DESIGN OF GATE NETWORKS

- DESIGN OF TWO-LEVEL NETWORKS:
  AND-OR and OR-AND NETWORKS

- MINIMAL TWO-LEVEL NETWORKS
  KARNAUGH MAPS
  MINIMIZATION PROCEDURE AND TOOLS
  LIMITATIONS OF TWO-LEVEL NETWORKS

- DESIGN OF TWO-LEVEL NAND-NAND and
  NOR-NOR NETWORKS

- PROGRAMMABLE LOGIC: PLAs and PALs

DESIGN OF TWO-LEVEL NETWORKS

IMPLEMENTATION:

Level 1: NOT GATES (optional)

Level 2: AND GATES

Level 3: OR GATES

LITERALS
(uncomplemented and complemented variables)
NOT GATES (IF NEEDED)

PRODUCTS: AND gates

SUM: OR gate

MULTIOUTPUT NETWORKS: ONE OR GATE USED FOR EACH OUTPUT

PRODUCT OF SUMS NETWORKS - SIMILAR

MODULO-64 INCREMENTER

Input: $0 \leq x \leq 63$
Output: $0 \leq z \leq 63$
Function: $z = (x+1) \bmod 64$

x	010101	x	001111
z	010110	z	010000

- RADIX-2 REPRESENTATION

$$z_i = \begin{cases} 1 \ \textbf{if} \ \ (x_i = 1 \ \textbf{and} \ \ \textit{there exists } j < i \textit{ such that } x_j = 0) \\ \quad \textbf{or} \ (x_i = 0 \ \textbf{and} \ x_j = 1 \textit{ for all } j < i) \\ 0 \ \textbf{otherwise} \end{cases}$$

$$\begin{aligned} z_5 &= x_5(x_4' + x_3' + x_2' + x_1' + x_0') + x_5' x_4 x_3 x_2 x_1 x_0 \\ &= x_5 x_4' + x_5 x_3' + x_5 x_2' + x_5 x_1' + x_5 x_0' + x_5' x_4 x_3 x_2 x_1 x_0 \\ z_4 &= x_4 x_3' + x_4 x_2' + x_4 x_1' + x_4 x_0' + x_4' x_3 x_2 x_1 x_0 \\ z_3 &= x_3 x_2' + x_3 x_1' + x_3 x_0' + x_3' x_2 x_1 x_0 \end{aligned}$$

$$z_2 = x_2x_1' + x_2x_0' + x_2'x_1x_0$$
$$z_1 = x_1x_0' + x_1'x_0$$
$$z_0 = x_0'$$

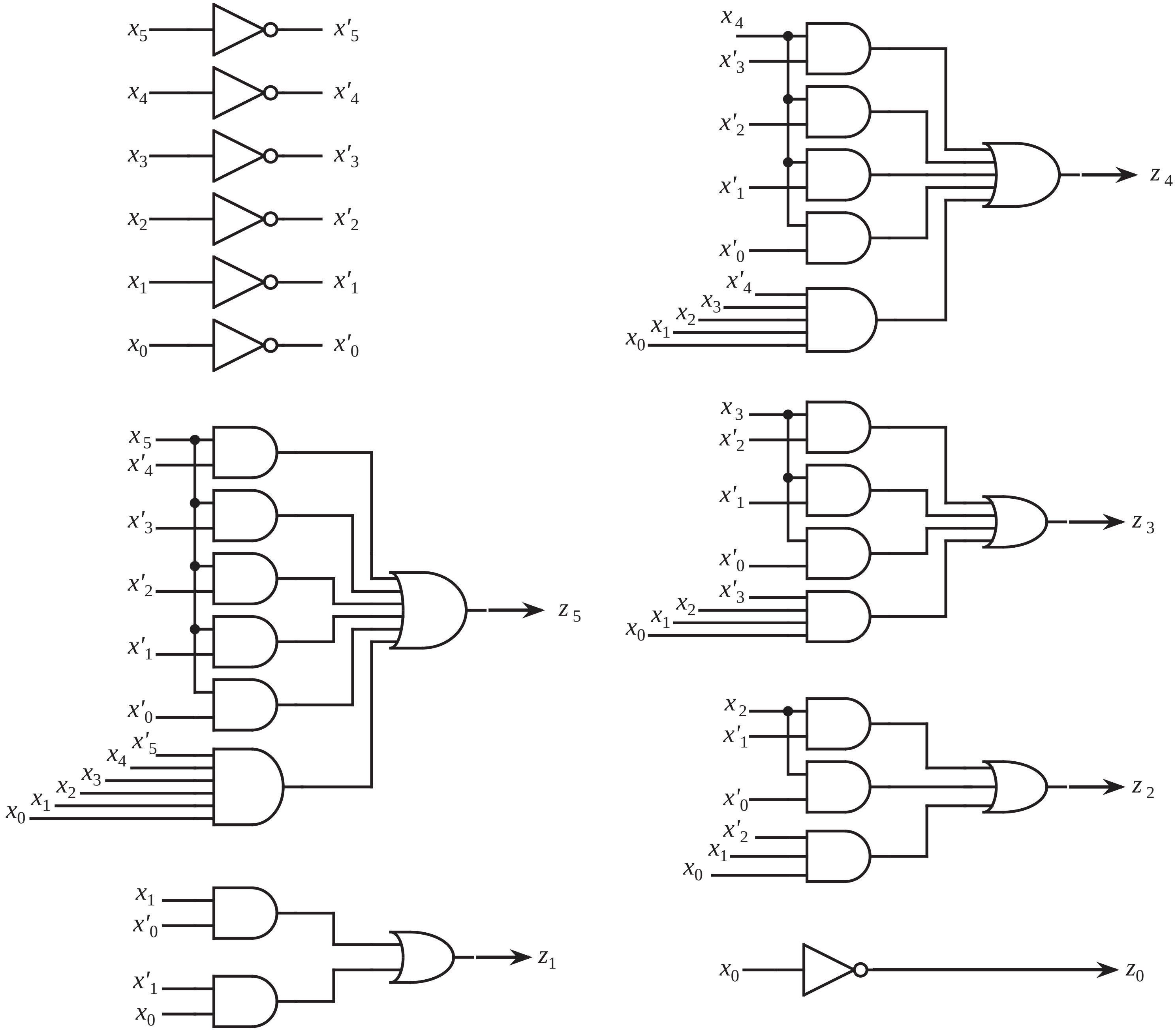


Figure 5.1: NOT-AND-OR MODULO-64 INCREMENTER NETWORK.

UNCOMPLEMENTED AND COMPLEMENTED INPUTS AVAILABLE

- TWO TYPES OF TWO-LEVEL NETWORKS:

AND-OR NETWORK $\Leftrightarrow$ SUM OF PRODUCTS (SP)
easily transformed into NAND-NAND NETWORK

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

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

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

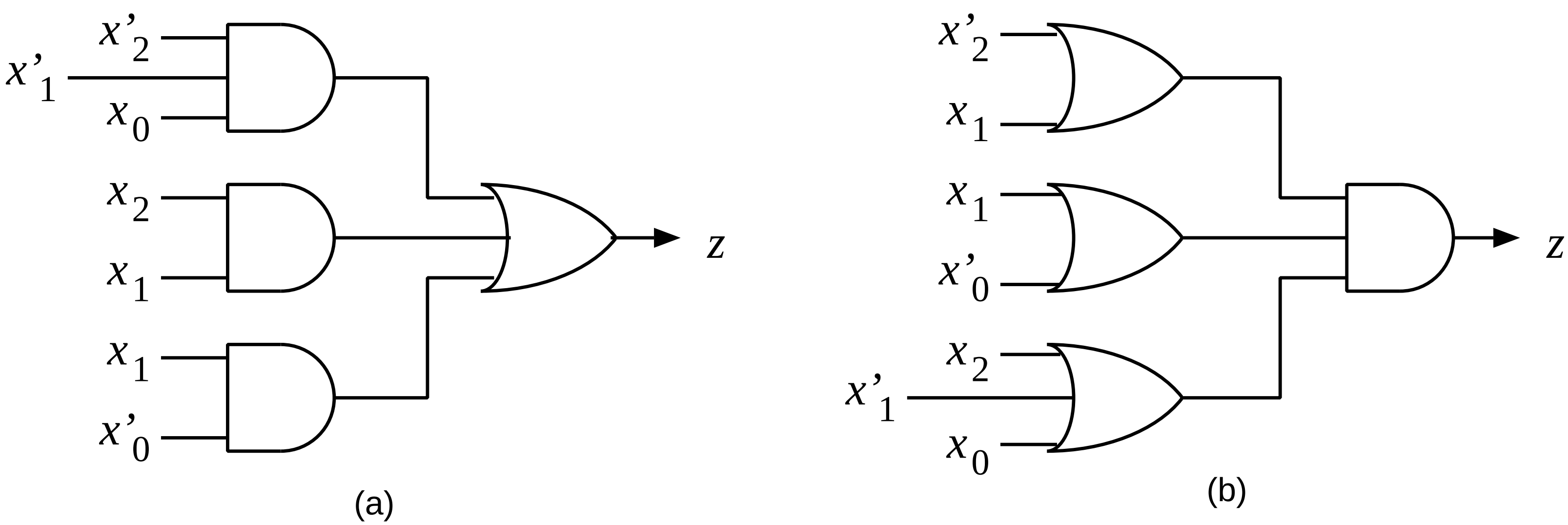


Figure 5.2: AND-OR and OR-AND NETWORKS.

