# Logic: From Greeks to philosophers to circuits.

COS 116
3/14/2005
Instructor: Sanjeev Arora

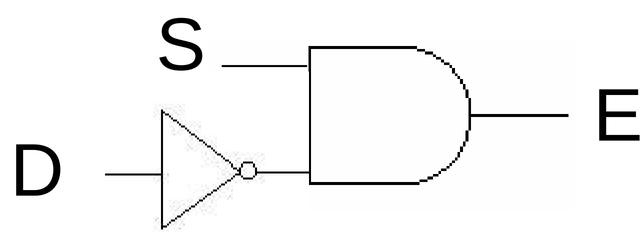
In addition to course handouts, many web-based resources; e.g.,
http://www.allaboutcircuits.com/vol_4/chpt_7/1.html

# 3 equivalent ways of representation

Ed goes to the party if Dan doesn't **and** Stella does

**Boolean Expression** $E = S \text{ AND } \overline{D} = S \cdot \overline{D}$

**Boolean Circuit**




**Truth table –** Gives value of E for every possible assignment to D, S.
TRUE=1; FALSE= 0.
(E is a "Boolean function" of D, S)

| D | S | E |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

# Boolean "algebra"

A **AND** B written as $A \cdot B$

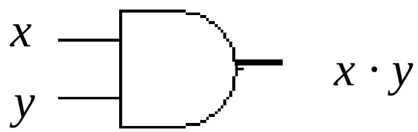
A **OR** B written as $A + B$

| | |
|---|---|
| $0 \cdot 0 = 0$ | $0 + 0 = 0$ |
| $0 \cdot 1 = 0$ | $1 + 0 = 1$ |
| $1 \cdot 1 = 1$ | $1 + 1 = 1$ ← Funny arithmetic |

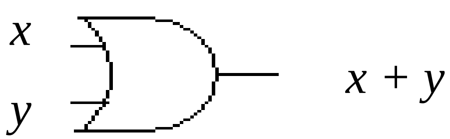
# Boolean gates

Shannon (1939)

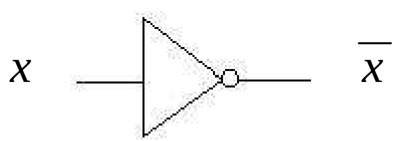
High voltage = 1
Low voltage = 0




Output voltage is high if both input voltages are high, otherwise output voltage low



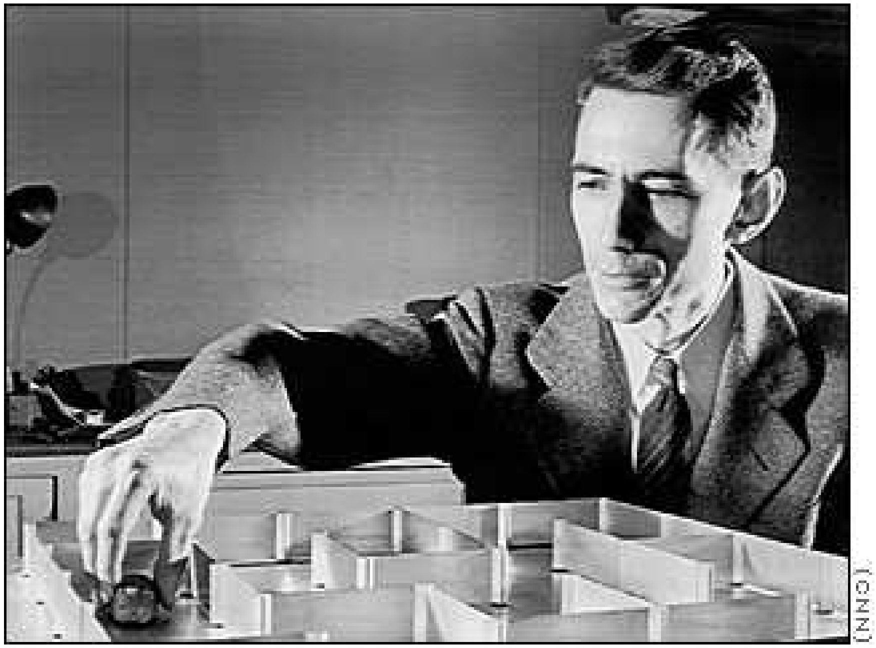

Output voltage is high if either input voltage is high, otherwise output voltage low




