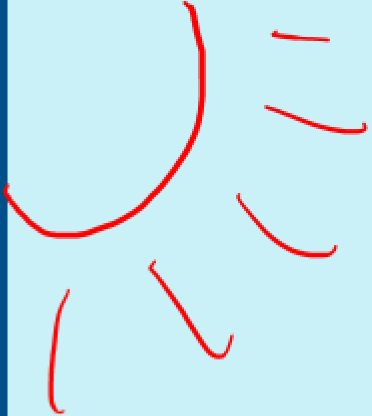
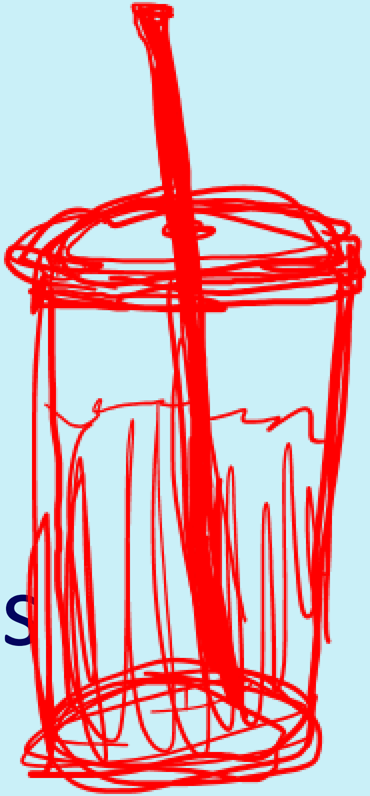


CS 340

Bits, Bytes, and Nybbles



Learning Goals

- Be able to understand and convert between binary, hexadecimal, and decimal values.
- Understand how things can be represented with bits.
    - Numbers
    - Really big numbers
    - Negative numbers
    - Characters (if we get to it)

Plan for Today

- Slides + Clickers
- Time to work on homework and ask staff questions
- Slides + Clickers
- Time to work on homework and ask staff questions

To get the 0.1% extra credit, answer a majority of the clicker questions

Last time...

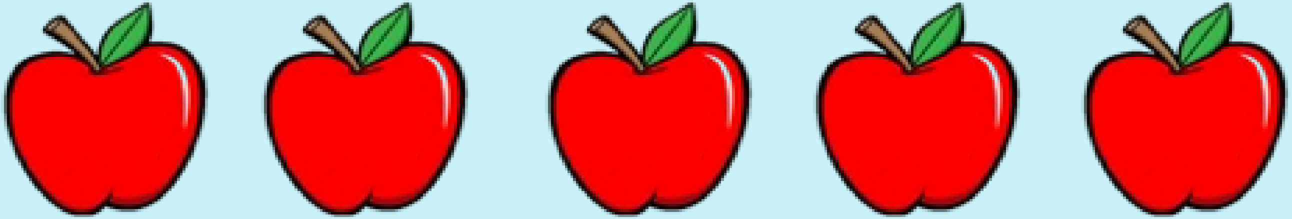
- The transistors we use can be in two states

1, 0

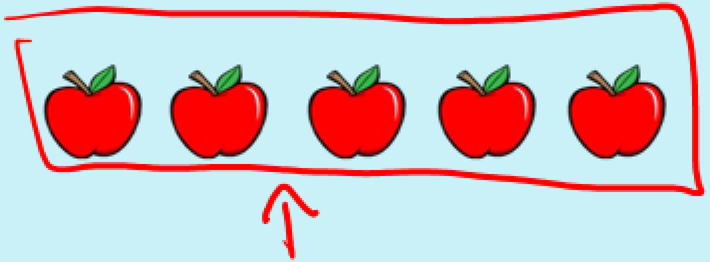
- We can do operations on 1’s and 0’s (AND, OR)
- We can store 1’s and 0’s

But what kind of things can we represent with just a 1 or 0?

