REG.NO.:

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT School of Electronics Engineering (SENSE)
CONTINUOUS ASSESSMENT TEST - 1
WINTER SEMESTER 2025-2026

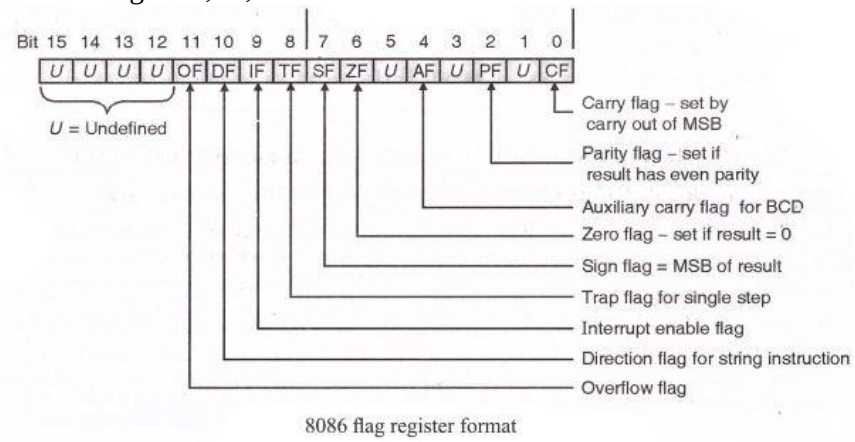
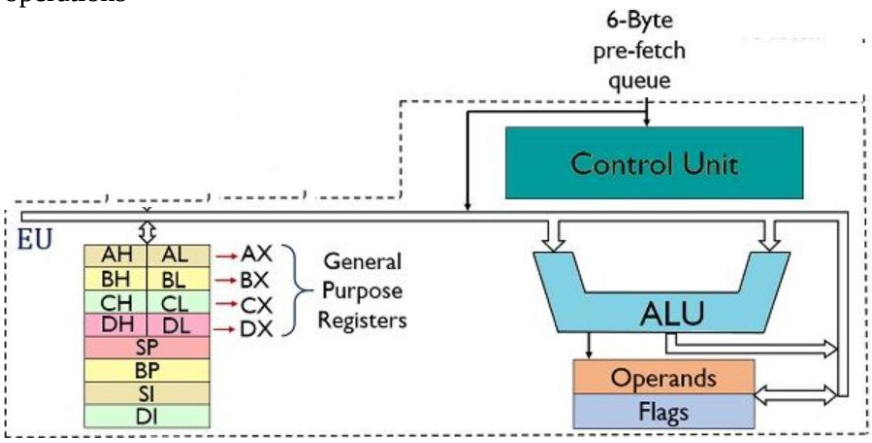
SLOT: C2

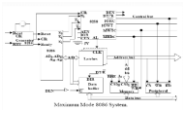
Programme Name & Branch : **M.Tech. (Integrated)-Computer Science and Engineering**
Course Code and Course Name : **CSI2006- Microprocessor and Interfacing Techniques**
Faculty Name(s) : **Paresh Sagar/ Arunkumar Chandrasekhar/Rajkishor Kumar/Sidharth Gautam/Vivek Rajpoot /Subhra Sankha Sarma**
Class Number(s) : **VL2025260501097**
Date of Examination : **29-Jan-2026**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes
  1. Explain the design aspects of a typical microprocessor and illustrate its capabilities.
  2. Practice and emulate assembly programs. To develop logic at assembly level for various operations.

