Keil uVision
Debugging Programs

WEL Labs, IITB

2016

Development Tools (Revision)

- **Coding – Editor => Entry of code into file(s)**
- **Translation – Assembler or Compiler**

  => Generate machine code from source code
- **Execution check – using Debugger to verify operation of program (on Simulator)**
- **Program – Programmer**

  => Put machine code in the chip

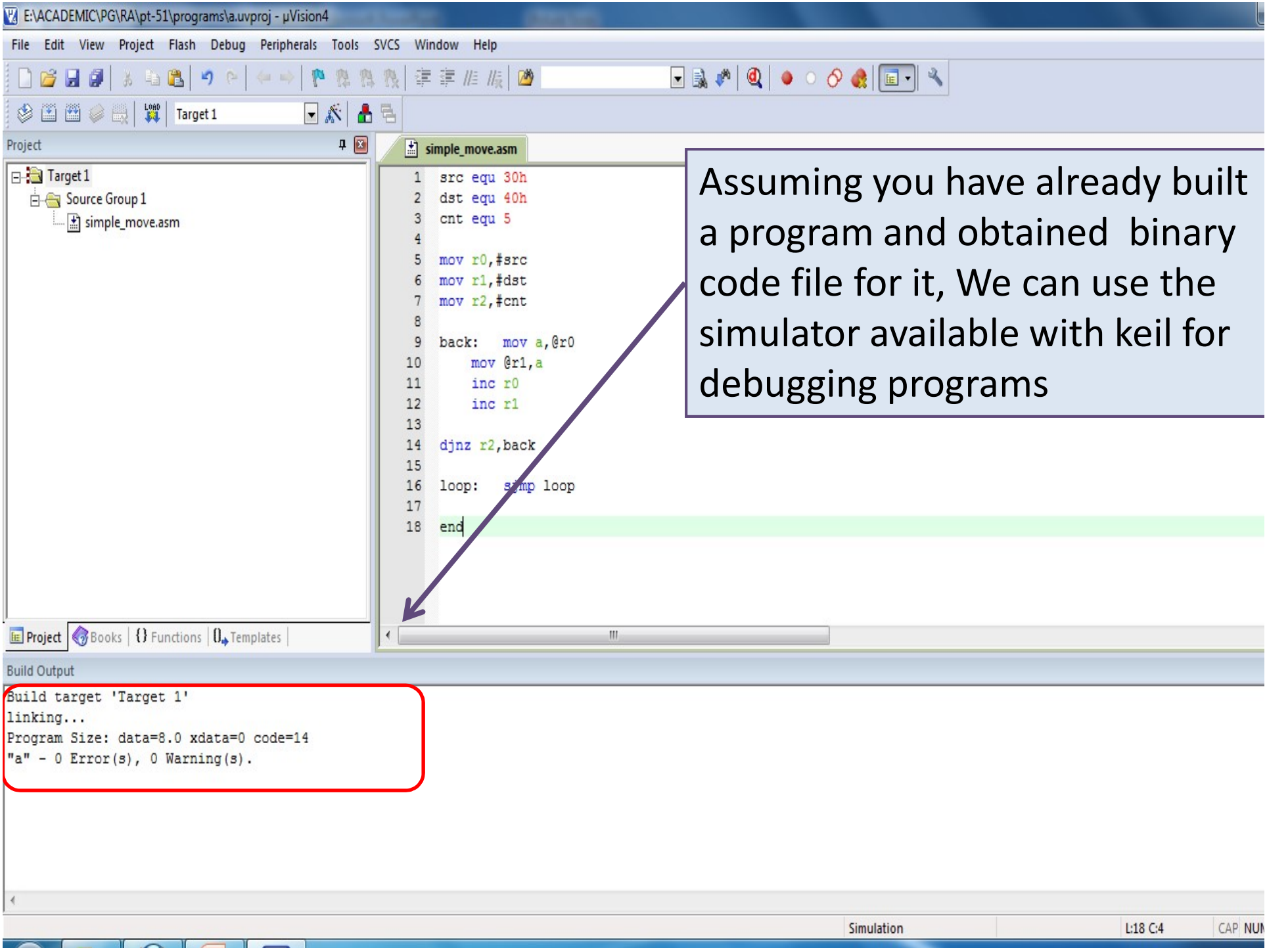
Single Point Solution – IDE e.g. Keil

Keil uVision IDE …

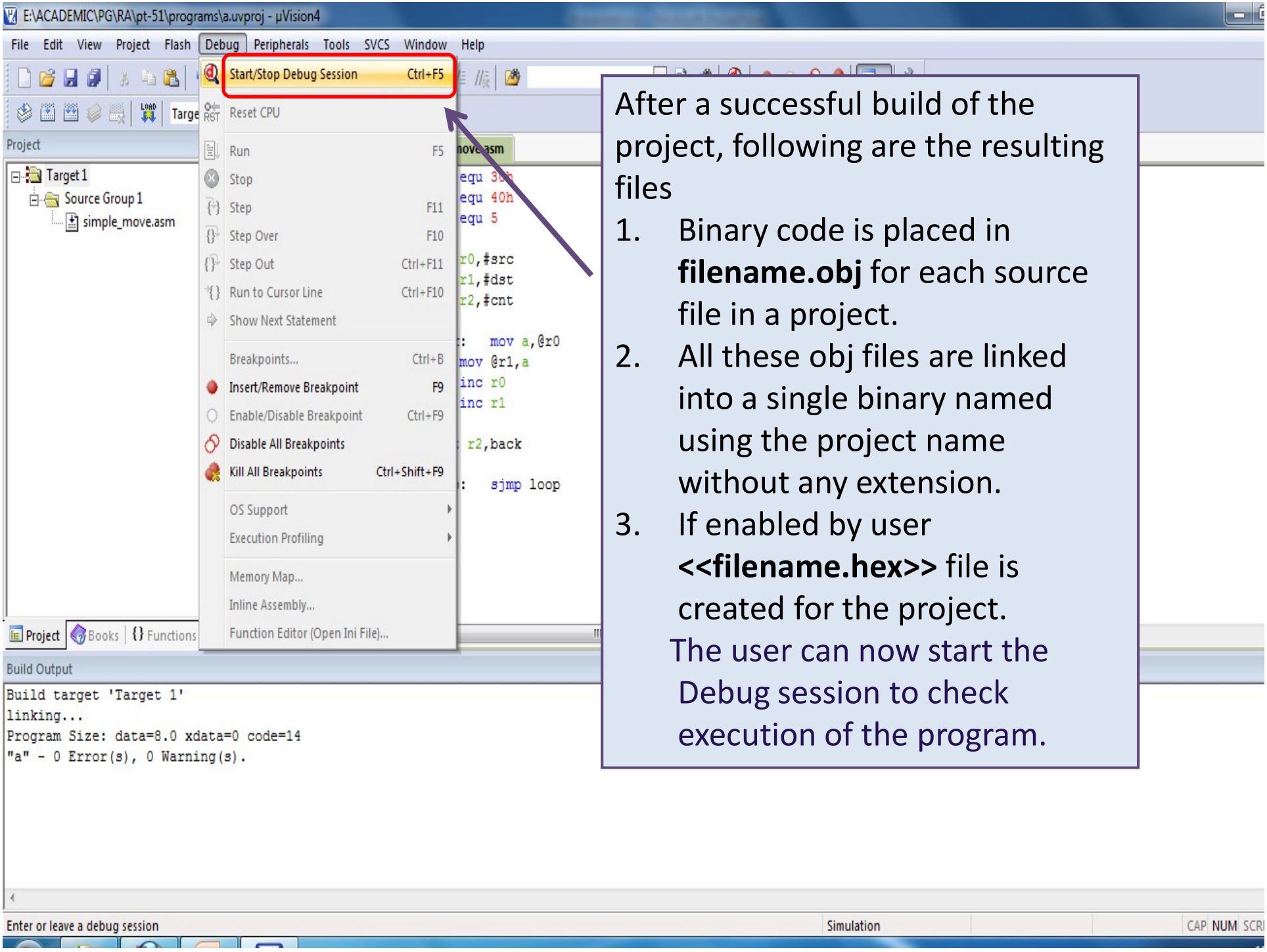
- **Project :** A **collection of files related to a particular programming task**.
- **Build :** The process in which **only the files modified since last build** are assembled/compiled for the chosen microcontroller device.
- **Rebuild :** The process in which **all files** **are assembled/compiled** irrespective of their modification state.
- **Debug : The process of finding errors happening during program execution and removing them.**

