Program Name : **Computer Engineering Program Group**

Program Code : **CO/CM/CW**

Semester : **Fourth**

Course Title : **Microprocessors**

Course Code : **22415**

1. RATIONALE

Microprocessor is the main component of computer where 8086 is the base of all upward developed processors till current processors. This course will cover the basics of 8086 and its architecture along with instruction set, assembly language programming with effective use of procedure and macros. This course also covers the architectural issues such as instruction set program and data types. On top that, the students are also introduced to the increasingly important area of parallel organization. This subject serves as a basic to develop hardware related projects. This course will enable the students to inculcate assembly language programming concepts and methodology to solve problems.

2. COMPETENCY

The aim of this course is to help the student to attain the following ***industry identified*** competency through various teaching learning experiences:

- **Develop assembly level language programming using 8086.**

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following industry oriented COs associated with the above mentioned competency:

a. Analyze the functional block of 8086 microprocessor.
b. Write assembly language program for the given problem.
c. Use instructions for different addressing modes.
d. Develop an assembly language program using assembler.
e. Develop assembly language programs using procedures, macros and modular programming approach.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme											
				Theory						Practical					
L	T	P		Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
4	-	2	6	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(*): Under the theory PA; Out of 30 marks, 10 marks of theory PA are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

Legends: L-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** - Practical; **C** – Credit, **ESE** - End Semester Examination; **PA** - Progressive Assessment, **'#':** No Theory Examination

