Gramin Shikshan Prasarak Mandal's

GRAMIN POLYTECHNIC

(ISO 9001:2008 Certified Institute)

Chapter 1- 8086-16 bit Microprocessor

By

Mr. Shinde G. B.

M.Tech. (Electronics Engineering)

Syllabus : **Total Marks:14**

1.1 8086 microprocessor:sailent features, pin description

1.2 Architecture of 8086:functional block diagram, register organisation.

1.3 Concept of pipelining

1.4 Memory segmentation, physical memory address generation

Unit Outcomes :

- Describe the functions of the given pin of 8086.
- Explain with sketches the working of given unit of 8086 microprocessor.
- State the functions of given registers of 8086 microprocessors
- Calculate physical address of given segmentation of 8086 microprocessor.

Block diagram of 8086 :

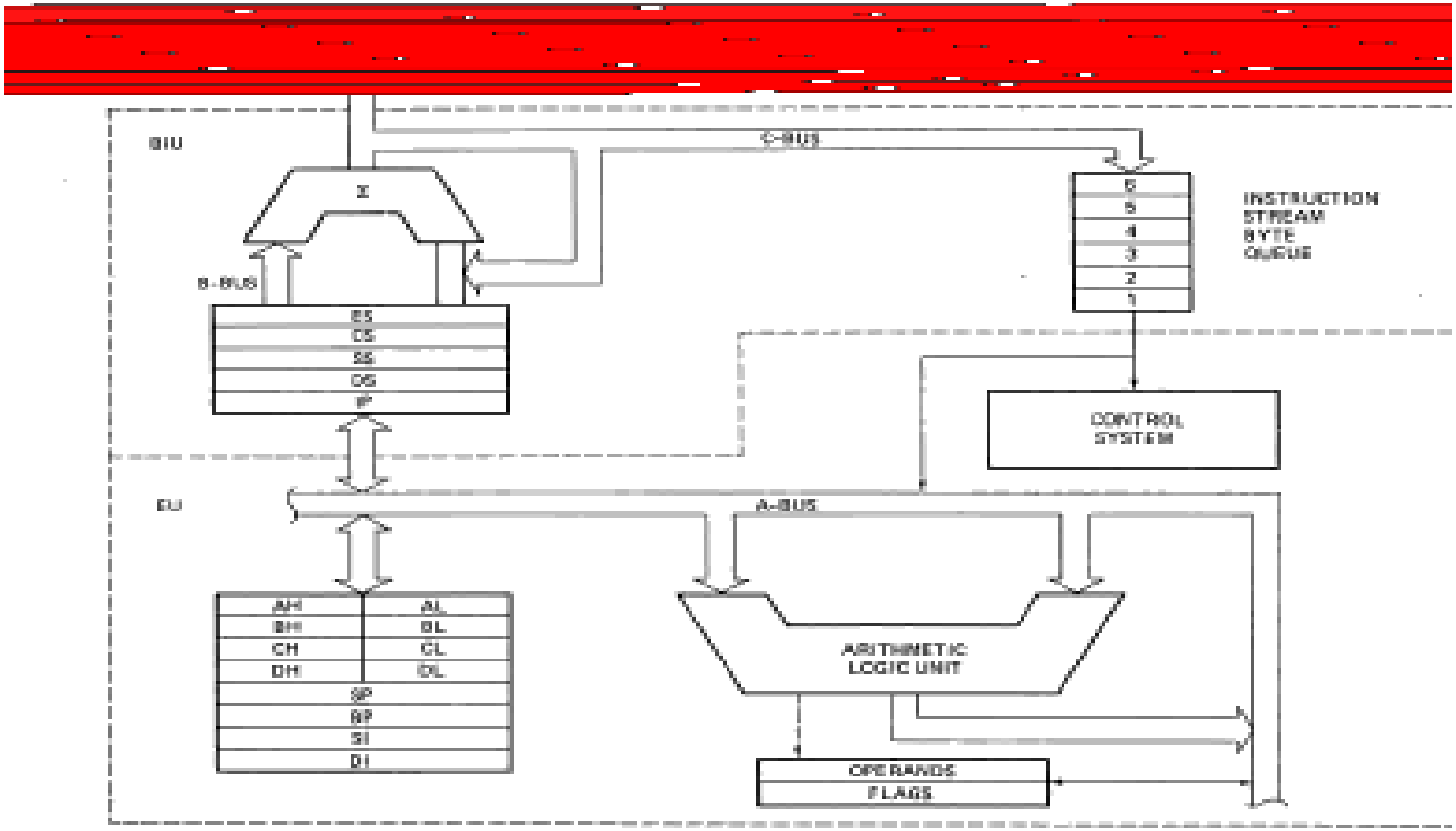


FIGURE 2-7 8086 internal block diagram. (*Intel Corp.*)

8086 Programmer's Model

BIU registers
(20 bit adder)

ES	Extra Segment
CS	Code Segment
SS	Stack Segment
DS	Data Segment
IP	Instruction Pointer

EU registers

AX	AH	AL	Accumulator
BX	BH	BL	Base Register
CX	CH	CL	Count Register
DX	DH	DL	Data Register
	SP		Stack Pointer
	BP		Base Pointer
	SI		Source Index Register
	DI		Destination Index Register
	FLAGS		

Contd…

8086 CPU is divided into two independent functional parts.

1.Execution unit(EU).

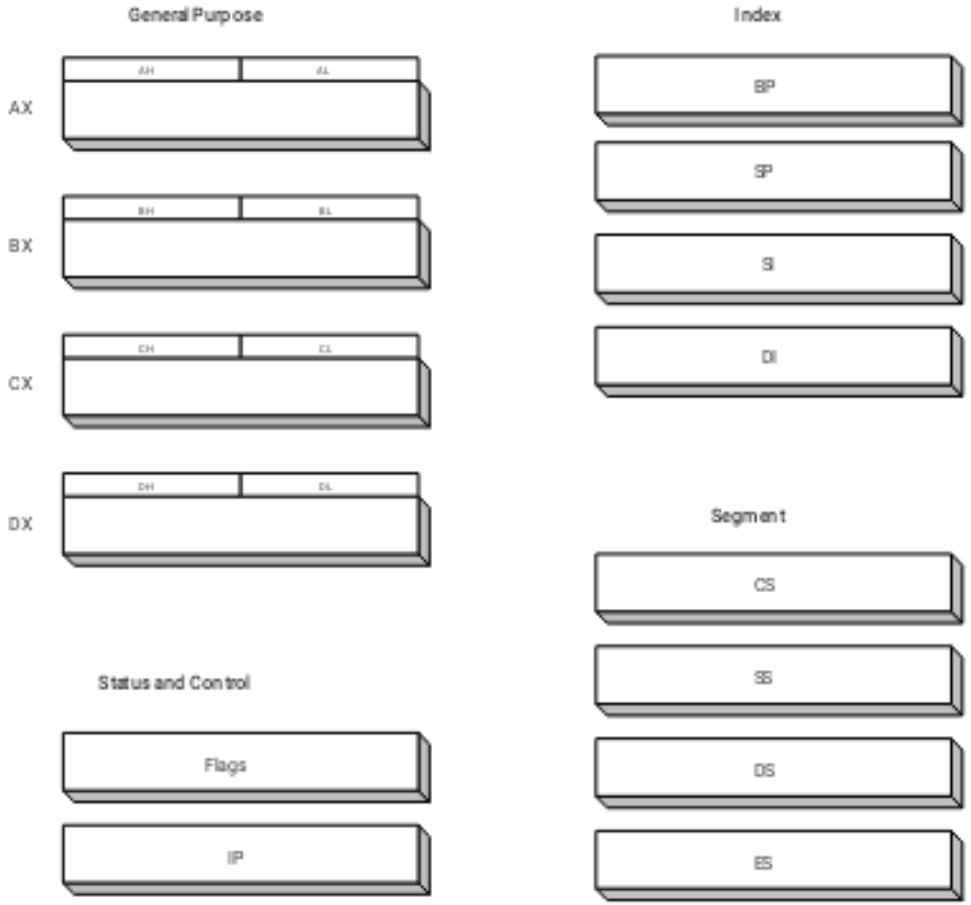
2.Bus interface unit(BIU).

Contd…

The function of execution unit are
To fetch the instruction.
To decode the instruction.
To execute the instruction.

EU also contains the control circuitry to perform various internal operations.
The decoder in EU decodes the instruction which is fetched from memory.
EU has 16 bit ALU, which can perform arithmetic and logical operation on 8 bit as well as 16 bit
EU also contains the flag register which of size16 bit

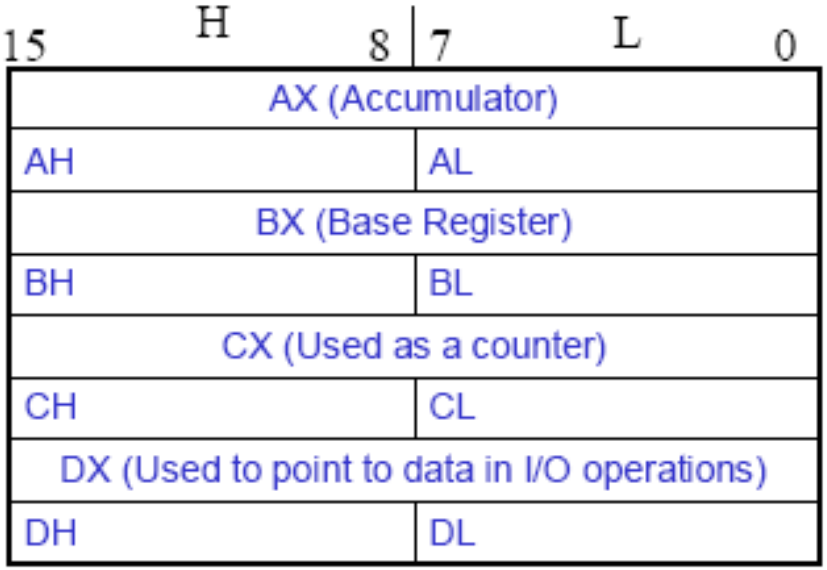
8086 REGISTERS



Contd…

Flag register contains 9 active flags.
5 flags in the lower byte are condition or status flags including OF flag.
The machine control flags are DF,IF and TF.

