# Assembly Language for Intel-Based Computers, 4th Edition

Kip R. Irvine

## Chapter 2: IA-32 Processor Architecture
## Included elements of the IA-64 bit

*Slides prepared by Kip R. Irvine*

*Revision date: 09/25/2002*

***Modified by Dr. Nikolay Metodiev Sirakov 2015***

- Chapter corrections (Web) Assembly language sources (Web)
- Printing a slide show

# IA-32 Processor Architecture

- Modes of operation
- Basic execution environment
- Floating-point unit
- Intel Microprocessor history

# Modes of Operation

- Legacy Mode – which runs the 32 bit in the 64 bit machine
- Protected mode
  - native mode (Windows, Linux)
- Real-address mode – not available in the 64 but machines
  - native MS-DOS
- System management mode
  - power management, system security, diagnostics

- Virtual-8086 mode
  - hybrid of Protected
  - each program has its own 8086 computer

# Basic Execution Environment

- Addressable memory
- General-purpose registers
- Index and base registers
- Specialized register uses
- Status flags
- Floating-point, MMX, XMM registers

# Addressable Memory

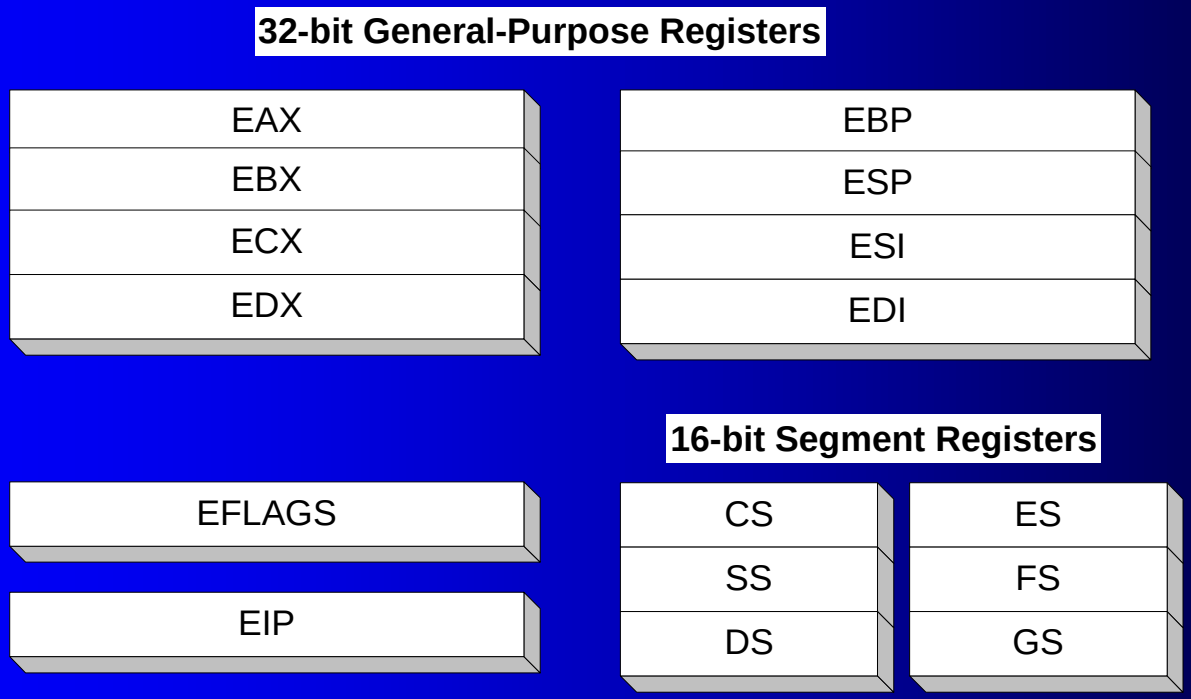
- $2^{64} - 1$ bytes could be addressed with the 64 CPU
- Protected mode
  - 4 GB
  - 32-bit address
- Real-address and Virtual-8086 modes
  - 1 MB space
  - 20-bit address

# General-Purpose Registers

Named storage locations inside the CPU, optimized for speed.

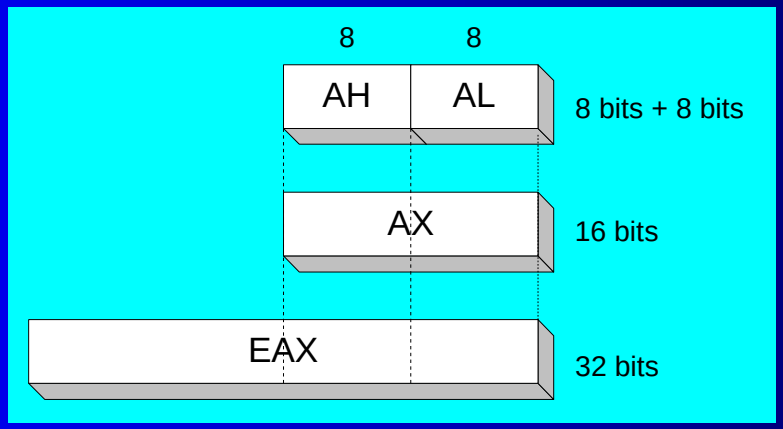
16 GPR in the 64 bit CPU- RAX, RBX,…., RSI, RDI'

