EECS 151/251 A Discussion 4

Feb 9, 2024

Content

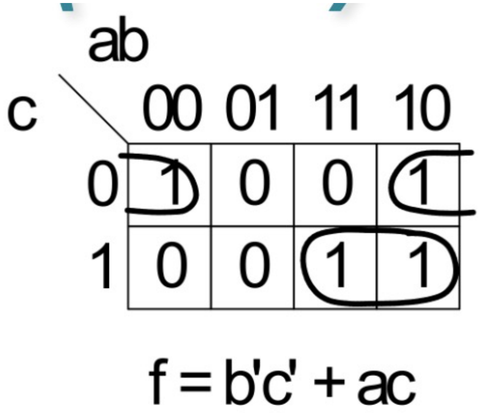
- K-maps
- Converting to NANDs and NORs
- FSMs
- Problems

Karnaugh Maps

- Tools to simplify Boolean Expressions
- Based on Uniting Theorem:
  - $xy' + xy = x (y' + y) = x$
- Go from function of 2 variables -> 1
- Rewrite truth table as a grid where adjacent cells have one variable change (00->01->11->10)
  - Note that the standard 00->01->10->11 does not work since 01->10 flips 2 variables
- Adjacent 1s represent product terms

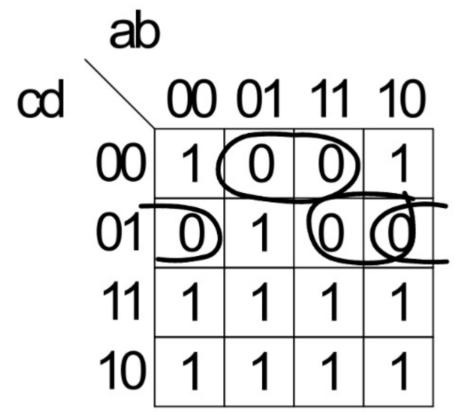
SOP K-maps

- Groups of 1s must be dimensions of 2^n (1x1, 2x2, 2x1, 4x2, etc..)
- Result in SOP expression
- Note that edges are adjacent!



POS K-Maps

- Find groups of 0 with same rules
- If constant terms are 0, use them, if constant terms are 1 use their complement
- Create POS expression
- How does this relate to DeMorgan's Law?



$$f = (b' + c + d)(a' + c + d')(b + c + d')$$

POS K-Map intuition

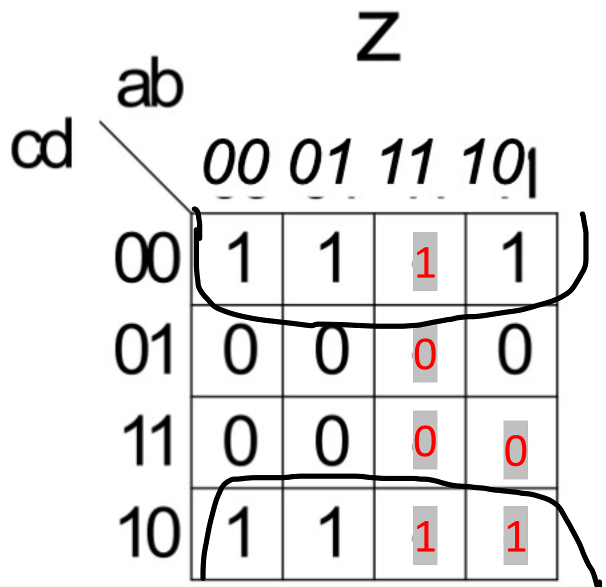
- You are just inverting the SOP expression for F'!
- When you invert SOP, you get POS of the inversion
  - SOP: bc'd' + ac'd + bc'd
  - Invert: (bc'd' + ac'd + bc'd)' = (bc'd')'(ac'd)'(bc'd)' =
  - (b' + c + d)(a' + c + d')(b + c+ d')

