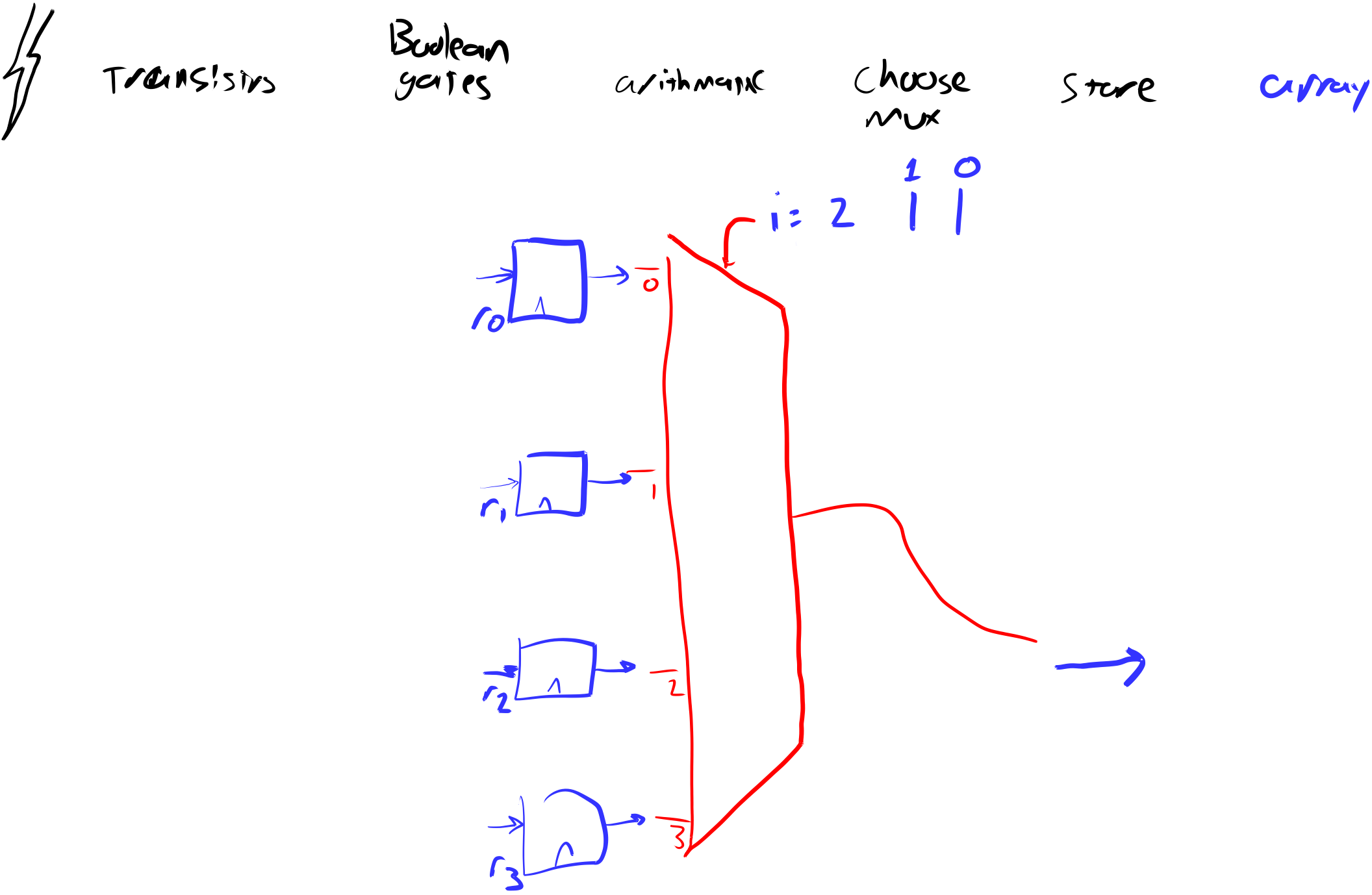
