Study of X-86 Family

Ayush Mishra
Student, Department of ETRX,
Shree L.R. Tiwari College of Engineering
Mira Road, Thane,
Mumbai, Maharashtra, India.

Shreya Shivalkar
Student, Department of ETRX,
Shree L.R. Tiwari College of Engineering,
Mira Road, Thane,
Mumbai, Maharashtra, India.

Shweta Yadav
Student, Department of ETRX,
Shree L.R. Tiwari College of Engineering,
Mira Road, Thane,
Mumbai, Maharashtra, India.

Anil Suthar
Student, Department of ETRX,
Shree L.R. Tiwari College of Engineering,
Mira Road, Thane,
Mumbai, Maharashtra, India.

Abstract-
The 8086 was introduced in 1978 as a fully 16-bit extension of Intel's 8-bit 8080 microprocessor, with memory segmentation as a solution for addressing more memory.
The term "X86" came into being because the names of several successors to Intel's 8086 processor end in "86", including the 80186, 80286, 80386 and 80486 processors.This paper is an in detail study of X-86 family consisting of its features, advantages, applications, instruction set and overview of other processors/extensions in the X-86 family.Architecture,Programmer's model and Various addressing modes are also explained briefly.

Keywords- X-86 family, Registers, Pipelining, Modes of operation, clock speed, Core

I. INTRODUCTION

The 8086 was introduced in 1978 as a fully 16-bit extension of Intel's 8-bit 8080 microprocessor, with memory segmentation as a solution for addressing more memory.The X86 instruction set architecture originated by Intel and has evolved over time by the addition of new instructions as well as the expansion to 64-bits. As of 2009, x86 primarily refers to IA-32 (Intel Architecture, 32-bit) and/or X86-64,the extension to 64 bit .Versions of the x86 instruction set architecture have been implemented by Intel, AMD and several other vendors,with each vendor having its own family of x86 processors.

II. FEATURES

- Wide Range of Clock Rates
- High Performance Processor (Up to 19 Times the 8086 Throughput)

- Large Address Space 16 Megabytes Physical/1 Gigabyte Virtual per Task
- Integrated Memory Management,
- Four-Level Memory Protection and Support for Virtual Memory and Operating
- High Bandwidth Bus Interface (25 Megabyte/Sec)
- Can function in Protected Mode and Real Mode[10]

IV. PROGRAMMERS MODEL

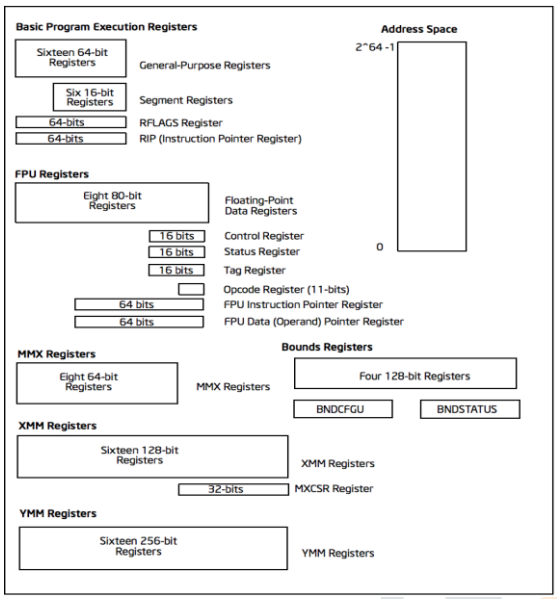

Fig 1. Programmers model

The programming model of the 8086 is considered to be program visible because its registers are used during application programming and are specified by the instructions.The programming model for a microprocessor shows the various internal registers that are accessible to the programmer. In the programming model there are

- 4 General Purpose registers(Data Registers)
- 4 Segment registers
- 2 Pointer registers
- 2 Index registers
- 1 Instruction Pointer register
- 1 Flag register

A.Basic Program execution registers.

