Unit 1

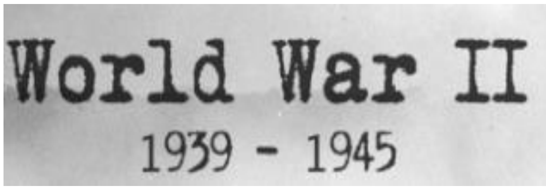
Introduction to Microprocessor

===

Definition of Computer

- ✓ Computer is a programmable machine.
- ✓ Computer is a machine that manipulates data according to a list of instructions.
- ✓ Computer is any device which aids humans in performing various kinds of computations or calculations.

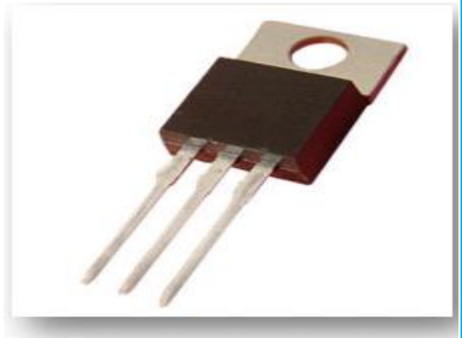
There are five generations of computer:



- ✓ **First generation – 1946 – 1958**
- ✓ **Second generation – 1959 – 1964**
- ✓ **Third generation – 1965 – 1970**
- ✓ **Fourth generation – 1971 – today**
- ✓ **Fifth generation – Today to future**

The first Generation computers used vacuum tubes for circuitry and magnetic drums for memory, and were often enormous, taking up entire rooms. • They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions.

Vacuum tube


Transistor

The Second Generation • Transistors replaced vacuum tubes and ushered in the second generation of computers. • One transistor replaced the equivalent of 40 vacuum tubes. • Allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable. • Still generated a great deal of heat that can damage the computer.

The Second Generation • Second-generation computers moved from cryptic binary machine language to symbolic, or assembly, languages, which allowed programmers to specify instructions in words. • Second-generation computers still relied on punched cards for input and printouts for output. • These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology.


Integrated Circuit

The Third Generation • The development of the integrated circuit was the hallmark of the third generation of computers. • Transistors were miniaturized and placed on silicon chips, called semiconductors, which drastically increased the speed and efficiency of computers. • Much smaller and cheaper compare to the second generation computers. • It could carry out instructions in billionths of a second.

The Third Generation • Users interacted with third generation computers through keyboards and monitors and interfaced with an operating system, which allowed the device to run many different applications at one time with a central program that monitored the memory. • Computers for the first time became accessible to a mass audience because they were smaller and cheaper than their predecessors.

Microprocessor

The Fourth Generation • The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip. • As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet. • Fourth generation computers also saw the development of GUIs, the mouse and handheld devices.

The Fifth Generation • Based on Artificial Intelligence (AI). • Still in development. • The use of parallel processing and superconductors is helping to make artificial intelligence a reality. • The goal is to develop devices that respond to natural language input and are capable of learning and self-organization. • There are some applications, such as voice recognition, that are being used today.

Analog and digital computer

Digital computer is that computer, which performs and accepts the converted binary number data. Today most of the computers are digital computers. It is basically an electronic based computer. Rather than continuous its input functions works discretely. Input can be in form of letters, numbers written and represented in special binary coded languages. Microcomputers, personal computers, network servers, super computers and multi-processor computers are kinds of digital computer. It uses two binary code or digits 0 and 1, which stands for no and yes, false and true, left and right. All instructions given to computer are performed through these two digits.

