



Alexandria University
Faculty of Engineering
Computer and Systems Engineering
Department
CSE 233: Computer Organization

Assembly – Lab 4

Team Members

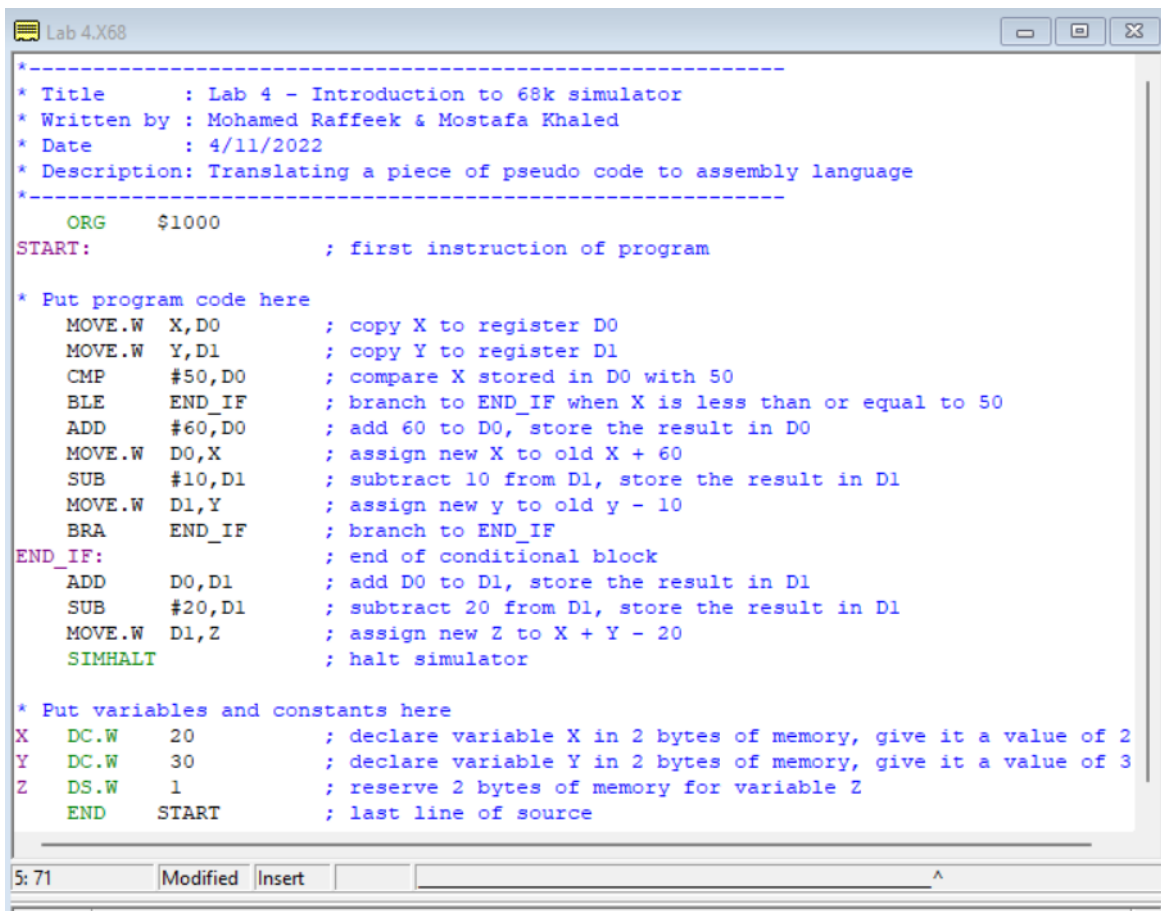
1. ID: 20011574 – محمد رفيق محمد السيد ابراهيم
2. ID: 20011923 – مصطفى خالد كمال احمد زايد

First: Problem Statement

Translating the following pseudo code using Easy68k translator:

```
X = 20;
Y = 30;
If X > 50 :
    X = X + 60
    Y = Y - 10
End If
Z = X + Y - 20
```

