Basic Electronics Engineering (Spring 2024)

Resources of PPT:

- www.google.com

- Digital Design, 4th Edition
  M. Morris Mano and Michael D. Ciletti

Combinatorial Circuit

Digital Circuits are of two types:

- Combinatorial: The combinational circuit is time-independent. The output it generates does not depend on any of its previous inputs.
- Sequential: Sequential circuits are the ones that depend on clock cycles. They depend entirely on the past as well as the present inputs for generating output.

What is a Combinational Circuit?

- The output of a Combinational Circuit depends entirely on the present input.
- It exhibits a faster speed.
- It is comparatively easier to design.
- No feedback is present between the input and output.
- The combinational circuit depends on time.
- Logic gates form the building blocks of such circuits.
- One can make use of it for both boolean and arithmetic operations.
- They don't hold the capacity of storing any state.
- These circuits do not have a clock- thus, they don't require triggering.
- They do not possess any memory element.
- Users can feasibly use as well as handle them.
- Example – Demultiplexer, Multiplexer, Decoder, Encoder, etc.

Combinatorial Circuit

What is a Sequential Circuit?

- The output of a Sequential Circuit depends on both- past as well as present inputs.
- It works at a comparatively slower speed.
- The design of these circuits is comparatively much tougher than the Combinational Circuit.
- A feedback path exists between the output and the input.
- The circuit is time-dependent.
- Flip-flops constitute the building blocks of such a circuit.
- People mainly use them for storing data and information.
- They possess the capability of storing any data state or retaining an earlier state at any given point.
- Because a Sequential circuit depends on a clock, it usually requires triggering.
- They always possess a memory element.
- A user may not be able to handle and use these circuits easily.
- For Example – Counters, Flip-flops, etc.

Combinatorial Circuit

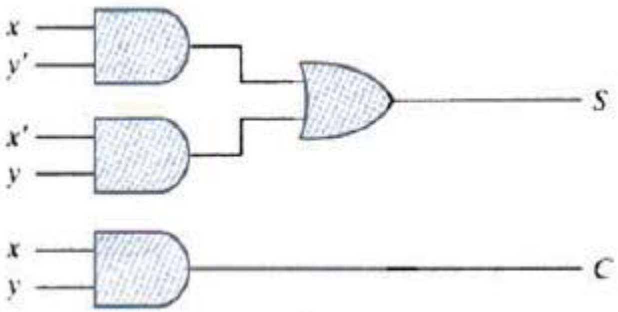
Half Adder

Half Adder

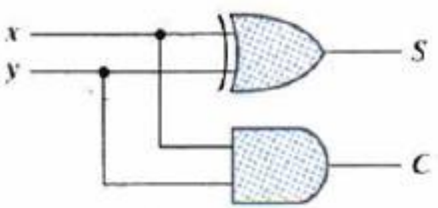
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The simplified Boolean functions for the two outputs can be obtained directly from the truth table. The simplified sum-of-products expressions are

