

Name: _____ Roll Number: _____ Section: _____

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Computer Organization and Assembly Language	Course
Program:	BS (Computer Science)	Code: EE229
Duration:	60 Minutes	Semester: Fall 2021
Paper Date:	21-Oct-2021	Total Marks: 30
Section:	All	Weightage: 15
Exam:	Midterm I	Page(s): 6
		Section: _____
		Roll No: _____

Instruction/Notes:

- Exam is Open book, Open notes.
- Properly comment your code.
- You CANNOT use an instruction NOT taught in class.
- If there is any ambiguity, take reasonable assumption. Questions during exam are not allowed.
- All other rules pertaining to examinations as per NUCES policy apply.
- Write your answer in the space provided. You can take extra sheets BUT they WON'T BE ATTACHED WITH THE QUESTION PAPER OR MARKED.

Question 1 [20 Marks]: Short Questions

- i. [5 marks] The value of Code Segment (CS) and Stack Segment (SS) Register is 4582H while the value of different registers is as follows:

BX: 2025h, IP: 0580h, DI: 4247h, BP: 4700h, SI: FFFFh

Write the physical address of the following memory locations. Also point out which type of wraparound is there if occurred, segment or whole memory?

	Memory Location	Physical Address in hex	Wraparound Type if occurred
a	[cs:bx + si]	47844	Within Segment
b	[bp + di + 10]	4E171	No wraparound occurred

(Show your working in this box:)

- a. Effective Address = bx+si = 2025 + FFFF = 12024 = 2024 (Wrapped around)
 Physical Address = CS * 10h + Offset (Effective Address) = 45820 + 02024 = 47844
- b. Effective Address = bp + di + 10 = 4700 + 4247 + 000A = 8951
 Physical Address = SS * 10h + Offset (Effective Address) = 45820 + 08951 = 4E171

Name: _____

Roll Number: _____

Section: _____

ii.

[2 Marks] Point out the addressing modes in each of the following instructions.

		Mode
a.	mov ax, [bx+si+100]	Base+Index+Offset Mode
b.	mov ax, [0500]	Direct mode
c.	mov al, [bp+si]	Base + Index Mode
d.	mov as, [si+2000]	Indexed Register Indirect Mode

iii.

[2 Marks] Mark each of these instructions Valid or Invalid. In case of Invalid, give one-line reason.

		Valid/ Invalid	Reason
a.	add 34BFh, bx	Invalid	
b.	mov cs, ds	Invalid	
c.	mov cs, 2345	Invalid	
d.	mov [num1], ax	Valid	

iv.

[3 Marks] What is the value of OF, CF and SF at the end of the following code? Is the jump taken or not?

```
[org 0x100]
    jmp start
num1: db 1Bh, 27h, 4Ch, 8Eh, 0h

start:
    mov al,[num1]
    mov bl,[ num1+1]
    add al, bl
    mov bl, [num1+2]
    add al, bl
    mov bl, [num1+3]
    add al, bl
    mov [num1+4], al

    mov ax, 0x4c00
```

1. Yes, the jump is taken.
2. OF = 0
3. SF = 1
4. CF = 1

Name: _____ Roll Number: _____ Section: _____

Int 0x21	
----------	--

- v. [3 Marks] By considering the data given in Q1(iv) (which is stored in memory label num1), you have to complete the following table and show the data placement in memory.

Hint: Remember memory storage is shown in hex numbers here.

	0	1	2	3	4
DS:0103	1B	27	4C	8E	0

- vi. [5 Marks] For the code given below, write the decimal values stored in memory labels var1 and var2 after the execution of the program. You also have to explain what this program is doing in one line.

```
[org 0x100]
jmp start

array: dw -1, 7, 9, -2, 2, 0
var1: dw 0
var2: dw 0

start: mov dx, [array]
       mov bx, array
       mov [var2], dx
       mov [var1], dx
       mov cx, 6

A1:   mov dx, [bx]
       cmp dx, [var1]
       jge A2

       mov [var1], dx

A2:   cmp dx, [var2]
       jle A3

       mov [var2], dx

A3:   add bx, 2
       sub cx, 1
       jnz A1

       mov ax, 0x4c00
       int 0x21
```

Smallest value will be stored in label var1, and largest value in label var2.

