Dear students,

Welcome to the world of Microprocessors!

You are about to learn, how these little chips became the central force behind the computer revolution. Brain child of Alan Turing, nurtured by genius minds like Von Neumann, Maurice Wilkes, and materialized by corporations like Intel, Samsung, Apple… these chips transformed the way the world computed.

From traffic signals to air traffic control, drones to satellites, fitness bands to pace makers, Microprocessors have brought to you, ever improving standards of health, travel, entertainment, communication etc.

In this book, you begin with learning the 8086 Architecture.
You then learn 8086 in depth including its Memory module, Instruction set, Programming and interfacing with its set of coprocessors and peripherals.

You will then proceed to learning powerful microprocessors like Pentium.
You will see the speed advantage achieved in Pentium using superscalar architecture and on chip cache memories. You will then witness the brilliance of Branch Prediction Algorithm which minimized the biggest drawback of pipelining.

Before we start, allow me to take the opportunity to thank my mother,
Prof. Veena D. Acharya.
A teacher all her life, she was the primary inspiration for me to pursue this noble profession.
Her blessings are reflected in the enthusiasm shown in this book and in my lectures.

Bharat D. Acharya.
B. E. Computer Science.
Founder, Bharat Acharya Education.
Teaching Microprocessors & Microcontrollers since 2000.

INTRODUCTION TO MICROPROCESSORS

Drones, stump vision cameras, mobile phones, RFID sensors, autonomous cars, streaming servers, your favorite shopping website… all of these are fruits of a seed called **"Microprocessor"**, planted years ago in mid 1970s and in fact conceived even earlier in 1940s.

So what does this Microprocessor (henceforth called μP) actually do… Why do we need to learn it… And most importantly, after completing our education, how will this knowledge be useful to us…

Where do we use a μP? Is a μP used in a fan, or in a tube light, or in a switch board? No, none of them.

Is it used in a mobile phone, a computer, a microwave oven, a washing machine? Yes, all of them! This is because they run on programs, and all those programs are executed by the microprocessor within these devices. **This is the main function of a μP, to execute programs**.

In our day to day encounters we come across several devices and appliances. If you feel any of them works on a program, you should most certainly realize, it must contain a microprocessor. Take a microwave oven as an example. The μP inside the oven isn't directly cooking the food. It is running programs that are responsible for rotating the dish, maintaining the correct temperature, counting the desired number of seconds, displaying the time remaining on the screen, and finally ringing the sweet alarm (ding!) informing us that the cooking is complete. All of these require programs, that are executed by the oven's μP. And who writes the programs? The engineer, yes that's you! It is this this combination of the engineers mind and the microprocessors execution abilities that has transformed the world in the past few decades and will continue to do so as both are getting ever so smart in the new age.

