Introduction To Computer Architecture

Instructor: Dr. Vinicius Prado da Fonseca (vpradodafons@online.mun.ca)

Computer Architecture vs Organization

Architecture (what this computer does):

- Attributes/operations visible to a programmer.
- Direct impact on logical execution.
- Defined by Instruction Set architecture (ISA), example:
  - armv4, RISC-V, powerPC, x86, etc

Organization (how it does):

- Operation units and their interconnection.
- Realize architectural specifications.

Computer Architecture vs Organization

Architectural design:

- Does this computer have Multiplication?

Organization:

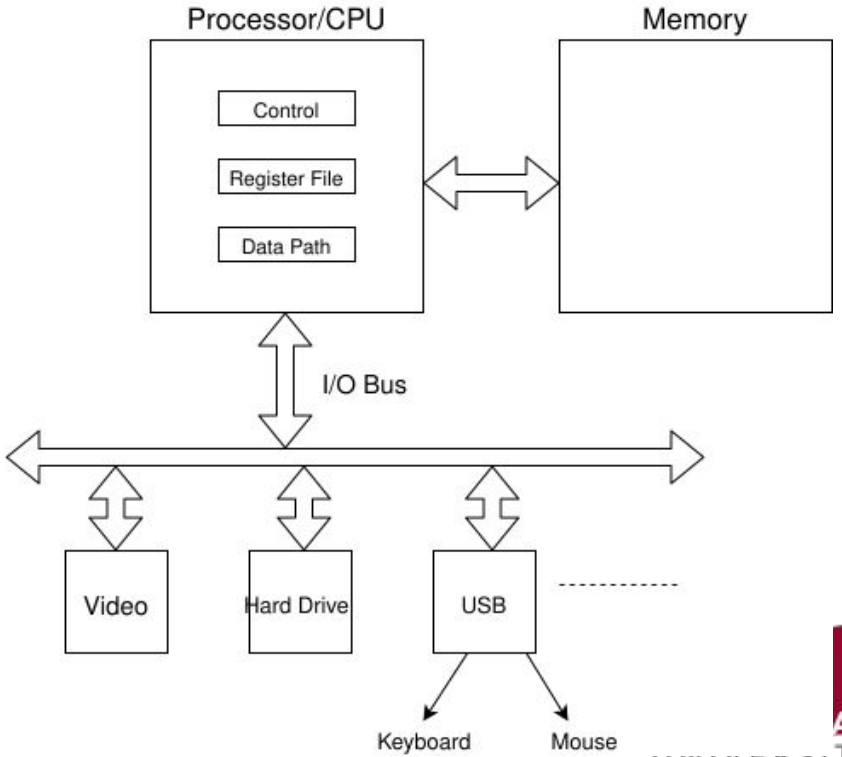
- Multiply unit? (solved on hardware)

  Or

- Repeated addition? (solved on the compiler)

Typical Computer

- CPU Central Processing Unit/Processor
- Memory
- Buses
- I/O Devices



CPU Central Processing Unit/Processor

- Program Counter (PC)
- Load data from memory
- Store data to memory
- Perform data operations on the register file (working storage)
- Conditional change the PC

```
while the computer has power:

     IR <= Memory[PC] //fetch an instruction

     Decode and execute instruction in the IR

     Adjust the PC

end while
```

Memory

Memory can be envisioned as an array.

- RAM - Random Access Memory
- SRAM - Static RAM (no refresh, cache memory)
- DRAM - Dynamic RAM (refresh, main memory)
- Flash - a read mostly memory containing the firmware

Random refers to having same access time for any position.

Buses

Groupings of wires that carry signals between components

- North Bridge (CPU, memory, PCIe)
- South Bridge (USB, audio, the system BIOS)
- PCI and PCIe
- SATA
- SCSI
- USB
- PS/2 , keyboard and mouse

I/O Devices

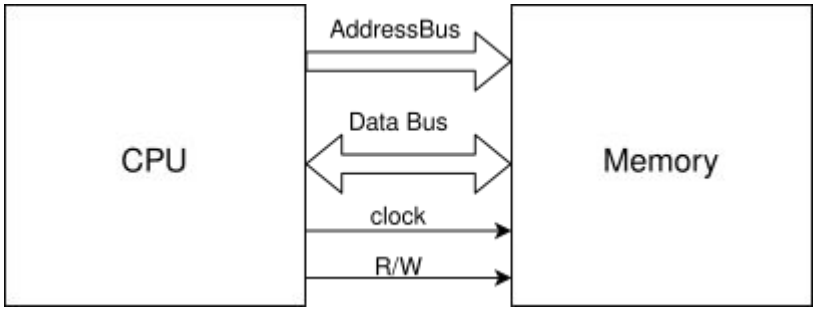
Any component that is not the CPU, memory or a bus. Usually need an interface controller.

- Video card
- Sound cards
- Hard drive controllers
- Network interfaces, wifi, Ethernet, etc.

I/O devices can also use interrupts.

