CS M51A
Logic Design of Digital Systems
Winter 2021

Some slides borrowed and modified from:
M.D. Ercegovac, T. Lang and J. Moreno, Introduction to Digital Systems.

Two-Level Systems

- TWO TYPES OF TWO-LEVEL NETWORKS:

AND-OR **NETWORK** $\Leftrightarrow$ SUM OF PRODUCTS (NAND-NAND NETWORK)

OR-AND **NETWORK** $\Leftrightarrow$ PRODUCT OF SUMS (NOR-NOR NETWORK)

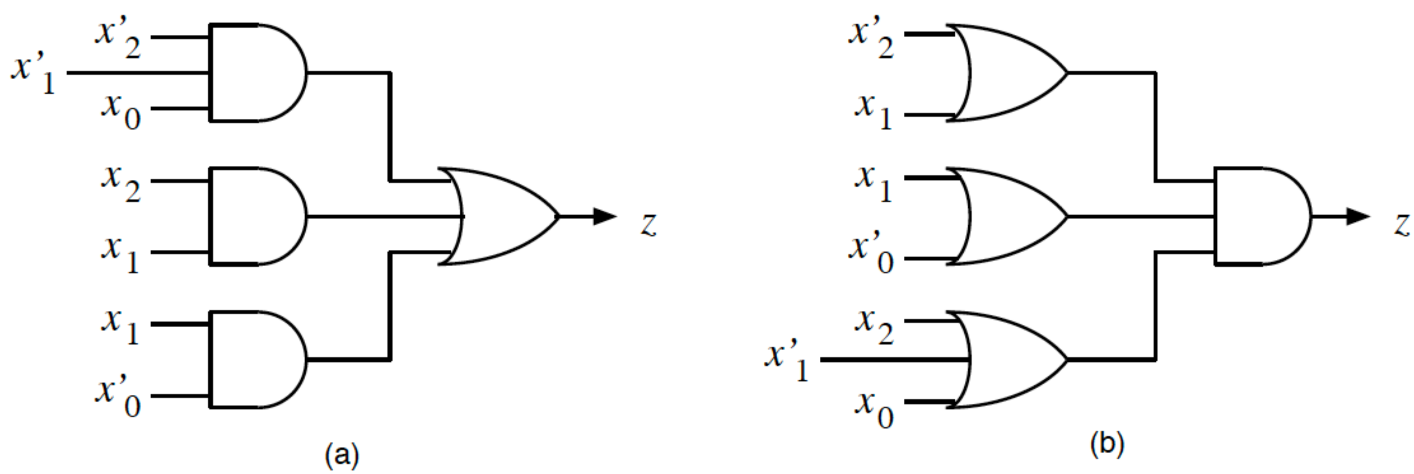


Figure 5.2: AND-OR and OR-AND NETWORKS.

$$E(x_2, x_1, x_0) = x_2'x_1'x_0 + x_2x_1 + x_1x_0'$$

$$E(x_2, x_1, x_0) = (x_2' + x_1)(x_1 + x_0')(x_2 + x_1' + x_0)$$

MINIMAL TWO-LEVEL NETWORKS

1. INPUTS: UNCOMPLEMENTED AND COMPLEMENTED

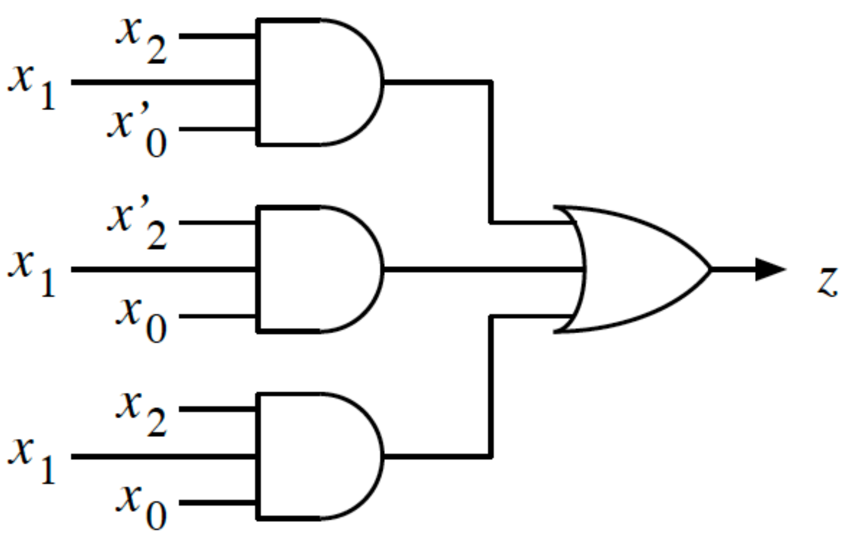
2. FANIN UNLIMITED

3. SINGLE-OUTPUT NETWORKS

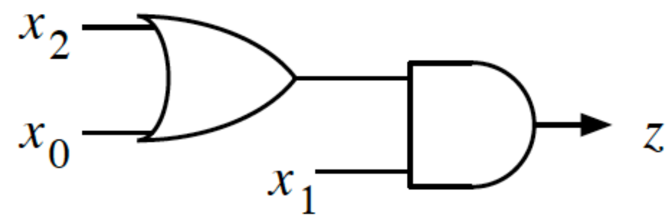
4. MINIMAL NETWORK:

   MINIMUM NUMBER OF GATES WITH MINIMUM NUMBER OF INPUTS

NETWORKS WITH DIFFERENT COST



Network A



Network B

EQUIVALENT BUT DIFFERENT COST

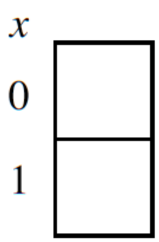
BOTH MINIMAL SP AND PS MUST BE OBTAINED AND COMPARED

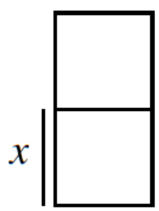
kARNAUGH MAPS

- 2-DIMENSIONAL ARRAY OF CELLS
- n VARIABLES $\longrightarrow 2^n$ CELLS

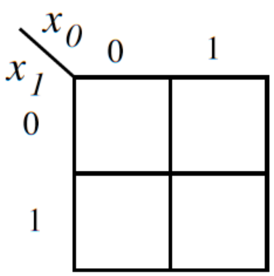
- REPRESENTING SWITCHING FUNCTIONS
- REPRESENTING SWITCHING EXPRESSIONS
- GRAPHICAL AID IN SIMPLIFYING EXPRESSIONS

