CHAPTER #3

LECTURE 7

Quine-McCluskey Method

- Programmed based approach
- Two step method
- Find Prime Implicants through exhaustive search
- Selecting minimal set of essential prime implicants
- Consider the function defined in Canonical Sum form as Sigma ABCD (1, 3, 6, 7, 8, 9, 11, 12,13, 14, 15)
- Compare each 1 in the first row with the 1 in the lower row

Quine-McCluskey Method (table1)

Minterm	A	B	C	D
1	0	0	0	1
3	0	0	1	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Quine-McCluskey Method

- Now arrange these entries on the basis of number of 1’s
- First group contains those entries where number of 1’s are 0
- Second group contains number of 1’s 1
- Third group contains number of 1’s 2 and so on

Quine-McCluskey Method (table2)

Minterm	A	B	C	D	used
1	0	0	0	1	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
12	1	1	0	0	✓
7	0	1	1	1	✓
11	1	0	1	1	✓
13	1	1	0	1	✓
14	1	1	1	0	✓
15	1	1	1	1	✓

Quine-McCluskey Method

- Now compare each entry of group 1 with group 2 and each entry of group 2 with group 3 and each entry of group 3 with group 4.
- A match is found if difference of number of bits is 1 otherwise its mismatch.
- If more than one bit of each binary entry is different then we will skip them.
- We will keep on doing this until there is no entry that match with other entry.

Quine-McCluskey Method (table3)

	A	B	C	D	used
1,3	0	0	-	1	✓
1,9	-	0	0	1	✓
8,9	1	0	0	-	✓
8,12	1	-	0	0	✓
3,7	0	-	1	1	✓
3,11	-	0	1	1	✓
6,7	0	1	1	-	✓
6,14	-	1	1	0	✓
9,11	1	0	-	1	✓
9,13	1	-	0	1	✓

Quine-McCluskey Method (table3)

	A	B	C	D	used
12,13	1	1	0	-	✓
12,14	1	1	-	0	✓
7,15	-	1	1	1	✓
11,15	1	-	1	1	✓
13,15	1	1	-	1	✓
14,15	1	1	1	-	✓

Quine-McCluskey Method (table4)

	A	B	C	D	used
1,3,9,11	-	0	-	1	
8,9,12,13	1	-	0	-	
3,7,11,15	-	-	1	1	
6,7,14,15	-	1	1	-	
9,11,13,15	1	-	-	1	
12,13,14,15	1	1	-	-	

Quine-McCluskey Method

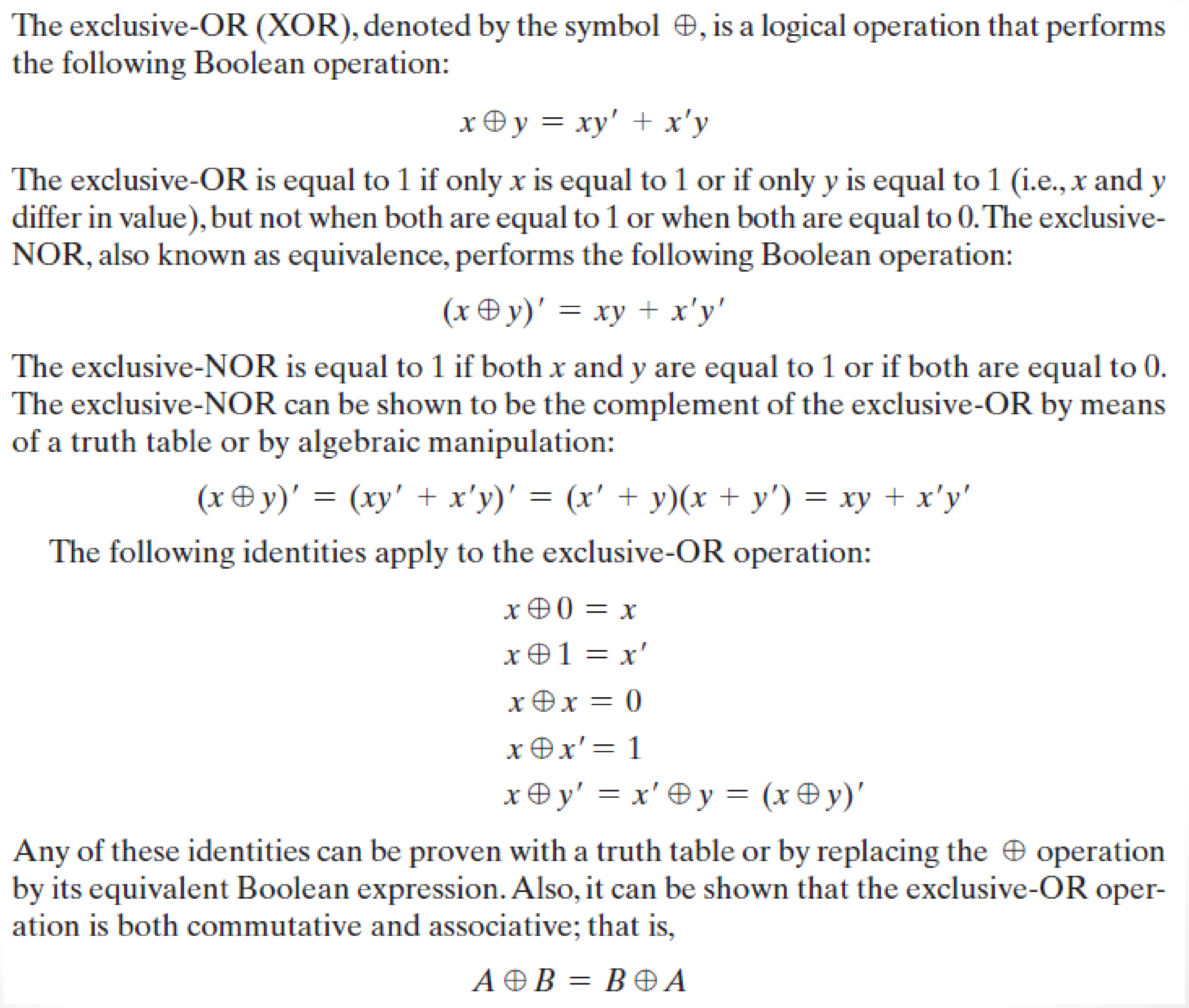
- Now these are prime implicants (having maximum number of 1’s)
- We will find essential prime implicants by putting them in table and circling those cross that have only single cross in columns.
- These are our essential prime implicants or minimized expression that can be verified from K-Map also

