Tutorial 09: ARM Pseudo Instructions

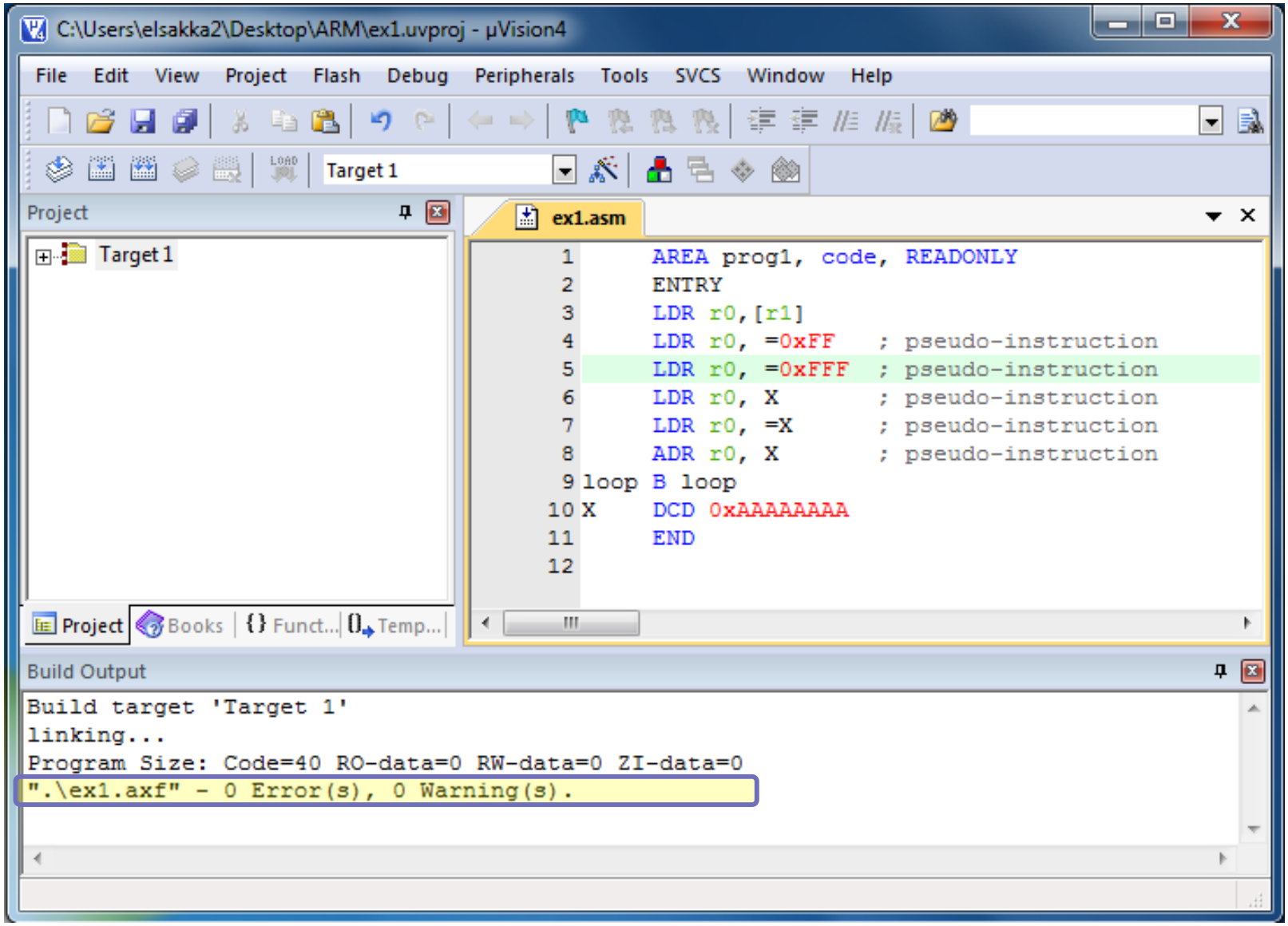
Computer Science Department
CS2208: Introduction to Computer Organization and Architecture
Winter 2021-2022
Instructor: Mahmoud R. El-Sakka
Office: MC-419
Email: elsakka@csd.uwo.ca
Phone: 519-661-2111 x86996

ARM pseudo-instructions

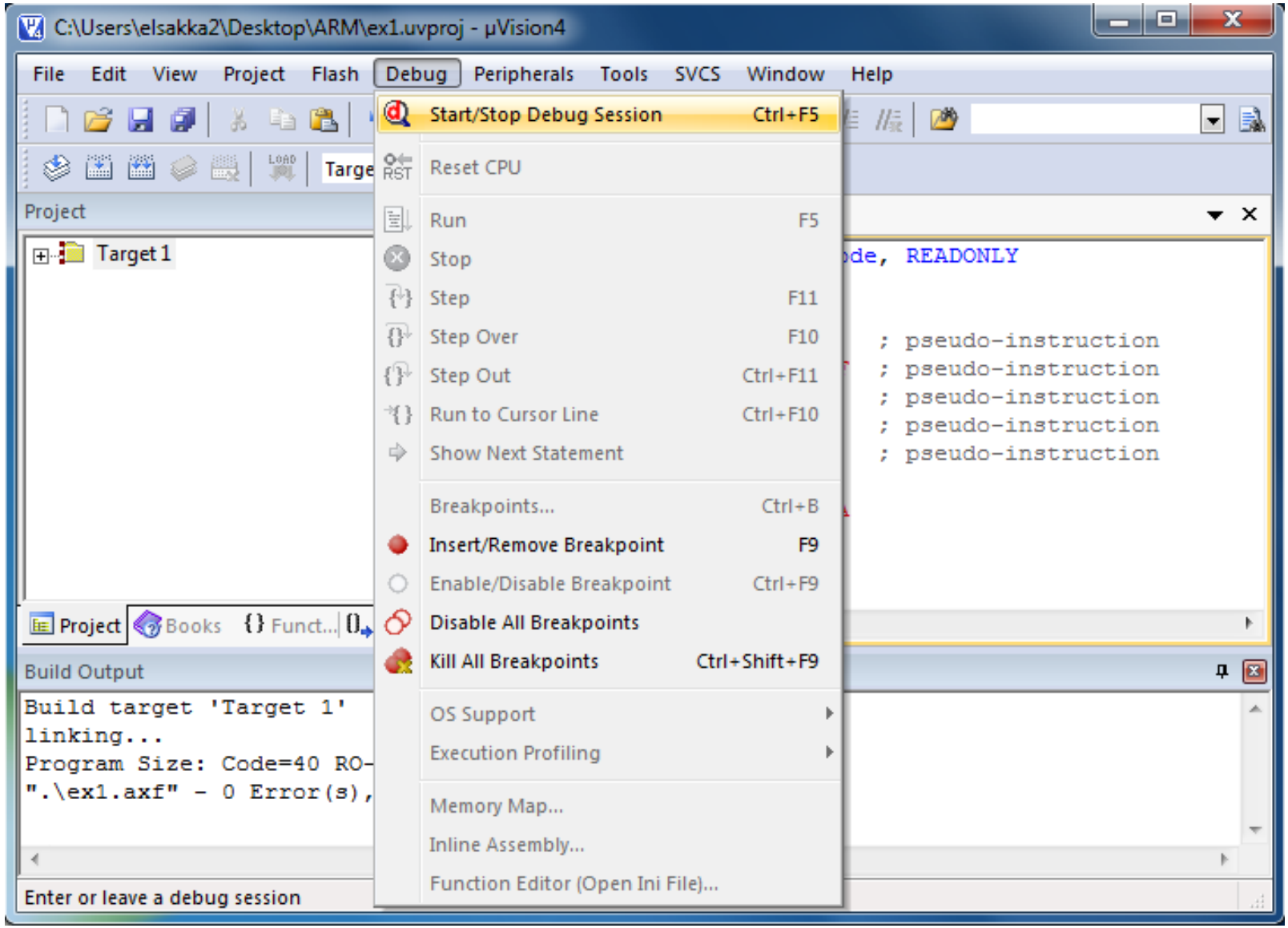
- The ARM assembler supports several *pseudo instructions* that are translated into the appropriate combination of ARM words at assembly time.
- Consider the following assembly program:

```
      AREA prog1, code, READONLY
      ENTRY
      LDR r0,[r1]
      LDR r0, =0xFF    ; pseudo-instruction
      LDR r0, =0xFFF   ; pseudo-instruction
      LDR r0, X        ; pseudo-instruction
      LDR r0, =X       ; pseudo-instruction
      ADR r0, X        ; pseudo-instruction
loop  B loop
X     DCD 0xAAAAAAAA
      END
```

