UCDAVIS
UNIVERSITY OF CALIFORNIA

# Internet of Things

## Senior Design Project Course

## *Processing - Part 1*

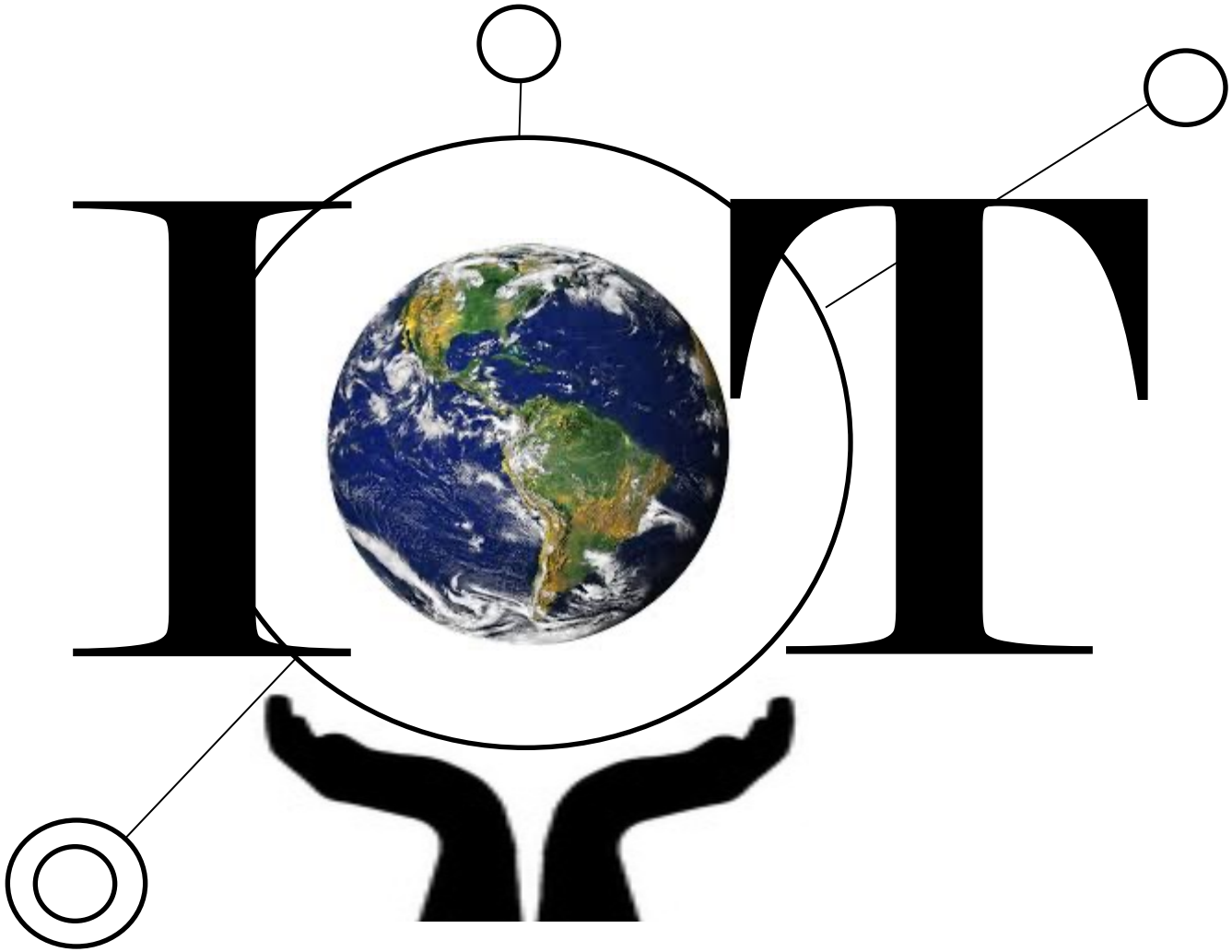
**Lecturer: Avesta Sasan**

University of California Davis

# Focus of Today's Lecture: **MCU**

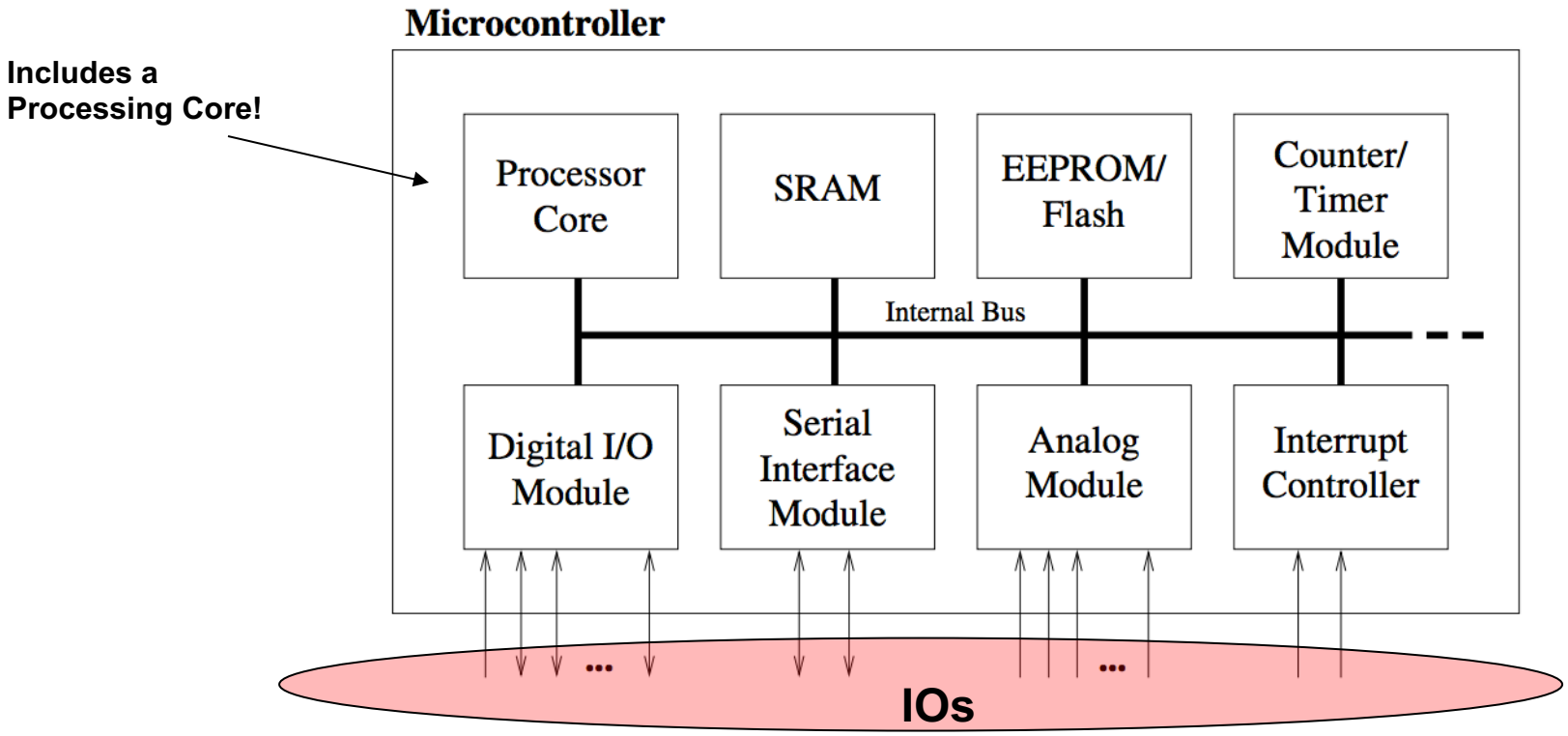

Image source: http://www.cchc.cl/informacion-a-la-comunidad/industria-de-la-construccion/personaje/

# Lets Get Started:



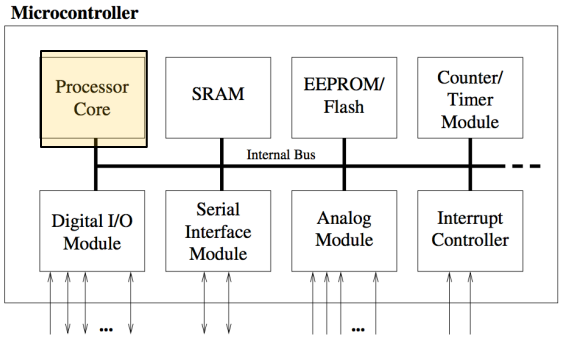
# Microcontroller Basic Design

- All components are connected via an **internal bus**.
- All components are **integrated on one chip**.
