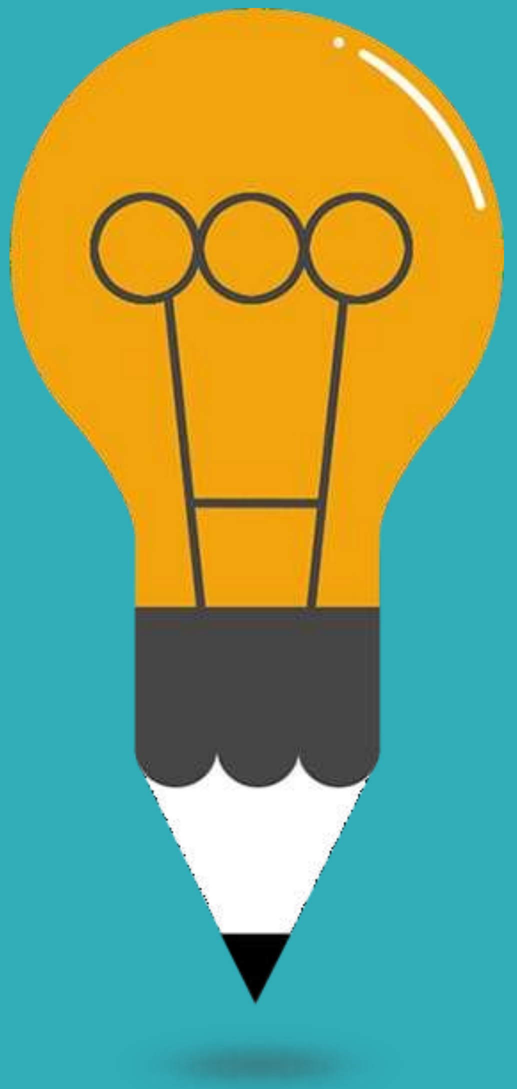


MICROPROCESSORS

8086 programmers model , Registers

PROF.POONAM PANGARKAR, ASSISTANT PROFESSOR , CSE-DATA SCIENCE ,APST THANE

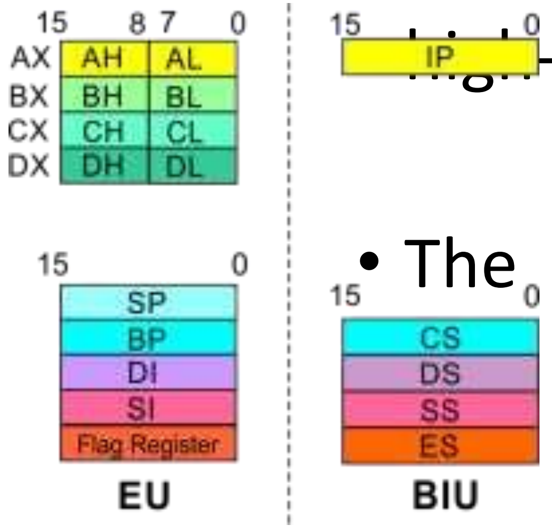
Architecture

Execution Unit (EU)

EU Registers

Accumulator Register (AX)

- Consists of two 8-bit registers AL and AH, which can be co and used as a 16-bit register AX.
- AL in this case contains the low order byte of the word, a high-order byte.
- The I/O instructions use the AX or AL for inputting / outp to or from an I/O port.



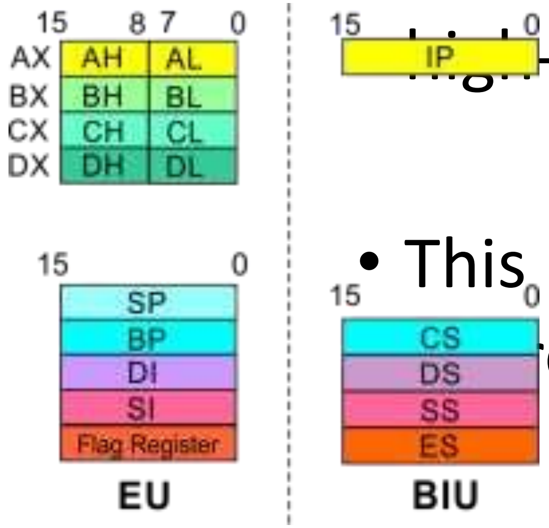
Architecture

Execution Unit (EU)

EU Registers

Base Register (BX)

- Consists of two 8-bit registers BL and BH, which can be co and used as a 16-bit register BX.
- BL in this case contains the low-order byte of the word, a high-order byte.
- This is the only general purpose register whose contents ca essing the 8086 memory.



