CS 5002

Jan 24th - week three :)

Admin

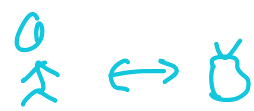
- Hw1 was due 6pm
- Hw2 out, due 1/31 6pm
- ICA #3 due 1/25 6pm

Agenda

1. Gates & Circuits
2. Representing numbers
3. Converting numbers

ICA

4. Signed numbers



$\boxed{\begin{array}{l}\text{0. Something we learned} \\ \text{Something we have Qs about}\end{array}}$

implication $P \Rightarrow Q$

$P \Rightarrow Q$ vs. $P \wedge Q$??

- draw both truth tables
- which one better respects the original statement

predicate is a generalization of logic statement

$P(x)$ $S(x)$ $T(x,y)$

predicate -> logic

1. plugin values

2. quantifiers $\forall$ $\exists$

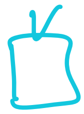
logical equivalence - show 2 logic statements are equivalent by applying laws

I. Gates and Circuits

Logic T/F $\wedge$ $\vee$ $\neg$

1/0

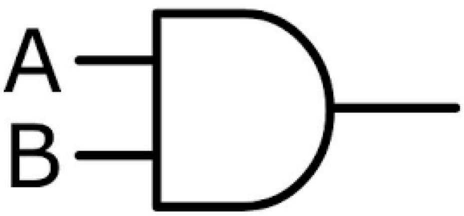
on/off



transistor
on/off

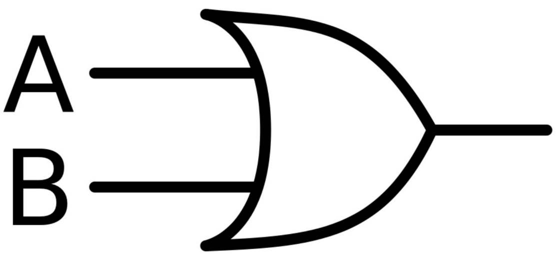
$\boxed{\text{Gate}}$: logical operation on one or more inputs

$\boxed{\text{Circuit}}$: sequence/combination of gates



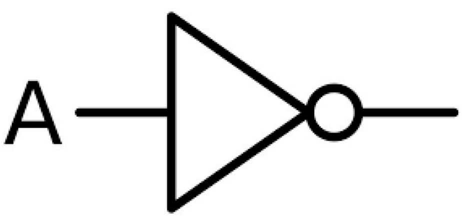
and

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1



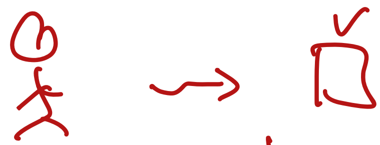
or

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1



not

A	$\neg A$
0	1
1	0

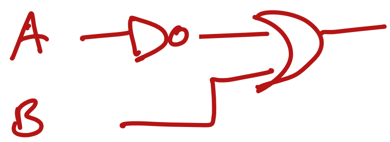


how can we build a circuit to solve a problem?

joinpd.com

lce rkb

$A \Rightarrow B$ $\neg A \vee B$

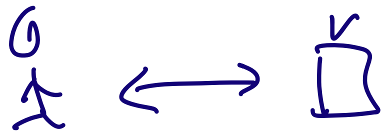


2. Representation of Numbers

62

our universe: unsigned numbers

in number systems – counting, arithmetic



base 10 decimal ↔ base 2 binary + base 16 hexadecimal

Decimal digit	Binary digit (bit)	Hex digit
0, 1, 2, ..., 9	0, 1	0, 1, 2, ..., 9, A, B, C, D, E, F

Decimal (base 10)

2	1	3	8
1000s	100s	10s	1s
10^3	10^2	10^1	10^0

$$= (1000)(2) + (100)(1) + (10)(3) + (1)(8)$$

$$= 2000 + 100 + 30 + 8$$

$$= 2{,}138_{10}$$

In general ...  ... for any base b to decimal, same steps

base 2 → base 10

base 16

Binary

$b=2$

$$\frac{1}{2^4}\quad \frac{0}{2^3}\quad \frac{1}{2^2}\quad \frac{1}{2^1}\quad \frac{0}{2^0}$$

16 8 4 2 1

$$(1)(16) + (0)(8) + (1)(4) + (1)(2) + (0)(1)$$

$$16 + 4 + 2 = 22$$

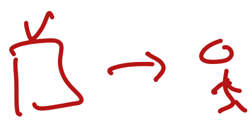
$$10110_2 = 22_{10}$$

Hex

$b=16$

$$\frac{0}{16^4}\quad \frac{0}{16^3}\quad \frac{8}{16^2}\quad \frac{A}{16^1}\quad \frac{4}{16^0}$$