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

   (minimal expression: min. number of terms with min. number of literals)

NETWORKS WITH DIFFERENT COST

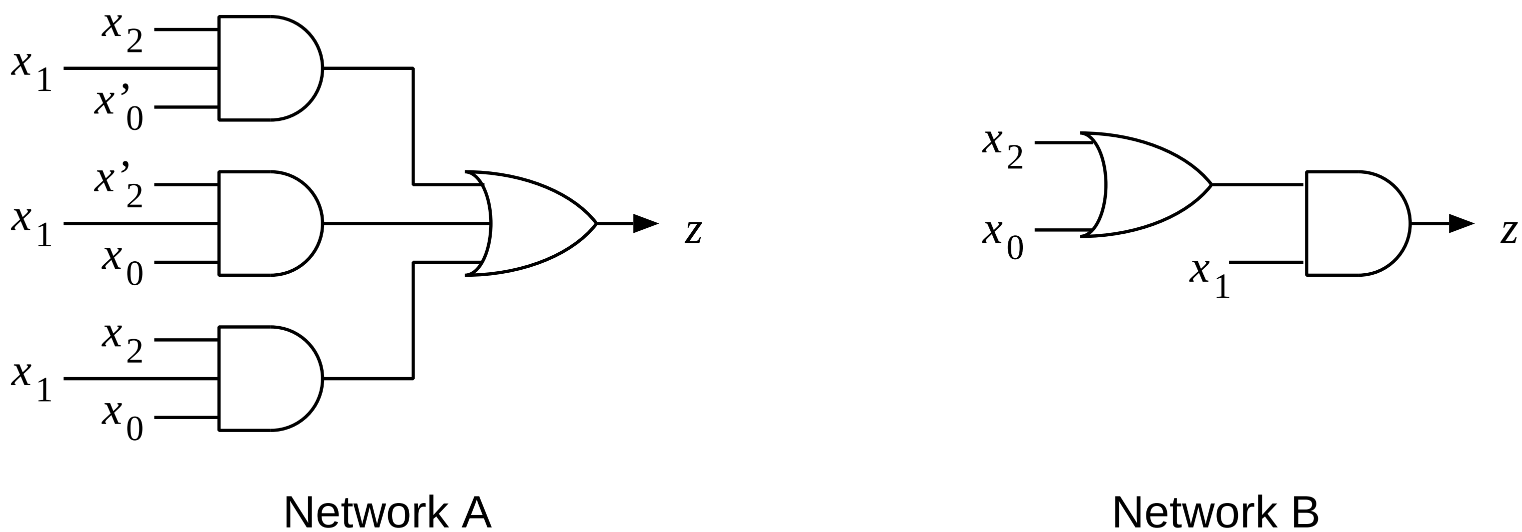


Figure 5.3: NETWORKS WITH DIFFERENT COST TO IMPLEMENT $f(x_2, x_1, x_0) = one\text{-}set(3,6,7)$.

MINIMAL EXPRESSIONS

- EQUIVALENT BUT DIFFERENT COST

$$
\begin{aligned}
E_1(x_2, x_1, x_0) &= x_2' x_1 x_0' + x_1' x_0 + x_2 x_0 \\
E_2(x_2, x_1, x_0) &= x_2 x_1 x_0 + x_2' x_1 x_0' + x_2' x_1' x_0 + x_2 x_1' x_0
\end{aligned}
$$

- BOTH MINIMAL SP AND PS MUST BE OBTAINED AND COMPARED

- BASIS:

$$
\begin{aligned}
ab + ab' &= a \quad \text{(for sum of products)} \\
(a + b)(a + b') &= a \quad \text{(for product of sums)}
\end{aligned}
$$

GRAPHICAL REPRESENTATION OF SWITCHING FUNCTIONS: kARNAUGH MAPS

- 2-DIMENSIONAL ARRAY OF CELLS
- n VARIABLES $\longrightarrow 2^n$ CELLS
- cell $i \longleftrightarrow$ ASSIGNMENT i

  ADJACENCY CONDITION

  ANY SET OF 2^r ADJACENT ROWS (COLUMNS):

