

A
Seminar report on

8086 Microprocessor

Submitted in partial fulfillment of the requirement for the award of degree
Of Electronics

SUBMITTED TO:

www.studymafia.org

SUBMITTED

BY:

www.studymafia.org

www.studymafia.org

Preface

I have made this report file on the topic **8086 Microprocessor** , I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude towho assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

INTRODUCTION

- 8086 is an enhanced version of 8085 that has been developed by Intel in 1976.
- It is a 16 bit Microprocessor. It has a powerful instruction set and it is capable to providing multiplication and division operations directly. It has 20 address lines and 16 data lines. So it can access up to 1 MB of memory.
- It supports two modes of operation: first is maximum mode and second is minimum mode. Minimum mode is applicable for system that has a single processor and maximum mode is used for the multiprocessor system.

- 8086 provides an additional features that it has an instruction queue capable to store six instruction bytes from the memory. The next instruction is fetched while the present instruction is being executed. So it makes the processor fast.

History

- Microprocessor journey started with a 4-bit processor called 4004; it was made by Intel Corporation in 1971. It was 1st single chip processor. Then the idea was extended to 8-bit processors like 8008, 8080 and then 8085 (all are Intel products). 8085 was a very successful one among the 8-bit processors; however its application is very limited bcoz of its slower computing speed and other quality factors.
- Some years later Intel came up with its 1st 16-bit processors 8086. at the same time other manufacturers were also making processors like 68000 (by Motorola), Zilog z-80, General instrument PIC16X, MOS Technology 6502, etc...In 1979 Intel released a modified version of 8086 as 8088. Next Intel started updating 80x86 series by introducing 80286, 80386, 80486, Pentium and then Pentium series.
- After 80486, the next processor in series was to be said 80586, but Intel named it as Pentium bcoz of its copyright problem. Further updating in Pentium resulted in Pentium-I, Pentium-II, etc.

