DECODER & ENCODER

Outline

- Decoder
- Three-to-Eight-Line Decoder
- Two-to-Four-line Decoder with Enable Input
- Combination of decoders
- Adder with decoders
- Encoder
- Octal-to-Binary Encoder
- Priority Encoder

DECODER

Decoder

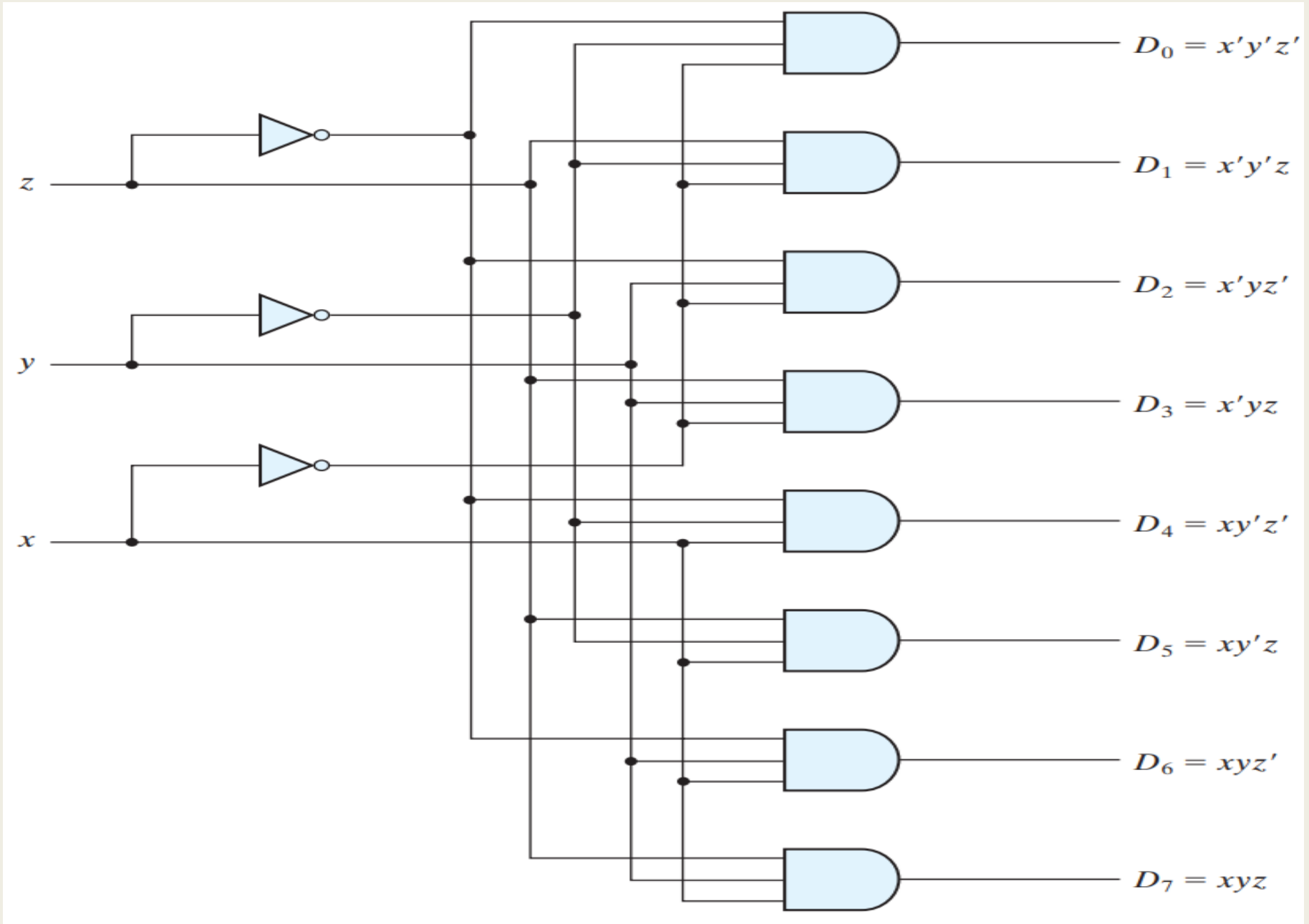
A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

If the n-bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.

