Basics of Specific Microprocessor Architecture and Its Specifications

Second Stage

Microprocessor

Computer Engineering Department

Lecture No. 2

Prepared by: Asst. Lect. Eng.
Hamza Waleed Hamza

8086 Microprocessor

The Intel 8086 is a 16-bit microprocessor designed by Intel Corporation between 1976-1978. It is also called the iAPX 86. This Intel 8086 microprocessor gave rise to the 16-bit architecture.

Processor	Year of Introduction	Numbers of Transistors	Clock Speed	Address bus	Data Bus	Addressable Memory
4004	1971	2300	108kHz	10bits	4bits	640Bytes
8008	1972	3500	200kHz	14bits	8bits	16K
8080	1974	6000	2MHz	16bits	8bits	64K
8085	1976	6500	5MHz	16bits	8bits	64K
8086	1978	29000	5MHz	20bits	16bits	1M
80286	1982	134000	8MHz	24bits	16bits	16M
80386	1985	275000	16MHz	32bits	32bits	4G
80486	1989	1.2M	25MHz	32bits	32bits	4G
Pentium	1993	3.1M	60MHz	32bits	32bits	4G
P1 – P4	1995-2000	3.1M-42M	150MHz-1.4GHz	32bits-36bits	32bits-64bits	4G-64G

- It is important to note that 80286, 80386, 80486, and Pentium-Pentium4 microprocessors are **upward compatible** with the 8086 Architecture. This mean that 8086/8088 code will run on the 80286, 80386, 80486, and Pentium Processors, but the reverse in not true if any of the new instructions are in use.

- It has multiplexed address and data bus AD0- AD15 and A16 – A19.

- 8086 is designed to operate in two modes, Minimum and Maximum. The minimum mode is selected by applying logic 1 to the MN / MX input pin. This is a single microprocessor configuration. The maximum mode is selected by applying logic 0 to the MN / MX input pin. This is a multi-microprocessors configuration.

- It can prefetching up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.

• It requires +5V power supply.

- The 8086 has a 20-bit address bus, so it can directly access (1Mb) memory locations. Each memory location is byte. Therefore, a sixteen-bit words are stored in two consecutive memory locations.
- The 8088 also has a 20-bit address bus, so it can also address 1MB memory locations.
- The Features of 8086 Microprocessor can generate 16-bit I/O address, hence it can access 2^{16} = 65536 I/O ports.
- The 8086 provides fourteen 16-bit registers.

Internal Architecture of 8086

8086 from the software point of view has two blocks Bus Interface Unit (BIU) and Execution Unit (EU) as in figure below. Both units operate asynchronously to give the 8086 an overlapping instruction fetch and execution mechanism which is called as Pipelining. This results in efficient use of the system bus and system performance.

