

ECE390

Computer Engineering II

Lecture 2



Dr. Zbigniew Kalbarczyk
University of Illinois at Urbana- Champaign

Outline



- Basic microprocessor and system architecture
- Memory
- Programming model
- Memory addressing
 - real mode
 - protected mode

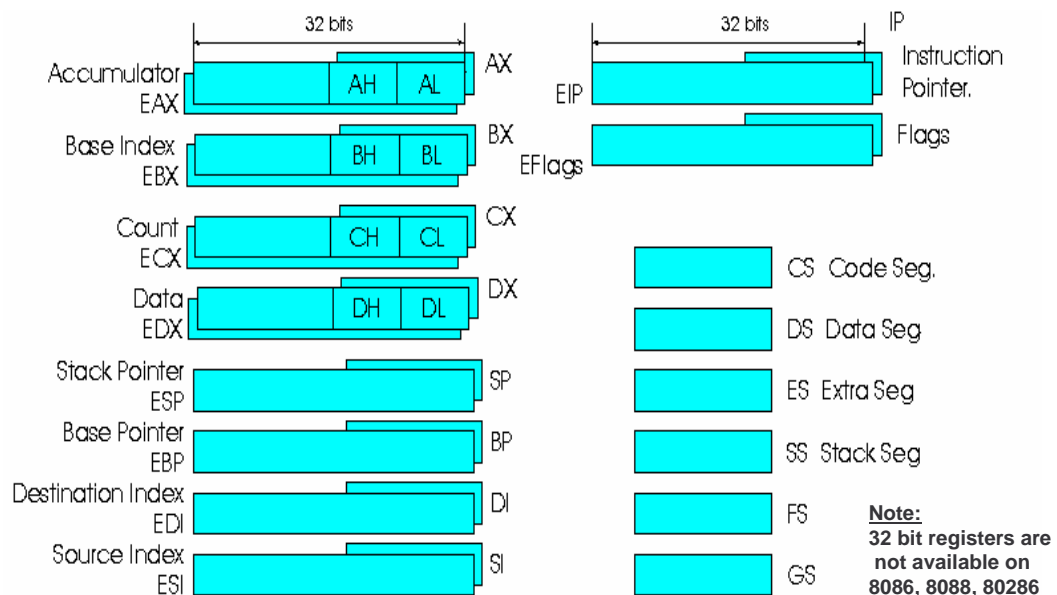
Microprocessor Architecture

Basic Components

- CPU Registers
 - special memory locations constructed from flip-flops and implemented on-chip
 - e.g., accumulator, count register, flag register
- Arithmetic and Logic Unit (ALU)
 - ALU is where most of the action take place inside the CPU
- Bus Interface Unit (BIU)
 - responsible for controlling the address and data busses when accessing main memory and data in the cache
- Control Unit and Instruction Set
 - CPU has a fixed set of instructions to work on, e.g., MOV, CMP, JMP

Programming Model

Registers



Programming Model Registers (examples)

- General-Purpose Registers
 - AX (accumulator) often holds the temporary result after an arithmetic and logic operation
 - BX (base) often holds the base (offset) address of data located in the memory
- Pointer and Index Registers
 - SP (stack pointer) used to address data in a LIFO (last-in, first-out) stack memory
 - BP (base pointer) often used to address an array of data in the stack memory

Programming Model Flag Register

- Flags indicate the condition of the microprocessor as well as its operation
- The flag bits change after many arithmetic and logic instructions execute
- Example flags,
 - **C(carry)** indicates carry after addition or a borrow after subtraction
 - **O(overflow)** is a condition that occurs when signed numbers are added or subtracted
 - **Z(zero)** indicates that the result of an arithmetic or logic operation is zero
 - **T(trap)** when the trap flag is set , it enables trapping through the on-chip debugging feature

Programming Model

Segment Registers

- Segment registers generate memory addresses along with other registers in the microprocessor
- CS(code) defines the starting address of the section of memory-holding code(programs and procedures used by programs)
- DS(data) a section of memory that contains most data used by a program
- ES(extra) an additional data segment
- SS(stack) defines the area of memory used for the stack
- FS and GS available on 80386 and 80486 allow two additional memory segments for access by programs