Simulator and Debugger

- **Simulation of microcontroller behavior while executing the user program**
- **The Debug mode UI allows the user to perform the following**
  1. **Observe microprocessor Registers, Memory, Ports and Peripherals**
  2. **Place breakpoints to stop simulation as specific instruction or on condition**
  3. **Monitor code under execution**
  4. **Modify data variables**
  5. **Monitor timing of execution**



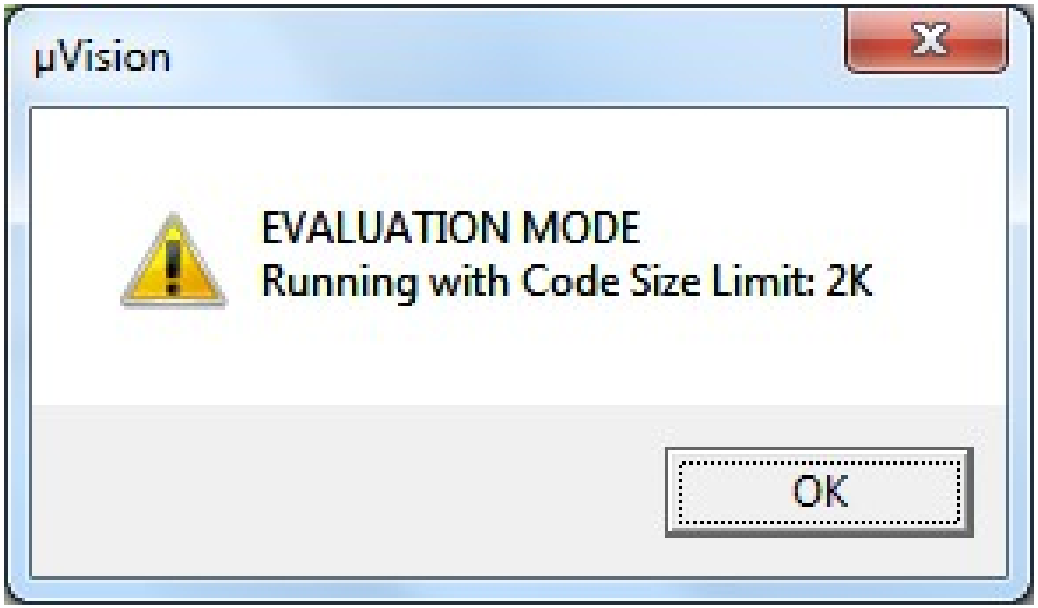
Assuming you have already built a program and obtained binary code file for it, We can use the simulator available with keil for debugging programs



After a successful build of the project, following are the resulting files

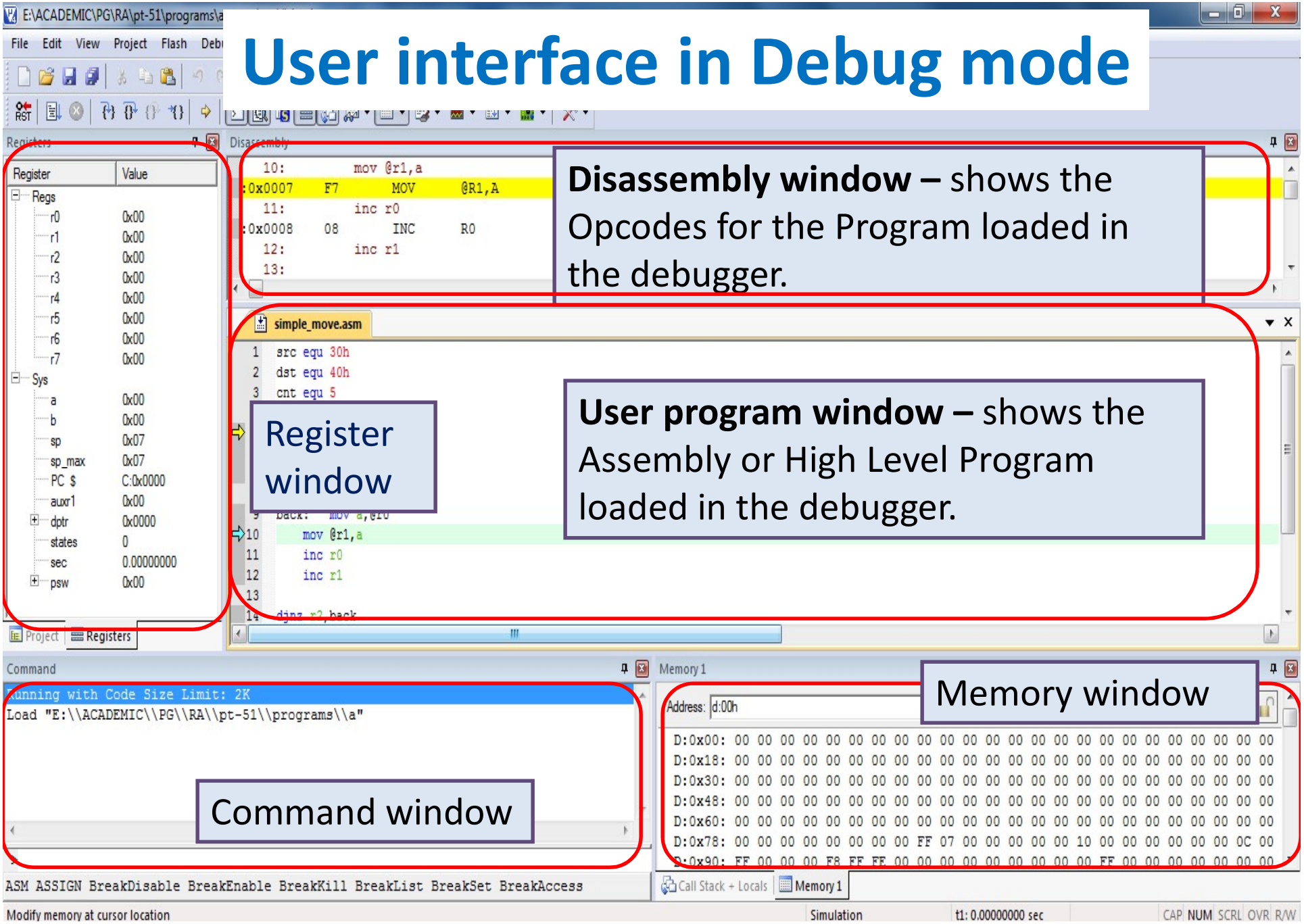
1. Binary code is placed in **filename.obj** for each source file in a project.
2. All these obj files are linked into a single binary named using the project name without any extension.
3. If enabled by user **<<filename.hex>>** file is created for the project.

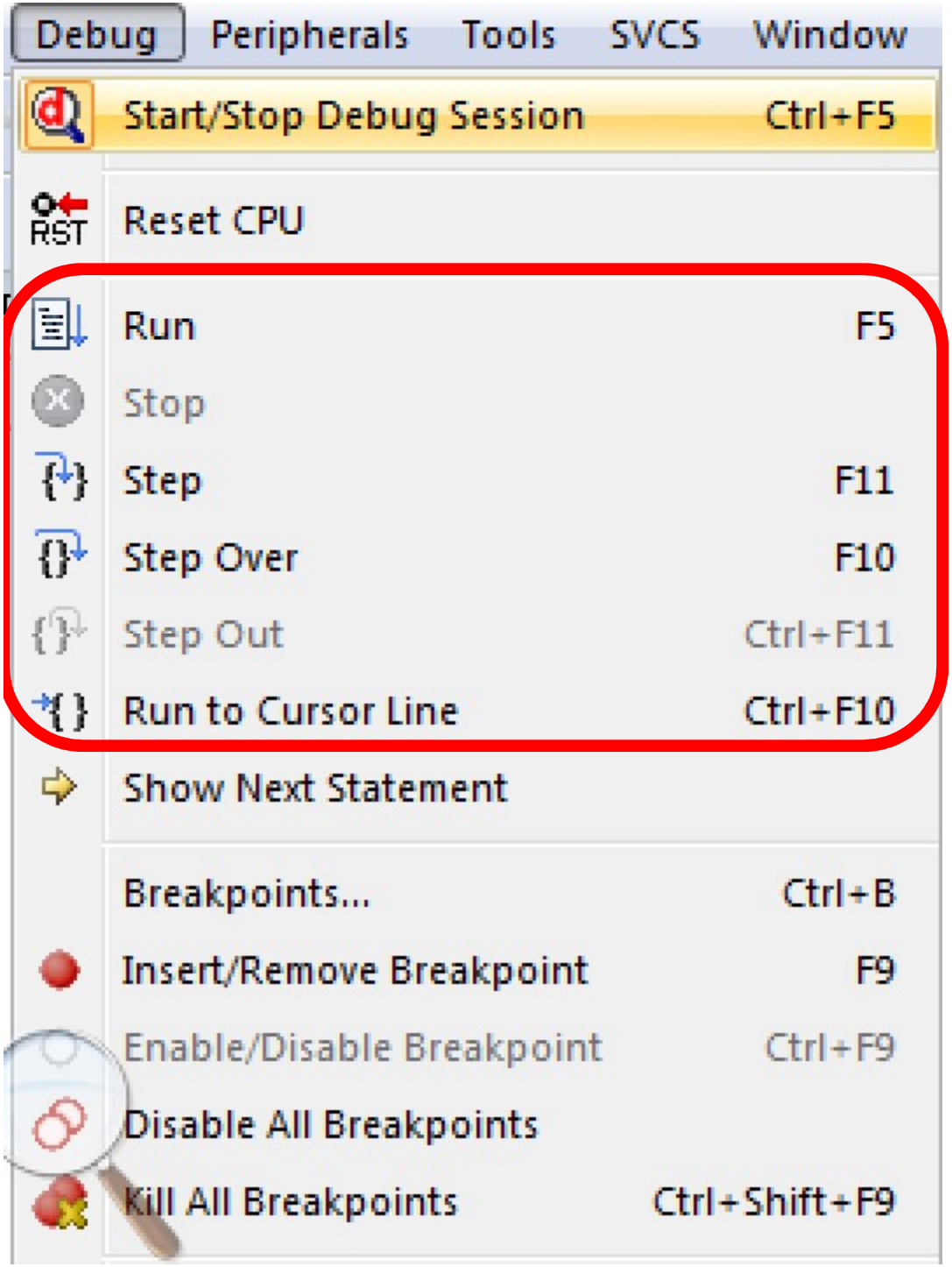
The user can now start the Debug session to check execution of the program.



A popup of Evaluation version is presented which shows the limitation of the mode.
On clicking “OK” the popup closes and the user is presented with the **Debug Mode** user interface (UI)

User interface in Debug mode





User can execute instructions in multiple modes :

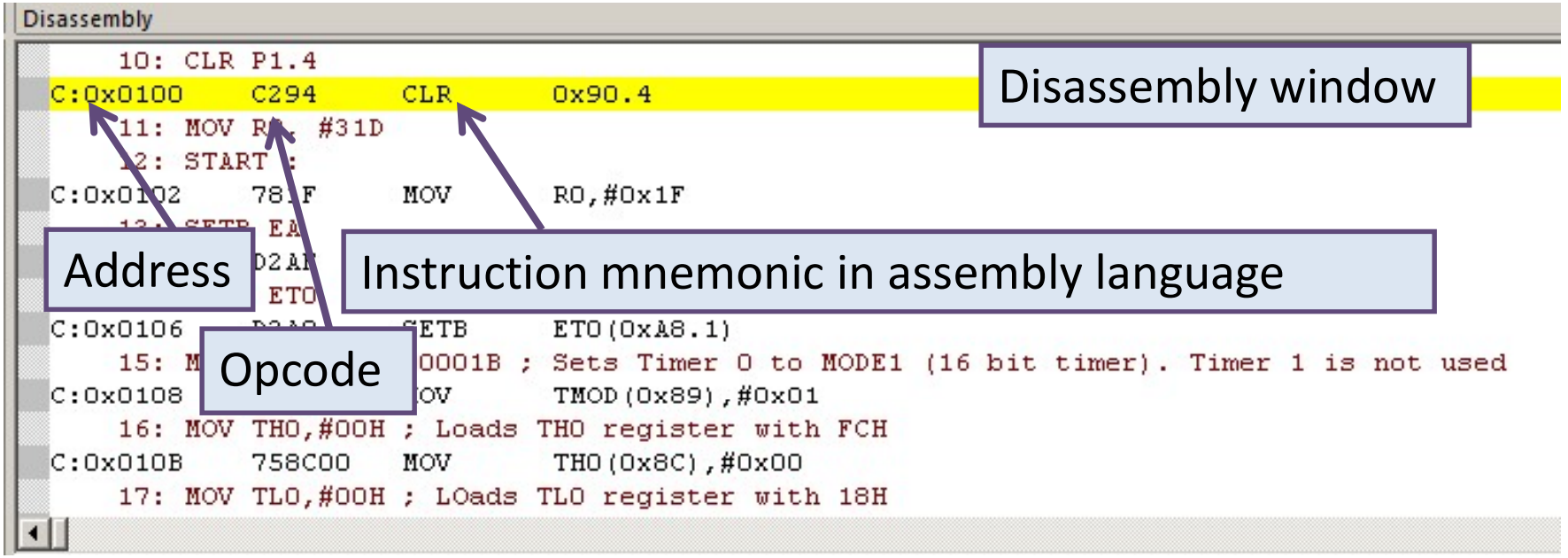
1. **Run (F5)** – Continues executing the program until the next active breakpoint is reached or till the program termination.

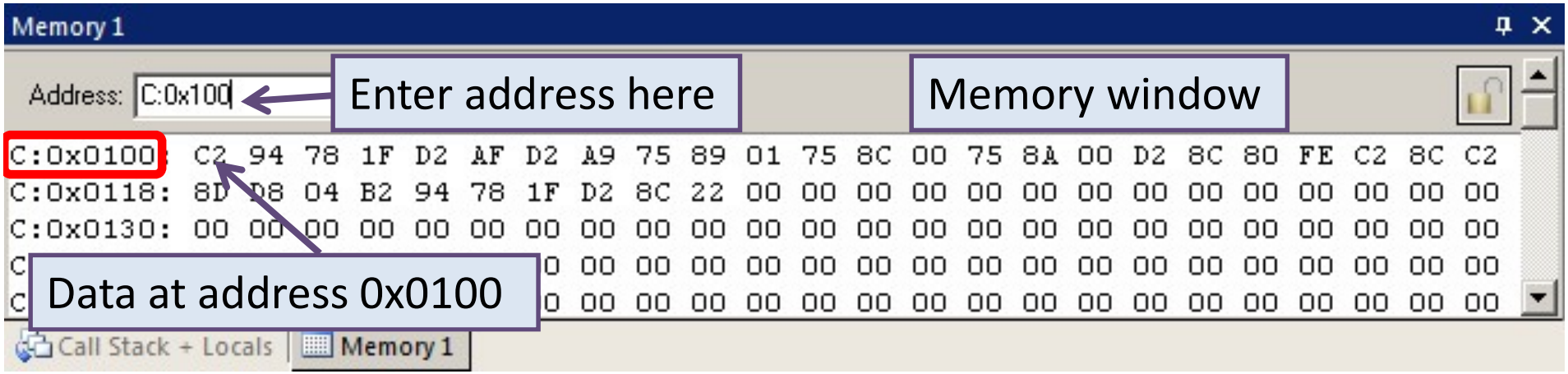
2. **Step (F11)--** Executes a single-step into a function; Executes the current instruction line.

