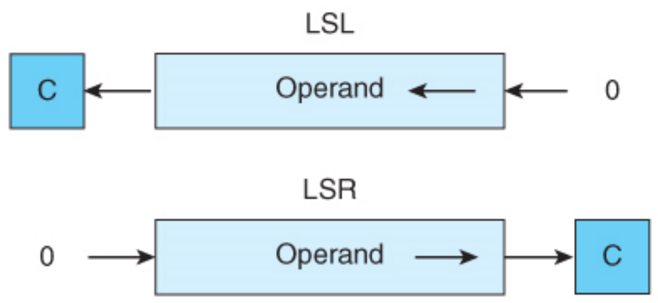
Tutorial 10: ARM Shift Instructions

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

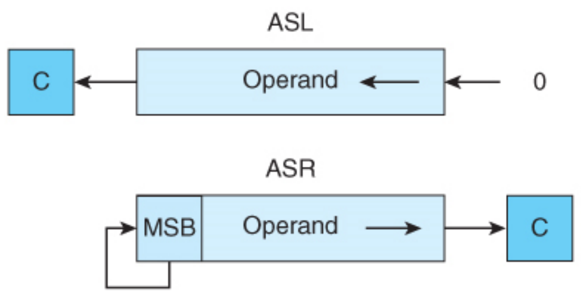
ARM's Data-Processing Instructions (Shift Operations)

- *Shift* operations move bits one ***or more*** places *left* or *right*.
  - ***Logical shifts***
    - *insert a 0* in the vacated position.
  - ***Arithmetic shifts***
    - *replicate the sign-bit* during a right shift
  - ***Circular shifts***
    - *the bit shifted out of one end is shifted in the other end* i.e., the register is treated as a ring
  - ***Circular shifts through carry***
    - *included the carry bit in the shift path*



In a logical shift, a zero is shifted in and the bit shifted out is copied to the carry bit of the condition code register.

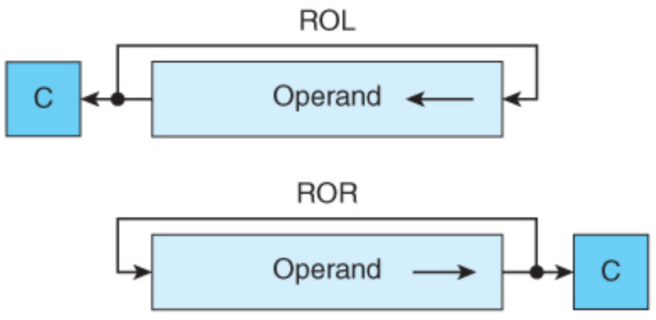
(a) Logical shift



In an arithmetic shift, the number is either multiplied by 2 (ASL) or divided by 2 (ASR). The sign of a two's complement number is preserved.

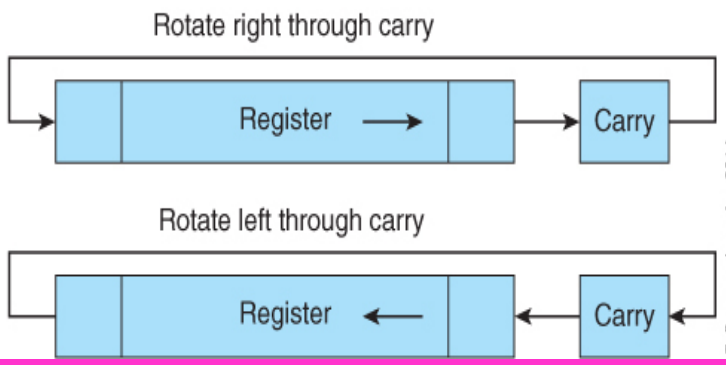
The bit shifted out is copied into the carry bit.

(b) Arithmetic shift



In a rotate operation, the bit shifted out is copied into the bit vacated at the other end (i.e., no bit is lost during a rotate). The bit shifted out is also copied into the carry bit.

(c) Rotate



ARM's Data-Processing Instructions (Shift Operations)

- ARM support both *static* and *dynamic* shifts (except *rotate through carry* instruction which allows *only one single shift* per instruction)
  - In *static shift*, the number of shift places is determined *when the code is written*
  - In *static shift*, the range of the number of shift places is as follow:
    - **LSL**: the range is from ***#0*** to ***#31*** ***(32 different values)***
    - **LSR**: the range is from ***#1*** to ***#32*** ***(32 different values)***
    - **ASR**: the range is from ***#1*** to ***#32*** ***(32 different values)***
    - **ROR**: the range is from ***#1*** to ***#31*** ***(31 different values)***

      The remaining value is used to encode RRX
      - **ROR** + ***a shift of #0*** ➔ **RRX**

Only 5 bits are needed to encode the amount of shifts.

In case of **LSR** and **ASR**, the value **#32** is encoded as **00000**

  - In *dynamic shift*, the number of shift places
    - is determined *when the code is executed, i.e., at run time*
    - If the number of dynamic shifts is *≥ 32*, zero will be stored in the destination

ARM's Data-Processing Instructions (Shift Operations)

- ARM implements only the following five shifts
    - ***LSL*** logical shift left
    - ***LSR*** logical shift right
    - ***ASR*** arithmetic shift right
    - ***ROR*** rotate right
    - ***RRX*** rotate right through carry (one shift)
- ***Other shift operations have to be synthesized by the programmer.***
    - An *arithmetic shift left* is effectively the same as a *logical shift left*
    - For a 32-bit value, an ***n*-bit** *rotate shift left* is identical to a ***32 – n*** *rotate shift right*
    - Rotate left through carry can be implemented by means of

      ```
      ADCS r0,r0,r0 ; add r0 to r0 with carry and set the flags
      ```

        - The instruction means r0 + r0 + C, i.e., 2 × r0 + C, i.e.,
            - shifting left the content of r0
            - store the value of C in the vacant bit to the left, and
            - storing the shifted out bit in the carry flag

ARM’s Data-Processing Instructions (Shift Operations)

- ARM *has no explicit shift operations!!.*
- ARM *combines shifting with other data processing operations*, where
  - the ***second operand*** in the arithmetic operation (i.e., the ***LAST parameter in the assembly arithmetic instruction***) is allowed to be shifted ***before*** it is used.
  - For example,
    `ADD r0,r1,r2,LSL #1` ; [r0] ← [r1] + [r2] × 2
    - logically shift left the contents of r2,
    - add the result to the contents of r1, and
    - put the results in r0
- ARM *also combines shifting with moving operations*
  - This way, a shift operation can be performed as a stand alone operation.
  - For example,
    `MOV r3,r3,LSL #1` ; [r3] ← [r3] × 2
  - ARM provides pseudo shift instructions, which are translated to `MOV` instructions.

```
LSL r3,r3,#1      ; will be converted to MOV r3,r3,LSL #1
```

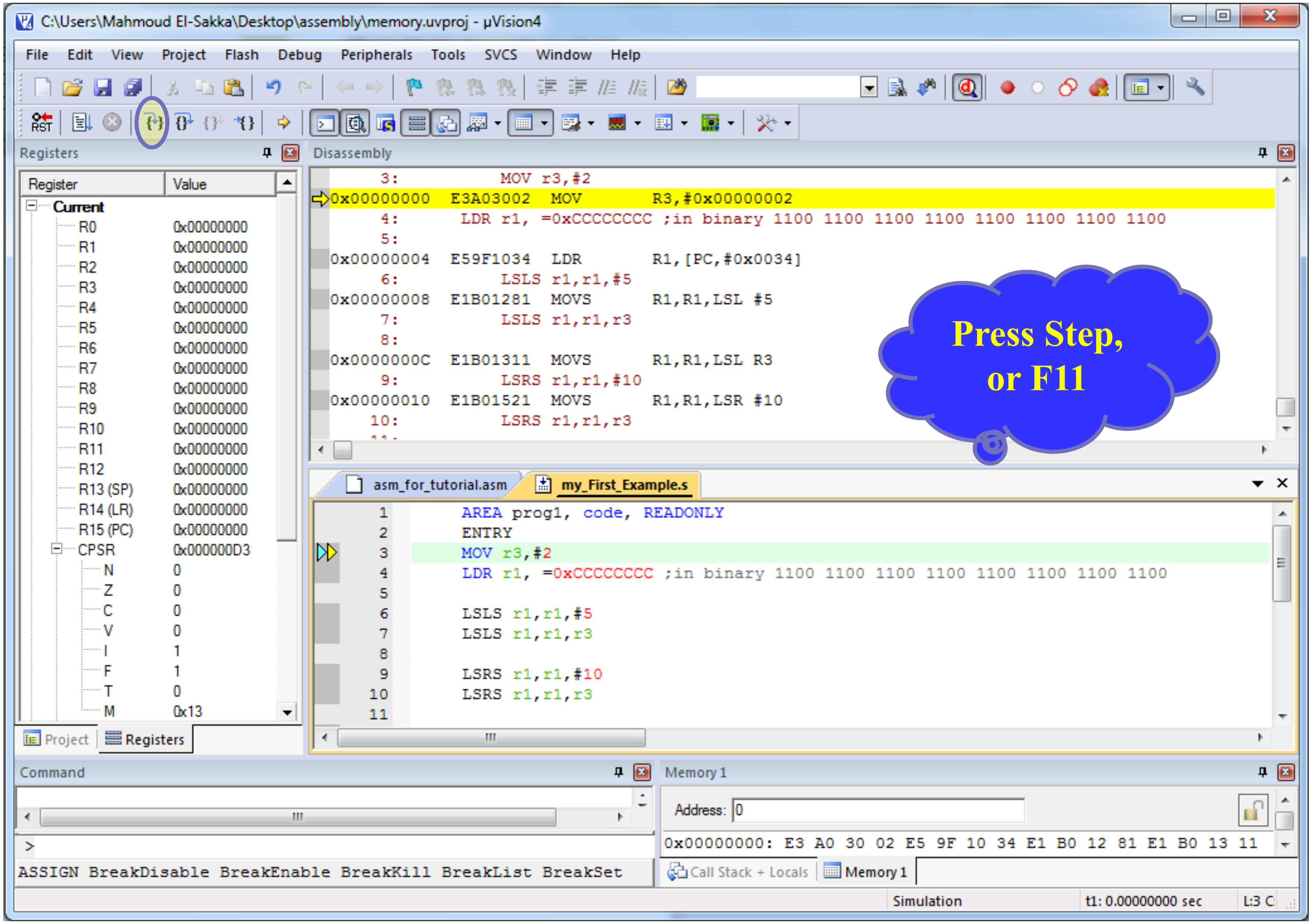
or simply

```
LSL r3,#1
```

ARM’s Data-Processing Instructions (Shift Operations)

```
AREA prog1, code, READONLY
ENTRY
MOV r3,#2
LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

LSLS r1,r1,#5   r1=r1·2^5
LSLS r1,r1,r3

LSRS r1,r1,#10
LSRS r1,r1,r3

ASRS r1,r1,#2
LSLS r1,r1,#15
ASRS r1,r1,#16

ASRS r1,r1,r3

RORS r1,r1,#4
RORS r1,r1,r3

RRXS r1,r1
RRXS r1,r1
RRXS r1,r1
RRXS r1,r1
END
```

ARM's Data-Processing Instructions (Shift Operations)



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
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

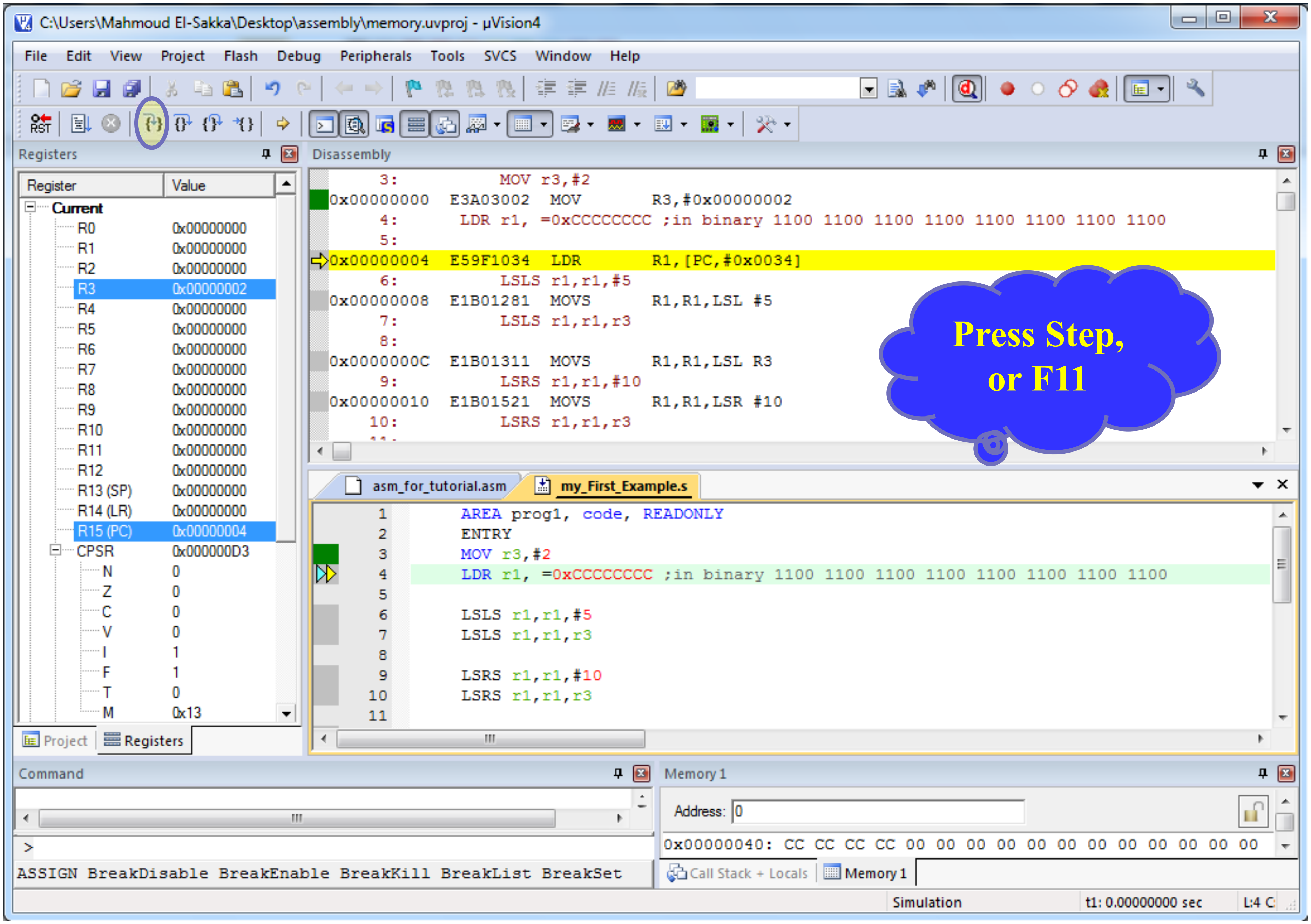
```
     3:          MOV r3,#2
0x00000000  E3A03002  MOV       R3,#0x00000002
     4:      LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR       R1,[PC,#0x0034]
     6:          LSLS r1,r1,#5
0x00000008  E1B01281  MOVS      R1,R1,LSL #5
     7:          LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
```

```
        AREA prog1, code, READONLY
        ENTRY
        MOV r3,#2
        LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3
```

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

```
Address: 0
0x00000000: E3 A0 30 02 E5 9F 10 34 E1 B0 12 81 E1 B0 13 11
```

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000002
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
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

