Tutorial 08: ARM Data Definition Directives

Computer Science Department
CS2208: Introduction to Computer Organization and Architecture
Winter 2020-2021
Instructor: Mahmoud R. El-Sakka
Office: MC-419
Email: elsakka@csd.uwo.ca
Phone: 519-661-2111 x86996

Music: *"Corporate Success"* by *Scott Holmes*, used under *Attribution-NonCommercial License*

ARM Assembly Directives

❑ Assembly language directives include:

AREA		To name a region of **code** or **data**
ENTRY		The execution starting point
END		The physical end of the program
name EQU	v. expr	Equate a *name* to the *value* of the v. expr ***Will not make any memory allocation, i.e.,*** ***Similar to*** **`#define`** ***in*** **C**
{label} DCD	v. expr {, v. expr } …	Set up one or more *32-bit constant* in memory *Must start at a multiple of 4 address location*
{label} DCW	v. expr {, v. expr } …	Set up one or more *16-bit constant* in memory *Must start at an even address location*
{label} DCB	v. expr {, v. expr } …	Set up one or more *8-bit constant* in memory *Can start anywhere*
{label} SPACE	size expr	Reserves a zeroed block of memory *Can start anywhere*
ALIGN		Ensures that next instruction is correctly aligned on 32-bit boundaries, i.e., to start at a multiple of 4 address location

ARM Assembly Directives

- The “**v. expr**” can be any *constant-value expression*, i.e., its value MUST be evaluated during assembly phase, not during execution.
- The “**v. expr**” examples:
  - 2*50/3 ➔ to be evaluated to 0x21 (i.e., 33)
  - ‘A’ ➔ to be evaluated to 0x41 (i.e., 65)
  - “ABC” ➔ to be evaluated to 0x414243

The single quotation for
a single character only.
It can be used with
DCB, **DCW** or **DCD**

The double quotation for a string.
It MUST be used with **DCB**

ARM Assembly Directives

- Some symbols in Keil assembler have different meanings, based on their location within the instruction:
  - **Equal sign "="**
    - at the opcode column *means* **DCB**
    - as a prefix to the 2nd operand of an LDR instruction *means* pseudo instruction

    Example 1:

    ```
    XYZ =    0x41  ;the = sign in this context means DCB, i.e.,
    XYZ DCB 0x41
    ```

    What will happen if the "=" sign is omitted?

    Example 2:

    ```
        LDR r0,=0x12345678  ; to LDR the 32-bit value 0x12345678  into r0
        LDR r0,=PPP         ; to LDR the 32-bit address of PPP  into r0
    ```

    the = sign in this context means the **LDR** here is a pseudo instruction

  - **Ampersand sign "&"**
    - at the opcode column *means* **DCD**
    - as a prefix to an operand *means* a HEX value (i.e., similar to **0x**)

    Example 3:

    ```
    AAA &    0x123456  ;the & sign in this context means DCD, i.e.,
    AAA DCD 0x123456
    ```

    Example 4:

    ```
        MOV r0,#&8F    ;the & sign in this context means a HEX value
    ```

  - **Percent sign "%"**
    - at the opcode column *means* **SPACE**

    Example 5:

    ```
    BBB %      0x40    ;the % sign in this context means SPACE, i.e.,
    BBB SPACE 0x40
    ```

Writing Numbers with Various Radix

❑ The Keil assembler uses

- a prefix **0x** or **&** to indicate ***hexadecimal*** constant, e.g.,

  ```
  MOV r1, #0x9C
  MOV r1, #&9C
  ```

  or

  ```
  DCD 0x9C
  DCD &9C
  ```

In ARM assembly,
the "#" means
Literal or immediate
addressing mode

- a prefix **2_** to indicate ***binary*** constant, e.g.,

  ```
  MOV r1, #2_10011100
  ```

  or

  ```
  DCD 2_10011100
  ```

In ARM assembly,
It is ***illegal*** to use **"#"** with
DCD, **DCW**, or **DCB**

- a prefix **8_** to indicate ***octal*** constant, e.g.,

  ```
  MOV r1, #8_234
  ```

  or

  ```
  DCD 8_234
  ```

- **no** prefix to indicate ***decimal*** constant, e.g.,

  ```
  MOV r1, #156
  ```

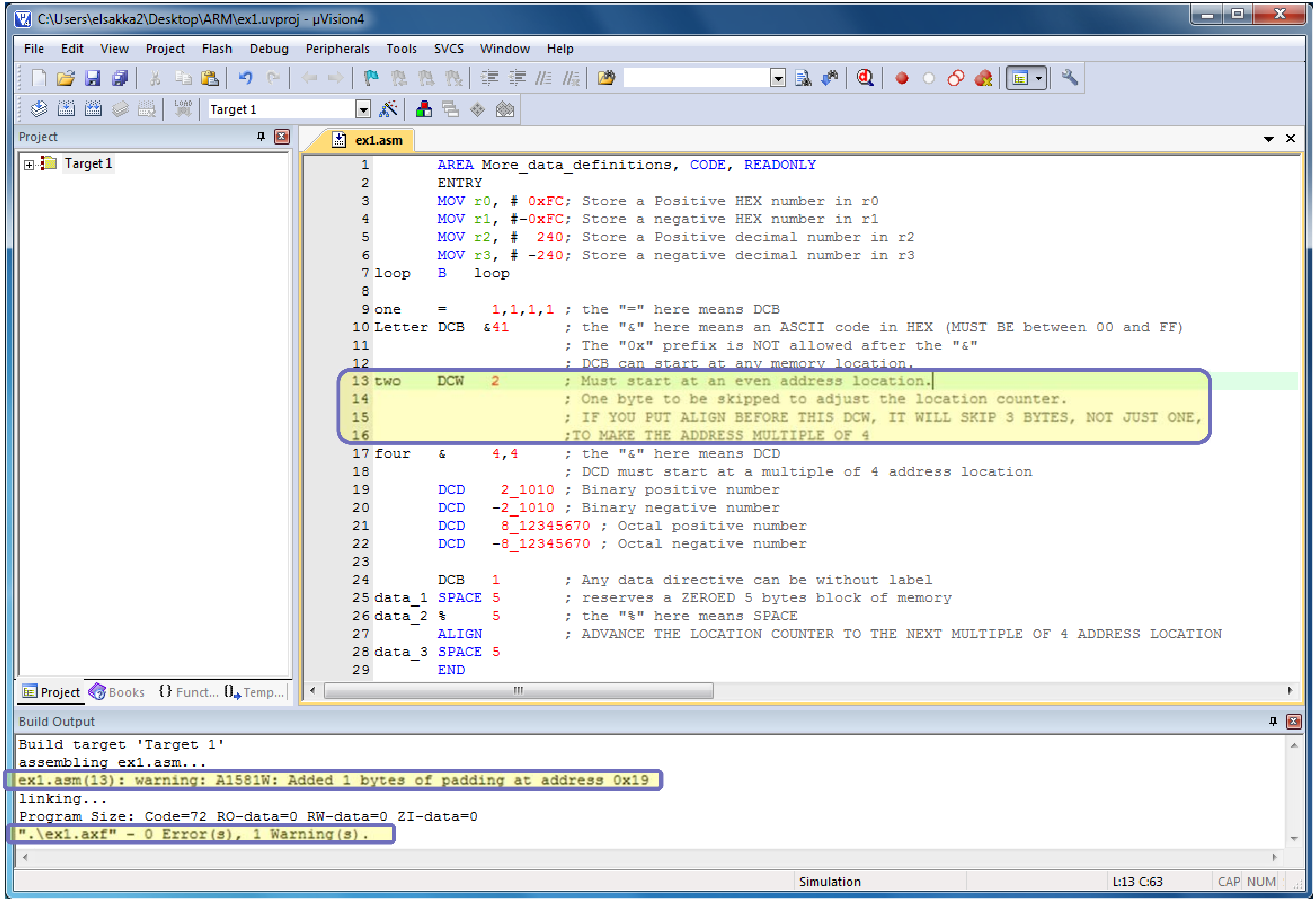
  or

  ```
  DCD 156
  ```

Data Definition Directives

```
       AREA More_data_definitions, CODE, READONLY
       ENTRY
       MOV r0, # 0xFC; Store a Positive HEX number in r0
       MOV r1, #-0xFC; Store a negative HEX number in r1
       MOV r2, #  240; Store a Positive decimal number in r2
       MOV r3, # -240; Store a negative decimal number in r3
loop   B   loop

one    =   1,1,1,1 ; the "=" here means DCB
Letter DCB &41     ; the "&" here means an ASCII code in HEX (MUST BE between 00 and FF)
                   ; The "0x" prefix is NOT allowed after the "&"
                   ; DCB can start at any memory location.
two    DCW  2      ; Must start at an even address location.
                   ; One byte to be skipped to adjust the location counter.
                   ; IF YOU PUT ALIGN BEFORE THIS DCW, IT WILL SKIP 3 BYTES, NOT JUST ONE,
                   ; TO MAKE THE ADDRESS MULTIPLE OF 4
four   &     4,4   ; the "&" here means DCD
                   ; DCD must start at a multiple of 4 address location
       DCD   2_1010      ; Binary positive number
       DCD   -2_1010     ; Binary negative number
       DCD    8_12345670 ; Octal positive number
       DCD   -8_12345670 ; Octal negative number

       DCB   1           ; Any data directive can be without label
data_1 SPACE 5           ; reserves a ZEROED 5 bytes block of memory
data_2 %     5           ; the "%" here means SPACE
       ALIGN     ; ADVANCE THE LOCATION COUNTER TO THE NEXT MULTIPLE OF 4 ADDRESS LOCATION
data_3 SPACE 5
       END
```

Data Definition Directives

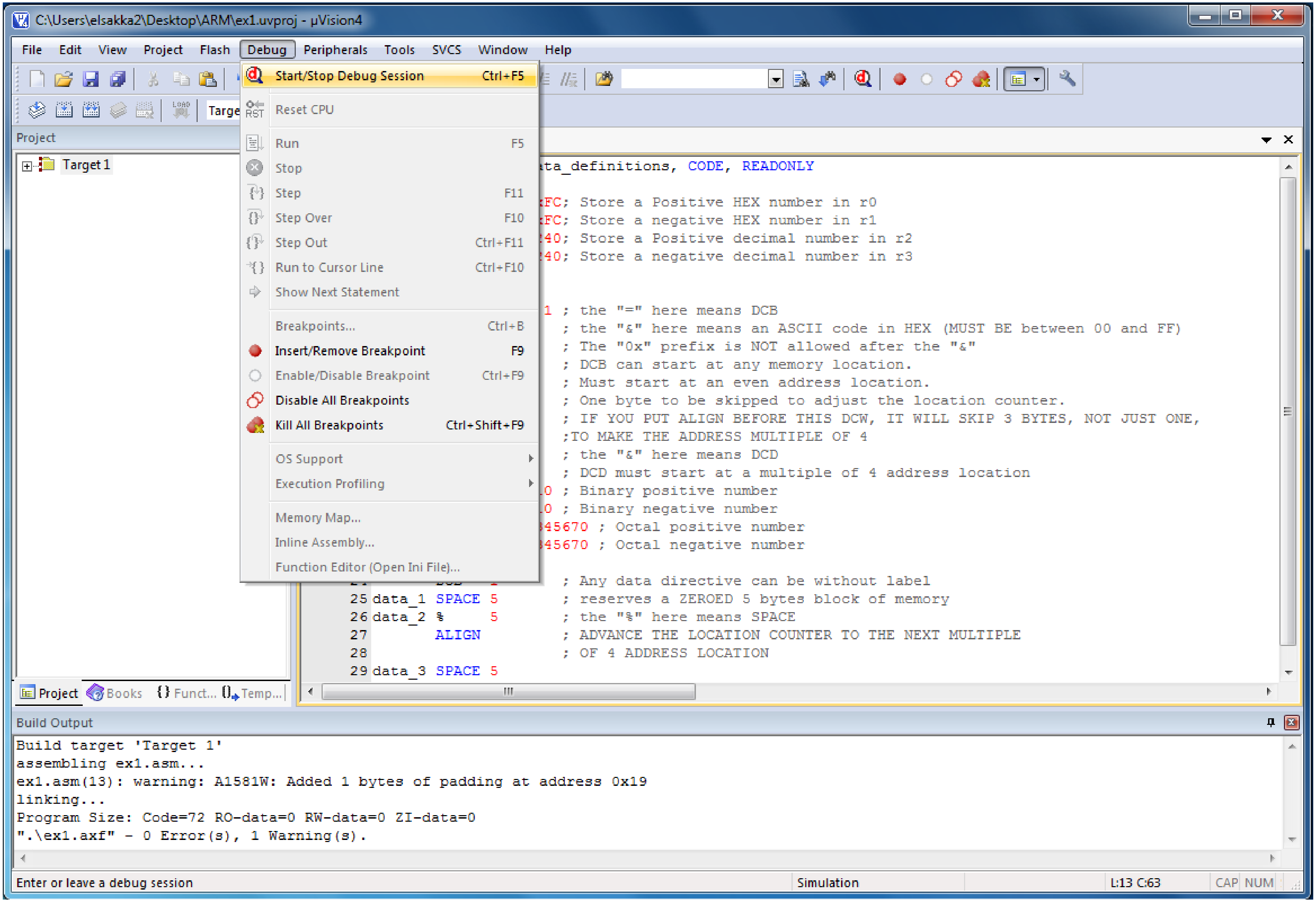


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
ex1.asm
Project Books {} Funct... Temp...
Build Output
Simulation
L:13 C:63
CAP NUM

```
 1         AREA More_data_definitions, CODE, READONLY
 2         ENTRY
 3         MOV r0, # 0xFC; Store a Positive HEX number in r0
 4         MOV r1, #-0xFC; Store a negative HEX number in r1
 5         MOV r2, #  240; Store a Positive decimal number in r2
 6         MOV r3, # -240; Store a negative decimal number in r3
 7 loop    B    loop
 8
 9 one     =      1,1,1,1 ; the "=" here means DCB
10 Letter DCB   &41       ; the "&" here means an ASCII code in HEX (MUST BE between 00 and FF)
11                        ; The "0x" prefix is NOT allowed after the "&"
12                        ; DCB can start at any memory location.
13 two     DCW    2       ; Must start at an even address location.
14                        ; One byte to be skipped to adjust the location counter.
15                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT WILL SKIP 3 BYTES, NOT JUST ONE,
16                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
17 four    &      4,4     ; the "&" here means DCD
18                        ; DCD must start at a multiple of 4 address location
19         DCD     2_1010 ; Binary positive number
20         DCD    -2_1010 ; Binary negative number
21         DCD     8_12345670 ; Octal positive number
22         DCD    -8_12345670 ; Octal negative number
23
24         DCB    1       ; Any data directive can be without label
25 data_1 SPACE 5         ; reserves a ZEROED 5 bytes block of memory
26 data_2 %      5        ; the "%" here means SPACE
27         ALIGN          ; ADVANCE THE LOCATION COUNTER TO THE NEXT MULTIPLE OF 4 ADDRESS LOCATION
28 data_3 SPACE 5
29         END
```

```
Build target 'Target 1'
assembling ex1.asm...
ex1.asm(13): warning: A1581W: Added 1 bytes of padding at address 0x19
linking...
Program Size: Code=72 RO-data=0 RW-data=0 ZI-data=0
".\ex1.axf" - 0 Error(s), 1 Warning(s).
```

