Binary Arithmetic

CS 2130: Computer Systems and Organization 1

August 29, 2022

Course Content

Where do I go to find course material?

- Collab: central hub for 2130 this semester
  - Course website for all content
  - Lecture recordings on Panopto
  - Q&A discussion on Piazza
  - Submit assignments through Gradescope
- Community and online TA office hours on Discord

Expectations and Evaluations

Covid-19 Policies

- Masks are always welcome in class (I will be wearing one)
- No eating or drinking in the classroom
- Attendance is **not** required, but engagement is
  - Watch lecture videos
- If you don't feel well, stay home, it will be okay
  - Will work with you–if you stay home–to ensure no effect to grade

This is a Large Class

How can you get your questions answered?

- Piazza (!!)
  - If you know an answer to someone else's question, answer it!
  - We're in it together for the semester
- Discord
- TAs (office hours and labs)
- My office hours

Speaking of Office Hours!

TA Office Hours

- In-person office hours in the evening
- Online office hours throughout the day
- More information on Wednesday!

Office Hour poll

Professor Hott

Who am I? Why teach 2130?

Professor Hott

Who am I? Why teach 2130?

Professor Hott

Who am I? Why teach 2130?

Professor Hott

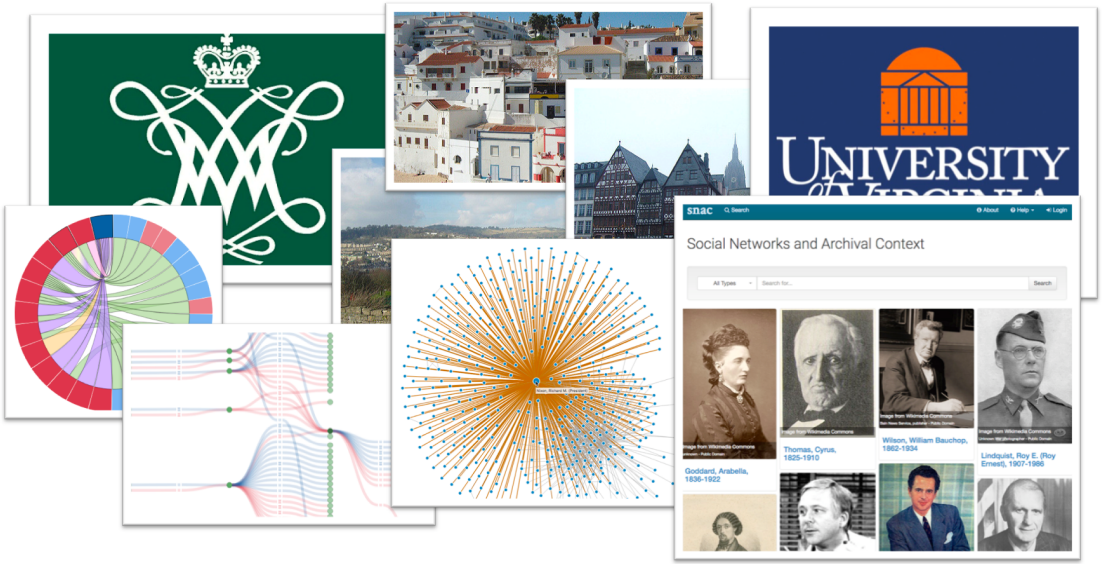
Who am I? Why teach 2130?

Professor Hott

Who am I? Why teach 2130?

Professor Hott

Who am I? Why teach 2130?



Questions?

So far...

So far, we have discussed:

- `and`, `or`, `not`, 0 and 1

So far...

So far, we have discussed:

- `and`, `or`, `not`, 0 and 1
- `nand`, `nor`, `xor`

So far...

So far, we have discussed:

- `and`, `or`, `not`, 0 and 1
- `nand`, `nor`, `xor`
- Transistors and how to make these gates (high level)

So far...

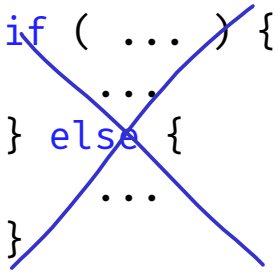
So far, we have discussed:

- `and`, `or`, `not`, 0 and 1
- `nand`, `nor`, `xor`
- Transistors and how to make these gates (high level)

Now let's build something powerful

Trinary Operator

General idea



Trinary operator

Trinary Operator

General idea

```
if ( ... ) {
    ...
} else {
    ...
}
```

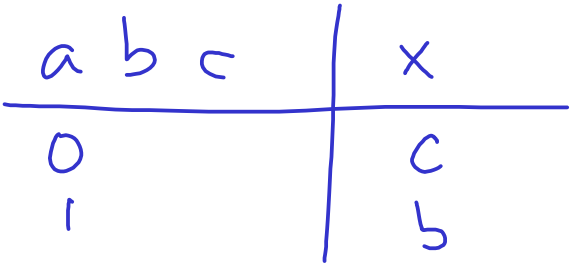
Trinary operator

- Python: `x = b if a else c`

Trinary Operator

General idea

```
if ( ... ) {
    ...
} else {
    ...
}
```

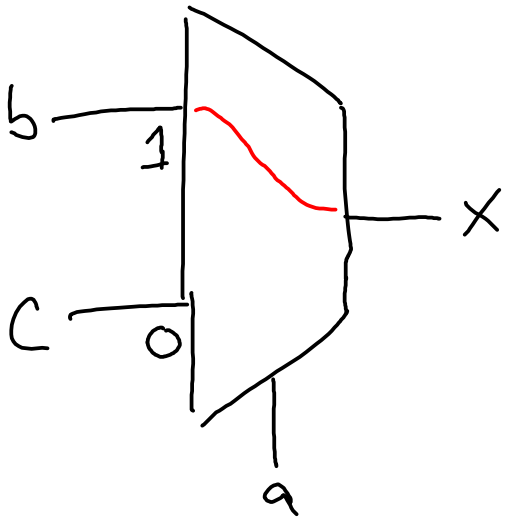


Trinary operator

- Python: `x = b if a else c`
- Java: `x = a ? b : c`

Multiplexer (mux)