Name: _____ Roll Number: _____ Section: _____

Question 2 [10 Marks]: Write an assembly language program to perform pairwise scan operation on an array such that:

Case 1: If second element of the pair is even, then multiply 1st and 2nd element through bit manipulation and store the result in place of the first element.

Case 2: If second element of the pair is odd, then add 1st and 2nd element and store the result in the location of the first element.

Case 3: If the array contains odd number of elements, then save the last element as it is.

Assume that last element of the array is -1 an indicator to stop the array iteration, as you don't know how to input array. Just assume generic array with end element -1.

Hint: You can find even and odd number by bit manipulation. See a sample run below for detail.

Sample Run:

Example 1, even sized array (excluding the last element)	Example 2, odd sized array (excluding the last element)
Input Array: 3, 5, 10, 9, 12, 16, -1	Input Array: 3, 5, 10, 9, 12, 16, 23, -1
Output Array: 8, 5, 19, 9, 192, 16, -1	Output Array: 8, 5, 19, 9, 192, 16, 23, -1

Name: _____

Roll Number: _____

Section: _____

Write your code below

[org 0x0100]

jmp start

data: dw 1, 8, 4, 2, 3, -1

start: mov bx, 0

loop1: cmp word [data+bx], -1 ;check to stop loop
jz end

 cmp word [data+bx+2], -1
 jz end

 mov ax, [data+bx+2] ; check next element either even or odd
 shr ax, 1
 jc odd

even: mov cl, 8 ; multiply by using bit operations
 mov ax, [data+bx]
 mov dx, [data+bx+2]
 mov word [data+bx], 0 ; set result on first place to zero

chkbit: shr dx, 1
 jnc skip
 add [data+bx], ax

skip: shl ax, 1
 dec cl
 jnz chkbit
 jmp next

odd: mov ax, [data+bx+2] ; add elements in case of odd
 add [data+bx], ax

next: add bx, 4 ; iterate to next elements
 jmp loop1

end: mov ax, 0x4c00
 int 0x21

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Computer Organization and Assembly Language BS(Computer Science)	Course Code:	EE229
Program:		Semester:	Sp-2021
Duration:	3 hours	Total Marks:	70
Paper Date:	08-Jul-2021	Page(s):	
Section:	ALL	Section:	1
Exam:	Final Exam	Roll No:	

Instruction/Notes:

Answer in the space provided

You can ask for rough sheets but they will not be graded or marked

In case of confusion or ambiguity, make a reasonable assumption.

Questions are not allowed

This is open book and open notes paper

Good luck!

16 PA
16
10

9345

9345

9435

9435

93450

93450

93450

9345

9345

Q1 (a) (2 Points): What is the Effective and Physical address generated by the following instruction.
Given CS = DS = SS = BX = BP = SI = DI = 9345h

* BP+SI+9435h

A A BF

Effective Address = BABFh -
Physical Address = 9EFOFh

Ph = EA * 8 / 10

①
93450 9345
GEE
OBABF + BABF
9EFOF

93450
OBABF
9EFOF

Q1 (b) (2 Points): What will be the value of cx, ax, si, and bx at the end of the following code?

```
[org 0x0100]
jmp start
num: dw 1, 2, 5, 0, 7, 4, 0, 3, 1, 6
start:
    mov bx, 0
    xor si, si
    mov cx, 8
g1:
    mov ax, [num+si]
    cmp ax, 0
    je g2
    add bx, ax
    add si, 2
    jmp g1
g2:
    mov ax, 0x4c00
    int 0x21
```

Answer
ax = 4c or 0
bx = 8
cx = 8
si = 6

Roll Number:

Section:

Q1 (c) (2 Points): Given below the listing file of a code. What is the size of the code i.e. .com file?

```

1
2 00000000 E91400 [org 0x100]
3 00000003 00000000000000000000000000000000 jmp start
4 0000000C 000400000003000100 num: dw 1, 2, 5, 0, 7, 4, 0, 3, 1, 6
5 00000015 060021
6
7 00000017 31DB2
8 00000019 31F62
9 0000001B B908003
10
11 0000001E 8B84[0300] *Q4
12 00000022 3D00003
13 00000025 74082
14 00000027 01C32
15 00000029 81C6020084
16 0000002D E2EF2
17
18 0000002F B8004C3
19 00000032 CD212

