CSE 241

Lecture 3

University at Buffalo
School of Engineering and Applied Sciences

Overview for this lecture

- Reminders
- Finish SOP/POS Problem
- Karnaugh Map
- More Logic Gates

Reminders

- HW1 Due at 11:59pm tonight
    - Closes out tomorrow at 11:59pm
- First Recitation Today
- Next Monday is Exam Prep
- Next Wednesday is the Exam

Exercise

$F(A,B,C) = \Pi(0,2,3,5)$; find canonical SOP and POS forms equations and gate diagrams. Use only 2 input logic gates.

Reminder

- Any logic function can be designed from a truth table in either of two canonical forms: SOP and POS.
- For SOP
  - Find minterms for rows with f = 1.
  - Make one AND for each minterm.
  - OR the ANDs together.
- For POS
  - Find maxterms for rows with f = 0.
  - Make one OR for each maxterm.
  - AND the ORs together.

Karnaugh Map (K-Map)

- Visual mapping tool
- Combines adjacent 1's (for SOP) into pairs
  - vertically
  - horizontally
  - NOT diagonally
- Combines adjacent pairs into blocks of 4, 8, etc.
- For POS, we combine 0's

Two Variable Minterms

x_1	x_2	
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

(a) Truth table

$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

(b) Karnaugh map

Three Variable Minterms

x_1	x_2	x_3	
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

(a) Truth table

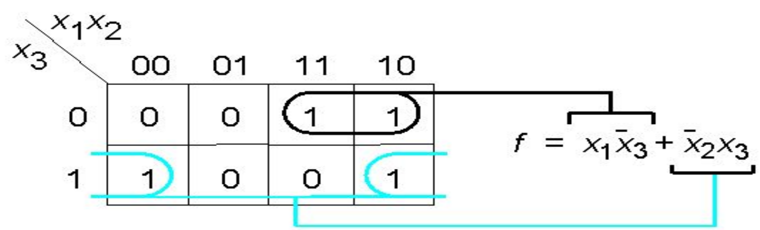
x_3 \ $x_1 x_2$	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

(b) Karnaugh map

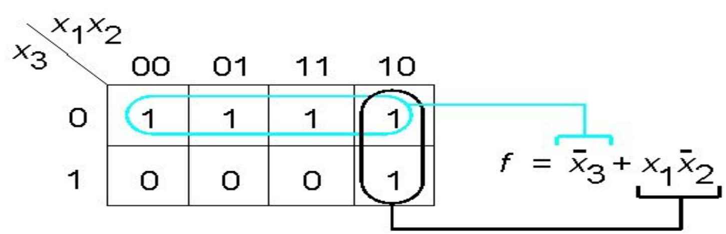
Labeling 1’s on SOP K-Maps

- 1s on SOP K-maps are minterms
- they are labeled the way minterms are
  - if row or column heading for a variable X is
    - 0 => X’
    - 1 => X

Three Variable K-Maps



(a) The function of Figure 2.18



(b) The function of Figure 4.1

Let's Practice Some K-Map Drawing

$F(A, B, C) = \sum(1,2,7)$

Exercise

$F(A, B, C) = \sum(0,1,6,7)$

Terms

- literal - true or complemented variable, e.g A, C’
- implicant - product that if 1 means function is 1
  - can be minterm, pair, etc., e.g. AB’, C, AC’D
- prime implicant (PI) - implicant that can’t be combined with another implicant;
  - => fewest literals for given 1
  - => largest block for given 1
- essential prime implicant (EPI) - PI needed for f
- distinguished 1 - 1 included in just one PI
- cover - group of implicants that include all 1s of f

Terminology Example

- $F(A,B,C) = \Sigma(0,1,4,6)$
- All Prime Implicants:
  - A'B', AC', B'C'
- Distinguished 1s:
  - 1, 6
- Essential PI:
  - A'B', AC'

