Assembly Language for x86 Processors

- X86 Processor Architecture

CS 271 Computer Architecture

Purdue University Fort Wayne

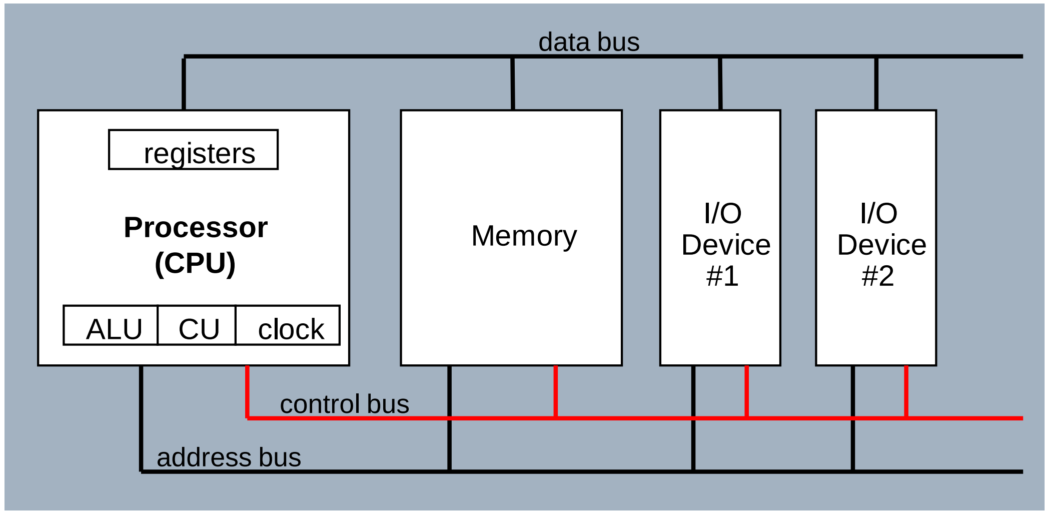
Outline

- Basic IA Computer Organization
- IA-32 Registers
- Instruction Execution Cycle

Basic IA Computer Organization

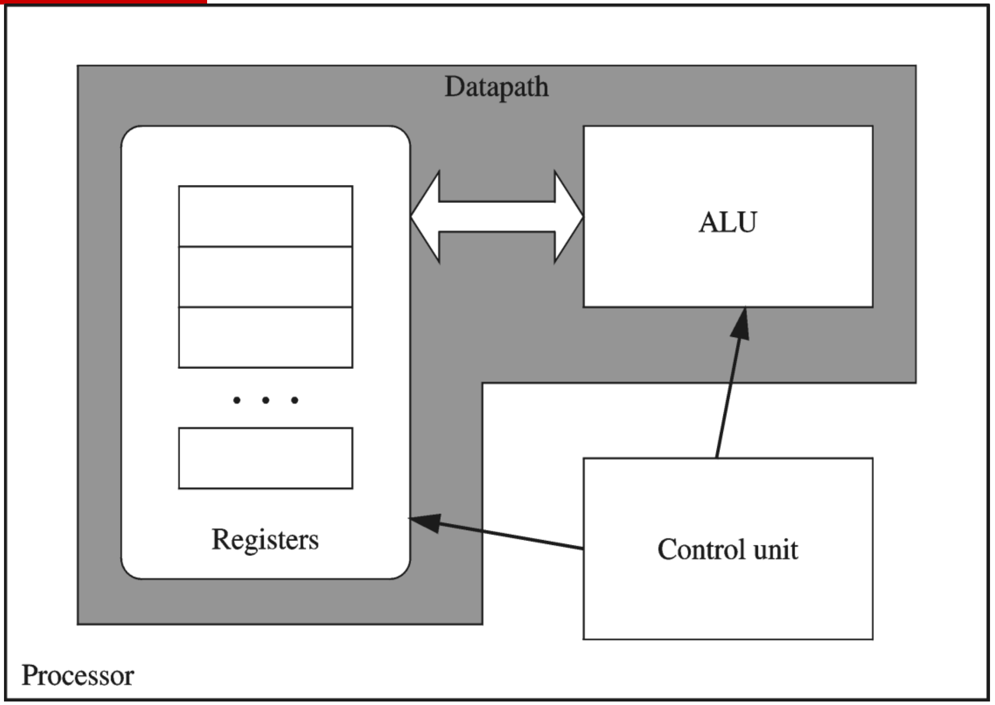
- Since the 1940's, the *Von Neumann* computers contains three key components:
  - Processor, called also the CPU (Central Processing Unit)
  - Memory and Storage Devices
  - I/O Devices
- Interconnected with one or more buses
  - Data Bus
  - Address Bus
  - Control Bus
- IA: Intel Architecture

