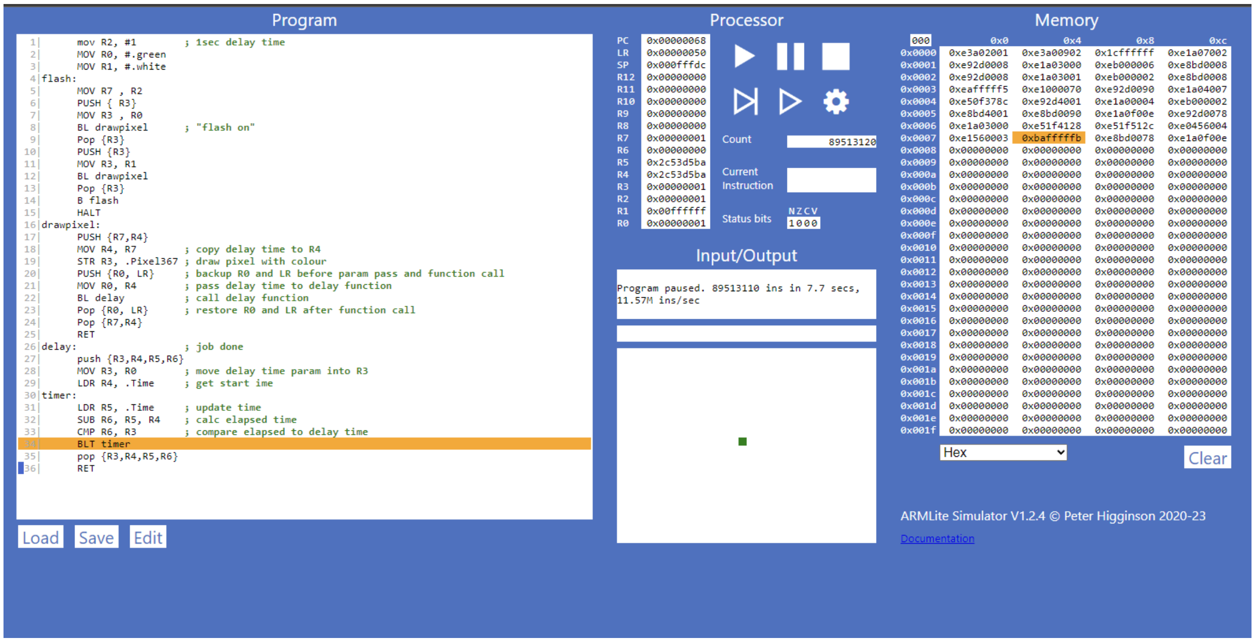
Name : Nguyen Quoc Thang

ID : 104193360

Lab10 : SUBMISSION

Excercise 10.1 a)



```
      mov R2, #1        ; 1sec delay time
      MOV R0, #.green
      MOV R1, #.white
flash:
      MOV R7 , R2
      PUSH { R3}
      MOV R3 , R0
      BL drawpixel      ; "flash on"
      Pop {R3}
      PUSH {R3}
      MOV R3, R1
      BL drawpixel
      Pop {R3}
      B flash
      HALT
drawpixel:
      PUSH {R7,R4}
      MOV R4, R7        ; copy delay time to R4
      STR R3, .Pixel367 ; draw pixel with colour
      PUSH {R0, LR}     ; backup R0 and LR before param pass and function call
      MOV R0, R4        ; pass delay time to delay function
      BL delay          ; call delay function
      Pop {R0, LR}      ; restore R0 and LR after function call
      Pop {R7,R4}
      RET
delay:                  ; job done
      push {R3,R4,R5,R6}
      MOV R3, R0        ; move delay time param into R3
      LDR R4, .Time     ; get start ime
timer:
      LDR R5, .Time     ; update time
      SUB R6, R5, R4    ; calc elapsed time
      CMP R6, R3        ; compare elapsed to delay time
      BLT timer
      pop {R3,R4,R5,R6}
      RET
```

```
PC   0x00000068
LR   0x00000050
SP   0x000fffdc
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000001
R6   0x00000000
R5   0x2c53d5ba
R4   0x2c53d5ba
R3   0x00000001
R2   0x00000001
R1   0x00ffffff
R0   0x00000001
```

