Faculty of Science, Engineering and Technology

Computer Systems

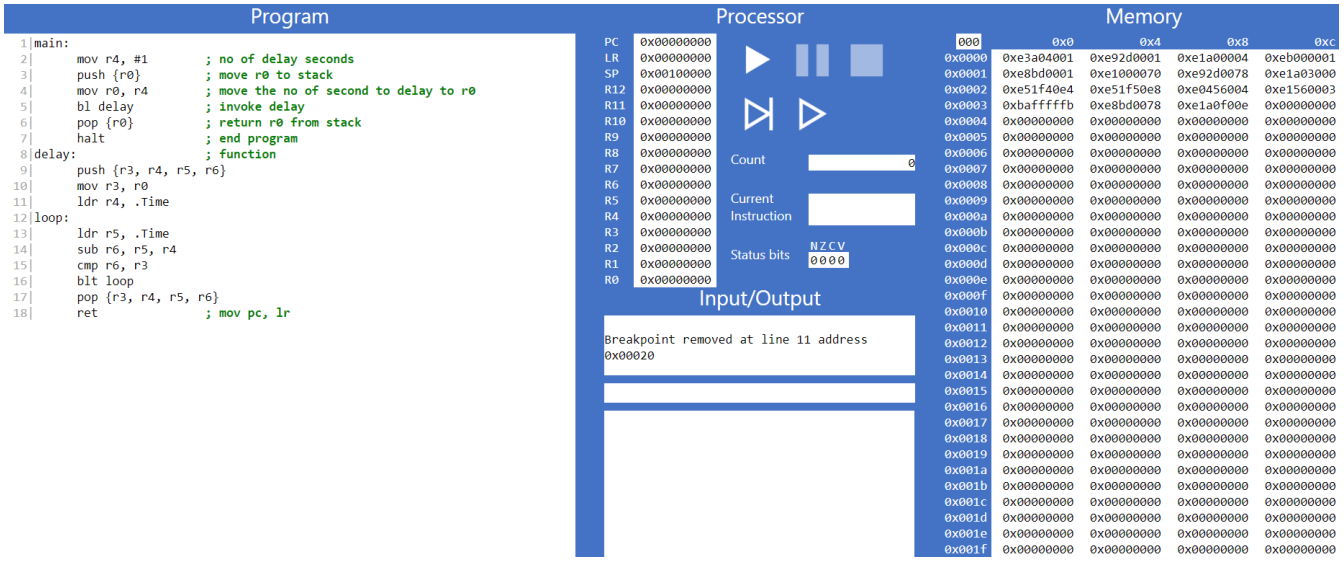
Week 10

Overview

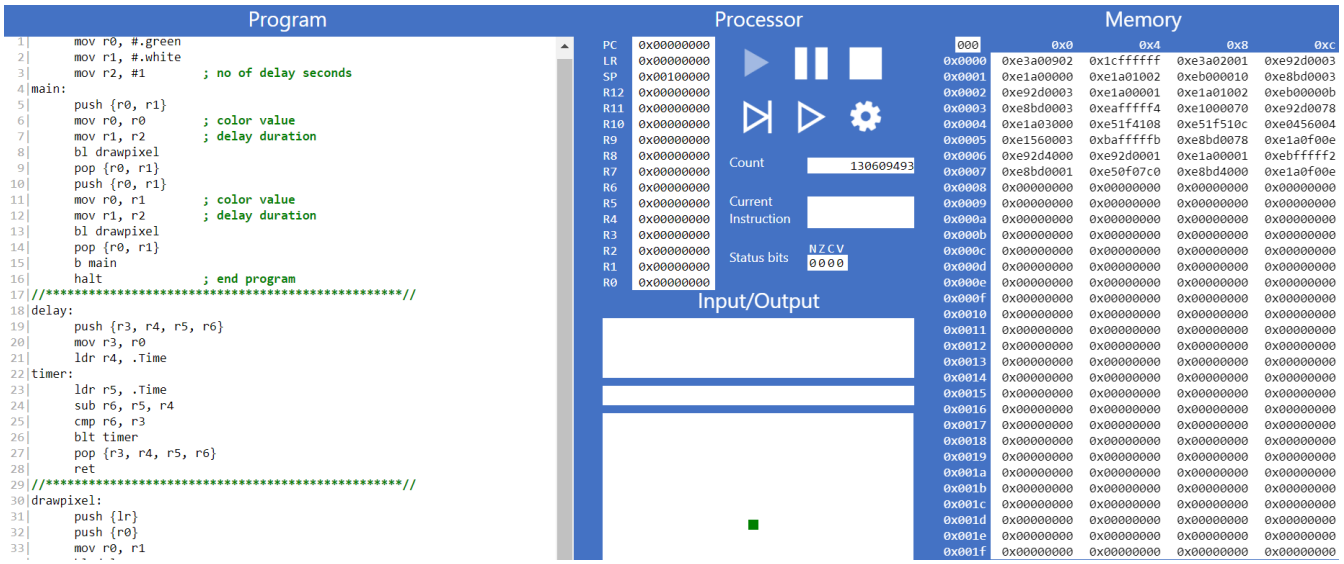
Name: Le Quang Hai
Student ID: 104175779

