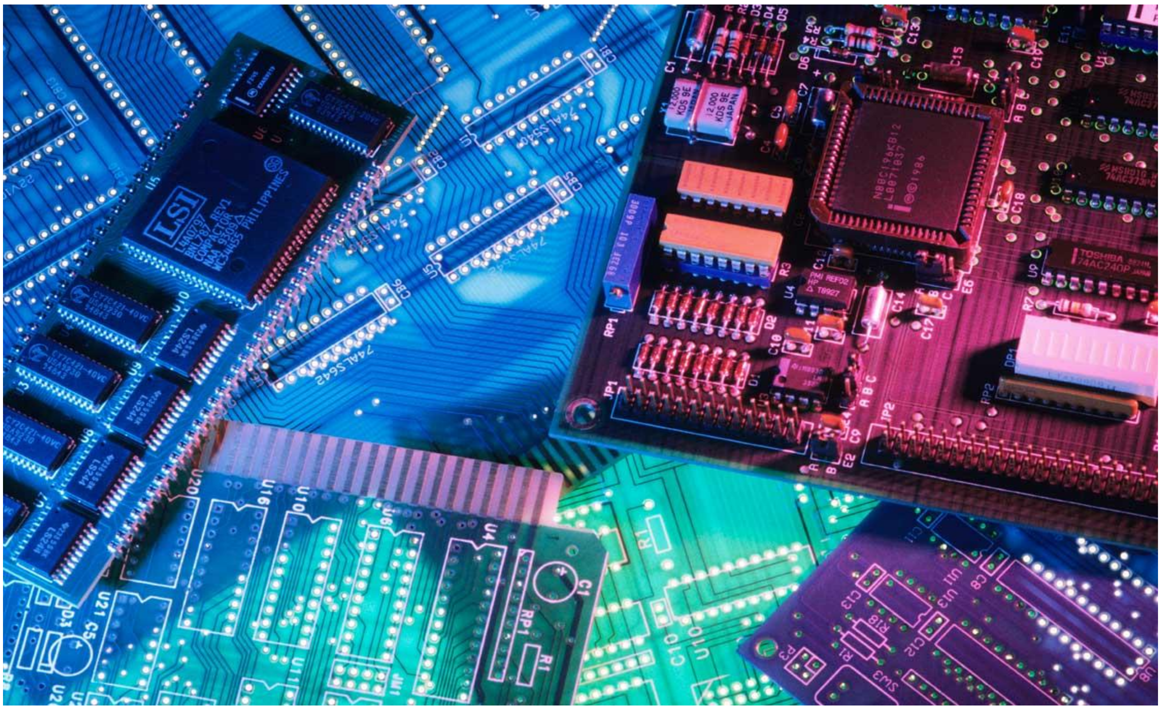
MICROPROCESSORS AND INTERFACING

III B.TECH I SEMESTER



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

LENDI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute, Approved by AICTE & Permanently Affiliated to JNTU-GV,
Vizianagaram)
(Accredited By NAAC with A Grade and Accredited by NBA Tier-1)
Jonnada (Village), Denkada (Mandal), Vizianagaram District – 535 005
Phone No. 08922-241111, 241112
E-Mail: lendi_2008@yahoo.com website: www.lendi.org

UNIT – II
Introduction to 8086 Microprocessor

Features of 8086, Architecture, Register organization of 8086, signal description of 8086, physical memory organization, general bus operation, I/O addressing capability, Minimum mode, maximum mode of 8086 system and timings.

INDEX

UNIT – II
INTRODUCTION TO 8086 MICROPROCESSOR

A **Microprocessor** is an Integrated Circuit with all the functions of a CPU however, it cannot be used stand-alone since unlike a microcontroller it *has no memory or peripherals*. 8086 does not have a RAM or ROM inside it. However, it has *internal registers* for storing intermediate and final results and interfaces with memory located outside it through the System Bus.

In case of 8086, it is a 16-bit **Integer processor** in a 40 pin, Dual Inline Packaged IC. The size of the internal registers (present within the chip) indicate how much information the processor can operate on at a time *(in this case 16-bit registers*) and how it moves data around internally within the chip, sometimes also referred to as the internal data bus. 8086 provides the programmer with 14 internal registers, each 16 bits or 2 Bytes wide.

FEATURES OF 8086 MICROPROCESSOR:

1. Intel 8086 was launched in 1978.
2. It was the first 16-bit microprocessor.
3. This microprocessor had major improvement over the execution speed of 8085.
4. It is available as 40-pin Dual-Inline-Package (DIP).
5. It is available in three versions:
- **a.** 8086 (5 MHz)
- **b.** 8086-2 (8 MHz)
- **c.** 8086-1 (10 MHz)

6. It consists of 29,000 transistors.
7. 8086 has a 20 bit address bus can access up to 2^{20} memory locations (1 MB).
8. It can support up to 64K I/O ports.
9. It provides 14, 16 -bit registers.
10. It has multiplexed address and data bus AD_0- AD_{15} and A_{16} – A_{19}.
11. It requires single phase clock with 33% duty cycle to provide internal timing.
12. 8086 is designed to operate in two modes, Minimum and Maximum.
13. It can prefetches up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.
14. It requires +5V power supply.

ARCHITECTURE OF 8086

The 8086 CPU is divided into two independent functional units:

1. Bus Interface Unit (BIU)
2. Execution Unit (EU)

Bus Interface Unit (BIU)

The bus interface unit is responsible for physical address calculations and predecoding instruction byte queue (6 byte long). It makes the system bus signals available for external devices. The 8086 addresses a segmented memory. The complete physical address which is 20-bit long is generated using segment and offset registers, each 16-bit long.

The function of BIU is to:

- Fetch the instruction or data from memory.
- Write the data to memory.
- Write the data to the port.
- Read data from the port.

