FRONT Page

NAME Rafi ULLAH

ID 14283

Depestment BS(CS) 5th semester

Subject Assembly language

Q No 1:

Solve each of The Following.

Part: 1:

$(64)_{10} = (?)_2$

2	64
2	32 - 0
2	16 - 0
2	8 - 0
2	4 - 0
2	2 - 0
	1 - 0

$\Rightarrow (1000000)_2 = (64)_{10}$ Ans

Q No: 1

Part 2: $(011111111)_2 = (?)_{10}$

$\Rightarrow (011111111)_2 = (?)_{10}$

$\Rightarrow (0\times2)^8 + (1\times2)^7 + (1\times2)^6 + (1\times2)^5 + (1\times2^0)^4 + (1\times2^0)^3 + (1\times2)^2 + (1\times2)^1 + (1\times2)^0$

$0 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$

$(255)_{10}$ Ans.

$(255)_{10}$

Q No 1:

part No 3:

$(4D7F)_{16} = (?)_{10}$

Ans:

$(4D7F)_{16} = (?)_{10}$

$4 \times 16^3 + D \times 16^2 + 7 \times 16^1 + F \times 16^0$

$4 \times 4096 + 13 \times 256 + 16 \times 7 + 15 \times 1$

$16384 + 3328 + 112 + 15$

$\Rightarrow (19839)_{10}$ Ans

Q No: 1 **part No 4:**

$(128)_{10} = (?)_{16}$

Ans $(128)_{10} = (?)_{16}$

16	128
16	8 - 0

$(80)_{16}$ Ans

Q No: 1 part No 5:

$(3A6F)_{16} = (?)_2$

Solution As

3 A 6 F

0011 1010 0110 1111 So

$(0011101001101111)_2 = (3A6F)_{16}$

Ans.

Q No: 1 **part No 6**

$$(11\,0000\,1111\,00101)_2 = (?)_{16}$$

Solution As

1100 0011 1110 0101

12 3 14 5

↓ ↓

C E

So $(C3E5)_{16}$ Ans.

Q No: 1 **part 7:**

$$(-16)_{10} = (?)_2$$

Because the decimal integer is negative

So the MSB is Binary will be 1

Now convert 16 into Binary

2	16
2	8 - 0
2	4 - 0
2	2 - 0
	1 - 0

=> 00010000

Now taking 2's complement

```
  0001 0000
  1110 1111
+          1
-----------
  1111 0000
```

$(-16)_{10} = (11110000)_2$ Ans

Q No: 1 part. No: 8

$(01111111)_2 - (00000111)_2$ 2's complent

Solution

1st complement of 00000111 = 11111000

Now add $\overset{1\ 1\ 1\ 1\ 1\ 1\ 1}{0\ 1\ 1\ 1\ 1\ 1\ 1\ 1}$

$1\ 1\ 1\ 1\ 1\ 0\ 0\ 0$

$\overset{\text{carry}}{\overset{\swarrow}{1}}\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1$

Now add carry to answer as

$\overset{1\ 1\ 1}{0\ 1\ 1\ 1\ 0\ 1\ 1\ 1}$

$+\ 1$

$0\ 1\ 1\ 1\ 0\ 0\ 0$

$(0111000)_2$ Ans

Q No 1: Part No 9

Q No 1: part No: 9

$(6D)_{16} - (3F)_{16}$

$(6D)_{16} = 0110\ 1101$

$(3F)_{16} = 0011\ 1111$

Take 1st complement of second number. As

$0011\ 1111 = 1100\ 0000$

Now add both number

$$\begin{array}{r} 0110\ 1101 \\ +\ 1100\ 0000 \\ \hline 1\ 0010\ 1101 \end{array}$$

carry

add this carry with Result

$$\begin{array}{r} 0010\ 1101 \\ +\ \ \ \ \ \ \ 1 \\ \hline 0010\ 1110 \end{array}$$

$(0010\ 1110)_2$ Ans

QNo: 1 Post No 10

10) $(11111111)_2 = \pm(?)_{10}$

Solution

$(11111111)_2 = \pm(?)_{10}$

Signed integer ai MSB is "1".

which show numbers is negative

=> 11111111

$-1\times2^7+1\times2^6+1\times2^5+1\times2^4+1\times2^3+1\times2^2+1\times2^1+1\times2^0$

$-128+64+32+16+8+4+2+1$

$\boxed{-1}$ Ans

Q NO: 4

part No: 1

Ans: A language whose source program can be compiled and run on a wide variety of computer system is said to be paitable.

