Programming Mode

Register Organization of 8086

8086 has a powerful set of registers containing general purpose and special purpose registers. All the registers of 8086 are 16-bit registers. The general purpose registers, can be used as either 8-bit registers or 16-bit registers. The general purpose registers are either used for holding data, variables and intermediate results temporarily or for other purposes like a counter or for storing offset address for some particular addressing modes etc. The special purpose registers are used as segment registers, pointers, index registers or as offset storage registers for particular addressing modes. We will categorize the register set into four groups, as follows:

General Data Register

Figure 3.1 shows the register organization of 8086. The registers AX, BX, CX and DX are the general purpose 16-bit registers. AX is used as 16-bit accumulator, with the lower 8-bits of AX designated as AL and higher 8-bits as AH. AL can be used as an 8-bit accumulator for 8-bit operations. This is the most important general purpose register having multiple functions, which will be discussed later.

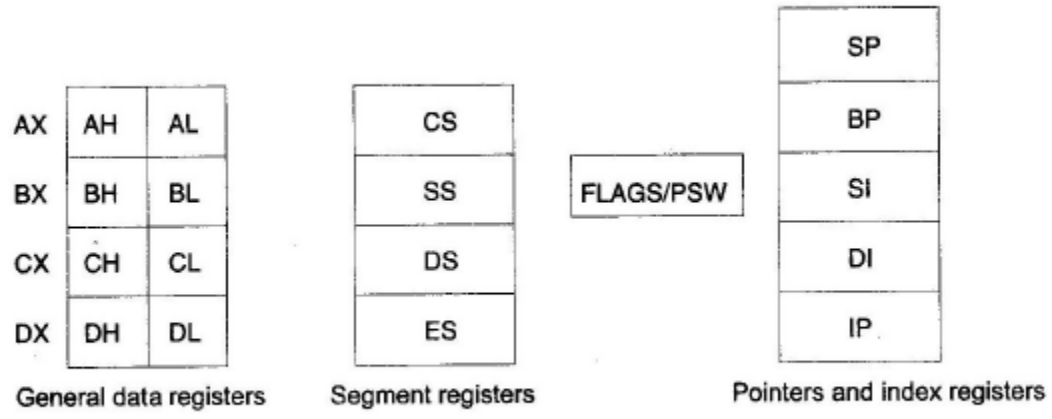

Figure 3.1: Register organisation of 8086

Usually the letters L and H specify the lower and higher bytes of a particular register. For example, CH means the higher 8-bits of the CX register and CL means the lower 8-bits of the CX register. The letter X is used to specify the complete 16-bit register. The register CX is also used as a default counter in case of string and loop instructions. The register BX is used as offset storage for forming physical addresses in case of certain addressing modes. DX register is a general-purpose register

which may be used as an implicit operand or destination in case of a few instructions. The detailed uses of these registers will be more clear when we discuss the addressing modes and the instruction set of 8086.

Segment Register

Unlike 8085, the 8086 addresses a segmented memory. The complete 1 megabyte memory, which the 8086 is able to address, is divided into 16 logical segments. 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). The code segment register is used for addressing a memory location in the code segment of the memory, where the executable program is stored. Similarly, the data segment register points to the data segment of the memory, where the data is resided. The extra segment also refers to a segment which essentially is another data segment of the memory. Thus the extra segment also contains data. The stack segment register is used for addressing stack segment of memory.

The stack segment is that segment of memory which is used to store stack data. The CPU uses the stack for temporarily storing important data, e.g. the contents of the CPU registers which will be required at a later stage. The stack grows down, i.e. the data is pushed onto the stack in the memory locations with decreasing addresses. When this information will be required by the CPU, they will be popped off from the stack.