How do we represent something like…



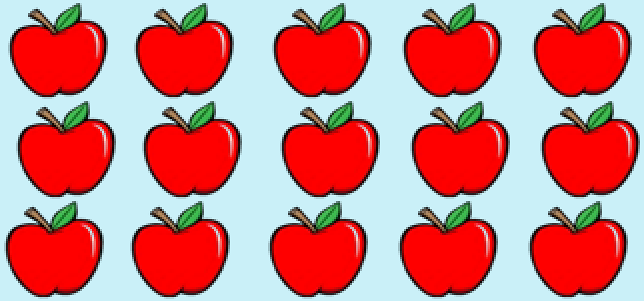
Decimal (base 10)



5

0 1 2 3 4 5 6 7 8 9

Decimal (base 10)



555

$5+5+5=15$

$\underline{5}\,\underline{2}\,\underline{4}$

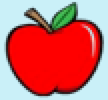
$5\cdot10^{2}+2\cdot10^{1}+4\cdot10^{0}=500+20-4$

Ways to Represent Numbers

- Decimal (base 10)
  - Digits : 0 1 2 3 4 5 6 7 8 9
- Binary (base 2)
  - Bits : 0 1
  - Header: 0b
- Hexadecimal (base 16)
  - Nybbles : 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - Header: 0x

Binary (base 2)

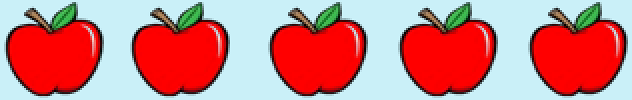
0, 1



1

Binary (base 2)

0, 1



632

110100110

0 1 0 1

$2^3=8$ $2^2=4$ $2^1=2$ $2^0=1$

2 4 8 16 32 64 128 256 512...

What is the following in decimal?

clicker.cs.illinois.edu

Q1

~Code~

340

1024 512 256 128 64 32 16 8 4 2 1

~~0b~~10000000000

Ob

10-03

=1024

What is the biggest number we could represent with 6 bits?

clicker.cs.illinois.edu

Q2

~Code~
340

$\boxed{111111} = 63$

$+\ 00000\boxed{1} = 1$

64

$\boxed{1000000}$

$= 64$

$0b\ \underset{32}{\underline{1}}\ \underset{16}{\underline{1}}\ \underset{8}{\underline{1}}\ \underset{4}{\underline{1}}\ \underset{2}{\underline{1}}\ \underset{1}{\underline{1}} = 63$

$\boxed{8+4+2+1}$

$\underline{9}\ \underline{9}\ \underline{9}$

Which number is bigger?

32 16 8 4 2 1 = 31

0b011111 vs. 32

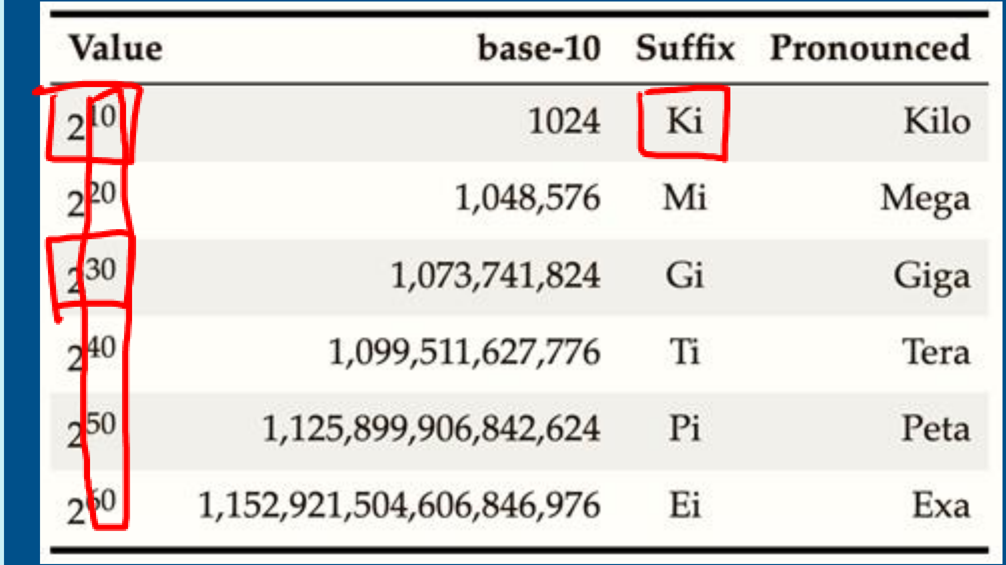
16+8+4+2+1

clicker.cs.illinois.edu

Q3

~Code~
340

Big Numbers in Binary



Value	base-10	Suffix	Pronounced
2^{10}	1024	Ki	Kilo
2^{20}	1,048,576	Mi	Mega
2^{30}	1,073,741,824	Gi	Giga
2^{40}	1,099,511,627,776	Ti	Tera
2^{50}	1,125,899,906,842,624	Pi	Peta
2^{60}	1,152,921,504,606,846,976	Ei	Exa

