

## CHAPTER 5: DESIGN OF COMBINATIONAL SYSTEMS

- MULTIPLE INPUTS & OUTPUTS
- MAIN OBJECTIVES: MINIMAL SIZE (COST), MINIMAL DELAY  
IN THIS COURSE

OTHER OBJECTIVES:  
MIN. POWER  
\$ ENERGY,  
REGULARITY

# GATES, # INPUTS

# LEVELS, LOADS,  
INTERCONNECTS

PROBLEM



HL SPECIFICATION



BL SPECIFICATION



{ SF<sub>i</sub> }



{ SE }

SF<sub>k</sub> → { CSP, CPS, SOP, POS }  
( TWO-LEVEL )

↓ [ MINIMIZATION OF SF<sub>S</sub> ]

MIN SOP, MIN POS

OUR METRIC FOR MINIMALITY:

MIN # GATES WITH

MIN # INPUTS

# 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)

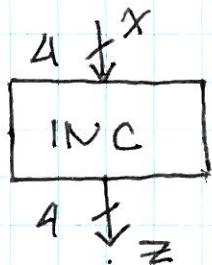
PRODUCTS: AND gates

SUM: OR gate

MULTIOUTPUT NETWORKS: ONE OR GATE USED FOR EACH OUTPUT

# PRODUCT OF SUMS NETWORKS - SIMILAR

EXAMPLE: MODULO-16 INCREMENTER



$$z = (x+1) \bmod 2^4 \quad x \in \{0, \dots, 15\}$$

$$z \in \{0, \dots, 15\}$$

X	0	1	0	1	0	1	1	0	0	1	1	1	0	0	0	?
z	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	?
	NC	F Lip			NC		F Lip		CHARGE		F Lip					

$$z_i = \begin{cases} 1 & \text{IF } (x_i = 1 \text{ AND } \exists j < i, x_j = 0) \text{ (a)} \\ 0 & \text{OR } (x_i = 0 \text{ AND } x_j = 1, \forall j < i) \text{ (b)} \\ 0 & \text{OTHERWISE} \end{cases}$$

$$z_0 = x'_0$$

$$\text{LET } S_k = x_k \cdot x_{k-1} \cdots x_0$$

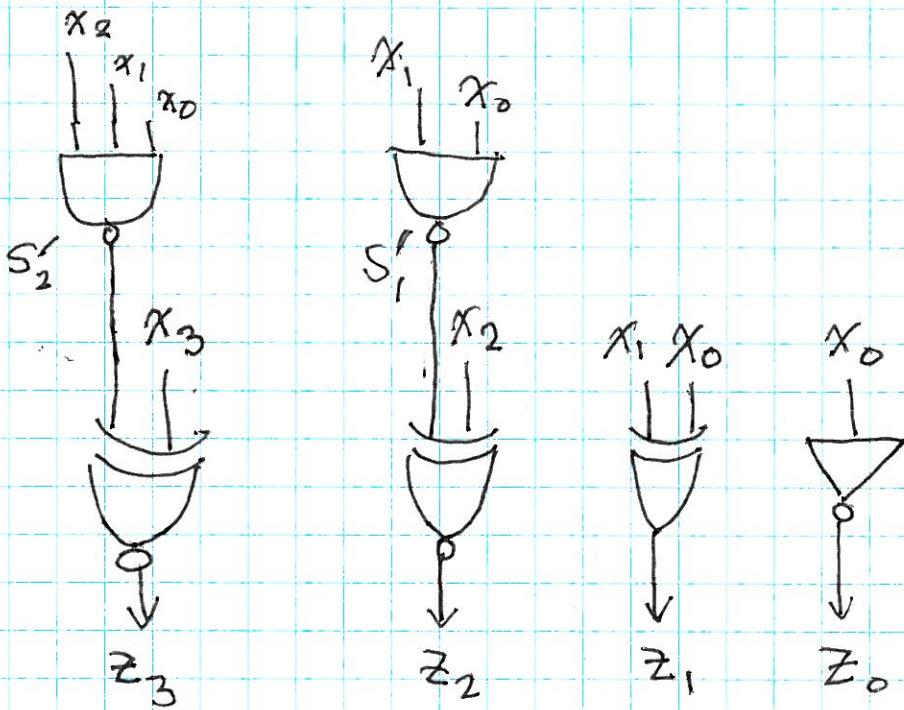
$$z_1 = \underbrace{x_1 x'_0}_{(a)} + \underbrace{x'_1 x_0}_{(b)} = x_1 \oplus S_0$$

$$z_2 = x_2 \underbrace{(x_1 x'_0 + x'_1 x_0 + x_1 x'_0)}_{x'_1 + x'_0} + x'_2 (x_1 x_0) = x_2 \oplus S_1$$

$$= x_2 (x'_1 + x'_0) + x'_2 (x_1 x_0)$$

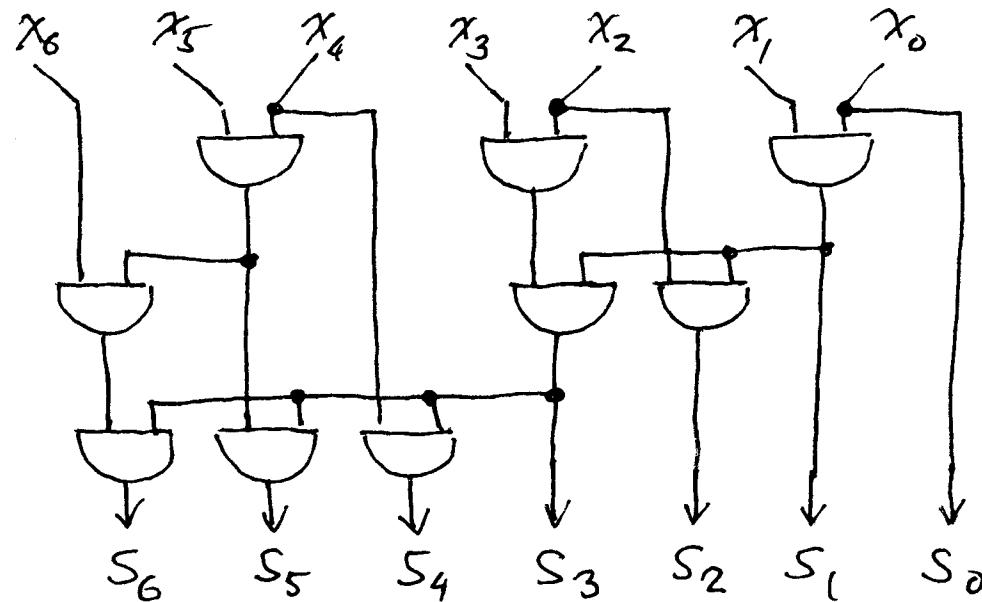
$$z_3 = x_3 (x'_2 + x'_1 + x'_0) + x'_3 (x_2 x_1 x_0) = x_3 \oplus S_2$$

DESIGN:



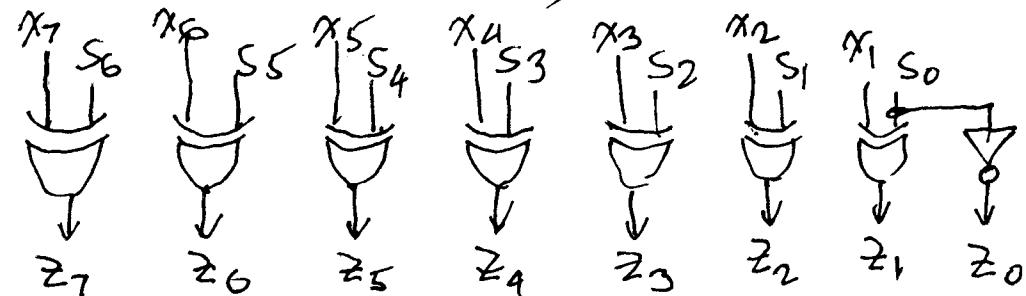
REDESIGN MOD-64 INCREMENTER USING  
THIS APPROACH AND COMPARE W.R.T. DELAY  
AND COST (IN EQUIVALENT GATES)

AN EFFICIENT, GENERAL DESIGN APPROACH FOR  $\{S_i\}$  NET  
 EXAMPLE: mod- $2^8$  INCREMENTER USES  $\{S_6, S_5, \dots, S_1, S_0\}$



THIS IS A PARALLEL PREFIX NETWORK

- + NUMBER OF LEVELS:  $\lceil \log_2(k-1) \rceil$  FOR  $k=8$ , 3 LEVELS
- + ONLY 2-INPUT GATES; - FANOUT DOUBLES PER LEVEL



# MODULO-64 INCREMENTER

---

**Input:**  $0 \leq x \leq 63$

**Output:**  $0 \leq z \leq 63$

**Function:**  $z = (x + 1) \bmod 64$

$$\begin{array}{r|cc} x & 010101 & x & 001111 \\ \hline z & 010110 & z & 010000 \end{array}$$

- RADIX-2 REPRESENTATION

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

$$\begin{aligned} z_5 &= x_5(x'_4 + x'_3 + x'_2 + x'_1 + x'_0) + x'_5 x_4 x_3 x_2 x_1 x_0 \\ &= x_5 x'_4 + x_5 x'_3 + x_5 x'_2 + x_5 x'_1 + x_5 x'_0 + x'_5 x_4 x_3 x_2 x_1 x_0 \\ z_4 &= x_4 x'_3 + x_4 x'_2 + x_4 x'_1 + x_4 x'_0 + x'_4 x_3 x_2 x_1 x_0 \\ z_3 &= x_3 x'_2 + x_3 x'_1 + x_3 x'_0 + x'_3 x_2 x_1 x_0 \end{aligned}$$

$$z_2 = x_2x'_1 + x_2x'_0 + x'_2x_1x_0$$

$$z_1 = x_1x'_0 + x'_1x_0$$

$$z_0 = x'_0$$

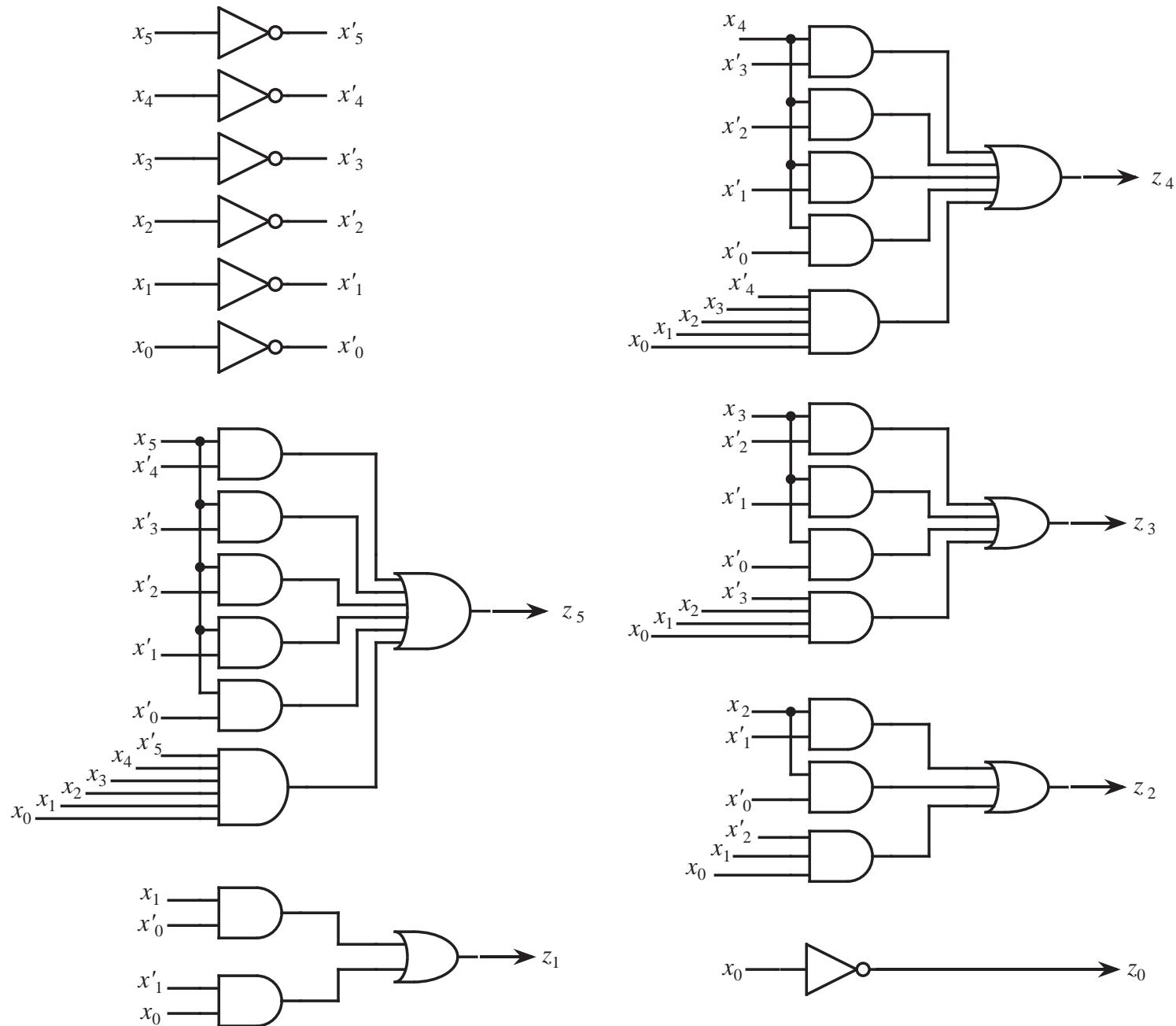


Figure 5.1: NOT-AND-OR MODULO-64 INCREMENTER NETWORK.

