



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.

The screenshot displays a computer system simulator with three main panels: Program, Processor, and Memory.

Program Panel: Shows assembly code for a delay function and a main program.

```

1|main:
2|  mov r4, #1          ; no of delay seconds
3|  push {r0}           ; move r0 to stack
4|  mov r0, r4           ; move the no of second to delay to r0
5|  bl delay            ; invoke delay
6|  pop {r0}            ; return r0 from stack
7|  halt                ; end program
8|delay:
9|  push {r3, r4, r5, r6} ; function
10| mov r3, r0
11| ldr r4, .Time
12|loop:
13|  ldr r5, .Time
14|  sub r6, r5, r4
15|  cmp r6, r3
16|  blt loop
17|  pop {r3, r4, r5, r6}
18|  ret                ; mov pc, lr
  
```

Processor Panel: Shows the state of the processor registers and control units.

- Count:** 0
- Current Instruction:** (empty)
- Status bits:** NZCV 0000
- Input/Output:** Breakpoint removed at line 11 address 0x00020

Memory Panel: Shows a memory dump with addresses from 0x0000 to 0x001f. The dump shows various hexadecimal values, including 0x0000, 0x0001, 0x0002, 0x0003, 0x0004, 0x0005, 0x0006, 0x0007, 0x0008, 0x0009, 0x000a, 0x000b, 0x000c, 0x000d, 0x000e, 0x000f, 0x0010, 0x0011, 0x0012, 0x0013, 0x0014, 0x0015, 0x0016, 0x0017, 0x0018, 0x0019, 0x001a, 0x001b, 0x001c, 0x001d, 0x001e, and 0x001f.

(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.

Program

```

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

```

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: 130609493
 Current Instruction:
 Status bits: NZCV 0000

Input/Output

Memory

000	0x0	0x4	0x8	0xc
0x0000	0xe3a00902	0x1cfffff	0xe3a03003	0xe92d0003
0x0001	0xe1a00000	0xe1a01002	0xeb000010	0xe8bd0003
0x0002	0xe92d0003	0xe1a00001	0xe1a01002	0xeb00000b
0x0003	0xe8bd0003	0xeafffff4	0xe1000070	0xe92d0078
0x0004	0xe1a03000	0xe51f4108	0xe51f510c	0xe0456004
0x0005	0xe1560003	0xbafffff	0xe8bd0078	0xe1a0f00e
0x0006	0xe92d4000	0xe92d0001	0xe1a00001	0xebfffff2
0x0007	0xe8bd0001	0xe50f07c0	0xe8bd4000	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) 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

Program

```

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:
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

```

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: NZCV 0000

Input/Output

Program assembled. Run or Step to execute

Memory

000	0x0	0x4	0x8	0xc
0x0000	0xe3a00902	0x1cfffff	0xe3a03003	0xe92d0003
0x0001	0xe1a00000	0xe3a01001	0xeb000017	0xe8bd0003
0x0002	0xe92d0003	0xe1a00001	0xe3a01001	0xeb000012
0x0003	0xe8bd0003	0xe2433001	0xe3530000	0x1afffff2
0x0004	0xe92d0001	0xe3a00002	0xe3a00002	0xe8bd0001
0x0005	0xeaffffec	0xe1000070	0xe92d0078	0xe1a03000
0x0006	0xe51f4124	0xe51f5128	0xe0456004	0xe1560003
0x0007	0xbafffff	0xe8bd0078	0xe1a0f00e	0xe92d4000
0x0008	0xe92d0001	0xe1a00001	0xebfffff2	0xe8bd0001
0x0009	0xe50f07dc	0xe8bd4000	0xe1a0f00e	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

Exercise 10.3

Program

```
52|    bl delay
53|    pop {r0}
54|    str r0, .Pixel367
55|    pop {lr}
56|    ret
57|//*****//
58|flashpattern:    ; take r0 as no of rapid, r1 as delay duration
59|    push {lr}
60|    push {r3, r4, r5, r6}
61|    mov r6, r3    ; delay duration
62|    mov r5, r2    ; no of rapids
63|    mov r4, r1    ; white
64|    mov r3, r0    ; green
65|flash:
66|// turn green
67|    push {r0, r1}
68|    mov r0, r3    ; color value
69|    mov r1, #1    ; delay duration
70|    bl drawpixel
71|    pop {r0, r1}
72|// turn white
73|    push {r0, r1}
74|    mov r0, r4    ; color value
75|    mov r1, #1    ; delay duration
76|    bl drawpixel
77|    pop {r0, r1}
78|// rapid counter
79|    sub r5, r5, #1
80|    cmp r5, #0
81|    bne flash
82|    push {r0}
83|    mov r0, r6    ; delay duration
84|    bl delay
```

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

231228029

Current

Instruction

Status bits

NZCV
0000

Input/Output

Memory

000	0x0	0x4	0x8	0xc
0x0000	0xe3a00902	0x1cffffff	0xe3a02002	0xe3a03005
0x0001	0xe92d000f	0xe1a00000	0xe1a01001	0xe1a02002
0x0002	0xe1a03003	0xeb000013	0xe8bd000f	0xeaffffff7
0x0003	0xe1000070	0xe92d0078	0xe1a03000	0xe51f4100
0x0004	0xe51f5104	0xe0456004	0xe1560003	0xbaffffffb
0x0005	0xe8bd0078	0xe1a0f00e	0xe92d4000	0xe92d0001
0x0006	0xe1a00001	0xebffffff2	0xe8bd0001	0xe50f07b8
0x0007	0xe8bd4000	0xe1a0f00e	0xe92d4000	0xe92d0078
0x0008	0xe1a06003	0xe1a05002	0xe1a04001	0xe1a03000
0x0009	0xe92d0003	0xe1a00003	0xe3a01001	0xebffffffd
0x000a	0xe8bd0003	0xe92d0003	0xe1a00004	0xe3a01001
0x000b	0xebffffffe8	0xe8bd0003	0xe2455001	0xe3550000
0x000c	0x1affffff2	0xe92d0001	0xe1a00006	0xebffffffd8
0x000d	0xe8bd0001	0xe8bd0078	0xe8bd4000	0xe1a0f00e
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