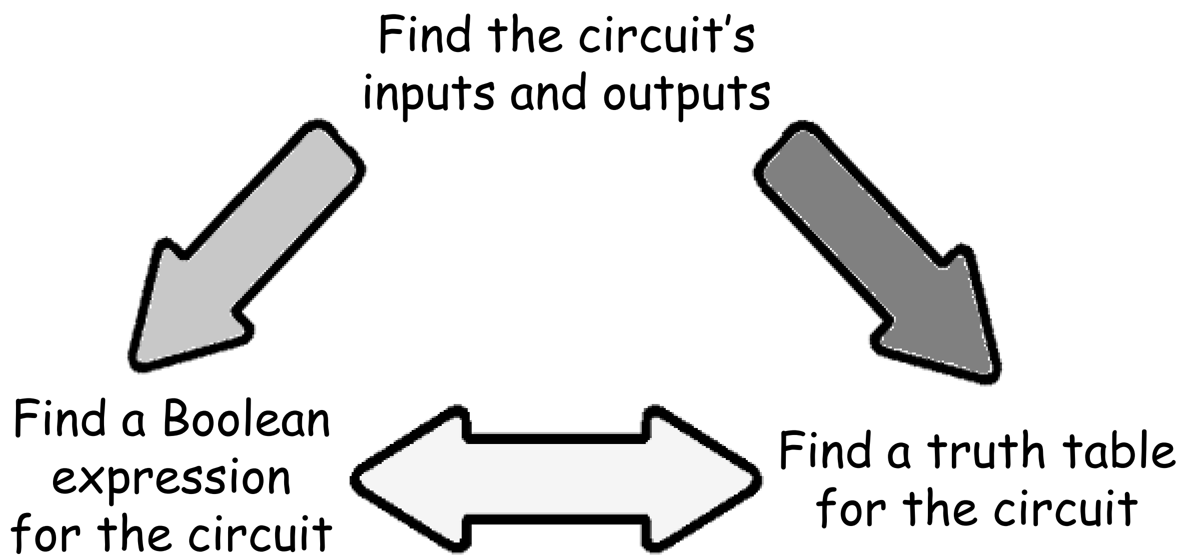
Boolean Algebra

Circuit analysis summary

- After finding the circuit inputs and outputs, you can come up with either an expression or a truth table to describe what the circuit does.
- You can easily convert between expressions and truth tables.



Boolean Functions summary

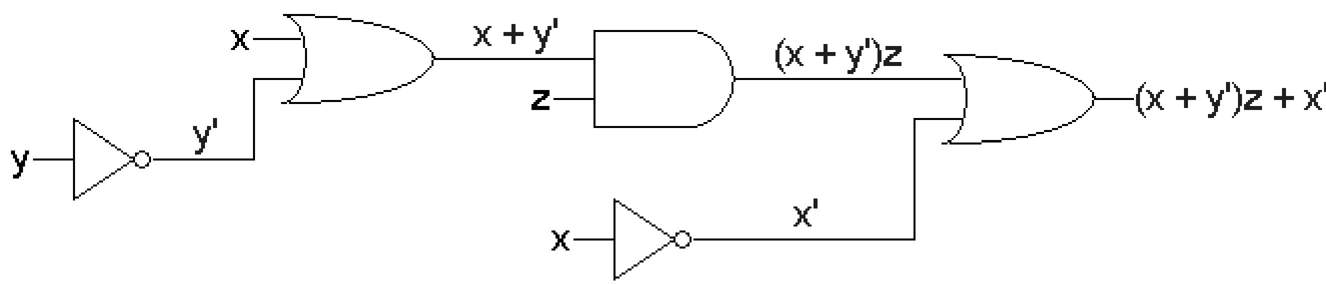
- We can interpret high or low voltage as representing true or false.
- A variable whose value can be either 1 or 0 is called a Boolean variable.
- AND, OR, and NOT are the basic Boolean operations.
- We can express Boolean functions with either an expression or a truth table.
- Every Boolean expression can be converted to a circuit.
- Now, we'll look at how Boolean algebra can help simplify expressions, which in turn will lead to simpler circuits.

Boolean Algebra

- Last time we talked about Boolean functions, Boolean expressions, and truth tables.
- Today we'll learn how to how use Boolean algebra to simplify Booleans expressions.
- Last time, we saw this expression and converted it to a circuit:

$$(x + y')z + x'$$

Can we make this circuit "better"?
•Cheaper: fewer gates
•Faster: fewer delays from inputs to outputs



Expression simplification

- Normal mathematical expressions can be simplified using the laws of algebra
- For binary systems, we can use Boolean algebra, which is superficially similar to regular algebra
- There are many differences, due to
  - having only two values (0 and 1) to work with
  - having a complement operation
  - the OR operation is not the same as addition

Formal definition of Boolean algebra

- A Boolean algebra requires
  - A set of elements **B**, which needs *at least* two elements (0 and 1)
  - Two binary (two-argument) operations OR and AND
  - A unary (one-argument) operation NOT
  - The axioms below must always be true (textbook, p. 42)
    - The magenta axioms deal with the complement operation
    - Blue axioms (especially 15) are different from regular algebra

