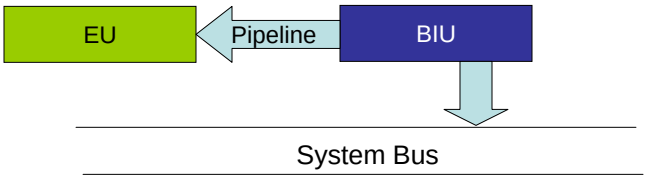
Registers in Microcomputers

Lecture # 3

8086 Architecture

- The BIU provides H/W functions, including
  - generation of the memory and I/O addresses for the transfer of data between the outside world -outside the CPU and the EU.
- The EU receives
  - program instruction codes and data from the BIU,
  - executes these instructions, and
  - store the results in the general registers.

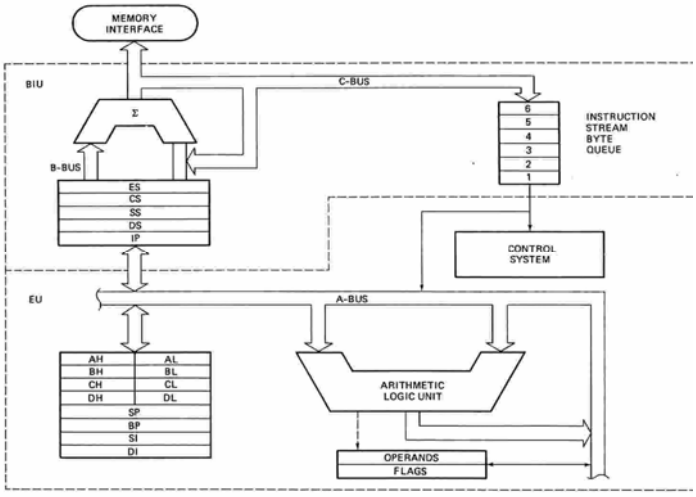


8086 Microprocessor

- 8086 is a 16 bit microprocessor
  - ALU
  - Internal registers and
  - Most of the instructions are of 16 bit binary word
- It has a 16 bit data bus- read or write 16 or 8 bit at a time to memory or to port (8bit bus for 8088)
- It has 20 bit address bus- can address 1 Mbyte of memory locations

