UNIT – II

OVERVIEW

- Introduction to 8086 microprocessors
- Architecture of 8086 processors
- Register Organization of 8086
- Memory Segmentation of 8086
- Pin Diagram of 8086
- Timing Diagrams for 8086
- Interrupts of 8086

UNIT-II

Features of 8086:

•It is a 16-bit μp.
•8086 has a 20 bit address bus can access up to 2^20 memory locations (1 MB).
•It can support up to 64K I/O ports.
•It provides 14, 16 -bit registers.
•It has multiplexed address and data bus AD0- AD15 and A16 – A19.
•It requires single phase clock with 33% duty cycle to provide internal timing.
•8086 is designed to operate in two modes, Minimum and Maximum.
•It can pre-fetches up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.
•It requires +5V power supply.
•A 40 pin dual in line package.

Architecture of 8086:

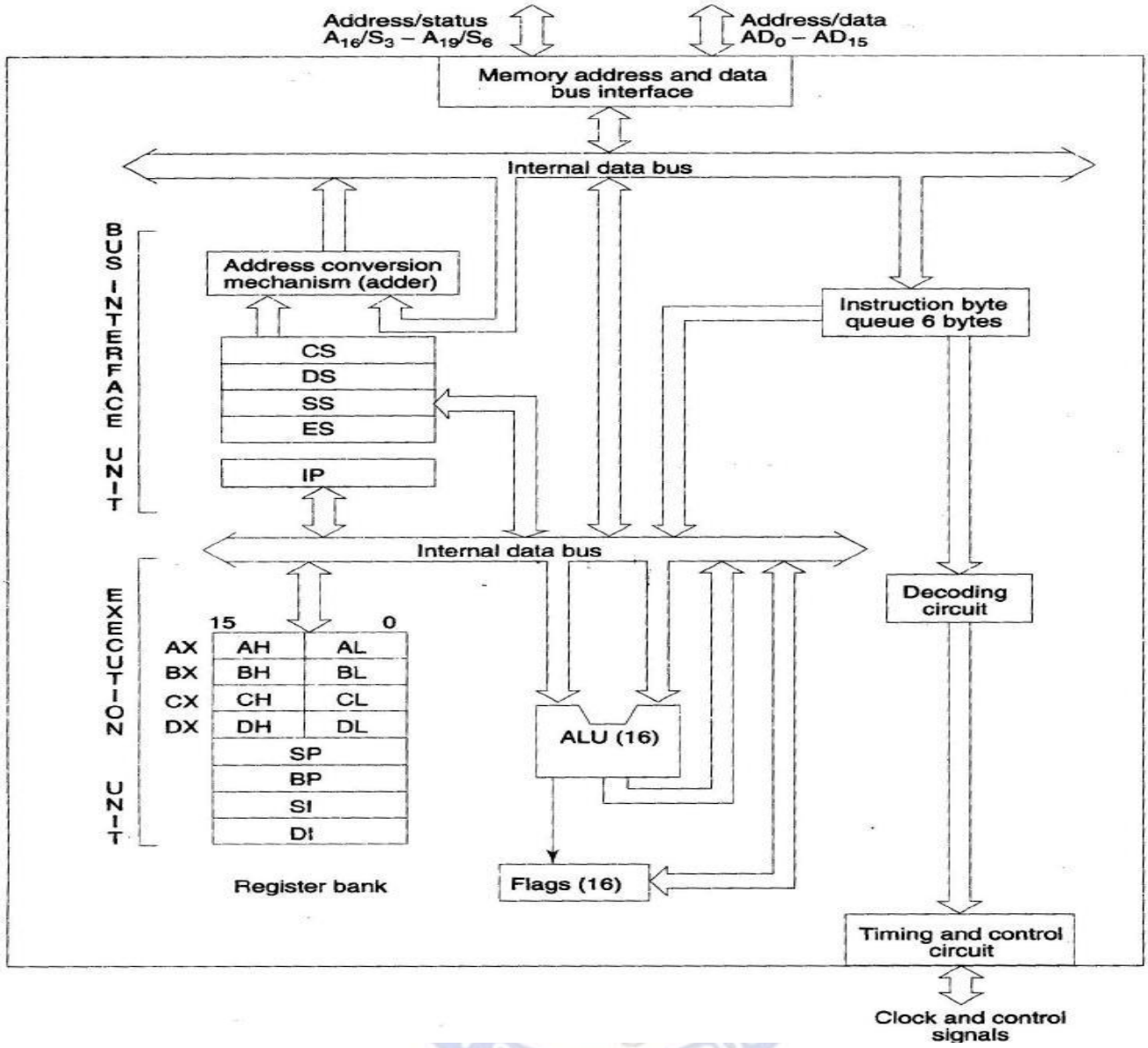
- 8086 has two blocks BIU and EU.
- The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands. The instruction bytes are transferred to the instruction queue.
- EU executes instructions from the instruction byte queue.
- 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.
- BIU contains Instruction queue, Segment registers, IP, address adder.
- EU contains control circuitry, Instruction decoder, ALU, Flag register.

Bus Interface Unit:

- It provides full 16 bit bidirectional data bus and 20 bit address bus.
- The BIU is responsible for performing all external bus operations.

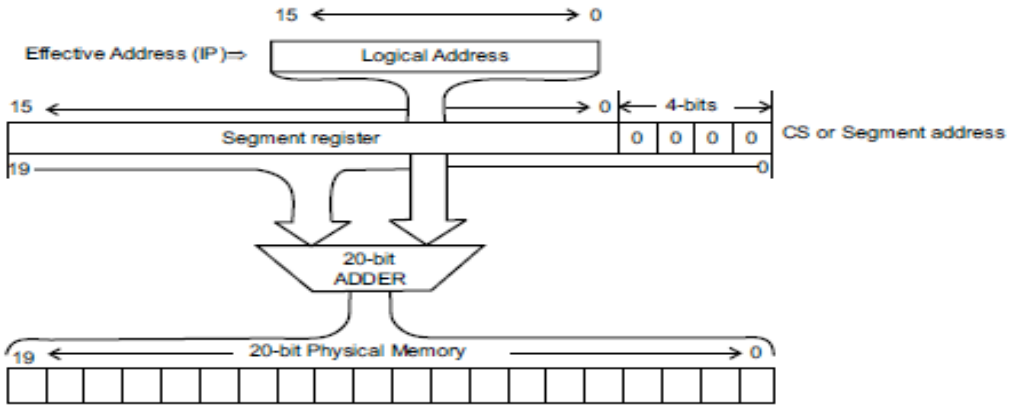
Specifically it has the following functions:

- Instructions fetch Instruction queuing, Operand fetch and storage, Address relocation and Bus control.
- The BIU uses a mechanism known as an instruction stream queue to implement pipeline architecture.
- This queue permits pre-fetch 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 pre-fetching the next sequential instruction.
- These pre-fetching instructions are held in its FIFO queue. With its 16 bit data bus, the BIU fetches two instruction 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 EU accesses the queue from the output end. It reads one instruction byte after the other from the output of the queue. If the queue is full and the EU is not requesting access to operand in memory.
- These intervals of no bus activity, which may occur between bus cycles, are known as **idle state**.



