**NUST SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Semester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Section: \_\_\_\_\_\_\_\_\_\_\_\_ |

Department of Electrical Engineering

EE- 222: Microprocessor Systems

**LAB 08: Floating Point Numbers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student’s Name | Reg. # | Lab Conduct and Report | Viva | Total |
| 10 | 5 | 15 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**LAB No.08**

**Objective:** The aim of this lab is to familiarize students about initiation and manipulation of floating point numbers

**Software Used:** MASM615

**Commands:**

Data Types:

* Real4
* Real8
* Real10

1. ***FINIT***
2. ***FLD as Mov***

FLD m32fp

FLD m64fp

FLD m80fp

FLD ST(i)

* The FLD1 instruction pushes 1.0 onto the register stack.
  + The FLDL2T instruction pushes log2 10 onto the register stack.
  + The FLDL2E instruction pushes log2 *e* onto the register stack.
  + The FLDPI instruction pushes pi onto the register stack.
  + The FLDLG2 instruction pushes log10 2 onto the register stack.
  + The FLDLN2 instruction pushes log*e* 2 onto the register stack.
  + The FLDZ (load zero) instruction pushes 0.0 on the FPU stack.

1. ***FST***

* The FST (store floating-point value) instruction copies a floating-point operand from the top of the FPU stack into memory.
* The FSTP (store floating-point value and pop) instruction copies the value in ST(0) to memory and pops ST(0) off the stack.

1. ***FADD***

FADD

FADD *m32fp*

FADD *m64fp*

FADD ST(0), ST(*i*)

FADD ST(*i*), ST(0)

1. ***FMUL***

Same as FADD

1. ***FCOM***

FCOM ;Compare ST(0) to ST(1)

FCOM *m32fp ;*Compare ST(0) to *m32fp*

FCOM *m64fp ;*Compare ST(0) to *m64fp*

FCOM ST(*i*)

* 1. Use the FNSTSW instruction to move the FPU status word into AX.
  2. Use the SAHF instruction to copy AH into the EFLAGS register.

1. ***READFLOAT***
2. ***WRITEFLOAT***
3. ***call ShowFPUStack***

**Task 1:** Write an assembly language program that asks user to enter 10 real numbers one by one. Add those numbers and display the result to the user.

**Task 2:** Write a program that takes radii of circle and gives area, circumference and diameter as output

**Task 3:** Write an assembly language program that asks user to enter 2 numbers and displays the greater one as an output

**Note: - Viva Marks = 05 Marks**

**The lab marks will be condensed to 10 Marks.**