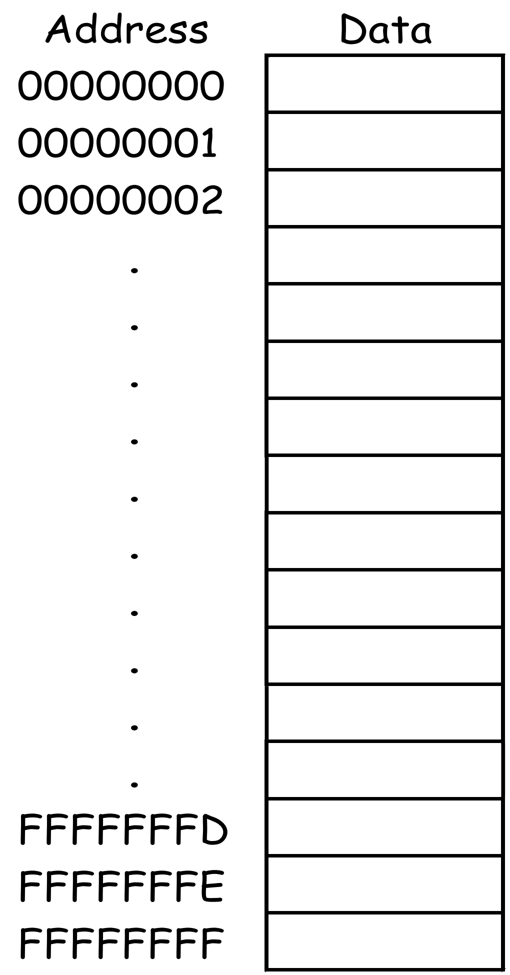
Memory, Microcontrollers and Microprocessors

Hoda Roodaki
hroodaki@kntu.ac.ir

Picture of Memory

- You can think of memory as being one big array of data.
  - The address serves as an array index.
  - Each address refers to one word of data.
- You can read or modify the data at any given memory address, just like you can read or modify the contents of an array at any given index.



Memory Address, Location and Size

- Memory sizes are usually specified in numbers of bytes (1 byte= 8 bits).
- The 2^{28}-bit memory translates into:

2^{28} bits / 8 bits per byte = 2^{25} bytes

- With the abbreviations below, this is equivalent to 32 megabytes.

	Prefix	Base 2	Base 10
K	Kilo	2^{10} = 1,024	10^{3} = 1,000
M	Mega	2^{20} = 1,048,576	10^{6} = 1,000,000
G	Giga	2^{30} = 1,073,741,824	10^{9} = 1,000,000,000

Memory Address, Location and Size

- Location - the smallest selectable unit in memory
  - All bits in location are read/written together
  - Cannot manipulate single bits in a location
- For k address signals, there are 2^k locations in memory device.
- Each location contains an n bit word.
- Memory size is specified as
  - Number of location x bits per location
    - 2^{24} x 16 RAM = 16M words, each 16 bits long
    - 24 address lines, 16 data lines
  - #bits
    - The total storage capacity is 2^{24} x 16 = 2^{28} bits

Memory Signal

- Memory signals:
  - Address bus - selects one of memory locations
    - Data bus
      - Read: the selected location's stored data is put on the data bus
      - Write (RAM): The data on the data bus is stored into the selected location
    - Control signals - specifies what the memory is to do
      - Control signals are usually active low
      - Most common signals are:
        - » CS: Chip Select; must be active to do anything
        - » OE: Output Enable; active to read data
        - » WR: Write; active to write data

Read-only memory (ROM)

- Non-volatile
  - If un-powered, its content retains
- Read-only
  - normal operation cannot change contents
- There are five basic ROM types:
  - PROM
  - OTP ROM (One Time Programmable EPROM)
  - EPROM(UV Erasable Programmable ROM)
  - EEPROM (Electrically Erasable and Programmable ROM)
  - Flash memory
- Each type has unique characteristics, but they are all types of memory with two things in common:
  - Data stored in these chips is **nonvolatile** -- it is not lost when power is removed.
  - Data stored in these chips is either **unchangeable** or requires a special operation to change.
    - This means that removing the power source from the chip will not cause it to lose any data.

