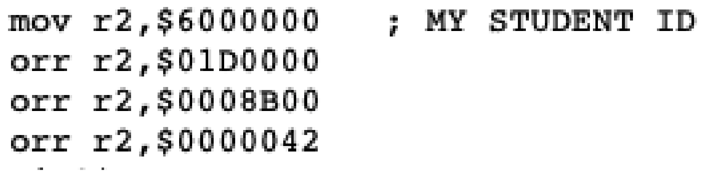
Aaryan Pujara 102599490

1. The mov works really fast, to do so it needs to be very efficient. mov works extraordinarily fast by sending all the information as 32 bits. That’s why mov only accepts some number (those with at least 24 bits set to 0).

2. use of ORR command to convert the value into executable number can work.

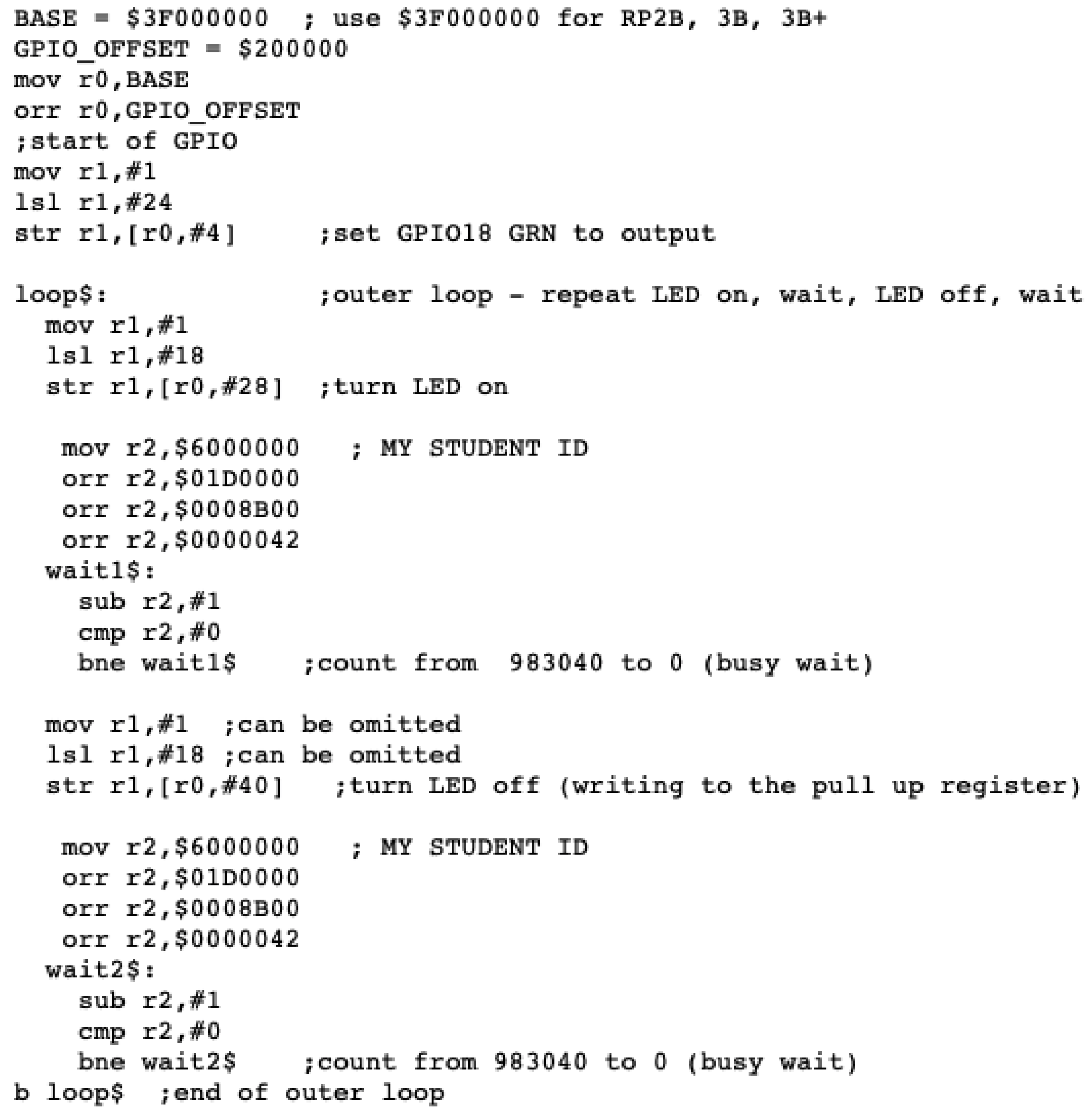


```
mov r2,$6000000    ; MY STUDENT ID
orr r2,$01D0000
orr r2,$0008B00
orr r2,$0000042
```

3.