• If the bus is already in the process of fetching an instruction when the EU request it to read or write operands from memory or I/O, the BIU first completes the instruction fetch bus cycle before initiating the operand read / write cycle.

• 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.



Physical address generation

Thus, Physical Address = Segment Register content 16 D + Offset

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

Execution Unit:

- The EU extracts instructions from 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.

Register organization of 8086:

The 8086 has four groups of the user accessible internal registers. They are the instruction pointer, four data registers, four pointer and index register, four segment registers. The 8086 has a total of fourteen 16-bit registers including a 16 bit register called the ***status register***, with 9 of bits implemented for status and control flags.

There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses 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.

•**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.

•**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.

•**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 and string manipulation.

•**Base** register consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

•**Count** register consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low order byte of the word, and CH contains the high-order byte. Count register can be used in Loop, shift/rotate instructions and as a counter in string manipulation,.

•**Data** register consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low order byte of the word, and DH contains the high-order byte. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

•The following registers are both general and index registers:

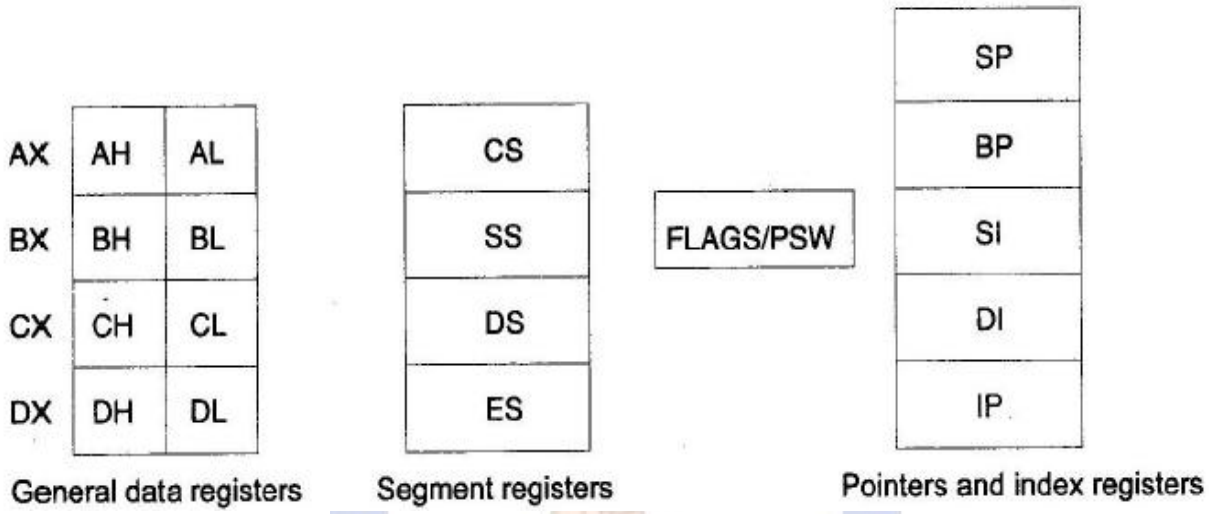
•**Stack Pointer** (SP) is a 16-bit register pointing to program stack.

•**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.

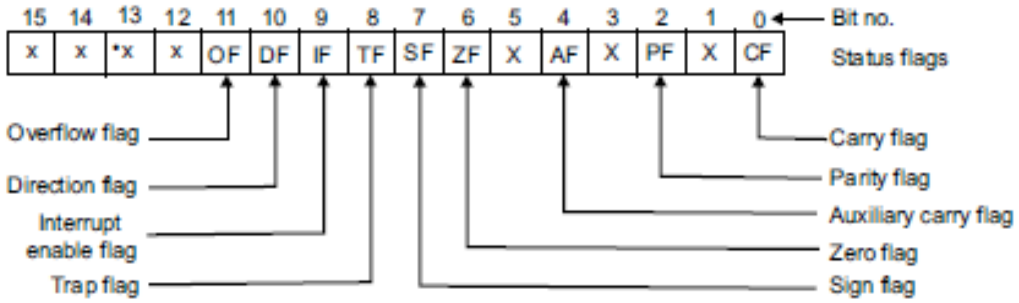
•**Source Index** (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.

•**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 addresses in string manipulation instructions.

Instruction Pointer (IP) register acts as a program counter for 8086. It points to the address of the next instruction to be executed. Its content is automatically incremented when the program execution of a program proceeds further. The contents of IP and CS register are used to compute the memory address of the instruction code to be fetched.



Flag register of 8086: It is a 16-bit register, also called flag register or Program Status Word (PSW). Seven bits remain unused while the rest nine are used to indicate the conditions of flags. The status flags of the register are shown below in Fig.



Status flags of Intel 8086

- Out of nine flags, six are condition flags and three are control flags. The control flags
- are TF (Trap), IF (Interrupt) and DF (Direction) flags, which can be set/reset by the
- programmer, while the **condition flags [OF (Overflow), SF (Sign), ZF (Zero), AF (Auxiliary**
- **Carry), PF (Parity) and CF (Carry)]** are set/reset depending on the results of some arithmetic or logical operations during program execution.
- **CF is set** if there is a carry out of the MSB position resulting from an addition operation or if a borrow is needed out of the MSB position during subtraction.
- **PF is set** if the lower 8-bits of the result of an operation contains an even number of 1's. AF is set if there is a carry out of bit 3 resulting from an addition operation or borrow required from bit 4 into bit 3 during subtraction operation.
- **ZF is set** if the result of an arithmetic or logical operation is zero.

- **SF is set** if the MSB of the result of an operation is 1. SF is used with unsigned numbers.
- OF is used only for signed arithmetic operation and is set if the result is too large to be fitted in the number of bits available to accommodate it.