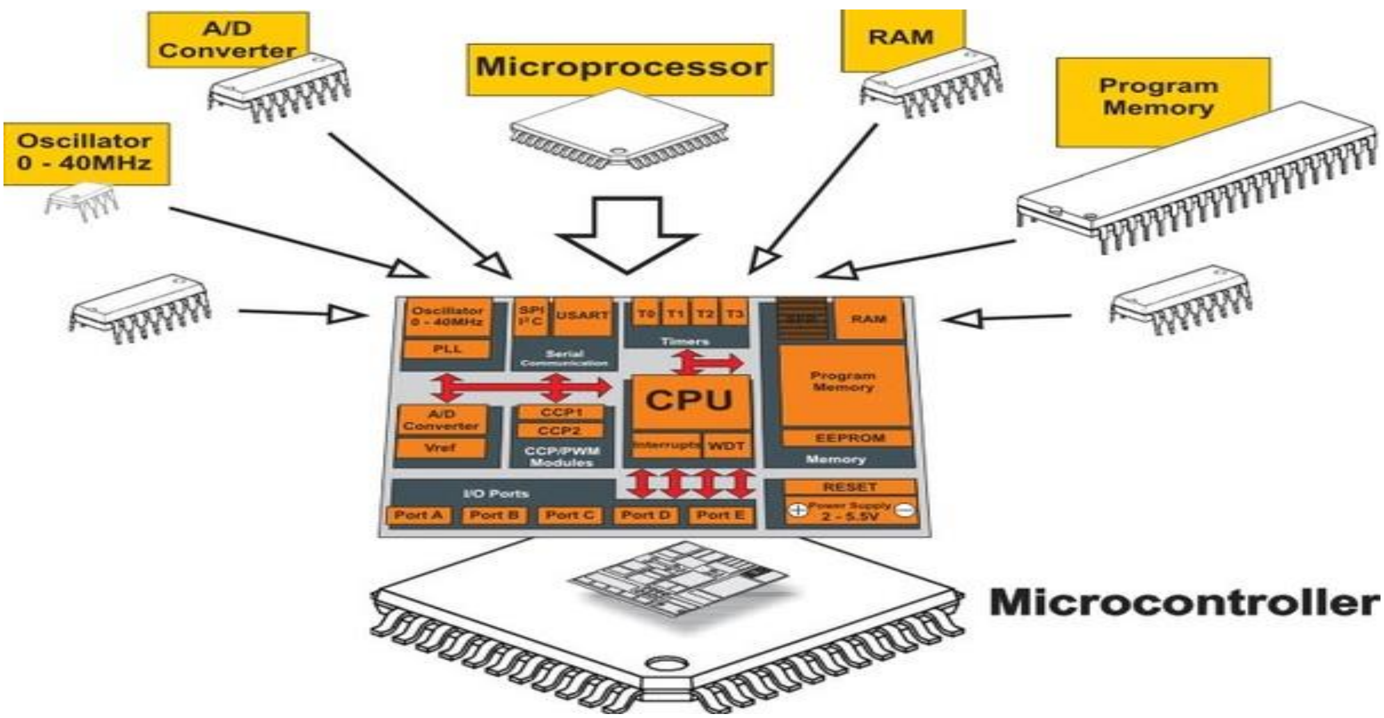
Analog computer is that computer, which performs the functions continuously irrespective of variations in input, which can be in form of fluctuation in voltage or temperature, electrical or mechanical parts. Oscilloscopes, thermostat and thermometer are examples of analog computer. In start analog computer was used to measure the electrical signal, current, frequency of signal and voltage, resistance of capacitor, etc. Working with analog computer is not so easy. Number of electrical and mechanical peripheral and supporting devices are required for it to function properly.

Microprocessor, microcomputer, microcontroller

Microcontroller and Microprocessors Are Integrated Circuits with several digital circuits inside them. It's hard to decide about any IC just by looking if it's a Microcontroller or Microprocessor, and often they are mixed together, because they both can be programmed in the similar way. One needs to check out their specifications in order to distinguish between.

Microcontroller: They are a whole complete system in themselves (*maybe sometimes specific Operating System too*). They consist of Processing Unit, RAM, ROM, other external peripherals. They generally have Harvard Architecture. The processing speed is comparatively slow. The addresses in programming are placed outside the IC. Eg: Arduino, 8051.

Microprocessor: They only consist of the Processing Unit. They *don't* have peripherals included in them. They generally have Von Neumann Architecture. The processing speed is comparatively faster. The addresses in programming have their own space. Eg: Raspberry pi.



Microprocessor is used in applications where *task is not predefined* or where *intensive processing* is required while Microcontroller is used for *specific propose*.

microprocessor chip contain only CPU(central processing unit),all other elements like memory are connected to it while in Microcontroller all elements including CPU are in one chip.

A microcomputer is a small, relatively inexpensive computer with a microprocessor as its central processing unit (CPU). It includes a microprocessor, memory, and minimal input/output (I/O) circuitry mounted on a single printed circuit board. Microcomputers became popular in the 1970s and 1980s with the advent of increasingly powerful microprocessors.

The predecessors to these computers, mainframes and minicomputers, were comparatively much larger and more expensive (though indeed present-day mainframes such as the IBM System z machines use one or more custom microprocessors as their CPUs). Many microcomputers (when equipped with a keyboard and screen for input and output) are also personal computers (in the generic sense).

Stored program concept and von-Neumann's architecture

Stored-program concept: Storage of instructions in computer memory to enable it to perform a variety of tasks in sequence or intermittently. The idea was introduced in the late 1940s by John von Neumann, who proposed that a program be electronically stored in binary-number format in a memory device so that instructions could be modified by the computer as determined by intermediate computational results. Other engineers, notably John W. Mauchly and J. Presper Eckert, contributed to this idea, which enabled digital computers to become much more flexible and powerful. Nevertheless, engineers in England built the first stored-program computer, the Manchester Mark I, shortly before the Americans built EDVAC, both operational in 1949.

Von Neumann Architecture:

John von Neumann, original name **János Neumann**, (born December 28, 1903, Budapest, Hungary—died February 8, 1957, Washington, D.C., U.S.), Hungarian-born American mathematician. As an adult, he appended *von* to his surname; the hereditary title had been granted his father in 1913. Von Neumann grew from child prodigy to one of the world's foremost mathematicians by his mid-twenties. Important work in set theory inaugurated a career that touched nearly every major branch of mathematics. Von Neumann's gift for applied mathematics took his work in directions that influenced quantum theory, automata theory, economics, and defense planning. Von