  32-bit (or i386)

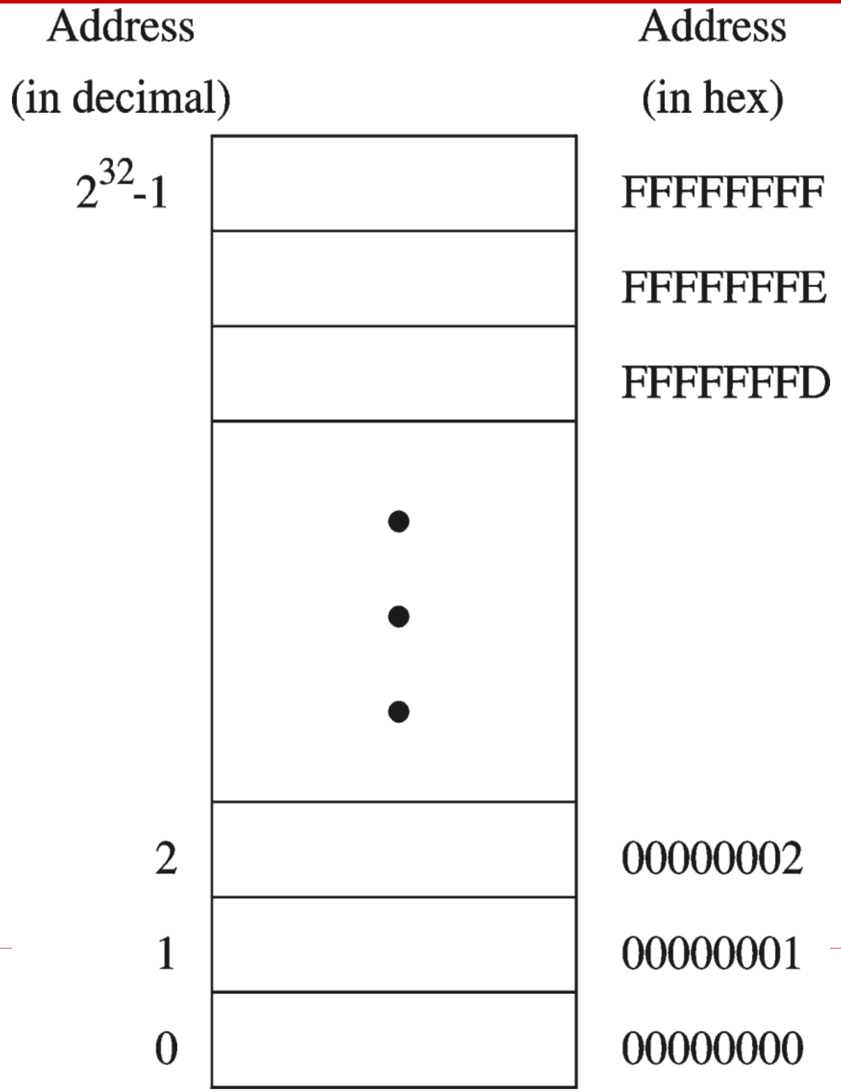


Processor

- The processor consists of
  - Datapath
    - ALU
    - Registers
  - Control unit
- ALU (Arithmetic logic unit)
  - Performs arithmetic and logic operations
- Control unit (CU)
  - Generates the control signals required to execute instructions



