Exercise Four

**3.11** Convert the following Boolean function from a sum-of-products form to a simplified

product-of-sums form.

*F*(*x, y, z*) = Σ(0, 1, 2, 5, 8, 10, 13)

**3.12** Simplify the following Boolean functions:

(b) *F*(*A, B, C, D*) = Σ(1, 3, 6, 9, 11, 12, 14)

**3.13** Simplify the following expressions to (1) sum-of-products and (2) products-of-sums:

(b) *ACD* ‘ + *C* ‘*D* + *AB* ‘+ *ABCD*

(c) (*A* ‘+ *B* + *D ‘)(A* ‘ + *B* ‘ + *C* ‘)(*A* ‘ + *B* ‘ + *C*)(*B* ‘ + *C* + *D* ‘)

**3.16** Simplify the following functions, and implement them with two-level NAND gate circuits:

(b) *F*(*A*, *B*, *C*, *D*) = *A* ‘*B* ‘*C* ‘*D* + *CD* + *AC* ‘*D*

(c) *F*(*A*, *B*, *C*) = (*A* ‘ + *C* ‘ + *D* ‘)(*A* ‘ + *C* ‘) (*C* ‘ + *D* ‘)

(d) *F*(*A*, *B*, *C*, *D*) = *A* ‘ + *B* + *D* ‘ + *B* ‘*C*

**3.17\*** Draw a NAND logic diagram that implements the complement of the following function:

*F*(*A, B, C, D*) =Σm(0, 1, 2, 3, 6, 10, 11, 14)

**3.18** Draw a logic diagram using only two-input NOR gates to implement the following function:

*F*(*A*, *B*, *C*, *D*) = (*A* ⊕ *B*)'(*C* ⊕ *D*)

**3.19** Simplify the following functions, and implement them with two-level NOR gate circuits:

(b) *F*(*w, x, y, z*) = Σm(0, 3, 12, 15)

(c) *F* (*x*, *y*, *z*) = [(*x* + *y*)(*x* + *z*)]’

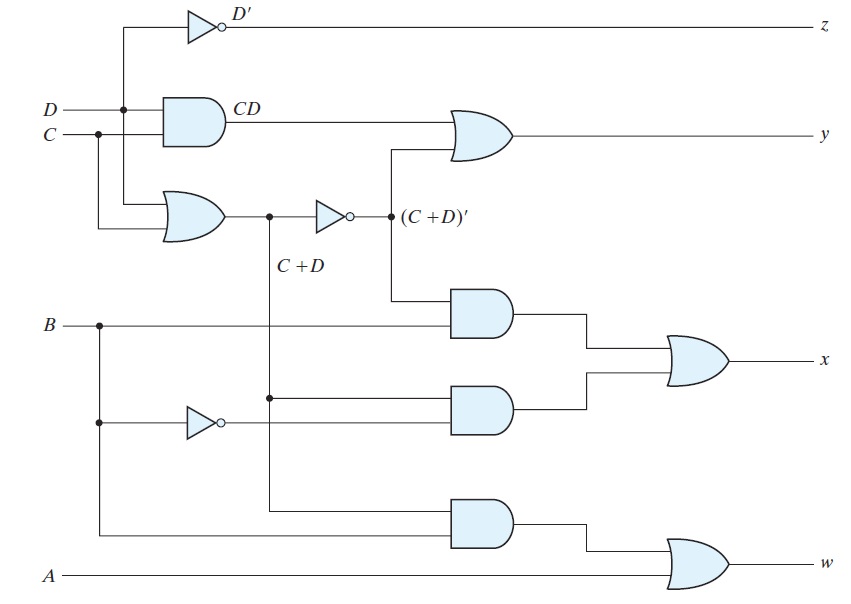
**3.20** Draw the multiple-level NOR circuit for the following expression:

*CD*(*B* + *C*)*A* + (*BC* ‘ + *DE* ‘)

**3.21** Draw the multiple-level NAND circuit for the following expression:

*W*(*x* + *y* + *z*) + *xyz*

**3.22** Convert the logic diagram of the following circuit into a multiple-level NAND circuit.



**3.24** Implement the following Boolean function *F*, using the two-level forms of logic

(a) NAND-AND,

(b) AND-NOR,

(c) OR-NAND, and

(d) NOR-OR:

*F*(*A*, *B*, *C*, *D*)= ΣM(0, 4, 8, 9, 10, 11, 12, 14)

**3.27** Show that the dual of the exclusive-OR is also its complement.

**3.30\*** Implement the following Boolean expression with exclusive-OR and AND gates:

*F* = *AB* ‘*CD* ‘+ *A*’*BCD* ‘ + *AB* ‘*C* ‘*D* + *A* ‘*BC* ‘*D*