CSCI 320 – Assignment 1

Created by: Thomas Hoerger

Objectives

This program is used to assign strings to memory address locations.

Address register direct	An
Address register indirect	(An)
Post-increment	(An)+
Pre-decrement	-(An)
Displacement	d(An)

Equipment Used

EASy 68K simulator

Procedure

The first assignment is a simple copy program. That is, you are to duplicate the code that is supplied on the last page of this assignment. The goal here is to have you interact with the EASy68K Assembler to create data in memory and execute a small program.

1. Assign the string 'ABCDEFGHIJKLMNOPWRSTUVWXYZ' to memory location 2000
2. Assign the string 'abcdefghijklmnopwrstuvwxyz' to memory location 2000
3. Assign the string '0123456789' to memory location 2000
4. Enter the code provided in the START section of the editor
5. Assemble the source code using the play button or F9
6. When the simulator opens, click the run button or F9
7. Bring up the memory window and scroll to address location 2200 to see the string printed

New Operations Learned

ORG
DC.L
DC.W
LEA.L
MOVE.B
Address register direct An

Address register indirect (An)
Post-increment (An)+
Pre-decrement -(An)
Displacement d(An)

Program Description

This program stores English Upper case letters from $00002000 , Lower case letters from $00002020 and numbers(0 to 9) from $00002040.

SOURCE CODE

```
    ORG    $2000

    DC.L 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

    ORG    $2020

    DC.L 'abcdefghijklmnopqrstuvwxyz'

    ORG    $2040

    DC.W '0123456789'


START:

    LEA.L  $002000,A2

    LEA.L  $002020,A3

    LEA.L  $002040,A4

    LEA.L  $002200,A1


    MOVE.B $002016,(A1)+

    MOVE.B $002024,(A1)+

    MOVE.B $00202B,(A1)+

    MOVE.B $002022,(A1)+

    MOVE.B $00202E,(A1)+

    MOVE.B $00202C,(A1)+

    MOVE.B $002024,(A1)+

    MOVE.B #32,(A1)+

    MOVE.B 19(A3) ,(A1)+

    MOVE.B 14(A3) ,(A1)+
```

```
MOVE.B #32,(A1)+
MOVE.B 2(A2) ,(A1)+
MOVE.B 18(A2) ,(A1)+
MOVE.B 2(A2) ,(A1)+
MOVE.B 8(A2) ,(A1)+
MOVE.B #32,(A1)+
MOVE.B 3(A4) ,(A1)+
MOVE.B 2(A4) ,(A1)+
MOVE.B (A4) ,(A1)+
MOVE.B #33,(A1)+
MOVE.B #33,(A1)+
MOVE.B #33,(A1)+
MOVE.B #32,(A1)+

SIMHALT             ; halt simulator

END    START        ; last line of source
```

Results

Before Execution

```
*-----------------------------------------------------------
* Title      : Easy 68K Lab 1.C
* Written by : Thomas Hoerger
* Date       : 1/27/24
* Description:
*-----------------------------------------------------------
    ORG    $2000
    DC.L 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    ORG    $2020
    DC.L 'abcdefghijklmnopqrstuvwxyz'
    ORG    $2040
    DC.W '0123456789'

START:
    LEA.L   $002000,A2
    LEA.L   $002020,A3
    LEA.L   $002040,A4
    LEA.L   $002200,A1

    MOVE.B  $002016,(A1)+
    MOVE.B  $002024,(A1)+
    MOVE.B  $00202B,(A1)+
    MOVE.B  $002022,(A1)+
    MOVE.B  $00202E,(A1)+
    MOVE.B  $00202C,(A1)+
    MOVE.B  $002024,(A1)+
    MOVE.B  #32,(A1)+
    MOVE.B  19(A3) ,(A1)+
    MOVE.B  14(A3) ,(A1)+
    MOVE.B  #32,(A1)+
    MOVE.B  2(A2) ,(A1)+
    MOVE.B  18(A2) ,(A1)+
    MOVE.B  2(A2) ,(A1)+
    MOVE.B  8(A2) ,(A1)+
    MOVE.B  #32,(A1)+
    MOVE.B  3(A4) ,(A1)+
    MOVE.B  2(A4) ,(A1)+
    MOVE.B  (A4) ,(A1)+
    MOVE.B  #33,(A1)+
    MOVE.B  #33,(A1)+
    MOVE.B  #33,(A1)+
    MOVE.B  #32,(A1)+


    SIMHALT             ; halt simulator

* Put variables and constants here

    END    START        ; last line of source
```

Figure 1 shows the code properly entered in the simulator.

Replace ORG $2000 DC.L (what you want to set to 2000) to set a string to memory location 2000.