1 Ki

$$2^{27} = 2^{20} \cdot 2^{7} =$$

128Mi

Translate the following…

0 1 2 3 4 5

1 2 4 8 16 32

clicker.cs.illinois.edu

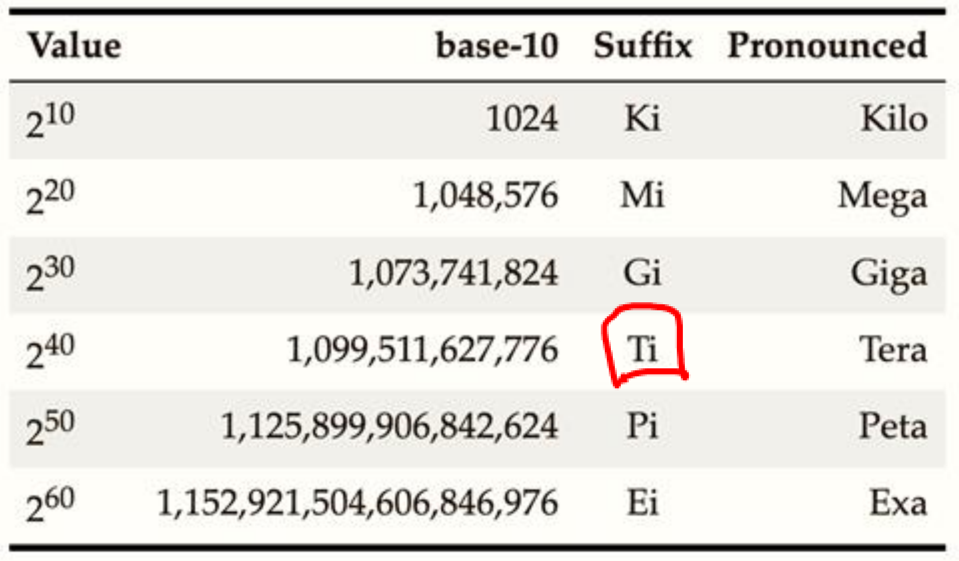
Q4

~Code~
340

2^(45) $= 2^{40} \cdot \boxed{2^5}$

74903,951 X

32Ti



Value	base-10	Suffix	Pronounced
2^{10}	1024	Ki	Kilo
2^{20}	1,048,576	Mi	Mega
2^{30}	1,073,741,824	Gi	Giga
2^{40}	1,099,511,627,776	Ti	Tera
2^{50}	1,125,899,906,842,624	Pi	Peta
2^{60}	1,152,921,504,606,846,976	Ei	Exa

1 Byte = 8 Bits

Your computer finds it useful to think in terms of bytes.

Common sizes of types

- Char - 1 byte
- Int - 4 bytes

A char is 1 byte, what is the biggest value a char can store?

clicker.cs.illinois.edu

Q5

~Code~
340

678,913,421

1	1	1	1	1	1	1	1	1
256	128	64	32	16	8	4	2	1

255

1111 1111

What is the biggest value 4 bits can hold?

clicker.cs.illinois.edu

Q6

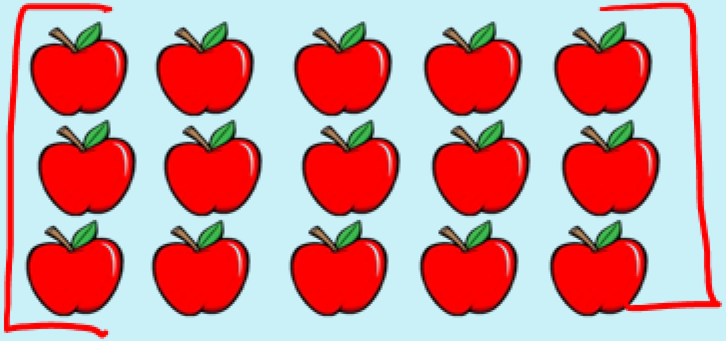
~Code~
340

$$\begin{array}{ccccc} & 1 & 1 & 1 & 1 \\ \downarrow & \overline{8} & \overline{4} & \overline{2} & \overline{1} \\ 16 & & & & \end{array}$$

