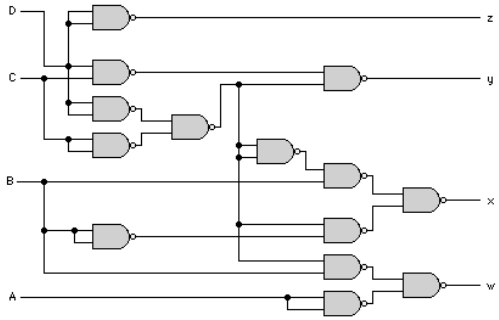
Embedded Systems Design, Spring 2025

Lecture 2

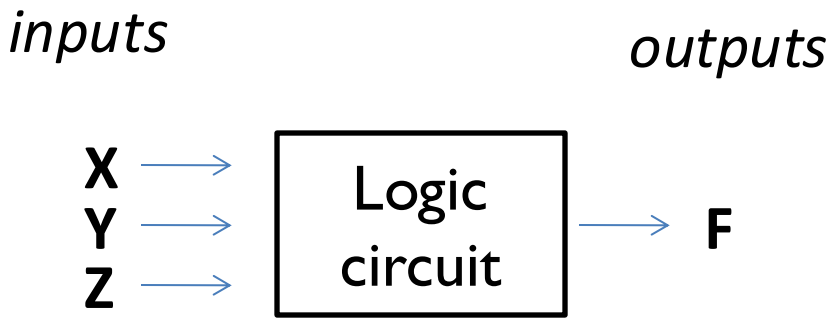


Combinational Logic Design Principles

Review from last time

- ?

Types of logic circuits



Combinational circuit: output depends on the input, operation described by a truth table.

Sequential circuit: output depends on input and previous inputs, operation described by a state table.

Logic circuits

- **Circuit analysis**:
  - Start with a logic diagram
  - Formal description of the circuit:
    - Truth table
    - Logic expression
- **Circuit synthesis**:
  - Start with a formal description
    - Truth table
    - Logic expression
  - Continue with a logic diagram

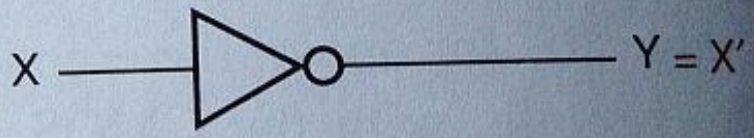
Switching algebra

- George Boole, 1854 – invented boolean algebra
- Claude Shannon, 1938 – switching algebra (using relays)
  - Variable X could have two values: 0 or 1
- Positive logic convention (opposite for negative logic convention)
  - 0 – low voltage
  - 1 – high voltage
- Axioms:
  - basic definitions assumed to be true
  - Everything else can be derived from them

Axioms

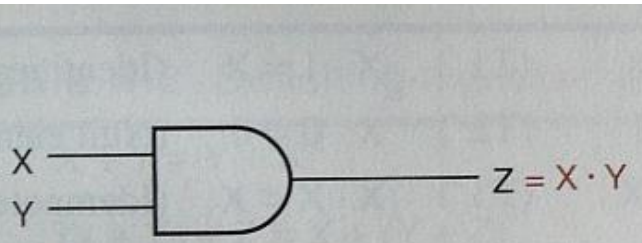
Axiom	Relation	Axiom'	Relation
A1	$X = 0$ if $X \neq 1$	A1'	$X = 1$ if $X \neq 0$
A2	If $X = 0$, then $X' = 1$	A2'	If $X = 1$, then $X' = 0$

Figure 4-1
Signal naming and algebraic notation for an inverter.



- X' is the same as $\neg X$, $\sim X$, $\overline{X}$

AND (·) , OR (+)



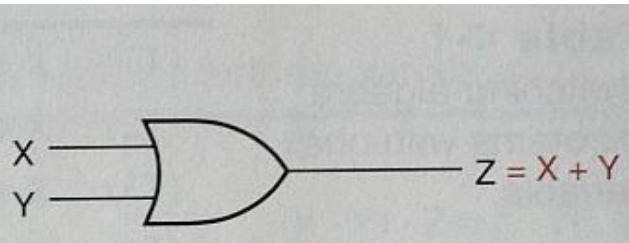


Figure 4-2
Signal naming and algebraic notation:
(a) AND gate;
(b) OR gate.

Axiom	Relation	Axiom'	Relation
A3	0 · 0 = 0	A3'	1 + 1 = 1
A4	1 · 1 = 1	A4'	0 + 0 = 0
A5	0 · 1 = 1 · 0 = 0	A5'	1 + 0 = 0 + 1 = 1

Single variable theorems

T	Relation	T	Relation	Name
T1	$X + 0 = X$	(T1)'	$X \cdot 1 = X$	Identities
T2	$X + 1 = 1$	(T2)'	$X \cdot 0 = 0$	Null elements
T3	$X + X = X$	(T3)'	$X \cdot X = X$	Idempotency
T4	$(X')' = X$	(T4)'		Involution
T5	$X + X' = 1$	(T5)'	$X \cdot X' = 0$	Complements