a.)General Purpose Registers

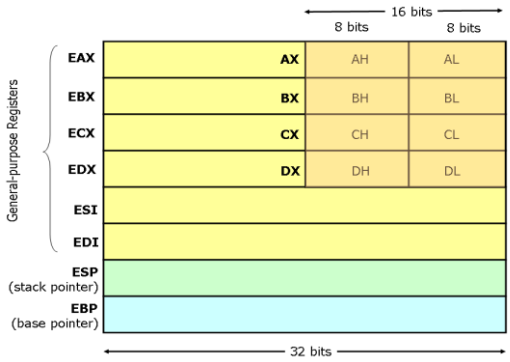

Fig 2. General Purpose Registers

8086 has 8 general purpose registers,labelled **AH,AL,BH,BL,CH,CL,DH and DL** .The 32 bit registers have four separately addressable parts and the 16-bit registers have 16-bit AX, 8-bit AH and AL,16-bit BX, 8-bit BH and BL, 16-bit CX, 8-bit CH and CL, 16-bit DX, 8-bit DH and DL . These registers can be used for arithmetic and logical operations on data . Many have specialized purposes.[3]

Register name	Size(in bits)	Purpose
AL, AH/AX	8/8/16	Known as accumulator. It holds results of arithmetic operations and function return values.
BL, BH/BX	8/8/16	Used to store the base address of the program.
CL, CH/CX	8/8/16	Used for loop and string operations.
DL, DH/DX	8/8/16	A general purpose registers. Also used for I/O operations.

Table 1

b.) Segment Registers .

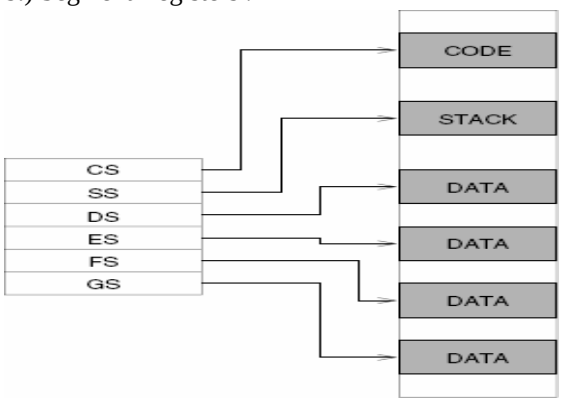

Fig 3. Segment Registers

The four segment registers CS, DS, ES and SS are the same as the segment registers found in Intel 8086 and Intel 286 processors and the FS and GS registers were introduced into the Intel 32 bit architecture.These registers are used to break up a program into parts. As it executes, the segment registers are assigned the base values of each segment. From here, offset values are used to access each command in the program.[4]

Segment Register	Size (bits)	Purpose
CS	16	Code segment register.Used for fetching instructions.
DS	16	Data segment register. Used for data accesses.
ES	16	Extra segment register. Used during string operations.
SS	16	Stack segment register. Base location of the stack segment.
FS	16	Extra segment register.
GS	16	Extra segment register.

Table 2

c.) Instruction pointer register

Instruction pointer register contains the offset in the current code segment for the next instruction to be executed. It is advanced from one instruction boundary to the next one in straight line code.Instruction pointer register cannot be accessed directly by software. It is controlled implicitly by control-transfer instructions such as **JMP, JCC, CALL, RET and IRET**, interrupts and exceptions. [6]

d.) Flag Registers

Flag is a flip-flop,which indicates some condition. Depending upon the value of result after any Arithmetic and Logical operations, the flag bits become **Set (1)** or **Reset (0)**.The 8086 has a 16-bit flag register with 9-active flags.These flags are of two types: 6 Status flags namely **carry flag, parity flag, auxiliary carry flag, zero flag, sign flag** and 3 Control flags namely **trap flag, interrupt flag and direction flag**.The Control flags are used to control certain operations. They are changed by the programmer.[8]

