General Purpose Registers of 8086

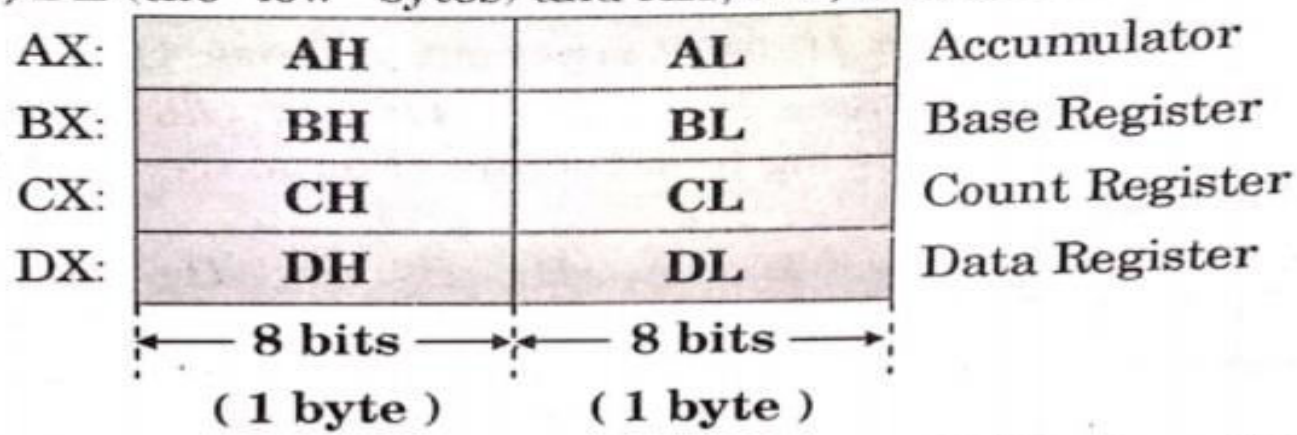
General Purpose Registers of 8086

- As shown in Fig. the execution unit (EU) has four general purpose 16-bit registers.
- Each one of them can be used for temporary storage of 8 bit data, 16-bit data or 32-bit data.
- 8-bit Registers - AH, AL, BH, BL, CH, CL, DHi, DL. Any of these registers can be used as an 8-bit operand.
- 16-bit Registers - AX, BX, CX, DX, SI, DI, SP, BP. Any of these registers can be used as a 16-bit operand.
- 32-bit Registers – DX : AX together can be used for 32-bit operand.
- These registers can be used for general purpose computing when their other specialized functions do not interfere.

- The above mentioned register pairs are referred to as follows.

Pair of general purpose registers	Referred to as
1. Pair of AH and AL	AX register
2. Pair of BH and BL	BX register
3. Pair of CH and CL	CX register
4. Pair of DH and DL	DX register

- In short, the four general-purpose 16-bit data registers : AX, BX, CX, and DX, each of these is a combination of two 8-bit registers which are separately accessible as AL, BL, CL, DL (the "low" bytes) and AH, BH, CH, and DH (the "high" bytes).



- For example, if AX contains the 16-bit number 1234H, then AL contains 34H and AH contains 12H.
- The upper and lower halves of the data registers are separately addressable. This means that each data register can be used as a single 16-bit register or as two 8-bit registers.

Special functions of general purpose registers of 8086

1. Register AX : Accumulator

AX is the "16-bit accumulator" while AL is "8-bit accumulator"

Accumulator has the following special functions :

(i) Some of the operations, such as Multiplication and Division, require that one of the operands be in the accumulator and also the result is stored in accumulator. Some other operations, such as Addition and Subtraction, may be applied to any of the registers (that is, any of the eight general- and special-purpose registers) but are more efficient when working with the accumulator.

(ii) It works as a via register for I/O accesses i.e. a data is routed through accumulator for the communication of the processor and I/O devices.

For OUT instruction the data in accumulator (AL for 8-bit data and AX for 16-bit data) can only be given to the output device.

For IN instruction the data taken from the input device can be taken only in accumulator (AL for 8-bit data and AX for 16-bit data)

(iii) It also works as a via register for string instructions. Whenever a data is to be brought from memory or given to memory in case of string operations it is routed through accumulator only.

2. Register BX : Base

BX is the "base" register;

(i) It is the only general-purpose register which may be used for indirect addressing. (various addressing modes are discussed in chapter 4.)

(ii) For example, the instruction MOV [BX], AX causes the contents of AX to be stored in the memory location whose address is given in BX.

3. Register CX : Counter

CX is the "count" register. It works as a default counter register for three instructions viz :

(i) The looping instructions (LOOP, LOOPE, and LOOPNE), to indicate the number of iterations

(ii) The shift and rotate instructions (RCL, RCR, ROL, ROR, SHL, SHR, and SAR), to indicate number of shifts or rotations (Here only CL is used and not entire CX)

(iii) The string instructions (with the prefixes REP, REPE, and REPNE) to indicate the size of the string block.

4. Register DX : Data

DX is the "data" register

(i) It is used together with AX for the word-size MUL and DIV operations, when the operand size is greater than the register AX i.e. operand is 32-bit.

(ii) It also holds the port number for the IN and OUT instructions. For 16-bit address accesses of I/O ports only DX can be used as a pointer.

Summary of Implicit use of General Purpose Registers

Registers	Operations
AX	Word multiply, Word divide, Word I/O and Word string
AL	Byte multiply, byte divide, byte I/O, byte string and decimal / ASCII arithmetic.
BX	Store address information
CX	Counter for String operations and loops
CL	Counter for Variable shift and rotate
DX	Word multiply, word divide, Indirect I/O