

UNIT-IV

Microprocessor-I

Microprocessor:-

It is a semiconductor device which is manufactured by ~~VLSI~~ technology which includes ALU control unit and a group of registers in a single integrated circuit.

Microcontroller:-

It is a device that includes microprocessor memory and I/O signal lines on a single chip fabricated using VLSI technology.

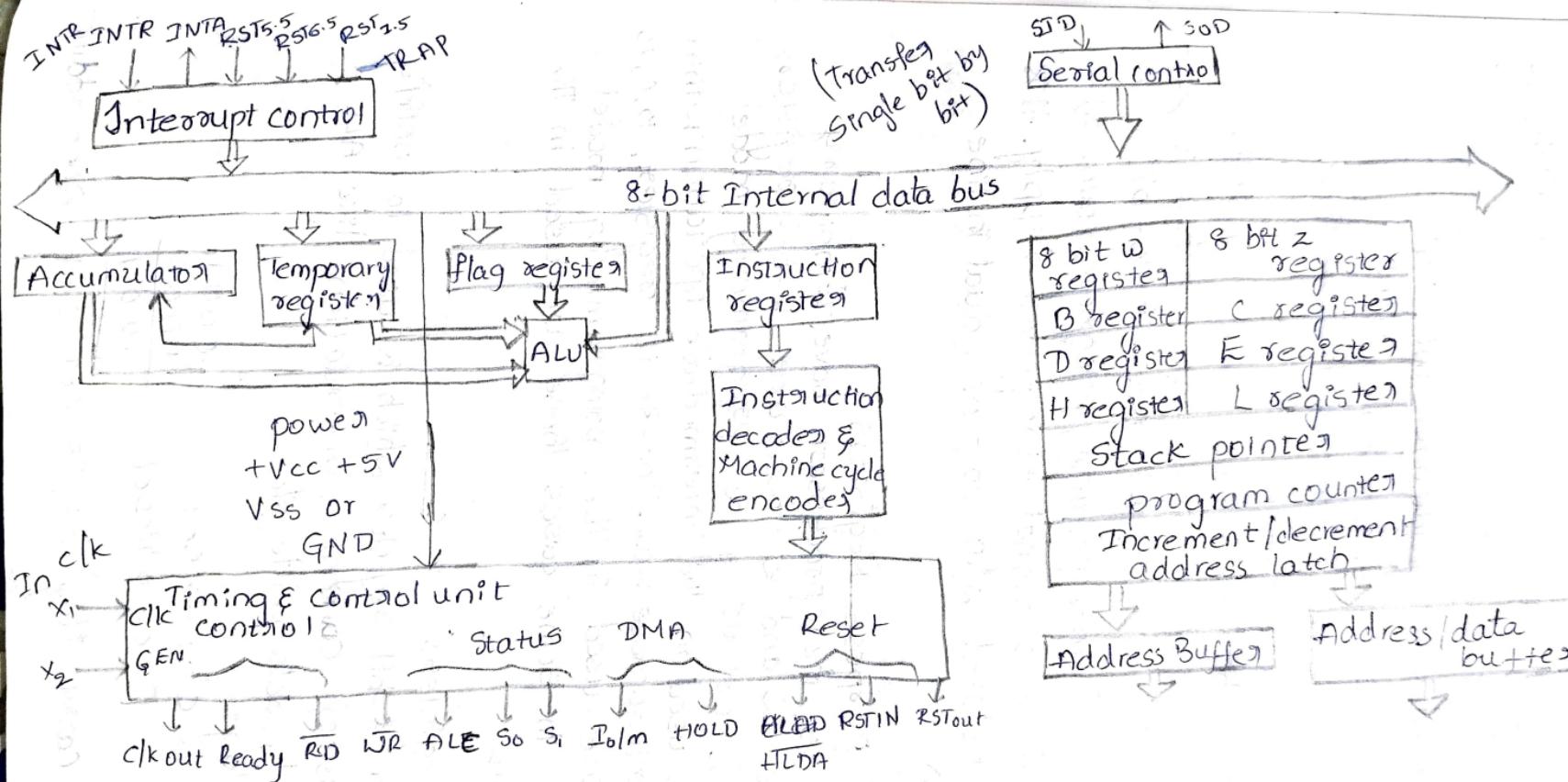
Bus :-

- A group of wires or lines used to transfer bits between the microprocessor and other components of the Computer System.
- A path used to carry signals such as connection b/w memory and microprocessor in a digital computer.

features of 8085 Microprocessor:-

- * It is a 8 bit microprocessor.
- * It has 16-bit address lines and hence can address upto $2^{16} = 65536$ bytes (64 KB) Memory locations through A_0 to A_{15} .
- * Data bus is group of 8 lines D_0 to D_7 .
- * In 8085 the lower 8 bit address A_0 to A_7 and data Bus D_0 to D_7 are multiplexed to reduce number of external pins but due to this external hardware (latch) is required to separate address lines and data lines.
- * It can operate with a 3 mega hertz clock frequency.