cd \ ab	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

$$f = (b' + c + d)(a' + c + d')(b + c + d')$$

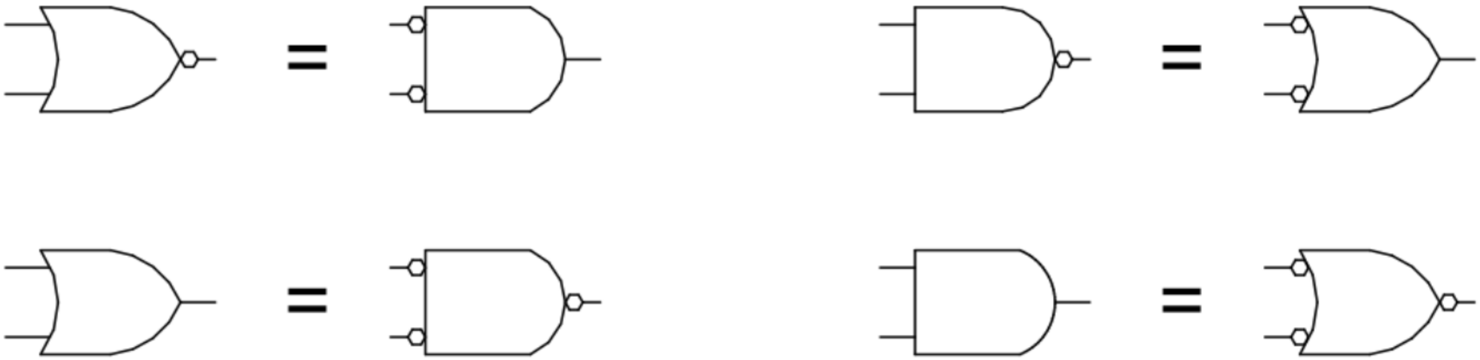
Don’t Cares

- You can circle as 0s or 1s, which ever lets you make bigger and fewer blocks in general!
- Z = d’



Converting to NANDs/NORs

- Add bubbles everywhere (inputs and outputs) when converting from ANDs to Ors (DeMorgan's)



Finite State Machines

- Ways to design Sequential Circuits (Universal)
- Moore: Output based only on current state
- Mealy: Output based on current state and input
- Transition between states with incoming input
- We will talk about how to implement them with logic circuits next week

Problem 1: K-Maps

1. Given the following truth table, first construct a Karnaugh map.

a	**b**	**c**	**d**	**y**
0	0	0	0	1
0	0	0	1	X
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	0
1	1	1	1	X

Table 1: Truth Table

2. Using the Karnaugh Map, give the most simplified SOP expression to describe truth table.
3. With the same Karnaugh Map, give the most simplified POS expression to describe the truth table.

1. Karnaugh Map

	$\overline{cd}$	$\bar{c}d$	cd	$c\bar{d}$
$\overline{ab}$	1	X	1	0
$\bar{a}b$	0	X	1	0
ab	0	X	X	0
$a\bar{b}$	1	0	X	0

2.

	$\overline{cd}$	$\bar{c}d$	cd	$c\bar{d}$
$\overline{ab}$	1	X	1	0
$\bar{a}b$	0	X	1	0
ab	0	X	X	0
$a\bar{b}$	1	0	X	0

$$y = cd + b'c'd'$$

3.

