University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

Lec. Dr. Taif Alawsi

Lec. 1: Digital Concepts: 2024-Sep-22

Lecture Outline

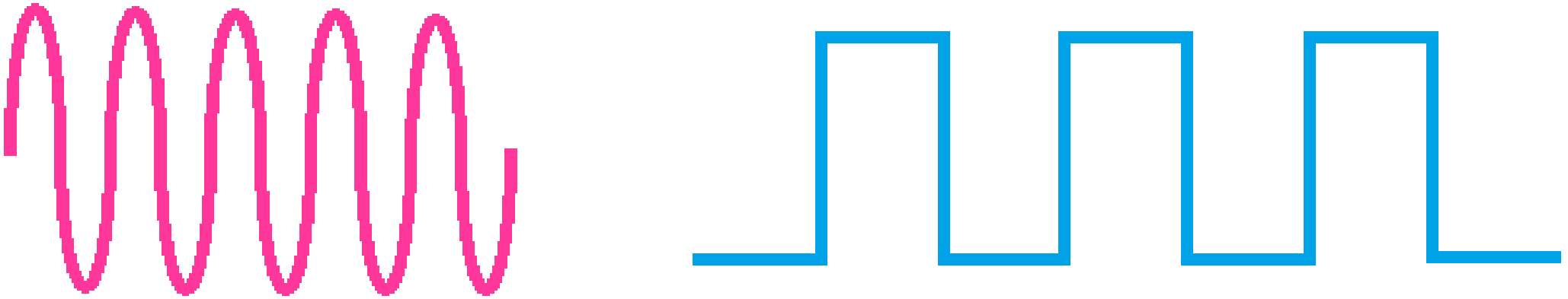
1. Aim of the course
2. Number System (Decimal VS Binary)
3. Counting in Binary
4. Conversions
5. HW

Aim of the Course

Study the following general topics

1. Number Systems
2. Arithmetic Operations
3. Logic Gates
4. Boolean Algebra
5. Logic Simplification
6. Combinational Logic Analysis
7. Function of combinational logic
8. Latches, Flip-Flops, Timers
9. Counters
10. Shift Registers
11. Memory

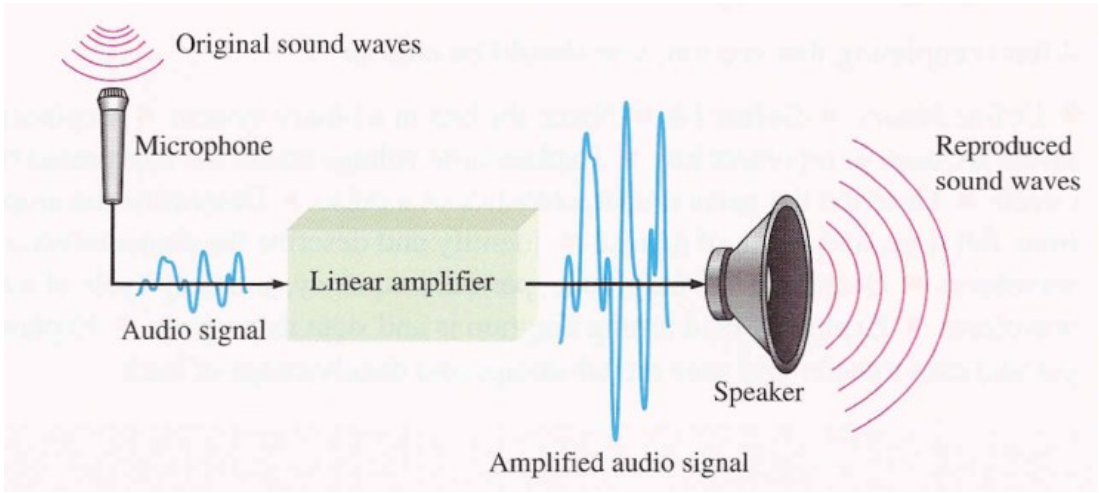
Analog VS Digital Signal

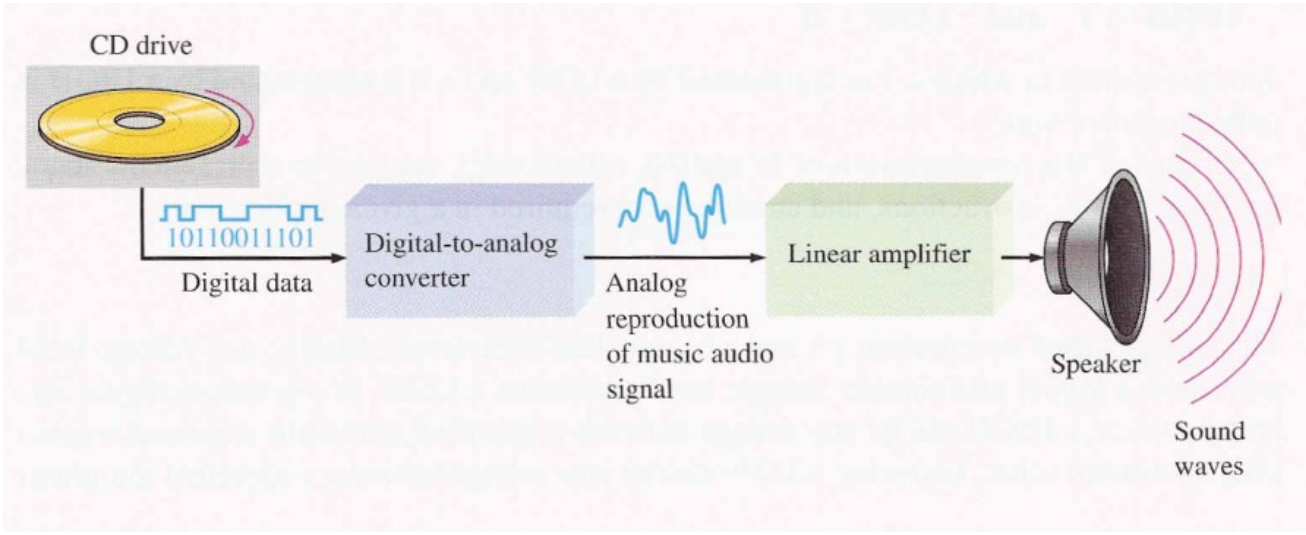


Analog Signal **Digital Signal**

Types of electronic devices or instruments:

- Analog
- Digital
- Combination analog and digital



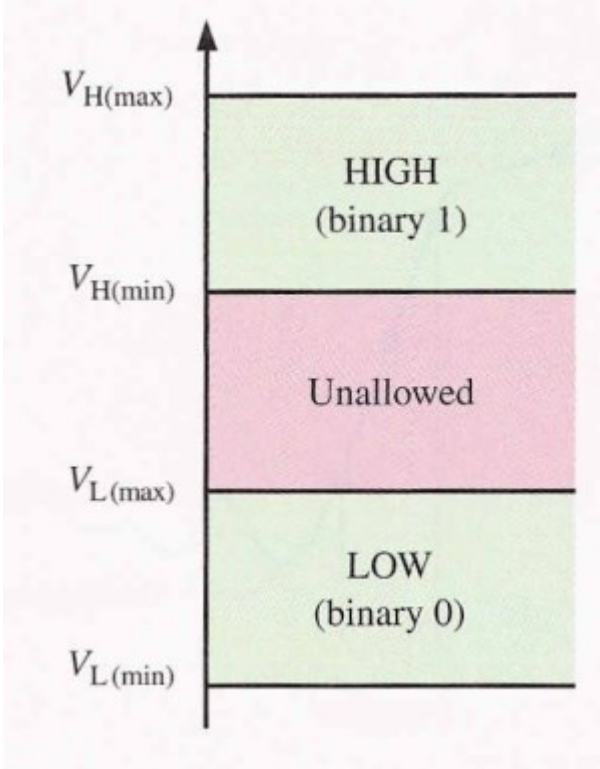


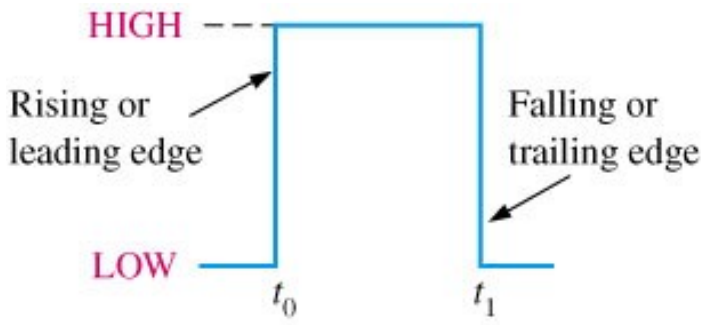
A. The conventional numbering system uses ten digits: 0,1,2,3,4,5,6,7,8, and 9.

B. The binary numbering system uses just two digits: **0** and **1.**

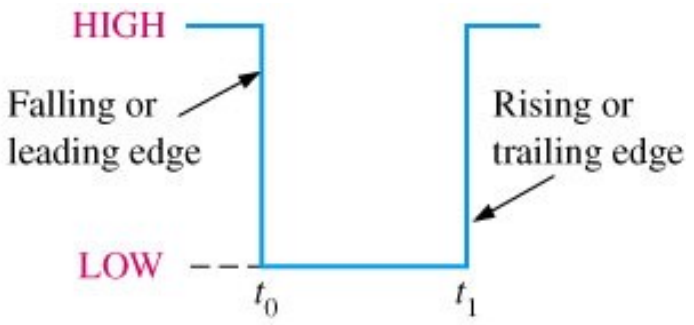
C. The two binary digits are designated **0** and **1**

D. They can also be called LOW and HIGH, where **LOW = 0** and **HIGH = 1**



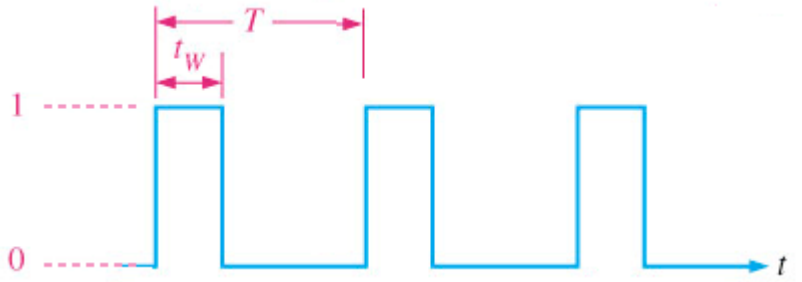


(a) Positive–going pulse



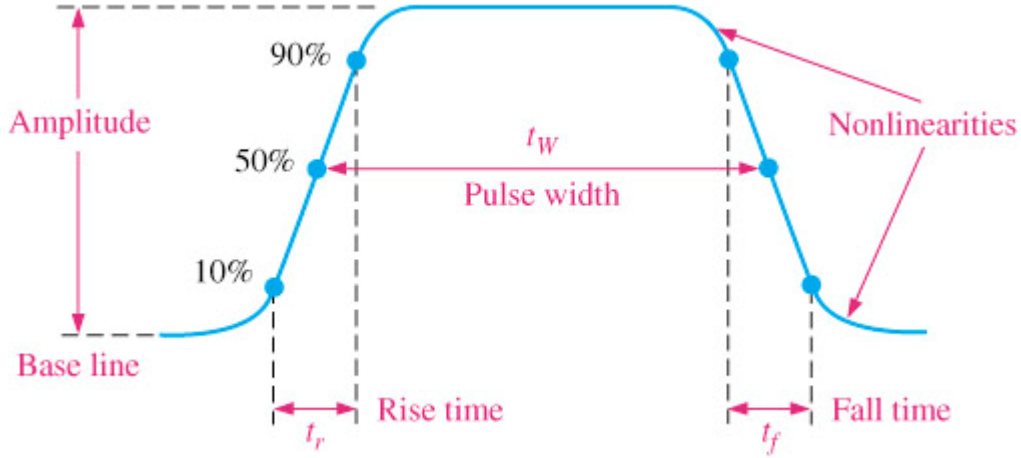
(b) Negative–going pulse

Digital Pulse



Major parts of a digital pulse

1. Base line
2. Amplitude
3. Rise time (t_r)
4. Pulse width (t_w)
5. Fall time (t_f)



$$f = \frac{1}{T}$$

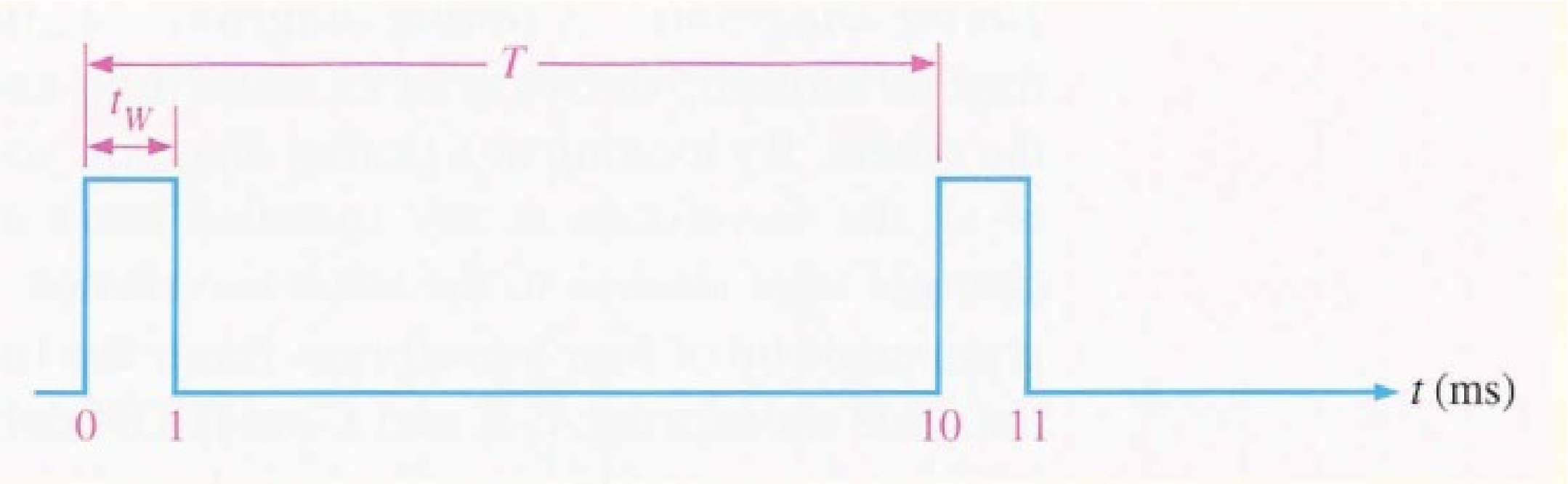
$$\text{Duty cycle} = \left(\frac{t_w}{T}\right)100\%$$

t_w = pulse width

T = period of the waveform

f = frequency of the waveform

Example 1



$$T = 10\ mS$$

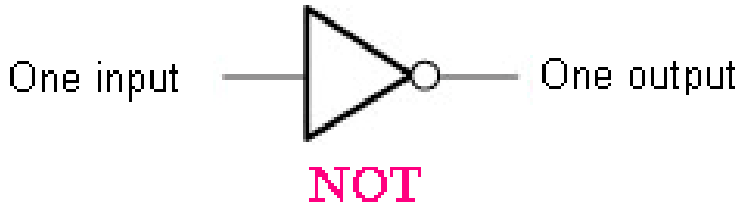
$$f = \frac{1}{T} = 100\ Hz$$

$$\text{Duty cycle} = \left(\frac{t_w}{T}\right)100\%$$

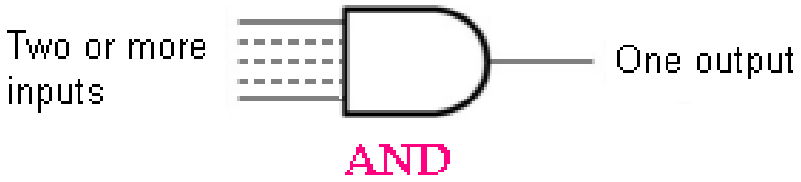
$$\text{Duty cycle} = \left(\frac{1\ ms}{10\ ms}\right)100\% = 10\%$$

Logic Gates

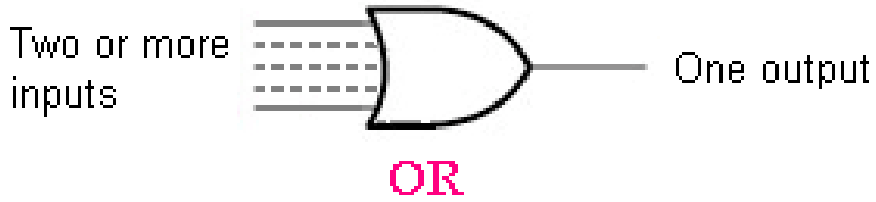
There are only three basic logic operations:



NOT ($\bar{\ }$)



AND (x)

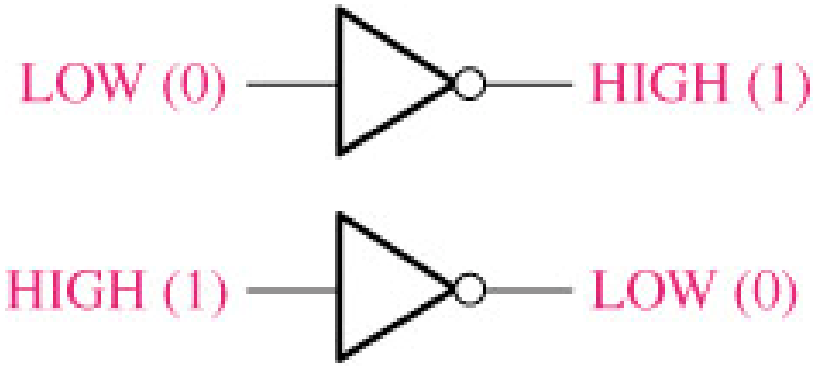


OR (+)

NOT Gate Operation

input is LOW, output is HIGH

input is HIGH, output is LOW

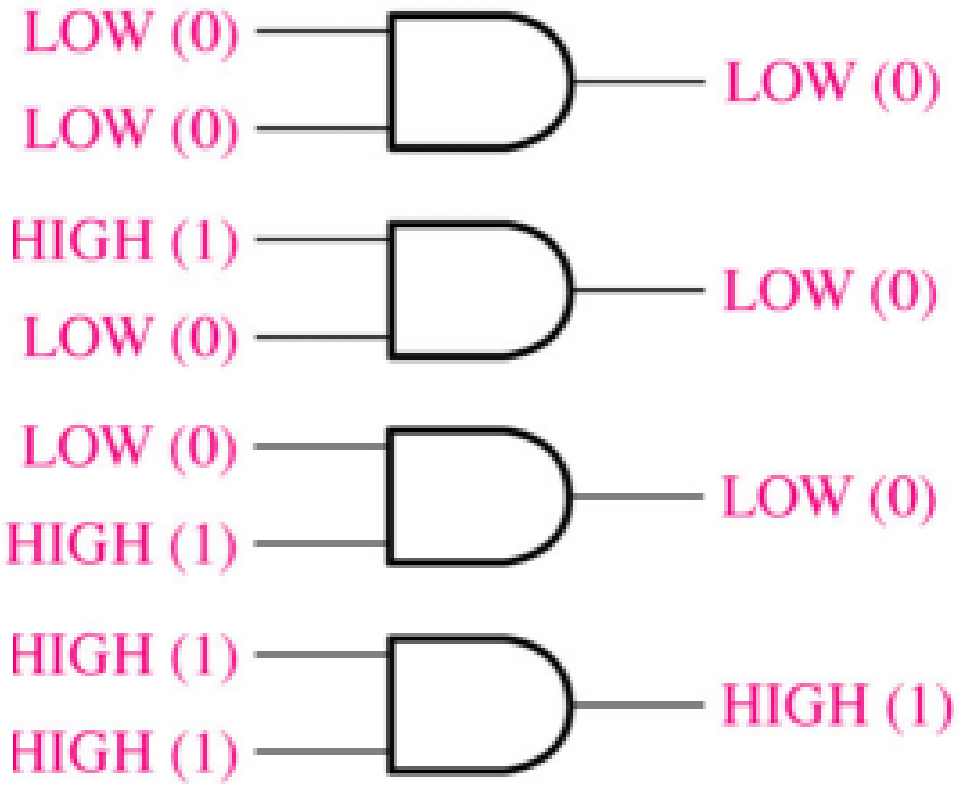


The output logic level is always opposite the input logic level.

AND Gate Operation

When any input is LOW, the output is LOW

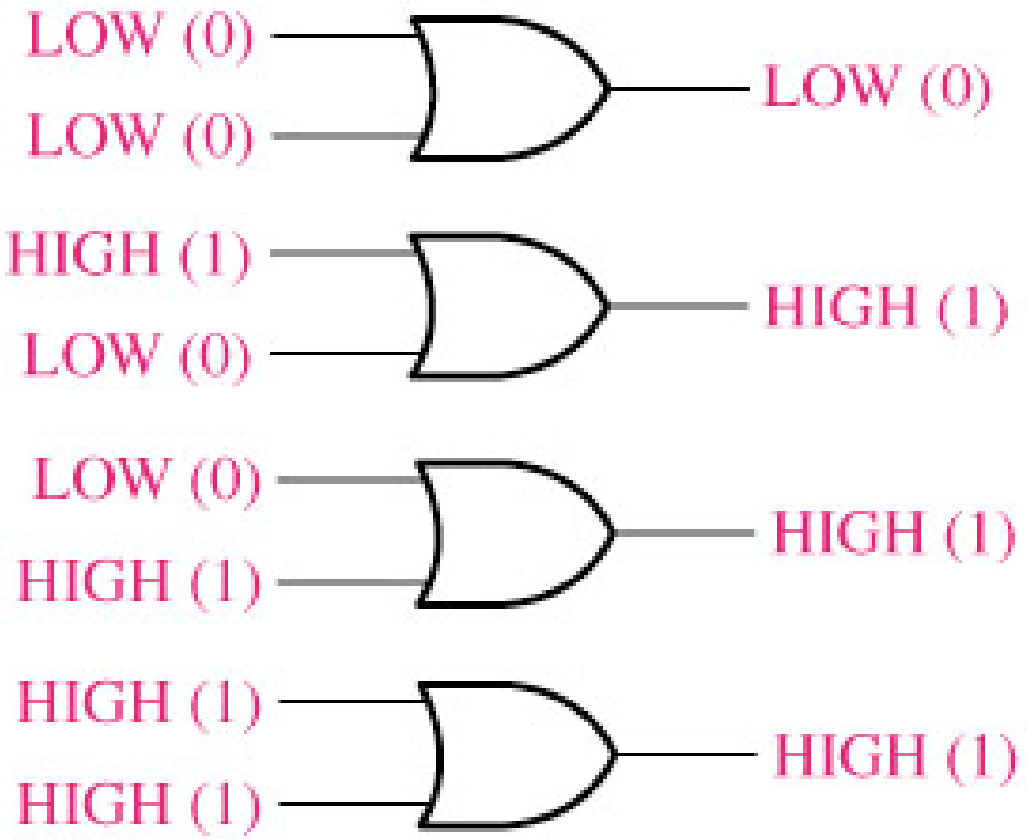
When both inputs are HIGH, the output is HIGH



OR Gate Operation

When any input is HIGH, the output is HIGH

When both inputs are LOW, the output is LOW

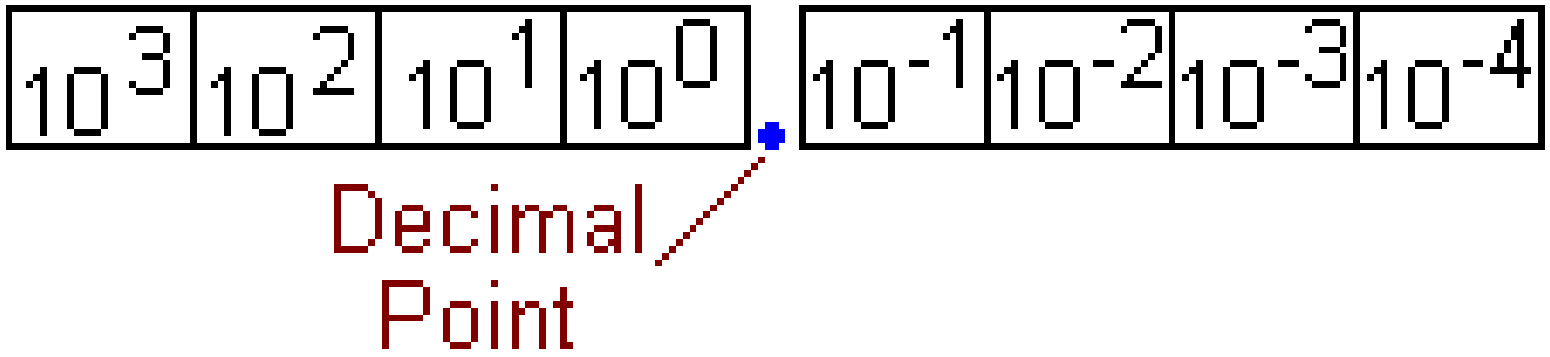


Decimal Numbers

The decimal number system has ten digits:

0, 1, 2, 3, 4, 5, 6, 7, 8, and 9

The decimal numbering system has a base of 10 with each position weighted by a factor of 10:



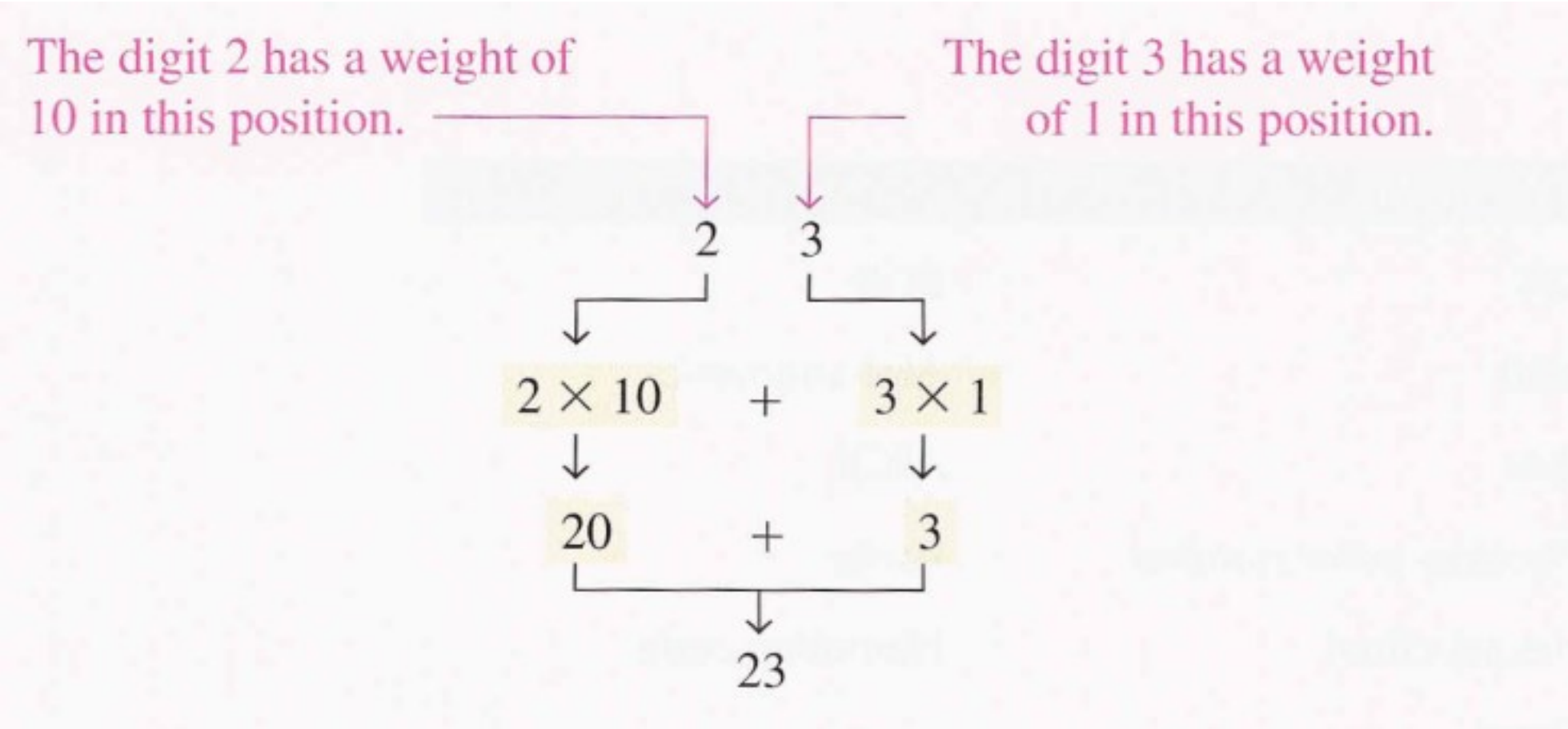
Binary Numbers

The binary number system has two digits:
0 and 1

The binary numbering system has a base of 2 with each position weighted by a factor of 2:

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

Sum of Weight



Conversions

Convert the binary whole number 1101101 to decimal.

Determine the weight of each bit that is a 1, and then find the sum of the weights to get the decimal number.

$$\text{Weight: } 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$

$$\text{Binary number: } 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1$$

$$1101101 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0$$

$$= 64 + 32 + 8 + 4 + 1 = \mathbf{109}$$

Conversions

Convert the fractional binary number 0.1011 to decimal.

Determine the weight of each bit that is a 1, and then sum the weights to get the decimal fraction.

Weight:		2^{-1}	2^{-2}	2^{-3}	2^{-4}
Binary number:	0 .	1	0	1	1

$$0.1011 = 2^{-1} + 2^{-3} + 2^{-4}$$
$$= 0.5 + 0.125 + 0.0625 = \mathbf{0.6875}$$

Conversions

Convert the following decimal numbers to binary:

(a) 12 **(b)** 25 **(c)** 58 **(d)** 82

(a) $12 = 8 + 4 = 2^3 + 2^2 \longrightarrow$ **1100**

(b) $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 \longrightarrow$ **11001**

(c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 \longrightarrow$ **111010**

(d) $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1 \longrightarrow$ **1010010**

Addition in Binary Rules

A	$0 + 0 = 0$	Sum of 0 with a carry of 0
B	$0 + 1 = 1$	Sum of 1 with a carry of 0
C	$1 + 0 = 1$	Sum of 1 with a carry of 0
D	$1 + 1 = 10$	Sum of 0 with a carry of 1
E	$1+1+1 = 11$	Sum of 1 with a carry of 1

Addition in Binary Examples

Add the following binary numbers:

(a) 11 + 11 **(b)** 100 + 10 **(c)** 111 + 11 **(d)** 110 + 100

The equivalent decimal addition is also shown for reference.

(a)	11	3	**(b)**	100	4	**(c)**	111	7	**(d)**	110	6
	+11	+3		+10	+2		+ 11	+3		+100	+4
	110	6		**110**	6		**1010**	10		**1010**	10

Subtraction in Binary Rules

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1 \qquad 0 - 1 \text{ with a borrow of } 1$$

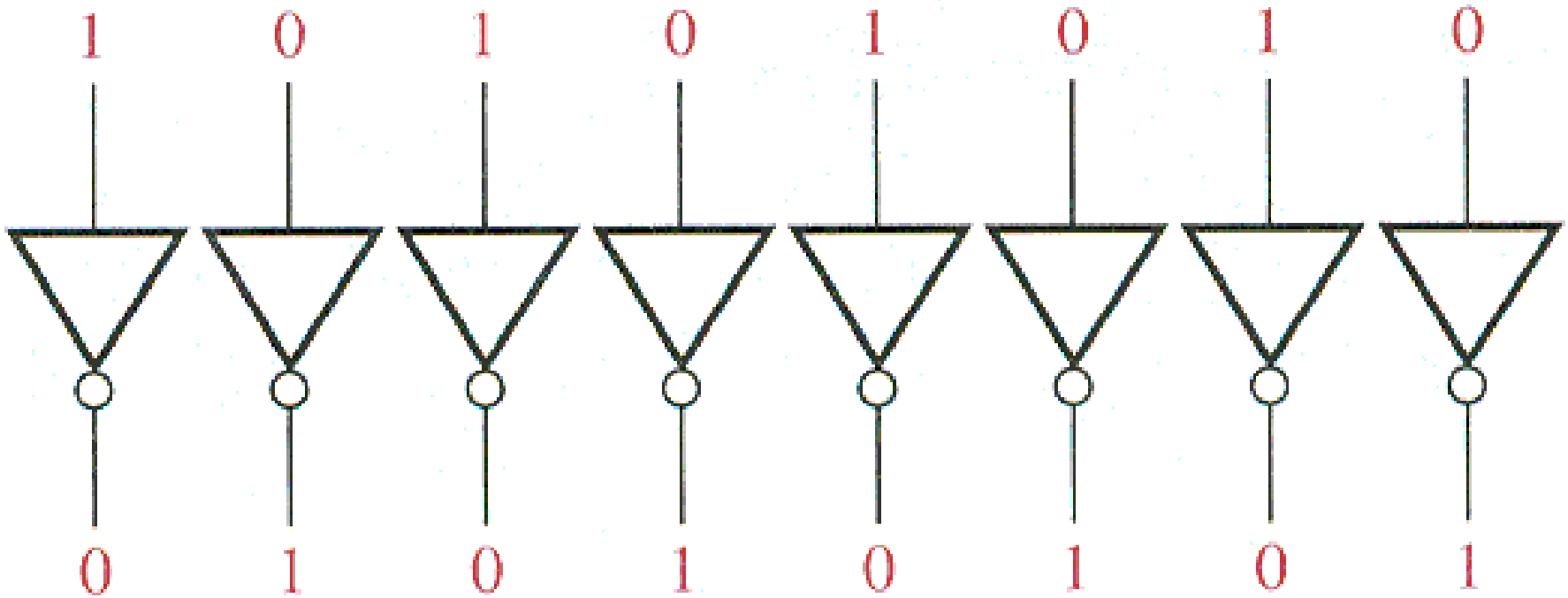
Subtraction in Binary Example

Perform the following binary subtractions:

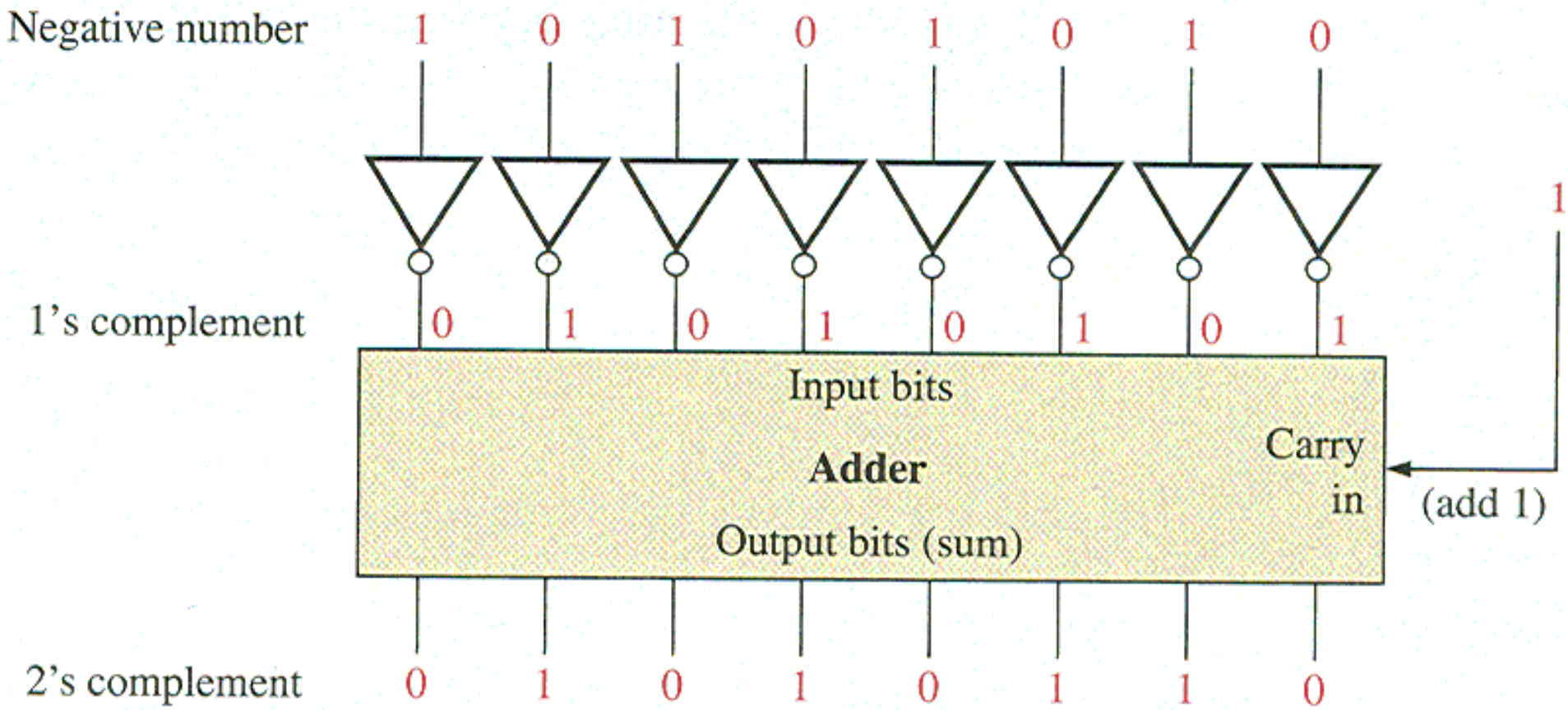
(a) $11 - 01$ **(b)** $11 - 10$

(a)	11		3	**(b)**	11		3
	− 01		− 1		− 10		− 2
	10		2		**01**		1

1’s complement



2’s complement

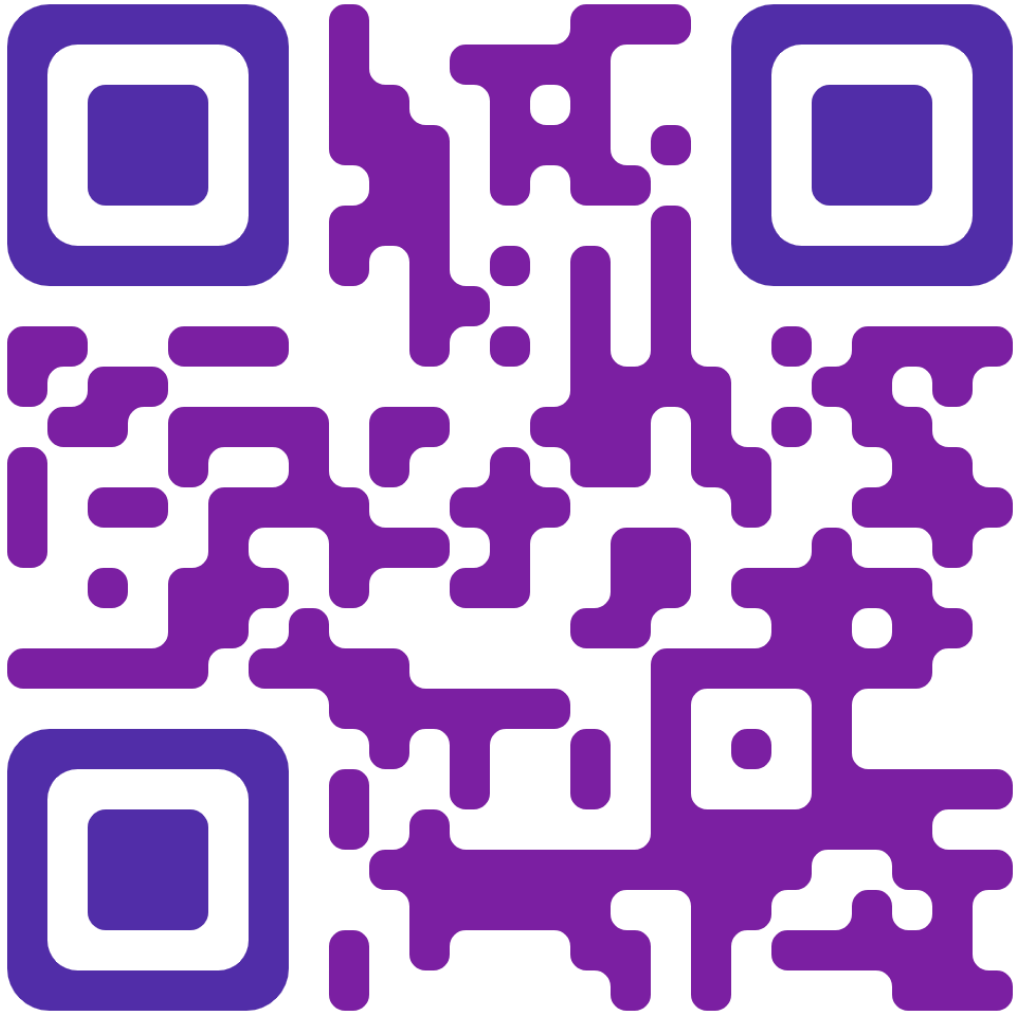


Home Work (Due date 08-10-2024)

1. Prepare a report on Digital gates applications in Laser Engineering (3 Pages)
2. Solve the following questions
   A. Convert the number 1011.10011 to decimal
   B. Convert the number 11 to binary
   C. Add $10_2 + 100_{10}$ and write the result both in binary and decimal
   D. Subtract $1010_2 - 10_{10}$ and write the result both in binary and decimal

HW Form: Scan QR

References: Scan QR



Telegram Group: Scan QR

University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

Lec. Dr. Taif Alawsi

Lec. 2: Number System Arithmetic : 2024-Oct-06

Lecture Outline

1. Decimal to binary conversion methods
2. Binary Multiplication and Division
3. Signed Binary Numbers
4. Hexadecimal Numbers: Conversion and Arithmetic
5. Octal Numbers
6. HW

Decimal and Binary Representation

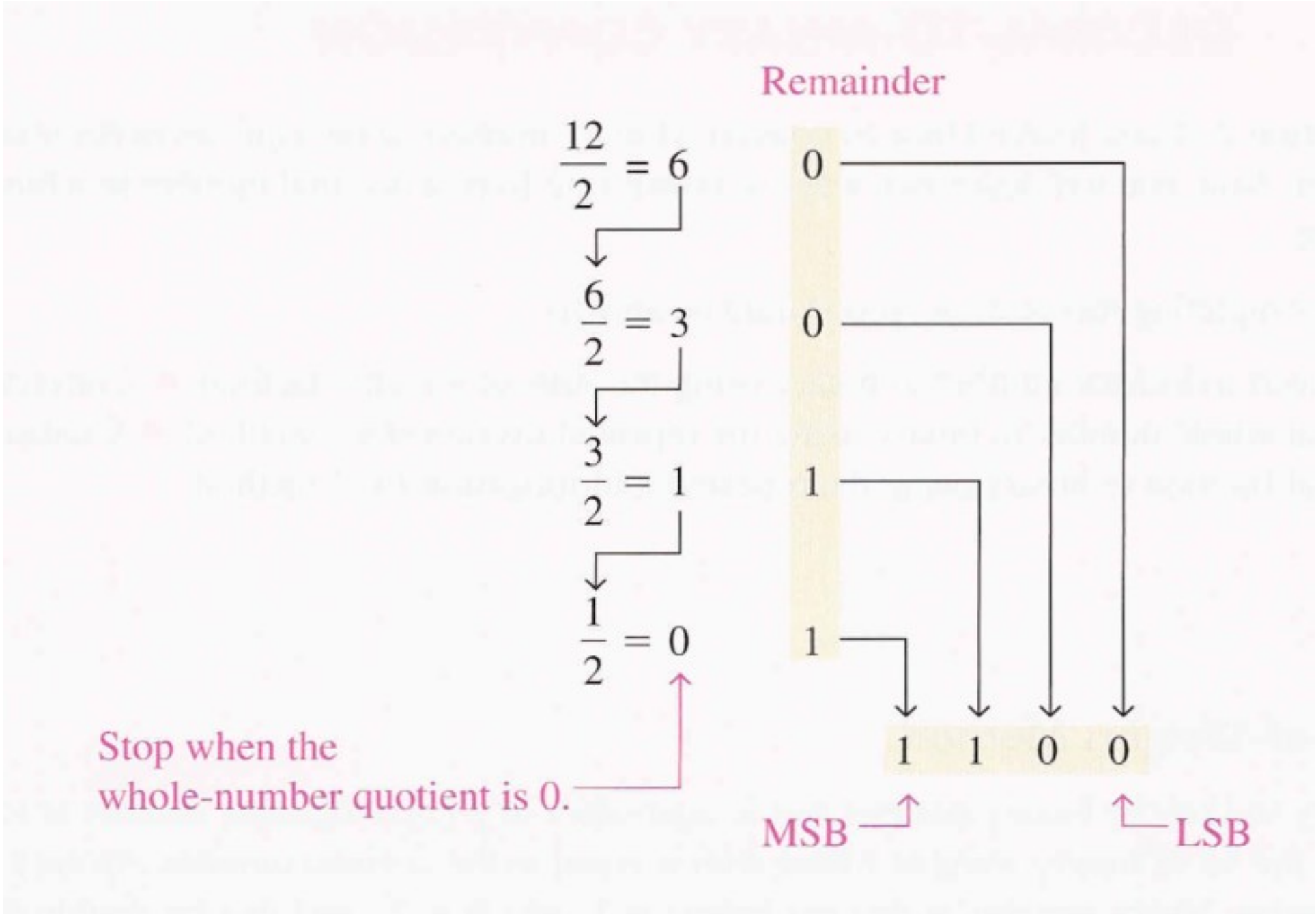
DECIMAL NUMBER	BINARY NUMBER			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Repeated Division-by-2 Method

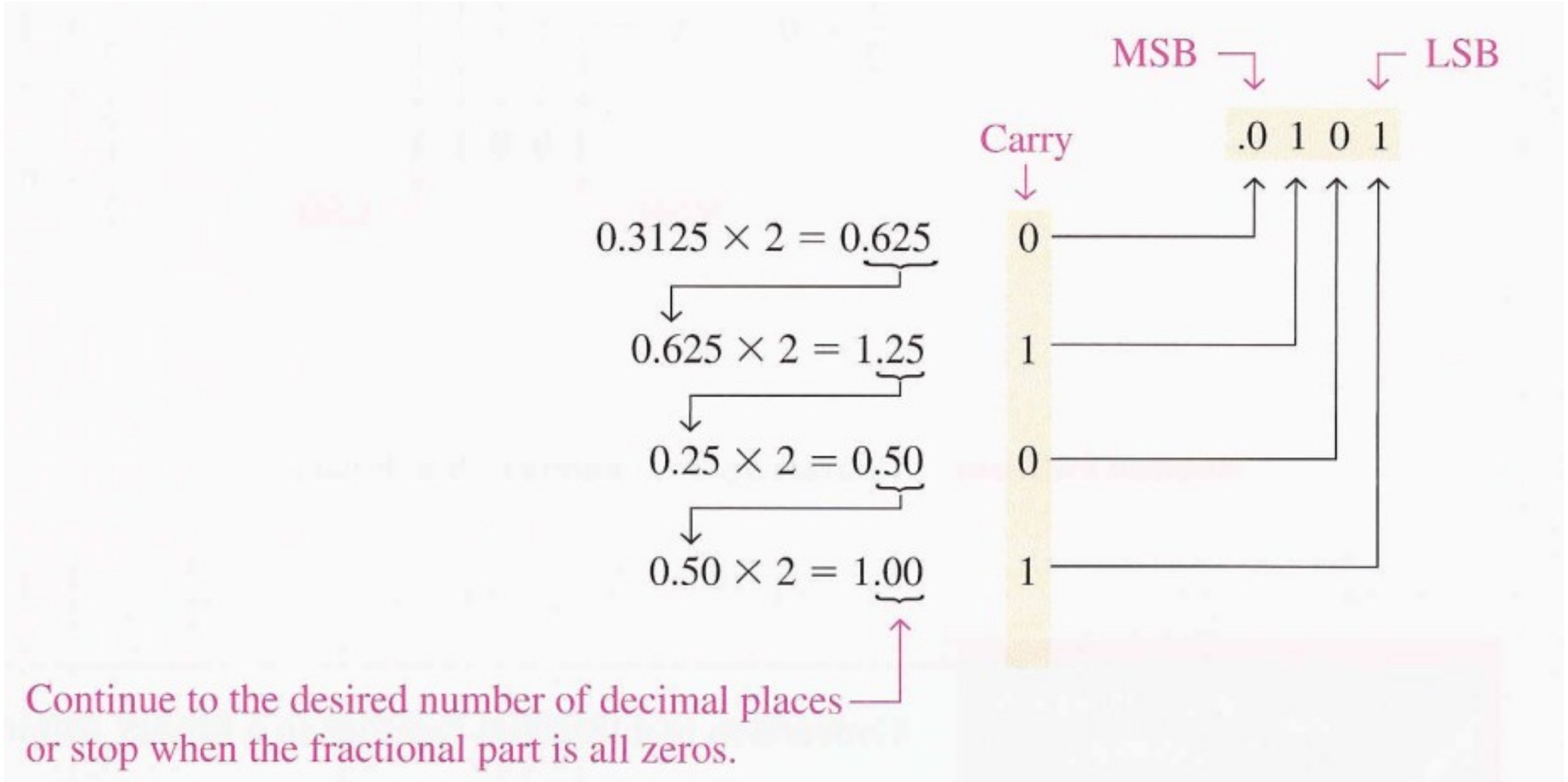
A systematic method of converting whole numbers from decimal to binary is the *repeated division-by-2* process. For example, to convert the decimal number 12 to binary, begin by dividing 12 by 2. Then divide each resulting quotient by 2 until there is a 0 whole-number quotient. The **remainders** generated by each division form the binary number. The first remainder to be produced is the LSB (least significant bit) in the binary number, and the last remainder to be produced is the MSB (most significant bit). This procedure is shown in the following steps for converting the decimal number 12 to binary.

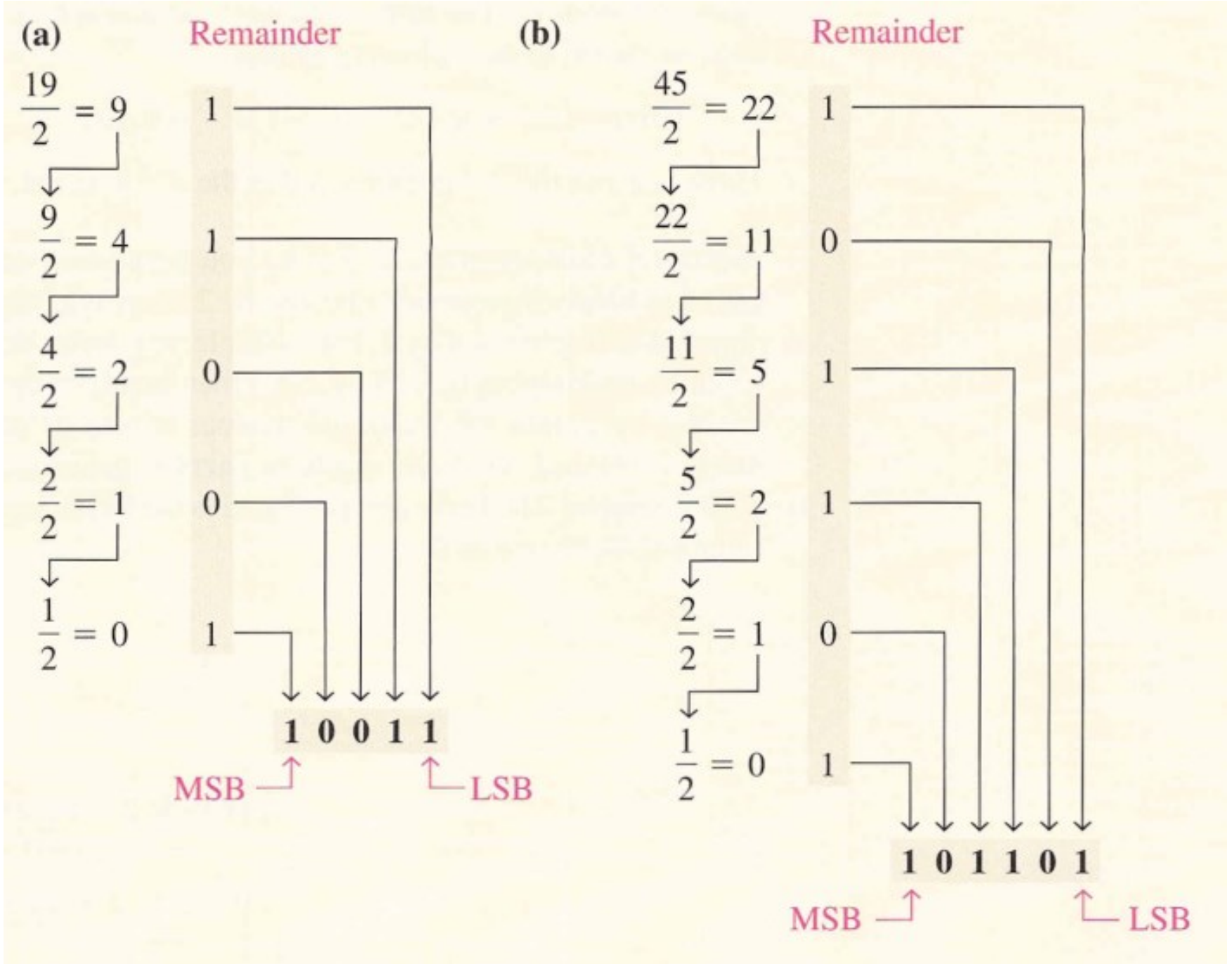
Convert the following decimal numbers to binary:

(a) 19 **(b)** 45


Remainder
12/2 = 6
6/2 = 3
3/2 = 1
1/2 = 0
0
0
1
1
1 1 0 0
Stop when the whole-number quotient is 0.
MSB
LSB

Repeated Multiplication by 2 As you have seen, decimal whole numbers can be converted to binary by repeated division by 2. Decimal fractions can be converted to binary by repeated multiplication by 2. For example, to convert the decimal fraction 0.3125 to binary, begin by multiplying 0.3125 by 2 and then multiplying each resulting fractional part of the product by 2 until the fractional product is zero or until the desired number of decimal places is reached. The carry digits, or **carries,** generated by the multiplications produce the binary number. The first carry produced is the MSB, and the last carry is the LSB. This procedure is illustrated as follows:


MSB
LSB
Carry
.0 1 0 1
0.3125 × 2 = 0.625
0
0.625 × 2 = 1.25
1
0.25 × 2 = 0.50
0
0.50 × 2 = 1.00
1
Continue to the desired number of decimal places
or stop when the fractional part is all zeros.


(a)
Remainder
19/2 = 9
9/2 = 4
4/2 = 2
2/2 = 1
1/2 = 0
1
1
0
0
1
1 0 0 1 1
MSB
LSB
(b)
Remainder
45/2 = 22
22/2 = 11
11/2 = 5
5/2 = 2
2/2 = 1
1/2 = 0
1
0
1
1
0
1
1 0 1 1 0 1
MSB
LSB

Binary Multiplication and Division

$$0 \times 0 = 0 \qquad 1 \times 0 = 0$$

$$0 \times 1 = 0 \qquad 1 \times 1 = 1$$

Perform the following binary multiplications:

(a) 11×11 **(b)** 101×111

(a)

	11	3
	× 11	× 3
Partial products	11	9
	+11	
	1001	

(b)

	111	7
	× 101	× 5
Partial products	111	35
	000	
	+111	
	100011	

Perform the following binary divisions:

(a) $110 \div 11$ **(b)** $110 \div 10$

(a)

$$\begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

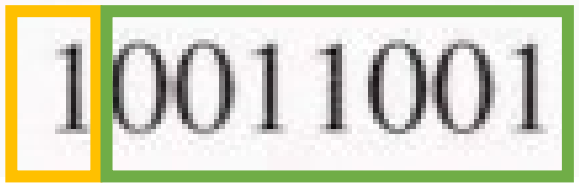
(b)

$$\begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Signed Binary Numbers

Digital systems, such as the computer, must be able to handle both positive and negative numbers. A signed binary number consists of both sign and magnitude information. The sign indicates whether a number is positive or negative, and the magnitude is the value of the number. There are three forms in which signed integer (whole) numbers can be represented in binary: sign-magnitude, 1's complement, and 2's complement. Of these, the 2's complement is the most important and the sign-magnitude is the least used. Noninteger and very large or small numbers can be expressed in floating-point format.

The decimal number -25 is expressed as



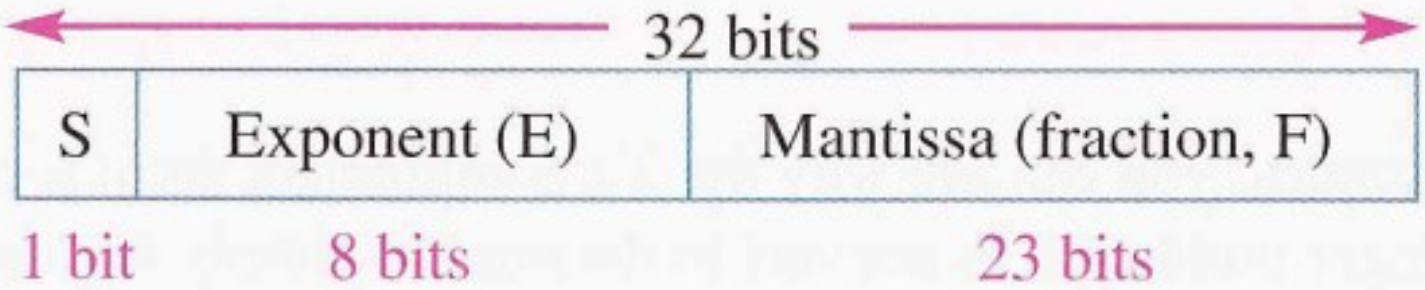
Sign Bit

Magnitude Bit

1 -ve

0 +ve

Single-Precision Floating-Point Binary Numbers In the standard format for a single-precision binary number, the sign bit (S) is the left-most bit, the exponent (E) includes the next eight bits, and the mantissa or fractional part (F) includes the remaining 23 bits, as shown next.



$$\text{Number} = (-1)^{S}(1 + F)(2^{E-127})$$

Q

Convert the decimal number 3.248×10^4 to a single-precision floating-point binary number.

A

Convert the decimal number to binary.

$$3.248 \times 10^4 = 32480 = 1111111011100000_2 = 1.11111011100000 \times 2^{14}$$

The MSB will not occupy a bit position because it is always a 1. Therefore, the mantissa is the fractional 23-bit binary number 11111011100000000000000 and the biased exponent is

$$14 + 127 = 141 = 10001101_2$$

The complete floating-point number is

0	**10001101**	**11111011100000000000000**

Hexadecimal Numbers

16^3	16^2	16^1	16^0
4096	256	16	1

DECIMAL	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Convert the following binary numbers to hexadecimal:

(a) 1100101001010111 **(b)** 1111111000101101001

(a) $\underbrace{1100}_{\downarrow}\underbrace{1010}_{\downarrow}\underbrace{0101}_{\downarrow}\underbrace{0111}_{\downarrow}$

C A 5 7 $= \mathbf{CA57}_{16}$

(b) $\underbrace{0011}_{\downarrow}\underbrace{1111}_{\downarrow}\underbrace{0001}_{\downarrow}\underbrace{0110}_{\downarrow}\underbrace{1001}_{\downarrow}$

3 F 1 6 9 $= \mathbf{3F169}_{16}$

Two zeros have been added in part (b) to complete a 4-bit group at the left.

Determine the binary numbers for the following hexadecimal numbers:

(a) $10A4_{16}$ **(b)** $CF8E_{16}$ **(c)** 9742_{16}

(a) 1 0 A 4

1000010100100

(b) C F 8 E

1100111110001110

(c) 9 7 4 2

1001011101000010

In part (a), the MSB is understood to have three zeros preceding it, thus forming a 4-bit group.

Convert the following hexadecimal numbers to decimal:

(a) $1C_{16}$ **(b)** $A85_{16}$

Remember, convert the hexadecimal number to binary first, then to decimal.

(a)

$$\overbrace{0001}^{1}\overbrace{1100}^{C} = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28}_{10}$$

(b)

$$\overbrace{1010}^{A}\overbrace{1000}^{8}\overbrace{0101}^{5} = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = \mathbf{2693}_{10}$$

Convert the following hexadecimal numbers to decimal:

(a) $E5_{16}$ **(b)** $B2F8_{16}$

Recall from Table 2–3 that letters A through F represent decimal numbers 10 through 15, respectively.

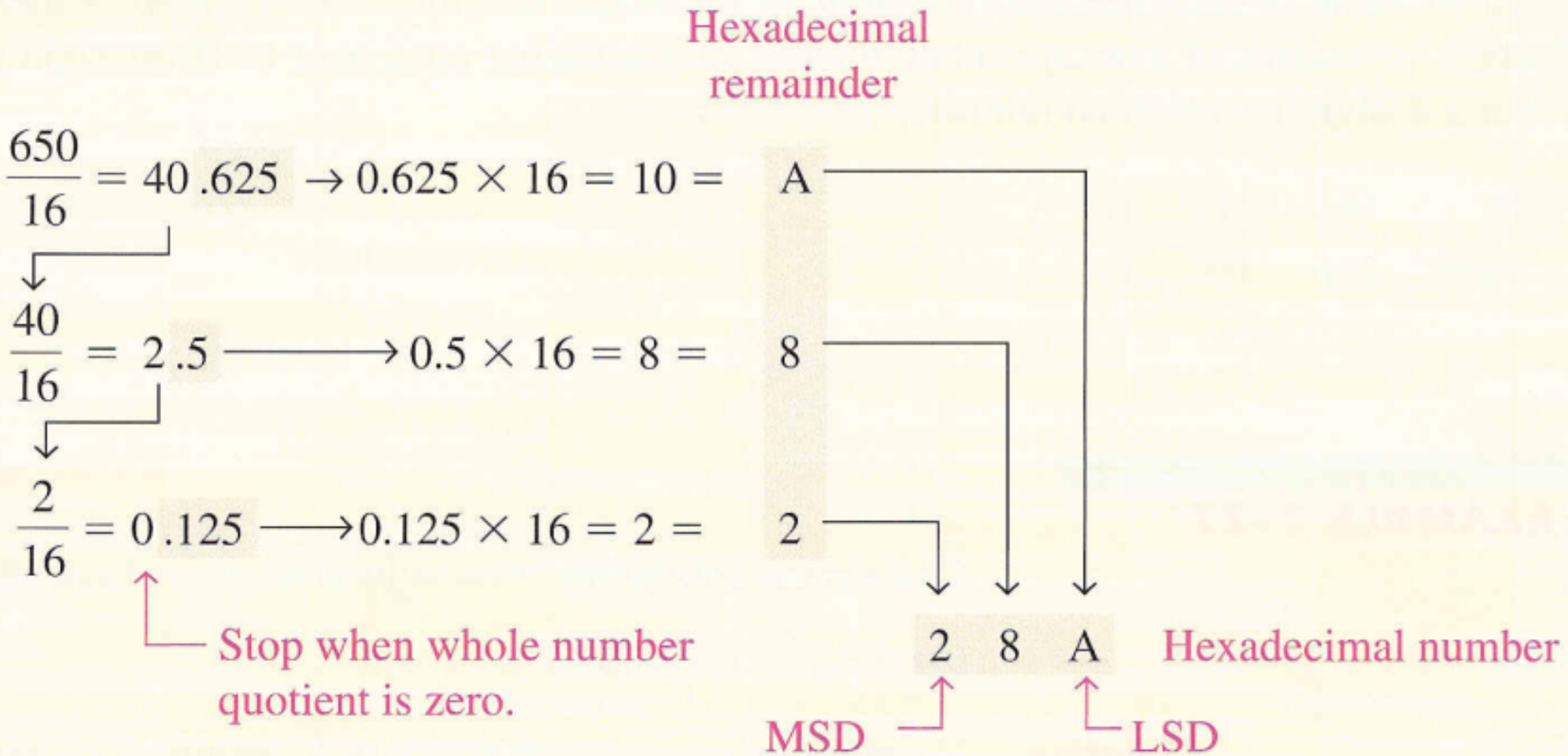
(a) $E5_{16} = (E \times 16) + (5 \times 1) = (14 \times 16) + (5 \times 1) = 224 + 5 = \mathbf{229}_{10}$

(b) $B2F8_{16} = (B \times 4096) + (2 \times 256) + (F \times 16) + (8 \times 1)$

$= (11 \times 4096) + (2 \times 256) + (15 \times 16) + (8 \times 1)$

$= 45{,}056 + 512 + 240 + 8 = \mathbf{45{,}816}_{10}$

Convert the decimal number 650 to hexadecimal by repeated division by 16.



Add the following hexadecimal numbers:

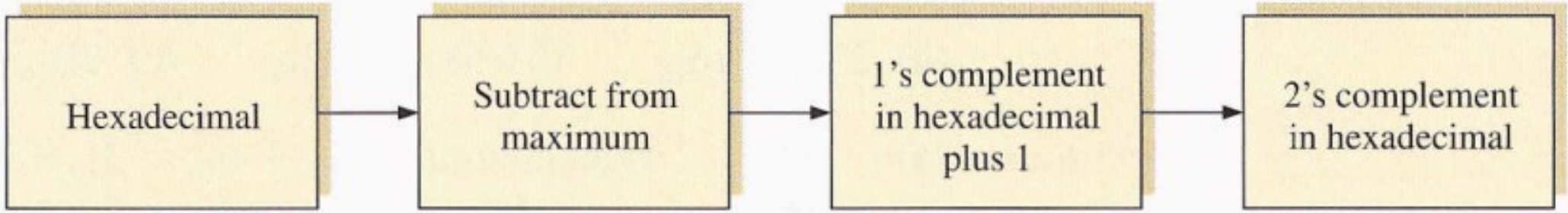
(a) $23_{16} + 16_{16}$ **(b)** $58_{16} + 22_{16}$ **(c)** $2B_{16} + 84_{16}$ **(d)** $DF_{16} + AC_{16}$

(a)
$$\begin{array}{r} 23_{16} \\ +16_{16} \\ \hline \mathbf{39}_{16} \end{array}$$
right column: $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$
left column: $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

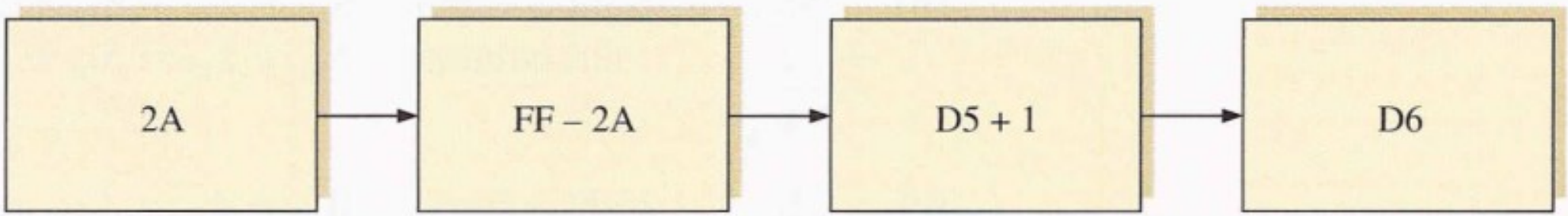
(b)
$$\begin{array}{r} 58_{16} \\ +\ 22_{16} \\ \hline \mathbf{7A}_{16} \end{array}$$
right column: $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$
left column: $5_{16} + 2_{16} = 5_{10} + 2_{10} = 7_{10} = 7_{16}$

(c)
$$\begin{array}{r} 2B_{16} \\ +\ 84_{16} \\ \hline \mathbf{AF}_{16} \end{array}$$
right column: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$
left column: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

(d)
$$\begin{array}{r} DF_{16} \\ +\ AC_{16} \\ \hline \mathbf{18B}_{16} \end{array}$$
right column: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$
$27_{10} - 16_{10} = 11_{10} = B_{16}$ with a 1 carry
left column: $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$
$24_{10} - 16_{10} = 8_{10} = 8_{16}$ with a 1 carry


Hexadecimal
Subtract from maximum
1's complement in hexadecimal plus 1
2's complement in hexadecimal

Example:


2A
FF – 2A
D5 + 1
D6

Subtract the following hexadecimal numbers:

(a) $84_{16} - 2A_{16}$ **(b)** $C3_{16} - 0B_{16}$

(a) $2A_{16} = 00101010$

2's complement of $2A_{16} = 11010110 = D6_{16}$ (using Method 1)

$$\begin{array}{rl} 84_{16} & \\ +\ D6_{16} & \text{Add} \\ \hline \not{1}5A_{16} & \text{Drop carry, as in 2's complement addition} \end{array}$$

The difference is $\mathbf{5A}_{16}$.

(b) $0B_{16} = 00001011$

2's complement of $0B_{16} = 11110101 = F5_{16}$ (using Method 1)

$$\begin{array}{rl} C3_{16} & \\ +\ F5_{16} & \text{Add} \\ \hline \not{1}B8_{16} & \text{Drop carry} \end{array}$$

The difference is $\mathbf{B8}_{16}$.

Octal Numbers

OCTAL DIGIT	0	1	2	3	4	5	6	7
BINARY	000	001	010	011	100	101	110	111

Convert each of the following octal numbers to binary:

(a) 13_8 **(b)** 25_8 **(c)** 140_8 **(d)** 7526_8

(a) 1 3 → **001011**

(b) 2 5 → **010101**

(c) 1 4 0 → **001100000**

(d) 7 5 2 6 → **111101010110**

Convert each of the following binary numbers to octal:

(a) 110101 **(b)** 101111001 **(c)** 100110011010 **(d)** 11010000100

(a) $\underbrace{110}_{6}\underbrace{101}_{5} = \mathbf{65}_8$

(b) $\underbrace{101}_{5}\underbrace{111}_{7}\underbrace{001}_{1} = \mathbf{571}_8$

(c) $\underbrace{1001}_{4}\underbrace{1001}_{6}\underbrace{1010}_{2} = \mathbf{4632}_8$

(d) $\underbrace{0110}_{3}\underbrace{1000}_{2}\underbrace{0100}_{0}\underbrace{}_{4} = \mathbf{3204}_8$

Home Work (Due date 20-10-2024)

1 Convert each octal number to decimal:

(a) 14_8 **(b)** 53_8 **(c)** 67_8 **(d)** 174_8

2 Add the binary numbers:

(a) 10 + 10 **(b)** 10 + 11 **(c)** 100 + 11
(d) 111 + 101 **(e)** 1111 + 111 **(f)** 1111 + 1111

3 Convert each decimal fraction to binary using the sum-of-weights method:

(a) 0.26 **(b)** 0.762 **(c)** 0.0975

4 Convert each binary number to decimal:

(a) 110011.11 **(b)** 101010.01 **(c)** 1000001.111
(d) 1111000.101 **(e)** 1011100.10101 **(f)** 1110001.0001
(g) 1011010.1010 **(h)** 1111111.11111

University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

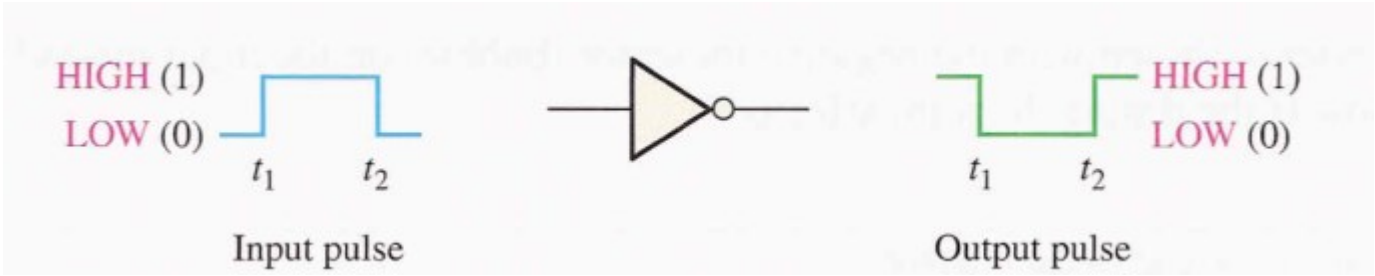
Lec. Dr. Taif Alawsi

Lec. 3: Logic Gates and their Applications: 2024-Oct-13

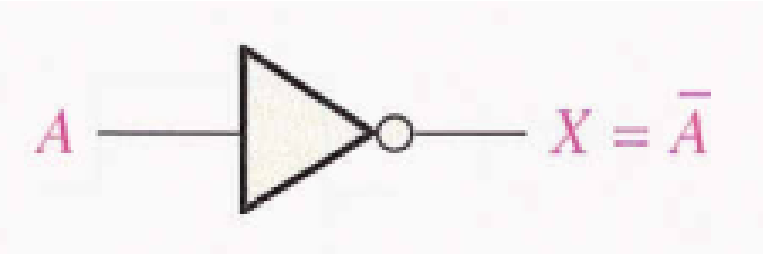
Lecture Outline

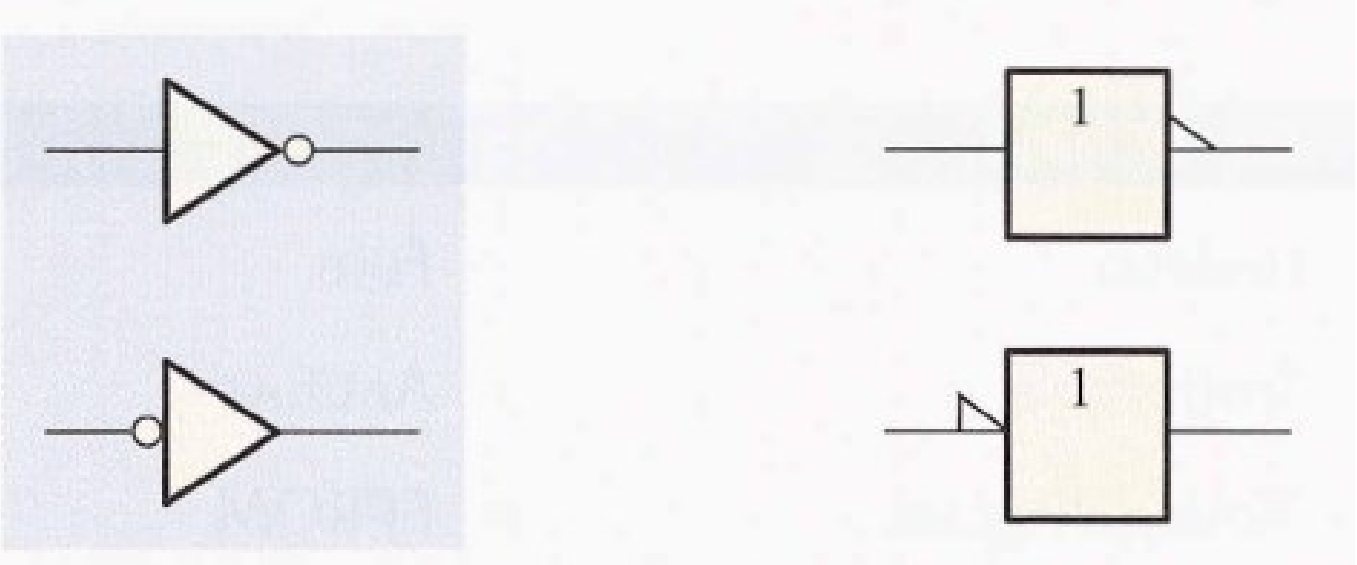
1. NOT GATE
2. AND GATE
3. OR GATE
4. NAND GATE
5. NOR GATE
6. XOR GATE
7. XNOR GATE
8. Writing Boolean Expression
9. IC
10. Datasheet
11. HW

NOT GATE

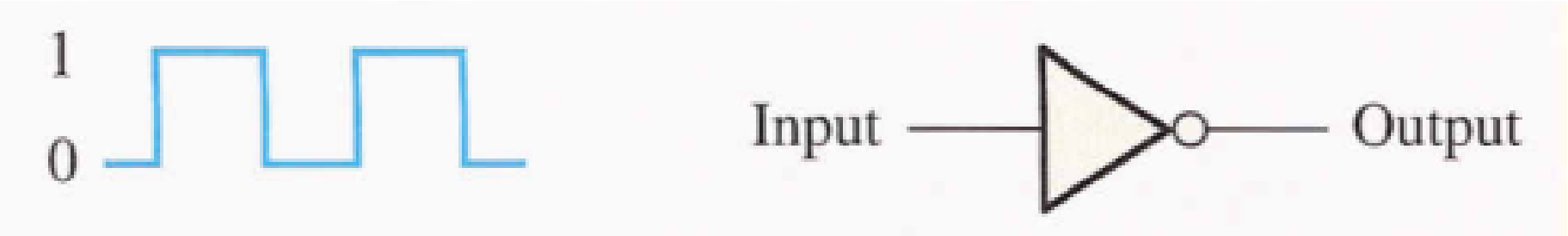


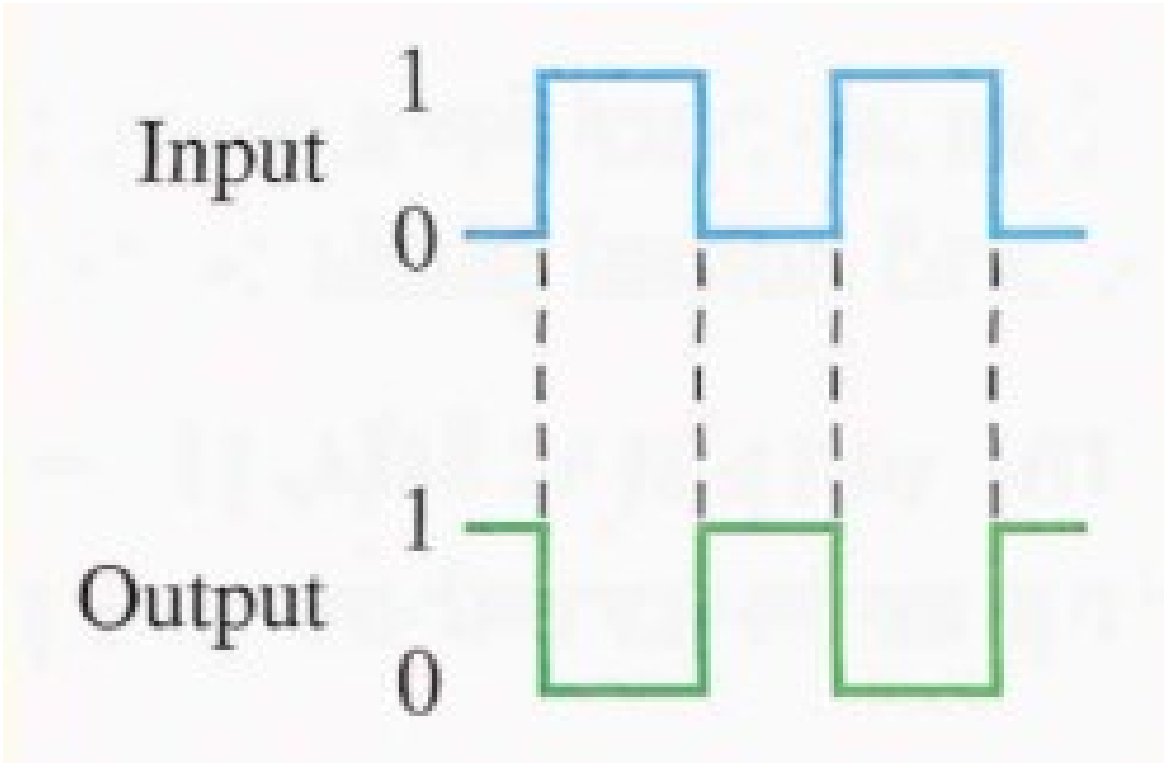
INPUT	OUTPUT
LOW (0)	HIGH (1)
HIGH (1)	LOW (0)



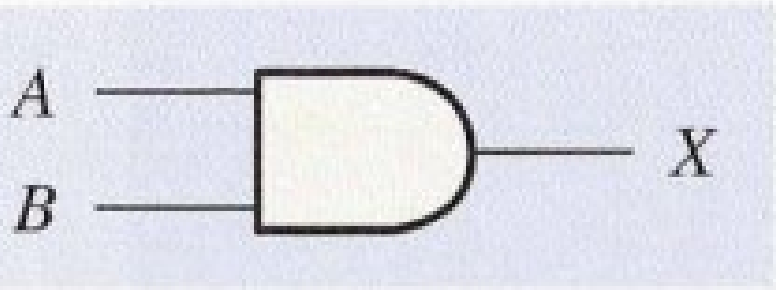


NOT GATE





AND GATE



$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

1 = HIGH, 0 = LOW

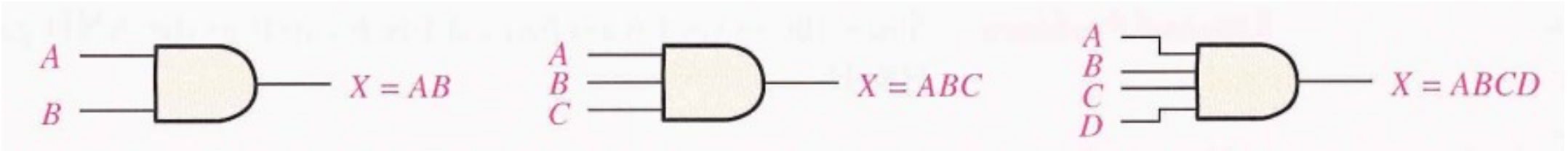
$$X = AB$$

For two input variables: $N = 2^2 = 4$ combinations

For three input variables: $N = 2^3 = 8$ combinations

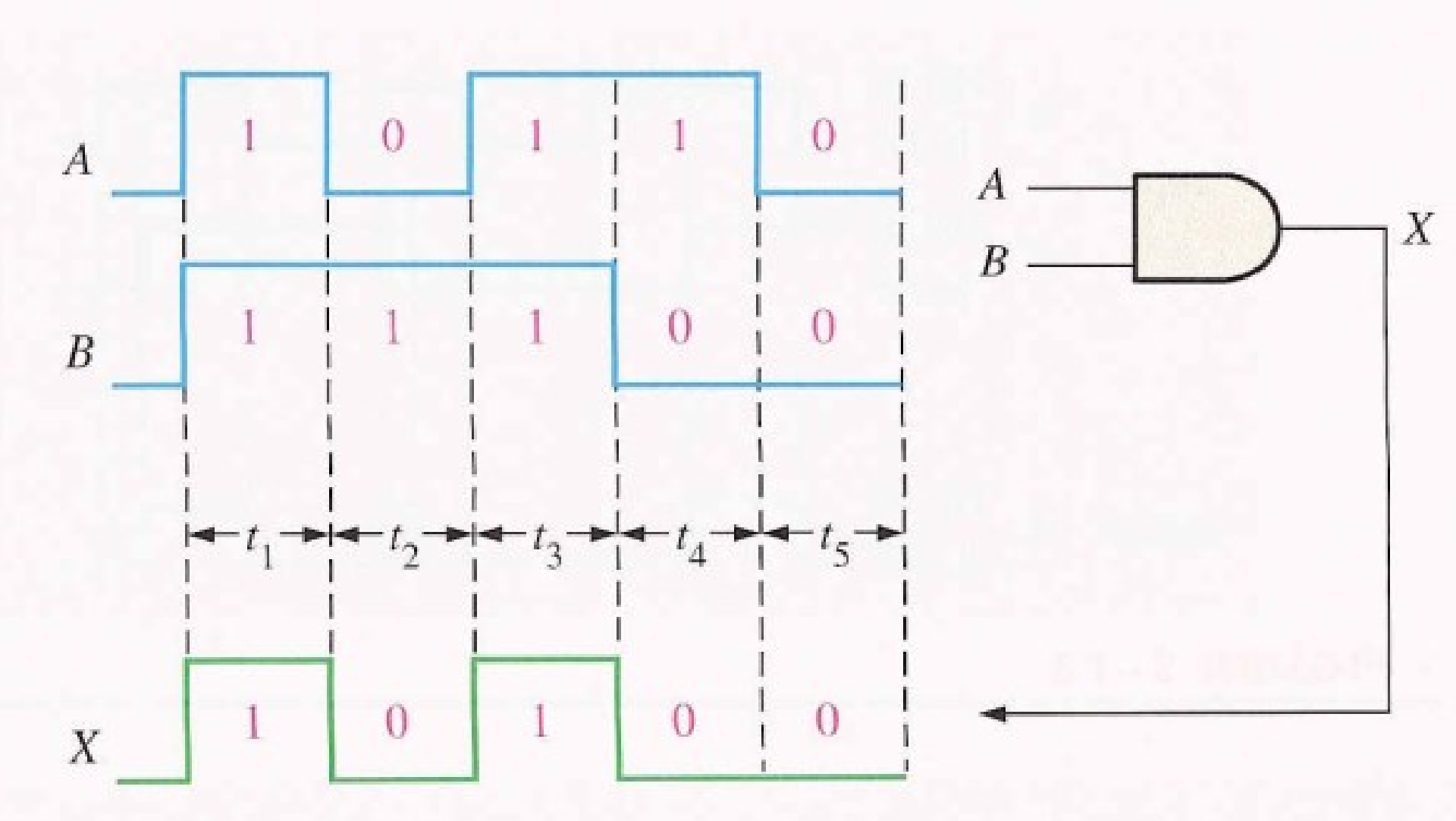
For four input variables: $N = 2^4 = 16$ combinations

AND GATE

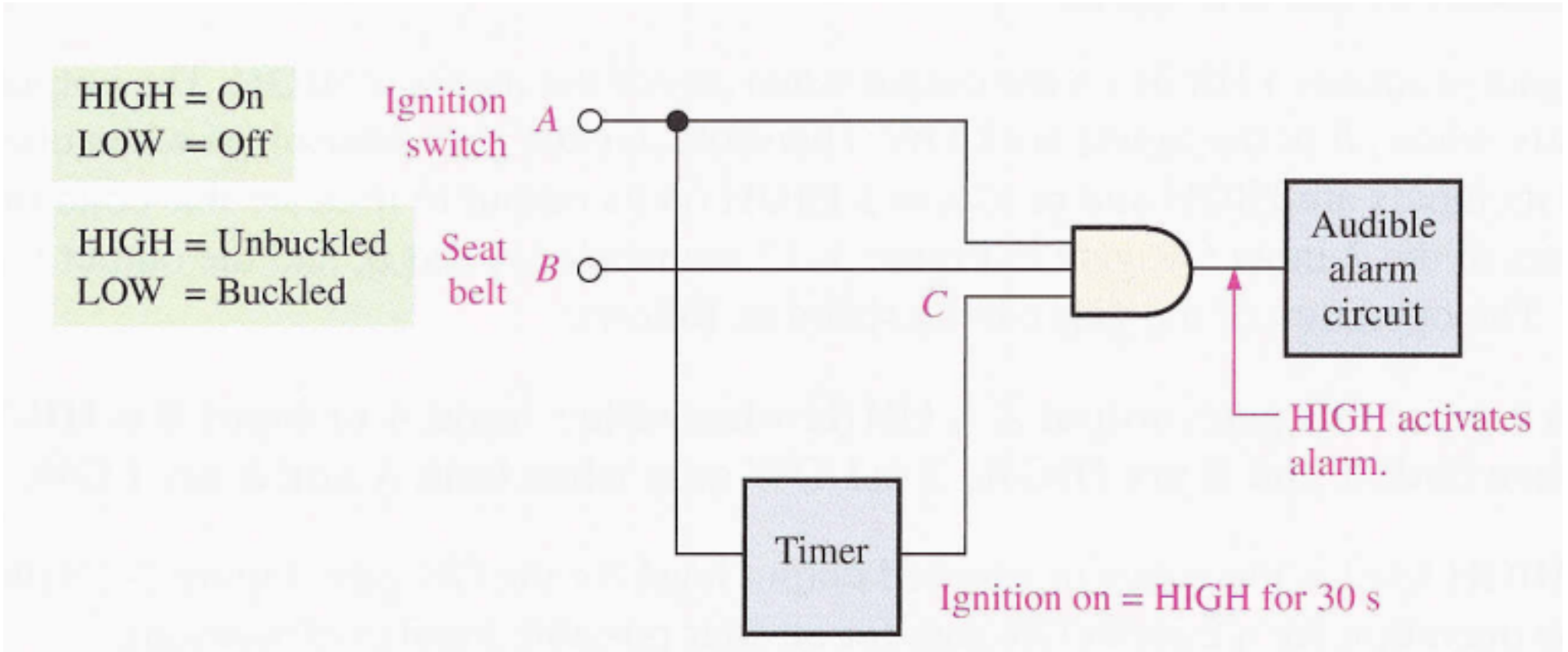


INPUTS			OUTPUT
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

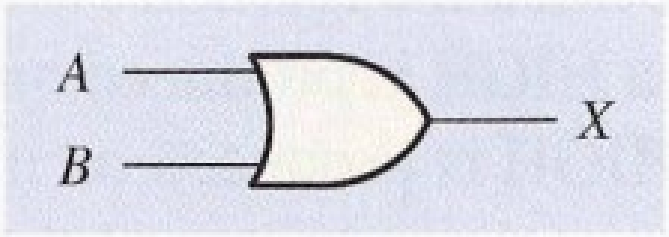
AND GATE

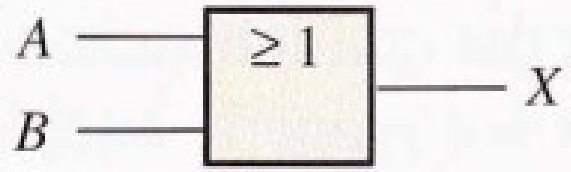


A simple seat belt alarm circuit using an AND gate.



OR GATE

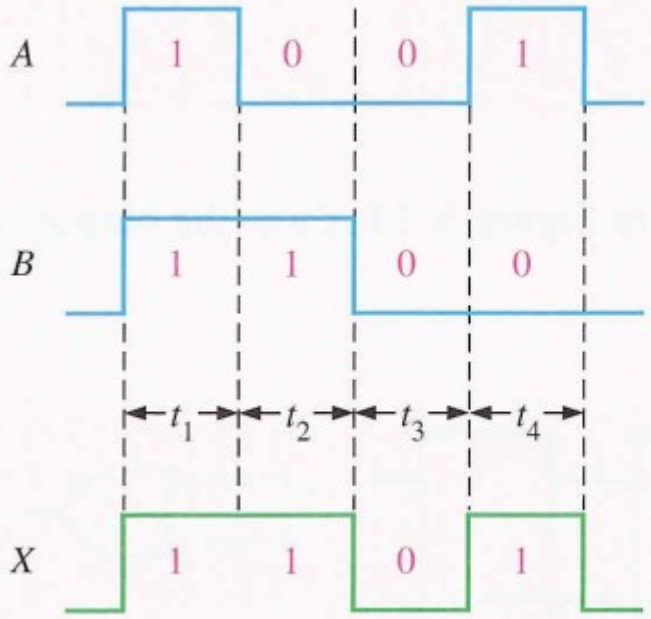


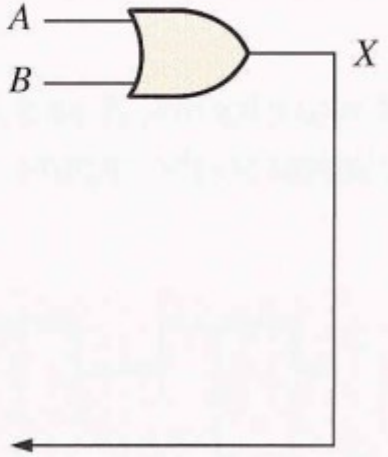


$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

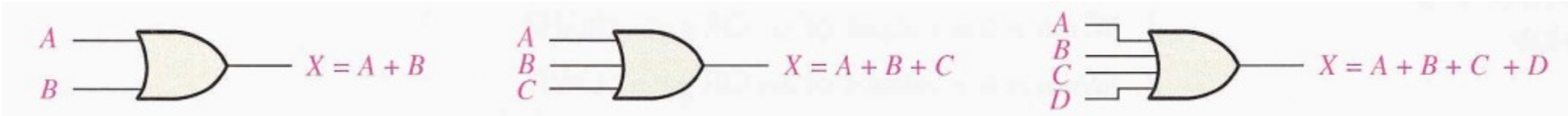
1 = HIGH, 0 = LOW



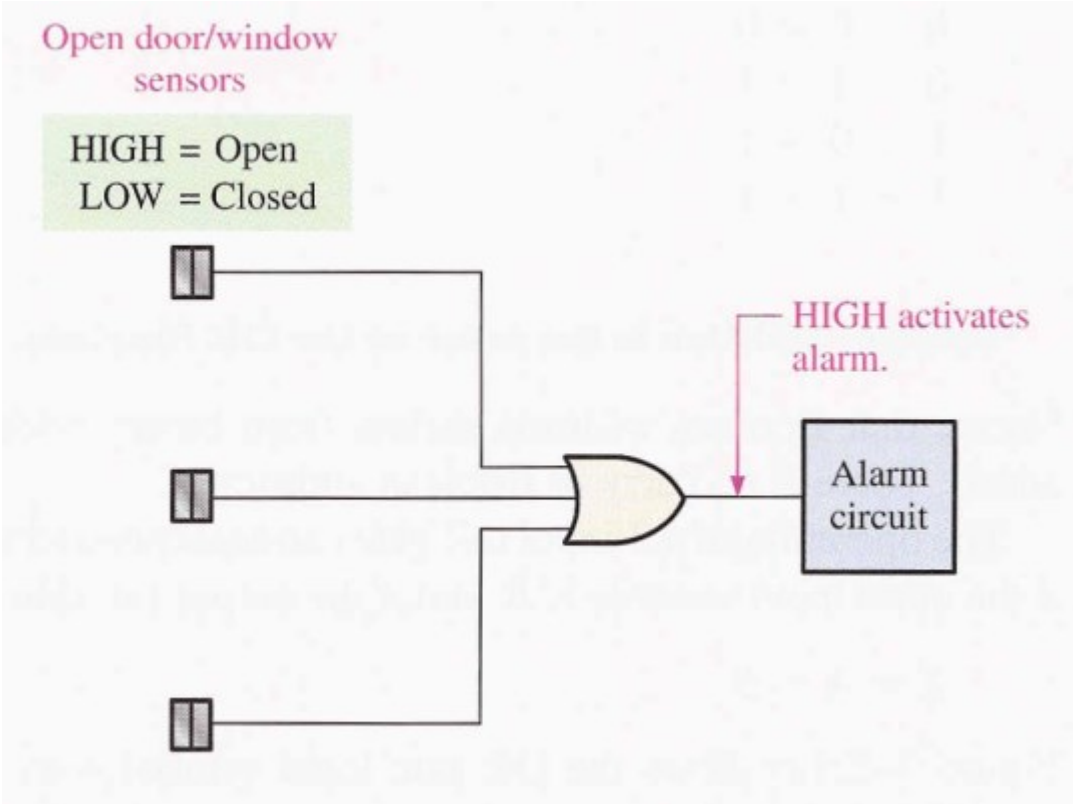


$$X = A + B$$

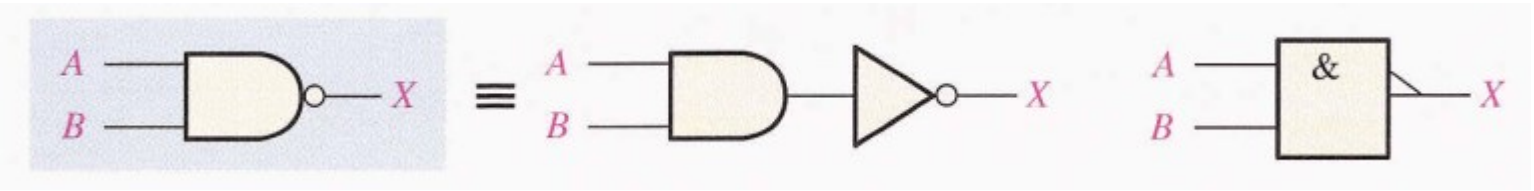
OR GATE

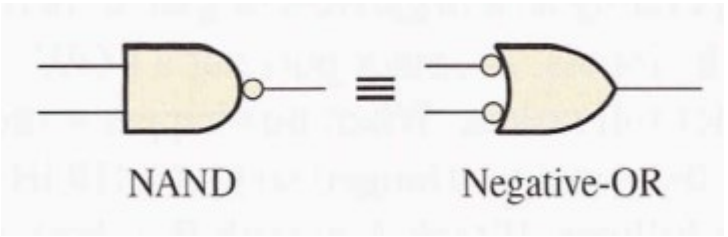


A simplified intrusion detection system using an OR gate.



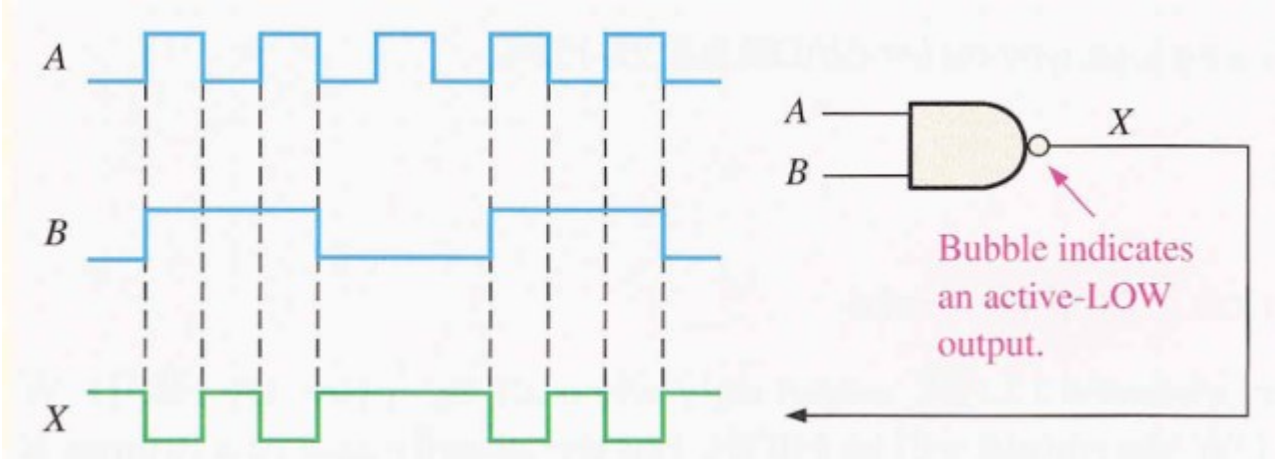
NAND GATE





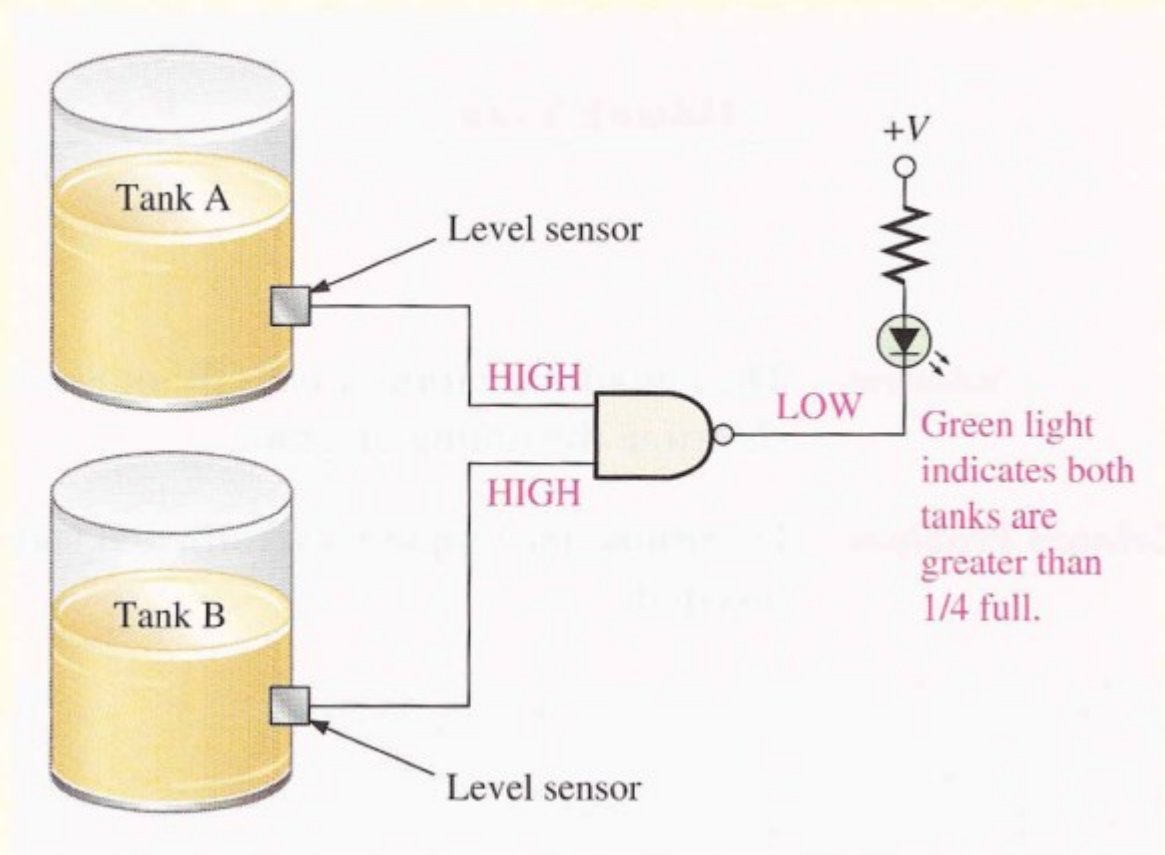
INPUTS		OUTPUT
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

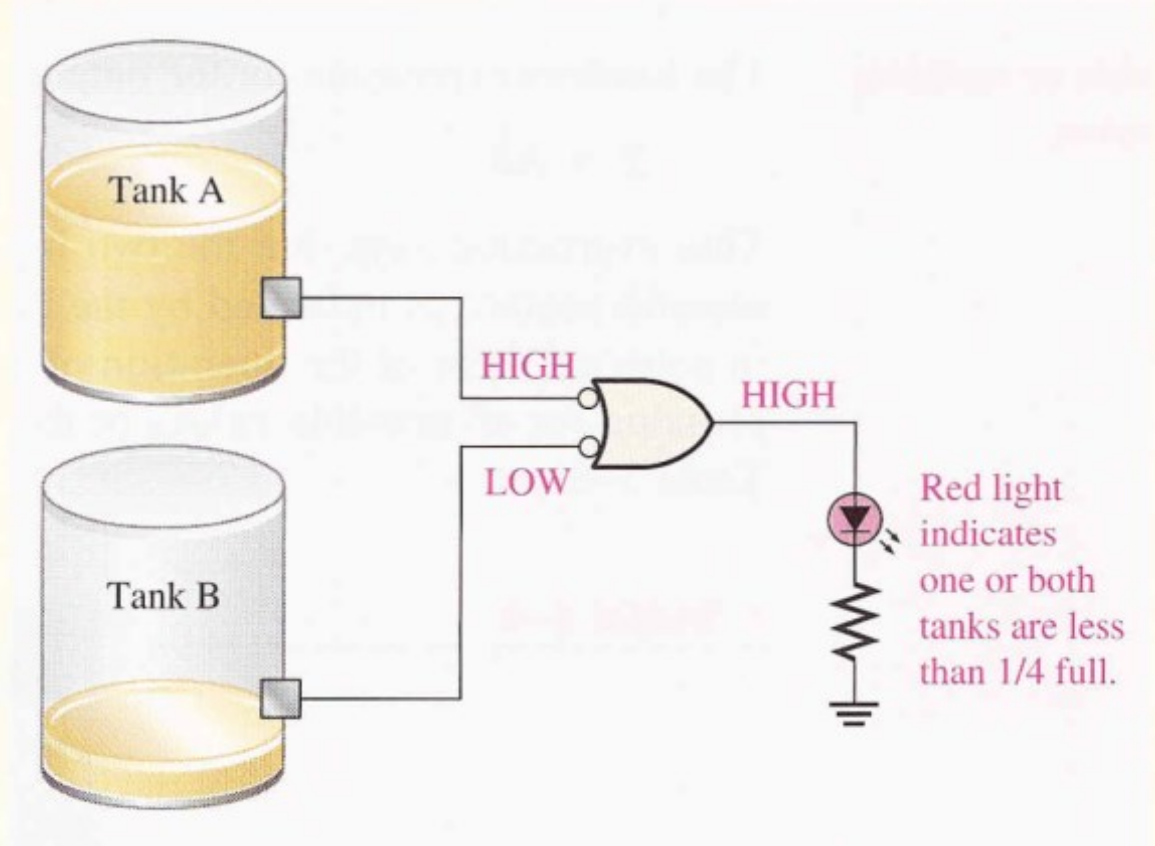
1 = HIGH, 0 = LOW.



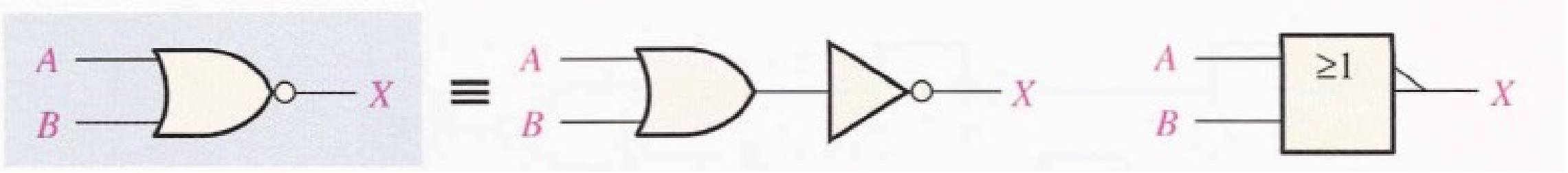
$$X = \overline{AB}$$

NAND GATE


Tank A
Level sensor
+V
HIGH
LOW
Green light indicates both tanks are greater than 1/4 full.
HIGH
Tank B
Level sensor

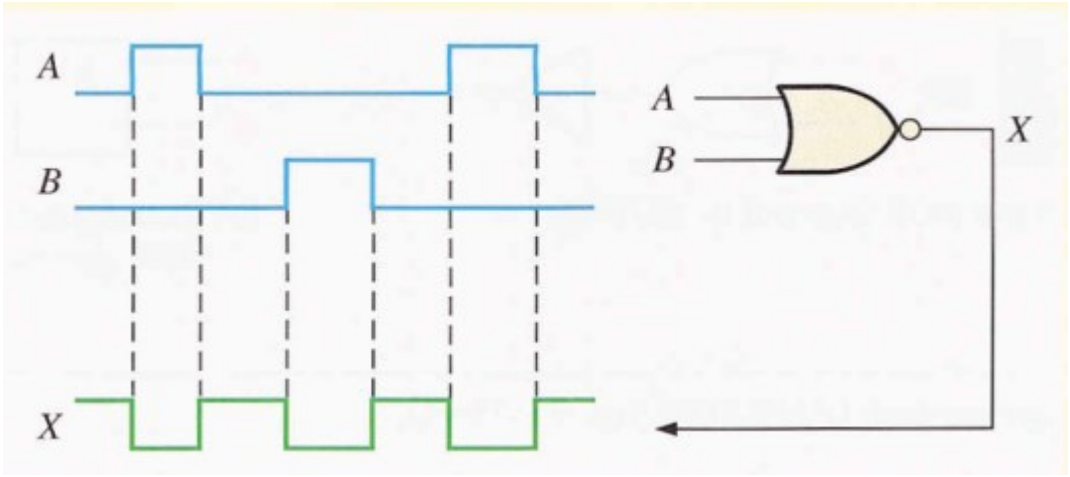

Tank A
HIGH
HIGH
LOW
Red light indicates one or both tanks are less than 1/4 full.
Tank B

NOR GATE

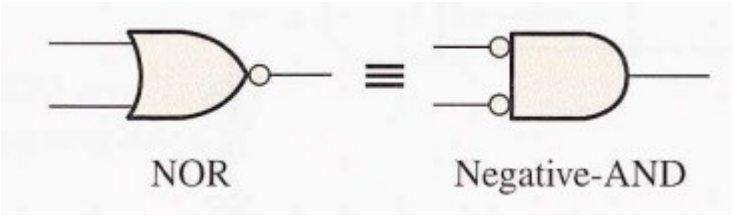


INPUTS		OUTPUT
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

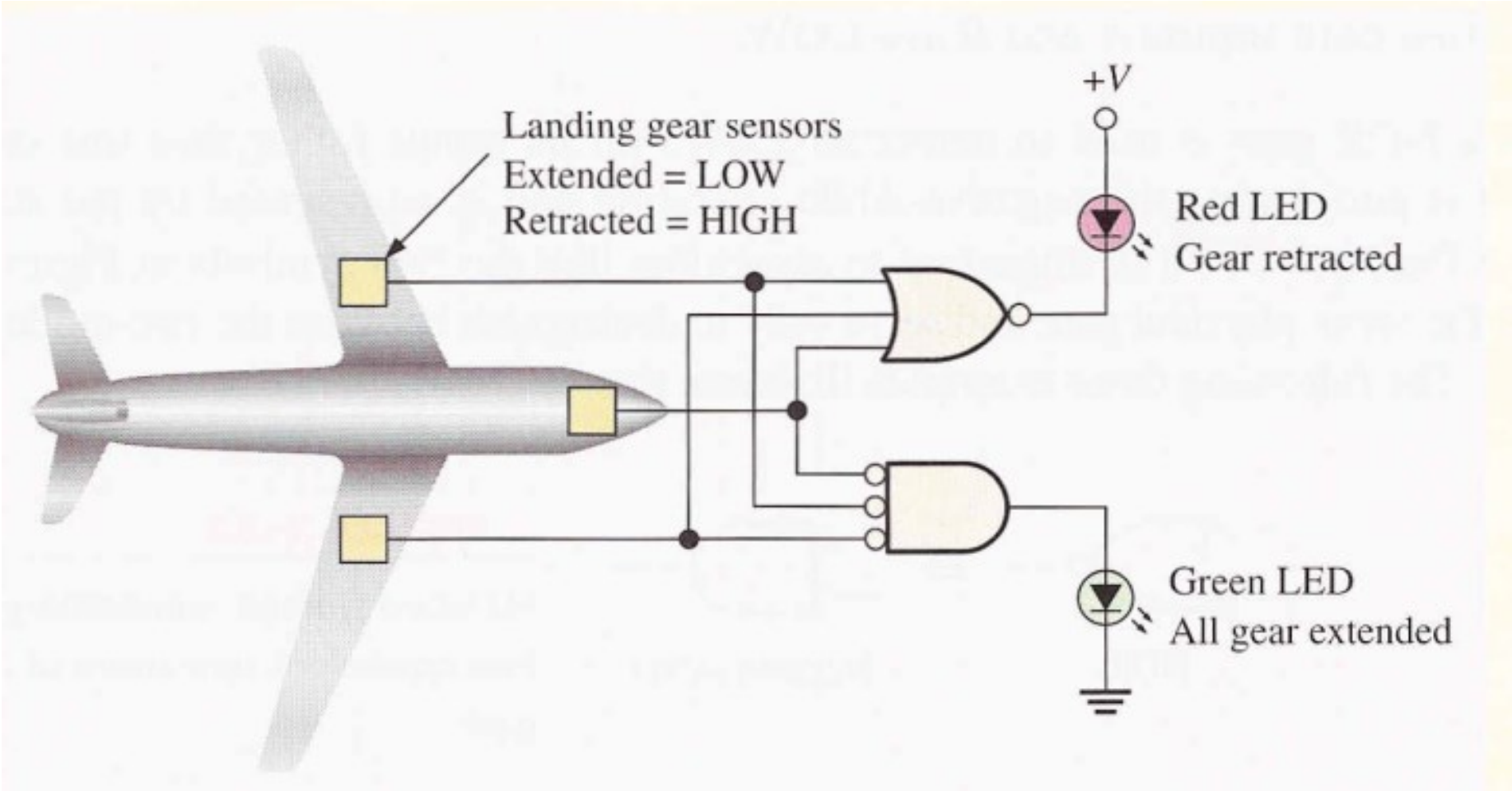
1 = HIGH, 0 = LOW.



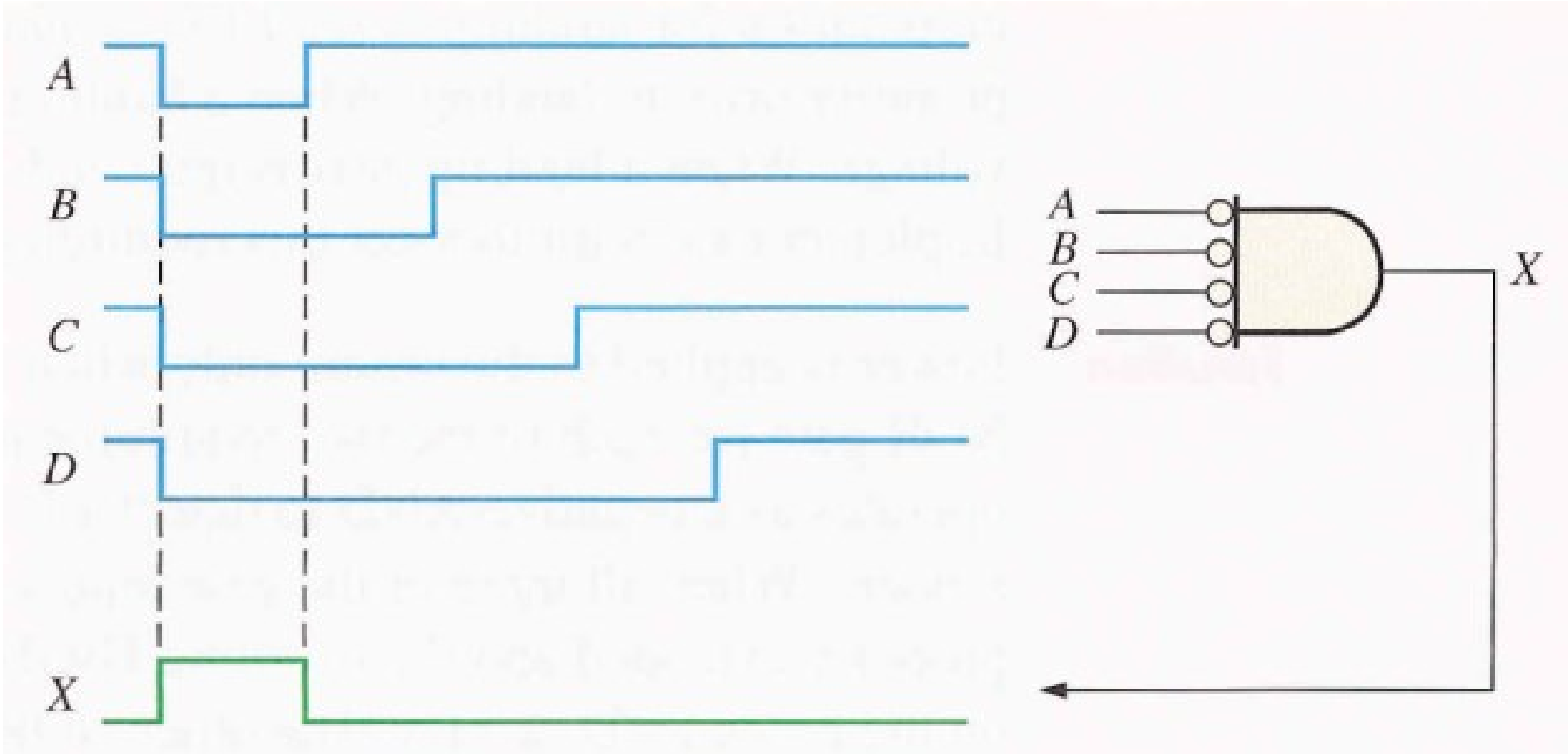
$$X = \overline{A + B}$$



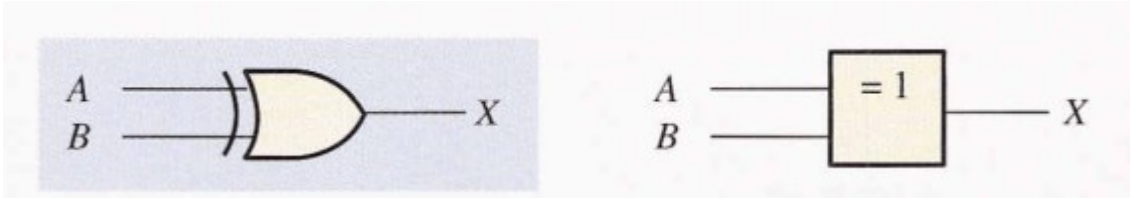
NOR GATE

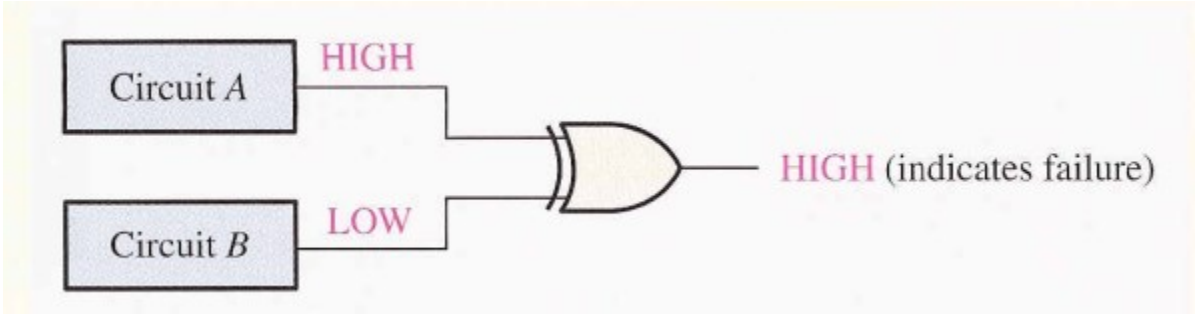

+V
Landing gear sensors
Extended = LOW
Retracted = HIGH
Red LED
Gear retracted
Green LED
All gear extended

NOR GATE



XOR GATE

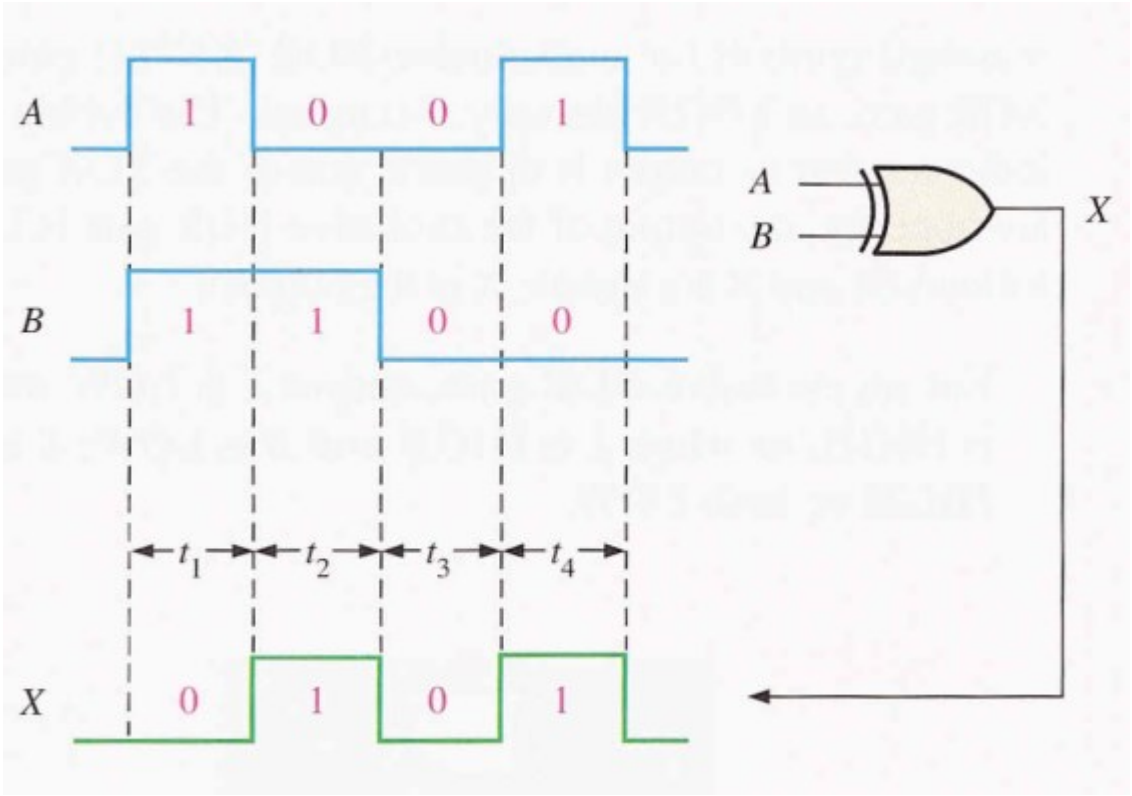




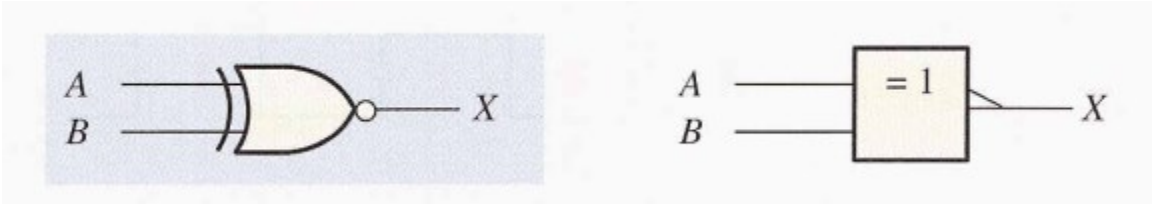
INPUTS		OUTPUT
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$$X = A \oplus B$$

$$X = \overline{A}B + A\overline{B}$$



XNOR GATE

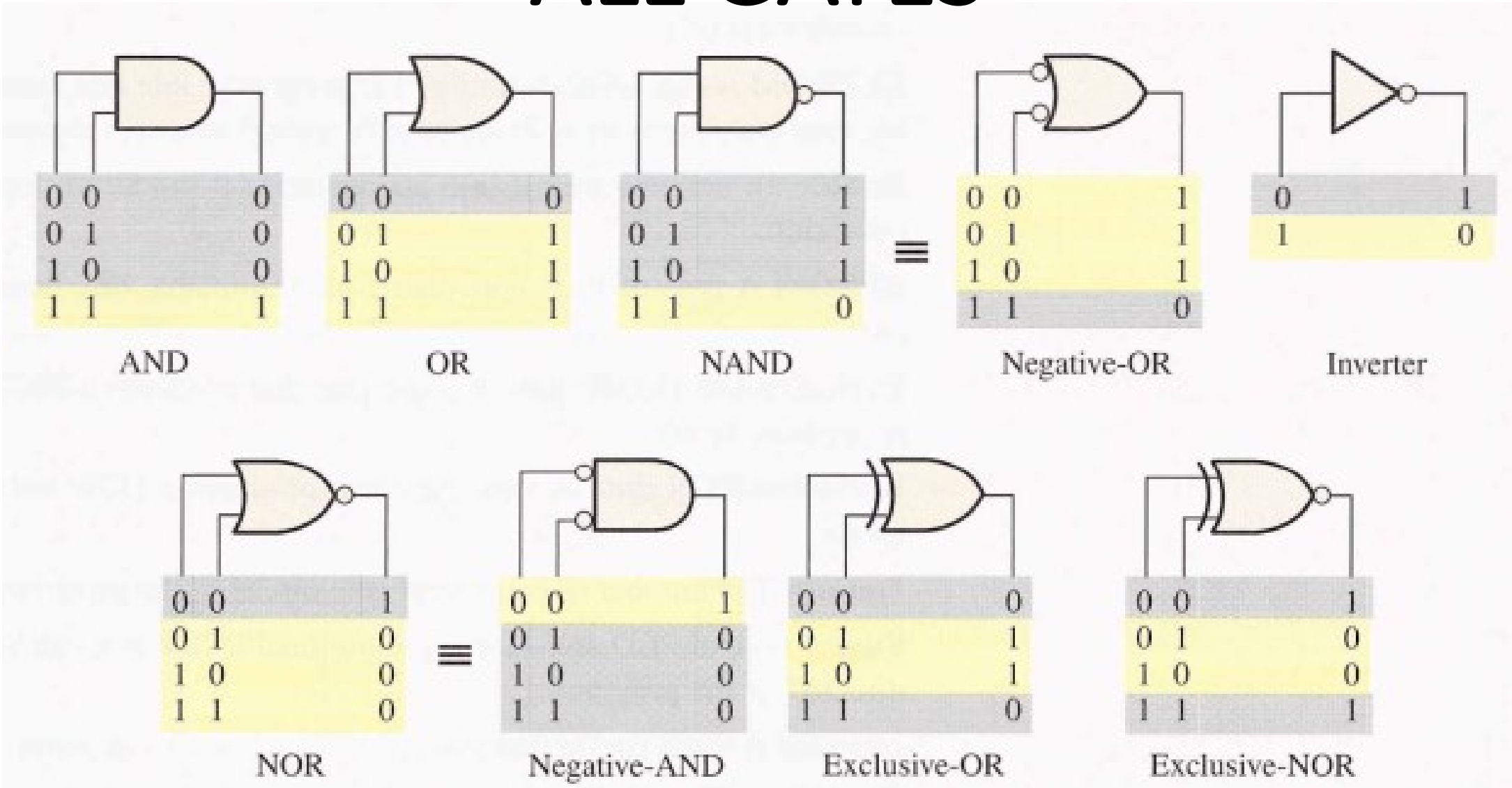


$$\mathbf{X = \overline{A \oplus B}}$$

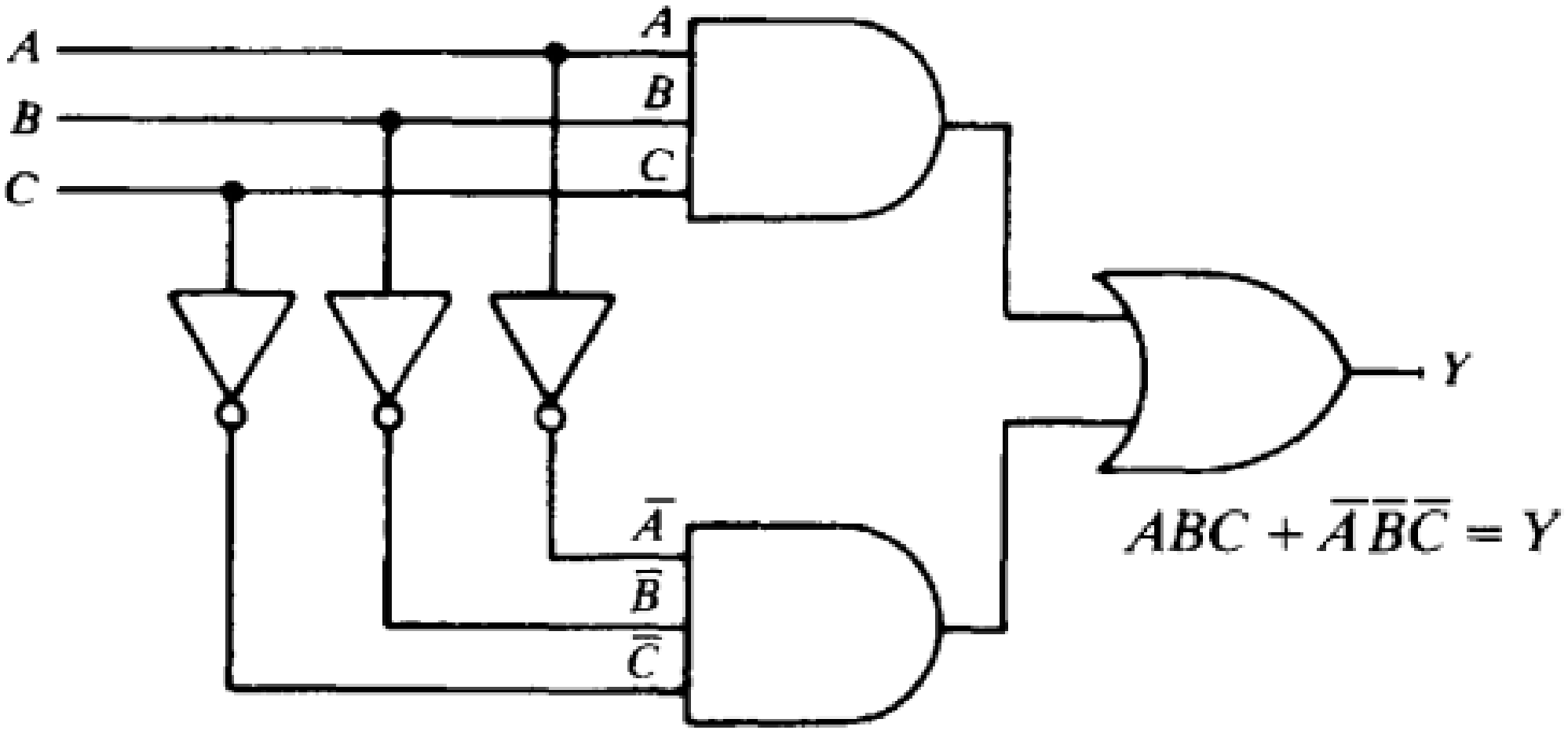
INPUTS		OUTPUT
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

$$X = \overline{(A \oplus B)} = (A.B + \overline{A}.\overline{B})$$

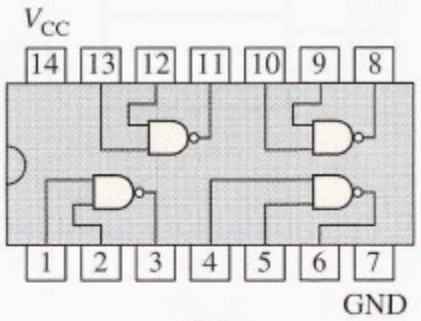
ALL GATES


0 0 0
0 1 0
1 0 0
1 1 1
AND
0 0 0
0 1 1
1 0 1
1 1 1
OR
0 0 1
0 1 1
1 0 1
1 1 0
NAND
≡
0 0 1
0 1 1
1 0 1
1 1 0
Negative-OR
0 1
1 0
Inverter
0 0 1
0 1 0
1 0 0
1 1 0
NOR
≡
0 0 1
0 1 0
1 0 0
1 1 0
Negative-AND
0 0 0
0 1 1
1 0 1
1 1 0
Exclusive-OR
0 0 1
0 1 0
1 0 0
1 1 1
Exclusive-NOR

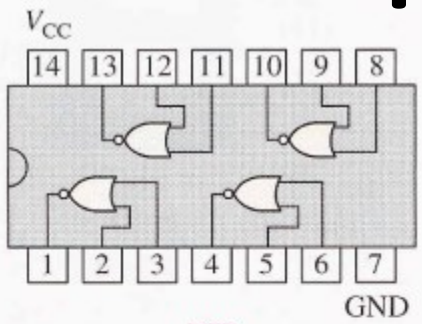
Writing Boolean Expression



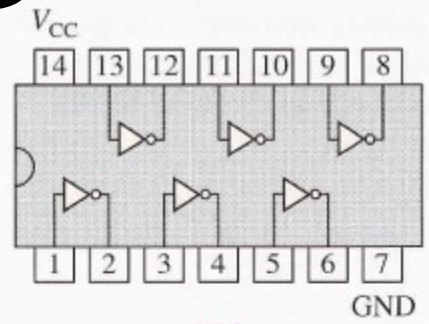
IC



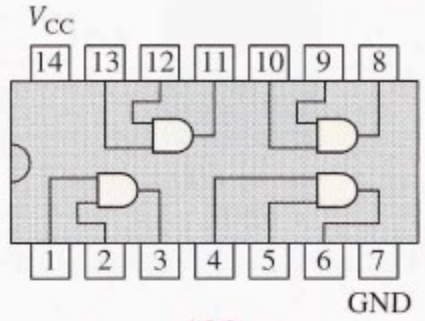
'00



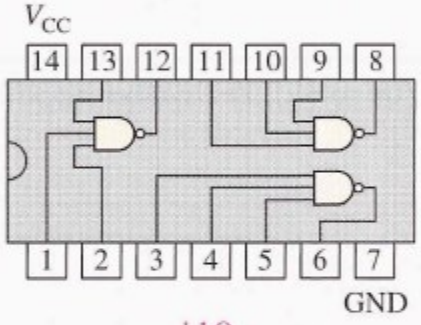
'02



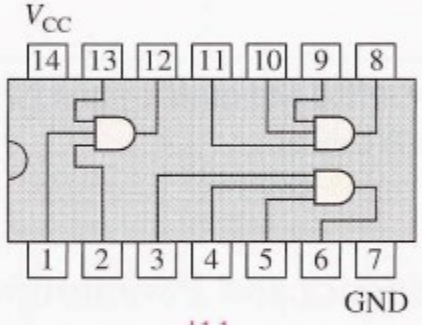
'04



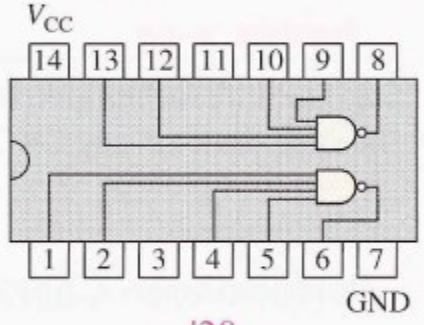
'08



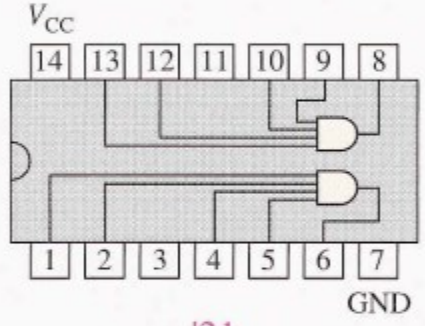
'10



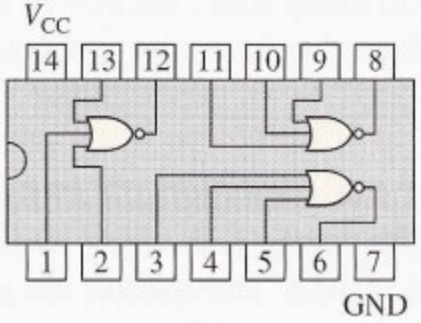
'11



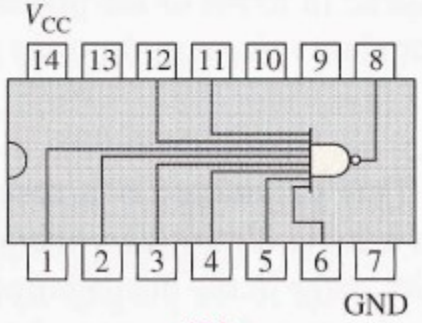
'20



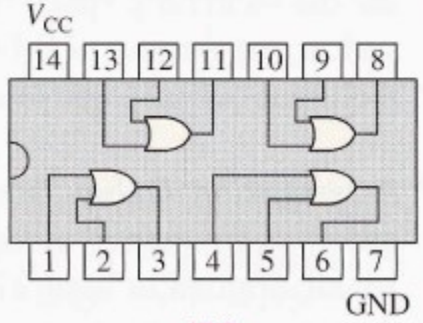
'21



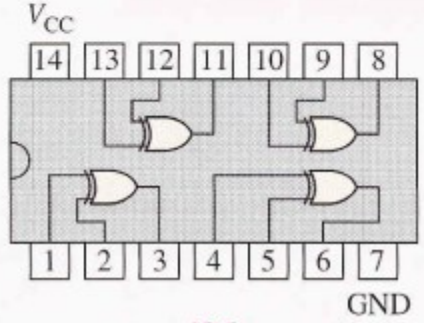
'27



'30



'32



'86

DATASHEET

QUAD 2-INPUT NAND GATE

• ESD > 3500 Volts

SN54/74LS00

QUAD 2-INPUT NAND GATE
LOW POWER SCHOTTKY

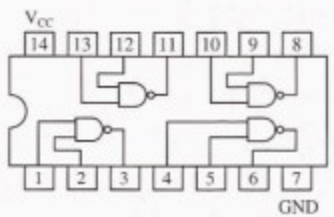
J SUFFIX
CERAMIC
CASE 632-08

N SUFFIX
PLASTIC
CASE 646-06

D SUFFIX
SOIC
CASE 751A-02

ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC



SN54/74LS00

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Typ	Max	Unit	Test Conditions
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74			0.8		
V_{IK}	Input Clamp Diode Voltage			–0.65	–1.5	V	V_{CC} = MIN, I_{IN} = –18 mA
V_{OH}	Ouput HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table
		74	2.7	3.5		V	
V_{OL}	Ouput LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V
I_{IL}	Input LOW Current				–0.4	mA	V_{CC} = MAX, I_N = 0.4 V
I_{OS}	Short Circuit Current (Note 1)		–20		–100	mA	V_{CC} = MAX
I_{CC}	Power Supply Current Total, Output HIGH				1.6	mA	V_{CC} = MAX
	Total, Output LOW				4.4		

NOTE 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

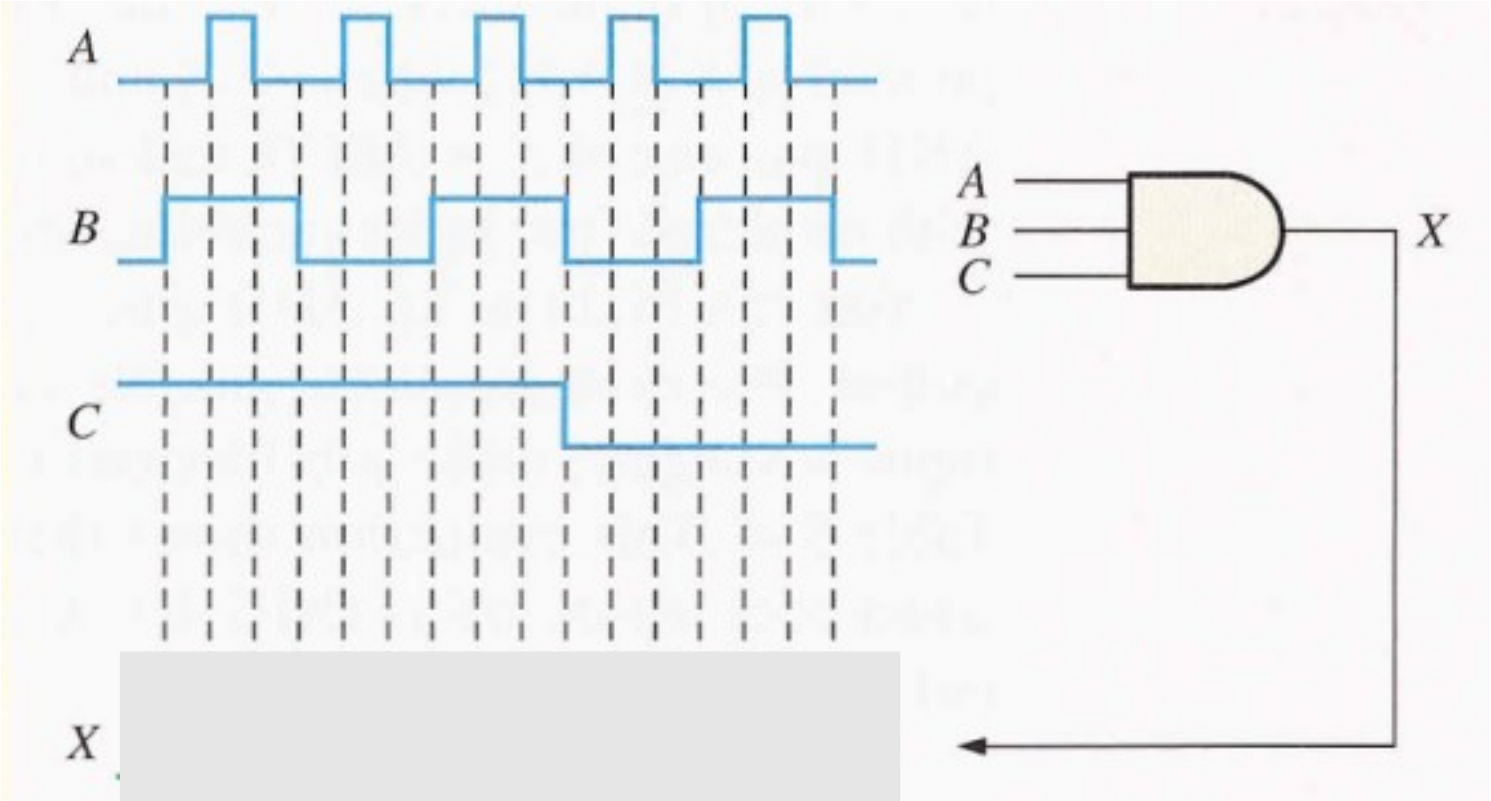
Symbol	Parameter	Limits Min	Typ	Max	Unit	Test Conditions
t_{PLH}	Turn-Off Delay, Input to Output		9.0	15	ns	V_{CC} = 5.0 V, C_L = 15 pF
t_{PHL}	Turn-On Delay, Input to Output		10	15	ns	

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T_A	Operating Ambient Temperature Range	54	–55	25	125	°C
		74	0	25	70	
I_{OH}	Output Current — High	54, 74			–0.4	mA
I_{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

