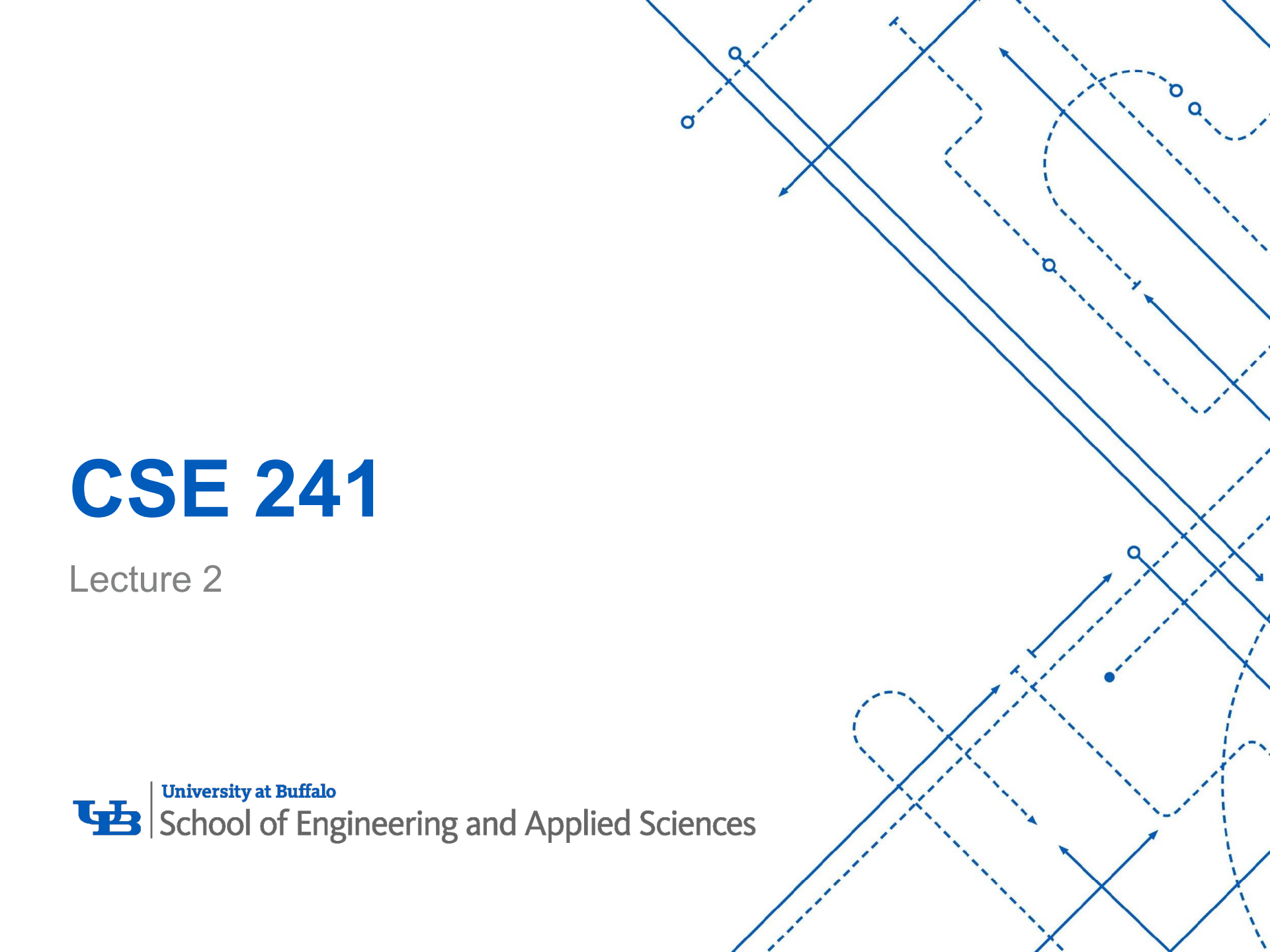


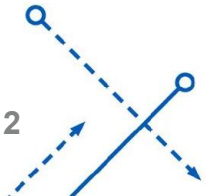
CSE 241

Lecture 2

University at Buffalo
School of Engineering and Applied Sciences

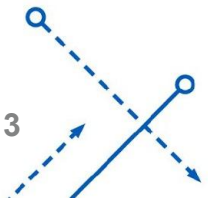
Overview for this lecture

- Reminders
- Boolean Algebra

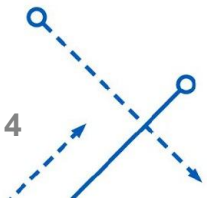


Reminders

- HW1 Due at 11:59pm tonight
  - Closes out tomorrow at 11:59pm
- First Recitation Today
- Next Monday is Exam Prep
- Next Wednesday is the Exam

