EE3002/IM2002

Microprocessors

Tutorial 8

1. State the validity for each of the following instructions. Explain why for those that is not valid.

   a. STM r4, {r4-r6}
   b. LDM r4, {r4, r7}
   c. LDM r4!, {r4, r5}
   d. STMIA r5!, {r5, r4, r9}
   e. LDMDA r2, {}
   f. STMDB r15!, {r0-r3, r4, lr}

a. STM r4, {r4-r6} – valid

b. LDM r4, {r4, r7} – valid

c. LDM r4!, {r4, r5}
With this register combination, writeback results in UNPREDICTABLE behavior

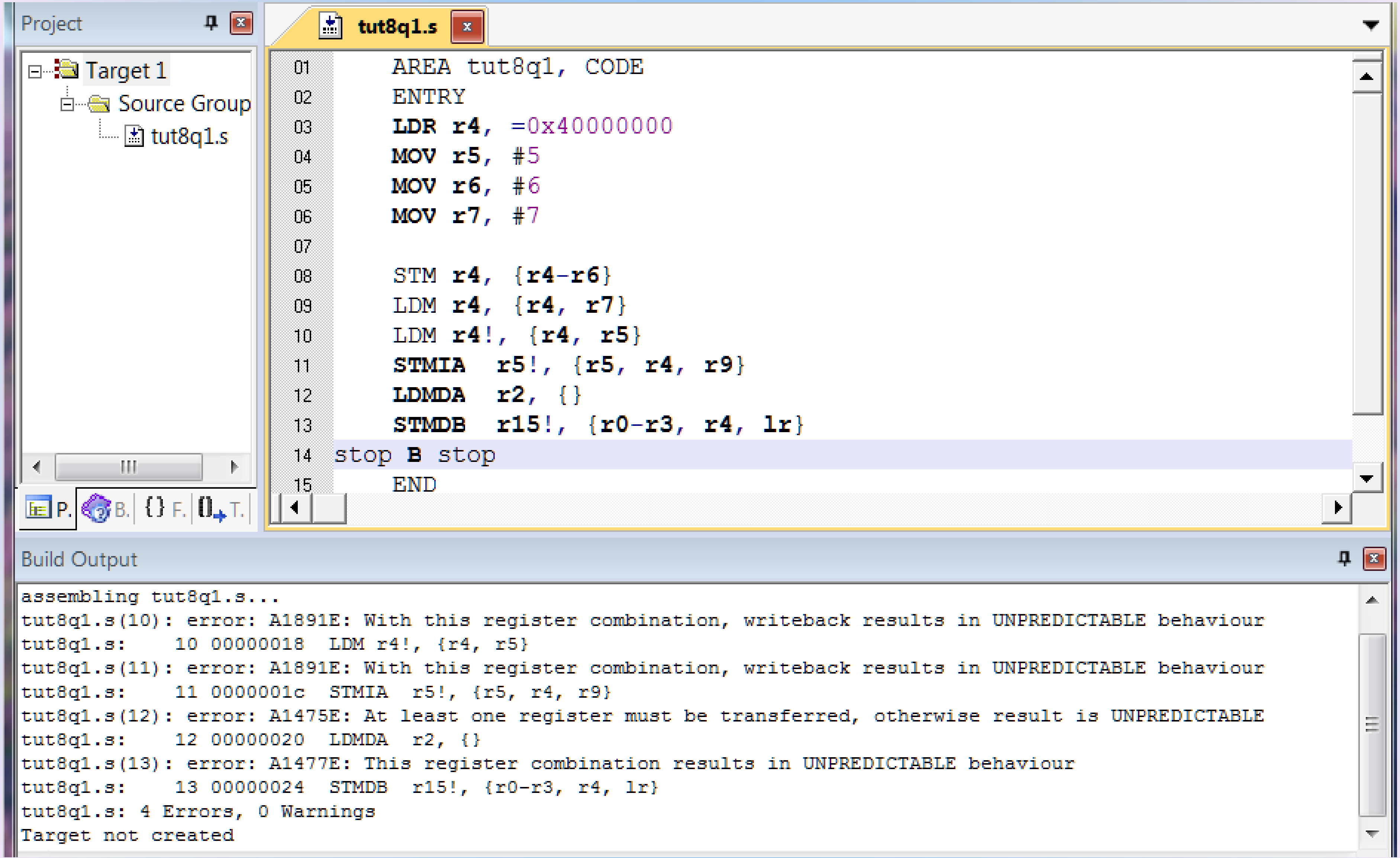
d. STMIA r5!, {r5, r4, r9}
Register r5 is included in the register list, as well as being used as a pointer. The results are unpredictable when write-back is used in such a situation.

e. LDMDA r2, {}

At least one register must be transferred, otherwise result is unpredictable

f. STMDB r15!, {r0-r3, r4, lr}

Using the program counter as the base pointer gives unpredictable results



```
01      AREA tut8q1, CODE
02      ENTRY
03      LDR r4, =0x40000000
04      MOV r5, #5
05      MOV r6, #6
06      MOV r7, #7
07
08      STM r4, {r4-r6}
09      LDM r4, {r4, r7}
10      LDM r4!, {r4, r5}
11      STMIA  r5!, {r5, r4, r9}
12      LDMDA  r2, {}
13      STMDB  r15!, {r0-r3, r4, lr}
14  stop B stop
15      END
```

```
assembling tut8q1.s...
tut8q1.s(10): error: A1891E: With this register combination, writeback results in UNPREDICTABLE behaviour
tut8q1.s:    10 00000018  LDM r4!, {r4, r5}
tut8q1.s(11): error: A1891E: With this register combination, writeback results in UNPREDICTABLE behaviour
tut8q1.s:    11 0000001c  STMIA  r5!, {r5, r4, r9}
tut8q1.s(12): error: A1475E: At least one register must be transferred, otherwise result is UNPREDICTABLE
tut8q1.s:    12 00000020  LDMDA  r2, {}
tut8q1.s(13): error: A1477E: This register combination results in UNPREDICTABLE behaviour
tut8q1.s:    13 00000024  STMDB  r15!, {r0-r3, r4, lr}
tut8q1.s: 4 Errors, 0 Warnings
Target not created
```

2.