Memory/CPU communication

- Store/retrieve data and instructions.
- Address bus selects the word
- Data bus for transfer
- R/W indicates direction
- Clock, signal to start transfer
  - setting the address lines to the words' location
  - setting the R/W signal to 1 for read
  - pulsing the clock line to request a read
  - waiting for the data line to be ready
- Writing: R/W to 0 for write

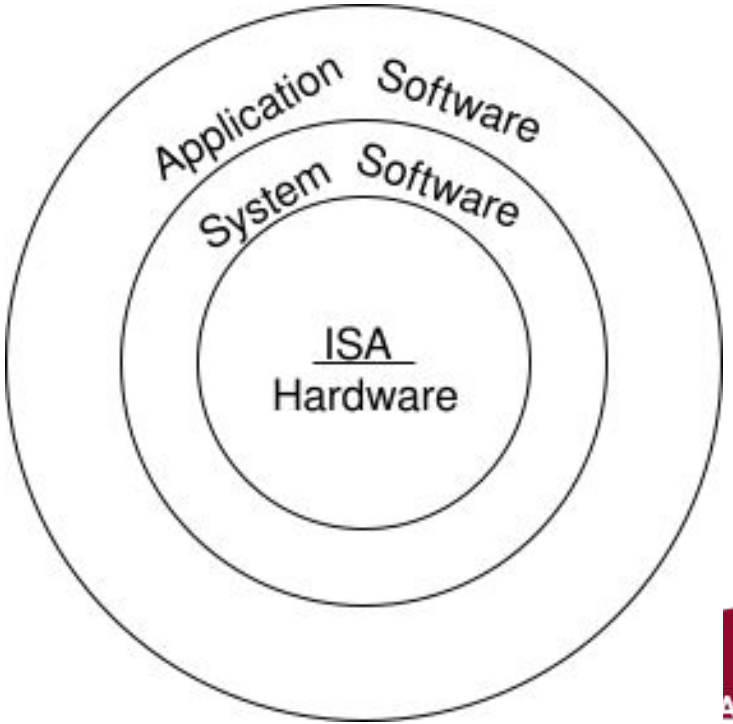


Memory Structure (n-words)

- One dimension array of n-bits equals one word.
- 8-bit is a common size
- 32 bits has max address of 2^{32}-1
- Memory management hardware to access more addresses

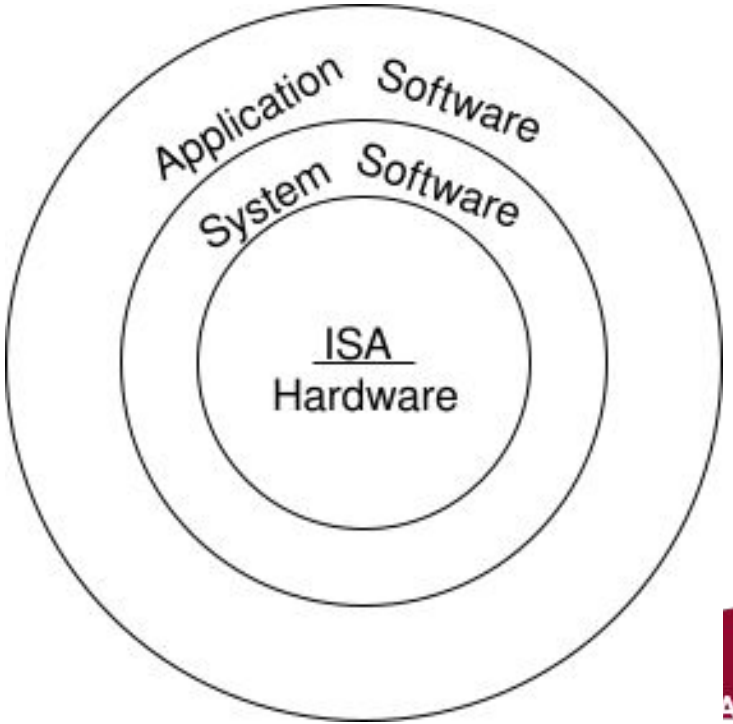
User View Of Computer Systems

- Instruction Set Architecture (ISA) interface system/application - hardware
- Encoding and behaviour of the instructions executed by the machine.
- Some common ISAs are:
  - x86 Intel/AMD (PCs and servers)
  - ARM (tablets, cellphones)
  - MIPs (embedded systems)
  - AVR (8 bit microcontroller)



User View Of Computer Systems

- Most programs use a high level language
- Machine code is responsible for all aspects, data/instructions placement, data interaction, instructions encoding
- Assembly language
  - Machine code representation
  - one-to-one mapping between assembler and machine code
- HLL -> Assembler -> Machine code
  - Virtual machines (bytecodes)



User View Of Computer Systems

- C

```
void swap(int v[], int k) {
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

- Assembly

```
swap:
    lsl    r3, r1, #2
    add    r3, r3, #4
    ldr    r2, [r0, r1, lsl #2]
    ldr    ip, [r0, r3]
    str    ip, [r0, r1, lsl #2]
    str    r2, [r0, r3]
    mov    pc, lr
```

Levels of Abstractions

- Physics
  - Electrons, atoms, semiconductors, electric fields
- Devices
  - MOSFETs - used by IC/chip manufacturers
- Analog Circuits
  - Transistors, capacitors, resistors, inductors, currents - Needed for embedded systems and timing analysis.
- Digital Circuits
  - AND, OR, NOT, gates, 0/1 - components used to build digital systems
- Logic
  - Boolean algebra, truth tables, K-maps, - used to implement/design digital circuits
- Micro-architecture
  - Adders, MUXes, registers
- Instruction Set Architecture
  - Interface between hardware and software. Operating systems, Software.

Questions?

Next: 02 - Numbers, Signed Numbers and Numbers as bits