Q. No	Question	M
1.	How does the EU (Execution Unit) decide which flags to update after execution of any instruction (Choose a suitable instruction)? Explain the EU with a suitable diagram.	10
	1. How does the EU (Execution Unit) decide which flags to update after instruction execution? Explain EU and Flag register with necessary diagrams The Execution Unit (EU) is responsible for: • Decoding instructions • Executing arithmetic and logical operations • Updating flags • Controlling internal data flow It does not directly access memory or I/O devices; instead, it requests such operations from the BIU. Main Components of the Execution Unit 1. Arithmetic Logic Unit (ALU): Performs: Arithmetic operations (ADD, SUB, INC, DEC), Logical operations (AND, OR, XOR, NOT), Generates results and conditions used to update flags 2. Instruction Decoder Decodes the instruction fetched by the BIU, Determines: Type of operation and addressing mode Which registers are involved , Generates control signals for execution 3. Control Unit Directs the sequence of operations inside the EU Controls: ALU operation, Register usage, Flag updating, Coordinates with BIU for memory/I/O access	

	4. Flag Register A 16-bit register, Stores status and control flags, Updated after instruction execution Flags include: Status flags: CF, PF, AF, ZF, SF, OF Control flags: TF, IF, DF  8086 flag register format 5. General Purpose Registers AX, BX, CX, DX . Used for: Temporary data storage, Arithmetic and logical operations 6. Pointer and Index Registers SP, BP, SI, DI, Used in: Stack operations, Indexed and based addressing modes ,String operations  	
2.	Describe the maximum mode of an 8086 microprocessor and explain the labelled timing diagram for a Memory Write cycle in the maximum mode, depicting all signals and timing states.	10



In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.

In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information .

In the maximum mode, there may be more than one microprocessor in the system configuration. The components in the system are same as in the minimum mode system.

The basic function of the bus controller chip IC8288, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE etc. using the information by the processor on the status lines.
The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.

It derives the outputs ALE, DEN, DT/R, MRDC, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.

AEN and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/PDEN output depends upon the status of the IOB pin.

INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.
IORC, IOWC are I/O read command and I/O write command signals respectively . These signals enable an IO interface to read or write the data from or to the address port.

The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.

All these command signals instructs the memory to accept or send data from or to the bus.

Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.
R0, S1, S2 are set at the beginning of bus cycle.8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.

In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4.

For an output, the AMWC or AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.

The status bit S0 to S2 remains active until T3 and become passive during T3 and T4.

If reader input is not activated before T3, wait state will be inserted between T3 and T4.

REG.NO.:

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT School of Electronics Engineering (SENSE)
CONTINUOUS ASSESSMENT TEST - 1
WINTER SEMESTER 2025-2026

SLOT: C2

3.	Find the physical memory address from where the 8086 accesses the data while executing & identify the addressing mode for the given instructions. [Given: DS =1000H, SS=2000H, ES 3000H, CS=8000H, BX=4000, SI= 5000H, DI=6000H, BP=7000H, AX=2021H, CX=4532H] a) MOV CX, [BP]-Index Addressing mode b) CMP BX, CX-register addressing mode c) MOV 2026H[SI][BX], AX , relative based index add mode d) MOV 10H[BX], AX, register relativ add mode e) DEC [5235H],direct memory addressing mode a) MOV CX, [BP]-**Index Addressing mode/ Based addressing** Effective Address = EA = BP = **7000H** Physical Address = 2000H × 10H + 7000H = 20000H + 7000H = **27000H** b) CMP BX, CX-**register addressing mode** No memory access (register to register) No physical memory address involved c) MOV 2026H[SI][BX], AX , **relative based index add mode/Based-Indexed** Effective Address = EA=2026H+SI+BX=2026H+5000H+4000H=**B026H** Physical Address = 10000H+B026H=**1B026H** d) MOV 10H[BX], AX, **register relativ add mode/ Register Relative** Effective Address = EA=10H+BX=**4010H** Physical Address = 1000H × 10H + 4010H = 10000H + 4010H = **14010H** e) DEC [5235H], **Direct memory addressing mode** Effective Address = EA = **5235H** Physical Address = 1000H × 10H + 5235H = 10000H + 5235H = **15235H**	10
4.	**a.** Show the status of the CF, AF, PF and OF flags after the addition of 939b H and B7B7 H in the following Instructions: MOV AX, 939B H; ADD AX B7B7 H; NOP	5

Step 1: Perform the addition

$$939B_H + B7B7_H$$

Low byte:

$$9B_H + B7_H = 152_H$$

→ Result = 52H, Carry = 1

High byte:

$$93_H + B7_H + 1 = 14B_H$$

→ Result = 4BH, Carry out = 1

Final Result

$$AX = 4B52_H$$

Flag Status

Carry Flag (CF)

Carry out of MSB occurred.

