INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Analyzing and Optimizing Combinational Logic Circuit

Sparsh Mittal

Combinational circuits

- Logic circuits for digital systems may be combinational or sequential.
- A combinational circuit consists of logic gates whose outputs at any time are determined from only the present combination of inputs.
- A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.

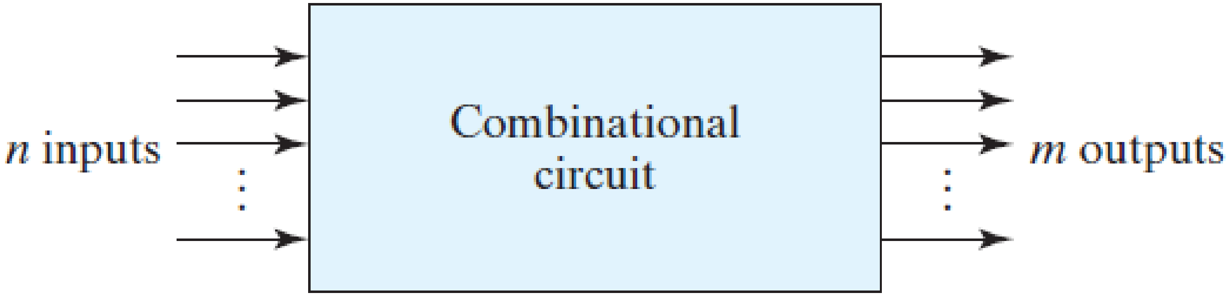


FIGURE 4.1
Block diagram of combinational circuit

Sequential circuits

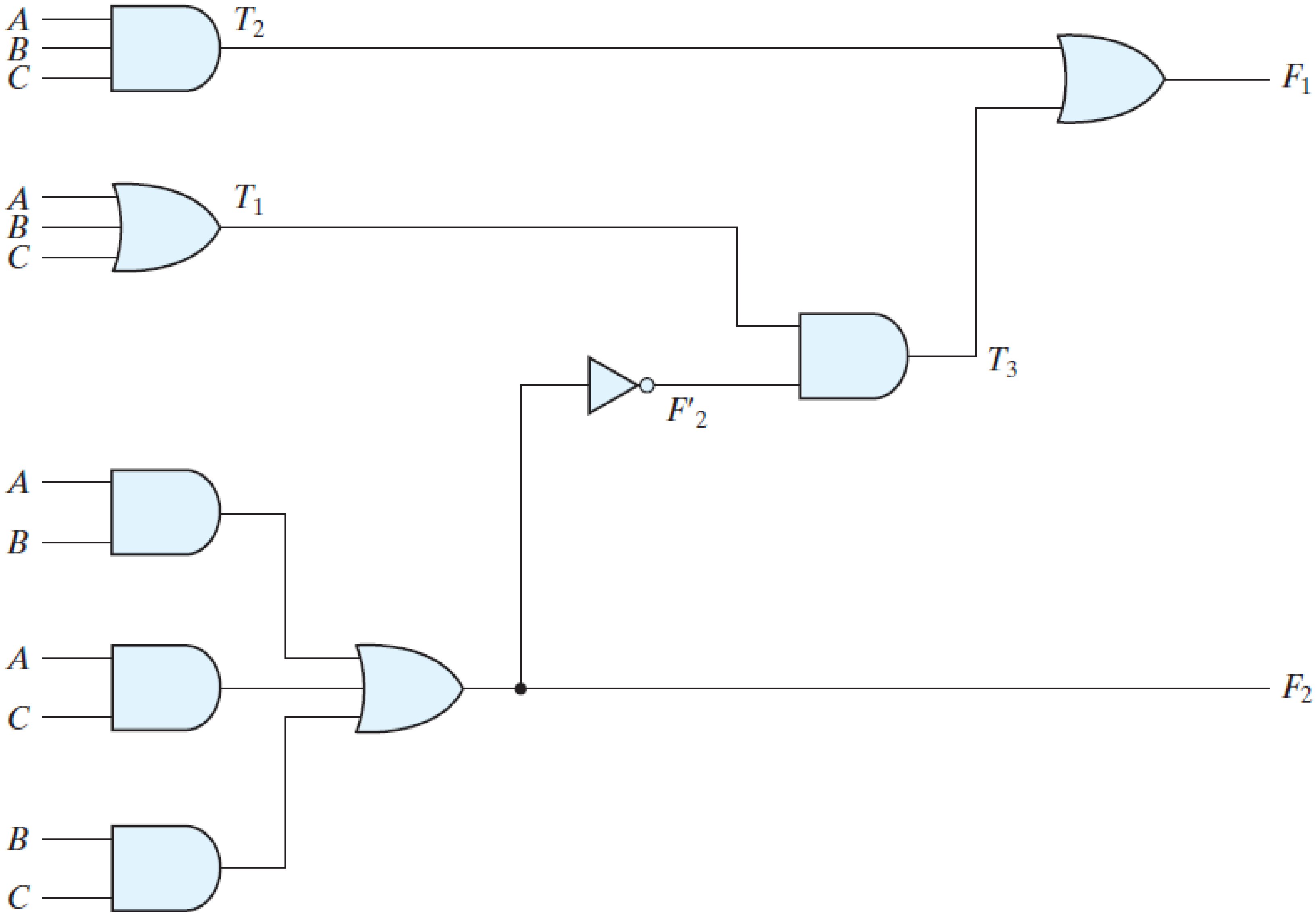
- Sequential circuits employ storage elements in addition to logic gates.
- Their outputs are a function of the inputs and the state of the storage elements.
- Because the state of the storage elements is a function of previous inputs, the outputs of a sequential circuit depend not only on present values of inputs, but also on past inputs, and the circuit behavior must be specified by a time sequence of inputs and internal states.

