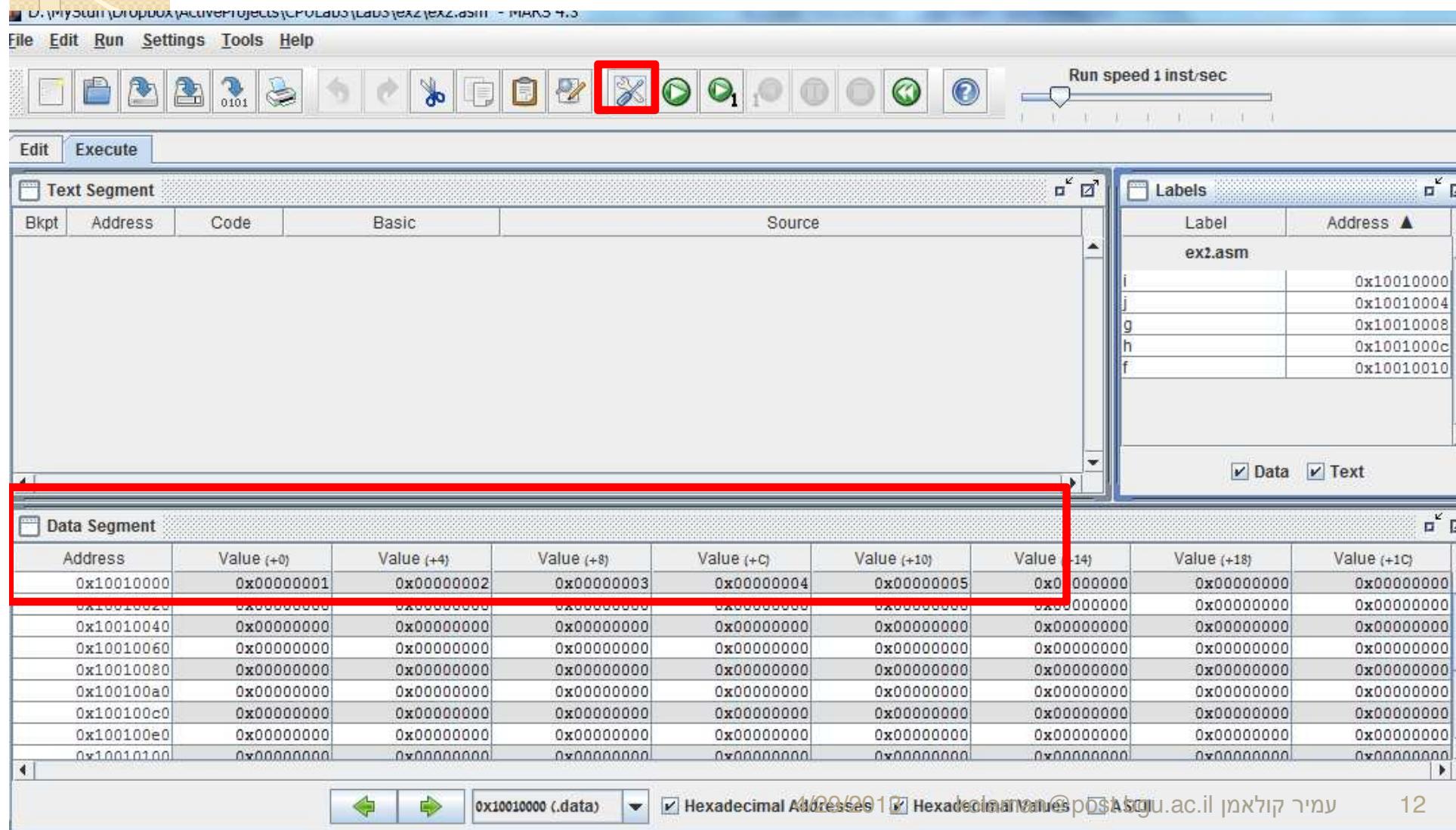


1. Edit display is indicated by highlighted tab.
- 2, 3. Typical edit and execute operations are available through icons and menus, dimmed-out when unavailable or not applicable.
4. WYSIWYG editor for MIPS assembly language code.

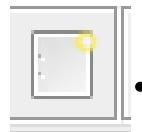
# Data segment and comments

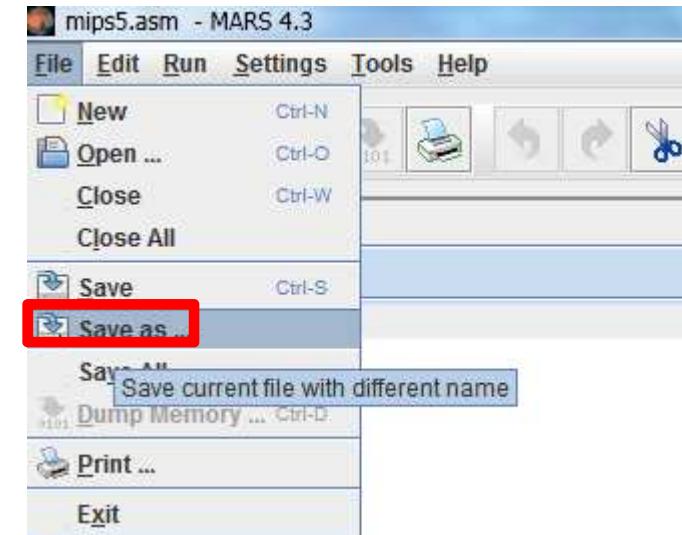
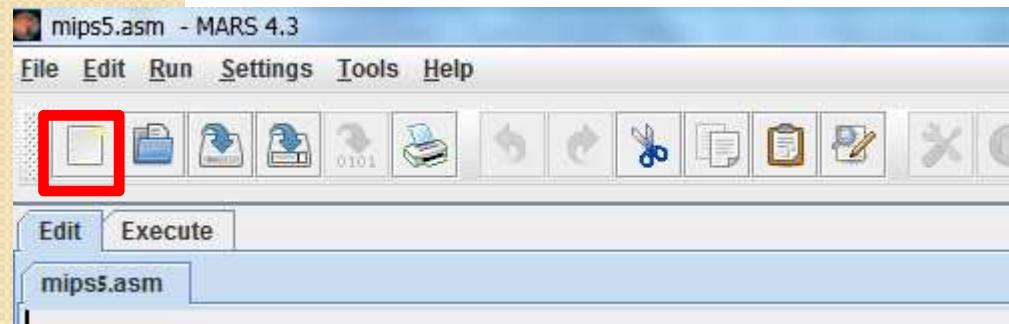
- data part starts with reserved word
- **.data**
- each data array has a **label**, a **type** **reset value** and **range**
- **A: .word 0:10**
- comments are marked with **#**

# After compiling, the address of labels and memory content can be viewed



# Exercise 1

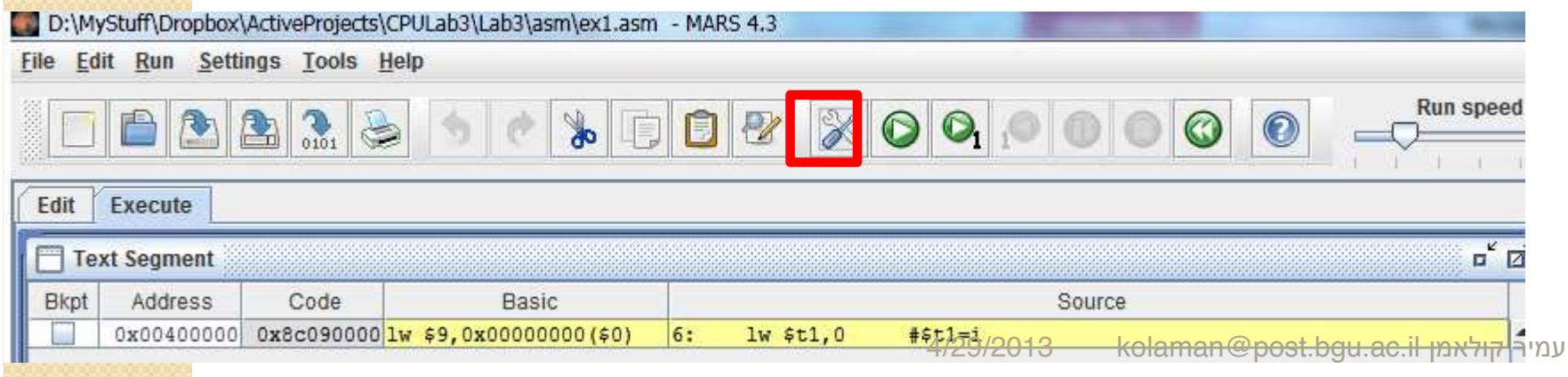
- Download MARS file from the internet.
- Use the menu bar File → New or the icon .
- File → Save as /lab3/ex1/ex1.asm



# Exercise I

Add the following data to the data section

1. initial value of l is 1
  2. initial value of j is 2
  3. initial value of g is 3
  4. initial value of h is 4
  5. initial value of f is 5
- Assemble using Run→Assemble or F3 or the icon



D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm - MARS 4.3

File Edit Run Settings Tools Help

Bkpt Address Code Basic Source

6: lw \$t1,0 #st1=i

4/29/2013 kolaman@post.bgu.ac.il

# Exercise I

- What is address of the label value i?
- Settings → Show Labels Window

The screenshot shows the MARS 4.3 assembly debugger interface. At the top, the 'Settings' menu is open, with the 'Show Labels Window (symbol table)' option highlighted and surrounded by a red box. Below the menu, the 'Execute' tab is selected in the toolbar. In the bottom right corner, there is a 'Labels' window containing a table:

Label	Address
ex1.asm	
i	

The 'Text Segment' window shows assembly code:

```
0x8e090000 lw $9,0x00000000($16) 11: lw $t1,0($s0)##t1=i
```

The status bar at the bottom displays the date '4/29/2013' and the email 'kolaman@post.bgu.ac.il'.

# Exercise I

- What is the starting address of the data segment?
- View memory content

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010100	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005
0x10010104	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010108	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001010C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010110	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010114	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010118	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001011C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

# Exercise 2

- Go to MARS window
- Use the menu bar File → Open or the Open icon 
- Open /Lab3/ex2/ex2.asm



D:\MyStuff\Dropbox\ActiveProjects\CPULat

File Edit Run Settings Tools Help

ex2.asm

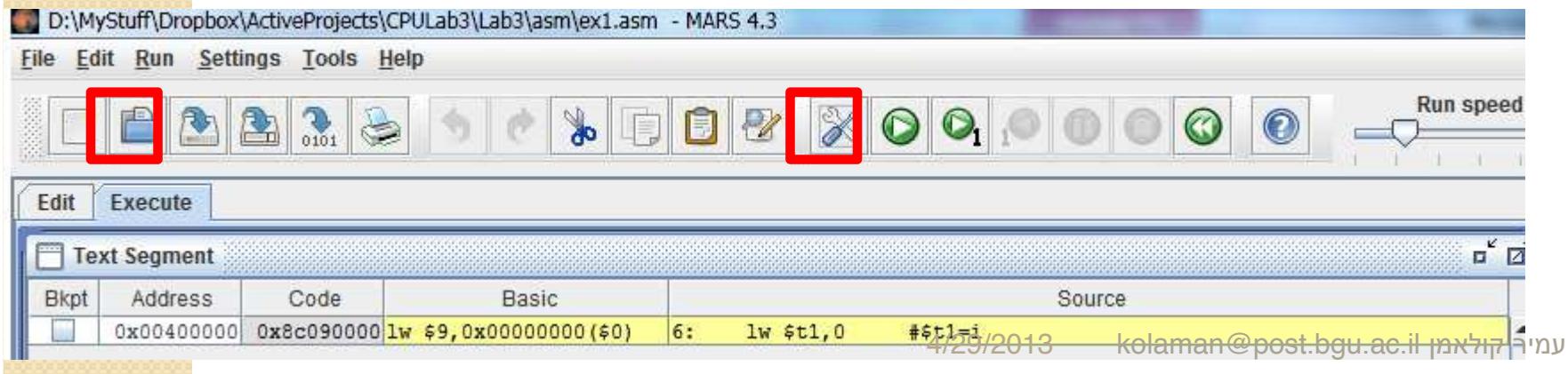
```
1 .data
2 i: .word 1
3 j: .word 2
4 g: .word 3
5 h: .word 4
6 f: .word 5
7 .text
```

# Exercise 2

- Assume that data part starts at address 0.
- write down the following computation in MIPS assembly:
  - $\$t1 = i$
  - $\$t2 = j$
  - $\$t3 = g$
  - $\$t4 = h$

# Exercise 3

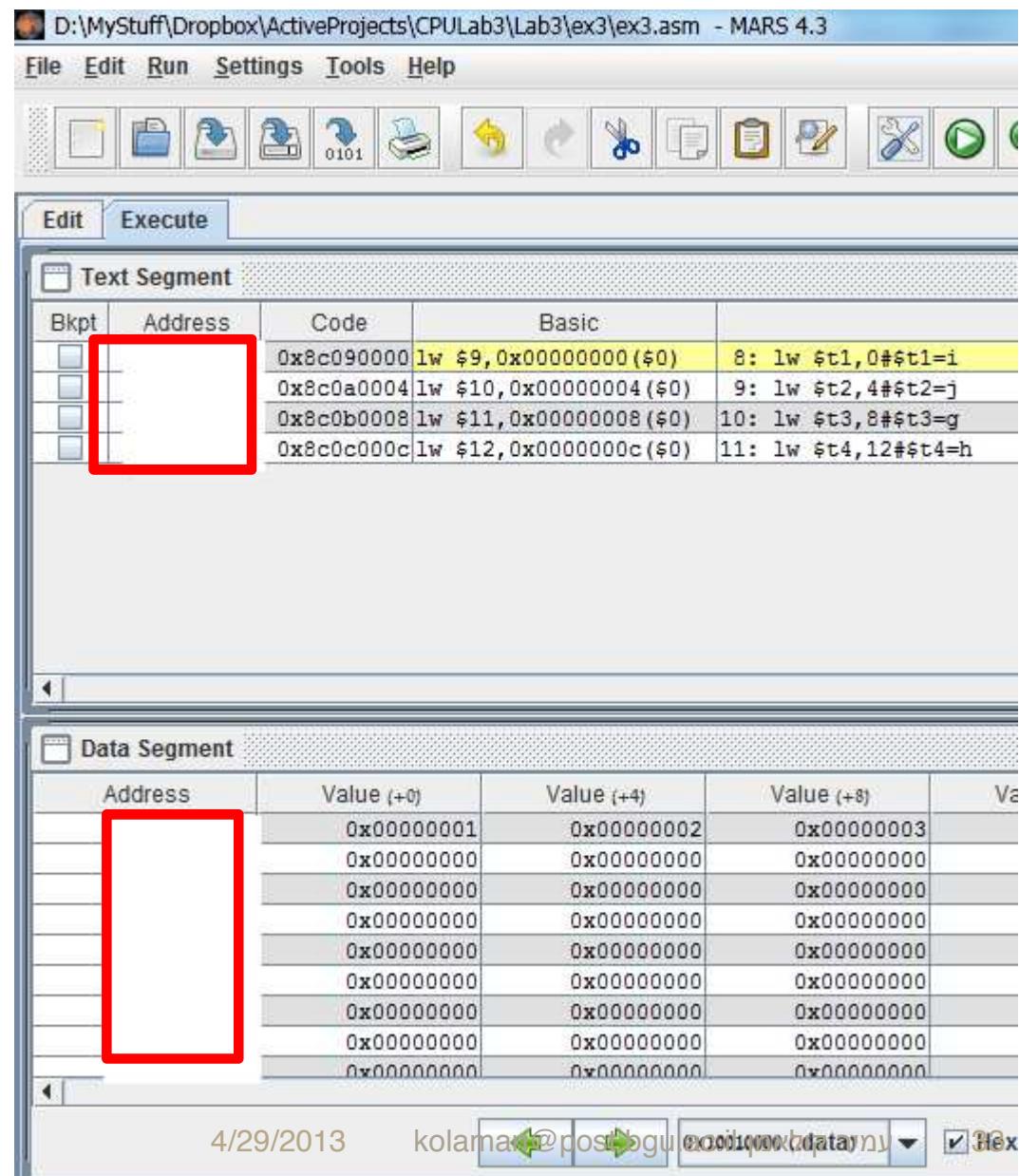
- Open MARS
- Use the menu bar File→Open or the Open icon 
- Open /lab3/ex3/ex3.asm
- Assemble using Run→Assemble or F3 or the icon 