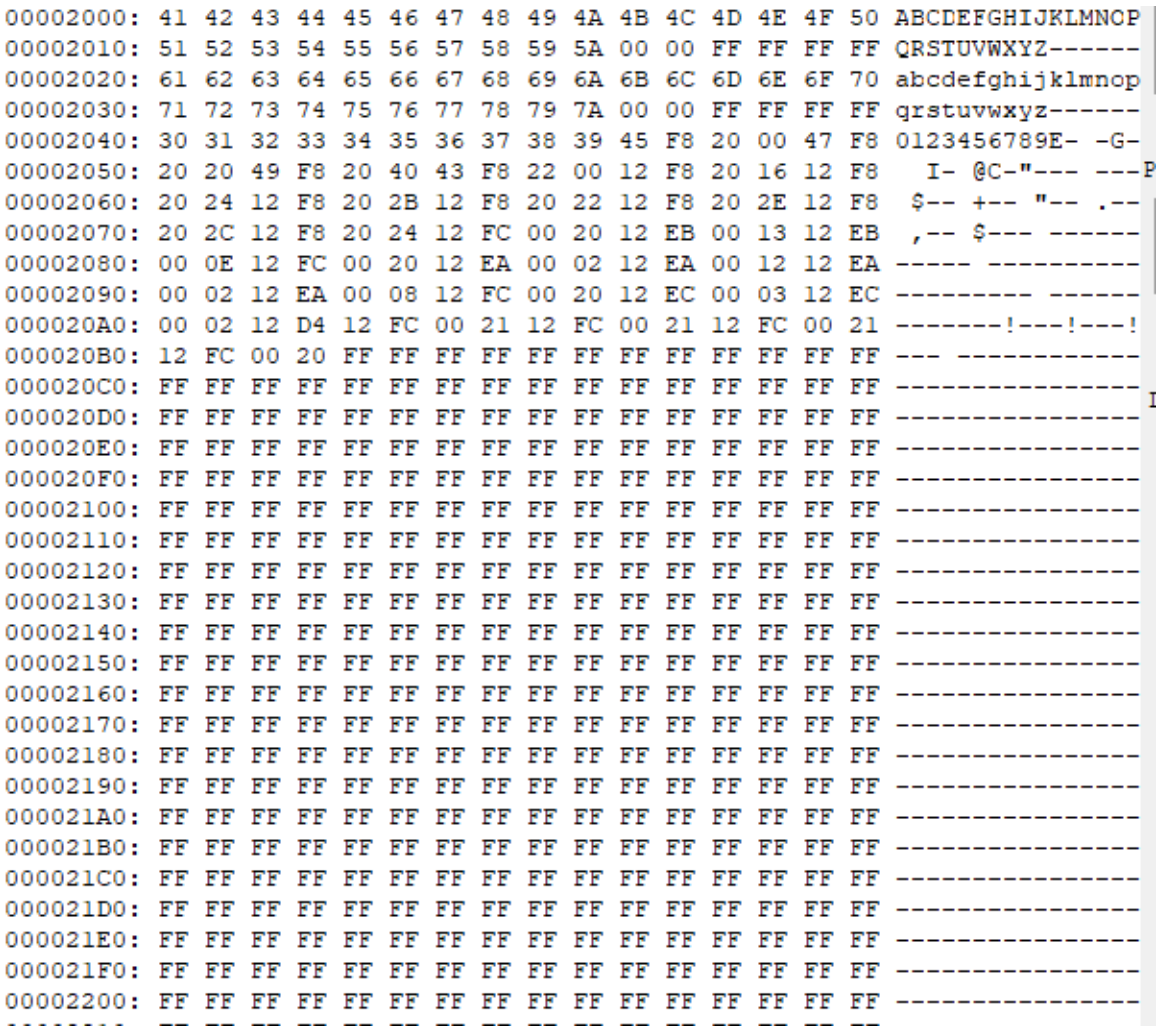

Figure 4 shows the memory window before the program is executed.

After Execution of the program

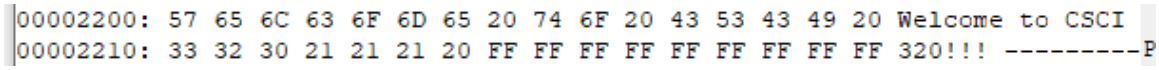

Figure 3 shows the memory window at address location 2200 and the string assigned to it.

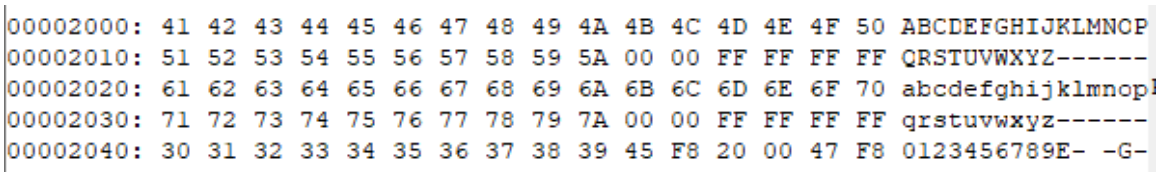

Figure 4 shows the memory window at address location 2000, 2010, 2020, 2030, and 2040 and the strings assigned to them.