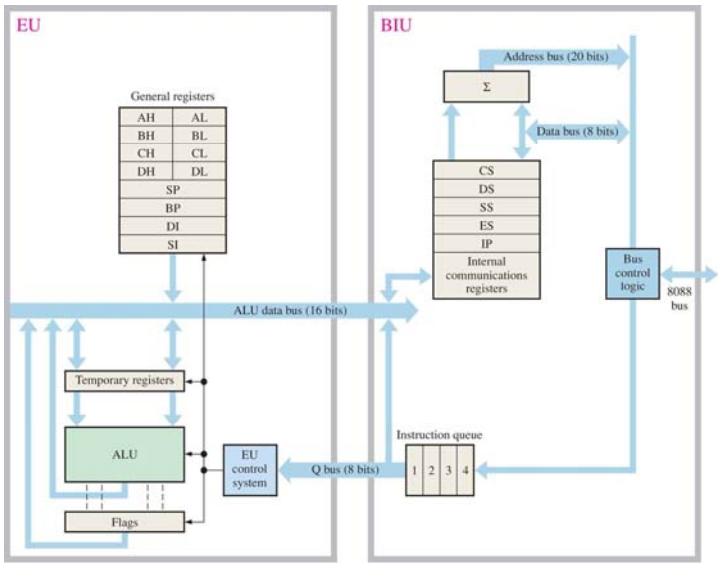
Architecture

A typical x86 Architecture

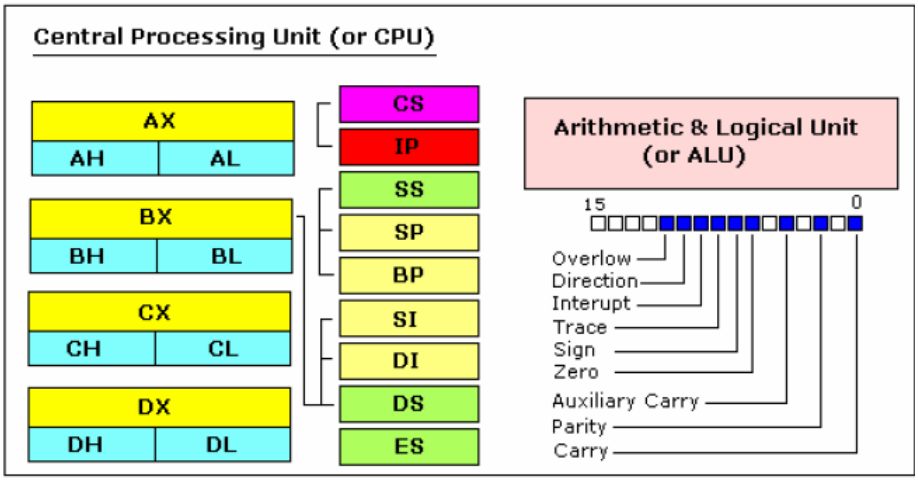


Intel 8086 Architecture, the 1st member of x86 family

Microprocessor Architecture



Inside CPU



8086 Registers

- The 8086 microprocessor has a total of **fourteen** registers that are accessible to the programmer.
  - Eight of the registers are known as **general purpose** registers located in EU, these can be used by the programmer for data manipulation.

GENERAL PURPOSE REGISTERS

- 8086 CPU has 8 general purpose registers, each register has its own name:
  - **AX** - the accumulator register (divided into **AH / AL**).
  - **BX** - the base address register (divided into **BH / BL**).
  - **CX** - the count register (divided into **CH / CL**).
  - **DX** - the data register (divided into **DH / DL**).
  - **SI** - source index register.
  - **DI** - destination index register.
  - **BP** - base pointer.
  - **SP** - stack pointer.

To access memory: **BX, SI, DI, BP are used**.
Combining these registers inside **[ ]** symbols, we can get different memory locations

Gen Purpose Registers

- EU contains eight Gen purpose registers as AH, AL, BH, CH, CL, DH and Dl
  - The first four registers are sometimes referred to as **data** registers.
    - Used individually for temporary storage of 8-bit data
    - Can be used to store 16-bit data, if grouped together as
  - AL and AH ref to as AX (Accumulator) used to hold temp results after arithmetic operation
  - BL and BH ref to as BX (Base) used to hold base address of data located in the memory
  - CL and CH ref to as CX (Count) to count certain loop instr eg shift counts
  - DL and DH ref to as DX (Data) gen purpose register holds data

8086 Registers

- The four **index** registers each of the registers is **16 bits long** i.e. can contain a 16-bit binary number can be used for
  - arithmetic operations but their use is usually concerned with the memory addressing modes of the 8086.
    - They are the **sp**, **bp**, **si** and **di** registers.
  - To index or point to memory
    - SP (Stack Pointer) used to address data in LIFO stack memory, when PUSH and POP instructions are executed
    - BP (Base Pointer) 16 bit-used to address an array of data in the stack memory
    - SI (Source index) 16 bit
    - DI (destination index) 16- bit, to hold 16 bit offset data

Pointers and index Registers

- Also general purpose registers, more often used for:
  - arithmetic operations but their use is usually concerned with the memory addressing modes of the 8086.
  - To index or point to memory
    - SP (Stack Pointer) used to address data in LIFO stack memory, when PUSH and POP instructions are executed
    - BP (Base Pointer) 16 bit-used to address the base of the stack
    - SI (Source index) 16 bit, for string and memory array copying
    - DI (destination index) 16- bit, to hold 16 bit offset