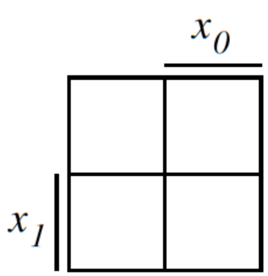
K-Map with one variable





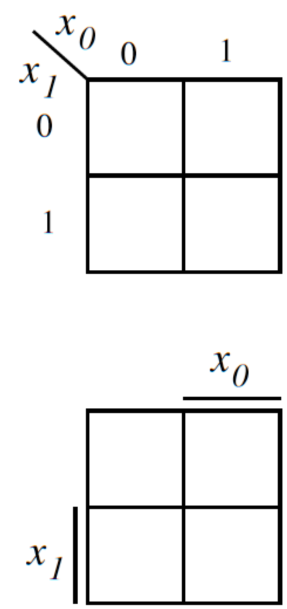
K-Map with two variables





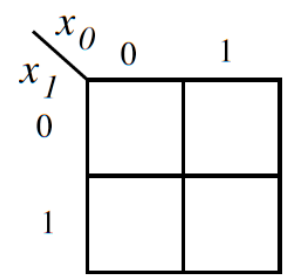
K-Map with two variables

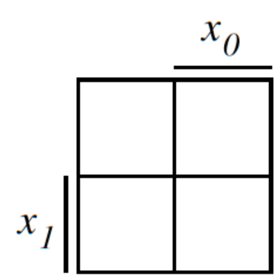
$$F = x_1 x'_0 + x'_1 x'_0$$



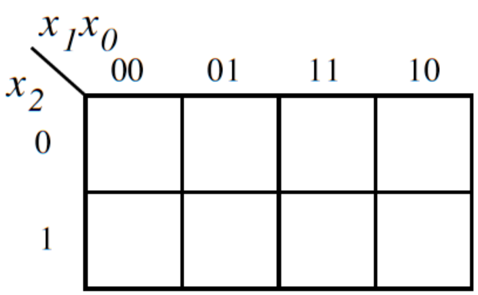
K-Map with two variables

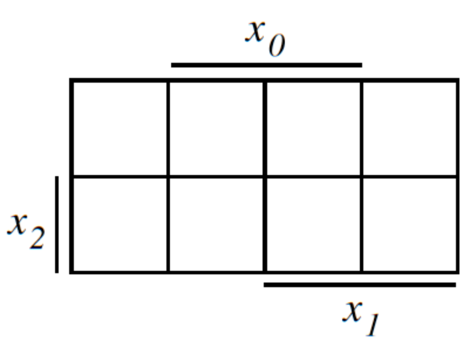
$$F=x_1x_0+x'_1x'_0$$





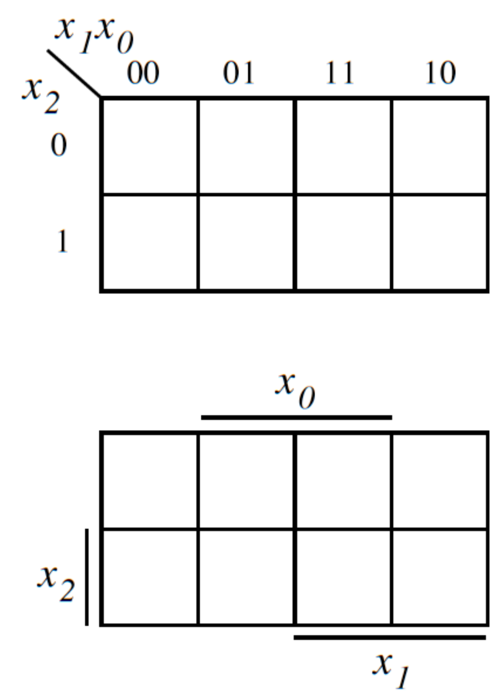
K-Map with three variables



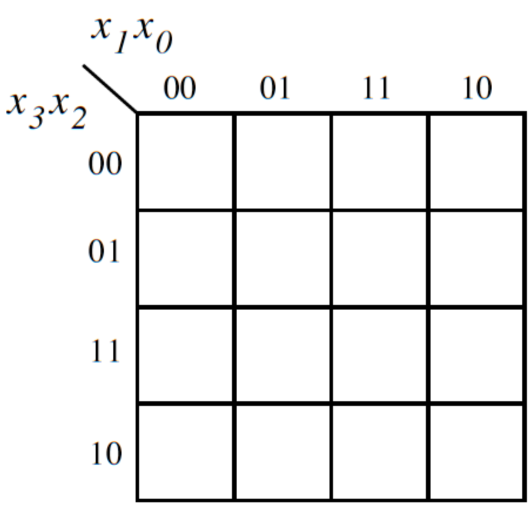


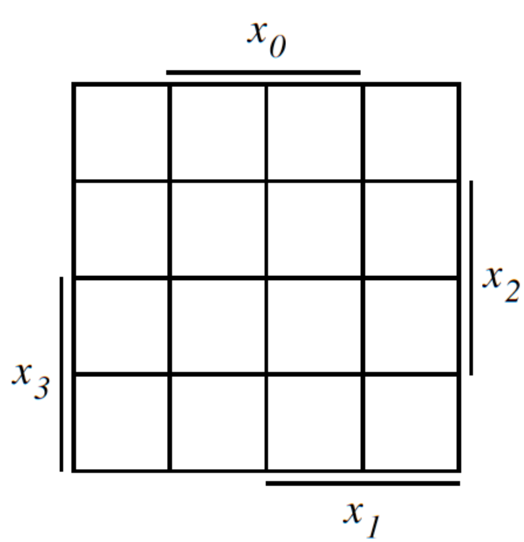
K-Map with three variables

$$F=x_2x_1x_0+x_2x_1x'_0+x_2x'_1x'_0$$



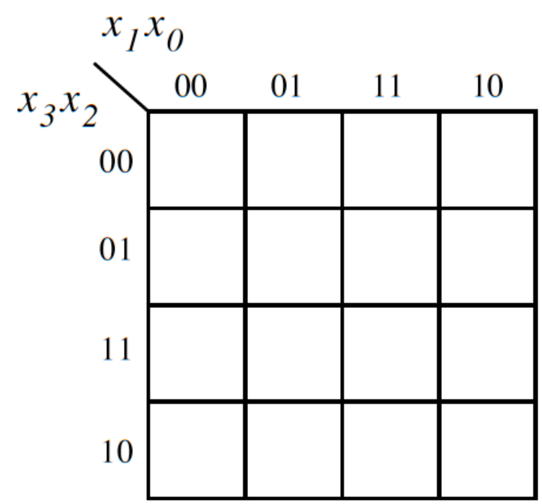
K-Map with four variables

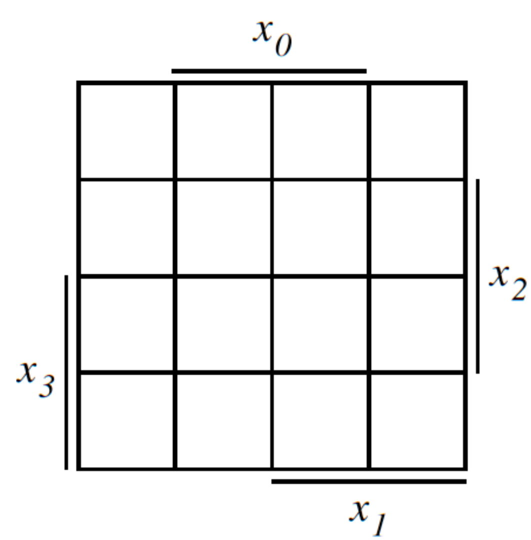




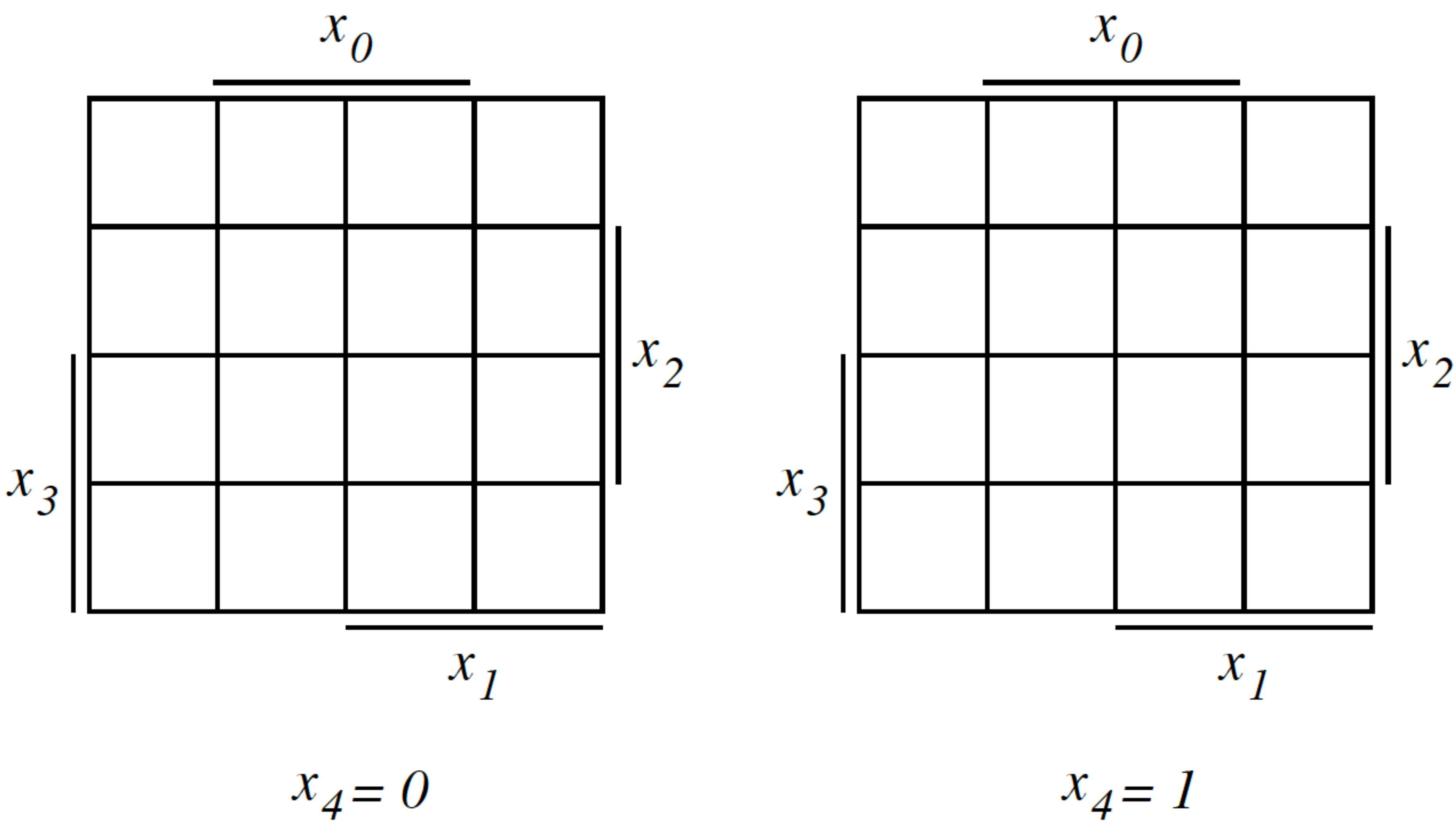
K-Map with four variables

$$F = x_3x_2x_1x_0 + x'_3x_2x_1x'_0 + x_3x_2x'_1x'_0$$





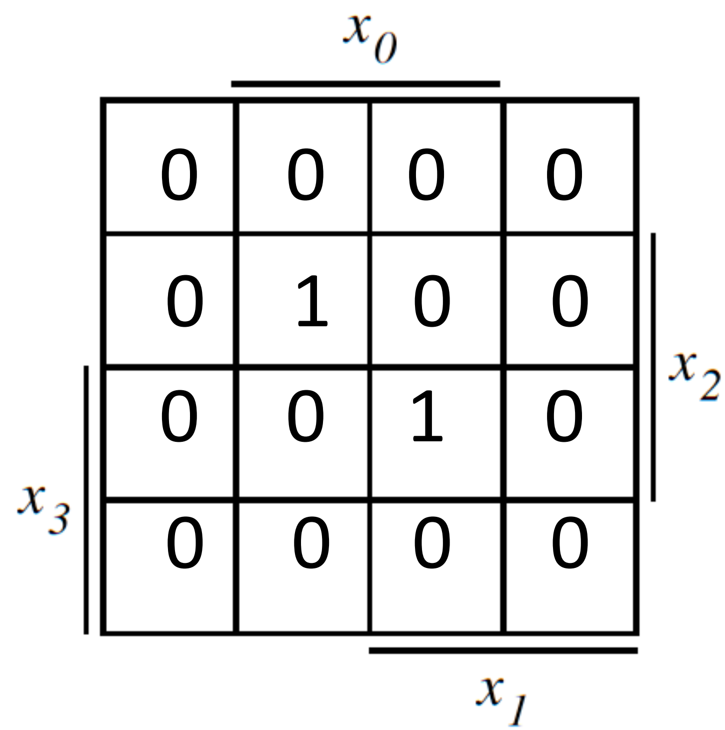
K-Map with five variables



Clicker Question

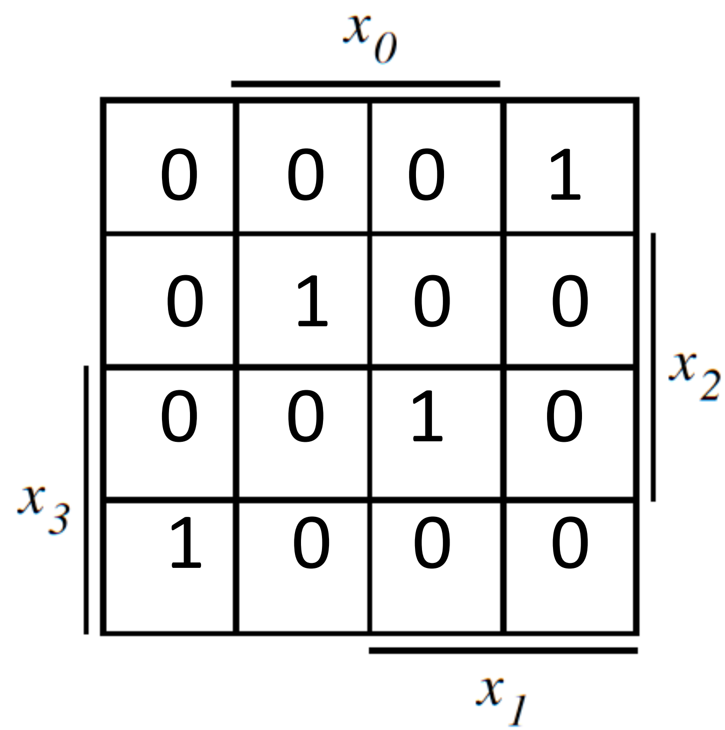