GENERAL PURPOSE REGISTERS



AX - the Accumulator
BX - the Base Register
CX - the Count Register
DX - the Data Register

Normally used for storing temporary results
Each of the registers is 16 bits wide **(AX, BX, CX, DX)**
Can be accessed as either 16 or 8 bits AX, AH, AL

General Purpose Registers

- **AX**
  - Accumulator Register
  - Preferred register to use in arithmetic, logic and data transfer instructions because it generates the shortest Machine Language Code
  - Must be used in multiplication and division operations
  - Must also be used in I/O operations

- **BX**
  - Base Register
  - Also serves as an address register

Contd…

- **CX**
  - Count register
  - Used as a loop counter
  - Used in shift and rotate operations

- **DX**
  - Data register
  - Used in multiplication and division
  - Also used in I/O operations

POINTER AND INDEX REGISTERS

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

- All 16 bits wide, L/H bytes are not accessible
- Used as memory pointers
  - Example: MOV AH, [SI]
    - *Move the byte stored in memory location whose address is contained in register SI to register AH*
- IP is not under direct control of the programmer

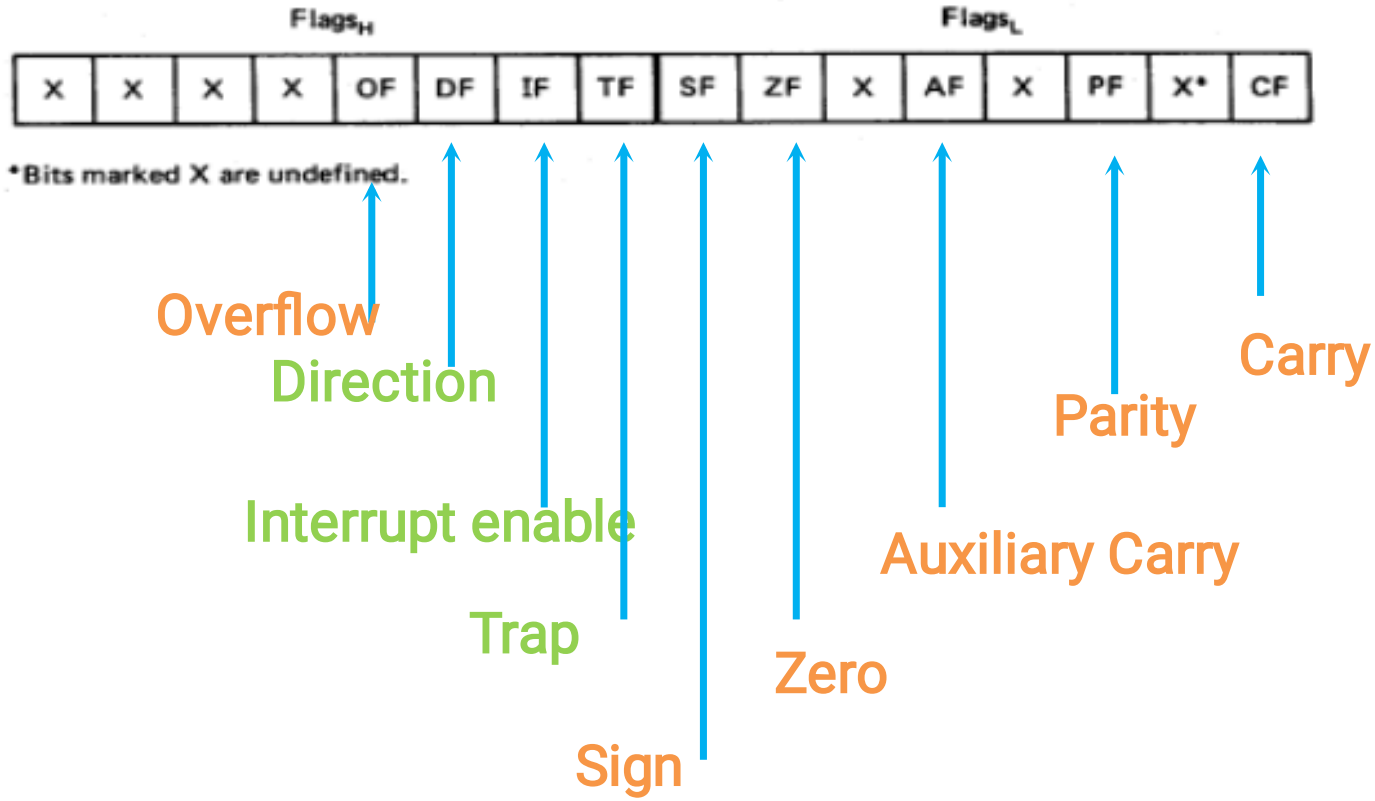
Contd…

- SP and BP are pointer registers which holds the 16 bit offset address within the particular segment

- SI and DI are the index registers.

- During the execution of string related intructions SI is used to store the offset of source data in data segment and DI is used to store the offset of destination data in data or extra segment.

FLAG REGISTER



6 are status flags
3 are control flag

Contd…

•Trap flag:
When trap flag is set the program can be run in single step mode.
It allows the user to execute one instruction of a program at a time for debugging.
•IF flag:
It is a interrupt enable/disable flag.
It is set if interrupt is enabled(INTR)
•Direction flag:
This flag is used in string operation.
If it is set string byte are read or write from higher memory address to lower memory address.

BUS INTERFACE UNIT(BIU)

The main function of the BIU is to send the address to:

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

BIU consist the various sections

- Segment registers.
- Instruction queue.

Contd…

Segment registers:
BIU has 4 segment registers which of having the size 16 bit that are CS,DS,SS and ES.

The memory pointers are used to point or address particular memory location in memory.

CS: the code segment register is used to address a memory location in the code segment of the memory where the opcode of the program is stored.

Contd…

•DS:The data segment DS register points to the data segment of the memory where the data has been stored.

•ES:The extra segment ES register is used to address the segment which is additional data segment used to store the data.

•SS:The stack segment SS register is used to point the stack location in the stack segment of the memory and used to store the temporarily on the stack such as content of CPU registers or intermediate results.

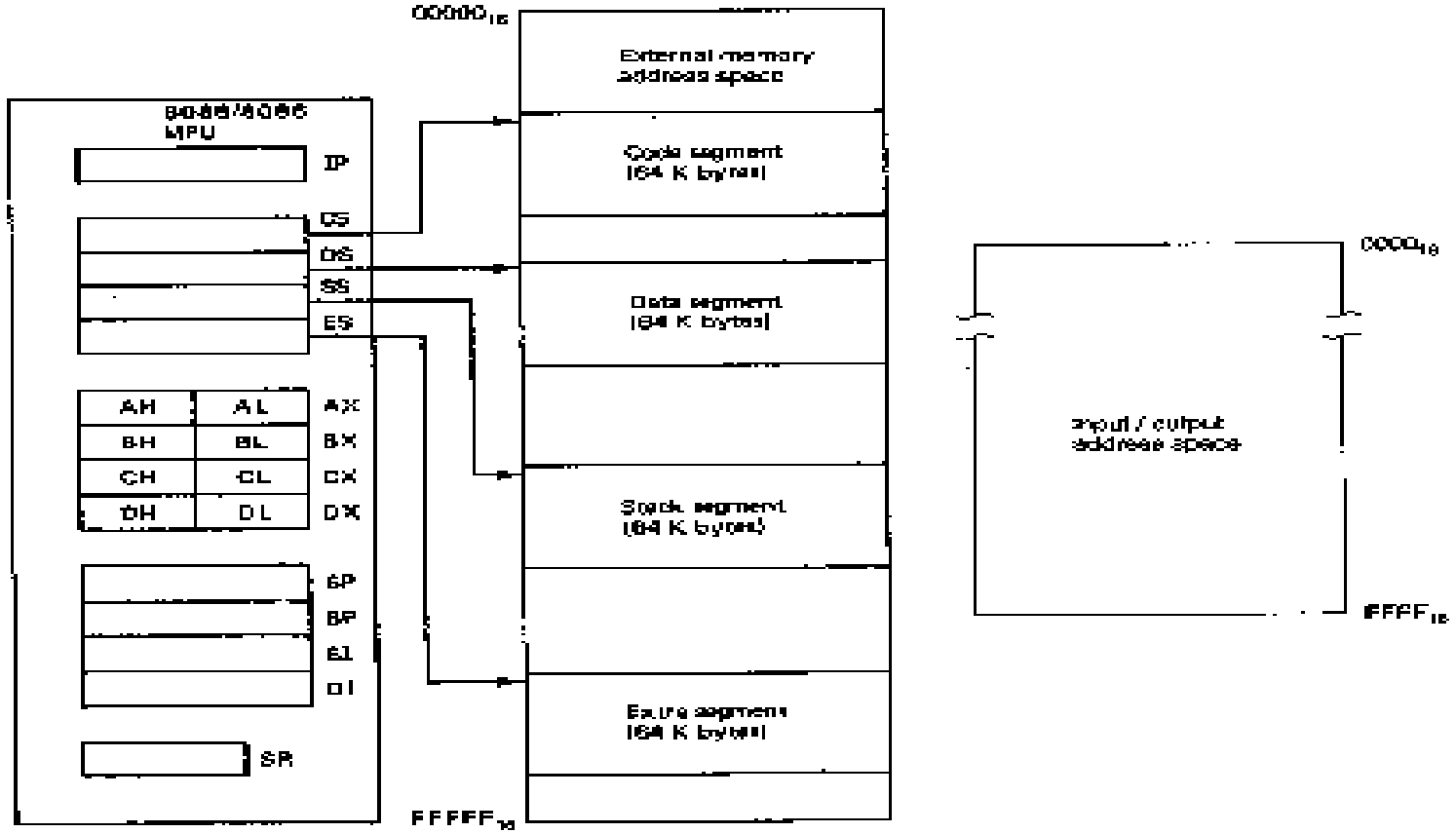
Instruction Queue(IQ):
To increase the execution speed BIU fetched as many as 6 instruction bytes ahead to time from memory.

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

Then all bytes are given to the EU one by one.

