Unit-05

Refer to my notes

Name of the Subject

→Microprocessor Architecture (MA)

Name of the reference Book:

Microprocessors Architecture, Programming & Applications with the 8085

Author

Ramesh Gaonkar

Publisher: PENRAM

Edition: 5th 2012

(Unit I to IV)

Unit-01

LOGIC DEVICES

Part I of Unit-01
Microprocessor, Microcomputers & Assembly Lang

# Microprocessor

Is it a h/w or s/w?
Ans: h/w

1. It is a multipurpose, programmable, clock driven, register based electronic device that reads binary instructions from storage devices called memory, accepts binary data as input and processes data according to those instructions and provides results as output.

Micro → small Processor → to process (offline)
device → to do work (XYZ)
→ to perform a task
[based on the instructions given it
+, −, ×, etc]

see & feel
μP → mu (micro)

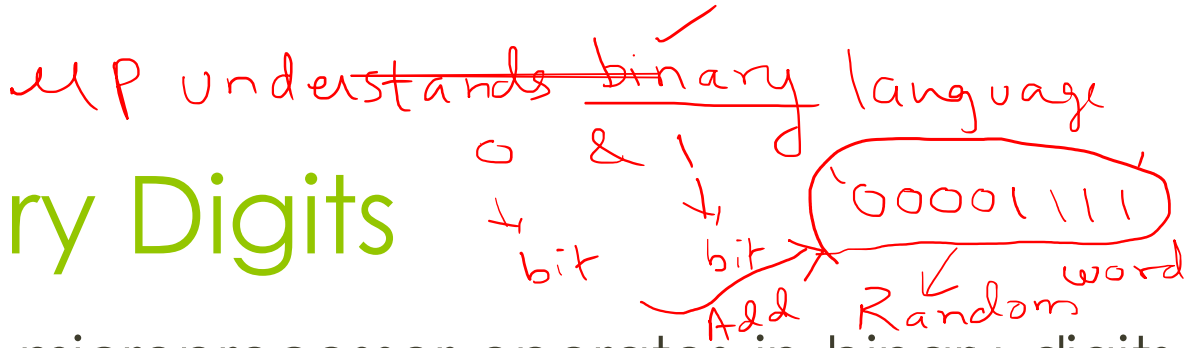
# Images of microprocessor

8085 µP




Intel
IBM
Atmel
Phillips
Motorala

# Binary Digits




1. The microprocessor operates in binary digits 0 and 1 known as bits.
2. Bit is an abbreviation fro the term binary digit.
3. The digits 0 and 1 are synonymous with low and high voltage respectively.
4. Each microprocessor recognizes and processes a group of bits called the word and they are classified according to their word length. Example: 8-bit processor.

8 bits = 1 byte

understand

8085 µP

length = 8 bits

A binary language is made of binary words.

# Q & A

1) Defn
A µP is a multipurpose, programmable, clock driven, register based electronic device which reads binary inst's from a storage device called memory, accepts binary data as I/P & processes data according to those inst's & provides results as O/P.

2) A µP is a h/w (h/w or s/w)

3) A µP understands binary language

4) A binary language is made up of binary words

5) 8085 µP is an 8 bit bit processor

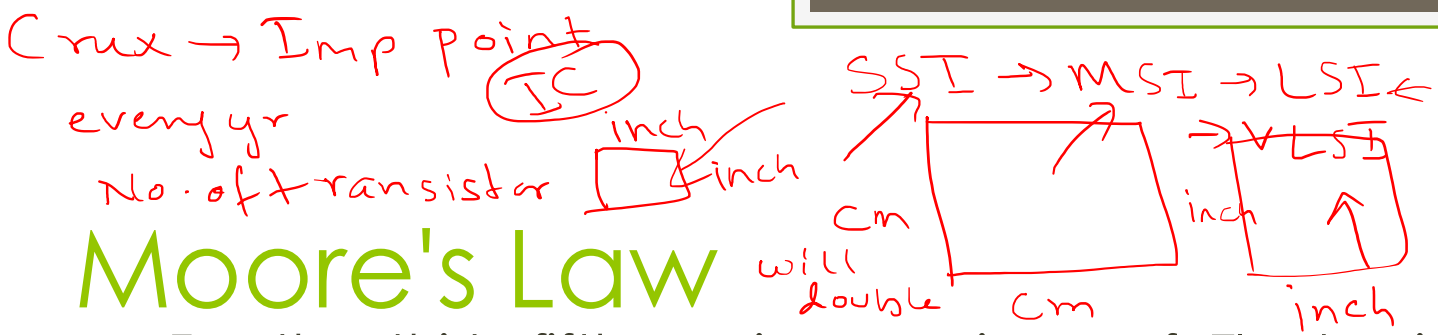
# A microprocessor as a Programmable Device

1. It means microprocessor can be instructed to perform a given task within its capability.
2. A programmer can select appropriate instructions and ask the microprocessor to perform various tasks on a given set of data. These instructions are simply patterns of 0s and 1s. The user selects instructions from the list and determines the sequence of execution for a given task. These instructions are entered in storage called memory, which can be read by the microprocessor.