Exercise 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.



(b) Then write the drawpixel function. This function should take two inputs: the color 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.

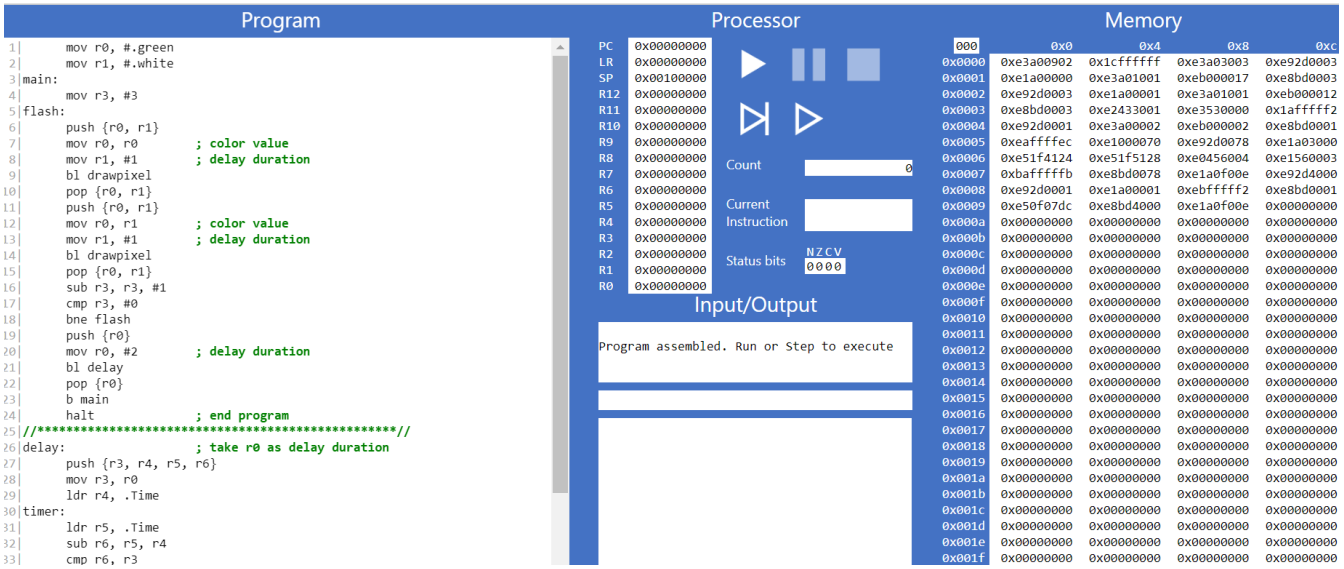



```
1       mov r0, #.green
2       mov r1, #.white
3       mov r2, #1          ; no of delay seconds
4   main:
5       push {r0, r1}
6       mov r0, r0          ; color value
7       mov r1, r2          ; delay duration
8       bl drawpixel
9       pop {r0, r1}
10      push {r0, r1}
11      mov r0, r1          ; color value
12      mov r1, r2          ; delay duration
13      bl drawpixel
14      pop {r0, r1}
15      b main
16      halt                ; end program
17  //************************************************//
18  delay:
19      push {r3, r4, r5, r6}
20      mov r3, r0
21      ldr r4, .Time
22  timer:
23      ldr r5, .Time
24      sub r6, r5, r4
25      cmp r6, r3
26      blt timer
27      pop {r3, r4, r5, r6}
28      ret
29  //************************************************//
30  drawpixel:
31      push {lr}
32      push {r0}
33      mov r0, r1
```

