Introduction to Microprocessor Systems

Richard Myrick T. Arellaga

Urdaneta City University

Policies and Guidelines

- **CLEAN AS YOU GO!**
- **PLAGIARISM IS A CRIME!**
- Take good care of tools and equipment used in the laboratory.
- Unauthorized use of computers to any non-class related things may result to **FAILURE** of the subject
- **LATE COMERS ARE NOT ALLOWED TO ENTER CLASS!**

Quizzes, Exams, Lab Activities, Project

- Grade $= (0.5)\text{CS} + (0.5)\text{Exam}$
- $\text{CS} = (0.2)\text{Exam} + (0.3)$ Lab Activities
- A final project will be required at the end of the subject

Historical Background

1. Mechanical Age
2. Electrical Age
3. Microprocessor Age

Mechanical Age

1. 500 BC Abacus
2. 1600s Pascaline
3. 1800s Charles Babbage Difference and Analytical Engine
4. Other

Electromechanical Age

1. Electrical Motor driven Pascaline based Machines
2. 1896 Herman Hollerith Tabulating Machine
3. 1941 Konrad Zuse Relay Logic Computer named Z3

Electrical Age

1. Invention of Vacuum Tube by Lee De Forest
2. 1943 Alan Turing Colossus
3. 1946 ENIAC designed by J Presper Eckert, J.W. Mauchly
4. Other Vacuum Tube Electronic Computers such as UNIVAC, EDVAC, Harvard Mark I, Harvard Mark II, etc.
5. Invention of Transistor by J. Barden, W. Bratain, W. Shockley
6. Invention of Integrated Circuit by Jack Kilby

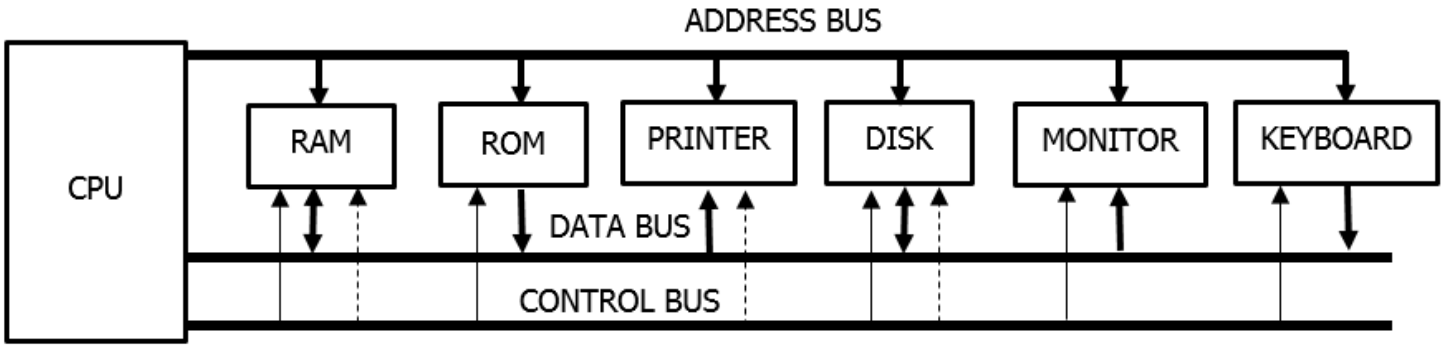
Microprocessor Age

- 1971, Intel 4004
- 1972, Intel 8008
- 1974, Intel 8080
- 1976, Z80
- 1980's onwards Intel x86
- Late 90's Intel Pentium, Intel Celeron, AMD K5, AMD K6
- early 2000's Intel Core, AMD64
- 2015 to present Ryzen, Core i9

Early Microcontrollers

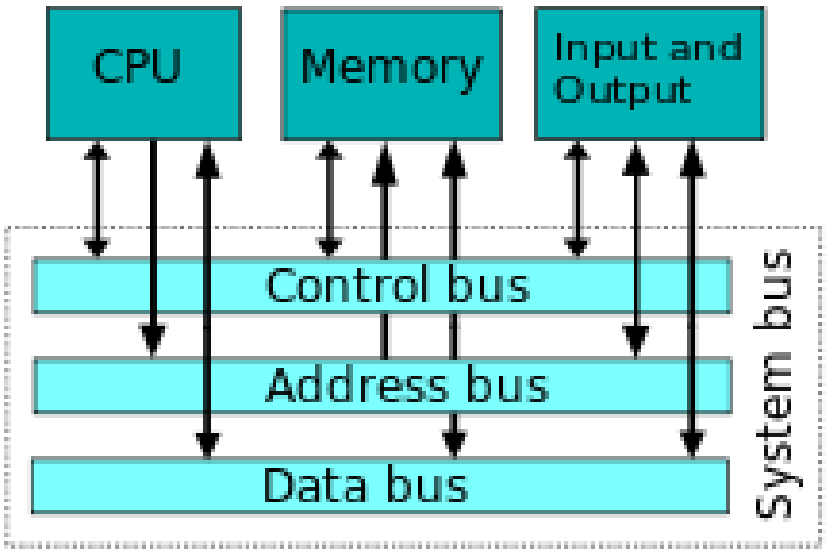
- 1970's Texas Instruments TMS1000 Family Designed by Gary Boone
- Intel MCS-48 also known as 8048 series
- Early 80's Intel MCS-51 also known as 8051 seriesl
- Zilog Z8 Family OTP MCU
- Early 90's EEPROM based Microcontrollers like PIC16C84
- Flash Based Microcontrollers like AVR, PIC16F84, and Zilog Encore etc.