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

SOFTWARE MODEL OF THE 8086 MICROPROCESSORS



The Stack

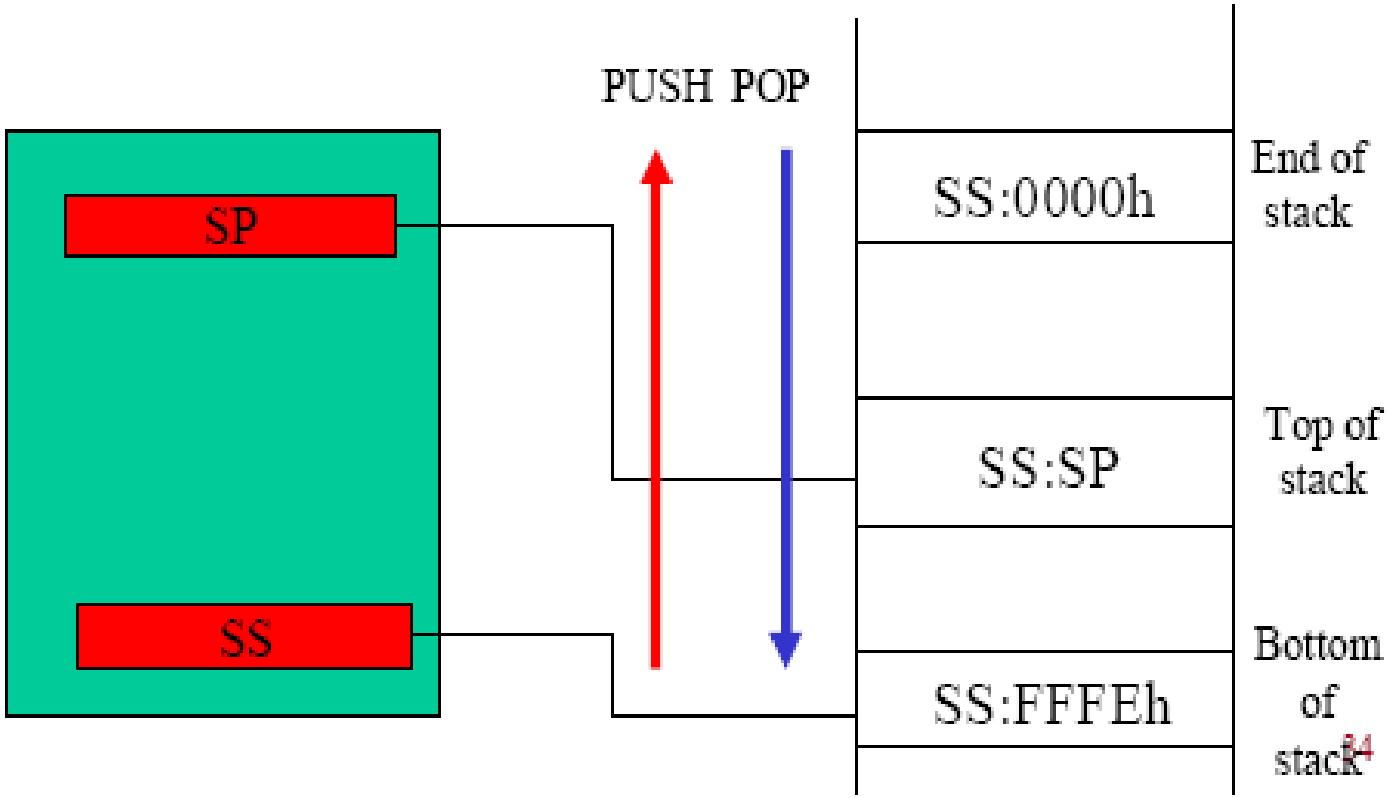
- The stack is used for temporary storage of information such as data or addresses.

- When a CALL is executed, the 8086 automatically PUSHes the current value of CS and IP onto the stack.

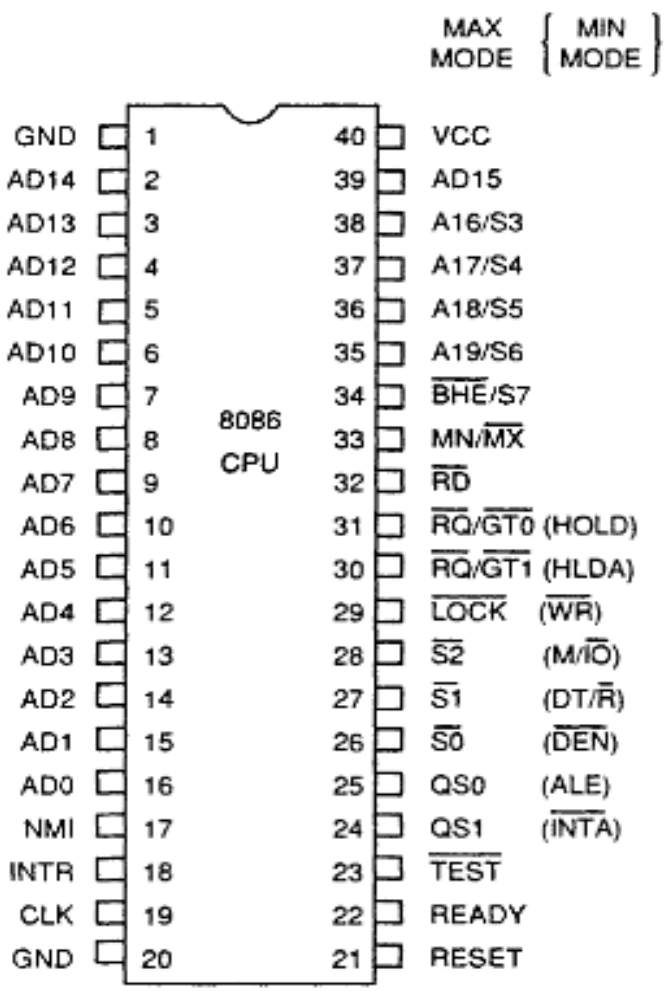
Other registers can also be pushed

Before return from the subroutine, POP instructions can be used to pop values back from the stack into the corresponding registers.

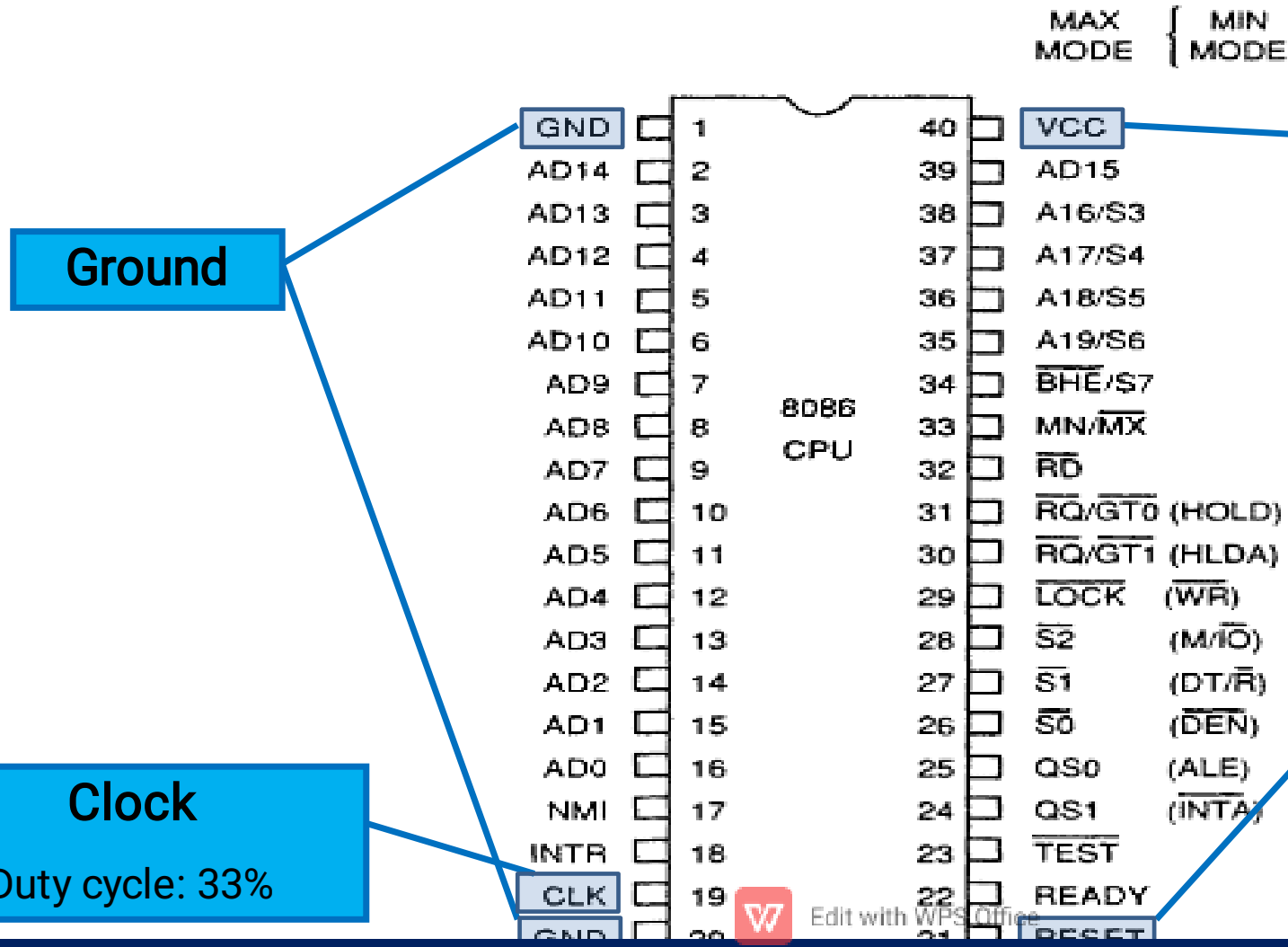
Contd…


PUSH POP
SP
SS
SS:0000h
SS:SP
SS:FFFEh
End of stack
Top of stack
Bottom of stack

INTEL 8086 - Pin Diagram



Contd...