# Exercise 3

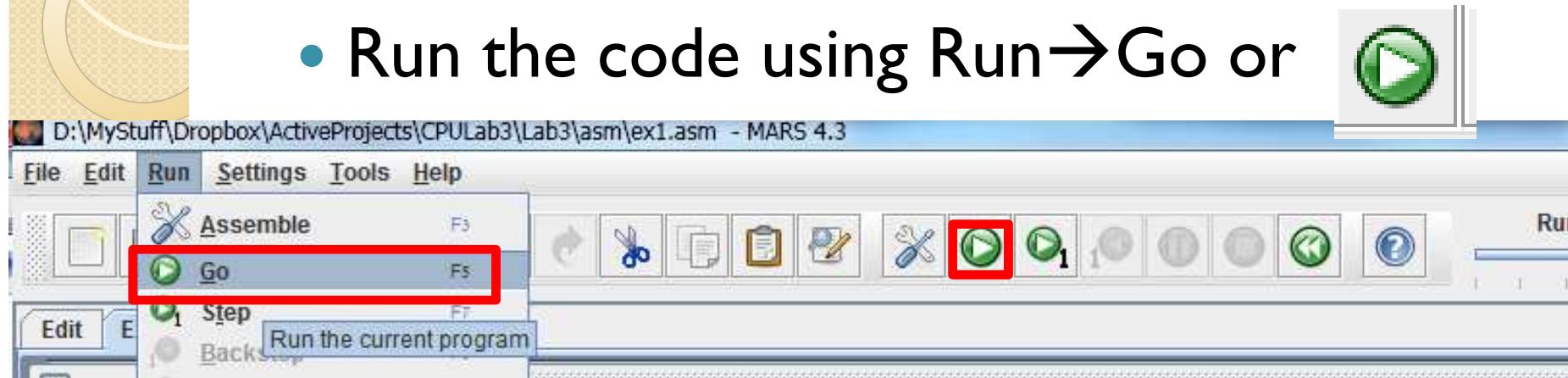
- What is the starting address of the data segment?

Text  
segment  
(program)?



# Exercise 3

- Run the code using Run → Go or



- What is the error in Mars Messages tab?

A screenshot of the MARS 4.3 software interface showing the "Mars Messages" tab. The tab is highlighted with a red box. The messages listed are:

```
Assemble: assembling D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm
Assemble: operation completed successfully.

Go: running ex1.asm

Go: execution terminated by null instruction.

Assemble: assembling D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm
Assemble: operation completed successfully.

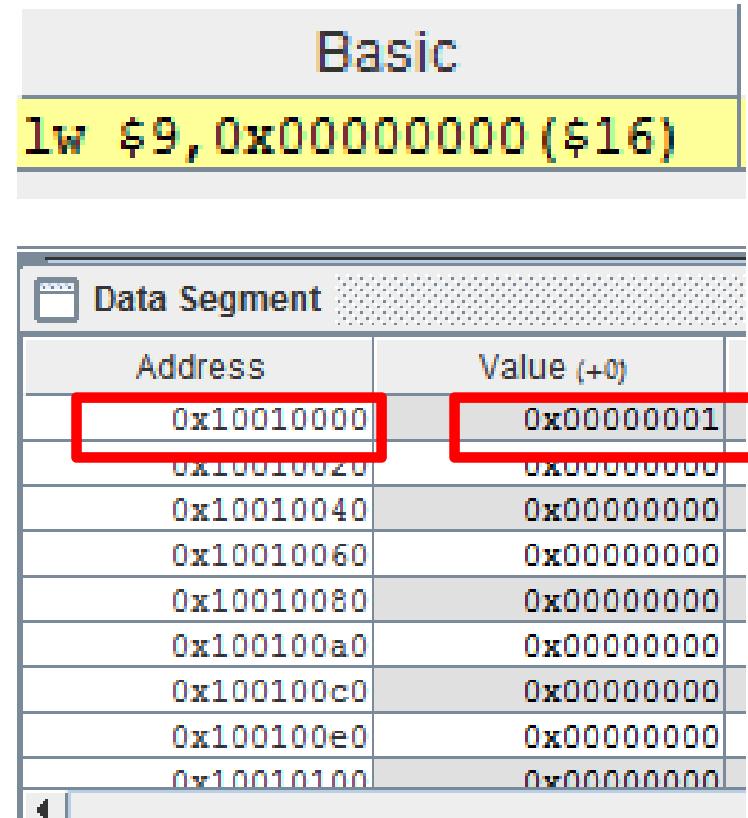
Go: running ex1.asm

Error in D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm line 11: Runtime exception at 0x00400000: address out of range 0x00000000
Go: execution terminated with errors.
```

The last message, "Error in D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm line 11: Runtime exception at 0x00400000: address out of range 0x00000000", is highlighted with a red box.

# Exercise 3

- Did we try reading data from the correct location of i?
- Try solving this using memory configuration

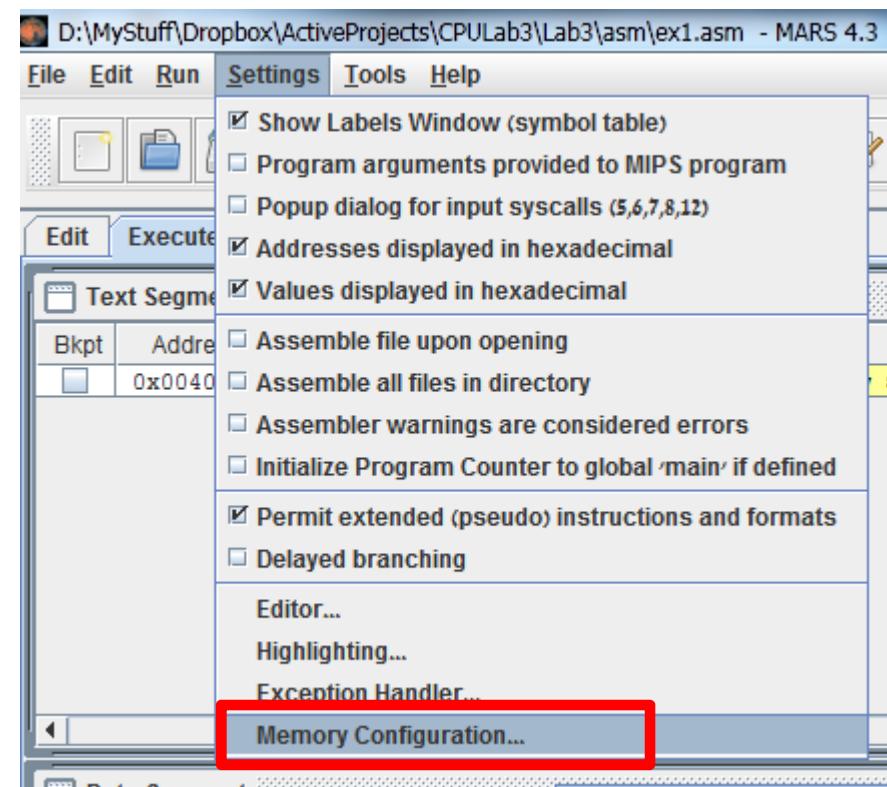


The screenshot shows a debugger interface. At the top, there is an assembly instruction: `lw $9, 0x00000000 ($16)`. Below it is a memory dump titled "Data Segment". The dump lists memory addresses from `0x10010000` to `0x10010100`, with their corresponding values. The first entry, `0x10010000` with value `0x00000001`, is highlighted with red boxes around both the address and the value.

Address	Value (+0)
0x10010000	0x00000001
0x10010020	0x00000000
0x10010040	0x00000000
0x10010060	0x00000000
0x10010080	0x00000000
0x100100a0	0x00000000
0x100100c0	0x00000000
0x100100e0	0x00000000
0x10010100	0x00000000

# Exercise 3

- Possible solution:
  - I. Change data address to start at 0 using Settings→Memory Configuration



# Exercise 3

- Run the code using Run→Go or

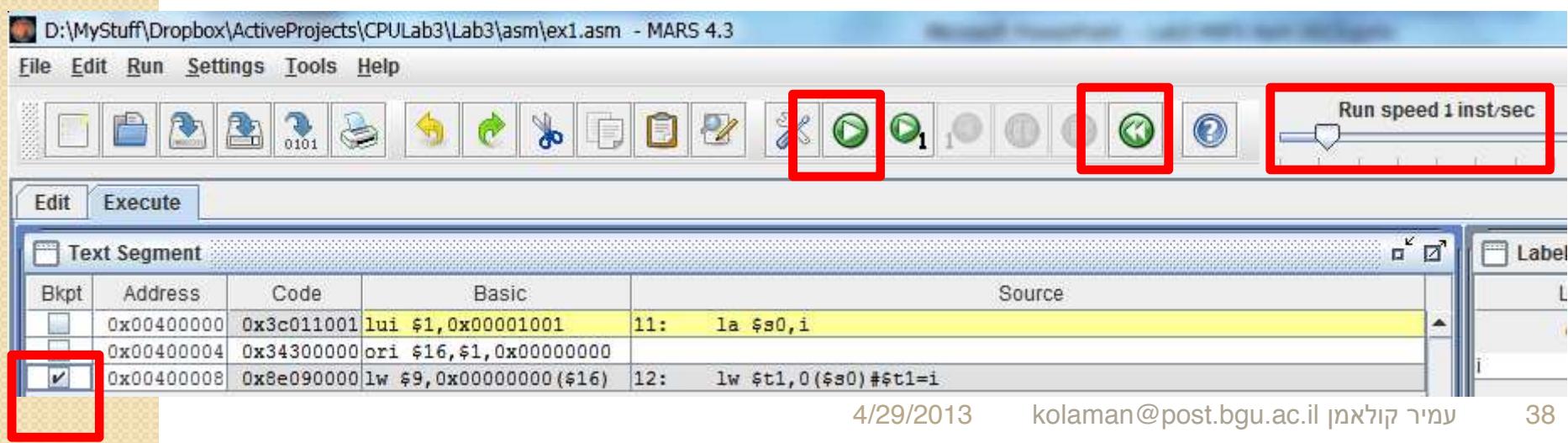


- What is the value of register \$t1?

A screenshot of the MARS 4.3 software interface after running the code. The window title is "D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm - MARS 4.3". The menu bar shows File, Edit, Run, Settings, Tools, Help. The "Edit" and "Execute" tabs are selected. In the center, there are three panes: "Text Segment", "Labels", and "Registers". The "Registers" pane shows a table of registers with their names, numbers, and values. The register \$t1 is highlighted with a red box, showing its value as 0x00000000. Other registers listed include \$zero, \$at, \$v0, \$v1, \$a0, \$a1, \$a2, \$a3, \$t0, and \$t2.

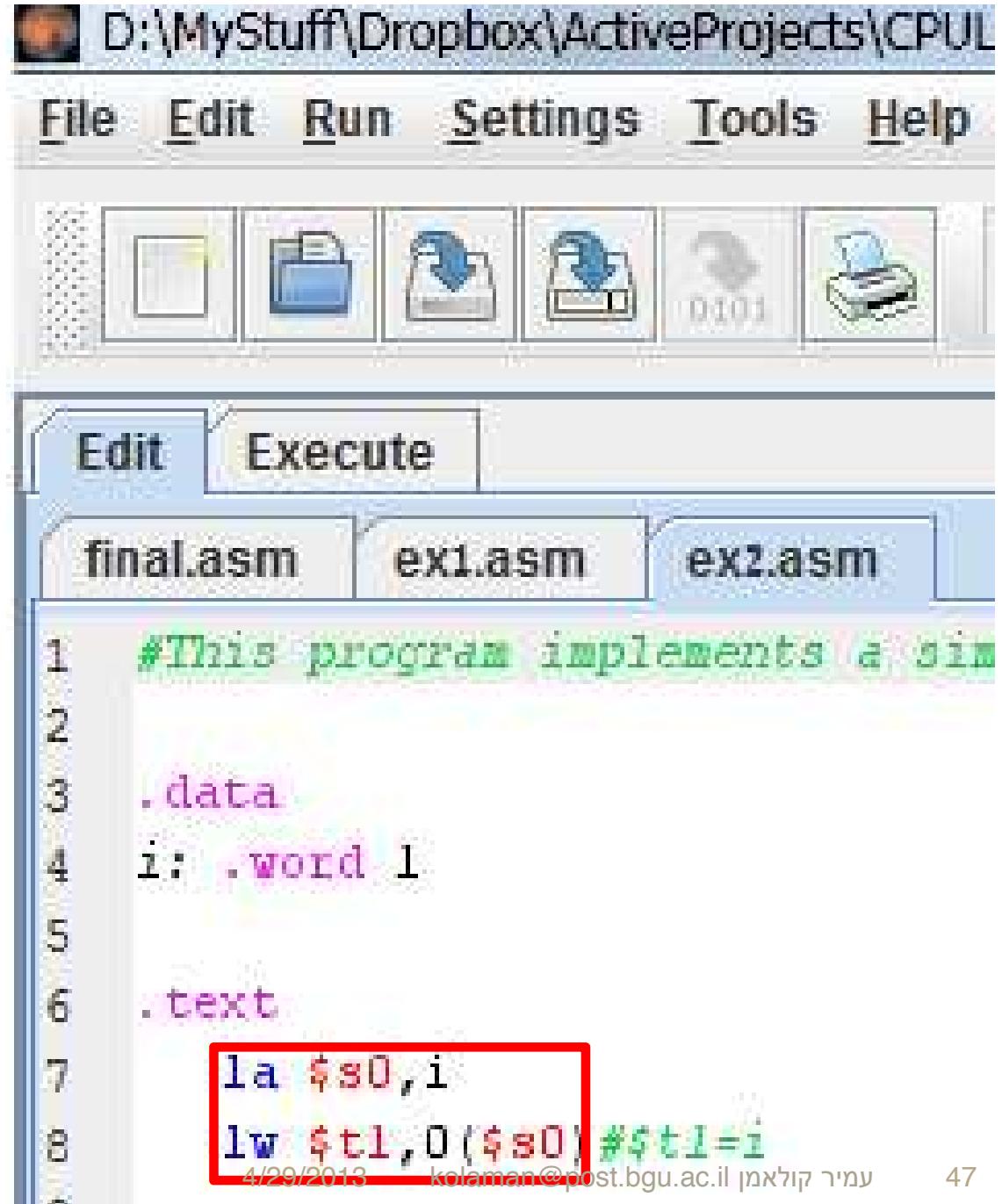
# Exercise 3

- Reset the run
- Change the run speed to 1 inst/sec
- Run the code
- Toggle a break point
- Reset and run again



# Exercise 3

- Better solution:
  2. Use relative addressing.  
By loading label address into \$s0 and referencing it.