Read-only memory (ROM)

- Content loading (programming) done many ways depending on device type
  - Programmed ROM (PROM): contents loaded at the factory
    - Hardwired - can’t be changed
    - Embedded systems
  - OTP (One Time Programmable): Programmed by user
    - Can be written to or programmed via a special device called a PROM programmer.
    - Typically, this device uses high voltages to permanently destroy or create internal links (fuses) within the chip. Consequently, a PROM can only be programmed once.

Read-only memory (ROM)

- UVPROM: reusable, erased by UV light
  - The EPROM device is programmed by forcing an electrical charge on a small piece of polysilicon material (called the floating gate) located in the memory cell. When this charge is present on this gate, the cell is "programmed," usually a logic "0," and when this charge is not present, it is a logic "1."
  - Prior to being programmed, an EPROM has to be erased. To erase the EPROM, it is exposed to an ultraviolet light for approximately 20 minutes.
- EEPROM: Electrically erasable; clears entire blocks with single operation.

ROM Usage

- ROMs are useful for holding data that never changes.
  - Arithmetic circuits might use tables to speed up computations of logarithms or divisions.
  - Many computers use a ROM to store important programs that should not be modified, such as the system BIOS.
  - Application programs of embedded systems, game machines, cell phones, vending machines, etc., are stored in ROMs.

Memories and functions

- ROMs are actually combinational devices, not sequential ones!
  - You can store arbitrary data into a ROM, so the same address will always contain the same data.
  - You can think of a ROM as a combinational circuit that takes an address as input, and produces some data as the output.

RAM

- Random-access memory, or RAM, provides large quantities of temporary storage in a computer system.
  - Memory cells can be accessed to transfer information to or from any desired location, with the access taking the same time regardless of the location.
- Volatility
  - Most RAMs lose their memory when power is removed
  - NVRAM = RAM + battery
- SRAM (Static RAM)
  - Memory behaves like latches or flip-flops
- DRAM (Dynamic Memory)
  - Memory lasts only for a few milliseconds
  - Must “refresh” locations by reading or writing

SRAM

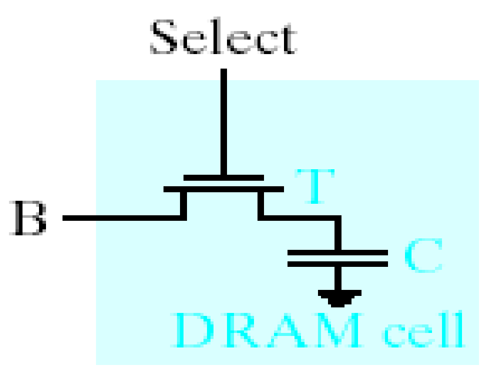
- It is a type of semiconductor memory that uses latching circuitry (Flip-flop) to store each bit. SRAM is still *volatile* in the conventional sense that data is eventually lost when the memory is not powered.

DRAM

- It is a type of random-access memory that stores each bit of data in a separate capacitor within an integrated circuit.
- The capacitor can be either charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1.
- Since even "nonconducting" transistors always leak a small amount, the capacitors will slowly discharge, and the information eventually fades unless the capacitor charge is refreshed periodically.

Dynamic memory

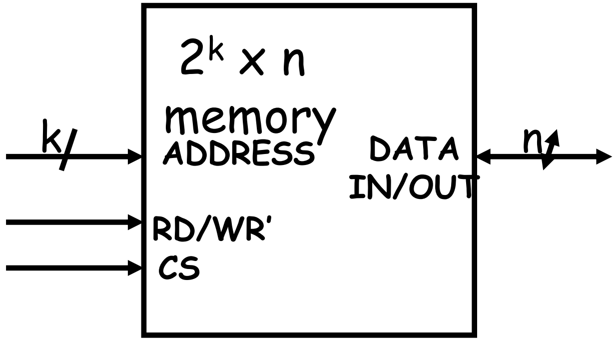
- DRAM cell: One transistor and one capacitor
  - 1/0 = capacitor charged/discharged



(a)