  ASSIGNMENTS DIFFER IN r VARIABLES

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

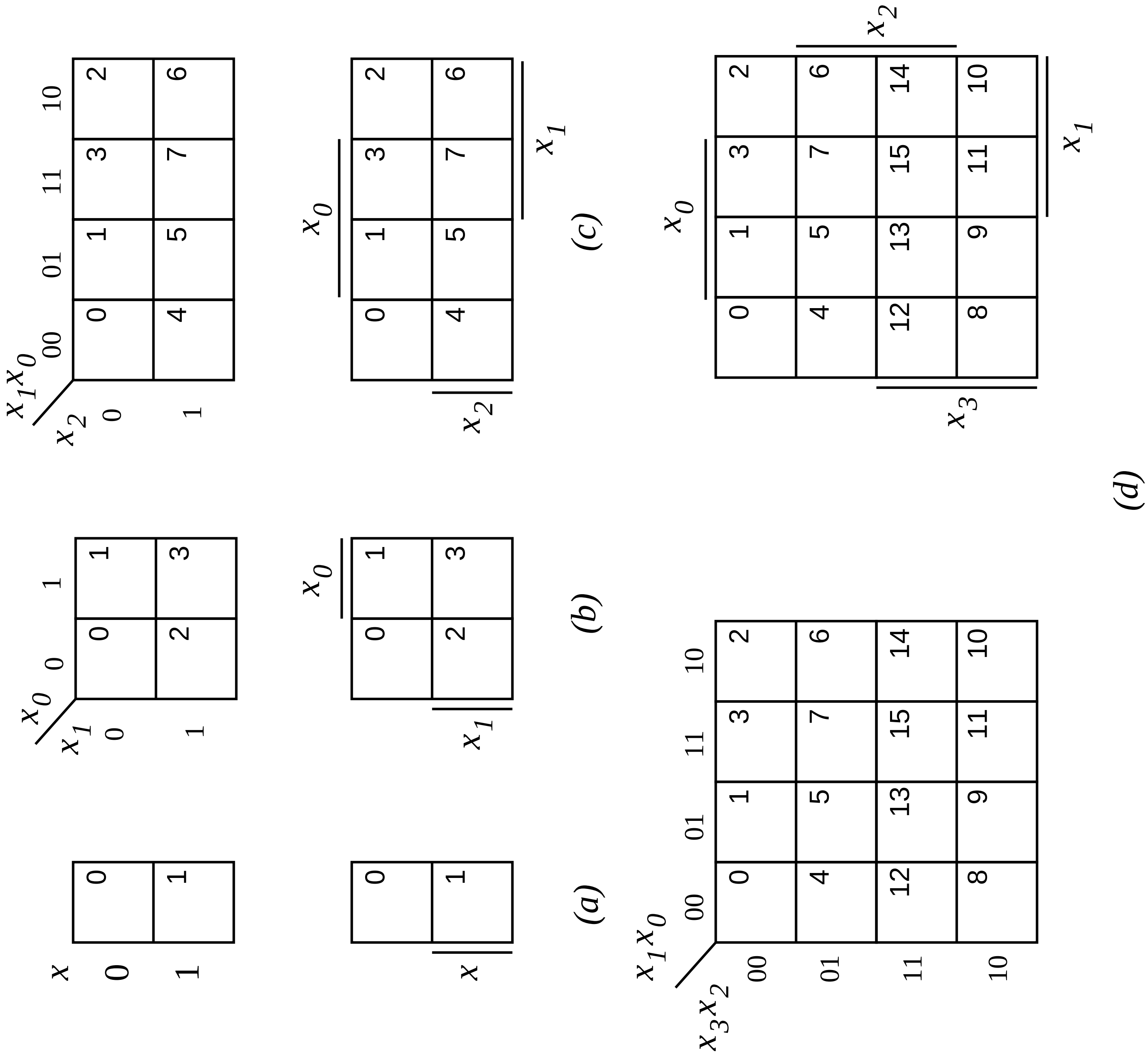


Figure 5.4: K-Maps

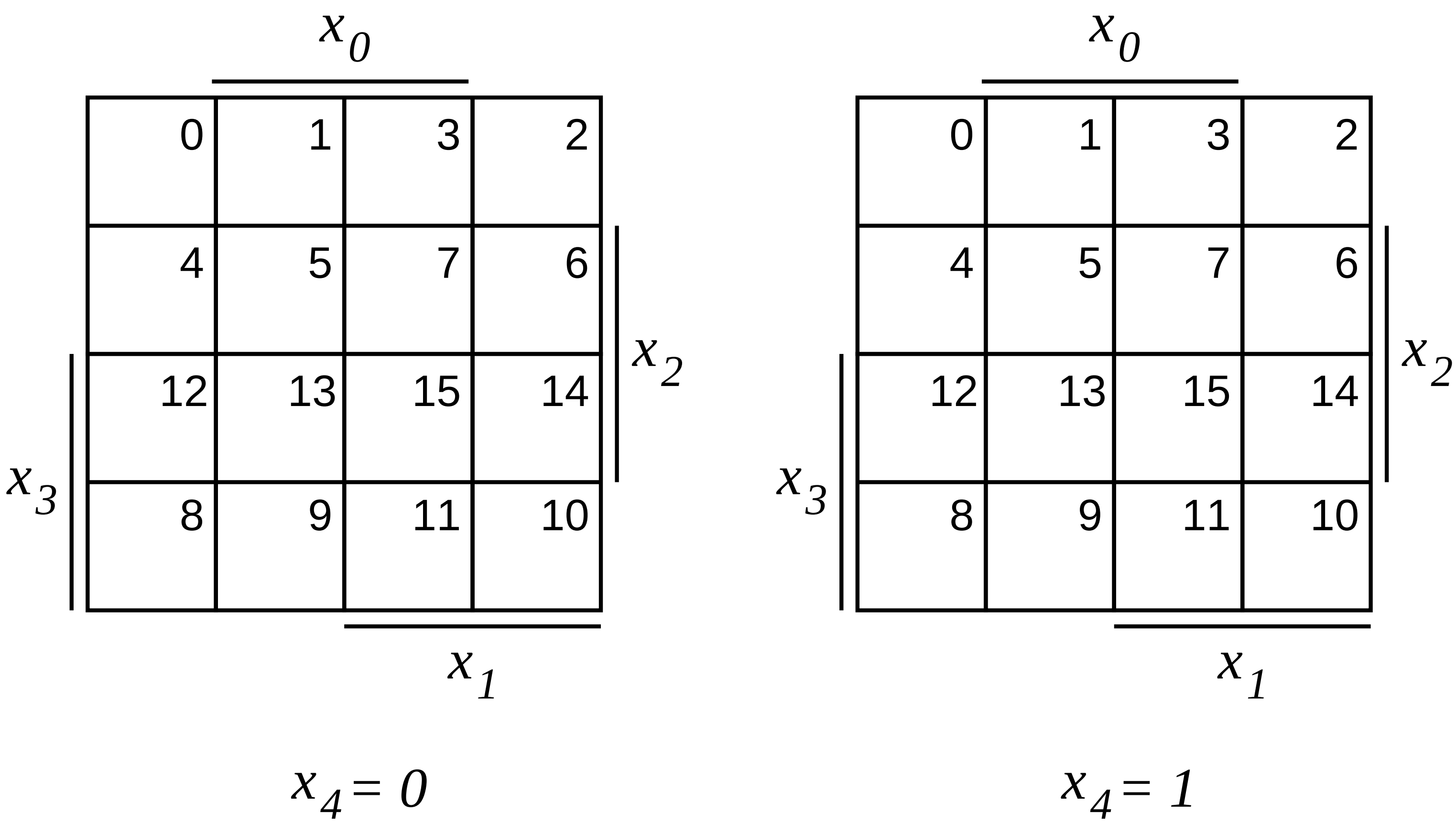


Figure 5.5: K-map FOR FIVE VARIABLES

REPRESENTATION OF SWITCHING FUNCTIONS

$$f(x_2, x_1, x_0) = \textit{one-set}(0,2,6)$$

x_0

	1	0	0	1
x_2	0	0	0	1

x_1

$$f(x_3, x_2, x_1, x_0) = \textit{zero-set}(1,3,4,6,10,11,13)$$

x_0

	1	0	0	1	
	0	1	1	0	x_2
x_3	1	0	1	1	
	1	1	0	0	

x_1

$$f(x_2, x_1, x_0) = [\textit{one-set}(0,4,5), \quad \textit{dc-set}(2,3)]$$

x_0

	1	0	–	–
x_2	1	1	0	0

x_1

RECTANGLES OF 1-CELLS AND 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

$$\begin{aligned} x_3x_1'x_0 &= x_3x_1'x_0(x_2 + x_2') \\ &= x_3x_2x_1'x_0 + x_3x_2'x_1'x_0 \\ &= m_{13} + m_9 \end{aligned}$$

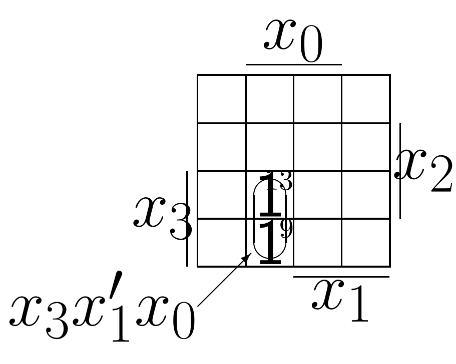


Figure 5.6

RECTANGLES OF 1-CELLS AND SUM OF PRODUCTS (cont.)

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

$$\begin{aligned} x_3x_0 &= x_3x_0(x_1 + x_1')(x_2 + x_2') \\ &= x_3x_2'x_1'x_0 + x_3x_2'x_1x_0 + x_3x_2x_1'x_0 + x_3x_2x_1x_0 \\ &= m_9 + m_{11} + m_{13} + m_{15} \end{aligned}$$

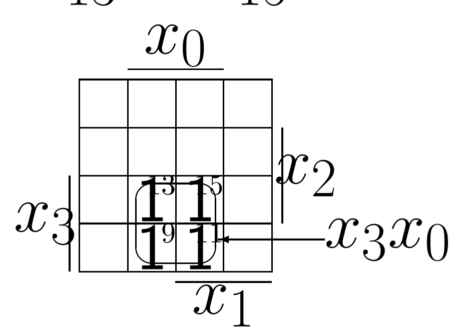


Figure 5.6

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

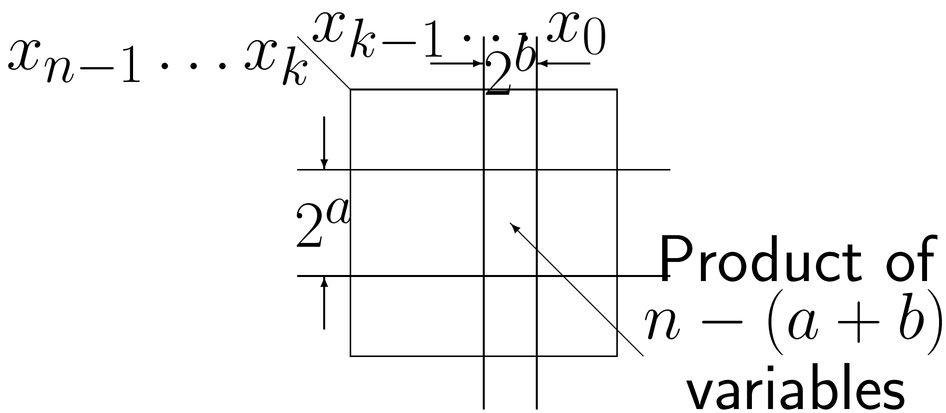


Figure 5.7: Representation of product of $n-(a+b)$ variables.