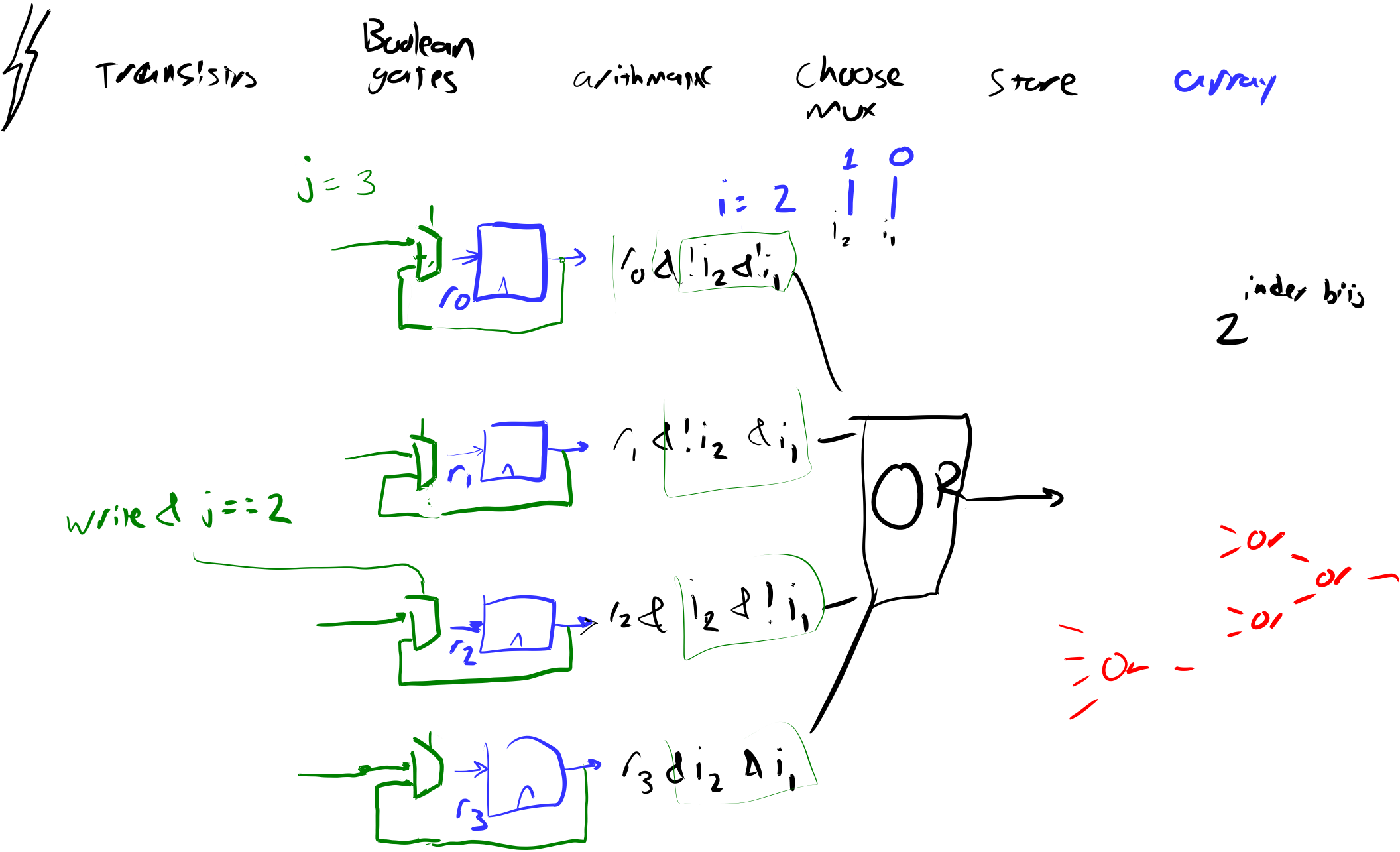