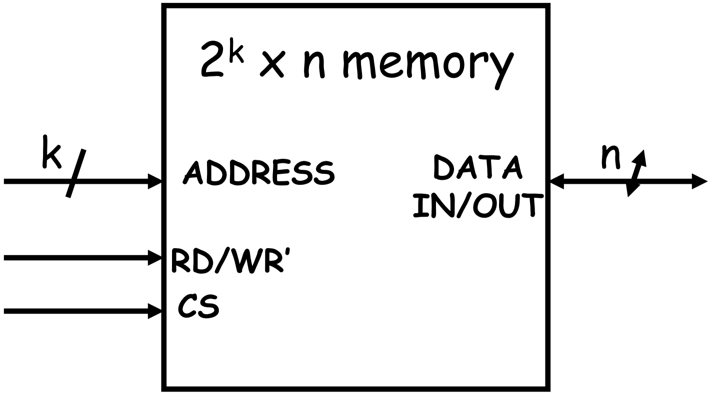
Block diagram of RAM

- This block diagram introduces the main interface to RAM.
  - A Chip Select, CS, enables or disables the RAM.
  - ADRS specifies the address or location to read from or write to.
  - RD/WR' selects between reading from or writing to the memory.
    - To read from memory, RD/WR' should be set to 1. DATA IN/OUT will be the n-bit value stored at ADRS.
    - To write to memory, we set RD/WR' to 0. DATA IN/OUT is the n-bit value to save in memory.



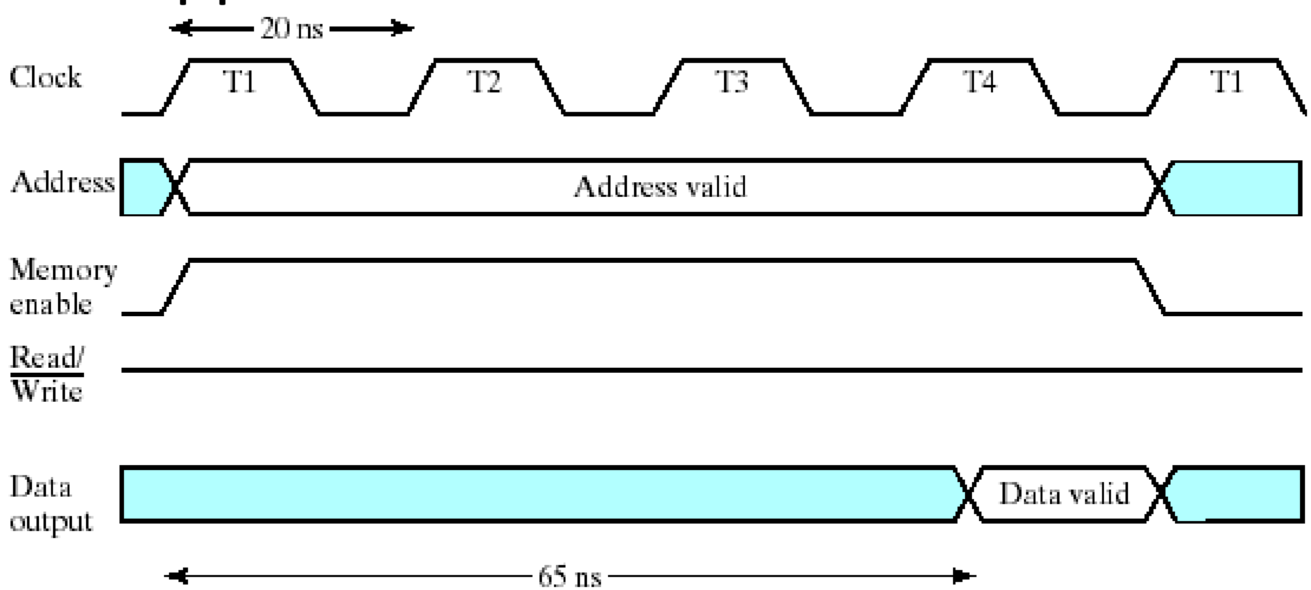
Reading from RAM

- To *read* from this RAM, the controlling circuit must:
  - Enable the chip by ensuring CS = 1.
  - Select the read operation, by setting RD/WR' = 1.
  - Send the desired address to the ADRS input.
  - The contents of that address appear on DATA IN/OUT after a little while.



Reading from RAM

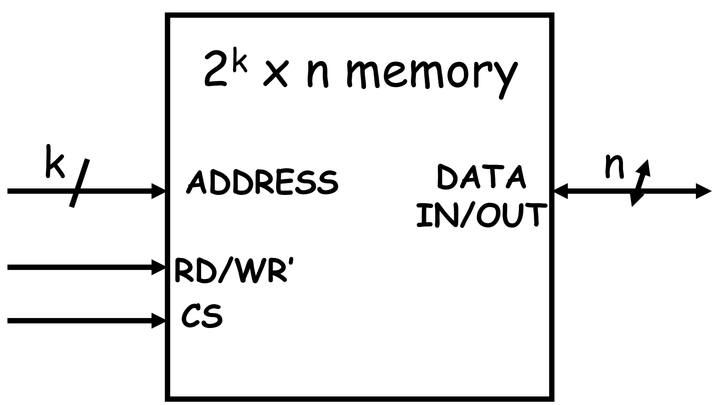
- 50 MHz CPU – 20 ns clock cycle time
- Memory access time= 65 ns
  - Maximum time from the application of the address to the appearance of the data at the Data Output.