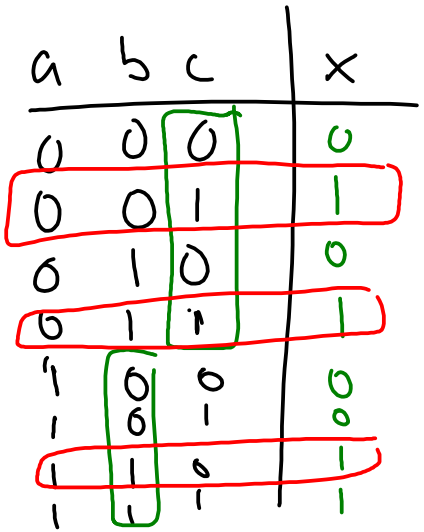
```
x = a ? b : c
```



Multiplexer (mux)

How can we build a mux out of what we have learned so far?

```
x = a ? b : c
```




```
(!a & !b & c) |
(!a & b & c) |
(a & b & !c) |
(a & b & c)
```

Multiplexer (mux)

Can be built from `and`, `or`, and `not`

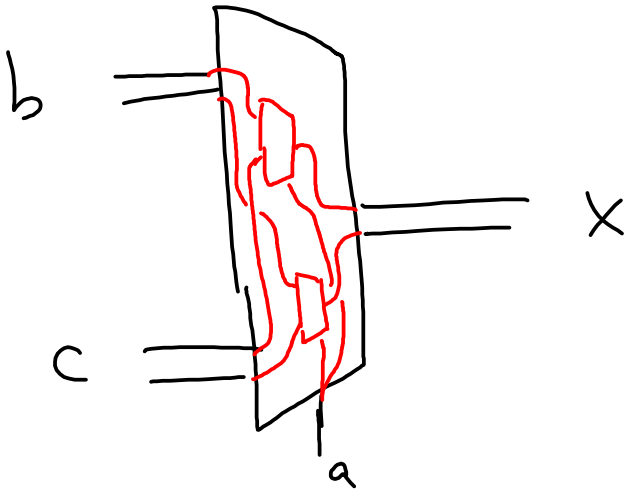
- Can be built using transistors
- Can physically put it in silicon!

Questions?

More bits!

2-bit Multiplexer (mux)

2-bit values instead of 1-bit values

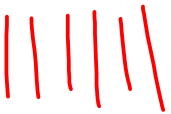


Multi-bit Values

- So far, only talking about 2 things
- Numbers, strings, objects, ...

Numbers

From our oldest cultures, how do we mark numbers?



Numbers

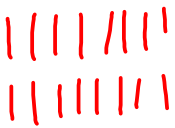
From our oldest cultures, how do we mark numbers?

- **unary** representation: make marks, one per ”thing”

Numbers

From our oldest cultures, how do we mark numbers?

- **unary** representation: make marks, one per ”thing”
  - Awkward for large numbers, ex: CS 2130?
  - Hard to tell how many marks there are



Numbers

From our oldest cultures, how do we mark numbers?

- **unary** representation: make marks, one per ”thing”
  - Awkward for large numbers, ex: CS 2130?
  - Hard to tell how many marks there are
- Update: group them!

Numbers

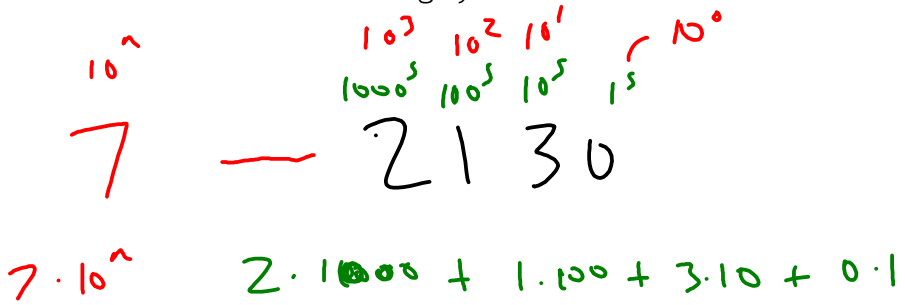
From our oldest cultures, how do we mark numbers?

- **unary** representation: make marks, one per ”thing”
  - Awkward for large numbers, ex: CS 2130?
  - Hard to tell how many marks there are
- Update: group them!
- Romans used new symbols: V L C D M

Numbers

From our oldest cultures, how do we mark numbers?

- Arabic numerals
  - Positional numbering system



0 1 2 3 4 5 6 7 8 9

Numbers

From our oldest cultures, how do we mark numbers?

- Arabic numerals
  - Positional numbering system
  - The `10` is significant:
    - 10 symbols, using 10 as base of exponent

Numbers

From our oldest cultures, how do we mark numbers?

- Arabic numerals
  - Positional numbering system
  - The `10` is significant:
    - 10 symbols, using 10 as base of exponent
  - The `10` is *arbitrary*
  - We can use other bases! $\pi, 2130, 2, \ldots$

Base-8 Example

Try to turn 134_8 into base-10: