II Tamaso ma Jyotirgamaya II
Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering & Technology

Someshwarnagar, Tal.Baramati, Dist. Pune (Pin : 412 306) Maharashtra, India
Approved by AICTE New Delhi, Recognised by Govt. of Maharashtra &
Affilited to Savitribai Phule Pune University, Pune, Id. No. PU/PN. Engg./445/2012
Ph : (02112) 253185, Fax : (02112) 283185

Ref. No. : SPCE/ / /2022 - 2023 Date :30 / 5 /2023

6.5.2 The institution reviews its teaching learning process, structures & methodologies of operations and learning outcomes at periodic intervals through IQAC set up as per norms and recorded the incremental improvement in various activities For second and subsequent cycles - Incremental improvements made for the preceding five years with regard to quality and post accreditation quality initiatives.

Sr. No	Two Examples of Institutional Reviews
1	Course File
2	Evaluation Process

SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY
SOMESHWARNAGAR
DTE CODE
EN6795

PRINCIPAL
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY
SOMESHWARNAGAR, TAL-BARAMATI, DIST-PUNE (Pin-412 306)

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology

(Formerly known as Someshwar Engineering College)

Gat. No. 53 Waghalwadi, Post. Someshwarnagar, Tal: Baramati,
Dist: Pune-412306, Maharashtra, India

Phone: 02112-283185| **Email:** principal@secsomeshwar.ac.in| **Website:** www.secsomeshwar.ac.in

Approved by AICTE, New Delhi and Recognized by Govt. of Maharashtra, Mumbai, Affilated to Savitribai Phule Pune University, Pune, ID No.: PU/PN/ENG/445/2012,
DTE Code: EN- 6795

Department Of Computer Engineering AY 2022-23

Semester I/II

Course File Completion Report

Subject: (Mrs. Gahadage S.S.) MP

Class: S.E.

Authority	Sign	Remark
Academic Coordinator		C
Head Of Department		C
Principal		C

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

ACADEMIC CALENDARS

SHRI SOMESHWAR SHIKSHAN PRASARAK MANDAL'S

SHARADCHANDRA PAWAR CLLEGE OFENGINEERING AND TECHNOLOGY, SOMESHWARNAGAR

Record No.:- ACD/R/01
Revision:- 00
Date:-16/06/2014

A.Y.:-2022-2023 | **ACADEMIC CALENDER** | **Semester:-II**

Week No.	Month	Week Days							No. of Working Days	Events
		MON	TUE	WED	THU	FRI	SAT	SUN		
1	JAN	23	24	25	26	27	28	29	5	Commencement of Teaching Sem II on23 Jan. TE and BE
2		30	31						2	Course File Checking on 25th Jan
	FEB			1	2	3	4	5	4	Republic Day Celebration on 26th Jan.
3		6	7	8	9	10	11	12	6	Display 1st week attendance on 31st Jan.
4		13	14	15	16	17	18	19	5	Commencement of Teaching Sem II on 6th Feb for SE
5		20	21	22	23	24	25	26	6	Sharad Somotsav 2K23 11th Feb to 17th Feb
6		27	28						2	Science Day Celebration on 28th Feb.
	MAR			1	2	3	4	5	4	Intenational Womens Day celibration on 8th March and 10th March
7		6	7	8	9	10	11	12	5	
8		13	14	15	16	17	18	19	6	Defaulter list of SE, TE and BE
9		20	21	22	23	24	25	26	5	Internal FeedBack
10		27	28	29	30	31			5	Unit Test I For S.E,T.E & B.E
	APR						1	2	1	Result analysis of Unit Test- I on 20th March. &Parent teachers meet online on 25th March
11		3	4	5	6	7	8	9	4	Commencement of Teaching Sem II on 1st April. for FE
12		10	11	12	13	14	15	16	5	SPPU In-Sem exam in 3rd April to 10th April
13		17	18	19	20	21	22	23	5	Counseling of Detained student's parents on 12th April
14		24	25	26	27	28	29	30	6	
15	MAY	1	2	3	4	5	6	7	4	SPPU Oral and Practical Examination
16		8	9	10	11	12	13	14	6	Submission and term work completion on 5th May
17		15	16	17	18	19	20	21	6	Conclusion of Term for BE and TE on 20th May
18		22	23	24	25	26	27	28	6	Conclusion of Term for SE on 31st May
19		29	30	31					3	SPPU Theory Examination
No. of Week Days		19	19	19	18	18	18			

HOLIDAYS
26/01 Republic Day
19/02 Chhatrapati Shivaji Maharaj Jayanti
18/02 Mahashivratri
07/03 Dhulivandan
22/03 Gudhipadawa
30/03 Ram Navami
04/04 Mahavir Jayanti
07/04 Good Friday
14/03 Dr. BabasahebAmbedkar Jayanti
22/04 Akshay tritiya
01/05 Maharshtra Din
05/05 Buddha Porni

NOTE:-

Principal Meet will be conduct as and when required

HOD Meet will be conduct as and when required

GFM Meet will be conduct as and when required

Continuous assessment of assignment/ experiments /project/seminar by respective Guide/Subject teacher once in month.

Academic Coordinator

Principal

	SHRI SOMESHWAR SHIKSHAN PRASARAK MANDAL'S SHARADCHANDRA PAWAR CLLEGE OFENGINEERING AND TECHNOLOGY, SOMESHWARNAGAR	Record No.:- ACD/R/01 Revision:- 00 Date:-16/06/2014
A.Y.:-2022-2023	DEPARTMENT OF COMPUTER ENGINEERING ACADEMIC CALENDER	Semester:-II

Week No.	Month	Week Days							Events
		MON	TUE	WED	THU	FRI	SAT	SUN	
1	JAN	23	24	25	26	27	28	29	Commencement of Teaching Sem II on23 Jan. TE and BE
2		30	31						Course File Checking on 25th Jan
	FEB			1	2	3	4	5	Republic Day Celebration on 26th Jan.
3		6	7	8	9	10	11	12	Display 1st week attendance on 31st Jan.
4		13	14	15	16	17	18	19	Commencement of Teaching Sem II on 6th Feb for SE
5		20	21	22	23	24	25	26	Sharad Somotsav 2K23 11th Feb to 17th Feb
6		27	28						22th Feb Guest lecture on Womens Health Issue
	MAR			1	2	3	4	5	Science Day Celebration on 28th Feb.
7		6	7	8	9	10	11	12	9th March Guest lecture on Latest Programming Technologies
8		13	14	15	16	17	18	19	Intenational Womens Day celibration on 8th March and 10th
9		20	21	22	23	24	25	26	March
10		27	28	29	30	31			Defaulter list of SE, TE and BE
	APR						1	2	Internal FeedBack
11		3	4	5	6	7	8	9	Unit Test I For S.E,T.E & B.E
12		10	11	12	13	14	15	16	Result analysis of Unit Test- I on 20th March. &Parent
13		17	18	19	20	21	22	23	Commencement of Teaching Sem II on 1st April. for FE
14		24	25	26	27	28	29	30	SPPU In-Sem exam in 3rd April to 10th April
15		1	2	3	4	5	6	7	Counseling of Detained student's parents on 12th April
16		8	9	10	11	12	13	14	
17	MAY	15	16	17	18	19	20	21	SPPU Oral and Practical Examination
18		22	23	24	25	26	27	28	Submission and term work completion on 5th May
19		29	30	31					Conclusion of Term for BE amd TE on 20th May
No. of Week Days		19	19	19	18	18	18		Conclusion of Term for SE on 31st May
									SPPU Theory Examination

HOLIDAYS
26/01 Republic Day
19/02 Chhatrapati Shivaji Maharaj Jayanti
18/02 Mahashivratri
07/03 Dhulivandan
22/03 Gudhipadawa
30/03 Ram Navami
04/04 Mahavir Jayanti
07/04 Good Friday
14/03 Dr. BabasahebAmbedkar Jayanti
22/04 Akshay tritiya
01/05 Maharshtra Din
05/05 Buddha Porni

NOTE:-

Principal Meet will be conduct as and when required

HOD Meet will be conduct as and when required

GFM Meet will be conduct as and when required

Continuous assessment of assignment/ experiments /project/seminar by respective Guide/Subject teacher once in month.

Academic Coordinator

HOD

Savitribai Phule Pune University

(Formerly University of Pune)

Circular No. 302 of 2022

Important Notification

Revised Dates of Commencement and Conclusion of terms of U.G. / P.G. Courses for the Academic Year 2022-23 for Affiliated Colleges / Recognised Institutes.

In reference to the earlier circular issued by the University bearing no. 173 dated 10.06.2022 the dates of commencement and conclusion of First Term and Second Term in the academic calendar for the academic year 2022-23, for the following courses are being revised as under.

Sr No	Name of the Courses , Faculties & Year	2022 - 2023			
		First Term		Second Term	
		Commencement	Conclusion	Commencement	Conclusion
1	Science & Technology				
	Science	20/06/2022	30/11/2022	26/12/2022	04/05/2023
	B.Engineering : II	17/08/2022	10/12/2022	02/01/2023	29/04/2023
	B.Engineering : III IV	18/07/2022	30/11/2022	02/01/2023	29/04/2023
	M.Engineering : II	18/07/2022	12/11/2022	09/01/2023	06/05/2023
	B.Architecture : II	08/08/2022	04/12/2022	19/12/2022	04/05/2023
	B.Architecture : III IV V	20/06/2022	08/11/2022	30/12/2022	15/05/2023
	M.Architecture:II	19/09/2022	07/01/2023	23/01/2023	20/05/2023
	B. Pharmacy: II III	01/08/2022	10/12/2022	02/01/2023	10/05/2023
	B. Pharmacy: IV	15/07/2022	03/12/2022	02/01/2023	10/05/2023
	M. Pharmacy : II	01/08/2022	10/12/2022	26/12/2022	30/06/2023
2	Commerce & Management				
	Commerce	20/06/2022	30/11/2022	26/12/2022	04/05/2023
	MBA II (Including SIP project of 8	01/09/2022	30/01/2023	15/02/2023	26/05/2023
	MCA II	01/09/2022	16/12/2022	02/01/2023	15/04/2023
	BHMCT II III IV	01/09/2022	16/12/2022	02/01/2023	15/04/2023
3	Humanities				
	Arts	20/06/2022	30/11/2022	26/12/2022	04/05/2023
	Mental Moral and Social Sciences				
	L.L.B. II	31/10/2022	31/01/2023	06/02/2023	15/05/2023
	L.L.B. III	04/07/2022	12/12/2022	08/01/2023	15/05/2023
	B.A. L.L.B. II	31/10/2022	31/01/2023	06/02/2023	15/05/2023
	B.A. L.L.B. III IV V	04/07/2022	12/12/2022	08/01/2023	15/05/2023
4	Inter disciplinary Studies				
	Education : II	15/09/2022	06/01/2023	17/01/2023	10/05/2023
	Physical Education : II	15/09/2022	06/01/2023	17/01/2023	10/05/2023
	B.Lib & M.Lib	15/07/2022	30/11/2022	02/01/2023	04/05/2023
	Fine Arts & Performing Art	20/06/2022	30/11/2022	26/12/2022	04/05/2023
	Journalism PG	15/07/2022	30/11/2022	02/01/2023	04/05/2023

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

MASTER TIME TABLE

SOMESHWAR SHIKSHAN PRASARAK MANDAL'S

Record No:-

SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING AND TECHNOLOGY, SOMESHWARNAGAR

Revision:-

Date:-

TIME TABLE

Department: COMPUTER **Class:** SE,TE,BE **Semester:** II **Academic Year:** 2022-23 **W.E.F.:** 23/01/2023

DAY	CLASS	9:00-10:00	10:00-11:00	11.00-11:45	11:45-12:45	12:45-1:45	1:45-2:00	2.00-3.00	3:00-4:00
MONDAY	SE	B1-PBL/ B2-DSA/ B3-PBL/ B4-DSA		LUNCH BREAK	M-III	SE	SHORT BREAK	PPL	OOP
	TE	AI	SMA		B1-DSBDA/B2-INTRENSHIP/B3-AI/B4-DSBDA			WT	DSBDA
	BE	HPC	DL		LIBRARY	NLP		PROJECT WORK	
TUESDAY	SE	B1-MP/ B2-DSA/B3-PBL/B4-DSA			M-III	PPL		OOA	SE
	TE	AI	DSBDA		B1-INTRENSHIP/B2-SMA/B3-WT/B4-DSBDA			LIBRARY	
	BE	HPC	NLP		LAB-V(DL LAB)			DL	NPTL
WEDNESDAY	SE	B1-OSA/ B2-PBL/ B3-MP/ B4-PBL			SE	DSA		PPL	CCC
	TE	DSBDA	AI		B1-WT/B2-DSBDA/B3-SMA/B4-INTRENSHIP			WT	SMA
	BE	BI	LIBRARY		LAB-VI (NLP LAB)			LAB-VI (BI LAB)	
THURSDAY	SE	B1-DSA/ B2-PBL/ B3-DSA/ B4-MP			M-III	DSA		MP	NPTL
	TE	WT	DSBDA		B1-SMA/B2-DSBDA/B3-INTERNSHIP/B4-WT			SMA	AI
	BE	HPC	IE		DL	NLP		PROJECT WORK	
FRIDAY	SE	B1-PBL/ B2-MP/ B3-OSA/ B4-PBL			M-III TUTORIAL			MP	NPTL
	TE	WT	SMA		B1-AI/B2-WT/B3-DSBDA/B4-SMA			B1-DSBDA/B2-AI/B3-DSBDA/B4-AI	
	BE		IE		PROJECT WORK			LAB-V (HPC LAB)	
SATURDAY	SE	LIBRARY			LIBRARY				
	TE	LIBRARY			INTRENSHIP				
	BE	PROJECT WORK			LIBRARY				

Time Table Incharge
Prof. Bhapkar.A.D

Head of the Dept.
Prof. Shah S.N.

Principal
Prof. Deokar S.A.

	SOMESHWAR SHIKSHAN PRASARAK MANDAL'S	Record No:- ACD/R/03
	Sharadchandra Pawar College of Engineering and Technology Someshwarnagar	Revision:-
		Date:-

Academic Year: 2022-23

INDIVIDUAL TIME TABLE

Department: Computer **Staff Name: Prof. Ghadage S. S.**

Day/Time	9:00-10:00	10:00-11:00	11.00-11:45	11:45-12:45	12:45-01:45	01:45-2:00	2.00-3.00	3.00-4.00
MONDAY		DL	LUNCH BREAK			SHORT BREAK		MP
TUESDAY	B1-MP/ B2-DSA/B3-PBL/B4-DSA			LAB-V(DL LAB)			DL	
WEDNESDAY	B1-DSA/ B2-PBL/B3-MP/B4-PBL							
THURSDAY	B1-DSA/ B2-PBL/B3-DSA/B4-MP			DL			MP	
FRIDAY	B1-PBL/ B2-MP/B3-DSA/B4-PBL						MP	
SATURDAY								

1. Microprocessor
2. Deep Learning

Prof. Ghadage S. S.
Subject Teacher

HEAD OF DEPARTMENT
COMPUTER ENGINEERING

Prof. Shah S.N
Head of Department

Dr. Deokar S.A
Principal

PRINCIPAL
Sharadchandra Pawar College of Engineering & Technology
Someshwarnagar, Tal. Baramati, Dist. Pune (Pin : 412 306)

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

VISION-MISSION

Institute Vision and Mission

Vision

✓ Our vision is to achieve excellence in technical education and make the engineers for socio-economic development of rural India.

Mission

- To prepare rural students for a productive and rewarding career in engineering profession.
- To provide students with comprehensive knowledge and fundamentals of engineering.
- To create barrier free environment through technical education in rural area.
- Development of technical human resource for socio-economic development of rural India.
- To impart value education and skill through technical education.

Vision-Mission (Computer Department)

Vision

✓ To be the front runner in Computer Engineering education and to foster the students into globally competent professionals with expertise in software development and aptitude for research and ethical values.

Mission

- Provide the ambience to become industry ready Professionals, Researchers and Entrepreneurs by offering courses on cutting edge technology and advanced laboratory courses for the students.
- Provide a conductive environment for faculty to engage in and train students in progressive and convergent research themes by establishing Centre's of Excellence.
- Impart high quality experiential learning to get expertise in modern software tools and to cater to the real time requirements of the industry.
- Inculcate problem solving and team building skills and promote lifelong learning with a sense of societal and ethical responsibilities.
- Offer continuing education programmers in the emerging areas for the benefit of stakeholders.

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

PEO's &PO's

PROGRAMME EDUCATIONAL OBJECTIVES:

PEO 1: Student should be able to analyse, formulate and solve/design engineering/real life problems based on his/her solid foundation in science and engineering.

PEO 2: Student should be able to apply their knowledge and skills for a successful career in diverse domains viz., industry/technical, research and higher education/academia with social consciousness.

PEO 3: Student should be able to work in team, multidisciplinary approach, professional development through continued education and an ability to relate engineering issues to broader social context.

PEO 4: Students grow professionally in their career through continuous development of technical skills, improvement of professional efficiency, and assumption of roles of responsibility in professional service.

Programme Outcomes (Computer Department)

- PO 1: Engineering Knowledge: Apply the knowledge of mathematics, science, Engineering fundamentals, and Computer Engineering to the solution of engineering problems.
- PO 2: Problem analysis: Identify, formulate, review research literature, and analyse complex Engineering problems reaching substantiated conclusions using first principles of engineering sciences.
- PO 3: Design/development of solutions: Design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration.
- PO 4: Conduct investigations of complex problems: Use research based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex Engineering activities with an understanding of the limitations.
- PO 6: The Engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Engineering practice.
- PO 7: Environment and sustainability: Understand the impact of the professional Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable developments.
- PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the Engineering practice.
- PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO 10: Communication: Communicate effectively on complex Engineering activities with the Engineering Community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO 11: Project management and finance: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi- disciplinary environments.
- PO 12: Life -long learning: Recognize the need for, and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

COURSE OBJECTIVE & COURSE OUTCOME

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, Someshwarnagar

Department of Computer Engineering

Subject- Microprocessor **(2019 Pattern)** **Class-**SE

Course Objectives and Course Outcomes

1	**Course Objectives**	To learn and distinguish the architecture and programmer's model of advanced processor.
		To identify the system level features and processes of advanced processors.
		To acquaint the learner with application instruction set and logic to build assembly languageprograms.
2	**Course Outcomes**	Exhibit skill of assembly language programming for the application.
		Classify Processor architectures.
		Illustrate advanced features of 80386 microprocessor.
		Compare and contrast different processor modes.
		Use interrupts mechanism in applications.
		Differentiate between Microprocessors and Microcontrollers.
		Identify and analyze the tools and techniques used to design, implement, and debug microprocessor-based systems.

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

SYLLABUS COPY

Savitribai Phule Pune University

Second Year of Computer Engineering (2019 Course)

(With effect from Academic Year 2020-21)

Semester-III

Course Code	Course Name	Teaching Scheme (Hours/Week)			Examination Scheme and Marks						Credit Scheme			
		Lecture	Practical	Tutorial	Mid-Sem	End-Sem	Term work	Practical	Oral	Total	Lecture	Practical	Tutorial	Total
210241	Discrete Mathematics	03	-	-	30	70	-	-	-	100	03	--	-	03
210242	Fundamentals of Data Structures	03	-	-	30	70	-	-	-	100	03	-	-	03
210243	Object Oriented Programming (OOP)	03	-	-	30	70	-	-	-	100	03	-	-	03
210244	Computer Graphics	03	-	-	30	70	-	-	-	100	03	-	-	03
210245	Digital Electronics and Logic Design	03	-	-	30	70	-	-	-	100	03	-	-	03
210246	Data Structures Laboratory	-	04	-	-	-	25	50	-	75	-	02	-	02
210247	OOP and Computer Graphics Laboratory	-	04	-	-	-	25	25	-	50	-	02	-	02
210248	Digital Electronics Laboratory	-	02	-	-	-	25	-	-	25	-	01	-	01
210249	Business Communication Skills	-	02	-	-	-	25	-	-	25	-	01	-	01
210250	Humanity and Social Science	-	-	01	-	-	25	-	-	25	-	-	01	01
210251	Audit Course 3													
										Total Credit	**15**	**06**	**01**	**22**
	Total	**15**	**12**	**01**	**150**	**350**	**125**	**75**	**-**	**700**	**-**	**-**	**-**	**-**

Semester-IV

Course Code	Course Name	Teaching Scheme (Hours/Week)			Examination Scheme and Marks						Credit Scheme			
		Lecture	Practical	Tutorial	Mid-Sem	End-Sem	Term work	Practical	Oral	Total	Lecture	Practical	Tutorial	Total
207003	Engineering Mathematics III	03	-	01	30	70	25	-	-	125	03	--	01	04
210252	Data Structures and Algorithms	03	-	-	30	70	-	-	-	100	03	-	-	03
210253	Software Engineering	03	-	-	30	70	-	-	-	100	03	-	-	03
210254	Microprocessor	03	-	-	30	70	-	-	-	100	03	-	-	03
210255	Principles of Programming Languages	03	-	-	30	70	-	-	-	100	03	-	-	03
210256	Data Structures and Algorithms Laboratory	-	04	-	-	-	25	25	-	50	-	02	-	02
210257	Microprocessor Laboratory	-	02	-	-	-	25	-	25	50	-	01	-	01
210258	Project Based Learning II	-	04	-	-	-	50	-	-	50	-	02	-	02
210259	Code of Conduct	-	-	01	-	-	25	-	-	25	-	-	01	01
210260	Audit Course 4													
										Total Credit	**15**	**05**	**02**	**22**
	Total	**15**	**10**	**02**	**150**	**350**	**150**	**25**	**25**	**700**	**-**	**-**	**-**	**-**

Home

Savitribai Phule Pune University
Second Year of Engineering (2019 Course)
210254: Microprocessor

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	03	**Mid_Semester(TH):** 30 Marks **End_Semester(TH):** 70 Marks

Prerequisite Courses : 210248: Digital Electronics and Logic Design

Companion Course : 210258: Microprocessor Laboratory

Course Objectives:

The course is intended to provide practical exposure to the students on microprocessors, design and coding knowledge on 80386 and introduction to microcontrollers.

- To learn and distinguish the architecture and programmer's model of advanced processor.
- To identify the system level features and processes of advanced processors.
- To acquaint the learner with application instruction set and logic to build assembly language programs.

Course Outcomes:

After successful completion of the course, the learner will be able to-

CO1: **E**xhibit skill of assembly language programming for the application.
CO2: **Classify** Processor architectures.
CO3: **Illustrate** advanced features of 80386 Microprocessor.
CO4: **Compare** and **contrast** different processor modes.
CO5: **Use** interrupts mechanism in applications
CO6: **Differentiate** between Microprocessors and Microcontrollers.
CO7: **Identify** and **analyze** the tools and techniques used to design, implement, and debug microprocessor-based systems.

Course Contents

Unit I	Introduction to 80386	(07 Hours)

Brief History of Intel Processors, 80386 DX Features and Architecture, Programmers Model, Operating modes, Addressing modes and data types.

Applications Instruction Set: Data Movement Instructions, Binary Arithmetic Instructions, Decimal Arithmetic Instructions, Logical Instructions, Control Transfer Instructions, String and Character Transfer Instructions, Instructions for Block Structured Language, Flag Control Instructions, Coprocessor Interface Instructions, Segment Register Instructions, Miscellaneous Instructions.

#Exemplar/Case Studies	Study-Evolution of Microprocessor
***Mapping of Course Outcomes for Unit I**	CO1,CO2

Unit II	Bus Cycles and System Architecture	(07 Hours)

Initialization- Processor State after Reset. Functional pin Diagram, functionality of various pins, I/O Organization, Memory Organization (Memory banks), Basic memory read and writes cycles with timing diagram.

Systems Architecture- Systems Registers (Systems flags, Memory Management registers, Control registers, Debug registers, Test registers), System Instructions.

#Exemplar/Case Studies	Study-Motherboard of Computer and it's components.
***Mapping of Course Outcomes for Unit II**	CO3

Unit III	Memory Management	(08 Hours)

Home

Global Descriptor Table, Local Descriptor Table, Interrupt Descriptor Table, GDTR, LDTR, IDTR. Formats of Descriptors and Selector, Segment Translation, Page Translation, Combining Segment and Page Translation.		
#Exemplar/Case Studies	Try creating an animation by using any of /Study of the tools to create and access all the type of possible segments in 80386DX.	
***Mapping of Course Outcomes for Unit III**	CO1,CO2	
Unit IV	**Protection**	**(08 Hours)**
Need of Protection, Overview of 80386DX Protection Mechanisms: Protection rings and levels, Privileged Instructions, Concept of DPL, CPL, RPL, EPL. Inter privilege level transfers using Call gates, Conforming code segment, Privilege levels and stacks. Page Level Protection, Combining Segment and Page Level Protection.		
#Exemplar/Case Studies	Study about- can the security of the system be compromised using CALL gates?	
***Mapping of Course Outcomes for Unit IV**	CO4, , CO6	
Unit V	**Multitasking and Virtual 8086 Mode**	**(08Hours)**
Multitasking- Task State Segment, TSS Descriptor, Task Register, Task Gate Descriptor, Task Switching, Task Linking, Task Address Space. **Virtual Mode** – Features, Memory management in Virtual Mode , Entering and leaving Virtual mode.		
#Exemplar/Case Studies	Study about multitasking implemented by using timing interrupt generated by internal clock of the system. Consider three different tasks: One displaying a string at first row accessing VRAM directly; Second Blinking the string with certain time interval and; Third clearing the screen.	
***Mapping of Course Outcomes for Unit V**	CO4, CO5, CO6	
Unit VI	**Interrupts, Exceptions, and Introduction to Microcontrollers**	**(07 Hours)**
Interrupts and Exceptions: Identifying Interrupts, Enabling and Disabling Interrupts, Priority among Simultaneous Interrupts and Exceptions, Interrupt Descriptor Table (IDT), IDT Descriptors, Interrupt Tasks and Interrupt Procedures, Error Code, and Exception Conditions. **Introduction to Microcontrollers**: Architecture of typical Microcontroller, Difference between Microprocessor and Microcontroller, Characteristics of microcontrollers, Application of Microcontrollers.		
#Exemplar/Case Studies	Try building a Minimum System using 8051 microcontroller (Provide complete architecture and component selection with rationale). Indicate Memory Map explicitly.	
***Mapping of Course Outcomes for Unit VI**	CO4,CO6, CO7	

Learning Resources

Text Books:

1. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2 Edition, 2006 ISBN 0-07-100462-9
2. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill,2004 ISBN 0-07-463841-6
3. Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630-011, December 1995.
4. Intel 80386 Hardware Reference Manual 1986, Intel Corporation, Order no.: 231732-001, 1986.
5. James Turley- "Advanced 80386 Programming Techniques", McGraw-Hill, ISBN: 10:0078813425, 13: 978-0078813429.

Home

Reference Books:

1. Chris H. Pappas, William H. Murray, "80386 Microprocessor Handbooks", McGraw-Hill Osborne Media, ISBN-10: 0078812429, 13: 978-0078812422.
2. Walter A. Triebel, "The 80386Dx Microprocessor: Hardware", Software, and Interfacing, Pearson Education, ISBN: 0137877307, 9780137877300.
3. Brey, Barry B, "8086/8088, 80286, 80386 and 80486 Assembly Language Programming", Prentice Hall, ISBN: 13: 9780023142475.
4. Mohammad Rafiquzzaman, "Microprocessors: Theory and Applications: Intel and Motorola", Prentice Hall, ISBN:-10:0966498011, 13:978:0966498011.
5. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012.
6. Assembly Language Step-by-step: Programming with Linux, 3rd Edition, Jeff Duntemann, Wiley ISBN:-10 0470497025, ISBN-13: 978-0470497029, 2009.

Intel 80386 Programmer's Reference Manual:

- http://intel80386.com/386htm/toc.htm
- https://css.csail.mit.edu/6.858/2014/readings/i386.pdf

MOOC/ Video Lectures available at:

- https://nptel.ac.in/courses/106/108/106108100/
- https://nptel.ac.in/courses/108/107/108107029/

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	2	-	-	-	-	-	-	-	-
CO2	2	-	1	-	-	-	-	-	-	-	-	-
CO3	2	-	2	-	-	-	-	-	-	-	-	-
CO4	2	-	2	-	-	-	-	-	-	-	-	-
CO5	2	-	2	-	-	-	-	-	-	-	-	-
CO6	2	1	-	-	-	-	-	-	-	-	-	-
CO7	2	1	1	1	-	-	-	-	-	-	-	-

Savitribai Phule Pune University
Second Year of Computer Engineering (2019 Course)
210257: Microprocessor Laboratory

Home

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Practical: 02 Hours/Week	01	**Term Work:** 25 Marks **Oral:** 25 Marks

Companion Course : 210254: Microprocessor

Course Objectives:

- To understand assembly language programming instruction set
- To understand different assembler directives with example
- To apply instruction set for implementing X86/64 bit assembly language programs

Course Outcomes:

On completion of the course, learner will be able to–

CO1. **Understand** and **apply** various addressing modes and instruction set to implement assembly language programs

CO2. **Apply** logic to **implement** code conversion

CO3. **Analyze** and **apply** logic to **demonstrate** processor mode of operation

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work is based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Use of open source software is encouraged.

In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System: 64-bit Open source Linux or its derivative.

Programming Tools: Preferably using Linux equivalent or MASM/TASM/NASM/FASM.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Virtual Laboratory:

- http://209.211.220.205/vlabiitece/mi/MI3.php

Suggested List of Laboratory Experiments/Assignments(any 10)

Sr. No.	Assignments

Home

1	Write an X86/64 ALP to accept five 64 bit Hexadecimal numbers from user and store them in an array and display the accepted numbers.
2	Write an X86/64 ALP to accept a string and to display its length.
3	Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.
4	Write a switch case driven X86/64 ALP to perform 64-bit hexadecimal arithmetic operations (+,-,*, /) using suitable macros. Define procedure for each operation.
5	Write an X86/64 ALP to count number of positive and negative numbers from the array.
6	Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. (Wherever necessary, use 64-bit registers).
7	Write X86/64 ALP to detect protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers also identify CPU type using CPUID instruction.
8	Write X86/64 ALP to perform non-overlapped block transfer without string specific instructions. Block containing data can be defined in the data segment.
9	Write X86/64 ALP to perform overlapped block transfer with string specific instructions Block containing data can be defined in the data segment.
10	Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. (use of 64-bit registers is expected).
11	Write X86 Assembly Language Program (ALP) to implement following OS commands i) COPY, ii) TYPE Using file operations. User is supposed to provide command line arguments
12	Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file. The text file has to be accessed during Program_1 execution and write FAR PROCEDURES in Program_2 for the rest of the processing. Use of PUBLIC and EXTERN directives is mandatory.
13	Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.
14	Write an X86/64 ALP password program that operates as follows: a. Do not display what is actually typed instead display asterisk ("*"). If the password is correct display, "access is granted" else display "Access not Granted"
15	Study Assignment: Motherboards are complex. Break them down, component by component, and Understand how they work. Choosing a motherboard is a hugely important part of building a PC. Study- Block diagram, Processor Socket, Expansion Slots, SATA, RAM, Form Factor, BIOS, Internal Connectors, External Ports, Peripherals and Data Transfer, Display, Audio, Networking, Overclocking, and Cooling. 4. https://www.intel.in/content/www/in/en/support/articles/000006014/boards-and-kits/desktop-boards.html

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	1	2	1	-	-	-	-	-	-	-	-
CO2	2	1	-	1	-	-	-	-	-	-	-	-
CO3	-	1	-	1	-	-	-	-	-	-	-	-

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

UNIVERSITY QUESTION PAPERS & MODEL SOLUTION

Total No. of Questions : 8]

SEAT No. :

P650

[Total No. of Pages : 2

[5869]-279
S.E. (Computer)
MICROPROCESSOR
(2019 Pattern) (Semester - IV) (210254)

Time : 2½ Hours] *[Max. Marks : 70*

Instructions to the candidates:

1) *Answer Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.*
2) *Neat diagrams must be drawn whenever necessary.*
3) *Figures to the right side indicate full marks.*
4) *Assume suitable data if necessary.*

Q1) a) With the help of a neat diagram, explain the Page Translation Process in 80386. **[6]**

b) Draw and explain General Selector Format. **[6]**

c) What is a Logical address, Linear address and Physical address? **[6]**

OR

Q2) a) Explain the use of following instructions in detail : **[6]**

i) SGDT ii) LIDT iii) SLDT

b) Explain the Segment Translation Process with a neat diagram of 80386. **[6]**

c) Enlist various types of system and non-system descriptors in the 80386. Explain their use in brief. **[6]**

Q3) a) Write a short note on CPL, DPL, and RPL. **[6]**

b) Explore the role of various fields in Page Level Protection. **[6]**

c) List and explain various Privilege Instructions. **[5]**

OR

P.T.O.

Q4) a) What is call gate? Explain how it is used in calling functions with higher privilege levels. **[6]**

b) Define the functions of Type Checking and Limit Checking in protection. **[6]**

c) Explain different levels of protection? State the rules of protection check. **[5]**

Q5) a) Explore the role of Task Register in multitasking and the instructions used to modify and read Task Register. **[6]**

b) Draw and Explain the Task State Segment of 80386. **[6]**

c) Difference between Real Mode and Virtual 8086 Mode. **[6]**

OR

Q6) a) Explain the TSS descriptor of 80386 with a neat diagram. **[6]**

b) Explore memory management in the Virtual 8086 Mode. **[6]**

c) List and explain various features of virtual 8086 Mode. **[6]**

Q7) a) Explain the process of Enabling and Disabling Interrupts in 80386. **[6]**

b) Differentiate and Explain the Interrupt gate and Trap gate descriptor. **[6]**

c) Differentiate between Microprocessor and Microcontroller. **[5]**

OR

Q8) a) With the help of the necessary diagram, explain the structure of IDT in 80386. **[6]**

b) Explain different types of exceptions in 80386 with suitable examples. **[6]**

c) Draw and Explain the Architecture of a Typical Microcontroller. **[5]**

❄ ❄ ❄

Total No. of Questions : 8]

SEAT No. :

PA-1240

[Total No. of Pages : 2

[5925]-263

S.E. (Computer)

MICROPROCESSOR

(2019 Pattern) (Semester - IV) (210254)

Time : 2½ Hours] *[Max. Marks : 70*

Instructions to the candidates:

1) ***Solve Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.***
2) ***Neat diagrams should be drawn wherever necessary.***
3) ***Use of Non-programmable Calculator is allowed.***
4) ***Assume suitable data if necessary.***

Q1) a) Explain the Segment Translation Process with a neat diagram of 80386. **[6]**

b) Differentiate and explain GDTR, LDTR, and IDTR. **[6]**

c) Demonstrate General Selector Format in brief. **[6]**

OR

Q2) a) Demonstrate General Descriptor Format available in various descriptor tables. **[6]**

b) With the necessary diagram, explain the page translation process in 80386. **[6]**

c) Explain the use of following instructions in detail: **[6]**

i) LGDT

ii) SIDT

iii) LLDT

Q3) a) What is call gate? Explain how it is used in calling functions with higher privilege levels. **[6]**

b) Explore five aspects of protection applied in segmentation. **[6]**

c) Explore the need for a protection mechanism in 80386. **[5]**

OR

P.T.O.

