CSCI 592

LAB ASSIGNMENT – 1

Written by

DINESH SEVETI

Date: 01-25-2025

OBJECTIVE

To create and execute a program in assembly language to manipulate data in memory and observe the outcomes.

TECHNOLOGY USED

- Easy68K Assembler software to run the code.
- Code given on Assignment to duplicate the code.

PROCEDURE

- The first step is to copy and paste the code from the given assignment page into the Easy68K Assembler software.
- **ORG** directives to define data in memory.
- Using **LEA** instructions to initialize addresses for each data segment.
- Then **MOVE.B** to manipulate data to specific memory locations.
- Halt the simulator using **SIMHALT** after processing.

OPERATIONS

- Defined uppercase alphabet characters at $2000, lowercase alphabet characters at $2020, and digits at $2040.
- Loaded memory addresses into address registers A2, A3, A4, and A1.
- Sequentially moved specific bytes from predefined memory segments into a new memory region starting at $2200.
- Inserted spaces (#32) and custom values (#33) in the output sequence.

ALGORITHM

- Define memory segments using **DC.L** and **DC.W** for storing characters and numbers.
- Use the **LEA.L** instruction to load the base addresses of these segments into address registers.
- Use **MOVE.B** to transfer specific bytes to a target memory location.
- Add spaces and custom data as needed in the sequence.
- Halt execution using **SIMHALT.**

DESCRIPTION

The program initializes memory with three different data types (uppercase letters, lowercase letters, and digits). It then selectively moves characters from these memory locations to a target region. The purpose of the program is to demonstrate data manipulation using assembly instructions in a simulator environment.

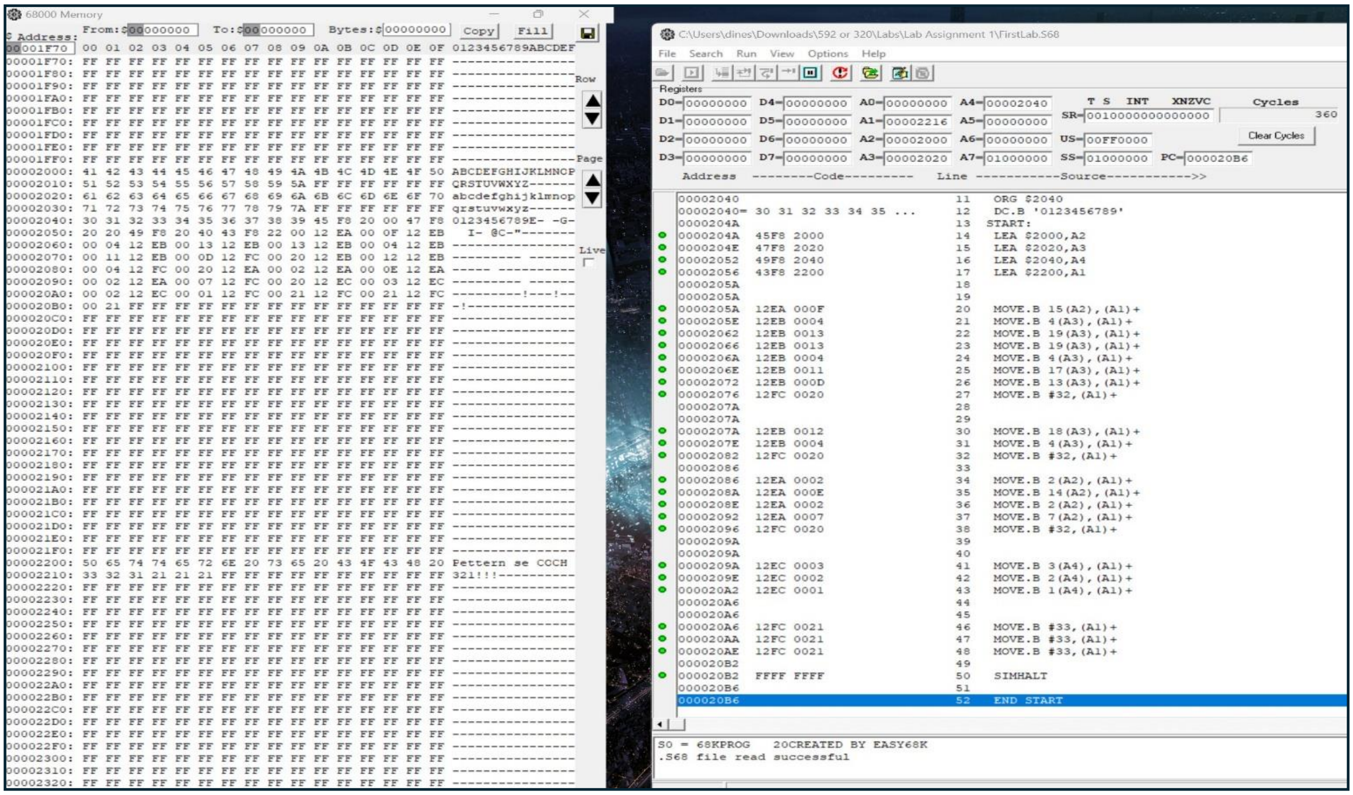
CODE LISTING

```
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
```

OBSERVATIONS

The program correctly moves the desired bytes to the target memory region. Spaces and symbols are inserted as specified in the instructions. Verified the memory layout and data using a simulator. Ensured all addresses and offsets were correctly calculated. Adjusted offsets and verified instructions to prevent overwriting unintended memory locations.

