Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Department of Information Technology

Continuous Assessment Test – I

Answer Key

Degree & Branch	B.Tech. Information Technology	**Semester**	IV
Subject Code & Name	UIT1403 MICROPROCESSORS AND MICROCONTROLLERS		
Time: 90 Minutes **Date: 29-03-2022**	**Answer All Questions**	**Maximum: 50 Marks**	

Course Outcome:

CO1 - Write programs to run on 8086 Microprocessor based systems.
CO2 - Design the system using memory chips and peripheral chips for microprocessor and microcontroller.
CO3 - Analyse, specify, design, write and test assembly language programs.

Part – A (6 × 2 = 12 Marks)

K2	1	The following are the content of Registers AX = 3000 CX = 1000 BX = A000 DX = 0100 CS = 80000 SS = F000 DS =123A SI = 341B DI =1000 MOV AX, [BX][SI][12] Calculate the Effective address of Source data. EA = 20 bit of DS + BX + SI + 12 = 123A0 + A000 + 341B + 12 = **1F7CD**	CO1	2.1.3
K2	2	AX = 1234 BX = 0000 MOV BX, AX Name the Flags that get affected after execution of the above Instruction. ANS: **No Flag is affected**	CO1	2.1.3
K1	3	What is the difference between Rotate and Shift Instruction in 8086? **Rotate : Uses Carry flag** **Shift: Don't Use Carry Flag. Zero will be inserted in LSB or MSB.**	CO3	1.3.1
K1	4	What is the length of the Instruction Format which takes Immediate Operand to Register? **Ans: 4 Bytes**	CO3	1.4.1
K2	5	Why the length of logical segment is 64KB in 8086?	CO3	1.4.1

		Ans: Length of Index Reg and Pointer Reg is **16 Bit.**		
K1	6	List two difference between maximum mode and minimum mode configuration of 8086. **Min Mode:** one processor in the system minimum mode. INTA(bar), ALE, DEN(bar), DT/R(bar), M/IO(bar), HLDA,HOLD and WR(bar) control signals. **Max Mode:** multiple processors in the system. QS1,QS0,S0(bar),S1(bar),S2(bar), LOCK(bar),RQ(bar)/GT1,RQ(bar)/GT0 control signals.	CO3	1.3.1

Part – B (3 × 6 = 18 Marks)

K2	7	AX = 1234 BX= 9999
		Write a Assembly language Program to Perform (AX – BX)
		
		MOV CX,0000H
		MOV AX, 1234H
		MOV BX, 9999H
		SUB BX
		JNC L1
		INC CX
		L1: MOV [8500], AX
		MOV [8502], CX
		HLT
			CO1	13.1.1

K2	8	Explain about the following String Manipulation Instructions in detail. REP, CMPS , MOVSB REP: This instruction is used as a prefix to other instructions, the instruction to which the REP prefix is provided, is executed repeatedly until the CX register becomes zero (at each iteration CX is automatically decremented by one). CMPS: Compare String Byte. If both the strings are completely equal, CX becomes zero, The ZF is set. Otherwise Reset. MOVSB: Move String Byte The starting byte of source string: SI + DS Address of the destination locations:DI + ES	CO1	13.1.1
K2	9	Explain about Instruction format of 8086.	CO3	3.1.1

