

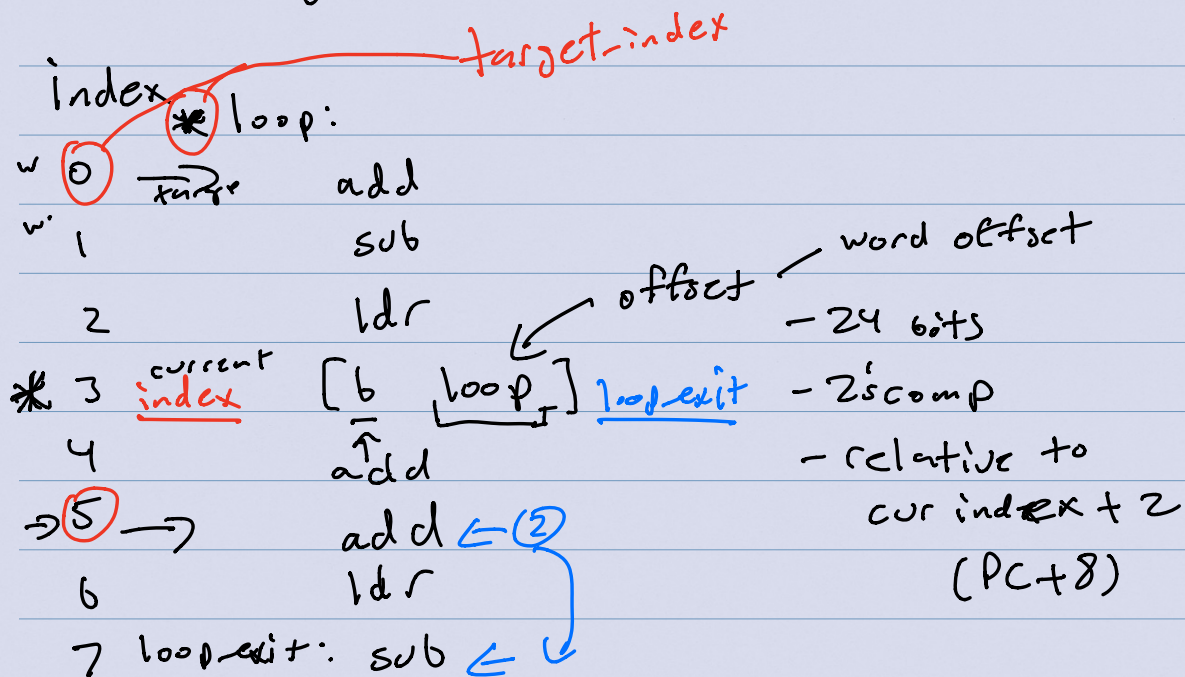
Project04

Branching - conditional branching, offset

Remaining formats - lsl, lsr, cmp, mov, mun

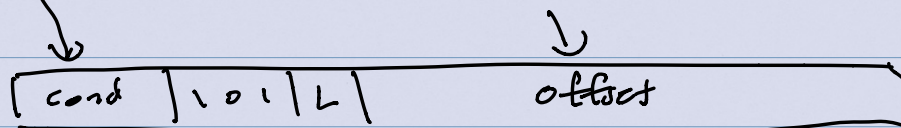
ELF (t.morrow) lib + example

Branching



codegen_b (ct, np) Σ
np → index

code name → get code
np → stmt.inst.b. label



$\text{int target_index} = \text{codegen_get_index}(np, \text{label})$
 $\text{int index} = \text{ct} \rightarrow \text{len}$

int offset

offset = -5

offset = cu

*

offset = target_index - (index + 2)

0 - (3 + 2)

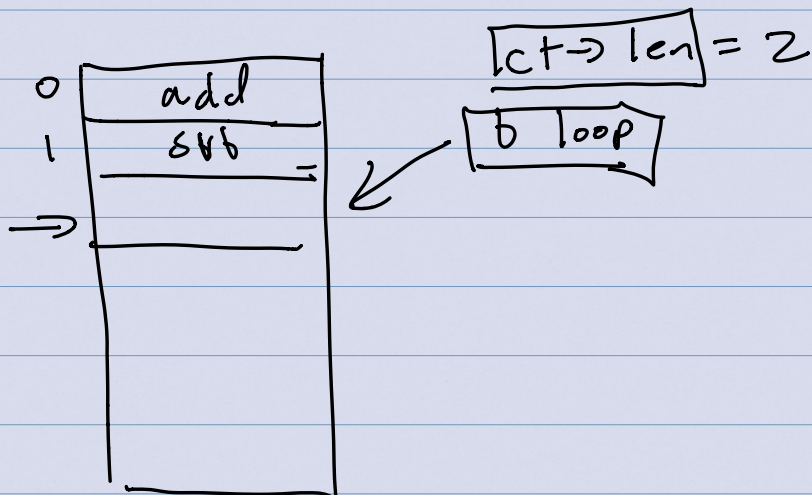
0 - 5

= -5

offset = 7 - (3 + 2)

= 7 - 5

= 2



```
int codegen_get_index(struct pre_node *np,
                    char *label, int level) {
```

```
    int i1 = -1;
    int i2 = -1;
```

ignore
- global

```
    if (np->type == INST) {
```

```
        if (strcmp(label, np->stmt.inst.label, SCAN_TOKEN_LEN) == 0) {
```

```
            i1 = level;
```

```
        } else if (np->type == SEQ) {
```

```
            i1 = codegen_get_index(np->stmt.seq.left, label, level);
```

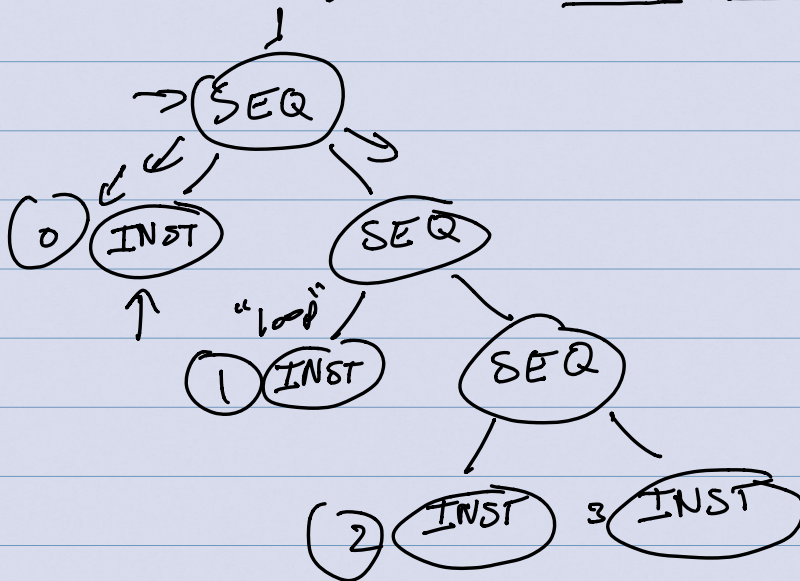
```
            i2 = codegen_get_index(np->stmt.seq.right, label, level + 1);
```

```
            if (i1 == -1) {
```

```
                i1 = i2;
```

```
            }
        }
    }
    return i1;
```

```
target_index = codegen_get_index(ct->tree, label, 0);
```



main
→ global sort-s → ELF

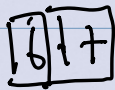
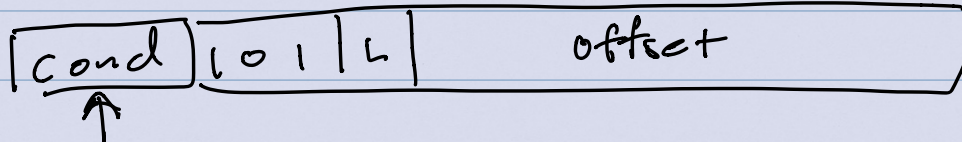
r0 add

r1 add

global Foo ← ignore

2 sb

↑



name

code.lookup(name, table)

rmem;

ldr r0, [r1, #4]

ldr r0, [r1]

⇒ ldr r0, [r1, #0]

procd

↓
memi

same machine code