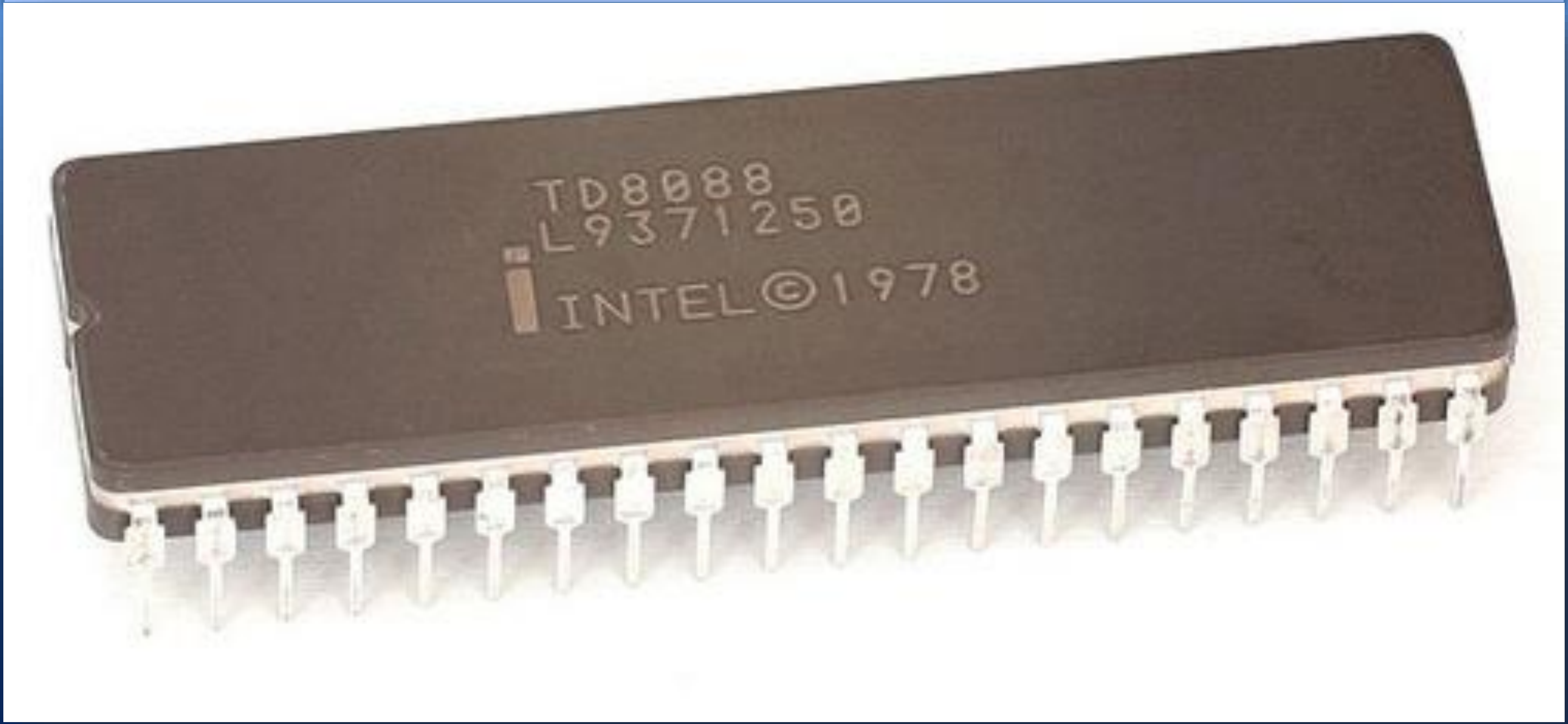
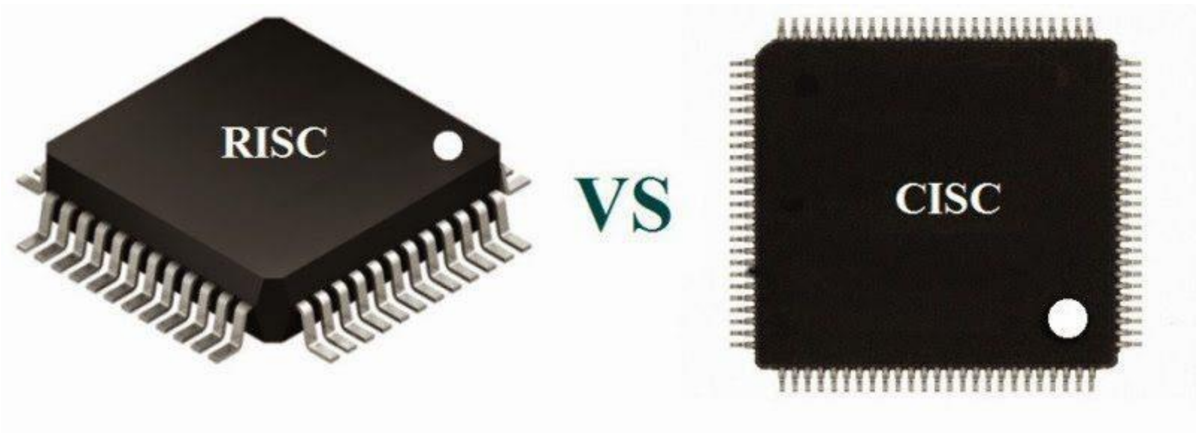
The Intel 8088 Architecture