$$S = x'y + xy'$$
$$C = xy$$



(a) $S = xy' + x'y$
$C = xy$



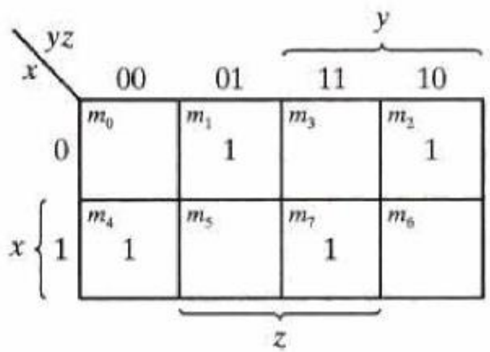
(b) $S = x \oplus y$
$C = xy$

Combinatorial Circuit

Full Adder

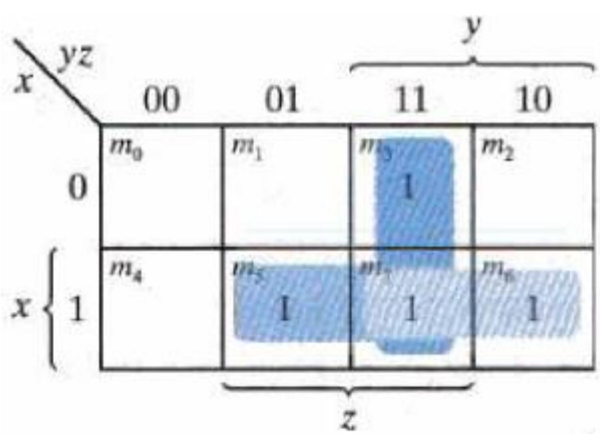
Full Adder

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



Truth Table for Sum S

$$S = x'y'z + x'yz' + xy'z' + xyz$$



Truth Table for Carry C

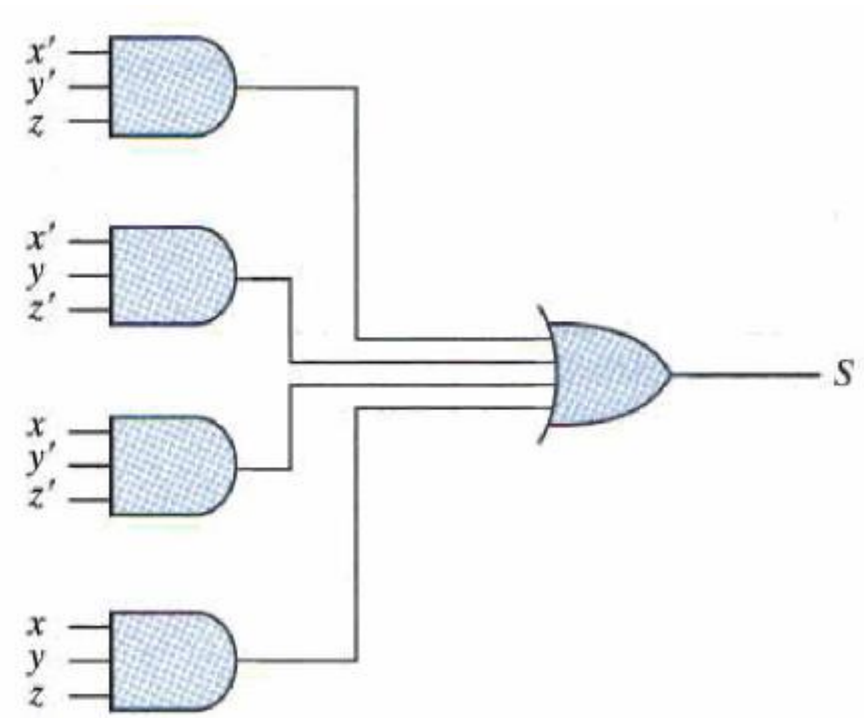
$$C = xy + xz + yz$$

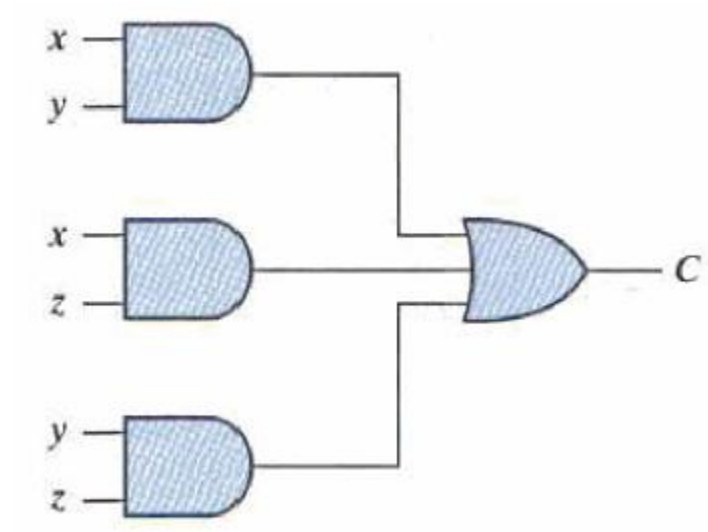
Combinatorial Circuit

Full Adder

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = xy + xz + yz$$



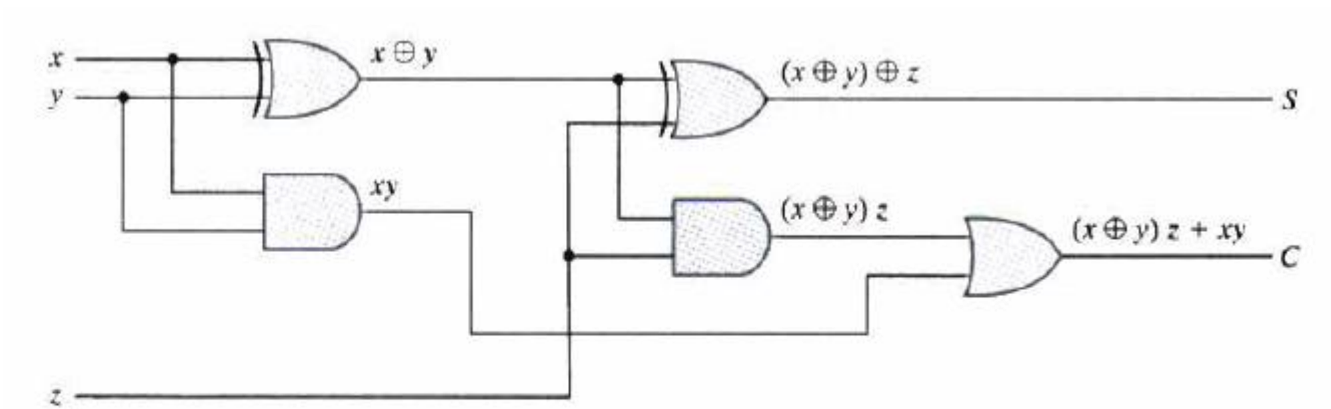


Combinatorial Circuit

Full Adder

$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$= z'(xy' + x'y) + z(xy + x'y')$$
$$= z'(xy' + x'y) + z(xy' + x'y)'$$
$$= z \oplus (x \oplus y)$$

$$C = xy + xz + yz$$
$$= xy + z(x + y)$$
$$= xy + z(x[y + y'] + y[x + x'])$$
$$= xy + z(xy + xy' + xy + x'y)$$
$$= xy + xyz + z(xy' + x'y)$$
$$= xy(1 + z) + z(xy' + x'y)$$
$$= xy + z(x \oplus y)$$



Half Subtractors

Half Subtractors are a type of digital circuit that calculates the arithmetic binary subtraction between two single-bit numbers. It is a circuit with two inputs and two outputs.

For two single-bit binary numbers A and B, a half subtractor produces two outputs.

