Lab 10 - Flashy Functions

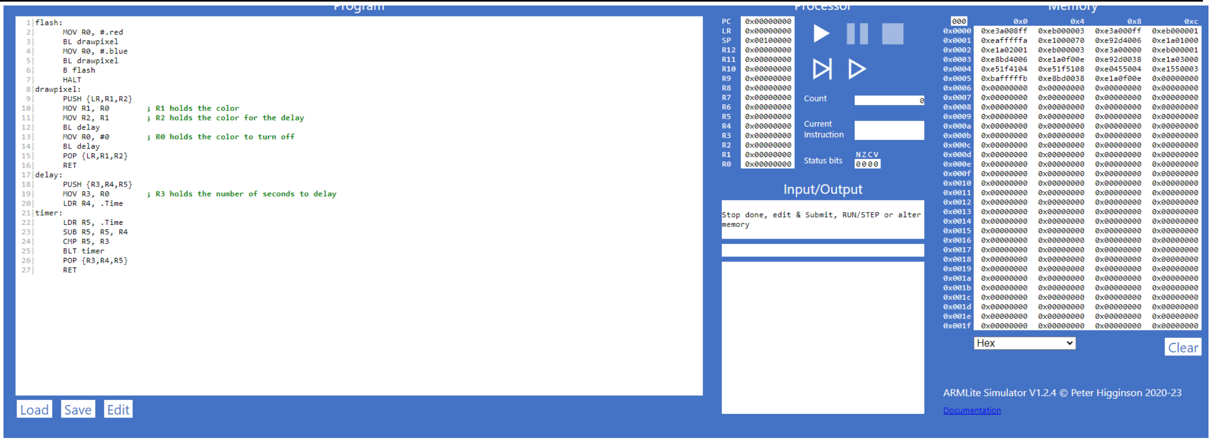
10.1/

(a) First write the delay function. This function should take a single input, the number of seconds to delay for and be called from the main program everytime there is a pause required.

```
17|delay:
18|      PUSH {R3,R4,R5}
19|      MOV R3, R0       ; R3 holds the number of seconds to delay
20|      LDR R4, .Time
```

(b) Then write the drawpixel function. This function should take two inputs: the colour of the pixel to draw, and the time delay between on and off. This function should also call the delay function to insert the pauses between on and off.

```
 8|drawpixel:
 9|     PUSH {LR,R1,R2}
10|      MOV R1, R0       ; R1 holds the color
11|      MOV R2, R1       ; R2 holds the color for the delay
12|      BL delay
13|      MOV R0, #0       ; R0 holds the color to turn off
14|      BL delay
15|      POP {LR,R1,R2}
16|      RET
```



```
 1|flash:
 2|      MOV R0, #.red
 3|      BL drawpixel
 4|      MOV R0, #.blue
 5|      BL drawpixel
 6|      B flash
 7|      HALT
 8|drawpixel:
 9|      PUSH {LR,R1,R2}
10|      MOV R1, R0        ; R1 holds the color
11|      MOV R2, R1        ; R2 holds the color for the delay
12|      BL delay
13|      MOV R0, #0        ; R0 holds the color to turn off
14|      BL delay
15|      POP {LR,R1,R2}
16|      RET
17|delay:
18|      PUSH {R3,R4,R5}
19|      MOV R3, R0        ; R3 holds the number of seconds to delay
20|      LDR R4, .Time
21|timer:
22|      LDR R5, .Time
23|      SUB R5, R5, R4
24|      CMP R5, R3
25|      BLT timer
26|      POP {R3,R4,R5}
27|      RET
```

```
PC   0x00000000
LR   0x00000000
SP   0x00100000
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000000
R6   0x00000000
R5   0x00000000
R4   0x00000000
R3   0x00000000
R2   0x00000000
R1   0x00000000
R0   0x00000000

Count 0
Status bits NZCV 0000

Stop done, edit & Submit, RUN/STEP or alter memory
```