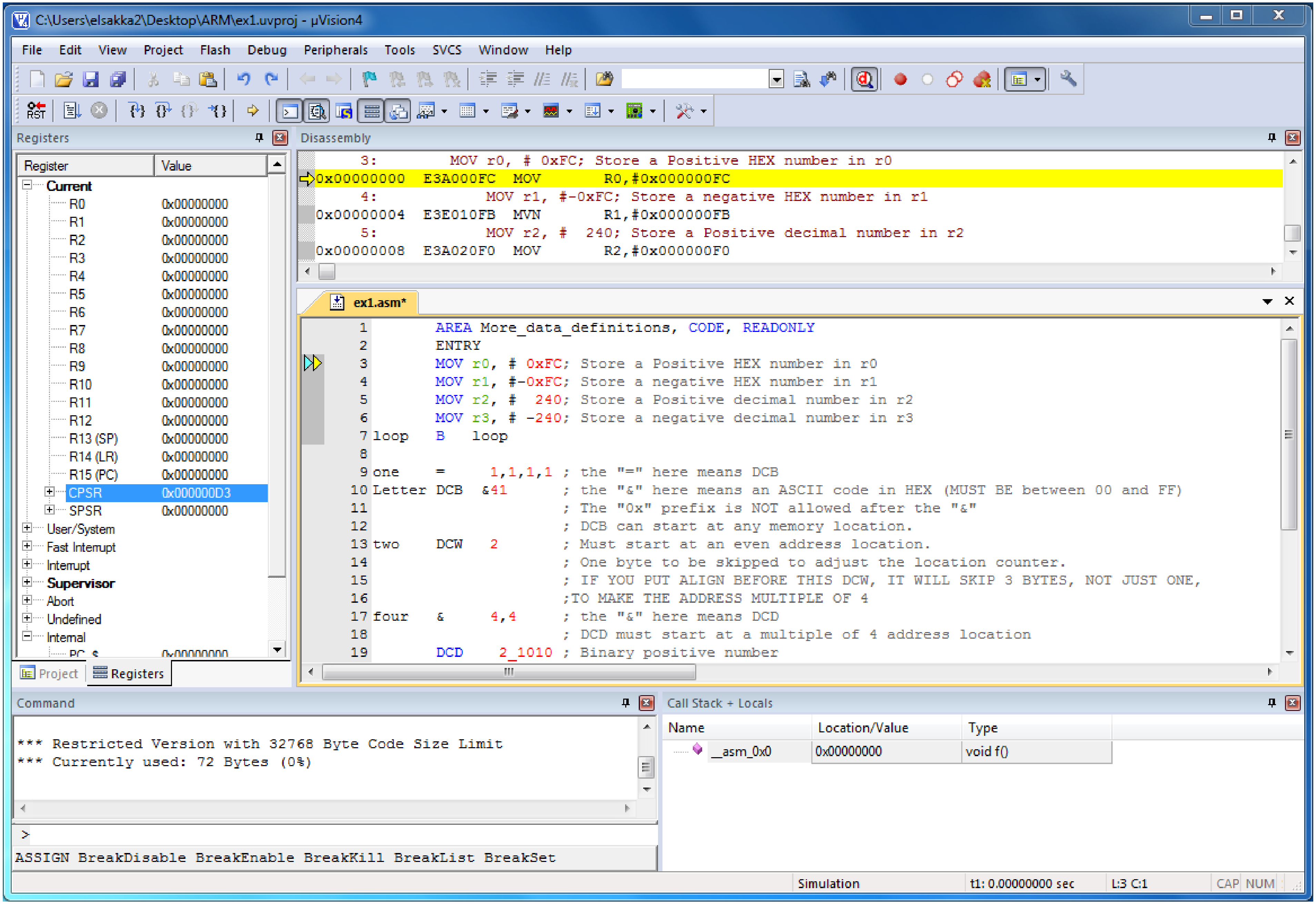
Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Targe
Project
Target 1
Start/Stop Debug Session Ctrl+F5
Reset CPU
Run F5
Stop
Step F11
Step Over F10
Step Out Ctrl+F11
Run to Cursor Line Ctrl+F10
Show Next Statement
Breakpoints... Ctrl+B
Insert/Remove Breakpoint F9
Enable/Disable Breakpoint Ctrl+F9
Disable All Breakpoints
Kill All Breakpoints Ctrl+Shift+F9
OS Support
Execution Profiling
Memory Map...
Inline Assembly...
Function Editor (Open Ini File)...
Project Books Funct... Temp...
Build Output
Enter or leave a debug session
Simulation
L:13 C:63
CAP NUM

```
ta_definitions, CODE, READONLY

FC; Store a Positive HEX number in r0
FC; Store a negative HEX number in r1
40; Store a Positive decimal number in r2
40; Store a negative decimal number in r3


1  ; the "=" here means DCB
   ; the "&" here means an ASCII code in HEX (MUST BE between 00 and FF)
   ; The "0x" prefix is NOT allowed after the "&"
   ; DCB can start at any memory location.
   ; Must start at an even address location.
   ; One byte to be skipped to adjust the location counter.
   ; IF YOU PUT ALIGN BEFORE THIS DCW, IT WILL SKIP 3 BYTES, NOT JUST ONE,
   ;TO MAKE THE ADDRESS MULTIPLE OF 4
   ; the "&" here means DCD
   ; DCD must start at a multiple of 4 address location
0  ; Binary positive number
0  ; Binary negative number
45670 ; Octal positive number
45670 ; Octal negative number

                         ; Any data directive can be without label
data_1 SPACE 5           ; reserves a ZEROED 5 bytes block of memory
data_2 %     5           ; the "%" here means SPACE
       ALIGN             ; ADVANCE THE LOCATION COUNTER TO THE NEXT MULTIPLE
                         ; OF 4 ADDRESS LOCATION
data_3 SPACE 5
```

```
Build target 'Target 1'
assembling ex1.asm...
ex1.asm(13): warning: A1581W: Added 1 bytes of padding at address 0x19
linking...
Program Size: Code=72 RO-data=0 RW-data=0 ZI-data=0
".\ex1.axf" - 0 Error(s), 1 Warning(s).
```

Data Definition Directives



C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Call Stack + Locals

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000

```
      3:            MOV r0, # 0xFC; Store a Positive HEX number in r0
0x00000000  E3A000FC  MOV       R0,#0x000000FC
      4:                MOV r1, #-0xFC; Store a negative HEX number in r1
0x00000004  E3E010FB  MVN       R1,#0x000000FB
      5:                MOV r2, #  240; Store a Positive decimal number in r2
0x00000008  E3A020F0  MOV       R2,#0x000000F0
```

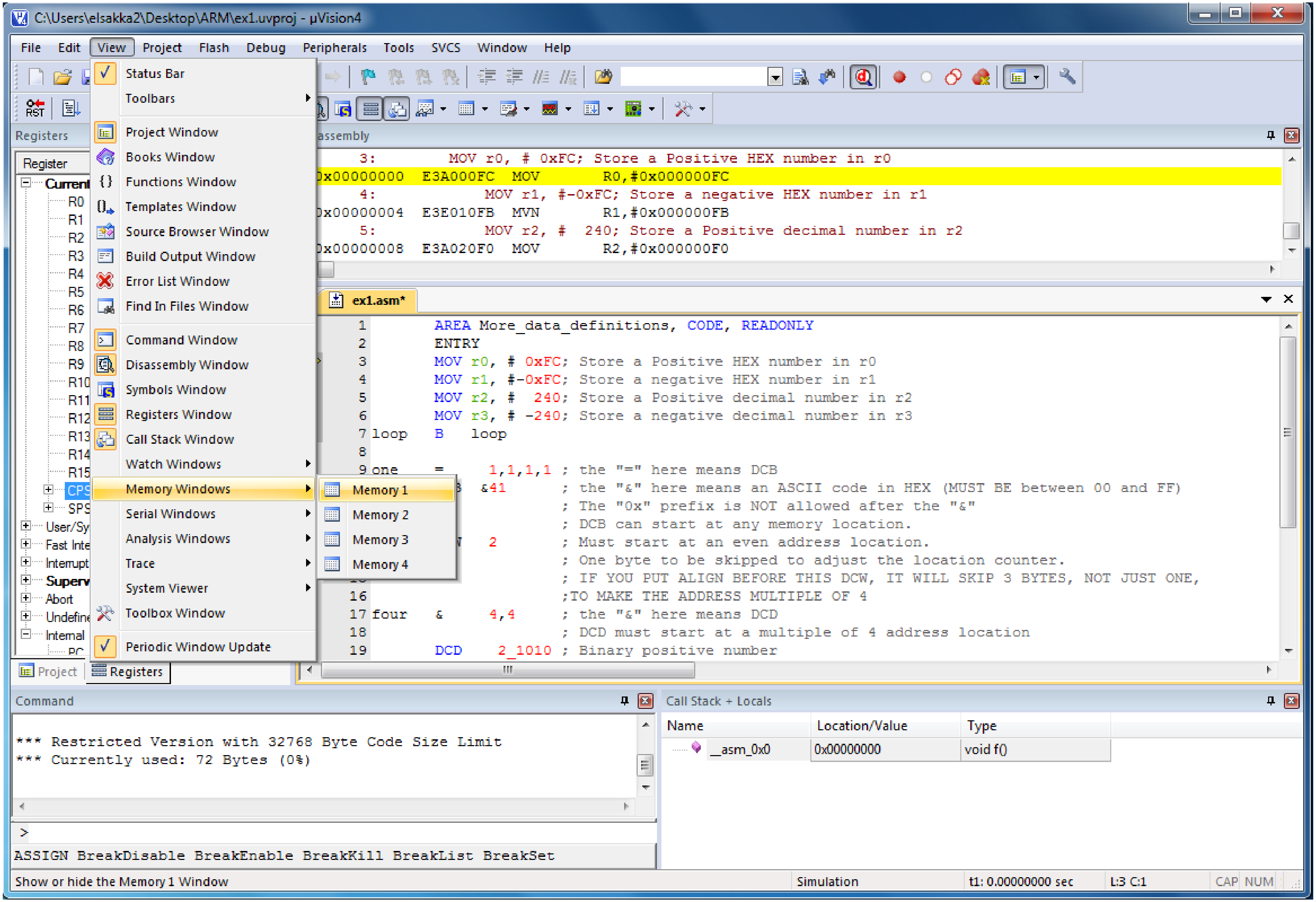
```
       AREA More_data_definitions, CODE, READONLY
       ENTRY
       MOV r0, # 0xFC; Store a Positive HEX number in r0
       MOV r1, #-0xFC; Store a negative HEX number in r1
       MOV r2, #  240; Store a Positive decimal number in r2
       MOV r3, # -240; Store a negative decimal number in r3
loop   B   loop

one    =      1,1,1,1 ; the "=" here means DCB
Letter DCB   &41      ; the "&" here means an ASCII code in HEX (MUST BE between 00 and FF)
                      ; The "0x" prefix is NOT allowed after the "&"
                      ; DCB can start at any memory location.
two    DCW    2       ; Must start at an even address location.
                      ; One byte to be skipped to adjust the location counter.
                      ; IF YOU PUT ALIGN BEFORE THIS DCW, IT WILL SKIP 3 BYTES, NOT JUST ONE,
                      ;TO MAKE THE ADDRESS MULTIPLE OF 4
four   &      4,4     ; the "&" here means DCD
                      ; DCD must start at a multiple of 4 address location
       DCD     2_1010 ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Name	Location/Value	Type
_asm_0x0	0x00000000	void f()

Simulation   t1: 0.00000000 sec   L:3 C:1   CAP NUM

Data Definition Directives



```
      3:          MOV r0, # 0xFC; Store a Positive HEX number in r0
0x00000000  E3A000FC  MOV       R0,#0x000000FC
      4:              MOV r1, #-0xFC; Store a negative HEX number in r1
0x00000004  E3E010FB  MVN       R1,#0x000000FB
      5:              MOV r2, #  240; Store a Positive decimal number in r2
0x00000008  E3A020F0  MOV       R2,#0x000000F0
```

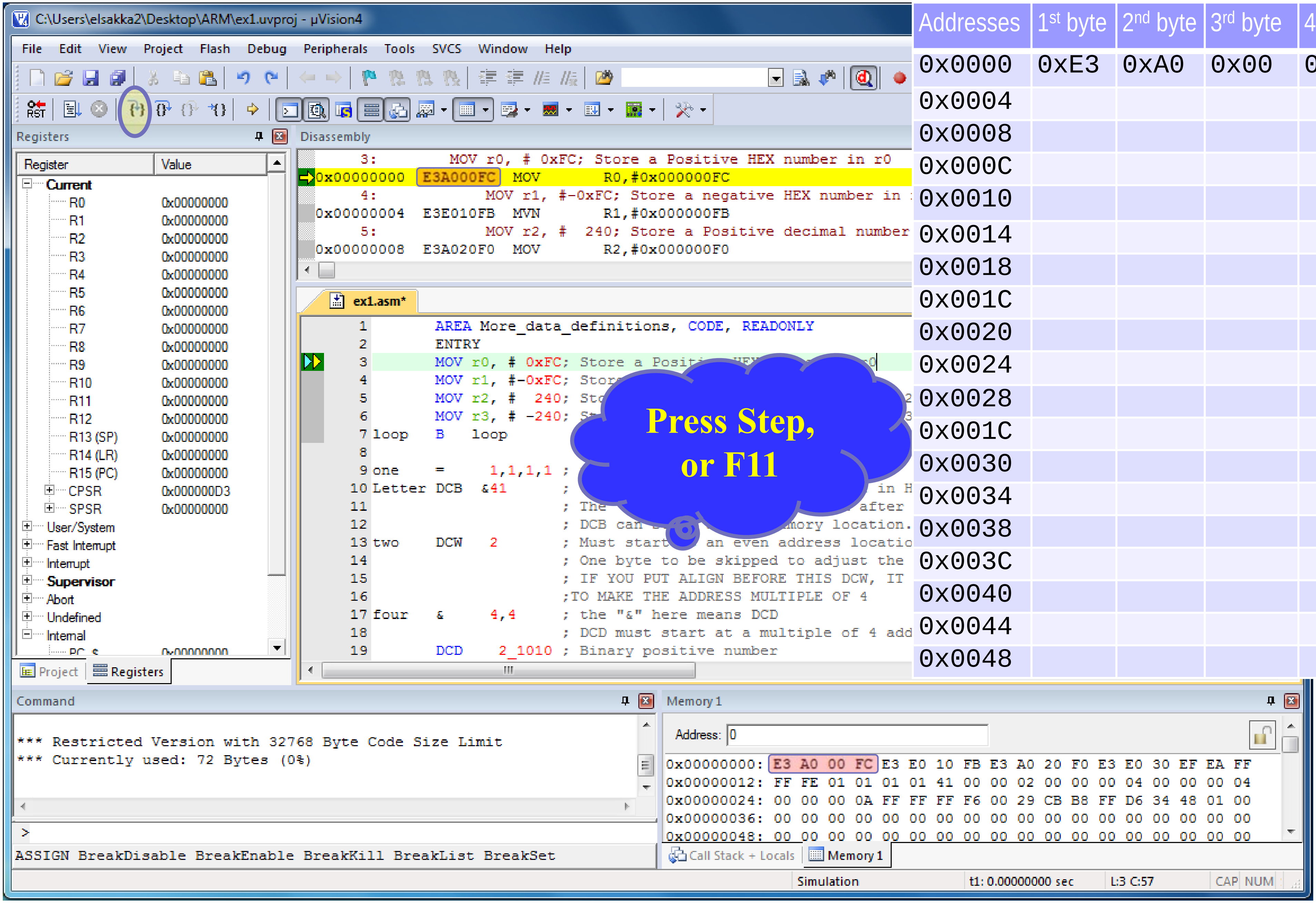
```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B   loop

one     =      1,1,1,1 ; the "=" here means DCB
           3  &41      ; the "&" here means an ASCII code in HEX (MUST BE between 00 and FF)
                       ; The "0x" prefix is NOT allowed after the "&"
                       ; DCB can start at any memory location.
           W   2       ; Must start at an even address location.
                       ; One byte to be skipped to adjust the location counter.
                       ; IF YOU PUT ALIGN BEFORE THIS DCW, IT WILL SKIP 3 BYTES, NOT JUST ONE,
                       ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4     ; the "&" here means DCD
                       ; DCD must start at a multiple of 4 address location
        DCD     2_1010 ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Name	Location/Value	Type
_asm_0x0	0x00000000	void f()

Show or hide the Memory 1 Window | Simulation | t1: 0.00000000 sec | L:3 C:1 | CAP NUM

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
Project
Registers
Command
Memory 1
Call Stack + Locals

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000

```
      3:            MOV r0, # 0xFC; Store a Positive HEX number in r0
0x00000000  E3A000FC  MOV         R0,#0x000000FC
      4:                MOV r1, #-0xFC; Store a negative HEX number in
0x00000004  E3E010FB  MVN         R1,#0x000000FB
      5:                MOV r2, #  240; Store a Positive decimal number
0x00000008  E3A020F0  MOV         R2,#0x000000F0
```

ex1.asm*

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positi
        MOV r1, #-0xFC; Stor
        MOV r2, #  240; Sto
        MOV r3, # -240; S
loop    B    loop

one     =      1,1,1,1 ;
Letter  DCB   &41      ;                                   in H
                       ; The                              after
                       ; DCB can                  mory location.
two     DCW    2       ; Must start    an even address locatio
                       ; One byte to be skipped to adjust the
                       ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                       ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4     ; the "&" here means DCD
                       ; DCD must start at a multiple of 4 add
        DCD     2_1010 ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

