

Building Computers: Fetch-Decode-Execute

CS 2130: Computer Systems and Organization 1

September 16, 2022

Announcements

- Homework 2 due on Gradescope, 11pm Monday
- Quiz 3 available at 5pm, due Monday at 8am
 - 24-hours to complete once opened
 - Do **not** submit until you are finished (you should save!)

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: $?$ $:$

Equals

Equals: =

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

Comparison operators

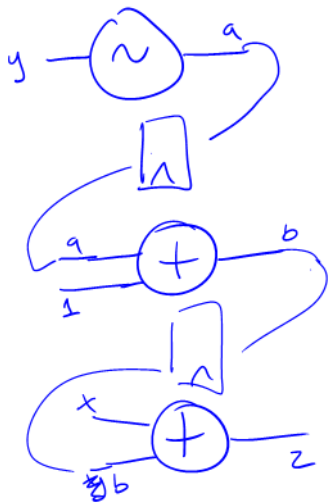
- == - xor then nor bits of output
- != - same as == without not of output
- < - consider $x < 0$
- >, <=, => are similar

Subtraction

$$a = \sim y$$

$$b = a + 1$$

$$z = x + \cancel{a} b$$



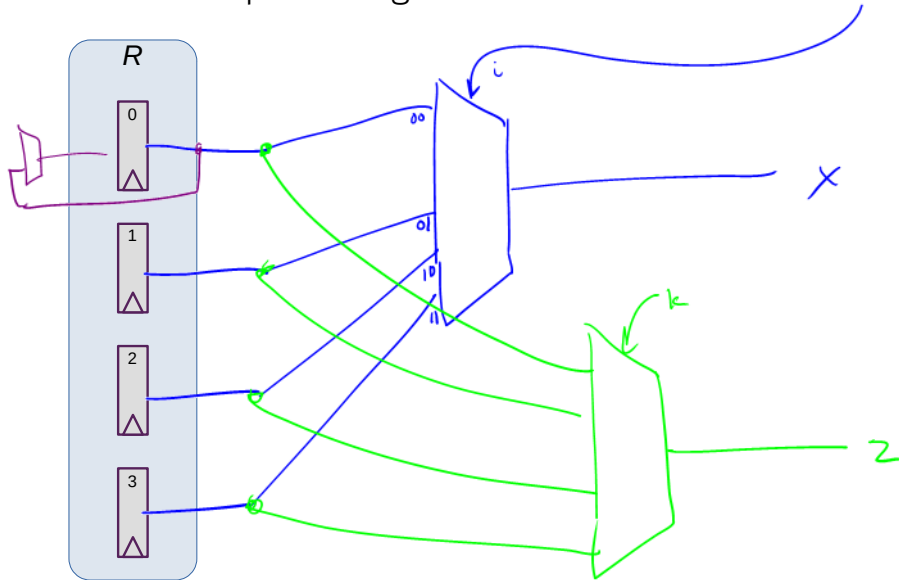
Indexing

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

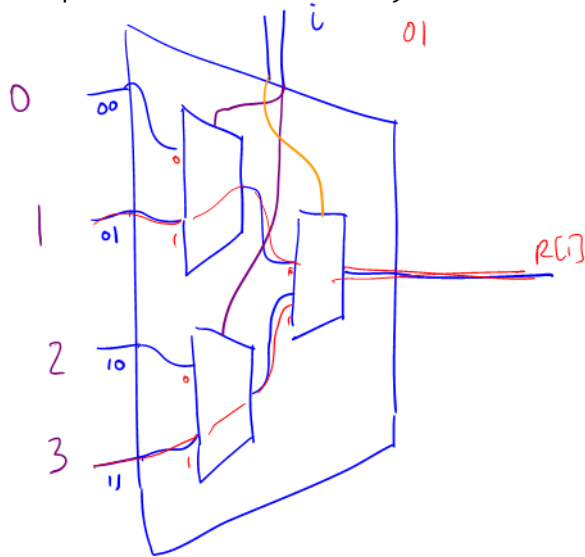
Reading

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



Aside: 4-input Mux

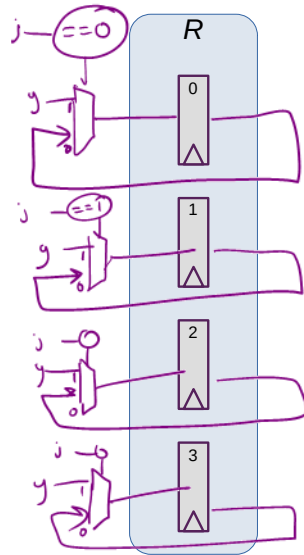
How do we build a 4-input mux? How many wires should i be?



Writing

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

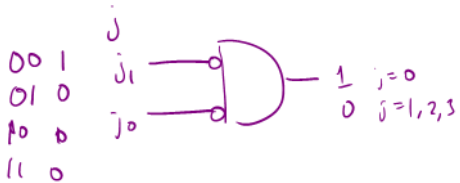
$j=1$
 $R[1]=y$



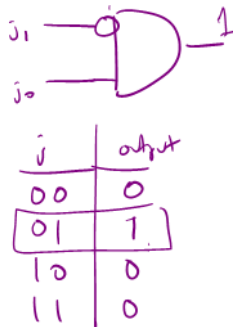
Aside: Creating $==0$ gates

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

$== 0$



$== 1$



Need one more thing to build computers