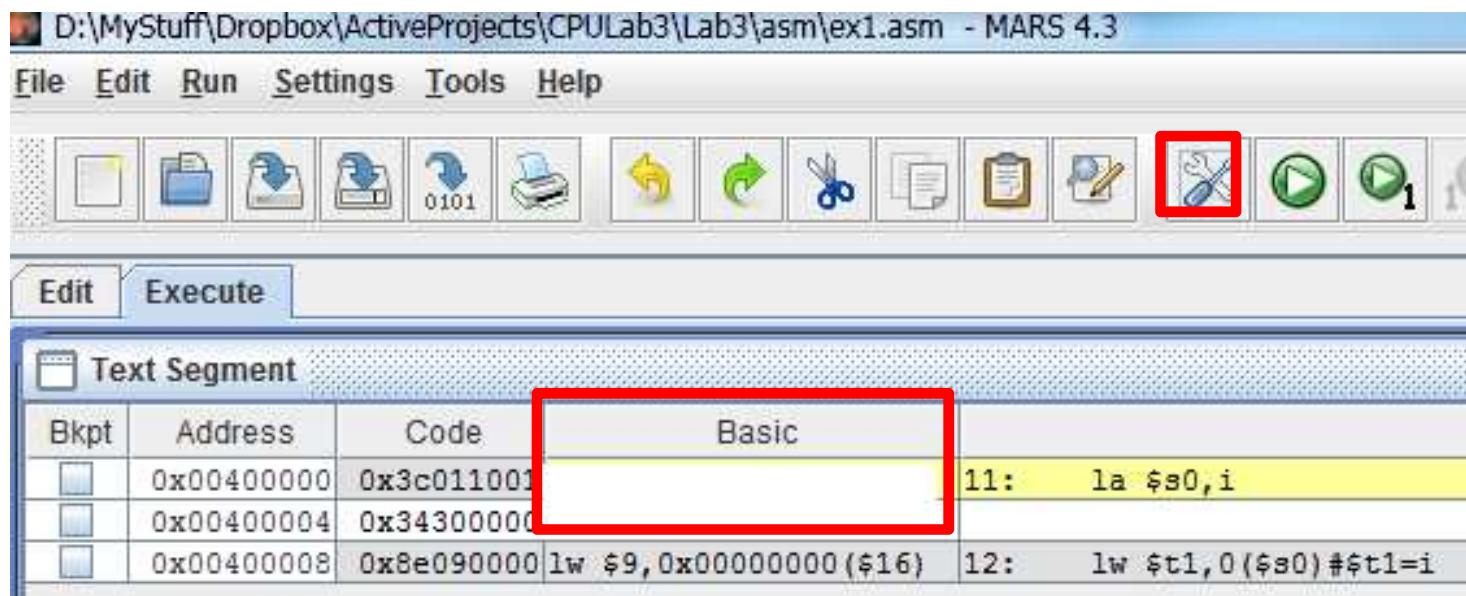
The screenshot shows a window titled 'D:\MyStuff\Dropbox\ActiveProjects\CPUL'. The menu bar includes File, Edit, Run, Settings, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main area has tabs for 'Edit' (selected) and 'Execute'. Under the tabs are three files: 'final.asm', 'ex1.asm', and 'ex2.asm', with 'ex2.asm' currently selected. The code editor displays the following assembly code:

```
1 #This program implements a sim
2
3 .data
4 i: .word 1
5
6 .text
7 la $s0,i
8 lw $t1,0($s0) #$t1=i
```

A red box highlights the instruction 'la \$s0,i' at line 7. The bottom status bar shows the date '4/29/2013' and the author's email 'kolamar@post.bgu.ac.il'. The right side of the status bar contains Hebrew text: 'עמיר קולארן' and 'kolamar@post.bgu.ac.il'.

# Exercise 3

- Assemble using Run → Assemble or F3 or the icon 
- What is the real instruction performing la?



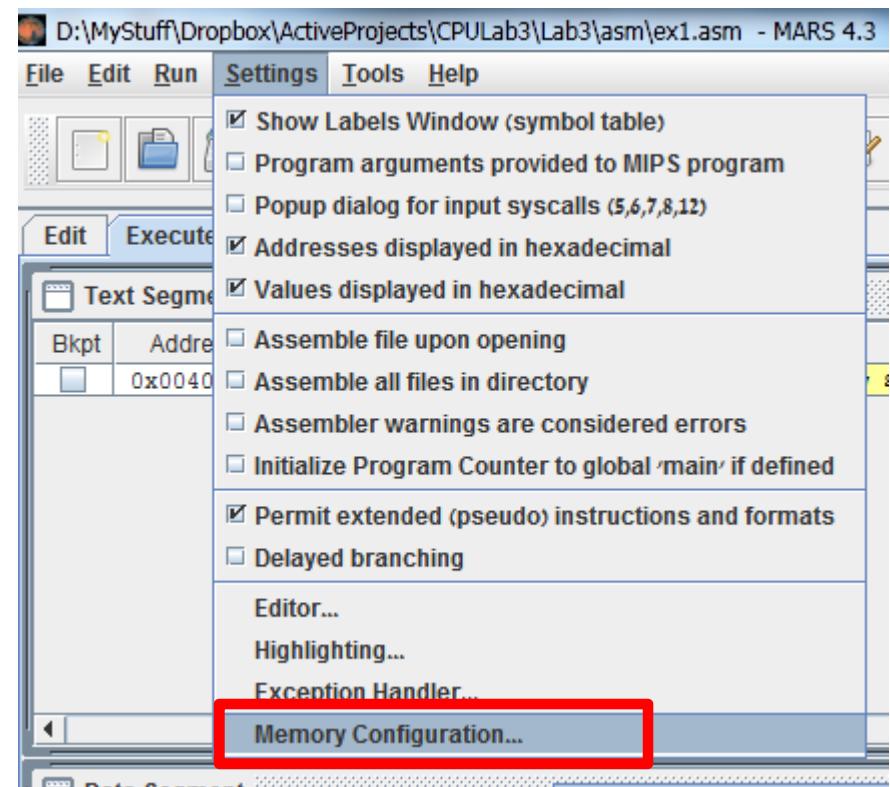
The screenshot shows the MARS 4.3 assembly editor interface. The title bar reads "D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm - MARS 4.3". The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar below has various icons, with the one for assembly (wrench and screwdriver) highlighted by a red box. The main window displays the assembly code in a table:

Text Segment			
Bkpt	Address	Code	Basic
	0x00400000	0x3c011001	11: la \$s0,i
	0x00400004	0x34300000	
	0x00400008	0x8e090000	12: lw \$t1,0(\$s0)##t1=i

The "Basic" column is highlighted by a red box. The assembly code rows are numbered 11 and 12.

# Exercise 3

Change data address to its default value using Settings → Memory Configuration



# Exercise 3

- What is the real instruction performing la?
- Why is it different from the previous memory setting?

The screenshot shows the MARS 4.3 assembly debugger interface. The window title is "D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\asm\ex1.asm - MARS 4.3". The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar contains various icons for file operations like Open, Save, and Print, along with a red square icon for assembly, which is highlighted with a red box. Below the toolbar is a tab bar with "Edit" and "Execute" selected. The main area displays the assembly code in a table:

Text Segment			
Bkpt	Address	Code	Basic
	0x00400000	0x3c011001	11: la \$s0,i
	0x00400004	0x34300000	
	0x00400008	0x8e090000	12: lw \$t1,0(\$s0)##t1=i

The "Basic" column is highlighted with a red box. The assembly code rows are also highlighted with yellow boxes.

# Exercise 4

- Open MARS
- Use the menu bar  
File→Open or the  
Open icon 
- Open  
/lab3/ex4/ex4.asm



D:\MyStuff\Dropbox\ActiveProject

File Edit Run Settings Tools

Open Save Save As Open Recent

Edit Execute

ex4.asm Final.asm ex2.asm

```
1 .data
2 i: .word 1
3 j: .word 2
4 g: .word 3
5 h: .word 4
6 f: .word 5
7 .text
8 la $s0,i
9 lw $t1,0($s0) #t1=i
10 lw $t2,4($s0) #t2=j
11 lw $t3,8($s0) #t3=g
12 lw $t4,12($s0) #t4=h
```

עמיר קולאמן post.bgu.ac.il 4/29/2013 55

## Exercise 4

- Set \$t0 to one if  $\$t1 < \$t2$
- if \$t0 is equal to zero jump to label ELSE
- write label IF: at line 15
- write label ELSE: at line 17
- write label END: at line 18



# Exercise 5

- Open MARS
- Use the menu bar

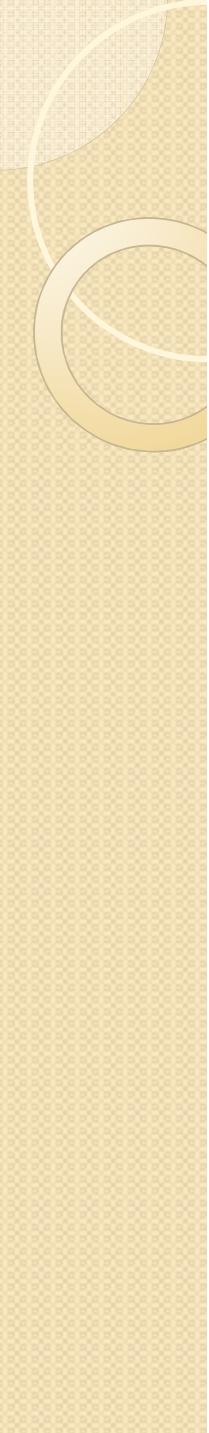
File → Open  
or the Open icon



- Open  
/lab3/ex5/ex5.  
asm

The screenshot shows the MARS 4.3 assembly editor interface. The title bar reads "D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\ex5\ex5.asm\* - MARS 4.3". The menu bar includes File, Edit, Run, Settings, Tools, and Help. A toolbar with various icons is located above the menu. Below the menu is a tab bar with "Edit" and "Execute" tabs, and tabs for files: "ex5.asm\*", "Final.asm", "ex2.asm", "ex4.asm", and "ex3.asm". The main window displays assembly code:

```
1 .data
2     i: .word 1
3     j: .word 2
4     g: .word 3
5     h: .word 4
6     f: .word 5
7 .text
8     la $s0,i
9     lw $t1,0($s0)#$t1=i
10    lw $t2,4($s0)#$t2=j
11    lw $t3,8($s0)#$t3=g
12    lw $t4,12($s0)#$t4=h
13    slt $t0,$t1,$t2 #if i<j then $t0=1
14    beq $t0,$zero,ELSE #if i>=j then go to else part
15 IF:   |
16
17 ELSE:
18 END:
```



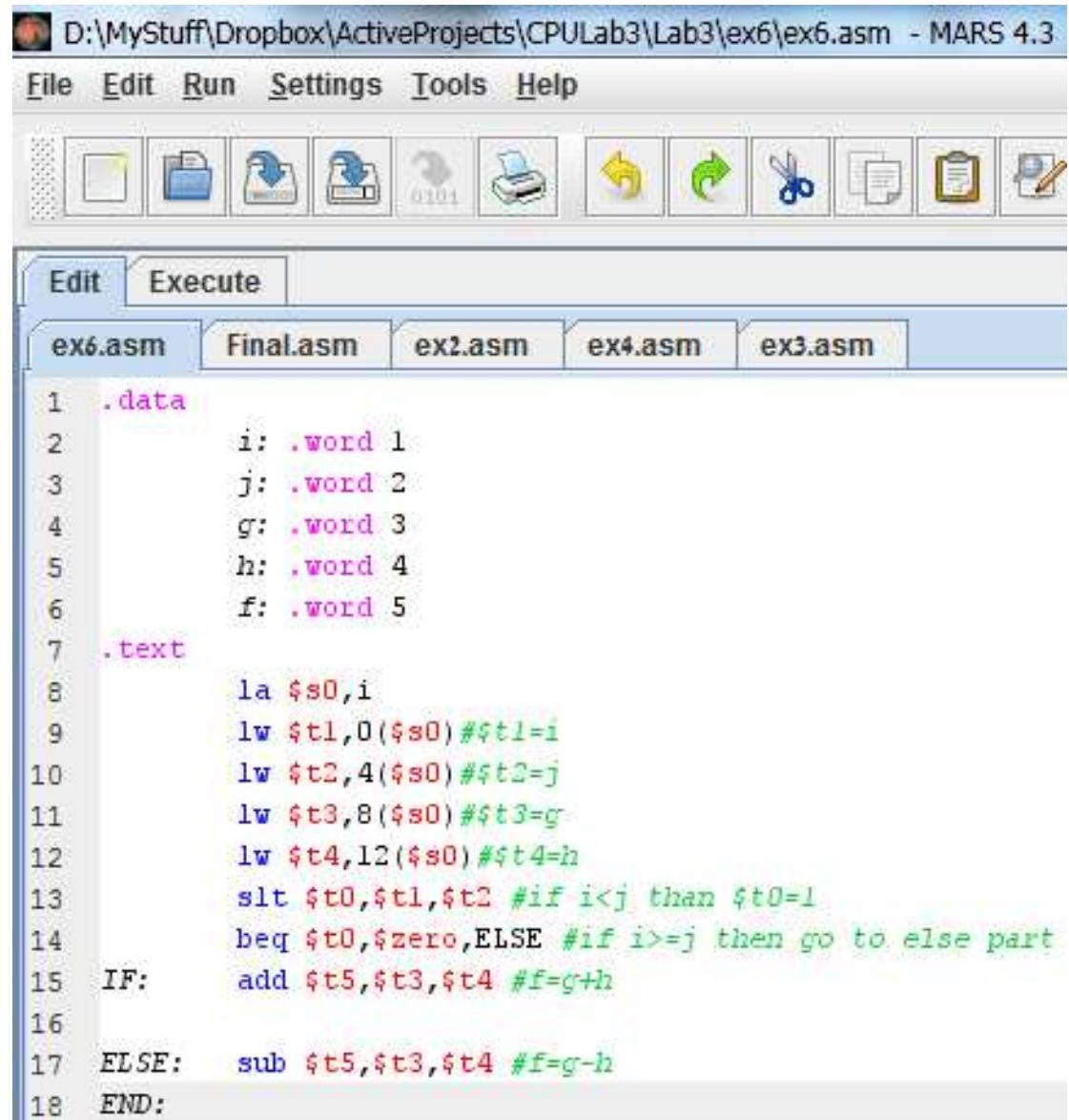
## Exercise 5

- Add to line 15
- $\$t5=\$t3+\$t4$
- Add to line 17
- $\$t5=\$t3-\$t4$



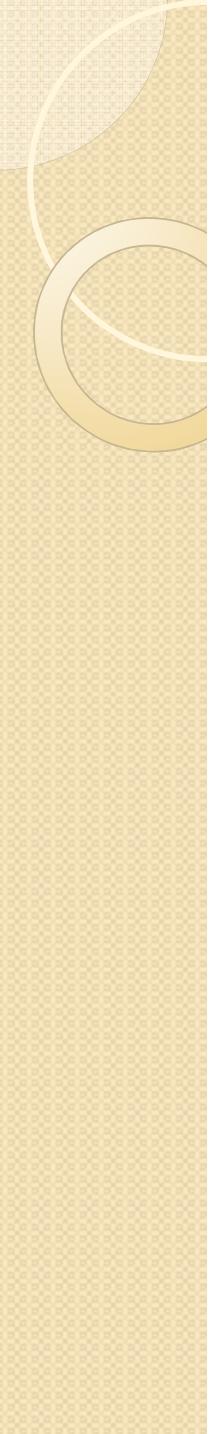
# Exercise 6

- Open MARS
- Use the menu bar  
File → Open  
or the Open icon 
- Open /lab3/ex6/ex6.asm