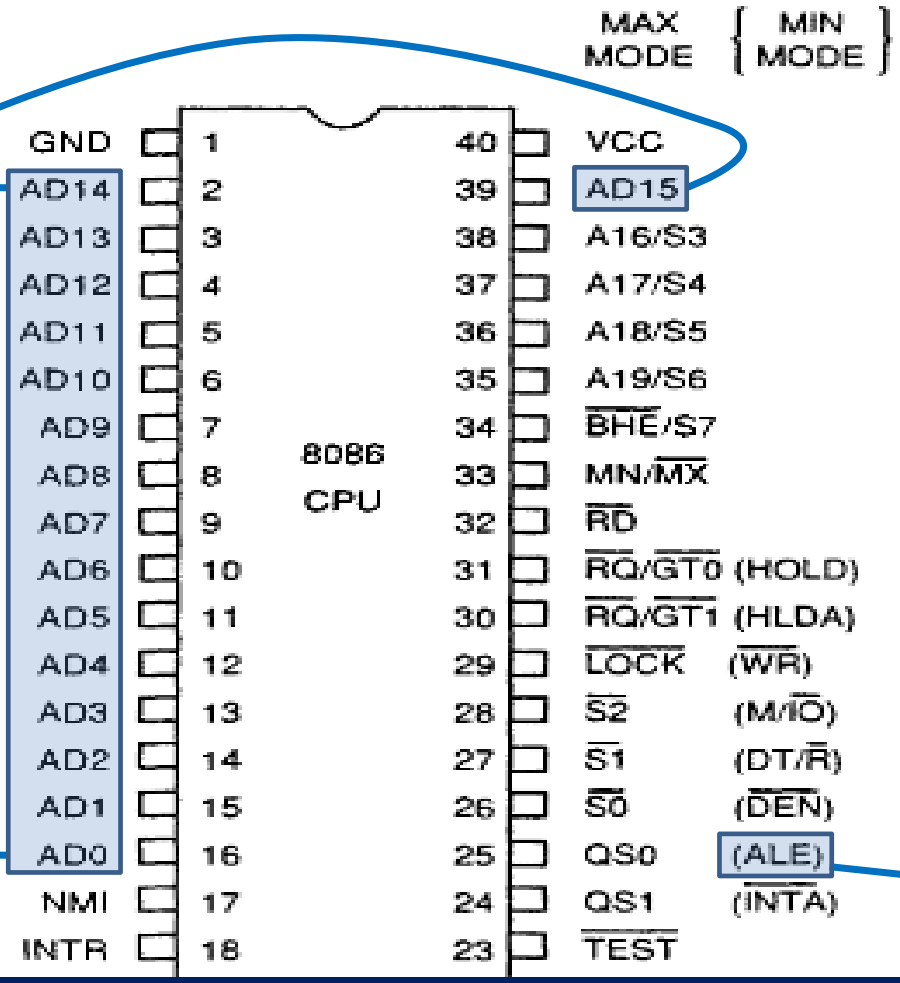
Contd…

RESET:

- It is system reset.
- When this signal goes high processor enters into reset state and terminate the current activity and start the execution from the FFFF0H.
- This signal is a active high signal.

Contd...

INTEL 8086 - Pin Details



Address/Data Bus:

Contains address bits A_{15}-A_0 when ALE is 1 & data bits D_{15} – D_0 when ALE is 0.

Address Latch Enable:

When high, multiplexed address/data bus contains address information.

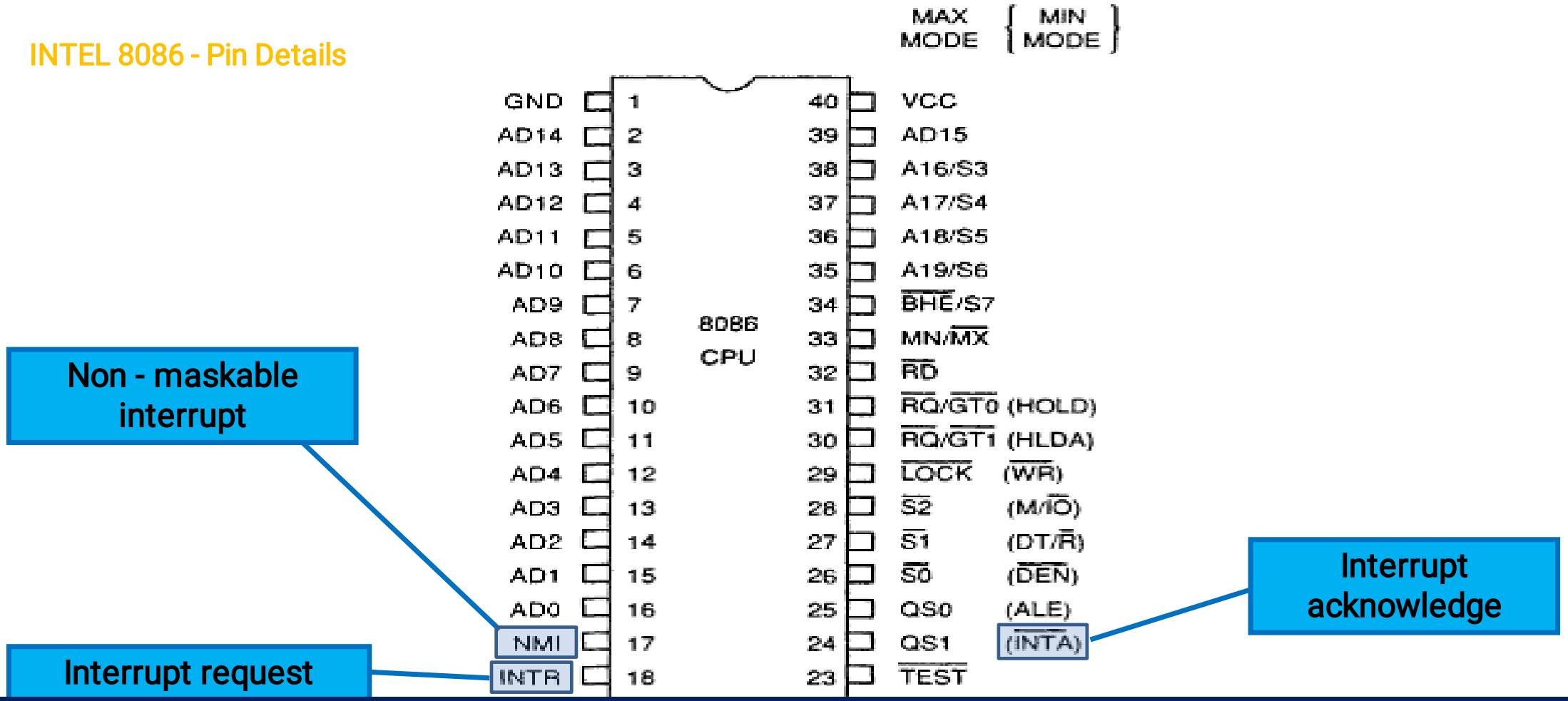
Contd...

AD0-AD15:

- These lines are time multiplexed bidirectional address/data bus.
- During T1 clock cycle of bus cycle these lines carry lower order 16 bit address.
- During T2,T3,T4 they carry data.
- So AD0 to AD7 line carry lower order byte of data and AD8 to AD15 carry higher order byte of data.

Contd...

INTEL 8086 - Pin Details



Contd...

INTR:

- It is a interrupt request.
- This is a level triggered interrupt request input and checked during the last clock cycle of each instruction to determine the availability of request.
- If any interrupt request is occurred, processor serves this interrupt.

NMI:

It is a non maskable interrupt that can not be ignored.

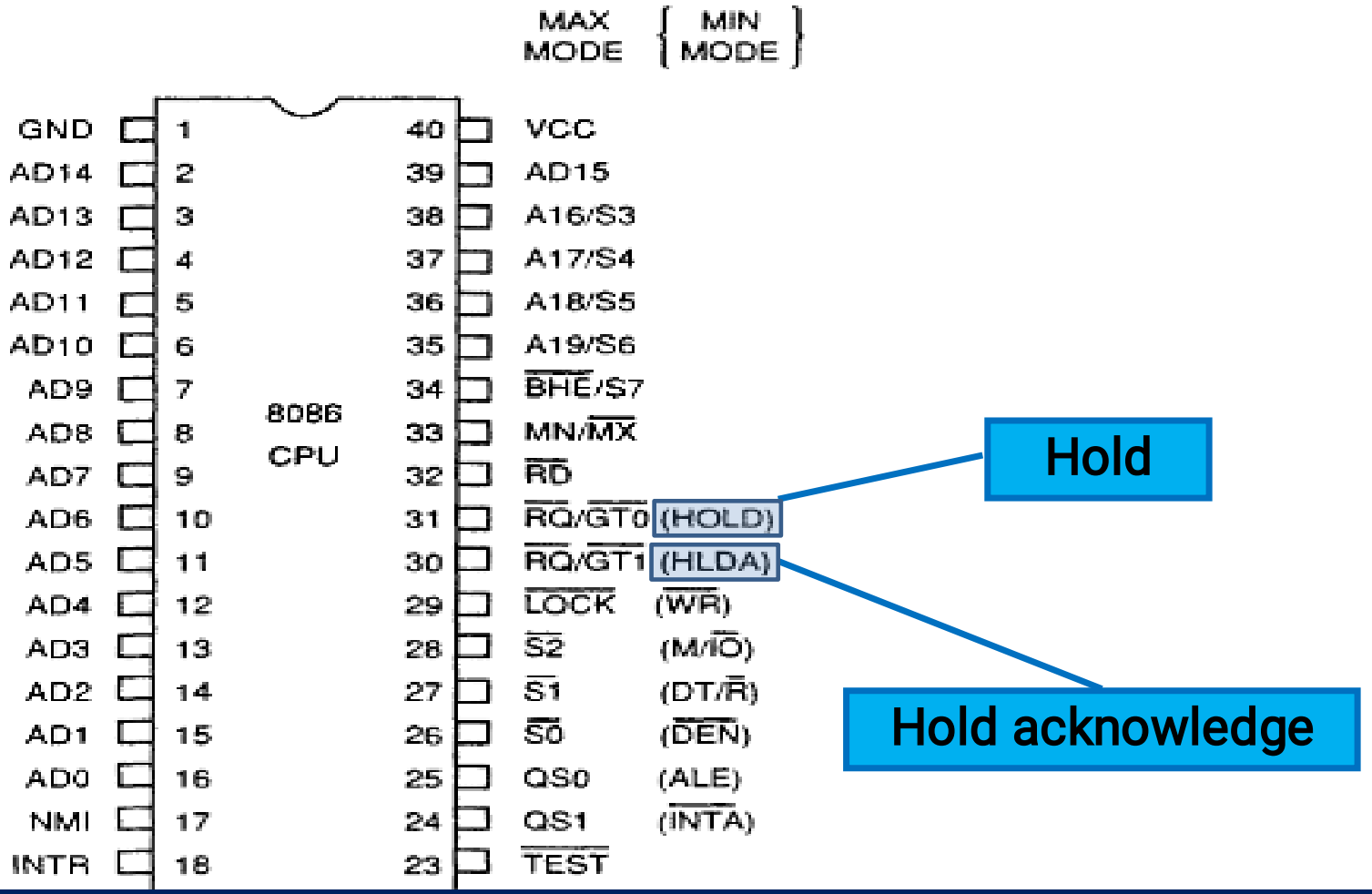
INTA:

It is a active low acknowledge signal.