(b) Read cycle

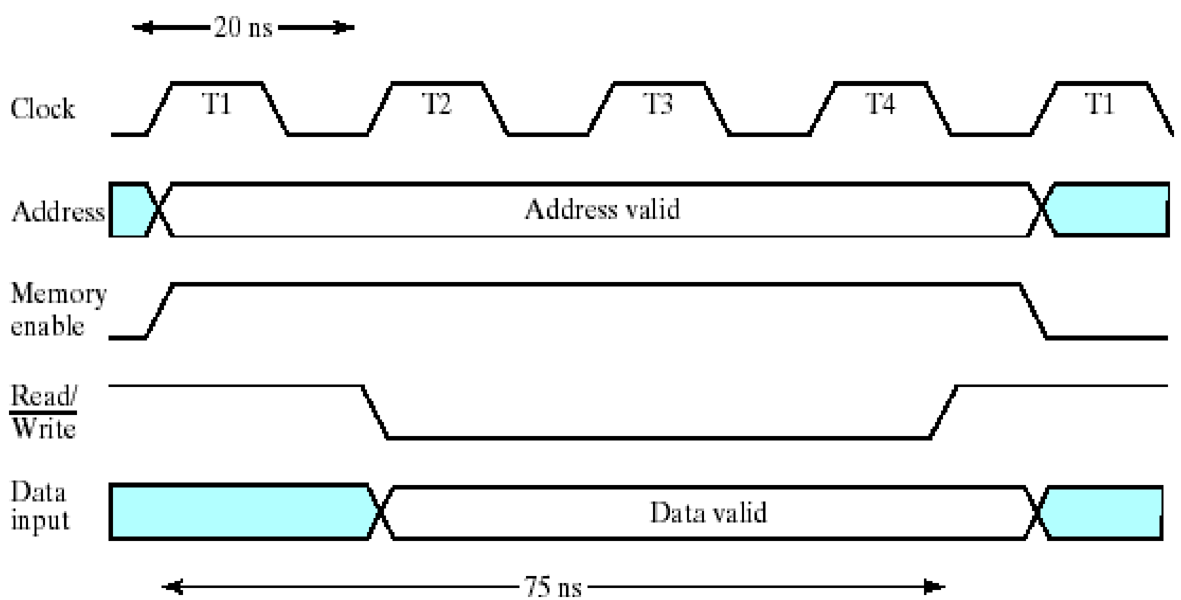
Writing RAM

- To *write* to this RAM, you need to:
  - Enable the chip by setting CS = 1.
  - Select the write operation, by setting RD/WR’ = 0.
  - Send the desired address to the ADRS input.
  - Send the word to store to the DATA IN/OUT.



Writing RAM

- 50 MHz CPU – 20 ns clock cycle time
- Write cycle time= 75 ns
  - Maximum time from the application of the address to the completion of all internal memory operations to store a word



(a) Write cycle

BUS

- **Bus** is a communication system that transfers data between components inside a computer, or between computers.
  - **Internal bus**
    - The internal bus, connects all the internal components of a computer, such as CPU and memory, to the motherboard. Internal data buses are also referred to as a local bus, because they are intended to connect to local devices. This bus is typically rather quick and is independent of the rest of the computer operations.
  - **External bus**
    - The external bus, or expansion bus, is made up of the electronic pathways that connect the different external devices, such as printer etc., to the computer