Instruction Pointer Register

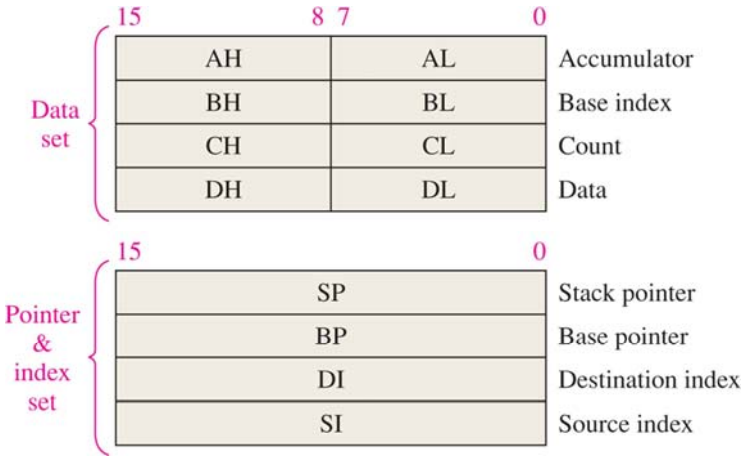
- IP (Instr Pointer) always used to address the next instruction by adding the contents of IP to code segment CS register
- This is a crucially important register
  - used to control which instruction the CPU executes.
  - The **ip**, or *program counter*, is used to store the memory location of the next instruction to be executed.
- The CPU checks the program counter to ascertain which instruction to carry out next.
  - It then updates the program counter to point to the next instruction.
  - Thus the program counter will always point to the next instruction to be executed.

8086 Registers

Segment Registers		
CS	Code Segment	16-bit number that points to the active code-segment
DS	Data Segment	16-bit number that points to the active data-segment
SS	Stack Segment	16-bit number that points to the active stack-segment
ES	Extra Segment	16-bit number that points to the active extra-segment
Pointer Registers		
IP	Instruction Pointer	16-bit number that points to the offset of the next instruction
SP	Stack Pointer	16-bit number that points to the offset that the stack is using
BP	Base Pointer	used to pass data to and from the stack
General-Purpose Registers		
AX 16 bit (AH +AL 8 bit)	Accumulator Register	mostly used for calculations and for input/output
BX (BH + BL)	Base Register	Only register that can be used as an index
CX (BH + BL)	Count Register	register used for the loop instruction
DX (BH + BL)	Data Register	input/output and used by multiply and divide
Index Registers		
SI	Source Index	used by string operations as source
DI	Destination Index	used by string operations as destination

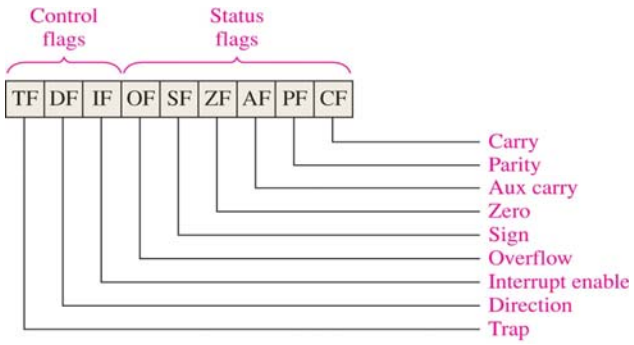
Registers in Microprocessor Architecture

- The general register set



Flag Registers

- Flag is a Flip-flop
  - which indicate some condition produced by the execution of an instruction
    - or Which controls certain operations of Execution Unit (EU)
- 16 bit flag register in EU contains 9 active flags



Find out the Flag registers functioning & location in x86 family

Flag Register

- Six flag are used to indicted some condition produced by an instruction
- These are set and reset by EU on the bases of the result of some arithmetic and logic operation
- These are:-
  - Carry Flag (CF)
  - Parity Flag (PF)
  - Auxiliary Carry Flag (AF)
  - Zero Flag (ZF)
  - Sign Flag (SF)
  - Overflow Flag (OF)

