

# Circuits and Code

---

## CS 2130: Computer Systems and Organization 1

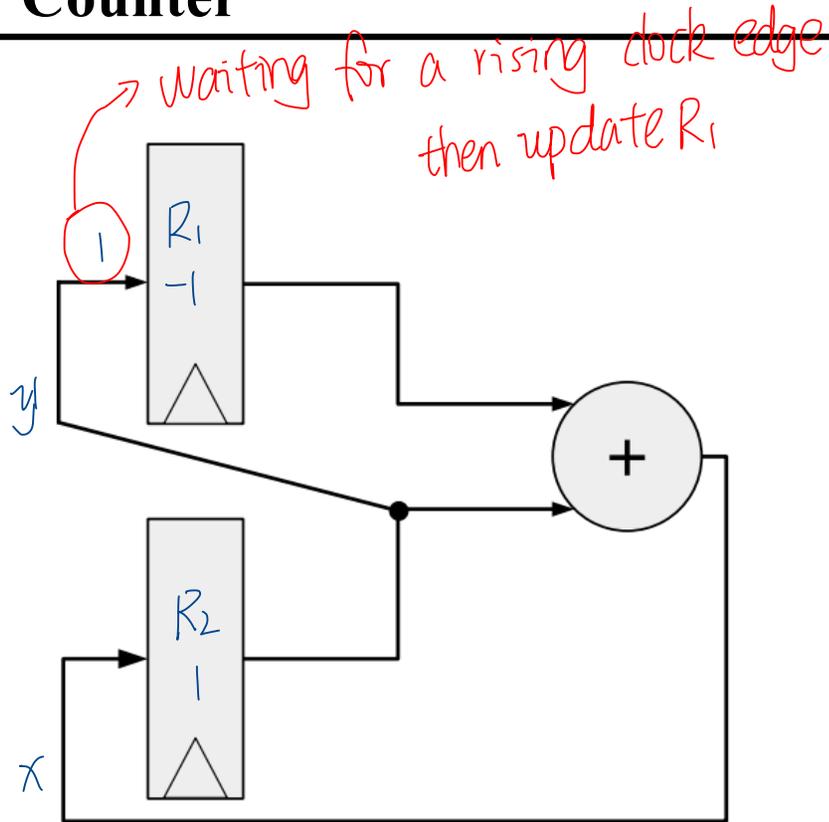
**Xinyao Yi** Ph.D.  
Assistant Professor

## Announcements

---

- Homework 1 due tonight
- Homework 2 available today, due next Monday

## Another Counter

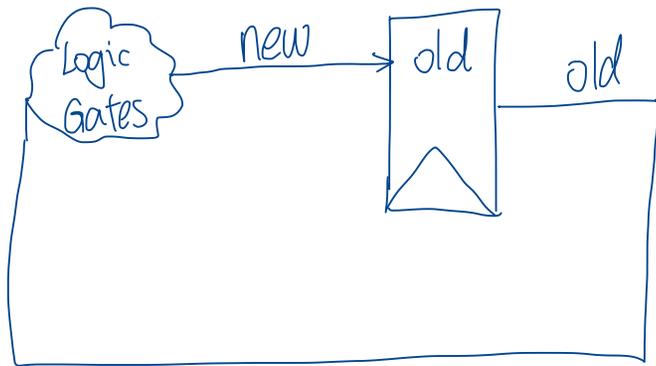


clock	$R_1$	$R_2$	$x$	$y$
0	-1	1	0	1
1	1	0	1	0
2	0	1	1	1
3	1	1	2	1
4	1	2	3	2
5			5	3

$x$ : Fibonacci Sequence

## Common Model in Computers

---



The register ignore  
all the calculations / updates  
until rising clock edge.