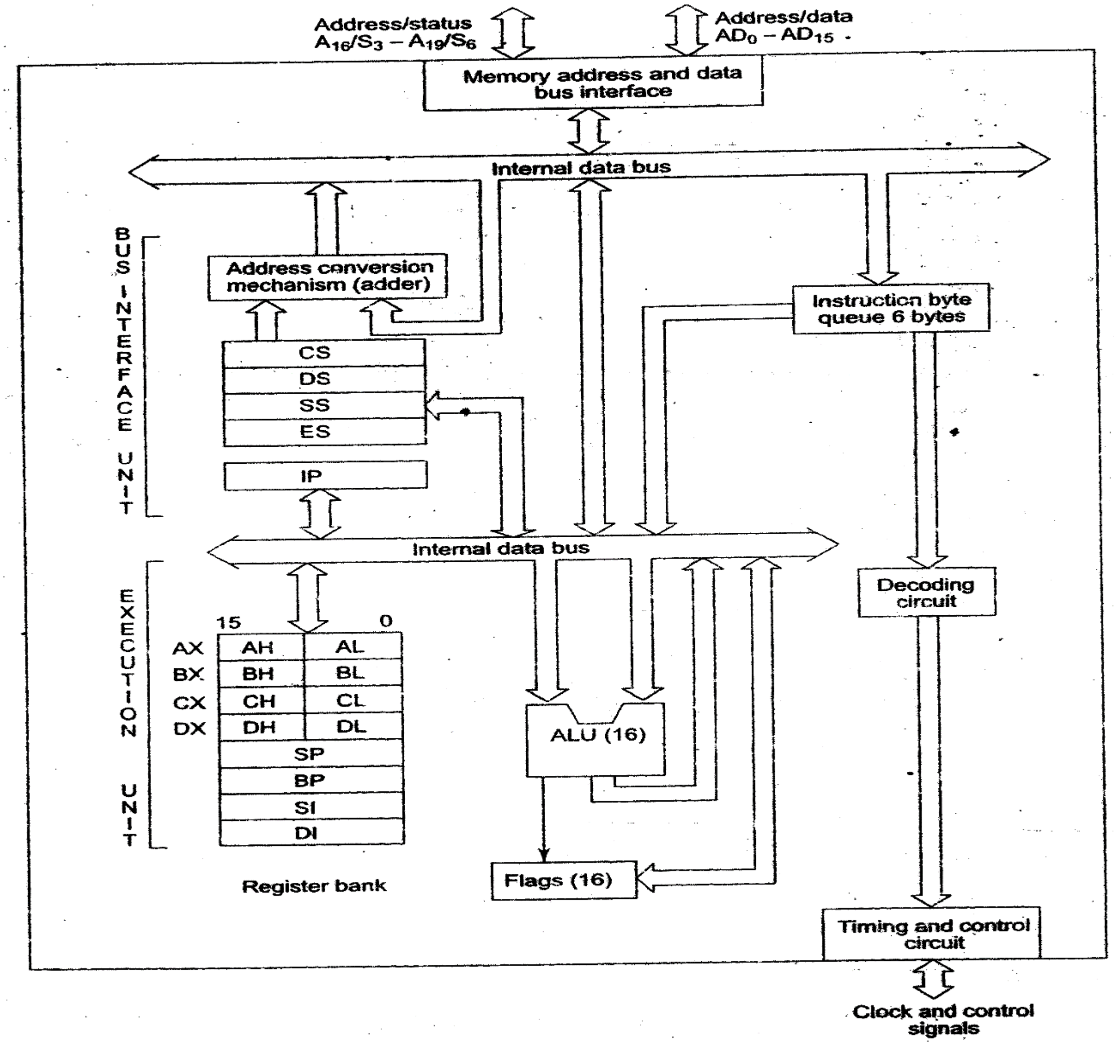


Figure: Architecture of 8086

Generating a physical address:

The content of segment register (segment address) is shifted left bit-wise four times. The content of an offset register (offset address) is added to the result of the previous shift operation. These two operations together produce a 20-bit physical address. For example, consider the segment address is 2010 H and the offset address is 3535 H.

The physical address is calculated as:

Segment Address	2010H		0010	0000	0001	0000
Shifted left by 4 bit positions		0010	0000	0001	0000	0000
Offset address	3535H		0011	0101	0011	0101
Physical address		0010	0011	0110	0011	0101
		2	3	6	3	5H

The segment address by the segment value 2010H can have offset value from 0000 H to FFFFH within it, i.e. Maximum 64K locations may be accommodated in the segment. The physical address range for this segment is from 20100 H to 300FFH.

The segment register indicates the base address of a particular segment and CS, DS, SS and ES are used to keep the segment address. The offset indicates the distance of the required memory location in the segment from the base address, and the offset may be the content of register IP, BP, SI, DI and SP. Once the opcode is fetched and decoded, the external bus becomes free while the Execution Unit is executing the instruction.

Instruction Queue

While the fetched instruction is executed internally, the external bus is used to fetch the machine code of the next instruction and arrange it in a queue called as pre decoded instruction byte queue. This is a 6 byte long queue, works in first-in first-out policy.

1. To increase the execution speed, BIU fetches as many as six instruction bytes ahead to time from memory.

2. All six bytes are then held in first in first out 6 byte register called instruction queue.

3. Then all bytes have to be given to EU one by one.

4. This pre fetching operation of BIU may be in parallel with execution operation of EU, which improves the speed execution of the instruction.

Execution Unit (EU)

The execution unit contains:

- Register set of 8086 except segment registers and IP.
- 16-bit ALU to perform arithmetic and logic operation
- 16-bit flag register reflects the results of execution by the ALU.
- Decoding units decodes the op-code bytes issued from the instruction byte queue.
- Timing and control unit generates the necessary control signals to execute the instruction op-code received from the queue.

The execution unit may pass the results to the bus interface unit for storing them in memory.

The functions of execution unit are:

- To tell BIU where to fetch the instructions or data from.
- To decode the instructions.
- To execute the instructions.

REGISTER ORGANIZATION OF 8086

All the registers of 8086 are 16-bit registers. The general purpose registers can be used as either 8-bit registers or 16-bit registers. The register set of 8086 can be categorized into 4 different groups. The register organization of 8086 is shown below:

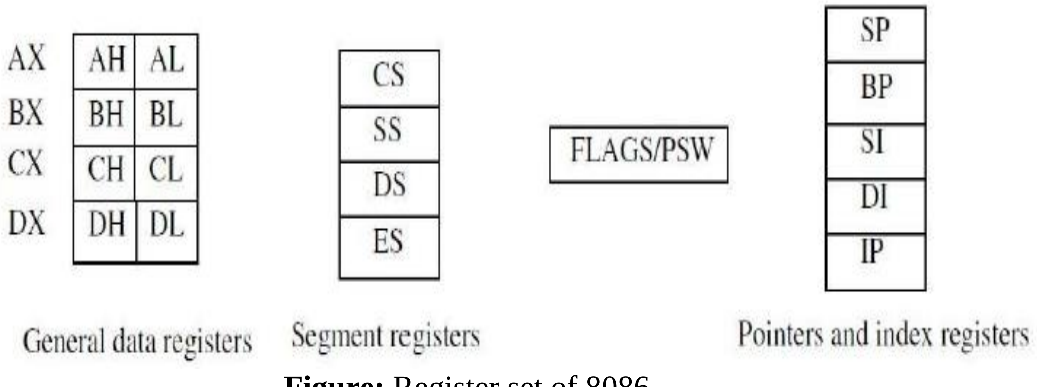


Figure: Register set of 8086

GENERAL PURPOSE REGISTERS

Accumulator (AX) register consists of 2 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 (BX) register consists of 2 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 (CX) register consists of 2 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 as a counter in string manipulation and shift/rotate instructions.

Data (DX) register consists of 2 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.

SPECIAL PURPOSE REGISTERS

Segment Registers:

The 8086 architecture uses the concept of segmented memory. 8086 able to address to address a memory capacity of 1 megabyte and it is byte organized. This 1 megabyte memory is divided into 16 logical segments. Each segment contains 64 Kbytes of memory. There are four segment register in 8086. They are:

- Segment register (CS)
- Data Segment register (DS)
- Extra Segment register (ES)
- Stack Segment register (SS)

Code Segment Register (CS): is used for addressing memory location in the code segment of the memory, where the executable program is stored.

Data Segment Register (DS): points to the data segment of the memory where the data is stored.

Extra Segment Register (ES): also refers to a segment in the memory which is another data segment in the memory.

Stack Segment Register (SS): is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

While addressing any location in the memory bank, the physical address is calculated from two parts:

- The first is segment address, the segment registers contain 16-bit segment base addresses, related to different segment.
- The second part is the offset value in that segment.

The advantage of this scheme is that in place of maintaining a 20-bit register for a physical address, the processor just maintains two 16-bit registers which is within the memory capacity of the machine.

Pointers and Index Registers:
The pointers contain offset within the particular segments.

- The pointer register *IP* **(Instruction Pointer)** contains offset within the code segment.
- The pointer register *BP* **(Base Pointer)** contains offset within the data segment.
- The pointer register *SP* (**Stack Pointer**) contains offset within the stack segment.

The index registers are used as general purpose registers as well as for offset storage in case of indexed, base indexed and relative base indexed addressing modes. The register *SI* **(Source Index)** is used to store the offset of source data in data segment. The register *DI* **(Destination Index)** is used to store the offset of destination in data or extra segment. The index registers are particularly useful for string manipulation.

The following table describes the default offset values to the corresponding memory segments.