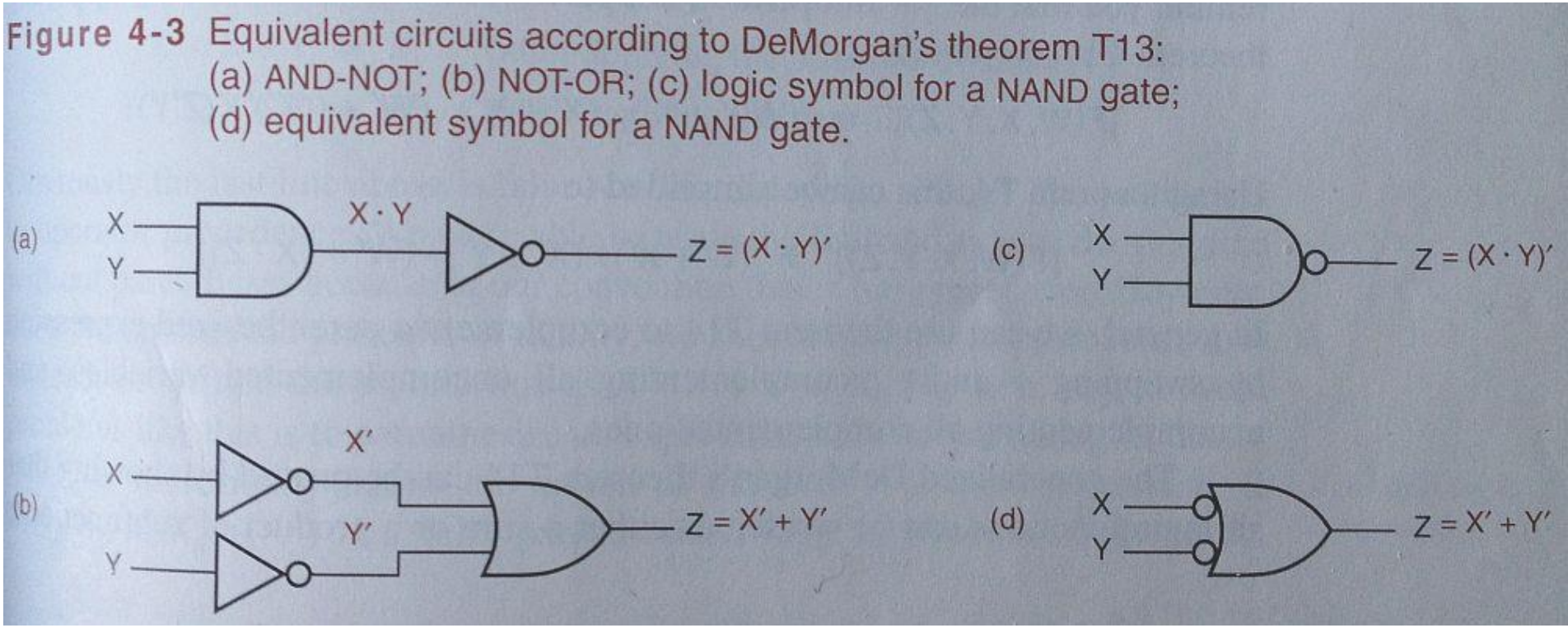
Multiple variable theorems

T	Relation	T	Relation	Name
T6	$X + Y = Y + X$	(T6)'	$X \cdot Y = Y \cdot X$	Commutativity
T7	$(X + Y) + Z = X + (Y + Z)$	(T7)'	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	Associativity
T8	$X \cdot Y + X \cdot Z = X \cdot (Y+Z)$	(T8)'	$(X + Y) \cdot (X + Z) = X+Y \cdot Z$	Distributivity
T9	$X + X \cdot Y = X$	(T9)'	$X \cdot (X + Y) = X$	Covering
T10	$X \cdot Y + X \cdot Y' = X$	(T10)'	$(X + Y) \cdot (X + Y') = X$	Combining
T11	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$			Consensus
T11'	$(X + Y) \cdot (X'+Z) \cdot (Y+Z) = (X + Y) \cdot (X' + Z)$			

More theorems

T	Relation	T	Relation	Name
T12	$X + X + \ldots + X = X$			Generalized idemp.
T12'	$X \cdot X \cdot \ldots \cdot X = X$			
T13	$(X_1 \cdot X_2 \cdot \ldots \cdot X_n)' = X_1' + X_2' + \ldots + X_n'$			**DeMorgan's theorems**
T13'	$(X_1 + X_2 + \ldots + X_n)' = X_1' \cdot X_2' \cdot \ldots \cdot X_n'$			
T14	$[F(X_1, X_2, \ldots, X_n, +, \cdot)]' = F(X_1', X_2', \ldots, X_n', \cdot, +)$			Generalized DeMorgan's th.
T15	$F(X_1, X_2, \ldots, X_n) = X_1 \cdot F(1, X_2, \ldots, X_n) + X_1' \cdot F(0, X_2, \ldots, X_n)$			Shannon's expansion

DeMorgan's theorems in use:



DeMorgan's theorems in use:

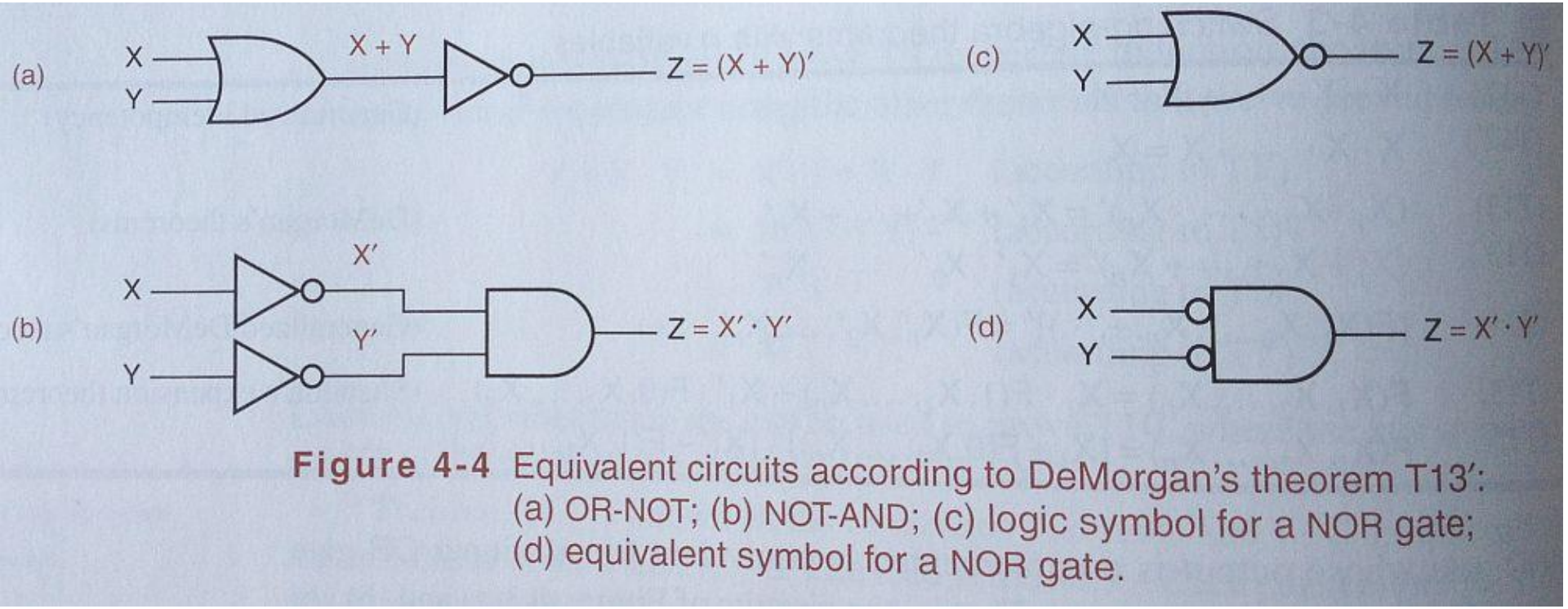
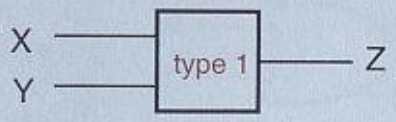


Figure 4-4 Equivalent circuits according to DeMorgan's theorem T13′: (a) OR-NOT; (b) NOT-AND; (c) logic symbol for a NOR gate; (d) equivalent symbol for a NOR gate.

