

Week 10 Lab - Part 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 every time there is a pause required.

The screenshot shows the ARMite Assembly Language Simulator interface. The main window displays the following assembly code:

```
1 MOV R0, #1
2 PUSH {R3, R4, R5, R6}
3 MOV R3, R0
4 LDR R4, .Time
5 timer:
6 LDR R5, .Time
7 SUB R6, R5, R4
8 CMP R6, R3
9 BLT timer
10 POP {R3, R4, R5, R6}
11 RET
```

The interface also includes a Processor section with registers (PC, LR, SP, R12, R11, R10, R9, R8, R7, R6, R5, R4, R3, R2, R1, R0) and their values, a Count register, Current Instruction, and Status bits (NZCV). The Memory section shows a hex dump of memory addresses from 0x0000 to 0x001f. The Input/Output section has a text area for memory input. The bottom status bar shows the temperature (14°C Cloudy) and the date (23/10/2022).

- (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.

The screenshot shows the ARMite Assembly Language Simulator V1.2a. The Program window contains the following assembly code:

```

1 MOV R1, #1
2 drawpixel:
3   PUSH {R4, R5}
4   MOV R4, R0
5   MOV R5, R1
6   STR R4, .Pixel367
7   PUSH {R0, LR}
8   MOV R0, R4
9   BL delay
10  POP {R0, LR}
11  POP {R4, R5}
12  RET
13 delay:

```

The Processor window shows the PC at 0x00000000, LR at 0x00000000, SP at 0x00100000, and R0 at 0x00000000. The status bits are NZCV 0000. The Memory window shows a hex dump starting with 0x00000000.

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

This LR makes it return to the initial function of the whole code instead of it to loop around the drawpixel function and staying stagnant.

The screenshot shows the ARMite Assembly Language Simulator V1.2a. The Program window contains the following assembly code:

```

1 mov R2, #1
2 flash:
3   MOV R0, #.green
4   MOV R1, R2
5   BL drawpixel
6   MOV R0, #.white
7   MOV R1, R2
8   BL drawpixel
9   B flash
10  HALT
11 drawpixel:
12  PUSH {R3,R4}
13  MOV R3, R0
14  MOV R4, R1
15  STR R3, .Pixel367
16  PUSH {R0, LR}
17  MOV R0, R4
18  BL delay
19  POP {R0, LR}
20  POP {R3,R4}
21  RET
22 delay:
23  push {R3,R4,R5,R6}
24  MOV R3, R0
25  LDR R4, .Time
26  timer:
27  LDR R5, .Time
28  SUB R6, R5, R4
29  CMP R6, R3
30  BLT timer
31  pop {R3,R4,R5,R6}
32  RET

```

The Processor window shows the PC at 0x00000000, LR at 0x00000000, SP at 0x00100000, and R0 at 0x00000000. The status bits are NZCV 0000. The Memory window shows a hex dump starting with 0x00000000.