```

start:
 xor bx, bx
 xor si, si
 mov cx, 8
 g1:
 mov ax, [num+si]
 cmp ax, 0
 je g2
 add bx, ax
 add si, 2
 loop g1
 g2:
 mov ax, 0x4c00
 int 0x21

Answer

52 bytes

Q1 (d) (2 Points): Write values of Overflow, Carry, Auxiliary Carry and Zero Flag for each of the following instruction.

Instruction	OF	CF	AF	ZF
mov ax, 0xEF mov bx, 0x11 add ax, bx	0	0	1	0

$$\begin{array}{r} \textcircled{1} \\ \begin{array}{r} \text{EF} \\ + \text{11} \\ \hline \text{100} \end{array} \end{array}$$

$$\begin{array}{r} \textcircled{0} \\ \begin{array}{r} \text{0 F5} \\ - \text{52} \\ \hline \text{147} \end{array} \end{array}$$

$$\begin{array}{r} \frac{1}{16} \\ \overline{20} \\ -16 \\ \hline 4 \end{array}$$

School of Computer Science

S	2	1	1	1	1	1	1
8 4 2 1	8 4 2 1						

F	8	4	2	1	8	4	2	1
	1	1	1	1	0	1	0	1
					0	1	0	1

Page 2 of 14

Roll Number: _____

Section: _____

Solution

National University of Computer and Emerging Sciences, Lahore Campus



Instruction/Notes:

Course: COAL
 Program: BS(Computer Science)
 Duration: 3 Hours
 Paper Date: 22-02-2021
 Section: All (Your section _____)
 Exam: Final

Course Code: EE229
 Semester: Fall 2020
 Total Marks: 110
 Weight: 45%
 Page(s): 11
 Roll. No.

This is an open note/book exam. All the answers should be written in provided space on this paper. Rough sheets can be used but will not be collected and checked. In case of any ambiguity, take reasonable assumption. Questions during exam are not allowed.
ATTEMPT ALL QUESTIONS UNLESS GIVEN EXPLICIT INSTRUCTIONS FOR YOUR SECTION.
SHARING CALCULATOR IS NOT ALLOWED.

Question 1: Short Questions [10 x 5 = 50 Marks]

- I. What will be the content of memory (in HEX) before and after the execution of the following code?

```
[org 0x0100]
jmp start
num1: db 0xA
dw 0x1234
dd 0xABCDEFGH09

start:
  mov ax, [num1+5]    CD
  add ax, [num1+2]    CD+12
  mov [num1], ax
  mov ax, 0x4c00 ; terminate program
  int 0x21
```

Memory Configuration BEFORE Execution:

Address	Num1+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
Content	0A	34	12	09	EF	CD	AB			

Memory Configuration AFTER Execution:

Address	Num1+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
Content	DF	34	12	09	EF	CD	AB			

- II. Write Assembly language code for the following

if dx <= cx, ax = 1, else ax = 2

```
cmp dx, cx
ja label1
mov ax = 1
jmp end-condition
```

label1:
 mov ax, 2
end-condition

III. Find the values of Sign, Carry, Overflow, and Zero flag given the values of registers and operations.

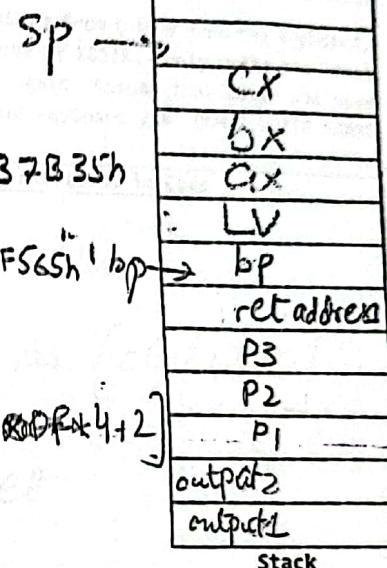
	CF	OF	SF	ZF
MOV AL, 10 MOV BL, 20 ADD AL, 10 SUB AL, BL MOV AL, 66H MOV BL, 1AH ADD BL, AL	0	0	0	1
	0	1	1	0

IV. A function takes three parameters P1, P2, P3, returns two values Output1, Output2, declares one local variable (local1), and saves AX, BX and CX registers. What will be the configuration of the stack for this function after pushing all these variables and registers? Fill in the given stack. Also specify where BP and SP should be pointing?

