CS1800
9/23 Tues.

Admin
- HW1 due 9pm
- HW2 out now, 9/30 9pm

Agenda
1. Two's complement math
2. Overflow
3. Gates + Circuits

0. Review — 4 bit two's complement

what bit is positive? 0

converting:
- 3_{10} 0011
- 5_{10} 0101
- 6_{10} 0110 → positive

zadding:
- 1101 +1 → 1110
- 1010 +1 → 1011
- 1001 +1 → 1010 → negative

1. Two's Complement Math

↳ signed numbers (pos, neg)
fixed # bits per number

Convert from decimal-binary, binary-decimal
- positive: just convert (pad if necessary)
- negative: take the complement, zad one

$-5_{10} = 1011$ [4 bit two's comp]
- take pos value
- complement
- zad one

$$\begin{array}{r} 0101 \quad (+5) \\ 1010 \quad (\text{comp}) \\ +1 \\ \hline 1011 \end{array}$$

$-6_{10} = 1010$
- take pos value
- complement
- zad one

$$\begin{array}{r} 0110 \quad (+6) \\ 1001 \quad (\text{comp}) \\ +1 \\ \hline 1010 \end{array}$$

binary to decimal

$1101_2 = \_\_\_\__{10}$? -3_{10}

Leftmost bit is sign bit — 0 pos, 1 neg

$$\begin{array}{r} 0010 \quad (\text{comp}) \\ +1 \\ \hline 0011 \end{array} \rightsquigarrow \text{pos value } (3)$$

Subtraction is addition — $x - y$, $x + -y$

practice:
- convert from decimal to binary first
- do the arithmetic
- chop off extra bit if/a

$-3-2$

$-3_{10} = 1101$
$-2_{10} = 1110$ +
1011 ⇝ -5_{10} :)

$3-6 = 1101_2$

$3_{10} = 0011$
$-6_{10} = 1010$ +
1101 ⇝ -3_{10} :) sanity check!

↑ chop off! ↳ sign bit

2. overflow in two's comp

↳ arithmetic operation gives a result that is too high (pos) or too low (neg) for our number of bits

can't just do the math and see if it's too high/low
↳ how to detect overflow: look for unexpected sign bit
- add two pos and get neg —
- add two neg and get pos —

ex) 4 bit two's comp
Highest pos: 0111 (7_{10})
lowest neg: 1000 (-8_{10})
xty and get >7 or <-8
that is overflow!

ex) 4 bit two's comp.
$-3-6$ $\quad -3_{10} = 1101$
$-6_{10} = 1010 +$
$\overline{\times 0111}$
chop off ~ 0111

practice 6-bit two's complement
- stay in binary
- do the arithmetic
- is it overflow?

① 111101 (-11_{10})
+ 011111 (31_{10})
$\overline{\times 010100}$ (20_{10}) :)
no overflow

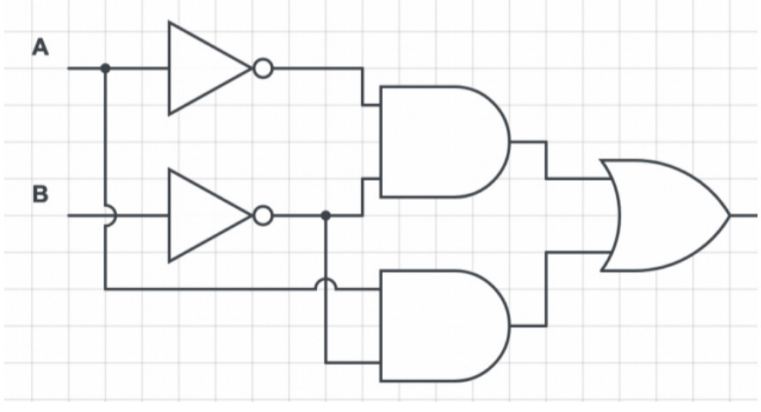
② 101111 (-17_{10})
+ 100111 (-25_{10})
$\overline{\times 010110}$ (22_{10})
↳ overflow :(

high: 31
low: -32

3. Circuits + gates

▷ not ▷ and ▷ or

- slides: English → truth table → logic → circuit
- practice now: circuit → logic statement → simplify → apply laws of equivalence
  simpler so much better!
  fewer gates → cheaper faster !!



① Circuit to logic
$(\neg A \wedge \neg B) \vee (A \wedge \neg B)$

② Simplify the logic
$(\neg B \wedge \neg A) \vee (\neg B \wedge A)$ comm.

$\neg B \wedge (\neg A \vee A)$ distro

$\neg B \wedge T$ Negation

$\neg B$ identity

(3) Better circuit!