ELCT708 MicroLab
Session #1
Introduction to Embedded Systems and Microcontrollers
What is common between these systems?
What is common between these systems?

Each consists of an internal smart computer system which performs a specified task.

An Embedded System
Embedded Systems

• An embedded system consists of the smart computer pre-programmed for the specific task and surrounded by other subsystems, sensors, and actuators.

• The computer is a part of a larger complete system often with hardware and mechanical parts.
Embedded Systems
A widespread growing field due to advances in IC technologies
- **Processors have shrunk** in size with **increased performance**
- **Power consumption** has drastically reduced.
- Embedded systems + Internet = networking of Embedded systems as a part of large system across networks.

Embedded Systems are everywhere in our daily life

- Clocks and watches, Remote control
- Washing machines, Microwave ovens
- Exercise equipment, Elevators
- Mobile phones, automobiles
- Printers, electronic games: wii
- + military, industrial, medical applications...
Microcontrollers Vs Microprocessors

A microprocessor is a processor on a single chip.

- Holds the binary code for each instruction as it is executed.
- Determines the operation to perform and sets in motion the necessary actions to perform it.
- Performs arithmetic and logic operations.

**General Block Diagram of a CPU (Microprocessor):**

- Instruction Register (IR)
- Program Counter (PC)
- Registers
- Instruction Decode & Control Unit
- Arithmetic & Logic Unit (ALU)
- Temporary storage of information.
- Holds the memory address of the next instruction to be executed.
Microcontrollers Vs Microprocessors

- A microcontroller is a microcomputer on a single chip
  = a microprocessor (CPU) + added circuitry on a single chip.
Microcontrollers Vs Microprocessors

- The microcontroller is a microcomputer on a single chip: CPU(microprocessor), memory, I/O interfaces, timers and other peripherals are on the same chip.

- Limitations in terms of program memory, data memory and I/O interfaces depending on the used microcontroller.

- Single purpose usage (Eg: mobile phone)

- Project design cost is reduced, Single chip ➔ less space, less power and less cost and higher reliability.

- Slow clock rates

- A microprocessor is a processor on a single chip used with other components on different chips.

- Configurable memory size and I/O interfaces

- General purpose usage(Eg: PC)

- Project design on different chips ➔ more space, more power and more cost

- High clock rates.
Microcontrollers Vs FPGA
Microcontrollers Vs FPGA

- A microcontroller is a pre-built device with a built-in CPU, fixed size memory and well defined I/O ports.
- The built-in features of the μcontroller are used to control devices and systems by programming it.
- Suited for control and medium performance.
- Software based programming (Assembly and C used).

- FPGA consists of large number of gates and Flip-Flops plus interconnection with these blocks.
- In an FPGA, the interconnects can be configured and programmed to implement any digital logic → Can implement a μcontroller or μprocessor on an FPGA.
- FPGA usually used where high speed processing is need.
- Hardware based programming (HDL is used).
Microcontrollers Classification

- **Bits**: 4, 8, 16, 32
- **Family**: Microchip PIC, Atmel AVR, Cypress PSoC, ARM, Others...
- **Instruction set**: RISC, CISC
- **Memory architecture**: Von-Neumann, Harvard

- **Examples**: Intel 8051, Microchip PIC, Atmel AVR, Cypress PSoC, ARM, Arduino Board, Others...
Microcontrollers Classification

**Instruction Set:**

**RISC** → Reduced Instruction Set
- Instruction set supports fewer addressing modes for arithmetic, logical and data transfer instructions.
- The benefits of RISC design simplicity are a smaller chip, smaller pin count, and very low power consumption.
- Example: PIC16F8xx family

**CISC** → Complex Instruction Set
- Instruction set supports many addressing modes for arithmetic, logical, data transfer and memory accesses instructions.
- Instructions are macro like in CISC architecture
- Example: Intel 8096 family
Microcontrollers Classification

Memory Architecture:

Von-Neumann Architecture:
- CPU
- Main memory for both data memory (RAM) and program instructions (ROM and possible RAM)
- Slower \(\rightarrow\) fetches instruction then data
- Simpler, lower cost \(\rightarrow\) only one memory is accessed

Harvard Architecture:
- CPU
- Program memory (ROM)
- Data memory (Registers file)
- Execution in parallel \(\rightarrow\) fast execution on expense of complexity.

ELCT 708 Session #1
Dr. Mohamed Abdel Ghany
Eng. Salma Hesham
PIC 16F877A Microcontroller

40-Pin PDIP

MCLR/VPP  1  40  RB7/PGD
RA0/AN0  2  39  RB6/PGC
RA1/AN1  3  38  RB5
RA2/AN2/VREF-/CVREF  4  37  RB4
RA3/AN3/VREF+  5  36  RB3/PGM
RA4/T0CKI/C1OUT  6  35  RB2
RA5/AN4/SS/C2OUT  7  34  RB1
RE0/RD/AN5  8  33  RB0/INT
RE1/WR/AN6  9  32  VDD
RE2/CS/AN7  10  31  Vss
VDD  11  30  RD7/PSP7
Vss  12  29  RD6/PSP6
OSC1/CLKI  13  28  RD5/PSP5
OSC2/CLKO  14  27  RD4/PSP4
RC0/T1OSO/T1CKI  15  26  RC7/RX/DT
RC1/T1OSI/CCP2  16  25  RC6/TX/CK
RC2/CCP1  17  24  RC5/SDO
RC3/SCK/SCL  18  23  RC4/SDI/SDA
RD0/PSP0  19  22  RD3/PSP3
RD1/PSP1  20  21  RD2/PSP2
Block Diagram of PIC 16F877A
Steps of a System Design using PIC 16F877A

1. Writing the required program code on the appropriate software
2. Connect the Hardware circuitry surrounding the PIC with all needed supplies, inputs and outputs
3. Program the code on the PIC then return it back to the hardware circuit and it should be controlling the circuit now

Example

```
LED Blinking at PORTB
MikroC-Code
Void main(){
  PORTB = 0;
  TRISB = 0;
  While(1){
    PORTB = ~ PORTB;
    Delay_ms(1000);
  }
}
```
Components for next Lab

- 1 PIC16F877A
- 1 Breadboard
- 1 LCD/ 7 segment
- 10 switches/push buttons
- 10 LEDs
- 1 battery 9V
- Voltage regulator 7805
- Crystal oscillator 4MHz
- 4 capacitors 22pF
- 5 Resistors 10KOhm
- 10 Resistors 330 Ohm
- 5 transistors 2N2222
- 2m wires