Terminology Exercise

- Find distinguished 1s, PI and EPI for

$$f(A,B,C) = \Sigma(2,4,5,6)$$

- PI
- distinguished 1s
- EPI

Four Variable Minterms

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

x_1 x_2 x_3 x_4

4 Variable K-Map

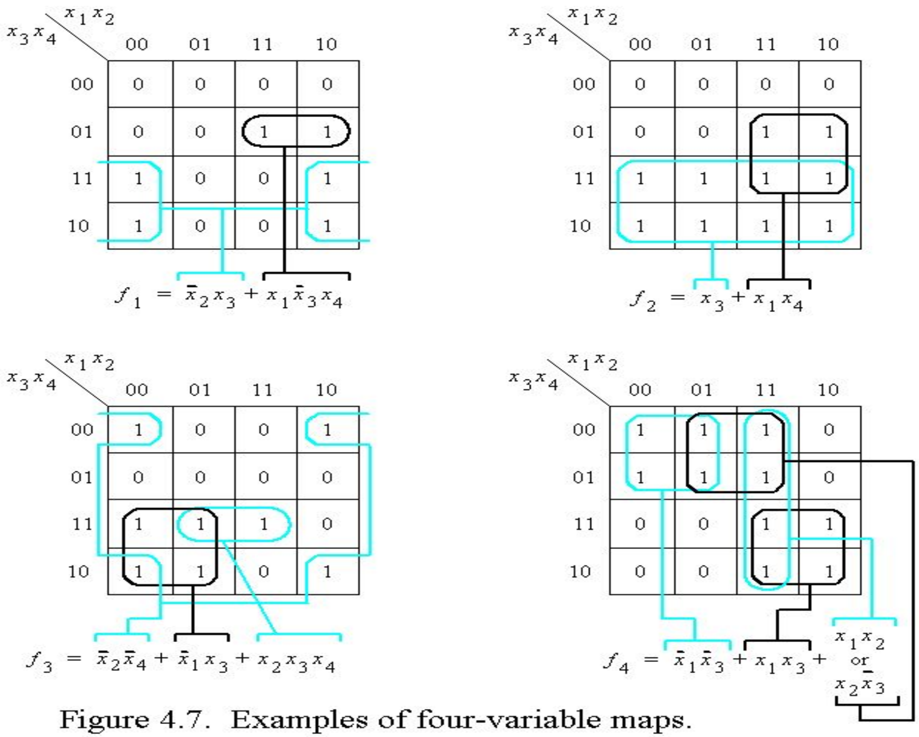


Figure 4.7. Examples of four-variable maps.

