INTEL 8086 MICRORROCESSOR

The INTEL 8086 was the first 16-bit microprocessor, it was developed using HMOS (high density short channel MOS) technology containing 29,000 transistors housed in 40 pin DIP package.

Features of 8086 :-

1. The 8086 is a 16-bit microprocessor. The term 16-bit means that its ALU, internal registers and most of its instructions are designed to work with 16-bit binary words.
2. The 8086 has a 16-bit data bus, so it can read from or write data to memory and I/O ports either 16-bits or 8-bits at a time.
3. The 8086 has a 20-bit address bus, so it can directly access 2^{20} or 1048576 (1 MB) memory locations. Each of the 1 MB memory location is 8-bit wide. Therefore, a sixteen bit word is stored in two consecutive memory locations.
4. The 8086 can generate 16-bit I/O address, hence it can access $2^{16} = 65536$ I/O ports.
5. The 8086 provides fourteen, 16-bit registers.
6. The 8086 has multiplexed address and data bus which reduces the number of pins needed, but does slow down the transfer of data.
7. The 8086 does not have an internal clock circuit. The 8086 requires an external one phase clock with a 33% duty cycle to provide optimized internal timing. It requires 5 MHz clock range.
8. It performs the arithmetic and logical operations on bit, byte, word and decimal numbers including multiply and divide.
9. The Intel 8086 is designed to operate in two modes, namely the minimum mode and the maximum mode. When only one 8086 microprocessor is to be used in a microcomputer system,

the 8086 is used in the minimum mode of operation. In this mode the microprocessor issues the control signal required by memory and I/O devices.

In multiprocessors (more than one processor in the system) system 8086 operates in maximum mode. In maximum mode, the control signals are generated with the help of internal bus controller (8288).

10). The Intel 8086 supports multiprogramming. In multiprogramming, the code for two or more processes is stored in memory at the same time and is executed in a time-multiplexed fashion.

11). An interesting feature of the 8086 is that it fetches up to six instruction bytes from memory and queue stores them in order to speed up instruction execution.

2015-16

ARCHITECTURE OF 8086

The architecture of 8086 can be divided into two units.
- BIU - Bus Interface Unit.
- EU - Execution Unit.

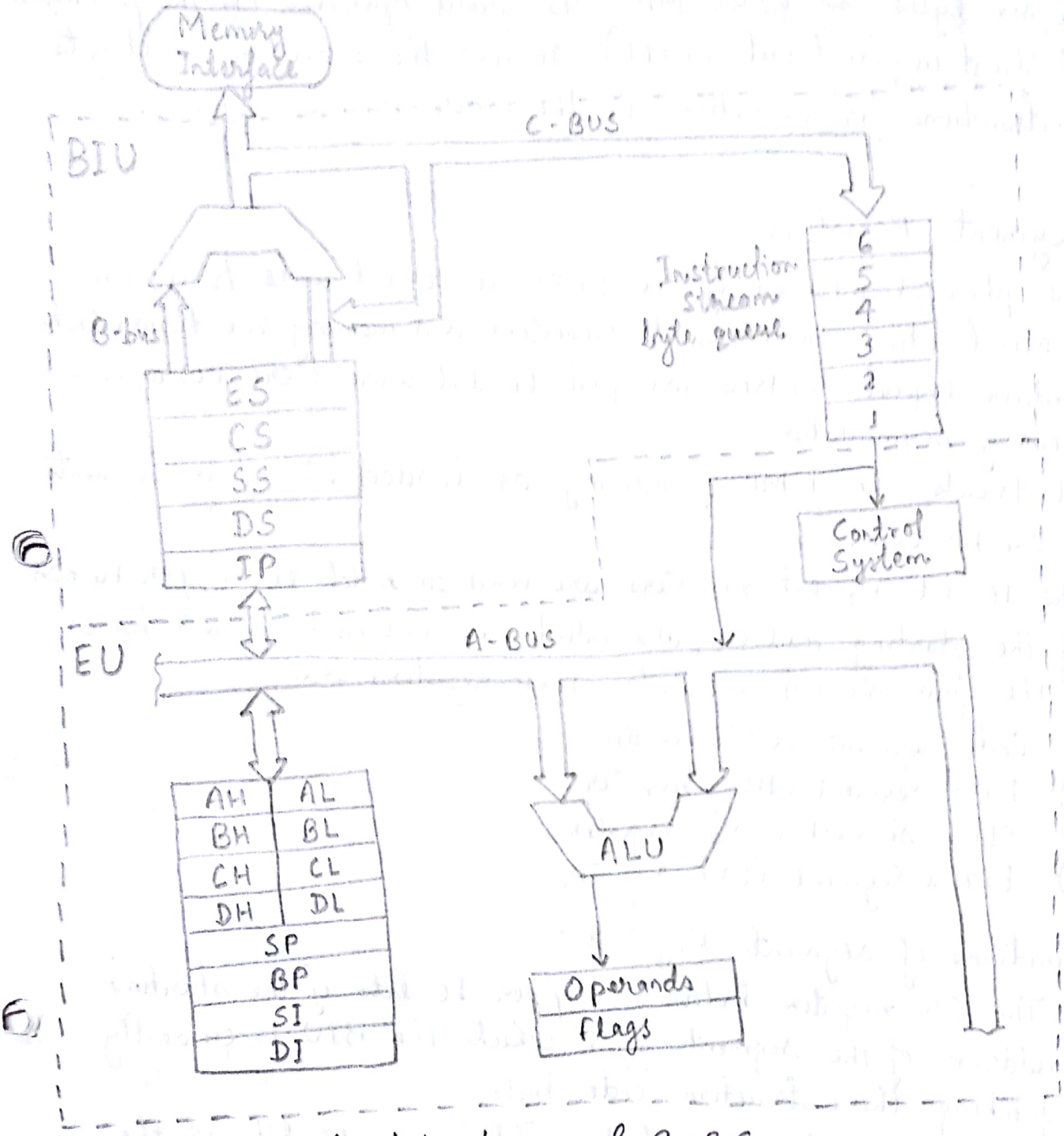
These two functional units can work simultaneously to increase system speed.