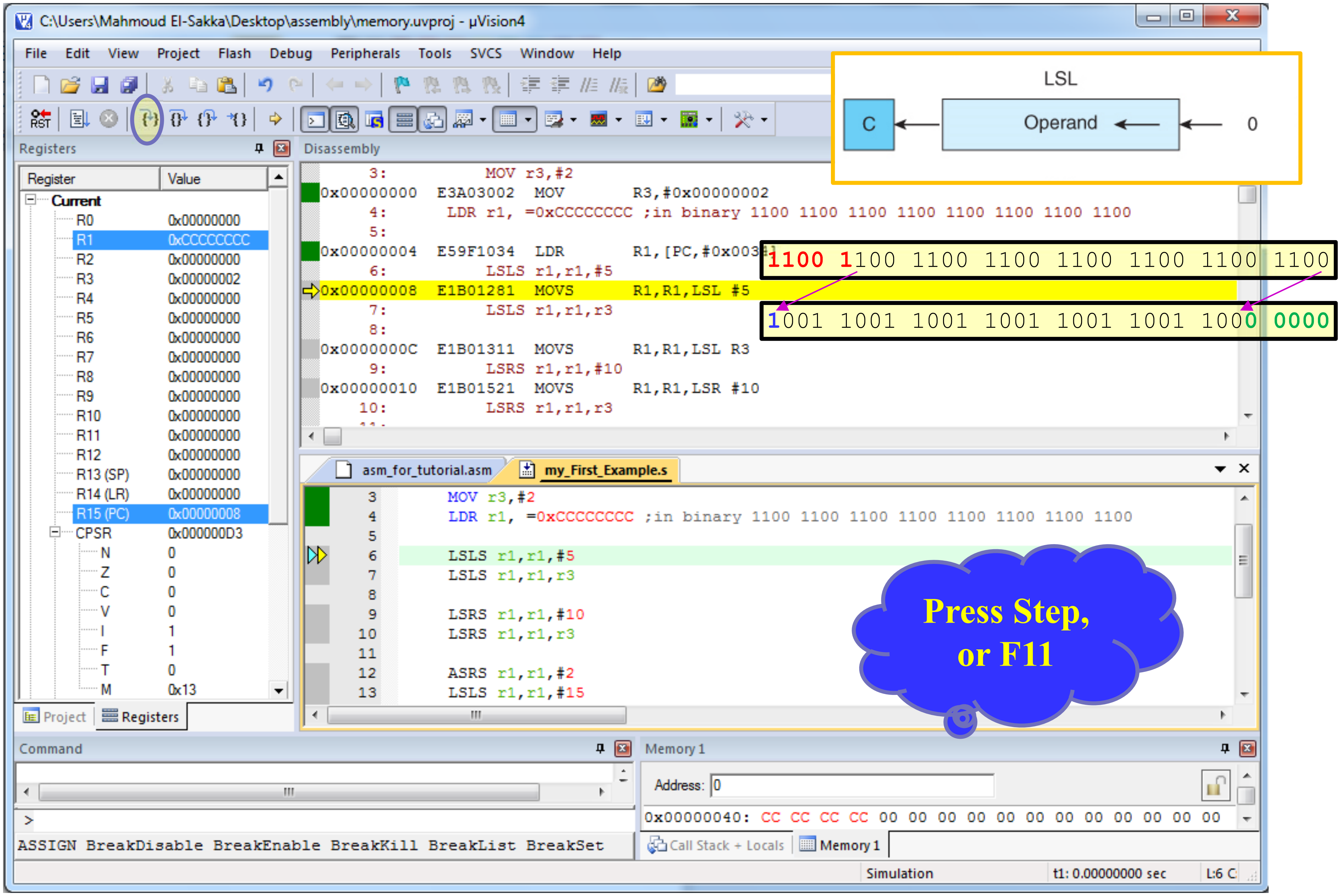
```
     3:          MOV r3,#2
0x00000000  E3A03002  MOV       R3,#0x00000002
     4:      LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR       R1,[PC,#0x0034]
     6:          LSLS r1,r1,#5
0x00000008  E1B01281  MOVS      R1,R1,LSL #5
     7:          LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
```

```
        AREA prog1, code, READONLY
        ENTRY
        MOV r3,#2
        LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
LSL
C
Operand
0
Press Step, or F11

Registers

Register	Value
Current	
R0	0x00000000
R1	0xCCCCCCCC
R2	0x00000000
R3	0x00000002
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
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
     3:         MOV r3,#2
0x00000000  E3A03002  MOV      R3,#0x00000002
     4:     LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR      R1,[PC,#0x0034]
     6:         LSLS r1,r1,#5
0x00000008  E1B01281  MOVS     R1,R1,LSL #5
     7:         LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS     R1,R1,LSL R3
     9:         LSRS r1,r1,#10
0x00000010  E1B01521  MOVS     R1,R1,LSR #10
    10:         LSRS r1,r1,r3
```

```
1100 1100 1100 1100 1100 1100 1100 1100
1001 1001 1001 1001 1001 1001 1000 0000
```

asm_for_tutorial.asm | my_First_Example.s

```
    MOV r3,#2
    LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

    LSLS r1,r1,#5
    LSLS r1,r1,r3

    LSRS r1,r1,#10
    LSRS r1,r1,r3

    ASRS r1,r1,#2
    LSLS r1,r1,#15
```

Command

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

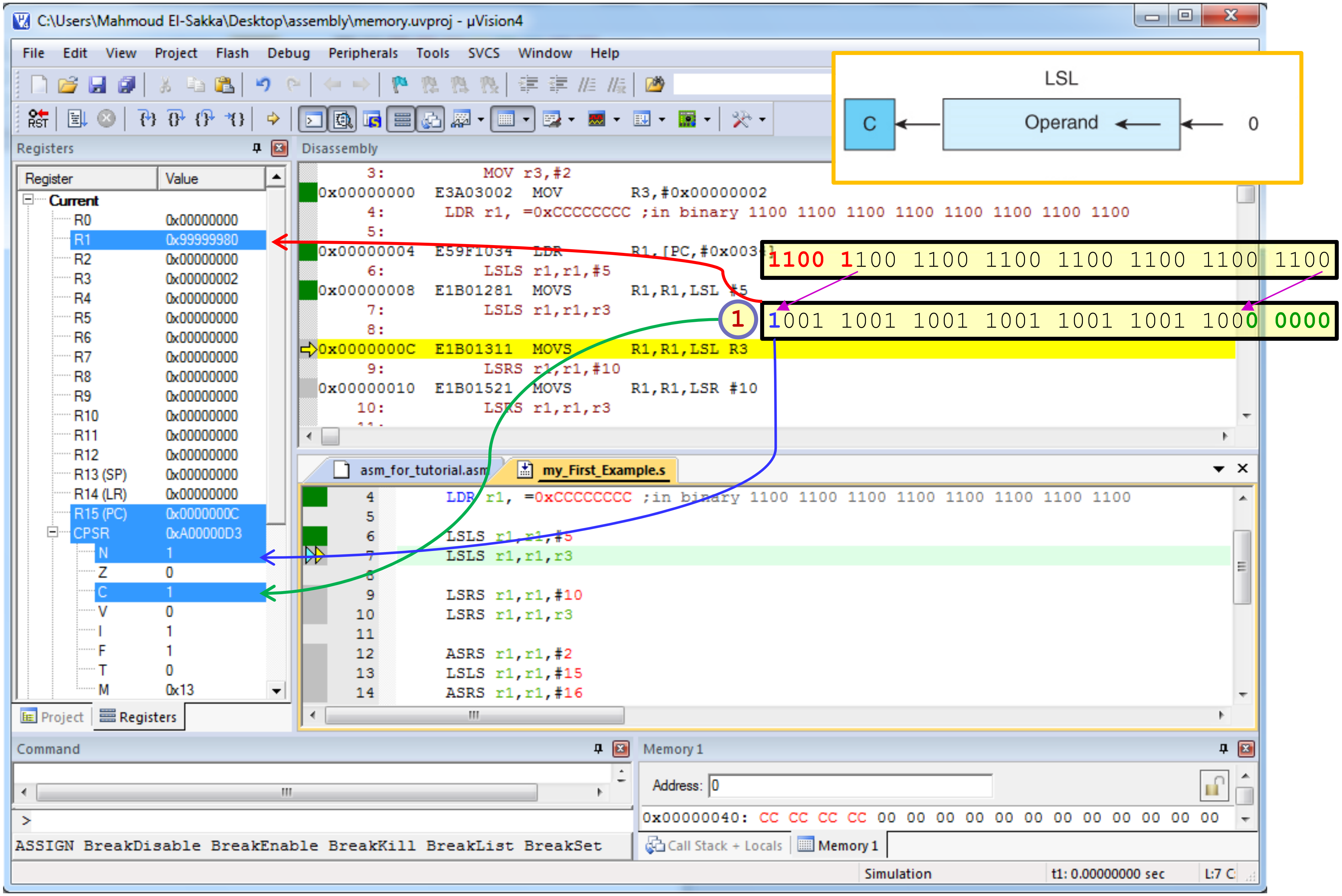
Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:6 C:

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x99999980
R2	0x00000000
R3	0x00000002
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	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
     3:             MOV r3,#2
0x00000000  E3A03002  MOV       R3,#0x00000002
     4:          LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR       R1,[PC,#0x0034]
     6:             LSLS r1,r1,#5
0x00000008  E1B01281  MOVS      R1,R1,LSL #5
     7:             LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:             LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:             LSRS r1,r1,r3
```

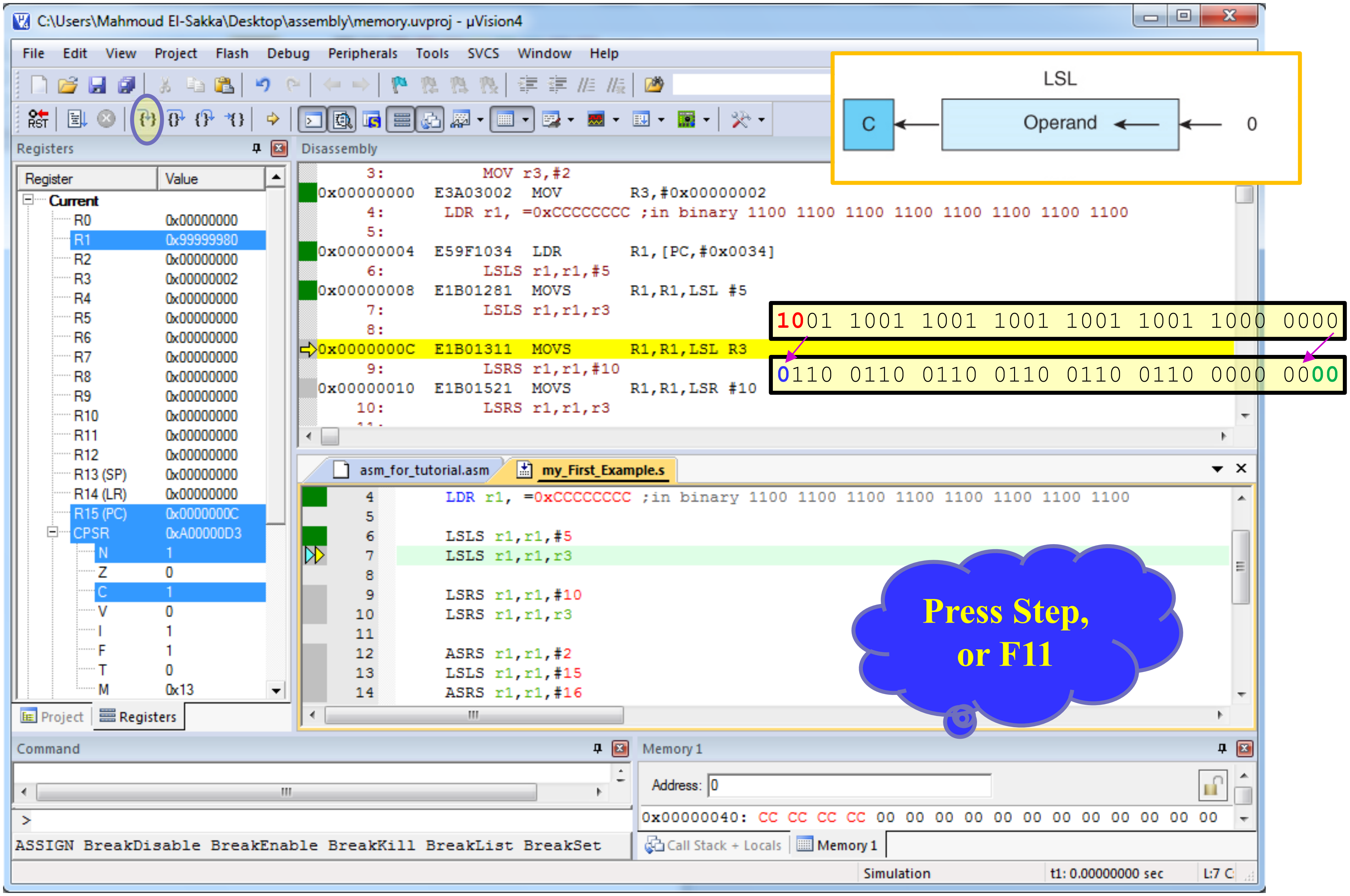
```
1100 1100 1100 1100 1100 1100 1100 1100
1001 1001 1001 1001 1001 1001 1000 0000
```

```
        LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16
```

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

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

ARM's Data-Processing Instructions (Shift Operations)



Registers

Register	Value
Current	
R0	0x00000000
R1	0x99999980
R2	0x00000000
R3	0x00000002
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	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
     3:         MOV r3,#2
0x00000000  E3A03002  MOV      R3,#0x00000002
     4:     LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR      R1,[PC,#0x0034]
     6:         LSLS r1,r1,#5
0x00000008  E1B01281  MOVS     R1,R1,LSL #5
     7:         LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS     R1,R1,LSL R3
     9:         LSRS r1,r1,#10
0x00000010  E1B01521  MOVS     R1,R1,LSR #10
    10:         LSRS r1,r1,r3
```

```
1001 1001 1001 1001 1001 1001 1000 0000
0110 0110 0110 0110 0110 0110 0000 0000
```

```
    LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

    LSLS r1,r1,#5
    LSLS r1,r1,r3

    LSRS r1,r1,#10
    LSRS r1,r1,r3

    ASRS r1,r1,#2
    LSLS r1,r1,#15
    ASRS r1,r1,#16
```

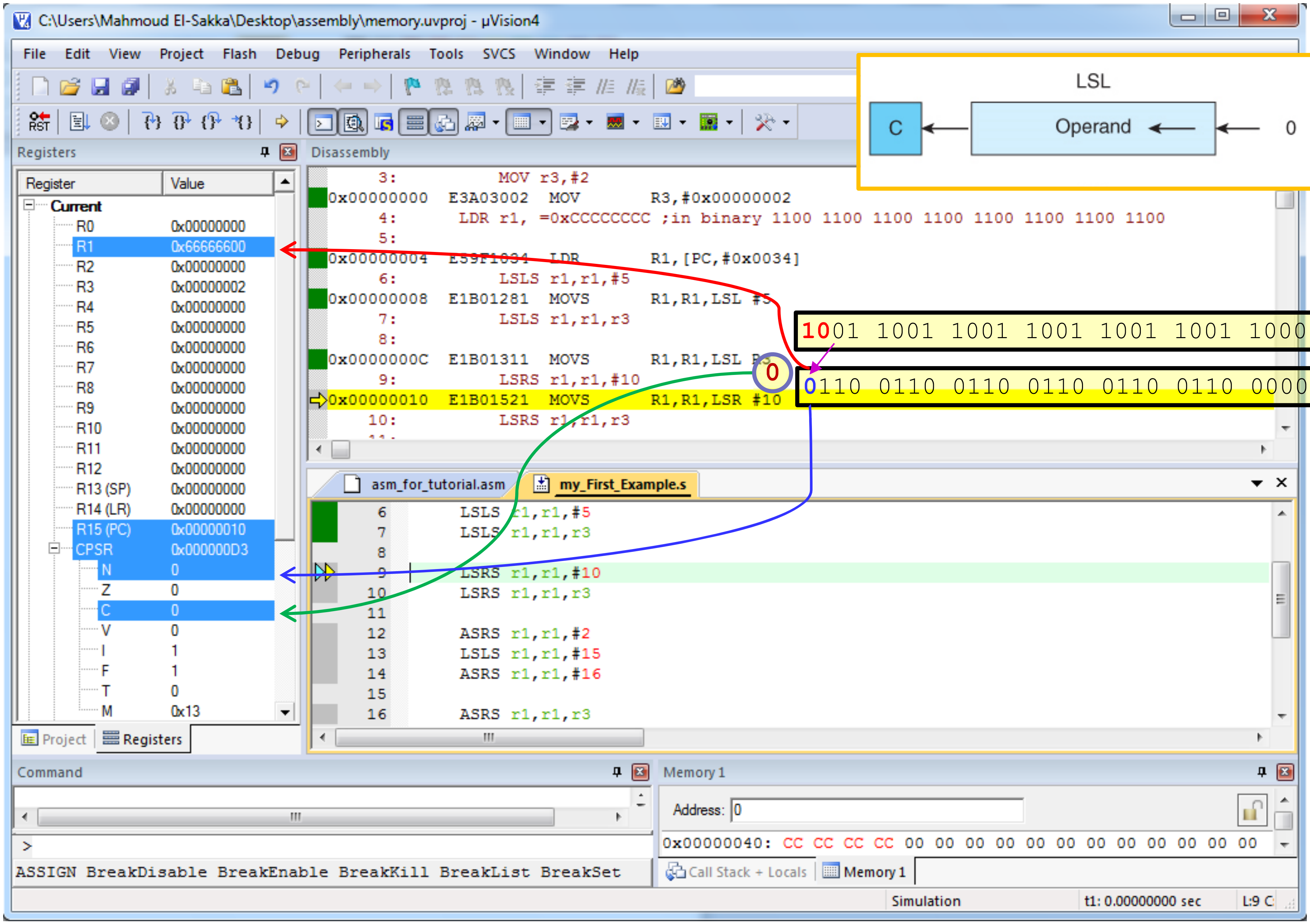
Command

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

Memory 1

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0x66666600
R2	0x00000000
R3	0x00000002
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
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
     3:          MOV r3,#2
0x00000000  E3A03002  MOV      R3,#0x00000002
     4:      LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
     5:
0x00000004  E59F1034  LDR      R1,[PC,#0x0034]
     6:          LSLS r1,r1,#5
0x00000008  E1B01281  MOVS     R1,R1,LSL #5
     7:          LSLS r1,r1,r3
     8:
0x0000000C  E1B01311  MOVS     R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS     R1,R1,LSR #10
    10:          LSRS r1,r1,r3
```

asm_for_tutorial.asm | my_First_Example.s

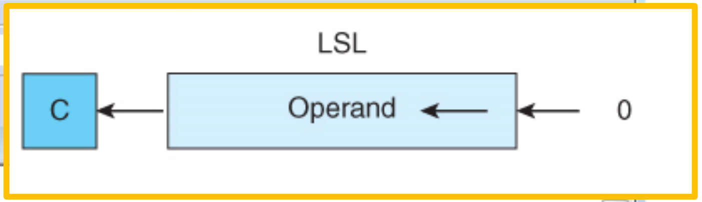
```
    LSLS r1,r1,#5
    LSLS r1,r1,r3