Let's Practice A Few

$F(A, B, C, D)=\sum(0,1,3,4,7,13,15)$

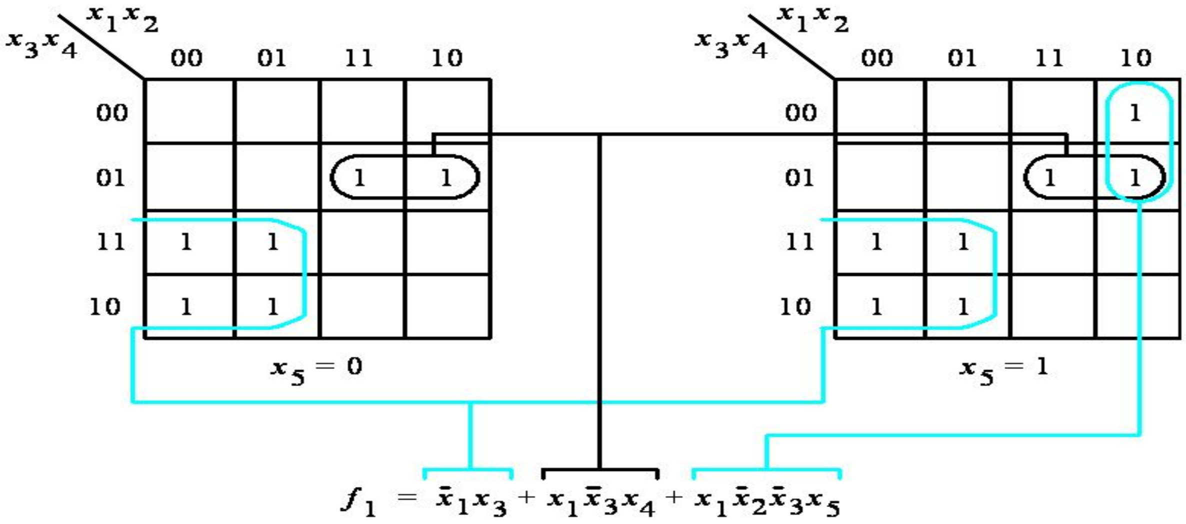
Exercise

$F(A, B, C, D) = \sum(1,2,7,9,10,11,15)$

5 Variable K-Map

- f(A,B,C,D,E)
- Use two 4-variable K-maps for B,C,D,E
  - one for A = 0
  - one for A = 1
- Map for A = 1 has cell numbers 16 higher correspondingly.
- Combine adjacent 1s in 3 dimensions
  - vertically
  - horizontally
  - out of the page

5 Variable K-Map



Let’s Practice One

$F(A, B, C, D, E) = \sum(1,2,7,12,16,18,22,23,26,29,31)$

Six Variable K-Map

- f(A,B,C,D,E,F)
  - Use four 4-variable K-maps for C,D,E,F
    - one for A,B = 0,0
    - one for A,B =0,1
    - one for A,B = 1,0
    - one for A,B = 1,1
  - Maps have corresponding cell numbers from 0,16,32,48
  - Combine adjacent 1s in 3 dimensions
    - vertically
    - horizontally
    - out of the page
- This is evil, we wont go beyond 5 variables

Higher Variable K-Maps

- 5- and 6-variable K-maps are borderline for identifying patterns.
- Meaning we'll only make you do it a little.
- And no 6-variable maps.
- Beyond 6 variables K-maps are not useful.
- Tabular techniques are preferred.

Minimization Procedure

- Map the function f.
- Find all PI, largest blocks first.
- Find distinguished 1s and EPI.
- Start implementing f with EPI.
- Do EPI include all 1s of f?
- If yes, f is done.
- If no, select a minimum number of other PI.

Minimization Guidelines

- Always prefer larger blocks to smaller.
- In 4-variable f, block of 4 has two literals, pair has three.
  - When EPI do not cover f, use best judgment.
  - If in doubt, arbitrarily add another PI and then select from remaining.
- To be sure of minimum cost, try again with each other nonessential PI.

Functions with no EPI

- Can be cyclic
- Arbitrarily start with one PI and then get others as necessary.
- Then try starting with each other PI until you find minimal circuit.
- In other words, “guess and check”.

Let's Do A Bit More Practice

$F(A, B, C, D) = \sum(1,2,9,13,14)$

Let's also note:

- PI
- distinguished 1s
- EPI

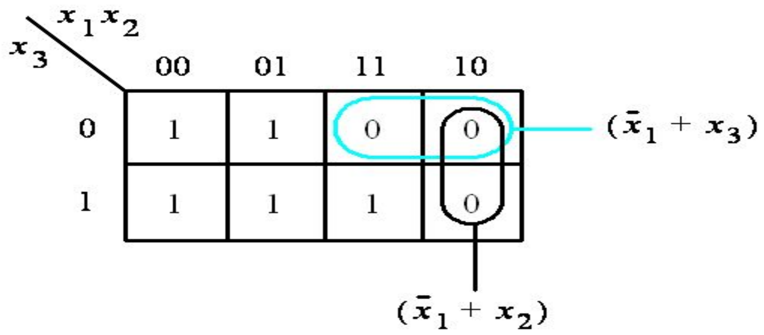
What About POS?

- 0s on POS K-maps are maxterms
  - Therefore, they are labeled the way maxterms are
- if row or column heading for a variable X is
  - 0 => X
  - 1 => X’

POS Minimization

- Just like SOP except works with 0s of f.
- Terms are same except implicates and distinguished 0s.
- Use sum term labeling. (0 => x, 1 => x')
- Map 0s of f.
- Find PI, largest blocks first.
- Find EPI and distinguished 0s.
- Implement f with EPI, adding other PI as necessary for minimal cost.