Quine-McCluskey Method (table5)

	1	3	6	7	8	9	11	12	13	14	15
$\overline{B}D$	$\textcircled{x}$	x				x	x				
$A\overline{C}$					$\textcircled{x}$	x		x	x		
CD		x		x			x				x
BC			$\textcircled{x}$	x						x	x
AD						x	x		x		x
AB								x	x	x	x

F=B’D+AC’+BC

EXCLUSIVE-OR Gate



The exclusive-OR (XOR), denoted by the symbol $\oplus$, is a logical operation that performs the following Boolean operation:

$$x \oplus y = xy' + x'y$$

The exclusive-OR is equal to 1 if only x is equal to 1 or if only y is equal to 1 (i.e., x and y differ in value), but not when both are equal to 1 or when both are equal to 0. The exclusive-NOR, also known as equivalence, performs the following Boolean operation:

$$(x \oplus y)' = xy + x'y'$$

The exclusive-NOR is equal to 1 if both x and y are equal to 1 or if both are equal to 0. The exclusive-NOR can be shown to be the complement of the exclusive-OR by means of a truth table or by algebraic manipulation:

$$(x \oplus y)' = (xy' + x'y)' = (x' + y)(x + y') = xy + x'y'$$

The following identities apply to the exclusive-OR operation:

$$x \oplus 0 = x$$
$$x \oplus 1 = x'$$
$$x \oplus x = 0$$
$$x \oplus x' = 1$$
$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

Any of these identities can be proven with a truth table or by replacing the $\oplus$ operation by its equivalent Boolean expression. Also, it can be shown that the exclusive-OR operation is both commutative and associative; that is,

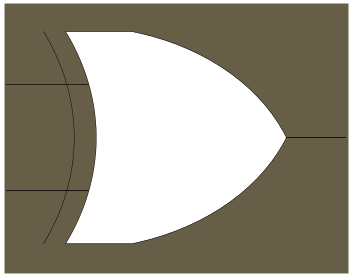
$$A \oplus B = B \oplus A$$

XOR Gate function

Its output can be found by modulo-2. Means we will take two bits like 0,1 and add them 0+1=1 and then divide by 2 and take remainder which is 1.
F=A'B+AB'