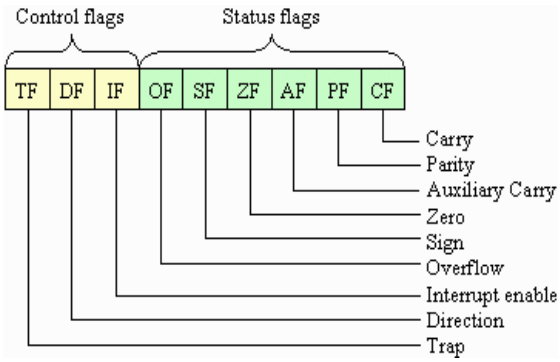

Fig 4. Flag Registers

Flag	Bit	Purpose
CF (Carry Flag)	0	Set if an arithmetic operation generates a carry or a borrow out of the most significant bit of the result, cleared otherwise.
PF (Parity Flag)	2	Set if the least-significant byte of the result contains an even number of 1 bit, cleared otherwise.
AF (Adjust Flag)	4	Set if an arithmetic operation generates a carry or a borrow out of bit 3 of the result, cleared otherwise.
ZF (Zero Flag)	6	Set if the result is zero, cleared otherwise
SF (Sign Flag)	7	Set equal to the most-significant bit of the result. 0 indicates a positive value, whereas 1 indicates a negative value.
AF (Auxiliary carry Flag)	1	Set if the integer result is too large a positive number or too small a negative number, cleared otherwise.
TF (Trap Flag)	0	Set if processor enters single-step mode,cleared otherwise
IF (Interrupt Flag)	0	Set if CPU recognizes Maskable interrupt requests,cleared otherwise.
DF (Direction flag)	0	Set if Instructions are processed high address to low address,cleared vice-versa.

Table 3

B. FPU Registers

There are Eight 80-bit floating point data registers.They are named as ST(0),ST(1),..ST(7).They are arranged in stacks and are used for all floating-point arithmetic operations.All floating-point instructions provide a 5-bit field that specifies which floating-point registers to use in the execution of the instruction. All floating-point instructions other than loads are performed on operands located in floating-point registers. Numbers are pushed onto the stack *from memory*, and are popped off the stack back *to memory*. There is no instruction allowing to transfer values directly to or from ALU registers. [5]

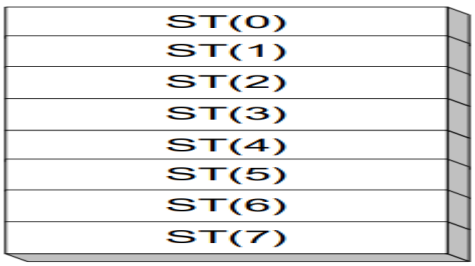

Fig 5. FPU Registers

C.MMX Registers

MMX defines eight registers, called **MM0** through **MM7,** and operations that operate on them. Each register is 64 bits wide and can be used to hold either 64-bit integers, or multiple smaller integers in a **"Packed"** format. It provides arithmetic and logic operations on 64-bit integer numbers.The extension contains 16 data registers of 64-bits and eight control registers of 32-bits.[2]

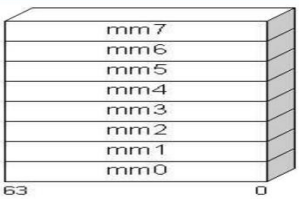

Fig 6. MMX Registers

D.XMM Registers

These registers can be accessed directly using the names **XMM0** to **XMM7**; and they can be accessed independently from the **X87 FPU** and **MMX** registers and the **General-purpose registers.**XMM registers can only be used to perform calculations on data; they cannot be used to address memory.Data can be loaded into or written from the registers to memory in 32-bit, 64-bit, and 128-bit increments.[4]

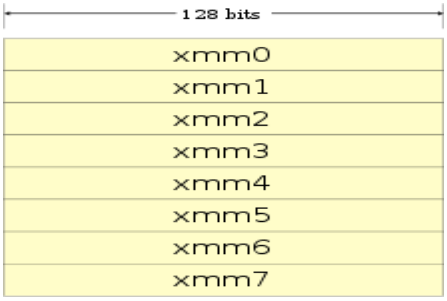

Fig 7. XXM Registers

5) Instruction Set of X-86 Microprocessors:

An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions that a microprocessor supports is called Instruction Set. 8086 has more than 20,000 instructions.Instructions are classified on the basis of functions they perform.The 8086 microprocessor supports 8 types of instructions −

- Data Transfer Instructions
- Arithmetic Instructions
- Bit Manipulation Instructions
- String Instructions
- Program Execution Transfer Instructions (Branch & Loop Instructions)
- Processor Control Instructions
- Iteration Control Instructions
- Interrupt Instructions

a.) Data Transfer Instructions:

The data transfer instructions are used to transfer data from one location to another. This transfer of data can be either from register to register, register to memory or memory to register.It is important to note here that the memory to memory transfer of data directly is not possible.All the store, move, load, exchange, input and output instructions belong to this category. These instructions do not affect any flags.[1]

