8086 Architecture

By,

Hitha Paulson
Assistant Professor, Dept of Computer Science
LF College, Guruvayoor

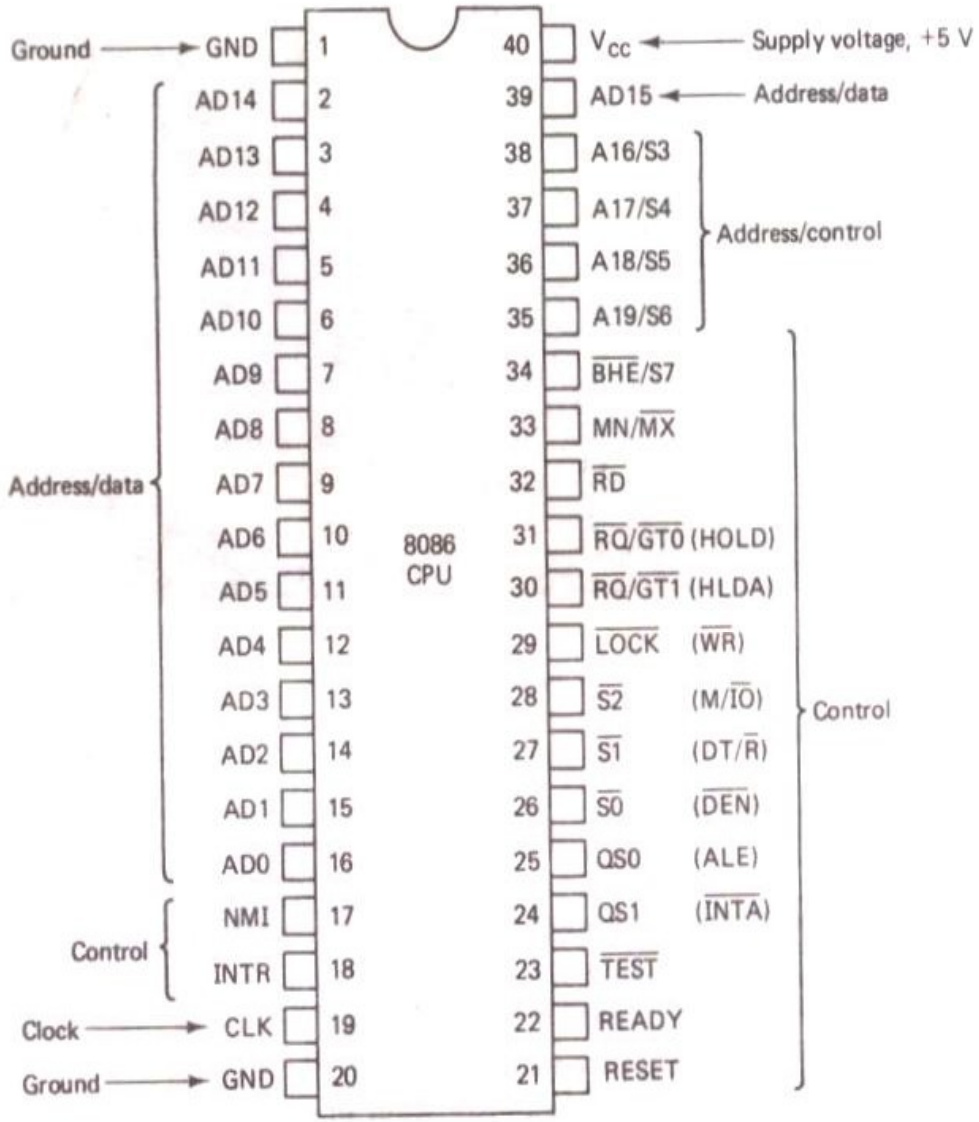
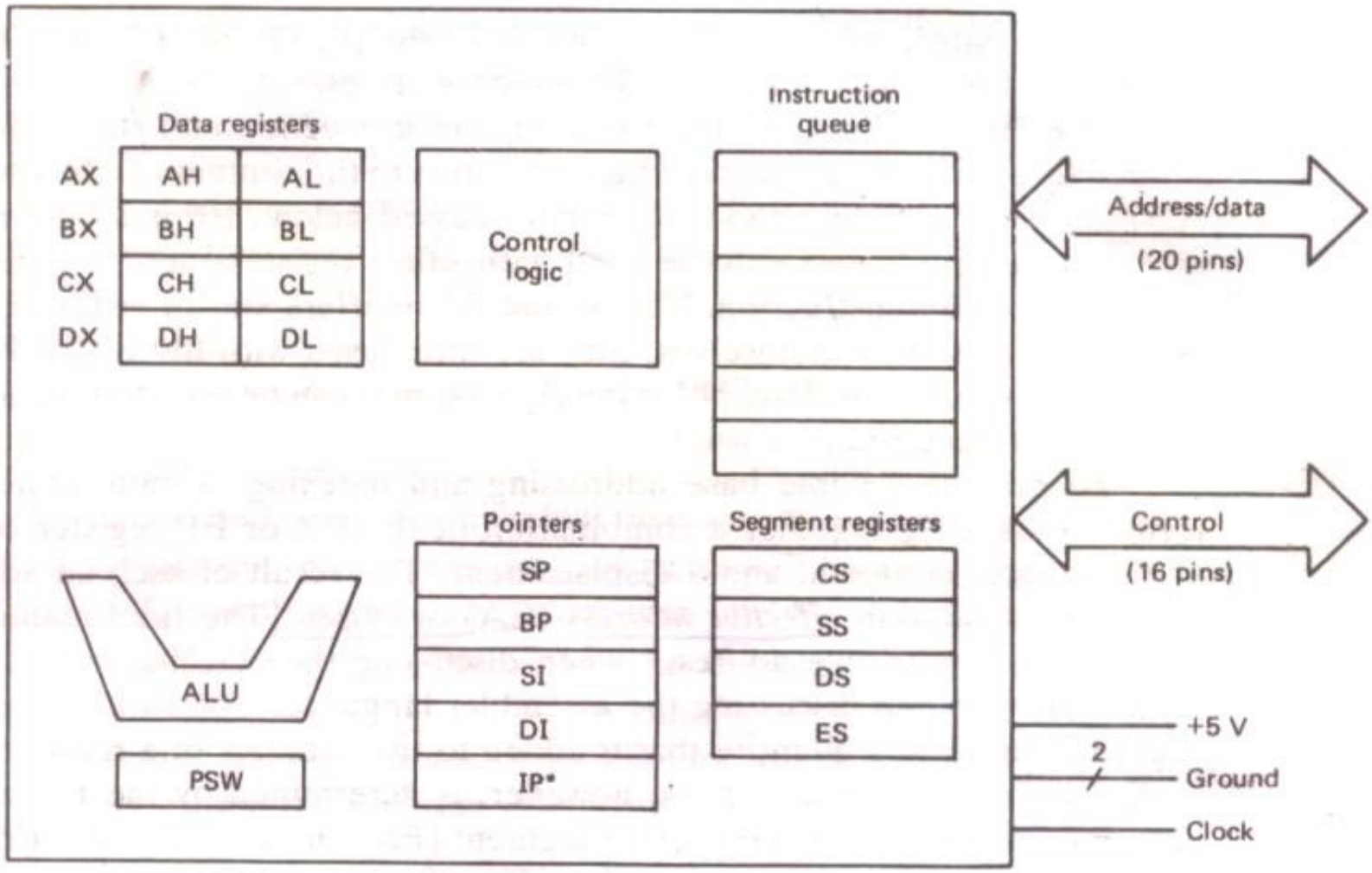


