

Section 6:

Code Generation

CS 164 @ UC Berkeley, Spring 2024

Reminders

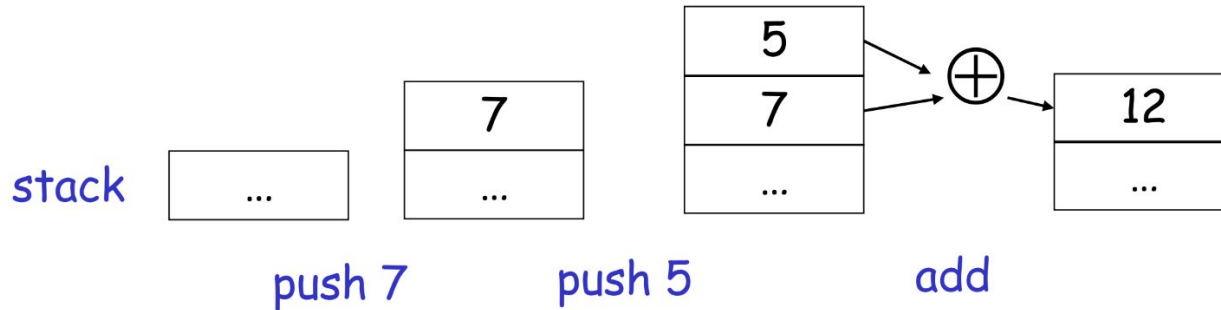
PA2 is released, and is due on Monday, March 18 at 11:59 PST.

Additional OH on course calendar.

WA 4 is due on Thursday, March 21 at 11:59 PST.

Reminder to take care of yourselves, and to prioritize your health! WAs are worth 5% of your grade so don't stress too much about them!

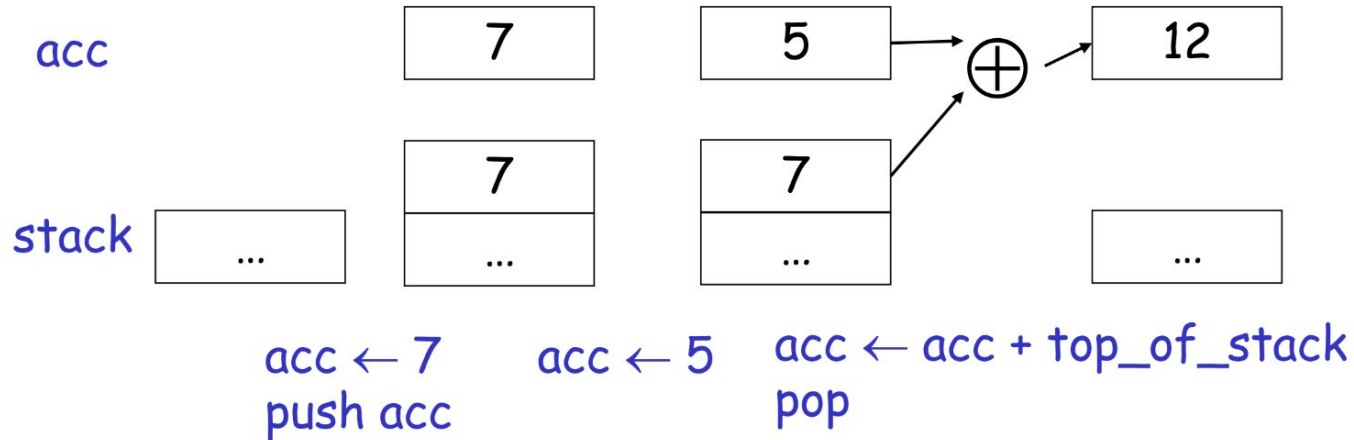
Stack Machine. Example



- Each instruction:
 - Takes its operands from the top of the stack
 - Removes those operands from the stack
 - Computes the required operation on them
 - Pushes the result on the stack

Stack Machine with Accumulator. Example

- Compute $7 + 5$ using an accumulator



In practice, with RISC-V:

$7 + 5$

`acc <- 7`

`push acc`

`acc <- 5`

`acc <- acc + stack_top`

`li a0, 7`

`sw a0, 0(sp); addi sp, sp, 4`

`li a0, 5`

`lw t1, 4(sp); add a0, a0, t1`

`pop`

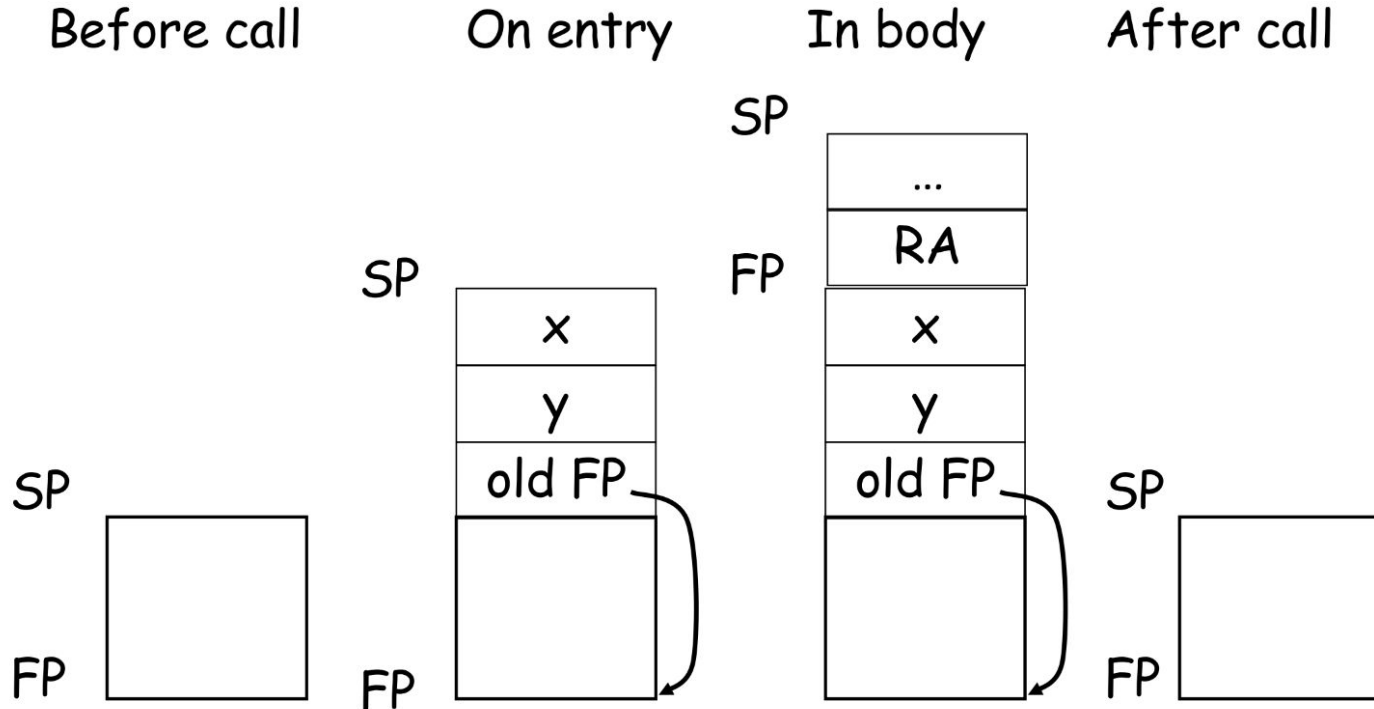
The cgen function

```
cgen(e1 + e2) =  
  cgen(e1)  
  push a0  
  cgen(e2)  
  t1 <- top  
  add a0, t1, a0  
  pop
```

```
cgen(f(e1,...,en)) =  
  push fp  
  cgen(en)  
  push a0  
  ...  
  cgen(e1)  
  push a0  
  jal f_entry
```

```
cgen(def f(x1,...,xn) = e) =  
  mv fp, sp  
  push ra  
  cgen(e)  
  ra <- top  
  addi sp, sp, z  
  lw fp, 0(sp)  
  jr ra
```

Calling Sequence. Example for $f(x,y)$.



What if we pre-allocate space for temps?

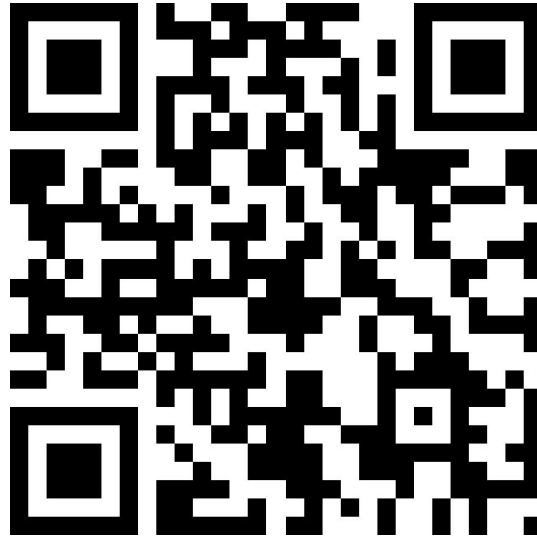
SP

FP

Temp NT(e)
...
Temp 1
RA
x_1
...
x_n
Old FP

Code Generation for + (revised)

```
cgen( $e_1 + e_2$ , nt) =  
    cgen( $e_1$ , nt)  
    sw a0, -nt(fp)  
    cgen( $e_2$ , nt + 4)  
    lw t1, -nt(fp)  
    add a0, t1, a0
```



Anonymous feedback form:
<http://tinyurl.com/SoraDisFeedback>