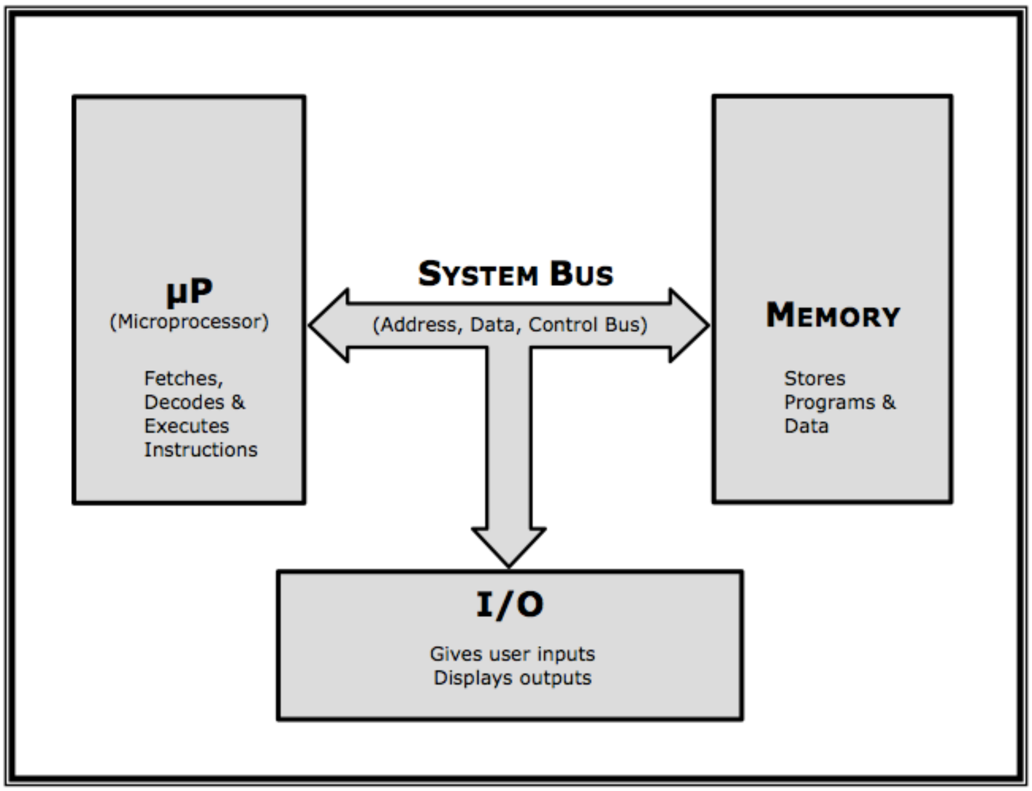
The simplest example of the use of a microprocessor, is in a **computer**. When you imagine a computer, lots of devices come to our mind, like the keyboard, mouse, printer, monitor and the big "box" also casually referred to by laymen as the CPU. Of course, you know their roles! The Keyboard and mouse are called input devices. Are they executing our programs? No! When we want to add two numbers, neither is the keyboard adding them nor is the mouse. So, are they required? Yes! To give inputs. That's their role. To provide inputs to the system. Similarly, the printer and the monitor are used to produce outputs. Hence these are called I/O devices (Input and Output devices). This leaves us with that big "box". Open it and you see an electric circuit board also called the motherboard. **On the motherboard, pretty much around the center lies a big chip, around the size of our palm, that's your μP**. You may notice in modern motherboards the μP is generally covered with a fan, to dissipate the heat generated by relentlessly performing millions and sometimes billions of operations per second. So how important is the μP? Well, remove it and our computer becomes a piece of junk! Every activity of your computer needs a program, lets get you in agreement with that. You watch a movie, play a song, surf the net, be on social media etc., all of these are programs. Executing them keeps the μP busy round the clock, hence the heat and the fan.

Let's say we are headed to the market right now, to buy the latest computer. Which microprocessor will you find inside? Yes, the Intel Core i7, or maybe Core i5 or i3, Intel also has recently released the Core i9 but its way too expensive to be mass produced as yet. So are we learning Core i7, i5, i3… not so early! These are a result of over 40 years of cutting-edge development making them way too advanced, to be understood by a beginner. We begin with learning the μP that started this whole x86 family… **The 8086 Microprocessor**.

BASIC ORGANIZATION OF A COMPUTER

A computer system, as we know it, consists of various components.
They can be broadly classified into three sections:
The Processor, Memory and I/O.

COMPUTER SYSTEM



THE PROCESSOR – "μP"

The heart of the computer is its μP (Microprocessor). Current generation computers use processors like Intel Core i3, i5 or i7 and so on. They have come a long way from the initial processors that you are about to learn E.g.: 8085, 8086 etc.

Back in the day (1940s), when micro-electronics was not invented, processors looked very different and were certainly not "micro" in appearance. They were created using huge arrays of physical switches which were operated manually and often occupied large rooms.

In the following decades, with the invention of micro-electronics, scientists managed to embed thousands of microscopic switches (transistors) inside a small chip, and called it a "**Micro-processor**".

Over the years, microprocessors grew in strength. From housing a few thousand transistors (8085) to containing more than a billion transistors (Core i7), the computational power has been increasing exponentially. Having said that, some of the basics still remain the same.

To put it simply, **the main function of a μP is to Fetch, Decode and Execute instructions.**

Instructions are a part of programs. Programs are stored in the memory. First, **μP fetches an instruction** from the memory. It then **decodes the instruction**. This means, it "understands" the binary pattern of the instruction, also called its opcode. Every instruction when stored in the memory is in its unique binary form, which indicates the operation to be performed. **This is called its opcode**. Upon decoding the opcode, μP understands the operation to be performed and hence "executes" the instruction. This entire process is called an **"Instruction cycle"**. This process is repeated for the next instruction. Like this, one by one, all instructions of a program are executed.
Of course, by new concepts like **pipelining**, **multitasking**, **multiprocessing** etc., this procedure has become very advanced and efficient today. You will get to learn all of them, in the due course of this ever-intriguing subject.

