Assembly Language:

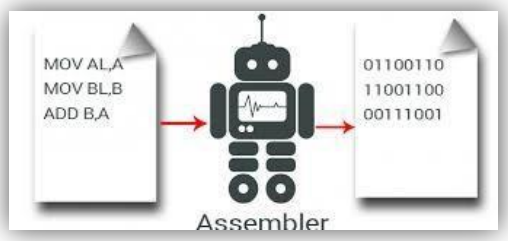
In computer science, an assembler is a program that turns assembly language into machine code. An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. Some people call these instructions assembler language and others use the term assembly language.

Assembly code is converted into executable machine code by a utility program referred to as an assembler. The conversion process is referred to as assembly, as in assembling the source code. Assembly language usually has one statement per machine instruction (1:1).

Most computers come with a specified set of very basic instructions that correspond to the basic machine operations that the computer can perform. For example, a "Load" instruction causes the processor to move a string of bits from a location in the processor's memory to a special holding place called a register. Assuming the processor has at least eight registers, each numbered, the following instruction would move the value (string of bits of a certain length) at memory location 3000 into the holding place called register 8: (L 8,3000).

The programmer can write a program using a sequence of these assembler instructions. This sequence of assembler instructions, known as the source code or source program. The assembler program takes each program statement in the source program and generates a corresponding bit-stream or pattern (a series of 0's and 1's of a given length). The output of the assembler program is called the object code or object program relative to the input source program.

The sequence of 0's and 1's that constitute the object program is sometimes called machine code. The object program can then be run (or executed) whenever desired. In the earliest computers, programmers actually wrote programs in machine code, but assembler languages or instruction sets were soon developed to speed up programming. Today, assembler programming is used only where very efficient control over processor operations is needed.

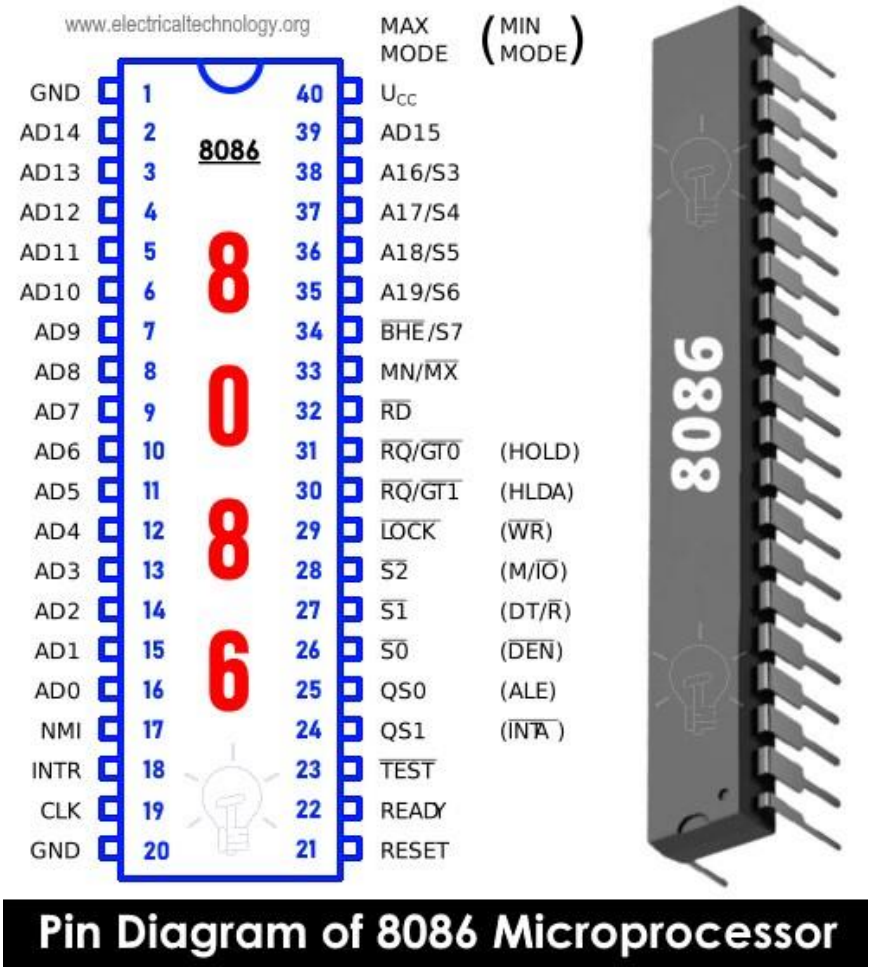


Microprocessor - 8086 Pin Configuration:

What is 8086 Microprocessor?

The 8086 is a 16-bit microprocessor introduced by Intel in 1976. It is the enhanced version of the 8085 microprocessor. It has a 16-bit data bus with a 16-bit ALU. The address bus is 20-bit long; therefore the accessible memory capacity of 8086 microprocessor is 2^{20} Bytes or 1MB. it is available in various versions with a clock frequency of 5 MHz, 8 MHz and 10 MHz.

8086 was the first 16-bit microprocessor available in 40-pin DIP (Dual Inline Package) chip. Let us now discuss in detail the pin configuration of a 8086 Microprocessor.



Pin Diagram of 8086 Microprocessor

Power supply and frequency signals:

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.

Address/data bus:

AD0-AD15. These are 16 address/data bus. AD0-AD7 carries low order byte data and AD8AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

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.

S7/BHE:

Bus High Enable/Status. It is available at pin 34 This signal is low during the first clock cycle, thereafter it is active, (During T1 it is low). It is used to indicate the transfer of data using data bus D8-D15. 8-bit device connected to the upper half of the data bus use BHE (Active / Low) signal. It is multiplexed with status signal S7.

Read ($\overline{RD}$):

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

Ready:

It is available at pin 22. It is an acknowledgment 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 used to reset the microprocessor. Registers, seg. regs, flags, CS: FFFFH, IP: 0000H.

INTR:

It is available at pin 18. It is an interrupt request signal, that can be disabled or ignored by the instructions of CPU. When interrupt occur, it can be handled after executing the current instruction. Interrupts help to handle lower priority tasks. The operation can be masked or made pending.

NMI: (non-maskable interrupt)

It is available at pin 17. The NMI is a hardware interrupt that cannot be ignored by the instructions of CPU. NMI used for emergency purposes e.g power failure

etc. Operation Cannot be masked or made pending. NMI helps to handle higher priority tasks.

$\overline{\text{TEST}}$:

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/$\overline{\text{MX}}$:

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.

INTA:

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

ALE: (Address Latch Enable)

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. ALE: Contains address bits A15-A0 when ALE is 1 & data bits D15 – D0 when ALE is 0.

DEN:

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

DT/R:

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

M/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 indicates the memory operation. It is available at pin 28.

WR:

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.

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.

QS1 and QS0:

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:

QS0	QS1	Status
0	0	No operation
0	1	First byte of op-code from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

S0, S1, S2:

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:

S2	S1	S0	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.

RQ/GT1 and RQ/GT0:

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. RQ/GT0 has a higher priority than RQ/GT1.