Combinational circuit

- A combinational circuit can be specified with a truth table that lists the output values for each combination of input variables.
- A combinational circuit also can be described by m Boolean functions, one for each output variable.
- Each output function is expressed in terms of the n input variables.
- **The diagram of a combinational circuit has logic gates with no feedback paths** or **memory elements.**
- A feedback path is a connection from the output of one gate to the input of a second gate whose output forms part of the input to the first gate.
- Feedback paths in a digital circuit define a sequential circuit

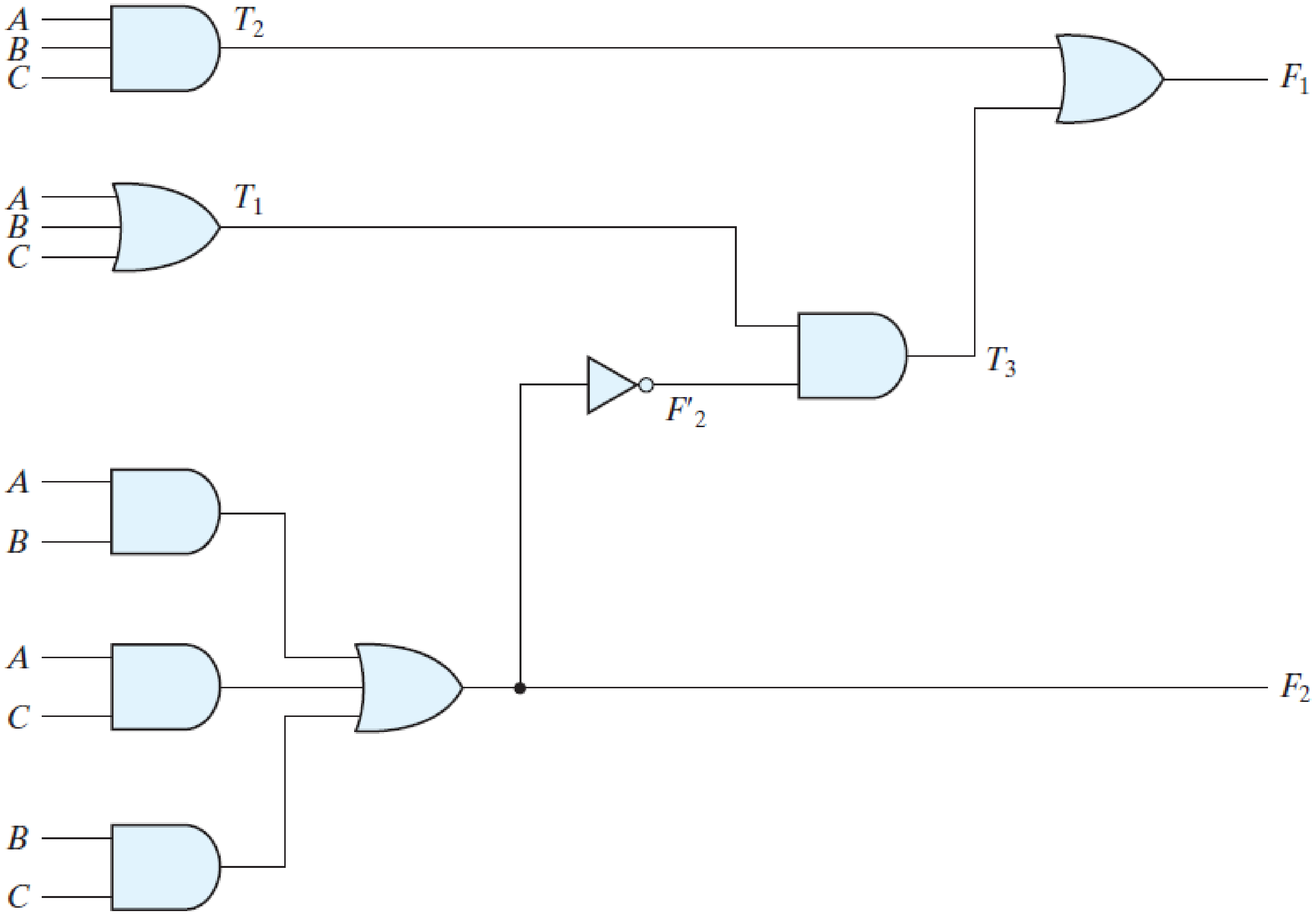
Example: analyzing a circuit

- Express F1 and F2 as function of A, B and C



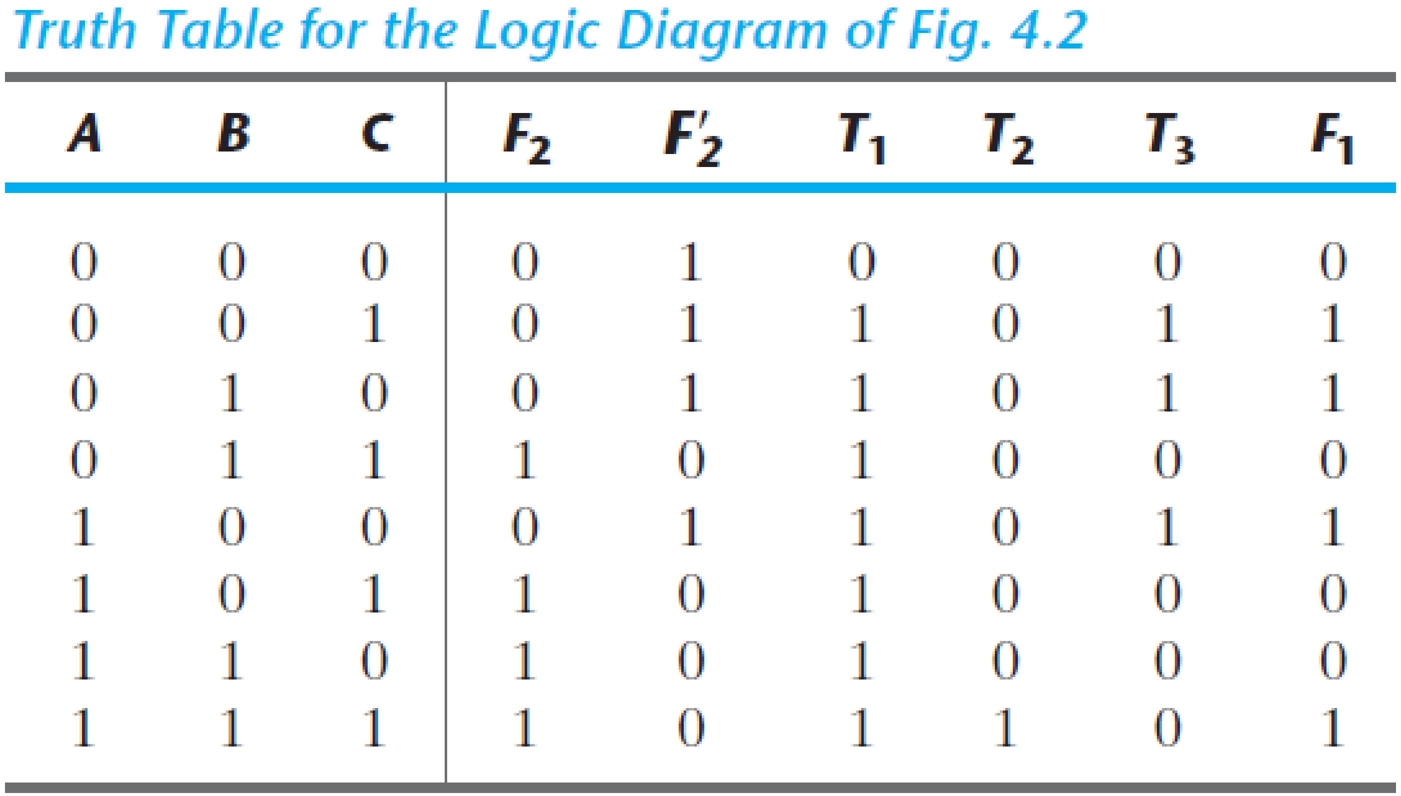
$$F_2 = AB + AC + BC$$
$$T_1 = A + B + C$$
$$T_2 = ABC$$