Architecture of 8085 Microprocessor :-



Addressing modes:

- * It supports 74 instructions with the following addressing modes.
 - 1) Immediate
 - 2) Register
 - 3) Direct
 - 4) Indirect
 - 5) Implied
- * It provides 5 hardware interrupts INTR, RST_{5.5}, RST_{6.5}, RST_{7.5}, TRAP
- * It has Serial IO control which allows Serial communication.
- * It has 8 bit accumulator, flag register, instruction registers, 6 8-bit general purpose registers (B, C, D, E, H, L) and 2 16-bit registers (Stack pointer, program counter)
- * It requires a Signal +5V power supply.
- * It is enclosed with 40 pins (dual in line package)

General purpose Registers

In 8085 microprocessor have temporary registers W and Z those are 8-bit registers which are not using the programming model. These are used in the intermediate operation within in the microprocessor. The 8-bit register, B, C, D, E, H, L are used in the programming model and also in some cases these are used in the BC, DE, HL

Flag register:-

It is a 8-bit register which is used to notify the status of current manipulation like ALU, data read, data write etc.

Carry flag:-

C = 1 → Set

C = 0 → Reset

0001

1000

1001

The carry flag bit is set $\frac{C=1}{C=1}$ when addition

of 2 numbers generate a carry or subtraction of 2 numbers generate a borrow otherwise it is '0' when no carry and no borrow.

Zero flag:-

If it is set that is $Z=1$ when the arithmetic or logic operation on any manipulation of strings generate a '0'. otherwise it is reset when non-zero result appear.

Auxiliary carry flag:-

If it is set that is $AC=1$ when carry is generated and transmitted through lower nibble to higher nibble otherwise it is reseted lower nibble.

eg:-
$$\begin{array}{r} \text{HIN} \quad \text{LN} \\ \begin{array}{r} 0011 \quad 1110 \\ 1001 \quad 1001 \\ \hline 1101 \quad 0111 \end{array} \end{array}$$

Parity flag:-

If it is set that is $P=1$ when even number of 1's in the lower byte of the 16-bit word otherwise it is reset when it is odd number of 1's.

Signed flag:-

If it is set that is $S=1$ when the most significant bit of the result having '1' that it is a negative number otherwise it is reset for $S=0$ positive number.

Eg $21 = 10101 = 00010101$

$-21 = \underline{11101010}$

= Negative Number

$$\begin{array}{r} 2 | 21 \\ 2 | 10 - 1 \\ 2 | 5 - 0 \\ 2 | 2 - 1 \\ 1 - 0 \end{array}$$

ALU:-

It performs all the arithmetic and logic operations in the length of 8 bit.

Instruction register:-

It is a 8-bit register which stores the all already executed instructions and used in the instruction decoder and machine type cycle.

encoder block.

Instruction decoder & Machine cycle encoder:-

It decodes the already executed instructions and the operational code generated by the hardware machinery of the system. It is in the form binary 0s and 1s. It is also called machine code.

It is decoded into physical address of the corresponding memory location of the particular instruction and the machine cycle encoder having the set of 7 different machine cycles as which I/O device and memory data was accessible.

Stack pointer:-

It is a 16 bit register. It stores the address of the top of the stack and also we can insert or delete the data from the stack by using stack pointer.

Push:-

Inserting an element push AX, 05H

Pop:-

Deleting an element pop AX, 06H

5
4
3
2
1

for inserting the data on stack we use the instruction push then stack pointer increment by 1 that is stack pointer +1. To the reverse the path then stack pointer decrement by 1 that is "SP - 1".

Program Counter:-

It is a special 16-bit register. It stores the address of each and every instruction which are to be executed.

Increment /decrement ^{address} latch:-

It increments and decrements the addresses which are present in the stack pointer and program counter for proper execution.

Address Buffer :-

It stores the addresses of the higher byte A₈ to A₁₅ and it is a unidirectional buffer and generates only addresses.

Write : Transmitting the data

Read : Receiving the data

Address / Data Buffer :-

It is a Bidirectional. It generates addresses and data multiplexes to each other AD₀ - AD₇

Timing and control unit :-

This unit maintains standard time for executing each and every instruction and supervises the read and write operation by generating controlling signals.

Clock generator :-

Circuit pulse is generated with the help of crystal oscillator having the X₁ and X₂.

clock out :-

The processor makes the output block based on the clock frequency speed.