Decoder

- Decoders are called *n-to-m-line* decoders, where m <= 2^n
- Purpose is to generate the 2^n (or fewer) minterms of n input variables
- Each combination of inputs will assert a unique output
- The name decoder is also used in conjunction with other code converters, such as a BCD-to-seven-segment decoder

Three-to-Eight-Line Decoder



Three-to-Eight-Line Decoder

- Binary-to-octal conversion
- The input variables represent a binary number
- The outputs represent the eight digits of a number in the octal number system

- In addition, a three-to-eight-line decoder can be used for decoding any three-bit code to provide eight outputs, one for each element of the code

Three-to-Eight-Line Decoder

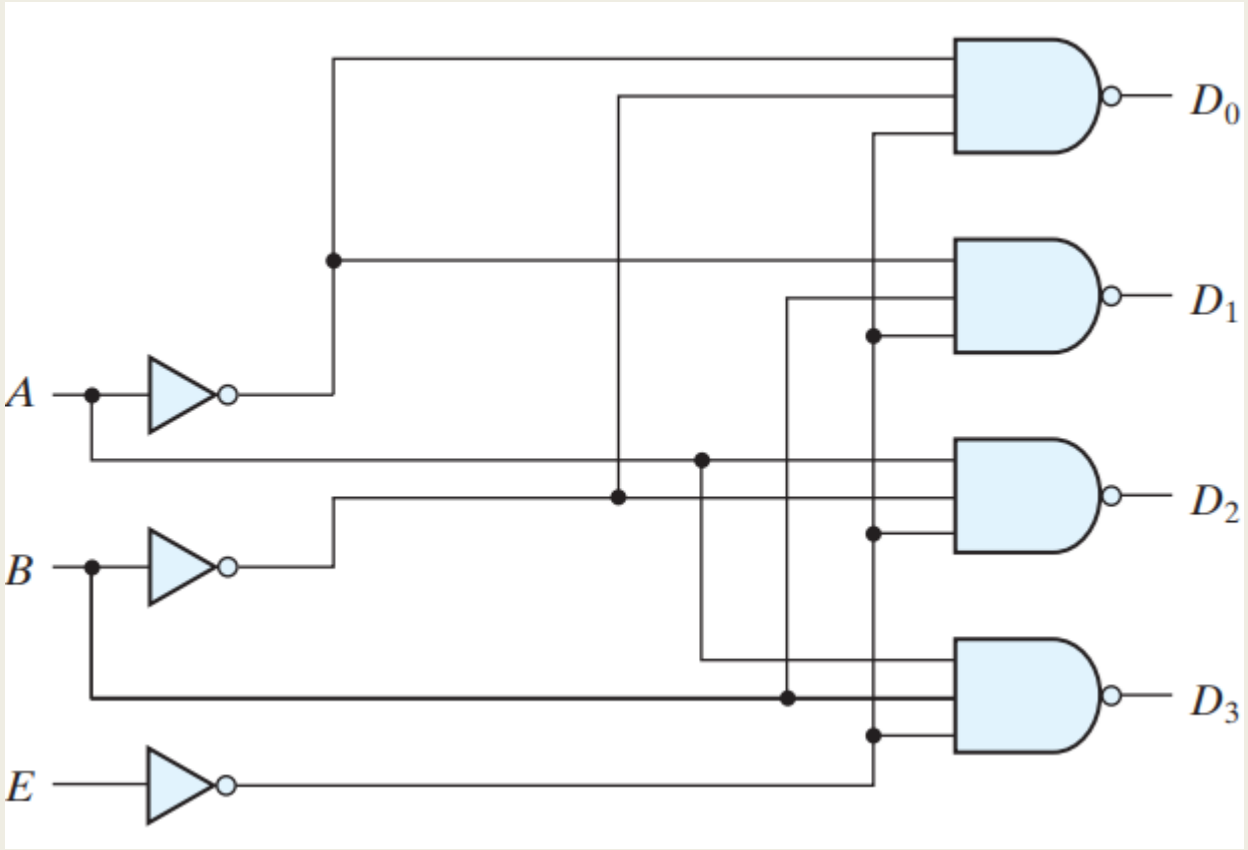
- For each possible input combination, there are seven outputs
  - **that are equal to 0 and**
  - **only one that is equal to 1**
- The output whose value is equal to 1 represents the minterm equivalent of the binary number currently available in the input lines.