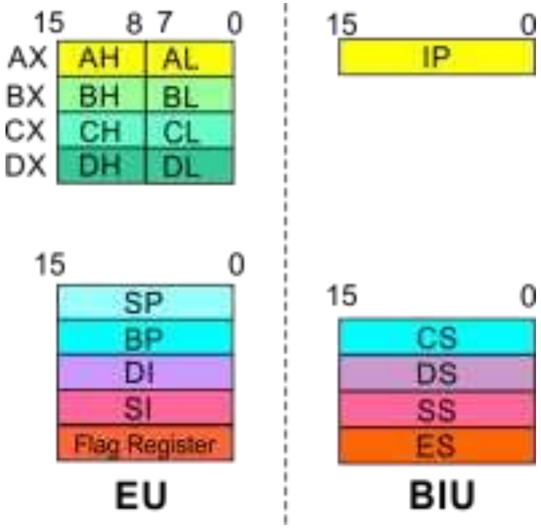
Architecture

Execution Unit (EU)

EU Registers

Counter Register (CX)

- **Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX.**
- **When combined, CL register contains the low order byte of the word, and CH contains the high-order byte.**
- **Instructions such as SHIFT, ROTATE and LOOP use the contents of CX as a counter.**



Example:

The instruction LOOP START automatically decrements CX by 1 without affecting flags and will check if [CX] = 0.

If it is zero, 8086 executes the next instruction; other wise the 8086 branches to the label START.

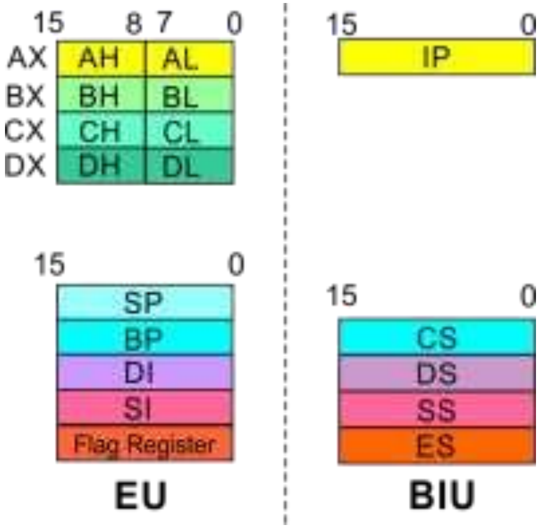
Architecture

Execution Unit (EU)

EU Registers

Data Register (DX)

- Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX.
- When combined, DL register contains the low order byte of the word, and DH contains the high-order byte.
- Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a 32 ÷ 16 division and the 16-bit reminder after division.



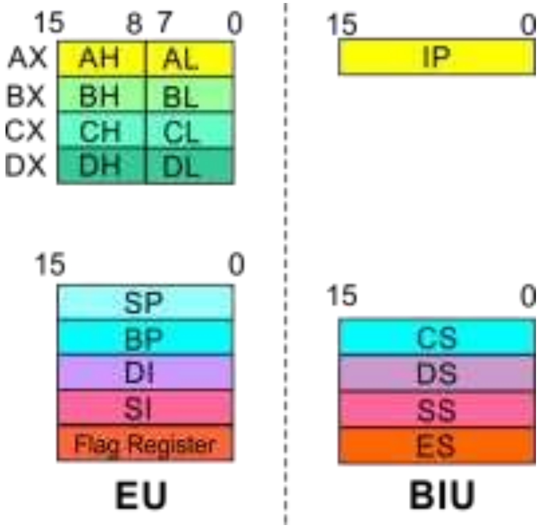
Architecture

Execution Unit (EU)

EU Registers

Stack Pointer (SP) and Base Pointer (BP)

- SP and BP are used to access data in the stack segment.
- SP is used as an offset from the current SS during execution of instructions that involve the stack segment in the external memory.
- SP contents are automatically updated (incremented/ decremented) due to execution of a POP or PUSH instruction.
- BP contains an offset address in the current SS, which is used by instructions utilizing the based addressing mode.



