VE270 Recitation Class

2020.5.27

Yiqing Zhao

Outline

- Boolean Algebra
- Minterms & Maxterms
- Logic Optimization
  1. By using Boolean Algebra
  2. Using K-map
  3. **Quine-McCluskey Method

Boolean Algebra

Terminology

- Variable: 0, 1, a, b, c, ...
- Literal: a, a', b, b', ...
- Product Term: abc, a, ...
- Sum Term: a+b+c, a, ...
- Sum-of-Product (SOP): $f(a,b,c) = ab' + a'b + ab'c$
- Product-of-Sum (POS): $f(a,b,c) = (a + b')(a' + b)(a' + b + c)$

Boolean Algebra

Terminology

- Example:
  1. a
  2. $a(b+c)$

Boolean Algebra

Basic Theorems

	Version (a)	Version (b)
T1: Identities	$x \cdot 1 = x$	$x + 0 = x$
T2: Null Elements	$x \cdot 0 = 0$	$x + 1 = 1$
T3: Idempotence	$x \cdot x = x$	$x + x = x$
T4: Complements	$x \cdot x' = 0$	$x + x' = 1$
T5:	$xy + xy' = x$	$(x + y)(x + y') = x$
T6: Commutativity	$x \cdot y = y \cdot x$	$x + y = y + x$
T7: Absorption	$x \cdot (x + y) = x$	$x + x \cdot y = x$
T8:	$x \cdot (x' + y) = xy$	$x + x'y = x + y$
T9: Associativity	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x + y) + z = x + (y + z)$
T10: Distributivity	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x + y) \cdot (x + z)$
T11: Consensus	$xy+x'z+yz=xy+x'z$	$(x+y)(x'+z)(y+z)=(x+y)(x'+z)$
T12: Involution	$(x')'=x$	
T13: De Morgen	$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$

Boolean Algebra

Basic Theorems

- Application:

  1. Prove $x + 1 = 1$ (Using Basic Theorems)
  2. Prove $(x_1 + x_2 + \ldots + x_n)' = x_1' x_2' \ldots x_n'$ (Using Induction)
  3. Consider the following expression: $E = \{x \mid [(y \mid x) \mid x]\} \mid [y \mid (z \mid x)]$. The operator symbol | is usually interpreted as $a \mid b = (ab)'$. Show that E is equivalent to a single literal L, i.e., $E = L$

Boolean Algebra

XOR Properties

	Version (a)	Version (b)
T1:	$x \oplus 0 = x$	$x \oplus 1 = x'$
T2:	$x \oplus x = 0$	$x \oplus x' = 1$
T3:	$x \oplus y' = x' \oplus y = (x \oplus y)'$	
T4: Commutative	$x \oplus y = y \oplus x$	
T5: Associative	$(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$	

Minterms & Maxterms

Definition

- Minterm m_i can be expressed as an AND (product) term of n literals
- Maxterms M_i can be expressed as an OR (sum) term of n literals
- Theorem: $m_i = (M_i)'$
- Example: For 4 variables a, b, c, d

1. $m_5 = a'bc'd \qquad (0101)$
2. $M_5 = a + b' + c + d' \qquad (0101)$

Minterms & Maxterms

Find Expression

- Addition of all minterms that produce a logic 1 for the corresponding output
- Multiplication of all maxterms that produce a logic 0 for the corresponding output

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Question: What is the minterm logic equation of F? What about the maxterm logic equation?

Minterms & Maxterms

Don't Cares

- Don't Cares: Output that is not completely specified, denoted as "x", can be 0/1

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	X
0	1	1	1
1	0	0	X
1	0	1	0
1	1	0	1
1	1	1	1

$$F = \sum m(0,3,6,7) \text{ with } d(2,4)$$

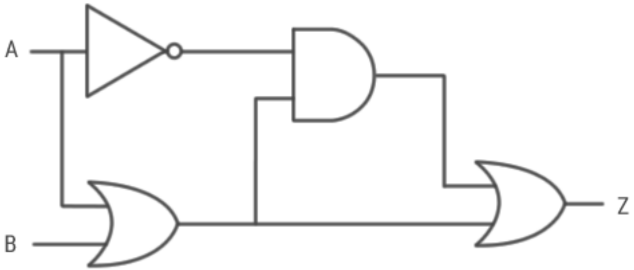
Logic Optimization

General Description

- We optimize the circuit to improve performance.
    1. Delay - Response time from input to output
    2. Size - Number of Transistors
- General Estimation:
    1. Every gate has delay of “1 gate-delay”
    2. Every gate input requires 2 transistors
    3. Ignore inverters
- Critical Path: Longest Delay Path from an input to output

Logic Optimization

General Description



Question:

How many transistors are used in this circuit?

What is the delay?

What is the critical path?

Logic Optimization

Using Boolean Algebra

- Using Boolean Algebra theorems to reduce size(delay)
- Example: $F = (a' + c' + d')(b' + c' + d')(a' + b + c + d)(b' + c + d)$

Logic Optimization

Using K-map

- A graphical technique to simplify the logic equation
- Procedure:
    1. Building
    2. Grouping and Canceling
    3. Writing equations

Logic Optimization

Building K-map

- 2-variable map

F: a \ b	0	1
0	0	1
1	2	3

- 3-variable map

F: a \ bc	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Logic Optimization

Building K-map

- 4-variable map

F ab\cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

- 5-variable map

F ab\cde	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Logic Optimization

Grouping & Cancelling

- Group in shape of rectangle or square
- Group the adjacent 1's until all the 1's are grouped
- The number of 1's in the group should be 2^N
- Collect as many 1's as possible
- No zeros in the group
- Edges wrap around
- If both primed and unprimed forms of a letter appear in a same group, the letter cancels

Logic Optimization

Writing Equations

- Prime Implicant (PI): A group that cannot be entirely contained by another implicant
- Essential PIs: If a cell is covered ONLY by that PI
- Theorem 1: Every irredundant SOP expression that specifies F is a sum of PIs of F.
- Theorem 2: An irreudndant SOP expression must contain every essential PI.

Logic Optimization

Using K-maps

- Example:
  $Z(a, b, c, d, e) = \sum m(0,1,12,15,16,20,23,25,31) + d(2,5,8,9,17,18,22,27)$

Logic Optimization

Using K-maps

- Building

Logic Optimization

Using K-maps

- Grouping and Canceling

Logic Optimization

Using K-maps

- PIs:
- Essential PIs:
- Final equation:

Logic Optimization

Using Quine-McCluskey Method

- List minterms by ascending group index
- Group adjacent minterms
- Group adjacent 3-literal product terms (Dashes must in the same position for grouped terms)
- The unchecked term is the corresponding PI
- Use Covering Table to find essential PI and the result

Logic Optimization

Using Quine-McCluskey Method

- Example:

  $$Z(a,b,c,d,e) = \sum m(0,1,12,15,16,20,23,25,31) + d(2,5,8,9,17,18,22,27)$$

Logic Optimization

Using Quine-McCluskey Method

Logic Optimization

Using Quine-McCluskey Method

Any Questions?