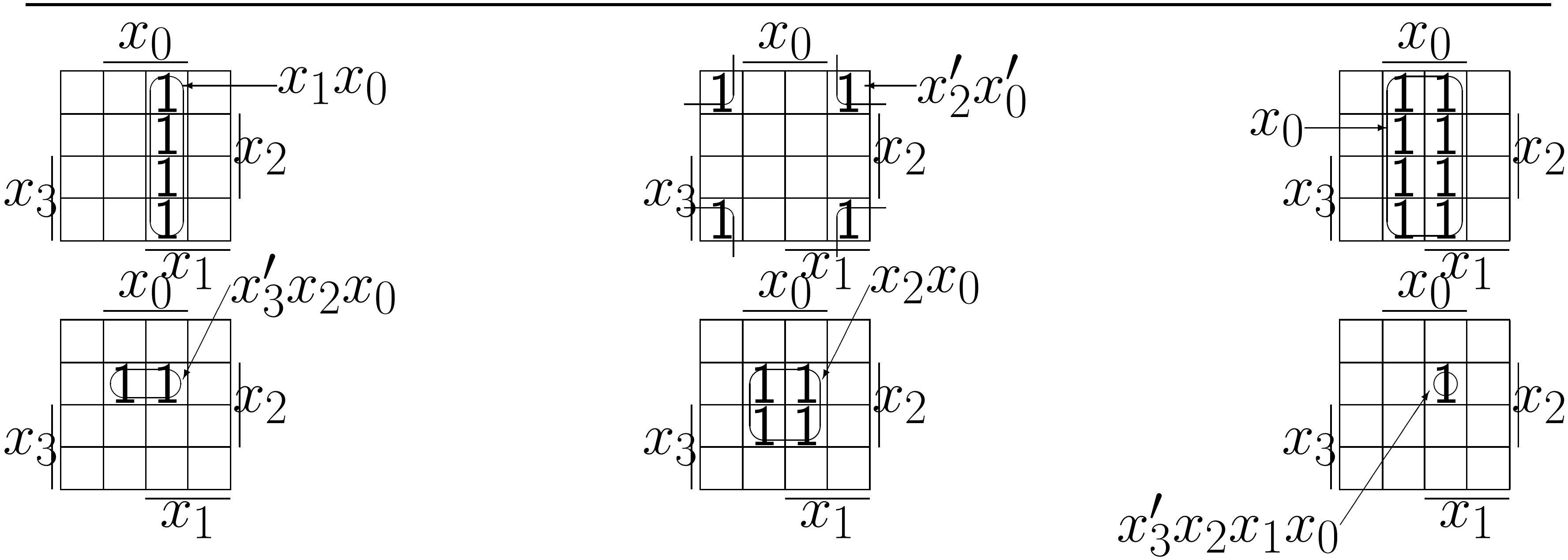
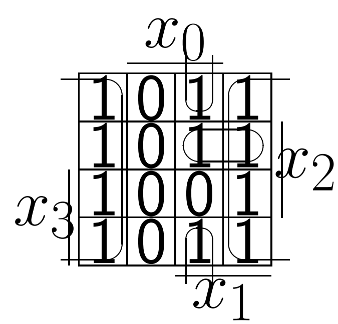


Figure 5.8: Product terms and rectangles of 1-cells.

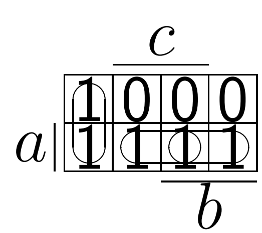
SUM OF PRODUCTS

represented in a K-map by the union of rectangles

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



$$E(a, b, c) = ab + ac + b'c'$$



RECTANGLES OF 0-CELLS AND PRODUCT OF SUMS

0-cell 13 CORRESPONDS TO THE MAXTERM

$$M_{13} = x_3' + x_2' + x_1 + x_0'$$

RECTANGLE OF $2^a \times 2^b$ 0-cells $\longleftrightarrow$ SUM TERM OF $n-(a+b)$ LITERALS

MINIMIZATION OF SUMS OF PRODUCTS

IMPLICANT: PRODUCT TERM FOR WHICH **f=1**

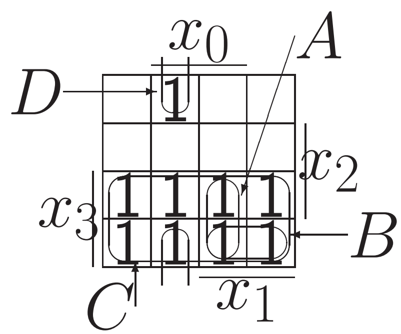


Figure 5.9: Implicant representation.

IMPLICANTS: $x_3'x_2'x_1'x_0$, ALL PRODUCT TERMS WITH x_3

PRIME IMPLICANT: IMPLICANT NOT COVERED BY ANOTHER IMPLICANT

PRIME IMPLICANTS: $x_2'x_1'x_0$, x_3

FIND ALL PIs

a) $f(x_2, x_1, x_0) = \textit{one-set}(2,4,6)$

x_0

	0	0	0	1
x_2	1	0	0	1

x_1

PIs: $x_2 x_0'$ and $x_1 x_0'$

b) $f(x_2, x_1, x_0) = \textit{one-set}(0,1,5,7)$

x_0

	1	1	0	0
x_2	0	1	1	0

x_1

PIs: $x_2' x_1'$, $x_2 x_0$, and $x_1' x_0$

c) $f(x_3, x_2, x_1, x_0) = \mathit{one\text{-}set}$(0,3,5,7,11,12,13,15)

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

PIs: x_2x_0, x_1x_0, $x_3x_2x_1'$, and $x_3'x_2'x_1'x_0'$

MINIMAL SUM OF PRODUCTS CONSISTS OF PRIME IMPLICANTS

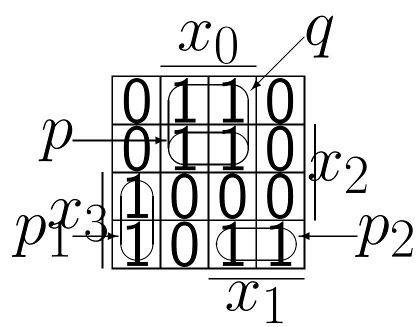
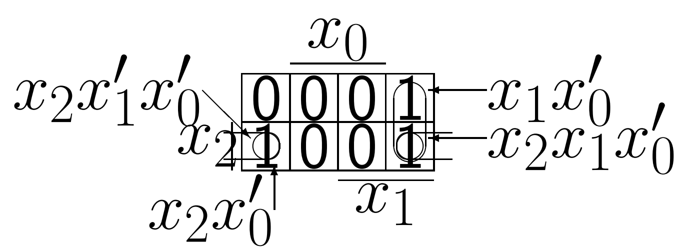


Figure 5.10: MINIMAL SUM OF PRODUCTS AND PRIME IMPLICANTS.

Example 5.9

$$E(x_2, x_1, x_0) = x_2x_1'x_0' + x_2x_1x_0' + x_1x_0'$$



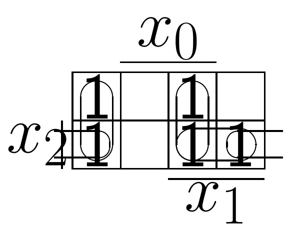
not PIs: $x_2x_1'x_0'$ and $x_2x_1x_0'$

PI: x_2x_0', x_1x_0'

REDUCED SP: $E(x_2, x_1, x_0) = x_2x_0' + x_1x_0'$

ESSENTIAL PRIME IMPLICANTS (EPI)

$p_e(\underline{a}) = 1$ and $p(\underline{a}) = 0$ FOR ANY OTHER PI p



EPIs: $x_1'x_0'$ and x_1x_0

NON-ESSENTIAL: x_2x_1, x_2x_0'.

- ALL EPIs ARE INCLUDED IN A MINIMAL SP

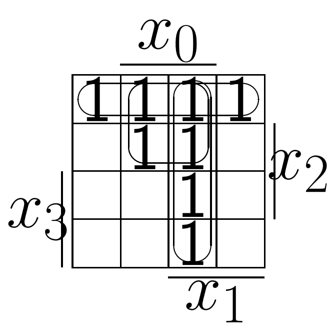
PROCEDURE FOR FINDING MIN SP

1. DETERMINE ALL PIs
2. OBTAIN THE EPIs
3. IF NOT ALL 1-CELLS COVERED, CHOOSE A COVER FROM THE REMAINING PIs

EXAMPLE 5.10

FIND A MINIMAL SP:

a) $E(x_3, x_2, x_1, x_0) = x_3'x_2' + x_3'x_2x_0 + x_1x_0$



- PIs: $x_3'x_2'$, $x_3'x_0$, and x_1x_0
- ALL EPIs
- UNIQUE MIN SP: $x_3'x_2' + x_3'x_0 + x_1x_0$

b) $E(x_2, x_1, x_0) = \Sigma\, m(0, 3, 4, 6, 7)$

		x_0	x_0	
	1		1	
x_2	1		1	1
			x_1	x_1

- PIs: $x_1'x_0'$, x_1x_0, x_2x_0', and x_2x_1

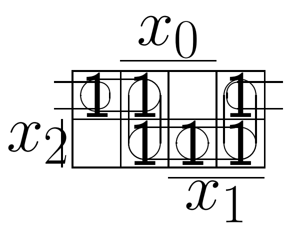
- EPIs: $x_1'x_0'$ and x_1x_0

- EXTRA COVER: x_2x_0' or x_2x_1

- TWO MIN SPs:

$$x_1'x_0' + x_1x_0 + x_2x_0' \quad \text{and} \quad x_1'x_0' + x_1x_0 + x_2x_1$$

c) $E(x_2, x_1, x_0) = \Sigma\, m(0, 1, 2, 5, 6, 7)$



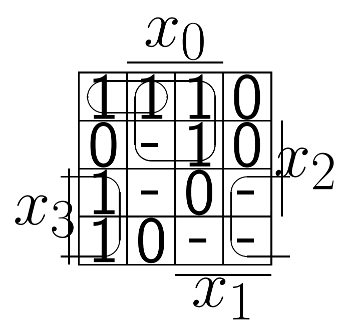
- PIs: $x_2'x_1'$, $x_2'x_0'$, x_2x_0, x_2x_1, $x_1'x_0$, and x_1x_0'

