**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

Date: 31/1/21

|  |  |  |
| --- | --- | --- |
| Name: Adithya M S | SRN: PES1UG19CS027 | Section: A |

Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_1\_\_\_

**Based on the value of the number in R0, Write an ALP to store 1 in R1 if R0 is zero, Store 2 in R1 if R0 is positive, Store 3 in R1 if R0 is negative.**

I.

mov r0,#5

cmp r0,#0

beq zero

bpl positive

mov r1,#3

swi 0x11

zero:

mov r1,#1

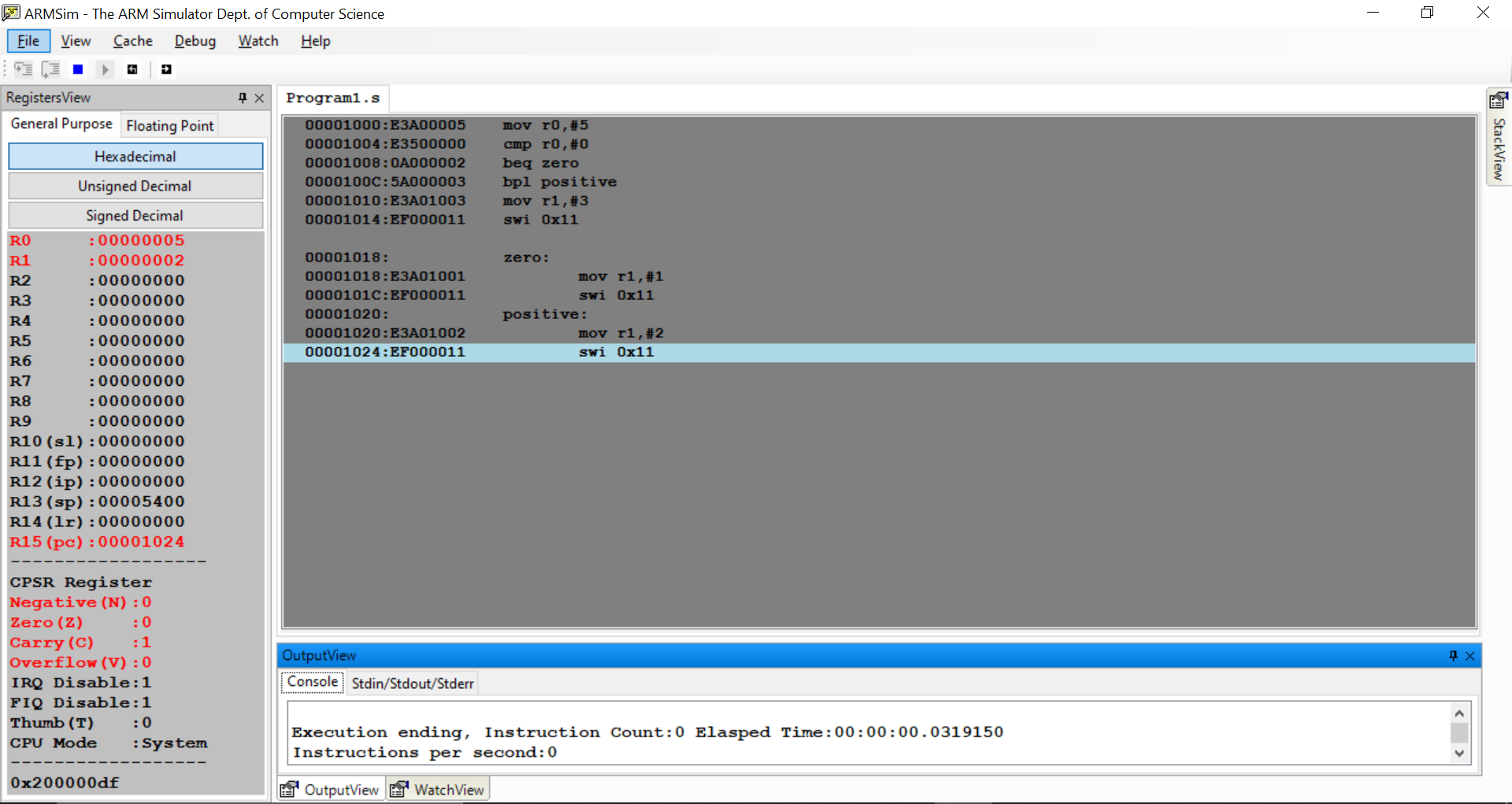
swi 0x11

positive:

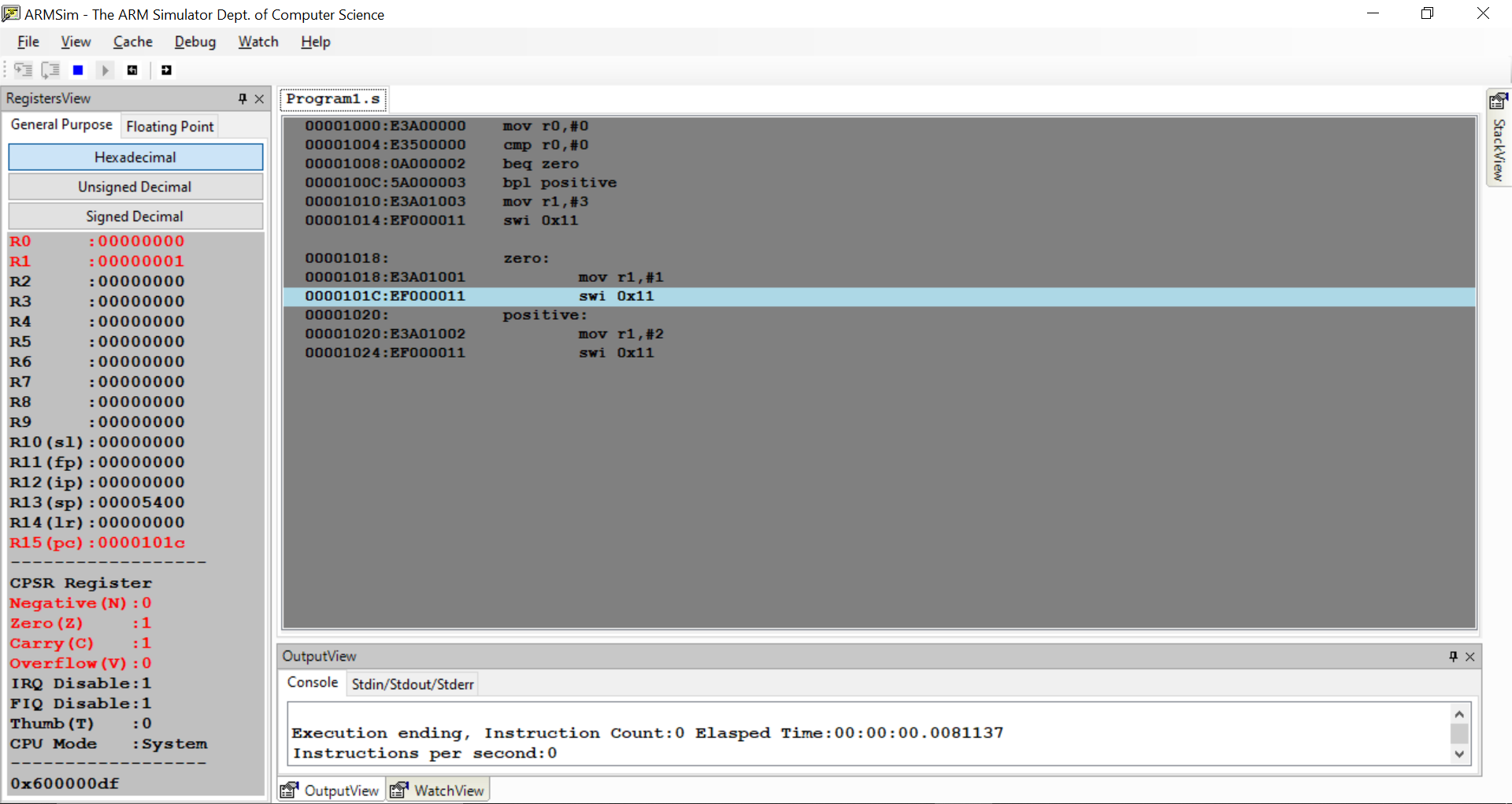
mov r1,#2

swi 0x11

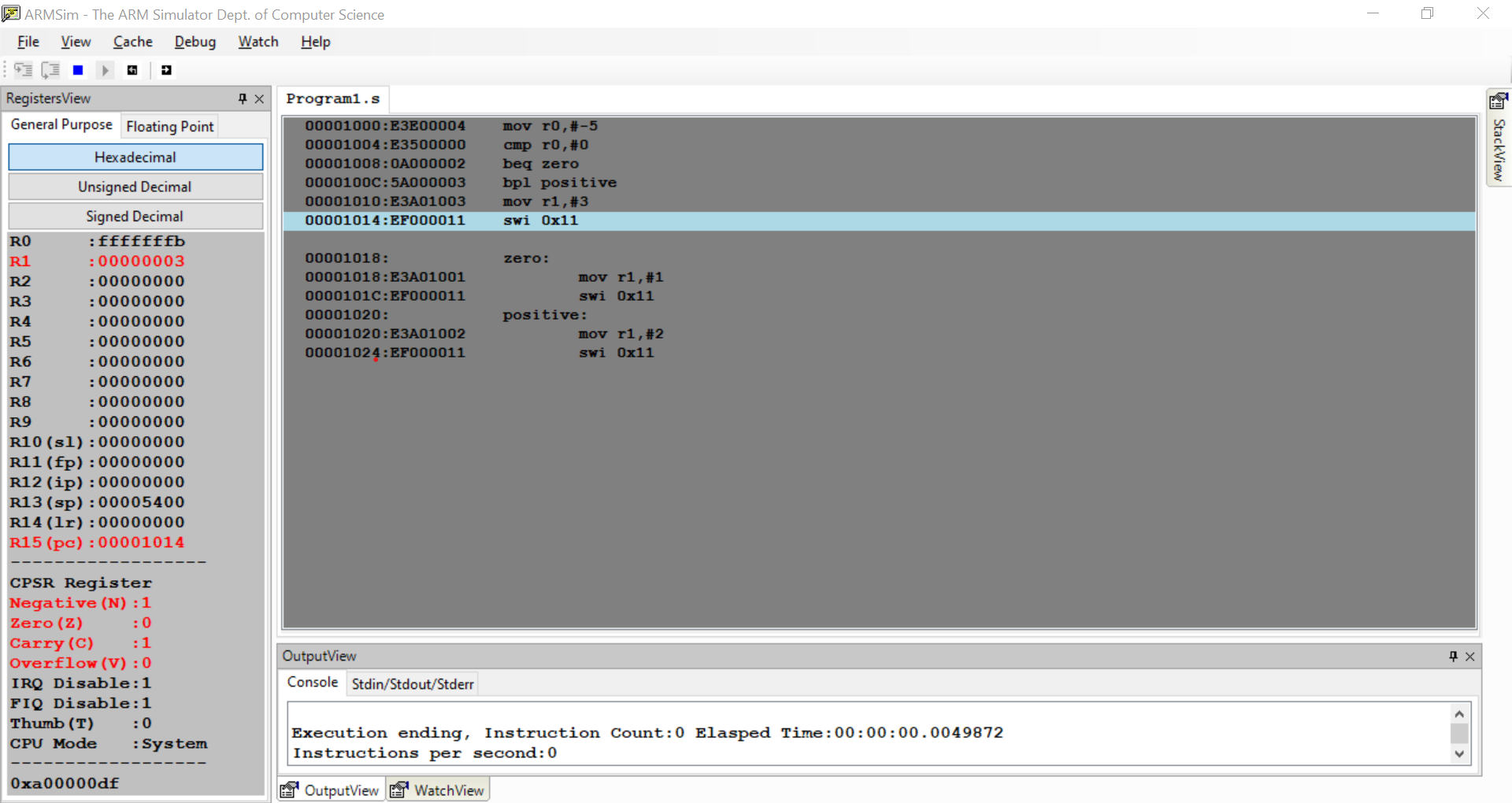
Example 1:



Example 2:



Example 3:



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_2\_\_\_

**Write an ALP to compare the value of R0 and R1, add if R0 = R1, else subtract**

I.

