Digital Electronics

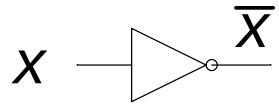
Dr. Md. Hasnat Kabir
Associate Professor
Dept. of Information and Communication Eng.

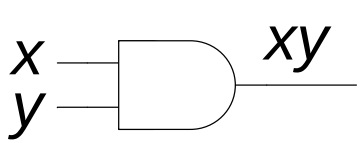
Review of Boolean algebra

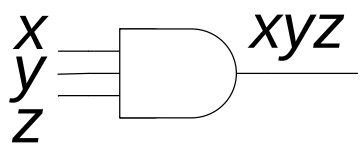
- Just like Boolean logic
- Variables can only be 1 or 0
  - Instead of true / false

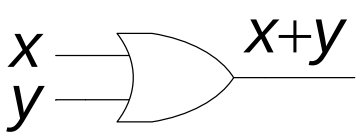
ict.up.ac.th/.../com_ar/ppt/04-logic-gates.ppt

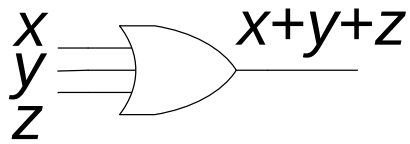
Basic logic gates

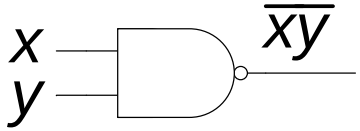
- Not 

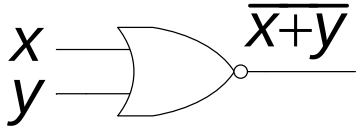
- And 

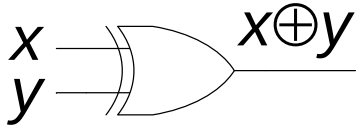


- Or 

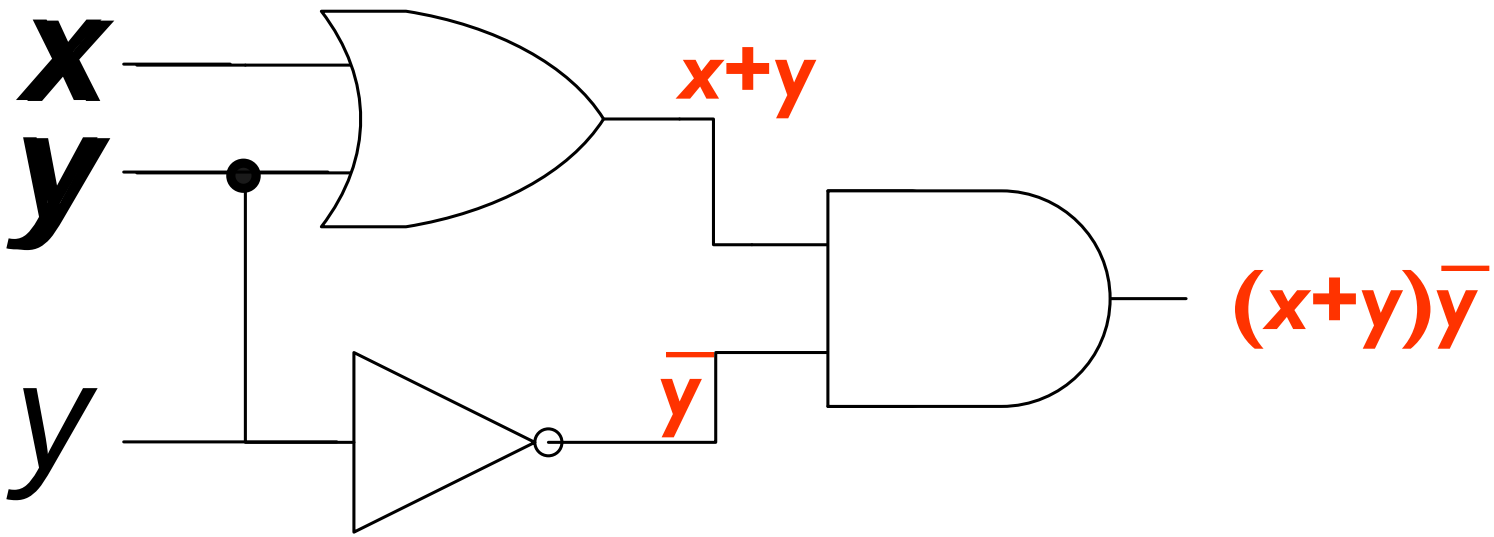


- Nand 

- Nor 

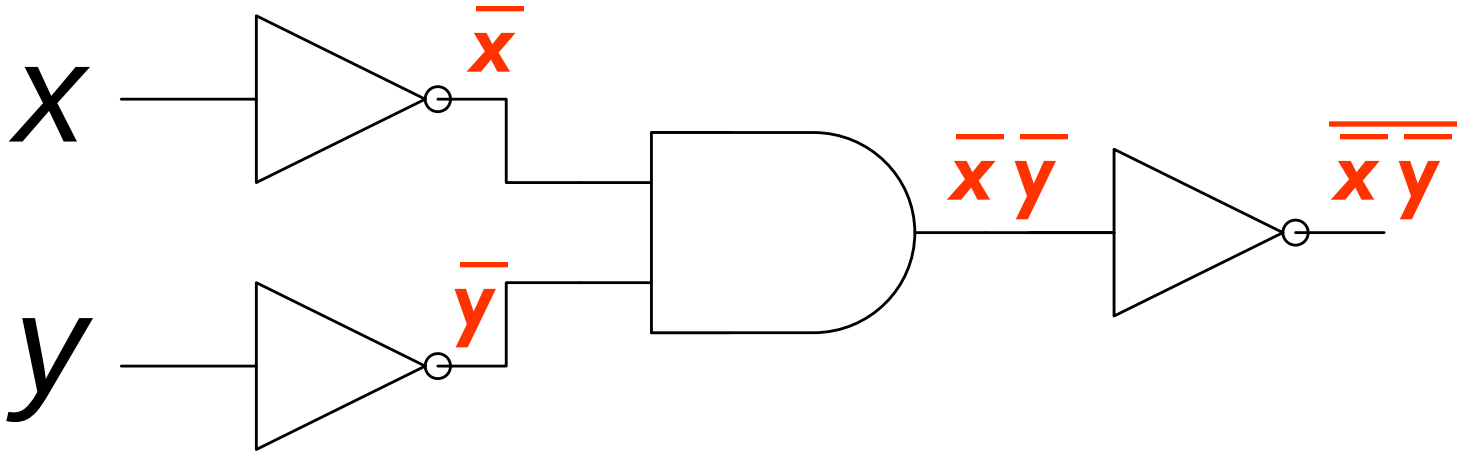
- Xor 

- Find the output of the following circuit



- Answer: $(x+y)\overline{y}$

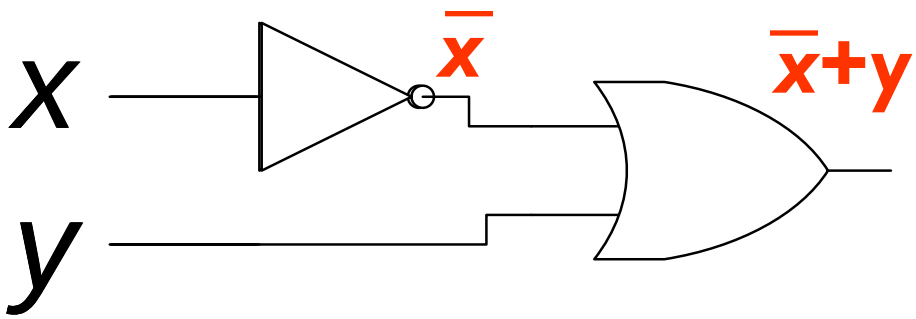
- Find the output of the following circuit



- Answer: $\overline{\overline{x}\,\overline{y}}$

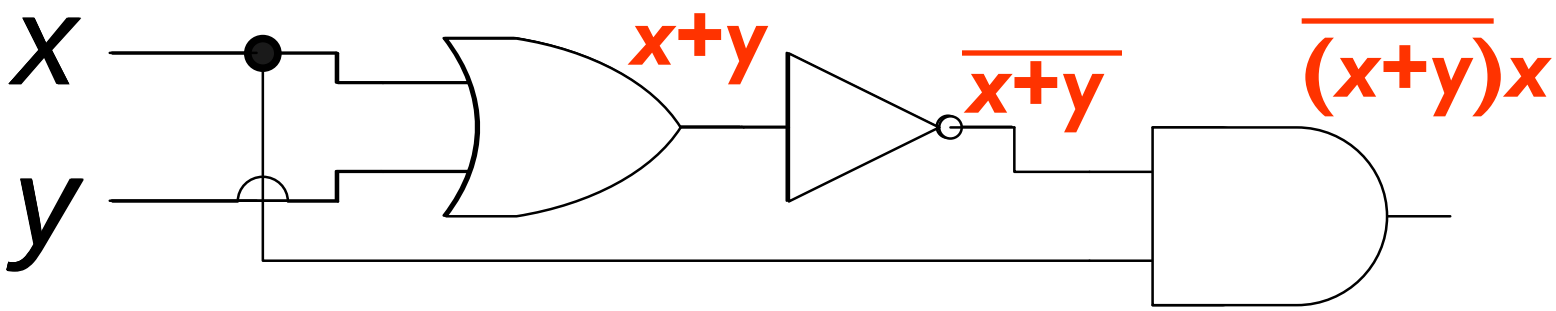
- Write the circuits for the following Boolean algebraic expressions

a) $\overline{x}+y$



- Write the circuits for the following Boolean algebraic expressions

b) $\overline{(x+y)}x$



How to add binary numbers

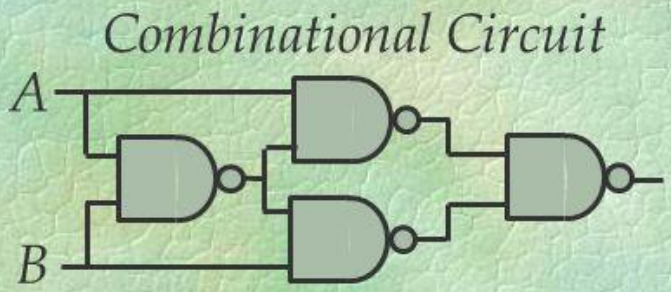
- Consider adding two 1-bit binary numbers *x* and *y*
  - 0+0 = 0
  - 0+1 = 1
  - 1+0 = 1
  - 1+1 = 10

x	*y*	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Carry is *x* AND *y*
- Sum is *x* XOR *y*
- The circuit to compute this is called a half-adder

Combinational Logic : Definition

- **Combinational Logic is a logical circuit consists of logic gates whose outputs at any time are determined directly from the present combination of inputs without regard to previous inputs**



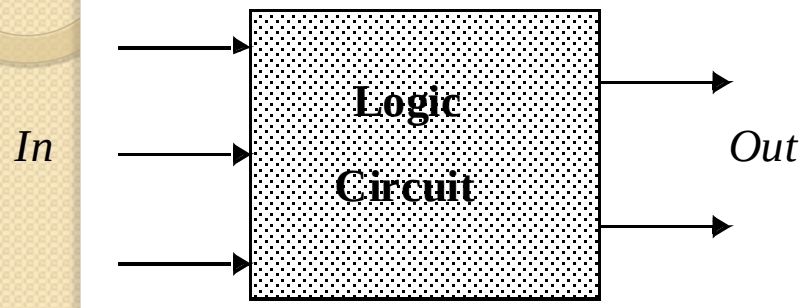
https://images.rambler.ru/search?query=combinational%20logic%2C%20ppt&adult=soft&i=0&utm_source=rambler-search&utm_medium=mixup&utm_campaign=images_up