When processor receives INTR signal it acknowledge the interrupt.

Contd...

INTEL 8086 - Pin Details



Contd...

HOLD:

When another master device Such as DMA controller needs the use of address bus data bus and control buses it sends a HOLD request to the processor through this line.

It is a active high input signal.

HLDA:

This is a active high output signal generated by the processor after receiving the HOLD signal.

Contd...

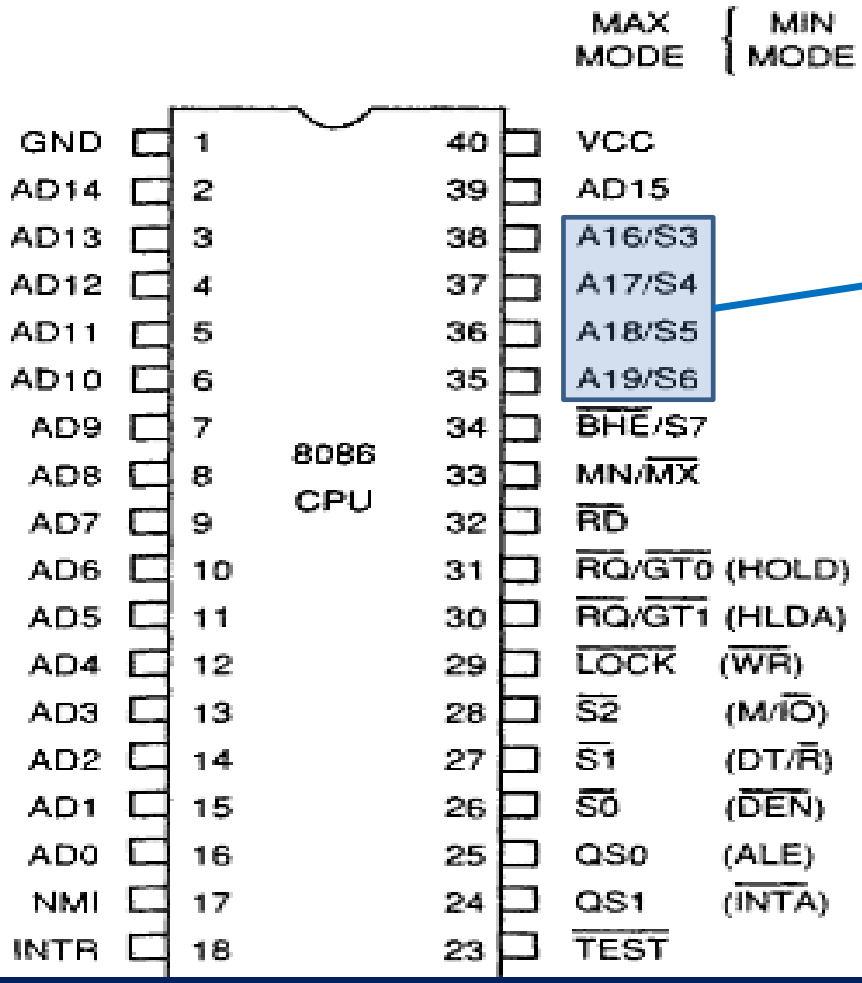
INTEL 8086 - Pin Details

S6: Logic 0.

S5: Indicates condition of IF flag bits.

S4-S3: Indicate which segment is accessed during current bus cycle:

S4	S3	*Function*
0	0	Extra segment
0	1	Stack segment
1	0	Code or no segment
1	1	Data segment



Address/Status Bus

Address bits A_{19} – A_{16} & Status bits S_6 – S_3

Contd...

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

- These are time sharing multiplexed address and status signal.
- During T1 clock cycle these line carry upper 4 bit address and during I/O these lines are low.

S3 and S4:

- S3 and S4 carry status signal and these status lines are used to identify memory segments.

S5:

It is a interrupt enable status signal i.e. condition of IF flag

Contd...

INTEL 8086 - Pin Details

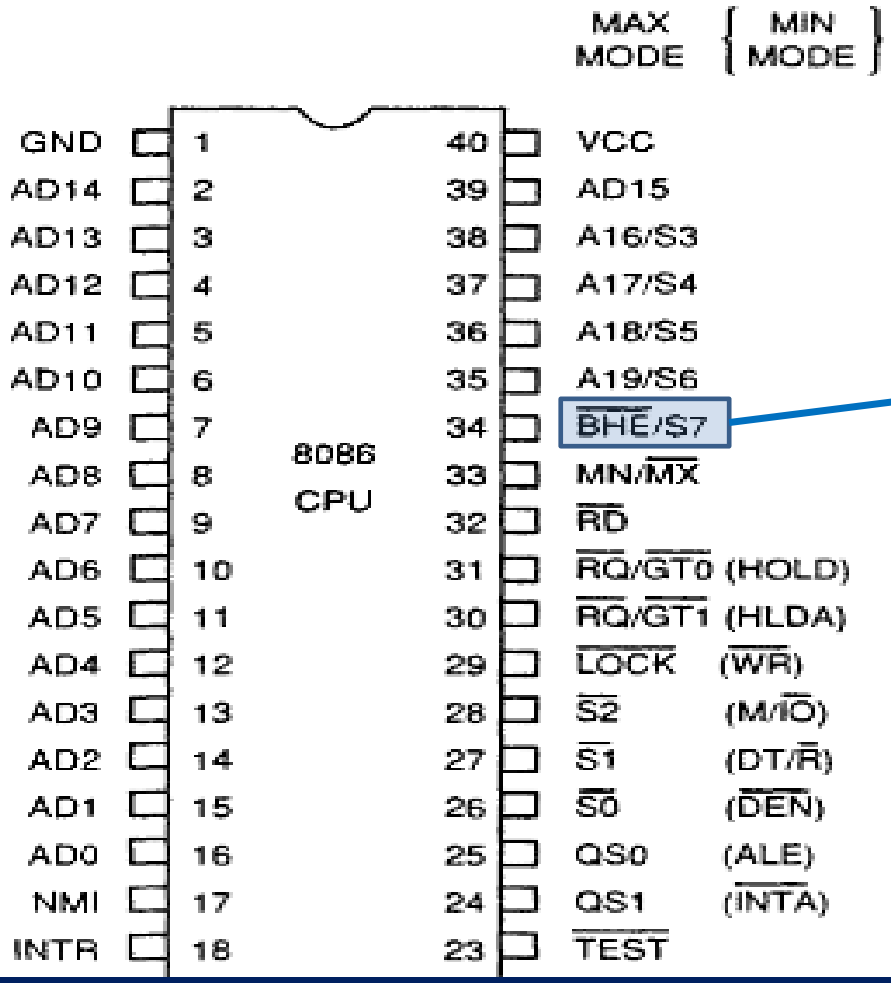
BHE#, A_0:

0,0: Whole word (16-bits)

0,1: High byte to/from odd address

1,0: Low byte to/from even address

1,1: No selection



Bus High Enable/S7

Enables most significant data bits D_{15} – D_8 during read or write operation.

S_7: Always 1.

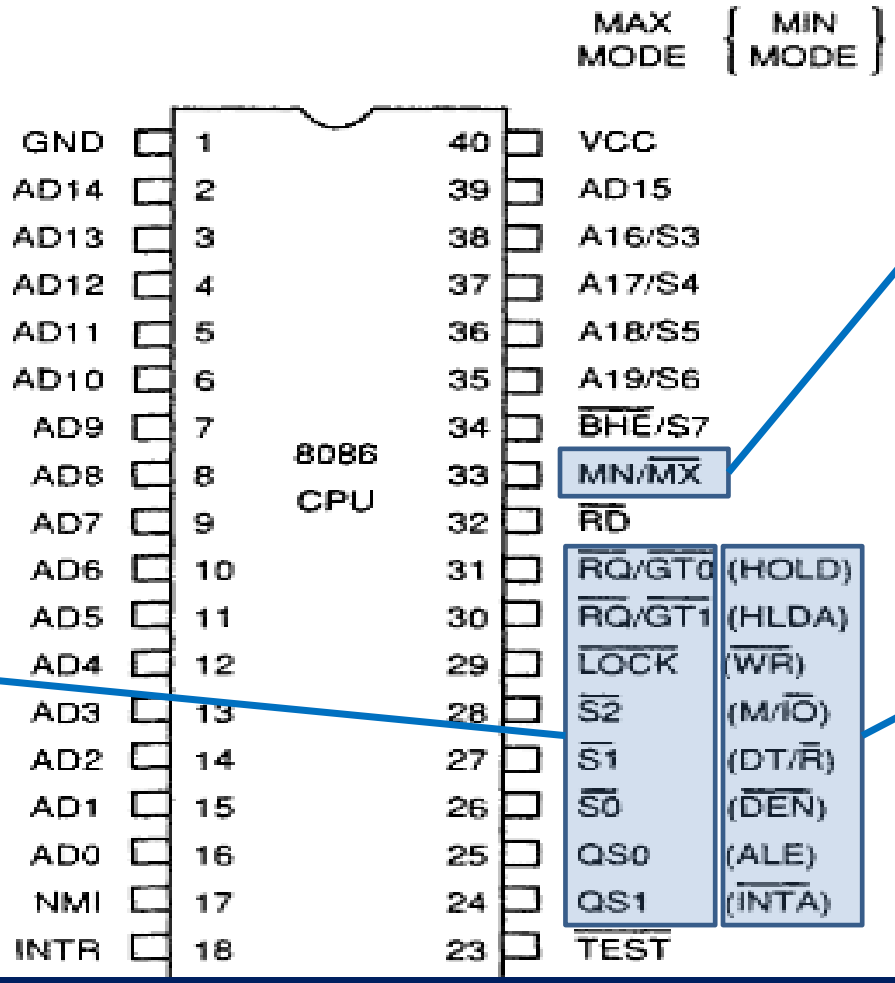
Contd...