Status:-

S₀, S₁ those are the status for the corresponding read and write operation.

DMA :- (Direct Memory access)

Io/M, Hold, HLDA these are the controlling signals to which IO device or memory the data was read or write.

RSTIN, RSTOUT

These signals are used for restoring the entire System from the beginning. It is a active low Signal. It is always connected to the ground terminal that is 0 volts.

Interrupt Control:-

As the name suggests it controls the interrupts during a process. When a microprocessor is executing a main program and whenever an interrupt occurs the microprocessor shifts the control from the main program to process the incoming request. After the request is complete the control goes back to the main program. There are 5 interrupt signals.

INTR, RST_{5.5}, RST_{6.5}, RST_{7.5}, TRAP

Serial Input/Output control:-

It controls the serial data communication by using the two instructions i) Serial input data

ii) Serial output data.

Accumulator:-

It is an 8-bit register used to perform arithmetic, logical, I/O and load/store operations. It is connected to internal data bus and ALU.

Differences b/w microprocessor and microcontroller

Microprocessor vs Microcontroller

- | | |
|---|---|
| 1) CPU is standalone RAM | 1) CPU, RAM, ROM, I/O and ROM, I/O and timers are all on a single chip. |
| 2) Designer can decide on the amount of ROM, RAM and I/O ports. | 2) Fixed amount of on chip ROM, RAM, I/O ports. |
| 3) Expensive | 3) Used for applications in which cost, power and space are critical |
| 4) General purpose | 4) Single purpose (Based on application) |
| 5) Difficult to use and design | 5) Easy to use and design. |

Diff features of 8086:-

- * It is a 16-bit microprocessor
- * It can be available in 40 pins dual in line package
- * It requires Vcc = +5 Volts power supply.
- * 8086 can also have 20 bit address bus
- * 8086 can have 16-bit data bus
- * 8086 can operate any one of the two modes namely minimum mode and maximum mode
- * Architecture of 8086 is Subdivided into 2 units
 - 1) execution unit
 - 2) Bus Interface unit
- * 8086 can have 14 registers of 16-bit size.
- * It supports 2 hardware interrupts and 256 software interrupts.
- * It allows pipelining in execution of instructions
- * Memory of 8086 is Segmented & logical Segments ES, CS, SS, DS
- * It has 16-bit flag register but it allows only 9 flags.
- * Higher execution speed.
- * Physical address can be calculated by using Segment address and offset address.

