Chapter two

Internal architecture of the 8086

The internal architecture of microprocessor describe its function components and there interaction . the 8086 internal architecture can be divided into two units these are :-

1- Bus interface unit

Responsible for performing all bus operation such as instruction fetching from memory and data transfer between the processor and the outside word .

2- Execution unit

Responsible for decoding and executing the instruction .

Both unit operate asynchronous to give the 8086 on over lapping instruction fetching and instruction execution mechanism .

This parallel processing of the BIU and EU reduce the fetch time and therefore improve system performance .

1- Bus Interface Unit (BIU)

The BIU is the 8086 interface to the outside word . it provide 0 full 16 bit bidirectional data bus and 20 – bits address bus .

The BIU is responsible for performing all external bus operation these are :-

1- Instruction fetch
2- Instruction queuing
3- Address relocation
4- Operand fetch
5- Bus control

To implement these function , **<u>the BIU contain</u>** the following parts:-

1- Segment register and instruction pointer
2- Object code queue
3- Address summer
4- Bus control logic

<u>Segment register & instruction pointer</u>

The physical address of the 8086 is 20-bits wide , but it register and memory location which contain logical address are just 16-bits wide . therefore the 1M byte physical memory of the 8086 is divided into 16 segment each one of 64-KB and only four segment are active at a time . the BIU segment register contains the starting address of the active segment in memory .

The four segment register are :-

Code segment (CS) register

Which identify the starting address of the code segment in memory the code segment contains program instruction .

2- Data segment (DS) register

Which contain the starting address of the current data segment in which data are stored , it provide a read /write memory space .

3- Stack segment (SS) register

Which contain the starting address for the current stack segment in memory in which the values of the instruction pointer , status flags , and other registers are pushed in case of interrupts or subroutine call .

4- Extra segment (EX) register

Which contains the address for the fourth active segment known extra segment this segment is usually used for data storage .

Instruction Queue

BIU contain an object code queue that permits prefetching of up to six bytes of instruction code . this permits parallel operations of both BIU & EU .

whenever the queue is not full , and the EU is not requesting it to read or write operand from memory , the BIU prefetching the next instructions are stored in first -in first- out (FIFO) queue .

Address summer

The BIU contains also dedicate adder which is used to generate the 30- bits physical address that is output on the address bus . this address is formed by adding the 16- bits segment address and the 16- bits offset address .

Segment address

It is located within one of the segment register and defines the beginning address of any 64- KB memory segment .

Offset address

It select any location within the 64- KB memory segment .

The default combination used in 8086 MP segment offset address showen in the table below :-

Segment	Offset	Purpose
CS	IP	Instruction address
SS	SP or BP	Stack address
DS	BX , DI , SI	Data address
ES	DI	String destination address

Physical address(PA) = [segment][0] + [offset]

EX :If the code segment register contain 1400ʰ and the IP register contains 1200ʰ . find the address of the next instruction to be fetched by the MP.

Sol

PA = 1400 0 + 01200 = 15200ʰ

H.W

If SS: SP = 1234 : 4200

Find the physical address for the memory location in the active stack segment .

Execution unit (EU)

The execution unit is responsible for **decoding and executing** all instructions . the EU **consist of three main part** :-

1- ALU
2- 8- general purpose register
3- Status and control flags (flag register)

The EU extract instructions from the top of queue in the BIU , decoding them , generate operand address if necessary , pass them to the BIU and requests it to perform the read or write bus cycles to memory or I/O , and performs the operation specified by the instruction on the operands , some of the status and control flags may be affected due to the execution of the instructions .