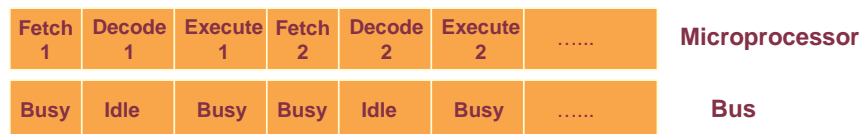
Z. Kalbarczyk

ECE390

Microprocessor Architecture

Instruction processing

- Processing of an instruction by microprocessor consists of three basic steps: (1) fetch instruction from the memory, (2) decode the instruction, and (3) execute (usually involves accessing the memory for getting operands and storing results)
- Operation of an early processor, e.g., 8085



Z. Kalbarczyk

ECE390

Microprocessor Architecture

Instruction processing

- Modern microprocessors can process several instructions simultaneously at various stages of execution
 - this ability is called *pipeline*
- Operation of the pipelined microprocessor, e.g., 80486



Z. Kalbarczyk

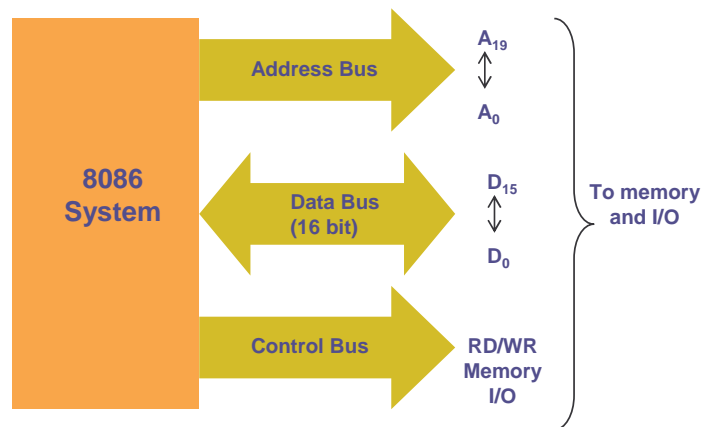
ECE390

System Architecture

Address Bus provides a memory address to the system memory and I/O address to the system I/O devices

Data Bus transfers data between the microprocessor and the memory and I/O attached to the system

Control Bus provides control signals that cause the memory or I/O to perform a read or write operation



Z. Kalbarczyk

ECE390

Processor Data and Address Bus Sizes

Examples

Processor	Data Bus	Address Bus	Max Addressable Memory
8088	8	20	1,048,576 (1Mb)
8086	16	20	1,048,576 (1Mb)
80286	16	24	16,777,216 (16Mb)
80386dx	32	32	4,294,976,296 (4Gb)
80486	32	32	4,294,976,296 (4Gb)
80586/Pentium (Pro)	64	32	4,294,976,296 (4Gb)

Memory

- Microprocessor addresses a maximum of 2^n different memory locations, where n is a number of bits on the address bus
- Logical Memory
 - 80x86 supports byte addressable memory
 - byte (8 bits) is a basic memory unit
 - e.g., when you specify address 24 in memory, you get the entire eight bits
 - when the microprocessors address a 16-bit word of memory, two consecutive bytes are accessed

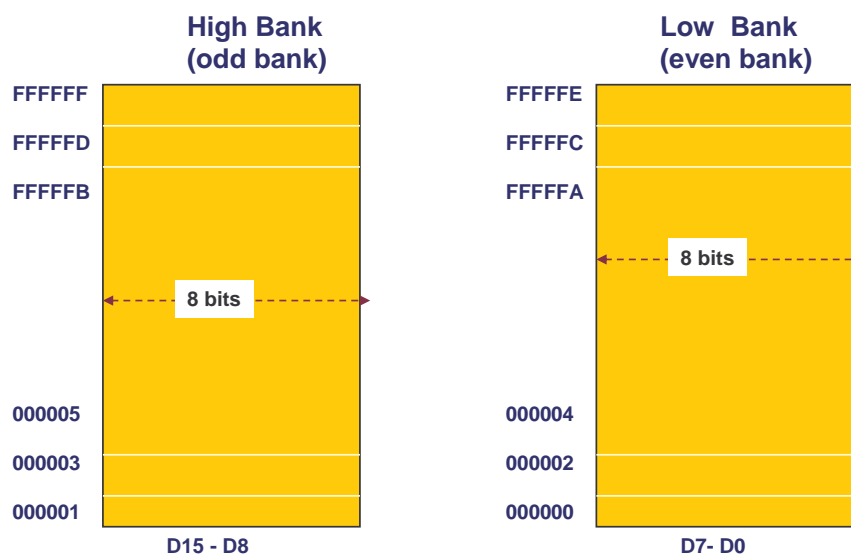
Memory (cont.)