- **A** is known as the Minuend Bit
- **B** is called Subtrahend Bit.
- **Output D** is the difference between the two input bits (A-B).
- **Output P** is the previous borrow between the two input bits (A-B).

The previous borrow is for the most significant bit (MSB).

Combinatorial Circuit : Half Subtractor

Case 1: A = 0, B = 0;

According to Binary subtraction, the difference of these numbers is 0 with no previous borrow.

0

− 0

______

0

______

Hence, **D= 0, P= 0**

Case 2: A= 0, B= 1;

According to Binary subtraction, the difference between these two numbers is 1 with a previous borrow of 1.

0

− 1

→1 (Previous Borrow)

______

1

______

Hence, **D= 1, P= 1**

Combinatorial Circuit : Half Subtractor

Case 3: A= 1, B= 0;

As per Binary subtraction, the difference between these two numbers is 1 with no previous borrow.

1

− 0

———————

1

———————

Hence, **D= 1, P= 0**

Case 4: A= 1, B= 1;

According to Binary subtraction, the difference between these two numbers is 1 with no previous borrow.

1

− 1

———————

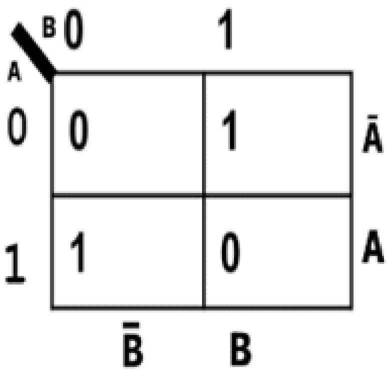
0

———————

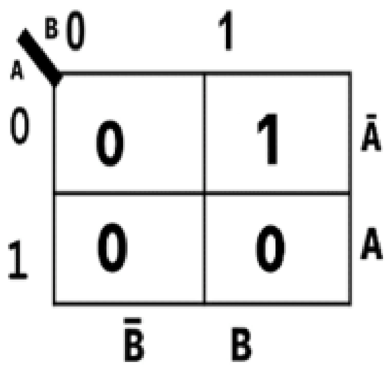
Hence, **D= 0, P= 0**

Combinatorial Circuit: Half Subtractor

A	B	Difference (S)	Previous Borrow (P)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