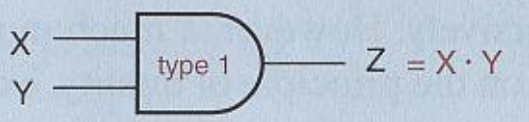
Example 1:

(a)



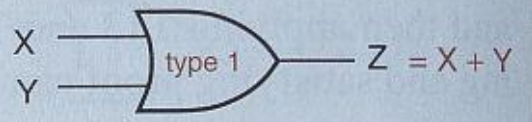
X	Y	Z
LOW	LOW	LOW
LOW	HIGH	LOW
HIGH	LOW	LOW
HIGH	HIGH	HIGH

(b)



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

(c)



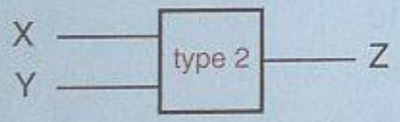
X	Y	Z
1	1	1
1	0	1
0	1	1
0	0	0

Figure 4-5 A "type-1" logic gate: (a) electrical function table; (b) logic function table and symbol with positive logic; (c) logic function table and symbol with negative logic.

Example 2

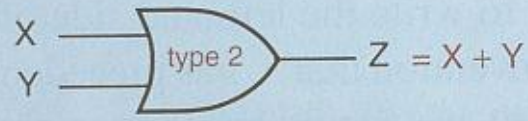
Figure 4-6 A "type-2" logic gate: (a) electrical function table; (b) logic function table and symbol with positive logic; (c) logic function table and symbol with negative logic.

(a)



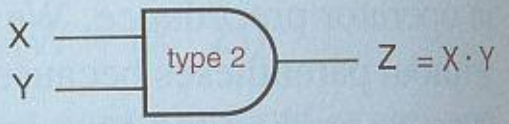
X	Y	Z
LOW	LOW	LOW
LOW	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	HIGH	HIGH

(b)



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

(c)



X	Y	Z
1	1	1
1	0	0
0	1	0
0	0	0

Duality

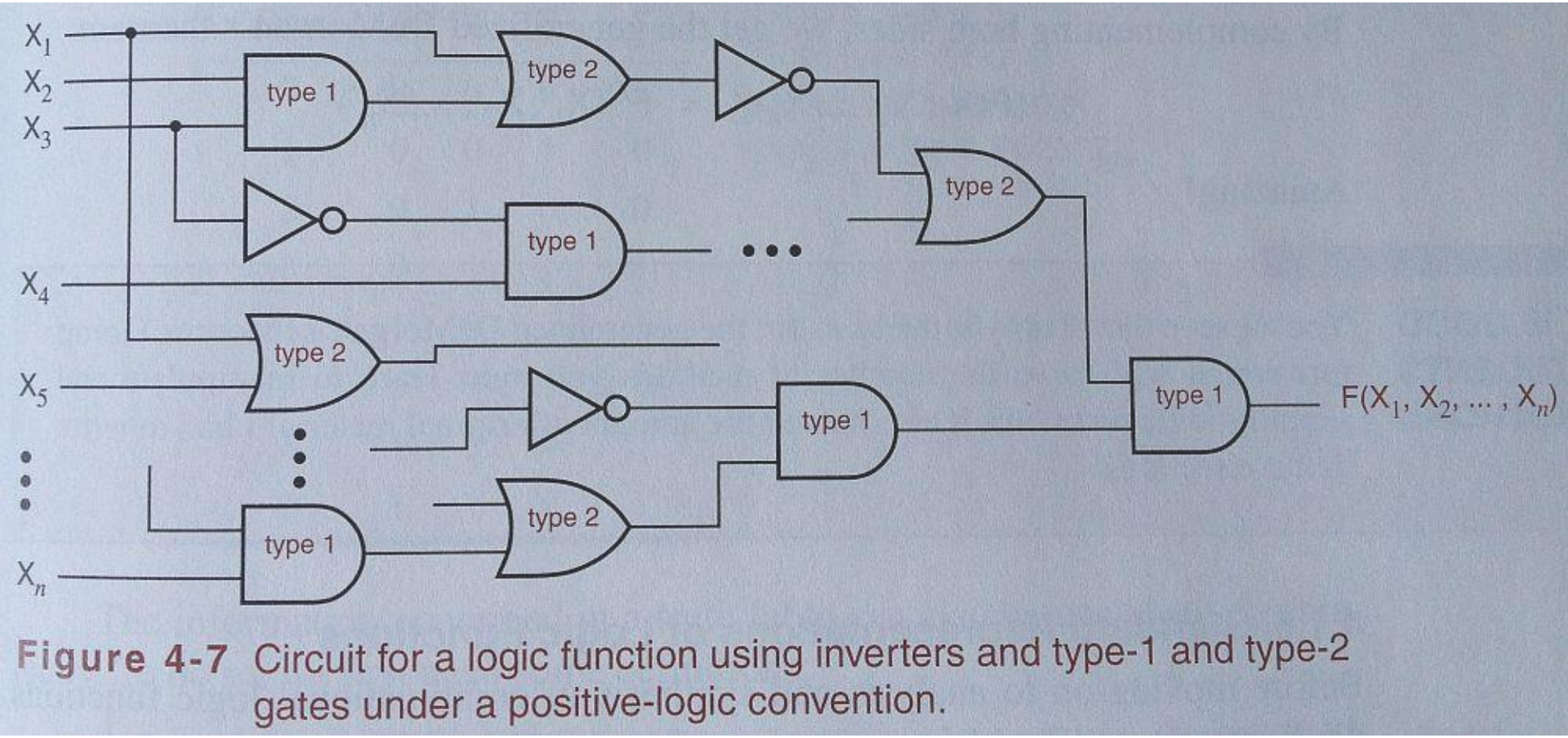


Figure 4-7 Circuit for a logic function using inverters and type-1 and type-2 gates under a positive-logic convention.

