UNIT - I: 8086/8088 MICROPROCESSORS

B. Rajasekhar Reddy

Introduction:

Microprocessor:

It is a semiconductor device consisting of electronic logic circuits manufactured by using either a large scale (LSI) or Very Large-Scale Integration (VLSI) Technique. It includes the ALU, register arrays and control circuits on a single chip.

The microprocessor has a set of instructions, designed internally, to manipulate data and communicate with peripherals. This process of data manipulation and communication is determined by the logic design of the microprocessor called the architecture.

The era microprocessors in the year 1971, the Intel introduced the first 4-bit microprocessor is 4004. Using this first portable calculator is designed. The following table shows the list of Intel microprocessors.

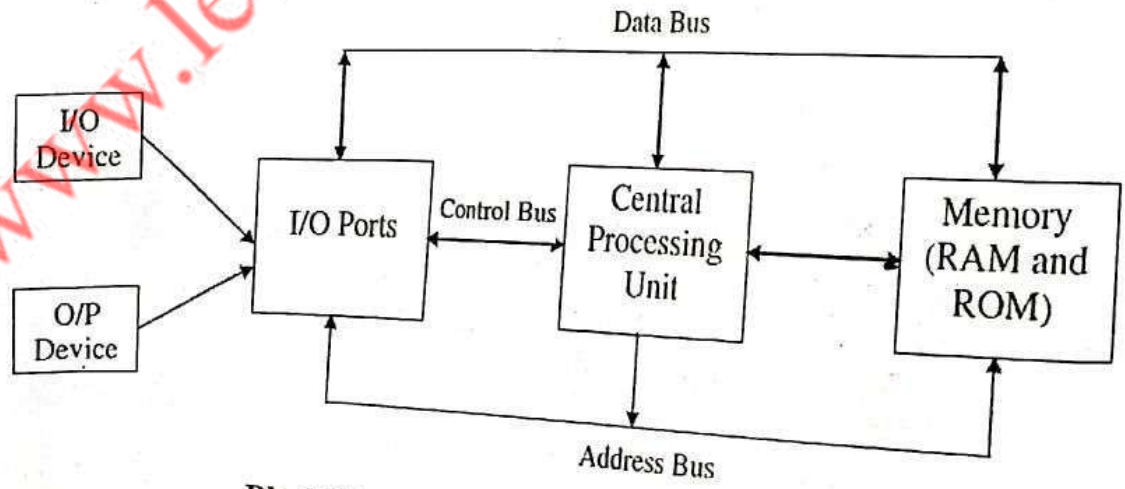
Year	Name	Bit Size
1971	4004	4
1972	8008	8
1974	8080	8
1977	8085	8
1978	8086	16
1979	8089	16
1982	80286	32
1985	80386	32
1989	80486	32
1993	80586(Pentium)	32
1995	Pentium Pro	32
1997	Pentium II	32
1999	Ecleron and Pentium III	32
2000	Pentium IV	32
2001	Intanium	64
2003	Pentium M processor	64
2005	Pentium IV and Xeon	64
2006	Pentium D 900	64

The different manufacturing companies are introduced different bit size microprocessors in the past decade is shown in the table

Company Name	Processor Name
AMD	Athlon
Cypress	CY7C601
DEC	ALPHA
Fujitsu	MBL8086
Harris	CS80C286
LSI Logic	LR 30000
National Semiconductor	NS321016N
SGS-Thomson	ST6X86
SUN-Micro	SRP1030
Texas Instruments	TMS390
Toshiba	TC85R4000
Zilog	Z80
Motorola	68000

A microcomputer system just as any other computer system, include two principal components Hardware and Software.

The memory is used to store both data and instructions that are currently being used. It is normally broken into several modules, each module containing several thousand locations. Each location may contain part or all of a datum or instruction and is associated with an identifier called a memory address. The CPU does its work by successfully inputting, or fetching instructions from memory and carrying out the tasks detected them.



Block Diagram of simple Microcomputer

Above figure shows block diagram of a simple microcomputer. The major parts are the central essing unit or CPU, memory and the input and output circuitry or Input/output. Connecting these parts three sets of parallel line is called buses and control bus. In a microcomputer the CPU is a icroprocessor and is often referred to as the microprocessor unit (MPU). Its purpose is to decode the instruction and use them to control the activity within the system. It performs all arithmetic and logical computations.

Memory: Memory section usually consists of a mixture of RAM and ROM. It may also magnetic floppy disks, magnetic hard disks or optical disks, to store the data.

Input/output: The input/output section allows the computer to take in data from the outside world or send data to the outside world. Peripherals such as keyboard, video display terminals. Printers and modem are connected to the input/output section. These allow the user and computer to communicate with each other. The actual physical devices used to interface the computer buses to external systems are often called ports. An input/output port allows data from keyboard, an analog to digital converter (ADC) or some other source to be read into the computer under the control of the CPU. An output port is used to send data from the computer to some peripheral, such as a video display terminal, a printer or a digital to analog converter (DAC).

Central Processing Unit (CPU): CPU controls the operation of the computer. In a microcomputer the CPU is a microprocessor. The CPU fetches the binary coded instructions from memory, decodes the instructions into a series of simple action and carries out these actions in sequence of steps.

CPU contains an address counter or instruction pointer register which holds the address of the next instruction or data item to be fetched from memory, general purpose register, which are used for temporary storage or binary data and circuitry, which generates the control bus signals.

Address Bus: The address bus consists of 16, 20, 24 or 32 parallel lines. On these lines the CPU sends out the address of the memory locations that are to be written to or read from. The number of memory locations that the CPU can addresses is determined by the number of address lines, then it can directly address 2n memory location. When the CPU reads data from or writes data to a port, it sends the port address on the address bus.

Ex: CPU has 16 address lines can address 216 or 65536 memory locations.

Data Bus: It consists of 8, 16, 32 parallel signal lines. The data bus lines are bidirectional. This means that the CPU can read, data from memory or from a port on these lines, or it can send data out to memory or to port on these lines.

Control Bus: The control bus consists of 4 to 10 parallel signals lines. The CPU sends out signals on the control bus enable the outputs of addressed memory devices or port devices. Typical control bus signal are memory read, memory write, I/O read and I/O write.

Hardware, Software and Firmware: hardware is the given to the physical devices and circuitry of computer. Software refers to collection of programs written for the computer. Firmware is the term gi programs stored in ROM's or in other devices which permanently keep their stored information.

Introduction to 16-bit Microprocessor:

The 16-bit Microprocessor families are designed primarily to complete with microcomputers and a oriented towards high-level languages. Their applications sometimes overlap those of the 8-bi microprocessors. The have powerful instruction sets and capable of addressing megabytes of memory. The era of 16-bit Microprocessors began in 1974 with the introduction of PACE chip by National Semiconductor. The Texas Instruments TMS9900 was introduced in the year 1976. The Intel 8086 commercially available in the year 1978, Zilog Z800 in the year 1979, The Motorola MC68000 in the year 1980.

The 16-bit Microprocessors are available in different pin packages.

Ex:	Intel 8086/8088	40 pin packages
	Zilog Z8001	40 pin packages
	Digital equipment LSI-II	40 pin packages
	Motorola MC68000	64 pin packages
	National Semiconductor NS16000	48 pin packages

The primary objectives of this 16-bit Microprocessor can be summarized as follows.

1. Increase memory addressing capability
2. Increase execution speed
3. Provide a powerful instruction set
4. Facilitate programming in high-level languages.

The INTEL 8086/8088 Features:

- Direct Addressing Capability 1 M Byte of Memory
- Architecture Designed for Powerful
- Assembly Language and Efficient High-Level Languages
- 14 Word, by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Bit, Byte, Word, and Block Operations
- 8 and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal including Multiply and Divide

Range of Clock Rates:

5MHz for 8086,

8MHz for 8086-2,

10 MHz for 8086-1

- MULTIBUS System Compatible Interface
- Available in EXPRESS
  - –Standard Temperature Range
  - –Extended Temperature Range
- Available in 40-Lead CERDIP and Plastic package

It is a 16-bit Microprocessor housed in a 40-pin Dual-Inline-Package (DIP) and capable of addressing 1Megabyte of memory, various versions of this chip can operate with different clock frequencies

i. 8086 (5 MHz)

ii. 8086-2 (8 MHz)

iii. 8086-1 (10 MHz).

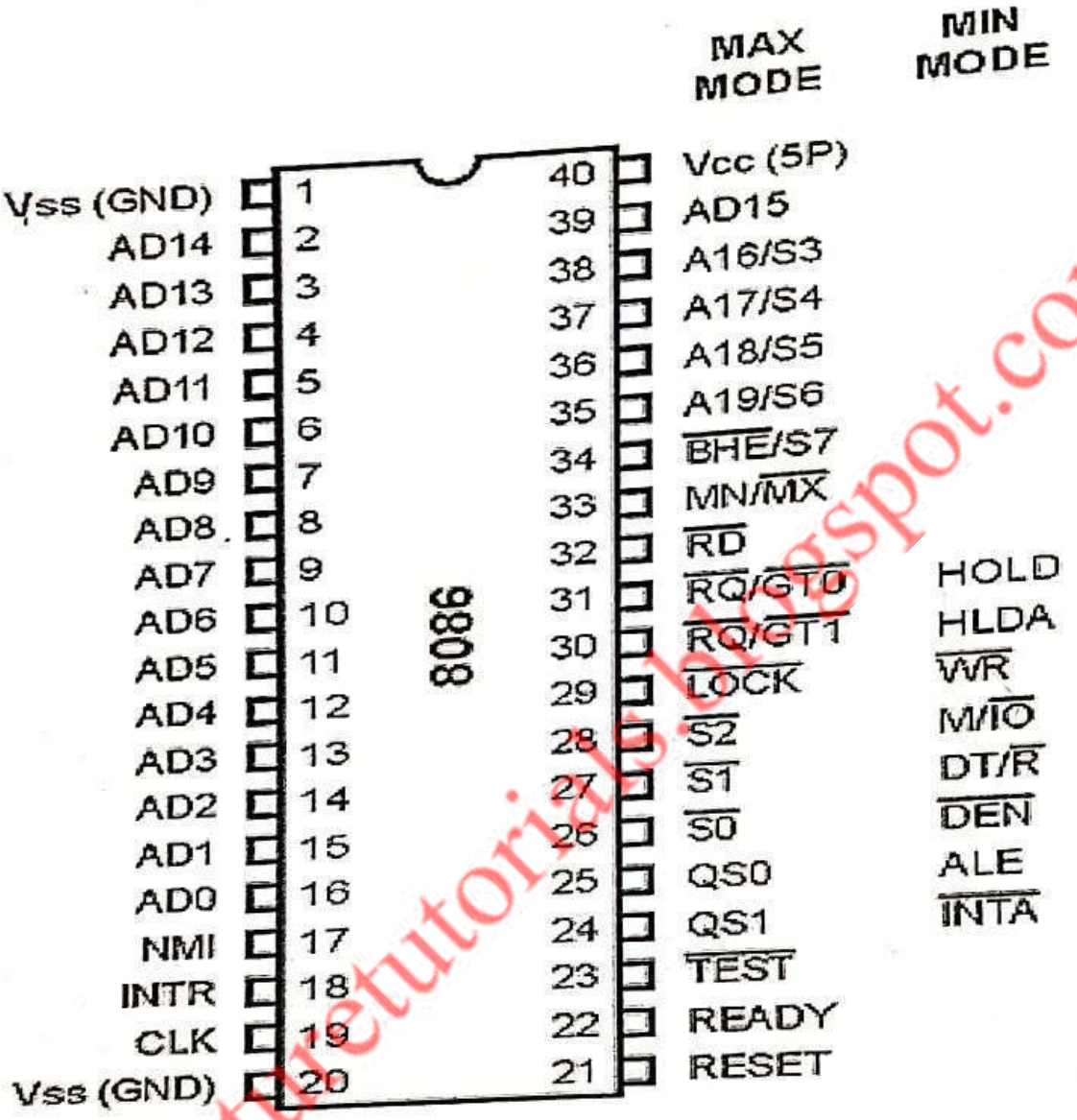
It contains approximately 29,000 transistors and is fabricated using the HMOS technology. The term 16-bit means that it's arithmetic logic unit, its internal registers and most of its instructions are designed to work with 16-bit binary word. The 8086 Microprocessor has a 16-bit data bus, so it can read from or write data to memory and ports either 16-bits or 8-bits at a time. The 8086 Microprocessor has 20-bit address bus, so it can address any one of 220 or 1,048,576 memory locations. Here 16-bit words will be stored in two consecutive memory locations. If the first byte of a word is at an even address, the 8086 can read entire word in one operation, if the first byte of the word is at an odd address the 8086 will read the first byte with one bus operation and the second byte with another bus operation.

