Part 7.1

7.1.1/ Value "0x00000065 " appeared , this is due to the number 101 changing the memory from a decimal to a hex.
7.1.2/ Value "0x00000101" appeared, this is due to the 0x101 value is already a hex therefore no value was converted
7.1.3/ Value "0x00000005" appeared, this is due to the 0b101 changing the value from decimal to hex
        The tooltip shows the binary follow by the decimal form of the value which should be default set to hex
         If we change the default from hex to decimal (unsigned) , the toolship will show binary and hex.
7.1.4/ No, it will not change when you change the representation of the data.

Part 7.2

7.2.1/ These column header memory address offsets go up in multiples of 0x4 because there are 4 bits in each
        memory word.

Part 7.3.

7.3.1/

7.3.2/ When the submit button is toggled - the code (instruction set) is executed and saved into the memory. Even though the memory has been saved - the program will not execute as the RUN button has yet to be toggled.

7.3.3/

The pop-up tooltip's value presents the position of each code in the memory

When attempting to modify some part of the code , these events occur after hitting the SUBMIT button

- The extra lines added disappeared
- The additional space added before instruction disappeared
- Both comment lines stayed in place and turned green
- The line with the omitted comma turned red

Part 7.4

<u>7.4.1/</u> The highlighting in both windows signifies the line code which was being executed and the location in the memory in which the instruction has been stored.

<u>7.4.2/</u> Upon toggling the red button , the next set of instructions is executed in the program.

<u>7.4.3/</u> The processor paused just before executing the line at breakpoint

<u>Part 7.5</u>

<u>7.5.1/</u> Add r1,r0,#8: it will take the decimal value 8 and add it to the value in r0 and store the result in r1

<u>7.5.2/</u>

7.5.3/

Program panel:
```
1    MOV R2,#300
2    ADD R3,R2,#92
3    ADD R4,R3,#18
4    SUB R5,R4,#64
5    SUB R6,R5,#21
6    ADD R7,R6,#5
7    HALT
```

Processor panel:
```
PC              0
LR              0
SP        1048576
R12             0
R11             0
R10             0
R9              0
R8              0
R7              0
R6              0
R5              0
R4              0
R3              0
R2              0
R1              0
R0              0

Count                     0
Current Instruction
Status bits   NZCV
              0000
```

Input/Output:
```
Program assembled. Run or Step to execute
```

Memory panel:
```
          0x0        0x4        0x8        0xc
0x0000  3818925899 3800182876 3800252434 3796127808
0x0001  3796197397 3800461317 3774873712          0
0x0002           0          0          0          0
0x0003           0          0          0          0
0x0004           0          0          0          0
0x0005           0          0          0          0
0x0006           0          0          0          0
0x0007           0          0          0          0
0x0008           0          0          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
```
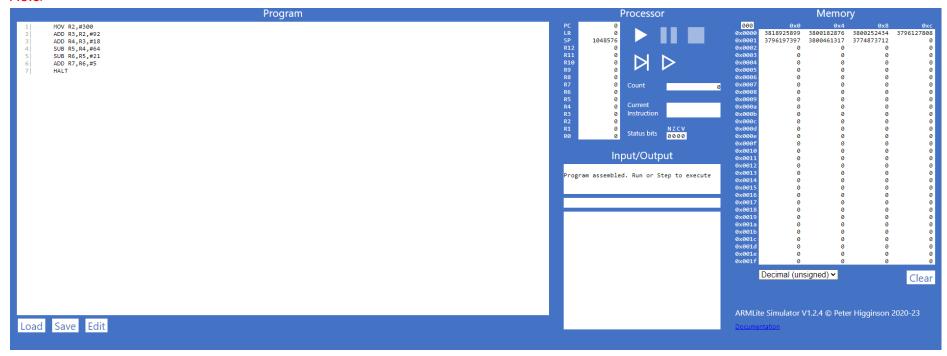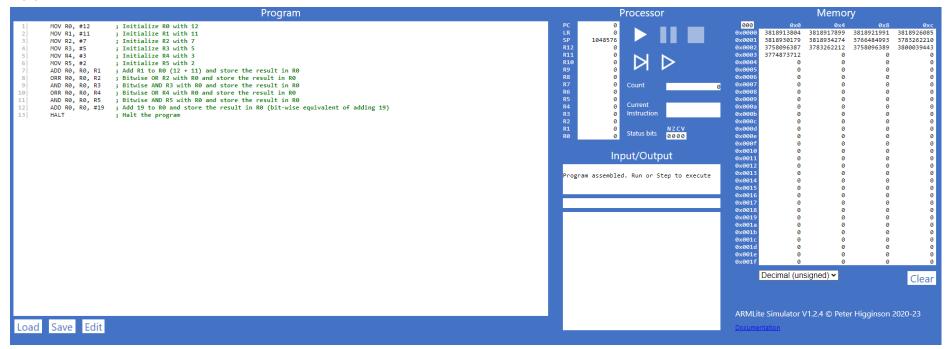Decimal (unsigned)   Clear

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23
Documentation

7.5.4/

Instruction	Decimal Value of the destination register after executing this instruction	Binary value of the destination register after executing this instruction
MOV R0,#1	1	0b00000000000000000000000000000001
AND R1,R0, #3	1	0b00000000000000000000000000000001
ORR R2,R1,#6	7	0b00000000000000000000000000000111