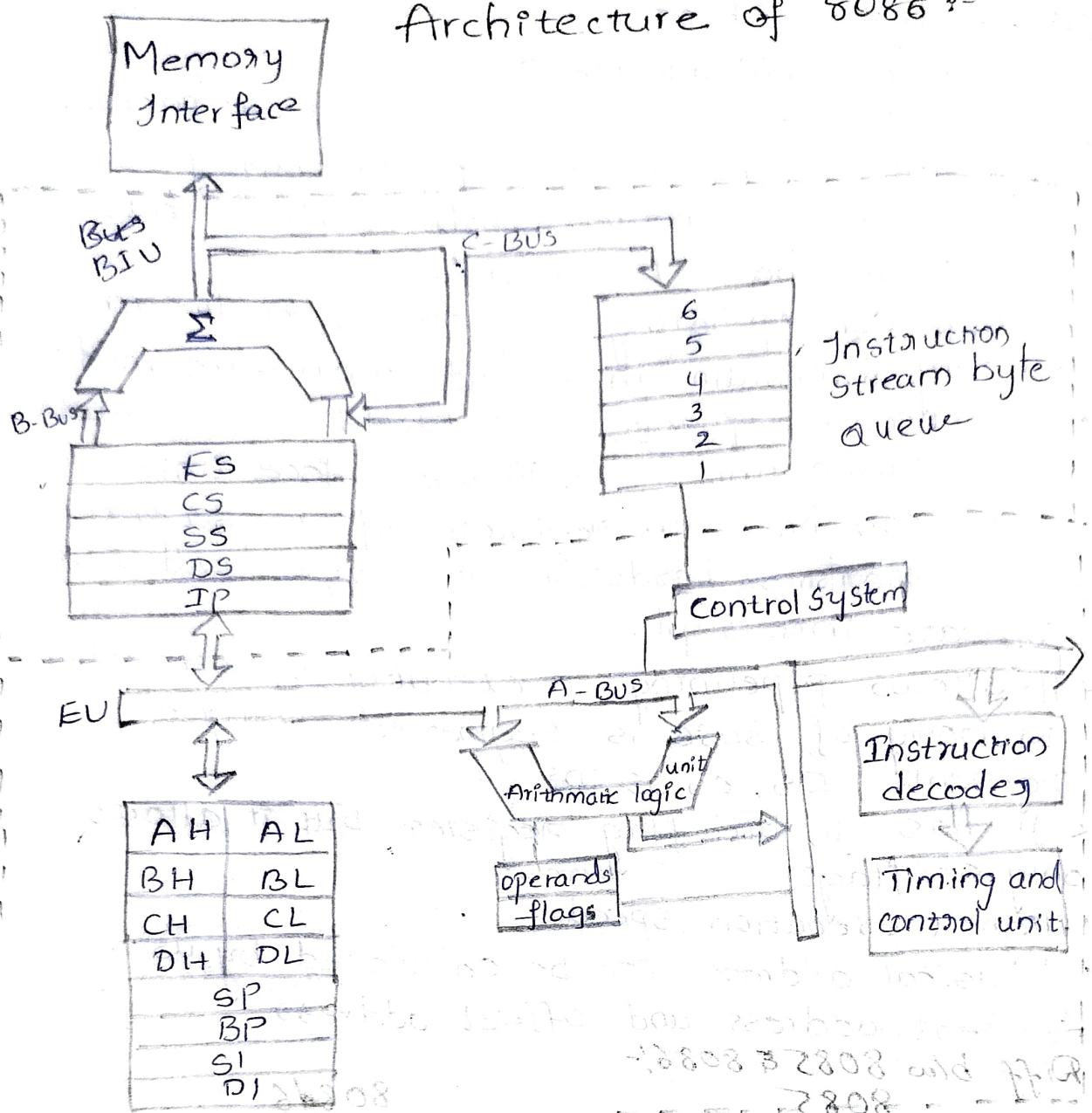
Diff b/w 8085 & 8086:-

8085

8086

- | | |
|--|---|
| 1) 8-bit microprocessor | 1) 16-bit microprocessor. |
| 2) 8-bit data bus Size | 2) 16-bit data bus Size |
| 3) 16-bit address bus Size | 3) 20-bit address bus size. |
| 4) 64 KB memory Capacity | 4) 1MB memory Capacity. |
| 5) 8-bit flag register | 5) 16-bit flag register. |
| 6) Memory is not Segmented | 6) Memory is Segmented |
| 7) ALU can perform only 8-bit operations at a time | 7) ALU can perform 16-bit operations at a time. |
| 8) 5 Hardware interrupts | 8) 2 Hardware Interrupts |
| 9) 8 Software Interrupts | 9) 256 Software Interrupts |

Architecture of 8086 :-



~~Flag register~~

A diagram illustrating a 16-bit register. The register is divided into two 8-bit bytes: the upper byte and the lower byte. The upper byte consists of bits 15 to 8, and the lower byte consists of bits 7 to 0. The register contains the following binary values:

U	U	U	U	OF	DF	FF	FF	SF	ZF	U	AF	U	PFU	CF	8
				OF				Sig	Zero						carry

Annotations indicate the upper byte (bits 15-8) and lower byte (bits 7-0). The label 'upper byte' is positioned below the first eight columns, and the label 'lower byte' is positioned below the last eight columns.

It is a 16-bit register in this 9 active flags and remaining are unused flags. The 9 active flags is again divided into 2 units they are status flags and control flags.

The status flags indicate the status of the result that is obtained after the execution of

arithmetic and logical operations. Direction flag, interrupt flag and Trap flag are the control flags they can control the operation of the CPU.

Carry flag :-

It holds the carry after 8-bit or 16-bit addition and holds borrow after 8-bit or 16-bit Subtraction. If carry = 1 it is set and otherwise it is reset.

Parity flag :-

If the lower 8-bit of the result is having odd parity (i.e. odd number of 1's) p-flag is said to '0'. If parity flag is said to '1' the lower 8-bit of result is having even parity.

Auxiliary carry flag :-

It holds the carry after addition the borrow after subtraction of the bits in bit-position-3 (least significant bit is treated as bit position-0) used by DAA (Decimal after addition), DAS instructions to adjust the value in after a BCD addition or subtraction.

Zero flag :-

Indicates that the result of an arithmetic or logic operation is '0' if $ZF=1$ the result is '0'. If $ZF=0$ the result is not a '0'.

Sign flag :-

It holds the arithmetic sign of the result after an arithmetic or logic operation is executed if $SF=1$ the sign bit is 1 the result is negative. If $SF=0$ the sign bit is 0 the result is positive.

Overflow flag :-

Signed numbers are represented in 2's complement form when signed numbers are added or subtracted overflow may occur indicate that the result has exceeded the capacity of the machine.

In 16-bit register. The minimum and maximum value of the signed numbers they can be stored in

-32768 ($= 8000H$) and ($+32767$) ($= FFFFH$)

for operations of an unsigned data overflow is ignored

Trap flag:-

It is set when single step by step execution done by the user otherwise reset.