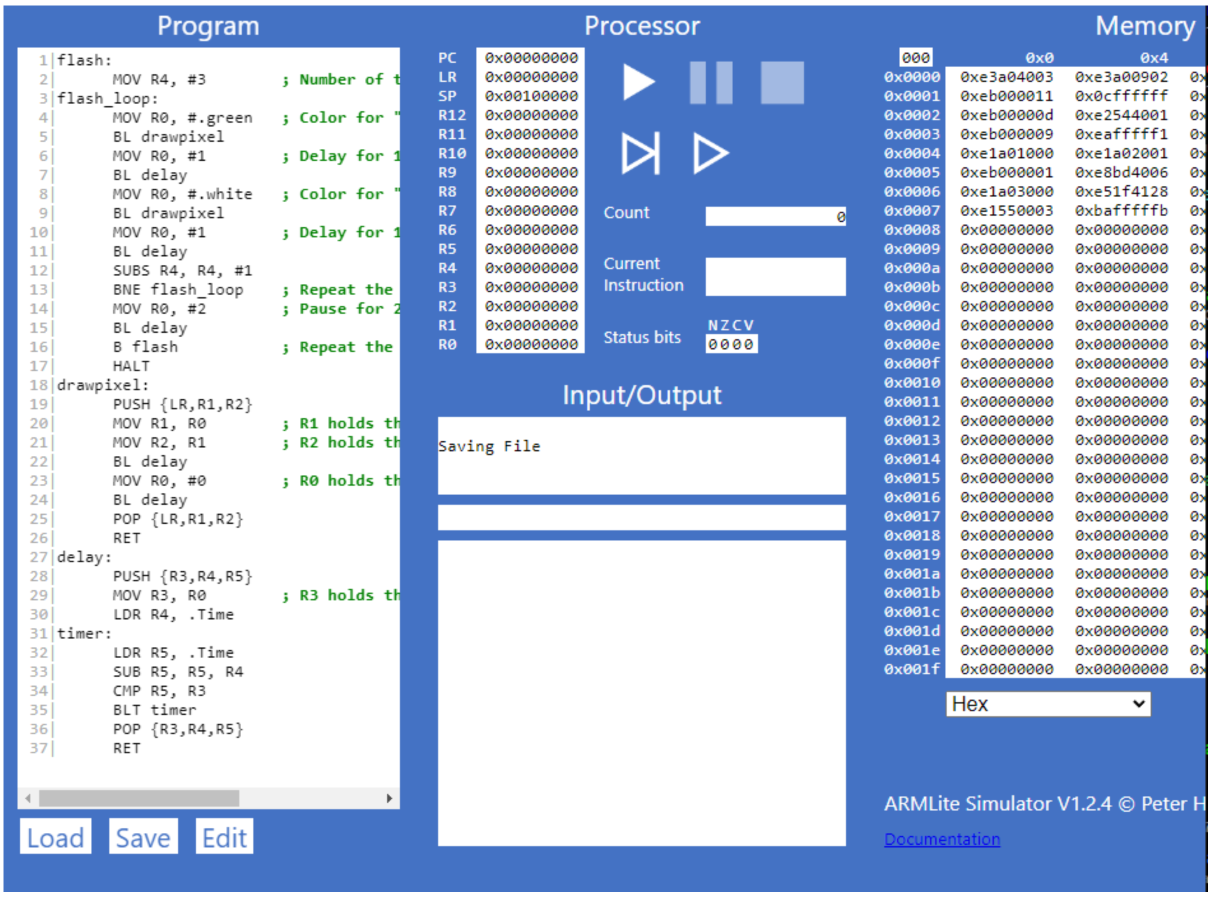
```
000           0x0         0x4         0x8         0xc
0x0000  0xe3a008ff  0xeb000003  0xe3a000ff  0xeb000001
0x0001  0xeafffffa  0xe1000070  0xe92d4006  0xe1a01000
0x0002  0xe1a02001  0xeb000003  0xe3a00000  0xeb000001
0x0003  0xe8bd4006  0xe1a0f00e  0xe92d0038  0xe1a03000
0x0004  0xe51f4104  0xe51f5108  0xe0455004  0xe1550003
0x0005  0xbafffffb  0xe8bd0038  0xe1a0f00e  0x00000000
0x0006  0x00000000  0x00000000  0x00000000  0x00000000
0x0007  0x00000000  0x00000000  0x00000000  0x00000000
0x0008  0x00000000  0x00000000  0x00000000  0x00000000
0x0009  0x00000000  0x00000000  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
```

(c)When you implemented drawpixel, what did you have to do with LR to make it work ? Why?

To make the drawpixel **function work, we had to preserve the value of LR on the stack (line 9) before branching to the** delay **function (line 12). This is because the** delay **function may modify the LR register, and we need to ensure that LR is properly restored when returning from the** delay **function. The POP**

instruction on line 15 restores LR, R1, and R2 to their previous values before returning from the drawpixel **function.**