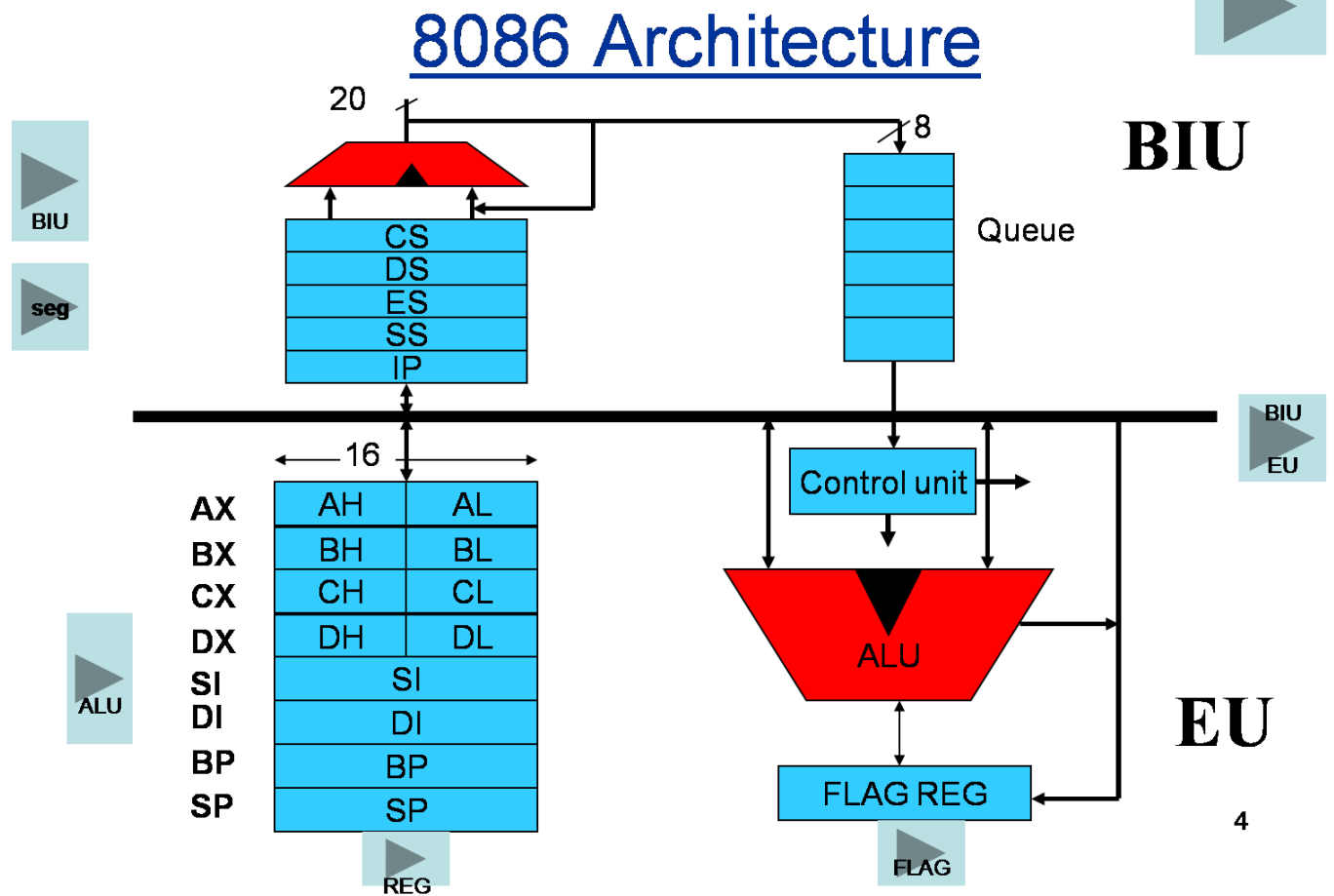
Features of 8086

- 8086 is a 40 pin IC.
- It is a 16-bit processor.
- Its operating voltage is 5 volts.
- Its operating frequency is 5 MHz
- Total memory addressing capacity is 1MB (external).
- It has 16-bit data bus and 20-bit address bus.
- It has fourteen 16-bit registers.
- Higher throughput (speed).
- It has around 20000 transistors in its circuitry and it is made in HMOS technology.

Block Diagram of Intel 8086

The 8086 CPU is divided into two independent functional units:

1. Bus Interface Unit (BIU)
2. Execution Unit (EU)



Features of 8086 Microprocessor

1. Intel 8086 was launched in 1978.
2. It was the first 16-bit microprocessor.
3. This microprocessor had major improvement over the execution speed of 8085.
4. It is available as 40-pin Dual-Inline-Package (DIP).
5. It is available in three versions:
 - a. 8086 (5 MHz)
 - b. 8086-2 (8 MHz)
 - c. 8086-1 (10 MHz)
6. It consists of 29,000 transistors.

Bus Interface Unit (BIU)

The function of BIU is to:

- Fetch the instruction or data from memory
- Write the data to memory
- Write the data to the port
- Read data from the port

Instruction Queue

1. To increase the execution speed, BIU fetches as many as six instruction bytes ahead to time From memory
2. All six bytes are then held in first in first out 6 byte register called instruction queue.
3. Then all bytes have to be given to EU one by one.
4. This pre fetching operation of BIU may be in parallel with execution operation of EU, which Improves the speed execution of the instruction

Execution Unit (EU)

The functions of execution unit are:

- To tell BIU where to fetch the instructions or data from.
- To decode the instructions.
- To execute the instructions.

The EU contains the control circuitry to perform various internal operations. A decoder in EU Decodes the instruction fetched memory to generate different internal or external control signals Required to perform the operation. EU has 16-bit ALU, which can perform arithmetic and logical Operations on 8-bit as well as 16-bit.

ADDRESSING MODES OF 8086

Addressing mode indicates a way of locating data or operands. Depending upon the data types used in the instruction and the memory addressing modes, any instruction may belong to one or more addressing modes or some instruction may not belong to any of the addressing modes. Thus the addressing modes describe the types of operands and the way they are accessed for executing an instruction. Here, we will present the addressing modes of the instructions depending upon their types. According to the flow of instruction execution, the instructions may be categorized as

- (i) **Sequential control flow instructions**
- (ii) **Control transfer instructions.**

Sequential control flow instructions are the instructions, which after execution, transfer control to the next instruction appearing immediately after it (in the sequence) in the program. For example, the arithmetic, logical, data transfer and processor control instructions are sequential control flow instructions. The control transfer instructions, on the other hand, transfer control to some predefined address somehow specified in the instruction after their execution. For example, INT, CALL, RET and JUMP instructions fall under this category.

The addressing modes for sequential control transfer instructions are explained as follows:

- **Immediate:** In this type of addressing, immediate data is a part of instruction, and appears in the form of successive byte or bytes.

Example: MOV AX, 0005H

In the above example, 0005H is the immediate data. The immediate data may be 8-bit or 16-bit in size.

- **Direct:** In the direct addressing mode, a 16-bit memory address (offset) is directly specified in the instruction as a part of it.

Example: MOV AX, [5000H]

Here, data resides in a memory location in the data segment, whose effective address may be computed using 5000H as the offset address and content of DS as segment address. The effective address, here, is $10H * DS + 5000H$.

- **Register:** In register addressing mode, the data is stored in a register and it is referred using the particular register. All the registers, except IP, may be used in this mode.

Example: MOV BX, AX.