- **Communicate** to outside world **via IOs**.



# Inside a Microcontroller!

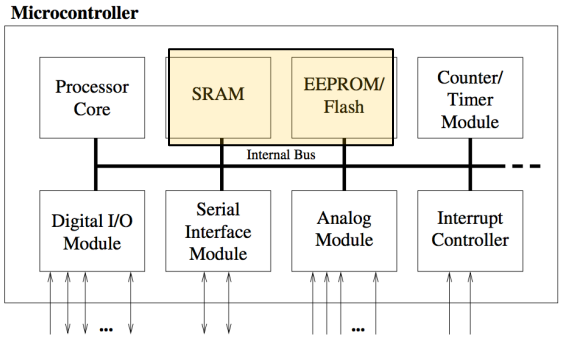
- **Processor Core:**
  - The CPU of the controller.
  - Contains the arithmetic logic unit (**ALU**), the **control unit**, and the **registers** (stack pointer, program counter, accumulator register, register file, . . . ).



# Inside a Microcontroller!

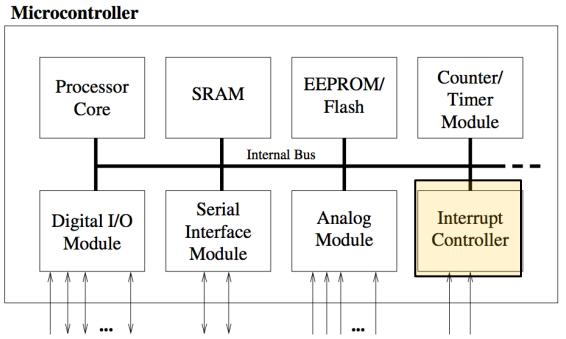
- **Processor Core:**
  - The CPU of the controller.
  - Contains the arithmetic logic unit (**ALU**), the **control unit**, and the **registers** (stack pointer, program counter, accumulator register, register file, . . . ).