Which Expression does present the k-map?

a) $F=x_3x'_2x'_1x_0+x'_3x_2x_1x'_0$
b) $F=x_3x_2x_1x_0+x_3x_2x_1x'_0$
c) $F=x_3x_2x_1x_0+x'_3x_2x'_1x'_0$
d) $F=x_3x_2x_1x'_0+x'_3x_2x_1x'_0$
e) none

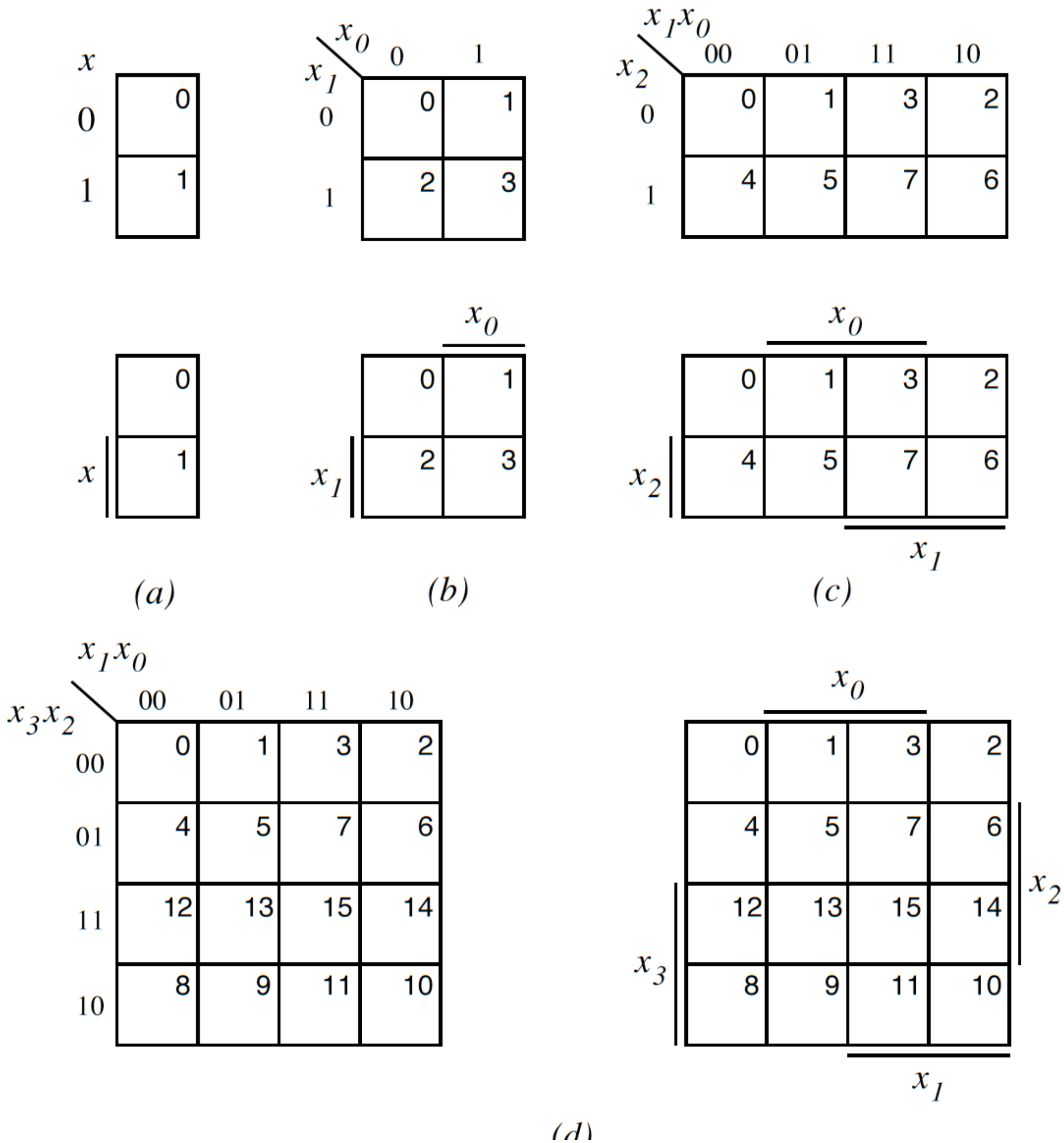


Presenting switching Function (SOP) using K-Map

F=



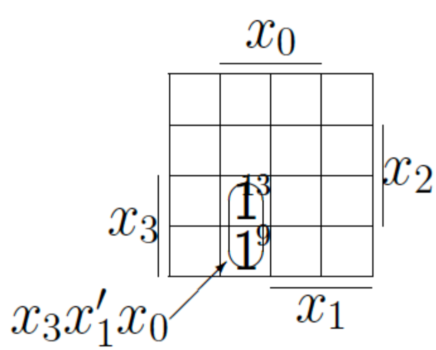
Indexing K-Map



Simplifying Sum of Products

1. MINTERM m_j CORRESPONDS TO 1-CELL WITH LABEL j.
2. PRODUCT TERM OF $n-1$ LITERALS $\longleftrightarrow$ RECTANGLE OF TWO ADJACENT 1-CELLS

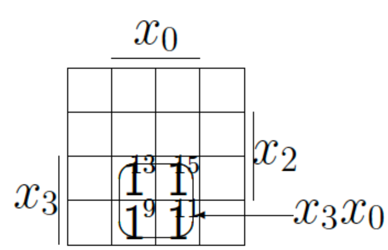
$$x_3x_2x_1'x_0 + x_3x_2'x_1'x_0$$
$$= m_{13} + m_9$$



Simplifying Sum of Products

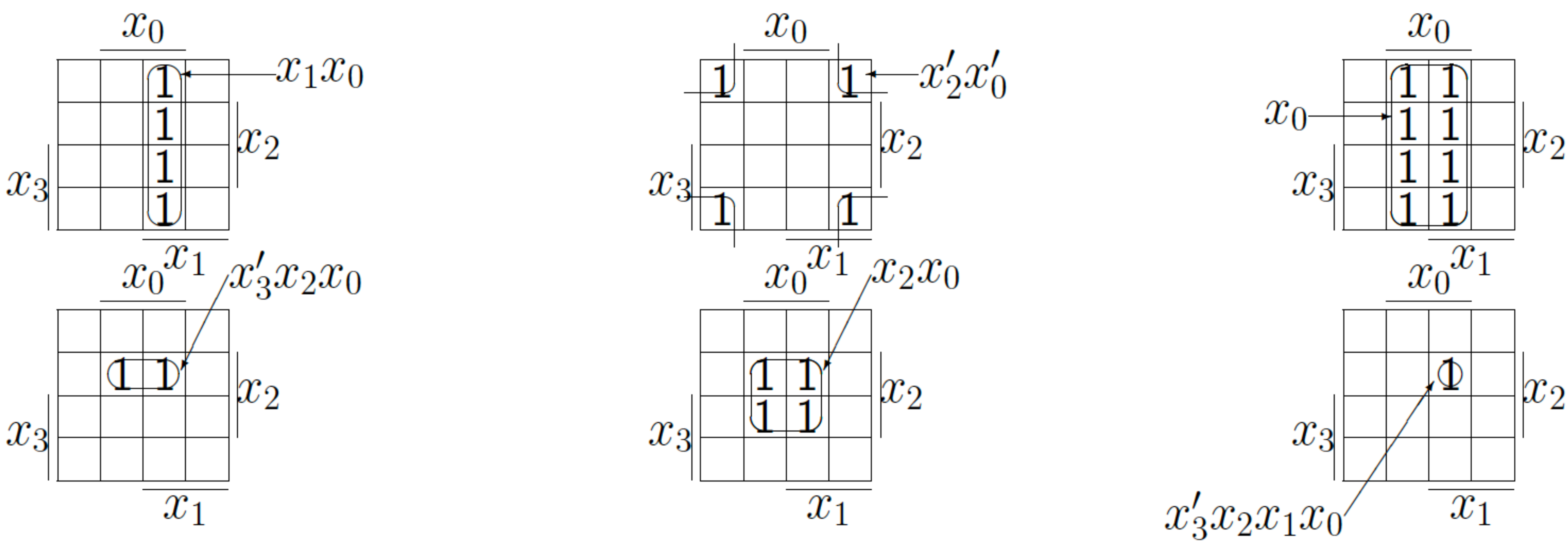
3. PRODUCT TERM OF $n-2$ LITERALS $\longleftrightarrow$ RECTANGLE OF FOUR ADJACENT 1-CELLS

$$x_3x_2'x_1'x_0 + x_3x_2'x_1x_0 + x_3x_2x_1'x_0 + x_3x_2x_1x_0$$
$$= m_9 + m_{11} + m_{13} + m_{15}$$



4. PRODUCT TERM OF $n-s$ LITERALS $\longleftrightarrow$ RECTANGLE OF 2^s ADJACENT 1-CELLS

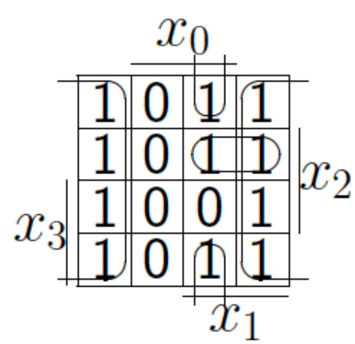
Simplifying Sum of Products



SUM OF PRODUCTS

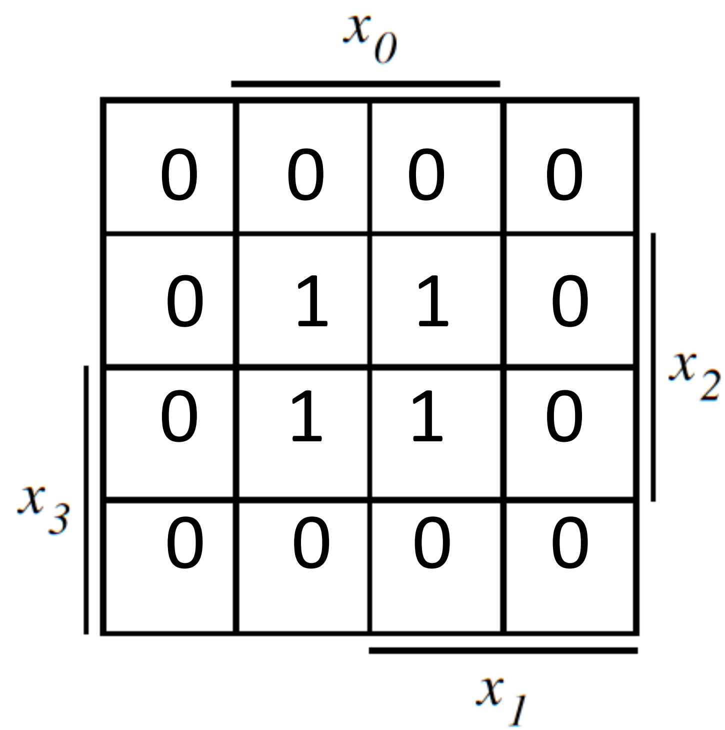
represented in a K-map by the union of rectangles

$$E(x_3, x_2, x_1, x_0) = x_3' x_2 x_1 + x_2' x_1 x_0 + x_0'$$



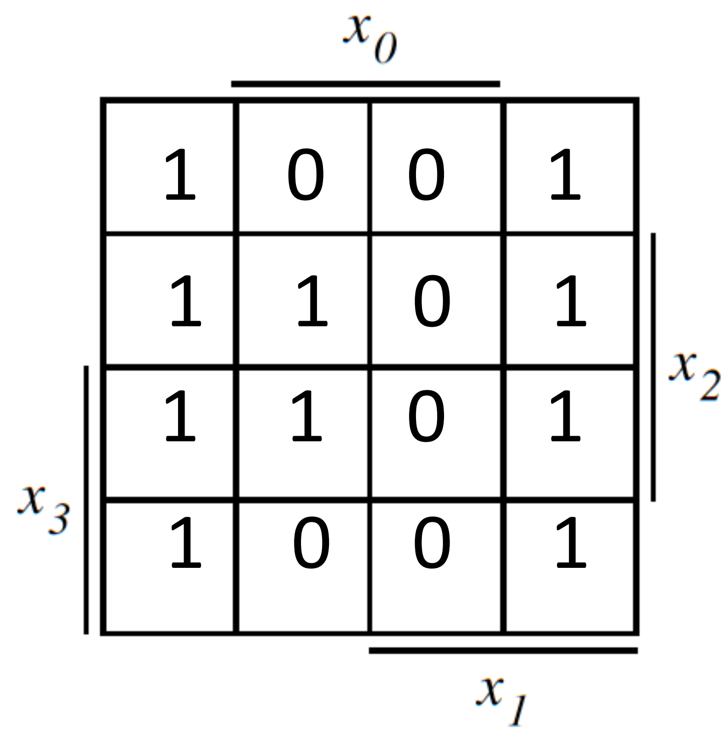
Simplifying Sum of Products - Examples

F=?



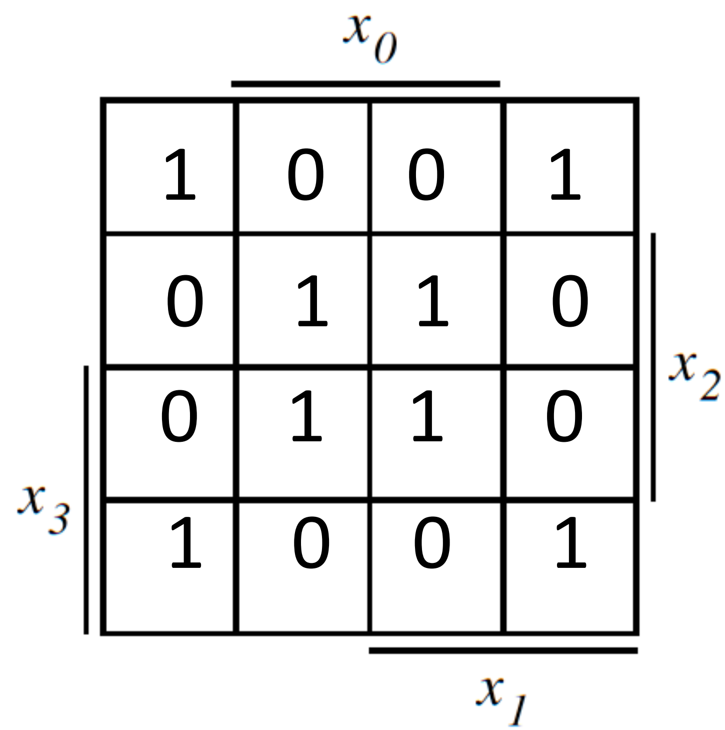
Simplifying Sum of Products - Examples

F=?



Simplifying Sum of Products - Examples

F=?

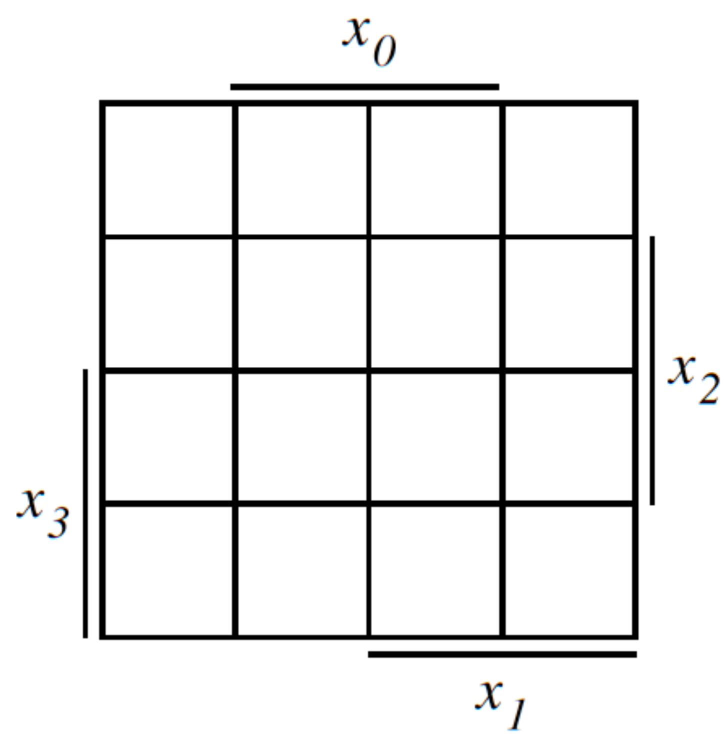


Clicker Question

Simplify this expression using k-map

$$F=x_3x_2x'_1x_0+x_3x_2x_1x'_0+x_3x_2x_1x_0+x_3x_2x'_1x'_0$$

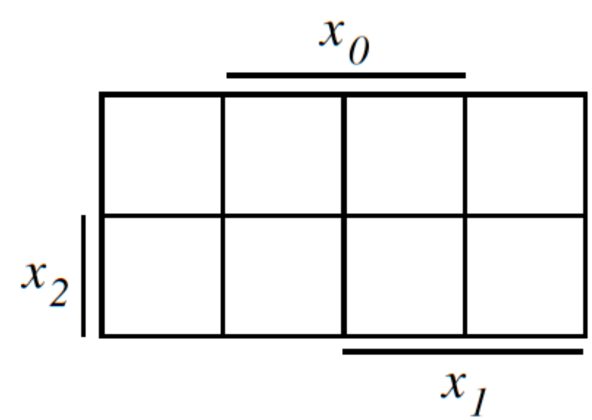
a) $F=x_3x'_2x'_1x_0+x'_3x_2x_1x'_0$
b) $F=x_3x_2x_1x_0+x_3x_2x'_1x'_0$
c) $F=x_3x_2$
d) $F=x_3x_2x_1$
e) none



Simplify this expression using k-map

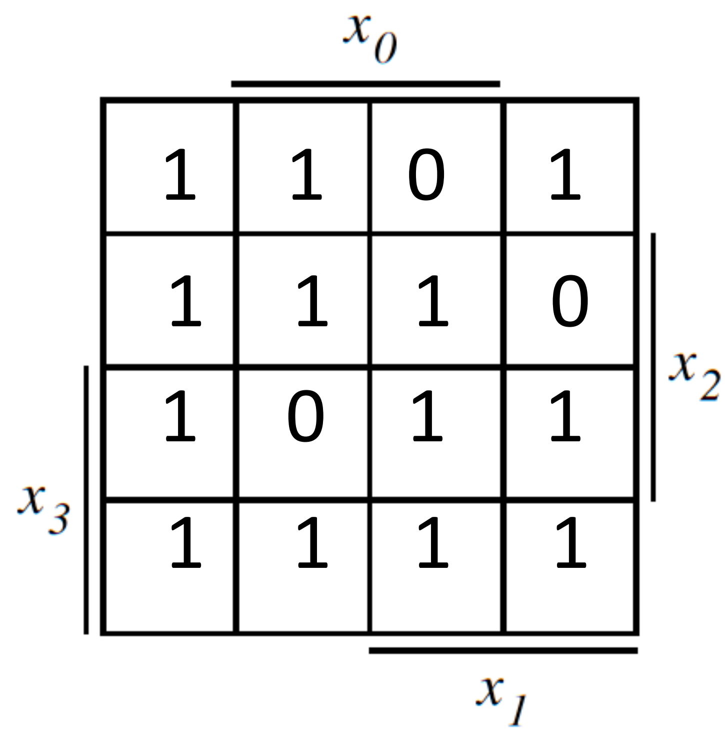
$$F = x_2x'_1x'_0 + x_2x_1x'_0 + x_1x'_0$$

a) $F = x'_2x'_1x_0 + x_2x_1x'_0$
b) $F = x_2x'_0 + x_1x'_0$
c) $F = x_1x_2$
d) $F = x_0x_2x_1$
e) none



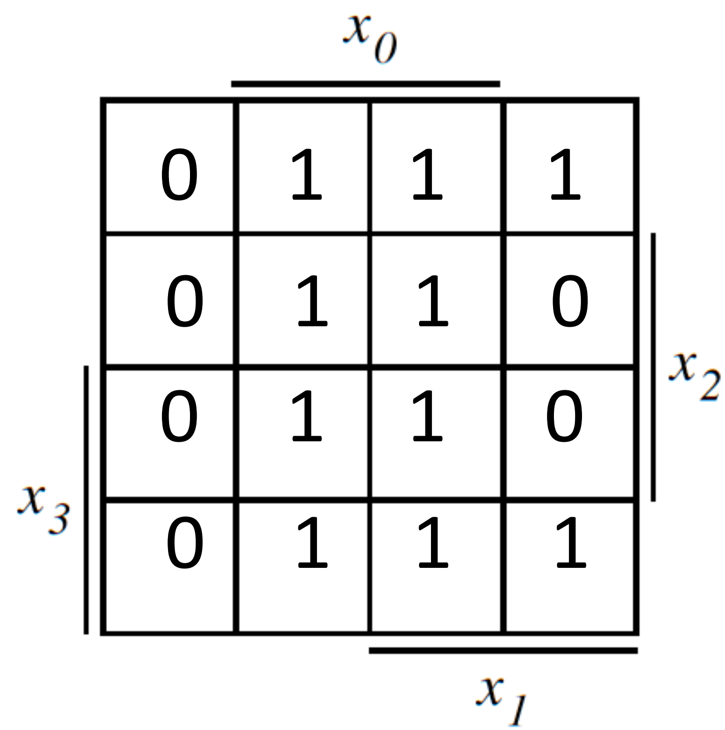
Presenting switching Function (POS) using K-Map

F=



Simplifying PRODUCT of SUMs - Examples

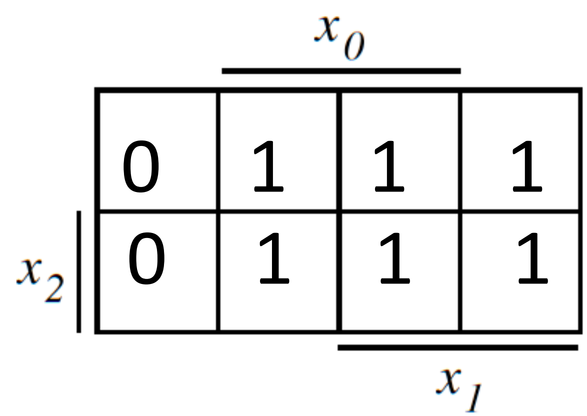
F=?



Clicker Question

Which one is the simplest correct expression?

a) $F=x'_2x'_1x_0+x_2x_1x'_0$
b) $F=x_2x_1x_0+x_2x'_1x'_0$
c) $F=x_1+x_0$
d) $F=x_2+x_1$
e) none



MINIMAL TWO-LEVEL GATE NETWORK DESIGN: EXAMPLE

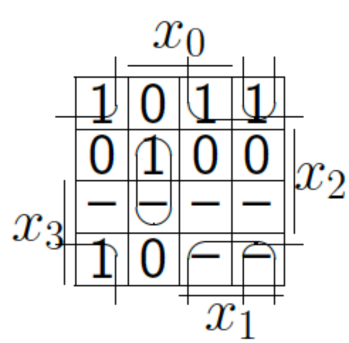
Input: $x \in \{0, 1, 2, ..., 9\}$, coded in BCD as