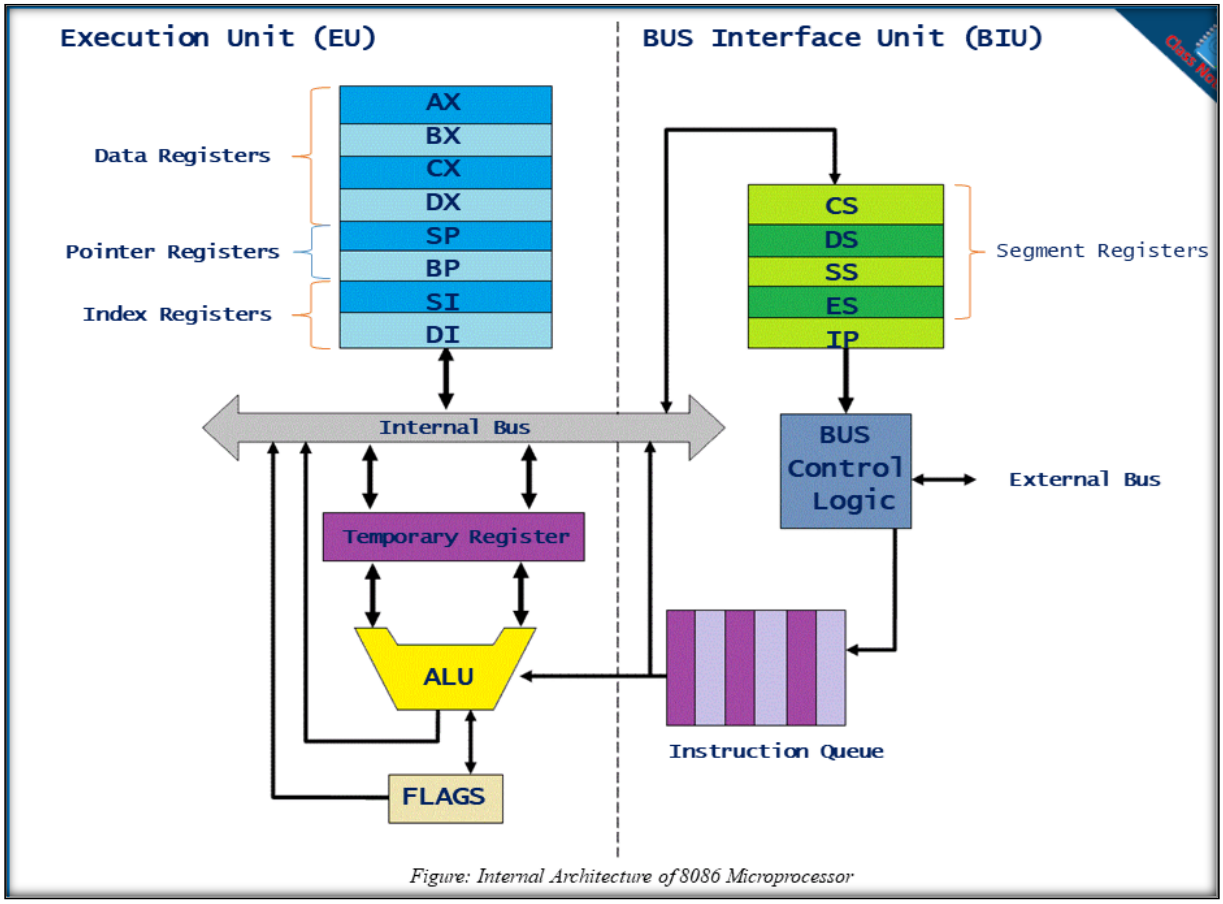


Figure: Internal Architecture of 8086 Microprocessor

Bus Interface Unit (BIU)

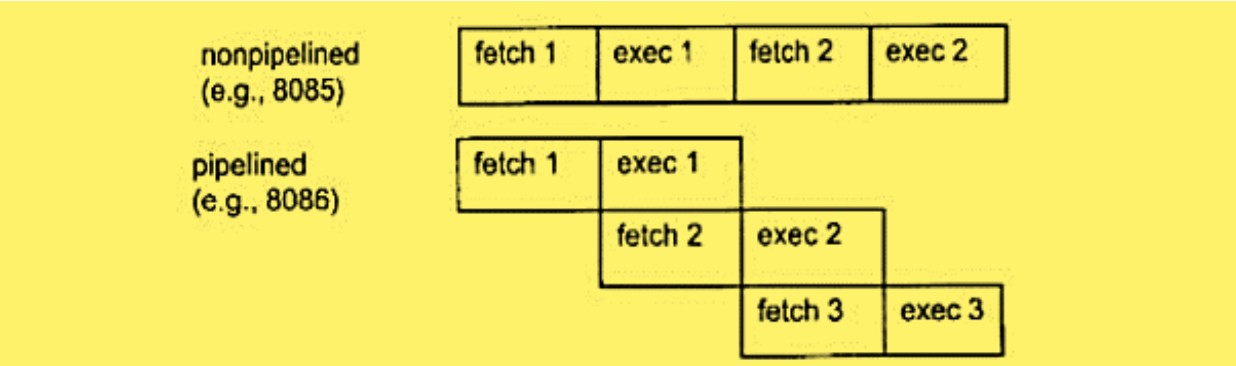
• The BIU contains Instruction queue, Segment registers, Instruction pointer, address generator.