Duality

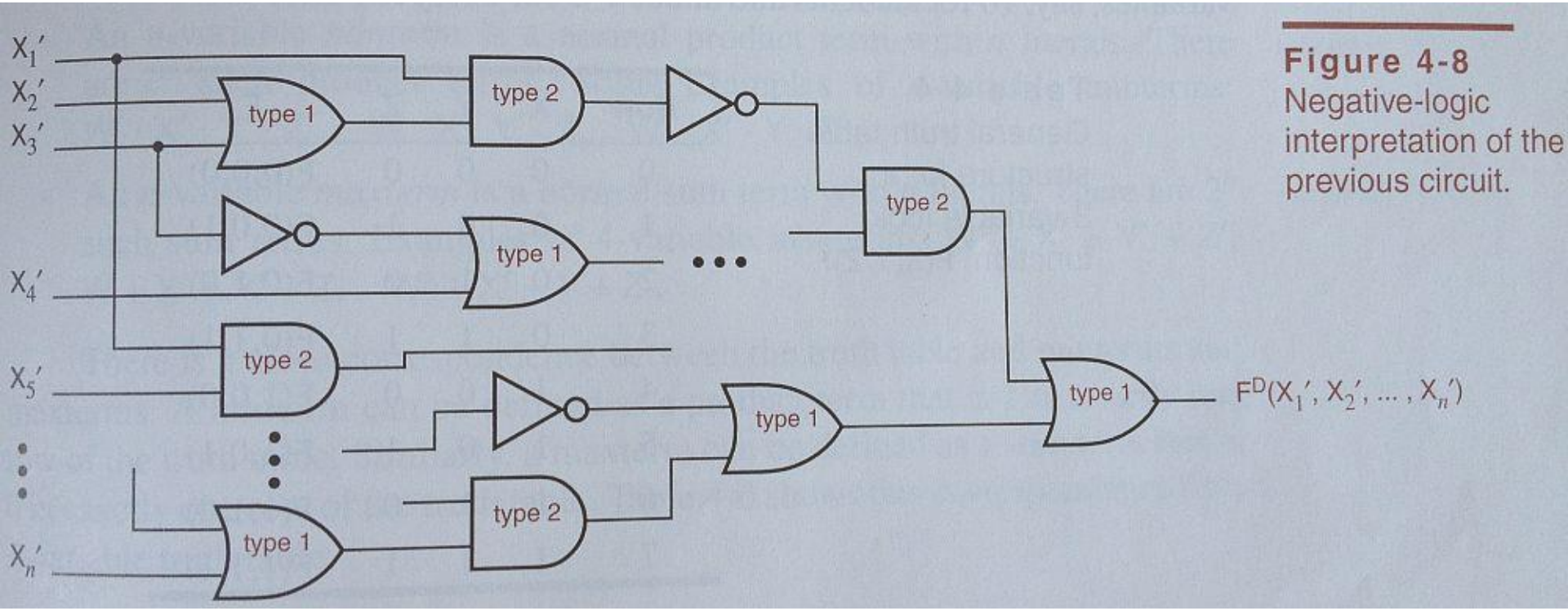


Figure 4-8
Negative-logic interpretation of the previous circuit.

Representation of Logic Functions

- Truth tables

Row	X	Y	Z	F
0	0	0	0	F(0,0,0)
1	0	0	1	F(0,0,1)
2	0	1	0	F(0,1,0)
3	0	1	1	F(0,1,1)
4	1	0	0	F(1,0,0)
5	1	0	1	F(1,0,1)
6	1	1	0	F(1,1,0)
7	1	1	1	F(1,1,1)

In general: F (X, Y, Z)

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Particular logic function

Naming

- *A literal* – a variable or its complement; X, X', Y, Y'
- *A product term*: single literal or logical product:
  - X, W· X· Y', W'· Y'
- *A sum term*: single literal or logical product:
  - X', W+X+Y', W'+ Y'
- *Sum of products* and *product of sums*
- *Normal term* – a product or sum, where each variable appears only once
  - Ex: W· X· Y', W+X'+Y'
- *N-variable minterm* – a product with n literals:
  - Ex: 4-variables minterm: W· X· Y'· Z'
- *N-variable maxterm* – a sum with n literals:
  - Ex: 4-variables maxterm: W+X+Y+Z'

Canonical sum and product

- *Canonical sum* – a sum of the minterms corresponding to the table rows for which the output is 1
  - $F = \sum_{X,Y,Z} (0,3,4,6,7)$ – minterm list
  - $F = X'Y'Z' + X'YZ + XY'Z' + XYZ' + XYZ$
  - 0 3 4 6 7
- *Canonical product* – a product of the maxterms corresponding to the table rows for which the output is 0
  - $F = \prod_{X,Y,Z} (1,2,5)$ – maxterm list
  - $(X+Y+Z')\,(X+Y'+Z)\,(X'+Y+Z')$
  - 1 2 5

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Combinational Circuit Analysis + Synthesis

- Analysis
  - Determine the behavior for different inputs
  - Find the logic equation and compute a truth table

- Synthesis
  - Formal description of its function
  - Suggest different circuit structures
  - Transform into a standard form available
    - Sum of products: used in PLA
    - Truth table -> lookup memory in FPGA