The screenshot shows the MARS 4.3 assembly editor interface. The title bar reads "D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\ex6\ex6.asm - MARS 4.3". The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Execute. Below the toolbar is a tab bar with tabs for "Edit" and "Execute", and files "ex6.asm", "Final.asm", "ex2.asm", "ex4.asm", and "ex3.asm". The main window displays the assembly code:

```
1 .data
2     i: .word 1
3     j: .word 2
4     g: .word 3
5     h: .word 4
6     f: .word 5
7 .text
8     la $s0,i
9     lw $t1,0($s0) #t1=i
10    lw $t2,4($s0) #t2=j
11    lw $t3,8($s0) #t3=g
12    lw $t4,12($s0) #t4=h
13    slt $t0,$t1,$t2 #if i<j then t0=1
14    beq $t0,$zero,ELSE #if i>=j then go to else part
15 IF:   add $t5,$t3,$t4 #f=g+h
16
17 ELSE: sub $t5,$t3,$t4 #f=g-h
18 END:
```

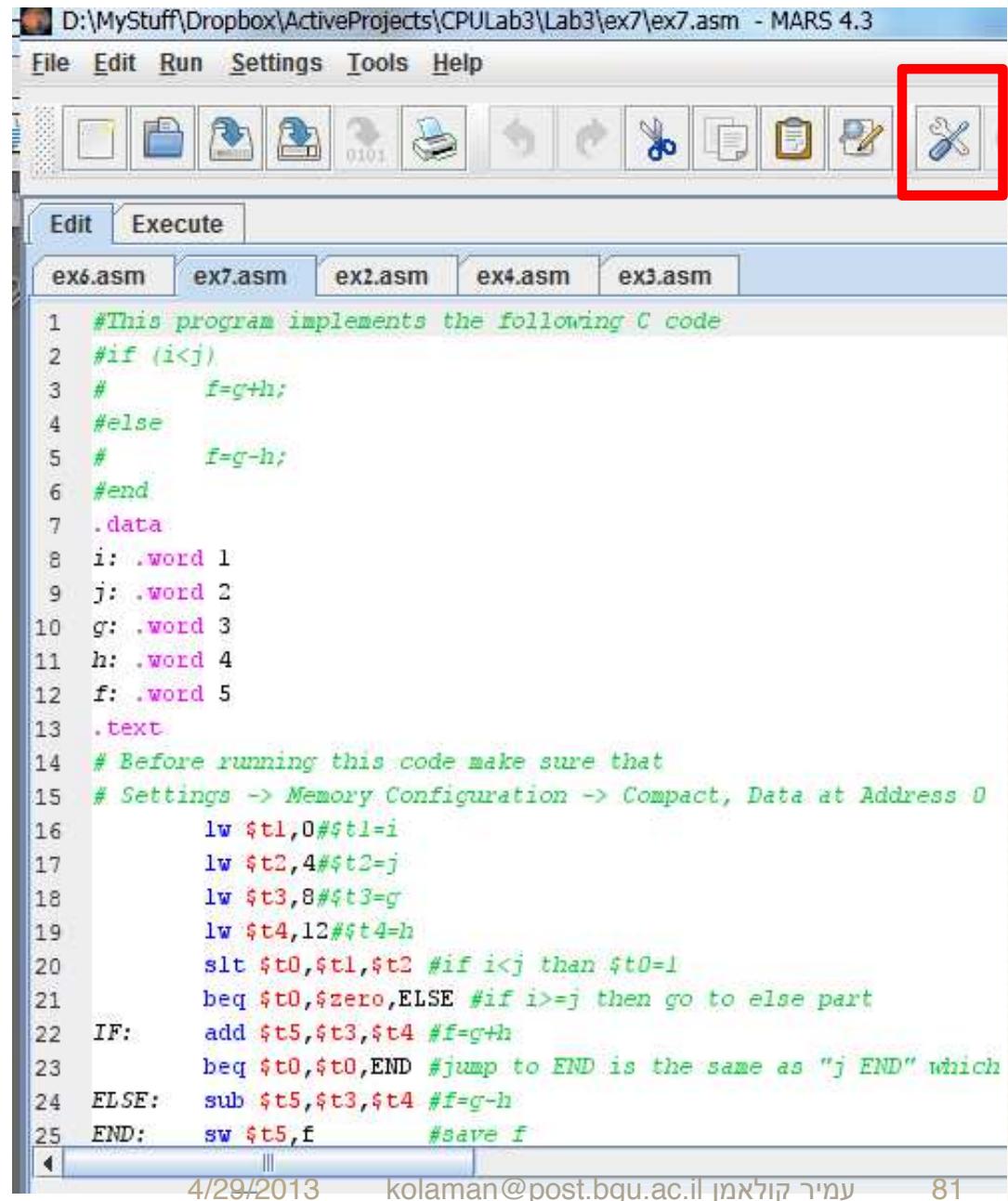


## Exercise 6

- Add an automatic jump from line 16 to line 18
- save the result stored in \$t5 to address of label f

# Exercise 7

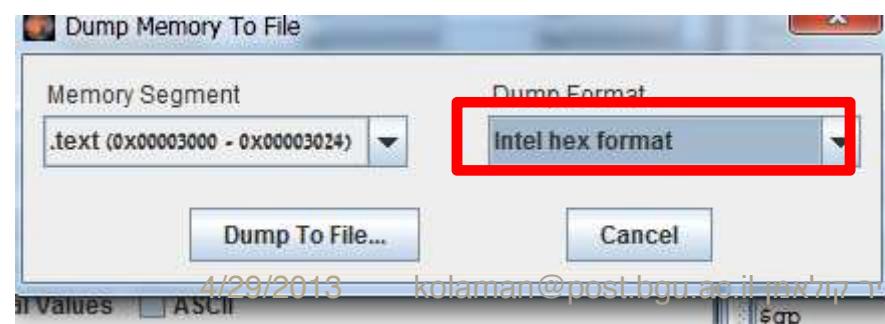
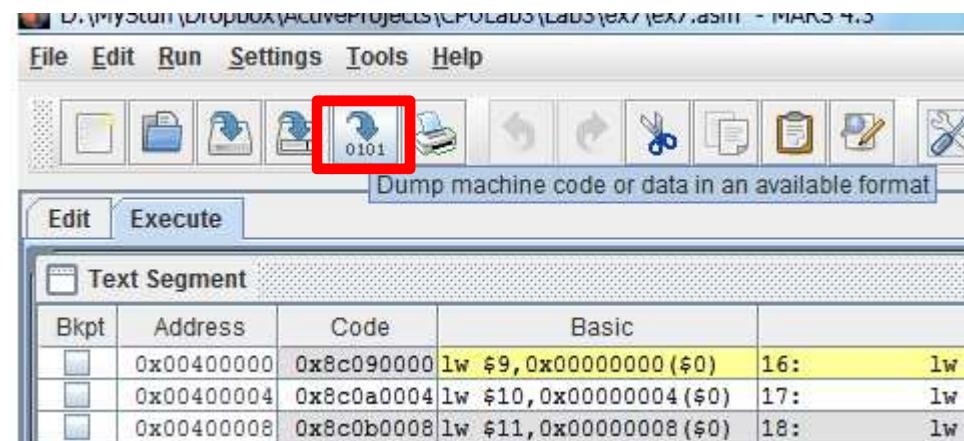
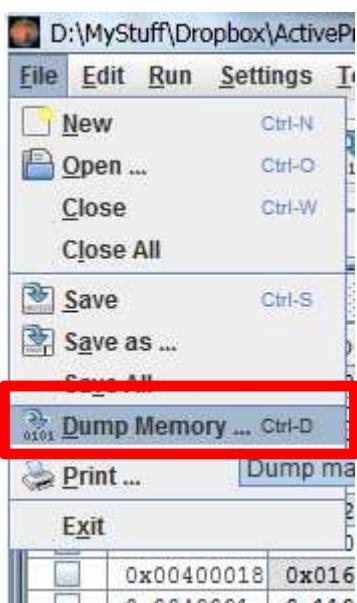
- Open MARS
- Use the menu bar  
File → Open or the Open icon 
- Open /lab3/ex7/ex7.asm
- Assemble the code



```
D:\MyStuff\Dropbox\ActiveProjects\CPULab3\Lab3\ex7\ex7.asm - MARS 4.3
File Edit Run Settings Tools Help
Edit Execute
ex6.asm ex7.asm ex2.asm ex4.asm ex3.asm
1 #This program implements the following C code
2 #if (i<j)
3 #    f=g+h;
4 #else
5 #    f=g-h;
6 #end
7 .data
8 i: .word 1
9 j: .word 2
10 g: .word 3
11 h: .word 4
12 f: .word 5
13 .text
14 # Before running this code make sure that
15 # Settings -> Memory Configuration -> Compact, Data at Address 0
16 lw $t1,0#$t1=i
17 lw $t2,4#$t2=j
18 lw $t3,8#$t3=g
19 lw $t4,12#$t4=h
20 slt $t0,$t1,$t2 #if i<j then $t0=1
21 beq $t0,$zero,ELSE #if i>=j then go to else part
22 IF: add $t5,$t3,$t4 #f=g+h
23 beq $t0,$t0,END #jump to END is the same as "j END" which
24 ELSE: sub $t5,$t3,$t4 #f=g-h
25 END: sw $t5,f      #save f
```

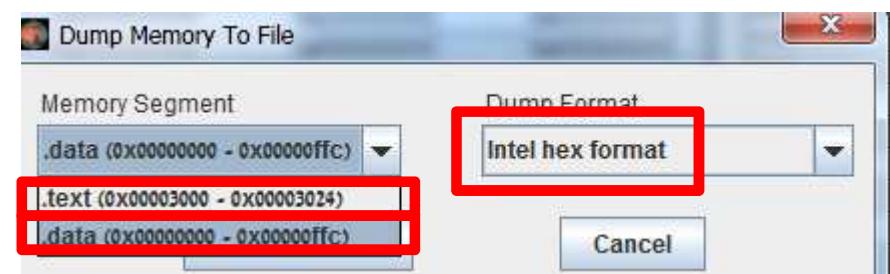
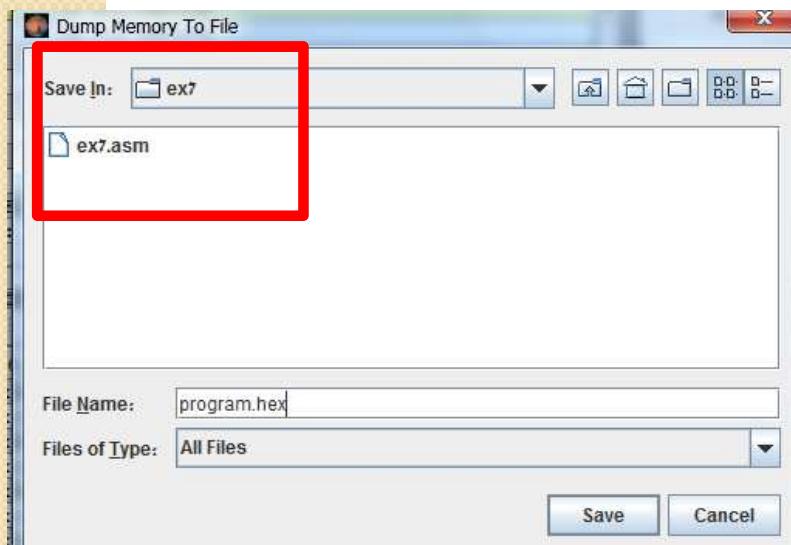
# Exercise 7

- Export text and data using Dump Memory option.
- Make sure Intel hex format



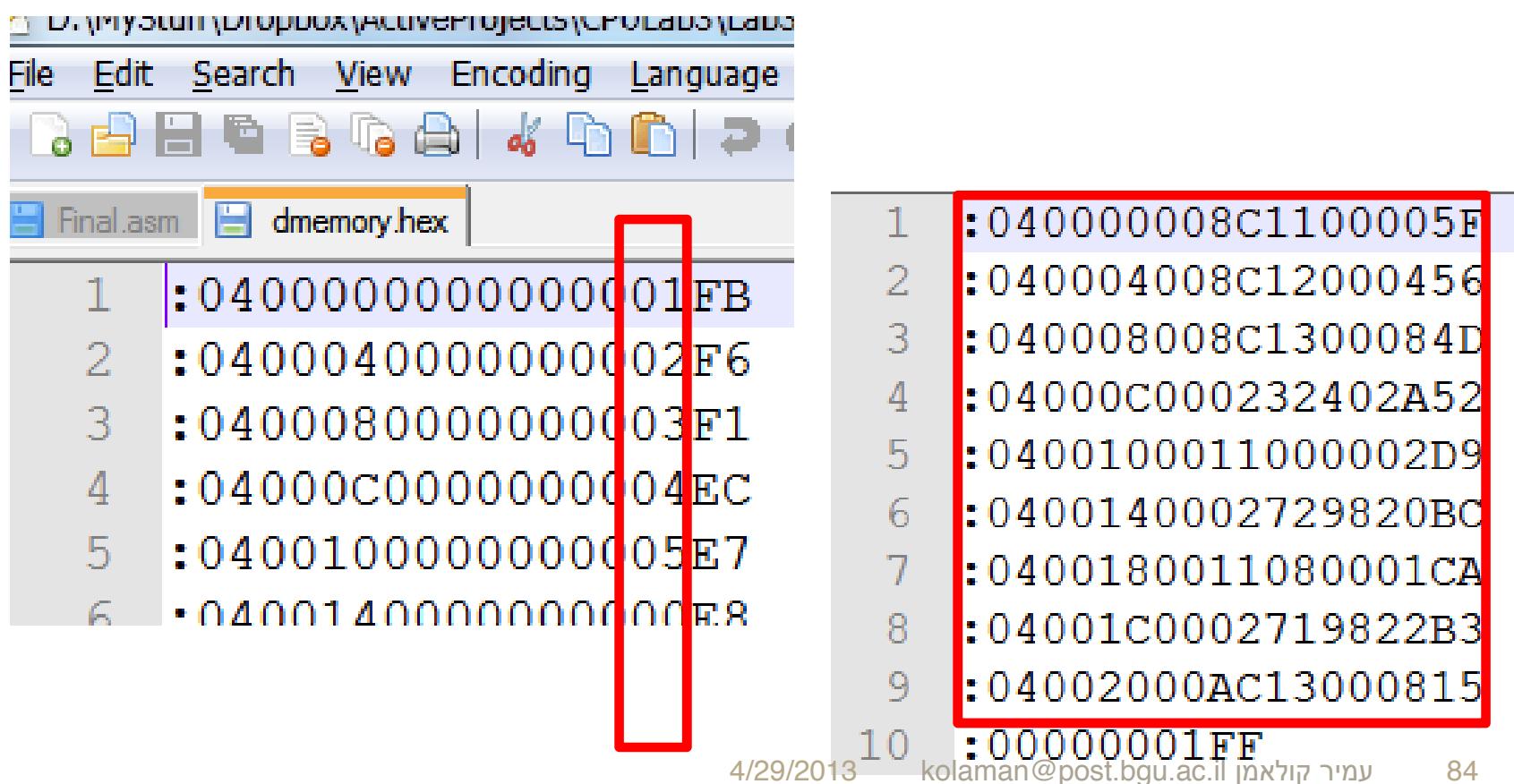
# Exercise 7

- Memory Segment .data save as **dmemory.hex**
- Memory Segment .text save as **program.hex**
- Make sure Intel hex format