mov r0,#5

mov r1,#3

cmp r0,r1

beq equal

subs r2,r0,r1

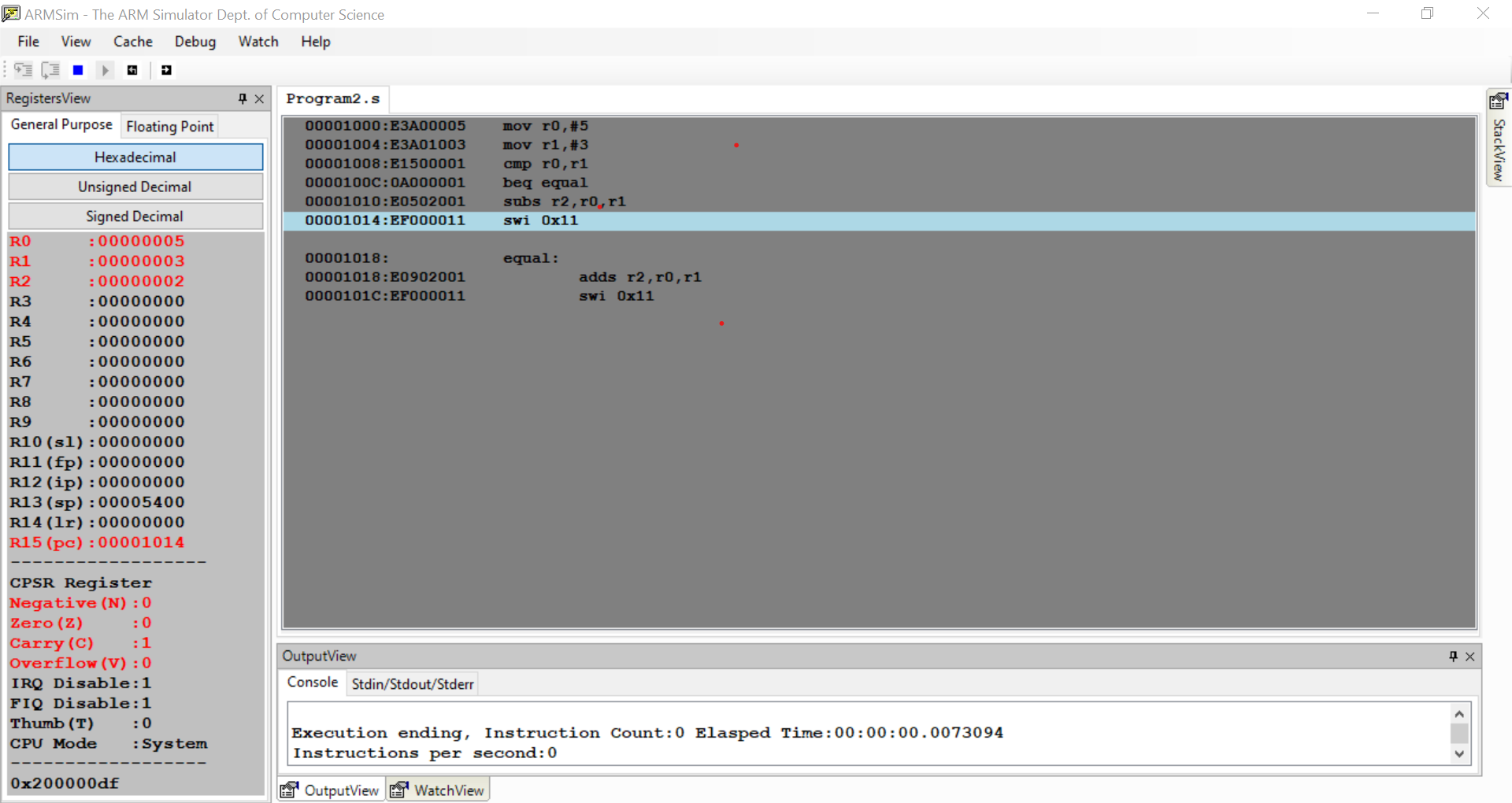
swi 0x11

equal:

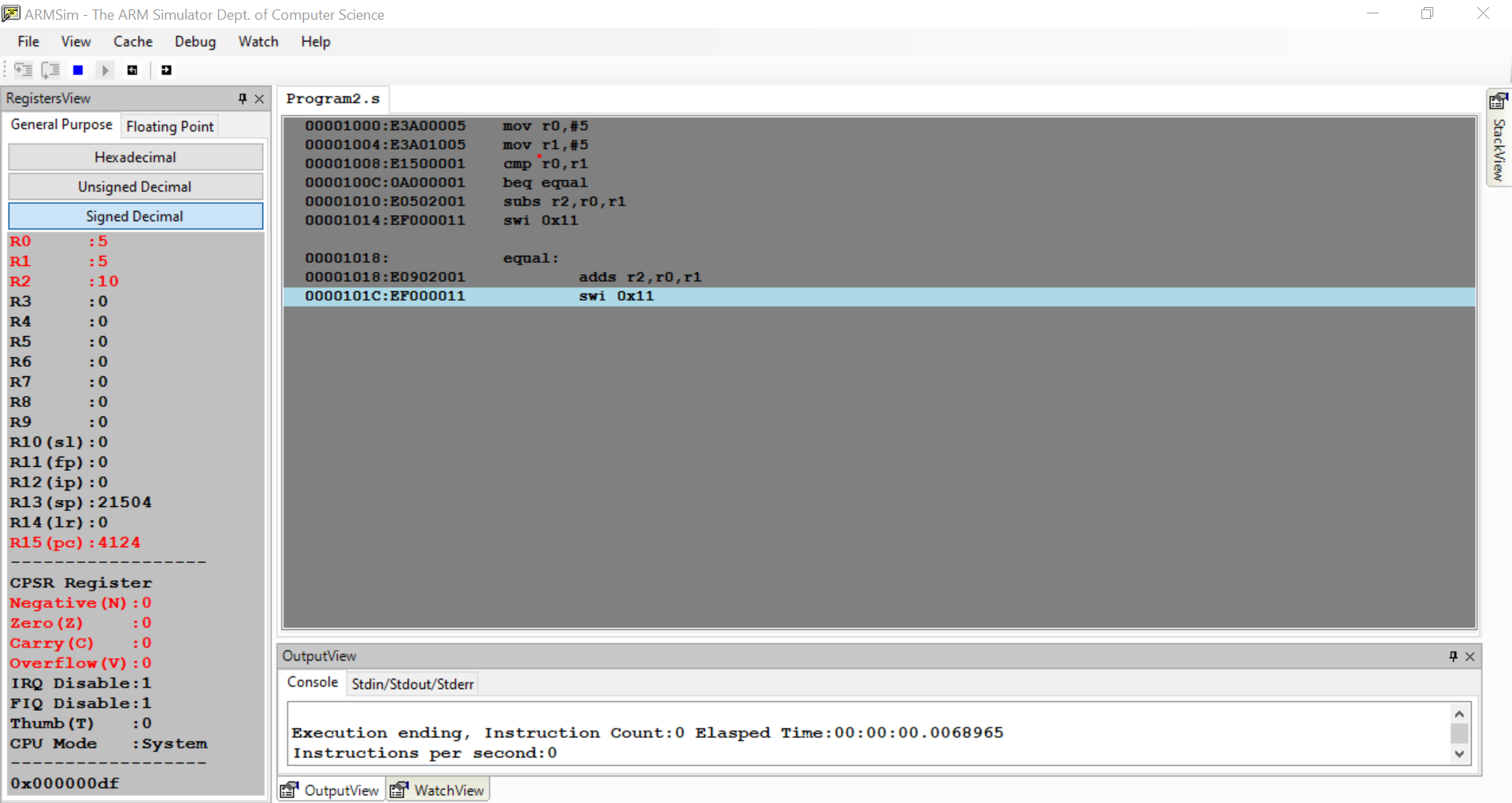
adds r2,r0,r1

swi 0x11

Example 1:



Example 2:



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_3\_\_\_

**Write an ALP to find the factorial of a number stored in R0. Store the value in R1 (without using LDR and STR instructions).Use only registers.**

I.

