

Program: BS(CS)-V
Course Name: Computer Organization & Assembly Language
Course Code: CS-501
Credit hours: 03
Total Hours: 48
Total Weeks: 16

COURSE OBJECTIVES

This is an introduction to computer organization, machine language programming and use of assembly language programming systems. Topics include basic instructional sets, coding arithmetic problems, logical operations, indexing, indirect addressing, MACRO commands, MACRO programming techniques and subroutines.

The course introduces students to write computer programs using assembly language and provides an opportunity for students to generalize the fundamental computer operations, and data types. Students will identify the 80 x 86 microprocessor family and the 8086 / 8088 microprocessor architectures; the instruction set, registers, memory-addressing interrupts.

After completing this course, students will be able to:

- Assess the architecture and operation of the machine and its operating system.
- Convert numbers among base-2, base-8, base-16, and base-10.
- Identify data definition directives.
- Identify different addressing modes: register-to-register, immediate, direct, indirect, based, indexed, and base-indexed addresses.
- Calculate indirect and indexed addresses.
- Use the syntax and semantics of data movement (MOVE instruction, offsets, stack operations) and arithmetic operations (shift and rotate instructions, signed arithmetic).
- Apply software interrupts, DOS function calls, BIOS-level video control operations.
- Use conditional processing operations and string processing operations.

This course will help and prepare the students to take advance courses in system programming and advance computer architecture in future.

Basic Concepts

Week-1

- Welcome to Assembly Language
 - Some Good Questions to Ask
 - Assembly Language Applications
- Virtual Machine Concept
 - The History of PC Assemblers

Week-2

- Data Representation
 - Binary Numbers
 - Binary Addition
 - Integer Storage Sizes
 - Hexadecimal Integers
 - Signed Integers
 - Character Storage
- Boolean Operations
 - Truth Tables for Boolean Functions

IA-32 Processor Architecture

Week-3

- General Concepts
 - Basic Microcomputer Design
 - Instruction Execution Cycle
 - Reading from Memory
 - How Programs Run
- IA-32 Processor Architecture
 - Modes of Operation
 - Basic Execution Environment
 - Floating-Point Unit
 - Intel Microprocessor History

Week-4

- IA-32 Memory Management
 - Real-Address Mode
 - Protected Mode
- Components of an IA-32 Microcomputer
 - Motherboard
 - Video Output
 - Memory
 - Input-Output Ports
- Input-Output System
 - How It All Works

Assembly Language Fundamentals

Week-5

- Basic Elements of Assembly Language
 - Integer Constants
 - Integer Expressions
 - Real Number Constants
 - Character Constants
 - String Constants
 - Reserved Words
 - Identifiers
 - Directives
 - Instructions
- Example: Adding Three Integers
 - Program Listing
 - Program Output
 - Program Description
 - Program Template
- Assembling, Linking, and Running Programs
 - The Assemble-Link-Execute Cycle

Week-6

- Defining Data
 - Intrinsic Data Types
 - Data Definition Statement
 - Defining BYTE and SBYTE Data
 - Defining WORD and SWORD Data
 - Defining DWORD and SDWORD Data
 - Defining QWORD Data
 - Defining TBYTE Data
 - Defining Real Number Data
 - Little Endian Order
 - Adding Variables to the AddSub Program
 - Declaring Uninitialized Data
- Symbolic Constants
 - Equal-Sign Directive
 - Calculating the Sizes of Arrays and Strings
 - EQU Directive
 - TEXTEQU Directive
- Real-Address Mode Programming
 - Basic Changes

Data Transfers, Addressing, and Arithmetic

Week-7

- Data Transfer Instructions
 - Introduction
 - Operand Types
 - Direct Memory Operands
 - MOV Instruction
 - Zero/Sign Extension of Integers
 - LAHF and SAHF Instructions
 - XCHG Instruction
 - Direct-Offset Operands
 - Example Program (Moves)
- Addition and Subtraction
 - INC and DEC Instructions
 - ADD Instruction
 - SUB Instruction
 - NEG Instruction
 - Implementing Arithmetic Expressions
 - Flags Affected by Arithmetic
 - Example Program (AddSub3)