V. Fill in the blanks

- If SP = 1735h and SS:SP = 37B35h, $\text{SS} + \text{SP} = 01735h + 37B35h = 37B35h$
SS = 0x 3640.
- If SS = FE07h and SS:BP = FF565h, $\text{FE070h} + \text{BP} = \text{FF565h}$ bp → BP = 0x 1475
- A parallel port is mapped to interrupt number 0x0F, address of its service routine (handler) can be found at 0 : 0 + 0Fx4. 0 (0x0D8+4+2)
- Timer tick comes 18.2 times per second.

Solution of Part (IV)



VI. The following code is trying to copy the arrays 0th to 9th element to 1st to 10th elements. For example if array initially 1,2,3,4,5,6,7,8,9,10,11,12 after code runs it should be 1,1,2,3,4,5,6,7,8,9,10,12 But there is a mistake in this code. Identify and correct the mistake. (Hint: Source and Destination are overlapping)

Code with Mistakes	Corrections (Write Correct Lines Only)
<pre> 00 push ds 01 pop es 02 Mov cx, 10 03 Mov si, array 04 Mov di, array+1 05 cld 06 Rep movsb 07 mov ax, 4C00h 08 int 21h 09 array: db 1,2,3,4,5,6,7,8,9,10,11,12 </pre>	<pre> → mov si, ax+9 → mov di, ax+array+10 → STD. </pre>

Roll Number: _____

Section: _____

- VII. Dry run the following code and find the values of given registers and arrays. Also, write in one line what this code is doing. Assume address of Array is 103h

```

01 JMP start:
02     array: db 5,6,3,7,8,9,1
03     len_of_array: db 7
04 start:
05     mov cx, [len_of_array]
06     push ds
07     pop es
08     mov si, array
09     mov di, array
10     cld
11     loop1:
12         lodsb
13         add al, 30
14         stosb
15     loop
    
```

Array: 35 36 33 37 38 39 31

SI: 10A

DI: 10A

What is this code doing?

Adding 30 to all elements
of array

- VIII. The following keyboard custom ISR is trying to block the numbers from 0 to 9 from the keyboard, the rest of the keys should work as before. Complete the code of KBSIR. (Only write the lines that are to be added in kbISR, don't write the whole code again. Assume that kbsir has been hooked by start and oldisr variable saves the values of old keyboard ISR. DON'T WRITE START CODE)

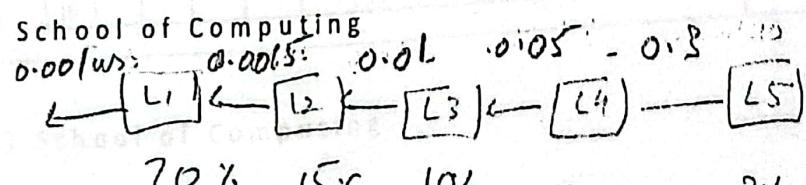
Code	Additions in Code
<pre> 01 oldisr: dd ; stores old isr 02 kbisr: 03 push ax 04 ;read a char from keyboard 05 ;in ax, 0x60; ascii in ah and scan code in 06 ;al 06 cmp ah, 30h; ascii of 0 07 JL exit 08 cmp ah, 39h ; ascii of 9 09 JG exit 10 exit: 11 mov al, 0x20 12 out 0x20, al 13 pop ax </pre>	<p>→ jmp far [oldISR]</p> <p>→ iret</p>