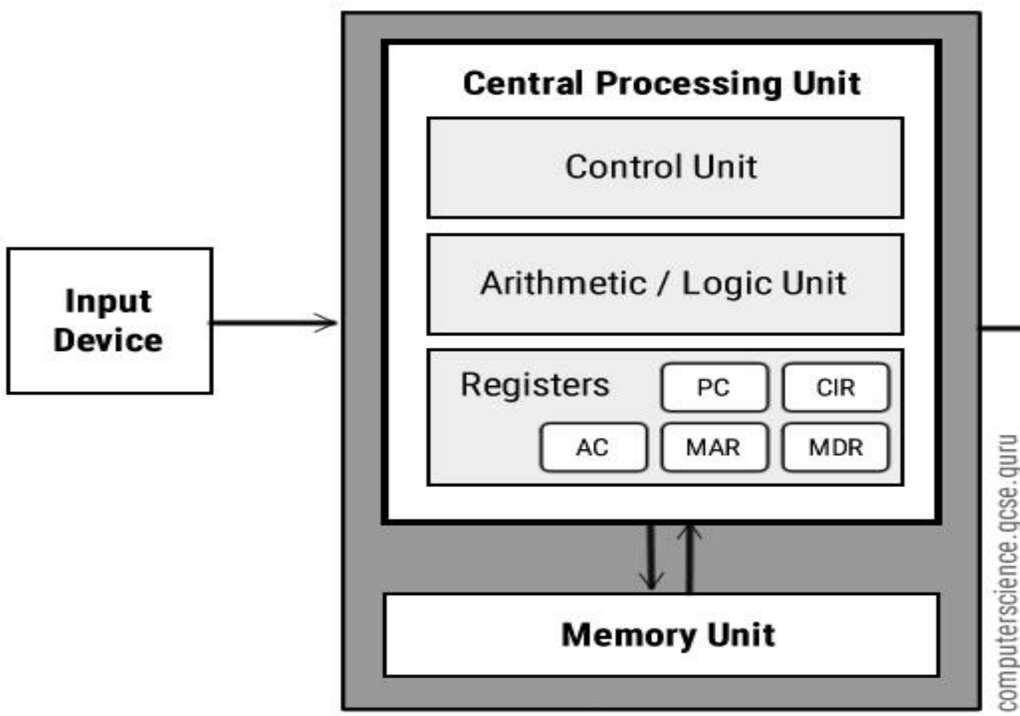
Neumann pioneered game theory and, along with Alan Turing and Claude Shannon, was one of the conceptual inventors of the stored-program digital computer.

Von Neumann architecture was first published by John von Neumann in 1945. His computer architecture design consists of a Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, Registers and Inputs/Outputs.

Von Neumann architecture is based on **the stored-program computer concept**, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today.

Central Processing Unit (CPU): The Central Processing Unit (CPU) is the electronic circuit responsible for executing the instructions of a computer program.

It is sometimes referred to as the microprocessor or processor. The CPU contains the ALU, CU and a variety of registers.



Registers: Registers are high speed storage areas in the CPU. All data must be stored in a register before it can be processed.

MAR Memory Address Register: Holds the memory location of data that needs to be accessed

MDR Memory Data Register: Holds data that is being transferred to or from memory

AC Accumulator: Where intermediate arithmetic and logic results are stored

PC Program Counter: Contains the address of the next instruction to be executed

CIR Current Instruction Register: Contains the current instruction during processing

Arithmetic and Logic Unit (ALU) allows arithmetic (add, subtract etc) and logic (AND, OR, NOT etc) operations to be carried out.

Control unit controls the operation of the computer's ALU, memory and input/output devices, telling them how to respond to the program instructions it has just read and interpreted from the memory unit. The control unit also provides the timing and control signals required by other computer components.

Buses are the means by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory. A standard CPU system bus is comprised of a control bus, data bus and address bus.

Address Bus Carries the addresses of data (but not the data) between the processor and memory. Data Bus Carries data between the processor, the memory unit and the input/output devices. Control Bus Carries control signals/commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer.

Memory Unit : The memory unit consists of RAM, sometimes referred to as primary or main memory. Unlike a hard drive (secondary memory), this memory is fast and also directly accessible by the CPU.

RAM is split into partitions. Each partition consists of an address and its contents (both in binary form).

The address will uniquely identify every location in the memory.

Loading data from permanent memory (hard drive), into the faster and directly accessible temporary memory (RAM), allows the CPU to operate much quicker.

General architecture of a microcomputer system showing control buses

or

The concept and architecture of a microcomputer

A microcomputer is a computer built on the basis of a microprocessor i.e. a processor implemented as an integrated circuit. Since all processors are now produced in the form of integrated circuits, we can say that all computers are microcomputers. The general method for constructing microcomputers consists in connecting to the microprocessor busses additional sub-systems such as memories and peripheral device controllers (input/output units).

