Lab 7

Thanh Minh

103809048

16.

16.1. Establish the base address of the GPIO registers

BASE=$3F000000

GPIO_OFFSET=$200000

mov r0,BASE

orr r0,GPIO_OFFSET

16.2.Program GPIO18 for writing

Mov r1,#1

Lsl r1, #24

Str r1,[r0,#4]

16.3. Set GPIO18 to ON?

Mov r1, #1

Lsl r1,#24

Str r1,[r0,#4]

16.4. Stop the instruction pointer (program counter) from continuing beyond the executable program code?

loop$:

b loop$

20.

20.1. What number bit is set (within the associated 32 bit block) to enable GPIO23 for writing?

#9

20.2. What is the byte offset from GPIO_BASE that this 32 bit block must be written to in memory?

#8

20.3. What number bit is set to set GPIO23 to ON (again within the 32 bit block associated with that GPIO pin)?

#28

20.4. What is the byte offset from GPIO_BASE that this 32 bit block must be written to memory?

200000

22.

22.1. Which exact snippet of code will need to change compared to turning the LED on ?

```
mov r1, #1

lsl r1, #23

str r1,[r0,#28]
```

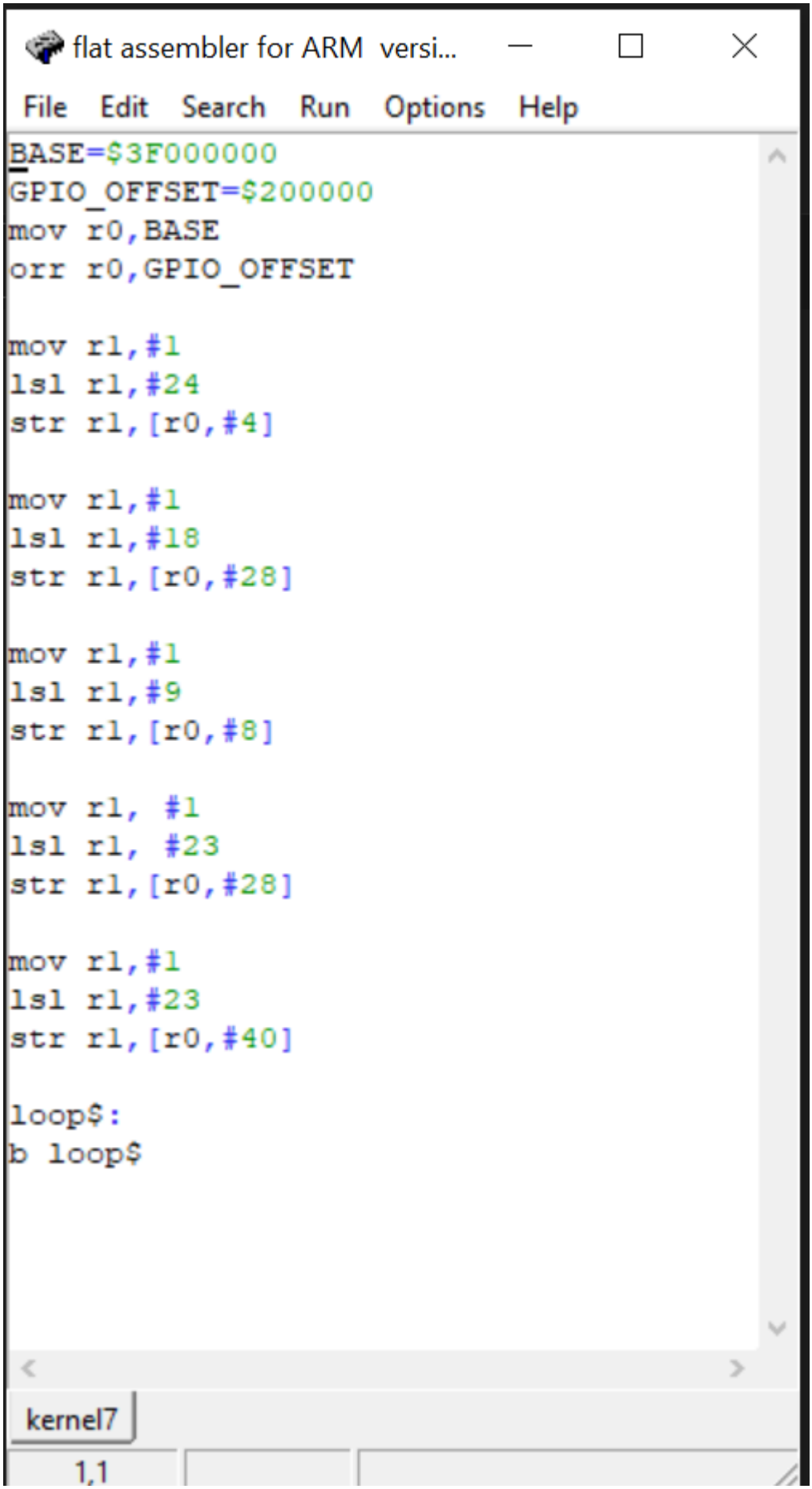
22.2. Provide the alternative code to turn the LED off (again you will need to refer to the GPIO register diagram). No need to demonstrate this working. We'll deal with flashing LEDs next week.

```
mov r1,#1

lsl r1,#23

str r1,[r0,#40]
```

My result


flat assembler for ARM versi...
File Edit Search Run Options Help
BASE=$3F000000
GPIO_OFFSET=$200000
mov r0,BASE
orr r0,GPIO_OFFSET
mov r1,#1
lsl r1,#24
str r1,[r0,#4]
mov r1,#1
lsl r1,#18
str r1,[r0,#28]
mov r1,#1
lsl r1,#9
str r1,[r0,#8]
mov r1, #1
lsl r1, #23
str r1,[r0,#28]
mov r1,#1
lsl r1,#23
str r1,[r0,#40]
loop$:
b loop$
kernel7
1,1