Segment	**Offset Registers**	**Function**
CS	IP	Address of the next instruction
DS	BX, DI, SI, BP	Address of data
SS	SP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

FLAG REGISTER

Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program. The 8086 flag register has 9 active flags and they are divided into two categories:

The complete bit configuration of 8086 is shown in the below figure.

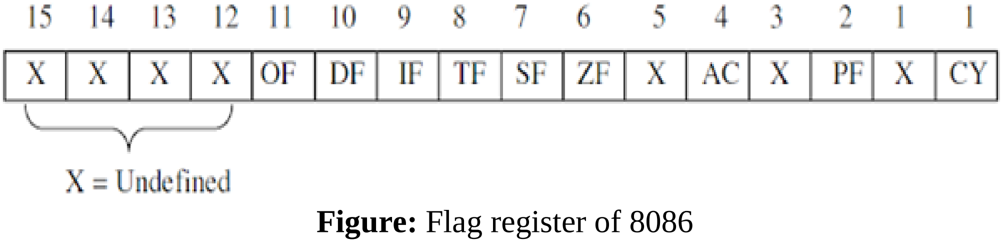


Figure: Flag register of 8086

1. Condition code or status flags
2. Machine control 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.

Over flow Flag (OF):

This flag is set, if an overflow occurs, i.e., if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Control Flags are as follows:

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):

This is used by string manipulation instructions. If this flag bit is ‘0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

SIGNAL DESCRIPTION OF 8086

The 8086 is a 16-bit microprocessor. This microprocessor operates in single processor or multiprocessor configurations to achieve high performance. The pin configuration of 8086 is shown in the below. Some of the pins serve a particular function in minimum mode (single processor mode) and others function in maximum mode (multiprocessor mode).

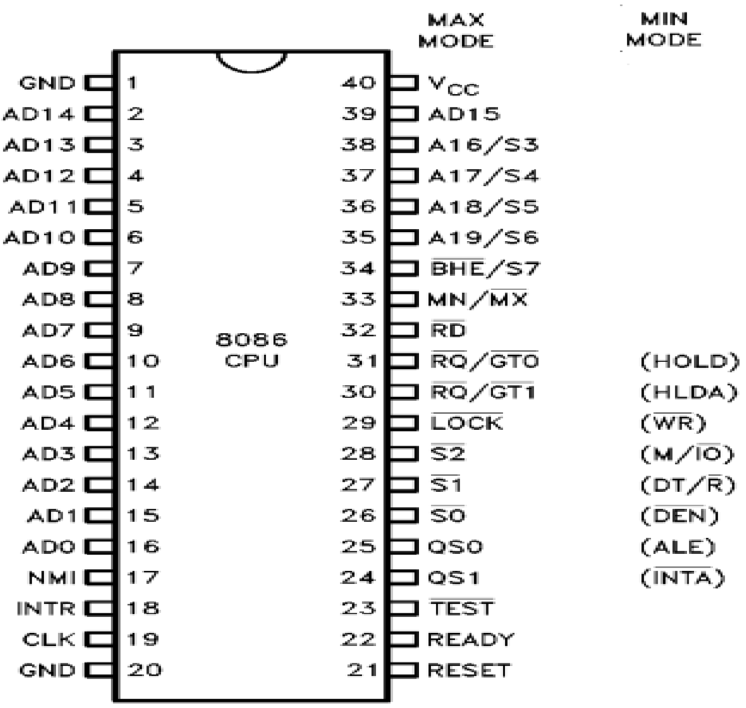


Figure: Pin diagram of 8086

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.

AD15-AD0:

These are the time multiplexed memory I/O address and data lines. Address remains on the lines during T1 state, while the data is available on the data bus during T2, T3, TW and T4. Here T1, T2, T3, T4 and TW are the clock states of a machine cycle. TW is a wait state. These lines are active high and float to a tristate during interrupt acknowledge and local bus hold acknowledge cycles.

A19/S6, A18/S5, A17/S4, A16/S3:

These are the time multiplexed address and status lines. During T1, these are the most significant address lines or memory operations. During I/O operations, these lines are low. During memory or I/O operations, status information is available on those lines for T2, T3, TW and T4 .The status of the **interrupt enable flag bit** (displayed on **S5**) is updated at the beginning of each clock cycle. The **S4 and S3** combined indicate **which segment register is presently being used** for memory accesses. These lines float to tristate during the local bus hold acknowledge. The status line **S6 is always low** (logical). The address bits are separated from the status bits using latches controlled by the ALE signal.

S4	S3	Indication
0	0	Alternate Data
0	1	Stack
1	0	Code or none
1	1	Data

BHE’/S7 (Active Low):

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. The status information is available during T2, T3 and T4. The signal is active low and is tristated during 'hold'. It is low during T1 for the first pulse of the interrupt acknowledge cycle.

BHE’	A0	Indication
0	0	Whole word
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None

RD’ (Read) (Active Low)**:**

Read signal, when low, indicates the peripherals that the processor is performing a memory or I/O read operation. **RD’** is active low and shows the state for T2, T3, and TW of any read cycle. The signal remains tristated during the 'hold acknowledge'.

READY:

This is the acknowledgement from the slow devices 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.

INTR-Interrupt Request:

This is a level triggered input. This is sampled during the last clock cycle 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 resetting the interrupt enable flag. This signal is active high and internally synchronized.

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.

NMI-Non-maskable Interrupt:

This is an edge-triggered input which causes a Type2 interrupt. The NMI is not maskable internally by software. A transition from low to high initiates the interrupt response at the end of the current instruction. This input is internally synchronized.

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.

CLK: Clock Input

The clock input provides the basic timing for processor operation and bus control activity. It's an asymmetric square wave with 33% duty cycle. The range of frequency for different 8086 versions is from 5MHz to 10MHz.

VCC:

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

MN/MX':

The logic level at this pin decides whether the processor is to operate in either minimum (single processor) or maximum (multiprocessor) mode.

GND – Ground

The following pin functions are for **the maximum mode operation** of 8086.