We begin learning with basic processors like **8085** or **8086**, but make no mistake, none of this is "outdated". Yes, your mobile phone or your computer today uses the most advanced processors (Apple's **A12 Bionic** etc.), but to run a **traffic light** or **TV remote** control you don't need a core i7 now, do you? And these are used by the millions across the world. They simply use processors of the same grade as an 8085 or an 8086, with different product numbers as they are made by various manufacturers.

MEMORY

Memory is used to **store information**.
It stores two kinds of information… **programs and data**. Yes, think about it! Everything that's stored in your computer's memory is either some program or some data. MS Word is a program, the Word Documents are data. Your Image Viewer is a program, the images are data. WhatsApp is a program, its messages are data. Instagram is a program, the feed on the wall is the data… and so on!

All programs and data are stored in the memory, in digitized form, where every information is represented in 1s and 0s called binary digits or simply bits.

There are various forms of memory devices.
The main memory also called primary memory consists of Ram and ROM.
Other memory devices like Hard disk, Floppy, CD/ DVD etc. are secondary storage devices.

Additionally there is also a high speed memory called Cache composed of SRAM.
For the majority portion of this book, you are dealing with the initial processors like 8086.
It will be in your best interest to think of Primary Memory only, whenever we speak of memory. That is because, secondary memory and high speed memories were implemented much later in the evolution of processors as the demand for mass storage and high speed performance started increasing. So, from now on in this book, unless specified otherwise, **the word memory refers to primary memory that is RAM and ROM.**

The memory is a series of locations.
Each location is identified by its own unique address.
Every memory location contains 1 Byte (8 bits) of data. There is a very good reason for this, and you will learn it when we discuss the topic of memory banking in 8086.

I/O DEVICES

I/O devices are used to enter programs and data as inputs and display or print the results as outputs.
We are all familiar with devices such as the keyboard, mouse, printer, monitor etc. Every form of computer system has a set of I/O devices for human interaction. A device like a touch screen performs dual functions of both input and output.

The μP, Memory and I/O are all connected to each other using the System Bus.

SYSTEM BUS

A Bus is a set of lines. They are used to transfer information, obviously in binary form.
A line connected to Vcc will carry a logic 1 and if connected to Gnd it will carry logic 0.
Hence one line can transfer 1 bit. If we want multiple bits to be transmitted together, then we use a set of lines grouped together and that's called a bus.

The size of a bus refers to the number of lines it contains and is always given in terms of bits.
E.g.:: An 8-bit bus has 8 lines, a 16 bit bus has 16 lines and so on.

There are three types of busses… Address, Data and Control Bus. Collectively they are called the system bus. Lets take a closer look at them.

i) ADDRESS BUS

It carries the address where the operation has to be performed. Say the processor wants to write some data at a memory location. Firstly the processor will give the desired address on the address bus. This address will select a unique memory location. That's when data will be transferred with that location.
It is therefore obvious that **bigger the address bus, more is the number of memory locations** it can address and hence bigger the total size of the memory. Here is the relation between size of the address bus and size of the memory.
An address bus of 1 bit can give a total of two addresses: 0 and 1.
Hence can access a total memory of two locations.
A 2-bit address bus can generate a total of 4 addresses 00, 01, 10, 11 and hence can access a total of 4 locations.
3-bit address bus… 8 locations and so on.

So here is the rule,

An N-bit address bus can totally access 2^N locations.

As mentioned earlier, one memory location contains one byte of data.
This brings us to the following conclusion.
A processor with an N-bit address bus can access a memory of 2^N Bytes.

Lets solve a typical VIVA (oral exam) question.

Given the size of address bus, you have to figure out the size of the memory.

ADDRESS BUS (N - BIT)	MEMORY (2^N BYTES)
4 – bit	2^4 = 16 Bytes
6 – bit	2^6 = 64 Bytes
10 – bit	2^{10} = 1024 Bytes = 1 KB
16 – bit	$2^{16} = 2^6 \times 2^{10}$ = 64 x 1 K = 64 KB … (8085)
20 – bit	$2^{20} = 2^{10} \times 2^{10}$ = 1 K x 1 K = 1 MB … (8086)
30 – bit	$2^{30} = 2^{10} \times 2^{20}$ = 1 K x 1 M = 1 GB
32 – bit	$2^{32} = 2^2 \times 2^{30}$ = 4 x 1 G = 4 GB …. (80386 and Pentium)
40 – bit	$2^{40} = 2^{10} \times 2^{30}$ = 1K x 1G = 1 TB … (Typical Hard Disk)

In case you found it a little difficult after the 4th row, that is because you may have got a little confused with the powers of 2. This issue will persist all along the subject. The smarter thing to do is once for all, lets get this hurdle past us. Lets get all powers of 2 clearly sorted. You will realize in the due course of this subject, how helpful this small exercise will prove to be. Frankly speaking there's rarely a topic in this subject that is not connected with some power of 2. Lets sort this, totally!