CF = 1

Auxiliary Carry Flag (AF)

Check lower nibble:

$$B_H + 7_H = 12_H$$

Carry from bit3 → bit4 occurred.

AF = 1

Parity Flag (PF)

Parity of lower byte (52H):

$$52_H = 0101\ 0010$$

Number of 1s = 3 (odd)

PF is set for even parity.

PF = 0

Overflow Flag (OF)

Both operands were negative (MSB=1) and result is positive (MSB=0).

Signed overflow occurred.

OF = 1

Flag	Status
CF	1
AF	1
PF	0
OF	1

b. Assume that two packed BCD numbers 73H and 29 H are stored in register AL and CL. Write an ALP to find their valid difference and what is the content of AL and status of CF after execution of above ALP? — 5

```
Assume AL = 73H, CL = 29H

MOV AL, 73H
MOV CL, 29H

SUB AL, CL   ; AL = 73H - 29H = 4AH (binary result)
DAS          ; Decimal Adjust after Subtraction -> valid BCD
HLT
```

	Step-by-step result 1. **After** `SUB AL, CL` $73_{BCD} - 29_{BCD} = 4A_H$ (not valid BCD since A > 9) 2. **After** `DAS` • Lower nibble A (>9) → subtract 06H $4A_H - 06_H = 44_H$ • Upper nibble now valid (4 ≤ 9), no borrow → no 60H subtraction **Final Answer** • **Content of AL = 44H** (valid BCD for decimal 44) • **Carry Flag (CF) = 0** (no borrow)	

	MOV AL, 73H ; Load first packed BCD number
	MOV CL, 29H ; Load second packed BCD number
	SUB AL, CL ; Subtract CL from AL
	DAS ; Decimal adjust after subtraction
	SUB 73H – 29H = 4AH
	Lower nibble A (> 9) → subtract 06H
	4AH – 06H = 44H	

5.	.MODEL SMALL
	.DATA
	NUM1 DW 1234H
	NUM2 DW 0111H
	RESULT DW 0000H
	.CODE
	MAIN PROC
	MOV AX, @DATA
	MOV DS, AX
		10

```
    MOV AX, NUM1
    SUB AX, NUM2
    DAS
    MOV CL, 01H
    ROR AX, CL
    MOV RESULT, AX
    MOV AH, 4CH
    INT 21H
MAIN ENDP
END MAIN
```

Analyze Register Status after the execution of the above assembly language program (ALP).

After `SUB AX, NUM2`

```ini
AX = 1123H
```

(1234H − 0111H = 1123H)

After `DAS`

AL = 23H is already valid BCD → no adjustment

```ini
AX = 1123H
```

After `ROR AX, CL`

`CL = 01H` → rotate right by 1 bit

```yaml
AX before = 1123H = 0001 0001 0010 0011₂
AX after  = 8891H = 1000 1000 1001 0001₂
```

```ini
AX = 8891H
```

Final Register Contents

AX

```ini
AX = 8891H
```

BX

BX is **never used or modified** in your program.

So:

```ini
BX = unchanged (whatever value it had before execution)
```

```ini
BX = 0000H
```

```
CX = 0001H
```

(CH = 00H, CL = 01H)

```
After SUB AX,NUM2  → AX = 1123H
After DAS          → AX = 1123H
After ROR AX,CL    → AX = 8891H

Final:
AX = 8891H
BX = 0000H
CX = 0001H
```

****