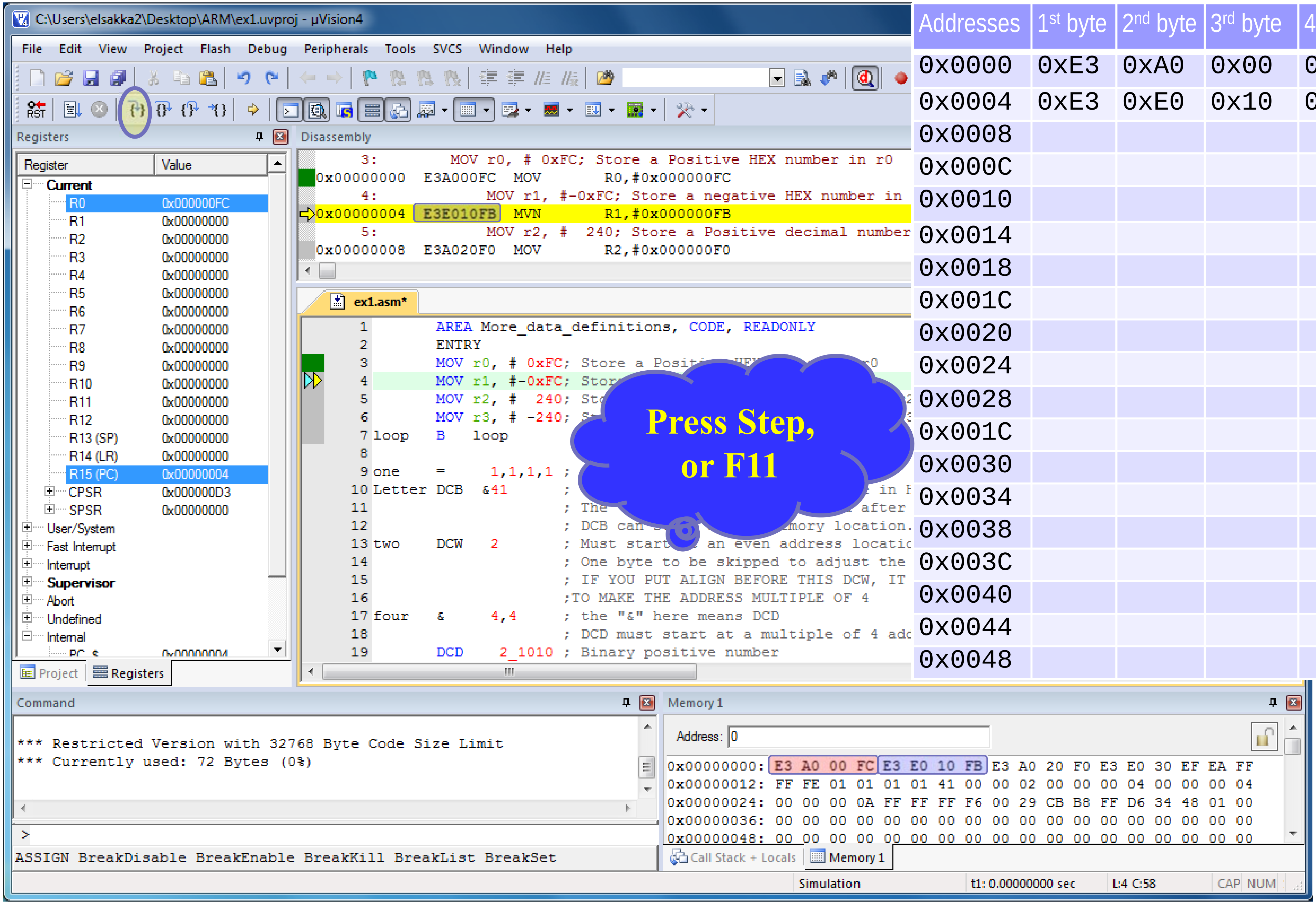
```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:3 C:57   CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004				
0x0008				
0x000C				
0x0010				
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Press Step, or F11

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1

Register	Value
Current	
R0	0x000000FC
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000004
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000004

```
     3:        MOV r0, # 0xFC; Store a Positive HEX number in r0
0x00000000  E3A000FC  MOV      R0,#0x000000FC
     4:        MOV r1, #-0xFC; Store a negative HEX number in
0x00000004  E3E010FB  MVN      R1,#0x000000FB
     5:        MOV r2, #  240; Store a Positive decimal number
0x00000008  E3A020F0  MOV      R2,#0x000000F0
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positi                r0
        MOV r1, #-0xFC; Stor
        MOV r2, #  240; Sto
        MOV r3, # -240; S
loop    B    loop

one     =       1,1,1,1 ;
Letter  DCB   &41       ;                              in H
                        ; The                          after
                        ; DCB can                 mory location.
two     DCW    2        ; Must start    an even address locatic
                        ; One byte to be skipped to adjust the
                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4     ; the "&" here means DCD
                        ; DCD must start at a multiple of 4 adc
        DCD     2_1010  ; Binary positive number
```

Press Step,
or F11

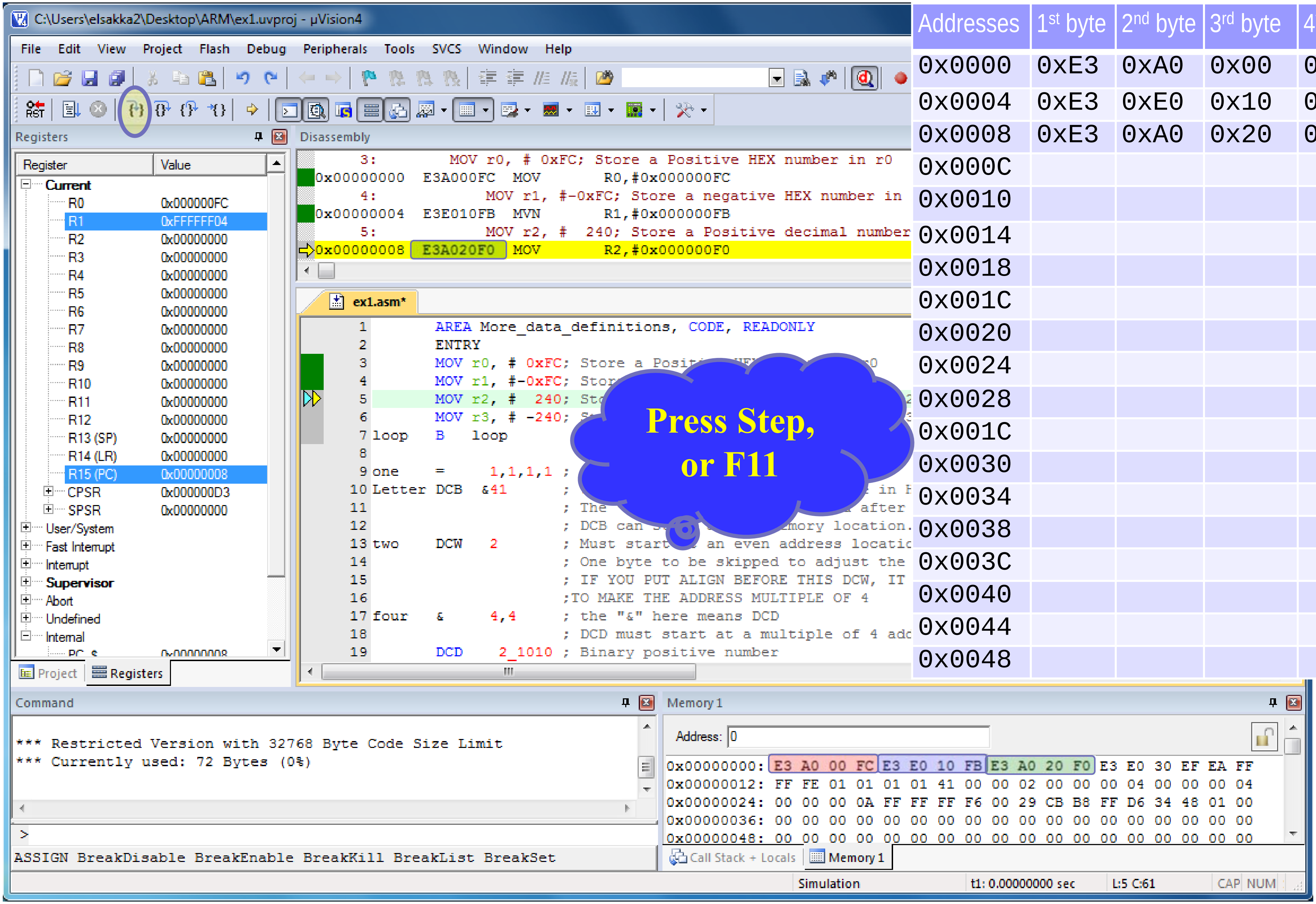
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Simulation    t1: 0.00000000 sec    L:4 C:58    CAP  NUM
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008				
0x000C				
0x0010				
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals
Press Step, or F11

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000008

```
     3:          MOV r0, # 0xFC; Store a Positive HEX number in r0
0x00000000  E3A000FC  MOV       R0,#0x000000FC
     4:              MOV r1, #-0xFC; Store a negative HEX number in
0x00000004  E3E010FB  MVN       R1,#0x000000FB
     5:              MOV r2, #  240; Store a Positive decimal number
0x00000008  E3A020F0  MOV       R2,#0x000000F0
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Posit                          r0
        MOV r1, #-0xFC; Stor
        MOV r2, #  240; Sto
        MOV r3, # -240; S
loop    B    loop

one     =      1,1,1,1 ;
Letter  DCB   &41      ;                                    in H
                       ; The                              after
                       ; DCB can                     mory location.
two     DCW    2       ; Must start     an even address locatic
                       ; One byte to be skipped to adjust the
                       ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                       ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4     ; the "&" here means DCD
                       ; DCD must start at a multiple of 4 add
        DCD    2_1010  ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

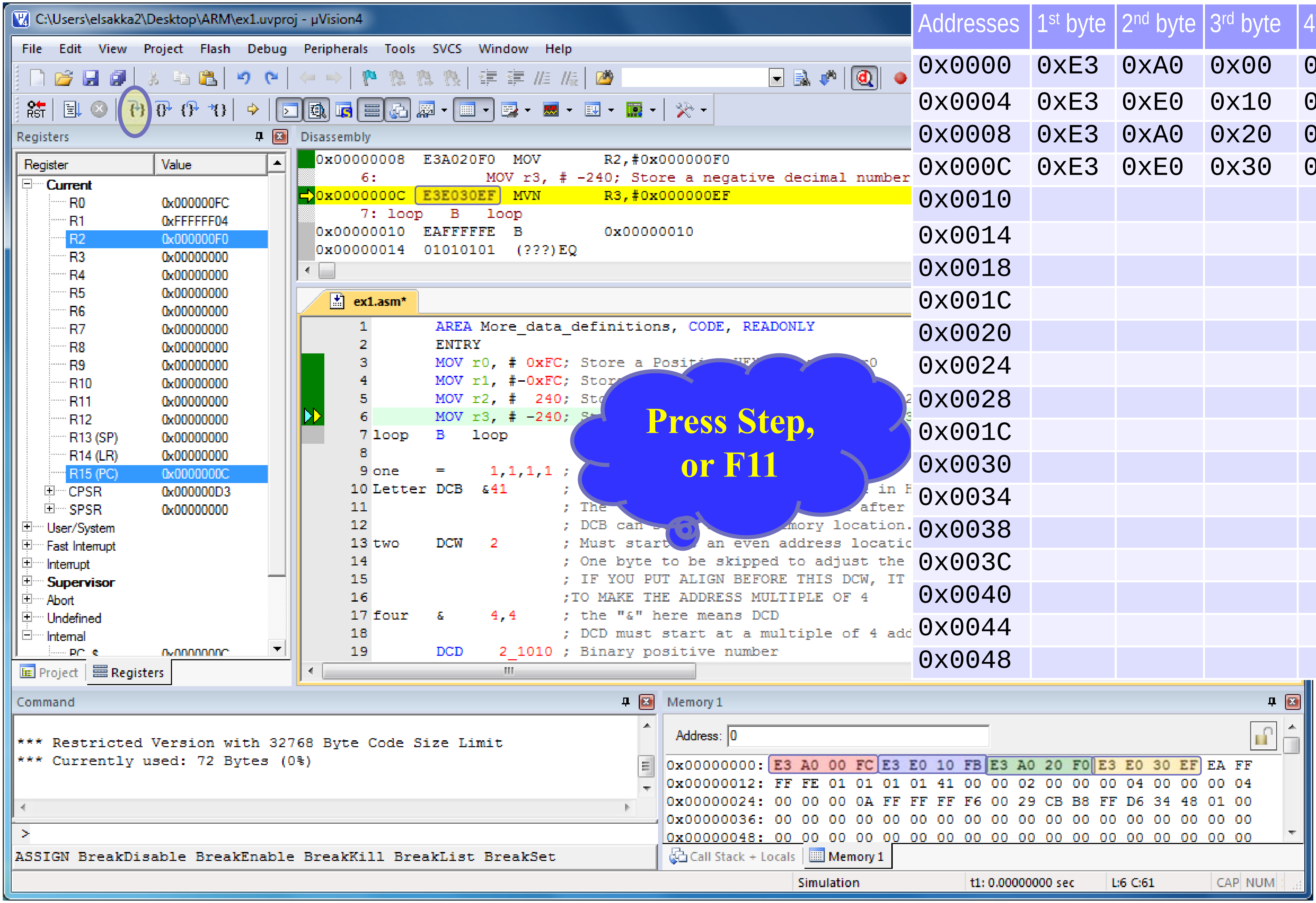
Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation t1: 0.00000000 sec L:5 C:61 CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C				
0x0010				
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x0000000C

```
0x00000008  E3A020F0  MOV      R2,#0x000000F0
     6:             MOV r3, # -240; Store a negative decimal number
0x0000000C  E3E030EF  MVN      R3,#0x000000EF
     7: loop   B   loop
0x00000010  EAFFFFFE  B        0x00000010
0x00000014  01010101  (???)EQ
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positi                    r0
        MOV r1, #-0xFC; Stor
        MOV r2, #  240; Sto                                 2
        MOV r3, # -240; S                                   3
loop    B    loop

one     =       1,1,1,1 ;
Letter  DCB   &41       ;                              in H
                        ; The                         after
                        ; DCB can                mory location.
two     DCW    2        ; Must start    an even address locatio
                        ; One byte to be skipped to adjust the
                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4     ; the "&" here means DCD
                        ; DCD must start at a multiple of 4 add
        DCD     2_1010  ; Binary positive number
```

