Computer Architecture
(CSC-3501)
Lecture 5
(29 Jan 2008)

Seung-Jong Park (Jay)
http://www.csc.lsu.edu/~sjpark

Announcement

Reason to Learn Boolean Algebra

- Digital computers contain circuits that implement Boolean functions.
- The simpler that we can make a Boolean function, the smaller the cir cuit that will result.
  - Simpler circuits are cheaper to build, consume less power, and run faste r than complex circuits.
- With this in mind, we always want to reduce our Boolean functions t o their simplest form.
- There are a number of Boolean identities that help us to do this.

Boolean Identities

- Most Boolean identities have an AND (product) form as well as an OR (sum) form. We give our identities using both forms. Our first group is rather intuitive:

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$x\overline{x} = 0$	$x + \overline{x} = 1$

- We can use Boolean identities to simplify the function: $F(X,Y,Z) = (X + Y)(X + \overline{Y})(\overline{X\overline{Z}})$

$(X + Y)(X + \overline{Y})(\overline{X\overline{Z}})$	Idempotent Law (Rewriting)
$(X + Y)(X + \overline{Y})(\overline{X} + Z)$	DeMorgan's Law
$(XX + X\overline{Y} + XY + Y\overline{Y})(\overline{X} + Z)$	Distributive Law
$((X + Y\overline{Y}) + X(Y + \overline{Y}))(\overline{X} + Z)$	Commutative & Distributive Laws
$((X + 0) + X(1))(\overline{X} + Z)$	Inverse Law
$X(\overline{X} + Z)$	Idempotent Law
$X\overline{X} + XZ$	Distributive Law
$0 + XZ$	Inverse Law
XZ	Idempotent Law

DeMorgan's law

- DeMorgan's law provides an easy way of finding the complement of a Boolean function.
- Recall DeMorgan's law states:

$$\overline{(xy)} = \overline{x} + \overline{y} \quad \text{and} \quad \overline{(x+y)} = \overline{x}\,\overline{y}$$

- DeMorgan's law can be extended to any number of variables.
- Replace each variable by its complement and change all ANDs to O Rs and all ORs to ANDs.
- Thus, we find the the complement of: $F(X,Y,Z) = (XY) + (\overline{X}Z) + (Y\overline{Z})$

$$\begin{aligned}\overline{F}(X,Y,Z) &= \overline{(XY) + (\overline{X}Z) + (Y\overline{Z})} \\ &= \overline{(XY)}\,\overline{(\overline{X}Z)}\,\overline{(Y\overline{Z})} \\ &= (\overline{X}+\overline{Y})(X+\overline{Z})(\overline{Y}+Z)\end{aligned}$$

Canonical Form of Boolean Function

- Through our exercises in simplifying Boolean expressions, we see th at there are numerous ways of stating the same Boolean expression.
  - These "synonymous" forms are *logically equivalent.*
  - Logically equivalent expressions have identical truth tables.
- In order to eliminate as much confusion as possible, designers expres s Boolean functions in *standardized* or *canonical* form.
- There are two canonical forms for Boolean expressions: sum-of-produ cts and product-of-sums.
  - Recall the Boolean product is the AND operation and the Boolean sum is the OR operation.
- In the sum-of-products form, ANDed variables are ORed together.
  - For example: $F(x,y,z) = xy + xz + yz$
- In the product-of-sums form, ORed variables are ANDed together:
  - For example: $F(x,y,z) = (x+y)(x+z)(y+z)$

Sum of Product Form

- It is easy to convert a function to sum-of-prod ucts form using its truth table.
- We are interested in the values of the variabl es that make the function true (=1).
- Using the truth table, we list the values of the variables that result in a true function value.
- Each group of variables is then ORed togeth er.

- The sum-of-products form for our function is:

$$F(x,y,z) = \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}\bar{z} + xy\bar{z} + xyz$$

$$F(x,y,z) = x\bar{z} + y$$

x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1