Hex digit: 0, 1, 2, .. 9, A, B, C, D, E, F



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

$$(8)(16^2) + (10)(16^1) + (4)(16^0) = 2212_{10}$$

3. Converting Between Bases

just covered: any base to decimal

now: decimal to any base

Euclid's division algorithm

$$n = p \cdot \underline{q} + \underline{r}$$

quotient remainder

$$22 = 11 \cdot q + r$$

$q = 2, r = 0$

$22 \div 11 = 2 + 0$

$$22 = 7 \cdot q + r$$

$q = 3, r = 1$

$$22 = 17 \cdot q + r$$

$q = 1, r = 5$

Apply Euclid to conversion:

- start with a number in base 10
- divide by target base
- track remainder
- construct the result

ex $13_{10} = \_\_\__{2}$?

target base = 2

remainder possible values:
0, 1, 2..., b-1

$13 \div 2 = 6 \text{ R } 1$

$6 \div 2 = 3 \text{ R } 0$

$3 \div 2 = 1 \text{ R } 1$

$1 \div 2 = 0 \text{ R } 1$

1101_2

$1101_2 = \_\_\__{10}$? sanity check

$(1)(2^0) + (1)(2^2) + (1)(2^3) = 1 + 4 + 8 = 13_{10}$:) yay!

confirmation

$784 = \_\_\__{2}$ (right) 1100010000_2

$784 = \_\_\__{16}$ (left) 310_{16}

$1100010000_2 = \_\_\__{10}$?

$1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^4 = 512 + 256 + 16 = \boxed{784_{10}}$

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

$3 \cdot 16^2 + 1 \cdot 16^1 + 0 \cdot 16^0 = 768 + 16 = \boxed{784_{10}}$

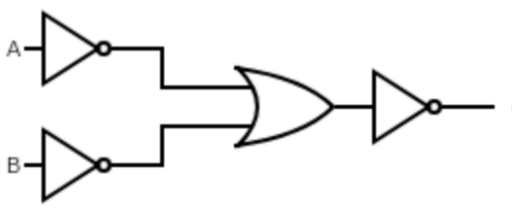
7:30

$$\left.\begin{array}{l} 784 \div 16 = 49 \ R0 \\ 49 \div 16 = 3 \ R1 \\ 3 \div 16 = 0 \ R3 \end{array}\right\} 310_{16}$$

ICA#3

Problem #1

Write out a logical expression for the following circuit (don't simplify!):



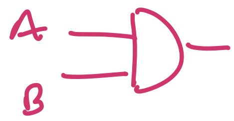
$\neg(\neg A \lor \neg B)$

now... we can simplify

$\neg\neg A \land \neg\neg B$ DeMorg

$A \land B$ Double negation

rewrite the circuit



Problem #2

Convert 29_{10} to unsigned binary.

$29 \div 2 = 14 \text{ R } 1$

$14 \div 2 = 7 \text{ R } 0$

$7 \div 2 = 3 \text{ R } 1$

$3 \div 2 = 1 \text{ R } 1$

$1 \div 2 = 0 \text{ R } 1$

11101_2

3. Signed Numbers

- positive, negative numbers
- how computers do subtraction

$78 - 35$ ~~xxx~~

$78 + -35$

even subtraction of pos #s needs signs

Binary, need both pos/neg values

- Computers have limited space

 78, 1357

 0110110 1001

- every value is repped with same # of bits

real life: # bits 32, 64

our life: choose # bits

- we need to do arithmetic

Result: system for , not intuitive for 

- pos 0
- neg 1

First attempt: sign + magnitude

01101 (+13)

11101 (-13)

Today's Systems: Two's Complement

- pos 0
- neg 1

leftmost bit to be the sign

All values have same # of bits

- start at 0, count up for positive

  count down for negative

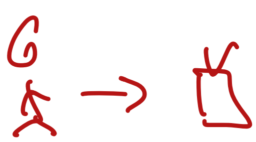
Universe: $\boxed{\text{3-bit two's complement}}$

3 bits to rep every value

leftmost bit is sign bit

two's (sign)	decimal
0 1 1	3
0 1 0	2
0 0 1	1
0 0 0	0
1 1 1	-1
1 1 0	-2
1 0 1	-3
1 0 0	-4

- rep more negative values than positive



Decimal → two's complement

If positive...

- make leftmost bit 0
- convert the number
- pad if needed

If negative

- pretend it's positive, convert to binary
- flip the bits
- add one

Side quest: adding bits

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 10$ . . . carry the one

ex) 6-bit two's complement

$9_{10} = \_\__2 ?$

0	0	1	0	0	1
sign	2^4	2^3	2^2	2^1	2^0

001001_2 in 6-bit two's complement

ex) $-9_{10} = \_\__2 ?$ 6 bit two's complement

pretend positive 001001

flip bits 110110

add one +1

110111 — -9 in 6 bit two's comp.

