COS10004 – Computer Systems

Name: Nguyen Manh Dung

Student ID: 104181789 (SWH01226)

LAB 8

Insert the error message into your submission document

7. Convert your student number to Hex

#104181= $196F5or $196155

8.1. Why does MOV only work with numbers with 24 bits set to 0?

Because there are 4 ROR bits left after accounting for the 20 op-code bits in the 24 bits total. The barrel shifter uses these 24 bits. Only 8 bits hold the necessary value to transfer.

8.2. How can MOV still be used for numbers that do not satisfy this ?

We have 64bit and 84bit mov instructions which can use more bits to store the

number value to move.

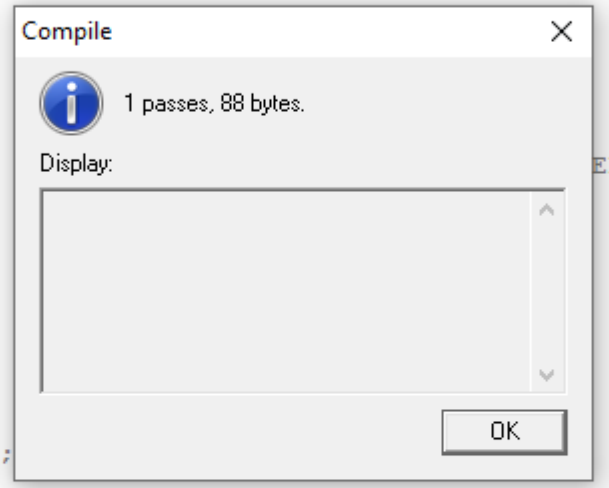
8.3. Identify the three bytes (as hex digits) needed to construct your student number, and write the code to load the entire number into a register.

I use $10000, $9600 and $F5 to construct my student number

flat assembler for ARM version 1.43 (built on fasm 1.73.02)

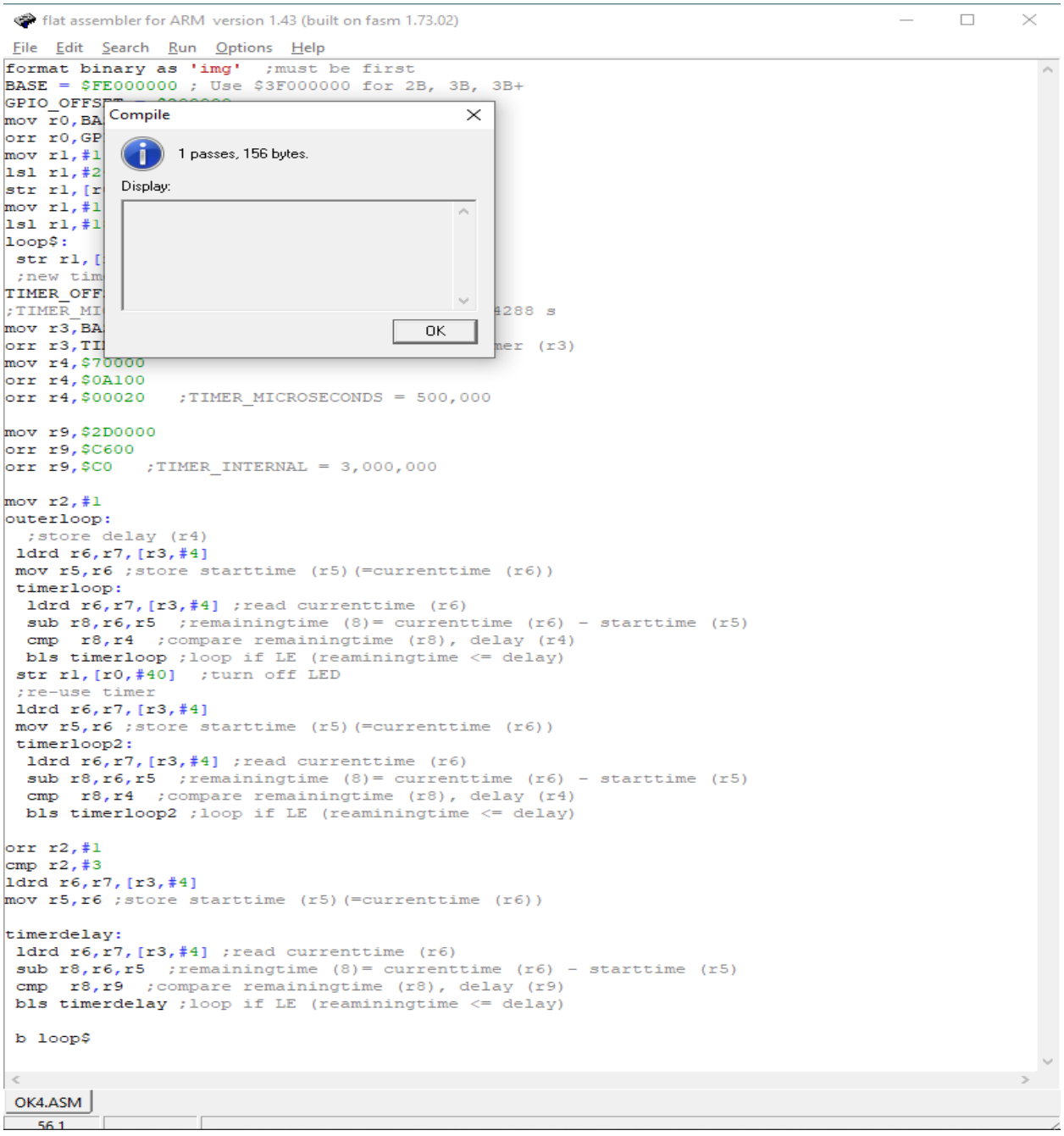
File Edit Search Run Options Help

```
format binary as 'kernel7.img'

BASE = $FE000000  ; use $3F000000 for RP2B, 3B, 3B+
GPIO_OFFSET = $200000
mov r0,BASE
orr r0,GPIO_OFFSET
;start of GPIO
mov r1,#1
lsl r1,#24
str r1,[r0,#4]

loop$:                                          ED off, wait
  mov r1,#1
  lsl r1,#18
  str r1,[r0,#28]

   mov r2,$0F0000
  wait1$:
    sub r2,#1
    cmp r2,#0
    bne wait1$     ;

  mov r1,#1  ;can be omitted
  lsl r1,#18 ;can be omitted
  str r1,[r0,#40]   ;turn LED off (writing to the pull up register)

  mov r2,$10000
  orr r2,$9600
  orr r2,$F5
  wait2$:
    sub r2,#1
    cmp r2,#0
    bne wait2$    ;count from 983040 to 0 (busy wait)
b loop$  ;end of outer loop
```



15. We are going to modify this code so that it implements 3 flashes followed by a pause. This will require 2 extra loops. Think about how you would do this, using comments or hand written notes to plan what modifications.

