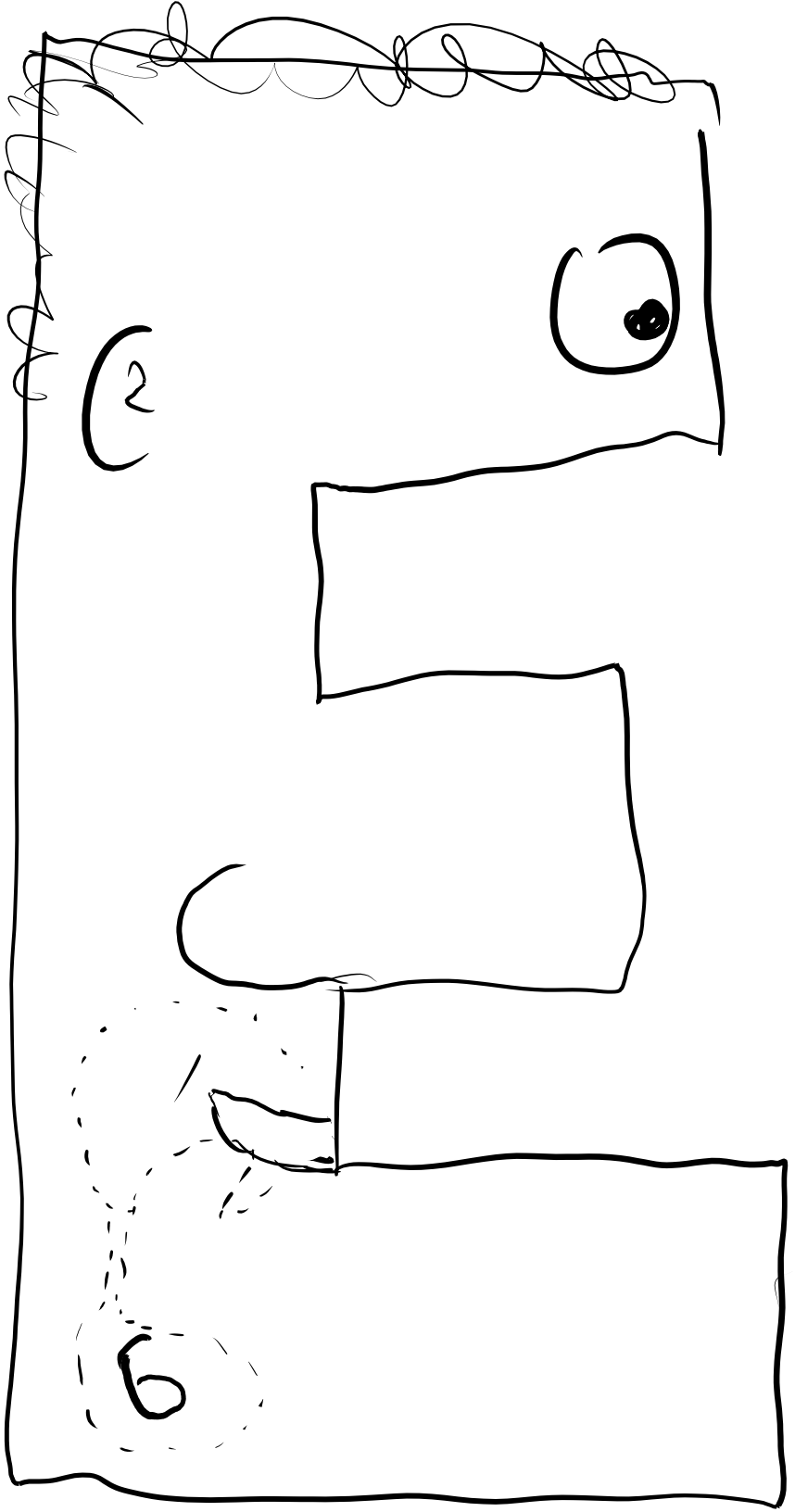


012345 789

8086 x86

80286

80386

80486

Pentium

12(%rax, %rcx, 8)