Q4) a) Explain the following terminologies. **[6]**

i) CPL

ii) DPL

iii) RPL

b) Explain different levels of protection. Describe the rules of protection check? **[6]**

c) Elaborate on the concept of combining segment protection and page level protection in 80386. **[5]**

Q5) a) Explore memory management in the Virtual 8086 Mode. **[6]**

b) Explain the TSS descriptor of 80386 with a neat diagram. **[6]**

c) Explore the role of Task Register in multitasking and the instructions used to modify and read Task Register. **[6]**

OR

Q6) a) Draw and explain the Task State Segment of 80386. **[6]**

b) With the necessary diagram, explain entering and leaving the virtual mode of 80386. **[6]**

c) Difference between Real Mode and Virtual 8086 Mode. **[6]**

Q7) a) Explain the following exception conditions with an example: Faults, Traps, and Aborts. **[6]**

b) With the help of the necessary diagram, explain the structure of IDT in 80386. **[6]**

c) List and elaborate on different applications of microcontrollers. **[5]**

OR

Q8) a) Differentiate and explain the Interrupt gate and Trap gate descriptor. **[6]**

b) How interrupts are handled in protection mode. Explain with the help of a neat diagram. **[6]**

c) Differentiate between Microprocessor and Microcontroller. **[5]**

X X X

Q.1

a) With the help of a neat diagram explain the page translation process in 80386

→

Conversion of linear address to physical address this is known as paging mechanism.

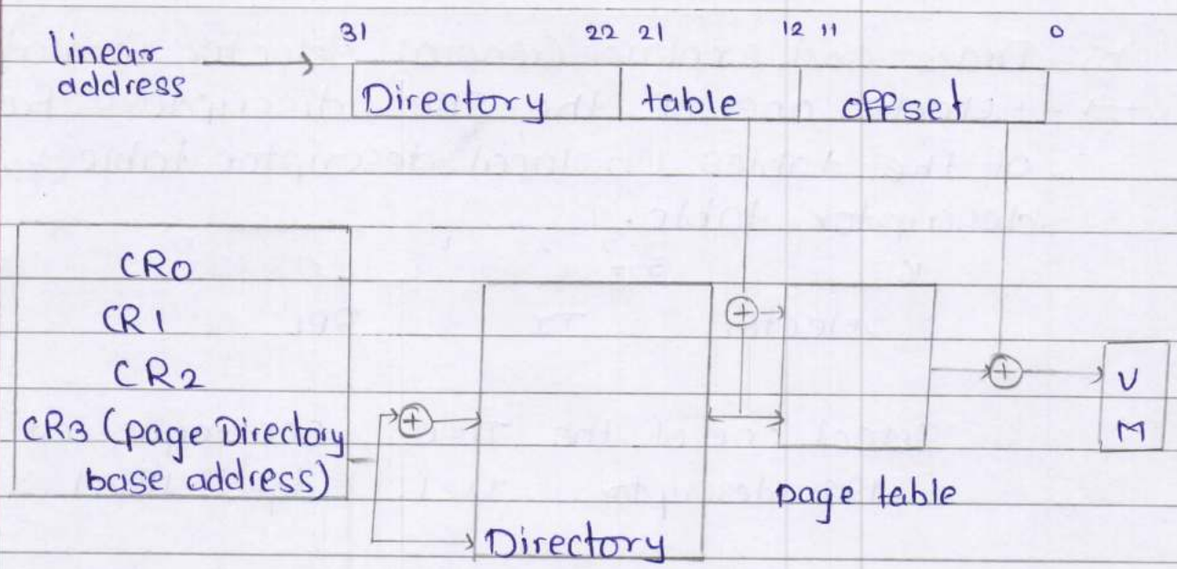


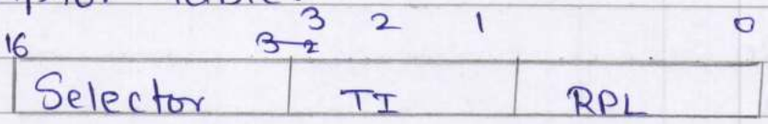
Fig. paging mechanism

- The MMU Consider the memory divided into pages of 4kB each. Hence the 4GB memory is divided into 1M page of 4kB each. To select one of these 1M pages 20 bit are required.
- CR3 consist of 20 bit page directory base address that selects the page called page directory.
- This page is also of 4kB that has 32 bit entry and hence has 1k. entries called as page directory entries
- To select one of these 1k entries 10 bit required hence the first 10 bits of the linear address ie. bit 31 to bit 32. Select directory has 20 bit that selects another page directory has 20 bit that selects another page directory called as page table.
- Since there are 1k entries in the page directory each of them selecting a page table. there can be 1k page tables.
- The page table size is also 4kB that has 32 bit.

entry and hence has 1k entries. To select one of the 1k entries 10 bits are required hence the next 10 bits of the linear address ie bit 21 to bit 12. Select one of the 1k entries called as page table entries.

b) Draw and explain General Selector format

→ - Selects one of the 8192 discriptors from one of the tables U12 local descriptor table & local descriptor table.



Select one of the 8192 descriptor
TI=0 GDC
TI=1 LDT
requested privileged level

Fig. Structure of the Selector

- GDTR points to Segment common for all programs while LDTR (Local Descriptor table register) point to Segment containing programs that are unique to an application
- Each Segment Consist of 8192 descriptors 8 bytes each and hence a total of 16k Segment in all can be addressed by any task.
- Hence to select a descriptor out of 8k descriptor Selector uses 13 MSB.
- One bit to Select the discrptor table called as table identify bit. Remaining two bit are used to pass on the Requested priviledged (RPL)

- when loading segment Selector the 80386 checks that -
- The Selector index is with in the descriptor table limit
- The Selector reference the Correct descriptor table

- The descriptor is of the correct type
- The Selector uses the correct priviledge level.

c) what is the logical address linear address and physical address

→ The logical address is converted to linear address with the help of segmentation mechanism while the linear address is converted to physical address with the help of paging mechanism.

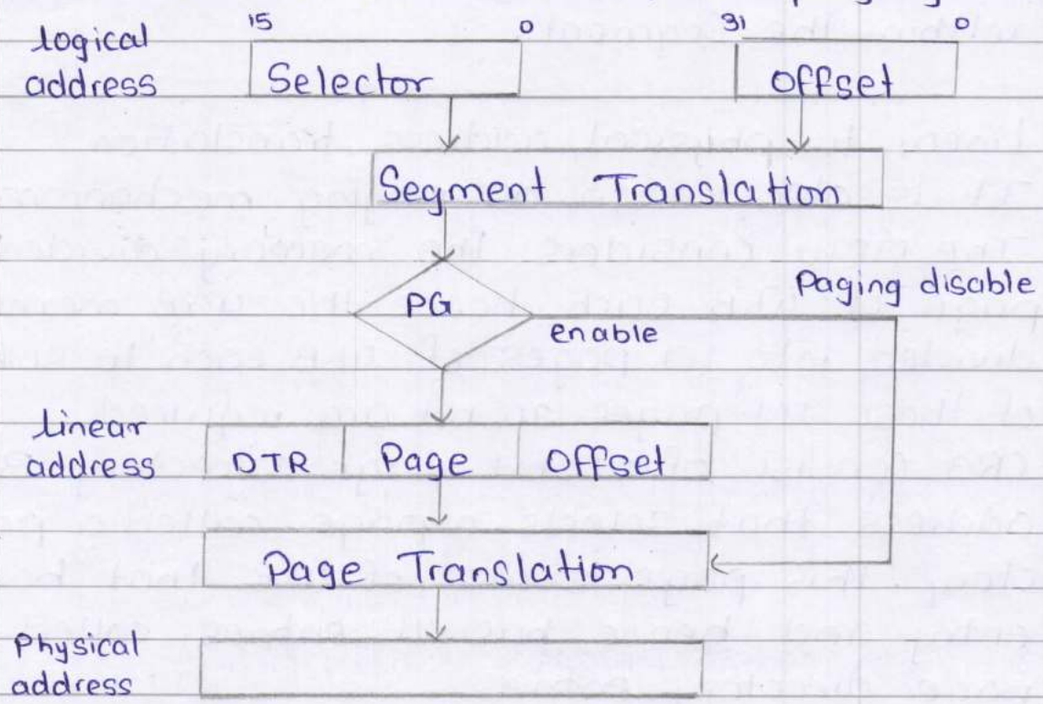


fig. Virtual or physical address translation

- Selector is used to select the one segment from system.
- offset is used to store the address of segment while another segment is in process or in working made
- PG is page enable or disable when PG=1 then page is enabled otherwise it is disable (PG=0)

1) Logical to linear address translation
- It is also known as segmentation mechanism
- Segmentation is the process in the which the main memory of a computer is logically divided into different segments each segment has it's own base

address.
- The Segment size in the protection mode for x86 can be anything between 1 byte to 4GB.
- The logical address in protected mode architecture is formed out of two components.

a) A 16-bit Selector that is used to determine the descriptor
b) 32-bit offset to give the internal address within the Segment.

2) Linear to physical address translation -
- It is also known as paging mechanism
- The MMU considers the memory divided into page of 4kB each. hence the 4GB memory is divided into 1M pages of 4kB each to select one of these 1M pages 20 bit are required.
- CR3 consist of 20 bit page Directory Base address that selects a page. called a page directory. this page is also of 4kB that has 32 bit entry and hence has 1k entries called as page directory entries.
- The entry in page directory has 20 bit that selects another page called as page table.

Q.

Q.2

a) Explain the use of following instruction in detail
1) SGDT 2) LIDT 3) SLDT

→

① SGDT -

- SGDT ISIDT copies the contents of the descriptor table register the six byte of memory indicated by the operand. the LIMLY field of the register is assigned to the first coord at the effective address. If the operand size attribute is 32 bits. the next three bytes are assigned the BASE field of the register and fourth byte is written with zero.
- SGDT and SIDT are used only in OS software they are not used in application program.
- Instruction - Instruction SGDT/SIDT Store global Interrupt descriptor table register.
- Operation - Descriptor store GDTR to m or store IDTR to m.

② LIDT -

- Load value in Global descriptor table
- Instruction - LGDT Source
- Operation - This instruction loads the value in the source to the Global descriptor table register
- This instruction is commonly eseecuted in real-address mode to allow processor initialization prior to switching to protected mode.
Example - LGDT ARRL

③ SLDT -

- SLDT Store the local descriptor table register in the two-byte register or memory location indicated by the effective address operand.
- This register is a selector that point into the global descriptor table. SLDT is used only in operating System. Software It is not used in

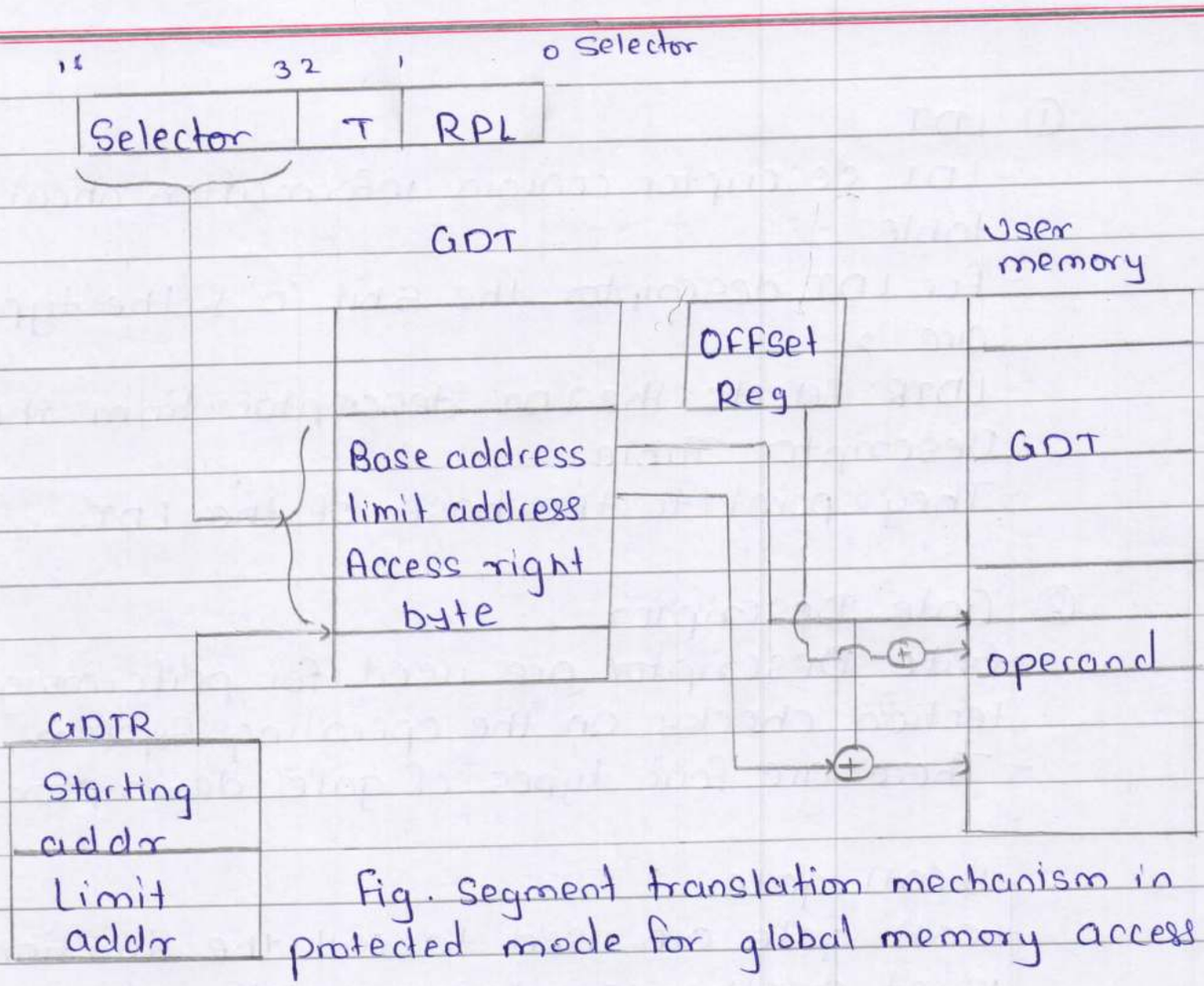
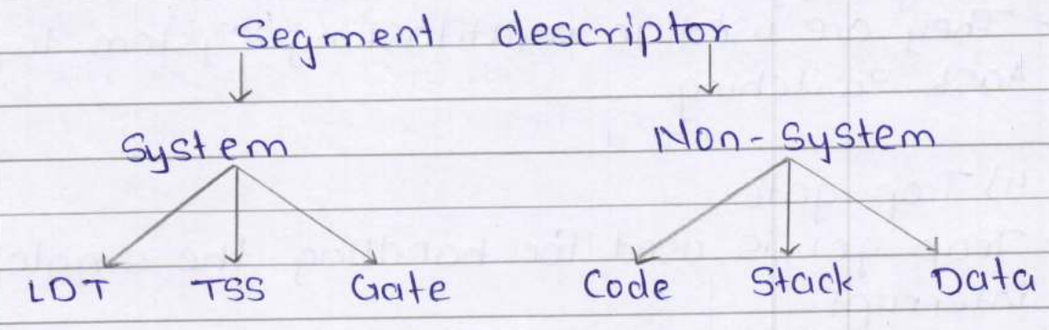


Fig. Segment translation mechanism in protected mode for global memory access

c) Enlist various types of System and non-system descriptor in 80386 Explain their use in brief.

→ There are two main categories of segment descriptors 1) System Segment Descriptor
2) Non-System Segment Descriptor



1) System Segment descriptor
- It provide the information of OS tables tasks and gates
- There are three types of Segment descriptor

2) Non-System Segment descriptor

① code

- used for addressing memory location in the code segment of the memory where the executable program is stored.

② Data

- point to the data segment of the memory where the data is stored.

③ Stack

- used for ~~me~~addressing stack segment of the memory
- Stack segment is that segment of memory which is used to store stack data.

Q.3

a) write short note on CPL, DPL and RPL

→ - most processors have only two protection levels but x86 architecture features four level of protection called priviledge level.

- They are designed to support the need of multitasking as to isolate and protect user programs from each other and the OS from unauthorized access.
- The priviledge level control the use of privileged instruction I|O instruction and access to segments and segment descriptor the x86 offers an additional type of protection on a page basic, when paging is enable.

Requested PL - RPL
Descriptor PL - DPL
Current PL - CPL
Effective PL - EPL

- Following PL's are used to maintain priviledged level Check:

Q.4

a) what is call gate ? Explain how it is used in calling function with higher previledged levels.

→ - To provide protection for control transfers among executable segments at different priviledged level the 80386 uses gates descriptors

- There are four kind of gate descriptor
  1) call gate
  2) Trap gate
  3) Interrupt gate
  4) Task gate

- A call gate descriptor may reside in the GDT or in an LDT.

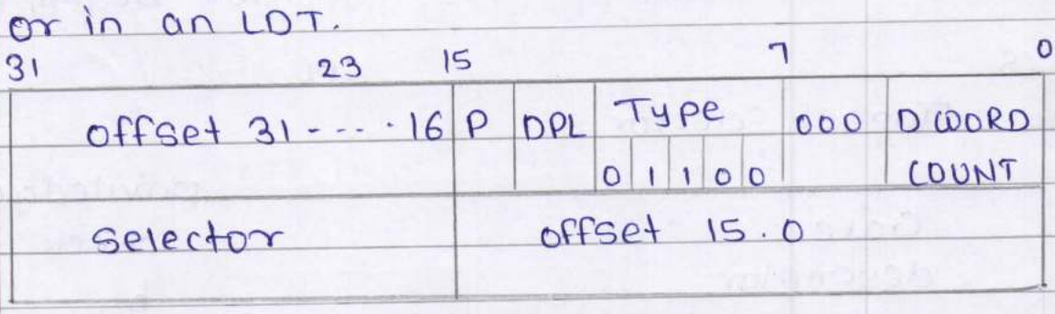


Fig. Format of 80386 call gate

- A call gate has two primary function

1) To define an entry point of a procedure
2) To Specify the priviledged level of the entry point

- call gate descriptor are used by call & jump instruction in the same manner as code segment descriptors.
- The Selctors and offset field of a gate from a pointer to the entry point of a procedure.
- A call gate gaurantee that all transition to another segment got to valid entry point rather than possibly into the middle for a procedure
- The far pointer operand does not point to the segment and offset of the target instruction rat-

② It Specifies the intended usage of a Segment.

- The type check basically checks the type of the descriptor with the selector use to access it.
- A descriptor may be executable it non executable and hence accordingly the code Seg. register or the data Segment register must be used for accessing.
- Also a descriptor could be System descriptor
- the redable bit in an executable segment descriptor specifies whether instruction are allowed to read from the Segment.

2) Limit Checking

- A limit Field as the name indicates gives the boundary of the Segment when added with the base address.
- The limit as compared with the offset to decide whether the access is beyound the limit of the Segment. This could otherwise cause the issue of the one task changing the Contents or accessing the data of another task.
- for all type of Segments except expand down data Segment the value of the limit is one less than the Size of the Segment.

- The processor causes a general protection excepta-ion in any of these cases.
- Attempt to access a memory byte of as address > limit.
- Attempt to access a memory word at an address ≥ a limit
- Attempt to access a memory double word at an address ≥ (limit 2)

Q. 5

a) Explain the role of task register in multitasking and the instruction used to modify and read task register.

→ - The task register identifies the currently executing task by pointing to the TSS.
- The task has both visible portion and an invisible portion
- The Selector in the visible portion selects TSS Segment in the GDT.
- The processor user the invisible portion to cache the base and limit value from the TSS Segment.

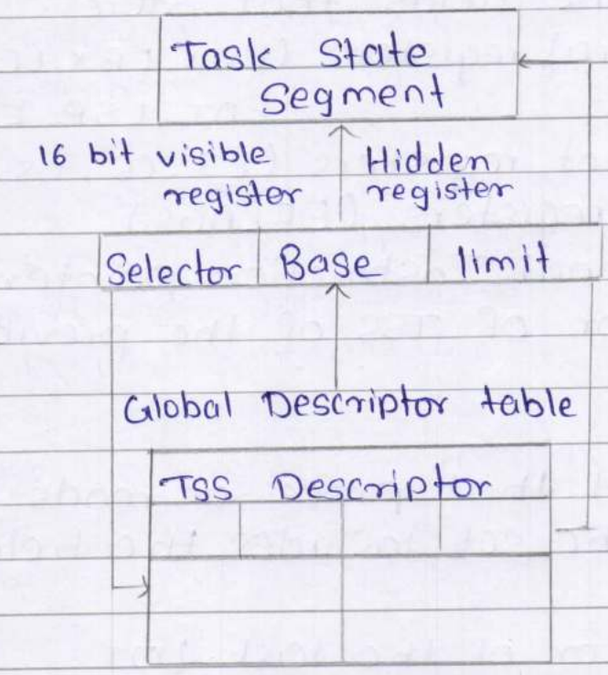


Fig. Task register.

- LTR loads the visible portion of the task register with the Selector operand. which must select a TSS descriptor in the GDT. ITR also loads the invisible portions with information from the TSS descriptor selected by the Operand.
- LTR is a priviledged instruction it may be executed only when CPL is zero. LTR is general used during System initialization to give an initial value to

c) Difference between Real mode and virtual 8086 mode

→

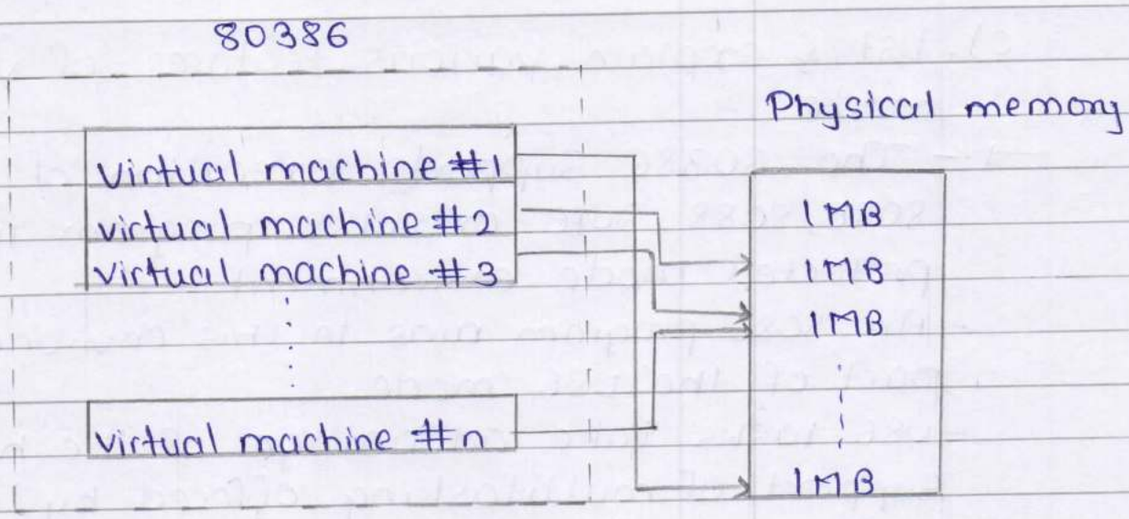
Parameter	Real mode	Virtual 8086 mode
General	It is default mode of the processor when it is switch ON. It is similar to 8086 & hence is called as real mode or real addressing mode.	Switching betⁿ real & protected mode is quite complicated as it required a restart. virtual mode allows 8086 task to be executed without restarting the processor
Use	It is used to execute the 8086 task	It is used to execute 8086 task without restarting processor.
memory addressing	In real mode memory upto 1MB to 64KB - 16 bytes can be accessed	In real mode memory upto 1MB to 64KB - 16 bytes can be accessed
Register access	All general purpose as well as memory management registers can be accessed.	Only general purposed register is accessible in virtual mode
Entering Mode	On Restart or RESET 80386 enters in the mode	Enabling of VM bits in the EFLags registers make the processor enter into virtual mode.
leaving Mode	when the PE bit is set 80386 enters protected mode & exists from the real mode.	Existing from the virtual mode is via an interrupt ~~mode~~ or exception.

b) Explore memory management in virtual mode

→

- The processor enters U86 mode to execute the 8086 program and returns to protected mode to execute the monitor or other 80386 tasks.
- To run Successfully in U86 mode on existing 8086 program needs the following -
  1) A U86 monitor
  2) Operating - System Services.

- The U86 monitor is 80386 protected mode code the executes at priviledge level zero.
- The monitor consists primarity of initialization and exception - handling procedures.



Each virtual machine is a seprate 8086 machine

paging mechanism allows 1MB spaces to be any where in 1GB physical memory.

- In general there are two options for implementing the 8086 operating System
  1) The 8086 operating System may run as part of 8086 code this approach is disirable for any of the following reason

of co-operation betn hardware and hardware.
- In the case of I/O slw can choose either to emulate I/O instructions. on to let the hardware execute them directly without software intervention.

b) Differentiate and explain the interrupt gate and trap gate descriptor

→

	Interrupt gate	Trap gate
1)	Interrupt are externally generated event	faults are Internally generate events
2)	It is a mechanism by which an external device can suspend the normal execution of the processor and get itself serviced.	A trap is an exception that is reported at the instruction boundary immediately after the instruction in which the exception was detected
3)	After servicing the interrupt the 8086 will execute the next sequential instruction before which the interrupt occured.	The instruction that caused the trap cannot be restarted
4)	Example - INT 32-255	Example - INT 2, INT 3, INT 4

Q.8

a) with the help of the necessary diagram explain the structure of IDT in 8086.

→ The interrupt descriptor table associates each interrupt or exception identifier with a descriptor for the instruction that service the associated event.

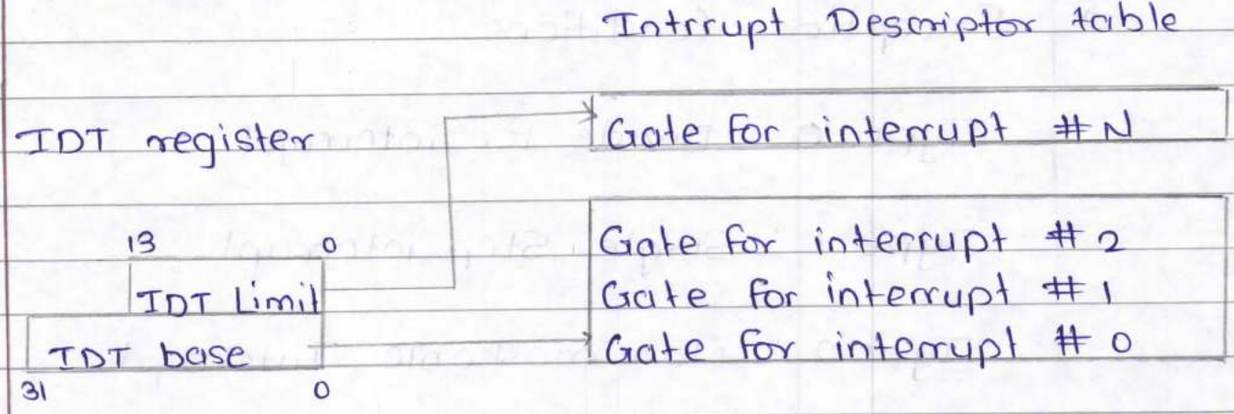


Fig. IDT register and table

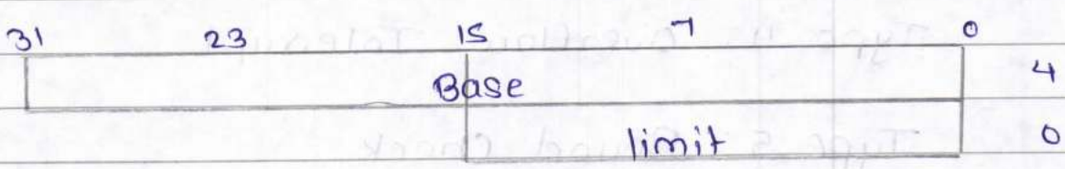


Fig. pseudo descriptor format for LIDT & SIDT

- like the GDT & LDT the IDT is an array of 8 byte descriptors. unlike the GDT and LDT's the first entry of the IDT may contain a descriptor.
- To form an index into the IDT, the processor multiplies the interrupt and exception identifier by eight. Because there are only 256 identifier the IDT need not contain more than 256 descriptors. It can contain fewer than 256 entries. entries are required only for interrupt identifiers that are actually used.
- The IDT may reside anywhere in physical memory
- LIDT loads the IDT register with linear base

Type 14: page fault

Type 15: Reserved

Type 16: Floating point error

Type 17: Alignment check

Type 18: Machine Check

c) Draw and explain the architecture of typical microcontroller

→

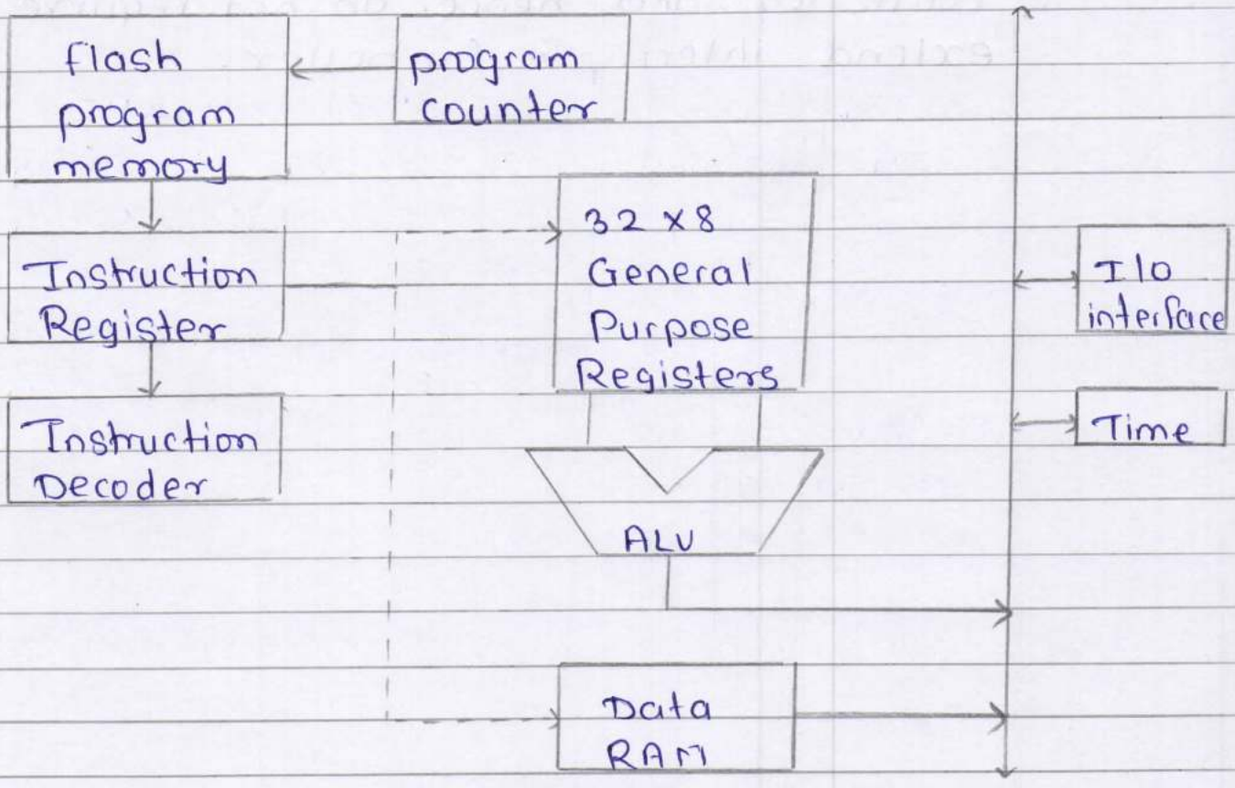


fig. microcontroller

-microcontroller ~~archi~~ architecture can be based on the hardware architecture van-Neuman architecture both offering different method of exchanging data bet^n the processor and memory

Question Paper

Microprocessor

Q1.

a) Explain the Segment translation process With neat diagram of 80386.

→

- Global memory location is One which is accessible to all the tasks. The Global descriptor table (GDT) is a Common table accessible to all the task
- The GDT is Selected by a 48-bit register Called as GDT register.

