x86 Processor Architecture

Outline

- What is x86 Processors?
- 32-bit x86 Processors
- 64-bit x86-64 Processors

What is x86 Processors?

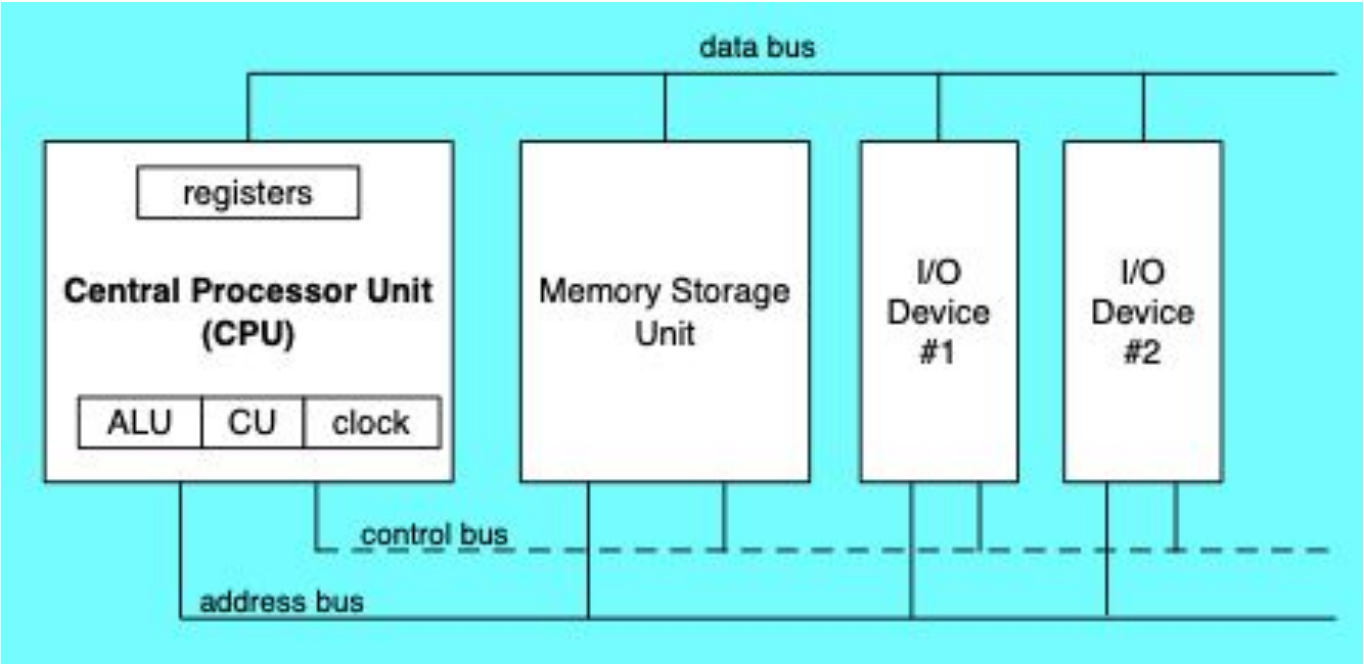
- The x86 architecture is a widely used computer processor architecture, originally launched by Intel in 1978
- The name comes from the naming of Intel's early processors, such as 8086, 80286, 80386 and 80486, among which "86" became the common identifier of this series of processors

Characteristics of x86 architecture

- **Backward compatibility**: New x86 processors can run software designed for older processors. This means that code written decades ago can still be run on modern x86 systems.
- **Widely supported**: Due to the popularity of the x86 architecture, almost all operating systems and software support this architecture. This includes major operating systems such as Windows, Linux, and macOS.

Characteristics of x86 architecture

- **Extensive hardware and software support**: There are a large number of hardware and software tools designed for the x86 architecture on the market, ensuring a good ecosystem and user support.
- **Complex instruction set**: x86 is a complex instruction set computer (CISC) whose instruction set contains a large number of instructions and features designed to complete complex tasks through a single instruction to reduce program code length and improve efficiency.