encoding, variable-length

# of instructions

CISC

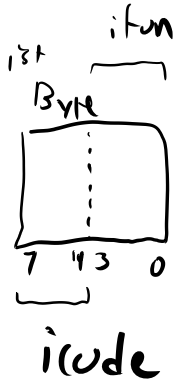
Complex Instruction Set Computer

RISC

Reduced Instruction Set Computer

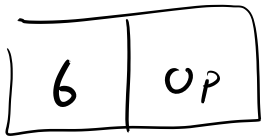
Halt 00

nop 10

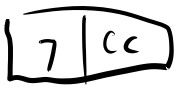


OPq

addq
subq
xorq
andq
⋮

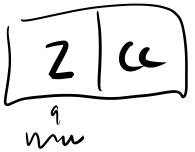


jmp
jle
jl
je
jne
jge
jg

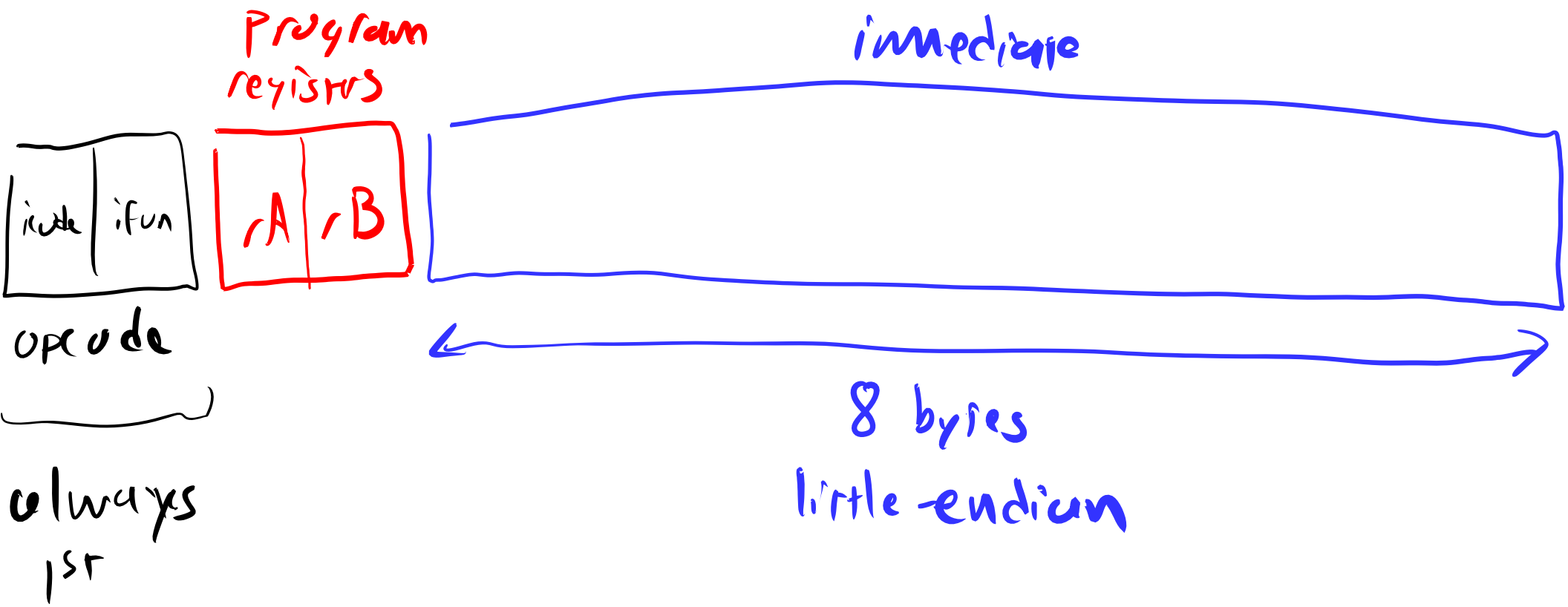


address

cmovle %rax, %rdx



registers


Program registers
immediate
icode
ifun
rA
rB
opcode
always 1st
8 bytes
little-endian

x86-64

%rax

$123

(%rax)

123

123(%rax)

123(%rax, %rdx)

⋮

123(%rax, %r8, 4)

1248

movq =

Y86-64

all inst: only 1 addr mode

OPq %r9, %r10

Jmp 123

callq 123

---

irmovq $123, %rax

rrmovq %rax, %rcx

mrmovq 123(%rax), %rcx

rmmovq %rcx, 123(%rax)

callq ~~foo~~ imm addr of addq

:

~~foo~~:
addq %rax, %rbx

x86
movq $12, (%rax)

irmovq
rmmovq

13 an address
imm → mem

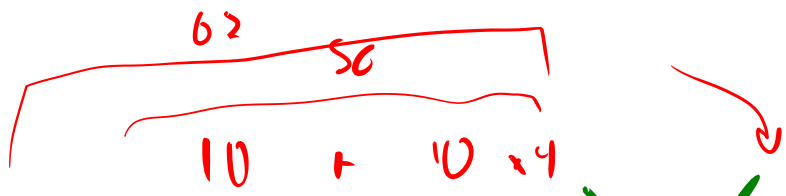
assembly
not
machine code

labels

immediate

cc
flag bits — implement then
not in code

cmpq — subq



leaq 12(%rax, %rax, 4), %rbx

```
rrmovq  %rax, %r14
addq    %rax, %rax
addq    %rax, %rax
addq    %r14, %rax
irmovq  $12, %r14
addq    %r14, %rax
rrmovq  %rax, %rbx
```

rax	62
rbx	62
r14	12


x86-64
Transl.
μop
microops