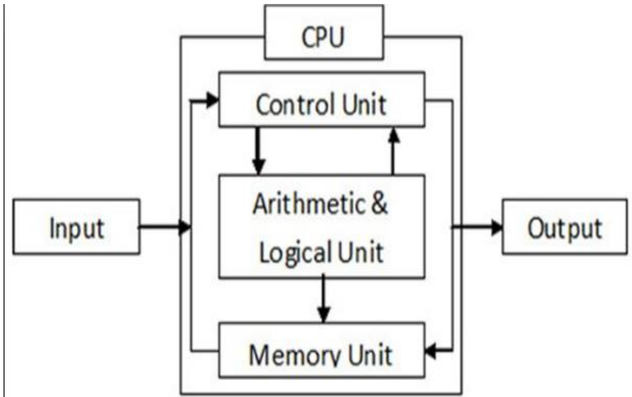


Fig. Block Diagram of Computer

The basic block diagram of a simple microcomputer is shown in the figure below. We can see there a microprocessor with three its busses going out: data bus, address bus and control bus. To these busses, the following devices are connected: operational memory composed of RAM (Random Access Memory) and ROM (Read Only Memory) memories, as well as input/output units to which peripheral devices are connected.

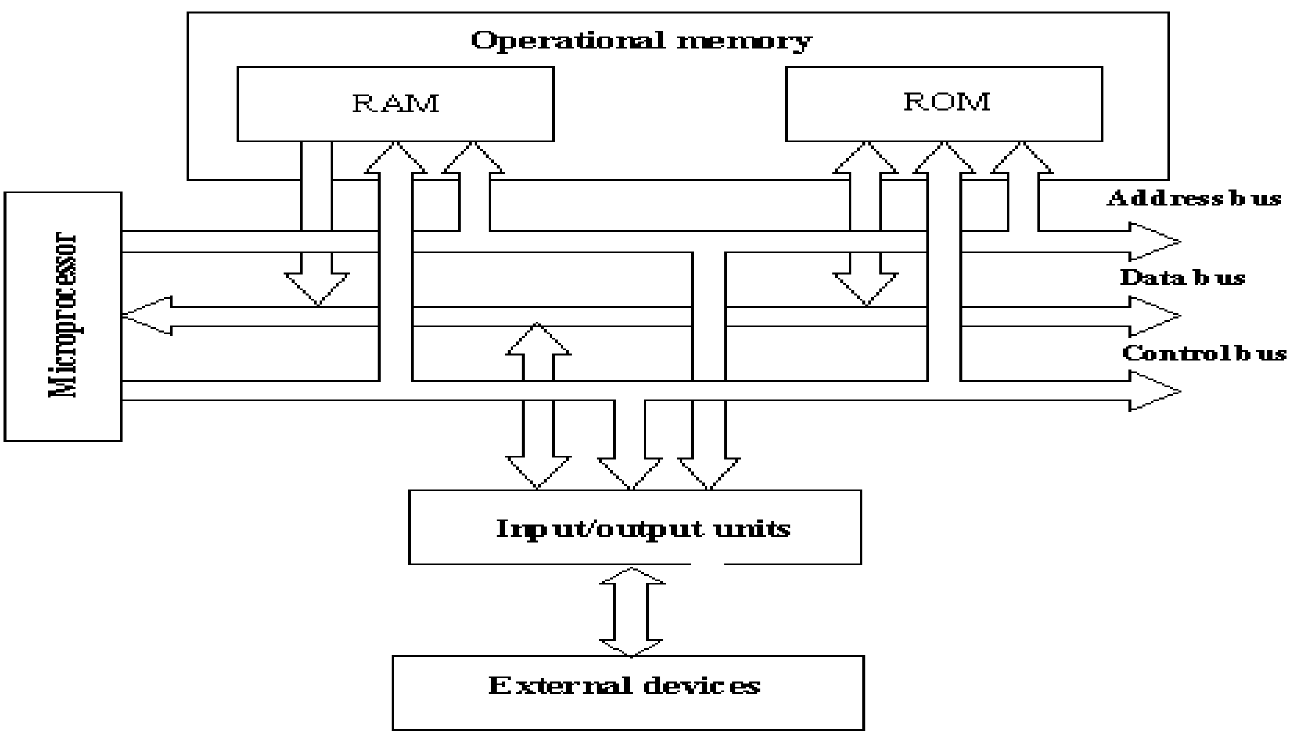


Fig: Simplified general scheme of a simple microcomputer

History of x86 microprocessors

(By Timeline Basis)

Here's a peek at the events and technologies that led to the development of Intel's x86 architecture, plus milestones in its 30-year reign.

1947: The transistor is invented at Bell Labs.

1965: Gordon Moore at Fairchild Semiconductor observes in an article for *Electronics* magazine that the number of transistors on a semiconductor chip doubles every year (Next Semester in Operating System). For microprocessors, it will double about every two years for more than three decades.

1968: Moore, Robert Noyce and Andy Grove found Intel Corp. to pursue the business of "Integrated Electronics."

Intel co-founder Gordon Moore

1969: Intel announces its first product, the world's first metal oxide semiconductor (MOS) static RAM, the 1101. It signals the end of magnetic core memory.

1971: Intel launches the world's first microprocessor, the 4-bit 4004, designed by Federico Faggin.

The Intel 4004

The 2,000-transistor chip is made for a Japanese calculator, but a far sighted Intel ad calls it "a micro programmable computer on a chip."

