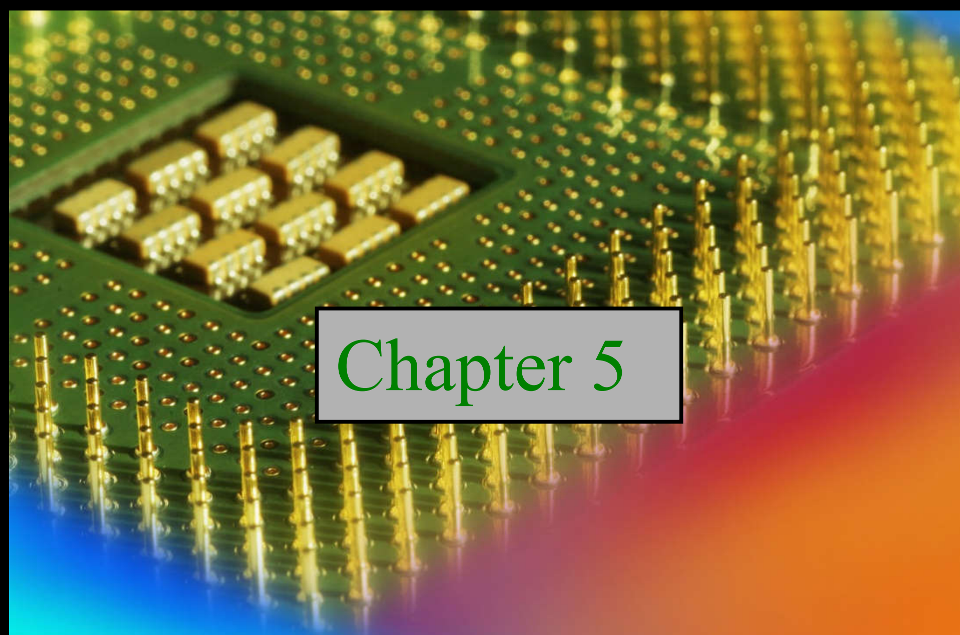
Digital Fundamentals

Tenth Edition

Floyd

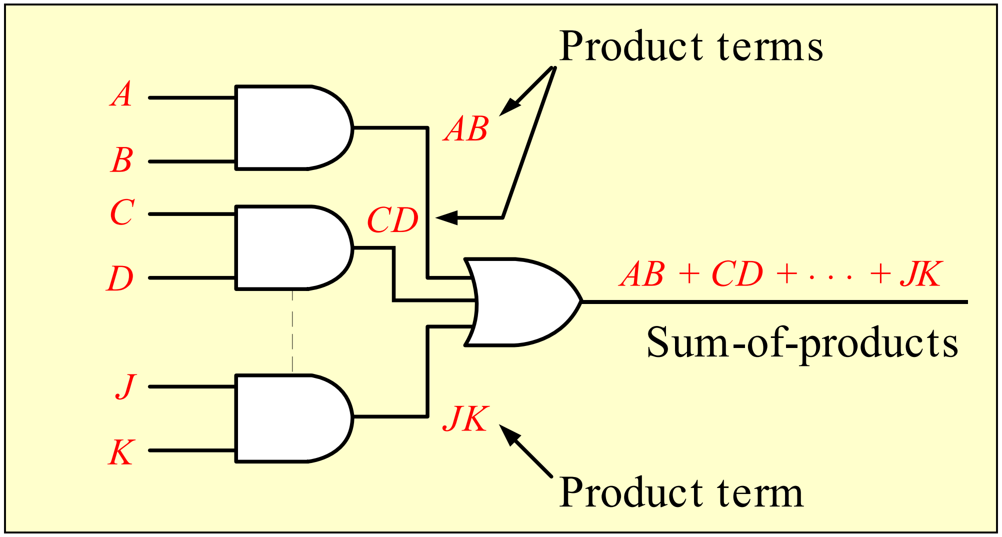


Chapter 5

Summary

Combinational Logic Circuits

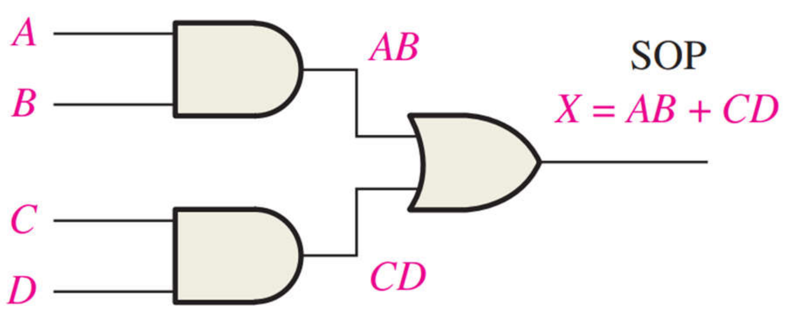
In Sum-of-Products (SOP) form, basic combinational circuits can be directly implemented with **AND-OR** combinations if the necessary complement terms are available.



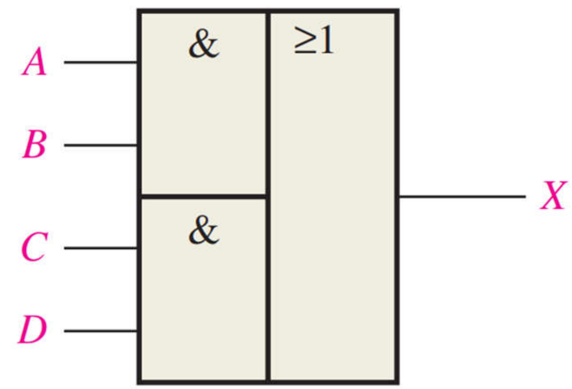
Summary

Combinational Logic Circuits

An example of an SOP implementation is shown. The SOP expression is an AND-OR combination of the input variables and the appropriate complements.



(a) Logic diagram (ANSI standard distinctive shape symbols)



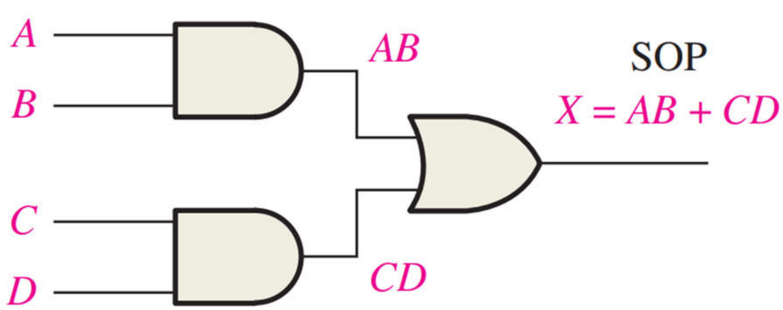
(b) ANSI standard rectangular outline symbol

For a 4-input AND-OR logic circuit, the output *X* is HIGH (1) if both input *A* and input *B* are HIGH (1) or both input *C* and input *D* are HIGH (1).

Summary

Combinational Logic Circuits

An example of an SOP implementation is shown. The SOP expression is an AND-OR combination of the input variables and the appropriate complements.



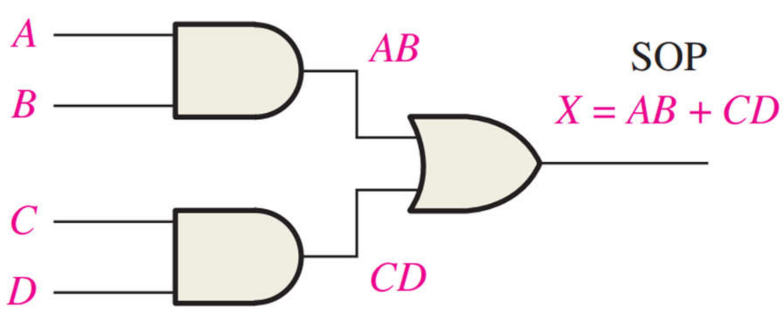
(a) Logic diagram (ANSI standard distinctive shape symbols)

Task 1:
Generate Truth Table for this logic circuit.

Summary

Combinational Logic Circuits

An example of an SOP implementation is shown. The SOP expression is an AND-OR combination of the input variables and the appropriate complements.



(a) Logic diagram (ANSI standard distinctive shape symbols)

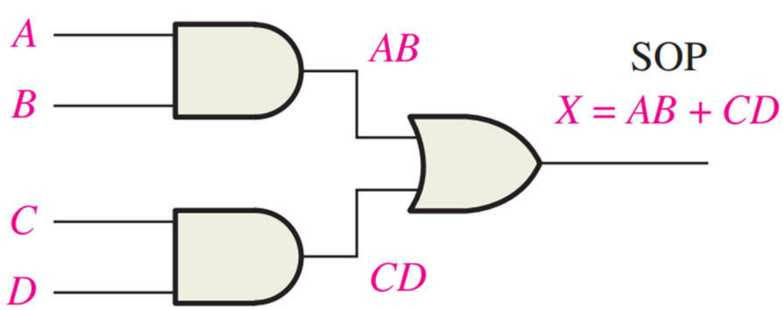
Task 2:

Use Truth Table to write logic expression in standard SOP form.

Summary

Combinational Logic Circuits

An example of an SOP implementation is shown. The SOP expression is an AND-OR combination of the input variables and the appropriate complements.



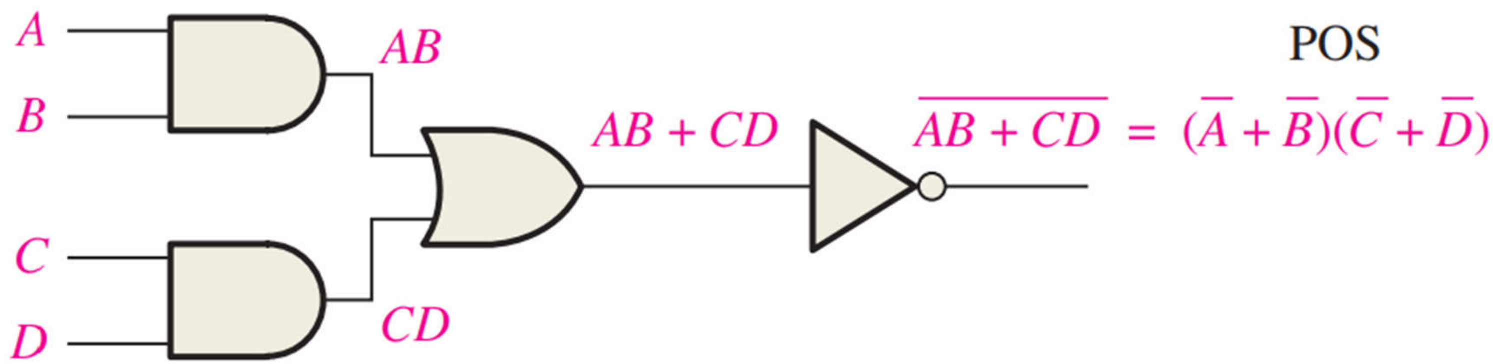
(a) Logic diagram (ANSI standard distinctive shape symbols)

Task 3:
Implement K-map to simplify this standard SOP form.

Summary

Combinational Logic Circuits

When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert** circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.

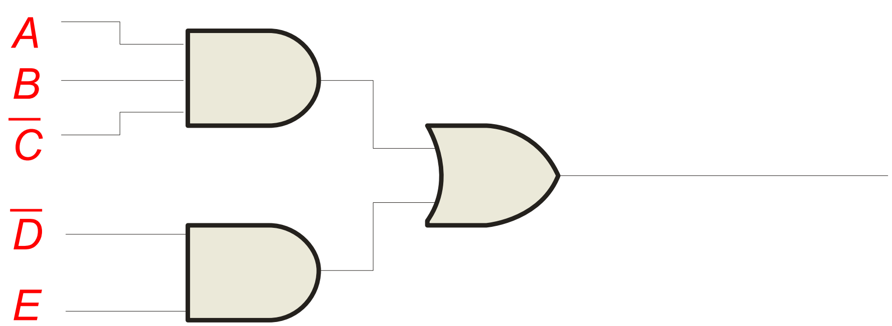


For a 4-input AND-OR-Invert logic circuit, the output *X* is LOW (0) if both input *A* and input *B* are HIGH (1) or both input *C* and input *D* are HIGH (1).

Summary

Combinational Logic Circuits

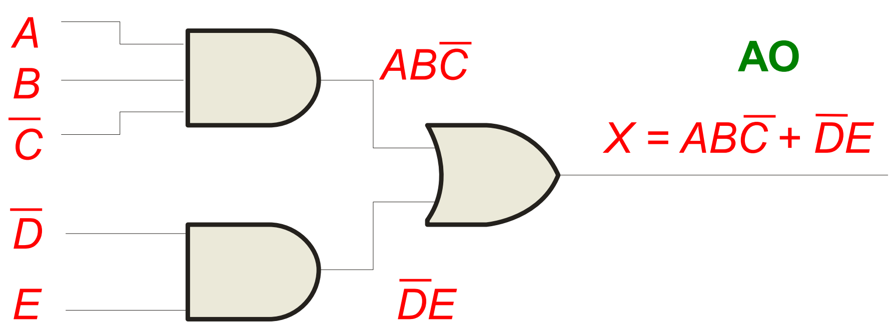
When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert** circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.



Summary

Combinational Logic Circuits

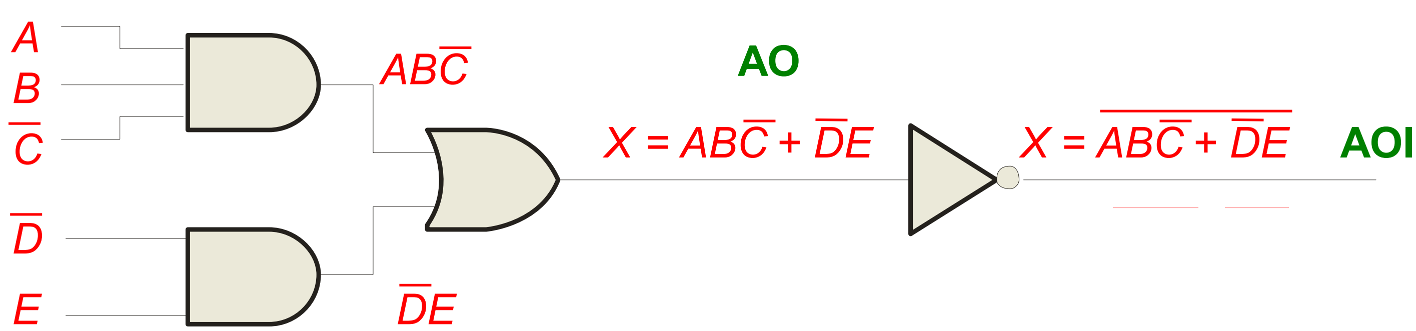
When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert** circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.



Summary

Combinational Logic Circuits

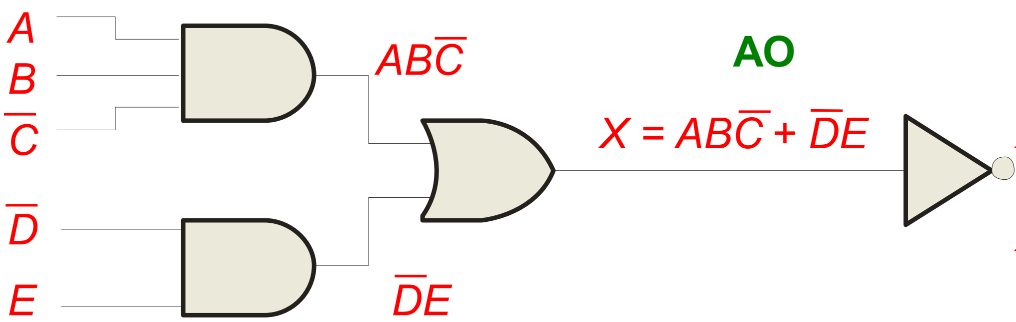
When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert** circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.



Summary

Combinational Logic Circuits

When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert** circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.



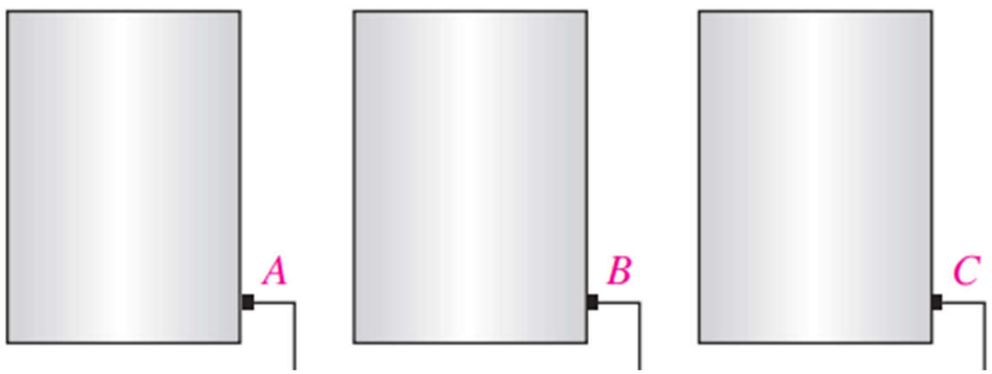
$X = \overline{AB\overline{C} + \overline{D}E}$ **AOI**

$X = (\overline{AB\overline{C}})(\overline{\overline{D}E})$ DeMorgan

$X = (\overline{A} + \overline{B} + C)(D + \overline{E})$ POS

In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

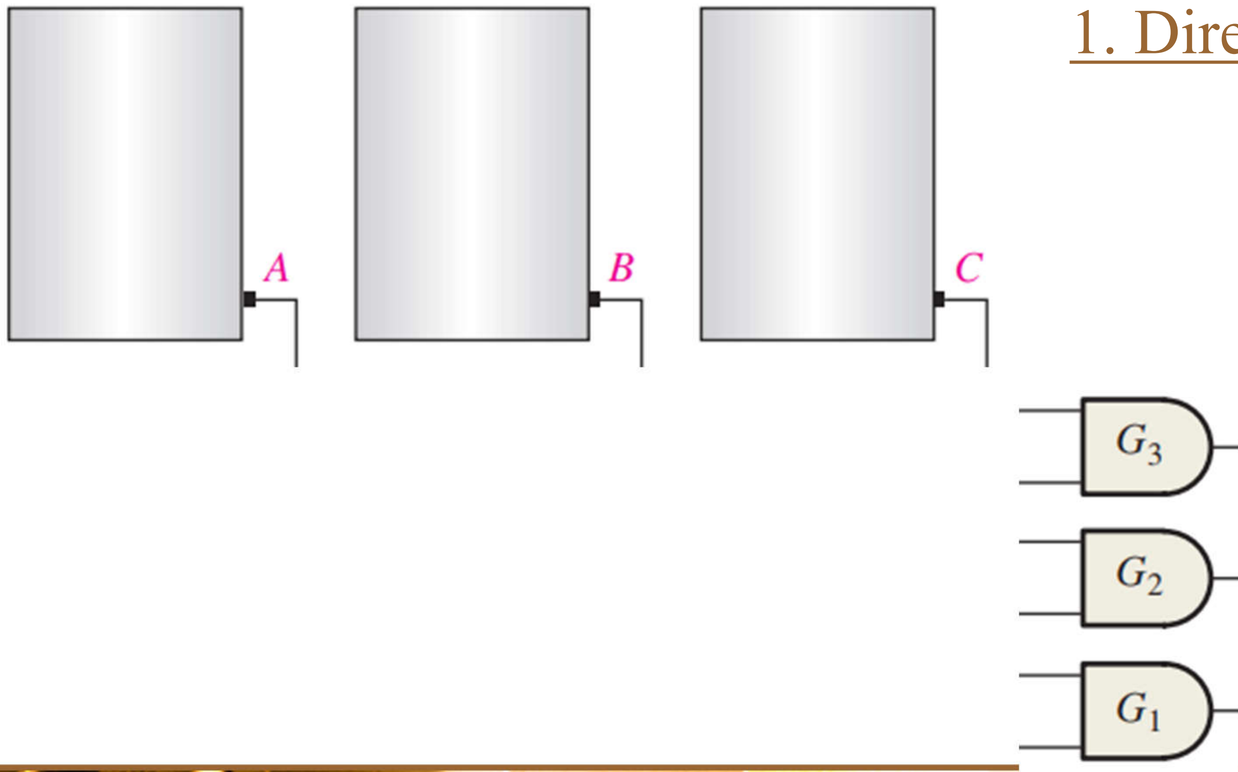


1. Direct method

In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

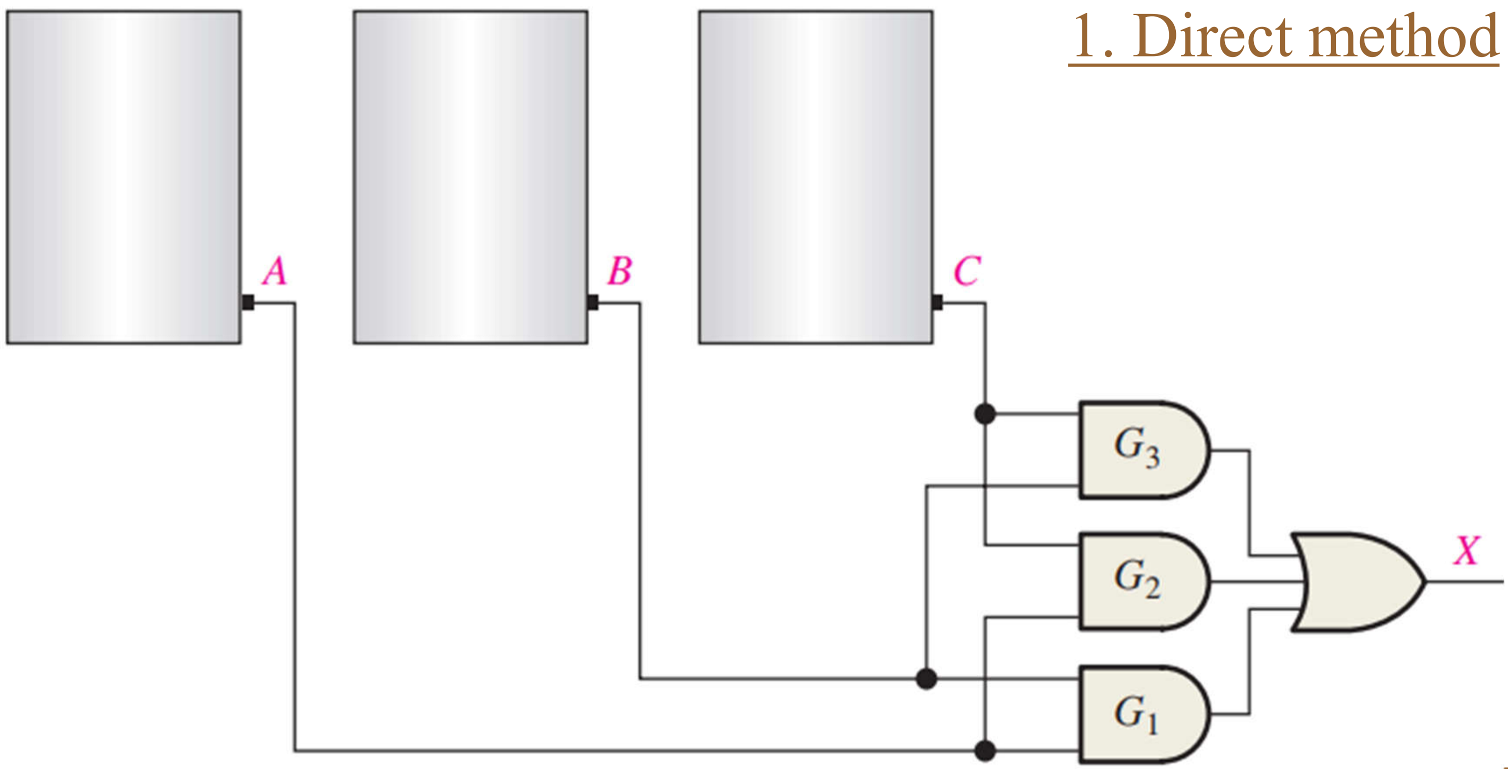
1. Direct method



In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

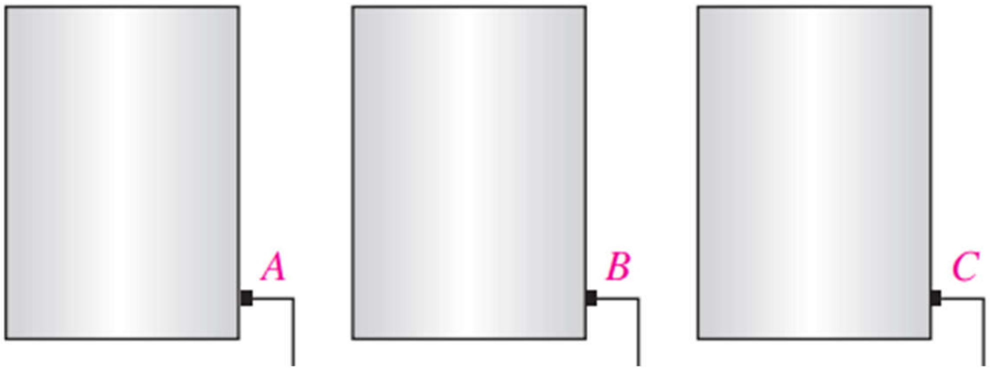
1. Direct method



Floyd, Digital Fundamentals, 10th ed

In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

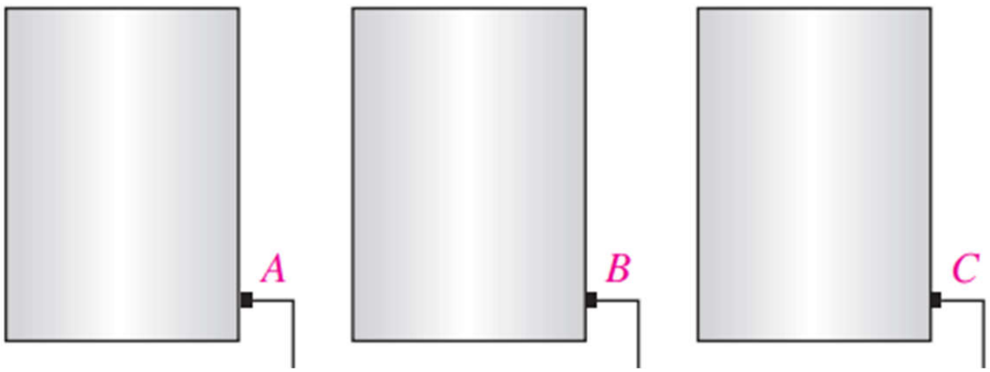


2. Truth table to logic expression

A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.



3. Truth table to K-map

		1	
	1	1	1

A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Summary

Exclusive-OR Logic

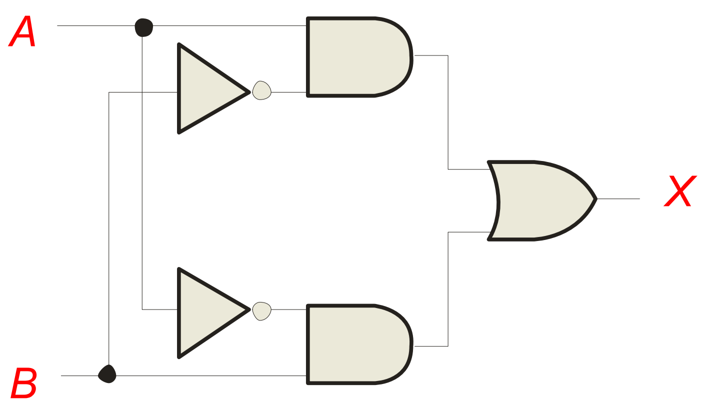
The truth table for an exclusive-OR gate is

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

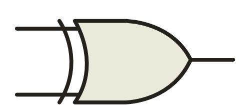
Notice that the output is HIGH whenever A and B disagree.

The Boolean expression is $X = \overline{A}B + A\overline{B}$

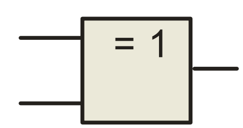
The circuit can be drawn as



Symbols:



Distinctive shape



Rectangular outline

Summary

Exclusive-NOR Logic

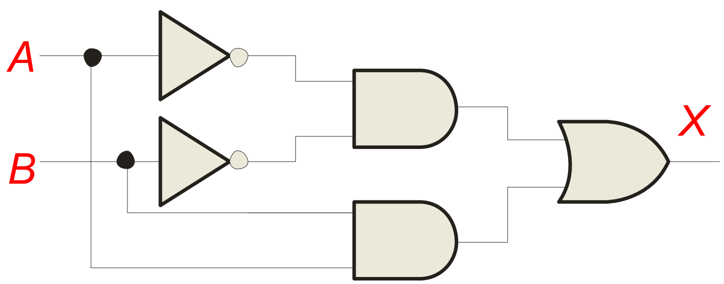
The truth table for an exclusive-NOR gate is

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

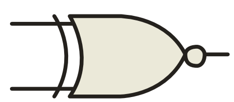
Notice that the output is HIGH whenever *A* and *B* agree.

The Boolean expression is $X = \overline{A}\overline{B} + AB$

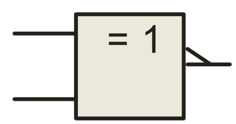
The circuit can be drawn as



Symbols:



Distinctive shape



Rectangular outline

Summary

Exclusive-NOR Logic

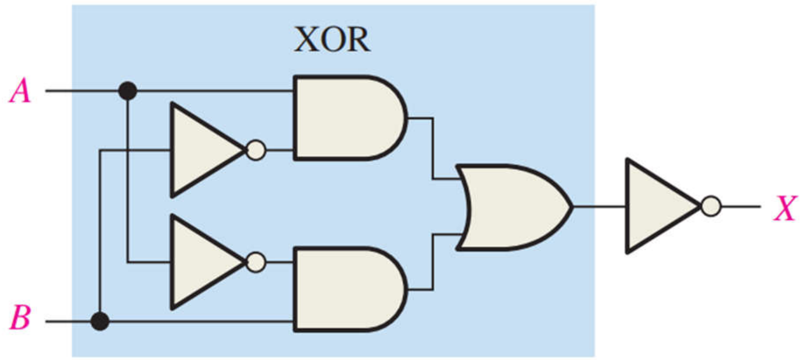
The truth table for an exclusive-NOR gate is

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Notice that the output is HIGH whenever A and B agree.

The Boolean expression is $X = \overline{A}\,\overline{B} + AB$

The circuit can be drawn as



(a) $X = \overline{A\overline{B} + \overline{A}B}$

Symbols:

Distinctive shape Rectangular outline

Summary

Exclusive-OR Logic

Exclusive-OR gates can be used in Parity encoding.

Try to implement an even-parity code generator for an original 2-bit code

Truth table for an exclusive-OR.

A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

Summary

Exclusive-OR Logic

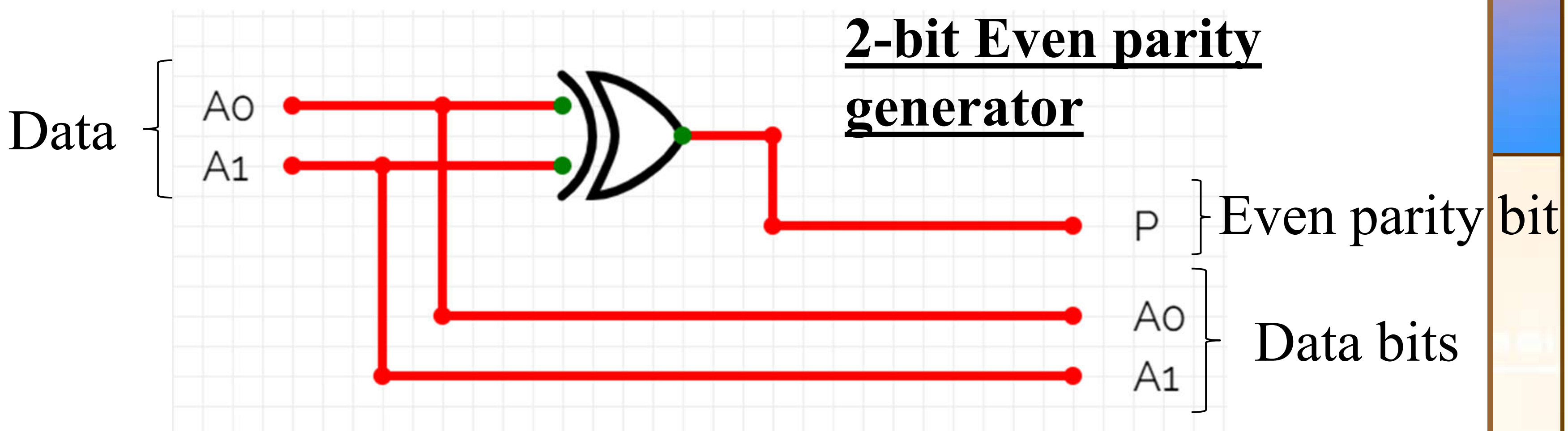
Exclusive-OR gates can be used in Parity encoding.

Try to implement an even-parity code generator for an original 2-bit code

Truth table for an exclusive-OR.

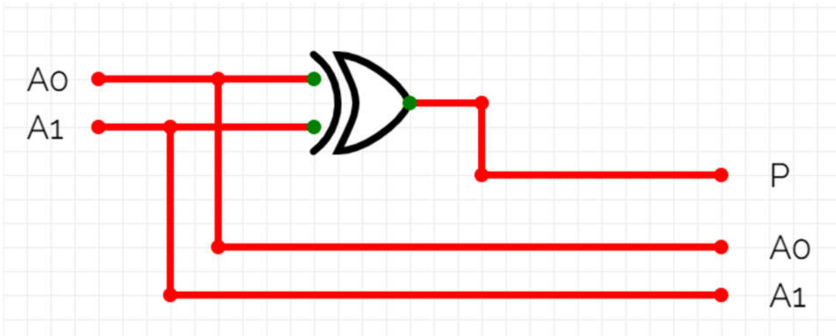
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

2-bit Even parity generator



Summary

Exclusive-OR Logic



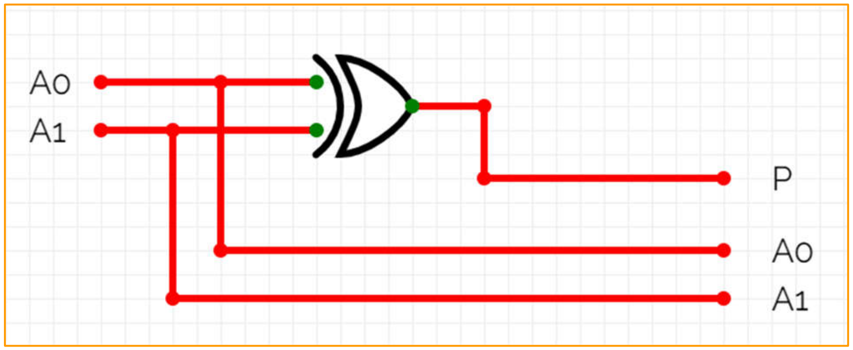
What about even parity checker circuit on the other end?

Truth table for an exclusive-OR.

A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

Summary

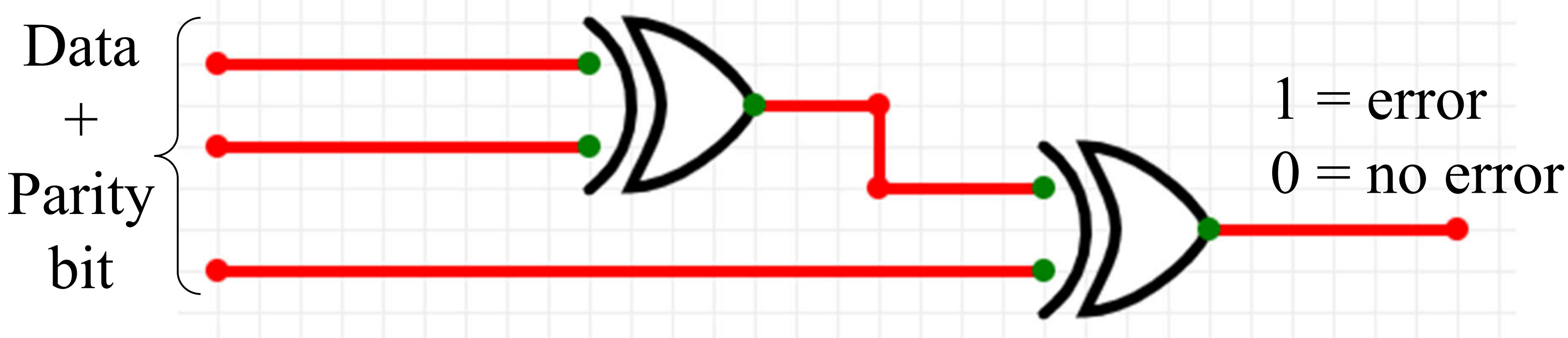
Exclusive-OR Logic



Truth table for an exclusive-OR.

A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

2-bit Even parity checker



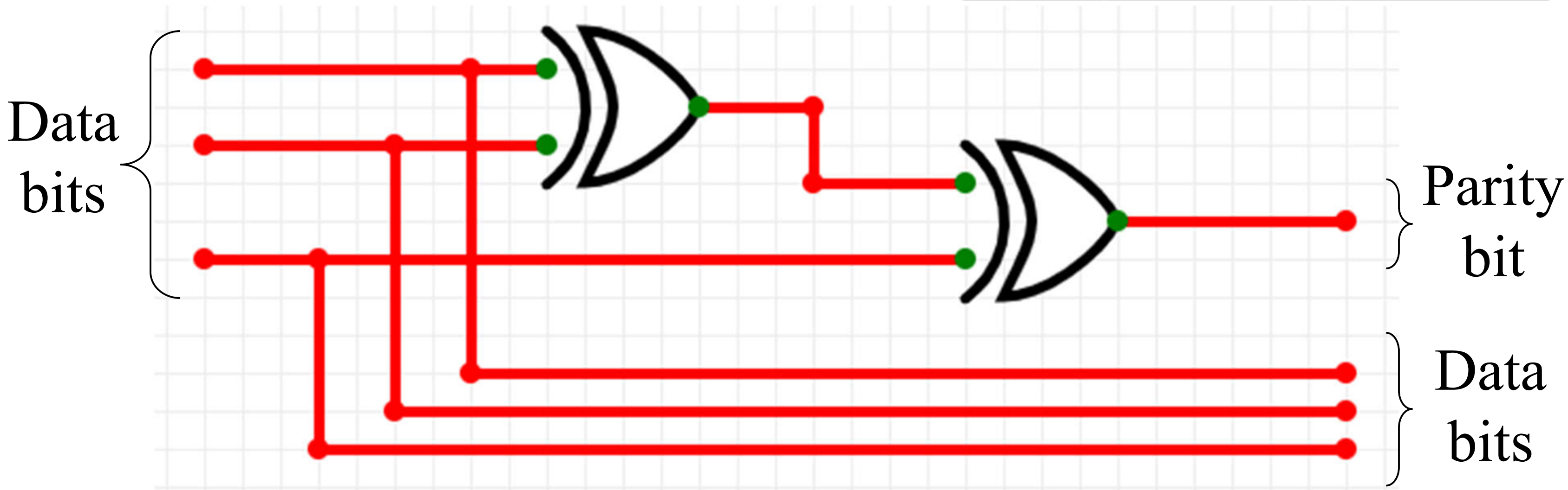
Summary

Exclusive-OR Logic

Now try to implement an even-parity code generator for an original 3-bit code.

Truth table for an exclusive-OR.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



EXAMPLE 5–3

Use exclusive-OR gates to implement an even-parity code generator for an original 4-bit code.

EXAMPLE 5–3

Use exclusive-OR gates to implement an even-parity code generator for an original 4-bit code.

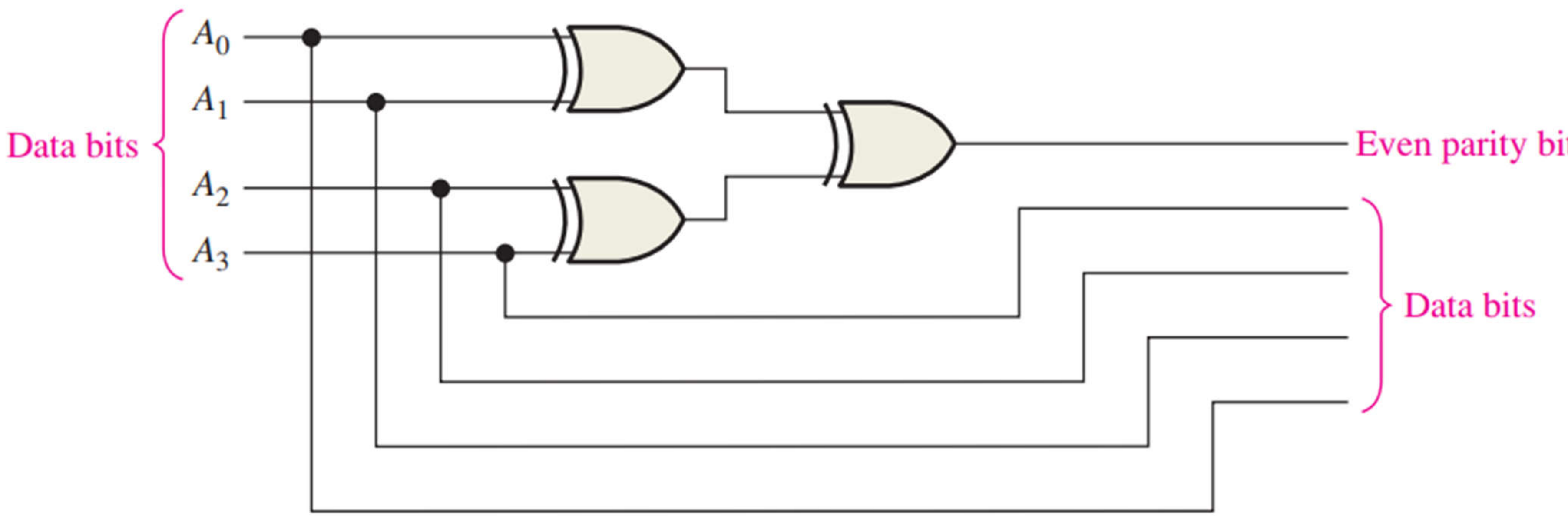


FIGURE 5–7 Even-parity generator.

EXAMPLE 5–3

Use exclusive-OR gates to implement an even-parity code generator for an original 4-bit code.

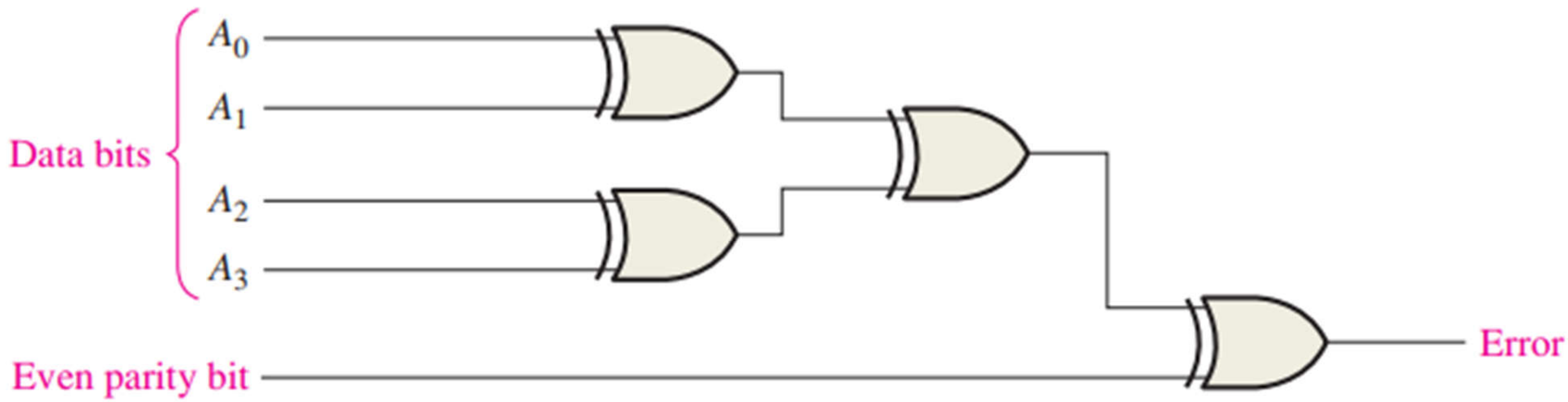


FIGURE 5–8 Even-parity checker.