$$T_3 = F'_2 T_1$$
$$F_1 = T_3 + T_2$$



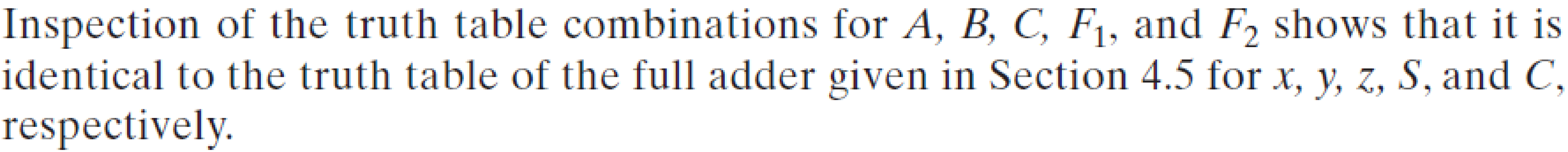
$$\begin{aligned}F_1 &= T_3 + T_2 = F'_2 T_1 + ABC = (AB + AC + BC)'(A + B + C) + ABC\\ &= (A' + B')(A' + C')(B' + C')(A + B + C) + ABC\\ &= (A' + B'C')(AB' + AC' + BC' + B'C) + ABC\\ &= A'BC' + A'B'C + AB'C' + ABC\end{aligned}$$

Determining truth table



Truth Table for the Logic Diagram of Fig. 4.2

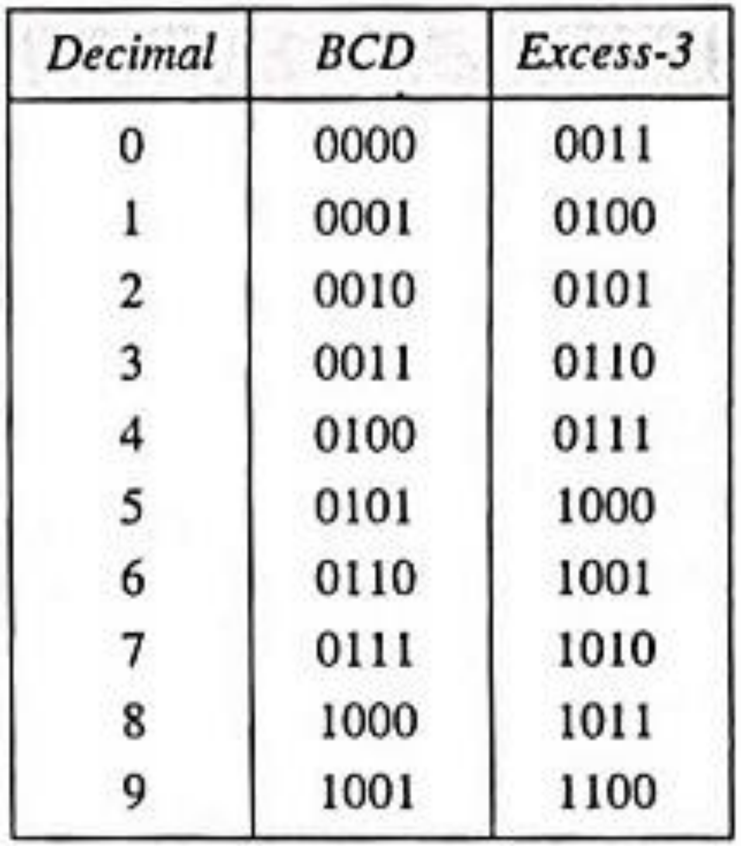
A	B	C	F_2	F'_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1



Inspection of the truth table combinations for A, B, C, F_1, and F_2 shows that it is identical to the truth table of the full adder given in Section 4.5 for x, y, z, S, and C, respectively.