Signal Description of 8086 Microprocessor:

The 8086 Microprocessor is a 16-bit CPU available in 3 clock rates, i.e. 5, 8 and 10MHz, packaged in a 40 pin CERDIP or plastic package. The 8086 Microprocessor operates in single processor or multiprocessor configurations to achieve high performance.

The pin configuration is as shown in figure. Some of the pins serve a particular function in minimum mode (single processor mode) and others function in maximum mode (multiprocessor mode) configuration.

The 8086 signals can be categorized in three groups. The first are the signals having common functions in minimum as well as maximum mode, the second are the signals which have special functions in minimum mode and third are the signals having special functions for maximum mode.



The following signals are common for both the minimum and maximum modes.

Power supply and GND

It uses 5V DC supply at Vcc pin 40, and uses ground at Vss pin 1 and 20 for its operation.

Clock signal

Clock signal is provided through Pin-19. It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

ess/data bus

0-AD15, these are 16 address/data bus. AD0-AD7 carries low order byte data and AD8-AD15 carries gher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit lata.

Address/status bus

A16-A19 /S3-S6. These are the 4 address/status buses. During the first clock cycle, it carries 4-bit address and later it carries status signals.

A17/S4	A16/S3	Function
0	0	Alternate Data
0	1	Stack
1	0	Code or none
1	1	Data

S7/$\overline{BHE}$

BHE stands for Bus High Enable. It is available at pin 34 and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

$\overline{BHE}$	A0	Function
0	0	Whole Word
0	1	Upper byte from or to odd address
1	0	Upper byte from or to even address
1	1	None

Read ($\overline{RD}$)

It is available at pin 32 and is used to read signal for Read operation.

Ready

It is available at pin 32. It is an acknowledgement signal from I/O devices that data is transferred. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

RESET

It is available at pin 21 and is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high for the first 4 clock cycles to RESET the microprocessor.

INTR

It is available at pin 18. It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine if the processor considered this as an interrupt or not.

NMI

It stands for non-maskable interrupt and is available at pin 17. It is an edge triggered input, which causes an interrupt request to the microprocessor.

TEST

This signal is like wait state and is available at pin 23. When this signal is high, then the processor has to wait for IDLE state, else the execution continues.

MN/$\overline{MX}$

It stands for Minimum/Maximum and is available at pin 33. It indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-versa.

Minimum Mode Signals:

$\overline{INTA}$

It is an interrupt acknowledgement signal and id available at pin 24. When the microprocessor receives this signal, it acknowledges the interrupt.

ALE

It stands for address enable latch and is available at pin 25. A positive pulse is generated each time the processor begins any operation. This signal indicates the availability of a valid address on the address/data lines.

$\overline{DEN}$

It stands for Data Enable and is available at pin 26. It is used to enable Transreceiver 8286. The transceiver is a device used to separate data from the address/data bus.

DT/$\overline{R}$

It stands for Data Transmit/Receive signal and is available at pin 27. It decides the direction of data flow through the transceiver. When it is high, data is transmitted out and vice-a-versa.

M/$\overline{IO}$

This signal is used to distinguish between memory and I/O operations. When it is high, it indicates I/O operation and when it is low indicating the memory operation. It is available at pin 28.

ands for write signal and is available at pin 29. It is used to write the data into the memory or the tput device depending on the status of M/$\overline{IO}$ signal.

HLDA

It stands for Hold Acknowledgement signal and is available at pin 30. This signal acknowledges the HOLD signal.

HOLD

This signal indicates to the processor that external devices are requesting to access the address/data buses. It is available at pin 31.

Maximum Mode Signals:

QS_1 and QS_0

These are queue status signals and are available at pin 24 and 25. These signals provide the status of instruction queue. Their conditions are shown in the following table –

QS_0	QS_1	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

$\bar{S}_0$, $\bar{S}_1$, $\bar{S}_2$

These are the status signals that provide the status of operation, which is used by the Bus Controller 8288 to generate memory & I/O control signals. These are available at pin 26, 27, and 28. Following is the table showing their status –

$\bar{S}_2$	$\bar{S}_1$	$\bar{S}_0$	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

LOCK

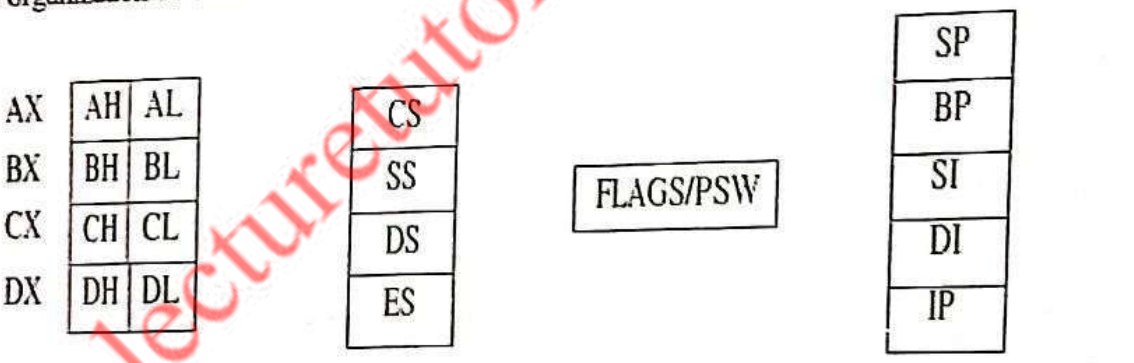
When this signal is active, it indicates to the other processors not to ask the CPU to leave the system bus. It is activated using the LOCK prefix on any instruction and is available at pin 29.

$\overline{RQ}/\overline{GT_1}$ and $\overline{RQ}/\overline{GT_0}$

These are the Request/Grant signals used by the other processors requesting the CPU to release the system bus. When the signal is received by CPU, then it sends acknowledgment. $\overline{RQ}/\overline{GT_0}$ has a higher priority than $\overline{RQ}/\overline{GT_1}$.

Register Organization of 8086:

8086 has a powerful set of registers containing general purpose and special purpose registers. All the registers of 8086 are 16-bit registers. The general-purpose registers, can be used either 8-bit registers or 16-bit registers. The general-purpose registers are either used for holding the data, variables and intermediate results temporarily or for other purpose like counter or for storing offset address for some particular addressing modes etc. The special purpose registers are used as segment registers, pointers, index registers or as offset storage registers for particular addressing modes. The below figure shows register organization of 8086. We will categorize the register set into four groups as follows:



Register Organization of 8086 Microprocessor

General Data Registers:

The registers AX, BX, CX, and DX are the general 16-bit registers.

AX Register: Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16- bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations, rotate and string manipulation.

egister: This register is mainly used as a **base register**. It holds the starting base location of a ory region within a data segment. It is used as offset storage for forming physical address in case of ain addressing mode.

X Register: It is used as default counter or **count register** in case of string and loop instructions.

DX Register: Data register can be used as a port number in I/O operations and implicit operand or destination in case of few instructions. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

Segment Registers:

To complete 1Mbyte memory is divided into 16 logical segments. The complete 1Mbyte memory segmentation is as shown in figure. Each segment contains 64Kbyte of memory. There are four segment registers.

Code Segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.

Stack Segment (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction. It is used for addressing stack segment of memory. The stack segment is that segment of memory, which is used to store stack data.

Data Segment (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions. It points to the data segment memory where the data is resided.

Extra Segment (ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions. It also refers to segment which essentially is another data segment of the memory. It also contains data.

Pointers and Index Registers:

The pointers contain within the particular segments. The pointers IP, BP, SP usually contain offsets within the code, data and stack segments respectively.

Stack Pointer (SP) is a 16-bit register pointing to program stack in stack segment.

Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.

Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions.

Flag Register:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	1
X	X	X	X	OF	DF	IF	TF	SF	ZF	X	AC	X	PF	X	CY

X = Undefined

Flag Register of 8086

Register determines the current state of the processor. They are modified automatically by CPU mathematical operations, this allows to determine the type of the result, and to determine conditions transfer control to other parts of the program.

The 8086-flag register as shown in the fig 1.6. 8086 has 9 active flags and they are divided into two categories:

1. Conditional Flags/ Status Flags
2. Control Flags

Conditional Flags

Conditional flags are as follows:

Carry Flag (CY): This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.

Auxiliary Flag (AC): If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), the AC flag is set i.e. carry given by D3 bit to D4 is AC flag. This is not a general-purpose flag, it is used internally by the Processor to perform Binary to BCD conversion.

Parity Flag (PF): This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1's, the Parity Flag is set and for odd number of 1's, the Parity flag is reset.

Zero Flag (ZF): It is set; if the result of arithmetic or logical operation is zero else it is reset.

Sign Flag (SF): In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set.

Control Flags

Control flags are set or reset deliberately to control the operations of the execution unit. Control flags are as follows:

Trap Flag (TF): It is used for single step control. It allows user to execute one instruction of a program at a time for debugging. When trap flag is set, program can be run in single step mode.

Interrupt Flag (IF): It is an interrupt enable/disable flag. If it is set, the maskable interrupt of 8086 is enabled and if it is reset, the interrupt is disabled. It can be set by executing instruction SIT and can be cleared by executing CLI instruction.

Direction Flag (DF): It is used in string operation. If it is set, string bytes are accessed from higher memory address to lower memory address. When it is reset, the string bytes are accessed from lower memory address to higher memory address.

Internal Architecture of 8086 Microprocessor:

The internal architecture 8086 microprocessor is as shown in the figure. The 8086 CPU is divided into two independent functional parts, the Bus interface unit (BIU) and execution unit (EU).

The Bus Interface Unit contains Bus Interface Logic, Segment registers, Memory addressing logic and a Six-byte instruction object code queue. The execution unit contains the Data and Address registers, the Arithmetic and Logic Unit, the Control Unit and flags.

The BIU sends out address, fetches the instructions from memory, read data from ports and memory, and writes the data to ports and memory. In other words, the BIU handles all transfers of data and addresses on the buses for the execution unit.

The execution unit (EU) of the 8086 tells the BIU where to fetch instructions or data from, decodes instructions and executes instruction. The EU contains control circuitry which directs internal operations. A decoder in the EU translates instructions fetched from memory into a series of actions which the EU carries out. The EU is having a 16-bit ALU which can add, subtract, AND, OR, XOR, increment, decrement, complement or shift binary numbers. The EU is decoding an instruction or executing an instruction which does not require use of the buses.

The Queue: The BIU fetches up to 6 instruction bytes for the following instructions. The BIU stores these prefetched bytes in first-in-first-out register set called a queue. When the EU is ready for its next instruction it simply reads the instruction byte(s) for the instruction from the queue in the BIU. This is much faster than sending out an address to the system memory and waiting for memory to send back the next instruction byte or bytes. Except in the case of JMP and CALL instructions, where the queue must be dumped and then reloaded starting from a new address, this prefetch-and-queue scheme greatly speeds up processing. Fetching the next instruction while the current instruction executes is called pipelining.