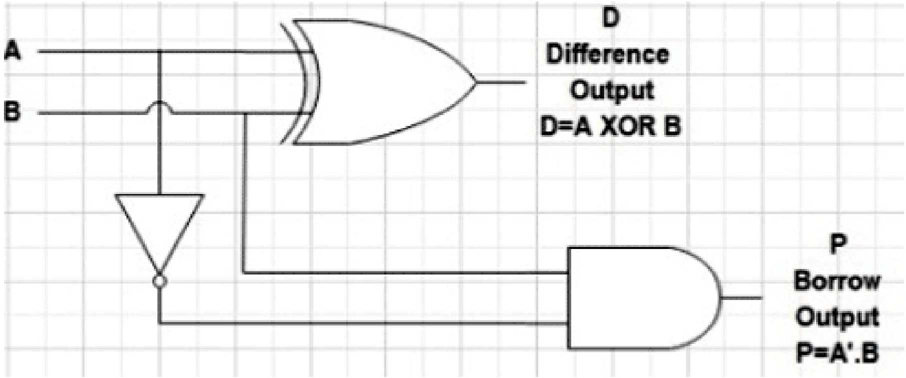
Half Subtractor Karnaugh Map for Difference



Half Subtractor K-Map for Borrow

D= A′B + AB′

P= A′•B



Combinatorial Circuit: Full Subtractor

For three single-bit binary numbers A, B, and P, the full subtractor circuit generates two single-bit binary outputs D (Difference), and Q (Borrow Output).

- **A** is the Minuend.
- **B** is the Subtrahend.
- **P** is the Previous Borrow Bit.
- **D** is the Difference between A, B, and P.
- **Q** is the Borrow Output.

Combinatorial Circuit: Full Subtractor

Case 1: A= 0, B= 0, and P= 0;

The difference of the three binary numbers 0, 0, and 0 produces a difference of 0 and generates no borrow output.

0

-0

-0

———————

0

———————

Hence, **D= 0, Q= 0**

Case 2: A= 0, B= 0, and P= 1;

The difference of the three binary numbers 0, 0, and 1 produces a difference of 1 and generates a borrow output bit.

0

-0

-1

→1

——————

1

——————

Hence, **D= 1, Q= 1**

Combinatorial Circuit: Full Subtractor

Case 3: A= 0, B= 1, and P= 0;

The difference of the three binary numbers 0, 1, and 0 produces a difference of 1 and generates a borrow output bit.

0

-1

→1

-0

———————

1

———————

Hence, **D= 1, Q= 0**

Case 4: A= 0, B= 1, and P= 1;

The difference of the three binary numbers 0, 1, and 1 produces a difference of 0 and generates a borrow output bit.

0

-1

→1

-1

——————

0

——————

Hence, **D= 0, Q= 1**

Combinatorial Circuit: Full Subtractor

Case 5: A= 1, B= 0, and P= 0;

The difference of the three binary numbers 1, 0, and 0 produces a difference of 1 and generates no borrow output.

1

-0

-0

———–

0

———–

Hence, **D= 1, Q= 0**

Case 6: A= 1, B= 0, and P= 1;

The difference of the three binary numbers 1, 0, and 1 produces a difference of 0 and generates no borrow output.

1

-0

-1

————

0

————

Hence, **D= 0, Q= 0**

Combinatorial Circuit: Full Subtractor

Case 7: A= 1, B= 1, and P= 0;

The difference between the three binary numbers 1, 1, and 0 produces a difference of 0 and generates no borrow output.

1

-0

-0

——————

0

——————

Hence, **D= 0, Q= 0**

Case 8: A= 1, B= 1, and P= 1;

The difference between the three binary numbers 1, 1, and 1 produces a difference of 1 and generates a borrow output bit.

The difference between A=1 and B=1 is 0. P=1 gets subtracted from this 0 to produce an output of 1 by generating a borrowed output bit.