  The three control flags of 8086 are TF, IF and DF. These three flags are programmable, i.e., can be set/reset by the programmer so as to control the operation of the processor.
- **When TF (trap flag) is set (=1),** the processor operates in **single stepping mode**—i.e., pausing after each instruction is executed. This mode is very useful during program development or program debugging.
- When an interrupt is recognized, TF flag is cleared. When the CPU returns to the main program from ISS (interrupt service subroutine), by execution of IRET in the last line of ISS, TF flag is restored to its value that it had before interruption.
- TF cannot be directly set or reset. So indirectly it is done by pushing the flag register on the stack, changing TF as desired and then popping the flag register from the stack.
- **When IF (interrupt flag) is set,** the maskable interrupt INTR is enabled otherwise disabled (i.e., when IF = 0).
- **IF can be set by executing STI instruction and cleared by CLI instruction.** Like TF flag, when an interrupt is recognized, IF flag is cleared, so that INTR is disabled. In the last line of ISS when IRET is encountered, IF is restored to its original value. When 8086 is reset, IF is cleared, i.e., resetted.
- DF (direction flag) is used in string (also known as block move) operations. **It can be set by STD instruction and cleared by CLD.** If DF is set to 1 and MOVS instruction is executed, the contents of the index registers DI and SI are automatically decremented to access the string from the highest memory location down to the lowest memory location.

PIN DIAGRAM OF 8086

The **8086** is internally a **16-bit MPU** and **externally** it has a **16-bit data bus**. It has the ability to address up to **1 MB** of memory via its **20-bit address bus**. In addition, it can address up to **64K of byte-wide input/output ports**.

• It is manufactured using **high-performance metal-oxide semiconductor (HMOS) technology**, and the circuitry on its chip is equivalent to approximately **29,000 transistors**.

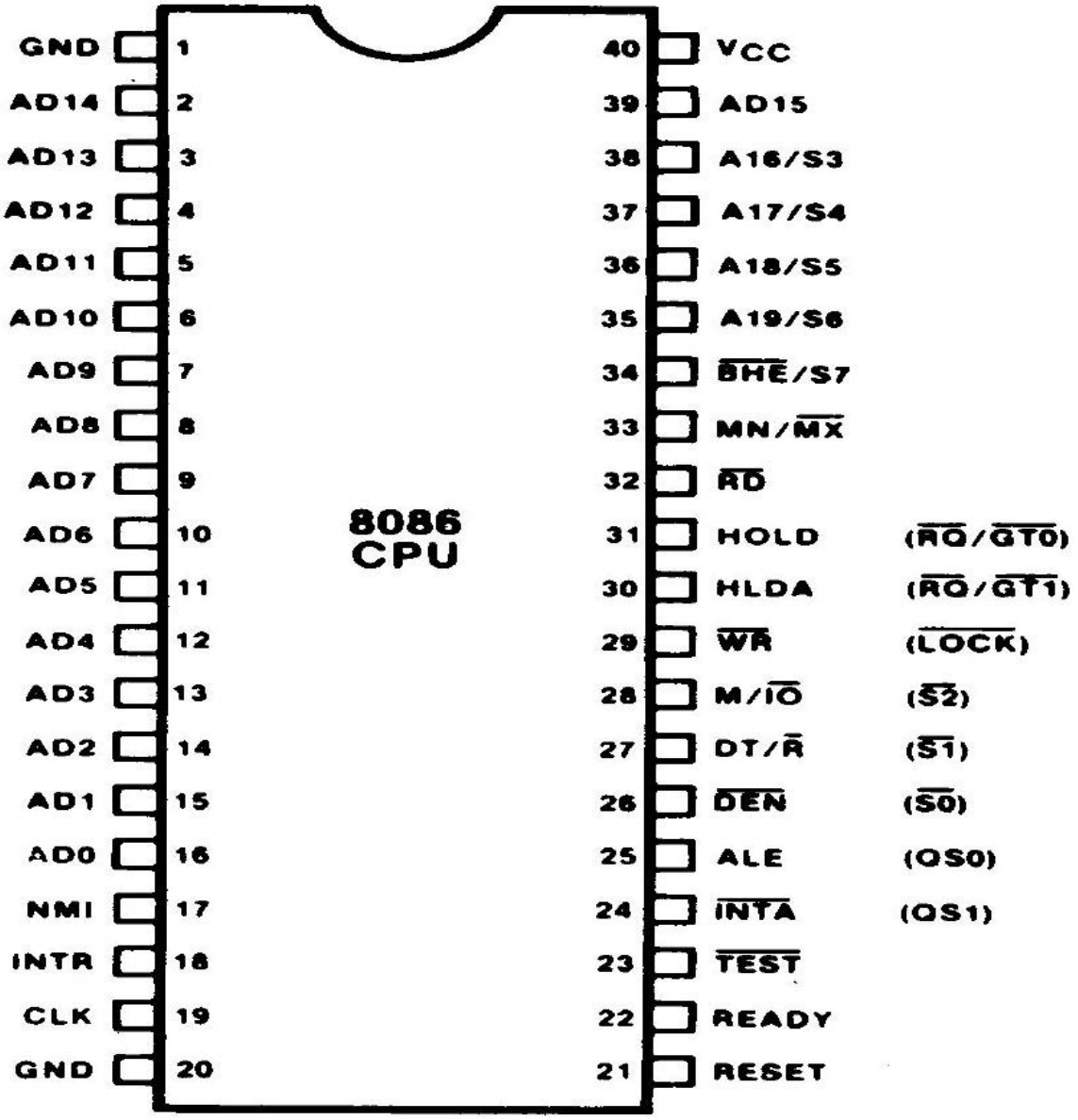
• The 8086 is housed in a **40-pin dual in-line package**. The signals pinned out to each lead are shown in figure.

The **address bus lines** A0 through A15 and **data bus lines** D0 through D15 are **multiplexed**. For this reason, these leads are labeled AD0 through AD15. By *multiplexed* we mean that the same physical pin carries an address bit at one time and the data bits at another time.

• The **8086** can be configured to work in either of **two modes**:

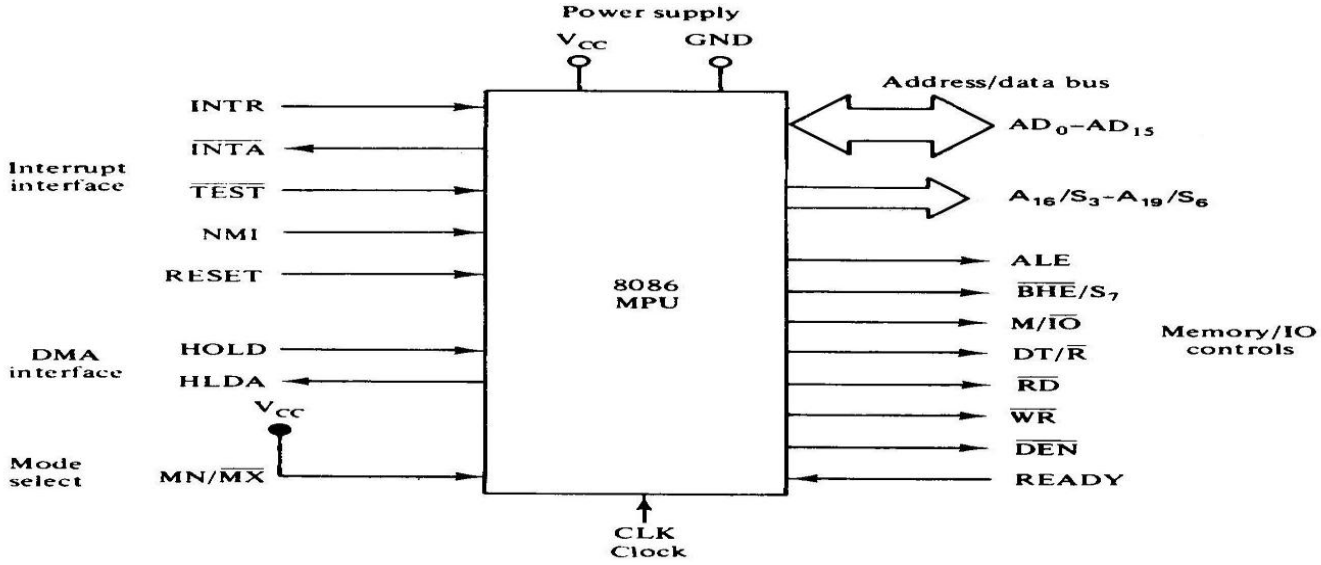
• The **minimum mode** is selected by applying **logic 1** to the **MN/MX** input lead. It is typically used for smaller **single microprocessor** systems.

• The **maximum mode** is selected by applying **logic 0** to the **MN/MX** input lead. It is typically used for larger **multiple microprocessor** systems.

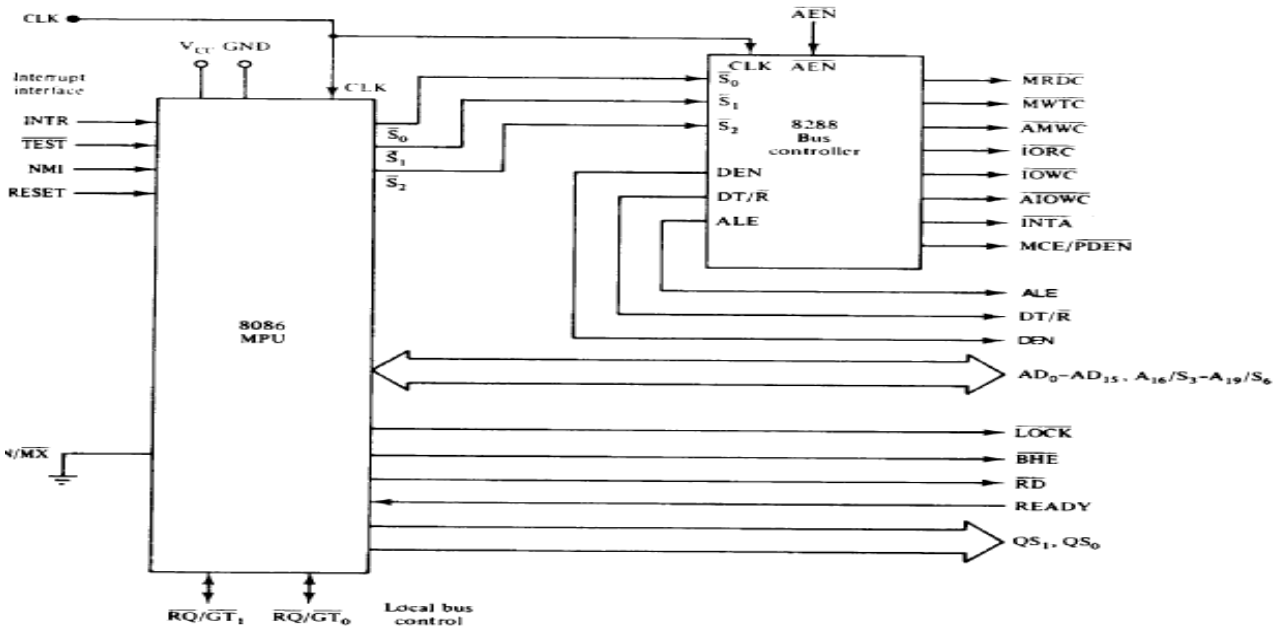


• Depending on the **mode** of operation selected, the **assignments** for a number of the **pins** on the microprocessor package are **changed**. The **pin functions** specified in **parentheses** pertain to the **maximum-mode**.

• In minimum mode, the **8086** itself **provides** all the **control signals** needed to implement the memory and I/O interfaces. In **maximum-mode**, a separate chip (the **8288 Bus Controller**) is used to help in sending control signals over the shared bus shown in figure.



MINIMUM MODE OF 8086



MAXIMUM MODE OF 8086

• **Address/Data Bus:** The address bus is **20 bits long** and consists of signal lines **A0** (LSB) through **A19** (MSB). However, only address lines **A0** through **A15** are used when accessing **I/O**.

• The **data bus** lines are **multiplexed** with address lines. For this reason, they are denoted as **AD0** through **AD15**. Data line **D0** is the LSB.

• **Status Signals:** The four most significant address lines **A16** through **A19** of the 8086 are multiplexed with **status signals S3** through **S6**. These status bits are output on the bus at the same time that data are transferred over the other bus lines.

The status of the Interrupt Enable Flag (IF) bit (displayed on S5) is updated at the beginning of each clock cycle.

S4, S3: together indicates which segment register is presently being used for memory access. These lines float at tristate off during the local bus hold acknowledge.

S6: It is always low.

S4	S3	Indications
0	0	Alternate data
0	1	Stack
1	0	Code or none
1	1	Data

BHE/S7-Bus High Enable/Status: The bus high enable signal is used to indicate the transfer of data over the higher order (D15-D8) data bus. It goes low for the data transfers over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. BHE is low during T1 for read, write and interrupt acknowledge cycles, when- ever a byte is to be transferred on the higher byte of the data bus.

BHE	A0	Indication
0	0	Whole word i.e AD15 – AD8
0	1	Upper byte from or to i.e AD15-AD8
1	0	Lower byte from or to even address i.e AD7-AD0
1	1	None

TEST: This input is examined by a 'WAIT' instruction. If the TEST input goes low, execution will continue, else, the processor remains in an idle state. The input is synchronized internally during each clock cycle on leading edge of clock.

RESET: This input causes the processor to terminate the current activity and start execution from FFFF0H. The signal is active high and must be active for at least four clock cycles. It restarts execution when the RESET returns low. RESET is also internally synchronized.

VCC: +5V power supply for the operation of the internal circuit. GND ground for the internal circuit.