While addressing any location in the memory bank, the physical address is calculated from two parts, the first is segment address and the second is offset. The segment registers contain 16-bit segment base addresses, related to different segments. Any of the pointers and index registers or BX may contain the offset of the location to be addressed. The advantage of this scheme is that in place of maintaining a 20-bit register for a physical address, the processor just maintains two 16-bit registers which are within the word length capacity of the machine. Thus the CS, DS, SS and ES segment registers respectively contain the segment addresses for the code, data, stack and extra segments of memory. It may be noted that all these segments are the logical segments. They may or may not be physically separated. In other words, a single segment may require more than one memory chip or more than one segment may be accommodated in a single memory chip.

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 data segment while the register DI is used to store the offset of destination in data or extra segment. The index registers are particularly useful for string manipulations.

Flag Registers

The 8086's PSW contains 16 bits, but 7 of them are not used. Each bit in the PSW is called a flag. The 8086 flags are divided into the conditional flags, which reflect the result of the previous operation involving the ALU, and the control flags, which control the execution of special functions. The flags are summarized in Figure 3.2. The lower byte in the PSW corresponds to the 8-bit PSW in the 8080 and contains all of the condition flags except OF.

The condition flags are:

SF (Sign Flag): Is equal to the MSB of the result. Since in 2's complement negative numbers have a 1 in the MSB and for nonnegative numbers this bit is 0, this flag indicates whether the previous result was negative or nonnegative.

ZF (Zero Flag): I's set to 1 if the result is zero and 0 if the result is nonzero.

PF (Parity Flag): 1's set to 1 if the low-order 8 bits of the result contain an even number of 1's; otherwise it is cleared.

CF (Carry Flag): An addition causes this flag to be set if there is a carry out of the MSB, and a subtraction causes it to be set if a borrow is needed. Other instructions also affect this flag and its value will bediscussed when these instructions are defined.

AF (Auxiliary Carry Flag): Is set if there is a carry out of bit 3 during an addition or a borrow by bit 3 during a subtraction. This flag is used exclusively for BCD arithmetic.

OF (Overflow Flag): Is set if an overflow occurs, i.e., a result is out of range. More specifically, for addition this flag is set when there is a carry into the MSB and no carry out of the MSB or vice versa. For subtraction, it is set when the MSB needs a borrow and there is no borrow from the MSB, or vice versa.

As an example, if the previous instruction performed the addition

```
  0010 0011 0100 0101
+ 0011 0010 0001 1001
---------------------
  0101 0101 01011110
```

then following the instruction:

SF = 0 ZF = 0 PF = 0 CF=0 AF=0 OF=0

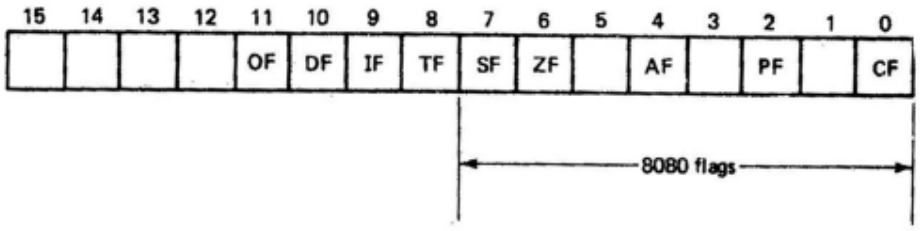

Figure 3.2: 8086's PSW.

If the previous instruction performed the addition

```
  0101 0100 0011 1001
+ 0100 0101 0110 1010
---------------------
  1001 1001 1010 0011
```

then the flags would be:

SF = 1 ZF = 0 PF = 1 CF = 0 AF = 1 OF = 1

then the flags would be:

SF = 1 ZF = 0 PF = 1 CF = 0 AF = 1 OF = 1

The control flags are:

DF (Direction Flag)—Used by string manipulation instructions. If clear, the string is processed from its beginning with the first element having the lowest address. Otherwise, the string is processed from the high address towards the low address.

IF (Interrupt Enable Flag)—If set, a certain type of interrupt (a maskable interrupt) can be recognized by the CPU; otherwise, these interrupts are ignored.

TF (Trap Flag)—If set, a trap is executed after each instruction.