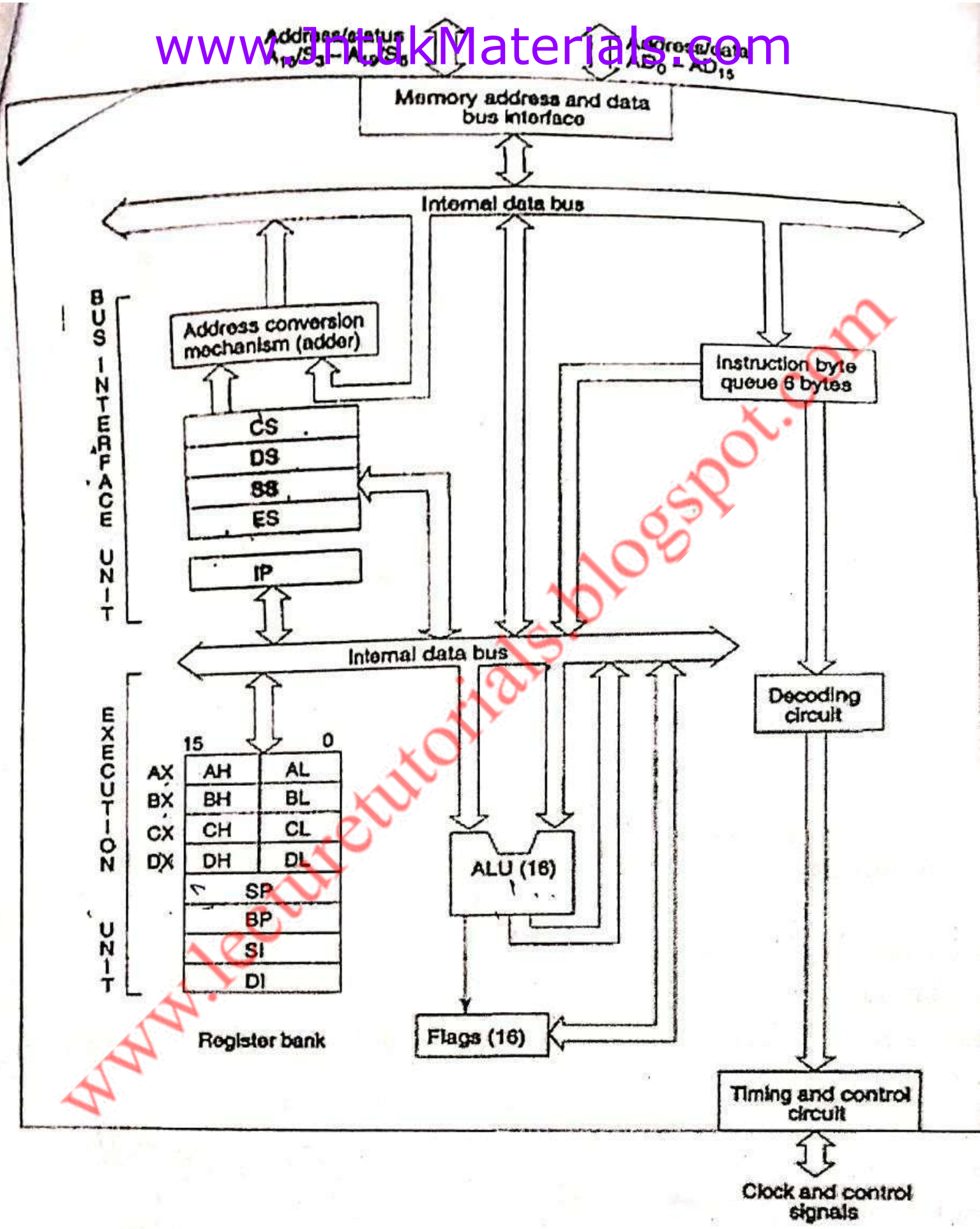


Fig 1: Internal Architecture of 8086 Microprocessor

Word Read

Each of 1 MB memory address of 8086 represents a byte wide location. 16-bit words will be stored in consecutive memory locations. If first byte of the data is stored at an even address, 8086 can read the entire word in one operation.

For example, if the 16-bit data is stored at even address 00520H is 9634H

MOV BX, [00520H]

8086 reads the first byte and stores the data in BL and reads the 2nd byte and stores the data in BH

BL= (00520H) i.e. BL=34H

BH= (00521H) BH=96H

If the first byte of the data is stored at an odd address, 8086 needs two operations to read the 16-bit data.

For example, if the 16-bit data is stored at even address 00521H is 3897H

MOV BX, [00521H]

In first operation, 8086 reads the 16-bit data from the 00520H location and stores the data of 00521H location in register BL and discards the data of 00520H location In 2nd operation, 8086 reads the 16 bit data from the 00522H location and stores the data of 00522H location in register BH and discards the data of 00523H location.

BL= (00521H) i.e. BL=97H

BH= (00522H) BH=38H

Byte Read:

MOV BH, [Addr]

For Even Address:

Ex: **MOV BH, [00520H]**

8086 reads the first byte from 00520 locations and stores the data in BH and reads the 2nd byte from the 00521H location and ignores it

BH = [00520H]

For Odd Address

Ex: **MOV BH, [00521H]**

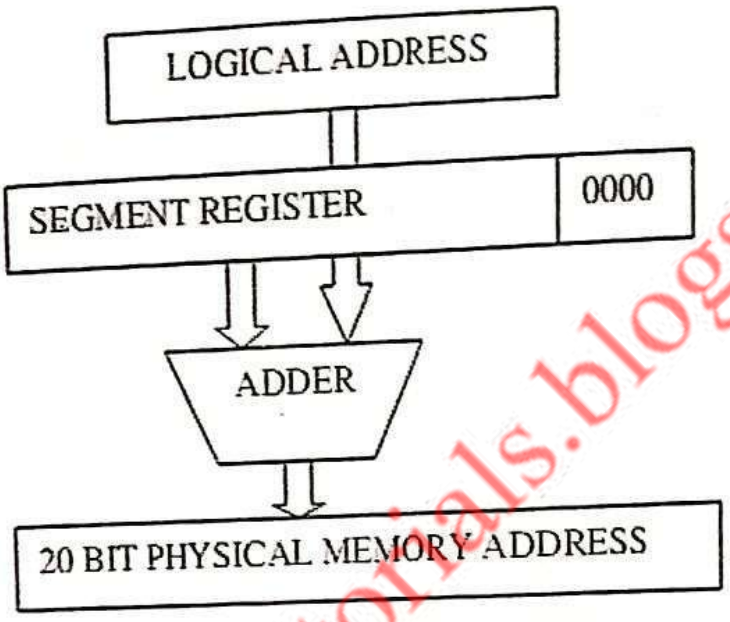
8086 reads the first byte from 00520H location and ignores it and reads the 2nd byte from the 00521 locations and stores the data in BH

BH = [00521H]

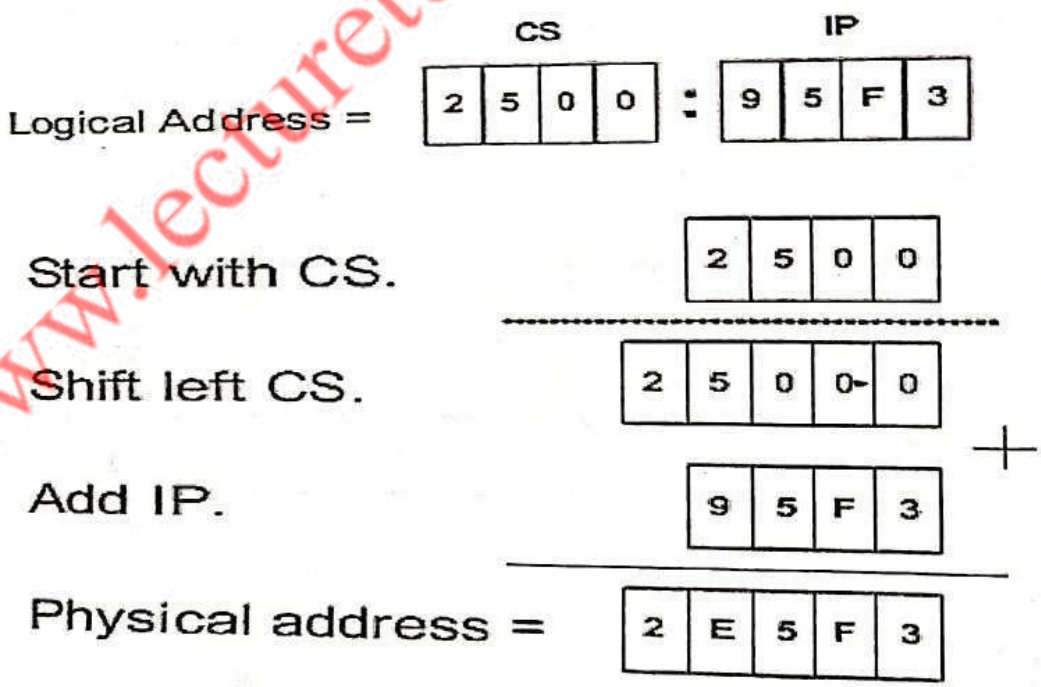
cal Address formation:

The 8086 addresses a segmented memory. The complete physical address which is 20-bits long is erated using segment and offset registers each of the size 16-bit. The content of a segment register also lled as segment address, and content of an offset register also called as offset address. To get total physical address, put the lower nibble 0H to segment address and add offset address. The figure shows formation of 20-bit physical address.

GENERATION OF 20-BIT PHYSICAL ADDRESS

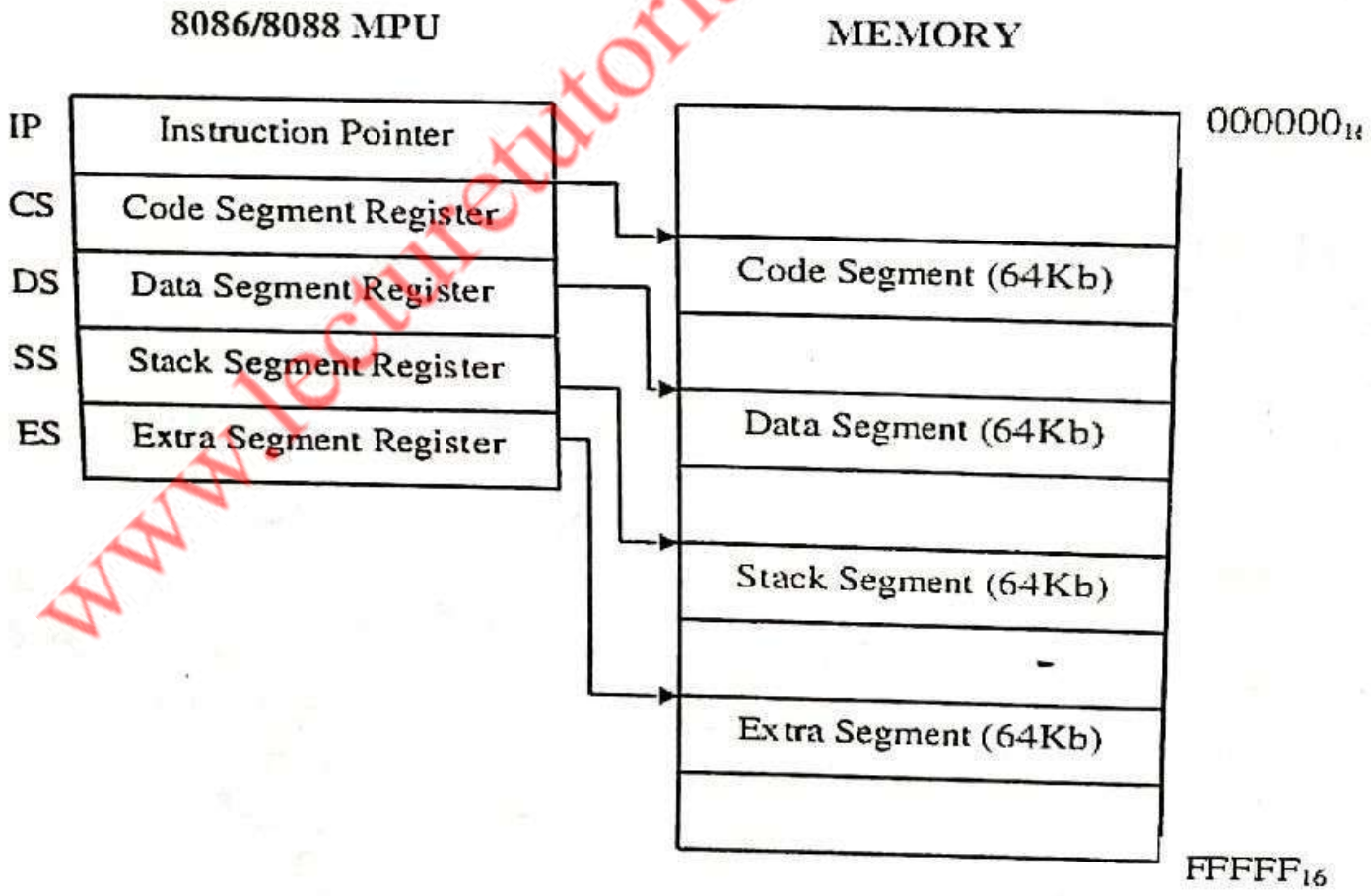


Physical Address Calculation (i):