- **Memory:**
  - May be split into program memory and data memory.
  - RAM, ROM, SRAM, FLASH/EEPROM



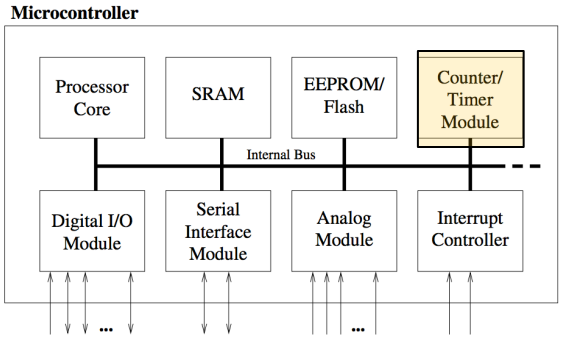
# Inside a Microcontroller!

- **Processor Core:**
  - The CPU of the controller.
  - Contains the arithmetic logic unit (**ALU**), the **control unit**, and the **registers** (stack pointer, program counter, accumulator register, register file, . . . ).
- **Memory:**
  - May be split into program memory and data memory.
  - RAM, ROM, SRAM, FLASH/EEPROM
- **Interrupt Controller:**
  - Interrupting the normal program flow in case of external or internal events.
  - When combined with sleep modes, they help to conserve power.



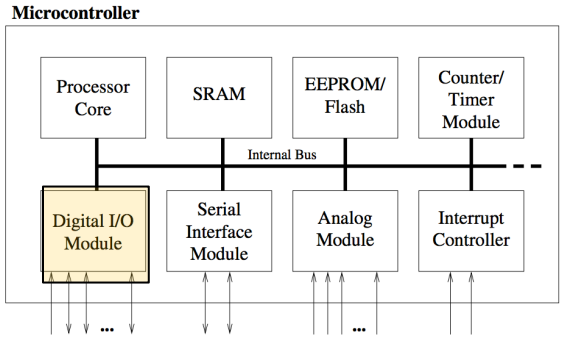
# Inside a Microcontroller!

- **Processor Core:**
  - The CPU of the controller.
  - Contains the arithmetic logic unit (**ALU**), the **control unit**, and the **registers** (stack pointer, program counter, accumulator register, register file, . . . ).
- **Memory:**
  - May be split into program memory and data memory.
  - RAM, ROM, SRAM, FLASH/EEPROM
- **Interrupt Controller:**
  - Interrupting the normal program flow in case of external or internal events.
  - When combined with sleep modes, they help to conserve power.
- **Timer/Counter:**
  - Used to timestamp events, measure intervals, or count events
  - 1-3 Timer/Counters per MCU
  - Some also contain Pulse Width Modulation (PWM).



# Inside a Microcontroller!

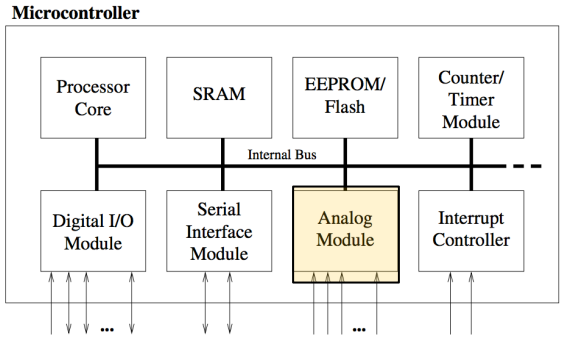
- **Digital I/O:**
  - Parallel digital I/O ports
    - from 3-4 to over 90



# Inside a Microcontroller!

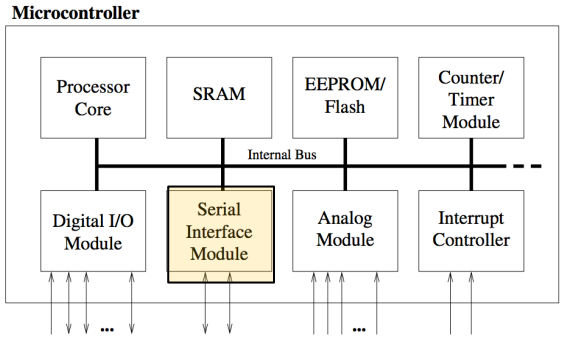
- **Digital I/O:**
  - Parallel digital I/O ports
    - from 3-4 to over 90

- **Analog I/O:**
  - 2-16 ports
  - Digital/Analog converters. (DAC, ADC)



# Inside a Microcontroller!

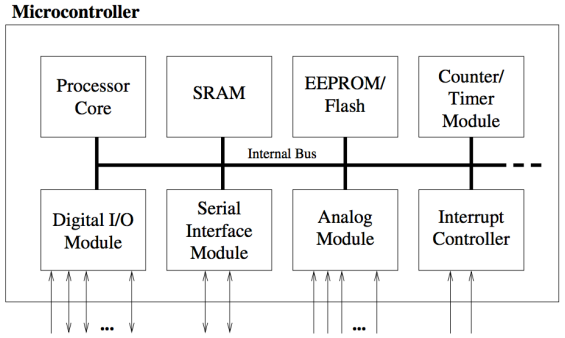
- **Digital I/O:**
  - Parallel digital I/O ports
    - from 3-4 to over 90
- **Analog I/O:**
  - 2-16 ports
  - Digital/Analog converters. (DAC, ADC)
- **Interfaces:**
  - For communication with the development PC in general
  - Most controllers offer several and varied interfaces like SPI and I2C.
  - Larger microcontrollers may also contain PCI, USB, or Ethernet interfaces.



# Inside a Microcontroller!

- **Watchdog Timer:**
  - Used to reset the controller in case of software "crashes".
  - Timer set to count down, if reaches 0, restart the microcontroller
  - To prevent restart, the software has to reset the watchdog.

- **Debugging Unit:**
  - Some include additional hardware to allow remote debugging so there is no need to download special debugging software.



# What is not Inside a MCU?

- No Cache!
- No MMU (maybe you see this in larger microcontrollers)
- No complicated pipeline (single or simple multicycle pipelines)
- No disk
- No FP ALU
- ….

stripped-down

**A microcontroller is a (stripped-down) processor which is equipped with memory, timers, (parallel) I/O pins and other on-chip peripherals.**

# Lets Review Related Terms:

- **Microprocessor:**


  - A normal CPU (Central Processing Unit).
  - Communicate with external devices by data bus
    - Peripheral devices (memory, floppy controller, USB controller, timer, . . . ) are connected to the bus.
  - Only contains data, address pins and a couple of control pins.
  - Can not operate stand-alone
    - At the very least it requires some memory and an output device

- **Mixed-Signal Controller:**
  - This is a microcontroller which can process both digital and analog signals.

- **Real-Time System:**
  - Reaction to an event has to occur within a specified time.
  - MCUs are very often used in Real Time Systems.

# Lets Review Related Terms:

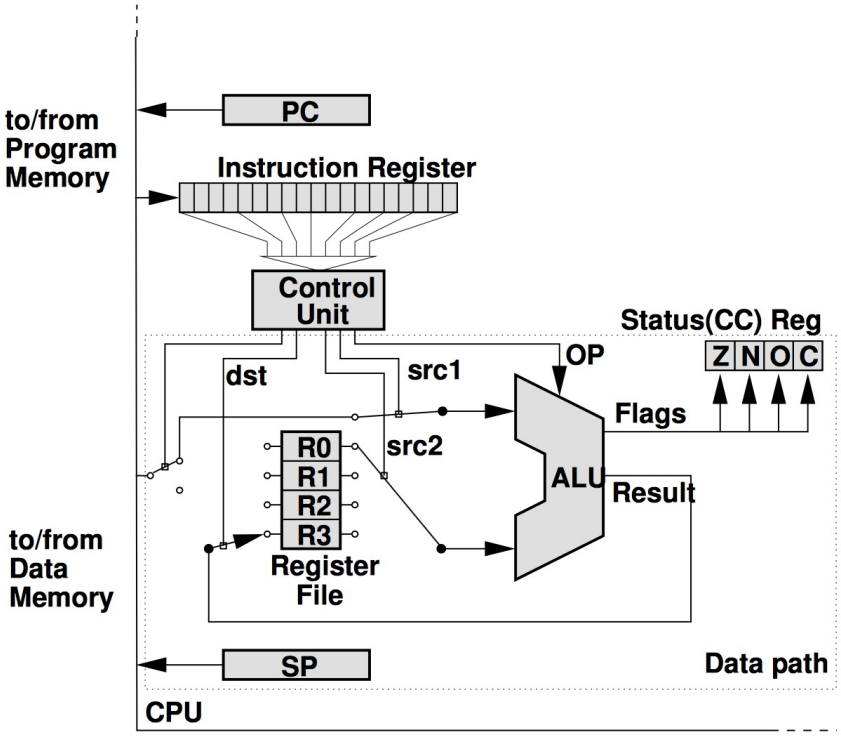
- **Embedded System:**
  - A microcontroller within a larger mechanical or electrical system
  - A complete device often including hardware and mechanical parts.

- **Embedded Processor vs Embedded Controller**
  - This term often occurs in association with embedded systems!
  - "embedded processor" is used for high-end devices (32 bits).
  - "embedded controller" is traditionally used for low-end devices (4, 8, 16 bits).

- **Digital Signal Processor (DSP):**
  - To process signals.
  - An important area of use is telecommunications.
  - Designed for fast addition and multiplication
  - Many vendors combine a controller with a DSP on one chip
    - e.g. Motorola's DSP56800.

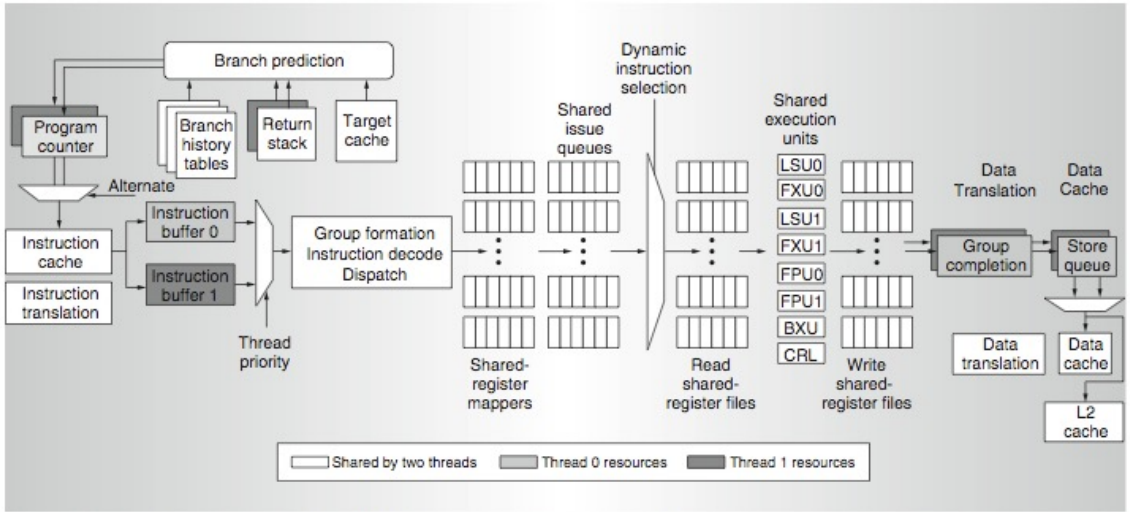
# Basic CPU architecture

- A very basic CPU architecture:
- **PC:** Program Counter: keep track of executed program.
- **Instruction Register**: keeps the incoming instruction for execution.
- **SP:** Stack Pointer
- **Control Unit:** decode the instruction and generate the control signals.
- **ALU:** Arithmetic Logic Unit: execute arithmetic operation on data
- **RF:** Register File: hold the data input to or output from ALU.
- **Status Reg:** Keep the status of the current instruction executed by ALU
  - **Z**: Zero
  - **N**: Negative
  - **O**: Overflow
  - **C:** Carry

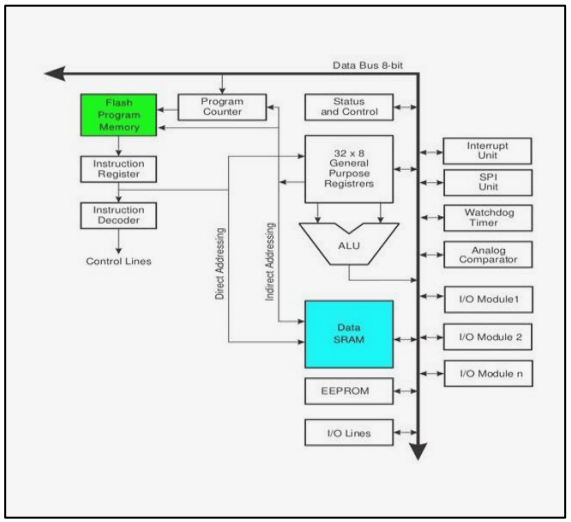


# Processor Complexity Varies Widely!

- Processor in MCU and in General Purpose Computers are very different in terms of capabilities.
  - MCU processor → Simple → light workload → low power
  - GP processor → Complex → Can handle anything → high power



