Binary Number System

Reese Hatfield

Introduction

- Computer Science
  - What are we actually studying?

- Determinism
  - Machines given instruction

Introduction

- Data as input
- Do some operation
- Perform some output
    - How would you represent this data?

Introduction

- English
  - "Do my homework"
- Specific?
  - "Write an english essay about Shakespeare"
- How would you make a computer that does this?

Introduction

- Seriously
    - How would you take a bunch of wires and power and create that machine?
- Start smaller
- How would you represent this data?

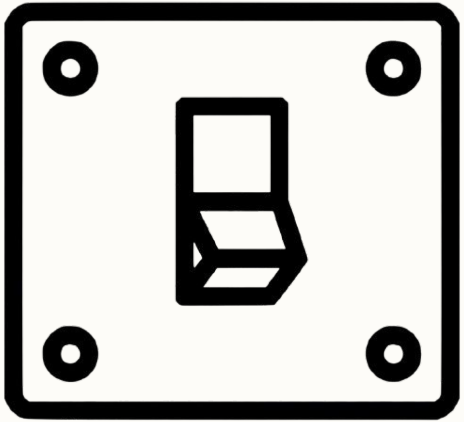
Introduction

- Look a wire
- Any wire
- Can I touch it?

- Well it depends
- On what?

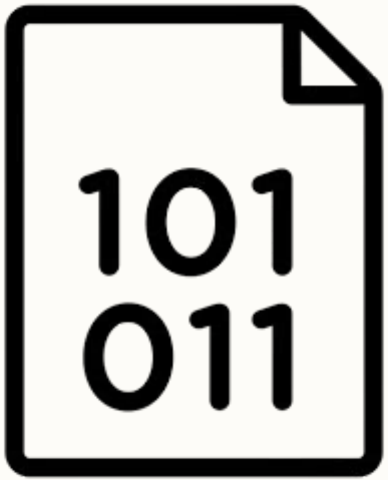
Introduction

- Will it shock me?
- If its powered, yes it will
- States
  - Powered (on)
  - Un-Power (off)
- Simplest representation



Introduction

- This is how we get binary
  - 0s and 1s
- People have studied number systems for a long time
- George Boole
  - Early 1800s
  - Boolean algebra
  - “Math with only 1s and 0s”



Introduction

- Claude Shannon
    - 1937
    - “Electricity is just 1s and 0s”
- Basis of computer
- We work in “higher level abstraction”
- E.g. base 10
- How to convert?

Introduction

- How do we do math with a computer?
- We need a conversion

- Take a number (base 10)
- Convert it to base 2
- This is hard ⇒ start w/ easier problems