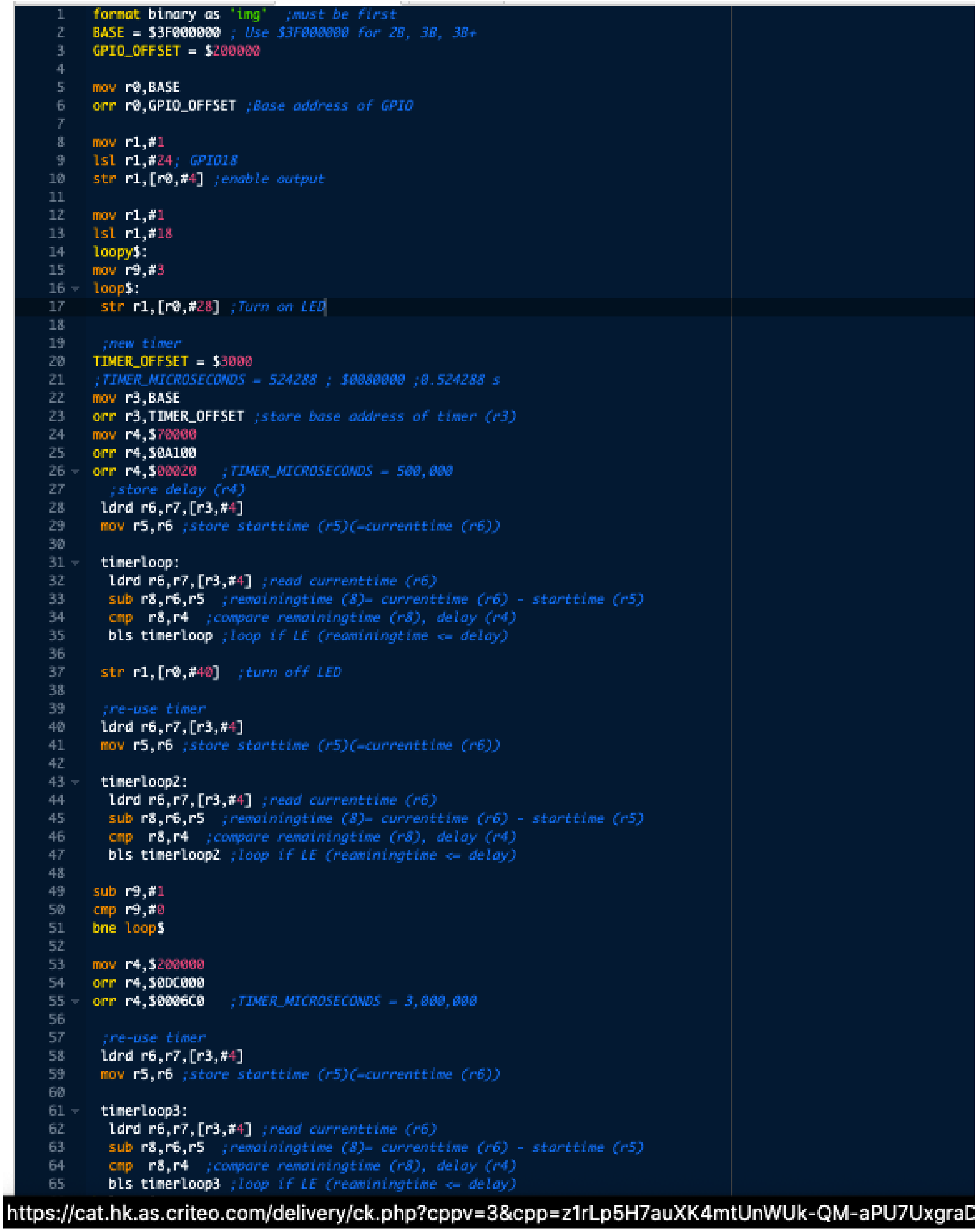
Aaryan Pujara 102599490

OK2



```
BASE = $3F000000  ; 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]     ;set GPIO18 GRN to output

loop$:             ;outer loop - repeat LED on, wait, LED off, wait
  mov r1,#1
  lsl r1,#18
  str r1,[r0,#28]  ;turn LED on

   mov r2,$6000000   ; MY STUDENT ID
   orr r2,$01D0000
   orr r2,$0008B00
   orr r2,$0000042
  wait1$:
    sub r2,#1
    cmp r2,#0
    bne wait1$    ;count from  983040 to 0 (busy wait)

  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,$6000000   ; MY STUDENT ID
   orr r2,$01D0000
   orr r2,$0008B00
   orr r2,$0000042
  wait2$:
    sub r2,#1
    cmp r2,#0
    bne wait2$    ;count from 983040 to 0 (busy wait)
b loop$  ;end of outer loop
```

OK4



```
format binary as 'img'  ;must be first
BASE = $3F000000 ; 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
loopy$:
mov r9,#3
loop$:
 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
  ;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 (reaminingtime <= 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 (reaminingtime <= delay)

sub r9,#1
cmp r9,#0
bne loop$

mov r4,$200000
orr r4,$0DC000
orr r4,$0006C0   ;TIMER_MICROSECONDS = 3,000,000

 ;re-use timer
 ldrd r6,r7,[r3,#4]
 mov r5,r6 ;store starttime (r5)(=currenttime (r6))

 timerloop3:
  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 timerloop3 ;loop if LE (reaminingtime <= delay)
```

https://cat.hk.as.criteo.com/delivery/ck.php?cppv=3&cpp=z1rLp5H7auXK4mtUnWUk-QM-aPU7UxgraD