# 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**)

$$E(x_2, x_1, x_0) = (x'_2 + x_1)(x_1 + x'_0)(x_2 + x'_1 + x_0)$$

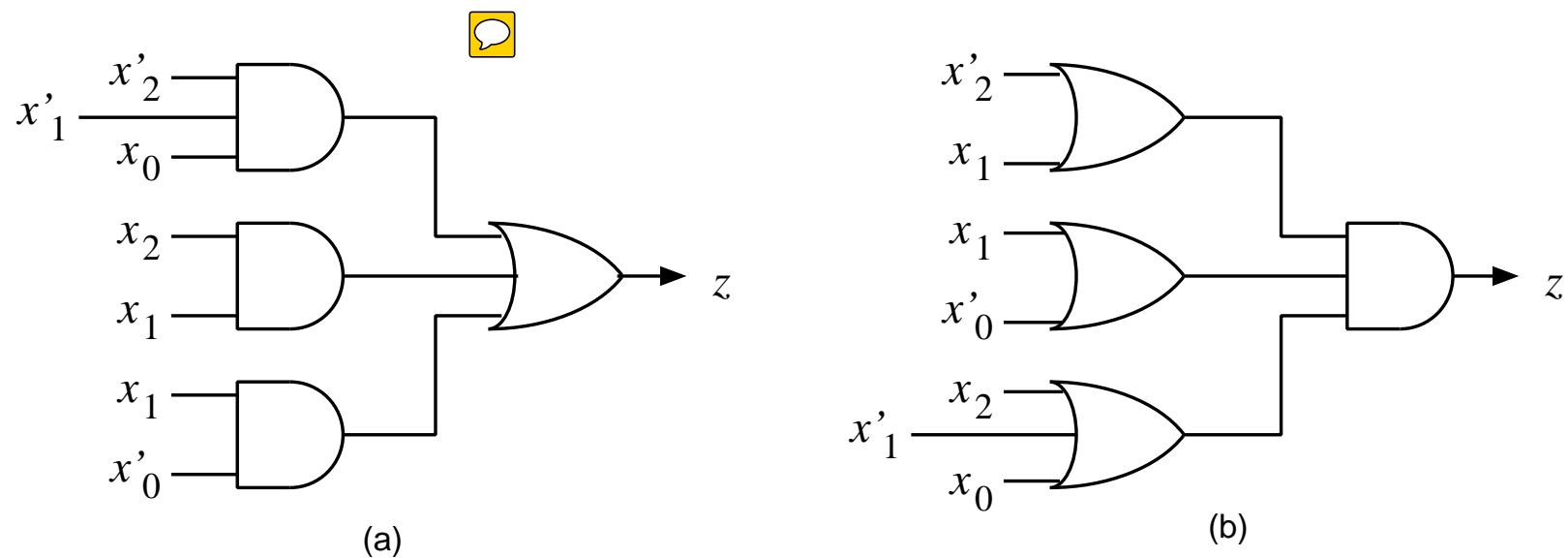


Figure 5.2: AND-OR and OR-AND NETWORKS.

# MINIMAL TWO-LEVEL NETWORKS

---

1. INPUTS: UNCOMPLEMENTED AND COMPLEMENTED



2. FANIN UNLIMITED

3. SINGLE-OUTPUT NETWORKS

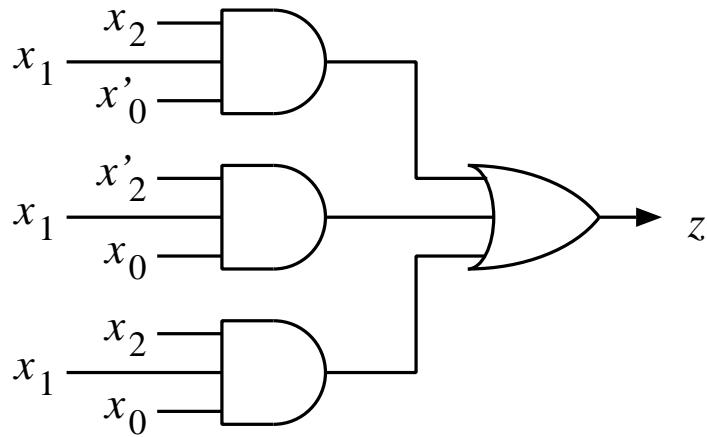
4. MINIMAL NETWORK:

MINIMUM NUMBER OF GATES WITH MINIMUM NUMBER  
OF INPUTS

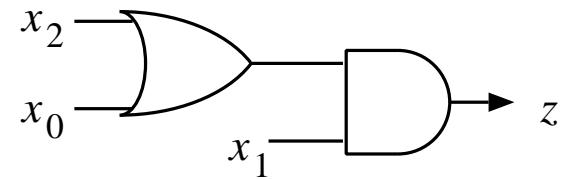
(minimal expression: min. number of terms with min. number  
of literals)

# NETWORKS WITH DIFFERENT COST

---



Network A



Network B

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

# MINIMAL EXPRESSIONS

---

- EQUIVALENT BUT DIFFERENT COST

$$E_1(x_2, x_1, x_0) = x'_2x_1x'_0 + x'_1x_0 + x_2x_0$$



$$E_2(x_2, x_1, x_0) = x_2x_1x_0 + x'_2x_1x'_0 + x'_2x'_1x_0 + x_2x'_1x_0$$



- BOTH MINIMAL SP AND PS MUST BE OBTAINED AND COMPARED
- BASIS:



$$ab + ab' = a \quad (\text{for sum of products})$$

$$(a + b)(a + b') = a \quad (\text{for product of sums})$$

## KEY IDEA

CONSIDER  $pt_1 + pt_2, pt_3$  - PRODUCT TERMS OF  $k$  LITERALS

1. IF  $pt_1$  AND  $pt_2$  DIFFER IN ONE LITERAL

THEY CAN BE REDUCED TO A SINGLE  $pt$

OF  $k-1$  LITERALS

$$f_1(x_3, x_2, x_1, x_0) = M_0 + M_4 = x_3' x_2' x_1' x_0' + x_3' x_2 x_1' x_0' \quad [10 \text{ INPUTS}, \\ 3 \text{ GATES}]$$

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

$$= x_3' x_1' x_0' \quad [3 \text{ INPUTS}, \\ 1 \text{ GATE}]$$

2. IF  $pt_1 + pt_2 + pt_3 + pt_4$  DIFFER IN TWO LITERALS,

THEY CAN BE REDUCED TO A SINGLE  $pt$

OF  $k-2$  LITERALS

$$f_2(x_3, x_2, x_1, x_0) = M_0 + M_1 + M_4 + M_5$$

$$= x_3' x_2' x_1' x_0' + x_3' x_2' x_1' x_0 + x_3' x_2 x_1' x_0' + x_3' x_2 x_1' x_0 \quad 20 \text{ LIT.}$$

$$= x_3' x_1' (x_2' x_0' + x_2' x_0 + x_2 x_0' + x_2 x_0) = x_3' x_1' \quad 5 \text{ GATES}$$

2 LIT. 1 GATE  
2 LITERALS



# 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

# KARNAUGH MAPS (K-MAPS)

- GRAPHICAL MINIMIZATION AID

SF  $\leftrightarrow$  2D TRUTH TABLE

KM  $\leftrightarrow$  SPECIAL 2D TRUTH TABLE: DIFF. LABELS OR  
ROWS (COLUMNS)

$f(x_2, x_1, x_0)$	$x_2$	$x_1 x_0$	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

$x_2$	$x_1 x_0$	00	01	11	10
0	$x_2$	0	1	3	2
1	$x_2$	4	5	7	6
0	$x_2$	0	1	0	0

$x_2$	$x_1 x_0$	0	1	3	2
0	$x_2$	1	1	0	1
1	$x_2$	0	1	0	0
0	$x_2$	0	1	0	0

$$m_0 + m_1 + m_2 + m_5$$

THESE PTS DIFFER IN 1 VARIABLE:  $x_0$

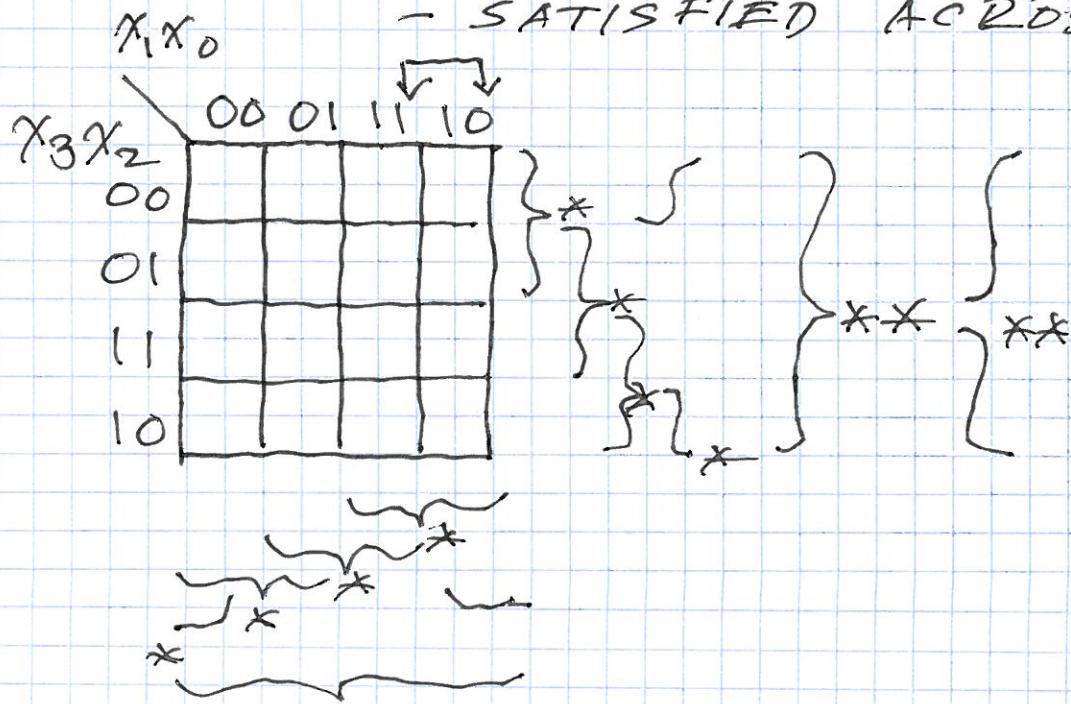
$$m_0 + m_1 = \underline{x_2' x_1' x_0' + x_2' x_1' x_0} = x_2' x_1'$$

ADJACENCY PROPERTY: ASSIGNMENTS FOR ANY SET

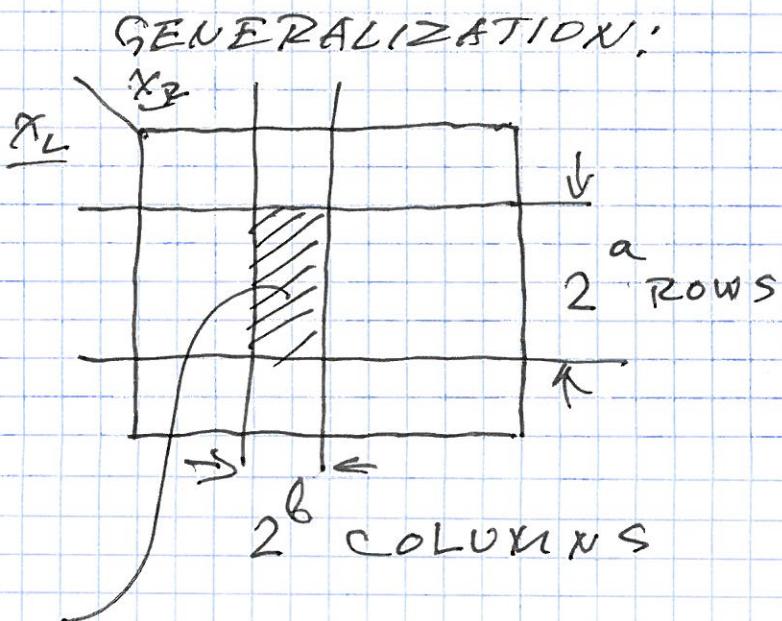
OF  $2^2$  ADJACENT ROWS (COLUMNS) DIFFER  
IN 2 VARIABLES

