x86 Processor Architecture

Muhammad Afzaal
m.afzaal@nu.edu.pk

Book Chapter

- “Assembly Language for x86 processors”
- Author “Kip R. Irvine”
- 6th Edition
- Chapter 2
  - Section 2.2.4
  - Section 2.3

Outline

- **Overview of Intel Microprocessors**
- x86 Memory Management

Intel Microprocessors

- Size of data bus is significant in processor evolution
- How to transfer 32-bit data with an 8-bit data bus?

Evolution of Intel Microprocessors (1/7)

- Intel 8086
  - 16-bit registers and 16-bit data bus
  - 20-bit address bus
    - Physical Address Space = 2^{20} bytes = 1MB
- Intel 8087
  - Floating point co-processor
  - Uses segmentation and real-address mode to address memory
  - Each segment can address 2^{16} bytes = 64 Kbytes
- Intel 8088
  - Uses 8-bit data bus
  - … that's why less expensive

Evolution of Intel Microprocessors (2/7)

- Intel 80286
  - 24-bit address bus so
    - Physical Address Space = 2^{24} bytes = 16 MB
- Intel 80386
  - First 32-bit processor with 32-bit general purpose registers
  - 32-bit data bus and 32-bit address bus so
    - Physical Address Space = 2^{32} bytes = 4GB

Evolution of Intel Microprocessors (3/7)

- Intel 80486, introduced in 1989
  - Improved version of 80386
  - On-chip Floating Point Unit
  - On-chip unified Instruction/Data cache of 8 KB
  - Introduced the concept of Pipelining
- Pentium (80586), introduced in 1993
  - 32-bit address bus
  - 64-bit internal data path (inside a processor)
  - Superscalar performance: can execute two instructions in one clock cycle
  - Separate 8KB instruction and 8KB data caches

Evolution of Intel Microprocessors (4/7)

- Pentium Pro, introduced in 1995
  - Three-way superscalar: can execute 3 instructions per clock cycle
  - 36-bit address bus ➔ up to 64 GB of physical address space
  - Introduced dynamic execution
    - Out-of-order execution
  - Integrated a 256 KB second level L2 cache on chip
- Pentium II was introduced in 1997
  - Added MMX instructions
- Pentium III was introduced in 1999
  - Added SSE instructions and eight new 128-bit XMM registers

Evolution of Intel Microprocessors (5/7)

- Pentium IV is a seventh generation x86 architecture
  - New micro-architecture design called Intel Netburst
  - Very deep instruction pipeline, scaling to very high frequencies
- Intel introduced hyper-threading technology in 2002
  - Allowed 2 programs to run simultaneously, sharing resources
- Xeon is Intel's name for its server-class microprocessors
  - Xeon chips generally have more cache
  - Support larger multiprocessor configurations

Evolution of Intel Microprocessors (6/7)

- Pentium M (Mobile) was introduced in 2003
  - Designed for low-power laptop computers
  - Modified version of Pentium III, optimized for power efficiency
  - Large L2 cache
  - Runs at lower clock than Pentium IV but with better performance
- Extended Memory 64-bit Technology (EM64T)
  - 64-bit superset of the IA-32 processor architecture
  - 64-bit general purpose registers and integer support
  - Number of general purpose registers increased from 8 to 64
  - 64-bit pointers and flat virtual address space
  - Large physical address space: up to 2^{40} = 1 Terabyte

Evolution of Intel Microprocessors (7/7)

- CISC – Complete Instruction Set Computer
  - Large and complex instruction set
  - Variable width instructions
  - Requires microcode interpreter
  - Example is Intel x86 family
- RISC – Reduced Instruction Set Computer
  - Small and simple instruction set
  - All instructions have the same width
  - Simpler instruction formats and addressing modes
  - Decoded and executed directly by hardware
  - Examples are ARM, MIPS, PowerPC, SPARC etc.

Outline

- Overview of Intel Microprocessors
- **x86 Memory Management**

x86 Operation Modes

- x86 architecture operates in 3 basic different modes
  - Protected Mode
    - All instructions and features of processor are available
    - Programs cannot reference memory outside their segments
    - Virtual-8086 is special mode of Protected Mode
  - Real-Address Mode
    - Programs can have direct access to memory and hardware devices
    - … so operating system can crash in this mode
  - System Management Mode
    - Implemented in processors customized for a particular system setup
    - Supports power management and system security

x86 Memory Management

- Memory is managed according to the operation mode
  - Real-Address Mode
  - Protected Mode

Real-Address Mode Memory Management

- 16-bit registers
- 20-bit address bus
  - Physical Address Space = 1MByte
  - Linear or Absolute Address range is 00000h to FFFFFh
- How 20-bit address can be accommodated in 16-bit Registers?
  - Solution lies in Segmented Memory