1. Add to r9 the value TIMER INTERNAL of 3 seconds. This causes the LED to turn off for three seconds when using timerdelay.
2. To create the outer loop that enables the LED to flash three times, use r2 (=#1) and BLS. r2 is increased by 1 after a loop until it reaches 4.
3. Make a timerdelay that pauses the LED for 3 seconds.




```
format binary as 'img'   ;must be first
BASE = $FE000000 ; Use $3F000000 for 2B, 3B, 3B+
GPIO_OFFSET = $200000
mov r0,BA
orr r0,GP
mov r1,#1
lsl r1,#2
str r1,[r
mov r1,#1
lsl r1,#1
loop$:
 str r1,[
 ;new tim
TIMER_OFF
;TIMER_MI                                    4288 s
mov r3,BA
orr r3,TI                                   mer (r3)
mov r4,$70000
orr r4,$0A100
orr r4,$00020    ;TIMER_MICROSECONDS = 500,000

mov r9,$2D0000
orr r9,$C600
orr r9,$C0    ;TIMER_INTERNAL = 3,000,000

mov r2,#1
outerloop:
  ;store delay (r4)
 ldrd r6,r7,[r3,#4]
 mov r5,r6 ;store starttime (r5)(=currenttime (r6))
 timerloop:
  ldrd r6,r7,[r3,#4] ;read currenttime (r6)
  sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
  cmp  r8,r4  ;compare remainingtime (r8), delay (r4)
  bls timerloop ;loop if LE (reamainingtime <= delay)
 str r1,[r0,#40]  ;turn off LED
 ;re-use timer
 ldrd r6,r7,[r3,#4]
 mov r5,r6 ;store starttime (r5)(=currenttime (r6))
 timerloop2:
  ldrd r6,r7,[r3,#4] ;read currenttime (r6)
  sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
  cmp  r8,r4  ;compare remainingtime (r8), delay (r4)
  bls timerloop2 ;loop if LE (reamainingtime <= delay)

orr r2,#1
cmp r2,#3
ldrd r6,r7,[r3,#4]
mov r5,r6 ;store starttime (r5)(=currenttime (r6))

timerdelay:
 ldrd r6,r7,[r3,#4] ;read currenttime (r6)
 sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
 cmp  r8,r9  ;compare remainingtime (r8), delay (r9)
 bls timerdelay ;loop if LE (reamainingtime <= delay)

 b loop$
```

My Code:

```
 format binary as 'img'  ;must be first
BASE = $FE000000 ; Use $3F000000 for 2B, 3B, 3B+
GPIO_OFFSET = $200000
mov r0,BASE
orr r0,GPIO_OFFSET ;Base address of GPIO
mov r1,#1
lsl r1,#24; GPIO18
str r1,[r0,#4] ;enable output
mov r1,#1
lsl r1,#18
loop$:
 str r1,[r0,#28] ;Turn on LED
 ;new timer
TIMER_OFFSET = $3000
;TIMER_MICROSECONDS = 524288 ; $0080000 ;0.524288 s
mov r3,BASE
orr r3,TIMER_OFFSET ;store base address of timer (r3)
mov r4,$70000
orr r4,$0A100
orr r4,$00020   ;TIMER_MICROSECONDS = 500,000

mov r9,$2D0000
orr r9,$C600
orr r9,$C0   ;TIMER_INTERNAL = 3,000,000

mov r2,#1
outerloop:
  ;store delay (r4)
 ldrd r6,r7,[r3,#4]
```

```
 mov r5,r6 ;store starttime (r5)(=currenttime (r6))
 timerloop:
  ldrd r6,r7,[r3,#4] ;read currenttime (r6)
  sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
  cmp  r8,r4  ;compare remainingtime (r8), delay (r4)
  bls timerloop ;loop if LE (reamainingtime <= delay)
 str r1,[r0,#40]  ;turn off LED
 ;re-use timer
 ldrd r6,r7,[r3,#4]
 mov r5,r6 ;store starttime (r5)(=currenttime (r6))
 timerloop2:
  ldrd r6,r7,[r3,#4] ;read currenttime (r6)
  sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
  cmp  r8,r4  ;compare remainingtime (r8), delay (r4)
  bls timerloop2 ;loop if LE (reamainingtime <= delay)

orr r2,#1
cmp r2,#3
ldrd r6,r7,[r3,#4]
mov r5,r6 ;store starttime (r5)(=currenttime (r6))

timerdelay:
 ldrd r6,r7,[r3,#4] ;read currenttime (r6)
 sub r8,r6,r5  ;remainingtime (8)= currenttime (r6) - starttime (r5)
 cmp  r8,r9  ;compare remainingtime (r8), delay (r9)
 bls timerdelay ;loop if LE (reamainingtime <= delay)

 b loop$
```