week 0110
Memory

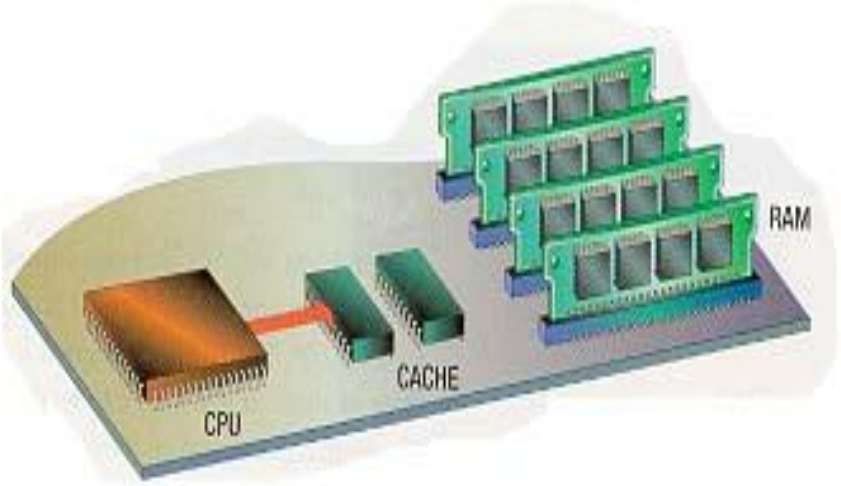
CCS1310

Computer Systems Architecture

Kostas Dimopoulos

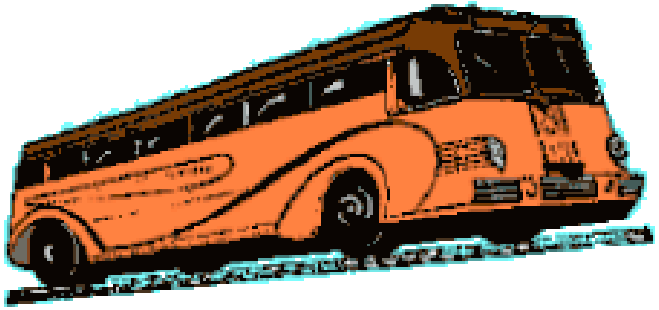
Lecture Outline

- What are buses?
- What is memory?
- How does memory operate?
- What are the different types of memory?
- What are all these types of memory and what are they good for?



Buses

- **Bus:** the physical connection that makes it possible to transfer data from one computer system location to another

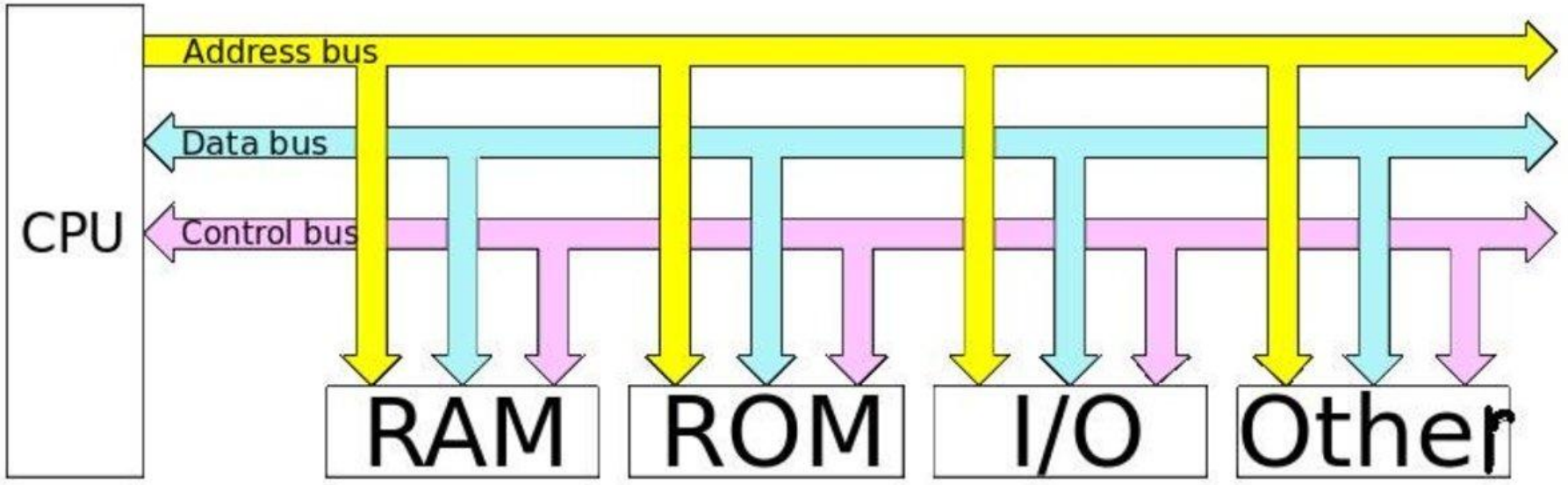


- Kinds of data transfer
  - peripherals - CPU
  - CPU - memory
  - between different points in the CPU

- **Bus width:** the number of bits (wires) that can be sent through the bus simultaneously
- **Bus speed**: how much data can move across the bus (MHz)

Busses

- Typically, a bus has 3 types of lines:
  - Address bus (defines who/where we talk to)
  - Data bus (defines what we talk about)
  - Control bus (defines the order of things, like start of transmission, end of transmission, acknowledge of



Kinds of Buses

- The bus used to connect CPU, memory, and a set of I/O module cards is called backplane, or system bus
- Buses may connect modules together in various ways
  - a bus may carry signals from a specific source to a specific destination (point-to-point)
    - point-to-point buses intended for connection to a plug-in device are often called **ports**
  - Alternatively a bus may be used to connect several points together (multipoint or multidrop or shared)

Expansion Bus Types

Type	Max Width	Max Speed	Transfer rate
Industry Standard Architecture (ISA)	16bits	8MHz	16MBps
Extended ISA (EISA)	32bits	8MHz	32MBps
Micro Channel Architecture (MCA)	32 bits	20MHz	80MBps
Video Electronics Standards Association (VESA)	32bits	50Hz	200MBps
Peripheral Component Interconnect (PCI)	64bits	66MHz	512MBps
PCI express (PCIe x16)			4GBps
Accelerated Graphics Port (AGP)	32bits	66MHz	2,133MBps
Universal Serial Bus (USB)	1bit	varies	40Gbps

Memory Basics

- What is memory?
  - Electronic circuitry that holds data and program instructions
  - Often called the primary storage, which creates the confusion with disk storage
    - **Memory** has direct link to the CPU
    - **Storage** refers to disk storage communicating with the CPU over the system bus
- Types of memory?
  - RAM
  - Cache
  - Virtual
  - ROM

Measuring Memory Size:
Kilobytes (KB): ~ 1 thousand bytes (2^{10})
Megabytes (MB): ~ 1 million bytes (2^{20})
Gigabytes (GB): ~ 1 billion bytes (2^{30})
Terabytes (TB): ~ 1 trillion bytes (2^{40})

Memory Categories

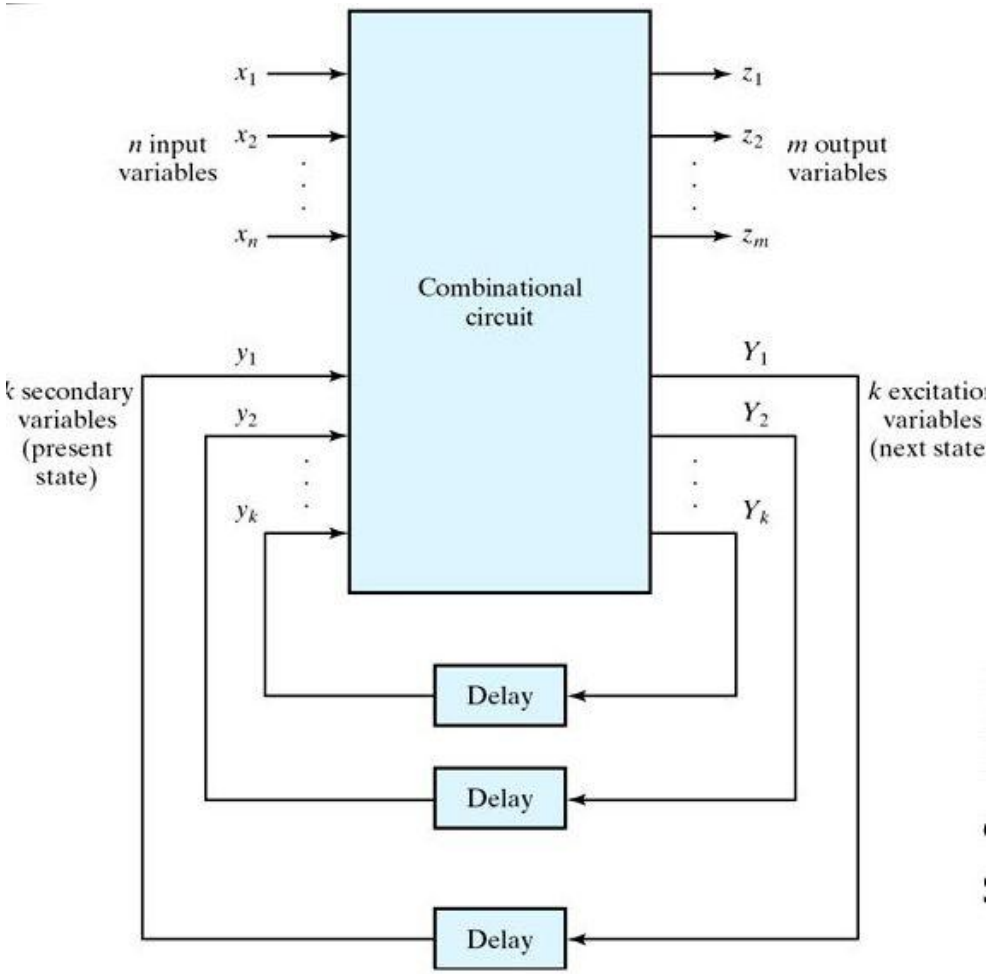
- Memory can be split into two main categories:
  - **Volatile memory**
    - loses any data as soon as the system is turned off; it requires constant power to remain viable (e.g. RAM)
  - **Nonvolatile memory**
    - does not lose its data when the system or device is turned off (e.g. ROM)

More Memory Categories

- Memory can also be categorized into:
  - **DRAM (Dynamic RAM)**
    - Less expensive, requires less electrical power, can be made smaller –with more bits of storage
    - Requires extra circuitry that **refreshes** memory
      - Memory controller reads the data and then rewrites it.
    - Example: RAM
    - Typical sizes: 4GB to up to 32GB
  - **SRAM (Static RAM)**
    - Faster to access than DRAM, does not require refreshing, but the complexity of each cell makes it prohibitively expensive for use as standard RAM
    - Example: Cache
    - Typical sizes: 1MB to 16MB

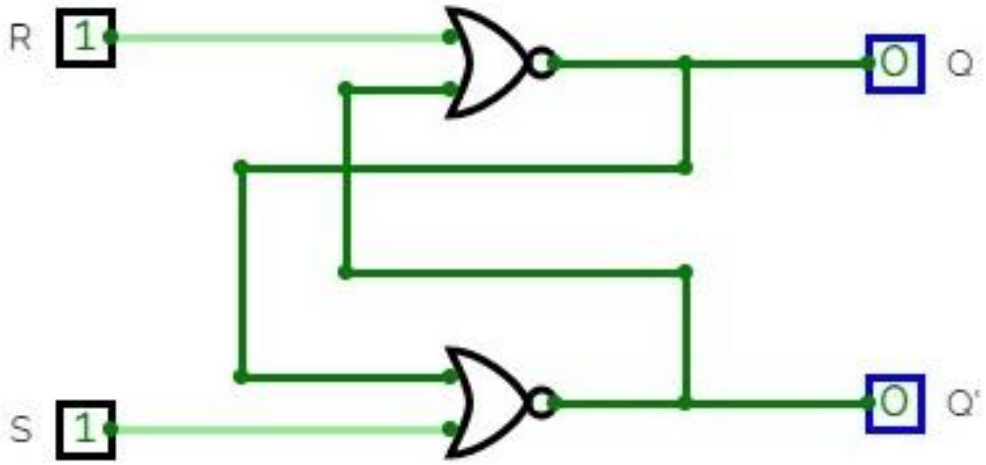
Sequential logic circuits

- Memory elements are made from sequential logic circuits:
  - Feedback: The output of the circuit depends not only from the inputs, but also from the previous outputs



The NOR S-R latch

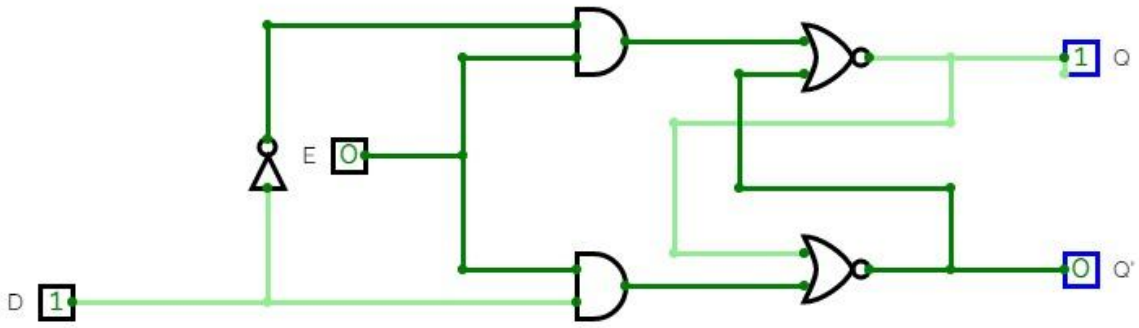
- The S-R latch is a circuit made from 2 NOR gates (or 2 NAND)
- Inputs are used to Set/Reset the output Q
- A second output is also the negated first output (Q')



Characteristic table				Excitation table			
S	R	Q_{next}	Action	Q	Q_{next}	S	R
0	0	Q	Hold state	0	0	0	X
0	1	0	Reset	0	1	1	0
1	0	1	Set	1	0	0	1
1	1	X	Not allowed	1	1	X	0

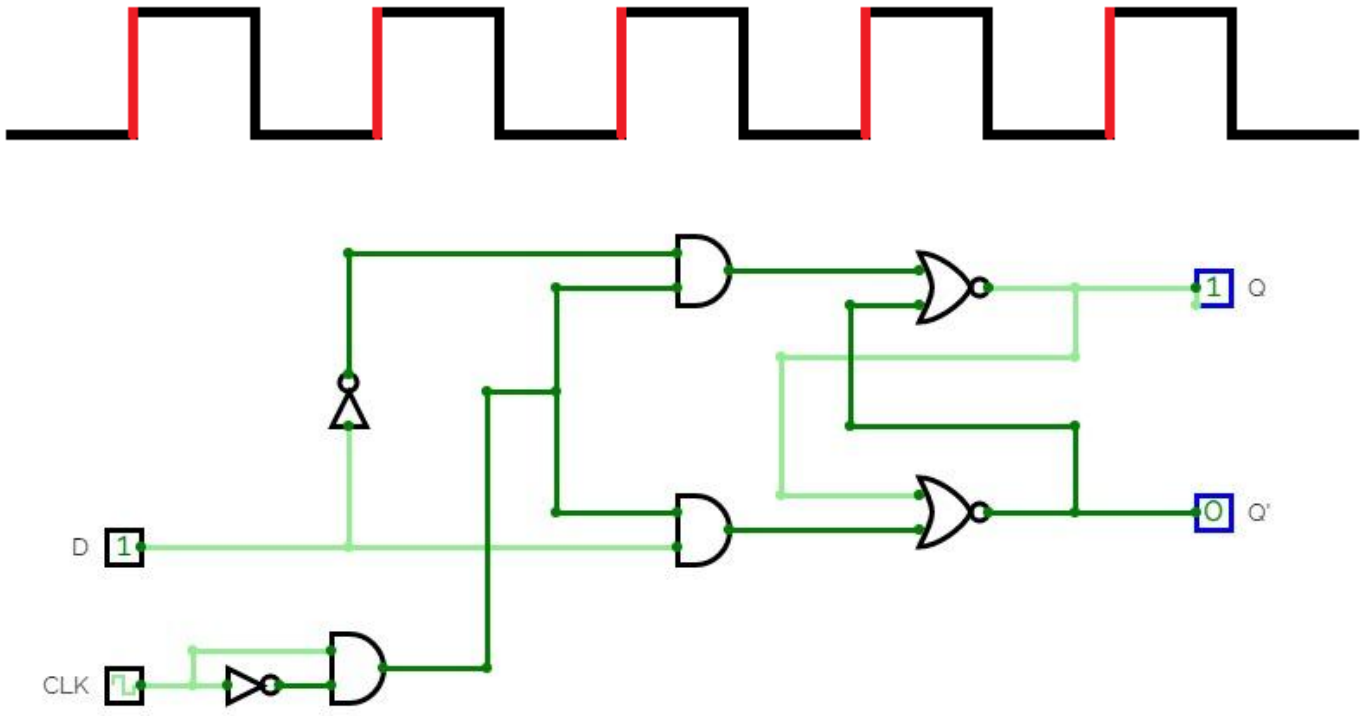
The D-Latch

- To avoid the invalid state of Set=Reset=1we add a negator between the Set and Reset, thus having a new input D
- Finally with 2 AND gates we create an Enable input that will enable the input D to be stored at Q when the E is true