Memory Address Space



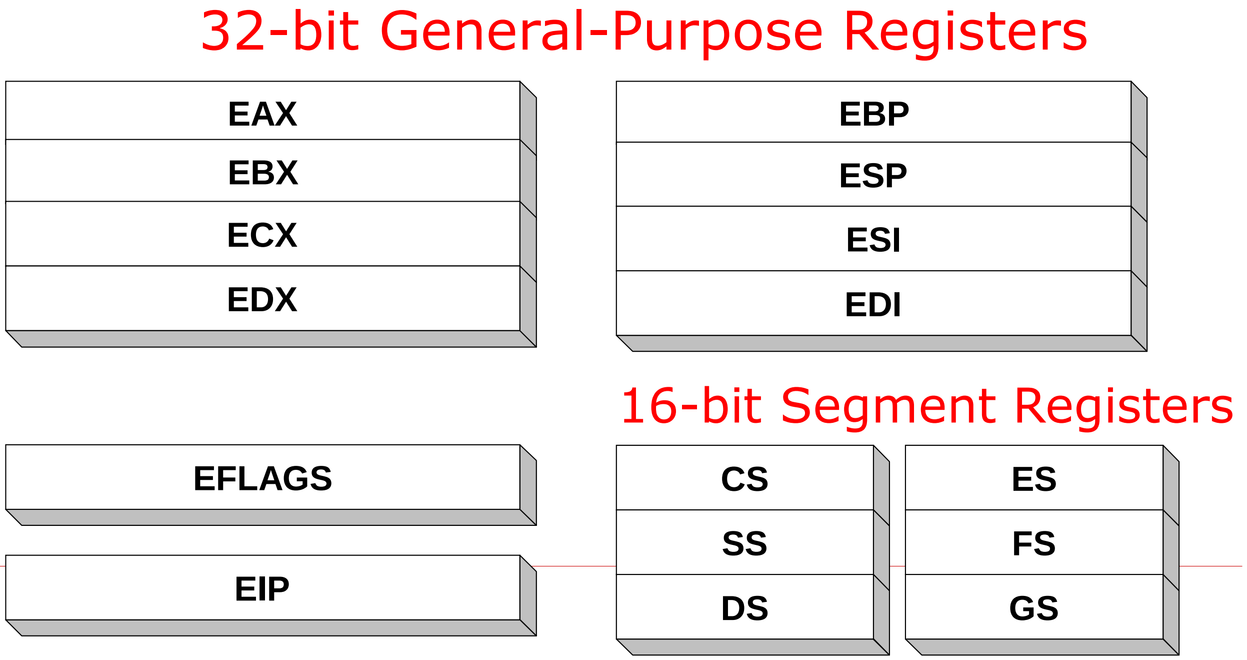
Address Space is the set of memory locations (bytes) that are addressable

Next ...

- Basic Computer Organization
- IA-32 Registers
- Instruction Execution Cycle

Registers

- Registers are high speed memory inside the CPU
  - Eight 32-bit general-purpose registers
  - Six 16-bit segment registers
  - Processor Status Flags (EFLAGS) and Instruction Pointer (EIP)



General-Purpose Registers

- Used primarily for arithmetic and data movement
  - `mov eax 10` ;move constant integer 10 into register eax
- Specialized uses of Registers
  - **eax** – Accumulator register
    - Automatically used by multiplication and division instructions
  - **ecx** – Counter register
    - Automatically used by LOOP instructions
  - **esp**– Stack Pointer register
    - Used by PUSH and POP instructions, points to top of stack
  - **esi** and **edi** – Source Index and Destination Index register
    - Used by string instructions
  - **ebp** – Base Pointer register
    - Used to reference parameters and local variables on the stack