- No EPIs

- TWO MIN SPs

$$x_2'x_1' + x_2x_0 + x_1x_0' \quad \text{and} \quad x_2'x_0' + x_1'x_0 + x_2x_1$$

MINIMAL SPs FOR INCOMPLETELY SPECIFIED FUNCTIONS



A minimal SP

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

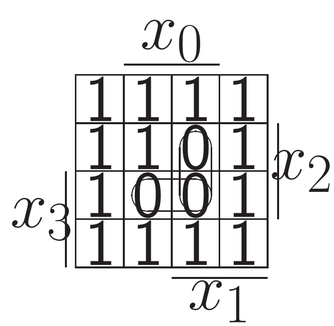
MINIMIZATION OF PRODUCTS OF SUMS

IMPLICATE: SUM TERM FOR WHICH $f = 0$.

PRIME IMPLICATE: IMPLICATE NOT COVERED BY ANOTHER IMPLICATE

ESSENTIAL PRIME IMPLICATE: AT LEAST ONE "CELL" NOT INCLUDED IN OTHER IMPLICATE

$f(x_3, x_2, x_1, x_0) = \mathit{zero\text{-}set}(7,13,15)$

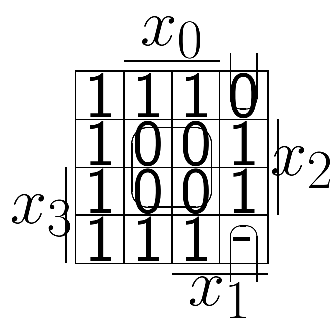


THE PRIME IMPLICATES: $(x_3' + x_2' + x_0')$ and $(x_2' + x_1' + x_0')$

BOTH ESSENTIAL

PROCEDURE FOR FINDING MIN PS

1. DETERMINE ALL PRIME IMPLICATES
2. DETERMINE THE ESSENTIAL PRIME IMPLICATES
3. FROM SET OF NONESSENTIAL PRIME IMPLICATES, SELECT COVER OF REMAINING 0-CELLS



- THE PRIME IMPLICATES: $(x_0' + x_2')$ and $(x_0 + x_2 + x_1')$
- BOTH ESSENTIAL, THE MINIMAL PS IS $(x_0' + x_2')(x_0 + x_2 + x_1')$

MINIMAL TWO-LEVEL GATE NETWORK DESIGN: EXAMPLE 5.14

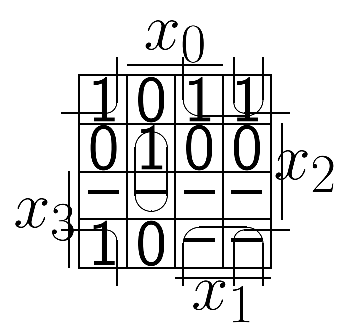
Input: $x \in \{0, 1, 2, ..., 9\}$, coded in BCD as

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

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: $z = x_2'x_1$ + $x_2'x_0'$ + $x_2x_1'x_0$

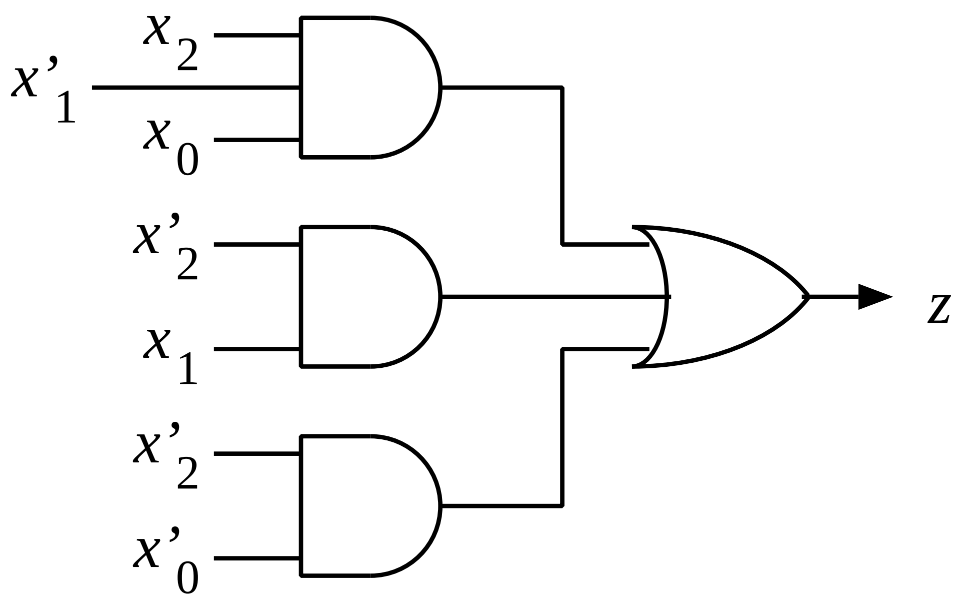


Figure 5.11: MINIMAL AND-OR NETWORK

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

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:

1	1	1	0
0	1	1	0
1	1	0	1
0	0	1	0

x_0, x_1, x_2, x_3

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)

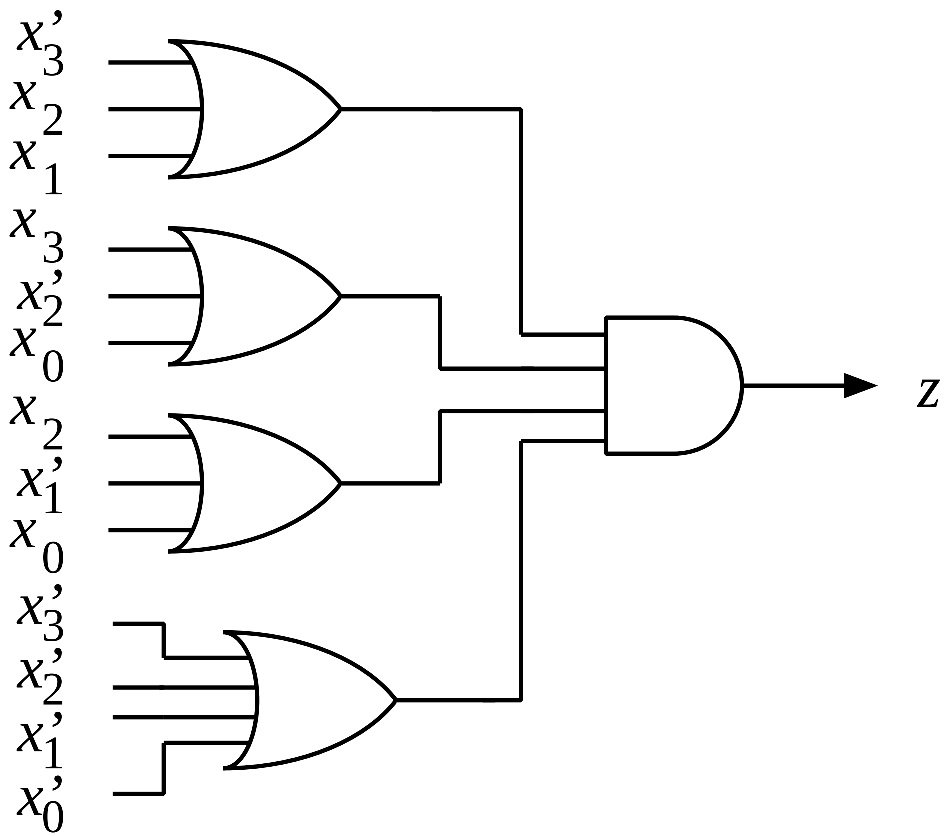


Figure 5.12: MINIMAL OR-AND NETWORK

QUINE-McCLUSKEY TABULAR METHOD

- Intended for minimizing s-expressions of more than 4 variables
- Suitable for computer-based methods
- Part 1: Determine the prime implicants (prime implicates) of the function. Use Xy+Xy' = X (or (X+y)(X+y') = X for min POS)
- Part 2: Select a set of prime implicants (prime implicates) that covers the function and has the minimum cost: construct PI chart
- The number of prime implicants is usually very large: obtaining a minimum cover is time-consuming

TABULAR FINDING OF PRIME IMPLICANTS

- Given a function f of n variables, determine the prime implicants
- **Column 1:** list all the minterms according to the number of 1's in the binary representation
- **Column 2:** list implicants with $n-1$ literals by pairing elements of Column 1 that differ in the value of one variable; mark the variable in which they differ with "-". Mark elements of Column 1 which form such a pair with N ("Not prime implicants")

  1011 and 1001 produce 10-1
- **Column 3:** list implicants with $n-2$ literals by pairing elements of Column 2 which have "-" in the same position and differ only in one variable. Mark as above.

  10-1 and 00-1 produce -0-1
- **Continue** this process until no more elements can be paired

- The prime implicants are elements not marked with N

EXAMPLE : $f(x_4, \ldots, x_0) = \Sigma\, m(1, 3, 4, 6, 9, 11, 12, 14, 16, 18, 21, 23, 24, 26, 29, 31)$

Minterms	4-literal prods	3-literal prods
00001 N	000-1 N	0-0-1
00100 N	0-001 N	0-1-0
10000 N	001-0 N	1-0-0
	0-100 N	
00011 N	100-0 N	1-1-1
00110 N	1-000 N	
01001 N		
01100 N	0-011 N	
10010 N	0-110 N	
11000 N	010-1 N	
	011-0 N	
01011 N	1-010 N	
01110 N	110-0 N	
10101 N		
11010 N	101-1 N	
	1-101 N	
10111 N		
11101 N	1-111 N	

PRIME IMPLICANT CHART

	1	3	4	6	9	11	12	14	16	18	21	23	24	26	29	31	Essential Prime
0-0-1	x	x			x	x											●
0-1-0			x	x			x	x									●
1-0-0									x	x			x	x			●
1-1-1											x	x			x	x	●

$$f(x_4, \ldots, x_0) = x_4' x_2' x_0 + x_4' x_2 x_0' + x_4 x_2' x_0' + x_4 x_2 x_0$$

DESIGN OF MULTIPLE-OUTPUT TWO-LEVEL GATE NETWORKS

- SEPARATE NETWORK FOR EACH OUTPUT: NO SHARING

EXAMPLE 5.16

Inputs: $(x_2, x_1, x_0), \quad x_i \in \{0, 1\}$
Output: $z \in \{0, 1, 2, 3\}$
Function: $z = \Sigma_{i=0}^{2} x_i$

1. THE SWITCHING FUNCTIONS IN TABULAR FORM ARE

x_2	x_1	x_0	z_1	z_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

EXAMPLE 5.16 (cont.)

2. THE CORRESPONDING K-MAPS ARE

$$z_1 \quad \begin{array}{c|cccc|} & & \overline{\quad x_0 \quad} & & \\ \hline & 0 & 0 & 1 & 0 \\ x_2 & 0 & 1 & 1 & 1 \\ \hline & & & \underline{\quad x_1 \quad} & \end{array} \qquad\qquad z_0 \quad \begin{array}{c|cccc|} & & \overline{\quad x_0 \quad} & & \\ \hline & 0 & 1 & 0 & 1 \\ x_2 & 1 & 0 & 1 & 0 \\ \hline & & & \underline{\quad x_1 \quad} & \end{array}$$

3. MINIMAL SPs:

$$\begin{aligned} z_1 &= x_2x_1 + x_2x_0 + x_1x_0 \\ z_0 &= x_2'x_1'x_0 + x_2'x_1x_0' + x_2x_1'x_0' + x_2x_1x_0 \end{aligned}$$

4. MINIMAL PSs:

$$\begin{aligned} z_1 &= (x_2 + x_0)(x_2 + x_1)(x_1 + x_0) \\ z_0 &= (x_2 + x_1 + x_0)(x_2 + x_1' + x_0') \\ &\quad\ (x_2' + x_1 + x_0')(x_2' + x_1' + x_0) \end{aligned}$$