Example – Analyzing this circuit

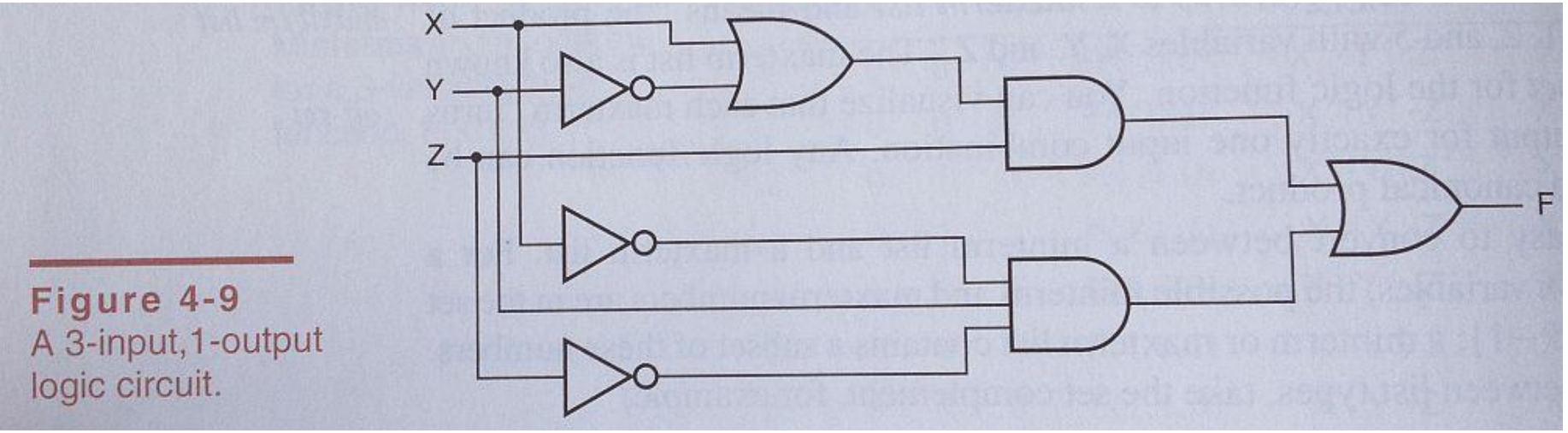


Figure 4-9
A 3-input,1-output logic circuit.

- Start to build the truth table
- Z will take a series of values: 0, 1, 0, 1, 0, 1, 0, 1
- Y will take a series of values: 0, 0, 1, 1, 0, 0, 1, 1
- X will take a series of values: 0, 0, 0, 0, 1, 1, 1, 1
- **EXERCISE: Determine the output for this circuit!**

Solution

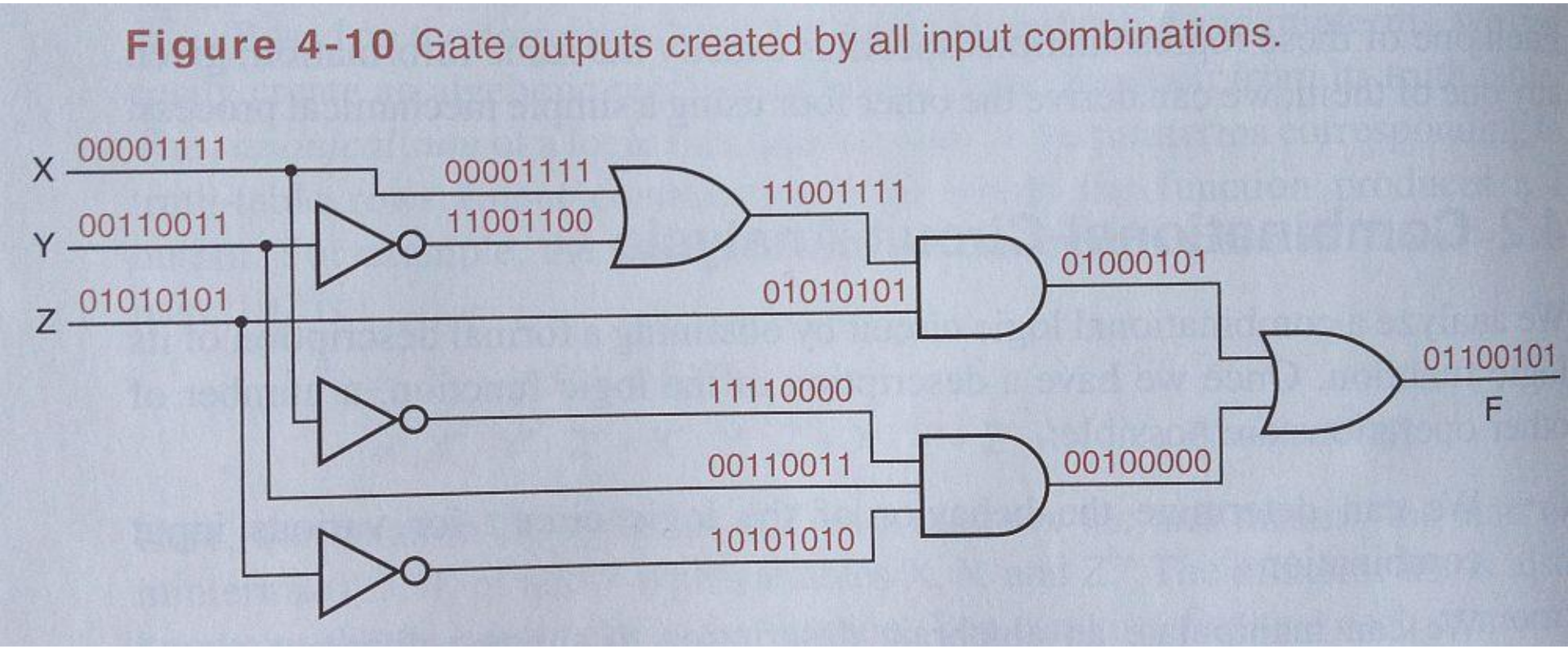


Figure 4-10 Gate outputs created by all input combinations.
X
Y
Z
00001111
00110011
01010101
00001111
11001100
11001111
01010101
01000101
11110000
00110011
10101010
00100000
01100101
F

Secondary approach

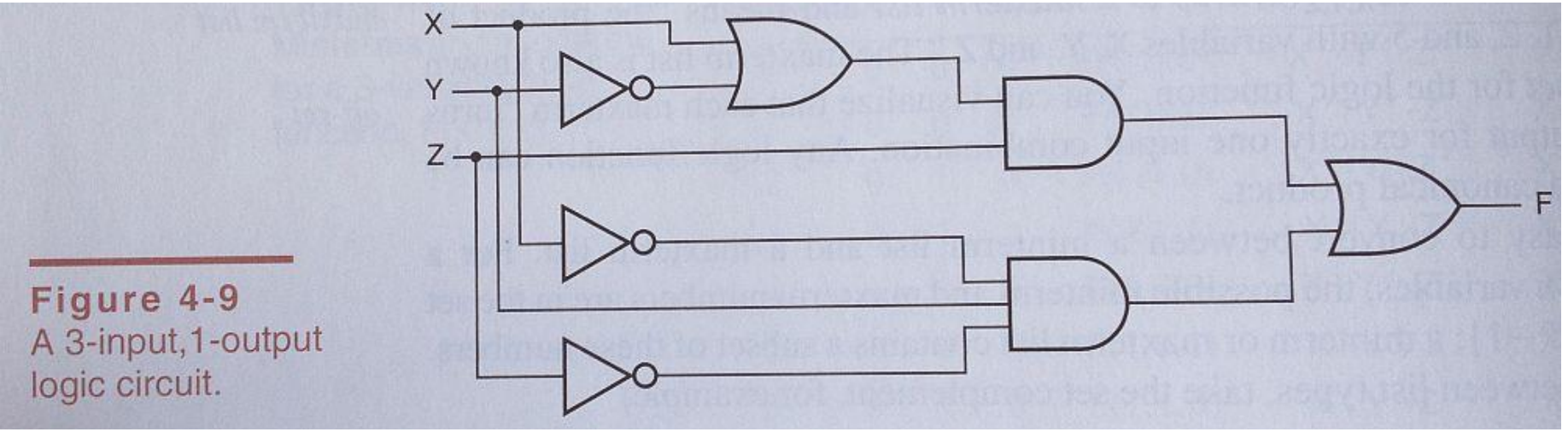


Figure 4-9
A 3-input,1-output logic circuit.