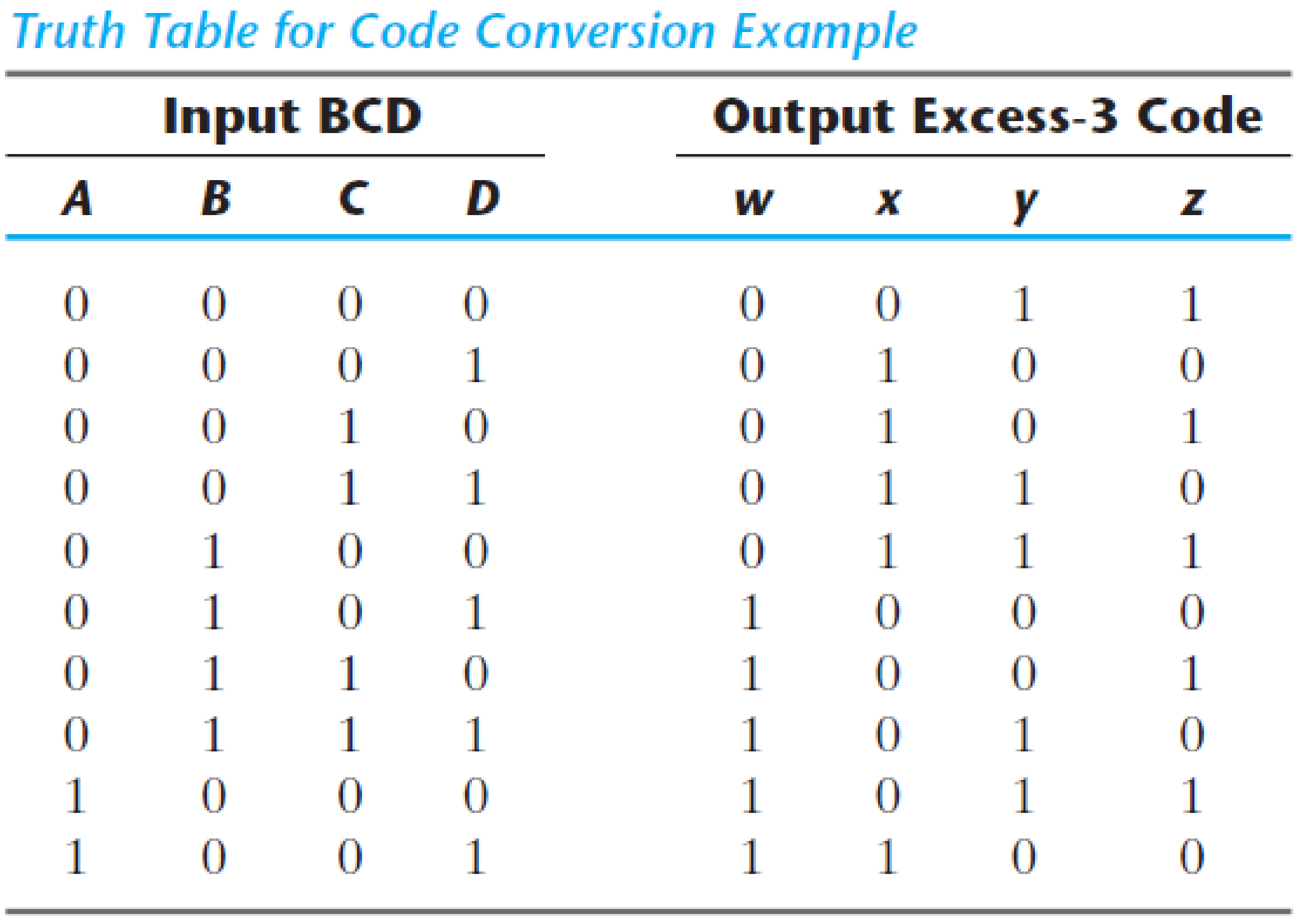
Code conversion example

- Converts binary coded decimal (BCD) to the excess-3 code for the decimal digits.



