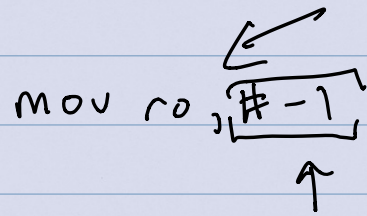
Project04

Branching - offset, conditional

Remaining Formats - lsl, lsr, cmp, mov, mvn

ELF lib + example (tomorrow)



1111 1111 (-1)

0000 0000

+1

0000 0001 (1)

01

0000 0001

1111 --- 1111 1110 (-2)

mov r0, #-1

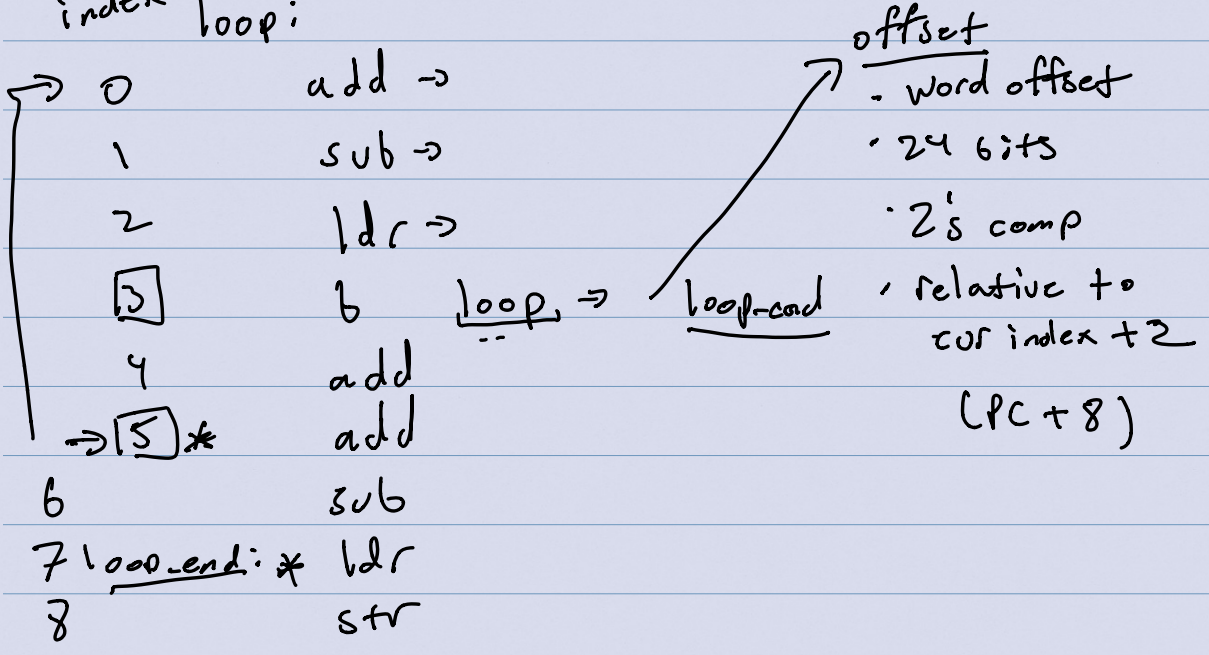
#-2

#-3

#-16

mvn

Branching



codegen for b loop

int index = 3

int target_index = 0

offset = target_index - (index + 2)

0 - (3 + 2)

0 - 5

= -5

int index = 3

int target_index = 7

offset = 7 - (3+2)