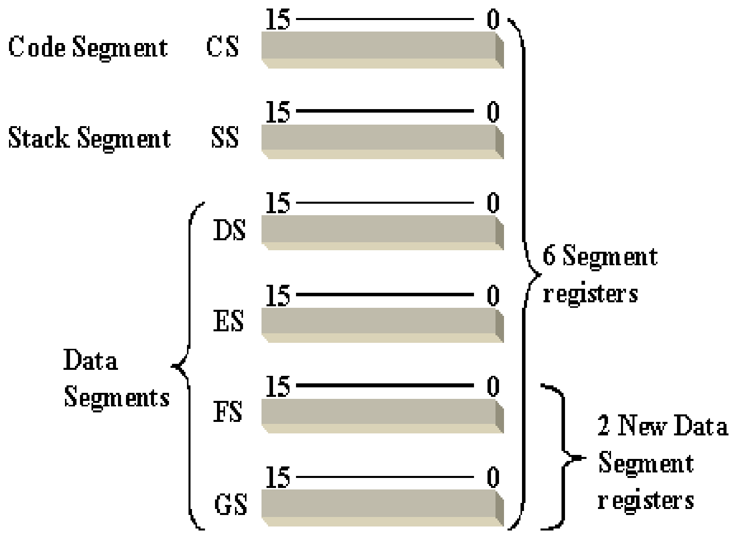
Special-Purpose & Segment Registers

- EIP = Extended Instruction Pointer
  - Contains address of next instruction to be executed
- EFLAGS = Extended Flags Register
  - Contains status and control flags
  - Each flag is a single binary bit
- Six 16-bit Segment Registers
  - Support segmented memory
  - Segments contain distinct contents
    - Code
    - Data
    - Stack

Special-Purpose & Segment Registers

=Program Counter (PC)

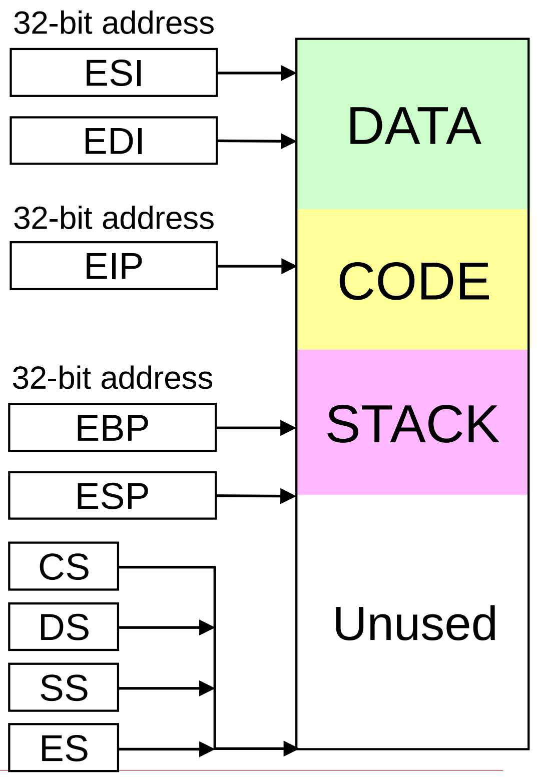
- EIP = **Extended** Instruction Pointer
  - Contains address of ***next*** instruction ***to be*** executed
- Six 16-bit Segment Registers
  - Support segmented memory
  - Segments contain distinct contents
    - **Code segment**
    - **Data segment**
    - **Stack segment**



Programmer View of Flat Memory

- Same base address for all segments
  - All segments are mapped to the same linear address space
- EIP Register
  - Points at next instruction
- ESI and EDI Registers
  - Contain data addresses
  - Used also to index arrays
- ESP and EBP Registers
  - ESP points at top of stack
  - EBP is used to address parameters and variables on the stack

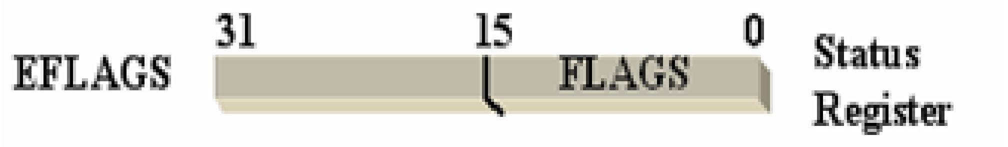
Linear address space of a program (up to 4 GB)