- Physical Memory
 - The physical memories of 80x86 family differ in width
 - e.g., 8088 memory is 8 bits wide,
 - 8086, 80286 memory is 16 bits wide, and
 - 80386dx, 80486 memory is 32 bits wide
 - for programming there is no difference in memory width, because the logical memory is always 8-bit wide
 - memory is organized in memory banks
 - a memory bank is an 8-bit wide section of the memory
 - e.g., the 16-bit microprocessors contain two memory banks to form 16-bit wide section of memory that is addressed as bytes or words

Z. Kalbarczyk

ECE390

Physical Memory System Example (16 bit microprocessor)



Z. Kalbarczyk

ECE390

Accessing Data in Memory

Example (16 bit microprocessor)

- Accessing word from an even address - L.O. byte from the address specified and the H.O. byte from the next consecutive address
- *What if you access a word on an odd address?*
- Example: access memory on address 125, i.e., we want to access data on address 125 (L.O.) and 126 (H.O.)
 - this requires two memory operations
 - read byte on address 125
 - read byte on address 126
 - swap the positions of these bytes internally since both entered the CPU on the wrong half of the data bus
 - 80x86 CPUs recognize this and perform transfer automatically
- Your programs can access words at any address and the CPU will properly access and swap the data in memory
- Think about the speed of your program when accessing words at odd addresses

Z. Kalbarczyk

ECE390

Memory Data Types

- Numbers
 - **bit** (e.g., 1) ; **nibble** = 4 bits
 - **DB**: byte = octet = 8 bits
 - **DW**: Word = 2 bytes = 16 bits (80x86 terminology)
 - **DD**: DoubleWord = 4 bytes = 32 bits (80x86 terminology)
 - Intel uses little endian format (i.e., LSB at lower address)
 - Signed Integers (2's complement)
- Text
 - Letters and characters (7-bit ASCII standard), e.g., 'A'=65=0x41
 - Extended ASCII (8-bit) allows for extra 128 graphics/symbols)
 - Collection of characters = Strings
 - Collection of Strings = Documents

Z. Kalbarczyk

ECE390

Memory Data Types (cont.)

- Programs
 - Commands (MOV, JMP, AND, OR, NOT)
 - Collections of commands = subroutines
 - Collection of subroutines = programs
- Floating point numbers (covered later)
- Images (GIF, TIF, JPG, BMP)
- Video (MPEG, QuickTime, AVI)
- Audio (voice, music)

Z. Kalbarczyk

ECE390

Example of Memory with Stored Data

Address	Data (8-bits)	Interpretation
0xFFFFF		
...		
0x75000	0x55	byte
...		
0x70009	'\$'	String
0x70008	'0'	
0x70007	'9'	
0x70006	'3'	
0x70004	'E'	
0x70003	'C'	
0x70002	'E'	
...		
0x60511	0x12	Word
0x60510	0x34	Word 3x1 integer array of 16-bit words
0x6050F	0x12	
0x6050E	0x34	
0x6050D	0x12	Word
0x6050C	0x34	
...		
0x55504	0xFE	JE-2 Program
0x55003	opcode	
0x55002	0x02	
0x55001	opcode	
...		
0x00000		

Z. Kalbarczyk

Real Mode Memory Addressing

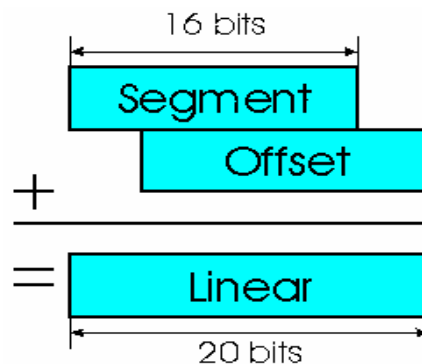
- 80286 - 80486 microprocessors operate in either the real or protected mode
- 8086, 8088, and 80186 only operate in the real mode
- Real mode operation allows the microprocessor to only address the first 1M byte of memory space (even if it is an 80486 microprocessor)
- Each of 80x86 processors operates in the real mode by default
- All real mode memory addresses consist of a *segment address* plus an *offset address*
 - **the segment address** (in one of the segment registers) defines the beginning address of any 64K byte memory segment
 - **the offset address** selects a location within the 64K byte memory segment

Z. Kalbarczyk

ECE390

Real Mode Memory Addressing (cont.)

