Part 1: Computer and Digital systems

* Computer systems
  + A picture containing graphical user interface

    Description automatically generated
  + Physical devices are made out of electronic components.
  + Micro architecture is a specific part of the physical device that allows you to perform specific tasks/electronic components put together to make the physical device
  + Machine language: Binary code (1s and 0s) the only language that the hardware will understand
  + Operating system the system that allows you to have an interface with your hardware (Mac OS, Linux, Windows); operate into specific or various machines, allows you to use the hardware. Recognizes the architectures and languages
  + System software: compiler, device driver, and commend interpreter; analyzing and translating code into something readable for the operating system to give to the hardware.
  + Application software: banking system, Microsoft office, web browser; particular computer software designed to perform a specific task sending inputs and outputs to the system software
* Hardware of computer systems
  + CPU (central process unit)
    - Made of ALU (arithmetic logic unit)
      * Uses registers to store the data needed for the instructions
      * Responsible for performing arithmetic operations (any mathematic operations)
      * Return the calculation to the register
      * Also performs logical operations such as true or false, etc.
    - Storage devices similar to ram
    - Fast temporary storage
    - Responsible for storing data rather than the instructions
    - Control unit allows communication between RAM and CPU
    - Instruction register changes every time there is a change in the instructions
    - Program counter is responsible counting the number of operations being performed
  + RAM (random access memory)
    - Any program you need to run must be sent to RAM before execution
    - Important for address
  + Input/output device
    - Keyboard (input) – screen (output)
    - Camera (both input and output)
    - Diagram

      Description automatically generated
* Von Neumann Architecture
  + Diagram

    Description automatically generated
  + No CPU in the von Neumann architecture
* Memory subsystem
  + Memory/RAM (Random access memory)
    - Memory cells (storage units) of a fixed size. Each cell has an address associated with it (bits)
    - All accesses to memory are to a specified address. A cell is a minimum unit of access (fetch/store a complete cell)
    - Time it takes to fetch/store a cell is the same for all cells
  + When the computer is running, both
* RAM
  + Need to distinguish between
    - Address of a memory cell and content of a memory cell
  + Memory width (W)
    - How many bits is each cell, typically 1 byte (8 bits)
  + Address width (N)
    - Bits used to represent each address, determines the maximum memory size = address space
    - If address width is N-bits then the address space is 2 ^N
  + Chart

    Description automatically generated
* Structure of memory subsystem
  + MAR stores address; controls positions and memory cells used
  + MDR responsible for store operation
  + Fetch (address)
    - Load address into MAR
    - Decode address in MAR
    - Copy content of memory cell w/ specified address into MDR
  + Store (address, value)
    - Load address into MAR
    - Load value into MDR
    - Decode address in MAR
    - Copy content of MDR into memory cell w/ specified address
  + Diagram

    Description automatically generated
* Structure of ALU
  + Registers
    - Very fast local memory cells, store operands and intermediate results
    - CCR (condition code register): special purpose register that stores result of <, =, >, operations
  + ALU circuitry:
    - Contains an array of circuits to do mathematical/logic operations.’
  + Bus:  
    nData path interconnecting the   
    registers to the ALU circuitry.
* Structure of the Control unit
  + PC (program counter)
    - Stores address of next instruction to fetch
  + IR (instruction register)
* Digital computer and information
  + Digital system: device that can read, write, or store information that is represented in numerical form
    - Use a physical quantities called signal to represent discrete information. Voltage (usually use between 0 volt and 5 volt), Currents.
  + These involve analog quantities:
  + Involves digital quantities:
* Limitations of Digital Techniques:
  + *The real world in mainly analog*
  + To deal with analog inputs, three steps must be followed
    - Convert real world
* Advantages of digital techniques
  + Digital systems are generally easier to design
  + Information storage is easy
  + Accuracy and precision are greater
  + Operation can be programmed
  + Digital circuits are less affected by noise
  + More digital circuitry can be fabricated on IC chips
* Representing Binary Quantities
  + In digital systems the
* Typical voltage Assignment
  + Binary 1: Any voltage between 2V to 5V
  + Binary 0: Any voltage between 0V to 0.8V
  + Not used: voltage between 0.8V and 2V
* Information Representation with Binary Digits
  + Binary digit is called a bit
  + Information is represented by a group of bit in the digital system
  + With n bits we can use 2n representations
  + With a bits how many discrete information can be represented?
    - 0 = FALSE = OFF
    - 1 = TRUE = ON
  + With n bits how many discrete information can be represented?
* Digital vs Analog
  + Added complexity and expense due to ADC, DAC
  + Extra time required to perform conversions
  + In most applications, digital techniques are favored because of the advantages discussed before
  + One notable exception: signal amplification is most easily achieved using analog circuitry
* Digital system
  + Gate – the smallest module (unit) in digital systems. There are AND, OR, NOT, XOR, NAND, and NOR basic gates. Each gate performs different operations based on input
  + Circuits – several gates are connected to create a circuit to perform more complicated tasks
* Logical Gates (NOT Gate)
  + A NOT gate accepts one input signal (0 or 1) and returns the opposite signal as an output
* Logical Gates (AND gate)
  + A two input AND gate accepts two input signals
  + If both are 1, the output is 1; otherwise the output is 0