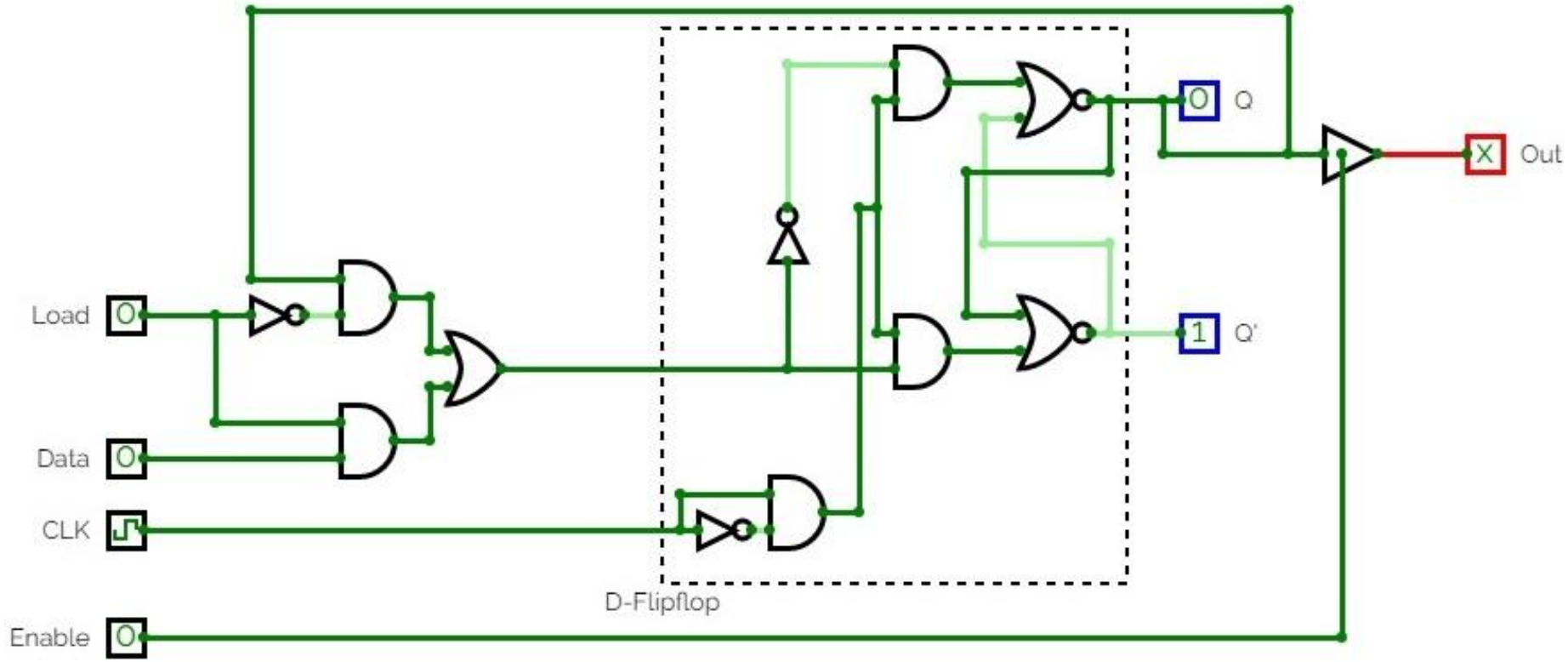
The D-flipflop

- The D-flipflop is an improvement that allows the enabling to occur on the rising edge of a clock



1 bit register

- With some extra circuit we can create a register.
- The extra circuit allows us to define when to write (Load) to the memory or read (Enable) from the memory (write to the bus)



A memory cell

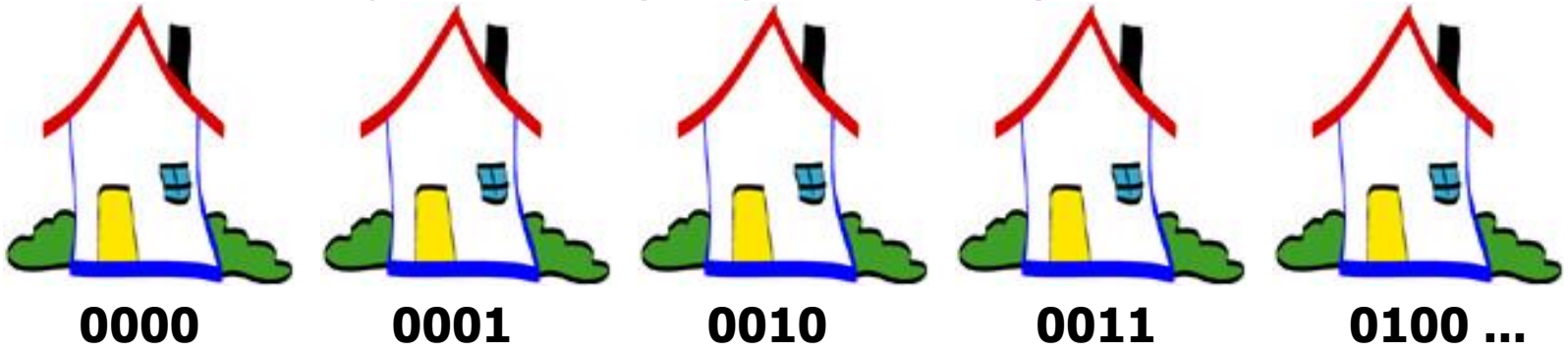
- As a D-flipflop can store 1 bit, 8 flipflops can store a byte.
- Thus 8 flipflops is 1 memory element
- A L2 cache of 8MBs how many D-flipflops?

Memory: RAM (Random Access Memory)

- Often called **main memory**: the place where CPU temporarily holds data during processing
- Properties:
  - Large grid of **digital circuits**, each represents **1 bit**
  - Transistors are used to change the value of the bit
  - Capacitors are grouped together into bytes
  - Each byte has a memory address
    - location of the byte in the memory grid
  - Circuits must have continuous power to maintain data integrity – **volatile!**

Memory Addressing

- Each memory location (**cell**) has a unique **address**



- We can write to, and read from, cells using these addresses
  - 32-bit addressing: can address $2^{32} \geq 4$ billion memory cells
  - 64-bit addressing: can address 2^{64} (huge)
- Basic machine language statement:
  - **OP CODE**: part of machine code (bits) specifying the operation (add, multiply...)
  - **OPERANDS**: part of machine code (bits) specifying address of data to operate on

Analogy: Spreadsheets

- Memory address is similar to WHAT in a spreadsheet?
- Unlike SS cell: holds far less!
  - 1 memory cell holds one **byte = 8 bits**
  - cannot address bits individually
- Comparing memory & spreadsheet cells
  - How many memory cells to hold one ASCII character?
  - How many memory cells required to hold a bit pattern that can represent a single pixel with 24-bit color?
  - What happens when you type something into a SS cell that already contains some data?
  - What happens if an Excel formula READS a data value from another cell? Are this cell's contents destroyed?