- Generation of 20-bit linear address from a **segment:offset** address
- in the real mode, each segment register (16 bits) is internally appended with a *0h* on its rightmost end (i.e., the segment is shifted left by 4 bits)
- The segment and the offset are then added to form 20-bit memory address.



Z. Kalbarczyk

ECE390

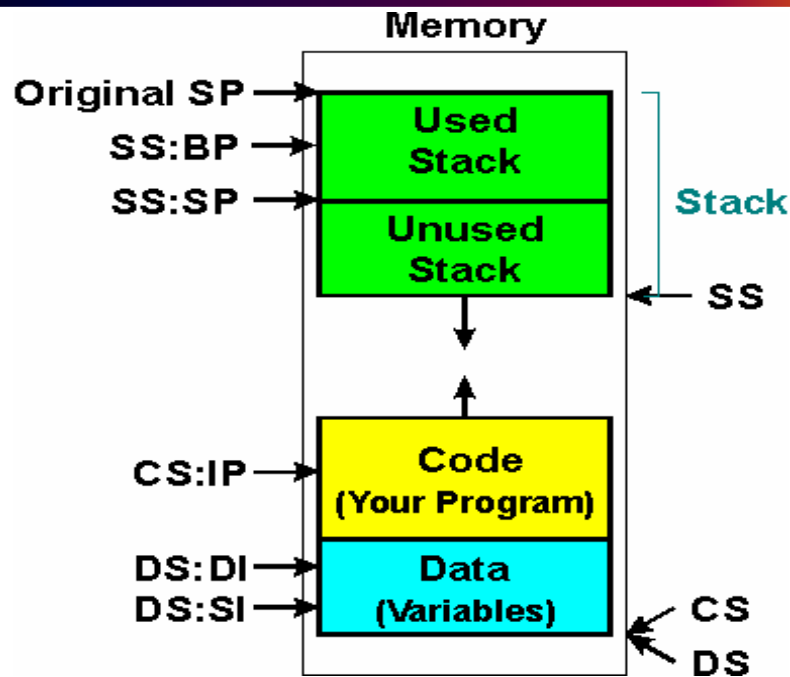
Real Mode Memory Addressing Examples

- (1) Linear address for **Segment:Offset** = 2222:3333 = 25553
Segment:offset address for Linear address=25553:
 - Many Answers - One possibility: 2222:3333
 - Many Answers - One possibility: 2000:5553
- (2) Linear address for **Segment:Offset** = 1200:F445 = 21445
Segment:offset address for Linear address=21445:
 - Many Answers - One possibility: 1200:F445
 - Many Answers - One possibility 2000:1445

Protected Mode Memory Addressing

- In 80286 and later processors the addressing capabilities of a microprocessor are extended by changing the function the CPU uses to convert a logical address to the linear address space
 - the protected mode processors use a look up table to compute the physical address
 - the segment value is used as an index into an array (segment descriptor table)
 - the contents of the selected array element provides the starting address for the segment
 - the CPU adds this value to the offset to obtain the physical address

Use of Segments



Z. Kalbarczyk

ECE390

Peripherals

- Memory-mapped devices (special memory locations in the normal address space of the CPU)
 - BIOS: 0xF0000-0xFFFFF (bootstrap, I/O calls)
 - Video: 0xA0000-0xBFFFF and vBIOS: 0xC0000-0xC7FFF
- I/O mapped devices (sound card, com ports, parallel port)
 - I/O addresses different than Memory addresses
 - Address Range: 0x0000 - 0xFFFF (16-bit)
- Interrupts
 - Notifies the CPU when an event has occurred
 - Timer [update clock] , serial I/O [input data], Parallel I/O [ready]
 - Network adapter [packet arrived]

Z. Kalbarczyk

ECE390