- **CISC** Technology
  - Complex Instruction Set Computing
  - Conventional computers
  - Many of the instructions are not used

- **RISC** Technology
  - Reduced Instruction Set Computing
  - Small subset of instructions
  - Increases speed
  - Programs with few complex instructions
    - Graphics
    - Engineering

COMPARISON

CISC

- Richer instruction set, some simple, some very complex.
- Instructions generally take more than 1 clock to execute.
- Instructions of a variable size.
- Instructions interface with memory in multiple mechanisms with complex addressing modes.
- No pipelining.
- Upward compatibility within a family.
- Microcode control.
- Work well with simpler compiler.

RISC

- Simple primitive instructions and addressing modes.
- Instructions execute in one clock cycle.
- Uniformed length instructions and fixed instruction format.
- Instructions interface with memory via fixed mechanisms.
- Pipelining.
- Instruction set is orthogonal (little overlapping of instruction functionality)
- Hardwired control.
- Complexity pushed to the compiler.

CISC vs. RISC

- **CISC-like**
  - Operate on memory directly

```
L18:
   incl %eax
   cmpl $0,(%edx,%eax,4)
   jne L18
```

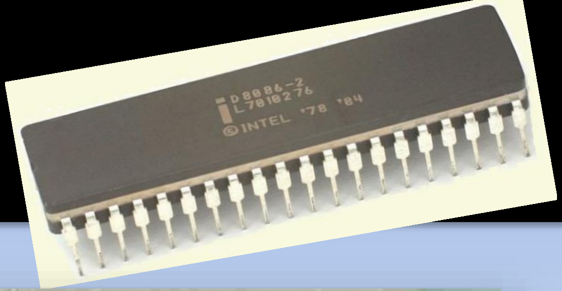
  - Smaller code size

- **RISC-like**
  - Can only operate on registers
  - Load-store architecture

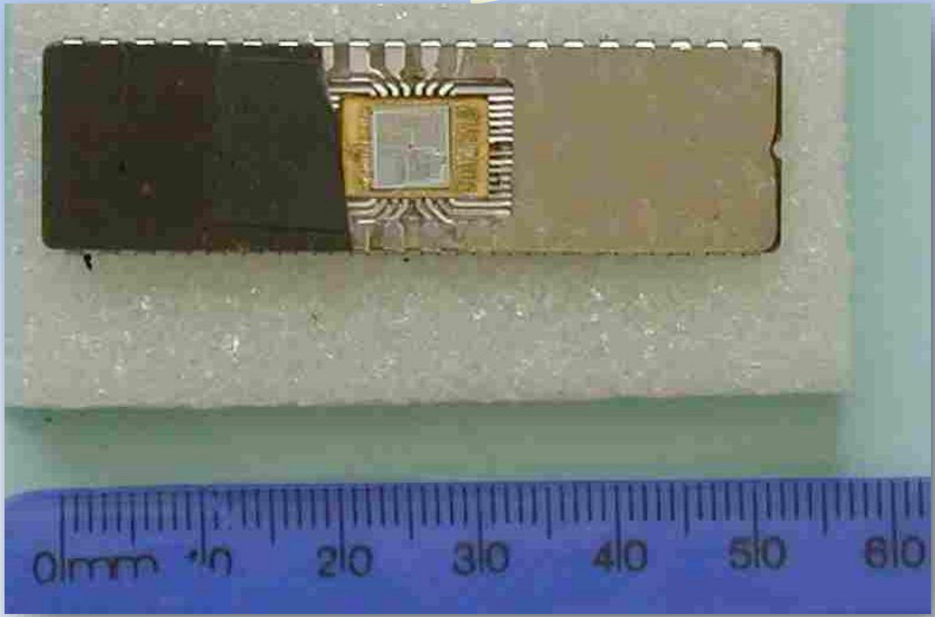
```
L5:
   movl (%edx),%eax
   incl %ecx
   addl $4,%edx
   testl %eax,%eax
   jne L5
```

  - Larger code size

Intel 8086 Microprocessor



Key Features:



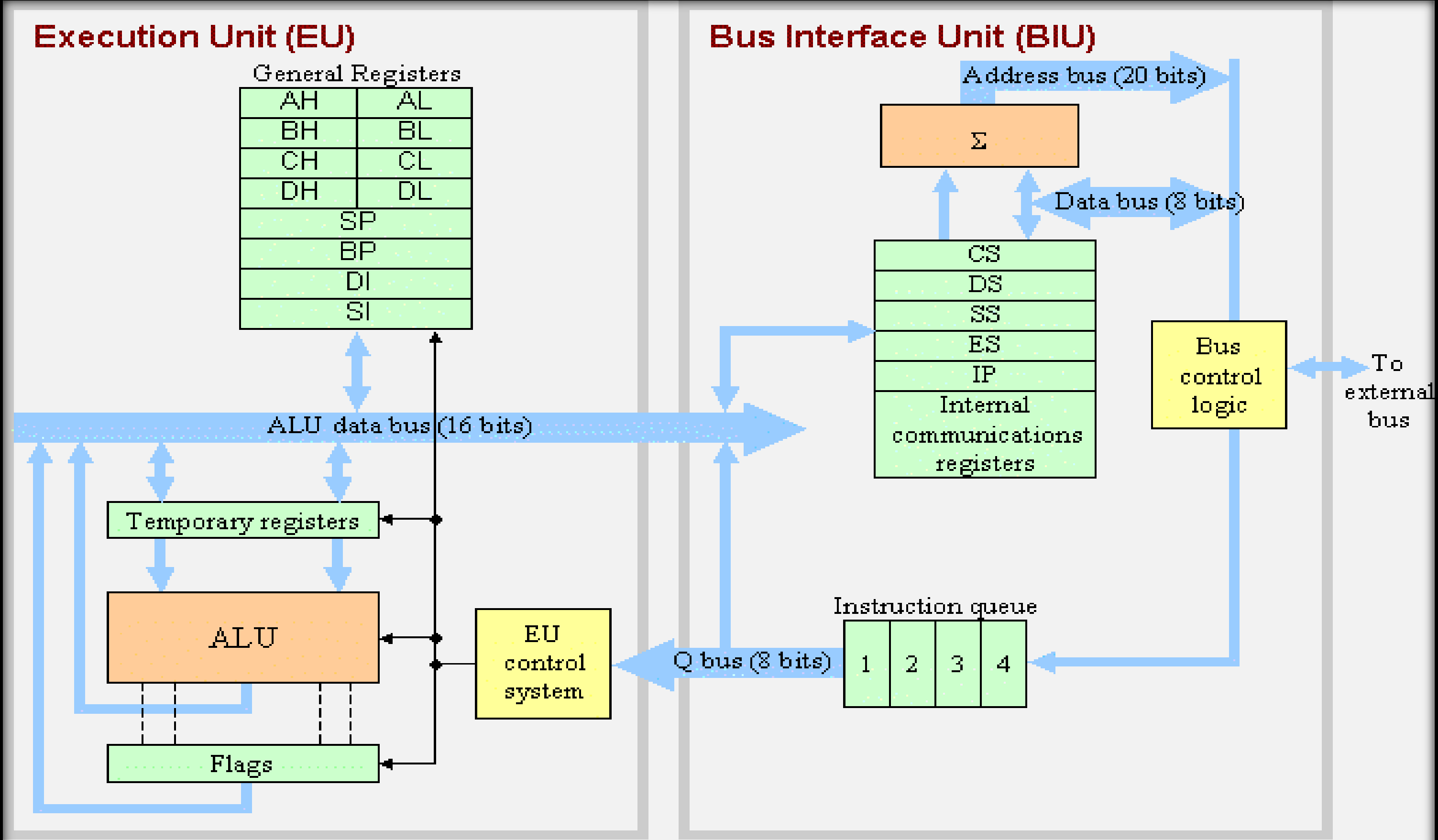
- Released by Intel in 1978
- Produced from 1978 to 1990s
- A 16-bit microprocessor chip.
- Max. CPU clock rate :
  5 MHz to 10 MHz
- Instruction set: x86-16
- Package: 40 pin DIP
- The 8086 gave rise to the x86 architecture of Intel's future processors