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

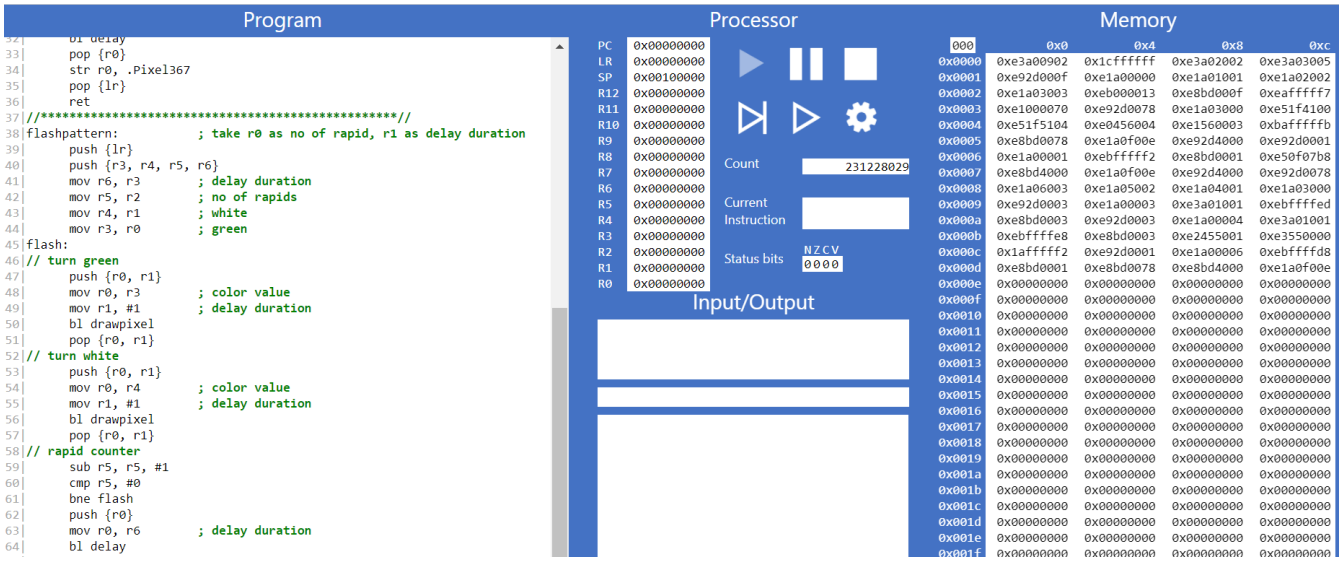
Ans: I have to push {lr} at the beginning of the drawpixel function so that the value of the next instruction address is stored in the stack because the lr value will change once I invoke another function from inside a function

Exercise 10.2




```
1       mov r0, #.green
2       mov r1, #.white
3   main:
4       mov r3, #3
5   flash:
6       push {r0, r1}
7       mov r0, r0          ; color value
8       mov r1, #1          ; delay duration
9       bl drawpixel
10      pop {r0, r1}
11      push {r0, r1}
12      mov r0, r1          ; color value
13      mov r1, #1          ; delay duration
14      bl drawpixel
15      pop {r0, r1}
16      sub r3, r3, #1
17      cmp r3, #0
18      bne flash
19      push {r0}
20      mov r0, #2          ; delay duration
21      bl delay
22      pop {r0}
23      b main
24      halt                ; end program
25  //************************************************//
26  delay:                  ; take r0 as delay duration
27      push {r3, r4, r5, r6}
28      mov r3, r0
29      ldr r4, .Time
30  timer:
31      ldr r5, .Time
32      sub r6, r5, r4
33      cmp r6, r3
```

Exercise 10.3




```
33      pop {r0}
34      str r0, .Pixel367
35      pop {lr}
36      ret
37  //************************************************//
38  flashpattern:           ; take r0 as no of rapid, r1 as delay duration
39      push {lr}
40      push {r3, r4, r5, r6}
41      mov r6, r3          ; delay duration
42      mov r5, r2          ; no of rapids
43      mov r4, r1          ; white
44      mov r3, r0          ; green
45  flash:
46  // turn green
47      push {r0, r1}
48      mov r0, r3          ; color value
49      mov r1, #1          ; delay duration
50      bl drawpixel
51      pop {r0, r1}
52  // turn white
53      push {r0, r1}
54      mov r0, r4          ; color value
55      mov r1, #1          ; delay duration
56      bl drawpixel
57      pop {r0, r1}
58  // rapid counter
59      sub r5, r5, #1
60      cmp r5, #0
61      bne flash
62      push {r0}
63      mov r0, r6          ; delay duration
64      bl delay
```