Count 89513120

Current Instruction

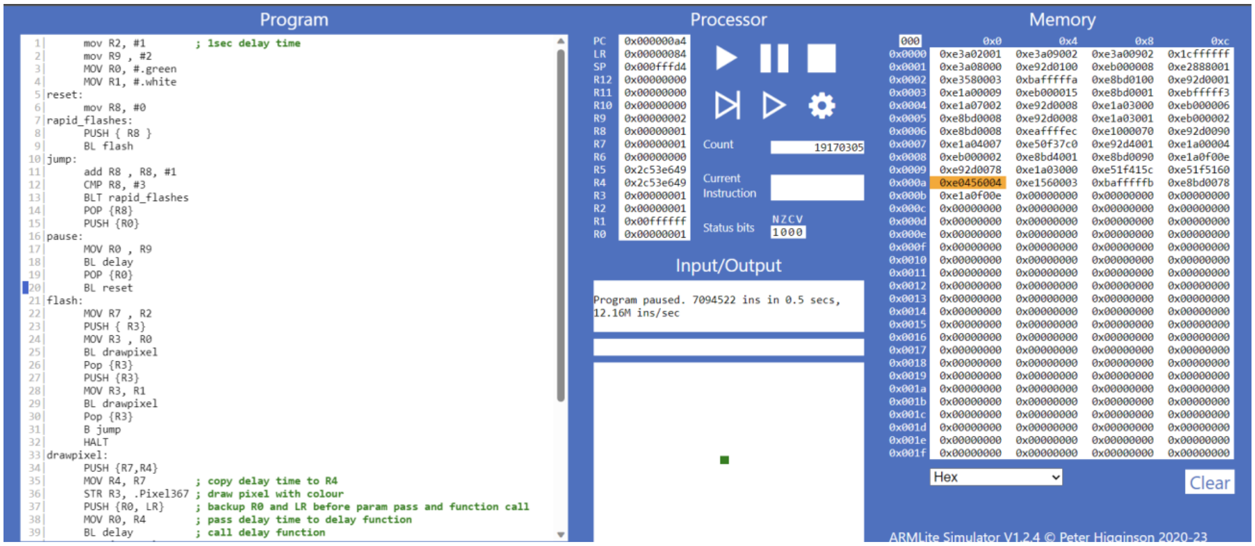
Status bits NZCV 1000

```
Program paused. 89513110 ins in 7.7 secs,
11.57M ins/sec
```

```
000            0x0         0x4         0x8         0xc
0x0000  0xe3a02001  0xe3a00902  0x1cffffff  0xe1a07002
0x0001  0xe92d0008  0xe1a03000  0xeb000006  0xe8bd0008
0x0002  0xe92d0008  0xe1a03001  0xeb000002  0xe8bd0008
0x0003  0xeafffff5  0xe1000070  0xe92d0090  0xe1a04007
0x0004  0xe50f378c  0xe92d4001  0xe1a00004  0xeb000002
0x0005  0xe8bd4001  0xe8bd0090  0xe1a0f00e  0xe92d0078
0x0006  0xe1a03000  0xe51f4128  0xe51f512c  0xe0456004
0x0007  0xe1560003  0xbafffffb  0xe8bd0078  0xe1a0f00e
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) Push LR into the stack to store the address of instruction and when the function complete it comback to instruction behind the function and continue execute the next instruction

Excercise 10.2



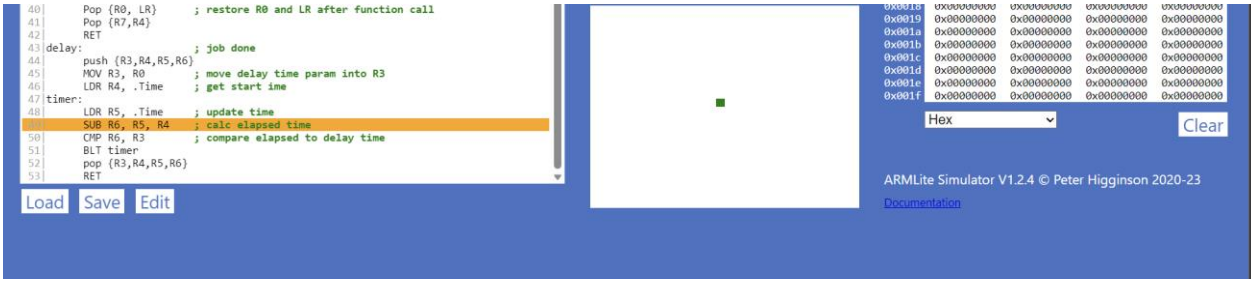
```
        mov R2, #1          ; 1sec delay time
        mov R9 , #2
        MOV R0, #.green
        MOV R1, #.white
reset:
        mov R8, #0
rapid_flashes:
        PUSH { R8 }
        BL flash
jump:
        add R8 , R8, #1
        CMP R8, #3
        BLT rapid_flashes
        POP {R8}
        PUSH {R0}
pause:
        MOV R0 , R9
        BL delay
        POP {R0}
        BL reset
flash:
        MOV R7 , R2
        PUSH { R3}
        MOV R3 , R0
        BL drawpixel
        Pop {R3}
        PUSH {R3}
        MOV R3, R1
        BL drawpixel
        Pop {R3}
        B jump
        HALT
drawpixel:
        PUSH {R7,R4}
        MOV R4, R7          ; copy delay time to R4
        STR R3, .Pixel367 ; draw pixel with colour
        PUSH {R0, LR}       ; backup R0 and LR before param pass and function call
        MOV R0, R4          ; pass delay time to delay function
        BL delay            ; call delay function
```