    LSRS r1,r1,#10
    LSRS r1,r1,r3

    ASRS r1,r1,#2
    LSLS r1,r1,#15
    ASRS r1,r1,#16

    ASRS r1,r1,r3
```

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

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

t1: 0.00000000 sec L:9 C

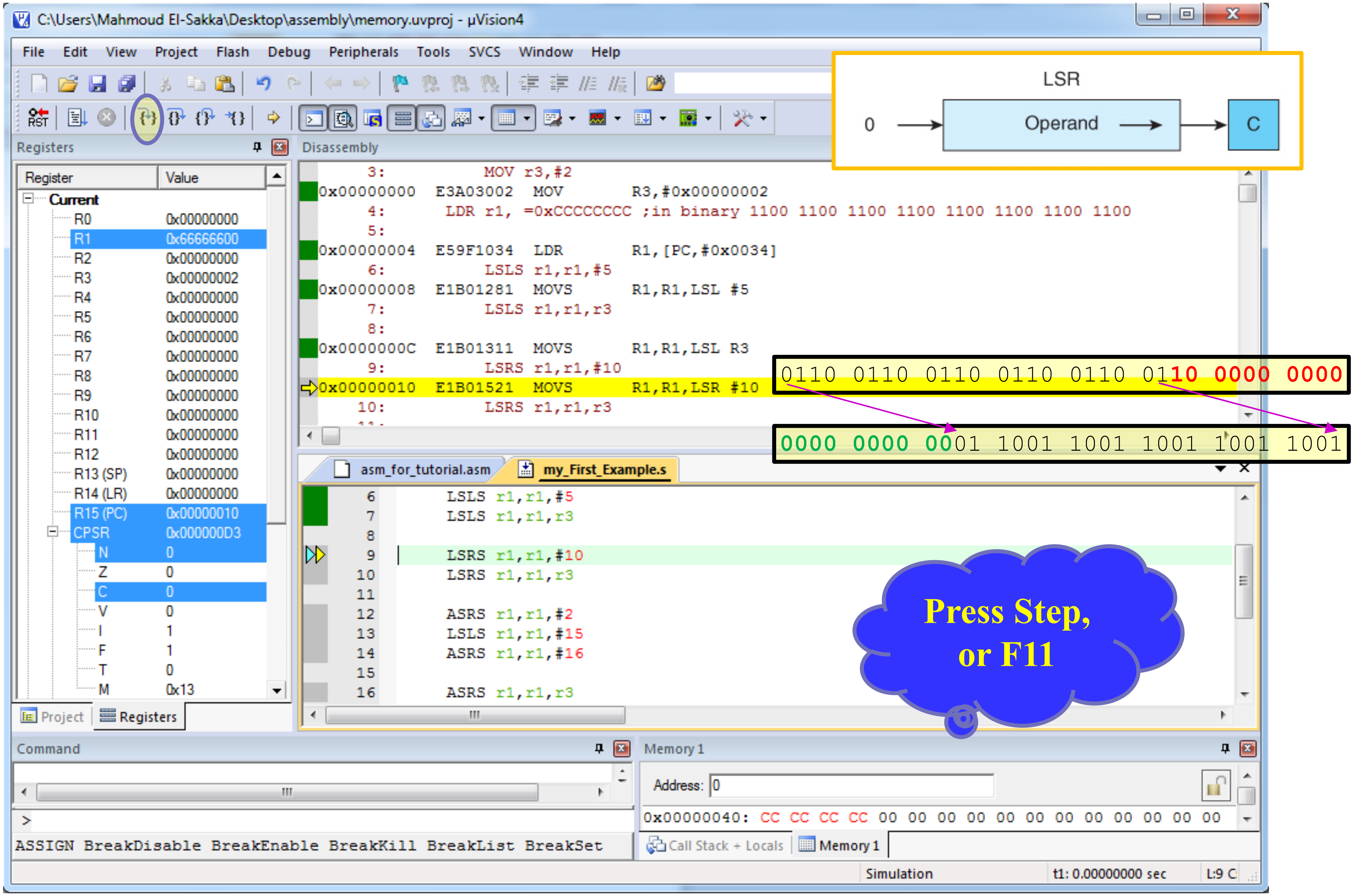


1001 1001 1001 1001 1001 1001 1000 0000

0110 0110 0110 0110 0110 0110 0000 0000

0

ARM's Data-Processing Instructions (Shift Operations)



Registers

Register	Value
Current	
R0	0x00000000
R1	0x66666600
R2	0x00000000
R3	0x00000002
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
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
      3:          MOV r3,#2
0x00000000  E3A03002  MOV      R3,#0x00000002
      4:     LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100
      5:
0x00000004  E59F1034  LDR      R1,[PC,#0x0034]
      6:          LSLS r1,r1,#5
0x00000008  E1B01281  MOVS     R1,R1,LSL #5
      7:          LSLS r1,r1,r3
      8:
0x0000000C  E1B01311  MOVS     R1,R1,LSL R3
      9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS     R1,R1,LSR #10
     10:          LSRS r1,r1,r3
```

```
0110 0110 0110 0110 0110 0110 0000 0000
0000 0000 0001 1001 1001 1001 1001 1001
```

```
        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3
```

Command

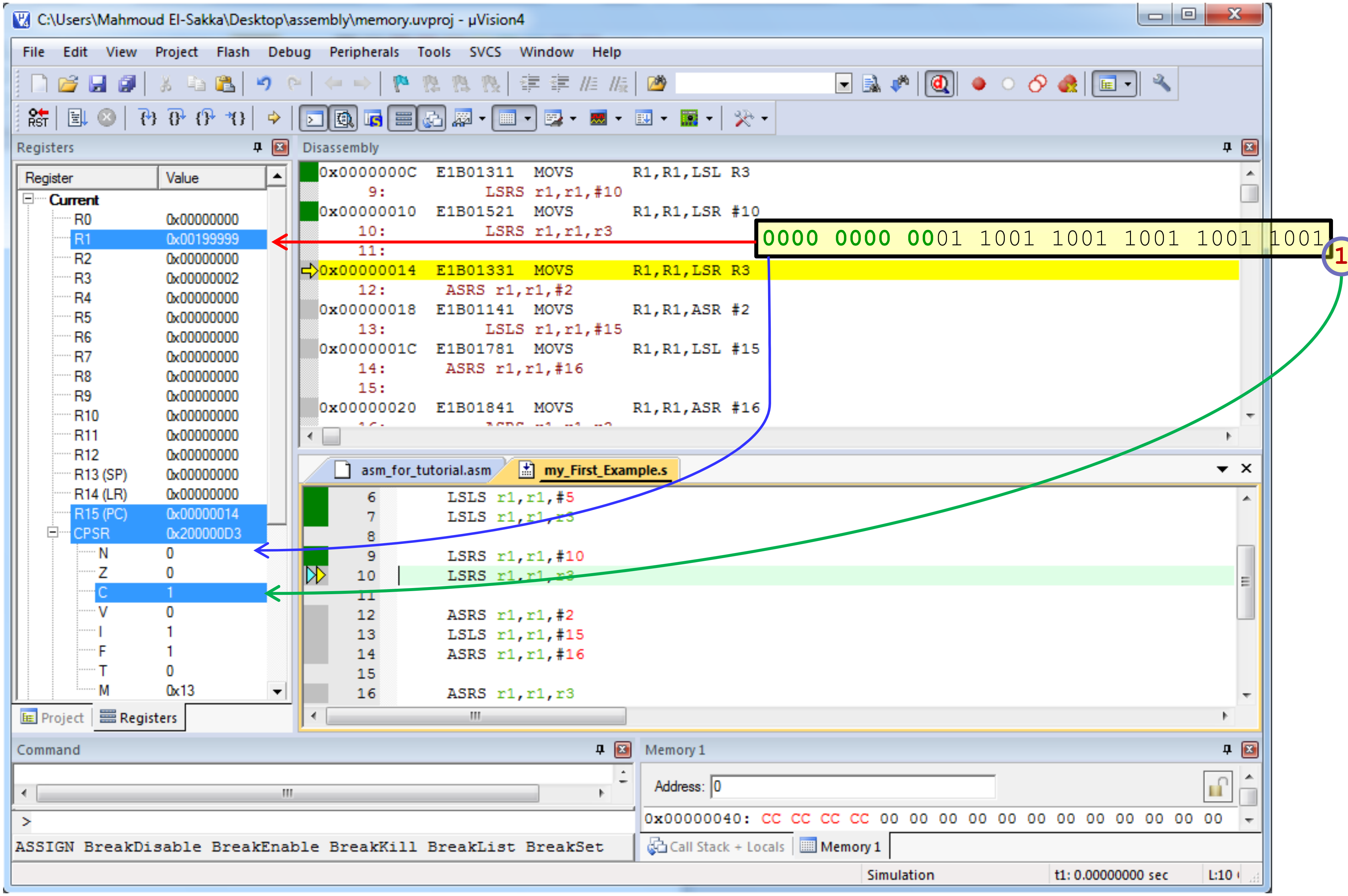
```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0x00199999
R2	0x00000000
R3	0x00000002
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)	0x00000014
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
    11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
    12:      ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

```
0000 0000 0001 1001 1001 1001 1001 1001
```

1

asm_for_tutorial.asm | my_First_Example.s

```
        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3
```

Memory 1
Address: 0

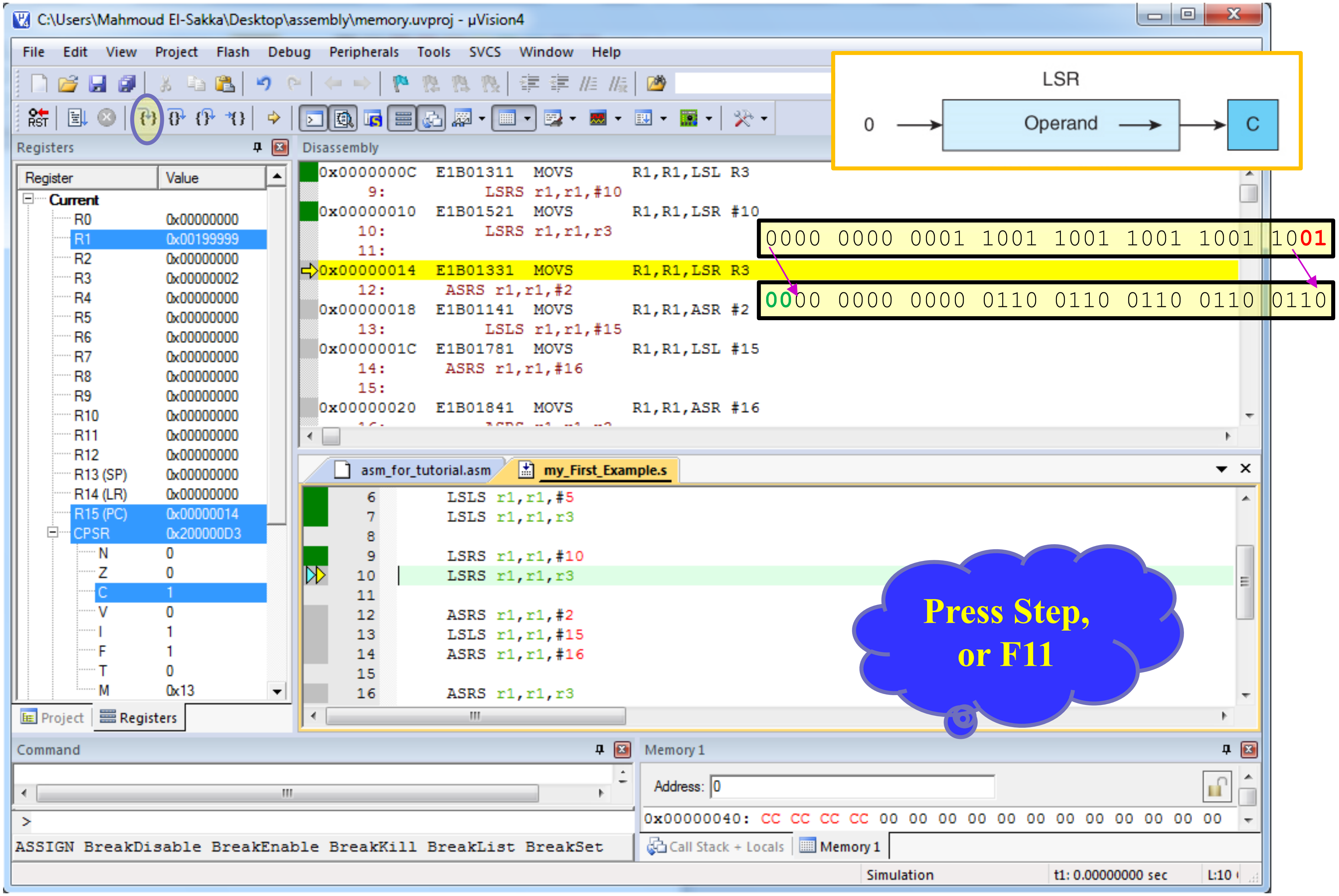
```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Command

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

Simulation t1: 0.00000000 sec L:10

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x00199999
R2	0x00000000
R3	0x00000002
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)	0x00000014
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:           LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:           LSRS r1,r1,r3
    11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
    12:       ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
    13:           LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:       ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

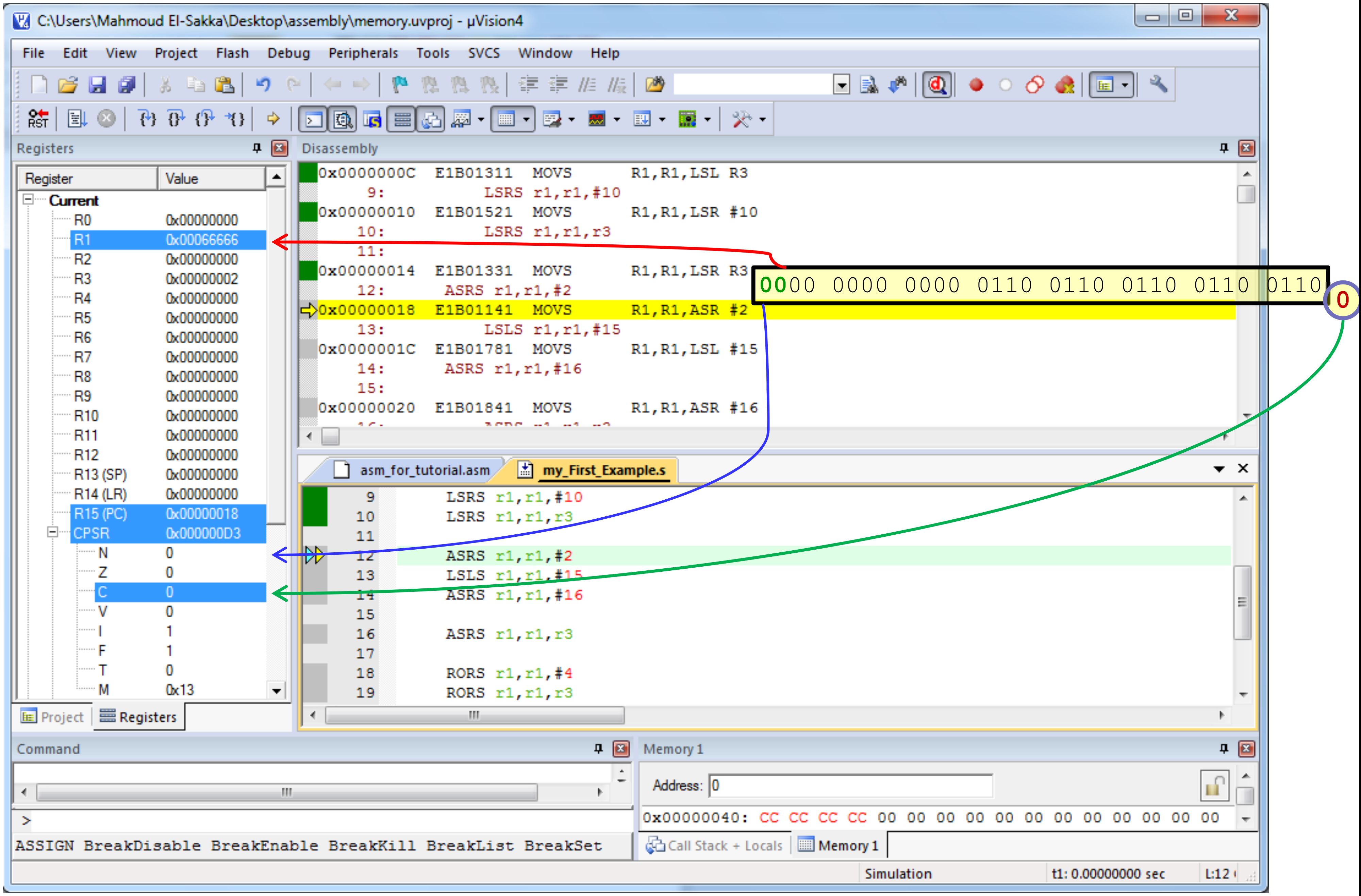
```
0000 0000 0001 1001 1001 1001 1001 1001
0000 0000 0000 0110 0110 0110 0110 0110
```

```
        LSLS r1,r1,#5
        LSLS r1,r1,r3

        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3
```

```
Address: 0
0x00000040:  CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

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

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00066666
R2	0x00000000
R3	0x00000002
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)	0x00000018
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
    11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
    12:      ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

0000 0000 0000 0110 0110 0110 0110 0110 0

asm_for_tutorial.asm | my_First_Example.s