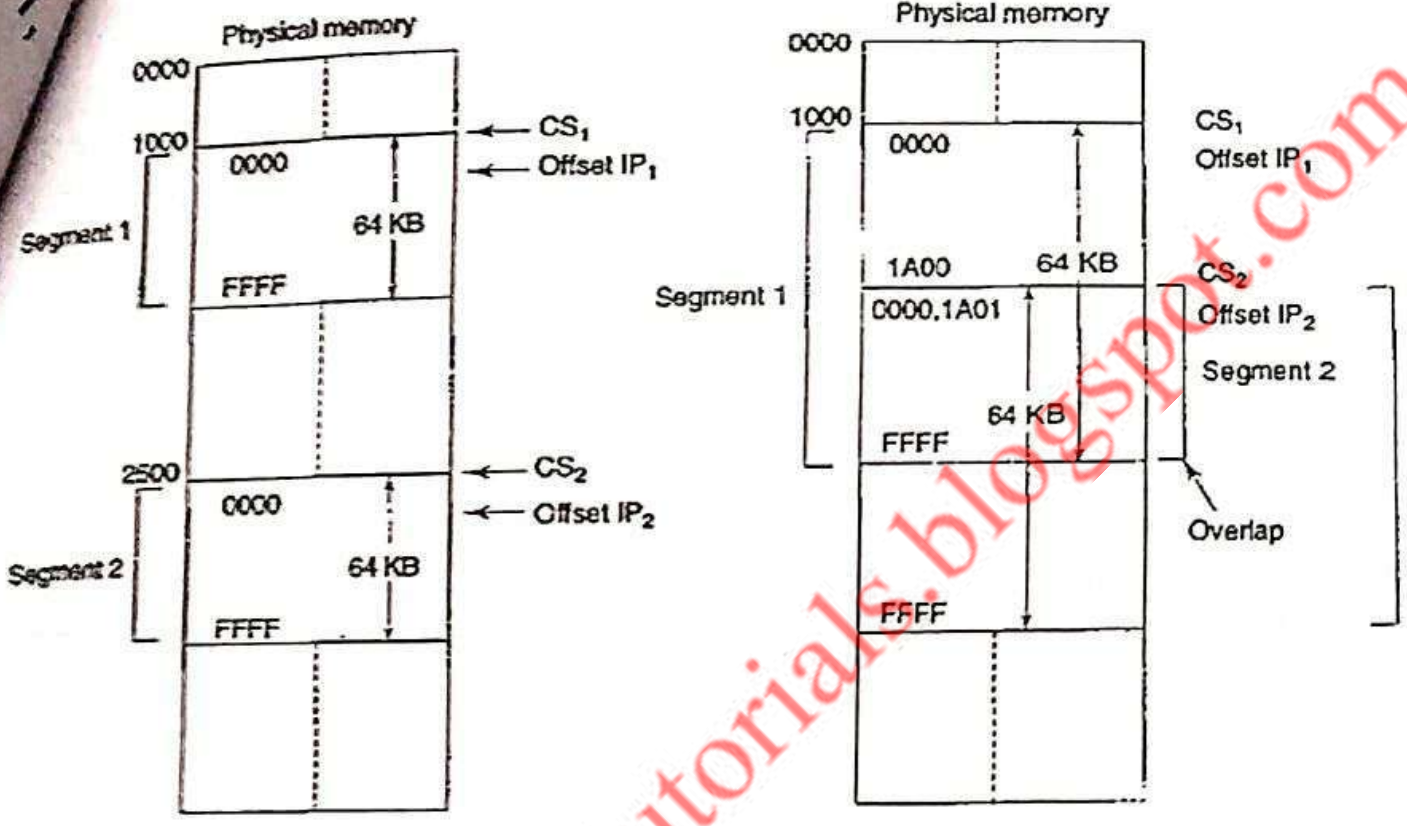


Physical Address Calculation (ii):

- If DS = 7FA2H and the offset is 438EH,
  - The logical address:
    7FA2:438E
  - The physical address:
    7FA20 + 438E = 83DAE
  - The upper range of the data segment:
    7FA20 + FFFF = 8FA1F
  - The lower range of the data segment:
    7FA20 + 0000 = 7FA20



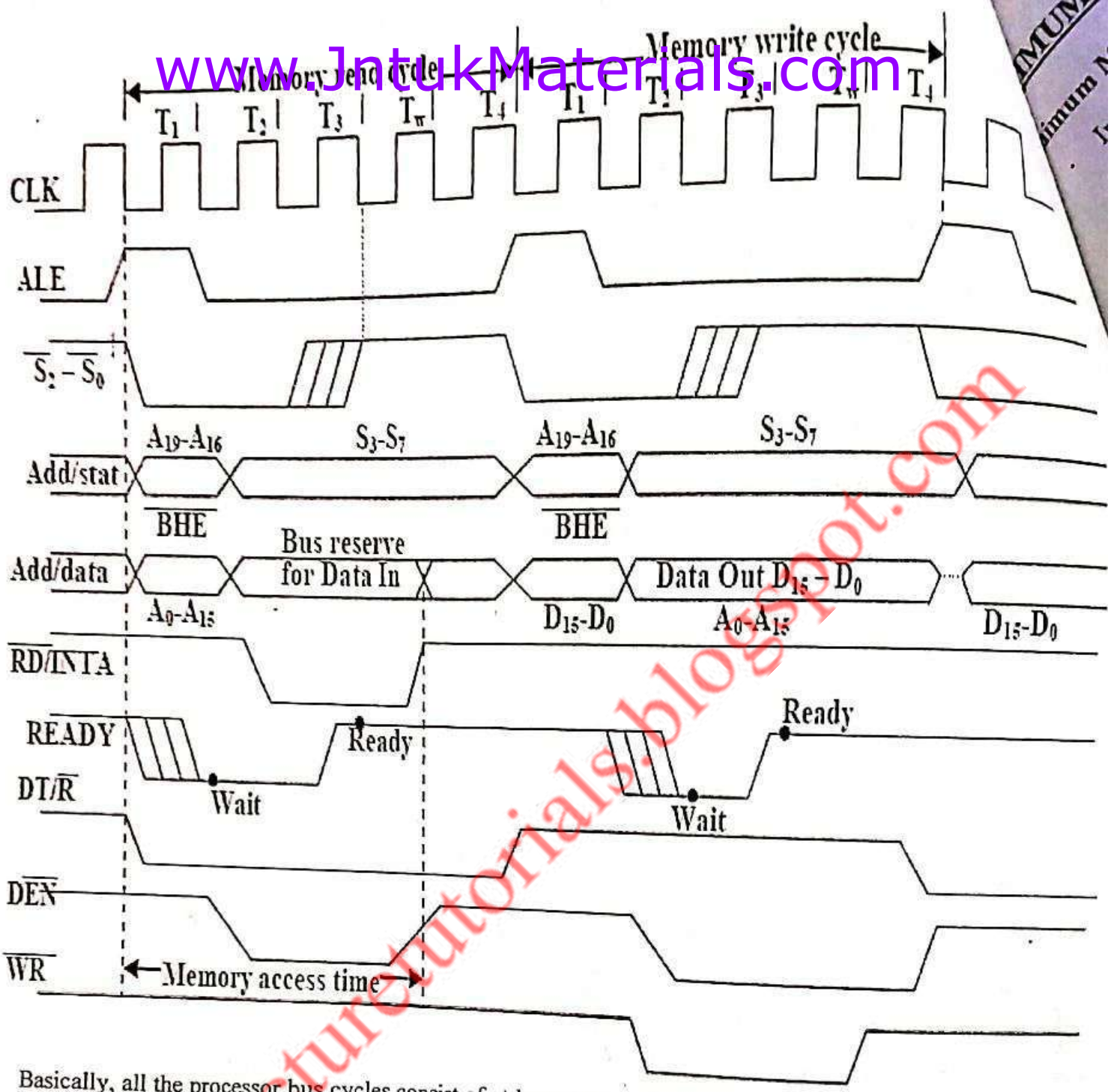
...mory Segmentation:



(a) *Non-overlapping Segments* (b) *Overlapping Segments*

General Bus Operation:

- The 8086 has a combined address and data bus commonly referred as a time multiplexed address and data bus.
- The main reason behind multiplexing address and data over the same pins is the maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package.
- The bus can be de-multiplexed using a few latches and transceivers, whenever required.



- Basically, all the processor bus cycles consist of at least four clock cycles. These are referred to as T1, T2, T3, T4. The address is transmitted by the processor during T1. It is present on the bus only for one cycle. The negative edge of this ALE pulse is used to separate the address and the data or status information. In maximum mode, the status lines $\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$ are used to indicate the type of operation.
- Status bits S3 to S7 are multiplexed with higher order address bits and the BHE signal. Address is valid during T1 while status bits S3 to S7 are valid during T2 through T4.

IUM MODE 8086 SYSTEM AND TIMINGS:

mum Mode

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/$\overline{MX}$ pin to logic 1.
- In this mode, all the control signals are given out by the microprocessor chip itself.

In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/$\overline{MX}$ pin to logic1. In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system. The remaining components in the system are latches, transceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.

The latches are generally buffered output D-type flip-flops, like, 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086. Transceivers are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signal. They are controlled by two signals, namely, $\overline{DEN}$ and DT/$\overline{R}$. The $\overline{DEN}$ signal indicates that the valid data is available on the data bus, while DT/$\overline{R}$ indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage. Usually, EPROMs are used for monitor storage, while RAMs for user's program storage. A system may contain I/O devices for communication with the processor as well as some special purpose I/O devices. The clock generator generates the clock from the crystal oscillator and then shapes it and divides to make it more precise so that it can be used as an accurate timing reference for the system. The clock generator also synchronizes some external signals with the system clock. The general system organization is shown in Fig. 1.1. Since it has 20 address lines and 16 data lines, the 8086 CPU requires three octal address latches and two octal data buffers for the complete address and data separation.

The working of the minimum mode configuration system can be better described in terms of the timing diagrams rather than qualitatively describing the operations. The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle. Fig 1.1(a) shows the read cycle timing diagram.

- The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal and also M/$\overline{IO}$ signal. During the negative going edge of this signal, the valid address is latched on the local bus. The $\overline{BHE}$ and A0 signals address low, high or both bytes.

- From T1 to T4, the M/$\overline{IO}$ signal indicates a memory or I/O operation.
- At T2 the address is removed from the local bus and is sent to the output. The bus is then tri-stated. The read ($\overline{RD}$) control signal is also activated in T2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers.
- After $\overline{RD}$ goes low, the valid data is available on the data bus. The addressed device will drive the READY line high, when the processor returns the read signal to high level, the addressed device will again tri-state its bus drivers.

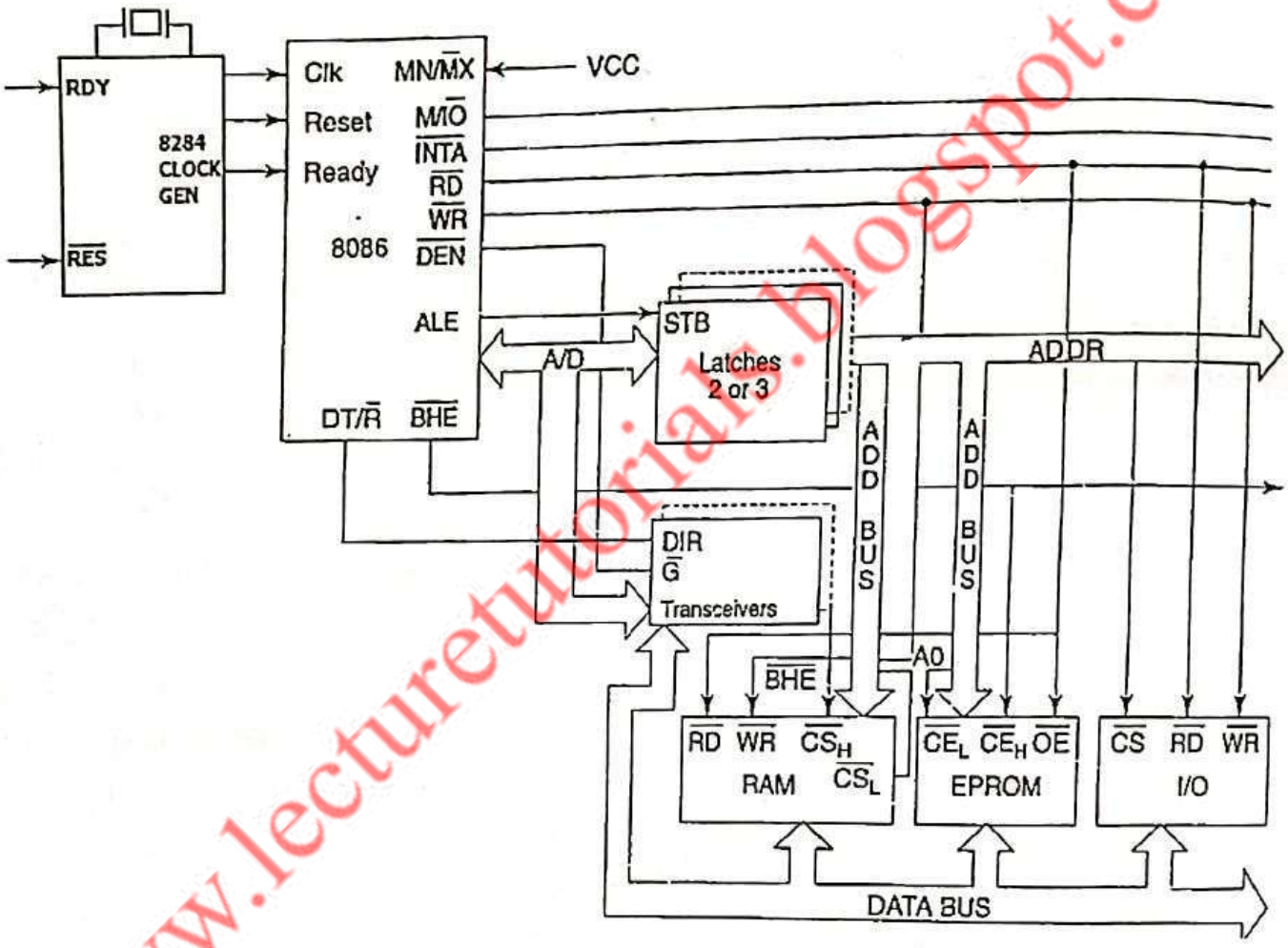


Fig 1.1: Minimum Mode 8086 System

Fig 1.1 (b) shows the write cycle timing diagram. A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O operation.