ROWS (COLUMNS) LABELED SO  
THAT THE ASSIGNMENTS FOR ANY  
SET OF  $2^2$  ADJACENT ROWS DIFFER  
ONLY IN 1 VARIABLE

- SATISFIED ACROSS BORDERS

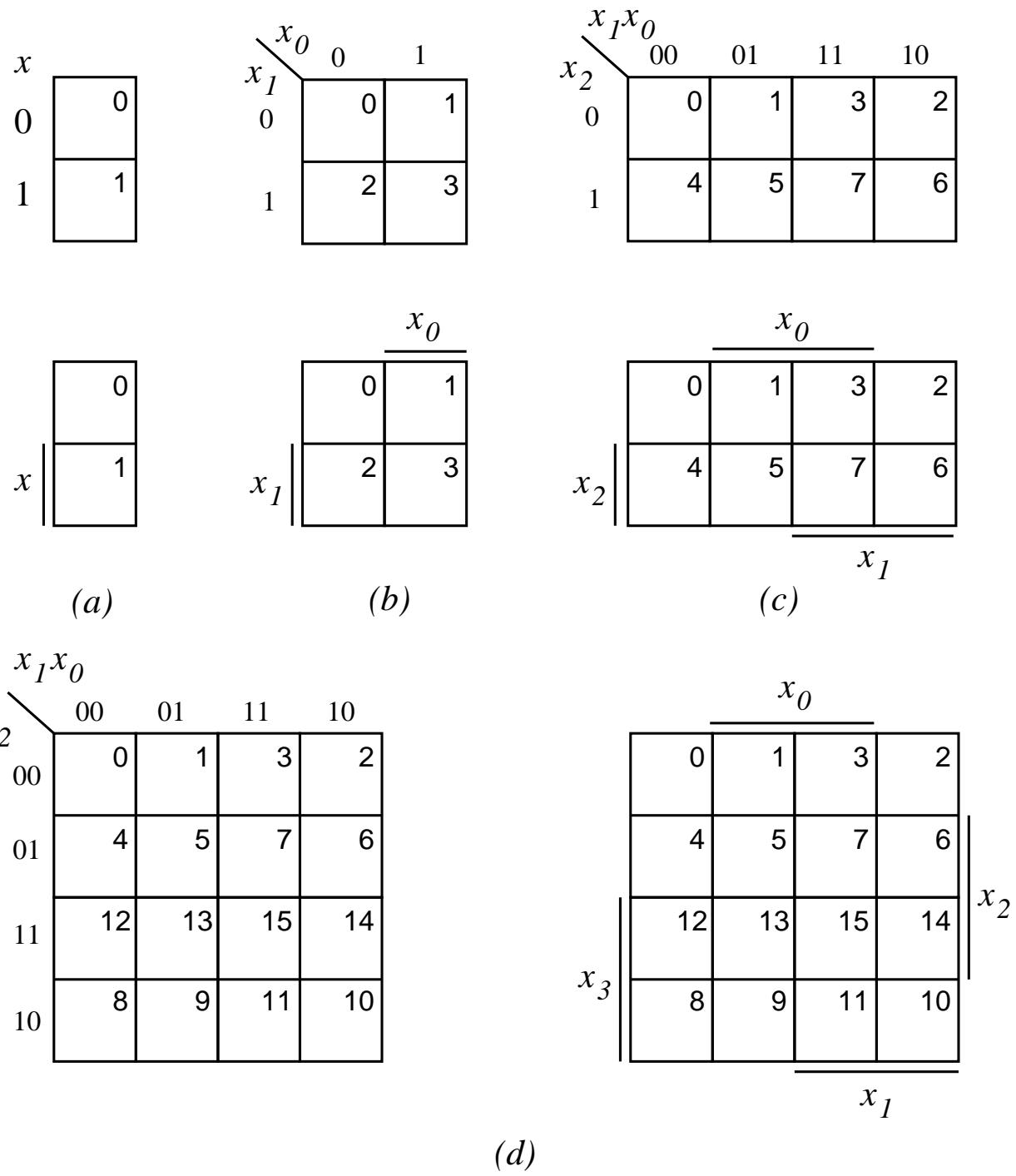


\* - ASSIGNMENTS  
DIFFER IN 1 VARIABLE  
\*\* - // IN 2 VARIABLES



n - (a+b) VARIABLES  
DO NOT CHANGE  
IN ASSIGNMENTS

Figure 5.4: K-Maps



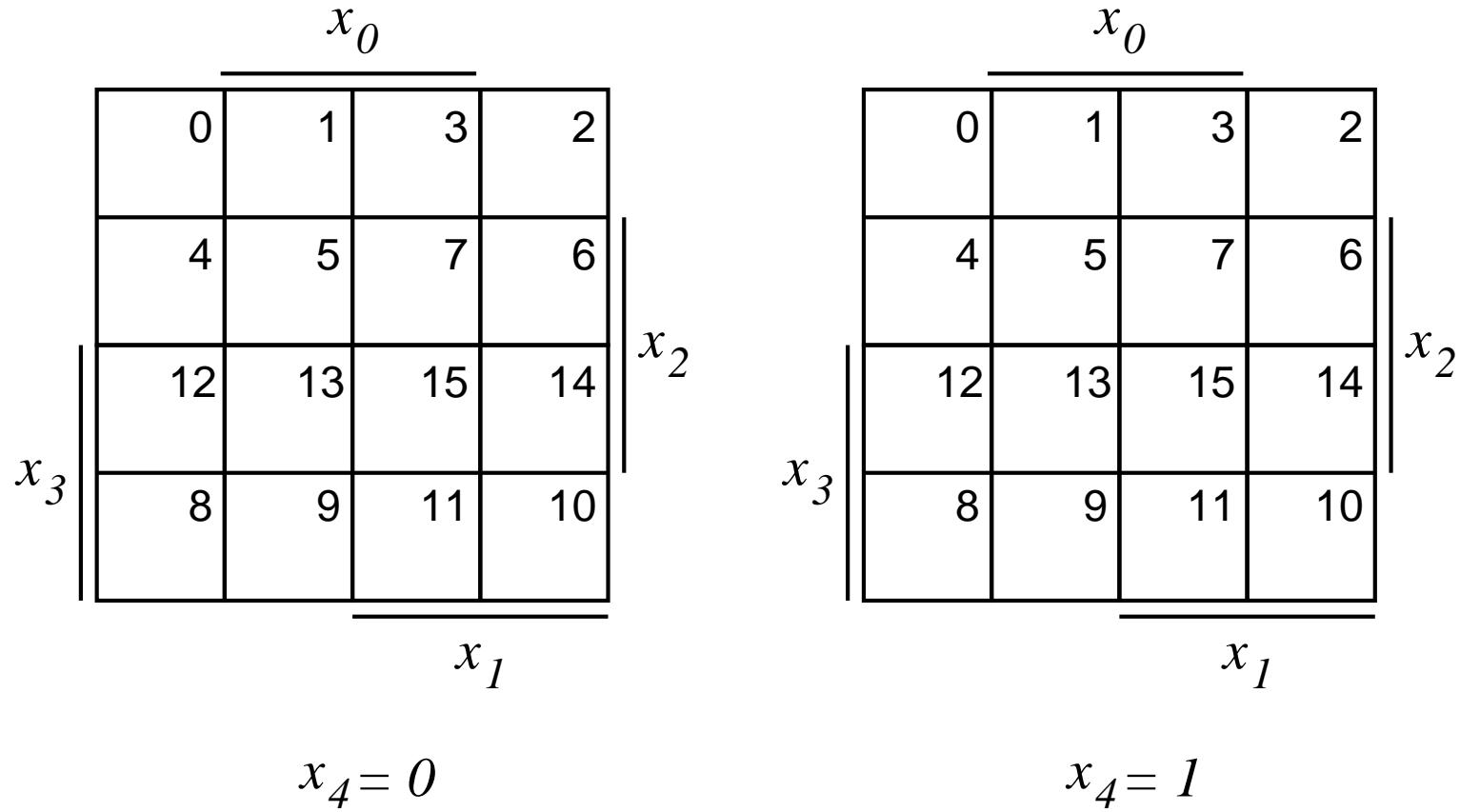


Figure 5.5: K-map FOR FIVE VARIABLES

## REPRESENTATION OF SF ON K-MAP

- TRIVIAL, IF YOU HAVE A CANONICAL EXPRESSION

MINTERM  $m_i \rightarrow 1\text{-CELL}_i$

MAXTERM  $M_j \rightarrow 0\text{-CELL}_j$

$$f(x_2, x_1, x_0) = \text{ONE-SET}(1, 3, 5) = \sum m(1, 3, 5)$$

$$= \prod M(0, 2, 4, 6, 7)$$

$x_0$	0	1	3	2
$x_2$	0	1	1	0
$x_1$	0	1	0	0

$$x_2' x_1 x_0 = m_3 \rightarrow 1\text{-CELL } 3$$

RECTANGLE OF 1-CELLS  $\leftrightarrow$  PRODUCT TERM

$$x_2 + x_1 + x_0 = M_0 \rightarrow 0\text{-CELL } 0$$

RECTANGLE OF 0-CELLS  $\leftrightarrow$  SUM TERM

$x_0$	0	1	3	2
$x_2$	0	1	3	2
$x_1$	0	1	0	0

$$m_4 + m_5 = \underline{x_3' x_2 x_1' x_0'} + \underline{x_3' x_2 x_1' x_0}$$

$$= x_3' x_2 x_1' \quad 2\text{ CELLS} \rightarrow 3\text{ VARS}$$

$$\begin{aligned} m_{10}, m_{11}, m_{14}, m_{15} &= (x_3' + x_2 + x_1' + x_0) \\ &\cdot (x_3' + x_2 + x_1' + x_0') \\ &\cdot (x_3' + x_2' + x_1' + x_0) \\ &\cdot (x_3' + x_2' + x_1' + x_0') \\ &= (x_3' + x_1') \end{aligned}$$

4 CELLS  
↓  
2 VADS

# REPRESENTATION OF SWITCHING FUNCTIONS

---

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

	$x_0$
$x_2$	1 0 0 1
	0 0 0 1
	$\overline{x_1}$

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

	$x_0$
$x_3$	1 0 0 1
	0 1 1 0
	1 0 1 1
	1 1 0 0
	$\overline{x_1}$

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

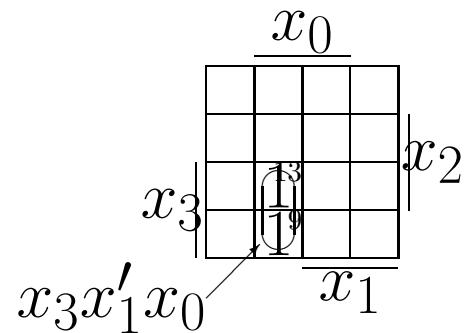
	$x_0$
$x_2$	1 0 - -
	1 1 0 0
	$\overline{x_1}$

# RECTANGLES OF 1-CELLS AND SUM OF PRODUCTS

---

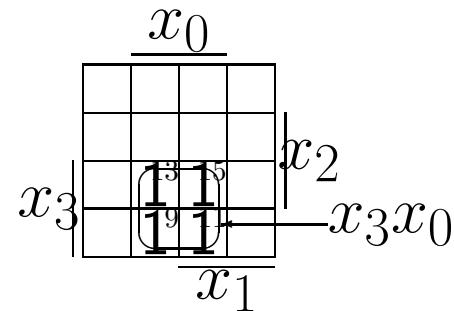
1. MINTERM  $m_j$  CORRESPONDS TO 1-CELL WITH LABEL  $j$ .
2. PRODUCT TERM OF  $n - 1$  LITERALS  $\longleftrightarrow$  RECTANGLE OF TWO ADJACENT 1-CELLS

$$\begin{aligned}
 x_3x'_1x_0 &= x_3x'_1x_0(x_2 + x'_2) \\
 &= x_3x_2x'_1x_0 + x_3x'_2x'_1x_0 \\
 &= m_{13} + m_9
 \end{aligned}$$

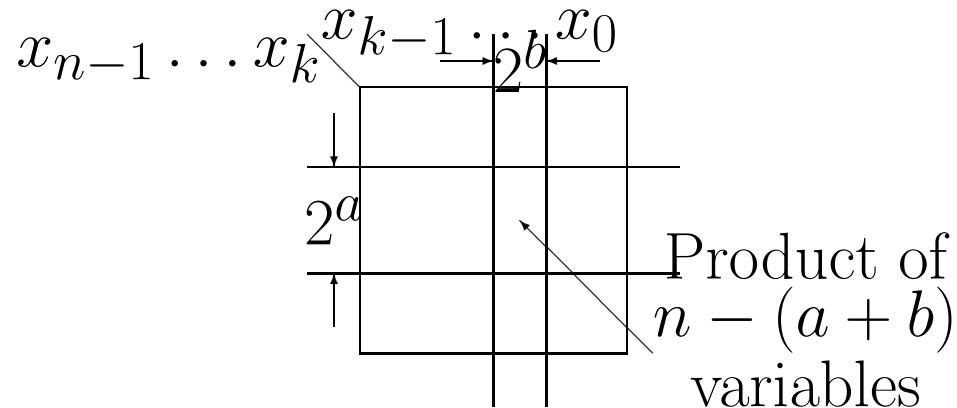


3. PRODUCT TERM OF  $n - 2$  LITERALS  $\longleftrightarrow$  RECTANGLE  
OF FOUR ADJACENT 1-CELLS

$$\begin{aligned}
 x_3x_0 &= x_3x_0(x_1 + x'_1)(x_2 + x'_2) \\
 &= x_3x'_2x'_1x_0 + x_3x'_2x_1x_0 + x_3x_2x'_1x_0 + x_3x_2x_1x_0 \\
 &= m_9 + m_{11} + m_{13} + m_{15}
 \end{aligned}$$

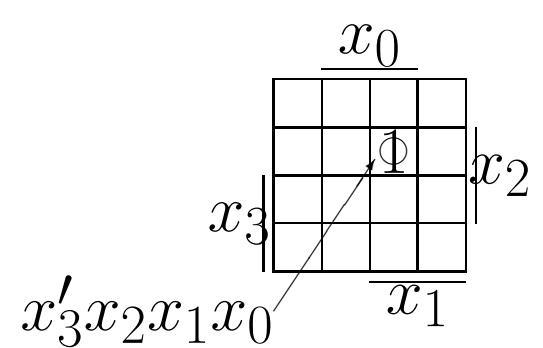
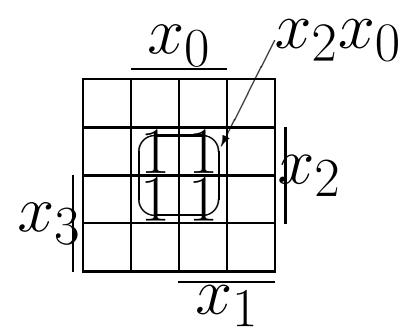
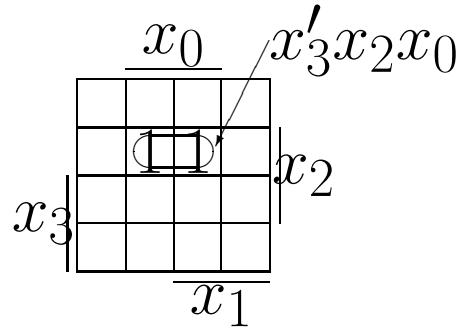
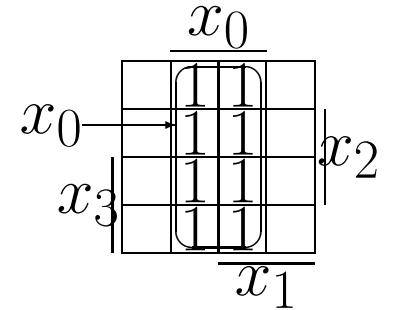
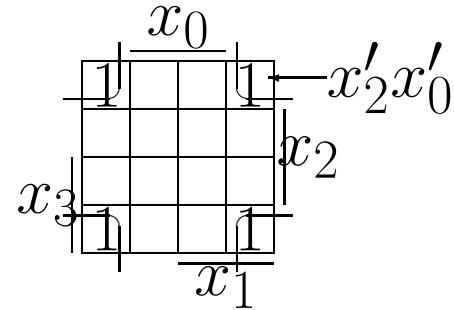
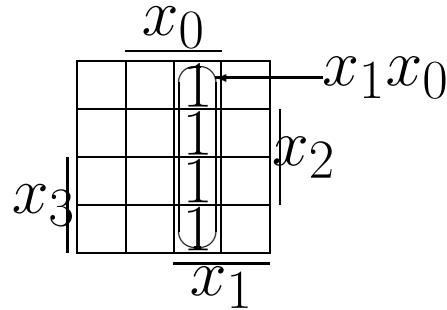


## 4. PRODUCT TERM OF $n - s$ LITERALS $\longleftrightarrow$ RECTANGLE OF $2^s$ ADJACENT 1-CELLS



# PRODUCT TERMS AND RECTANGLES OF 1-CELLS

19

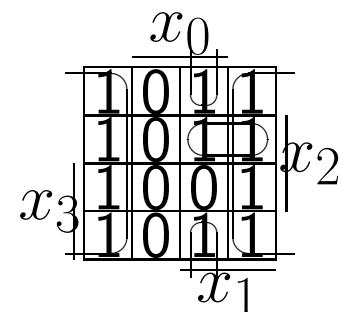


# SUM OF PRODUCTS

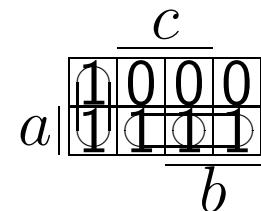
---

REPRESENTED IN A K-MAP BY THE UNION OF RECTANGLES

$$E(x_3, x_2, x_1, x_0) = x'_3x_2x_1 + x'_2x_1x_0 + x'_0$$



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



# 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 CORRESPONDS TO

SUM TERM OF  $n - (a + b)$  LITERALS

**IMPLICANT:**

$x_2$	$x_1$	$x_0$	$f(x)$	$x_2 x_0$	$x_2' x_1 x_0'$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	1	1	0