```
        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3
```

Command

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet

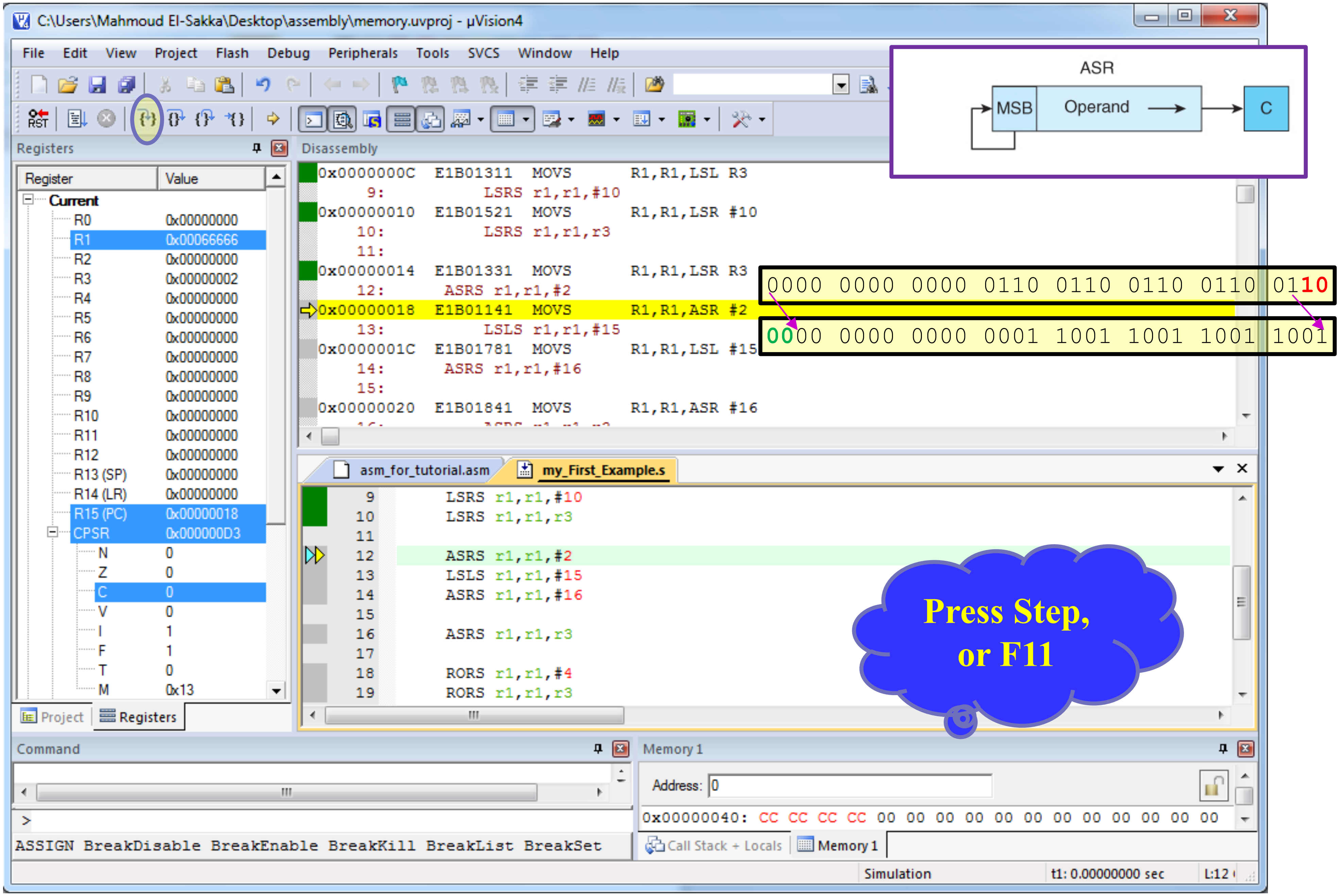
Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation t1: 0.00000000 sec L:12

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
ASR
MSB
Operand
C
Press Step, or F11

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00066666
R2	0x00000000
R3	0x00000002
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)	0x00000018
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
      9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
     10:          LSRS r1,r1,r3
     11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
     12:      ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
     13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
     14:      ASRS r1,r1,#16
     15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

```
0000 0000 0000 0110 0110 0110 0110 0110
0000 0000 0000 0001 1001 1001 1001 1001
```

asm_for_tutorial.asm | my_First_Example.s

```
        LSRS r1,r1,#10
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3
```

Command

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

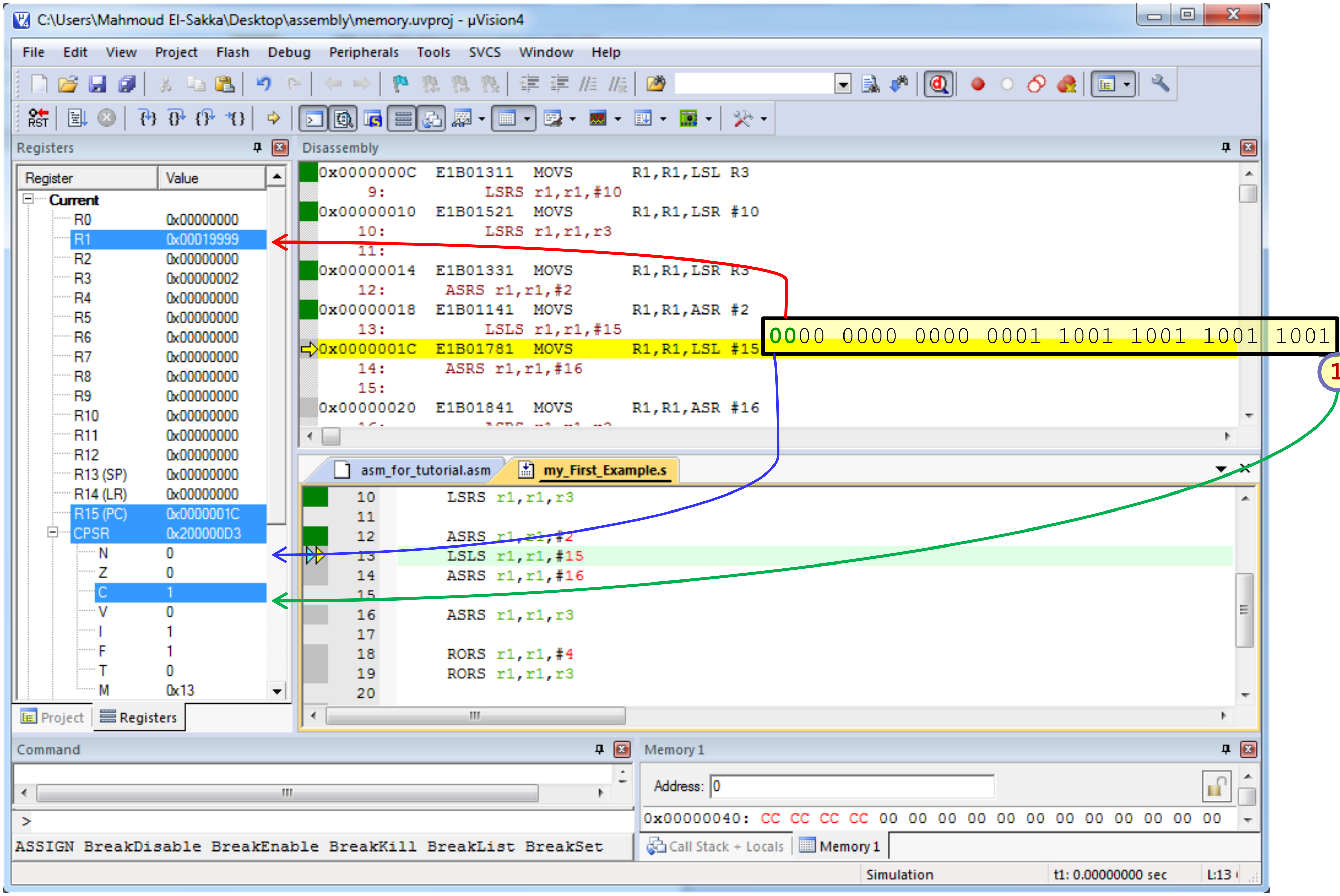
Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:12

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00019999
R2	0x00000000
R3	0x00000002
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)	0x0000001C
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
    11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
    12:      ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

```
0000 0000 0000 0001 1001 1001 1001 1001
```

1

asm_for_tutorial.asm | my_First_Example.s

```
10          LSRS r1,r1,r3
11
12          ASRS r1,r1,#2
13          LSLS r1,r1,#15
14          ASRS r1,r1,#16
15
16          ASRS r1,r1,r3
17
18          RORS r1,r1,#4
19          RORS r1,r1,r3
20
```

Command

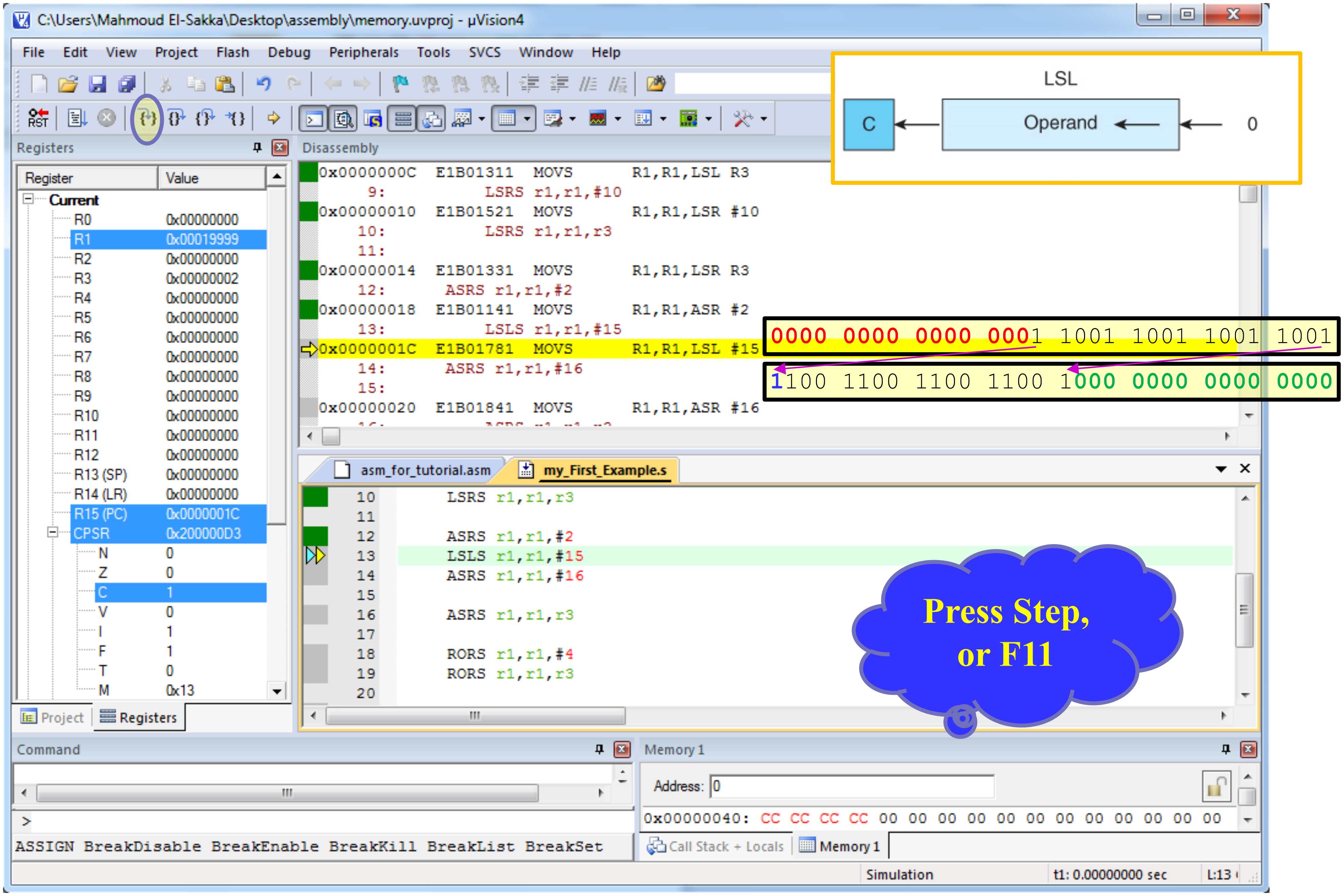
```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x00019999
R2	0x00000000
R3	0x00000002
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)	0x0000001C
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
0x0000000C  E1B01311  MOVS      R1,R1,LSL R3
     9:          LSRS r1,r1,#10
0x00000010  E1B01521  MOVS      R1,R1,LSR #10
    10:          LSRS r1,r1,r3
    11:
0x00000014  E1B01331  MOVS      R1,R1,LSR R3
    12:      ASRS r1,r1,#2
0x00000018  E1B01141  MOVS      R1,R1,ASR #2
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
```

```
0000 0000 0000 0001 1001 1001 1001 1001
1100 1100 1100 1100 1000 0000 0000 0000
```

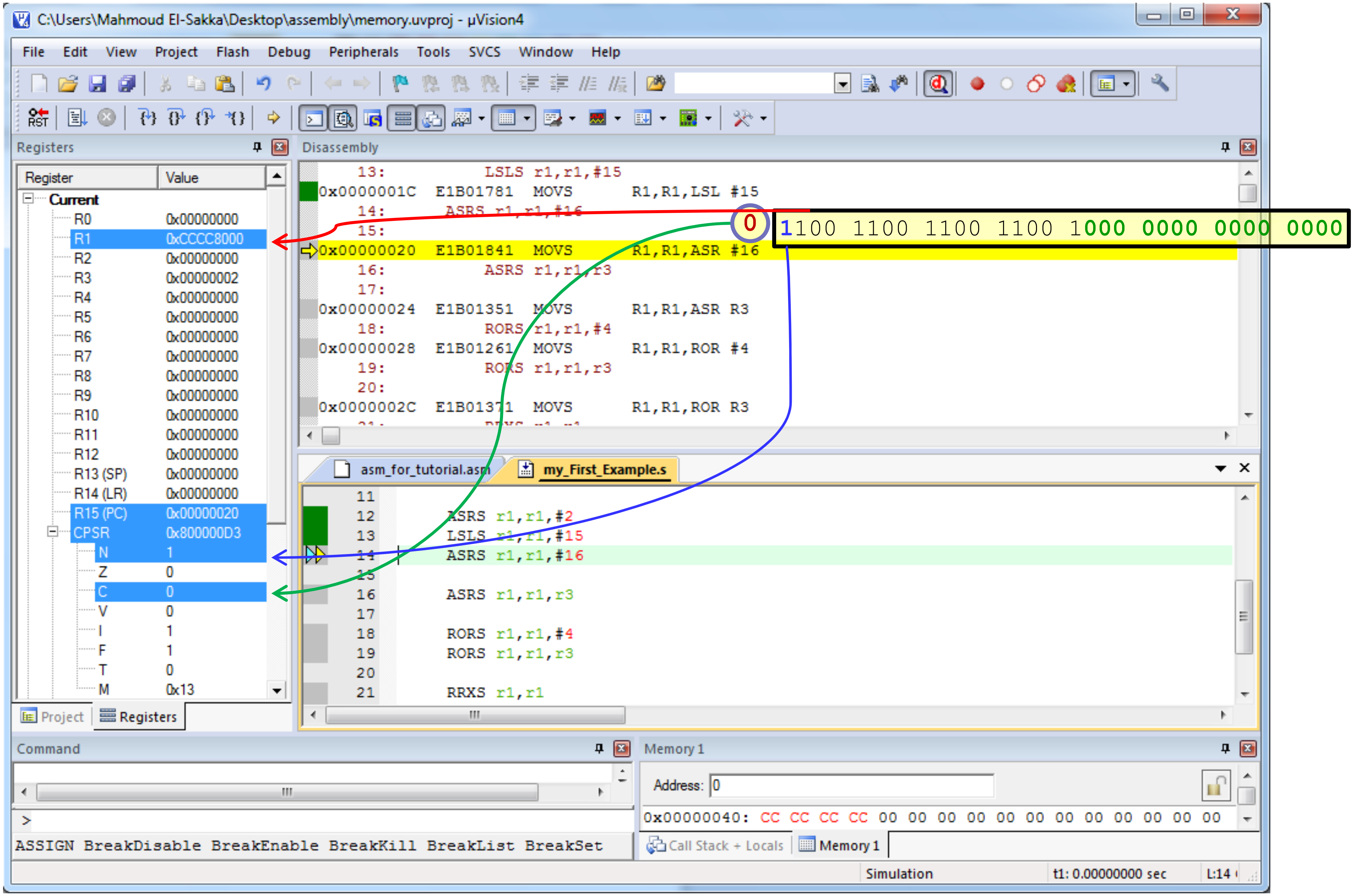
```
        LSRS r1,r1,r3

        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

```

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

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0xCCCC8000
R2	0x00000000
R3	0x00000002
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)	0x00000020
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:          ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:          ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:          RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:          RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

0

1100 1100 1100 1100 1000 0000 0000 0000

asm_for_tutorial.asm   my_First_Example.s

```
        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
```

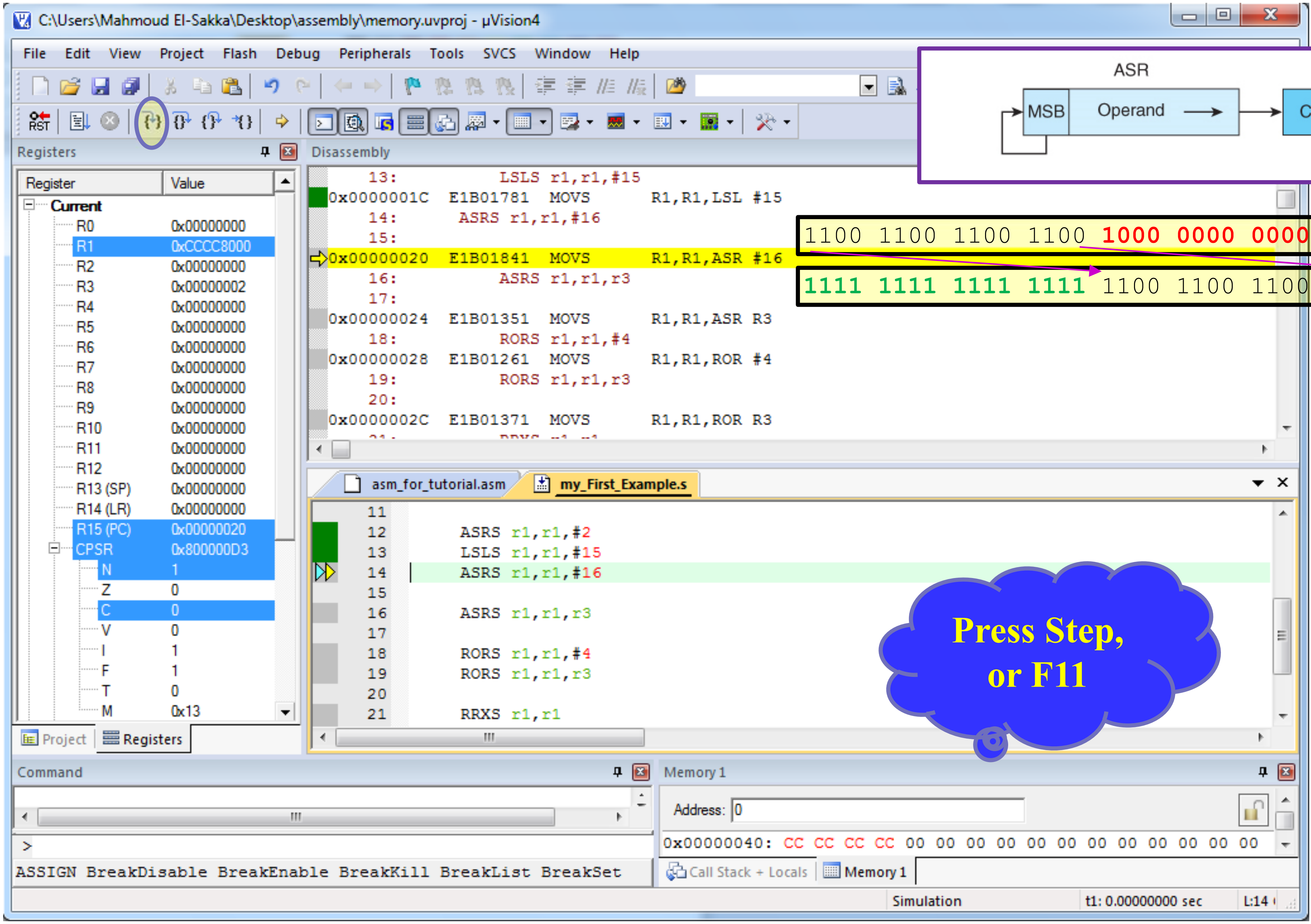
```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

t1: 0.00000000 sec   L:14

ARM's Data-Processing Instructions (Shift Operations)



