**Microprocessor and Assembly Programming Laboratory**

**B.Tech. IV Semester**



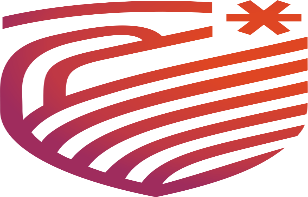
**Name : SATYAJIT GHANA**

**Roll Number : 17ETCS002159**

**Department : Computer Science and Engineering**

**Faculty of Engineering & Technology**

**Ramaiah University of Applied Sciences**



**Ramaiah University of Applied Sciences**

Private University Established in Karnataka State by Act No. 15 of 2013

|  |  |
| --- | --- |
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Year/Semester | 2017/4th Semester |
| Name of the Laboratory | Microprocessor and Assembly Programming Laboratory |
| Laboratory Code | CSC214A |

**List of Experiments**

|  |  |
| --- | --- |
| 1. Data transfer operations |  |
| 1. Arithmetic operations |  |
| 1. Logical operations |  |
| 1. Array manipulation |  |
| 1. Controlling execution flow using conditional instructions |  |
| 1. String manipulation |  |
| 1. Searching an element in an array |  |
| 1. Sorting an array |  |
| 1. Inline assembly program for code optimization |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No.** |  |  |  |  | | **Lab Experiment** | **Viva**  **(6)** | **Results**  **(7)** | **Documentation**  **(7)** | **Total Marks**  **(20)** |
| 1 | Data transfer operations |  |  |  |  |
| 2 | Arithmetic operations |  |  |  |  |
| 3 | Logical operations |  |  |  |  |
| 4 | Array manipulation |  |  |  |  |
| 5 | Controlling execution flow using conditional instructions |  |  |  |  |
| 6 | String manipulation |  |  |  |  |
| 7 | Searching an element in an array |  |  |  |  |
| 8 | Sorting an array |  |  |  |  |
| 9 | Inline assembly program for code optimization |  |  |  |  |
| 10 | Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks | | | |  |
|  | **Total Marks** | | | |  |

# Laboratory 1

Title of the Laboratory Exercise: Data transfer operations

1. Introduction and Purpose of Experiment

Students will be able to define data of different data types and perform data transfer operations on the data

1. Aim and Objectives

Aim

To develop assembly language program to perform data transfer operations on different data.

Objectives

At the end of this lab, the student will be able to

* + Define data of different data types
  + Perform data transfer operations
  + Create a simple assembly language program
  + Use GAS assembler
  + Understand GNU debugger

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions

1. Perform the following data transfer operations

|  |  |
| --- | --- |
| 1. 32 bit integer data to a | General Purpose register  Segment Register  Memory |
| 2. 16 bit integer data to a | General Purpose register  Segment Register  Memory |
| 3. 8 bit integer data to a | General Purpose register  Segment Register  Memory |
| 4. 32 bit integer data from a General purpose register to a  *(Repeat the same for 16 bit integer data and 8 bit integer data)* | General Purpose register  Segment Register  Memory |
| 5. 32 bit integer data from memory to a  *(Repeat the same for 16 bit integer data and 8 bit integer data)* | General Purpose register  Segment Register  Memory |
| 6. 32 bit integer data from memory to | Memory region |

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 2

Title of the Laboratory Exercise: Arithmetic Operations

1. Introduction and Purpose of Experiment

Students will be able to perform all arithmetic operations using assembly instructions

1. Aim and Objectives

Aim

To develop assembly language program to perform all arithmetic operations.

Objectives

At the end of this lab, the student will be able to

* + Identify the appropriate assembly language instruction for the given arithmetic operations
  + Perform all arithmetic operations using assembly language instructions
  + Understand different data types and memory used
  + Get familiar with assembly language program by developing simple programs

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions
2. Consider the following source code fragment

*Int a,b,c,d;*

*a= (b + c)-d + (b\*c) / d;*

Assume that *b, c, d* are in registers. Develop an assembly language program to perform this assignment statements. Assume that *b, c* are in registers and *d* in memory. Develop an assembly language program to perform this assignment statements.

1. For the following values of A and B, predict the values of the N, Z, V and C flags produced by performing the operation A + B. Perform the following operation. Load the values of A and B into two registers. Perform an addition of the two registers. Read the flags after each addition and compare those flag values with your predictions. Comment on the results. When the data values are signed numbers, what do the flags mean? Does their meaning change when the data values are unsigned numbers?

0xFFFF0000 0xFFFFFFFF 0x67654321 (A)

+ 0x87654321 +0x12345678 + 0x23110000 (B)

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 3

Title of the Laboratory Exercise: Logical operations

1. Introduction and Purpose of Experiment

Students will be able to perform all logical operations using assembly instructions.

1. Aim and Objectives

Aim

To develop assembly language program to perform all logical operations

Objectives

At the end of this lab, the student will be able to

* + Identify the appropriate assembly language instruction for the given logical operations
  + Perform all logical operations using assembly language instructions
  + Get familiar with assembly language program by developing simple programs

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions:
   1. Consider the following source code fragment

*Int a,b,c,d;*

*a= (b AND c) XOR d;*

*a=(b XOR c) OR d;*

Assume that *b, c, d* are in registers. Develop an assembly language program to perform this assignment statements. Assume that *b, c* are in registers and *d* in memory. Develop an assembly language program to perform this assignment statements.

