**Module 1**

1. List the registers used in 8086.

**Answer**

8086 has powerful set of registers. It is divided into two:

i) General purpose

ii) Special purpose

General Data Registers

Registers AX, BX, CX and DX are general purpose registers.

The AX, BX, CX, and DX registers can be considers as two 8-bit registers, a High byte and a Low byte.

Data registers: Hold data for an operation to be performed .There are 4 data registers

(AX, BX, CX, DX)

Address registers: Hold the address of an instruction or data element.

Segment registers (CS, DS, ES, SS).

Pointer registers (SP, BP, IP).

Index registers (SI, DI) .

Status register: Keeps the current status of the processor. On an IBM PC the status register is called the FLAGS register .

In total there are fourteen 16-bit registers in an 8086/8088.Instructions execute faster if the data is in a register.

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.

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. 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: 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.

Segment Registers - CS, DS, SS and ES Are Address registers. Store the memory addresses of instructions and data.

1. Describe function of the following signals of 8086:- a) INTR

b) READY c) HOLD

**Answer**

a) READY: This is the acknowledgement from the slow device or memory that they have completed the data transfer. The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086. The signal is active high. 1.

b) INTR-Interrupt Request: This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle. This can be internally masked by resulting the interrupt enable flag. This signal is active high and internally synchronized.

c) HOLD, HLDA- Acknowledge: When the HOLD line goes high, it indicates to the processor that another master is requesting the bus access. The processor, after receiving the HOLD request, issues the hold acknowledge signal on HLDA pin, in the middle of the next clock cycle after completing the current bus cycle. At the same time, the processor floats the local bus and control lines. When the processor detects the HOLD line low, it lowers the HLDA signal. HOLD is an asynchronous input, and is should be externally synchronized.

1. What is the significance of pre decoded instruction byte

queue?

**Answer**

The 8086 architecture has 6-byte instruction pre-fetch queue. Thus even the largest (6 -bytes) instruction can be pre-fetched from the memory and stored in the pre-fetch. This results in a faster execution of the instructions. In 8085 an instruction is fetched, decoded and executed and only after the execution of this instruction, the next one is fetched.

By pre-fetching the instruction, there is a considerable speeding up in instruction execution in 8086. This is known as instruction pipelining.

At the starting the CS: IP is loaded with the required address from which the execution is to be started. Initially, the queue will be empty and the microprocessor starts a fetch operation to bring one byte (the first byte) of instruction code, if the CS:IP address is odd or two bytes at a time, if the CS:IP address is even.

The first byte is a complete opcode in case of some instruction (one byte opcode instruction) and is a part of opcode, in case of some instructions (two byte opcode instructions), the remaining part of code lie in second byte. The second byte is then decoded in continuation with the first byte to decide the instruction length and the number of subsequent bytes to be treated as instruction data.

The queue is updated after every byte is read from the queue but the fetch cycle is initiated by BIU only if at least two bytes of the queue are empty and the EU may be concurrently executing the fetched instructions.

1. Explain the concept of segmented memory? What are it’s

advantages?

**Answer**

The BIU has four 16-bit segment registers. These are the Code Segment (CS) register, the Data Segment (DS) register, the Stack Segment (SS) register, and the Extra Segment (ES) register. The 8086’s one-megabyte memory is divided into segments of up to 64K bytes each. The 8086 can directly address four segments (256K byte within the 1 Mbytes memory) at a particular time. Programs obtain access to code and data in the segments by changing the segment register contents to point to the desired segments. All program instructions must be located in main memory pointed to by the 16-bit CS register with a 16-bit offset in the segment contained in the 16-bit instruction pointer (IP).

1. Explain the use of following signals of 8086:-a) DEN

b) RQ/GT c) LOCK

**Answer**

a) DEN – Data Enable: This signal indicates the availability of valid data over the address/data lines. It is used to enable the trans-receivers(bidirectional buffers)to separate the data from the multiplexed address/data signal. It is active from the middle of T2 until the middle of T4. This is tri-stated during ‘hold acknowledge’ cycle.

b) RQ / GT0 , RQ / GT1 – Request/Grant: These pins are used by the other local bus master in maximum mode, to force the processor to release the local bus at the end of the processor current bus cycle. Each of the pin is bidirectional with RQ/GT0 having higher priority than RQ/GT1. RQ/GT pins have internal pull-up resistors and may be left unconnected.

c) LOCK: This output pin indicates that other system bus master will be prevented from gaining the system bus, while the LOCK signal is low. The LOCK signal is activated by the ‘LOCK’ prefix instruction and remains active until the completion of the next instruction. When the CPU is executing a critical instruction which requires the system bus, the LOCK prefix instruction ensures that other processors connected in the system will not gain the control of the bus.