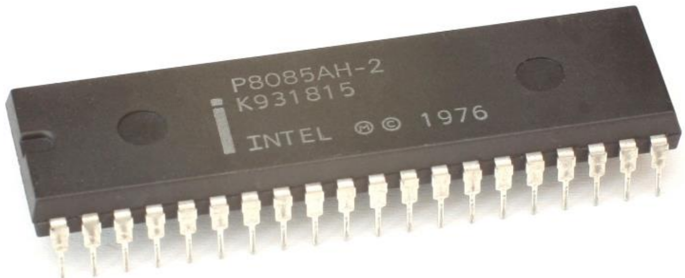
1972: Intel announces the 8-bit 8008 processors. Teenagers Bill Gates and Paul Allen try to develop a programming language for the chip, but it is not powerful enough.

The Intel 8080 (GNU FDL 1.2)

1974: Intel introduces the 8-bit 8080 processors, with 4,500 transistors and 10 times the performance of its predecessor.

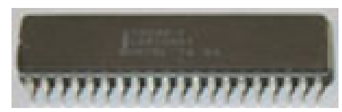
1975: The 8080 chip finds its first PC application in the Altair 8800, launching the PC revolution. Gates and Allen succeed in developing the Altair Basic language, which will later become Microsoft Basic, for the 8080.

1976: **Intel 8085** is an 8-bit microprocessor. It is a software-binary compatible with the more-famous Intel 8080 with only two minor instructions added to support its added interrupt and serial input/output features.



1976: The x86 architecture suffers a setback when Steve Jobs and Steve Wozniak introduce the Apple II computer using the 8-bit 6502 processors from MOS Technology. PC maker Commodore also uses the Intel competitor's chip.

1978: Intel introduces the 16-bit 8086 microprocessors. It will become an industry standard.



The Intel 8086 *(GNU FDL 1.2)*

1979: Intel introduces a lower-cost version of the 8086, the 8088, with an 8-bit bus.

1980: Intel introduces the 8087 math co-processor.

1981: IBM picks the Intel 8088 to power its PC. An Intel executive would later call it "the biggest win ever for Intel."

The IBM PC *(GNU FDL 1.2)*

1982: IBM signs Advanced Micro Devices as second source to Intel for 8086 and 8088 microprocessors.

1982: Intel introduces the 16-bit 80286 processors with 134,000 transistors.

1984: IBM develops its second-generation PC, the 80286-based PC-AT. The PC-AT running MS-DOS will become the de facto PC standard for almost 10 years.

1985: Intel exits the dynamic RAM business to focus on microprocessors, and it brings out the 80386 processor, a 32-bit chip with 275,000 transistors and the ability to run multiple programs at once.

1986: Compaq Computer leapfrogs IBM with the introduction of an 80386-based PC.

1987: VIA Technologies is founded in Fremont, Calif., to sell x86 core logic chip sets.

1989: The 80486 is launched, with 1.2 million transistors and a built-in math co-processor. Intel predicts the development of multicore processor chips some time after 2000.

The Intel 80386 *(GNU FDL 1.2)*

Late 1980s: The complex instruction set computing (CISC) architecture of the x86 comes under fire from the rival reduced instruction set computing (RISC) architectures of the Sun Sparc, the IBM/Apple/Motorola PowerPC and the MIPS processors. Intel responds with its own RISC processor, the i860.

The AMD Am486, an Intel 486 competitor *(GNU FDL 1.2)*

1990: Compaq introduces the industry's first PC servers, running the 80486.

1993: The 3.1 million transistor, 66-MHz Pentium processor with superscalar technology is introduced.

1994: AMD and Compaq form an alliance to power Compaq computers with Am486 microprocessors.

1995: The Pentium Pro, a RISC slayer, debuts with radical new features that allow instructions to be anticipated and executed out of order. That, plus an extremely fast on-chip cache and dual independent buses, enable big performance gains in some applications.

1997: Intel launches its 64-bit Epic processor technology. It also introduces the MMX Pentium for digital signal processor applications, including graphics, audio and voice processing.

Intel's Pentium Pro *(GNU FDL 1.2)*

1998: Intel introduces the low-end Celeron processor.

1999: VIA acquires Cyrix Corp. and Centaur Technology, makers of x86 processors and x87 co-processors.

2000: The Pentium 4 debuts with 42 million transistors.

2003: AMD introduces the x86-64, a 64-bit superset of the x86 instruction set.

2004: AMD demonstrates an x86 dual-core processor chip.

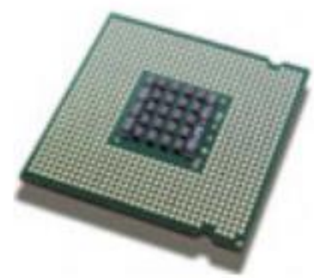
AMD64, a rebranding of x86-64

2005: Intel ships its first dual-core processor chip.

2005: Apple announces it will transition its Macintosh computers from PowerPCs made by Freescale (formerly Motorola) and IBM to Intel's x86 family of processors.

2005: AMD files antitrust litigation charging that Intel abuses "monopoly" to exclude and limit competition. (The case is still pending in 2008.)

2006: Dell Inc. announces it will offer AMD processor-based systems.