5. SP AND PS EXPRESSIONS HAVE THE SAME COST

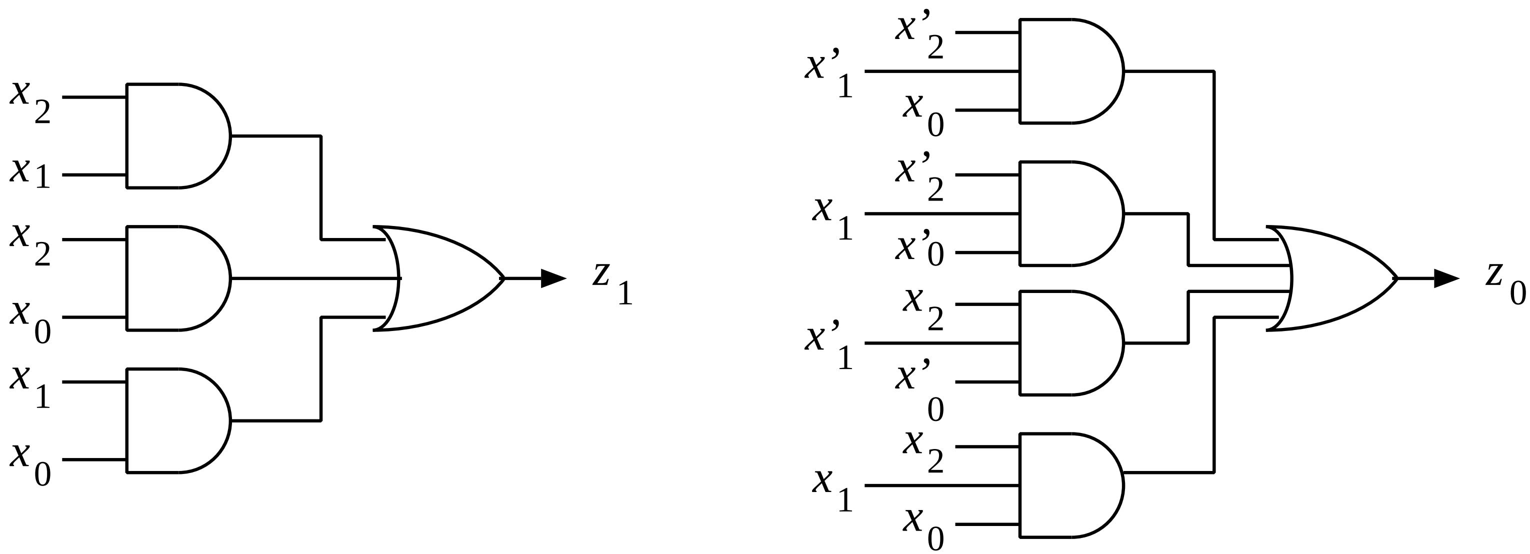


Figure 5.13: MINIMAL TWO-OUTPUT AND-OR NETWORK

TWO-LEVEL NAND-NAND AND NOR-NOR NETWORKS

$$E = p_1 + p_2 + p_3 + \ldots + p_n$$

$p_1, p_2, \ldots$ ARE PRODUCT TERMS

$$E = (p_1' \cdot p_2' \cdot p_3' \ldots p_n')'$$

or

$$E = NAND(NAND_1, NAND_2, NAND_3, \ldots, NAND_n)$$

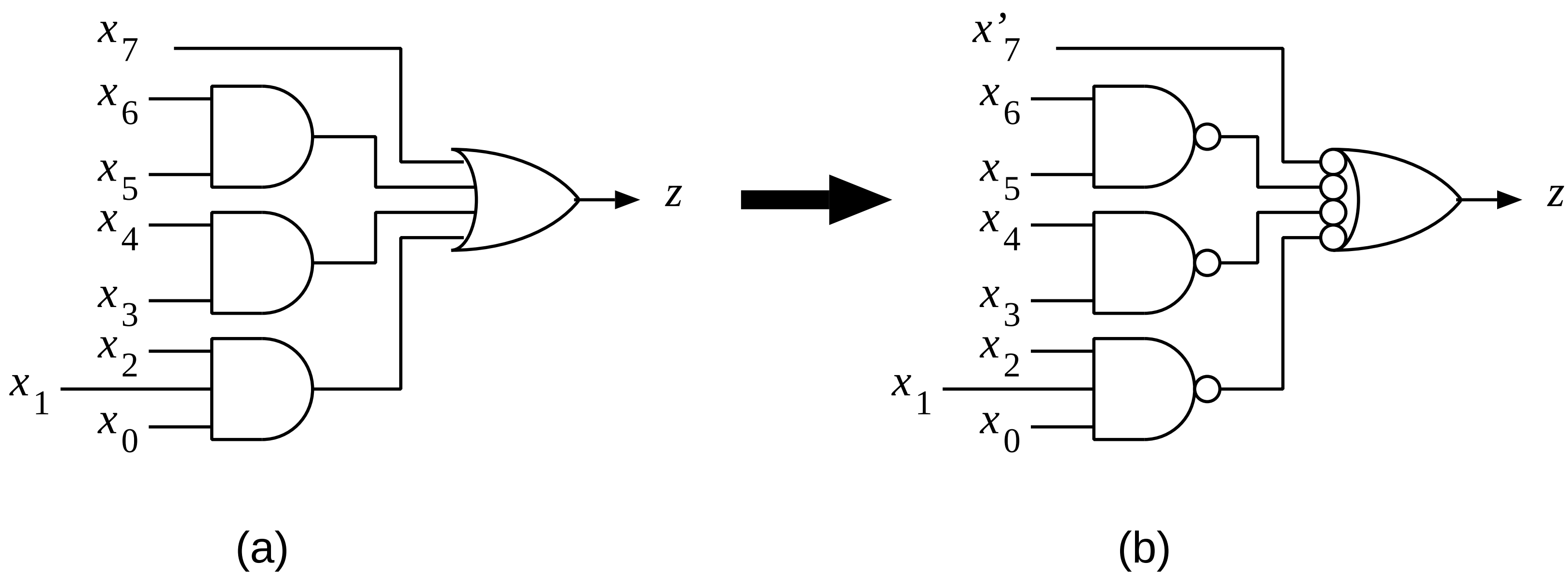


Figure 5.15: TRANSFORMATION OF AND-OR NETWORK INTO NAND-NAND NETWORK

EXAMPLE: NOR NETWORK

$$z = x_5'(x_4 + x_3')(x_2 + x_1 + x_0)$$

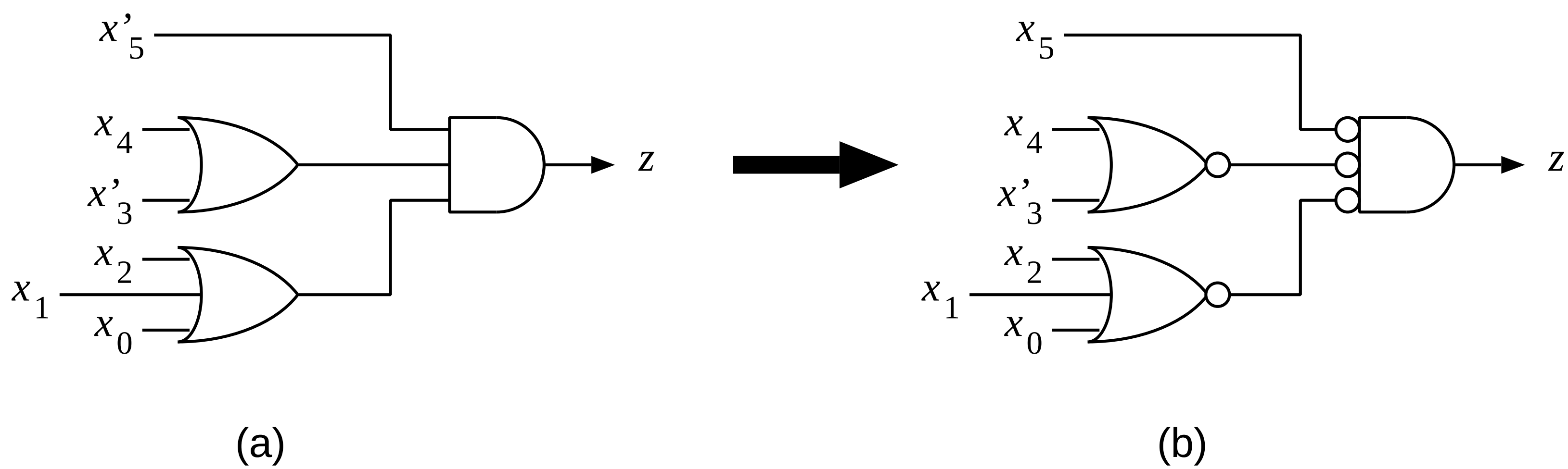


Figure 5.16: EQUIVALENT OR-AND AND NOR-NOR NETWORKS

LIMITATIONS OF TWO-LEVEL NETWORKS

1. THE REQUIREMENT OF UNCOMPLEMENTED AND COMPLEMENTED INPUTS

   IF NOT SATISFIED, AN ADDITIONAL LEVEL OF NOT GATES NEEDED

2. A TWO-LEVEL IMPLEMENTATION OF A FUNCTION MIGHT REQUIRE A LARGE NUMBER OF GATES AND IRREGULAR CONNECTIONS

3. EXISTING TECHNOLOGIES HAVE LIMITATIONS IN THE FAN-IN OF THE GATES

4. THE PROCEDURE ESSENTIALLY LIMITED TO THE SINGLE-OUTPUT CASE

5. THE COST CRITERION OF MINIMIZING THE NUMBER OF GATES IS NOT ADEQUATE FOR MANY MSI/LSI/VLSI DESIGNS

PROGRAMMABLE MODULES: PLAs and PALs

- STANDARD (FIXED) STRUCTURE
- CUSTOMIZED (PROGRAMMED) FOR A PARTICULAR FUNCTION
  - DURING THE LAST STAGE OF FABRICATION
  - WHEN INCORPORATED INTO A SYSTEM
- FLEXIBLE USE
- MORE EXPENSIVE AND SLOWER THAN FIXED-FUNCTION MODULES
- OTHER TYPES DISCUSSED IN Chapter 12