Press Step, or F11

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

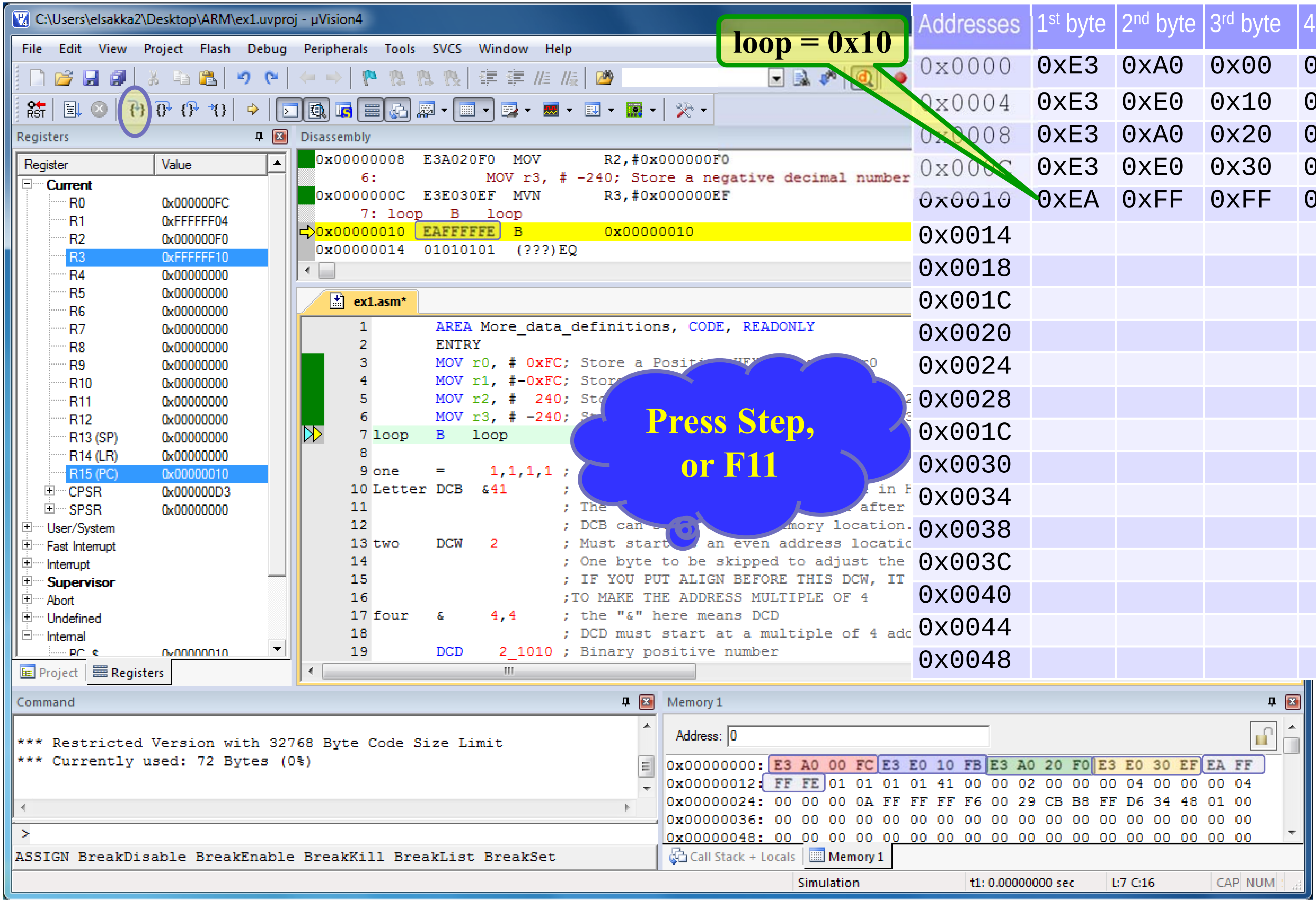
Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:6 C:61   CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010				
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

```
Register        Value
Current
  R0            0x000000FC
  R1            0xFFFFFF04
  R2            0x000000F0
  R3            0xFFFFFF10
  R4            0x00000000
  R5            0x00000000
  R6            0x00000000
  R7            0x00000000
  R8            0x00000000
  R9            0x00000000
  R10           0x00000000
  R11           0x00000000
  R12           0x00000000
  R13 (SP)      0x00000000
  R14 (LR)      0x00000000
  R15 (PC)      0x00000010
  CPSR          0x000000D3
  SPSR          0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC $          0x00000010
```

```
0x00000008  E3A020F0  MOV       R2,#0x000000F0
     6:             MOV r3, # -240; Store a negative decimal number
0x0000000C  E3E030EF  MVN       R3,#0x000000EF
     7: loop   B   loop
0x00000010  EAFFFFFE  B         0x00000010
0x00000014  01010101  (???)EQ
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positi
        MOV r1, #-0xFC; Stor
        MOV r2, #  240; St
        MOV r3, # -240; S
loop    B   loop

one     =       1,1,1,1 ;
Letter  DCB   &41       ;                                  in H
                        ; The                             after
                        ; DCB can                  mory location.
two     DCW    2        ; Must start    an even address locatic
                        ; One byte to be skipped to adjust the
                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4     ; the "&" here means DCD
                        ; DCD must start at a multiple of 4 add
        DCD     2_1010  ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

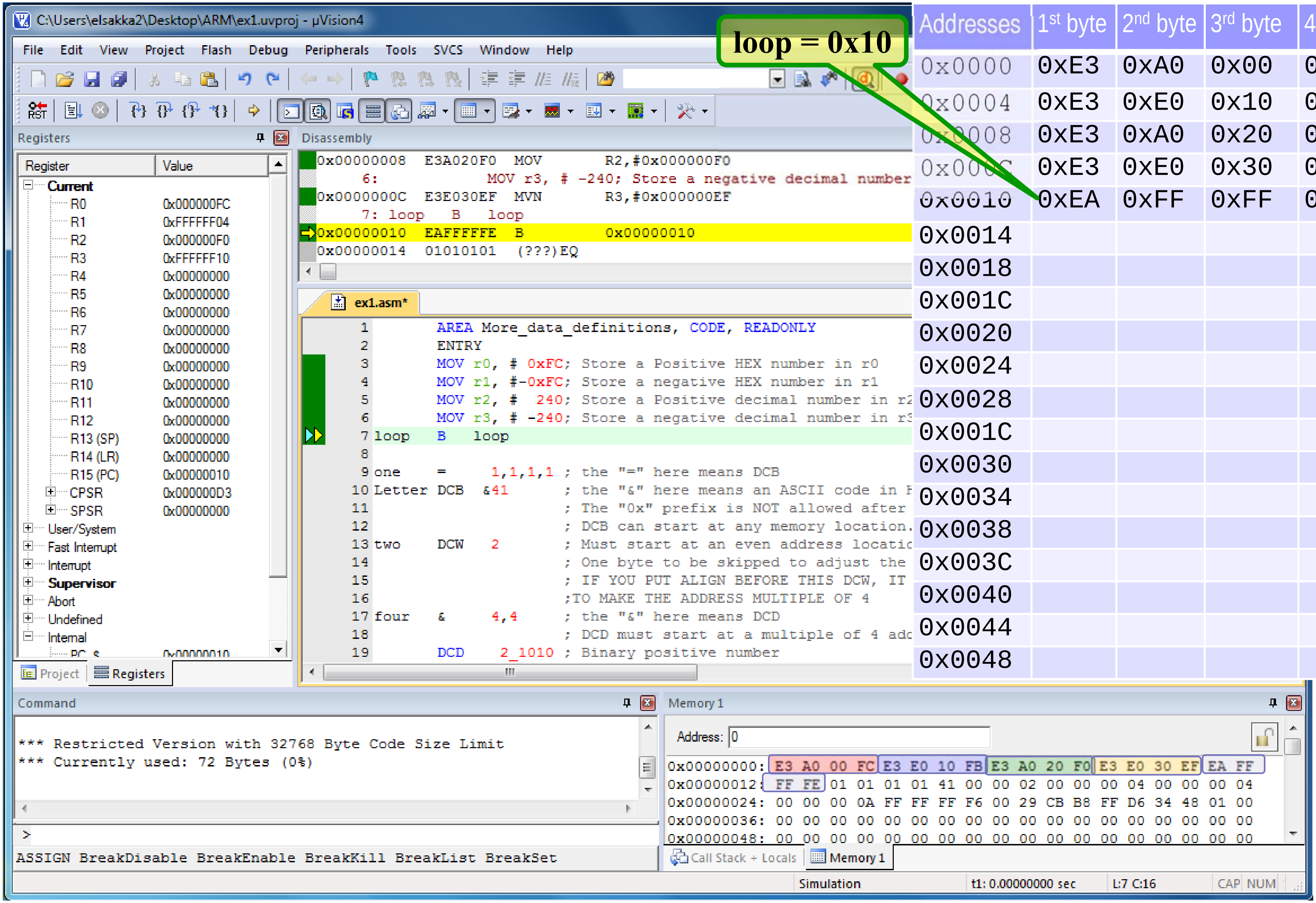
```
Simulation    t1: 0.00000000 sec    L:7 C:16    CAP NUM
```

loop = 0x10

Press Step, or F11

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives



C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

Disassembly

```
0x00000008  E3A020F0  MOV       R2,#0x000000F0
     6:            MOV r3, # -240; Store a negative decimal number
0x0000000C  E3E030EF  MVN       R3,#0x000000EF
     7: loop   B   loop
0x00000010  EAFFFFFE  B         0x00000010
0x00000014  01010101  (???)EQ
```

ex1.asm*

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B   loop

one     =     1,1,1,1  ; the "=" here means DCB
Letter  DCB   &41      ; the "&" here means an ASCII code in H
                       ; The "0x" prefix is NOT allowed after
                       ; DCB can start at any memory location.
two     DCW   2        ; Must start at an even address locatic
                       ; One byte to be skipped to adjust the
                       ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                       ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &     4,4      ; the "&" here means DCD
                       ; DCD must start at a multiple of 4 adc
        DCD   2_1010   ; Binary positive number
```

Command

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Memory 1

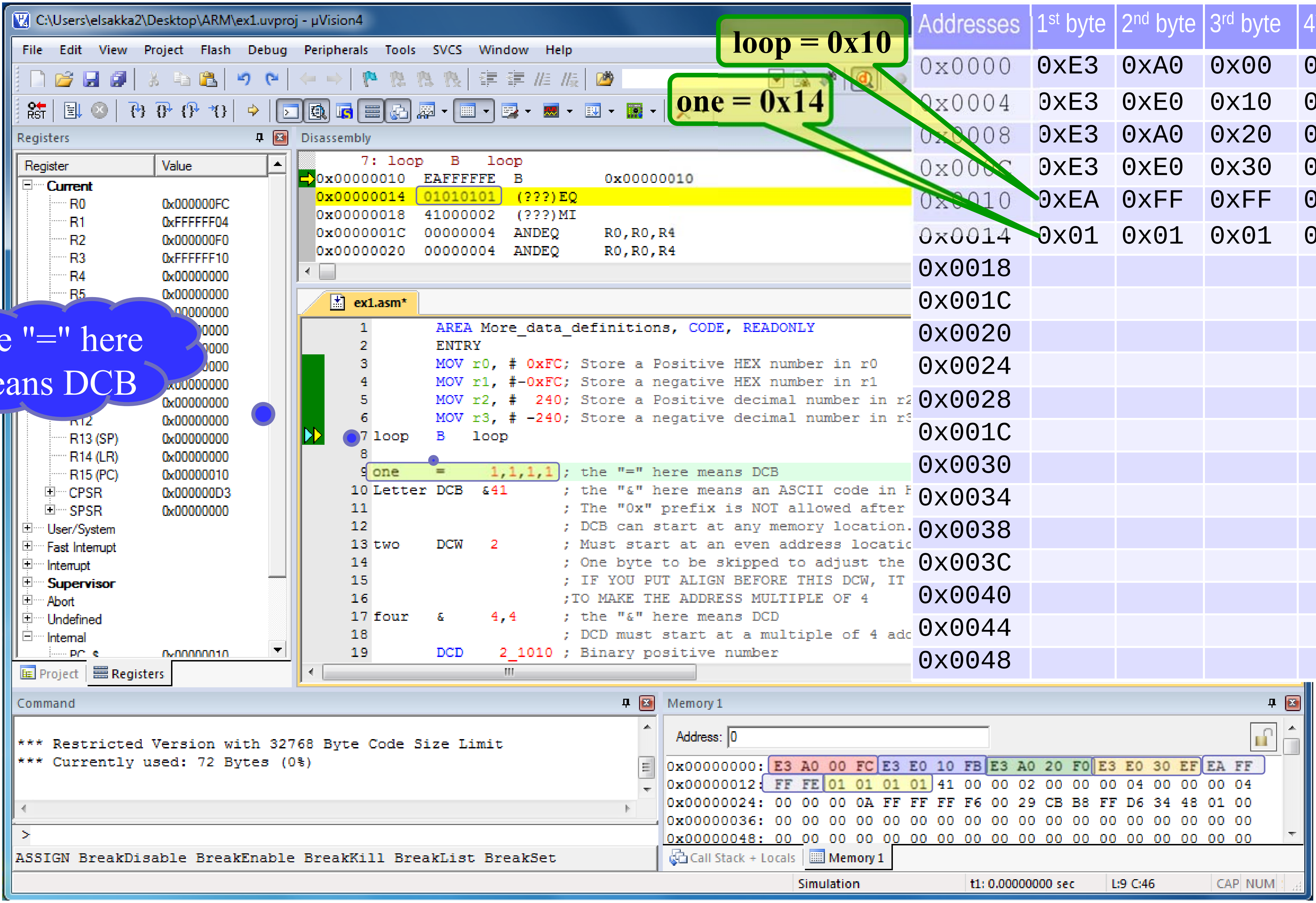
Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

loop = 0x10

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014				
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1

loop = 0x10

one = 0x14

The "=" here means DCB

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
	00000000
	0000
	0000
	0000
	00000000
	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
      7: loop    B    loop
0x00000010  EAFFFFFE  B         0x00000010
0x00000014  01010101  (???)EQ
0x00000018  41000002  (???)MI
0x0000001C  00000004  ANDEQ     R0,R0,R4
0x00000020  00000004  ANDEQ     R0,R0,R4
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B    loop

one     =       1,1,1,1 ; the "=" here means DCB
Letter DCB  &41         ; the "&" here means an ASCII code in H
                        ; The "0x" prefix is NOT allowed after
                        ; DCB can start at any memory location.
two     DCW    2        ; Must start at an even address locatio
                        ; One byte to be skipped to adjust the
                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4     ; the "&" here means DCD
                        ; DCD must start at a multiple of 4 add
        DCD     2_1010  ; Binary positive number
```

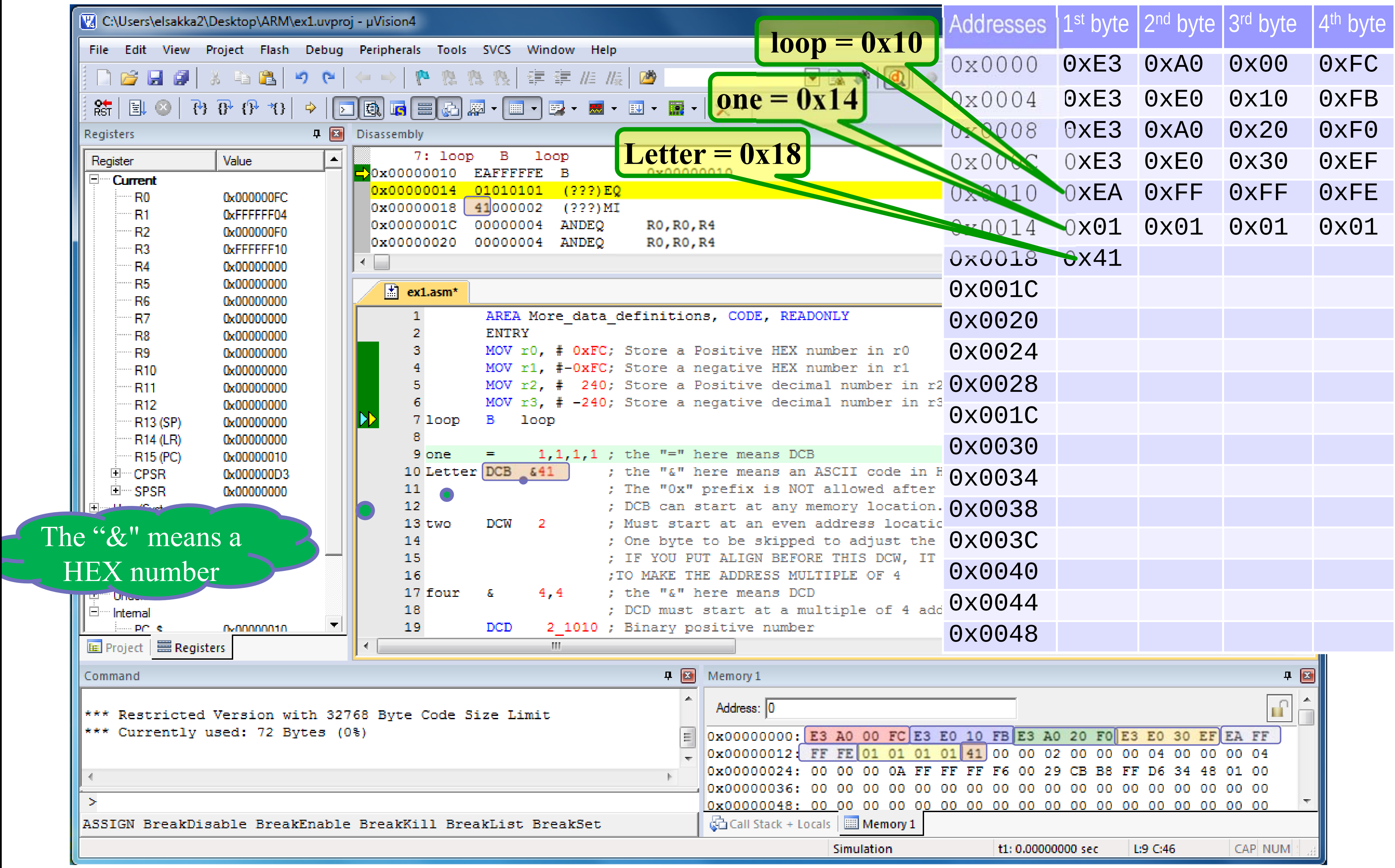
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0

0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation | t1: 0.00000000 sec | L:9 C:46 | CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018				
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives



loop = 0x10

one = 0x14

Letter = 0x18

The “&" means a HEX number

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
Internal	
PC $	0x00000010

```
     7: loop    B    loop
0x00000010  EAFFFFFE  B
0x00000014  01010101  (???)EQ
0x00000018  41000002  (???)MI
0x0000001C  00000004  ANDEQ     R0,R0,R4
0x00000020  00000004  ANDEQ     R0,R0,R4
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B    loop

one     =       1,1,1,1 ; the "=" here means DCB
Letter  DCB   &41       ; the "&" here means an ASCII code in H
                        ; The "0x" prefix is NOT allowed after
                        ; DCB can start at any memory location.
two     DCW    2        ; Must start at an even address locatic
                        ; One byte to be skipped to adjust the
                        ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                        ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4     ; the "&" here means DCD
                        ; DCD must start at a multiple of 4 add
        DCD      2_1010 ; Binary positive number
```

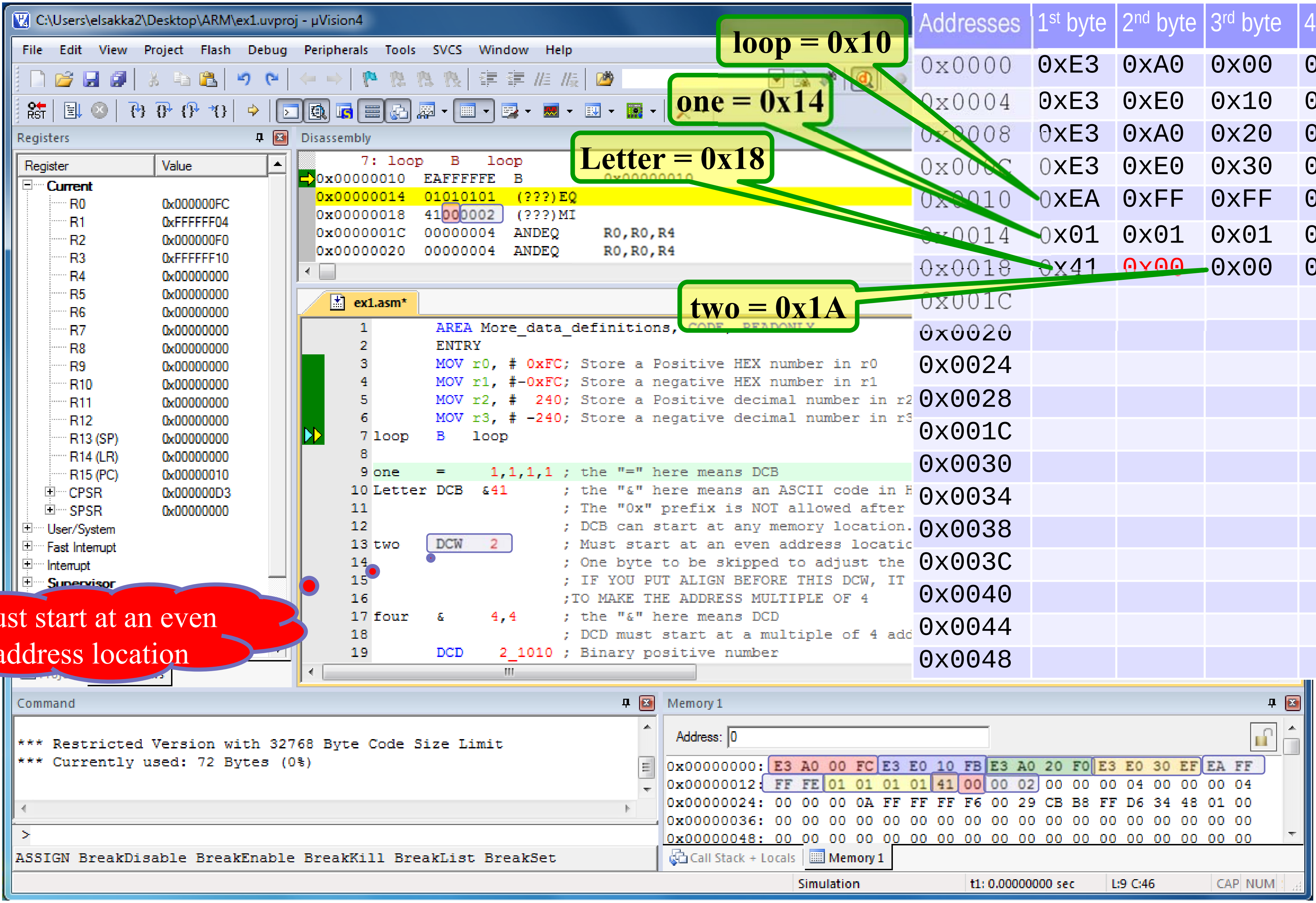
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41			
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Command
Memory 1
Call Stack + Locals
Memory 1
Simulation
t1: 0.00000000 sec
L:9 C:46
CAP NUM

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	

```
      7: loop    B    loop
0x00000010  EAFFFFFE  B      0x00000010
0x00000014  01010101  (???)EQ
0x00000018  41000002  (???)MI
0x0000001C  00000004  ANDEQ      R0,R0,R4
0x00000020  00000004  ANDEQ      R0,R0,R4
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B    loop

one     =      1,1,1,1 ; the "=" here means DCB
Letter  DCB   &41      ; the "&" here means an ASCII code in H
                       ; The "0x" prefix is NOT allowed after
                       ; DCB can start at any memory location.
two     DCW    2       ; Must start at an even address locatic
                       ; One byte to be skipped to adjust the
                       ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                       ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4     ; the "&" here means DCD
                       ; DCD must start at a multiple of 4 add
        DCD    2_1010  ; Binary positive number
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C				
0x0020				
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

loop = 0x10

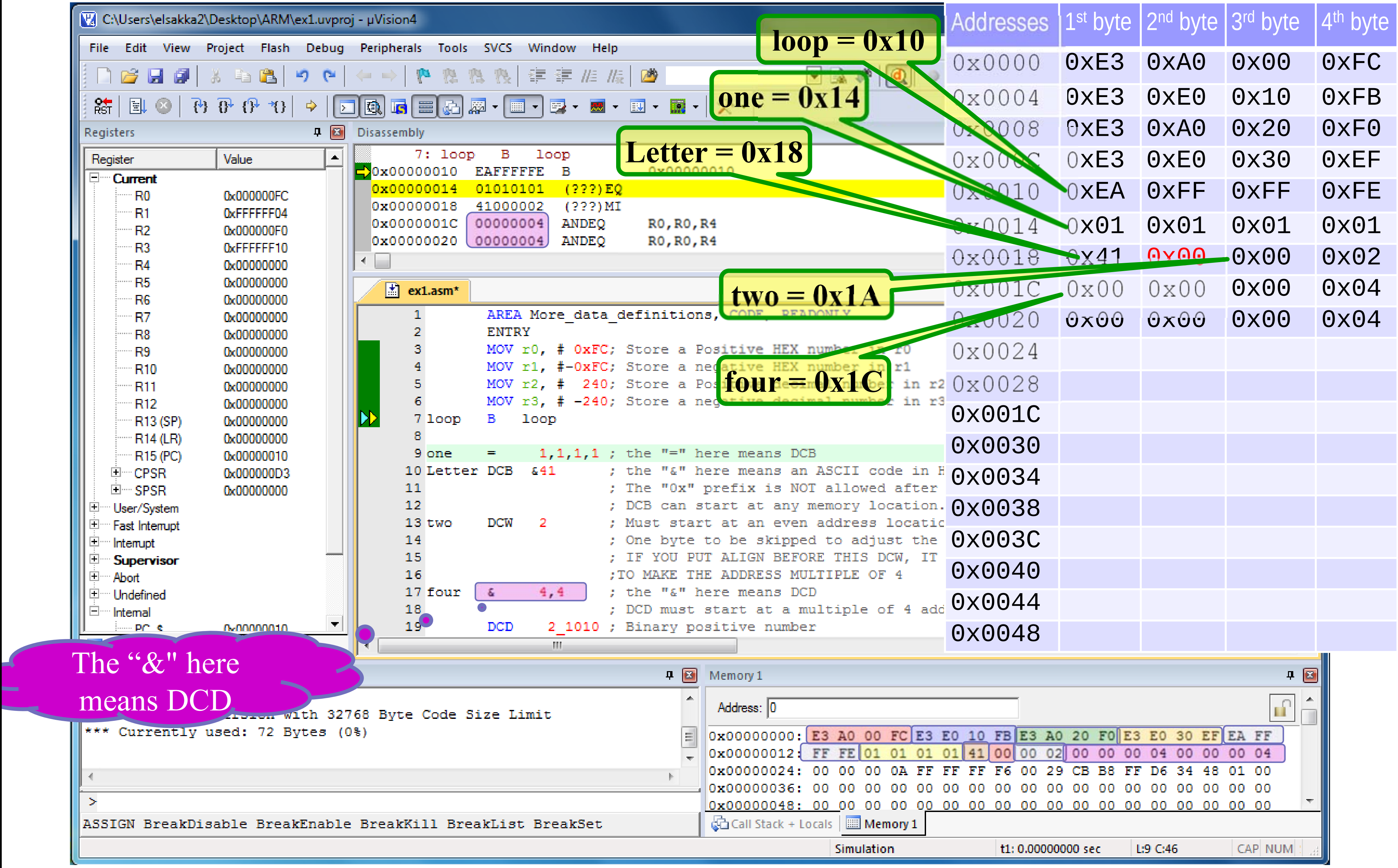
one = 0x14

Letter = 0x18

two = 0x1A

Must start at an even address location

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Memory 1
Call Stack + Locals
Memory 1
Simulation
t1: 0.00000000 sec
L:9 C:46
CAP NUM

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
     7: loop    B    loop
0x00000010  EAFFFFFE  B        0x00000010
0x00000014  01010101  (???)EQ
0x00000018  41000002  (???)MI
0x0000001C  00000004  ANDEQ     R0,R0,R4
0x00000020  00000004  ANDEQ     R0,R0,R4
```

```
        AREA More_data_definitions, CODE, READONLY
        ENTRY
        MOV r0, # 0xFC; Store a Positive HEX number in r0
        MOV r1, #-0xFC; Store a negative HEX number in r1
        MOV r2, #  240; Store a Positive decimal number in r2
        MOV r3, # -240; Store a negative decimal number in r3
loop    B    loop

one     =       1,1,1,1  ; the "=" here means DCB
Letter  DCB   &41        ; the "&" here means an ASCII code in H
                         ; The "0x" prefix is NOT allowed after
                         ; DCB can start at any memory location.
two     DCW    2         ; Must start at an even address locatic
                         ; One byte to be skipped to adjust the
                         ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                         ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4      ; the "&" here means DCD
                         ; DCD must start at a multiple of 4 add
        DCD    2_1010    ; Binary positive number
```

```
ision with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

loop = 0x10

one = 0x14

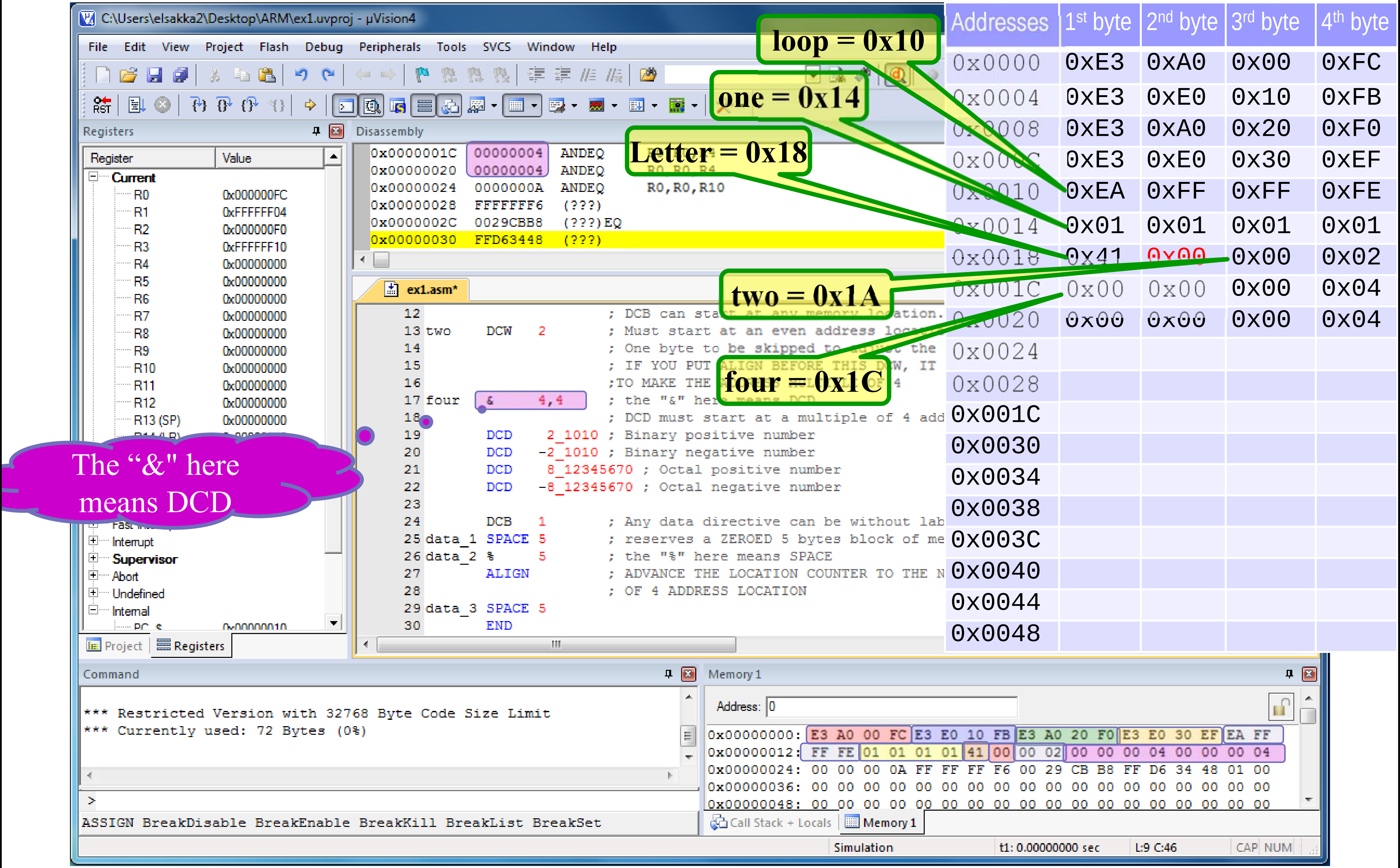
Letter = 0x18

two = 0x1A

four = 0x1C

The “&" here means DCD

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

The “&" here means DCD

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
Fast Int...	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000001C  00000004  ANDEQ    R0,R0,R4
0x00000020  00000004  ANDEQ    R0,R0,R4
0x00000024  0000000A  ANDEQ    R0,R0,R10
0x00000028  FFFFFFF6  (???)
0x0000002C  0029CBB8  (???)EQ
0x00000030  FFD63448  (???)
```

```
                                 ; DCB can start at any memory location.
two     DCW    2                 ; Must start at an even address loca
                                 ; One byte to be skipped to adjust the
                                 ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                                 ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4               ; the "&" here means DCD
                                 ; DCD must start at a multiple of 4 add
        DCD     2_1010 ; Binary positive number
        DCD    -2_1010 ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1                 ; Any data directive can be without lab
data_1  SPACE  5                 ; reserves a ZEROED 5 bytes block of me
data_2  %      5                 ; the "%" here means SPACE
        ALIGN                    ; ADVANCE THE LOCATION COUNTER TO THE N
                                 ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

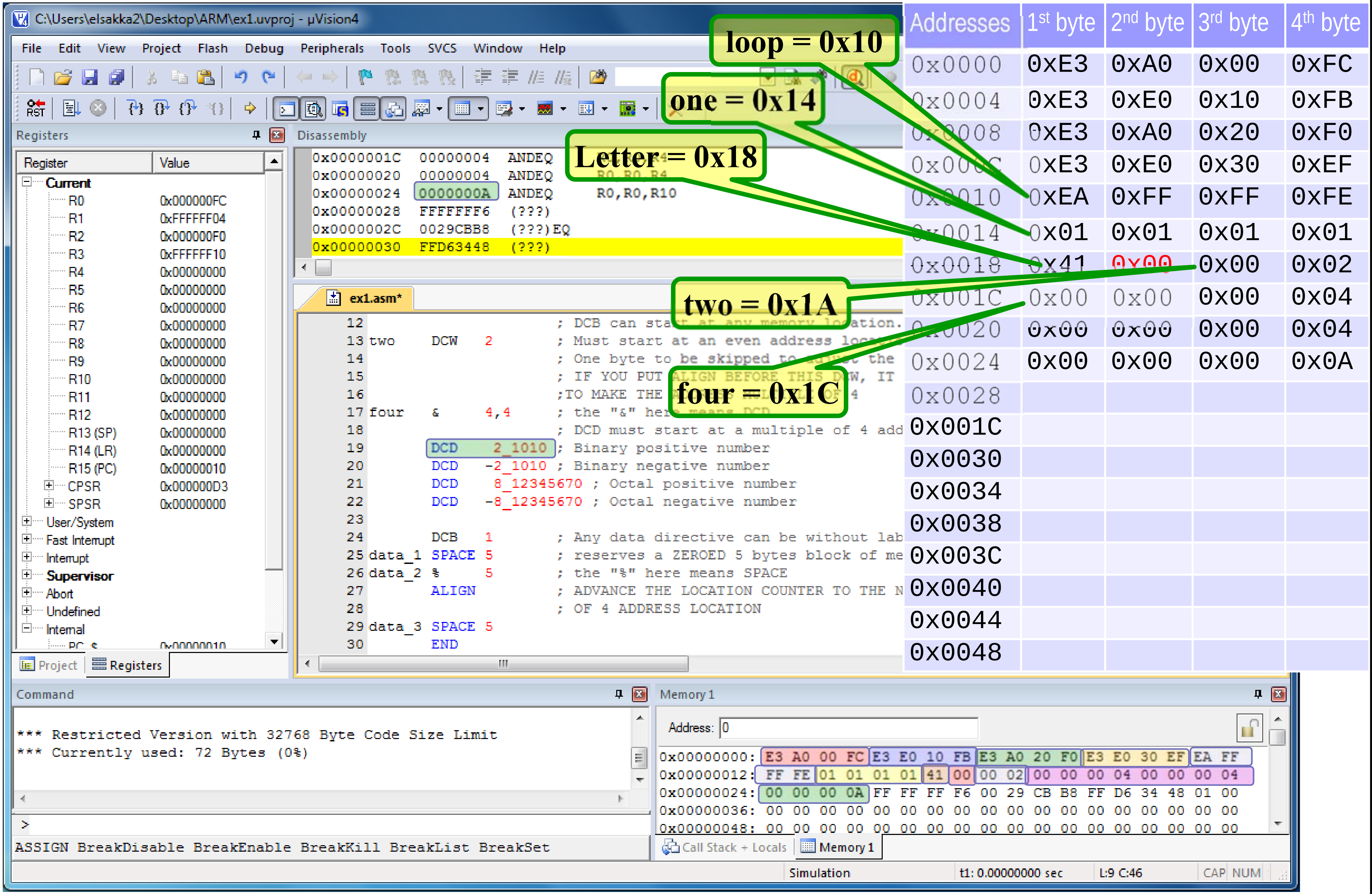
Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:9 C:46   CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024				
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000001C   00000004   ANDEQ
0x00000020   00000004   ANDEQ      R0,R0,R4
0x00000024   0000000A   ANDEQ      R0,R0,R10
0x00000028   FFFFFFF6   (???)
0x0000002C   0029CBB8   (???)EQ
0x00000030   FFD63448   (???)
```

```
                               ; DCB can start at any memory location.
two      DCW    2              ; Must start at an even address loca
                               ; One byte to be skipped to adjust the
                               ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                               ;TO MAKE THE ADDRESS MULTIPLE OF 4
four     &      4,4            ; the "&" here means DCD
                               ; DCD must start at a multiple of 4 add
         DCD     2_1010        ; Binary positive number
         DCD    -2_1010        ; Binary negative number
         DCD     8_12345670    ; Octal positive number
         DCD    -8_12345670    ; Octal negative number

         DCB    1              ; Any data directive can be without lab
data_1   SPACE  5              ; reserves a ZEROED 5 bytes block of me
data_2   %      5              ; the "%" here means SPACE
         ALIGN                 ; ADVANCE THE LOCATION COUNTER TO THE N
                               ; OF 4 ADDRESS LOCATION
data_3   SPACE  5
         END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:9 C:46   CAP NUM

loop = 0x10

one = 0x14

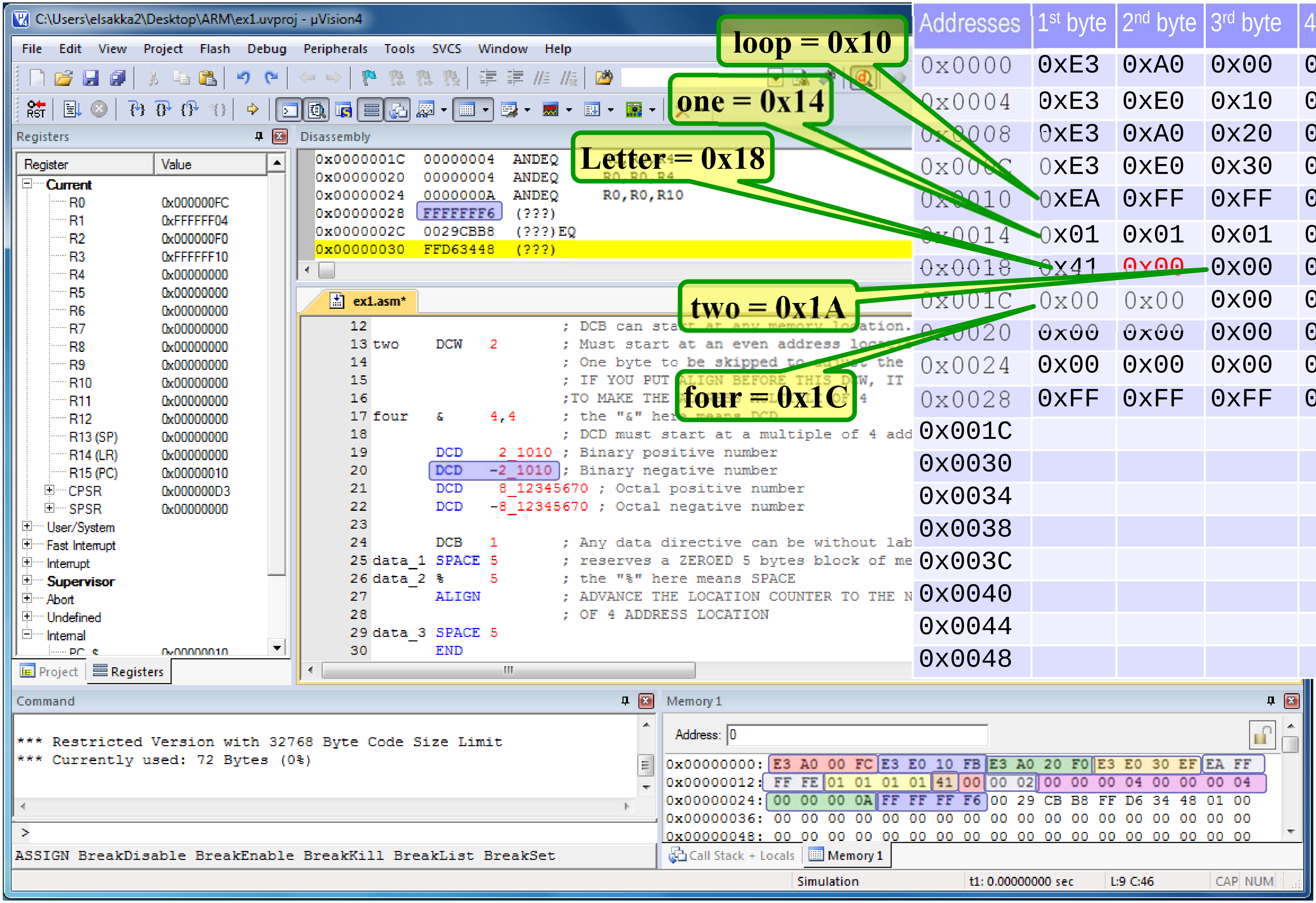
Letter = 0x18

two = 0x1A

four = 0x1C

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028				
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000001C  00000004  ANDEQ      R0,R0,R4
0x00000020  00000004  ANDEQ      R0,R0,R4
0x00000024  0000000A  ANDEQ      R0,R0,R10
0x00000028  FFFFFFF6  (???)
0x0000002C  0029CBB8  (???)EQ
0x00000030  FFD63448  (???)
```

```
two     DCW    2          ; DCB can start at any memory location.
                          ; Must start at an even address loca
                          ; One byte to be skipped to adjust the
                          ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                          ;TO MAKE THE                    OF 4
four    &      4,4        ; the "&" here means DCD
                          ; DCD must start at a multiple of 4 add
        DCD     2_1010    ; Binary positive number
        DCD    -2_1010    ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1          ; Any data directive can be without lab
data_1  SPACE  5          ; reserves a ZEROED 5 bytes block of me
data_2  %      5          ; the "%" here means SPACE
        ALIGN             ; ADVANCE THE LOCATION COUNTER TO THE N
                          ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Simulation   t1: 0.00000000 sec   L:9 C:46   CAP NUM
```

loop = 0x10

one = 0x14

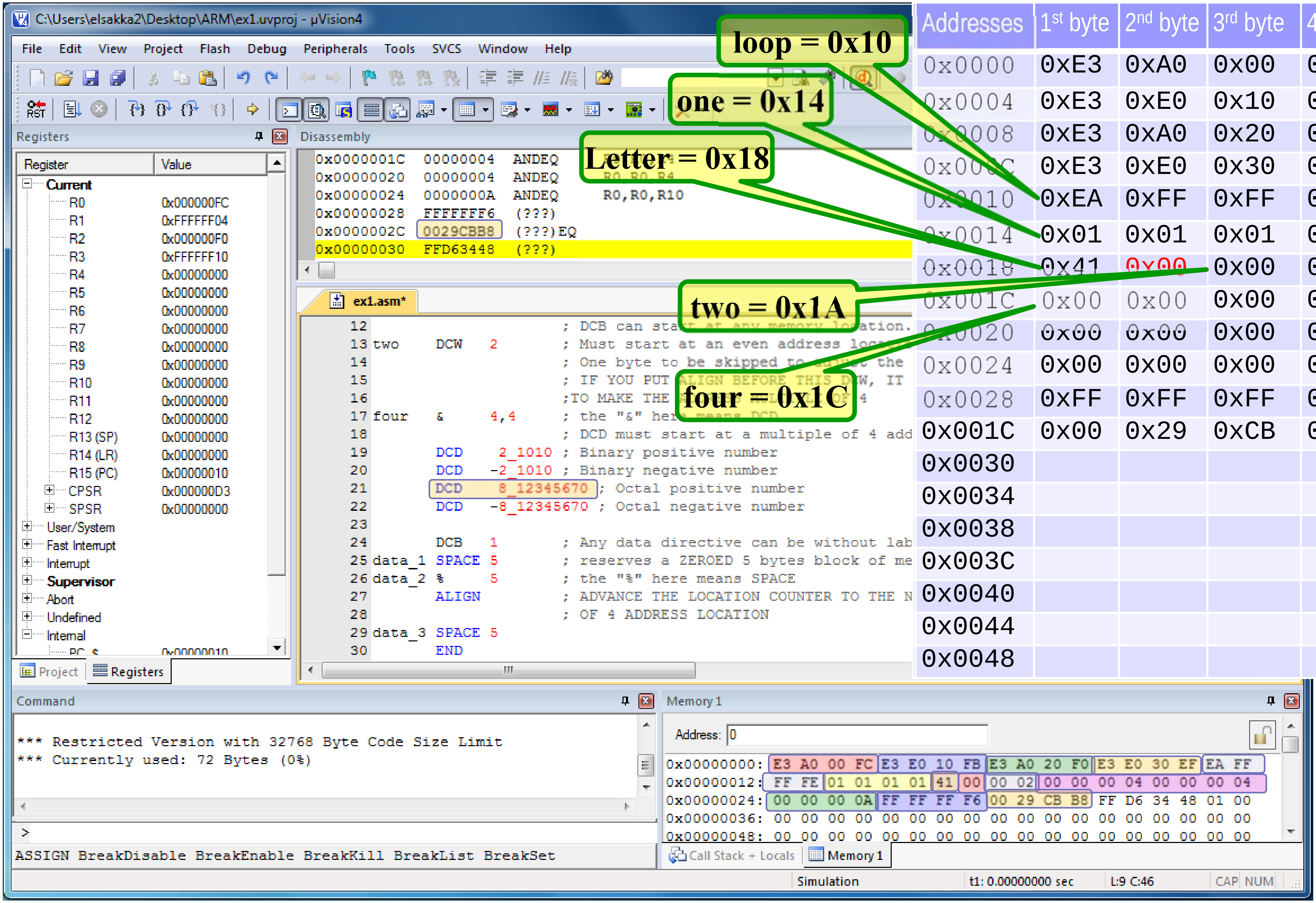
Letter = 0x18

two = 0x1A

four = 0x1C

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C				
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives



Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000001C   00000004   ANDEQ
0x00000020   00000004   ANDEQ        R0,R0,R4
0x00000024   0000000A   ANDEQ        R0,R0,R10
0x00000028   FFFFFFF6   (???)
0x0000002C   0029CBB8   (???)EQ
0x00000030   FFD63448   (???)
```

```
                           ; DCB can start at any memory location.
two     DCW    2           ; Must start at an even address loca
                           ; One byte to be skipped to adjust the
                           ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                           ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4         ; the "&" here means DCD
                           ; DCD must start at a multiple of 4 add
        DCD     2_1010     ; Binary positive number
        DCD    -2_1010     ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1           ; Any data directive can be without lab
data_1  SPACE  5           ; reserves a ZEROED 5 bytes block of me
data_2  %      5           ; the "%" here means SPACE
        ALIGN              ; ADVANCE THE LOCATION COUNTER TO THE N
                           ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030				
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

loop = 0x10

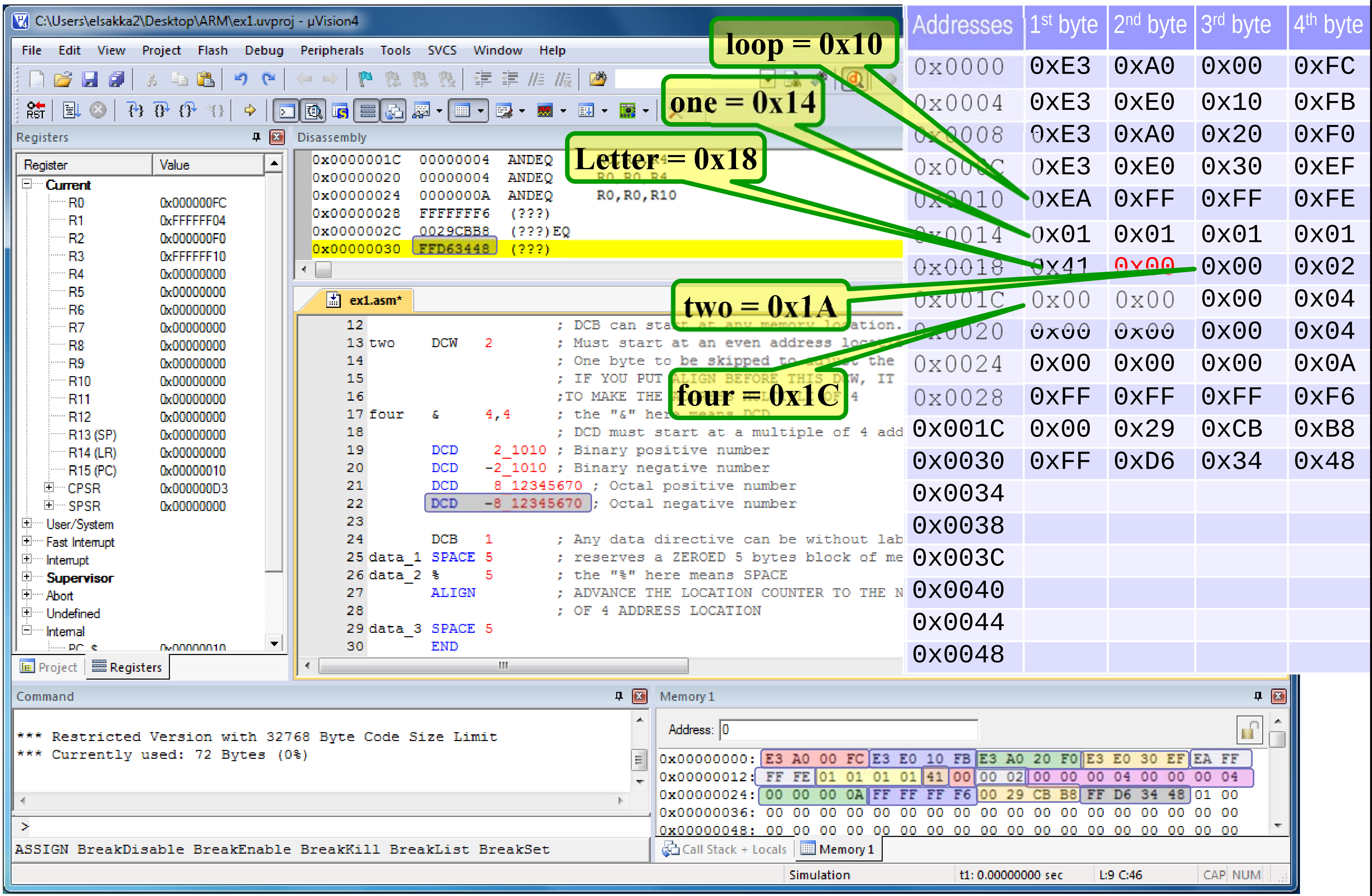
one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
RST
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000001C   00000004   ANDEQ
0x00000020   00000004   ANDEQ      R0,R0,R4
0x00000024   0000000A   ANDEQ      R0,R0,R10
0x00000028   FFFFFFF6   (???)
0x0000002C   0029CBB8   (???)EQ
0x00000030   FFD63448   (???)
```

```
                        ; DCB can sta                  ation.
two     DCW    2        ; Must start at an even address loca
                        ; One byte to be skipped to          the
                        ; IF YOU PUT ALIGN BEFORE THIS D W, IT
                        ;TO MAKE THE                    OF 4
four    &      4,4      ; the "&" her
                        ; DCD must start at a multiple of 4 add
        DCD     2_1010  ; Binary positive number
        DCD    -2_1010  ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1        ; Any data directive can be without lab
data_1  SPACE  5        ; reserves a ZEROED 5 bytes block of me
data_2  %      5        ; the "%" here means SPACE
        ALIGN           ; ADVANCE THE LOCATION COUNTER TO THE N
                        ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

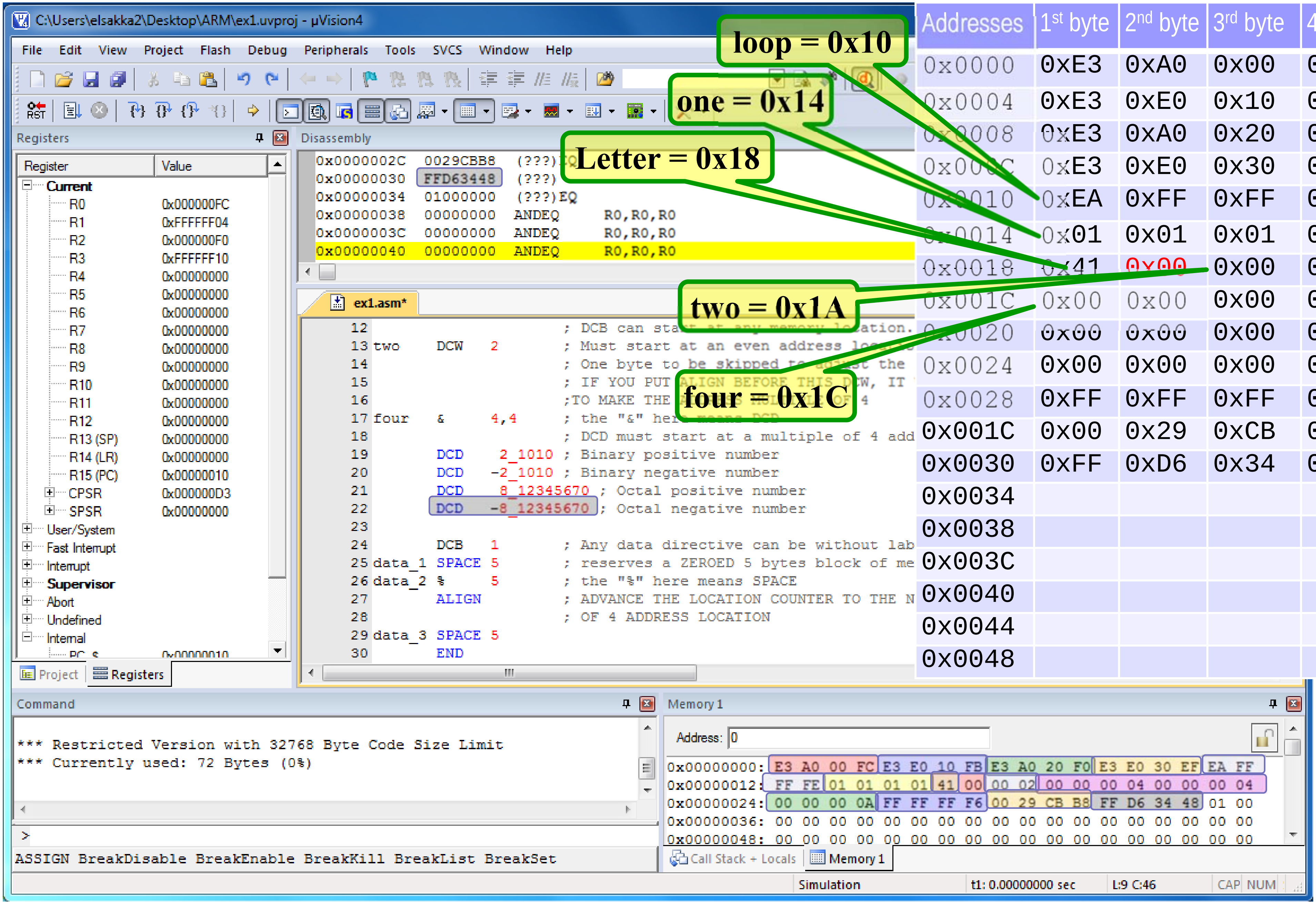
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation t1: 0.00000000 sec L:9 C:46 CAP NUM

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000002C  0029CBB8  (???)EQ
0x00000030  FFD63448  (???)
0x00000034  01000000  (???)EQ
0x00000038  00000000  ANDEQ    R0,R0,R0
0x0000003C  00000000  ANDEQ    R0,R0,R0
0x00000040  00000000  ANDEQ    R0,R0,R0
```

```
                                ; DCB can start at any memory location.
two     DCW     2               ; Must start at an even address lo
                                ; One byte to be skipped to adjust the
                                ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                                ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &       4,4             ; the "&" here means DCD
                                ; DCD must start at a multiple of 4 add
        DCD      2_1010 ; Binary positive number
        DCD     -2_1010 ; Binary negative number
        DCD      8_12345670 ; Octal positive number
        DCD     -8_12345670 ; Octal negative number

        DCB     1               ; Any data directive can be without lab
data_1  SPACE   5               ; reserves a ZEROED 5 bytes block of me
data_2  %       5               ; the "%" here means SPACE
        ALIGN                   ; ADVANCE THE LOCATION COUNTER TO THE N
                                ; OF 4 ADDRESS LOCATION
data_3  SPACE   5
        END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

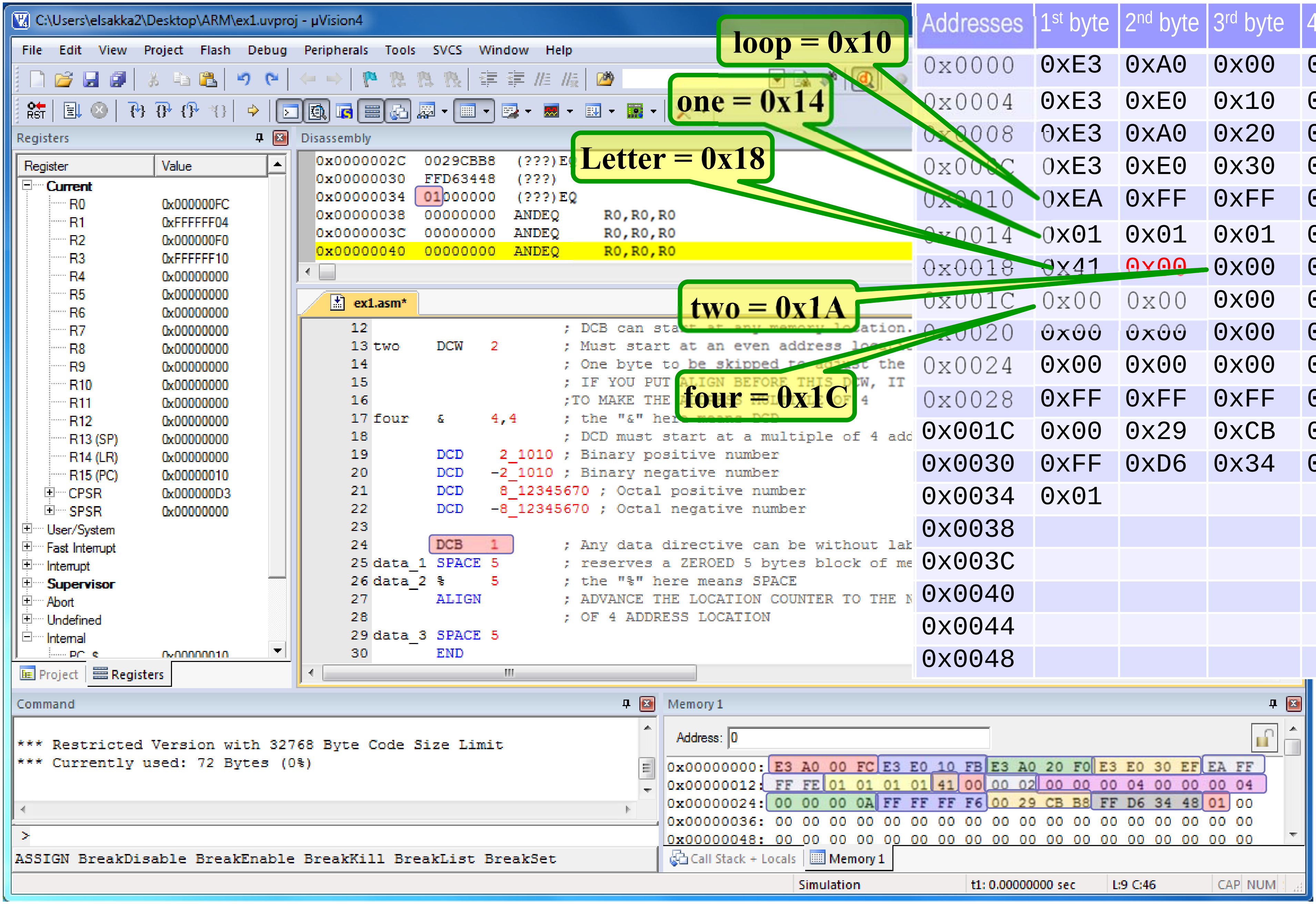
```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation | t1: 0.00000000 sec | L:9 C:46 | CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034				
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives



```
Register        Value
Current
  R0            0x000000FC
  R1            0xFFFFFF04
  R2            0x000000F0
  R3            0xFFFFFF10
  R4            0x00000000
  R5            0x00000000
  R6            0x00000000
  R7            0x00000000
  R8            0x00000000
  R9            0x00000000
  R10           0x00000000
  R11           0x00000000
  R12           0x00000000
  R13 (SP)      0x00000000
  R14 (LR)      0x00000000
  R15 (PC)      0x00000010
  CPSR          0x000000D3
  SPSR          0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC $          0x00000010
```

```
0x0000002C  0029CBB8   (???)EQ
0x00000030  FFD63448   (???)
0x00000034  01000000   (???)EQ
0x00000038  00000000   ANDEQ      R0,R0,R0
0x0000003C  00000000   ANDEQ      R0,R0,R0
0x00000040  00000000   ANDEQ      R0,R0,R0
```

```
                              ; DCB can start at any memory location.
two     DCW    2              ; Must start at an even address location
                              ; One byte to be skipped to adjust the
                              ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                              ;TO MAKE THE ADDRESS MULTIPLE OF 4
four    &      4,4            ; the "&" here means DCD
                              ; DCD must start at a multiple of 4 add
        DCD     2_1010 ; Binary positive number
        DCD    -2_1010 ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1              ; Any data directive can be without lak
data_1  SPACE  5              ; reserves a ZEROED 5 bytes block of me
data_2  %      5              ; the "%" here means SPACE
        ALIGN                 ; ADVANCE THE LOCATION COUNTER TO THE N
                              ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

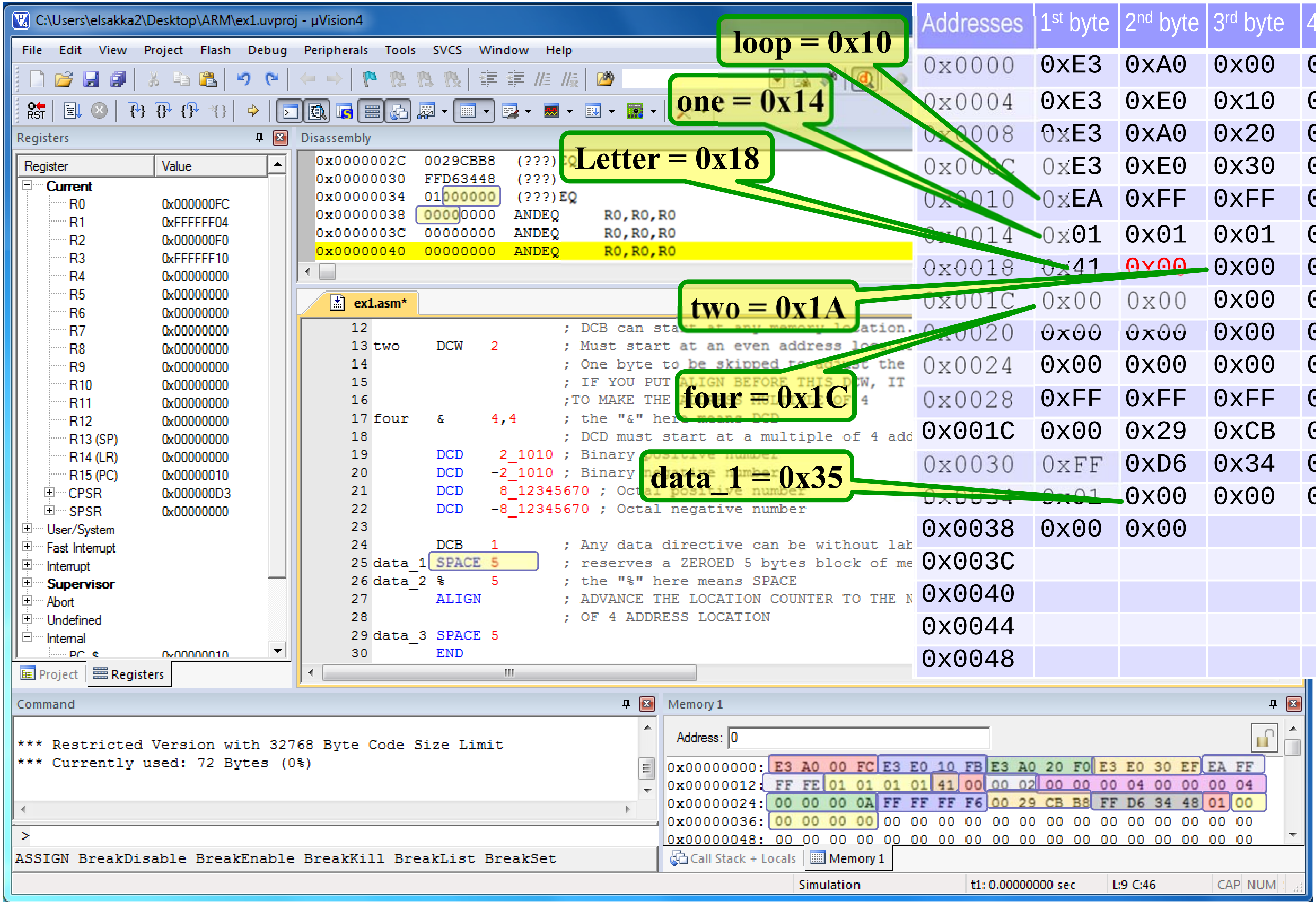
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Simulation    t1: 0.00000000 sec    L:9 C:46    CAP NUM
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034	0x01			
0x0038				
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Project
Registers
Command
Memory 1
Call Stack + Locals

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

data_1 = 0x35

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000002C   0029CBB8   (???)EQ
0x00000030   FFD63448   (???)
0x00000034   01000000   (???)EQ
0x00000038   00000000   ANDEQ      R0,R0,R0
0x0000003C   00000000   ANDEQ      R0,R0,R0
0x00000040   00000000   ANDEQ      R0,R0,R0
```

```
                                  ; DCB can sta
two      DCW    2                 ; Must start at an even address lo
                                  ; One byte to be skipped to         the
                                  ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                                  ;TO MAKE THE                     OF 4
four     &      4,4               ; the "&" here means DCD
                                  ; DCD must start at a multiple of 4 add
         DCD     2_1010 ; Binary positive number
         DCD    -2_1010 ; Binary negative number
         DCD     8_12345670 ; Octal positive number
         DCD    -8_12345670 ; Octal negative number

         DCB    1                 ; Any data directive can be without lab
data_1   SPACE  5                 ; reserves a ZEROED 5 bytes block of me
data_2   %      5                 ; the "%" here means SPACE
         ALIGN                    ; ADVANCE THE LOCATION COUNTER TO THE N
                                  ; OF 4 ADDRESS LOCATION
data_3   SPACE  5
         END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

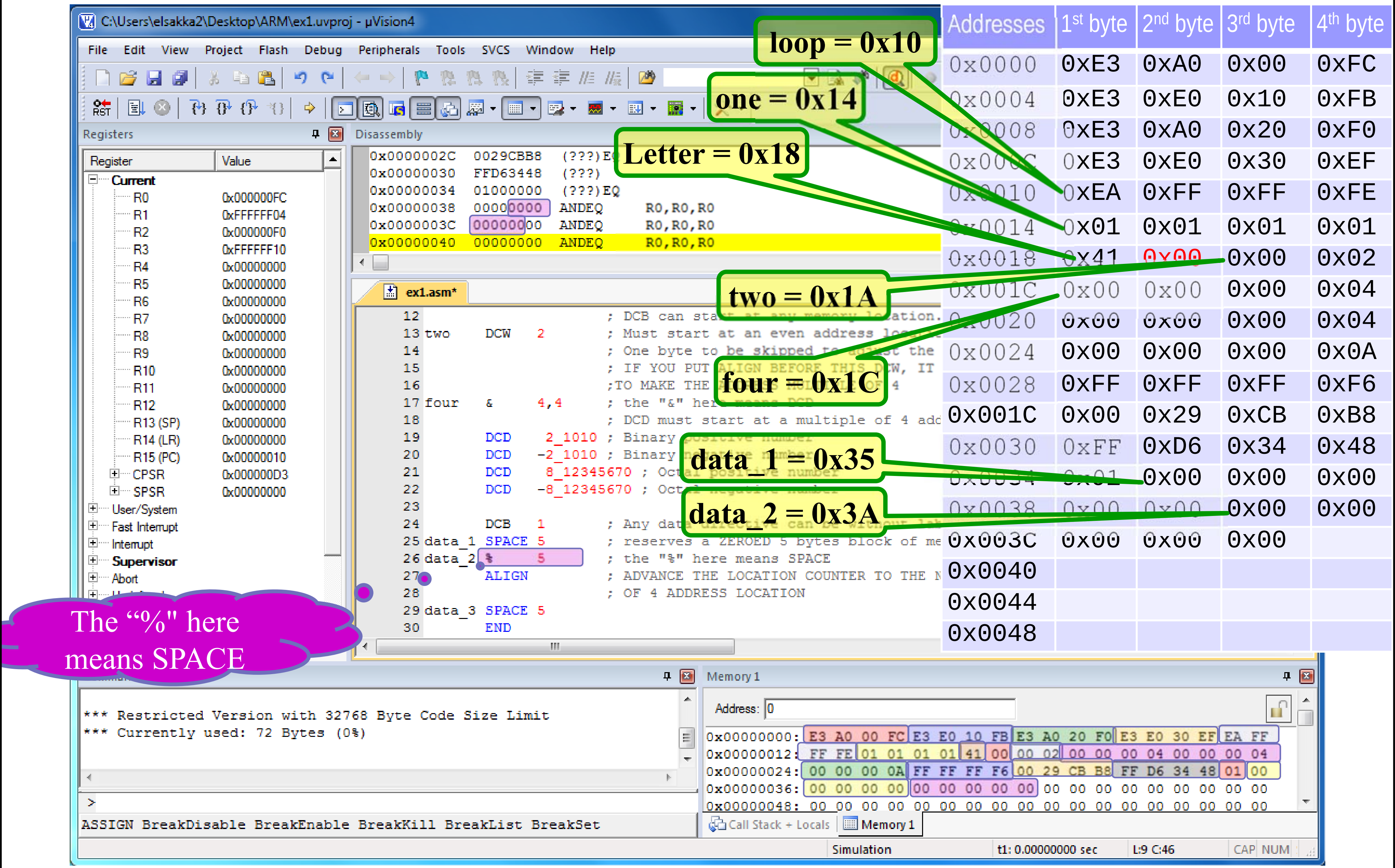
Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:9 C:46   CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034	0x01	0x00	0x00	0x00
0x0038	0x00	0x00		
0x003C				
0x0040				
0x0044				
0x0048				

Data Definition Directives



C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Memory 1
Call Stack + Locals
Simulation
t1: 0.00000000 sec
L:9 C:46
CAP NUM

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	