# Intel Microprocessors- Historical Prospective

| Processor | Year of Introduction | Number of transistors | Initial clock speed | Address bus | Data Bus | Addressable memory |
|---|---|---|---|---|---|---|
| 4004 | 1971 | 2300 | 108KHz | 10 -bit | 4-bit | 640 bytes |
| 8085 | 1976 | 6500 | 5MHz | 16-bit | 8-bit | 64K |
| 80486 | 1989 | 1.2M | 25MHz | 32-bit | 32-bit | 4G |
| Pentium | 1993 | 3.1 M | 60MHz | 32-bit | 32/64-bit | 4G |
| Pentium 4 | 2000 | 42M | 1.4GHz | 36-bit | 64-bit | 64G |
| | | | | | | |

# Moore's Law




- For the thirty-fifth anniversary issue of Electronics Magazine, which was published on April 19, 1965, Gordon E. Moore, who was working as the director of research and development (R&D) at Fairchild Semiconductor at the time, was asked to predict what was going to happen in the semiconductor components industry over the next ten years. His response was a brief article entitled, *"Cramming more components onto integrated circuits"*.
- An observation made by Intel co-founder Gordon **Moore** in 1965. He noticed that the number of transistors per square inch on integrated circuits had doubled every year since their invention. **Moore's law** predicts that this trend will continue into the foreseeable future.

# Q & A

↳ 3 components: µP, Memory, I/O (Input-output devices)

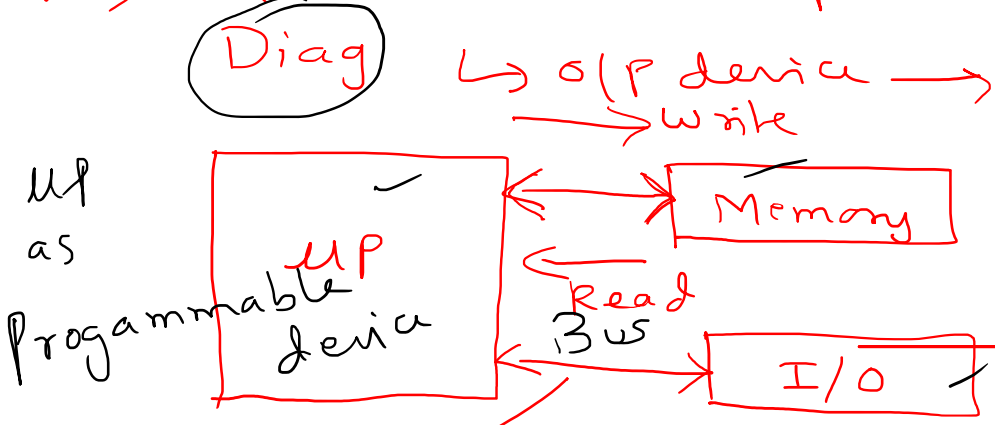
↳ µP → to process the data rcd from I/P devic as per the instructions stored in memory & generate the result & display the results on o/P devic

↳ Memory → to store data & instructions

↳ I/P devic → to provide I/P data

Diag

↳ o/P devic → display the result




µP can read from the memory

µP can write into the memory

µP can read from I/P devic & write to the o/P devic

# Organization of a Microprocessor Based System

1. For diagram no-1 refer to the White Board.
2. Explanation:

a) As seen from the diagram we can see the following are the parts of a microprocessor based system ( example microcomputer):

i) The microprocessor.
ii) Memory
iii) Input
iv) Output
v) System bus

# Organization of a Microprocessor Based System

b) The microprocessor is a clock driven SC device consisting of electronic circuits manufactured by using LSI or VLSI. It is capable of performing various computing functions and making decisions to change the sequence of program execution. The microprocessor can be divided into 3 parts:

1) **ALU ( Arithmetic and Logic Unit):** It performs various computing functions such as addition and subtraction and logic operations such as AND, OR and XOR.
2) **Register Array:** It consists of various registers identified by letters such as B.C.D.E.H AND L. These registers are used to store data temporarily during the execution of a program and are accessible to the user through instructions.
3) **Control Unit:** It provides the necessary timing and control signals to all the operations in the microcomputer. It controls the flow of data between the microprocessor and memory and peripherals.