10.2/



Program

```
 1|flash:
 2|      MOV R4, #3          ; Number of t
 3|flash_loop:
 4|      MOV R0, #.green     ; Color for "
 5|      BL drawpixel
 6|      MOV R0, #1          ; Delay for 1
 7|      BL delay
 8|      MOV R0, #.white     ; Color for "
 9|      BL drawpixel
10|      MOV R0, #1          ; Delay for 1
11|      BL delay
12|      SUBS R4, R4, #1
13|      BNE flash_loop      ; Repeat the
14|      MOV R0, #2          ; Pause for 2
15|      BL delay
16|      B flash             ; Repeat the
17|      HALT
18|drawpixel:
19|      PUSH {LR,R1,R2}
20|      MOV R1, R0          ; R1 holds th
21|      MOV R2, R1          ; R2 holds th
22|      BL delay
23|      MOV R0, #0          ; R0 holds th
24|      BL delay
25|      POP {LR,R1,R2}
26|      RET
27|delay:
28|      PUSH {R3,R4,R5}
29|      MOV R3, R0          ; R3 holds th
30|      LDR R4, .Time
31|timer:
32|      LDR R5, .Time
33|      SUB R5, R5, R4
34|      CMP R5, R3
35|      BLT timer
36|      POP {R3,R4,R5}
37|      RET
```

Processor

```
PC   0x00000000
LR   0x00000000
SP   0x00100000
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000000
R6   0x00000000
R5   0x00000000
R4   0x00000000
R3   0x00000000
R2   0x00000000
R1   0x00000000
R0   0x00000000

Count                 0
Current Instruction
Status bits  N Z C V
             0 0 0 0
```