```
PC   0x000000a4
LR   0x00000084
SP   0x000fffd4
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000002
R8   0x00000001
R7   0x00000001
R6   0x00000000
R5   0x2c53e649
R4   0x2c53e649
R3   0x00000001
R2   0x00000001
R1   0x00ffffff
R0   0x00000001

Count          19170305
Status bits    NZCV
               1000
```

```
Program paused. 7094522 ins in 0.5 secs,
12.16M ins/sec
```

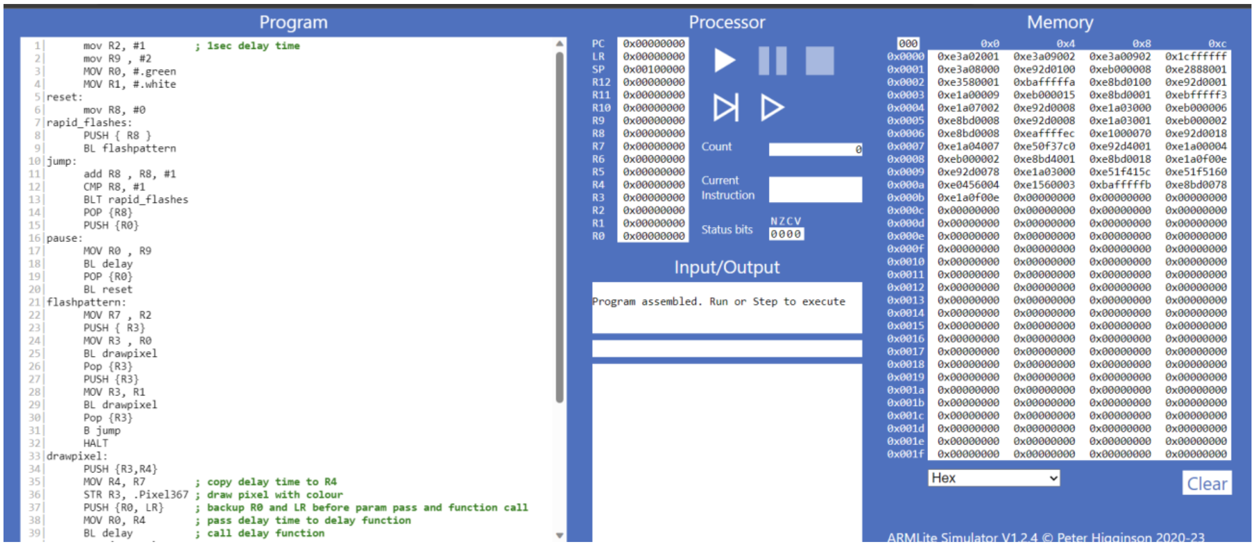
```
000           0x0         0x4         0x8         0xc
0x0000  0xe3a02001  0xe3a09002  0xe3a00902  0x1cffffff
0x0001  0xe3a08000  0xe92d0100  0xeb000008  0xe2888001
0x0002  0xe3580003  0xbafffffa  0xe8bd0100  0xe92d0001
0x0003  0xe1a00009  0xeb000015  0xe8bd0001  0xebfffff3
0x0004  0xe1a07002  0xe92d0008  0xe1a03000  0xeb000006
0x0005  0xe8bd0008  0xe92d0008  0xe1a03001  0xeb000002
0x0006  0xe8bd0008  0xeaffffec  0xe1000070  0xe92d0090
0x0007  0xe1a04007  0xe50f37c0  0xe92d4001  0xe1a00004
0x0008  0xeb000002  0xe8bd4001  0xe8bd0090  0xe1a0f00e
0x0009  0xe92d0078  0xe1a03000  0xe51f415c  0xe51f5160
0x000a  0xe0456004  0xe1560003  0xbafffffb  0xe8bd0078
0x000b  0xe1a0f00e  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
```