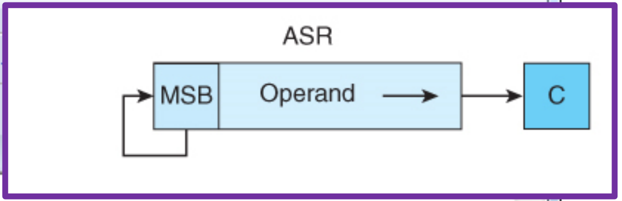
Register	Value
Current	
R0	0x00000000
R1	0xCCCC8000
R2	0x00000000
R3	0x00000002
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000020
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:          ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:          RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:          RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

```
        ASRS r1,r1,#2
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
```

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

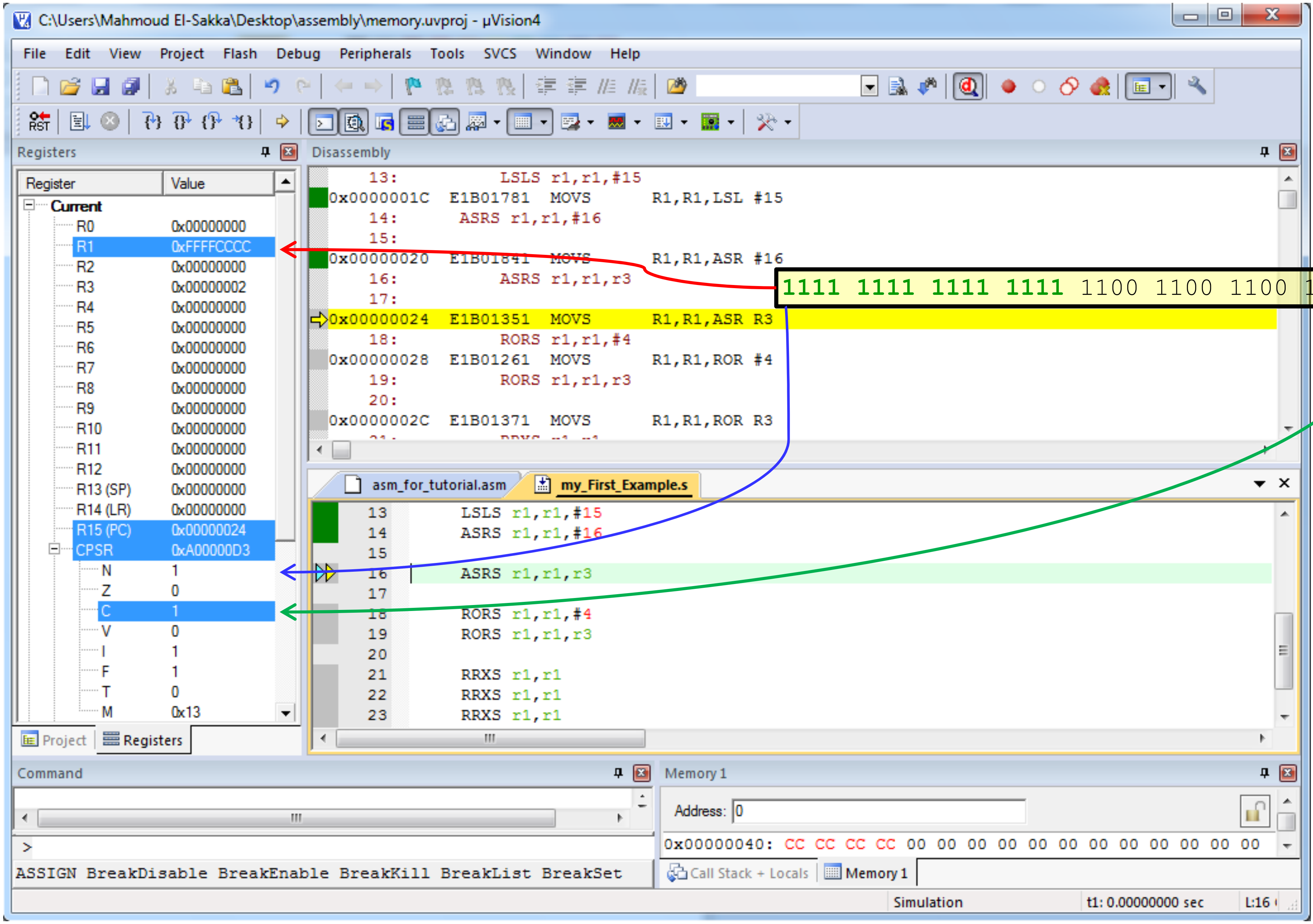
```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```



1100 1100 1100 1100 **1000 0000 0000 0000**

1111 1111 1111 1111 1100 1100 1100 1100

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0xFFFFCCCC
R2	0x00000000
R3	0x00000002
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)	0x00000024
CPSR	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
    13:         LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:     ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:         ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:         RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:         RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

```
1111 1111 1111 1111 1100 1100 1100 1100
```

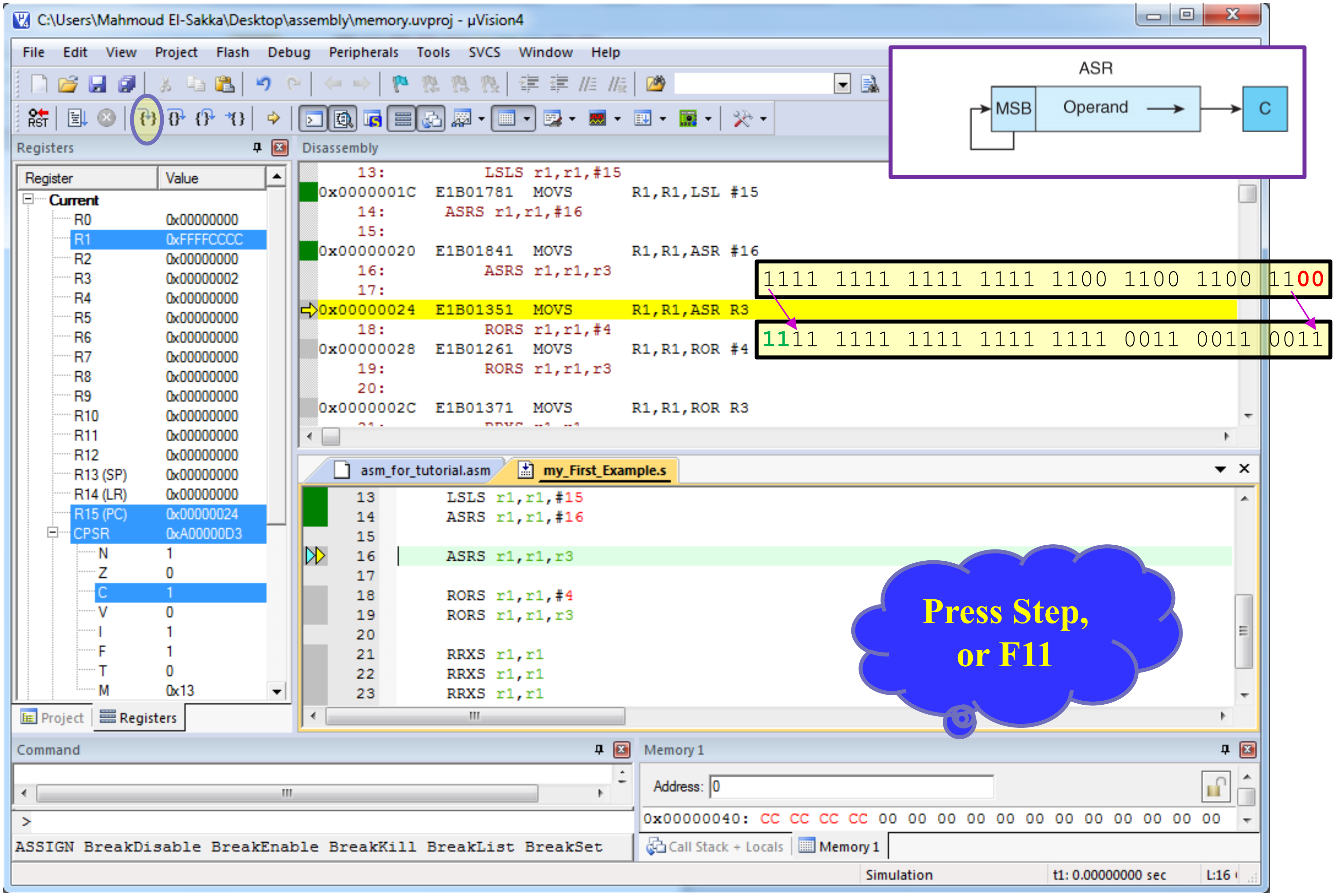
```
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
```

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

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM’s Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0xFFFFCCCC
R2	0x00000000
R3	0x00000002
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)	0x00000024
CPSR	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
    13:           LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:        ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:           ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:           RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:           RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

```
1111 1111 1111 1111 1100 1100 1100 1100
1111 1111 1111 1111 1111 0011 0011 0011
```

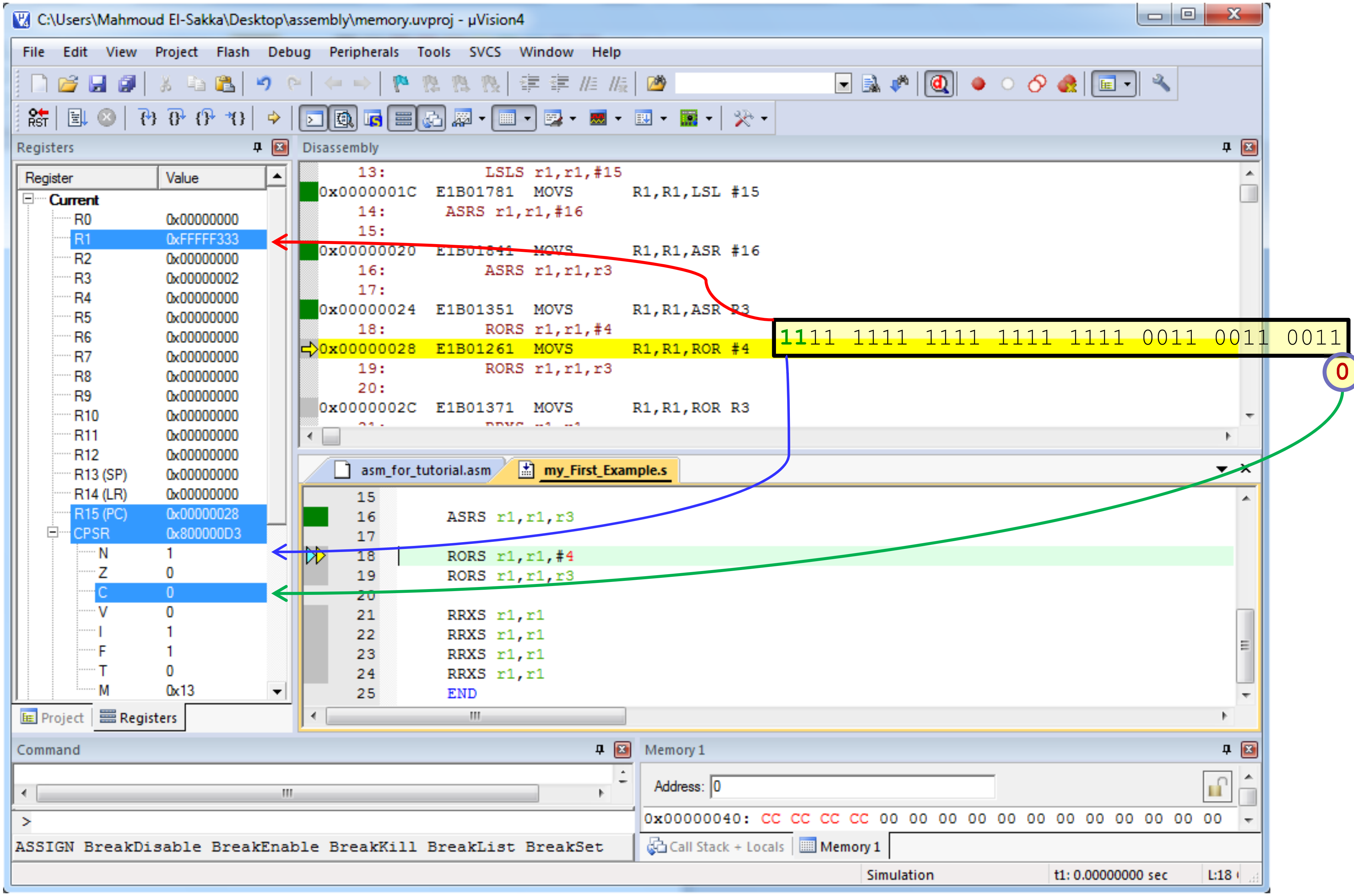
```
        LSLS r1,r1,#15
        ASRS r1,r1,#16

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Simulation    t1: 0.00000000 sec    L:16
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0xFFFFF333
R2	0x00000000
R3	0x00000002
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)	0x00000028
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
    13:         LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:     ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:         ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:         RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:         RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

1111 1111 1111 1111 1111 0011 0011 0011

0

asm_for_tutorial.asm | my_First_Example.s

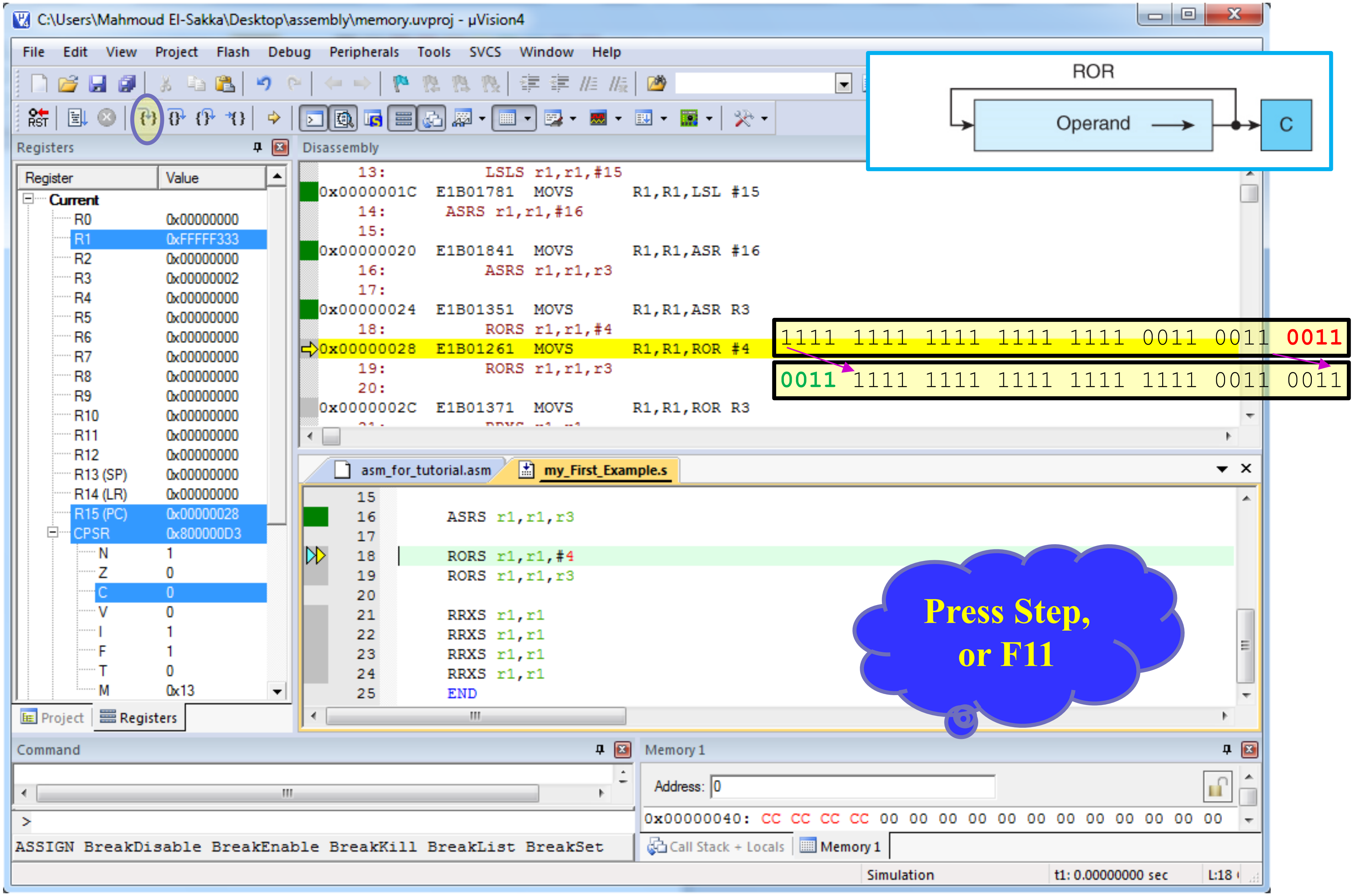
```
     ASRS r1,r1,r3

     RORS r1,r1,#4
     RORS r1,r1,r3

     RRXS r1,r1
     RRXS r1,r1
     RRXS r1,r1
     RRXS r1,r1
     END
```