• Control Signals:

• When ***Address latch enable*** (**ALE**) is **logic 1** it signals that a **valid address** is on the bus. This address can be latched in external circuitry on the **1-to-0 edge** of the pulse at ALE.

• **M/IO** (*memory*/IO) tells external circuitry whether a memory or I/O transfer is taking place over the bus. **Logic 1** signals a **memory operation** and **logic 0** signals an **I/O operation**.

• **DT/R** (*data transmit/receive*) signals the **direction of data transfer** over the bus. **Logic 1** indicates that the bus is in the **transmit mode** (i.e., data are either written into memory or to an I/O device). **Logic 0** signals that the bus is in the **receive mode** (i.e., reading data from memory or from an input port).

• The *bank high enable* (**BHE**) signal is used as a **memory enable signal** for the **most significant byte** half of the data bus, **D8** through **D15**.

• **WR** (*write*) is switched to **logic 0** to signal external devices that **valid output data** are on the bus.

• **RD** (*read*) indicates that the MPU is performing a **read of data** off the bus. During read operations, one other control signal, **DEN** (*data enable*), is also supplied. It enables external devices to supply data to the microprocessor.

• The **READY** signal can be used to **insert wait states** into the bus cycle so that it is extended by a number of clock periods. This signal is supplied by a **slow** memory **or I/O subsystem** to signal the MPU when it is ready to permit the data transfer to be completed.

• Interrupt Signals:

• *Interrupt request* (**INTR**) is an **input** to the 8086 that can be used by an **external device** to **signal** that it needs to be **serviced**. **Logic 1** at INTR represents an active interrupt request.

• When the MPU **recognizes an interrupt request**, it indicates this fact to external circuits with logic 0 at the *interrupt acknowledge* (**INTA**) output.

• On the **0-to-1 transition** of *non maskable interrupt* (**NMI**), control is passed to a non maskable **interrupt service routine** at completion of execution of the current instruction. NMI is the interrupt request with highest priority and **cannot be masked by software**.

• The **RESET** input is used to provide a **hardware reset** for the MPU. Switching RESET to **logic 0** initializes the internal registers of the MPU and initiates a reset service routine.

• DMA Interface Signals:

• When an **external device** wants to **take control** of the **system bus**, it signals this fact to the MPU by switching HOLD to the **logic level 1**.

• When in the hold state, lines **AD0** through **AD15**, **A16/S3** through **A19/S6**, **BHE**, **M/IO**, **DT/R**, **WR, RD**, **DEN** and **INTR** are all put in the **high-Z** state. The MPU signals external devices that it is in this state by switching **HLDA** to **1**.

SYSTEM CLOCK:

• To **synchronize** the internal and external operations of the microprocessor a *clock* (**CLK**) **input signal** is used. The CLK can be generated by the **8284 clock generator IC**.

• The **8086** is manufactured in three speeds: **5 MHz**, **8 MHz** and **10 MHz**.

MAXIMUM MODE SIGNALS:

S2, S1, S0 (Status lines): These are the status lines which reflect the type of operation, being carried out by the processor. These lines active during T4 of the previous cycle & remain active during T1 & T2 of the current bus cycle.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Indication
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1	0	0	Code Access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive

LOCK:

· This output pin indicates that other system bus masters will be prevented from gaining the system bus, while the LOCK=0.

· The LOCK signal is activated by the LOCK prefix instruction and remains active until the completion of the next instruction.

· This floats to tri-state off during 'hold acknowledge'.

QS1, QS0 (Queue status):

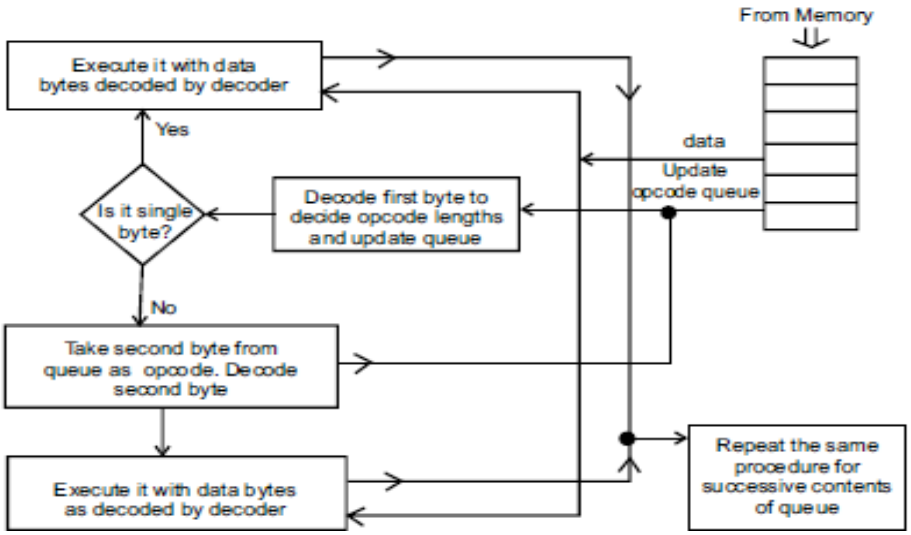
· These lines give information about the status of the code-prefetch queue.

· These are active during the CLK cycle after which the queue operation is performed.

· The 8086 architecture has a 6-byte instruction pre-fetch queue.

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

After decoding the first byte, the decoding circuit decides whether the instruction is of single opcode byte or double opcode byte. If it is **single opcode byte**, the next bytes are treated as data byte depending upon the decoded instruction length; otherwise, the next byte in the queue is treated as the **second byte** of the instruction opcode. 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.



RQ/GT0, RQ/GT1 (Request/Grant):

These pins are used by other local bus masters, in maximum mode, to force the processor to release the local bus at the end of the processor's current bus cycle. Each of the pins is bidirectional with RQ_0/GT_0 having higher priority than RQ_1/GT_1.

Minimum Mode 8086 System

•In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/MX pin to logic 1.

•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.