0, 1, 2, 3 ... 15

= 15

Hexadecimal (base 16)



0xF

$0, 1, 2 \ldots 8, 9, \overset{10}{A}, \overset{11}{B}, \overset{12}{C}, \overset{13}{D}, \overset{14}{E}, \overset{15}{F}$

Big Idea

- 4 bits can always be translated to 1 hexadecimal nybble

_ _ _ _

0110

8 4 2 1

4+2=6

0x6

10 11 12 13 14

A B C D E

1110 8+4+2=14 =0x~~14~~ E

How many nybbles are needed to represent 1 byte?

clicker.cs.illinois.edu

Q7

~Code~
340

1 byte = 8 bits

2

0xF1

Ways to Represent Numbers

- Decimal (base 10)
  - Digits : 0 1 2 3 4 5 6 7 8 9
- Binary (base 2)
  - Bits : 0 1
  - Header: 0b
- Hexadecimal (base 16)
  - Nybbles : 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - Header: 0x

Big Ideas

- 8 bits = 1 byte
- 1 byte = 2 nybbles

Talking in bits can be hard to parse for humans… hex is easier!

0b 1011 1001 1111 0111 0101

0xB9F75

↑ ↑ $16^0 \cdot 5$

How many bytes do you need to store this value?

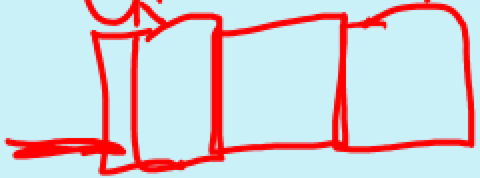
clicker.cs.illinois.edu

Q8

~Code~
340

0xB9F75

1 byte



3 bytes

1 byte = 8 bits

0567

Conversions

1,372

1. Binary =
2. Hexadecimal =

0x55C

1 2 4 8 16 32 64 128 256 512 1024 2048

$$\begin{array}{r} 1372 \\ -\ 1024 \\ \hline 348 \end{array} \quad \begin{array}{r} 348 \\ -\ 256 \\ \hline 92 \end{array} \quad \begin{array}{r} 92 \\ -\ 64 \\ \hline 28 \end{array} \quad \begin{array}{r} 28 \\ -\ 16 \\ \hline 12 \end{array} \quad \begin{array}{r} 12 \\ -\ 8 \\ \hline 4 \end{array}$$

5

0 1 0 1   0 1 0 1   1 1 0 0

~~1024 512 256 128 64 32 16~~ 8 4 2

10 mins to work on HW:

Zone 1. Base conversions

Negative/Signed Values

Also bit addition!