Memory Cells

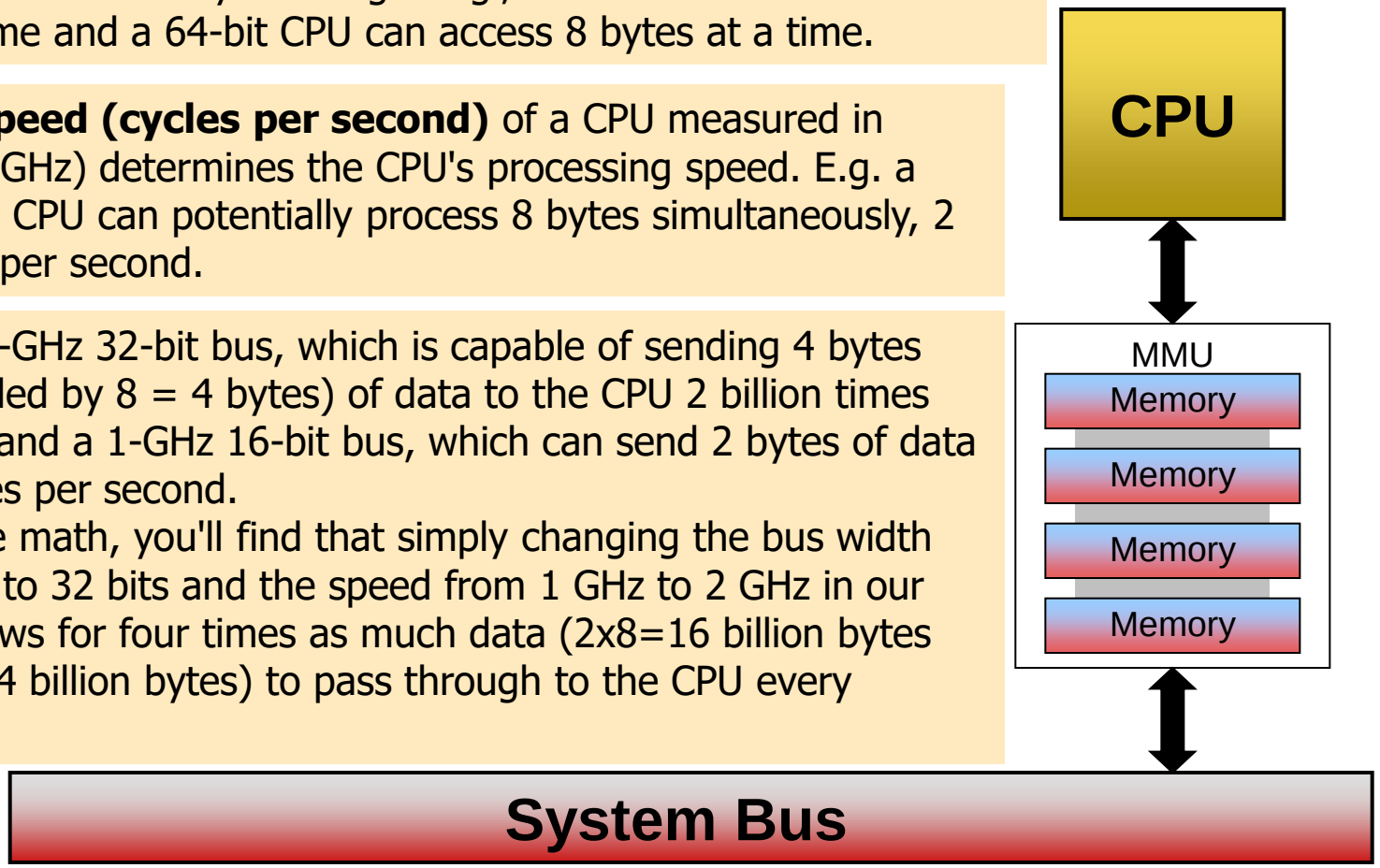
- So, each memory cell (location) can hold **8 bits**
- Those 8 bits, in turn, could represent, at any given time:
  - Part of an integer or real number
    - Integers: typically 32 bits
    - Real numbers: typically 64 bits
  - One alphanumeric character (1 character byte)
  - Small portion of a sound sample
  - One pixel value for 8-bit color
  - Only 1/3rd of a pixel's value for 24-bit color
  - Small part of an instruction

CPU-Memory Relation

The **data bus width** of a CPU tells us how many bytes of data it can transfer from the memory in one go. E.g., a 16-bit bus can access 2 bytes at a time and a 64-bit CPU can access 8 bytes at a time.

The **clock speed (cycles per second)** of a CPU measured in **Gigahertz** (GHz) determines the CPU's processing speed. E.g. a 64-bit 2-GHz CPU can potentially process 8 bytes simultaneously, 2 billion times per second.

Consider a 2-GHz 32-bit bus, which is capable of sending 4 bytes (32 bits divided by 8 = 4 bytes) of data to the CPU 2 billion times per second, and a 1-GHz 16-bit bus, which can send 2 bytes of data 1 billion times per second.
If you do the math, you'll find that simply changing the bus width from 16 bits to 32 bits and the speed from 1 GHz to 2 GHz in our example allows for four times as much data (2x8=16 billion bytes versus 1x4=4 billion bytes) to pass through to the CPU every second.

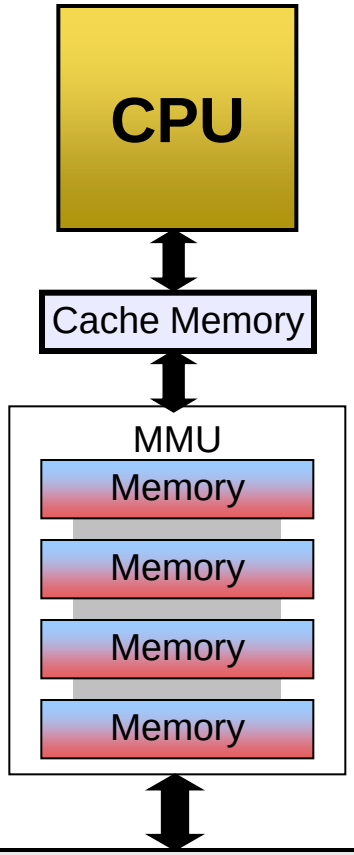


Cache Memory

- In the past main memory technology was too slow for chips!
- Solution:
  - Put smaller and faster memory **between CPU and main memory** to store frequently used information
  - This was the creation of the cache memory
- Now days speeds are comparable, but a system can still benefit from using cache to store frequently used information between the CPU cores (shared cache).

