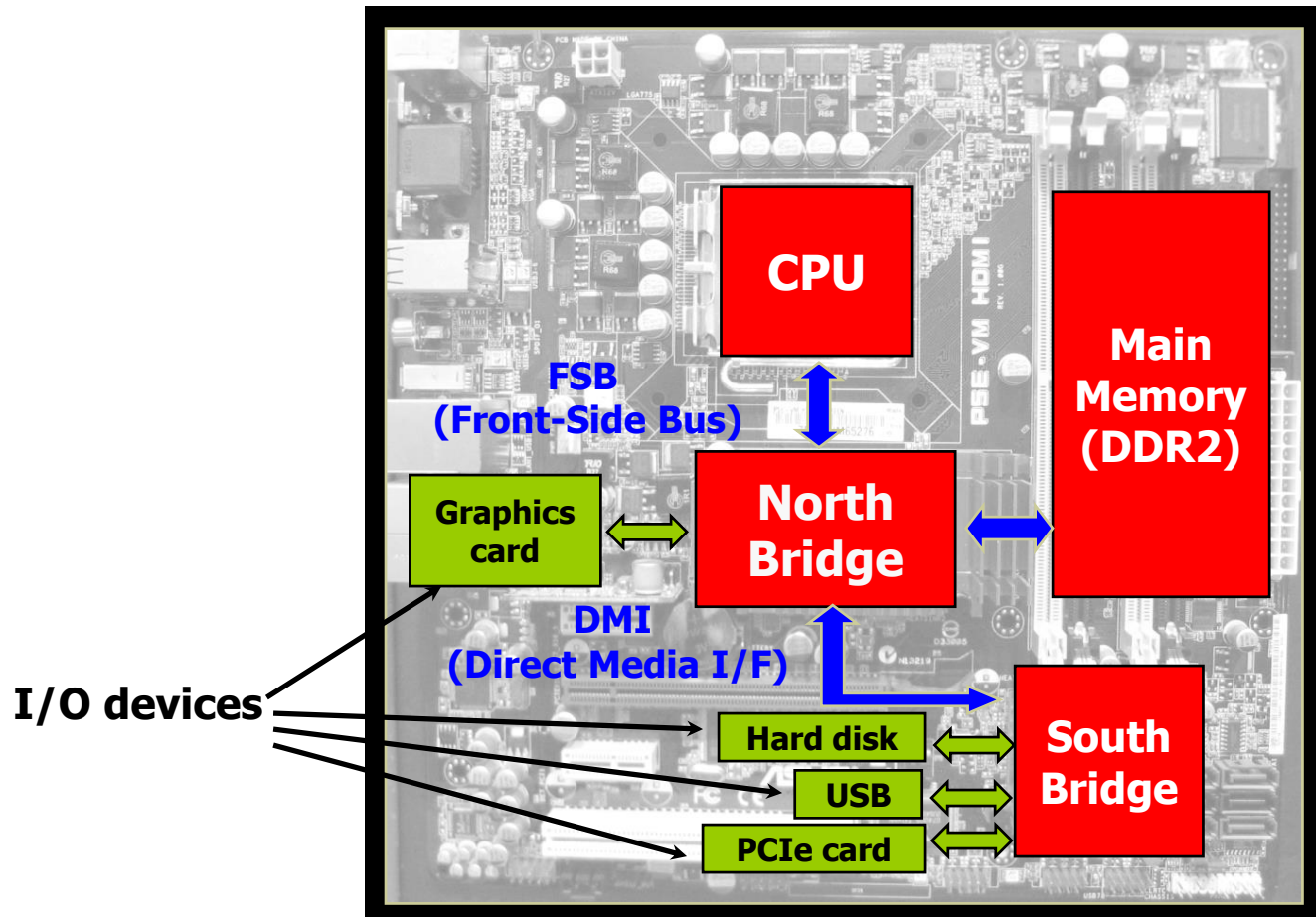
**250 GB SSD
(Solid State Drive)**

Flash Types

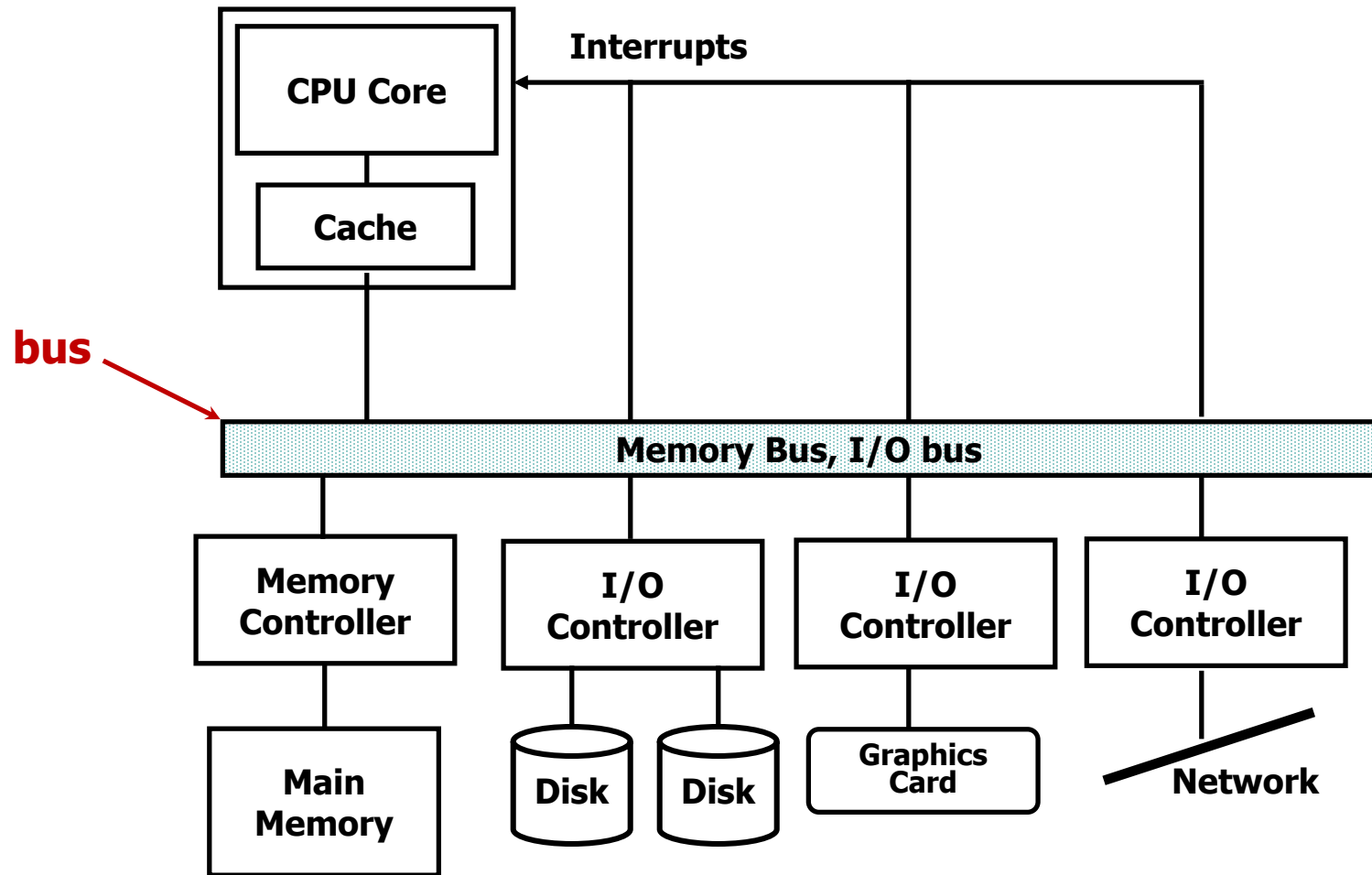
- **NOR** flash: bit cell like a NOR gate
 - Random read/write access
 - Used for BIOS memory
- **NAND** flash: bit cell like a NAND gate
 - Denser (bits/area)
 - Cheaper per GB
 - Read is on a page basis (page is typically 512B, 2048B or 4096B)
 - Used for USB keys and SSDs
- Limitations of Flash memory
 - Erasure on a block basis (typical block sizes are 64KB, 128KB, or 256KB)
 - Erasing means setting all bits in a block to 1
 - Flash bits wears out after 100,000 accesses for NOR flash and after 10,000~100,000 accesses for NAND flash
 - Wear leveling: hardware controller remaps data to less used blocks



A Computer System



A Typical I/O System Schematic (Simplified)



I/O Interconnection

- A **bus** is a shared communication link
 - A single set of wires used to connect multiple components
 - Composed of address bus, data bus, and control bus (read/write)
 - Advantages
 - Versatile – new devices can be added easily and can be moved between computer systems that use the same bus standard
 - Low cost – a single set of wires is shared in multiple ways
 - Disadvantages
 - Communication bottleneck – bus **bandwidth** limits the maximum I/O **throughput**
- The maximum bus speed is largely limited by
 - The **length** of the bus
 - The **number** of devices on the bus

I/O Interconnection (Cont)

- I/O devices and interconnection largely contribute to the performance of computer system
- Traditionally, parallel shared wires had (have) been used to connect I/O devices
- As the clock frequency increases for communicating with I/O devices, parallel shared wires suffer from clock skew and interference among wires
- Industry transitioned from parallel shared buses to high-speed serial point-to-point interconnections

Types of Buses

- **Processor-memory bus**

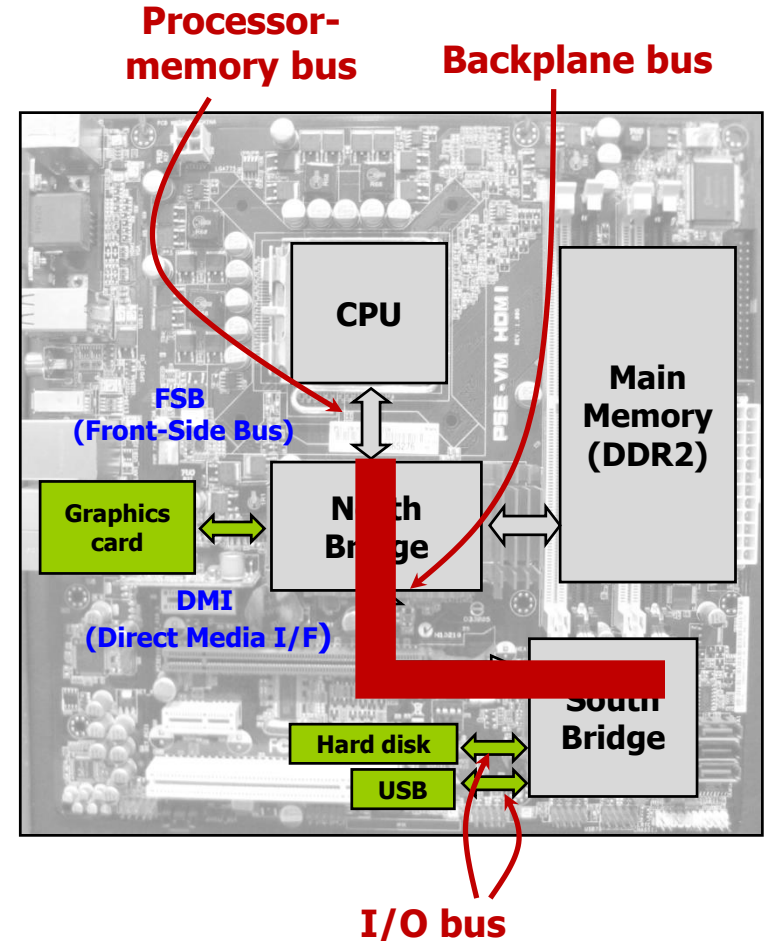
- Front Side Bus (FSB), proprietary bus
 - Replaced by QPI (QuickPath Interconnect) in Intel
 - Replaced by Hypertransport in AMD
- Short and high speed
- Matched to the memory system to maximize the memory-processor bandwidth
- Optimized for cache block transfers

- **Backplane (backbone) bus**

- Industry standard
 - e.g., PCIexpress
- Allow processor, memory and I/O devices to coexist on a single bus
- Used as an intermediary bus connecting I/O busses to the processor-memory bus

- **I/O bus**

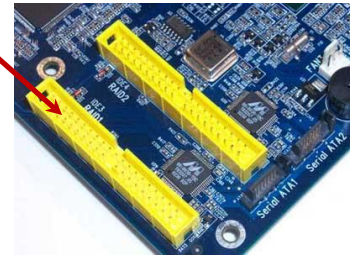
- Industry standard
 - e.g., SATA, USB, Firewire
- Usually is lengthy and slower
- Needs to accommodate a wide range of I/O devices



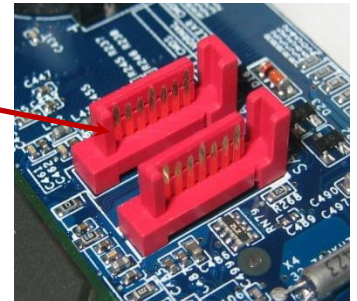
Disk Interface Standards

- ATA (Advanced Technology Attachment)
 - Evolved from IDE (Integrated Drive Electronics) standard
 - Parallel ATA: Interface standard for the connection of storage devices such as hard disks, solid-state drives, and CD-ROM drives in computers
 - Serial ATA (SATA) offers faster and more efficient data transfer compared to PATA. SATA has mostly replaced PATA in PCs
- SCSI (Small Computer Systems Interface)
 - Pronounced “skuzzy”
 - Used for hard disks and tape drives

PATA Interface



SATA Interface



SCSI Connectors

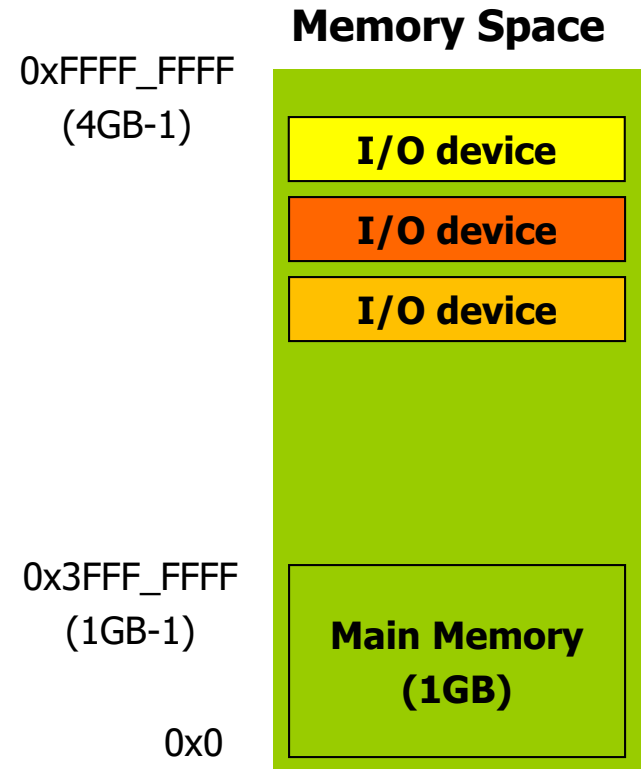


How Does CPU Access I/O Devices?

- All the I/O devices have registers implemented, so software programmers can use them to control the devices
 - Then, for programming, where and how to write to or read from?
 - There are 2 ways to access I/O devices
 - Memory-mapped I/O
 - I/O-mapped I/O

- **Memory-mapped I/O**

- I/O device is mapped to a memory space
- CPU generates a **memory transaction** to access I/O device
- To access I/O device
 - In MIPS, use `lw` or `sw` instructions
 - In x86, use `mov` instruction



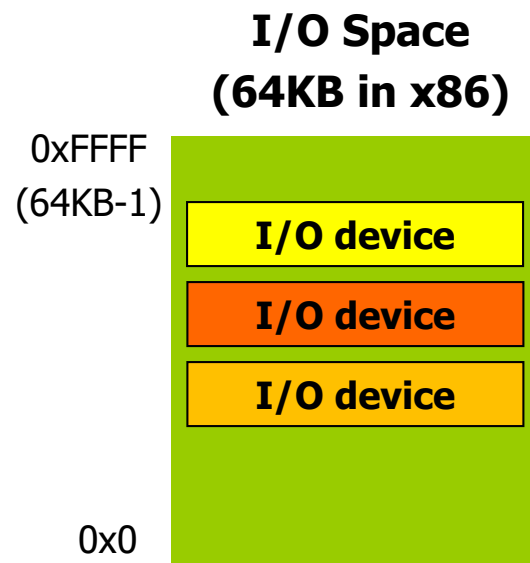
How CPU Accesses I/O Devices?

- **I/O-mapped I/O**

- I/O devices are mapped to I/O space
- CPU generates **I/O transaction** to access I/O device
- To access I/O device
 - In x86, there are `in` and `out` instructions.
 - In x86, I/O space is 64KB

- To differentiate memory space and I/O space, there should be hardware support

- ISA support
 - In x86, `mov` instruction for memory transaction and `in, out` instruction for I/O transaction
- Physical pin from processor indicating the transaction type (memory or I/O)
 - For example, the pin is driven to "1" for memory transaction or "0" for I/O transaction



How I/O Communicates with CPU?

- **Polling**

- CPU periodically checks the status of I/O devices to determine its need for service
 - CPU is totally in control
 - Can waste a lot of CPU time due to speed differences

- **Interrupt**

- I/O device issues an interrupt to indicate that it needs attention
- An I/O interrupt is **asynchronous** wrt (with respect to) instruction execution
 - It is not associated with any instruction, so doesn't prevent any instruction from completing
 - You can pick your own convenient point in the pipeline to handle the interrupt

DMA (Direct Memory Access)

- Typically, moving data from one place to another involve CPU instructions
 - Load (l_w) from a location (e.g. memory in an I/O device)
 - Store (s_w) to another location (e.g. main memory)
 - Moving a large chunk of data with CPU instructions could take a large fraction of CPU time
- DMA has the ability to transfer large blocks of data **directly** to/from the memory **without involving the processor**
 1. The processor initiates the DMA transfer by supplying source and destination addresses, the number of bytes to transfer
 2. The DMA controller manages the entire transfer (possibly thousand of bytes in length), arbitrating for the bus
 3. When the DMA transfer is complete, the DMA controller interrupts the processor to inform that the transfer is complete
- There may be multiple DMA devices in one system
 - Processor and DMA controllers contend for bus cycles and for memory

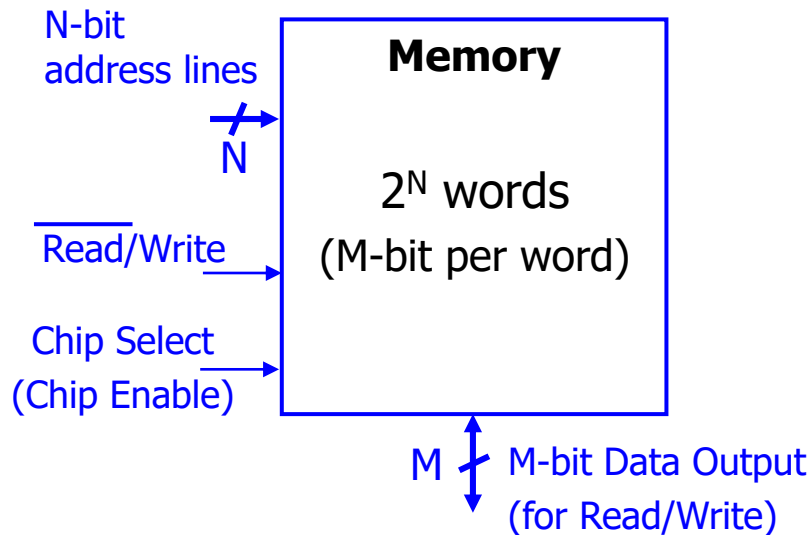
Concluding Remarks

- That's it! Folks
- I hope you have enjoyed (?) this class
- If you think that you successfully finished the course, you should have confidence in computers now!
- But, you just have laid the foundations for the further study of computer science and engineering