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
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

Two-to-four-line decoder with enable input

- Decoders include one or more enable inputs **(E)** to control the circuit operation
- Two-to-four-line decoder with an enable input constructed with NAND gates
- The decoder is enabled when E is equal to 0 **(i.e., active-low enable)**

Two-to-four-line decoder with enable input



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	1

Two-to-four-line decoder with enable input

- **Only one output can be equal to 0 at any given time; all other outputs are equal to 1**
- The output whose value is equal to 0 represents the minterm selected by inputs A and B
- The circuit is disabled when E is equal to 1, regardless of the values of the other two inputs

Combination of decoders

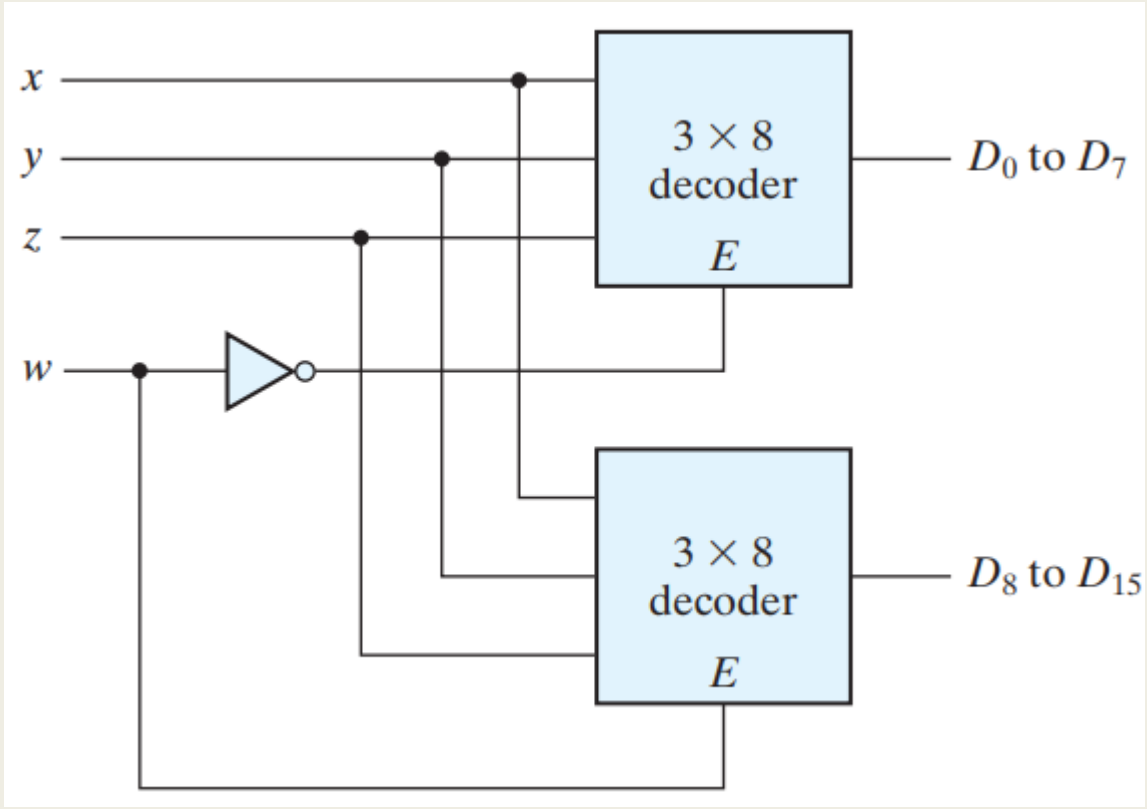
- Decoders with enable inputs can be connected together to form a larger decoder circuit.
- **Two 3-to-8-line decoders** with enable inputs connected to form **a 4-to-16-line decoder**.