Design Procedure

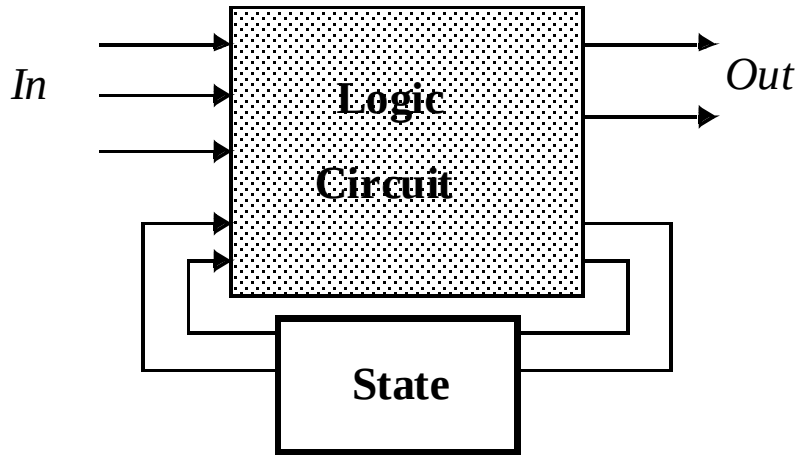
- From Boolean function to logic diagram
- Steps for procedure:
- 1. The problem is stated.
- 2. The number of available input variables and required output variables is determined.
- 3. The input and output variables are assigned letter symbols.

- 4. The truth table that defines the required relationships between inputs and outputs is derived.
- 5. The simplified Boolean function for output is obtained.
- 6. The logic diagram is drawn.

Combinational vs. Sequential Logic



(a) Combinational



(b) Sequential

Output = *f*(*In*)

Output = *f*(*In, Previous In*)

Half Adder

- A combinational circuit that performs the addition of two bits is called a half adder.
- The truth table for the half adder is listed below:

Table 4-3
Half Adder

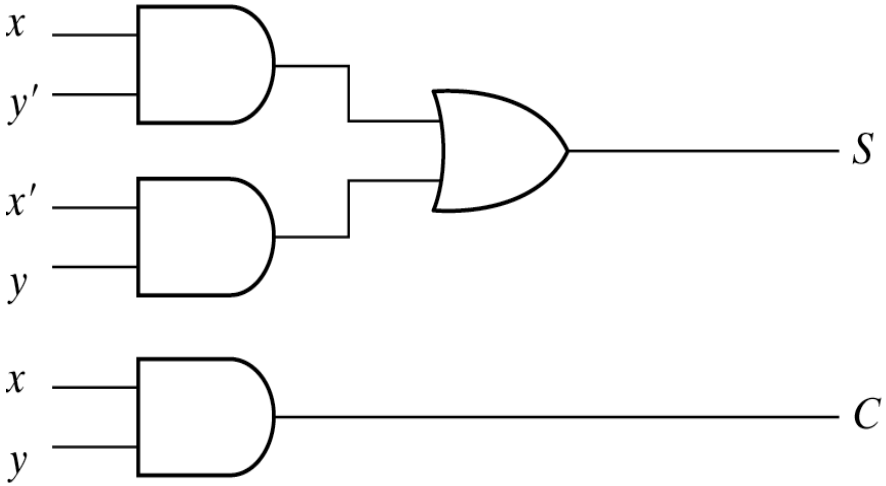
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

S: Sum
C: Carry

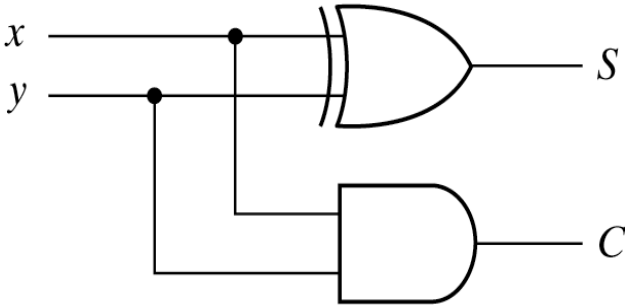
$$S = x'y + xy'$$

$$C = xy$$

Implementation of Half-Adder



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



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

Full-Adder

- One that performs the addition of three bits(two significant bits and a previous carry) is a full adder.

Table 4-4
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

Simplified Expressions

$x \backslash yz$	00	01	11	10
0		1		1
1	1		1	

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

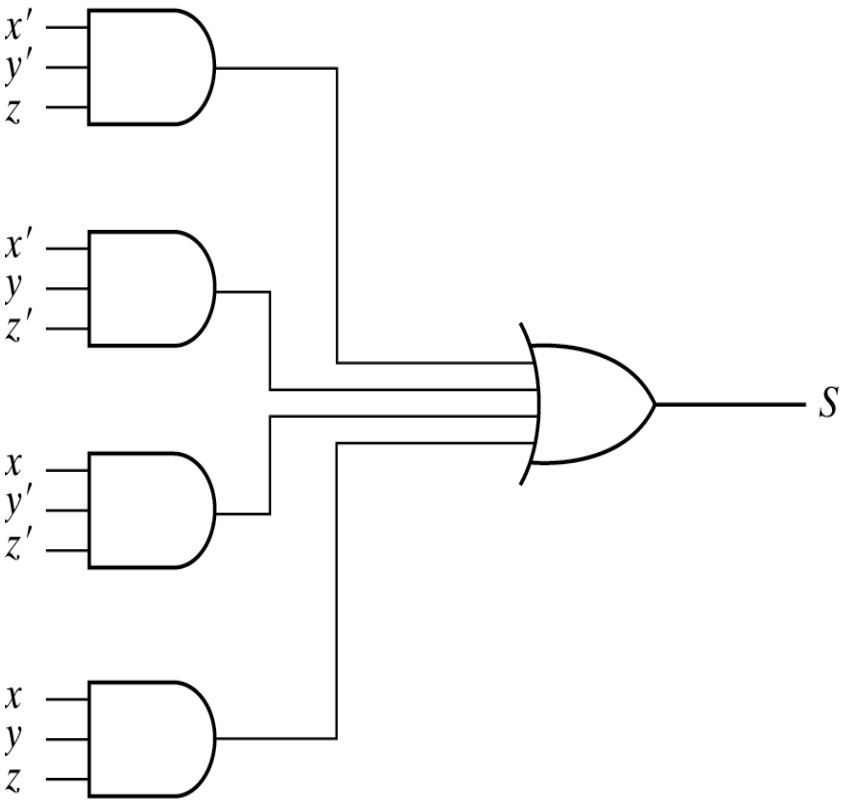
$x \backslash yz$	00	01	11	10
0			1	
1		1	1	1