Figure 2-1 8086 pin assignments. (Reprinted by permission of Intel Corporation. Copyright 1981.)

PIN DIAGRAM

- 16 bit processor
- 40 pins
- Approximately 29000 transistors
- 20 address pins
- Clock 5 MHz,8 MHz, 10 MHz
- 1 Vcc , 2 ground
- 16 control lines

8086 General Architecture



*For the 8086 the program counter is called the instruction pointer (IP).

Register Organisation of 8086

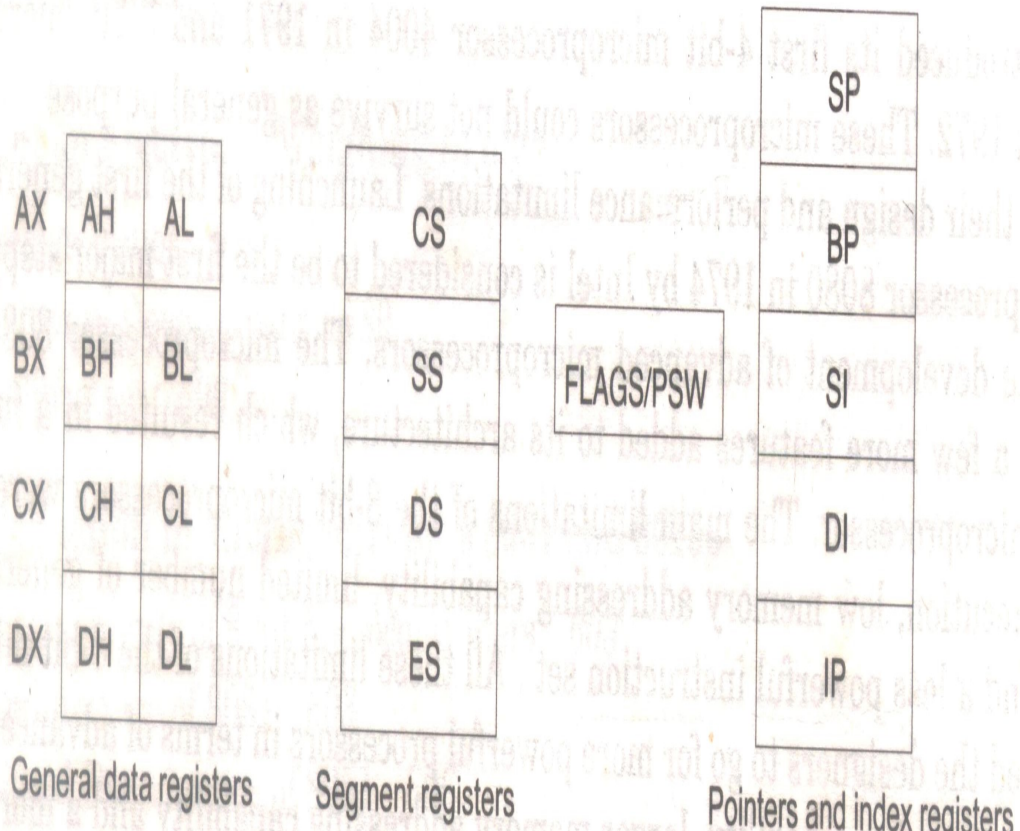
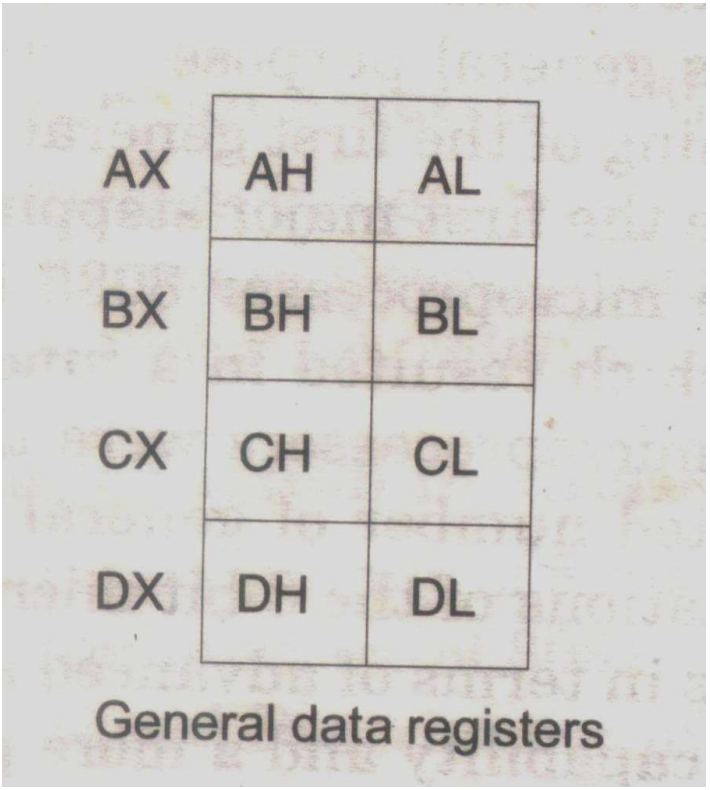


Fig. 1.1 Register organisation of 8086

- All registers are 16 bit
- General purpose :
  - Can be used as 8 or 16 bit
- Special purpose :-
  - Segment, pointers, index