* 1. Consider the following source code fragment

*Int a,b,c,d;*

*A = (b\*c) / d;*

Perform multiplication and division by shift operations

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 4

Title of the Laboratory Exercise: Array manipulation

1. Introduction and Purpose of Experiment

Students will be able to access elements in an array using indexed addressing mode.

1. Aim and Objectives

Aim

To develop assembly language program to access the elements in an array or to access a particular element in an array using indexed addressing mode.

Objectives

At the end of this lab, the student will be able to

* + Perform array manipulation using indexed addressing mode
  + Discuss indexed addressing modes
  + Access the appropriate element in an array

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions:
   1. Assume an array of 25 integers. Translate this C statement/assignment using the indexed form:

*x = array [15] + y*

* 1. Create an array with 10 integers. Develop an assembly language program to print the elements of the array.
  2. Develop assembly language program to perform the following array assignment in C:

*for ( i = 0; i <= 10; i++)*

*{*

*a[i] = 10;*

*}*

* 1. Develop an assembly language program to generate the first n numbers in Fibonacci series.

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 5

Title of the Laboratory Exercise: Controlling execution flow using conditional instructions

1. Introduction and Purpose of Experiment

Students will be able to perform control flow operations using conditional instructions

1. Aim and Objectives

Aim

To develop assembly language program to perform control flow operations using conditional instructions.

Objectives

At the end of this lab, the student will be able to

* + Identify the appropriate assembly language instruction for the given conditional operations
  + Perform all conditional operations using assembly language instructions
  + Get familiar with assembly language program by developing simple programs

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions

Develop an assembly language program to perform the following

1. Print all ‘n’ natural numbers in reverse order
2. Sum of all ‘n’ natural numbers
3. Print all even numbers in ‘n’ natural numbers
4. Print all odd numbers in ‘n’ natural numbers
5. Compute GCD for the given two natural numbers
6. Compute LCM for the given two natural numbers
7. Develop an assembly language program to compute the parity of a hexadecimal number stored in the Register1. If Register1 has odd number of ones, update Register2 with 0x01. If Register1 has even number of ones, update Register2 with 0x00. Note: Register1 and Register2 can be any General Purpose Registers.

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 6

Title of the Laboratory Exercise: String manipulation

1. Introduction and Purpose of Experiment

Students will be able to perform all string manipulations in assembly language

1. Aim and Objectives

Aim

To develop assembly language program to perform all string operations like inserting a byte, deleting a byte and copying a string as a sub-string

Objectives

At the end of this lab, the student will be able to

* + Identify instructions for performing string manipulation
  + Use indexed addressing mode
  + Apply looping instructions in assembly language
  + Use data segment to represent arrays

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions

Develop an assembly language program to perform the following

1. Copy the contents of MSG1 to MSG2
2. Copy the contents of MSG1 to MSG3 in reverse order
3. Copy the contents of MSG3 to MSG4 after ‘n’ th character in MSG4
4. Insert a byte in MSG1
5. Delete a byte in MSG1
6. Develop an assembly language program to compare two strings and print a message “Equal” if they are equal, “Not Equal” if they are not equal.
7. Calculations/Computations/Algorithms
8. Presentation of Results
9. Analysis and Discussions
10. Conclusions
11. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 7

Title of the Laboratory Exercise: Searching an element in an array

1. Introduction and Purpose of Experiment

Students will be able to perform search operations in an array of integers or characters

1. Aim and Objectives

Aim

To develop assembly language program to perform search operations in an array

Objectives

At the end of this lab, the student will be able to

* + Identify instructions to be used in assembly language
  + Perform search operations in assembly language

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions

Develop an assembly language program to perform the following:

1. Searching an element in an array of ‘n’ numbers
2. Searching a character from a string
3. Read a sentence with at least one special character and search for the special character and print it. E.g., consider the input {youremailid@msruas.ac.in }

Output: @, .

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 8

Title of the Laboratory Exercise: Sorting

1. Introduction and Purpose of Experiment

Students will create assembly code with sorting techniques and nested loops

1. Aim and Objectives

Aim

To develop assembly language program to perform sorting using nested loop structures

Objectives

At the end of this lab, the student will be able to

* + use nested loops in assembly
  + perform sorting in ascending/ descending order
  + Build complex looping logic in assembly language

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create laboratory report documenting the work

1. Questions

Develop an assembly language program to perform the following

* + 1. Arrange an array of ’n’ numbers in ascending order
    2. Arrange an array of ’n’ numbers in descending order
    3. Determine the second smallest number in an array.

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks

# Laboratory 9

Title of the Laboratory Exercise: Inline assembly language programs for code optimisation

1. Introduction and Purpose of Experiment

Students will create C programs with inline assembly code for code optimisation

1. Aim and Objectives

Aim

To develop inline assembly language program for code optimisation

Objectives

At the end of this lab, the student will be able to

* + Identify inline assembly language calls
  + Explain optimization of program by exploiting architectural features in target computer
  + Create C programs with inline assembly code

1. Experimental Procedure

1. Write algorithm to solve the given problem

2. Translate the algorithm to assembly language code

3. Run the assembly code in GNU assembler

4. Create a laboratory report documenting the work

1. Questions:

Develop a C program without inline assembly instructions and find out the code size and memory used for the program. Develop the C program with inline assembly instructions and find out the code size and memory used for the program. Compare the results.

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Signature and date Marks