General instruction form for the 8086

- An instruction can be coded with 1 to 6 bytes

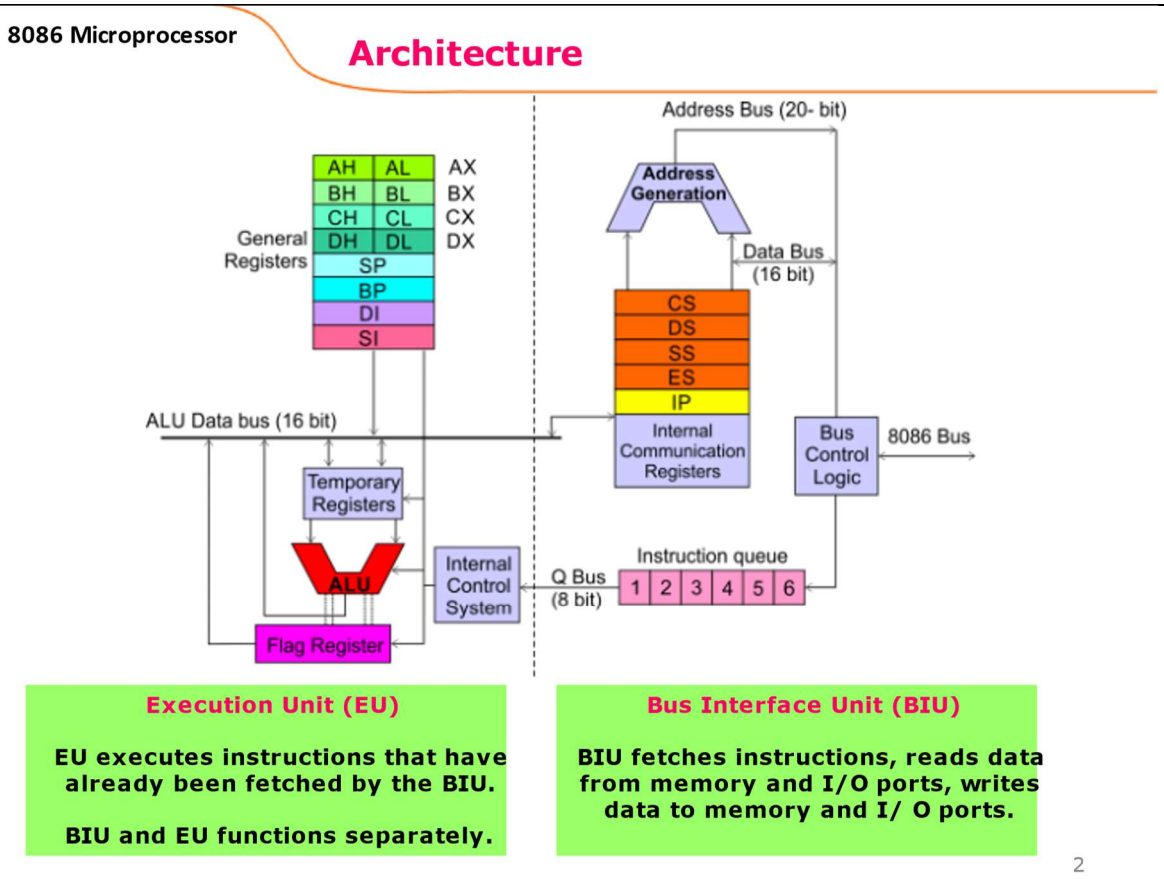
Opcode - 6	D - 1	W -1	1st byte
MOD - 2	Reg - 3	R/M - 3	2nd byte
Displacement or data (optional) up to 4 bytes			

Mov mem16, reg16 89(opcode) , mod reg r/m, (address)

Mov [8500], DX

Part – C (2 × 10 = 20 Marks)

K3	10	a. Draw and explain the internal architecture of 8086. b. Calculate the Physical address for the given Logical address **1980 : 78FE** **Ans: 19800 + 78FE = 210FE**	CO3	1.3.1



OR

K3	11	a. Describe various addressing modes of 8086 with examples. b. Calculate the Effective address for the given Instruction and Specify the type of Addressing Mode. MOV [BX + 5], DX **Register Relative Addressing.** **EA = 10H * DS + 5H +[BX]**	CO3	1.3.1

8086 Microprocessor

Addressing Modes

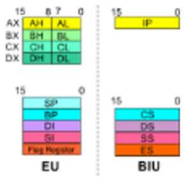
1. **Register Addressing**
2. **Immediate Addressing**
3. **Direct Addressing**
4. **Register Indirect Addressing**
5. **Indexed Addressing**
6. **Register Relative Addressing**
7. **Based Index Addressing**
8. **Relative Based Indexed**

The effective address is formed by adding an 8 or 16-bit displacement with the sum of contents of any of the base registers (BX or BP) and any one of the index registers, in a default segment

Example:

MOV AX, 50H [BX] [SI]

EA = 10H * DS + [BX] + [SI] + 50H



16

K3	12	a. Draw the Status Register format and Explain about each flag in detail with Instructions. b. Identify the flags that are affected on execution of **DAA** Instruction. AC – Affected OF – Not affected.	CO3	1.3.1

8086 Microprocessor

Architecture — Execution Unit (EU)

Flag Register

Auxiliary Carry Flag
This flag is set if there is an overflow out of bit3 ie carry from lower nibble to higher nibble Used in BCD Operation Not available for programmer

Carry Flag
This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction

Sign Flag
This flag is set, when the result of any computation is negative

Zero Flag
This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag
This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Over flow Flag
This flag is set if result is out of range. For addition this flag is set when there is a carry in to the MSB and no carry out of the MSB or vice versa

Trap Flag
If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag
This is used by string manipulation instructions If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto decrementing mode.

Interrupt Flag
Causes the 8086 to recognize external mask interrupts; clearing I disables these interrupts.

		OR		
K3	13	The 8086 signals are categorized in 3 Groups. a. Signal common for both Minimum and Maximum modes. b. Signal for Minimum mode operation c. Signal for Maximum mode operation. Identify the signal for Minimum mode operation and Explain its function. HOLD – Hold Request – Input Signal HLDA – Hold Ack – Output Data Enable Data Transmit/ Receive ALE Interrupt ACK Memory/ IO	CO3	1.3.1