**Course name:** Computer Organization and Assembly Language   
**Course code:** EE218   
**Course Structure:** Theory: 4 / Lab: 0   
**Credit hours:** 4   
**Objectives:** - To introduce basic concepts of computer organization - To illustrate the computer organization concepts by Assembly language programming - To provide knowledge of Assembly language for the most recent processor such as Intel Pentium processor   
**Pre-requisites:** Digital Logic Design   
**Text books and references:** - Assembly Language for Intel Based Computer, 4th Edition, by Kip R. Irvine   
**Recommended books:** - The 80x86 Family Design, Programming, and Interfacing, 4th Edition, by John Uffenbeck - Assembly language for x86 processors by Kip R. Irvine - The intel microprocessors, 4th edition, by Barry b. bery   
**Lab projects:** Lab 1. Assembler’s Installation Lab 2. Installation and Configuration of Textpad Editor Lab 3. Basics of Assembly Language & Interfacing with C++ Lab 4. Using Data Types and Symbolic Constants Lab 5. Real Mode Debugging in CodeView Lab 6. Data Transfer Instructions Lab 7. Flags, Offset & Operators Lab 8. JMP, LOOP Instructions and Procedures Lab 9. Conditional Processing Lab 10. Conditional Structures Lab 11. Integer Arithmetic Lab 12. Strings and Arrays Lab 13. Structures and Macros Lab 14. Conditional-Assembly Directives Lab 15. Lab exam

**Course Outline:**

Chapter 1: Basic Concepts   
1.1: Assembly Language Introduction, 1.2: Virtual Machine Concept, 1.3: Data Representation,   
1.4: Boolean Operators   
Chapter 2: IA-32 Processor Architecture   
2.1: General Concepts, 2.2: IA-32 Processor Architecture, 2.3: IA-32 Memory Management,   
2.4: Components of an IA-32 Microcomputer   
Chapter 3: Assembly Language Fundamentals   
3.1 Basic Elements of Assembly Language, 3.2: Adding Three Integers, 3.3: Assembling, Linking, and Running Programs, 3.4: Defining Data, 3.5: Symbolic Constants   
Chapter 4: Data Transfers, Addressing and Arithmetic   
4.1: Data Transfer Instructions, 4.2: Addition and Subtraction, 4.3: Data-Related Operators and Directives, 4.4: Indirect Addressing, 4.5: JMP and LOOP Instructions   
Chapter 5: Procedures   
5.1: Introduction to Procedures, 5.2: Linking to an External Library, 5.3: The Book’s Link Library, 5.4: Stack Operations, 5.5: Defining and Using Procedures, 5.5: Program Design Using Procedures   
Chapter 6: Conditional Processsing   
6.1: Introduction to Conditional Processing, 6.2: Boolean and Comparison Instructions, 6.3: Conditional Jumps, 6.4: Conditional Loop Instructions and Structures, 6.5 Finite-State Machines   
Chapter 7: Integer Arithmetic   
7.1: Shift and Rotate instructions and its Applications, 7.2: Multiplication and Division Instructions, 7.3: Extended Addition and Subtraction   
Chapter 8: Advanced Procedures   
8.1: Local Variables, 8.2: Stack Parameters, 8.3: Stack Frames, 8.4: Recursion, 8.5: Creating Multimodule Programs   
Chapter 9: Advanced Procedures   
9.1: Strings and Arrays, 9.2: Selected String Procedures, 9.3: Two-Dimensional Arrays 9.4: Searching and Sorting Integer Arrays   
Chapter 10: Structures and Macros   
10.1: Structures, 10.2: Macros, 10.3: Conditional-Assembly Directives, 10.4: Defining Repeat Blocks   
Chapter 11: 32-Bit Windows Programming   
11.1: Console input, 11.2: Console output, 11.3: Reading and Writing Files, 11.4: Console Window Manipulation, 11.5: Controlling the Cursor   
Chapter 11: 32-Bit Windows Programming   
11.6: Controlling the Text Color, 11.7: Time and Date Functions, 11.8: Writing a Graphical Windows, 11.9: 1A-32 Memory Management (Tentative)   
Chapter 12: High level Language Interface   
12.1: Inline Assembly Code,12.2: Linking to C++   
Chapter 13: 16-Bit MS-DOS Programming   
13.1: MS-DOS and the IBM-PC, 13.2: MS-DOS Function Calls   
Chapter 14: DISK Fundamentals   
14.1: Disk Storage Systems, 14.2: File Systems, 14.3: Disk Directory   
Chapter 15: Bios-Level Programming (Tentative)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Course name:** Computer Organization & Assembly Language Lab   
**Course code:** EE218   
**Course Structure:** Theory: 0 / Lab: 1   
**Credit hours:** 1   
**Objectives:** The overall objectives are: - To become familiar with the basic components of 8086 instruction set architecture - To inspect and modify 8086 processor registers and memory. - To use the MASM assembler to develop and run assembly language programs. - To introduce students to several programming techniques, which are used with assembly programming languages.   
**Pre-requisites:** EE109 - Digital Logic Design   
**Text books and references:** Assembly language for Intel-based computers By kip r. Irvine   
**Recommended books:** Assembly language for Intel-based computers By kip r. Irvine   
**Lab projects:** None   
**Course Outline:** Introduction to Assemblers and Text Editors, Installation of MASM, Assembling, Linking and executing “.asm” files using command prompt Installation & Configuration of TextPad Editor Basic Assembly Commands, Assembly Language Interfacing with C++ Data Types in Assembly language, Symbolic Constants Real Mode Debugging in CodeView Data Transfers, Addressing, and Arithmetic Flags Affected by Arithmetic, OFFSET Operator, PTR Operator, TYPE Operator, LENGTHOF Operator, SIZEOF Operator, LABEL Operator JMP, LOOP Instructions and Procedures Boolean and Comparison Instructions, Conditional Jumps, BT (Bit Test) Instruction, Conditional Loop Instructions Structured IF Statements, Compound Expressions (Logical AND, OR Operators), WHILE Loops Shift and Rotate Instructions, Multiplication and Division Instructions, Extended Addition and Subtraction String Primitive Instructions, String Procedures, Searching and Sorting Integer Arrays Structures, Unions and Macros IF, ELSE and ENDIF Directives, The IFIDN and IFIDNI Directives, Macro Functions, WHILE, REPEAT, FOR and FORC Directives Interrupts