QCI

Day 1: The classical computing paradigm

Introductions!

About me

Tanay Biradar

About me

Tanay Biradar

Bay Area, California

About me

Tanay Biradar

Bay Area, California

CS, speedcubing, 中文

Introduce yourselves!

Name

Where you're from

Interests

Classical Computing

Storing information

Storing information

Transistors → circuits

Storing information

Transistors → circuits

Logic gates

Bit

Smallest unit of information

8 bits = 1 byte

1000 bytes = 1kB

10^6 bytes = 1MB

Representing the Bit

booleans: on/off, 0/1, true/false

Representing the Bit

booleans: on/off, 0/1, true/false

Coin, magnets

Representing the Bit

booleans: on/off, 0/1, true/false

Coin, magnets

Switches

The Transistor

Computer circuits

Made of semiconductors

Made of semiconductors

Function as electronic switches

Made of semiconductors

Function as electronic switches

Billions on modern chips

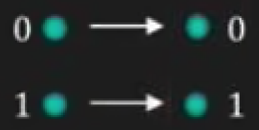
Logic gates

Computing with switches

4 single-bit gates

Identity

$$f(x) = x$$

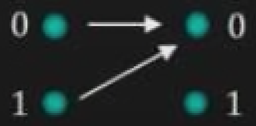


Negation

$$f(x) = \neg x$$

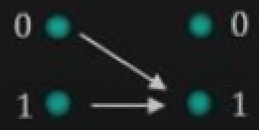
Constant-0

$$f(x) = 0$$



Constant-1

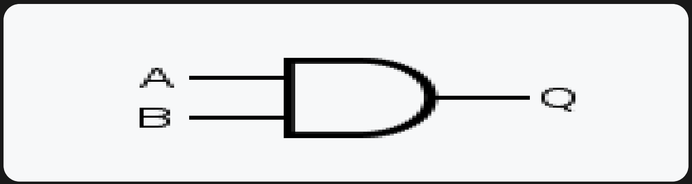
$$f(x) = 1$$



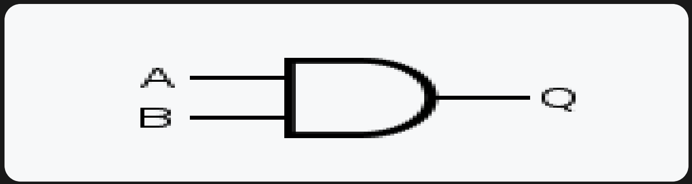
Multi-bit gates

Understand, don't memorize

AND gate



AND gate



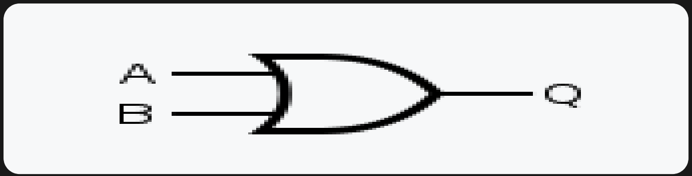
0 0 → 1

0 1 → 0

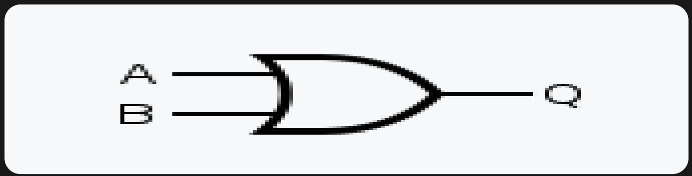
1 0 → 0

1 1 → 1

OR gate



OR gate



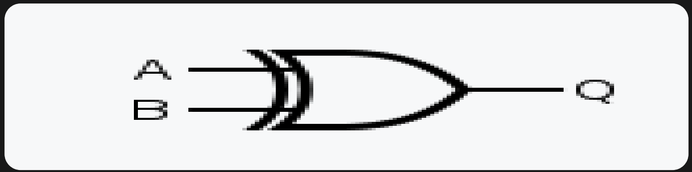
0 0 → 0

0 1 → 1

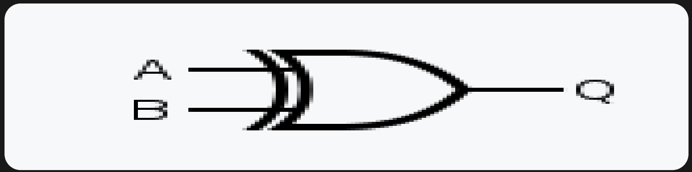
1 0 → 1

1 1 → 1

XOR gate



XOR gate



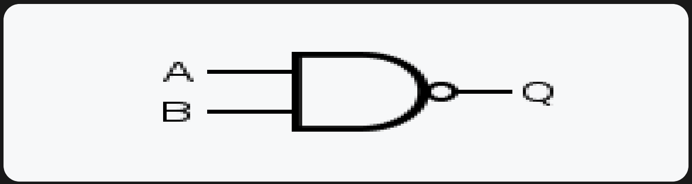
0 0 → 0

0 1 → 1

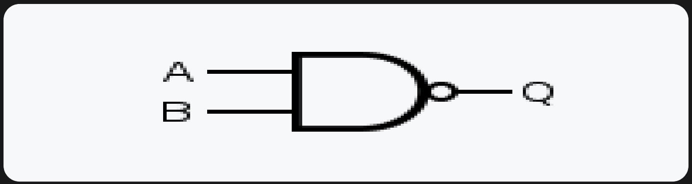
1 0 → 1

1 1 → 0

NAND gate



NAND gate



0 0 → 1

0 1 → 1

1 0 → 1

1 1 → 0

Exercises

(0 OR 1) AND 1

(0 OR 1) AND 1

NOT((1 XOR 1) OR 0)

(0 OR 1) AND 1

NOT((1 XOR 1) OR 0)

0 NAND 0

0 XOR (0 NAND 1)

0 XOR (0 NAND 1)

1 OR (1 AND XOR(0 OR (0 NAND NOT(1))))

Programming Exercise

Write a function

`myFunc(s: str) -> bool`

that computes a running XOR

```
myFunc("0101")
```

```
myFunc("0101")
```

```
myFunc("0101")
```

0 XOR 1 → 1

```
myFunc("0101")
```

0 XOR 1 → 1

101

```
myFunc("101")
```

```
myFunc("101")
```

1 XOR 0 → 1

```
myFunc("101")
```

1 XOR 0 → 1

11

```
myFunc("11")
```

```
myFunc("11")
```

1 XOR 1 → 0

```
myFunc("11")
```

1 XOR 1 → 0

0

Sample Solution

```python
def myFunc(s) -> bool: # recursive implementation
  a, b = s[0] == '1', s[1] == '1'

  if len(s) > 2:
    temp = '1' if (a ^ b) else '0' # perform xor
    s = list(s[1:len(s)]) # XOR result -> 0th string item
    s[0] = temp
    return myFunc(''.join(s))
  else: # base case
    return '1' if (a ^ b) else '0'
```

Challenge: Solve XOR iteratively

Next time: Quantum Mechanics & Quantum Computing

Questions?

Thank you!

References

- https://en.wikipedia.org/wiki/Bit
- https://en.wikipedia.org/wiki/Transistor
- https://www.youtube.com/watch?v=F_Riqjdh2oM
- https://en.wikipedia.org/wiki/Logic_gate