Inside a Computer:

Mainboard / Motherboard:
- Provides physical connectivity for all of the devices, included the "system bus" and all peripheral buses. If the CPU is the brain, this is the circulatory system.

CPU:
- The "Brain" of the computer, this is where all calculations are done. Registers the "CU" and "ALU", "L1".

Main Memory:
- This is where code and data are stored. When the computer is shutdown, this is lost

Secondary Storage:
- Hard Drive or Solid State, this is the permanent storage system

Tertiary Storage:
- CD-Rom, DVD, Blu-Ray or even Tape. This is the OFFLINE storage system.

Video Card / GPU:
Stores information to display on screen, can do complex calculations related to "Decimal" Numbers.

Power Supply:
Provides clean conversion from line voltage to 12V and 5V needed inside the computer.

What's Common Between Them?

All Computers Have:
- ATLEAST 1 CPU
- Main Memory, where code and data is stored temporarily
- Secondary Storage, where info. is stored permanently

Most Computers Have:
- A Video grapic controller, where images can be rendered for Display
- Network Interface, for communication
- Peripheral Interface like USB, Thunder Bolt, Firewire, SCSI etc.

- All components meet the requirements to bring data into computer (Input), process data, and can produce Output

- The components either do 1 or 2 things in regards to "Input", "Processing", and "Output"

- Most Devices are either I/O Devices, or Processing

Communications Between Devices

- Internal Communications in a machine is done via "bus"

Bus:
- Physical pathway for communication between two or more devices. Can move from one component to the other, or vice versa.
- Bi-Directional Communication Pathway
- Physically exsist in System: Microscopic Copper Lines on Mainboard

- The system bus is the main pathway between the CPU and main memory, but also carries data to and from Input/Output (I/O) devices.
- I/O devices have acess to system bus for communication to memory and CPU
- System Bus is "Primarily" used priority for CPU and Main Memory communications, and has to be done very quickly

The CPU : Central Processing Unit

- The "Brain" of the Computer

- Where all Data Processing, and commands run inside of a computer

- Single Piece of Silicon in the form of a chip

- Only location in computer system where code can be run
  - Only runs "machine Language" Code.
- Machine Language code = Machines ; Higher-Level Code = Humans

- Operates on a "fetch - decode - execute" cycle
  - Runs this cycle very quickly

- Each type of CPU has its own "instructions" which it understands
  - CPU has a "language" for itself

- Each CPU has a small amount of memory, called "registers" which it uses to perform operations and store results.
  - Registers : Store information temporarily to do immediate processing
  - Only a few, have to be used sparingly when writing Machine Code

- CPU may also have "cache" memory
  - As programmers, don't really have lots of capability to work with "cache" in regards to whats inside the CPU.

Machine - Language

- Computers only understand very basic commands:
  Move, Add, Subtract, Multiply, Compare, Jump, etc.

- Won't see higher-level commands as you would in a high-level language
  - No function calls; not alot of commands besides basic math operations.

- There are alot of machine-language instructions, but are limited and can't be added to
- Instructions are designed physically into the chip
- Programmed by manufacturer, and physically written in the chip on silicone.
- CANNOT EXPAND. No capability to add instructions
- HAVE CAPABILITY to build higher-level languages and convert them to the machine-language instructions

- The Designer of the CPU puts the capability to perform these operations physically on the CPU Silicone Chip.

Instruction Set

- The designers of the CPU create a set of instructions that the CPU can perform AKA "Instruction Set".

- Small set of Instructions, about 100 instructions or so
  - Each can be represented by a numeric value
  - When CPU recieves an instruction it performs that task

- One instruction might have multiple codes associated with it depending on variables associated with that instruction
  - Fundementally, small set of instructions with limited operations

- We need to work with high-level language code to work with machine language code, so that it can run on CPU.