	$\overline{cd}$	$\bar{c}d$	cd	$c\bar{d}$
$\overline{ab}$	1	X	1	0
$\bar{a}b$	0	X	1	0
ab	0	X	X	0
$a\bar{b}$	1	0	X	0

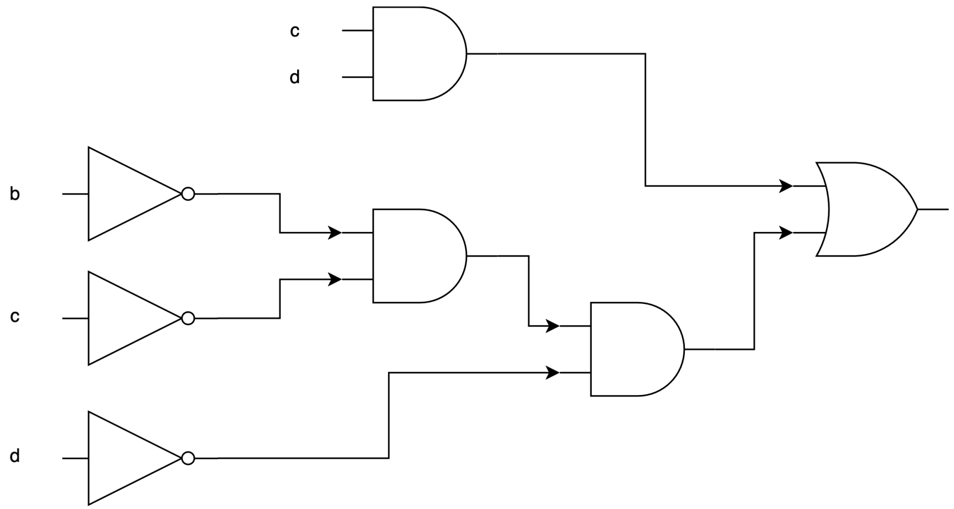
$$y = (a' + d')(b' + c)(c' + d)$$

Problem 2: Converting to a NAND circuit

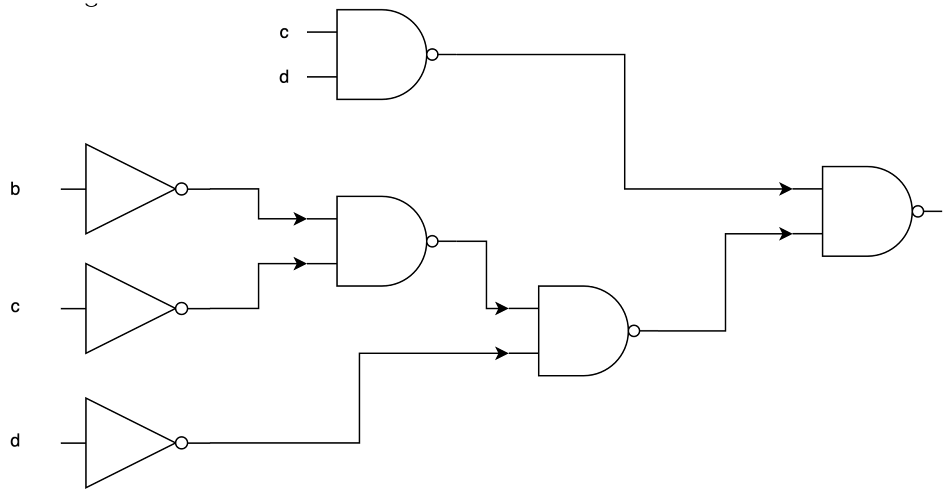
1. From the previous SOP expression, please give the AND/OR circuit diagram described by the expression.
2. Convert the previous diagram to a circuit with only NANDs and Inverters.

$$y = cd + b'c'd'$$

1



2



Berkeley
UNIVERSITY OF CALIFORNIA

Problem 3: State Machine to Detect 3 Divisibility

Draw the State Diagram (Moore) which takes as input a single bit of an n-bit binary number, and outputs whether the number is divisible by 3. The first bit received will be the MSB. For example for the bit stream 1,1,0,0,1,0,1,0,1 (receiving from the left first) we will have:

Input Bit	Total Number	Output of Machine
1	1	0
1	3 (11)	1
0	6 (110)	1
0	12 (1100)	1
1	25 (11001)	0
0	50 (110010)	0
1	101 (1100101)	0
0	202 (11001010)	0
1	405 (110010101)	1

Solution: