SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10004 – Computer System

Lab 09

Student Name: Dang Vi Luan

Student ID: 103802759

Question 4:

4.1

Kernel7.asm:

Setting up initial value for the program including the value for recursion, room for stack. It is the equivalent of "main function" in other high-level language I would say. It also calls other function in the program.

Factorialj.asm:

It takes an N value from Kernel7.asm and calculate the factorial value of N. The algorithm, however, has flaw as it would not work for N values below 2. I will try to fix this error as another leveling up practice.

TIMER.asm:

A dumb timer that uses all the CPU power for subtracting numbers, it acts as the initial timer for our program that will be replaced by DELAY.asm

GPIO.asm:

This code contains two functions, one for setting up the LED by using the normal process we did in previous labs.

The other is for flashing the LED by turning it on and off. After each tick the LED will pause for 0.5 seconds.

These two functions will be called by Kernel7.asm

DELAY.asm

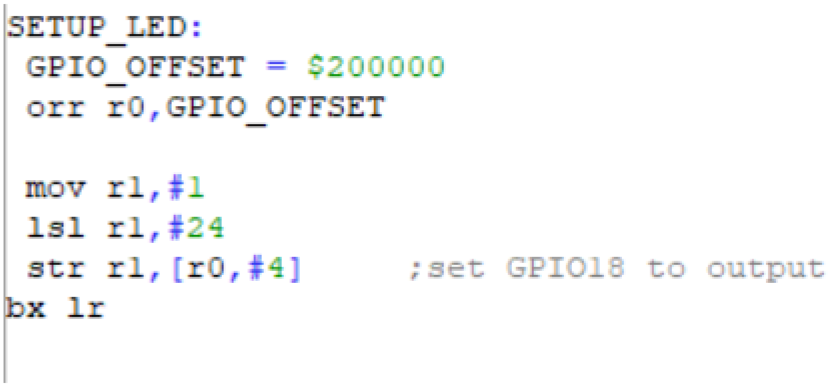
The better timer that we have done in Lab08, which will later be called in Flash function

4.2

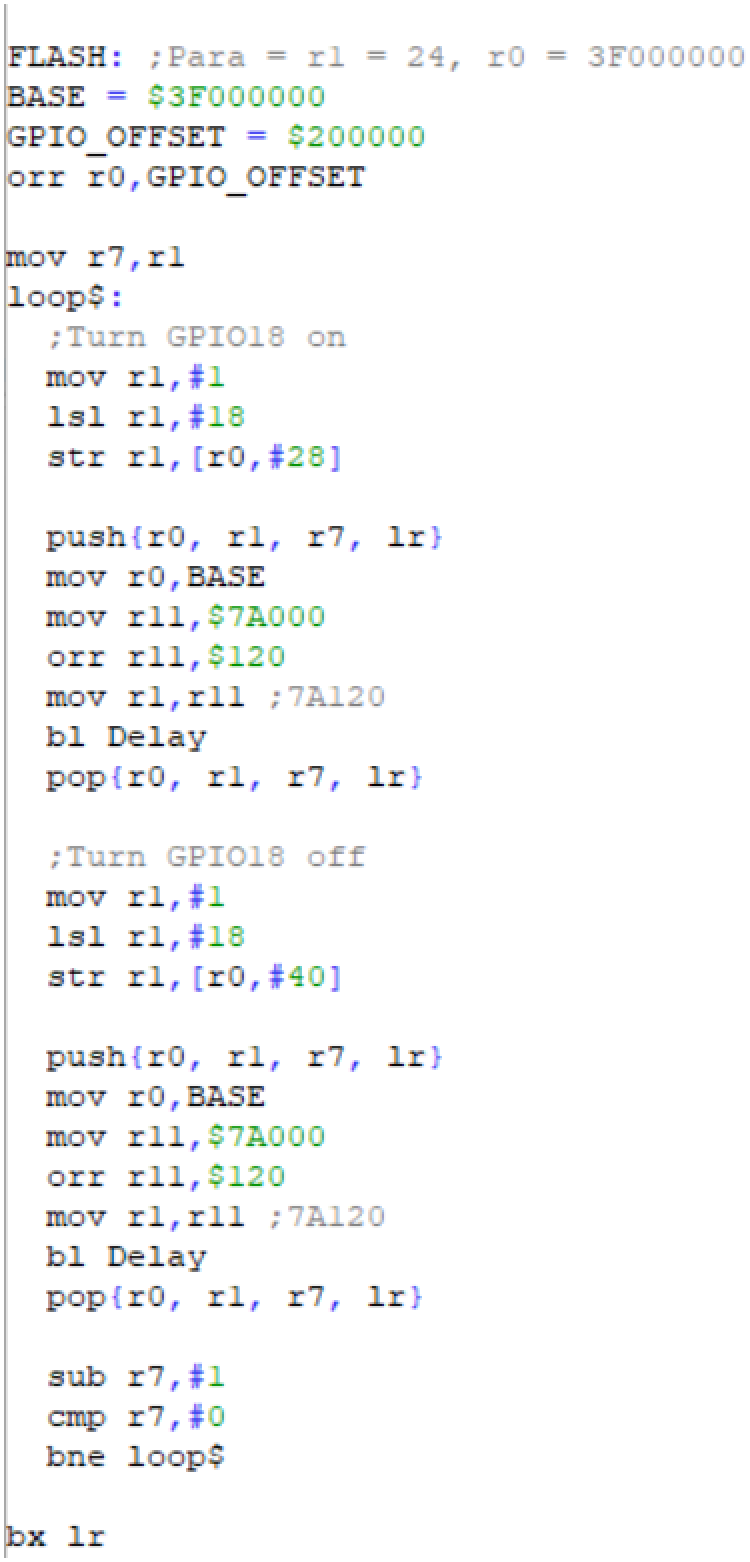
The register that holds the input to the program is register 1, which holds the input of factorial function and later the output of that function will determine the number of flashes our program will produce. In this particular case the input is 4.

Question 5

Separate GPIO function for initializing LED and flashing LED and move them to their own file:

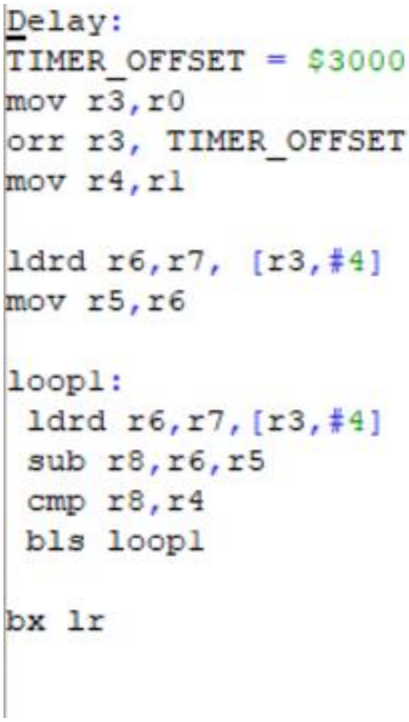


```
SETUP_LED:
 GPIO_OFFSET = $200000
 orr r0,GPIO_OFFSET

 mov r1,#1
 lsl r1,#24
 str r1,[r0,#4]      ;set GPIO18 to output
bx lr
```



```
FLASH: ;Para = r1 = 24, r0 = 3F000000
BASE = $3F000000
GPIO_OFFSET = $200000
orr r0,GPIO_OFFSET

mov r7,r1
loop$:
  ;Turn GPIO18 on
  mov r1,#1
  lsl r1,#18
  str r1,[r0,#28]

  push{r0, r1, r7, lr}
  mov r0,BASE
  mov r11,$7A000
  orr r11,$120
  mov r1,r11 ;7A120
  bl Delay
  pop{r0, r1, r7, lr}

  ;Turn GPIO18 off
  mov r1,#1
  lsl r1,#18
  str r1,[r0,#40]

  push{r0, r1, r7, lr}
  mov r0,BASE
  mov r11,$7A000
  orr r11,$120
  mov r1,r11 ;7A120
  bl Delay
  pop{r0, r1, r7, lr}

  sub r7,#1
  cmp r7,#0
  bne loop$

bx lr
```

A better timer to replace TIMER.asm:



```
Delay:
TIMER_OFFSET = $3000
mov r3,r0
orr r3, TIMER_OFFSET
mov r4,r1

ldrd r6,r7, [r3,#4]
mov r5,r6

loop1:
 ldrd r6,r7,[r3,#4]
 sub r8,r6,r5
 cmp r8,r4
 bls loop1

bx lr
```