# Exercise 7

- Open dmemory.hex in a text editor
- Open as program.hex in a text editor



```
File Edit Search View Encoding Language
Final.asm dmemory.hex
1 :0400000000000001FB
2 :0400040000000002F6
3 :0400080000000003F1
4 :04000C0000000004EC
5 :0400100000000005E7
6 :0400140000000006E8
7 :0400000000000007FF
8 :040004000000000856
9 :04000800000000094D
10 :04000C000000000A52
11 :040010001100000B9
12 :0400140002729820BC
13 :0400180011080001CA
14 :04001C0002719822B3
15 :04002000AC13000815
16 :00000001FF
```

4/29/2013 kolaman@post.bgu.ac.il 84

# Exercise 7

- find command
- **lw \$12,c**
- **beq \$8,\$8, l**
- **add \$13,\$11,\$12**

1	:040000008C09000067
2	:040004008C0A00045E
3	:040008008C0B000855
4	:04000C008C0C000C4C
5	:04001000012A402A57
6	:040014001100000215
7	:04001800016C6820FF
8	:04001C0011080001C6
9	:04002000016C6822E5
10	:04002400AC0D00100F
11	:00000001FF
12	

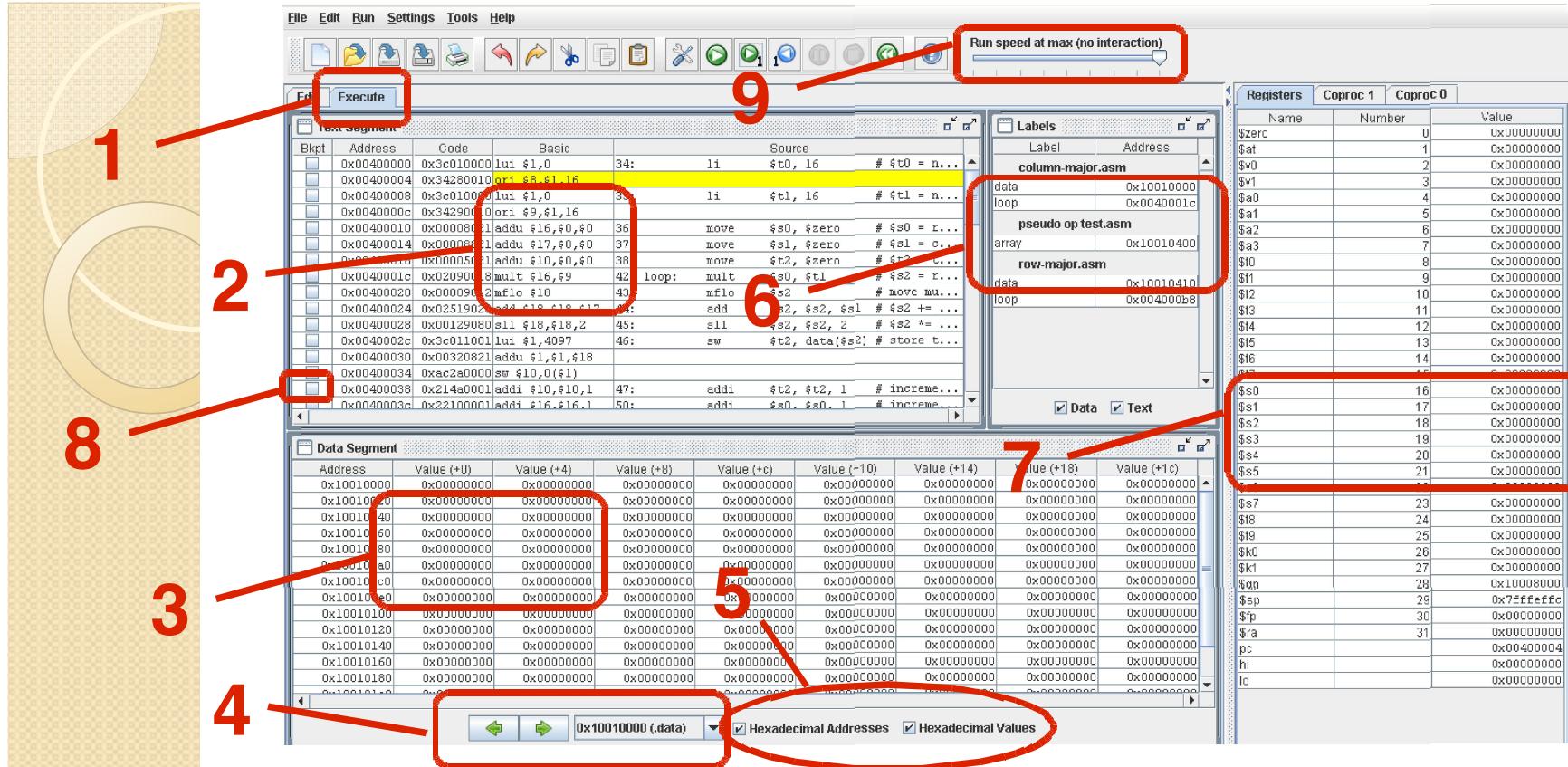
# Exercise 7

- open for edit  
lab3/ex7/rtlMIPS/dmemory.vhd
- lab3/ex7/rtlMIPS/ifetch.vhd
- find the reference to  
we just saved.

```
inst_memory: altsyncram

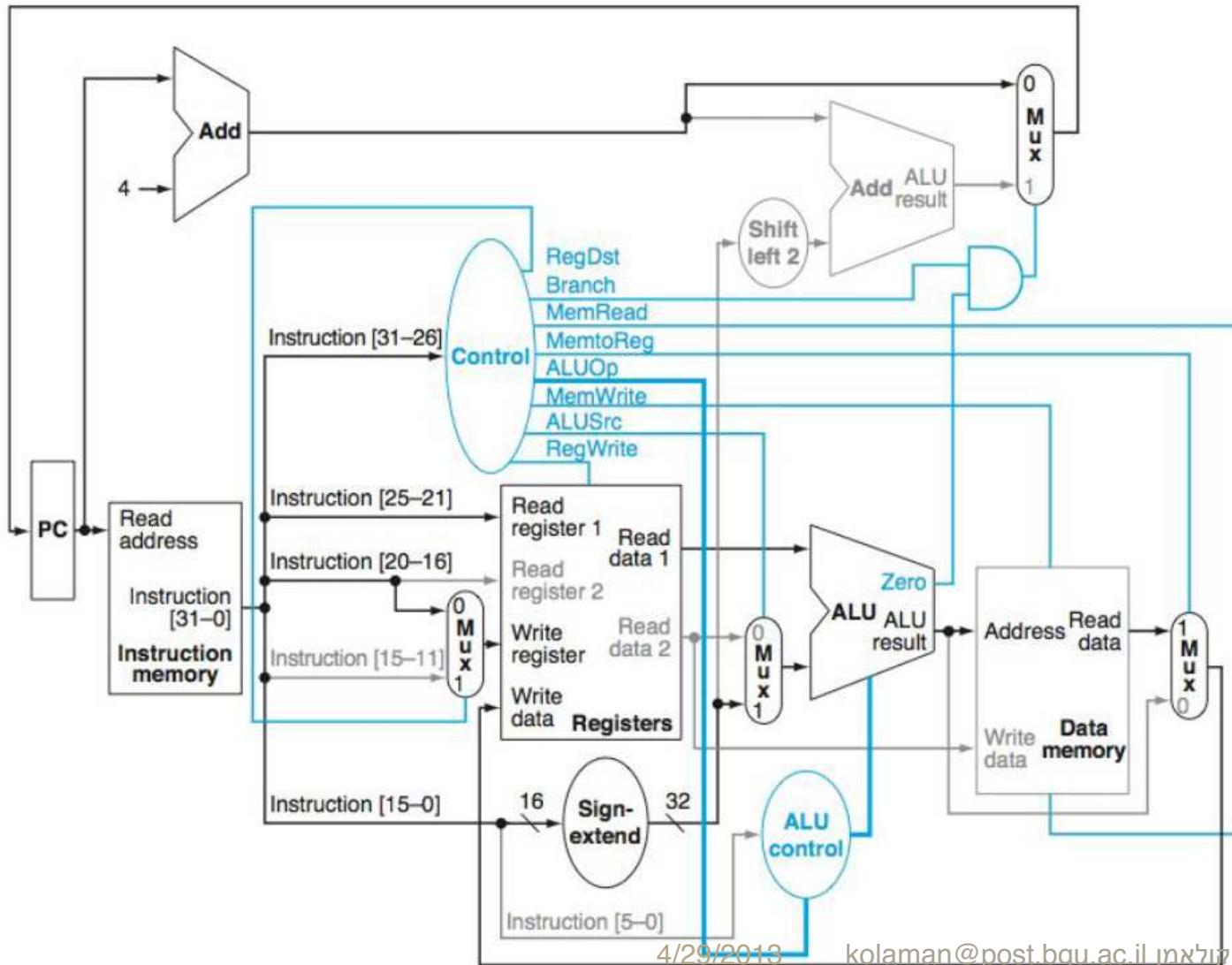
GENERIC MAP (
    operation_mode => "ROM",
    width_a => 32,
    widthad_a => 8,
    lpm_type => "altsyncram",
    outdata_reg_a => "UNREGISTERED",
    init_file => "program.hex",
    intended_device_family => "Cyclone"
)
```

```
data_memory : altsyncram
GENERIC MAP (
    operation_mode => "SINGLE_PORT",
    width_a => 32,
    widthad_a => 8,
    lpm_type => "altsyncram",
    outdata_reg_a => "UNREGISTERED",
    init_file => "dmemory.hex",
    intended_device_family => "Cyclone"
)
PORT MAP (
    wren_a => memwrite,
    clock0 => write_clock,
    address_a => address,
    data_a => write_data,
    q_a => read_data );
```

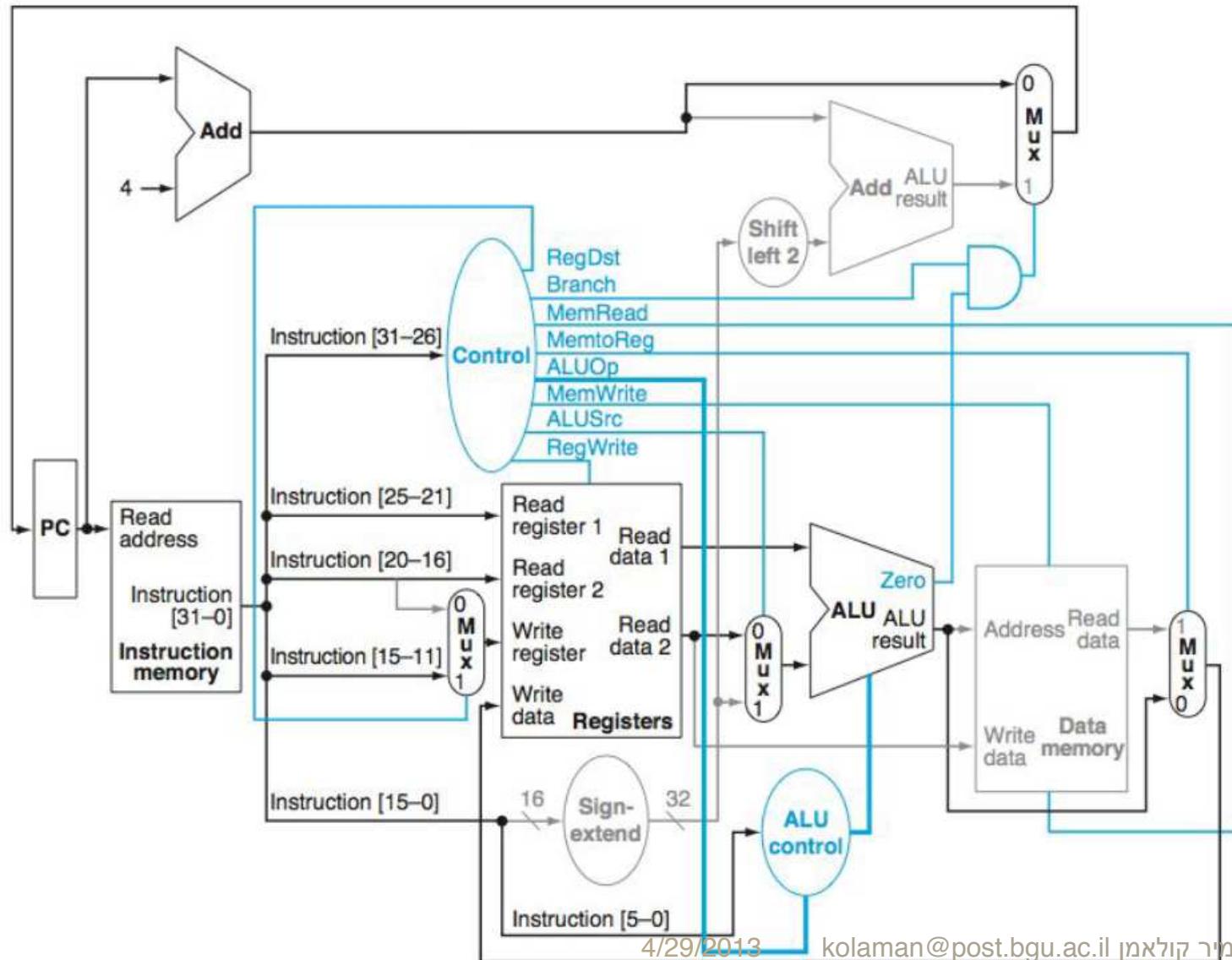


1. Execute display is indicated by highlighted tab.
2. Assembly code is displayed with its address, machine code, assembly code, and the corresponding line from the source code file. (Source code and assembly code will differ when pseudoinstructions have been used.)
3. The values stored in Memory are directly editable (similar to a spreadsheet).
4. The window onto the Memory display is controlled in several ways: previous/next arrows and a menu of common locations (e.g., top of stack).
5. The numeric base used for the display of data values and addresses (memory and registers) is selectable between decimal and hexadecimal.
6. Addresses of labels and data declarations are available. Typically, these are used only when single-stepping to verify that an address is as expected.
7. The values stored in Registers are directly editable (similar to a spreadsheet).
8. Breakpoints are set by a checkbox for each assembly instruction. These checkboxes are always displayed and available.
9. Selectable speed of execution allows the user to “watch the action” instead of the assembly program finishing directly.

