



Jiangxi University of Science and Technology

DIGITAL DESIGN

Chapter 4 Combinational Logic





Chapter 4 Combinational Logic

- Logic circuits for digital systems may be **combinational** or **sequential**.
- A combinational circuit consists of input variables, logic gates, and output variables.

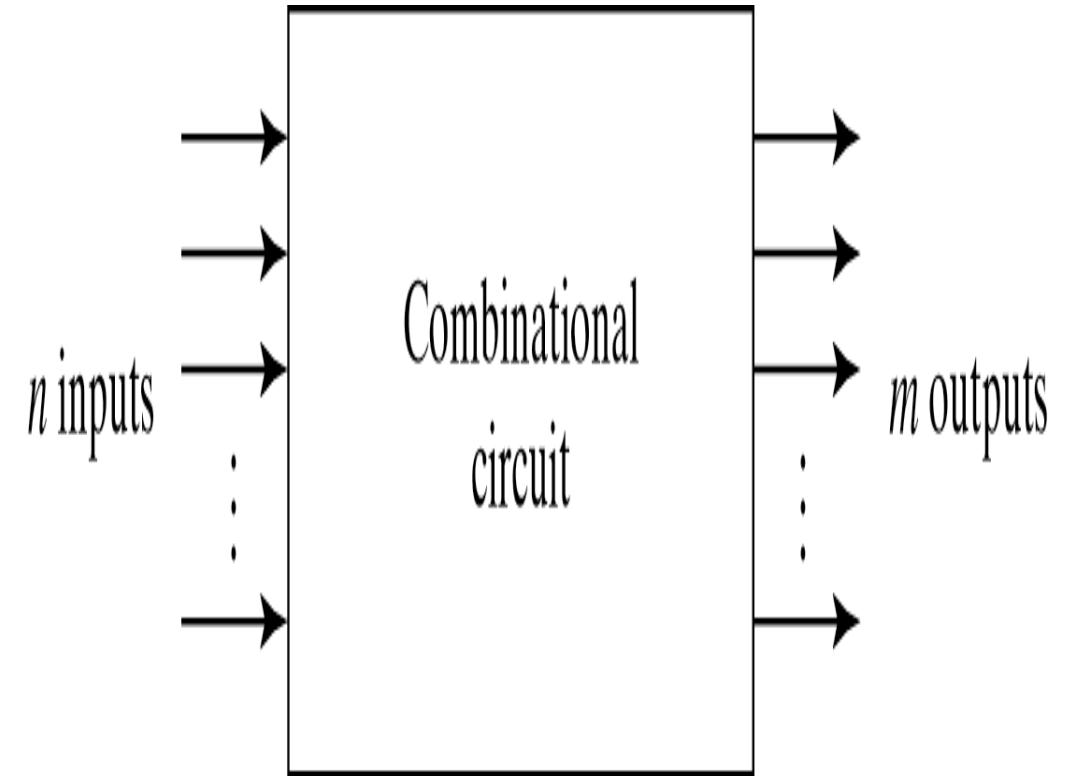


Fig. 4-1 Block Diagram of Combinational Circuit



Where they are used

- **Multiplexers**

- Selectors for routing data to the processor, memory, I/O
- Multiplexers route the data to the correct bus or port.

- **Decoders**

- are used for selecting things like a bank of memory and then the address within the bank. This is also the function needed to 'decode' the instruction to determine the operation to perform.

- **Encoders**

- are used in various components such as keyboards.



Analysis procedure

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

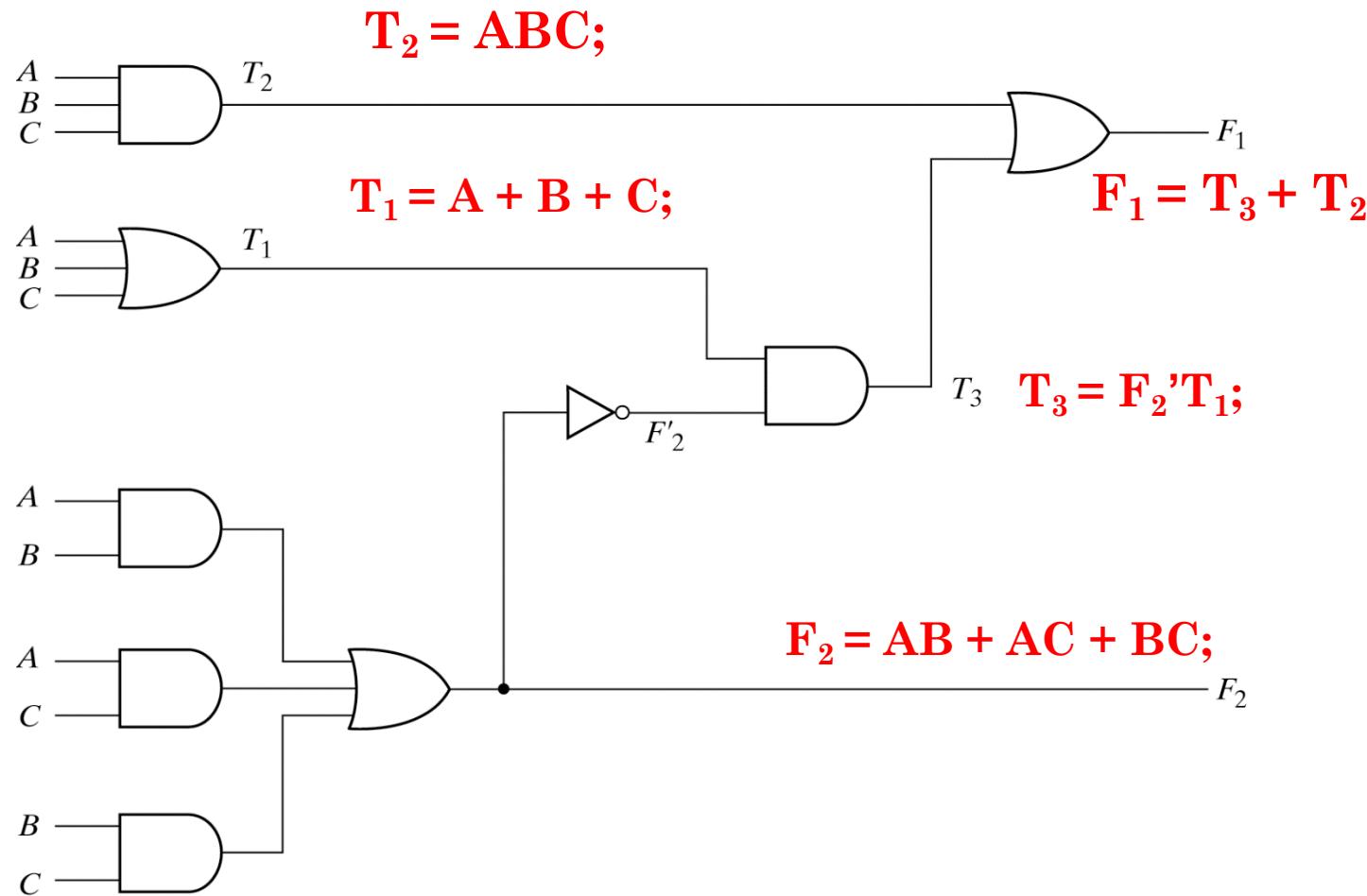
1. Label all gate outputs that are a function of input variables with arbitrary symbols. Determine the Boolean functions for each gate output.
2. Label the gates that are a function of input variables and previously labeled gates with other arbitrary symbols. Find the Boolean functions for these gates.
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.



Combinational Logic Design

- A process with 5 steps
 - Specification
 - Formulation
 - Optimization
 - Technology mapping
 - Verification

Example



$$\begin{aligned}F_1 &= T_3 + T_2 \\&= F_2' T_1 + ABC \\&= A'BC' + A'B'C + AB'C' + ABC\end{aligned}$$

Fig. 4-2 Logic Diagram for Analysis Example

Derive truth table from logic diagram

- We can derive the truth table in Table 4-1 by using the circuit of Fig.4-2.

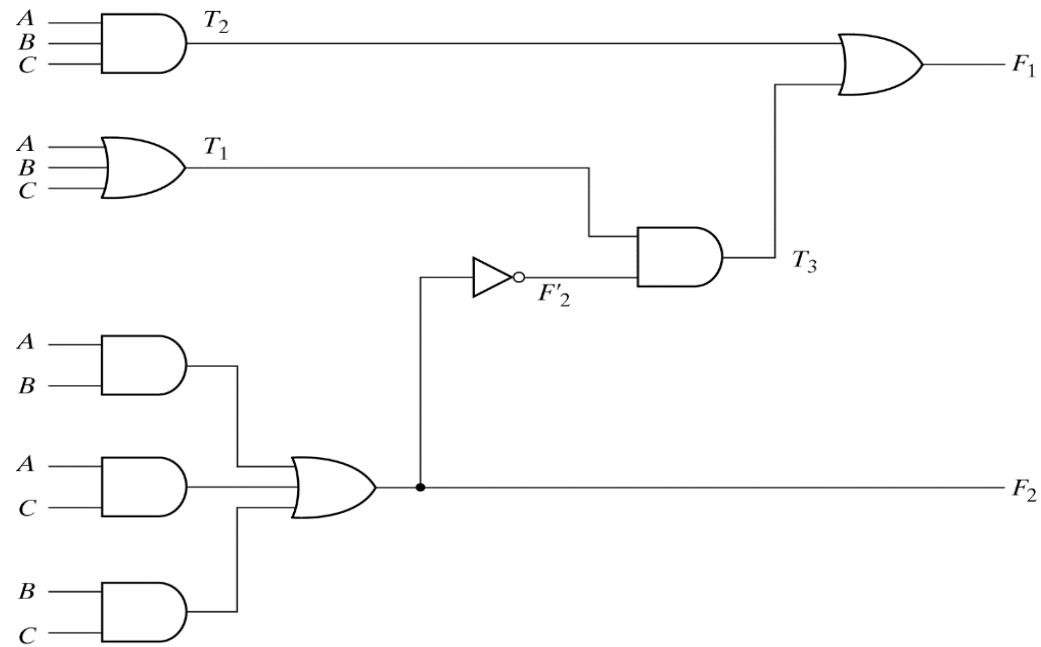


Fig. 4-2 Logic Diagram for Analysis Example

Table 4-1
Truth Table for the Logic Diagram of Fig. 4-2

A	B	C	F_2	F_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	0	1	1

4-3. Design procedure

- Table is a Code-Conversion example, first, we can list the relation of the BCD and Excess-3 codes in the truth table.
- Inputs: a BCD input, A,B,C,D with A as the most significant bit and D as the least significant bit.
- Outputs: an Excess-3 output W,X,Y,Z that corresponds to the BCD input.
- Internal operation – circuit to do the conversion in combinational logic.

Decimal Digit	Input BCD	Output Excess-3
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0



Formulation of BCD-to-Excess-3

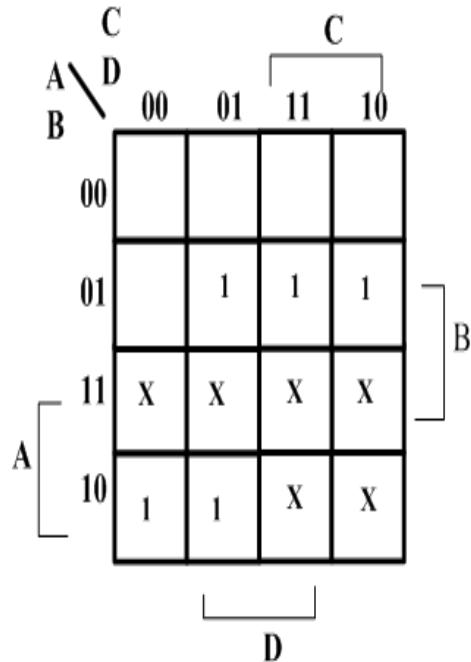
- Excess-3 code is easily formed by adding a binary 3 to the binary or BCD for the digit.
- There are 16 possible inputs for both BCD and Excess-3.
- It can be assumed that only valid BCD inputs will appear so the six combinations not used can be treated as don't cares.