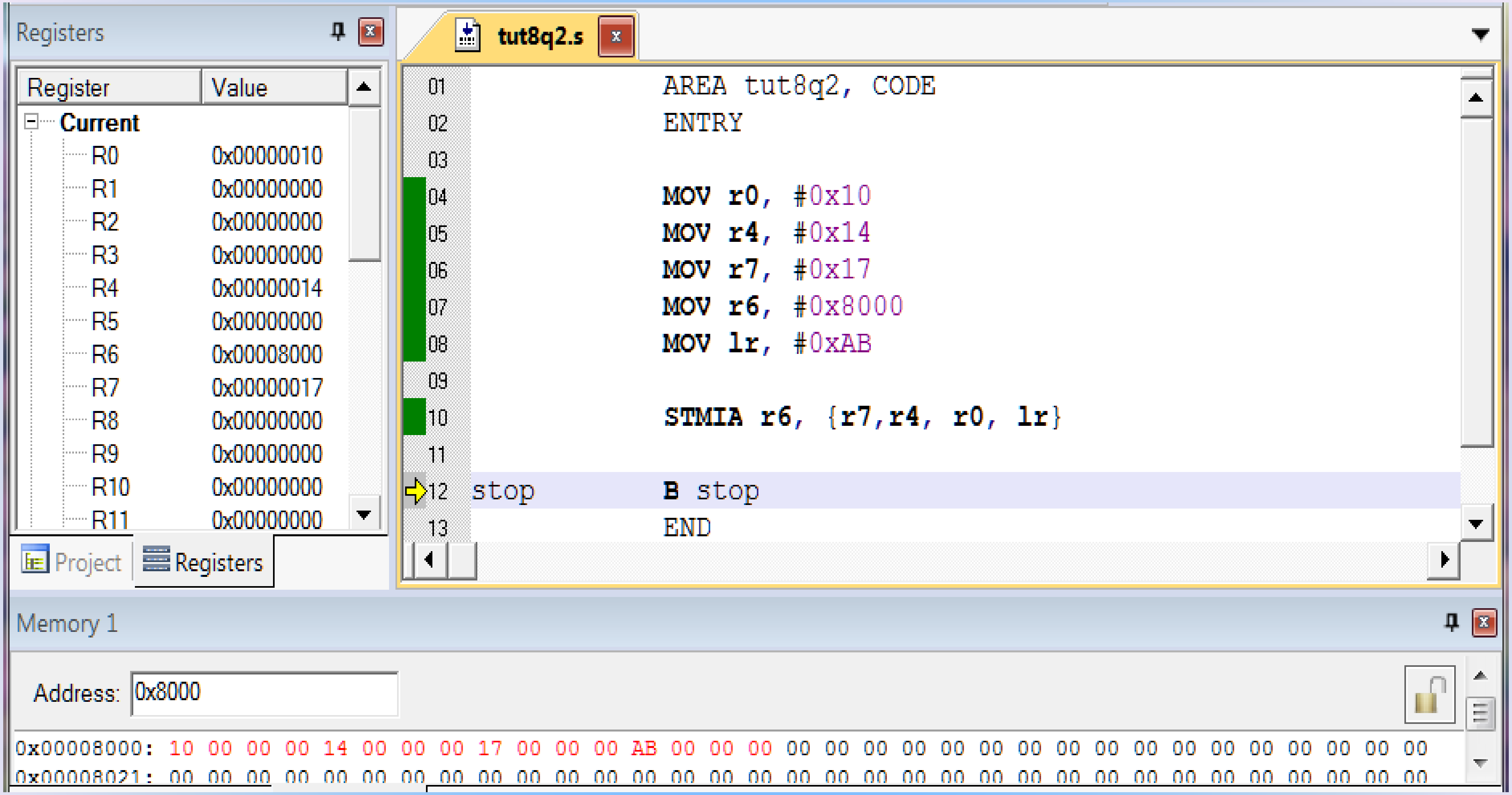
If register r6 holds the address 0x8000 and you execute the instruction

STMIA r6, {r7, r4, r0, lr}

Write the memory address now holds the value that was in registers r0, r4, r7 and lr respectively.

Answer:

Address 0x8000 holds value in r0 (= 0x10)
Address 0x8004 holds value in r4 (= 0x14)
Address 0x8008 holds value in r7 (= 0x17)
Address 0x800C holds value in lr (= 0xAB)



Register	Value
Current	
R0	0x00000010
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000014
R5	0x00000000
R6	0x00008000
R7	0x00000017
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000

```
            AREA tut8q2, CODE
            ENTRY

            MOV r0, #0x10
            MOV r4, #0x14
            MOV r7, #0x17
            MOV r6, #0x8000
            MOV lr, #0xAB

            STMIA r6, {r7,r4, r0, lr}

stop        B stop
            END
```

