Chapter4: Combinational Logic

Lecture6- Study Decoders, Function Implementation using Decoders

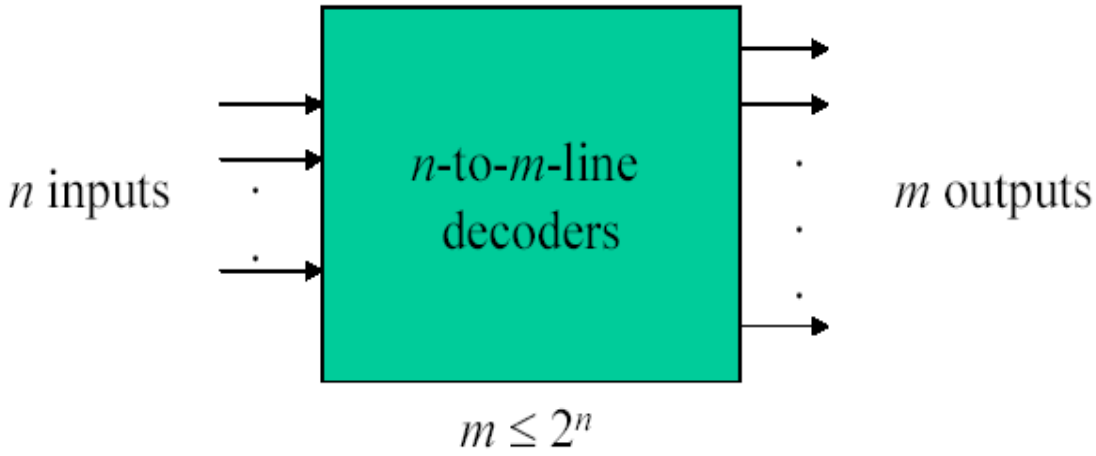
Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Objectives

- Study design and applications of Decoders
- Function implementation using Decoder blocks

Decoders

- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. Only one output can be active (high) at any time.
- Decoders are a class of combinational logic circuits that convert a set of input variables representing a code into a set of output variables representing a different code. The relationship between the input and output codes can be expressed in a truth table i.e 4-to-10-lines decoder circuit.
- If the n-bit coded information has unused combinations, the decoder has fewer than 2^n output



Decoder Applications

- Generate minterms/complement of minterms and are used for functions implementation.
- Memory addressing- Decoders are widely used in the memory system of a computer where they respond to the address code generated by the central processor to activate a particular memory location.
- Code Conversion. Example is BCD-to-7-segment decoder.
- DeMUX function.
- Used in conjunction with counters to decode (detect) counter states and provide timing or sequencing signals.
- Provide enabling inputs when used in the design of MUXs with tri-state gates.
- Computers communicate with peripheral devices (printers, modems, scanners, keyboards, video monitors, external disk drives and other computers) by sending and/or receiving data through I/O ports. Decoders are used to select I/O as determined by the computer to receive or send data.

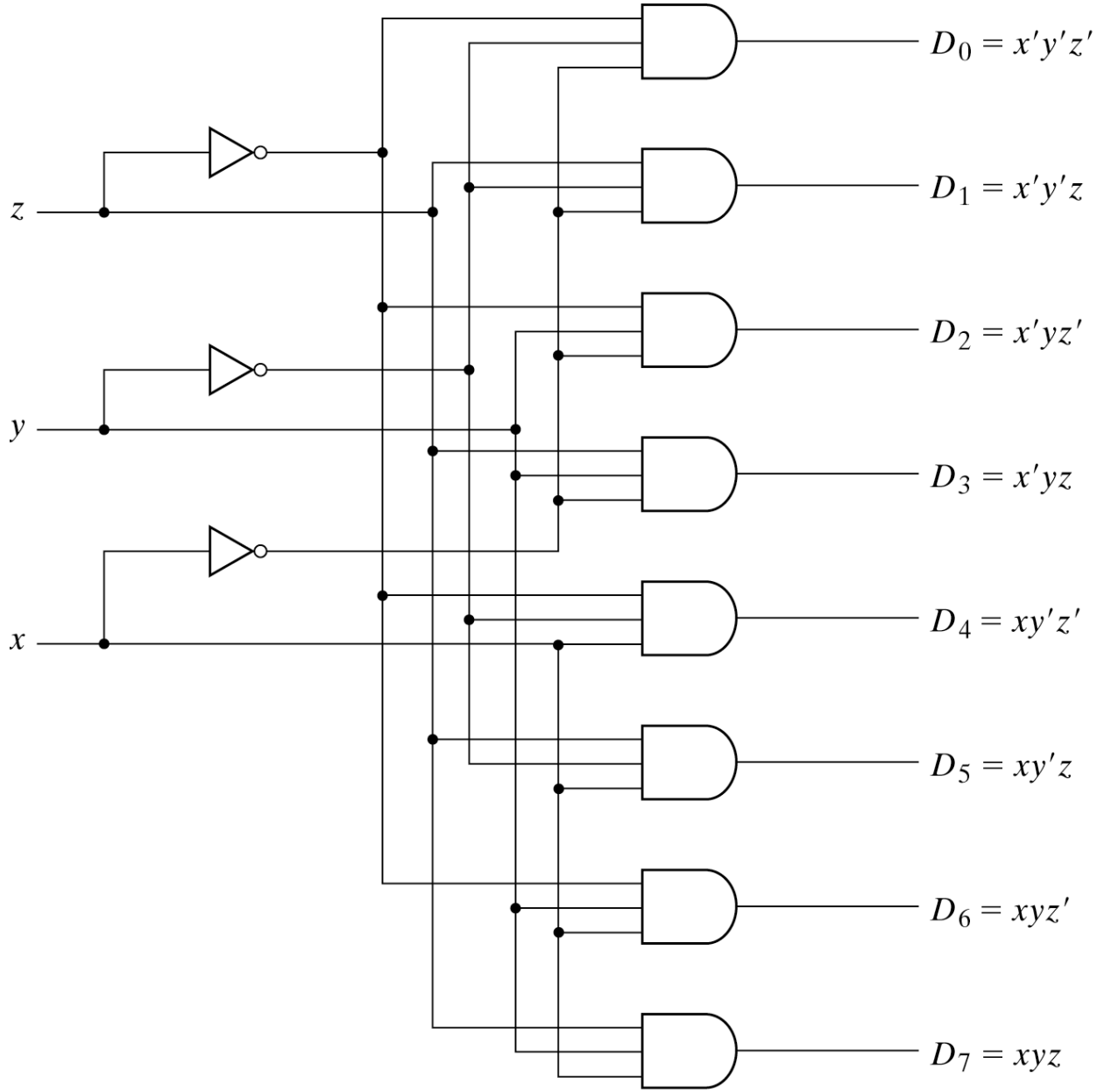


Fig. 4-18 3-to-8-Line Decoder

Implementation of Full Adder with a Decoder

- There are three inputs and eight outputs so we need 3-to-8-line decoder
- Two OR gates are required for logical sum of the desired minterms

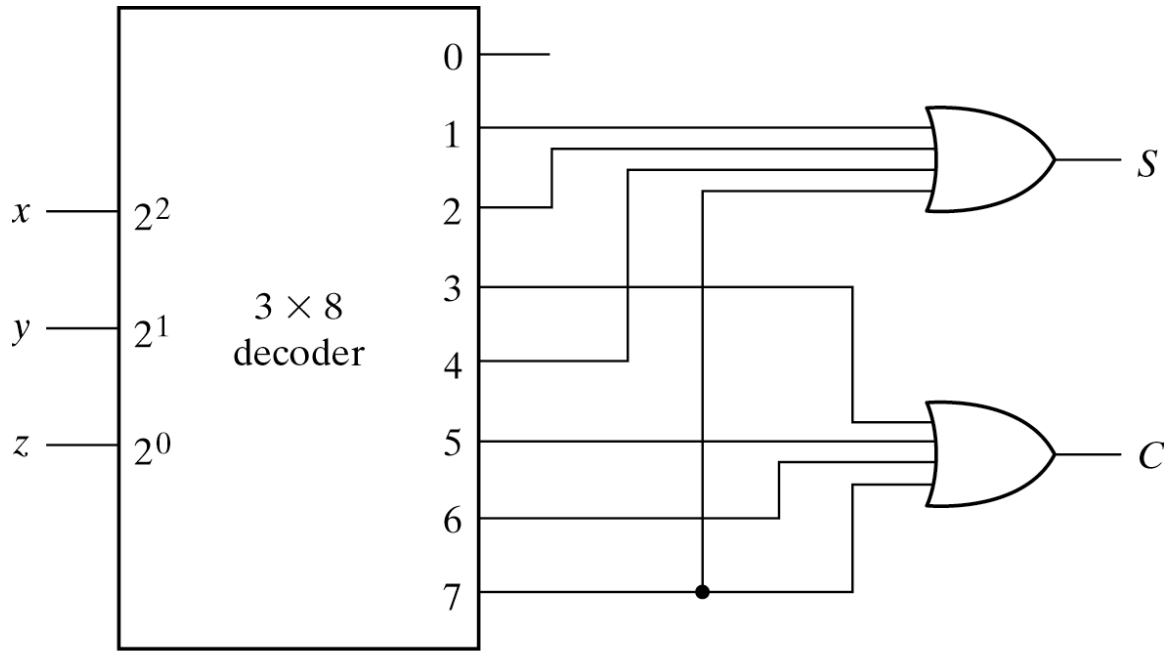
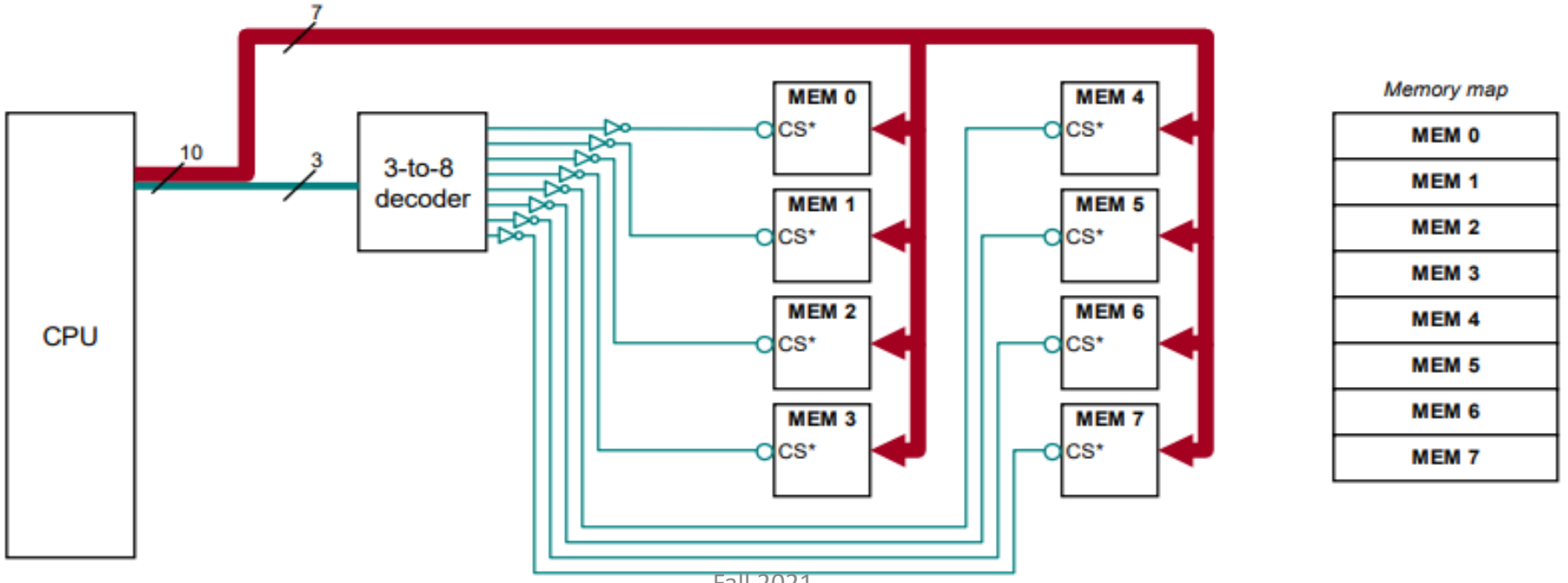


Fig. 4-21 Implementation of a Full Adder with a Decoder

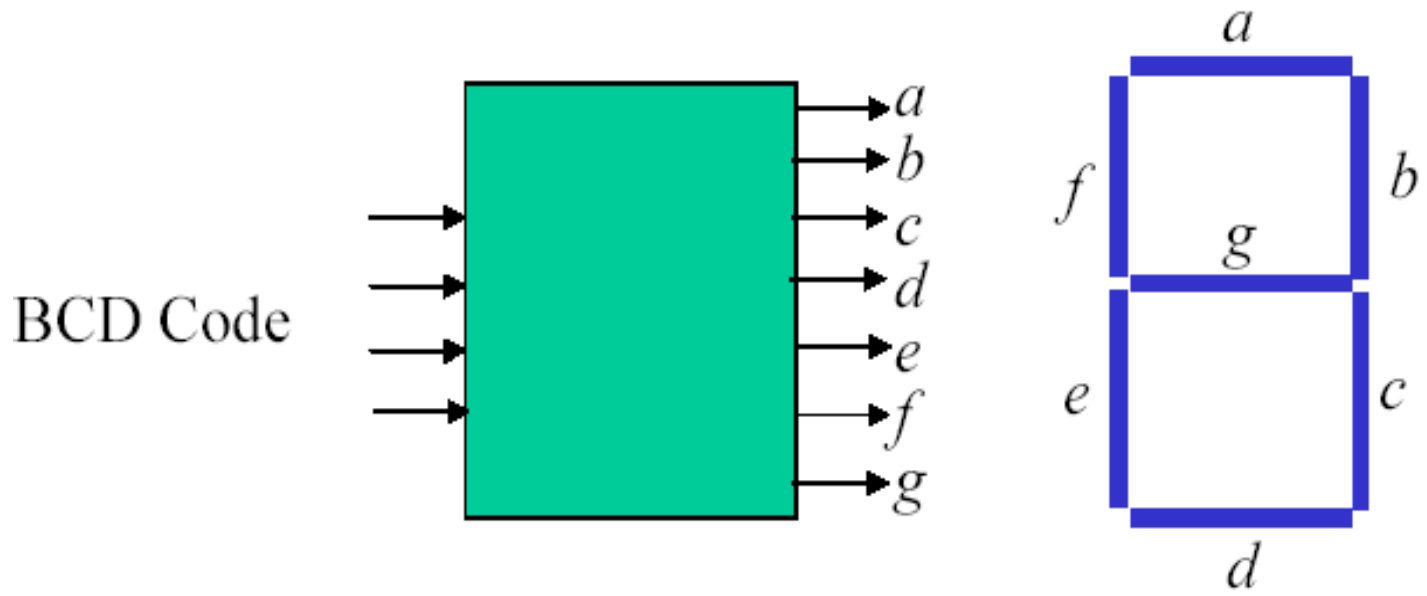
A very simple example

- **Let's assume a very simple microprocessor with 10 address lines (1KB memory)**
- **Let's assume we wish to implement all its memory space and we use 128x8 memory chips**
- **SOLUTION**
  - We will need 8 memory chips (8x128=1024)
  - We will need 3 address lines to select each one of the 8 chips
  - Each chip will need 7 address lines to address its internal memory cells



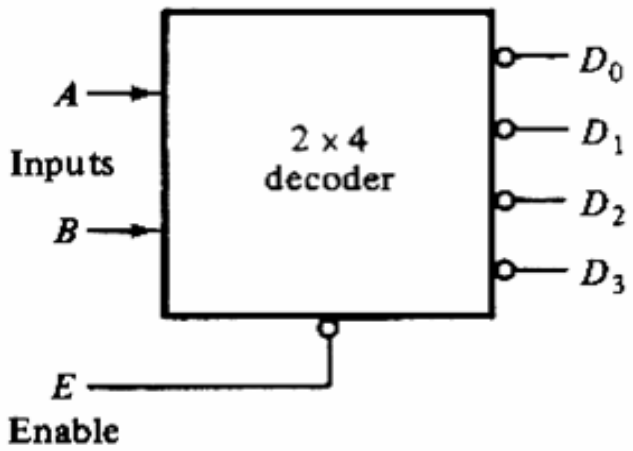
Decoder Example (Code Converter)

- BCD-to-seven-segment display converter is one common example of code converters, which converts one BCD digit into information suitable for driving a digit-oriented display.