- IX. [For All Sections Except Section E] Consider 5 cache levels. It takes 0.001μs to read from cache L1, 0.0025μs to read from L2, 0.01μs from L3, 0.05 μs from L4 and 0.3 μs from L5. Data is found in L1 70% of time, 15% of the time in L2, 10% of the time in L3, 3% of the time in L4 and 2% of the time in L5. What is the average access time? (μs stands for micro second 1 μs = 10⁻⁶ s)

$$\begin{aligned}
 & (0.001) * 70\% \\
 & + (0.001 + 0.0015) * 15\% \\
 & + (0.001 + 0.0015 + 0.0025) * 10\% \\
 & + (0.001 + 0.0015 + 0.0025 + 0.005) * 3\% \\
 & + (0.001 + 0.0015 + 0.0025 + 0.005 + 0.03) * 2\%
 \end{aligned}$$

$\rightarrow 100 = 0.01145 \mu\text{s}$

FAST School of Computing



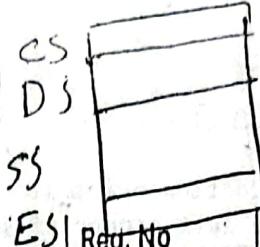
70% 15% 10% 3% 2%

Page 3 | 11



PA of CS = seg of CS x 10
 Segmented offset add DS

LA of CS = CS : IP
 Section: Antu
 Exam: Sessional II



IP
 BX, DI, ST
 SP, BP
 DT

Instruction/Notes:

This is an open note/book exam. All the answers should be written in provided space on this paper. Rough sheets can be used but will not be collected and checked. In case of any ambiguity, take reasonable assumption. Questions during exam are not allowed.

Question 1: Find the physical address of the following given the values of registers and variables.
 You answers should be in hex (Show calculations on given space, no marks without calculations) [5 marks]

CS= FFF3h DS= 00BFh SS= 0A77h ES=F810h num= 8954h
 BX= FF20h IP= 0020h SP= EE00h SI=0010h

1 point each

If you have not shown complete answer, you will lose mark, as it's needed to show concepts of use

$$1. [num] PS : Num = 00BFO + 8954 = 09544$$

$$2. [BX+0200h] 00BFO + (FF20 + 200) = 00BFO + 0120 = 00D10$$

$$3. [CS:BX+SI] FFF30 + (FF20 + 10h) = FFF30 + FF30 = 0FE60$$

$$4. [IP] FFF30 + 0020h = FFF50h$$

$$5. [SP] 0A770 + EE00 = 19570$$

Mov Ax,[BX][SI] → DS

PA = DS AND [BX+SI]

Question 2: The following code is to print a blinking "7" in red color with blue background on the 5th row and 10th column of screen. Fill in the blanks with correct screen offset and word is to be written on it. [5 marks]

Mov ax, 0xb800

Mov es, ax

Mov di, 820

Mov [es:di], 9C37 or 9437

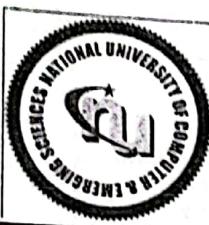
Show calculations of Q2 here

FAST School of Computing

1 di
 1 '7'
 1 blank
 1 BG color
 1 FG color

Page 1 | 6

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Computer Organization and Assembly Language BS(Computer Science)	Course Code: Semester: Total Marks: Page(s): Section: Roll No:	EE229 Fall 2019 95 12
Program:			
Duration:	3 hours		
Paper Date:	27-Dec-2019		
Section:	ALL		
Exam:	Final Exam		

Instruction/Notes:

Answer in the space provided
You can ask for rough sheets but they will not be graded or marked
In case of confusion or ambiguity, make a reasonable assumption.
Questions are not allowed
This is open book and open notes paper

Good luck!

Q1 (2 Points): What is the Effective and Physical address generated by the following instruction.
Given CS = DS = SS = BX = BP = SI = DI = 9543h

- BP+SI+9543h

Effective Address =Bfc9

Physical Address A13F9

$$\begin{array}{l} \text{Effective} = \\ \quad \underline{\begin{array}{r} 9543 \\ 9543 \\ 9543 \end{array}} \\ \quad \underline{\text{BFC9}} \end{array}$$

P.A =

$$\begin{array}{r} 95430 \\ 0BFC9 \\ \hline A13F9 \end{array}$$

Q2 (4 Points): What will be the value of cx, ax, si, and bx at the end of the following code?

```
[org 0x0100]
jmp start
num: dw 5, 9, 7, 4, 0, 3, 8, 2, 1, 6
start:
xor bx, bx
xor si, si
mov cx, 10
f1:
mov ax, [num+si]
cmp ax, 0
je f2
add bx, ax
add si, 2
loop f1
f2:
mov si, 0
mov ax, 0x4c00
int 0x21
```

Answer

Ax = 4c00 or 0
Bx = 25
Cx=6
Si = 0

Roll Number: _____

Section: _____

Q3 (4 Points): Which of the following instructions are valid and which are invalid, give reason?

