

LABORATORY ASSIGNMENT /SHEET /JOB /PROJECT ACTIVITY PLANNING (LP)

Academic Year: 2026-27(ODD)

Date: 01/07/2026

Institute Name: K. K. Wagh Polytechnic, Nashik

MSBTE Code: 0078

Program and Code: Information Technology

Course Code & Abbr.: 313305 (DTM)

Course Name: Digital Techniques & Microprocessors

Name of Faculty: Mr. R.J.Shinde

Class: SYIF-PARAM

Semester: 3rd

Scheme: K

Course Index: 304 **Learning Hrs.** 32

• Teaching-Learning & Assessment Scheme:

Course Code	Course Title	Abbr	Course Category/s	Learning Scheme					Credits	Assessment Scheme												Total Marks
				Actual Contact Hrs./Week			SLH	NLH		Paper Duration	Theory				Based on LL & TL				Based on SL			
															Practical							
				CL	TL	LL	FA-TH	SA-TH			Total	FA-PR		SA-PR		SLA						
												Max	Min	Max	Min	Max	Min	Max	Min			
313305	DIGITAL TECHNIQUES AND MICROPROCESSORS	DTM	AEC	3	-	2	1	6	3	3	30	70	100	40	25	10	25@	10	25	10	175	

Abbreviations: CL- Class Room Learning , TL- Tutorial Learning, LL-Laboratory Learning, SLH-Self Learning Hours, NLH-Notional Learning Hours, FA - Formative Assessment, SA -Summative assessment, IKS – Indian Knowledge System, SLA - Self Learning Assessment
Legends: @ Internal Assessment, # External Assessment, *# On Line Examination, @\$ Internal Online Examination

• COURSE LEVEL LEARNING OUTCOMES (COS)

Students will be able to achieve & demonstrate the following COs on completion of course based learning

- CO1 - Test logic gates and digital systems.
- CO2 -Use basic combinational and sequential logic circuits employing digital ICs.
- CO3 - Perform operations on registers using 8086 instructions.
- CO4 - Use 8086 microprocessor environment to build and execute assembly language programs.
- CO5 - Develop assembly language programming in 8086 to implement loops and branching instructions.

• COs, Practical Laboratory Learning Outcome (LLOs) and Mapping:

Sr. No	Pr. No	COs	LLO	Name of Experiments/Assignment/ Sheet/ Job/ Project Activity	Planned Date A/B/C		Actual date of Performance	Remark
					From	To		
1	1	CO 1	LLO	* Verification of truth table of	A-07/07/2026	A-07/07/2026	A-	
			1.1 LLO	basic logic gates, special logic gates and Identify various	B-03/07/2026	B-03/07/2026	B-	
			1.2 LLO	Logic gate ICs.	C-02/07/2026	C-02/07/2026	C-	
			1.3 LLO					
2	2	CO	1.4 LLO	Implementation and verification of expression using universal logic gate ICs	A-14/07/2026	A-14/07/2026	A-	
			2.1 LLO		B-10/07/2026	B-10/07/2026	B-	
					C-09/07/2026	C-09/07/2026	C-	

3	3	CO 2	LLO 3.1 LLO 3.2	Verification of De-Morgan's theorems using basic logic gates	A-21/07/2026 B-17/07/2026 C-16/07/2026	A-21/07/2026 B-17/07/2026 C-16/07/2026	A- B- C-		
4	4	CO 2	LLO 4.1 LLO 4.2	* Conversion of expression to Sum-of- Product (SOP) and Product-of-Sum (POS)	A-28/07/2026 B-24/07/2026 C-23/07/2026	A-28/07/2026 B-24/07/2026 C-23/07/2026	A- B- C-		
5	5	CO 2	LLO 5.1 LLO 5.2	* Implement Multiplexer and Demultiplexer logic (The practical may be performed using virtual lab)	A-04/08/2026 B-31/07/2026 C-30/07/2026	A-04/08/2026 B-31/07/2026 C-30/07/2026	A- - C-		
6	7	CO 3	LLO 7.1	* Verification of contents of general purpose, segment registers, flags and memory locations of different segments during execution of the program	A-18/08/2026 B-07/08/2026 C-06/08/2026	A-18/08/2026 B-07/08/2026 C-06/08/2026	A- B- C-		
7	8	CO 4	LLO 8.1 LLO 8.2	* Assembly language programming for addition and subtraction for hexadecimal numbers	A-25/08/2026 B-14/08/2026 C-13/08/2026	A-25/08/2026 B-14/08/2026 C-13/08/2026	A- B- C-		
8	9	CO 4	LLO 9.1 LLO 9.2	Apply assembly language programming logic for addition, subtraction and multiplication for BCD numbers.	A-01/09/2026 B-21/08/2026 C-20/08/2026	A-01/09/2026 B-21/08/2026 C-20/08/2026	A- B- C-		
9	10	CO 4	LLO 10.1	* Assembly language programming for multiplication and division	A-08/09/2026 B-28/08/2026 C-27/08/2026	A-08/09/2026 B-28/08/2026 C-27/08/2026	A- B- C-		
10	12	CO 5	LLO 12.1 LLO 12.2	* Assembly language programming for sorting of data	A-15/09/2026 B -04/09/2026 C-03/09/2026	A-15/09/2026 B -04/09/2026 C-03/09/2026	A- B- C-		
11	13	CO 5	LLO 13.1	Assembly language programming for transfer of block of data	A-22/09/2026 B-11/09/2026 C-10/09/2026	A-22/09/2026 B-11/09/2026 C-10/09/2026	A- B- C-		
12	15	CO 5	LLO 15.1 LLO 15.2	* Implement shift and rotate instructions on given data	A-29/09/2026 B-18/09/2026 C-17/09/2026	A-29/09/2026 B-18/09/2026 C-17/09/2026	A- B- C-		
13		CO 2	LLO 2.1 LLO 2.2	Implementation and Verification of expression using Special Logic Gate ICs	A-06/10/2026 B-09/10/2026 C-24/09/2026	A-06/10/2026 B-09/10/2026 C-24/09/2026	A- B- C-		

ASSESSMENT METHODOLOGIES/TOOLS

A. Formative assessment (Assessment for Learning) (FA-TH)

Two offline unit tests of 30 marks and average of two unit test.

For formative assessment of Laboratory learning 25 marks marks will be consider for out of 30 marks Each practical will be assessed considering 60% weightage to process, 40% weightage to product.

B. Summative Assessment (Assessment of Learning) (SA-TH)

End semester assessment of 70 marks.

End semester summative assessment of 25 marks for laboratory learning

SUGGESTED MICRO PROJECT / ASSIGNMENT/ ACTIVITIES FOR SPECIFIC LEARNING /SKILLS DEVELOPMENT (SELF LEARNING)

Micro project

- Find the factorial of a given number using 8086 assembly language programming.
- Separate odd and even numbers from a given array using an assembly language program.
- Design shift register using JK Flipflop.
- Design a Burglar alarm using electronic components and digital ICs.

Assignment

- Write an assembly language program using 8086 to generate the Fibonacci series.
- Draw and implement the circuit on the breadboard for 1:16 DEMUX using 1:8 DEMUX. State the ICs used alongwith their description.

Self Learning Activity

- Develop an assembly language program to add 8-bit and 16-bit Unsigned numbers (using procedure).
- Write an assembly language program to add and subtract two BCD numbers(using MACRO).
- Write an ALP to multiply two BCD numbers (using MACRO).

Mr. R.J.Shinde
(Name & Signature of Staff)

Prof. M. S. Karande
(Name & Signature of HOD)