8-bit, 16-bit and 32-bit microcontrollers

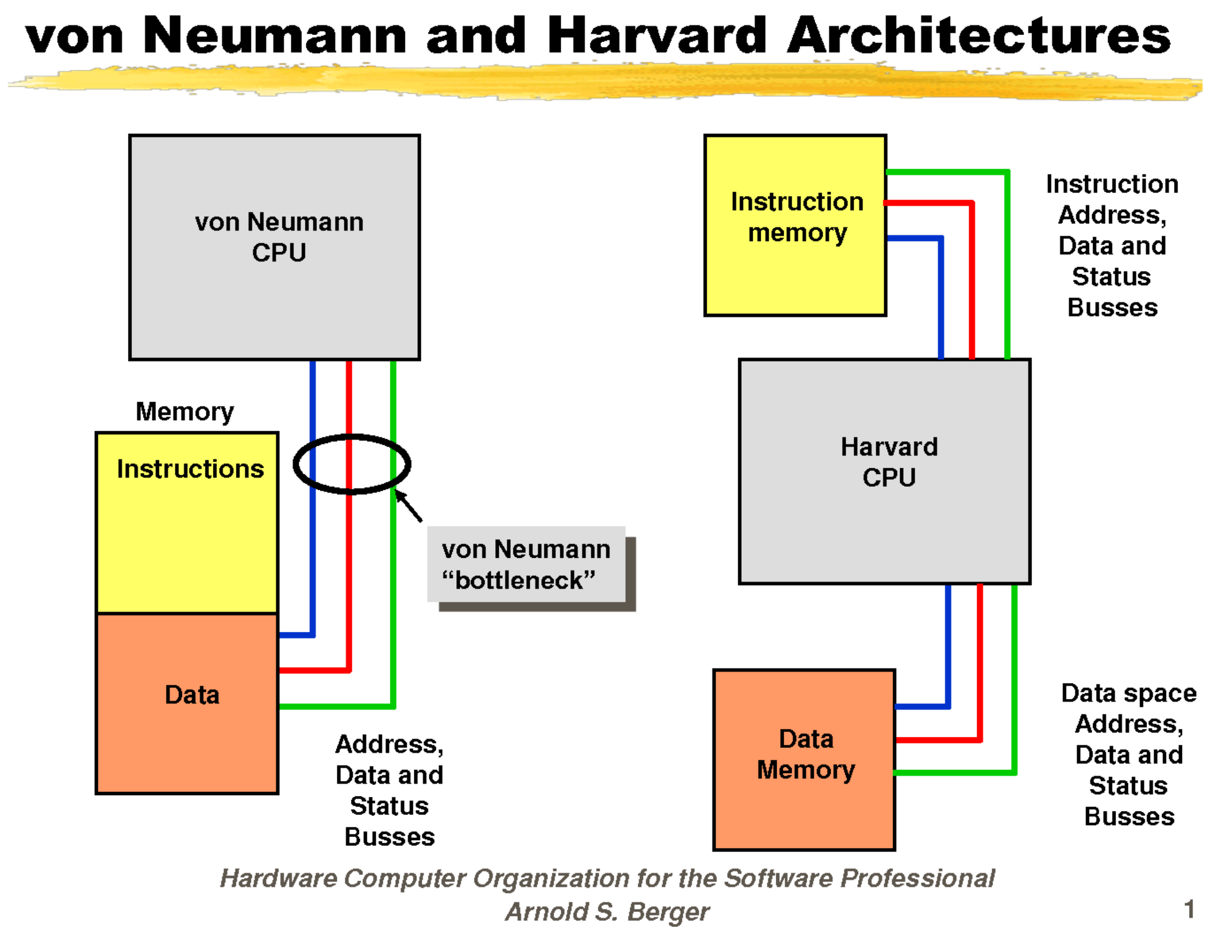
- Going from 8 to 16 to 32 bit microcontrollers means you will have fewer restraints on resources, particularly memory, and the width of registers used for doing arithmetic and logical operations.
  - The 8, 16, and 32-bit generally refers to both the size of the internal and external data busses and also the size of the internal register(s) used for arithmetic and logical operations .

Harvard & von Neumann architecture

- Harvard Architecture
  - Computer architectures that used physically separate storage and signal pathways for their instructions and data.
  - CPU can read both an instruction and data from memory at the same time that makes it faster.

- von Neumann architecture
  - CPU can Read an instruction or data from/to the memory.
  - Read, Write can`t occur at the same time due to same memory and signal pathway for data and instructions.