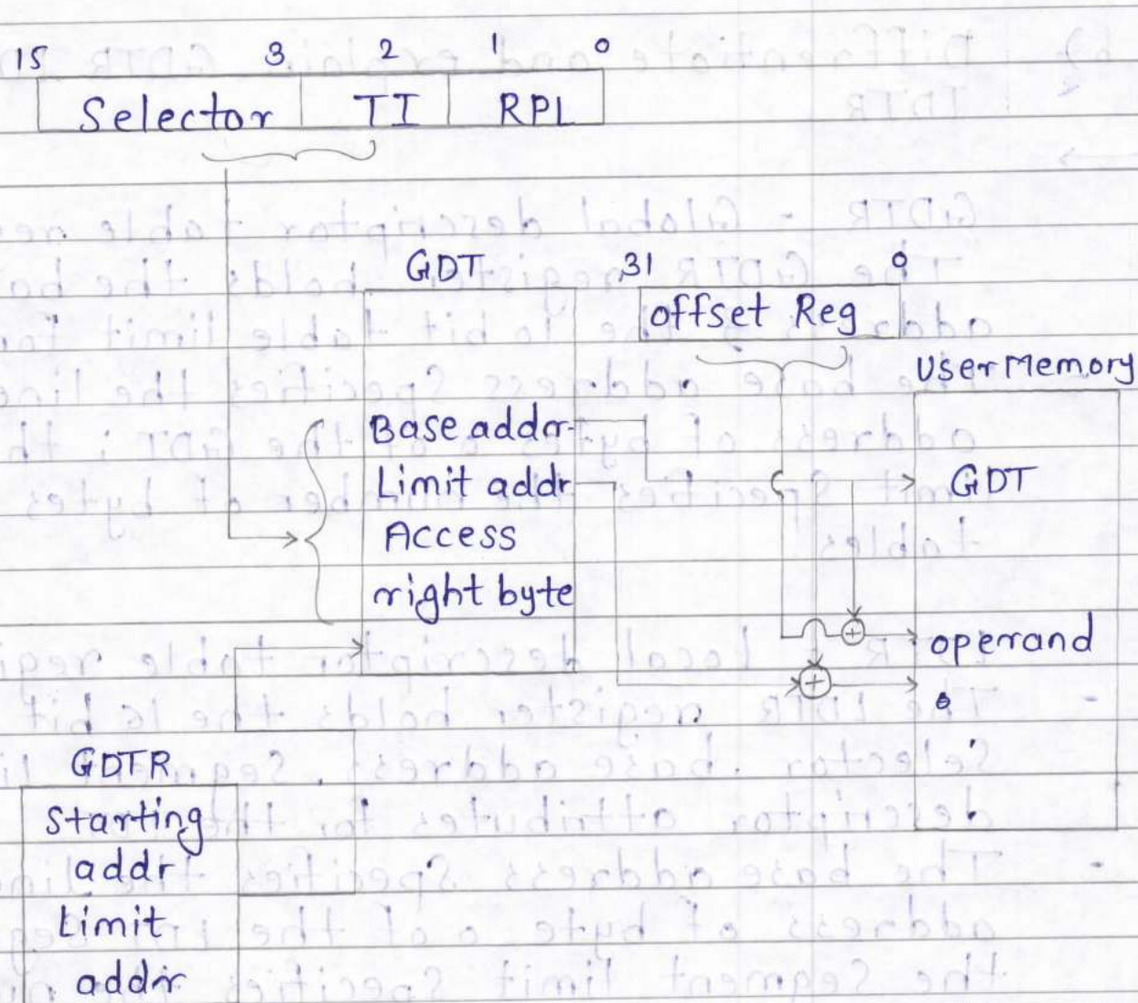


Fig. Segment translation mechanism in protected mode for global memory access.

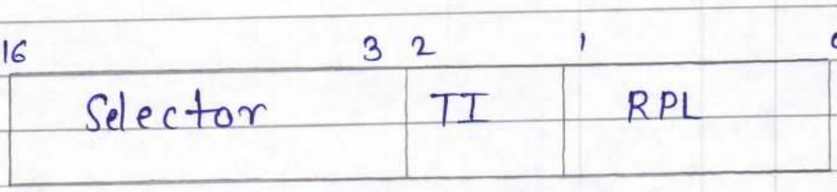
- GDT has foll⁹ two Fields:

and 16 bit table limit for the IDT.
- The base address Specifies the linear address of bytes 0 of the IDT, the table limit Specifies the number of bytes in the table.
- The LIDT & SIDT instruction load & store the IDTR register, respectively.

c) Demonstrate general Selector format in breif.

→

- Selects One of the 18192 descriptor from One of the tables viz local descriptor table and local descriptor table.



Select one of the 8192 descriptor

TI = 0 GDT requested
TI = 1 LDT priviledge level

Fig. Structure of the Selector.

- GDTR (Global Descriptor Table Register) points to Segment Common for all programs. While LDTR (Local Common Descriptor Table Register). points to Segment Containing programs that are unique to an application.
- Each Segment Consist of 8192 descriptor of 8 bytes each & hence a total of 16 K Segments in all Can be addressed by any task.
- Hence to Select a descriptor Out of 8K descriptor Selector uses 13 MSB.
- When loading Segment Selector the 80386 checks that-

Q 2.
a) Demonstrate General descriptor format available in various descriptor tables.

→

7	Base (B31-B24)	G	D	0	AV	Limit (L19-L16)	8
5	Access Right Byte	Base address (B23-B16)					4
3	Base address (B-15-Bo)						2
1	Limit (L15 -Lo)						0

Fig. Descriptor register format

- Limit (Lo -Lq) When added with the base address gives the last address of the Segment In Case of 286 there is a 24 bit base address and 16-bit limit while in 386 We have 32 bits base address & 20 bit limit address
- Granularity bit (G)
- When G = 0 , 20 bit limit Specifies the Segment Size from 1 byte to 1 MB.
- When G = 1 , 20 bit limit is to be multiplied by 4k hence Segment Size Varies from 4KB to 4GB.
- When AV = 0, it indicates the Corresponding Segment is used by Some other task & hence it is not available.
- When AV = 1, it indicates the Corresponding Segment is used by any other task & hence is available.
- D indicates data Size.
- When D = 0, it indicate 16-bit os instructions
- When D = 1, it indicates 32-bit os instructions

Access Right Byte (ARB)

7	6	5	4	3	2	1	0
P	DPL	S	E	ED	C	RW	A

Fig. ARB Structure

- Hence the next 10 bits of the linear address i.e bit 21 to bit 12, Select One of the 1K entries Called as page table entries.

c) Explain the use of following instructions in details.
① LGDT ② SIDT ③ LLDT

→

① LGDT - Load Global descriptor Table.
- The LGDT instruction loads a linear base address & limit Value from a Six-bytes data operand in memory into the GDTR or LDTR, respectively.
- LGDT appear in Operating System Software they are not used in application programs. They are the only instructions that directly load a linear address in 80386 protected mode
- flag affected - None.

② SIDT - Store Interrupt Descriptor Table
- SIDT Copies the Contents of the descriptor table register the six bytes of memory indicated by the operand.
- The LIMIT Field of the register is assigned the first Word at the effective address.
- If the operand Size attribute is 32 bits, the next three bytes are assigned the BASE field of the register & the fourth bytes is Written With Zero.
- flag affected - None.

③ LLDT - Load Local Descriptor table;-
- LLDT loads the Local Descriptor table register.
- The Word operand to LLDT should Contain a Selector the global descriptor table.

Q3.
a) What is Call gate ? Explain how it is used in Calling function With higher priveledged level.

→

To provide a protection for Control transfers among executable Segments at different priviledge levels, the 80386 uses gate descriptors.

- There are four kinds of gate descriptor.
  1. Call gates
  2. Trap gates
  3. Interrupt gates
  4. Task gate.
- A Call gate descriptor may reside in the GDT or in the LDT.

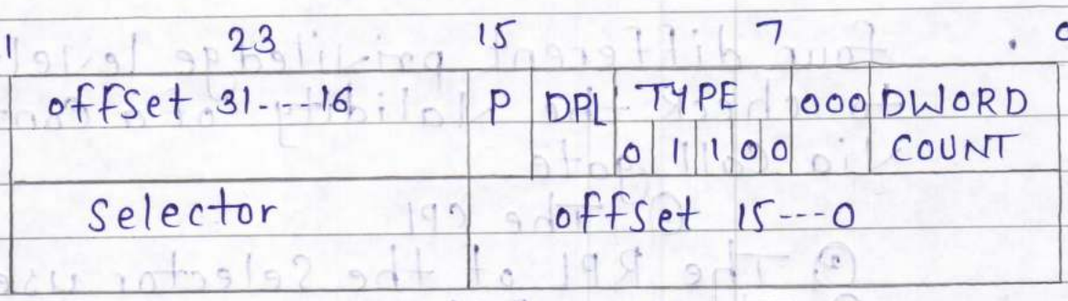


Fig. format of 80386 Call gate

- A Call gate has two primary functions.
  1. To define an entry point of a procedure.
  2. To Specify the previledged level of the entry point.
- Call gate descriptor are used by Call & jump instruction in the same manner as Code Segment descriptors.

- All five methods of protection check are application to Segment translation etc.
  1. Type checking
  2. Limit checking
  3. Restriction of addressable domain
  4. Restriction of procedure entry points.
  5. Restriction of instruction Set.

- The Segment is the unit of protection and Segment descriptors Store protection parameters.
- protection checks are performed automatically by the CPU when the selector of a Segment descriptor is loaded into a Segment register. and With every Segment access.

① Type checking :-

- The type feild of a descriptor has two functions.

① It distinguishes among different descriptor formats.
② It Specifies the intended usage of a Segment

- The type check basically checks the type of the descriptor With the Selector used to access it.

② Limit checking -

- The limit field as the name indicates gives the boundary of the Segment When added With the base address.
- The direction of data Segments used as stack Will expand in downwards direction.

Q.4

a) Explain the following terminologies.
① CPL ② DPL ③ RPL.

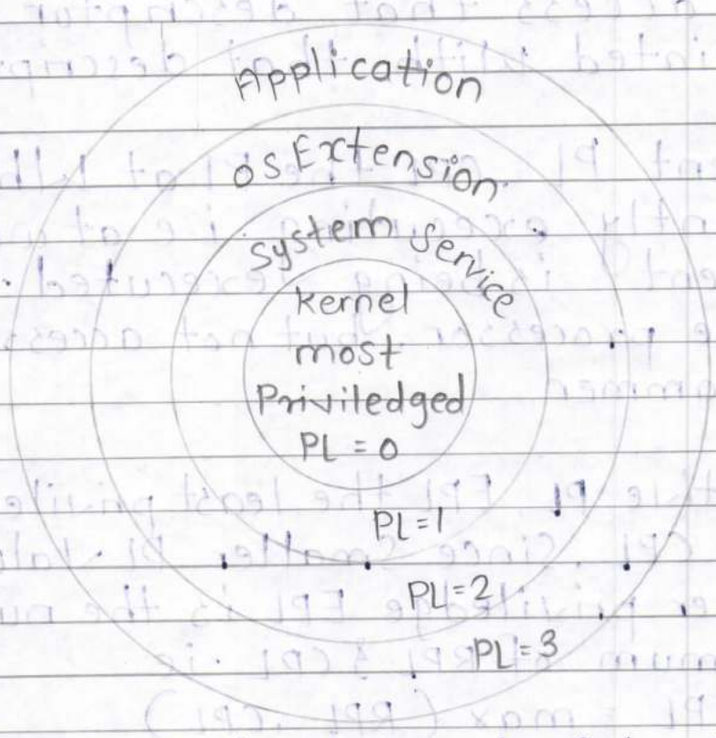


Fig. priviledged level of task in x86 architecture.

- Most processors have only two protection levels but x86 architecture features four levels of protection called priviledge level
- The x86 architecture offers on additional type of protection on a page basis when paging is enabled.

Requested PL ⟶ RPL
Descriptor PL ⟶ DPL
Current PL ⟶ CPL
Effective PL ⟶ EPL

* foll⁹ PL is are used to maintain priviledged level check;

a. Requestor PL, RPL the PL of the orignal task

protection in the 80386 have five aspects-

1. Limit checking
2. Type checking.
3. Restriction of addressable domain.
4. Restriction of procedure entry point.
5. Restriction of instruction set.

- The restriction of address domain, procedure entry point & instruction set is also called as priviledged check.

- The Selector in the visible portion Selectors TSS descriptor in the GDT.
- The processor uses the invisible portion to cache the base & limit values from the TSS descriptor.
- LTR loads the visible portion of the task register with the Selector operand, which must Select a TSS descriptor in the GDT. LTR also loads the invisible portion with information from the descriptor Selected by the operand.
- LTR is a priviledged instruction it may be executed only when CPL is zero. LTR is generally used during System initialization to give an initial value to the task register, therefore the Contents of TR are changed by task switch operations.

entering & leaving the Virtual mode of 80386.

→

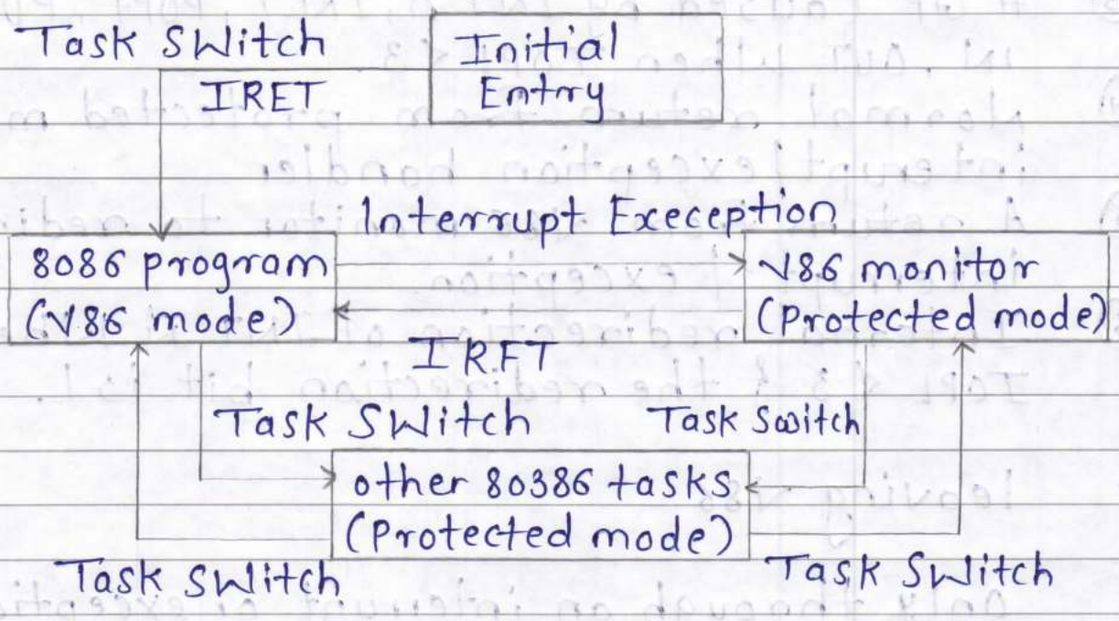


Fig. Entering & leaving Virtual 8086 mode.

* Entering V86 mode.
  - CPU runs in V86 if EFlags [VM] = 1
  - How to Set it
    - Task SWitch
      - Reading EFlags iron TSS before Segment register loading.
    - Return from interrupt handler IRET
      - Reading EFIAGS from the stack.
  - VM Flags checks.
    - Segment register loading.
      - How to set Segment register Cache.
    - Instruction decoding.
    - Instructions not Supported by V86
    - Instruction Sensitive on IOPL
    - Access right
      - V86 always run With CPL = 3.

* Entering and leaving VM86.

① Task SWich

		i.e Virtually going into real addressing mode.
Memory Addressing	In real mode memory upto 1MB + 64KB - 16 bytes Can be accessed.	In real mode memory upto 1MB + 64KB - 16 bytes Can be accessed.
Entering the mode	On restart of RESET 80386 enters in the mode.	Enabling of VM bits in the EFlags register makes the processor enter into Virtual mode.
Loading the mode	When the PE bit is Set 80386 enters protected mode & exists from the real mode	Existing from the Virtual mode is via an interrupt or exception.

- Contain a descriptor.
- The IDT may reside anywhere in physical memory.

00000H. Each 4 byte entry in the IVT consists of a CS : IP pair that specifies the address of the first instn in the interrupt service routine (ISR).
- An 8 bit vector number is shifteted two bits to the left to form an index into the IVT.

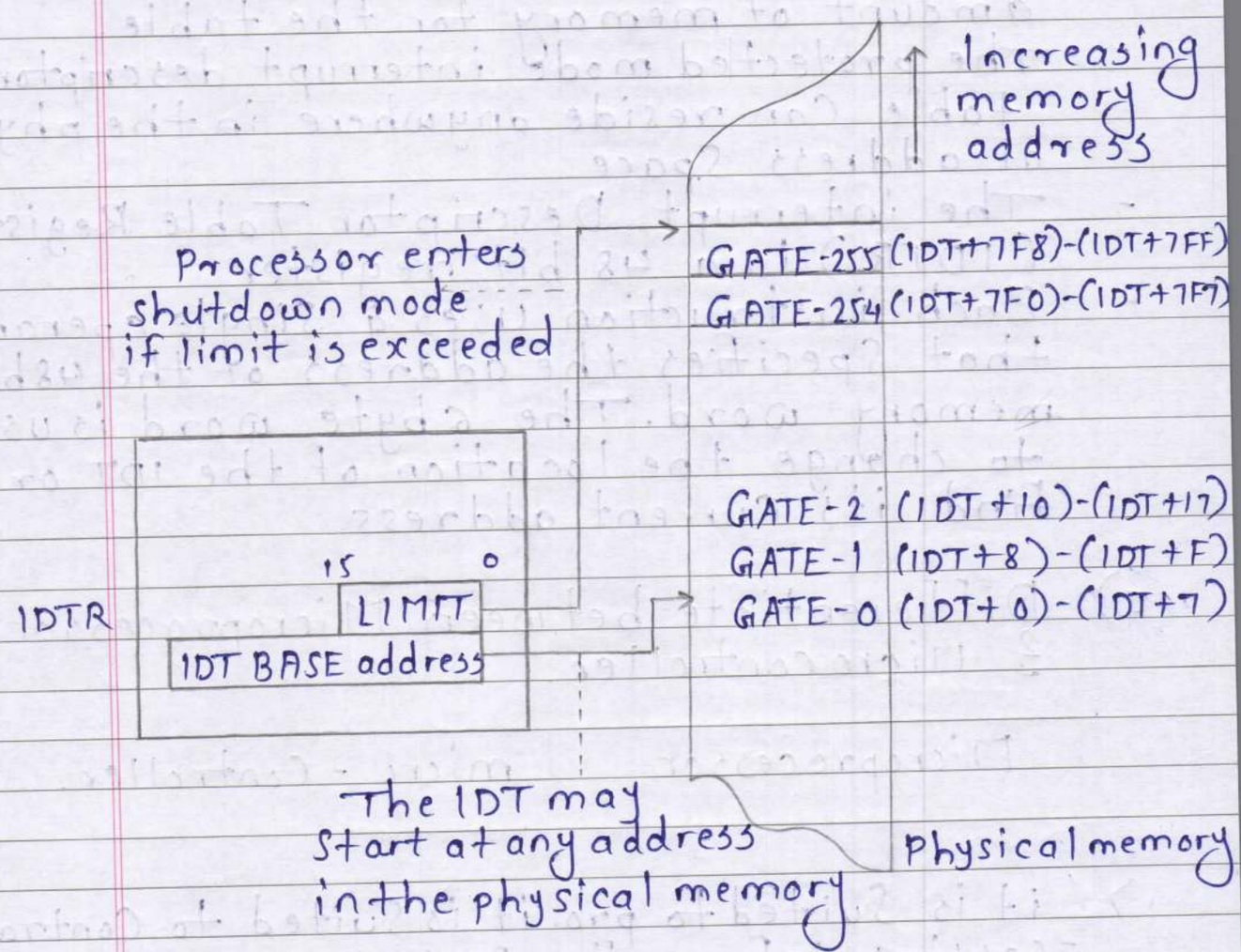


Fig. The procdected mode interrupt descriptor table.

- The IDT consprises 8 bytes gate descriptors for track trap or interrupt gates

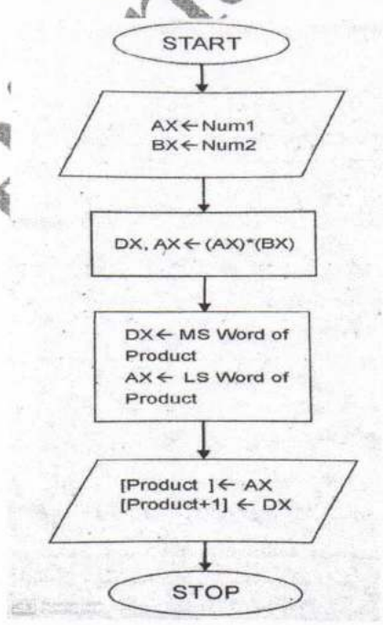
	to interface I/O device.	timing Counting.
3>	it have very wide has widths. It has large memory address & lots of data.	it has narrow buses it has relatively small memory address space & less data.
4>	microprocessor are very fast.	it are relatively slow since most I/O devices being Controlled are relatively slow.
5>	microprocessor are expensive.	microcontroller are cheap.

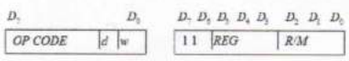
Subject Name: MICROPROCESSOR **Model Answer**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

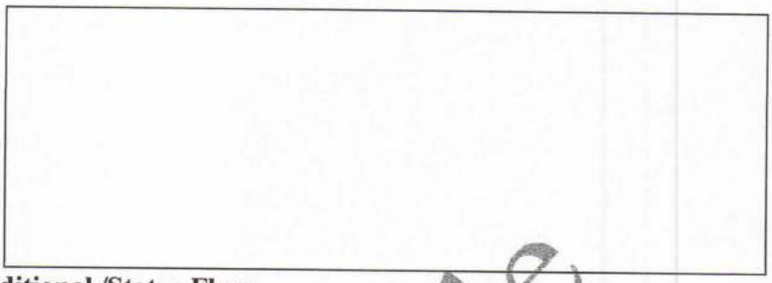
Q. No.	Sub Q. N.	Answer	Marking Scheme
1.		**Attempt any Five of the following:**	**10M**
	a	**State the function of READY and INTR pin of 8086**	**2M**
	Ans	**Ready**: It is used as acknowledgement from slower I/O device or memory. It is Active high signal, when high; it indicates that the peripheral device is ready to transfer data. **INTR** This is a level triggered interrupt request input, checked during last clock cycle of each instruction to determine the availability of request. If any interrupt request is occurred, the processor enters the interrupt acknowledge cycle.	Each correct function 1M
	b	**What is role of XCHG instruction in assembly language program? Give example**	**2M**
	Ans	**Role of XCHG:** This instruction exchanges the contents of a register with the contents of another register or memory location. **Example:** XCHG AX, BX ; Exchange the word in AX with word in BX.	Correct role:1M Correct example : 1M

			(any other example allowed)
	c	**List assembly language programming tools.**	**2M**
	Ans	1. Editors 2. Assembler 3. Linker 4. Debugger.	Each ½ M
	d	**Define Macro.Give syntax.**	**2M**
	Ans	**Macro:** Small sequence of the codes of the same pattern are repeated frequently at different places which perform the same operation on the different data of same data type, such repeated code can be written separately called as Macro. **Syntax:** Macro_name MACRO[arg1,arg2,.....argN) ..... End	Definition1M Syntax 1M
	e	**Draw flowchart for multiplication of two 16 bit numbers.**	**2M**
	Ans		Correct flowchart: 2M(consider any relevant flowchart also)
	f	**Draw machine language instruction format for Register-to-Register transfer.**	**2M**

	Ans		Correct diagram 2M
	g	**State the use of STC and CMC instruction of 8086.**	**2M**
	Ans	STC – This instruction is used to Set Carry Flag. CF←1 CMC – This instruction is used to Complement Carry Flag. CF← ~ CF	Each correct use 1M
2.		**Attempt any Three of the following:**	**12M**
	a	**Give the difference between intersegment and intrasegment CALL**	**4M**
	Ans		Any 4 points 1M each

Sr.no	Intersegment Call	Intrasegment Call
1.	It is also called Far procedure call	It is also called Near procedure call.
2.	A far procedure refers to a procedure which is in the different code segment from that of the call instruction.	A near procedure refers to a procedure which is in the same code segment from that of the call instruction
3	This procedure call replaces the old CS:IP pairs with new CS:IP pairs	This procedure call replaces the old IP with new IP.
4.	The value of the old CS:IP pairs are pushed on to the stack SP=SP-2 ;Save CS on stack SP=SP-2 ;Save IP (new offset address of called procedure)	The value of old IP is pushed on to the stack. SP=SP-2 ;Save IP on stack(address of procedure)
5.	More stack locations are required	Less stack locations are required

		6.	Example :- Call FAR PTR Delay	Example :- Call Delay	
	b	**Draw flag register of 8086 and explain any four flags.**			4M
	Ans	**Flag Register of 8086**			Correct diagram 2M Any 4 flag explanation :1/2 M each



Conditional /Status Flags

C-Carry Flag : It is set when carry/borrow is generated out of MSB of result. (i.e D_7 bit for 8-bit operation, D_{15} bit for a 16 bit operation).

P-Parity Flag This flag is set to 1 if the lower byte of the result contains even number of 1's otherwise it is reset.

AC-Auxiliary Carry Flag This is set if a carry is generated out of the lower nibble, (i.e. From D3 to D4 bit)to the higher nibble

Z-Zero Flag This flag is set if the result is zero after performing ALU operations. Otherwise it is reset.

S-Sign Flag This flag is set if the MSB of the result is equal to 1 after performing ALU operation , otherwise it is reset.

O-Overflow Flag This flag is set if an overflow occurs, i.e. if the result of a signed operation is large enough to be accommodated in destination register.

Control Flags

T-Trap Flag If this flag is set ,the processor enters the single step execution mode.

I-Interrupt Flag it is used to mask(disable) or unmask(enable)the INTR interrupt. When this flag is set,8086 recognizes interrupt INTR. When it is reset INTR is masked.

		D-Direction Flag It selects either increment or decrement mode for DI &/or SI register during string instructions.	
	c	**Explain assembly language program development steps.**	**4M**
	Ans	**1. Defining the problem:** The first step in writing program is to think very carefully about the problem that the program must solve. **2. Algorithm:** The formula or sequence of operations to be performed by the program can be specified as a step in general English is called algorithm. **3. Flowchart:** The flowchart is a graphically representation of the program operation or task. **4. Initialization checklist:** Initialization task is to make the checklist of entire variables, constants, all the registers, flags and programmable ports **5. Choosing instructions**: Choose those instructions that make program smaller in size and more importantly efficient in execution. **6. Converting algorithms to assembly language program:** Every step in the algorithm is converted into program statement using correct and efficient instructions or group of instructions.	Correct steps 4M
	d	**Explain logical instructions of 8086.(Any Four)**	**4M**
	Ans	**Logical instructions.** **1) AND- Logical AND** **Syntax : AND destination, source** **Operation** **Destination ←destination AND source** **Flags Affected :CF=0,OF=0,PF,SF,ZF** This instruction AND's each bit in a source byte or word with the same number bit in a destination byte or word. The result is put in destination. **Example: AND AX, BX** • **AND AL,BL** • **AL 1111 1100** • **BL 0000 0011** **----------** • **AL←0000 0000 (AND AL,BL)** **2) OR – Logical OR** Syntax **:OR destination, source**	Any 4 instruction correct explanation 1M each

Operation

Destination ← OR source

Flags Affected :CF=0,OF=0,PF,SF,ZF

This instruction OR's each bit in a source byte or word with the corresponding bit in a destination byte or word. The result is put in a specified destination.

Example :

- OR AL,BL
- AL 1111 1100
- BL 0000 0011

- AL←1111 1111

3) NOT – Logical Invert

Syntax : NOT destination

Operation: Destination← NOT destination

Flags Affected :None

The NOT instruction inverts each bit of the byte or words at the specified destination.

Example

NOT BL

BL = 0000 0011

NOT BL gives 1111 1100

4) XOR – Logical Exclusive OR

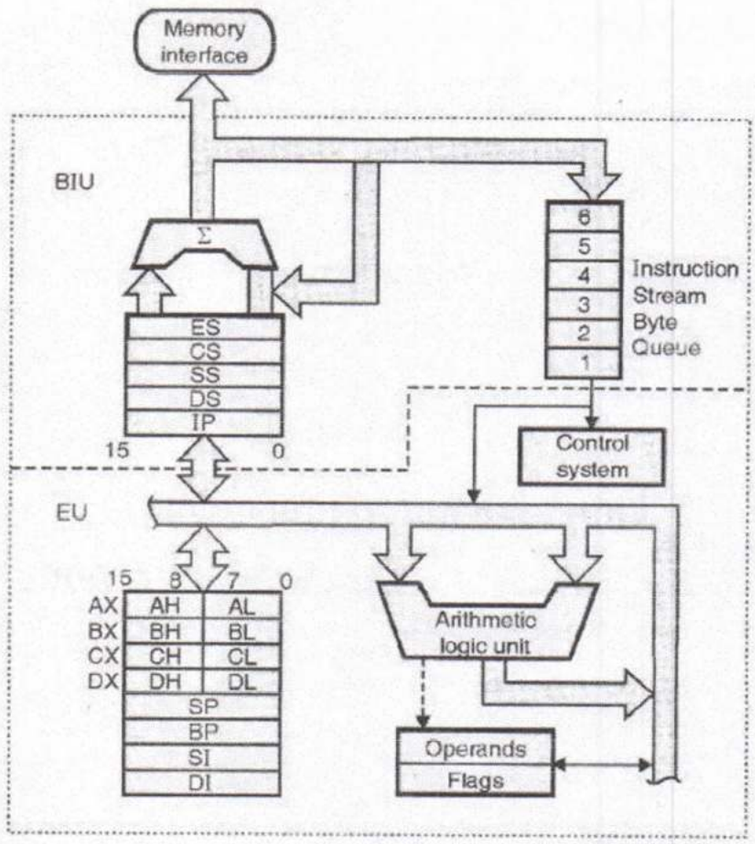
Syntax : **XOR destination, source**

Operation : **Destination Destination XOR source**

Flags Affected :CF=0,OF=0,PF,SF,ZF

This instruction exclusive, OR's each bit in a source byte or word with the same number bit in a destination byte or word.

		Example(optional) **XOR AL,BL** • AL 1111 1100 • BL 0000 0011 --- • **AL←1111 1111 (XOR AL,BL)** **5)TEST** **Syntax : TEST Destination, Source** This instruction AND's the contents of a source byte or word with the contents of specified destination byte or word and flags are updated, , flags are updated as result ,but neither operands are changed. **Operation performed:** Flags ⟵ set for result of (destination AND source) **Example: (Any 1)** TEST AL, BL ; AND byte in BL with byte in AL, no result, Update PF, SF, ZF. e.g **MOV AL, 00000101** **TEST AL, 1 ; ZF = 0.** **TEST AL, 10b ; ZF = 1**	
3.		**Attempt any Four of the following:**	
	a	**Draw functional block diagram of 8086 microprocessor.**	**4M**
	Ans		Block diagram 4M



8086 internal architecture

	b	**Write an ALP to add two 16-bit numbers.**	**4M**
	Ans	DATA SEGMENT NUMBER1 DW 6753H NUMBER2 DW 5856H SUM DW 0 DATA ENDS CODE SEGMENT ASSUME CS: CODE, DS: DATA START: MOV AX, DATA	Data segment initialization 1M, Code segment 3M

		MOV DS, AX
		MOV AX, NUMBER1
		MOV BX, NUMBER2
		ADD AX, BX
		MOV SUM, AX
		MOV AH, 4CH
		INT 21H
		CODE ENDS
		END START	

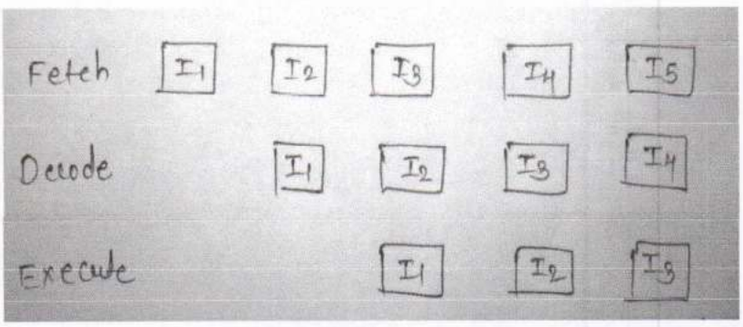
	c	**Write an ALP to find length of string.**	4M

	Ans	Data Segment
		STRG DB 'GOOD MORNING$'
		LEN DB ?
		DATA ENDS
		CODE SEGMENT
		START:
		ASSUME CS: CODE, DS : DATA
		MOV DX, DATA
		MOV DS,DX
		LEA SI, STRG
		MOV CL,00H
		MOV AL,'$'
		NEXT: CMP AL,[SI]
		JZ EXIT
		ADD CL,01H
		INC SI
			program - 4 M

		JMP NEXT EXIT: MOV LEN,CL MOV AH,4CH INT 21H CODE ENDS	
	d	**Write an assembly language program to solve $p= x^2+y^2$ using Macro.(x and y are 8 bit numbers.**	**4M**

	Ans	.MODEL SMALL
		PROG MACRO a,b
		MOV al,a
		MUL al
		MOV bl,al
		MOV al,b
		MUL al
		ADD al,bl
		ENDM
		.DATA
		x DB 02H
		y DB 03H
		p DB DUP()
		.CODE
		START:
		MOV ax,data
		MOV ds,ax
		PROG x, y
			program - 4 M

		MOV p,al
		MOV ah,4Ch
		Int 21H
		END	

4.		**Attempt any Three of the following:**	
	a	**What is pipelining? How it improves the processing speed.**	
	Ans	• In 8086, pipelining is the technique of overlapping instruction fetch and execution mechanism. • To speed up program execution, the BIU fetches as many as six instruction bytes ahead of time from memory. The size of instruction prefetching queue in 8086 is 6 bytes. • While executing one instruction other instruction can be fetched. Thus it avoids the waiting time for execution unit to receive other instruction. • BIU stores the fetched instructions in a 6 level deep FIFO . The BIU can be fetching instructions bytes while the EU is decoding an instruction or executing an instruction which does not require use of the buses. • When the EU is ready for its next instruction, it simply reads the instruction from the queue in the BIU. • This is much faster than sending out an address to the system memory and waiting for memory to send back the next instruction byte or bytes. • This improves overall speed of the processor 	Explanation 3 M, Diagram 1 M
	b	**Write an ALP to count no.of 0's in 16 bit number.**	**4M**
	Ans	DATA SEGMENT N DB 1237H Z DB 0	Program 4 M

		DATA ENDS
		CODE SEGMENT
		ASSUME DS:DATA, CS:CODE
		START:
		MOV DX,DATA
		MOV DS,DX
		MOV AX, N
		MOV CL,08
		NEXT: ROL AX,01
		JC ONE
		INC Z
		ONE: LOOP NEXT
		HLT
		CODE ENDS
		END START	

	c	**Write an ALP to find largest number in array of elements 10H, 24H, 02H, 05H, 17H.**	**4M**

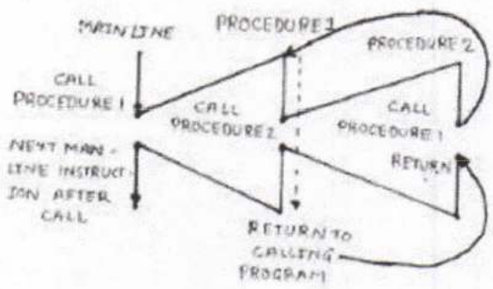
	Ans	DATA SEGMENT
		ARRAY DB 10H,24H,02H,05H,17H
		LARGEST DB 00H
		DATA ENDS
		CODE SEGMENT
		START:
		ASSUME CS:CODE,DS:DATA
		MOV DX,DATA
		MOV DS,DX
		MOV CX,04H
		MOV SI ,OFFSET
		ARRAY MOV AL,[SI]
		UP: INC SI
		CMP AL,[SI]
		JNC NEXT
		MOV AL,[SI]
		NEXT: DEC CX
		JNZ UP
		MOV LARGEST,AL
		MOV AX,4C00H
		INT 21H
		CODE ENDS
		END START
			Program - 4 M

	d	**Write an ALP for addition of series of 8-bit number using procedure.**	**4M**
	Ans	**DATA SEGMENT** NUM1 DB 10H,20H,30H,40H,50H RESULT DB 0H CARRY DB 0H	Program - 4 M

		DATA ENDS
		CODE SEGMENT
		ASSUME CS:CODE, DS:DATA
		START: MOV DX,DATA
		MOV DS, DX
		MOV CL,05H
		MOV SI, OFFSET NUM1
		UP: CALL SUM
		INC SI
		LOOP UP
		MOV AH,4CH
		INT 21H
		
		SUM PROC; Procedure to add two 8 bit numbers
		MOV AL,[SI]
		ADD RESULT, AL
		JNC NEXT
		INC CARRY
		NEXT: RET
		SUM ENDP
		CODE ENDS
		END START	

	e	**Describe re-entrant and recursive procedure with schematic diagram.**	**4M**
	Ans	In some situation it may happen that Procedure 1is called from main program Procrdure2 is called from procedure1And procrdure1 is again called from procdure2. In this situation program execution flow reenters in the procedure1. These types of procedures are called re enterant procedures. The RET instruction at the end of procrdure1 returns to procedure2. The RET instruction at the end of procedure2 will return the execution to procedure1.Procedure1 will again executed from where it had stopped at the time of calling procrdure2 and the RET instruction at the end of this will return the program execution to main program. The flow of program execution for re-entrant procedure is as shown in FIG.	**Re-entrant 2 M, recursive 2 M**

Sketch :



Recursive Procedure

A recursive procedure is a procedure which calls itself. Recursive procedures are used to work with complex data structures called trees. If the procedures is called with N (recursion depth) = 3. Then the n is decremented by one after each procedure CALL and the procedure is called until n = 0. Fig. shows the flow diagram and pseudo-code for recursive procedure.

