Terminology

Last updated 8/31/21

Terminology

- General Purpose Processor
  - User Programmable
    - Intended to run end user selected programs
  - Application Independent
    - PowerPoint, Chrome, Twitter, Angry birds, …
- Embedded Processor
  - Not User Programmable
    - Programmed by manufacturer
  - Application Driven
    - Non-smart phone, appliances, missiles, automobiles, …
    - Very wide and very deep applications profile

Terminology

- General Purpose Processor
  - Key Characteristics
    - 32/64 bit operations
    - Support non-real-time/time-sharing operating systems
    - Support complex memory systems
      - Multi-level cache
      - DRAM
      - Virtual memory
    - Support DMA-driven I/O
    - Complex CPU structures
      - Extensive Pipelining
      - Superscalar execution
      - Out-of-order execution (OOO)

Terminology

- General Purpose Processor
  - Examples
    - ARM 7, 9, Cortex A8, A9,A15
    - Intel Pentiums, Ix, Core ix…
    - AMD Phenom, Athleron, Opteron
    - Apple A4, A5, A6
    - TI OMAPs

Terminology

- Embedded Processor
  - Key Characteristics
    - 4/8/16/32 bit operations
    - Support real-time operating systems
    - Relatively simple memory systems
    - Memory mapped I/O
    - Simple CPU structures
      - Few registers
      - Limited Instructions
    - Support for multiple I/O schemes
    - Wide range of peripheral support
      - A/D – D/A
      - Sensors
      - Extensive interrupt support

Terminology

- Embedded Processor
  - Examples
    - ARM Cortex Rx, Mx
      - ST Nucleo
      - TI MSP
    - Atmel AVR
      - Old Arduino platform

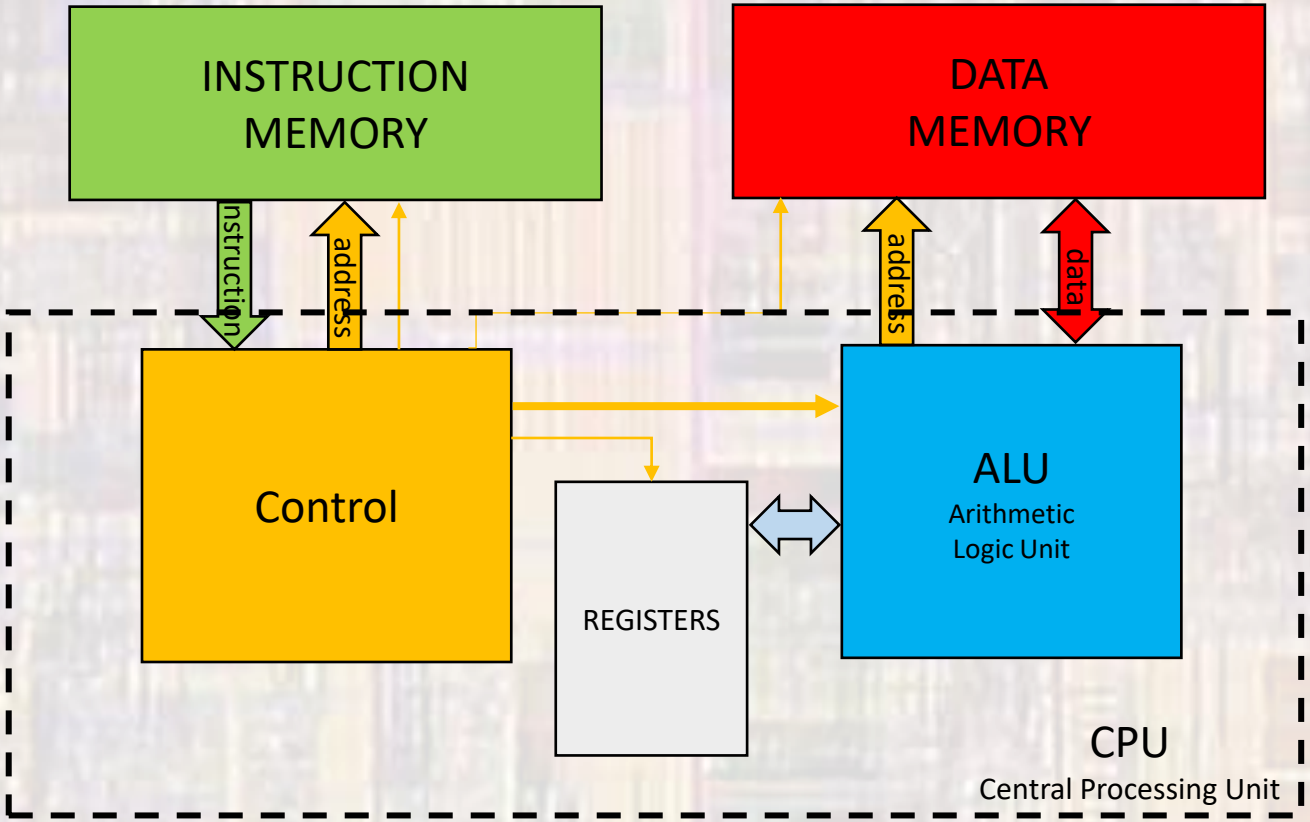
Terminology

Instruction Sets

- CISC – Complex Instruction Set Computer
  - Name didn't even exist until RISC was defined
  - Used in most processors until about 1980
  - One instruction holds multiple actions
    - Load data from location, add, write data to new location
  - Many times the instructions were designed to emulate high level language constructs
- RISC – Reduced Instruction Set Computer
  - Developed in the '80s
  - Most prevalent architecture today
  - Sometimes called a load/store architecture
  - Instructions are simple
    - Load data from location
    - Add
    - Store data to location
- RISC dominates today
  - Much easier to take advantage of advanced structures like Pipelining, Superscalar, OoO

Terminology

- Memory Structure
  - Harvard Architecture – separate Instruction and Data memories



Terminology

- Memory Structure
  - Modified Harvard – upper level common memories