transistors
Boolean gates
arithmetic
choose mux
store
array
i: 2 1 0
r0
r1
r2
r3
0
1
2
3


Transistors
Boolean gates
arithmatic
Choose mux
Store
array
j = 3
i = 2
1 0
i_2 i_1
$r_0 \& !i_2 \& !i_1$
$r_1 \& !i_2 \& i_1$
$r_2 \& i_2 \& !i_1$
$r_3 \& i_2 \& i_1$
OR
write & j == 2
r_0
r_1
r_2
r_3
index bits
$2^{}$
or
or
or
or

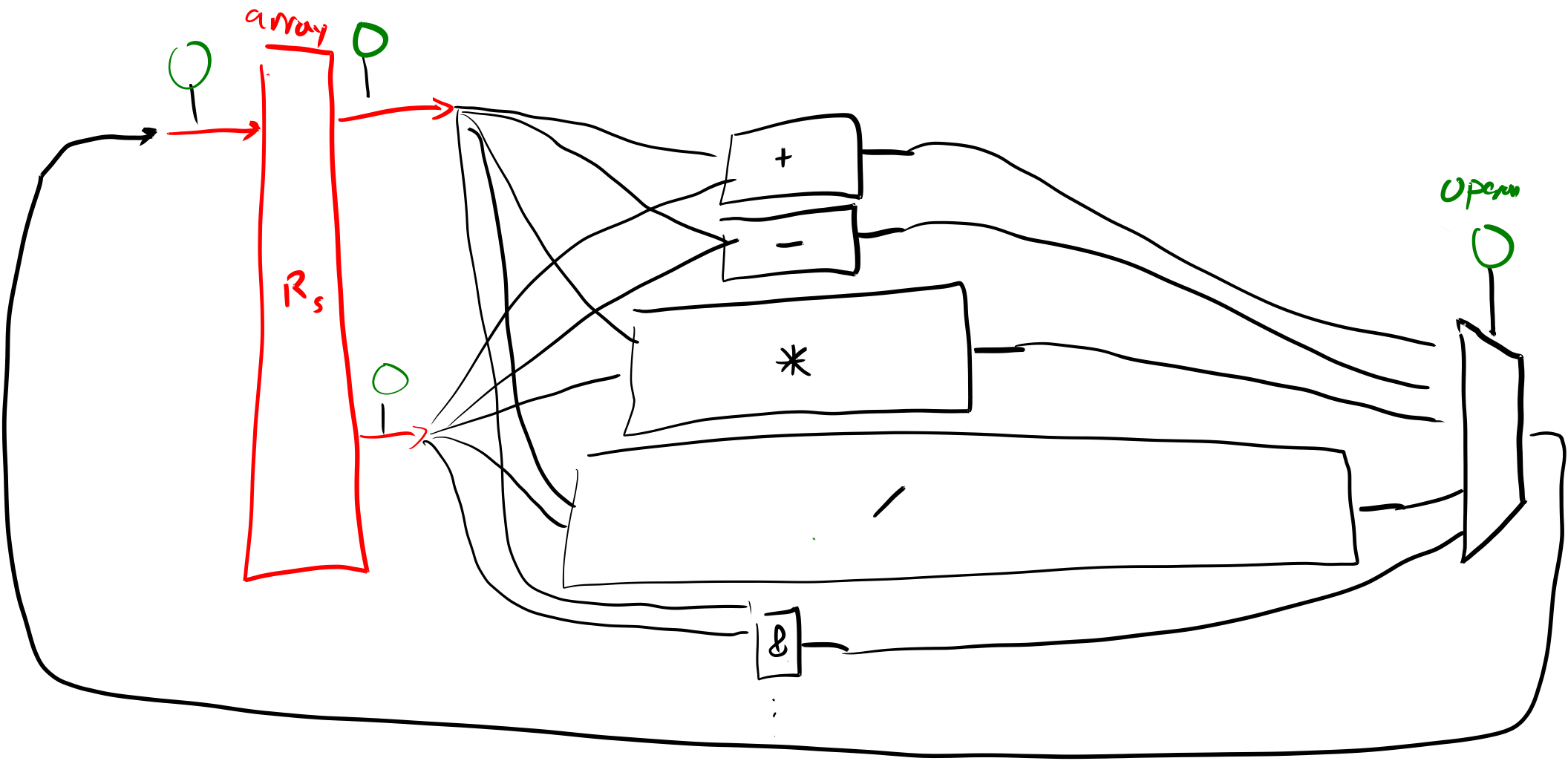
Code