Real-Address Mode Memory Management

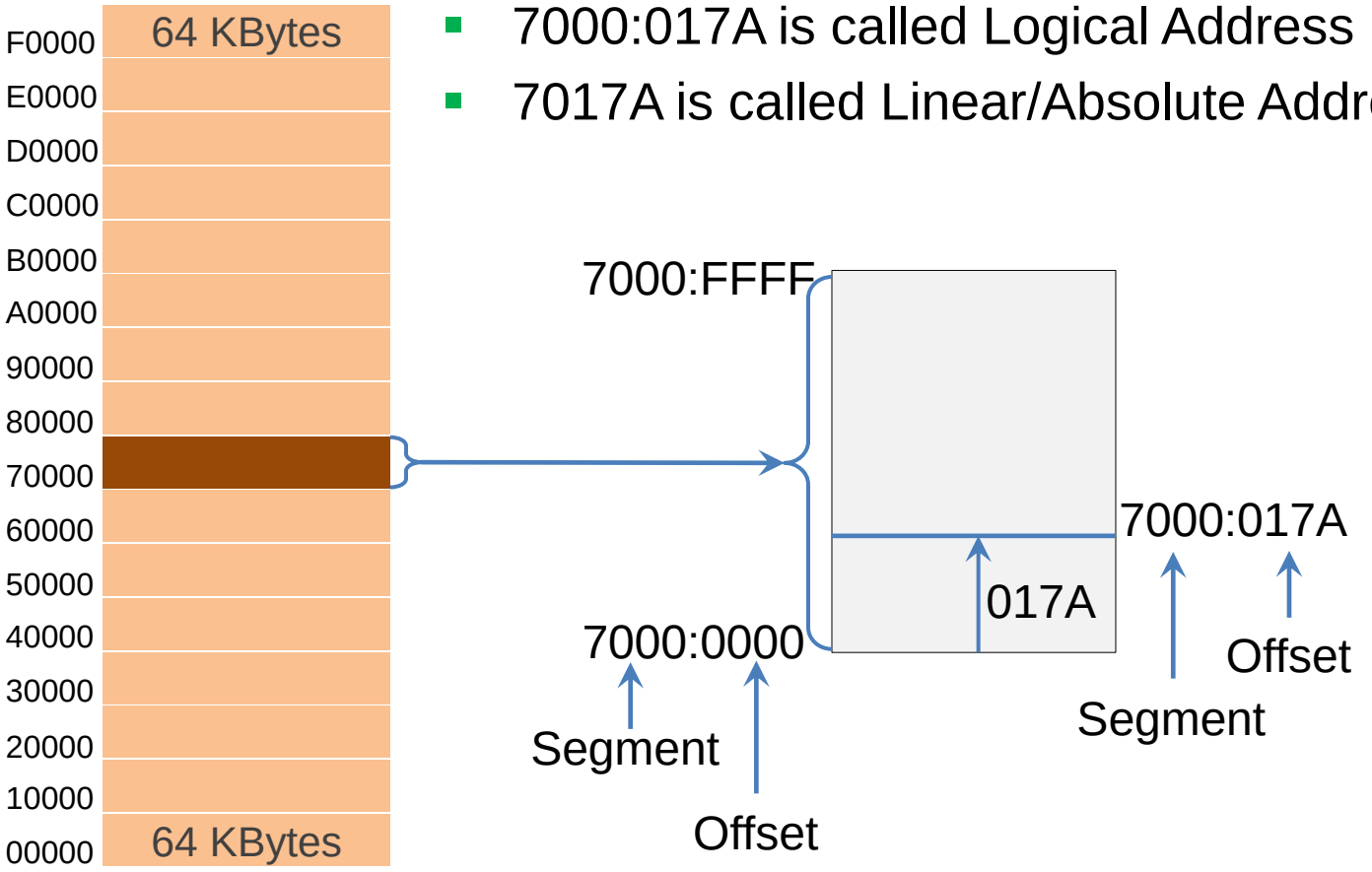
- Programs cannot use linear address directly because of 16-bit registers
- Addresses are represented using 2 16-bit registers
  - One 16-bit value is called Segment Value which is placed in segment register
  - Other 16-bit value is called Offset value

Segmented Memory Scheme

- All memory is divided in 64KBytes segments
- Each segment begins at an address having a zero in its last hex digit
  - … this zero is omitted when representing segment value in 16-bit registers
    - A segment value of $E000_{16}$ refers to the segment address of $E0000_{16}$

- If we have Physical Address Space of 2^{20} bytes, and size of each segment is 64KBytes (2^{16}bytes), how many segments do we have?