Address: 0x8000

```
0x00008000: 10 00 00 00 14 00 00 00 17 00 00 00 AB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00008021: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

3. Assume that memory and registers r0 through r3 appears as follows:

Address			Register
0x8010	0x00000001	0x13	r0
0x800C	0xFEEDDEAF	0xFFFFFFFF	r1
0x8008	0x00008888	0xEEEEEEEE	r2
0x8004	0x12340000	0x8000	r3
0x8000	0xBABE0000		

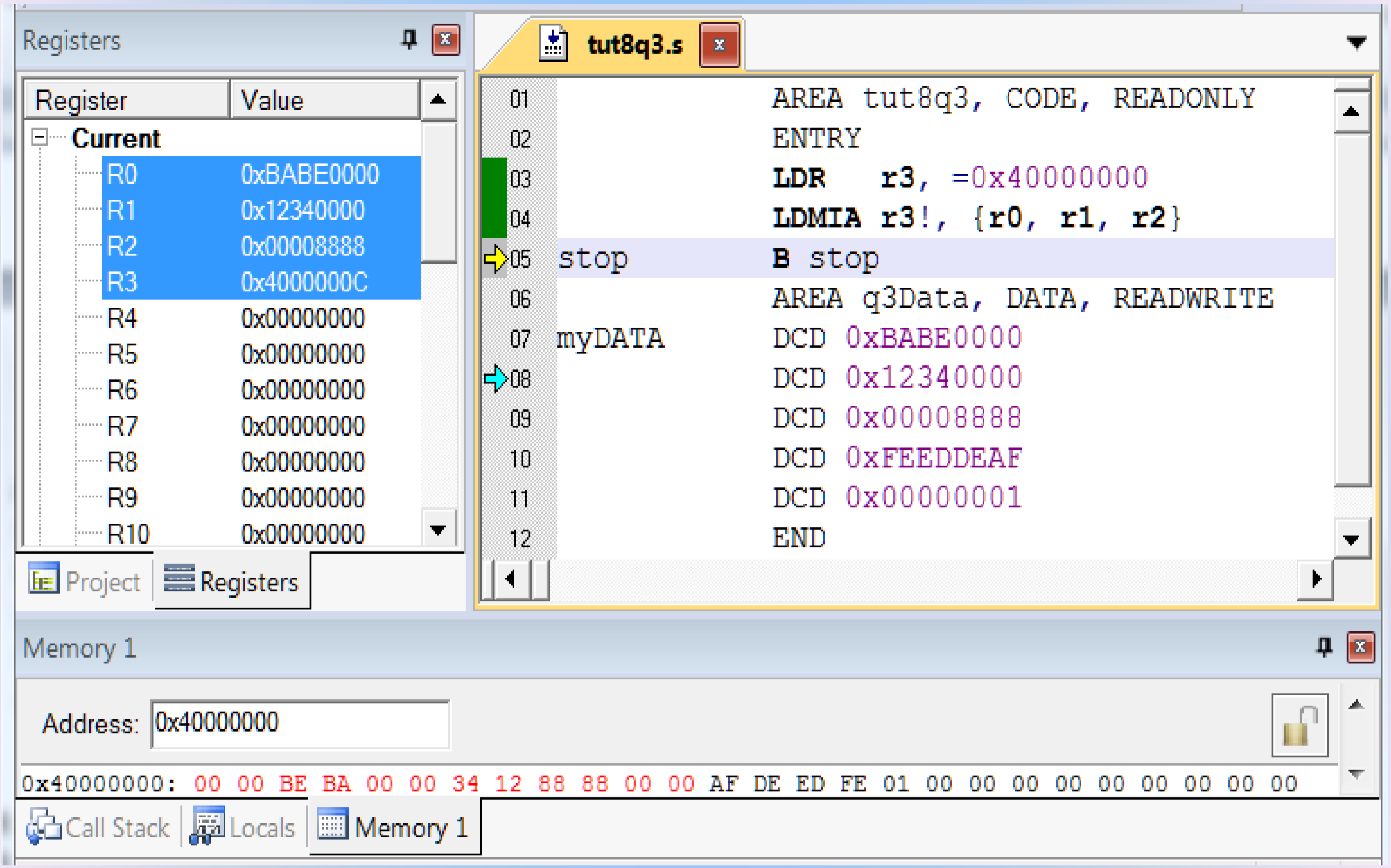
Give the contents of the memory and registers that have been changed after executing the instruction

```
LDMIA  r3!, {r0, r1, r2}
```

Q3 Ans:

Memory contents stay the same.

r0 = 0xBABE0000,
r1 = 0x12340000,
r2 = 0x00008888,
r3 = 0x0000800C



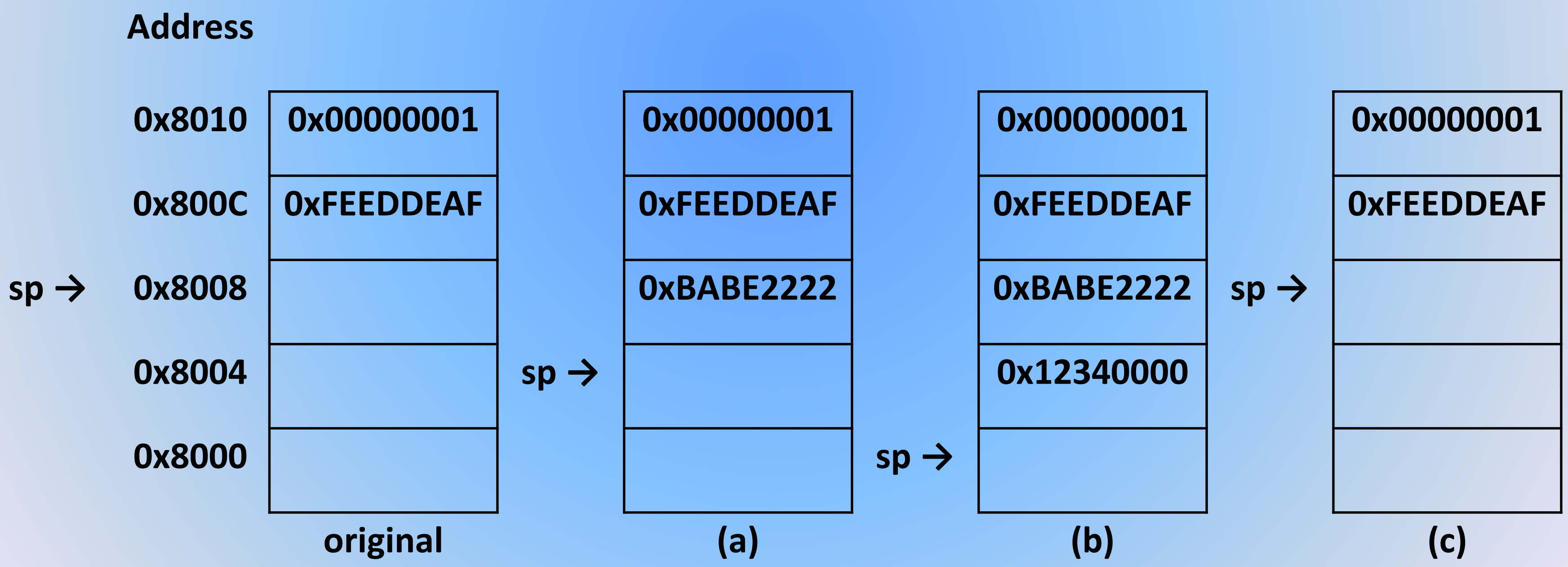
Register	Value
Current	
R0	0xBABE0000
R1	0x12340000
R2	0x00008888
R3	0x4000000C
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000

