3827 OH

Eumin Hong (eh2890)

Columbia University

February 8, 2022

Overview

Announcements

Announcements: Upcoming Assessments

- HW2 is due on Friday 2/11
- HW3 is due on Friday 2/18
- HW4 is due on Friday 2/25
- Homework is due at 11:59 pm (follow Gradescope time for deadline)

Announcements: Poll Results

- Online OH (except MIPS): 80%
- Email only: 62.5%
- Will create Slack for those who would like to join; will check frequently

Announcements: Feedback Form

- Form: `https://forms.gle/cnUmKVNYN7WvRbHA6`

Notable Concepts

Notable Concepts: 2's Complement Overflow Logic

- Why can you just check the last two carries to detect overflow in 2's complement?
- Let P be a positive number and N be a positive number. Consider the cases where the carries are different (implying overflow; carries are left to right):

1. $P + P$: if carries are 0 and 1, then result is negative
2. $N + N$: if carries are 1 and 0, then result is positive
3. $P + N$: if carries are 0 and 1, then result is positive
    - To see that the result should be positive, compare P and N
    - Let A be P without the MSB and B be N without MSB
    - In order for the second carry to be 1, $A > B$ (unsigned comparison), so $A < \overline{B}$
    - We consider $\overline{B}$ since we are actually comparing $|P|$ and $|N|$, and so $|P| < |N|$ and output should be negative
4. $P + N$: if carries are 1 and 0, then result is negative (same logic as above)

Notable Concepts: SoP, PoS

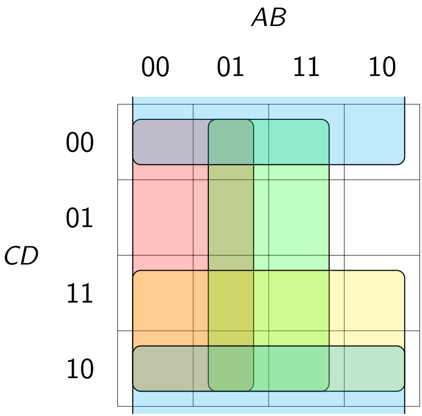
- Sum of Products: OR of product terms; e.g. $F = \overline{Y} + \overline{X}Y\overline{Z} + XY$
- Product of Sums: AND of sum terms; e.g. $F = X(\overline{Y} + Z)(X + Y + \overline{Z})$
- SoP and PoS are not the simplest form, just a canonical form
    - E.g. $F = X \oplus Y$ is simpler than $F_{\text{SoP}} = X\overline{Y} + \overline{X}Y$ and $F_{\text{PoS}} = (X + Y)(\overline{X} + \overline{Y})$

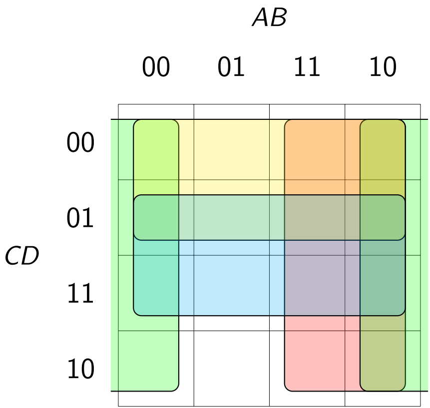
Notable Concepts: Minterms and Maxterms

- Minterm: product term with all variables appearing exactly once (either complemented or not); e.g. $\overline{A}BC\overline{D}$
  - Evaluates to 1 for exactly one variable assignment combination, 0 otherwise
- Maxterm: sum term with all variables appearing exactly once (either complemented or not); e.g. $A + \overline{B} + \overline{C} + D$
  - Evaluates to 0 for exactly one variable assignment combination, 1 otherwise

Notable Concepts: Minterms, Maxterms, and K-maps

- How do minterms and maxterms relate to K-maps?
- For minterm $\overline{A}BC\overline{D}$ (left) and maxterm $A+\overline{B}+\overline{C}+D$ (right):





Notable Concepts: Converting SoP to PoS

1. Compute $\overline{F}$: swap AND and OR operations, flip literals
2. Convert $\overline{F}$ to SoP form
3. Compute $\overline{\overline{F}} = F$: swap AND and OR operations, flip literals

- Complemented F twice
- Complementing F is SoP form gets $\overline{F}$ in PoS form
- Example: converting $F = \overline{A}\,\overline{C} + \overline{B}$ into PoS form

1. $\overline{F} = (A + C)B$
2. $\overline{F} = AB + BC$
3. $\overline{\overline{F}} = F = (\overline{A} + \overline{B})(\overline{B} + \overline{C})$

Notable Concepts: Simplifying with K-Maps

- Try to take largest essential prime implicant
- Remember that K-maps are "wrapped around":

CD \ AB	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Notable Concepts: Recognizing XOR in K-Maps

- K-maps for $A \oplus B$ (left) and $A \oplus B \oplus C$ (right):

B \ A	0	1
0	0	1
1	1	0

C \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

- Note the checkered pattern
- Minterms/maxterms have more literals than the XOR operations

Notable Concepts: XOR in K-Maps Example

CD \ AB	00	01	11	10
00	1	0	1	0
01	0	0	0	0
11	0	0	0	0
10	0	1	0	1

- Note that the middle two rows (D) are all 0

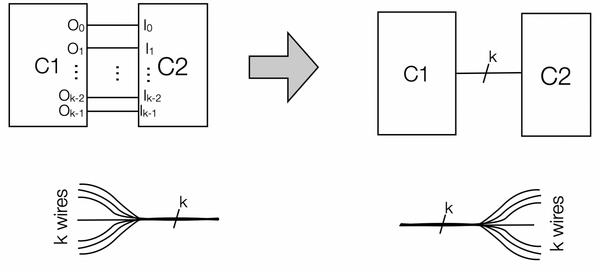
- The final expression is of the form $F = \overline{D}(F')$ where F' is mostly XORs
- Remove middle two rows:

C \ AB	00	01	11	10
0	1	0	1	0
1	0	1	0	1

- $F = \overline{D}(\overline{A \oplus B \oplus C})$

Notable Concepts: Multi-Wire Notation

- Can have multiple wires in parallel (from `3827_Lecture_04.pdf`, slide 30):



- But remember: these are just wires
  - E.g. how to get the MSB of a k-bit number represented by k wires?

Notable Concepts: Combinational Circuits

- Enabler: for "turning on/off" inputs
  - High-level circuit for other circuits, "enable" functionality can be added with an input labelled "E"
- Decoder: set one of 2^k outputs to 1 (others are 0) based on unsigned binary representation of k-bit input
- Multiplexer (MUX): pass one of 2^k inputs as output based on k-bit "selector" input
- Shifter: shifts bits of k-bit input
  - Can have input to control how much bits are shifted by, or can be hardcoded
- Adders (unsigned, 2's complement): half/full adders can make ripple carry adder
  - Carry lookahead also exists (reduced depth with increased parallelism)

Notable Concepts: Decoder/MUX Notation

- Must label everything in your combinational circuits
- Consider the two MUXes:

Homework 2

Homework 2: Problem 1

Simplify the following Boolean expressions to a minimum number of literals:

a. $\overline{B}C + \overline{B}\,\overline{C}D$
b. $\overline{(\overline{Y} + Z)}(Y + \overline{Z})$
c. $\overline{C}\,\overline{D} + C\overline{D}A + DA$
d. $X(\overline{C}Y + \overline{C}\,\overline{Y}) + X(\overline{W}\,\overline{C} + \overline{W}C)$
e. $(\overline{B}\,\overline{C} + BD + \overline{C}D)(\overline{B} + C + \overline{B}C)$

Homework 2: Problem 2

Reduce the following Boolean expressions to the indicated number of literals:

a. $Y + \overline{Z}(W + \overline{Y + W})$ to two literals
b. $\overline{Y}X + Y\overline{X}Z + \overline{Y}\,\overline{X}$ to three literals
c. $\overline{Y}X(\overline{W} + ZW) + X(Y + \overline{Y}\,\overline{Z}W)$ to one literal

Homework 2: Problem 3

Using DeMorgan's theorem (as many times as necessary), express the function $F = \overline{X}Z + X\overline{Z}Y + \overline{X}\,\overline{Y}$ with only:

(a) OR and complement operations

(b) AND and complement operations

Homework 2: Problem 4

Find the complement of the following expressions:

a. $\overline{B}\,\overline{D} + BD$
b. $(C + \overline{B}D)(C + \overline{B} + \overline{D})(\overline{C}\,\overline{B} + \overline{D})$
c. $\overline{W}\,\overline{Y}(Z\overline{X} + \overline{Z}X) + WY(Z + X)(\overline{Z} + \overline{X})$
d. $(\overline{B} + \overline{D})AC + E$

Homework 2: Problem 5

Convert the following into sum-of-products and product-of-sums forms:

- (a) $(\bar{A} + \bar{B}\bar{D})(\bar{D} + A\bar{C})$
- (b) $(\bar{Z} + Y)\bar{X}(\bar{X} + Z) + X$
- (c) $(\bar{Z} + XW + \overline{W}Y)(\bar{X} + \bar{Z}\bar{Y})$

Homework 2: Problem 6

Simplify the following boolean expressions using a K-map:

(a) $B\overline{C} + CD + B\overline{D} + \overline{B}\,\overline{C}D$

(b) $\overline{Y}\,\overline{Z} + \overline{X}\,\overline{Z} + ZX\overline{Y}$

(c) $XW + \overline{X}\,\overline{Z} + X\overline{Z}\,\overline{W}$

Homework 2: Problem 7

Simplify in Sum-of-product form via a K-map (indicate what you identify as prime implicants and essential prime implicants). Recall that $m(i)$ is the product term whose variables are complemented when and only when their position corresponds to a '0' in the binary representation of i, e.g., $m(5) = \overline{W}X\overline{Y}Z$.

(a) $F(W, X, Y, Z) = \sum m(0, 4, 5, 7, 9, 12, 13, 14)$
(b) $F(A, B, C, D) = \sum m(0, 1, 2, 4, 5, 8, 9, 10, 11, 13, 15)$
(c) $F(A, B, C, D) = \sum m(0, 2, 3, 5, 7, 8, 10, 11, 14, 15)$
(d) $F(W, X, Y, Z) = \sum m(0, 1, 2, 5, 6, 7, 8, 9, 10, 13, 14, 15)$
(e) $F(W, X, Y, Z) = \sum m(0, 1, 2, 5, 8, 9, 11, 12)$
(f) $F(A, B, C, D) = \sum m(1, 3, 6, 7, 9, 13, 14, 15)$

Homework 2: Problem 8

Simplify into Sum-of-product form the following Boolean functions F together with the don't care conditions d:

(a) $F(W, X, Y, Z) = \sum m(0, 1, 3, 5, 7), d(W, X, Y, Z) = \sum m(2, 4, 6)$
(b) $F(A, B, C, D) = \sum m(2, 6, 7, 11, 14, 15), d(A, B, C, D) = \sum m(0, 8, 12, 13)$
(c) $F(A, B, C, D) = \sum m(2, 7, 9, 10, 15), d(A, B, C, D) = \sum m(3, 6, 14)$