Output voltage is high if input voltage is low, otherwise output voltage low

# Claude Shannon (1916-2001)

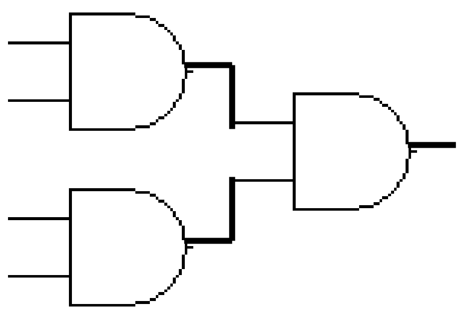
founder of many fields (circuits, information theory, artificial intelligence…)




With "Theseus" mouse

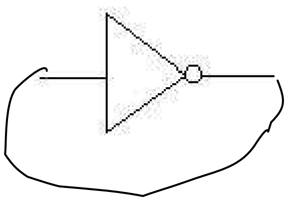
# Combinational circuit

- Boolean gates connected by wires



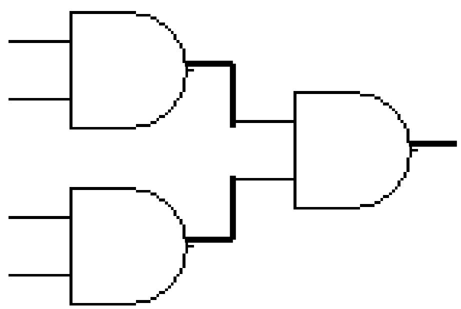
Wires: transmit voltage
(and hence value)

- Important: no cycles allowed

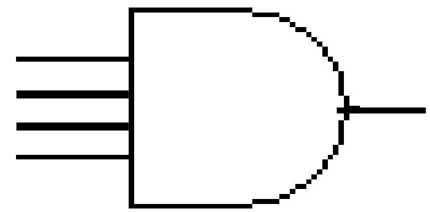


# Examples

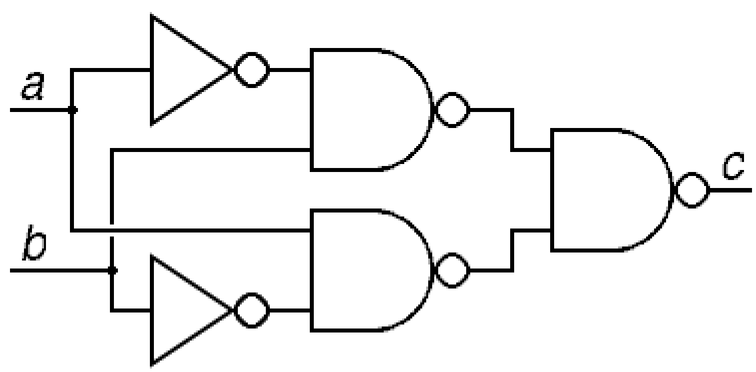
4-way AND



(Sometimes we use this for shorthand)

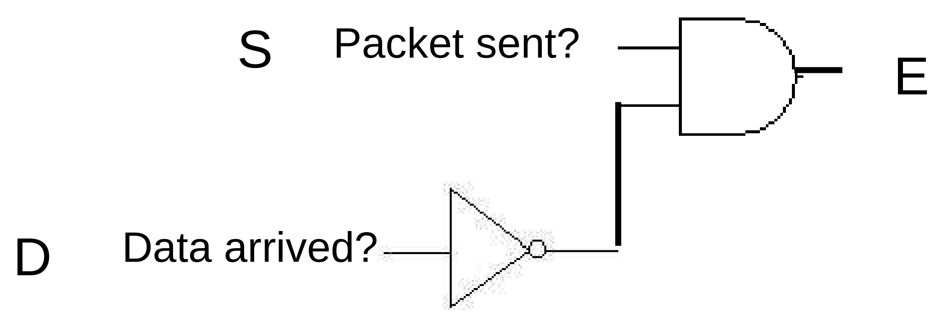


More complicated example



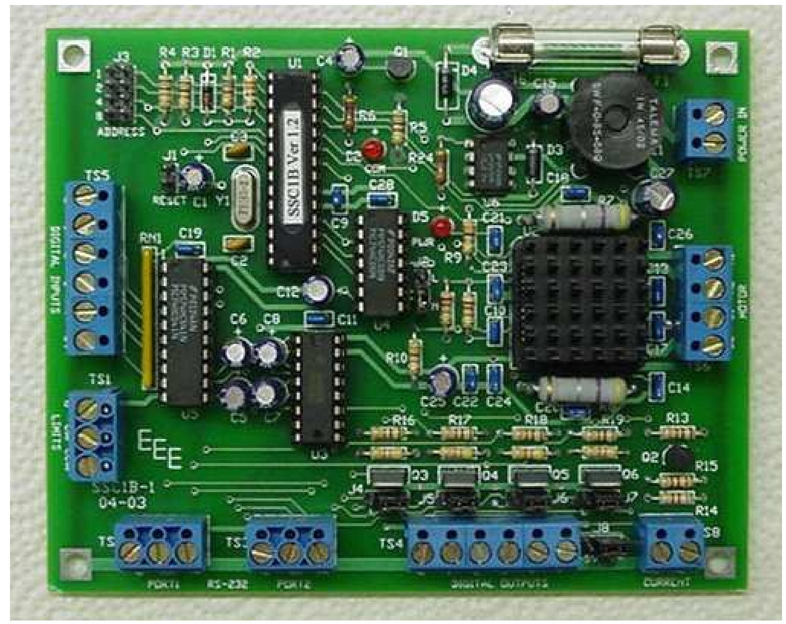
# Combinational circuits and control

- “If data has not arrived and packet has been sent, send a signal”

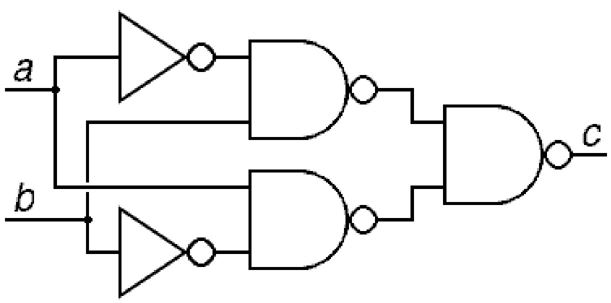


# Circuits compute functions

- Every combinational circuit computes a Boolean function of its inputs



Inputs




Outputs

# Ben Revisited

Ben only rides to class if he overslept, but even then if it is raining he'll walk and show up late to class (he really hates to bike in the rain). But if there's an exam that day he'll bike if he overslept, even in the rain.

**B**: Ben Bikes
**R**: It is raining
**E**: There is an exam today
**O**: Ben overslept

Give boolean expression for B in terms of R, E and O

# Ben’s truth table

| O | R | E | B |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# Going from truth table to Boolean expression

- Take OR of all input combinations that lead to 1

$$B = O \cdot \overline{R} \cdot \overline{E} + O \cdot \overline{R} \cdot E + O \cdot R \cdot E$$

| O | R | E | B |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Aside: AND, OR, and NOT gates suffice to implement every boolean function!

# Sizes of representations

- For *k* variables:

| A | B | … | X |
|---|---|---|---|
| 0 | 0 | … | 0 |
| 0 | 0 | … | 0 |
| 0 | 1 | … | 0 |
| 0 | 1 | … | 1 |
| … | … | … | … |
| … | … | … | … |
| 1 | 1 | … | 1 |

$2^k$ (rows) $k + 1$ (columns)

| $k$ | 10 | 20 | 30 |
|---|---|---|---|
| $2^k$ | 1024 | 1048576 | 1073741824 |

Tools for reducing size:
(a) circuit optimization (b) modular design

# Expression simplification

- Some simple rules:

  $x \cdot 1 = x$

  $x \cdot 0 = 0$

  $x + 0 = x$

  $x + 1 = 1$

  $x + x = x \cdot x = x$

  $x \cdot (y + z) = x \cdot y + x \cdot z$

  $x + (y \cdot z) = (x+y) \cdot (x+z)$

$x \cdot y + x \cdot \overline{y}$

$= x \cdot (y + \overline{y})$

$= x \cdot 1$

$= x$

De Morgan's Laws:

$\overline{x \cdot y} = \overline{x} + \overline{y}$

$\overline{x + y} = \overline{x} \cdot \overline{y}$

# Simplifying Ben's circuit

- $B = O \cdot \overline{R} \cdot \overline{E} + O \cdot \overline{R} \cdot E + O \cdot R \cdot E$