```
                AREA tut8q3, CODE, READONLY
                ENTRY
                LDR   r3, =0x40000000
                LDMIA r3!, {r0, r1, r2}
stop            B stop
                AREA q3Data, DATA, READWRITE
myDATA          DCD 0xBABE0000
                DCD 0x12340000
                DCD 0x00008888
                DCD 0xFEEDDEAF
                DCD 0x00000001
                END
```

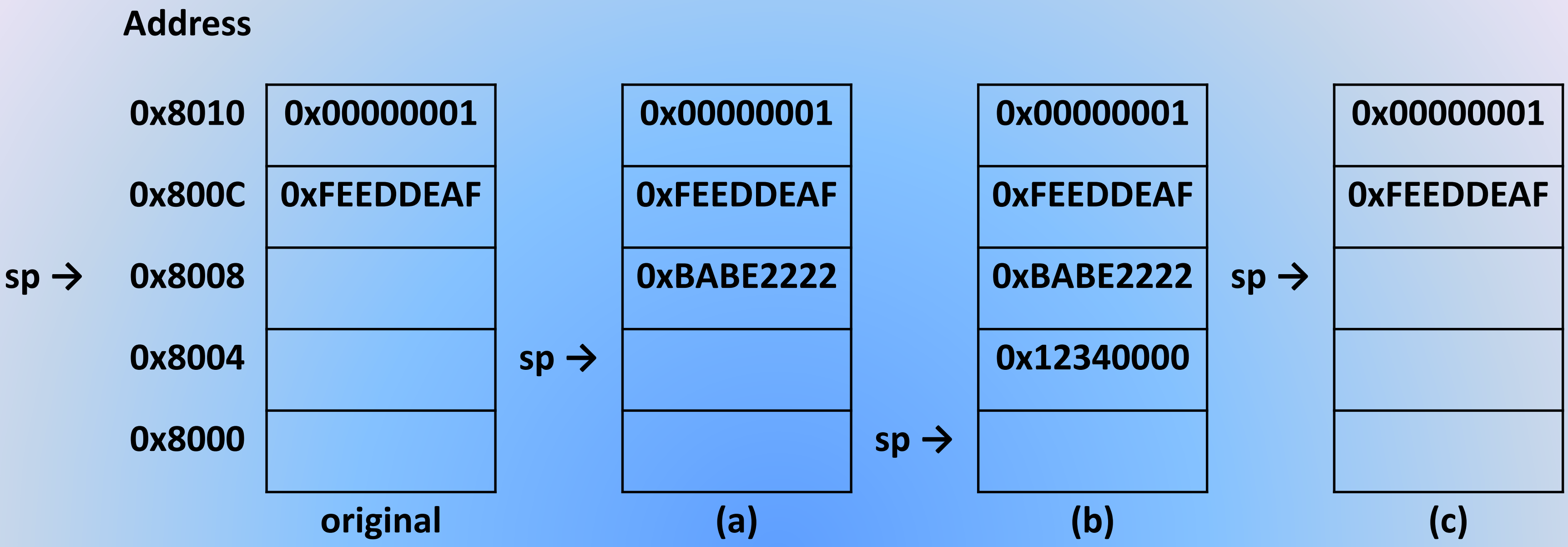
Address: 0x40000000

