

LAB 9

Name: DhruPAD Thorat

Student id: 102634728

Provide a brief description of the Code:

- Kernel7.asm is the code which we will use for implementing the FLASH and SETUP_LED functions.
- Timer.asm code will act as the system's dumb time. A wait has been implemented in the dumb timer.
- Factorial.asm is the code for flashing the LED by using the input provided earlier (i.e. 4). Therefore, the code will use the factorial of the input which will be 24. Also, there is a function labelled as FACTORIAL. When r1 is equal to 1 and then we pop off the value of the stack and put it in r1 and then we branch back to link register(lr).
- GPIO.asm consists of the code which is used for accessing the LED which is controlled by a GPIO register. Here, in the Flash function the value is backed up in link register and is pushed on to the stack before TIMER function is called and it is then popped off from the stack after TIMER was executed. This was done because the LED was flashing more than 24 times.
- Register r1 holds the input for the program which is 4 that will be used by our system which is then moved to the stack.

Kernel7.ASM code:

```
;Calculate
mov r1,#4 ;input
mov sp,$1000 ;make room on the stack
mov r0,r1
bl FACTORIAL
mov r7,r0 ;store answer
BASE = $3F000000 ;RP2 and RP3 ;GPIO_SETUP

mov r0,BASE
bl SETUP_LED

push {r0,r1}
mov r0,BASE
mov r1,r7
bl FLASH
pop {r0,r1}

wait:
b wait
include "TIMER.asm"
include "factorialj.asm"
include "GPIO.asm"
```

GPIO.ASM code:

```
SETUP_LED:
GPIO_OFFSET = $200000
orr r0,GPIO_OFFSET
mov r1,#1
lsl r1,#24
str r1,[r0,#4]
bx lr

FLASH:
orr r0, GPIO_OFFSET
mov r7,r1
loop$:
    mov r1,#1
    lsl r1,#18
    str r1,[r0,#28] ;LED on
    mov r2,$0F0000
    push {lr}
    bl TIMER
    pop {lr}

    mov r1,#1
    lsl r1,#18
    str r1,[r0,#40] ;LED off
    mov r2,$0F0000
    push {lr}
    bl TIMER
    pop {lr}

sub r7,r7,#1
cmp r7,#0
bne loop$
bx lr
```