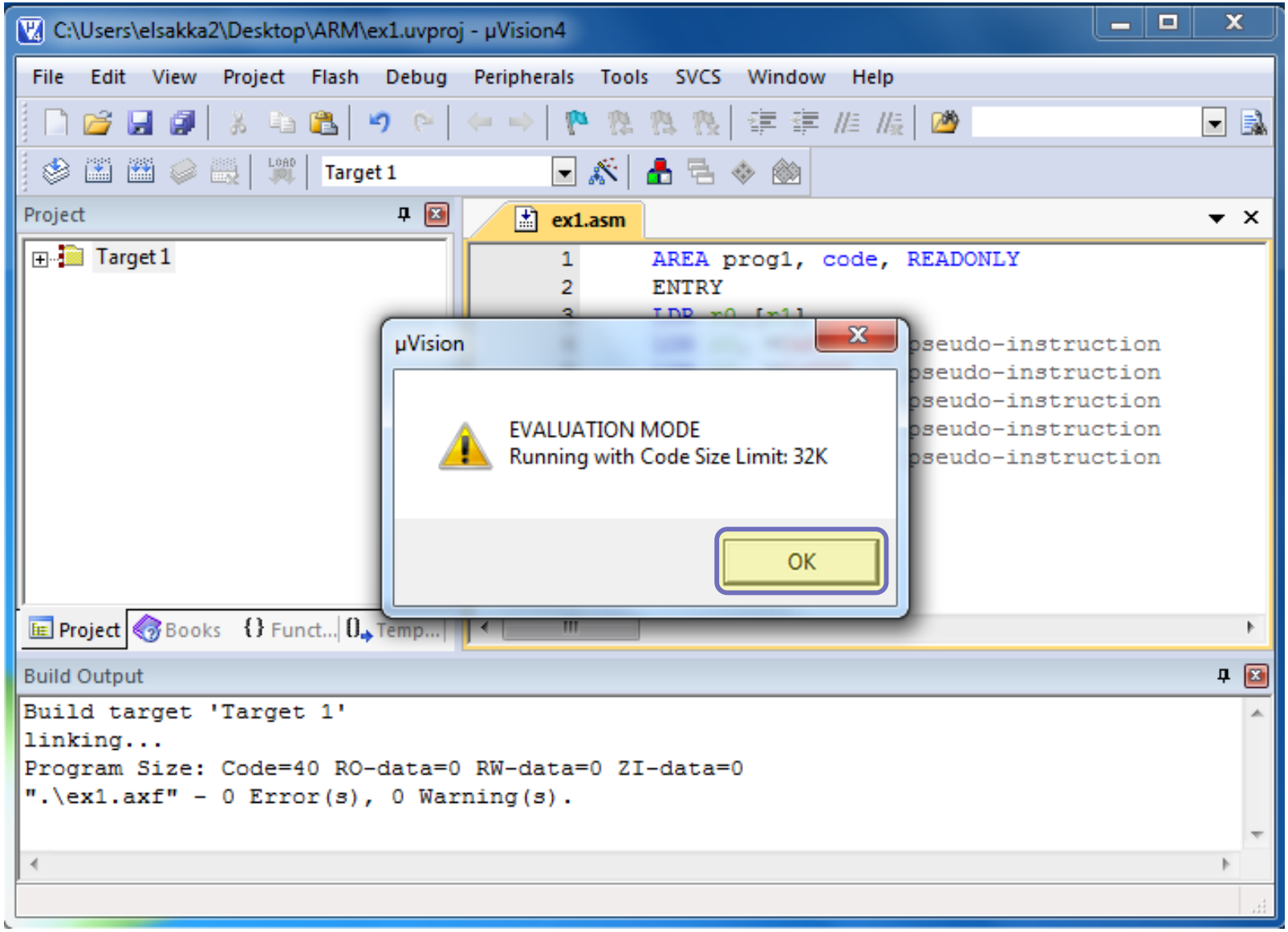
ARM pseudo-instructions



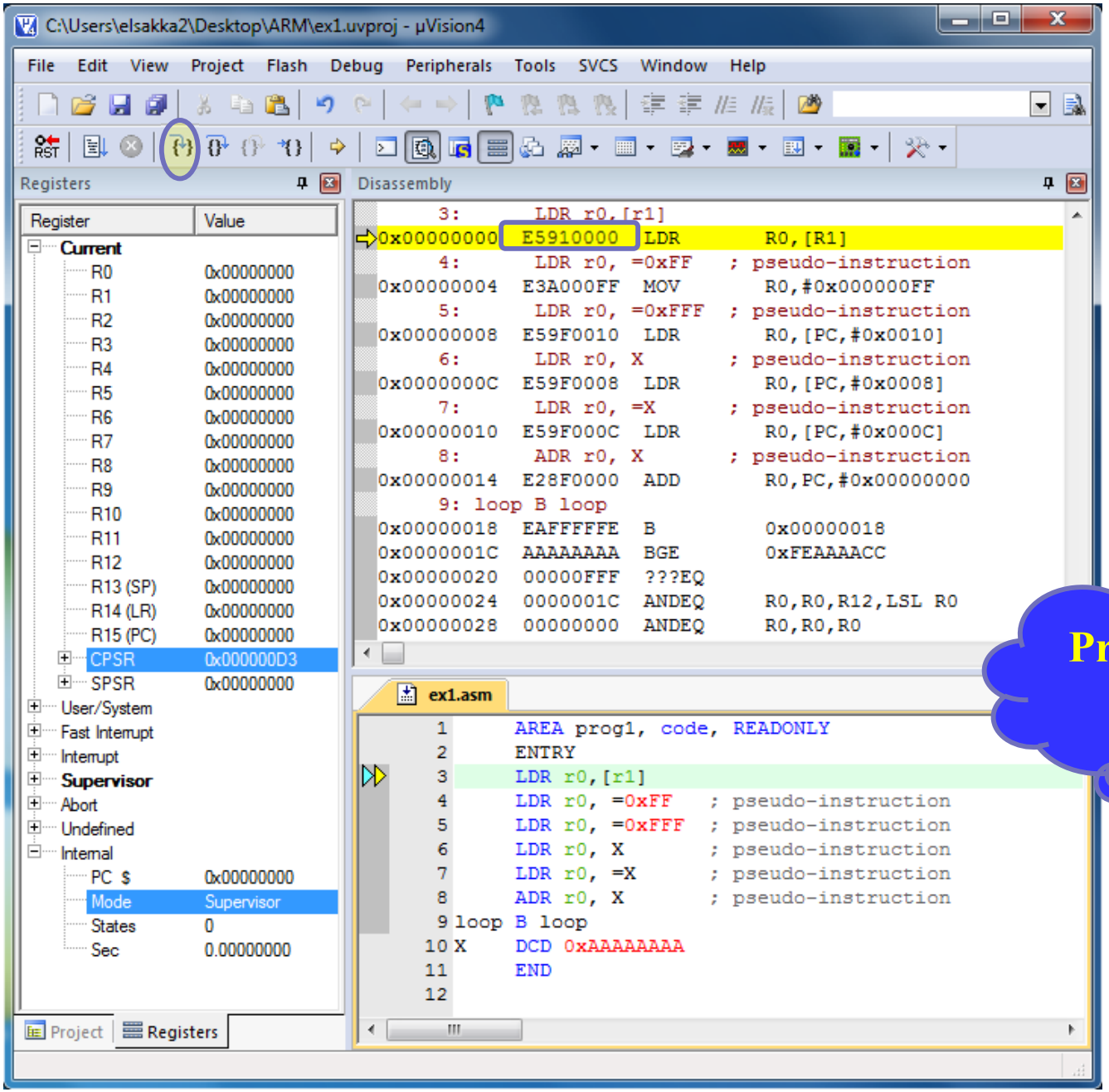
ARM pseudo-instructions



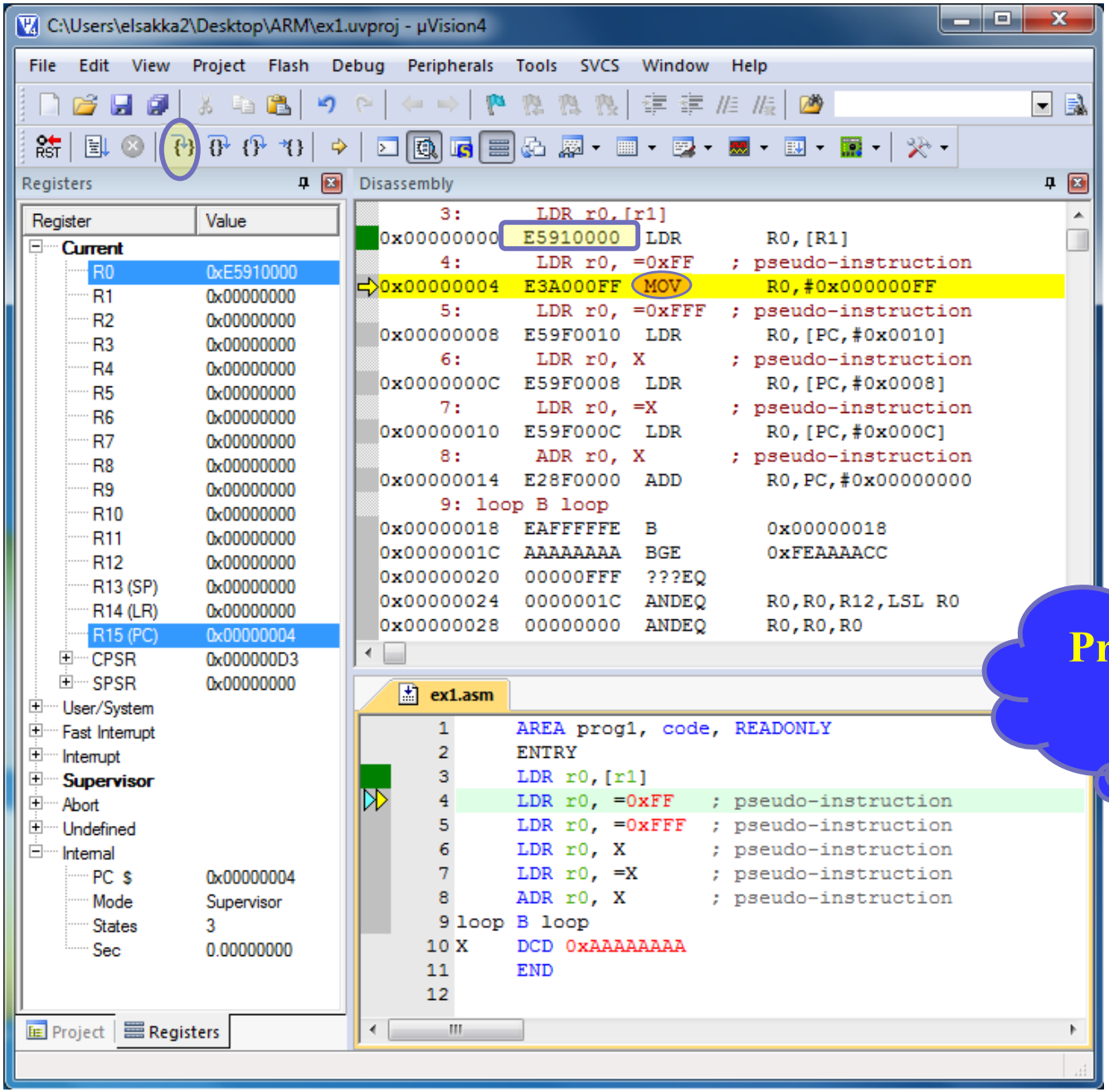
ARM pseudo-instructions



ARM pseudo-instructions



ARM pseudo-instructions

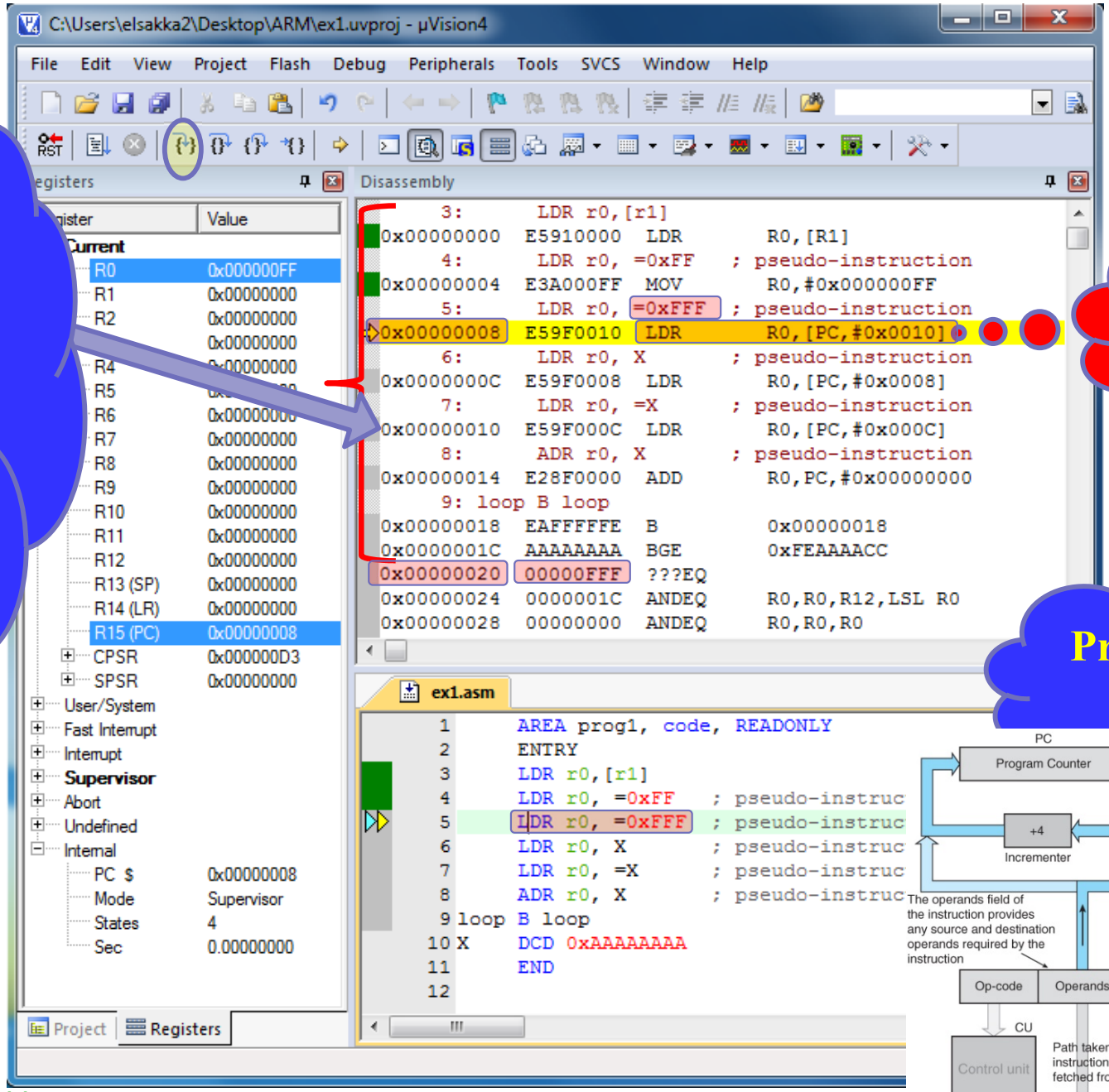


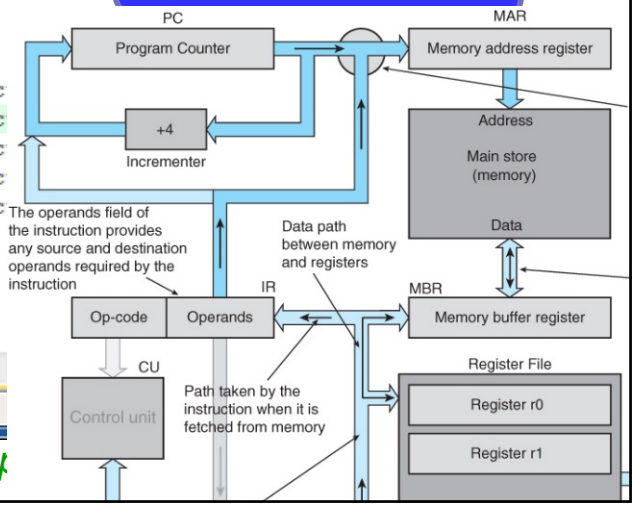
ARM pseudo-instructions

When executing the instruction at location 0x00000008, the PC value will be 0x00000010

How is this offset calculated?

Press Step, or F11



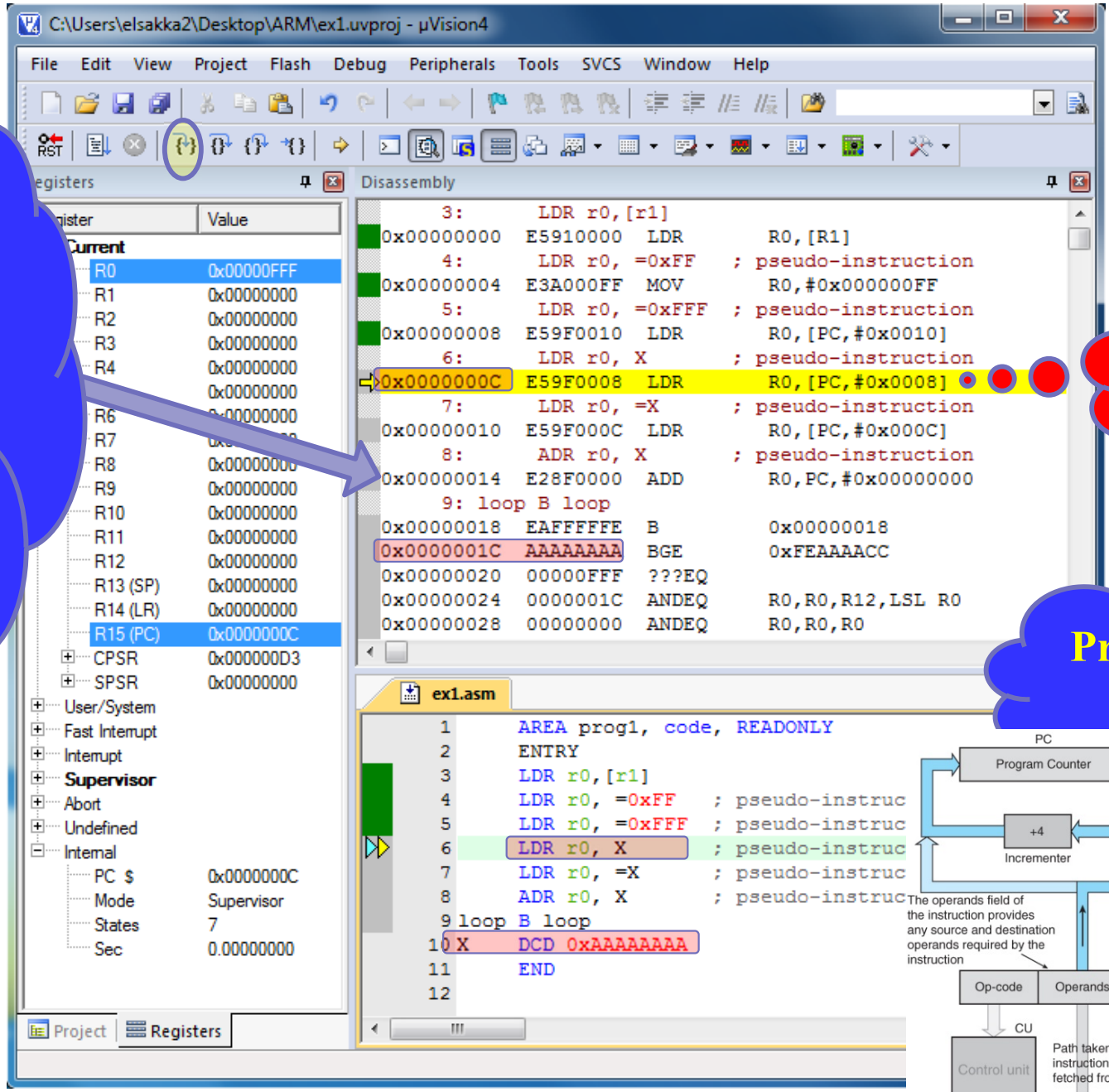


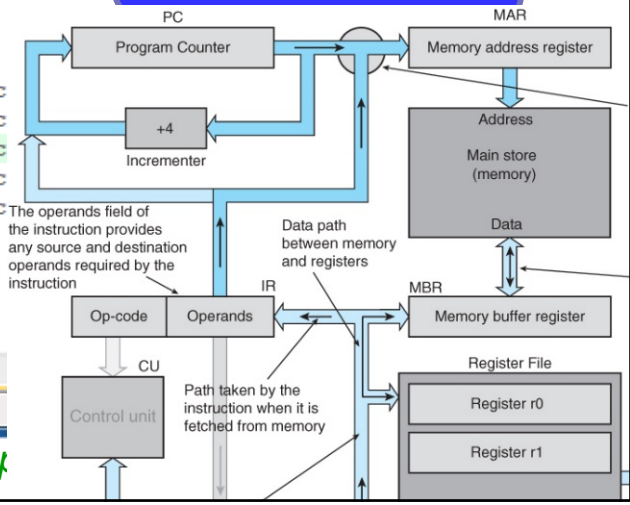
ARM pseudo-instructions

When executing the instruction at location 0x0000000C, the PC value will be 0x00000014

How is this offset calculated?

Press Step, or F11



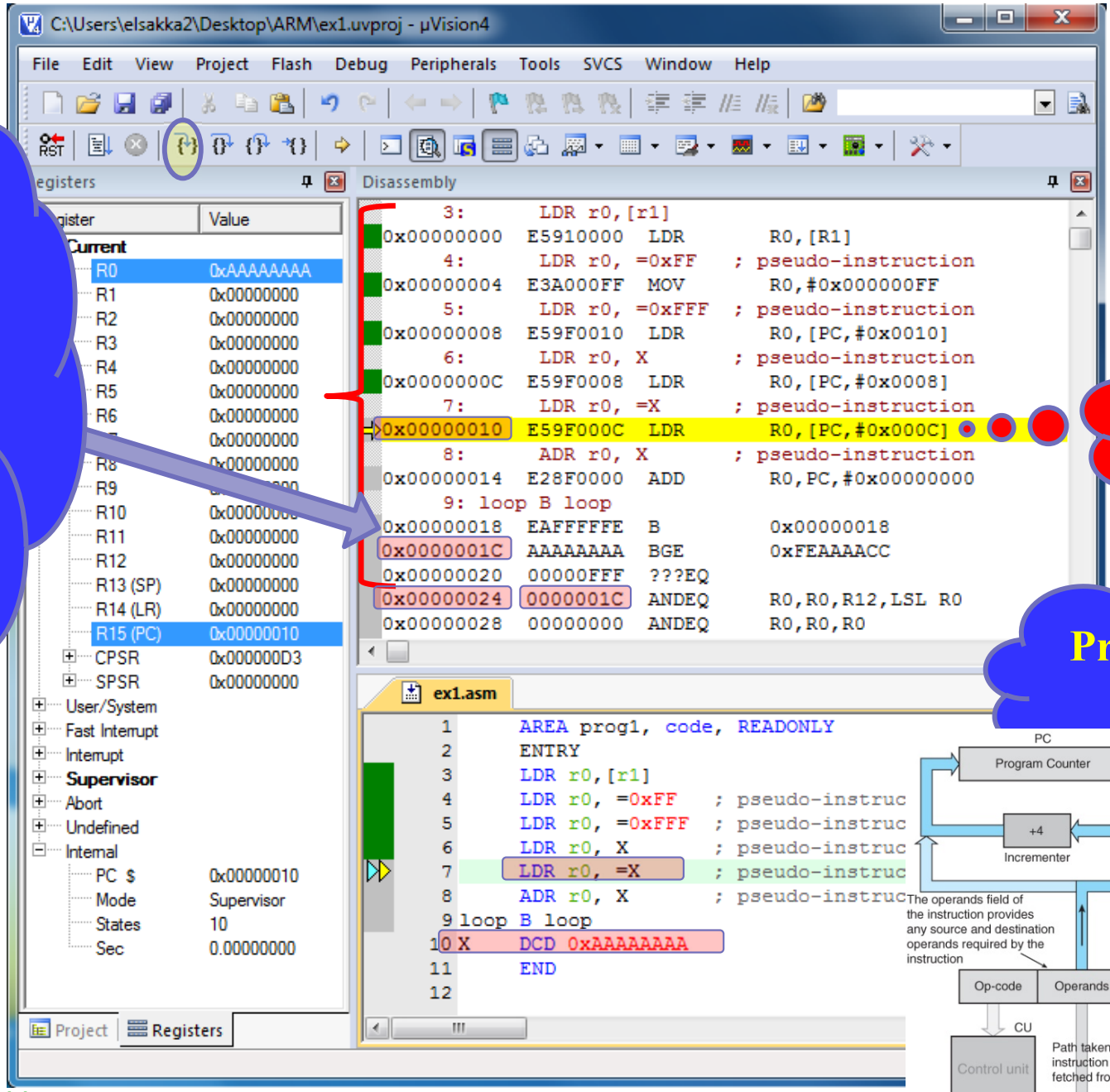


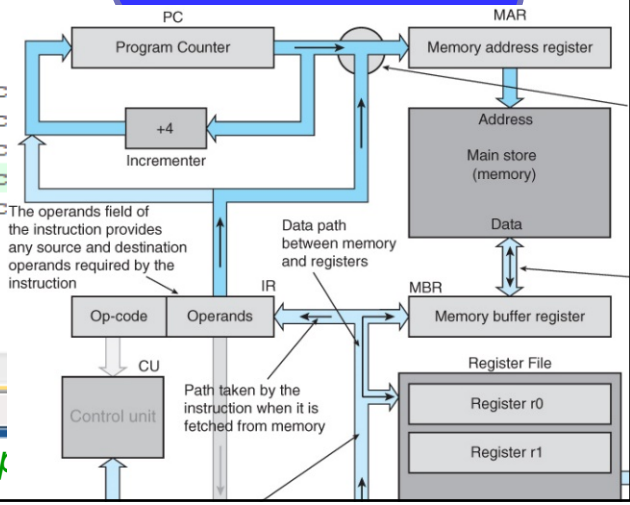
ARM pseudo-instructions

When executing the instruction at location 0x00000010, the PC value will be 0x00000018

How is this offset calculated?

Press Step, or F11





ARM pseudo-instructions

When executing the instruction at location 0x00000014, the PC value will be 0x0000001C

Compare the translations of
`LDR r0, =X`
and
`ADR r0, X`

How is this offset calculated?

Press Step, or F11

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

File Edit View Project Flash Debug Peripherals Tools SVCS Window

Registers

Register	Value
Current	
R0	0x0000001C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	
R11	
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000014
Mode	Supervisor
States	13
Sec	0.00000000

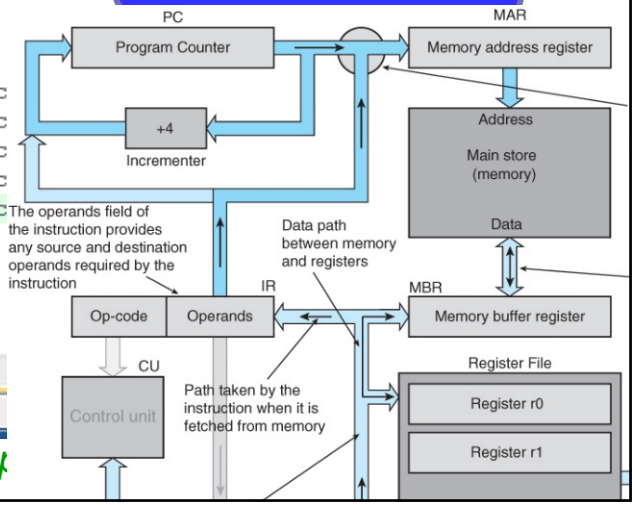
Project Registers

Disassembly

```
     3:      LDR r0,[r1]
0x00000000  E5910000  LDR       R0,[R1]
     4:      LDR r0, =0xFF   ; pseudo-instruction
0x00000004  E3A000FF  MOV       R0,#0x000000FF
     5:      LDR r0, =0xFFF  ; pseudo-instruction
0x00000008  E59F0010  LDR       R0,[PC,#0x0010]
     6:      LDR r0, X       ; pseudo-instruction
0x0000000C  E59F0008  LDR       R0,[PC,#0x0008]
     7:      LDR r0, =X      ; pseudo-instruction
0x00000010  E59F000C  LDR       R0,[PC,#0x000C]
     8:      ADR r0, X       ; pseudo-instruction
0x00000014  E28F0000  ADD       R0,PC,#0x00000000
     9: loop B loop
0x00000018  EAFFFFFE  B         0x00000018
0x0000001C  AAAAAAAA  BGE       0xFEAAAACC
0x00000020  00000FFF  ???EQ
0x00000024  0000001C  ANDEQ     R0,R0,R12,LSL R0
0x00000028  00000000  ANDEQ     R0,R0,R0
```

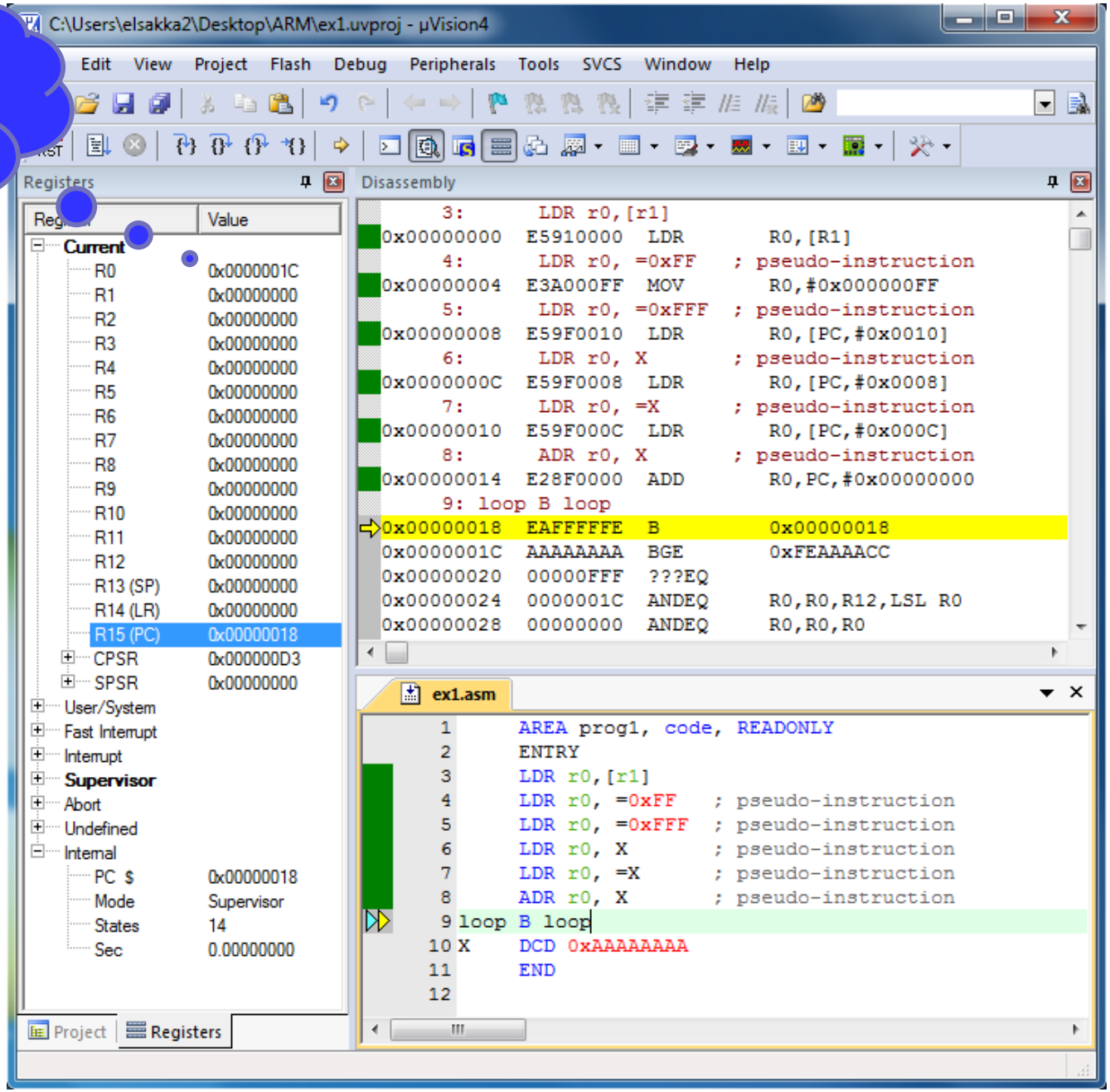
ex1.asm

```
1       AREA prog1, code, READONLY
2       ENTRY
3       LDR r0,[r1]
4       LDR r0, =0xFF   ; pseudo-instruc
5       LDR r0, =0xFFF  ; pseudo-instruc
6       LDR r0, X       ; pseudo-instruc
7       LDR r0, =X      ; pseudo-instruc
8       ADR r0, X       ; pseudo-instruc
9 loop  B loop
10 X    DCD 0xAAAAAAAA
11      END
12
```



ARM pseudo-instructions

Same address (no change)



ARM pseudo-instructions

- Consider we changed the previous program as follow:

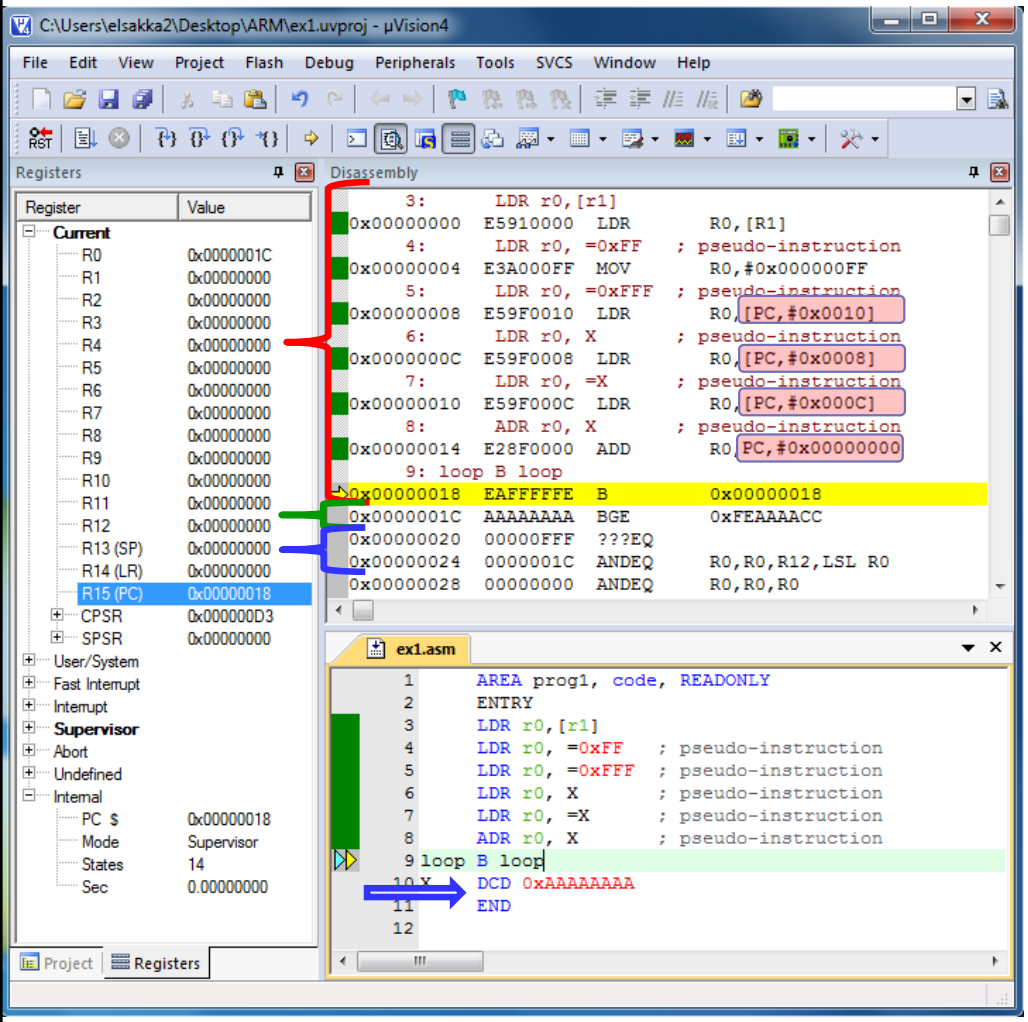
```
AREA prog1, code, READONLY
     ENTRY
     LDR r0,[r1]
     LDR r0, =0xFF    ; pseudo-instruction
     LDR r0, =0xFFF   ; pseudo-instruction
     LDR r0, X        ; pseudo-instruction
     LDR r0, =X       ; pseudo-instruction
     ADR r0, X        ; pseudo-instruction
loop B loop
X    DCD 0xAAAAAAAA
     END
```

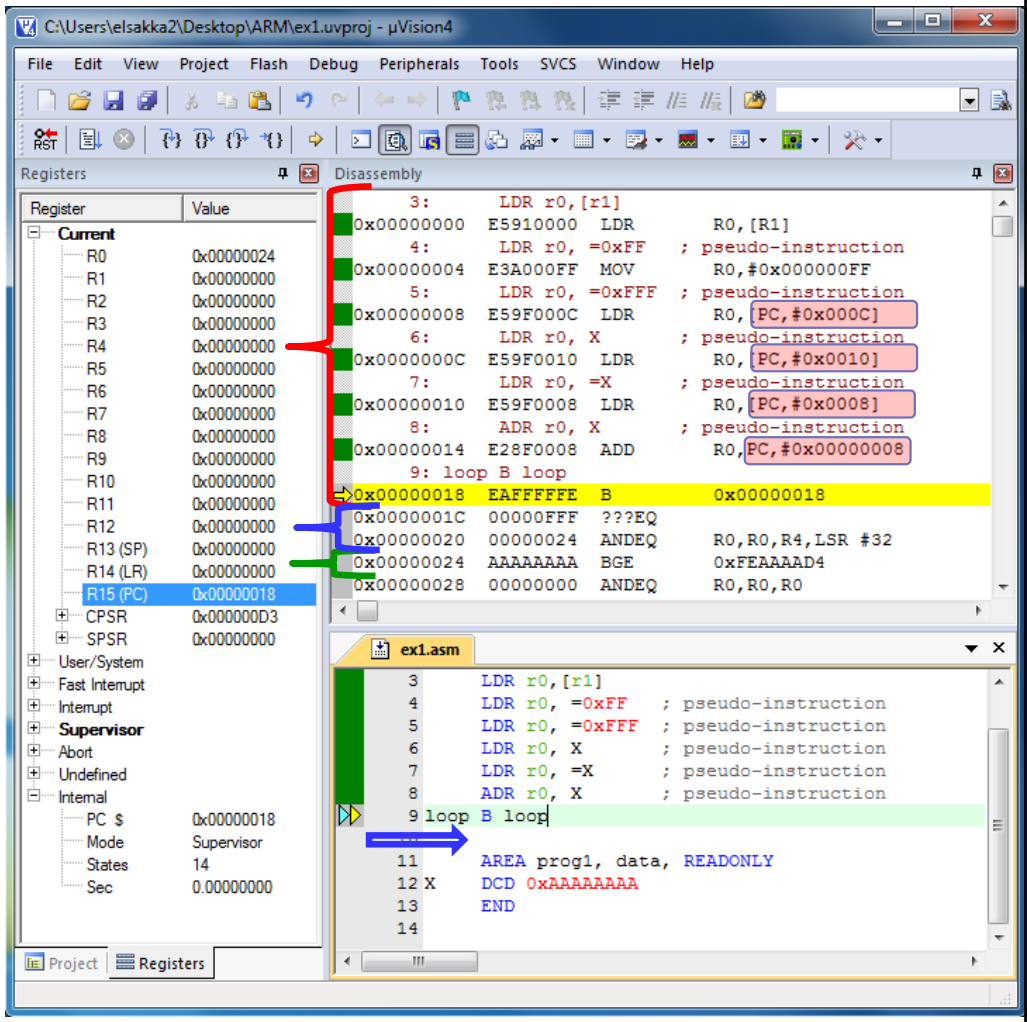
```
     AREA prog1, code, READONLY
     ENTRY
     LDR r0,[r1]
     LDR r0, =0xFF    ; pseudo-instruction
     LDR r0, =0xFFF   ; pseudo-instruction
     LDR r0, X        ; pseudo-instruction
     LDR r0, =X       ; pseudo-instruction
     ADR r0, X        ; pseudo-instruction
loop B loop

     AREA prog1, data, READONLY
X    DCD 0xAAAAAAAA
     END
```

- What is the effect of this change on the generated code?

ARM pseudo-instructions



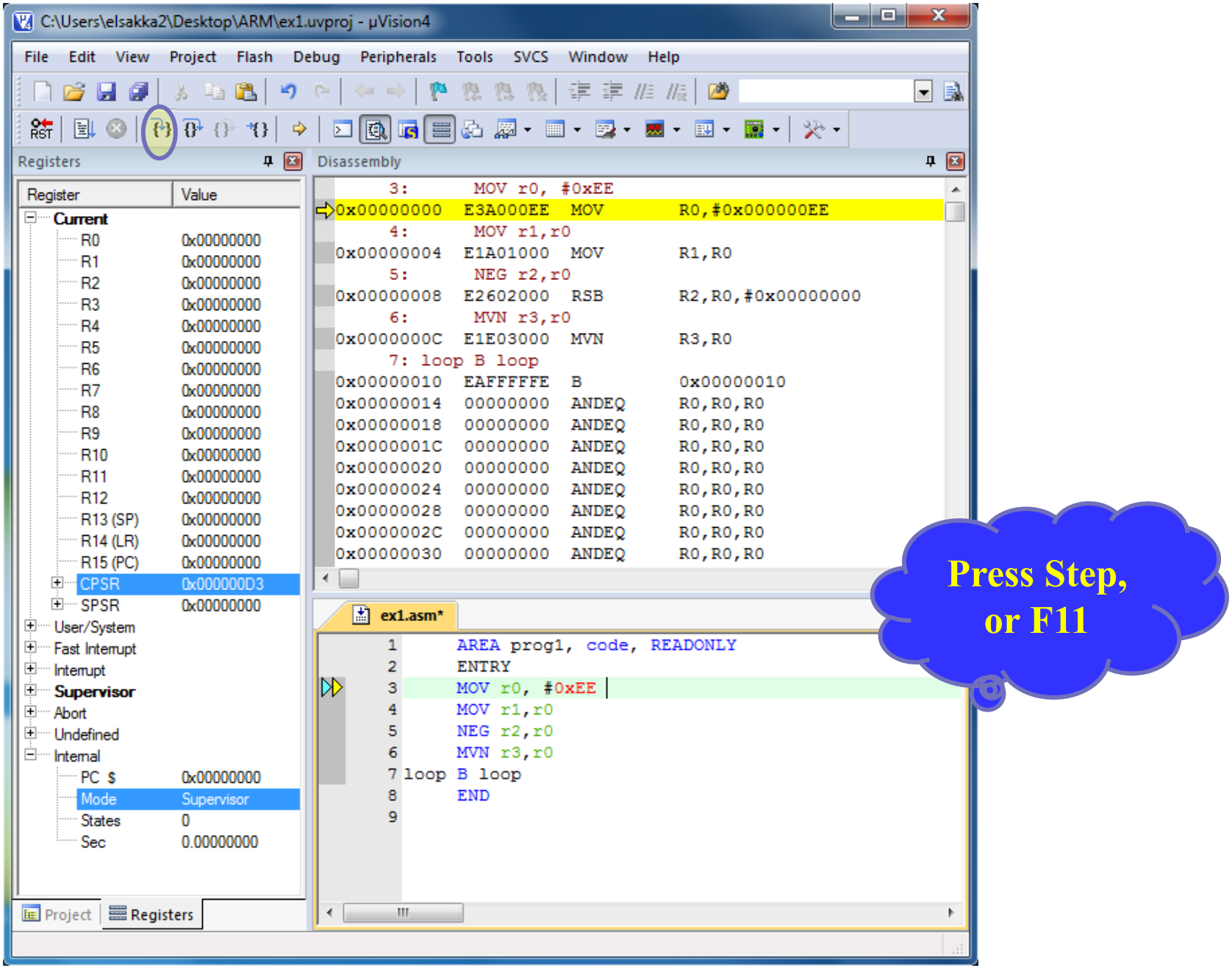


ARM pseudo-instructions

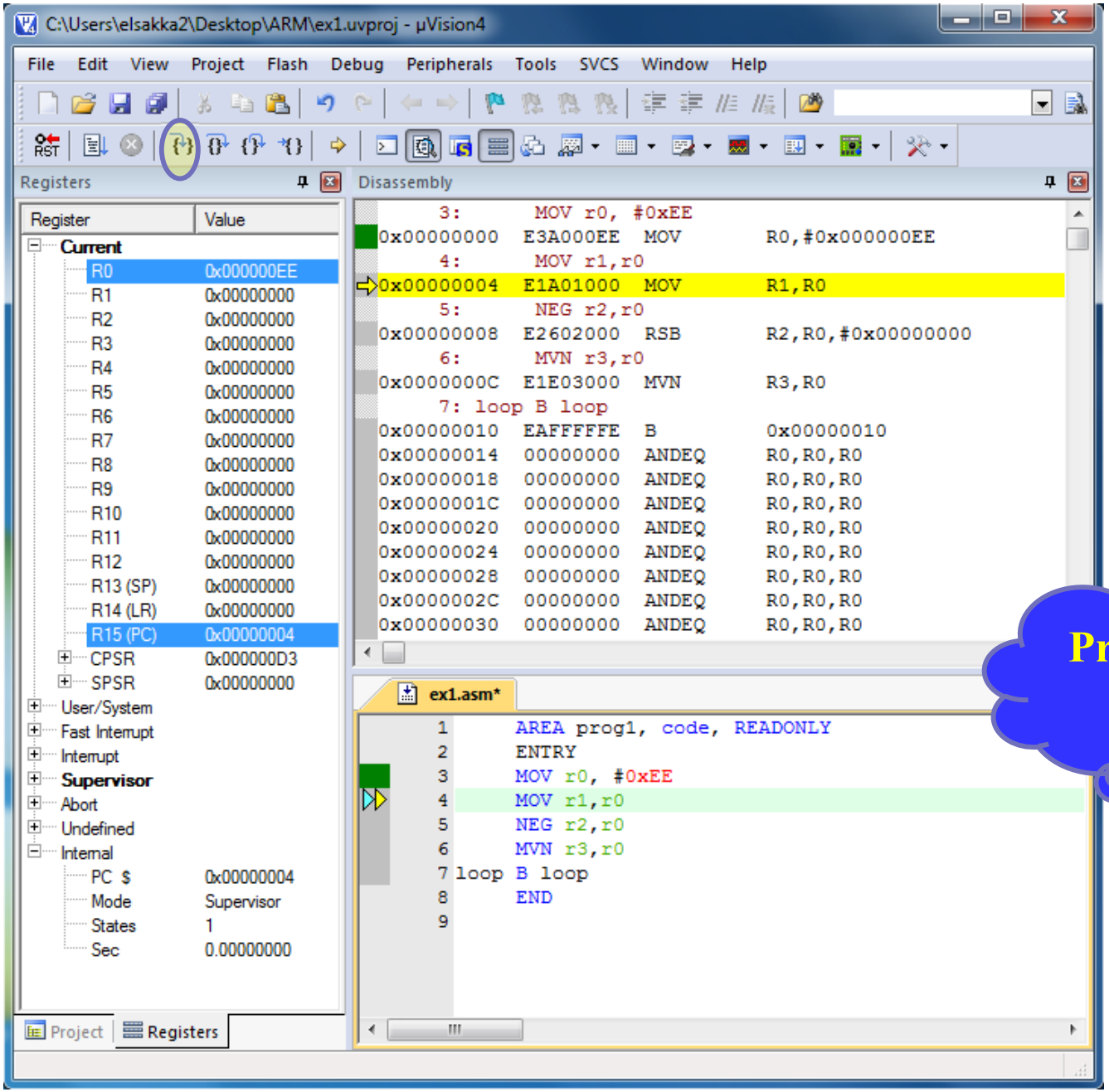
- Consider the following assembly program:

```
      AREA prog1, code, READONLY
      ENTRY
      MOV r0, #0xEE
      MOV r1,r0
      NEG r2,r0
      MVN r3,r0
loop  B loop
      END
```

