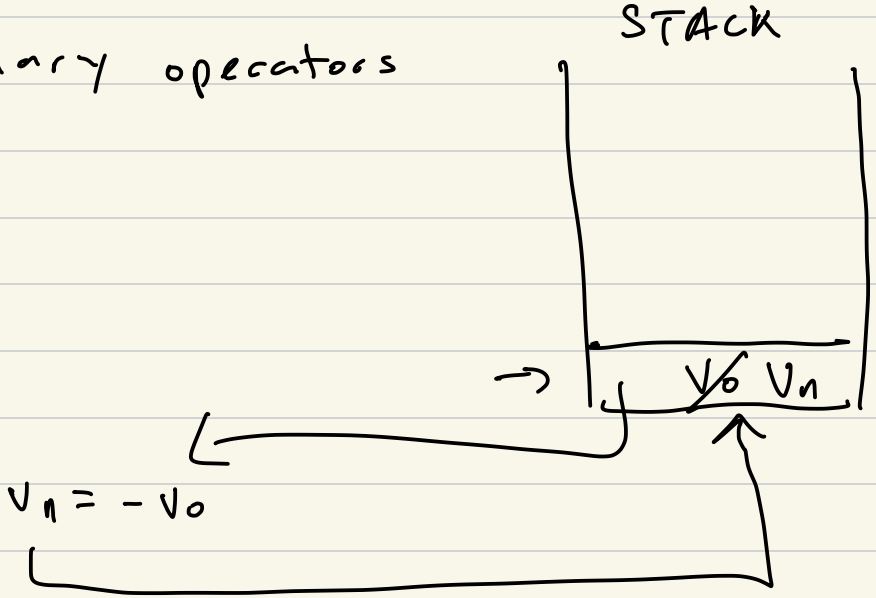
CS631-01 RISC-V Code Gen & Machine Code

Unary operators



./ntlang -e "a0+a1+3" -c Foo > foo.s

① └→ PREAMBLE (main in assembly)

codegen_main.c

└→ codegen_main.s

write ↑ to stdout

② CODEGEN for expr

compile_expr (tree, config)

└→ emit assembly code

PREAMBLE debugging

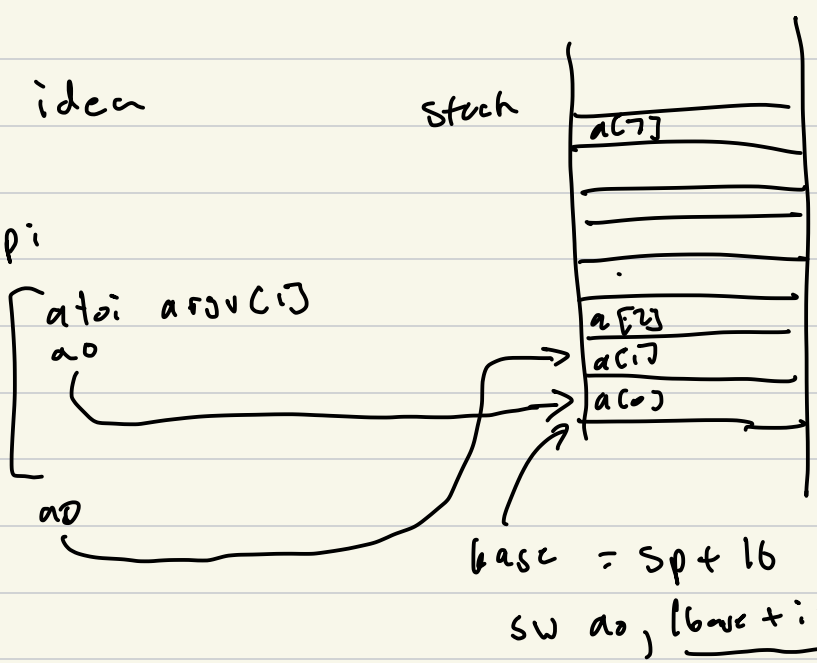
```
foo:
    add a0,a0,a1
    ret
```

args

1). /rtlang -e "1+2" -c codegen_func_s > codegen_func_s.s

2) gcc -o codegen codegen_main.c codegen_func_s.s

Basic idea



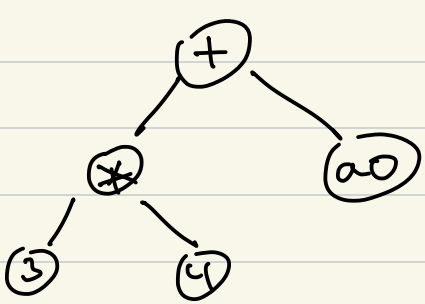
```
mv  t0, base
lw  a0, 0(t0)
lw  a1, 4(t1)
    .
    .
    .
call codegen_func_s
```

```
# a0 - int argc
# a1 - char *argv[]

main:
```

Constant folding

$(3 * 4) + a0$



Machine code

Binary form of Assembly

00B50533

add a0, a0, a1