- **Register Indirect:** Sometimes, the address of the memory location, which contains data or operand, is determined in an indirect way, using the offset registers. This mode of addressing is known as register indirect mode. In this addressing mode, the offset address of data is in either BX or SI or DI registers. The default segment is either DS or ES. The data is supposed to be available at the address pointed to by the content of any of the above registers in the default data segment.

Example: MOV AX, [BX]

Here, data is present in a memory location in DS whose offset address is in BX. The effective address of the data is given as $10H * DS + [BX]$.

- **Indexed:** In this addressing mode, offset of the operand is stored in one of the index registers. DS and ES are the default segments for index registers SI and DI respectively. This mode is a special case of the above discussed register indirect addressing mode.

Example: MOV AX, [SI]

Here, data is available at an offset address stored in SI in DS. The effective address, in this case, is computed as $10H * DS + [SI]$.

- **Register Relative:** In this addressing mode, the data is available at an effective address formed by adding an 8-bit or 16-bit displacement with the content of any one of the registers BX, BP, SI and DI in the default (either DS or ES) segment. The example given before explains this mode.

Example: MOV Ax, 50H [BX]

Here, effective address is given as $10H * DS + 50H + [BX]$.

- **Based Indexed:** The effective address of data is formed, in this addressing mode, by adding content of a base register (any one of BX or BP) to the content of an index register (any one of SI or DI). The default segment register may be ES or DS.

Example: MOV AX, [BX] [SI]

Here, BX is the base register and SI is the index register. The effective address is computed as $10H * DS + [BX] + [SI]$.

- **Relative Based Indexed:** The effective address is formed by adding an 8-bit or 16-bit displacement with the sum of contents of any one of the bases registers (BX or BP) and any one of the index registers, in a default segment.

Example: MOV AX, 50H [BX] [SI]

Here, 50H is an immediate displacement, BX is a base register and SI is an index register. The effective address of data is computed as $160H * DS + [BX] + [SI] + 50H$.

For the control transfer instructions, the addressing modes depend upon whether the destination location is within the same segment or a different one. It also depends upon the method of passing the destination address to the processor. Basically, there are two addressing modes for the control transfer instructions, viz. inter-segment and intra-segment addressing modes.

If the location to which the control is to be transferred lies in a different segment other than the current one, the mode is called inter-segment mode. If the destination location lies in the same segment, the mode is called intra-segment.

Addressing Modes for Control Transfer Instruction

- Intra-segment direct mode: In this mode, the address to which the control is to be transferred lies in the same segment in which the control transfer instruction lies and appears directly in the instruction as an immediate displacement value. In this addressing mode, the displacement is computed relative to the content of the instruction pointer IP.
- The effective address to which the control will be transferred is given by the sum of 8 or 16 bit displacement and current content of IP. In case of jump instruction, if the signed displacement (d) is of 8 bits (i.e. $-128 < d < +128$), we term it as short jump and if it is of 16 bits (i.e. $-32768 < +32768$), it is termed as long jump.
- Intra-segment Indirect Mode: In this mode, the displacement to which the control is to be transferred is in the same segment in which the control transfer instruction lies, but it is passed to the instruction indirectly. Here, the branch address is found as the content of a register or a memory location. This addressing mode may be used in unconditional branch instructions.
- Inter-segment Direct Mode: In this mode, the address to which the control is to be transferred is in a different segment. This addressing mode provides a means of branching from one code segment to another code segment. Here, the CS and IP of the destination address are specified directly in the instruction.
- Inter-segment Indirect Mode: In this mode, the address to which the control is to be transferred lies in a different segment and it is passed to the instruction indirectly, i.e. contents of a memory block containing four bytes, i.e. IP (LSB), IP (MSB), CS (LSB) and CS (MSB) sequentially. The starting address of the memory block may be referred using any of the addressing modes, except immediate mode.

This types of processor replica the hardware on the micro-processor so that it can perform numerous instructions at the same time. These replica resources can be committed arithmetic logic units or multipliers. Superscalars comprise of several operational units. Superscalar micro-processors carry out more than one command throughout a single clock cycle by concurrently transmitting numerous instructions to superfluous operational units in the processor.

- **The Application Specific Integrated Circuit:**

Also known as ASIC microprocessor is intended for extremely precise purposes, which possibly will comprise- automotive emissions control or Personal Digital Assistants computers. ASICs at times is produced to specification, but can also be manufactured by making use of off-the-shelf gears.

- **Digital Signal Multiprocessors (DSPs):**

DSPs are unique micro-processors employed to decode & encode video, or convert digital or video to analog and vice-versa. These operations need a micro-processor particularly excellent at carrying out mathematical calculations. DSP chips are generally employed in SONAR, mobile telephones, RADAR, home theater audio gears and cable set-top boxes.

References

- www.google.org
- www.wikipedia.com
- www.studymafia.org