IF  $f(\underline{x}) = 1$  WHEN  $pt(\underline{x}) = 1$   
THEN  $pt(\underline{x})$  IS AN IMPLICANT OF  $f(\underline{x})$

$$f(\underline{x}) = m_2 + m_5 + m_7$$

$$pt_1(\underline{x}) = x_2 x_0 = m_5 + m_7$$

$$pt_2(\underline{x}) = x_2' x_1 x_0' = m_2$$

$pt_1$  &  $pt_2$  ARE  
IMPLICANTS OF  
 $f(\underline{x})$

$m_5$  AND  $m_7$   
ALSO IMPLICANTS

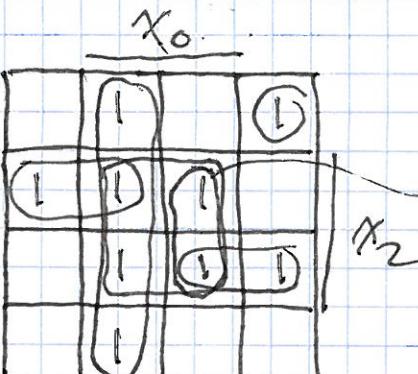
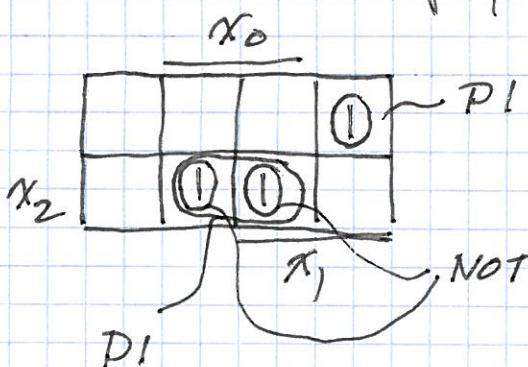
**PRIME IMPICANT (PI):**

AN IMPLICANT NOT

COVERED BY ANY

OTHER IMPLICANT

$pt_1$        $pt_2$



ALL PIS EXCEPT

$x_2 x_1 x_0$

WHICH IS COVERED  
BY  $x_2 x_0$

ANY MORE IMPLICANTS? 8 MORE

LIST ALL IMPLICANTS

$m_1$ ,  $(m_2)$ ,  $m_4$ ,  $m_5$ ,  $m_7$ ,  $m_9$ ,  $m_{13}$ ,  $m_{14}$ ,  $m_{15}$ ,  $(x_1' x_0)$ ,  $(x_2 x_0)$ ,  $(x_3' x_2 x_1)$ ,  $(x_3 x_2 x_1)$

|| EVERY MINIMAL SUM OF PRODUCTS CONSISTS  
OF A SUM OF PRIME IMPLICANTS

$$E_1 = P_1 + P_2 \dots + P + \dots + P_n$$

↑  
NOT PI

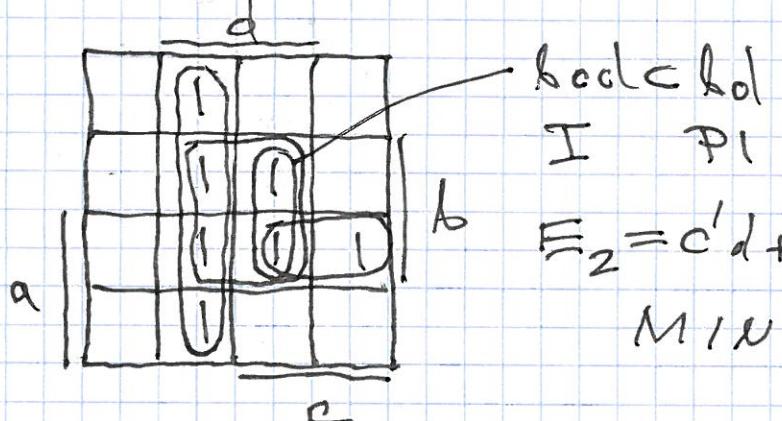
SHOW  $E_1$  IS NOT A MINIMAL SOP

SINCE  $P$  IS AN IMPLICANT, THERE IS A  
 $P_1 \subseteq$  THAT COVERS  $P$

$$E_2 = P_1 + P_2 + \dots + Q + \dots + P_n \text{ IS A MIN SOP}$$

EXAMPLE:

$$E_1 = c'd + bcd + abc$$



APPLY THIS ARGUMENT  
TO EVERY PT IN  $E$ .

NOT A PI

⇒ WILL PRODUCE  
AN EXPRESSION  
CONSISTING ONLY  
OF PIS ⇒ FEWER  
LITERALS

- A MINIMAL EXPRESSION MAY NOT CONTAIN ALL PIS.
- WHICH PIS TO INCLUDE?
- TWO CLASSES OF PIS:

**ESSENTIAL (EPI)**, **NONESSENTIAL PI**

**MUST USE**                            **MAY USE**

A PI  $p_e$  OF  $f$  IS AN EPI  
 IF THERE EXISTS AN ASSIGNMENT  $\underline{a}$   
 SUCH THAT  $p_e(\underline{a}) = 1$  AND  $p_e(\underline{a}) = 0$   
 FOR ANY OTHER PI

		<u>c</u>
		1
<u>a</u>	1	1
	0	1
		b

$$f(a, b, c) = \sum m(1, 3, 4, 6, 7)$$

PIS:  $ac'$  EPI

$a' b$

$b' c$

$a' c$  EPI

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

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

## FIND ALL PIs

---

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

	$x_0$
$x_2$	0 0 0   1
	1 0 0   0
	$\overline{x_1}$

PIs:  $x_2x'_0$  and  $x_1x'_0$

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

	$x_0$
$x_2$	1 1 0 0
	0 0 1 0
	$\overline{x_1}$

PIs:  $x'_2x'_1$ ,  $x_2x_0$ , and  $x'_1x_0$

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

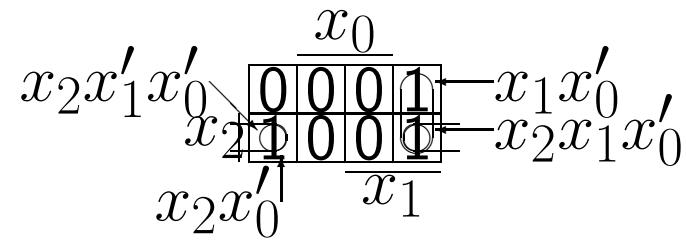
	$x_0$
1	0 1 0
0	1 1 0
1	0 1 0
0	0 1 0
	$x_1$

PIs:  $x_2x_0$ ,  $x_1x_0$ ,  $x_3x_2x'_1$ , and  $x'_3x'_2x'_1x'_0$

## Example 5.9

---

$$E(x_2, x_1, x_0) = x_2x'_1x'_0 + x_2x_1x'_0 + x_1x'_0$$



not PIs:  $x_2x'_1x'_0$  and  $x_2x_1x'_0$

PI:  $x_2x'_0$ ,  $x_1x'_0$

REDUCED SP:  $E(x_2, x_1, x_0) = x_2x'_0 + x_1x'_0$

## PROCEDURE FOR FINDING MIN SP

---

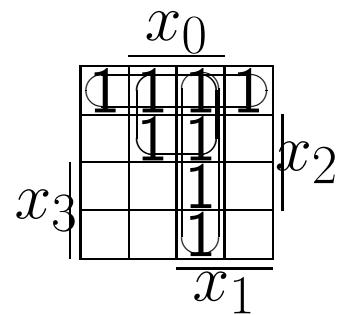
1. DETERMINE ALL PIs
2. OBTAIN THE EPIs
3. IF NOT ALL 1-CELLS COVERED, CHOOSE A COVER FROM THE REMAINING PIs

## EXAMPLE 5.10

---

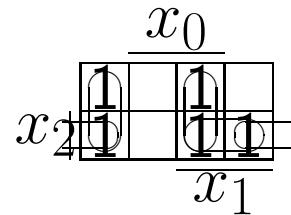
FIND A MINIMAL SP:

a)  $E(x_3, x_2, x_1, x_0) = x'_3x'_2 + x'_3x_2x_0 + x_1x_0$



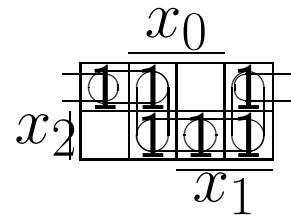
- PIs:  $x'_3x'_2$ ,  $x'_3x_0$ , and  $x_1x_0$
- ALL EPIS
- UNIQUE MIN SP:  $x'_3x'_2 + x'_3x_0 + x_1x_0$

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



- Pls:  $x'_1x'_0$ ,  $x_1x_0$ ,  $x_2x'_0$ , and  $x_2x_1$
- EPs:  $x'_1x'_0$  and  $x_1x_0$
- EXTRA COVER:  $x_2x'_0$  or  $x_2x_1$
- TWO MIN SPs:  
 $x'_1x'_0 + x_1x_0 + x_2x'_0$  and  $x'_1x'_0 + x_1x_0 + x_2x_1$

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



- Pls:  $x'_2x'_1$ ,  $x'_2x'_0$ ,  $x_2x_0$ ,  $x_2x_1$ ,  $x'_1x_0$ , and  $x_1x'_0$
- No EPs
- TWO MIN SPs

$$x'_2x'_1 + x_2x_0 + x_1x'_0 \text{ and } x'_2x'_0 + x'_1x_0 + x_2x_1$$

# MINIMAL SPs FOR INCOMPLETELY SPECIFIED FUNCTIONS<sup>32</sup>

---

		$x_0$	
		1 1 1 0	
		0 - 1 0   $x_2$	
$x_3$	1 - 0 -		
	1 0 - -		
			$x_1$

A minimal SP

$$E(x_3, x_2, x_1, x_0) = x_3x'_0 + x'_3x_0 + x'_3x'_2x'_1$$

# 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

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

			$x_0$
1	1	1	1
1	1	0	1
$x_3$	1	0	1
1	1	1	1
			$x_1$

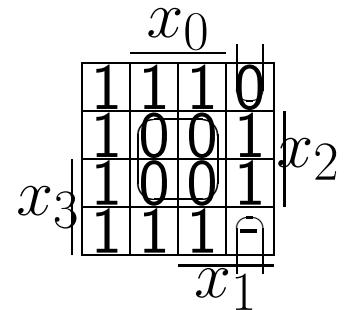
THE PRIME IMPLICATES:  $(x'_3 + x'_2 + x'_0)$  and  $(x'_2 + x'_1 + x'_0)$

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<sup>35</sup>

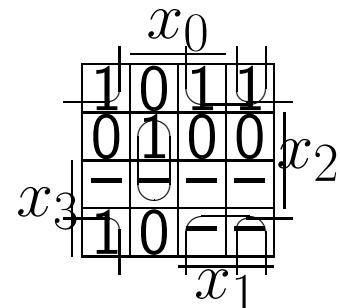
## 5.14

---

Input:  $x \in \{0, 1, 2, \dots, 9\}$ , 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:  $z = \begin{cases} 1 & \text{if } x \in \{0, 2, 3, 5, 8\} \\ 0 & \text{otherwise} \end{cases}$

THE VALUES  $\{10, 11, 12, 13, 14, 15\}$  ARE “DON’T CARES”



MIN SP:  $z = x'_2 x_1 + x'_2 x'_0 + x_2 x'_1 x_0$

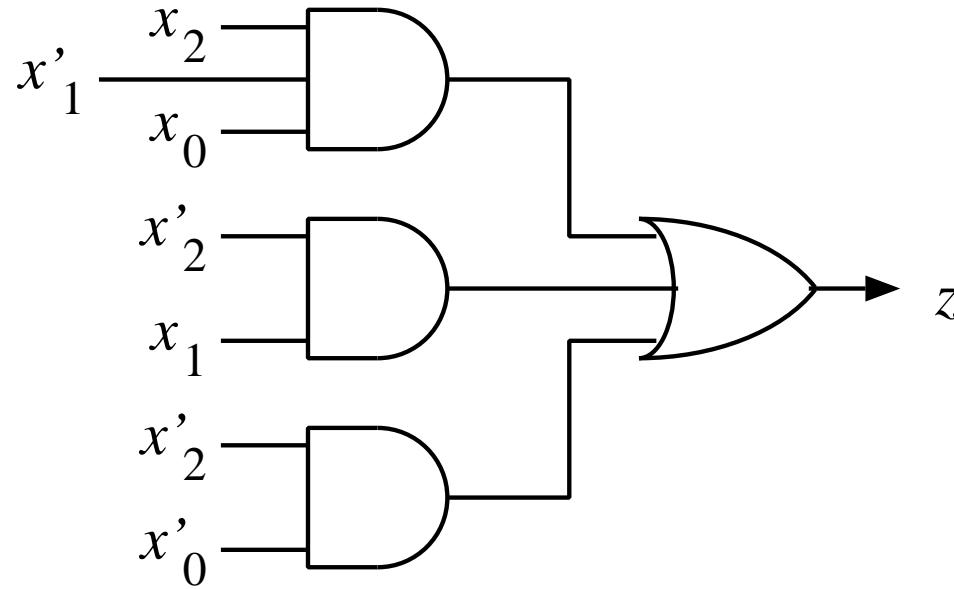


Figure 5.9: MINIMAL AND-OR NETWORK

$$\text{MIN PS: } z = (x'_2 + x'_1)(x'_2 + x_0)(x_2 + x_1 + x'_0)$$

## EXAMPLE 5.15

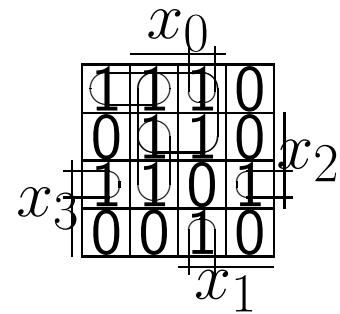
---

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

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

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

THE K-MAP:



min SP:  $z = x'_3x_0 + x'_3x'_2x'_1 + x_2x'_1x_0 + x_3x_2x'_0 + x'_2x_1x_0$   
           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)