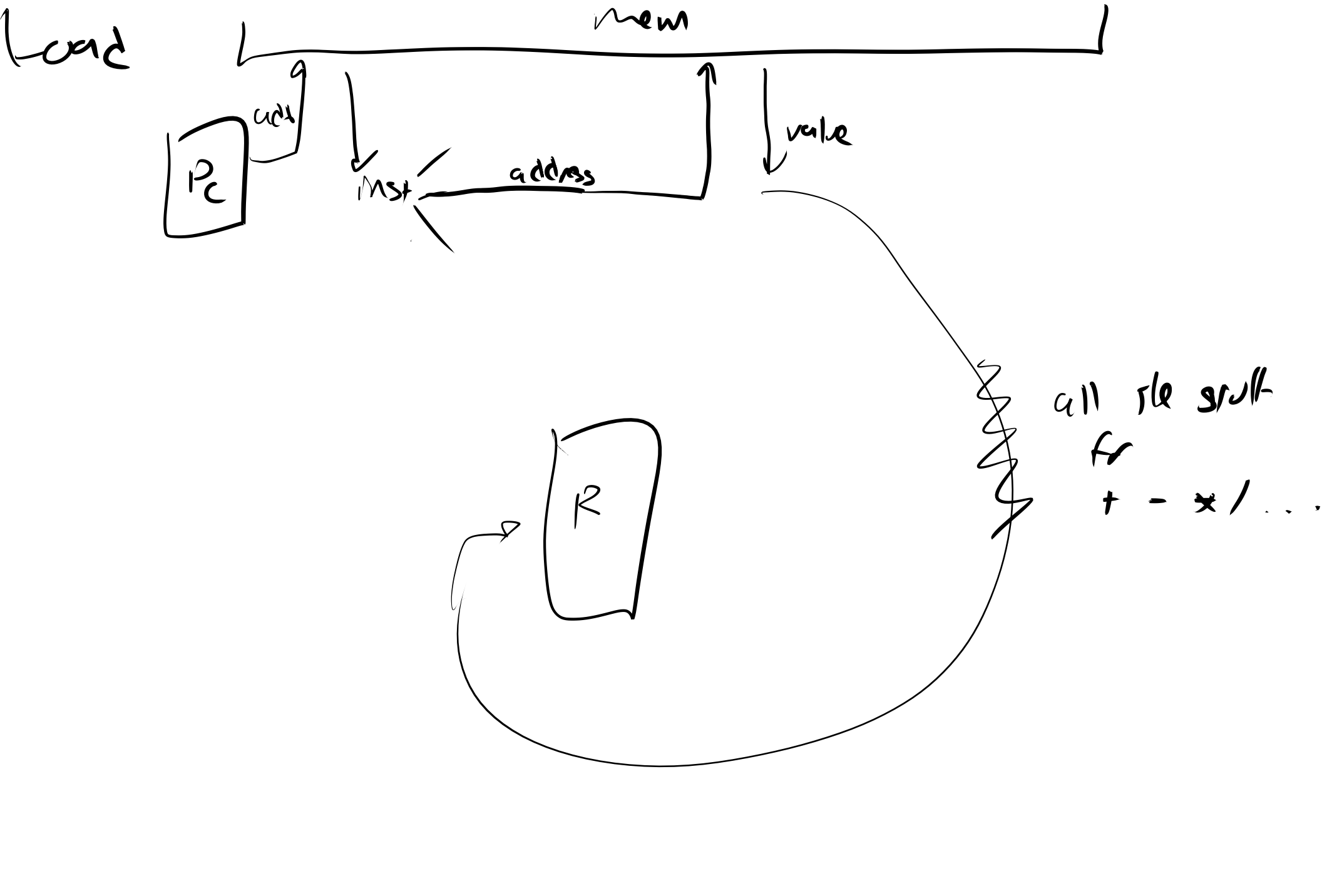
list of actions

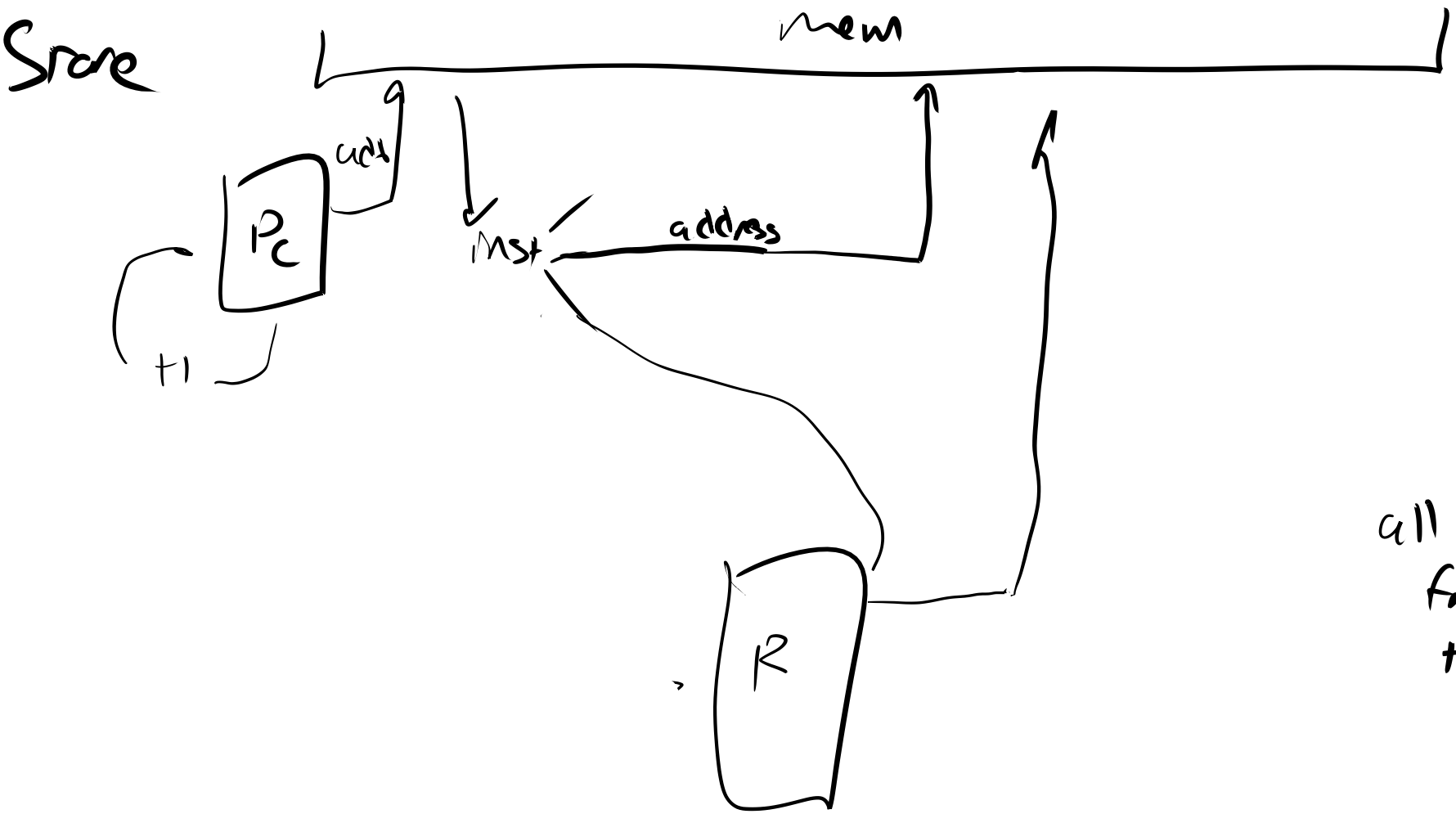
- ✓ $r_0 = r_1 + r_2$
- $r_0 = Memory[r_1]$
- $Memory[r_0] = r_1$
- if r_0 then goto r_1