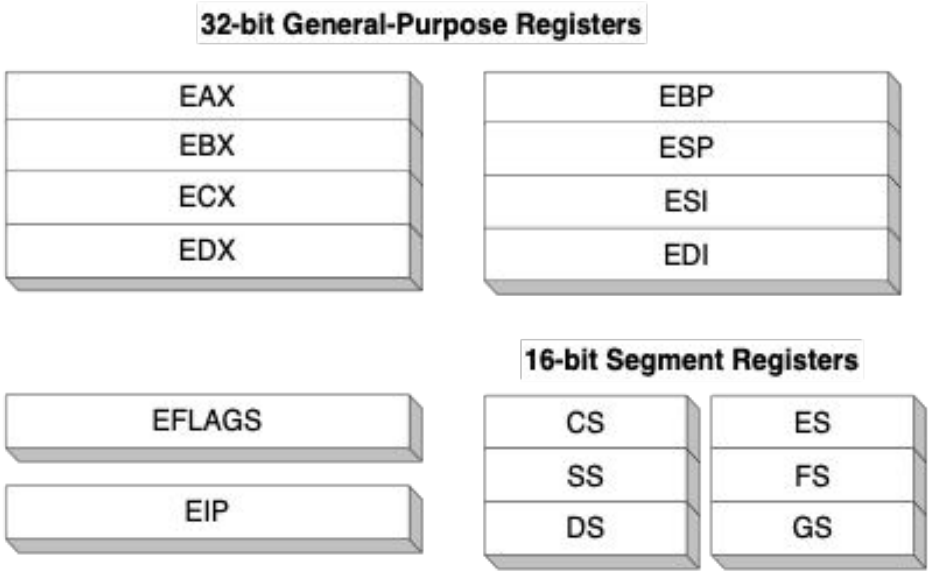

data bus
registers
Central Processor Unit
(CPU)
ALU
CU
clock
Memory Storage
Unit
I/O
Device
#1
I/O
Device
#2
control bus
address bus

Outline

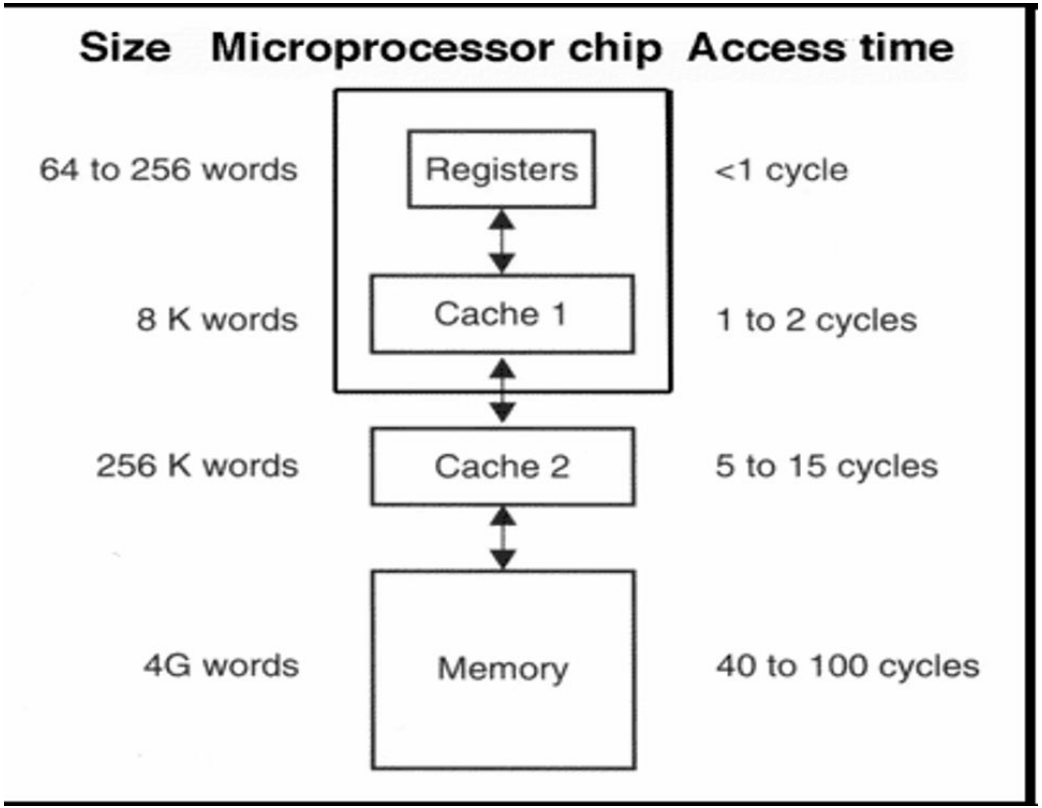
- What is x86 Processors?
- 32-bit x86 Processors
- 64-bit x86-64 Processors