1. Which are the index registers in 8086? Explain it’s

significance.

**Answer**

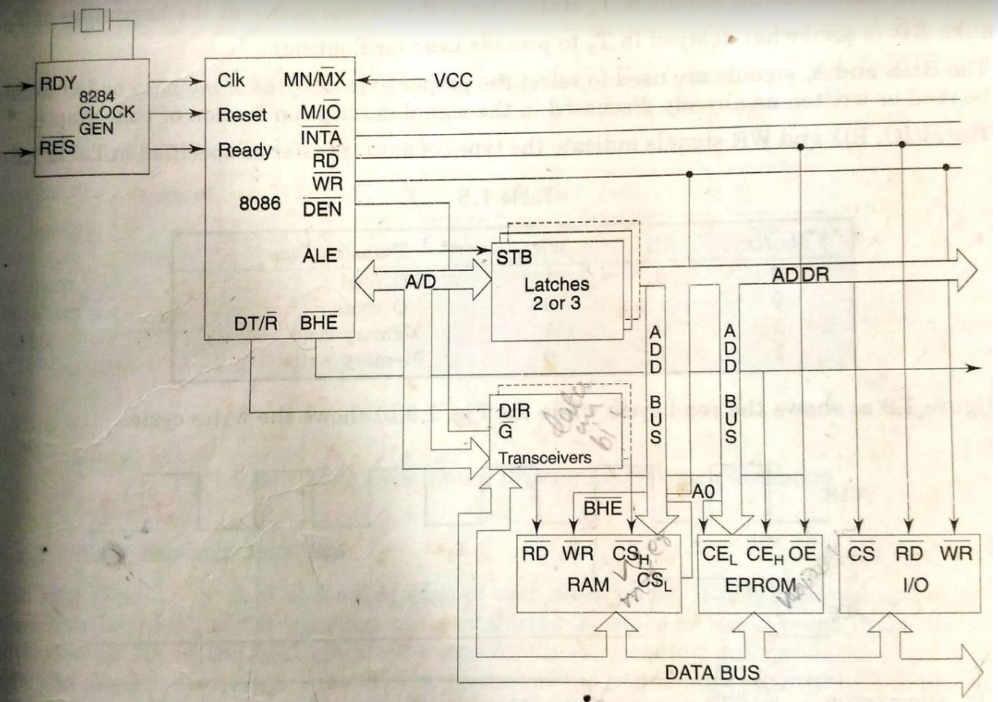
SI: Source Index register: is required for some string operations. When string operations are performed, the SI register points to memory locations in the data segment which is addressed by the DS register. Thus, SI is associated with the DS in string operations.

DI: Destination Index register: 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.

1. Explain the minimum mode working of 8086 with timing

Diagrams?