3. **Step Over (F10) –** Executes a single-step over a function.

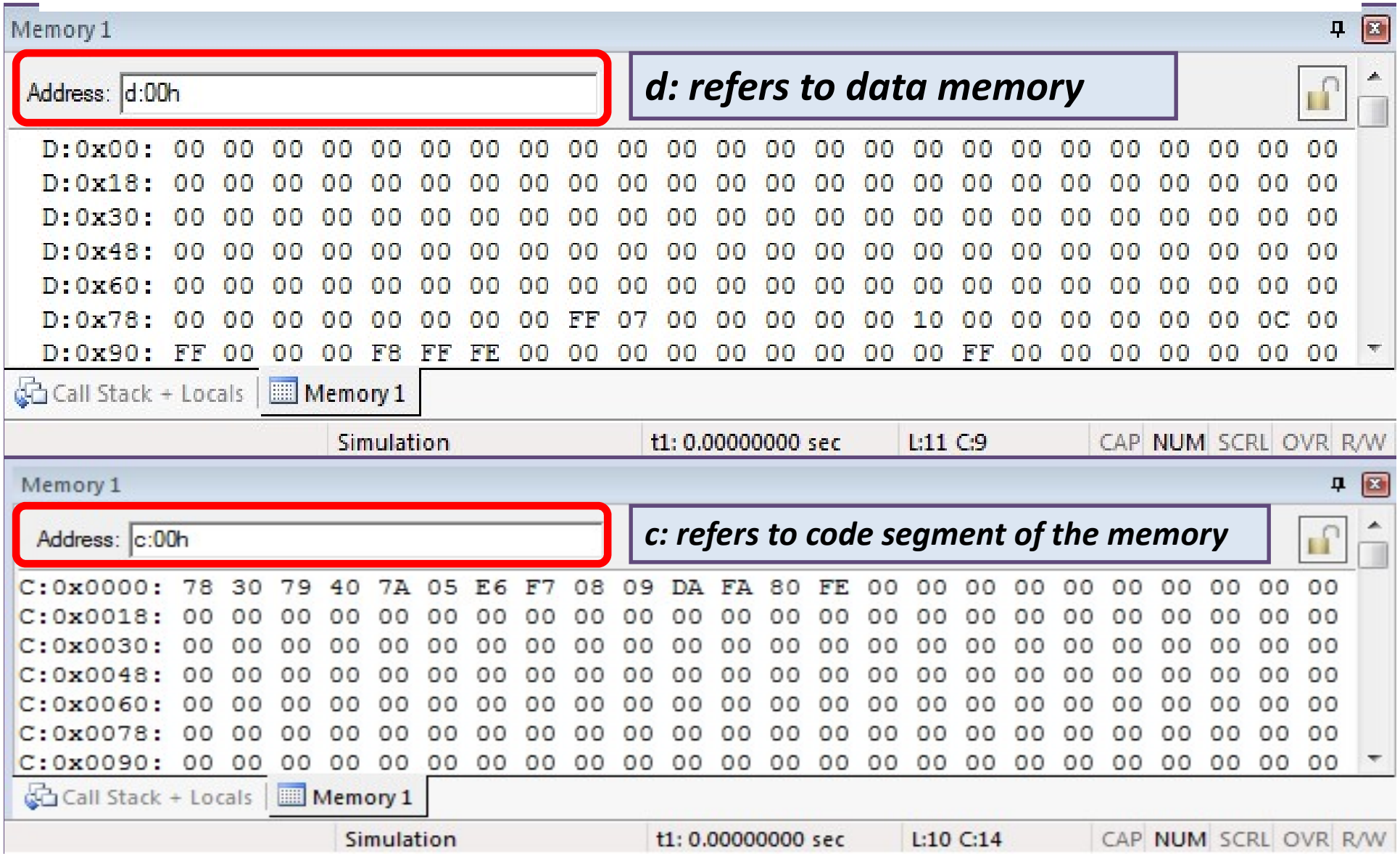
4. Run to Cursor Line **(Ctrl+F10)** Allows user to place a cursor and run the program till that line.

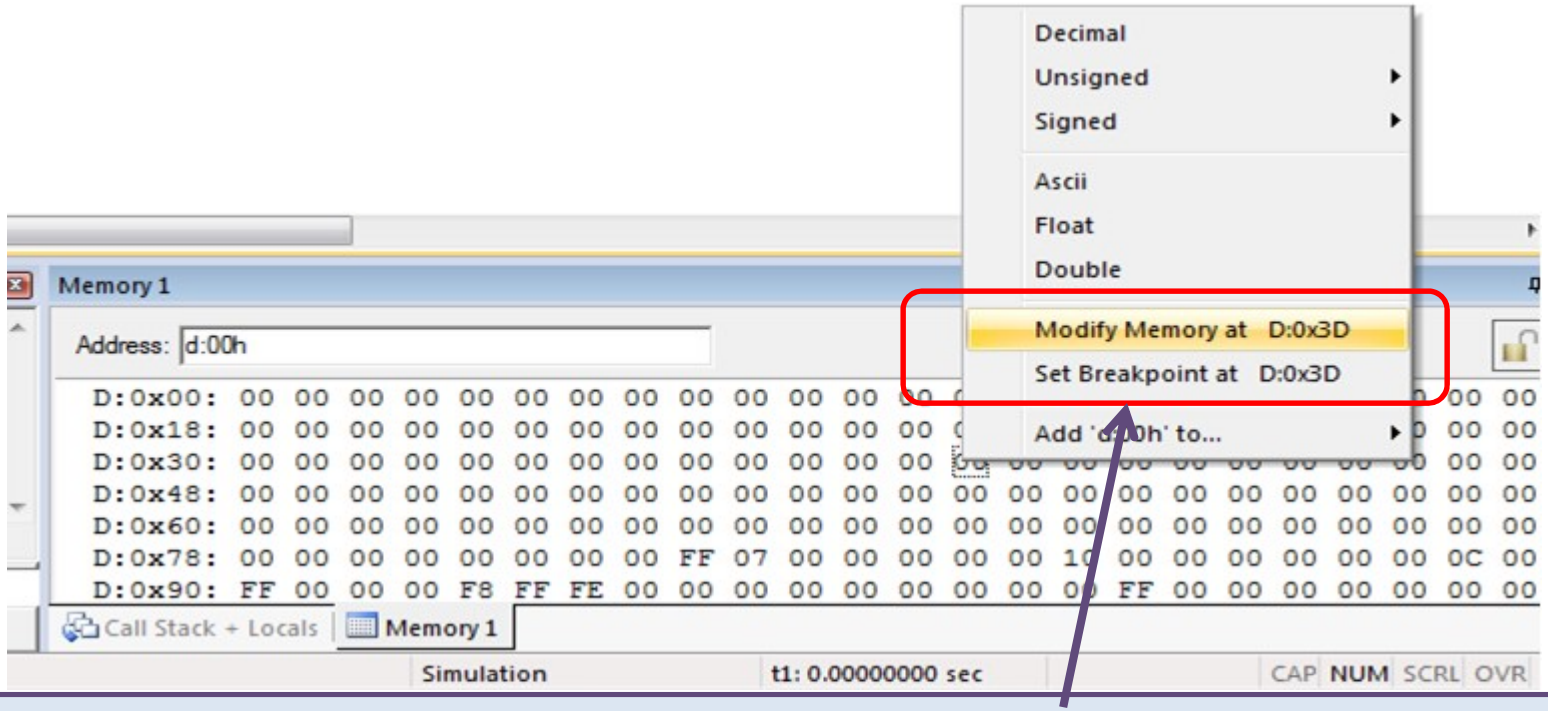
Details of Disassembly and Memory window