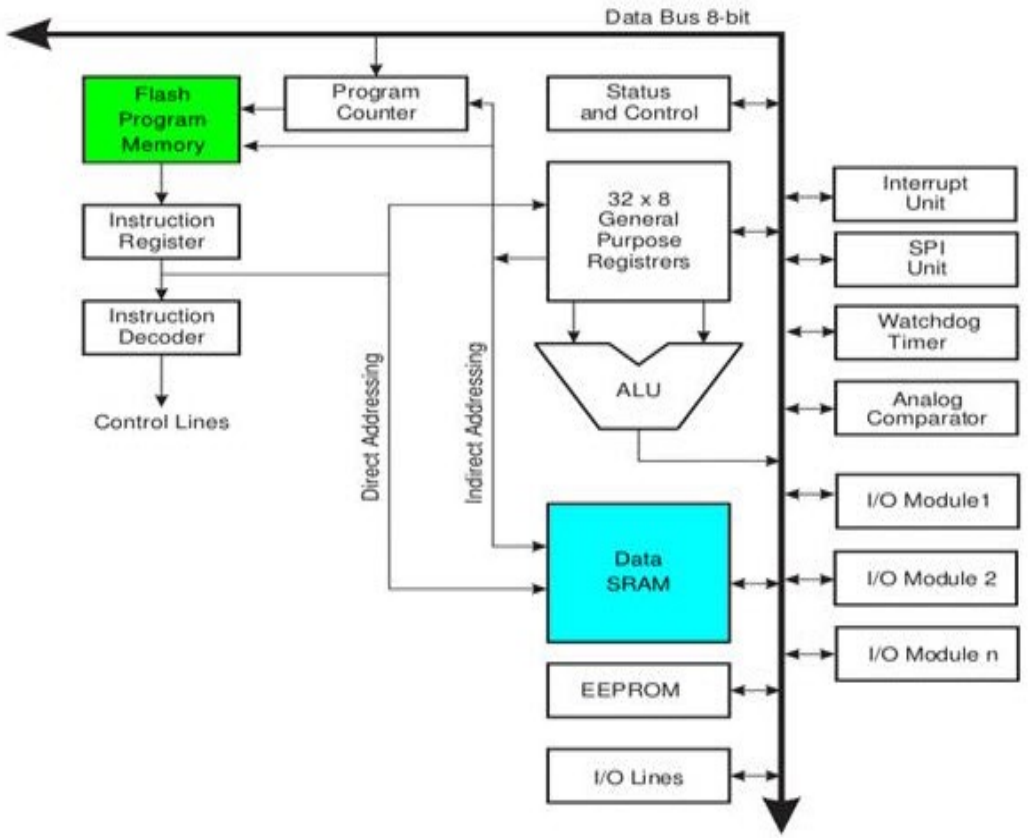

**IBM Power5 Microprocessor**




**Arduino Microprocessor**

# Example: Arduino Processor:

- Uses the Harvard architecture
  - The program code and program data have separate memories
- Single level pipeline to execute the instructions in order
- 32 x 8 bit general purpose registers
- Single clock cycle access time
- Single cycle ALU operation




**Simple**

# Example: IBM Power5

- 2 cores, out-of-order execution
- 100-entry instruction window in each core
- 8-wide instruction fetch, issue, execute
- Large, local+global hybrid branch predictor
- 1.5MB, 8-way L2 cache
- Aggressive stream based prefetching

**Complex**

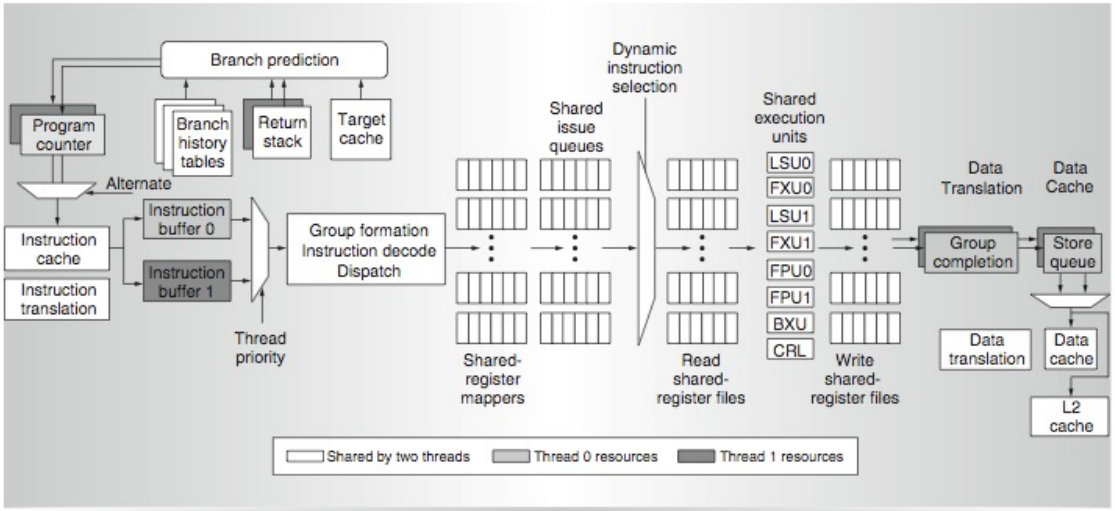
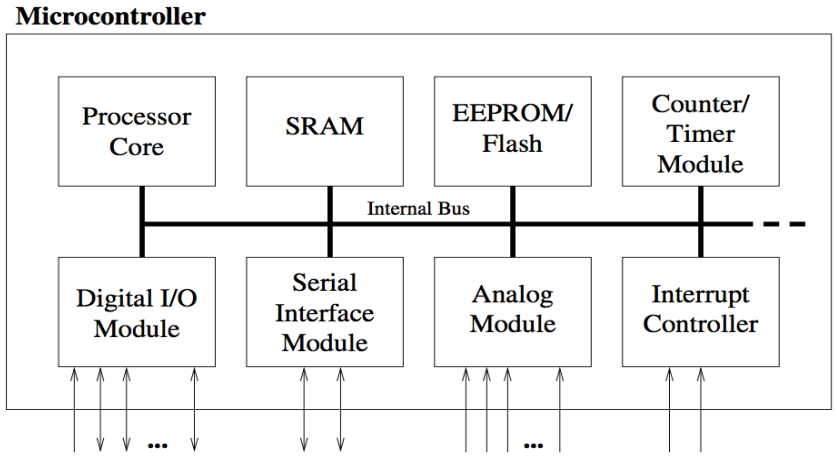
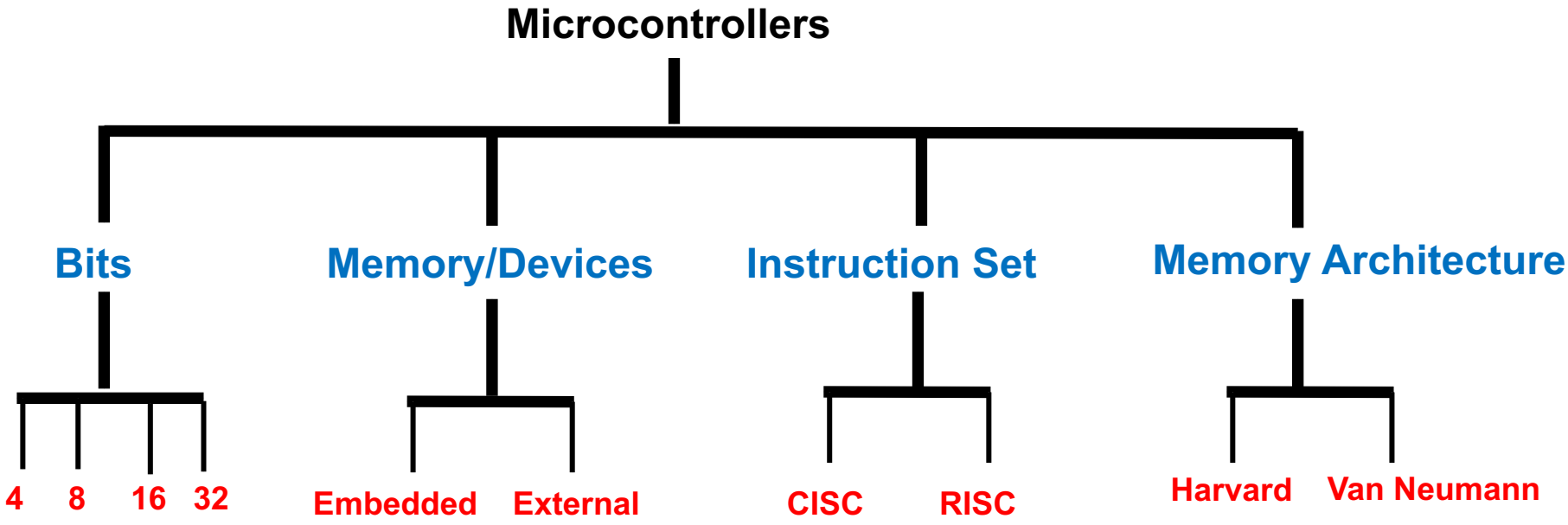