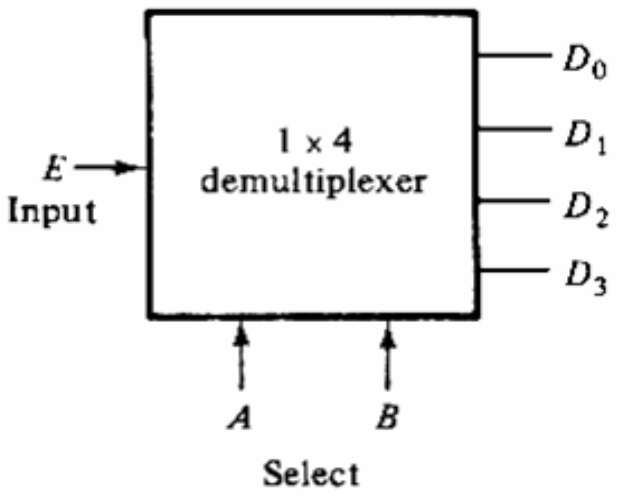


Demultiplexer

- A decoder with an enable input (Figure 4-19) can function as demultiplexer (1-to-4-line demultiplexer)
  - ➢E is taken as data input line and A and B are takes as selection inputs



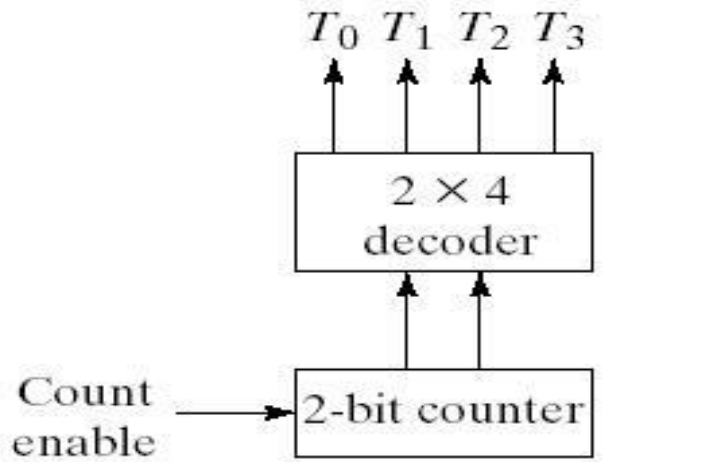
(a) Decoder with enable



(b) Demultiplexer

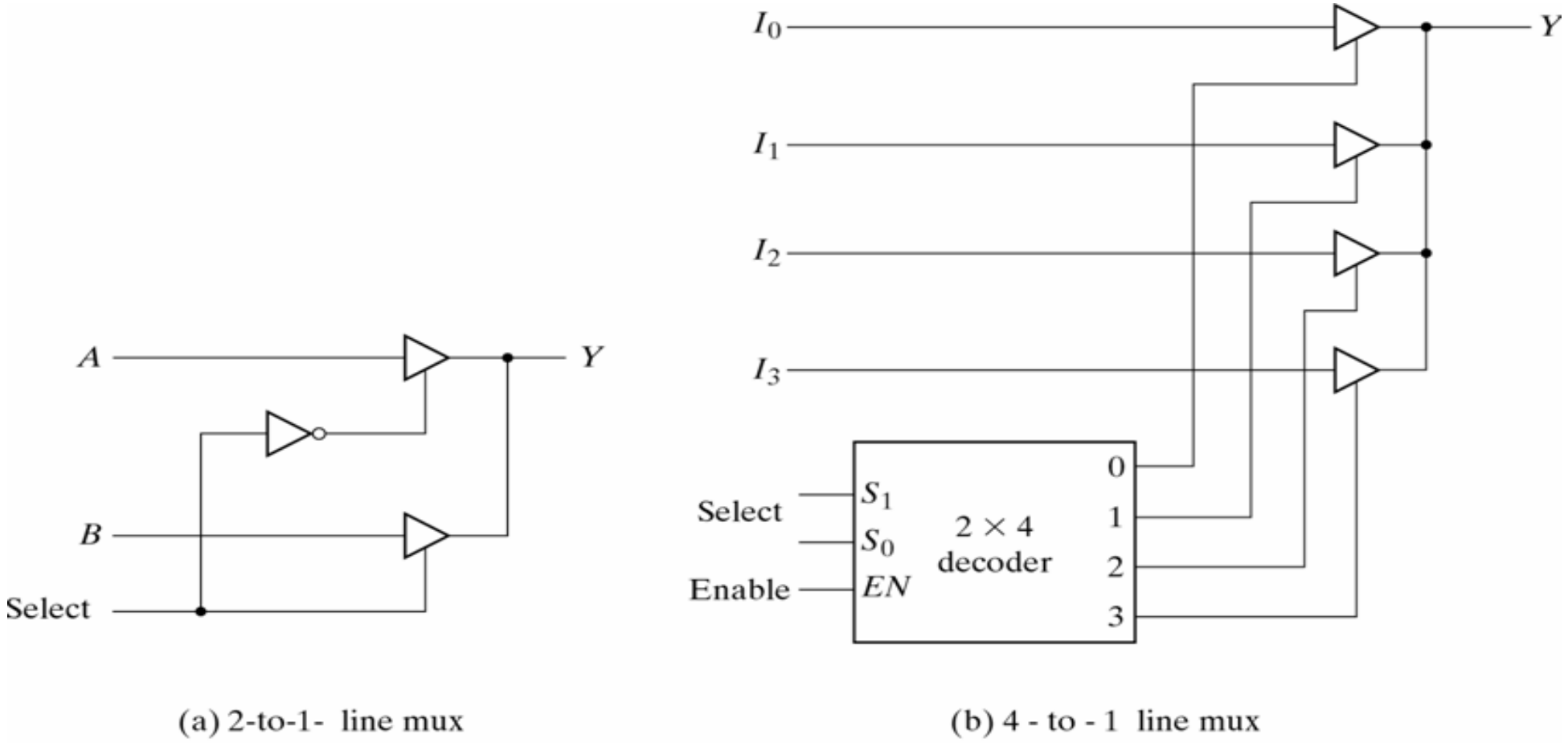
Timing Signals Generation using Counters

- Counters may be used to generate timing signals to control the sequence of operations in a digital system. 2^n timing signals can be generated using an *n*-bit binary counter together with an *n*-to-2^n-line decoder



(b) Counter and decoder

MUX Design using Tri-State gates and Decoders



(a) 2-to-1- line mux

(b) 4 - to - 1 line mux

Fig. 4-30 Multiplexers with Three-State Gates

3-to-8-Line Decoder

- A 3-to-8-Line Decoder is a decoder in which three inputs are decoded into eight outputs, each representing one of the minterms of the three input variables.
- Each one of the eight AND gates generates one of the minterms.
- A particular application of this decoder is binary-to-octal conversion, however 3-to-8-line decoder can be used for decoding any 3-bit code to provide eight outputs, one for each element of the code