BHE/S7:

- Bus high enable signal is used to indicate the transfer of data over higher order(D15-D8) data bus
- It goes low for the data transfer over D8-D15.
- BHE in combine with A0 determines whether a byte or word will be transferred from / to memory location.

Contd...

INTEL 8086 - Pin Details



Min/Max mode

Minimum Mode: +5V

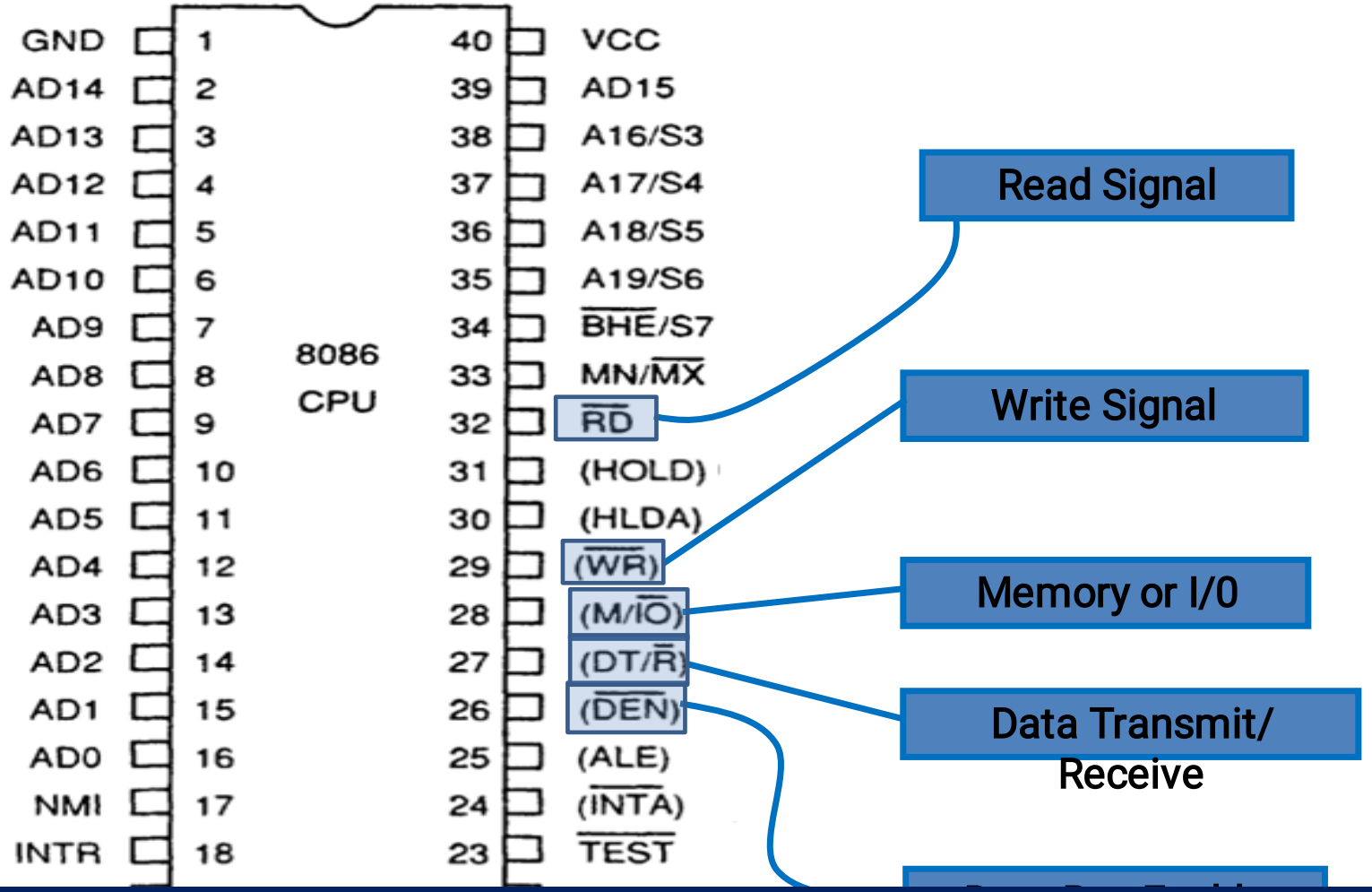
Maximum Mode: 0V

Minimum Mode Pins

Maximum Mode Pins

Contd...

Minimum Mode- Pin Details



Contd…

RD(read):

- It is an active low read signal generated by the processor to indicate that the processor is performing read operation with memory or I/O depending on the status of M/IO signal.

WR(write):

- It is the active low signal issued by the processor to write data to memory or I/O device depending on the status of M/IO signal.

M/IO(status signal)

- This signal is issued by the processor to distinguish memory access from an I/O access.

Contd…

- when this signal is high memory is accessed and when this signal is low an I/O device is accessed.

DT/R(data transmit/receive):

- This output signal is used to decide the direction of data flow.
- When the processor sends data out this signal is high and when the processor receives data then this signal is low.

DEN(data enable):

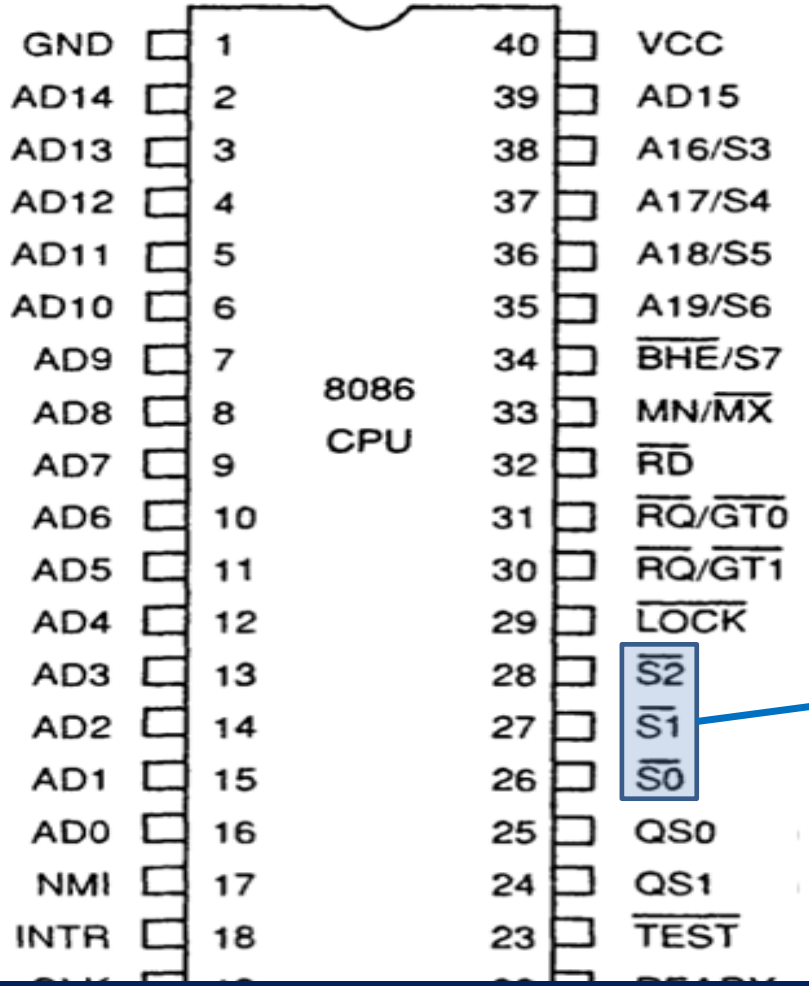
- It is an active low signal issued by the processor to indicate the availability of valid data over AD0-AD15.

Contd...

Maximum Mode - Pin Details

S2 S1 S0

000: INTA
001: read I/O port
010: write I/O port
011: halt
100: code access
101: read memory
110: write memory
111: none -passive



Status Signal

Inputs to 8288 to generate eliminated signals due to max mode.

Contd...

S0 S1 S2 :

