VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH
(Deemed to-be University U/s 3 of the UGC Act, 1956)

Micro Processors & Interfacing

16CS307

Mr. M Krishna Chennakesava Rao,

Asst. Professor, Dept. of ECE,

VFSTR University

Architecture of 8086 Microprocessor

Register organization of 8086

Why Registers ?

- To hold data, variables, intermediate results temporarily
- for counters (**CX**), to store OFFSET address (**BX**)

Registers in 8086

- 8086 has **Fourteen**, 16-bit Registers.
- In 8086, registers are categorized into **4 types**

1. General Data Registers (**4**)
2. Segment Registers (**4**)
3. Flag Register (**1**)
4. Pointers and Index Registers (**5**)

Architecture of 8086 Microprocessor

Register organization of 8086

AX	AH	AL	Accumulator
BX	BH	BL	Base → used to store OFFSET for forming physical address
CX	CH	CL	Counter in string & loop instructions
DX	DH	DL	Destination/ Implicit

General data register

CS
SS
DS
ES

Segment registers

FLAGS / PSW

SP
BP
SI
DI
IP

Pointers and index registers

X specifies 16-bits
H specifies higher 8-bits
L specifies lower 8-bits

Architecture of 8086 Microprocessor

Register organization of 8086

The registers AX, BX, CX, and DX are the **general purpose** 16-bit registers.

AX Register: Accumulator register consists of two 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, rotate and string manipulation.

BX Register: This register is mainly used as a **base register**.
It holds the **starting base location of a memory region** within a data segment.
It is **used as offset storage for forming physical address** in case of certain addressing mode.

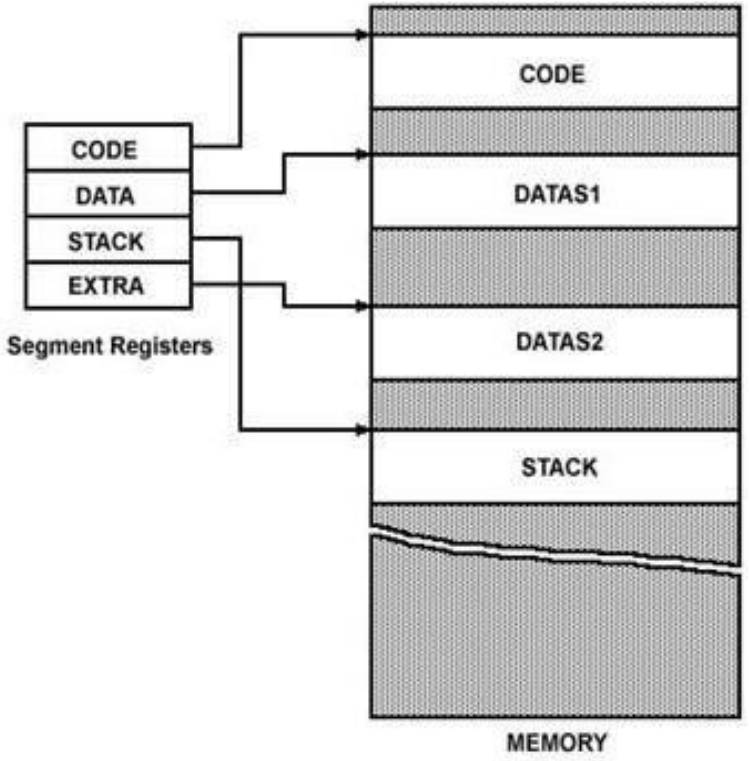
CX Register: It is used as default counter or **count register** in case of string and loop instructions.

DX Register: Data register can be used as a port number in I/O operations and implicit operand or destination in case of few instructions. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