## Code to Build Circuits from Gates

*We are going to build the computer.*

Write code to build circuits from gates

- Gates we already know:  $\&$ ,  $|$ ,  $\wedge$ ,  $\sim$
- Operations we can build from gates:  $+$ ,  $\ominus$
- Others we can build:

*you will build one in the lab/assignment.*

$$\begin{array}{r}
 * : \quad 2130 \\
 \times \quad 1101 \\
 \hline
 \quad \quad 2130 \\
 \quad \quad 0000 \\
 \quad 2130 \\
 \hline
 2130
 \end{array}$$

*(left shift and add)*

*/, %  
↔ build divide, you got module for free.*

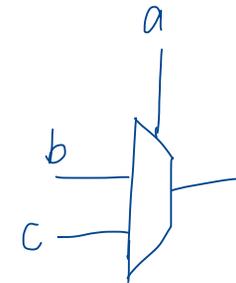
## Code to Build Circuits from Gates

---

Write code to build circuits from gates

- Gates we already know:  $\&$ ,  $|$ ,  $\wedge$ ,  $\sim$
- Operations we can build from gates:  $+$ ,  $-$
- Others we can build:
- Ternary operator:  $?:$

$a == 0 ? b : c$

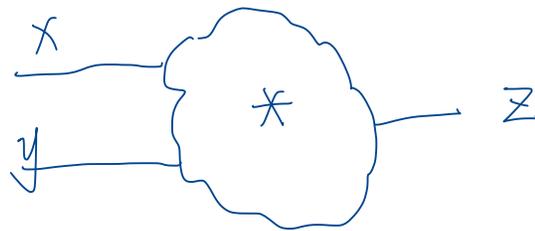


## Equals

---

Equals: =

- Attach with a wire (i.e., connect things)
- Ex:  $z = x * y$



attaching the circuit to  
whatever I'm trying to output

## Equals

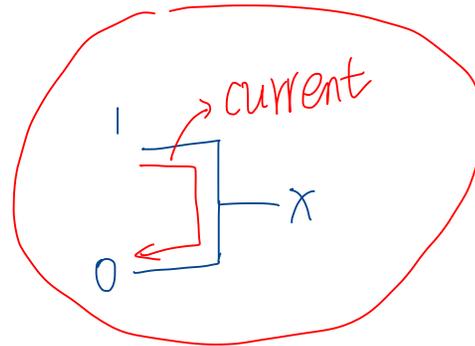
Equals: =

- Attach with a wire (i.e., connect things)
- Ex:  $z = x * y$
- What about the following?

$x = 1$

$x = 0$

I tried to  
re-write  $x$ ,  
but for hardware



doesn't work!

level, it doesn't work.

## Equals

---

Equals: =

- Attach with a wire (i.e., connect things)
- Ex:  $z = x * y$
- What about the following?  
 $x = 1$   
 $x = 0$
- **Single assignment:** each variable can only be assigned a value once

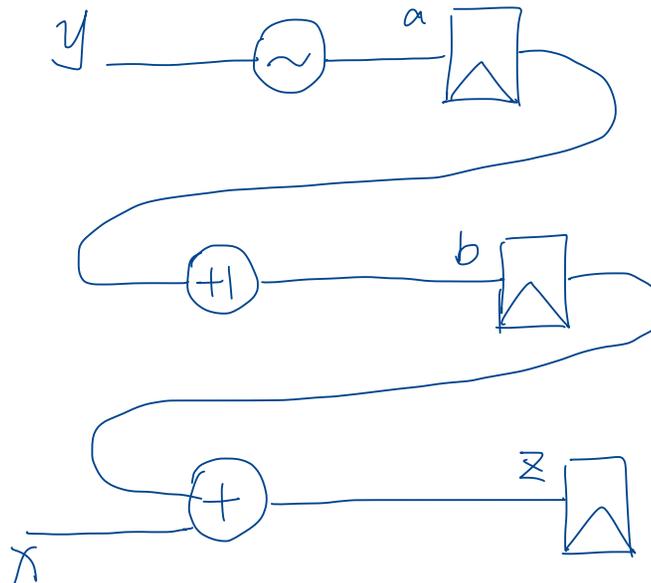
# Subtraction $z = x - y = x + (-y) = x + \sim y + 1$

$$z = x + \sim y + 1$$

$$a = \sim y$$

$$b = a + 1$$

$$z = x + y b.$$



## Comparisons

---

Each of our comparisons in code are straightforward to build:

- `==` - xor then nor bits of output
  - bits are equal iff the XOR is 0.

Compute  $d = x \oplus y$

if all bits of  $d$  are 0  $\rightarrow x == y$

## Comparisons

---

Each of our comparisons in code are straightforward to build:

- `==` - xor then nor bits of output
- `!=` - same as `==` without not of output

## Comparisons

---

Each of our comparisons in code are straightforward to build:

- `==` - xor then nor bits of output
- `!=` - same as `==` without not of output
- `<` - consider  $x < 0$

if  $x$  is 32 bits:

$$(x \gg 31) \& 1$$

We want to check the sign bit: if sign bit is 1, it's negative

## Comparisons

---

Each of our comparisons in code are straightforward to build:

- `==` - xor then nor bits of output
- `!=` - same as `==` without not of output
- `<` - consider  $x < 0$
- `>`, `<=`, `=>` are similar

$$x < y \Rightarrow x - y < 0 \quad ((x + (\sim y + 1)) > 31) \& 1$$

## Indexing

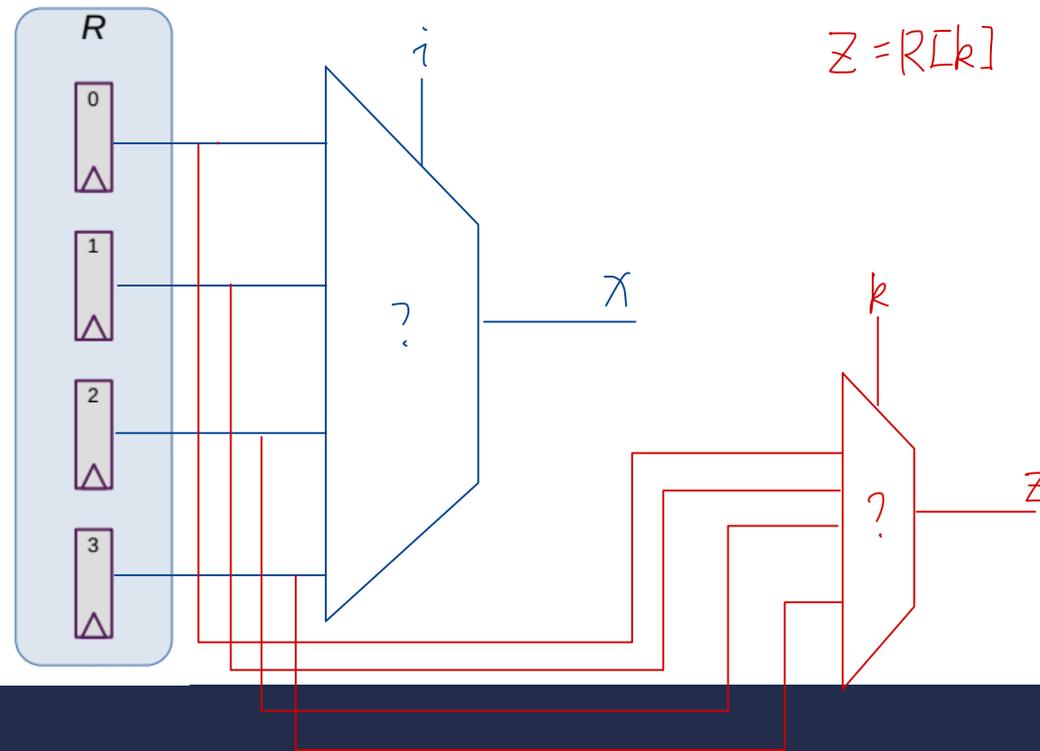
---

Indexing with square brackets: [ ]

- **Register bank (or register file)** - an array of registers
  - Can programmatically pick one based on index
  - I.e., can determine which register while running
- Two important operations:
  - $x = R[i]$  - Read from a register
  - $R[j] = y$  - Write to a register