```
0x40000000: 00 00 BE BA 00 00 34 12 88 88 00 00 AF DE ED FE 01 00 00 00 00 00 00 00 00
```

4. Suppose that a stack appears as shown in the original diagram below. Give the instruction(s) that would push or pop data so that memory appears in the order shown. In other words, what instruction would be necessary to go from original state to that shown in (a) and then (b) and then (c)?



	Address	original		(a)		(b)		(c)
	0x8010	0x00000001		0x00000001		0x00000001		0x00000001
	0x800C	0xFEEDDEAF		0xFEEDDEAF		0xFEEDDEAF		0xFEEDDEAF
sp →	0x8008			0xBABE2222		0xBABE2222	sp →	
	0x8004		sp →			0x12340000		
	0x8000				sp →			



Address	original	(a)	(b)	(c)
0x8010	0x00000001	0x00000001	0x00000001	0x00000001
0x800C	0xFEEDDEAF	0xFEEDDEAF	0xFEEDDEAF	0xFEEDDEAF
sp → 0x8008		0xBABE2222	0xBABE2222	sp →
0x8004		sp →	0x12340000	
0x8000			sp →	

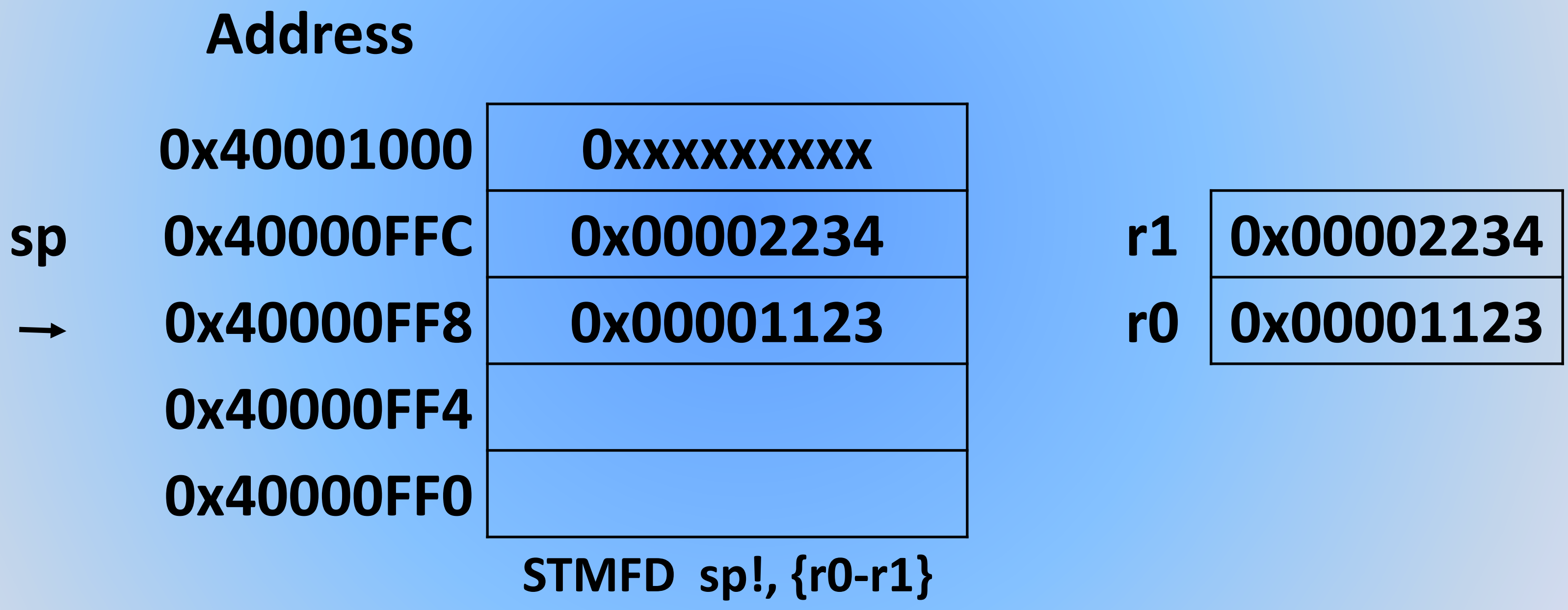
```
LDR      sp,   =0x00008008
LDR      r1 ,  =0xBABE2222
LDR      r0 ,  =0x12340000
STMDA    sp!, {r1}                    ;gets you to (a)
STMDA    sp!, {r0}                    ;gets you to (b)
LDMIB    sp!, {r0, r1}                ;gets you to (c)
```

5. Registers r0, r1 and r2 contain 0x1123, 0x2234 and 0x3356 respectively. What are the contents of each register after this sequence of instructions is executed? Suppose sp = 0x40001000 before the execution. Also show the contents of the stack.

