Lab Report 3.C – More Addressing Modes

Description:

In this lab, we were asked to load a linked list into memory and insert an element between elements 2 and 3.

Equipment Used:

- EASy 68K simulator

Procedure:

I started by initializing parts of memory using DC.L to copy the initial linked list into memory (table 1).

Table 1 A list of values to be placed at $2100 in memory.

Address	Data
$74A8	41414141
$74AC	000074D0
$74B0	00000000
$74B4	00000000
$74B8	43434343
$74BC	000074C0
$74C0	44444444
$74C4	00000000
$74C8	00000000
$74CC	00000000
$74D0	42424242
$74D4	000074B8

Next, I initialized the registers A6, A1, and A2 with the addresses of elements 1, 2, and 3, respectively.

Then I wrote a few lines of code to insert the new element at $74B0 into the linked list. Finally, I assembled the source code, ran the program, and took screenshots of my memory display.

Program Description:

This program starts by storing a list of data around $74A8 (seen in table 1). Then it stores values in registers A6, A1, and A2. Finally, the program uses the address stored in A5 to modify the linked list and insert an existing element.

Results:

Before and after photos of memory are also provided below. As you can see, the address of the second element (located at $74D0) now points to $74B0 and the inserted element points towards $74B8.

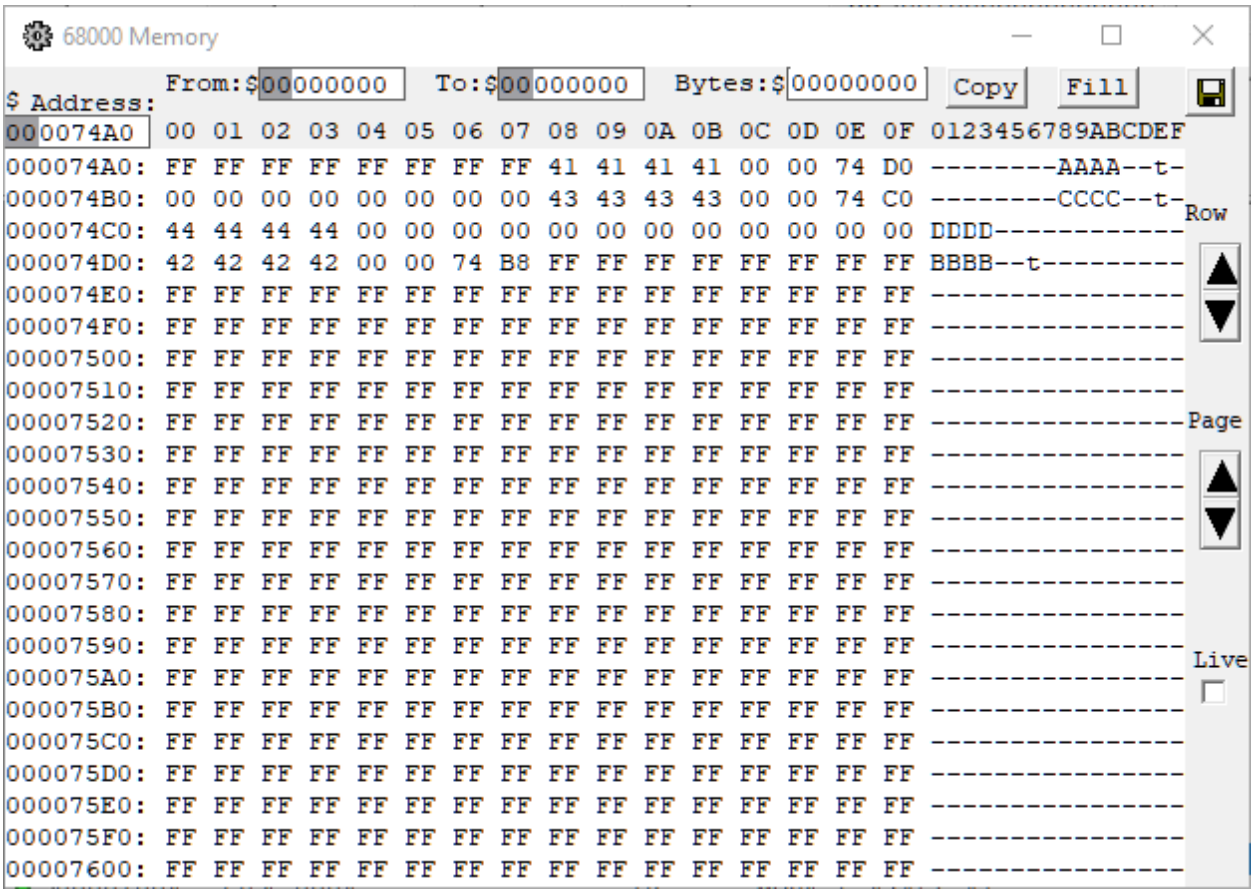

Figure 1 The memory window before running the program.