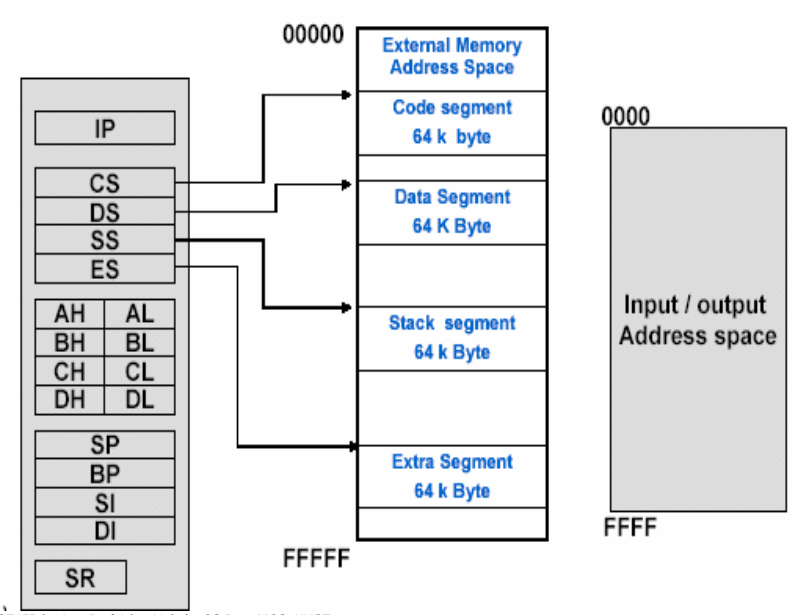
Flag Register

- Remaining three flags are used to control certain operations of the processor
- These are deliberately set and reset flags with specific instructions through program
  - Trap Flag (TF), used for single stepping through program
  - Interrupt Flag (IF), used to allow or block interrupts
  - Direction Flag (DF), used with string instructions

Segment Registers

- These are unique to x86 microprocessor family, and are designed because
  - All Index and pointer registers are of 16 bit wide, and the memory is of 20-bit (1 M)
  - Requires 20 bit address
- Index and pointer registers are not wide enough to address directly to any memory location
  - To resolve memory is segmented
- Segment of memory
  - A block of 64 Kbytes of memory addressed by a special register called a “segment register”

Software Model of 8086



Segment Registers

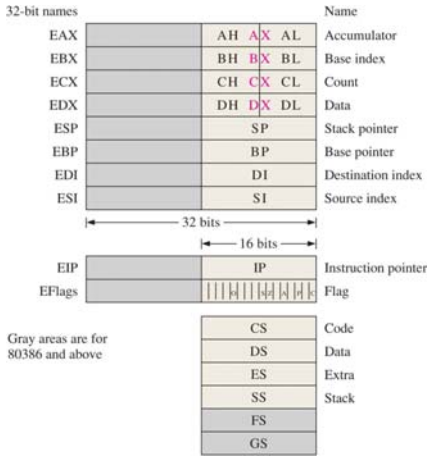
- Within the 1 MB of memory space the 8086/88 defines four 64K-byte memory blocks called the code segment, stack segment, data segment, and extra segment.
  - Each of these blocks of memory is used differently by the processor.
    - The code segment holds the program instruction codes.
    - The data segment stores data for the program.
    - The extra segment is an extra data segment (often used for shared data).
    - The stack segment is used to store interrupt and subroutine return addresses.

Segment Registers - CS, DS, SS and ES

- Store the memory addresses of instructions and data
- Memory Organization
  - Each byte in memory has a 20 bit address starting with 0 to 2^{20}-1 or 1 MB of addressable memory
  - Addresses are expressed as 5 hex digits from 00000 - FFFFF
  - Problem: 20 bit addresses are TOO BIG to fit in 16 bit registers!
  - Solution: Memory Segment
    - Block of 64K (65,536) consecutive memory bytes
      - A segment number is a 16 bit number
      - Segment numbers range from 0000 to FFFF
      - Within a segment, a particular memory location is specified with an offset An offset also ranges from 0000 to FFFF

Microprocessor Architecture

- Registers for the Intel processors from 8086/8088 through Pentium



Instruction fetch and execute

- At the beginning of each instruction cycle
  - Processor fetches an instruction from the memory
    - A register “PC”- program counter holds the address of the instruction to be fetched next
    - Processor always increment the PC after each instruction unless told otherwise
  - The fetched instruction is loaded into a register located in the processor – “ the instruction Register (IR)”
  - The instruction contains bits – to specify the action the processor is to take