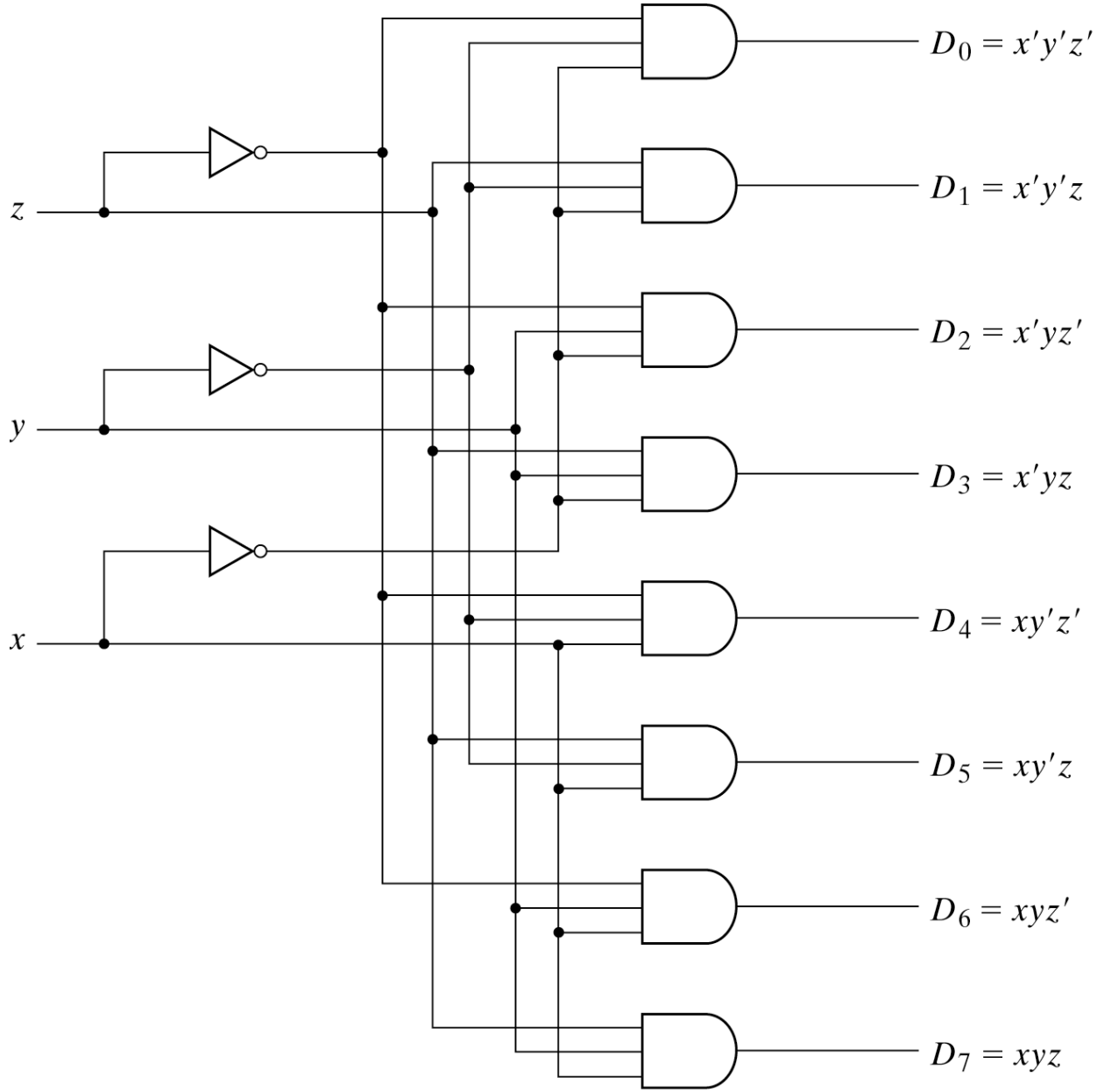


Fig. 4-18 3-to-8-Line Decoder

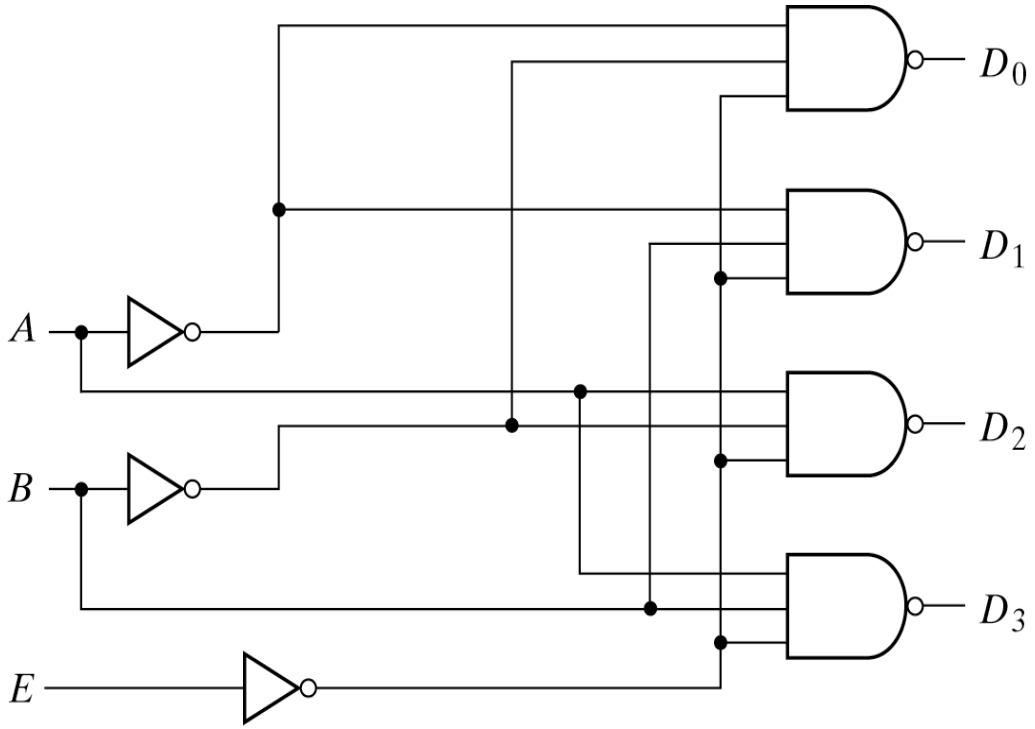
3-to-8-Line Decoder Truth Table

Inputs			Outputs							
X	Y	Z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Decoders with NAND gates

- Some decoders are constructed with NAND gates. Since a NAND gate produces the AND operation with inverted output, it becomes more economical to generate the decoder minterms in their complemented form. Decoder include one or more enable inputs to control the circuit operation
- A 2-to4-line decoder with an enable input is shown next. (see Fig on next slide).
  - The circuit operates with complement outputs and a complement enable input.
  - The decoder is enabled when *E* is equal to 0 and disabled when E = 1
  - The output whose value is equal to 0 represents the minterm selected by inputs *A* and *B*.
  - Only one output can be zero at any given time, all other outputs are 1
- Some decoders have two or more enable inputs that must satisfy a given logic condition