```
STMFD  sp!, {r0-r1}          ;PUSH r0, r1
STMFD  sp!, {r2}             ;PUSH r2
LDMFD  sp!, {r0, r2}         ;POP r0, r2
LDMFD  sp!, {r1}             ;POP r1
```

Initially, registers r0, r1 and r2 contain 0x1123, 0x2234 and 0x3356 respectively

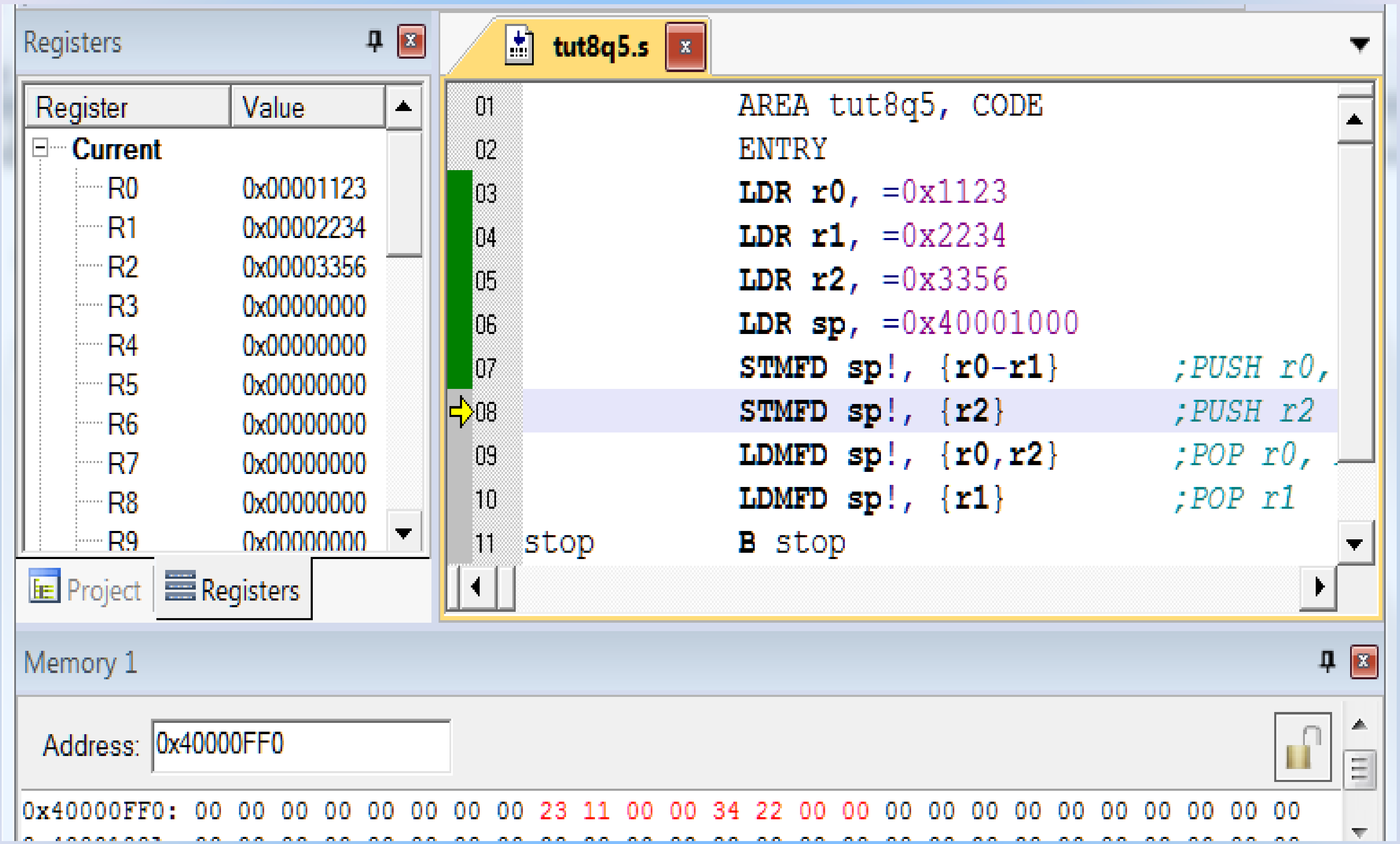
PUSH r0, r1



	Address	
	0x40001000	0xxxxxxxxx
sp	0x40000FFC	0x00002234
→	0x40000FF8	0x00001123
	0x40000FF4	
	0x40000FF0	

STMFD sp!, {r0-r1}

r1	0x00002234
r0	0x00001123



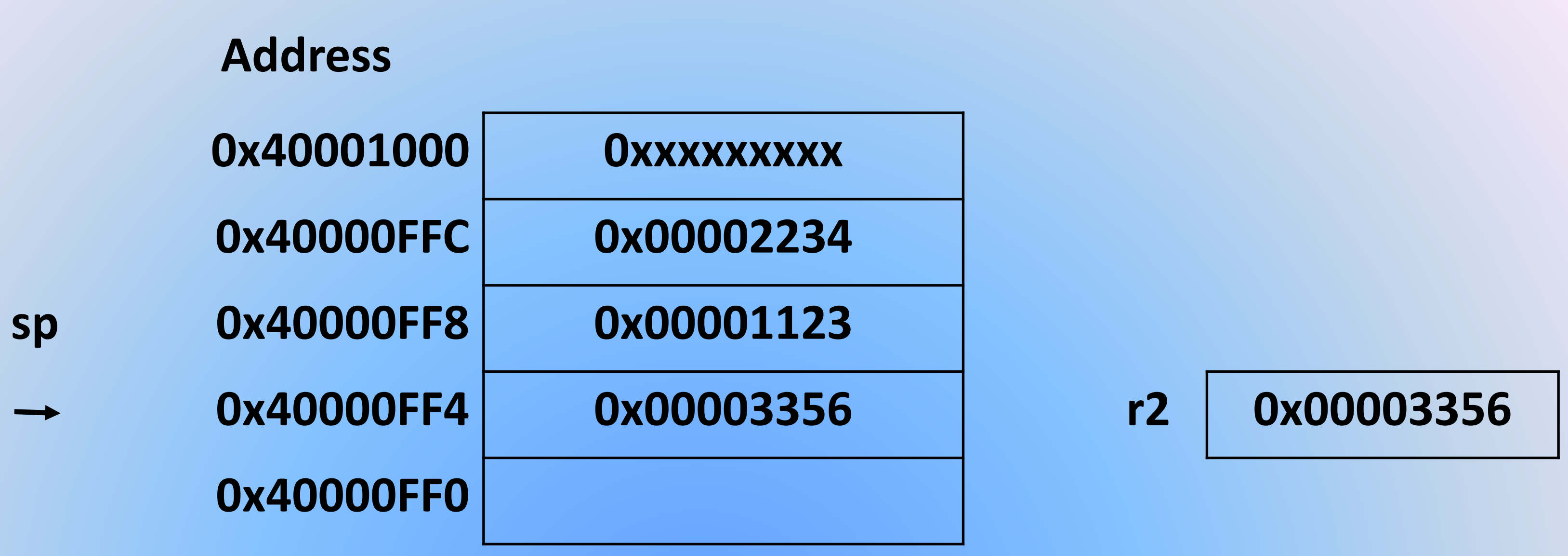
Register	Value
Current	
R0	0x00001123
R1	0x00002234
R2	0x00003356
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000

```
            AREA tut8q5, CODE
            ENTRY
            LDR r0, =0x1123
            LDR r1, =0x2234
            LDR r2, =0x3356
            LDR sp, =0x40001000
            STMFD sp!, {r0-r1}          ;PUSH r0,
            STMFD sp!, {r2}             ;PUSH r2
            LDMFD sp!, {r0,r2}          ;POP r0, .
            LDMFD sp!, {r1}             ;POP r1
stop        B stop
```

Address: 0x40000FF0