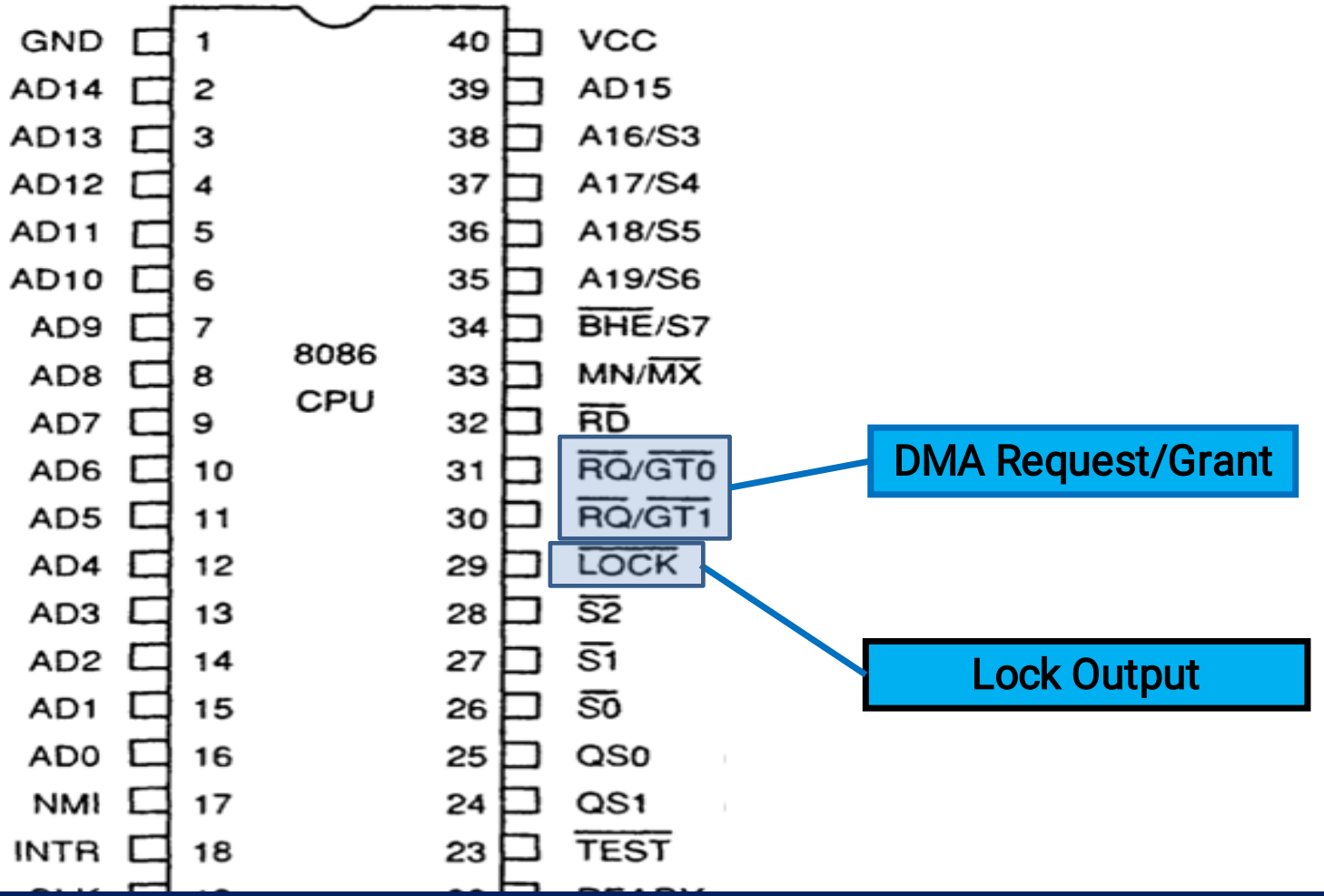
- These status signal shows the type of operation being carried out by the processor

Maximum Mode - Pin Details

Lock Output

Used to lock peripherals off the system

Activated by using the LOCK: prefix on any instruction



Contd...

LOCK:

- This is an active low output signal used to prevent the other system bus master from gaining the system buses.
- It is a active low signal.
- When it goes low all interrupts are masked and HOLD request is not granted

RQ/GT0,RQ/GT1(request/grant):

- These pins are used by the other local bus master in maximum mode to gain the control of local buses at the end of processors current bus cycle
- After receiving the request on these lines cpu sends the acknowledge signal on the same lines.

Contd...

Maximum Mode - Pin Details

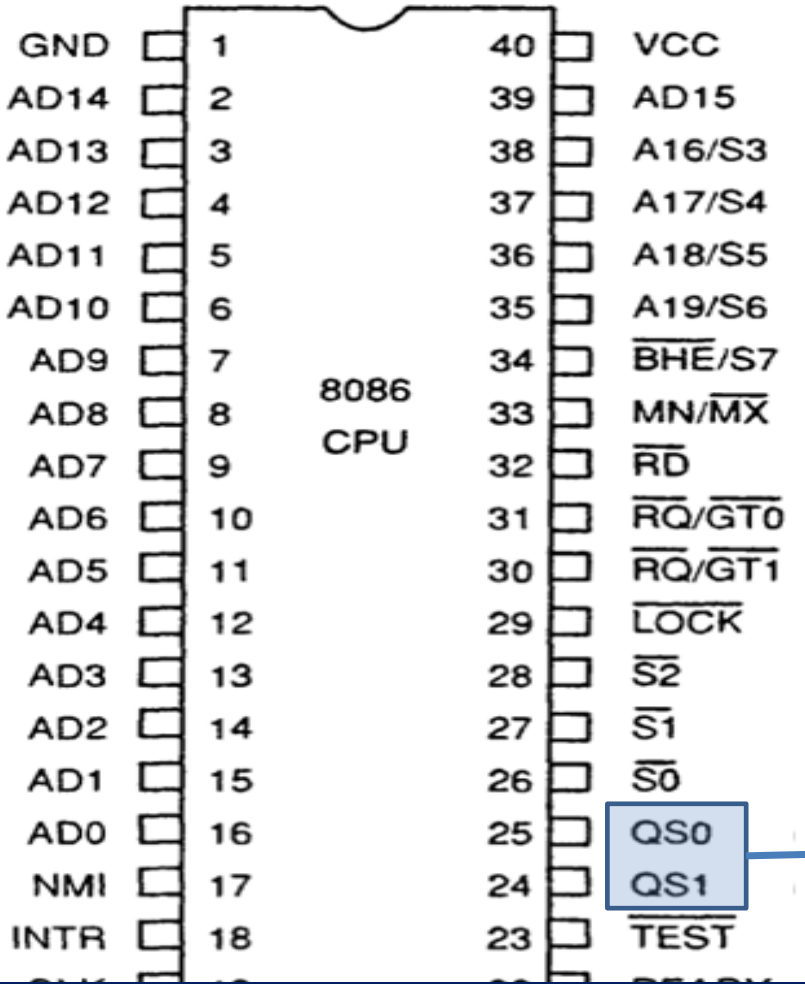
QS1 QS0

00: Queue is idle

01: First byte of opcode

10: Queue is empty

11: Subsequent byte of opcode



Queue Status

Used by numeric coprocessor (8087)

Contd…

QS0 QS1:

- These lines provides the information about the statues of instruction queue during the clock cycle.

TEST:

- This signal is used to test the status of the math co-processor.(8087)
- The busy pin of 8087 is connected to this pin of 8086.

Contd...

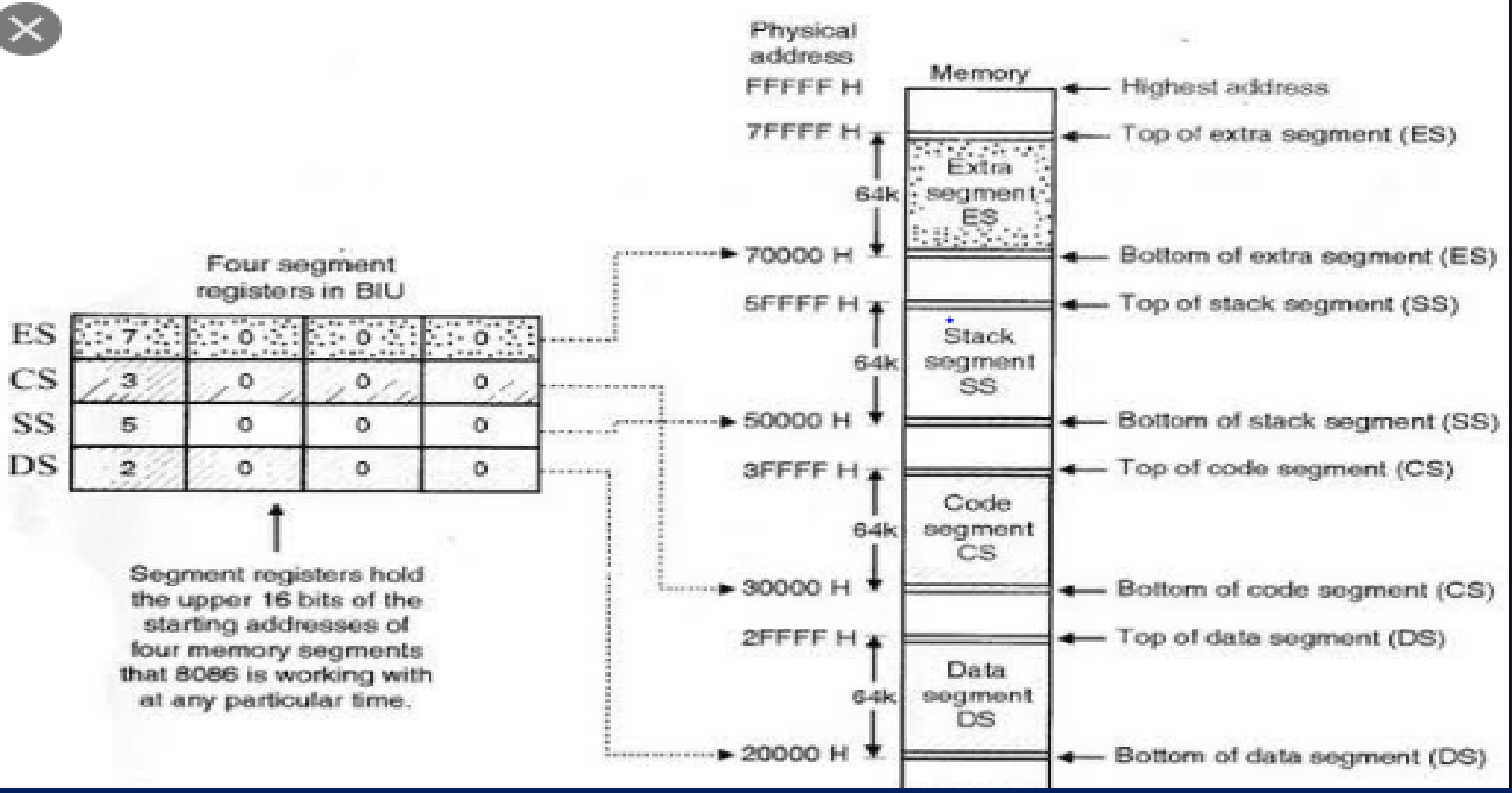
Memory segmentation:

- Segmentation is the process in which the main memory of a computer is logically divided into the different segments and each segment has its own base address.
- It is basically used to enhance the speed of the execution of the computer system.
- So that the processor is able to fetch and execute the data from the memory easily and fast.
- Memory in 8086 is segmented in different segments and this memory management techniques is called as segmentation.
- The complete physical memory is divided into the number of logical segments.
- Size of each segment is 64Kbyte.
- These segments are addressed by one of the segment register that are DS,CS,SS and ES.