POS Example



Another POS Example

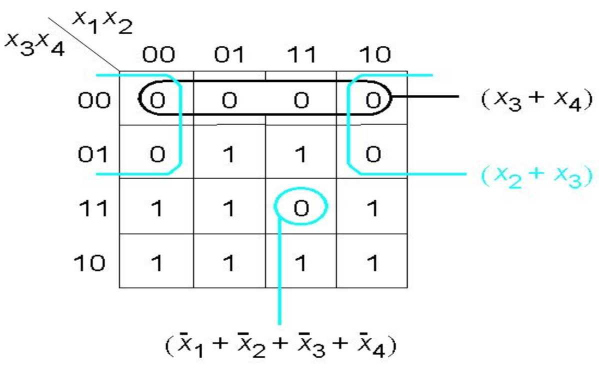


Figure 4.14. POS minimization of $f = \Pi M(0, 1, 4, 8, 9, 12, 15)$.

Let’s Do A Little POS Practice - Example

$F(x,y,z)=\prod(0,1,4,6)$

Exercise

$F(w,x,y,z)=\prod(0,2,7,8,12,15)$

Incompletely Specified Functions

- don't care condition (d)--input condition that either
  - can't occur
  - or doesn't matter if it occurs.
- incompletely specified circuit
  - circuit with one or more don't cares
- Treat don't cares as 1s (for SOP) in forming PI
  - allows forming bigger blocks
- Ignore don't cares in selecting PI.
- Never select a PI only to cover a don't care!!

Example

- $F(A,B,C,D,E) = \Sigma(1,9,10,17,21,27) + d(3,5,11,25,26)$

Exercise

- $F(A,B,C,D) = \Sigma(1,6,7) +$
$$d(3,5,13,15)$$

Let’s Review Gate Diagrams Using K-Maps To Get the Minimized Form

$F(A,B,C,D)=\sum(0,1,4,6,8,10,12,14,15)$

Let’s Expand Out Our Logic Gate Collection

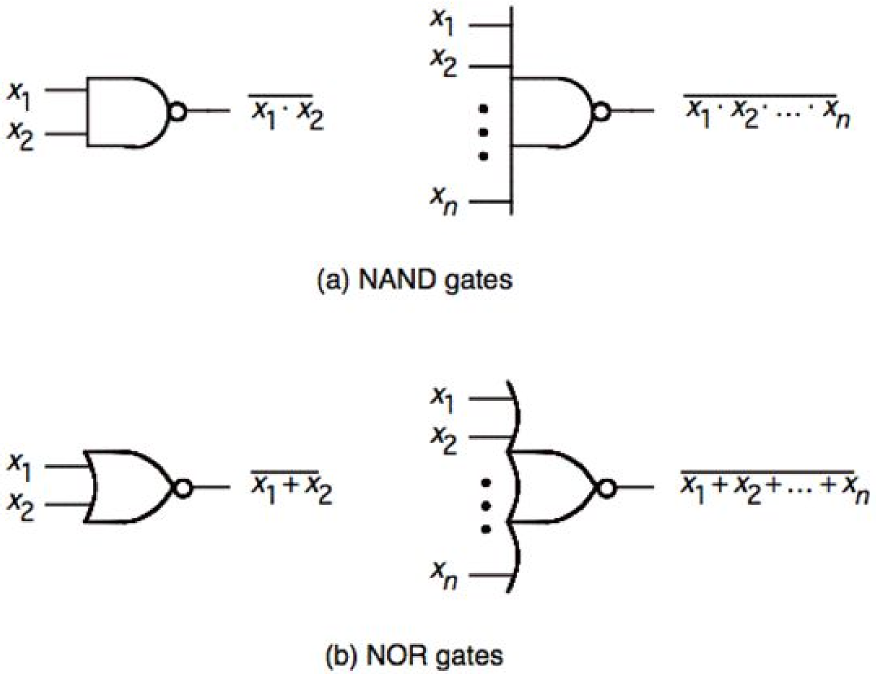
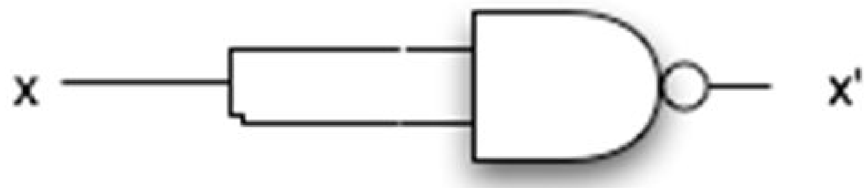
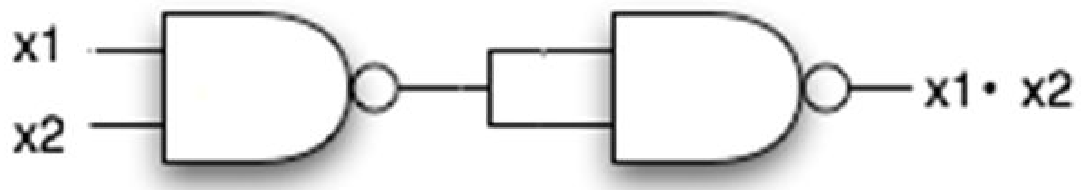