```
0x40000FF0: 00 00 00 00 00 00 00 00 23 11 00 00 34 22 00 00 00 00 00 00 00 00 00 00 00 00
```

PUSH r2

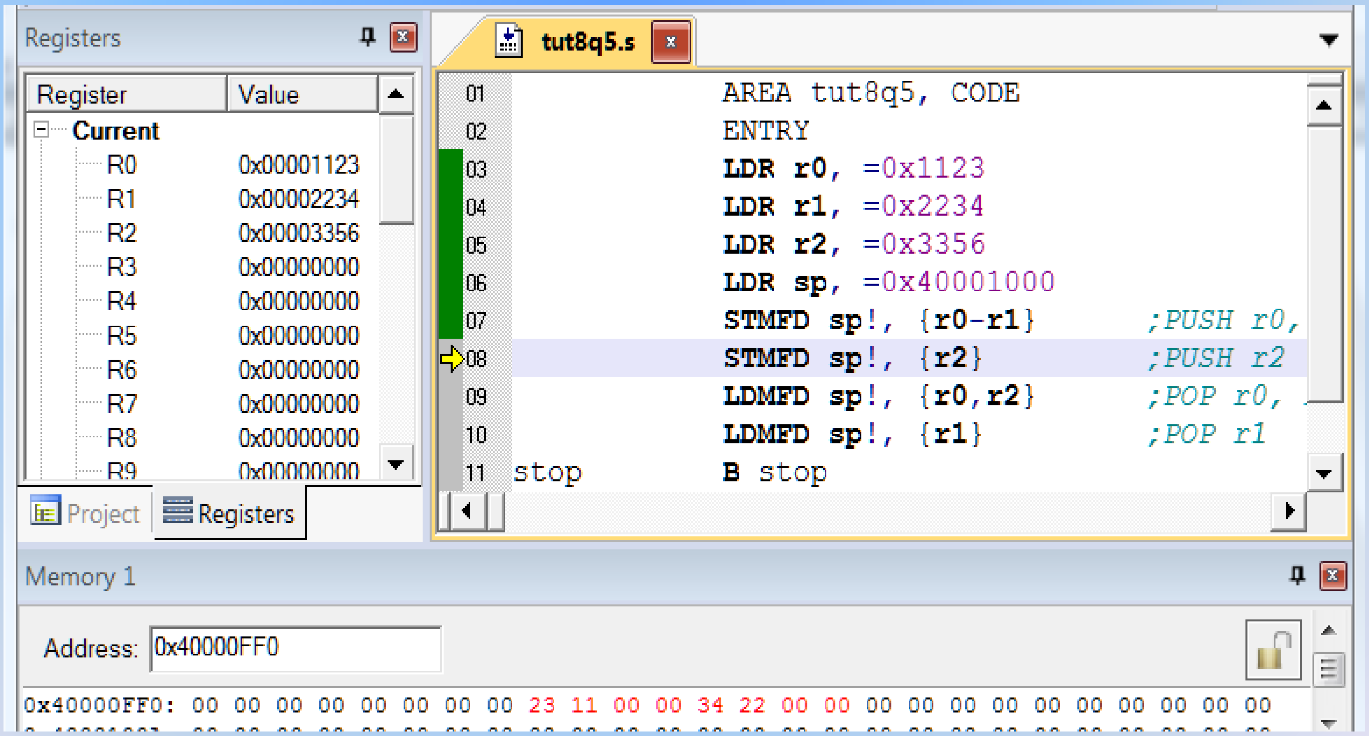


Address

	Address	
	0x40001000	0xxxxxxxxx
	0x40000FFC	0x00002234
sp	0x40000FF8	0x00001123
→	0x40000FF4	0x00003356
	0x40000FF0	

r2 0x00003356

STMFD sp!, {r2}



Register	Value
Current	
R0	0x00001123
R1	0x00002234
R2	0x00003356
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000

```
                AREA tut8q5, CODE
                ENTRY
                LDR r0, =0x1123
                LDR r1, =0x2234
                LDR r2, =0x3356
                LDR sp, =0x40001000
                STMFD sp!, {r0-r1}          ;PUSH r0,
                STMFD sp!, {r2}             ;PUSH r2
                LDMFD sp!, {r0,r2}          ;POP r0, .
                LDMFD sp!, {r1}             ;POP r1
stop            B stop
```

Address: 0x40000FF0

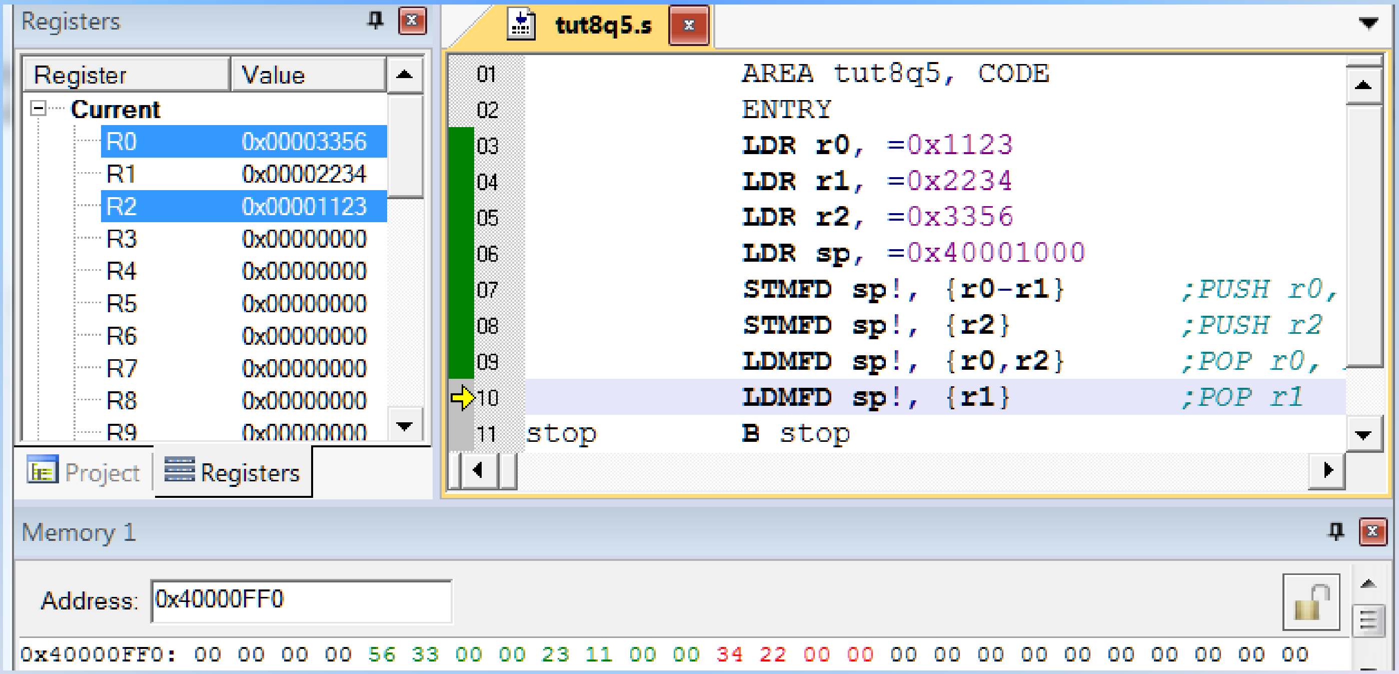
```
0x40000FF0: 00 00 00 00 00 00 00 00 23 11 00 00 34 22 00 00 00 00 00 00 00 00 00 00 00 00
```

POP r0, r2

	Address	
sp	0x40001000	0xxxxxxxxx
→	0x40000FFC	0x00002234
	0x40000FF8	0x00001123
	0x40000FF4	0x00003356
	0x40000FF0	

r2	0x00001123
r0	0x00003356

LDMFD sp!, {r0, r2}



Register	Value
Current	
R0	0x00003356
R1	0x00002234
R2	0x00001123
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000