Case 1:

When w=0, the top decoder is enabled and the other is disabled.
The bottom decoder outputs are all 0's, and the top eight outputs generate minterms 0000 to 0111.

Case 2 (together with case 1):

When w=1, the enable conditions are reversed.
The bottom decoder outputs generate minterms 1000 to 1111, while the outputs of the top decoder are all 0's.



Adder with decoder

- Decoder **(with some OR gates)** can be used to implement any combinational circuit, for example a Full Adder
- Truth table of Full Adder, we get

Inputs			Outputs	
X	**Y**	**Z**	**S**	**C**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The Canonical SOP of the output functions:

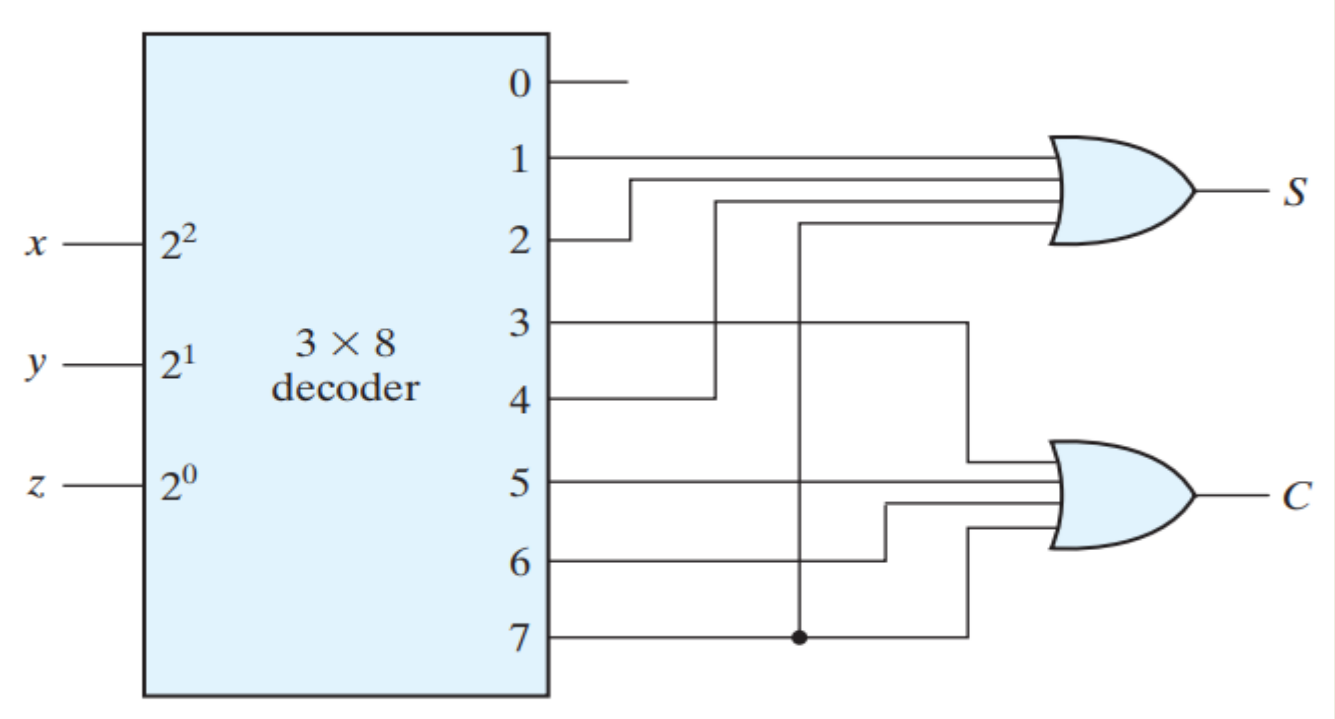
$$S(X, Y, Z) = \sum(1, 2, 4, 7)$$

$$C(X, Y, Z) = \sum(3, 5, 6, 7)$$

Adder with decoder

- **three inputs and a total of eight minterms -------- a three-to-eight-line decoder**

- The decoder generates the eight minterms for x, y, and z
- The OR gate for output S forms the logical sum of minterms 1, 2, 4, and 7
- The OR gate for output C forms the logical sum of minterms 3, 5, 6, and 7



ENCODER

Encoder

- Encoder is a digital circuit that performs the inverse operation of a decoder
- An encoder has 2^n (or fewer) input lines and **n** output lines.
- The output lines, as an aggregate, generate the binary code corresponding to the input value

- **Example of an encoder :** the octal-to-binary encoder

Octal-to-Binary Encoder

- Eight inputs (one for each of the octal digits) and three outputs

- Outputs generate the corresponding binary number
- The encoder can be implemented with OR gates whose inputs are determined directly from the truth table
- **Main constraint:** only one input has a value of 1 at any given time.

