Project03 grades → need project03 for Project04

Lab07

cmp

b / bl

beq / bne / blt / ...

lsl , lsr

mvn

---

mul

E0020092 mul r2, r2, r0

rd rm rs

1110	0000 0000	0010	0000	0000	1001	0010
cond	mul bits AS 00	rd	rn	rs	mul bits	rm
E		2	0	0		2
AL						

---

ldr r2, [r0] == ldr r2, [r0, #0]

mem:

mov r0, r1

add r0, r1, r2

mov r2, #1	add r0, r1, r2
dp1;	dp3
mov r2, r3	add r0, r1, #1
dp2	dp2;

cmp r0, r1

dp_cmp_2 ←

dpword2

struct {int rn; int rm;} dpcmp2;

---

lsl → multply

lsr → divide by 2

(rd) (rn) (?) ↓

→ lsl r4, r2, #2 ⟶

(rm)

ldr r12, [r0, r4]

mvn

mov r2, #-2

↓

mvn r2, #1 ← 0000 0001

31

1111 . . . 11111 1110 -2

Branch

b find_max_loop

EAFF FFF7

word offset / inst offset: −9

1110 | 1010 | 1111 1111 1111 1111 1111 0111

cond (E = AL) | b, L

0000 0000 0000 0000 0000 1000
+1
0000 0000 0000 0000 0000 1001 → 9

24:
1111 1111 1111 1111 1111 0111

<< 2 → 26:

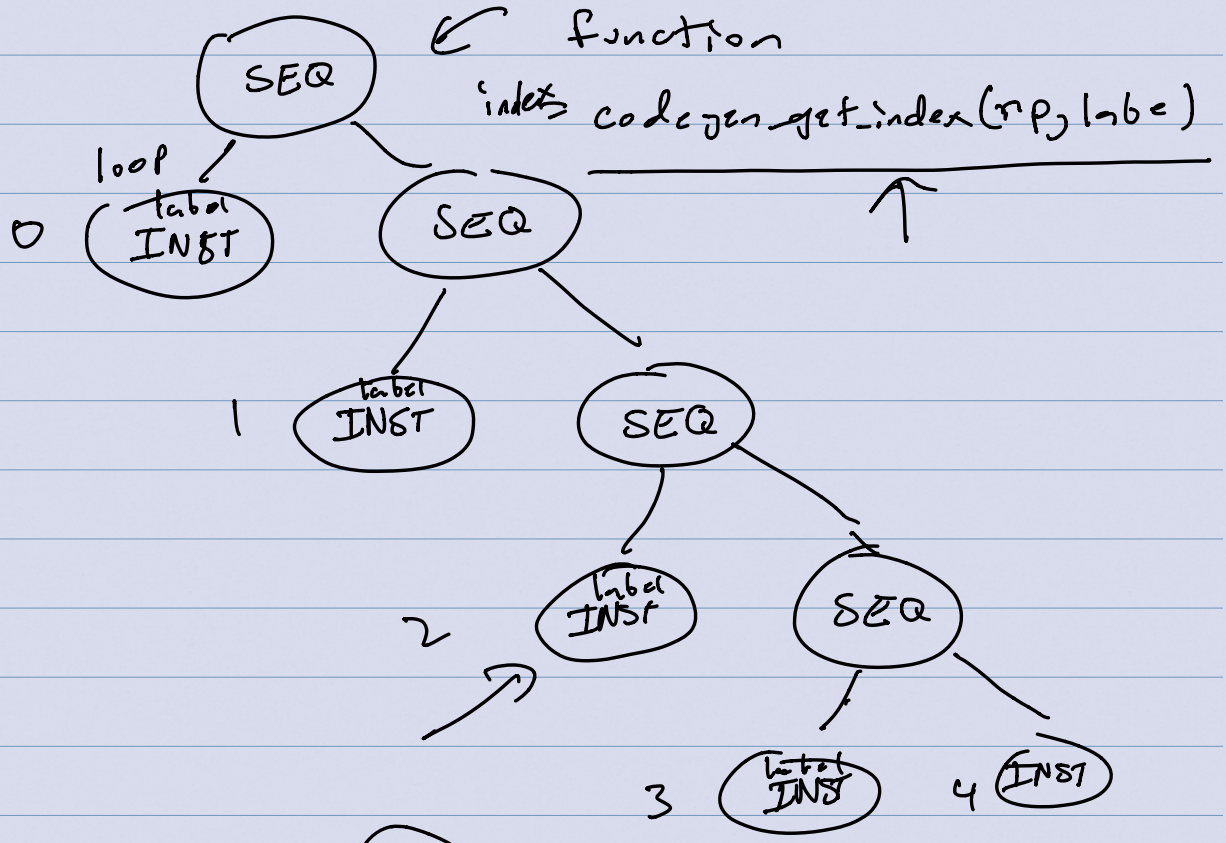
01 1111 1111 1111 1111 1101 1100

sign extend

1111 1111 1111 1111 : 1111 1111 1101 1100

PC+8 −36

(i) → b label (j) ←



(i) b loop

(j)

offset # [i - j]

2's comp

24 bits

PC + (2)

↑ word offset.

.global find_max_s

↑

ELF