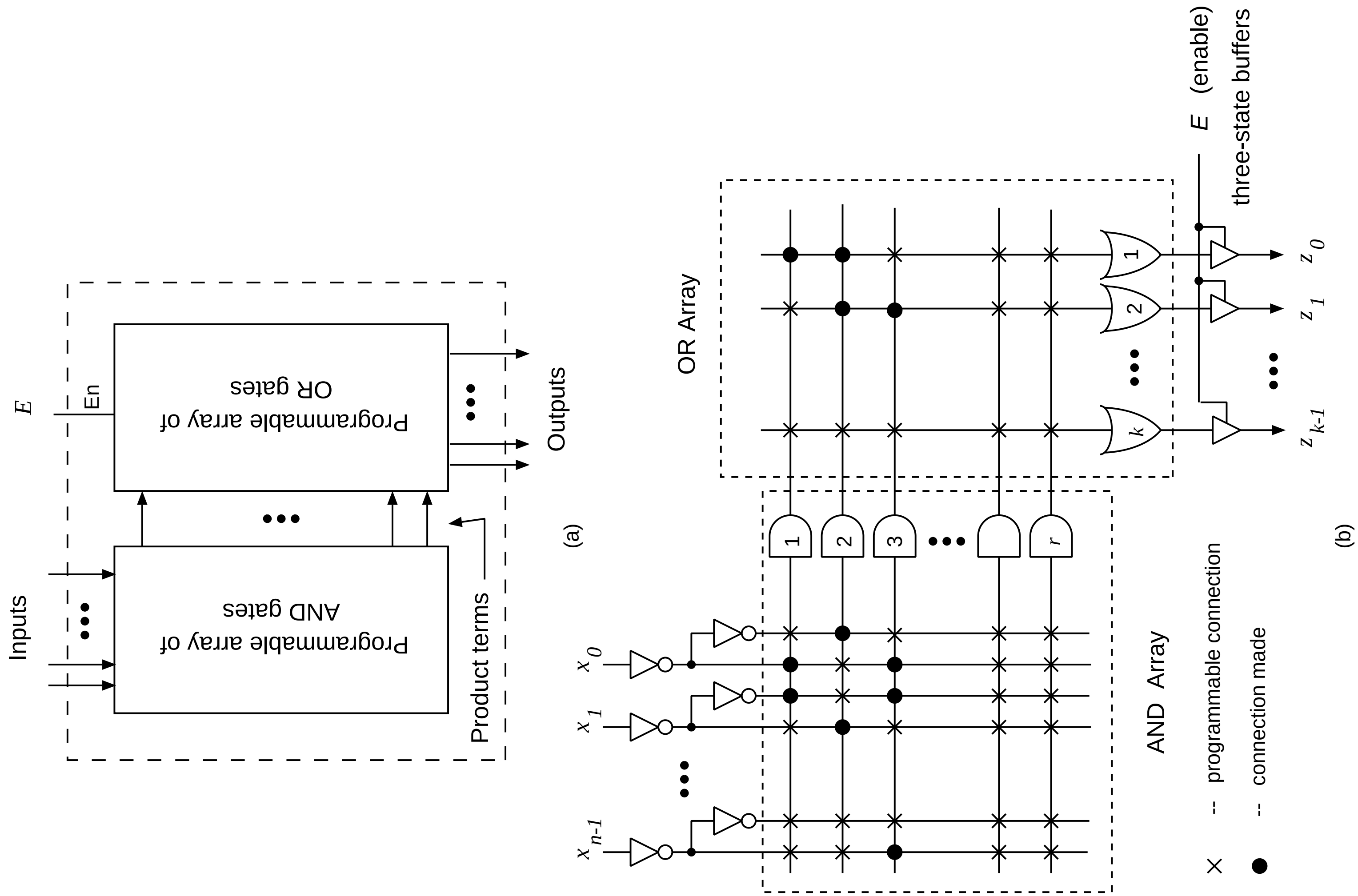
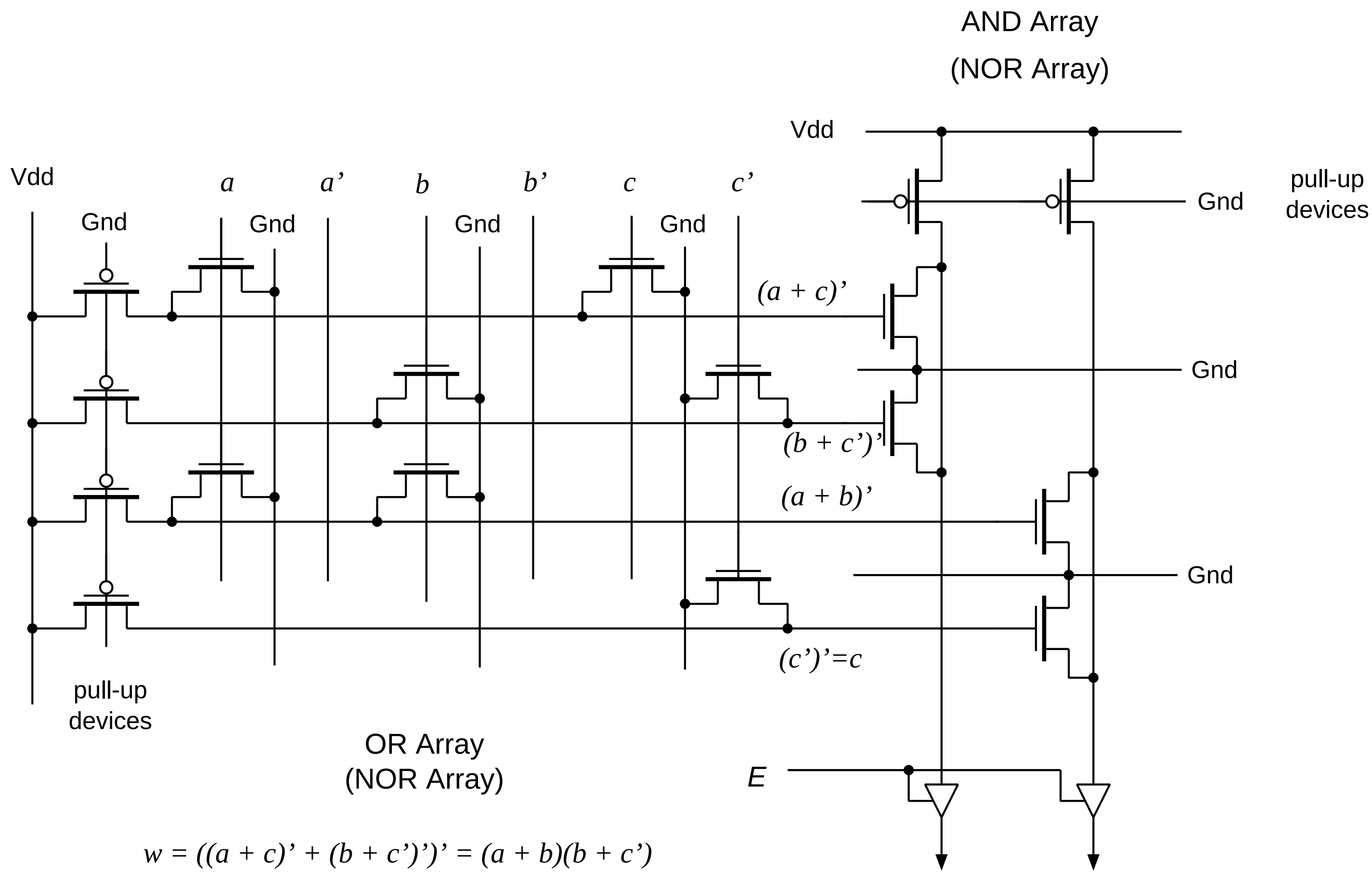


Figure 5.17: PROGRAMMABLE LOGIC ARRAY (PLA): a) BLOCK DIAGRAM; b) LOGIC DIAGRAM.

MOS PLA (OR-AND VERSION)



$$w = ((a + c)' + (b + c')')' = (a + b)(b + c')$$

$$z = ((a + b)' + c)' = (a + b)\, c'$$

Figure 5.18: EXAMPLE OF PLA IMPLEMENTATION AT THE CIRCUIT LEVEL: FRAGMENT OF A MOS PLA .

IMPLEMENTATION OF SWITCHING FUNCTIONS USING PLAs

A BCD-to-Gray CONVERTER

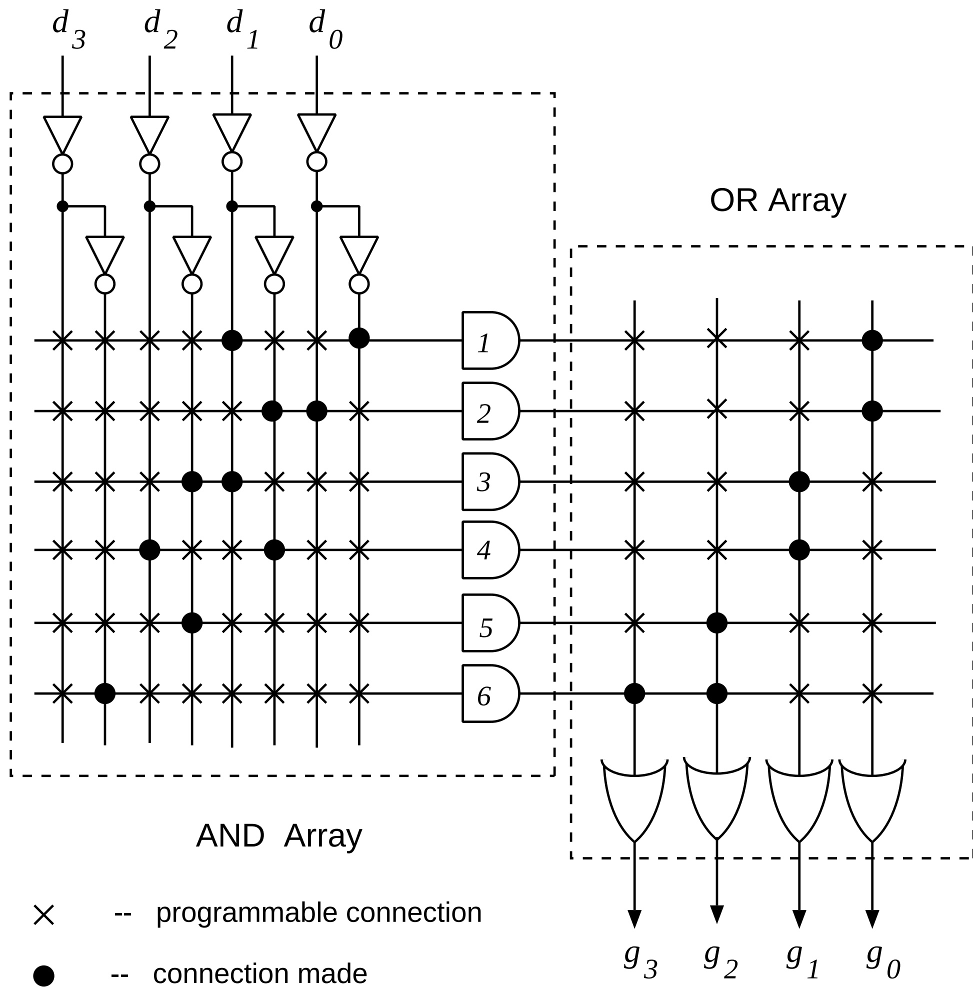
Inputs: $\underline{d} = (d_3, d_2, d_1, d_0), \quad d_j \in \{0,1\}$
Outputs: $\underline{g} = (g_3, g_2, g_1, g_0), \quad g_j \in \{0,1\}$

Function:

i	$d_3d_2d_1d_0$	$g_3g_2g_1g_0$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

EXPRESSIONS:

$$\begin{aligned} g_3 &= d_3 \\ g_2 &= d_3 + d_2 \\ g_1 &= d_2'd_1 + d_2d_1' \\ g_0 &= d_1d_0' + d_1'd_0 \end{aligned}$$



Note: a PLA chip would have more rows and columns then shown here

Figure 5.19: PLA IMPLEMENTATION OF BCD-Gray CODE CONVERTER.