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Boolean Output functions:

$$z = D_1+D_3+D_5+D_7$$

$$y = D_2+D_3+D_6+D_7$$

$$x = D_4+D_5+D_6+D_7$$

Octal-to-Binary Encoder

Limitation:

- **only one input can be active at any given time**
- **If two inputs are active simultaneously, the output produces an undefined combination.**
- **Example:** if D3 and D6 are 1 simultaneously, the output of the encoder will be 111 because all three outputs are equal to 1. The output 111 does not represent either binary 3 or binary 6. **[PROBLEM !!!]**

- **SOLUTION:** encoder circuits must establish **an input priority** to ensure that only one input is encoded: **higher priority for inputs with higher subscript numbers**
- **Example:** if both D3 and D6 are 1 at the same time, the output will be 110 because D6 has higher priority than D3

Priority Encoder

- A priority encoder is an encoder circuit that includes the priority function.
- The operation of the priority encoder is such that if two or more inputs are equal to 1 at the same time, **the input having the highest priority will be considered**

Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Truth table of a four-input priority encoder

- With two outputs, x and y, a third output **V (Valid bit indicator output)**
  - **V = 1,** when one or more inputs are equal to 1
  - If all inputs are 0, there is no valid input and **V=0**
- The other two outputs are not inspected when **V=0** are specified as the **don't-care Conditions.**
- **Don't Care Conditions** (Xs) in inputs are useful:
  - Instead of listing all 16 minterms of four variables, the truth table uses an X to represent either 1 or 0.
  - Example, X100 represents the two minterms 0100 and 1100.

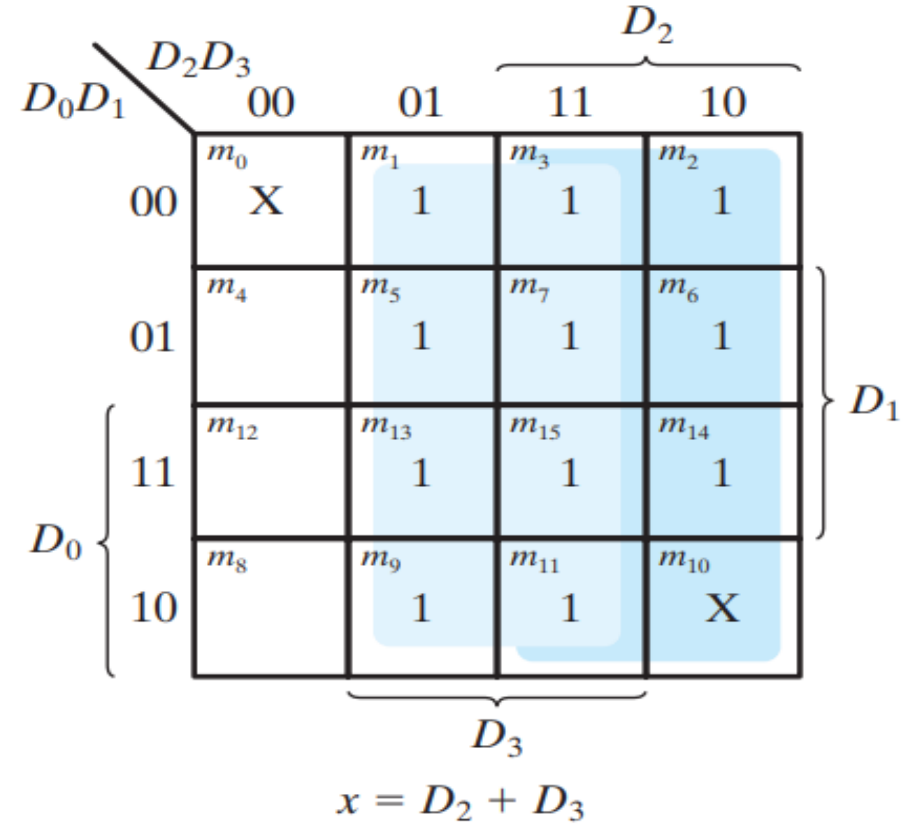
Priority Encoder

Boolean Output functions:

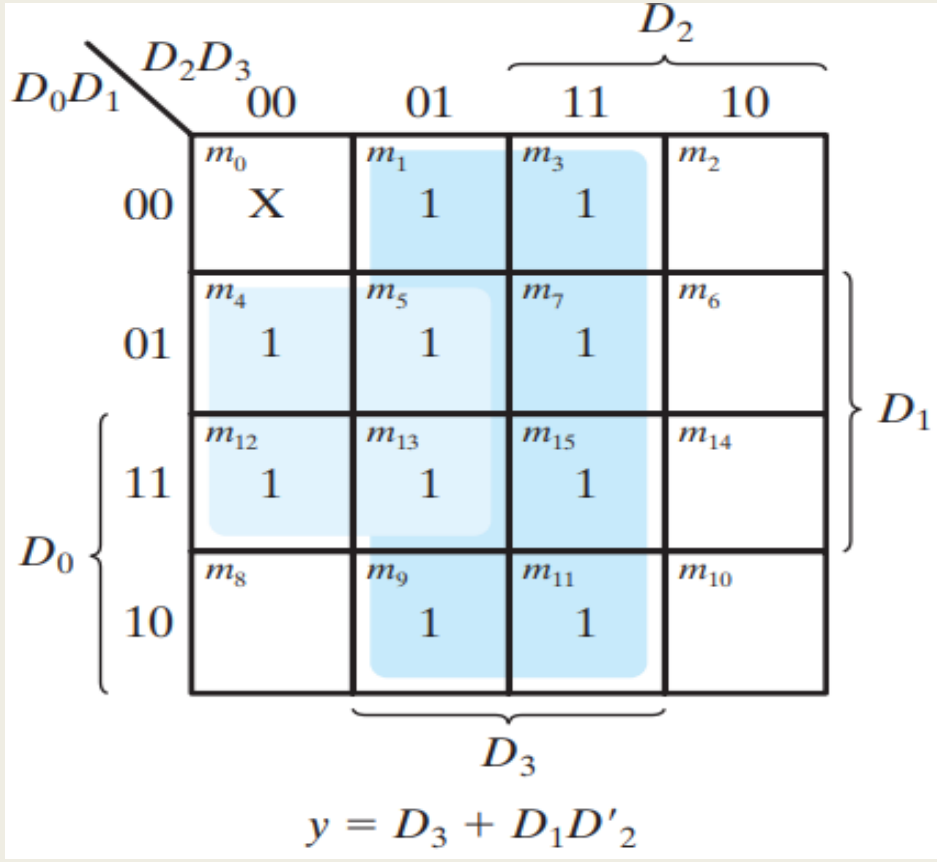
$x = D_2 + D_3$

$y = D_3 + D_1 D'_2$

$V = D_0 + D_1 + D_2 + D_3$

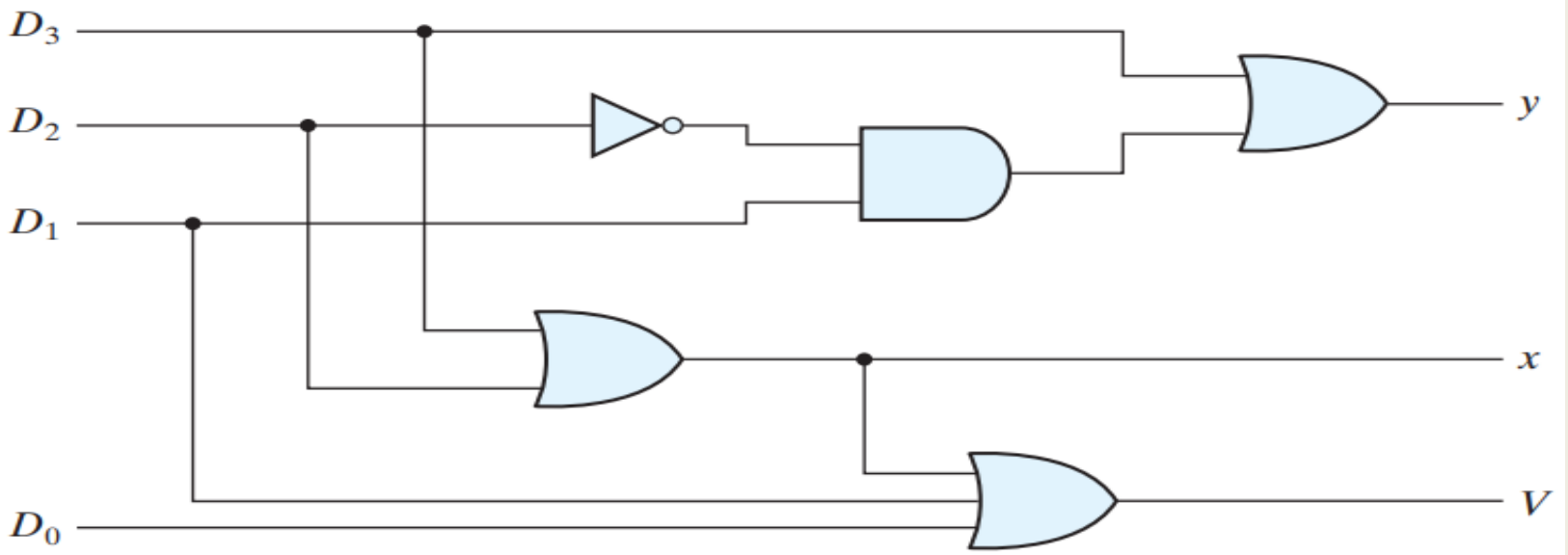


$x = D_2 + D_3$



$y = D_3 + D_1 D'_2$

Priority Encoder



Logic Diagram of a four-input priority encoder