Figure 4. Power5 instruction data flow (BXU = branch execution unit and CRL = condition register logical execution unit).

# Microcontroller Classification



# Microcontroller Data Width

- Data width is the size of internal bus and size of ALU inputs.
- The larger the data width
  - Greater the precision
  - Higher the performance
  - Higher the complexity
- Note that workload should match processor complexity.
  - Using over or under capable MCU result in larger energy consumption for execution of task.
- Examples of MCU with different data widths:
  - **8 bit microprocessors:** Intel 8031/8051, PIC1x and Motorola MC68HC11 families
  - **16 bit microprocessors:** extended 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families
  - **32 bit microprocessors:** Intel/Atmel 251 family, PIC3x

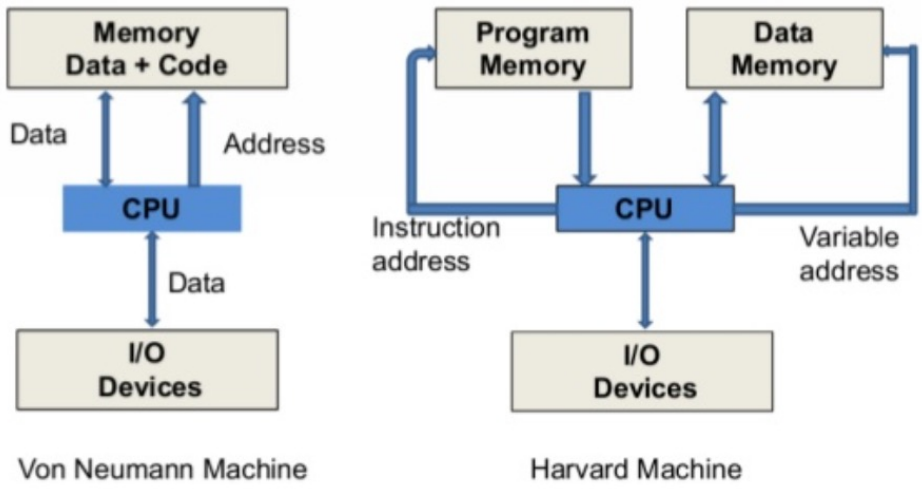
# Processor Architecture (RISC vs CISC)

- **RISC: Reduced Instruction Set Architecture**
  - Has simple, hard-wired instructions
  - Instructions take one or a few clock cycles to execute.
  - Few instructions
  - Few addressing modes.
  - The instruction set is rather simple.
  - Execution of instructions is very fast, but instructions are simple

- **CISC: Complex Instruction Set Architecture**
  - Has complex micro-coded instructions
  - Instructions can take many clock cycles to execute.
  - Powerful instructions and addressing modes.
  - In comparison to RISC, CISC takes longer to execute its instructions, but the instruction set is more powerful.