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

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

Simulation t1: 0.00000000 sec L:18

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0xFFFFF333
R2	0x00000000
R3	0x00000002
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)	0x00000028
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
    13:          LSLS r1,r1,#15
0x0000001C  E1B01781  MOVS      R1,R1,LSL #15
    14:      ASRS r1,r1,#16
    15:
0x00000020  E1B01841  MOVS      R1,R1,ASR #16
    16:          ASRS r1,r1,r3
    17:
0x00000024  E1B01351  MOVS      R1,R1,ASR R3
    18:          RORS r1,r1,#4
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:          RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
```

```
1111 1111 1111 1111 1111 0011 0011 0011
0011 1111 1111 1111 1111 1111 0011 0011
```

```
        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

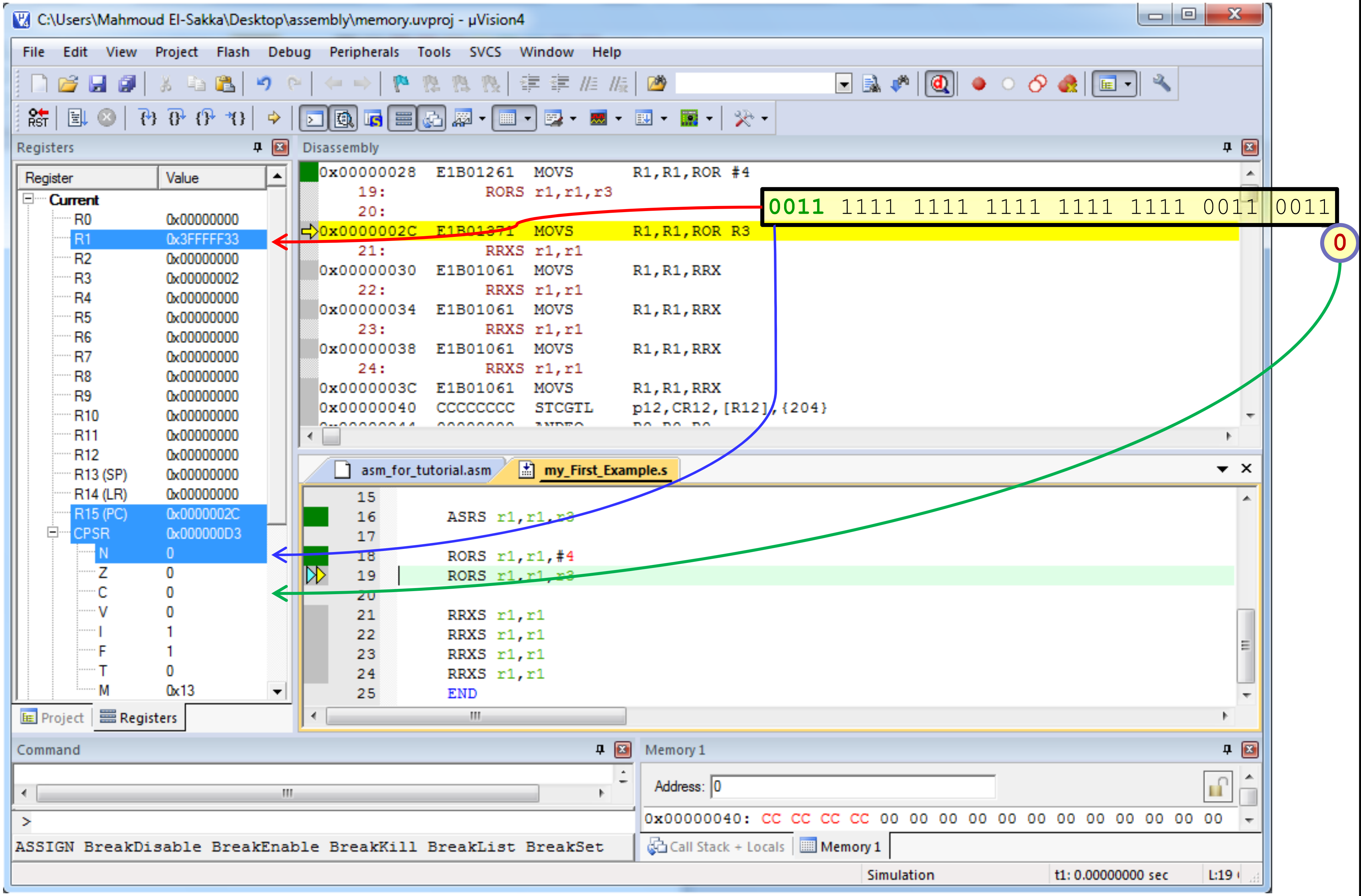
```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000000 sec   L:18

ARM's Data-Processing Instructions (Shift Operations)



0011 1111 1111 1111 1111 1111 0011 0011

Register	Value
Current	
R0	0x00000000
R1	0x3FFFFF33
R2	0x00000000
R3	0x00000002
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)	0x0000002C
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

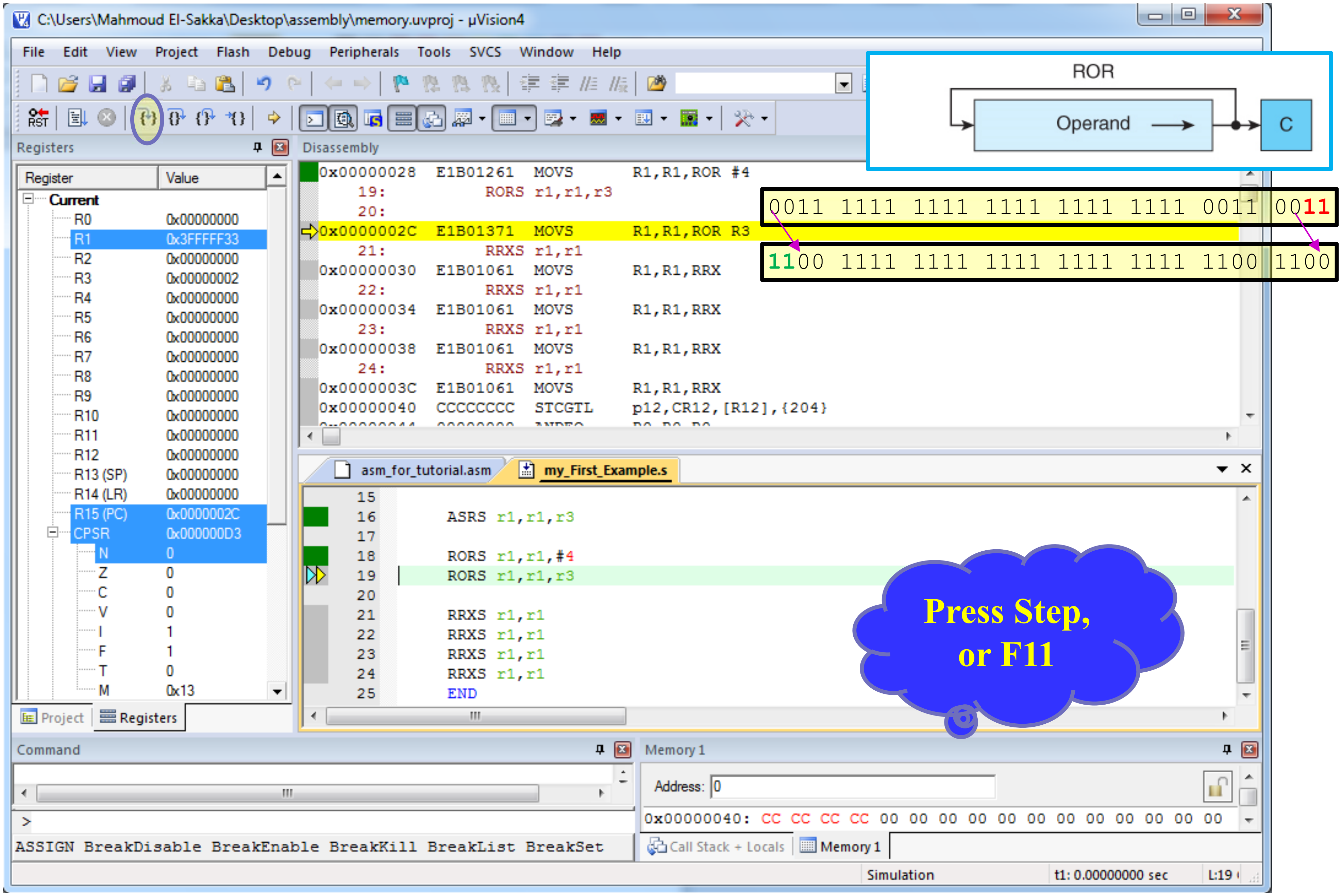
```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:         RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:         RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:         RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:         RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:         RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

```
        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
ROR
Operand
C
Press Step, or F11
Project Registers
Command
Memory 1
Call Stack + Locals
Simulation
t1: 0.00000000 sec
L:19

Register	Value
Current	
R0	0x00000000
R1	0x3FFFFF33
R2	0x00000000
R3	0x00000002
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)	0x0000002C
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
Disassembly
0x00000028  E1B01261  MOVS     R1,R1,ROR #4
    19:          RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS     R1,R1,ROR R3
    21:          RRXS r1,r1
0x00000030  E1B01061  MOVS     R1,R1,RRX
    22:          RRXS r1,r1
0x00000034  E1B01061  MOVS     R1,R1,RRX
    23:          RRXS r1,r1
0x00000038  E1B01061  MOVS     R1,R1,RRX
    24:          RRXS r1,r1
0x0000003C  E1B01061  MOVS     R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL   p12,CR12,[R12],{204}
```

```
0011 1111 1111 1111 1111 1111 0011 0011
1100 1111 1111 1111 1111 1111 1100 1100
```

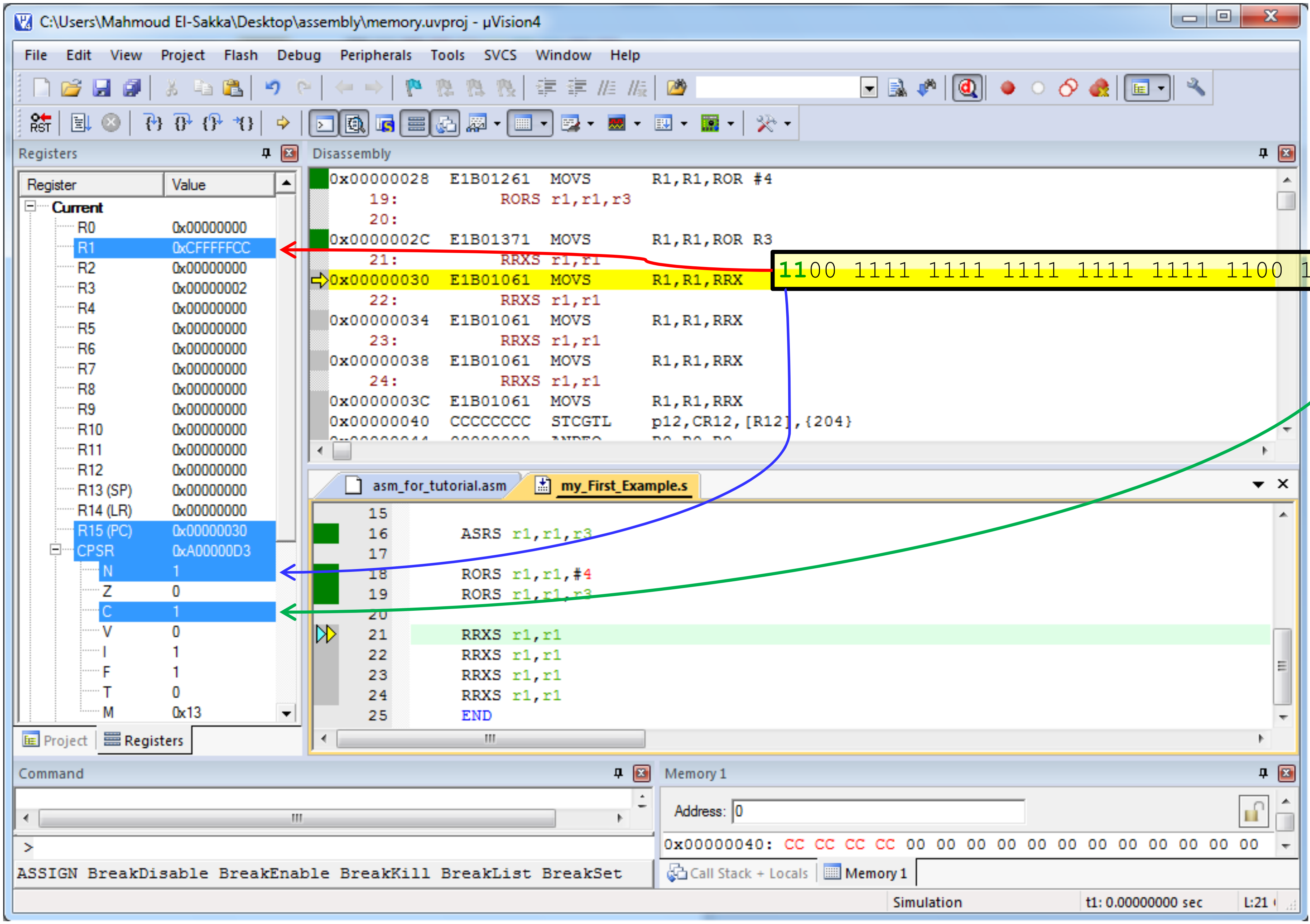
asm_for_tutorial.asm my_First_Example.s

```

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

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

```
Address: 0
0x00000040:  CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM’s Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Register	Value
Current	
R0	0x00000000
R1	0xCFFFFFCC
R2	0x00000000
R3	0x00000002
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)	0x00000030
CPSR	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:           RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:           RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:           RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:           RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:           RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