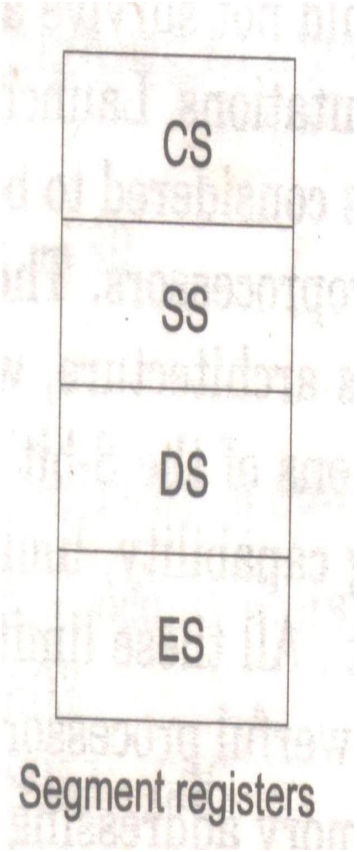
General data registers



- L & H – lower and higher byte

- AX – Accumulator
- CX – Default counter
- BX – Offset storage
- DX – Implicit operand or destination register

SEGMENT REGISTERS



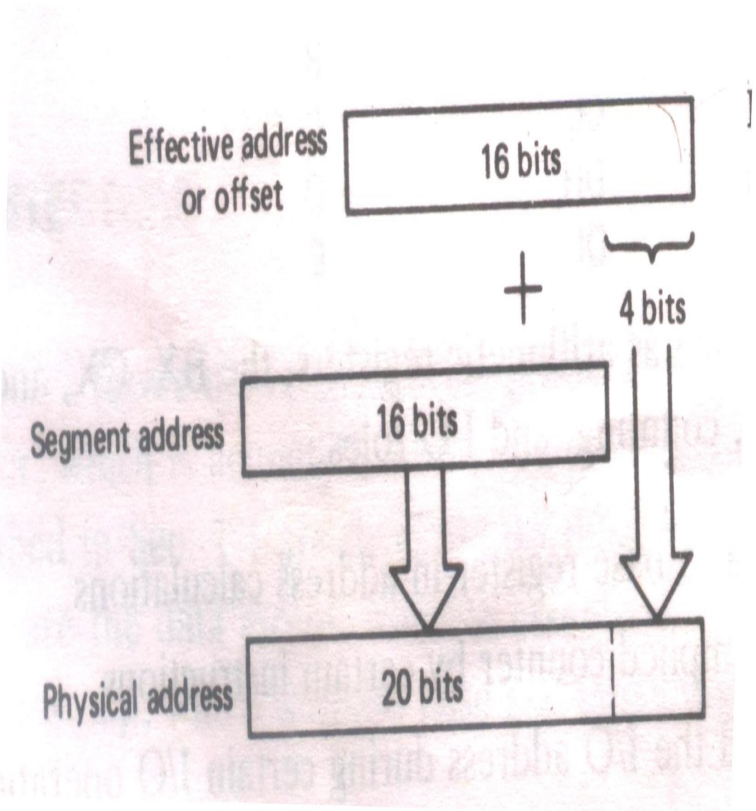
- 8086 addresses segmented memory.
- The one mega byte memory(2^{20}) that 8086 able to address is divided into 16 logical segments with 64Kbytes each
- Four segment registers are there (which stores the base addresss)
  - Code segment Register
  - Stack segment Register
  - Data segment Register
  - Extra segment Register

Memory Segments

- Code segment
  - segment where executable program is stored
- Stack segment
  - Store stack data
- Data segment
  - Segment where data is resided
- Extra segment
  - Another data segment

Physical Address calculation

Procedure



Example

Segment address ⟶ 1005H

Offset address ⟶ 5555H

Segment address ⟶ 1005H ⟶ 0001 0000 0000 0101

Shifted by 4 bit positions ⟶ 0001 0000 0000 0101 0000

+

Offset address ⟶ 0101 0101 0101 0101

Physical address ⟶ 0001 0101 0101 1010 0101

1 5 5 A 5

Where is the Offset ?

- Any of the pointers, index registers or BX contains the offset address depending upon the addressing mode