- Idea: 2’s complement

0111 = 7

4 2 1

~~1000 +1~~

1001 = 9

8 4 2 1

450

+ 550

1000

- Flip bits
- add 1

1 1 1

0111

+ 1001

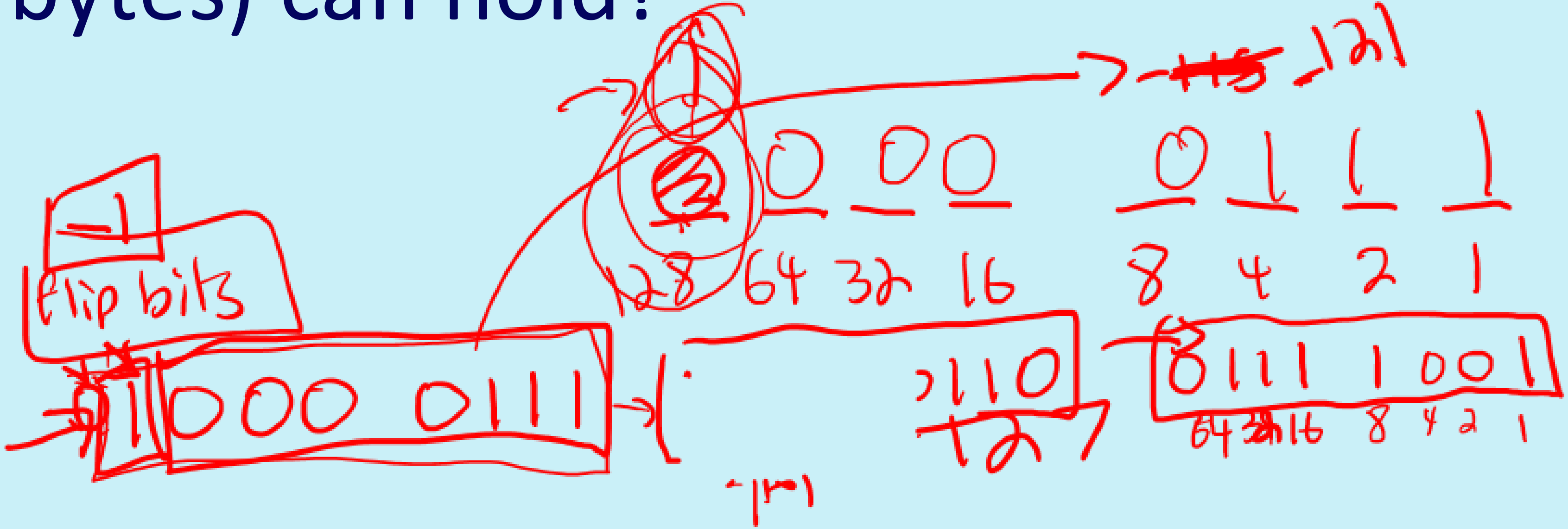
10000

What is the biggest positive number a signed char(1 bytes) can hold?

clicker.cs.illinois.edu

Q9

~Code~
340



How to represent a character

Representing things besides numbers…

- Char
- Int

1000 1101 = 276

1000 1100

0111 0011 = 216

→216

UTF-8

Representing things besides numbers…

Code Point - 4 byte value that maps to a symbol

UTF-8 - A character encoding for code points

0100 000

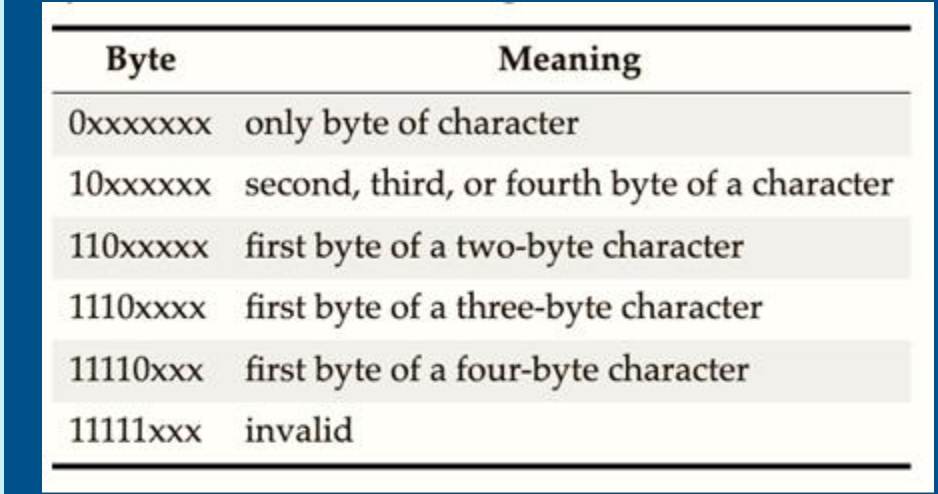
-32

→ 1100 000 → subtract 1 bit

~~0~~ 1011 111

UTF-8

111010
111001



Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

111000

110111

001000 = -8
8 4 2 1

What is coming up

- Finish HW (due before class next Thursday (12:30pm))
- Work on MP 1 (due 11:59pm Tuesday)
- Read website text for more details and information
- Tuesday’s Topic: Assembly and UTF-8