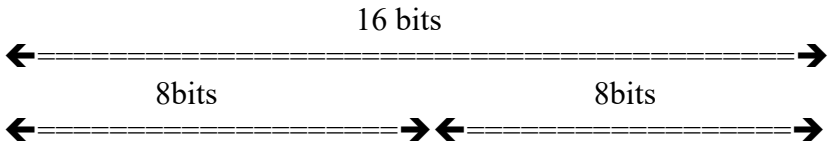
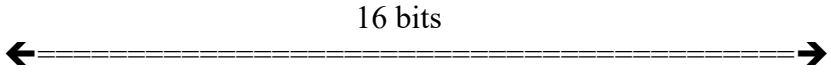
8086 Registers

General purpose registers



AL	**AH**	**AX**
BL	**BH**	**BX**
CL	**CH**	**CX**
DL	**DH**	**DX**

AX: Accumulator, Word multiply, word divide, word I /O
BX: Base, Store address information
CX: Count, String operation, loops
DX: Data,
Word multiply, word divide, indirect I/O (Used to hold I/O address during I/O instructions. If the result is more than 16-bits, the lower order 16-bits are stored in accumulator and higher order 16-bits are stored in DX register)



SP
BP
SI
DI

SP: Stack pointer
BP: Base pointer
SI: Source index
DI: Destination index

Pointer And Index Registers

- used to keep offset addresses.
- Used in various forms of memory addressing.
- In the case of SP and BP the default reference to form a physical address is the Stack Segment (SS-will be discussed under the BIU)
- The index registers (SI & DI) and the BX generally defaults to the Data segment register (DS).

SP: Stack pointer
– Used with SS to access the stack segment
BP: Base Pointer
– Primarily used to access data on the stack
– Can be used to access data in other segments

- SI Source Index register
  – is required for some string operations
  – When string operations are performed, the SI register points to memory locations in the data segment which is addressed by the DS register. Thus, SI is associated with the DS in string operations.
- DI: Destination Index register
  – is also required for some string operations.
  – When string operations are performed, the DI register points to memory locations in the data segment which is addressed by the ES register. Thus, DI is associated with the ES in string operations.

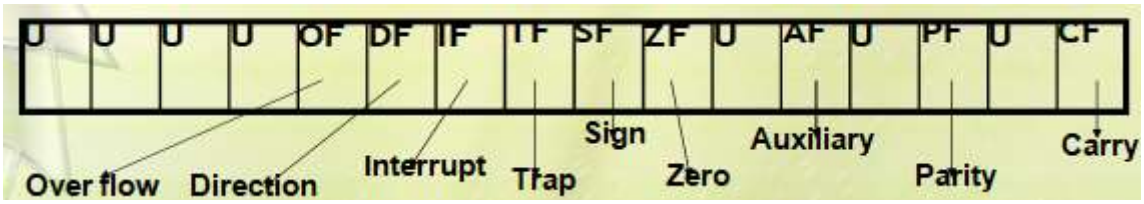
- The SI and the DI registers may also be used to access data stored in arrays

Segment registers

- In 8086/88 the processors have 4 segments registers
- Code Segment register (CS), Data Segment register (DS), Extra Segment register (ES) and Stack Segment (SS) register.
- All are 16 bit registers.
- Each of the Segments registers store the upper 16 bit address of the starting address of the corresponding segments.

Flags Register

Flag Name	**Set**	**Clear**
Overflow(yes/no)	OV	NV
Direction(increment/decrement)	DN	UP
Interrupt(enable/disable)	EI	DI
Sign(negative/positive)	NG	PL
Zero(yes/no)	ZR	NZ
Auxiliary carry(yes/no)	AC	NA
Parity(even/odd)	PE	PO
Carry(yes/no)	CY	NC