Harvard & von Neumann architecture



The Steps to Execute an Instruction

- The way the central processing unit, in association with memory, executes a computer program.
- Before an instruction can be executed, program instructions and data must be placed into memory from an input device or a secondary storage device.
- Once the necessary data and instruction are in memory, the central processing unit performs the following four steps for each instruction:

The Steps to Execute an Instruction

- The control unit fetches (gets) the instruction from memory.
- The control unit decodes the instruction (decides what it means) and directs that the necessary data be moved from memory to the arithmetic/logic unit.
- The arithmetic/logic unit executes the arithmetic or logical instruction. That is, the ALU is given control and performs the actual operation on the data.
- The arithmetic/logic unit stores the result of this operation in memory or in a register.

Principle of pipeline

Cycle	In	In processing				Out (Finished)
1. Cycle	Instr 1 →	F_1				
2. Cycle	Instr 2 →	F_2	D_1			
3. Cycle	Instr 3 →	F_3	D_2	E_1		
4. Cycle	Instr 4 →	F_4	D_3	E_2	WB_1	→ Instr 1
5. Cycle	Instr 5 →	F_5	D_4	E_3	WB_2	→ Instr 2

CISC & RISC

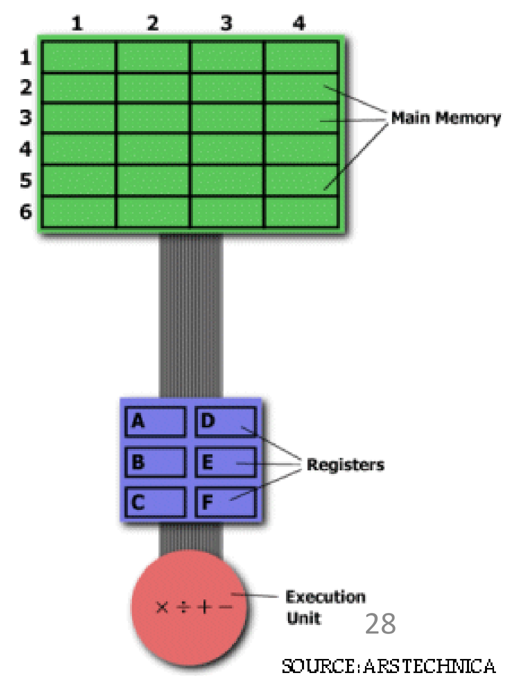
- CISC
  - Complex set of instructions like (CISC)

- RISC
  - Instruction execute in single cycle

  - Architecture which reduces the chip complexity by simpler processing instructions.
  - RISC architecture CPUs capable of executing only a very limited (simple) set of instructions.

CISC & RISC

- CISC Approach
  - Command: MULT 2:3, 5:2
    - Complete the task in few assembly line code
    - TASK multiply 2:3, 5:2 locations numbers
    - put output in 5:2 location
      - complex instruction.
        - » does`t complete in one cycle execution.



CISC & RISC

- RISC Approach
  - RISC processors only use simple instructions
  - Executed within one clock cycle.
  - "MULT" command divided into three separate commands:
    - LOAD A, 2:3
    - LOAD B, 5:2
    - PROD A, B
    - STORE 2:3, A
- Single Cycle Execution

CISC & RISC

- Reduced Instruction Set Computers Advantage
  - Fast Execution of Instructions due to simple instructions for CPU.
  - RISC chips require fewer transistors, which makes them cheaper to design and produce.
  - Single-clock, reduced instruction only Register to register:
    - "LOAD" and "STORE" are independent instructions

CISC & RISC

CISC	RISC
Complex instruction taking multiple cycles	Simple instructions taking 1 cycle
Any instruction may reference memory	Only Load/Store reference memory
Not pipelined or less pipelined	Highly pipelined
Variable format instructions	Fixed format instructions

Microprocessor vs. Microcontroller

- Microprocessor is an IC which has only the CPU inside them.
  - These microprocessors don't have RAM, ROM, and other peripheral on the chip.
  - A system designer has to add them externally to make them functional.
  - Application of microprocessor includes Desktop PC's, Laptop, etc.
  - Microprocessor find applications where tasks are unspecific like developing software, games, websites, photo editing, creating documents etc.
  - In such cases the relationship between input and output is not defined. They need high amount of resources like RAM, ROM, I/O ports etc.
- Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip.
  - At times it is also termed as a mini computer or a computer on a single chip.
  - Microcontrollers are designed to perform specific tasks.
    - Specific means applications where the relationship of input and output is defined. Depending on the input, some processing needs to be done and output is delivered.
  - Since the applications are very specific, they need small resources like RAM, ROM, I/O ports etc and hence can be embedded on a single chip.
  - This in turn reduces the size and the cost.