Figure 2.20. NAND and NOR gates.

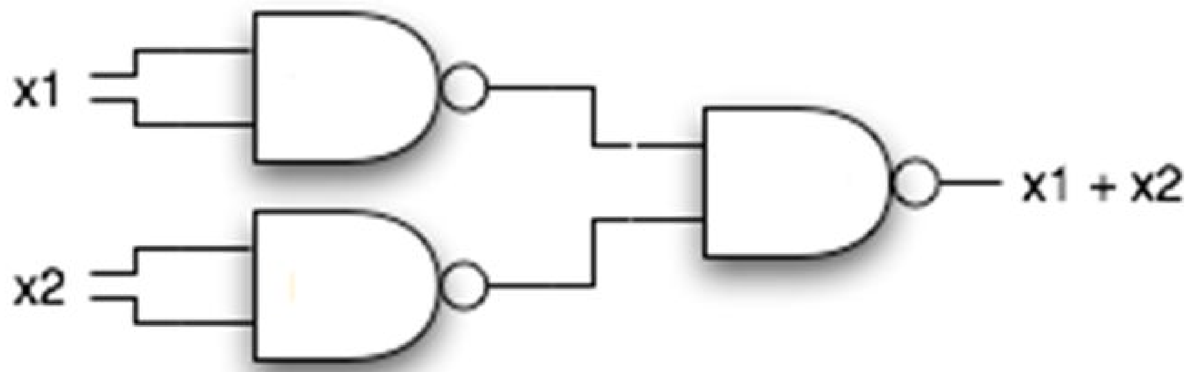
- You can create any combinational function in two levels of logic using AND and OR and NOT.
- You can do all of the same functions using only NAND or NOR gates.