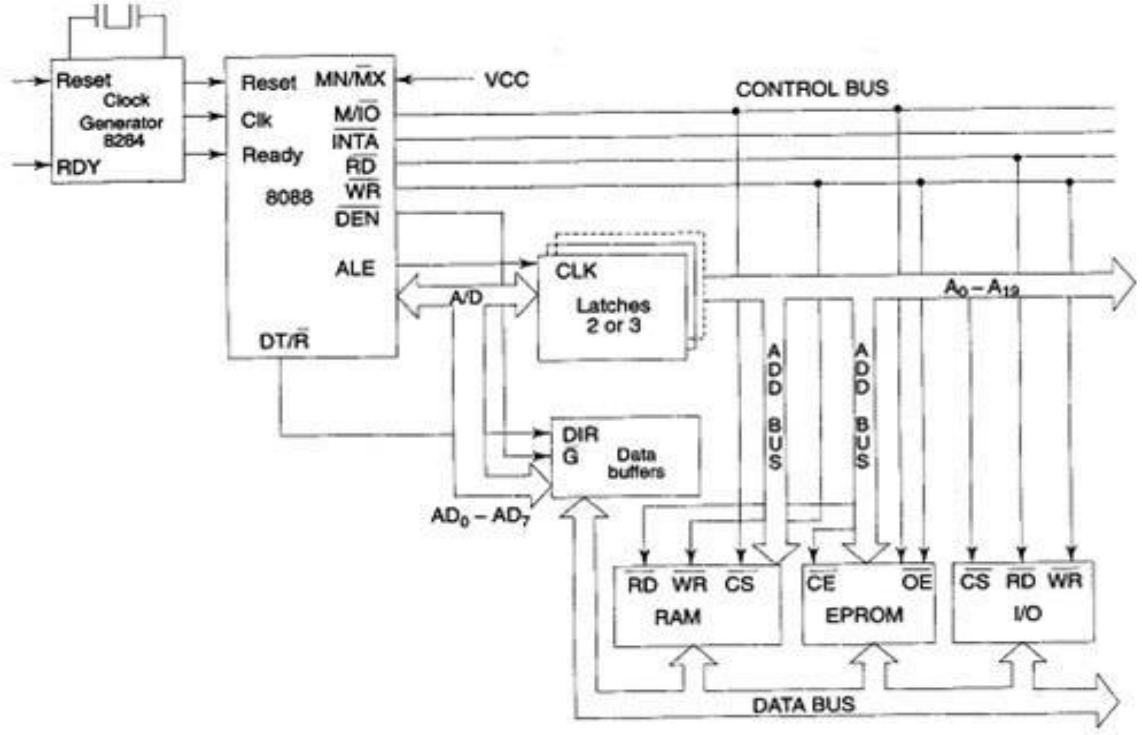
•The remaining components in the system are latches, transreceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.

•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.

•Transreceivers 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 signals.

•They are controlled by two signals namely, DEN and DT/R.

•The DEN signal indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage.



MINIMUM MODE SYSTEM

•Usually, EPROMs are used for monitor storage, while RAM for users program storage. A system may contain I/O devices.

•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.

•The read cycle begins in T1 with the assertion of address latch enable (ALE) signal and also M / IO signal. During the negative going edge of this signal, the valid address is latched on the local bus.

•The BHE and A0 signals address low, high or both bytes. From T1 to T4, the M/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 tristated. The read (RD) control signal is also activated in T2.

•The read (RD) signal causes the address device to enable its data bus drivers. After 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 tristate its bus drivers.

•A write cycle also begins with the assertion of ALE and the emission of the address. The M/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.

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

•The BHE and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or write.

Maximum Mode 8086 System

•In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.

In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information.

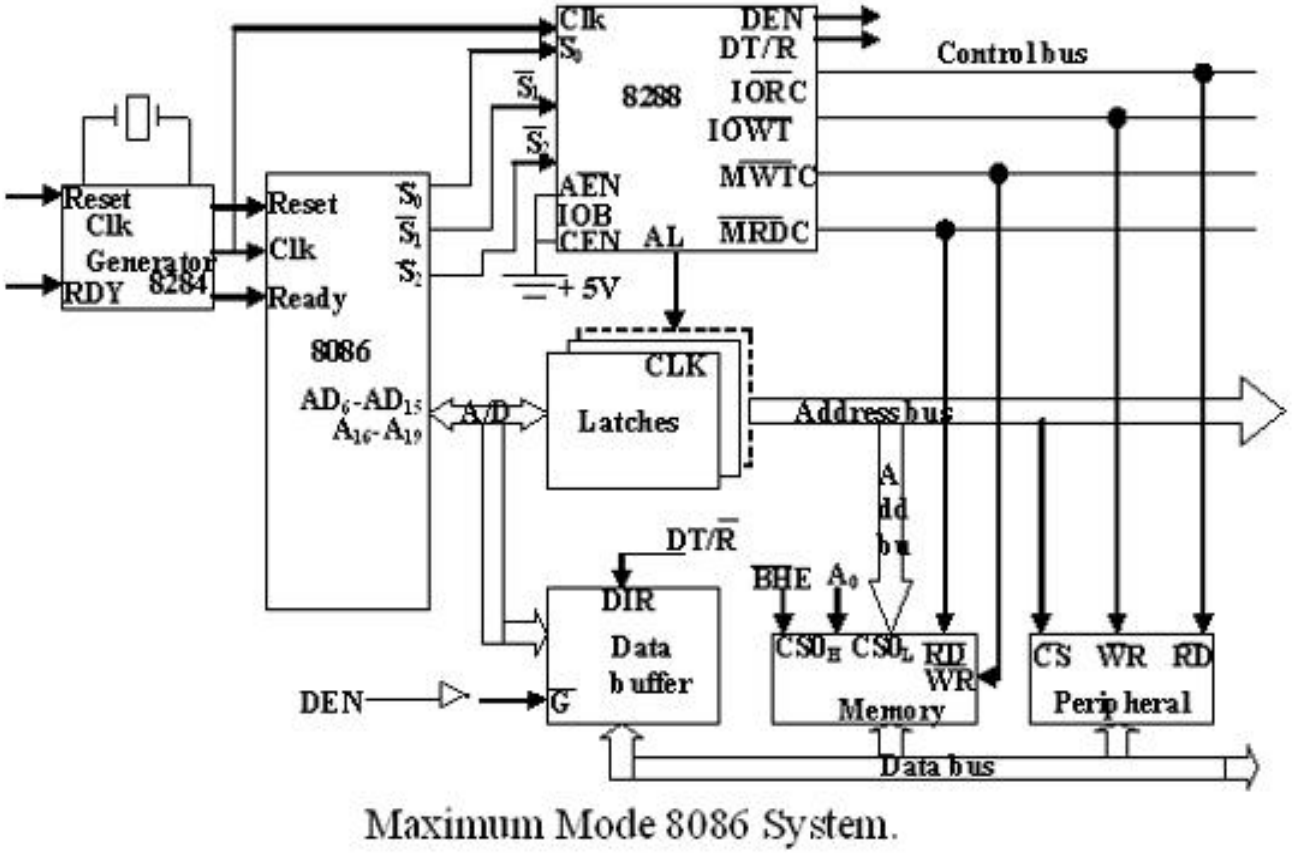
•In the maximum mode, there may be more than one microprocessor in the system configuration.

•The components in the system are same as in the minimum mode system.