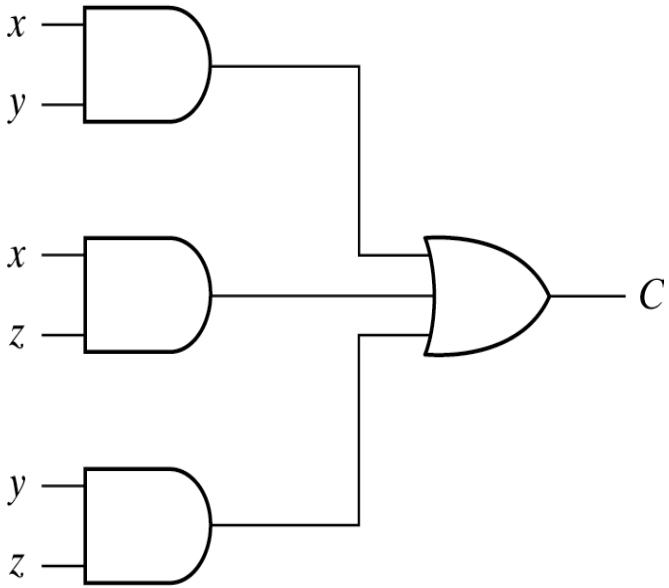
$$S = xy + xz + yz$$

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

C = xy + xz + yz

Full adder implemented





Another implementation

- Full-adder can also implemented with two half adders and one OR gate (Carry Look-Ahead adder).

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

$$C = z(xy' + x'y) + xy = xy'z + x'yz + xy$$

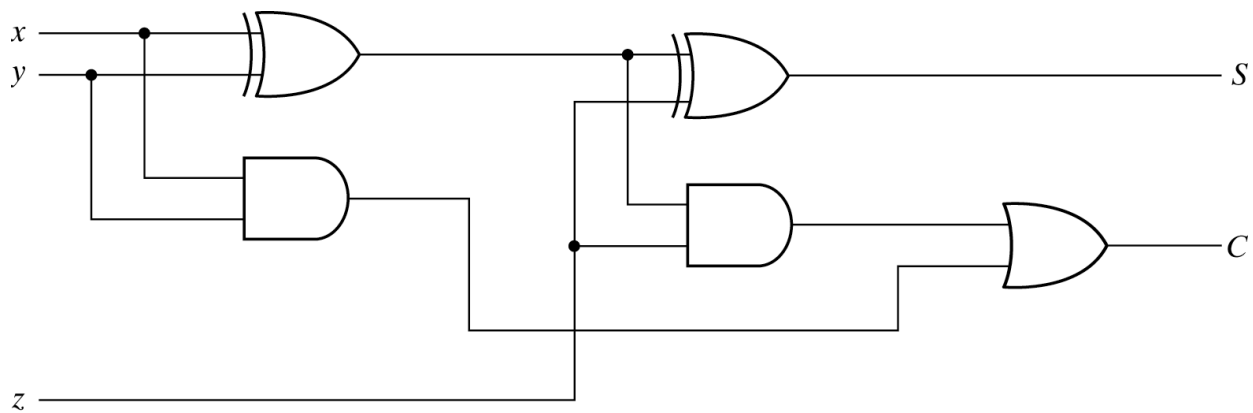
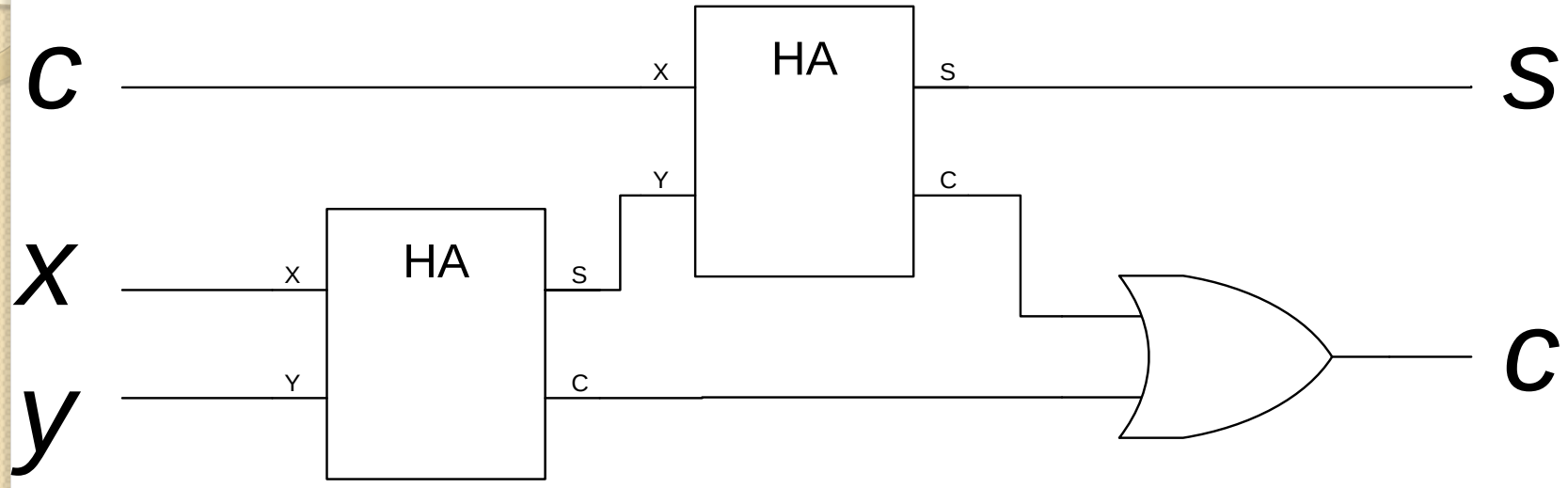
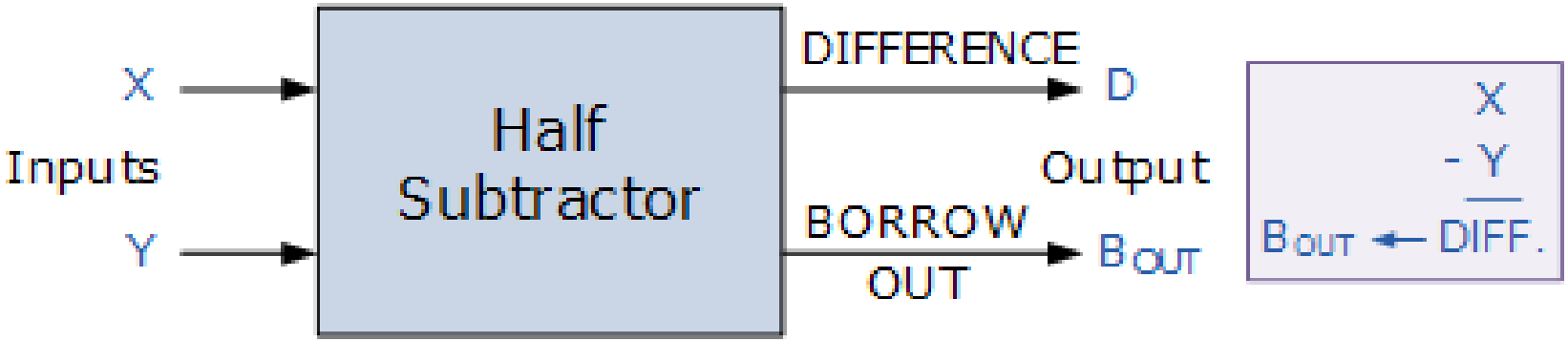


Fig: Implementation of Full Adder with Two Half Adders and an OR Gate

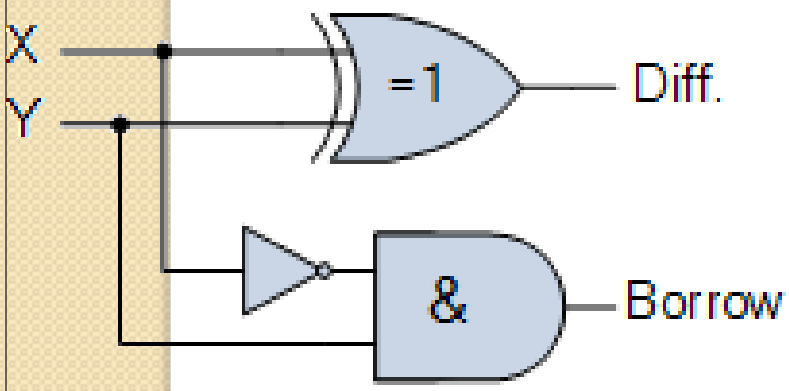

c
x
y
HA
X
Y
S
C
HA
X
Y
S
C
S
c

A Half Subtractor Circuit

- A half subtractor is a logical circuit that performs a subtraction operation on two binary digits.
- The half subtractor produces a difference and a borrow bit for the next stage.



http://www.electronics-tutorials.ws/combination/binary-subtractor.html

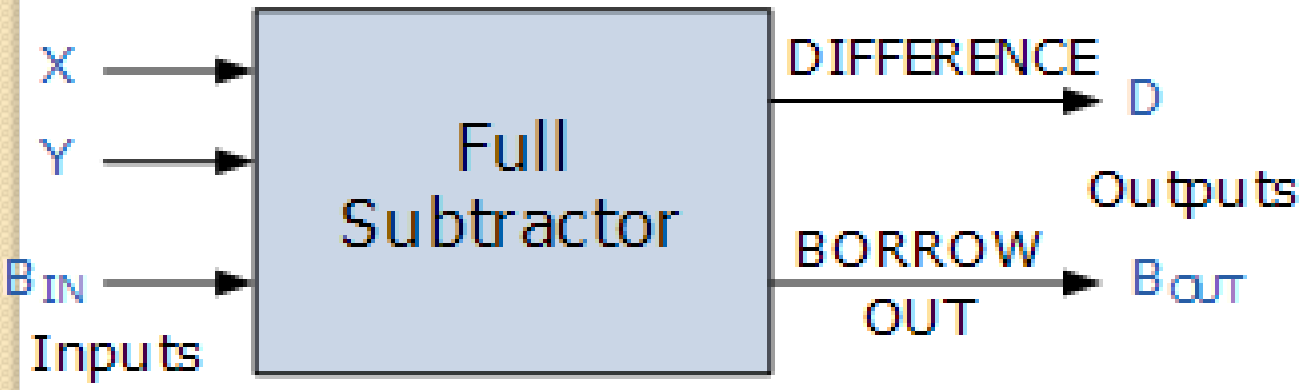
Symbol	Truth Table			
	Y	X	DIFFERENCE	BORROW
	0	0	0	0
	0	1	1	0
	1	0	1	1
	1	1	0	0

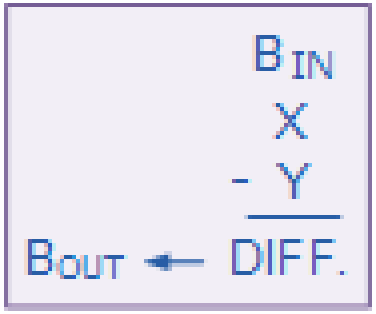
D = x'y + xy'

B = x'y

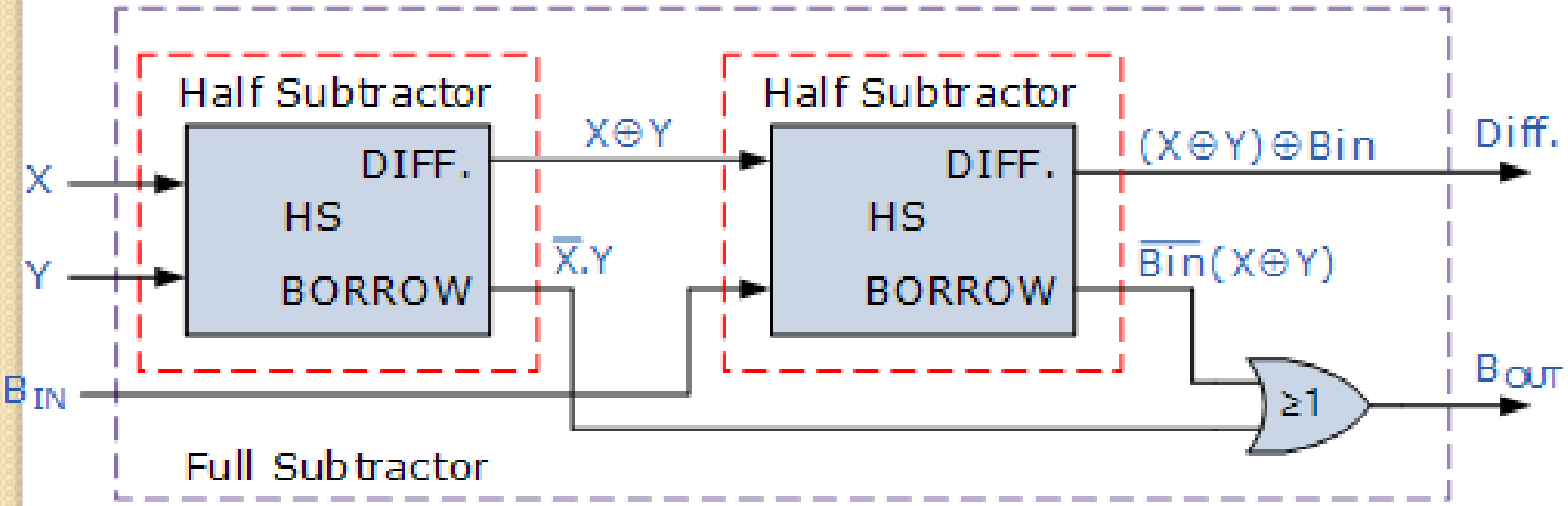
Full Binary Subtractor

- It has three inputs.
- The two single bit data inputs X (minuend) and Y (subtrahend) and an additional *Borrow-in* (B-in) input to receive the borrow generated by the subtraction process from a previous stage as shown below.



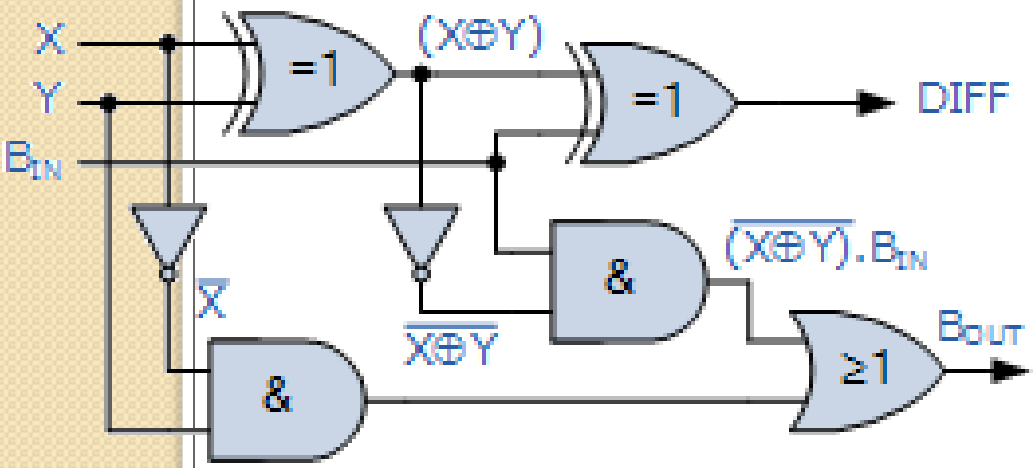


- The full subtractor can also be thought of as two half subtractors connected together, with the first half subtractor passing its borrow to the second half subtractor as follows.



Full Subtractor Truth Table

Symbol



Truth Table

B-in	Y	X	Diff.	B-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

- Then the Boolean expression for a full subtractor is as follows.
- For the **DIFFERENCE** (D) bit:
- D = $(\bar{X}.\bar{Y}.B_{IN}) + (\bar{X}.Y.\bar{B}_{IN}) + (X.\bar{Y}\bar{B}_{IN}) + (X.Y.B_{IN})$
- which can be simplified too:
- D = (X XOR Y) XOR B_{IN} = $(X \oplus Y) \oplus B_{IN}$
- For the **BORROW OUT** (B_{OUT}) bit:
- $B_{OUT} = (\bar{X}.\bar{Y}.B_{IN}) + (\bar{X}.Y.\bar{B}_{IN}) + (\bar{X}.Y.B_{IN}) + (X.Y.B_{IN})$
- which will also simplify too:
- B_{OUT} = X AND Y OR $(\overline{X\ XOR\ Y})B_{IN} = \bar{X}.Y + (\overline{X \oplus Y})B_{IN}$

Analysis procedure

- To obtain the output Boolean functions from a logic diagram, proceed as follows:

1. Label all gate outputs that are a function of input variables with arbitrary symbols. Determine the Boolean functions for each gate output.

2. Label the gates that are a function of input variables and previously labeled gates with other arbitrary symbols. Find the Boolean functions for these gates.

Analysis procedure

3. Repeat the process outlined in step 2 until the outputs of the circuit are obtained.

4. By repeated substitution of previously defined functions, obtain the output Boolean functions in terms of input variables.

Binary parallel adder

- This is also called **Ripple Carry Adder** ,because of the construction with full adders are connected in cascade.

Subscript i:	3	2	1	0	
Input carry	0	1	1	0	C_i
Augend	1	0	1	1	A_i
Addend	0	0	1	1	B_i
Sum	1	1	1	0	S_i
Output carry	0	0	1	1	C_{i+1}

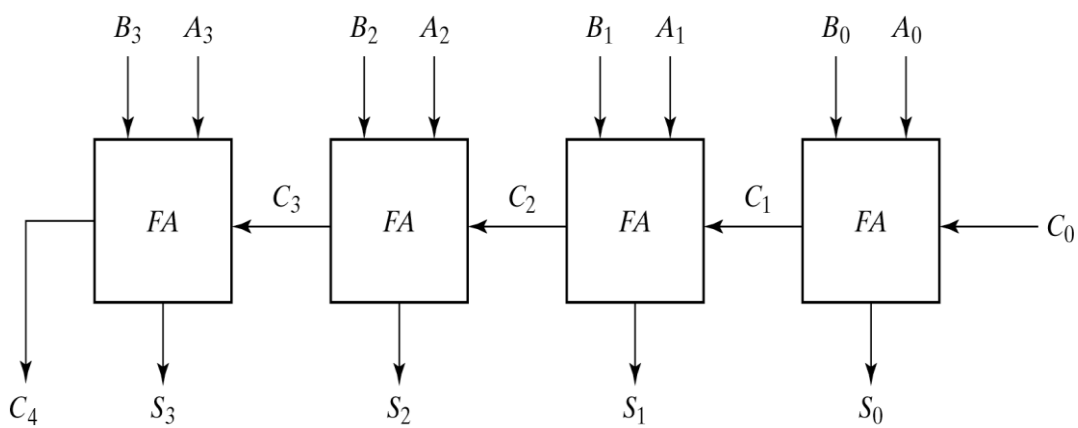


Fig. 4-9 4-Bit Adder

Decimal adder

BCD adder can't exceed 9 on each input digit. K is the carry.

Table 4-5
Derivation of BCD Adder

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

Rules of BCD adder

- When the binary sum is greater than 1001, we obtain a non-valid BCD representation.
- The addition of binary 6(0110) to the binary sum converts it to the correct BCD representation and also produces an output carry as required.
- To distinguish them from binary 1000 and 1001, which also have a 1 in position Z_8, we specify further that either Z_4 or Z_2 must have a 1.

$$C = K + Z_8Z_4 + Z_8Z_2$$

Implementation of BCD adder

- A decimal parallel adder that adds n decimal digits needs n BCD adder stages.
- The output carry from one stage must be connected to the input carry of the next higher-order stage.

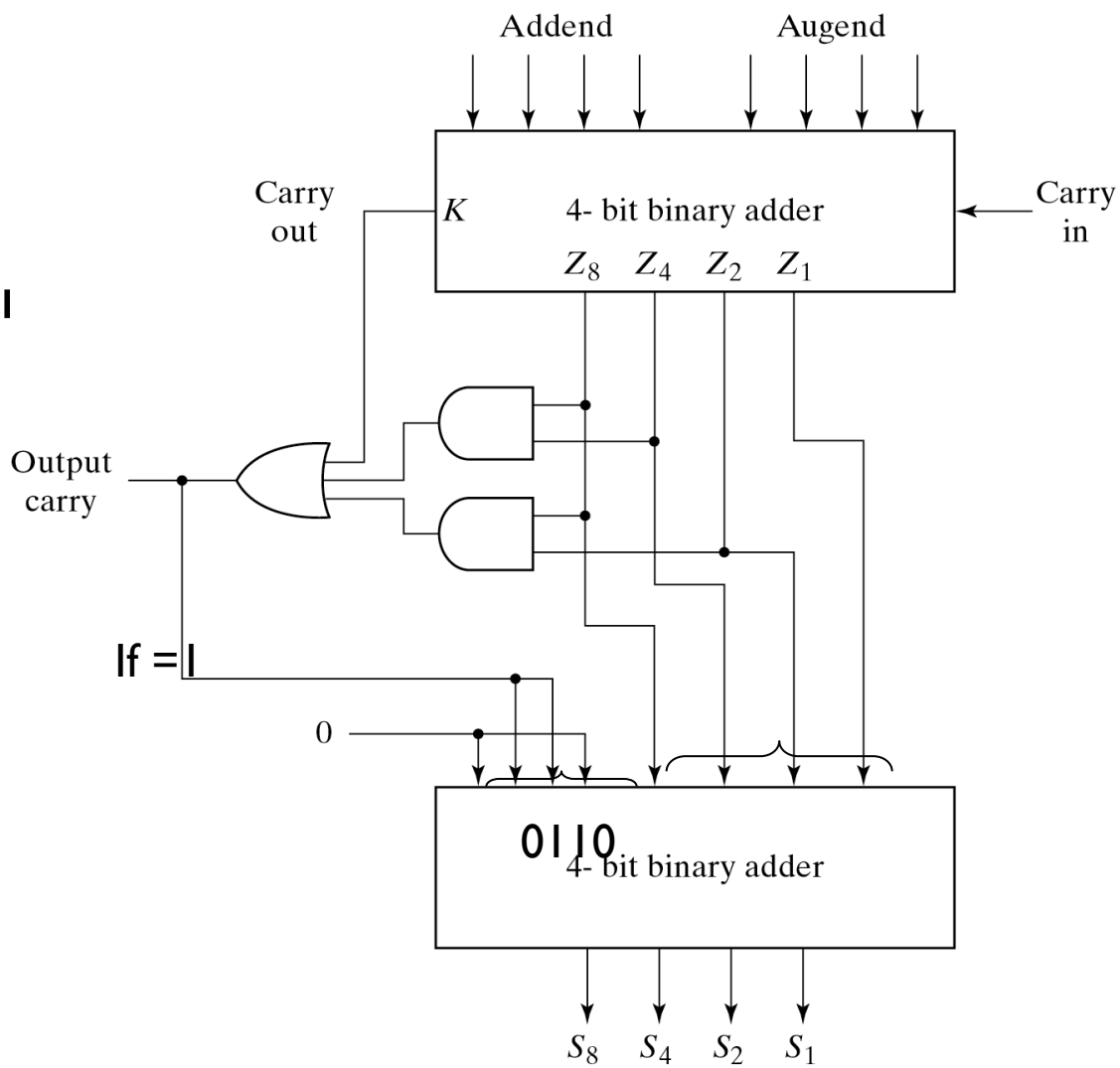


Fig. 4-14 Block Diagram of a BCD Adder

- A magnitude comparator is a combinational circuit that compares two numbers A & B to determine whether: $A > B$, or $A = B$, or $A < B$

Decoders

- The decoder is called n-to-m-line decoder, where $m \leq 2^n$.
- the decoder is also used in conjunction with other code converters such as a BCD-to-seven_segment decoder.
- 3-to-8 line decoder: For each possible input combination, there are seven outputs that are equal to 0 and only one that is equal to 1.

Implementation and truth table

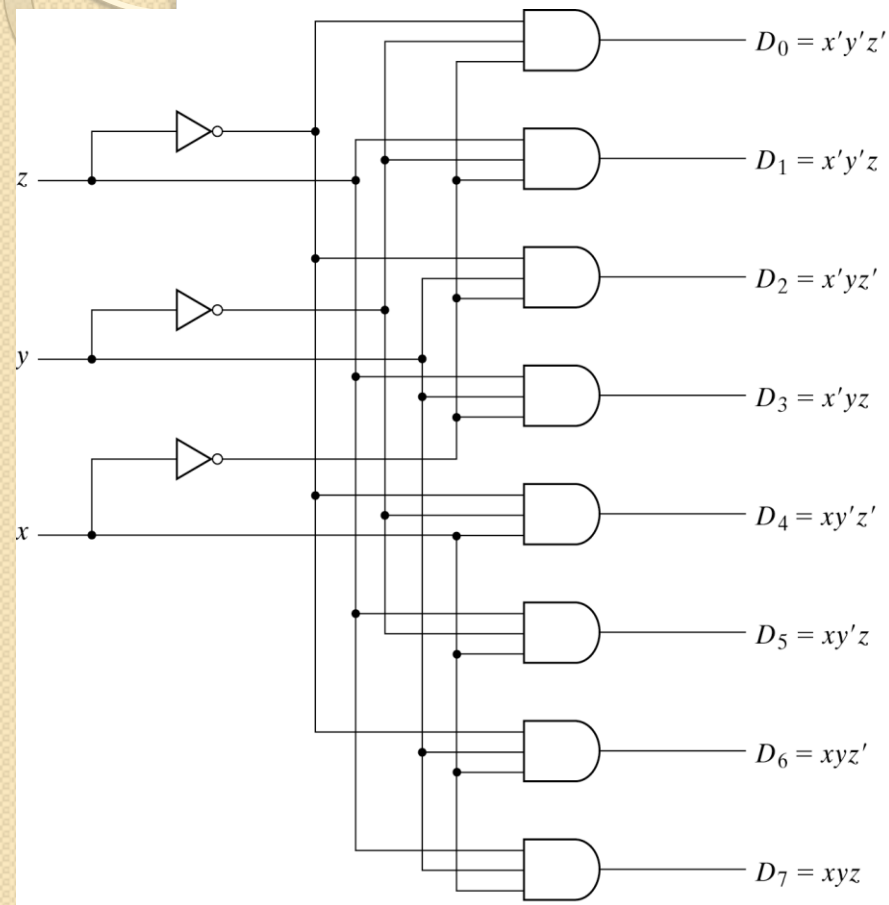


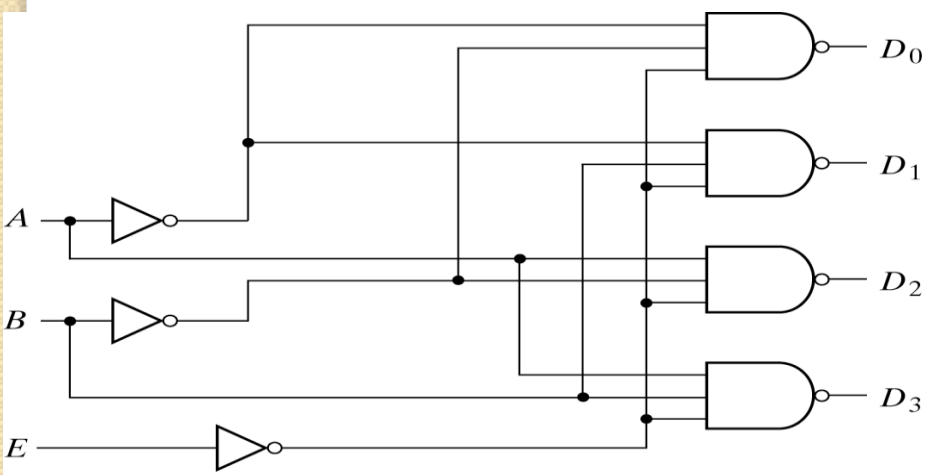
Fig. 4-18 3-to-8-Line Decoder

Table 4-6
Truth Table of a 3-to-8-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

Decoder with enable input

- Some decoders are constructed with NAND gates, it becomes more economical to generate the decoder minterms in their complemented form.
- As indicated by the truth table , only one output can be equal to 0 at any given time, all other outputs are equal to 1.



(a) Logic diagram

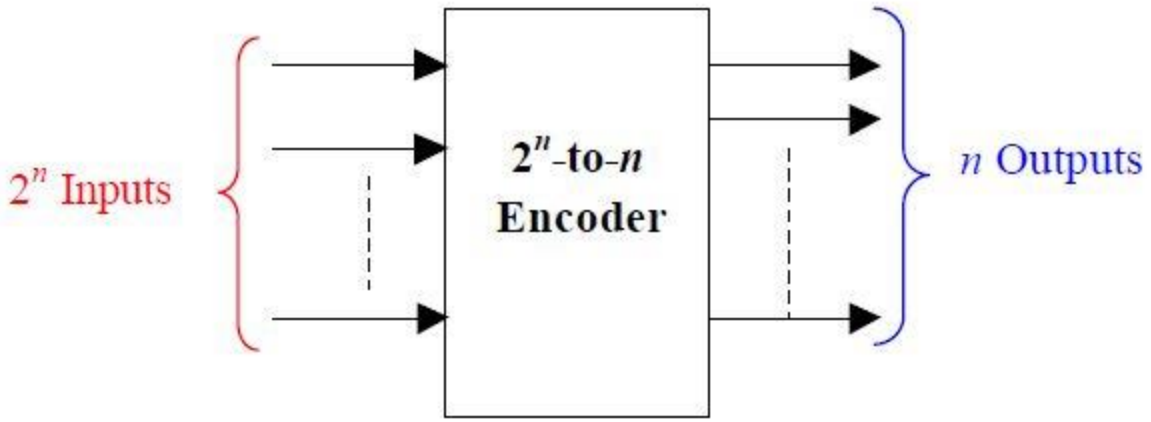
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

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

Encoders

- An encoder performs the inverse operation of a decoder.
- It has 2^n inputs, and n output lines.
- Only one input can be logic 1 at any given time (active input). All other inputs must be 0's.
- Output lines generate the binary code corresponding to the active input.



Encoders

Table 4-7
Truth Table of 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

$z = D_1 + D_3 + D_5 + D_7$

$y = D_2 + D_3 + D_6 + D_7$

$x = D_4 + D_5 + D_6 + D_7$