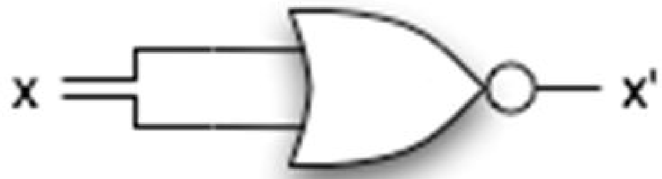
Replacing a NOT with NAND



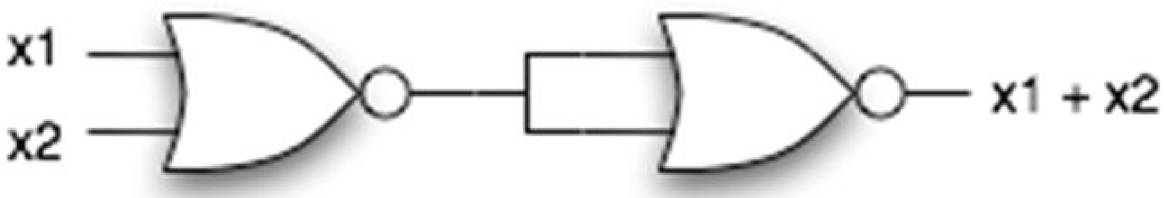
Replacing an AND with NAND



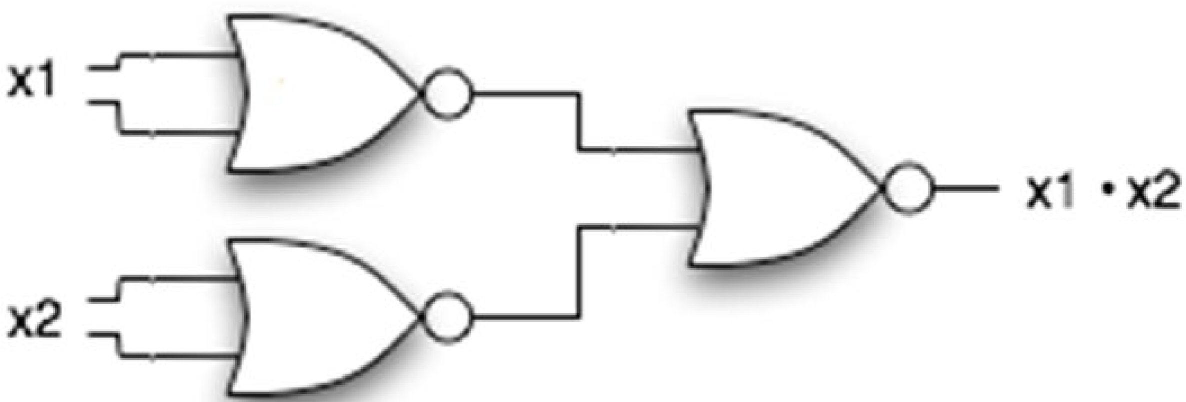
Replacing an OR with NAND



Replacing a NOT with NOR

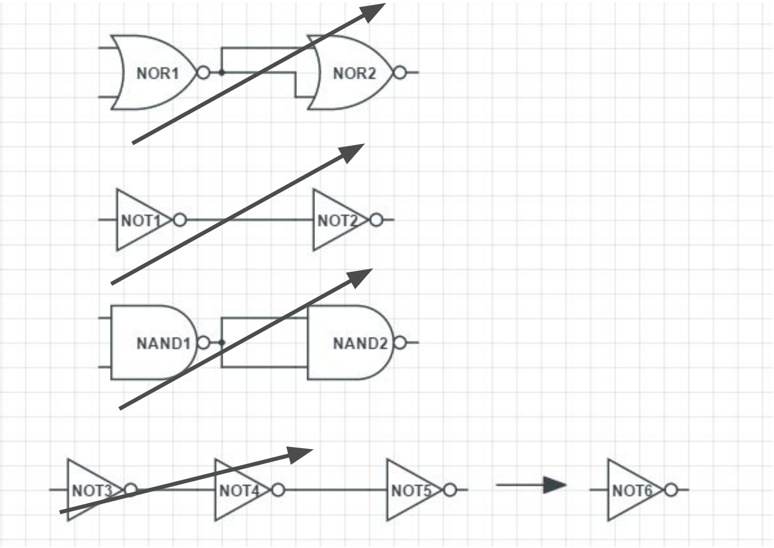


Replacing an OR with NOR



Replacing an AND with NOR

Reduction on Logic Diagrams



Let’s Take That Last Example And...

Implement it with only 2 Input NAND Gates

Implement it with only 2 Input NOR Gates

Exercise

Use K-Map to reduce and implement with only

- 2 input NAND Gates
- 2 input NOR Gates

$F(X, Y, Z)=\sum(1,2,3,7)$

What Is Next

- Your Exam- This is the end of material for the exam
  - Monday
    - First half of class we will do new stuff
    - Second half of class we will do practice exam
      - This is done in small groups
      - Open resource
      - For credit
- Then What?
  - Putting this to use- Combinational Logic Circuits