```
        Pop {R0, LR}        ; restore R0 and LR after function call
        Pop {R7,R4}
        RET
delay:                      ; job done
        push {R3,R4,R5,R6}
        MOV R3, R0          ; move delay time param into R3
        LDR R4, .Time       ; get start ime
timer:
        LDR R5, .Time       ; update time
        SUB R6, R5, R4      ; calc elapsed time
        CMP R6, R3          ; compare elapsed to delay time
        BLT timer
        pop {R3,R4,R5,R6}
        RET
```

```
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
```

Excercise 10.3



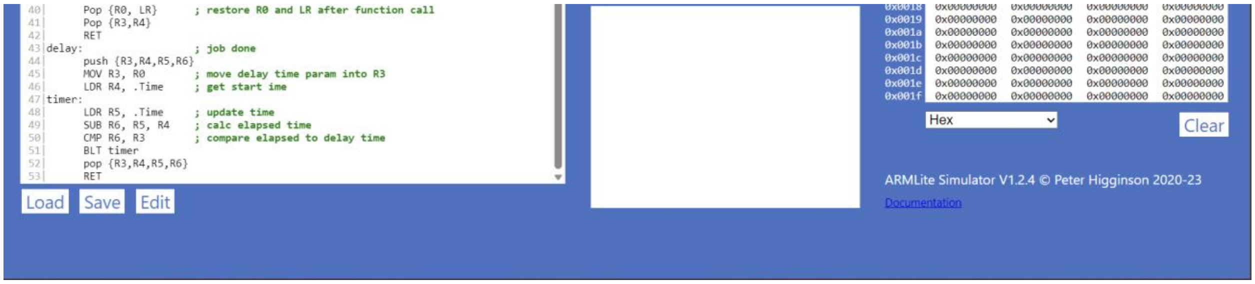
```
        mov R2, #1          ; 1sec delay time
        mov R9 , #2
        MOV R0, #.green
        MOV R1, #.white
reset:
        mov R8, #0
rapid_flashes:
        PUSH { R8 }
        BL flashpattern
jump:
        add R8 , R8, #1
        CMP R8, #1
        BLT rapid_flashes
        POP {R8}
        PUSH {R0}
pause:
        MOV R0 , R9
        BL delay
        POP {R0}
        BL reset
flashpattern:
        MOV R7 , R2
        PUSH { R3}
        MOV R3 , R0
        BL drawpixel
        Pop {R3}
        PUSH {R3}
        MOV R3, R1
        BL drawpixel
        Pop {R3}
        B jump
        HALT
drawpixel:
        PUSH {R3,R4}
        MOV R4, R7          ; copy delay time to R4
        STR R3, .Pixel367 ; draw pixel with colour
        PUSH {R0, LR}       ; backup R0 and LR before param pass and function call
        MOV R0, R4          ; pass delay time to delay function
        BL delay            ; call delay function
```