BUS INTERFACE UNIT (BIU)

- The bus interface unit is the 8086 interface to the outside world. It provides full 16-bit bidirectional data bus and 20-bit address bus.

Functions of Bus Interface Unit :-

a) It sends address of the memory or I/O.
b) It fetches instruction from memory
c) It reads data from port/memory.
d) It writes data into port/memory
e) It supports instruction queuing
f) It provides address relocation facility.



Architecture of 8086

To implement these functions the BIU contains the instruction queue, segment registers, instruction pointer, address summer and bus control logic.

Instruction Queue - To speed up program execution, the BIU fetches six instruction bytes ahead of time from the memory. These prefetched instruction bytes are held for the execution unit in a group of registers called queue. The size of queue

is six bytes for 8086 MP. The queue operates on the principle of first-in-first-out (FIFO), so that the execution unit gets instruction for execution in the order they are fetched.

2014-15 2013-14 2012-13

Segment Registers

- The physical address of the 8086 is 20-bit wide to access 1 MB location. However, its registers and memory locations which contain logical address are just 16-bit wide. Here 8086 uses memory segmentation.
- It treats the 1 MB of memory as divided into four segments of 64 KB each.
- The 16-bit segment registers are used to hold the upper 16-bits of the starting address (also called base address or segment base) of the four memory segments. These registers are
  a) Code Segment (CS) register
  b) Data Segment (DS) register
  c) Stack segment (SS) register
  d) Extra Segment (ES) register.

Functions of Segment Registers :

1). The CS register holds the upper 16-bits of the starting address of the segment from which the BIU is currently fetching the instruction code byte.

2) The SS register is used to hold upper 16-bits of the starting address for the program stack (all stack related instructions will operate on stack).