- In T2 after sending the address in T1 the processor sends the data to be written to the addressed location.

...e data remains on the bus until middle of T4 state. The $\overline{WR}$ becomes active at the beginning of T2 (unlike $\overline{RD}$ is somewhat delayed in T2 to provide time for floating).

The $\overline{BHE}$ and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or written. The M/$\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ signals indicate the types of data transfer as specified in Table

M/$\overline{IO}$	$\overline{RD}$	$\overline{WR}$	Transfer Type
0	0	1	I/O read
0	1	0	I/O write
1	0	1	Memory read
1	1	0	Memory write

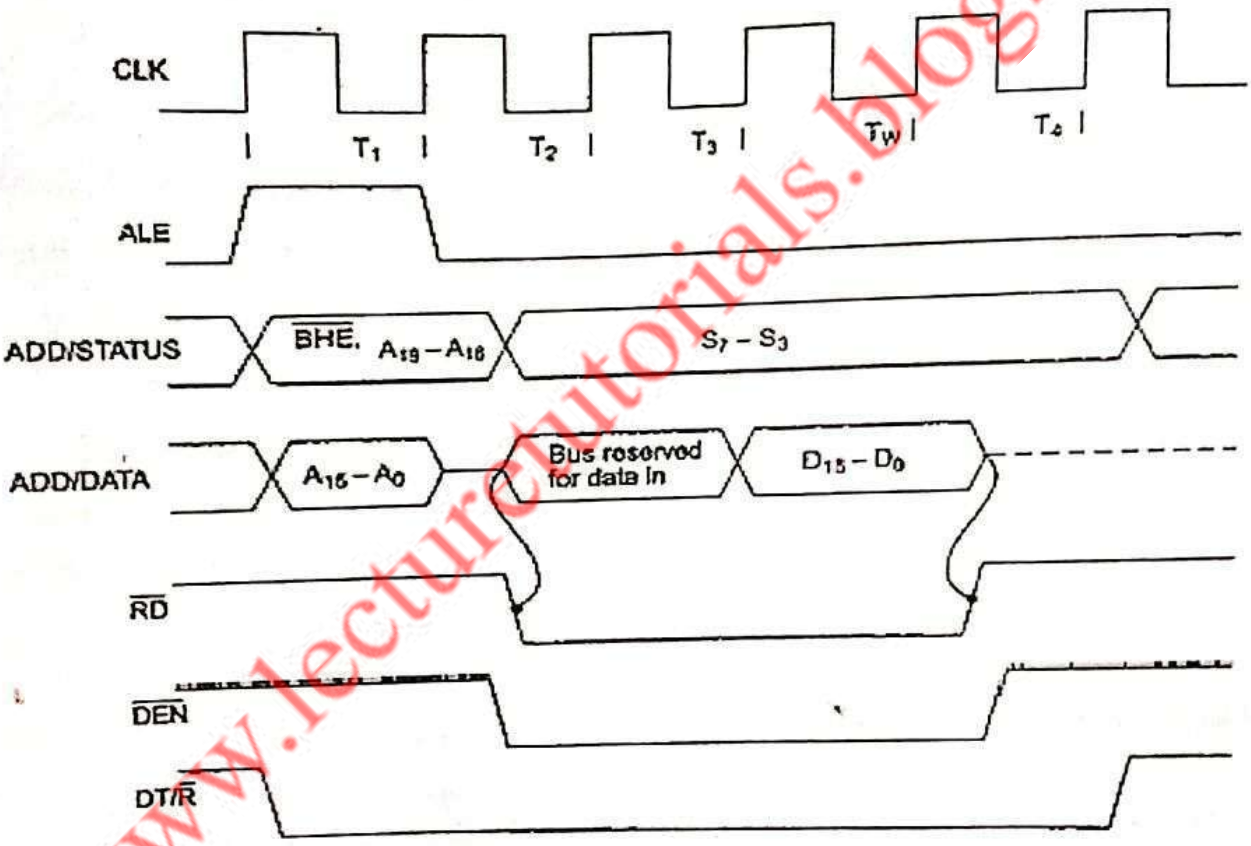


Fig.1.1 (a): Read Cycle Timing Diagram for Minimum Mode

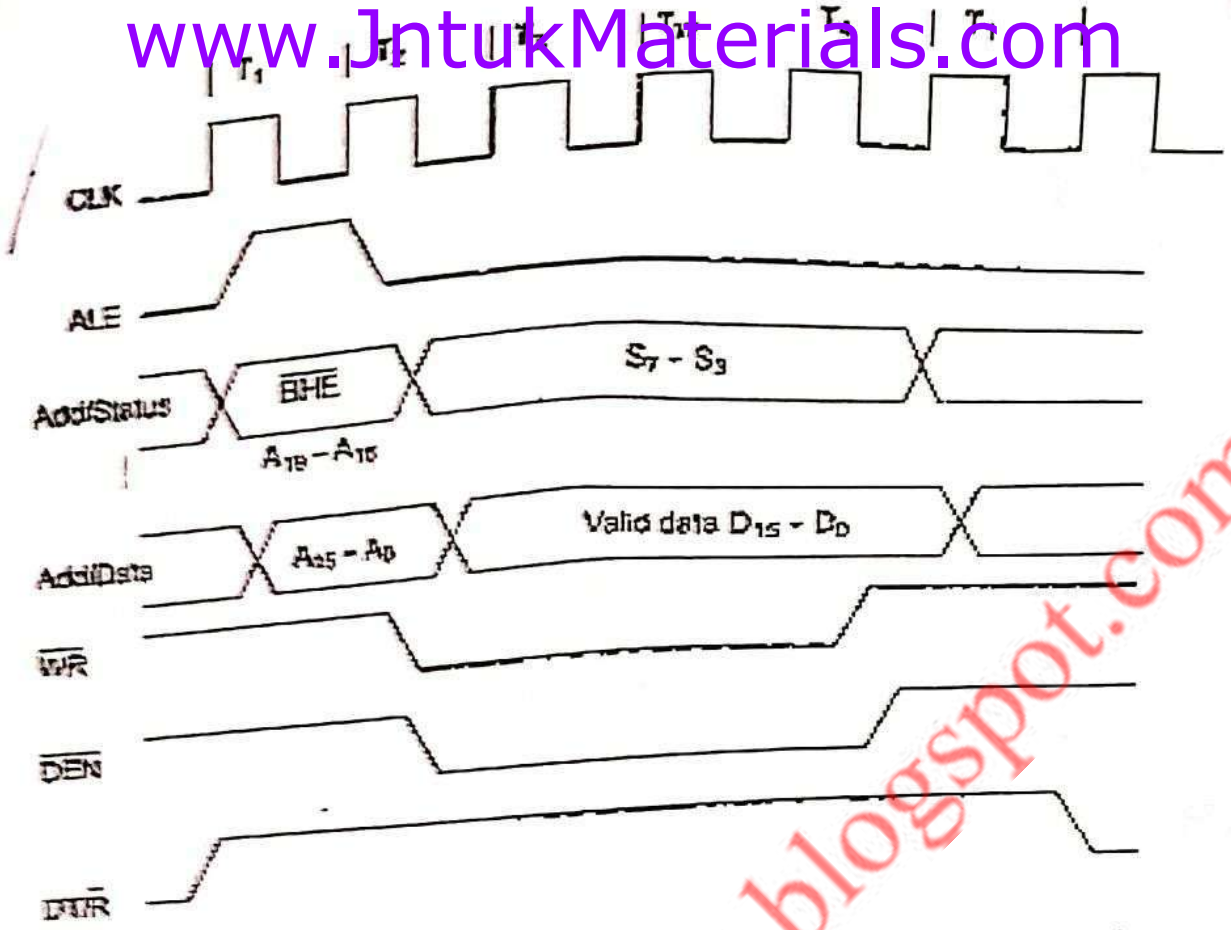


Fig.1.1 (b): Write Cycle Timing Diagram for Minimum Mode

HOLD Response Sequence

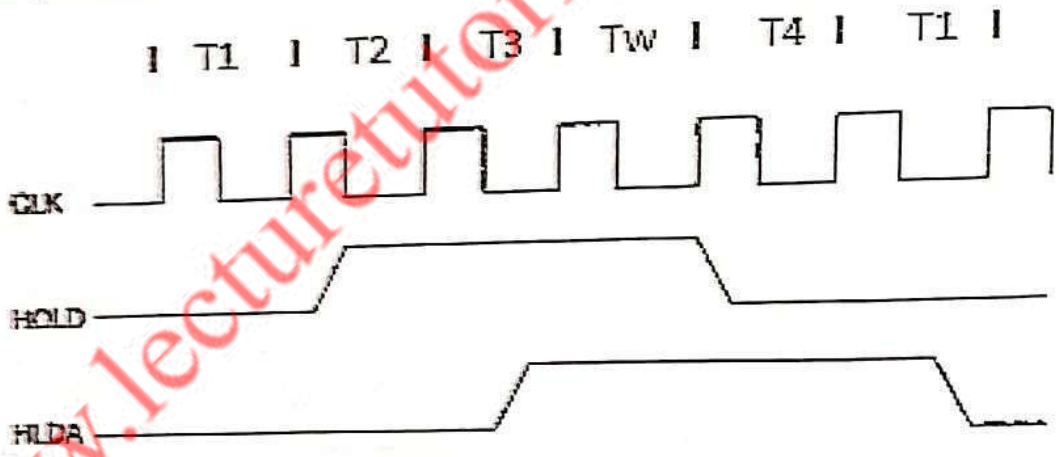


Fig 1.1(c): Bus Request and Bus Grant Timings in Minimum Mode System

The HOLD pin is checked at the end of each bus cycle. If it is received active by the processor before T4 of the previous cycle or during T1 state of the current cycle, the CPU activities HLDA in the next clock cycle and for the succeeding bus cycles, the bus will be given to another requesting master the control of the bus is not regained by the processor until the requesting master does not drop the HOLD pin low. When the request is dropped by the requesting master, the HLDA is dropped by the processor at the trailing edge of the next clock as shown in fig 1.1 (c).

Maximum Mode

In the maximum mode, the 8086 is operated by strapping the MN/$\overline{MX}$ pin to ground.

- In this mode, the processor derives the status signal $\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$. Another chip called bus controller derives the control signal using this status information.
- In the maximum mode, there may be more than one microprocessor in the system configuration.

In the maximum mode, the 8086 is operated by strapping the MN/$\overline{MX}$ pin to ground. In this mode, the processor derives the status signals $\overline{S}_2$, $\overline{S}_1$ and $\overline{S}_0$. Another chip called bus controller derives the control signals using this status information. In the maximum mode, there may be more than one microprocessor in the system configuration. The other components in the system are the same as in the minimum mode system. The general system organization is as shown in the fig1.2.

The basic functions of the bus controller chip IC 8288, is to derive control signals like $\overline{RD}$ and $\overline{WR}$ (for memory and I/O devices), $\overline{DEN}$, DT/$\overline{R}$, ALE, etc. using the information made available by the processor on the status lines. The bus controller chip has input lines $\overline{S}_2$, $\overline{S}_1$ and $\overline{S}_0$ and CLK. These inputs to 8288 are driven by the CPU. It derives the outputs ALE, $\overline{DEN}$, DT/$\overline{R}$, $\overline{MWTC}$, $\overline{AMWTC}$, $\overline{IORC}$, $\overline{IOWC}$ and $\overline{AIOWC}$. The $\overline{AEN}$, IOB and CEN pins are especially useful for multiprocessor systems. $\overline{AEN}$ and IOB are generally grounded. CEN pin is usually tied to +5V.

The significance of the MCE/$\overline{PDEN}$ output depends upon the status of the IOB pin. If IOB is grounded, it acts as master cascade enable to control cascaded 8259A; else it acts as peripheral data enable used in the multiple bus configurations. $\overline{INTA}$ pin is used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.

$\overline{IORC}$, $\overline{IOWC}$ are I/O read command and I/O write command signals respectively. These signals enable an IO interface to read or write the data from or to the addressed port. The $\overline{MRDC}$, $\overline{MWTC}$ are memory read command and memory write command signals respectively and may be used as memory read and write signals. All these command signals instruct the memory to accept or send data from or to the bus. For both of these write command signals, the advanced signals namely $\overline{AIOWC}$ and $\overline{AMWTC}$ are available. They also serve the same purpose, but are activated one clock cycle earlier than the $\overline{IOWC}$ and $\overline{MWTC}$ signals, respectively. The maximum mode system is shown in fig. 1.2

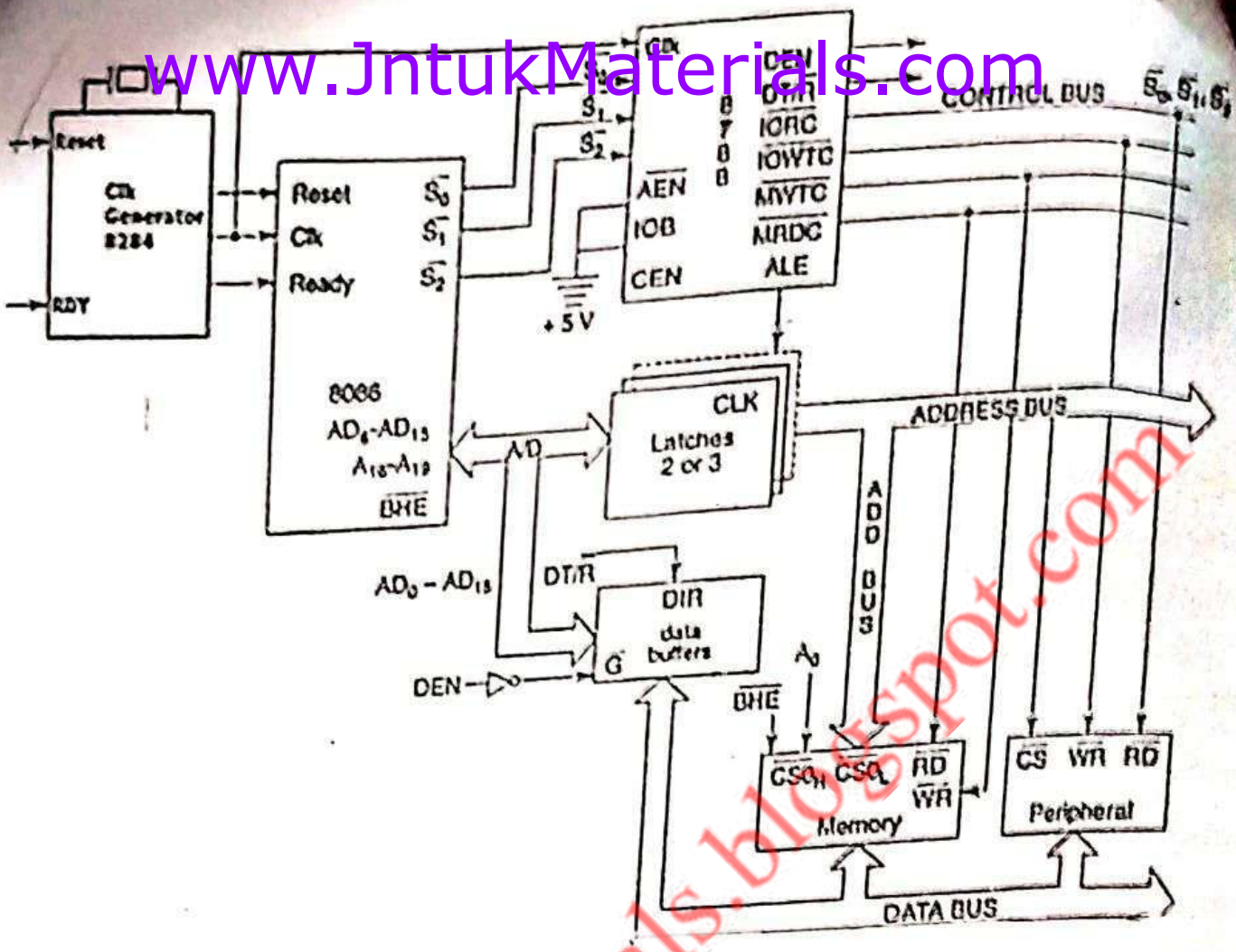


Fig 1.2: Maximum Mode 8086 System

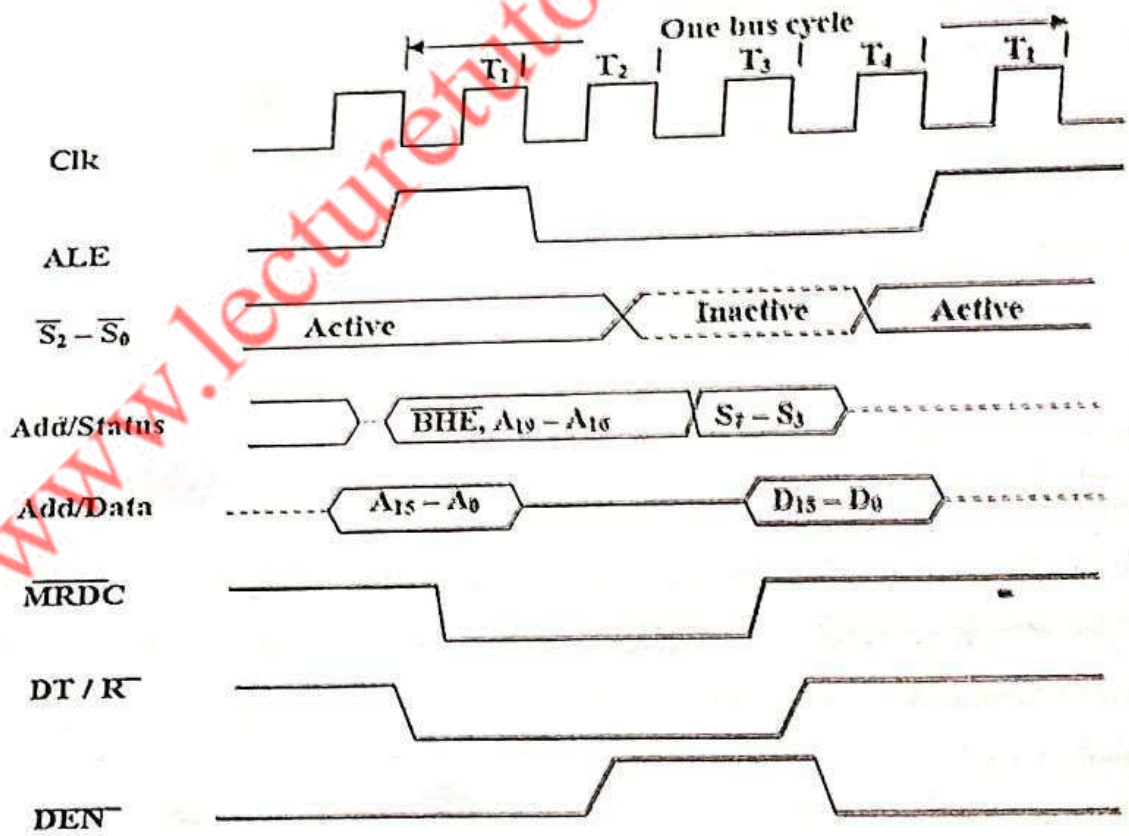


Fig. 1.2 (a): Memory Read Timing in Maximum Mode

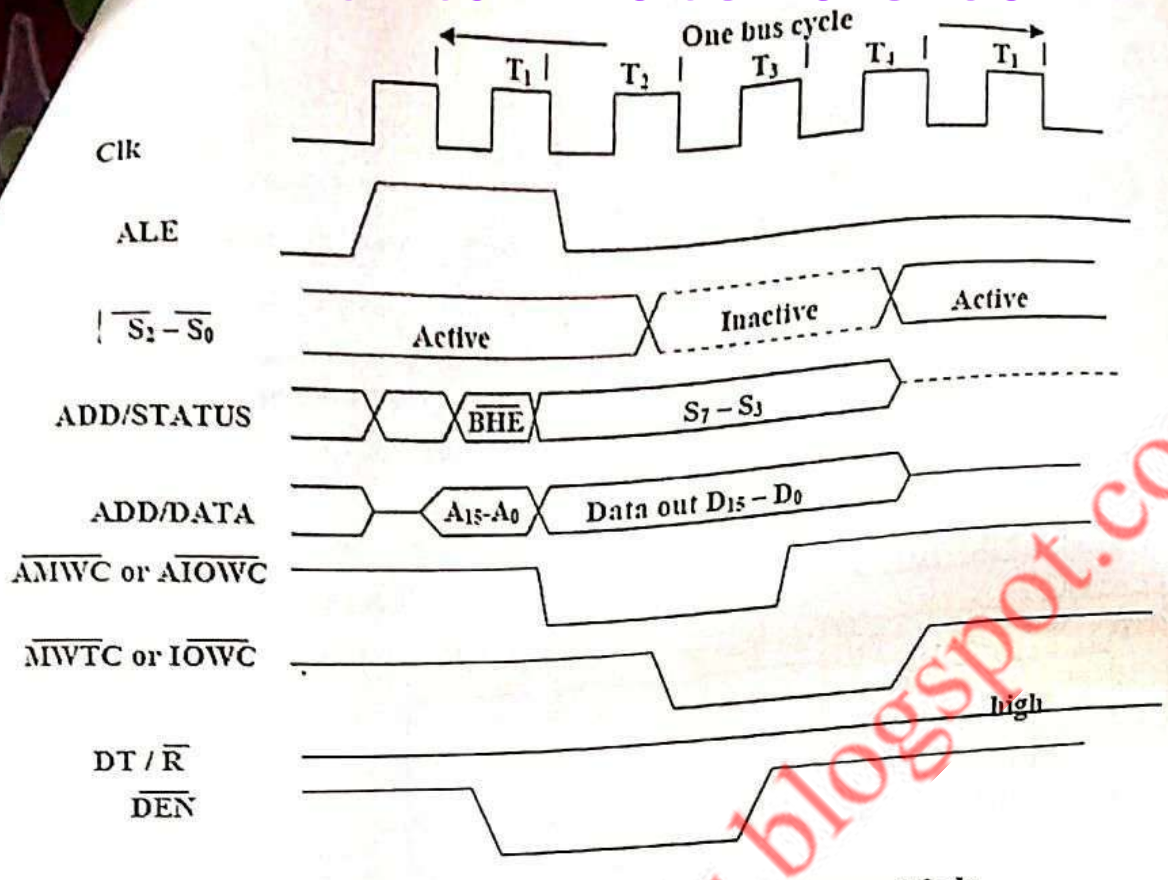


Fig. 1.2 (b): Memory Write Timing in Maximum Mode

The maximum mode system timing diagrams are also divided in two portions as read (input) and write (output) timing diagrams. The address/data and address/status timings are similar to the minimum mode. ALE is asserted in T1, just like minimum mode. The only difference lies in the status signals used and the available control and advanced command signals. The fig. 1.2 (a) shows the maximum mode timings for the read operation while the fig. 1.2 (b) shows the same for the write operation.

Timings for $\overline{RQ/GT}$ Signals: -

The request/ grant response sequence contains a series of three pulses as shown in the timing diagram. When a request is detected for valid HOLD request are satisfied, the processor issues a grant pulse over the $\overline{RQ/GT}$ pin immediately during the T4 (current) or T1 (next) state. When the requesting master receives this pulse, it accepts the control of the bus. The requesting master uses the bus till it requires. When it is ready to relinquish the bus, it sends a release pulse to the processor (host) using the $\overline{RQ/GT}$ pin.