```
                AREA tut8q5, CODE
                ENTRY
                LDR r0, =0x1123
                LDR r1, =0x2234
                LDR r2, =0x3356
                LDR sp, =0x40001000
                STMFD sp!, {r0-r1}          ;PUSH r0,
                STMFD sp!, {r2}             ;PUSH r2
                LDMFD sp!, {r0,r2}          ;POP r0,
                LDMFD sp!, {r1}             ;POP r1
stop            B stop
```

Address: 0x40000FF0

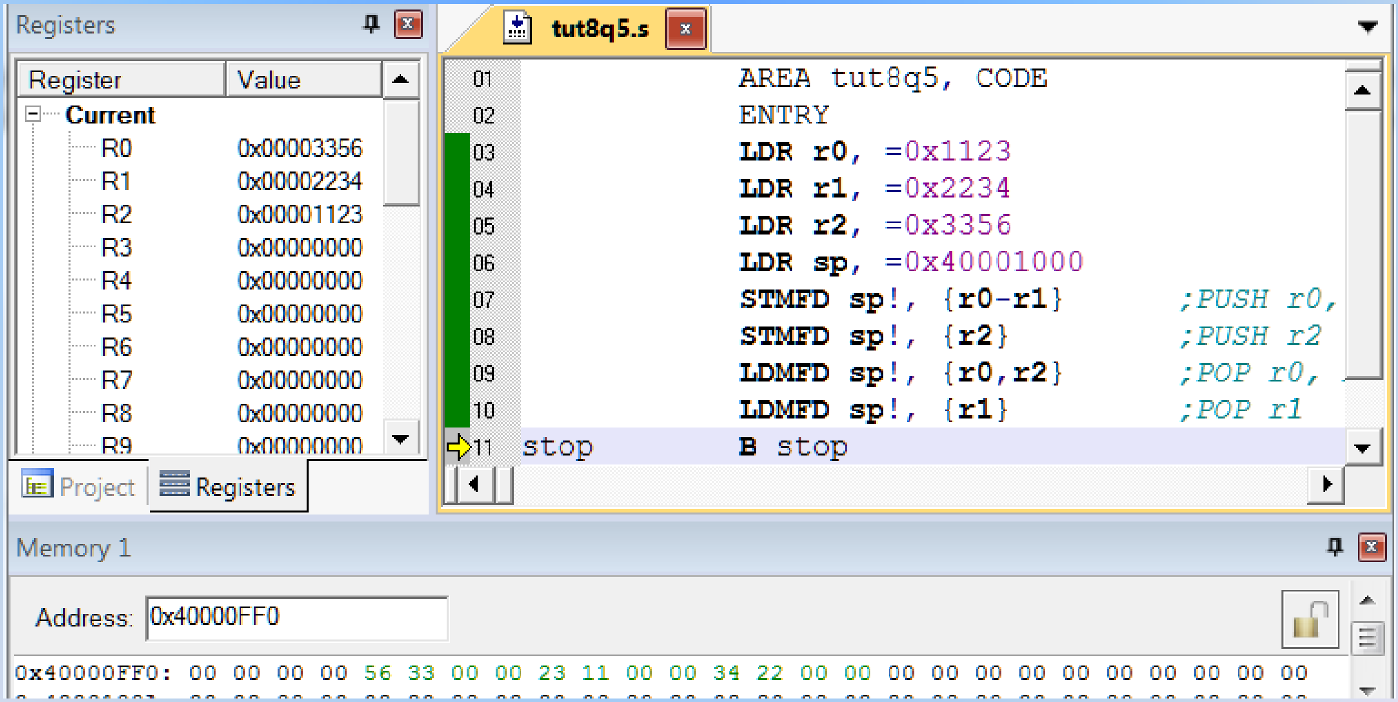
```
0x40000FF0: 00 00 00 00 56 33 00 00 23 11 00 00 34 22 00 00 00 00 00 00 00 00 00 00 00 00
```

POP r1

sp	Address	
→	0x40001000	0xxxxxxxxx
	0x40000FFC	0x00002234
	0x40000FF8	0x00001123
	0x40000FF4	0x00003356
	0x40000FF0	

r1 0x00002234

LDMFD sp!, {r1}



Register	Value
Current	
R0	0x00003356
R1	0x00002234
R2	0x00001123
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000

```
            AREA tut8q5, CODE
            ENTRY
            LDR r0, =0x1123
            LDR r1, =0x2234
            LDR r2, =0x3356
            LDR sp, =0x40001000
            STMFD sp!, {r0-r1}       ;PUSH r0,
            STMFD sp!, {r2}          ;PUSH r2
            LDMFD sp!, {r0,r2}       ;POP r0,
            LDMFD sp!, {r1}          ;POP r1
stop        B stop
```

Address: 0x40000FF0

```
0x40000FF0: 00 00 00 00 56 33 00 00 23 11 00 00 34 22 00 00 00 00 00 00 00 00 00 00 00 00
```

Eventually, registers r0, r1, r2 and sp contain 0x3356, 0x2234, 0x1123 and 0x40001000 respectively