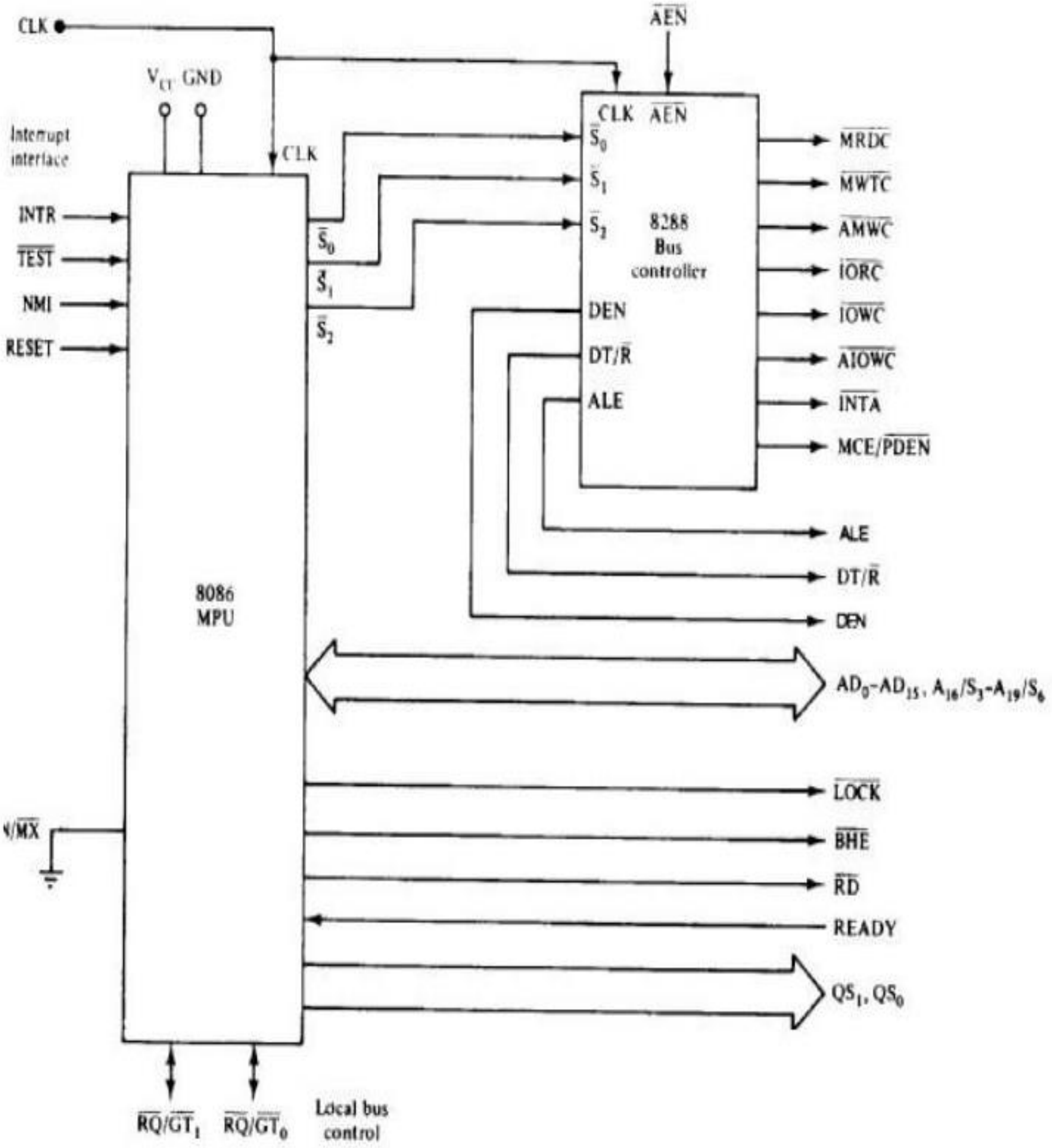


Figure: Maximum mode pins and operation of 8086 microprocessor

RQ'/GT1' RQ'/GT0': Request/Grant:

These pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with RQ'/GT0' having higher priority than RQ' /GT_1'. The Request/ Grant sequence is as follows:

1. A pulse one clock wide from another bus master requests the bus access to 8086.

2. During T4 (current) or T1 (next) clock cycle, a pulse one clock wide from 8086 to the requesting master, indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state at next clock cycle. The CPU's bus interface unit is likely to be disconnected from the local bus of the system.

3. A one clock wide pulse from another master indicates to 8086 that the 'hold' request is about to end and the 8086 may regain control of the local bus at the next clock cycle.

QS_1, QS_0: **Queue Status:**

These lines give information about the status of the code pre-fetch queue. These are active during the CLK cycle after which the queue operation is performed.

QS_I	QS_0	Status
0	0	No Operation
0	1	First Byte of Op Code from Queue
1	0	Empty the Queue
1	1	Subsequent Byte from Queue

S_0', S_1', S_2':

These are the status lines which reflect the type of operation, being carried out by the processor. These become active during T4 of the previous cycle and remain active during T1 and T2 of the current bus cycle. The status lines return to passive state during T3 of the current bus cycle so that they may again become active for the next bus cycle during T4. Any change in these lines during T3 indicates the starting of a new cycle, and return to passive state indicates end of the bus cycle.

(M/IO')	(DT/R')		
S_2	S_1	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':

This output pin indicates that other system bus masters will be prevented from gaining the system bus, while the signal is low. The signal is activated by the 'LOCK' prefix instruction and remains active until the completion of the next instruction. This floats to tristate off during "hold acknowledge". 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. The 8086, while executing the prefixed instruction, asserts the bus lock signal output, which may be connected to an external bus controller.

The following pin functions are for the **minimum mode operation** of 8086.

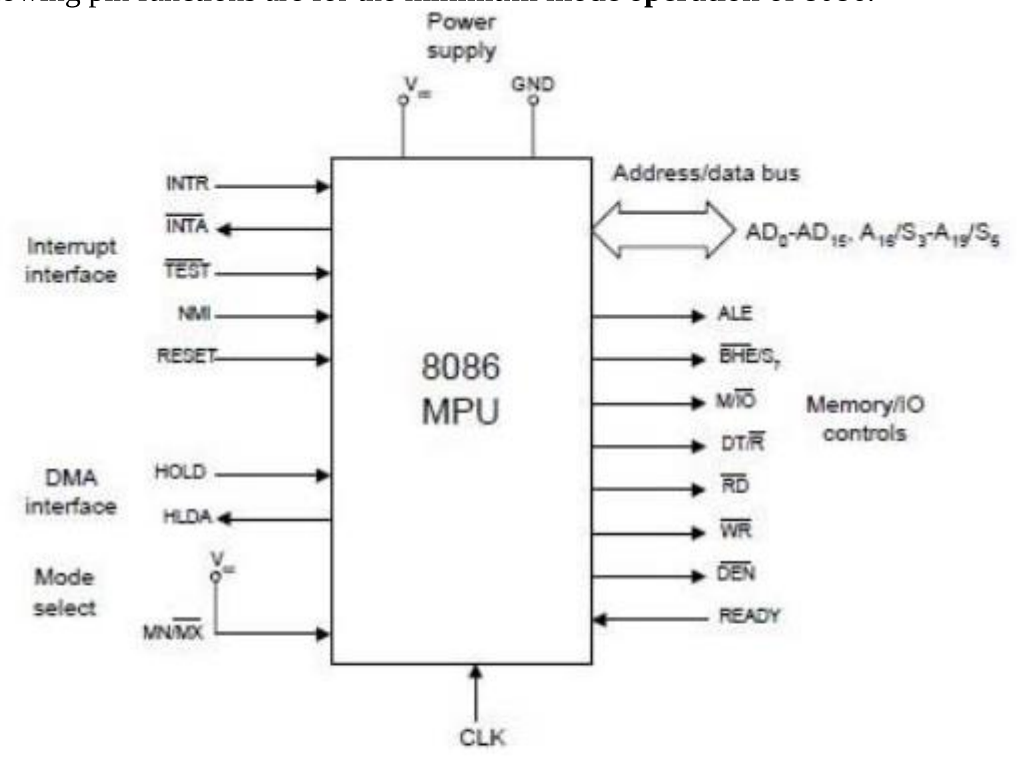


Figure: Minimum mode pins and operation of 8086 microprocessor

DT/R': Data Transmit/Receive:

This output is used to decide the direction of data flow through the transceivers (bidirectional buffers). When the processor sends out data, this signal is high and when the processor is receiving data, this signal is low. Logically, this is equivalent to S1 in maximum mode. Its timing is the same as M/I/O'. This is tristated during 'hold acknowledge'.

DEN': Data Enable:

This signal indicates the availability of valid data over the address/data lines. It is used to enable the transreceivers (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. DEN is tristated during 'hold acknowledge' cycle.

HOLD/HOLDA:

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 (instruction) 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 it should be externally synchronized.

INTA’: Interrupt Acknowledge:

This signal is used as a read strobe for interrupt acknowledge cycles. In other words, when it goes low, it means that the processor has accepted the interrupt. It is active low during T2, T3 and TW of each interrupt acknowledge cycle.

M/IO’:

This is a status line logically equivalent to S2 in maximum mode. When it is low, it indicates the CPU is having an I/O operation, and when it is high, it indicates that the CPU is having a memory operation. This line becomes active in the previous T4 and remains active till final T4 of the current cycle. It is tristated during local bus "hold acknowledge".

WR’: Write:

It stands for write signal and is available at pin 29. It is used to write the data into the memory or the output device depending on the status of M/IO signal.