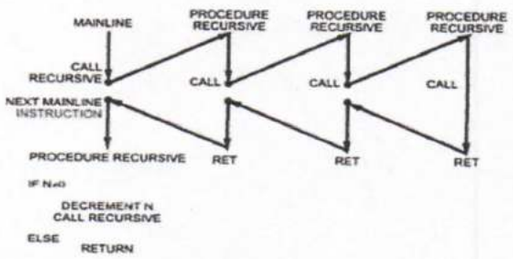
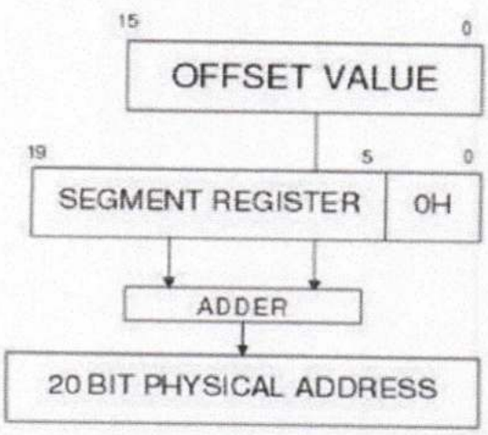


Fig. Flow diagram and pseudo-code for recursive procedure

5.		**Attempt any Two of the following:**	**12 M**
	a	**Define logical and effective address. Describe physical address generation process in 8086. If DS=345AH and SI=13DCH. Calculate physical address.**	**6M**
	Ans	**A logical address** is the address at which an item (memory cell, storage element) appears to reside from the perspective of an executing application program. A logical address may be different from the physical address due to the operation of an address translator or mapping function. **Effective Address or Offset Address**: The offset for a memory operand is called the operand's effective address or EA. It is an unassigned 16 bit number that expresses the operand's distance in bytes from the beginning of the segment in which it resides. In 8086 we have base registers and index registers.	Define each Term :1M. Physical Address Generation. Description : 2 M & Calculation 2 M

		Generation of 20 bit physical address in 8086:- 1. Segment registers carry 16 bit data, which is also known as base address. 2. BIU appends four 0 bits to LSB of the base address. This address becomes 20-bit address. 3. Any base/pointer or index register carries 16 bit offset. 4. Offset address is added into 20-bit base address which finally forms 20 bit physical address of memory location  DS=345AH and SI=13DCH Physical adress = DS*10H + SI = 345AH * 10H + 13DCH = 345A0+13DC = 3597CH	
	b	**Explain the use of assembler directives. 1) DW 2) EQU 3) ASSUME 4) OFFSET 5) SEGMENT 6) EVEN**	**2M**
	Ans	**DW (DEFINE WORD)** The DW directive is used to tell the assembler to define a variable of type word or to reserve storage locations of type word in memory. The statement MULTIPLIER DW 437AH, for example, declares a variable of type word named MULTIPLIER, and initialized with the value 437AH when the program is loaded into memory to be run. **EQU (EQUATE)** EQU is used to give a name to some value or symbol. Each time the assembler finds the given name in the program, it replaces the name with the value or symbol you equated with that name.	Each Directive Use : 1M each

		Example **Data SEGMENT** **Num1 EQU 50H** **Num2 EQU 66H** **Data ENDS** Numeric value 50H and 66H are assigned to Num1 and Num2. **ASSUME** ASSUME tells the assembler what names have been chosen for Code, Data Extra and Stack segments. Informs the assembler that the register CS is to be initialized with the address allotted by the loader to the label CODE and DS is similarly initialized with the address of label DATA. **OFFSET** OFFSET is an operator, which tells the assembler to determine the offset or displacement of a named data item (variable), a procedure from the start of the segment, which contains it. **Example** **MOV BX;** **OFFSET PRICES;** It will determine the offset of the variable PRICES from the start of the segment in which PRICES is defined and will load this value into BX. **SEGMENT** The SEGMENT directive is used to indicate the start of a logical segment. Preceding the SEGMENT directive is the name you want to give the segment. For example, the statement CODE SEGMENT indicates to the assembler the start of a logical segment called CODE. The SEGMENT and ENDS directive are used to "bracket" a logical segment containing code of data **EVEN (ALIGN ON EVEN MEMORY ADDRESS)** As an assembler assembles a section of data declaration or instruction statements, it uses a location counter to keep track of how many bytes it is from the start of a segment at any time. The EVEN directive tells the assembler to increment the location counter to the next even address, if it is not already at an even address. A NOP instruction will be inserted in the location incremented over.	
	c	**Describe any four string instructions of 8086 assembly language.**	**2M**
	Ans	**1] REP:** REP is a prefix which is written before one of the string instructions. It will cause During length counter CX to be decremented and the string instruction to be repeated until CX becomes 0.	each correct instruction 1½ M each

Two more prefix.

REPE/REPZ: Repeat if Equal /Repeat if Zero.

It will cause string instructions to be repeated as long as the compared bytes or words Are equal and CX≠0.

REPNE/REPNZ: Repeat if not equal/Repeat if not zero.

It repeats the strings instructions as long as compared bytes or words are not equal

And CX≠0.

Example: REP MOVSB

2] MOVS/ MOVSB/ MOVSW - Move String byte or word.

Syntax:

MOVS destination, source

MOVSB destination, source

MOVSW destination, source

Operation: ES:[DI]<----- DS:[SI]

It copies a byte or word a location in data segment to a location in extra segment. The offset of source is pointed by SI and offset of destination is pointed by DI.CX register contain counter and direction flag (DE) will be set or reset to auto increment or auto decrement pointers after one move.

Example

LEA SI, Source

LEA DI, destination

CLD

MOV CX, 04H

REP MOVSB

3] CMPS /CMPSB/CMPSW: Compare string byte or Words.

Syntax:

CMPS destination, source

CMPSB destination, source

CMPSW destination, source

Operation: Flags affected <------ DS:[SI]- ES:[DI]

It compares a byte or word in one string with a byte or word in another string. SI Holds the offset of source and DI holds offset of destination strings. CS contains counter and DF=0 or 1 to auto increment or auto decrement pointer after comparing one byte/word.

Example

LEA SI, Source

LEA DI, destination

CLD

MOV CX, 100

REPE CMPSB

4] SCAS/SCASB/SCASW: Scan a string byte or word.

Syntax:

SCAS/SCASB/SCASW

Operation: Flags affected <------ AL/AX-ES: [DI]

It compares a byte or word in AL/AX with a byte /word pointed by ES: DI. The string to be scanned must be in the extra segment and pointed by DI. CX contains counter and DF may be 0 or 1.

When the match is found in the string execution stops and ZF=1 otherwise ZF=0.

Example

LEA DI, destination

MOV Al, 0DH

MOV CX, 80H

CLD

REPNE SCASB

		5] LODS/LODSB/LODSW: Load String byte into AL or Load String word into AX. Syntax: LODS/LODSB/LODSW Operation: AL/AX < ----- DS: [SI] IT copies a byte or word from string pointed by SI in data segment into AL or AX.CX may contain the counter and DF may be either 0 or 1 **Example** LEA SI, destination CLD LODSB **6] STOS/STOSB/STOSW (Store Byte or Word in AL/AX)** Syntax STOS/STOSB/STOSW Operation: ES:[DI] <------AL/AX It copies a byte or word from AL or AX to a memory location pointed by DI in extra segment CX may contain the counter and DF may either set or reset	
6.		**Attempt any Two of the following:**	**12M**
	a	**Describe any 6 addressing modes of 8086 with one example each.**	**6M**
	Ans	**1. Immediate addressing mode:** An instruction in which 8-bit or 16-bit operand (data) is specified in the instruction, then the addressing mode of such instruction is known as Immediate addressing mode. **Example:** MOV AX,67D3H **2. Register addressing mode** An instruction in which an operand (data) is specified in general purpose registers, then the addressing mode is known as register addressing mode.	Any 6 mode with example 1 M each

Example:

MOV AX,CX

3. Direct addressing mode

An instruction in which 16 bit effective address of an operand is specified in the instruction, then the addressing mode of such instruction is known as direct addressing mode.

Example:

MOV CL,[2000H]

4. Register Indirect addressing mode

An instruction in which address of an operand is specified in pointer register or in index register or in BX, then the addressing mode is known as register indirect addressing mode.

Example:

MOV AX, [BX]

5. Indexed addressing mode

An instruction in which the offset address of an operand is stored in index registers (SI or DI) then the addressing mode of such instruction is known as indexed addressing mode.

DS is the default segment for SI and DI.

For string instructions DS and ES are the default segments for SI and DI resp. this is a special case of register indirect addressing mode.

Example:

MOV AX,[SI]

6. Based Indexed addressing mode:

An instruction in which the address of an operand is obtained by adding the content of base register (BX or BP) to the content of an index register (SI or DI) The default segment register may be DS or ES

Example:

MOV AX, [BX][SI]

7. Register relative addressing mode: An instruction in which the address of the operand is obtained by adding the displacement (8-bit or 16 bit) with

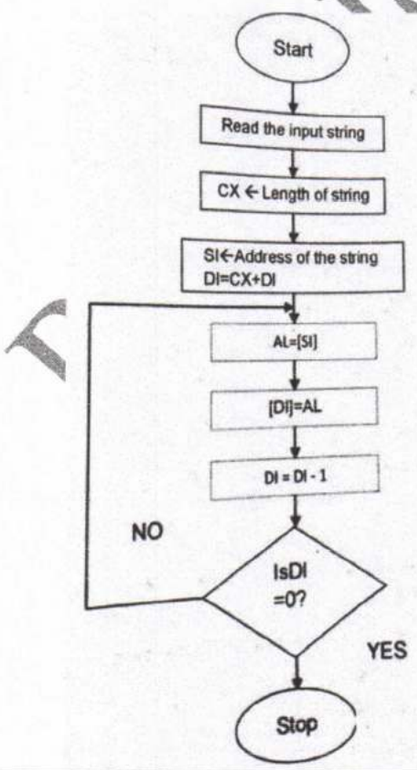
		the contents of base registers or index registers (BX, BP, SI, DI). The default segment register is DS or ES. **Example:** MOV AX, 50H[BX] **8. Relative Based Indexed addressing mode** An instruction in which the address of the operand is obtained by adding the displacement (8 bit or 16 bit) with the base registers (BX or BP) and index registers (SI or DI) to the default segment. **Example:** MOV AX, 50H [BX][SI]	
	b	**Select assembly language for each of the following** **i) rotate register BL right 4 times** **ii) multiply AL by 04H** **iii) Signed division of AX by BL** **iv) Move 2000h in BX register** **v) increment the counter of AX by 1** **vi) compare AX with BX**	**6M**
	Ans	i) MOV CL, 04H RCL AX, CL1 Or MOV CL, 04H ROL AX, CL Or MOV CL, 04H RCR AX, CL1	Each correct instruction 1M

		Or MOV CL, 04H ROR AX, CL ii) MOV BL,04h MUL BL iii) IDIV BL iv) MOV BX,2000h v) INC AX vi) CMP AX,BX	
	c	**Write an ALP to reverse a string. Also draw flowchart for same.**	