RESULTS



```
$ Address:  From:$00000000   To:$00000000   Bytes:$00000000
00001F70    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
00001F70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001F80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001F90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00001FF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002000: 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 ABCDEFGHIJKLMNOP
00002010: 51 52 53 54 55 56 57 58 59 5A FF FF FF FF FF FF QRSTUVWXYZ------
00002020: 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
00002030: 71 72 73 74 75 76 77 78 79 7A FF FF FF FF FF FF qrstuvwxyz------
00002040: 30 31 32 33 34 35 36 37 38 39 45 F8 20 00 47 F8 0123456789E- -G-
00002050: 20 20 49 F8 20 40 43 F8 22 00 12 EA 00 0F 12 EB   I- @C-"-------
00002060: 00 04 12 EB 00 13 12 EB 00 13 12 EB 00 04 12 EB ----------------
00002070: 00 11 12 EB 00 0D 12 FC 00 20 12 EB 00 12 12 EB --------- ------
00002080: 00 04 12 FC 00 20 12 EA 00 02 12 EA 00 0E 12 EA ----- ----------
00002090: 00 02 12 EA 00 07 12 FC 00 20 12 EC 00 03 12 EC --------- ------
000020A0: 00 02 12 EC 00 01 12 FC 00 21 12 FC 00 21 12 FC ---------!---!--
000020B0: 00 21 FF FF FF FF FF FF FF FF FF FF FF FF FF FF -!--------------
000020C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000020D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000020E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000020F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002100: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002110: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002120: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002130: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002140: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002150: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002160: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002170: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002180: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002190: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000021F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002200: 50 65 74 74 65 72 6E 20 73 65 20 43 4F 43 48 20 Pettern se COCH
00002210: 33 32 31 21 21 21 FF FF FF FF FF FF FF FF FF FF 321!!!----------
00002220: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002230: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002240: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002250: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002260: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002270: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002280: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002290: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
000022F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002300: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002310: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00002320: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
```

C:\Users\dines\Downloads\592 or 320\Labs\Lab Assignment 1\FirstLab.S68

```
D0=00000000  D4=00000000  A0=00000000  A4=00002040       T S  INT   XNZVC     Cycles
D1=00000000  D5=00000000  A1=00002216  A5=00000000  SR=0010000000000000           360
D2=00000000  D6=00000000  A2=00002000  A6=00000000  US=00FF0000
D3=00000000  D7=00000000  A3=00002020  A7=01000000  SS=01000000  PC=000020B6
  Address   --------Code---------   Line -----------Source----------->>
00002040                             11    ORG $2040
00002040= 30 31 32 33 34 35 ...      12    DC.B '0123456789'
0000204A                             13   START:
0000204A  45F8 2000                  14    LEA $2000,A2
0000204E  47F8 2020                  15    LEA $2020,A3
00002052  49F8 2040                  16    LEA $2040,A4
00002056  43F8 2200                  17    LEA $2200,A1
0000205A                             18
0000205A                             19
0000205A  12EA 000F                  20    MOVE.B 15(A2),(A1)+
0000205E  12EB 0004                  21    MOVE.B 4(A3),(A1)+
00002062  12EB 0013                  22    MOVE.B 19(A3),(A1)+
00002066  12EB 0013                  23    MOVE.B 19(A3),(A1)+
0000206A  12EB 0004                  24    MOVE.B 4(A3),(A1)+
0000206E  12EB 0011                  25    MOVE.B 17(A3),(A1)+
00002072  12EB 000D                  26    MOVE.B 13(A3),(A1)+
00002076  12FC 0020                  27    MOVE.B #32,(A1)+
0000207A                             28
0000207A                             29
0000207A  12EB 0012                  30    MOVE.B 18(A3),(A1)+
0000207E  12EB 0004                  31    MOVE.B 4(A3),(A1)+
00002082  12FC 0020                  32    MOVE.B #32,(A1)+
00002086                             33
00002086  12EA 0002                  34    MOVE.B 2(A2),(A1)+
0000208A  12EA 000E                  35    MOVE.B 14(A2),(A1)+
0000208E  12EA 0002                  36    MOVE.B 2(A2),(A1)+
00002092  12EA 0007                  37    MOVE.B 7(A2),(A1)+
00002096  12FC 0020                  38    MOVE.B #32,(A1)+
0000209A                             39
0000209A                             40
0000209A  12EC 0003                  41    MOVE.B 3(A4),(A1)+
0000209E  12EC 0002                  42    MOVE.B 2(A4),(A1)+
000020A2  12EC 0001                  43    MOVE.B 1(A4),(A1)+
000020A6                             44
000020A6                             45
000020A6  12FC 0021                  46    MOVE.B #33,(A1)+
000020AA  12FC 0021                  47    MOVE.B #33,(A1)+
000020AE  12FC 0021                  48    MOVE.B #33,(A1)+
000020B2                             49
000020B2  FFFF FFFF                  50    SIMHALT
000020B6                             51
000020B6                             52    END START
```

```
S0 = 68KPROG   20CREATED BY EASY68K
.S68 file read successful
```

CONCLUSIONS

The program successfully demonstrates memory manipulation using assembly instructions. Proper use of offsets and addressing ensures precise data extraction and insertion. The lab reinforces concepts of low-level programming, such as memory segmentation and byte-level operations.