Computer Organization



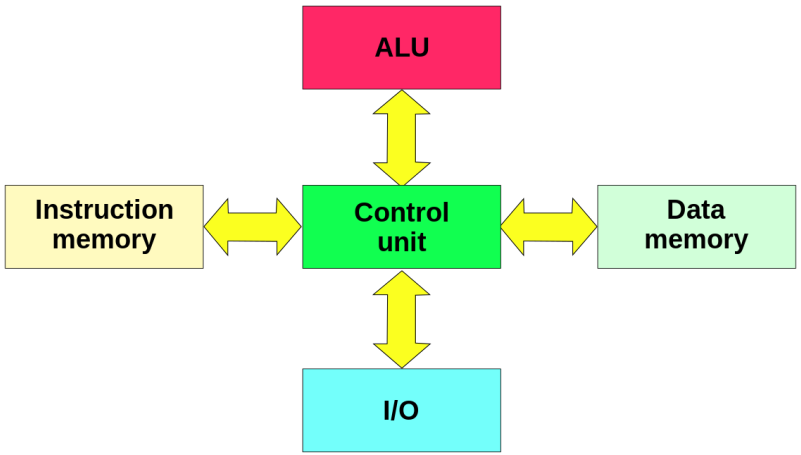
- Central Processing Unit(CPU) - Where decisions are made, computations are performed and input/output request are delegated.
- Memory - Stores information being processed by the CPU
- Input Devices - Allows people to supply information to computers
- Output Devices - Allows people to receive information from computers

Von Neumann Architecture



1. Originally referred to as Stored Program Computer
2. Instruction Fetch and Data Operation could not occur at the same time.
3. Data Memory and Program Memory share same bus and referred to as the Von Neumann bottleneck

Harvard Architecture



- Program memory and Data Memory have separate bus
- Data Operation and Instruction fetch can be done simultaneously

Software Architecture

1. Complex Instruction Set Computer
   - It is a computer where single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) and/or are capable of multi-step operations or addressing modes within single instructions.
2. Reduced Instruction Set Computer
   - It is a CPU design strategy based on the insight that simplified (as opposed to complex) instructions can provide higher performance if this simplicity enables much faster execution of each instruction.