Intel's first dual-core chip, the Pentium D

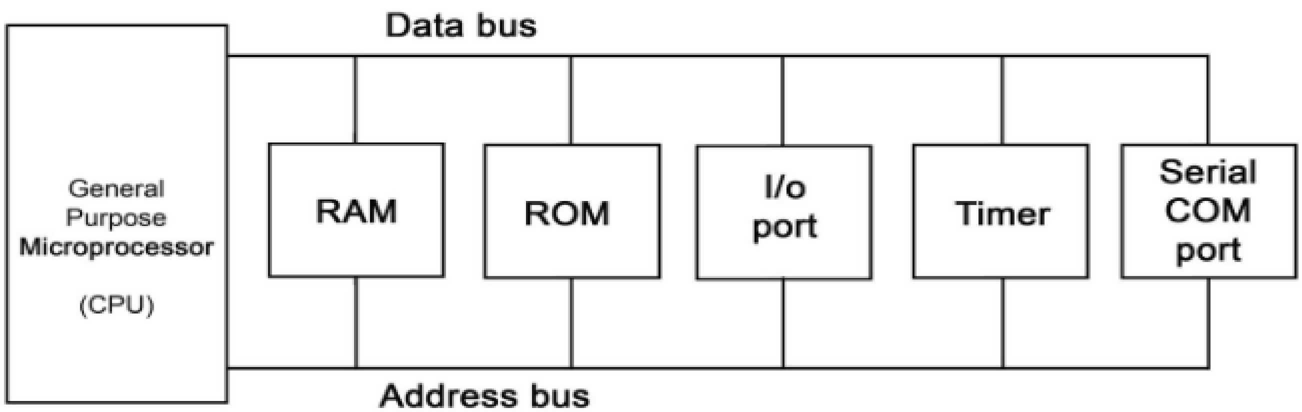
NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

Evolution of Microprocessors

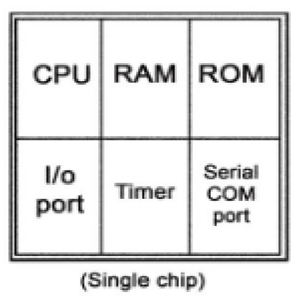
Block diagram of a typical microprocessor and microcontroller

Microcontroller contains : CPU, RAM, ROM, I/O ports, and, counter, timers etc is on a single chip.
Microprocessor contains: CPU stands alone, RAM, ROM, I/O ports, and, counter, timers. etc.

(a). GENERAL PURPOSE MICROPROCESSOR SYSTEM



(b). MICROCONTROLLER



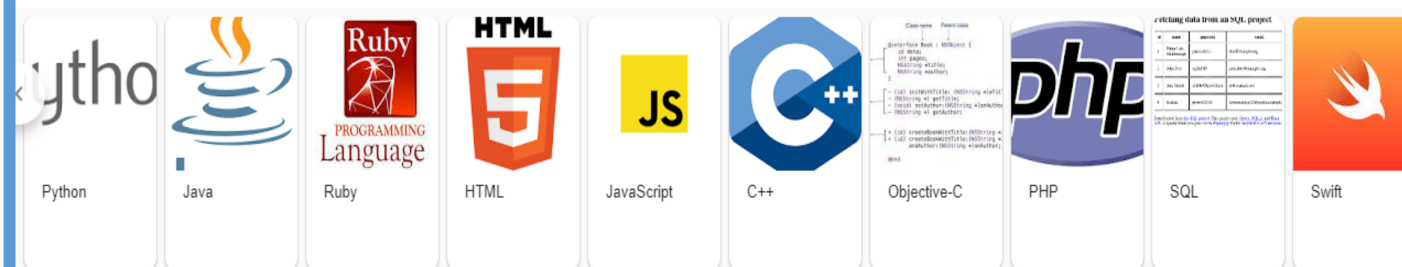
Programming languages

Compiler: It's a computer program(s) that transforms source code written in a programming language into machine language that is the target language which usually has a binary form known as object code.

Interpreter: It translates high level instructions into an intermediate form, it translates the code into the intermediate form line by line a caries out specific actions.

Assembler: It is a program that takes basic computer instruction(s) and converts then into a pattern of bits that the computer's processor can use to perform its basic operations. The language used to program the assembler is called assembly language.

- Compiler Takes **Entire** program as input whereas Interpreter Takes **Single** instruction as input.
- Intermediate Object Code is **Generated** in case of compiler whereas in case of interpreter **No** Intermediate Object Code is **Generated.**
- **Memory Requirement** is **More**(Since Object Code is Generated) in case of compiler whereas **Memory Requirement** is **Less** in case of interpreter.
- **Errors** are displayed after **entire program** is checked in case of compiler. Hence debugging is comparatively hard. In case of an interpreter, **Errors** are displayed for **every instruction** interpreted (if any). An interpreter continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.



High Level Languages: are written in a form that is close to our human language, enabling to programmer to just focus on the problem being solved.

No particular knowledge of the hardware is needed as high level languages create programs that are portable and not tied to a particular computer or microchip.

These programmer friendly languages are called 'high level' as they are far removed from the code instructions understood by the computer.

Examples include: C++, Java, Pascal, Python, Visual Basic.

Advantages

- Easier to modify as it uses English like statements
- Easier/faster to write code as it uses English like statements
- Easier to debug during development due to English like statements
- Portable code – not designed to run on just one type of machine

Low level languages are used to write programs that relate to the specific architecture and hardware of a particular type of computer.