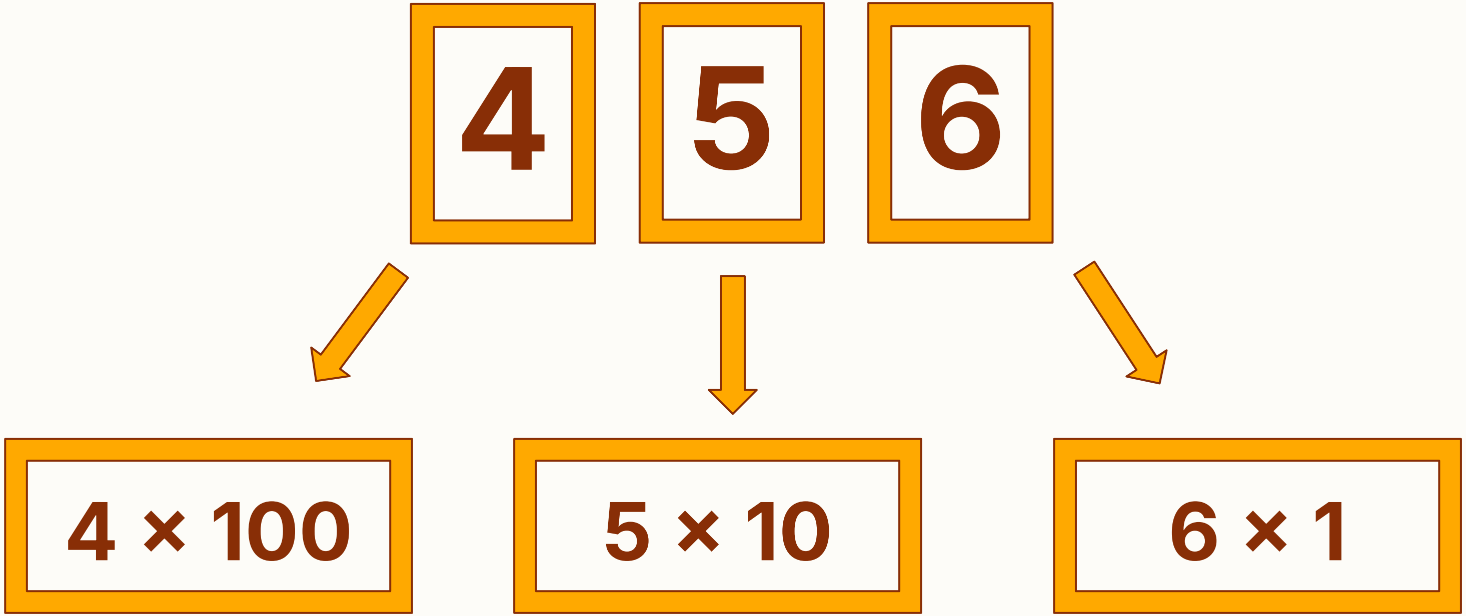
Introduction

- What does each piece of a base 10 number mean?

- Take 456

4 5 6

Introduction



Introduction

- How could we represent the same number (or really amount)
- With only 1s and 0s

4 5 6

Introduction

- Hard for a few reasons
- We don’t have digits 2-9

- We can’t “multiply by 10”

Introduction

4 5 6

4 × 10^2 + 5 × 10^1 + 6 × 10^0

Introduction

- We’re secretly using exponents
- “Base 10”
- So we use 10 as a base
  - In Base 10

4	5	6

Introduction

- Before we can convert
- Let's look at what a binary number "would" look like
- What is this number?

1	0	1	0	0	1

Introduction

- Each “decimal” place
- Has a specific meaning

- 2^place

1	0	1	0	0	1

Introduction

1	0	1	0	0	1
2^5	2^4	2^3	2^2	2^1	2^0

Introduction

- We need to know the powers of two
- This might feel familiar
  - These numbers often "feel" right"
  - Especially when working in a computer context

POWERS OF 2

2^1	=	2
2^2	=	4
2^3	=	8
2^4	=	16
2^5	=	32
2^6	=	64
2^7	=	128
2^8	=	256
2^9	=	512
2^{10}	=	1024

Introduction

1	0	1	0	0	1
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1

Introduction

- We've done something important
- In base 10, we multiplied the value by the place

$$\boxed{4}\ \boxed{5}\ \boxed{6}$$

$$\boxed{4 \times 10^2} + \boxed{5 \times 10^1} + \boxed{6 \times 10^0}$$

1	0	1	0	0	1
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1 × 32	0 x 16	1 × 8	0 x 4	0 x 2	1 × 1

1	0	1	0	0	1
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1

1×32 + 1×8 + 1×1 = **41**

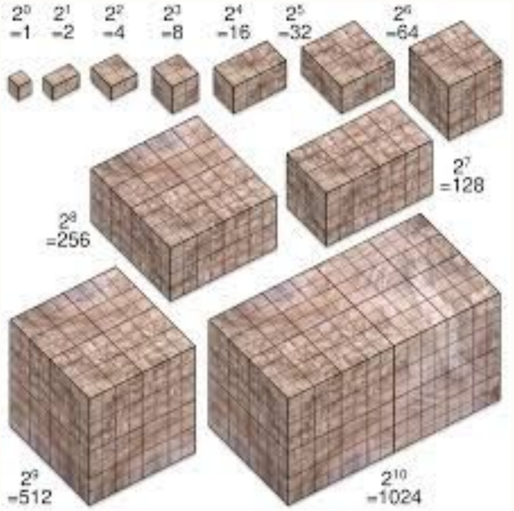
0×16

0×4

0×2

Introduction

- How does this look at scale?
  - Weirdly efficient
  - Despite losing 8 digits
- If we start to add more and more numbers
  - What does this look like



Binary

- How does addition work here?

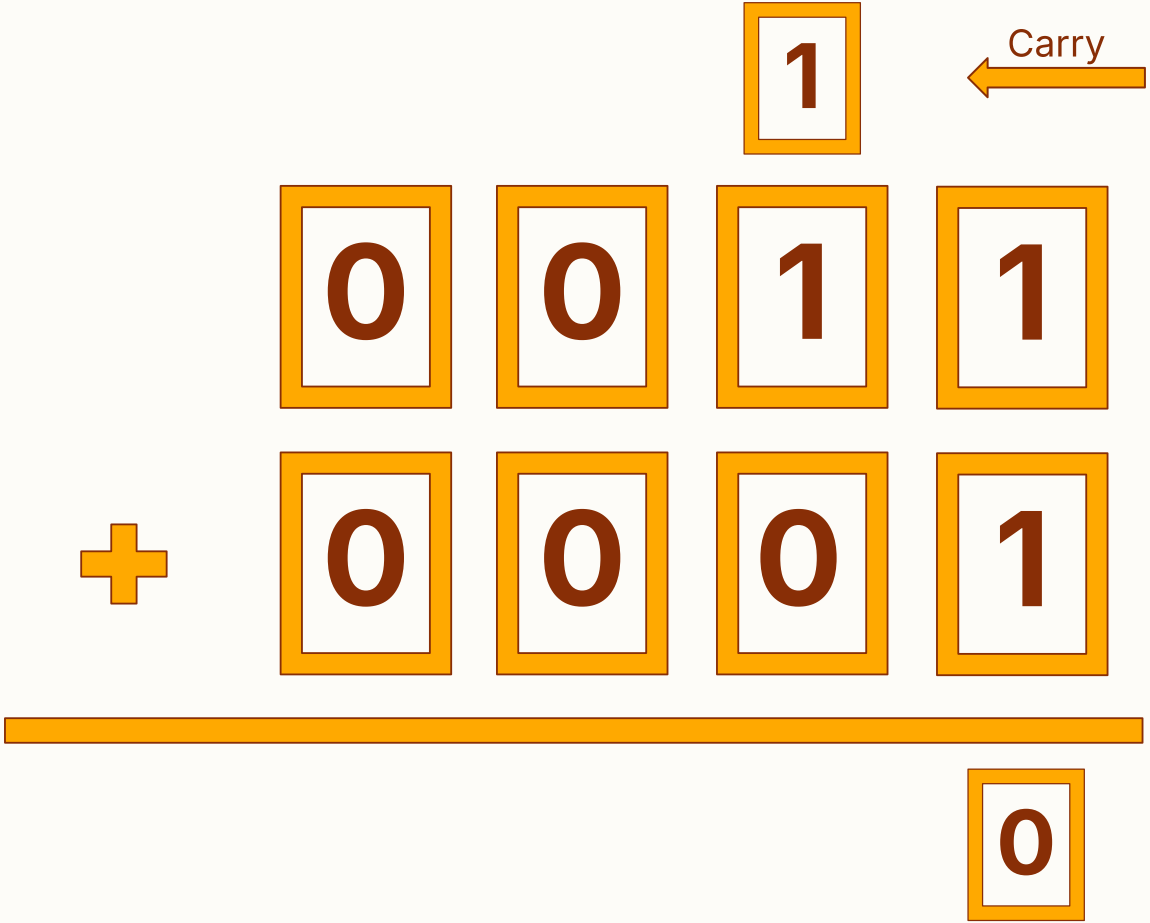
$$\begin{array}{r} 0011 \\ +\ \ 0001 \\ \hline \end{array}$$

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
...	...