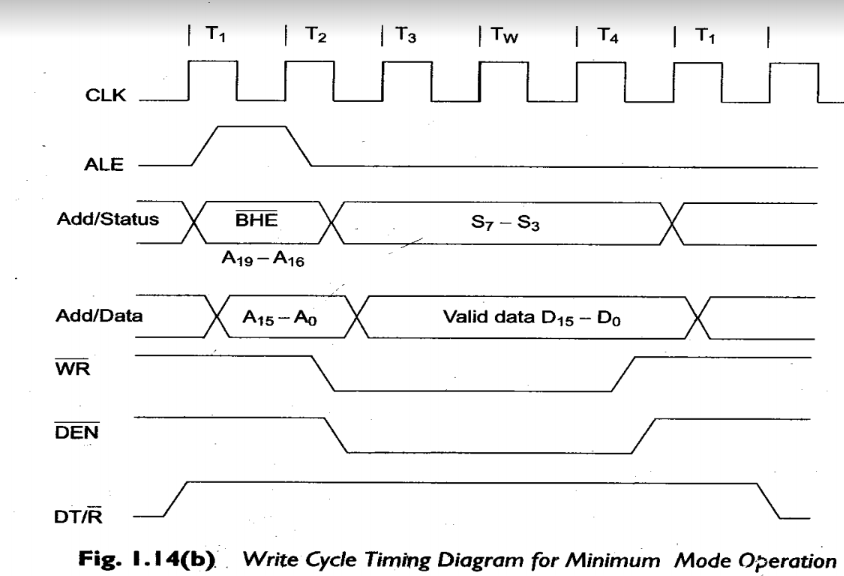
**Answer**

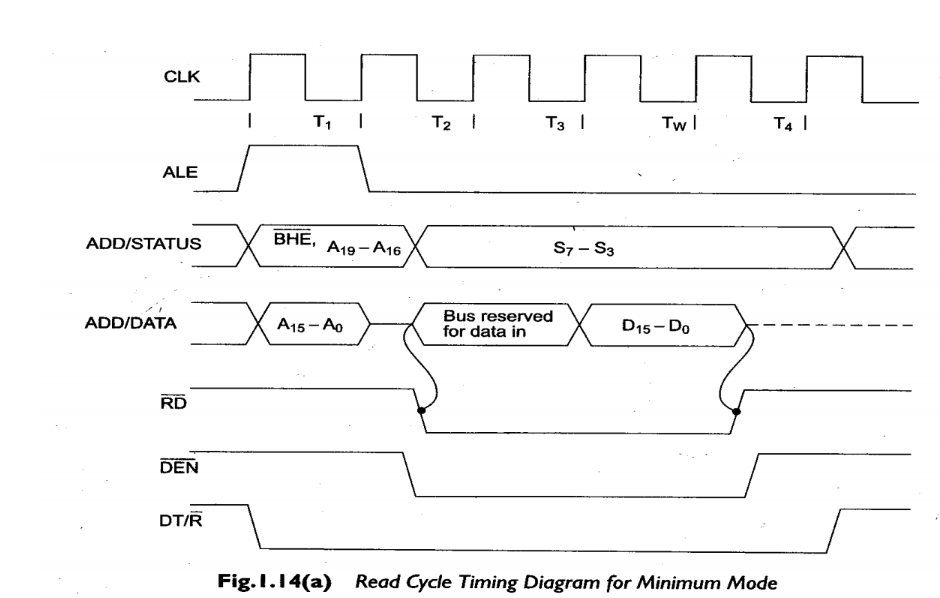


Minimum mode 8086 system

Turns MN/MX pin to logic 1

Control signals are given by microprocessor chip itself. Single microprocessor with latches, trans-receivers, clock generator memory & IO devices. Latches are used to separate valid address from address/data signals controlled by ALE. Trans receivers are bidirectional buffers. Also termed as data amplifiers. Controlled by DEN or DT/R.





1. Discuss the register organization of 8086.

**Answer**

Refer question 1.

1. Draw and discuss the flag register of 8086.

**Answer**

Conditional flags: They are set according to some results of arithmetic operation. You do not need to alter the value yourself.

Control flags: Used to control some operations of the MPU. These flags are to be set by you in order to achieve some specific purposes.

CF (carry) Contains carry from leftmost bit following arithmetic, also contains last bit from a shift or rotate operation.

OF (overflow) Indicates overflow of the leftmost bit during arithmetic.

DF (direction) Indicates left or right for moving or comparing string data.

IF (interrupt) Indicates whether external interrupts are being processed or ignored.

