Digital Logic Optimization I

CSCI 2050U - Computer Architecture

Randy J. Fortier
@randy_fortier

Ontario Tech University

Outline

- Review: Logical equivalence
- Boolean algebra axioms
- Boolean algebraic simplification

Boolean Algebraic Simplification

- Boolean algebraic simplification is one way to optimize a circuit
  - Simplification involves using a set of *axioms*/logical equivalences to eliminate operators, or even variables from an expression
  - A smaller expression normally translates to a simpler/cheaper circuit
  - The resulting expression must be *logically equivalent*

Review: Logical Equivalence

- Two Boolean algebraic expressions are logically equivalent iff they have the same truth table
- Example:

a	**b**	**a'+b'**	**(ab)'**
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Boolean Algebra Axioms

CSCI 2050U - Computer Architecture

Boolean Algebra - Axioms

- Some axioms are intuitive:

AND Form	***OR Form***	***Name***
1x = x	0 + x = x	Identity law
0x = 0	1 + x = 1	Null law
xx = x	x + x = x	Idempotent law
xx′ = 0	x + x′ = 1	Inverse law

Boolean Algebra - Axioms

- Some axioms are intuitive:

AND Form	*Name*
$x'' = x$	Double complement law

Boolean Algebra - Axioms

- Other axioms are familiar from (non-Boolean) algebra:

AND Form	***OR Form***	***Name***
`xy = yx`	`x + y = y + x`	Commutative law
`(xy)z = x(yz)`	`(x + y) + z = x + (y + z)`	Associative law
`x + yz = (x + y)(x + z)`	`x(y + z) = xy + xz`	Distributive law

Boolean Algebra - Axioms

- Others have been proved by Mathematicians:

AND Form	***OR Form***	***Name***
$x(x + y) = x$	$x + xy = x$	Absorption law
$(xy)' = x' + y'$	$(x + y)' = x'y'$	DeMorgan's law

Boolean Algebraic Simplification

CSCI 2050U - Computer Architecture

Ontario Tech University

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                             (identity)
```

Boolean Algebraic Simplification

- Let’s do an example:

```
f(x,y,z) = xy + x’z + yz
=  xy + x’z + yz1                              (identity)
=  xy + x’z + yz(x + x’)                 (inverse)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                          (identity)
=  xy + x'z + yz(x + x')             (inverse)
=  xy + x'z + yzx + yzx'             (distributive)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x’z + yz
=  xy + x’z + yz1                        (identity)
=  xy + x’z + yz(x + x’)           (inverse)
=  xy + x’z + yzx + yzx’           (distributive)
=  xy + x’z + x(yz) + x’(yz)       (commutative x2)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                          (identity)
=  xy + x'z + yz(x + x')           (inverse)
=  xy + x'z + yzx + yzx'           (distributive)
=  xy + x'z + x(yz) + x'(yz)       (commutative x2)
=  xy + x'z + (xy)z + (x'z)y       (associative x2)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                            (identity)
=  xy + x'z + yz(x + x')               (inverse)
=  xy + x'z + yzx + yzx'               (distributive)
=  xy + x'z + x(yz) + x'(yz)           (commutative x2)
=  xy + x'z + (xy)z + (x'z)y           (associative x2)
=  xy + (xy)z + x'z + (x'z)y           (commutative)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                          (identity)
=  xy + x'z + yz(x + x')             (inverse)
=  xy + x'z + yzx + yzx'             (distributive)
=  xy + x'z + x(yz) + x'(yz)         (commutative x2)
=  xy + x'z + (xy)z + (x'z)y         (associative x2)
=  xy + (xy)z + x'z + (x'z)y         (commutative)
=  xy(1 + z) + x'z(1 + y)            (distributive x2)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                         (identity)
=  xy + x'z + yz(x + x')                  (inverse)
=  xy + x'z + yzx + yzx'                  (distributive)
=  xy + x'z + x(yz) + x'(yz)              (commutative x2)
=  xy + x'z + (xy)z + (x'z)y              (associative x2)
=  xy + (xy)z + x'z + (x'z)y              (commutative)
=  xy(1 + z) + x'z(1 + y)                 (distributive x2)
=  xy1 + x'z1                             (null x2)
```

Boolean Algebraic Simplification

- Let's do an example:

```
f(x,y,z) = xy + x'z + yz
=  xy + x'z + yz1                                (identity)
=  xy + x'z + yz(x + x')              (inverse)
=  xy + x'z + yzx + yzx'              (distributive)
=  xy + x'z + x(yz) + x'(yz)          (commutative x2)
=  xy + x'z + (xy)z + (x'z)y          (associative x2)
=  xy + (xy)z + x'z + (x'z)y          (commutative)
=  xy(1 + z) + x'z(1 + y)             (distributive x2)
=  xy1 + x'z1                                        (null x2)
=  xy + x'z                                          (identity
x2)
```

Boolean Algebraic Simplification

- Let's compare:

```
f(x,y,z) = xy + x'z + yz          (5 gates, 10 transistors)


f(x,y,z) =  xy + x'z                   (4 gates, 7 transistors)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

$f(x,y,z) = xy'z' + xy'z + xyz' + xyz$

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z) = xy'z' + xy'z + xyz' + xyz
= xy'(z'+z) + xy(z'+z)                (distributive x 2)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z)  = xy'z'  + xy'z + xyz' + xyz
= xy'(z'+z)  + xy(z'+z)                    (distributive x 2)
= xy'1 + xy1                               (inverse x 2)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z) = xy'z' + xy'z + xyz' + xyz
= xy'(z'+z) + xy(z'+z)                  (distributive x 2)
= xy'1 + xy1                            (inverse x 2)
= xy' + xy                              (identity x 2)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z) = xy'z' + xy'z + xyz' + xyz
= xy'(z'+z) + xy(z'+z)                (distributive x 2)
= xy'1 + xy1                          (inverse x 2)
= xy' + xy                            (identity x 2)
= x(y'+y)                             (distributive)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z)  = xy'z'  + xy'z + xyz' + xyz
= xy'(z'+z)  + xy(z'+z)                (distributive x 2)
= xy'1 + xy1                           (inverse x 2)
= xy'  + xy                            (identity x 2)
= x(y'+y)                              (distributive)
= x1                                   (inverse)
```

Boolean Algebraic Simplification

- Let's do another example (our `Neg` expression from earlier):

```
f(x,y,z) = xy'z' + xy'z + xyz' + xyz
= xy'(z'+z) + xy(z'+z)               (distributive x 2)
= xy'1 + xy1                         (inverse x 2)
= xy' + xy                           (identity x 2)
= x(y'+y)                            (distributive)
= x1                                 (inverse)
= x                                  (identity)
```

Boolean Algebraic Simplification

- Let's compare:

```
f(x,y,z) = xy'z' + xy'z + xyz' + xyz    (7 gates, 17 trans)


f(x,y,z) = x                            (0 gates, 0 trans)
```

Boolean Algebraic Simplification

- The main problem with Boolean algebraic simplification is that it is not obvious what to do next
  - We need an oracle to tell us which axiom to use next
  - This is not an algorithmic approach
  - We need something systematic, so we can never fail if we follow the steps
- Boolean algebraic simplification has been shown to be NP-complete in the general case
  - You'll learn more about this later, but in short, NP-complete is a class of problems where we aren't sure there is an efficient solution (in fact, we're pretty sure there isn't)

Wrap-Up

- Logical equivalence
- Boolean algebra axioms
- Boolean algebraic simplification
  - An NP-complete problem

What is next?

- Karnaugh maps