Decimal	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

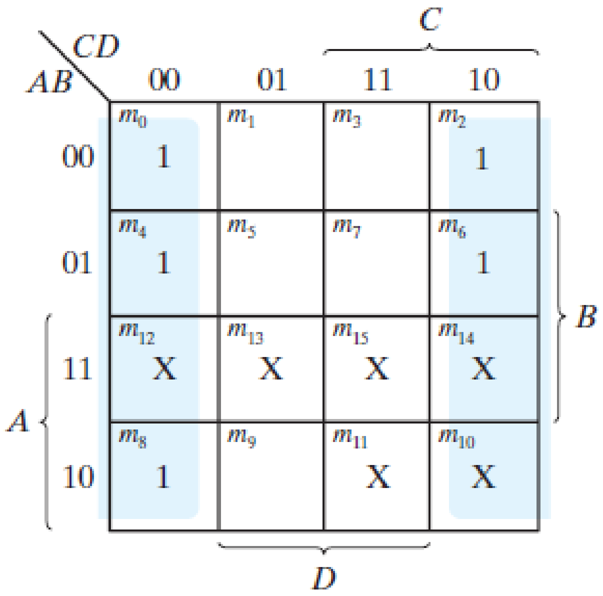
BCD to Excess-3 conversion



Truth Table for Code Conversion Example

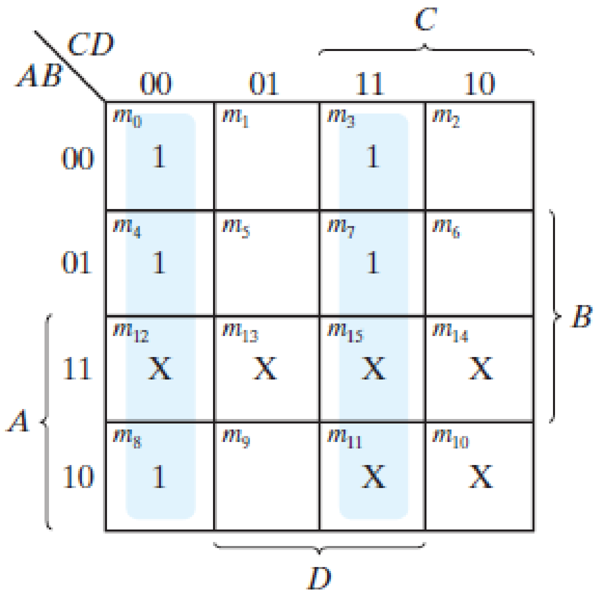
Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

- We designate the four input binary variables by symbols *A, B, C,* and *D,* and four output variables by *w, x, y*, and *z.*
- Four binary variables may have 16 bit combinations, but only 10 are listed in the truth table.
- The six input bit-combinations not listed for the input variables are don’t-care combinations.
- We assume that they will never occur.
- Therefore, we are at liberty to assign to the output variables either a 1 or a 0, whichever gives a simpler circuit.



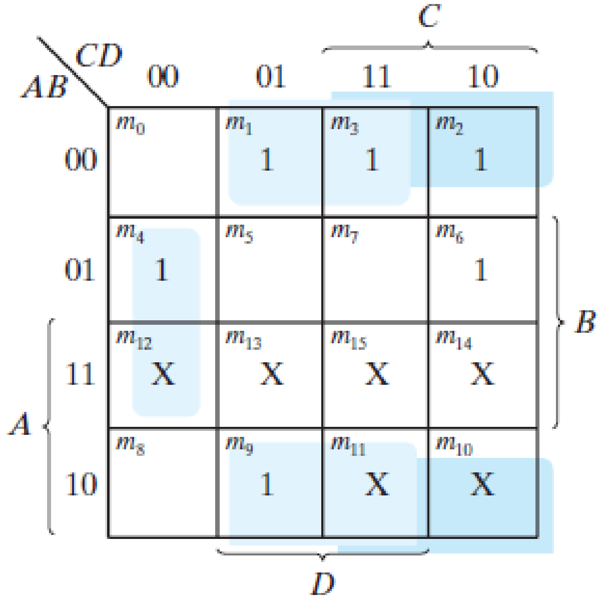
AB \ CD	00	01	11	10
00	m_0 1	m_1	m_3	m_2 1
01	m_4 1	m_5	m_7	m_6 1
11	m_{12} X	m_{13} X	m_{15} X	m_{14} X
10	m_8 1	m_9	m_{11} X	m_{10} X