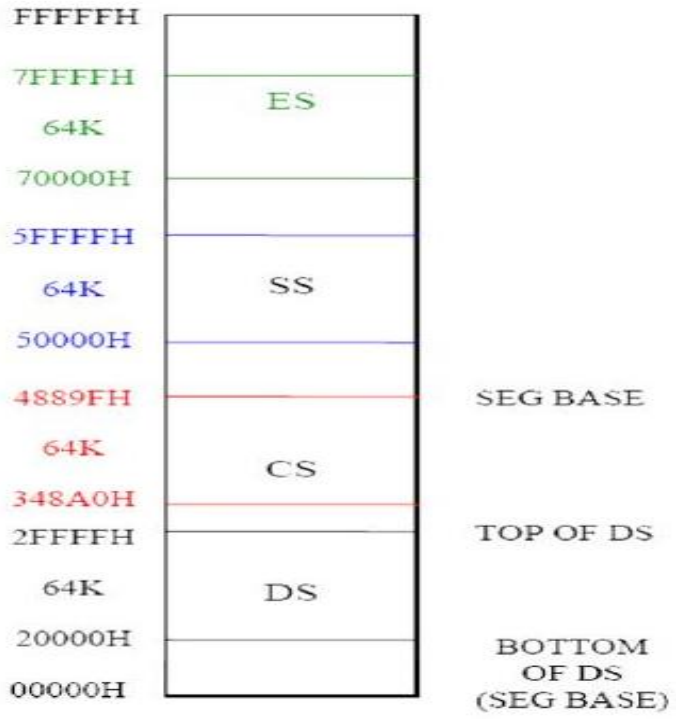
Register organization of 8086 cont'd…

Segment Registers of 8086

- 8086 processor addresses **segmented** version of **memory** (2^{20} =1MB size).
  Size of each segment is 64KB. ($2^{20} = 2^4\,2^{16} = 2^4$. 64KB= **Sixteen** 64KB **logical** segments)
- **Physical address =[Base address x 10] + OFFSET address**



Memory Segments of 8086



Segment Registers

Register organization of 8086 cont'd…

Pointers and Index Registers of 8086

SP	**Stack Pointer**	➢ Holds address of the **top of the stack**. ➢ Stores **OFFSET** within a segment
BP	**Base Pointer**	Stores **OFFSET** within a segment
IP	**Instruction Pointer**	Stores the **address of the next instruction** to be fetched
SI	**Source Index**	➢ Stores **OFFSET** ➢ also used as general purpose registers ➢ used for string related operations
DI	**Destination index**	

Architecture

Registers and Special Functions

Register	Name of the Register	Special Function
AX	16-bit Accumulator	Stores the 16-bit results of arithmetic and logic operations
AL	8-bit Accumulator	Stores the 8-bit results of arithmetic and logic operations
BX	Base register	Used to hold base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions
DX	Data Register	Used to hold data for multiplication and division operations
SP	Stack Pointer	Used to hold the offset address of top stack memory
BP	Base Pointer	Used to hold the base value in base addressing using SS register to access data from stack memory
SI	Source Index	Used to hold index value of source operand (data) for string instructions
DI	Data Index	Used to hold the index value of destination operand (data) for string operations

Register organization of 8086 cont'd…

Flag Register of 8086

FLAGS/ PSW

U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AC	U	PF	U	CY

9 active flags can be divided into 2 groups.
Conditional flags----6 b) Control flags----3

OV=overflow flag DF=direction flag TF=trap flag IF=interrupt flag SF=sign flag

ZF=zero flag AC=auxiliary carry flag PF=parity flag CY=carry flag

- **Flag Register** determines the **current state** of the processor.
- They are modified automatically by CPU after mathematical operations.
- It allows to determine the type of the result.
- Determines the conditions to transfer control to other parts of the program.

Execution Unit (EU)

Flag Register

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Sign Flag

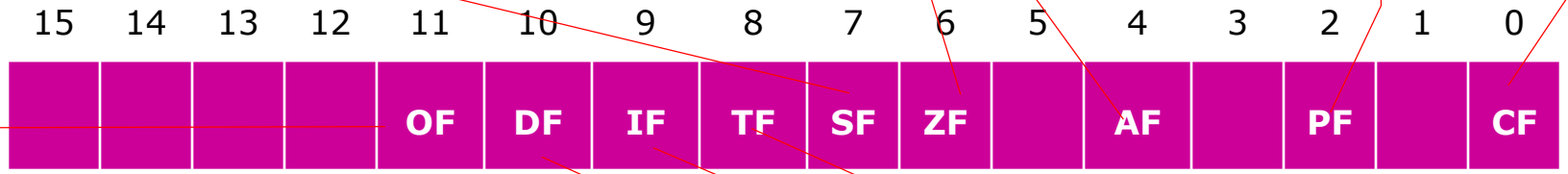
This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

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.

Tarp Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

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.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.