(ex) $-24_{10} = \_\_\__2$ 6 bit two's comp

- pretend positive 011000
- flip bits 100111

Add one: $0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$ ... 0 with carry of one

flipped bits

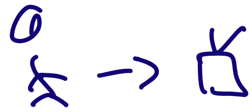
$$
\begin{array}{r}
^{1}^{1}{}^{1}\\
100111\\
+\ 1\\
\hline
101000
\end{array}
$$

~~...~~ (-24)

$29_{10} = \_\_\__2$? 6 bit two's complement

0	1	1	1	0	1
sign	2^4	2^3	2^2	2^1	2^0

011101_2



goals:
same # bits ✓
pos/neg ✓
arithmetic ?

9	001001
-9	110111
-24	101000
29	011101

6 bit universe: what's highest positive #?

011111 = 31_{10}

lowest negative?

100000 = -32_{10}

$\boxed{\text{Arithmetic with binary}}$

9 - 24   9 + -24

$$\begin{array}{rr} 9 & 001001 \\ -24 & +101000 \\ \hline & 110001 \end{array}$$

neg

 (neg)

- flip bits
- add one
- convert (as if pos)

$$\begin{array}{lr} & 110001 \\ \text{flip} & 001110 \\ \text{add} & +1 \\ \hline & 001111 \end{array}$$

001111 $= 15_{10}$

8 4 2 1

-15_{10}

$\boxed{-24 + 29}$

ex $29 - 24$

29	011101
-24	+ 101000
	~~1~~000101

extra bit!! :o

Always chop it off

Final answer: remaining 6 bits

$000101 = 5_{10}$:)

ex $-24 - 9$ ... $-24 + -9$

-24	101000
-9	+ 110111
	~~1~~011111

extra bit :(

011111 6 bits! :(

• added two negatives + got a positive

overflow !!! value is too high (pos) or too low (neg) to fit in my # bits

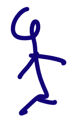
recognize overflow

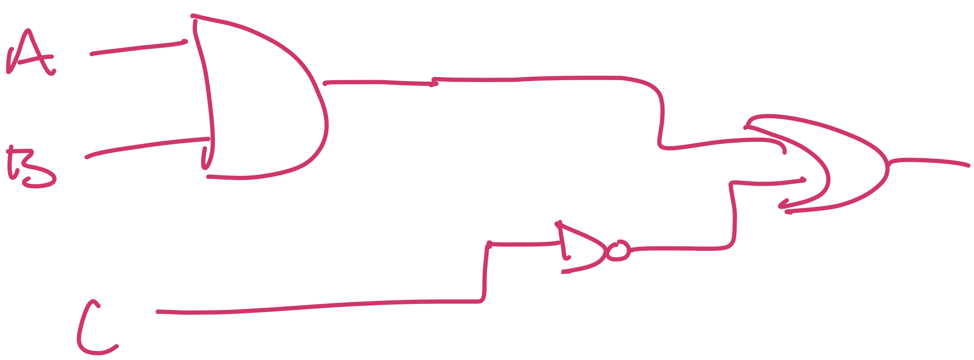
- add two pos, get neg
- add two negs, get pos

ind. of chopping off extra bit

looking at 2s comp

- does expected result fit in range?





Convert from base b to decimal

$b = 10$

$$\underset{1000}{\underline{1}} \quad \underset{100}{\underline{3}} \quad \underset{10}{\underline{4}} \quad \underset{1}{\underline{7}}$$

$1000 \cdot 1 + 100 \cdot 3$
$+ 10 \cdot 4 + 1 \cdot 7$
$= 1,347$

b

$$\underset{b^3}{\underline{1}} \quad \underset{b^2}{\underline{3}} \quad \underset{b^1}{\underline{4}} \quad \underset{b^0}{\underline{7}}$$

$1 \cdot b^3 + 3 \cdot b^2 + 4 \cdot b^1$
$+ 7 \cdot b^0$

$b = 8$

$$\underset{8^3}{\underline{1}} \quad \underset{8^2}{\underline{3}} \quad \underset{8^1}{\underline{4}} \quad \underset{8^0}{\underline{7}}$$

$1 \cdot 8^3 + 3 \cdot 8^2 + 4 \cdot 8^1$
$+ 7 \cdot 8^0$

$b = 16$

$$\underset{16^3}{\underline{1}} \quad \underset{16^2}{\underline{3}} \quad \underset{16^1}{\underline{4}} \quad \underset{16^0}{\underline{7}} = 16^3 + 3 \cdot 16^2 + 4 \cdot 16$$
$+ 7 \cdot 1$

100000 = — 6?

flip bits 011111

add one $+1$

$\overline{}$

100000

$1 \cdot 2^5 = 32$

-32

3 A B 2

4 bits

3	0011
A	1010
B	1011
2	0010

→ 0011101010110010