1. $x + 0 = x$	2. $x \bullet 1 = x$	
3. $x + 1 = 1$	4. $x \bullet 0 = 0$	
5. $x + x = x$	6. $x \bullet x = x$	
7. $x + x' = 1$	8. $x \bullet x' = 0$	
9. $(x')' = x$		
10. $x + y = y + x$	11. $xy = yx$	Commutative
12. $x + (y + z) = (x + y) + z$	13. $x(yz) = (xy)z$	Associative
14. $x(y + z) = xy + xz$	15. $x + yz = (x + y)(x + z)$	Distributive
16. $(x + y)' = x'y'$	17. $(xy)' = x' + y'$	DeMorgan's

Comments on the axioms

- The associative laws show that there is no ambiguity about a term such as x + y + z or xyz, so we can introduce multiple-input primitive gates:

- The left and right columns of axioms are duals
  - exchange all ANDs with ORs, and 0s with 1s
- The dual of *any* equation is always true

1. $x + 0 = x$	2. $x \bullet 1 = x$	
3. $x + 1 = 1$	4. $x \bullet 0 = 0$	
5. $x + x = x$	6. $x \bullet x = x$	
7. $x + x' = 1$	8. $x \bullet x' = 0$	
9. $(x')' = x$		
10. $x + y = y + x$	11. $xy = yx$	Commutative
12. $x + (y + z) = (x + y) + z$	13. $x(yz) = (xy)z$	Associative
14. $x(y + z) = xy + xz$	15. $x + yz = (x + y)(x + z)$	Distributive
16. $(x + y)' = x'y'$	17. $(xy)' = x' + y'$	DeMorgan's

Are these axioms for real?

- We can show that these axioms are valid, given the definitions of AND, OR and NOT

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

- The first 11 axioms are easy to see from these truth tables alone. For example, x + x' = 1 because of the middle two lines below (where y = x')

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

Proving the rest of the axioms

- We can make up truth tables to prove (both parts of) DeMorgan's law
- For $(x + y)' = x'y'$, we can make truth tables for $(x + y)'$ and for $x'y'$

x	y	x + y	(x + y)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

x	y	x'	y'	x'y'
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

- In each table, the columns on the left (x and y) are the inputs. The columns on the right are outputs.
- In this case, we only care about the columns in blue. The other "outputs" are just to help us find the blue columns.
- Since both of the columns in blue are the same, this shows that $(x + y)'$ and $x'y'$ are equivalent

Simplification with axioms

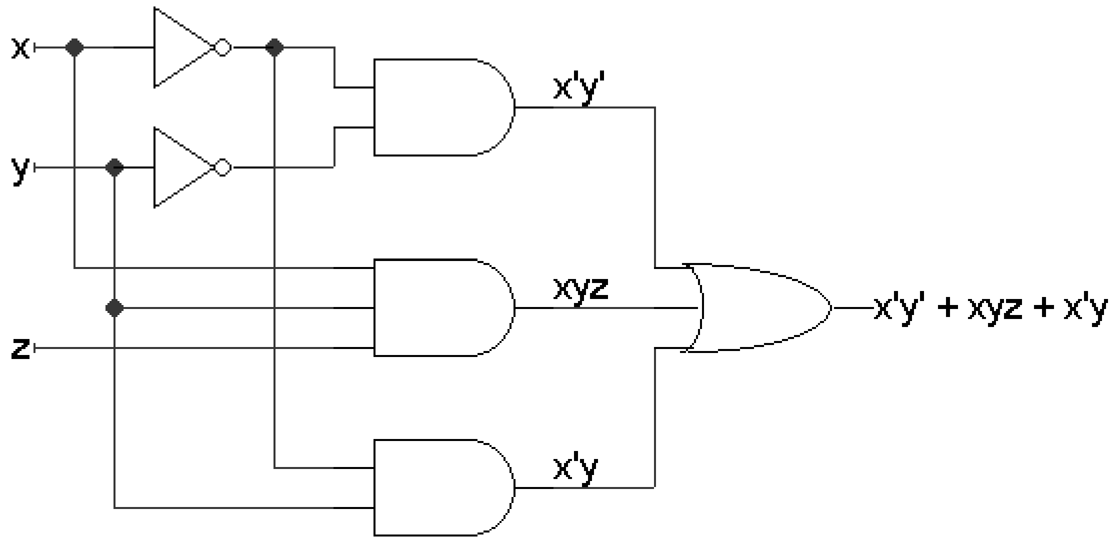
- We can now start doing some simplifications