Interrupt flag:-

It is set when an interrupt occurs while executing an application program, otherwise it is reset.

Direction flag:-

It is set when string operations are done from highest to lowest address otherwise it is reset.

Register Organization of 8086 :-

Accumulator		Code		Stack		Data		Pointers	
AX	AH	AL	CS	SP	BP	SI	DI	TP	IP
BX	BH	BL	SS						
CX	CH	CL	DS						
DX	DH	DL	ES						

General purpose Segment pointers &

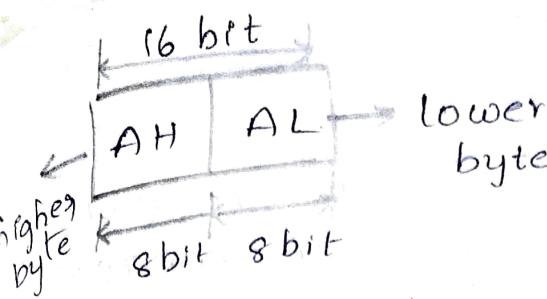
registers

Index register.

General purpose registers:-

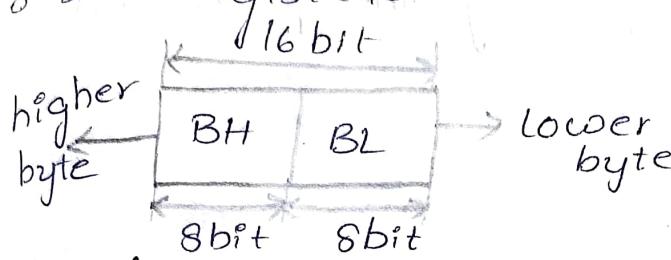
It has 4 general purpose registers which are 16-bit registers

1) AX (Accumulator) :- It stores the default given values and also resulted outcomes. It divides into lower byte and higher byte which are 2 8-bit registers.



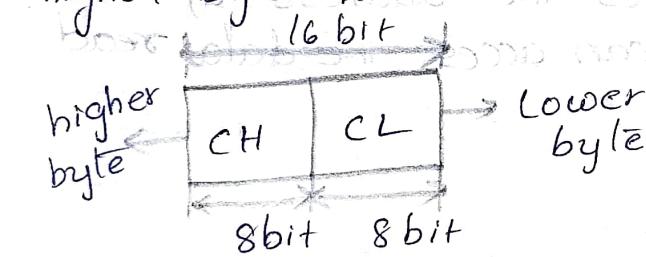
3) BX (starting address or Base registers):-

It stores the base or starting address of second operand given by the user. It divides into Lower byte and higher byte which are 2 8-bit registers.



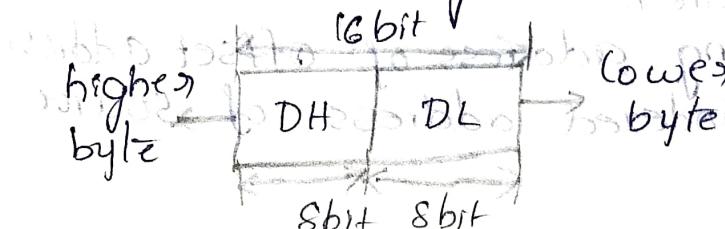
3) CX (count register):-

It stores the count values of repeated string manipulation like shift left, shift right, rotate right, rotate left. It divides into lower byte and higher byte which are 2 8-bit registers.



4) DX (Data register):-

It is used to store the overloaded data in the accumulator and also remainder in the 16-bit division and 16-bit multiplication whether it is signed or unsigned bits. It is also used for data read and write through in and out ports. It divides into lower byte and higher byte which are 2 8-bit registers.



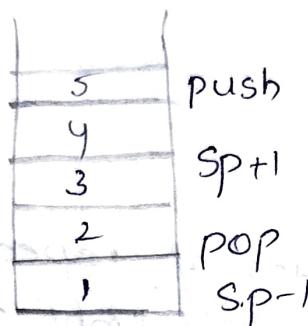
Pointers & Indexing registers:-

Pointers register:-

This register are used for current execution of the program we have two pointer registers
1) Stack pointer 2) Base pointer.

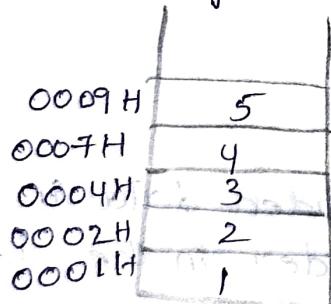
Stack pointer:-

It stores the address at the top of the stack and we can also used to insert and delete the data from the stack by using the push and pop operation.