= 7 - 5

= 2

Foo: add r0, r0, r1

0 add

1 sub

2 b foo ←

ct->len = 2

---

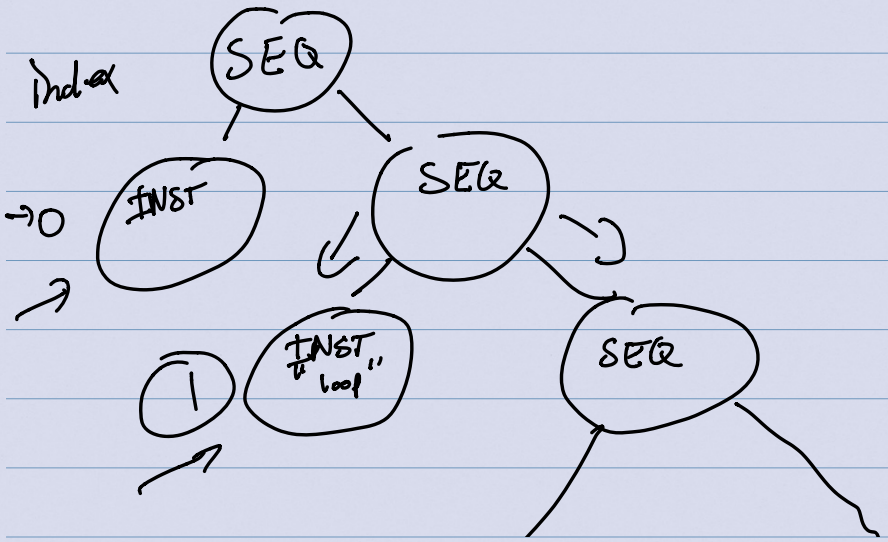
codegen_b (ct, np) {

int index = ct->len; ?

parse_tree

int target_index = codegen_get_index(, label, 0)

cond




```
int codegen_get_index(struct parse_node_st *np, char *label, int level){
    int l1 = -1;
    int l2 = -1;

    if (np->type == INST) {
        if (strncmp(label, np->stmt.inst.label, SCANTOKEN_LEN) == 0) {
            l1 = level;
        }
    } else if (np->type == SEQ) {
        l1 = codegen_get_index(np->stmt.seq.left, label, level);
        l2 = codegen_get_index(np->stmt.seq.right, label, level+1);

        if (l1 == -1) {
            l1 = l2;
        }
    }

    return l1;
}
```

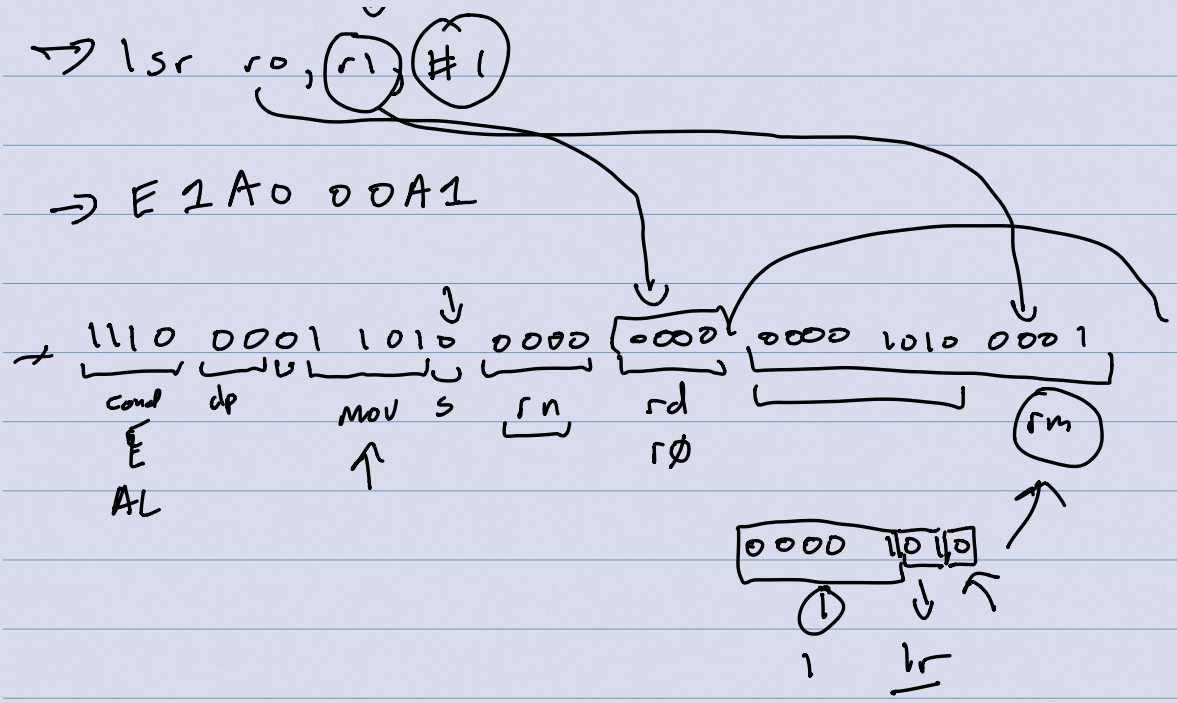
---

conditional execution

beq

blt

moveq


lsr r0, r1, #1
E1A0 00A1
1110 0001 1010 0000 0000 0000 1010 0001
cond dp MOV S rn rd rm
E
AL
r0
0000 11 0 11 0
1
lr