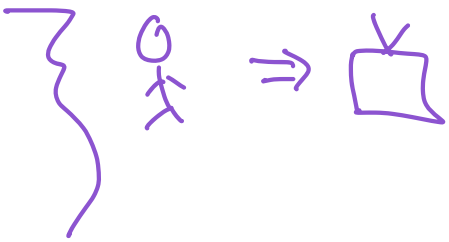
CS1800
9/12 - Tues.

Admin

- notes on course website
- recitations start this week → Quiz due F/18
  do it during recitation :)
- Qs during lecture → Piazza Live Q+A

Agenda

1. Representation of Numbers
2. Converting Between Bases
3. Arithmetic in other bases



1. Representation of Numbers

Universe: unsigned (just for today)

- computers: 0/1
  - What can we possibly solve?
    - lots! → start: True, False
      - 0 off
      - 1 on
- How do computers rep. numbers?
  - decimal, base 10: 0, 1, 2, ..., 9
  - binary, base 2: 0, 1 — bit = = binary digit
- numbers

...	10^4	10^3	10^2	10^1	10^0
			2	3	4
	10000	1000	100	10	1

expansion: $2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$

$= 2 \cdot 100 + 3 \cdot 10 + 4 \cdot 1$

- numbers (binary)

...	2^4	2^3	2^2	2^1	2^0 (base 2)
	16_{10}	8_{10}	4_{10}	2_{10}	1_{10}
	10000_2	1000_2	100_2	10_2	1_2

number in binary

1	1	0	1
2^3	2^2	2^1	2^0
8	4	2	1

expansion: $1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 13$

$$1101_2 = 13_{10}$$

-  compromise: Base 16 (hexadecimal)

Hex digits: $0, 1, 2, \ldots, 9, A, B, C, D, E, F$

ex $8A4_{16} = \_\_\_\__{10}$

$8 \cdot 16^2 + 10 \cdot 16^1 + 4 \cdot 16^0 = 2212_{10}$

- hex: can rep bigger values with fewer digits

In general $n_b = d_k \cdot b^k + d_{k-1} \cdot b^{k-1} + \ldots d_0 \cdot b^0$

2. Converting Between Bases

(non decimal?)

Base 2 → decimal ✓

Base 16 → decimal ✓

Decimal → another base

$\_\_\__{10} = \_\_\__2$?

$= \_\_\__{16}$?

How: Euclid

$n = p \cdot q + r$

(ex) $13_{10} = 10 \cdot \underline{1} + \underline{3}$ → already knew it.

$13_{10} = \underline{1} \cdot 2^3 + \underline{1} \cdot 2^2 + \underline{0} \cdot 2^1 + \underline{1} \cdot 2^0$

1101_2

Euclid's Division

$n_{10} = \_\_\__b$

In decimal... $n = p \cdot q + r$

- q: quotient
- p: base
- r: remainder, $r < \text{base}$

- r is part of the answer (coefficients)
- take q and repeat the process

(ex) decimal → decimal

$n = pq + r$

$1234_{10} = \_\_\_\__{10}$?

(1) $1234 = 10 \cdot 123 + 4$ (p, q, r)

(2) $123 = 10 \cdot 12 + 3$ (p, q, r)

(3) $12 = 10 \cdot 1 + 2$ (p, q, r)

(4) $1 = 10 \cdot 0 + 1$ (r)

remainders bottom to top

1234_{10}

$784_{10} = \_\_\__8$?

1420_8

$784 = 8 \cdot \underset{q}{98} + \underset{r}{0}$

$98 = 8 \cdot 12 + 2$

$12 = 8 \cdot 1 + 4$

$1 = 8 \cdot 0 + 1$ → 1420

$784_{10} = \_\_\__{16}$?

310_{16}

$784 = 16 \cdot \underset{q}{49} + \underset{r}{0}$

$49 = 16 \cdot 3 + 1$

$3 = 16 \cdot 0 + 3$ → 310

$310_{16} = \_\_\__{10}$?

$3 \cdot 16^2 + 1 \cdot 16^1 + 0 \cdot 16^0$

$= 3 \cdot 256 + 1 \cdot 16$

$= 784_{10}$

$1420_8 = \_\_\__{10}$?

$1 \cdot 8^3 + 4 \cdot 8^2 + 2 \cdot 8^1 + 0 \cdot 8^0$

$= 784_{10}$

3. Arithmetic In Other Bases

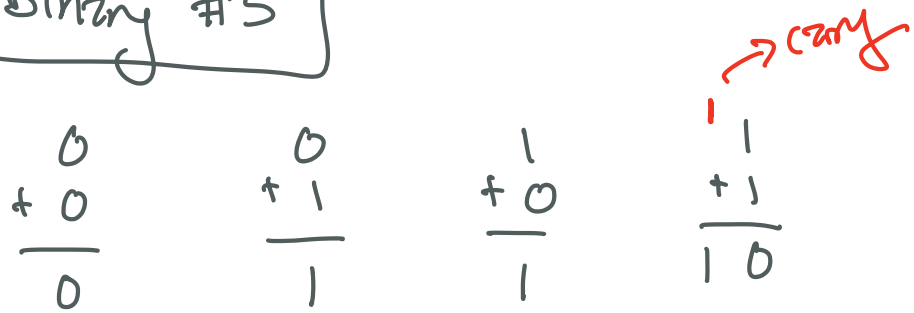
= What we do in decimal kind of works =

Binary

- Converted to base 2
- need to do math

What is the biggest value we can store in one bit?

Adding Binary #s



Convert to decimal

$$\begin{array}{r} 37 \\ +\ 21 \\ \hline 58 \end{array}$$

$$\begin{array}{r} 1\ 0\ 0\ 1\ 0\ 1 \\ +\quad 1\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 0\ 1\ 0 \end{array}$$

Sanity check!

$111010_2 = \_\_\_\__{10}$?

$1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1$

$32 + 16 + 8 + 2 = \boxed{58}$:)

multiplication

$$
\begin{array}{r}
92 \\
\times\ 23 \\
\hline
276 \\
+\ 184 \\
\hline
2116
\end{array}
$$

Binary multiplication is the same!

$$
\begin{array}{cccc}
0 & 0 & 1 & 1 \\
\times\ 0 & \times\ 1 & \times\ 0 & \times\ 1 \\
\hline
0 & 0 & 0 & 1
\end{array}
$$

Binary multiplication

$$
\begin{array}{r}
1011 \\
\times\ 0101 \\
\hline
1011 \\
0000 \\
+\ 1011 \\
\hline
110111
\end{array}
$$

Convert each number to dec

Sanity check:

11_{10}

5_{10}

$5 \times 11 = 55_{10}$

110111

$32 + 16 + 4 + 2 + 1 = 55$!!

1.8+

1 1 0 1

8 4 2 1

$\Rightarrow 1\cdot4 + 0\cdot2 + 1\cdot1 = 13$

084