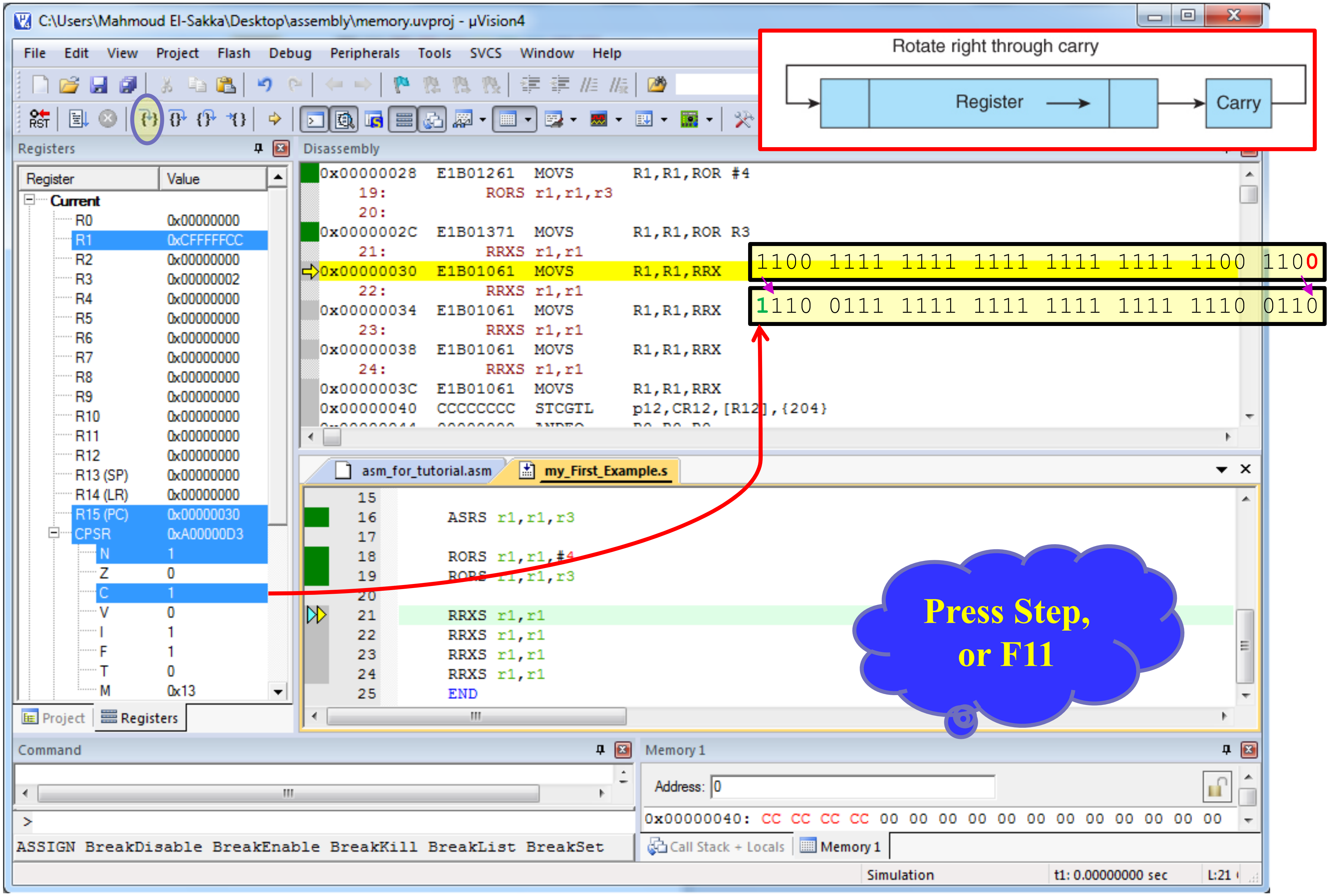
```
1100 1111 1111 1111 1111 1111 1100 1100
```

```

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Rotate right through carry
Register
Carry
Press Step, or F11
asm_for_tutorial.asm
my_First_Example.s
Project
Registers
Command
Memory 1
Call Stack + Locals
Simulation
t1: 0.00000000 sec
L:21

Register	Value
Current	
R0	0x00000000
R1	0xCFFFFFCC
R2	0x00000000
R3	0x00000002
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)	0x00000030
CPSR	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:          RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:          RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:          RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:          RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:          RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

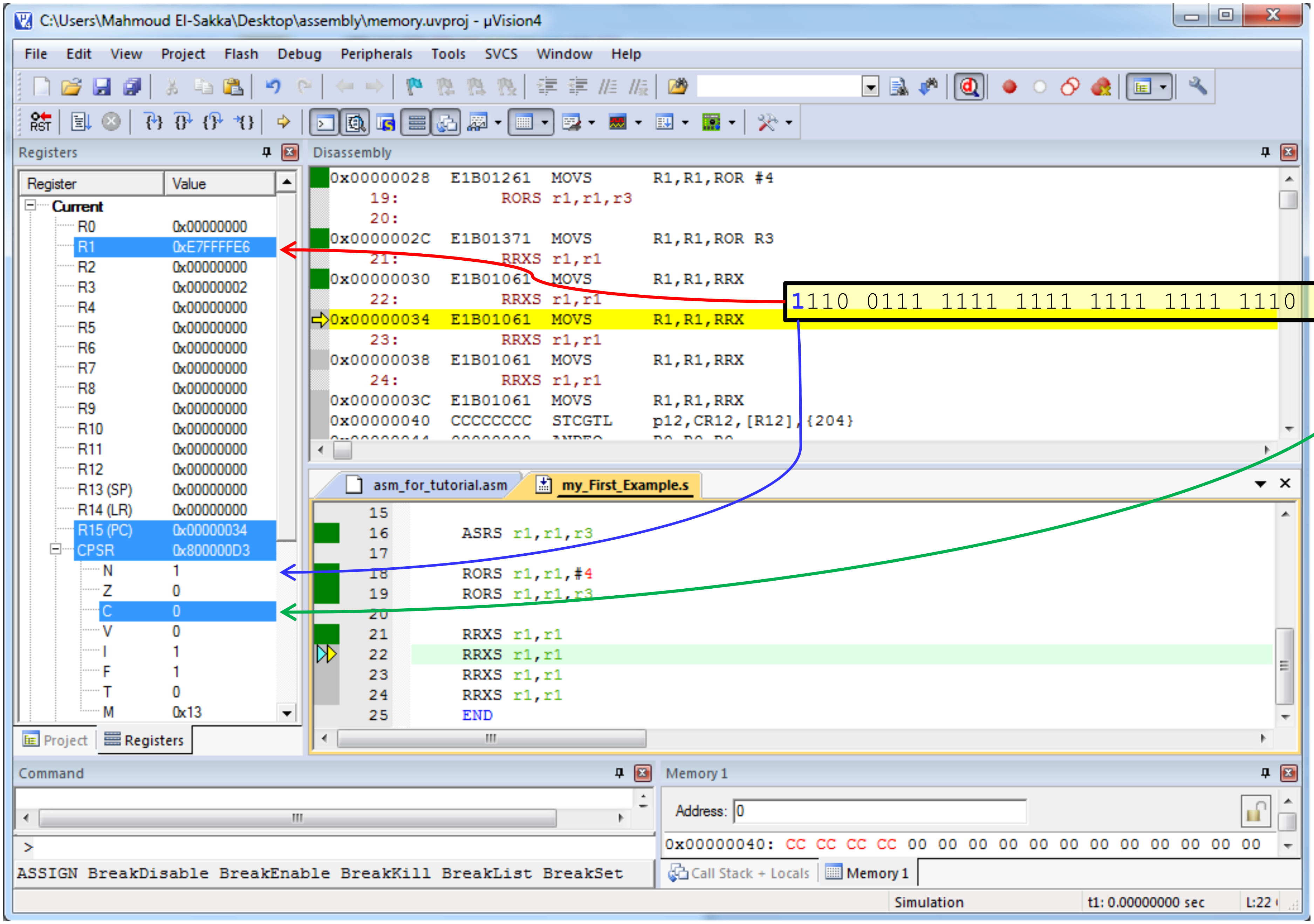
```
1100 1111 1111 1111 1111 1111 1100 1100
1110 0111 1111 1111 1111 1111 1110 0110
```

```

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

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

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
asm_for_tutorial.asm
my_First_Example.s
Project
Registers
Command
Memory 1
Call Stack + Locals
Simulation

Register	Value
Current	
R0	0x00000000
R1	0xE7FFFFE6
R2	0x00000000
R3	0x00000002
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)	0x00000034
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:            RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:            RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:            RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:            RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:            RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

1110 0111 1111 1111 1111 1111 1110 0110

0

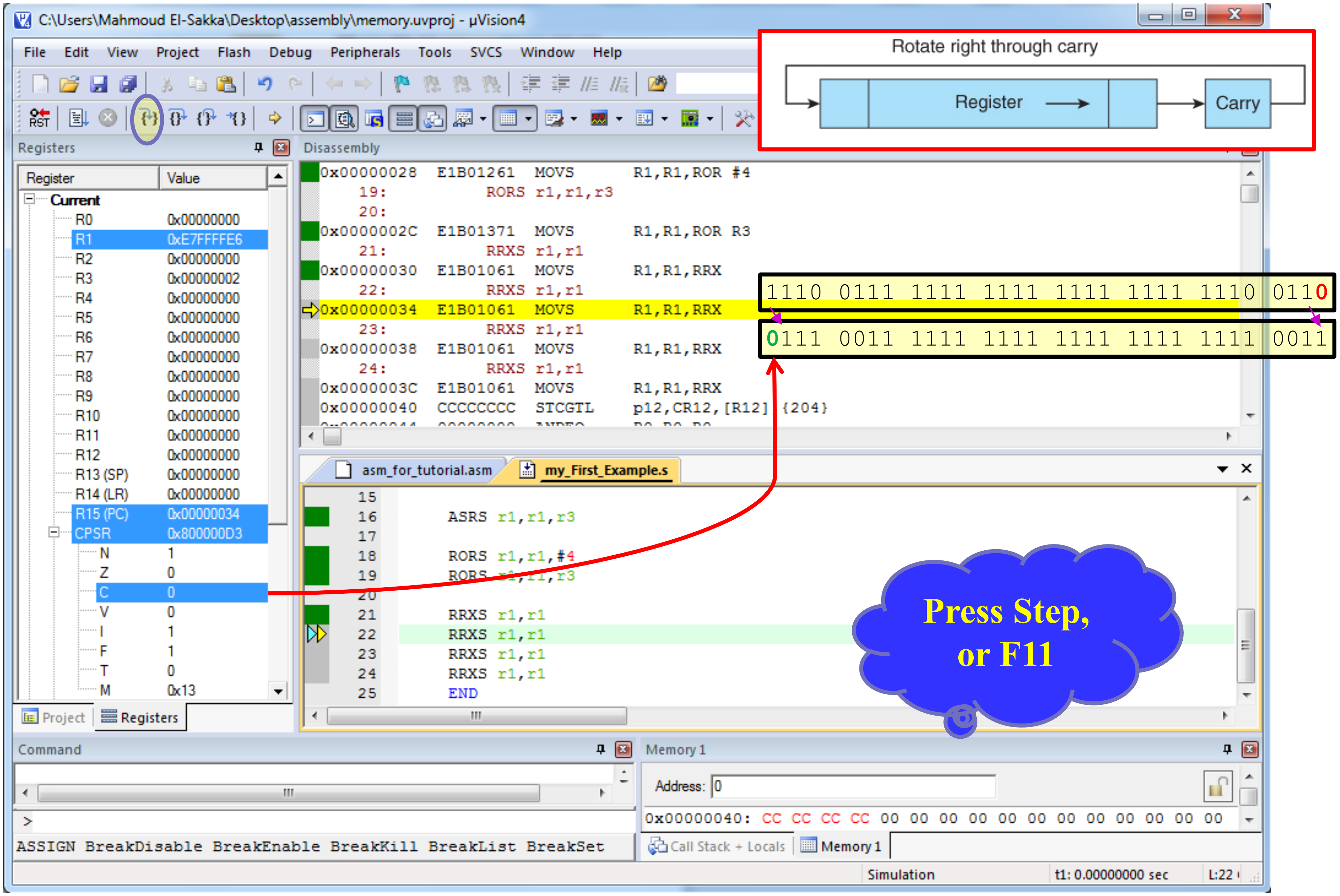
```

        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

```
t1: 0.00000000 sec    L:22
```

ARM's Data-Processing Instructions (Shift Operations)


C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - μVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Rotate right through carry
Register
Carry
Press Step, or F11
Project
Registers
asm_for_tutorial.asm
my_First_Example.s
Call Stack + Locals
Memory 1
Simulation
t1: 0.00000000 sec
L:22

Registers

Register	Value
Current	
R0	0x00000000
R1	0xE7FFFFE6
R2	0x00000000
R3	0x00000002
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)	0x00000034
CPSR	0x800000D3
N	1
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:            RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:            RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:            RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:            RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:            RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

```
1110 0111 1111 1111 1111 1111 1110 0110
0111 0011 1111 1111 1111 1111 1111 0011
```

```
        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

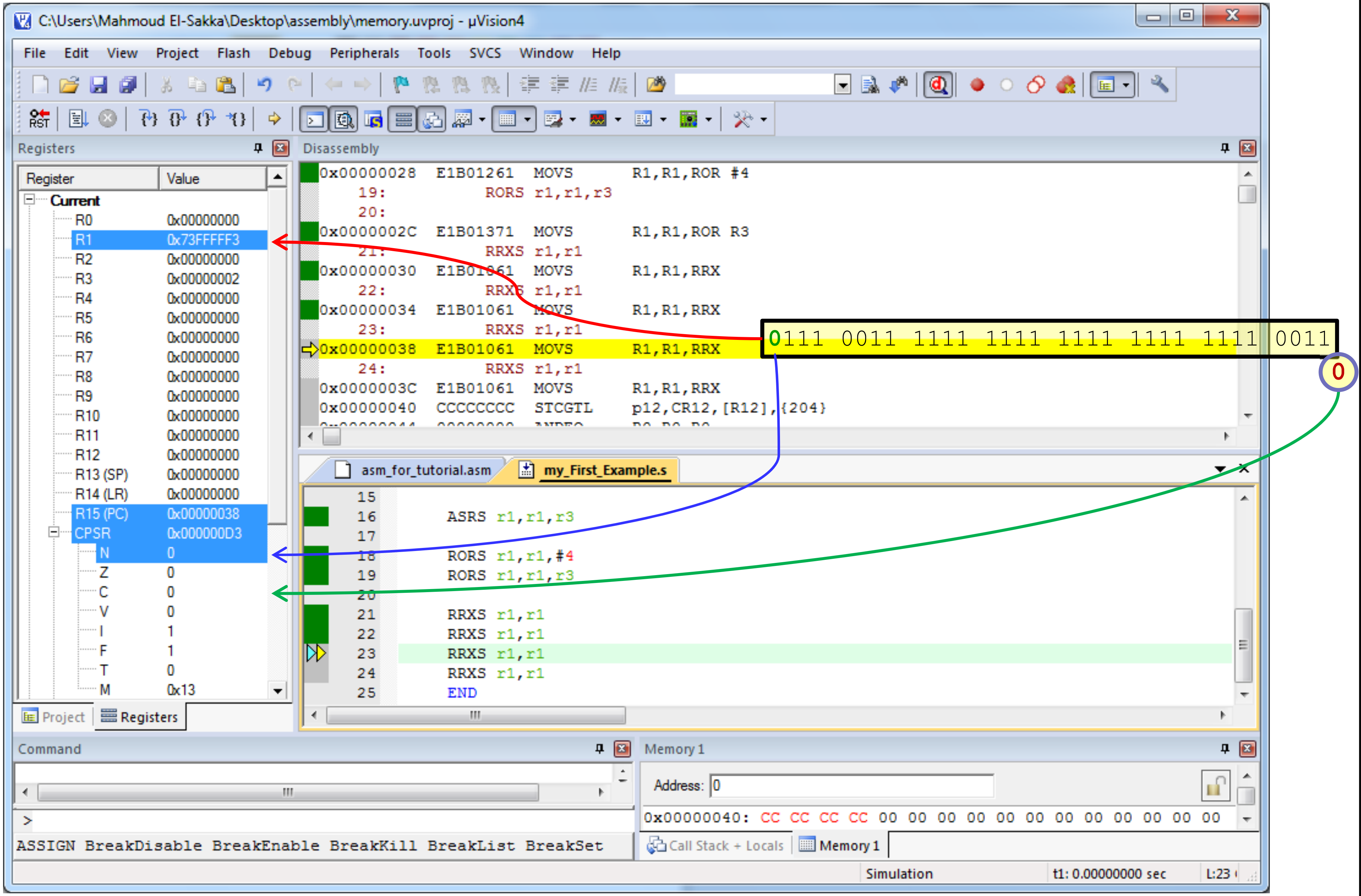
Command

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

Memory 1

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x73FFFFF3
R2	0x00000000
R3	0x00000002
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)	0x00000038
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:             RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:             RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:             RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:             RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:             RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

0111 0011 1111 1111 1111 1111 1111 0011

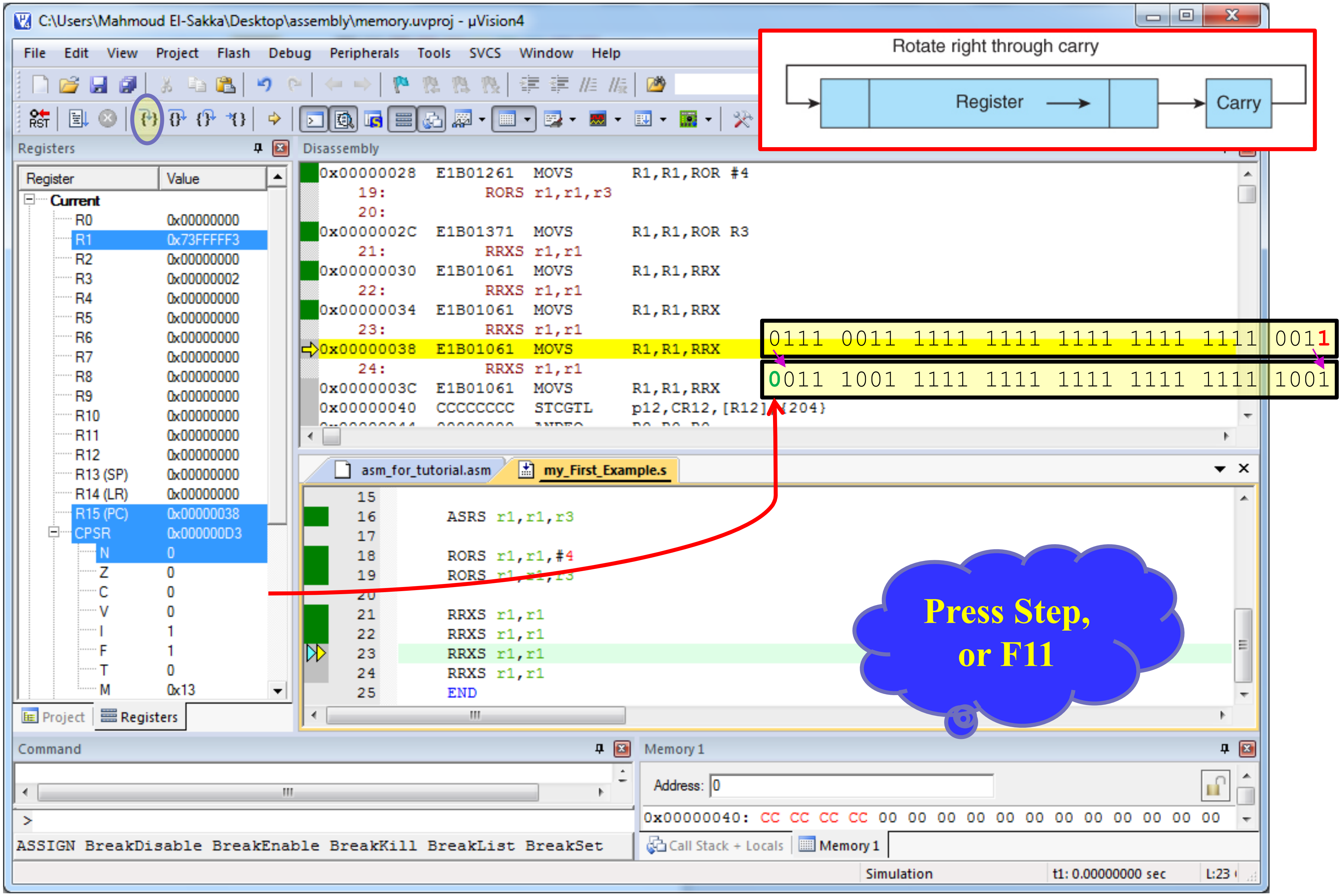
0

```
15
16        ASRS r1,r1,r3
17
18        RORS r1,r1,#4
19        RORS r1,r1,r3
20
21        RRXS r1,r1
22        RRXS r1,r1
23        RRXS r1,r1
24        RRXS r1,r1
25        END
```

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

```
Address: 0
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x73FFFFF3
R2	0x00000000
R3	0x00000002
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)	0x00000038
CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS     R1,R1,ROR #4
    19:           RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS     R1,R1,ROR R3
    21:           RRXS r1,r1
0x00000030  E1B01061  MOVS     R1,R1,RRX
    22:           RRXS r1,r1
0x00000034  E1B01061  MOVS     R1,R1,RRX
    23:           RRXS r1,r1
0x00000038  E1B01061  MOVS     R1,R1,RRX
    24:           RRXS r1,r1
0x0000003C  E1B01061  MOVS     R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL   p12,CR12,[R12],{204}
```

```
0111 0011 1111 1111 1111 1111 1111 0011
0011 1001 1111 1111 1111 1111 1111 1001
```