1

-1

-1

→1

—————

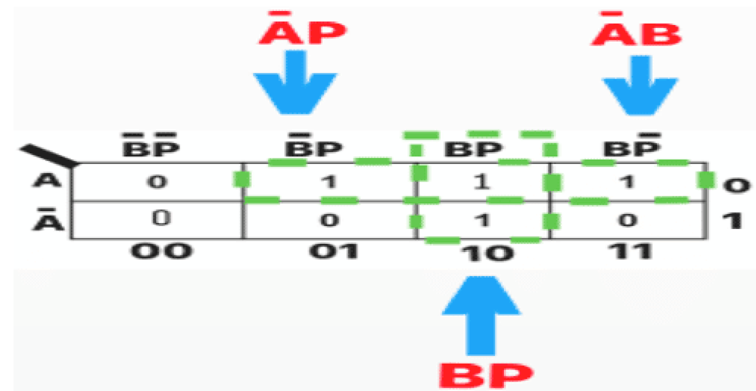
1

—————

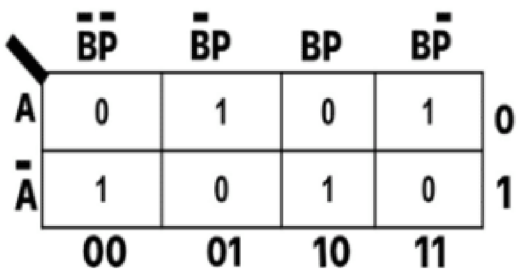
Hence, **D= 1, Q= 1**

Combinatorial Circuit: Full Subtractor

A	B	P	Difference (D)D= A-B-P	Borrow Output (Q)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Full Subtractor karnaugh Map for Borrow



Full Subtractor karnaugh Map for Difference

$$Q = \bar{A}P + \bar{A}B + BP$$

$$D = (\bar{A}\,\bar{B}\,P) + (\bar{A}\,B\,\bar{P}) + (A\,\bar{B}\,\bar{P}) + (A\,B\,P)$$

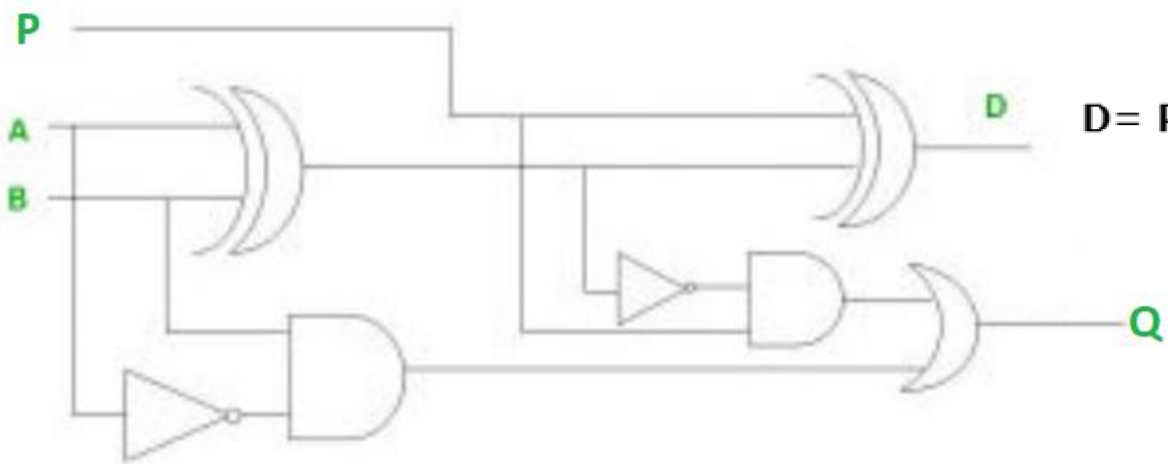
$$D = P\,(\bar{A}\,\bar{B} + A\,B) + \bar{P}\,(\bar{A}\,B + A\,\bar{B})$$

$$D = P\,\overline{(\bar{A}\,B + A\,\bar{B})} + \bar{P}\,(\bar{A}\,B + A\,\bar{B})$$