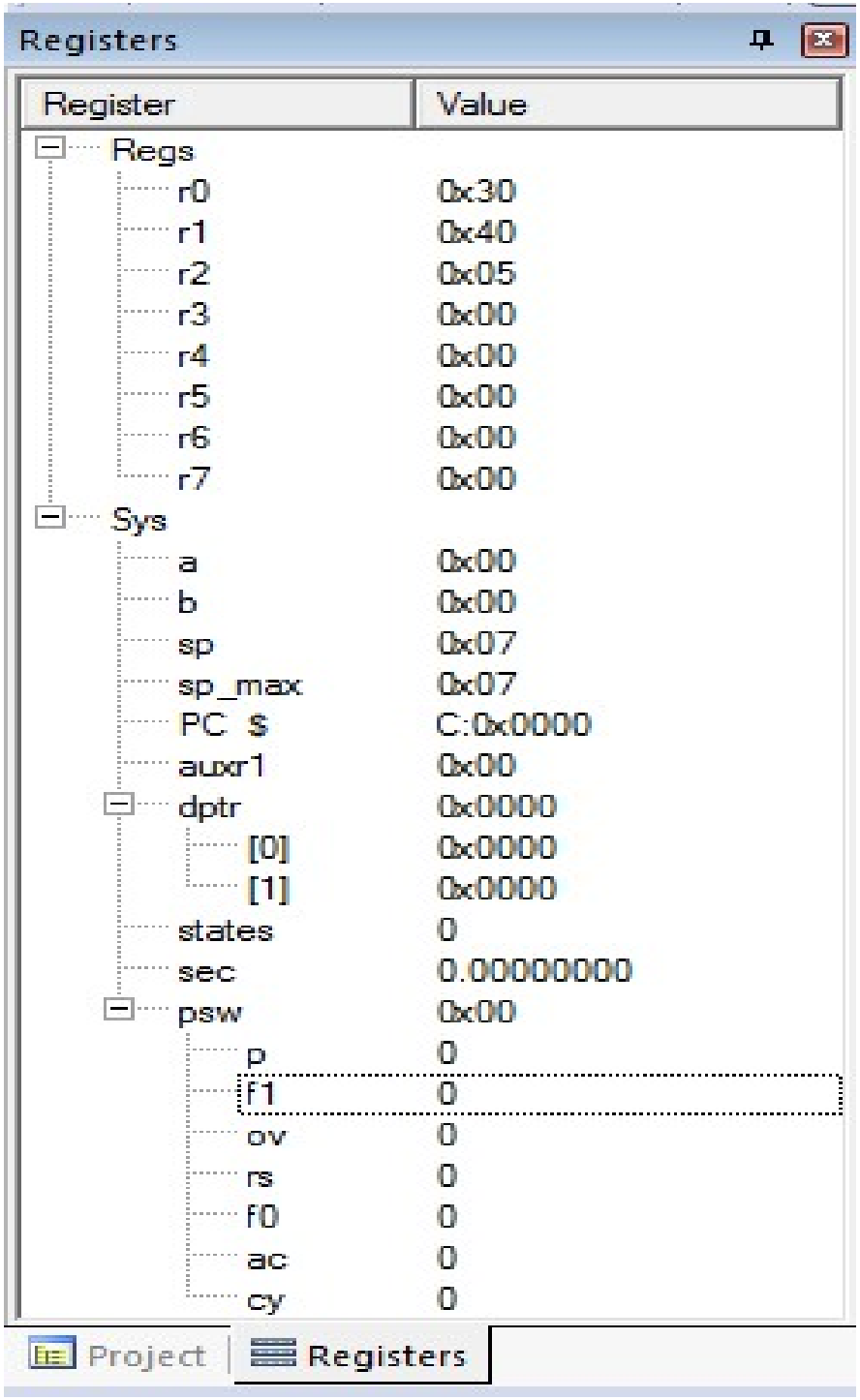
Code and Data memory access



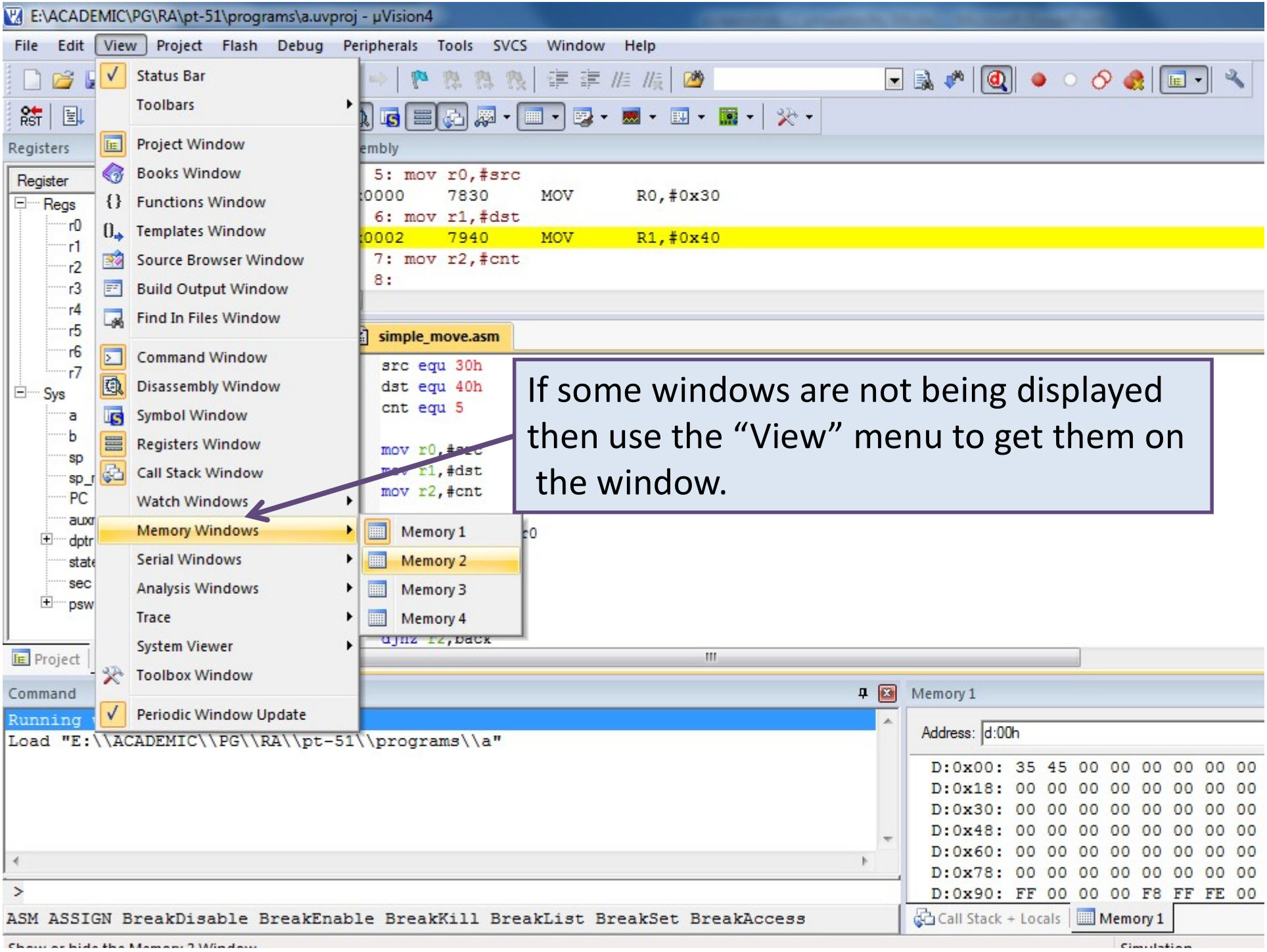


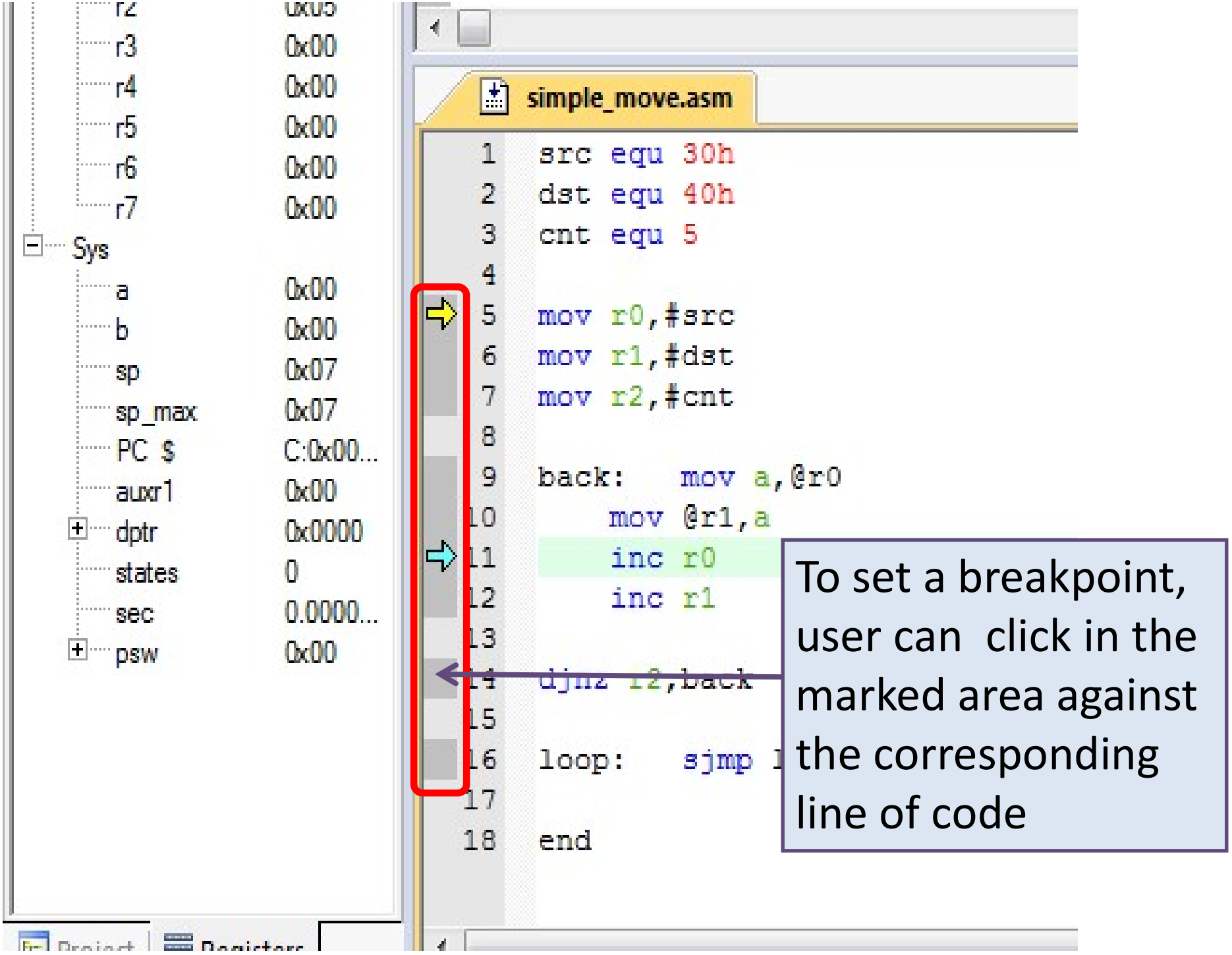
During execution, user can right click on the required memory location in the memory window to modify RAM