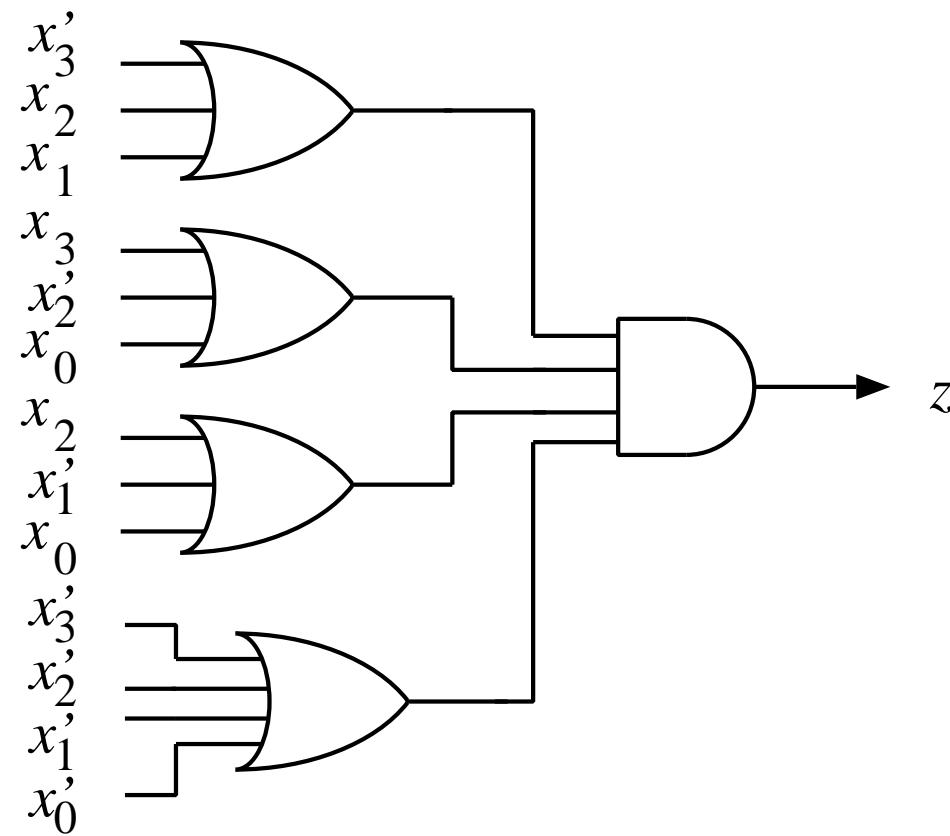


Figure 5.10: MINIMAL OR-AND NETWORK

CS M51A

## EXCESS-3 TO BCD CONVERTER

	$X-3$				$BCD$			
	$x_3$	$x_2$	$x_1$	$x_0$	$y_3$	$y_2$	$y_1$	$y_0$
0	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	1
2	0	1	0	1	0	0	1	0
3	0	1	1	0	0	0	1	1
4	0	1	1	1	0	1	0	0
5	1	0	0	0	0	1	0	1
6	1	0	0	1	0	1	1	0
7	1	0	1	0	0	1	1	1
8	1	0	1	1	1	0	0	0
9	1	1	0	0	1	0	0	1

$$d.e.-SET = (0, 1, 2, 13, 14, 15)$$

FIND PRODUCTS OF  
SUMS AND COMPARE!

$$y_0 = x_0'$$

$y_1:$	$\frac{x_0}{x_0}$
-	0
0	1
0	-
0	1

$$y_1 = x_1 \oplus x_0$$

$y_2:$	$\frac{x_0}{x_0}$
-	0
0	0
0	-
1	0

$$y_2 = x_2' x_0' + x_2 x_1 x_0 + x_3 x_1' x_0$$

$y_3:$	$\frac{x_0}{x_0}$
-	0
0	0
1	-
0	0

$$y_3 = x_3 x_2 + x_3 x_1 x_0$$

CS M51A

EXAMPLE: QUOTIENT GENERATOR

$$x \in \{0, 1, \dots, 15\} \quad q = \lfloor \frac{x}{6} \rfloor \quad q \in \{0, 1, 2\} \quad \varepsilon = (q_1, q_0)$$

NOTE:  $x \leftrightarrow m_i(x_3, x_2, x_1, x_0)$

$$x \in \{m_0, m_1, m_2, m_3, m_4, m_5\} \quad q = 0 \quad q_1 = 0 \quad q_0 = 0$$

$$x \in \{m_6, \dots, m_9\} \quad q = 1 \quad q_1 = 0 \quad q_0 = 1$$

$$x \in \{m_{12}, \dots, m_{15}\} \quad q = 2 \quad q_1 = 1 \quad q_0 = 0$$

$x_0$				
0	1	3	2	
0	0	0	0	
4	5	7	6	
0	0	0	0	
12	13	15	14	$x_2$
(1)	1	1	1	
8	9	4	10	
0	0	0	0	

$x_0$				
0	0	0	0	
0	0	(1)	(1)	
0	0	0	0	$x_2$
(1)	1	1	1	

$$q_1 = x_3 x_2$$

$$\begin{aligned} q_0 &= x_3 x_2' + x_3' x_2 x_1 \quad \text{W/ 3 GATES, 7 INPUTS} \\ &= (x_3 + x_2)(x_3' + x_2')(x_2' + x_1) \quad \text{4 GATES, 9 INPUTS} \end{aligned}$$

$$x, y \in \{0, 1, 2, 3\} \quad p = x \cdot y \in \{0, 1, 2, 3, 4, 6, 9\}$$

$$x = (x_1, x_0) \quad y = (y_1, y_0) \quad p = (p_3, p_2, p_1, p_0)$$

[ THROUTH TABLE  $\rightarrow SF_0 \rightarrow \sum_m \rightarrow K$ -MAPS  $\rightarrow$  min SOP ]

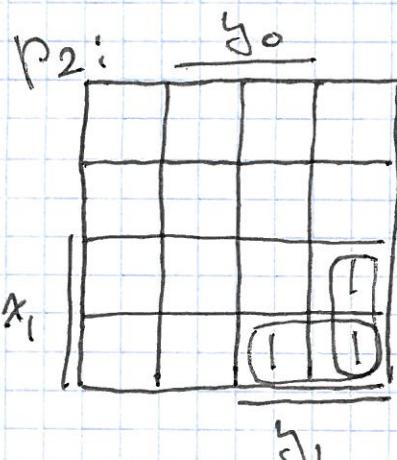
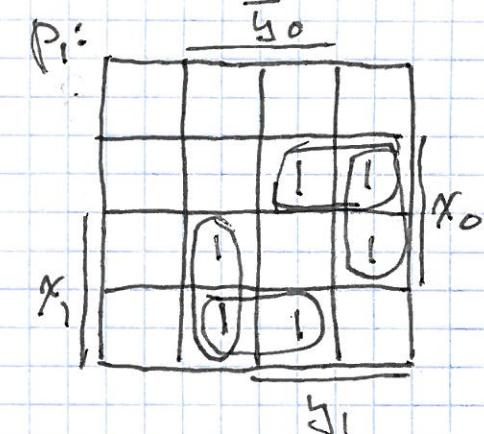
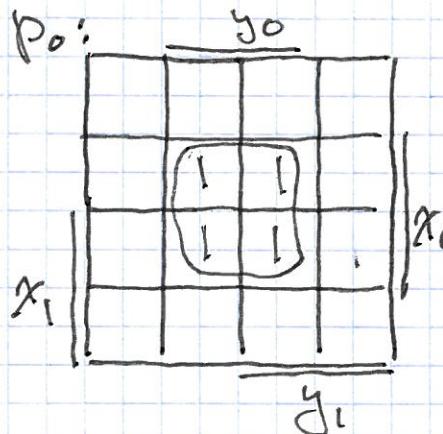
$x$	$x_1$	$x_0$	$y_1$	$y_0$	$y$	$p$	$p_3$	$p_2$	$p_1$	$p_0$
0	0	0	0	0	0	0	0	0	0	0
	0	1	1	0	1	0	0	0	0	0
	1	0	2	0	2	0	0	0	0	0
	1	1	3	0	3	0	0	0	0	0
<hr/>	1	0	1	0	0	0	0	0	0	0
	0	1	1	1	1	1	0	0	0	1
	1	0	2	2	2	2	0	0	1	0
	1	1	3	3	3	3	0	0	1	1
<hr/>	2	1	0	0	0	0	0	0	0	0
	0	1	1	2	2	2	0	0	1	0
	1	0	2	4	4	4	0	1	0	0
	1	1	3	6	6	6	0	1	1	0
<hr/>	3	1	1	0	0	0	0	0	0	0
	0	1	1	3	3	3	0	0	1	1
	1	0	2	6	6	6	0	1	1	0
	1	1	3	9	1	9	0	0	0	1

$$p_0 = \sum_m (5, 7, 13, 15)$$

$$p_1 = \sum_m (6, 7, 9, 11, 13, 14)$$

$$p_2 = \sum_m (10, 11, 14)$$

$$p_3 = m_{15} = x_1 x_0 y_1 y_0$$



$$p_2 = x_1 x_0' y_1 + x_1 y_1 y_0'$$

2-AND 1  
3-AND 6  
2-OR 1  
4-OR 1  
4-AND 1  
No. of INPUTS:

10 GATES

$$2 + 16 + 8 + 4 = 30$$

# DESIGN OF MULTIPLE-OUTPUT TWO-LEVEL GATE NETWORKS

---

- SEPARATE NETWORK FOR EACH OUTPUT: NO SHARING
- EXAMPLE 5.16

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

# 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

$z_1$	$\underline{x_0}$								
$x_2$	<table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td></tr> </table>	0	0	1	0	0	1	1	1
0	0	1	0						
0	1	1	1						
	$\underline{x_1}$								

$z_0$	$\underline{x_0}$								
$x_2$	<table border="1"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	1	0	1	1	0	1	0
0	1	0	1						
1	0	1	0						
	$\underline{x_1}$								

3. MINIMAL SPs:

$$z_1 = x_2x_1 + x_2x_0 + x_1x_0$$

$$z_0 = x'_2x'_1x_0 + x'_2x_1x'_0 + x_2x'_1x'_0 + x_2x_1x_0$$

4. MINIMAL PSs:

$$z_1 = (x_2 + x_0)(x_2 + x_1)(x_1 + x_0)$$

$$\begin{aligned} z_0 = & (x_2 + x_1 + x_0)(x_2 + x'_1 + x'_0) \\ & (x'_2 + x_1 + x'_0)(x'_2 + x'_1 + x_0) \end{aligned}$$