Cache Memory

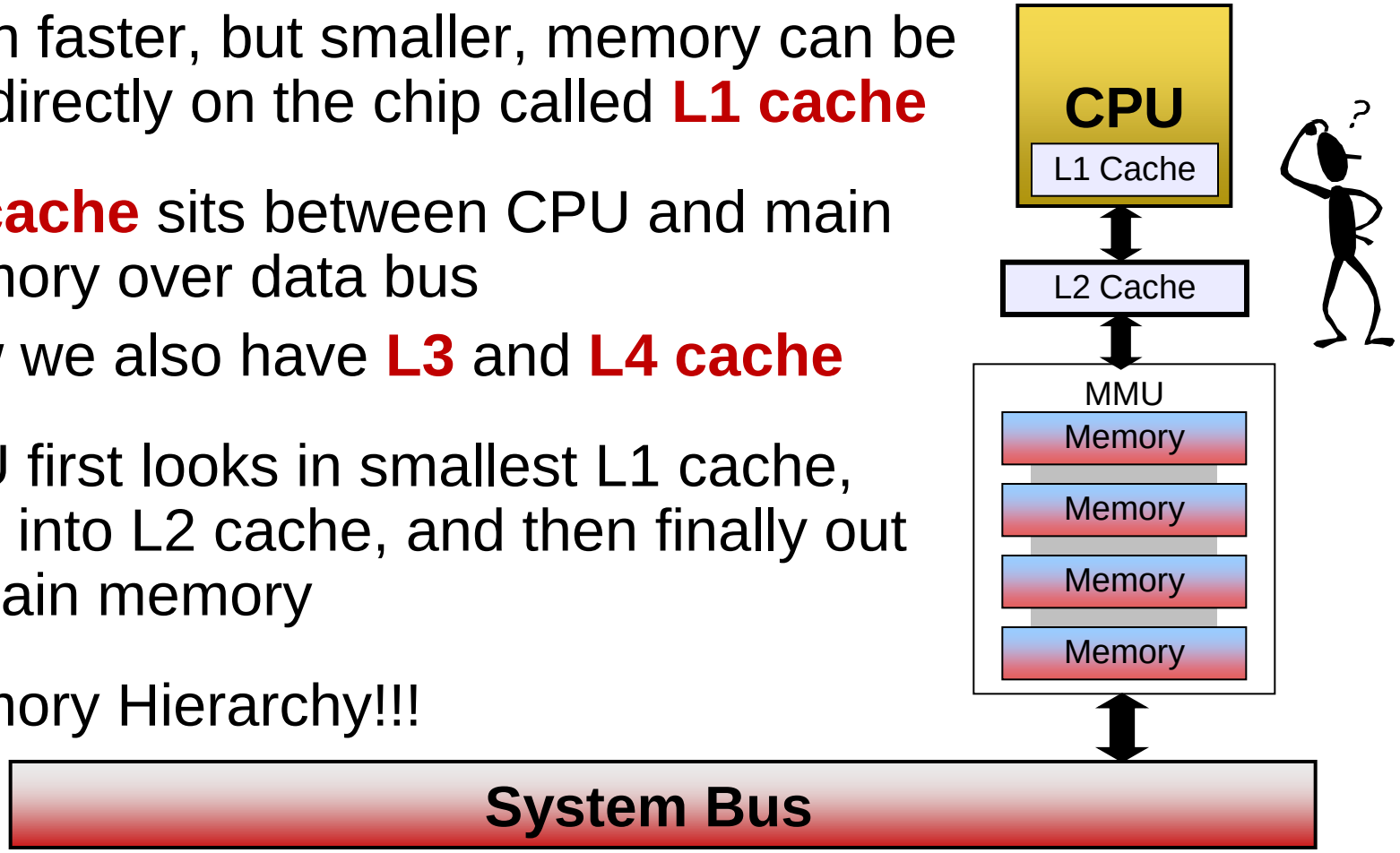
- CPU looks in cache before going to main memory
- Much faster when data is found:
  cache hit
- Much slower if data is not found
  cache miss
- In practice, due to **locality of reference principle**, entire memory system operates close to speed of cache memory!



Locality of reference: what has been accessed previously is more likely to be accessed again, and what is close to what has been accessed previously is more likely to be accessed

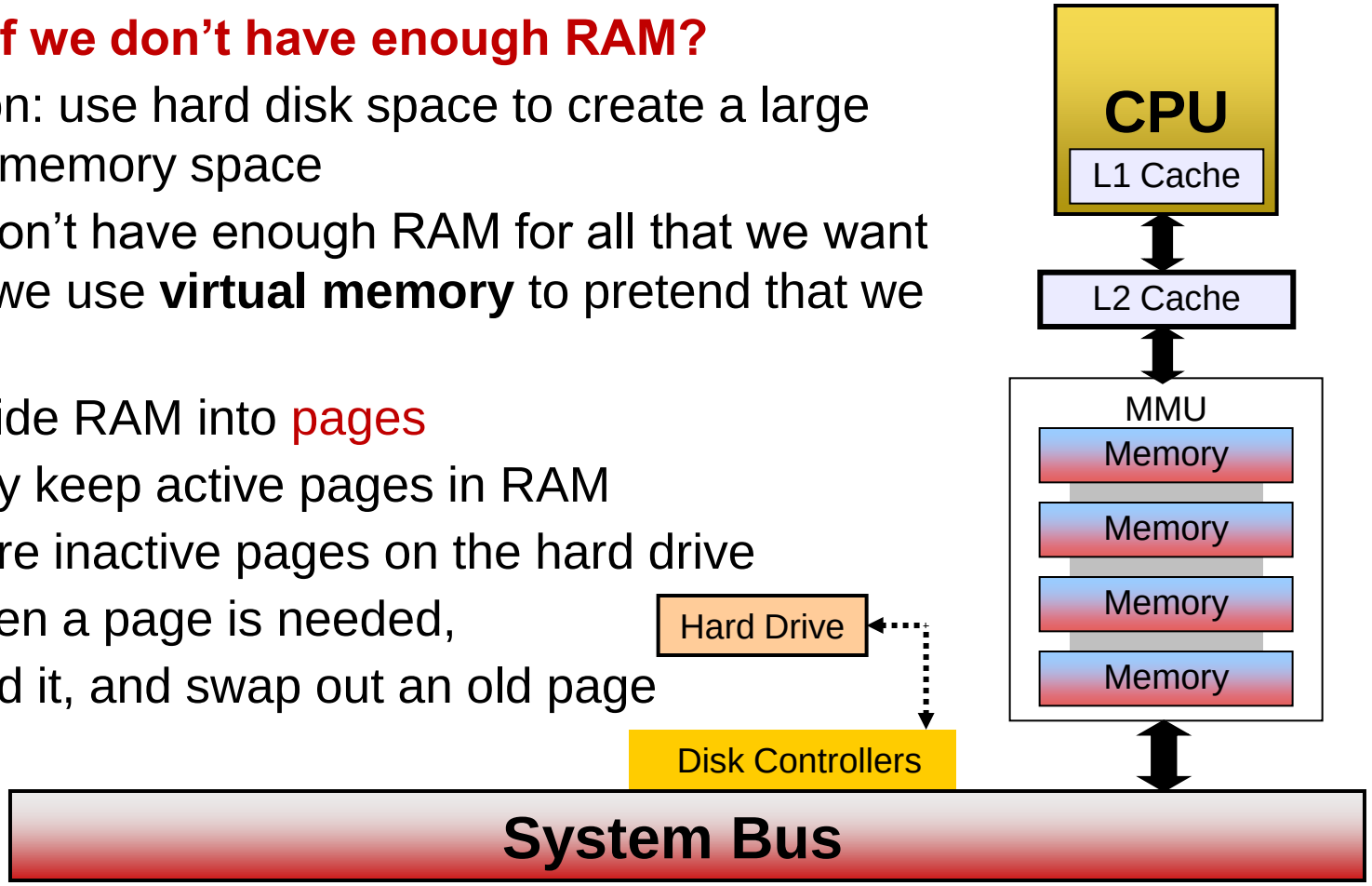
Cache Memory

- Even faster, but smaller, memory can be put directly on the chip called **L1 cache**
- **L2 cache** sits between CPU and main memory over data bus
- Now we also have **L3** and **L4 cache**
- CPU first looks in smallest L1 cache, then into L2 cache, and then finally out to main memory
- Memory Hierarchy!!!