- Use algebraic rules
  - Build up a paranthesized logic expression from the structure of the circuit
  - Start at the inputs, go towards outputs
  - Simplifying can be done (or not)
- **EXERCISE: Write the output function!**

Solution

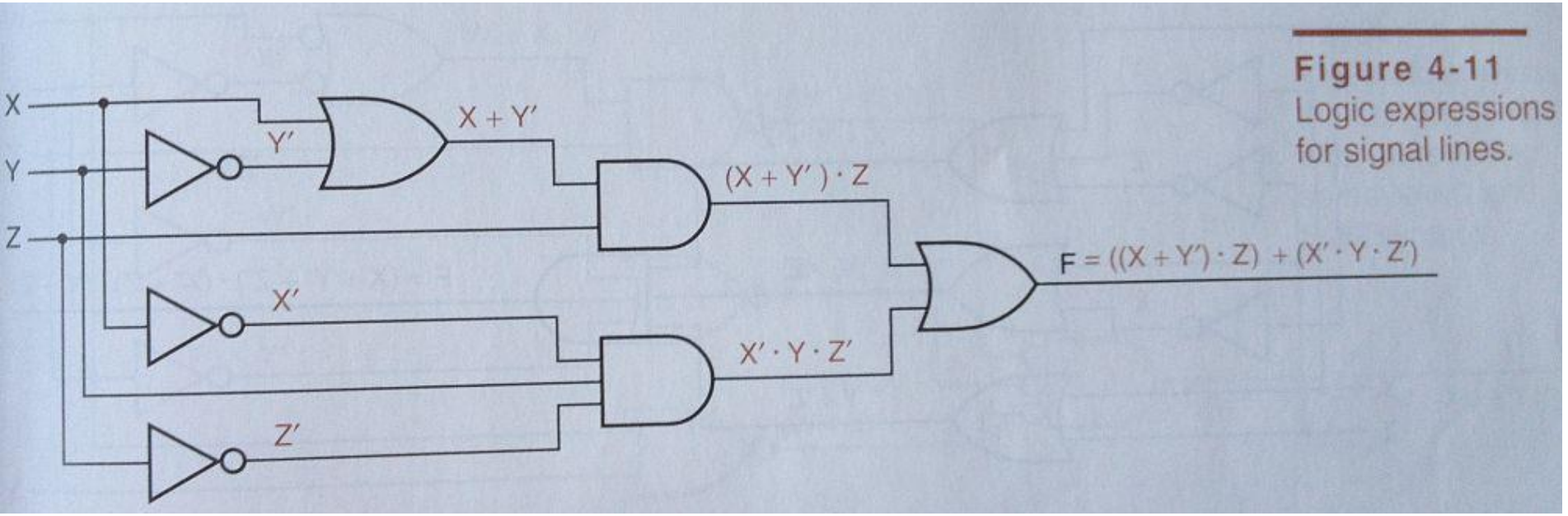


Figure 4-11
Logic expressions for signal lines.

- Simplifying gives:
  - F = X · Z + Y' · Z + X' · Y · Z'
- **EXERCISE: draw the new circuit!**

Solution

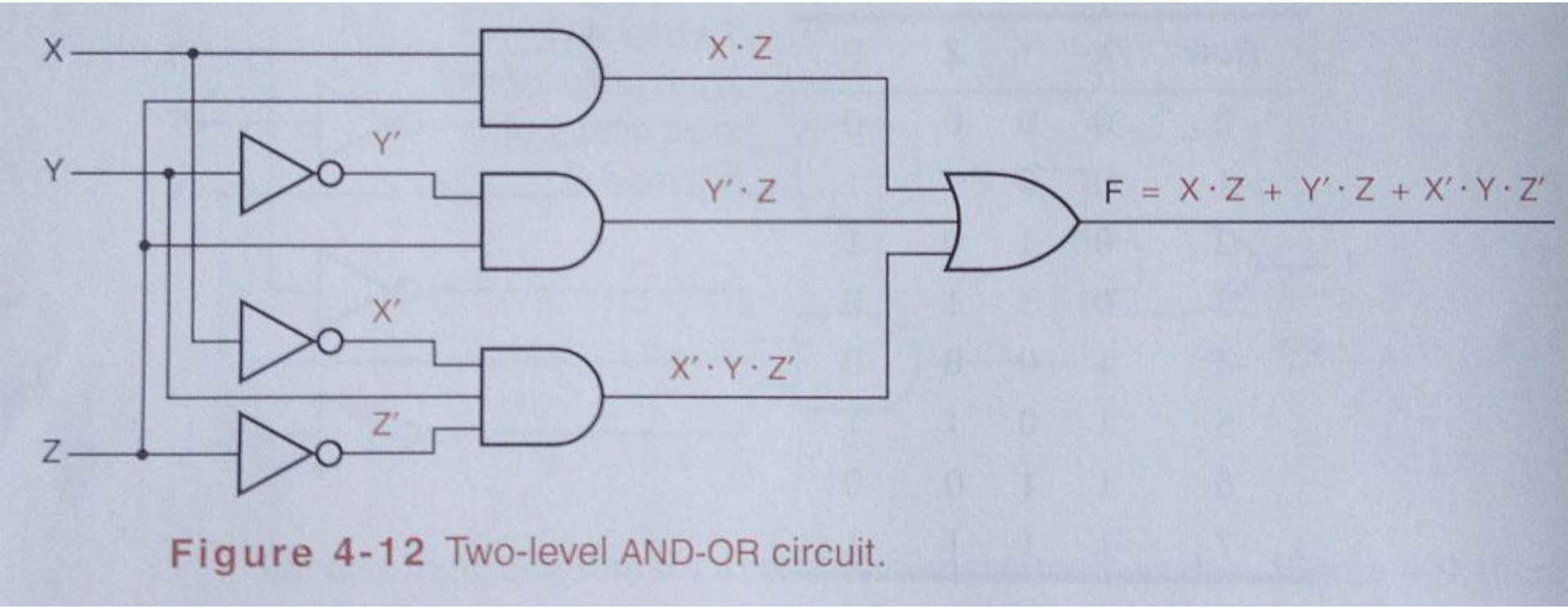


Figure 4-12 Two-level AND-OR circuit.