  $= O \cdot (\overline{R} \cdot \overline{E} + \overline{R} \cdot E + R \cdot E)$

  $= O \cdot (\overline{R} \cdot (\overline{E} + E) + R \cdot E)$

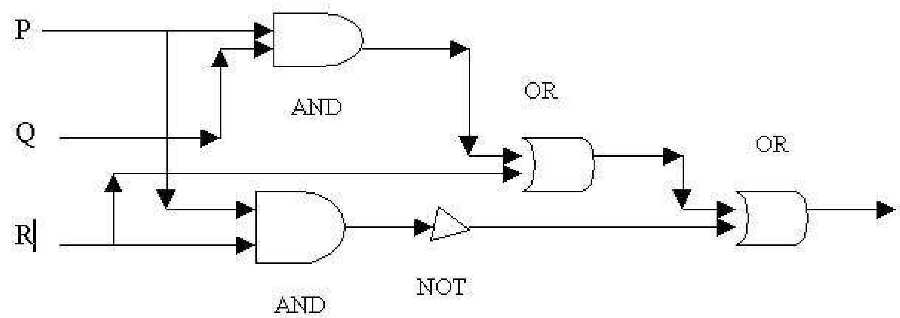
  $= O \cdot (\overline{R} + R \cdot E)$

  $= O \cdot ((\overline{R} + R) \cdot (\overline{R} + E))$

  $= O \cdot (\overline{R} + E)$

# Something to think about: How hard is Circuit Verification?

- Given a circuit, decide if it is trivial (either it always outputs 1 or always outputs 0 no matter the input)




- Alternative statement: Decide if there is any setting of the inputs that makes the circuit evaluate to 1.

  Time required?

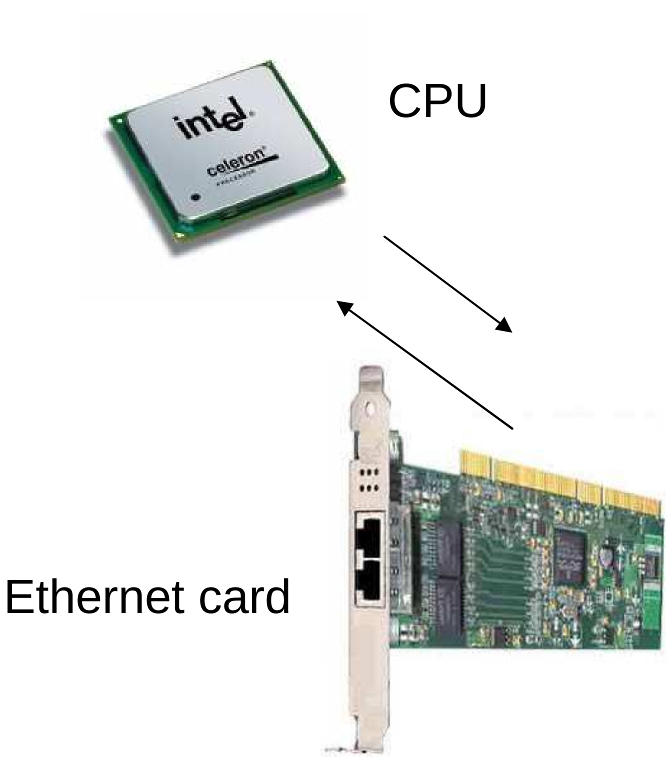
# Boole's reworking of Clarke's "proof" of existence of God (see handout)

- General idea: Try to prove that Boolean expressions $E_1$, $E_2$, …, $E_k$ cannot simultaneously be true
- Method: Show $E_1 \cdot E_2 \cdot \ldots \cdot E_k = 0$
- Discussion for next time: What exactly does Clarke's "proof" prove? How convincing is such a proof to you?

Also: Do Google search for "Proof of God's Existence."

# Going beyond combinational circuits

- Need 2-way communication between circuits (i.e., need to allow cycles!)




- Need memory (scratchpad)

# Circuit for Binary Addition?

```
25= 11001
29= 11101
--------------
   110110
```

Read handout: Will discuss next time.

# Worked out example: Going from truth table to Boolean expression

- Take OR of all input combinations that lead to 1

$$X = \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot C + A\cdot B\cdot\overline{C} + A\cdot B\cdot C$$

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

"Majority"