data. Functionality for selecting the number system in which the memory contents are to be displayed is also available .

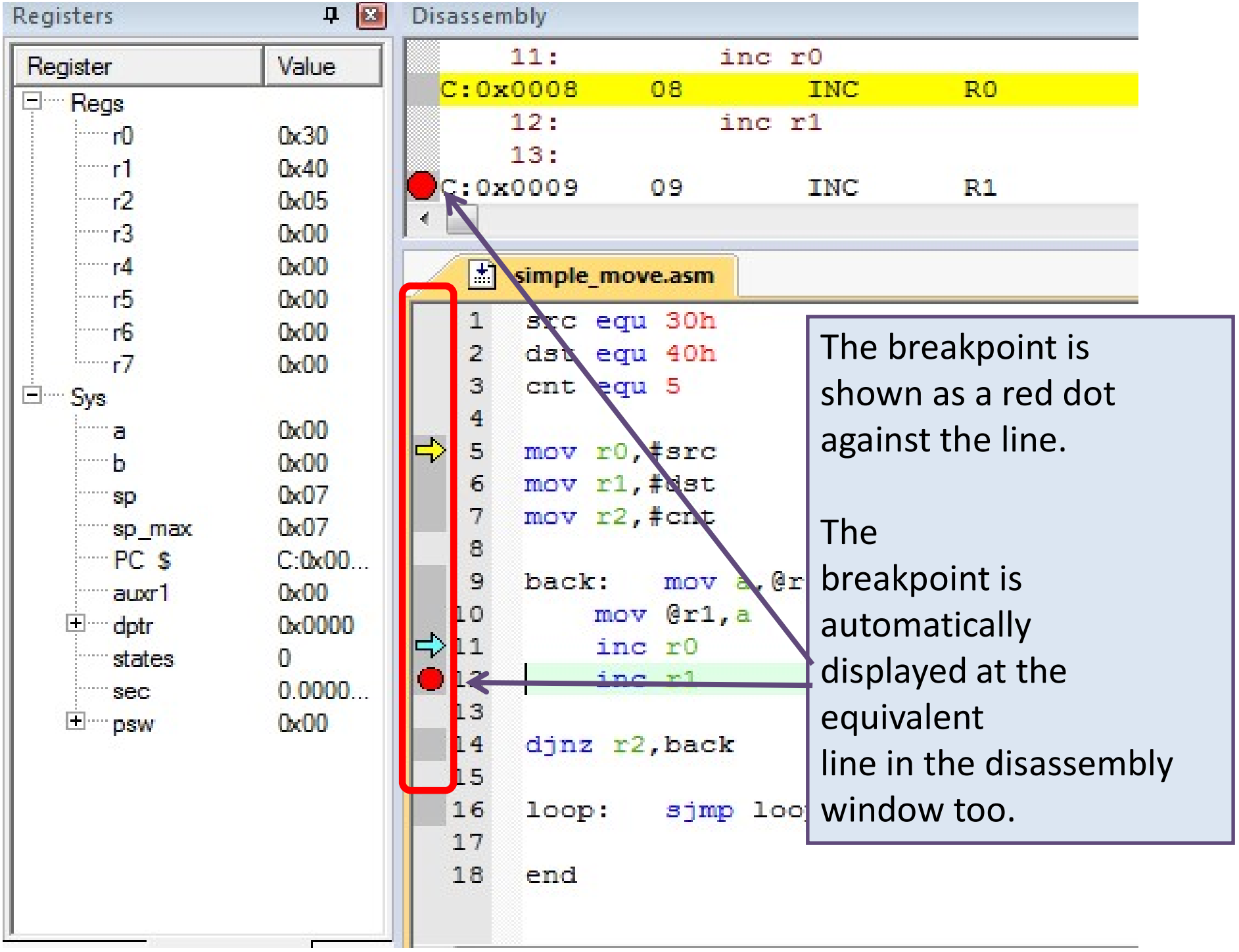
Note: To initialize memory contents on hardware, user has to add necessary instructions in the program code.