mov r0,#5

mov r1,#1

loop:

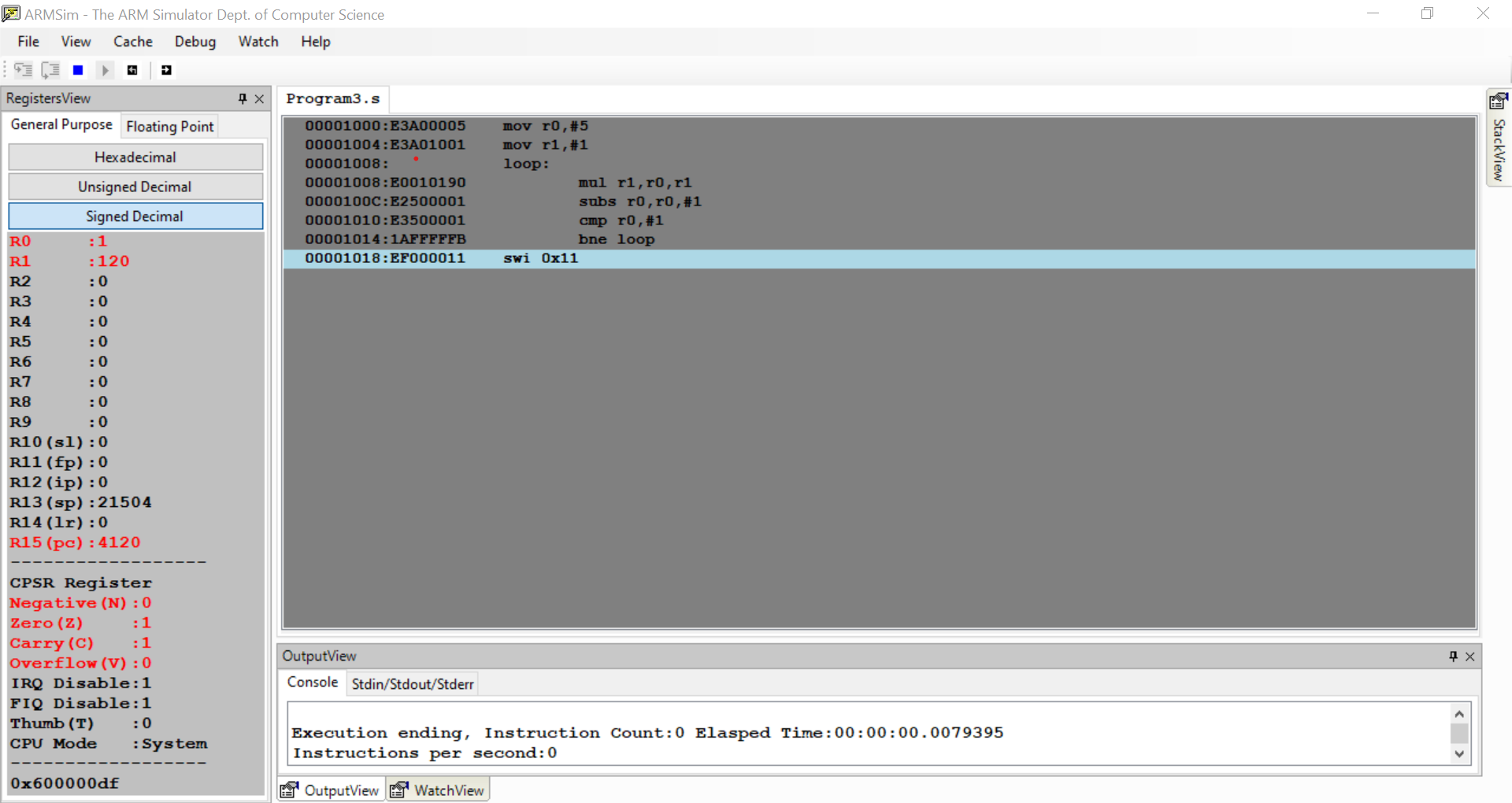
mul r1,r0,r1

subs r0,r0,#1

cmp r0,#1

bne loop

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_4a\_\_\_

**Write an ALP to add two 32 bit numbers loaded from memory and store the result in memory.**

I.

.data

A:.word 53920142

B:.word 38296104

C:.word 0

.text

ldr r1,=A

ldr r2,=B

ldr r3,=C

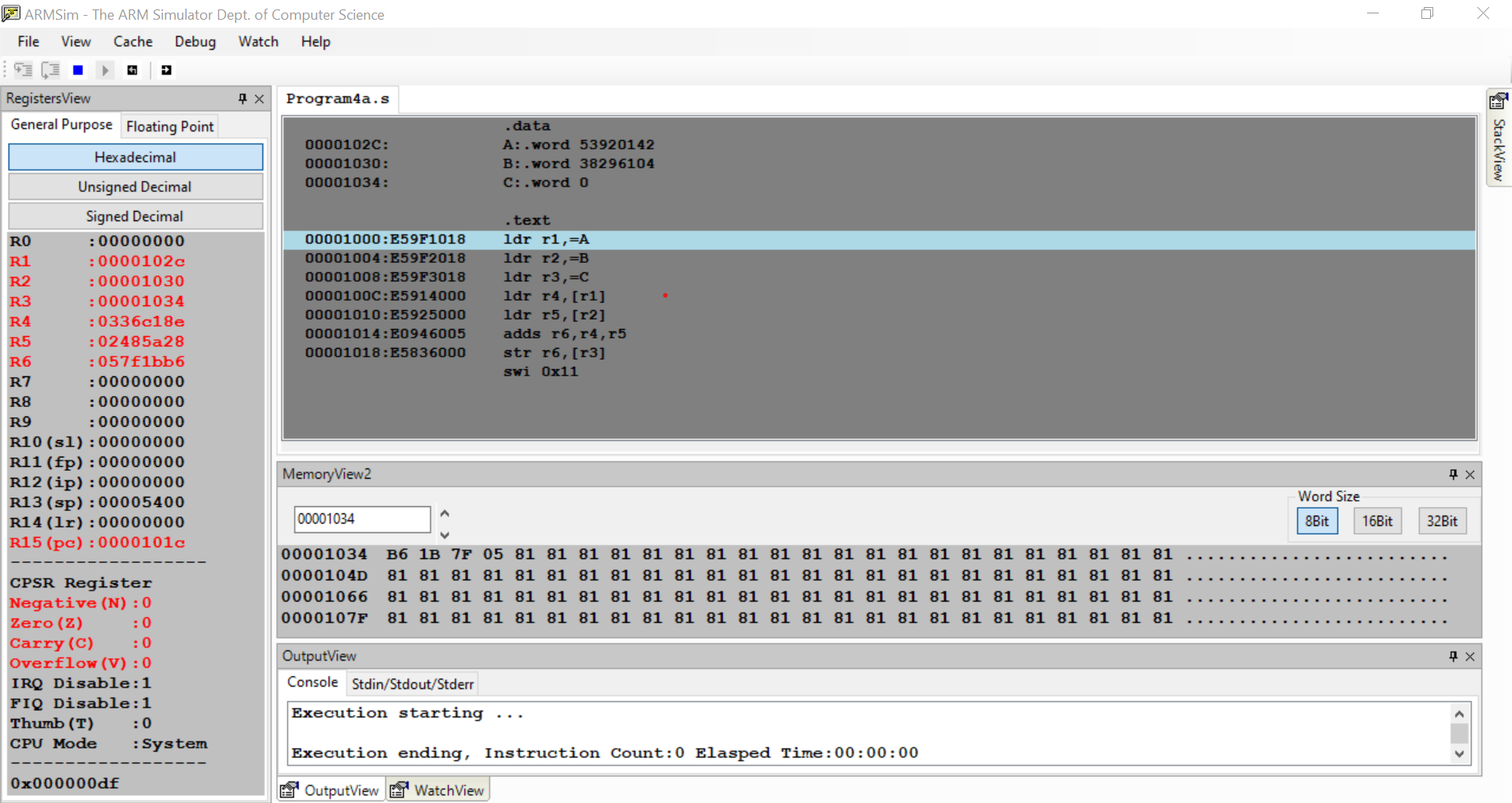
ldr r4,[r1]