Common manufacturers:

Intel, AMD, NEC, Fujitsu, OKI, Siemens AG, Texas Instruments, Mitsubishi.

The Intel 8088, released in 1979, was a slightly modified chip with an external 8-bit data bus and is notable as the processor used in the original IBM PC.

The Basic Architecture of Intel 8088



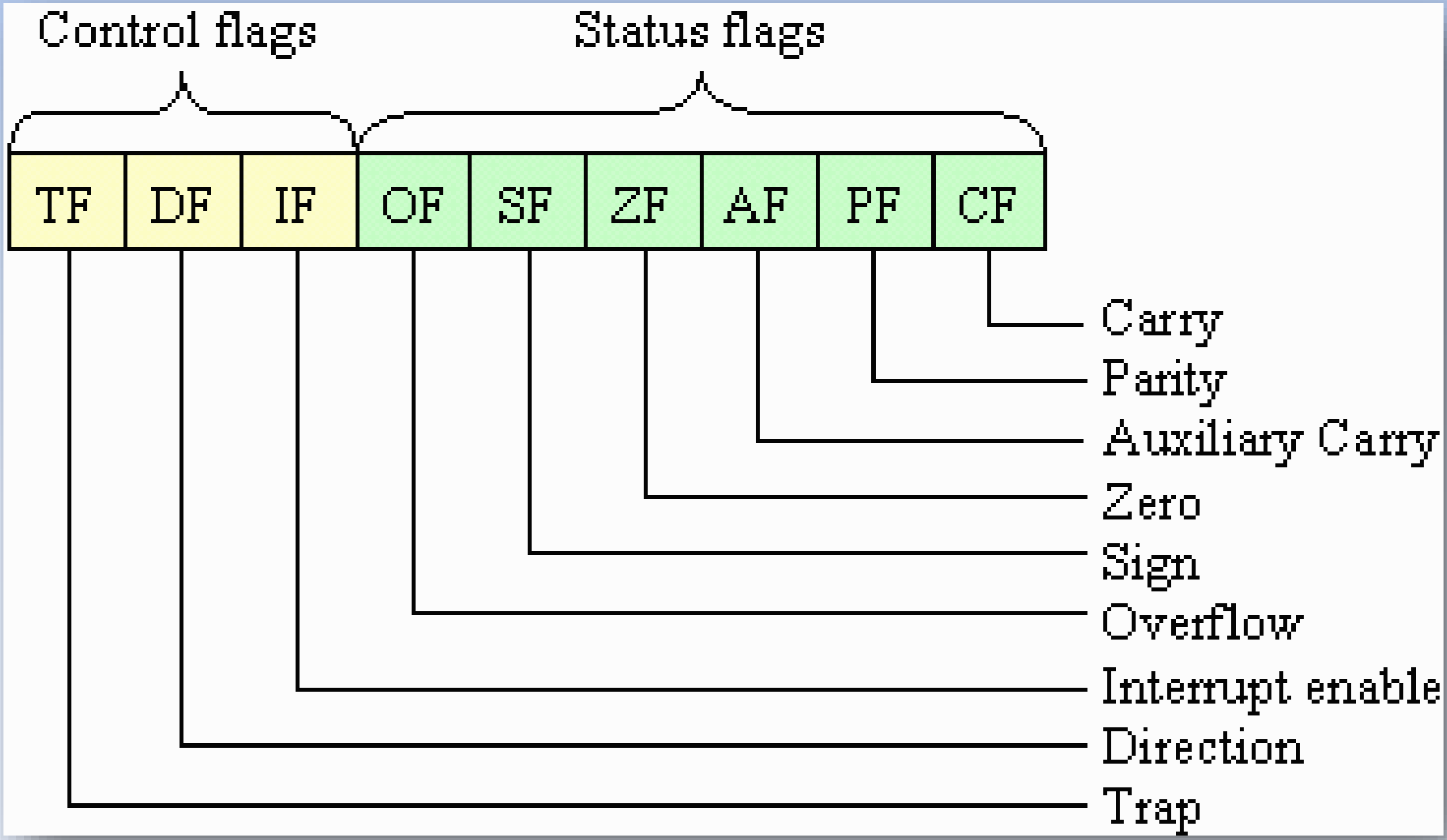
🟩 -- Static registers (groups of D Flip-Flops) used to hold or transfer binary data

🟧 -- Logic gate circuits designed to perform arithmetic or logical functions

🟨 -- Logic gate circuits designed to provide internal control to processor

🟦 -- Internal data busses used to pass information between components

The Flags



The group of flags in the figure identified as **control flags** are used to control how the processor runs. These are typically **controlled by the user's software**.
The group of green flags are **status flags** are usually set by the previous operation.

The Flags - Example

If you were to add the binary number 10110101_2 and 10010110_2, how would the flags be set?

First, let's add the two numbers to see what the result is.

$$\begin{array}{r} 1\,0\,1\,1\,0\,1\,0\,1 \\ +\,1\,0\,0\,1\,0\,1\,1\,0 \\ \hline 1\,0\,1\,0\,0\,1\,0\,1\,1 \end{array}$$

Now just go from left to right through the status flags.

- OF=1 -- There was an overflow, i.e., adding two negative numbers resulted in a positive number.
- SF=0 -- The result is positive.
- ZF=0 -- The result does not equal zero.
- AF=0 -- For now we won't worry about the auxiliary flag.
- PF=0 -- For now we won't worry about the parity flag.
- CF=1 -- There was a carry.

ALU, Execution Unit (EU), IP

- **Arithmetic Logic Unit**

is the **computation portion of the EU**. Any time arithmetic or logic needs to be performed on numbers, the numbers are sent from the general registers to the ALU, the ALU performs the function, and the result is sent back to the general registers.

- **EU Control System**

is a set of gates that control the timing, passing of data, and other items within the execution unit.