Optimization – BCD-to-Excess-3

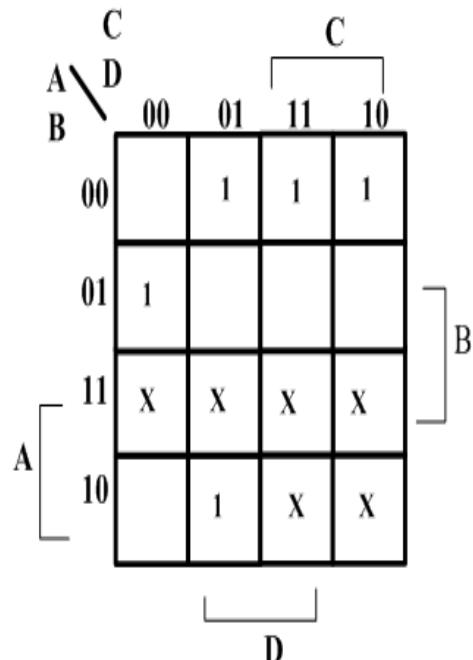


- Lay out K-maps for each output, W X Y Z

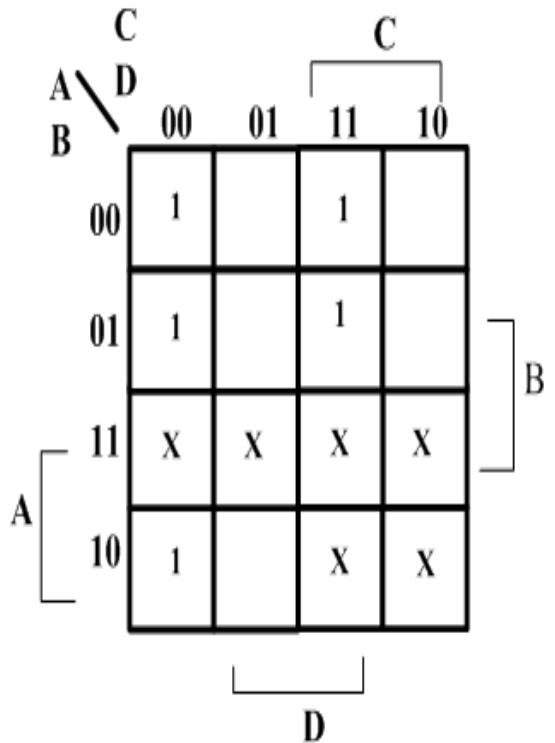
K-map for W



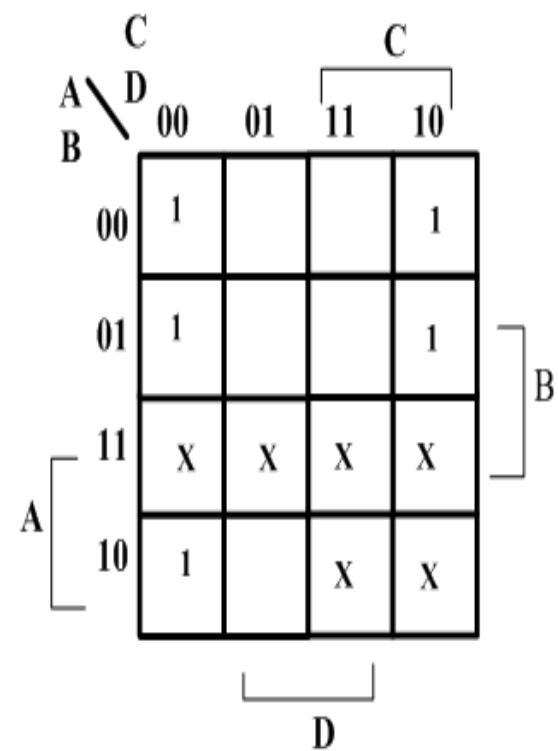
K-map for X



K-map for Y



K-map for Z



- A step in the digital circuit design process.

Karnaugh map

2. For each symbol of the Excess-3 code, we use 1's to draw the map for simplifying Boolean function.

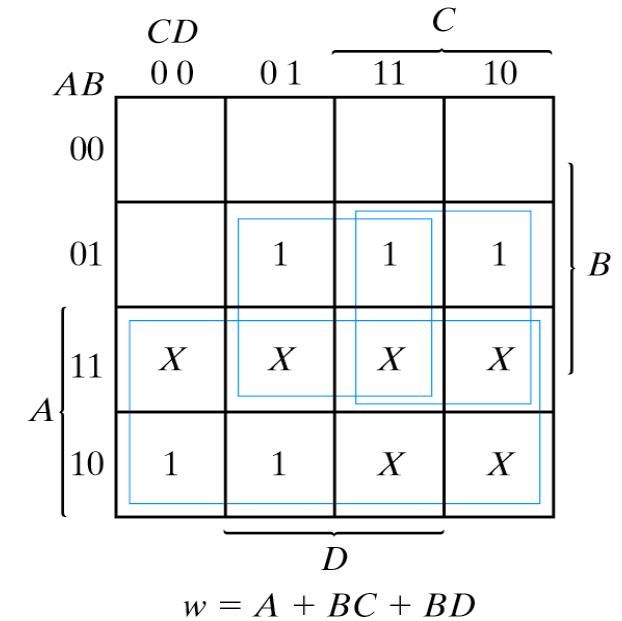
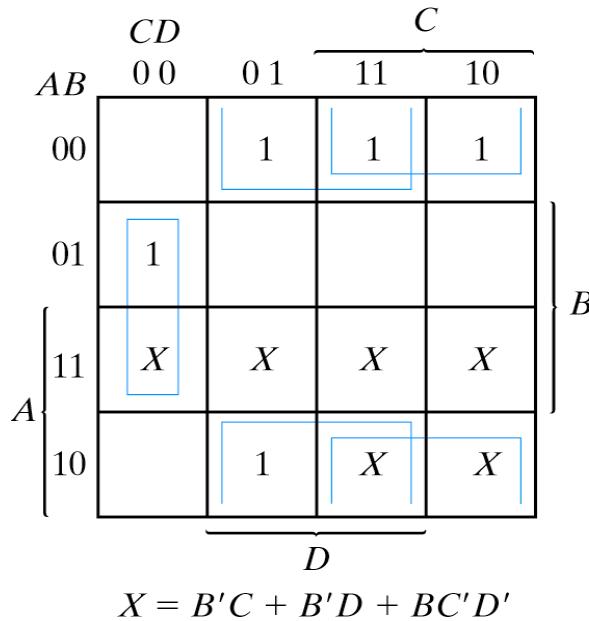
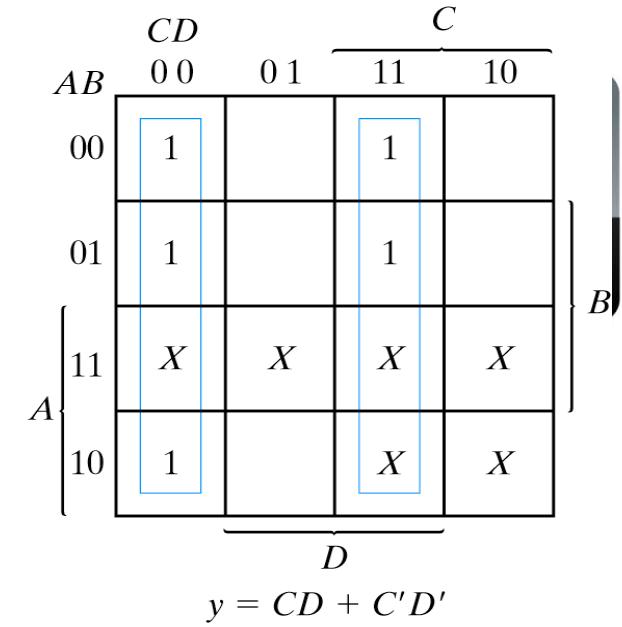
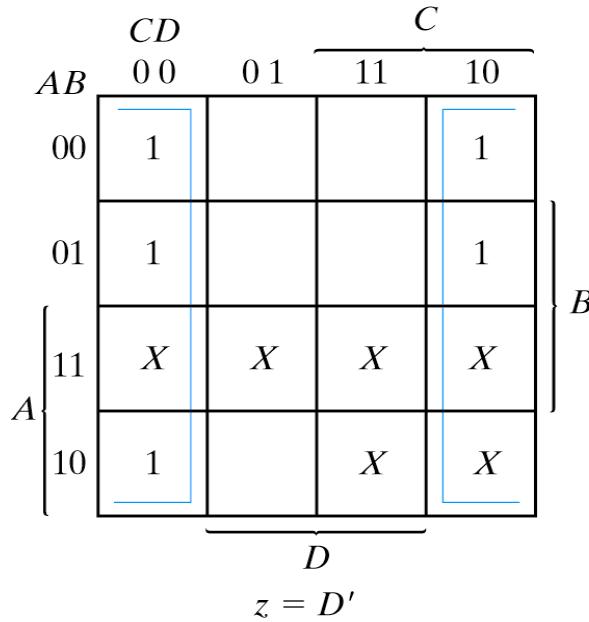


Fig. 4-3 Maps for BCD to Excess-3 Code Converter

Expressions for W X Y Z

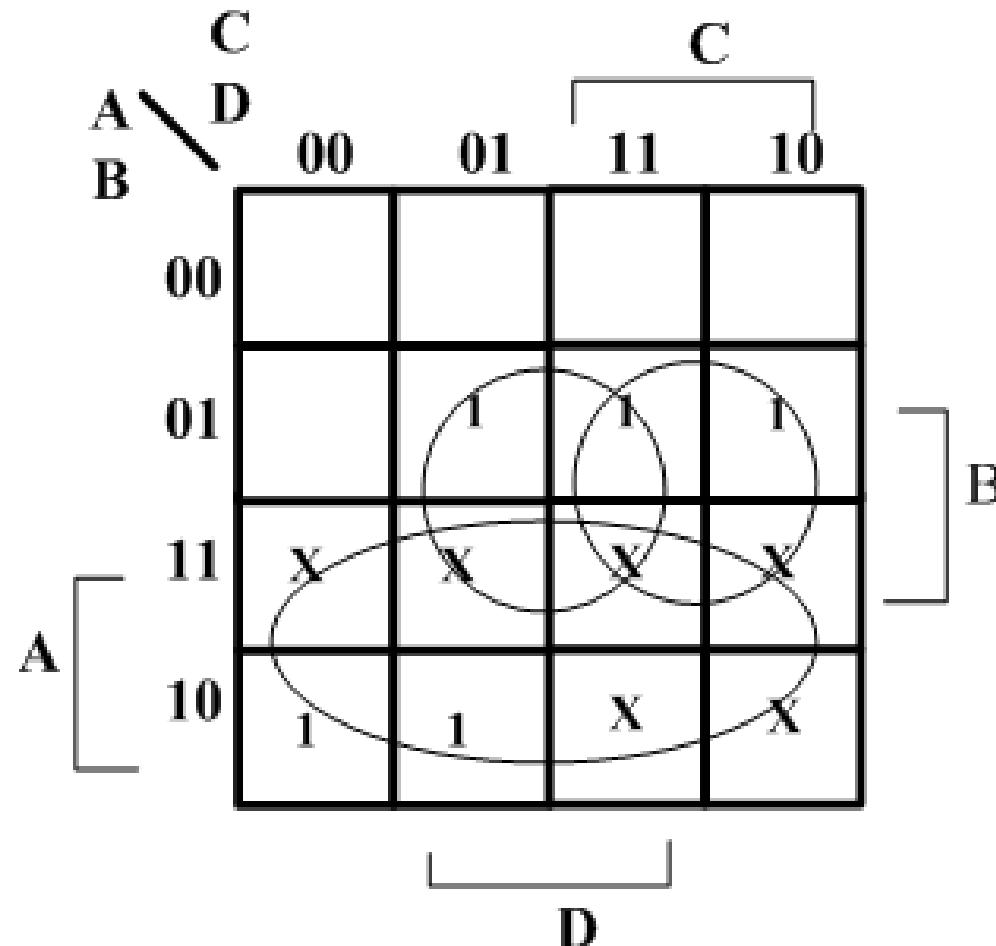
- $W(A,B,C,D) = \Sigma m(5,6,7,8,9) + d(10,11,12,13,14,15)$
- $X(A,B,C,D) = \Sigma m(1,2,3,4,9) + d(10,11,12,13,14,15)$
- $Y(A,B,C,D) = \Sigma m(0,3,4,7,8) + d(10,11,12,13,14,15)$
- $Z(A,B,C,D) = \Sigma m(0,2,4,6,8) + d(10,11,12,13,14,15)$

Minimize K-Maps

$$W(A,B,C,D) = \Sigma m(5,6,7,8,9) + d(10,11,12,13,14,15)$$

- W minimization
- Find $W = A + BC + BD$

K-map for W

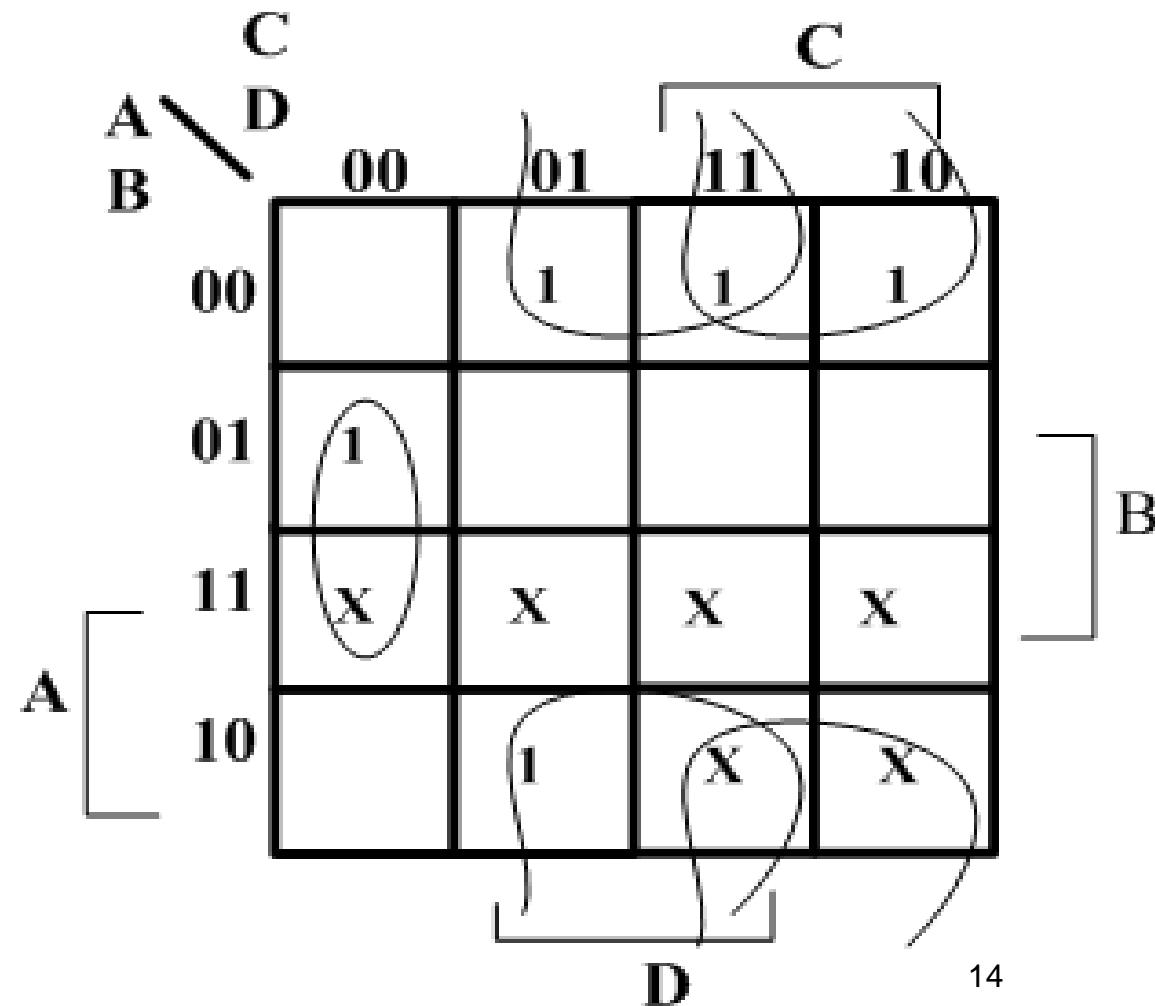


Minimize K-Maps

$$X(A,B,C,D) = \Sigma m(1,2,3,4,9) + d(10,11,12,13,14,15)$$

- X minimization
- Find $X = BC'D' + B'C + B'D$

K-map for X

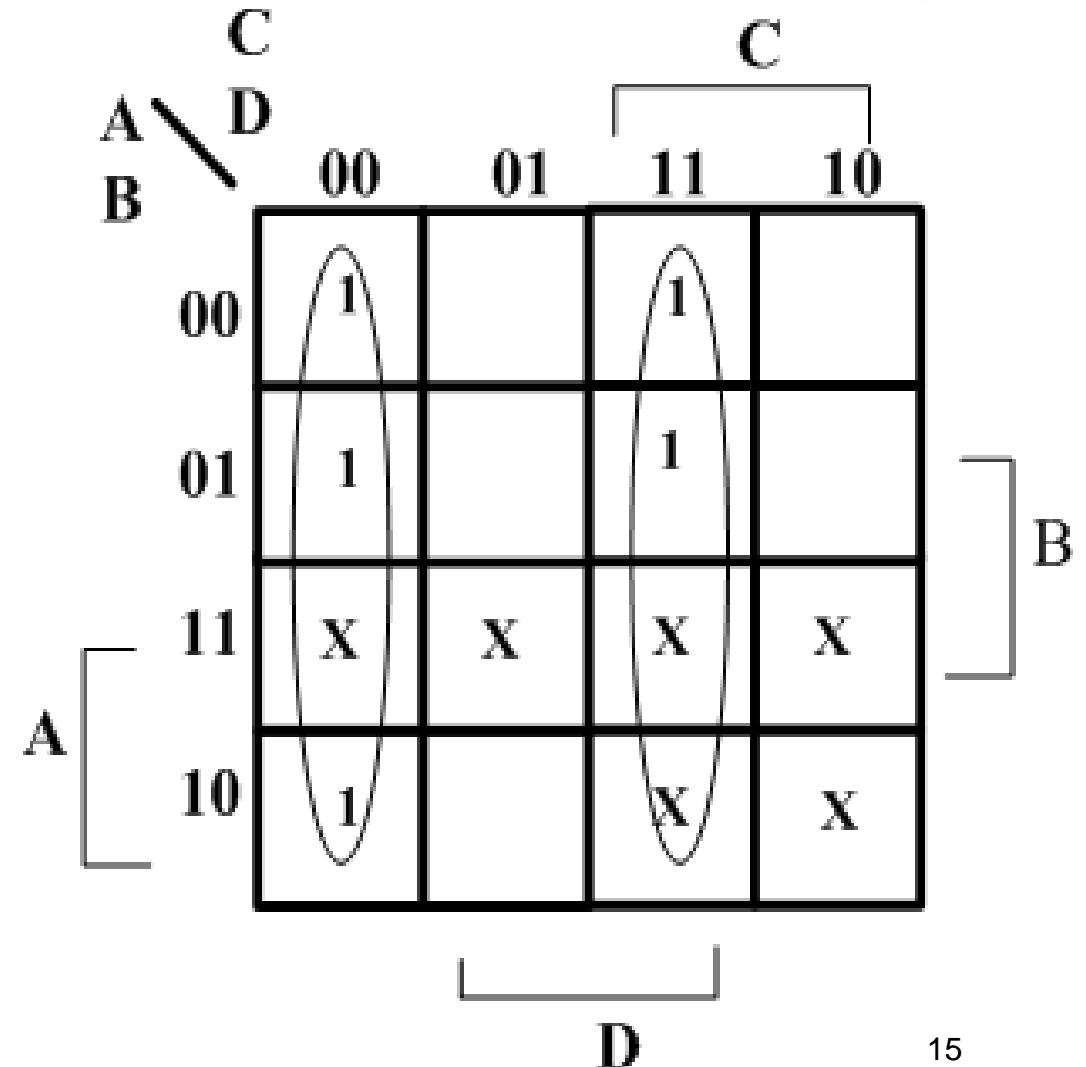


Minimize K-Maps

$$Y(A,B,C,D) = \Sigma m(0,3,4,7,8) + d(10,11,12,13,14,15)$$

- Y minimization
- Find $Y = CD + C'D'$

K-map for Y

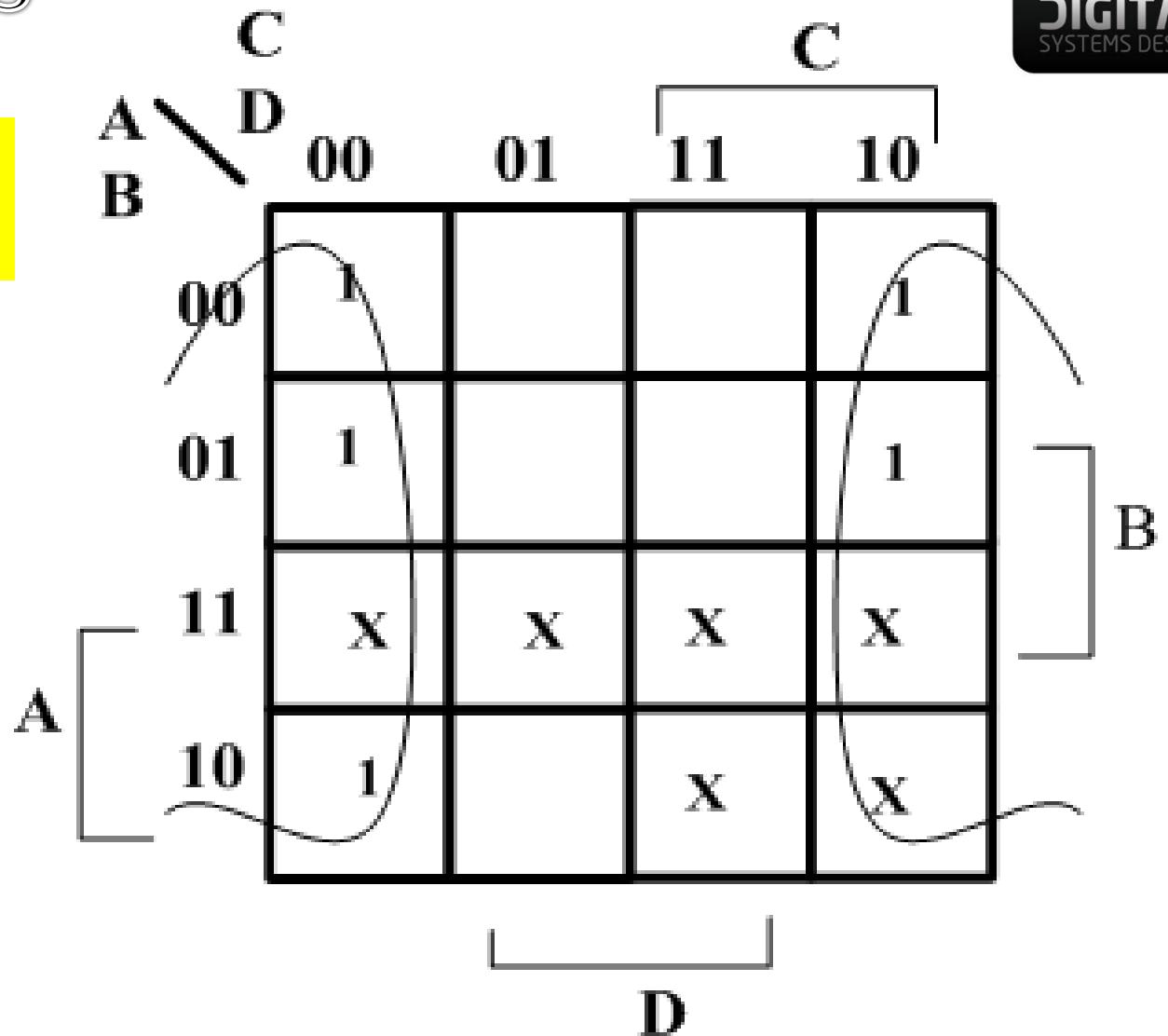


Minimize K-Maps

$$Z(A,B,C,D) = \Sigma m(0,2,4,6,8) + d(10,11,12,13,14,15)$$

- Z minimization
- Find $Z = D'$

K-map for Z



Circuit implementation



$$z = D'; y = CD + C'D' = CD + (C + D)'$$

$$x = B'C + B'D + BC'D' = B'(C + D) + B(C + D)'$$

$$w = A + BC + BD = A + B(C + D)$$

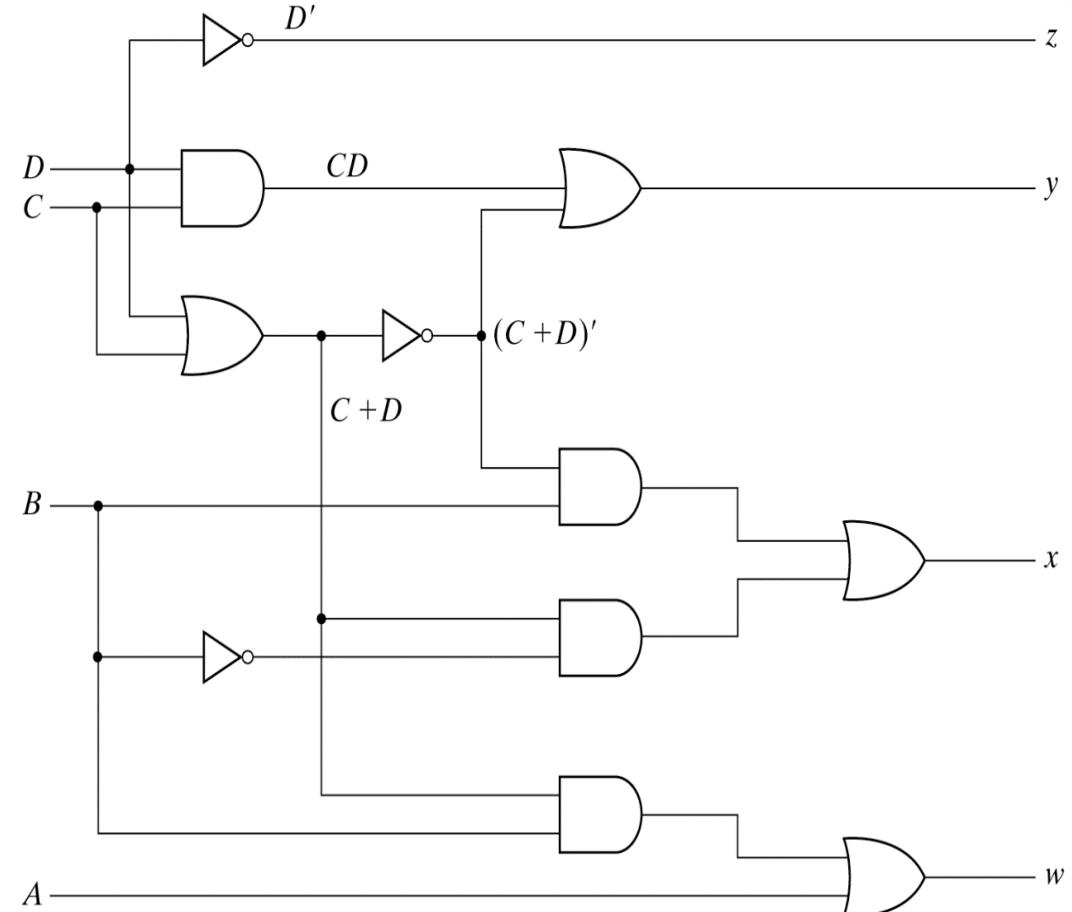


Fig. 4-4 Logic Diagram for BCD to Excess-3 Code Converter



Jiangxi University of Science and Technology

Adder /Subtractor

Half Adder

Full Adder

Half Subtractor

Half Adder

- **Half Adder:** is a combinational circuit that performs the addition of two bits, this circuit needs two binary inputs and two binary outputs.