ALE: Address Latch Enable:

This output signal indicates the availability of the valid address on the address/data lines, and is connected to latch enable input of latches. This signal is active high and is never tristated.

PHYSICAL MEMORY ORGANIZATION

In an 8086 based system, the 1Mbyte memory is physically organized as odd bank and even bank, each of 512kbytes, addressed in parallel by the processor. Byte data with even address is transferred on $D7 - D0$ and byte data with an odd address is transferred on $D15 - D8$ bus lines. The processor provides two enable signals, *BHE’* and *A*0 for selecting of either even or odd or both the banks.

$\overline{BHE}$	A_0	**Function**
0	0	Whole word
0	1	Upper byte/ odd address
1	0	Lower byte/even address
1	1	None

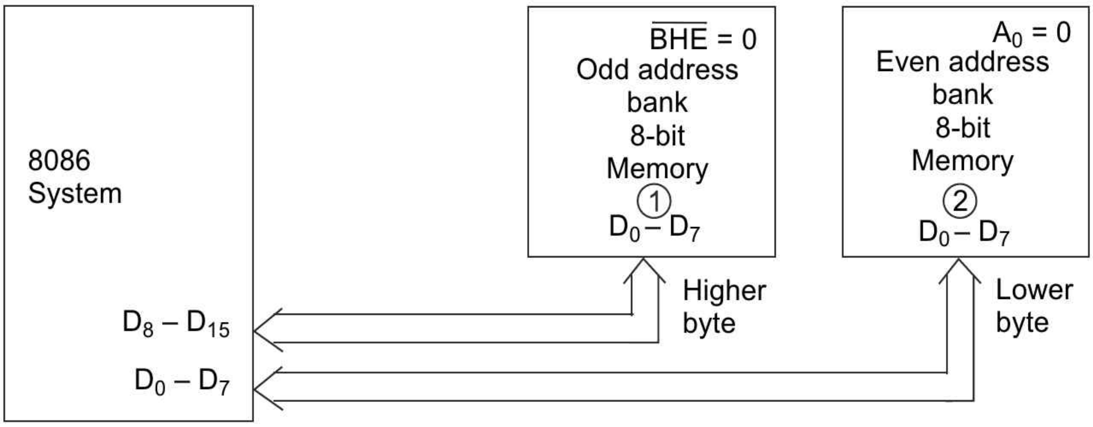


Figure: Physical Memory Organization

If the address bus contains $FFFFO_H$

If $\overline{BHE} = 0$, then it reads the 16 bits data from memory location $FFFFO_H$ and $FFFF1_H$.

If $\overline{BHE} = 1$, it reads the 8 bits data from memory location $FFFFO_H$

Certain locations in memory are reserved for specific CPU operations. Locations from address FFFF0H through FFFFFH are reserved for operations including a jump to the initial program loading routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be.

Locations 00000H through 003FFH are reserved for interrupt operations. Each of the 256 possible interrupt types has its service routine pointed to by a 4-byte pointer element consisting of a 16-bit segment address and a 16-bit offset address. The pointer elements are assumed to have been stored at the respective places in reserved memory prior to occurrence of interrupts.

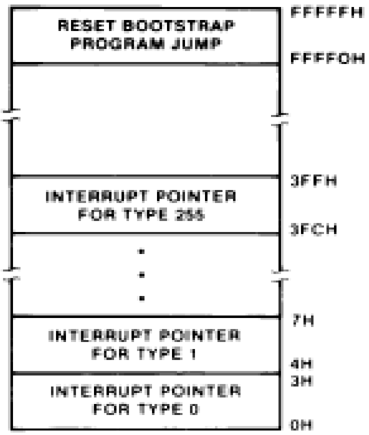

RESET BOOTSTRAP PROGRAM JUMP
FFFFFH
FFFF0H
3FFH
INTERRUPT POINTER FOR TYPE 255
3FCH
7H
INTERRUPT POINTER FOR TYPE 1
4H
3H
INTERRUPT POINTER FOR TYPE 0
0H

Figure: Reserved memory locations

GENERAL BUS OPERATION

The 8086 has a combined address and data bus commonly referred to as a time multiplexed bus. This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This ‘‘local bus’’ can be buffered directly and used throughout the system with address latching provided on memory and I/O modules. In addition, the bus can also be demultiplexed at the processor with a single set of address latches if a standard non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3 and T4. The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a ‘‘NOT READY’’ indication is given by the addressed device, ‘‘Wait’’ states (TW) are inserted between T3 and T4. Each inserted ‘‘Wait’’ state is of the same duration as a CLK cycle. Periods can occur between 8086 bus cycles. These are referred to as ‘‘Idle’’ states (Ti) or inactive CLK cycles. The processor uses these cycles for internal house-keeping.

During T1 of any bus cycle the ALE (Address Latch Enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/MX strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

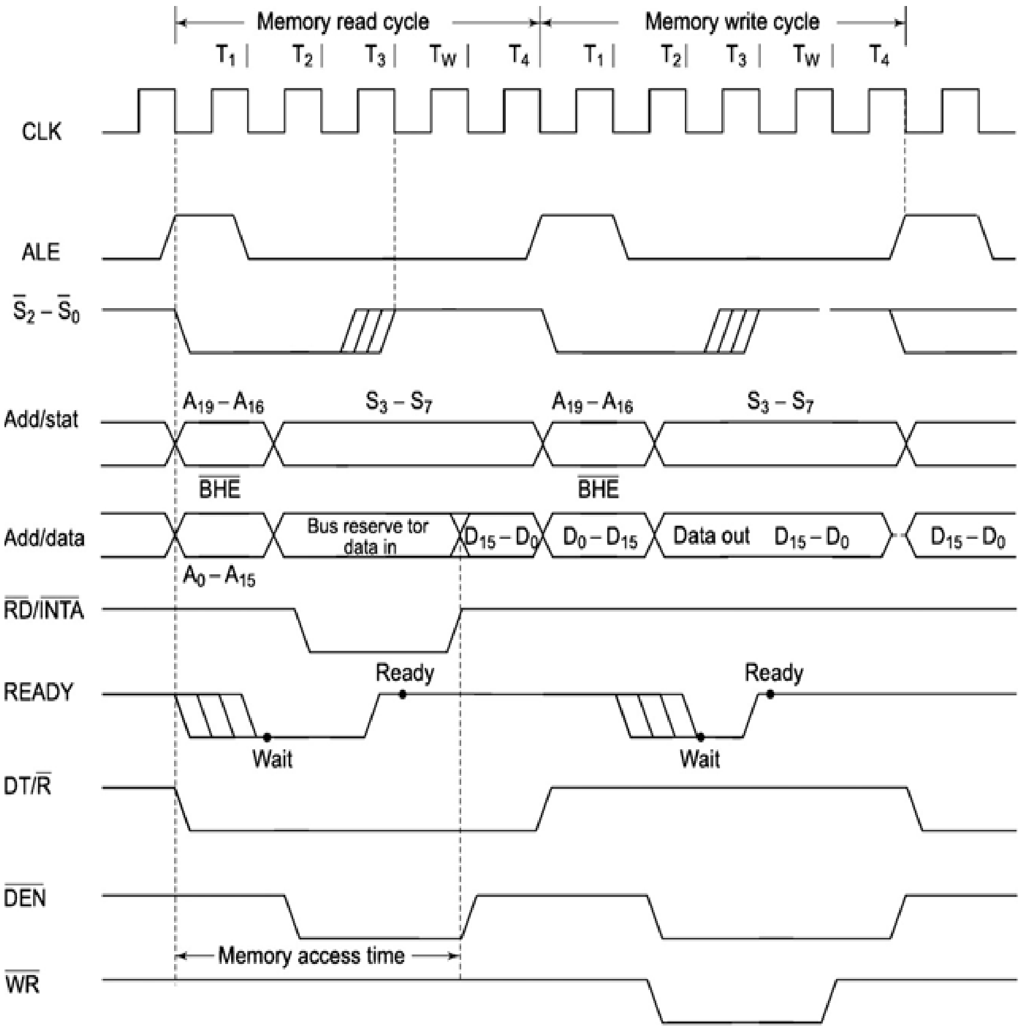


Figure: General bus operation of 8086 processor

Maximum Mode

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

ii. 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.