Registers

- Registers are high-speed storage locations directly inside the CPU
- Intel basic program execution registers
  - 8 general-purpose registers
  - 6 segment registers
  - EFLAGS: processor status flag
  - EIP: instruction pointer

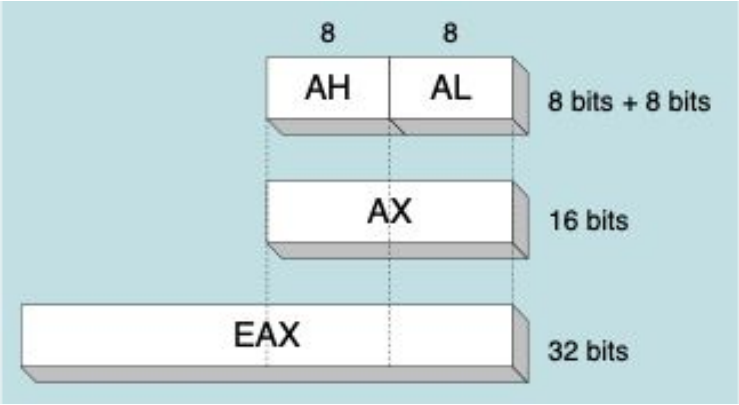


Memory Hierarchy



General-Purpose Registers

- Used for arithmetic and data movement
  - EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP
- Use 8-bit name, 16-bit name, or 32-bit name
- Applies to EAX, EBX, ECX, and EDX



32-bit	16-bit	8-bit (high)	8-bit (low)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL

Index and Base Registers

- Some registers have only a 16-bit name for their lower half:

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP

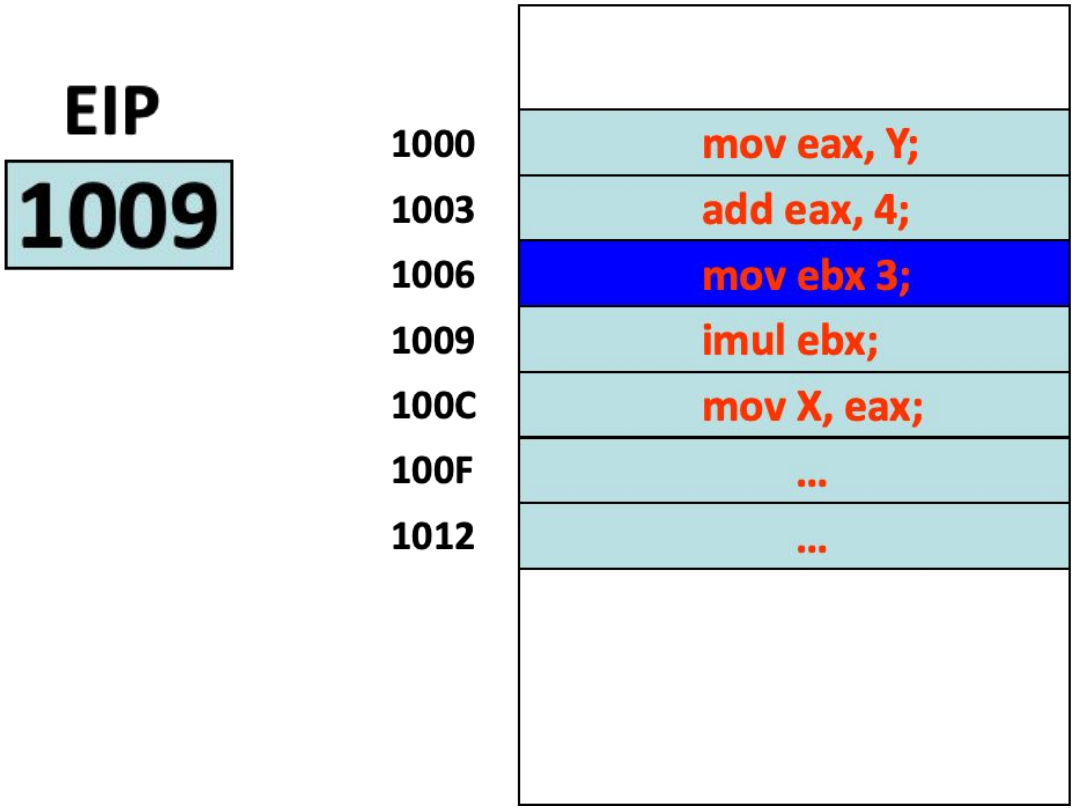
Some Specialized Register Uses

- EAX – extended accumulator register
  - Used by multiplication and division instructions
- ECX – loop counter
- ESP – stack pointer, extended stack pointer register
- ESI, EDI – index registers
  - Extended source/destination index registers
- EBP – extended frame pointer (stack)
  - Used by high-level languages to reference function parameters and local variables

Some Specialized Register Uses

- **Segment register**: as base locations for pre-assigned memory areas
  - CS (code segment): Hold instruction
  - DS (data segment): Hold global variables
  - SS (stack segment): Hold local variables and function parameters
  - ES, FS, GS: additional segments
- **EIP – instruction pointer**
  - Containing the address of the next instruction to be executed

EIP


EIP
1009
1000
mov eax, Y;
1003
add eax, 4;
1006
mov ebx 3;
1009
imul ebx;
100C
mov X, eax;
100F
...
1012
...

Some Specialized Register Uses

- EFLAGS register
  - Status and control flags
  - Each flag is a single binary bit
  - **Control Flag**
    - Control the operation of the CPU
    - Example, IF Flag (interrupt enable flag)

Some Specialized Register Uses

- EFLAGS register
  - **Status Flag**
    - Reflect the outcome of some CPU operations, ex:
    - Carry Flag (CF): unsigned arithmetic out of range
    - Overflow Flag (OF): signed arithmetic out of range
    - Sign Flag: result is negative
    - Zero Flag: result is zero
    - Auxiliary Carry Flag: carry from bit 3 to bit 4
    - Parity Flag: sum of 1 bits is an even number