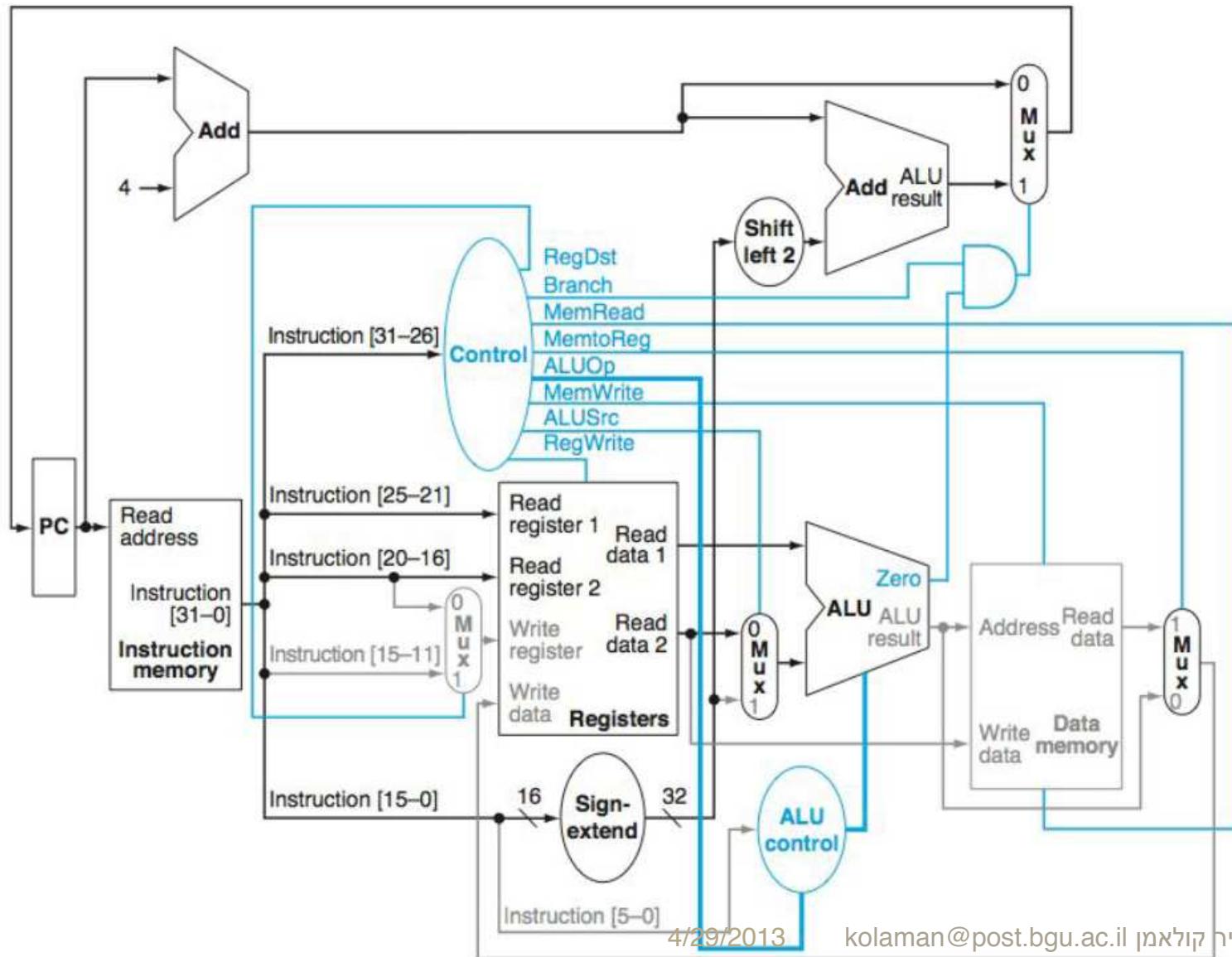
# Load Instruction



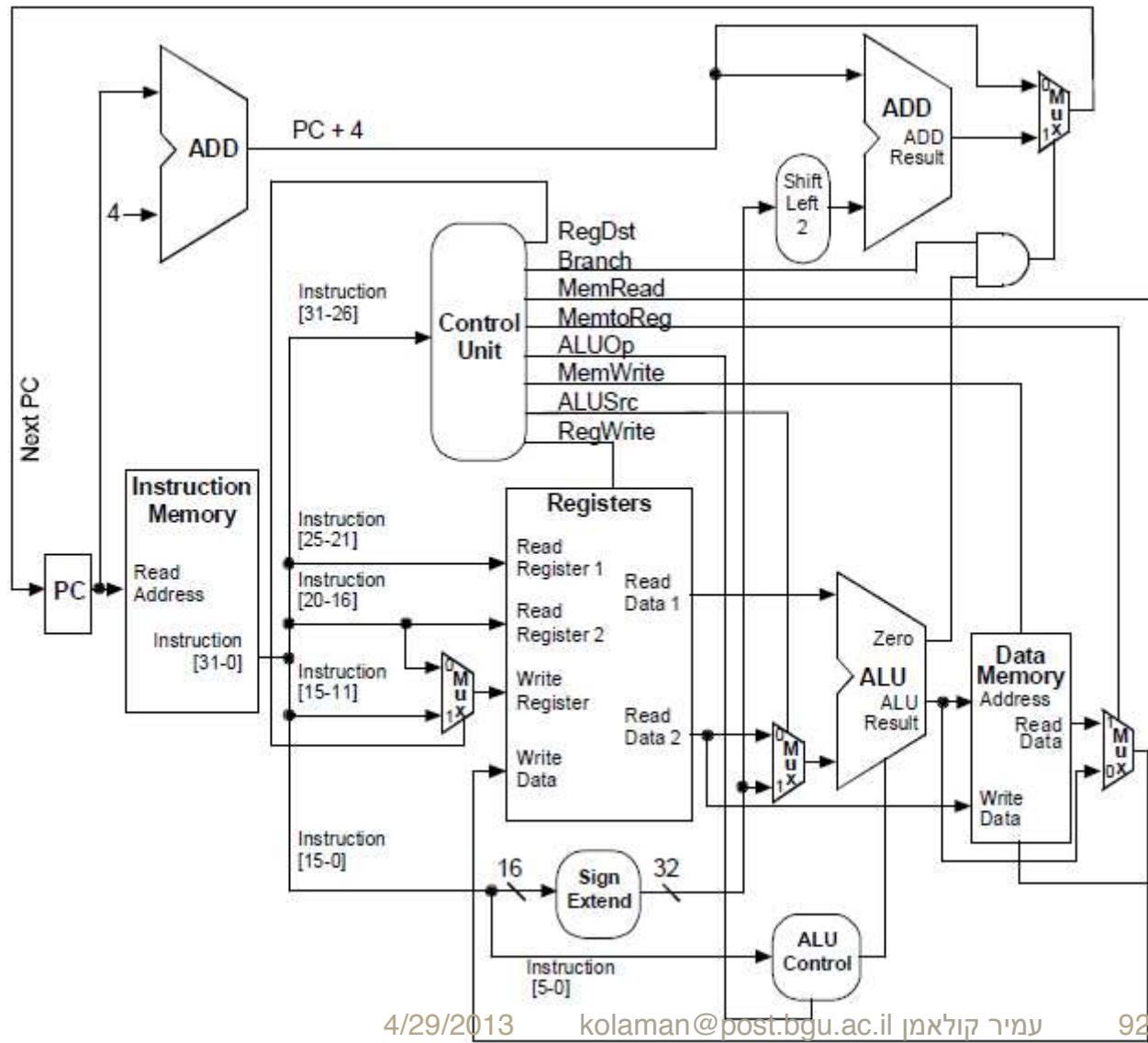
# R-type instruction



# Branch equal Instruction

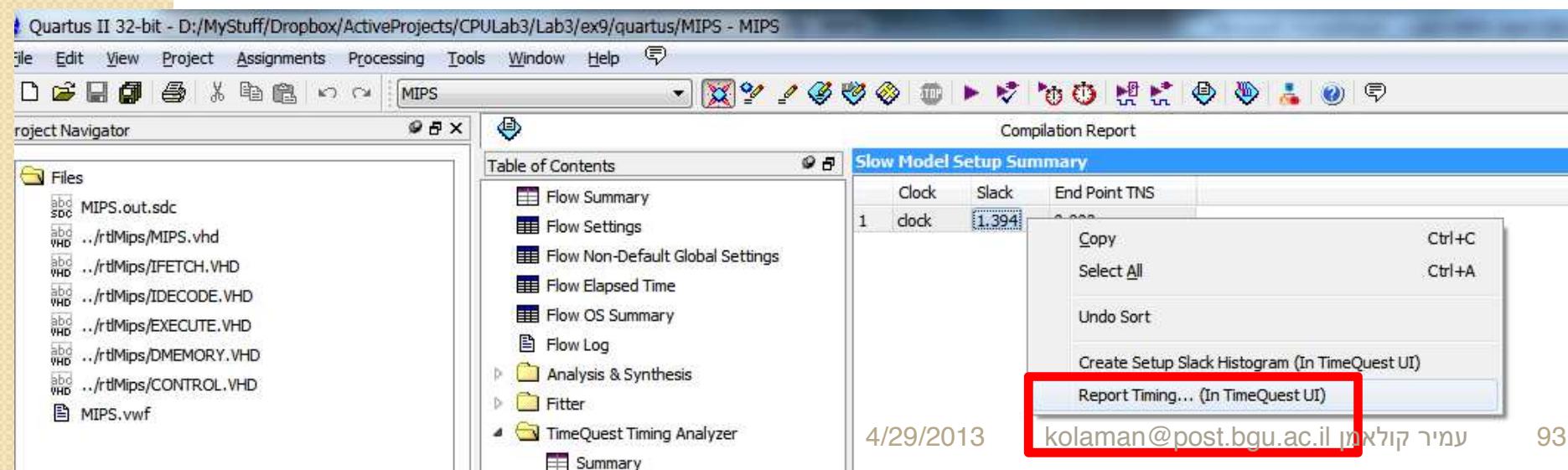
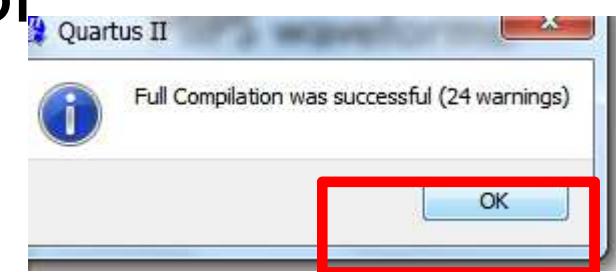
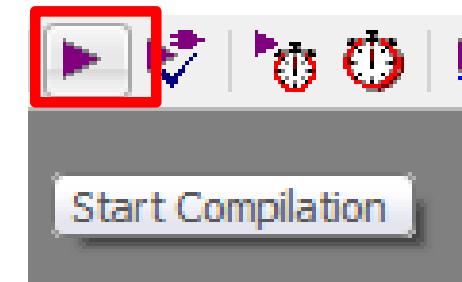


# Full diagram of our MIPS



# Exercise 9

- Open Quartus project in lab3/ex9/quartus/MIPS.qpf
- Compile the design.
- Create a timing report for slow model



Quartus II 32-bit - D:/MyStuff/Dropbox/ActiveProjects/CPULab3/Lab3/ex9/quartus/MIPS - MIPS

File Edit View Project Assignments Processing Tools Window Help

MIPS

Project Navigator

Files

- MIPS.out.sdc
- ../rtlMips/MIPS.vhd
- ../rtlMips/IFETCH.VHD
- ../rtlMips/IDECODE.VHD
- ../rtlMips/EXECUTE.VHD
- ../rtlMips/DMEMORY.VHD
- ../rtlMips/CONTROL.VHD
- MIPS.vwf

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- TimeQuest Timing Analyzer
- Summary

Compilation Report

Slow Model Setup Summary

Clock	Slack	End Point TNS
clock	1.394	0.000

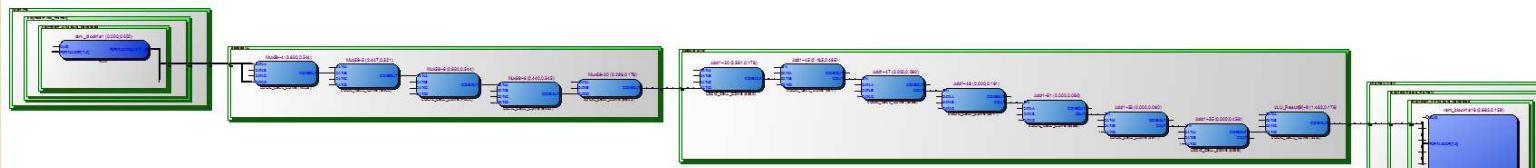
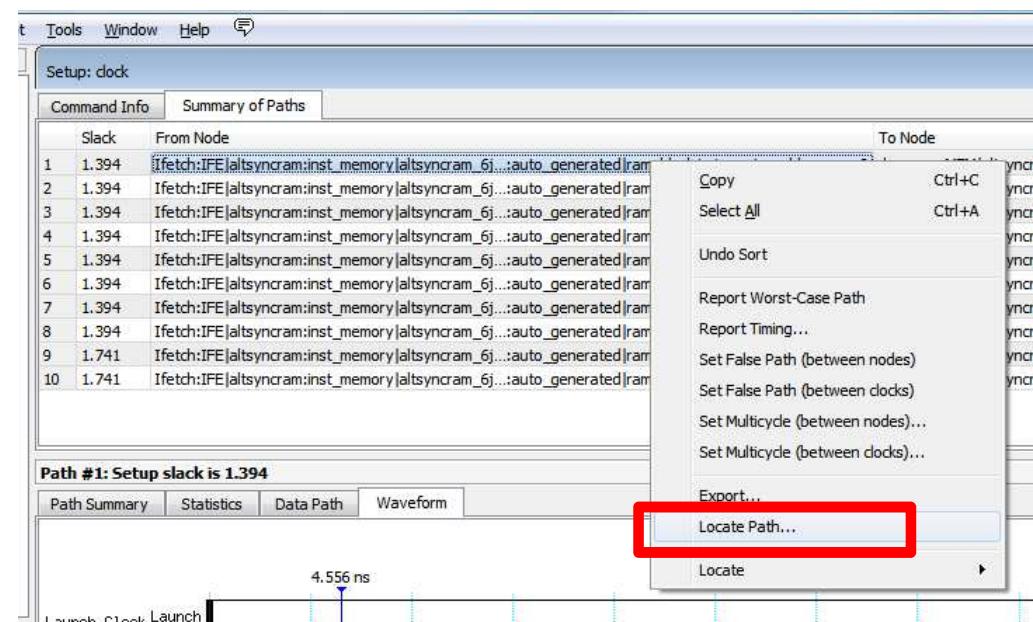
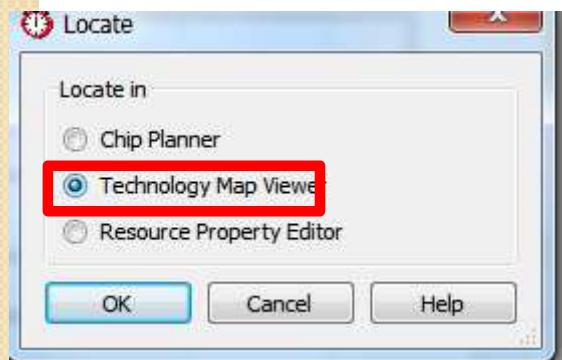
Context menu options:

- Copy
- Select All
- Ctrl+C
- Ctrl+A
- Undo Sort
- Create Setup Slack Histogram (In TimeQuest UI)
- Report Timing... (In TimeQuest UI) **highlighted with red box**

4/29/2013 kolaman@post.bgu.ac.il 93

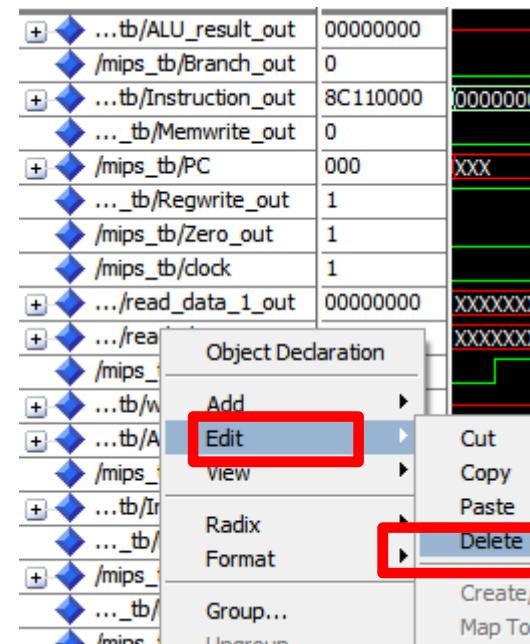
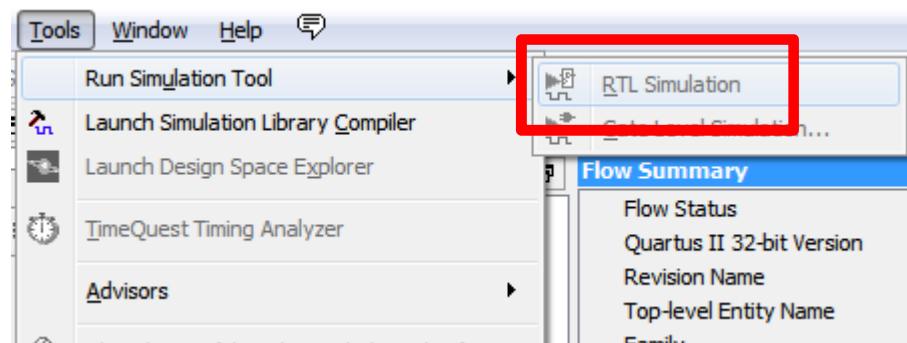
# Exercise 9

- Find the location of the longest data path?
- Where is it located?