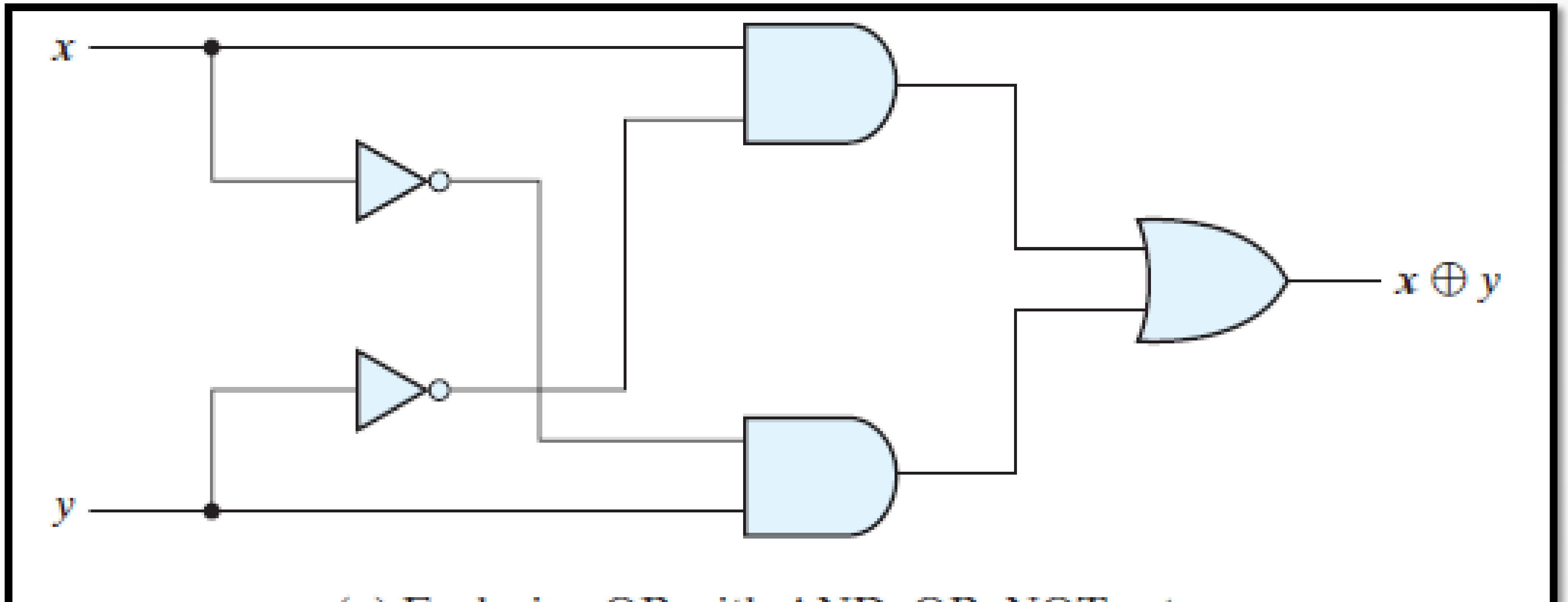
Input		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

$$F = A \oplus B$$



It is used to detect odd number of errors in data transmission

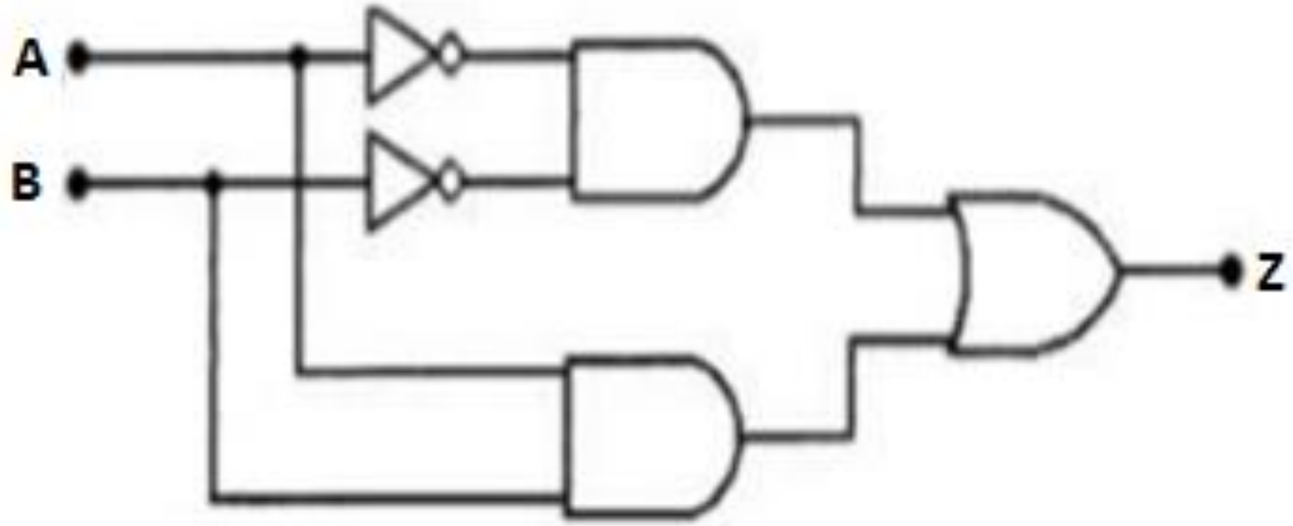
EXCLUSIVE-OR Gate



(a) Exclusive-OR with AND–OR–NOT gates

X’Y+XY’