They are closer to the native language of a computer (binary), making them harder for programmers to understand.

Low level refers to:

- Assembly Language
- Machine Code

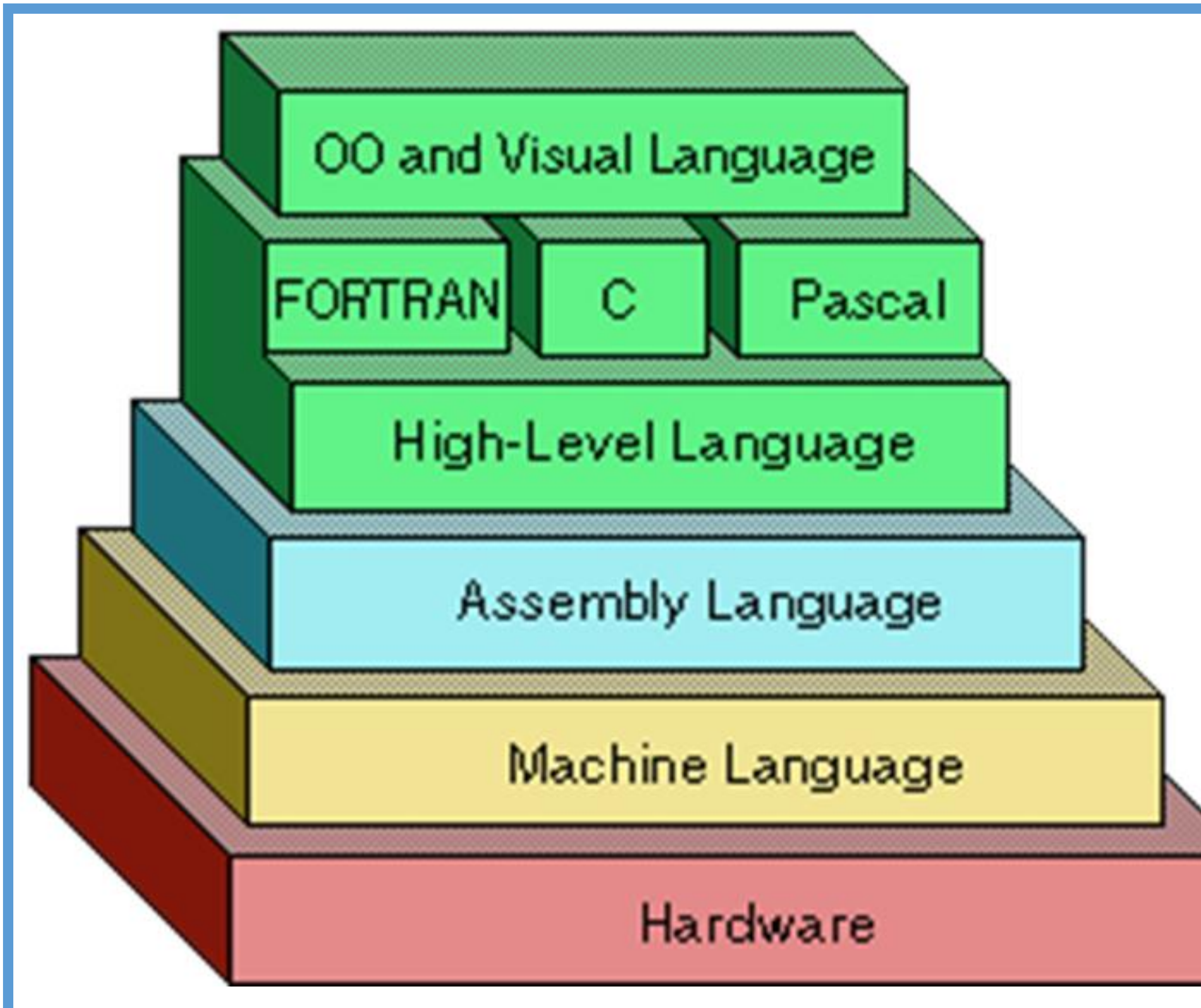
Assembly Language: Few programmers write programs in low level assembly language, but it is still used for developing code for specialist hardware, such as device drivers.

It is easy distinguishable from a high level language as it contains few recognizable human words but plenty of mnemonic code.

Advantages

- Can make use of special hardware or special machine-dependent instructions (e.g. on the specific chip)
- Translated program requires less memory
- Write code that can be executed faster
- Total control over the code
- Can work directly on memory locations

Machine Code: Programmers rarely write in machine code (binary) as it is difficult to understand.



Introduction to Simple as Possible (SAP) computers

The Simple-As-Possible (SAP)-1 computer is a very basic model of a microprocessor explained by Albert Paul Malvino. The SAP-1 design contains the basic necessities for a functional Microprocessor. Its primary purpose is to develop a basic understanding of how a microprocessor works, interacts with memory and other parts of the system like input and output.

SAP 1:

- W bus – A single 8-bit bus for address and data transfer.
- 16 Bytes memory (RAM). R
- registers are accumulator and B-register each of 8 bits.
- Program counter – initializes from 0000 to 1111 during program execution.