Decoders with NAND gates



(a) Logic diagram

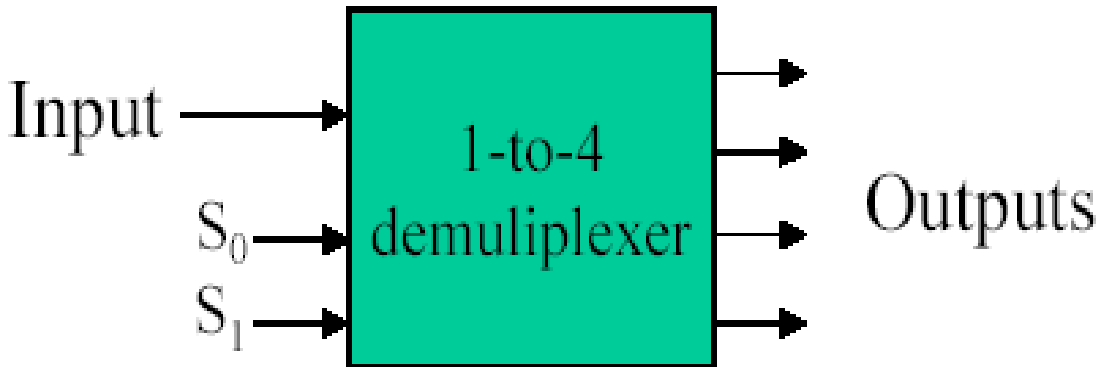
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

Fig. 4-19 2-to-4-Line Decoder with Enable Input

Demultiplexer

- A demultiplexer is a circuit that receives information from a single line and directs it to one of 2^n possible output lines.
- The selection of a specific output is controlled by the bit combination of n selection lines.



Constructing large Decoders

- Decoders with enable inputs can be connected together to form a larger decoder circuit.
  - ➢ two 3-to-8 decoder can be connected to form a 4-to-16 decoder
  - ➢ The top decoder outputs generates minterms 0000 to 0111 and the bottom decoder outputs generate minterms 1000 to 1111.

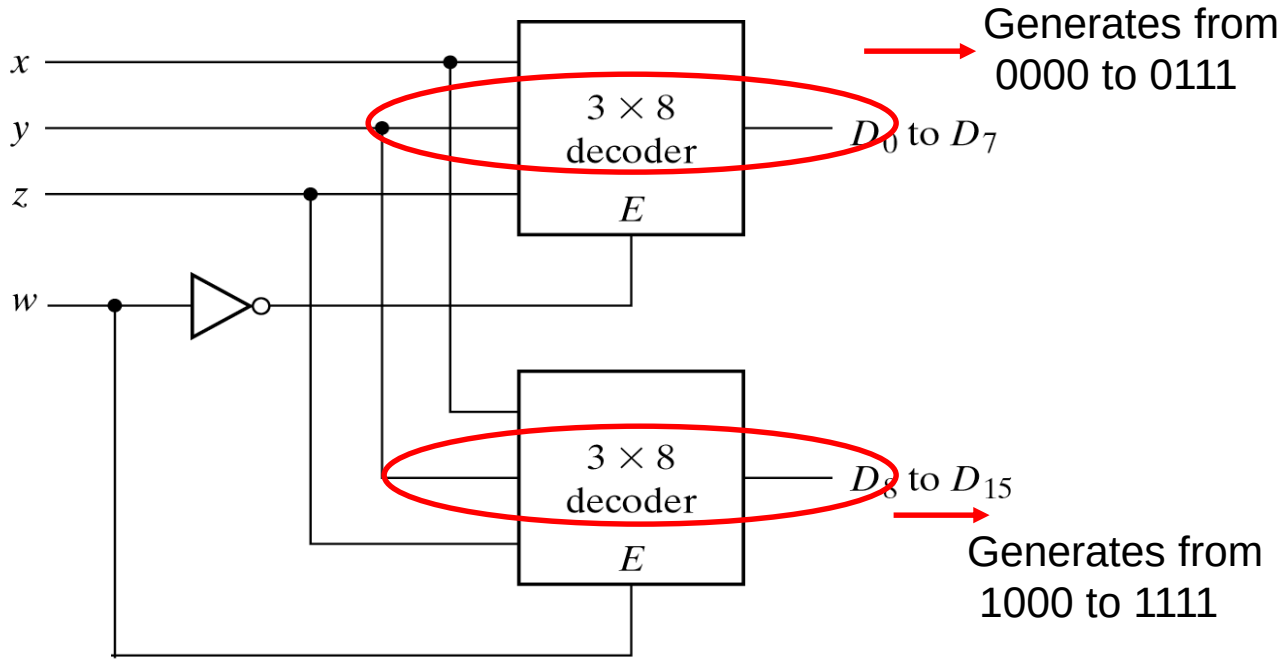


Fig. 4-20 4 × 16 Decoder Constructed with Two 3 × 8 Decoders

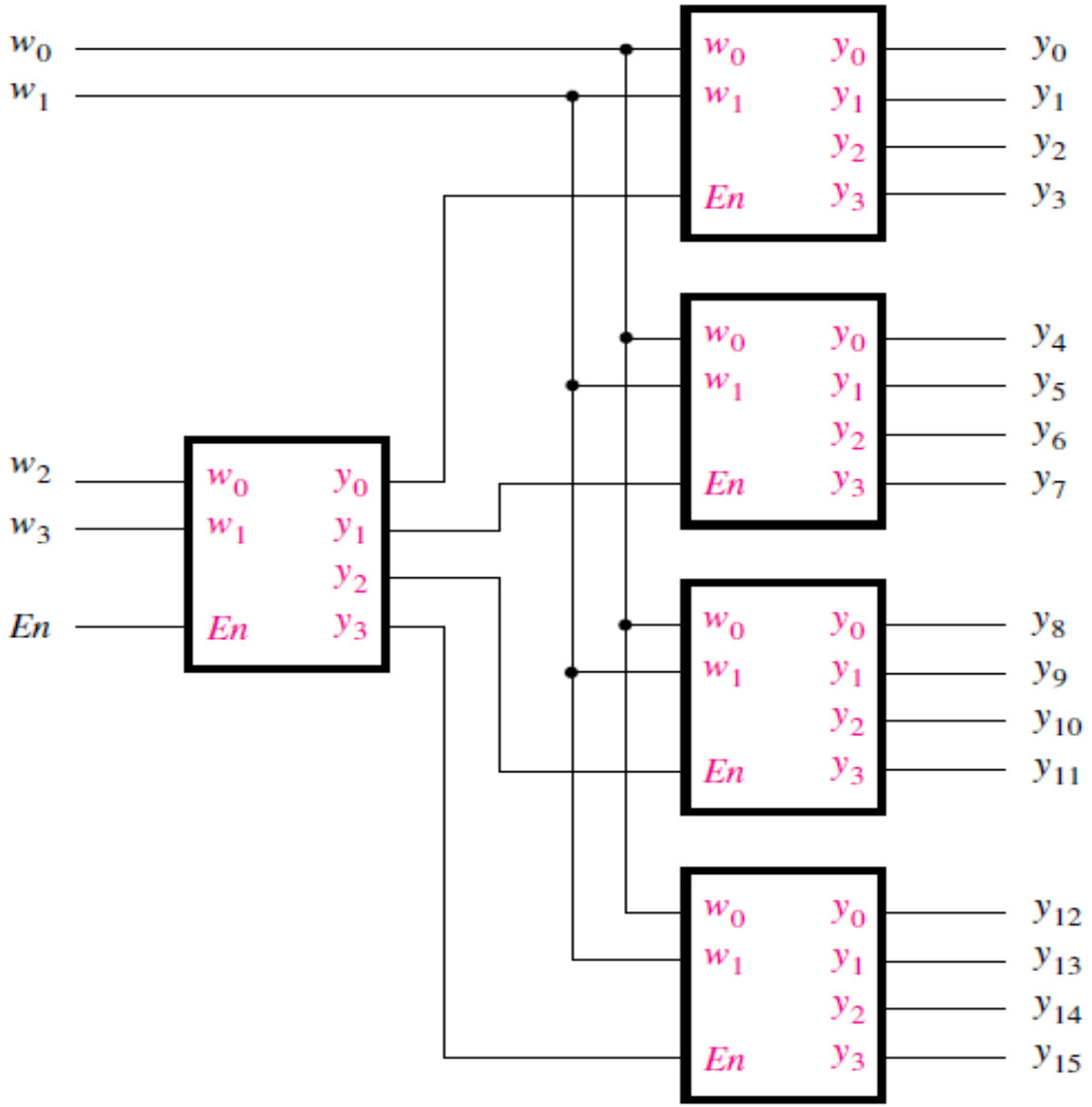


Figure 4.16 A 4-to-16 decoder built using a decoder tree.

