

Lab 7 – Computer Systems (Mohamed Shafi Uzman Fassy - 102608927)

7.1.1

Memory				
000	0x0	0x4	0x8	0xc
0x0000	0x00000065	0x00000000	0x00000000	0x00000000

The memory word displays 0x00000065 because it converts the 101 decimal value into hexadecimal.

7.1.2

0x0001 0x00000101 0x00000000 0x00000000 0x00000000

Using 0x at the beginning of the value is a way of representing hexadecimal values therefore the memory word would display 0x00000101.

7.1.3

0x0002 0x00000005 0x00000000 0x00000000 0x00000000

Using 0b at the beginning of the value is a way of representing binary values therefore 101 digits are converted into hexadecimal values which would be '5'.

The ToolTip function displays the binary and decimal values of the hexadecimal memory word which has been displayed in the grid of memory addresses.

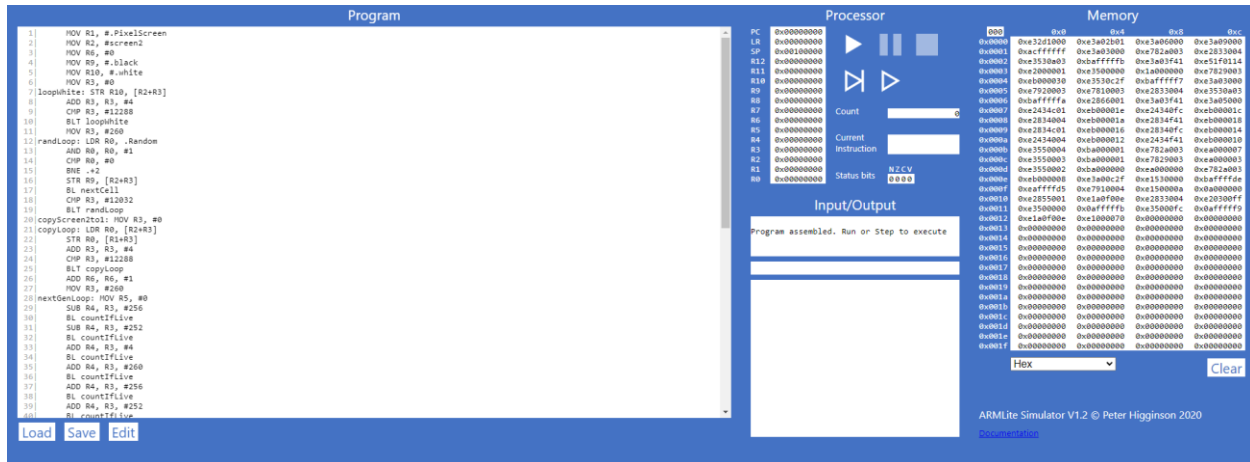
7.1.4

Changing the representation of the data in memory does not change the representation of rows and columns-headers as the rows and columns are there to represent the memory address location instead of a value.

7.2.1

Since each memory word cell represents a 32bit number therefore each hexadecimal digit is representing 4bits therefore each cell contains 8 hexadecimal values which represent a 4bit string.

7.3.1



7.3.2

Based on the Von Neuman Architecture, the memory that is used as data also uses instructions / line of code's memory to share the same common memory address which allows instructions / processes to be interrupted and stores data easily if needed while higher priority tasks are being executed.

7.3.3

The hexadecimal values are a representation of the memory locations of the line of code that is being executed on the memory space.

7.4.1

The highlighting indicates what line of code is being executed on the program window while the highlighting on the memory window highlights where that line of code has been stored on a memory address.

7.4.2

The Step button runs the code by executing the next line of code by a single step on each press of the button.

7.4.3

It pauses just before the breaking point.

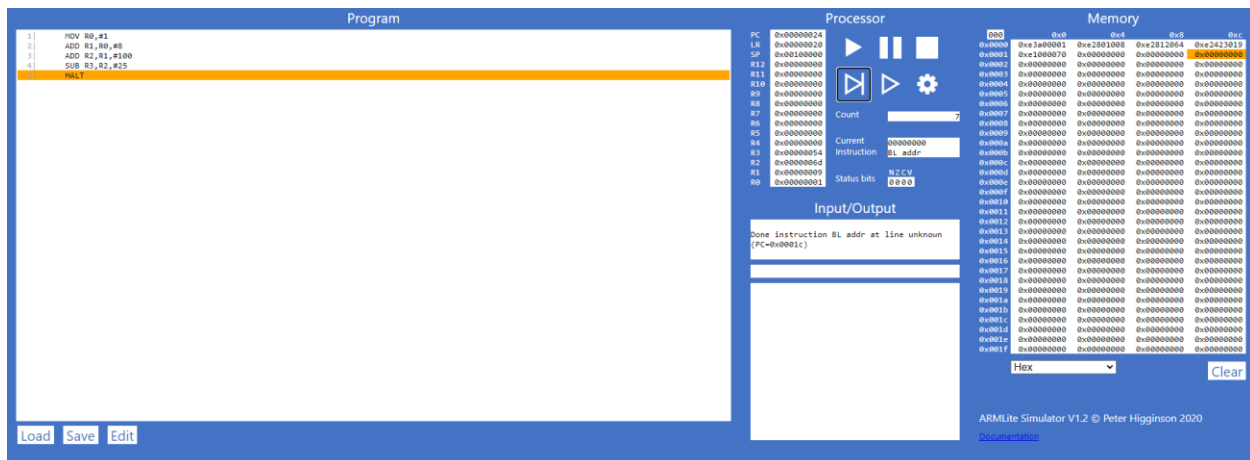
7.5.1

ADD R1,R0,#8 - It should add the value in R0 with the decimal value of 8 and store the value R1 while R0 value stays the same while R1's value should be 9.

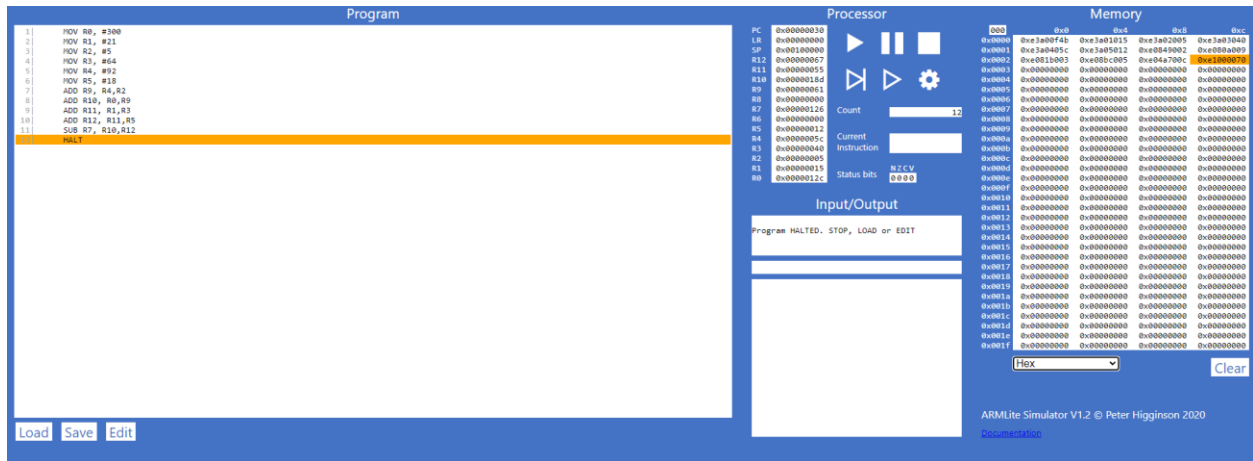
ADD R2,R1,#100 - should add the value 9 from R1, the decimal value of 100 and then store it in R2 which will display the hex value of 6d.

SUB R3,R2,#25 - should subtract the decimal value of 25 from R2 and store the final result in R3 which will represent the value as 54 but in decimal value 84 which is the target value.