	Ans	**Program:**
		DATA SEGMENT
		STRB DB 'GOOD MORNINGS'
		REV DB 0FH DUP(?)
		DATA ENDS
		CODE SEGMENT
		START:ASSUME CS:CODE,DS:DATA
		MOV DX,DATA
		MOV DS,DX
		LEA SI,STRB
		MOV CL,0FH
		LEA DI,REV
		ADD DI,0FH
		UP:MOV AL,[SI]
			Program 4 M
			flowchart 2 M

```
MOV [DI],AL
INC SI
DEC DI
LOOP UP
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

Flowchart:



SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S

SOMESHWAR ENGINEERING COLLEGE

SOMESHWARNAGAR

SAVITRIBAI PHULE PUNE UNIVERSITY

Contents Beyond the syllabus to bridge syllabus gap

- Study of different components mounted on to CPU.
- Types of different processors and their Applications.
- Study of different case studies for better understanding of the concept.
- Study of assembly language.
- Systematic Execution of high level language code.
- Study of different coprocessors and controllers to enhance the performance of processor.
- An interfacing of processor with coprocessor and controller.

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

QUESTION BANK

Question Bank

a) State the function of ALE and Ready pin of 8086.
b) What is the role TEST instruction in Assembly language programming?
c) List the major steps in developing an Assembly language program.
d) Define Procedure and write its syntax.
e) Draw the flowchart for Multiplication of two 16 bit numbers.
f) What is stack? state its significance.
g) What is the use of REP in string related instruction?
h) Give the difference between Inter segment and Intra segment CALL.
i) What is pipelining? How it improves the processing speed?
j) State the Assembler Directives used in 8086 programming and describe the function of any two.
k) Draw the Machine language instruction format for Register to Register transfer and state the function of each bit.
l) Describe Memory segmentation in 8086 and list its advantages.
m) Write an ALP to perform 32 bit by 16-bit division of unsigned numbers.
n) Write an ALP to count number of '1' in 16-bit number.
o) Compare Procedure and macro based on i) length of code ii) generation of object code iii) Calling method iv) Passing parameter.
p) Draw and explain the flag register of 8086.
q) Write an ALP to count the number of positive and negative numbers in array.
r) Write an ALP to find the smallest number in the Array.
s) Write an ALP for addition of two 8 bit BCD numbers using MACRO.
t) Describe re-entrant and Recursive procedure with diagram.
u) Define logical and effective address. Describe physical address generation process in 8086.Calculate physical address by taking suitable DS, CS and IP.
v) Describe how an assembly language program is developed and debugged using system tools such as editors, assemblers, linkers and debuggers.
w) Describe any six Addressing modes of 8086 with suitable example.
x) With examples, describe any four String instructions in 8086 assembly language.
y) Select the instruction for each of the following
   i)Rotate register BH left 4 times.
   ii)Multiply AL by 08H.
   iii)Signed division of BL and AL.
   iv)Move 4000H in BX register.
   v)Load offset 1000H in register BX.
   vi)Rotate BX to left 4 times through carry.
z) Write an ALP for concatenation of two strings. Draw flowchart and assume suitable data.

Sample Oral questions for MA

Guidelines for oral exams

1. Oral should be based on lab assignments and theory.
2. Examiner can ask questions either on particular assignments or on all or from theory
3. Minimum Expectation from students
4. Logic and meaning of each and every instructions used in program
5. Tools used for Assembly Language programming (i.e. should know generation of .obj , .exe)
6. NASM basics
7. Different sections in NASM
8. different commands used to generate object file, linking and .exe file

Theory Related Questions:

9. What is real mode
10. What is protected mode
11. What is V86 mode
12. How switching between modes happens?
13. Structure of GDTR, IDTR, LDTR(Local Discripter), TR(Task Regi)
14. Structure of all control registers, DR registers. TR registers.
15. Structure of segment descriptors.
16. What are different types of privileges?
17. What are TLBs(Translation look ahead buffer)?
18. What are call gates?
19. What is page level protection?
20. TSS(Task segment status Resister) descriptors
21. What is difference between ADD and ADC instruction?
22. What is difference between SHR and SAR?
23. Difference between DIV and IDIV
24. Different protected mode specific instructions (for eg. SMSW)
25. Difference between loop and jnz instruction
26. What is multicore architecture?
27. What are different multicore architectures?
28. What is cache memory?
29. Different Addressing modes.
30. What is effective address?
31. What is physical Address?
32. What is linear address?
33. How to calculate effective address in 20 bit and in 32 bit architectures?
34. Which bit tells about the paging?
35. What are different segment registers in 8086 and 80386 architectures?
36. List all general purpose registers, index and pointer registers.

37. Difference from 80386 to 8086 architecture
38. Flag registers in 8086 and 80386.
39. What are privilege instructions? Give examples?
40. What is an instruction queue? Explain?
41. What are different math co-processors available?
42. Describe the pipelining architecture for microprocessors
43. Practical Related Question:
44. What are different program development tools?
45. Explain the use of assembler and linker.
46. Explain different types of memory models used.
47. Explain the use of Index registers i.e. destination and source index.
48. What is the use of MOVSB and MOVSW instruction?
49. What is the use of MACRO?
50. What is procedure?
51. State the main difference between MACRO and PROCEDURE.
52. Explain int 80h and its function call values.
53. Explain the logic of displaying number.
54. Different instructions used in lab assignments
55. What are different modes used in file reading assignment
56. What are different jump instructions?
57. Explain the syscall and its function values.
58. What are different options used in compiling and linking files in nasm
59. Learn instruction set of 80386 with examples.
60. What are different data types?
61. How to display string message on the screen?
62. How to get input from users?
63. What is option of newline?
64. What are 64 bit and 32 bit applications
65. What are different registers in 64 bit, 32 bit and 16 bit application?

Sspm's
Someshwar Engineering College
Department of Computer Engineering

Subject: Microprocessor

Test Question Paper

Q1)
(*a*) Explain Physical Address generation for 80386 if paging is enabled. [6]
(*b*) Explain Flag register of 80386. [4]

OR

Q2)
a) Explain shift and rotate instructions of 80386. [6]
b) List and explain coprocessor interface instructions of 80386. [4]

Q3)
a) With the help of diagram explain the 80386 mechanism to translate logical address to linear address. [8]

b) Explain LEA and XLAT instructions. [4]

OR

Q4)
a) What is the use of Interrupt Flag? [3]
b) Explain any *three* control transfer instructions of 80386. [6]

Q5)
a) Explain any four Flag Control Instructions. [3]
b) What is the use of the following instructions in 80386? Mention which flags gets effected with each instruction: ADC, DIV, CMP. [6]

OR

Q6)
a) Draw and explain the system address and system segment registers. [5]
b) What is the use of following instructions: Wait, LOCK. [4]

Q7)
a) Explain the following instructions, mention flags affected: CWD, BT, LAHF. [6]
b) Explain MSW. [6]

OR

SAVITRIBAI PHULE PUNE UNIVERSITY
SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S
SOMESHWAR ENGINEERING COLLEGE
SOMESHWARNAGAR

Question Bank of Unit number 1

Q1) Write note on History of 8086 microprocessor.

Q 2) what is need of segmentation?

Q 3) Describe structure of segments in physical memory

Q 4) Explain 8086 register block diagram

Q 5) Draw and Explain functional block diagram of 8086 microprocessor.

Q6) Draw and Explain Flag register of 8086.

Q7) Write note on Data types of 8086 microprocessor.

Q 10) Draw the flowchart for switching from real mode to protected mode and returning back.

Q11) Draw and explain LDT, GDT, IDT

Q12) Explain addressing Modes supported by 8086 processor.

Q13) Explain any four Data Movement Instructions.

Q14) Explain any four Binary Arithmetic Instructions.

Q15) Explain any four Decimal Arithmetic Instructions, Logical Instructions.

Q16) Explain any four Control Transfer Instructions.

Q17) Explain any four String and Character Transfer Instructions or Instructions for Block Structured Language,

Q18) Explain any four Flag Control Instructions.

Q19) Explain any four Coprocessor Interface Instructions.

Q20) Explain any four Segment Register Instructions or Miscellaneous Instructions.

SAVITRIBAI PHULE PUNE UNIVERSITY
SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S
SOMESHWAR ENGINEERING COLLEGE
SOMESHWARNAGAR

Question Bank of Unit number 2

Q 11) Draw and explain Control register of 80386.

Q2) Draw and explain test register of 80386.

Q3) Draw and explain debug register of 80386.

Q4) Explain any four Systems Instructions.

Q5) Writ note on Logical Address generation, linear address generation and Physical address generation in real mode of 80386 for Cs=1101, IP= 1010.

Q6) Explain Physical Address generation for 80386 if paging is enabled.

Q7) Explain Flag register of 80386.

Q8) Explain shift and rotate instructions of 80386.

Q9) Draw and explain the format of selector.

Q10) With the help of diagram explain the 80386 mechanism to translate logical address to linear address.

Q11) List and explain coprocessor interface instructions of 80386.

Q12) With the help of diagram explain 80386 applications register set.

Q13) Explain LEA and XLAT instructions.

Q14) Draw and explain segment descriptor.

Q15) What is the use of Interrupt Flag ?

Q16) Explain any *three* control transfer instructions of 80386.

Q17) what is the use of the following instructions in 80386? Mention which flags gets effected with each instruction: ADC, DIV, CMP.

Q18) What is the use of following instructions: Wait, LOCK.

Q19) Explain segment address translation in detail.

Q20) Draw and explain the system address and system segment registers.

Q21) Explain the following instructions, mention flags affected: CWD, BT, LAHF.

Q22) Explain MSW.

Q23) Explain the following instructions, mention flags affected: LIDT, CLD, MOVS.

SAVITRIBAI PHULE PUNE UNIVERSITY
SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S
SOMESHWAR ENGINEERING COLLEGE
SOMESHWARNAGAR

Question Bank of Unit number 3

Q 1) what is the difference between IVT of real mode and IDT of protected mode of 80386? Explain in details.

Q2) Draw the flowchart for switching from protected mode to virtual mode and returning back.

Q 3) Explain segment level protection.

Q4) Explain page level protection.

Q 5) Explain Task State Segment and TSS Descriptor.

Q6) Explain Task Register and Task Gate Descriptor in detail.

Q7) Explain Task Switching when target task at higher privilege level than the privilege level of current Task.

Q8) what is Task Linking.

Q9) Explain concept of Task Address Space.

Q10) List aspects of protection related to pages.

Q11) with appropriate diagram explain the concept of privilege levels in 80386.

Q12) How call gate descriptor is used to locate the procedure in another code segment? How protection is provided?

Q13) Define faults.

Q14) Explain how 80386 identifies interrupts.

Q15) By which two ways, 80386 allows I/O to be performed? Explain each in details.

Q16) With the help of suitable diagram, explain how call gate descriptor is used to change the privilege levels in protected mode?

Q17) Explain the procedure of handling interrupts in protected mode.

Q18) What is the role of TSS in multitasking? Explain I/O permission bitmap in TSS.

Q19) Draw the format of interrupt gate and trap gate descriptor. What is the difference between them ?

Q20) What is CPL and RPL ?

SAVITRIBAI PHULE PUNE UNIVERSITY
SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S
SOMESHWAR ENGINEERING COLLEGE
SOMESHWARNAGAR

Question Bank of Unit number 4

Q 1) Explain I/O Addressing.

Q2) List and Explain I/O Instructions.

Q3) Draw and Explain I/O permission bitmap.

Q4) Explain Protection mechanism for I/O

Q5) what are the different types of interrupt? Explain how interrupts are enabled and disabled.

Q6) Draw and explain structure of IDT.

Q7) Specify priority of interrupts while handling multiple interrupts.

Q8) List five aspects of protection in the 80386.

Q9) Draw and explain TSS.

Q10) Write short note on multitasking feature of 80386.

Q11) List different sources of interrupts and explain different ways by which 80386 can enable and disable interrupts.

Q12) Write short note on task linking.

Q13) List mechanism which provide protection for I/O functions and explain the role of IOPL in providing protection for I/O functions

SAVITRIBAI PHULE PUNE UNIVERSITY
SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S
SOMESHWAR ENGINEERING COLLEGE
SOMESHWARNAGAR

Question Bank of Unit number 5

Q 1) Explain Processor State after Reset, and Software Initialization for Real Address Mode.

Q2) Draw and explain Switching from Real to Protected Mode and vice varsa.

Q3) Explain Software Initialization for Protected Mode.

Q4) Explain TLB Testing.

Q5) Draw and explain Debug Registers.

Q6) Explain different debugging techniques.

Q7) Draw and explain Structure of V86 Stack.

Q8) Explain procedure for Entering and Leaving Virtual 8086 Mode.

Q9) Explain features of virtual mode.

Q10) Explain 80386 processor state after RESET.

Q11) Write a short note on "Switching to protected mode".

Q12) List the features of 80386 architecture that supports debugging.

Q13) With neat diagram explain the process of linear address formation in V86 mode.

Q14) Write short note on "Instruction Address Breakpoint".

Q15) What are content of CR0 register after RESET in 80386 ? Explain all related bits.

Q16) Write short note on "protection within a V86 task".

Q17) Explain various debugging features of 80386.

Q18) Explain, how test registers are used in testing TLB?

SAVITRIBAI PHULE PUNE UNIVERSITY

SHREE SOMESHWAR SHIKSHAN PRASARAK MANDAL'S

SOMESHWAR ENGINEERING COLLEGE

SOMESHWARNAGAR

Question Bank of Unit number 6

Q 1) Explain different pins of microprocessor.

Q2) Explain System Clock, Bus States, Phases in detail.

Q3) Explain pipelined read Bus cycle.

Q4) Explain Non-pipelined write Bus cycle.

Q5) Explain pipelined read followed by Non-pipelined write Bus Cycles.

Q6) Explain 80387 NDP Control Register bits for Coprocessor support.

Q7) Explain 80387 Register Stack.

Q8) Explain Data Types supported by 80387.

Q9) Explain Load and Store Instructions.

Q10) Explain Trigonometric and Transcendental Instructions.

Q11) Explain Interfacing signals of 80386DX with 80387.

Q12) Explain following signals: ADS#, READY#, NA#.

Q13)Write note on CLK2 and processor internal clock.

Q14) Explain following signals: BE0# through BE3#.

Q15) Explain following signals: PEREQ, Busy#, ERROR#.

Q16) Explain the following signals: W/R#, D/C#, M/ IO#.

Q17) Explain any four 80387 constant instructions.

Q18) Explain the following signals: INTR#, NMI#, RESET#.

Q19) Explain any six 80387 data transfer instructions.

Sspm's
Someshwar Engineering College
Department of Computer Engineering

Subject: Microprocessor

Test Question Paper

Q1)
(*a*) what is need of segmentation? [3]
(*b*) Draw and Explain functional block diagram of 8086 microprocessor. [4]

OR

Q2)
a) Draw and Explain Flag register of 8086. [3]
b) Draw the flowchart for switching from real mode to protected mode and returning back. [4]

Q3)
a) What Draw and explain LDT, GDT, IDT. [4]
b) Explain addressing Modes supported by 8086 processor. [4]

OR

Q4)
a) Explain any four Data Movement Instructions. [3]
b) Explain any four Control Transfer Instructions. [4]

Q5)
a) Explain any four Flag Control Instructions. [3]
b) Explain any four Coprocessor Interface Instructions. [4]

OR

Q6)
a) Draw and explain Control register of 80386. [4]
b) Draw and explain test register of 80386. [4]

Q7)
a) Explain Draw and explain debug register of 80386. [4]
b) Draw and explain Microprogrammed Control Unit. [4]

OR

Q8)
a) Explain any four Systems Instructions. [4]
b) Writ note on Logical Address generation, linear address generation and Physical address generation in real mode of 80386 [4]

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

PRACTICAL PLAN

	SHRI SOMESHWAR SHIKSHAN PRASARAK MANDAL'S **SOMESHWAR ENGINEERING COLLEGE, SOMESHWARNAGAR**	Record No:-ACD/R/06 Revision:-00 Date:-16/06/2014

PRACTICAL PLAN

Department: Computer **Academic Year:** 2022-23 **Semester: II** **Class:** S.E.

Subject: *Microprocessor Laboratory* **Date:**

Teaching Scheme **Lectures/Week:** 03 Hrs **Practical/Week:** 02 Hrs Tutorials/**Week:** ----

Examination Scheme: **Practical:** **PR/TW: 25** **OR/TW:** 25

Exp. No	Name of Experiment	Batch	Planned Date	Conducted Date	Sign of Faculty
1	Write X86/64 ALP to accept five 64 bit hexadecimal no. from user and store them in an array and display the accepted no.	S1	14/2	21/02/2023	
		S2	10/2	24/02/2023	
		S3	15/2	22/02/2023	
		S4	9/2	23/02/2023	
2	Write X86/64 ALP to accept a string and to display its length.	S1	21/2	28/02/2023	
		S2	17/2	3/3/2023	
		S3	22/2	1/03/2023	
		S4	16/2	2/03/2023	
3	Write X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit no.	S1	~~24~~/3	14/03/2023	
		S2	24/2	10/03/2023	
		S3	1/3	8/03/2023	
		S4	23/2	9/03/2023	
4	Write switch case driven x86/64 ALP to perform 64 bit hexadecimal arithmetic operations (+,-,*,/) using suitable macros. Define procedures for each operation.	S1	21/3	21/3/2023	
		S2	10/3	24/03/2023	
		S3	8/3	29/03/2023	
		S4	9/3	23/03/2023	

Exp. No	Name of Experiment	Batch	Planned Date	Conducted Date	Sign of Faculty
5	Write X86/64 ALP to count number of positive and negative numbers from the array.	S1	28/3	28/03/2023	
		S2	17/3	31/03/2023	
		S3	15/3	29/03/2023	
		S4	16/3	18/4/2023	
6	Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result .	S1	11/4	11/04/2023	
		S2	24/3	21/04/2023	
		S3	29/3	21/04/2023	
		S4	23/3	28/04/2023	
7	Write X86/64 ALP to detect protected mode and display the values of GDTR,LDTR,IDTR,TR and MSW register also identify CPU type using CPUID instruction.	S1	18/4	18/04/2023	
		S2	31/3	28/04/2023	
		S3	29/4	26/04/2023	
		S4	13/4	27/04/2023	
8	Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method.	S1	25/4	25/04/2023	
		S2	21/4	4/5/2023	
		S3	26/4	3/05/2023	
		S4	20/4	4/5/2023	
9	Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.	S1	2/5	2/05/2023	
		S2	28/4	4/5/2023	
		S3	3/5	3/5/2023	
		S4	27/4	4/5/2023	

Exp. No	Name of Experiment	Batch	Planned Date	Conducted Date	Sign of Faculty
10	Write an X86/64 ALP password program that operates as follows: a. Do not display what is actually typed instead display asterisk ("*"). If the password is correct display, "access is granted" else display "Access not Granted".	S1	9/5	9/5/2023	
		S2	12/5	12/5/2023	
		S3	10/5	10/5/2023	
		S4	4/5	10/5/2023	
11	Write X86 menu driven Assembly Language Program (ALP) to implement OS (DOS) commands TYPE, COPY using file operations. User is supposed to provide command line arguments.	S1	16/5	16/5/2023	
		S2	19/5	19/5/2023	
		S3	17/5	17/5/2023	
		S4	11/5	15/5/2023	
12	Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file. The text file has to be accessed during Program_1 execution and write FAR PROCEDURES in Program_2 for the rest of the processing. Use of PUBLIC and EXTERN directives is mandatory.	S1	23/5	23/5/2023	
		S2	26/5	26/5	
		S3	24/5	24/5	
		S4	18/5	18/5	

Subject In charge

Head of the Department

HEAD OF DEPARTMENT
COMPUTER ENGINEERING

Principal

PRINCIPAL
Sharadchandra Pawar College of Engineering & Technology
Someshwarnagar, Tal-Baramati, Dist-Pune, Baramati-413306

Sharadchandra Pawar College of Engineering & Technology * Someshwarnagar *

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

TEACHING PLAN

	SHRI SOMESHWAR SHIKSHAN PRASARAK MANDAL'S **SHARADCHANRA PAWAR COLLEGE OF ENGINEERING AND TCHENOLOGY, SOMESHWARNAGAR**	Record No:-ACD/R/05 Revision:-00 Date:-16/06/2014

TEACHING PLAN

Department: Computer **Academic Year:** 2022-23 **Semester:** IV

Class: S.E **Subject:** *Microprocessor* **Date:**

Teaching Scheme **Lectures/Week:** 03 **Practical/Week:** 01 **Tutorials/Week:** 00

Examination Scheme: **Insem:** 30 **Online:** NA **Endsem:** 70

Lect No	Planned Date	Topics planned	References	Method used	Conducted Date	Sign of Faculty
1	6/2	**Unit I: 80386DX- Introduction to 80386.** Brief History of Intel Processor,80386 DX Features and Architecture	1	C/B, P	9/2/2023	
2	9/2	Programmers Model, Operating Modes, Addressing Modes and data types	1	C/B, P	10/2/2023	
3	10/2	Data Movement Instructions, Binary Arithmetic Instructions	1	C/B, P	20/2/2023	
4	13/2	Decimal Arithmetic Instructions , Logical Instructions	1	C/B, P	23/2/2023	
5	16/2	Control Transfer Instructions , String and Character Transfer Instructions, Instructions for Block Structured Language	1	C/B, P	24/2/2023	
6	11/2	Flag Control Instructions, Coprocessor Interface Instructions	1	C/B, P	27/2/2023	
7	20/2	**Unit II: Bus Cycle And System Architecture**. Segment Register Instructions, Miscellaneous Instructions.	1	C/B, P	2/3/2023	
8	23/2	Initialization - Processor State after Reset.	1	C/B, P	3/3/2023	
9	23/2	Functional Pin Diagram, Functionality of I/O pins.	1	C/B, P	6/3/2023	
10	24/2	I/O Organization (Memory Banks),Basic Memory Read and Write cycles with Timing Diagram .	2	C/B, P	9/3/2023	
11	27/2	System Architecture – System Registers, System Flags	2	C/B, P	10/3/2023	
12	27/2	Memory Management Registers.	2	C/B, P	13/3/2023	

Lect No	Planned Date	Topics planned	References	Method used	Conducted Date	Sign of Faculty
13	2/3	Control Registers, Debug Registers ,	T2	P,C/B	16/3/2023	
14	2/3	Test Registers, System Instructions	T2	P,C/B	17/3/2023	
15	8/3	**Unit III: Memory Management** Global Descriptor Table	T2	P,C/B	20/3/2023	
16	9/3	Local Descriptor Table	T2	P,C/B	23/3/2023	
17	6/3	Interrupt Descriptor Table	T2	P,C/B	24/3/2023	
18	6/3	GDTR, LDTR, IDTR	T2	P,C/B	27/3/2023	
19	9/3	Formats of Descriptors and Selectors	T2	P,C/B	30/3/2023	
20	9/3	Segment Translation	T2	P,C/B	31/3/2023	
21	10/3	Page Translation	T2	P,C/B	3/4/2023	
22	10/3	Combining Segment and Page Translation	T2	P,C/B	13/4/2023	
23	13/3	**Unit IV: Protection** Need of Protection , Overview of 80386DX Protection Mechanisms	T2	P,C/B	17/4/2023	
24	16/3	Protection rings and levels	T2	P,C/B	17/4/2023	
25	17/3	Privillaged Instructions	T2	P,C/B	20/4/2023	
26	20/3	Concept of DPL,CPL,RPL,EPL	T2	P,C/B	20/4/2023	
27	23/3	Inter privilege level transfers using Call Gates	T2	P,C/B	21/4/2023	
28	24/3	Confirming Code Segments	T2	P,C/B	21/4/2023	
29	27/3	Privilage Levels and Stacks, Page level Protection	T2	P,C/B	24/4/2023	
30	31/3	Combining Segment and Page Level Protection	T2	P,C/B	27/4/2023	
31	3/4	**Unit V: Multitasking and Virtual 8086 Mode** Multitasking – Task State Segment	T2	P,C/B	27/4/2023	
32	6/4	TSS Descriptor	T2	P,C/B	28/4/2023	
33	10/4	Task Register	T2	P,C/B	4/5/2023	
34	13/4	Task Gate Descriptor	T2	P,C/B	4/5/2023	
35	17/4	Task Switching, Task Linking	T2	P,C/B	8/5/2023	

Lect No	Planned Date	Topics planned	References	Method used	Conducted Date	Sign of Faculty
36	20/4	Task Address Space, Virtual Mode - Features	T2	P,C/B	8/5/2023	
37	21/4	Memory Management in Virtual Mode	T2	P,C/B	11/5/2023	
38	24/4	Entering and Leaving Virtual Mode	T2	P,C/B	11/5/2023	
39	27/4	**Unit VI: Interrupts , Exception and Introduction to Microcontroller** Interrupts and Exceptions : Identify Interrupts, Enabling and Disabling Interrupts	T1	P,C/B	12/5/2023	
40	28/4	Priority among Simultaneous Interrupts and Exception	T1	P,C/B	15/5/2023	
41	4/5	Interrupt Descriptor Table (IDT)	T1	P,C/B	18/5/2023	
42	8/5	IDT Descriptor	T1	P,C/B	18/5/2023	
43	11/5	Interrupt Task and Interrupt Procedures	T1	P,C/B	19/5/2023	
44	12/5	Error Codes and Exception Conditions	T1	P,C/B	22/5/23	
45	15/5	Introduction to Microcontrollers: Architecture of Typical Microcontrollers	T1	P,C/B	25/5/23	
46	18/5	Difference Between Microprocessor And Microcontroller	T1	P,C/B	27/5/23	
47	19/5	Characteristics of Microcontroller	T1	P,C/B	28/5/23	
48	22/5	Application of Microcontrollers	T1	P,C/B	29/5/23	

SUMMARY

Unit No.	Title	Total no. of Lectures	Planned Date of Completion	Actual Date of Completion
1	80386DX- Introduction to 80386.	07	20/2/2023	2/3/2023
2	Bus Cycle And System Architecture.	07	2/3/2023	17/3/2023
3	Memory Management	08	10/3/2023	19/4/2023
4	Protection	08	31/3/2023	27/4/2023
5	Multitasking and Virtual 8086 Mode	08	24/4/2023	11/5/2023
6	Interrupts , Exception and Introduction to Microcontroller	07	22/5/2023	29/5/23

Text Books:

T1. Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630-011, December 1995.
T2. James Turley, —Advanced 80386 Programming Techniques‖, McGraw-Hill, ISBN: 10: 0078813425, 13: 978-0078813429.
T3. Intel 387DX Math coprocessor, Order no.: 240448-005, March1992.

Reference Books:

R1. Chris H. Pappas, William H. Murray, —80386 Microprocessor Handbooks‖, McGraw-Hill Osborne Media, ISBN-10: 0078812429, 13: 978-0078812422.
R 2. Walter A. Triebel, —The 80386Dx Microprocessor: Hardware‖, Software, and Interfacing, Pearson Education, ISBN: 0137877307, 9780137877300.
R 3. Brey, Barry B, —8086/8088, 80286, 80386 and 80486 Assembly Language Programming‖, Prentice Hall, ISBN: 13: 9780023142475.
R 4. Mohammad Rafiquzzaman, —Microprocessors: Theory and Applications: Intel and Motorola", Prentice Hall, ISBN:-10:0966498011, 13:978:0966498011.
R 5.K. Bhurchandi, A. Ray, —Advanced Microprocessors and Peripherals‖, McGraw Hill Education, Third Edition, ISBN: 978-1-25-900613-5.
R 6. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012.
R 7. Assembly Language Step-by-step: Programming with Linux, 3rd Edition, Jeff Duntemann, Wiley ISBN:-10 0470497025, ISBN-13: 978-0470497029, 2009.

Methodology Used:

C /B: Chalk & Board
P: Power point Presentation
V/A :Video Audio Lectures

Sharadchandra Pawar College of Engineering & Technology Someshwarnagar

HEAD OF DEPARTMENT COMPUTER ENGINEERING

Subject In charge **Head of the Department** **Principal**

PRINCIPAL
Sharadchandra Pawar College of Engineering & Technology
Someshwarnagar, Tal. Baramati, Dist. [illegible] Pin : 412 306)

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

CONTINUOUS ASSESSMENT RECORD

SSPM's SHARADCHANDRA P+A2:AH34AWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2022-23** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Ghadage S. S.**

Roll No.	Name of Candidate	EXP 1				EXP 2				EXP 3				EXP 4				EXP 5				EXP 6				EXP 7				EXP 8			
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
CO-201	ADSUL VAISHANAVI VIJAY	1	2	4	7	2	2	2	6	2	2	1	5	2	4	4	10	2	3	2	7	1	3	3	7	1	2	2	5	2	2	2	6
CO-202	ATOLE PRADIP SHIVAJI	1	1	4	6	1	1	3	5	1	3	3	7	1	2	3	6	1	2	4	7	2	4	4	10	2	3	2	7	1	2	4	7
CO-203	AVATE RUTUJA SUNIL	2	2	4	8	2	2	3	7	0	2	4	6	2	1	4	7	0	4	3	7	1	2	3	6	0	4	2	6	2	4	3	9
CO-204	BADGE YASHODEEP DYANESHWAR	1	3	3	7	1	3	4	8	2	2	4	8	1	2	4	7	1	1	4	6	2	1	2	5	1	2	3	6	0	0	4	4
CO-205	BHANDLAKAR POOJA BHAUSAHEB	2	4	4	10	2	4	3	9	1	2	3	6	2	3	3	8	1	0	3	4	0	3	4	7	1	2	3	6	1	4	1	6
CO-206	BHANDWALKAR SHREYASH SUNIL	2	1	3	6	1	4	0	5	0	3	2	5	2	4	2	8	2	0	2	4	0	4	3	7	0	3	3	6	2	1	3	6
CO-207	BHOSALE JAYASH MILIND	2	0	3	5	2	4	0	6	2	1	2	5	1	1	1	3	0	1	1	2	2	2	3	7	2	3	2	7	0	4	3	7
CO-208	BHOSALE KARAN JALINDAR	1	4	2	7	2	2	2	6	2	3	0	5	0	2	2	4	0	2	0	2	1	4	1	6	1	1	4	6	2	1	2	5
CO-209	BHOSALE SAMIR SHAMAKANT	1	3	0	4	2	4	2	8	1	2	2	5	1	3	3	7	2	3	3	8	0	3	2	5	1	2	3	6	0	4	2	6
CO-210	BHUJBAL RUTIKA MAHAVIR	1	3	2	6	0	1	1	2	1	2	3	6	0	4	4	8	1	4	2	7	1	2	4	7	2	3	2	7	1	0	4	5