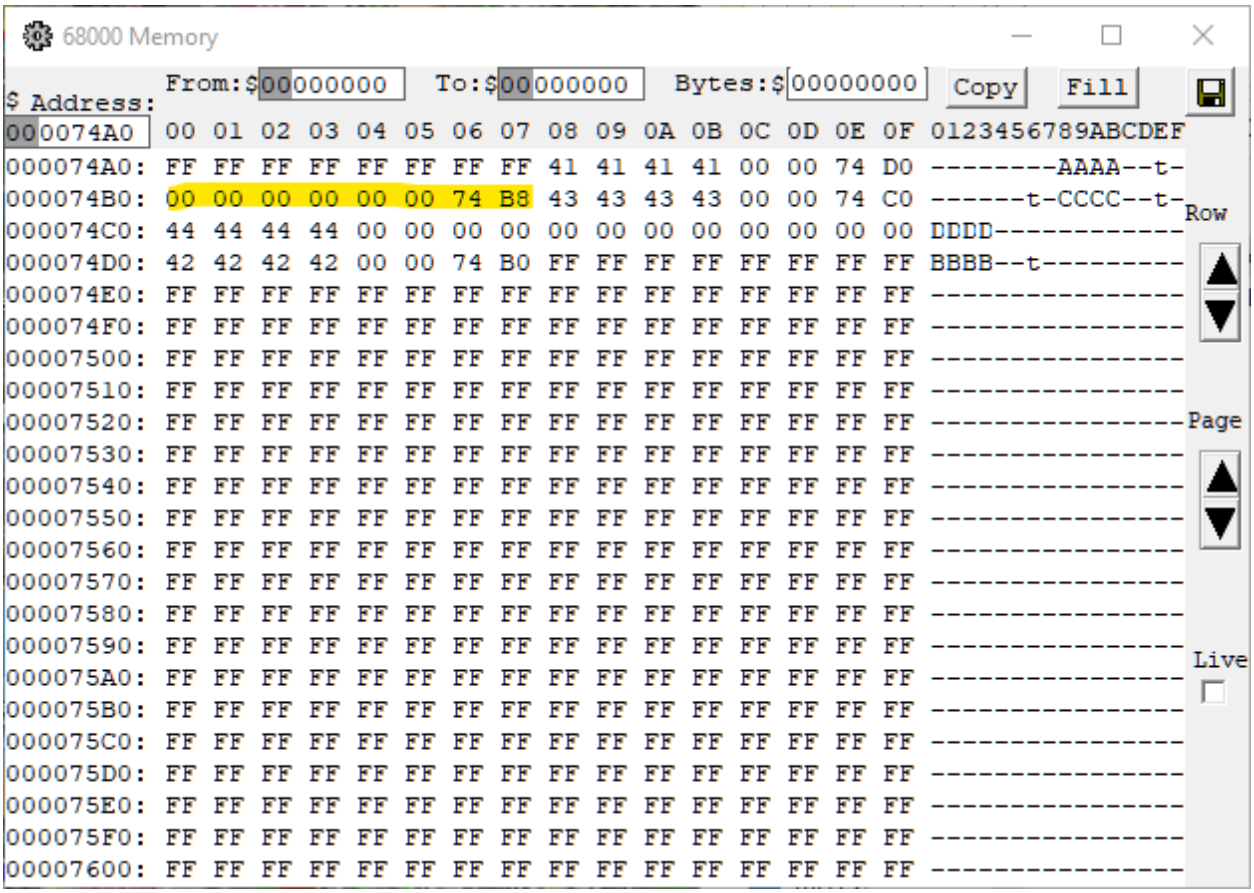

Figure 2 The memory window after executing the full program. The inserted element is highlighted in yellow.

Conclusions:

This lab helped me better understand how linked lists are be implemented at a basic level. After using linked lists extensively in CSCI 301, it was interesting to see how they can be actually represented in memory. The concept of storing addresses in memory is also a useful concept that can be applied to many, many data structures.

Code Listing:

```
ORG $74A8
DC.L $41414141
DC.L $000074D0
DC.L $00000000
DC.L $00000000
DC.L $43434343
DC.L $000074C0
```

```
    DC.L $44444444
    DC.L $00000000
    DC.L $00000000
    DC.L $00000000
    DC.L $42424242
    DC.L $000074B8

    ORG    $1000
START:                  ; first instruction of program
    LEA $0074A8,A6
    MOVE.L 4(A6),A1
    MOVE.L 4(A1),A2

    LEA.L $000074B0,A5
    MOVE.L A2,4(A5) ; Updates which element A5 points to
    MOVE.L A5,4(A1) ; Updates which element the previous element points to

    SIMHALT             ; halt simulator
    END    START        ; last line of source
```