$$\underline{x} = (x_3, x_2, x_1, x_0), \;\; x_i \in \{0, 1\}$$

Output: $z \in \{0, 1\}$

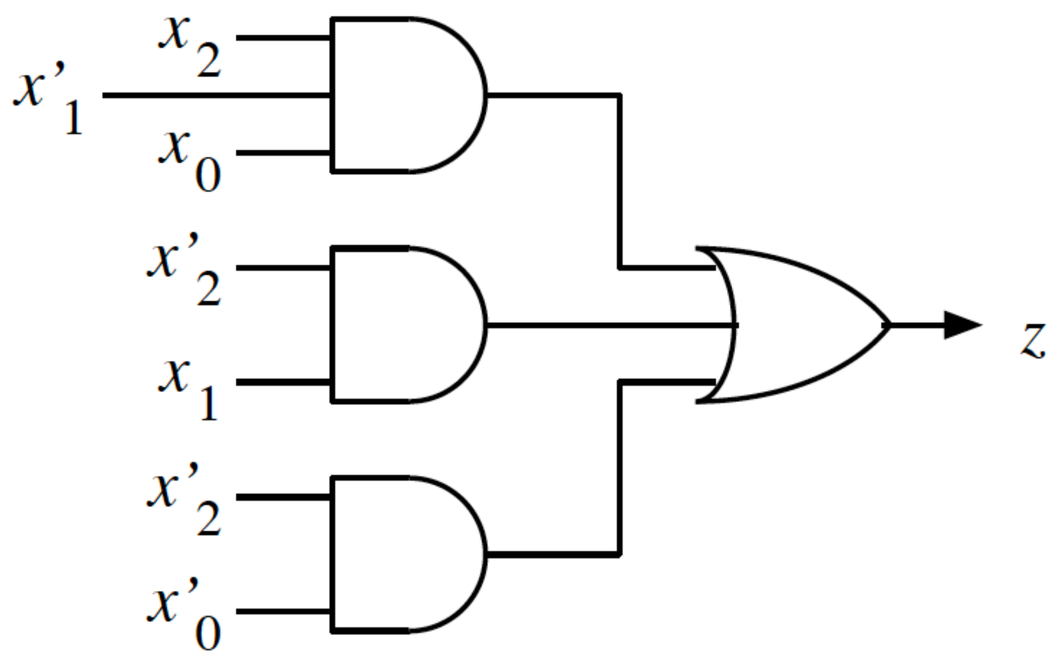
Function: $z = \begin{cases} 1 \;\; \textbf{if} \;\; x \in \{0, 2, 3, 5, 8\} \\ 0 \;\; \textbf{otherwise} \end{cases}$

THE VALUES {10,11,12,13,14,15} ARE "DON'T CARES"



MIN SP: $z = x_2' x_1 + x_2' x_0' + x_2 x_1' x_0$

MIN PS: $z = (x_2' + x_1')(x_2' + x_0)(x_2 + x_1 + x_0')$


x_2
x'_1
x_0
x'_2
x_1
x'_2
x'_0
z

EXAMPLE 5.15

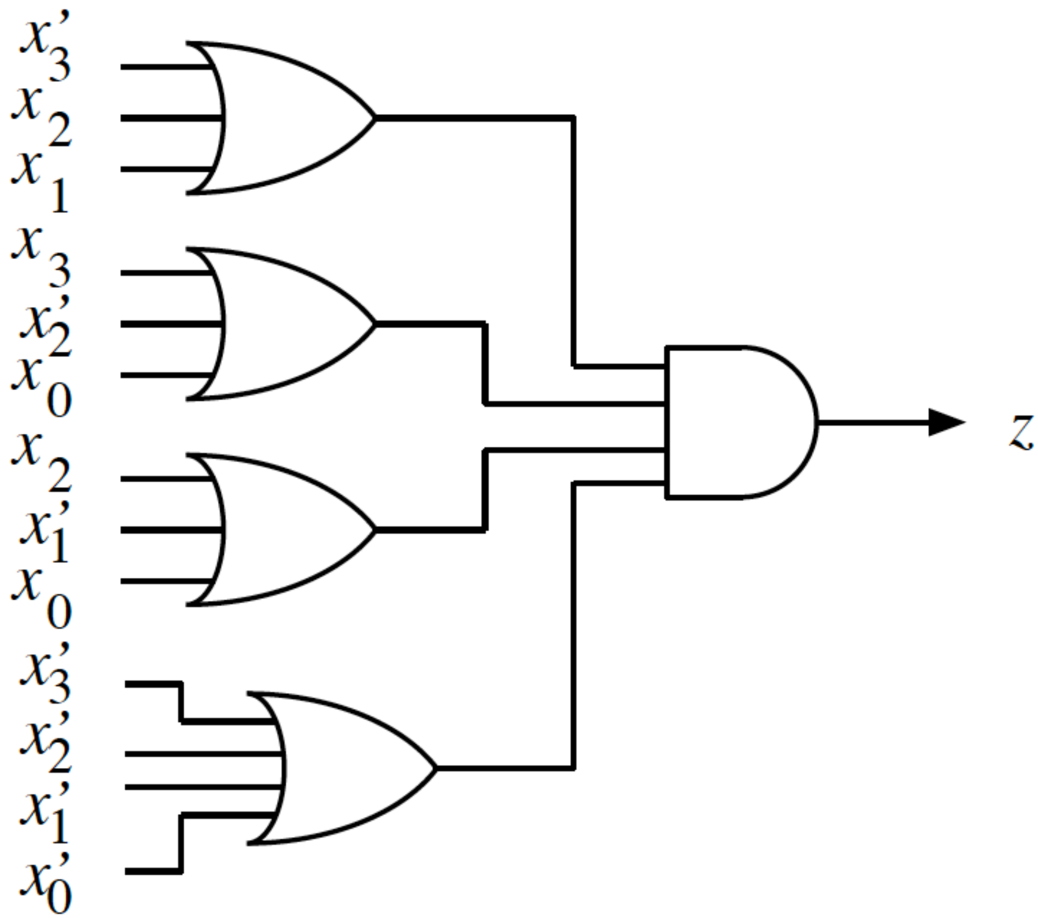
Input: $x \in \{0, 1, 2, ..., 15\}$
represented in binary code by $\underline{x} = (x_3, x_2, x_1, x_0)$
Output: $z \in \{0, 1\}$

Function: $z = \begin{cases} 1 & \textbf{if } x \in \{0, 1, 3, 5, 7, 11, 12, 13, 14\} \\ 0 & \textbf{otherwise} \end{cases}$

THE K-MAP:

		x_0			
	1	1	1	0	
	0	1	1	0	x_2
x_3	1	1	0	1	
	0	0	1	0	
			x_1		

min SP: $z = x_3'x_0 + x_3'x_2'x_1' + x_2x_1'x_0 + x_3x_2x_0' + x_2'x_1x_0$
min PS: $z = (x_3' + x_2 + x_1)(x_3 + x_2' + x_0)(x_2 + x_1' + x_0)(x_3' + x_2' + x_1' + x_0')$
COST(PS) $<$ COST(SP)


x_3'
x_2
x_1
x_3
x_2'
x_0
x_2
x_1'
x_0
x_3'
x_2'
x_1'
x_0'
z