Microprocessor Vs Microcontroller

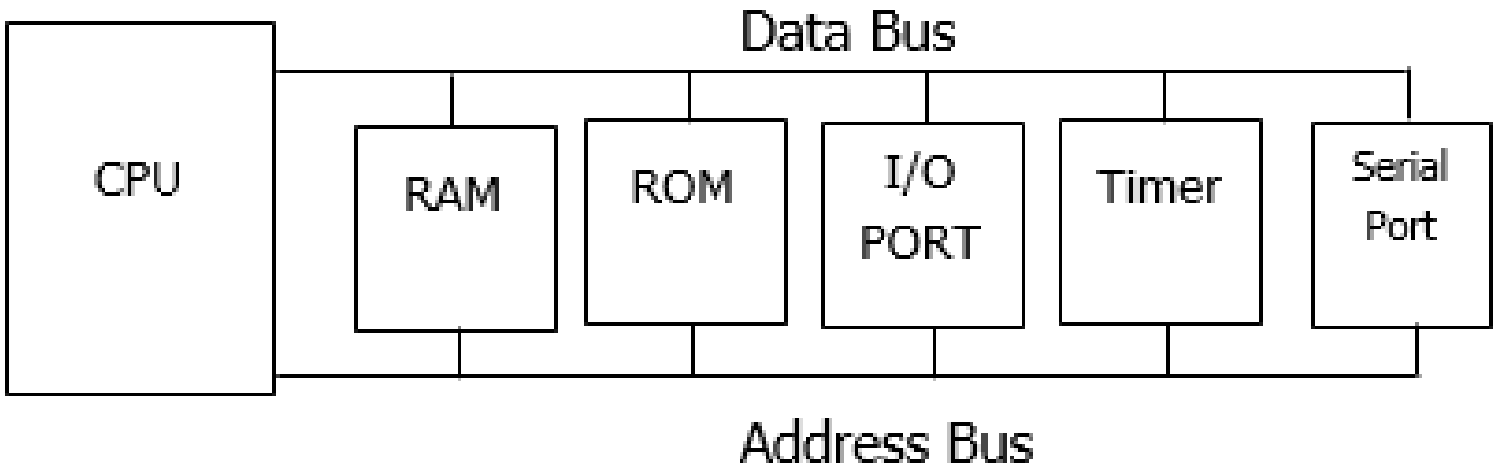


Figure: Microprocessor

Microprocessor Vs Microcontroller

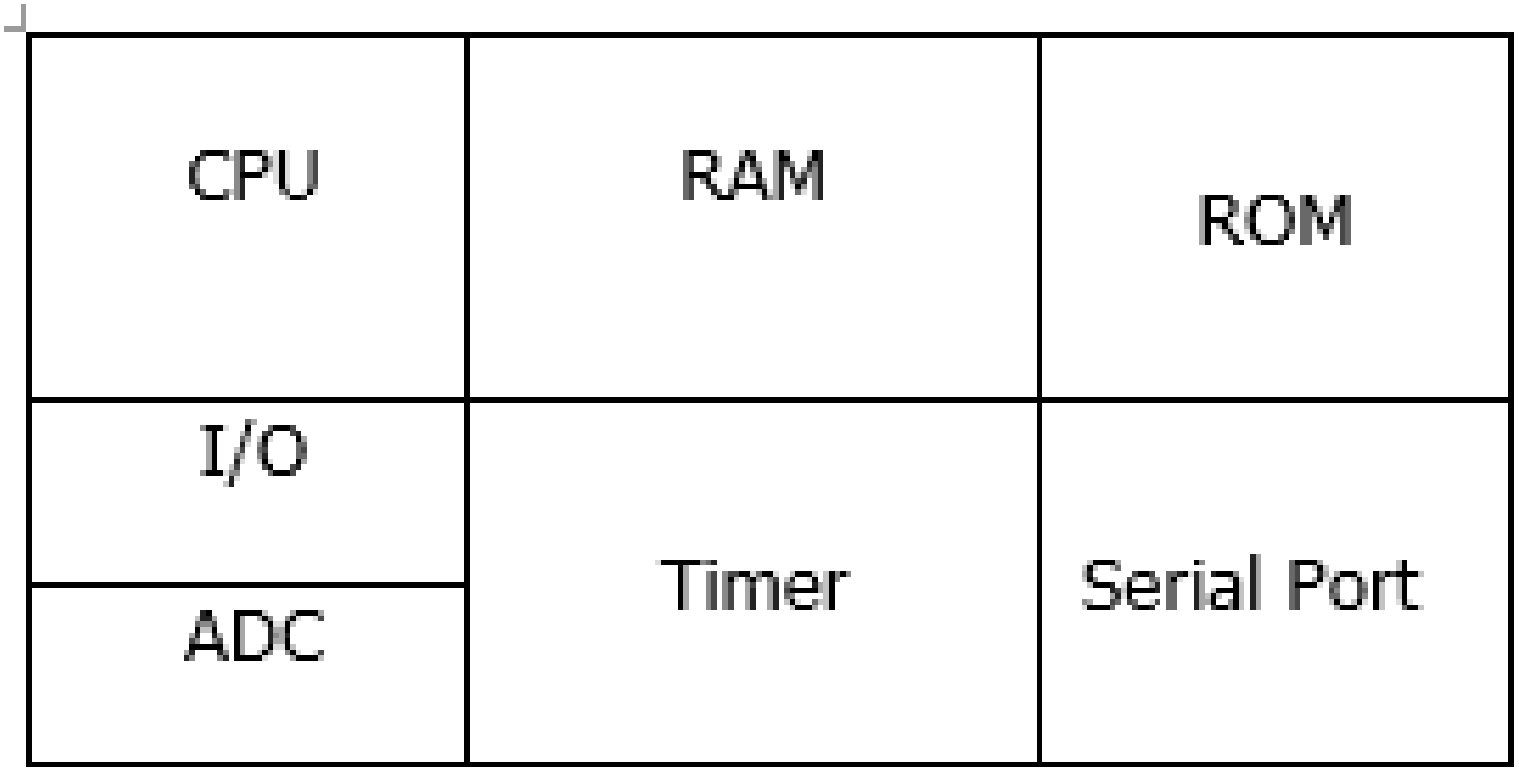


Figure: Microcontroller

Applications of Microcontrollers

- Consumer Electronics
- Home Appliances
- Business Equipment
- Automobile
- etc

Design Challenges

- Non-recurring Engineering cost
- Unit Cost
- Size
- Performance
- Power
- Flexibility
- Time-to-Market
- Time-to-Prototype
- Correctness
- Safety

Choosing a Microcontroller

- Efficiency and Cost Effectiveness
- Easy to develop product around it.
- Ready available now and in the near future.

Characteristics of Embedded System

- Single function - an embedded system usually executes only one program repeatedly until power is cut off.
- Tightly constrained - All computing systems have constraints on design metrics, but those on embedded systems can be especially tight.
- Reactive and Real-time - many embedded systems must continually react to changes in the system's environment and must computer certain results in real time without delay.