Powers of 2

2^N	VALUE
2^0	0
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024 = 1K
2^{20}	$2^{10} \times 2^{10}$ = 1K x 1K = 1M
2^{24}	$2^4 \times 2^{20}$ = 16 x 1M = 16M
2^{28}	$2^8 \times 2^{20}$ = 256 x 1M = 256M
2^{30}	$2^{10} \times 2^{20}$ = 1K x 1M = 1G
2^{36}	$2^6 \times 2^{30}$ = 64 x 1G = 64G
2^{40}	$2^{10} \times 2^{30}$ = 1K x 1G = 1T

Similarly, there is one more very important fundamental that needs to be sorted out before we start learning the bigger concepts. You need to be sure of number representation and number conversions. You may have been familiar with various number systems like Decimal, Hexadecimal, Binary etc… Out of all of them, Hexadecimal and Binary are the two most widely used number systems in our course of learning this subject.

Every number we write, either in programs or in theory examples, is a hexadecimal number. When this number gets stored inside the computer, it is in binary form This simply means, you must be very well versed with hex-binary conversions as this will be needed while understanding various examples.

A single hexadecimal digit ranges from 0H… FH. This has 16 options. Hence, to represent the number in binary we need 4 bits as 2^4 =16. The following table shows this conversion.

HEX	BINARY
0 H	0000
1 H	0001
2 H	0010
3 H	0011
4 H	0100
5 H	0101
6 H	0110
7 H	0111
8 H	1000
9 H	1001
A H	1010
B H	1011
C H	1100
D H	1101
E H	1110
F H	1111

NO! You do not need to mug this up. There is a simple trick of "8421" that can get you any value from the above table. Consider the number 5 which is basically 4+1. Now consider the four binary bits corresponding to values 8, 4, 2 and 1. Since we need the equivalent of 5 which is 4 plus 1, we need 4 and 1 but we don't need 8 and 2 so in positions of 8 and 2 we put a "0" and in positions of 4 and 1 we put a "1". So we get 0101. Lets take the example of 0 H. We don't need any of them (8 or 4 or 2 or 1). So we put a "0" for all of them and hence get 0000. If we need F (which is 15), it is 8 + 4 + 2 + 1 so we need all of them and hence the binary equivalent is 1111. Take 9 as an example, 9 = 8 + 1. So we need 8 and 1 but not 4 and 2 so we put a "1" for 8 and 1 positions and a "0" for 4 and 2 positions giving us 1001. Hope you can now convert any hex digit into binary.

Now consider a two digit number like 25H. To convert this to binary we need to substitute the 4 bit equivalents of 2 and 5 respectively. 2H is 0010 and 5H is 0101.

Hence the number 25H in binary will be 0010 0101.
Similarly a number like 74H will be 0111 0100 in binary. 89H will be 1000 1001 and so on.

As you noticed, the numbers 25H, 74H and 89H are all 2 digits in hexadecimal and hence need 8 bits in binary. Such numbers are called 8 bit numbers. The range of 8 bit and 16 bit numbers is mentioned below:

8 BIT NUMBERS (ALSO CALLED A "BYTE")

HEX	BINARY
00 H	0000 0000
01 H	0000 0001
...	...
68 H	0110 1000
...	...
FE H	1111 1110
FF H	1111 1111

16 BIT NUMBERS (ALSO CALLED A "WORD")

HEX	BINARY
0000 H	0000 0000 0000 0000
0001 H	0000 0000 0000 0001
...	...
4831 H	0100 1000 0011 0001
...	...
FFFE H	1111 1111 1111 1110
FFFF H	1111 1111 1111 1111

ii) DATA BUS

It carries data to and from the processor.
The size of data bus determines how much data can be transferred in one operation (cycle).
Bigger the data bus, faster the processor, as it can transfer more data in one cycle.

iii) CONTROL BUS

It Carries control signals like RD, WR etc.
These signals determine the kind of operation that will be performed on the system bus.