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

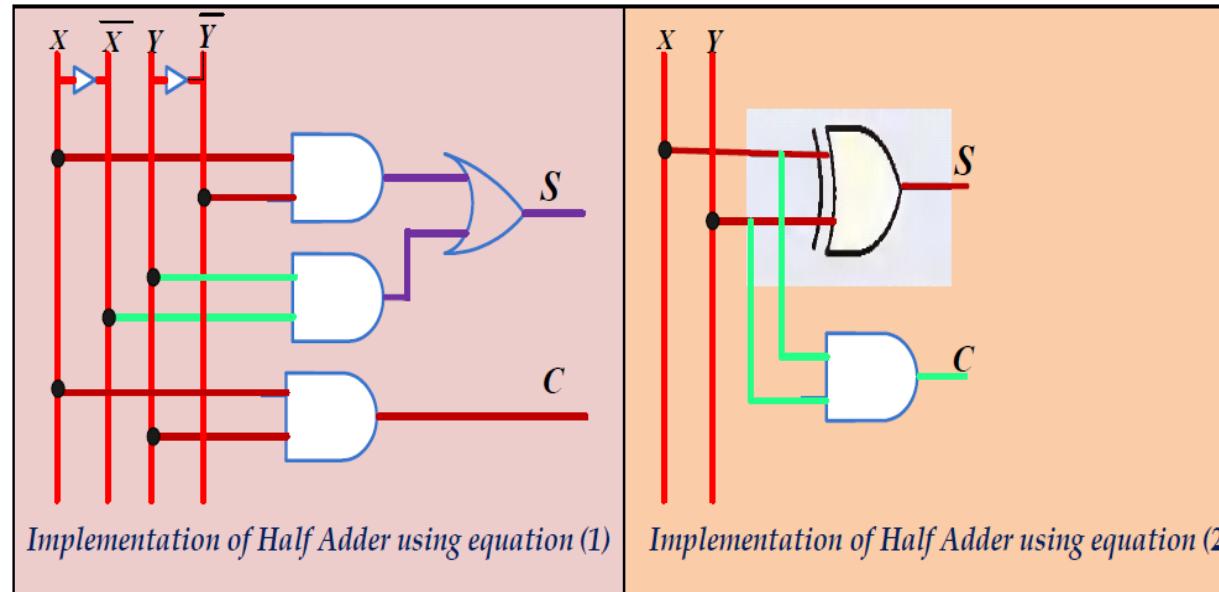
Truth table

The simplified Boolean function from the truth table:

$$\left. \begin{array}{l} S = \bar{X}Y + X\bar{Y} \\ C = XY \end{array} \right\} \text{(Using sum of product form)}$$

Where **S** is the sum and **C** is the carry.

$$\left. \begin{array}{l} S = X \oplus Y \\ C = XY \end{array} \right\} \text{(Using XOR and AND Gates)}$$



Binary Addition by Hand

- You can add two binary numbers one column at a time starting from the right, just like you add two decimal numbers.
- But remember it's binary. For example, $1 + 1 = 10$ and you have to carry!

The initial carry in is implicitly 0

$$\begin{array}{r}
 & 1 & 1 & 1 & 0 \\
 & 1 & 0 & 1 & 1 \\
 + & 1 & 1 & 1 & 0 \\
 \hline
 & 1 & 1 & 0 & 0 & 1
 \end{array}$$

Carry in
 Augend
 Addend
 Sum

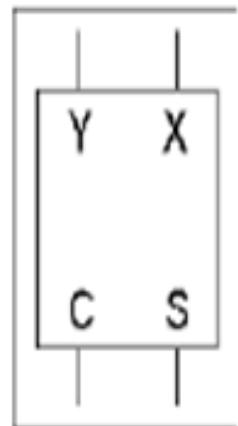
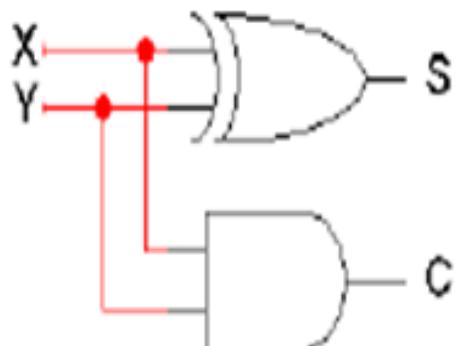
most significant bit (MSB) least significant bit (LSB)

Adding Two Bits

- We'll make a hardware adder based on our human addition algorithm.
- We start with a **half adder**, which adds two bits X and Y and produces a two-bit result: a **sum S** (the right bit) and a **carry out C** (the left bit).
- Here are truth tables, equations, circuit and block symbol.

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$\begin{aligned}C &= XY \\S &= X'Y + XY' \\&= X \oplus Y\end{aligned}$$



Adder

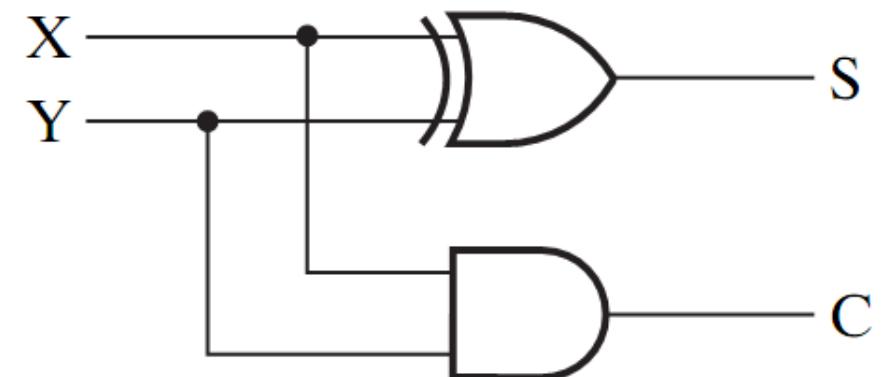
- Design an Adder for 1-bit numbers?
- **1. Specification:**
2 inputs (X,Y)
2 outputs (C,S)
- **2. Formulation:**

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Adder

- Design an Adder for 1-bit numbers?
 - 1. Specification:
 - 3. Optimization/Circuit
- 2. Formulation:

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



From the equation it is clear that this 1-bit adder can be easily implemented with the help of **EXOR Gate** for the output ‘SUM’ and an **AND Gate** for the carry.

Half Adder

A combinational circuit that performs the addition of two bits is called a **half adder**.

- This adder is called a Half Adder
- Q: Why?

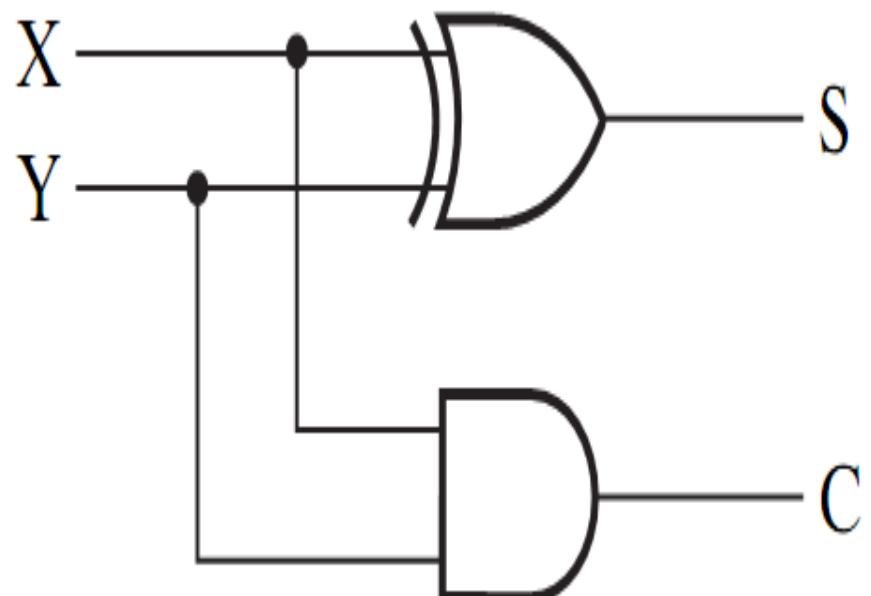
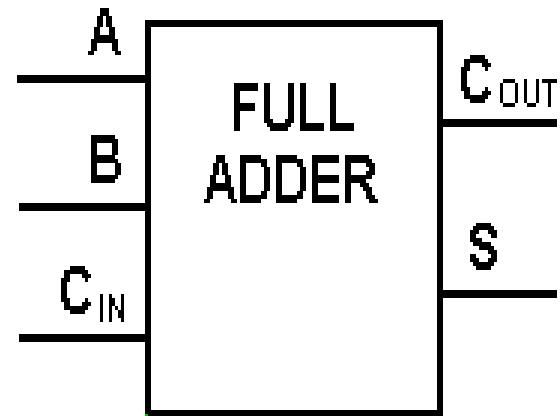
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

S: Sum

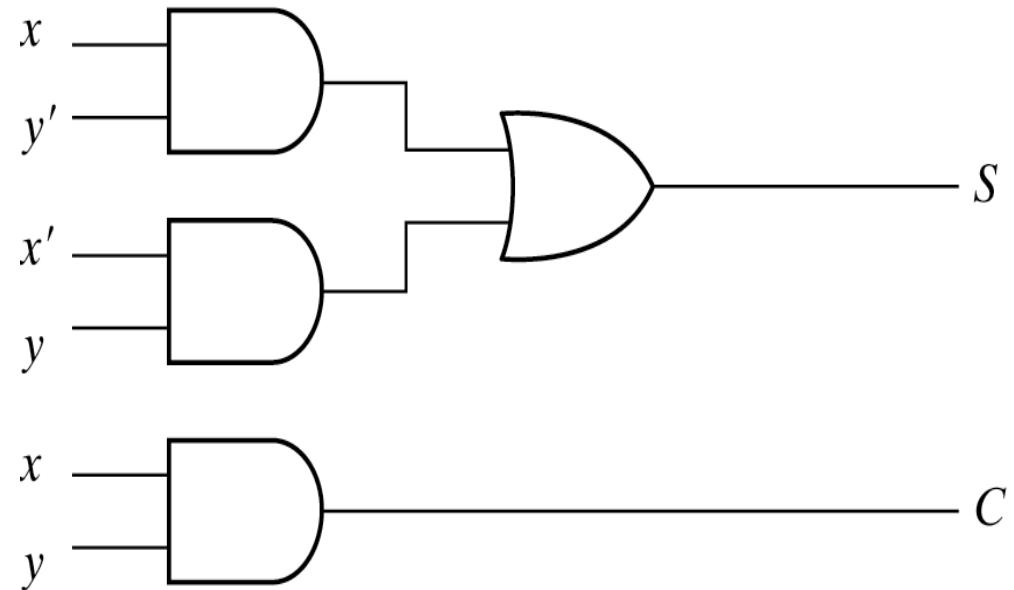
C: Carry

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

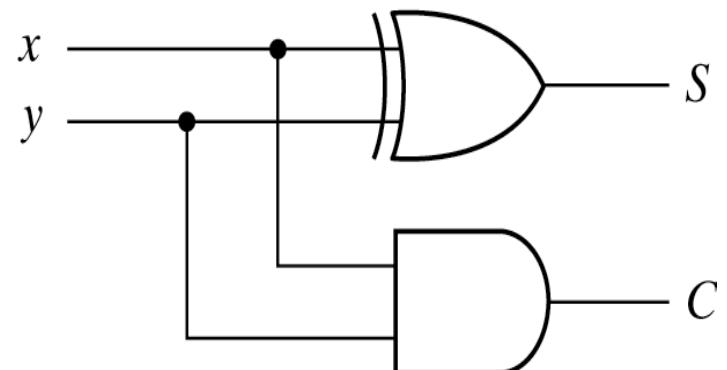
$$C = xy$$



Implementation of Half-Adder



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



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

Fig. 4-5 Implementation of Half-Adder

Full-Adder

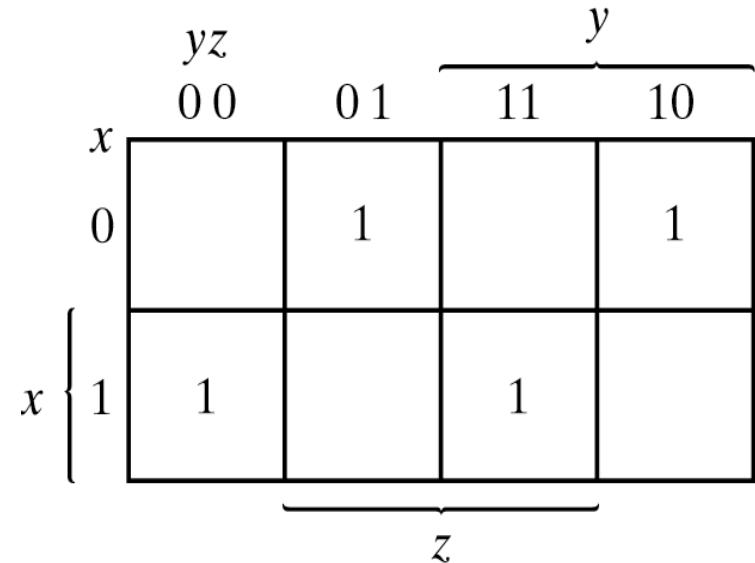
- One that performs the addition of three bits(two significant bits and a previous carry) is a full adder.

A combinational circuit that adds 3 input bits to generate a Sum bit and a Carry bit

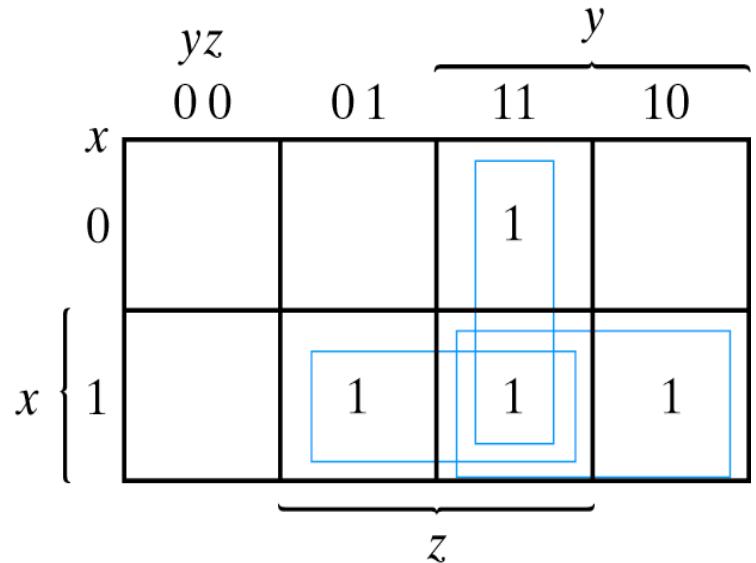
Table 4-4
Full Adder

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



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



$$\begin{aligned} S &= xy + xz + yz \\ &= xy + xy'z + x'yz \end{aligned}$$

Fig. 4-6 Maps for Full Adder

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

$$C = xy + xz + yz$$

Full adder implemented in SOP

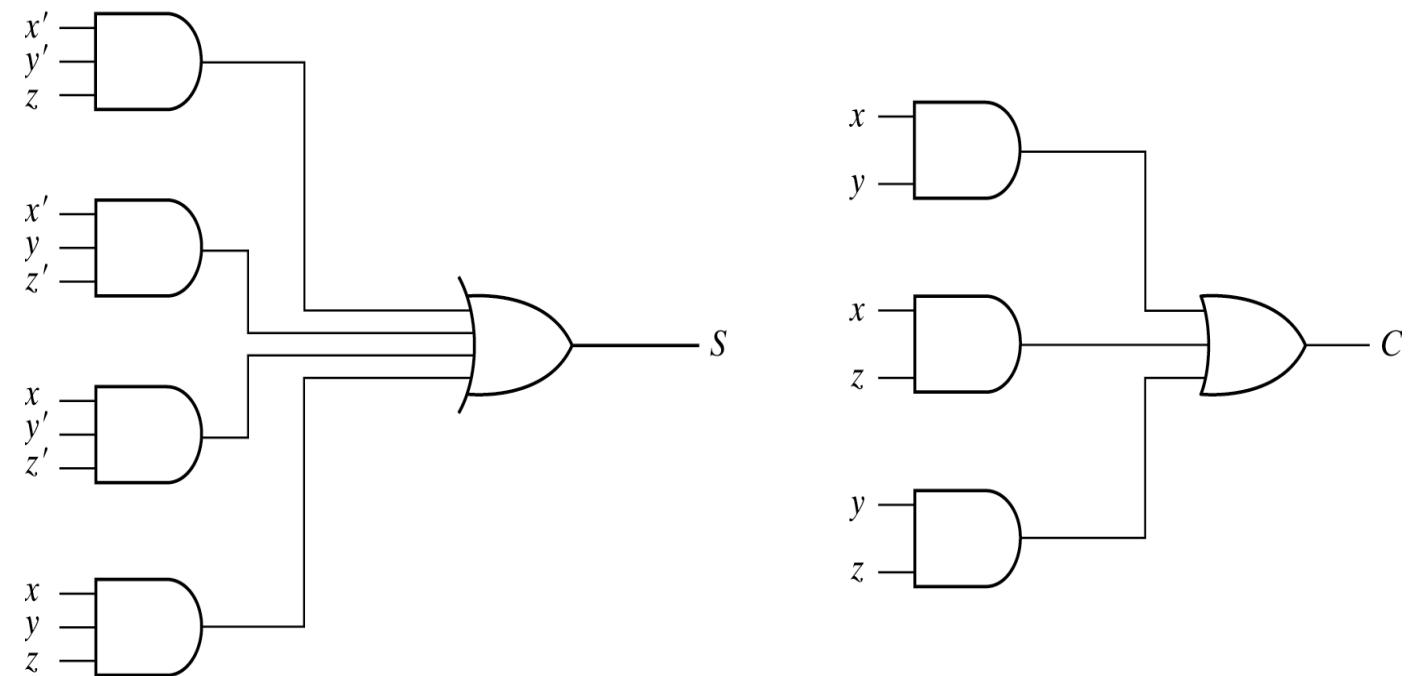


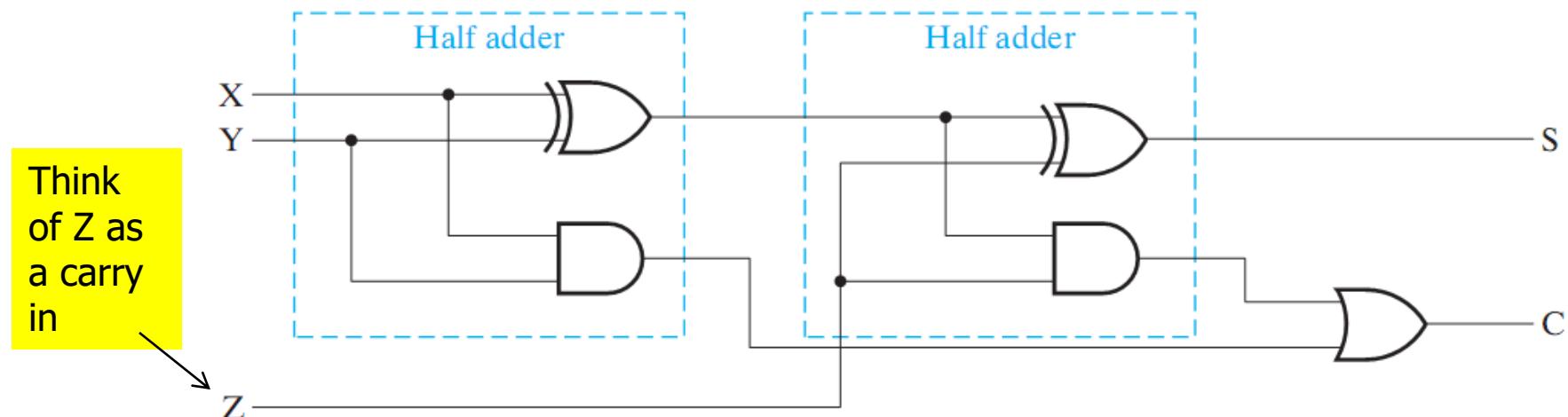
Fig. 4-7 Implementation of Full Adder in Sum of Products

Full Adder = 2 Half Adders

Manipulating the Equations:

$$S = (X \oplus Y) \oplus Z$$

$$C = XY + XZ + YZ = XY + Z(X \oplus Y)$$



Src: Mano's Book

Another implementation

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

$$\begin{aligned}
 S &= z \oplus (x \oplus y) \\
 &= z'(xy' + x'y) + z(xy' + x'y)' \\
 &= xy'z' + x'y'z + xyz + x'y'z
 \end{aligned}$$

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

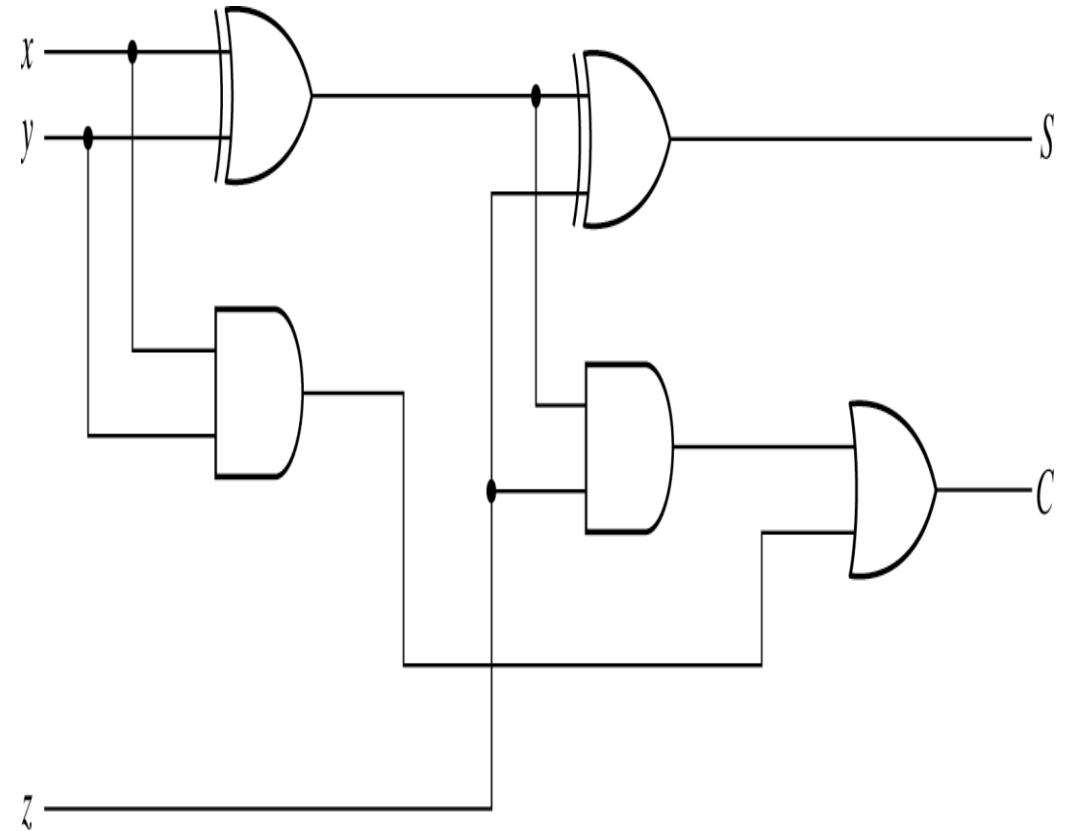


Fig. 4-8 Implementation of Full Adder with Two Half Adders and an OR Gate



Subtractor

- **Subtractor** : Subtractor is the one which used to subtract two binary number and provides **Difference** and **Borrower** as a output. Basically we have two types of subtractor.
- **Half Subtractor**
- **Full Subtractor**

The arithmetic operation, subtraction of two binary digits has four possible elementary operations, namely,

$$0 - 0 = 0$$

$$0 - 1 = 1 \quad (\text{with 1 borrow})$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Half Subtractor

- **Half Subtractor :** Half Subtractor is used for subtracting one single bit binary number from another single bit binary number(in general, subtraction of 2 bits). The truth table of Half Subtractor is shown below.