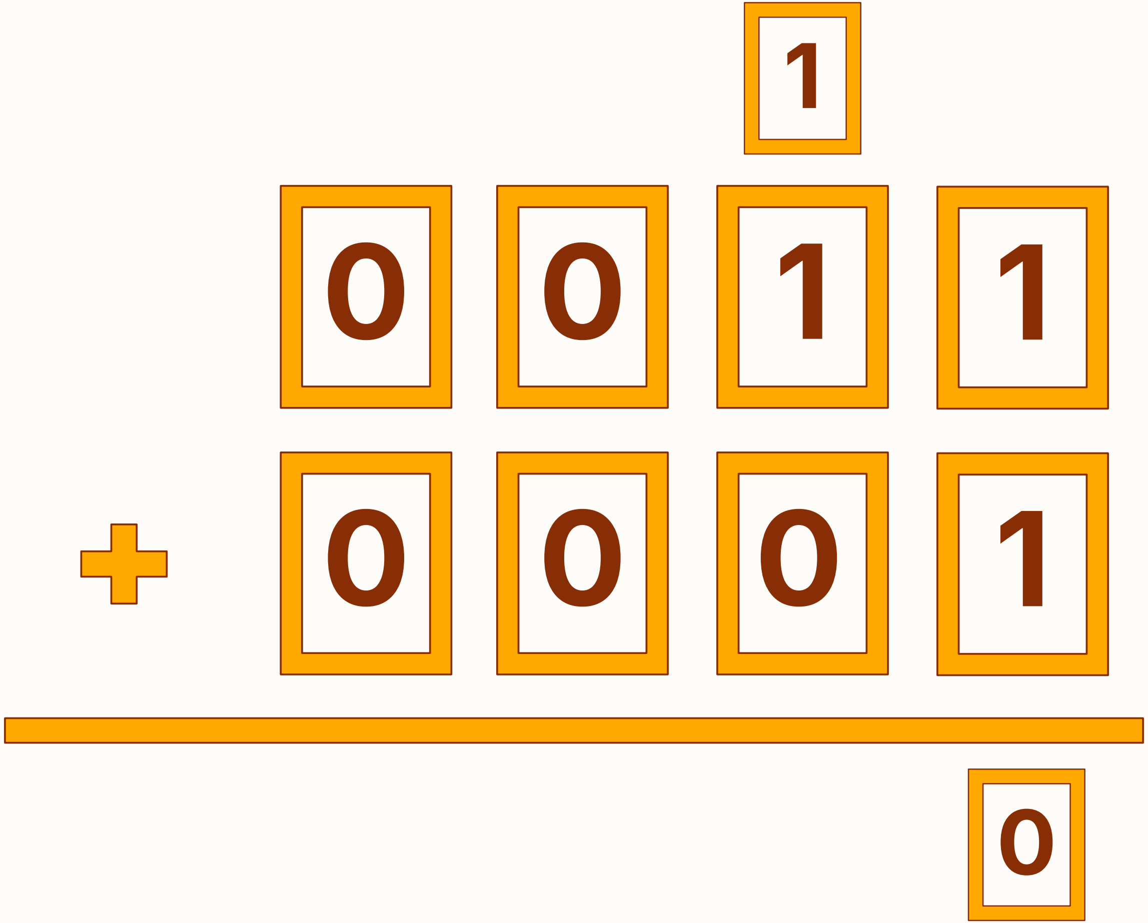
- What is 1 + 1 ?

	0	0	1	1
+	0	0	0	1

- What is 1 + 1 ?
- 10



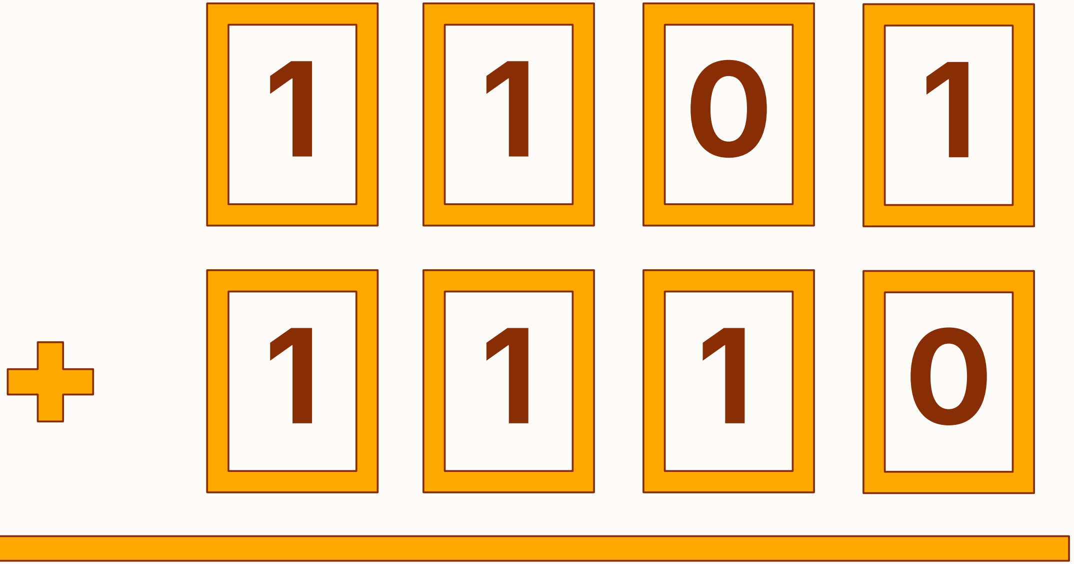
- What is 1 + 1 ?
- 10
- 1 + 1 + 0



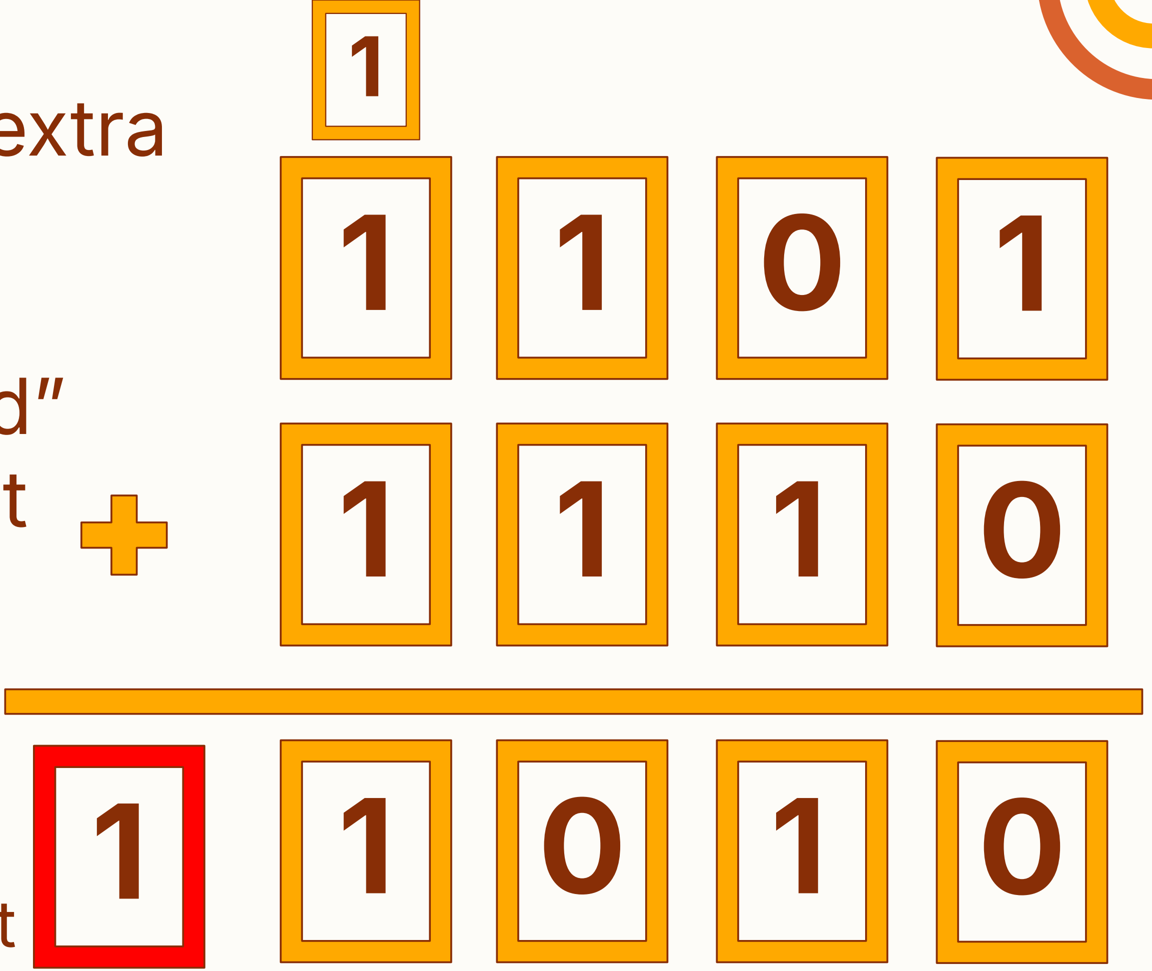
- What is 1 + 1 ?
- 10
- 1 + 1 + 0

$$
\begin{array}{ccccc}
 & & 1 & 1 & \\
 & 0 & 0 & 1 & 1 \\
+ & 0 & 0 & 0 & 1 \\
\hline
 & 0 & 1 & 0 & 0
\end{array}
$$

- Another example
- Whiteboard

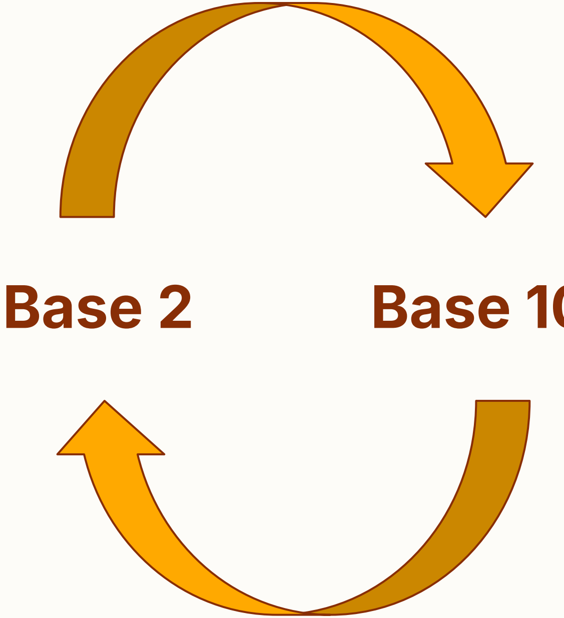


- We had an extra digit
- "Overflowed" into the next place
- Many devices cannot handle this ⇒ incorrect results



Binary

- But we wanna go the other way!
- Base 10 → Base 2
- Each digit "doubles" its potential value every place
- Do the opposite



Binary

- While quotient is not 0
  - Divide the number by 2
  - Note its remainder
    - Will be either 1 or 0