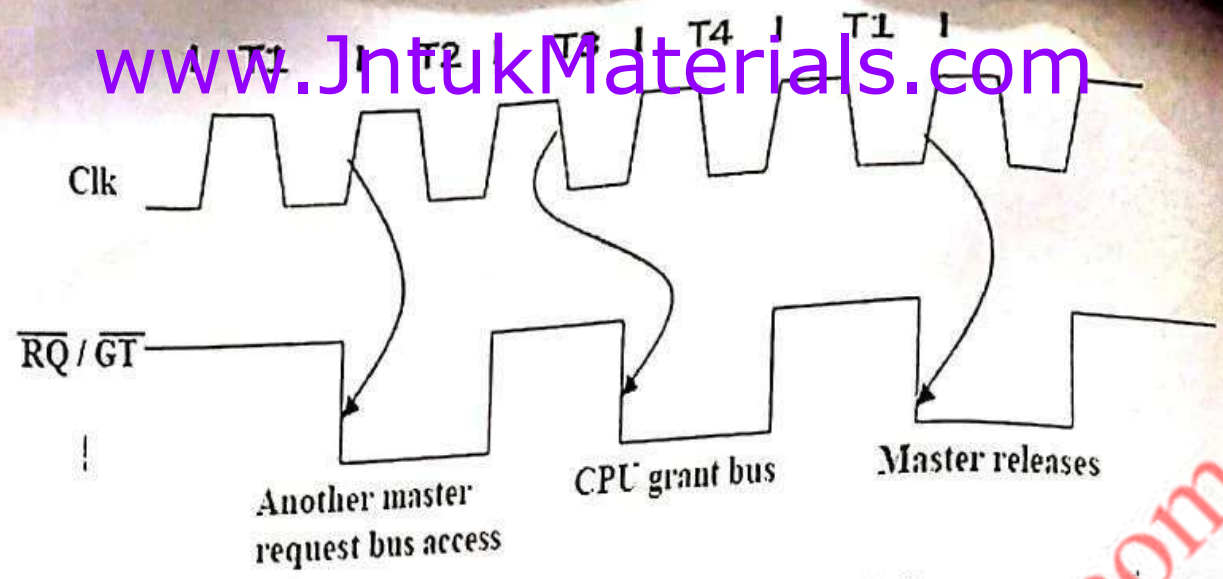
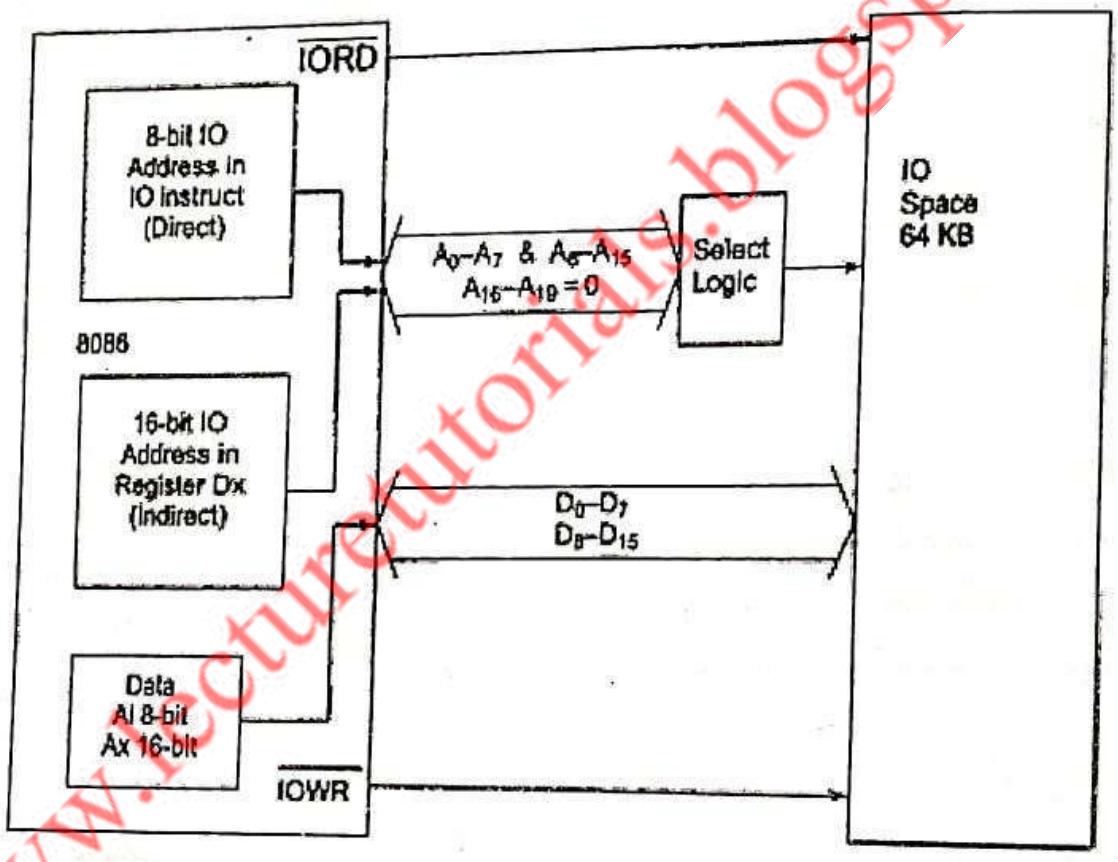


Fig1.2 (c): $\overline{RQ/GT}$ Timings in Maximum Mode

I/O Addressing Capability:



The 8086 can generate 16-bit of I/O address. Thus, it can address up to 64 Kbytes I/O locations or 32K word I/O locations. The 16-bit I/O address appears on A0 to A15 address lines; A16 to A19 lines are at logic 0 during the I/O operations. The 16-bit DX register is used as 16-bit I/O address pointer to address up to 64K devices in in-direct addressing mode. The I/O instructions with direct addressing mode can directly address one or two of the 256 I/O byte locations.

I/O ports are addressed in the same manner as memory locations. Even addressed bytes are transferred on the D7 – D0 bus lines and Odd addressed bytes on D15 – D8.

rupt and its Need

The microprocessors allow normal program execution to be interrupted in order to carry out a specific task/work. The processor can be interrupted in the following ways

i) by an external signal generated by a peripheral,

ii) by an internal signal generated by a special instruction in the program,

iii) by an internal signal generated due to an exceptional condition which occurs while executing an instruction. (For example, in 8086 processors, divide by zero is an exceptional condition which initiates type 0 interrupt and such an interrupt is also called execution).

In general, the process of interrupting the normal program execution to carry out a specific task/work is referred to as interrupt.

The interrupt is initiated by a signal generated by an external device or by a signal generated internal to the processor. When a microprocessor receives, an interrupt signal it stops executing current normal program, save the status (or content) of various registers (IP, CS and flag registers in case of 8086) in stack and then the processor executes a subroutine/procedure in order to perform the specific task/work requested by the interrupt. The subroutine/procedure that is executed in response to an interrupt is also called Interrupt Service Subroutine. (ISR). At the end of ISR, the stored status of registers in stack is restored to respective registers, and the processor resumes the normal program execution from the point (instruction) where it was interrupted.

The external interrupts are used to implement interrupt driven data transfer scheme. The interrupts generated by special instructions are called software interrupts and they are used to implement system services/calls (or monitor services/calls). The system/monitor services are procedures developed by system designer for various operations and stored in memory. The user can call these services through software interrupts. The interrupts generated by exceptional conditions are used to implement error conditions in the system.

Interrupt Driven Data Transfer Scheme

The interrupts are useful for efficient data transfer between processor and peripheral. When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal. Upon receiving an interrupt signal, the processor suspends the current program execution, save the status in stack and executes an ISR to perform the data transfer between the peripheral and processor. At the end of ISR the processor status is restored from stack and processor resume its normal program execution. This type of data transfer scheme is called interrupt driven data transfer scheme.

The data transfer between the processor and peripheral devices can be implemented either by polling technique or by interrupt method. In polling technique the processor has to periodically poll or check the status/readiness of the device and can perform data transfer only when the device is ready. In polling technique, the processor time is wasted, because the processor has to suspend its work and check the status of the device in predefined intervals.

Alternatively, if the device interrupts the processor to initiate a data transfer whenever it is ready then the processor time is effectively utilized because the processor need not suspend its work and check the status of the device in predefined intervals. For an example, consider the data transfer from a keyboard to the processor.

Normally a keyboard has to be checked by the processor once in every 10 milli seconds for a key press. Therefore, once in every 10 milli seconds the processor has to suspend its work and then check the keyboard for a valid key code. Alternatively, the keyboard can interrupt the processor, whenever a key is pressed and a valid key code is generated. In this way, the processor need not waste it's time to check the keyboard once in every 10 milli seconds.

Classification of Interrupts

In general, the interrupts can be classified in the following three ways:

1. Hardware and software interrupts
2. Vectored and Non-Vectored interrupt
3. Maskable and Non-Maskable interrupts

The interrupts initiated by external hardware by sending an appropriate signal to the interrupt pin of the processor is called hardware interrupt. The 8086 processor has two interrupt pins INTR and NMI. The interrupts initiated by applying appropriate signal to these pins are called hardware interrupts of 8086.

The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, if software interrupt instruction is encountered then the processor initiates an interrupt. The 8086 processor has 256 types of software interrupts. The software interrupt instruction is INT n, where n is the type number in the range 0 to 255.

When an interrupt signal is accepted by the processor, if the program control automatically branches to a specific address (called vector address) then the interrupt is called vectored interrupt. The automatic branching to vector address is predefined by the manufacturer of processors. (In these vector addresses the interrupt service subroutines (ISR) are stored). In non-vectored interrupts the interrupting device should supply the address of the ISR to be executed in response to the interrupt. All the 8086 interrupts are vectored interrupts. The vector address for an 8086 interrupt is obtained from a vector table implemented in the first 1kb memory space (00000h to 03FFFh).

The processor has the facility for accepting or rejecting hardware interrupts by programming the ...ssor to reject an interrupt is referred to as masking or disabling and programming the processor to ...cept an interrupt is referred to as unmasking or enabling. In 8086 the interrupt flag (IF) can be set to one ...o unmask or enable all hardware interrupts and IF is cleared to zero to mask or disable a hardware interrupts except NMI.

The interrupts whose request can be either accepted or rejected by the processor are called maskable interrupts. The interrupts whose request has to be definitely accepted (or cannot be rejected) by the processor are called non-maskable interrupts. Whenever a request is made by non-maskable interrupt, the processor has to definitely accept that request and service that interrupt by suspending its current program and executing an ISR. In 8086 processors, all the hardware interrupts initiated through INTR pin are maskable by clearing interrupt flag (IF). The interrupt initiated through NMI pin and all software interrupts are non-maskable.

Sources of Interrupts in 8086.

An interrupt in 8086 can come from one of the following three sources.

1. One source is from an external signal applied to NMI or INTR input pin of the processor. The interrupts initiated by applying appropriate signals to these input pins are called hardware interrupts.
2. A second source of an interrupt is execution of the interrupt instruction "INT n", where n is the type number. The interrupts initiated by "INT n" instructions are called software interrupts.
3. The third source of an interrupt is from some condition produced in the 8086 by the execution of an instruction. An example of this type of interrupt is divide by zero interrupt. Program execution will be automatically interrupted if you attempt to divide an operand by zero. Such conditional interrupts are also, known as exceptions

Interrupts Cycle of 8086:

The 8086 microprocessor has 256 types of interrupts. INTEL has assigned a type number to each interrupt. The type numbers are in the range of 0 to 255. The 8086 processor has dual facility of initiating these 256 interrupts. The interrupts can be initiated either by executing "INT n" instruction where n is the type number or the interrupt can be initiated by sending an appropriate signal to INTR input pin of the processor.

For the interrupts initiated by software instruction" INT n ", the type number is specified by the instruction itself. When the interrupt is initiated through INTR pin, then the processor runs an interrupt acknowledge cycle to get the type number. (i.e., the interrupting device should supply the type number through D0- D7 lines when the processor requests for the same through interrupt acknowledge cycle).

The kinds of interrupts and their designated types are summarized in figure by illustrating the layout of their pointers within the memory. Only the first five types have explicit definitions; the other

types may be used by interrupt instructions or external interrupts. From the figure, it is seen that the ty[illegible] associated with a division error interrupt is 0.