Another circuit

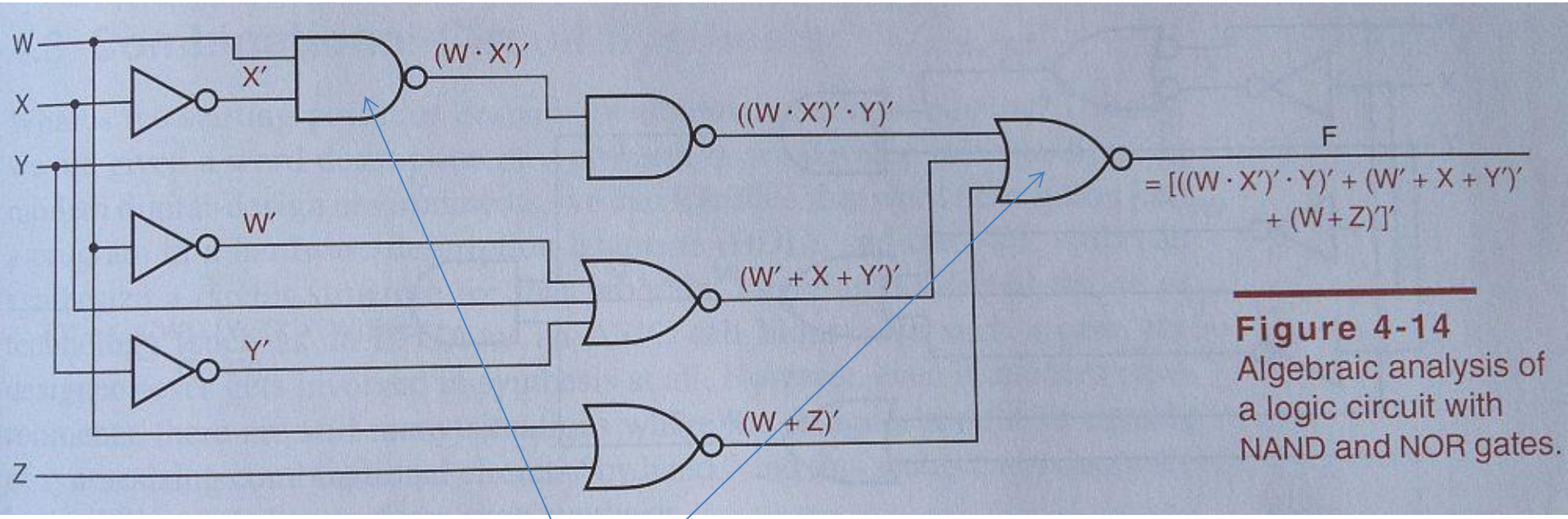


Figure 4-14
Algebraic analysis of a logic circuit with NAND and NOR gates.

- **EXERCISE: Use DeMorgan's theorem to replace some of the blocks!**

Solution

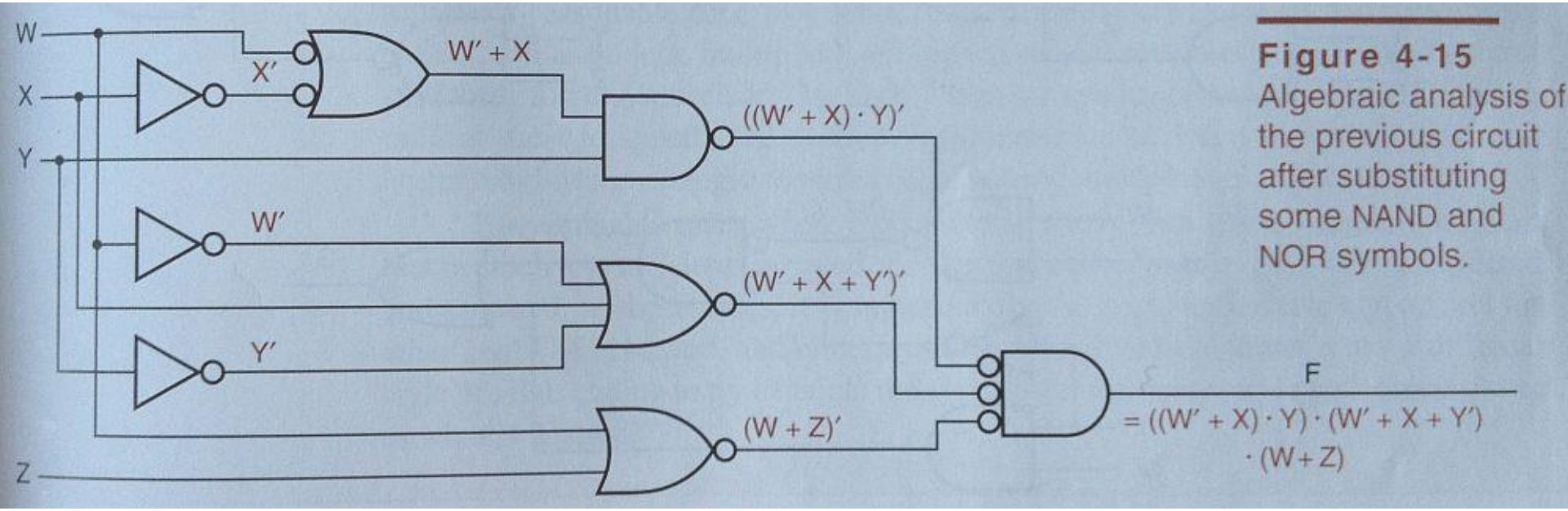


Figure 4-15
Algebraic analysis of the previous circuit after substituting some NAND and NOR symbols.

- Could this circuit be simplified even more?

