Registers

Covering topics

- Data Registers
- Address Registers
- Memory Organization
- Status Registers

Registers

- Information inside the microprocessor is stored in registers .
- Registers are classified according to the functions they perform.
- In general:
  - **Data Registers:** Holds data for operations.
  - **Address Registers:** Holds address of instruction or data.
  - **Status Registers:** Keeps the current status of the processor.
- 8086 has four general ***data registers***, ***address registers*** are divided into *segment*, *pointer* and *index* registers, and the status register is called ***FLAGS*** register.
- Each register has a size of 16-bit
- There are fourteen registers, which we are going to describe as follow.

Data Registers (AX,BX,CX and DX)

- These are registers available to programmers for general data manipulation.
- As these registers 16-bit register, hence there high and low bytes data of the these registers can be accessed separately.
- The high byte of AX is called AH and the low byte is AL. Similarly the other registers have the same low and high byte.

Data Registers Functions

- AX (Accumulator Register):
  - Preferred registers use in arithmetic, logic and control transfer instructions.
  - In multiplication and division operations, one of the number(value) involved must be in AX or AL.
  - Input and output operations also require the use of AL and AX.
- BX (Base Register):
  - Serve as address register.
- CX (Count Register):
  - The CX serve as a loop counter.
  - Another example of using CX as counter is REP(repeat) instruction, which controls a special class of instruction called string operations.
  - CL is used as a counter in instructions that shift and rotate bits.
- DX (Data Register):
  - Use in multiplication and division.
  - Also used in Input and Output operations.

Segment Registers

- Store addresses of instructions and data in memory.
- These addresses are used by processor to access memory locations.
- In order to discuss the segment register let us first begin with memory organization.

Memory Organization

Memory is a collection of bytes having an address starting with 0.

The 8086 processor assigns a 20-bit physical address to its memory locations (1mb=1048,576 bytes)

000 0000 0000 0000 0000

000 0000 0000 0000 0001

000 0000 0000 0000 0010

000 0000 0000 0000 0011

We usually express it in hex digit. the highest address is FFFFFh.

0001h, 0002h, 0003h, 0004h, 0005h

Before discussing segment register, we need to understand the idea of memory segments, which is a direct consequence using a 20-bit address in a 16-bit processor. The addresses are too large to fit in 16-bit registers or word, so this problem is solved by portioning its memory into segments.

Memory Segment and Offset

- A memory segment is 2^{16} or (64KB) of consecutive memory bytes.
- Each segment is identified by a segment number which is 16-bit, starting with 0, and highest segment number is FFFFh.
- Within a segment a memory location is specified by giving an offset. This is the number of bytes from the beginning of the segment. With 64-kb segment, the offset can be given as a 16-bit number. The first bye in a segment has offset 0. The last offset in a segment is FFFFh.

Segment: Offset Address

- A memory location specified by segment : offset which is called logical address. For example 4AEF:34DC means 34DC offset with in 4AEF segment.
- To obtain 20-bit physical address the 8086 processor first shift the segment number 4 bits to left(equal to multiplying by 10h) and then add the offset. Thus physical address for A4FB:4872 is
  - **A4FB0h**
    +4872h

    A9822h

Program Segment

- A machine language program consist of ***instruction*** and ***data***, and a data structure called ***stack*** used by processor to implement procedure call.
- The program code, data and stack are loaded into code segment, data segment and stack segment respectively.
- 8086 has four segment registers to hold segment numbers (CS,DS,SS) and if a program needs to access a second data segment, it can use ES register.

Pointer and Index Registers(SP,BP,SI,DI)

- These register actually point to memory locations(contain the offset address).
- These registers can be used in arithmetic and other operations.
- **Stack Pointer(SP):** This register is used in conjunction with ***SS*** for accessing the stack segment.
- **Base Pointer(BP):** It is used primarily to access data on the stack.
- **Source Index(SI):** Used to point to memory locations in the data segment, addressed by ***DS***. By incrementing the contents of ***SI***, we can easily access consecutive memory locations.
- **Destination Index(DI):** This register perform the same function as ***SI***. There is a class of instruction called string operations, that use ***DI*** to access memory locations addressed by ***ES***.

Instruction Pointer

- To access instructions, the 8086 uses CS and IP registers.
- The IP contains offset and the CS contains segment number for the instruction.
- IP is updated each time an instruction is executed so to point to next instruction.
- Unlikely other register, IP can’t be manipulated directly by an instruction.

Status Register

- Used to indicate status of the processor.
- It does this by setting of individual bits called flags (status flags and control flags);
- Status Flags: Indicates the result of an instruction executed by the processor. For example during subtraction if result is 0, ZF is set to 1.
- Control Flags: These Flags enable or disable certain operations of the processor. For example if the IF flag is set to 0, input from the keyboard will be ignored by the processor.