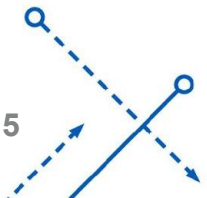


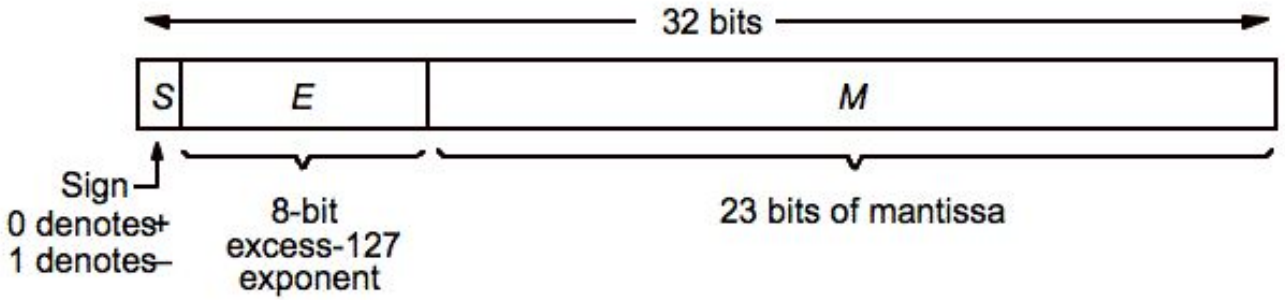
Any Questions?



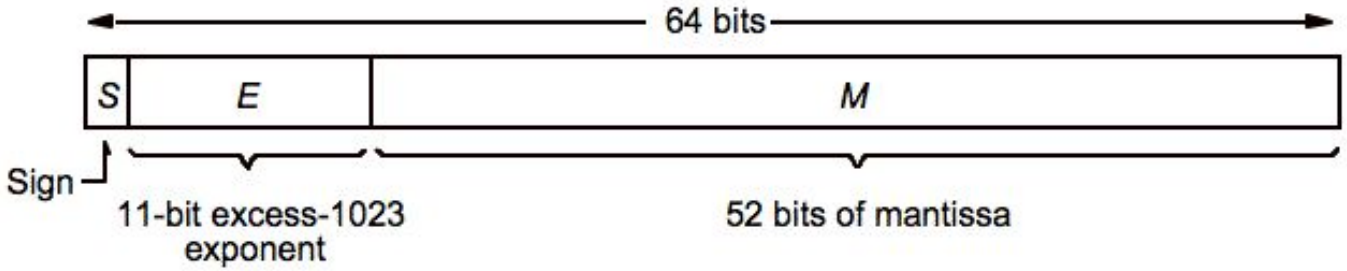
Floating Point Numbers

- Mantissa has significant digits.
- Exponent has power of 2
- Number has form
  - mantissa x $2^{exponent}$
- Typically normalized so mantissa has one assumed bit to the left of the radix point
- “Binary Scientific Notation”





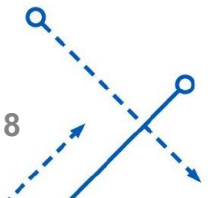
(a) Single precision



(c) Double precision

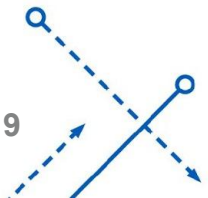
- Single-precision (32 bits)
  - left bit is sign (0 = +, 1 = -)
  - 23-bit mantissa
  - precision of about seven decimal digits
  - MSB bit of 1 is omitted
  - 8-bit sign (excess-127) (Exponent = E-127)
  - E = 0 denotes exactly 0
  - E = 255 denotes infinity
  - Range of about $10^{\pm 38}$
  - Value = $\pm 1.M \times 2^{E - 127}$

- Double-precision (64 bits)
  - left bit is sign (0 = +, 1 = -)
  - 52 bit mantissa
  - left bit assumed to be 1
  - precision of about 16 decimal digits
  - 11 bit exponent
  - excess-1023
  - range of about $10^{\pm 308}$
  - Value = $1.M \times 2^{(E - 1023)}$



How do we calculate it?

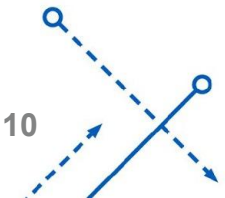
1. Determine sign
   a. This can be done at almost any step, I like to do it first
2. Convert the whole part of the number to binary
3. Convert the fraction part to binary
4. Put the whole and fraction parts in binary together
5. Calculate the exponent for scientific notation
6. Calculate the exponent field
7. Put it all together



Let's do an example

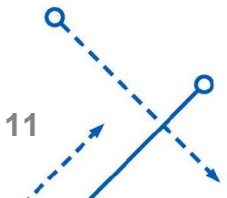
124.1495 to Floating Point

You will practice for Homework



Let’s go the other direction

11011001010010010010010100000000



BCD Numbers

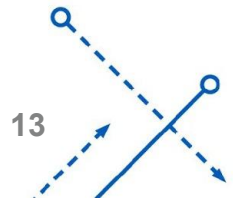
- Binary Coded Decimal
- Cuts the binary system off where we stop with decimal values
- Larger values require “two” digits

Decimal	Binary	BCD	Hex
0	0000	0000	0
1	0001	0001	1
2	0010	0010	2
3	0011	0011	3
4	0100	0100	4
5	0101	0101	5
6	0110	0110	6
7	0111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E

How do we add BCD?

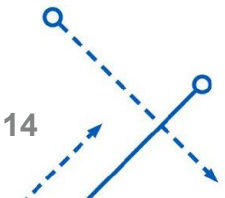
- Add like normal in Binary
- Then force a carry when appropriate
- When is it appropriate?

- How do you force a carry?



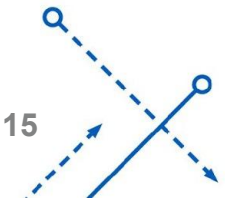
Example

Let’s add 5+9 in BCD



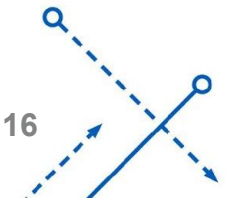
Exercise

You try adding 18 and 7 in BCD



Other Representations

- Gray Code
  - Shifts only one digit's value at a time
- Excess X
  - Places 0 at X and allows for positive and negative values



What is Boolean Algebra?

- It all started with George Boole
- Later we get DeMorgan's Laws
- Based on Binary Number System

Quick Tid Bit- Who Was George Boole?

- 1815-1864
- Mathematician
- Approached logic as mathematics that are reduced to 0s and 1s
- Developed Boolean Algebra

Claude Shannon

- 1916-2001
- Born in Petoskey, MI
- Master's Thesis at MIT
  - "A Symbolic Analysis of Relay and Switching Circuits"
  - This is a two-value Boolean Algebra known as *switching algebra*
- One of the engineers who coined the term "Artificial Intelligence"
  - Along with Marvin Minsky, John McCarty, and Nathaniel Rochester

What are the fundamentals of Boolean Algebra

- 0+0=0
- 0+1=1
- 1+1=1
- 0*0=0
- 0*1=0
- 1*1=1
- Think of it as True (1), False(0), and(*), & or(+)

Why do we need this instead of just using addition?

- Addition is limited
- We can do a lot more with logic
  - Including Addition

How do we solving these problem?

- There are going to be many cases to look at
  - 2 States for every variable
- We use Truth Tables
- Or we have equations (aka expressions)
  - depends what the question asks for
  - For example: X=A+B

What is a Truth Table

- A visual, tabular representation of a logical function.
- Lists all the inputs and the output for a function.
- Order of inputs is binary counting order
- If there are n inputs, you need 2^n lines in your truth table.
- Know your powers of 2!!!
- You will see lots of truth tables in this class.

What is a Truth Table?

- For our equation example: X=A+B
- All cases possible are represented

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Core Logic Operations

- AND
- OR
- NOT

What is the notation for these operations?

- Logical AND
  - $f = x_1 \bullet x_2 = x_1 x_2$
  - C/C++/Java/Verilog: L = x_1 && x_2;
- Logical OR
  - $f = x_1 + x_2$
  - C/C++/Java/Verilog: L = x_1 || x_2;
- Logical NOT
  - $f = x_1'$
  - C/C++/Java/Verilog: L = ~x_1;

Truth Table for AND and OR

x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1
		AND	OR

3 Input Version

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Some Generalizations

- AND
  - True iff all inputs are true, otherwise false
  - 1 iff all inputs are 1, otherwise 0
- OR
  - True iff any input is true, otherwise false
  - 1 iff any input is 1, otherwise 0

What is the Precedence?

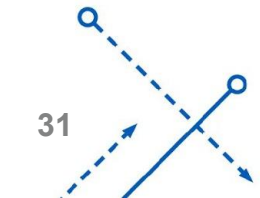
- Precedence = Order of Operations

1. Parenthesis
2. NOT
3. AND
4. OR

Let's Do An Example

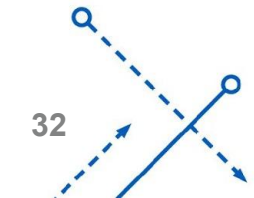
Let's Make The Truth Table for the Function:

F=x'+xy



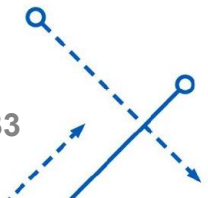
Now an exercise- Make the truth table

F= (x+y)+(x'y')



Boolean Postulates

1. Closure- regardless which operation you do the result stays in the set.
2. Commutative
    a. x+y=y+x
    b. x*y=y*x
3. Identity
    a. anything + 0 = itself
    b. anything times 1 is itself
4. Inverse
    a. if x=0, then x'=1
    b. if x=1, then x'=0
5. Distributive
    a. x*(y+z)=(x*y)+(x*z)
    b. x+(y*z)=(x+y)(x+z)
6. There exists at least 2 elements in the set; x, y s.t. x≠y

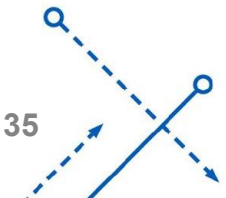


Single Variable Postulates and Theorems

- Postulate 2a: x+0=x
- Postulate 2b: x*1=x
- Postulate 5a: x+x'=1
- Postulate 5b: x*x'=0
- Theorem 1a: x+x=x
- Theorem 1b: x*x=x
- Theorem 2a: x+1=1
- Theorem 2b: x*0=0
- Theorem 3, involution: (x')'=x

Duality

- Did you notice that each axiom and theorem was two parts? This illustrates the principle of "duality".
- To find the dual of a function:
  - Change • → +
  - Change + → •
  - Change 0 → 1
  - Change 1 → 0
- May not seem like a useful concept… yet.
- Take-away point: The dual of a true statement is also a true statement.



More Useful Postulates and Theorems

- Postulate 3, Commutative
  - a: x+y=y+x
  - b: xy=yx
- Theorem 4, associative
  - a: x+(y+z)=(x+y)+z
  - b: x(yz)=(xy)z
- Postulate 4, distributive
  - a: x(y+z)=xy+xz
  - b: x+yz=(x+y)(x+z)
- Theorem 5, DeMorgan
  - a: (x+y)'=x'y'
  - b:(xy)'=x'+y'
- Theorem 6, Absorption
  - a: x+xy=x
  - b: x(x+y)=x

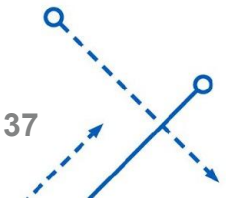
What is a Literal?

A Literal is another way to say term

For example: F=x+y has 2 literals in the equation

F=y has only 1 literal

F=xy+zy has 2 literals



Let's Consider The Complement Of A Function

What is the Complement of F=(A+B+C)?

We will use DeMorgan's Law and then simplify

What is the difference between Dual and Complement?