Contd…

- These segment registers holds the starting address of a particular segments.
- The CPU of 8086 is able to address the 1Mbyte of physical memory.
- The complete 1 Mbyte memory can be divided into the 16 segments each of 64 Kbyte in size.
- So we need offset address or displacement address or effective address to address a specific memory location within a segment.
- The offset address is 16 bit so the maximum offset value will be FFFFH.
- The 16 bit offset or displacement is added to the 16 bit segment base register after shifting the contain of it towards left by one digit to get 20 bit physical address

Contd...


Physical address
Memory
FFFFF H
Highest address
7FFFF H
Top of extra segment (ES)
Extra segment ES
64k
70000 H
Bottom of extra segment (ES)
5FFFF H
Top of stack segment (SS)
Stack segment SS
64k
50000 H
Bottom of stack segment (SS)
3FFFF H
Top of code segment (CS)
Code segment CS
64k
30000 H
Bottom of code segment (CS)
2FFFF H
Top of data segment (DS)
Data segment DS
64k
20000 H
Bottom of data segment (DS)
Four segment registers in BIU
ES 7 0 0 0
CS 3 0 0 0
SS 5 0 0 0
DS 2 0 0 0
Segment registers hold the upper 16 bits of the starting addresses of four memory segments that 8086 is working with at any particular time.

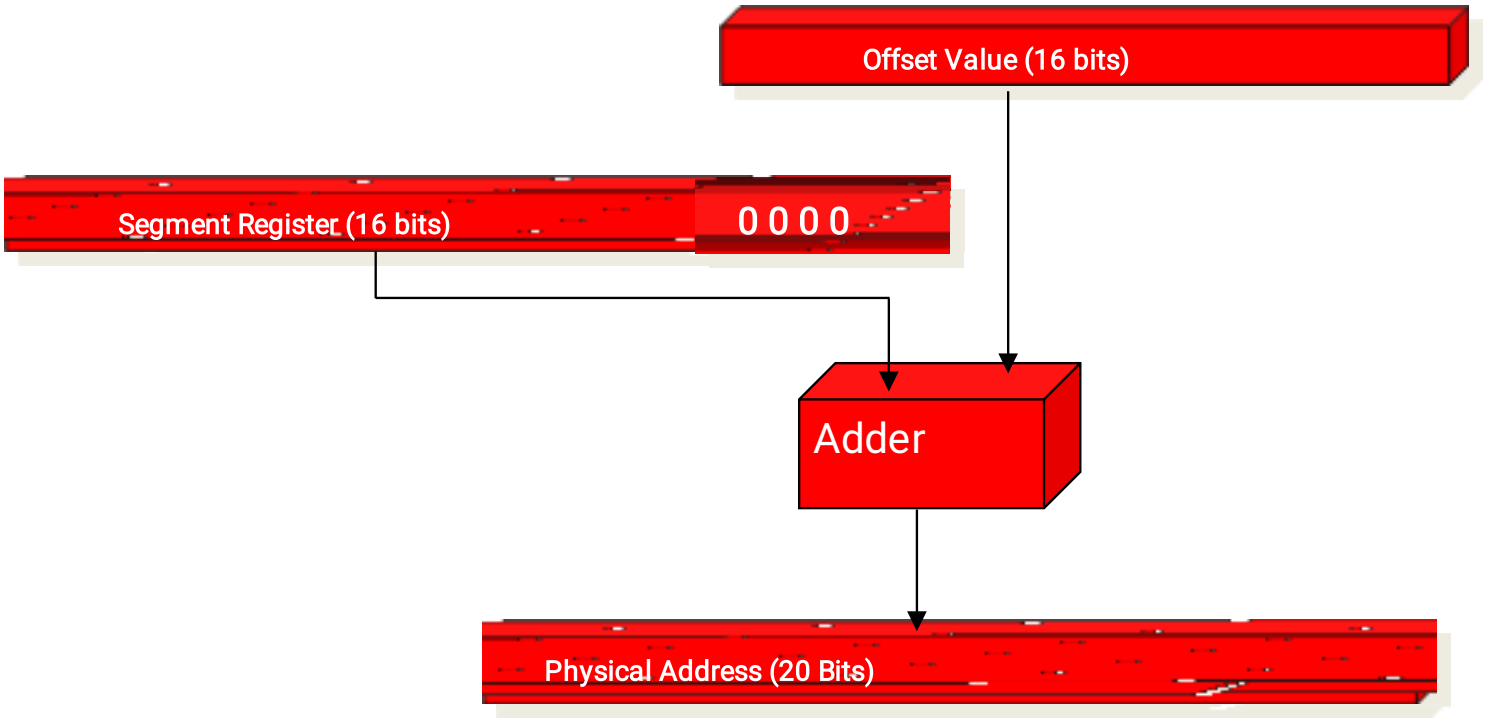
Contd...

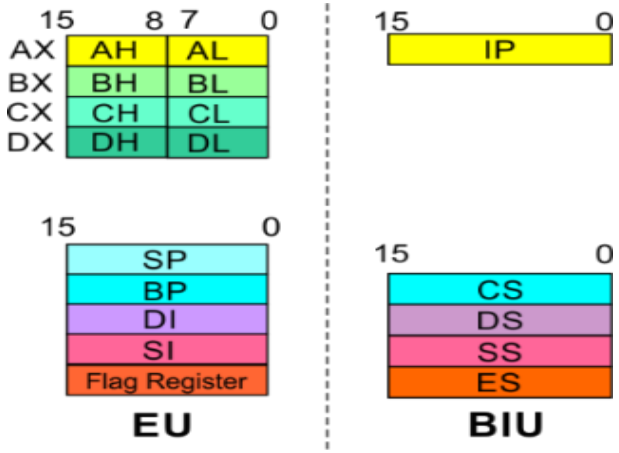
Advantages of the Segmentation The main advantages of segmentation are as follows:

- It provides a powerful memory management mechanism.
- Data related or stack related operations can be performed in different segments.
- Code related operation can be done in separate code segments.
- It allows to processes to easily share data.
- It allows to extend the address ability of the processor, i.e. segmentation allows the use of 16 bit registers to give an addressing capability of 1 Megabytes. Without segmentation, it would require 20 bit registers.

Contd...

Physical Address Generation





1.What is displacement? How does it determine memory address in a MOV [2000],CL instruction?

Ans. Assume DS=A000H,Memory address in MOV [2000H],CL is given below

Zero is inserted ↓

```
A  0  0  0  0     DS
   2  0  0  0     Displacement
.........................................
PA= A  2  0  0  0
```

2.State the Default segment base and offset pair register .if CS=1000H and IP=2000H then from which memory location 8086 reads an instruction.

Ans. The default segment base and offset pair register are CS:IP and SS:SP

Zero is inserted

```
      1  0  0  0  0    CS
         2  0  0  0    IP
.....
      PA= 1  2  0  0  0
```

3.Calculate the physical address generated by
i)4370 : 561E
ii)7A32 : 0028

3.If DS=C239H and SI=8ABCH then calculate physical address.

Ans. Given data DS=C239H , SI=8ABCH(IP)

Zero is inserted ↓

C	2	3	9	0	DS
	8	A	B	C	IP

…………………………………………………………………………

PA= ???????

Concepts of pipelining in 8086:

- Pipelining is simply pre-fetching instruction and lining up them in queue.
- The process of fetching the next instruction when the present instruction is being executed is called as pipelining.
- Pipelining has become possible due to the use of 6 byte queue.
- BIU (Bus Interfacing Unit) fills in the queue until the entire queue is full.
- BIU restarts filling in the queue when at least two locations of queue are vacant.

Contd…

Concepts of pipelining

- The Technique used to enable an instruction to complete with each clock cycle is called as pipelining.
- In Pipelined processor, the fetch, decode and execute operation are performed in parallel.
- So,pipelinging improve the execution speed of microprocessor.

	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6
Instruction 1	Fetch	Decode	Execute			
Instruction 2		Fetch	Decode	Execute		
Instruction 3			Fetch	Decode	Execute	
Instruction 4				Fetch	Decode	Execute

Advantages of Pipelining

- ➤Pipelining enables many instructions to be executed at the same time.
- ➤It allows execution to be done in fewer cycles.
- ➤Speed up the execution speed of the processor.
- ➤More efficient use of processor.

•The execution unit always reads the next instruction byte from the queue in BIU. This is faster than sending out an address to the memory and waiting for the next instruction byte to come.

•In short pipelining eliminates the waiting time of EU and speeds up the processing. The 8086 BIU will not initiate a fetch unless and until there are two empty bytes in its queue. 8086 BIU normally obtains two instruction bytes per fetch.

Interrupts in 8086

An interrupt is a condition that halts the microprocessor temporarily to work on a different task and then return to its previous task.

Interrupt is an event or signal that request to attention of CPU. This halt allows peripheral devices to access the microprocessor.

Whenever an interrupt occurs the processor completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler.

ISR is a program that tells the processor what to do when the interrupt occurs. After the execution of ISR, control returns back to the main routine where it was interrupted.

The different types of interrupts present in 8086 microprocessor are given by

Contd...

Hardware Interrupts –
Hardware interrupts are those interrupts which are caused by any peripheral device by sending a signal through a specified pin to the microprocessor. There are two hardware interrupts in 8086 microprocessor.

They are:*(A) NMI (Non Mask able Interrupt)* – It is a single pin non mask able hardware interrupt which cannot be disabled. It is the highest priority interrupt in 8086 microprocessor. After its execution, this interrupt generates a TYPE 2 interrupt. IP is loaded from word location 00008 H and CS is loaded from the word location 0000A H.

(B) INTR (Interrupt Request) – It provides a single interrupt request and is activated by I/O port. This interrupt can be masked or delayed. It is a level triggered interrupt. It can receive any interrupt type, so the value of IP and CS will change on the interrupt type received.

Contd...

software Interrupts – These are instructions that are inserted within the program to generate interrupts. There are 256 software interrupts in 8086 microprocessor. The instructions are of the format INT type where type ranges from 00 to FF. The starting address ranges from 00000 H to 003FF H. These are 2 byte instructions. IP is loaded from type * 04 H and CS is loaded from the next address give by (type * 04) + 02 H.

Some important software interrupts are:

(A) *TYPE 0* corresponds to division by zero(0).

(B) *TYPE 1* is used for single step execution for debugging of program.

(C) *TYPE 2* represents NMI and is used in power failure conditions.

(D) *TYPE 3* represents a break-point interrupt.

(E) *TYPE 4* is the overflow interrupt.

Thank You