ldr r5,[r2]

adds r6,r4,r5

str r6,[r3]

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_4b\_\_\_

**Write an ALP to add two 16 bit numbers loaded from memory and store the result in memory.**

I.

.data

A:.hword 17

B:.hword 15

C: .hword

.text

ldr r1,=A

ldr r2,=B

ldr r3,=C

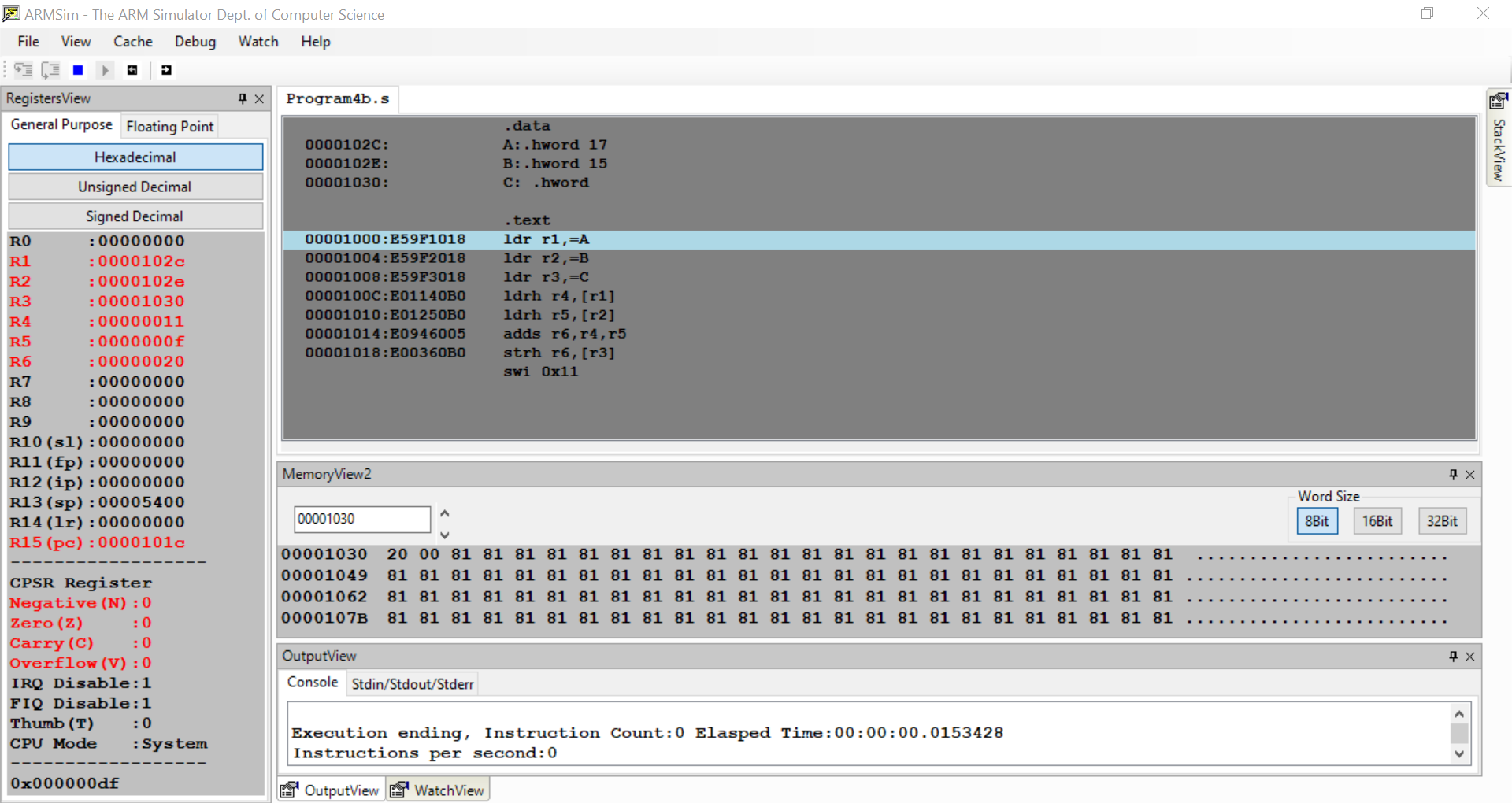
ldrh r4,[r1]

ldrh r5,[r2]

adds r6,r4,r5

strh r6,[r3]

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_5a\_\_\_

**Write an ALP to find GCD of two numbers (without using LDR and STR instructions).Both numbers are in registers. Use only registers.**

I.

mov r0,#15

mov r1,#20

cmp r0,r1

beq end

loop:

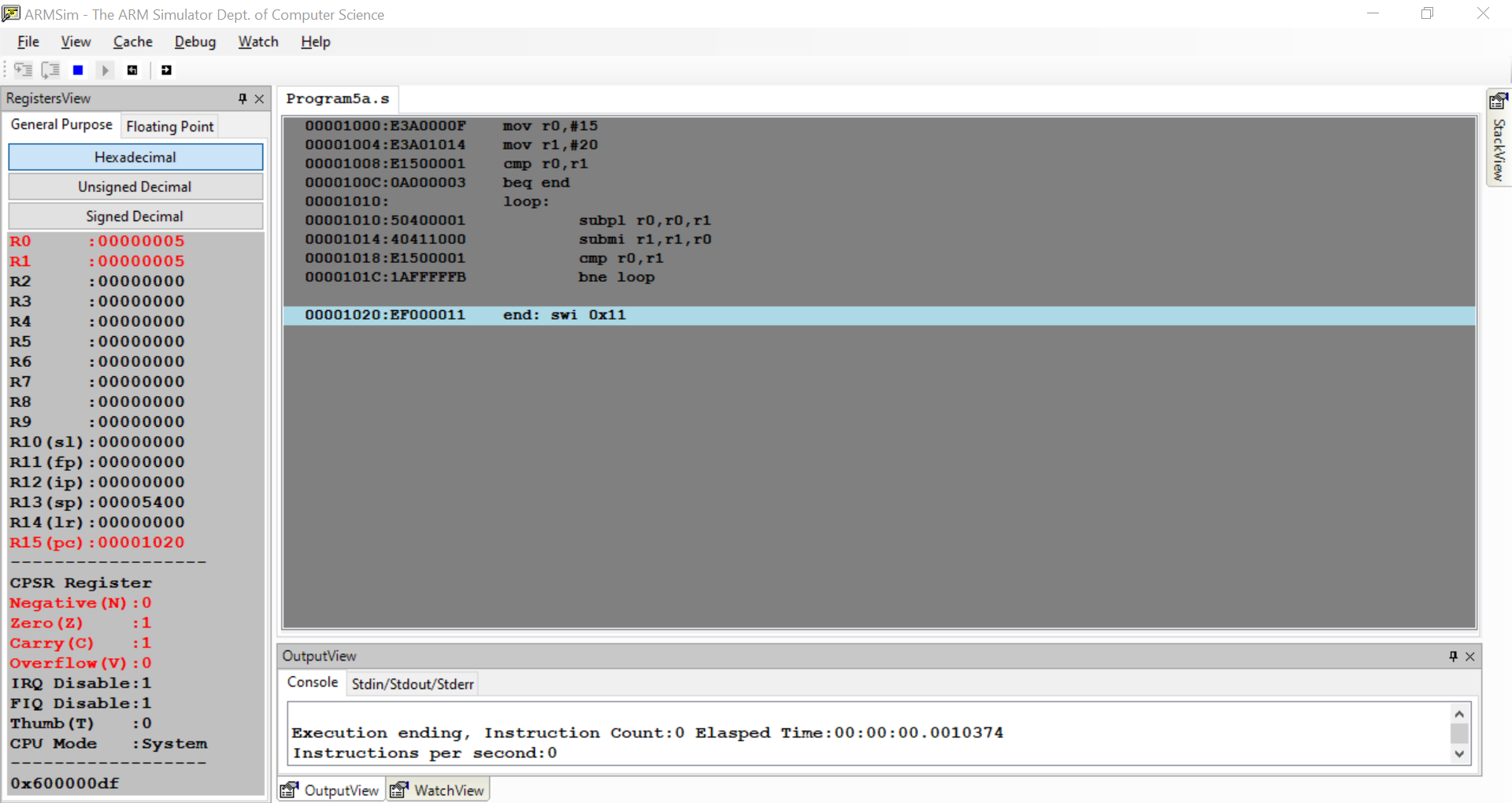
subpl r0,r0,r1

submi r1,r1,r0

cmp r0,r1

bne loop

end: swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_5b\_\_\_

**Write an ALP to find the GCD of given numbers (both numbers in memory). Store result in memory.**

I.

.data

A:.word 15

B:.word 20

C:.word

.text

ldr r0,=A

ldr r1,=B

ldr r2,=C

ldr r3,[r0]

ldr r4,[r1]

cmp r3,r4

beq end

loop:

subpl r3,r3,r4

submi r4,r4,r3

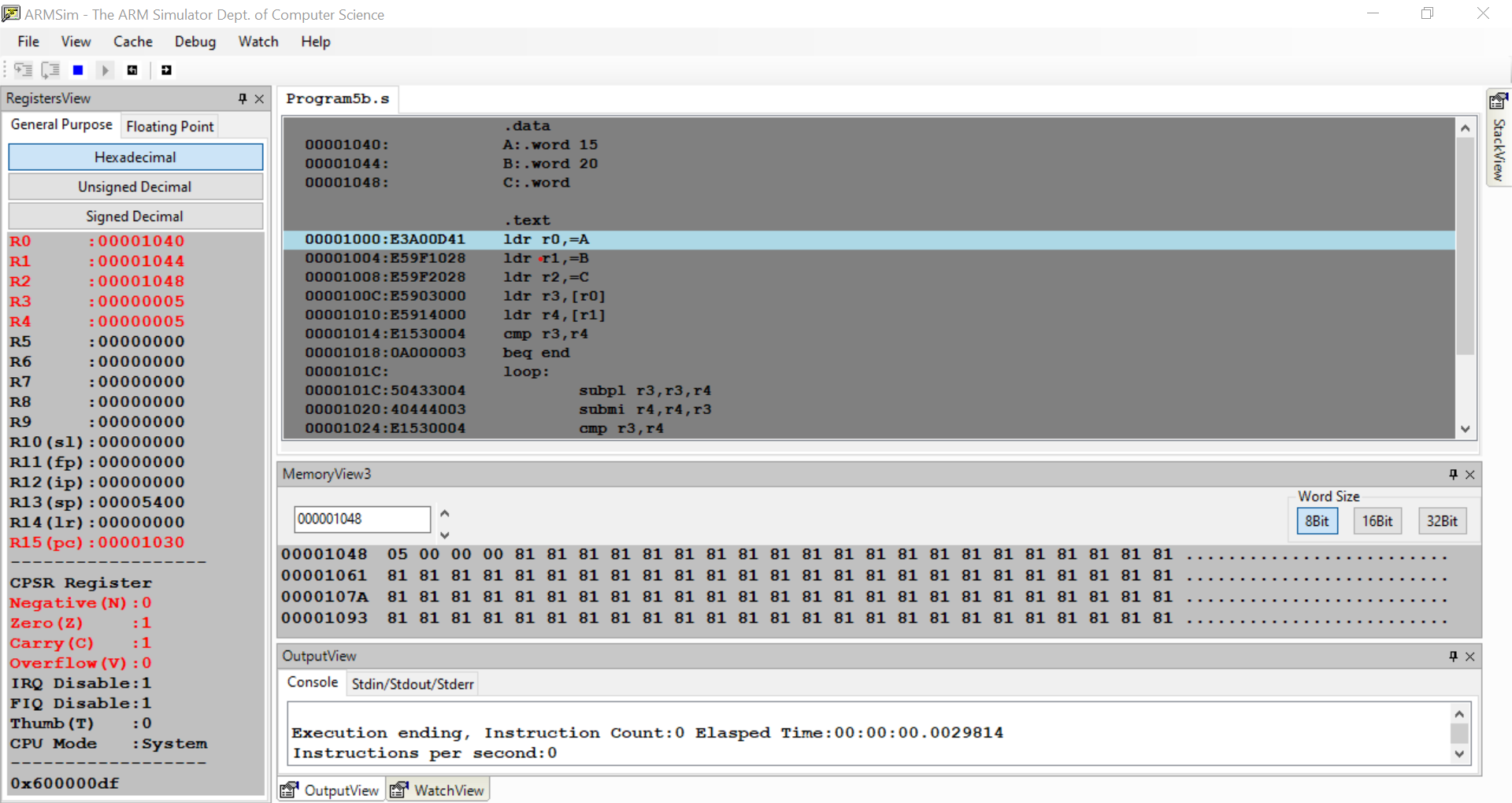
cmp r3,r4

bne loop

end:

str r3,[r2]

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_6a\_\_\_

**Write an ALP to add an array of ten 32-bit numbers from memory**

I.