```
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
```

```
Program assembled. Run or Step to execute
```

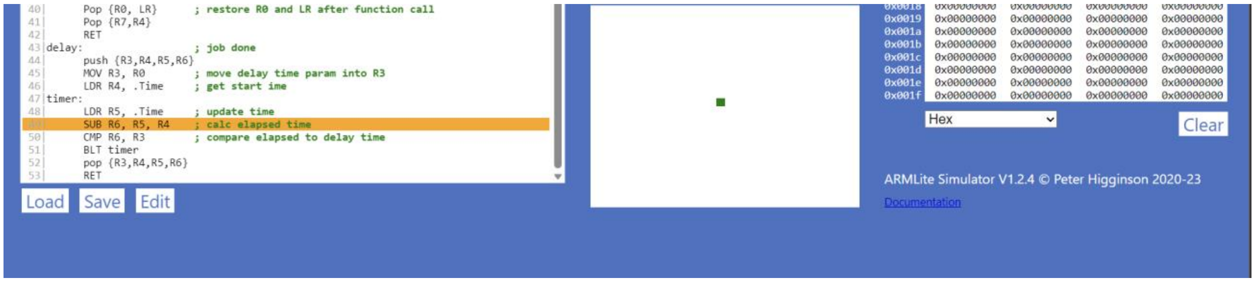
```
000           0x0         0x4         0x8         0xc
0x0000  0xe3a02001  0xe3a09002  0xe3a00902  0x1cffffff
0x0001  0xe3a08000  0xe92d0100  0xeb000008  0xe2888001
0x0002  0xe3580001  0xbafffffa  0xe8bd0100  0xe92d0001
0x0003  0xe1a00009  0xeb000015  0xe8bd0001  0xebfffff3
0x0004  0xe1a07002  0xe92d0008  0xe1a03000  0xeb000006
0x0005  0xe8bd0008  0xe92d0008  0xe1a03001  0xeb000002
0x0006  0xe8bd0008  0xeaffffec  0xe1000070  0xe92d0018
0x0007  0xe1a04007  0xe50f37c0  0xe92d4001  0xe1a00004
0x0008  0xeb000002  0xe8bd4001  0xe8bd0018  0xe1a0f00e
0x0009  0xe92d0078  0xe1a03000  0xe51f415c  0xe51f5160
0x000a  0xe0456004  0xe1560003  0xbafffffb  0xe8bd0078
0x000b  0xe1a0f00e  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
```



```
        Pop {R0, LR}        ; restore R0 and LR after function call
        Pop {R3,R4}
        RET
delay:                      ; job done
        push {R3,R4,R5,R6}
        MOV R3, R0          ; move delay time param into R3
        LDR R4, .Time       ; get start ime
timer:
        LDR R5, .Time       ; update time
        SUB R6, R5, R4      ; calc elapsed time
        CMP R6, R3          ; compare elapsed to delay time
        BLT timer
        pop {R3,R4,R5,R6}
        RET
```

```
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
```

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23



```
        Pop {R0, LR}        ; restore R0 and LR after function call
        Pop {R7,R4}
        RET
delay:                      ; job done
        push {R3,R4,R5,R6}
        MOV R3, R0          ; move delay time param into R3
        LDR R4, .Time       ; get start ime
timer:
        LDR R5, .Time       ; update time
        SUB R6, R5, R4      ; calc elapsed time
        CMP R6, R3          ; compare elapsed to delay time
        BLT timer
        pop {R3,R4,R5,R6}
        RET
```

```
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
```

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23