$z = D'$



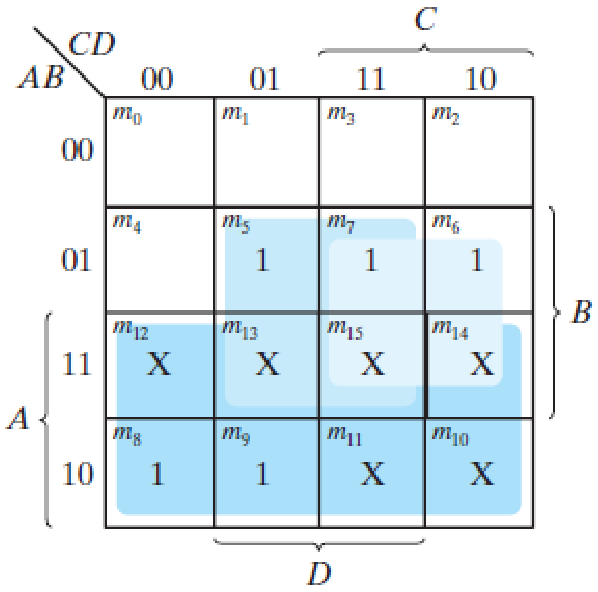
AB \ CD	00	01	11	10
00	m_0 1	m_1	m_3 1	m_2
01	m_4 1	m_5	m_7 1	m_6
11	m_{12} X	m_{13} X	m_{15} X	m_{14} X
10	m_8 1	m_9	m_{11} X	m_{10} X

$y = CD + C'D'$



AB \ CD	00	01	11	10
00	m_0	m_1 1	m_3 1	m_2 1
01	m_4 1	m_5	m_7	m_6 1
11	m_{12} X	m_{13} X	m_{15} X	m_{14} X
10	m_8	m_9 1	m_{11} X	m_{10} X

$x = B'C + B'D + BC'D'$

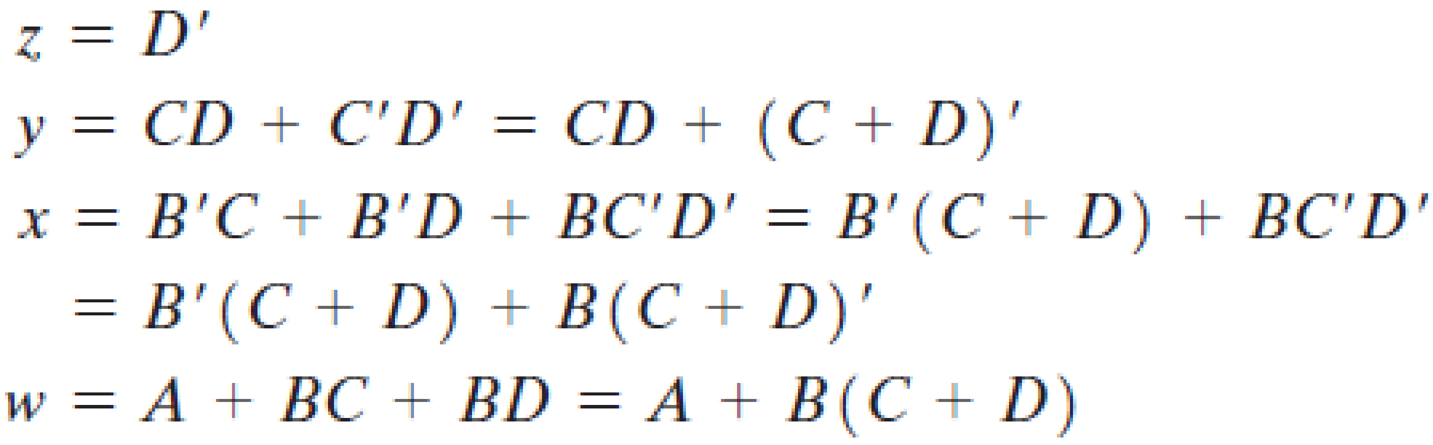


AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5 1	m_7 1	m_6 1
11	m_{12} X	m_{13} X	m_{15} X	m_{14} X
10	m_8 1	m_9 1	m_{11} X	m_{10} X

