Vocational Skill Courses (VSC)

Name of the Course: Assembly Language Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Introduction: The 8085 Assembly Language Programming course covers the principles and practices of writing low-level software that controls the 8085 microprocessor. This course provides an in-depth understanding of the 8085 microprocessor architecture and its instruction set, as well as how to write, debug, and optimize assembly language programs for this microprocessor.
		Relevance and Usefulness: The course is relevant to computer science/engineering students interested in learning about microprocessors and embedded systems programming. The course provides the fundamental knowledge and skills required to design and implement computer systems with low-level software control. Assembly language programming is the foundation of modern computer technology, which makes the course relevant to anyone interested in computer systems and programming.
		Application and Interest: The course is essential for students aspiring to work in the field of embedded systems, microcontroller/microprocessor programming, or any programming role that involves low-level software development. By the end of the course, students will be able to write efficient and optimized assembly language programs that control the functionality of a microprocessor.
		Connection with Other Courses: 8085 Assembly Language Programming is a fundamental course that provides an understanding of how computer systems work at the lowest level. It connects with several other computer science courses, such as Computer Organization and Architecture, Operating Systems, Compiler Design, and Embedded Systems Design.
		Demand in the Industry and Job Prospects: There is a high demand in the industry for programmers who possess knowledge of low-level software development, such as programming microprocessors with assembly language. Many industries, including aerospace, automotive, healthcare, and consumer electronics, require low-level software development skills in their employees. Job prospects for graduates with expertise in 8085 Assembly language

			programming are abundant in these sectors. Job roles may include embedded software engineer, hardware engineer, firmware developer, software developer, and testing/validation engineer.
2	**Vertical :**		Vocational Skill Course(VSC)
3	**Type :**		Practical
4	**Credits :**		2 credits (60 hours in a semester)
5	**Hours Allotted :**		60 Hours
6	**Marks Allotted:**		50 Marks
7	**Course Objectives(CO):**		

CO 1. To gain a thorough understanding of the 8085 microprocessor architecture and its associated instruction set.

CO 2. To develop the ability to write and debug assembly language programs for the 8085 microprocessor.

CO 3. To learn the principles of computer organization and how they relate to the 8085 microprocessor.

CO 4. To become proficient in the use of 8085 assembly language programming tools, simulators, and debuggers.

CO 5. To learn how to interface different input/output devices with the 8085 microprocessor.

CO 6. To understand the concept of interrupts and how they are used in 8085 assembly language programming.

8	**Course Outcomes(CO):**

OC 1. Explain the architecture of the 8085 microprocessor and its associated instruction set.

OC 2. Identify the different types of registers and their functions in the microprocessor.

OC 3. Describe the memory organization and addressing modes of the 8085 microprocessor.

OC 4. Write assembly language programs for the 8085 microprocessor using various instructions and addressing modes.

OC 5. Debug and troubleshoot assembly language programs for the 8085 microprocessor using simulators and debuggers.

OC 6. Implement conditional branching and looping constructs in assembly language programs.

OC 7. Use 8085 assembly language programming tools, such as editors, assemblers, and emulators for developing and testing programs.

OC 8. Simulate microprocessor operations using emulators and debuggers.

OC 9. Connect input/output devices, such as LEDs, switches, and displays, to the 8085 microprocessor.

OC 10.

9	**Modules:-**
	Module 1:

1. Perform the following Operations related to memory locations.	
a. Store the data byte 32H into memory location 4000H.	
b. Exchange the contents of memory locations 2000H and 4000H	**30 Hrs**
2. Simple assembly language programs.	

a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
b. Subtract two 8-bit numbers.
c. Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
d. Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.
e. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
f. Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.
g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.

3. **Packing and unpacking operations.**
a. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.
b. Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.

4. **Register Operations**
a. Write a program to shift an eight bit data four bits right. Assume that data is in register C.
b. Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair
c. Write a set of instructions to alter the contents of flag register in 8085.
d. Write a program to count number of l's in the contents of D register and store the count in the B register.

5. **Multiple memory locations.**
a. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H
b. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.
c. Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.

d. Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H

e. Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.

Module 2:	
1. Calculations with respect to memory locations. a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order. b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem: c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H. d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H f. Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H g. Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array I, array2 and array3 are 2200H, 2300H and 2400H, respectively **2. Assembly programs on memory locations.** a. Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H b. Write assembly language program with proper comments for the following: c. A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.	**30 Hrs**

d. Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character.
e. A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively
f. Write an assembly language program to generate Fibonacci number.
g. Program to calculate the factorial of a number between 0 to 8.

3. String operations in assembly programs.
a. Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters
b. Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.
c. Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H.
d. Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H
e. DAA instruction is not present. Write a sub routine which will perform the same task as DAA.

4. Calculations on memory locations.
a. To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10
b. Arrange an array of 8 bit unsigned no in descending order
c. Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H
d. Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H.
e. Write a simple program to Split a HEX data into two nibbles and store it in memory

5. Operations on BCD numbers.
a. Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.
b. Subtract the BCD number stored in E register from the number stored in the D register
c. Write an assembly language program to multiply 2 BCD numbers

| 10 | **Text Books** |
| | 1. 8080A/8085 Assembly Language Programming, Lance A. Leventhel, Osborne, 1978 |

11	**Reference Books**	
	1. Microprocessors Architecture, Programming and Applications with the 8085, Fifth Edition, Penram Publications, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	