

Question Bank (K scheme)

Unit Test: II

Name of subject: MICROPROCESSOR

Subject code: 22415 Course: CO

Semester: IV

1. Write any four-bit manipulation instructions of 8086.

Ans. **NOT** – Used to invert each bit of a byte or word.

AND – Used for adding each bit in a byte/word with the corresponding bit in another byte/word.

OR – Used to multiply each bit in a byte/word with the corresponding bit in another byte/word.

XOR – Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.

2. Define procedure and write its syntax

Ans.

Procedure: A procedure is group of instructions that usually performs one task. It is a reusable section of a software program which is stored in memory once but can be used as often as necessary.

A procedure can be of two types. 1) Near Procedure 2) Far Procedure

Syntax :- Procedure can be defined as

Procedure_name PROC

Procedure_name

ENDP

For Example

Addition PROC near

Addition ENDP

3. State any two difference between TEST and AND instructions.

Ans.

TEST	AND
This instruction logically ANDs the source with the destination but the result is not stored anywhere.	This instruction logically ANDs the source with the destination and stores the result in destination. .
e. g .TEST BL ,CL The result is not saved anywhere.	e.g. AND BL , CL The result is saved in BL register

4. Write an ALP to add two 8 bit numbers.

Ans.

.model small

.data

a db 06h

```
b db 12h
ends
.code
start:
mov ax,@data
mov ds,ax
mov al,a
mov bl,b
add al,bl
int 3
ends
end start
```

5. State the use of DAA instruction in BCD addition.

Ans. The DAA (Decimal Adjust after Addition) instruction makes the result in Packed BCD from after BCD addition is performed. It works only on AL register.

6. Define immediate addressing mode with suitable example

Ans.

An instruction in which 8 bit or 16 bit operand (data) is specified in instruction itself then the addressing mode of such instruction is called as immediate addressing mode.

Eg.

MOV AX,7120H

7. State the use of STC and CMC instruction of 8086.

8. Write an ALP to perform addition of two 16 bit BCD numbers.

Ans.

```

.model small
.data
    num1    dw    9999h
    num2    dw    9999h
    res_lsw  dw    0
    res_msb  db    0
.code
    mov ax,@data                ;Initialise data segment
    mov ds,ax
    mov al,byte ptr num1        ;Add LSB first
    add al,byte ptr num2        ;convert result to BCD
    daa
    mov byte ptr res_lsw,al      ;Store LSB of result
    mov al,byte ptr num1+1
    adc al,byte ptr num2+1      ;Add MSB next
    daa                          ;Convert result to BCD
    mov byte ptr res_lsw+1,al    ;Store result
    jnc exit                    ;Check result > 16bit
    inc res_msb                 ;if yes, increment res_msb by 1
it : ends
end

```

9. Describe the function of the following instructions:

(i) DAA (ii) CMP (iii) ADC (iv) JNC

Ans.

1) DAA: Decimal adjust after addition

This instruction is used to make sure the result of adding two packed BCD numbers is adjusted to be a legal BCD number. The result of the addition must be in AL for DAA to work correctly. If the lower nibble in AL after an addition is greater than 9 or AF was set by the addition, then the DAA instruction will add 6 to the lower nibble in AL. If the result in the upper nibble of AL is now greater than 9 or if the carry flag was set by the addition or correction, then the DAA instruction will add 60H to AL.

Let AL = 59 BCD, and BL = 35 BCD

ADD AL, BL AL = 8EH; lower nibble > 9, add 06H to AL

DAA AL = 94 BCD, CF = 0

Let AL = 88 BCD, and BL = 49 BCD
ADD AL, BL AL = D1H; AF = 1, add 06H to AL
DAA AL = D7H; upper nibble > 9, add 60H to AL
AL = 37 BCD, CF = 1

2) CMP: Compare

This instruction compares the source operand, which may be a register or an immediate data or a memory location, with a destination operand that may be a register or a memory location.

Example: -

```
CMP BX, 0100H
CMP AX, 0100H
CMP [5000H], 0100H
CMP BX, [SI]
CMP BX, CX
```

3) ADC: Add with Carry

ADC Destination, Source

This instruction performs the same operation as ADD instruction, but adds the carry flag to the result.

Example: -

```
ADC 0100H
ADC AX, BX
ADC AX, [SI]
ADC AX, [5000]
ADC [5000], 0100H
```

4) JNC: - Stands for 'Jump if Not Carry'

It checks whether the carry flag is reset or not. If yes, then jump takes place, that is: If CF = 0, then jump.

ADD AL, BL Add two bytes

JNC NEXT If the result within acceptable range, continue

10. Write an ALP to find largest number in array of 5 elements.

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
```

11. Differentiate between Procedure and Macros.

Ans.

Procedure	Macro
Procedures are used for large group of instructions to be repeated	Procedures are used for small group of instructions to be repeated.
Object code is generated only once in memory.	Object code is generated every time the macro is called.
CALL & RET instructions are used to call procedure and return from procedure.	Macro can be called just by writing its name.
Length of the object file is less	Object file becomes lengthy.
Directives PROC & ENDP are used for defining procedure.	MACRO and ENDM are used for defining MACRO
Directives More time is required for its execution	Less time is required for it's execution
Procedure can be defined as Procedure_name PROC ----- ----- Procedure_name ENDP	Macro can be defined as MACRO-name MACRO [ARGUMENT,..... ARGUMENT N] ----- ----- ENDM
For Example Addition PROC near ----- Addition ENDP	For Example Display MACRO msg ----- ENDM

12. Write an ALP to find length of string.

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
```

```
JMP
NEXT EXIT: MOV LEN,CL
MOV AH,4CH
INT 21H
CODE ENDS
```

13. Write an ALP to count number 1' in 8 bit number.

Ans.

```
DATA SEGMENT
N DB 12H
Z DB 0
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA, CS:CODE
START:
MOV DX,DATA
MOV DS,DX
MOV AL, N
MOV CL,04
NEXT: ROL AL,01
JNC ONE
INC Z
ONE: LOOP NEXT
HLT
CODE ENDS
END START
```

14. Explain any four Addressing Modes of 8086.

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.

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]
```

15. Write an ALP to multiply two 16 bit signed numbers.

Ans.

```
.model small
```

```
.data
```

```
A db 2222h
```

```
B db 1111h
```

```
Ends
```

```
.code
```

```
Mov ax,@data
```

```
Mov ds,ax
```

```
Mov AX,a
```

```
Mov BX,b
```

```
IMul BX
```

```
Int 03h
```

```
Ends
```

```
End
```

16. Write a MACRO to perform 32 bit by 16 bit division of unsigned numbers.

Ans.

```
.model small
```

```
Div1 macro no1,no2
```

```
mov ax,no1
```

```
div no2
```

```
endm
```

```
.data
```

```
num1 dw 12346666h
```

```
num2 dw 2222h
```

```
.code
```

```
mov ax,@data
```

```
mov ds,ax
```

```
div1 num1,num2
```

ends
end