Q No: 4 part No: 2

A high leved language may not be provide for direct hardware access. even if it does awkword coding tecnirues possible maintain problem.

Q No: 4: part No: 4

W = 1110110 0

X = 00010011

Y = 00111100

W = 11101100

X = 00010011

Y = 00111100

So dat is:

$Z = W \vee Y \wedge \neg Y$

$Y = 00111100$

$\neg Y = 11000011$

$\Rightarrow Y \wedge \neg Y = 00010011$

11000011

AND 00000011 AND

$\Rightarrow W \wedge X \wedge \neg Y$

11101100

00000011 OR

11101111

QNo4: **Part No: 6**

Ans: Conventional memory is out side the CPU and it responds more slowly to access request. Registers are hard-wired inside the CPU.

QNo: 4 **part No 5**

Ans: $\neg(A \vee B)$

create table

A	B	$\neg A$	$\neg A \vee B$
F	T	T	T
F	T	T	T
T	F	F	F
T	F	F	T

QNo:4 Part No: 7

Ans: XMM Register:

The x86 architecture also contain eight 128-bit register called XMM registers. they are used by Streaming SIMD extension to the instruction.

2) MMX Registers:

MMX tecnology improves the performance of Intel processor when implementing advanced multimedia and communication application. The 64 bit MMX register support special instruction called SIMD (Single-instruction) Multiple-DATA.

3) Segment Registers:

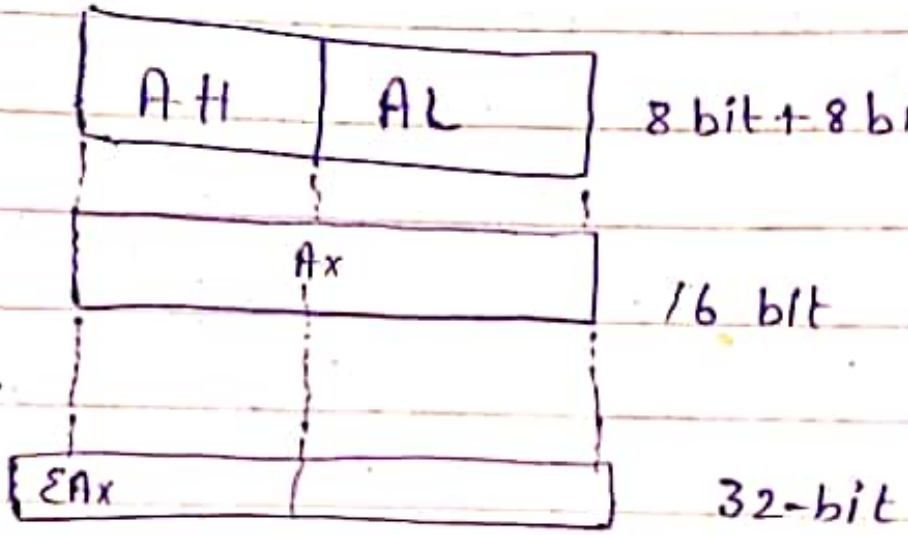
In real address mode 16-bit segment regsiter indicate base addresses of preassigned memory areas named segments

Reg No

QNO: 4 Page-19

4)

* General-purpose register:



* Basic programming Execution Register:

Ans: Registers are high-speed storage location directly inside the CPU designed to be accessed at much higher speed than conventional memory. when a processing loop is optimized for speed.

Q No: 4 | Part No: 3

Answere:

Unicode is a character encoding standard that has widespread acceptance. They store letter and other characters by assiniy a number for each one. Before unicode was inrented. there was hundered of different encoding system for assining these numbers. No single encoding could contain enough characters.