5. SP AND PS EXPRESSIONS HAVE THE SAME COST

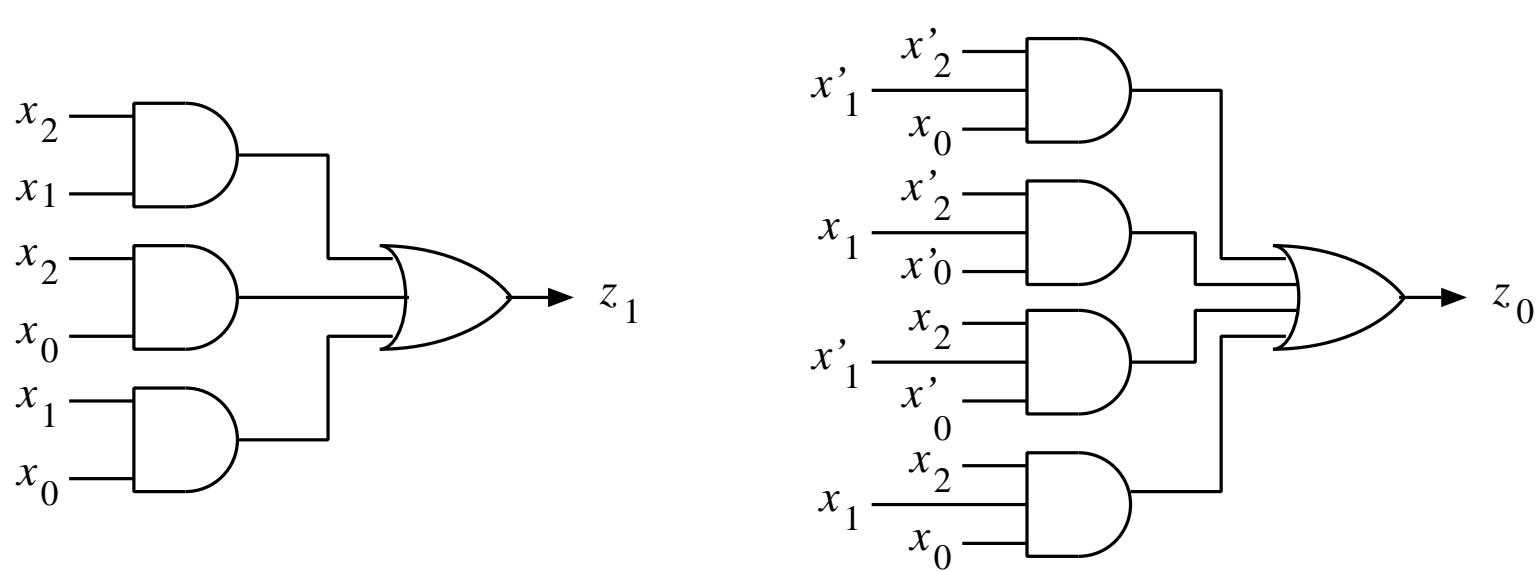


Figure 5.11: MINIMAL TWO-OUTPUT AND-OR NETWORK

## TWO-LEVEL NAND-NAND AND NOR-NOR NETWORKS

---

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

$p_1, p_2, \dots$  ARE PRODUCT TERMS

$$E = (p'_1 \cdot p'_2 \cdot p'_3 \dots p'_n)'$$

or

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

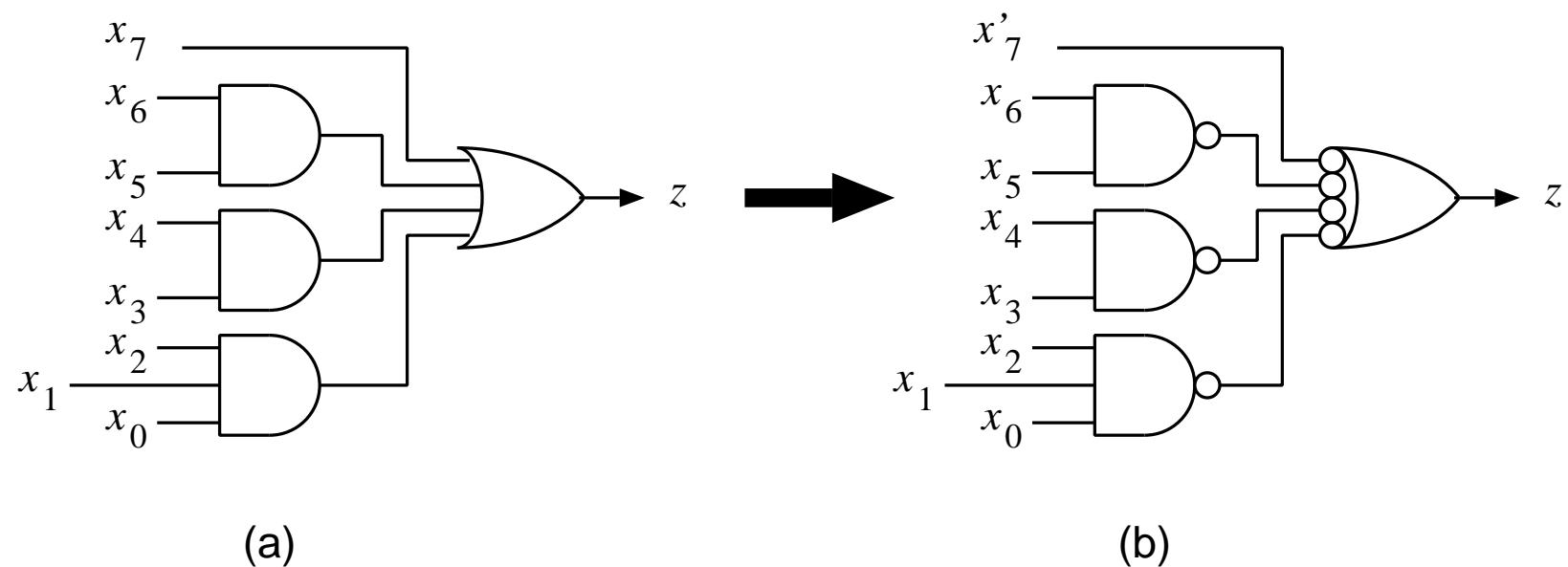


Figure 5.15: TRANSFORMATION OF AND-OR NETWORK INTO NAND-NAND NETWORK

## EXAMPLE: NOR NETWORK

---

$$z = x'_5(x_4 + x'_3)(x_2 + x_1 + x_0)$$

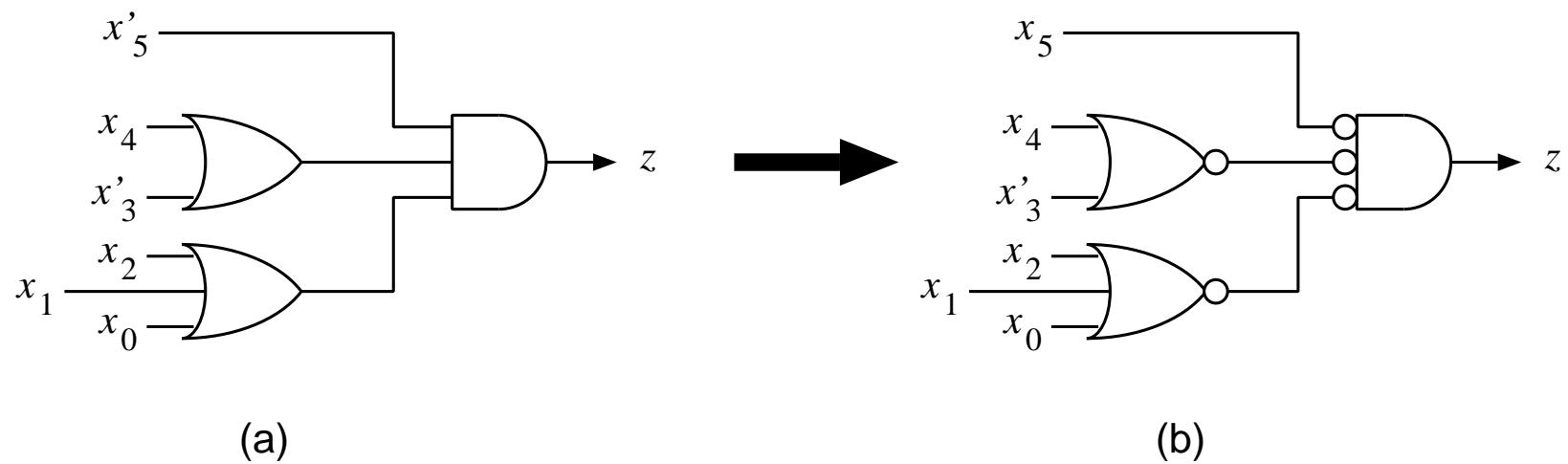


Figure 5.16: EQUIVALENT OR-AND AND NOR-NOR NETWORKS

## LIMITATIONS OF TWO-LEVEL NETWORKS

---

1. THE REQUIREMENT OF UNCOMPLEMENTED AND COMPLEMENTED INPUTS  
IF NOT SATISFIED, AN ADDITIONAL LEVEL OF  $\text{NOT}$  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

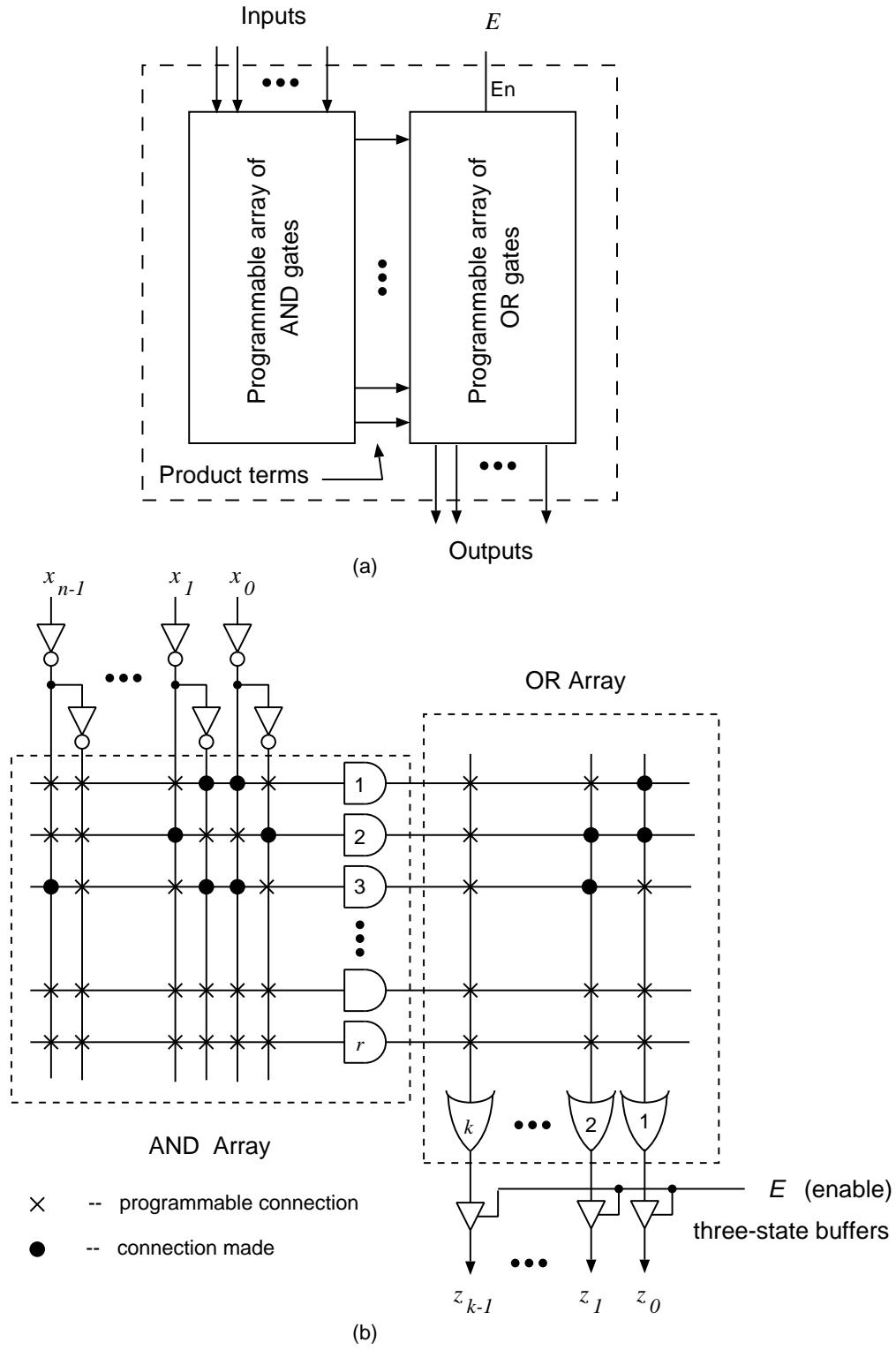
## 5. THE COST CRITERION OF MINIMIZING THE NUMBER OF GATES IS NOT ADEQUATE FOR MANY MSI/LSI/VLSI DESIGNS

# PROGRAMMABLE MODULES: PLAs

---

- 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)

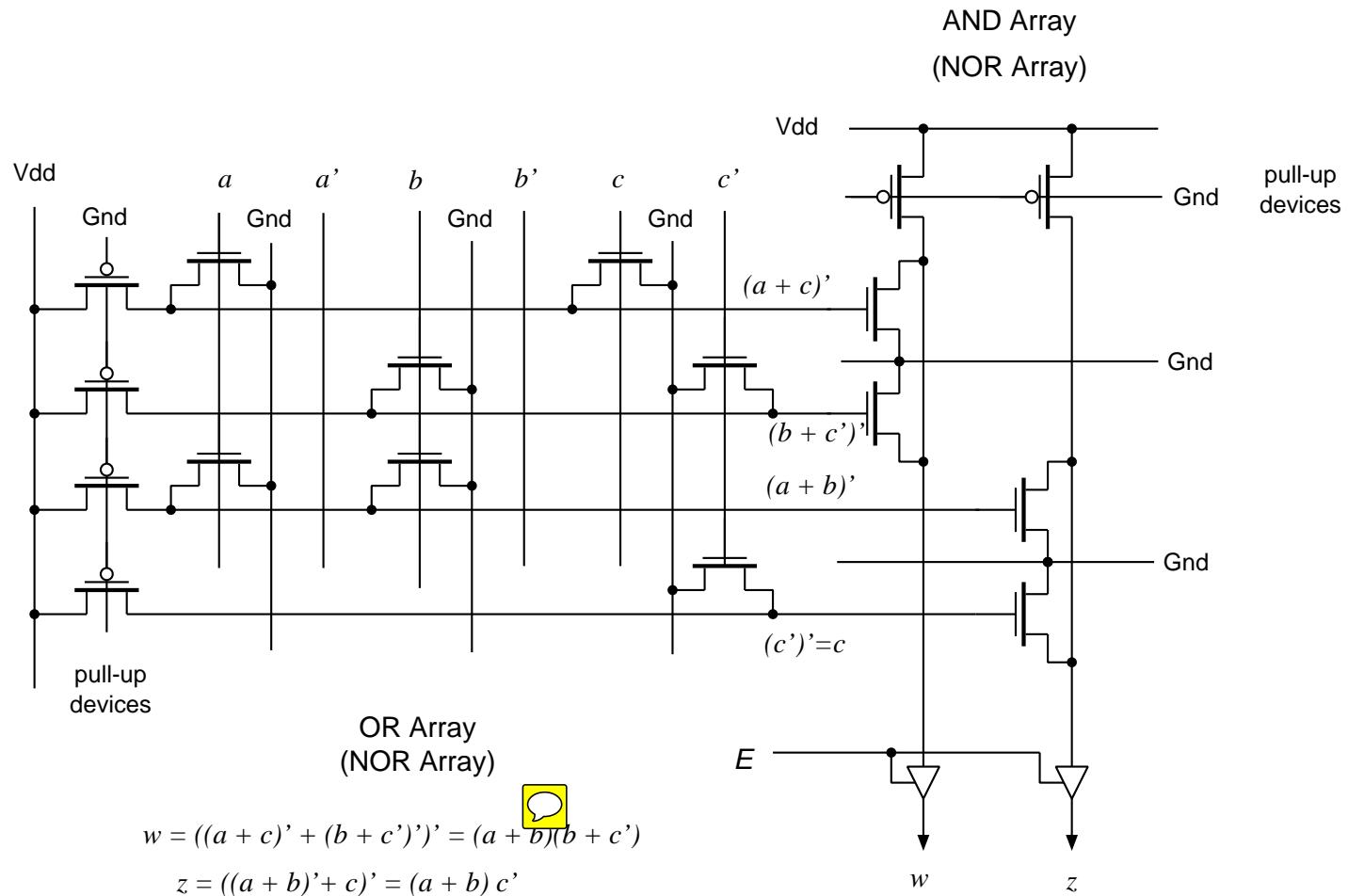


Figure 5.18: EXAMPLE OF PLA IMPLEMENTATION AT THE CIRCUIT LEVEL: FRAGMENT OF A MOS PLA .

# IMPLEMENTATION OF SWITCHING FUNCTIONS USING PLAs

---

## A BCD-to-Gray CONVERTER

Inputs:  $\underline{d} = (d_3, d_2, d_1, d_0)$ ,  $d_j \in \{0, 1\}$

Outputs:  $\underline{g} = (g_3, g_2, g_1, g_0)$ ,  $g_j \in \{0, 1\}$

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

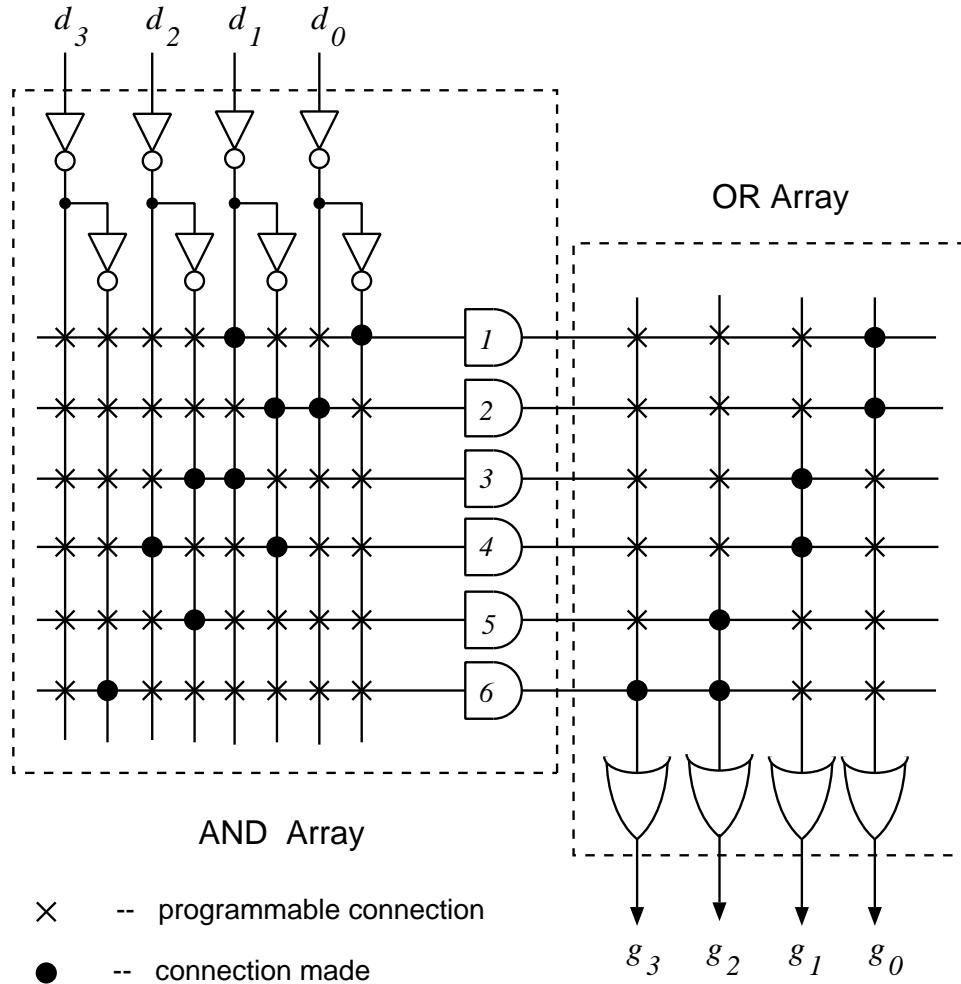
EXPRESSIONS:

$$g_3 = d_3$$

$$g_2 = d_3 + d_2$$

$$g_1 = d'_2d_1 + d_2d'_1$$

$$g_0 = d_1d'_0 + d'_1d_0$$



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

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

# FIELD PROGRAMMABLE GATE ARRAYS (FPGA)

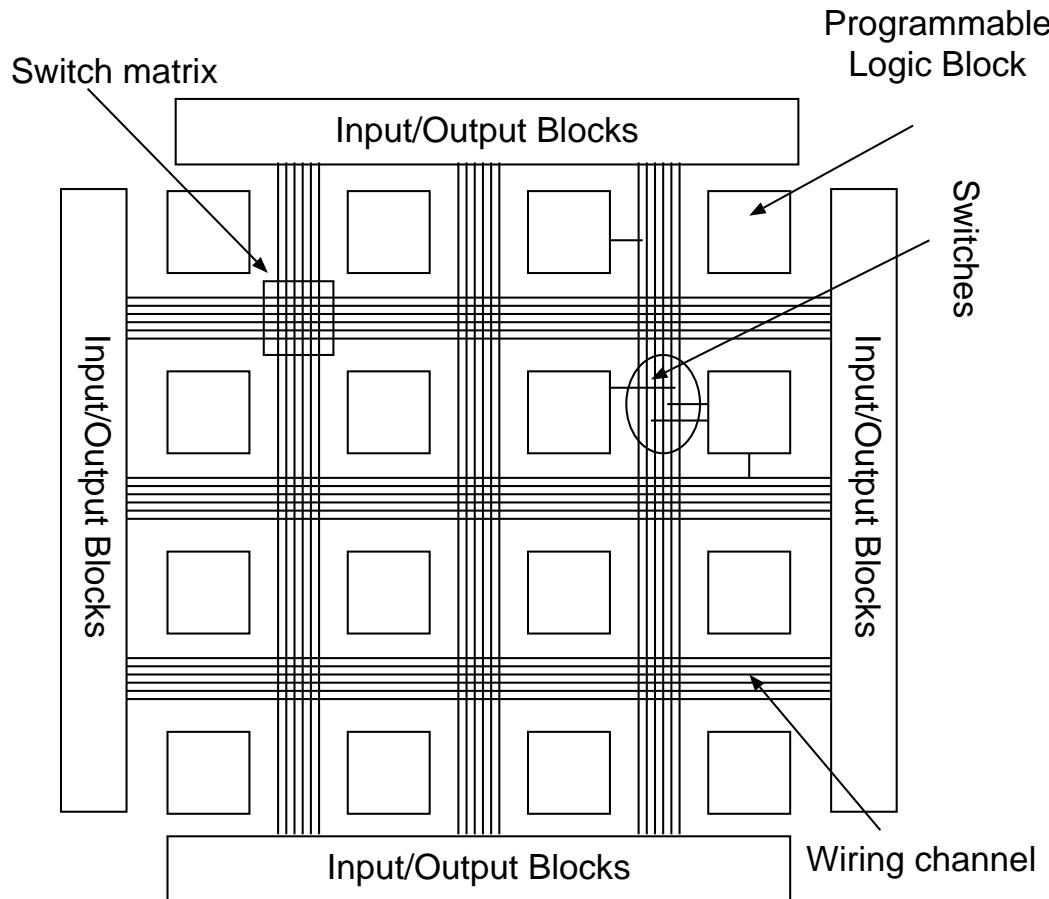


Figure 12.11: ORGANIZATION OF AN FPGA CHIP.

# BASIC APPROACHES IN PROGRAMMING OF FPGAs

---

- ON-CHIP STATIC RAM LOADED WITH CONFIGURATION BIT PATTERNS (SRAM-FPGA). (volatile)
- ANTIFUSE-PROGRAMMED DEVICES PROGRAMMED ELECTRICALLY TO PROVIDE CONNECTIONS THAT DEFINE CHIP CONFIGURATION
- ARRAY-STYLE EPROM and EEPROM PROGRAMMED DEVICES  
USING SEVERAL PLAs AND A SHARED INTERCONNECT MECHANISM

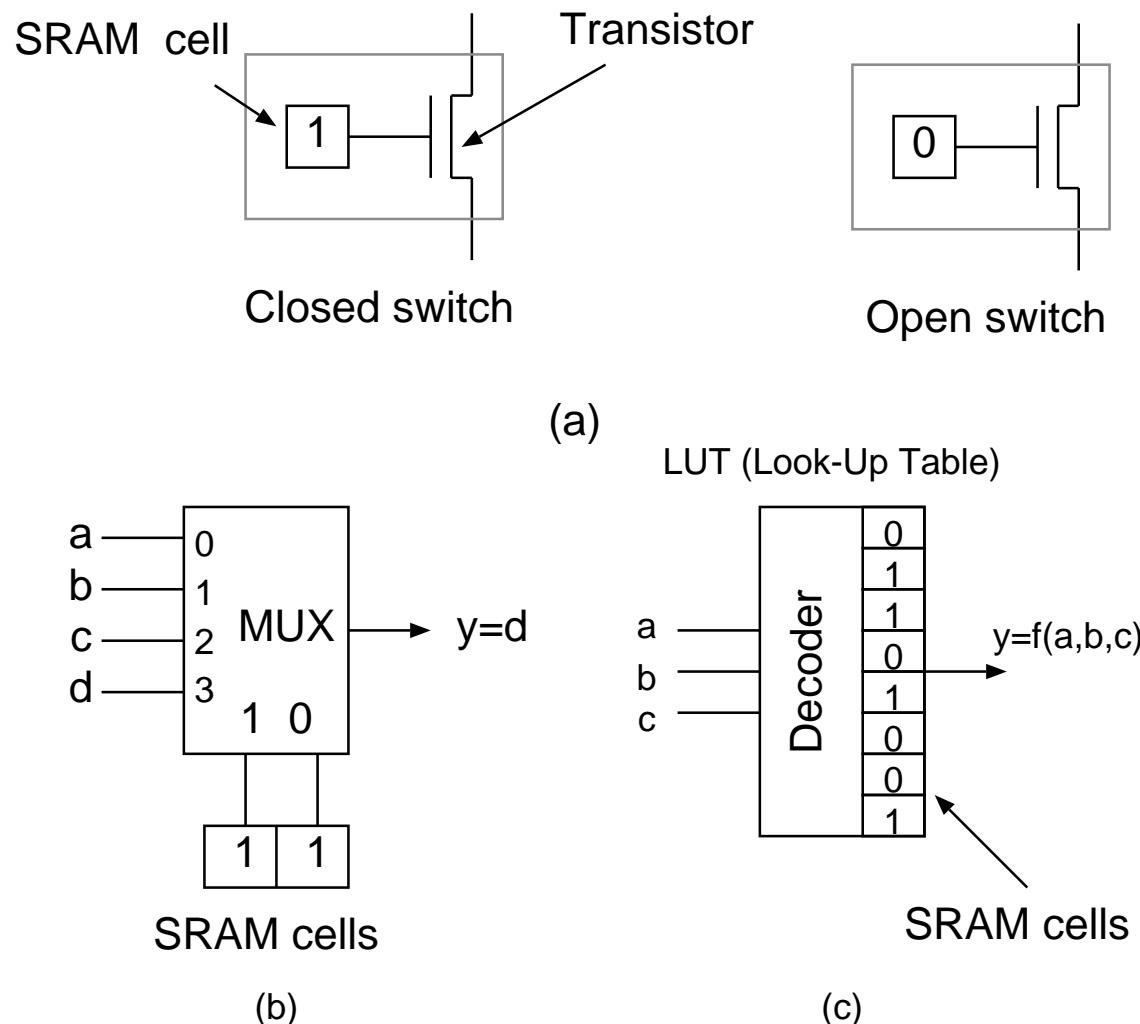


Figure 12.12: SRAM FPGA PROGRAMMABLE COMPONENTS: (a) Switch. (b) 4-input multiplexer. (c) Look-up table (LUT).

# Example: XILINX XC2000

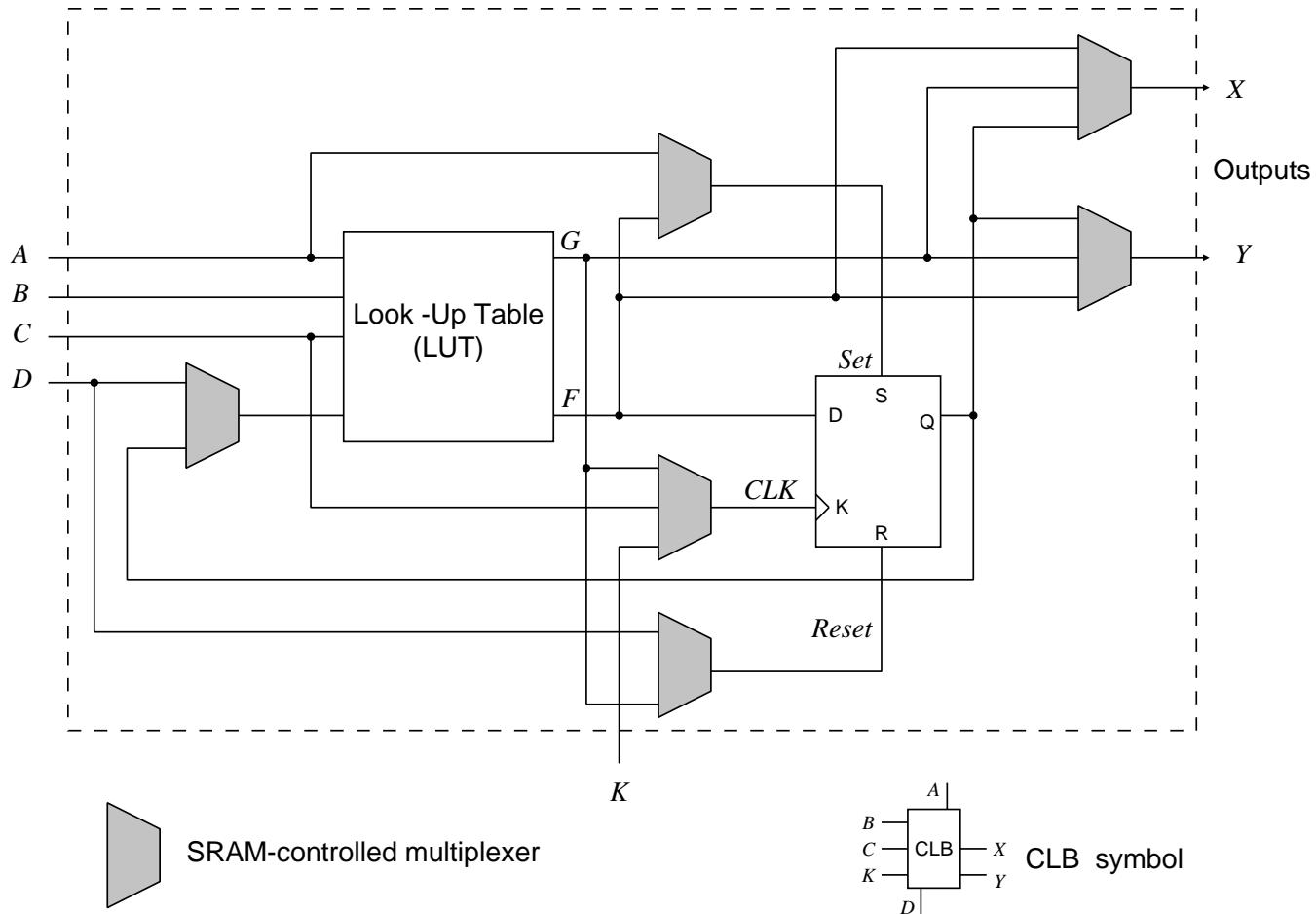


Figure 12.13: A CONFIGURABLE LOGIC BLOCK (CLB) (Courtesy of Xilinx, Inc.)

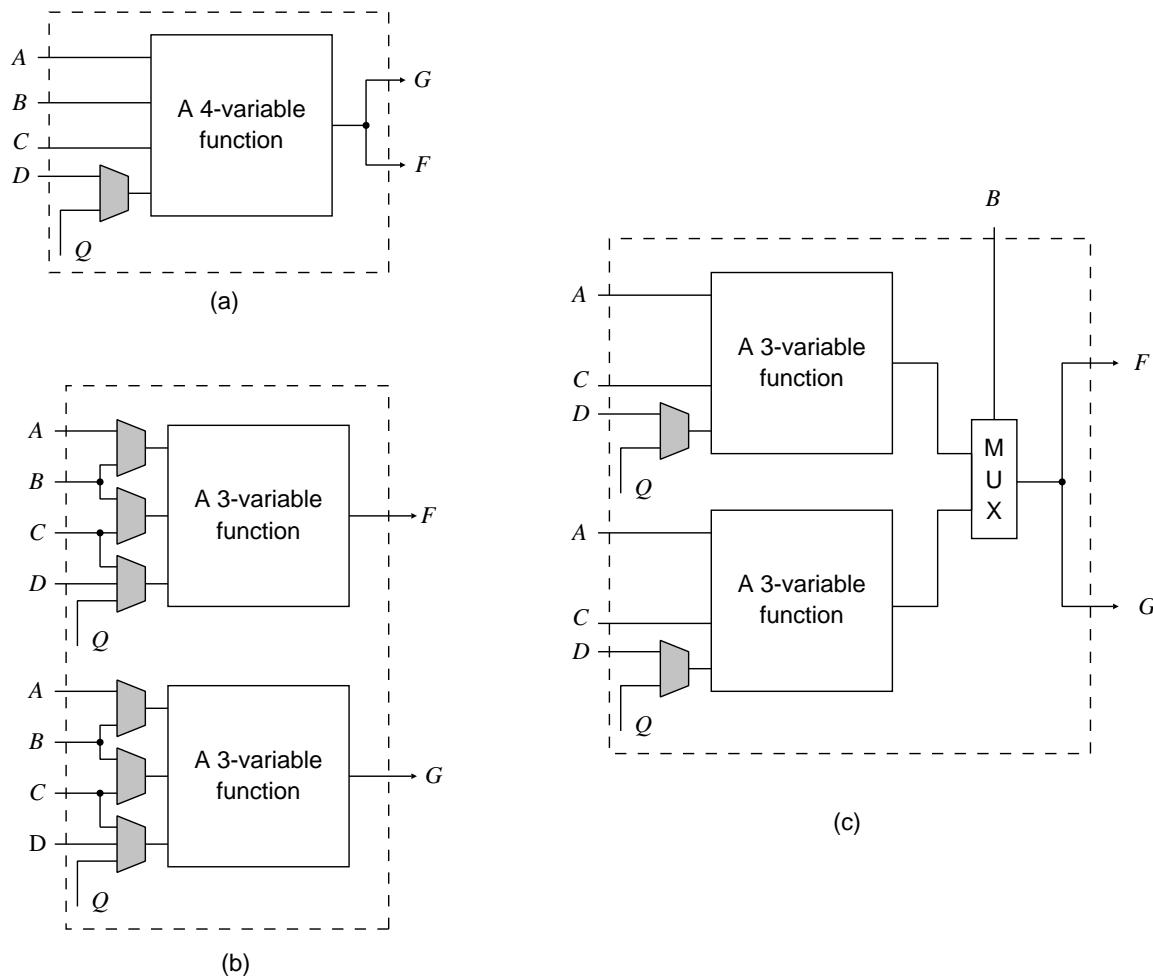


Figure 12.14: SRAM-FPGA options in generating functions: (a) One 4-variable function. (b) Two 3-variable functions. (c) Selection between two functions of 3 variables. (Courtesy of Xilinx, Inc.)