# Exercise 9

- Run rtl simulation
- Mark all signals  
Ctrl+A and right click mouse and delete all the signals.



# Exercise 9

- load
- /lab3/ex9/quartus/simulation/modelsim/LWwave.do



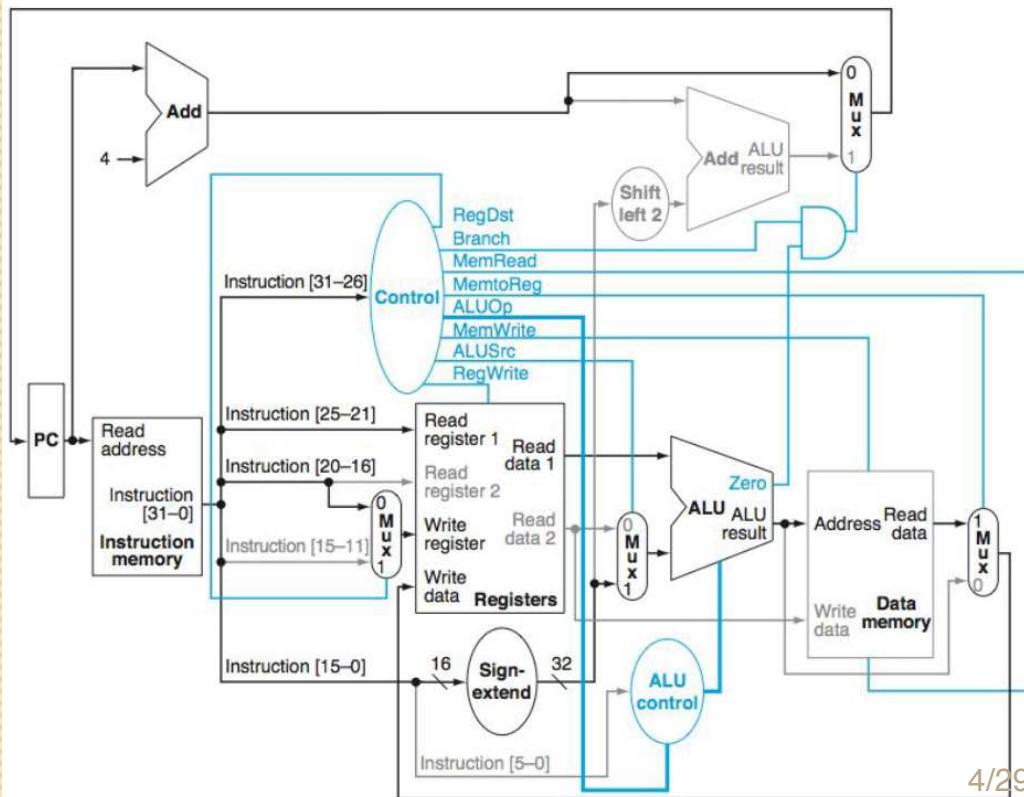
4/29/2013

עמיר קולמן kolaman@post.bgu.ac.il

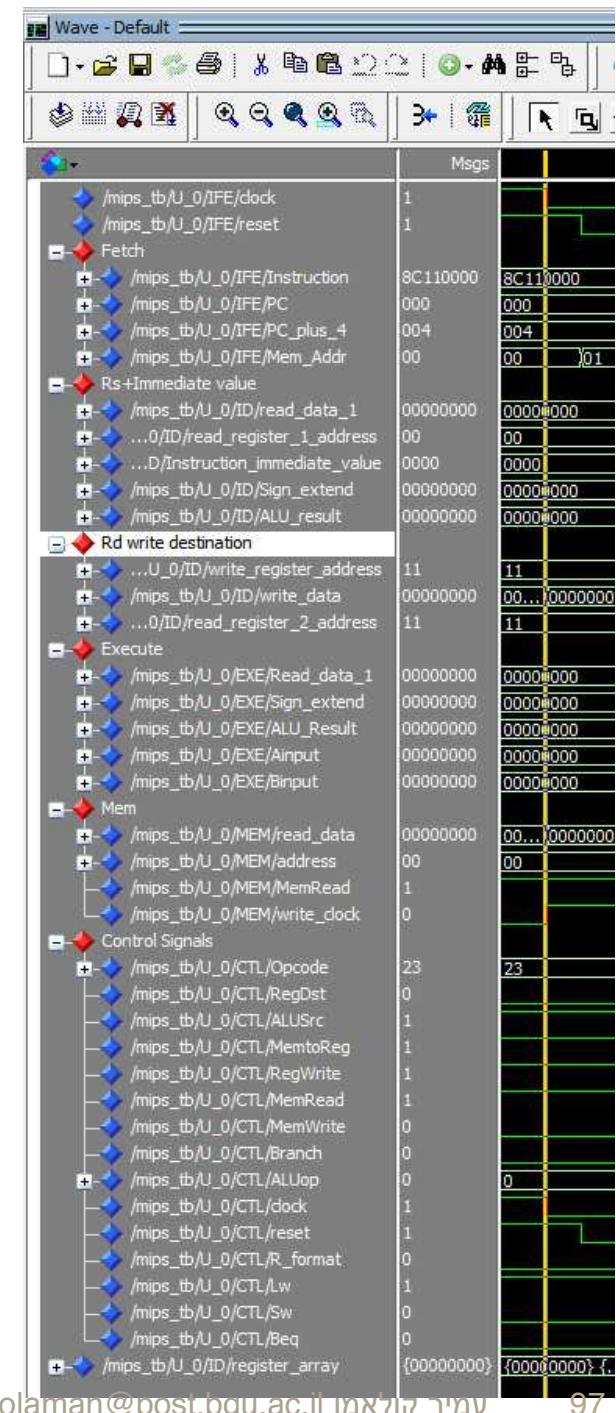
96

# Exercise 9

- Compare wave-forms to the MIPS diagram of load instruction



4/29/2013

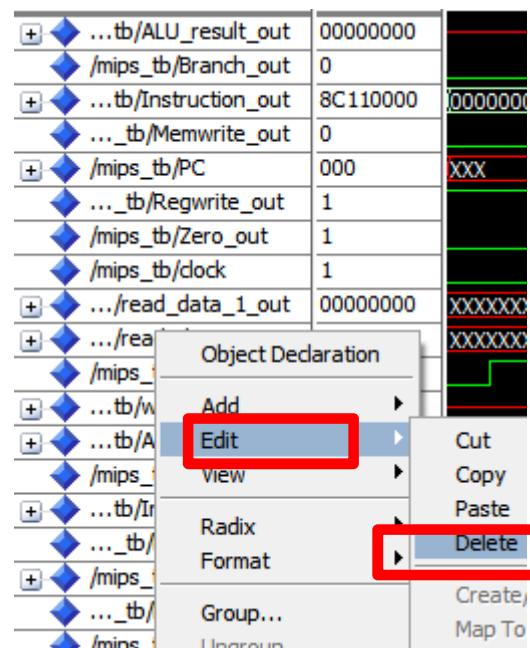


עמיר קולמן kolman@post.bgu.ac.il

97

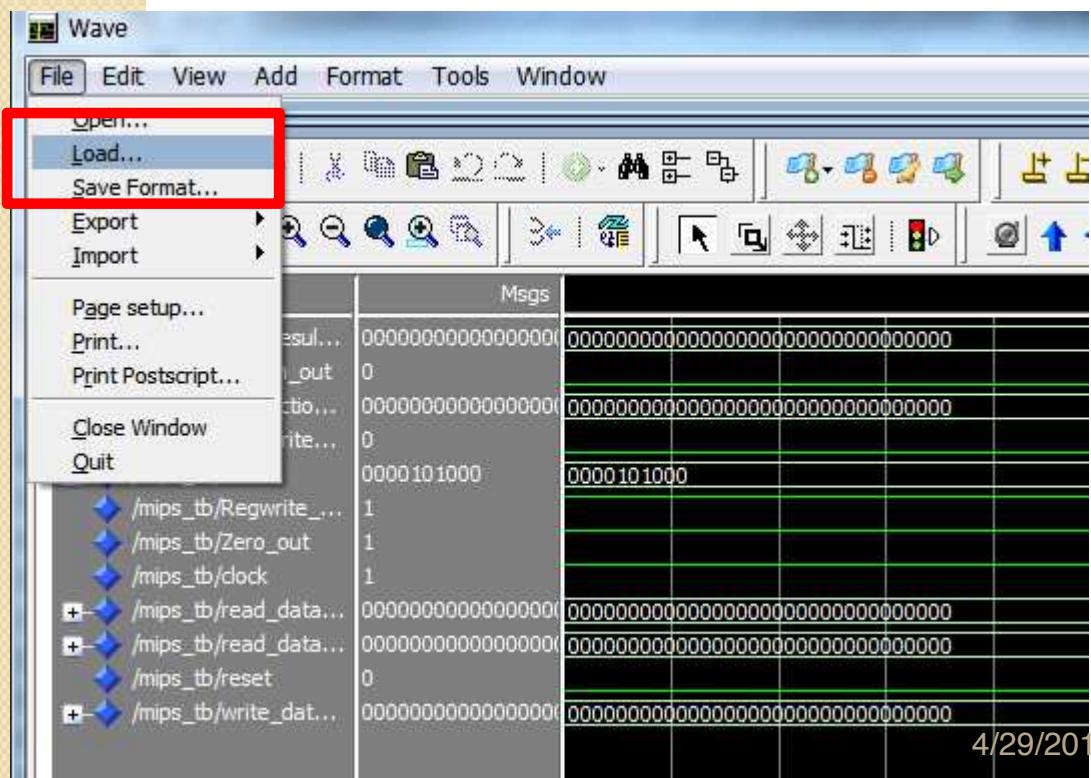
# Exercise 9

- Mark all signals  
Ctrl+A and right click mouse and delete all the signals.



# Exercise 9

- load
- /lab3/ex9/quartus/simulation/modelsim/ADDwave.d
  -



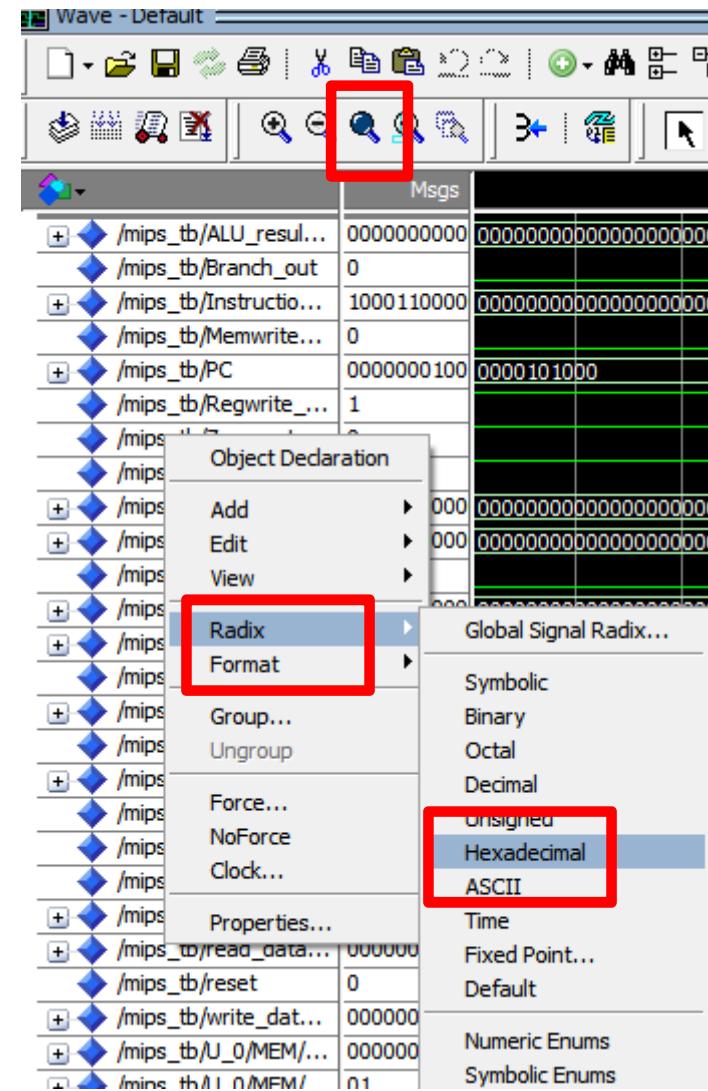
4/29/2013

עמיר קולמן kolaman@post.bgu.ac.il

99

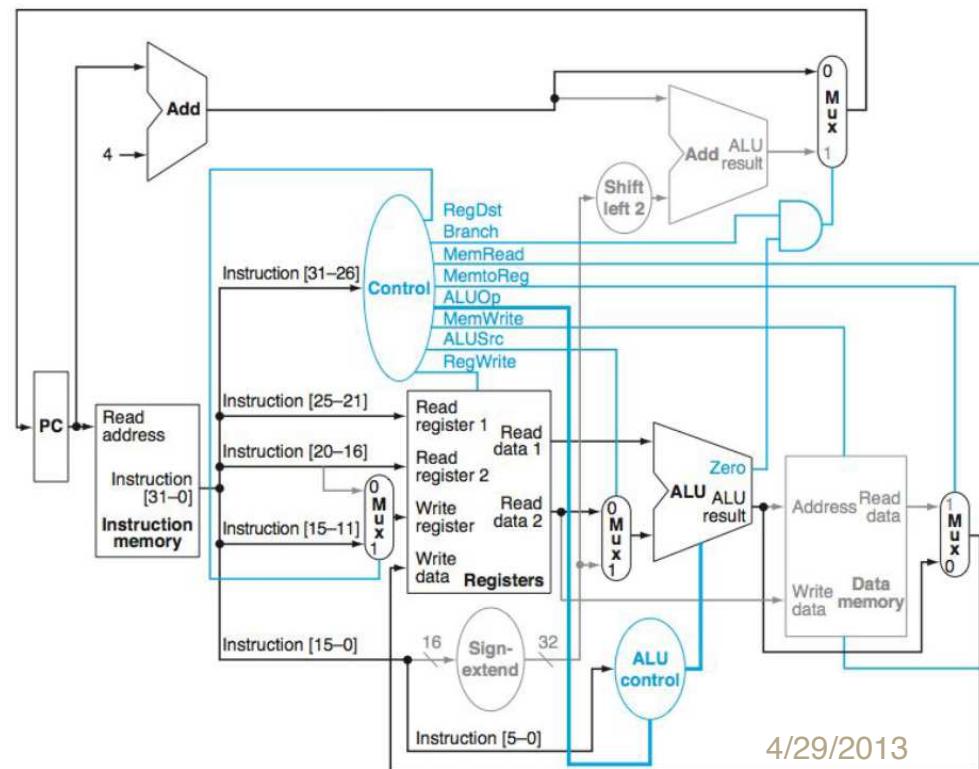
# Exercise 9

- Mark all signals  
Ctrl+A and right click mouse.
- Set waveform radix to hexadecimal.
- Zoom Full