Pag-13 Page-13

QNo.5

Discuss the following MASM directive in detail:

INCLUDE

.386 .MODEL .STACK

.DATA .CODE PROC ENDP

Solution:

.386:

```
model flat
.stack 4096
ExitProcess PROTO
dwExitCode:DWORD
```

The .386 directive identifies it as a 32-bit programm. Line2 use the flat memory model and window requires the model convention to be used.

Line3 set aside 4096 bytes of storage Line4 declare a phototype for the exit process function.

Q No: 5

* . MODEL:

This tells the assembler which memory model to use In 32-bit program, we use the flat memory model which is associated with the processor's protected mode.

Q No: 5

* . ~~STACK~~ . STACK

The Stack directive tell how many bytes of memory to reserve for the runtime Stack 4096 happen to correspond to the size of a memory page in this processor system for managing memory.

Q No: 5

* . DADA

The DATA Directive creates a near data segment.

. This DATA Segment contain the frequently used data for your programmer.

. DATA segment can occupy

- up to 64-K in MS-DOS
- or up to 512 megabyte under

filed moded in window NT

QNo:5

* . CODE

. eODE

main pROC:

It is beginning of the eode area of the program (meanings what is after wood is usually the main procedure)

QNo:5

* pROC

MASK Start end end of a procedure block called label.

The statement in the block can be called with the ecall instruction or INVOKE directives

Syntax

label pROC [distance] [language_type]

[[public] | pRIVATE| EXPORT]]

[[< prototypeArg >]]

* [[uses reglist]] [[parameter [[:tag]] ---]]

* [[FRAME [[: chandles_address]]]]

Statement

labed End.

ENDP :

Masles the end of procedure name previously begin with PROC

syntax :

~~See PROC~~

name ENDP

Remarks:

See PROC

Q No: 2 Part No 1

Write NOTE on Each Following:

part No: 1

Embeded system.

A embeded system is a computer system a combination of a computer processor, computer memory and input/output peripheral devices that has a dedicated function within a larger machanical or eclectrical system

Q No: 2 part No: 2

Device Drivers:

Device Drives are program that translate general operating system command into specific reference to hardware detail that only the manufacture knows. Drivers are hardware depended and operating system-specific. A device driver is a file that lets the computer know

the configuration and specification of a contain hardware device

Q No 2: part No:3

virtual machine concept:

A virtual machine (VM) is a software program or operating system that not only exibits the behavios of a separate computer but is also capable of performing task such as running application and program like a separate computer

Q No 2: part No: 4

instruction execution cycle:

Ans: There three execution cycle

1) fetch
2) decode
3) execute

Page-19

Q No 2: part No 5:

Motherboard chipset:

Ans: The chipset is the "glue" that connect the microprocessor to the rest of the motherboard and therefore to the rest of the computer. on a pc it consist of two basic parts

the motherbridge and the Southbridge

All of the computer communicate with the cpu through the chipset.

Q No 2: part No 6

Access Level for input-output operation:

Ans: In computer input/output or I/O (or, informally, io or IO) is the communication b/w an information processing system such as a computer and the outside world possible human or another information processing system input are the signals

or data received by the system and output are the signals or data sent from it. The term can also be used as part of an action; to perform I/O" is to perform an input or output.

QNo 2: Part No: 7

Basic part of assembly language instruction.

Ans:

There are four basic part of instruction.

Ans: [label:-] mnemonic [operands] [; comment]

QNO: 3

Differentiate the Following:

(a) Assembly language and high level language:

Ans. An Assembly language directly control the physical hardware. A high level language is much more abstract, which must be compiled/translated is a set up from this. High level language act the middle-man. between human thinking and machine language.

NO: 3

Part: B

Difference between protected mode and real mode.