•The basic function of the bus controller chip IC8288, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE etc. using the information by the processor on the status lines.

•The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.

•It derives the outputs ALE, DEN, DT/R, MRDC, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.



Maximum Mode 8086 System.

•AEN and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/PDEN output depends upon the status of the IOB pin.

•If IOB is grounded, it acts as master cascade enable to control cascade 8259A, else it acts as peripheral data enable used in the multiple bus configurations.

•INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.

•IORC, 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 address port.

•The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.

•All these command signals instructs the memory to accept or send data from or to the bus.

•For both of these write command signals, the advanced signals namely AIOWC and AMWTC are available.

•Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.

•R0, S1, S2 are set at the beginning of bus cycle.8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.

•In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4. For an output, the AMWC or

AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.

•The status bit S0 to S2 remains active until T3 and become passive during T3 and T4**.**

•If reader input is not activated before T3, wait state will be inserted between T3 and T4.

TIMING DIAGRAMS FOR 8086 IN MINIMUM MODE

BUS CYCLE AND TIME STATES

• A **bus cycle or machine cycle** defines the sequence of events when the MPU communicates with an external device, which starts with an address being output on the system bus followed by a read or write data transfer.

• Types of bus cycles:

Memory Read Bus Cycle

Memory Write Bus Cycle

Input/output Read Bus Cycle

Input/output Write Bus Cycle

One cycle of clock is called a state or t-state. The bus cycle of the 8086 microprocessor consists of at least four clock periods. These four time states are called T1, T2, T3 and T4. This group of states is called a **MACHINE CYCLE.**

The total time required to fetch and execute an instruction is called an **instruction cycle**. An instruction cycle consists of one or more machine cycle.

The following figure **shows** a **memory read cycle** of the **8086**:

• During **period T1**,

- The 8086 outputs the **20-bit address** of the memory location to be accessed on its multiplexed **address/data bus**. **BHE** is also output along with the address during T1.
- At the same time a pulse is also produced at **ALE**. The **trailing edge** or the **high level** of this pulse is used to **latch** the address in external circuitry.
- Signal **M/IO** is set to **logic 1** and signal **DT/R** is set to the **0 logic level** and both are maintained throughout all four periods of the bus cycle.

• Beginning with **period T2**,

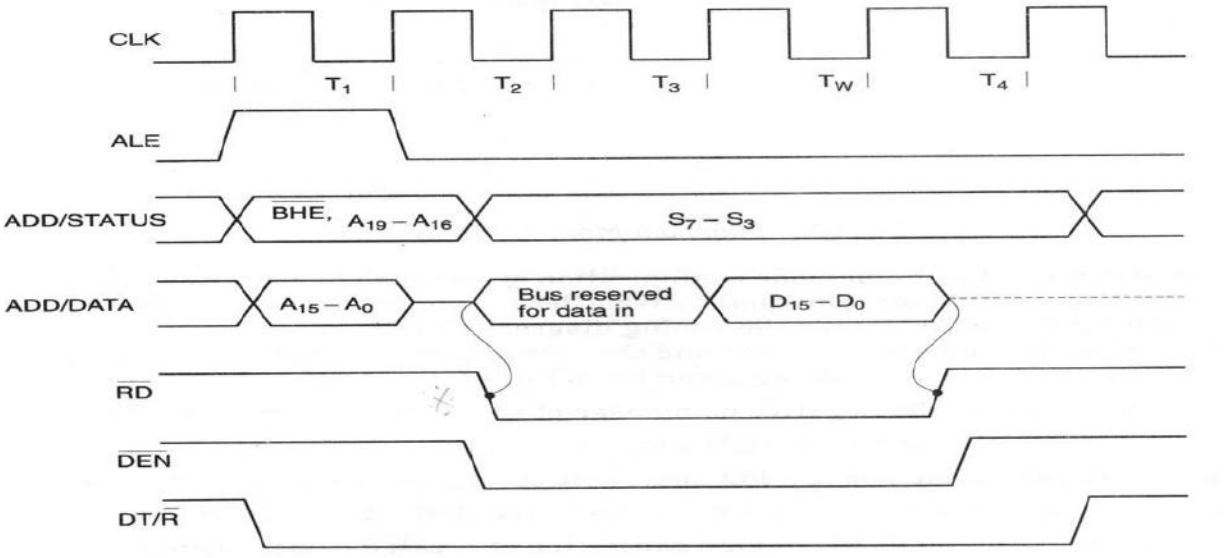
- Status bits **S3** through **S6** are output on the upper four address bus lines. This status information is maintained through periods **T3** and **T4**.
- On the other hand, address/data bus lines **AD0 through AD7** are put in the **high-Z state** during **T2**.
- Late in period **T2**, **RD** is switched to **logic 0**. This indicates to the memory subsystem that a read cycle is in progress. **DEN** is switched to **logic 0** to enable external circuitry to allow the data to move from memory onto the microprocessor's data bus.

• During **period T3**,

- The memory must provide **valid data** during **T3** and maintain it until after the processor terminates the read operation. The data read by the 8086 microprocessor can be carried over all **16 data bus** lines.

• During **T4**,

- The 8086 switches **RD** to the inactive **1 logic level** to terminate the read operation. **DEN** returns to its inactive logic level late during **T4** to disable the external circuitry.



MEMORY READ CYCLE FOR 8086 IN MINIMUM MODE

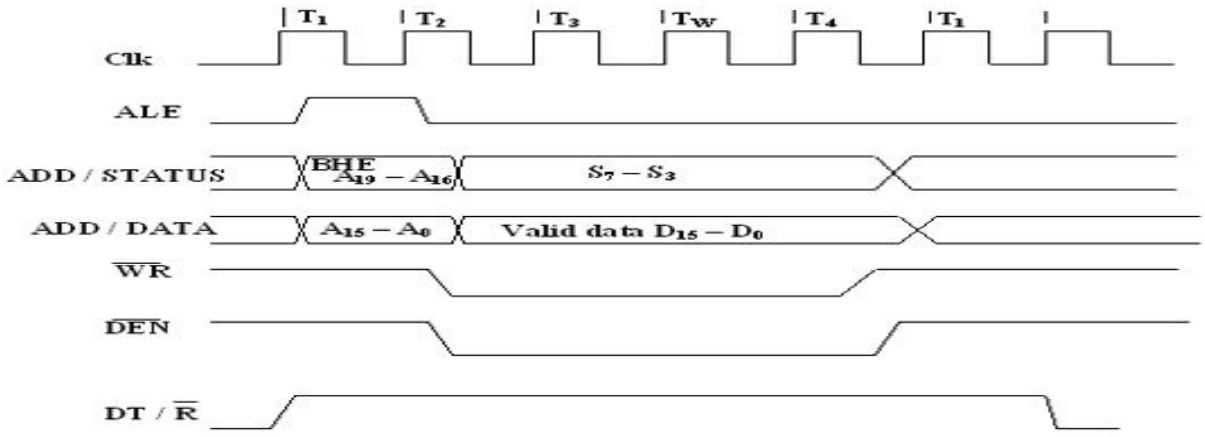
The following figure shows a **memory write cycle** of the 8086:

• During **period T1**,

- The **address** along with **BHE** is output and latched with the **ALE** pulse.
- **M/IO** is set to **logic 1** to indicate a memory cycle.
- However, this time **DT/R** is switched to **logic 1**. This signals external circuits that the 8086 is going to **transmit data** over the bus.