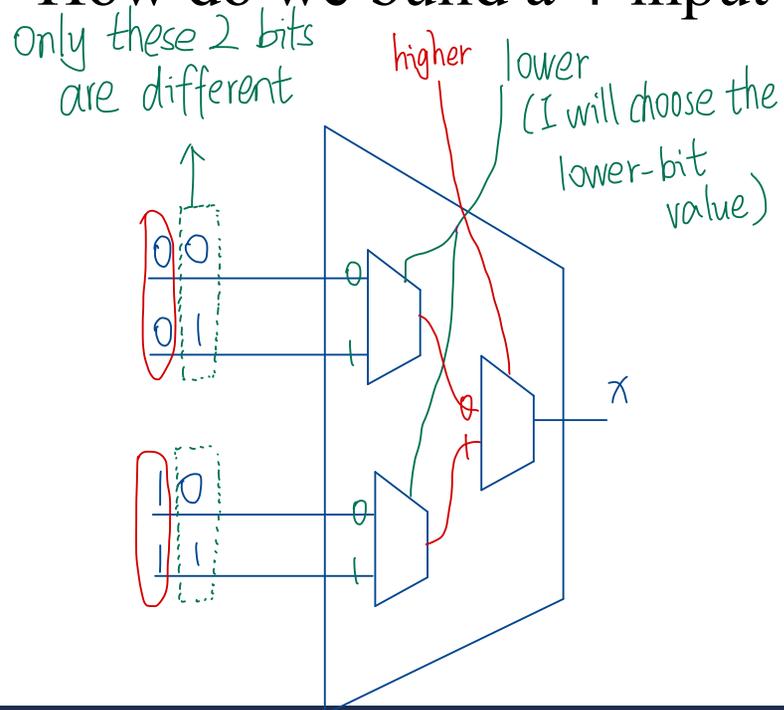
# Reading

$x = R[i]$  - connect output of registers to  $x$  based on index  $i$



## Aside: 4-input Mux

How do we build a 4-input mux? How many wires should  $i$  be?



$i$  should be 2 bits.

How about 8-input mux?

$2^3 = 8$  so  $i$  should be 3 bits.

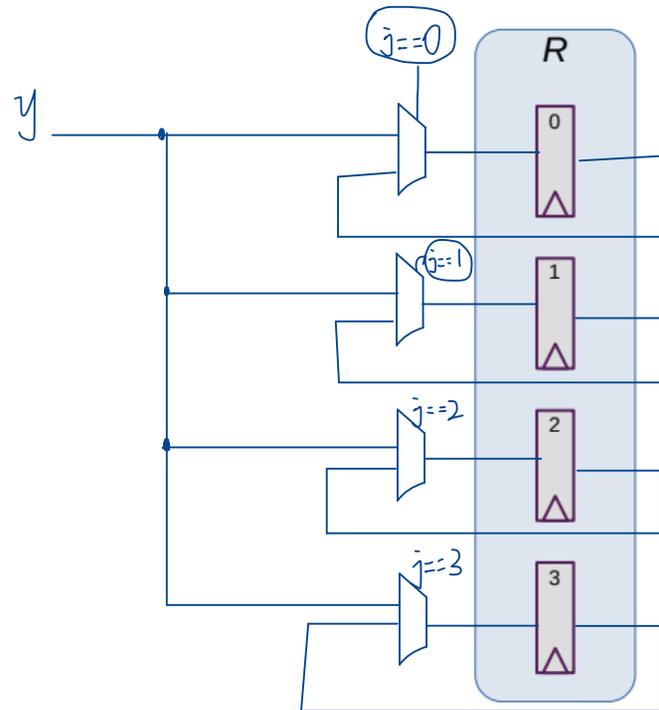
## Writing

$R[j] = y$  - connect  $y$  to input of registers based on index  $j$

2 things to be considered.

①. How do I choose which register I want to hook it up to?

②. For all other registers that I'm not writing  $y$  to, they need to keep the current value. I don't want to lose it.

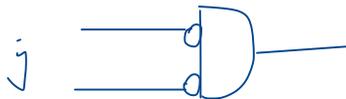


## Aside: Creating $== 0$ gates

How do we build gates that check for  $j == w$ ?

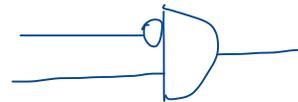
$\bar{j} == 0$

$j$	$== 0$
0 0	1
0 1	0
1 0	0
1 1	0



$j == 1$

$j$	$== 1$
0 0	0
0 1	1
1 0	0
1 1	0



---

Need one more thing to build computers

*We need a place to store stuff.*