Week-8

- Data-Related Operators and Directives
 - OFFSET Operator
 - ALIGN Directive
 - PTR Operator
 - TYPE Operator
 - LENGTHOF Operator
 - SIZEOF Operator
 - LABEL Directive
- Indirect Addressing

- Indirect Operands
- Arrays
- Indexed Operands
- Pointers
- JMP and LOOP Instructions
 - JMP Instruction
 - LOOP Instruction
 - Summing an Integer Array
 - Copying a String

Procedures

Week-9

- Introduction
- Linking to an External Library
 - Background Information
- The Book's Link Library
 - Overview
 - Individual Procedure Descriptions
 - Library Test Programs
- Stack Operations
 - Runtime Stack
 - PUSH and POP Instructions

Week-10

- Defining and Using Procedures
 - PROC Directive
 - CALL and RET Instructions
 - Example: Summing an Integer Array
 - Flowcharts
 - Saving and Restoring Registers
- Program Design Using Procedures
 - Integer Summation Program (Design)

Conditional Processing

Week-11

- Introduction
- Boolean and Comparison Instructions
 - The CPU Flags
 - AND Instruction
 - OR Instruction
 - XOR Instruction
 - NOT Instruction
 - TEST Instruction
 - CMP Instruction
 - Setting and Clearing Individual CPU Flags
- Conditional Jumps
 - Conditional Structures
 - Jcond Instruction
 - Types of Conditional Jump Instructions
 - Conditional Jump Applications
 - Bit Testing Instructions

- Conditional Loop Instructions
 - LOOPZ and LOOPE Instructions
 - LOOPNZ and LOOPNE Instructions

Week-12

- Conditional Structures
 - Block-Structured IF Statements
 - Compound Expressions
 - WHILE Loops
 - Table-Driven Selection
- Application: Finite-State Machines
 - Validating an Input String
 - Validating a Signed Integer
- Using the .IF Directive
 - Signed and Unsigned Comparisons
 - Compound Expressions
 - .REPEAT and .WHILE Directives

Integer Arithmetic

Week-13

- Introduction
- Shift and Rotate Instructions
 - Logical Shifts and Arithmetic Shifts
 - SHL Instruction
 - SHR Instruction
 - SAL and SAR Instructions
 - ROL Instruction
 - ROR Instruction
 - RCL and RCR Instructions
 - SHLD/SHRD Instructions
- Shift and Rotate Applications
 - Shifting Multiple Doublewords
 - Binary Multiplication
 - Displaying Binary Bits
 - Isolating a Bit String
- Multiplication and Division Operations
 - MUL Instruction
 - IMUL Instruction
 - DIV Instruction
 - Signed Integer Division
 - Implementing Arithmetic Expressions

Week-14

- Extended Addition and Subtraction
 - ADC Instruction
 - Extended Addition Example
 - SBB Instruction
- ASCII and Packed Decimal Arithmetic
 - AAA Instruction
 - AAS Instruction
 - AAM Instruction
 - AAD Instruction

-Packed Decimal Arithmetic

16-Bit MS-DOS Programming

Week-15

- MS-DOS and the IBM-PC
 - Memory Organization
 - Redirecting Input-Output
 - Software Interrupts
 - INT Instruction
 - MS-DOS Function Calls (INT 21h)
 - Selected Output Functions
 - Hello World Program Example
 - Selected Input Functions
 - Date/Time Functions

Week-16

- Standard MS-DOS File I/O Services
 - Create or Open File (716Ch)
 - Close File Handle (3Eh)
 - Move File Pointer (42h)
 - Get File Creation Date and Time
 - Selected Library Procedures
 - Example: Read and Copy a Text File
 - Reading the MS-DOS Command Tail
 - Example: Creating a Binary File

Total Marks: 100

Recommended Books:

ASSEMBLY LANGUAGE FOR INTEL-BASED COMPUTERS 4TH Edition,
Kip R. Irvine Prentice Hall International edition

Reference Books:

1. Assembly Language Programming & Organization of the IBM PC 1st Edition,
Ytha Yu and Charles Marut McGraw Hill International Edition
2. ASSEMBLY LANGUAGE FOR PRIMER FOR THE IBM PC & XT 1st Edition,
Robert Lafore Wait Group International Edition