Yes, remove double negations

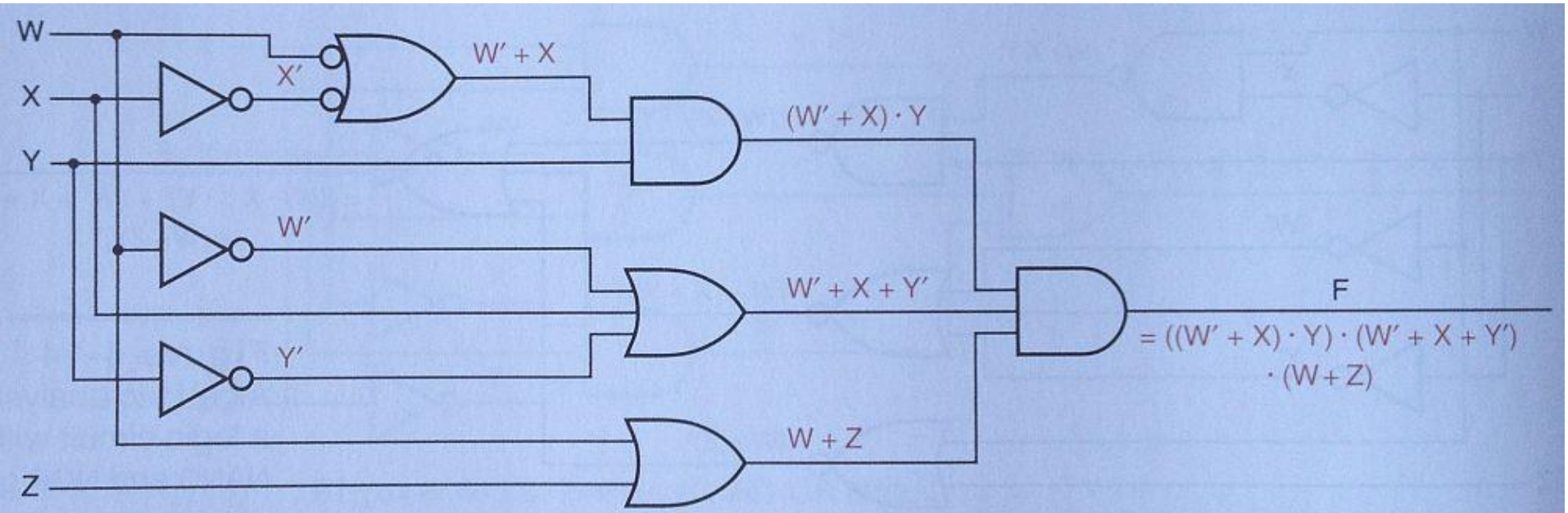
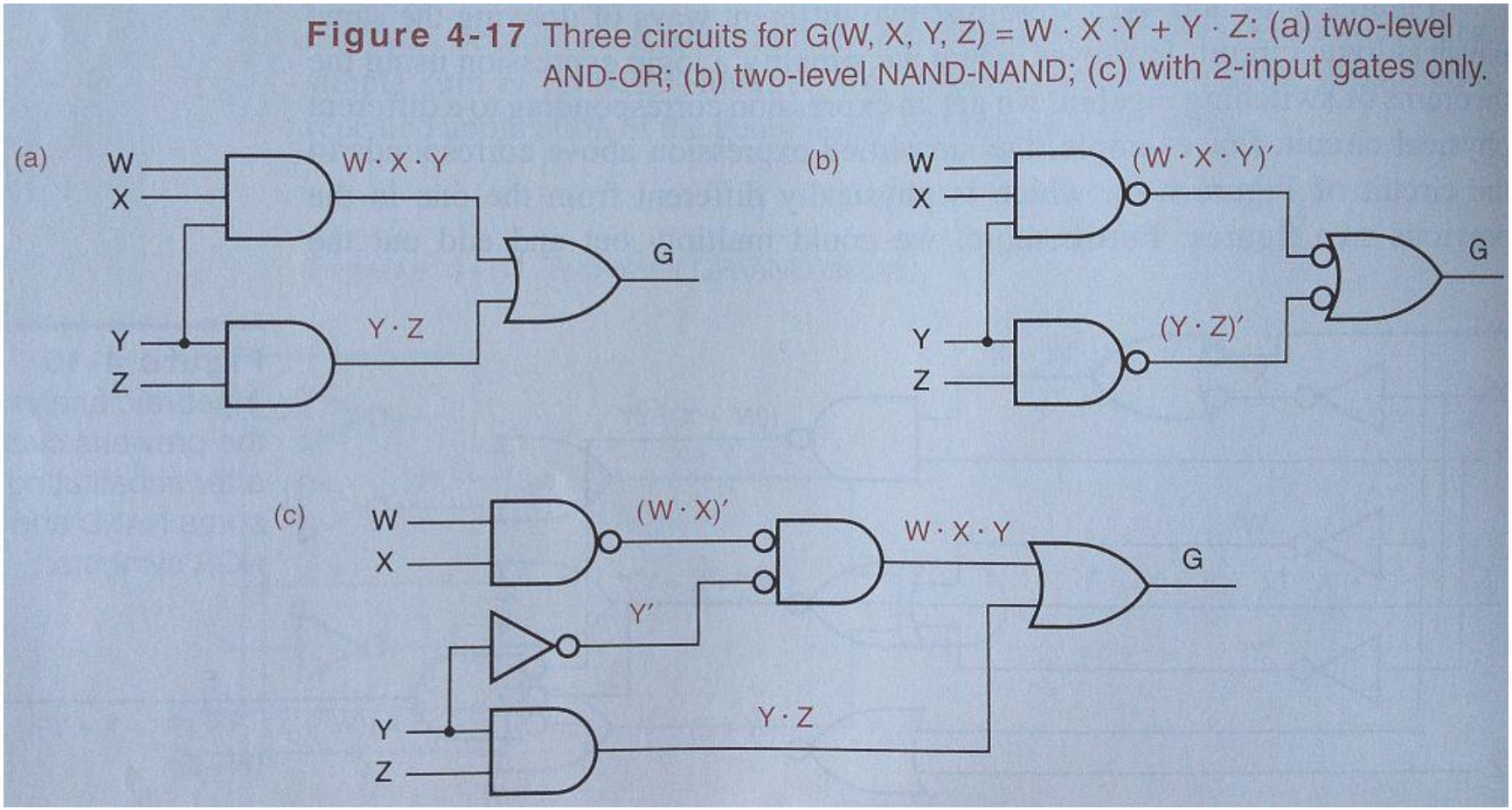


Figure 4-16 A different circuit for same logic function.

Different ways to describe the same circuit

- $G(W, X, Y, Z) = W \cdot X \cdot Y + Y \cdot Z$



Overview of assignment 1

- Hints:
  - A + AB = A
  - A + A'B = A+B
  - (A+B) (A+C) = A + BC

- Other questions?