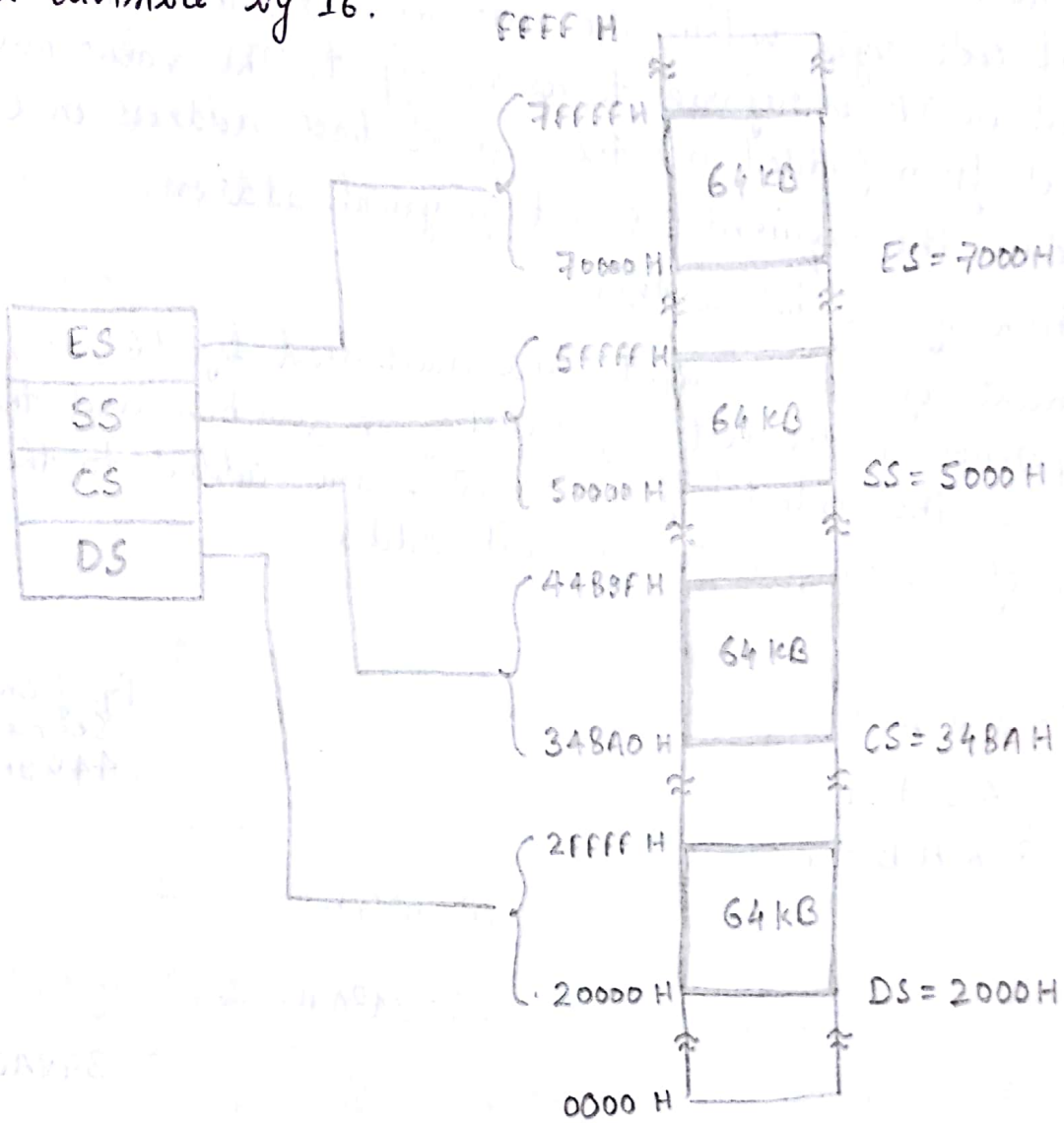
3) The ES register and DS register are used to hold the upper 16-bit of the starting address of the two memory segments which are used for data.

Rules for Memory Segmentation :-

1). The four segments can overlap for small programs. In a

minimum system all four segments can start at address 00000 H.

2.) The segment can begin/start at any memory address which is divisible by 16.



Advantages of Memory Segmentation :-

1.) It allows the memory addressing capacity to be 1 MB even though the address associated with individual instruction is of only 16-bit.

2) It allows instruction code, data, stack and portion of program to be more than 64 kB long by using more than one code, data, stack and extra segment.

3) It facilitates use of separate memory areas for program data and stack.

4) It permits a program or its data to be put in allotted areas of memory, each time the program is executed, i.e., program can be relocated which is very useful in multiprogramming.

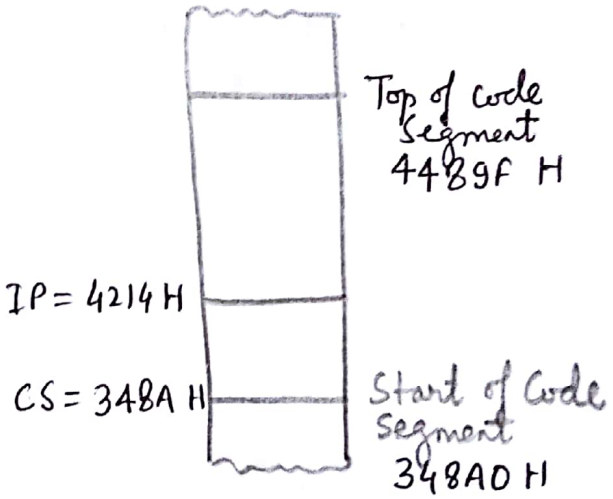
Instruction Pointer :-

The instruction pointer register holds the 16-bit address of the next code byte within the code segment. The value ~~its~~ contained in IP is referred to as an offset. This value must be offset from (added to) the segment base address in CS to produce the required 20-bit physical address.