CO-211	CHAVAN DIPTI RAMCHANDRA	2	3	2	7	1	2	2	5	2	3	2	7	2	1	4	7	2	4	2	8	1	3	2	6	2	4	2	8	2	0	3	5
CO-212	CHAVAN SAURABH DANAJI	1	3	2	6	2	4	0	6	0	2	1	4	0	3	3	6	2	4	2	8	1	1	2	4	0	1	3	4	0	4	1	5
CO-213	DHAVALE JAYASHRI ANIL	2	2	2	6	0	3	2	5	2	1	3	6	2	4	1	7	2	2	2	6	1	4	2	7	2	2	3	7	1	0	2	5
CO-214	DHOLE PRIYANKA SHIVAJI	2	1	3	6	1	3	3	7	0	2	2	4	0	2	2	4	1	3	3	7	0	4	3	7	1	4	2	7	2	2	3	7
CO-215	DHUMAL ATHARVE RAVINDRA	0	2	3	5	2	2	0	4	1	2	2	5	1	2	0	3	2	2	3	7	2	3	2	7	0	1	3	4	0	4	1	5
CO-216	GADADE ONKAR GORAKH	1	2	2	5	2	3	2	7	1	4	3	8	0	4	4	8	1	3	3	7	0	0	4	4	0	3	2	5	1	1	5	7
CO-217	GAIKWAD GAYATRI BALASO	1	2	3	6	0	0	4	4	1	2	3	6	0	3	3	6	2	1	3	6	2	4	3	9	2	3	3	6	2	2	3	7
CO-218	GAIKWAD SAYALI RAMESH	1	2	1	4	2	1	1	4	2	3	3	8	1	1	2	4	2	0	1	3	0	2	3	5	2	3	2	7	2	1	0	3
CO-219	GANGANNNA RAHUL SANJAY	2	1	2	5	0	2	4	6	2	1	2	5	2	2	4	8	1	1	3	5	1	2	2	5	0	1	4	5	1	2	2	5
CO-220	GHADGE VAIBHAVI BALASAHEB	2	1	3	6	2	0	2	4	1	3	3	7	1	2	4	7	2	2	4	8	1	2	3	6	0	2	3	5	0	4	3	7
CO-221	GHADGE VAISHNAVI BALASAHEB	0	3	1	4	2	4	2	8	2	2	4	8	2	2	3	7	1	3	2	6	1	3	2	6	1	2	3	6	0	4	2	6
CO-222	HALNOR KOMAL BALASAHEB	2	4	2	8	1	3	2	6	0	4	2	6	2	4	2	8	2	4	3	9	0	3	4	7	2	3	2	7	2	1	3	6
CO-223	INAMDAR ARBAJ TAYYAB	0	4	3	7	1	2	3	6	2	4	1	7	1	4	1	6	2	2	3	7	2	4	2	8	1	1	4	6	2	1	2	5
CO-224	JADHAV AVANTIKA RAHUL	0	4	3	7	0	4	2	6	0	1	4	5	0	3	3	6	2	4	3	9	1	2	3	6	1	3	2	6	1	2	3	6
CO-225	JADHAV SNEHA NITIN	2	3	2	7	0	4	3	7	1	4	3	8	1	1	2	4	1	2	2	5	0	3	2	5	1	2	4	7	2	0	3	5
CO-226	JAGDALE GANESH JALINDAR	2	3	1	6	0	3	2	5	1	2	4	7	2	2	2	6	1	4	3	8	2	1	3	6	1	4	4	9	0	2	2	4
CO-227	JAGTAP AKSHADA ANKUSH	1	4	2	7	0	3	1	4	1	2	2	5	2	0	0	2	1	3	4	8	1	1	3	4	1	3	3	7	2	2	3	7
CO-228	JOSHI ADITYA SANJAY	0	4	4	8	1	2	2	5	2	4	1	7	2	1	4	7	2	2	2	6	0	4	1	5	2	3	2	7	1	3	1	4
CO-229	KADAM BHUSHAN MALLIKARJUN	1	2	2	5	2	1	0	3	2	1	3	6	2	2	3	7	1	0	2	3	1	4	3	8	2	2	3	7	1	1	4	6
CO-230	KADAM NEELRAJ SATYAWAN	1	2	2	5	2	1	0	3	1	3	2	6	1	0	2	3	2	2	4	8	1	4	2	7	0	3	4	7	0	4	2	6
CO-231	KADAM SAKSHI RAJENDRA	1	3	3	7	0	3	2	5	2	2	2	6	1	0	1	2	2	2	3	7	2	3	2	7	0	4	2	6	0	1	3	4
CO-232	KAKADE SAURABH RUSHIKANT	2	3	2	7	1	2	1	4	2	2	4	8	0	3	1	4	1	3	1	4	2	1	2	5	1	2	2	5	1	1	1	3
CO-233	KAKADE SHIVANJALI MOHAN	1	2	1	4	2	1	0	3	1	4	0	5	1	3	2	6	1	4	2	7	1	2	3	6	2	2	4	8	1	1	2	4
CO-234	KALASKAR GANESH DNYANESHWAR	2	1	1	4	2	1	2	6	1	2	3	6	2	3	0	5	2	1	3	6	2	3	3	8	1	4	3	8	1	1	3	5
CO-235	KARNWAR RUTUJA BHAUSAHEB	0	3	2	5	1	2	2	5	2	3	1	6	2	3	4	9	1	4	4	9	0	4	2	6	2	1	2	5	2	3	0	5
CO-236	KHANDALE ASHWINI ANIL	2	2	0	4	0	3	1	4	2	1	1	4	2	1	1	4	2	3	3		2	2	1	5	1	0	4	5	2	3	1	6

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA P+A2:AH34AWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2022-23** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Ghadage S. S.**

Roll No.	Name of Candidate	EXP 1				EXP 2				EXP 3				EXP 4				EXP 5				EXP 6				EXP 7				EXP 8			
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
CO-237	KINGARE PAYAL DATTATRAY	2	2	1	5	2	1	2	5	1	1	2	4	1	3	4	8	1	3	2	7	1	3	2	6	2	1	2	5	2	1	2	5
CO-238	KUTAL PRATIKSHA RAJENDRA	0	3	4	7	0	3	4	7	0	3	4	7	1	0	1	2	1	4	2	7	0	1	4	5	1	3	3	7	0	3	4	7
CO-239	LAMBOTE MAYA TANAJI	1	1	2	4	2	0	1	3	1	0	1	2	2	2	3	7	2	1	0	3	2	0	1	3	0	2	0	2	1	0	4	5
CO-240	MAKWANA YASH RANJIT	1	0	0	1	1	2	3	6	2	2	3	7	2	4	0	6	2	1	2	5	2	2	3	7	2	4	2	8	1	3	2	6
CO-241	MANE ROHAN SANJAY	0	4	3	7	2	4	0	6	1	4	0	5	2	1	2	5	1	3	4	8	1	4	0	5	1	3	1	5	2	1	1	4
CO-242	MANE SAURABH SANJAY	2	2	2	6	1	1	2	4	1	1	2	5	0	3	4	7	1	0	1	2	1	1	2	4	1	2	0	3	2	2	3	7
CO-243	MARDANE JAY PRAVIN	2	2	1	5	0	3	4	7	2	3	4	9	1	0	1	2	0	2	3	5	0	3	4	7	0	1	2	3	1	4	0	5
CO-244	MOHITE YASHRAJ SHRIKANT	1	1	4	6	2	0	1	3	2	4	3	9	2	2	3	7	2	4	0	6	2	0	1	3	1	3	4	8	2	1	2	5
CO-245	MUNDHE AKSHAY HARIPANDIT	1	3	2	6	1	2	3	6	1	3	2	6	2	4	0	6	1	1	2	4	2	2	3	7	1	0	1	2	0	3	4	7
CO-246	NAGAWADE SAURAV SUBARAV	0	4	3	7	1	4	0	5	1	1	2	4	1	1	2	4	0	3	4	7	1	4	0	5	2	2	3	7	0	0	1	1
CO-247	NAIKWADE ARTI HANMANT	2	4	3	9	0	1	2	3	1	3	4	8	1	3	4	8	1	0	1	2	1	1	2	4	1	4	0	5	1	2	3	6
CO-248	NARUTE KIRAN LALASO	0	3	0	3	2	3	4	9	2	0	1	3	1	0	1	2	0	2	3	5	0	3	4	7	2	1	2	5	1	4	0	5
CO-249	NIGADE RAHUL DYANESHWAR	1	0	2	3	2	0	1	3	0	2	3	5	2	2	3	7	2	4	0	6	0	0	1	1	1	3	4	8	2	1	2	5
CO-250	PAPAL KOMAL MARUTI	2	0	3	5	1	2	3	6	2	4	0	6	2	4	0	6	2	1	2	5	1	2	3	6	1	0	1	2	1	3	4	8
CO-251	PATIL PRAJAKTA VIKAS	1	3	4	8	0	4	0	4	1	1	2	4	0	1	2	3	1	3	4	8	2	4	0	6	2	2	3	7	2	0	1	3
CO-252	PAWAR ABHISHEK MADHAV	1	2	1	4	1	1	2	4	1	3	4	8	0	3	4	7	1	0	1	2	2	1	2	5	1	4	0	5	2	2	3	7
CO-253	PAWAR AYUSHKA BHANUDAS	2	2	0	4	0	3	4	7	2	0	1	3	2	0	2	4	0	2	3	5	1	3	4	8	1	1	2	4	1	4	0	5
CO-254	PAWAR ROHIT BALASO	2	3	0	5	2	0	1	3	2	2	3	7	1	1	3	6	2	4	0	6	1	0	1	2	3	3	4	9	2	0	4	6
CO-255	PAWAR TEJAS PRADIP	1	4	2	7	0	2	3	5	2	4	0	6	1	4	0	5	2	1	2	5	2	2	3	7	2	0	1	3	1	3	2	6
CO-256	RANAWARE ADITYA DIPAK	0	4	4	8	1	4	0	5	1	1	2	4	1	1	2	4	1	3	4	8	1	4	0	5	0	2	3	6	1	1	0	2
CO-257	RASKAR MANJUSHA SAVATA	1	3	3	7	2	1	2	5	1	3	4	8	2	3	4	9	1	0	1	2	2	1	2	5	2	4	0	7	2	1	2	5
CO-258	RAUT ADITI SHIVAJI	1	3	1	5	2	3	4	9	2	0	1	3	2	0	1	3	2	2	3	7	1	3	4	8	2	1	2	5	1	3	4	8
CO-259	RAUT SARTHAK SAMDAS	1	4	2	7	1	0	1	2	1	2	3	6	1	2	3	6	1	4	0	5	2	0	1	3	1	3	4	8	2	0	1	3
CO-260	ROKADE DYANESHWARI MOHAN	2	2	3	7	1	2	3	6	1	4	0	5	2	4	0	6	2	1	2	5	1	2	3	6	1	0	1	2	0	2	3	5
CO-261	SALUNKE NAMRATA CHAKRADHAR	0	3	1	4	1	4	0	5	2	1	2	5	1	1	2	4	1	3	4	8	1	4	0	5	2	2	3	7	1	4	0	5
CO-262	SALUNKE SWARAJ NITIN	0	2	4	6	2	1	2	5	0	3	4	7	2	3	4	9	2	0	1	3	0	1	2	3	2	4	0	6	2	1	2	5
CO-263	SALUNKHE SAKSHI RAJENDRA	2	3	0	5	0	3	4	9	2	0	1	3	0	0	1	1	2	2	3	7	1	3	4	8	1	1	2	4	1	3	4	8
CO-264	SALUNKHE VIVEK RAJESH	2	3	1	6	1	0	1	2	2	2	3	7	0	2	3	5	1	4	0	5	0	0	1	1	1	3	4	8	1	0	1	2
CO-265	SHINDE ISHA SURESH	1	2	2	5	0	2	3	5	1	4	0	5	1	4	0	5	1	1	2	4	1	2	3	6	2	4	2	8	2	2	3	7
CO-266	SHINDE SANDHYA NANDKUMAR	2	4	0	6	1	4	0	5	1	1	2	4	0	1	2	3	2	3	4	9	2	4	0	6	1	1	2	4	1	4	0	5
CO-267	SONAWANE ABHAY JAGDISH	1	4	3	8	1	1	2	4	2	3	4	9	0	3	4	7	2	0	1	3	0	1	2	3	2	3	4	9	0	1	2	3
CO-268	TAKAWALE VAISHNAVI VITTHAL	1	0	2	3	2	3	4	9	1	0	1	2	2	0	1	3	2	2	3	7	1	3	4	8	2	0	1	3	1	3	4	8
CO-269	TANTAK ADITI MADHUKAR	0	1	3	4	2	0	1	3	2	2	3	7	2	2	3	7	1	4	0	5	2	0	1	3	1	2	3	0	1	0	1	2
CO-270	TAWARE SAKSHI SANDIP	2	2	4	8	1	2	3	6	1	4	0	5	1	4	0	5	0	1	2	3	1	2	3	6	1	4	0	5	2	2	3	7
CO-271	THAKRE RAJ NARESH	2	1	3	6	2	4	0	6	1	1	2	4	1	1	2	4	0	3	4	7	2	4	0	6	0	1	2	5	1	4	0	5
CO-272	YADAV POOJA BHIMRAO	1	3	2	6	1	1	2	4	1	3	4	8	2	3	4	9	1	0	1	2	2	2	2	6	2	3	4	9	2	4	4	10

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2021-22** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Jagtap T.V.**

Roll No.	Name of Candidate	EXP 9				EXP 10				EXP 11				EXP 12				Total	Out of 25
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D		
CO201	Shinde Kiran Satish	1	1	2	4	0	1	4	5										
CO202	Bandal Saurabh Hanumant	2	3	4	9	1	4	3	8										
CO203	Bhosale Aishwarya Sanjay	0	0	1	1	2	4	2	8										
CO204	Bunge Sakshi Dnyaneshwar	1	2	3	6	1	4	1	6										
CO205	Chavan Pushpak Mahadev	2	4	0	6	2	3	0	5										
CO206	Durgade Avdhut Mohan	0	1	2	3	2	3	1	6										
CO207	Gawade Swapnali Uttam	0	3	4	7	0	3	2	5										
CO208	Golande Ashwini Dilip	1	1	2	4	1	1	3	5										
CO209	Jagtap Apeksha Navnath	2	3	4	9	2	1	4	7										
CO210	Jagtap Prajakta Sadanand	1	0	1	2	0	1	3	4										
CO211	Kharade Kiran Macchindra	0	2	3	5	1	1	0	2										
CO212	Kolte Vishal Nayabrao	2	4	0	6	2	1	1	4										
CO213	Mane Rohit Mahadev	1	0	1	2	2	1	2	5										
CO214	Durgade Mayuresh Sandip	2	2	3	7	2	3	3	8										
CO215	More Somesh Dnyaneshwar	1	4	0	5	1	1	4	6										
CO216	More Sukhada Rajendra	2	1	2	5	1	3	4	8										
CO217	Mutekar Himmat Bapurao	2	3	4	9	0	4	4	8										
CO218	Nigade Aditya Dnyaneshwar	2	0	1	3	1	2	4	7										
CO219	Sanas Diya Atul	2	2	3	7	2	0	2	4										
CO220	Sankpal Revati Shantaram	2	4	0	6	2	2	0	4										
CO221	Shendage Namrata Ramesh	1	1	2	4	0	4	1	5										
CO222	Shinde Ayush Shahaji	1	3	4	8	2	1	3	6										
CO223	Shinde Kaveri Rajendra	0	0	1	1	2	3	0	5										
CO224	Shraddha Shahaji Gadekar	1	2	2	5	0	4	2	6										
CO225	Taware Prasad Sanjay	2	3	4	9	1	2	1	4										
CO226	Tekawade Yogesh Uttam	0	4	3	7	1	0	2	3										
CO227	Viraj Vikram Khalate	1	2	1	4	1	3	4	8										
CO228	Wakase Sakshi Ashok	2	1	2	5	1	3	4	8										
CO229	Surve Divya Deveedas	0	3	4	7	2	4	2	8										
CO230	Shinde Sunny Harishchandra	2	0	1	3	2	0	0	2										
CO231	Kewat Rutik Sunil	0	2	3	5	2	2	2	6										
CO232	Rananavare Shraddha Sudanrao	1	4	1	6	1	3	2	6										
CO233	Adsul Yash Dnyaneshwar	0	3	2	5	0	1	3	4										
CO234	Kadam Gaurav Vikas	0	1	3	4	1	2	0	3										
CO235	Pisal Sakshi Ganesh	2	3	2	7	0	3	3	6										
CO236	Bhandwalkar Rutik Santosh	2	4	2	8	1	4	0	5										

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering	**Academic Year: 2021-22**	**Semester: II**
Class: SE (Computer)	**Subject: Microprocessor**	**Subject In Charge: Prof. Jagtap T.V.**

Roll No.	Name of Candidate	EXP 9				EXP 10				EXP 11				EXP 12				Total	Out of 25
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D		
CO237	Suryawanshi Manjiri Hanumant	0	1	2	8	1	1	2	4										
CO238	Adsul Prasad Pralhad	1	3	4	8	2	3	4	9										
CO239	Thorat Snehal Santosh	1	0	4	5	0	4	3	7										
CO240	Jamdade Vaibhav Arvind	1	3	2	6	1	2	1	4										
CO241	Hajare Tushar Kisan	1	1	0	2	0	0	1	1										
CO242	Gaikwad Revati Dipak	2	1	2	5	2	2	3	7										
CO243	Navale Ishita Ravindra	2	3	4	9	2	4	0	6										
CO244	Shilimkar Vaibhavi Sarjerao	0	0	1	1	0	1	2	3										
CO245	Jagtap Pruthviraj Udayshing	1	2	3	6	1	3	4	8										
CO246	Tamhane Shivani Ramesh	2	4	0	6	2	0	1	3										
CO247	Prathamesh Sharad Hande	0	1	2	3	0	2	3	5										
CO248	Howal Geeta Sanjay	0	3	4	7	1	4	0	5										
CO249	Gadekar Shrutika Bhanudas	2	0	1	3	2	1	2	5										
CO250	Sutar Rashika Ramesh	2	2	3	7	0	3	4	7										
CO251	Chavan Apeksha Dilip	2	4	0	6	2	0	1	3										
CO252	Pansare Vishal Sanjay	2	1	2	5	2	2	3	7										
CO253	Kawale Sagar Dattatraya	2	3	4	9	0	4	0	4										
CO254	Khomane Gaurav Vijay	2	0	1	3	2	1	2	5										
CO255	Dhumal Snehal Navnath	1	2	3	6	1	3	4	8										
CO256	Garud Renuka Pravin	2	4	0	6	0	0	1	1										
CO257	Suryavanshi Vaishnav P.	0	1	2	3	2	2	3	7										
CO258	Shinde Vishnukant Laxman	1	3	4	8	2	4	0	6										
CO259	Tambe Aishwarya Sanjay	1	0	1	2	0	1	2	3										
CO260	Jagdale Swapnil Somnath	2	2	3	7	1	3	4	8										
CO261	Deshmukh Medhavi Suryakant	0	4	0	4	2	0	1	3										
CO262	Punde Siddhi Ganesh	1	1	2	4	2	2	3	7										
CO263	Omkar Anand Mandlik	0	3	4	7	0	4	0	4										
CO264	Gadhave Sonali Pandurang	1	0	0	1	1	1	2	4										
CO265	Baile Pranita Adhir	0	4	3	7	1	3	4	8										
CO266	Pisal Vaishnavi Ajit	0	2	1	3	2	0	1	3										
CO267	Gole Diksha Mohan	2	0	1	3	0	2	3	5										
CO268	Kalyankar Omkar Amar	2	2	3	7	2	4	0	6										
CO269	Kumbhar Rohit Anil	0	4	0	4	2	1	2	5										
CO270	Tompe Anuja Prakash	2	1	2	5	0	3	4	7										
CO271	Dange Samiksha Sanjay	2	3	4	9	1	0	1	2										
CO272	Kamble Abhishek Sanjay	0	0	4	4	1	2	3	6										

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2021-22** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Jagtap T.V.**

Roll No.	Name of Candidate	EXP 1				EXP 2				EXP 3				EXP 4				EXP 5				EXP 6				EXP 7				EXP 8			
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
CO201	Shinde Kiran Satish	1	2	3	6	2	2	3	7	1	2	4	7	1	2	3	6	2	2	2	6	1	2	3	6	1	2	3	6	1	2	3	6
CO202	Bandal Saurabh Hanumant	2	4	2	7	2	0	4	6	2	0	3	5	2	4	2	8	1	4	0	5	2	2	3	7	2	4	0	6	2	4	4	10
CO203	Bhosale Aishwarya Sanjay	0	4	3	5	1	2	3	6	0	3	2	5	1	1	2	4	2	3	2	7	0	4	2	6	0	4	4	8	0	2	3	5
CO204	Bunge Sakshi Dnyaneshwar	2	4	2	6	2	0	3	5	2	4	1	7	2	3	4	9	0	4	2	6	2	4	2	8	0	3	2	5	2	2	4	8
CO205	Chavan Pushpak Mahadev	2	3	3	7	0	0	4	4	1	1	0	2	1	2	3	6	0	1	4	5	2	2	0	4	2	3	3	8	1	4	3	8
CO206	Durgade Avdhut Mohan	1	3	1	5	2	3	0	5	0	2	4	6	2	4	2	8	1	4	2	7	2	1	3	6	0	3	2	5	0	3	2	5
CO207	Gawade Swapnali Uttam	2	2	3	7	2	3	3	8	2	0	2	4	1	3	2	6	2	1	3	6	0	3	2	5	2	1	2	5	0	3	2	5
CO208	Golande Ashwini Dilip	2	2	1	5	1	2	3	6	2	3	3	8	2	2	3	7	2	0	2	4	1	2	4	7	1	3	0	4	1	1	2	4
CO209	Jagtap Apeksha Navnath	0	4	3	7	2	4	0	6	2	4	1	7	1	2	3	6	0	3	3	6	1	4	2	7	1	2	3	6	2	3	2	7
CO210	Jagtap Prajakta Sadanand	0	4	4	8	0	3	3	6	1	0	0	1	2	0	2	4	0	0	1	1	1	3	1	5	2	4	2	8	2	3	0	5
CO211	Kharade Kiran Macchindra	1	2	2	5	0	0	3	3	1	4	0	5	1	0	3	4	1	3	2	6	0	2	2	4	2	0	3	5	2	0	3	5
CO212	Kolte Vishal Nayabrao	2	2	3	7	2	1	2	5	1	1	1	3	2	1	0	3	2	3	3	8	2	2	3	7	2	3	0	5	1	4	4	9
CO213	Mane Rohit Mahadev	2	0	4	6	0	3	3	6	0	2	2	4	1	2	3	6	1	2	1	4	0	2	3	5	2	1	2	5	1	0	5	6
CO214	Durgade Mayuresh Sandip	2	4	4	10	1	2	1	4	2	0	3	5	2	2	0	4	2	4	3	9	2	3	2	7	1	1	3	5	0	3	2	5
CO215	More Somesh Dnyaneshwar	0	4	2	8	2	1	2	5	1	3	4	8	1	2	3	6	2	2	1	5	0	1	2	3	0	2	4	6	1	4	2	7
CO216	More Sukhada Rajendra	1	0	4	5	0	2	4	6	2	4	4	10	2	3	3	8	2	4	2	8	1	0	3	4	1	3	1	5	2	3	2	7
CO217	Mutekar Himmat Bapurao	2	3	3	8	2	2	2	6	1	3	4	8	1	3	3	7	1	2	3	6	1	3	2	6	0	4	2	6	0	0	4	4
CO218	Nigade Aditya Dnyaneshwar	1	3	2	6	1	3	0	4	0	1	4	5	2	4	3	9	0	3	2	5	1	4	3	8	2	4	0	6	2	2	1	5
CO219	Sanas Diya Atul	2	2	1	5	1	2	0	3	2	2	4	8	1	4	1	6	1	3	3	7	1	0	0	1	2	3	1	6	2	1	2	5
CO220	Sankpal Revati Shantaram	0	1	3	4	0	4	3	7	1	0	4	5	2	0	3	5	1	0	2	3	0	2	3	5	0	2	2	4	2	1	3	6
CO221	Shendage Namrata Ramesh	2	4	1	7	2	4	1	7	2	3	4	9	0	4	2	6	1	0	3	4	2	4	3	9	2	1	1	4	1	3	2	6
CO222	Shinde Ayush Shahaji	2	4	0	6	0	1	3	4	2	3	2	7	0	4	1	5	1	4	3	8	2	1	2	5	1	4	1	6	2	3	4	9
CO223	Shinde Kaveri Rajendra	1	0	3	4	2	2	2	6	0	2	2	4	0	0	4	4	0	0	0	0	2	3	2	7	0	3	1	4	0	4	2	6
CO224	Shraddha Shahaji Gadekar	1	3	2	6	0	0	4	4	1	1	0	2	1	4	3	8	1	4	2	7	2	4	2	8	2	4	1	7	2	1	3	6
CO225	Taware Prasad Sanjay	0	3	1	4	1	4	3	8	1	2	1	4	2	4	2	8	0	2	3	5	2	0	3	5	1	4	1	6	0	4	4	8
CO226	Tekawade Yogesh Uttam	0	4	2	6	2	2	1	5	2	0	2	4	0	3	3	6	1	2	3	7	0	3	2	5	1	2	1	4	1	2	2	5
CO227	Viraj Vikram Khalate	2	4	1	7	2	2	2	6	2	3	0	5	2	2	3	7	2	3	2	7	1	3	3	7	0	2	2	4	2	3	3	8
CO228	Wakase Sakshi Ashok	1	4	0	5	0	2	3	5	0	3	3	6	1	1	2	4	2	1	2	5	0	3	2	5	2	3	2	7	2	0	2	4
CO229	Surve Divya Deveedas	2	1	3	6	0	2	0	2	2	2	0	4	0	2	3	5	2	4	4	10	2	0	2	4	2	3	2	7	0	2	3	5
CO230	Shinde Sunny Harishchandra	2	2	0	4	1	2	2	5	1	1	4	6	2	0	3	5	2	4	3	9	0	0	4	4	0	0	3	3	2	0	2	4
CO231	Kewat Rutik Sunil	1	2	2	5	0	3	4	7	2	2	0	4	0	4	0	4	2	3	2	7	2	4	2	8	1	1	0	2	0	1	3	4
CO232	Rananavare Shraddha Sudanrao	2	2	0	4	2	4	2	8	2	4	2	8	1	4	4	9	2	0	2	4	2	4	0	6	1	2	4	7	1	4	2	7
CO233	Adsul Yash Dnyaneshwar	0	3	3	6	0	4	1	5	0	3	0	3	2	2	3	7	2	3	0	5	2	0	2	4	2	3	4	9	1	3	1	5
CO234	Kadam Gaurav Vikas	2	2	2	8	2	0	2	4	0	4	1	5	0	2	1	3	2	0	4	6	0	0	4	4	0	4	4	8	2	0	2	4
CO235	Pisal Sakshi Ganesh	2	1	3	6	2	0	3	5	1	2	0	3	2	0	2	4	2	3	3	8	2	3	2	7	2	4	2	8	0	3	4	7
CO236	Bhandwalkar Rutik Santosh	1	0	4	5	1	3	1	5	2	1	3	6	1	0	2	3	0	1	2	5	1	1	0	2	0	2	3	5	2	0	2	4

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2021-22** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Jagtap T.V.**

Roll No.	Name of Candidate	EXP 1				EXP 2				EXP 3				EXP 4				EXP 5				EXP 6				EXP 7				EXP 8			
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
CO237	Suryawanshi Manjiri Hanumant	1	2	3	6	1	2	3	6	0	1	2	3	0	1	2	3	1	0	1	2	0	3	4	7	2	4	3	9	1	4	3	8
CO238	Adsul Prasad Pralhad	2	4	3	7	2	4	2	8	0	3	4	7	1	3	4	8	1	2	3	6	0	0	1	1	1	2	1	4	2	2	1	5
CO239	Thorat Snehal Santosh	0	2	3	5	0	4	3	7	1	0	1	2	2	0	1	3	1	4	0	5	2	2	3	7	0	0	1	1	0	0	1	1
CO240	Jamdade Vaibhav Arvind	2	4	2	8	2	1	3	6	1	2	3	6	0	2	3	5	1	1	2	4	2	4	0	6	2	2	3	7	2	2	3	7
CO241	Hajare Tushar Kisan	0	4	4	8	2	2	0	4	2	4	0	6	0	4	0	4	0	3	4	7	0	1	2	3	2	4	0	6	0	4	0	4
CO242	Gaikwad Revati Dipak	1	2	3	6	2	4	2	8	2	1	2	5	1	1	2	4	1	0	1	2	1	3	4	8	2	1	2	5	2	0	2	4
CO243	Navale Ishita Ravindra	0	3	2	5	0	1	3	4	8	3	4	7	0	3	4	7	1	2	3	6	0	0	1	1	2	3	4	9	2	1	3	6
CO244	Shilimkar Vaibhavi Sarjerao	2	1	3	6	0	3	1	4	1	0	1	2	1	0	1	2	0	4	0	4	2	2	3	7	0	0	1	1	2	4	0	6
CO245	Jagtap Pruthviraj Udayshing	0	4	2	6	1	3	2	6	2	2	3	7	0	2	3	5	2	1	2	5	0	4	0	4	1	2	3	6	1	1	2	4
CO246	Tamhane Shivani Ramesh	2	1	3	6	2	2	2	6	0	4	0	4	2	4	0	6	0	3	4	7	1	1	2	4	2	4	0	6	0	3	4	7
CO247	Prathamesh Sharad Hande	0	1	2	3	0	2	4	6	2	1	2	5	2	1	2	5	1	0	1	2	1	3	4	8	0	1	2	3	0	0	1	1
CO248	Howal Geeta Sanjay	1	3	2	6	2	4	4	8	1	3	4	8	2	3	4	9	2	2	3	7	1	0	1	2	0	3	4	7	1	2	3	6
CO249	Gadekar Shrutika Bhanudas	0	3	1	4	2	4	3	9	0	0	1	1	1	0	1	2	0	4	0	4	2	2	3	7	2	0	1	3	1	4	0	5
CO250	Sutar Rashika Ramesh	2	4	4	10	0	3	2	5	1	2	3	6	0	2	3	5	0	1	2	3	0	4	0	4	2	2	3	7	1	1	2	4
CO251	Chavan Apeksha Dilip	0	4	2	6	1	1	3	4	0	4	0	4	0	4	0	4	1	3	4	8	2	1	2	5	0	0	4	4	1	3	4	8
CO252	Pansare Vishal Sanjay	2	2	4	8	2	2	2	6	2	1	2	5	1	1	2	4	2	0	1	3	0	3	4	7	2	0	1	3	2	0	1	3
CO253	Kawale Sagar Dattatraya	1	0	3	4	0	2	4	6	0	3	4	7	2	3	4	9	1	2	3	6	1	0	1	2	0	2	3	5	1	2	3	6
CO254	Khomane Gaurav Vijay	2	2	4	8	2	0	2	4	0	0	1	1	3	0	1	3	2	4	0	6	0	2	3	5	1	4	0	6	1	4	0	5
CO255	Dhumal Snehal Navnath	2	2	3	7	2	0	2	4	2	2	3	7	0	2	3	5	1	1	2	4	0	4	0	4	1	1	2	4	1	1	2	4
CO256	Garud Renuka Pravin	1	3	2	6	1	3	2	6	2	4	0	6	0	4	0	4	2	3	4	9	1	1	2	4	1	3	4	8	1	3	4	8
CO257	Suryavanshi Vaishnav P.	1	3	0	4	0	3	1	4	2	1	2	5	2	1	2	5	1	0	1	2	2	3	4	9	1	0	1	2	2	0	1	3
CO258	Shinde Vishnukant Laxman	2	0	3	5	1	3	2	6	1	3	4	8	2	3	4	9	2	2	3	7	2	0	1	3	0	2	3	5	2	2	3	7
CO259	Tambe Aishwarya Sanjay	2	4	2	8	1	2	2	5	2	0	1	3	2	0	1	3	0	4	0	4	2	2	3	7	1	4	0	5	2	4	0	6
CO260	Jagdale Swapnil Somnath	2	4	0	6	2	3	0	5	0	2	3	5	1	2	3	6	0	1	2	3	2	4	0	6	0	1	2	3	1	1	2	4
CO261	Deshmukh Medhavi Suryakant	1	0	3	4	2	0	0	2	0	4	0	4	1	4	0	5	1	3	4	8	2	1	2	5	2	3	4	9	1	3	4	8
CO262	Punde Siddhi Ganesh	2	3	2	7	2	4	4	10	2	1	2	5	1	1	2	4	2	0	1	3	2	3	4	9	0	0	1	1	1	0	1	2
CO263	Omkar Anand Mandlik	0	1	2	3	2	4	2	8	2	3	4	9	0	3	4	7	0	2	3	5	1	0	1	2	2	2	3	7	0	2	3	5
CO264	Gadhave Sonali Pandurang	0	3	4	7	2	4	1	7	1	0	1	2	0	0	1	1	0	4	0	4	2	2	3	7	0	4	0	4	0	4	0	4
CO265	Baile Pranita Adhir	2	2	2	6	2	1	2	5	1	2	3	6	0	2	3	5	2	1	2	5	2	4	0	6	2	1	2	5	1	4	3	8
CO266	Pisal Vaishnavi Ajit	1	3	4	8	2	3	1	6	2	4	0	6	1	4	0	5	0	3	4	7	2	1	2	5	0	3	4	7	1	2	4	7
CO267	Gole Diksha Mohan	1	2	2	5	2	4	0	6	0	1	2	3	2	1	2	5	2	0	1	3	1	3	4	8	1	0	1	2	0	0	1	1
CO268	Kalyankar Omkar Amar	2	3	2	7	2	0	4	6	0	3	4	7	1	3	4	8	1	2	3	6	1	0	1	2	1	2	3	6	2	2	3	7
CO269	Kumbhar Rohit Anil	2	2	3	7	2	2	2	6	1	0	1	2	2	0	1	3	2	4	0	6	2	2	3	7	1	4	0	5	2	4	0	6
CO270	Tompe Anuja Prakash	2	0	4	6	2	2	2	6	2	2	3	7	1	2	3	6	2	1	2	5	2	4	0	6	2	1	2	5	2	1	2	5
CO271	Dange Samiksha Sanjay	2	1	3	6	2	2	3	7	0	4	0	4	2	4	0	6	2	3	4	9	1	1	2	4	2	3	4	9	2	3	4	9
CO272	Kamble Abhishek Sanjay	2	0	2	4	2	3	0	5	1	1	2	4	1	1	2	4	2	0	1	3	2	3	4	9	2	0	1	3	1	2	3	6

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF A2:T30ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering	**Academic Year: 2021-22**		**Semester: II**
Class: SE (Computer)	**Subject: Microprocessor**		**Subject In Charge: Prof. Ghadage S. S.**

Roll No.	Name of Candidate	EXP 9				EXP 10				EXP 11				EXP 12				Total	Out of 25
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D		
CO-201	ADSUL VAISHANAVI VIJAY	1	1	2	4	0	4	4	8	1	1	2	4	1	2	1	4	90	16
CO-202	ATOLE PRADIP SHIVAJI	2	3	4	9	1	4	4	9	0	3	4	7	2	3	4	9	91	17
CO-203	AVATE RUTUJA SUNIL	0	1	4	5	0	3	3	6	2	0	1	3	0	0	1	1	89	18
CO-204	BADGE YASHODEEP DYANESHWAR	1	3	2	6	1	3	2	6	2	2	3	7	2	2	3	7	84	15
CO-205	BHANDLAKAR POOJA BHAUSAHEB	1	0	1	2	1	1	1	3	1	4	0	5	1	4	2	7	93	18
CO-206	BHANDWALKAR SHREYASH SUNIL	1	2	3	6	1	1	2	4	1	1	2	4	2	1	3	6	100	20
CO-207	BHOSALE JAYASH MILIND	1	4	0	5	1	1	4	6	0	3	4	7	0	2	3	5	95	19
CO-208	BHOSALE KARAN JALINDAR	1	1	2	4	0	1	3	4	1	0	1	2	1	0	4	5	83	15
CO-209	BHOSALE SAMIR SHAMAKANT	1	3	4	8	2	1	2	5	2	2	3	7	1	1	2	4	94	19
CO-210	BHUJBAL RUTIKA MAHAVIR	1	0	1	2	2	3	2	7	0	4	0	4	2	3	4	9	94	19
CO-211	CHAVAN DIPTI RAMCHANDRA	1	2	3	6	0	1	2	3	2	1	2	5	0	0	3	3	95	19
CO-212	CHAVAN SAURABH DANAJI	1	4	0	5	2	3	4	9	2	3	4	9	2	2	1	5	94	19
CO-213	DHAVALE JAYASHRI ANIL	2	1	2	4	2	4	3	9	1	0	2	3	0	3	2	5	93	18
CO-214	DHOLE PRIYANKA SHIVAJI	2	3	4	9	2	2	1	5	2	3	4	9	2	4	3	9	100	20
CO-215	DHUMAL ATHARVE RAVINDRA	1	0	1	2	1	0	1	2	1	0	1	2	0	1	2	3	95	19
CO-216	GADADE ONKAR GORAKH	2	2	3	7	2	2	3	7	2	2	3	7	1	0	4	5	95	19
CO-217	GAIKWAD GAYATRI BALASO	0	4	0	4	2	4	0	6	2	4	1	7	2	3	2	7	94	19
CO-218	GAIKWAD SAYALI RAMESH	2	1	2	5	2	1	2	5	0	2	3	5	2	1	0	3	95	19
CO-219	GANGANNNA RAHUL SANJAY	2	3	4	9	2	3	4	9	1	4	0	5	1	4	3	8	100	20
CO-220	GHADGE VAIBHAVI BALASAHEB	2	0	1	3	1	4	3	8	2	1	2	5	0	2	1	3	100	20
CO-221	GHADGE VAISHNAVI BALASAHEB	1	2	3	6	1	2	1	4	2	3	4	9	1	6	1	2	90	16
CO-222	HALNOR KOMAL BALASAHEB	2	4	0	6	1	0	4	5	2	0	1	3	2	2	3	7	94	19
CO-223	INAMDAR ARBAJ TAYYAB	1	1	2	4	2	2	2	7	2	2	3	7	2	4	0	6	99	20
CO-224	JADHAV AVANTIKA RAHUL	2	3	4	9	0	1	0	1	1	4	0	5	0	1	2	3	99	20
CO-225	JADHAV SNEHA NITIN	1	0	1	2	0	1	2	3	2	1	2	5	1	3	4	8	93	18
CO-226	JAGDALE GANESH JALINDAR	2	2	3	7	2	3	4	9	1	3	4	8	2	0	1	3	93	18
CO-227	JAGTAP AKSHADA ANKUSH	2	4	0	6	2	0	1	3	2	0	1	3	0	2	3	5	93	18
CO-228	JOSHI ADITYA SANJAY	1	1	2	4	2	2	3	7	2	2	3	7	1	4	0	5	99	20
CO-229	KADAM BHUSHAN MALLIKARJUN	1	3	4	8	2	4	0	6	2	4	0	6	2	1	2	5	95	19
CO-230	KADAM NEELRAJ SATYAWAN	1	0	1	2	1	1	2	4	2	1	2	5	0	3	4	7	95	19
CO-231	KADAM SAKSHI RAJENDRA	0	2	3	5	2	2	4	8	1	3	4	8	1	0	1	2	94	19
CO-232	KAKADE SAURABH RUSHIKANT	2	4	0	6	1	0	1	2	1	0	1	2	1	2	3	6	99	20
CO-233	KAKADE SHIVANJALI MOHAN	2	1	2	5	2	2	3	7	1	2	3	6	2	4	0	6	95	19
CO-234	KALASKAR GANESH DNYANESHWAR	1	3	4	8	0	4	0	4	1	4	0	5	0	1	2	3	94	19
CO-235	KARNWAR RUTUJA BHAUSAHEB	2	0	1	3	2	1	2	5	0	1	2	3	1	3	4	8	99	20
CO-236	KHANDALE ASHWINI ANIL	0	2	3	5	1	3	4	8	2	3	4	9	2	0	2	4	89	16

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

SSPM's SHARADCHANDRA PAWAR COLLEGE OF A2:T30ENGINEERING & TECHNOLOGY

SOMESHWAR

ASSESMENT RECORD

Department:Computer Engineering | **Academic Year: 2021-22** | **Semester: II**

Class: SE (Computer) | **Subject: Microprocessor** | **Subject In Charge: Prof. Ghadage S. S.**

Roll No.	Name of Candidate	EXP 9				EXP 10				EXP 11				EXP 12				Total	Out of 25
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D		
CO-237	KINGARE PAYAL DATTATRAY	1	2	4	7	2	2	2	6	2	2	1	5	2	4	4	10	100	20
CO-238	KUTAL PRATIKSHA RAJENDRA	1	1	4	6	1	1	3	5	1	3	3	7	1	2	3	6	91	17
CO-239	LAMBOTE MAYA TANAJI	2	2	4	8	2	2	3	7	0	2	4	6	2	1	4	7	89	16
CO-240	MAKWANA YASH RANJIT	1	3	3	7	1	3	4	8	2	2	4	8	1	2	4	7	94	19
CO-241	MANE ROHAN SANJAY	2	4	4	10	2	4	3	9	1	2	3	6	2	3	3	8	99	20
CO-242	MANE SAURABH SANJAY	2	1	2	5	1	4	0	5	0	1	2	3	1	1	2	4	93	18
CO-243	MARDANE JAY PRAVIN	2	3	4	9	2	1	2	5	0	3	4	7	2	3	4	9	91	17
CO-244	MOHITE YASHRAJ SHRIKANT	1	0	1	2	0	3	3	6	1	0	1	2	2	1	2	5	99	20
CO-245	MUNDHE AKSHAY HARIPANDIT	1	2	3	6	0	0	1	1	0	2	3	5	2	3	4	9	95	19
CO-246	NAGAWADE SAURAV SUBARAV	1	4	0	5	1	2	3	6	2	4	0	6	1	1	2	4	93	18
CO-247	NAIKWADE ARTI HANMANT	2	1	2	5	2	4	0	6	2	1	2	5	0	3	4	7	95	19
CO-248	NARUTE KIRAN LALASO	1	3	4	8	0	1	2	3	0	3	4	7	1	1	2	4	92	18
CO-249	NIGADE RAHUL DYANESHWAR	2	0	1	3	1	3	4	8	1	0	1	2	1	3	4	8	100	20
CO-250	PAPAL KOMAL MARUTI	2	2	3	7	2	0	1	3	2	2	3	7	1	1	2	4	95	19
CO-251	PATIL PRAJAKTA VIKAS	0	4	4	8	2	2	3	7	2	4	0	6	1	3	4	8	83	15
CO-252	PAWAR ABHISHEK MADHAV	1	3	2	6	2	4	0	6	0	1	2	3	1	1	2	4	99	20
CO-253	PAWAR AYUSHKA BHANUDAS	1	1	0	2	1	1	2	4	1	3	4	8	2	3	4	9	93	19
CO-254	PAWAR ROHIT BALASO	1	3	4	8	0	3	4	7	2	0	1	3	1	1	2	4	100	20
CO-255	PAWAR TEJAS PRADIP	2	2	1	5	2	0	1	3	0	2	3	5	2	3	4	9	100	20
CO-256	RANAWARE ADITYA DIPAK	2	0	0	2	2	2	3	7	1	4	0	5	2	1	2	5	94	19
CO-257	RASKAR MANJUSHA SAVATA	0	2	4	6	2	4	0	6	0	1	2	3	2	3	4	9	91	18
CO-258	RAUT ADITI SHIVAJI	2	3	1	6	2	1	2	5	2	3	4	9	1	1	2	4	100	20
CO-259	RAUT SARTHAK SAMDAS	0	0	1	1	2	3	4	9	0	0	1	1	2	3	4	9	91	17
CO-260	ROKADE DYANESHWARI MOHAN	0	2	3	5	2	0	1	3	2	2	3	7	2	1	2	5	86	15
CO-261	SALUNKE NAMRATA CHAKRADHAR	2	4	2	8	2	2	3	7	2	4	0	6	2	3	4	9	93	18
CO-262	SALUNKE SWARAJ NITIN	2	4	3	9	0	4	0	4	2	1	2	5	1	0	1	2	99	20
CO-263	SALUNKHE SAKSHI RAJENDRA	1	4	2	7	1	1	2	4	1	3	4	8	1	2	3	6	95	19
CO-264	SALUNKHE VIVEK RAJESH	0	1	3	4	2	3	4	9	1	0	1	2	1	4	0	5	99	20
CO-265	SHINDE ISHA SURESH	1	0	1	2	0	0	1	1	1	3	3	6	2	1	2	5	83	15
CO-266	SHINDE SANDHYA NANDKUMAR	1	2	3	6	1	2	3	6	1	4	0	5	0	3	4	7	99	20
CO-267	SONAWANE ABHAY JAGDISH	1	4	0	5	0	4	0	4	1	1	2	4	2	1	2	5	93	19
CO-268	TAKAWALE VAISHNAVI VITTHAL	2	1	2	5	1	1	2	4	1	3	4	8	0	3	4	7	84	15
CO-269	TANTAK ADITI MADHUKAR	1	3	4	8	2	3	4	9	0	0	1	1	2	0	1	2	85	15
CO-270	TAWARE SAKSHI SANDIP	2	0	1	3	1	0	1	2	2	2	3	7	0	2	3	5	95	19
CO-271	THAKRE RAJ NARESH	0	2	3	5	1	2	3	6	2	4	0	6	1	4	1	6	85	15
CO-272	YADAV POOJA BHIMRAO	2	4	0	6	1	4	0	5	0	1	2	3	1	4	4	9	100	20

A : Regular attendance = 02 marks, B- Timely completion = 04 marks C- Neat and clean writing = 04 marks D-Total

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

STUDENT ATTENDANCE RECORD

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Class: SE-Practical Attendance(B1) Subject teacher: Ghadage S.S. Subject: MPL

Roll no	STUDENT NAME	21/2	28/2	7/3	21/3	28/3	11/4	18/4	25/4	25/4	9/5	16/5
CO-201	ADSUL VAISHANAVI VIJAY	P	P	P	A	P	P	A	P	P	P	A
CO-202	ATOLE PRADIP SHIVAJI	P	P	P	A	P	P	P	P	A	P	P
CO-203	AVATE RUTUJA SUNIL	P	A	P	P	A	A	P	P	P	A	P
CO-204	BADGE YASHODEEP DYANESHWAR	P	P	P	P	P	P	A	P	A	P	P
CO-205	BHANDLAKAR POOJA BHAUSAHEB	P	P	A	P	A	P	P	A	·	P	P
CO-206	BHANDWALKAR SHREYASH SUNIL	P	P	A	P	A	P	P	P	P	P	P
CO-207	BHOSALE JAYASH MILIND	P	P	P	P	P	A	P	P	P	A	P
CO-208	BHOSALE KARAN JALINDAR	P	P	P	P	P	P	A	P	P	P	P
CO-209	BHOSALE SAMIR SHAMAKANT	P	P	A	P	P	P	P	A	P	P	P
CO-210	BHUJBAL RUTIKA MAHAVIR	P	A	A	P	P	A	P	P	P	A	P
CO-211	CHAVAN DIPTI RAMCHANDRA	P	A	A	P	A	P	P	P	P	P	P
CO-212	CHAVAN SAURABH DANAJI	A	A	A	P	A	P	P	P	P	A	P
CO-213	DHAVALE JAYASHRI ANIL	P	P	P	A	P	P	P	P	P	P	A
CO-214	DHOLE PRIYANKA SHIVAJI	P	P	P	P	A	P	P	A	P	P	P
CO-215	DHUMAL ATHARVE RAVINDRA	P	P	A	P	A	P	P	P	P	A	P
CO-216	GADADE ONKAR GORAKH	A	P	P	A	P	P	A	P	A	P	P
CO-217	GAIKWAD GAYATRI BALASO	P	P	A	P	P	A	P	P	P	P	P
CO-218	GAIKWAD SAYALI RAMESH	P	P	P	A	P	P	P	P	P	P	A

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Class: SE-Practical Attendance(B2) Subject teacher: Ghadage S.S Subject: MPL

Roll no	STUDENT NAME	24/2	3/3	10/3	24/3	31/3	21/4	28/4	5/5	5/5	12/5	19/5
CO-219	GANGANNNA RAHUL SANJAY	P	P	P	A	P	P	A	P	P	P	A
CO-220	GHADGE VAIBHAVI BALASAHEB	P	P	A	A	P	P	P	P	A	P	P
CO-221	GHADGE VAISHNAVI BALASAHEB	P	P	P	A	P	A	P	P	P	A	P
CO-222	HALNOR KOMAL BALASAHEB	P	A	P	P	P	P	P	A	P	P	P
CO-223	INAMDAR ARBAJ TAYYAB	P	P	A	P	P	P	P	P	P	P	A
CO-224	JADHAV AVANTIKA RAHUL	P	P	P	P	P	A	A	P	P	P	P
CO-225	JADHAV SNEHA NITIN	A	P	P	P	P	A	P	P	P	A	P
CO-226	JAGDALE GANESH JALINDAR	P	P	P	P	P	P	P	P	P	P	P
CO-227	JAGTAP AKSHADA ANKUSH	P	P	P	P	A	P	P	P	A	P	P
CO-228	JOSHI ADITYA SANJAY	P	P	P	P	A	P	P	P	P	P	P
CO-229	KADAM BHUSHAN MALLIKARJUN	P	P	A	P	P	P	P	A	P	P	P
CO-230	KADAM NEELRAJ SATYAWAN	P	P	P	P	A	P	P	P	P	P	P
CO-231	KADAM SAKSHI RAJENDRA	P	A	A	P	P	P	P	P	P	A	P
CO-232	KAKADE SAURABH RUSHIKANT	P	P	P	A	A	P	P	P	P	A	P
CO-233	KAKADE SHIVANJALI MOHAN	P	A	P	P	P	P	P	P	P	P	P
CO-234	KALASKAR GANESH DNYANESHWAR	P	P	P	P	A	P	P	A	P	P	P
CO-235	KARNWAR RUTUJA BHAUSAHEB	P	P	P	P	P	P	P	P	P	P	P
CO-236	KHANDALE ASHWINI ANIL	P	A	P	P	P	P	P	P	A	P	P

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Class: SE-Practical Attendance(B3) Subject teacher: Ghadage S.S. Subject: MPL

Roll no	STUDENT NAME	28/2	1/03	8/3	8/3	8/3	12/4	26/4	3/5	5/5	10/5	17/5												
CO-237	KINGARE PAYAL DATTATRAY	A	P	P	P	P	P	A	P	P	P	P												
CO-238	KUTAL PRATIKSHA RAJENDRA	P	P	P	P	A	P	P	A	P	P	P												
CO-239	LAMBOTE MAYA TANAJI	P	A	P	P	P	P	P	A	P	A	P												
CO-240	MAKWANA YASH RANJIT	P	P	P	P	P	A	A	P	P	A	P												
CO-241	MANE ROHAN SANJAY	P	P	A	A	P	P	P	P	P	P	A												
CO-242	MANE SAURABH SANJAY	P	P	P	A	A	P	P	P	P	P	P												
CO-243	MARDANE JAY PRAVIN	P	P	P	A	P	P	A	P	A	P	P												
CO-244	MOHITE YASHRAJ SHRIKANT	P	P	P	P	P	A	P	P	P	P	A												
CO-245	MUNDHE AKSHAY HARIPANDIT	P	P	P	P	A	A	P	P	P	P	P												
CO-246	NAGAWADE SAURAV SUBARAV	P	A	P	P	A	P	P	P	P	A	P												
CO-247	NAIKWADE ARTI HANMANT	P	P	A	P	P	P	A	P	P	P	P												
CO-248	NARUTE KIRAN LALASO	P	A	P	A	P	P	P	P	P	A	P												
CO-249	NIGADE RAHUL DYANESHWAR	P	P	P	A	P	P	P	A	P	P	P												
CO-250	PAPAL KOMAL MARUTI	P	P	A	P	P	P	A	P	P	P	A												
CO-251	PATIL PRAJAKTA VIKAS	P	P	A	P	P	P	P	P	P	P	P												
CO-252	PAWAR ABHISHEK MADHAV	P	A	P	P	A	P	P	P	A	P	P												
CO-253	PAWAR AYUSHKA BHANUDAS	P	A	A	P	P	P	P	P	P	P	P												
CO-254	PAWAR ROHIT BALASO	P	P	P	P	A	P	A	P	P	P	P												

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Class: SE-Practical Attendance(B4) Subject teacher: Ghadage S.S. Subject: MPL

Roll no	STUDENT NAME	8/2	9/2	9/3	8/3	13/3	20/3	27/3	4/5	4/5	11/5	18/5												
CO-255	PAWAR TEJAS PRADIP	P	A	A	P	P	P	P	P	P	A	P												