Combinational Logic Implementation

- A decoder provides the 2^n minterms of n input variables.
- Any function is can be expressed in sum of minterms.
- Use a decoder to make the minterms and an external OR gate to make the logical sum.
- In this way any combinational circuit with n inputs and m outputs can be implemented with an n-to-2^n line decoder and m OR gates. Such implementation needs that the Boolean function is expressed in sum of minterms
- For example: consider a full adder.
  - $S(x,y,z) = \Sigma(1,2,4,7)$
  - $C(x,y,z) = \Sigma(3,5,6,7)$

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Implementation of Full Adder with a Decoder

- There are three inputs and eight outputs so we need 3-to-8-line decoder
- Two OR gates are required for logical sum of the desired minterms

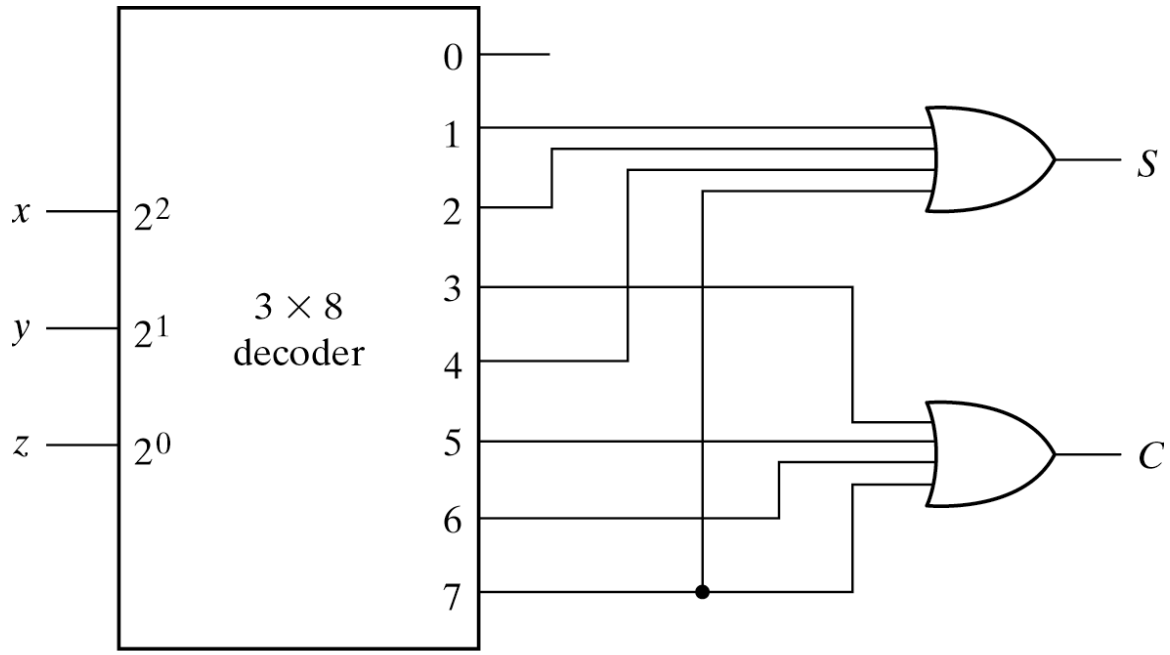


Fig. 4-21 Implementation of a Full Adder with a Decoder

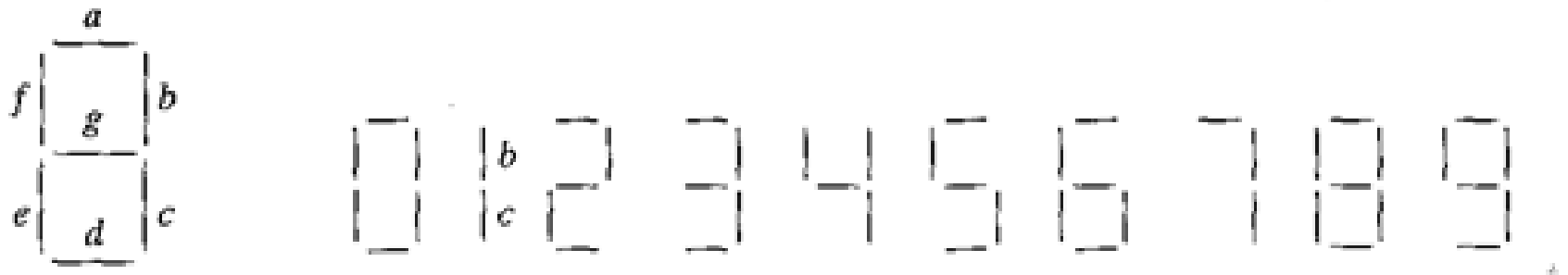
Implementation of Full Adder with a Decoder

- A function with long list of minterms requires an OR gate with large number of inputs
- A function having a list of K minterms can be expressed in its complemented form F′ with 2^n-K minterms
- If the number of minterms in a function is greater than $2^n/2$ then F´ can be expressed with fewer minterms
- In such case it is advantageous to use a NOR gate to sum the minterms of F´. The output of the NOR gate complements this sum and generates the normal output F

Problem Solving Session

Problem: 4-9

A BCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in a display indicator used for displaying the decimal digit in a familiar form. The seven outputs of the decoder (a, b, c, d, e, f, g) select the corresponding segments in the display as shown in Fig. P4-9(a). The numeric display chosen to represent the decimal digit is shown Fig. P4-9(b). Implement your design with a 4:16 line Decoder constructed from two 3:8 line Decoders with enable input, one NOT gate and minimum number of external gates. The six invalid combinations should result in a blank display.



(a) Segment designation

(b) Numerical designation for display

FIGURE P4-9

Solution:

Design procedure:

1. Derive the truth table that defines the required relationship between inputs and outputs.

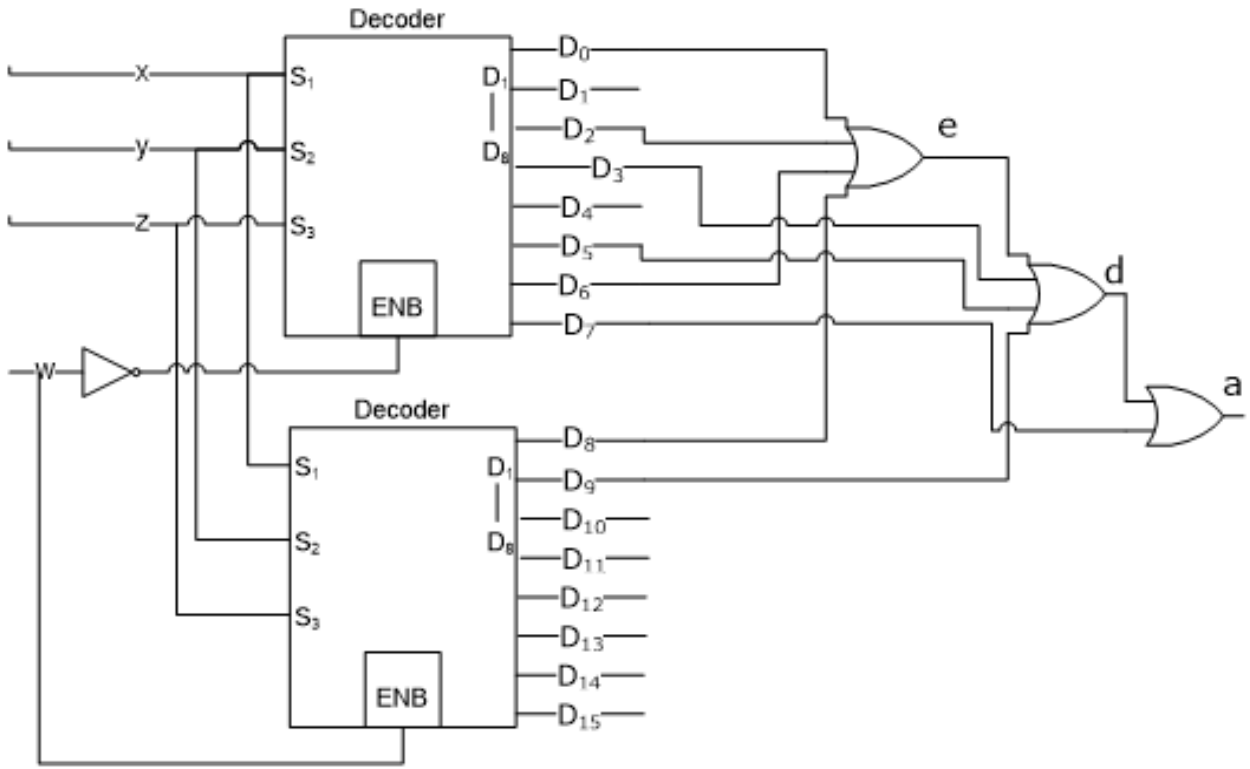
w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	X	0	0	0	0	0	0	0
1	1	X	X	0	0	0	0	0	0	0

2. Express the Boolean expressions for the outputs (a-g) in sum of minterms

$a(w,x,y,z)=\Sigma(0,2,3,5,6,7,8,9)$
$b(w,x,y,z)=\Sigma(0,1,2,3,4,7,8,9)$
$c(w,x,y,z)=\Sigma(0,1,3,4,5,6,7,8,9)$
$d(w,x,y,z)=\Sigma(0,2,3,5,6,8,9)$
$e(w,x,y,z)=\Sigma(0,2,6,8)$
$f(w,x,y,z)=\Sigma(0,4,5,6,8,9)$
$g(w,x,y,z)=\Sigma(2,3,4,5,6,8,9)$

3. Draw the logic circuit. Two 3-to-8-line decoders with enable inputs have been connected to form a 4-to-16-line decoder. Together they generate all the minterms of the input variables. OR gates are to be used to implement each of the functions a-g. The inputs to each OR gate are selected from the decoder outputs according to the list of minterm of each function.

The diagram below shows the circuit for output a, d and e. The same procedure should be followed to include the remaining functions and complete the logic circuit.



Problem: 4-23

Draw the logic diagram of a 2-to-4 line decoder using NOR gates only. Include an enable input.

Solution:

Design procedure:

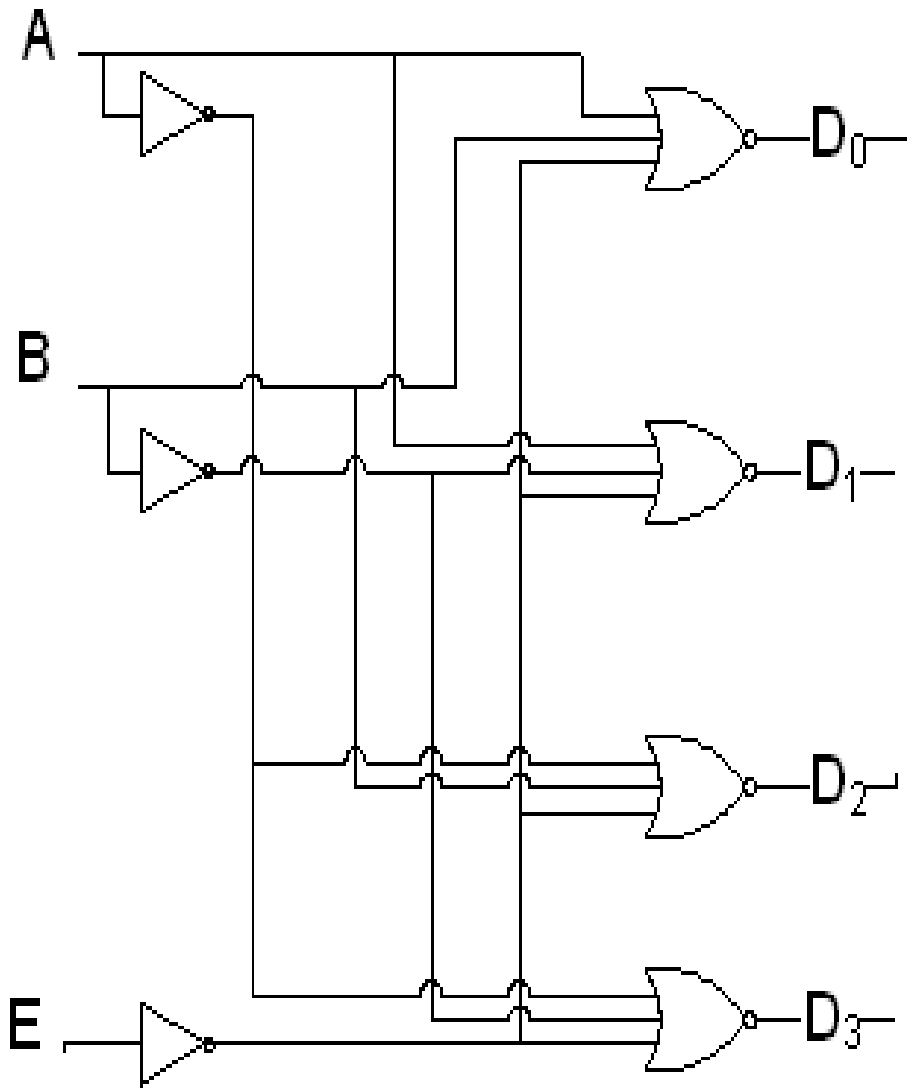
1. The truth table for the circuit.

E	A	B	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$D_0 = EA'B' = (E'+A+B)'$ $\quad$ $D_1 = EA'B = (E'+A+B')'$