EXCLUSIVE-NOR Gate

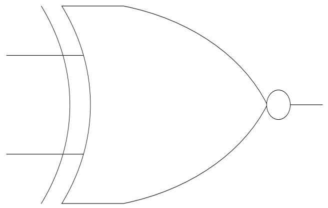


Z=A.B+A’B’

XNOR Gate function

Input		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$$F = \overline{A \oplus B}$$



It is used to detect even number of errors in data transmission

EXCLUSIVE-OR As Parity Generation and Checking

- Exclusive-OR functions are very useful in systems requiring error detection and correction codes.
- A parity bit is used for the purpose of detecting errors during the transmission of binary information.
- A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even.
- The message, including the parity bit, is transmitted and then checked at the receiving end for errors.
- An error is detected if the checked parity does not correspond with the one transmitted.
- The circuit that generates the parity bit in the transmitter is called a *parity generator.*

EXCLUSIVE-OR As Parity Generation and Checking

- The circuit that checks the parity in the receiver is called a *parity checker.*
- As an example, consider a three-bit message to be transmitted together with an even-parity bit.
- Table 3.3 shows the truth table for the parity generator.
- The three bits— *x*, *y*, and *z* —constitute the message and are the inputs to the circuit.
- The parity bit *P* is the output.
- For even parity, the bit *P* must be generated to make the total number of 1's (including *P*) even.
- From the truth table, we see that *P* constitutes an odd function because it is equal to 1 for those minterms whose numerical values have an odd number of 1's.

EXCLUSIVE-OR As Parity Generation and Checking

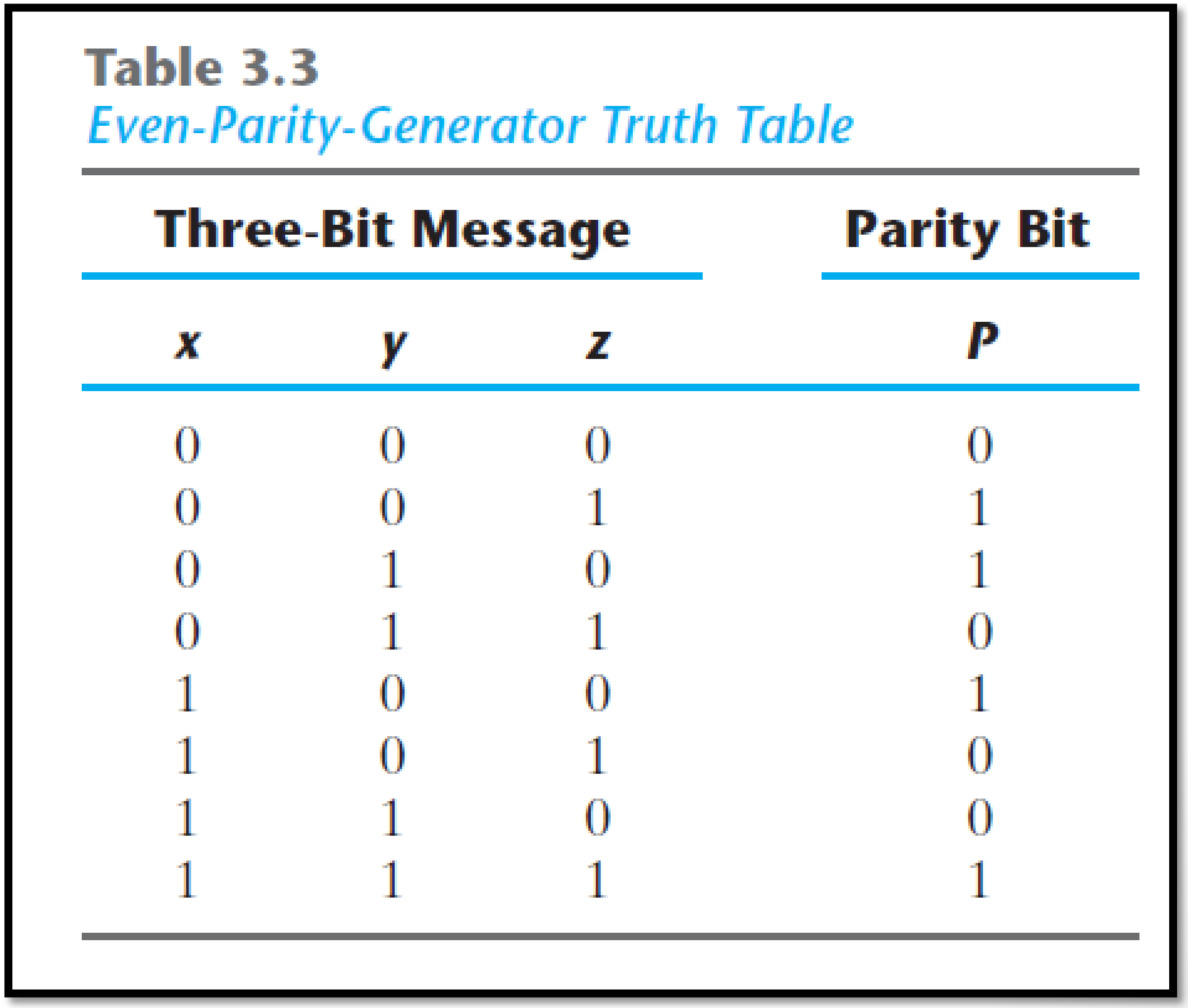


Table 3.3
Even-Parity-Generator Truth Table

Three-Bit Message			Parity Bit
x	*y*	*z*	*P*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

EXCLUSIVE-OR As Parity Generation and Checking

- Therefore, *P* can be expressed as a three-variable exclusive-OR function:
- $P = x \oplus y \oplus z$
- The logic diagram of parity generator is shown in Fig. 3.34 (a).
- The three bits in the message, together with the parity bit, are transmitted to their destination, where they are applied to a parity-checker circuit to check for possible errors in the transmission.
- Since the information was transmitted with even parity, the four bits received must have an even number of 1's.
- An error occurs during the transmission if the four bits received have an odd number of 1's, indicating that one bit has changed in value during transmission.

EXCLUSIVE-OR As Parity Generation and Checking

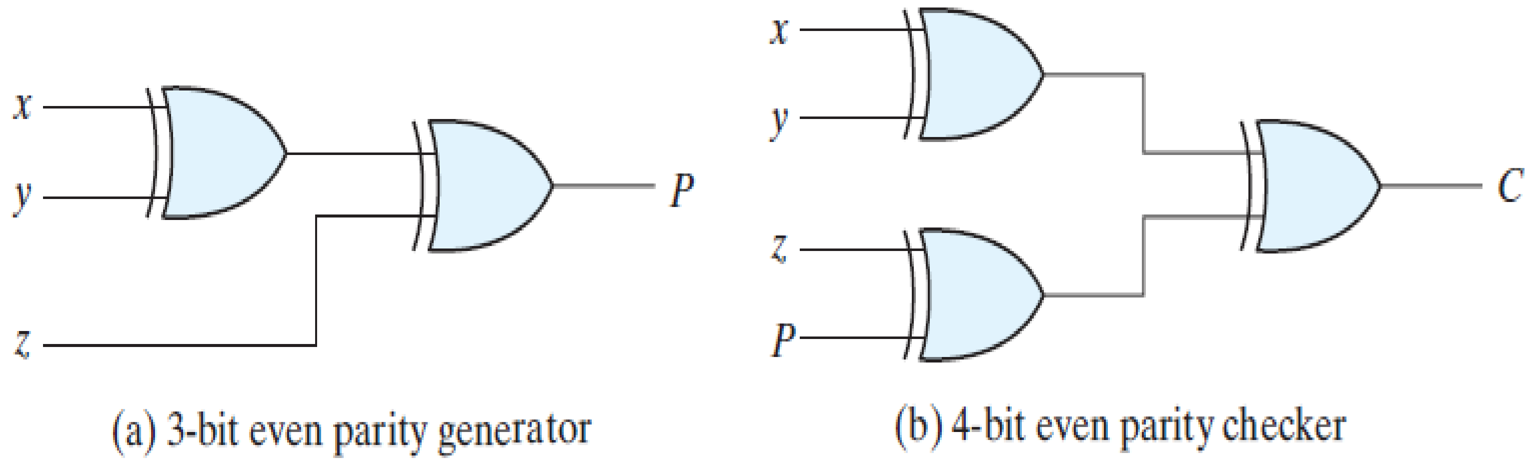

x
y
z
P
(a) 3-bit even parity generator
x
y
z
P
C
(b) 4-bit even parity checker

FIGURE 3.34
Logic diagram of a parity generator and checker

EXCLUSIVE-OR As Parity Generation and Checking

- The output of the parity checker, denoted by *C* , will be equal to 1 if an error occurs—that is, if the four bits received have an odd number of 1's.
- Table 3.4 is the truth table for the even-parity checker.
- From it, we see that the function *C* consists of the eight minterms with binary numerical values having an odd number of 1's.
- The table corresponds to the map of Fig. 3.33 (a), which represents an odd function.
- The parity checker can be implemented with exclusive-OR gates:
- $C = x \oplus y \oplus z \oplus P$

EXCLUSIVE-OR As Parity Generation and Checking

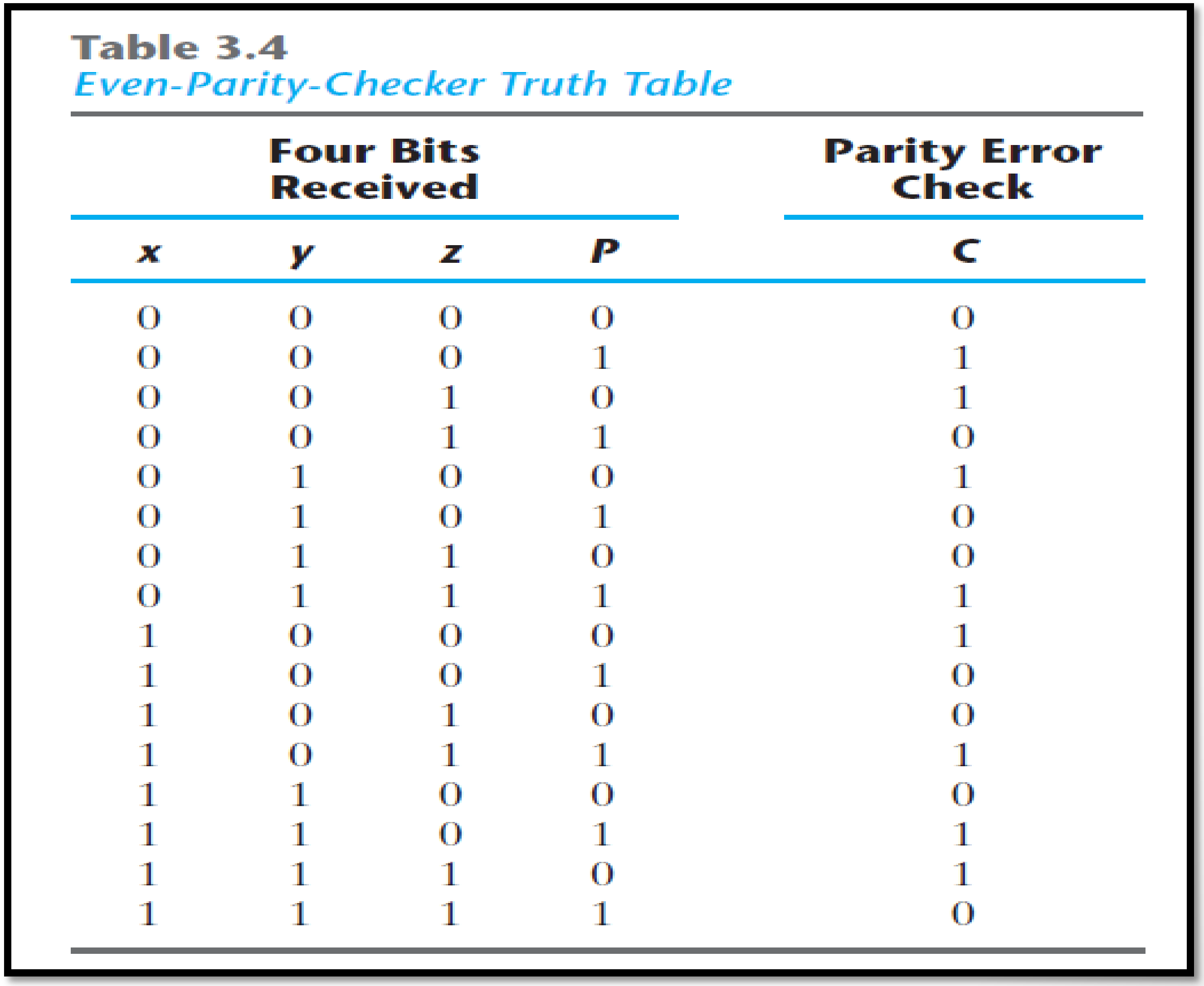


Table 3.4
Even-Parity-Checker Truth Table

Four Bits Received				Parity Error Check
x	*y*	*z*	*P*	*C*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

INTEGRATED CIRCUITS

- An integrated circuit (IC) is fabricated on silicon semiconductor crystal, called a *chip,* containing the electronic components for constructing digital gates.
- The various gates are interconnected inside the chip to form the required circuit.
- The chip is mounted in a ceramic or plastic container, and connections are welded to external pins to form the integrated circuit.
- The number of pins may range from 14 on a small IC package to several thousand on a larger package.
- Each IC has a numeric designation printed on the surface of the package for identification.
- Vendors provide data books, catalogs, and Internet websites that contain descriptions and information about the ICs that they manufacture.

INTEGRATED CIRCUITS (Levels)

- Digital ICs are often categorized according to the complexity of their circuits, as measured by the number of logic gates in a single package.
- The differentiation between those chips which have a few internal gates and those having hundreds of thousands of gates is made by customary reference to a package as being either a small-, medium-, large-, or very large-scale integration device.

INTEGRATED CIRCUITS (Levels)

- *Small-scale integration* (SSI) devices contain several independent gates in a single package.
- The inputs and outputs of the gates are connected directly to the pins in the package.
- The number of gates is usually fewer than 10 and is limited by the number of pins available in the IC.

INTEGRATED CIRCUITS (Levels)

- *Medium-scale integration* (MSI) devices have a complexity of approximately 10 to 1,000 gates in a single package.
- They usually perform specific elementary digital operations.
- MSI digital functions are introduced in Chapter 4 as decoders, adders, and multiplexers and in Chapter 6 as registers and counters.

INTEGRATED CIRCUITS (Levels)

- *Large-scale integration* (LSI) devices contain thousands of gates in a single package.
- They include digital systems such as processors, memory chips, and programmable logic devices.
- Some LSI components are presented in Chapter 7 .

INTEGRATED CIRCUITS (Levels)

- *Very large-scale integration* (VLSI) devices now contain millions of gates within a single package.
- Examples are large memory arrays and complex microcomputer chips.
- Because of their small size and low cost, VLSI devices have revolutionized the computer system design technology, giving the designer the capability to create structures that were previously uneconomical to build.

INTEGRATED CIRCUITS (Logic Families)

- Digital integrated circuits are classified not only by their complexity or logical operation, but also by the specific circuit technology to which they belong.
- The circuit technology is referred to as a *digital logic family*.
- Each logic family has its own basic electronic circuit upon which more complex digital circuits and components are developed.
- The basic circuit in each technology is a NAND, NOR, or inverter gate.

INTEGRATED CIRCUITS (Logic Families)

- TTL→ transistor–transistor logic;
- ECL→ emitter-coupled logic;
- MOS→ metal-oxide semiconductor;
- CMOS→ complementary metal-oxide semiconductor.

INTEGRATED CIRCUITS (Logic Families)

- TTL has been in use for 50 years and standard.
- The term transistor-transistor is because both logic function and amplification is done by transistor. In TTL logic family, analog value from 0 V to 0.8 V is logic 0 and 2 V to 5 V is logic 1. Advantages of the TTL logic families include high switching speed (125 MHz), less noise and more current (3 mA) driving capability.

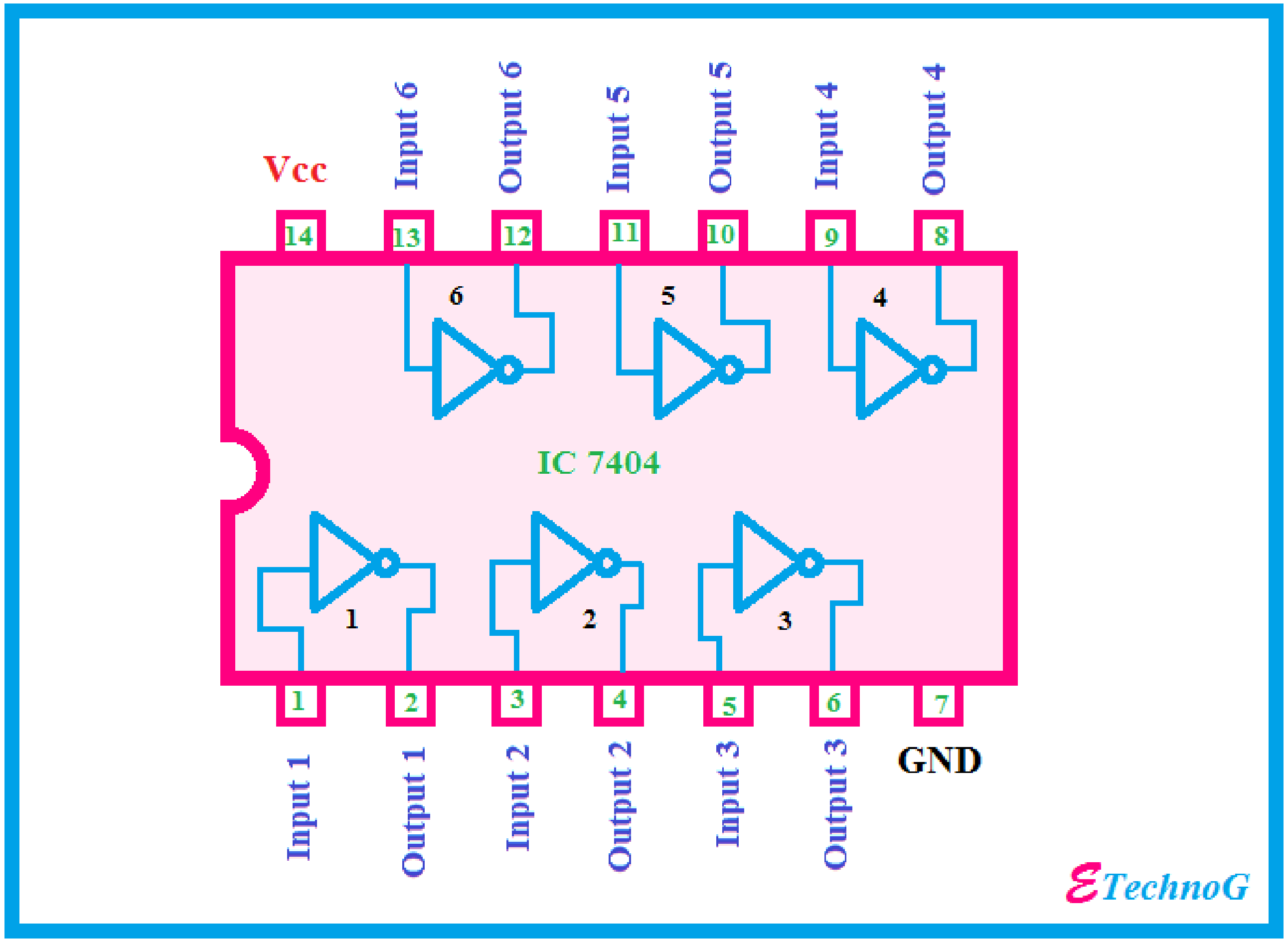
INTEGRATED CIRCUITS (Logic Families)

- ECL has an advantage in systems requiring high-speed operation.
- For logic one it is -0.9 and for logic zero, it is -1.75v.
- This improves the speed of operation at the expense of noise margin. Propagation time for an ECL gate is 0.5 to 2ns, which is very less. ECL logic families requires large currents therefore power dissipation is 3 to 10 times higher than that of TTL logic families. Because of its large power consumption and high requirement of silicon area, CMOS logic gates are preferred over ECL logic families in large scale integrated circuits.

INTEGRATED CIRCUITS (Logic Families)

- MOS is suitable for circuits that need high component density.
- CMOS is preferable in systems requiring low power consumption, such as digital cameras, personal media players and is essential for VLSI design; therefore, CMOS has become the dominant logic family.
- Because of high noise immunity and low static power dissipation, now CMOS logic families is most preferred in large scale integrated circuits.
- The CMOS model number will have a C in the middle of it, e.g., the 74C04 is the CMOS equivalent to the TTL 7404.

INTEGRATED CIRCUITS (Logic Families)



INTEGRATED CIRCUITS (Logic Families)

- The most important parameters distinguishing logic families:
- *Fan-out* specifies the number of standard loads that the output of a typical gate can drive without impairing its normal operation. A standard load is usually defined as the amount of current needed by an input of another similar gate in the same family.
- *Fan-in* is the number of inputs available in a gate.

INTEGRATED CIRCUITS (Logic Families)

- *Power dissipation* is the power consumed by the gate that must be available from the power supply.
- *Propagation delay* is the average transition delay time for a signal to propagate from input to output.
- *Noise margin* is the maximum external noise voltage added to an input signal that does not cause an undesirable change in the circuit output.