$w = A + BC+ BD$

Further optimization

- The expressions obtained may be manipulated algebraically for using common gates for two or more outputs.
- This manipulation illustrates the flexibility obtained with multiple-output systems when implemented with three or more levels of gates:



$$
\begin{aligned}
z &= D' \\
y &= CD + C'D' = CD + (C + D)' \\
x &= B'C + B'D + BC'D' = B'(C + D) + BC'D' \\
&= B'(C + D) + B(C + D)' \\
w &= A + BC + BD = A + B(C + D)
\end{aligned}
$$

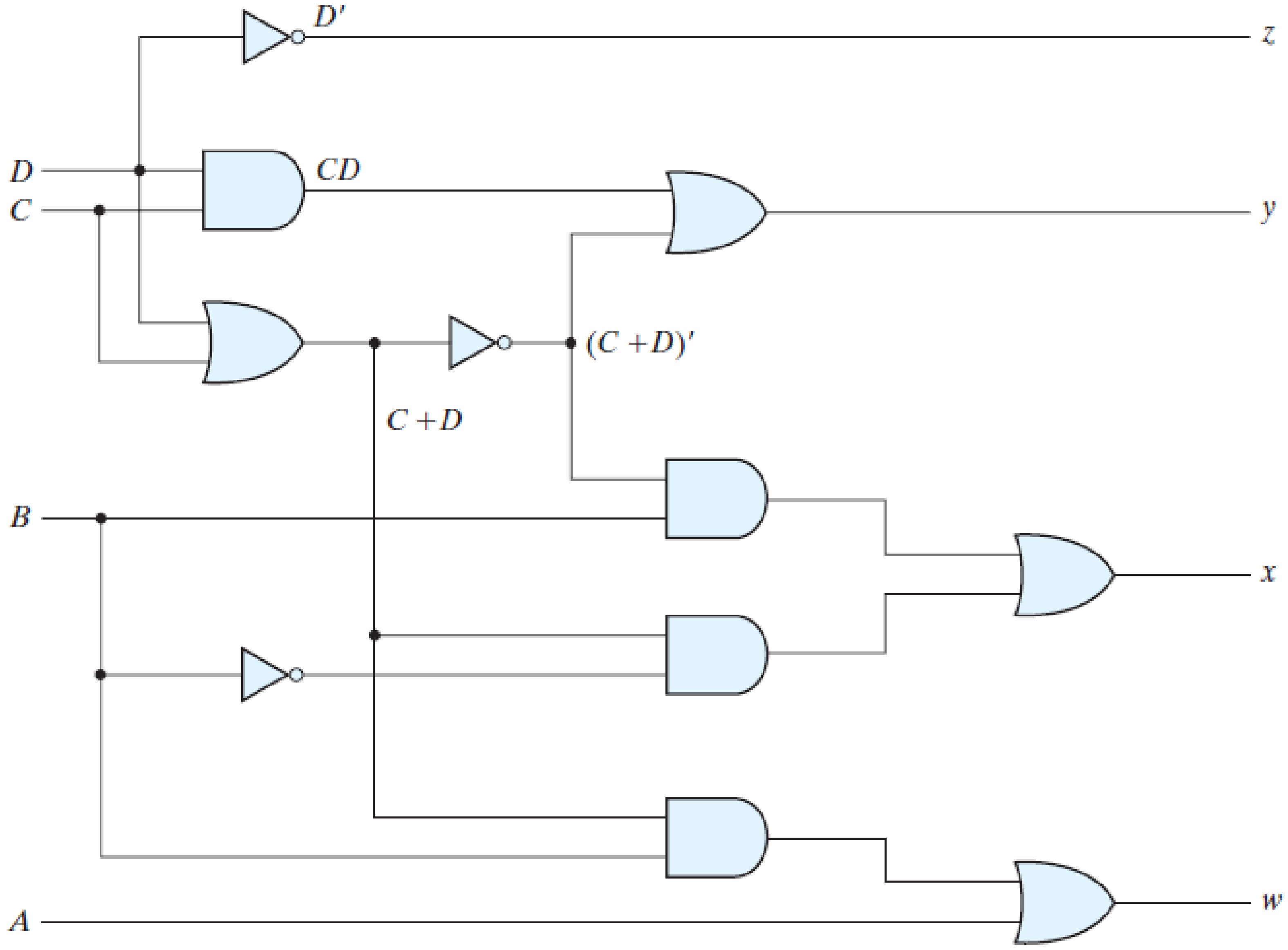


FIGURE 4.4

Logic diagram for BCD-to-excess-3 code converter

Three-level logic circuit requires fewer gates, all of which in turn require no more than two inputs