For eg: MOV, PUSH, IN, OUT etc

b.) Arithmetic Instructions:

These instructions perform the arithmetic operations, like addition, subtraction, multiplication and division along with the respective ASCII and decimal adjust instructions. The increment and decrement operations also belong to this type of instructions. The arithmetic instructions affect all the condition code flags.Unlike in 8085 microprocessor, in 8086 microprocessor the destination operand need not be the accumulator.[4]

For eg: ADD, SUB, MUL, DEC etc.

c.)Logical Instructions:

These instructions are used to perform operations where data bits are involved, i.e. logical AND, OR, XOR, NOT and TEST operations etc.. In an 8086 microprocessor, the destination operand need not be the accumulator. For logical instruction ,the source may be an immediate number, a register or a memory location. The destination may be a register or a memory location. The source and destination both cannot be memory locations in the same instruction.[5]

For eg: ROL, ROR, RCR, RCL etc.

d.)String Instructions:

String is a series of data bytes or words available in memory at consecutive locations. Their memory is always allocated in a sequential order.. It is a series of the same type of data items in sequential memory locations. They can move strings, compare strings, search for a specific value within a string, initialize a string to a fixed value, and do other primitive operations on strings. Using string instructions may speed up your array manipulation code considerably.[7]

For eg: REP, MOVS, INS, OUTS etc.

e.) Branch and loop Instructions:

These instructions are used to execute the given instructions for a number of timesIt is also called program execution transfer instruction. Instructions of this group transfer program execution from the normal sequence of instructions to the specified destination or target. When the instruction is executed, the code segment (CS) and Instruction pointer registers get loaded with new values of CS and Ip corresponding to the Location to be transferred. [10]

For eg: LOOP, CALL, JMP, JC etc.

f.) Processor Control Instructions:

These instructions are used to control the processor action by setting/resetting the flag values They do not affect any other flag. They are responsible for setting(1) or resetting(0) the values of flag registers.[9]

For eg: STC, CLC, CMC, etc.

g.) Interrupt Instructions:

These instructions are used to call the interrupt during program execution. They are the instructions that interrupt the flow of a program and can cause the program to either to end or to skip or to Go temporarily to a subroutine either within the program or outside the program [9]

For eg: INT, INTO, IRET .[4]

V. FAMILY OF X-86

The term "x86" came into being because the names of several successors to Intel's 8086 processor end in "86", including the 80186, 80286, 80386 and 80486 processors.In the 1980s and early 1990s, when the 8088 and 80286 were still in common use, the term x86 usually represented any 8086-compatible CPU. Today, x86 usually implies a binary compatibility also with the 32-bit instruction set of the 80386.

a.) 8086 Microprocessor

8086 Microprocessor is an enhanced version of 8085Microprocessor that was designed by Intel in 1976. It is a 16-bit Microprocessor having 20 address lines and16 data lines that provides up to 1MB storage. It consists of a powerful instruction set, which provides operations like multiplication and division easily.

It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for systems having multiple processors and Minimum mode is suitable for systems having a single processor.

b.) 80186 Microprocessor

The 80186 series was generally intended for embedded systems, as microcontrollers with external memory. Therefore, to reduce the number of integrated circuits required, it included features such as clock generator, interrupt controller, timers, wait state generator, DMA channels, and external chip select lines.

Multiply and divide also showed great improvement being several times as fast as on the original 8086 and multi-bit shifts were done almost four times as quickly as in the 8086.

c.) 80826 Microprocessor

The 80286 is the first member of the family of advanced microprocessors with memory management and protection abilities. The 80286 CPU, with its 24-bit address bus is able to address 16 Mbytes of physical memory. Various versions of 80286 are available that runs on 12.5 MHz, 10 MHz and 8 MHz clock frequencies. 80286 is upwardly compatible with 8086 in terms of instruction set.

d.) 80386 Microprocessor

The Intel 80386, also known as i386 or just 386, is a 32-bit microprocessor introduced in 1985. The Internal Architecture of 80386 is divided into 3 sections. Central processing unit(CPU) , Memory management unit(MMU), Bus interface unit(BIU) and Central processing unit. Further divided into Execution unit(EU) and Instruction unit(IU). The Memory management unit consists of a Segmentation unit and Paging unit.