- **Instruction Pointer**

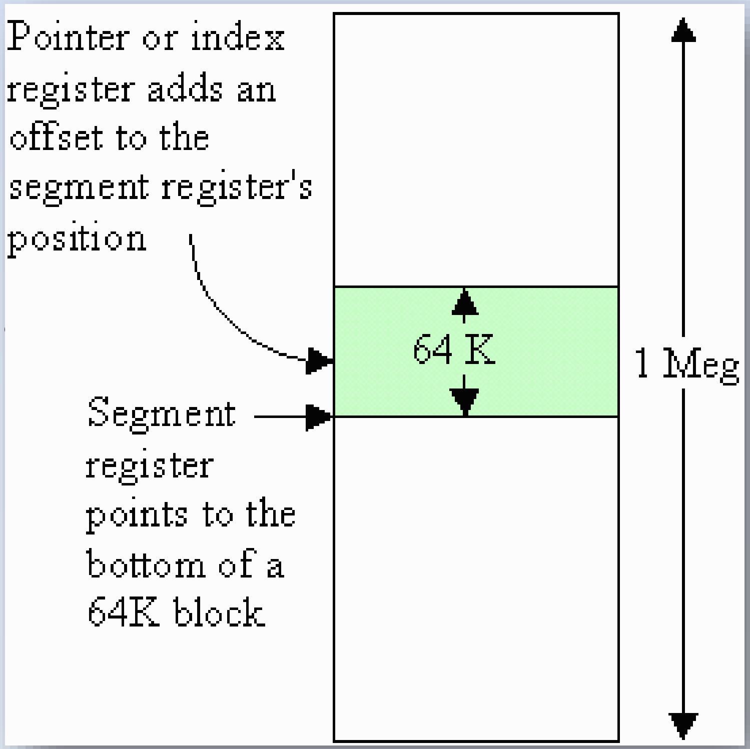
can be found toward the bottom of the group of registers in the center of the BIU. This register is an address register just like the SP, BP, DI, and SI registers in the EU. The difference is its purpose. **The IP points to the next instruction to execute from memory**.

Segment Registers

In the center of the BIU section of the processor organizational block diagram is a set of registers labeled CS, DS, SS, and ES.

These 4 registers are the segment registers & are used in conjunction with the pointer and index registers to store and retrieve items from the memory space.

Our address registers are 16 bits wide while the address space of the 8088 is 20 bits. (memory of 8088 is 2^{20} = 1MB). So how does this work? Are 4 of the address lines just ignored since we can only send 16 bits of information from our addressing registers?


Pointer or index register adds an offset to the segment register's position
64 K
1 Meg
Segment register points to the bottom of a 64K block

Segment Registers

A number that looks something like: **3241:A34E**

This number is actually the **combination of 2 registers**: **a segment register** and a **pointer or index register**. It is the combination of these 2 16-bit registers that creates the 20-bit address line.

To do this, take the value in the segment register and shift if left four places, i.e., add four zeros to the right side of the number. In our example above, 3241_{16} = 0011 0010 0100 0001_2 becomes 32410_{16} = 0011 0010 0100 0001 0000_2. This value is then added to the pointer or index register. This makes the value from our example:

0011 0010 0100 0001 0000 (hexadecimal 3 24 10)
+ 1010 0011 0100 1110 (hexadecimal A3 4E)
0011 1100 0111 0101 1110 (hexadecimal 3 C7 5E)

This computation takes place in the "Address Summing Block" located directly above the segment registers in the BIU in the organizational block diagram.

Therefore, the process of trying to access a single location in the 8088 processor's memory space takes three things:

a 16-bit segment address contained in one of the segment registers;

a 16-bit offset address contained in a pointer or index register; and

a 20-bit physical address which is the output from the address summing block.

80286 / 80386

- 80826 has 24 bits allocated to storing the base address.
- **16 MB** Memory locations.

- 80386 & above use 32 bits for storing the base address.
- **4GB** of memory locations.

80286 has 16 bits to define offset (its internal architecture).

80386 & above have 32 bits (32 bit internal architecture)

Intel 80486 (alias i486)

- introduced in 1989
- tightly pipelined x86 design
- the first x86 chip to use more than a million transistors, due to a large **on-chip cache** & an **integrated floating-point unit**.
- It represents a 4th generation of binary compatible CPUs since the original 8086 of 1978.
- A 50 MHz 80486 executed around 40 million instructions per second