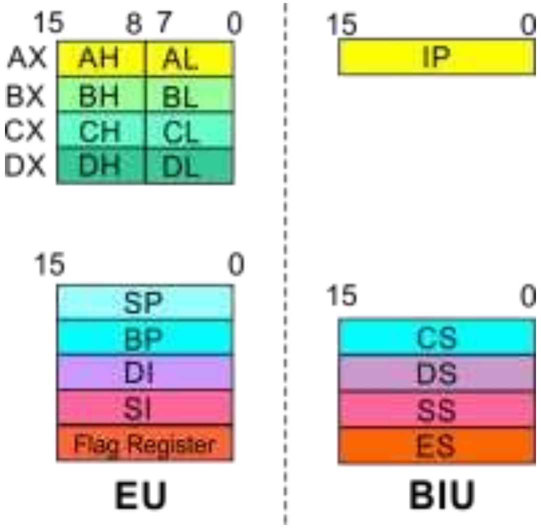
Architecture

Execution Unit (EU)

EU Registers

Source Index (SI) and Destination Index (DI)

- **Used in indexed addressing.**
- **Instructions that process data strings use the SI and DI r egisters together with DS and ES respectively in order to distinguish between the source and destination addresses .**



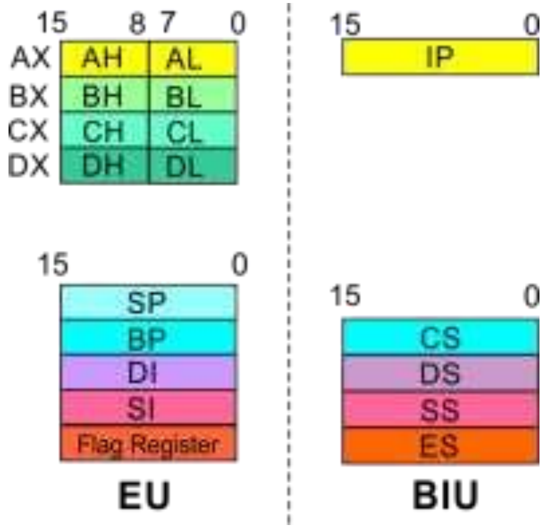
Architecture

Execution Unit (EU)

EU Registers

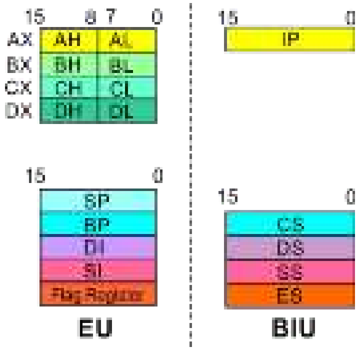
Source Index (SI) and Destination Index (DI)

- **Used in indexed addressing.**
- **Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to distinguish between the source and destination addresses.**