Like Adders, here also equation of ***Difference*** and ***Borrow*** is calculated

$$\text{Difference} = A'B + AB' = A \oplus B$$

$$\text{Borrow} = A'B$$

Half Subtractor-Truth Table			
Input		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor

As we know,

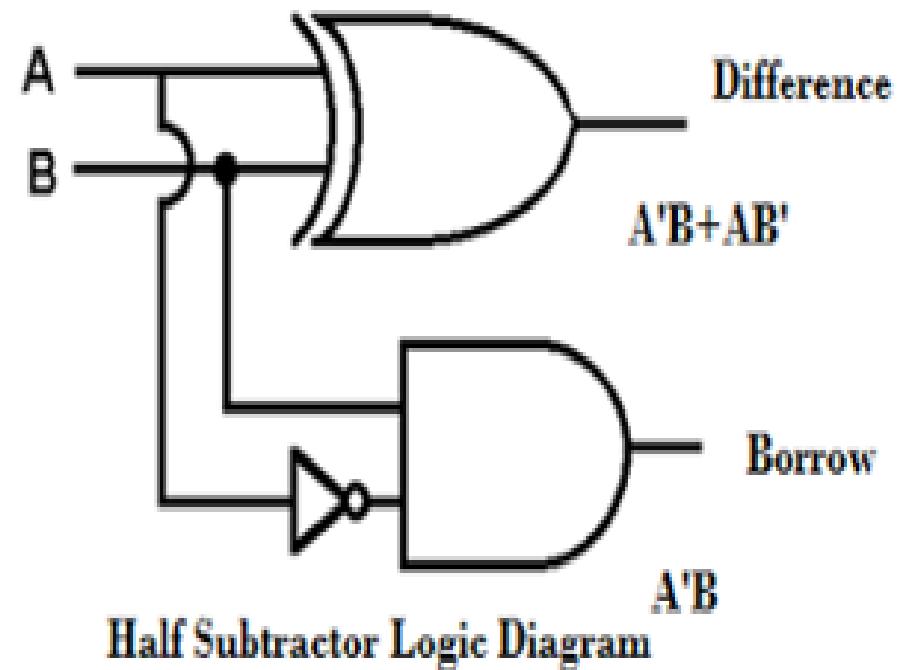
$$\text{Difference} = A'B + AB' = A \oplus B$$

$$\text{Borrow} = A'B$$

	$B'C'$	$B'C$	BC	BC'
A'	0	0	0	0
A	1	1	1	1

So, the **Circuit Diagram** using Karnaugh map is shown here

While, the **Logic Diagram** of Half Subtractor is



Full Subtractor

- **Full Subtractor :** A logic Circuit Which is used for Subtracting Three Single bit Binary numbers is known as Full Subtractor. The Truth Table of Full Subtractor is Shown Below.
- A full subtractor circuit can be implemented with **two half subtractors** and one **OR** gate.

Full Subtractor-Truth Table				
Input			Output	
A	B	C	Difference	Borrow
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...

Truth Table Explanation

Note: Borrow is set when the answer is negative

Difference is set when the magnitude of the difference is 1. as

$$0 - 0 - 1 = -1 \quad D=1, B=1$$

$$0 - 1 - 0 = -1$$

$$0 - 1 - 1 = -2 \quad D=0, B=1$$

$$1 - 0 - 0 = 1 \quad D=1, B=0$$

00 difference is 0

10 difference is a positive 1

11 difference is a negative 1

01 difference is negative 2

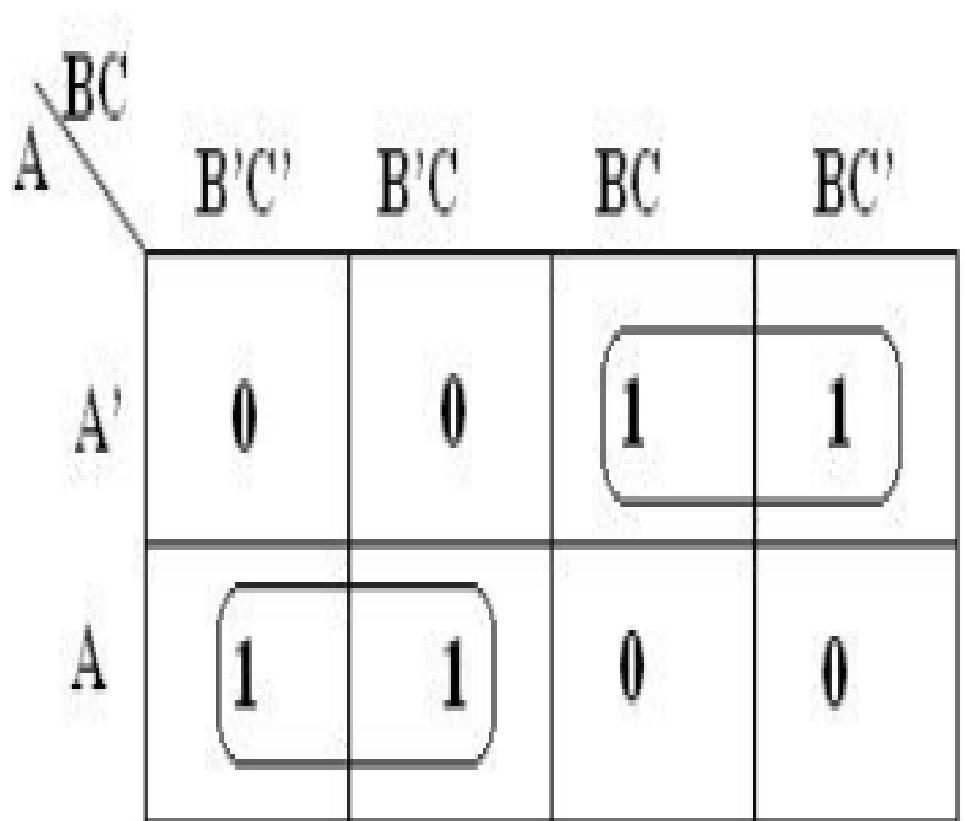
The only time you can get a **0 for difference** and a **1 for borrow** is when your answer is **-2**.

Full Subtractor

- Using Karnaugh maps the reduced expression for the output bits can be obtained as (*Circuit diagram of Difference*)
- Difference= $A'B'C+A'BC'+AB'C'+ABC$
- Reduce it like adder
- Then We got
- Difference= $A \oplus B \oplus C$

Borrow

$$\begin{aligned}
 &=A'B'C+A'BC'+A'BC+ABC \\
 &=A'B'C+A'BC'+A'BC+A'BC+A'BC+ABC \\
 &\xrightarrow{\text{-----}} A'BC=A'BC+A'BC+A'BC \\
 &=A'C(B'+B)+A'B(C'+C)+BC(A'+A)
 \end{aligned}$$



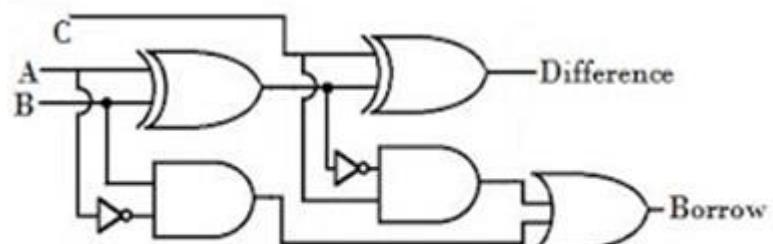
Full Subtractor

- Using Karnaugh maps the reduced expression for the output bits can be obtained as (*Circuit diagram of Borrow*)

$$\text{Borrow} = A'C + A'B + BC$$

	$B'C'$	$B'C$	BC	BC'
A'	0	1	0	1
A	0	1	0	1

While, the logic diagram of Full subtractor is here
(2-half-sub and one OR gate)



Full Subtractor-Logic Diagram

Binary adder

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

<i>Subscript i:</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}

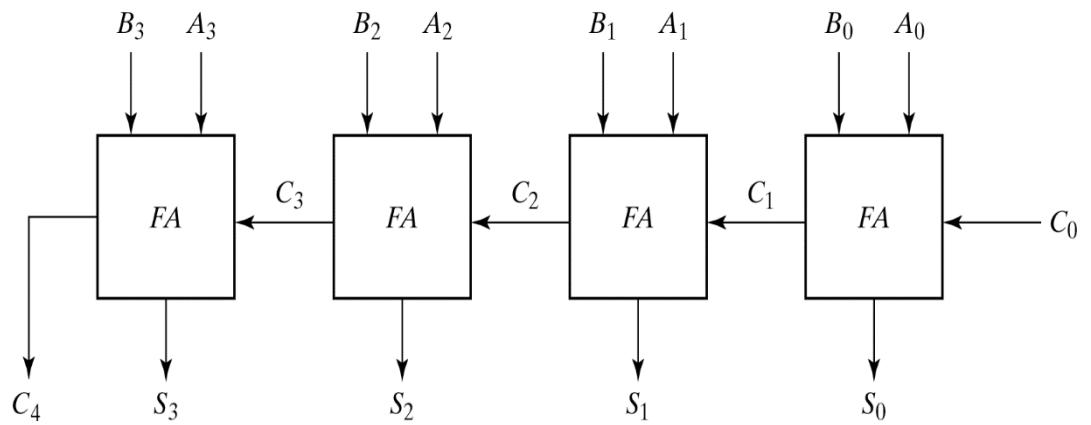


Fig. 4-9 4-Bit Adder

Carry Propagation

- Fig.4-9 causes an unstable factor on carry bit, and produces a longest propagation delay.
- The signal from C_i to the output carry C_{i+1} , propagates through an AND and OR gates, so, for an n-bit RCA, there are $2n$ gate levels for the carry to propagate from input to output.

<i>Subscript i:</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
	<hr/>				
Sum	1	1	1	0	S_i
Output carry	0	0	1	1	C_{i+1}

Carry Propagation

- Because the propagation delay will affect the output signals on different time, so the signals are given enough time to get the precise and stable outputs.
- The most widely used technique employs the principle of carry look-ahead to improve the speed of the algorithm.

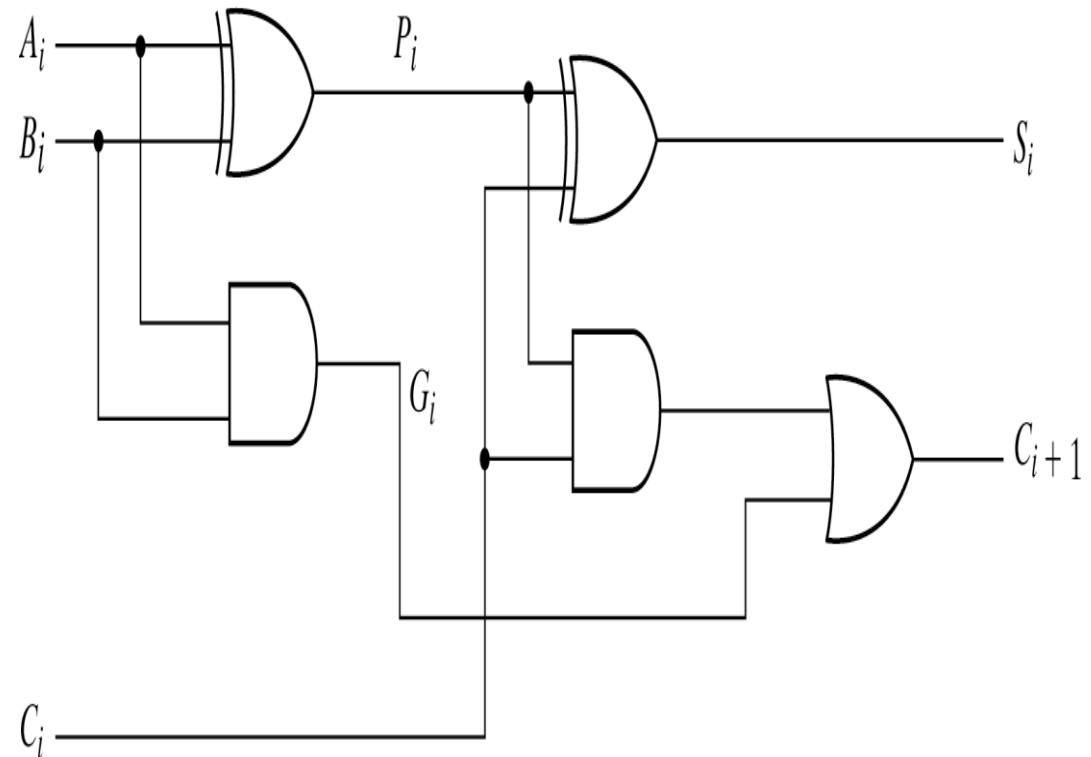


Fig. 4-10 Full Adder with P and G Shown

Boolean functions

$$P_i = A_i \oplus B_i \quad \text{steady state value}$$

$$G_i = A_i B_i \quad \text{steady state value}$$

Output sum and carry

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

G_i : carry generate P_i : carry propagate

C_0 = input carry

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

- C_3 does not have to wait for C_2 and C_1 to propagate.

Logic diagram of carry look-ahead generator



- C_3 is propagated at the same time as C_2 and C_1 .

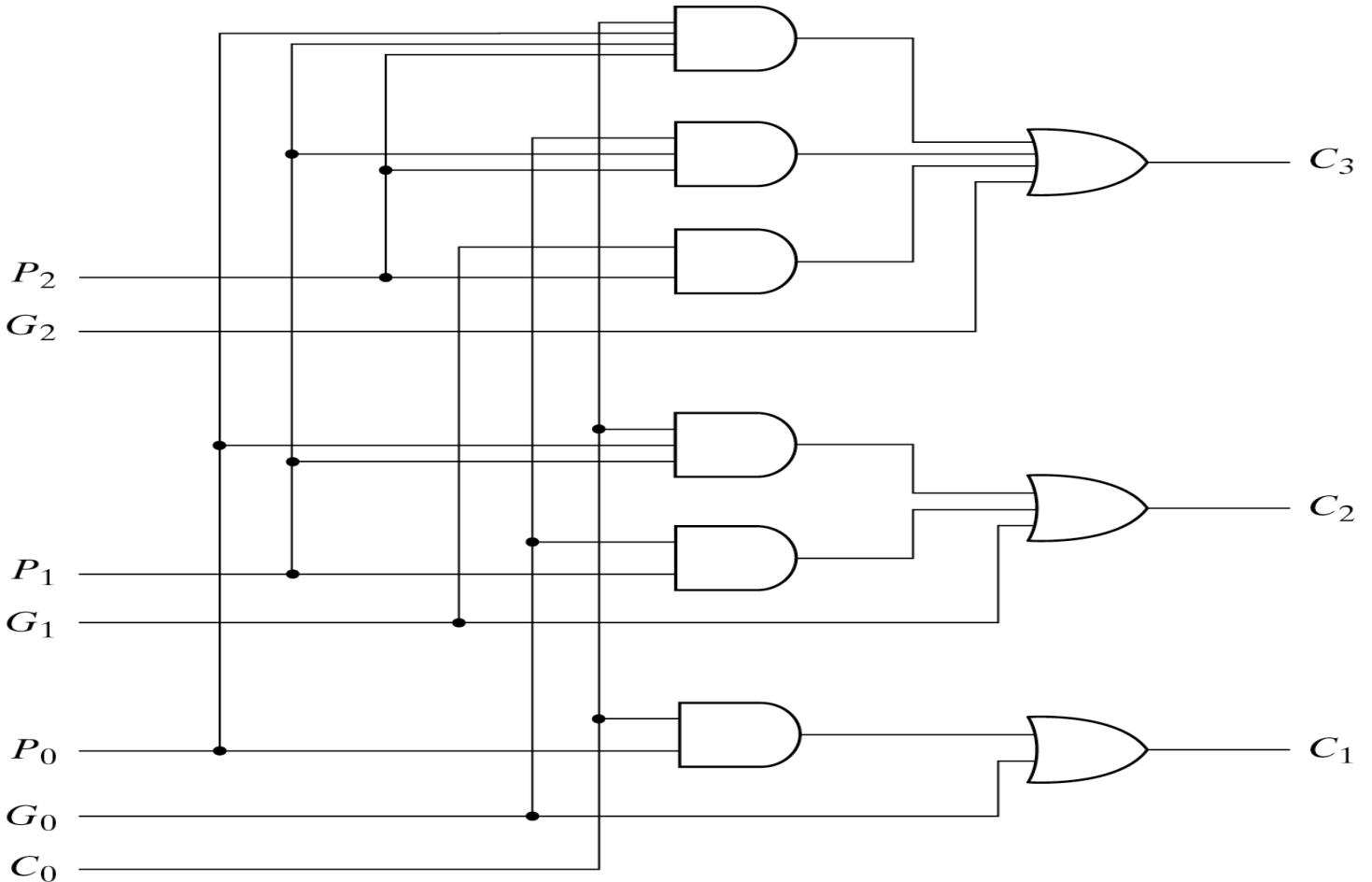


Fig. 4-11 Logic Diagram of Carry Lookahead Generator

4-bit adder with carry lookahead

- Delay time of n-bit CLAA = XOR + (AND + OR) + XOR

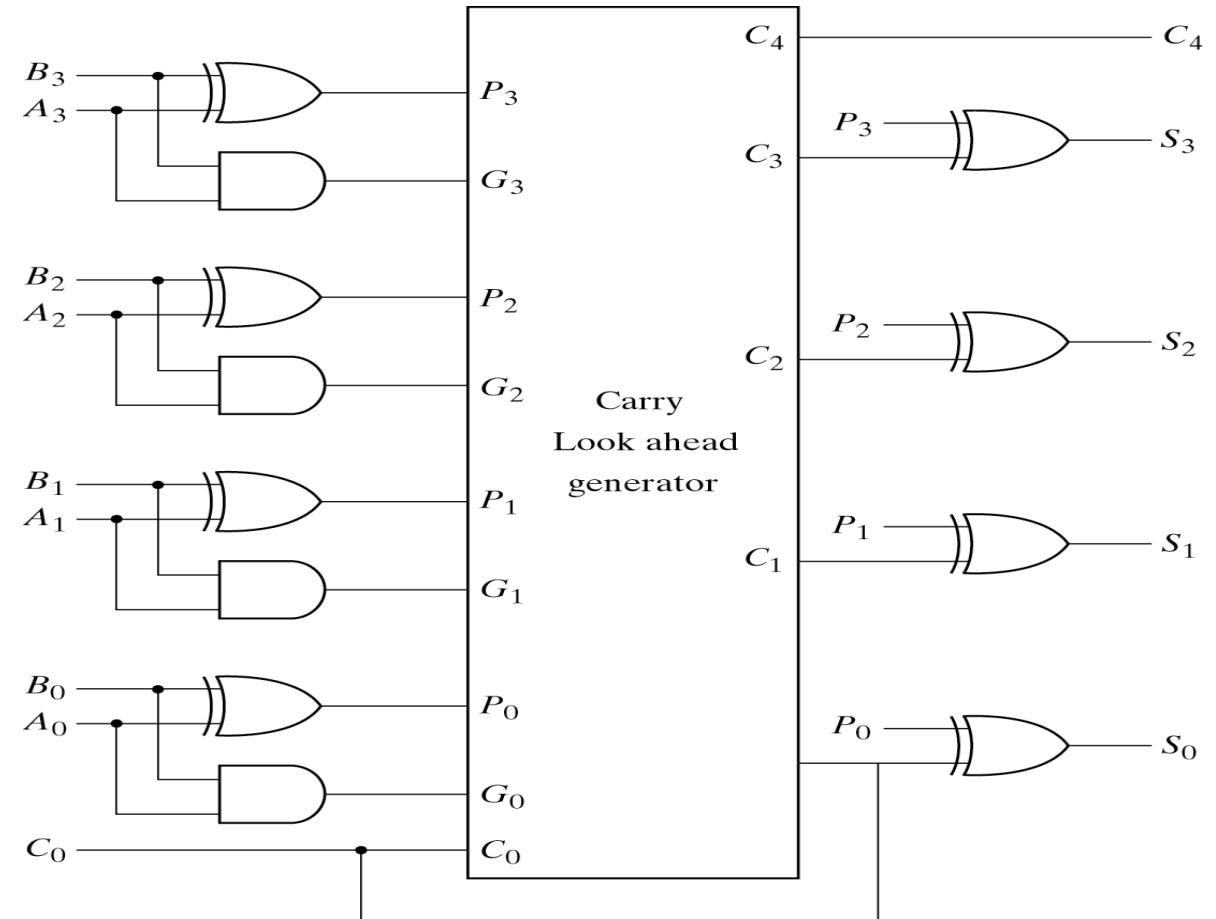


Fig. 4-12 4-Bit Adder with Carry Lookahead

Binary subtractor

$M = 1 \rightarrow \text{subtractor} ; M = 0 \rightarrow \text{adder}$

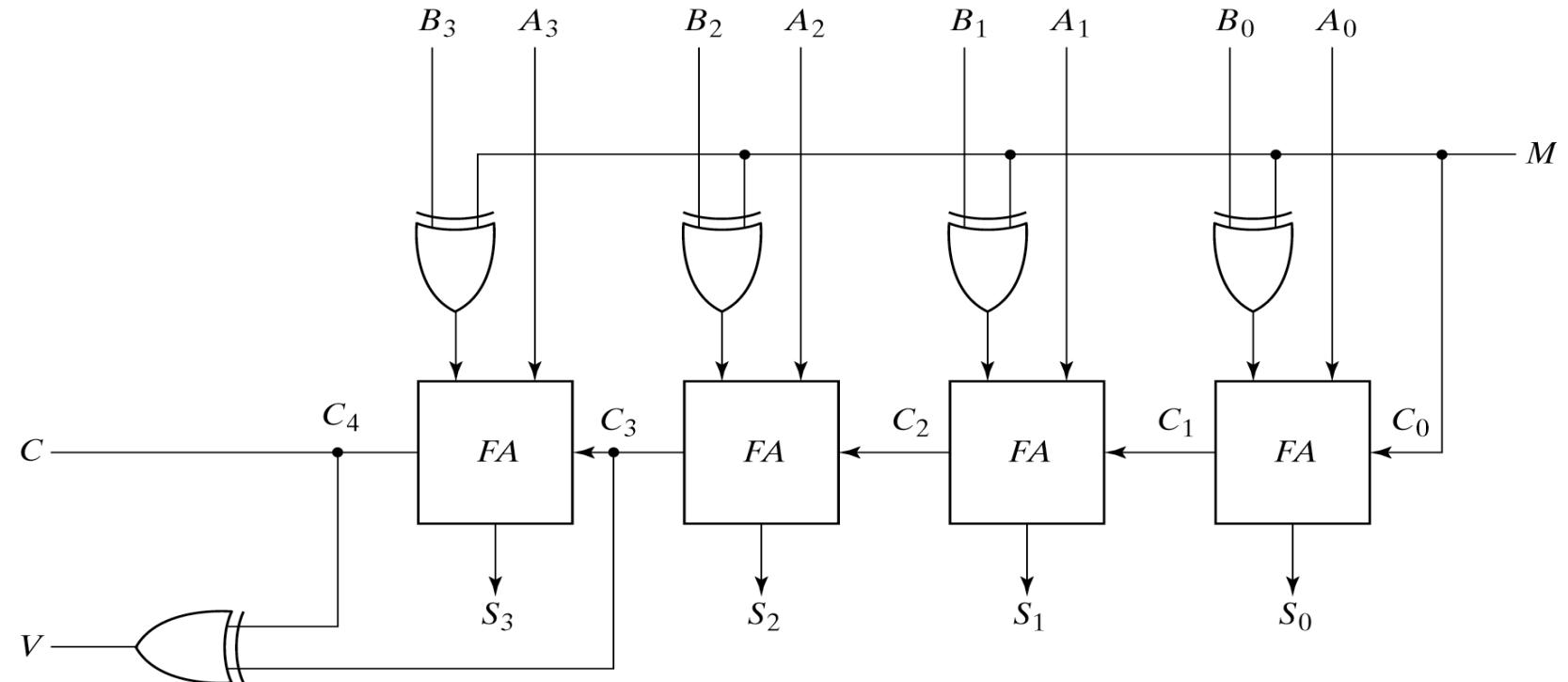


Fig. 4-13 4-Bit Adder Subtractor



Overflow

- It is **worth** noting Fig.4-13 that binary numbers in the **signed-complement system** are added and subtracted by the same basic addition and subtraction rules **as unsigned numbers**.
- Overflow is a problem in digital computers because the number of bits that hold the number is finite and a result that contains $n+1$ bits cannot be accommodated.



Overflow on signed and unsigned

- When two **unsigned** numbers are added, an overflow is detected from the **end carry out of the MSB position**.
- When two **signed** numbers are added, the sign bit is treated as part of the number and the end carry does not indicate an overflow.
- An **overflow can't occur** after an addition if one number is **positive** and the other is **negative**.
- An overflow may occur if the two numbers added are both positive or both negative.

4-5 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



Jiangxi University of Science and Technology

Some IC information

Wiring Diagrams

TTL Family

TTL Series	Prefix	Example IC
Standard TTL	74	7404 (hex inverter)
Schottky TTL	74S	74S04
Low-power Schottky TTL	74LS	74LS04
Advanced Schottky TTL	74AS	74AS04
Advanced low-power Schottky TTL	74ALS	74ALS04

CMOS Family

CMOS Series	Prefix	Example IC
Metal-gate CMOS	40	4001
Metal-gate, pin-compatible with TTL	74C	74C02
Silicon-gate, pin-compatible with TTL, high-speed	74HC	74HC02
Silicon-gate, high-speed, pin-compatible and electrically compatible with TTL	74HCT	74HCT02
Advanced-performance CMOS, not pin or electrically compatible with TTL	74AC	74AC02
Advanced-performance CMOS, not pin but electrically compatible with TTL	74ACT	74ACT02



Power and Ground

- To use digital IC, it is necessary to make proper connection to the IC pins.
- Power: labeled V_{cc} for the TTL circuit, labeled V_{DD} for CMOS circuit.
- Ground



Logic-level Voltage Ranges

- For TTL devices, V_{CC} is normally 5V.
- For CMOS circuits, V_{DD} can range from 3-18V.
- For TTL, logic 0 : 0-0.8V, logic 1:2-5V
- For CMOS, logic 0 : 0-1.5V, logic 1:3.5-5V

8-3 TTL Data Sheets

Data sheet for the 74ALS00 NAND gate IC



electrical characteristics over recommended operating free-air temperature range unless otherwise noted

PARAMETER	TEST CONDITIONS	SN54ALS00A			SN74ALS00A			UNIT
		MIN	TYPT†	MAX	MIN	TYPT†	MAX	
V _{IK}	V _{CC} = 4.5 V, I _I = -18 mA			-1.2			-1.5	V
V _{OH}	V _{CC} = 4.5 V to 5.5 V, I _{OH} = -0.4 mA	V _{CC} - 2			V _{CC} - 2			V
V _{OL}	V _{CC} = 4.5 V	I _{OL} = 4 mA		0.25	0.4	0.25	0.4	V
		I _{OL} = 8 mA				0.35	0.5	
I _I	V _{CC} = 5.5 V, V _I = 7 V			0.1		0.1		mA
I _{IH}	V _{CC} = 5.5 V, V _I = 2.7 V			20		20		μA
I _{IL}	V _{CC} = 5.5 V, V _I = 0.4 V			-0.1		-0.1		mA
I _O ‡	V _{CC} = 5.5 V, V _O = 2.25 V	-20	-112	-30	-112			mA
I _{CCH}	V _{CC} = 5.5 V, V _I = 0		0.5	0.85	0.5	0.85		mA
I _{CCL}	V _{CC} = 5.5 V, V _I = 4.5 V		1.5	3	1.5	3		mA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current, I_{OS}.

	Minimum	Typical	Maximum
V _{OL} (V)	—	0.35	0.5
V _{OH} (V)	2.5	3.4	—
V _{IL} (V)	—	—	0.8
V _{IH} (V)	2.0	—	—

**74ALS series
voltage levels.**

8-4 TTL Series Characteristics



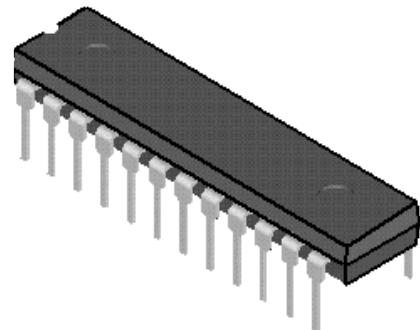
Typical TTL series characteristics.

	74	74S	74LS	74AS	74ALS	74F
Performance ratings						
Propagation delay (ns)	9	3	9.5	1.7	4	3
Power dissipation (mW)	10	20	2	8	1.2	6
Max. clock rate (MHz)	35	125	45	200	70	100
Fan-out (same series)	10	20	20	40	20	33
Voltage parameters						
$V_{OH}(\text{min})$ (V)	2.4	2.7	2.7	2.5	2.5	2.5
$V_{OL}(\text{max})$ (V)	0.4	0.5	0.5	0.5	0.5	0.5
$V_{IH}(\text{min})$ (V)	2.0	2.0	2.0	2.0	2.0	2.0
$V_{IL}(\text{max})$ (V)	0.8	0.8	0.8	0.8	0.8	0.8

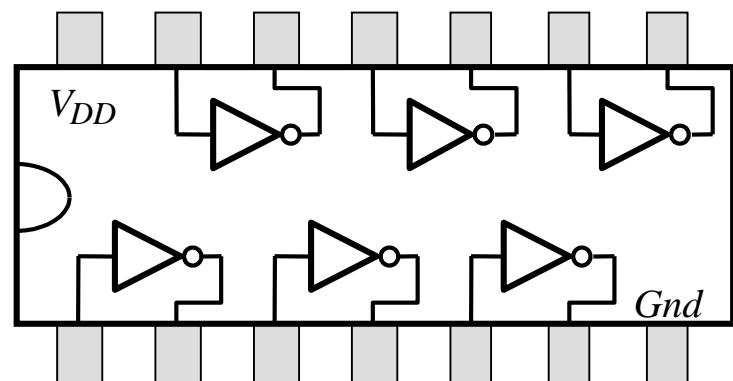
TTL vs CMOS

Parameter	CMOS							TTL			
	4000B	74HC	74HCT	74AC	74ACT	74AHC	74AHCT	74	74LS	74AS	74ALS
$V_{IH}(\text{min})$	3.5	3.5	2.0	3.5	2.0	3.85	2.0	2.0	2.0	2.0	2.0
$V_{IL}(\text{max})$	1.5	1.0	0.8	1.5	0.8	1.65	0.8	0.8	0.8	0.8	0.8
$V_{OH}(\text{min})$	4.95	4.9	4.9	4.9	4.9	4.4	3.15	2.4	2.7	2.7	2.5
$V_{OL}(\text{max})$	0.05	0.1	0.1	0.1	0.1	0.44	0.1	0.4	0.5	0.5	0.5
V_{NH}	1.45	1.4	2.9	1.4	2.9	0.55	1.15	0.4	0.7	0.7	0.7
V_{NL}	1.45	0.9	0.7	1.4	0.7	1.21	0.7	0.4	0.3	0.3	0.4

Standard Logic Gate: Pin-out



(a) Dual-inline package

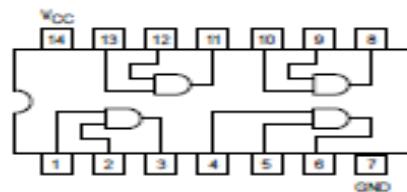


(b) Structure of 7404 chip

Standard Logic Gate: Pin-out



QUAD 2-INPUT AND GATE



SN54/74LS08

QUAD 2-INPUT AND GATE
LOW POWER SCHOTTKY



ORDERING INFORMATION

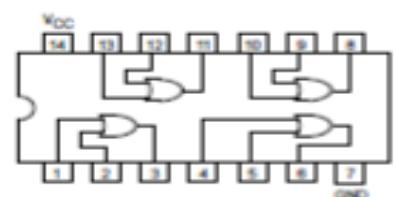
SN54LS08J Ceramic
SN74LS08N Plastic
SN74LS08D SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter	54	Min	Typ	Max	Unit
V _{CC}	Supply Voltage	74	4.5	4.75	5.0	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA



QUAD 2-INPUT OR GATE



SN54/74LS32

QUAD 2-INPUT OR GATE
LOW POWER SCHOTTKY



ORDERING INFORMATION

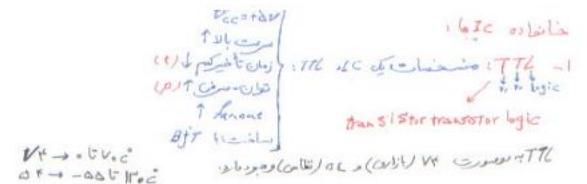
SN54LS32J Ceramic
SN74LS32N Plastic
SN74LS32D SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter	54	Min	Typ	Max	Unit
V _{CC}	Supply Voltage	74	4.5	4.75	5.0	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA



Standard Logic Gate: Pin-out



t	P	
medium	medium	V ₄₀₀ انواع استاندارد
High	Low	V ₄₀₀ + نوع كوميون
L	H	V ₄₀₀ + نوع سالك
L	m ⁺	V ₄₀₀ + نوع ترانزستور
L	m	V ₄₀₀ + نوع ميغافيت
L	L	V ₄₀₀ + نوع ميجا فلتر

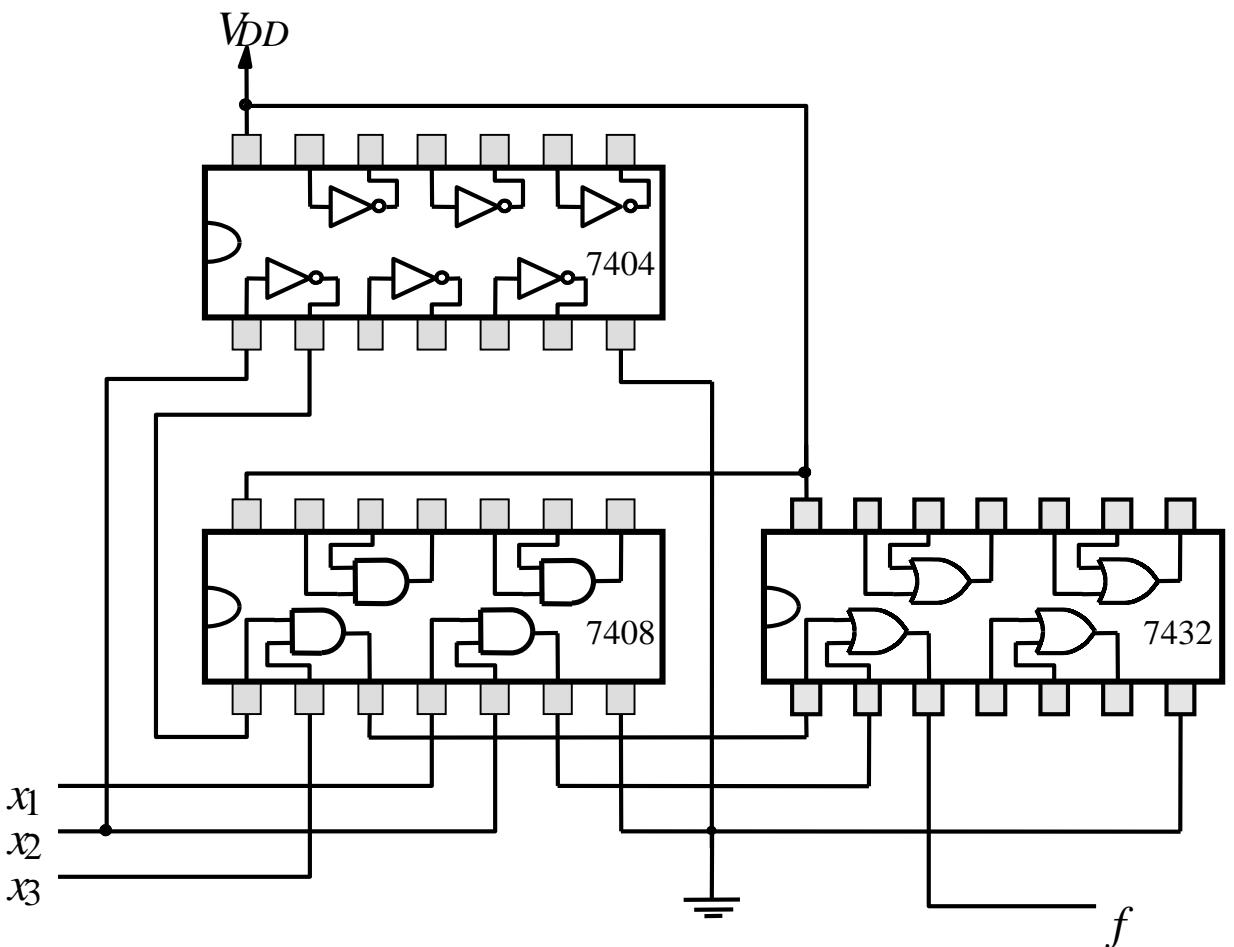
Nand	L = V ₄₀₀
Nor	L = V ₄₀₀
Not	L = V ₄₀₀
And	L = V ₄₀₀
Or	L = V ₄₀₀
Xor	L = V ₄₀₀

MOSFET	نوع ميكرو إلكتروني
V _{CC} = +5VDC	نوع
Inputs	Cmos
Outputs	Cmos
Noise	Cmos
V ₄₀₀	- استاندارد
V ₄₀₀	+ ارجاع بايدج
V ₄₀₀	- ارجاع بايدج
V ₄₀₀	+ ارجاع بايدج

Example

Draw the circuit diagram and the wiring diagram for the following Boolean expression:

$$F = x'_2 \cdot x_3 + x_1 \cdot x_2$$



Reference