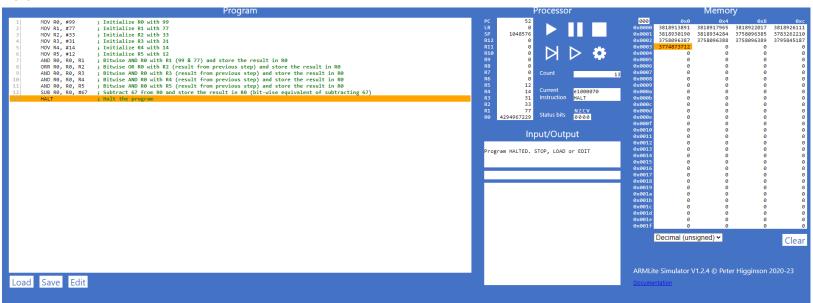| EOR R3,R2,#2	5	0b00000000000000000000000000000101
LSL R4, R3, #5	160	0b00000000000000000000000010100000
LSR R5, R2, R4	0	0

7.5.5/

The code

```
1|  MOV R0, #12     ; Initialize R0 with 12
2|  MOV R1, #11     ; Initialize R1 with 11
3|  MOV R2, #7      ; Initialize R2 with 7
```

```
4|  MOV R3, #5      ; Initialize R3 with 5
5|  MOV R4, #3      ; Initialize R4 with 3
6|  MOV R5, #2      ; Initialize R5 with 2
7|  ADD R0, R0, R1  ; Add R1 to R0 (12 + 11) and store the result in R0
8|  ORR R0, R0, R2  ; Bitwise OR R2 with R0 and store the result in R0
9|  AND R0, R0, R3  ; Bitwise AND R3 with R0 and store the result in R0
10| ORR R0, R0, R4  ; Bitwise OR R4 with R0 and store the result in R0
11| AND R0, R0, R5  ; Bitwise AND R5 with R0 and store the result in R0
12| ADD R0, R0, #19 ; Add 19 to R0 and store the result in R0 (bit-wise equivalent of adding 19)
13| HALT            ; Halt the program
```

7.5.6

The code

```
1|  MOV R0, #99     ; Initialize R0 with 99
```

```
2|  MOV R1, #77     ; Initialize R1 with 77
3|  MOV R2, #33     ; Initialize R2 with 33
4|  MOV R3, #31     ; Initialize R3 with 31
5|  MOV R4, #14     ; Initialize R4 with 14
6|  MOV R5, #12     ; Initialize R5 with 12
7|  AND R0, R0, R1  ; Bitwise AND R0 with R1 (99 & 77) and store the result in R0
8|  ORR R0, R0, R2  ; Bitwise OR R0 with R2 (result from previous step) and store the result in R0
9|  AND R0, R0, R3  ; Bitwise AND R0 with R3 (result from previous step) and store the result in R0
10| AND R0, R0, R4  ; Bitwise AND R0 with R4 (result from previous step) and store the result in R0
11| AND R0, R0, R5  ; Bitwise AND R0 with R5 (result from previous step) and store the result in R0
12| SUB R0, R0, #67 ; Subtract 67 from R0 and store the result in R0 (bit-wise equivalent of subtracting 67)
13| HALT            ; Halt the program
```

Part 7.6

7.6.1/ The result in R1 is R0 (in binary) moved to the left by 18 bits

7.6.2/ Binary of 1 is 0b00000000000000000000000000000001
        Binary of –1 is 0 b 11111111111111111111111111111111
        Binary of 2 is 0b00000000000000000000000000000010
        Binary of -2 is 0b11111111111111111111111111111110

7.6.3/

```
1|    MOV R2, #0
2|    MOV R0, #5
3|    SUB R1, R2,R0
4|    LSR R3, R1, #32
5|    HALT
```

PC   20
LR   0
SP   1048576
R12   0
R11   0
R10   0
R9   0
R8   0
R7   0
R6   0
R5   0
R4   0
R3   0
R2   0
R1   -5
R0   5

Count

Current
Instruction   <1000070
               HALT

Status bits   N Z C V
               0 0 0 0

Input/Output

Program HALTED. STOP, LOAD or EDIT

	0x0	0x4	0x8	0xc
0x0000	-476065312	-476053499	-532541460	-509595615
0x0001	-520003584	0	0	0
0x0002	0	0	0	0
0x0003	0	0	0	0
0x0004	0	0	0	0
0x0005	0	0	0	0
0x0006	0	0	0	0
0x0007	0	0	0	0
0x0008	0	0	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 (signed)   Clear

ARMLite Simulator V1.2 © Peter Higginson 2020

Documentation

Load Save Edit

7.6.4/

Program

```
1|    MOV R1, R0        ; Copy the positive integer from R0 to R1
2|    MVN R1, R1        ; Flip all the bits in R1 to get the negative version
```

Processor

PC   0
LR   0
SP   1048576
R12   0
R11   0
R10   0
R9   0
R8   0
R7   0
R6   0
R5   0
R4   0
R3   0
R2   0
R1   0
R0   0

Count

Current
Instruction

Status bits   N Z C V
               0 0 0 0

Input/Output

Program assembled. Run or Step to execute

Memory

	0x0	0x4	0x8	0xc
0x0000	3785363456	3789557761	0	0
0x0001	0	0	0	0
0x0002	0	0	0	0
0x0003	0	0	0	0
0x0004	0	0	0	0
0x0005	0	0	0	0
0x0006	0	0	0	0
0x0007	0	0	0	0
0x0008	0	0	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

Load Save Edit