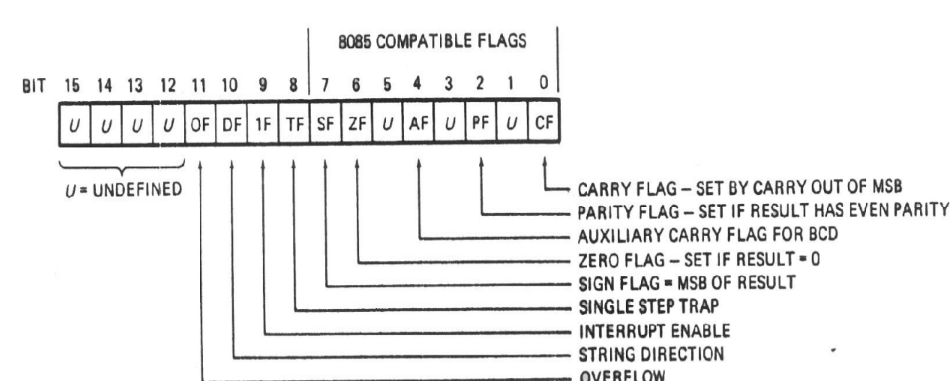
TF (trap) Permits operation of the processor in single step mode.

SF (sign) Contains the resulting sign of an arithmetic operation (1=negative)

ZF (zero) Indicates when the result of arithmetic or a comparison is zero. (1=yes)

AF (auxiliary carry) Contains carry out of bit 3 into bit 4 for specialized arithmetic.

PF (parity) Indicates the number of 1 bits that result from an operation.



1. Draw and explain the internal architecture of 8086.

**Answer**

The 8086 has two parts, the Bus Interface Unit (BIU) and the Execution Unit (EU).The BIU fetches instructions, reads and writes data, and computes the 20-bit address.

The EU decodes and executes the instructions using the 16-bit ALU.

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

The BIU fetches instructions using the CS and IP, written CS:IP, to compute the 20-bit address. Data is fetched using a segment register (usually the DS) and an effective address (EA) computed by the EU depending on the addressing mode.

The EU contains the following 16-bit registers:

AX - the Accumulator

BX - the Base Register

CX - the Count Register

DX - the Data Register

SP - the Stack Pointer \ defaults to stack segment

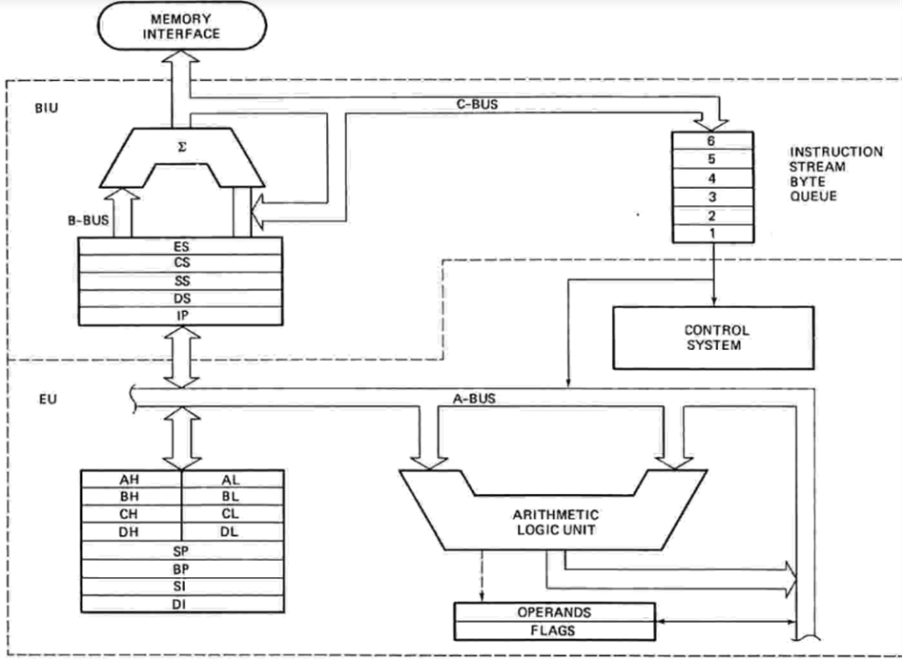
BP - the Base Pointer

SI - the Source Index Register

DI - the Destination Register

These are referred to as general-purpose registers, although, as seen by their names, they often have a special-purpose use for some instructions.

The AX, BX, CX, and DX registers can be considers as two 8-bit registers, a High byte and a Low byte.This allows byte operations and compatibility with the previous generation of 8-bit processors, the 8080 and8085. 8085 source code could be translated in 8086 code and assembled.



