

LAB 08

ARMLite

8.1

1| mov r0, #15
2| str r0, .WriteSignedNum
3| mov r1, #dp1
4| str r1, .WriteString
5| mov r2, #dp2
6| str r2, .WriteString
7| mov r3, #num
8| ldr r3, .InputNum
9| mov r4, #dp3
10| str r4, .WriteString
11| sub r0, r0, r3
12| str r0, .WriteSignedNum
13| halt
14| dp1: .asciz "remaining\n"
15| dp2: .asciz "how many do you to remove (1-3)?\n\n"
16| num: .word 0
17| dp3: .asciz "the remaining match stick are"

PC 52
LR 0
SP 1048576
R12 0
R11 0
R10 0
R9 0
R8 0
R7 0
R6 0
R5 0
R4 104
R3 1
R2 63
R1 52
R0 14

Count 13
Current Instruction
Status bits NZCV 0000

how many do you to remove (1-3)?
the remaining match stick are 14
Program HALTED. STOP, LOAD or EDIT
1

Load Save Edit

Processor

Memory

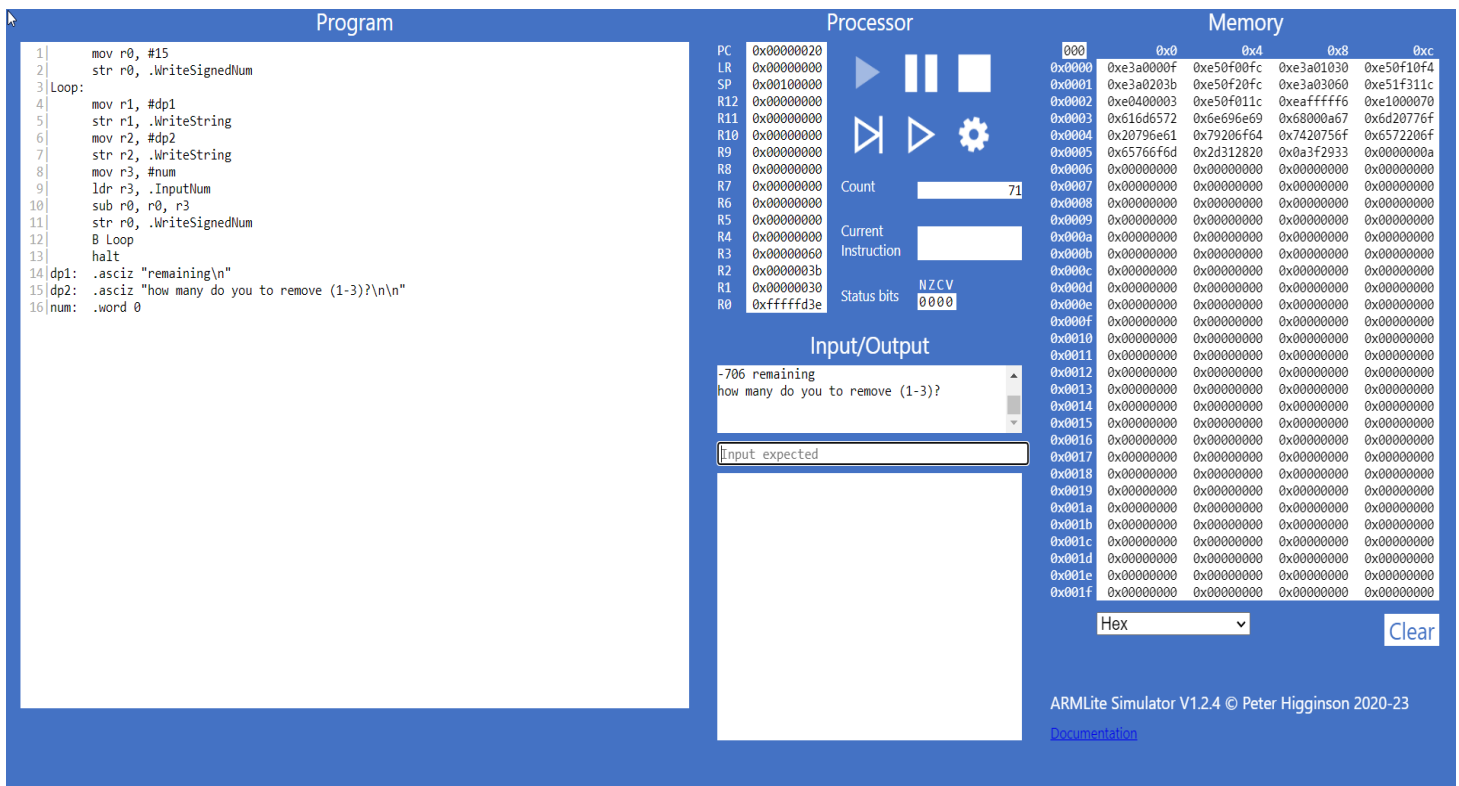
Input/Output

	0x0	0x4	0x8	0xc
0x0000	3818913807	3842965756	3818917940	3842969844
0x0001	3818922047	3842973948	3818926180	3844026652
0x0002	3818930280	3842982156	3762290691	3842965796
0x0003	3774873712	1634559346	1852403305	1744833127
0x0004	1830844271	544829025	2032168804	1948284271
0x0005	1701978223	1702260589	758196256	171911475
0x0006	10	0	543516788	1634559346
0x0007	1852403305	1634541671	543712116	1667855475
0x0008	1918967915	101	0	0
0x0009	0	0	0	0
0x000a	0	0	0	0
0x000b	0	0	0	0
0x000c	0	0	0	0
0x000d	0	0	0	0
0x000e	0	0	0	0
0x000f	0	0	0	0
0x0010	0	0	0	0
0x0011	0	0	0	0
0x0012	0	0	0	0
0x0013	0	0	0	0
0x0014	0	0	0	0
0x0015	0	0	0	0
0x0016	0	0	0	0
0x0017	0	0	0	0
0x0018	0	0	0	0
0x0019	0	0	0	0
0x001a	0	0	0	0
0x001b	0	0	0	0
0x001c	0	0	0	0
0x001d	0	0	0	0
0x001e	0	0	0	0
0x001f	0	0	0	0

Decimal (unsigned) Clear

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23
[Documentation](#)

8.2



If the number of match sticks left cross 0 the value or answer goes into negative (-706). By using 2s compliment principal

8.2.2(a)

What is the condition that needs to be satisfied in order for this loop to occur? Write this as a comparison using an inequality (ie., less than, greater than, less than or equal, greater than or equal)?

Ans: - The input value must be between 1-3

8.2.2(b)

What two ARM assembly instructions could be used to create a branch that only occurs under this condition?

Ans: - cmp, blt and bgt

8.2.2(c)

Based on the instructions you outlined in 8.2.2(b), what status bit would be set to 1 if the loop was to repeat?

Ans: - The N and Z bits should be set to 1 when the instruction is executed. When the entered value is less than 3, only N bit is set, when we compared entered value with 3 the result was in this case negative.

8.2.2(d)

What are all the modifications needed to the current program to implement this feature? Make the required modifications to your program to perform the task.

The screenshot shows the ARMLite Simulator V1.2.4. The Program window displays the following assembly code:

```

1 | mov r0, #15
2 | Loop: str r0, .WriteSignedNum
3 |      mov r1, #dp1
4 |      str r1, .WriteString
5 |      mov r2, #dp2
6 |      Loop2: str r2, .WriteString
7 |            mov r3, #num
8 |            ldr r3, .InputNum
9 |            cmp r3, #1
10 |           blt Loop2
11 |           b condition1
12 | condition1: cmp r3, #3
13 |            bgt Loop2
14 |            b condition2
15 | condition2: sub r0, r0, r3
16 |            cmp r0, #0
17 |            bgt Loop
18 |            str r5, .WriteString
19 |            halt
20 | dp1: .asciz "remaining\n"
21 | dp2: .asciz "how many do you to remove (1-3)?\n\n"
22 | dp3: .asciz "There are none Left!!!!"
23 | num: .word 0

```

The Processor window shows the following status:

- PC: 0x00000050
- LR: 0x00000000
- SP: 0x00100000
- R11: 0x00000000
- R10: 0x00000000
- R9: 0x00000000
- R8: 0x00000000
- R7: 0x00000000
- R6: 0x00000000
- R5: 0x00000000
- R4: 0x00000000
- R3: 0x00000003
- R2: 0x0000005b
- R1: 0x00000050
- R0: 0x00000000

The Memory window shows the following data:

000	0x0	0x4	0x8	0xc
0x0000	0xe3a0000f	0xe50f00fc	0xe3a01050	0xe50f10f4
0x0001	0xe3a0205b	0xe50f20fc	0xe3a03098	0xe51f311c
0x0002	0xe3530001	0xbaf0ffff	0xeaf0ffff	0xe3530003
0x0003	0xcac0ffff	0xeaf0ffff	0xe4000003	0xe3500000
0x0004	0xcac0ffff	0xe3a0507e	0xe50f5130	0xe1000070
0x0005	0x616d5572	0xe6e9e6e9	0x68000a67	0x6d2776f
0x0006	0x20796e61	0x79206f64	0x7420756f	0x6572206f
0x0007	0x65766f6d	0x2d312820	0x0a3f2933	0x6854000a
0x0008	0x20657265	0x20657261	0x656e6f6e	0x66054c20
0x0009	0x21212174	0x00002121	0x00000000	0x00000000
0x000a	0x00000000	0x00000000	0x00000000	0x00000000
0x000b	0x00000000	0x00000000	0x00000000	0x00000000
0x000c	0x00000000	0x00000000	0x00000000	0x00000000
0x000d	0x00000000	0x00000000	0x00000000	0x00000000
0x000e	0x00000000	0x00000000	0x00000000	0x00000000
0x000f	0x00000000	0x00000000	0x00000000	0x00000000
0x0010	0x00000000	0x00000000	0x00000000	0x00000000
0x0011	0x00000000	0x00000000	0x00000000	0x00000000
0x0012	0x00000000	0x00000000	0x00000000	0x00000000
0x0013	0x00000000	0x00000000	0x00000000	0x00000000
0x0014	0x00000000	0x00000000	0x00000000	0x00000000
0x0015	0x00000000	0x00000000	0x00000000	0x00000000
0x0016	0x00000000	0x00000000	0x00000000	0x00000000
0x0017	0x00000000	0x00000000	0x00000000	0x00000000
0x0018	0x00000000	0x00000000	0x00000000	0x00000000
0x0019	0x00000000	0x00000000	0x00000000	0x00000000
0x001a	0x00000000	0x00000000	0x00000000	0x00000000
0x001b	0x00000000	0x00000000	0x00000000	0x00000000
0x001c	0x00000000	0x00000000	0x00000000	0x00000000
0x001d	0x00000000	0x00000000	0x00000000	0x00000000
0x001e	0x00000000	0x00000000	0x00000000	0x00000000
0x001f	0x00000000	0x00000000	0x00000000	0x00000000