CO-256	RANAWARE ADITYA DIPAK	P	P	P		P	P	P	P	A	P	P												
CO-257	RASKAR MANJUSHA SAVATA	A	A	P	P	P	P	P	A	P	P	P												
CO-258	RAUT ADITI SHIVAJI	P	P	A	P	P	A	P	P	P	P	A												
CO-259	RAUT SARTHAK SAMDAS	P	P	P	A	P	P	P	P	P	P	A												
CO-260	ROKADE DYANESHWARI MOHAN	P	P	P	P	P	P	P	P	P	A	P												
CO-261	SALUNKE NAMRATA CHAKRADHAR	P	A	P	P	P	P	P	P	P	A	P												
CO-262	SALUNKE SWARAJ NITIN	P	P	P	A	A	P	P	P	P	P	P												
CO-263	SALUNKHE SAKSHI RAJENDRA	P	P	P	P	P	P	P	P	A	P	P												
CO-264	SALUNKHE VIVEK RAJESH	P	A	P	P	A	P	P	P	P	P	P												
CO-265	SHINDE ISHA SURESH	P	P	P	A	P	P	P	A	P	P	P												
CO-266	SHINDE SANDHYA NANDKUMAR	P	P	P	P	P	P	P	A	P	P	P												
CO-267	SONAWANE ABHAY JAGDISH	P	P	A	P	P	P	P	P	A	P	P												
CO-268	TAKAWALE VAISHNAVI VITTHAL	P	P	P	A	P	P	P	A	P	P	P												
CO-269	TANTAK ADITI MADHUKAR	P	A	P	P	A	P	P	P	P	P	P												
CO-270	TAWARE SAKSHI SANDIP	P	P	P	P	P	P	P	P	A	P	P												
CO-271	THAKRE RAJ NARESH	P	P	A	P	P	P	P	A	P	P	P												
CO-272	YADAV POOJA BHIMRAO	P	P	P	P	P	P	P	P	A	P	P												

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Lecture Attendance Sheet

Class: SE Subject teacher: Prof. Ghadage S.S. Subject: Microprocessor

Roll no	STUDENT NAME	9/2	10/2	20/2	23/2	24/2	27/2	2/3	3/3	6/3	9/3	10/3	13/3	16/3	17/3	20/3	23/3	24/3	27/3	30/3	31/3	13/4	13/4	17/4	17/4
CO-201	ADSUL VAISHANAVI VIJAY	A	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
CO-202	ATOLE PRADIP SHIVAJI	P	P	A	P	A	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
CO-203	AVATE RUTUJA SUNIL	A	P	P	A	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
CO-204	BADGE YASHODEEP DYANESHWAR	P	P	P	P	A	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
CO-205	BHANDLAKAR POOJA BHAUSAHEB	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P
CO-206	BHANDWALKAR SHREYASH SUNIL	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	A	P	P	P	P	P
CO-207	BHOSALE JAYASH MILIND	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
CO-208	BHOSALE KARAN JALINDAR	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P
CO-209	BHOSALE SAMIR SHAMAKANT	P	P	A	P	P	P	P	P	P	P	A	P	A	P	P	P	P	P	P	P	P	P	P	P
CO-210	BHUJBAL RUTIKA MAHAVIR	P	A	P	P	A	P	P	P	A	P	P	P	P	P	A	P	P	A	P	P	P	P	P	P
CO-211	CHAVAN DIPTI RAMCHANDRA	P	P	P	A	P	P	P	P	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P	P
CO-212	CHAVAN SAURABH DANAJI	P	P	P	A	P	P	P	P	P	P	P	P	A	P	A	P	A	P	P	P	P	P	P	P
CO-213	DHAVALE JAYASHRI ANIL	P	P	P	A	P	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P	P
CO-214	DHOLE PRIYANKA SHIVAJI	P	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P	A	P	P	P	P	P	P
CO-215	DHUMAL ATHARVE RAVINDRA	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P	A	P	A	P	P	P	P	P	P
CO-216	GADADE ONKAR GORAKH	P	P	P	P	P	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P	P	P
CO-217	GAIKWAD GAYATRI BALASO	P	P	P	A	P	P	P	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	P	P
CO-218	GAIKWAD SAYALI RAMESH	P	P	P	P	A	P	P	A	P	P	A	P	P	P	A	P	P	P	P	P	P	P	P	P
CO-219	GANGANNNA RAHUL SANJAY	P	A	P	A	P	P	P	P	P	P	P	P	A	P	P	A	P	P	A	P	P	P	P	P
CO-220	GHADGE VAIBHAVI BALASAHEB	P	P	A	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P	P	P	P
CO-221	GHADGE VAISHNAVI BALASAHEB	A	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P
CO-222	HALNOR KOMAL BALASAHEB	P	A	P	A	P	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
CO-223	INAMDAR ARBAJ TAYYAB	P	P	A	P	P	A	P	P	P	P	P	A	P	P	A	P	P	P	P	P	P	P	P	P
CO-224	JADHAV AVANTIKA RAHUL	P	P	P	A	P	P	P	A	P	P	A	P	P	A	P	A	P	P	P	A	P	P	P	P
CO-225	JADHAV SNEHA NITIN	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	A	P	A	P	P	P	P	P
CO-226	JAGDALE GANESH JALINDAR	P	P	A	P	P	A	P	A	P	P	A	P	P	A	P	P	P	A	P	A	P	P	P	P
CO-227	JAGTAP AKSHADA ANKUSH	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	A	P	P	P	P	P	P	P	P
CO-228	JOSHI ADITYA SANJAY	P	P	A	P	P	P	P	P	P	P	P	P	P	A	P	P	A	P	P	A	P	P	P	P
CO-229	KADAM BHUSHAN MALLIKARJUN	P	P	P	A	P	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P	P	P
CO-230	KADAM NEELRAJ SATYAWAN	P	P	P	P	A	P	A	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P
CO-231	KADAM SAKSHI RAJENDRA	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P
CO-232	KAKADE SAURABH RUSHIKANT	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P	P
CO-233	KAKADE SHIVANJALI MOHAN	P	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P	P	P	P
CO-234	KALASKAR GANESH DNYANESHWAR	P	P	A	P	P	P	P	P	A	P	P	A	P	P	A	P	P	P	P	A	P	P	P	P
CO-235	KARNWAR RUTUJA BHAUSAHEB	P	P	P	P	A	P	A	P	P	P	A	P	P	P	P	A	P	P	A	P	P	P	P	P

CO-236	KHANDALE ASHWINI ANIL	P	A	P	A	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P		
CO-237	KINGARE PAYAL DATTATRAY	P	P	P	P	P	A	A	P	P	A	P	P	A	P	P	P	A	P	P	A	P	P	P	P		
CO-238	KUTAL PRATIKSHA RAJENDRA	P	P	A	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P		
CO-239	LAMBOTE MAYA TANAJI	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	P	A	P	P	A	P	P	P	P		
CO-240	MAKWANA YASH RANJIT	P	P	P	P	P	P	A	P	P	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P		
CO-241	MANE ROHAN SANJAY	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	A	P	P	P	P	P		
CO-242	MANE SAURABH SANJAY	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P		
CO-243	MARDANE JAY PRAVIN	P	P	A	P	P	P	P	A	P	P	A	P	P	P	A	P	A	P	A	P	P	P	P	P		
CO-244	MOHITE YASHRAJ SHRIKANT	P	P	P	A	P	P	P	P	A	P	P	A	A	P	P	P	A	P	P	A	P	P	P	P		
CO-245	MUNDHE AKSHAY HARIPANDIT	P	P	A	P	P	P	A	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	P	P		
CO-246	NAGAWADE SAURAV SUBARAV	P	P	P	P	P	P	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	P	P	P		
CO-247	NAIKWADE ARTI HANMANT	P	P	P	P	P	P	P	P	P	A	P	P	P	P	A	P	P	P	P	A	P	P	P	P		
CO-248	NARUTE KIRAN LALASO	P	A	A	P	P	P	P	A	P	P	P	A	P	P	P	P	A	P	P	P	P	P	P	P		
CO-249	NIGADE RAHUL DYANESHWAR	P	P	P	P	P	A	P	P	A	P	A	P	P	P	P	A	P	P	P	P	A	P	P	P		
CO-250	PAPAL KOMAL MARUTI	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	P	A	P	P	A	P	P	P	P		
CO-251	PATIL PRAJAKTA VIKAS	A	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	A	P	P	P		
CO-252	PAWAR ABHISHEK MADHAV	P	P	P	P	P	A	P	P	A	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P		
CO-253	PAWAR AYUSHKA BHANUDAS	A	P	P	P	A	P	A	P	P	P	A	P	P	P	P	P	A	P	P	P	A	P	P	P		
CO-254	PAWAR ROHIT BALASO	A	P	P	P	A	P	P	A	P	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P		
CO-255	PAWAR TEJAS PRADIP	P	A	P	P	P	P	A	P	P	P	A	P	P	A	P	P	A	P	A	P	A	P	P	P		
CO-256	RANAWARE ADITYA DIPAK	P	P	P	A	P	P	P	A	P	P	P	P	A	P	P	A	P	P	P	P	P	P	P	P		
CO-257	RASKAR MANJUSHA SAVATA	P	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	P	P	A	P	P	P	P	P		
CO-258	RAUT ADITI SHIVAJI	P	P	A	P	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P		
CO-259	RAUT SARTHAK SAMDAS	P	A	P	P	P	P	P	P	A	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P		
CO-260	ROKADE DYANESHWARI MOHAN	P	P	P	A	A	P	P	P	P	P	P	A	P	P	P	P	A	P	P	P	P	P	P	P		
CO-261	SALUNKE NAMRATA CHAKRADHAR	P	P	A	A	P	P	A	P	P	P	P	A	P	P	P	P	A	P	P	P	P	P	P	P		
CO-262	SALUNKE SWARAJ NITIN	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	P		
CO-263	SALUNKHE SAKSHI RAJENDRA	P	P	P	P	P	P	P	A	P	P	P	A	P	P	A	P	P	P	P	A	P	P	P	P		
CO-264	SALUNKHE VIVEK RAJESH	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	A	P	P	P	P		
CO-265	SHINDE ISHA SURESH	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	A	P	A	P	A	P	P	P	P		
CO-266	SHINDE SANDHYA NANDKUMAR	P	P	P	P	P	P	A	P	P	P	A	P	P	P	P	A	P	P	P	A	P	P	P	P		
CO-267	SONAWANE ABHAY JAGDISH	P	P	P	P	P	P	P	A	P	P	P	A	P	P	A	P	P	A	P	P	P	P	P	P		
CO-268	TAKAWALE VAISHNAVI VITTHAL	P	P	A	P	P	P	P	P	A	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P		
CO-269	TANTAK ADITI MADHUKAR	P	P	P	A	P	P	P	P	P	A	P	P	P	A	P	P	P	P	P	P	P	P	P	P		
CO-270	TAWARE SAKSHI SANDIP	P	A	P	A	P	P	P	P	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P	P		
CO-271	THAKRE RAJ NARESH	P	A	P	A	P	P	P	P	P	P	A	P	P	A	P	P	A	P	P	A	P	P	P	P		
CO-272	YADAV POOJA BHIMRAO	P	P	A	P	P	P	P	A	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P	P		

Sharadchandra Pawar College of Engineering and Technology,Someshwarnagar.

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year:2022-2023

Lecture Attendance Sheet

Class: SE **Subject teacher:** Microprocessor **Subject:** Prof. Ghadage S.S.

Roll no	STUDENT NAME	20/1	20/1	21/1	21/1	24/1	25/1	27/1	28/1	1/2	1/2	3/2	8/2	11/2	11/2	12/2	15/2	18/2	18/2	19/2					
CO-201	ADSUL VAISHANAVI VIJAY	P	P	P	P	P	P	A	P	P	P	A	P	P	P	A	P	P	P	P					
CO-202	ATOLE PRADIP SHIVAJI	P	A	A	A	P	P	P	P	P	.	.	P	P	P	.	P	P	P	.					
CO-203	AVATE RUTUJA SUNIL	A	P	A	A	A	P	P	P	P	P	P	P	.	P	P	P	.	P	P					
CO-204	BADGE YASHODEEP DYANESHWAR	P	P	P	P	P	A	P	A	P	P	.	P	P	P	P	P	P	.	P					
CO-205	BHANDLAKAR POOJA BHAUSAHEB	P	P	P	P	A	P	A	P	P	P	.	P	P	P	.	P	P	P	P					
CO-206	BHANDWALKAR SHREYASH SUNIL	P	A	A	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P					
CO-207	BHOSALE JAYASH MILIND	P	P	A	P	P	P	P	P	P	P	P	P	P	.	P	P	.	P	P					
CO-208	BHOSALE KARAN JALINDAR	P	P	P	P	P	A	A	P	P	.	P	P	P	P	.	P	P	P	P					
CO-209	BHOSALE SAMIR SHAMAKANT	P	P	P	A	P	A	A	P	P	.	P	P	P	P	P	P	P	P	P					
CO-210	BHUJBAL RUTIKA MAHAVIR	P	P	P	P	P	P	A	P	P	P	.	P	P	P	P	P	P	P	.					
CO-211	CHAVAN DIPTI RAMCHANDRA	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P	.	P	P					
CO-212	CHAVAN SAURABH DANAJI	A	P	A	P	P	P	P	A	P	.	P	P	P	P	P	P	.	P	P					
CO-213	DHAVALE JAYASHRI ANIL	P	P	A	P	A	A	A	P	.	P	P	P	P	P	P	P	P	P	P					
CO-214	DHOLE PRIYANKA SHIVAJI	P	A	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P					
CO-215	DHUMAL ATHARVE RAVINDRA	P	P	A	A	P	A	A	P	P	.	P	P	P	P	P	.	P	P	P					
CO-216	GADADE ONKAR GORAKH	P	P	P	A	A	P	A	A	P	P	.	P	P	P	P	P	.	P	P					
CO-217	GAIKWAD GAYATRI BALASO	P	P	A	A	A	A	A	P	P	P	P	.	P	P	P	P	P	.	P					
CO-218	GAIKWAD SAYALI RAMESH	P	P	P	P	A	P	P	P	P	P	P	.	P	P	P	P	P	P	.					
CO-219	GANGANNNA RAHUL SANJAY	A	P	A	P	P	P	P	A	P	P	.	P	P	P	P	P	P	P	P					
CO-220	GHADGE VAIBHAVI BALASAHEB	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P	P	.	P	P					
CO-221	GHADGE VAISHNAVI BALASAHEB	A	P	P	P	P	A	P	P	P	P	P	P	P	P	.	P	P	P	P					
CO-222	HALNOR KOMAL BALASAHEB	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P	P	P					
CO-223	INAMDAR ARBAJ TAYYAB	P	A	P	P	P	P	A	P	.	P	P	P	P	P	.	P	P	P	P					
CO-224	JADHAV AVANTIKA RAHUL	P	P	P	P	A	P	P	P	P	P	.	P	P	P	P	P	P	P	P					
CO-225	JADHAV SNEHA NITIN	P	P	P	A	P	P	P	A	P	P	P	P	.	.	P	P	.	P	.					
CO-226	JAGDALE GANESH JALINDAR	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P					
CO-227	JAGTAP AKSHADA ANKUSH	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P	P	P	P	P					
CO-228	JOSHI ADITYA SANJAY	A	P	A	P	P	P	P	P	P	P	P	P	P	.	P	P	P	.	P					
CO-229	KADAM BHUSHAN MALLIKARJUN	P	P	P	P	P	A	P	P	P	P	.	P	P	P	P	P	P	P	P					
CO-230	KADAM NEELRAJ SATYAWAN	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P	P	P	P					
CO-231	KADAM SAKSHI RAJENDRA	P	P	P	A	A	P	P	P	P	.	P	P	P	P	P	P	P	P	.					
CO-232	KAKADE SAURABH RUSHIKANT	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P	P	P	.	P					
CO-233	KAKADE SHIVANJALI MOHAN	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P	.	P	P	P					
CO-234	KALASKAR GANESH DNYANESHWAR	P	P	P	P	A	P	P	A	P	P	P	P	P	P	P	P	P	.	P					
CO-235	KARNWAR RUTUJA BHAUSAHEB	P	A	P	P	P	P	P	P	P	.	P	P	P	P	P	P	P	P	P					

CO-236	KHANDALE ASHWINI ANIL	P	P	P	A	A	A	P	P	P	P	.	P	P	P	P	P	P	.	P	P					
CO-237	KINGARE PAYAL DATTATRAY	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-238	KUTAL PRATIKSHA RAJENDRA	P	P	P	P	A	A	P	P	.	P	P	P	P	P	P	P	P	.	P	P					
CO-239	LAMBOTE MAYA TANAJI	P	A	A	A	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P					
CO-240	MAKWANA YASH RANJIT	P	P	P	A	P	P	P	P	P	P	.	P	P	P	P	P	P	P	P	.					
CO-241	MANE ROHAN SANJAY	P	P	P	P	P	P	A	A	P	.	P	P	P	P	P	P	.	P	P	P					
CO-242	MANE SAURABH SANJAY	P	P	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P	.	P	P					
CO-243	MARDANE JAY PRAVIN	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-244	MOHITE YASHRAJ SHRIKANT	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-245	MUNDHE AKSHAY HARIPANDIT	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-246	NAGAWADE SAURAV SUBARAV	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P	P	P	.	P	P					
CO-247	NAIKWADE ARTI HANMANT	A	P	A	A	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-248	NARUTE KIRAN LALASO	P	P	P	P	A	A	P	P	.	P	P	P	P	P	P	P	P	.	P	P					
CO-249	NIGADE RAHUL DYANESHWAR	P	P	P	P	P	A	A	A	P	P	.	P	P	P	P	P	P	.	P	P					
CO-250	PAPAL KOMAL MARUTI	P	A	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-251	PATIL PRAJAKTA VIKAS	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-252	PAWAR ABHISHEK MADHAV	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-253	PAWAR AYUSHKA BHANUDAS	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-254	PAWAR ROHIT BALASO	P	P	P	A	A	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-255	PAWAR TEJAS PRADIP	P	P	P	A	A	A	P	P	P	P	.	P	P	P	P	P	P	.	P	P					
CO-256	RANAWARE ADITYA DIPAK	A	A	A	P	P	A	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-257	RASKAR MANJUSHA SAVATA	P	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P	P	.	P	P					
CO-258	RAUT ADITI SHIVAJI	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-259	RAUT SARTHAK SAMDAS	P	P	P	P	P	P	P	A	.	P	P	P	P	P	P	P	P	.	P	P					
CO-260	ROKADE DYANESHWARI MOHAN	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-261	SALUNKE NAMRATA CHAKRADHAR	A	A	A	P	P	P	P	P	P	.	P	P	P	P	P	P	P	.	P	P					
CO-262	SALUNKE SWARAJ NITIN	P	P	P	P	A	A	A	P	P	P	.	P	P	P	P	P	P	.	P	P					
CO-263	SALUNKHE SAKSHI RAJENDRA	P	A	P	P	P	A	P	P	P	.	P	P	P	P	P	P	P	.	P	P					
CO-264	SALUNKHE VIVEK RAJESH	P	P	P	P	P	A	A	P	P	P	.	P	P	P	P	P	P	.	P	P					
CO-265	SHINDE ISHA SURESH	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	.	P	P					
CO-266	SHINDE SANDHYA NANDKUMAR	P	P	P	P	P	A	A	A	P	P	P	.	P	.	P	P	P	P	P	P					
CO-267	SONAWANE ABHAY JAGDISH	P	P	P	P	P	P	P	P	.	P	P	P	.	P	P	P	P	.	P	P					
CO-268	TAKAWALE VAISHNAVI VITTHAL	P	P	P	P	P	A	A	A	P	.	P	.	P	P	P	P	P	.	P	P					
CO-269	TANTAK ADITI MADHUKAR	P	P	P	P	P	A	A	A	P	P	P	.	P	.	P	P	P	.	P	P					
CO-270	TAWARE SAKSHI SANDIP	P	A	P	P	P	A	P	P	P	P	P	.	P	P	P	P	P	.	P	P					
CO-271	THAKRE RAJ NARESH	P	P	P	P	P	A	A	A	.	P	P	P	P	P	P	P	P	.	P	P					
CO-272	YADAV POOJA BHIMRAO	P	P	A	P	A	A	A	P	P	P	P	P	P	P	.	.	P	P	P	P					

Shri Someshwar Shikshan Prasarak Mandal's

Sharadchandra Pawar College of Engineering and Technology, SomeshwarNagar

Department of Computer Engineering

SUBJECT NOTES

Exceptions and Interrupts

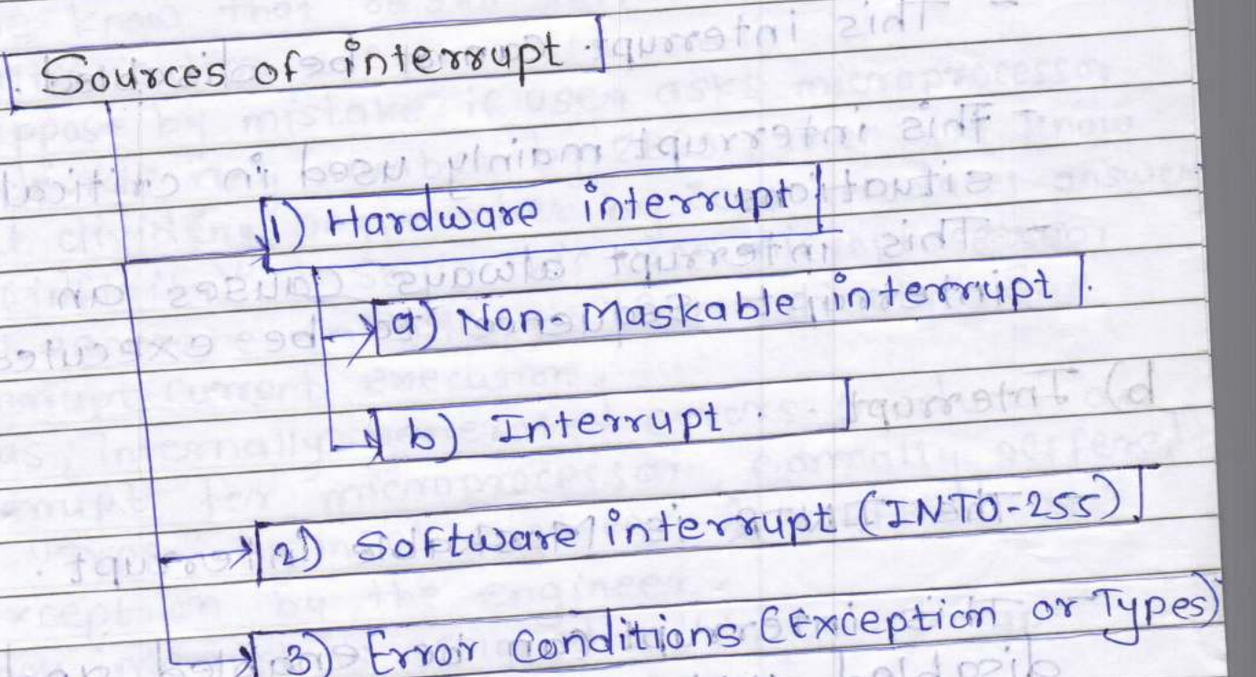
Page No. 1
Date

* Identifying Interrupts.

Q. List different Sources of interrupts. [Dec 3M]

- When a device requests, the microprocessor suspends the execution of the current program & gives survice to the I/O device. This feature is called as interrupt.

- Interrupt Sources
  - In 80386, we have three sources of interrupt.



-1) Hardware interrupt.
- In this type of interrupt, physical pins are provided in the chip.
- This interrupts are generated by changing the logic levels on the 80386's interrupt pins.

- When considering the precedence of interrupts for multiple simultaneous interruppts, the following guidlines apply:

1) INTR is the only maskable interrupt & if detected simultaneously with other interrupts, resetting of IF by the other interrupts will mask INTR. These causes the INTR to be the lowest priority interrupt serviced after all other interrupts unless the other interrupts service routine re-enable interrupts.

2) of non-maskable interrupts (NMI, single step & software generated i.e. INT n, INTO), in general, single step has lowest priority (will be serviced last). INT n has the highest priority & will be serviced first, followed by NMI, followed by single step.

* Enabling & Disabling interrupts.

Q.) Explain different way by which 80386 can enable & Disable interrupts.

→ The 80386 microprocessor services interrupts & exceptions after the completion of an instructn & before the execution of the next instruction begins. In case of string instructions, the interrupts & exceptions can be between repetitions. At the instruction boundaries some flag settings or conditions can enable/disable interrupts & exceptions.

Interrupt	Faults	Trap	Abort
the interrupt occurred.	If detected during the instruction, the fault is reported with the machine restored to a state that permits the instruction to be restarted.		in system tables.
a) Example:- INT 32-255.	Example:- INT 0, INT 5, INT 6.	Example:- INT 2, INT 3, INT 4.	Example:- INT 8, INT 9.

* Exception priorities -

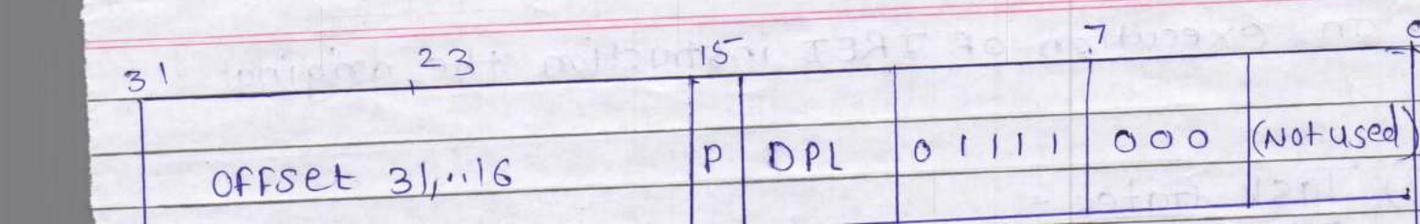
- If more than one interrupt or exception is pending at an instruction boundary, the processor services one of them at a time.
- The priority among classes of interrupt & exception sources is shown in below figure / Table.
- The processor first services a pending interrupt or exception from the class that has the highest priority, transferring control to the first instruction of the interrupt handler.
- Lower priority exceptions are discarded; lower priority interrupts are held pending.
- Discarded exceptions will be rediscovered when the interrupt handler returns control to the point of interruption.

- Real mode uses a 1kB Interrupt vector Table (IVT) ~~sart~~ Starting at address 00000H. Each 4 byte entry in the IVT consists of a CS; IP pair. that specifies the address of the first instruction Service routine [ISR].
- An 8 bit vecter number is shifted two bits to the left to from an index into the IVT.
- The protected mode depends on the interrupt Descriptor Table (IDT) to support the interrupts & exception.
- The IDT Comprises 8 byte gate descriptors for task, trap or interrupt gates.
- The IDT has a maximum size of 256 descripter. The size of the IDT is Controlled by a 16 bit limit value stored in the Interrupt Descriptor Table Register [IDTR].
- The interrupt Descripter table Contains gate Descripters & not vecters.
Table can be located anywhere in the memory.
- The protected mode interrupt descriptor table Can be reside anywhere in the physical address space.
- The interrupt Descripter Table Register (IDTR) is a 48 bit register.
The instructions LIDT (Load interrupt Descriptor Table register) & SIDT (store interrupt Descripter Table register) are used with the IDTR.
- The LIDT is a privileged instruction & is executed whenever the CPL is zero.
- The SIDT is a non-privileged instruction.

* Interrupt Descriptor Table (IDT)

Q. What is IDT & how to locate IDT?

- The ~~interrupt~~ (IDT) associates each interrupt or exception identifier with the descripter for the instructions that service the associated event.
- Like the GDT & LDTs, the IDT is an array of 8-byte descriptors.
  Unlike the GDT & LDTs, the first entry of the IDT may contain a descriptor.
- To form an index into the IDT, the processor multiplies the interrupt or exception identifier by eight. Beacuse there are only 256 identifiers, the IDT needs not contain more than 256 descriptor.
- The IDT may reside anywhere in physical Memory. The processor locates the IDT by means of the IDT register (IDTR).
  The instructions LIDT & SIDT operate on the IDTR.
- Both instructions have one explicit operand: the address in memory of a 6-byte area.
- LIDT (load IDT register) loads the IDT register with the linear base address and limit value contained in the memory operand.
- This instruction can be executed only when the CPL is zero. It is normally used the initialization logic of an operating system when creating an IDT. An operating system may also use it to change from one IDT to another.
- SIDT (store IDT register) copies the base & limit value stored in IDTR to a memory location. This instruction can be executed at any privilege level.

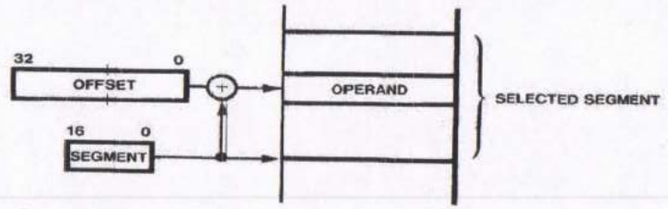


MEMORY ORGANIZATION AND SEGMENTATION

- Segmented model: collection of up to 16,383 linear address spaces.
- Viewed by an applications program (called the *logical address space)*
- The processor maps the 64 terabyte logical address space onto the physical address space (4 GB) by the address translation mechanisms.
- Each of these linear subspaces is called a *segment*.
- A segment is a unit of contiguous address space
- Segment sizes may range from 1 byte up to a maximum of 2^{32} bytes (4 gigabytes).

MEMORY ORGANIZATION AND SEGMENTATION

- A complete pointer in this address space consists of two parts.



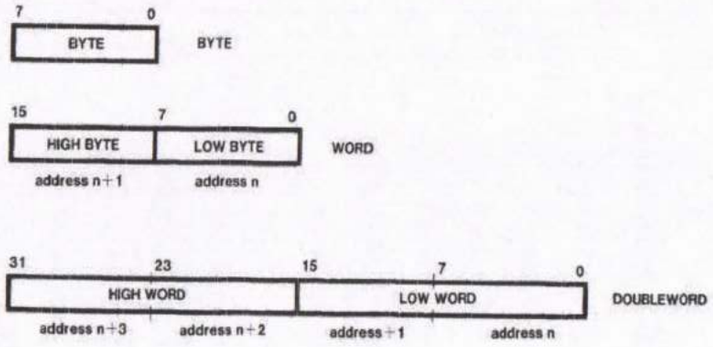
- I. A *segment selector*, which is a 16-bit field that identifies a segment.
- 2. An *offset*, which is a 32-bit ordinal that addresses to the byte level within a segment.

Data Types:

- **Bytes, words, and doublewords** are the fundamental data types
- **Integer:** A signed binary numeric value contained in a 32-bit doubleword, 16-bit word, or 8-bit byte. All operations assume a 2's complement representation.
  - range of an 8-bit integer is -128 through +127
  - 16-bit integers may range from -32,768 through +32,767
  - 32-bit integers may range from -231 through +231-1

Execution unit DATA TYPES

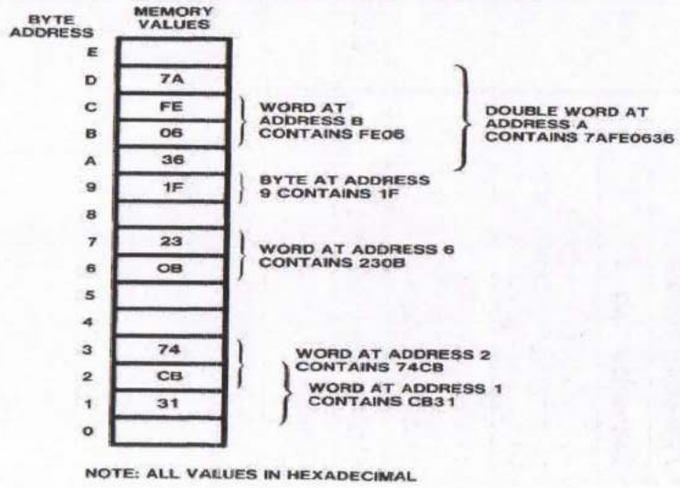
- Fundamental data types:



- **Ordinal:** An unsigned binary numeric value contained in a 32-bit doubleword, 16-bit word, or 8-bit byte. All bits are considered in determining magnitude of the number.
  - range of an 8-bit ordinal number is 0-255;
  - 16 bits can represent values from 0 through 65,535;
  - 32 bits can represent values from 0 through 232-1.
- **Near Pointer:** A 32-bit logical address. A near pointer is an offset within a segment.
- **Far Pointer:** A 48-bit logical address of two components: a 16-bit segment selector component and a 32-bit offset component.
- **String:** A contiguous sequence of bytes, words, or doublewords. A string may contain from zero bytes to 232-1 bytes (4 gigabytes).

- **Bit field:** A contiguous sequence of bits. A bit field may begin at any bit position of any byte and may contain up to 32 bits.
- **Bit string:** A contiguous sequence of bits. A bit string may begin at any bit position of any byte and may contain up to 232-1 bits.
- **BCD:** A byte (unpacked) representation of a decimal digit in the range 0 through 9. Unpacked decimal numbers are stored as unsigned byte quantities. One digit is stored in each byte.
- **Packed BCD:** A byte (packed) representation of two decimal digits, each in the range 0 through 9. One digit is stored in each half-byte.

MEMORY ORGANIZATION AND SEGMENTATION



NOTE: ALL VALUES IN HEXADECIMAL

Registers

The 80386 has eight 32-bit general purpose registers which may be used as either 8 bit, 16 bit or 32 bit registers.

•A 32-bit register known as an extended register, is represented by the register name with prefix E.

•Example : A 32 bit register corresponding to AX is EAX

•So the general purpose registers of 386 are EAX, EBX, ECX, EDX, EBP, ESP, ESI and EDI

Registers

- **BP, SP, SI, DI represents the lower 16 bit of their 32 bit counterparts, and can be used as independent 16 bit registers.**
- **The 16 bit flag register is available along with 32 bit counterpart EFLAGS.**

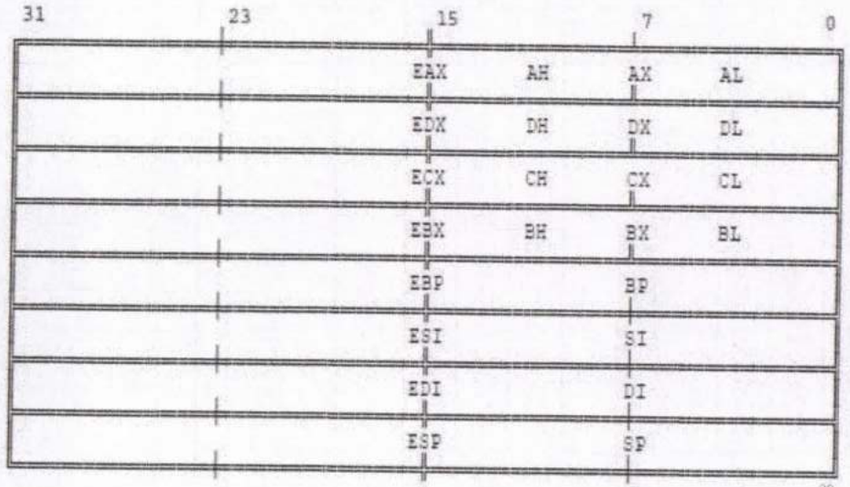
Register Set

- The 80386 contain total 16 registers

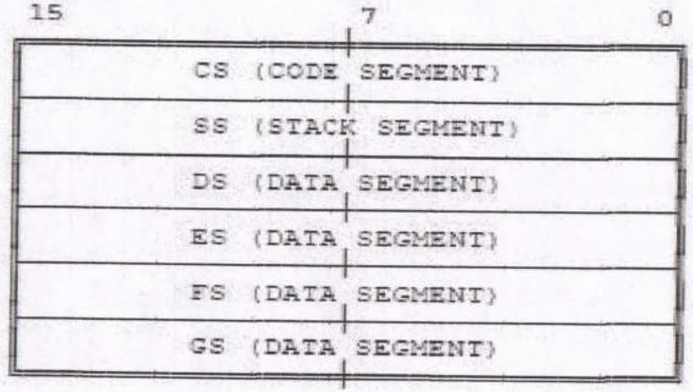
These registers grouped as:

1. General
2. Segment
3. Status and Instruction
4. Control Registers
5. System Address Registers
6. Debug Registers
7. Test Registers

General Registers

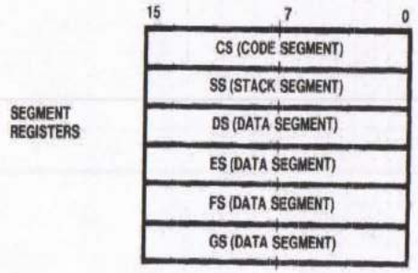


Segment Registers

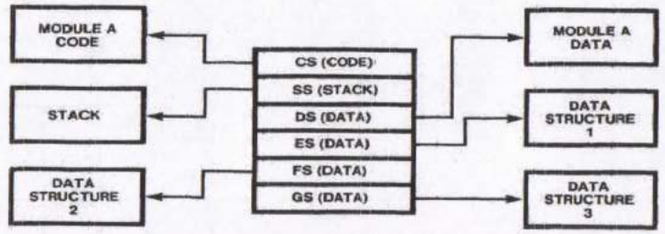


SEGMENT REGISTERS

- Six segments of memory may be immediately accessible to an executing 80386 program.
- The segment registers CS, DS, SS, ES, FS, and GS are used to identify these six current segments.
- Each of these registers specifies a particular kind of segment, as characterized by the associated mnemonics ("code," "data," or "stack").



CS Register

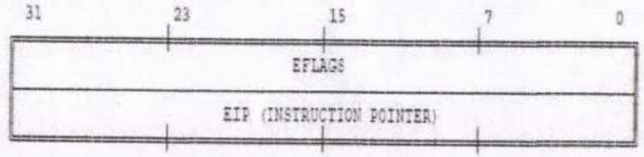


- CS: The segment containing the currently executing sequence of instructions is known as the current code segment.
- The 80386 fetches all instructions from this code segment, using as an offset the contents of the instruction pointer.

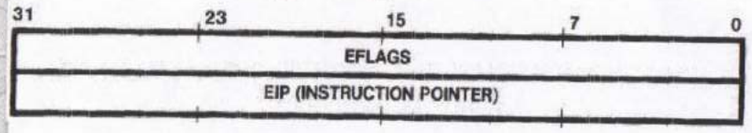
SS and ES ,DS,FS,GS Register

- SS: Subroutine calls, parameters, and procedure activation records usually require to allocate memory as a stack.
- All stack operations use the SS register to locate the stack.
- **Data Registers:** The DS, ES, FS, and GS registers allow the specification of four data segments.
- Access different types of data structures;
- Types of data structures:
- Current module, Exported data, Dynamically created data structure and data Shared with another task.

Status and IP

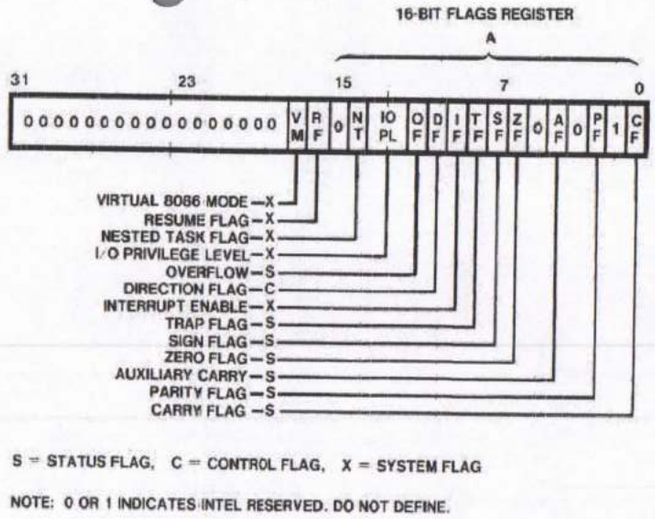


Flag Register



- The Flag register of 80386 is a 32 bit register.
- Out of the 32 bits, Intel has reserved bits D18 to D31, D5 and D3 and set to 0
- While D1 is always set at 1.
- Two extra new flags are added to the 80286 flag to derive the flag register of 80386.
- They are VM and RF flags.

Flag Register



S = STATUS FLAG, C = CONTROL FLAG, X = SYSTEM FLAG

NOTE: 0 OR 1 INDICATES INTEL RESERVED. DO NOT DEFINE.

VM Bit - Virtual Mode Flag

- If this flag is set to VM=1, the 80386 enters the virtual 8086 mode within the protection mode.
- When VM bit is 0, 386 operates in protected mode
- This is to be set only when the 80386 is in protected mode.
- This bit can be set using IRET instruction or any task switch operation only in the protected mode.

RF-Bit Resume Flag

- If RF=1, 386 ignores debug faults and does not take another exception so that an instruction can be restarted after a normal debug exception.
- If RF=0, 386 takes another debug exception to service debug faults
- This flag is used with the debug register breakpoints.
- It is checked at the starting of every instruction cycle and if it is set=1, any debug fault is ignored during the instruction cycle.
- The RF is automatically reset after successful execution of every instruction, except for IRET and POPF instructions

RF- Resume Flag...

- Also, it is not automatically cleared after the successful execution of JMP, CALL and INT instruction causing a task switch.

- **VM** (Virtual 8086 Mode): If set while the Intel386 DX is in Protected Mode, the Intel386 DX will switch to Virtual 8086 operation.
- The VM bit can be set only in Protected Mode, by the IRET instruction (if current privilege level e 0)

- **RF** (Resume Flag): The RF flag is used in conjunction with the debug register breakpoints.
- When RF is set, it causes any debug fault to be ignored on the next instruction.

- **NT** (Nested Task): This flag applies to Protected Mode.
- NT is set to indicate that the execution of this task is nested within another task
- The value of NT in EFLAGS is tested by the IRET instruction to determine whether to do an inter-task return or an intra-task return.

IOPL (Input / Output Privilege Level)

- This two-bit field applies to Protected Mode. IOPL indicates the numerically maximum CPL(current privilege level) value permitted to execute I/O instructions without generating an Exception
- It also indicates the maximum CPL value allowing alteration of the IF (INTR Enable Flag) bit when new values are popped into the EFLAG register

- **IF** (INTR Enable Flag): The IF flag, when set, allows recognition of external interrupts signaled on the INTR pin.

- **TF** (Trap Enable Flag): When TF is set, the Intel386 DX generates an exception 1 trap after the next instruction is executed.
- When TF is reset, exception 1 traps occur only as a function of the breakpoint addresses loaded into debug registers DR0-DR3.

- **OF** (Overflow Flag) : It is set if the operation resulted in a signed overflow. Signed overflow occurs when the operation resulted in carry/borrow into the sign bit (high-order bit) of the result.

- **DF** (Direction Flag) : DF defines whether ESI and/or EDI registers post-decrement or post-increment during the string instructions.
- Post-decrement occurs if DF is set

Flags

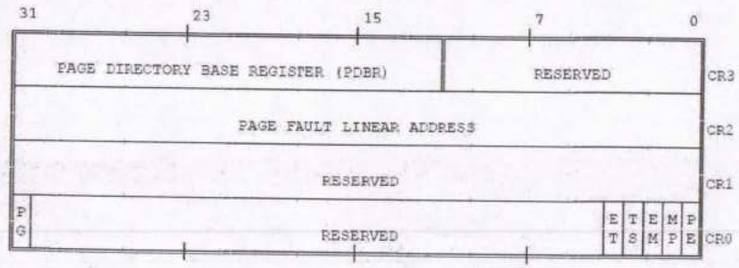
- The arithmetic instructions use CF, SF, ZF, AF, PF, CF