Backup Slides

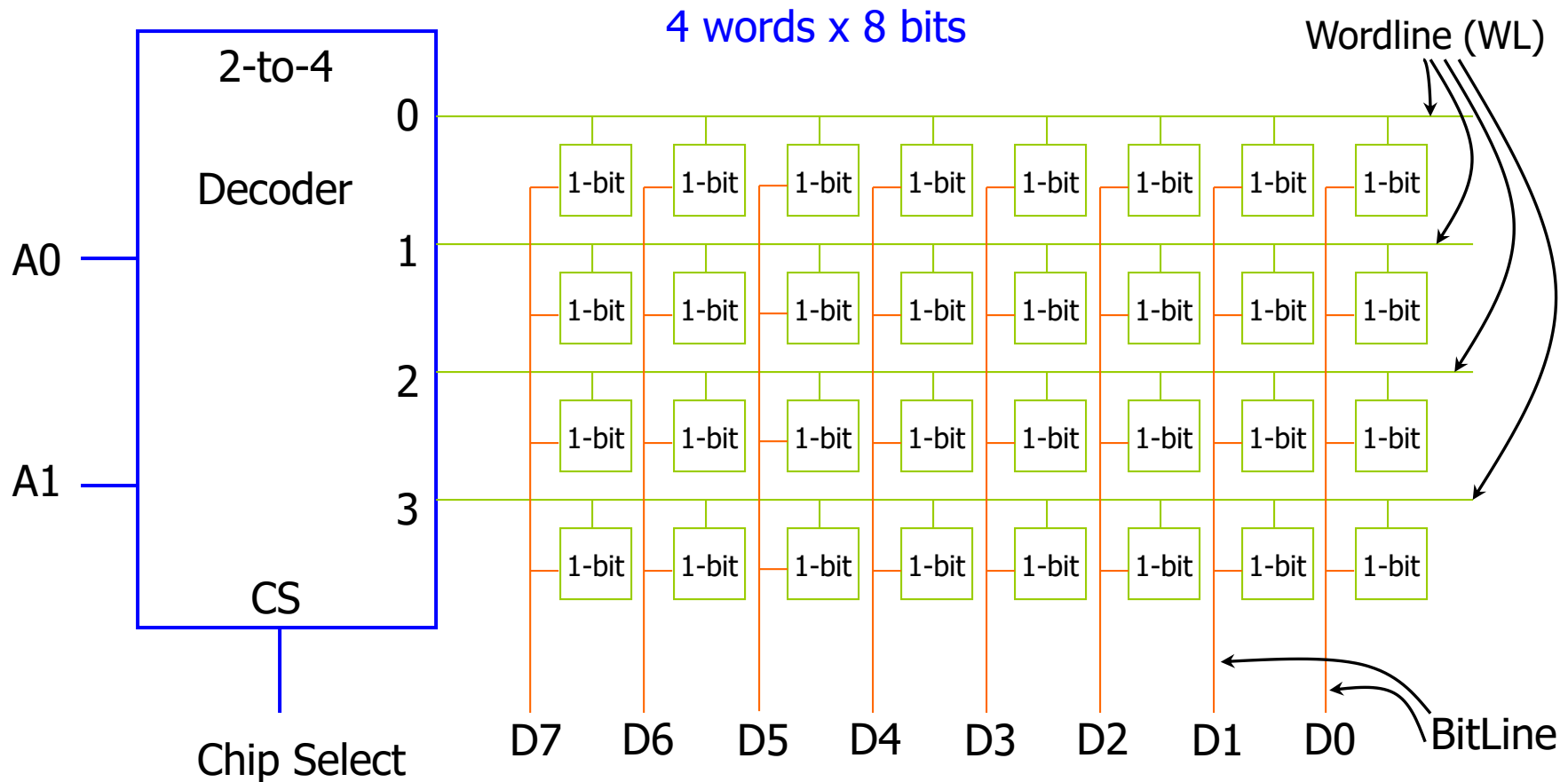
Block Diagram of Memory

- An M-bit data value can be read or written at each unique N-bit address

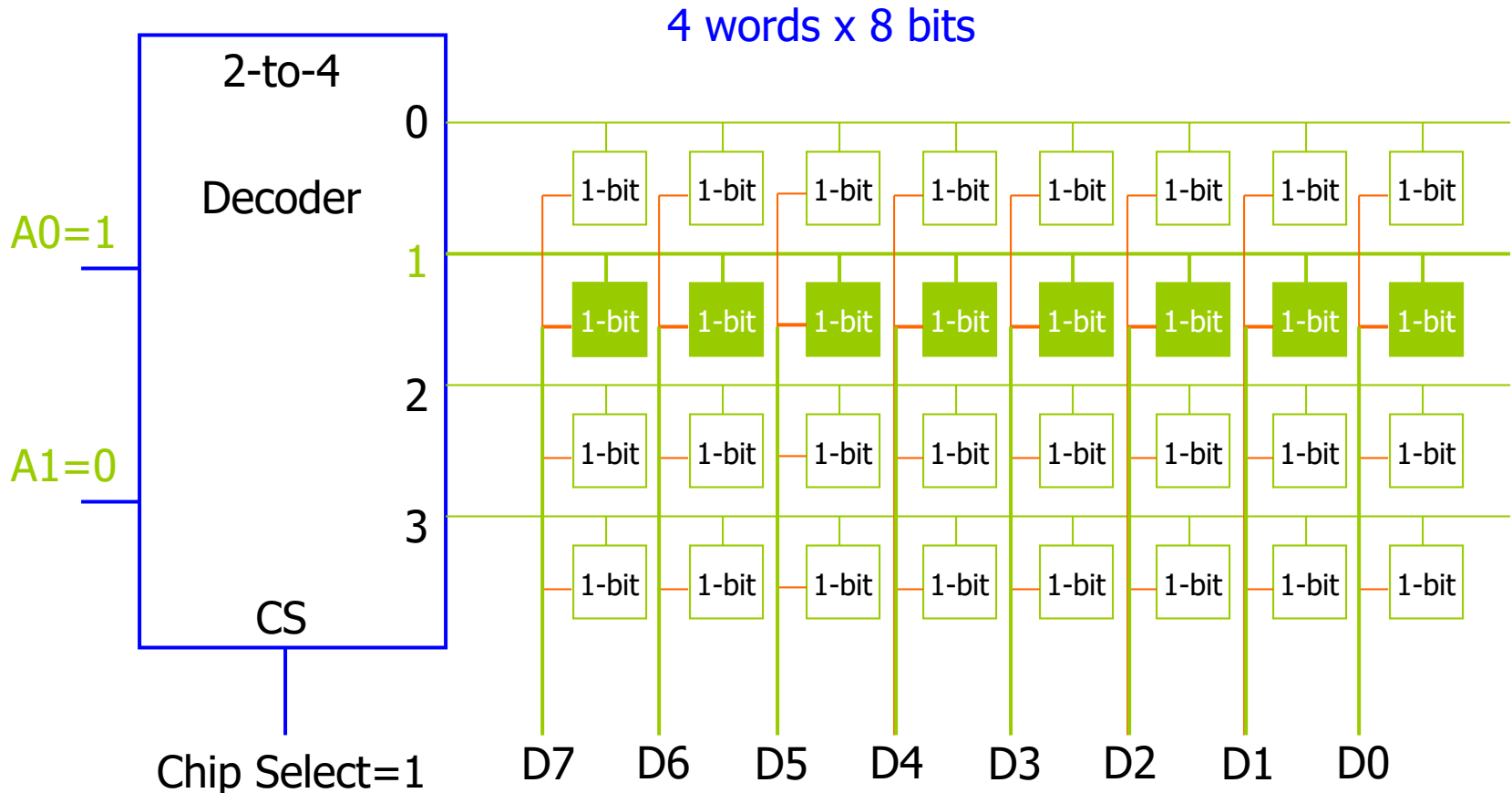


- Example: Byte-addressable 2MB memory
 - $M = 8$ (because of byte-addressability)
 - $N = 21$ (1 word = 8-bit)

Memory Organization Example



How to Address Memory



Access address = 0x1