EFLAGS

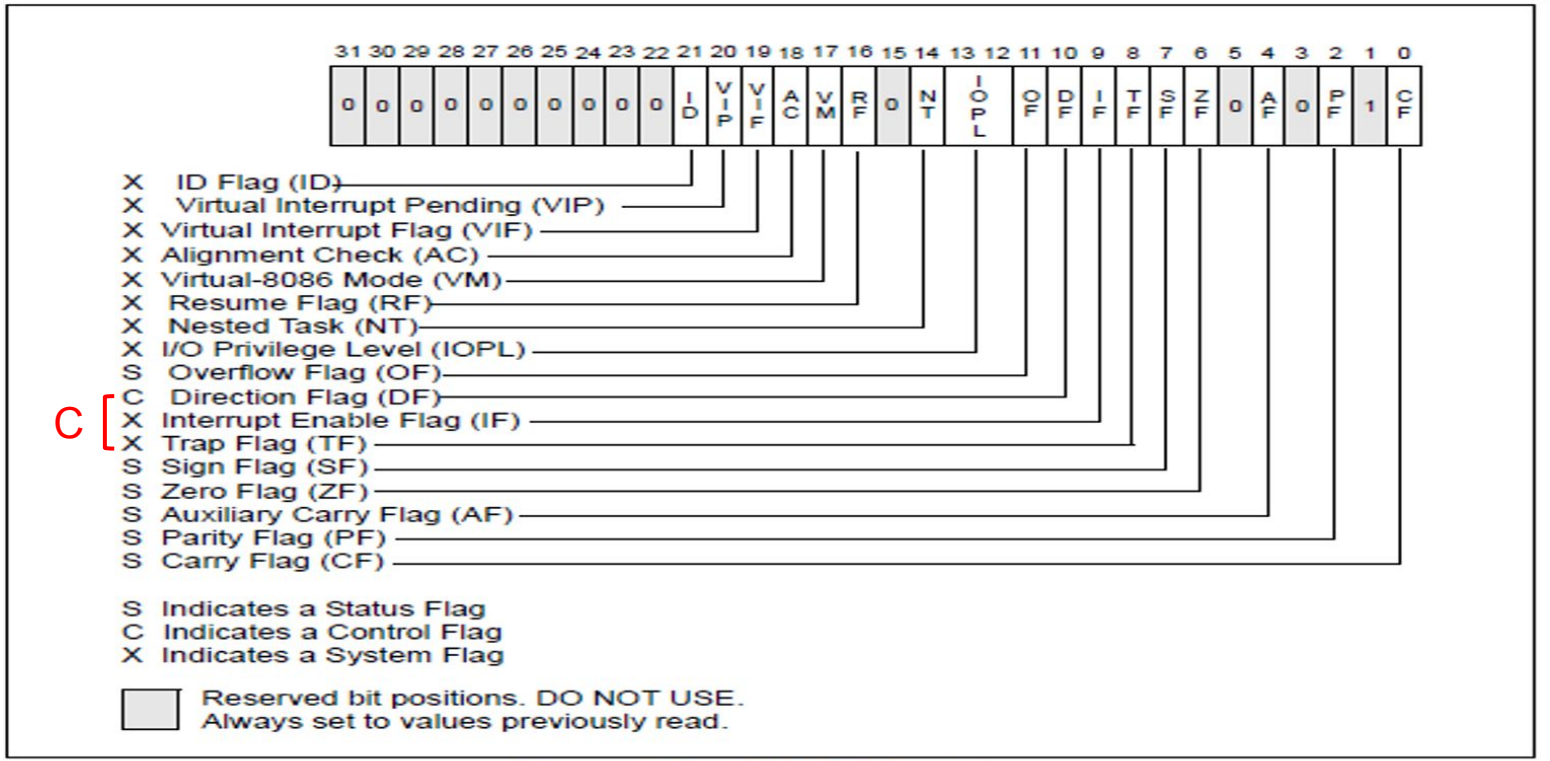


Figure 3-8. EFLAGS Register

System Registers (Skip)

- Only permit access by programs running at the highest privilege level (level 0), e.g., the Window XP
- IDTR (Interrupt Descriptor Table Register)
- GDTR (Global Descriptor Table Register)
- LDTR (Local Descriptor Table Register)
- Task Registers
- Control Registers: CR0, CR2, CR3, CR4
- Model-Specific Registers

Other Registers (Skip)

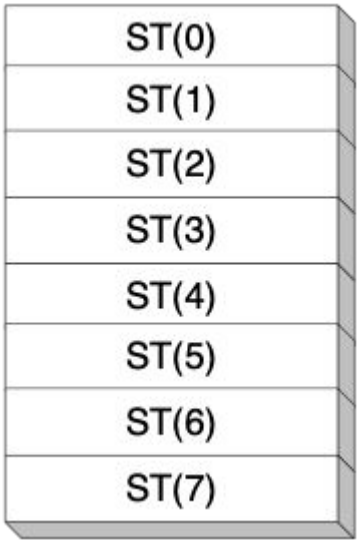
Registers for multimedia programming

- Eight 64-bit MMX register
- Eight 128-bit XMM registers
- For single-instruction multiple-data (SIMD) operations

Registers for Floating-Point Operations (Skip)

Floating-point unit:

- Eight 80-bit floating-point data registers
  - ST(0), ST(1), . . . , ST(7)
  - arranged in a stack
  - used for all floating-point arithmetic
- Two 48-bit Pointer registers
- Three 16-bit Control registers
- One opcode register



Outline

- What is x86 Processors?
- 32-bit x86 Processors
- 64-bit x86-64 Processors

64-Bit Processors

64-Bit Operation Modes

- Compatibility mode : can run existing 16-bit and 32-bit applications (Windows supports only 32-bit apps in this mode)
- 64-bit mode : Windows 64 uses this

Basic Execution Environment

- Addresses can be 64 bits (48 bits, in practice)
- 16 64-bit general purpose registers
- 64-bit instruction pointer named RIP

64-Bit General Purpose Registers

- 32-bit general purpose registers:
  - EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D, R9D, R10D, R11D, R12D, R13D, R14D, R15D
- 64-bit general purpose registers:
  - RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8, R9, R10, R11, R12, R13, R14, R15