$x = *y$

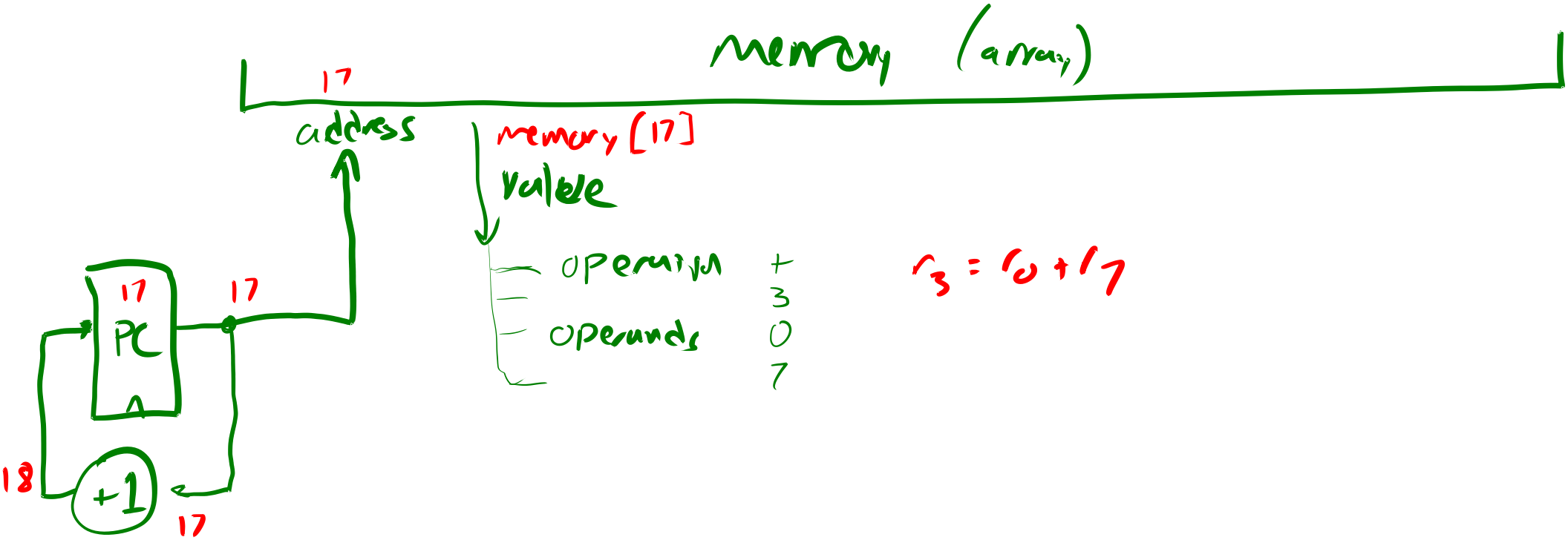
$r_0 = r_1 + r_2$

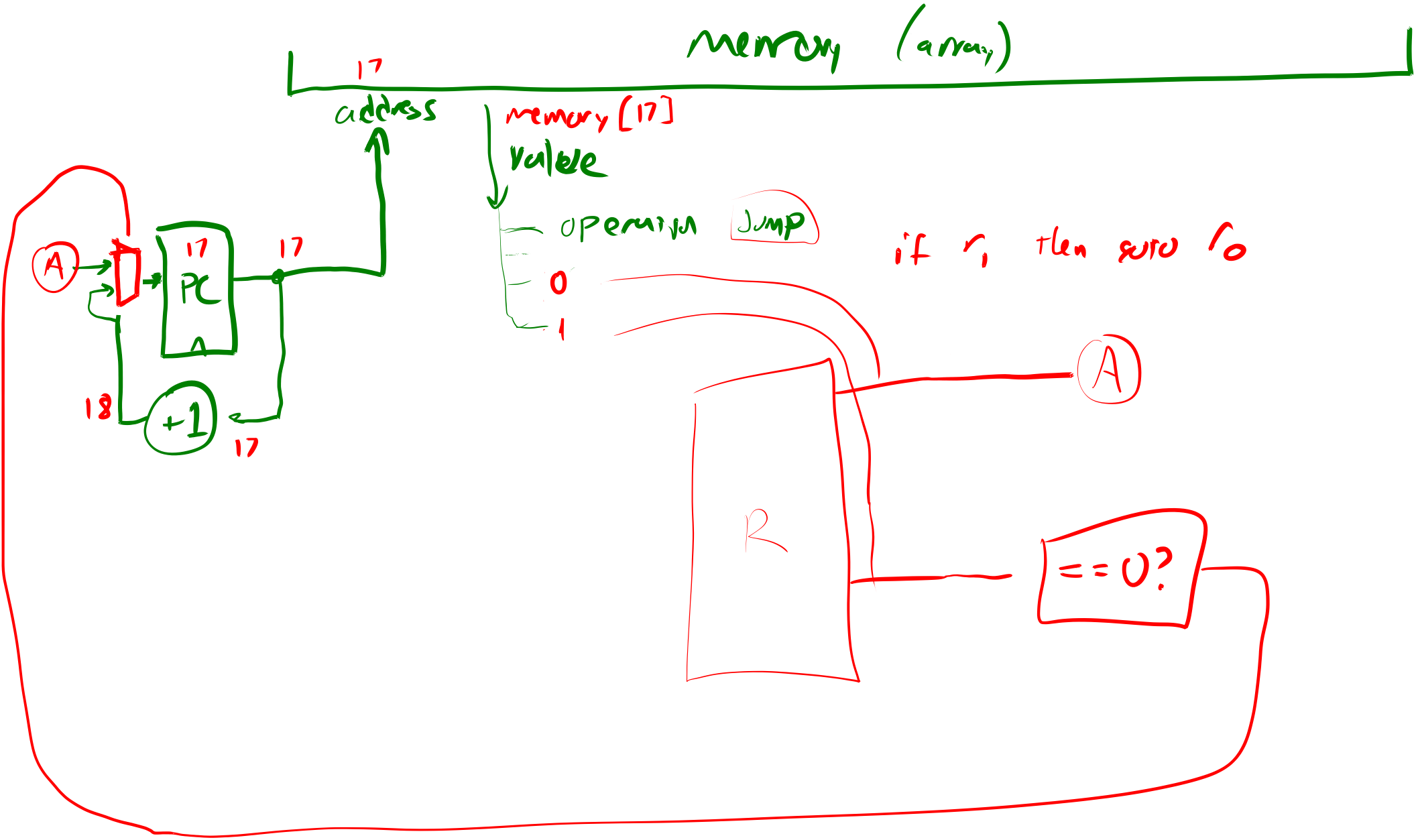


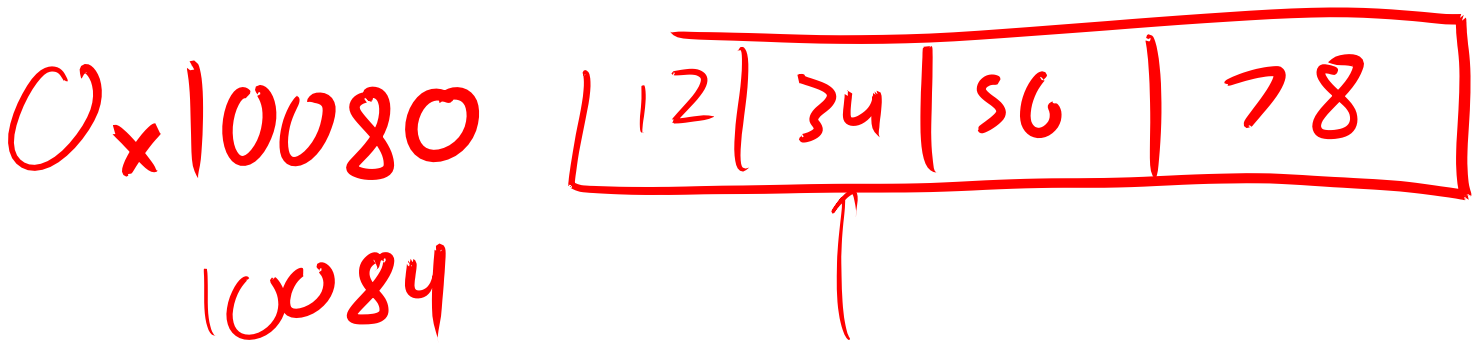

load
mem
PC
addr
inst
address
value
R
all the stuff
for
+ - * / ...



all the stuff
for
+ - * / . . .


memory (array)
17
address
memory[17]
value
operation
+
3
operands
0
7
17
17
PC
18
+1
17
r3 = r0 + r7


memory (array)
17
address
memory[17]
value
operation
Jump
if r1 then goto r0
0
1
A
17
PC
17
18
+1
17
A
R
==0?