Base pointer:-

It stores the address of all the data items in the stack that is it stores the address of beginning and ending. We can access the data read and write at any time.



Index Registers:-

It stores the starting address of operand registers. We have 2 types 1) Source indexing
2) destination indexing

Source indexing register:-

It stores the starting address of offset address or base address on indexed address of source operand register.

destination Indexing register:-

It stores the starting address of destination operand register.

MOV AL, 05H

MOV BL, 04H

ADD AL, BL

INT 03H

END START

Instruction pointer:-

It is a special type of register which stores the address of each and every instruction in the programming code that means it stores the address of next instruction.

Segment registers :-

The whole memory of 8086 is 1 MB and then it is divided into 4 Segment registers.

1) Code Segment:-

It is a 16-bit register which stores the instruction given by the user that is the programming code.

2) Stack Segment:- It stores the address of the current execution which are stored in the top of the stack.

3) Data Segment:- It stores the given input data items and obtained data which are numericals and α-numericals.

4) extra Segment:- It stores the data items which are overloaded by data segment.

Instruction Queue:-

It is a 16-byte length queue which stores the already executed instructions and forwarded in the manner of first in and first out.

The executed instruction are given to the decoding circuit which is in the execution unit.

Decoding circuit:-

In this block it gathers the already executed instructions convert their OP-code (operational code) into physical address to the corresponding memory location and also generates machine cycles. t_1, t_2, t_4 to the particular data read and data write.

Timing and control unit:-

This block generates standard timing of execution that is $\text{Instruction} = \frac{1}{400 \text{ ns}} (\text{nano seconds})$
 $= 2.5 \text{ MIPS}$
Mega instruction per second

Control unit:-

This block supervises over the execution it generates controlling signals that is

RD, WR, BUSY, WAIT.

Adder (Address conversion adder):

This block adds and stores the OP-code which are converted into physical address of their corresponding memory location. These physical addresses which are stored in the corresponding memory locations of the main memory.

Pin diagram of 8086:-

It is a 40-pins dual in line package by intel company in 1978. It is a 16-bit data bus line and 20 bit address line. It can be operated in 2 modes 1) Minimum mode 2) Maximum mode. If $MN/\bar{MX} = 1$ i.e. logic high it is connected to +VCC i.e. it is in minimum mode. If $MN/\bar{MX} = 0$ i.e. logic low it is connected to ground terminal i.e. it is a maximum mode.

(GND)	1	40	Vcc(5V)
AD14	2	39	AD15
AD13	3	38	AI6/S3
AD12	4	37	AI7/S4
AD11	5	36	AI8/S5
AD10	6	35	AI9/S6
AD9	7	34	BHE/S7
AD8	8	33	MN/MR
AD7	9	32	RD
AD6	10	31	R&G/TD HOLD
AD5	11	30	R&G/TD-HLDA
AD4	12	29	LOCK WR
AD3	13	28	S2 M/I/O
AD2	14	27	S1 DT/R
AD1	15	26	S0 DEN
AD0	16	25	QS ₀ ALE
NMI	17	24	QS ₁ INTA
INTR	18	23	TEST
CLK	19	22	READY
(GND)	20	21	RESET

Pin 1, 20, 40 :-

Pin 1, 20 are dedicated to ground terminal and pin 40 is dedicated for supply voltage i.e. 5V.

Pin 16 to 2 & 39

This 16 pins are dedicated for multiplexed address and data bus pin 39 for AD15 respectively.

When address latch enable is equal to 1 the multiplexed address and data bus will initiate the address to the corresponding read or write otherwise address latch enable is equal to zero ALE=0 which loads the data into the data bus for read or write.

Pin 35 to 38 :-

These 4 pins are used for multiplexed address and status bus when ALE=1 it decodes the address for the corresponding read or write.

Otherwise when $\overline{ALE} = 0$ it generates the status for Read or Write machine instruction cycle.

Pin 34 :-

This pin is dedicated for bus high enable signal when $\overline{BHE} = 0$ it reads or write the data to and from D8 to D5 higher byte of register. It is associated with status bit S_7 when $\overline{BHE} = 1$ i.e. S_7 is 0

Pin 32 :-

This pin is dedicated for read operation when $\overline{RD} = 0$. It reads the data into the memory or I/O devices otherwise $\overline{RD} = 1$ no read operation

Pin 23 :-

This pin is dedicated for test signal when $\overline{Test} = 0$ it will insert the wait state in between read or write operation cycles.



Otherwise $\overline{Test} = 1$ it just read or write the data

Pin 21 :-

It is dedicated for RESET. It is an active low signal when low logic is connected to this signal it restart the entire signal from the beginning.