(a) mov db[num], 55h
invalid db is not correct syntax

(b) mov bx, [num+dx] invalide dx cannot be used as offset

(c) mov [num+si+di], ax invalid, two index registers cannot added

(d) mov bx, [num+bp+si] valid

Q4 (2 Points): Given below the listing file of a code. What is the size of the code i.e. .com file?

1		[org 0x0100]
2	00000000 E91400	jmp start
3	00000003 050009000700040000-	num: dw 5, 9, 7, 4, 0,
4	0000000C 000300080002000100-	3, 8; 2, 1, 6
5	00000015 0600	
6		
7	00000017 31DB	start:
8	00000019 31F6	xor bx, bx
9	0000001B B90A00	xor si, si
10		mov cx, 10
11	0000001E 8B84[0300]	f1:
12	00000022 3D0000	mov ax, [num+si]
13	00000025 7408	cmp ax, 0
14	00000027 01C3	je f2
15	00000029 81C60200	add bx, ax
16	0000002D E2EF	add si, 2
17		loop f1
18	0000002F BE0000	f2:
19	00000032 B8004C	mov si, 0
20	00000035 CD21	mov ax, 0x4c00
		int 0x21

Answer

55 bytes or 37h

Roll Number: _____

Section: _____

Q5 (4 Points): Write values of Overflow, Carry, Auxiliary Carry and Zero Flag for each of the following set of instructions.

Instruction	OF	CF	AF	ZF
mov al, 0xDB mov bl, 0x25 add al, bl	0	1	1	1
mov al, 0xDB mov bl, 0x25 add ax, bx	0	0	1	0

Q6 (2 Points): Which of the following code converts a number to its character equivalent? Assume the following data declaration number: db 5

- (a) mov ax, [number]
add ax, 30
- (b) mov al, 30h
add al, [number]
- (c) mov ax, db[number]
add ax, 30

Q7 (4 Points): Write output of following code, Also mention row number and column number of screen where data will print. If there is any junk character in output, then use '#' symbol to represent it.

```
[org 0x0100]

Start:
jmp start
message: db 'Hello World', 'World is so
big'

start:
mov ah, 0x13
```

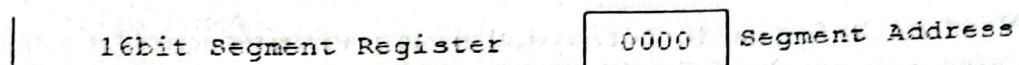
Row number: 23

Column number: 10

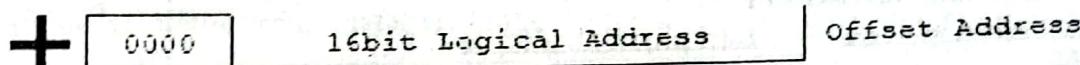
Output: ##Hello Wor

(white/gray background and black foreground and blinking)

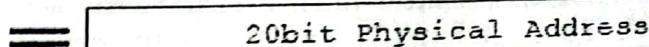
15-----0



15-----0



19-----0



a. 1DDD:0436

1DDD0 segment

+00436 offset

=1E206 physical memory address

Hexadecimal	Decimal
A	10
B	11
C	12
D	13
E	14
F	15

b. 1234:7920

12340

+07920

=19C60

c. 74F0: 2123

74F00

+02123

=77023

d. 0000:6727

00000

+06727

=06727

e. FFFF: 4336

FFFF0

+04336

=104326

f. 1080:0100

10800

+00100

=10900

g. AB01: FFFF

AB010

+0FFFF

=BB00F

19. What are the first and the last physical memory addresses accessible using the following segment values?

First 0x0000 and last 0xFFFF

First physical address Last physical address

a. 1000

$$\begin{array}{rcl} 0x10000 & & 0x10000 \\ +0x00000 & & +0x0FFF \\ \hline =0x10000 & & =0x1FFFF \end{array}$$