# Organization of a Microprocessor Based System

c) **Memory:** it stores binary information as instructions and data. To execute programs the microprocessor reads the instructions and data from the memory and performs computing operations in ALU. Results are either transferred to the output section for display or stored in memory for later use.

The memory block consists of two sections: ROM and RAM.

The ROM is used to store programs that do not need any alterations. The monitor program of a single board microcomputer is stored in ROM. This program interprets the information entered through keyboard and provides equivalent binary digits to the microprocessor.

The RAM is called as user memory and is used to store user programs and data. In a single board microcomputers, the monitor program monitors the HEX keys and stores those instructions and data in R/W memory.

# I/O

d) It contains two types of devices: input and output, they are called as Peripherals.

The input devices such as keyboard, switches and A/D converter transfer binary information ( data and instructions) from outside world to the microprocessor.

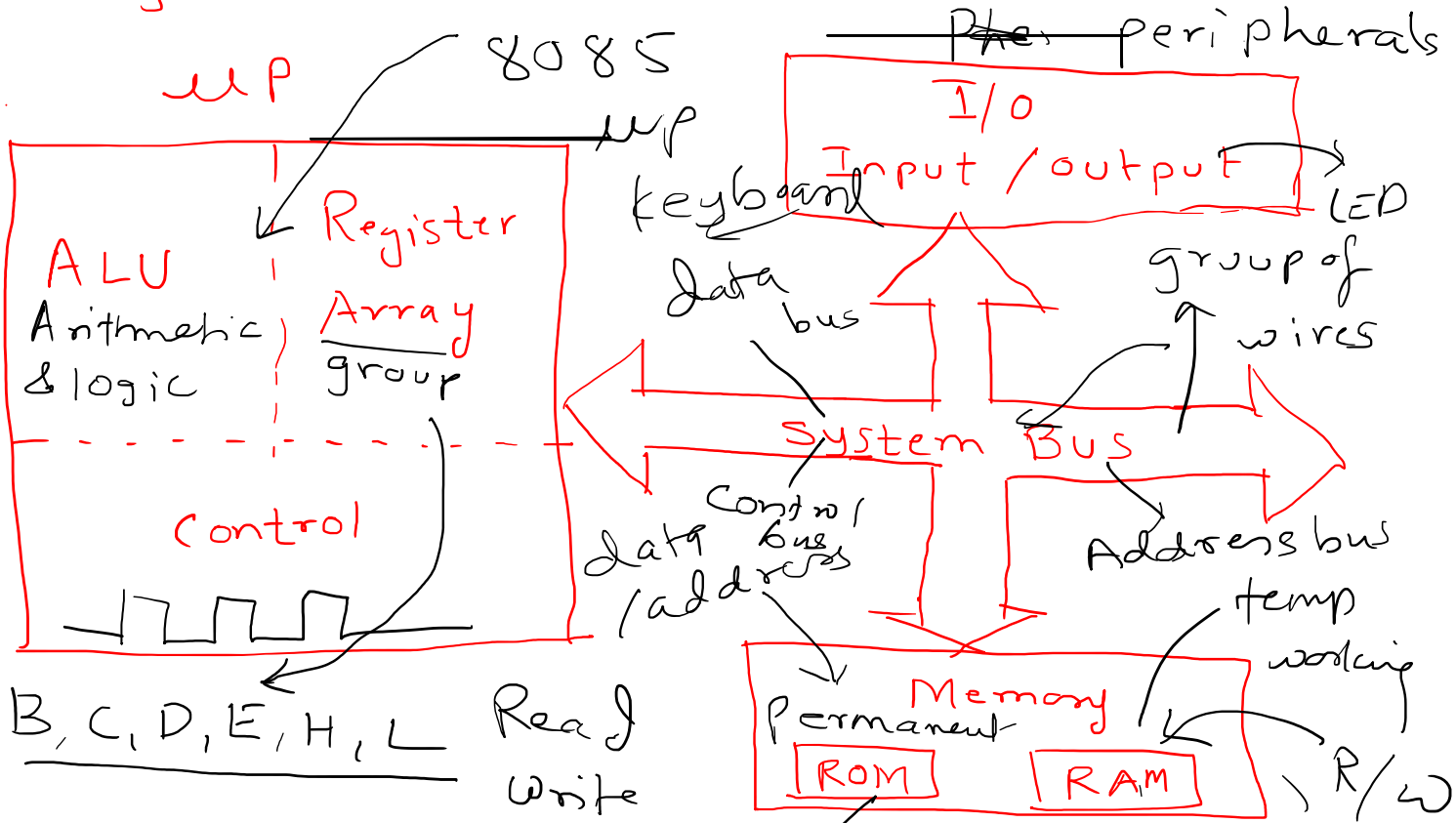
The keyboard can be ASCII or HEX. HEX keyboard has 16 data keys ( 0 to 9, A to F) and some additional function keys.