Pin 22 :-

It is dedicated for READY signal when $READY = 1$ the address bus and data bus makes the read or write operation effectively

Pin 17 :-

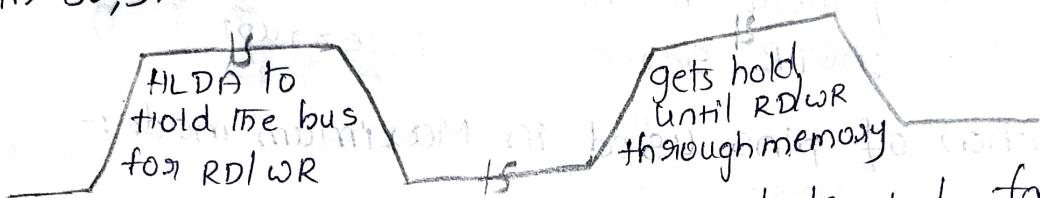
Non Maskable Interrupt input is used to request a hardware interrupt it cannot be disabled by software. It is a positive edge-triggered interrupt and when it occurs type 2 interrupt occurs in 8086.

pin 18:-
When $IF=1$ if INTR is held high then 8086 gets interrupted if $IF=0$ then INTR interrupt is disabled.

pin 19:-
It is dedicated for clock signal which is generated from external crystal or oscillator to initiate the functional blocks in the device. 8086 has 33 percent duty cycle having 5 MHz CPU clock frequency speed.

Functions of pins used in minimum mode.

pin 30,31:-



HOLD, HLDA these Two pins are dedicated for hold and hold acknowledgement by DMA controller when $HOLD=1$ at that time the System bus is in Read or write operation through memory and then HLDA gives the response for the HOLD signal to how much it is in hold state until it releases the bus.

pin 29 :-
it is dedicated for write operation when $WR=0$ write operation through memory, and I/O device

pin 28 :-
This pin is dedicated for memory and I/O device when $M/I/O=1$ then read or write through memory otherwise $M/I/O=0$ read or write through I/O device.

pin 27 :-
This is for data transmission receive when $DT/R=1$ data was transmitted through memory/ I/O device otherwise $DT/R=0$ data was received through Memory/ I/O device.

pin 26:-

This is for data enable signal when DEN=0
data was loaded into data bus either read
or write operations.

pin 25:-

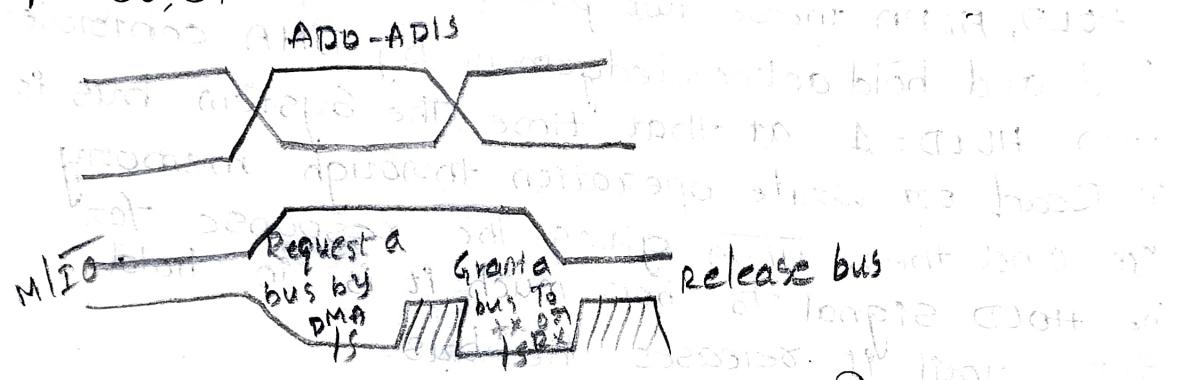
This pin is decodes the address from
multiplexed address and data bus. or
address and status bus. When ALE=0
the data and also status bus.

pin 24:-

When interrupt arises it just gives the
acknowledgement to the interrupt request signal
and then INTR serves the interrupt.

function of pins used in Maximum mode:-

pin 30,31:-



Qs, QS, Queue operation

- \Rightarrow NO operation
- \Leftarrow \Rightarrow Queue operation

This two pins are dedicated for
request and grant signals from DMA
controller when read and write data
through DMA. It request a bus grant the
bus for read (or) write and then
release the data.

Pin 24,25:-

These pins are multiplexed with address and data bus. It loads

- QSD Q3. Queue operation
No operator
- 0 0 First byte of operand read from the queue
- 0 1 Queue empty
- 1 0 Second byte of operand read from the queue.