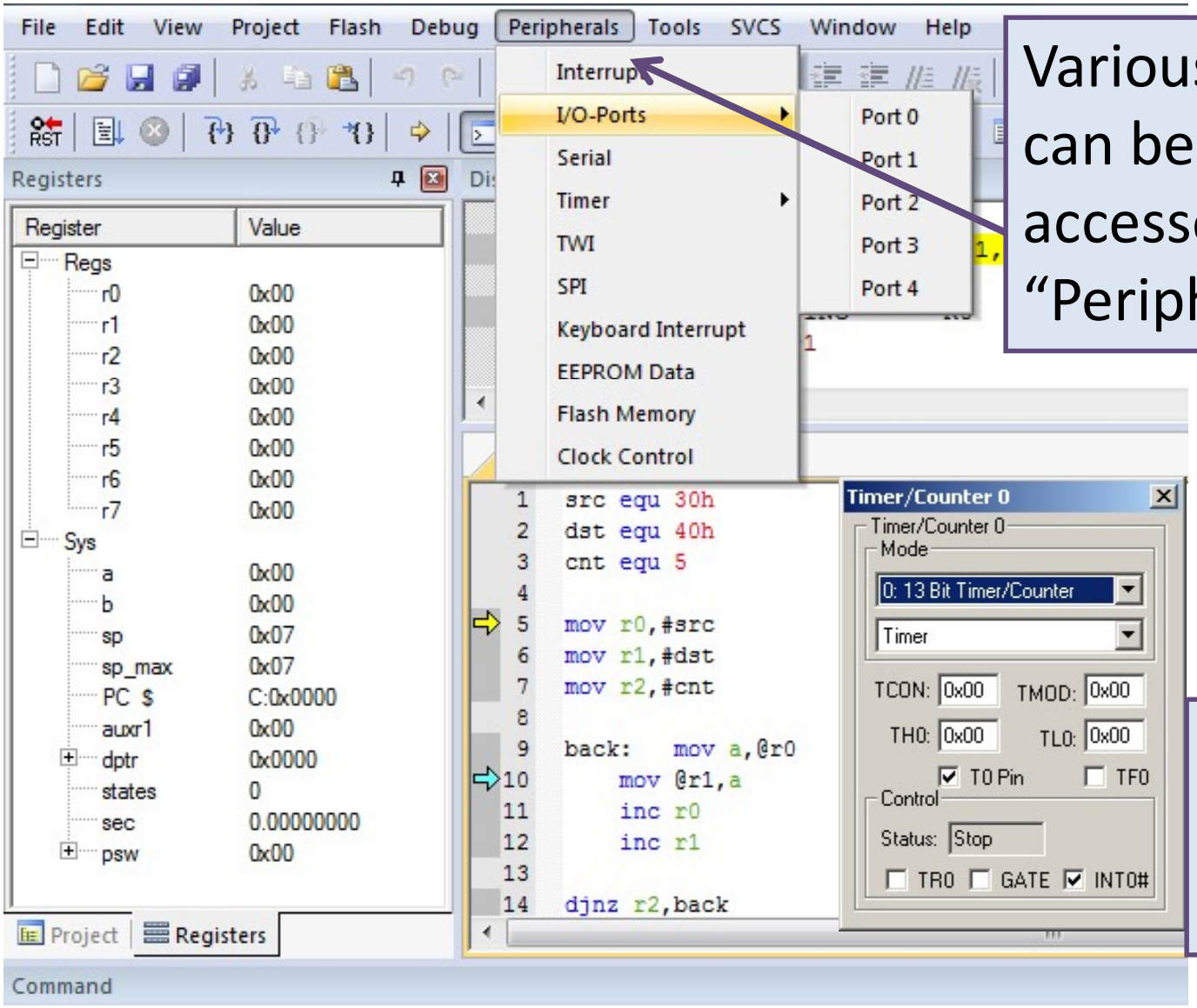
The Registers window provides access to all the registers including the flag register , DPTRs etc.


E:\ACADEMIC\PG\RA\pt-51\programs\a.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Status Bar
Toolbars
Project Window
Books Window
Functions Window
Templates Window
Source Browser Window
Build Output Window
Find In Files Window
Command Window
Disassembly Window
Symbol Window
Registers Window
Call Stack Window
Watch Windows
Memory Windows
Serial Windows
Analysis Windows
Trace
System Viewer
Toolbox Window
Periodic Window Update
Memory 1
Memory 2
Memory 3
Memory 4
5: mov r0,#src
6: mov r1,#dst
7: mov r2,#cnt
8:
MOV R0,#0x30
MOV R1,#0x40
simple_move.asm
src equ 30h
dst equ 40h
cnt equ 5
mov r1,#dst
mov r2,#cnt
If some windows are not being displayed then use the “View” menu to get them on the window.
Load "E:\\ACADEMIC\\PG\\RA\\pt-51\\programs\\a"
Address: d:00h
D:0x00: 35 45 00 00 00 00 00 00
D:0x18: 00 00 00 00 00 00 00 00
D:0x30: 00 00 00 00 00 00 00 00
D:0x48: 00 00 00 00 00 00 00 00
D:0x60: 00 00 00 00 00 00 00 00
D:0x78: 00 00 00 00 00 00 00 00
D:0x90: FF 00 00 00 F8 FF FE 00
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Memory 1


r2 0x05
r3 0x00
r4 0x00
r5 0x00
r6 0x00
r7 0x00
Sys
a 0x00
b 0x00
sp 0x07
sp_max 0x07
PC $ C:0x00...
auxr1 0x00
dptr 0x0000
states 0
sec 0.0000...
psw 0x00
Project
Registers
simple_move.asm
1 src equ 30h
2 dst equ 40h
3 cnt equ 5
4
5 mov r0,#src
6 mov r1,#dst
7 mov r2,#cnt
8
9 back: mov a,@r0
10 mov @r1,a
11 inc r0
12 inc r1
13
14 djnz r2,back
15
16 loop: sjmp l
17
18 end
To set a breakpoint, user can click in the marked area against the corresponding line of code


Registers
Register
Value
Regs
r0 0x30
r1 0x40
r2 0x05
r3 0x00
r4 0x00
r5 0x00
r6 0x00
r7 0x00
Sys
a 0x00
b 0x00
sp 0x07
sp_max 0x07
PC $ C:0x00...
auxr1 0x00
dptr 0x0000
states 0
sec 0.0000...
psw 0x00
Disassembly
11: inc r0
C:0x0008 08 INC R0
12: inc r1
13:
C:0x0009 09 INC R1
simple_move.asm
1 src equ 30h
2 dst equ 40h
3 cnt equ 5
4
5 mov r0,#src
6 mov r1,#dst
7 mov r2,#cnt
8
9 back: mov a,@r
10 mov @r1,a
11 inc r0
12 inc r1
13
14 djnz r2,back
15
16 loop: sjmp loo
17
18 end
The breakpoint is shown as a red dot against the line.
The breakpoint is automatically displayed at the equivalent line in the disassembly window too.

Peripherals menu

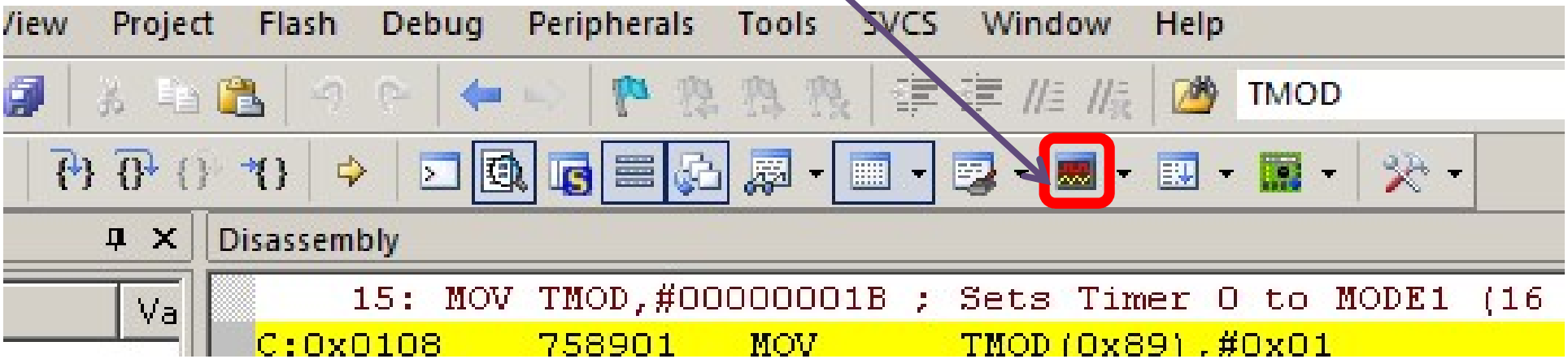


Various Peripherals can be accessed through the “Peripherals” menu.