  - This becomes your rightmost digit
  - Repeat, updating original number

- Algorithm
  - Series of steps
  - ~Pseudocode

- Lets try an example: 293

293 / 2	⇛ 146 r1
146 / 2	⇛ 73 r0
73 / 2	⇛ 36 r1
36 / 2	⇛ 18 r0
18 / 2	⇛ 9 r0
9 / 2	⇛ 4 r1
4 / 2	⇛ 2 r0
2 / 2	⇛ 1 r0
1 / 2	⇛ **0** r1

- Lets try an example: 293

Least Significant bit

293 / 2	⇒ 146 r1	1
146 / 2	⇒ 73 r0	0
73 / 2	⇒ 36 r1	1
36 / 2	⇒ 18 r0	0
18 / 2	⇒ 9 r0	0
9 / 2	⇒ 4 r1	1
4 / 2	⇒ 2 r0	0
2 / 2	⇒ 1 r0	0
1 / 2	⇒ **0** r1	1

• • •

Continue until quotient is 0

Most Significant bit

Binary

Most Significant Bit

$$1\,0\,0\,1\,0\,0\,1\,0\,1_2 = 293_{10}$$

Least Significan
Bit

Binary

- Can we do complex binary addition?
- Two methods
  - Yes ⇒ line up digits
  - Yes ⇒ Convert, add convert

- What about other bases?