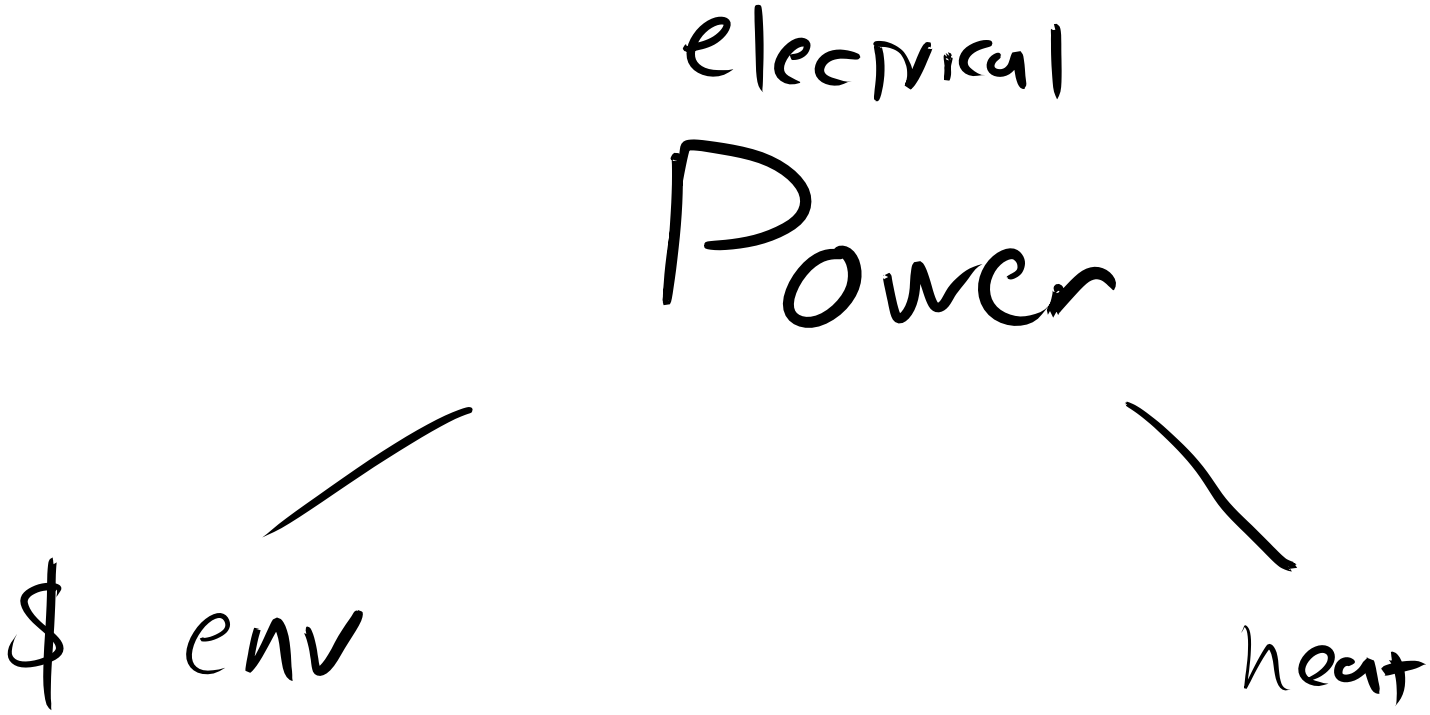


int *xx = ...;

char *xy = x;

(x+1) == (y+1) → false

(x) == (y) → true



16GiB

memory

128Gib

2^{37} bits