# Von Neumann vs Harvard Architecture

- If memory and data are in the same memory or not?
- **Von Neumann Architecture:**
  - program and data are stored together and are accessed through the same bus.
  - program and data accesses may conflict

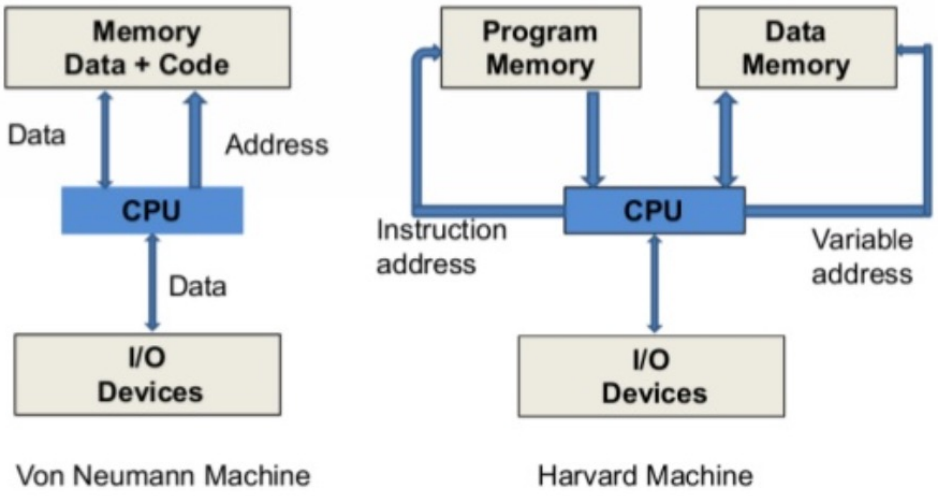



- **Harvard Architecture:**
  - program and data are in separate memories which are accessed via separate buses.
  - code accesses do not conflict with data accesses
    - improves system performance.

# Tradeoff!

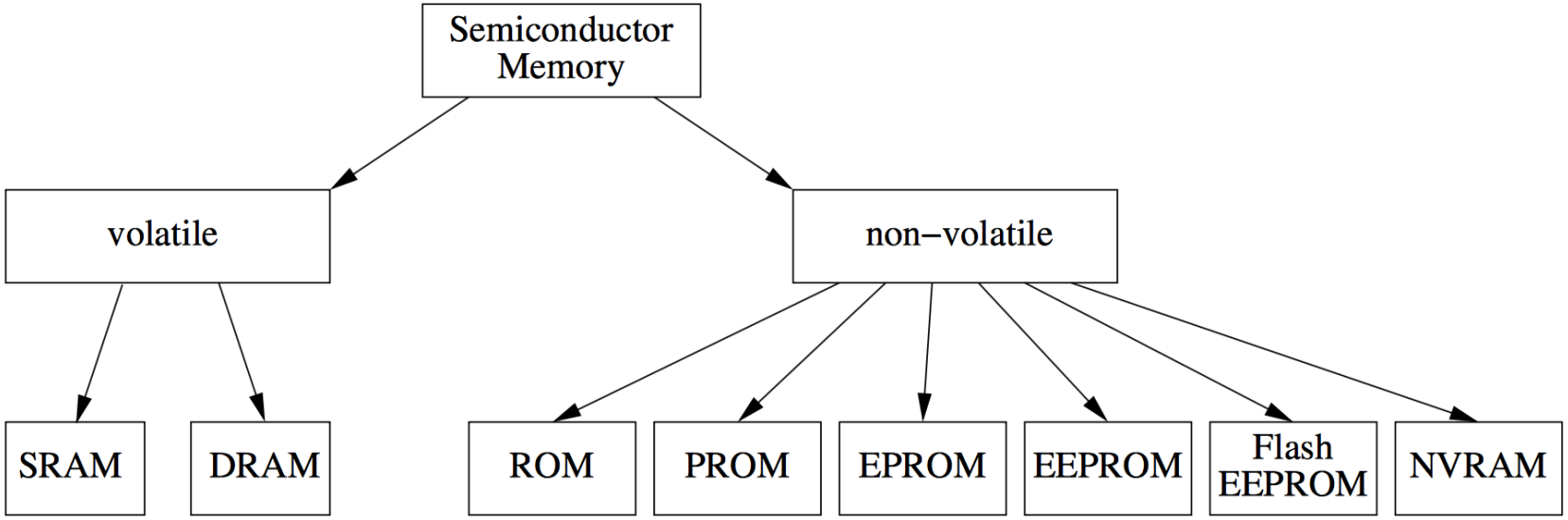
- **Harvard Architecture:** requires more hardware,
  - two busses and either two memory chips or a dual-ported memory (a memory chip which allows two independent accesses at the same time).

- **Von Neumann Architecture:**
  - conflict in simultaneous need to access instruction and data causes *bottleneck, leading to unwelcome delays.*
  - *A.K.A **Princeton Architecture***



# Memory

- Semiconductor memories could be categorized into:
  - **Volatile:** looses value if supply is disconnected
  - **Nonvolatile:** keeps the value if supply is disconnected




NVRAM

# Microcontroller vs. Microprocessor

**Microprocessor**

- CPU is stand-alone,  RAM, ROM, I/O, timer are separate
- designer can decide on the  amount of ROM, RAM and I/O ports.
- Expensive
- Versatility
- General-purpose
- High processing power
- High power consumption
- Instruction sets focus on processing-intensive operations
- Typically 32/64 – bit
- Typically deeply pipelined (5-20 stages)

**Microcontroller**

- CPU, RAM, ROM, I/O and timer are all on a single chip
- fixed amount of on-chip ROM, RAM, I/O ports
- For applications in which cost, power and space are critical
- Single (or limited) purpose (control-oriented)
- Low processing power
- Low power consumption
- Bit-level operations
- Instruction sets focus on control and bit-level operations
- Typically 8-16 bit
- Typically single-cycle/two-stage pipeline