b. 0FFF

$$\begin{array}{rcl} 0x0FFF0 & & 0x0FFF0 \\ +0x00000 & & +0x0FFF \\ \hline =0x0FFF0 & & =0x1FFEF \end{array}$$

c. 1002

$$\begin{array}{rcl} 0x10020 & & 0x10020 \\ +0x00000 & & +0x0FFF \\ \hline =0x10020 & & =0x2001F \end{array}$$

d. 0001

$$\begin{array}{rcl} 0x00010 & & 0x00010 \\ +0x00000 & & +0x0FFF \\ \hline =0x00010 & & =0x1000F \end{array}$$

e. E000

$$\begin{array}{rcl} 0xE0000 & & 0xE0000 \\ +0x00000 & & +0x0FFF \\ \hline =0xE0000 & & =0xEFFFF \end{array}$$

Note:

- That we have added Base & Offset addresses to calculate Physical addresses. Moreover also notice that we placed a Zero in Base & Offset address to produce a 20-bit address.
- I have used hexadecimal representation as shown by 0x used in the beginning of a number.

20. Write instructions that perform the following operations.

a Conv BL into CL

What is the difference between DATA LABEL and CODE LABEL?

Data Label is the label that we use to define data as we defined memory locations num1,num2etc in our programs. Code Label is the label that we have on code as we see in the case of conditional jump (Label 11) and is normally used for loop control statements.

2. List the seven addressing modes available in the 8088 architecture.

1. Direct
2. Base register Indirect
3. Indexed register Indirect
4. Base register indirect + offset
5. Index register indirect +offset
6. Base+index+offset

3. Differentiate between effective address and physical address.

The effective address is the address generated by the program, after all, transformations, such as index registers, offsets, addressing mode, etc. have been made. The physical address is the address generated by the hardware, after performing whatever lookups through the page table, etc. have been made. The effective address, or a virtual address, is the concern of the program. The physical address, or real address, is the concern of the operating system.

4. What is the effective address generated by the following instructions? Every instruction is independent of others. Initially BX=0x0100, num1=0x1001, [num1]=0x0000, and SI=0x0100

- a. mov ax, [bx+12]
- b. mov ax, [bx+num1]
- c. mov ax, [num1+bx]
- d. mov ax, [bx+si]

- a. $bx+12 = 0x0100 + 0xc = 0x010c$
- b. $bx+num1 = 0x0100 + 0x1001 = 0x1101$
- c. $num1+bx = 0x1001 + 0x0100 = 0x1101$
- d. $bx+si = 0x0100 + 0x100 = 0x0200$

5. What is the effective address generated by the following combinations if they are valid. If not give reason. Initially BX=0x0100, SI=0x0010, DI=0x0001, BP=0x0200, and SP=0xFFFF

- a. bx-si
- b. bx-bp
- c. bx+10

Physical Address Calculation:

As

SI = FFFF

BX = FFFF

Segment : DS = FFFF

SI+BX= FFFF+FFFF= FFFE= offset address

Physical address = segment x 10 + offset address

= FFFF0+FFFE

= 0FFEE

X4C00
JW.O

9. Write instructions to do the following.

a. Copy contents of memory location with offset 0025 in the current data segment into AX.

b. Copy AX into memory location with offset 0FFF in the current data segment.

c. Move contents of memory location with offset 0010 to memory location with offset 002F in the current data segment.

Solution:

a. mov SI, 0x0025

mov AX, [SI]

b. mov [DI], 0x0FFF

mov [DI], AX

c. mov DI, 0x002F

mov SI, 0x0010

mov AX, [SI]

mov [DI], AX

10. Write a program to calculate the square of 20 by using a loop that adds 20 to the accumulator 20 times.

Solution:

[org 0x0100]

mov bx,20

mov cx,20

mov ax,0

l1:

add ax, bx

sub cx, 1

jnz l1

mov [total], ax