The output devices transfer data from the microprocessor to the outside world. These include LEDs, CRT, video screen, printer, X-Y plotter, magnetic tape of D/A converter.

# System Bus

e) **It is a communication path** between the microprocessor and peripherals. It is a group of wires to carry bits. All peripherals ( and memory) share the same bus, the microprocessor communicates with only one peripheral at a time. The timing is provided by the control unit of the microprocessor.

Diagram:




Orgⁿ of a μP based system

* Arithmetic operations → +, ×, ÷, − etc
  logical operations → OR, AND, NOT, XOR

# How does a microprocessor work?

1. The instructions and data are stored sequentially in the memory.
2. The microprocessor fetches the first instruction from its memory sheet, decodes it and executes that instruction.
3. The sequence of fetch, decode and execute is continued until the microprocessor comes across an instruction to stop.
4. During the entire process the microprocessor uses the system bus to fetch the binary instructions and data from the memory. It uses registers from the register section to store data temporarily and it performs computing function in the ALU section.
5. Finally it sends out the results in binary using the same bus lines , to the output devices.

Questions

1) Define & explain the term 'µp'

2) Explain µP as a programmable device

3) Explain the orgn of µP based system.

# Microprocessor Instruction Set and Computer Languages

1. Microprocessors recognize and operate in binary numbers. But each microprocessors has its own binary words, meanings and language.
2. A word is formed by combining number of bits for a given machine.
3. **The word is defined as the number of bits the microprocessor recognizes and processes at a time**.
4. A byte is defined as group of 8 bits.
5. Nibble stands for group of 4 bits.

# Microprocessor Instruction Set and Computer Languages

6. Machine language: a binary language that can be understood by a particular machine.

A machine language is made up of instructions and instructions are made up of a word or combination of words.

An instruction is a binary pattern entered through an input device in memory to command the microprocessor to perform that specific function.

The binary language with pre-determined instruction set is called as machine language.

# Microprocessor Instruction Set and Computer Languages

7. **Assembly Language**: Since it is difficult for the programmer to remember and use the machine language code , each manufacturer of a microprocessor has devised a symbolic code for each instruction , called mnemonic.

**A mnemonic for a particular instruction consist of letters that suggest the operation to be performed by that instruction**.

**A complete set of mnemonics for a particular microprocessor is called assembly language.**

Note that both machine language and assembly language are low level languages.

# Microprocessor Instruction Set and Computer Languages

8. A mnemonic can be written electronically on a computer using a program called as an Editor in the ASCII code and translated into binary code by using a program called as an assembler.

# Microprocessor Instruction Set and Computer Languages

9. **Programming languages that are intended to be a machine independent are called as HLL( High Level Language).** Examples: BASIC, PASCAL, C, C++, Java.

The programs written in HLL are converted into machine language by a program called as compiler or interpreter.

These programs accept English like statements as their input called as Source code. The compiler or the interpreter then translates the source code into the machine language compatible with the microprocessor being used in the system This translation in the machine language is called as object code.

Each microprocessor needs its own compiler or interpreter for each high level language.

# Microprocessor Instruction Set and Computer Languages

10. **Compilers and interpreters require** large memory space because an instruction in English requires several machine codes to translate it into binary. On other hand there is one-to-one correspondence between assembly language and machine code.

The assembly languages are compact and require less memory space. They are more efficient than the high level language programs.

The primary advantage of HLL is in troubleshooting ( debugging) programs.

# Microprocessor Instruction Set and Computer Languages

11. In certain applications such as traffic control and appliance control where programs are small and compact , assembly language is suitable.. Similarly in real time applications assembly language is highly desirable.

**Applications where programs are large and memory is not a limitation, HLL is desirable. Examples: word processors, video games, billing, accounting, etc.**