# PROGRAMMABLE INTERCONNECT

---

1. DIRECT INTERCONNECTIONS BETWEEN HORIZONTALLY AND VERTICALLY ADJACENT CLBS – PROVIDE FAST SIGNAL PATHS BETWEEN ADJACENT MODULES
2. GENERAL-PURPOSE INTERCONNECT CONSISTS OF VERTICAL AND HORIZONTAL WIRING SEGMENTS BETWEEN SWITCH MATRICES
3. LONG VERTICAL AND HORIZONTAL LINES SPAN THE WHOLE CLB ARRAY

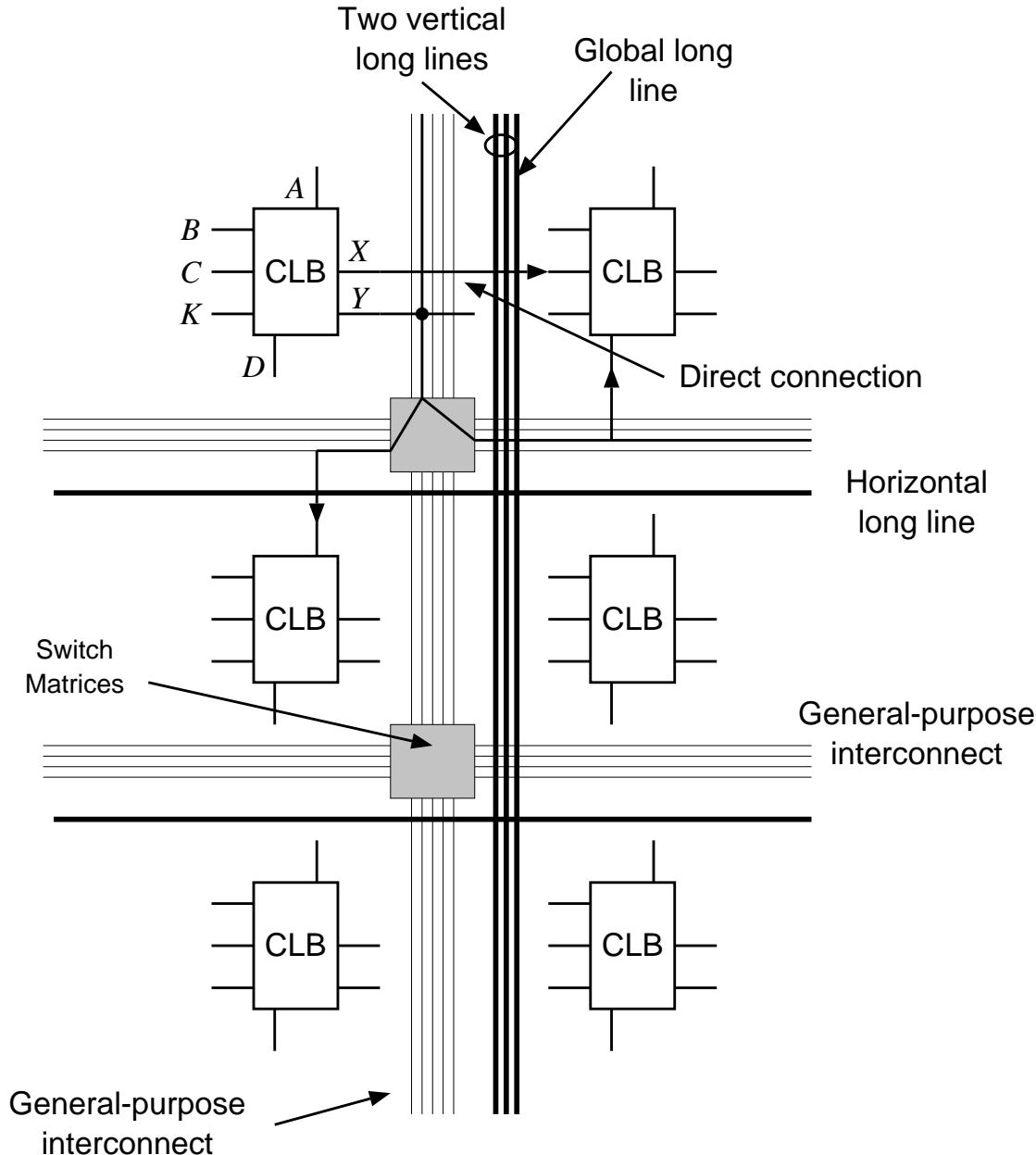


Figure 12.15: PROGRAMMABLE INTERCONNECT. (Courtesy of Xilinx, Inc.)

## Example 12.5: BCD ADDER MODULE

---

- IMPLEMENT A ONE-DIGIT BCD ADDER USING A SRAM-FPGAMODULE OF XC2000 TYPE

INPUTS:  $\underline{x} = (x_3, x_2, x_1, x_0)$ ,  $x_j \in \{0, 1\}$ ,  $x \in \{0, \dots, 9\}$

$\underline{y} = (y_3, y_2, y_1, y_0)$ ,  $y_j \in \{0, 1\}$ ,  $y \in \{0, \dots, 9\}$

$c_{in} \in \{0, 1\}$

OUTPUTS:  $\underline{s} = (s_3, s_2, s_1, s_0)$ ,  $s_j \in \{0, 1\}$ ,  $s \in \{0, \dots, 9\}$

$c_{out} \in \{0, 1\}$

FUNCTION:  $x + y + c_{in} = 10c_{out} + s$

- COMPUTE  $16u + v = x + y + c_{in} \in \{0, \dots, 19\}$  using a 4-bit binary adder

## Example 12.5 (cont.)

---

- THREE CASES:

$$u = 0 \quad v \leq 9 \quad s = v \quad c_{out} = 0$$

$$u = 0 \quad v > 9 \quad s = v - 10 = (v + 6) \bmod 16 \quad c_{out} = 1$$

$$u = 1 \quad s = v + 16 - 10 = v + 6 \quad c_{out} = 1$$

$\Rightarrow$  BCD OUTPUT

$$s = \begin{cases} (v + 6) \bmod 16 & \text{if } u = 1 \text{ or } v \geq 10 \\ v & \text{otherwise} \end{cases}$$

$$c_{out} = \begin{cases} 1 & \text{if } u = 1 \text{ or } v \geq 10 \\ 0 & \text{otherwise} \end{cases}$$

cont.

---

THE CONDITION  $u = 1$  or  $v \geq 10$  CORRESPONDS TO  
SWITCHING EXPRESSION

$$t = u + v_3v_2 + v_3v_1$$

## Example 12.5 (cont.)

---

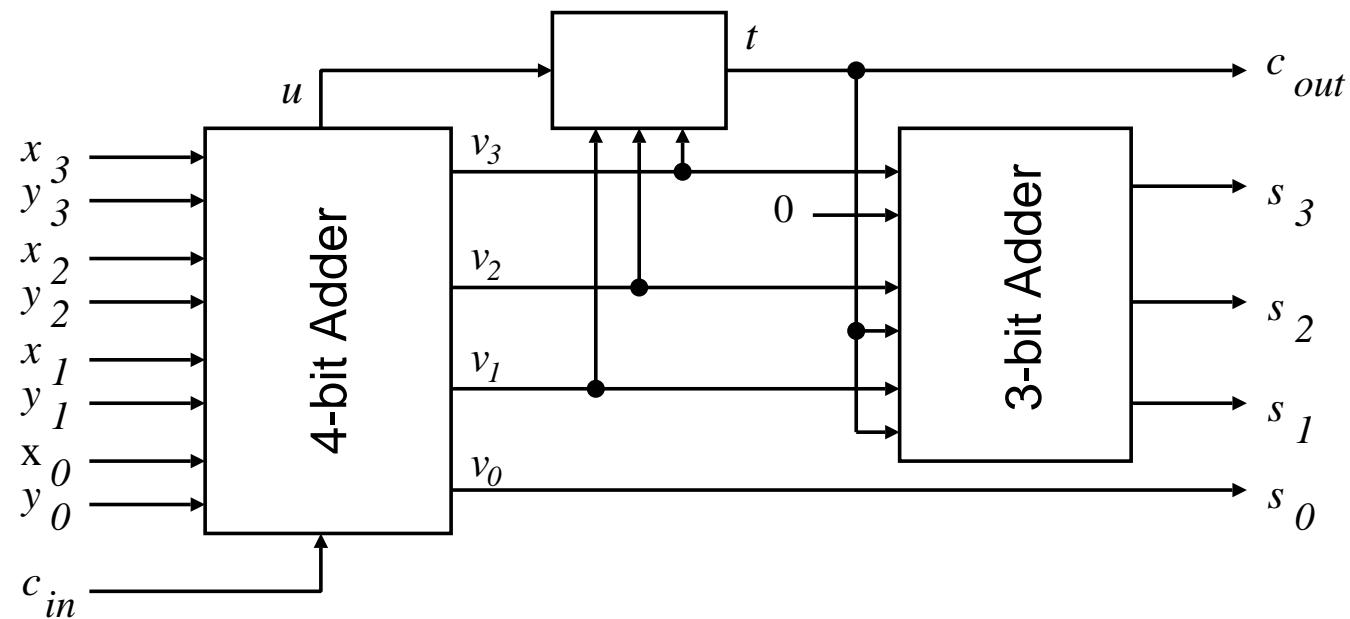


Figure 12.16: IMPLEMENTATION OF BCD ADDER MODULE

## Example 12.5 (cont.)

---

- SIMPLIFICATION OF THE 3-BIT ADDER

$$s_3 = v_3 \oplus t(v_2 + v_1)$$

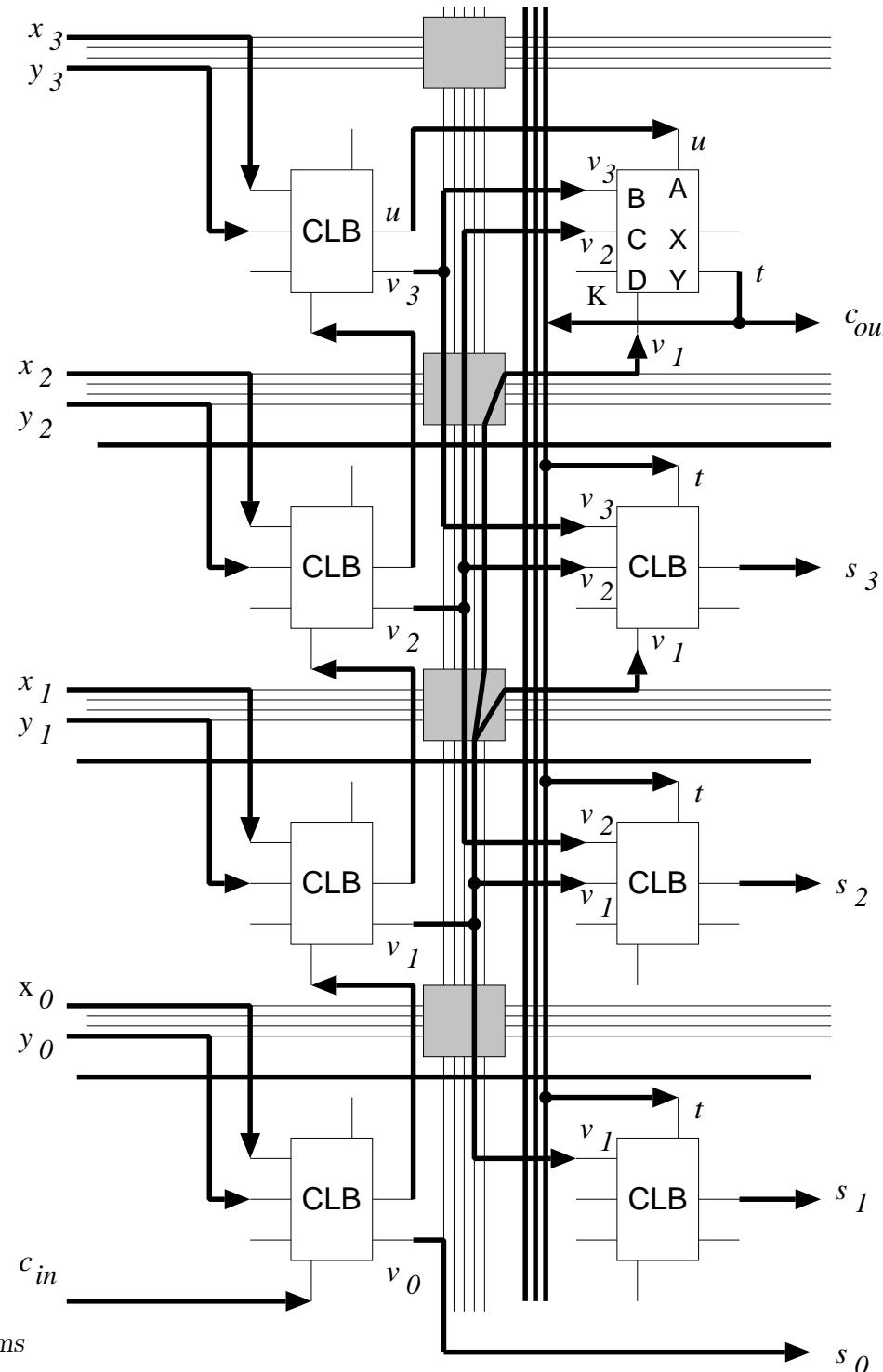
$$s_2 = v_2 \oplus tv'_1$$

$$s_1 = v_1 \oplus t$$

MOREOVER,

$$s_0 = v_0$$

$$c_{out} = t$$



## DESIGN WITH FPGAs

---

INVOLVES INTENSIVE USE OF CAD TOOLS AND MODULE LIBRARIES

**Design entry :** A SCHEMATIC ENTRY OR A BEHAVIORAL DESCRIPTION

**Implementation :**

- PARTITION OF DESIGN INTO SUBMODULES THAT CAN BE MAPPED ONTO CLBs,
- PLACEMENT OF SUBMODULES ONTO CHIP, AND
- ROUTING OF SIGNALS TO CONNECT THE SUBMODULES