Second: Code



```
-----
* Title       : Lab 4 - Introduction to 68k simulator
* Written by  : Mohamed Raffeeek & Mostafa Khaled
* Date       : 4/11/2022
* Description: Translating a piece of pseudo code to assembly language
-----
      ORG      $1000
START:                ; first instruction of program

* Put program code here
      MOVE.W   X,D0      ; copy X to register D0
      MOVE.W   Y,D1      ; copy Y to register D1
      CMP      #50,D0     ; compare X stored in D0 with 50
      BLE      END_IF    ; branch to END_IF when X is less than or equal to 50
      ADD      #60,D0     ; add 60 to D0, store the result in D0
      MOVE.W   D0,X       ; assign new X to old X + 60
      SUB      #10,D1     ; subtract 10 from D1, store the result in D1
      MOVE.W   D1,Y       ; assign new y to old y - 10
      BRA      END_IF     ; branch to END_IF
END_IF:              ; end of conditional block
      ADD      D0,D1      ; add D0 to D1, store the result in D1
      SUB      #20,D1     ; subtract 20 from D1, store the result in D1
      MOVE.W   D1,Z       ; assign new Z to X + Y - 20
      SIMHALT              ; halt simulator

* Put variables and constants here
X      DC.W    20         ; declare variable X in 2 bytes of memory, give it a value of 2
Y      DC.W    30         ; declare variable Y in 2 bytes of memory, give it a value of 3
Z      DS.W    1         ; reserve 2 bytes of memory for variable Z
      END      START     ; last line of source
```

Third: Outputs

C:\Users\moham\Desktop\Lab 4.S68

File Search Run View Options Help

Registers

D0=00000014	D4=00000000	A0=00000000	A4=00000000	T S INT XNZVC	Cycles
D1=0000001E	D5=00000000	A1=00000000	A5=00000000	SR=0010000000000000	78
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=0000103C