• It provides a full 16 bit bidirectional data bus and 20 bit address bus.

• It performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands.

• The instruction bytes are transferred to the instruction queue. The BIU uses a mechanism known as an instruction stream queue to implement a ***pipeline architecture.***

Pipelining is implemented in 8086 µp by splitting the internal structure into two sections BIU and EU. The two sections work simultaneously, the BIU accessing memory to fetch and store data employing the queue buffer. At the same time, EU executes the instructions previously fetched.



The queue permits prefetching of up to six bytes of instruction code. Whenever the queue of the BIU is not full, it has room for at least two more bytes and at the same time the EU is not requesting it to read or write operands from memory, the BIU is free to look ahead in the program by prefetching the next sequential instruction.

• These prefetching instructions are held in its First In First Out (FIFO) queue. With its 16 bit data bus, the BIU fetches two bytes in a single memory cycle. After a byte is loaded at the input end of the queue, it automatically shifts up through the FIFO to the empty location nearest the output.

• The BIU is also responsible for generating bus control signals such as those for memory read or write and I/O read or write.

• The BIU contains the following registers:

IP - the Instruction Pointer

CS - the Code Segment Register

DS - the Data Segment Register

SS - the Stack Segment Register

ES - the Extra Segment Register

Even though the 8086 has a 1Mbyte address space, not all this memory is active at one time. Actually, the 1Mbytes of memory are partitioned into 64Kbyte (65,536) *segments*. Each **segment** is assigned a ***Base Address*** that identifies its starting point (identify its lowest address byte-storage location). Only four of these 64Kbyte segments are active a time: the ***code segment, stack segment, data segment,*** **and** ***extra segment***. The addresses of these four segments are held in the four segment registers.

• The BIU also contains a dedicated adder which is used to generate the 20 bit physical address that is output on the address bus. This address is formed by adding an appended 16 bit **segment address** and a 16 bit **offset address**.

• For example, the physical address of the next instruction to be fetched is formed by combining the current contents of the code segment CS register and the current contents of the instruction pointer IP register.

EXECUTION UNIT (EU)

• The EU is responsible for decoding and executing all instructions.

• The EU extracts instructions from the top of the queue in the BIU, decodes them, generates operands if necessary, passes them to the BIU and requests it to perform the read or write bus cycles to memory or I/O and perform the operation specified by the instruction on the operands.

• During the execution of the instruction, the EU tests the status and control flags and updates them based on the results of executing the instruction.

• If the queue is empty, the EU waits for the next instruction byte to be fetched and shifted to top of the queue.

• When the EU executes a branch or jump instruction, it transfers control to a location corresponding to another set of sequential instructions. Whenever this happens, the BIU automatically resets the queue and then begins to fetch instructions from this new location to refill the queue.

• EU contains Control circuitry, Instruction decoder, ALU, Pointer, Index and general registers, Flag register.

General data registers

- AX, BX, CX, DX are the general data registers. Low and High bytes of the data registers can be accessed separately
  - AH, BH, CH, DH are the high bytes
  - AL, BL, CL, and DL are the low bytes
- Data Registers are general purpose registers but they also perform special functions
- AX
  - Accumulator Register
  - Preferred register to use in arithmetic, logic and data transfer instructions because it generates the shortest Machine Language Code
  - Must be used in multiplication and division operations
  - Must also be used in I/O operations
- BX
  - Base Register
  - Also serves as an address register
  - Used in array operations
  - Used in Table Lookup operations (XLAT)

- CX
  - Count register
  - Used as a loop counter
  - Used in shift and rotate operations
- DX
  - Data register
  - Used in multiplication and division
  - Also used in I/O operations