5. COURSE MAP(with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

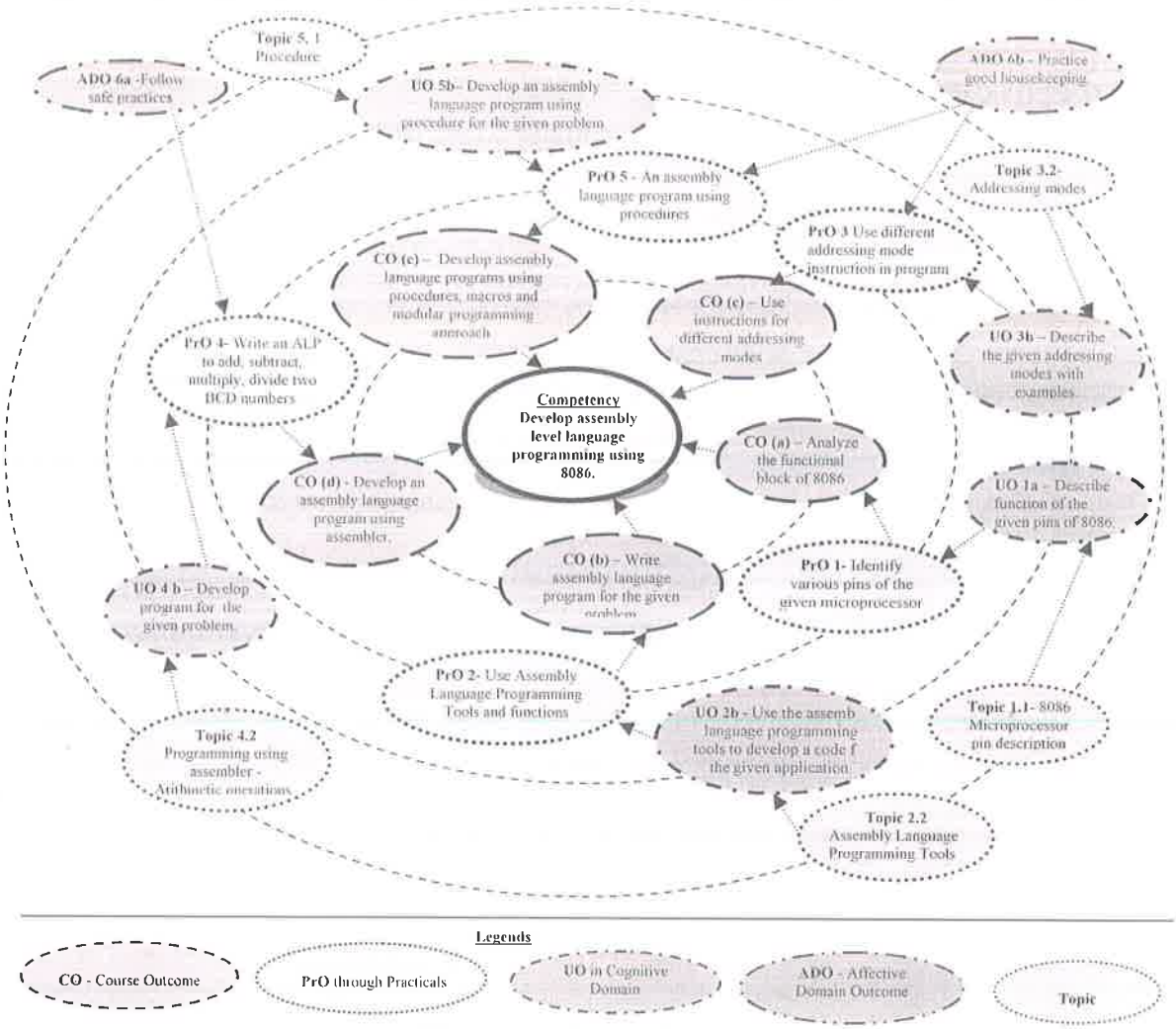

Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Identify various pins of the given microprocessor.	I	02*
2	Use Assembly Language Programming Tools and functions	II	02*
3	Use different addressing mode instruction in program (a) Write an Assembly Language Program (ALP) to add two given 8 and 16 bit numbers.	III	02*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	(b) Write an Assembly Language Program (ALP) to subtract two given 8 and 16 bit numbers.		
4	(a) Write an ALP to multiply two given 8 and 16 bit unsigned numbers. (b) Write an ALP to multiply two given 8 and 16 bit signed numbers.	III	02
5	(a) Write an ALP to perform block transfer data using string instructions (b) Write an ALP to perform block transfer data without using string instructions.	III	02
6	(a) Write an ALP to compare two strings without using string instructions. (b) Write an ALP to compare two strings using string instructions	III	02
7	(a) Write an ALP to divide two unsigned numbers (b) Write an ALP to divide two signed numbers	III	02
8	Write an ALP to add, subtract, multiply, divide two BCD numbers.	IV	02
9	Implement loop in assembly language program (a) Write an ALP to find sum of series of Hexadecimal Numbers. (b) Write an ALP to find sum of series of BCD numbers.	IV	02*
10	(a) Write an ALP to find smallest number from array of n numbers. (b) Write an ALP to find largest number from array of n numbers.	IV	02 *
11	(a) Write an ALP to arrange numbers in array in ascending order. (b) Write an ALP to arrange numbers in array in descending order.	IV	02
12	(a) Write an ALP to arrange string in reverse order (b) Write an ALP to find string length. (c) Write an ALP to concatenation of two strings.	IV	02
13	(a) Write an ALP to check a given number is ODD or EVEN. (b) Write an ALP to count ODD and/or EVEN numbers in array.	IV	02
14	(a) Write an ALP to check a given number is POSITIVE or NEGATIVE (b)Write an ALP to count POSITIVE and/or NEGATIVE numbers in array.	IV	02
15	(a) Write an ALP to count number of '1' in a given number (b) Write an ALP to count number of '0' in a given number	IV	02
16	An assembly language program using procedures (a) Write an ALP for addition, subtraction, multiplication and division. (b) Write an ALP using procedure to solve equation such as Z = (A+B)*(C+D)	V	02*
17	Write an assembly language program using macros. (a) Write an ALP for addition, subtraction, multiplication and division. (b) Write an ALP using MACRO to solve equation such as 01Z =	V	02*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	(A+B)*(C+D)		
	Total		**34**