Segmentation unit allows the use of two address components and viz. Segment and offset for reliability and sharing of code and data. It allows segments of size 4Gbytes at max. The Paging unit organizes the physical memory in terms of pages of 4kbytes size each.

e.) 80486 Microprocessor

The Intel 80486, also known as the i486 or 486, is the successor model of 32-bit x86 microprocessor to the Intel 80386. Introduced in 1989, the 80486 improved on the performance of the 80386DX thanks to on-die L1 cache and floating-point unit, as well as an improved, five-stage tightly-coupled pipelined design.

One of the feature included in a 80486 is a built in math coprocessor. This coprocessor is essentially the same as the 80387 processor used with a 80386, but being integrated on the chip allows it to execute math instructions about three times as fast as a 80386/387 combination. 80486 is an 8Kbyte code and data cache.To make room for the additional signals, the 80486 is packaged in a 168 pin, pin grid array package instead of the. The 132 pin PGA used for the 80386.

f.) Pentium Microprocessor

Pentium, Family of microprocessors developed by Intel Corp. Introduced in 1993 as the successor to Intel's 80486 microprocessor, the Pentium contained two processors on a single chip and about 3.3 million transistors. Using a CISC (complex instruction set computer) architecture, its main features were a 32-bit address bus, a 64-bit data bus, built-in floating-point and memory-management units, and two 8KB caches.

It was available with processor speeds ranging from 60 megahertz (MHz) to 200 MHz. The Pentium quickly became the processor of choice for personal computers. It was superseded by ever faster and more powerful processors, the Pentium Pro (1995), the Pentium II (1997), the Pentium III (1999), and the Pentium 4 (2000). The Pentium processor has a memory space of 4 GB (232 bytes) and a separate I/O space with 64 KB of addressable locations. The memory space is organized as a sequence of 64-bit quantities. Each 64-bit location has eight individually addressable bytes at consecutive memory addresses. The I/O space is organized as a sequence of 32-bit quantities. Each 32-bit quantity has four individually addressable bytes at consecutive memory addresses.

g.) Pentium Pro Microprocessor

It's a sixth-generation x86 microprocessor developed and manufactured by Intel introduced on November 1, 1995. The Pentium Pro microprocessor belongs to the CISC (Complex Instruction Set Computers) machines. Processors of the Pentium Pro family are mostly present in majority of personal computers. The term 'Pentium processor' refers to an Intel x86 family of microprocessors that share a common architecture and instruction set. The Pentium Pro is capable of both dual- and quad-processor configurations.

Pentium Pro clock speeds were 150, 166, 180 or 200 MHz with a 60 or 66 MHz external bus clock. Some users chose to overclock their Pentium Pro chips, with the 200 MHz version often being run at 233 MHz, the 180 MHz version often being run at 200 MHz, and the 150 MHz version often being run at 166 MHz. The chip was popular in symmetric multiprocessing configurations, with dual and quad SMP server and workstation setups being commonplace.

h.) Core Microprocessor

Core is a brand encompassing a range of Intel's consumer 64-bit x86-64 single, dual, and quad-core microprocessors based on the Core microarchitecture. Each contains two cores, packaged in a multi-chip module. The introduction of Core relegated the Pentium brand to the mid-range market, and reunified laptop and desktop CPU lines for marketing purposes under the same product name. The Core 2 brand was introduced in mid-July of 2006, comprising the Solo , Duo, Quad-core and in 2007, the Extreme sub brands. Intel Core processors with vPro technology which was designed for businesses purposes include the dual-core and quad-core branches.Although Woodcrest

processors are also based on the Core architecture, they are available under the Xeon brand

VI. ADDRESSING MODES

The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

The 8086 memory addressing modes provide flexible access to memory, allowing you to easily access variables, arrays, records, pointers, and other complex data types. The key to good assembly language programming is the proper use of memory addressing modes[8]

.

a.)Immediate addressing mode

The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.The 16-bit effective address (EA) is taken directly from the displacement field of the instruction. The displacement (unsigned 16-bit or sign-extended 8-bit number) is stored in the location following the instruction opcode.