Address -----Code----- Line -----Source----->>

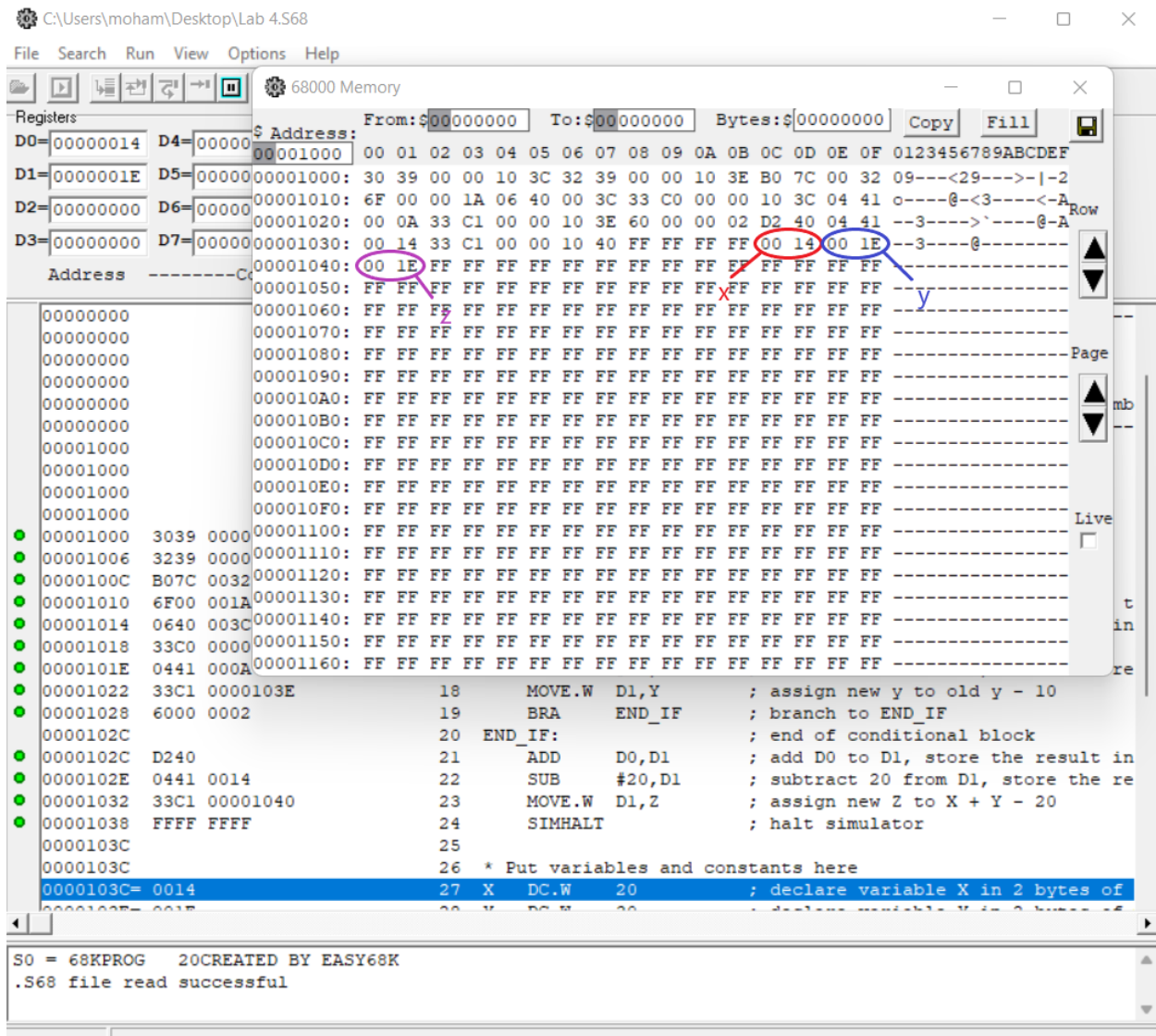
```

00000000      1  *-----
00000000      2  * Title      : Lab 4 - Introduction to 68k simulator
00000000      3  * Written by : Mohamed Raffeeek & Mostafa Khaled
00000000      4  * Date      : 4/11/2022
00000000      5  * Description: Translating a piece of pseudo code to assemb
00000000      6  *-----
00001000      7      ORG      $1000
00001000      8  START:                ; first instruction of program
00001000      9
00001000     10  * Put program code here
00001000     11      MOVE.W  X,D0        ; copy X to register D0
00001006     12      MOVE.W  Y,D1        ; copy Y to register D1
0000100C     13      CMP      #50,D0      ; compare X stored in D0 with 50
00001010     14      BLE      END_IF      ; branch to END_IF when X is less t
00001014     15      ADD      #60,D0      ; add 60 to D0, store the result in
00001018     16      MOVE.W  D0,X        ; assign new X to old X + 60
0000101E     17      SUB      #10,D1      ; subtract 10 from D1, store the re
00001022     18      MOVE.W  D1,Y        ; assign new y to old y - 10
00001028     19      BRA      END_IF      ; branch to END_IF
0000102C     20  END_IF:          ; end of conditional block
0000102C     21      ADD      D0,D1      ; add D0 to D1, store the result in
0000102E     22      SUB      #20,D1      ; subtract 20 from D1, store the re
00001032     23      MOVE.W  D1,Z        ; assign new Z to X + Y - 20
00001038     24      SIMHALT          ; halt simulator
0000103C     25
0000103C     26  * Put variables and constants here
0000103C= 0014 27  X  DC.W  20        ; declare variable X in 2 bytes of
0000103E= 001E 28  Y  DC.W  20        ; declare variable Y in 2 bytes of

```

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

XNZVC
000000 → flags



Memory locations
Every variable takes 2 bytes since it was defined as a word (.W) in the data area of the code