The Input/Output window shows the following text:

```

how many do you to remove (1-3)?
There are none Left!!!!
Program HALTED. STOP, LOAD or EDIT

```

8.3.1(a)

What bit-wise operation can we perform on the register holding the 32-bit pattern to set all bits in the register to zero except the least significant 2 bits? Write this as a single line of code.

Ans: - The bitwise operation is AND when we will use that operation with random number 0b000000000000000000000000000011.

8.3.1(b)

The screenshot shows the ARMLite Simulator V1.2.4. The Program window displays the following assembly code:

```

1 | select1:
2 | mov r4, #1
3 | and r4, r4, #3
4 | cmp r4, #0
5 | bgt select2
6 | b select2
7 | select2:
8 | mov r7, r4
9 | halt

```

The Processor window shows the following status:

- PC: 0x00000050
- LR: 0x00000000
- SP: 0x00100000
- R11: 0x00000000
- R10: 0x00000000
- R9: 0x00000000
- R8: 0x00000000
- R7: 0x00000000
- R6: 0x00000000
- R5: 0x00000000
- R4: 0x00000000
- R3: 0x00000000
- R2: 0x00000000
- R1: 0x00000000
- R0: 0x00000000

The Memory window shows the following data:

000	0x0	0x4	0x8	0xc
0x0000	0xe3a0000f	0xe50f00fc	0xe3a01050	0xe50f10f4
0x0001	0xe3a0205b	0xe50f20fc	0xe3a03098	0xe51f311c
0x0002	0xe3530001	0xbaf0ffff	0xeaf0ffff	0xe3530003
0x0003	0xcac0ffff	0xeaf0ffff	0xe4000003	0xe3500000
0x0004	0xcac0ffff	0xe3a0507e	0xe50f5130	0xe1000070
0x0005	0x616d5572	0xe6e9e6e9	0x68000a67	0x6d2776f
0x0006	0x20796e61	0x79206f64	0x7420756f	0x6572206f
0x0007	0x65766f6d	0x2d312820	0x0a3f2933	0x6854000a
0x0008	0x20657265	0x20657261	0x656e6f6e	0x66054c20
0x0009	0x21212174	0x00002121	0x00000000	0x00000000
0x000a	0x00000000	0x00000000	0x00000000	0x00000000
0x000b	0x00000000	0x00000000	0x00000000	0x00000000
0x000c	0x00000000	0x00000000	0x00000000	0x00000000
0x000d	0x00000000	0x00000000	0x00000000	0x00000000
0x000e	0x00000000	0x00000000	0x00000000	0x00000000
0x000f	0x00000000	0x00000000	0x00000000	0x00000000
0x0010	0x00000000	0x00000000	0x00000000	0x00000000
0x0011	0x00000000	0x00000000	0x00000000	0x00000000
0x0012	0x00000000	0x00000000	0x00000000	0x00000000
0x0013	0x00000000	0x00000000	0x00000000	0x00000000
0x0014	0x00000000	0x00000000	0x00000000	0x00000000
0x0015	0x00000000	0x00000000	0x00000000	0x00000000
0x0016	0x00000000	0x00000000	0x00000000	0x00000000
0x0017	0x00000000	0x00000000	0x00000000	0x00000000
0x0018	0x00000000	0x00000000	0x00000000	0x00000000
0x0019	0x00000000	0x00000000	0x00000000	0x00000000
0x001a	0x00000000	0x00000000	0x00000000	0x00000000
0x001b	0x00000000	0x00000000	0x00000000	0x00000000
0x001c	0x00000000	0x00000000	0x00000000	0x00000000
0x001d	0x00000000	0x00000000	0x00000000	0x00000000
0x001e	0x00000000	0x00000000	0x00000000	0x00000000
0x001f	0x00000000	0x00000000	0x00000000	0x00000000

The Input/Output window shows the following text:

```

Program HALTED. STOP, LOAD or EDIT

```

8.4