Home Work

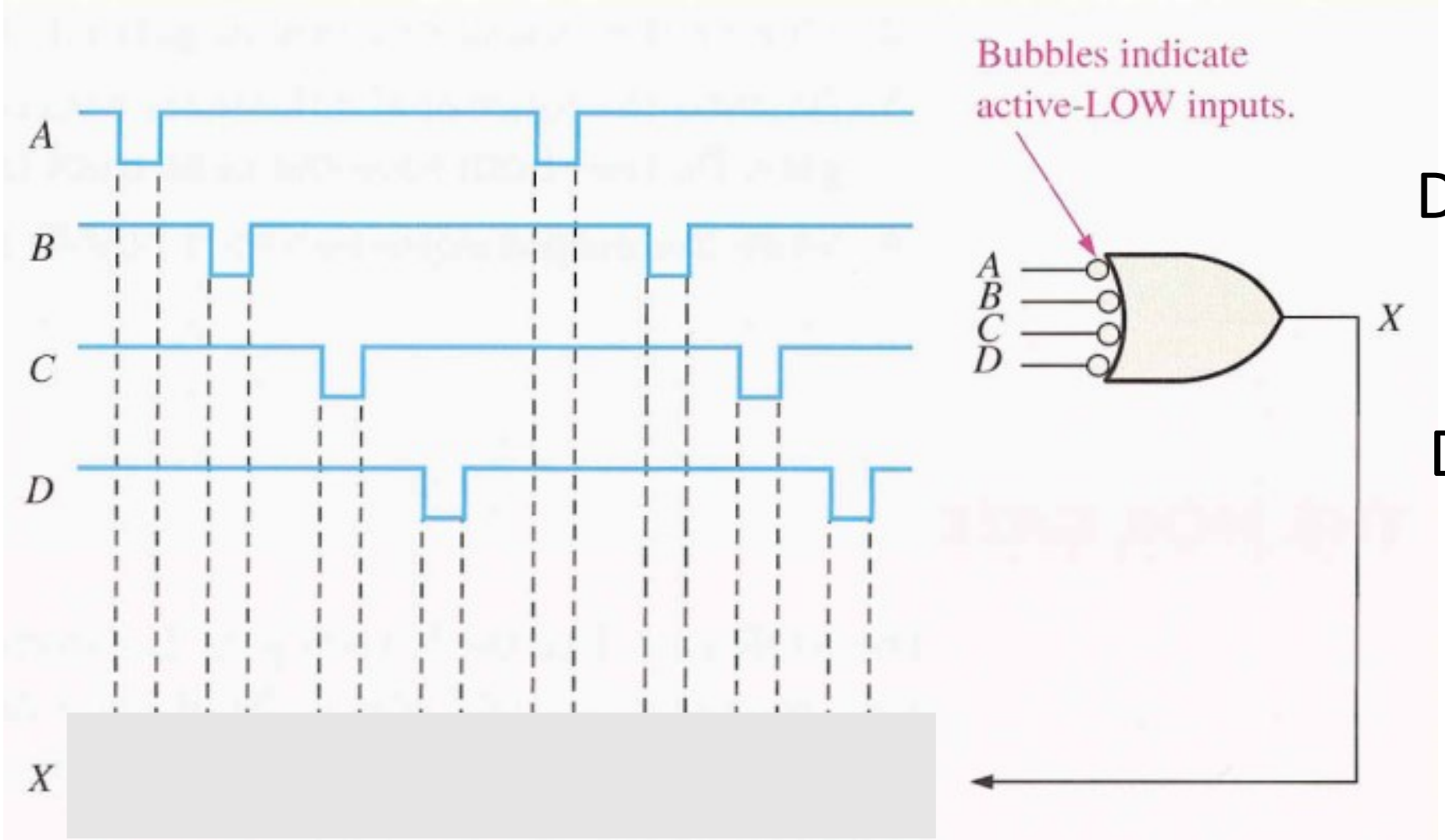
1



DRAW THE OUTPUT TIMING DIAGRAM

Home Work

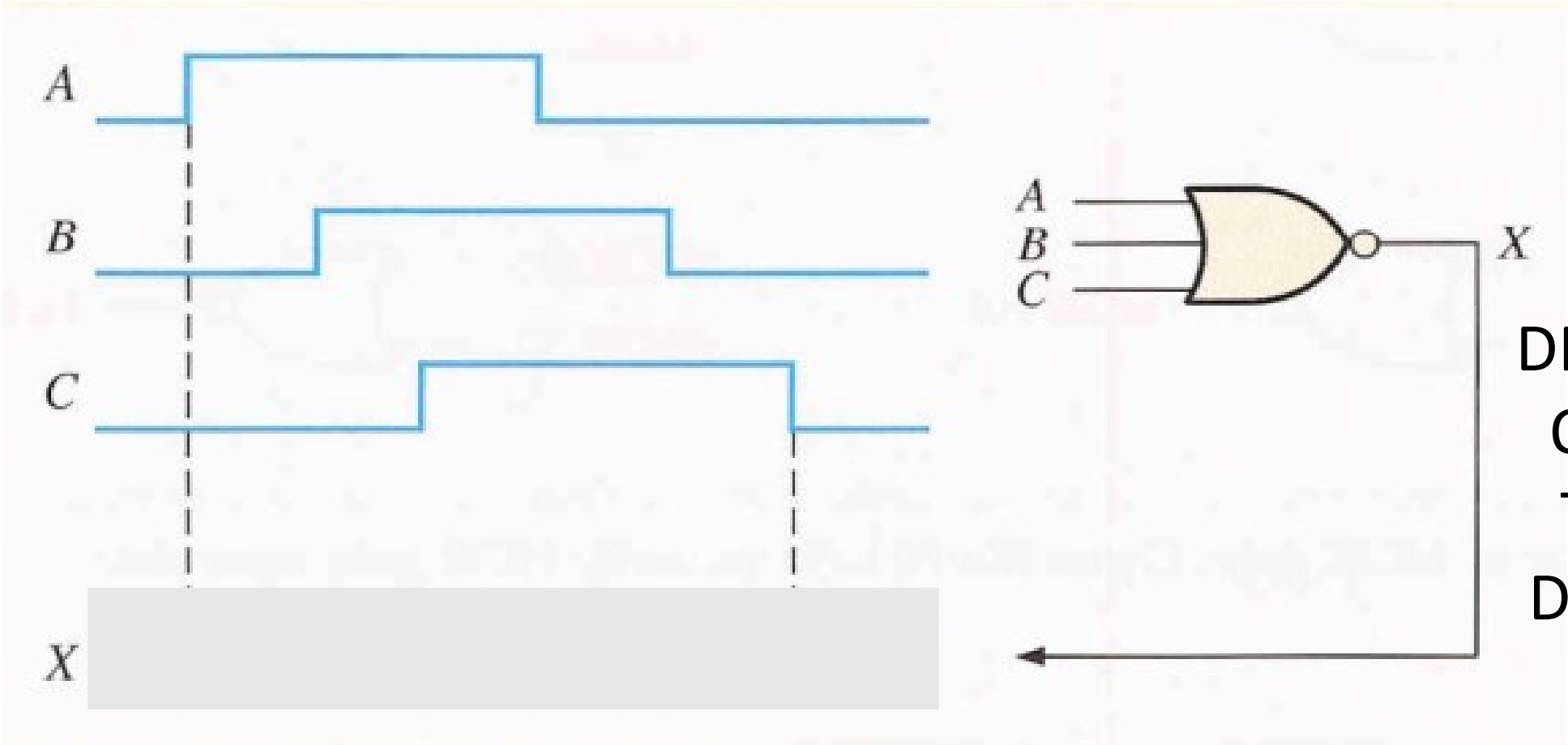
2



DRAW THE OUTPUT TIMING DIAGRAM

Home Work

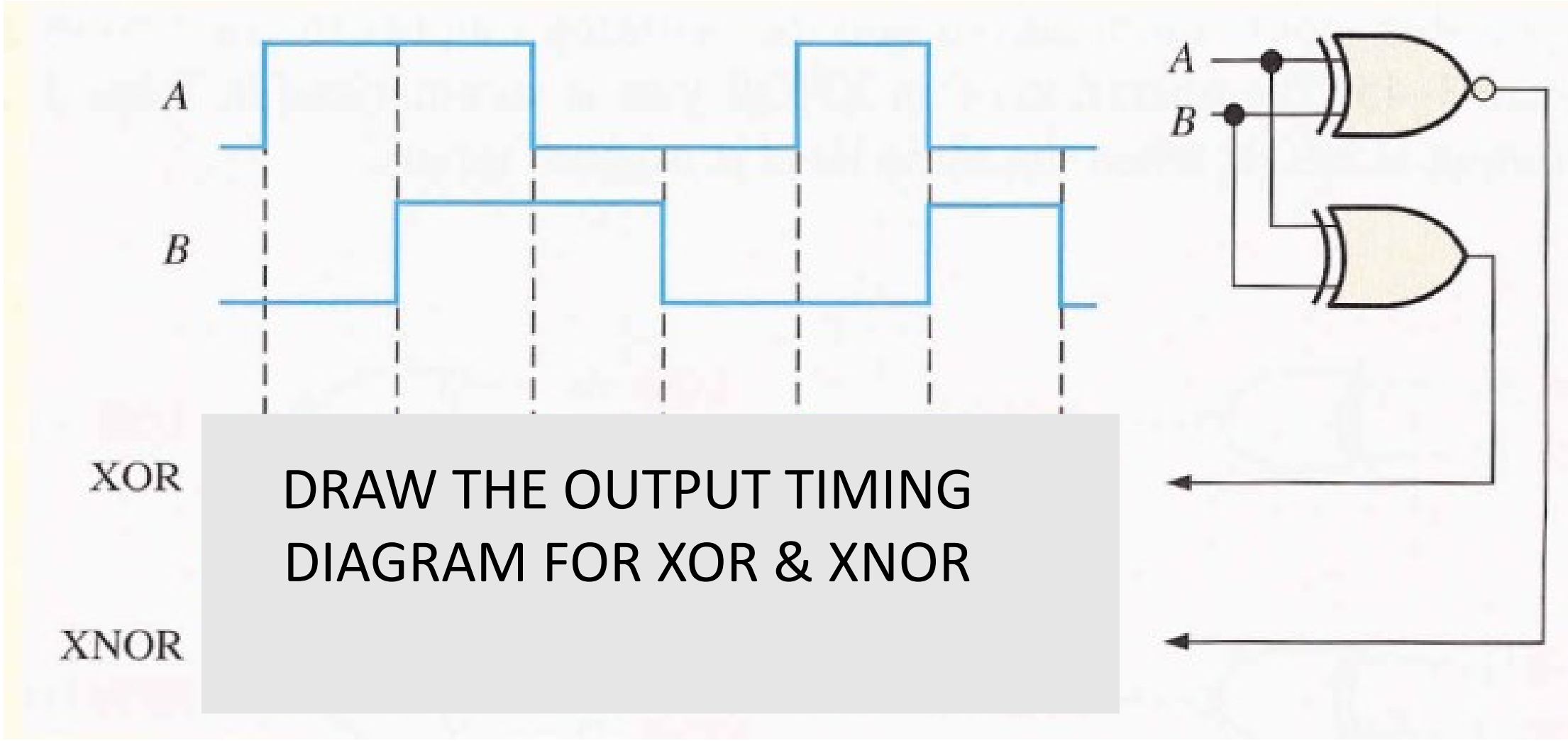
3



DRAW THE OUTPUT TIMING DIAGRAM

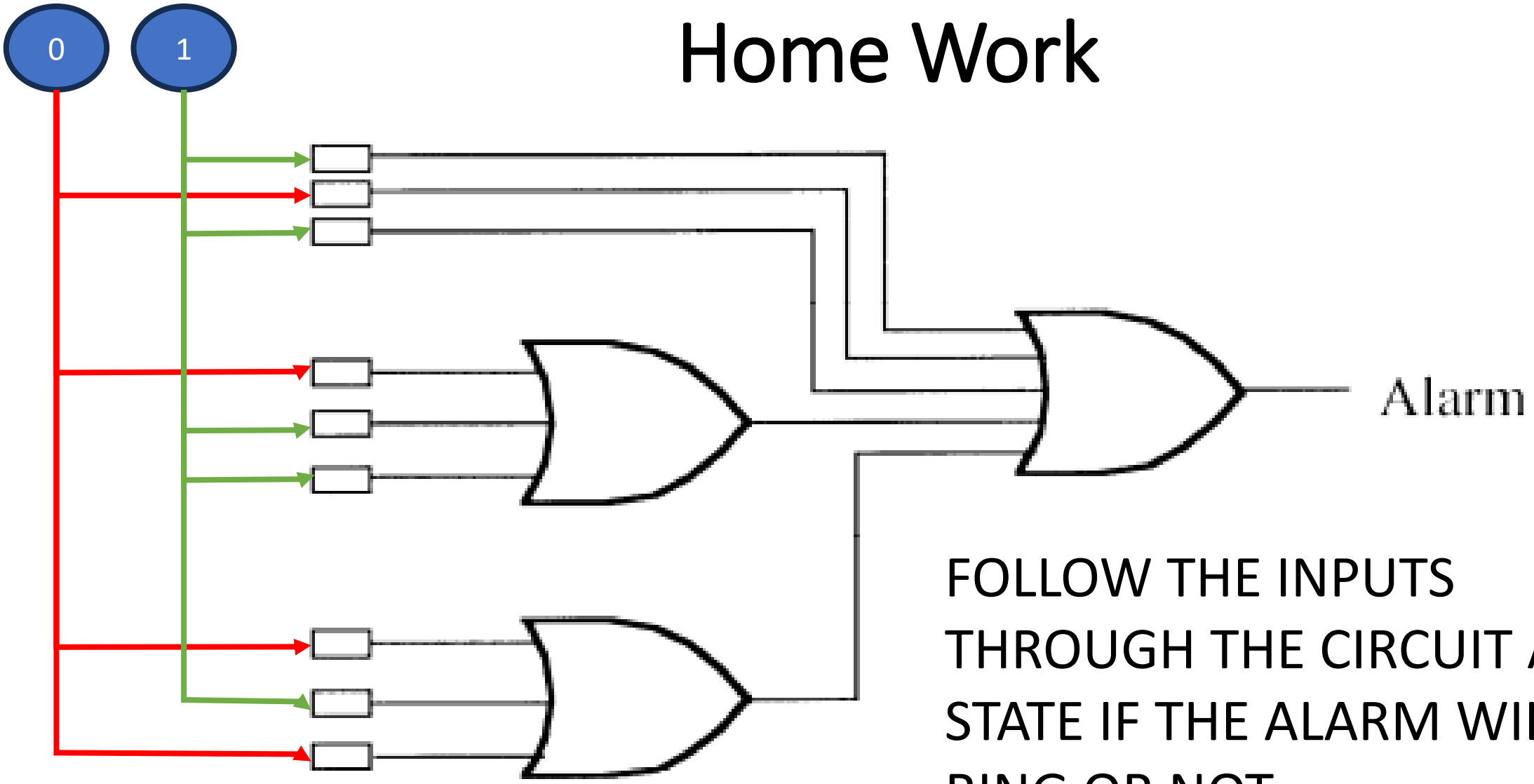
Home Work

4



DRAW THE OUTPUT TIMING DIAGRAM FOR XOR & XNOR

Home Work

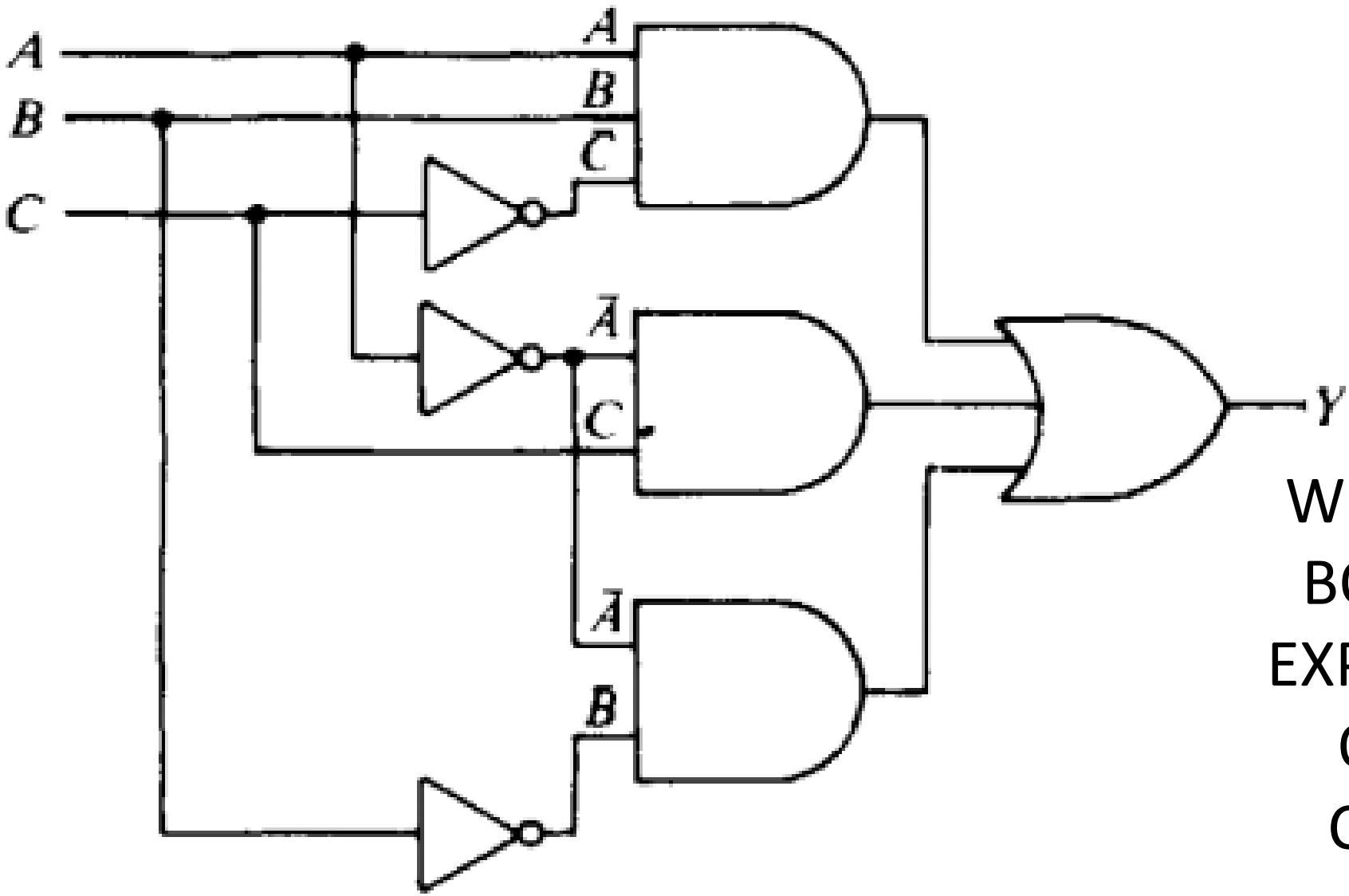


5

FOLLOW THE INPUTS THROUGH THE CIRCUIT AND STATE IF THE ALARM WILL RING OR NOT

Home Work

6



WRITE THE BOOLEAN EXPRESSION OF THE CIRCUIT

University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

Lec. Dr. Taif Alawsi

Lec. 4: Boolean Algebra and Logic Simplification : 2024-Oct-20

Lecture Outline

1. Boolean Algebra
2. DeMorgan's Theorems
3. SOP & POS & Truth Table
4. K-Map
5. HW

Simple properties

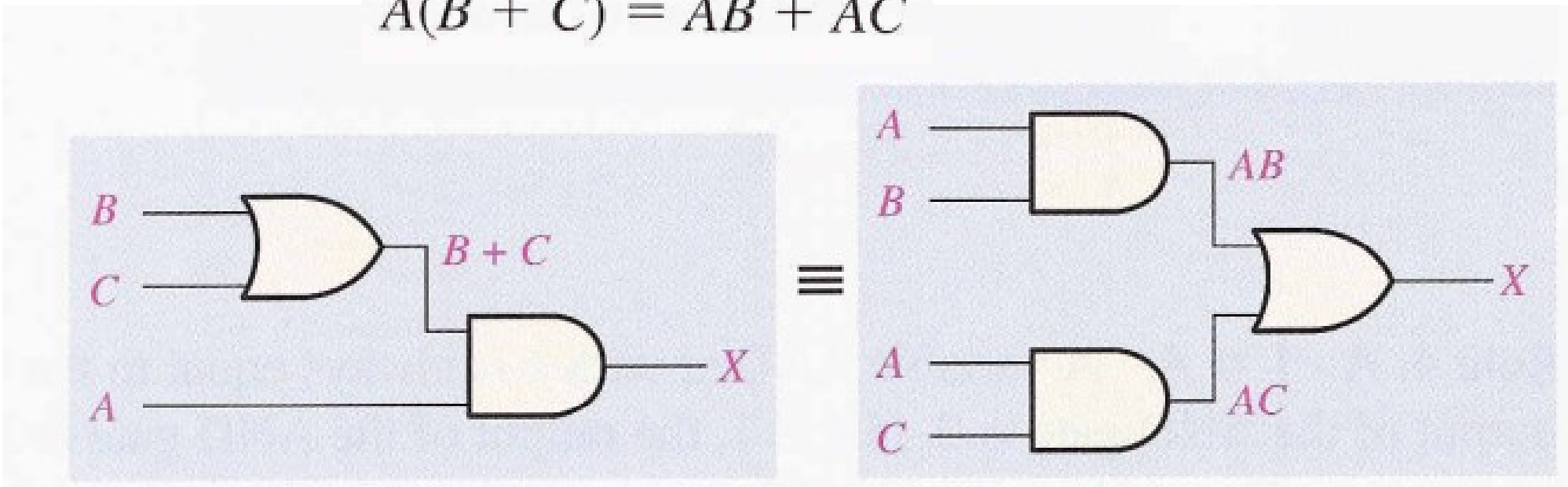
$$A + B = B + A$$

$$AB = BA$$

$$A + (B + C) = (A + B) + C$$

$$A(BC) = (AB)C$$

$$A(B + C) = AB + AC$$



Boolean Rules

1. $A + 0 = A$

2. $A + 1 = 1$

3. $A \cdot 0 = 0$

4. $A \cdot 1 = A$

5. $A + A = A$

6. $A + \overline{A} = 1$

7. $A \cdot A = A$

8. $A \cdot \overline{A} = 0$

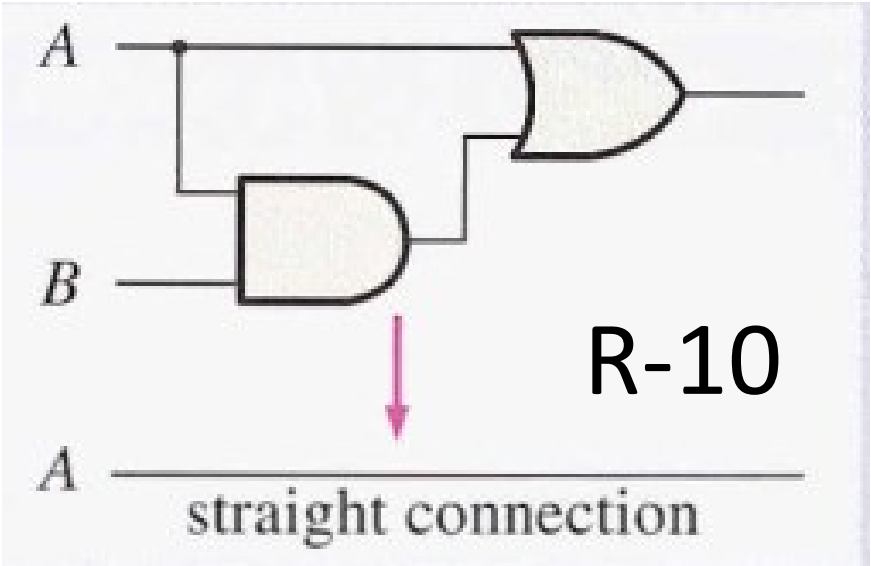
9. $\overline{\overline{A}} = A$

10. $A + AB = A$

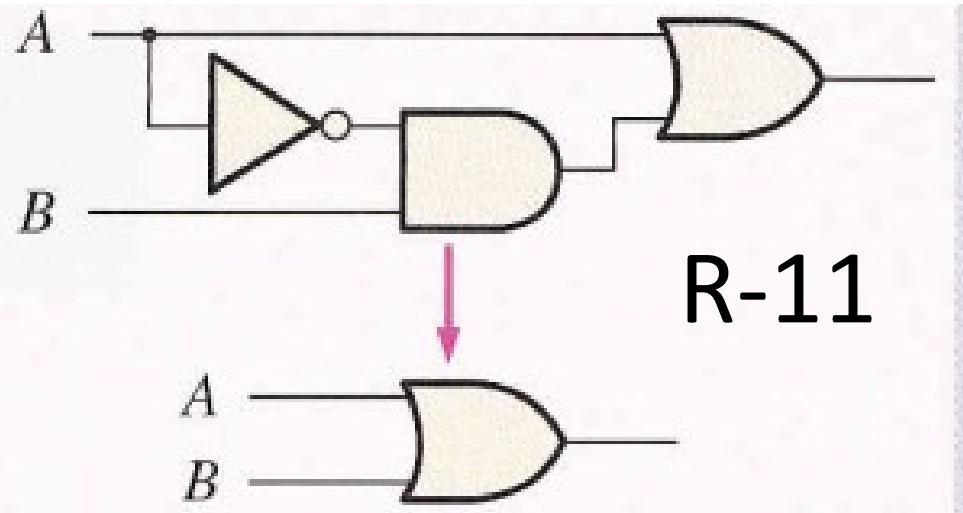
11. $A + \overline{A}B = A + B$

12. $(A + B)(A + C) = A + BC$

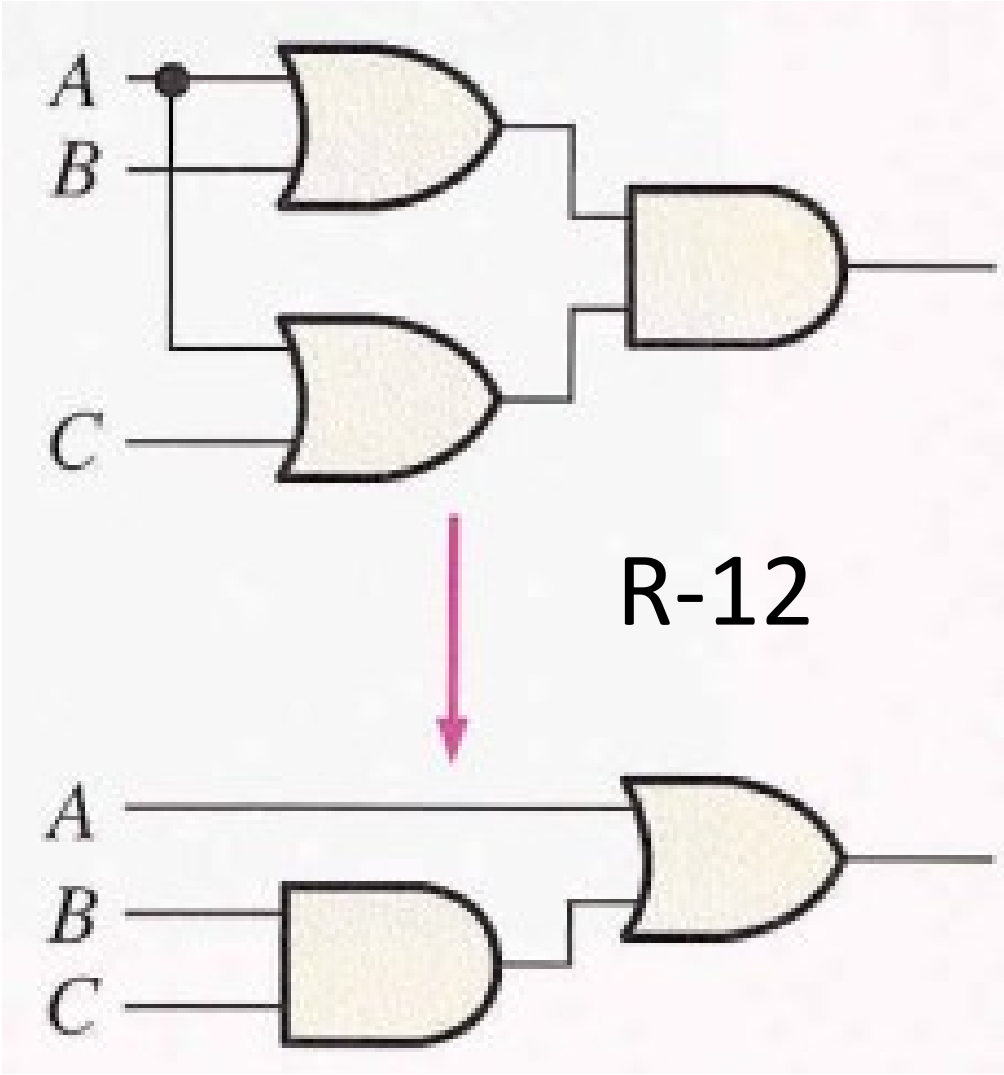
Rules Simplify Logic



R-10



R-11



R-12

De Morgan's Theorems

Augustus De Morgan

1 $\overline{XY} = \overline{X} + \overline{Y}$

2 $\overline{X + Y} = \overline{X}\,\overline{Y}$

Applications of De Morgan's Theorems

Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X + Y + Z}$.

EX-1

ANS

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$$

EX-2

Apply DeMorgan's theorems to each of the following expressions:

(a) $\overline{(A + B + C)D}$ **(b)** $\overline{ABC + DEF}$ **(c)** $\overline{A\overline{B} + \overline{C}D + EF}$

ANS

(a) Let $A + B + C = X$ and $D = Y$. The expression $\overline{(A + B + C)D}$ is of the form $\overline{XY} = \overline{X} + \overline{Y}$ and can be rewritten as

$$\overline{(A + B + C)D} = \overline{A + B + C} + \overline{D}$$

Next, apply DeMorgan's theorem to the term $\overline{A + B + C}$.

$$\overline{A + B + C} + \overline{D} = \overline{A}\,\overline{B}\,\overline{C} + \overline{D}$$

(b) Let $ABC = X$ and $DEF = Y$. The expression $\overline{ABC + DEF}$ is of the form $\overline{X + Y} = \overline{X}\,\overline{Y}$ and can be rewritten as

$$\overline{ABC + DEF} = (\overline{ABC})(\overline{DEF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{ABC}$ and $\overline{DEF}$.

$$(\overline{ABC})(\overline{DEF}) = (\overline{A} + \overline{B} + \overline{C})(\overline{D} + \overline{E} + \overline{F})$$

(c) Let $A\overline{B} = X$, $\overline{C}D = Y$, and $EF = Z$. The expression $\overline{A\overline{B} + \overline{C}D + EF}$ is of the form $\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$ and can be rewritten as

$$\overline{A\overline{B} + \overline{C}D + EF} = (\overline{A\overline{B}})(\overline{\overline{C}D})(\overline{EF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{A\overline{B}}$, $\overline{\overline{C}D}$, and $\overline{EF}$.

$$(\overline{A\overline{B}})(\overline{\overline{C}D})(\overline{EF}) = (\overline{A} + B)(C + \overline{D})(\overline{E} + \overline{F})$$

EX-3

Using Boolean algebra techniques, simplify this expression:

$$AB + A(B + C) + B(B + C)$$

ANS

Step 1: Apply the distributive law to the second and third terms in the expression, as follows:

$$AB + AB + AC + BB + BC$$

Step 2: Apply rule 7 ($BB = B$) to the fourth term.

$$AB + AB + AC + B + BC$$

Step 3: Apply rule 5 ($AB + AB = AB$) to the first two terms.

$$AB + AC + B + BC$$

Step 4: Apply rule 10 ($B + BC = B$) to the last two terms.

$$AB + AC + B$$

Step 5: Apply rule 10 ($AB + B = B$) to the first and third terms.

$$B + AC$$

Sum of Products (SOP)

$$AB + ABC$$

$$ABC + CDE + \overline{B}C\overline{D}$$

EX-4

Determine the binary values for which the following standard SOP expression is equal to 1:

$$ABCD + A\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$

ANS

The term $ABCD$ is equal to 1 when $A = 1$, $B = 1$, $C = 1$, and $D = 1$.

$$ABCD = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

The term $A\overline{B}\,\overline{C}D$ is equal to 1 when $A = 1$, $B = 0$, $C = 0$, and $D = 1$.

$$A\overline{B}\,\overline{C}D = 1 \cdot \overline{0} \cdot \overline{0} \cdot 1 = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

The term $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$ is equal to 1 when $A = 0$, $B = 0$, $C = 0$, and $D = 0$.

$$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} = \overline{0} \cdot \overline{0} \cdot \overline{0} \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

The SOP expression equals 1 when any or all of the three product terms is 1.

Product of Sums (POS)

$$(\overline{A} + B)(A + \overline{B} + C)$$

$$(\overline{A} + \overline{B} + \overline{C})(C + \overline{D} + E)(\overline{B} + C + D)$$

$$(A + B)(A + \overline{B} + C)(\overline{A} + C)$$

Truth Table Generation from SOP

Develop a truth table for the standard SOP expression $\overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + ABC$.

INPUTS A	B	C	OUTPUT X	PRODUCT TERM
0	0	0	0	
0	0	1	1	$\overline{A}\,\overline{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\overline{B}\,\overline{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Truth Table Generation from POS

Determine the truth table for the following standard POS expression:

$$(A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)$$

INPUTS A	B	C	OUTPUT X	SUM TERM
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \overline{B} + C)$
0	1	1	0	$(A + \overline{B} + \overline{C})$
1	0	0	1	
1	0	1	0	$(\overline{A} + B + \overline{C})$
1	1	0	0	$(\overline{A} + \overline{B} + C)$
1	1	1	1	

SOP & POS Generation from Truth Table

INPUTS			OUTPUT
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$011 \longrightarrow \overline{A}BC$
$100 \longrightarrow A\overline{B}\,\overline{C}$
$110 \longrightarrow AB\overline{C}$
$111 \longrightarrow ABC$

SOP

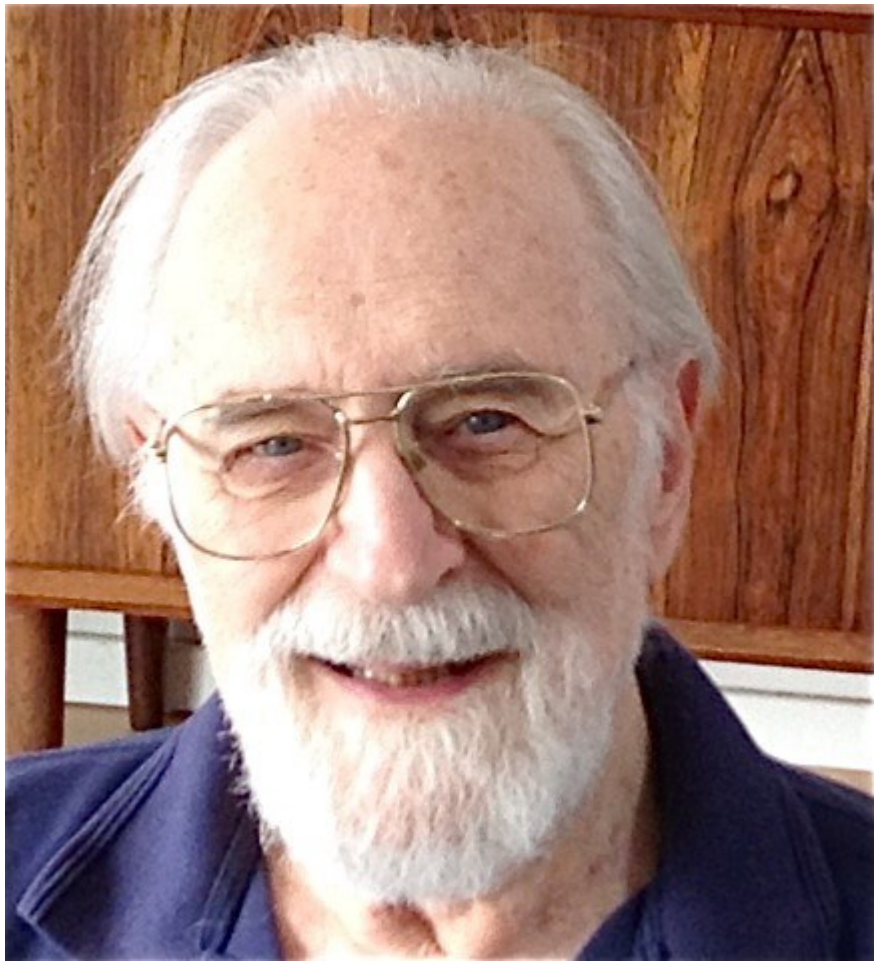
$$X = \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

$000 \longrightarrow A + B + C$
$001 \longrightarrow A + B + \overline{C}$
$010 \longrightarrow A + \overline{B} + C$
$101 \longrightarrow \overline{A} + B + \overline{C}$

POS

$$X = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + \overline{C})$$

Karnaugh Map (K-Map)

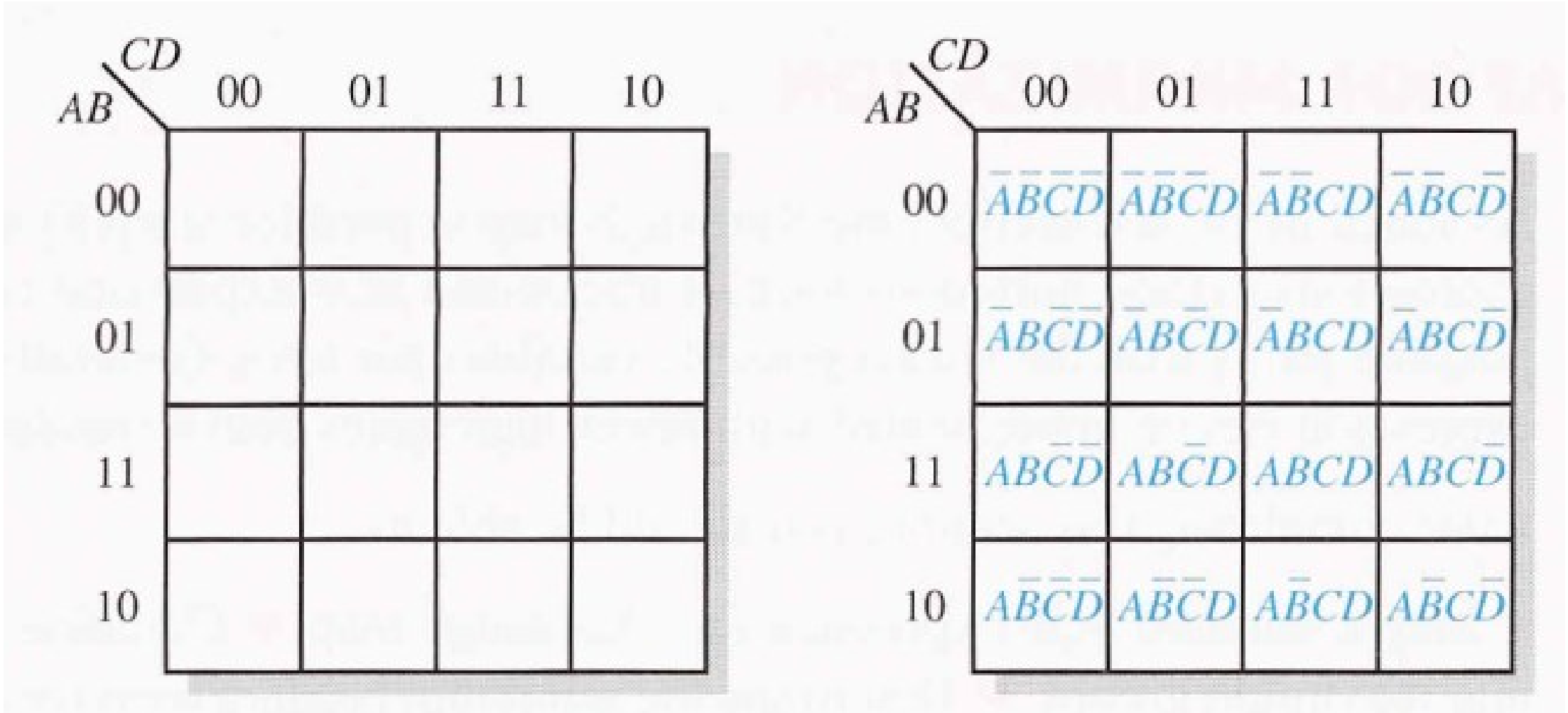


Maurice Karnaugh

AB \ C	0	1
00		
01		
11		
10		

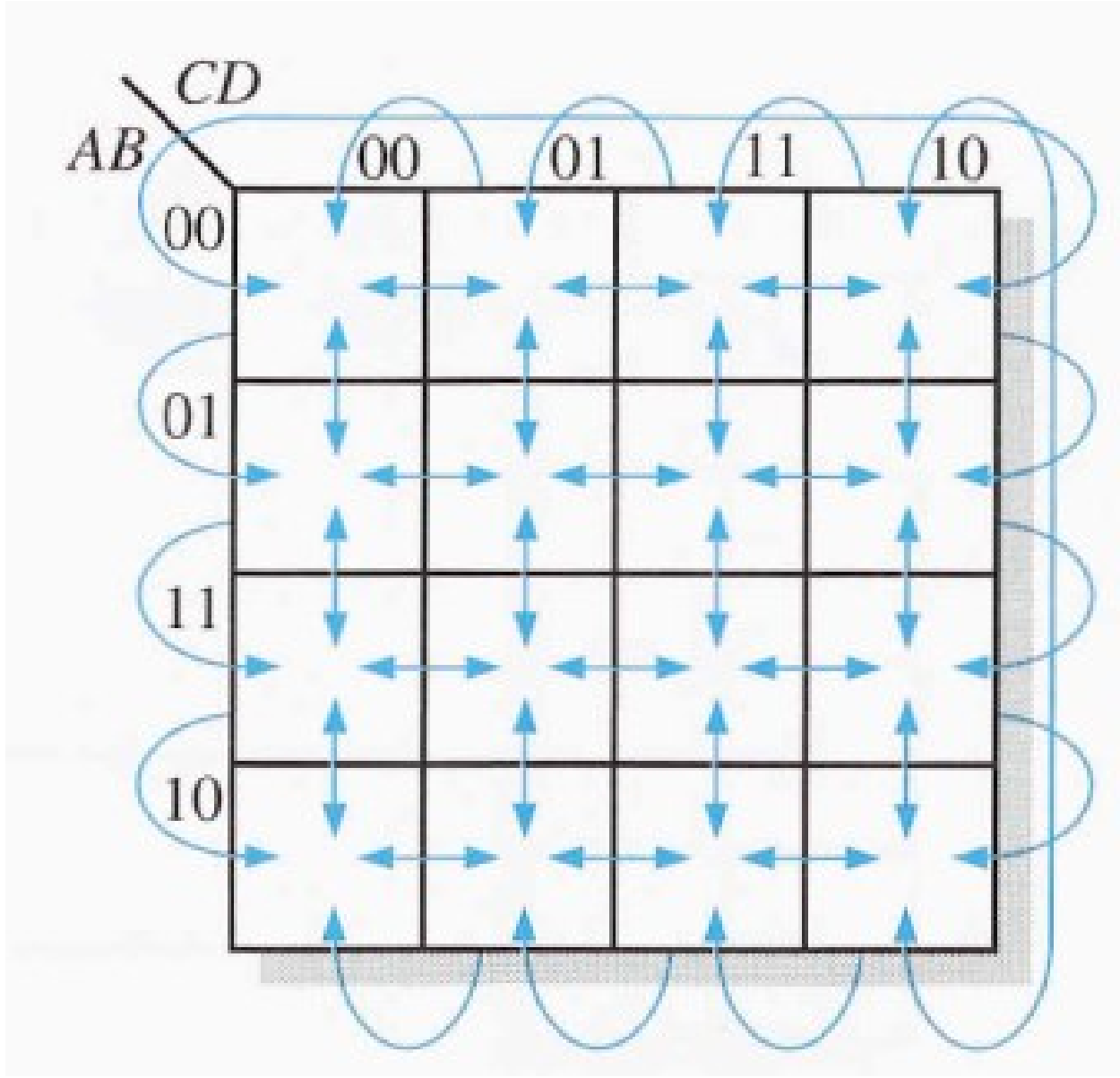
AB \ C	0	1
00	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$
01	$\bar{A}B\bar{C}$	$\bar{A}BC$
11	$AB\bar{C}$	ABC
10	$A\bar{B}\bar{C}$	$A\bar{B}C$

4-Input K-Map

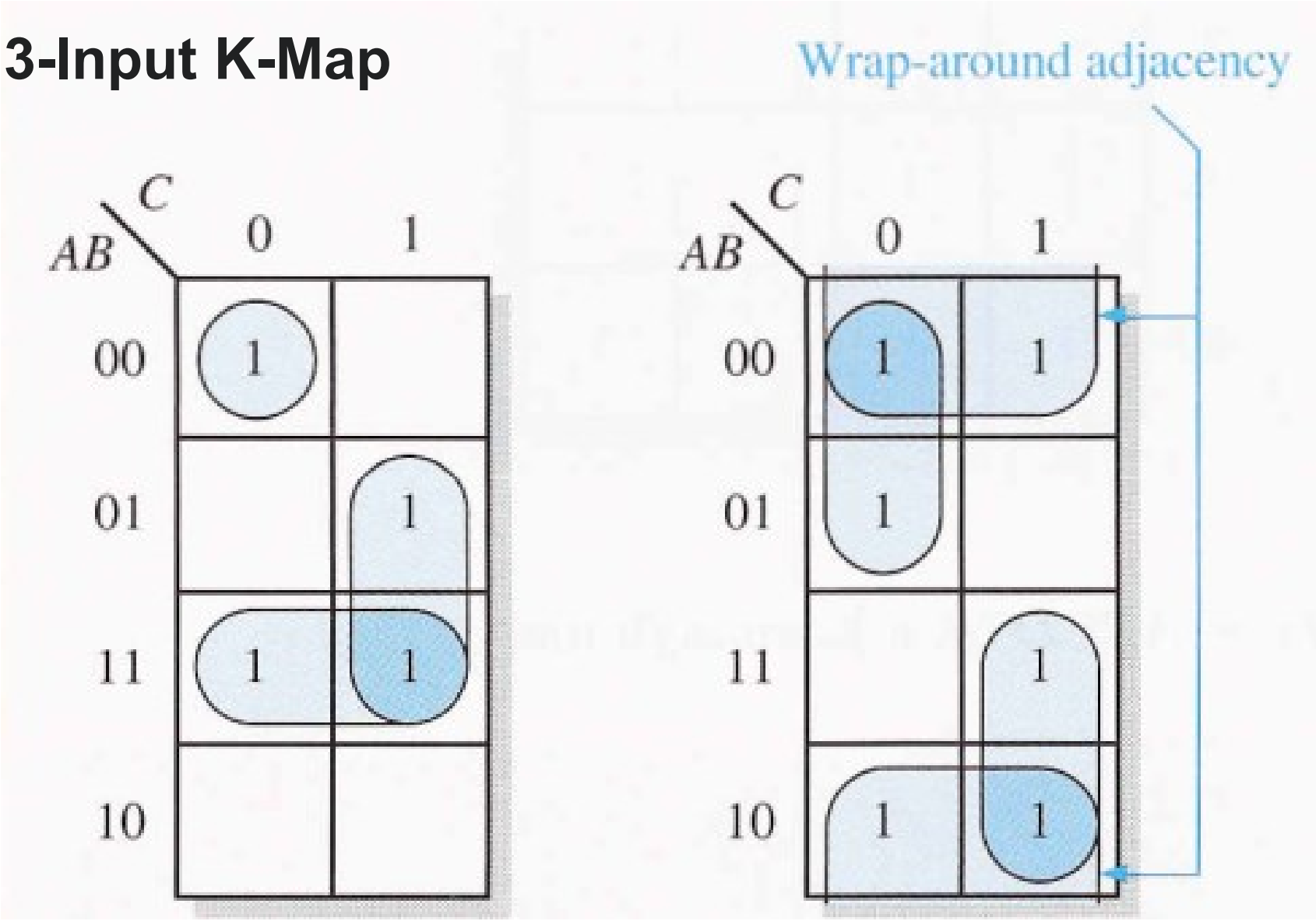

CD
AB
00
01
11
10
00
01
11
10

CD
AB
00
01
11
10
$\bar{A}\bar{B}\bar{C}\bar{D}$ $\bar{A}\bar{B}\bar{C}D$ $\bar{A}\bar{B}CD$ $\bar{A}\bar{B}C\bar{D}$
$\bar{A}B\bar{C}\bar{D}$ $\bar{A}B\bar{C}D$ $\bar{A}BCD$ $\bar{A}BC\bar{D}$
$AB\bar{C}\bar{D}$ $AB\bar{C}D$ $ABCD$ $ABC\bar{D}$
$A\bar{B}\bar{C}\bar{D}$ $A\bar{B}\bar{C}D$ $A\bar{B}CD$ $A\bar{B}C\bar{D}$

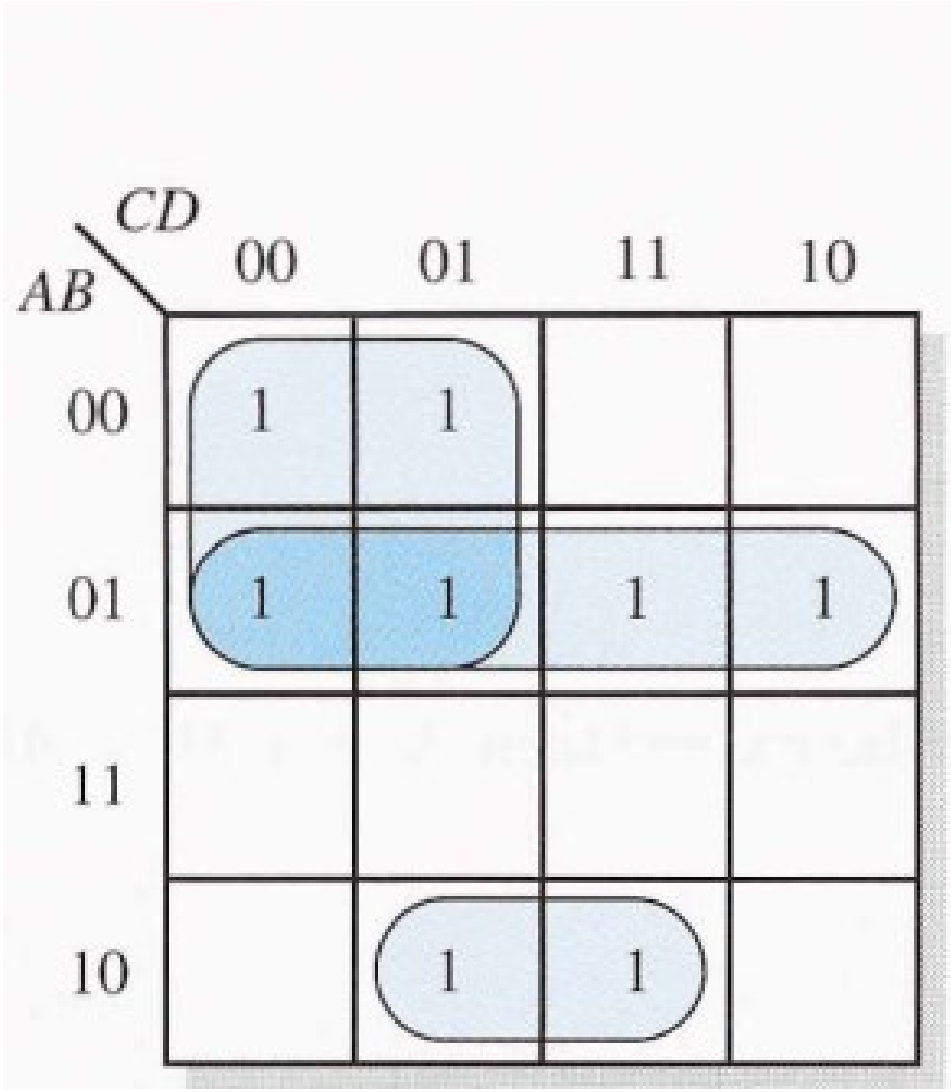
Cell Adjacency

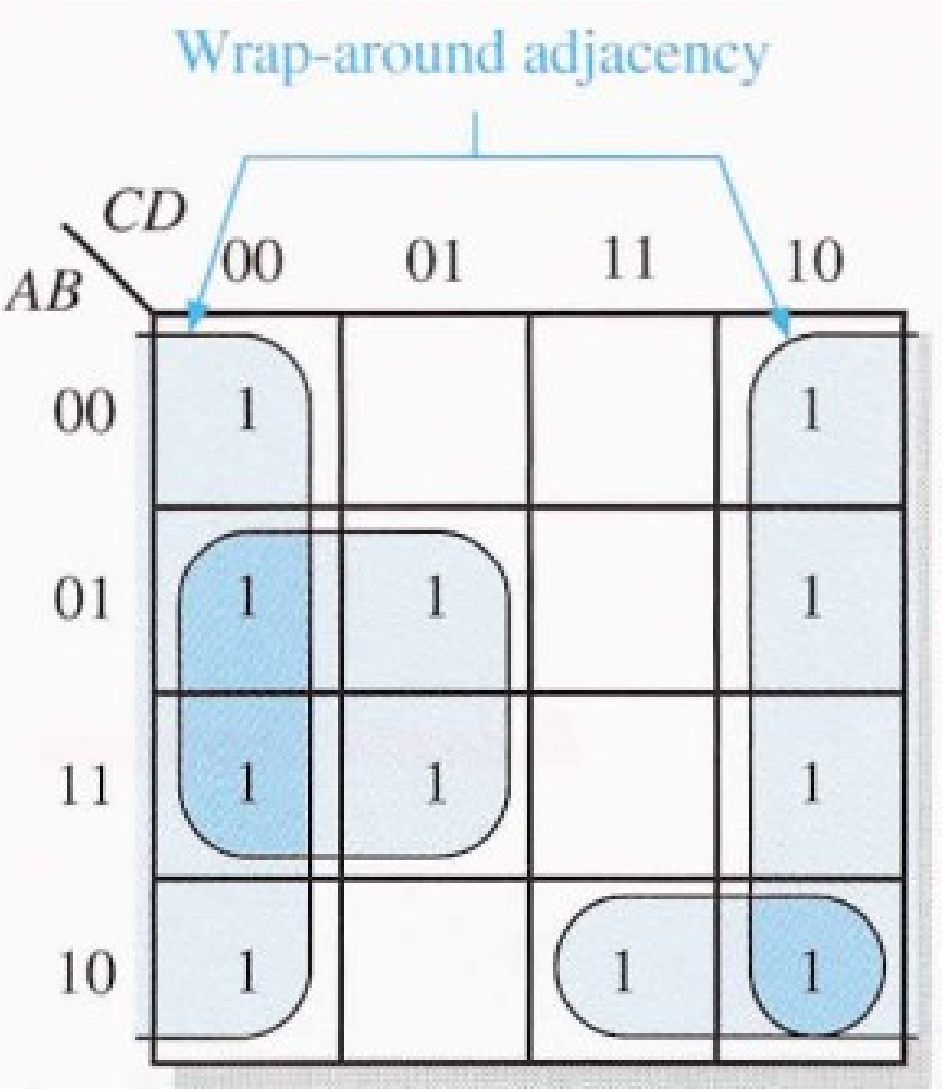


Solving 3-Input K-Map



Solving 4-Input K-Map





K-Map Rules for SOP

Determine the minimum product term for each group.

a. For a 3-variable map:

(**1**) A 1-cell group yields a 3-variable product term

(**2**) A 2-cell group yields a 2-variable product term

(**3**) A 4-cell group yields a 1-variable term

(**4**) An 8-cell group yields a value of 1 for the expression

b. For a 4-variable map:

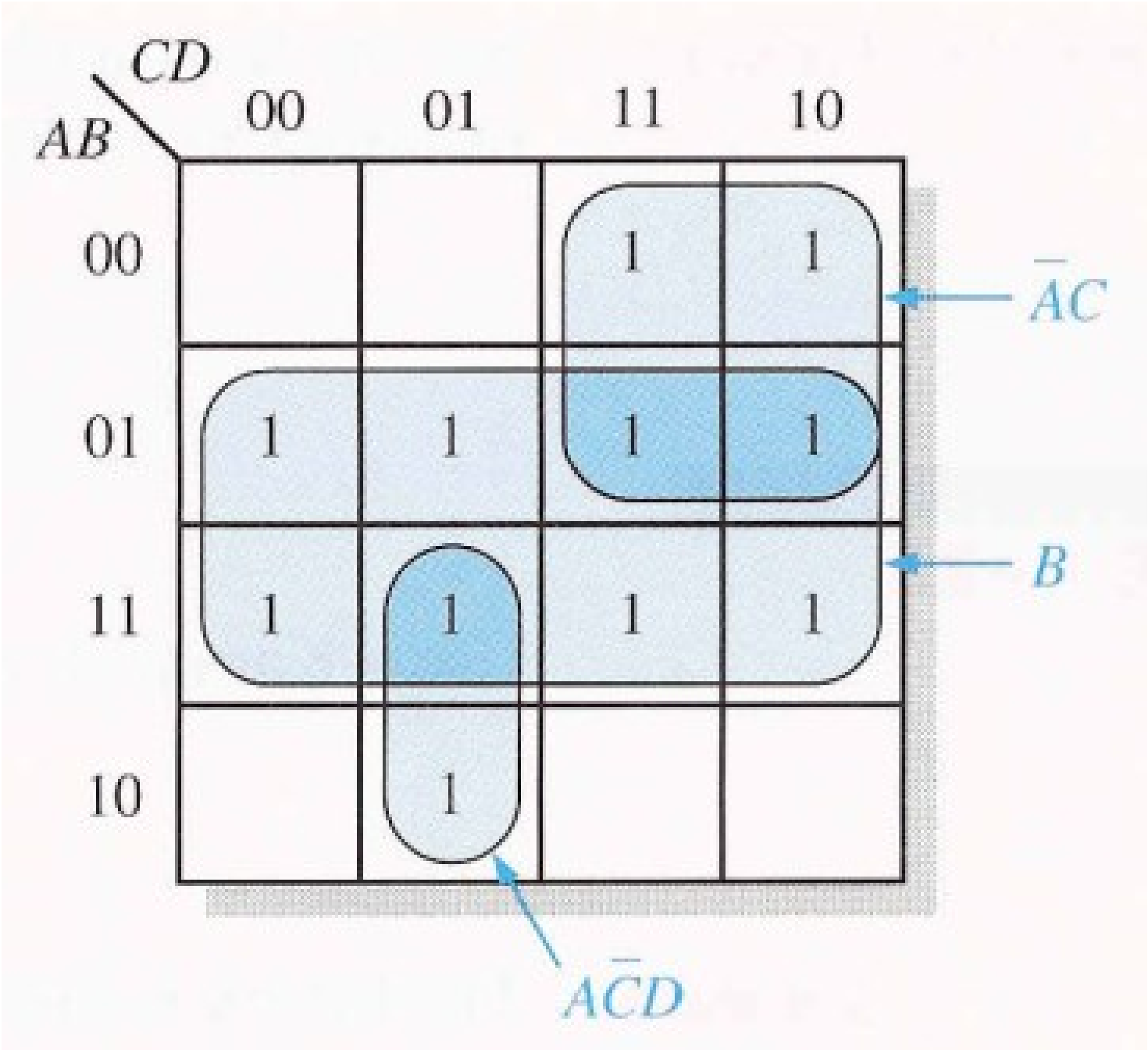
(**1**) A 1-cell group yields a 4-variable product term

(**2**) A 2-cell group yields a 3-variable product term

(**3**) A 4-cell group yields a 2-variable product term

(**4**) An 8-cell group yields a 1-variable term

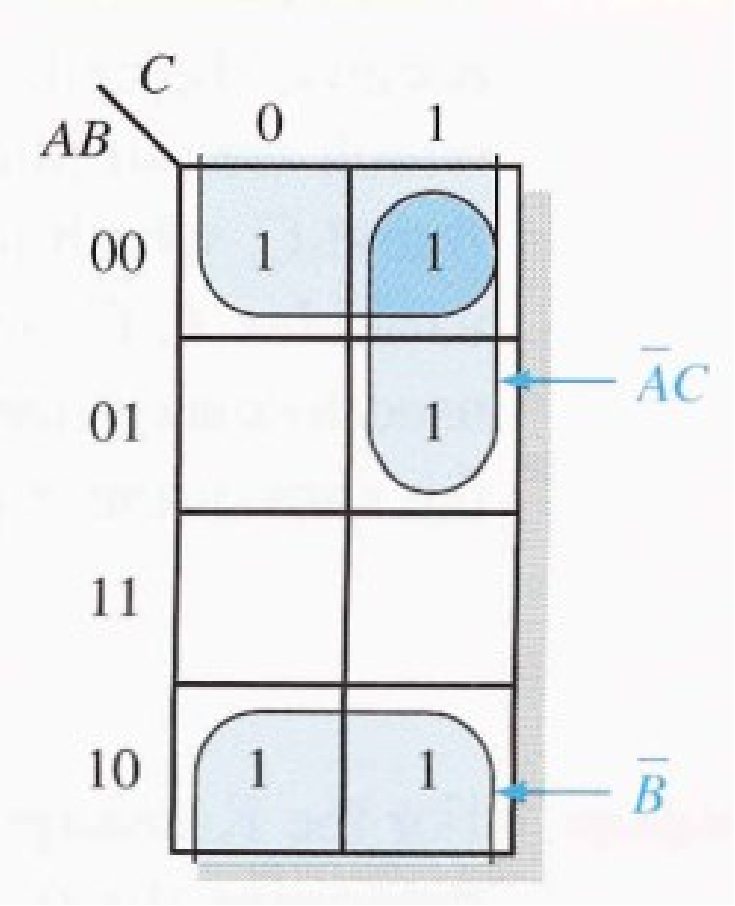
(**5**) A 16-cell group yields a value of 1 for the expression



SOP from K-Map

$$B + \overline{A}C + A\overline{C}D$$

SOP Minimization Using 3-Input K-Map



Use a Karnaugh map to minimize the following standard SOP expression:

$$A\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C}$$

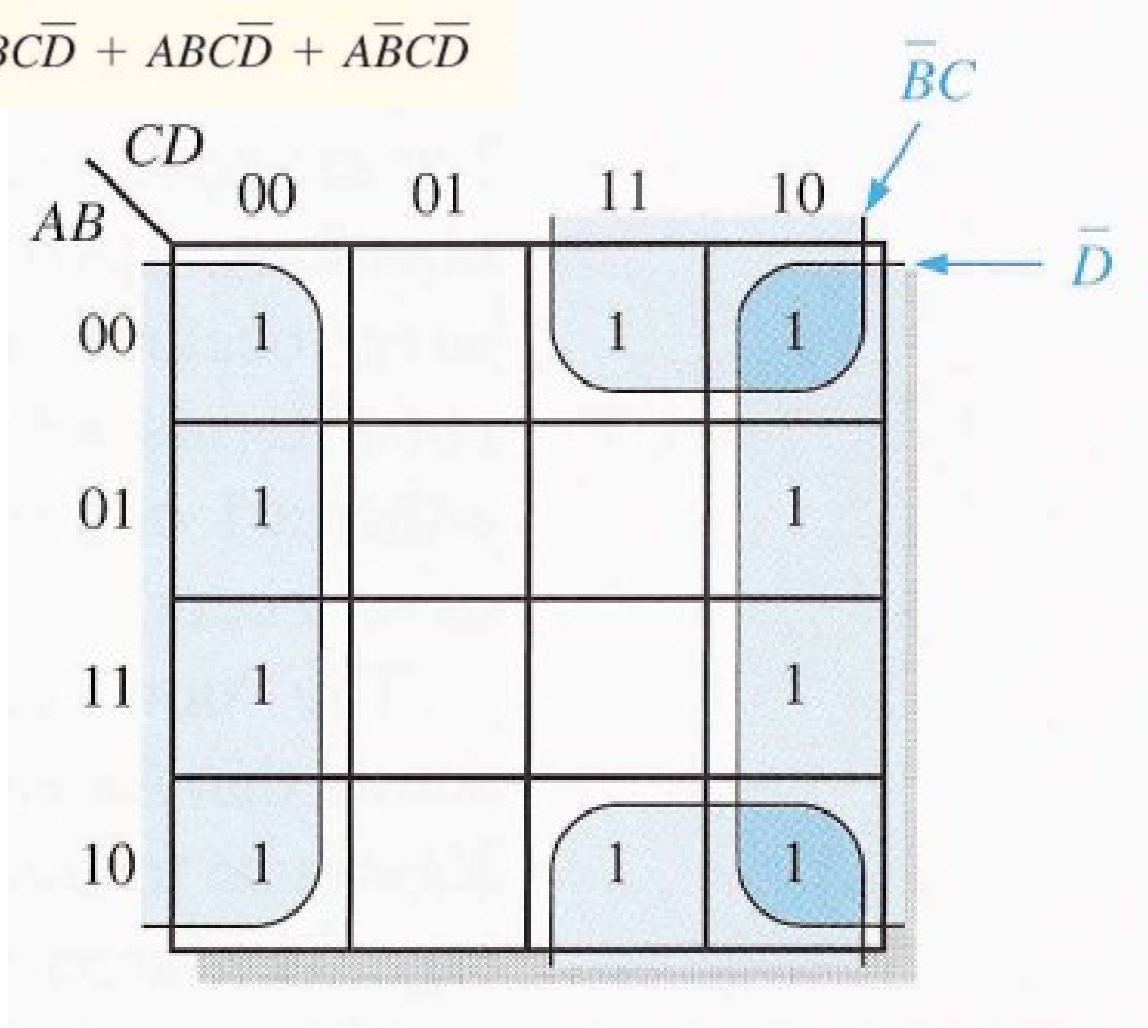
$$\overline{B} + \overline{A}C$$

SOP Minimization Using 4-Input K-Map

Use a Karnaugh map to minimize the following SOP expression:

$$\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D}$$

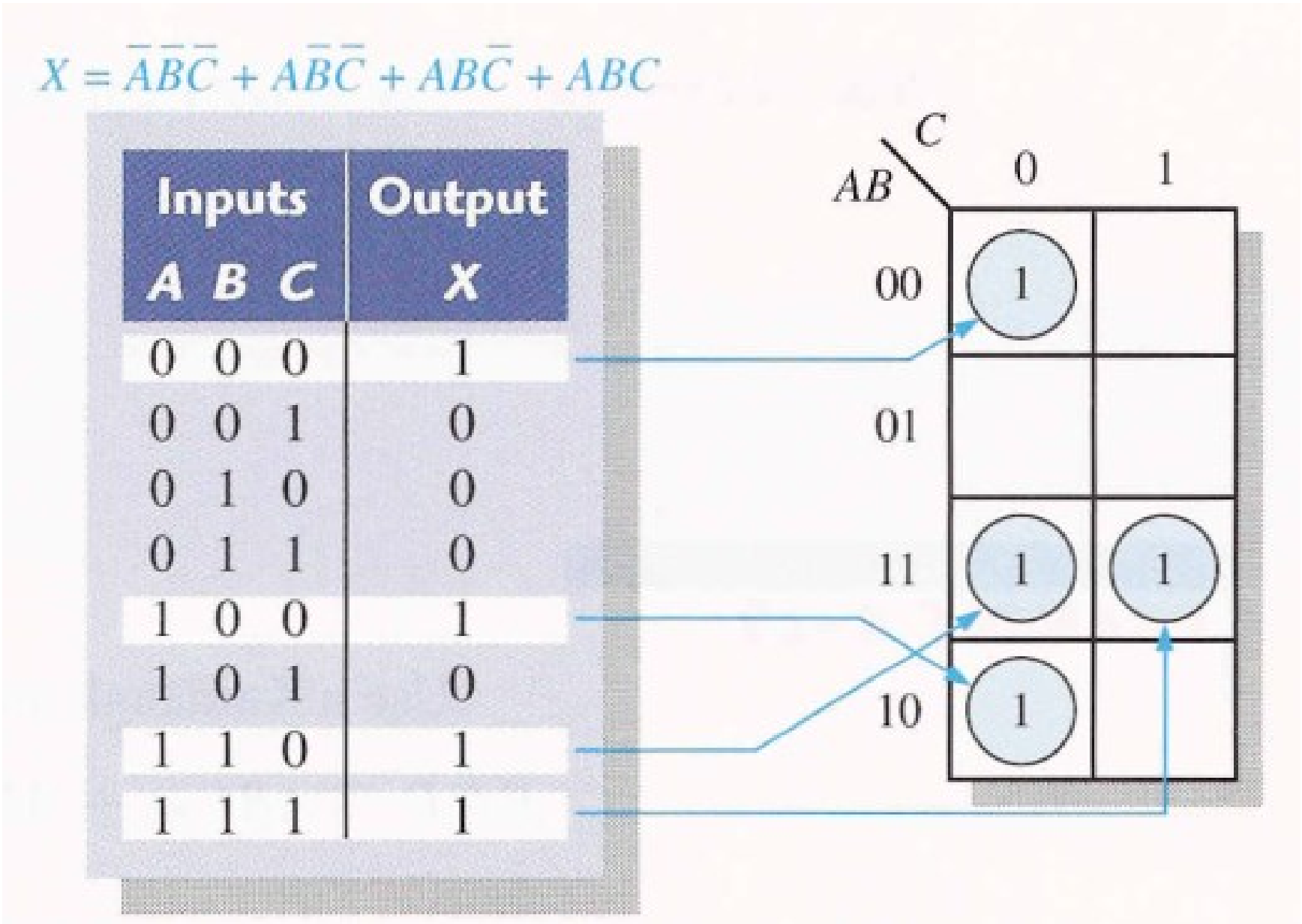
$$\overline{D} + \overline{B}C$$



Truth Table & K-Map

$$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

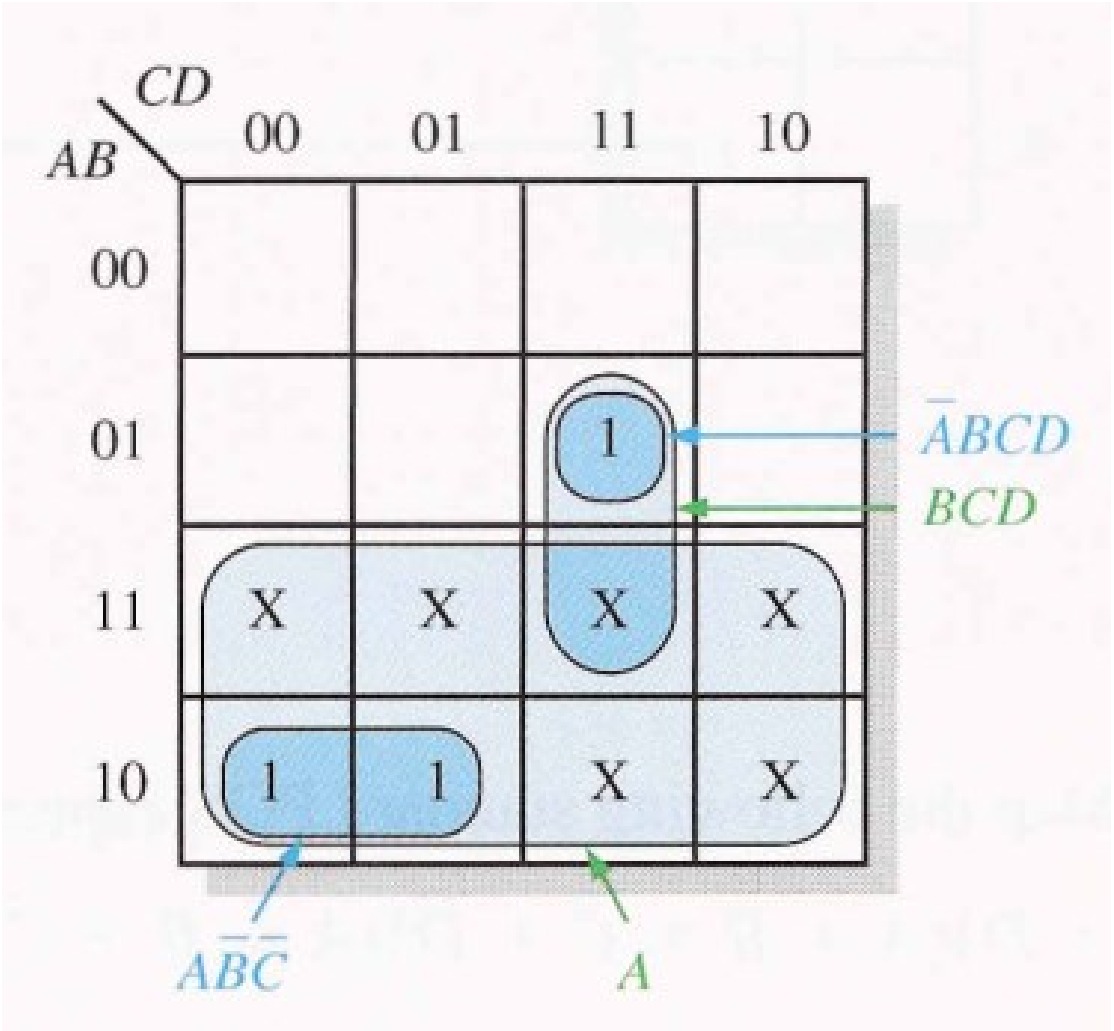
Inputs			Output
A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



“Don’t Care” Condition

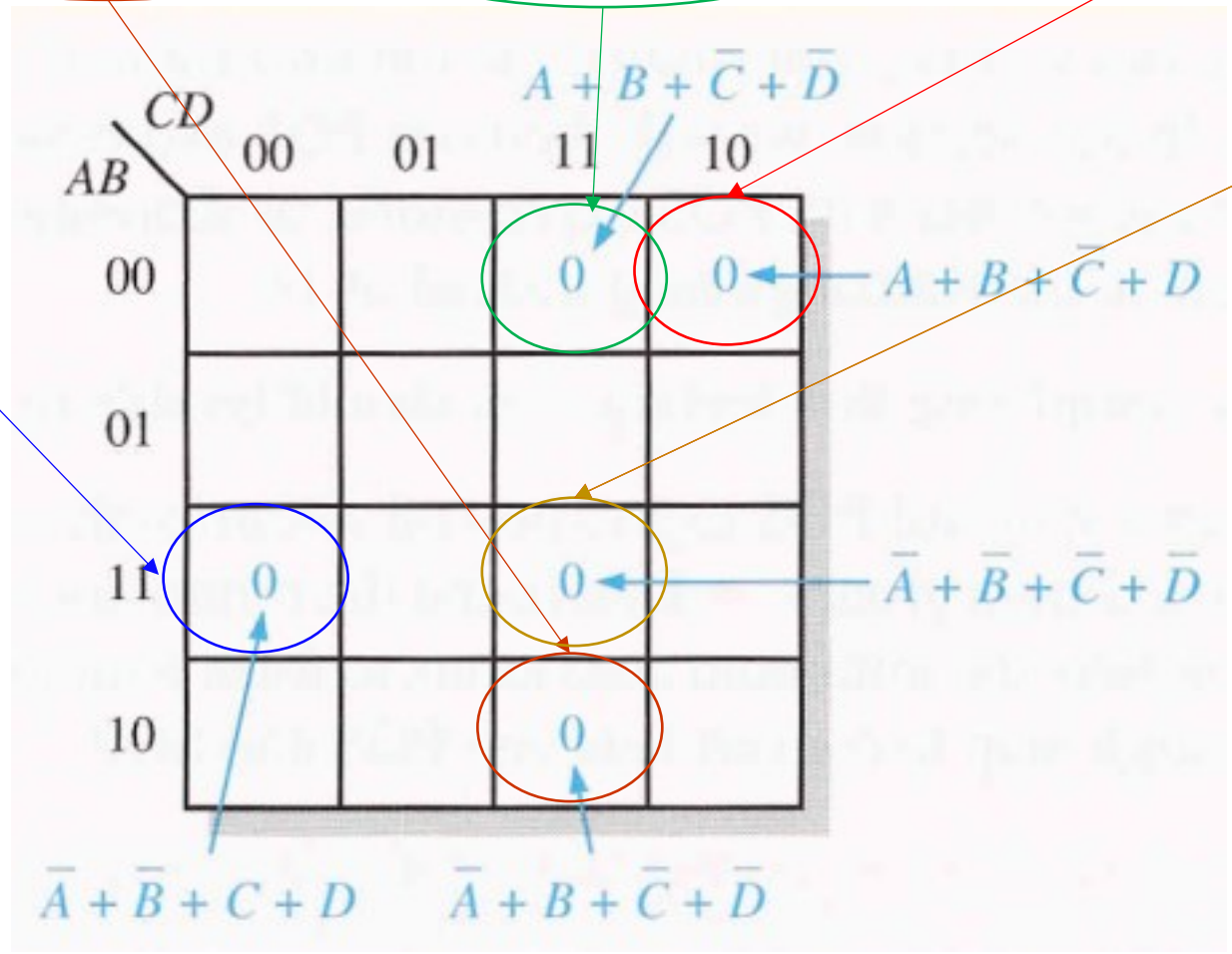
Inputs A B C D	Output Y
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	X
1 0 1 1	X
1 1 0 0	X
1 1 0 1	X
1 1 1 0	X
1 1 1 1	X

Don’t cares



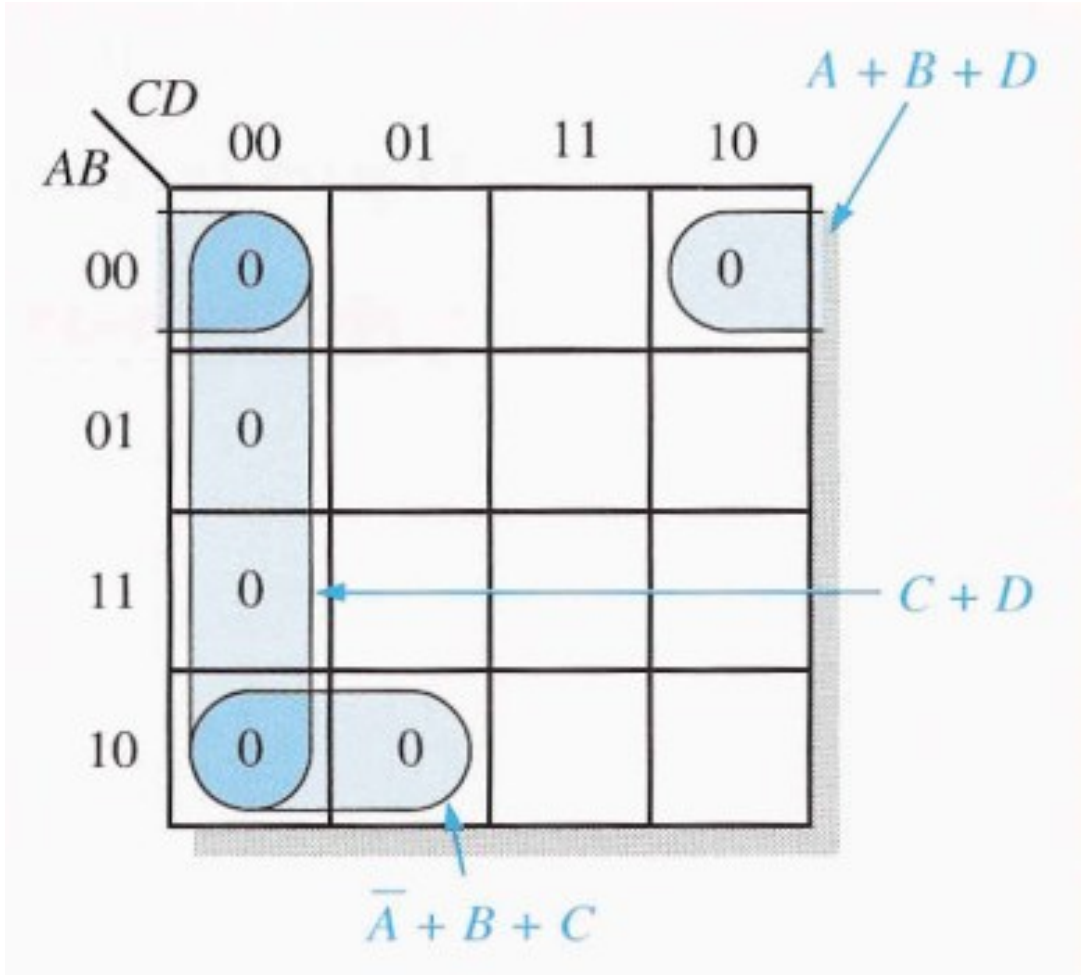
POS Mapping Using K-Map

$(\overline{A} + \overline{B} + C + D)(\overline{A} + B + \overline{C} + \overline{D})(A + B + \overline{C} + D)(\overline{A} + \overline{B} + \overline{C} + \overline{D})(A + B + \overline{C} + \overline{D})$



