

LAB 4: Types of Operations

Objectives:

- a) Understand the major categories of common operations used in assembly language.
- b) Understand why there are various operations for each category.

1) Background: In this lab you will investigate operation types. The computer has 4 major operations. They are input, output, processing and storage. To accomplish these tasks an assembly language must have facility to transfer data, control the flow of logic, and to perform arithmetic and logical operations. This gives four basic types of instructions we will investigate in this lab.

- (1) Data transfer operators move data from one location to another. These instructions usually require operands for source of the data and a destination where the data will be moved. The push and pop instructions are special cases. With push instructions the destination is implicit as the top of the stack and with the pop instructions the source is implicit as the top of the stack.
- (2) Transfer of Control operators changes the sequence in which instructions are performed. This is how loops, decision statements and function calls are performed. In assembly these are accomplished with jump and branch instructions.
- (3) Arithmetic operators perform operations such as +, -, *, / and other mathematical operations.
- (4) Logical operators perform logical operations such as AND, OR, NOT, and XOR. They also include comparison, bit shift and bit rotation operators.

2) Exercises:

- a) Go to the Frances-A website and compile the default program.
 - i) Put each assembly instruction into the correct category of either: Data Transfer, Transfer of control, Arithmetic, Logical, or Other.

b) Type in the following code :

```
int main(){
int x = 1, y =0, z =5;
while (x < z)
{ y = y + x;
x = x+1;
}
}
```

- i) Put each instruction into the correct category of either: Data Transfer, Transfer of control, Arithmetic, Logical, or Other.
- ii) For each of the Arithmetic operators list the value of the operand(s).
- iii) For each of the Data Transfer operators, state the source and destination of the data.
- iv) For each of the Transfer of Control operators explain where control is being transferred in reference to the source code.

v) Step through the assembly code. At which step(s) in the assembly code is x being incremented.

c) Type in the following code:

```
int main(){  
    int n = 1;  
    float y =5;  
    char c ='A';  
}
```

- i) List each instruction that assigns the value to each of the three variables.
- ii) Why are these Data Transfer operators different? What does each mean?