Virtual Memory

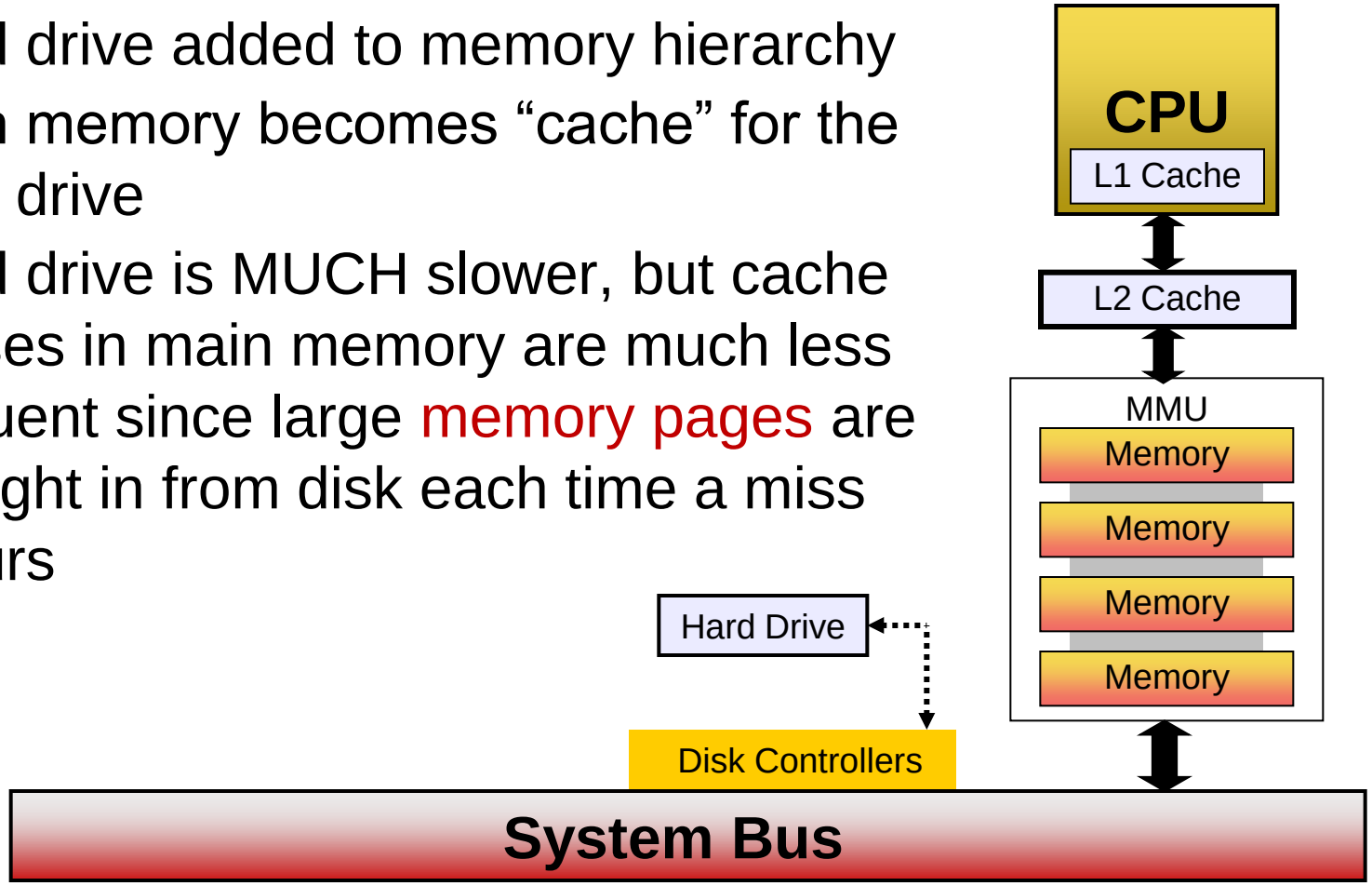
- **What if we don't have enough RAM?**
- Solution: use hard disk space to create a large virtual memory space
- If we don't have enough RAM for all that we want to do, we use **virtual memory** to pretend that we do
  - divide RAM into pages
  - only keep active pages in RAM
  - store inactive pages on the hard drive
  - when a page is needed, load it, and swap out an old page



Thrashing

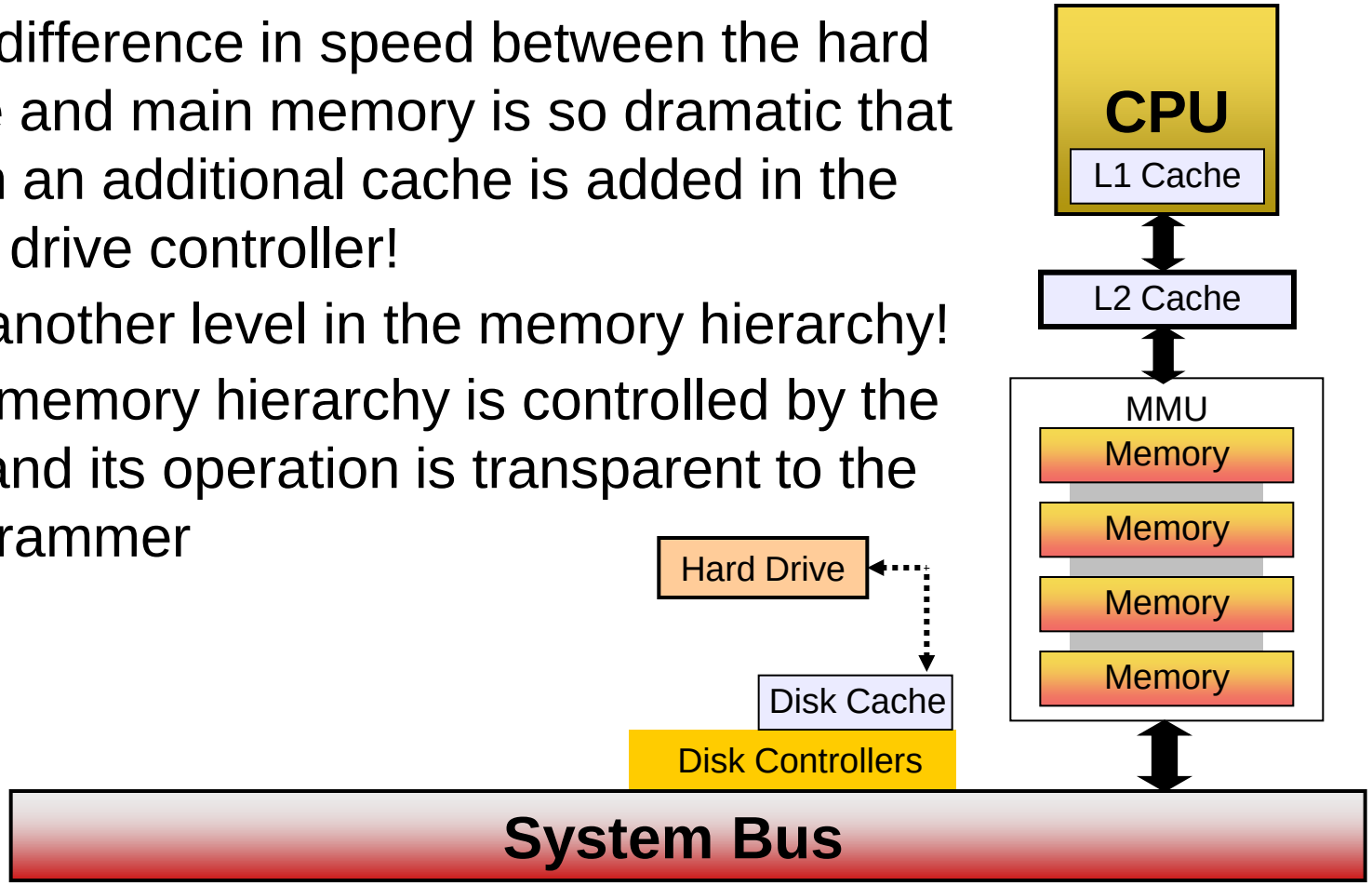
Virtual Memory

- Hard drive added to memory hierarchy
- Main memory becomes “cache” for the hard drive
- Hard drive is MUCH slower, but cache misses in main memory are much less frequent since large memory pages are brought in from disk each time a miss occurs