Instruction register

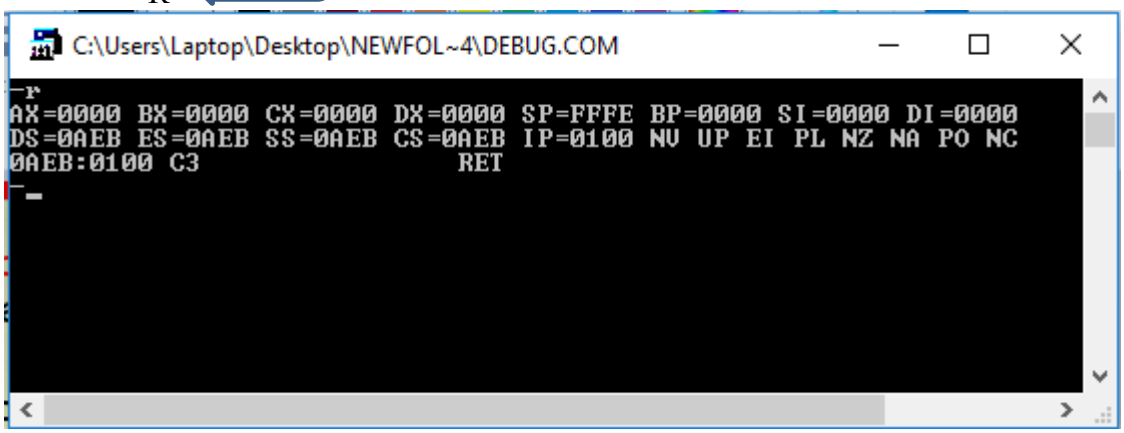
IP or PC :
• The instruction pointer register contains a 16-bit offset address of instruction that is to be executed next.
• The IP always references the Code segment register (CS).

Register: R [register]

How to display the registers of the 8086? By using debugger command (R):

Enter an ‘R or r ' at the first DEBUG prompt, DEBUG will display something similar to this:

- R

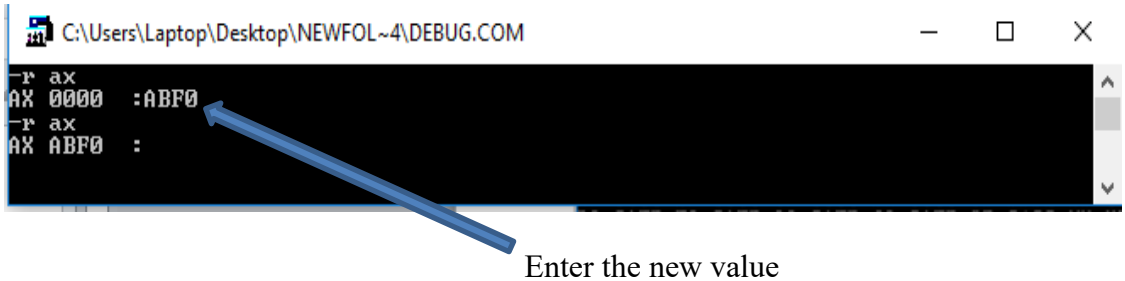


How to display a specific register of the 8086? Also by using debugger command (R):

- R AX

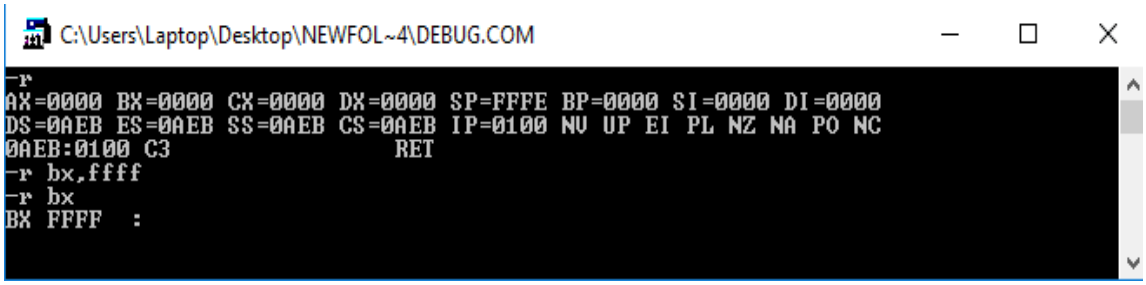
How to change the value of a specific register? Also by using debugger command (R):

- R AX 



Enter the new value

Or



How to move the value of register to another?

- R AX,BX