- The control flag DF controls "STRING" instruction
- Clearing DF flag causes string instructions to auto increment or to process string from low to high address

Hidden Registers/ Program invisible registers/ Special Registers

Control Registers

- The 80386 has four 32 bit control registers CR0, CR1, CR2 and CR3 to hold global machine status.
- CR1 is not used in 386 and reserved for future use.
- Load and store instructions are available to access these registers.

Control Registers



System Address Registers

- The 386 supports 4 types of descriptor table:
  - Global descriptor table (GDT),
  - Local descriptor table (LDT),
  - Interrupt descriptor table (IDT)
  - Task state segment descriptor (TSS).
- Four special registers are defined to hold the base address of these tables
  - Global descriptor table Register (GDTR)
  - Local descriptor table Register (LDTR)
  - Interrupt descriptor table Register (IDTR)
  - Task state segment descriptor Register (TR).

Debug Registers

- Intel has provided a set of 8 debug registers for hardware debugging.
- DR4 and DR5 are Intel reserved.
- The initial four registers DR0 to DR3 store four program controllable breakpoint addresses,
- DR6 and DR7 respectively hold breakpoint status and breakpoint control information.

Debug Registers

breakpoint control info	DR7
breakpoint status	DR6
RESERVED	DR5
RESERVED	DR4
Linear breakpoint address 3	DR3
Linear breakpoint address 2	DR2
Linear breakpoint address 1	DR1
Linear breakpoint address 0	DR0

31 0

Test Registers

- Two test register are provided by 80386 for page caching namely test control and test status register.

INSTRUCTION FORMAT

- The information encoded in an 80386 instruction includes a specification of ;
- Operation to be performed (Opcode).
- Type of the operands to be manipulated,
- Location of these operands.

Opcode Source(M/ Reg) Destination(M/ Reg)

Operand Selection

- In the instruction itself(immediate operand)
- In a register
- In memory
- At an I/O port

- Implicit operand
- Explicit operand
- Implicit and Explicit Operand

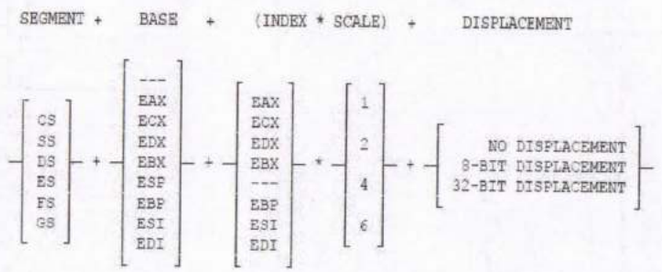
INSTRUCTION FORMAT

- Two-operand instructions of the 80386 permit operations of the following kinds:
- • Register-to-register
- • Register-to-memory
- • Memory-to-register
- • Immediate-to-register
- • Immediate-to-memory
- Certain string instructions and stack manipulation instructions transfer data from memory to memory.
- Push and pop stack operations allow transfer between memory operands and the memory-based stack.

- Immediate Operands
- Register Operands
- Memory Operands
- Segment Selection

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference that uses ESP or EBP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination.
Destination Strings	Extra (ES)	Destination of string instructions.

Effective Address Computation



Effective Address Computation...

- **Displacement:** Indicates the offset of the operand . Used to directly address a statically allocated scalar operand.
- **Base:** Offset is specified indirectly in one of the general registers, as for based variables.
- **Base+displacement:**
- To index into static array when element size is not 2,4,8 bytes.
- Access item of record. Displacement component locates an item within record.
- **(Index*scale) + displacement:** Provides efficient indexing into a static array when element size is 2,4,8 bytes.

Effective Address Computation...

- **Base + Index + Displacement:** Two registers used together support either a two dimensional array (where displacement determine beginning of array) or one of several instances of an array of records (where displacement indicates an item in the record.)
- **Base + (Index * Scale) + displacement:** This combination provides efficient indexing of a two-dimensional array when element of the array are 2,4,8 bytes wide.

Interrupts and Exceptions

- Two mechanism for interrupting program execution
- **Exceptions** are synchronous events that are the responses of the CPU to certain conditions detected during the execution of an instruction.
- **Interrupts** are asynchronous events typically triggered by external devices needing attention.

Interrupts and Exceptions...

Table 2-2. 80386 Reserved Exceptions and Interrupts

Vector Number	Description
0	Divide Error
1	Debug Exceptions
2	NMI Interrupt
3	Breakpoint
4	INTO Detected Overflow
5	BOUND Range Exceeded
6	Invalid Opcode
7	Coprocessor Not Available
8	Double Exception
9	Coprocessor Segment Overrun
10	Invalid Task State Segment
11	Segment Not Present
12	Stack Fault
13	General Protection
14	Page Fault
15	(reserved)
16	Coprocessor Error
17-32	(reserved)

APPLICATIONS INSTRUCTION SET

- To write application software for the 80386 executing in protected virtual-address mode.
- **DATA MOVEMENT INSTRUCTIONS**
- They fall into the following classes:
- 1. General-purpose data movement instructions.
- 2. Stack manipulation instructions.
- 3.Type-conversion instructions.

General-Purpose Data Movement Instructions

- MOV (Move) transfers a byte, word, or double word from the source operand to the destination operand.
- The MOV instruction is useful for transferring data along any of these paths
- • To a register from memory
- • To memory from a register
- • Between general registers
- • Immediate data to a register
- • Immediate data to a memory
- XCHG (Exchange) swaps the contents of two operands.

Stack Manipulation Instructions

- PUSH (Push) decrements the stack pointer (ESP), then transfers the source operand to the top of stack indicated by ESP
- PUSH is often used to place parameters on the stack before calling a procedure.
- The PUSH instruction operates on memory operands, immediate operands, and register .
- PUSHA (Push All Registers) saves the contents of the eight general registers on the stack..
- The processor pushes the general registers on the stack in the following order:
- EAX, ECX, EDX, EBX, the initial value of ESP before EAX was pushed, EBP, ESI, and EDI.

Type Conversion Instructions

- The type conversion instructions convert bytes into words, words into double words, and double words into 64-bit items (quad-words).
- CWD, CDQ, CBW, and CWDE
- CWD (Convert Word to Doubleword)
- CBW (Convert Byte to Word)
- CDQ (Convert Doubleword to Quad-Word)
- CWDE (Convert Word to Doubleword Extended)
- MOVSX (Move with Sign Extension)
- MOVZX (Move with Zero Extension)

BINARY ARITHMETIC INSTRUCTIONS

** Addition and Subtraction Instructions

- ADD D , S (sets CF is there is carry)
- ADC D , S (D= D+S+C)
- INC D (Increment Byte,Word or Doubleword by 1)
- SUB D , S (sets CF is there is borrow)
- SBB D , S (D= D-S-C)
- DEC D (Decrement Byte,Word or Doubleword by 1)

BINARY ARITHMETIC INSTRUCTIONS

** Comparison and Sign Change Instructions

- CMP D, S (Destination-Source)

  Updates OF, SF, ZF, AF, PF and CF

- NEG D

  Subtracts a signed integer operand from zero

BINARY ARITHMETIC INSTRUCTIONS

** Multiplication and Divide Instructions

- MUL S (Unsigned Integer Multiply)
- IMUL S (Signed Integer Multiply)
- DIV S (Unsigned Integer Divide)

Dividend	Quotient	Remainder
AX	AL	AH
DX:AX	AX	DX
EDX:EAX	EAX	EDX

- IDIV S (Signed Integer Divide)

  Uses same registers as in DIV

DECIMAL ARITHMETIC INSTRUCTIONS

- Decimal Arithmetic is performed by combining the binary arithmetic instructions with decimal arithmetic instructions.
- Decimal Arithmetic instructions are used in one of the following ways
  - To adjust the results of a previous binary arithmetic operation to produce a valid packed or unpacked decimal result.
  - To adjust the inputs to a subsequent binary arithmetic operation so that the operation will produce a valid packed or unpacked decimal result.
- These instructions operate only on the AL or AH registers. Most utilize the AF flag.

DECIMAL ARITHMETIC INSTRUCTIONS

** Packed BCD Adjustment Instructions

- **DAA (Decimal Adjust after Addition)**
  - Adjusts the result of adding two valid packed decimal operands in AL.
  - DAA instruction gives us correct decimal output instead of hexadecimal.
  - Carry flag is set if carry was needed.
- **DAS (Decimal Adjust after Subtraction)**
  - Adjusts the result of Subtracting two valid packed decimal operands in AL.
  - DAS instruction gives us correct decimal output instead of hexadecimal.
  - Carry flag is set if borrow was needed.

DECIMAL ARITHMETIC INSTRUCTIONS
** Unpacked BCD Adjustment Instructions

- **AAA (Ascii Adjust After Addition)**
- AL contain valid unpacked decimal number and AH=00
- AAA must always follow addition of two unpacked decimal operands in AL.
- Carry flag is set and AH is incremented if a carry is necessary.
- **AAS (Ascii Adjust After Subtraction)**
- AL contain valid unpacked decimal number and AH=00
- AAS must always follow Subtraction of one unpacked decimal operands from another in AL.
- Carry flag is set and AH is incremented if a borrow is necessary.

DECIMAL ARITHMETIC INSTRUCTIONS
** Unpacked BCD Adjustment Instructions

- **AAM (Ascii Adjust After Multiplication)**
- Corrects multiplication of two unpacked decimal number.
- The high order digit is left in AH, the low order digit in AL.
- **AAD (Ascii Adjust After Division)**
- Modifies numerator in AH and AL for unpacked decimal operands divide operation.
- Quotient produced will be valid unpacked decimal.
- The high order digit is left in AH, the low order digit in AL.
- Adjusts the result in AL and make AH=00

LOGICAL INSTRUCTIONS

- The group of logical instructions includes:
- The Boolean operation instructions
- Bit test and modify instructions
- Bit scan instructions
- Rotate and shift instructions
- Byte set on condition

The Boolean operation instructions

- NOT (Not)

Inverts the bits in the specified operand to form a one"s complement of the operand. Has no effect on flags.

- AND, OR, and XOR

AND- is useful instruction for turning a particular bit off. (Turn to 0)

OR- is useful instruction for setting a particular bit on. (Turn to 1)

XOR- is useful instruction for clearing a register. Or useful for toggling particular bit without changing other bits.

Bit test and modify instructions

- This group of instructions operates on a single bit which can be in memory or in a general register.
- These instructions first assign the value of the selected bit to CF, the carry flag.
- Then a new value is assigned to the selected bit, as determined by the operation.

Table 3-1. Bit Test and Modify Instructions

Instruction	Effect on CF	Effect on Selected Bit
Bit (Bit Test)	CF ← BIT	(none)
BTS (Bit Test and Set)	CF ← BIT	BIT ← 1
BTR (Bit Test and Reset)	CF ← BIT	BIT ← 0
BTC (Bit Test and Complement)	CF ← BIT	BIT ← NOT(BIT)

Bit scan instructions

- These instructions scan a word or doubleword for a one-bit and store the index of the first set bit into a register.
- The bit string being scanned may be either in a register or in memory.
- Affects ZF=1 if word is zero, otherwise clear ZF
- **BSF (Bit Scan Forward)** scans from low-order to high-order (starting from bit index zero).
- **BSR (Bit Scan Reverse)** scans from high-order to low-order (starting from bit index 15 of a word or index 31 of a doubleword).

Shift and Rotate Instructions

- These instructions fall into the following classes:
- Shift instructions
- Double shift instructions
- Rotate instructions

SHIFT INSTRUCTIONS

- The bits in bytes, words, and double words may be shifted arithmetically or logically.
- CF always contains the value of the last bit shifted out of the destination operand.
- OF is set if the value of the high-order (sign) bit was changed by the operation.
- SAL (Shift Arithmetic Left)
- SHL (Shift Logical Left)
- SHR (Shift Logical Right)
- SAR (Shift Arithmetic Right)
- ROL (Rotate Left)
- ROR (Rotate Right)
- RCL (Rotate Through Carry Left)

SAL/SHL

- SAL (Shift Arithmetic Left) shifts the destination byte, word, or double word operand left by one or by the number of bits specified in the count operand
- The processor shifts zeros in from the right (low-order) side of the operand as bits exit from the left (high-order) side.
- Sal AX,CL

	OF	CF	OPERAND
BEFORE SHL OR SAL	X	X	1000100010001000100010001111
AFTER SHL OR SAL BY 1	1	1 ←	0001000100010001000100011110 ← 0
AFTER SHL OR SAL BY 10	X	0 ←	0010001000100010001111000000000 ← 0

SHL (WHICH HAS THE SYNONYM SAL) SHIFTS THE BITS IN THE REGISTER OR MEMORY OPERAND TO THE LEFT BY THE SPECIFIED NUMBER OF BIT POSITIONS. CF RECEIVES THE LAST BIT SHIFTED OUT OF THE LEFT OF THE OPERAND. SHL SHIFTS IN ZEROS TO FILL THE VACATED BIT LOCATIONS. THESE INSTRUCTIONS OPERATE ON BYTE, WORD, AND DOUBLEWORD OPERANDS.

SHR (Shift Logical Right)

		OPERAND	CF
BEFORE SHR		1000100010001000100010001111	X
AFTER SHR BY 1	0 →	0100010001000100010001000111 →	1
AFTER SHR BY 10	0 →	0000000000100010001000100010 →	0

SHR SHIFTS THE BITS OF THE REGISTER OR MEMORY OPERAND TO THE RIGHT BY THE SPECIFIED NUMBER OF BIT POSITIONS. CF RECEIVES THE LAST BIT SHIFTED OUT OF THE RIGHT OF THE OPERAND. SHR SHIFTS IN ZEROS TO FILL THE VACATED BIT LOCATIONS.

SAR (Shift Arithmetic Right)

- The processor preserves the sign of the operand by shifting in 0 on the left (high-order) side if the value is positive
- or by shifting by 1 if the value is negative.
- SAR is rounded toward negative infinity

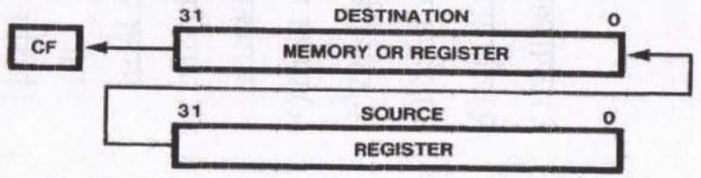
	POSITIVE OPRAND	CF	OF
BEFORE SAR	0100010001000100010001000111	X	X
AFTER SAR BY 1 →	0010001000100010001000100011 →	1	0

	NEGATIVE OPERAND	CF	OF
BEFORE SAR	1100010001000100010001000111	X	X
AFTER SAR BY 1 →	1110001000100010001000100011 →	1	0

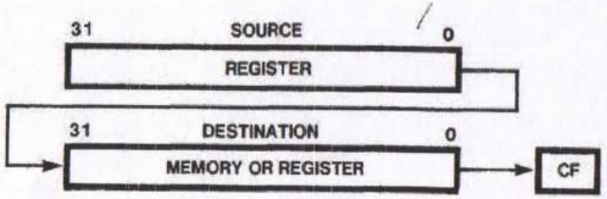
SAR PRESERVES THE SIGN OF THE REGISTER OR MEMORY OPERAND AS IT SHIFTS THE OPERAND TO THE RIGHT BY THE SPECIFIED NUMBER OF BIT POSITIONS. CF RECEIVES THE LAST BIT SHIFTED OUT OF THE RIGHT OF THE OPERAND.

DOUBLE-SHIFT INSTRUCTIONS

- These instructions provide the basic operations needed to implement operations on long unaligned bit strings.
- The double shifts operate either on word or double word operands, as follows:
- SHLD (Shift Left Double) :shifts bits of the *R/M* field to the left, while shifting high-order bits from the Reg field into the *R/M* field on the right.
- The result is stored back into the *R/M* operand.
- The Reg field is not modified.



SHRD (Shift Right Double) shifts bits of the *R/M* field to the right, while shifting low-order bits from the Reg field into the *R/M* field on the

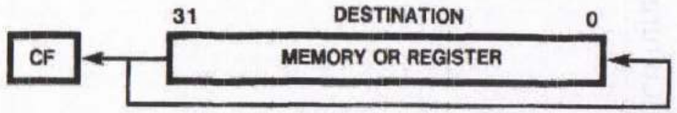


ROTATE INSTRUCTIONS

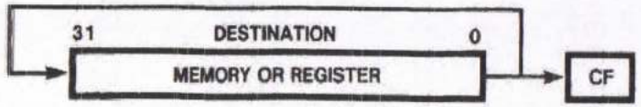
- Rotate instructions allow bits in bytes, words, and double words to be rotated.
- Bits rotated out of an operand are not lost as in a shift, but are "circled" back into the other "end" of the operand.
- Rotates affect only the carry and overflow flags.
- CF may act as an extension of the operand.
- CF always contains the value of the last bit rotated out.

ROL and ROR

- ROL (Rotate Left) rotates the byte, word, or double word destination operand left by one or by the number of bits specified in the count operand .

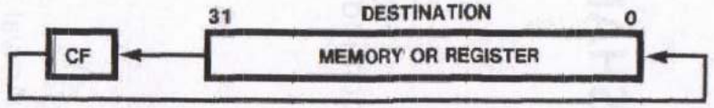


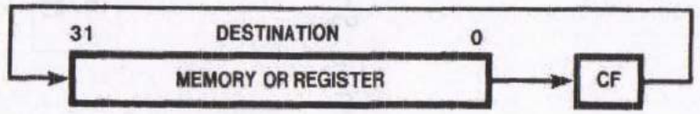
- ROR(Rotate Right)



RCL and RCR

- It treats CF as a high-order one-bit extension of the destination operand.





CONTROL TRANSFER INSTRUCTIONS

- **Unconditional Transfer Instructions:**
- **JMP-** JMP is a one-way transfer of execution; it does not save a return address on the stack.
- **CALL-** activates an out-of-line procedure, saving on the stack the address of the instruction following the CALL for later use by a RET (Return) instruction.***stack
- **RET-** terminates the execution of a procedure and transfers control through a back-link on the stack to the program that originally invoked the procedure.*** back link on the stack EIP
- **IRET-** returns control to an interrupted procedure. IRET differs from RET in that it also pops the flags from the stack into the flags register.

Conditional Transfer Instructions:

Unsigned Conditional Transfers		
Mnemonic	**Condition Tested**	**"Jump If..."**
JA/JNBE	(CF or ZF) = 0	above/not below nor equal
JAE/JNB	CF = 0	above or equal/not below
JB/JNAE	CF = 1	below/not above nor equal
JBE/JNA	(CF or ZF) = 1	below or equal/not above
JC	CF = 1	carry
JE/JZ	ZF = 1	equal/zero
JNC	CF = 0	not carry
JNE/JNZ	ZF = 0	not equal/not zero
JNP/JPO	PF = 0	not parity/parity odd
JP/JPE	PF = 1	parity/parity even
Signed Conditional Transfers		
Mnemonic	**Condition Tested**	**"Jump If..."**
JG/JNLE	((SF xor OF) or ZF) = 0	greater/not less nor equal
JGE/JNL	(SF xor OF) = 0	greater or equal/not less
JL/JNGE	(SF xor OF) = 1	less/not greater nor equal
JLE/JNG	((SF xor OF) or ZF) = 1	less or equal/not greater
JNO	OF = 0	not overflow
JNS	SF = 0	not sign (positive, including 0)
JO	OF = 1	overflow
JS	SF = 1	sign (negative)

Conditional

- LOOP
- LOOPE (Loop While Equal) and LOOPZ (Loop While Zero)

These instructions automatically decrement the ECX register before testing ECX and ZF for the branch conditions.

ECX=0 and ZF=0 ignore

- LOOPNE (Loop While Not Equal) and LOOPNZ (Loop While Not Zero)

ECX=0 and ZF=1 ignore

- JCXZ (Jump if ECX Zero) branches to the label specified in the instruction if it finds a value of zero in ECX.

Software Generated Interrupts

- **INT n (Software Interrupt)** activates the interrupt service routine that corresponds to the number coded within the instruction. The interrupt service routine terminates with an IRET instruction that returns control to the instruction that follows INT.
- **INTO (Interrupt on Overflow)** invokes interrupt 4 if OF is set.
- **BOUND (Detect Value Out of Range)** verifies that the signed value contained in the specified register lies within specified limits. An interrupt (INT 5) occurs if the value contained in the register is less than the lower bound or greater than the upper bound.

STRING AND CHARACTER TRANSLATION INSTRUCTIONS

- 1. A set of primitive string operations
  - MOVS — Move String
  - CMPS — Compare string
  - SCAS — Scan string
  - LODS — Load string
  - STOS — Store string
- 2. Indirect, indexed addressing, with automatic incrementing or decrementing of the indexes.

Indexes:

  - ESI — Source index register
  - EDI — Destination index register
- Control flag:
  - DF — Direction flag
- Control flag instructions:
  - CLD Clear direction flag instruction
  - STD — Set direction flag instruction
- 3. Repeat prefixes
  - REP Repeat while ECX not zero
  - REPE/REPZ Repeat while equal or zero
  - REPNE/REPNZ Repeat while not equal or not zero

Repeat Prefixes:

Prefix	Termination Condition 1	Termination Condition 2
REP	ECX = 0	(none)
REPE/REPZ	ECX = 0	ZF = 0
REPNE/REPNZ	ECX = 0	ZF = 1

Indexing and Direction flag Control

- The addresses of the operands of string primitives are determined by the ESI and EDI registers.
- ESI points to source operands. By default, ESI refers to a location in the segment indicated by the DS segment register. A segment-override prefix may be used, however, to cause ESI to refer to CS, SS, ES, FS, or GS.
- EDI points to destination operands in the segment indicated by ES; no segment override is possible.
- The direction flag determines whether they are incremented or decremented.

String Instructions

- **MOVS (Move String)** moves the string element pointed to by ESI to the location pointed to by EDI. The MOVS instruction, when accompanied by the REP prefix, operates as a memory-to-memory block transfer. **MOVSB, MOVSW, MOVSD**
- **CMPS (Compare Strings)** subtracts the destination string element (at ES:EDI) from the source string element (at ESI) and updates the flags AF, SF, PF, CF and OF. If the string elements are equal, ZF=1; otherwise, ZF=0. CMPSB compares bytes, CMPSW compares words, and CMPSD compares doublewords.
- **SCAS (Scan String)** subtracts the destination string element at ES:EDI from EAX, AX, or AL and updates the flags AF, SF, ZF, PF, CF and OF. If values are equal, ZF=1; otherwise, ZF=0.

- **LODS (Load String)** places the source string element at ESI into EAX for doubleword strings, into AX for word strings, or into AL for byte strings.LODS increments or decrements ESI according to DF.

- **STOS (Store String)** places the source string element from EAX, AX, or AL into the string at ES:DSI. STOS increments or decrements EDI according to DF.

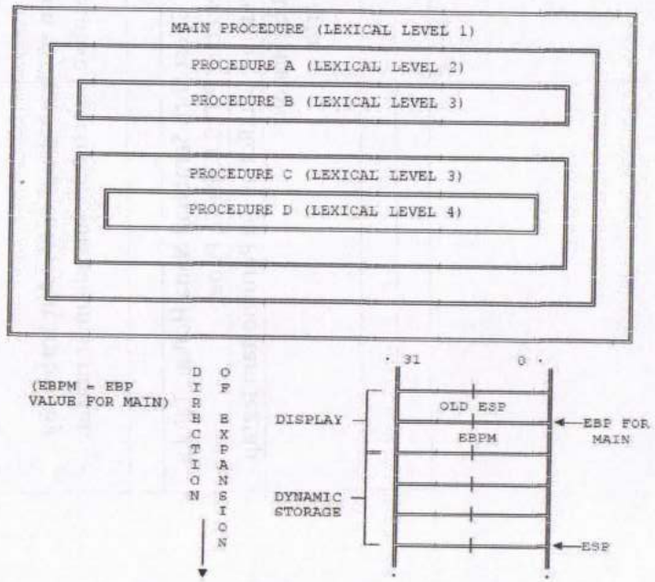
Instructions for Block Structured Languages

- Instructions in this section provide machine-language support for functions normally found in high-level languages.

- **ENTER:** creates a stack frame that may be used to implement the scope rules of block structured high-level languages.
- **LEAVE:** A LEAVE instruction at the end of a procedure complements an ENTER at the beginning of the procedure to simplify stack management and to control access to variables for nested procedures.

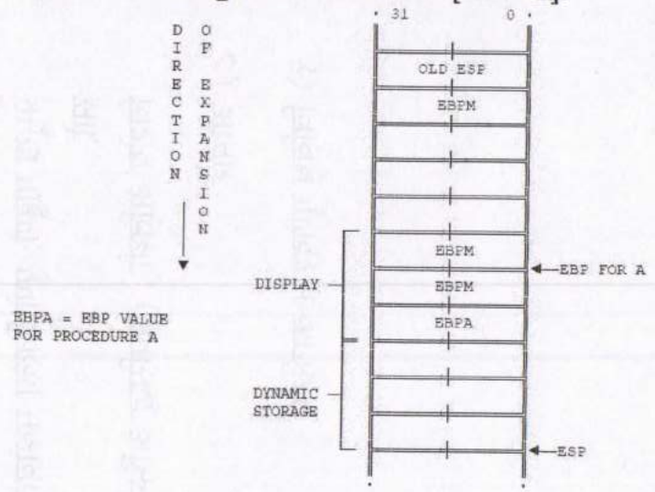
Enter

- Includes two parameters. The first parameter specifies the number of bytes of dynamic storage to be allocated on the stack for the routine being entered. The second parameter corresponds to the lexical nesting level (0-31) of the routine.
- The specified lexical level determines how many sets of stack frame pointers the CPU copies into the new stack frame from the preceding frame.
- This list of stack frame pointers is sometimes called the **display.**
- **EX. ENTER 2048,3**

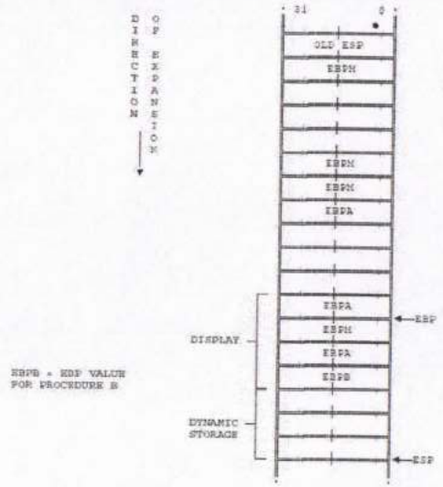


- **ESP** serves as a starting point for all PUSH and POP operations within that procedure.
- To enable a procedure to address its display, ENTER leaves **EBP** pointing to the beginning of the new stack frame.
- **ENTER** provides variable access to next lexical level procedure through a display that provides addressability to the calling program's stack frame.
- 1. MAIN PROGRAM has variables at fixed locations.
- 2. PROCEDURE A can access only the fixed variables of MAIN.
- 3. PROCEDURE B can access only the variables of PROCEDURE A and MAIN.
  PROCEDURE B cannot access the variables of PROCEDURE C or D.
- 4. PROCEDURE C can access only the variables of PROCEDURE A and MAIN.
- PROCEDURE C cannot access the variables of PROCEDURE B or D.
- 5. PROCEDURE D can access the variables of PROCEDURE C, PROCEDURE A, and MAIN.
  PROCEDURE D cannot access the variables of PROCEDURE B.

- Procedure A can access variables in MAIN since MAIN is at level 1. Therefore the base for the dynamic storage for MAIN is at [EBP-2].



- B can access variables in A and MAIN by fetching from the display the base addresses of the respective dynamic storage areas.



LEAVE

- **LEAVE** (Leave Procedure) reverses the action of the previous ENTER instruction. The LEAVE instruction does not include any operands.
- LEAVE copies EBP to ESP to release all stack space allocated to the procedure by the most recent ENTER instruction.
- Then LEAVE pops the old value of EBP from the stack.

FLAG CONTROL INSTRUCTIONS

Carry and Direction flag control Instructions

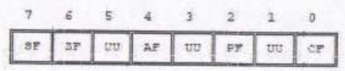
Flag Control Instruction	Effect
STC (Set Carry Flag)	CF <- 1
CLC (Clear Carry Flag)	CF <- O
CMC (Complement Carry Flag)	CF <- NOT (CF)
CLD (Clear Direction Flag)	DF <- O
STD (Set Direction Flag)	DF <- 1

FLAG CONTROL INSTRUCTIONS

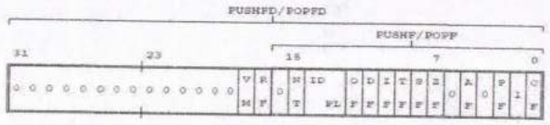
Flag Transfer Instructions

- The flag transfer instructions allow a program to alter the other flag other than CF and DF bits with the bit manipulation instructions after transferring these flags to the stack or the AH register.
- **LAHF** (Load AH from Flags) copies SF, ZF, AF, PF, and CF to AH bits 7, 6, 4, 2, and 0, respectively (see Figure 3-22).The contents of the remaining bits (5, 3, and 1) are undefined.The flags remain unaffected.
- **SAHF** (Store AH into Flags) transfers bits 7, 6, 4, 2, and 0 from AH into SF, ZF, AF, PF, and CF, respectively

LAHF and SAHF



- **PUSHF** (Push Flags) decrements ESP by two and then transfers the low-order word of the flags register to the word at the top of stack pointed to by ESP. The variant PUSHFD decrements ESP by four, then transfers both words of the extended flags register to the top of the stack pointed to by ESP (the VM and RF flags are not moved, however).
- **POPF** (Pop Flags) transfers specific bits from the word at the top of stack into the low-order byte of the flag register, then increments ESP by two.The variant POPFD transfers specific bits from the doubleword at the top of the stack into the extended flags register (the RF and VM flags are not changed, however), then increments ESP by four.



COPROCESSOR INTERFACE INSTRUCTIONS

- The 80386 also has features to support emulation of the numeric coprocessor when the coprocessor is absent.
- ESC (Escape) : **Used by Coprocessor** is a 5-bit sequence that begins the opcodes that identify floating point numeric instructions.
- ESC pattern tells 80386 to send the opcode and addresses of operands to numeric coprocessor.
- The numeric coprocessor uses the escape instructions to perform high-performance, high-precision floating point arithmetic.

COPROCESSOR INTERFACE INSTRUCTIONS

- **WAIT** (Wait)

- Suspends (80386) program execution until the 80386 CPU detects that the BUSY pin is inactive.
- This condition indicates that the coprocessor has completed its processing and CPU may obtain result.

SEGMENT REGISTER INSTRUCTIONS (In Groups)

Segment-register transfer instructions.

MoV ••• , SegReg

MoV SegReg, •••

PUSH SegReg

POP SegReg

Control transfers to another executable segment.:

JMP far

CALL far

RET far

Data pointer instructions.

LOS

LES

LFS

LGS

LSS

Data Pointer Instructions

- LDS (Load Pointer Using DS)

  LDS ESI, STRING_X

The source operand must be a memory operand, and the destination operand must be a general register. DS receives the segment-selector of the pointer. The destination register receives the offset part of the pointer, which points to a specific location within the segment.

- LES (Load Pointer Using ES)

  LES EDI, DESTINATION_X

operates identically to LDS except that ES receives the segment selector rather than DS.

- **LFS** (Load Pointer Using FS)

Operates identically to LDS except that FS receives the segment selector rather than DS.

- **LGS** (Load Pointer Using GS)

Operates identically to LDS except that GS receives the segment selector rather than DS.

- **LSS** (Load Pointer Using SS)

Operates identically to LDS except that SS receives the segment selector rather than DS.

Miscellaneous Instructions

- **LEA (Load Effective Address)**

Transfers the offset of the source operand (rather than its value) to the destination operand. The source operand must be a memory operand, and the destination operand must be a general register. This instruction is especially useful for initializing registers before the execution of the string primitives (ESI, EDI)

LEA EBX, EBCDIC_TABLE

- **NOP (No Operation)**

NOP (No Operation) occupies a byte of storage but affects nothing but the instruction pointer, EIP.

- **XLAT (Translate)**

XLAT (Translate) replaced a byte in the AL register with a byte from a user-coded translation table. When XLAT is executed, AL should have the unsigned index to the table addressed by EBX. XLAT changes the contents of AL from table index to table entry. EBX is unchanged.

Feedback

- Teaching Method
- Am I audible?
- Am I interactive with you?
- Study Material (PPT, Notes etc.)
- Any Suggestions are welcome...

Solve

- Explain 80386 Architecture
- Write a procedure for „**ASCII to HEX**" and „**HEX to ASCII** "Conversion.
- List and Explain General Purpose Registers.
- Explain following flags from EFLAG register
  1. VM 2. IOPL 3. RF
- Explain following instructions
  1. XOR 2. PUSHA 3. CALL 4. JMP

THANK YOU
All THE BEST !!