Memory Organization

Each byte in memory has a 20 bit address starting with 0 to (2^20 -1) or 1 meg of addressablememory.Addresses are expressed as 5 hex digits from 00000 – FFFFF.

Problem: But 20 bit addresses are TOO BIG to fit in 16 bit registers!.

Solution: Memory Segment .Block of 64K (65,536) consecutive memory bytes.A segment number is a 16 bit number.Segment numbers range from 0000 to FFFF.Within a segment, a particular memory location is specified with an offset.An offset also ranges from 0000 to FFFF .

The BIU has a dedicated adder for determining physical memory addresses

The BIU has four 16-bit segment registers. These are the Code Segment (CS) register, the Data Segment (DS) register, the Stack Segment (SS) register, and the Extra Segment (ES) register.

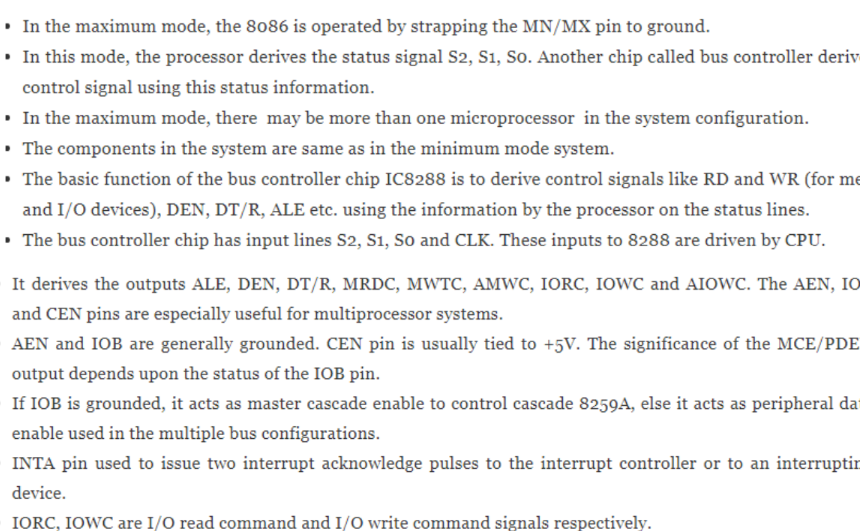
The 8086’s one-megabyte memory is divided into segments of up to 64K bytes each.

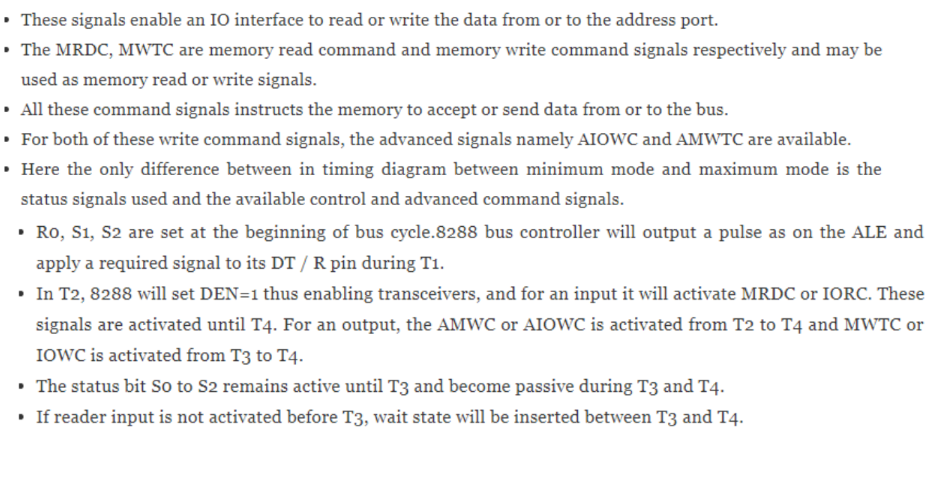
Refer question 1 and 9.