- Memory Address Register (MAR) to store memory addresses.
- Adder/ Subtracted for addition and subtraction instructions.
- A Control Unit
- A Simple Output.
- 6 machine state reserved for each instruction
- The instruction format of SAP-1 Computer is (XXXX) (XXXX)

Differences between SAP-1 and SAP-2 Architecture

SAP-1	SAP-2
It has 8-bit bus.	It has 16-bit bus.
PC is 4-bit.	PC is 16-bit.
It does not have hexadecimal keyboard encoder.	It has hexadecimal keyboard encoder.
It has single input.	It has two input ports.
MAR receives 4-bit address from PC.	MAR receives 16-bit address from PC.
It does not have ROM.	It has 2 KB ROM.
It has 16 Byte memory.	It has 62 KB memory.
It does not have MDR.	It has MDR.
It has only adder/subtractor.	It has ALU.
It does not have flag.	It has 2 flags.
It does not have temporary register.	It has temporary register.
It has single register (B).	It has 2 registers (B and C).
It has single output port.	It has 2 output ports.
It has 5 instruction sets.	It has 42 instruction sets.

SAP-3

- architectural differences with SAP-1, SAP-2
- registers and move instructions
- arithmetic instructions and carry flag
- multi-precision addition and subtraction

- rotate and compare
- arithmetic, logic immediate instructions
- jump instructions
- 16-bit register pairing and 16-bit load, arithmetic
- indirect instructions

Syllabus

Microprocessors

EG2107 EX

Total: 7 hour /week

Year: II | **Lecture: 3 hours/week**
Semester: I | **Tutorial: 1 hours/week**
Practical : 3 hours/week

Course Description:

This course deals with fundamentals of microprocessor, basic low level microprocessor programming, interfacing and introduction to basic programmable devices.

Course Objectives:

After completing this course the students will be able to:

1. understand the working principle of a computer
2. understand the working principle of microprocessor
3. understand the process of writing and executing low level language
4. know how to interface devices with a computer

Course Contents:

Unit1. Introduction to Microprocessor: [8]

- 1.1. History of computer development
- 1.2. Analog and digital computer
- 1.3. Microprocessor, microcomputer, microcontroller
- 1.4. Stored program concept and von-Neumann's architecture
- 1.5. General architecture of a microcomputer system showing control buses
- 1.6. History of x86 microprocessors
- 1.7. Block diagram of a typical microprocessor and microcontroller
- 1.8. Programming languages
- 1.9. Instruction set of microprocessors
- 1.10. Introduction to Simple as Possible (SAP1,SAP2,SAP3) computers

Unit2. Microprocessor architecture and the instruction set: [8]

- 2.1. Internal architecture of 8085 microprocessor
- 2.2. Instruction and data formats
- 2.3. Instruction classifications
- 2.4. Addressing modes in 8085
- 2.5. 8085 Instruction set

Unit3. Assembly language programming for 8085: [9]

- 3.1. Introduction to assembly language and assemblers
- 3.2. Simple assembly language programs
- 3.3. Programs using loops, counters, delays
- 3.4. Table processing
- 3.5. Subroutine and stack
- 3.6. Code conversion ASCII/BCD/Binary

Activate Winc

Unit4. Interfacing I/O and memory devices: [10]

4.1. 8085 machine cycles and bus timing

- Fetch and execute cycles

- Memory read/write machine cycle
- I/O read/write machine cycle

4.2. Address Decoding

- Unique and non-unique address decoding
- Address decoding for I/O and memory devices

4.3. Interfacing I/O devices

- Interfacing Input Devices
- Interfacing Output Devices
- Address decoding using block decoders
- Interfacing Memory-mapped I/O

4.4. Memory Interfacing

- Memory structure and its requirement
- RAM and ROM chips
- Address decoding using NAND and block decoders

4.5. Direct memory access

Unit5. 8085 Interrupt processing: [6]

5.1. Programmed I/O

5.2. Interrupt Driven I/O

5.3. The 8085 Interrupt

5.4. 8085 Vectored Interrupts

5.5. Restart and software instructions

Unit6. Introduction to general purpose programmable peripheral devices: [4]

6.1. 8255 Programmable Peripheral Interface

6.2. 8254(8253) Programmable Interval Timer

6.3. 8259 Programmable Interrupt Controller

6.4. 8251 USART

Practical: [45]

The practical exercise shall cover the low level program from simple programs for data transfer to complex programs for table processing

1. Basics of microcomputer system through the 8085 microprocessor trainer kit
2. Programs that uses data transfer instructions
3. Programs that uses arithmetic instructions
4. Programs that uses logical instructions
5. Programs with conditional and unconditional branching
6. Programs with conditional and unconditional subroutine call and stack
7. Programs involving loops and counters
8. Programs that involves masking and checking numbers
9. Programs to manipulate table of numbers
10. Program for BCD and ASCII manipulation
11. Programs to perform multiplication and division
12. Programs to read and write from the port

Reference books:

1. Ramesh S. Gaonkar, "8085 Microprocessor programming and interfacing", New Age
2. John Uffenbeck, "The 8080, 8085 & Z-80 Programming, Interfacing and Troubleshooting", PHI
3. Albert Paul Malvino, Jerald A. Brown, "Digital Computer Electronics", McGraw-Hill