If F=x'yz+xyz'

The Dual is: (x'+y+z)*(x+y+z')

The dual is the interchange of AND and OR operators and 1's and 0's

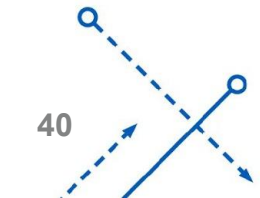
The Complement is: F'=(x+y'+z')*(x'+y'+z)

The complement is the interchange of 0's and 1's, so we have to NOT all the literals of the dual to get the complement of the original function

Let's look at the equations and Truth Table for this.

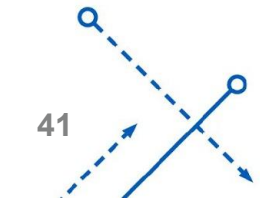
Take the dual of:

x+z+y+(tyz)



Take the complement of

F=AC+B'D+CD+A'D'

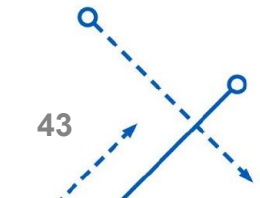


Some Terminology

- Normal form= x
- Complement form = x'
- Minterm = standard product term = abc
  - denoted on truth table with m_i
- Maxterm = standard sum term = (a+b+c)
  - denoted on truth tables with M_i
- Canonical= all literals contain all variables
- Reduced notation= Output_Variable(Input_Variables)=∑m(minterms)

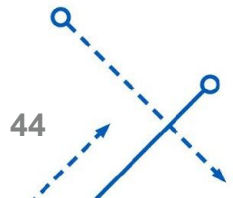
  or Output_Variable(Input_Variables)=∏M(maxterms)

Let's make a truth table with canonical min and max terms, and their designation



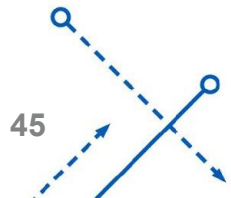
Let's Create the Truth Table and SOP

$F(A,B,C,D)=\sum m(0,1,4,8,13,15)$



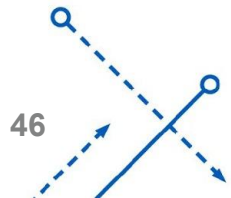
You Try

$F(A,B,C)=\sum m(0,1,2,7)$



What about POS?

F(A,B,C,D)=∏M(0,1,4,10,15)



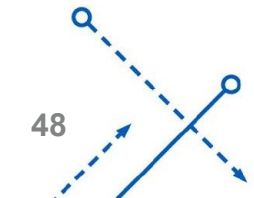
You Try

$F(A,B,C)=\prod M(1,3,7)$

What about simplification?

This is when you apply the Boolean Axioms

For now, anyway

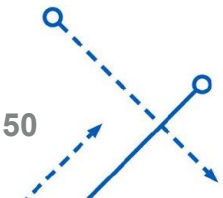


Why do we want to simplify

- When it is time to implement physically there are some limitations
  - Like the number of inputs on a gate or how many gates you have

Let's Simplify An Equation Algebraically

F=X+X+X+X+X+YX+YX+X'YZ



Exercise

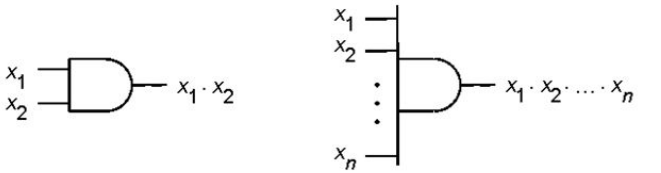
Simplify Algebraically:

F=Y+Y+YX+YX'+X'

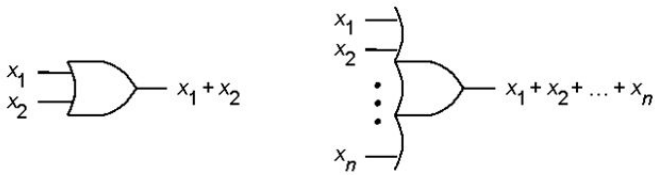
What is a gate

- Logic gates are electronic circuits
- Logic gate symbols are visual ways to represent logic operations
  - These are representing actual circuit elements
  - They give some variety since you will get sick of truth tables
- Some gates have one input and some have multiple inputs