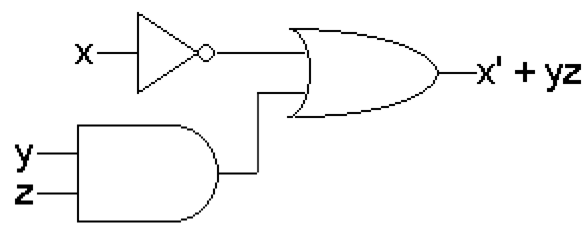
$x'y' + xyz + x'y$	
$= x'(y' + y) + xyz$	[Distributive; $x'y' + x'y = x'(y' + y)$]
$= x' \bullet 1 + xyz$	[Axiom 7; $y' + y = 1$]
$= x' + xyz$	[Axiom 2; $x' \bullet 1 = x'$]
$= (x' + x)(x' + yz)$	[Distributive]
$= 1 \bullet (x' + yz)$	[Axiom 7; $x' + x = 1$]
$= x' + yz$	[Axiom 2]

1. $x + 0 = x$	2. $x \bullet 1 = x$	
3. $x + 1 = 1$	4. $x \bullet 0 = 0$	
5. $x + x = x$	6. $x \bullet x = x$	
7. $x + x' = 1$	8. $x \bullet x' = 0$	
9. $(x')' = x$		
10. $x + y = y + x$	11. $xy = yx$	Commutative
12. $x + (y + z) = (x + y) + z$	13. $x(yz) = (xy)z$	Associative
14. $x(y + z) = xy + xz$	15. $x + yz = (x + y)(x + z)$	Distributive
16. $(x + y)' = x'y'$	17. $(xy)' = x' + y'$	DeMorgan's

Let's compare the resulting circuits

- Here are two different but *equivalent* circuits.
- In general the one with fewer gates is "better":
  - It costs less to build
  - It requires less power
  - But we had to do some work to find the second form





Some more laws

- Here are some more useful laws. Notice the duals again!

1. $x + xy = x$	4. $x(x + y) = x$
2. $xy + xy' = x$	5. $(x + y)(x + y') = x$
3. $x + x'y = x + y$	6. $x(x' + y) = xy$
$xy + x'z + yz = xy + x'z$	$(x + y)(x' + z)(y + z) = (x + y)(x' + z)$

- We can prove these laws by either
  - Making truth tables:

x	y	x	x'y	x + x'y
0	0			
0	1			
1	0			
1	1			

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

  - Using the axioms:

$$
\begin{aligned}
x + x'y &= (x + x')(x + y) && [\text{ Distributive }] \\
&= 1 \bullet (x + y) && [\ x + x' = 1\ ] \\
&= x + y && [\text{ Axiom 2 }]
\end{aligned}
$$

The complement of a function

- The complement of a function always outputs 0 where the original function outputted 1, and 1 where the original produced 0.
- In a truth table, we can just exchange 0s and 1s in the output column(s)

$f(x,y,z) = x(y'z' + yz)$

x	y	z	f(x,y,z)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

→

x	y	z	f'(x,y,z)
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

Complementing a function algebraically

- You can use DeMorgan's law to keep "pushing" the complements inwards

$$f(x,y,z) = x(y'z' + yz)$$

$$\begin{aligned} f'(x,y,z) &= (\ x(y'z' + yz)\ )' && [\text{ complement both sides }] \\ &= x' + (y'z' + yz)' && [\text{ because } (xy)' = x' + y'\ ] \\ &= x' + (y'z')'\ (yz)' && [\text{ because } (x + y)' = x'\ y'\ ] \\ &= x' + (y + z)(y' + z') && [\text{ because } (xy)' = x' + y', \text{ twice}] \end{aligned}$$

- You can also take the dual of the function, and then complement each literal
  - If $f(x,y,z) = x(y'z' + yz)$…
  - …the dual of f is $x + (y' + z')(y + z)$…
  - …then complementing each literal gives $x' + (y + z)(y' + z')$…
  - …so $f'(x,y,z) = x' + (y + z)(y' + z')$

Summary so far

- So far:
  - A bunch of Boolean algebra trickery for simplifying expressions and circuits
  - The algebra guarantees us that the simplified circuit is *equivalent* to the original one

- Next:
  - Introducing some standard forms and terminology
  - An alternative simplification method
  - We'll start using all this stuff to build and analyze bigger, more useful, circuits

i>clicker®

- $(x+y')(x+y) + x'yz$
  - A: $x + xy + y'x + x'yz$
  - B: 0
  - C: $x + yz$
  - D: $x + x'yz$

1.	$x + xy = x$	4.	$x(x + y) = x$
2.	$xy + xy' = x$	5.	$(x + y)(x + y') = x$
3.	$x + x'y = x + y$	6.	$x(x' + y) = xy$
$xy + x'z + yz = xy + x'z$		$(x + y)(x' + z)(y + z) = (x + y)(x' + z)$	

i▸clicker®

- $ab'c + a'bc' + b(a'c' + a'(a+c'))'$

A: $b+ac$
B: $b+ba+ca$
C: $ca+cb+ba+bc'$
D: abc

1.	$x + xy = x$	4.	$x(x + y) = x$
2.	$xy + xy' = x$	5.	$(x + y)(x + y') = x$
3.	$x + x'y = x + y$	6.	$x(x' + y) = xy$
$xy + x'z + yz = xy + x'z$		$(x + y)(x' + z)(y + z) = (x + y)(x' + z)$	