POS Simplification Using K-Map

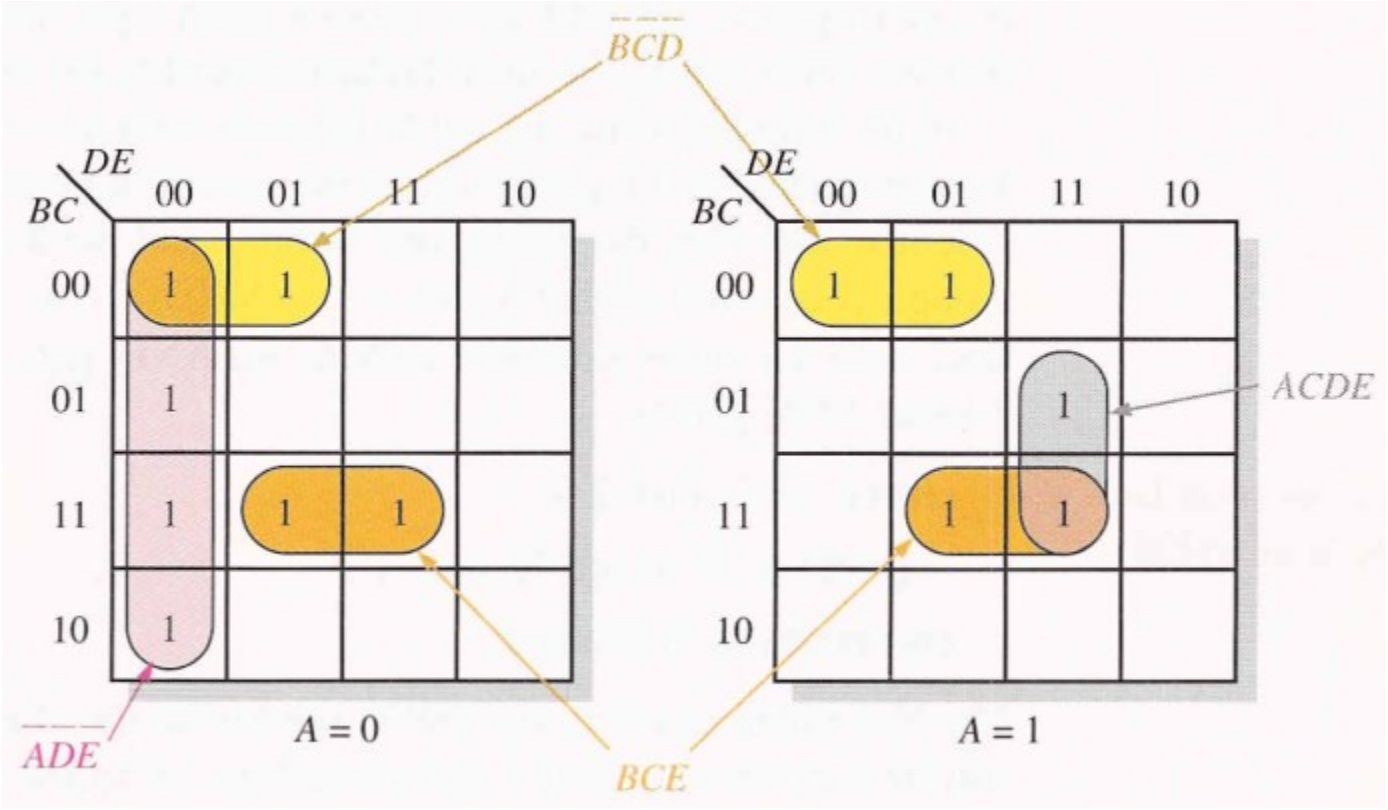
$$(B + C + D)(A + B + \overline{C} + D)(\overline{A} + B + C + \overline{D})(A + \overline{B} + C + D)(\overline{A} + \overline{B} + C + D)$$



5-Input K-Map

Use a Karnaugh map to minimize the following standard SOP 5-variable expression:

$$X = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}\,\overline{E} + \overline{A}\,\overline{B}\,C\,\overline{D}\,\overline{E} + \overline{A}\,B\,C\,\overline{D}\,\overline{E} + \overline{A}\,B\,\overline{C}\,\overline{D}\,\overline{E} + \overline{A}\,\overline{B}\,\overline{C}\,D\,\overline{E} + \overline{A}\,B\,C\,D\,\overline{E}$$
$$+ \overline{A}\,B\,C\,D\,E + A\,\overline{B}\,\overline{C}\,\overline{D}\,\overline{E} + A\,\overline{B}\,\overline{C}\,D\,\overline{E} + A\,B\,\overline{C}\,D\,\overline{E} + A\,B\,C\,D\,E + \overline{A}\,B\,C\,D\,E$$



Home Work

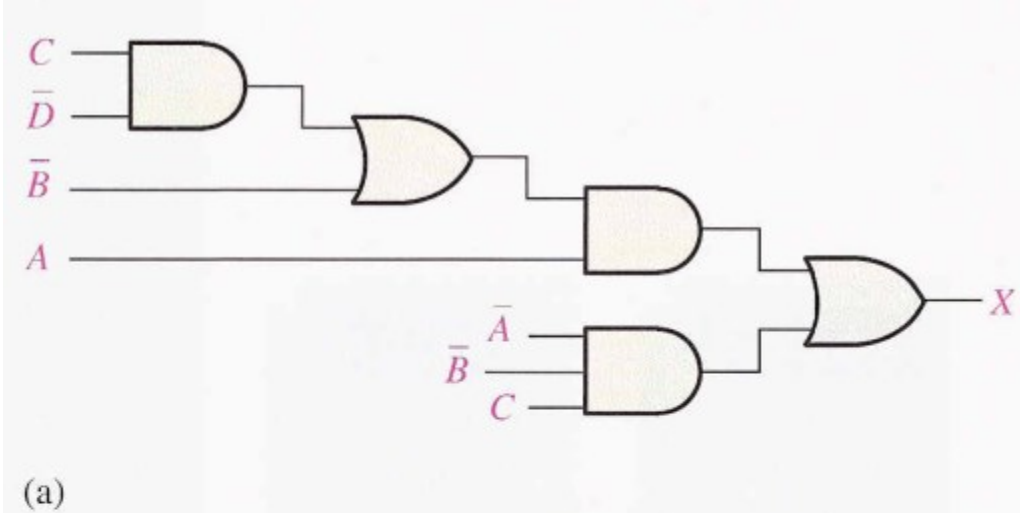
1- Apply DeMorgan's theorems to the following

$$\overline{(\overline{A + B})(\overline{C + D})(\overline{E + F})(\overline{G + H})}$$

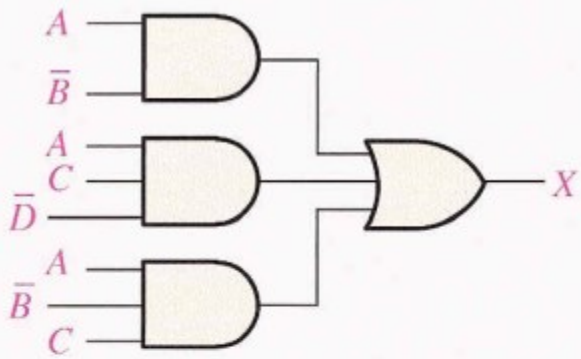
2 Determine the binary values of the variables for which the following standard POS expression is equal to 0:

$$(A + B + C + D)(A + \overline{B} + \overline{C} + D)(\overline{A} + \overline{B} + \overline{C} + \overline{D})$$

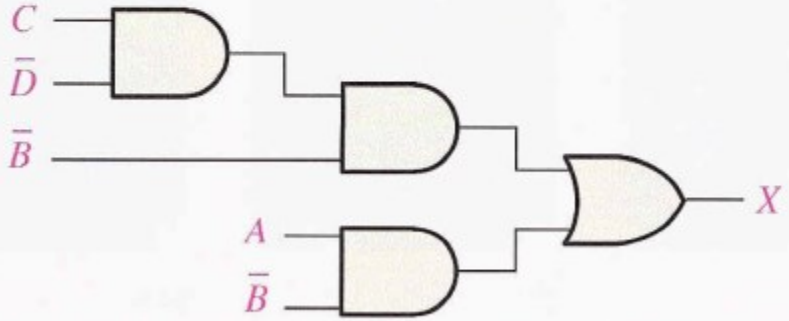
3- Which Circuits are Equivalent?



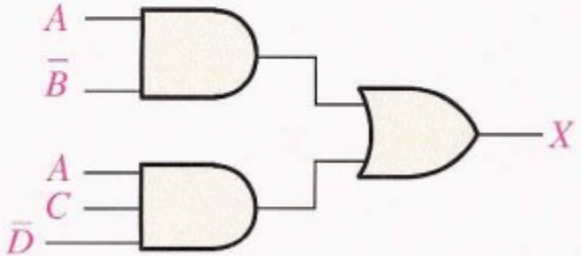
(a)



(b)



(c)



(d)

4 Develop a truth table for each of the SOP expressions:

(a) $\overline{A}B + AB\overline{C} + \overline{A}\,\overline{C} + A\overline{B}C$ **(b)** $\overline{X} + Y\overline{Z} + WZ + X\overline{Y}Z$

5 Use a Karnaugh map to reduce each expression to a minimum SOP form:

(b) $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + ABCD + ABC\overline{D}$

(c) $\overline{A}B(\overline{C}\,\overline{D} + \overline{C}D) + AB(\overline{C}\,\overline{D} + \overline{C}D) + A\overline{B}\,\overline{C}D$

6 Minimize the following SOP expression using a Karnaugh map:

$$X = \overline{A}B\overline{C}D\overline{E} + \overline{A}\,\overline{B}\,\overline{C}DE + A\overline{B}\,\overline{C}DE + AB\overline{C}\,\overline{D}\,\overline{E} + \overline{A}BCD\overline{E} + \overline{A}BC\overline{D}E$$
$$+ \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}\,\overline{E} + \overline{A}\,\overline{B}CDE + AB\overline{C}D\overline{E} + AB\overline{C}DE$$

University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

Lec. Dr. Taif Alawsi

Lec. 5: Combinational Logic Analysis : 2024-Oct-27

Lecture Outline

1. Circuit Design
2. Circuit Minimization
3. Circuit from Expression
4. Circuit from Truth Table
5. Universal NAND – Universal NOR
6. Multiple Input Timing Diagram
7. HW

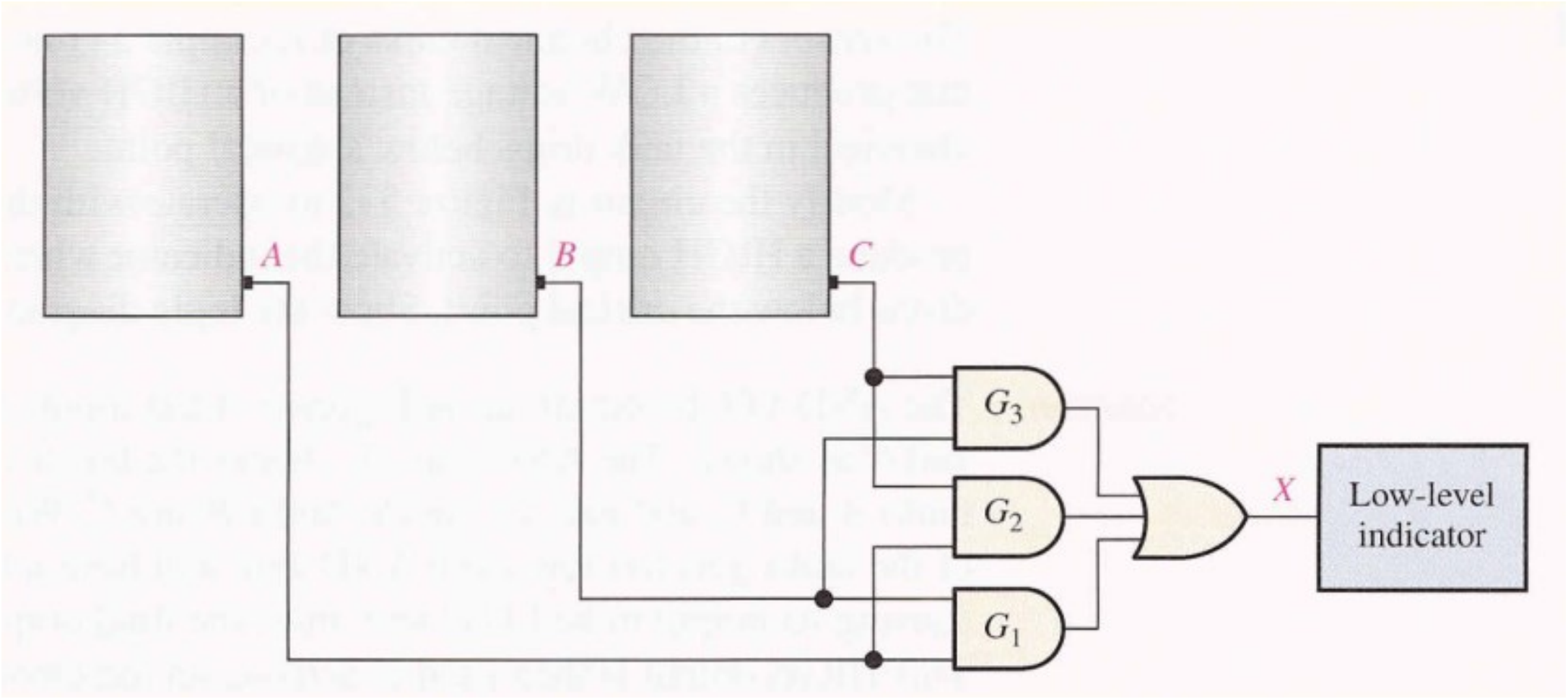
Design Problems

Ex-1

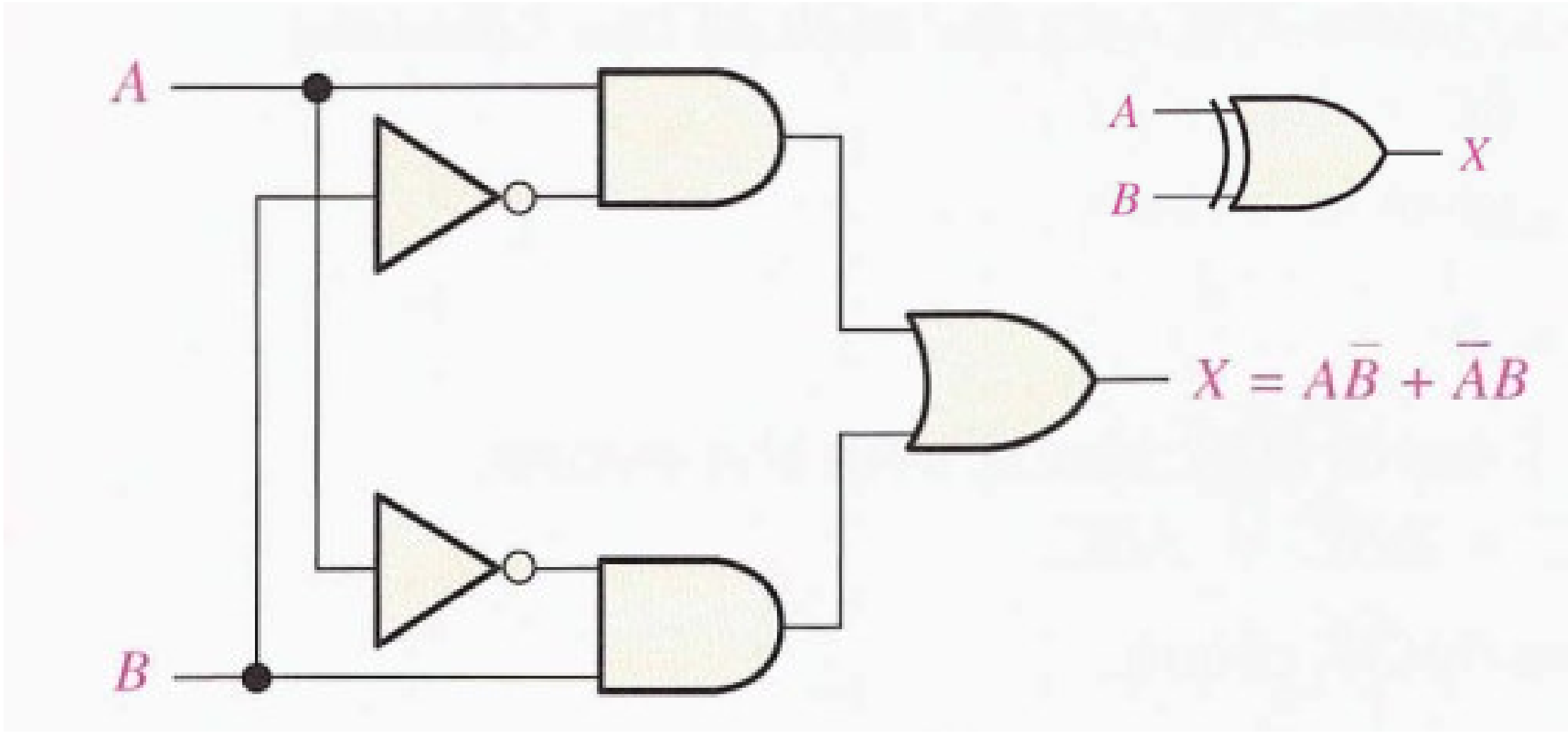
In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

Sol



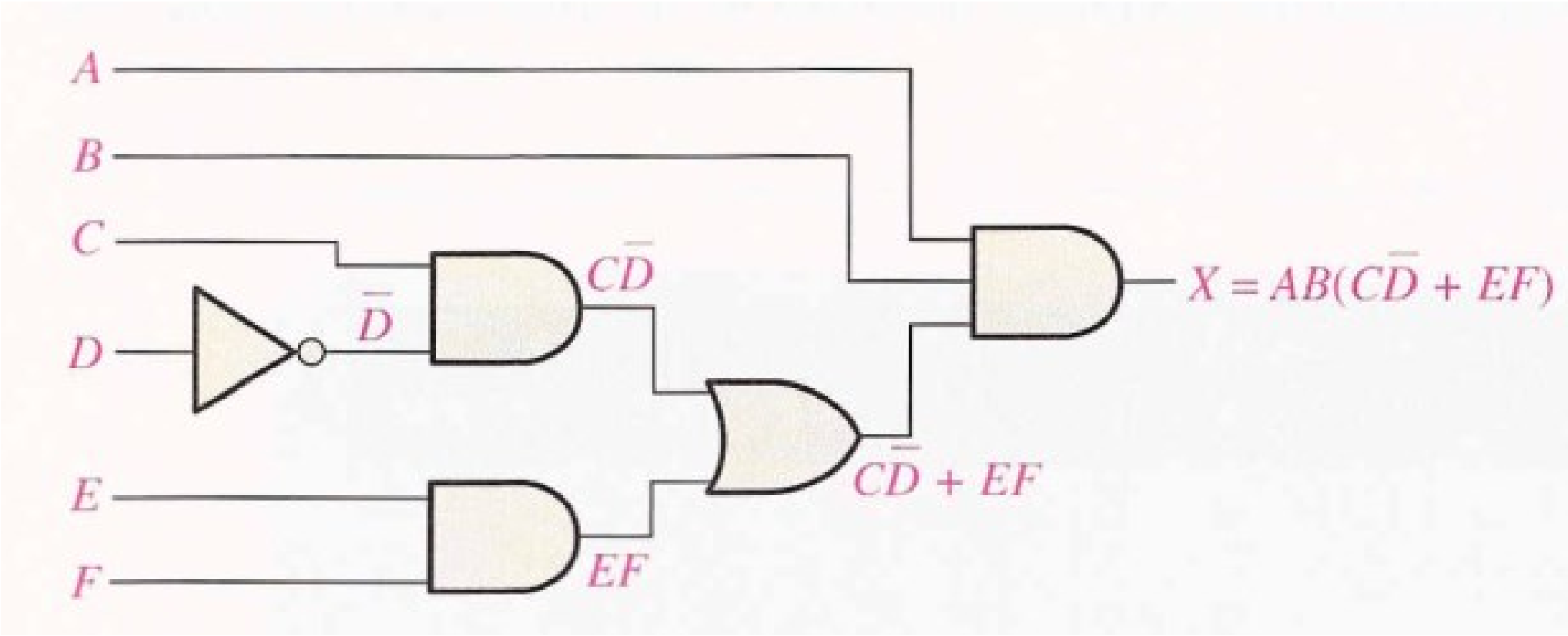
XOR Gate Equivalent Circuit



Circuit from Expression

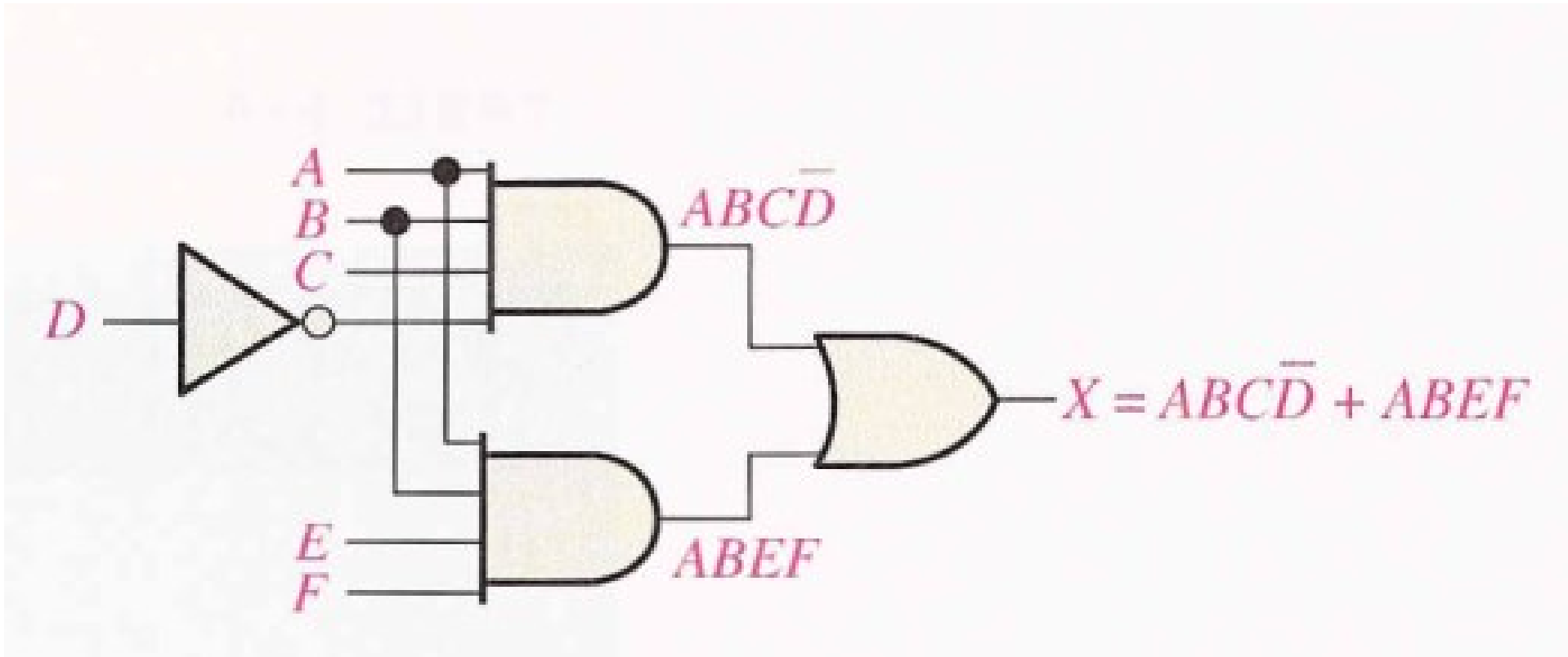
Ex-2

$$AB(C\overline{D} + EF)$$



Ex-3

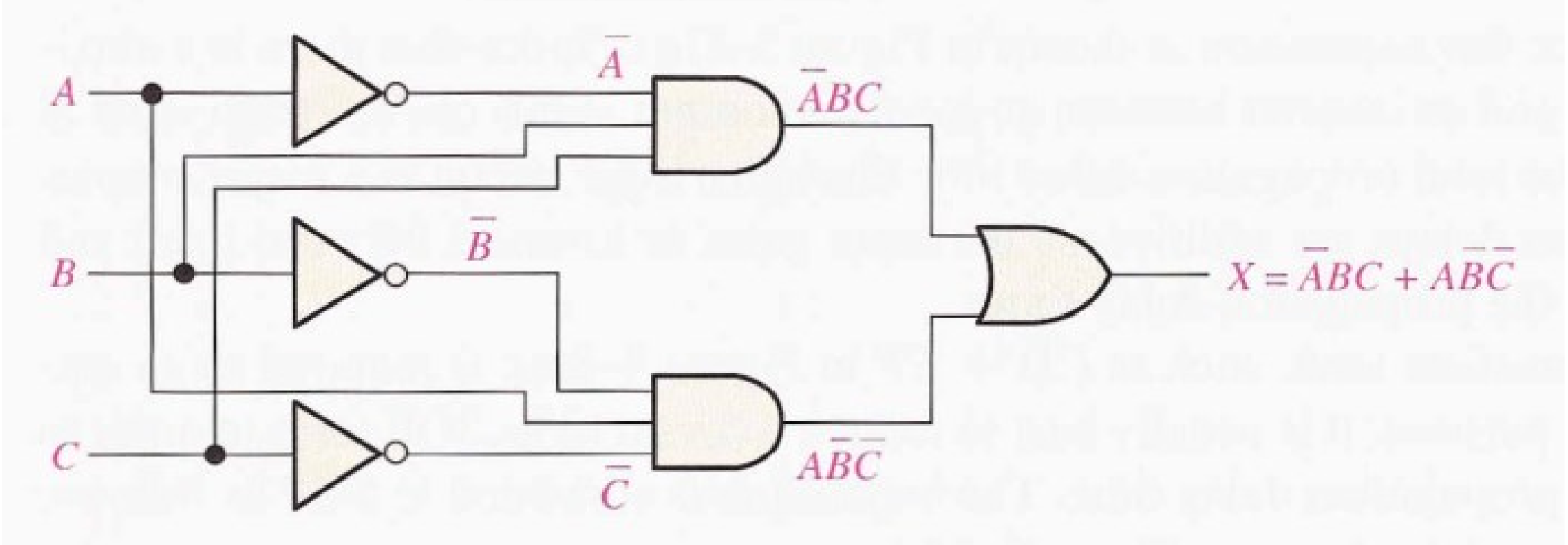
$$ABC\overline{D} + ABEF$$



Circuit from Expression

Ex-4

$X = \bar{A}BC + A\bar{B}\bar{C}$

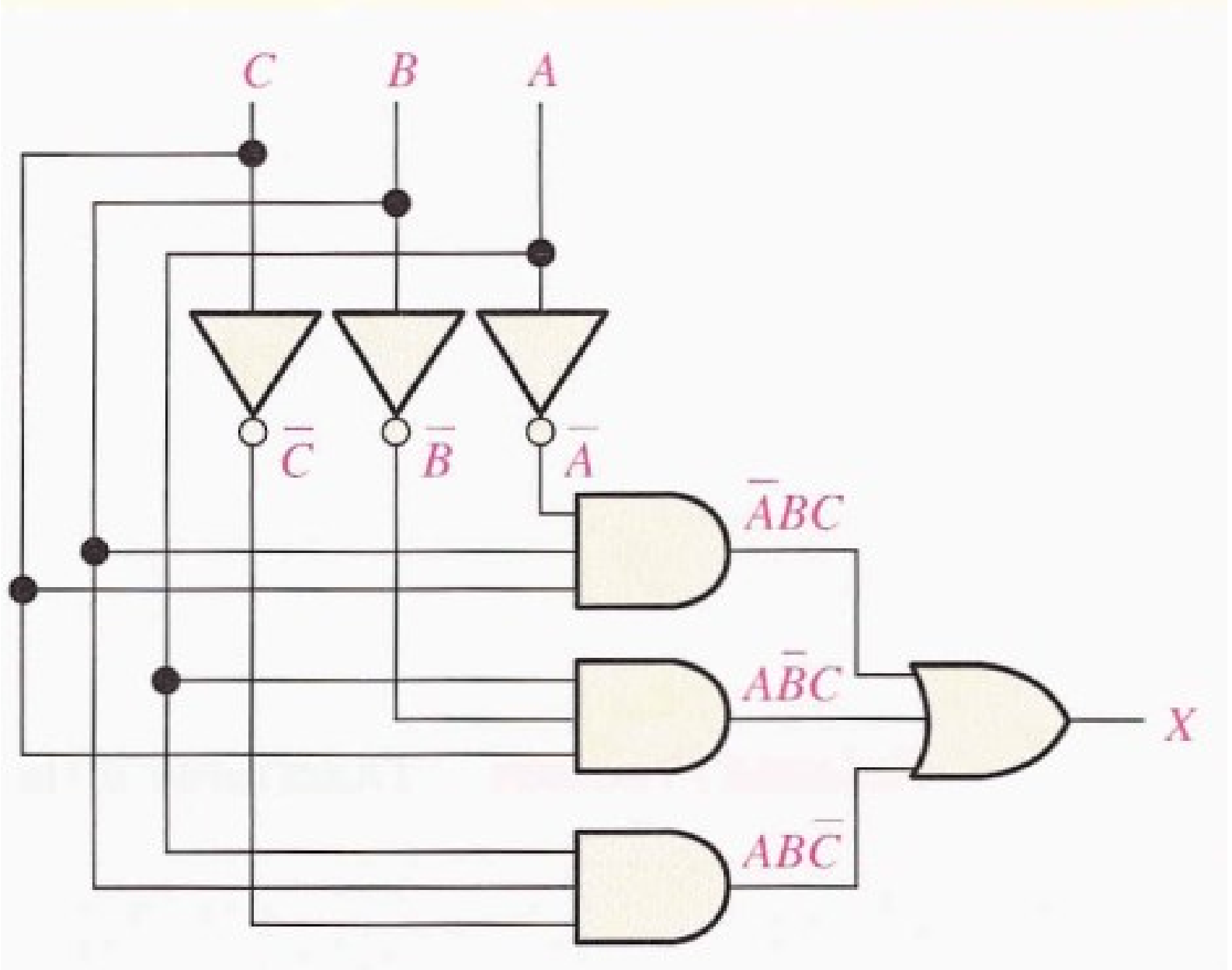


Circuit from Truth Table

Ex-5

INPUTS A	B	C	OUTPUT X	PRODUCT TERM
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC$
1	0	0	0	
1	0	1	1	$A\overline{B}C$
1	1	0	1	$AB\overline{C}$
1	1	1	0	

$$X = \overline{A}BC + A\overline{B}C + AB\overline{C}$$

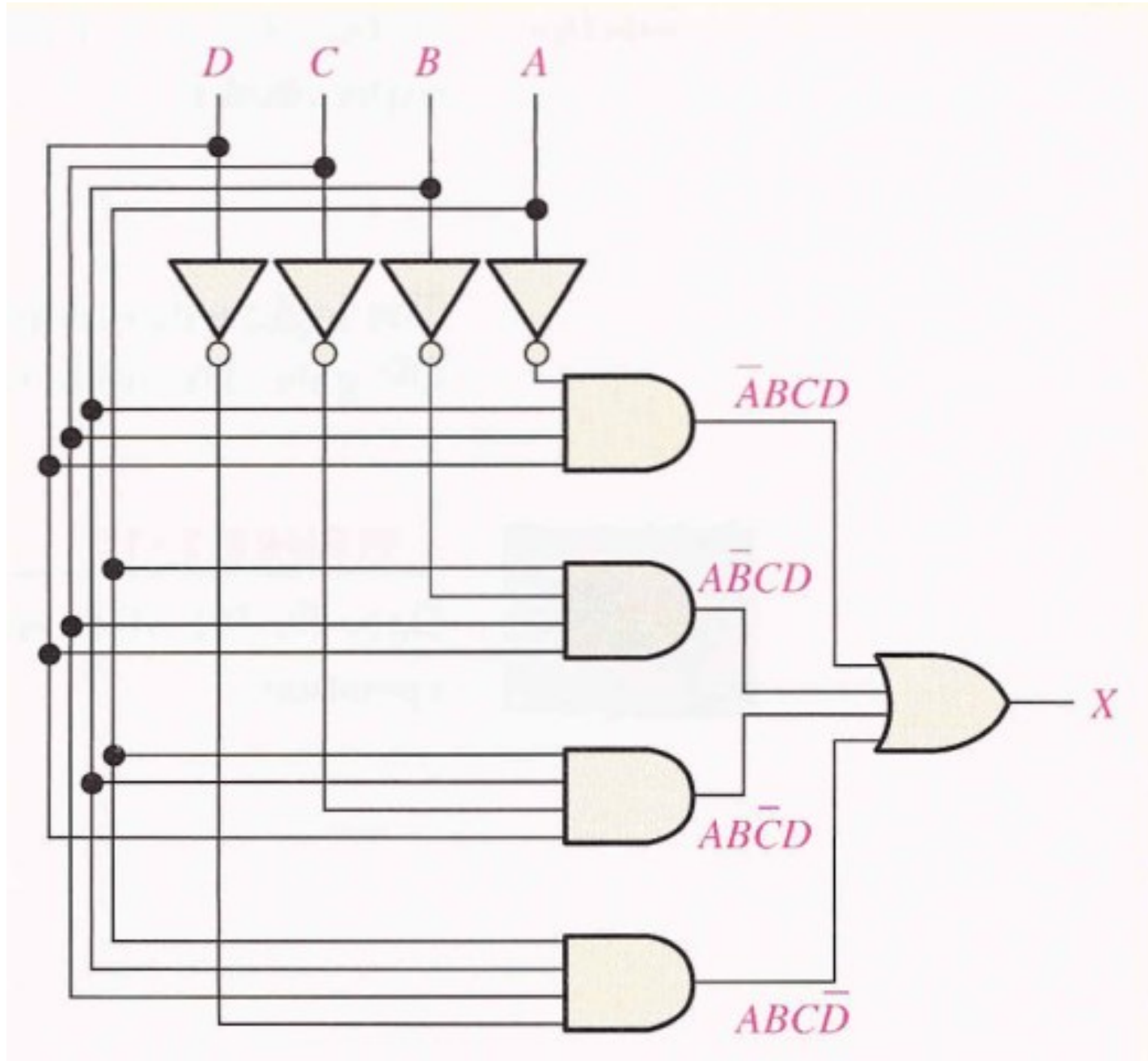

C
B
A
$\overline{C}$
$\overline{B}$
$\overline{A}$
$\overline{A}BC$
$A\overline{B}C$
$AB\overline{C}$
X

Ex-6

Develop a logic circuit with four input variables that will only produce a 1 output when exactly three input variables are 1s.