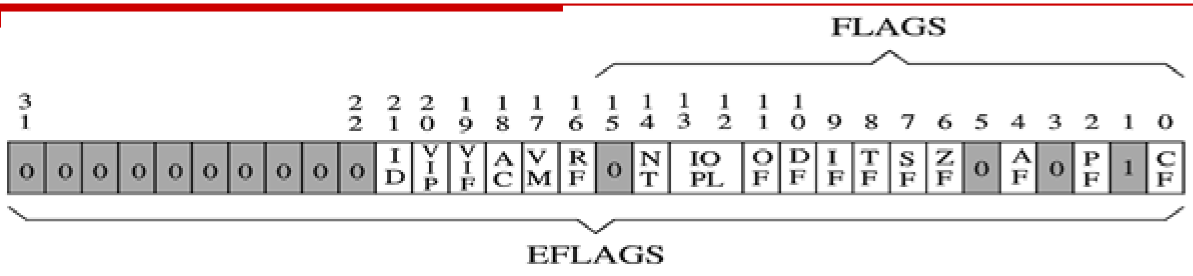
base address = 0
for all segments

Special-Purpose & Segment Registers

- For each operation that performed in CPU, there must be *a mechanism* to determine if the operation is success or not
  - The flags are used for this purpose
- IA-32 uses a single register: EFLAGS = **Extended** Flags Register (32 bits)
  - Contains status and control flags
  - Each flag is *a single binary bit*



EFLAGS Register



Status flags

CF = Carry flag
PF = Parity flag
AF = Auxiliary carry flag
ZF = Zero flag
SF = Sign flag
OF = Overflow flag

Control flags

DF = Direction flag

System flags

TF = Trap flag
IF = Interrupt flag
IOPL = I/O privilege level
NT = Nested task
RF = Resume flag
VM = Virtual 8086 mode
AC = Alignment check
VIF = Virtual interrupt flag
VIP = Virtual interrupt pending
ID = ID flag

- Status Flags
  - Status of arithmetic and logical operations
- Control and System flags
  - Control the CPU operation
- Programs can set and clear individual bits in the EFLAGS register

Status Flags

- Carry Flag
  - Set when unsigned arithmetic result is out of range
- Overflow Flag
  - Set when signed arithmetic result is out of range
- Sign Flag
  - Copy of sign bit, set when result is negative
- Zero Flag
  - Set when result is zero
- Auxiliary Carry Flag
  - Set when there is a carry from bit 3 to bit 4
- Parity Flag
  - Set when parity is even
  - Least-significant byte in result contains even number of 1s

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

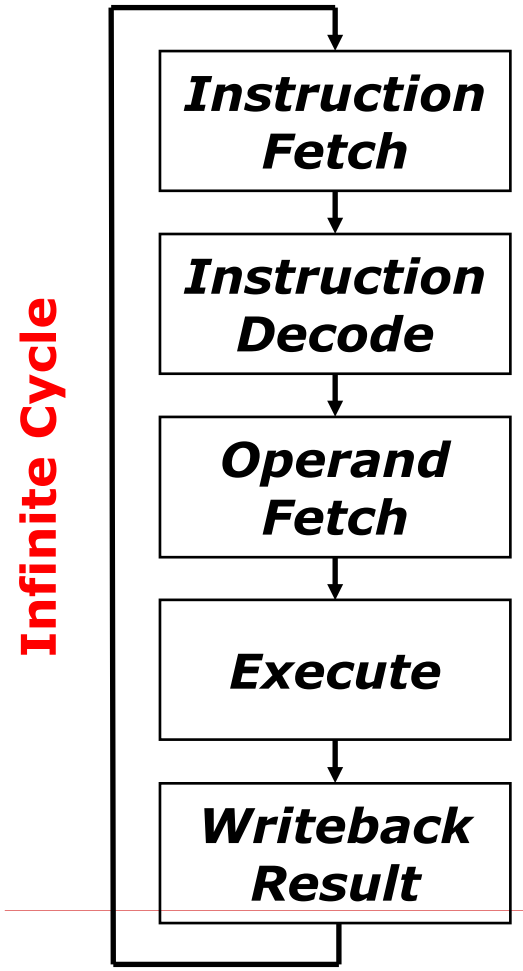
Next ...

- Basic Computer Organization
- IA-32 Registers
- Instruction Execution Cycle

Fetch-Execute Cycle: The Heart-beat of CPU

- Each machine language instruction is first fetched from the memory and stored in an **Instruction Register** or simply **IR**.
- The address of the instruction to be fetched is stored in a register called the **Instruction Pointer** or **EIP**. In some computers this register is called **Program Counter** or simply **PC**.
- After fetching the instruction, the **EIP** (or **PC**) is incremented to point to the address of the next instruction.
- The fetched instruction is decoded (to determine what needs to be done) and executed by the CPU.

Instruction Execute Cycle



Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

Compute result value and status

Deposit results in storage for later use