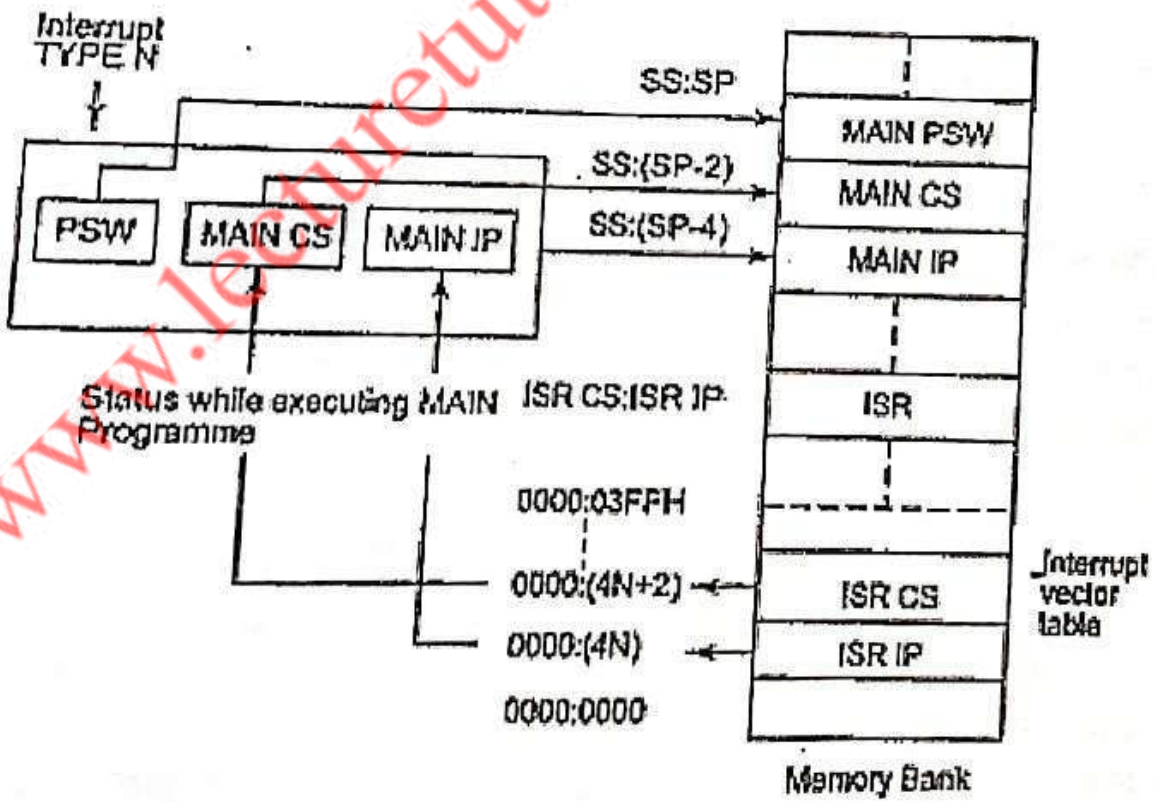
Therefore, if a division by 0 is attempted, the processor will push the current contents of the PSW, CS and IP into the stack, fill the IP and CS registers from the addresses 00000 to 00003, and continue executing at the address indicated by the new contents of IP and CS.

A division error interrupt occurs any time a DIV or IDIV instruction is executed with the quotient exceeding the range, regardless of the IF (Interrupt flag) and TF (Trap flag) status.

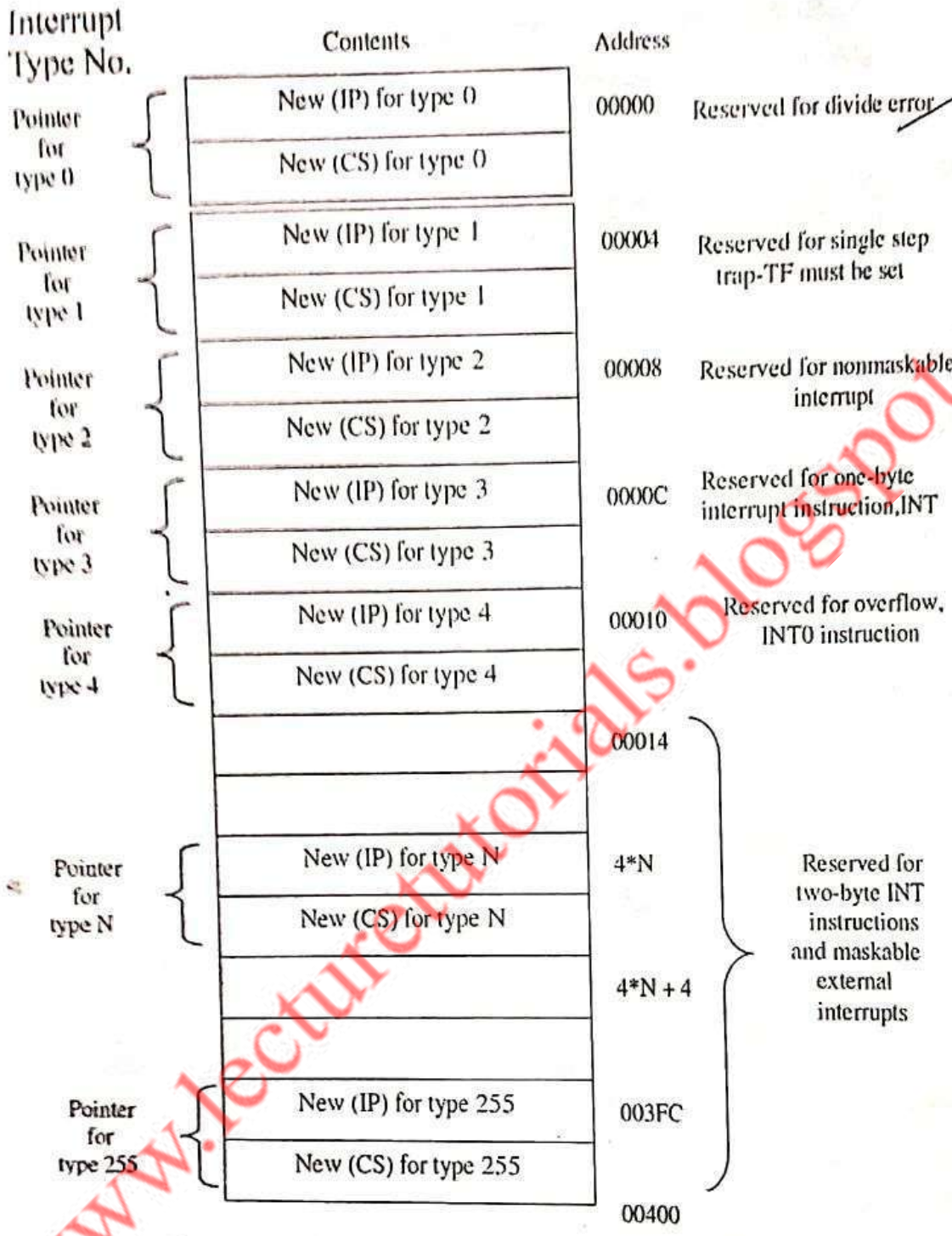
The type 1 interrupt is the single-step interrupt (Trap interrupt) and is the only interrupt controlled by the TF flag. If the TF flag is enabled, then an interrupt will occur at the end of the next instruction that will cause a branch to the location indicated by the contents of 00004H to 00007H.The single step interrupt is used primarily for debugging which gives the programmer a snapshot of his program after each instruction is executed.

The type 2 interrupt is the non-maskable external interrupt. It is the only external interrupt that can occur regardless of the IF flag setting. It is caused by a signal sent to the CPU through the non-maskable interrupt pin.

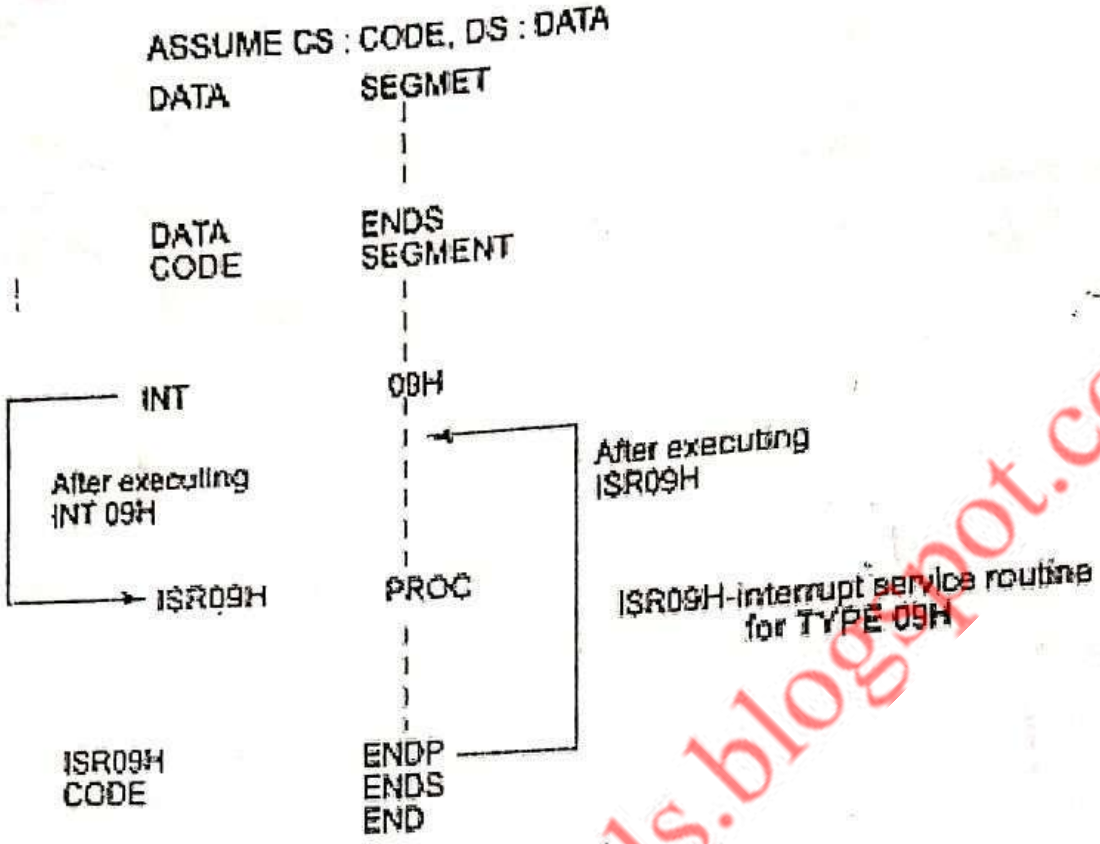
The remaining interrupt types correspond to interrupts instructions imbedded in the interrupt program or to external interrupts. The interrupt instructions are summarized below and their interrupts are not controlled by the IF flag.

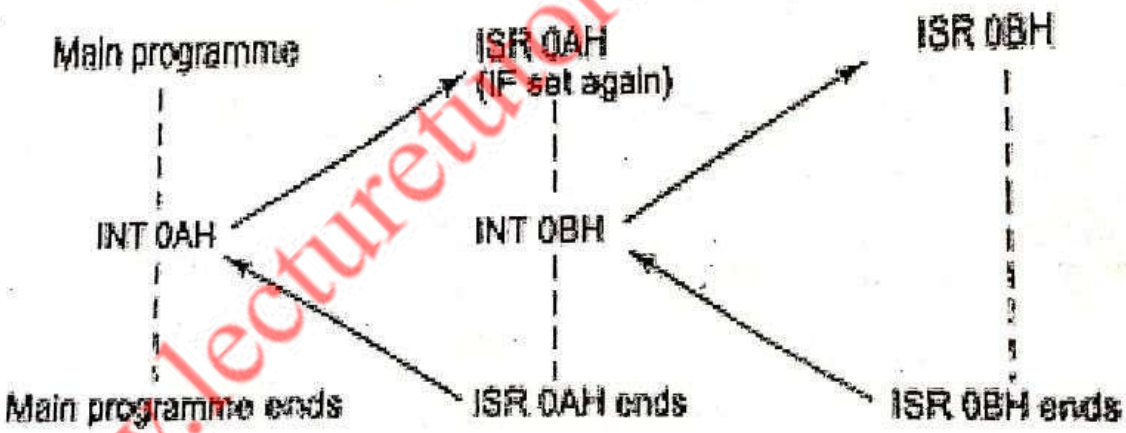


Interrupt Response Sequence



Structure of Interrupt vector table of 8086





Transfer of Control for Nested Interrupts