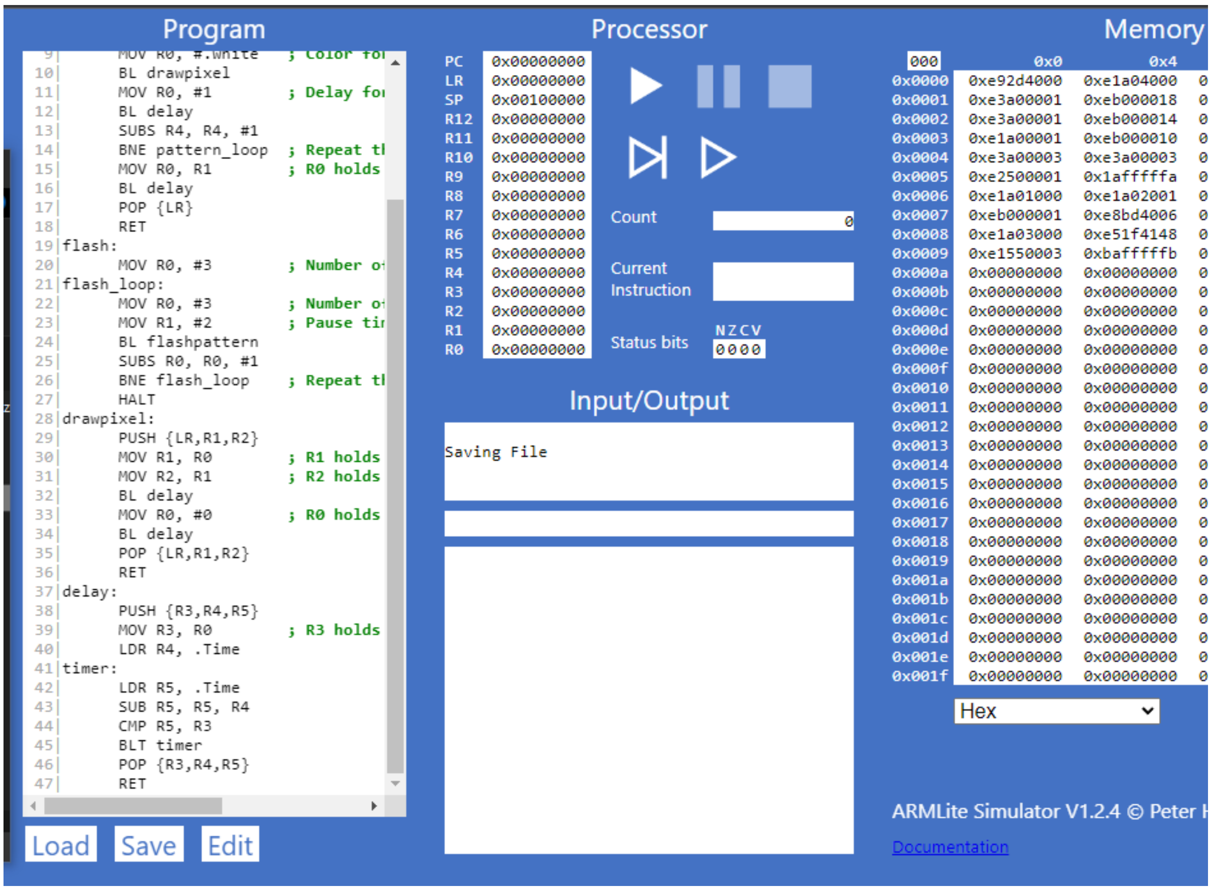
Input/Output

```
Saving File
```

Memory

```
000          0x0         0x4
0x0000  0xe3a04003  0xe3a00902  0x
0x0001  0xeb000011  0x0cffffff  0x
0x0002  0xeb00000d  0xe2544001  0x
0x0003  0xeb000009  0xeafffff1  0x
0x0004  0xe1a01000  0xe1a02001  0x
0x0005  0xeb000001  0xe8bd4006  0x
0x0006  0xe1a03000  0xe51f4128  0x
0x0007  0xe1550003  0xbafffffb  0x
0x0008  0x00000000  0x00000000  0x
0x0009  0x00000000  0x00000000  0x
0x000a  0x00000000  0x00000000  0x
0x000b  0x00000000  0x00000000  0x
0x000c  0x00000000  0x00000000  0x
0x000d  0x00000000  0x00000000  0x
0x000e  0x00000000  0x00000000  0x
0x000f  0x00000000  0x00000000  0x
0x0010  0x00000000  0x00000000  0x
0x0011  0x00000000  0x00000000  0x
0x0012  0x00000000  0x00000000  0x
0x0013  0x00000000  0x00000000  0x
0x0014  0x00000000  0x00000000  0x
0x0015  0x00000000  0x00000000  0x
0x0016  0x00000000  0x00000000  0x
0x0017  0x00000000  0x00000000  0x
0x0018  0x00000000  0x00000000  0x
0x0019  0x00000000  0x00000000  0x
0x001a  0x00000000  0x00000000  0x
0x001b  0x00000000  0x00000000  0x
0x001c  0x00000000  0x00000000  0x
0x001d  0x00000000  0x00000000  0x
0x001e  0x00000000  0x00000000  0x
0x001f  0x00000000  0x00000000  0x
```

10.3/



```
        MOV R0, #.white    ; Color for
        BL drawpixel
        MOV R0, #1         ; Delay for
        BL delay
        SUBS R4, R4, #1
        BNE pattern_loop   ; Repeat th
        MOV R0, R1         ; R0 holds
        BL delay
        POP {LR}
        RET
flash:
        MOV R0, #3         ; Number of
flash_loop:
        MOV R0, #3         ; Number of
        MOV R1, #2         ; Pause tim
        BL flashpattern
        SUBS R0, R0, #1
        BNE flash_loop     ; Repeat th
        HALT
drawpixel:
        PUSH {LR,R1,R2}
        MOV R1, R0         ; R1 holds
        MOV R2, R1         ; R2 holds
        BL delay
        MOV R0, #0         ; R0 holds
        BL delay
        POP {LR,R1,R2}
        RET
delay:
        PUSH {R3,R4,R5}
        MOV R3, R0         ; R3 holds
        LDR R4, .Time
timer:
        LDR R5, .Time
        SUB R5, R5, R4
        CMP R5, R3
        BLT timer
        POP {R3,R4,R5}
        RET
```

```
PC   0x00000000
LR   0x00000000
SP   0x00100000
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000000
R6   0x00000000
R5   0x00000000
R4   0x00000000
R3   0x00000000
R2   0x00000000
R1   0x00000000
R0   0x00000000

Count                 0
Current Instruction
Status bits   NZCV
              0000
```

```
Saving File
```

```
000          0x0         0x4
0x0000   0xe92d4000  0xe1a04000  0
0x0001   0xe3a00001  0xeb000018  0
0x0002   0xe3a00001  0xeb000014  0
0x0003   0xe1a00001  0xeb000010  0
0x0004   0xe3a00003  0xe3a00003  0
0x0005   0xe2500001  0x1afffffa  0
0x0006   0xe1a01000  0xe1a02001  0
0x0007   0xeb000001  0xe8bd4006  0
0x0008   0xe1a03000  0xe51f4148  0
0x0009   0xe1550003  0xbafffffb  0
0x000a   0x00000000  0x00000000  0
0x000b   0x00000000  0x00000000  0
0x000c   0x00000000  0x00000000  0
0x000d   0x00000000  0x00000000  0
0x000e   0x00000000  0x00000000  0
0x000f   0x00000000  0x00000000  0
0x0010   0x00000000  0x00000000  0
0x0011   0x00000000  0x00000000  0
0x0012   0x00000000  0x00000000  0
0x0013   0x00000000  0x00000000  0
0x0014   0x00000000  0x00000000  0
0x0015   0x00000000  0x00000000  0
0x0016   0x00000000  0x00000000  0
0x0017   0x00000000  0x00000000  0
0x0018   0x00000000  0x00000000  0
0x0019   0x00000000  0x00000000  0
0x001a   0x00000000  0x00000000  0
0x001b   0x00000000  0x00000000  0
0x001c   0x00000000  0x00000000  0
0x001d   0x00000000  0x00000000  0
0x001e   0x00000000  0x00000000  0
0x001f   0x00000000  0x00000000  0
```

ARMLite Simulator V1.2.4 © Peter H

Documentation