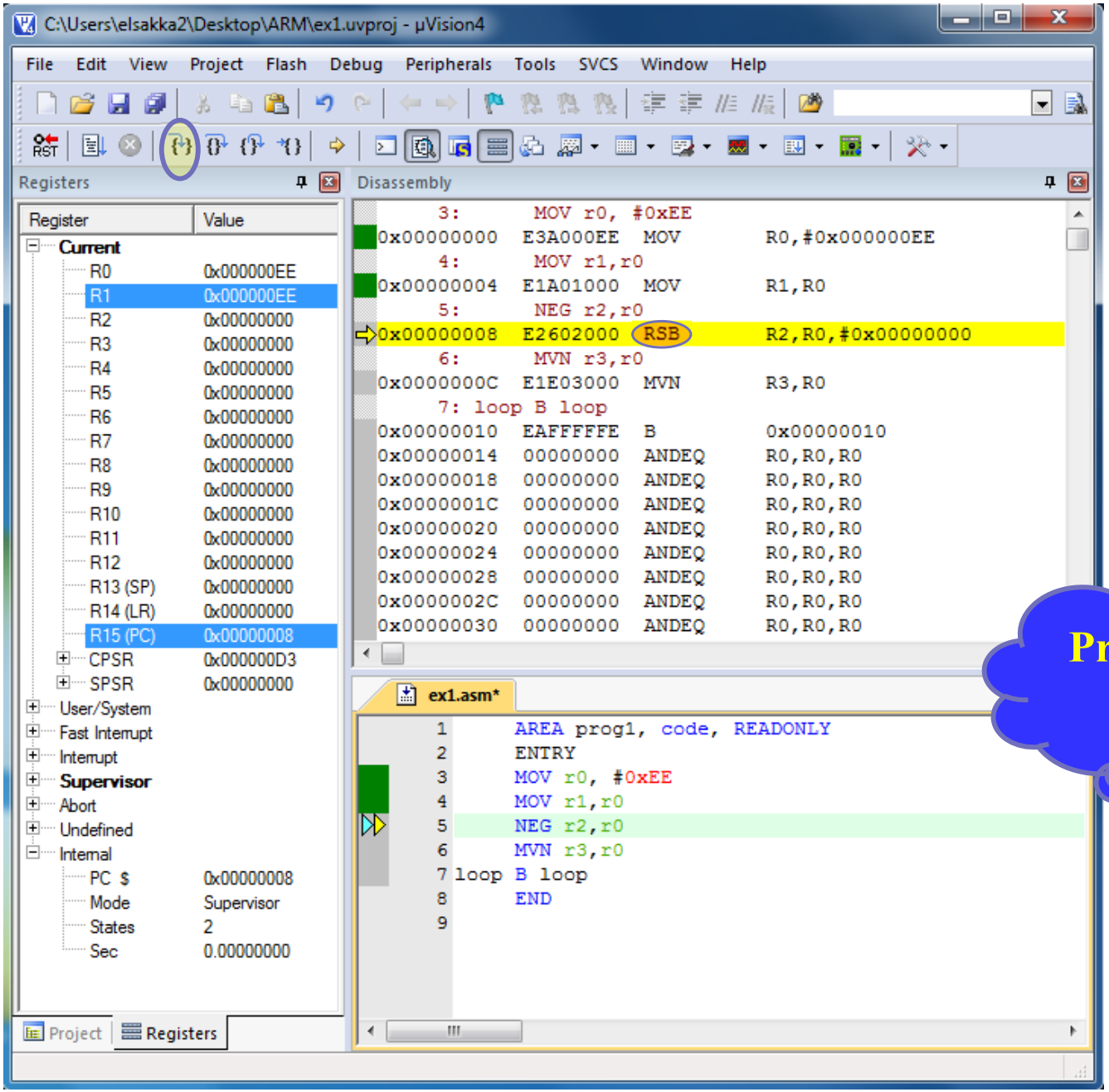
ARM pseudo-instructions



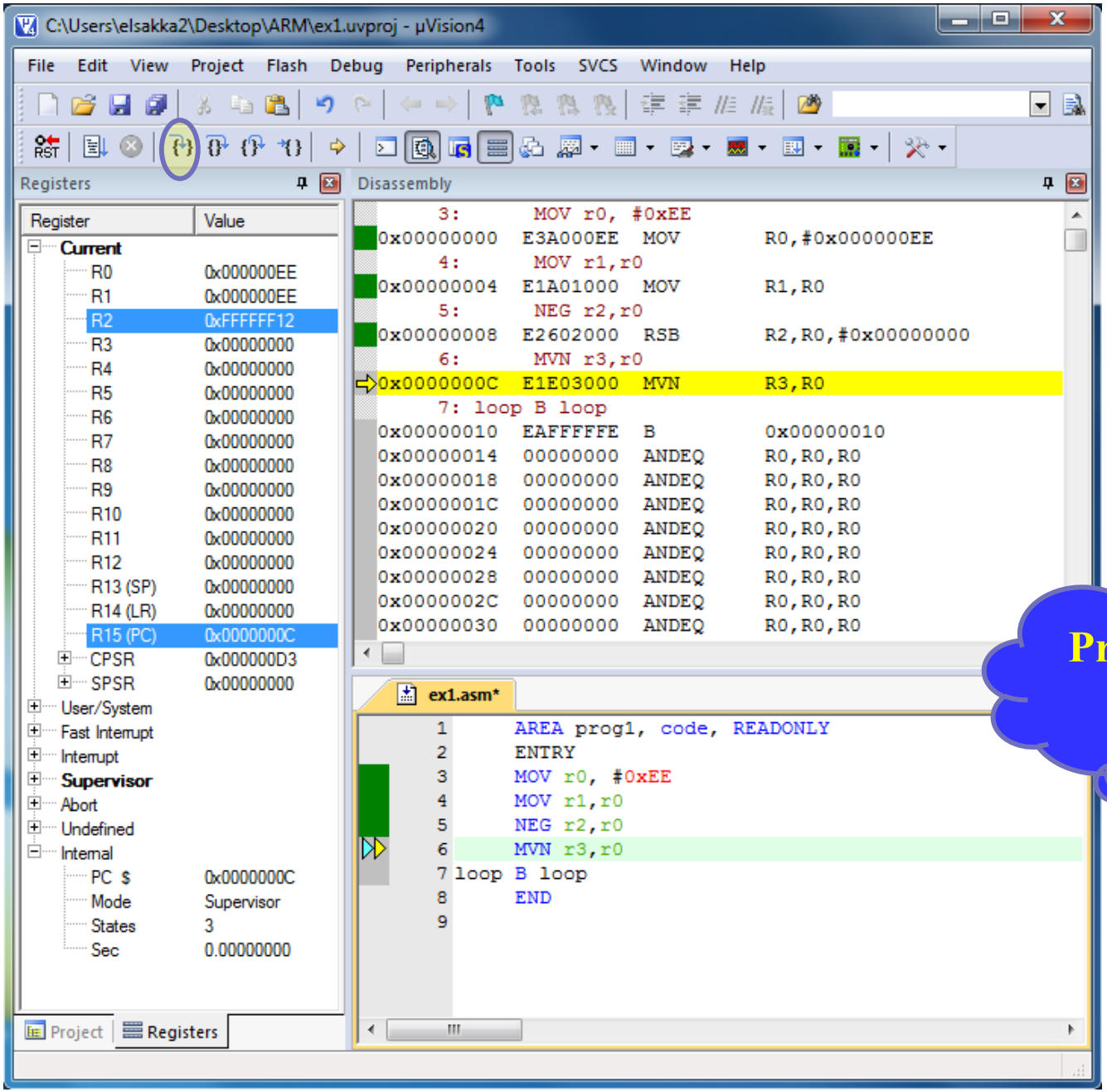
ARM pseudo-instructions



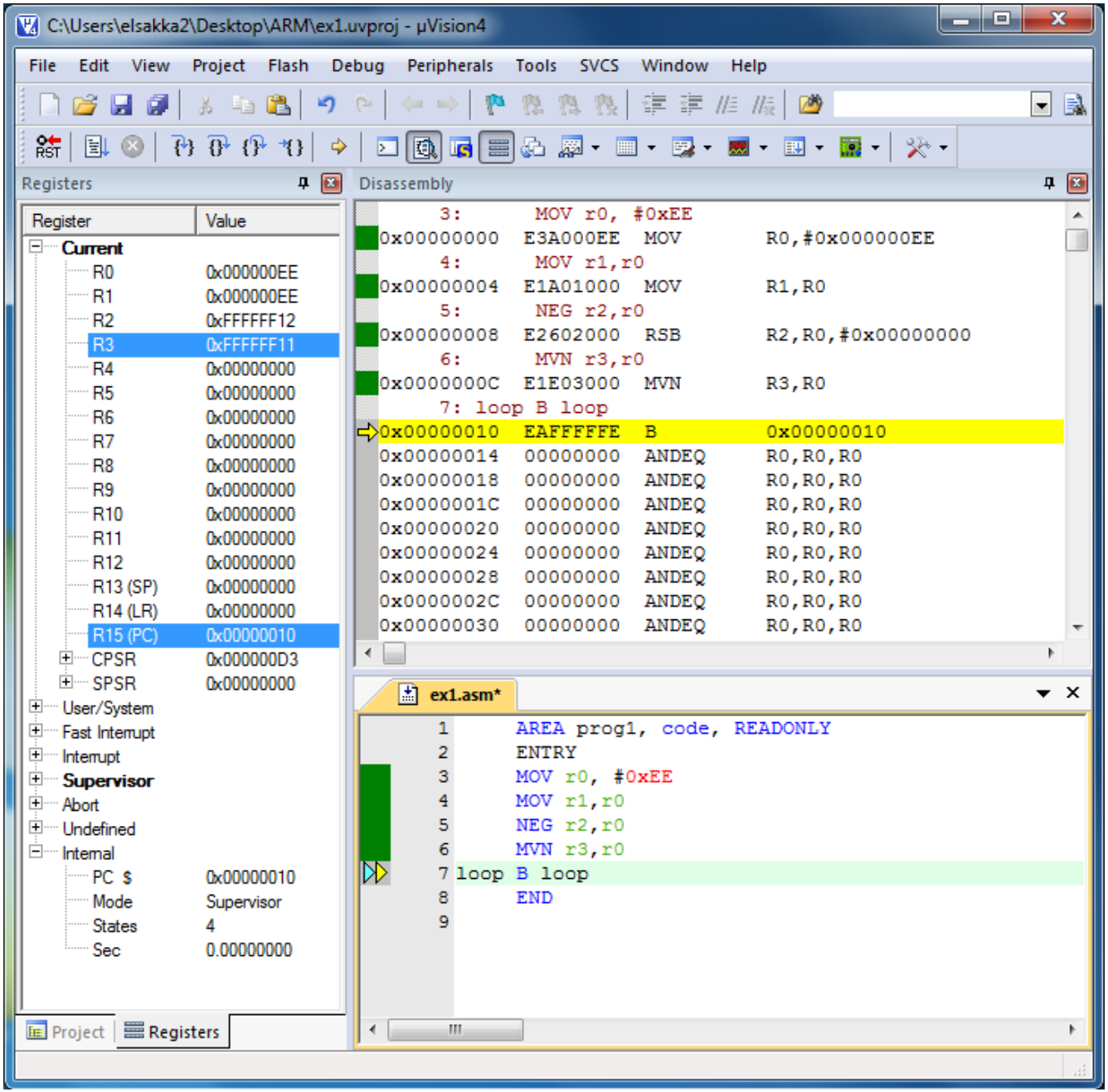
ARM pseudo-instructions



ARM pseudo-instructions



ARM pseudo-instructions



ARM pseudo-instructions

- Consider we changed the previous program as follow:

```
 AREA prog1, code, READONLY
     ENTRY
     MOV r0, #0xEE
     MOV r1,r0
     NEG r2,r0
     MVN r3,r0
loop B loop
     END
```

```
 AREA prog1, code, READONLY
     ENTRY
     MOV r0, #-0xEE
     MOV r1,r0
     NEG r2,r0
     MVN r3,r0
loop B loop
     END     END
```

- What is the effect of this change on the generated code?

ARM pseudo-instructions