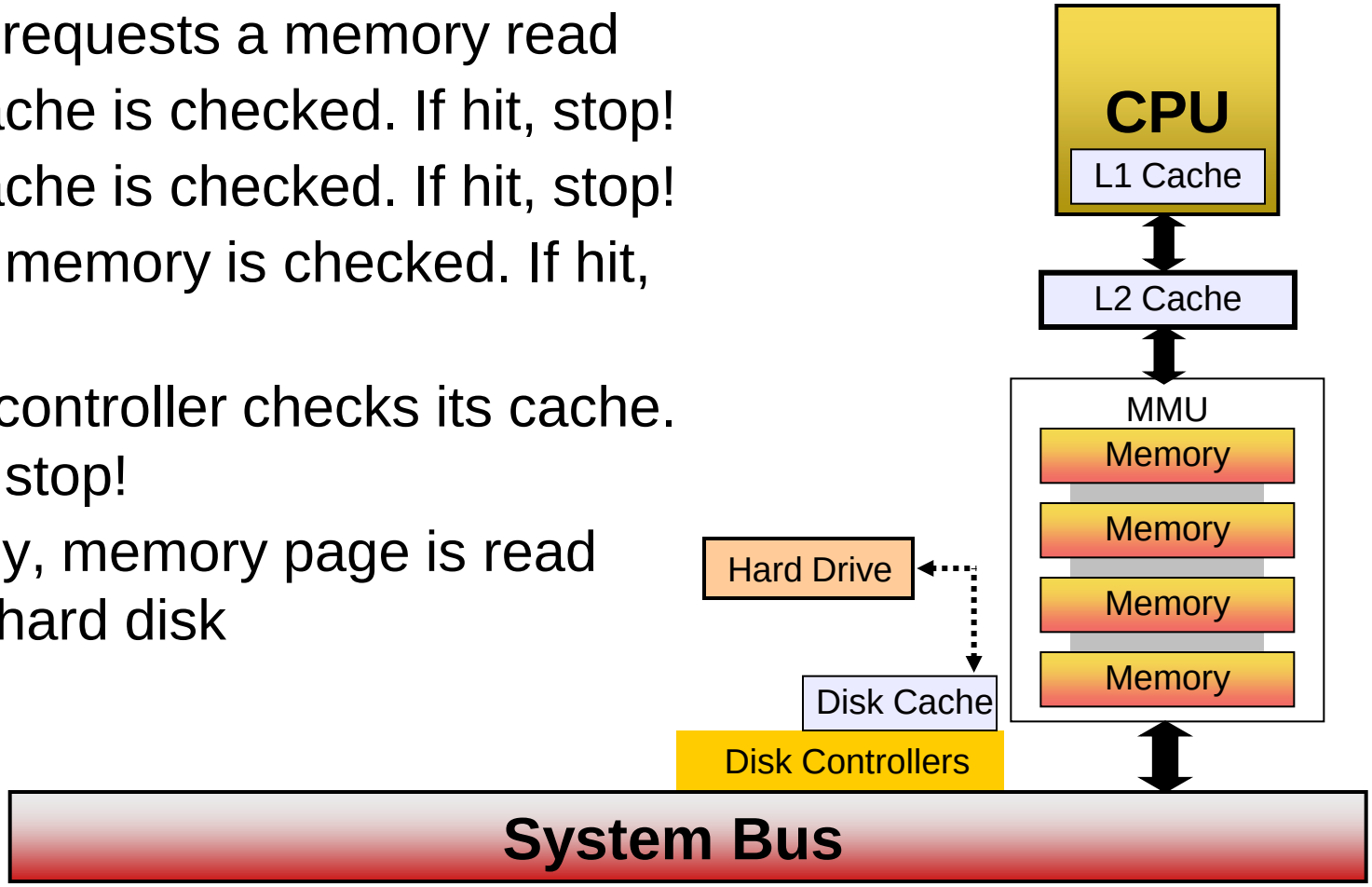
Adding to the Memory Hierarchy…

- The difference in speed between the hard drive and main memory is so dramatic that often an additional cache is added in the hard drive controller!
- Yet another level in the memory hierarchy!
- The memory hierarchy is controlled by the OS and its operation is transparent to the programmer

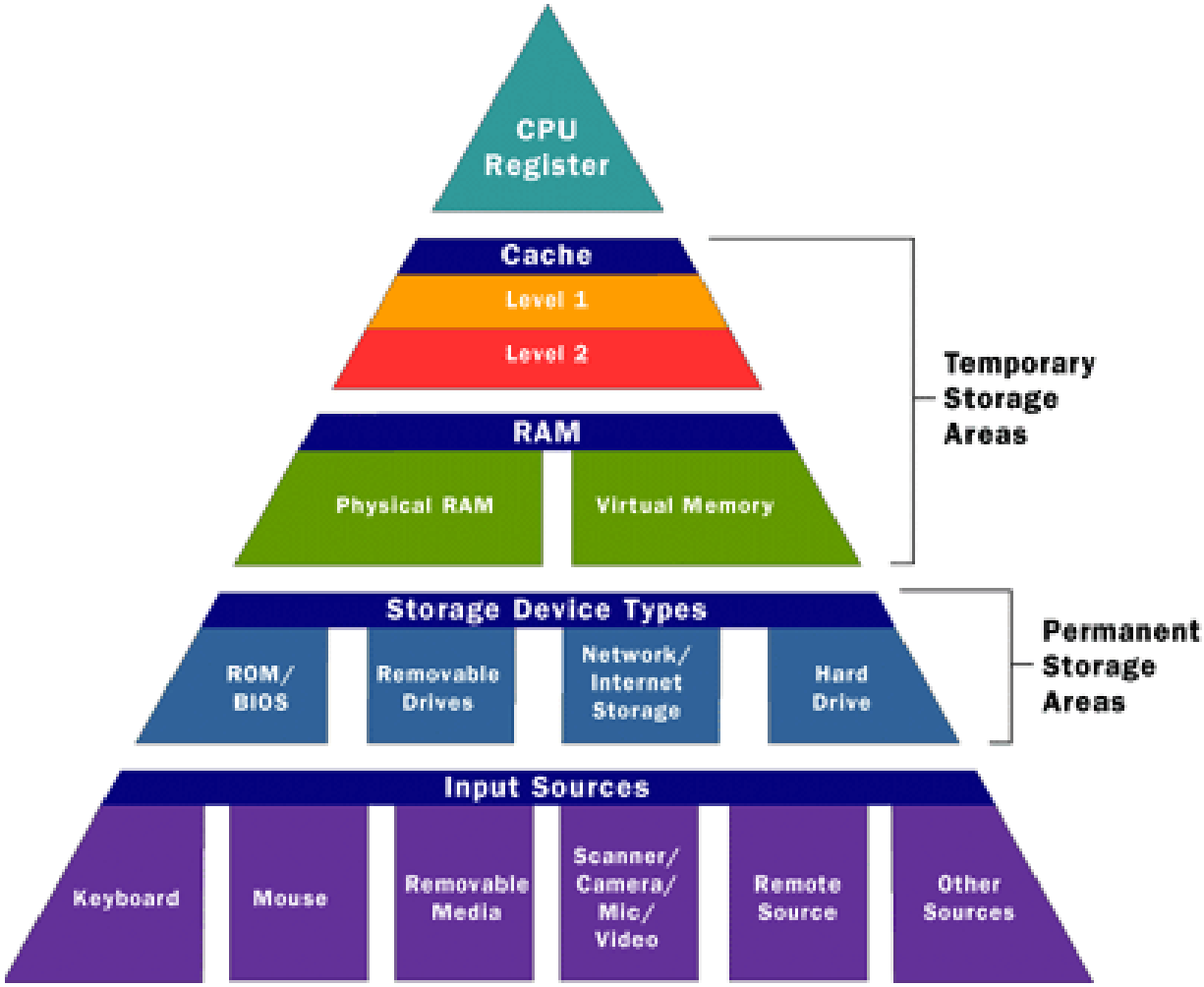


Data Access Procedure within the Complete Memory Hierarchy

- CPU requests a memory read
- L1 cache is checked. If hit, stop!
- L2 cache is checked. If hit, stop!
- Main memory is checked. If hit, stop!
- Disk controller checks its cache. If hit, stop!
- Finally, memory page is read from hard disk



Memory Hierarchy



Memory: ROM (Read Only Memory)

- An integrated circuit programmed with specific data when manufactured
- **Hard-wired** memory containing instructions for:
  - starting the computer
  - running system diagnostics
  - controlling low-level I/O (BIOS)
- Typically small

The way ROM chips work requires the programming of perfect and complete data when the chip is created. You cannot reprogram or rewrite a ROM chip. If it is incorrect or data needs to be updated you have to throw it away and start over. Creating ROM chips from scratch is time-consuming and very expensive in small quantities.

Types of ROM

- There are five basic types of ROM each of which has unique characteristics but with two things in common:
  - Data is nonvolatile (not lost when power is removed)
  - Data is unchangeable (or requires special operation to change)

PROM (programmable read-only memory): can be bought inexpensively and coded by a programmer. Can only be programmed once.

EPROM (Erasable Programmable Read-Only Memory): chips can be rewritten many times with a special toolkit. To rewrite, it has to be removed and erased.