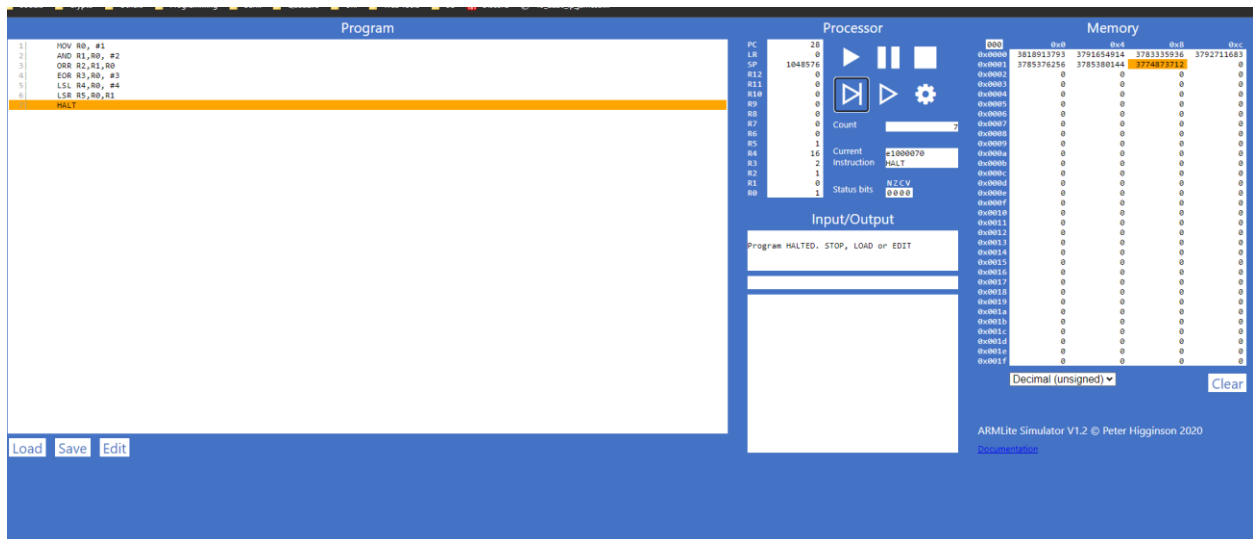
7.5.2



7.5.3



7.5.4



Instruction	Decimal value of the destination registers after executing this instruction	Binary value of the destination register after executing this instruction
MOV R0, #1	1	0b.....01
AND R1,R0, #2	0	0b.....00
ORR R2,R1,R0	1	0b.....01
EOR R3,R0, #3	2	0b.....10
LSL R4,R0, #4	16	0b.....10000
LSR R5,R0,R1	1	0b.....01

7.5.5

The screenshot displays the ARMLite Simulator V1.2 interface, which is divided into several sections:

- Program:** A list of assembly instructions with line numbers. The current instruction is `HALT` at line 12.

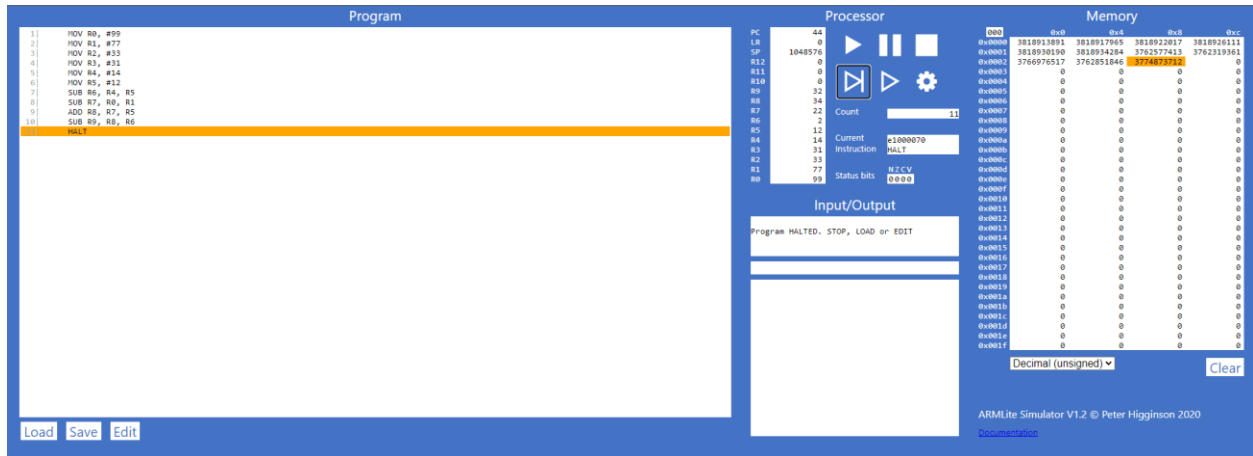

```

1 MOV R0, #12
2 MOV R1, #11
3 MOV R2, #7
4 MOV R3, #5
5 MOV R4, #3
6 MOV R5, #2
7 LSL R6, R0, #4
8 SUB R7, R6, R1
9 SUB R8, R7, R2
10 SUB R9, R8, R3
11 ADD R10, R9, R4
12 HALT
      
```
- Processor:** A panel showing the current state of the processor.
 - PC:** 52
 - SP:** 1048576
 - Count:** 15
 - Current Instruction:** `HALT`
 - Status bits:** `RZCV` (0000)
- Input/Output:** A section with a text area displaying `Program HALTED. STOP, LOAD or EDIT`.
- Memory:** A table showing memory addresses and their corresponding values.

Address	Value
0x0000	381891804
0x0001	3818938179
0x0002	3762782210
0x0003	3762851643
0x0004	3767157980
0x0005	3767185412
0x0006	0
0x0007	0
0x0008	0
0x0009	0
0x000a	0
0x000b	0
0x000c	0
0x000d	0
0x000e	0
0x000f	0
0x0010	0
0x0011	0
0x0012	0
0x0013	0
0x0014	0
0x0015	0
0x0016	0
0x0017	0
0x0018	0
0x0019	0
0x001a	0
0x001b	0
0x001c	0
0x001d	0
0x001e	0

At the bottom of the interface, there are buttons for `Load`, `Save`, and `Edit`. The footer text reads: `ARMLite Simulator V1.2 © Peter Higginson 2020` with a link to [Documentation](#).

7.5.6



7.6.1

Because the command shifts by 18 digits from the right to left from R0 value which is in binary and assigns to the value in R1 as R1 has the value that does not relate to 9999.

7.6.3

For 1: #1

For -1: #4294967295 [$2^{32}-1$]

For 2: #2

For -2: #4294967292 [$2^{32}-2$]

7.6.4

[illegible]