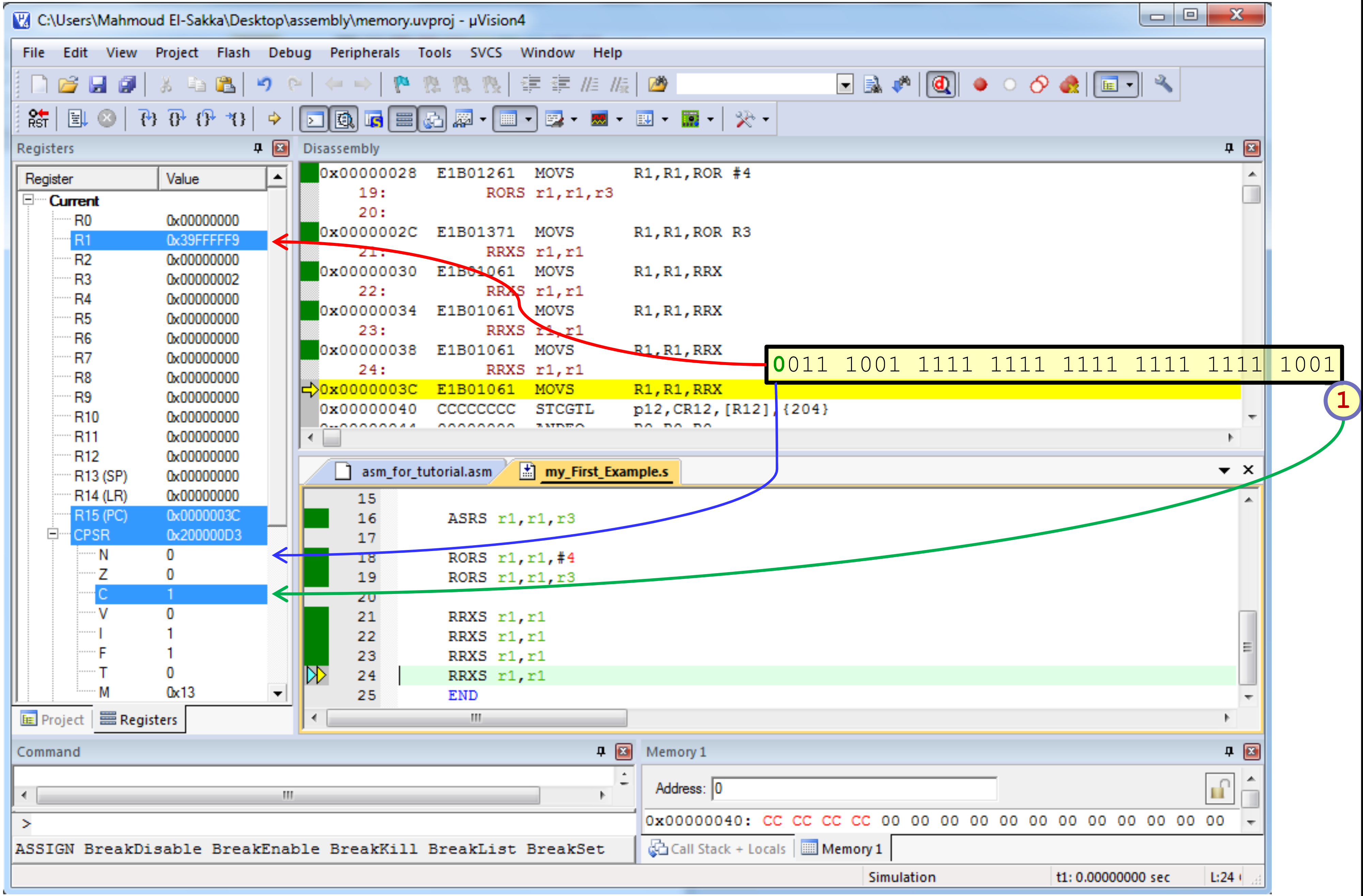
```
        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

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

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



Register	Value
Current	
R0	0x00000000
R1	0x39FFFFF9
R2	0x00000000
R3	0x00000002
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)	0x0000003C
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

```
0x00000028  E1B01261  MOVS     R1,R1,ROR #4
    19:           RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS     R1,R1,ROR R3
    21:           RRXS r1,r1
0x00000030  E1B01061  MOVS     R1,R1,RRX
    22:           RRXS r1,r1
0x00000034  E1B01061  MOVS     R1,R1,RRX
    23:           RRXS r1,r1
0x00000038  E1B01061  MOVS     R1,R1,RRX
    24:           RRXS r1,r1
0x0000003C  E1B01061  MOVS     R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL   p12,CR12,[R12],{204}
```

```
0011 1001 1111 1111 1111 1111 1111 1001
```

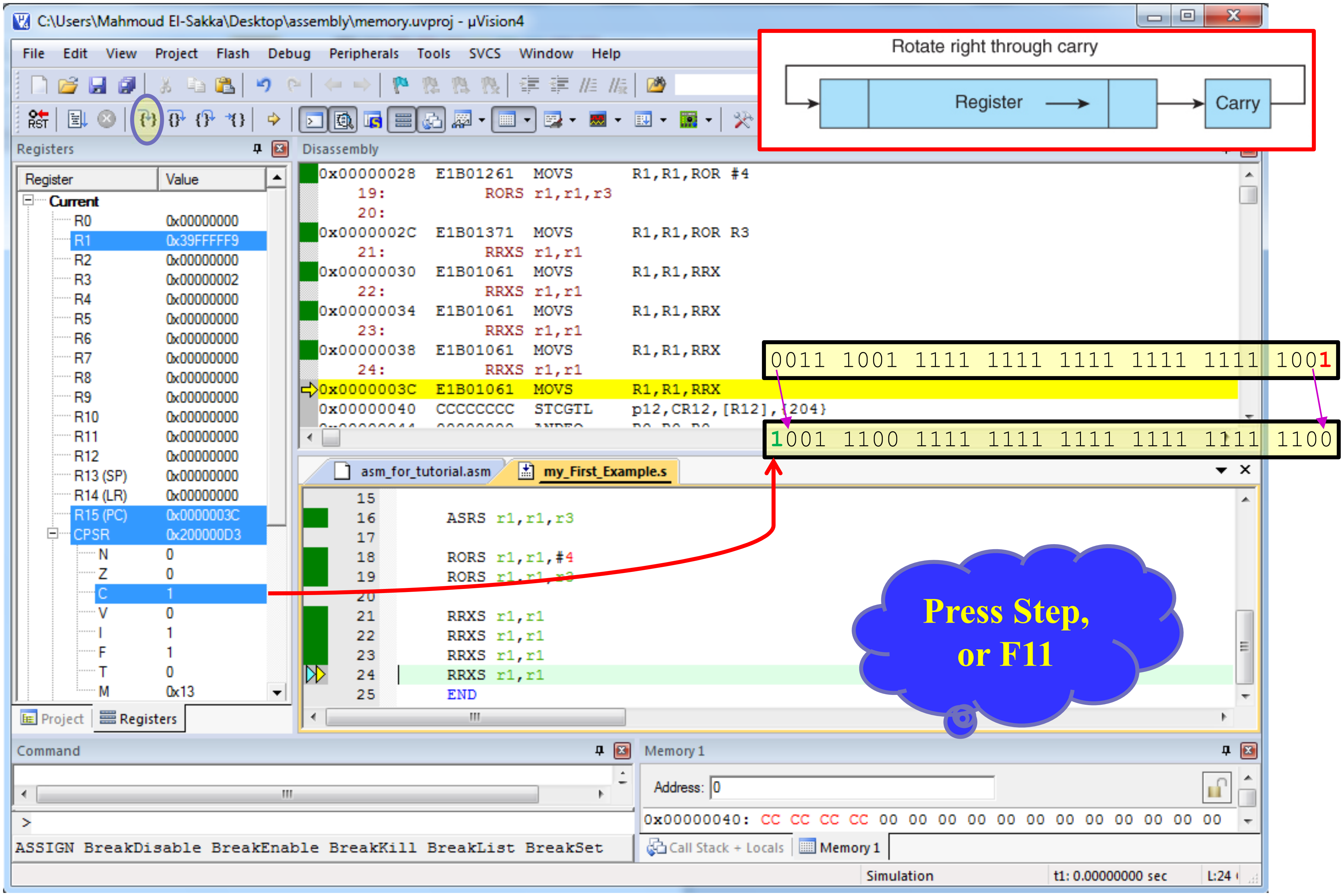
```
15
16      ASRS r1,r1,r3
17
18      RORS r1,r1,#4
19      RORS r1,r1,r3
20
21      RRXS r1,r1
22      RRXS r1,r1
23      RRXS r1,r1
24      RRXS r1,r1
25      END
```

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

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Rotate right through carry
Register
Carry
Press Step, or F11

Registers

Register	Value
Current	
R0	0x00000000
R1	0x39FFFFF9
R2	0x00000000
R3	0x00000002
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)	0x0000003C
CPSR	0x200000D3
N	0
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
0x00000028  E1B01261  MOVS      R1,R1,ROR #4
    19:           RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:           RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:           RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:           RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:           RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
```

```
0011 1001 1111 1111 1111 1111 1111 1001
1001 1100 1111 1111 1111 1111 1111 1100
```

asm_for_tutorial.asm | my_First_Example.s

```

    ASRS r1,r1,r3

    RORS r1,r1,#4
    RORS r1,r1,r3

    RRXS r1,r1
    RRXS r1,r1
    RRXS r1,r1
    RRXS r1,r1
    END
```

Command

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

Memory 1

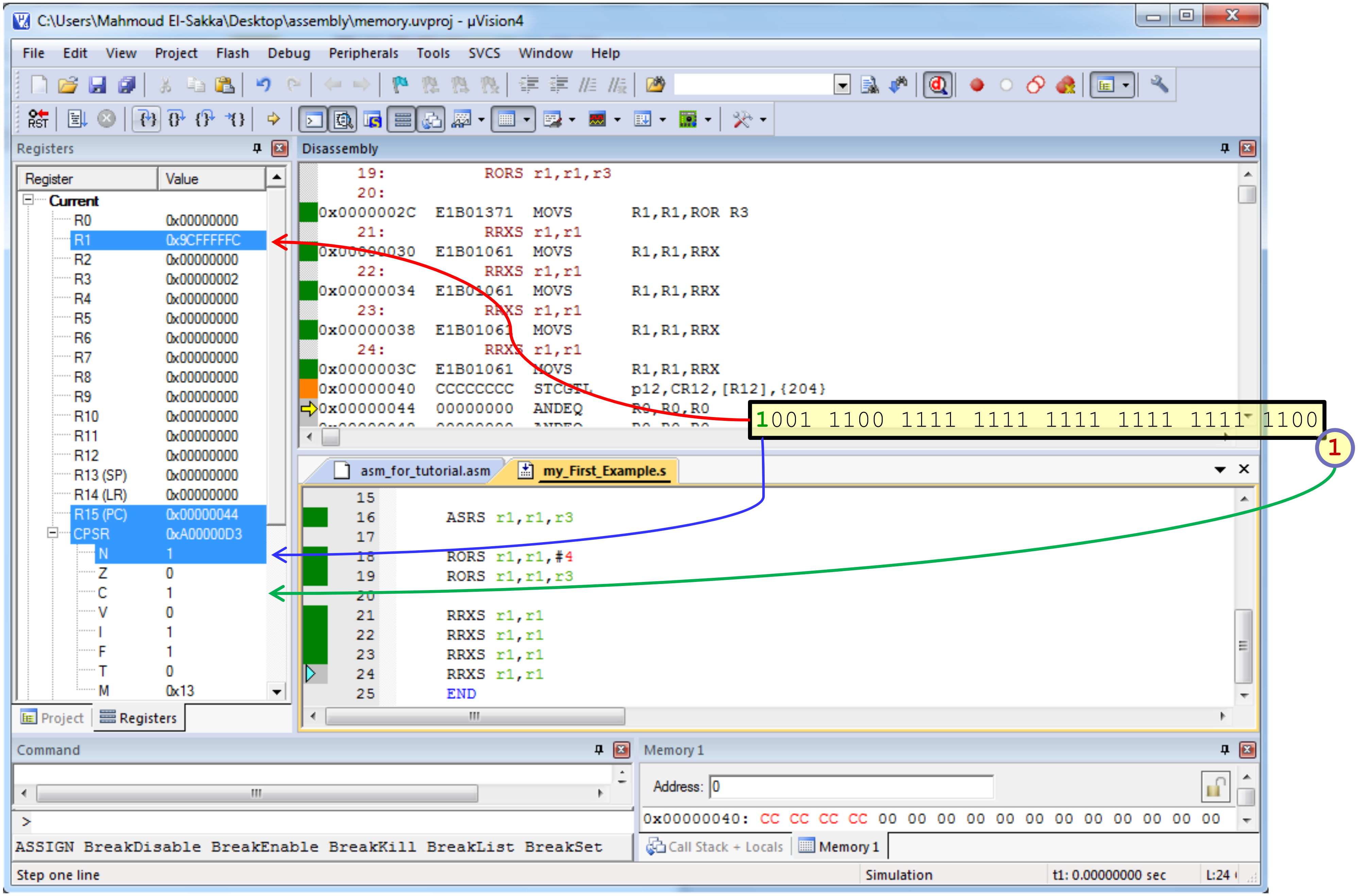
Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Call Stack + Locals | Memory 1

Simulation   t1: 0.00000000 sec   L:24

ARM's Data-Processing Instructions (Shift Operations)



C:\Users\Mahmoud El-Sakka\Desktop\assembly\memory.uvproj - µVision4

Registers

Register	Value
Current	
R0	0x00000000
R1	0x9CFFFFFC
R2	0x00000000
R3	0x00000002
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)	0x00000044
CPSR	0xA00000D3
N	1
Z	0
C	1
V	0
I	1
F	1
T	0
M	0x13

Disassembly

```
    19:         RORS r1,r1,r3
    20:
0x0000002C  E1B01371  MOVS      R1,R1,ROR R3
    21:         RRXS r1,r1
0x00000030  E1B01061  MOVS      R1,R1,RRX
    22:         RRXS r1,r1
0x00000034  E1B01061  MOVS      R1,R1,RRX
    23:         RRXS r1,r1
0x00000038  E1B01061  MOVS      R1,R1,RRX
    24:         RRXS r1,r1
0x0000003C  E1B01061  MOVS      R1,R1,RRX
0x00000040  CCCCCCCC  STCGTL    p12,CR12,[R12],{204}
0x00000044  00000000  ANDEQ     R0,R0,R0
```

1001 1100 1111 1111 1111 1111 1111 1100

1

asm_for_tutorial.asm | my_First_Example.s

```
        ASRS r1,r1,r3

        RORS r1,r1,#4
        RORS r1,r1,r3

        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        RRXS r1,r1
        END
```

Memory 1

Address: 0

```
0x00000040: CC CC CC CC 00 00 00 00 00 00 00 00 00 00 00 00
```

Command

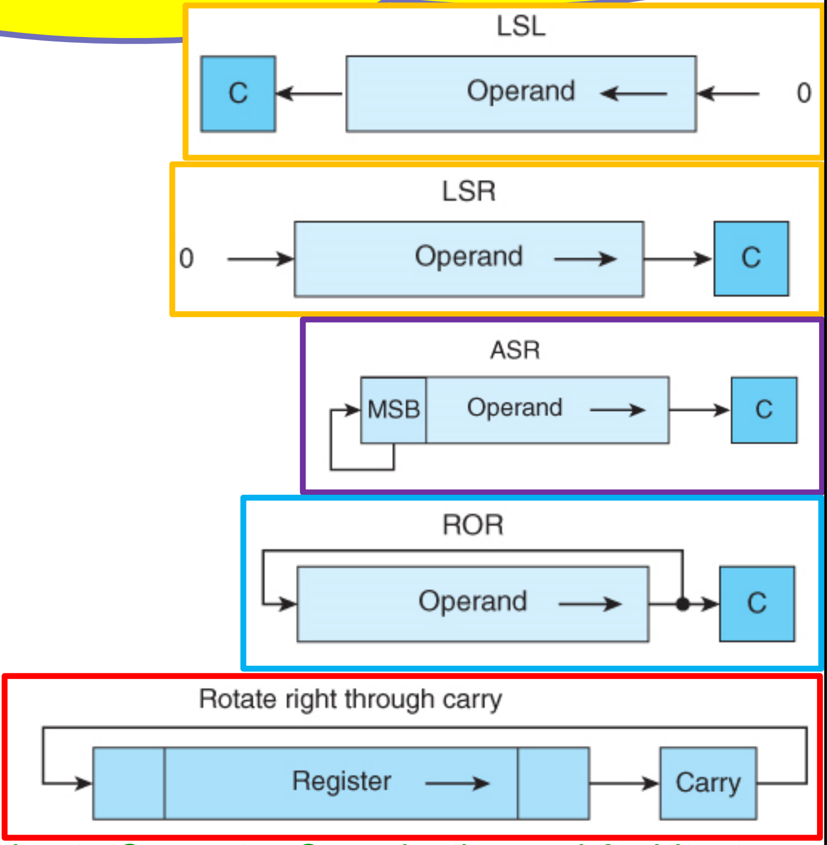
```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Step one line | Simulation | t1: 0.00000000 sec | L:24

ARM's Data-Processing Instructions (Shift Operations)

```
AREA prog1, code, READONLY
ENTRY
MOV r3,#2
LDR r1, =0xCCCCCCCC ;in binary 1100 1100 1100 1100 1100 1100 1100 1100

LSL r1,r1,#5
LSL r1,r1,r3

LSR r1,r1,#10
LSR r1,r1,r3

ASR r1,r1,#2
LSL r1,r1,#15
ASR r1,r1,#16

ASR r1,r1,r3

ROR r1,r1,#4
ROR r1,r1,r3

RRX r1,r1
RRX r1,r1
RRX r1,r1
RRX r1,r1
END
```

Repeat the example again without the "S"


LSL
C
Operand
0
LSR
0
Operand
C
ASR
MSB
Operand
C
ROR
Operand
C
Rotate right through carry
Register
Carry