Real mode: Real mode is an operational mode that allows newer intel 286 processor to acquire the characteristics of the reduced 8086 or 8088 processors Real mode provide a higher

clock speed but only restricts that processor to a 16-bit and a minimum of 1mb RAM instruction protected mode: The protected mode is a 32-bit operating mode that is identified on inted 80286 or later on. This mode allow the termination of a failed program without restarting the computer or runnings programm it offer access to addressing by virtual memory, expanded memory and multiple task while it protecting programm against memory over writing

No: 3

post: C

Assembler And Linker:

Ans: The main ouput produce by assembler an input assembly language of the file into an object file in (ELF): ELF files produced by the assembler are relocatable files that hold code and/or data they are input file for linker

Q No 3:

Part: D

Difference b/w instruction and directive.

Ans: Instruction: An instruction is a task to be carried out by the processors at run time instruction are assembled into machine code and eventually linked into the final executable

Directive

A Directive is an instruction to the assembler telling it how to treat the data it is asked to assemble. Directive only used at assembly time and although they may effect ~~they may~~ the way the code is generated they don't result in any code generation themselves.

QNo:3 part: E

Difference between Code label and DATA label:

CODE LABEL:

CODE Label is the labeld that we have an code as we see in case of condition jump (Label-1) and it normally used for loop control statement

QNo:3 part: F

Difference between Line comment and block comment:

Ans:

Line comment:

A single Line comment and as implied only applied to a single line in the source code" (the program)

Block comment:

A block comment and refers usually refes to a paragraph of text. A block comment has a start symbol and an end symbol and everything between is ignored by the computer

Q No: 3

(Part No: G)

page-25

Q No: 3 Part: G

Difference between Equal-sign directive and EQU directive

Ans:

. EQual-Sign Directive:

The Equal-sign directive associates a Symbol name with an integer expression

This Syntax is

name = Expression

Expression is a 32-bit integer

- May be redefined
- Name is called symbolic constant

EQU-directive:

Define a Symbol as either an integer or text expression

- cannot be redefined
- The EQU directive give a symbolic name to a numeric constant a register-relative value or a PC-relative value

QUESTION No: 6:

Q No: 6 Part: 1: ~~Page-26~~

Part 1: A:

write a program the calculate the following expression A=(A+B)-(C+D)

Ans:

A=(A+B)-(C+D)

.data; data segment, read

and write

Var A BYTE 10

Var B BYTE 20

Var C BYTE 30

Var D BYTE 40

Final val BYTE ?

. code; code. segment, read-only

main PROC

mov al, VAR A

Mov bl, var B

mov Cl, Var C

mov dl, var D

ADD al, bl; compute (A+B)

ADD Cl, dl; compute (A+B)

SUB al, Cl; compute (A+B)-(C+D)

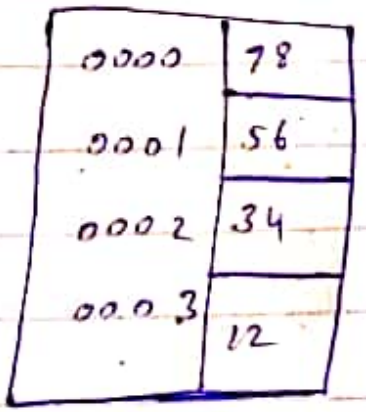
Mov Final val, al

```
Call Dump Regs
exit
main ENDP
End main
```

QNo: 6 Part: B

Ans: When placed in memory at offset 0000, 78h would be stored in the first byte 56h would be stored in the second byte and the remains bytes would be at offset 0002 and 0003 as shown in figure 3.14

Little-endian representation of 12345678h.



QNo. 6 Part: C

Ans:

```
.data
String byte "Assembly language is easy",0
Stringsize byte ?
```

• code

mov eax, size of string1
mov string size, eax

Q No: 6 Part: D

Let the Arithmetic operation is $x = -a + (b - c)$

INCLUDE Irvine32.inc

• data

x DWORD ? : uninitialized variable.
a DWORD 10 ; initialize variable 'a'
b DWORD 26 ; initialize variable 'b'
c DWORD 15 ; initialize variable 'c'

• code

main PROC ;
mov eax, a ; mov value of 'a' into 'eax'
neg eax ; EAX = -10
mov ebx, b ; move value of 'b' into 'ebx'
sub ebx, c ; EBX = 11
add eax, ebx ; perform -10+11
mov x, eax ; 1
call DumpRegs
exit
main ENDP
END main