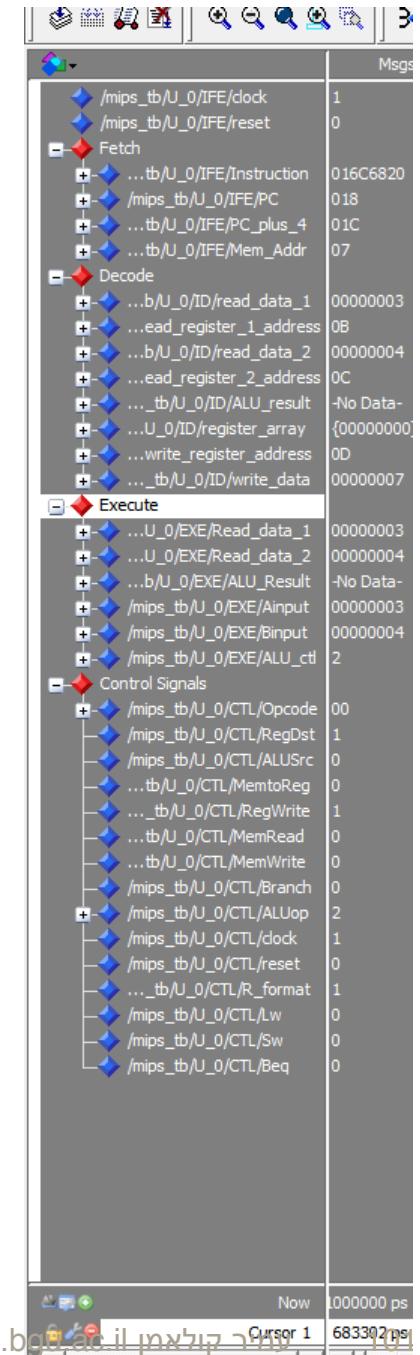
# Exercise 9

- Compare wave-forms to the MIPS diagram of load instruction



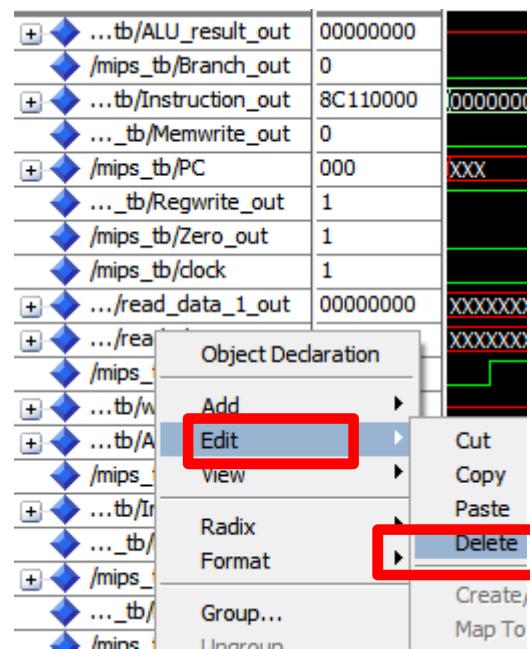
4/29/2013

kolaman@post.bgu.ac.il | מילר קולמן | 10



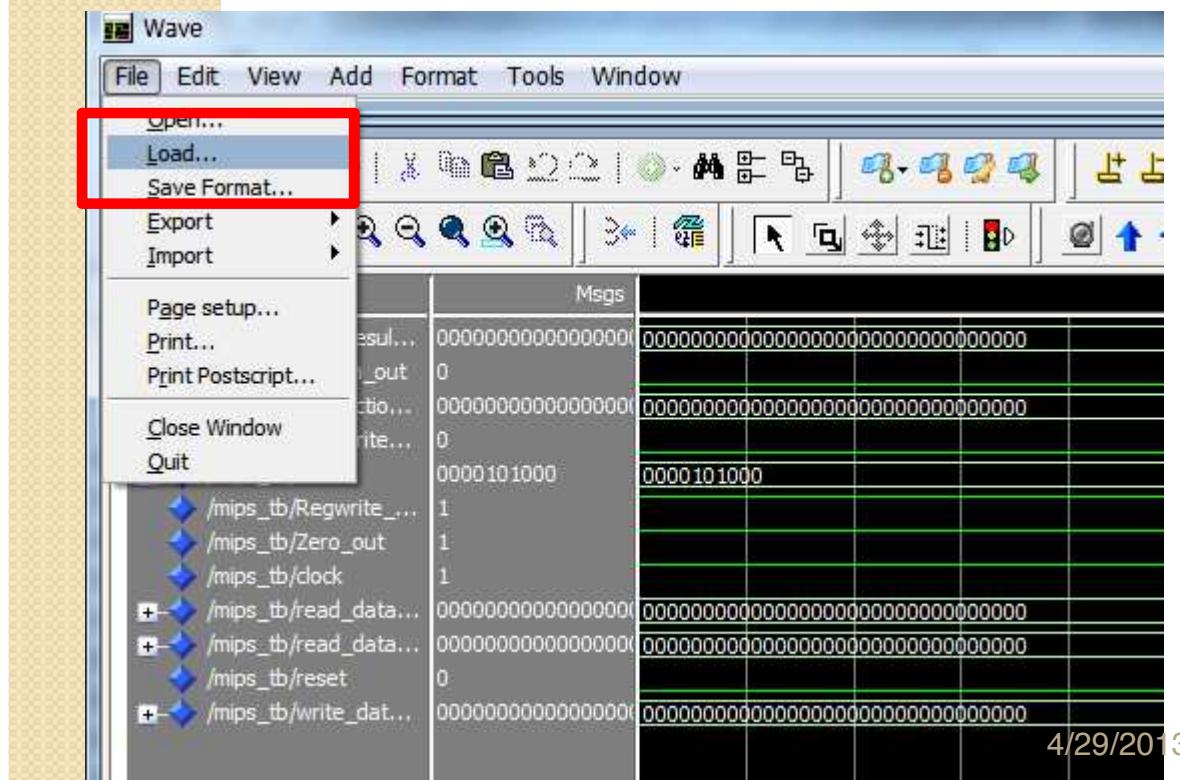
# Exercise 9

- Mark all signals  
Ctrl+A and right click mouse and delete all the signals.



# Exercise 9

- load
- /lab3/ex9/quartus/simulation/modelsim/BEQwave.do



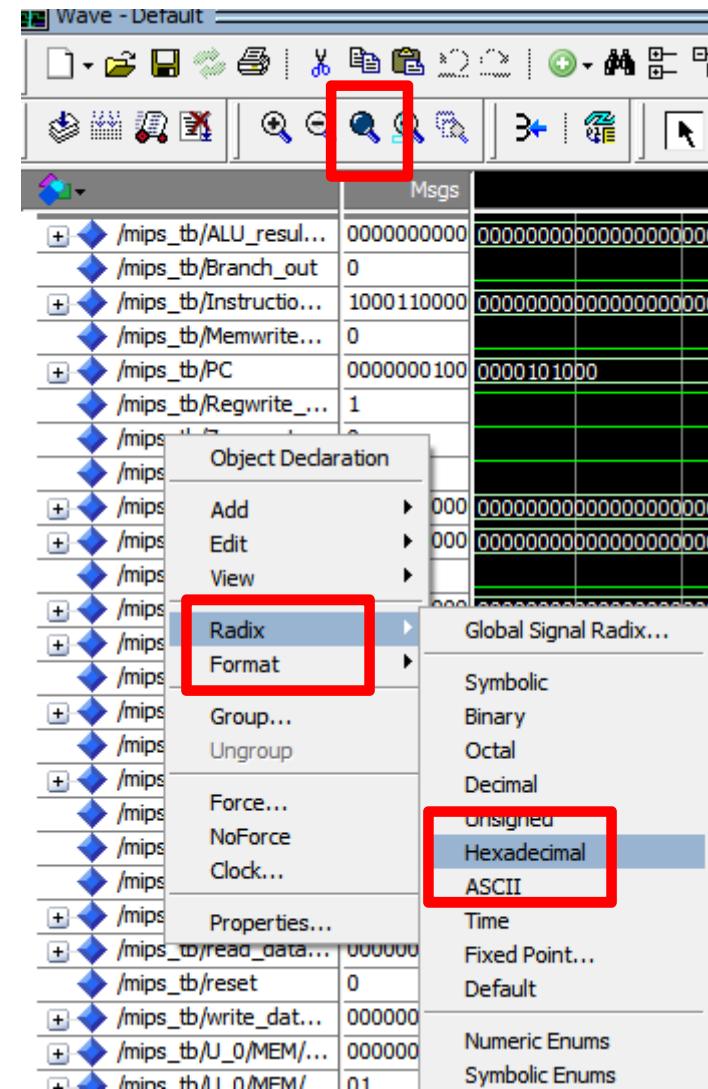
4/29/2013

עמיר קולמן [kolaman@post.bgu.ac.il](mailto:kolaman@post.bgu.ac.il)

103

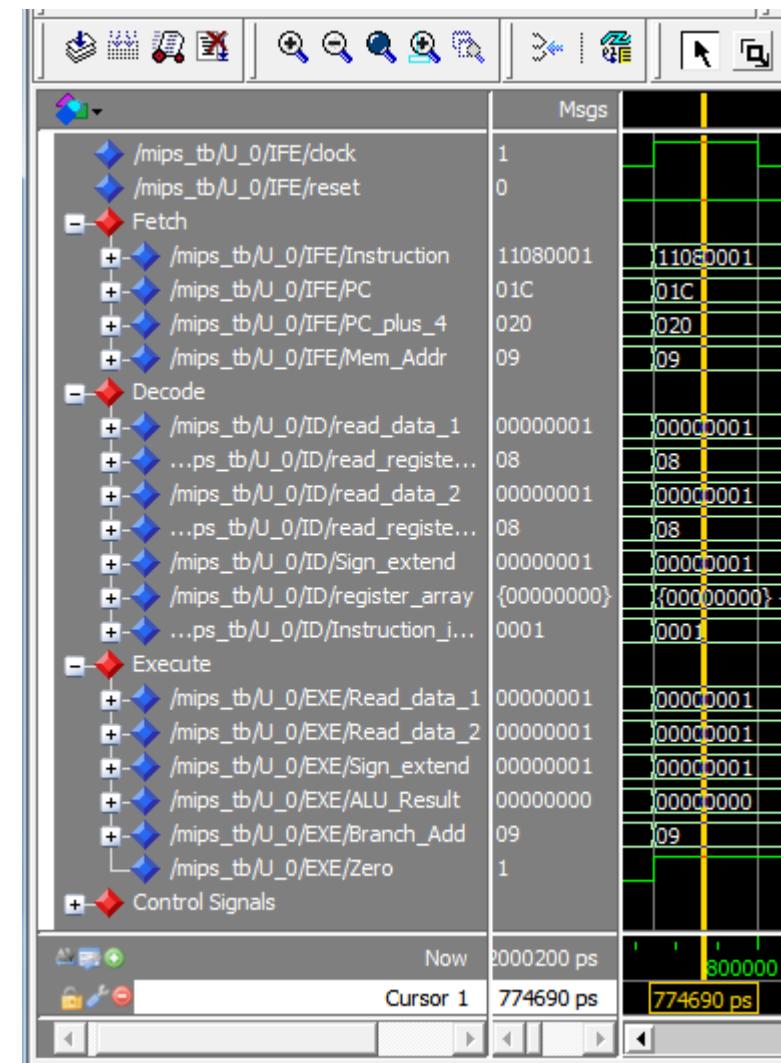
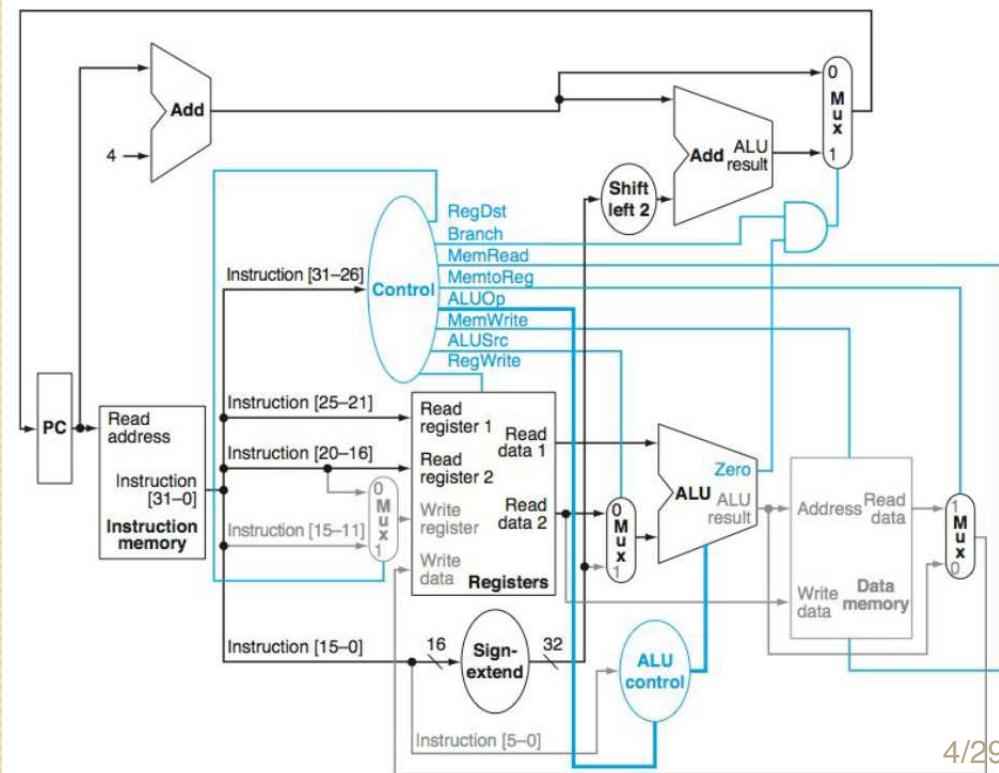
# Exercise 9

- Mark all signals  
Ctrl+A and right click mouse.
- Set waveform radix to hexadecimal.
- Zoom Full



# Exercise 9

- Compare wave-forms to the MIPS diagram of BEQ instruction



4/29/2013

עמיר קולמן post.bgu.ac.il

105