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

Minimum Mode

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

ii. In this mode, all the control signals are given out by the microprocessor chip itself.

I/O ADDRESSING CAPABILITY

The 8086 processor can address up to 64K byte I/O registers or 32K word registers. The I/O address appears on the address lines A0 to A15 for one clock cycle (T1). It may then be latched using the ALE signal. The upper address lines (A16 - A19) are at logic 0 level during the I/O operations.

The 16-bitregister DX is used as 16-bit I/O address pointer, with full capability to address up to 64K devices. In memory mapped I/O interfacing, the I/O device addresses are treated as memory locations in page 0, i.e. segment address 0000H. Even addressed bytes are transferred on D7-D0 and odd addressed bytes are transferred on D8-D15 lines.

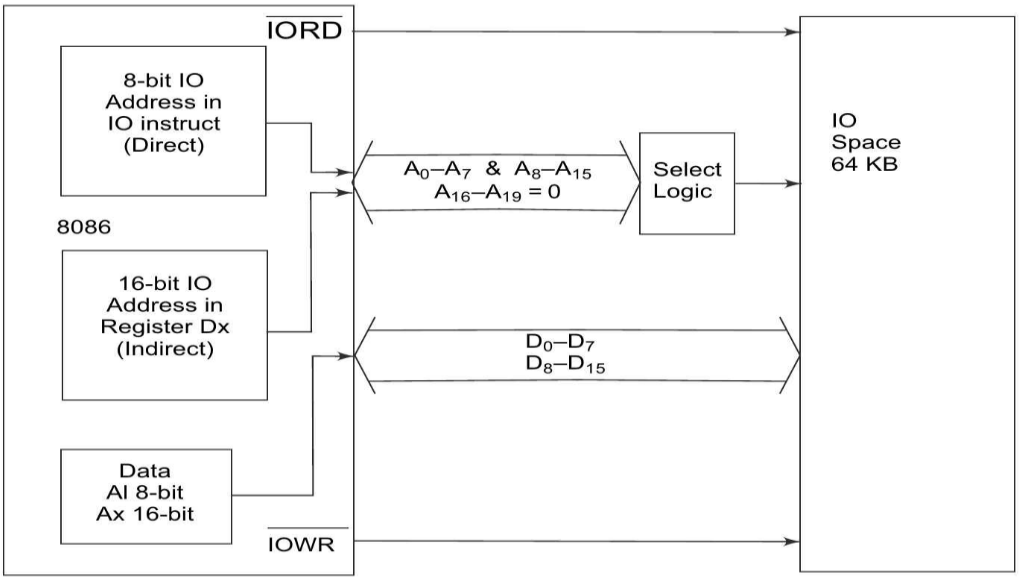


Figure: 8086 IO addressing

Example:

IN AL, 0C8H; Input a byte from port 0C8 to AL

IN AX, 34H; Input a word from port 34H to AX

OUT 3BH, AL; Copy the content of AL to port 3BH

OUT 2CH, AX; Copy the contents of AX to port 2CH

MOV DX, 0FF78H; Initialize DX to point to port

IN AL, DX; Input a byte from 8-bit port 0FF78H to AL

IN AX, DX; Input a word from 16-bit port 0FF78H to AL

MOV DX, 0FFF8H; Initialize DX to point to port

OUT DX, AL; Copy the content of AL to 0FFF8H

OUT DX, AX; Copy the content of AX to 0FFF8H

MINIMUM MODE OF 8086 SYSTEM AND TIMINGS

In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/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, 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.

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

The **DEN**' signal **indicates that the valid data** is available on the data bus, while **DT/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 users 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. 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.

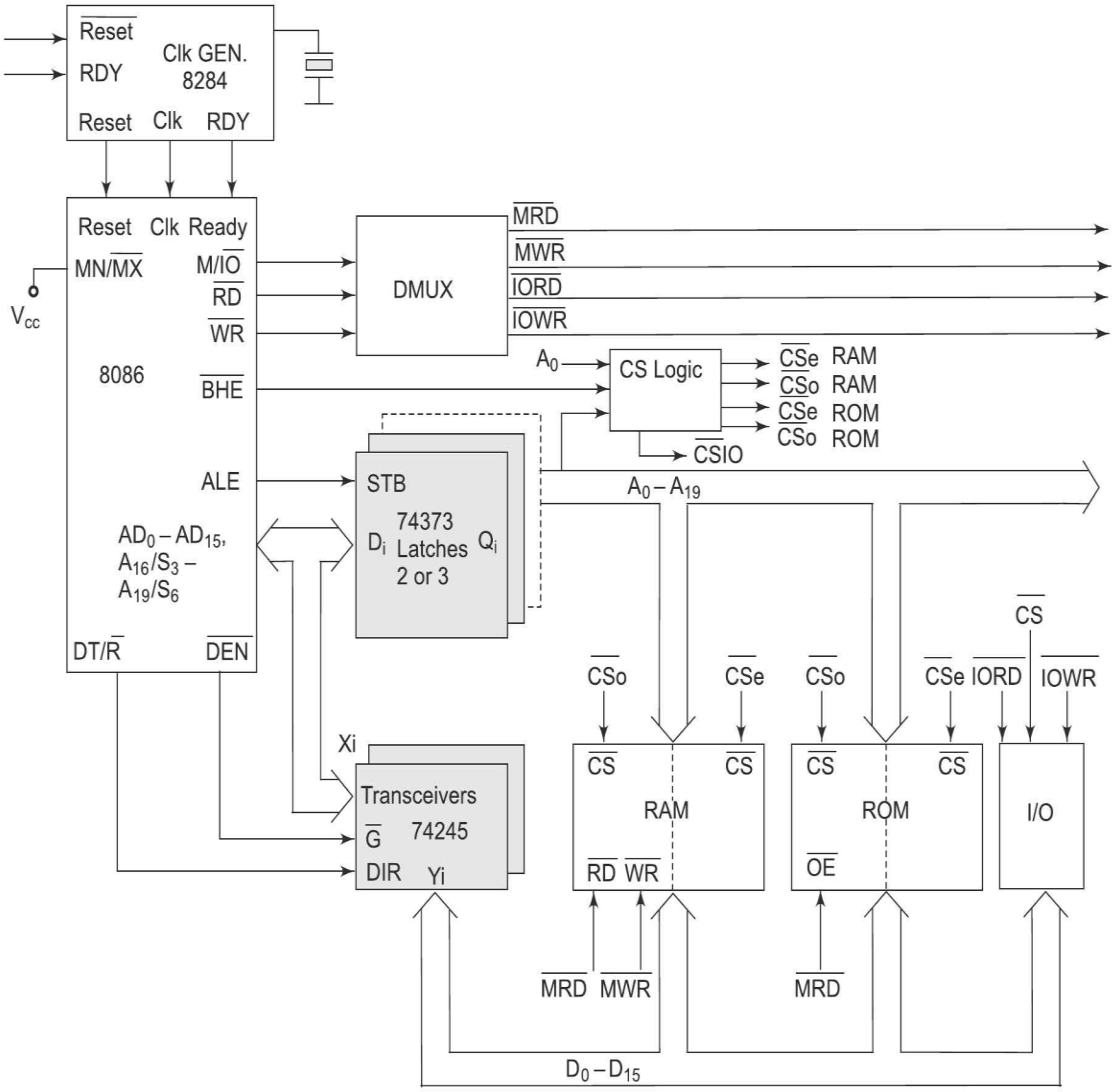


Figure: Minimum mode 8086 system

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.