Memory and Storage

Registers

≈ KiB

- 6 gates each, ≈ 24 transistors
- Efficient, fast
- Expensive!
- Ex: local variables

Memory

≈ GiB

- Two main types: SRAM, DRAM
- DRAM: 1 transistor, 1 capacitor per bit
- DRAM is cheaper, simpler to build
- Ex: data structures, local variables

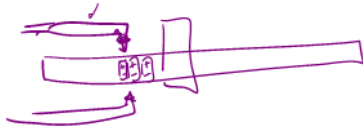
These do not persist between power cycles

Memory and Storage

Disk

≈ GiB-TiB

- Two main types: flash (solid state), magnetic disk
- Magnetic drive
 - Platter with physical arm above and below
 - Cheap to build
 - Very slow! Physically move arm while disk spins



- Ex: files

Data on disk does persist between power cycles

Putting it all together

Code

How do we run code? What do we need?

Example Code

...

8: x = 16

9: y = x

10: x += y

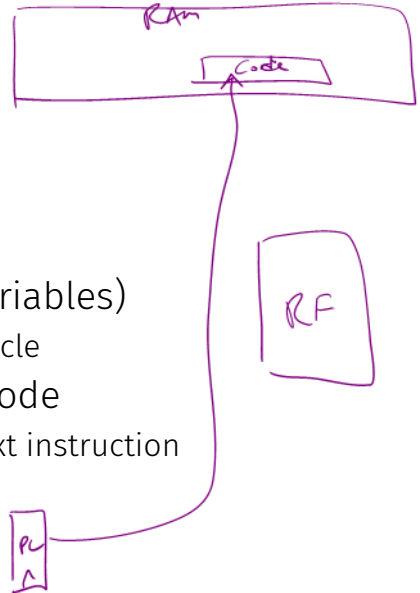
...

What is the value of x after line 10? 32

Bookkeeping

What do we need to keep track of?

- **Code** - the program we are running
 - RAM (Random Access Memory)
- **State** - things that may change value (i.e., variables)
 - Register file - can read and write values each cycle
- **Program Counter (PC)** - where we are in our code
 - Single register - byte number in memory for next instruction



Building a Computer

Building a Computer

$$y = x^{r2}$$
$$x + = y \leftarrow$$

$$y = x \rightarrow 5 + 3$$