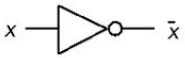
Basic Gates



(a) AND gates



(b) OR gates

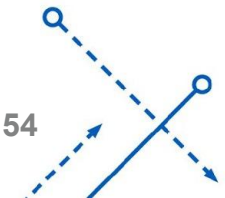


(c) NOT gate

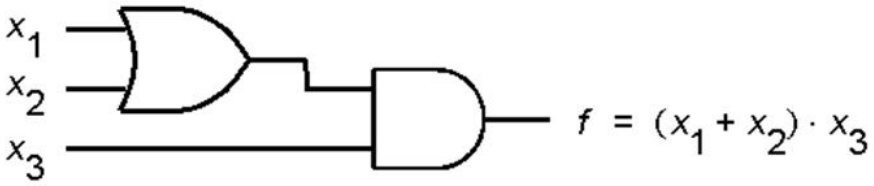
Figure 2.8. The basic gates.

Gate Behavior

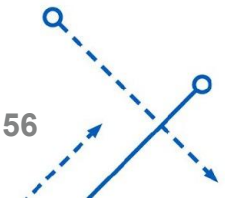
- AND gates produce an output of 1 iff all n inputs are 1.
- OR gates produce an output of 1 if any of the n inputs are 1.
- NOT gates just output the opposite of whatever they originally got.
- It can be shown that any combinational logic function can implemented with these three gates.
- Therefore, it is a complete logic set.



Putting It Together

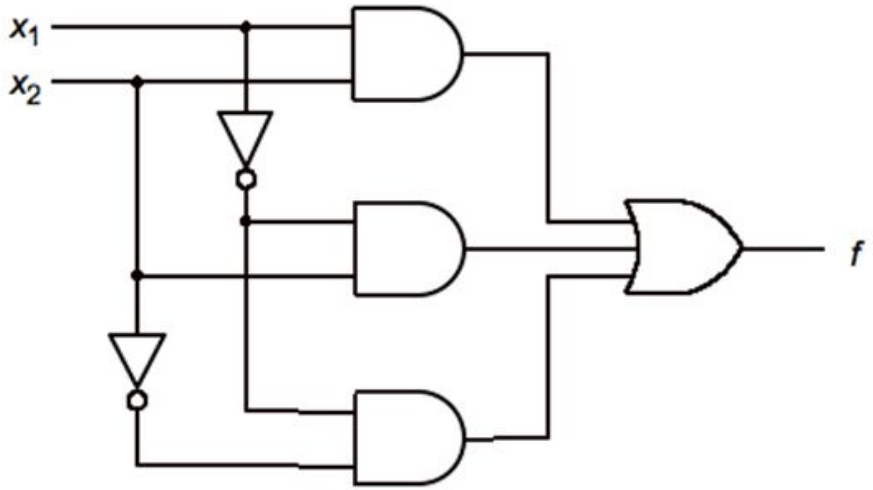


Let's Draw Out Some Functions

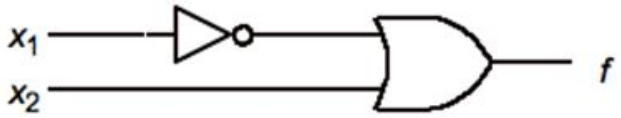


What is the difference between Canonical and Minimal?

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1



(a) Canonical sum-of-products



(b) Minimal-cost realization

What is the Cost?

- Cost=Inputs+Gates
- Costs=#Literals+#Terms+#Nots

Let's Practice Some

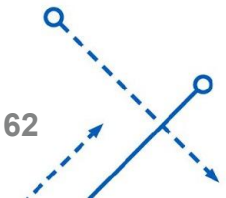
It will become more important later

When to Do Canonical? When to do Minimal?

- If you are told- do which one you are told
- If you are not told- whichever you want

What about the number of inputs on the gates?

- When told- work with the number of inputs specified
- When not told- you decide



Exercise

F(A,B,C) = ΠM(0,2,3,5); find canonical SOP and POS forms equations and gate diagrams. Use only 2 input logic gates.

Questions?