Fetch-Execute Cycle or Fetch-Execute-Decode Cycle

- Moves one instruction from main memory, into a register in CPU. This register is called the "Instruction Register" (FETCH)

- "Decodes" that instruction; if necessary moves in any additional parameters so it copies those parameters from main memory (DECODE)

- Once everything is loaded in the CPU, the CPU can actually "execute" that instruction, and do the task that is asked. (EXECUTE)
  Example: Add 2 numbers
  FETCH: Memory Location, to register
  DECODE: Addition Instruction
  EXECUTE: Add the numbers together

- Happens very quickly and needs to happen millions of times per second in the CPU
- Each operation takes $\approx 10^{-9}$ nanoseconds to perform operation

- The process repeats with the next instruction in the sequence

Memory

- The instructions and all the data need to come from somewhere

- In order for code to be executed, it has to be in a register built into the CPU (Instruction Register)

- Can't store everything in registers because that memory is in Bytes. Doesn't store alot of code
  - Only want to store ONE instruction directly in CPU
    - Each additional Byte of memory cost alot of money to add to each CPU
    - Manufactures want to keep cost as low as possible

- Instead we create a memory heirarchy
  - Each layer of heirarchy adds alittle bit more space
  - Each layer then cost a little less
  - BUT runs a little slower

The Memory Heirarchy

- At the top of Heirarchy: Registers
  - Built Directly into the CPU
  - Only a few of them
  - Size Measured in Bytes (Use SPARINGLY)
  - Time to Access each Byte: Nanoseconds
  - ONLY Place instructions can be executed

- Second Layer: Cache (L1) & Cache (L2)
  - Measured in Megabytes (L2) or Kilobytes (L1)
  - Nanosecond Access Time
  - Slower than register, can help SPEED up Processing
  - Processor Designers take care of Cache, not terribly useful for programmers

- Third Level: RAM (DDR / Main Memory)
  - Measured in gigabytes
  - A lot slower access time: Over 10nanoseconds
  - Factor of 10 slower than registers
  - Only place we can store code and data
  - RAM is Volatile Memory
  - Once computer is shut down, contents of RAM are erased
  - RAM loads code to registers one-by-one

Secondary Storage
- Measured in Terabytes
- Much slower access time: Milliseconds
- A million times slower than RAM
- Only permanent storage device inside the system
- Usually in form of a hard drive or SSD

Tertiary Storage: Tape Drives, Flash Drives Measured by your immagination

RAM : Random Access Memory

- Can be accessed in the same amount of time or Random Time
  - Can access any Byte of main memory at the same amount of time

- Areas of memory are broken down to bytes, with each byte being able to be accessed independently of the others
  - Accessed in terms of the Data types which might have particular size
  - Example: C++ int of size 4bytes equivalent to "D-Word"

- When computer is turned off, everything in RAM is lost
  - Once electrical power is shutdown, RAM no longer contains info.

- When running a program, all the machine language instructions are brought into RAM and one-by-one pulled into the CPU by the Fetch-Execute Cycle, into "Instruction Register," then Processed

- Absolutely Critical Portion of the System

- Physical Device where we can temporarily store information.

Secondary Storage

Broken Down into 2 types: Hard Disk Drives & Solid State Drives

- Hard Disk Drives (HDD) also known as "spinning" drives
  - Contain magnetic material discs which rotate together at a constant velocity
  - Measured in RPM
  - Published by Manufacture
  - Contains "Read Heads", which move to different radii on the disk
  - Allow the system to access any position via it's three-dimensional polar coordinates
- Defines a particular rotation on the disk whether we want to read or write
- Accessing first innermost radius then the outermost radius takes significant amount of time compared to moving to an adjacent radius
- Each Radius is a "Track" Each Track holds info
- The time to access each track depends on RPM, moving adjacent is quicker than moving from outermost/innermost due to RPM speed, needing full rotations to go to
- Benefit: Sizes are significant (1TB, 4TB, etc.)

- Solid State Disk Drive
  - Contains a number of chips like USB Flash drives
  - Don't have anything that moves
  - Data is stored electronically in these chips
  - Chip store data regardless of power
  - Due to cost, these drives are smaller than HDDs but perform faster

Networking

- The world is now globally connected

- Data can come from anywhere, as long as the data and we are connected to the same network

- Networks are connected via the Internet
  - Makes it so we can access data from anywhere
  - Important to understand the flow of data and networks

Physical Connections

- A lot of ways computers can be connected physically
  - Most popular: Copper, Fiber, Wireless

Copper:
- Standard ethernet cable or a Metallic Connection
- Usually inside cables are Copper, hence Name
- Usually 8 wires / 4 pairs of cables twisted in a Unshilled-Twisted Pair (UTP)

Fiber:
- Transmits data via light through glass
- Lot less Attenuation, lose lot less data
- Goes further Distances, and are faster
- Cost a lot more

Wireless:
- Virtually Everywhere
- Wifi is one, but there are many (Microwave, Radio etc.)

Protocols
- How do we know when we start sending data?
- How do we know who its from?
- How do we know where its going to?
- Idea of Protocol is important to physical connections: It's a language how computers, MUST BE WELL-DEFINED

Types of physical connections
- Ethernet: Defines Protocol & Type of Connection exp. 100 base T
- Wifi (802.11): Defined by IEEE many sub-local groups
- ATM: Asynchronous Transfer Mode, used in Back-Bone Networks

Packets

- Before, phones were connected via long cables, now they are connected via Packets

- Packets: Small amount of info. (1000 B or 1500 B), sent from One program running on machine to another program
- Can't be sent directly! Needs Protocols, which add info to the packet

- We send it by encapsulating packet and give it to lower level protocols. Creates Hierarchy

- Data is sent this way to know which destination is supposed to reciene this data
- Exp. If we have a website open in Chrome & IE, its important for computer to route the response appropriately
- This done via Protocols
- Important to know the path back

Layers we commonly use

Application:
- HTTP: Hyper Text Transport Protocol, application level protocol used to send data to/from websites
- SMTP: Simple Mail Transport Protocol used to pass mail b/w mail servers
- IMAP: Internet Access Message Protocol, used to "obtain" mail

Logically Protocols (Networks)
- Broken into 2 layers: Connection Oriented Vs. Connection less
- We need to "simulate a connection" = Connection Oriented
- If we don't need guarantee of delivery: Connection less
- Connection Oriented = UDP
- Connection less = TCP
  - Exp. Watching a video via HTTP over TCP
- Need to know how to route networks GLOBALLY:
  - Each Machine around the world is REQUIRED to Have an IP
  - IP: Internet Protocol
  - Must be Unique for every computer globally

- Your IP Address is Unique to your computer, if info needs to be sent, can be done so by sending data to IP Address

- IP Addresses are grouped into larger networks

Physical
- Often adds a header and footer (to indicate the end)
- Mostly concerned with local addressing and how to deal with eccentricities of their physical medium
- Example: Ethernet
- We move down protocol, caulladd a HTTP Header, TCP Header, IP Header and a physical header like Ethernet

CS Bridge Module 1 Fundamentals of System Hardware

What is a Computer?

- Electro-Mechanical Device which takes input, does processing, and produces Output. (Basic Form)

Types of Computers:

- Mainframe: Very Large Computer which used to act as a "central point" for all computing done on a Campus Exp. A university may have 1 Mainframe for each campus, where all processing is done.

- Server: Exsist in a Datacenter or Dedicated Room, lot of capability, lot of RAM, lot of CPU. Not used by most people, serves usually 1 purpose

- Desktop: Would be used by an "End User" in 1 physical location.

- Laptop: Carry it around, use it as an regular Computer

- Tablet: Same as Laptop, without keyboard

- Portable Phone ("Smart Phone"): By Definition is a compute