



# ISA

Patentable?

Instruction Set Architecture

=  
+=  
&=  
reg mem imm  
go to it

- =

\* =

/ =

\*  
+  
/  
-  
%  
% =

all in fbar  
and all reg - mem

⋮

Jump

← =  
= =  
! =  
⋮  
always

Copyright

How - can i copy

Parent

what - can i do

```

main( ) {
  PC+3 → ret
  goto foo
  foo( )
  print "."
}
foo( )
  PC+3 → ret
  goto foo
foo( )
  PC+3 → ret
  goto foo
buzz( )
  PC+3 → ret
  goto buzz
}

```

```

foo:
  print .
  goto ret

```

```

addr      instr
20        PC+3 → ret
21        goto foo
23

```

```

buzz( ) { ret → r17
  foo( ) PC+3 → ret
         goto foo
  foo( ) PC+3 → ret
         goto foo
         goto r17
}

```

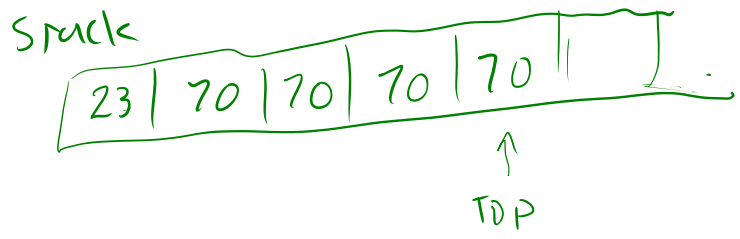
recursion

```

foo(x) {
  if(x > 0)
    foo(x-1)
  else print "hi"
}

```

↓  
byte 70



call foo / push addr of add  
add go to foo

ret POP into PC

reg  
Stack pointer

condition codes  
Flags

ney?  
zero?  
overflow?  
:

Jumps

< 0  
= 0

Subtract

goto if

~~lt=0~~

goto if lt= zero

!e  
e  
ne  
not equal to zero

x86-64 - laptop desktop

ARM 7  
8  
9 - phone tablet

r[a] = r[b] if le

↳ almost any inst can be conditional