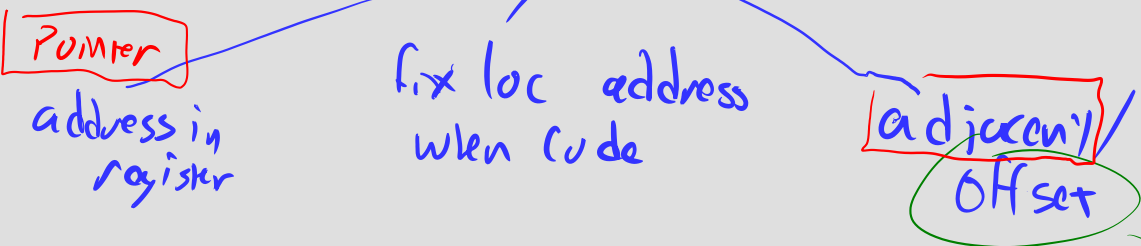




What can we do w/ ISA parts

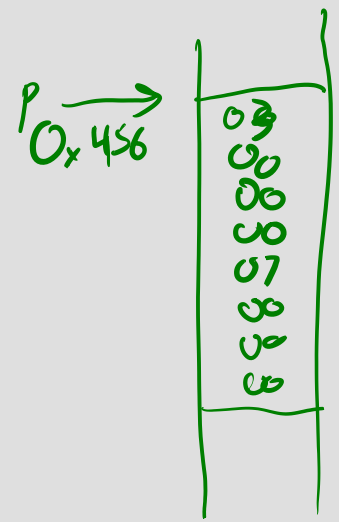
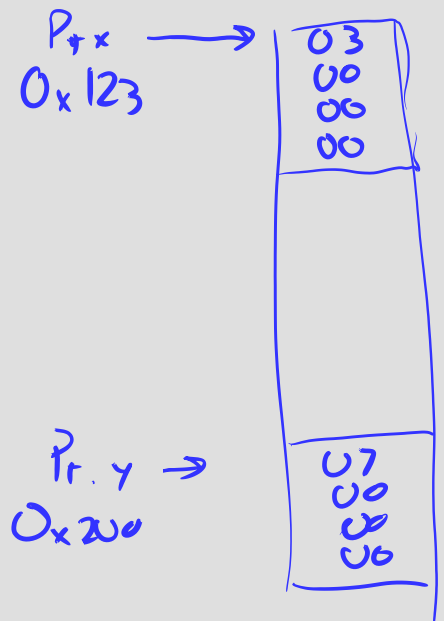
• data → bits  
find



Class Pt {

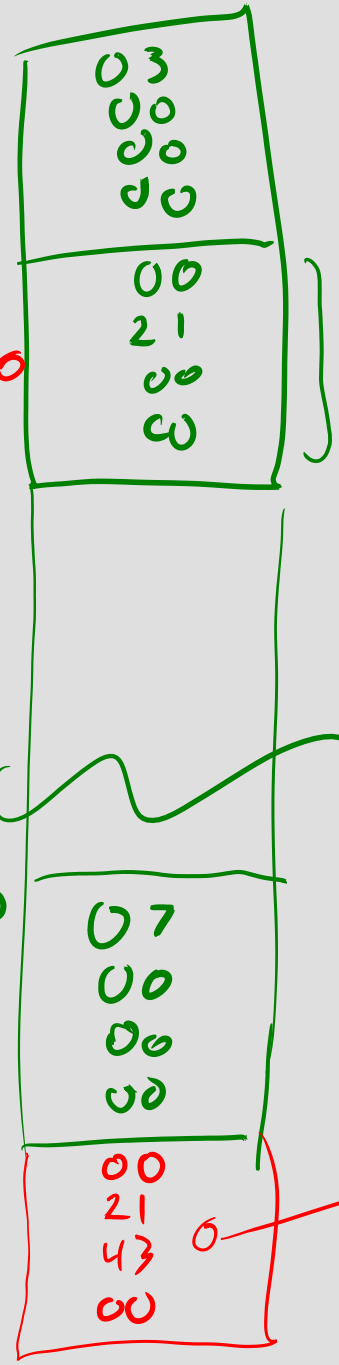
int x; → 32 bit 2's comp  
int y; → 32 bit 2's comp

p = new Pt(3, 7)



addr  
P.x = P  
addr  
P.y = P+4

P → 0x200



address of new port

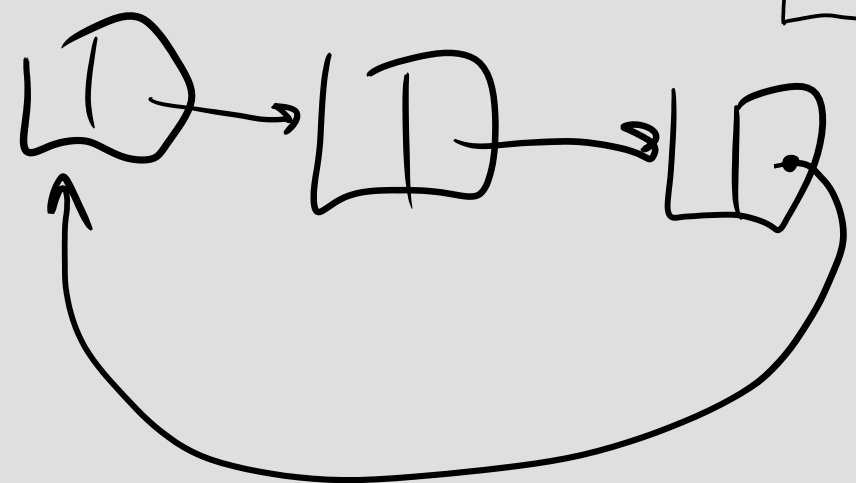
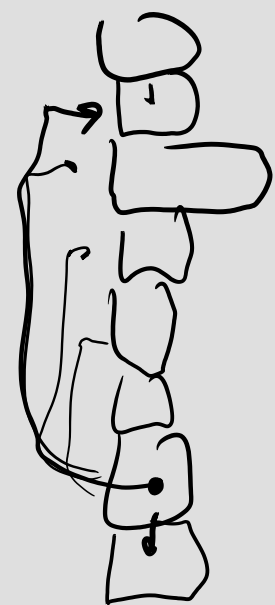
0x2100

0x432100



Program → a lot of instructions

- adj
  - jump
  - both
  - neither
- pc+1 offset  
icode 7 → Jump



if

while

for

functions

+  $\left( \begin{array}{l} \text{int} \\ \text{fp} \end{array} \right)$

$$\begin{array}{r} 00011111 \\ + 00011111 \\ \hline 00111110 \end{array}$$
  

$$00011111$$

$1.11 \times 2^0$

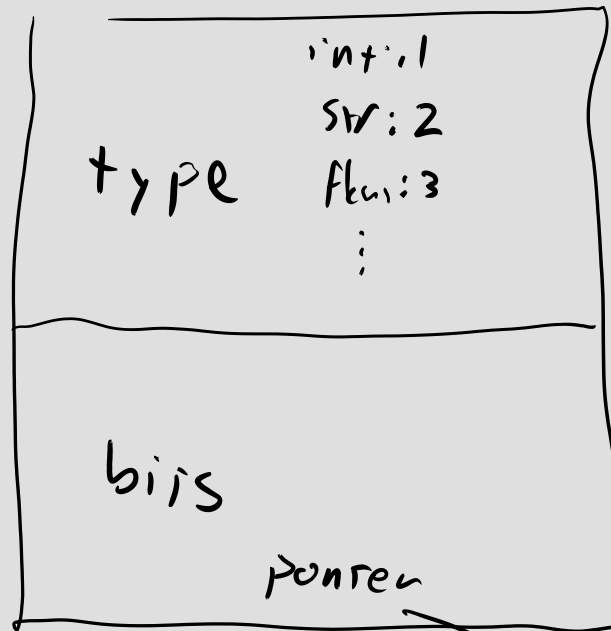
int  $x = 3$

$x = x +_{fp} x$

$1.111 \times 2^{-4}$

$\rightarrow 0.0001111$

$x = 3$   
 $x = x * 2 \rightarrow 6$   
 $x = "hi"$   
 $x = x * 2 \rightarrow "hihi"$



\*  
if type = int  
==  
else x) = str  
==  
==

interface A {

int foo()

}

class B implements A {

int foo() { return 7; }

}

class C implements A {

foo() { return 8; }

}

A x = new B();

x.foo();

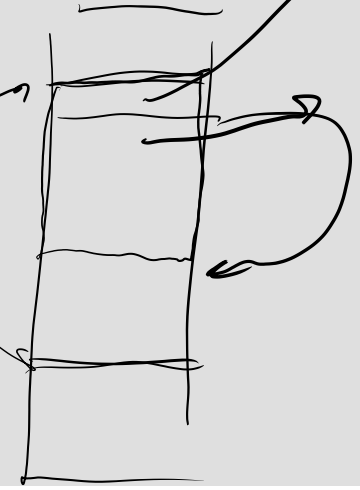
x → fptr → +8

x = new B();

x.foo();

foo → run  
+0

x [ addr ]





New

$x = \text{int } [8];$

code address [ ]

```
switch (int) {  
  case 0:  
  case 1:  
  :  
}
```

x →



```
run {  
  code,  
  code,  
  code,  
} [int]
```