1. Using or-gates and/or nor gates along with a 3-to-8-line decoder of the type shown in figure below, realize the following pairs of expressions. In each case, the gates should be selected so as to minimize their total number of input terminals.

a. $f_1(x,y,z)=\sum m(1,3)$

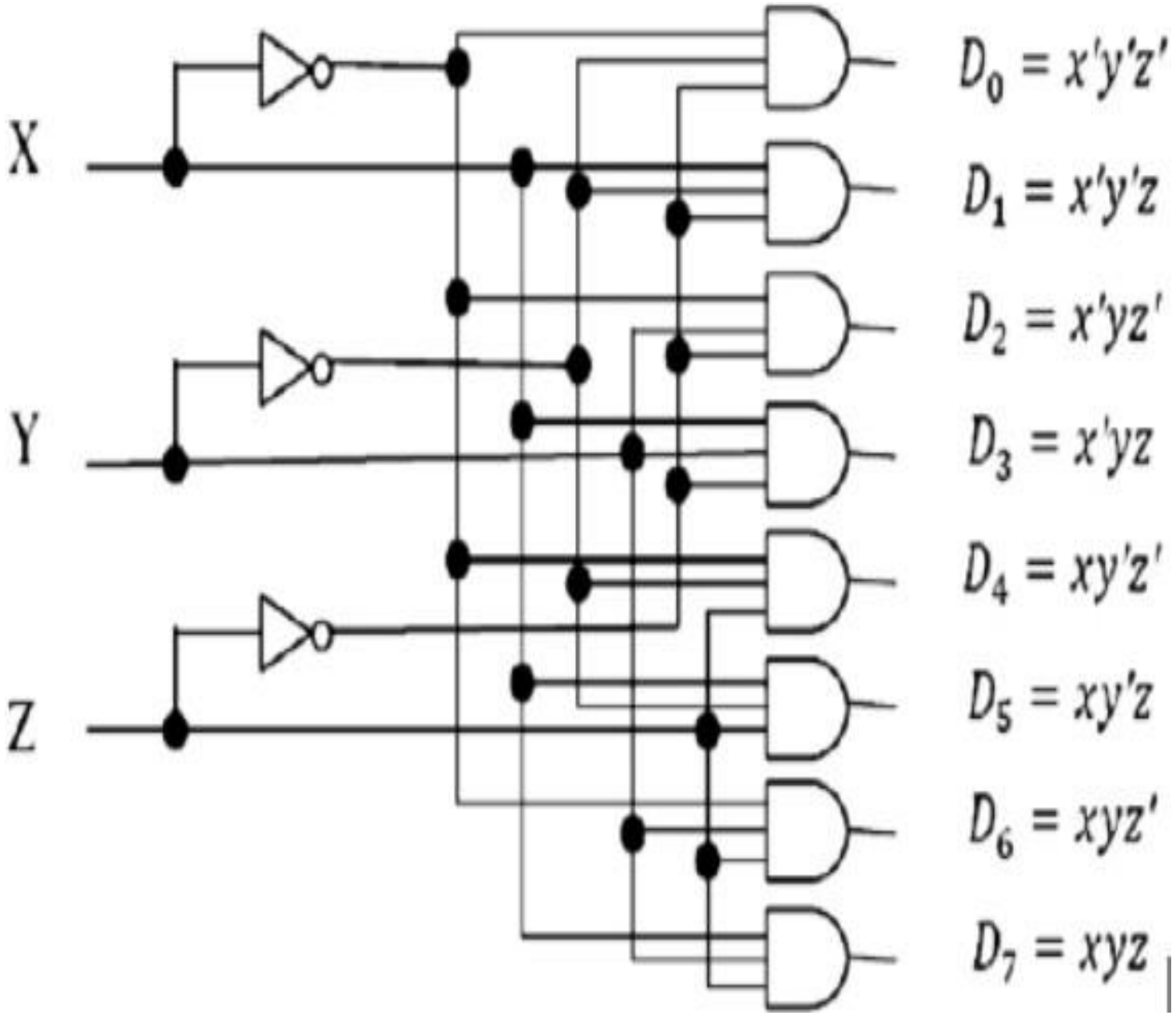
$f_2(x,y,z)=\sum m(3,6,7)$

b. $f_1(x,y,z)=\sum m(0,1,5,6,7)$

$f_2(x,y,z)=\sum m(1,2,3,6,7)$

c. $f_1(x,y,z)=\sum m(0,2,4)$

$f_2(x,y,z)=\sum m(1,2,4,5,7)$

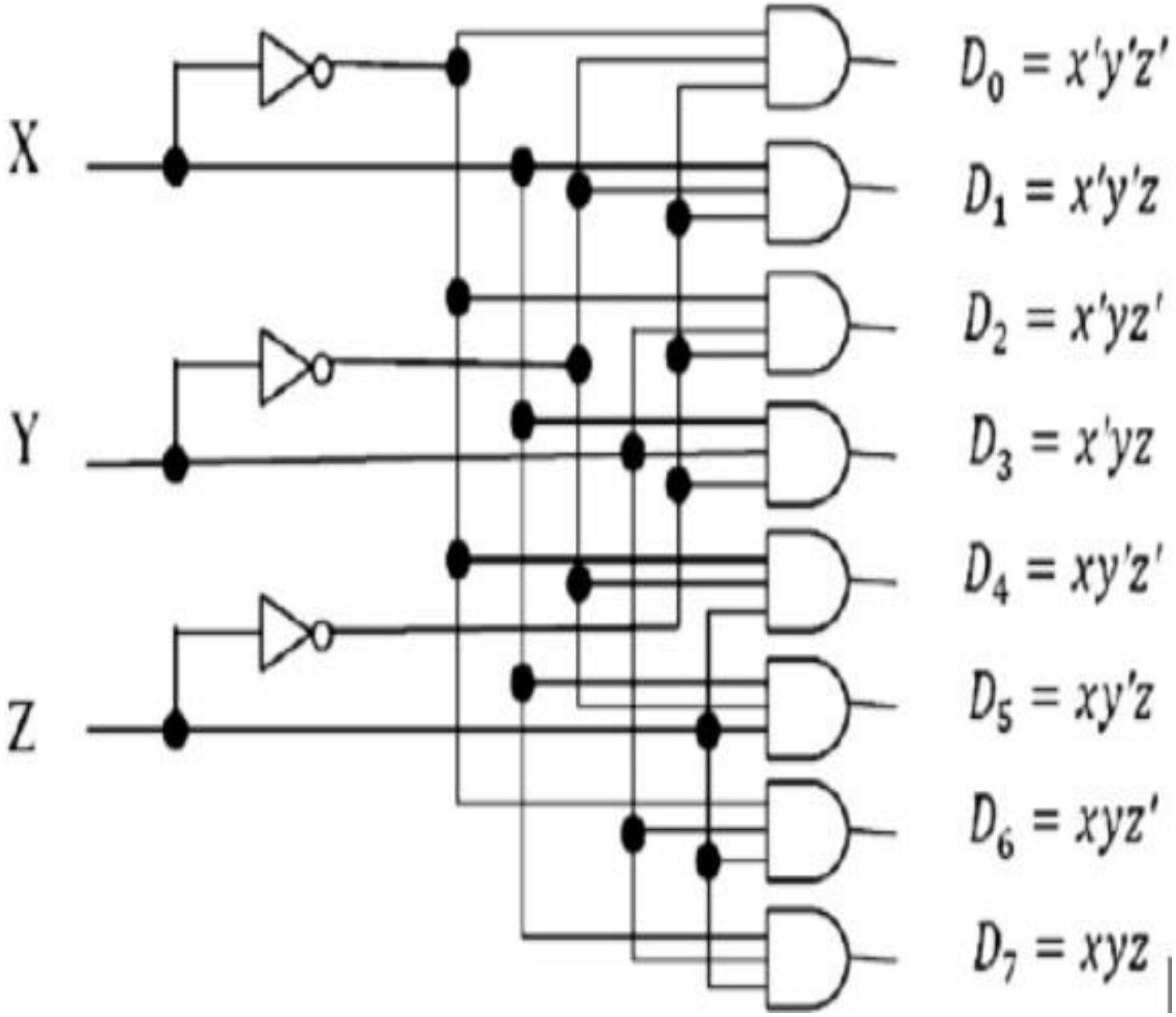

X
Y
Z
$D_0 = x'y'z'$
$D_1 = x'y'z$
$D_2 = x'yz'$
$D_3 = x'yz$
$D_4 = xy'z'$
$D_5 = xy'z$
$D_6 = xyz'$
$D_7 = xyz$

2. Using or-gates and/or nor gates along with a 3-to-8-line decoder of the type shown in figure below, realize the following pairs of expressions. In each case, the gates should be selected so as to minimize their total number of input terminals.

a. $f_1(x,y,z) = \prod M(0,3,5,6,7)$
$f_2(x,y,z) = \prod M(2,3,4,5,7)$

b. $f_1(x,y,z) = \prod M(0,1,7)$
$f_2(x,y,z) = \prod M(1,5,7)$

c. $f_1(x,y,z) = \prod M(1,2,5)$
$f_2(x,y,z) = \prod M(0,1,3,5,7)$


X
Y
Z
$D_0 = x'y'z'$
$D_1 = x'y'z$
$D_2 = x'yz'$
$D_3 = x'yz$
$D_4 = xy'z'$
$D_5 = xy'z$
$D_6 = xyz'$
$D_7 = xyz$

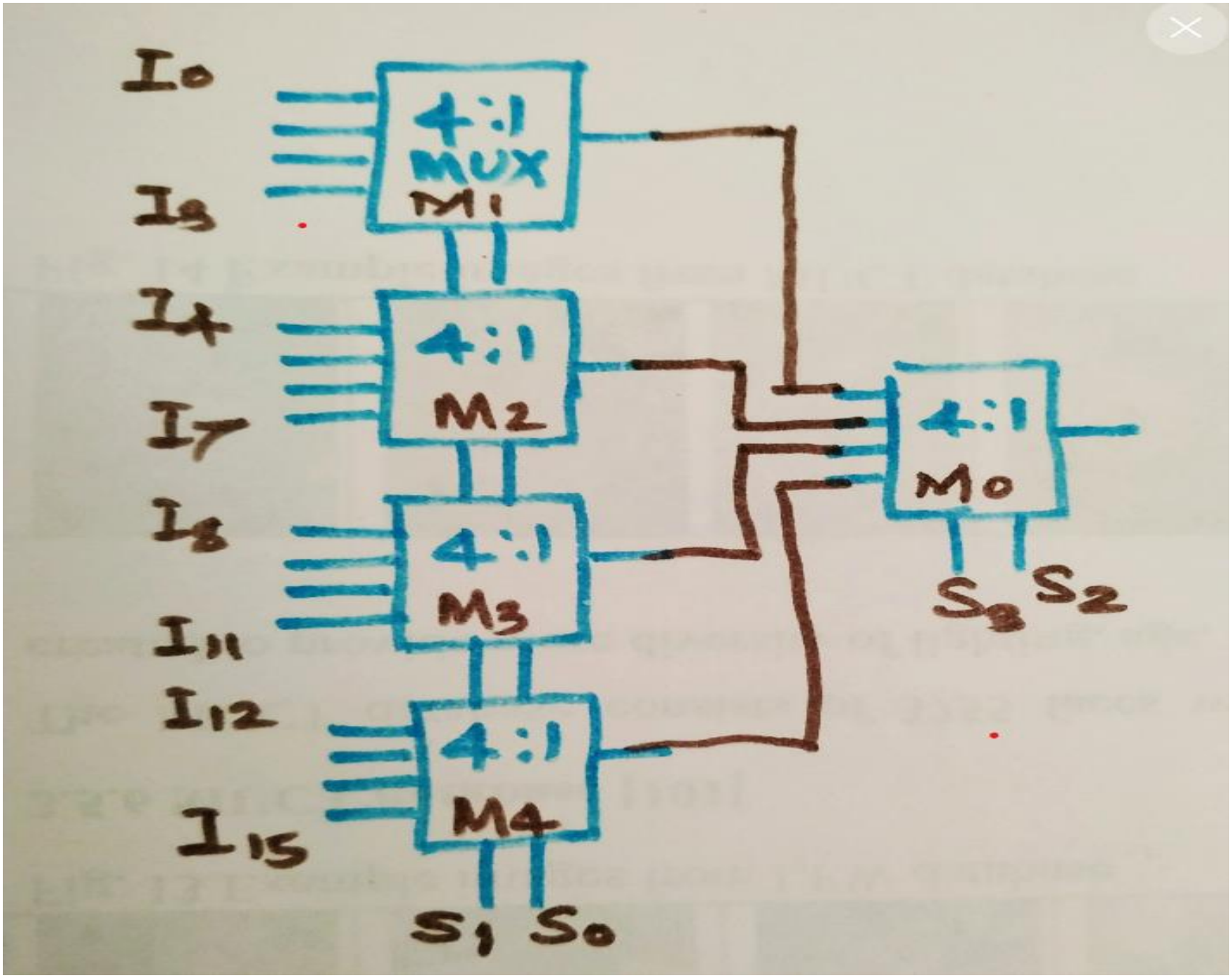
3.Using a 4 to 16 decoder constructed from NAND gates and having an enable input E′, design an excess 3 to 8421 code converter. select gates so as minimize their total number of input terminals.

4.Write the condensed truth table for a 4-to-2-line priority encoder with a valid output where the highest priority is given to the input having the highest index. Determine the minimal sum equations for the three outputs.

5.Figure given below the structure of a 16-to-1- line multiplexer constructed from only 4-to-1-line multiplexers. Other structures are possible depending upon the type of multiplexer used. Construct a multiplexer tree for a 16-to-1-line multiplexer.
a. using only 2-to-1-line multiplexers.

b. using 2-to-1 line and 4-to-1-line multiplexers (Note: three different structures are possible)

c. using 2-to-1 line and 8-to-1-line multiplexers (Note: two different structures are possible)


I0
I3
I4
I7
I8
I11
I12
I15
4:1 MUX M1
4:1 M2
4:1 M3
4:1 M4
4:1 M0
S3
S2
S1
S0