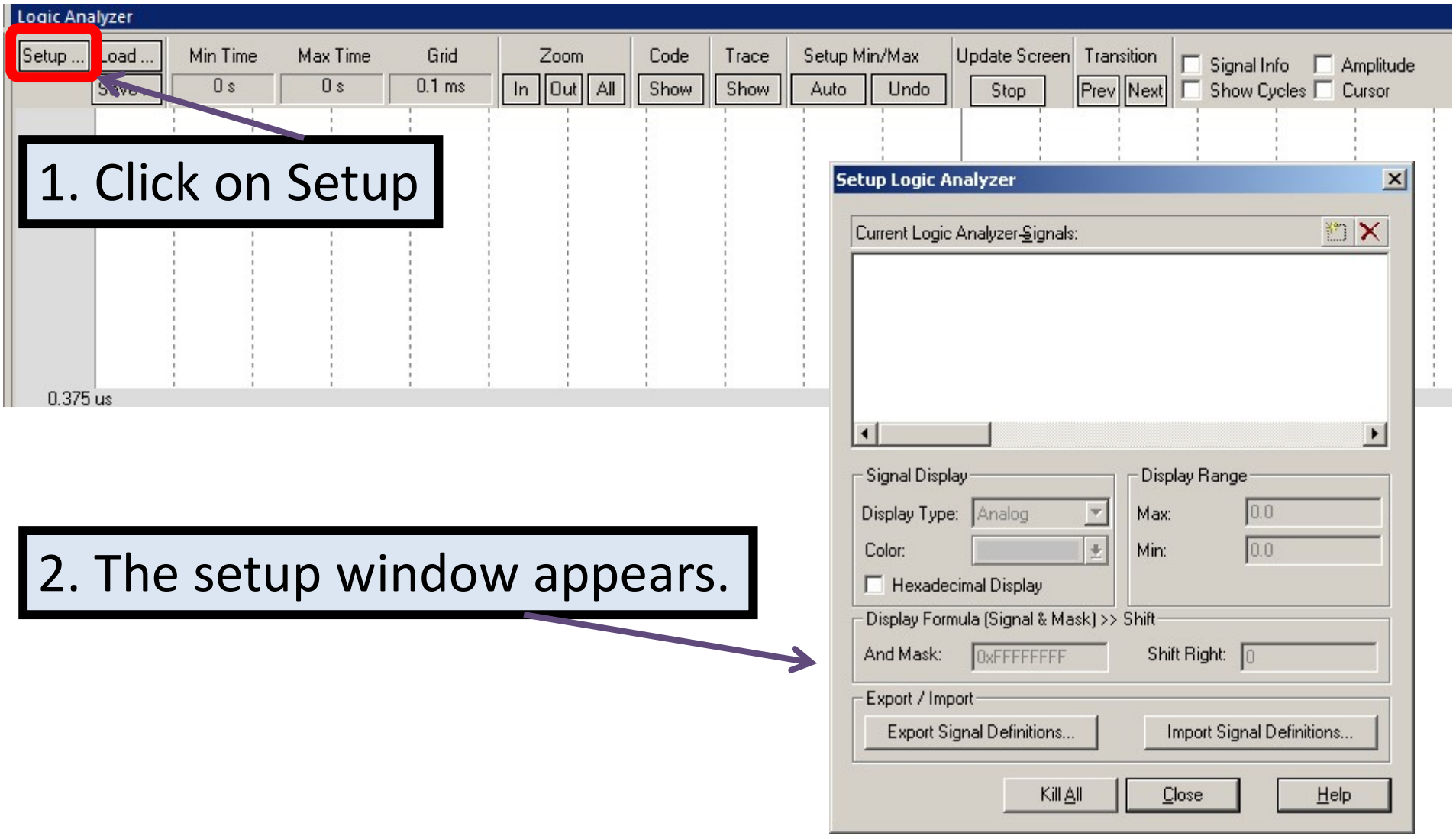
Window corresponding to Timer 0

Logic analyzer

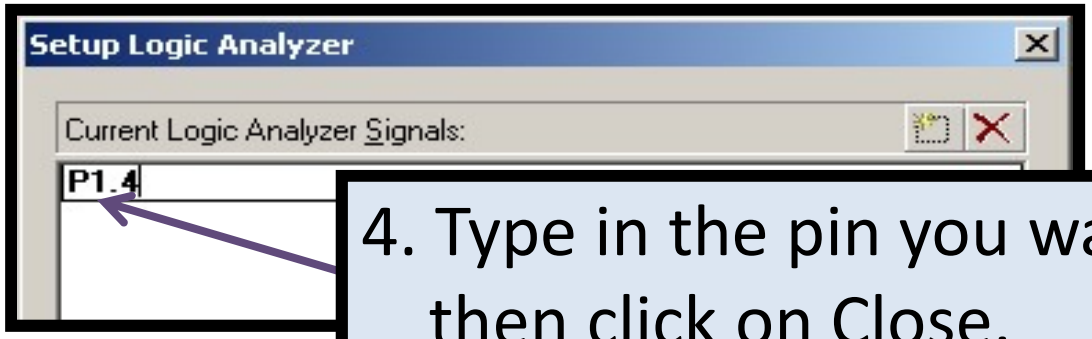
To start the logic analyzer click on the highlighted icon or go to
View > Analysis Window > Logic Analyzer.



Logic Analyzer window

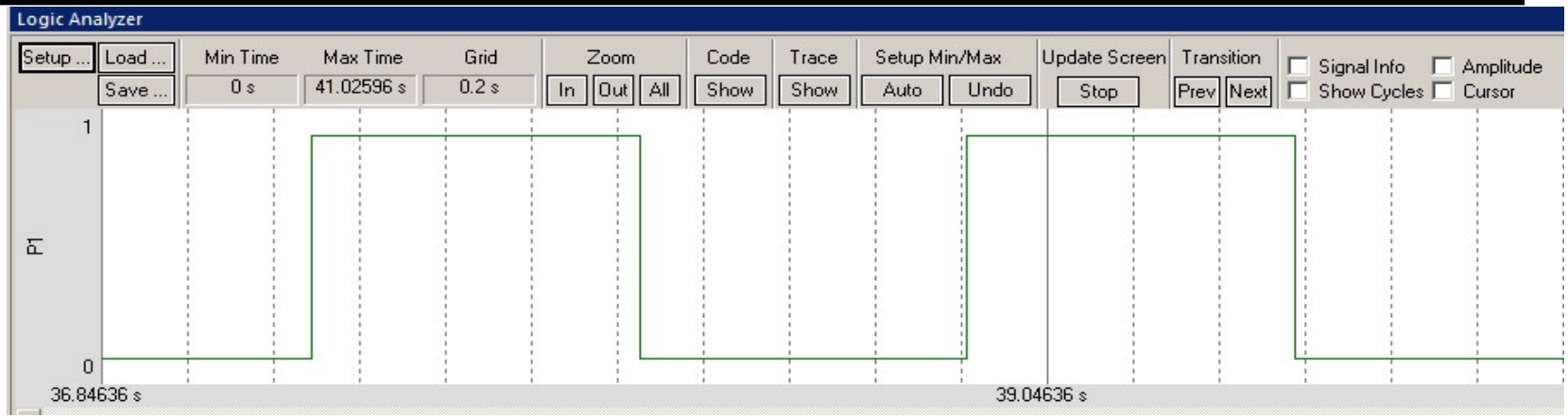


3. Click on the “New/insert” icon.



4. Type in the pin you want to monitor then click on Close.

5. After running a simulation, you can pause it and look at the timing waveforms to debug your code.



Questions ?

Thank you

WEL Labs, IITB

2016

For doubts/errors in this PPT contact :
Suryakant Toraskar e-mail: smtoraskar.iitbombay@gmail.com location : WEL5