$$D = P \oplus (\bar{A}\,B + A\,\bar{B})$$

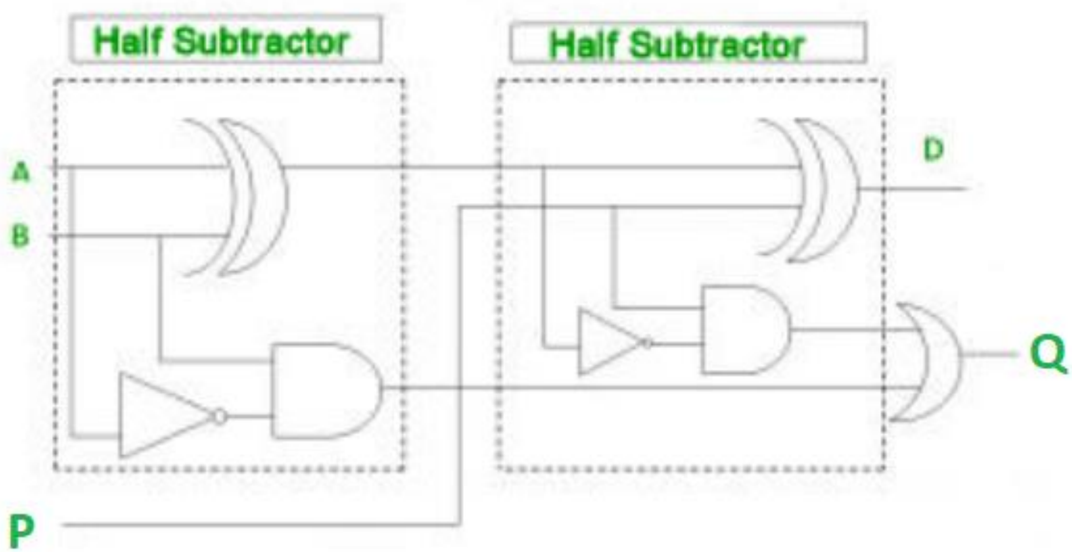
$$D = P \oplus A \oplus B$$

Combinatorial Circuit: Full Subtractor



$D = P \oplus A \oplus B$

$Q = \bar{A}P + \bar{A}B + BP$



Combinatorial Circuit

DECODERS

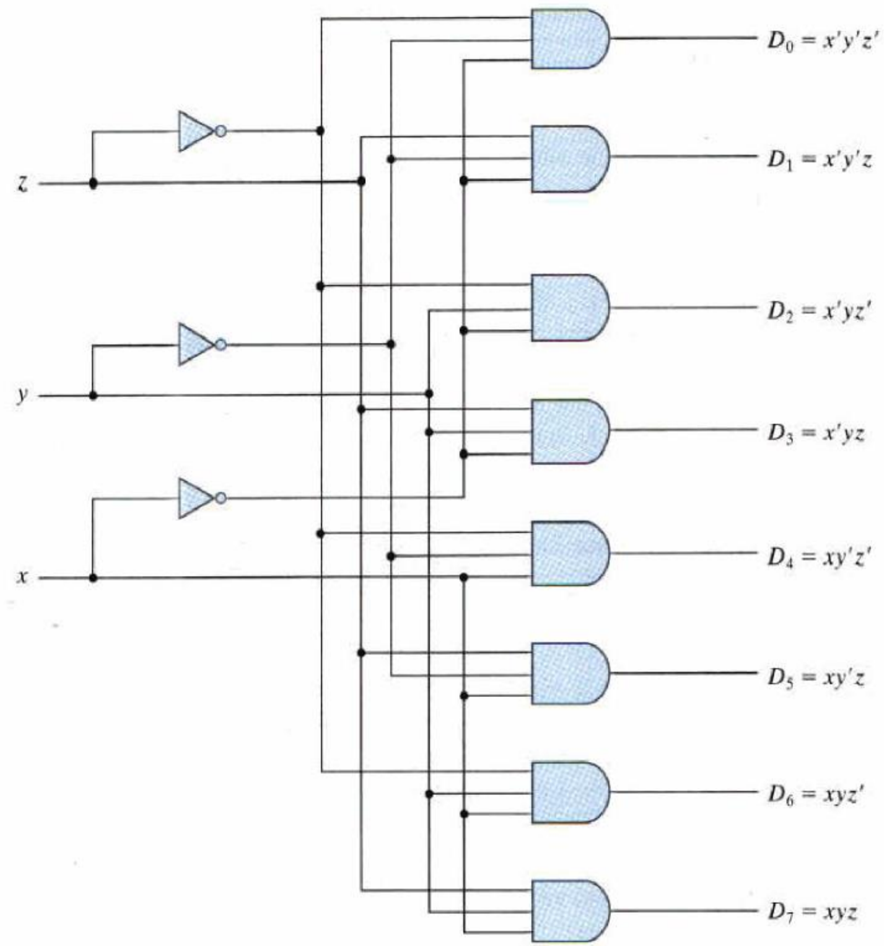
Discrete quantities of information are represented in digital systems by binary codes. A binary code of n bits is capable of representing up to 2^n distinct elements of coded information. A *decoder* is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. If the n-bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.

Truth Table of a Three-to-Eight-Line Decoder

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

Combinatorial Circuit

DECODERS



Combinatorial Circuit

ENCODERS

Truth Table of an Octal-to-Binary Encoder

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

$$z = D_1 + D_3 + D_5 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$x = D_4 + D_5 + D_6 + D_7$$