OVERLAPPING & NON OVERLAPPING SEGMENTS

- Non overlapping – separate physical address

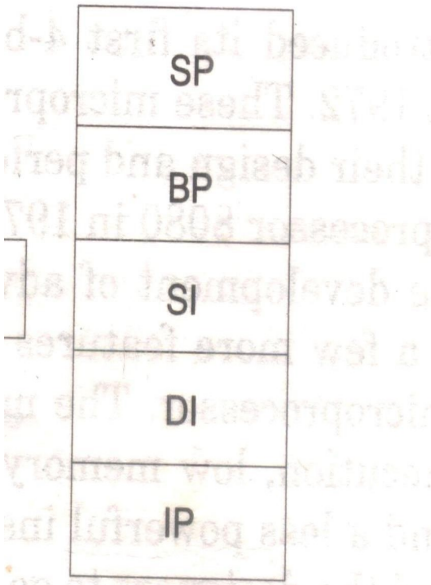
- Overlapping – same physical address with separate segment base and offset addressing

  ie CS1+IP1 = CS2 + IP2

ADVANTAGES OF SEGMENTED MEMORY

- Allow the memory capacity to be 1 mega byte even though the address associated with individual instructions are only 16 bits
- Allow the instruction , data or stack portion of a program to be more than 64 k bytes long by using more code , data or stack segment
- Facilitate the use of separate memory areas for a program , its data and the stack
- Permit a program/data to be put into different areas of memory each time the program is executed (relocation)

POINTERS AND INDEX REGISTERS

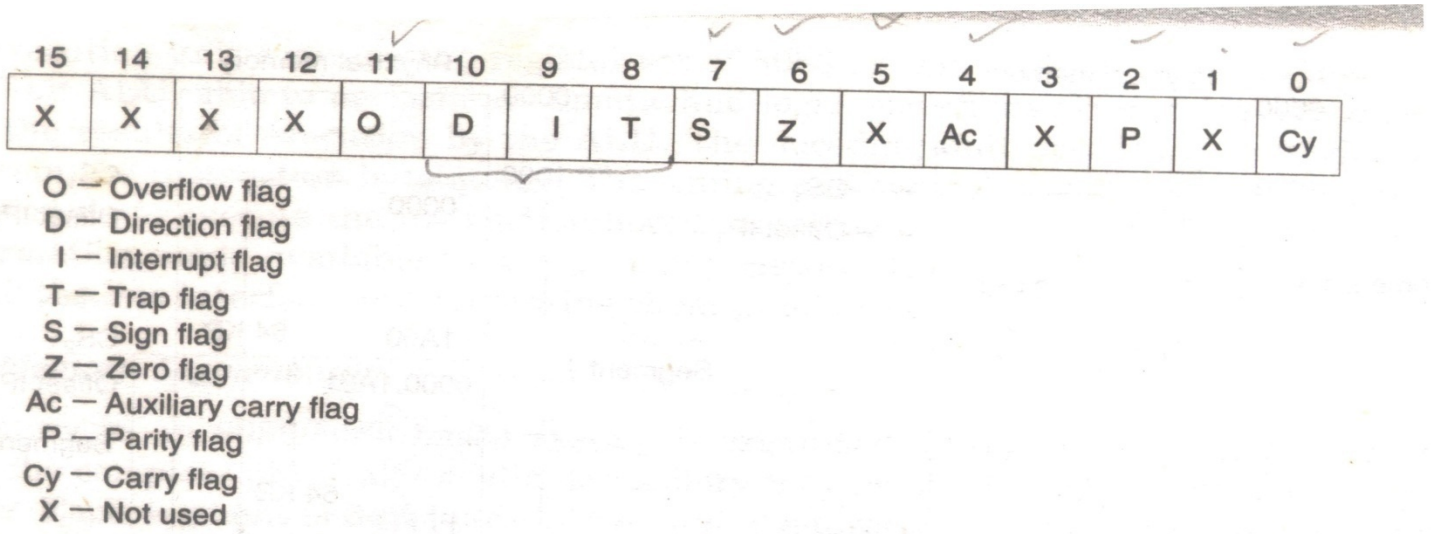


Pointers and index registers

- IP,BP,SP offset of code ,data and stack segment
- SI – to store the offset of source data in the data segment
- DI - to store the offset of destination data in the data or extra segment

FLAG OR PSW

- Content indicates the result of computations in the ALU. It also contains some flag bits to control the CPU operation.



FLAGS

- Lower byte of the flag with overflow flag =

  condition code flag register

- Higher byte = **control flag register**

FLAGS - DESCRIPTION

- S – Sign flag
  - Set when any computation result is negative (MSB bit)
- Z – Zero Flag
  - Set when computation result is zero
- P – Parity flag
  - Set when the lower byte of the result contains even number of ones
- C – Carry flag
  - Set when there is carry out of MSB in case of addition or borrow in case of subtraction
- T – Trap flag
  - If this flag is set the processor enters the single step execution mode ie a trap interrupt is generated.