Generation of 20-bit address :-

- The contents of the CS register are multiplied by 16, i.e., shifted by 4 position to the left by inserting 4 zero bits and then the offset, i.e., the contents of IP register are added to the shifted contents of CS to generate physical address.

CS	3	4	8	A	0
+ IP		4	2	1	4
Physical Address	3	8	A	B	4



Execution Unit (EU)

Execution Unit (EU) works in parallel with BIU which informs BIU about the location at which the next instruction or data is to be fetched. The phases of execution are

a) fetch – fetching of instruction from the queue.
b) decode – decoding of instruction
c) execute – operations are performed on data
d) write – storing the result at destination.

Execution unit consists of :-

1. Control circuitry and Instruction decoders
2. Arithmetic and Logic Unit (ALU)
3. Flag Register
4. General Purpose Registers
5. Stack Point Register
6. Pointers and Index Registers.

1) Control Circuitry and Instruction Decoder :-

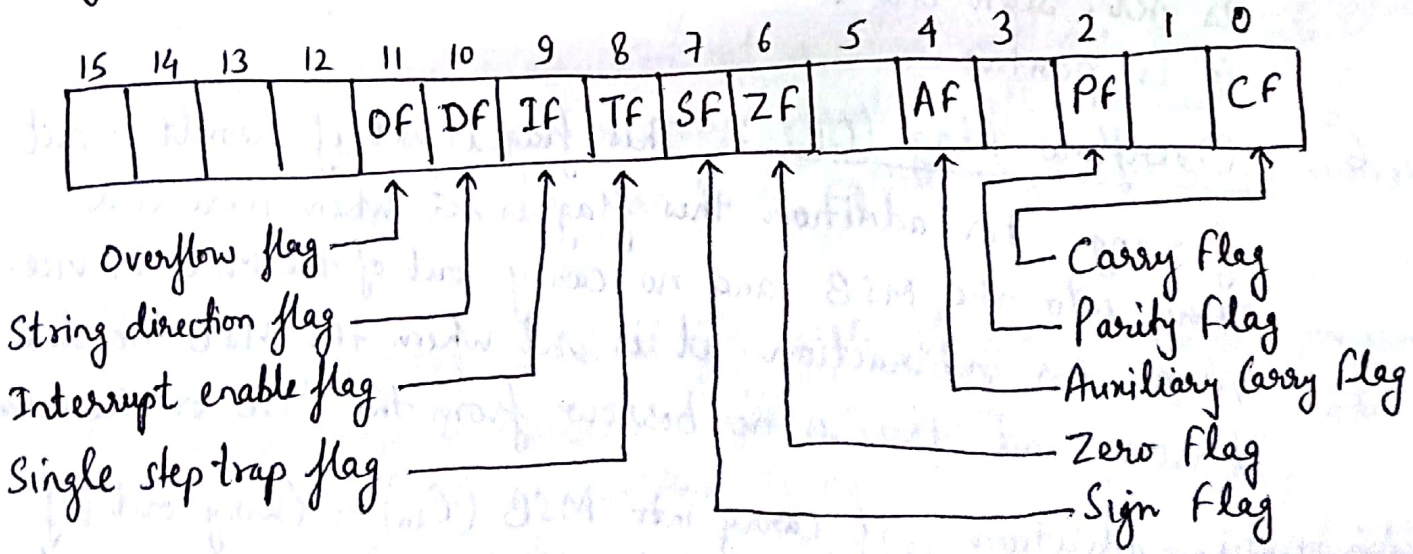
Control circuit of the EU directs all the internal operations of the processor. The instruction in the EU translates the instruction fetched from the memory into a series of actions carried out by the execution unit.