Note

i. A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. A judicial mix of minimum 24 or more practical need to be performed, out of which, the practicals marked as '' are compulsory, so that the student reaches the 'Precision Level' of Dave's 'Psychomotor Domain Taxonomy' as generally required by the industry.*

ii. The 'Process' and 'Product' related skills associated with each PrO is to be assessed according to a suggested sample given below:

S.No.	Performance Indicators	Weightage in %
1	Able to write algorithm and draw flow chart.	20
2	Use Assembly language programming tools to create, edit, assemble and link the assembly language programs.	40
3	Debug, test and execute the programs	20
4	Able to answer oral questions.	10
5	Submission of report in time.	10
	Total	**100**

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

a. Follow safety practices.
b. Practice good housekeeping.
c. Demonstrate working as a leader/a team member.
d. Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year
- 'Organizing Level' in 2nd year and
- 'Characterizing Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	PrO. S. No.
1	Hardware: Personal computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	For all Experiments

S. No.	Equipment Name with Broad Specifications	PrO. S. No.
2	Operating system: Windows XP/Windows 7 onward	
3	Software: Editor: EDIT, NOTEPAD Assembler: TASM/MASM Linker: TLINK/LINK Debugger: TD/Debug of Windows Operating System	

8. UNDERPINNING THEORY COMPONENTS

The following topics are to be taught and assessed in order to develop the sample UOs given below for achieving the COs to attain the identified competency. More UOs could be added.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I 8086-16 Bit Microprocessor	1a. Describe function of the given pin of 8086. 1b. Explain with sketches the working of given unit in 8086 microprocessor. 1c. State functions of the given registers of 8086 microprocessor. 1d. Calculate the physical address for the given segmentation of 8086 microprocessor.	1.1 8086 Microprocessor: Salient features, Pin descriptions 1.2 Architecture of 8086: Functional Block diagram, Register organization 1.3 Concepts of pipelining 1.4 Memory segmentation, Physical memory addresses generation
Unit– II The Art of Assembly Language Programming	2a. Describe the given steps of program development /execution. 2b. Write steps to develop a code for the given problem using assembly language programming. 2c. Use relevant command of debugger to correct the specified programming error. 2d. Describe function of the given assembler directives with example.	2.1 Program development steps:Defining problem and constrains, Writing Algorithms, Flowchart, Initialization checklist, Choosing instructions, Converting algorithms to assembly language programs 2.2 Assembly Language Programming Tools: Editors, Assembler, Linker, Debugger 2.3 Assembler directives
Unit– III Instruction Set of 8086 Microprocessor	3a. Determine the length of the given instruction. 3b. Describe the given addressing modes with examples. 3c. Explain the operation performed by the given instruction during its execution. 3d. Identify the addressing modes in the given instructions.	3.1 Machine Language Instruction format 3.2 Addressing modes 3.3 Instruction set, Groups of Instructions: Arithmetic instructions, Logical Instructions, Data transfer instructions, Bit manipulation instructions, String Operation instructions, Program control transfer or branching instructions, Process control instructions

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit-IV Assembly Language Programming	4a. Use the given model of assembly language programs for the given problem. 4b. Develop the relevant program for the given problem. 4c. Apply relevant control loops in the program for the given problem. 4d. Use string instructions for the given strings/block to manipulate its elements.	4.1 Model of 8086 assembly language programs 4.2 Programming using assembler : Arithmetic operations on Hex and BCD numbers, Sum of Series, Smallest and Largest numbers from array, Sorting numbers in Ascending and Descending order, Finding ODD,EVEN, Positive and Negative numbers in the array, Block transfer, String Operations - Length, Reverse, Compare, Concatenation, Copy, Count Numbers of '1' and '0' in 16 bit number.
Unit –V Procedure and Macro	5a. Apply the relevant 'parameter-passing' method in the given situation. 5b. Develop an assembly language program using the relevant procedure for the given problem. 5c. Develop an assembly language program using MACROS for the given problem. 5d. Compare procedures and macros on the basis of the given parameters.	5.1 Procedure: Defining and calling Procedure - PROC, ENDP, FAR and NEAR Directives; CALL and RET instructions; Parameter passing methods, Assembly Language Programs using Procedure 5.2 Macro: Defining Macros, MACRO and ENDM Directives, Macro with parameters, Assembly Language Programs using Macros