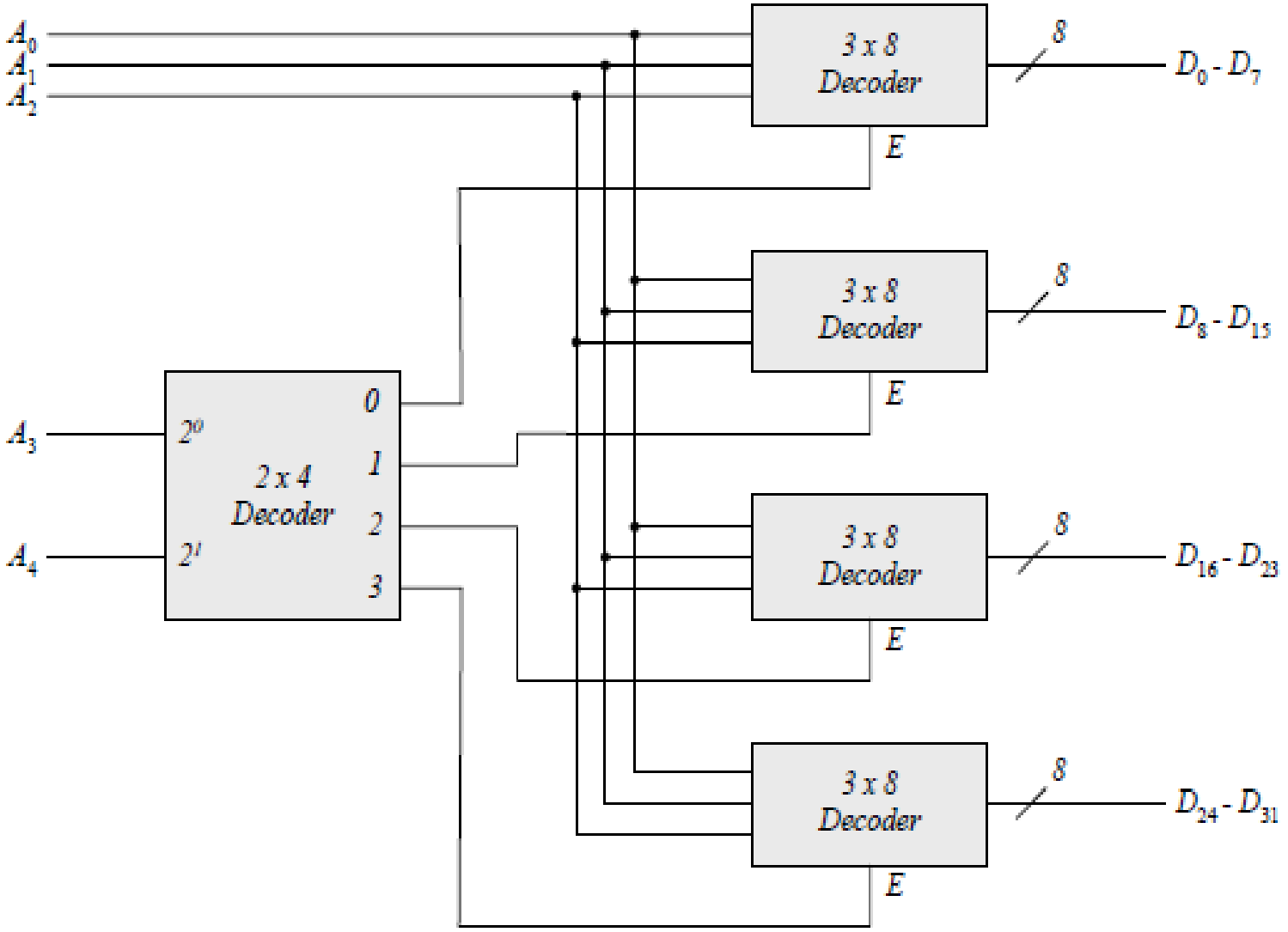
$D_2 = EAB' = (E'+A'+B)'$ $\quad$ $D_3 = EAB = (E'+A'+B')'$

2. The logic diagram



Problem: 4-25

Construct a 5-to-32 line Decoder with four 3-to-8 line Decoders with an enable input and a 2-to-4 line Decoder. Use block diagrams for the components.


A_0
A_1
A_2
A_3
A_4
2 x 4 Decoder
2^0
2^1
0
1
2
3
3 x 8 Decoder
E
8
D_0 - D_7
D_8 - D_{15}
D_{16} - D_{23}
D_{24} - D_{31}

Problem: 4-27

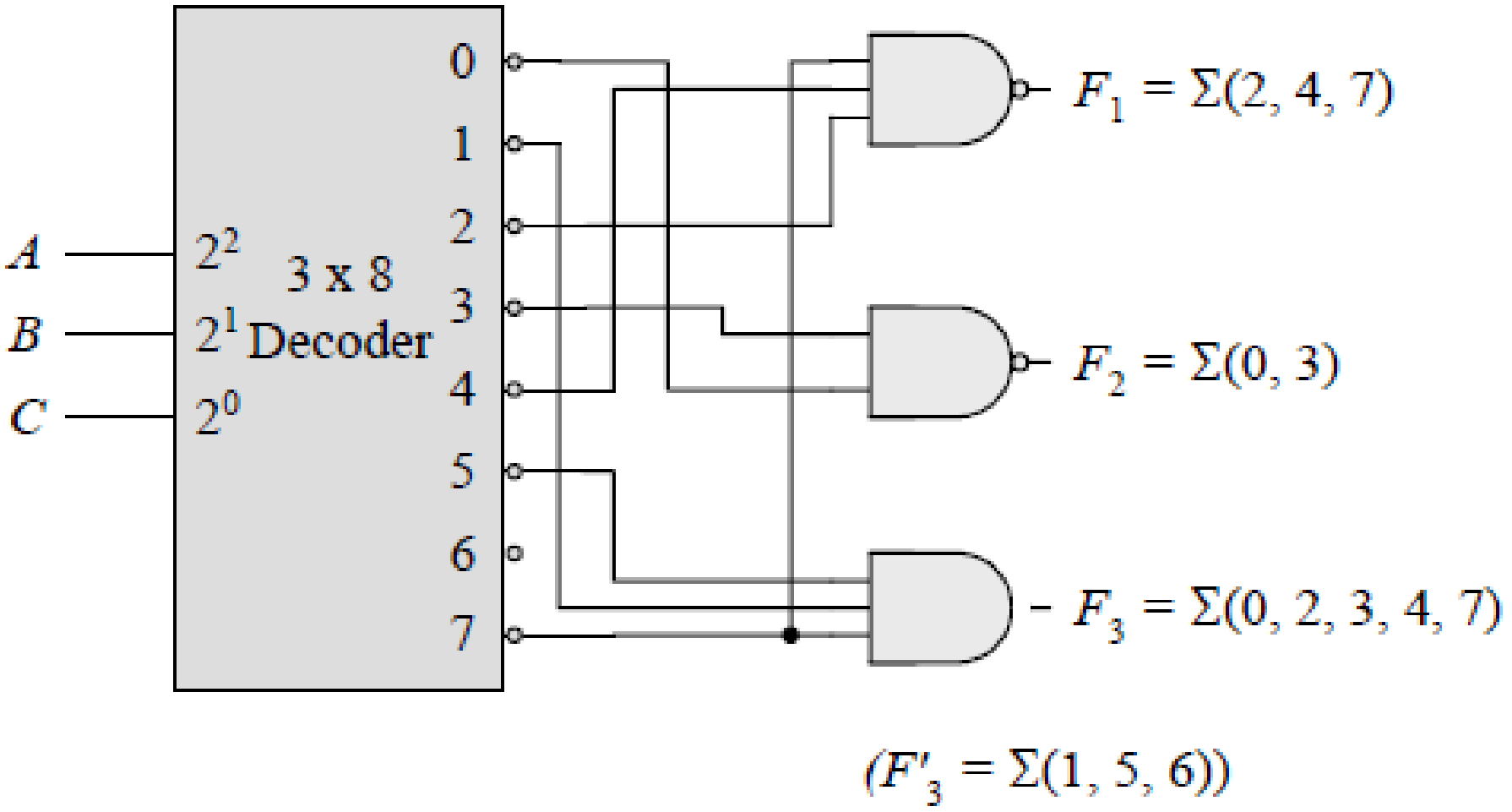
CA combinational circuit is specified by the following three Boolean equations:-

$F_1(A,B,C)=\Sigma(3,5,6)$

$F_2(A,B,C)=\Sigma(1,4,6)$

$F_3(A,B,C)=\Sigma(2,3,5,6,7)$

Implement the circuit with a Decoder constructed with NAND gates and external NAND or AND gates connected to the Decoder outputs. Use a block diagram for the Decoder. Minimize the number of inputs in the external gates.


A
B
C
2^2
2^1
2^0
3 x 8
Decoder
0
1
2
3
4
5
6
7
$F_1 = \Sigma(2, 4, 7)$
$F_2 = \Sigma(0, 3)$
$F_3 = \Sigma(0, 2, 3, 4, 7)$
$(F'_3 = \Sigma(1, 5, 6))$

The End