Pin - 26, 27, 28:

- $\overline{S_1}$ 0 0 \leftrightarrow Read from memory
- S_2 0 0 \leftrightarrow Interrupt acknowledgement
- 0 0 1 \leftrightarrow Read from I/O devices
- 0 0 1 \leftrightarrow Write from I/O devices
- 0 1 0 \leftrightarrow Write (Halt)
- 0 1 1 \leftrightarrow Halt (H STOP)
- 1 0 0 \leftrightarrow Op-code is fetched
- 1 0 1 \leftrightarrow Read from memory
- 1 1 0 \leftrightarrow Write from memory
- 1 1 1 \leftrightarrow In active state (passive)

Pin 29: must be written to zero, when it is zero, this pin is dedicated for lock, when it undergoes lock position during read or write operation.

Addressing modes of 8086 [Explanation]

These are used for specifying the instruction Read or write within the register and use of physical Address.

1. Immediate Addressing Mode:

In this mode the content which is transferred to the corresponding register.

Syntax: MOV AX, 0012H

\Rightarrow The content 0012 was transferred into AX register.

Direct: In this mode the memory address of the corresponding location was directly moved into register.

Syntax: MOV AX, [5000H]

→ Offset address of 5000H is moved Ax register.

Register Direct Addressing mode:
In this mode Content in between two registers.

Register Indirect Addressing mode:
In this mode the transfer of offset address of one register was transferred into another register.

Syntax: MOV AX, [BX]

Indexed Addressing Mode:

In this mode either Source index (or) destination index offset address was transferred into another register.

Syntax:

MOV AX, [SI] or [DI] or [BX]

Based Indexed Addressing Mode:-

In this mode offset address of Source index (or) destination index is added with base register address and moved to another register.

Syntax:

MOV AX, [SI][BX]

Relative Base Indexed Addressing Mode:

In this mode offset address of Source index (or) destination index is added with address of base register and any content given by the user was moved to another register.

Syntax:

MOV AX, S0H[SI][BX]

Register Relative Addressing Mode:

In this mode offset address of Source index (or) destination index is added with any content was transferred to another register.

Syntax:

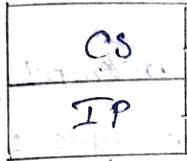
MOV AX, S0H[BX]

Interrupts of 8086

Interrupt vector Table
A variable interrupt
vectors (224)

Reserved interrupt
vectors (27)

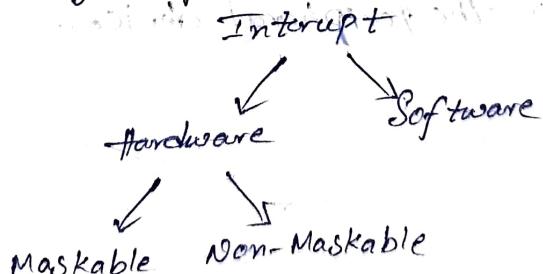
Dedicated
interrupted vectors (5)



003 FFH	Type FFH vector
003 FCH	Type 21H vector (available)
-	Type 9AH vector (available)
-	Type 1FH vector (available)
-	Type 05H vector (available)
00014H	Type 04H vector (available)
-	Type 03H vector [1 byte Int Instruction]
0000CH	Type 02H vector (NMI)
0000BH	Type 01H vector (Trap or Single Step)
00008H	Type 00H vector (Divide-by-0 error)
00007H	
00004H	
>00003H	
>00002H	
>00001H	
>00000H	

8-bits

- Interrupt is a method of creating a temporary halt during program execution & allows peripheral devices to access the microprocessors.
- The microprocessor corresponds to that interrupt with an interrupt service routine (ISR)
- Two types of interrupts;



- It should not be enabled using clear interrupt flag.
- The interrupt INTR is activated by an I/O port.
- If the interrupt is enable and non maskable interrupt (NMI) is disable then the microprocessor first complete the current instruction.

Software Interrupts!

These are instructions that are inserted within the program to generate interrupts.

1. Type - 00 H → Divide by zero (0) error
2. Type - 01 H → Trap (or) Single Step
3. Type - 03 H → Non Maskable Interrupt
4. Type - 04 H → 1 byte Interrupt instruction

Hardware Interrupts!

It is caused by peripheral devices by sending a signal through a specified pin to the microprocessor.

Non Maskable Interrupt:

This cannot be disabled by using interrupt flag (IF). It can also be processed by executing the type-2 Software interrupt.

Interrupt Request (INTR):

This can be either enabled by setting IF=1 (or) disabled by resetting the interrupt flag IF=0. The peripheral device sends a request on the pin if 8086 accepts this interrupt, then it sends an acknowledgement to the peripheral devices.