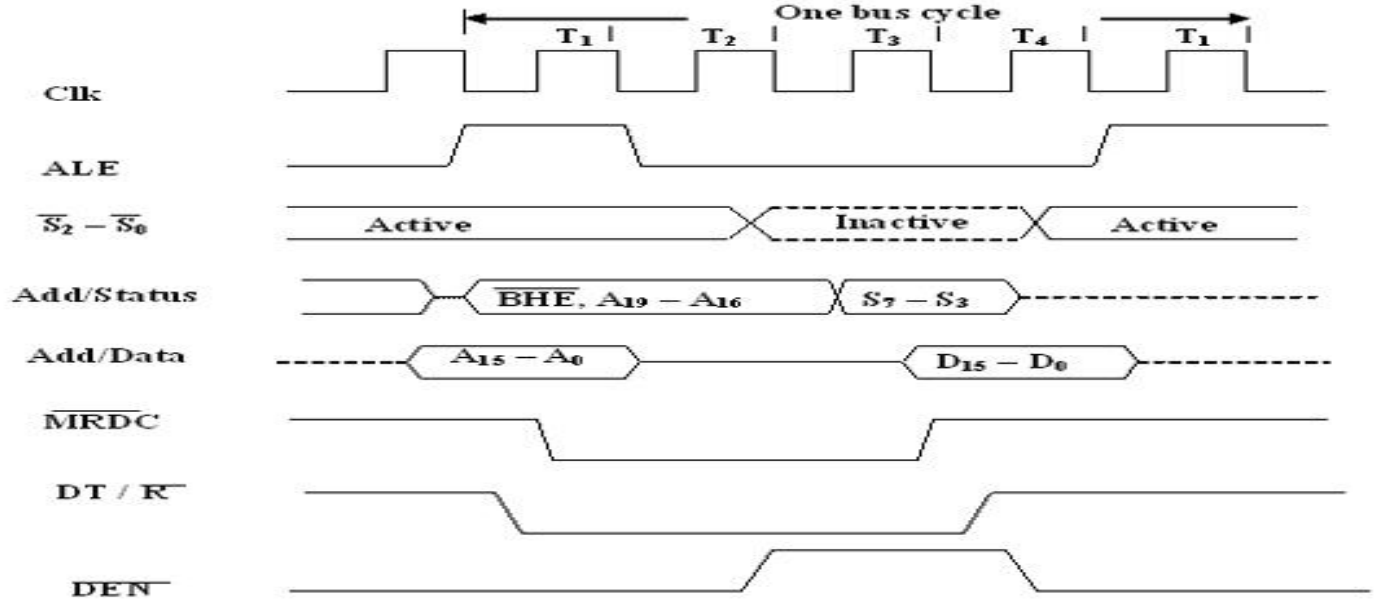
• Beginning with **period T2**,

- **WR** is switched to **logic 0** telling the memory subsystem that a write operation is to follow.
- The 8086 puts the **data** on the bus late in **T2** and maintains the data valid through **T4**. Data will be carried over all **16 data bus lines**.
- **DEN** enables the external circuitry to provide a path for data from the processor to the memory.

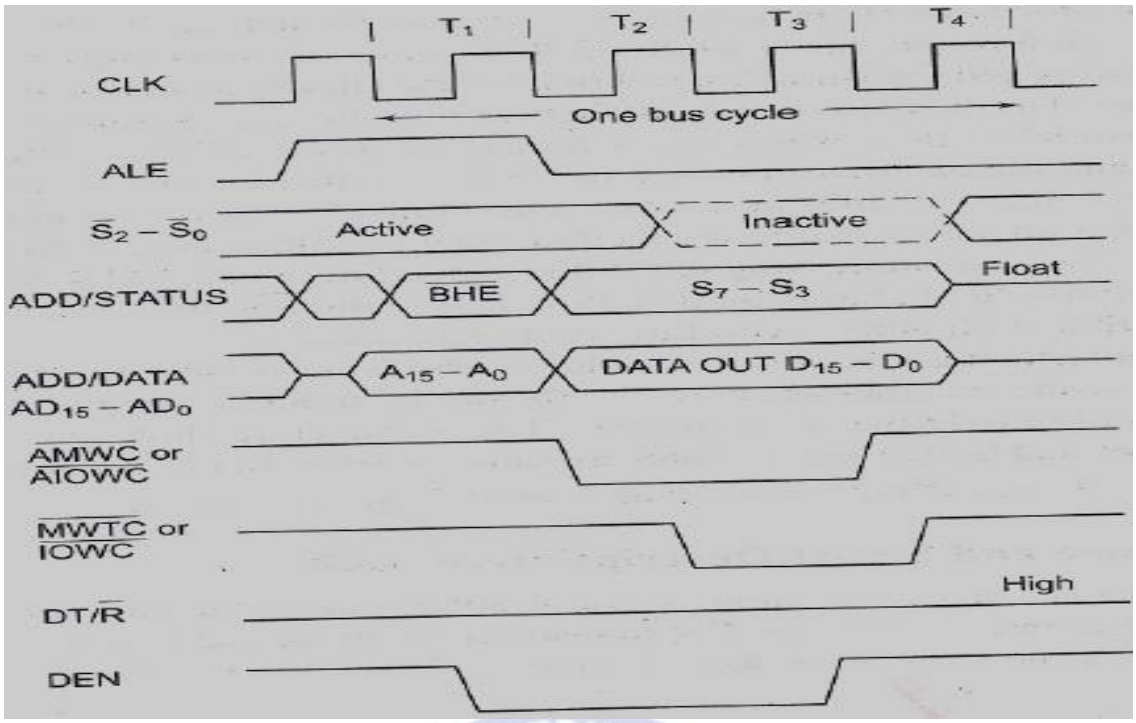


Write Cycle Timing Diagram for Minimum Mode

MAXIMUM MODE TIMING DIGRAMS



Memory Read Timing in Maximum Mode



WRITE CYCLE TIMING DIAGRAM FOR 8086