Architecture

8086 registers categorized into 4 groups



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register

Architecture

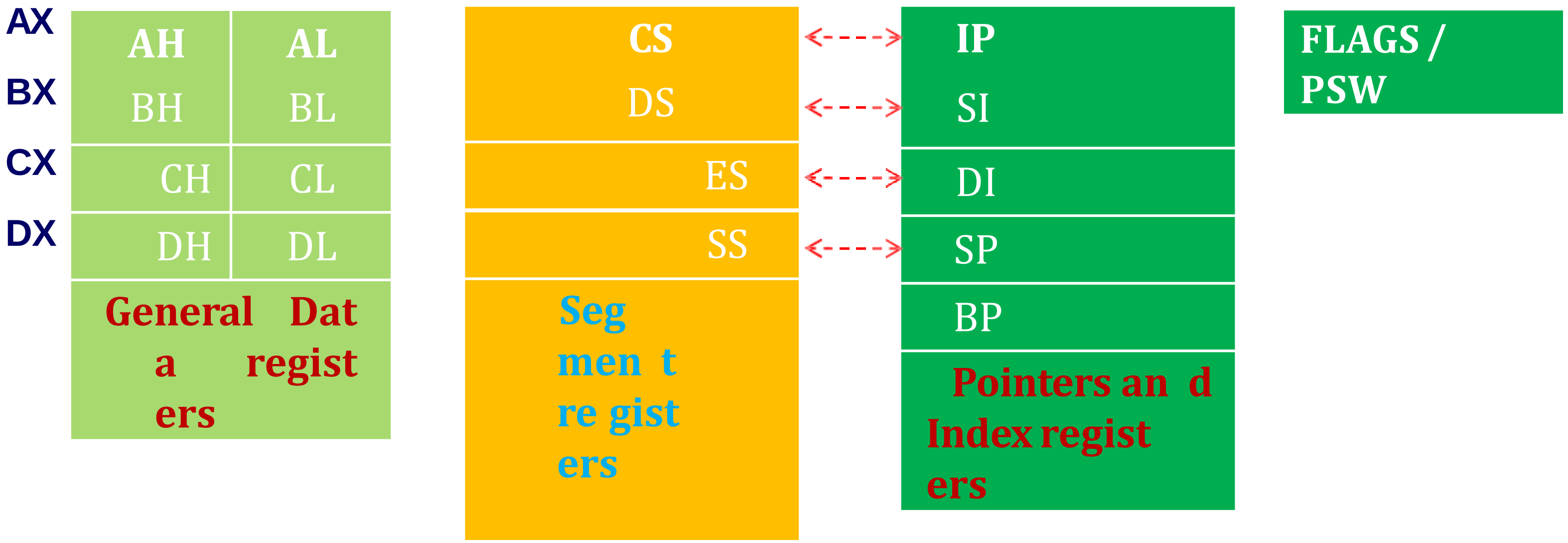
Registers and Special Functions

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 of 8086

8086 has a powerful set of registers that can be grouped as

- General Data register
- Segment registers
- Pointers & Index registers
- FLAG
- Only GPRs can be accesses as 8/16-bit while others as 16-bit only



AX AH AL

BX BH BL

CX CH CL

DX DH DL

General Data registers

CS <----> IP

DS <----> SI

ES <----> DI

SS <----> SP

BP

Segment registers

Pointers and Index registers

FLAGS / PSW

Special Purpose Registers:

- **Special Purpose Registers:** The special purpose registers are
    - ➢ Segment registers
    - ➢ Pointers and index registers

- **Segment Registers :** Unlike 8085, the 8086 addresses a segmented memory of 1MB, which the 8086 is able to address. The 1 MB is divided into 16 logical segments (16 X 64 KB = 1024 KB = 1 MB). Each segment thus contains 64 Kbytes of memory. There are four segment registers, viz. Code Segment Register (CS), Data Segment Register (DS), Extra Segment Register (ES) and Stack Segment Register (SS).

Special Purpose Registers:

Pointers and Index Registers

The pointers contain offset within the particular segments. The pointers IP, BP and SP usually contain offsets within the code, data and stack segments respectively. The index registers are used as general purpose registers as well as for offset storage in case of indexed, based indexed and relative based indexed addressing modes. The register SI is generally used to store the offset of source data in DMS while the register DI is used to store the offset of destination in DMS or EMS. The index registers are particularly useful for string manipulations.

Flag Register

- The FLAG is nothing but group of flip-flops which are affected (SET or RESET) immediately after an arithmetic or logical operation performed by the ALU.
- The flags of 8086 can be divided into two types: Conditional Flags and Control Flags
- Conditional Flags are affected immediately after an arithmetic or logical operation performed by the ALU. The SET or RESET condition of each flag is used to indicate the status of the result generated by the ALU.The 8086 has 6 conditional flags, out of which 5 are similar to the 8085 while Overflow flag is the additional flag.
- Control Flag **are not affected by Arithmetic o**r logical operation performed by the ALU but programmer can SET or RESET these Flags to Control certain operation/Instructions.