FLAGS - DESCRIPTION

- I – Interrupt Flag
  - If this flag is set the maskable interrupts are recognized by CPU
- D – Direction Flag
  - If this is zero, string is processed from lowest to highest address (auto incrementing mode)
- AC – Auxiliary Carry flag
  - Set if there is a carry from the lowest nibble
- O – Overflow Flag
  - Set if there is overflow, ie.. If the result of the signed operation is large enough to be accommodated in the destination register

6 byte Instruction queue

- The 6 byte queue is continually being filled whenever the system bus is not needed for some other operation
- This look ahead feature can significantly increase the CPU throughput
- If a branch is taken then the instruction queue is flushed out.
- Instruction Pipelining

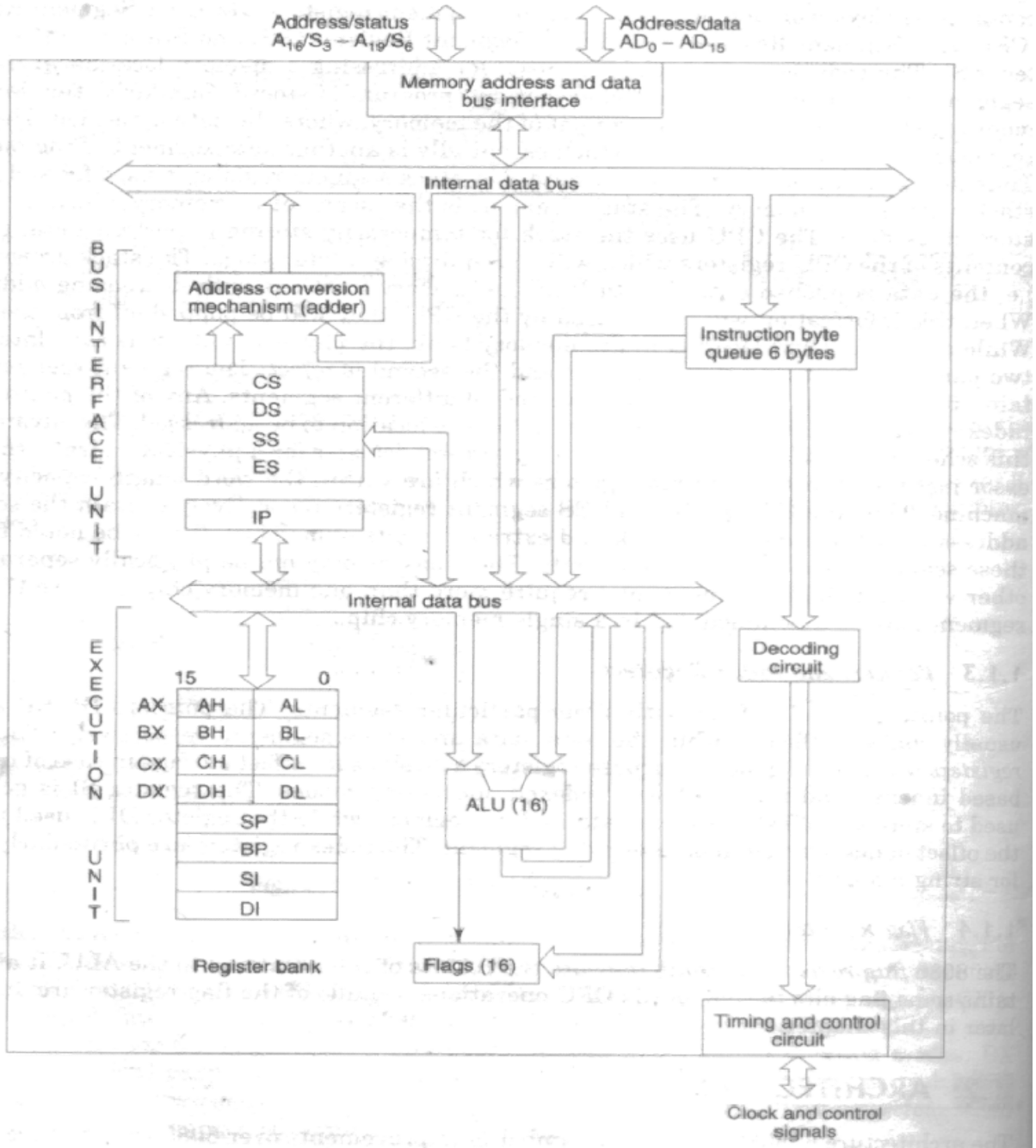


Fig. 1.2 *8086 Architecture*