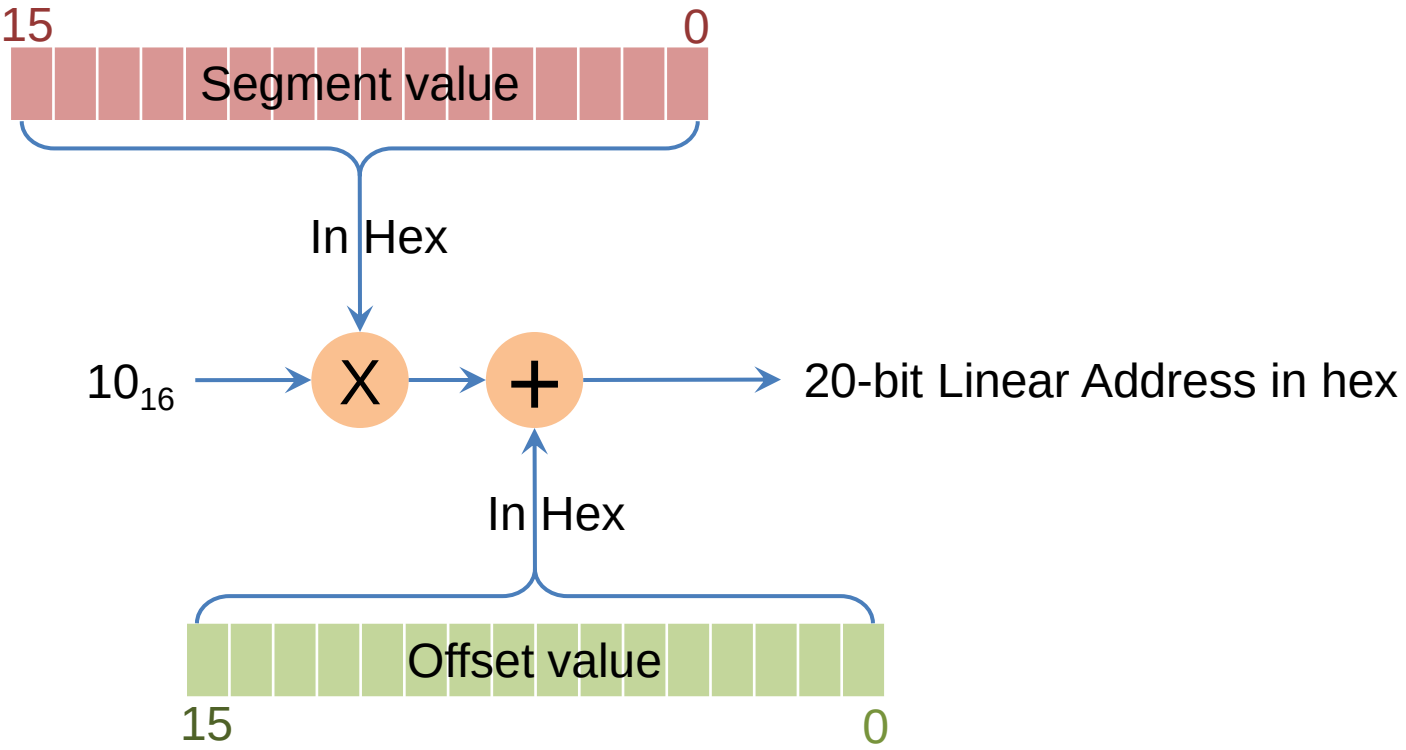
Segmented Memory Map



- 7000:017A is called Logical Address
- 7017A is called Linear/Absolute Address

20-bit Linear Address Calculation

- How to convert a 16-bitSeg:16-bitOffset logical address into 20-bit linear/absolute address



Protected Mode Memory Management

- Restricts application programs from accessing system hardware
- Logical Address consists of
  - 16-bit segment selector (Segment Registers)
  - 32-bit offset value
- Program's linear address space is 4GBytes
  - What is address bus width?
- A typical protected-mode program has three segments
  - Code Segment
  - Data Segment
  - Stack Segment
- Segment Registers point to the Segment Descriptor Table
  - … Segment Descriptor Tables are created by OS to keep track of individual program segments

Paging

- Paging divides the linear address space into fixed-sized block called pages
- IA-32 uses pages of size 4 KB
- Pages are allocated space on main memory
  - … if main memory is full, then on hard disk
- Complete set of pages mapped by OS is called Virtual Memory

Paging

- We want to run many programs in parallel
- Main memory may not be enough to load all of them at the same time
- Disk storage is cheaper and we have plenty of it
- Paging gives an illusion that we have unlimited main memory
- The more a program depends on paging, the slower it runs
- Higher amount of main memory means less usage of paging