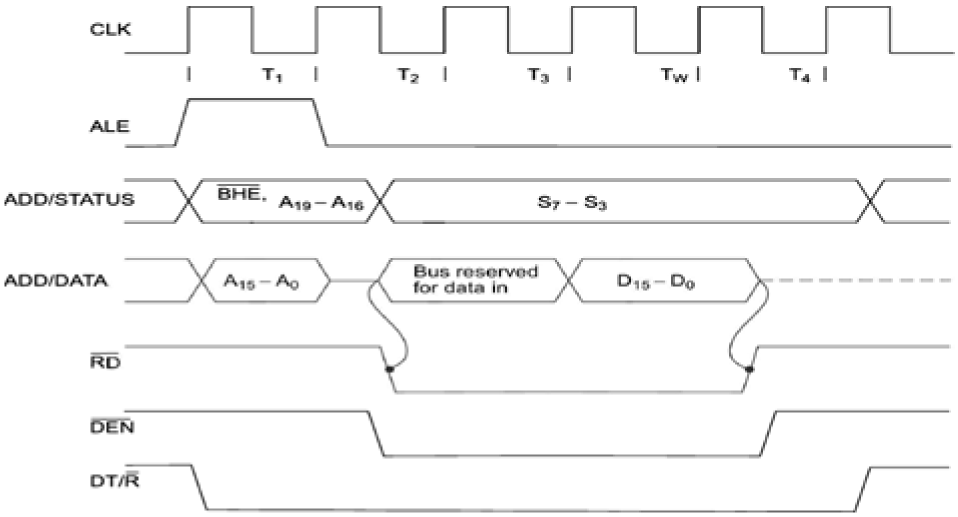


Figure: Read timing diagram of 8086 minimum mode system

The above figure 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/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 Tl 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 addressed 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.

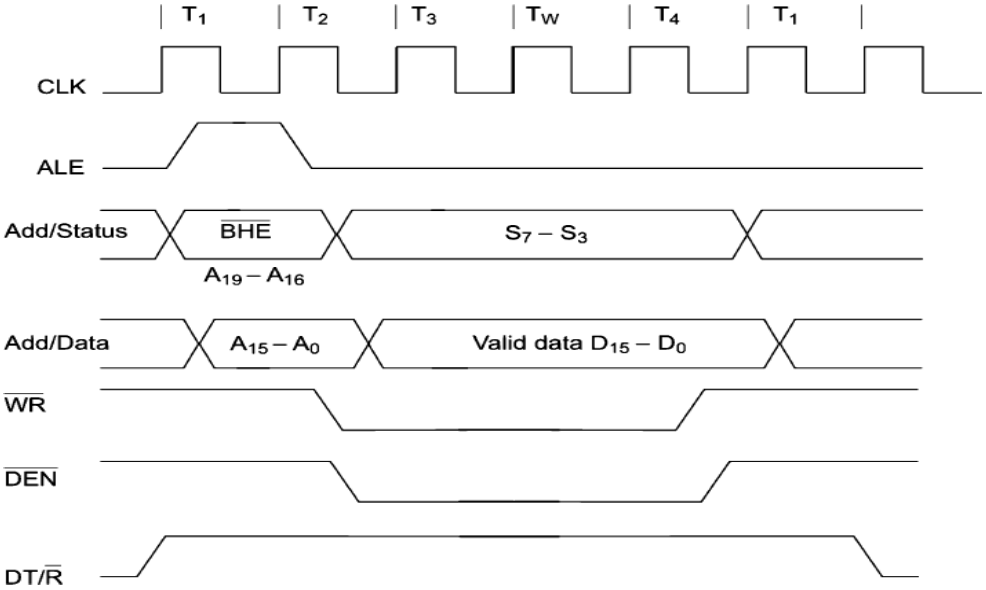


Figure: Write timing diagram of 8086 minimum mode system

The above figure shows the write cycle timing diagram. 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 Tl 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 ofT2 (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 written. The M/IO', RD' and WR' signals indicate the types of data transfer as specified in Table

M/IO'	**RD'**	**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

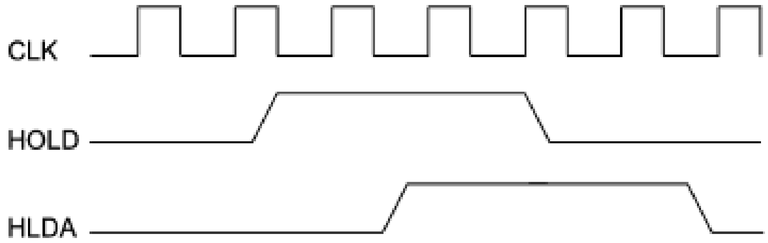


Figure: Bus request and Bus Grant Timings in Minimum mode

MAXIMUM MODE OF 8086 SYSTEM AND TIMINGS

In the maximum mode, the 8086 is operated by strapping the MN/MX' pin to ground. In this mode, the processor derives the status signals S2', S1' and S0'. 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 below figure.

The basic functions 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 made available by the processor on the status lines. The bus controller chip has input lines S2', S1' and S0' and CLK. These inputs to 8288 are driven by the CPU. It derives the outputs ALE, DEN', DT/R', MWTC', AMWC', IORC', IOWC' and AIOWC'. The AEN', IOB and CEN pins are especially useful for multiprocessor systems. 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 cascaded 8259A; else it acts as peripheral data enable used in the multiple bus configurations. INTA' pin is 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 addressed port. The MRDC', 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 AIOWC' and AMWTC' are available. They also serve the same purpose, but are activated one clock cycle earlier than the IOWC' and MWTC' signals, respectively.