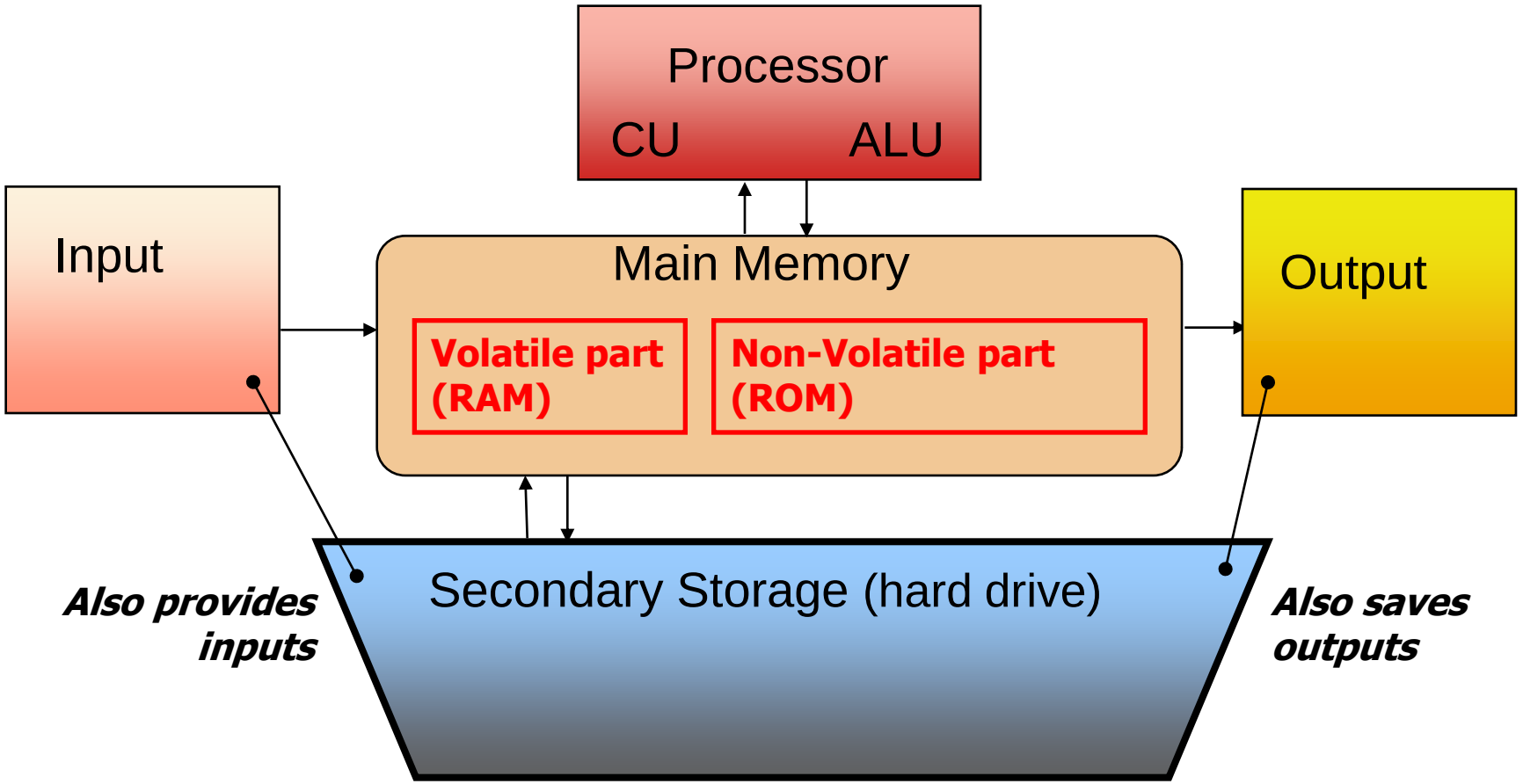
EEPROM (Electrically Erasable Programmable Read-Only Memory): The chip does not have to removed to be rewritten. The entire chip does not have to be completely erased to change a specific portion of it. Changing the contents does not require additional dedicated equipment. Disadvantage: slow to be rewritten (changes occur 1 byte at a time).

FLASH MEMORY: Type of EEPROM which works much faster since it writes data in chunks usually 512 bytes in size.

Memory: BIOS

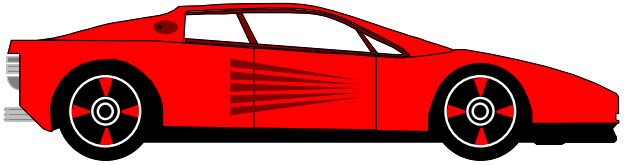
- One of the most common uses of Flash memory is for the **Basic Input/Output System** of the computer
- What does the BIOS do?
  - Provides instructions to load the operating system
  - Performs a POST (power-on-self-test): makes sure that all different hardware components are working properly
  - Activates other BIOS chips (graphics card)
  - Provides a set of low level routines that the OS uses to interface with different hardware devices (keyboard, screen, serial and parallel ports) especially during the booting

Computer System Overview



Speed Issues

- **L1 cache**
  - Memory is accessed at full microprocessor speed
- **L2 cache**
  - Memory access of type SRAM
- **Main memory**
  - Memory access of type DRAM
  - 2GB to 8GB in size
- **Hard Disk**
  - Mechanical and slow
  - 500MB to 8TB in size



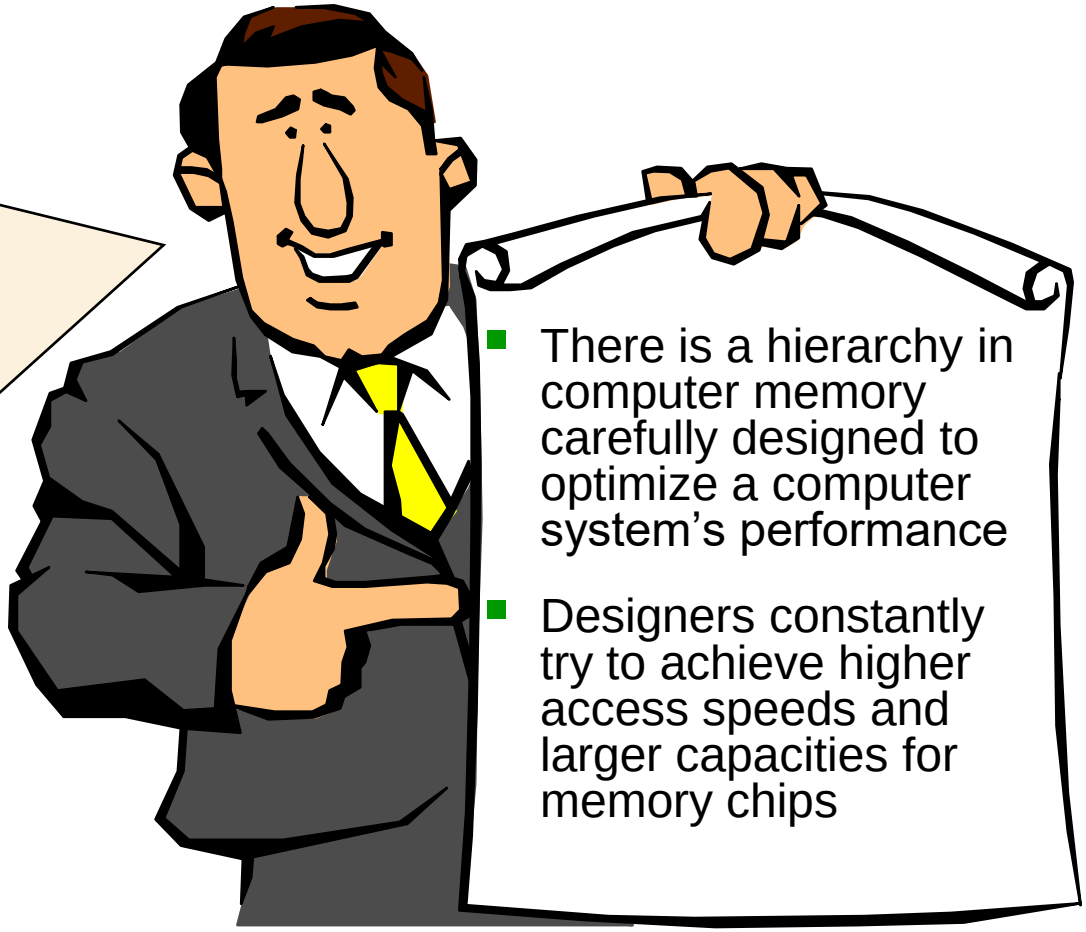
Summary

Today we went over the basics about the memory of a computer system.

As you saw, there are many different types of memory each of which has unique characteristics and functionality.

As future computer scientists you must know as many details as possible.

The fact and the trend are…



- There is a hierarchy in computer memory carefully designed to optimize a computer system's performance
- Designers constantly try to achieve higher access speeds and larger capacities for memory chips