```
0x0000002C  0029CBB8  (???)E
0x00000030  FFD63448  (???)
0x00000034  01000000  (???)EQ
0x00000038  00000000  ANDEQ     R0,R0,R0
0x0000003C  00000000  ANDEQ     R0,R0,R0
0x00000040  00000000  ANDEQ     R0,R0,R0
```

```
                                ; DCB can sta                 cation.
two     DCW     2               ; Must start at an even address lo
                                ; One byte to be skipped         the
                                ; IF YOU PUT                 W, IT
                                ;TO MAKE THE                   4
four    &       4,4             ; the "&" her
                                ; DCD must start at a multiple of 4 add
        DCD      2_1010 ; Binary
        DCD     -2_1010 ; Binary
        DCD      8_12345670 ; Oct
        DCD     -8_12345670 ; Oct

        DCB     1               ; Any dat
data_1  SPACE 5                 ; reserves a ZEROED 5 bytes block of me
data_2  %       5               ; the "%" here means SPACE
        ALIGN                   ; ADVANCE THE LOCATION COUNTER TO THE N
                                ; OF 4 ADDRESS LOCATION
data_3  SPACE 5
        END
```

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

data_1 = 0x35

data_2 = 0x3A

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034	0x01	0x00	0x00	0x00
0x0038	0x00	0x00	0x00	0x00
0x003C	0x00	0x00	0x00	
0x0040				
0x0044				
0x0048				

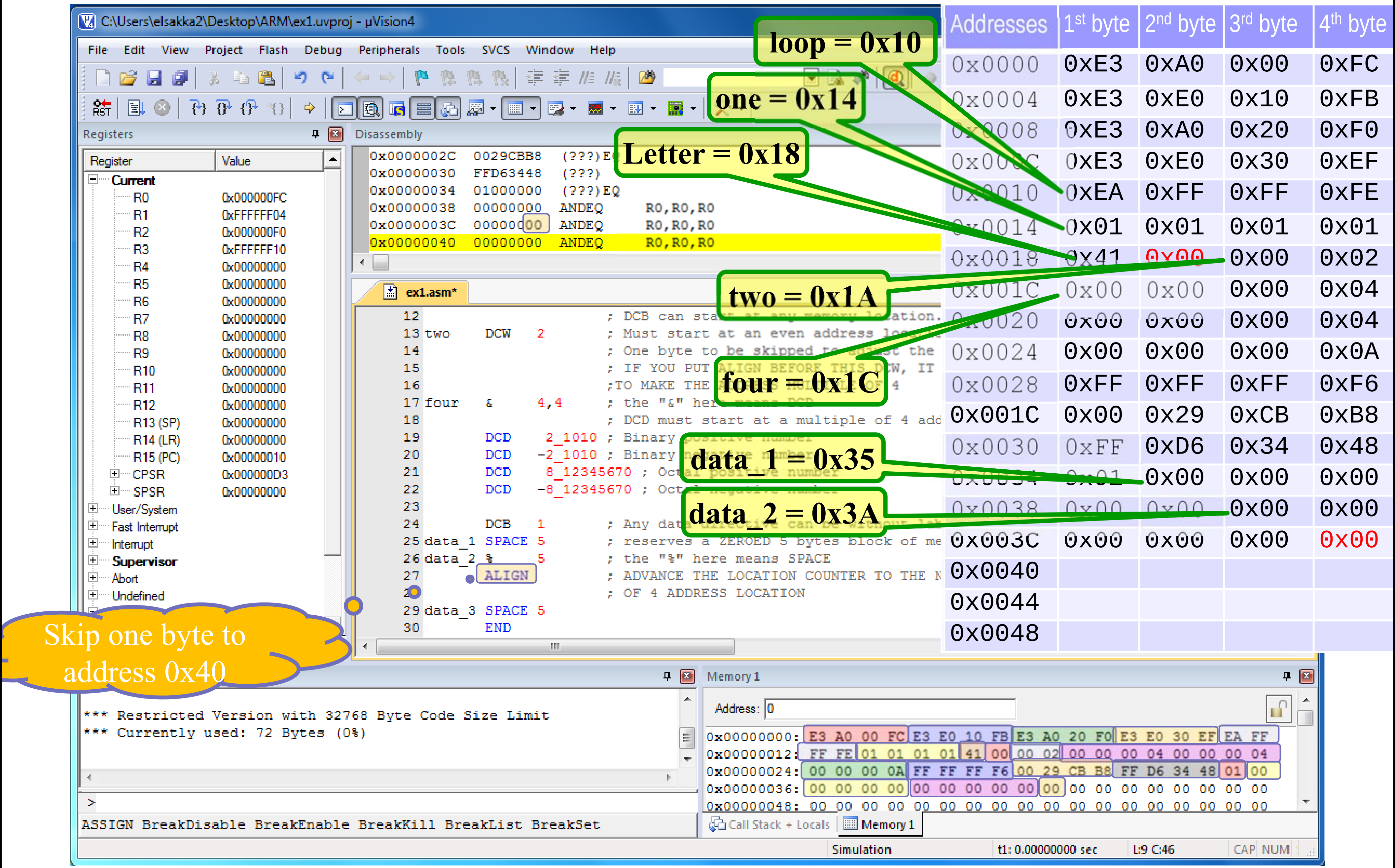
The "%" here means SPACE

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Data Definition Directives


C:\Users\elsakka2\Desktop\ARM\ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
ex1.asm*
Memory 1
Call Stack + Locals
Memory 1

Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	

```
0x0000002C  0029CBB8   (???)E
0x00000030  FFD63448   (???)
0x00000034  01000000   (???)EQ
0x00000038  00000000   ANDEQ      R0,R0,R0
0x0000003C  00000000   ANDEQ      R0,R0,R0
0x00000040  00000000   ANDEQ      R0,R0,R0
```

```
                                   ; DCB can sta                  ation.
two      DCW     2                 ; Must start at an even address lo
                                   ; One byte to be skipped t          the
                                   ; IF YOU PUT ALIGN BEFORE THIS DCW, IT
                                   ;TO MAKE THE                   F 4
four     &       4,4               ; the "&" her
                                   ; DCD must start at a multiple of 4 add
         DCD      2_1010 ; Binary
         DCD     -2_1010 ; Binary
         DCD      8_12345670 ; Oct
         DCD     -8_12345670 ; Oct

         DCB     1                 ; Any dat
data_1   SPACE   5                 ; reserves a ZEROED 5 bytes block of me
data_2   %       5                 ; the "%" here means SPACE
         ALIGN                     ; ADVANCE THE LOCATION COUNTER TO THE N
                                   ; OF 4 ADDRESS LOCATION
data_3   SPACE   5
         END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
Address: 0
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation | t1: 0.00000000 sec | L:9 C:46 | CAP NUM

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034	0x01	0x00	0x00	0x00
0x0038	0x00	0x00	0x00	0x00
0x003C	0x00	0x00	0x00	0x00
0x0040				
0x0044				
0x0048				

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

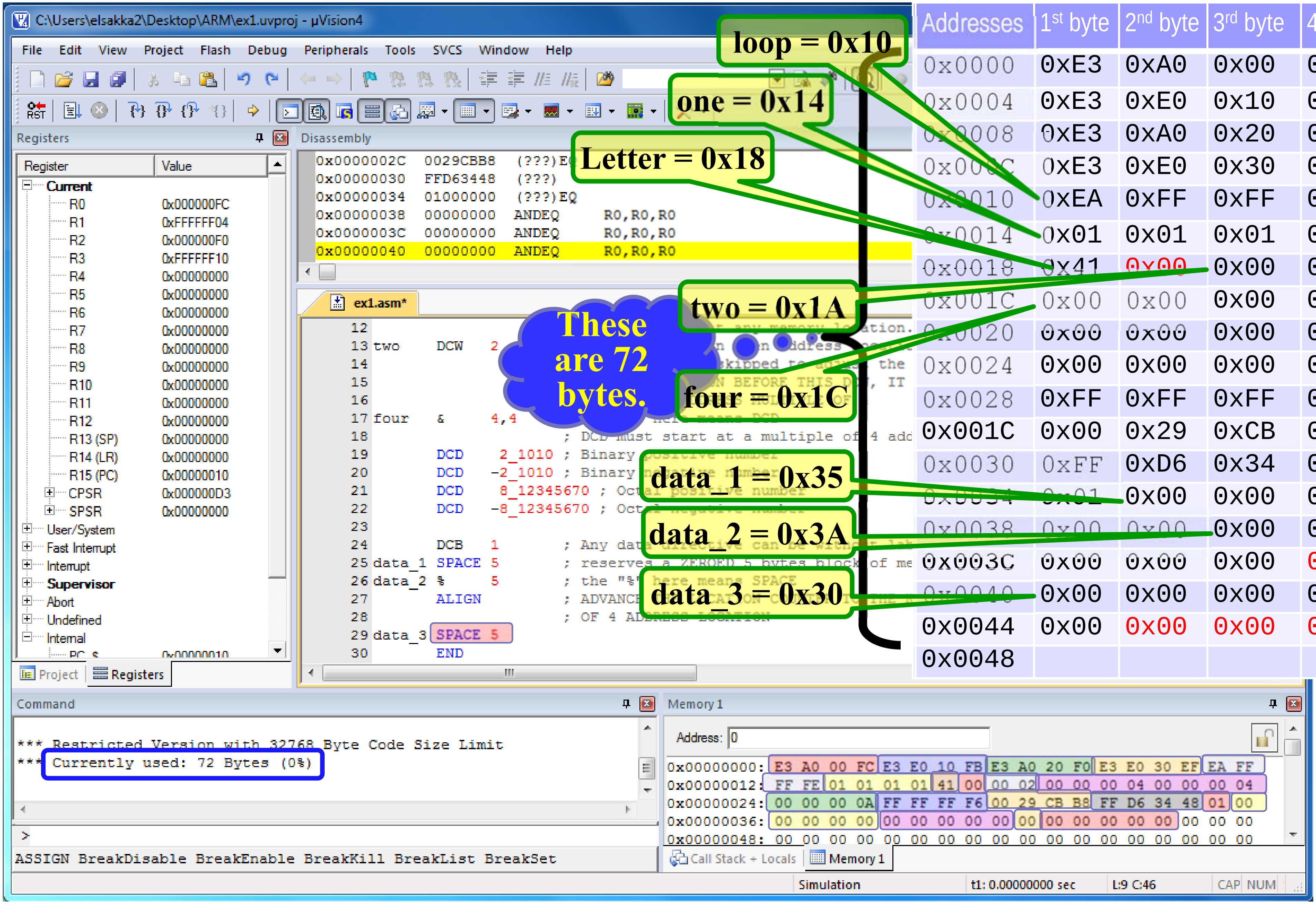
four = 0x1C

data_1 = 0x35

data_2 = 0x3A

Skip one byte to address 0x40

Data Definition Directives



Register	Value
Current	
R0	0x000000FC
R1	0xFFFFFF04
R2	0x000000F0
R3	0xFFFFFF10
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000010

```
0x0000002C  0029CBB8  (???)EQ
0x00000030  FFD63448  (???)
0x00000034  01000000  (???)EQ
0x00000038  00000000  ANDEQ     R0,R0,R0
0x0000003C  00000000  ANDEQ     R0,R0,R0
0x00000040  00000000  ANDEQ     R0,R0,R0
```

```
two     DCW    2              ; ... any memory location.
                              ; ... an address ...
                              ; ... skipped to ... the
                              ; ...N BEFORE THIS D..., IT
four    &      4,4            ; ... HOLDS ... OF
                              ; ... here means DCD
                              ; DCD must start at a multiple of 4 add
        DCD     2_1010 ; Binary positive number
        DCD    -2_1010 ; Binary negative number
        DCD     8_12345670 ; Octal positive number
        DCD    -8_12345670 ; Octal negative number

        DCB    1              ; Any data directive can be without lab
data_1  SPACE  5              ; reserves a ZEROED 5 bytes block of me
data_2  %      5              ; the "%" here means SPACE
        ALIGN                 ; ADVANCE ... LOCATION ... THE ...
                              ; OF 4 ADDRESS LOCATION
data_3  SPACE  5
        END
```

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 72 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000000: E3 A0 00 FC E3 E0 10 FB E3 A0 20 F0 E3 E0 30 EF EA FF
0x00000012: FF FE 01 01 01 01 41 00 00 02 00 00 00 04 00 00 00 04
0x00000024: 00 00 00 0A FF FF FF F6 00 29 CB B8 FF D6 34 48 01 00
0x00000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Addresses	1st byte	2nd byte	3rd byte	4th byte
0x0000	0xE3	0xA0	0x00	0xFC
0x0004	0xE3	0xE0	0x10	0xFB
0x0008	0xE3	0xA0	0x20	0xF0
0x000C	0xE3	0xE0	0x30	0xEF
0x0010	0xEA	0xFF	0xFF	0xFE
0x0014	0x01	0x01	0x01	0x01
0x0018	0x41	0x00	0x00	0x02
0x001C	0x00	0x00	0x00	0x04
0x0020	0x00	0x00	0x00	0x04
0x0024	0x00	0x00	0x00	0x0A
0x0028	0xFF	0xFF	0xFF	0xF6
0x001C	0x00	0x29	0xCB	0xB8
0x0030	0xFF	0xD6	0x34	0x48
0x0034	0x01	0x00	0x00	0x00
0x0038	0x00	0x00	0x00	0x00
0x003C	0x00	0x00	0x00	0x00
0x0040	0x00	0x00	0x00	0x00
0x0044	0x00	0x00	0x00	0x00
0x0048				

loop = 0x10

one = 0x14

Letter = 0x18

two = 0x1A

four = 0x1C

data_1 = 0x35

data_2 = 0x3A

data_3 = 0x30

These are 72 bytes.