For eg:ADD AL

b.)Register Indirect Addressing Mode:

This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI. Indirect addressing is generally used for variables containing several elements like arrays.

For eg::MOV AX, [BX]

c.)Based addressing mode

In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.The physical memory address is calculated according to the base register.When the stack is accessed, the 20-bit physical address is computed from BP and SS.

For eg: ADD AX, [BX+SI].

d) Indexed Addressing Mode :

In this mode, the effective address is calculated by adding the unsigned 16-bit or sign-extended 8-bit displacement and the contents of SI or DI.The effective address is the sum of index register and displacement.

For eg:MOV AX, [SI+2000]

e.)Based-index addressing mode

The based indexed addressing modes are simply combinations of the register indirect addressing modes. In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.

Example: ADD CX, [AX+SI]

f.)Based indexed with displacement mode

In this addressing mode, the operands offset is computed by adding the base register contents. An Index registers contents and 8 or 16-bit displacement.These addressing modes are a slight modification of the base/indexed addressing modes with the addition of an eight bit or sixteen bit constant.

Example:
MOV AX, [BX+DI+08]

g.)String addressing mode

This addressing mode is related to string instructions. In this the value of SI and DI are auto incremented and decremented depending upon the value of directional flag.[7]

Example: MOVS B

VI. MODES OF OPERATION

A. Protected Mode

Protected mode was first added to the x86 architecture in 1982 with the release of Intel's 80286 processor, and later extended with the release of the 80386 in 1985. Due to the enhancements added by protected mode, it has become widely adopted and has become the foundation for all subsequent enhancements to the x86 architecture. Protected mode has a number of features designed to enhance an operating system's control over application software, in order to increase security and system stability. These additions allow the operating system

to function in a way that would be significantly more difficult or even impossible without proper hardware support.[]

B.) Real Mode

Real mode programming involved manipulating data between 0 and 1 MB. The memory there was accessible by any program running in memory. And while, generally speaking, there weren't multiple programs running in memory at the same time, the OS had to set up things called "interrupt handlers" because the x86 architecture is interrupt driven. The processor calculates the physical address of a memory reference by shifting the value of a segment register to the left by 4 binary digits and then adding the offset address to this value. Thus, two 16-bit values (segment and offset) are combined to form a single 20-bit physical address. There are no linear addresses in real mode.[9]

VII. ADVANTAGES OF X-86 FAMILY

- It was the world's first general purpose microprocessor.
- Protected mode and Real mode were introduced.
- Processors were highly pipelined.
- Clock speed was increased.
- Development of "NetBurst" architecture.
- Notions like "Hyper-Threading", and "Multi-Core" chips were introduced.
- Due to X-86 microprocessors, "Multi-core" desktop versions were developed.[1]

IX. APPLICATION OF X-86 FAMILY

- It's used for small applications like for calculators, scientific calculators & small arithmetic operations.
- protected virtual-address mode enabled Multitasking
- Used as an in-circuit emulator.
- Memory unit was subdivided into a Segmentation unit and Paging unit.
- Faster math calculation due to 8Kbyte code and data cache.
- Used in modems, telephone, digital telephone sets, and also in air reservation systems and railway reservation systems.[3]

X. CONCLUSION

In this article I have discussed briefly about all the arithmetic and logical instructions of X-86 microprocessors and all the registers are shown in above.The microprocessor is one of the most significant inventions in the field of technology because of its effect on technologic engineering, different aspects of life, and of course how it can store data larger than the size of the actual chip. The impacts the microprocessor had had on life and whether it was for the better the invention was made is a controversy that still goes to this day.

XI. REFERENCES

1. Kinnari Prakashan, TPS Computer Science - II, STD. XII.
2. John Crisp, Introduction of Microprocessor and Microcontroller,
3. R.D.Supekar, Vaishali Ghule, Taraka Ghamande, Vilas Karmude, XIIth Computer Science, Revised Ed.
4. www.eeeguide.com
5. en.wikibooks.org
6. https://www.javatpoint.com/instruction-set-of-8086
7. https://www.studytonight.com/computer-architecture/addressingmodes-instructioncycle
8. www.javatpoint.com/instruction-set-of-8086
9. www.tutorialspoint.com/microprocessor/microprocessor_8086_instruction_sets.htm
10. www.google.com