***Note**: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'.*

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	8086- 16 Bit Microprocessor	08	02	02	10	14
II	The Art of Assembly Language Programming	12	-	02	06	08
III	Instruction Set of 8086 Microprocessor	16	02	04	10	16
IV	Assembly Language Programming	16	02	02	16	20
V	Procedure and Macro	12	02	02	08	12
Total		**64**	**8**	**12**	**50**	**70**

***Legends:** R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)*
***Note**: This specification table provides general guidelines to assist students for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual*

distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related ***co-curricular*** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

a. Prepare journals based on practical performed in laboratory.
b. Library/E-Book survey regarding assembly language programming used in Computer industries.
c. Prepare power point presentation for showing different types of Assembly language Programming Applications.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

a. Massive open online courses (***MOOCs***) may be used to teach various topics/sub topics.
b. ***'L' in item No. 4*** does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
c. About ***15-20% of the topics/sub-topics*** which is relatively simpler or descriptive in nature is to be given to the students for ***self-directed learning*** and assess the development of the COs through classroom presentations (see implementation guideline for details).
d. With respect to item No.10, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
e. Guide student(s) in undertaking micro-projects.
f. No. of practical's selection to be performed should cover all units.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be individually undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should not exceed three.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than 16 (sixteen) student engagement hours during the course. The student ought to submit micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects are given here. Similar micro-projects could be added by the concerned faculty:

a. **Conversion of Number System** - Each group will develop a program to perform following operations (Any One):
   i. Convert Hexadecimal number to equivalent BCD.

ii. Convert BCD number to Equivalent Hexadecimal Number

b. **Array** - Each group will develop a program to perform following operations (Any One):
   i. Separate ODD and EVEN number from given array, store them in separate array and find the sum.
   ii. Separate ODD and EVEN number from given array, store them in separate array and find the smallest or largest among them.
   iii. Separate ODD and EVEN number from given array, store them in separate array and sort numbers in ascending or descending order.

c. **Basic mathematic functions** - Each group will develop a program to perform following operations (Any One):
   i. Generate Fibonacci Series
   ii. Find Factorial of Number

d. **String Manipulation project** - Each group will develop a program to perform following operations (Any One):
   i. Convert lower case string to upper case string and vice versa.
   ii. Check string for Palindromes.
   iii. Search given character in string; find how many times it is present in string and its position.

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Microprocessor and interfacing (programming and hardware)	Hall, Douglas V.	McGraw Hill Education, New Delhi, 2015, ISBN-13: 978-0070257429
2	The 8088 and 8086 Microprocessors	Triebel, Walter A., Singh, Avtar	Pearson Publications, New Delhi, 2015, ISBN 13: 9780130930811
3	Microprocessors and Microcontrollers	Latha, C., Murugeshwari, B.	SCITECH Publications, Chennai, 2015, ISBN: 978-81-8371-702-1

14. SUGGESTED SOFTWARE/LEARNING WEBSITES

a. Assembler TASM/MASM, Linker LINK/TLINK, Debugger OS Debuge/TD
b. www.intel.com
c. www.pcguide.com/ref/CPU
d. www.CPU-World.com/Arch/
e. www.techsource .com / engineering- parts/microprocessor.html
f. https://www.elprocus.com/8051-assembly-language-programming/
g. https://www.tutorialspoint.com/assembly_programming/
h. http://www.slideshare.net/search/slideshow?searchfrom=headerandq=assembly+language+programming+of+8086andud=anyandft=allandlang=**andsort=