2) Arithmetic and Logic Unit :-

Arithmetic and Logic Unit performs 8-bit or 16-bit mathematical operations such as additions, subtraction, multiplication, division, data conversion and logical operations like NOT, OR or AND. It also performs register increment, decrement & shift operation.

3) Flag Register :-

A flag is a flip-flop which indicates some condition produced by the execution of an instruction or control certain operations of the EU. The flag register contains nine active flags :-



Six flags are used to indicate some condition produced by instruction :-

1. Carry Flag (CF) : In case of addition this flag is set if there is a carry out of the MSB. The carry flag also serves as a borrow flag for subtraction. In case of subtraction it is set when borrow is needed.
2. Parity Flag (PF) : It is set to 1 if result of byte operation or lower byte of the word operation contains an even number of ones ; otherwise it is zero.
3. Auxiliary Carry Flag (AF) : This flag is set ~~if the result of operation~~, if there is an overflow out of bit 3, i.e., carry from lower nibble to higher nibble (D_3-bit to D_4-bit). This flag is used for BCD operation and it is not available to the programmer.
4. Zero Flag (ZF) : The zero flag is set if the result of operation in ALU is zero and flag resets if the result is non-zero.
5. Sign Flag (SF) : After the execution of arithmetic or logical operation, if the MSB result is 1, the sign bit is set. Sign bit 1 indicates result is negative; otherwise it is positive.
6. Overflow Flag (OF) : This flag is set if result is out of range. For addition this flag is set when there is a carry into the MSB and no carry out of the MSB or vice-versa. For subtraction, it is set when the MSB needs a borrow and there is no borrow from the MSB, or vice-versa.

(i) In addition, if carry into MSB (C_{in}) = Carry out of MSB (C_{out}), then the result will be in the range and

OF = 0 ; If $C_{in} \neq C_{out}$, then result is out of range and OF = 1.

eg -

$C_{in} \rightarrow$ 0 1

	+3	0 0 1 1
+	+2	0 0 1 0
	+5	0 0 1 0 1

C_{out} = 0

As $C_{in} = C_{out}$, OF = 0

$C_{in} \rightarrow$ 1 1 1

	+7	0 1 1 1
+	+3	0 0 1 1
	+10	1 0 1 0

C_{out} = 0

As $C_{in} \neq C_{out}$, OF = 1.

(ii) In subtraction, if borrow into the MSB (B_{in}) = Borrow out of the MSB (B_{out}), the result will be within range and OF = 0. If $B_{in} \neq B_{out}$ then OF = 1.

Controlling Flags

The remaining three flags are used to control certain operations of the processor :-

1) Trap Flag (TF) - One way to debug a program is to run the program one instruction at a time and see the contents of used registers and memory variables after execution of every instruction. This process is called single stepping through a program.

Trap flag is used for single stepping through a program. If set, a trap is executed after execution of each instruction, i.e., interrupt service routine is executed which displays various registers and memory variable contents on the display

after execution of each instruction. Thus a programmer can easily trace and correct errors in the program.

2). Interrupt flag (IF) :- It is used to allow/prohibit the interruption of a program. If set, a certain type of interrupt (a maskable interrupt) can be recognized by 8086; otherwise, these interrupts are ignored.

3) Direction flag (DF) : It is used with string instruction. If DF = 0, the string is processed from its beginning with the first element having the lowest address. Otherwise, the string is processed from the high address towards the low address.

Trap flag -(TF) : The method of detecting error in the program is called as software debugging. For software debugging, microprocessor will execute only one instruction at a time. If TF = 0, then microprocessor will execute the complete program in a single operation, but if TF = 1, then microprocessor will execute the program in single stepping mode, i.e., after execution of each instruction, programmer can verify the contents of different registers and flags. In this way the error in the program can be detected.

General Purpose Registers

- The EU has 8 general purpose registers labeled AH, AL, BH, BL, CH, CL, DH and DL. These registers can be used individually for temporary storage of 8-bit data.
- The AL register is also called accumulator.
- Certain pairs of these general purpose registers can be used together to store 16-bit data, such as AX, BX, CX and DX.

Pointers and Index Registers

- All segment registers are 16-bit. But it is necessary to put 20-bit address (physical address) on the address bus. To get 20-bit physical address one more register is associated with each segment register the way IP is associated with CS.
- These additional registers belong to the pointer and index group. The pointer and index group consists of instruction pointer (IP), stack pointer (SP), Base Pointer (BP), Source Index (SI) and Destination Index (DI) registers.