Pointer and Index Registers

- Contain the offset addresses of memory locations
- Can also be used in arithmetic and other operations
- SP: Stack pointer
  - Used with SS to access the stack segment
- BP: Base Pointer
  - Primarily used to access data on the stack
  - Can be used to access data in other segments
- SI: Source Index register
  - is required for some string operations
  - When string operations are performed, the SI register points to memory locations in the data segment which is addressed by the DS register. Thus, SI is associated with the DS in string operations.
- DI: Destination Index register
  - It is also required for some string operations.
  - When string operations are performed, the DI register points to memory locations in the data segment which is addressed by the ES register. Thus, DI is associated with the ES in string operations.
- The SI and the DI registers may also be used to access data stored in arrays

• **Flags** is a 16-bit register containing 9 one bit flags.

• **Overflow Flag** (OF) - set if the result is too large positive number, or is too small negative number to fit into destination operand.

• **Direction Flag** (DF) - if set then string manipulation instructions will auto-decrement index registers. If cleared then the index registers will be auto-incremented.

• **Interrupt-enable Flag** (IF) - setting this bit enables maskeable interrupts.

• **Single-step Flag** (TF) - if set then single-step interrupt will occur after the next instruction.

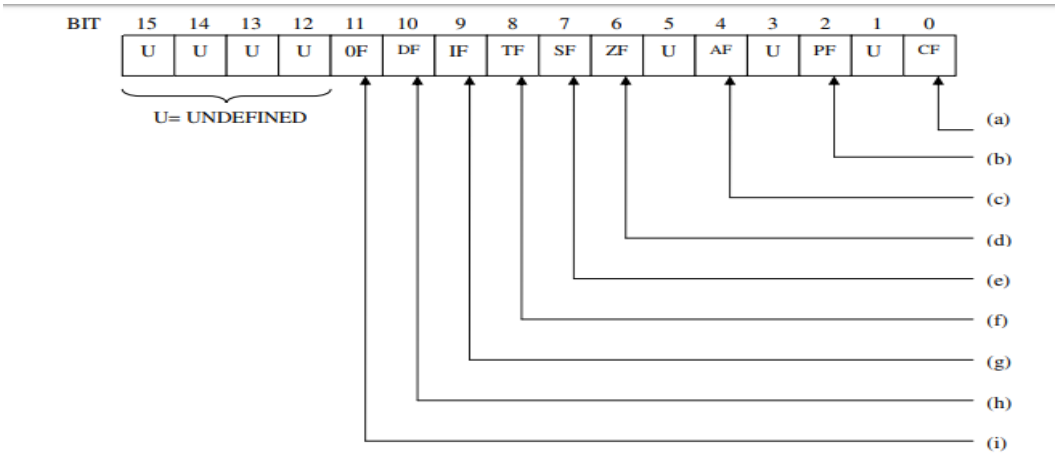
• **Sign Flag** (SF) - set if the most significant bit of the result is set.

• **Zero Flag** (ZF) - set if the result is zero.

• **Auxiliary carry Flag** (AF) - set if there was a carry from or borrow to bits 0-3 in the result of previous operation.

• **Parity Flag** (PF) - set if parity (the number of "1" bits) of the result is even.

• **Carry Flag** (CF) - set if there was a carry from or borrow to the most significant bit during last result calculation.



(a) : CARRY FLAG – SET BY CARRY OUT OF MSB
(b) : PARITY FLAG – SET IF RESULT HAS EVEN PARITY
(c) : AUXILIARY CARRY FLAG FOR BCD
(d) : ZERO FLAG – SET IF RESULT = 0
(e) : SIGN FLAG = MSB OF RESULT
(f) : SINGLE STEP TRAP FLAG
(g) : INTERRUPT ENABLE FLAG
(h) : STRING DIRECTION FLAG
(i) : OVERFLOW FLAG