R8,….,R15



# Accessing Parts of Registers

- Use 8-bit name, 16-bit name, or 32-bit name
- Applies to EAX, EBX, ECX, and EDX
- Take into account 32 bit division for the RAX,…..,R15




| 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 (1 of 2)

- The corresponding 64 bit registers carry the same purpose
- General-Purpose
  - EAX – accumulator
  - ECX – loop counter
  - ESP – stack pointer
  - ESI, EDI – index registers
  - EBP – extended frame pointer (stack)
- Segment
  - CS – code segment
  - DS – data segment
  - SS – stack segment
  - ES, FS, GS - additional segments

# Some Specialized Register Uses (2 of 2)

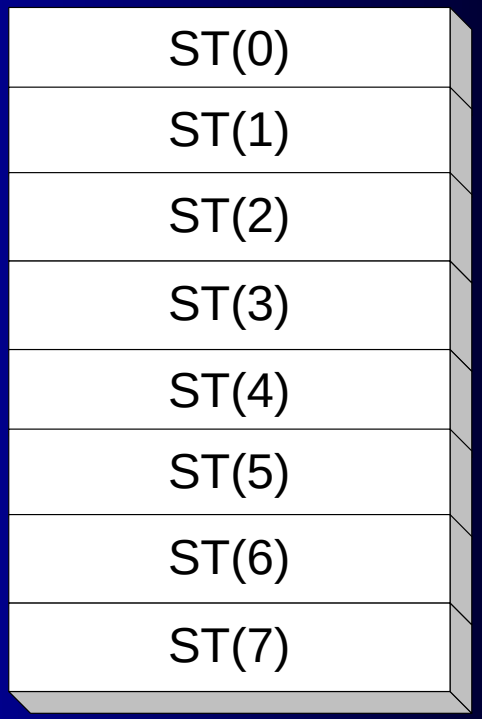
- The flags are the same in the 64-bit machine
- EIP – instruction pointer
- EFLAGS
  - status and control flags
  - each flag is a single binary bit

# Status Flags

- Carry - CF
  - unsigned arithmetic out of range
- Overflow-OF
  - signed arithmetic out of range
- Sign- SF
  - result is negative
- Zero- ZF
  - result is zero
- Auxiliary Carry -AC
  - carry from bit 3 to bit 4
- Parity -PC
  - sum of 1 bits is an even number

# Floating-Point, MMX, XMM Registers

- The 64 bit Intel AMD processors have 16 registers for Floating Point, each of them has 128 or 256 bit size
- Eight 80-bit floating-point data registers
  - ST(0), ST(1), . . . , ST(7)
  - arranged in a stack
  - used for all floating-point arithmetic
- Eight 64-bit MMX registers
- Eight 128-bit XMM registers for single-instruction multiple-data (SIMD) operations



# Intel Microprocessor History

- Intel 8086, 80286
- IA-32 processor family
- P6 processor family
- CISC and RISC

# Early Intel Microprocessors

- Intel 8080
  - 64K addressable RAM
  - 8-bit registers
  - CP/M operating system
  - S-100 BUS architecture
  - 8-inch floppy disks!
- Intel 8086/8088
  - IBM-PC Used 8088
  - 1 MB addressable RAM
  - 16-bit registers
  - 16-bit data bus (8-bit for 8088)
  - separate floating-point unit (8087)

# The IBM-AT

- Intel 80286
  - 16 MB addressable RAM
  - Protected memory
  - several times faster than 8086
  - introduced IDE bus architecture
  - 80287 floating point unit

# Intel IA-32 Family

- Intel386
  - 4 GB addressable RAM, 32-bit registers, paging (virtual memory)
- Intel486
  - instruction pipelining
- Pentium
  - superscalar, 32-bit address bus, 64-bit internal data path

# Intel P6 Family

- Pentium Pro
  - advanced optimization techniques in microcode
- Pentium II
  - MMX (multimedia) instruction set
- Pentium III
  - SIMD (streaming extensions) instructions
- Pentium 4
  - NetBurst micro-architecture, tuned for multimedia

# CISC and RISC

- CISC – complex instruction set
  - large instruction set
  - high-level operations
  - requires microcode interpreter
  - examples: Intel 80x86 family
- RISC – reduced instruction set
  - simple, atomic instructions
  - small instruction set
  - directly executed by hardware
  - examples:
    - ARM (Advanced RISC Machines)
    - DEC Alpha (now Compaq)