Stack pointer (SP) : The stack pointer (SP) register contains the 16-bit offset from the start of the segment to the top of stack. For stack operation, physical address is produced by adding the contents of Stack Pointer register to the segment base address in SS.

eg.

SS ⇒	4000 0
+ SP ⇒	9F20
Physical address	49F20 H

Here four bits (0 H) is appended to SS = 4000H and then the SP = 9F20H is added to obtain the physical address.

Base Pointer, Source Index and Destination Index

These three 16-bit registers can be used as general purpose registers. However, their main use is to hold 16-bit offset of the data word in one of the segments.

Base Pointer – We can use the BP register instead of SP for accessing the stack using the based addressing mode. In this case, the 20-bit physical stack address is calculated from BP and SS.

Source Index – Source Index (SI) is used to hold the offset of a data word in the data segment. In this case, the 20-bit physical address is calculated from SI and DS.

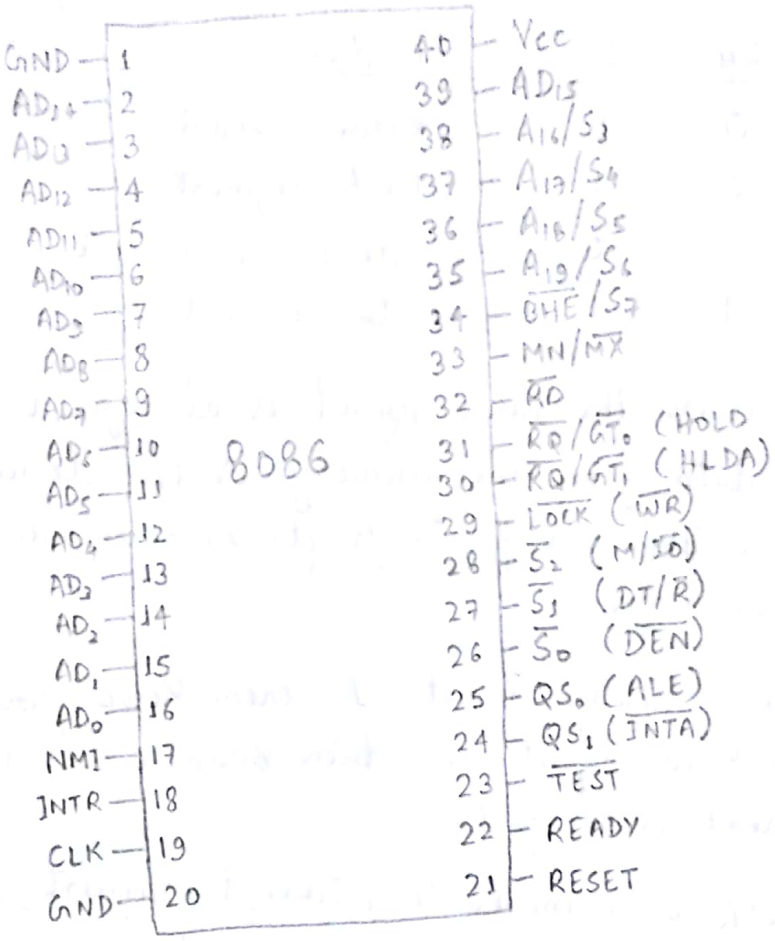
Destination Index (DI) – The ES register points to the extra segment in which data is stored. Storing instruction always uses ES and DI to determine the 20-bit physical address for destination.

CS	—	IP
DS	—	BX, DI, SI
ES	—	DI
SS	—	SP, BP

PIN DIAGRAM OF 8086

2016-17 (5M)

(Minimum mode)



Pin Description

$AD_{15} - AD_0$:- The pin $AD_{15} - AD_0$ are used for data as well as the least significant 16-bit address. The pins are used in the following way -

a) During T_1 state of a machine cycle the microprocessor sends out address on $AD_{15} - AD_0$.

b) During later part of a machine cycle the microprocessor sends out data or receives data on these pins.

$A_{19}/S_6 - A_{16}/S_3$ - The address/status bus are multiplexed to provide address signal $A_{19} - A_{16}$ and also status bits $S_6 - S_3$. These pins also attain a high impedance state during hold acknowledge.

The status bit S_6 always remain at logic 0, bit S_5 indicates the condition of the IF flag bit, and S_4 and S_3 show which segment is accessed during the current bus cycle.

S_4	S_3	Function
0	0	Extra Segment
0	1	Stack Segment
1	0	Code or no segment
1	1	Data segment.

$\overline{RD}$ - Whenever the read signal is at logic 0, the data bus is to read data from the memory or I/O devices connected to the system. This pin floats to its high-impedance state during a hold acknowledge.

READY - When ready input = 1, then 8086 performs normal operation. If ready input = 0, then 8086 enters in a wait state from next clock cycle.

INTR - INTR is a maskable interrupt request, which is level triggered. For INTR held high when IF = 1, the 8086 enters an interrupt acknowledge cycle (INTA becomes active) after the current instruction has completed.

$\overline{TEST}$ - 8086 has a software instruction WAIT. If WAIT instruction is given to 8086, then 8086 will check the $\overline{TEST}$ pin. If $\overline{TEST}$ = 1, then 8086 enters in WAIT state and it will remain in this state till $\overline{TEST}$ become zero.

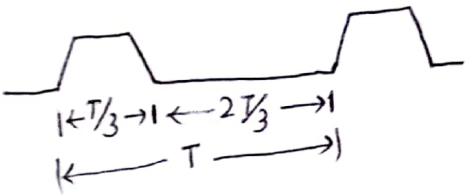
NMI - NMI is a positive going edge triggered, vectored non-maskable interrupt. This cannot be enabled and cannot be disabled using software instructions. It can be used in emergency situations like power supply failure and emergency shut-off etc.

RESET - When a logic 1 is sent into the 8086 on the

on the reset pin, for atleast four clock cycles, the 8086 microprocessor gets reset, i.e., it starts from initial state.

- When the 8086 detects the positive-going edge of a pulse on RESET, it stops all activities until the signal goes low. When RESET becomes low, the 8086 initializes the system as follows:

IP	0000 H
CS	FFFF H
DS	0000 H
SS	0000 H
ES	0000 H
Queue	Empty
Flags	Clear



CLK - The clock pin will provide the basic timing signal to the microprocessor. The clock signal must have a duty cycle of 33% (high for one-third and low for two-third of the clock period) to provide proper internal timing for the 8086.

Vcc - The power supply input provides a +5.0V ± 10% signal to the microprocessor.

GND - The ground supply to the microprocessor.

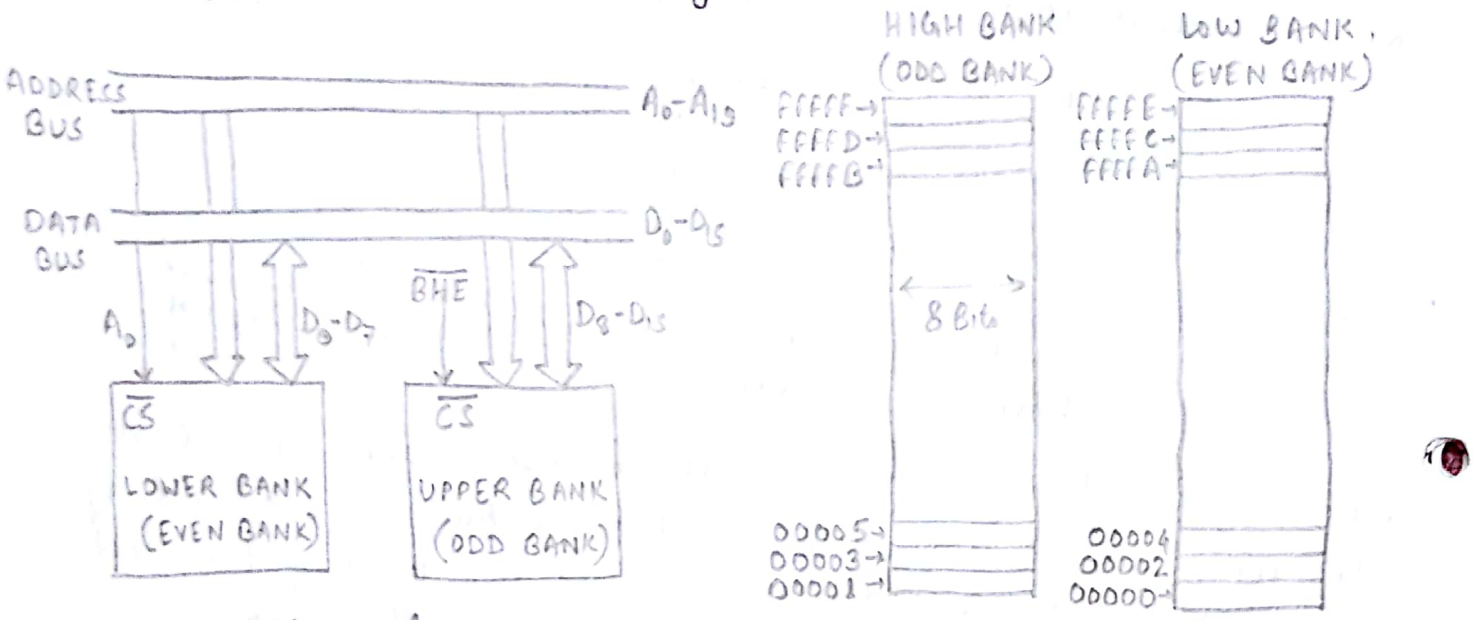
MN/$\overline{MX}$ - When this pin is at +5V, the CPU enters into minimum mode and enters into maximum mode when it is connected to ground.

$\overline{BHE}$/S_7 - The bus high enable pin will help the 8086 to enable the most significant data bus bits (D_{15}-D_8) during a read or a write operation.

⟹ Physically in 8086 based microcomputer, the main memory can be viewed as two banks, each of 512 KB maximum. The banks are called Lower (Even) bank and Upper (odd) bank. To select a location in these banks, a 19-bit address (as 2^{19} = 512K) is sent by the 8086 on A_{19}-A_1. The lower bank contains bytes with only even addresses. The data lines of

the lower bank is connected to $D_0 - D_7$ of 8086. The lower bank is selected by A_0.

The upper bank contains bytes with only odd addresses. The data lines of the upper bank is connected to $D_8 - D_{15}$ of 8086. The upper bank is selected by $\overline{BHE}$.



Physical memory system of 8086 & its accessing.

2014-15

Minimum Mode Pins

M/$\overline{IO}$ - The M/$\overline{IO}$ selects memory or I/O, which indicates that the microprocessor address bus contains either a memory address or an I/O port address.

$\overline{WR}$ - It indicates that 8086 is outputting data to a memory or I/O device. At the time when $\overline{WR}$ is at logic 0, the data bus contains valid data for memory or I/O.

$\overline{INTA}$ - The interrupt acknowledge signal is a response to the INTR input pin. The $\overline{INTA}$ pin is normally used to gate the interrupt vector number on to the data bus in response to an interrupt request.

ALE - Address latch enable shows that the 8086 address/

data bus contains address information, where address can be a memory address or I/O port number.

DT/$\overline{R}$ – The data transmit/receive signal shows that the microprocessor, data is transmitting (DT/$\overline{R}$ = 1) or receiving (DT/$\overline{R}$ = 0) data. This signal is used to enable external data bus buffers.

$\overline{DEN}$ – Whenever 8086 performs data transfer, i.e., either reading or writing operation, then 8086 gives $\overline{DEN}$ = 0. This signal is used to enable bi-directional tri-state buffer during DMA operation.

HOLD – Hold request
HLDA – Hold acknowledge

These two pins are used during direct memory access. The 'HOLD' is used whenever DMA operation is required by some I/O device. The microprocessor sends acknowledge signal on the pin HLDA.

2014-15

Maximum Mode Pins

To acheive maximum mode for use with external coprocessor, the MN/$\overline{MX}$ pin must be connected to ground.

$\overline{S_2}$, $\overline{S_1}$ & $\overline{S_0}$ – State bits indicate the function of the current bus cycle.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Function
0	0	0	Interrupt acknowledge
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. (not defined)

$\overline{RQ}/\overline{GT_1}$ & $\overline{RQ}/\overline{GT_0}$ – The request/grant pins request direct memory access (DMA) during maximum mode operation. Both these lines are bi-directional and are needed to request and grant a DMA operation.

$\overline{LOCK}$ – The lock output is used to lock peripherals off the system. ~~This pin is activated by using the clock~~ $\overline{LOCK}$ prefix for an instruction allows a microprocessor to make sure that another microprocessor does not take control of the system bus during execution of a critical instruction which uses the system bus.

QS_1 and QS_0 – The queue state bits show the states of the internal instruction queue.

QS_1	QS_0	Function
0	0	Queue is idle (no operation)
0	1	First byte of opcodes
1	0	Queue is empty
1	1	Subsequent byte of opcode.