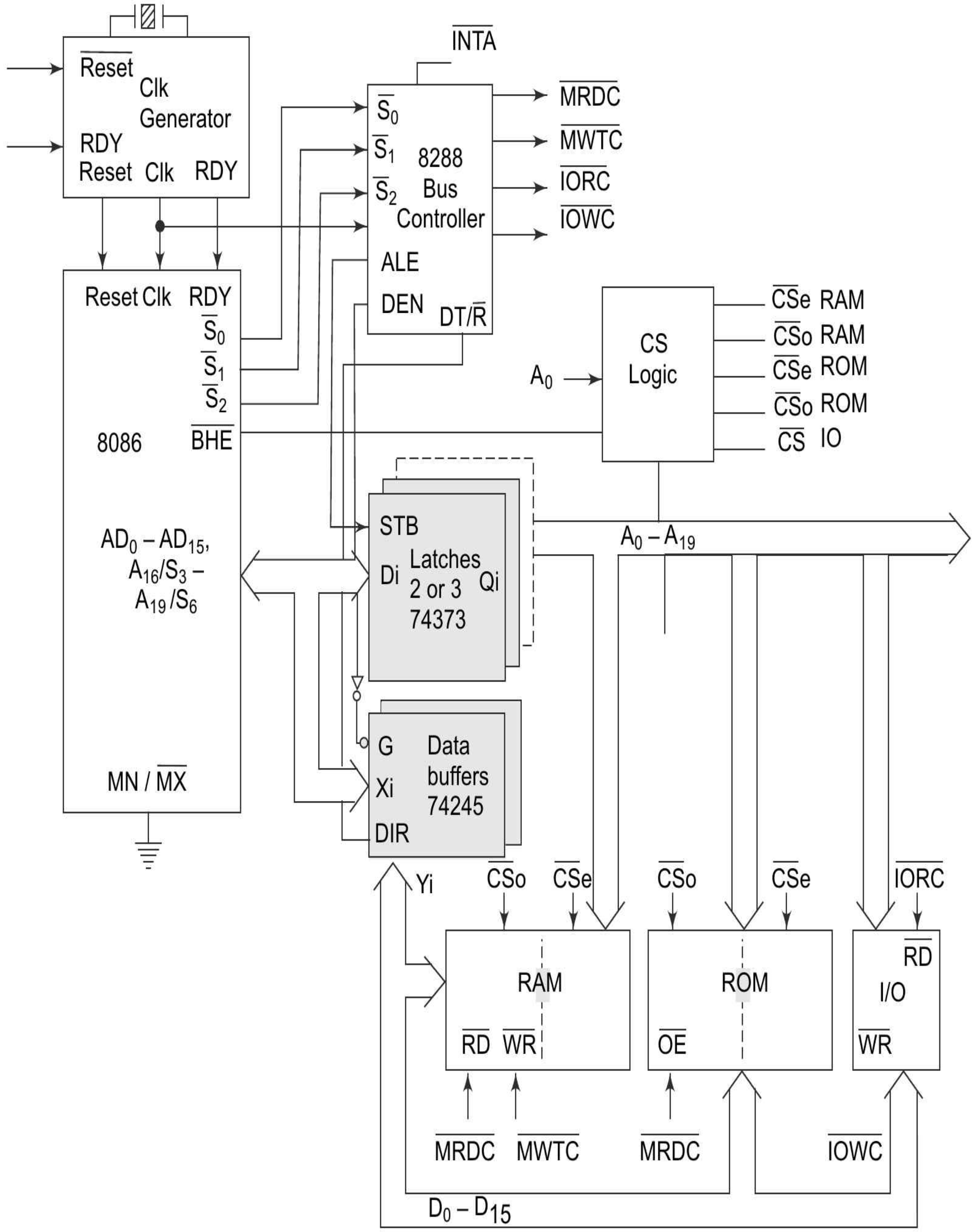


Figure: Maximum mode 8086 system

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.

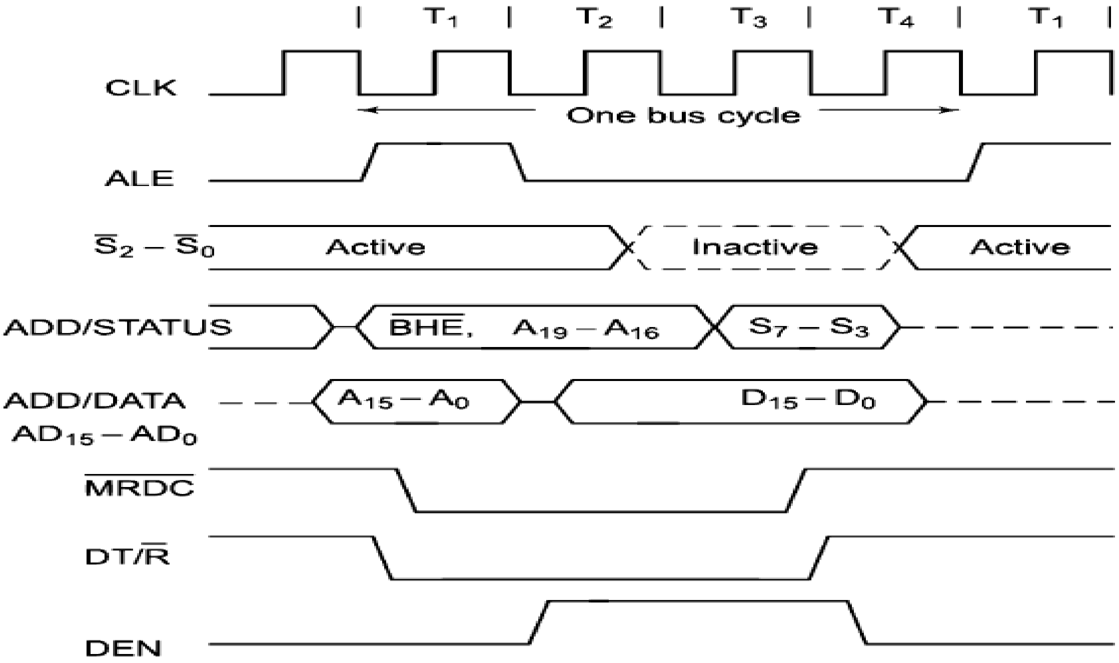


Figure: Read timing diagram of 8086 maximum mode system

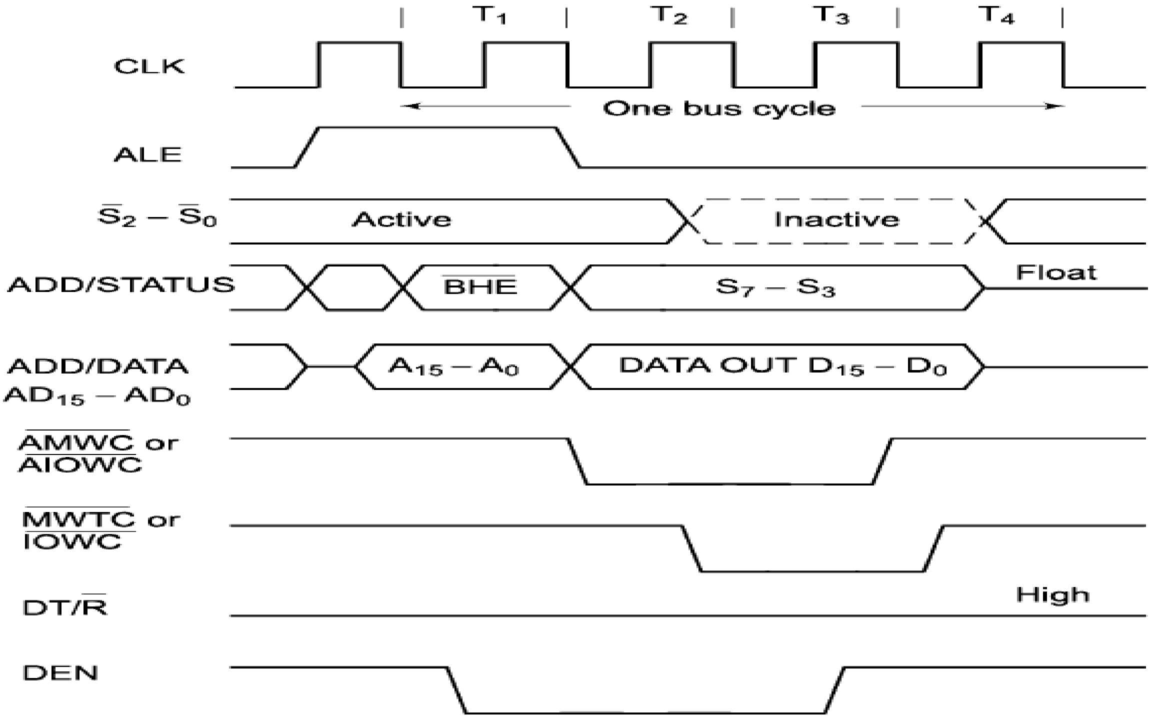


Figure: Write timing diagram of 8086 maximum mode system

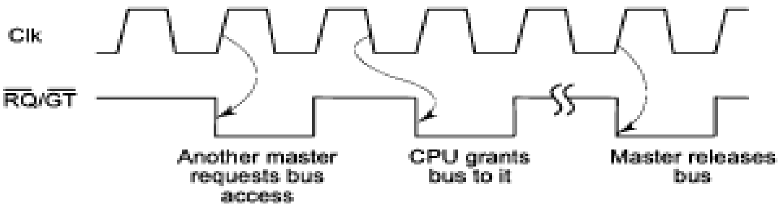


Figure: RQ’/GT’ Timing in Maximum mode