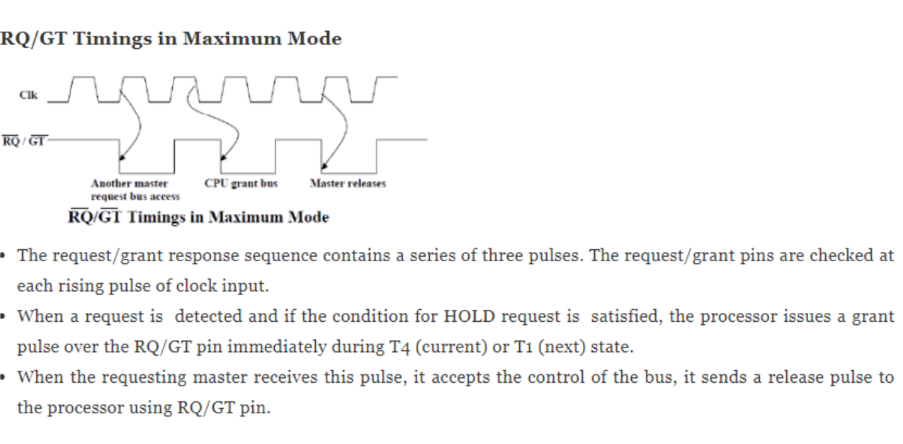
1. With the help of timing diagrams analyze the maximum

mode working of 8086?

**Answer**

****

****



1. a) Explain the procedure for generating physical address and

generate the physical address corresponding to the segment

address 1055H and offset address 5555H. Also find their upper range and lower range.

b) Compare the architectural and signal differences between

8086 and 8088.

**Answer**

a) The 8086 can directly address four segments (256K byte within the 1 Mbytes memory) at a particular time. Programs obtain access to code and data in the segments by changing the segment register contents to point to the desired segments. All program instructions must be located in main memory pointed to by the 16-bit CS register with a 16-bit offset in the segment contained in the 16-bit instruction pointer (IP). The BIU computes the 20-bit physical address internally using the programmer-provided logical address (16-bit contents of CS and IP) by logically shifting the contents of CS four bits to left and then adding the 16-bit contents of IP. In other words, the CS is multiplied by 1610 by the BIU for computing the 20-bit physical address. This means that all instructions of a program are relative to the contents of the CS register multiplied by 16 and then offset is added provided by the 16-bit contents of IP.

For example, if [CS] = 456A16 and [IP] =162016, then the 10-bit physical address is generated by the BIU as follows:

Four times logically shifted [CS] to left = 456A016+ [IP] as offset = 162016

20-bit physical address = 46CC016

The BIU always inserts four Zeros for the lowest 4-bits of the 20-bit starting address (physical) of a segment. In the other words, the CS contains the base or start of the current code segment, and IP contains the distance or offset from this address to the next instruction byte to be fetched. Note that immediate data are considered as part of the code segment.

Another example the physical address of the logical address A4FB:4872 is A4FB0+ 4872 = A9822

In the question, it is given that segment address 1055H and offset address 5555H

Physical address = 10 H \* 1055 +5555= 10550+5555H=15AA5 H

upper range =10 H \* 1055 +FFFF H

lower range=10 H \* 1055 +0000 H

b)

**Difference Table**

|  |  |
| --- | --- |
| **8086** | **8088** |
| 8086 is a 16 bit microprocessor. | 8088 is a 16 bit microprocessor. |
| It has 16 bit data bus. | It has 8 bit data bus. |
| It has 16 bit ALU. | It has 16 bit ALU. |
| 8086 requires memory banking to transfer 16 bit data at a time. | 8088 does not require memory banking as it has an 8 bit data bus. |

**Comparison Table**

|  |  |
| --- | --- |
| 8086 performs faster memory operations as it can transfer 16 bits in one cycle. | 8088 performs slower memory operations as it can transfer only 8 bits in one cycle. |
| 8086 supports pipeline architecture. | 8088 supports pipeline architecture. |
| 8086 has a 6 byte pre-fetch queue  for pipelining. | 8088 has a 4 byte pre-fetch queue  for pipelining. |
| 8086 has an M/ pin to differentiate between memory and I/O operations. | 8088 has an IO/ pin to differentiate between memory and I/O operations. |
| 8086 BIU will fetch new bytes into the Pipelining queue when 2 bytes of the queue are empty. | 8088 BIU will fetch a new byte into the pipelining queue when 1 byte of the queue are empty. |
| 8086 has 9 flags. | 8088 has 9 flags. |