.data

A:.word 20,40,10,30,60,90,70,50,80,100

.text

ldr r0,=A

mov r1,#10

mov r2,#0

loop:

ldr r3,[r0],#4

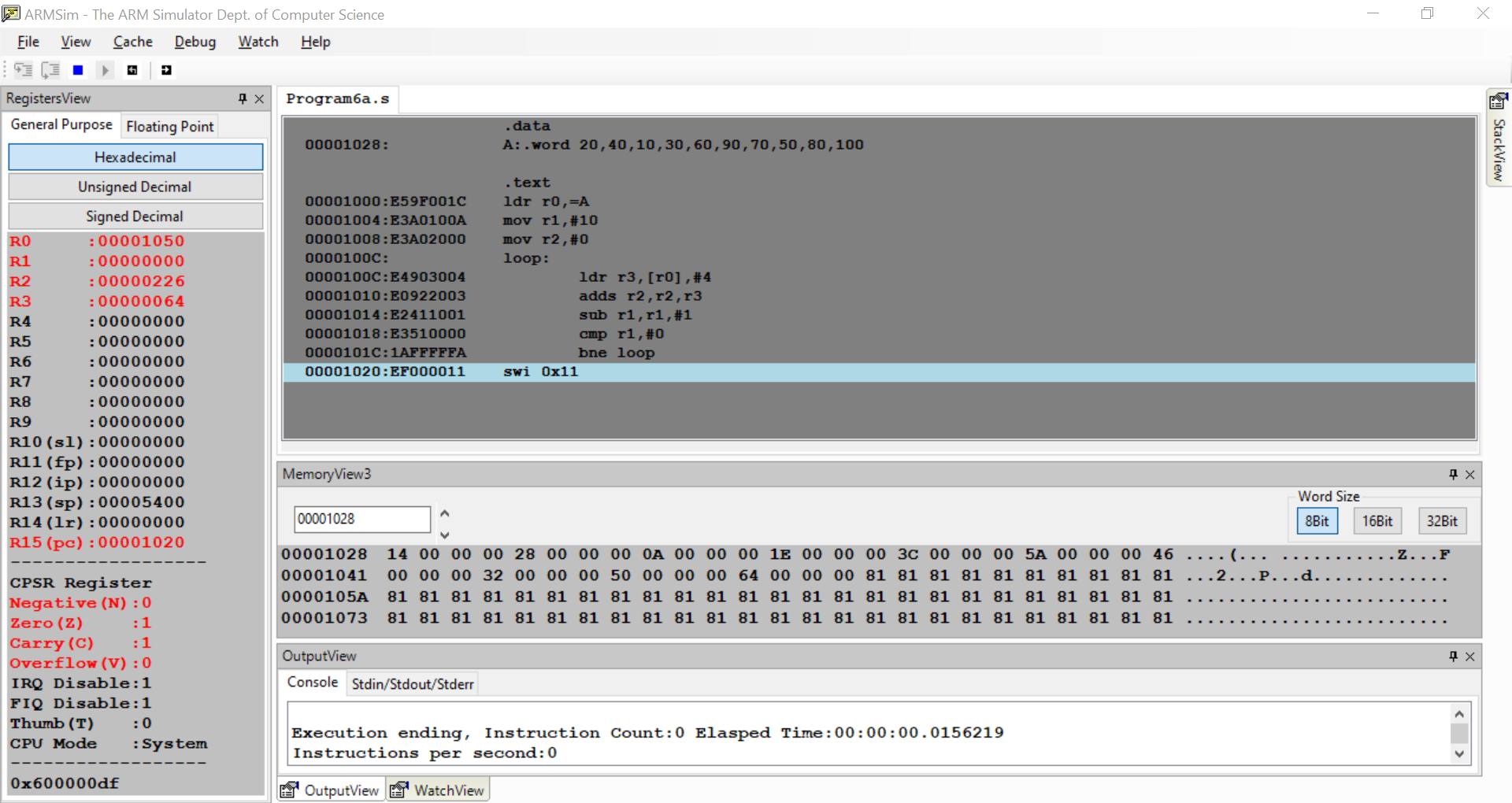
adds r2,r2,r3

sub r1,r1,#1

cmp r1,#0

bne loop

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_6b\_\_\_

**Write an ALP to add array of ten 8-bit numbers taking data from memory location stored as byte data**

I.

.data

A:.byte 20,40,10,30,60,90,70,100,80,50

.text

ldr r0,=A

mov r1,#10

mov r2,#0

loop:

ldrb r3,[r0],#1

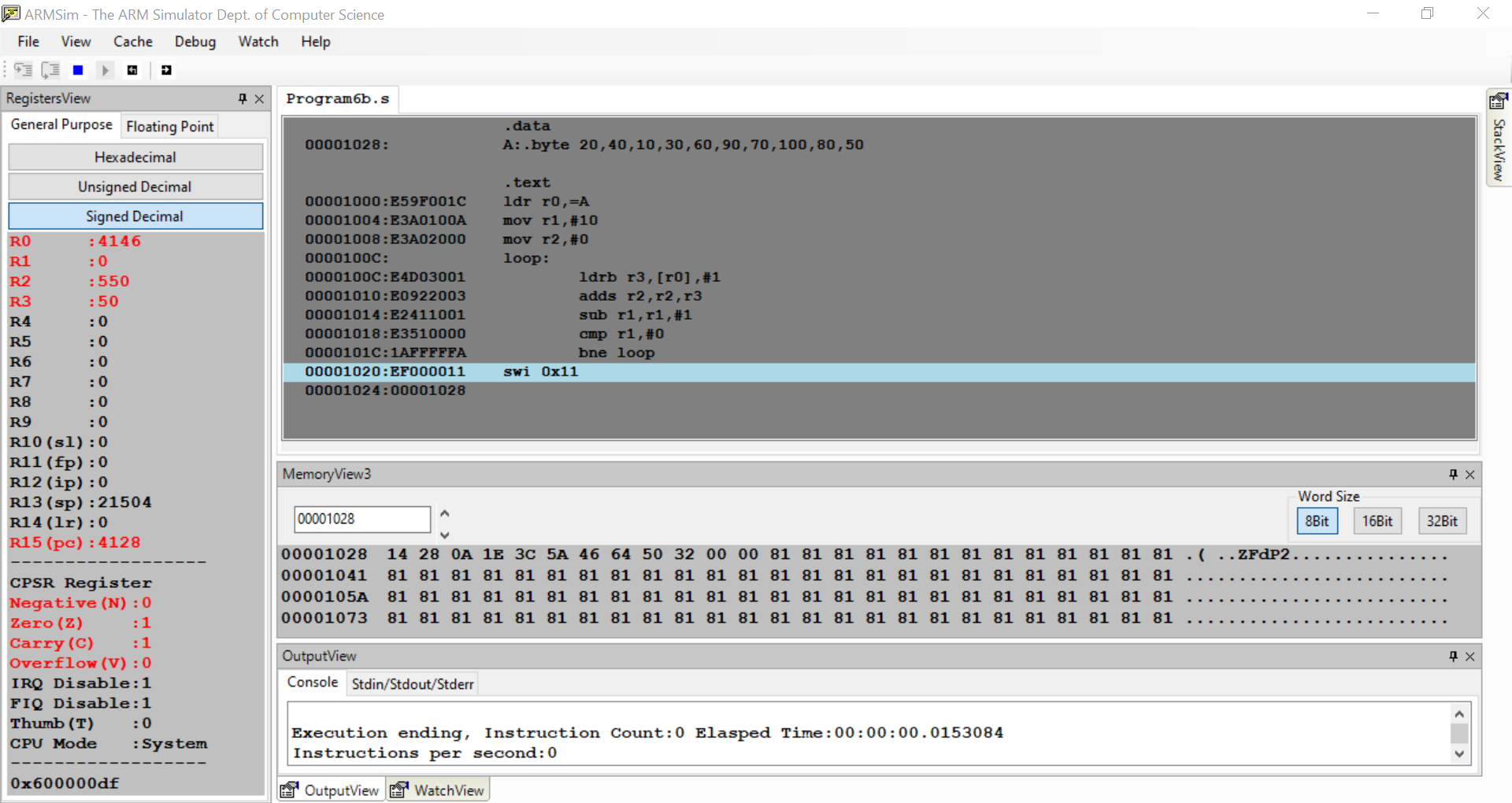
adds r2,r2,r3

sub r1,r1,#1

cmp r1,#0

bne loop

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_7\_\_\_

**Write an ALP to multiply using barrel shifter.**

**35\*R0**

I.

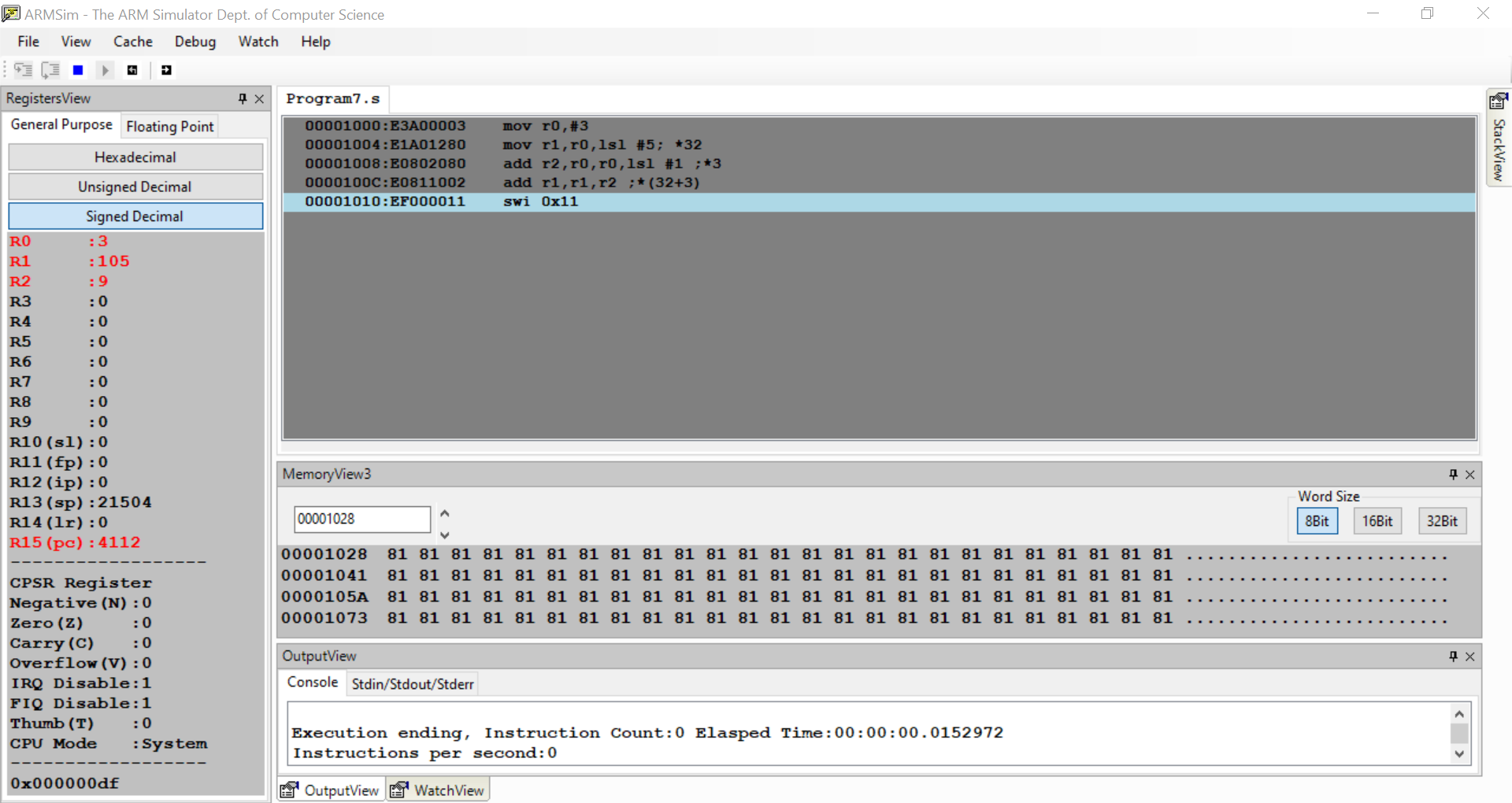
mov r0,#3

mov r1,r0,lsl #5; \*32

add r2,r0,r0,lsl #1 ;\*3

add r1,r1,r2 ;\*(32+3)

swi 0x11



Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_8\_\_\_

**Write an ALP to evaluate the expression (A+B) + (3\*B), where A and B are memory location.**

I.

.data

A:.word 13

B:.word 5

.text

ldr r0,=A

ldr r1,=B

ldr r2,[r0]

ldr r3,[r1]

add r4,r2,r3

add r5,r3,r3,lsl #1

add r6,r4,r5

swi 0x11