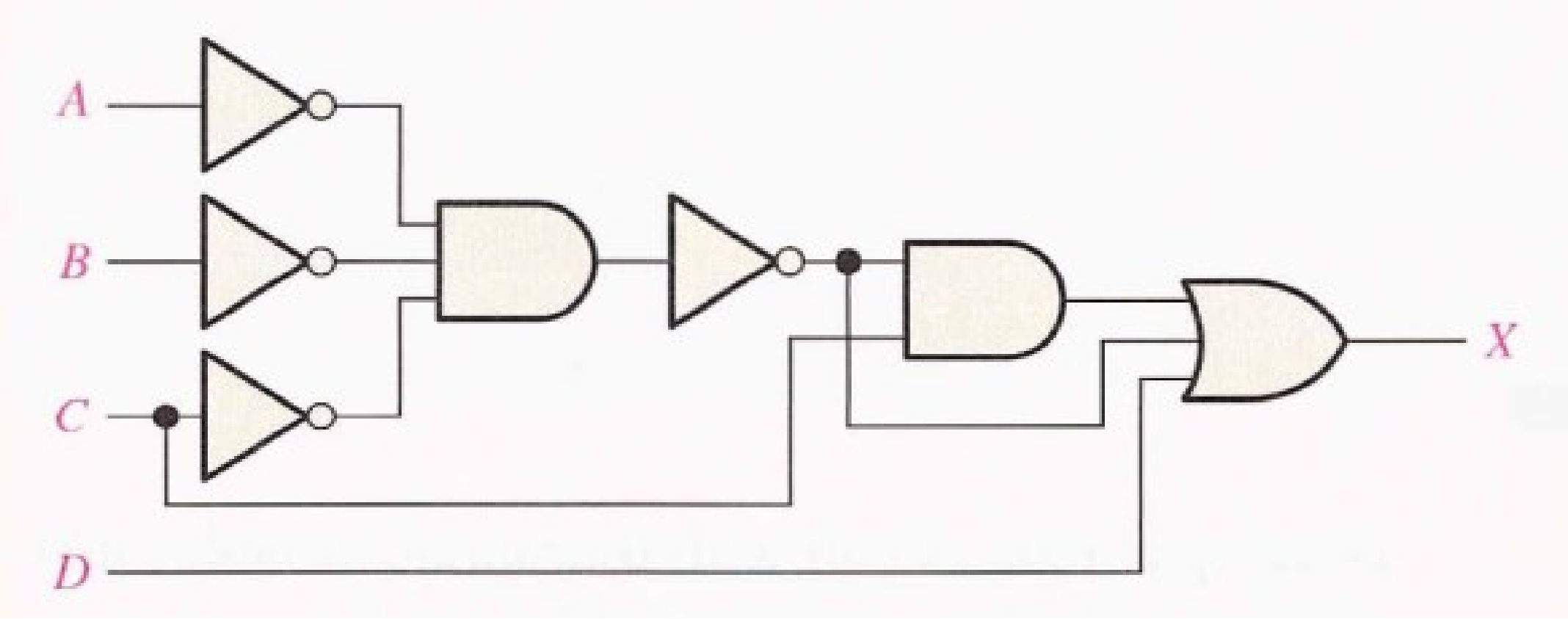
A	B	C	D	PRODUCT TERM
0	1	1	1	$\overline{A}BCD$
1	0	1	1	$A\overline{B}CD$
1	1	0	1	$AB\overline{C}D$
1	1	1	0	$ABC\overline{D}$

$$X = \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D}$$


D
C
B
A
$\bar{A}BCD$
$A\bar{B}CD$
$AB\bar{C}D$
$ABC\bar{D}$
X

Circuit Minimization

Ex-7



The expression for the output of the circuit is

$$X = (\overline{\overline{A}\,\overline{B}\,\overline{C}})C + \overline{\overline{A}\,\overline{B}\,\overline{C}} + D$$

Applying DeMorgan's theorem and Boolean algebra,

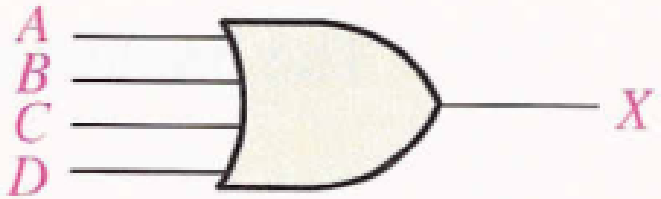
$$X = (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})C + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + D$$

$$= AC + BC + CC + A + B + C + D$$

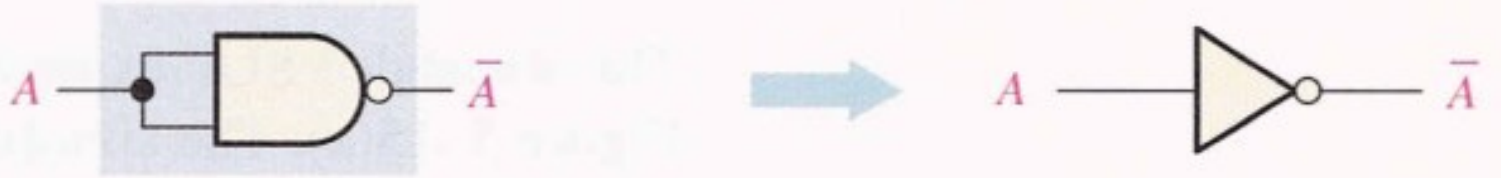
$$= AC + BC + C + A + B + \not{C} + D$$

$$= C(A + B + 1) + A + B + D$$

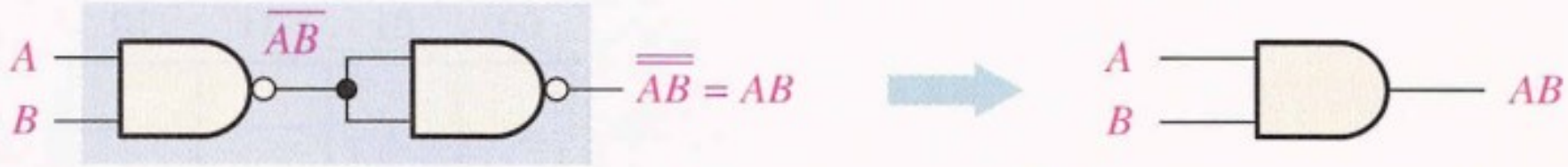
$$X = A + B + C + D$$



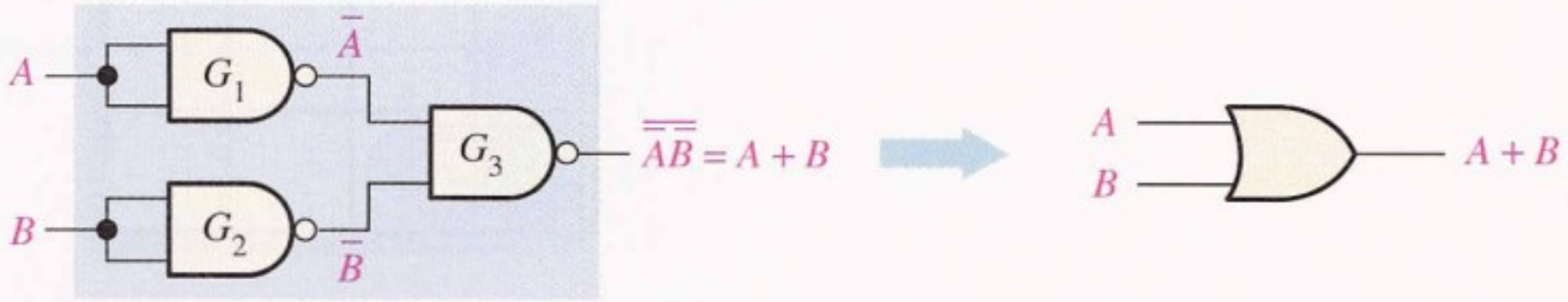
Universal NAND



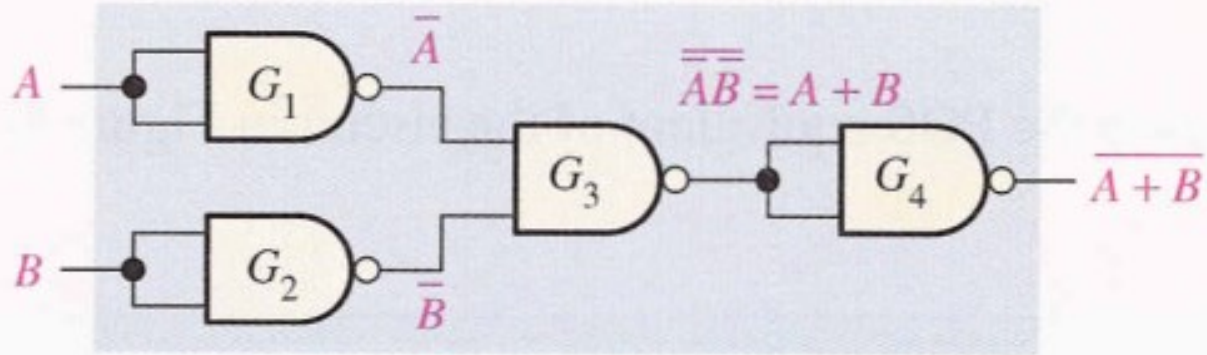
(a) One NAND gate used as an inverter

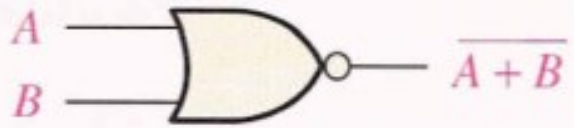


(b) Two NAND gates used as an AND gate



(c) Three NAND gates used as an OR gate

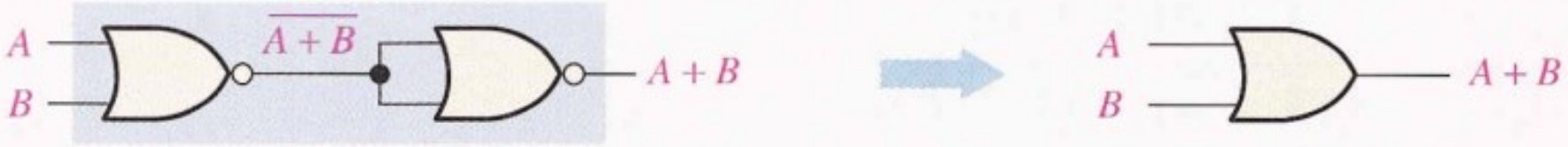




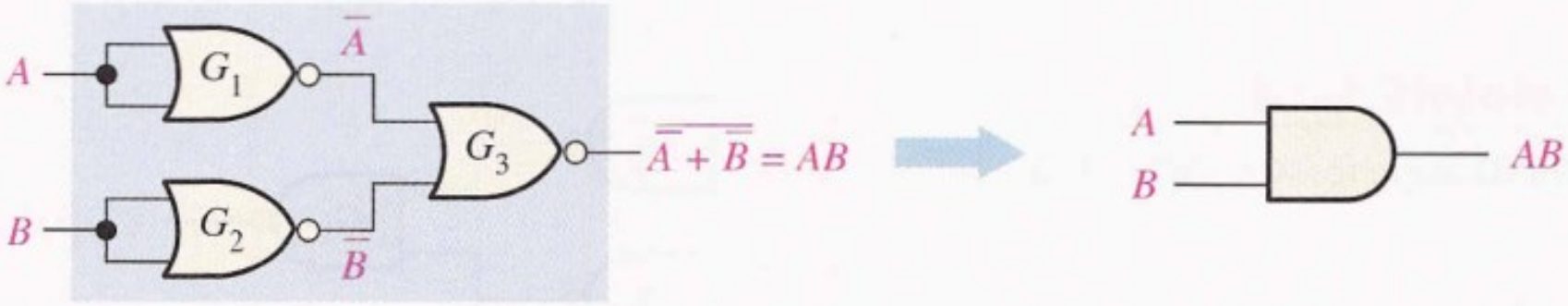
(d) Four NAND gates used as a NOR gate

Universal NOR

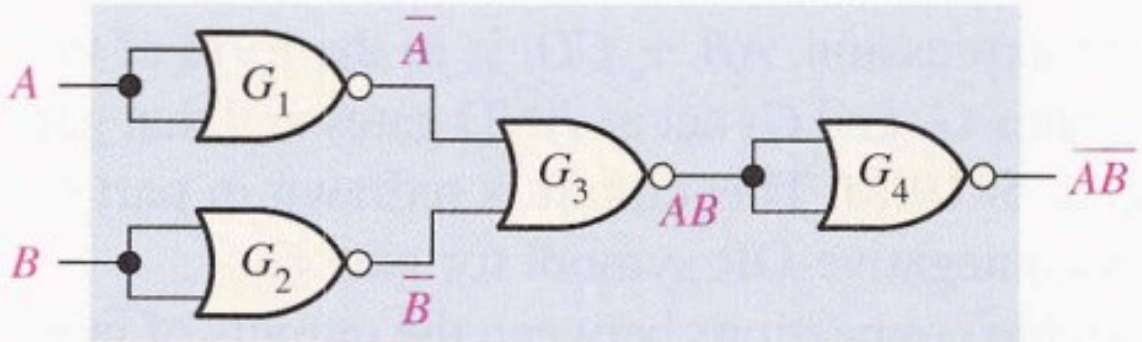
(a) One NOR gate used as an inverter

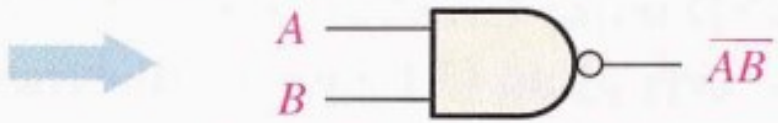


(b) Two NOR gates used as an OR gate



(c) Three NOR gates used as an AND gate

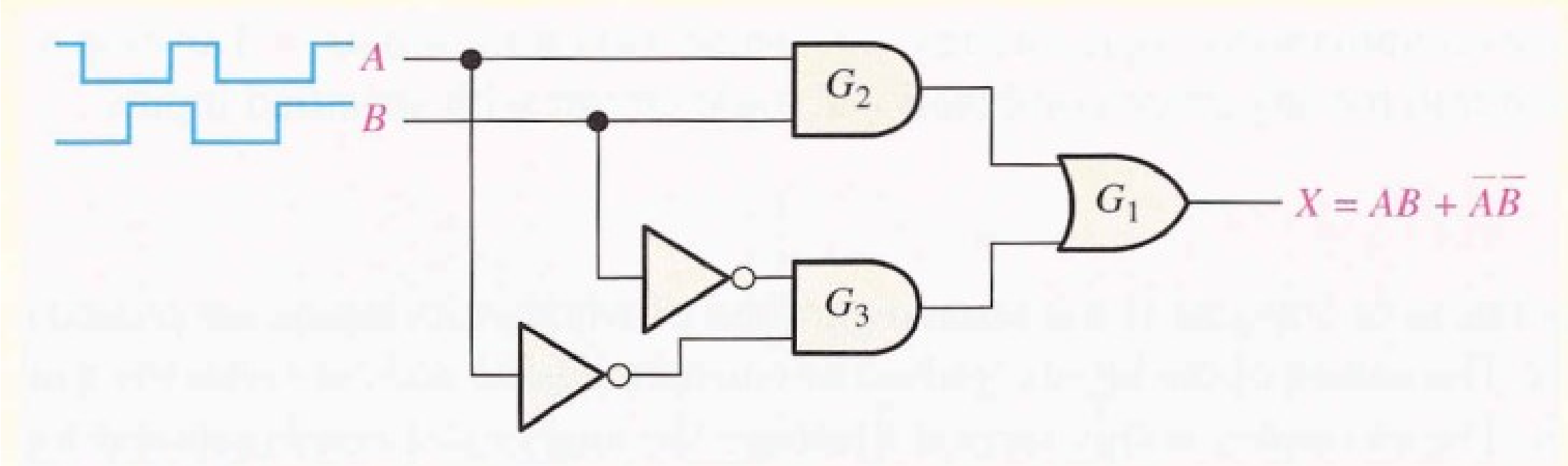


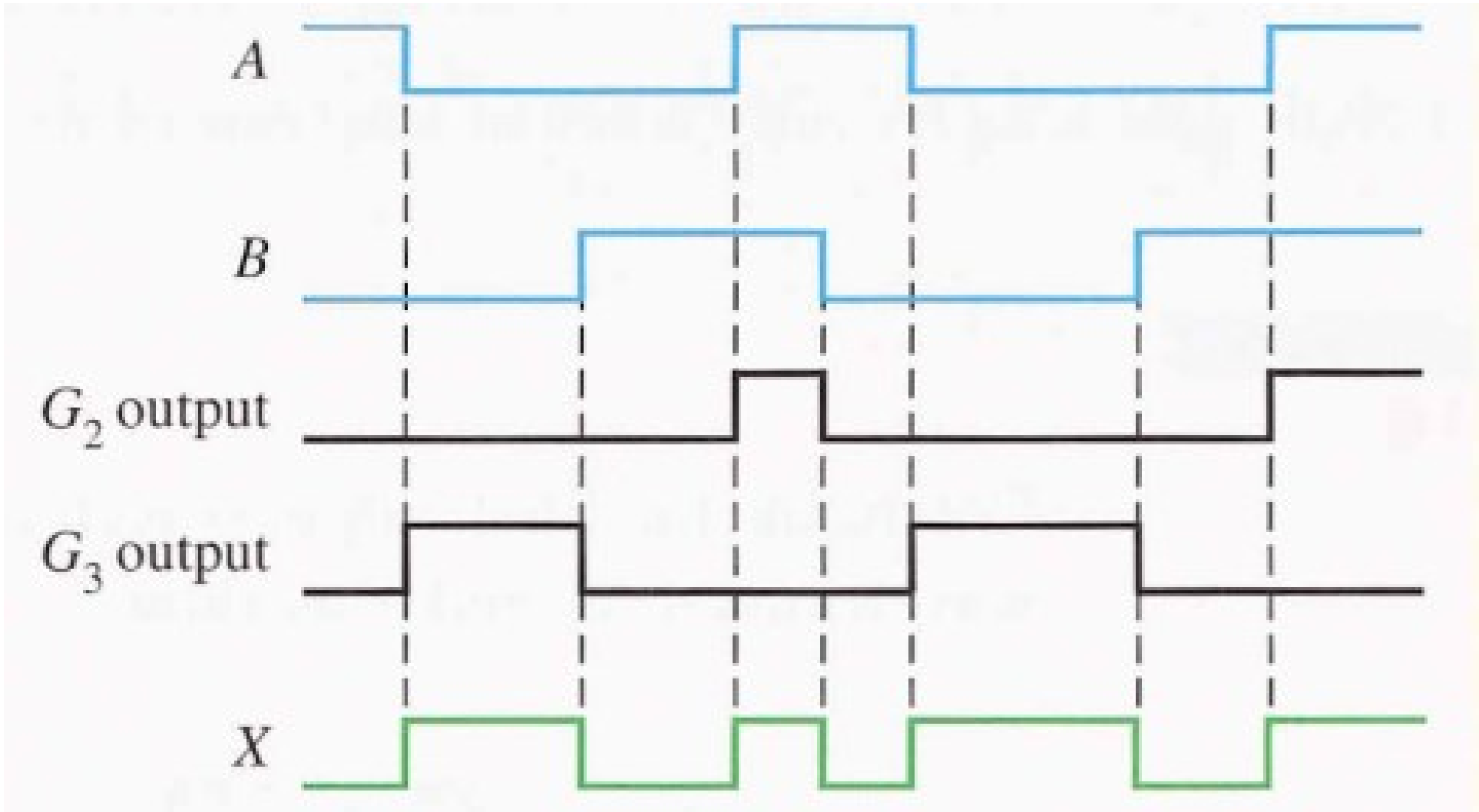


(d) Four NOR gates used as a NAND gate

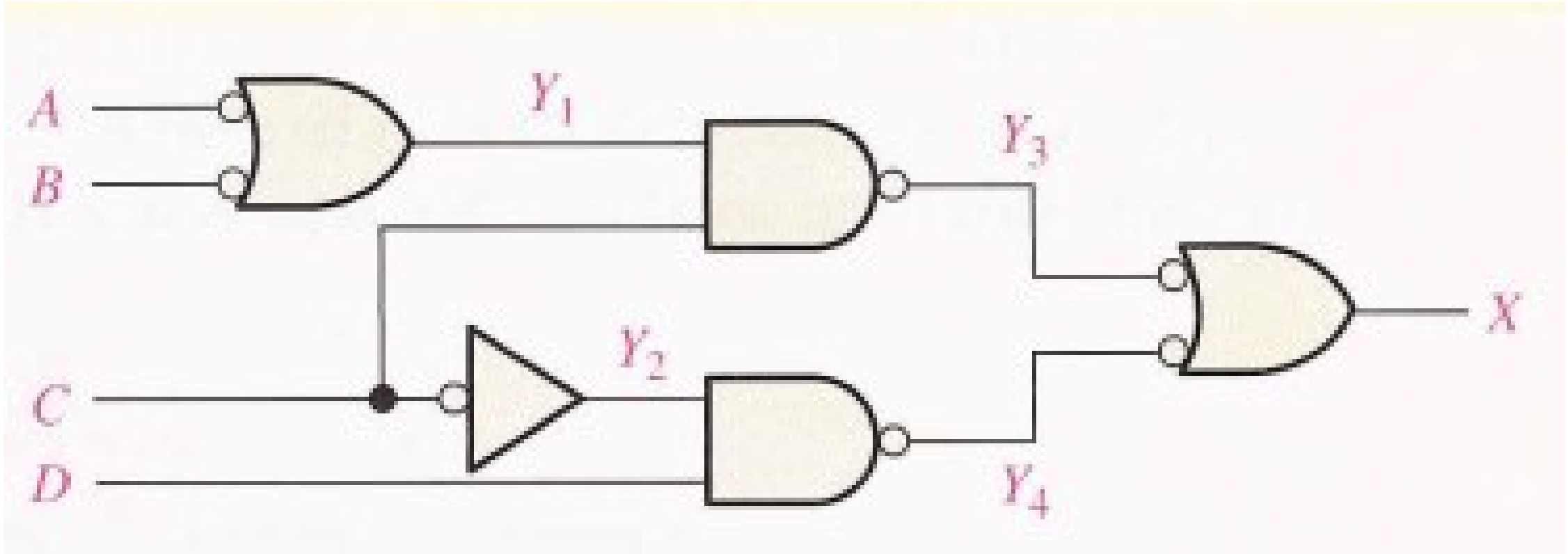
Timing Diagram for Combinational Circuit

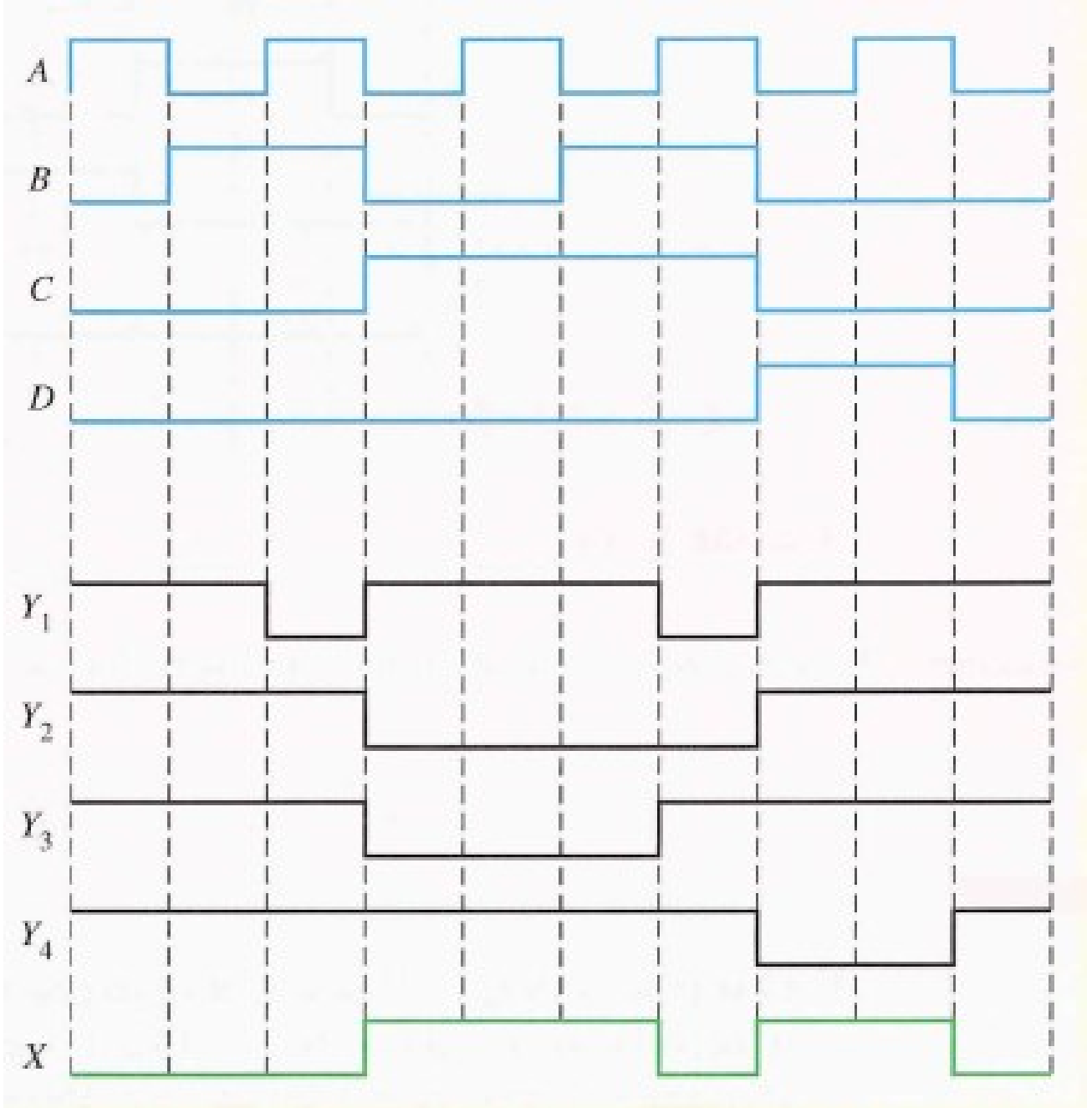
Ex-8




A
B
G_2 output
G_3 output
X

Ex-9

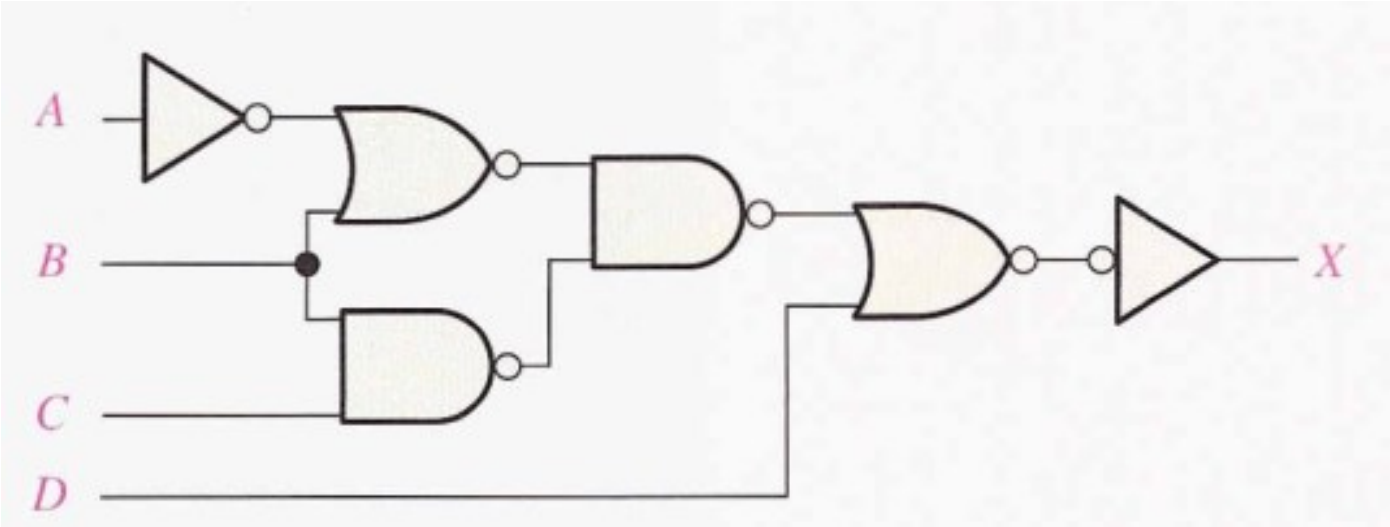



A
B
C
D
Y_1
Y_2
Y_3
Y_4
X

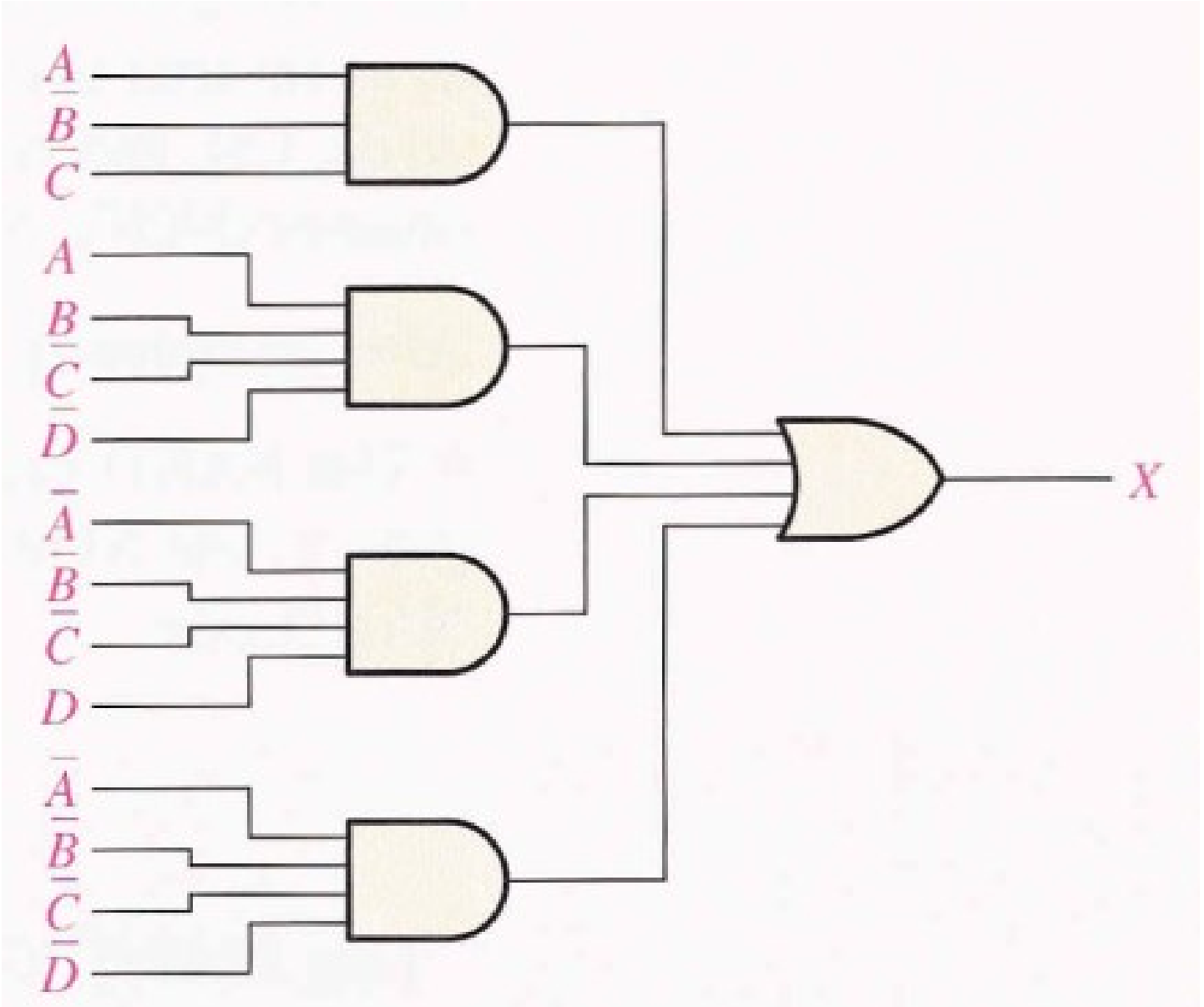
Home Work

1 Implement the expression $X = \overline{(\overline{A} + \overline{B} + \overline{C})DE}$ by using NAND logic.

2- Develop a truth Table for the circuit



3- Minimize the circuit



University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

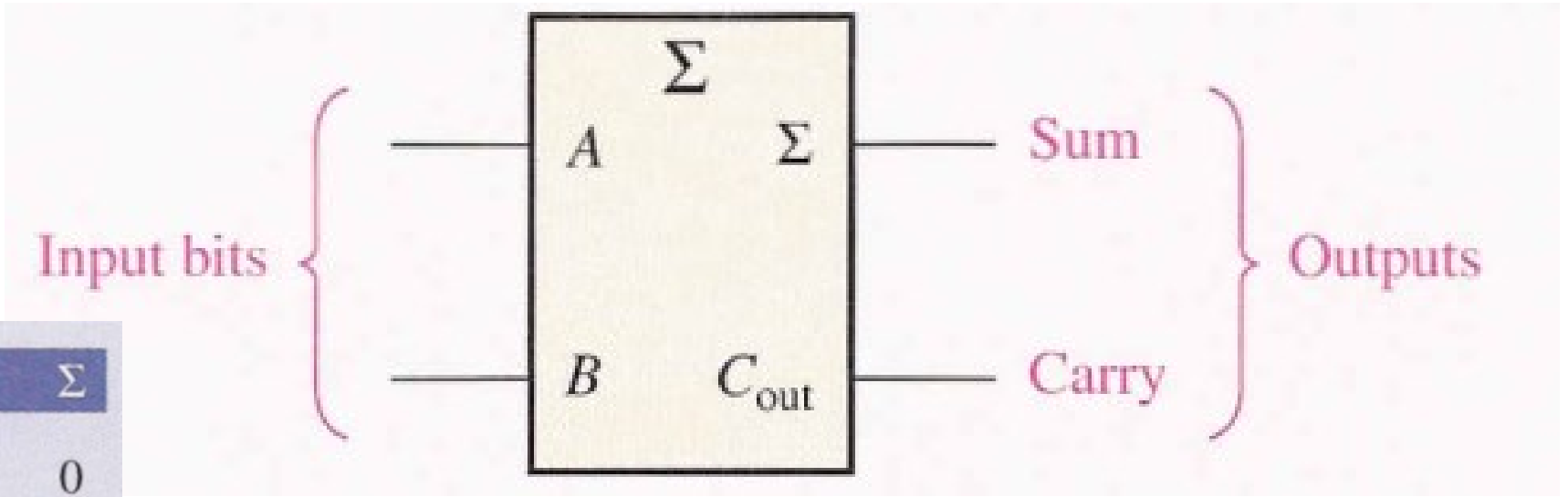
Lec. Dr. Taif Alawsi

Lec. 6: Functions of Combinational Logic 1: 2024-Nov-03

Lecture Outline

1. Half Adders
2. Full Adders
3. Parallel Binary Adders
4. Cascading
5. Comparators
6. IC Sets
7. HW

Half Adder



A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Σ = sum

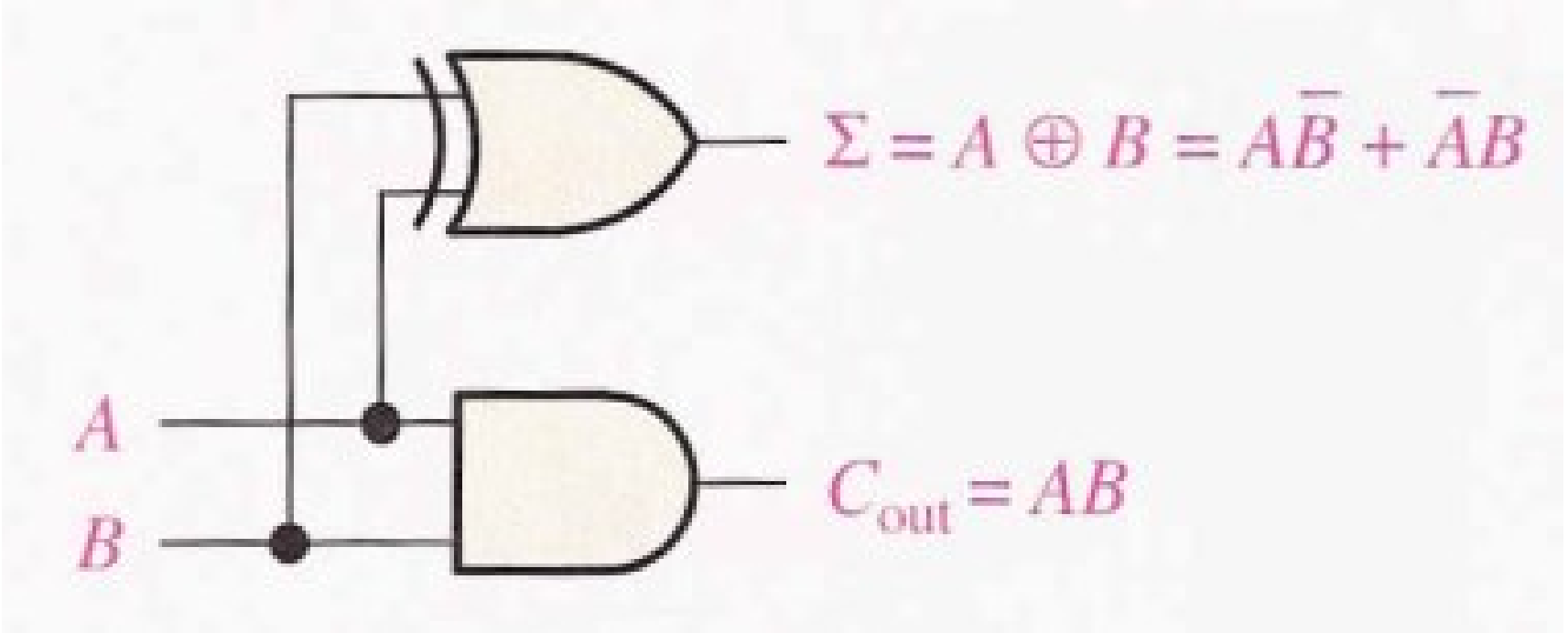
C_{out} = output carry

A and B = input variables (operands)

$$\Sigma = A \oplus B$$

$$C_{out} = AB$$

Half Adder Equivalent Circuit



Full Adder

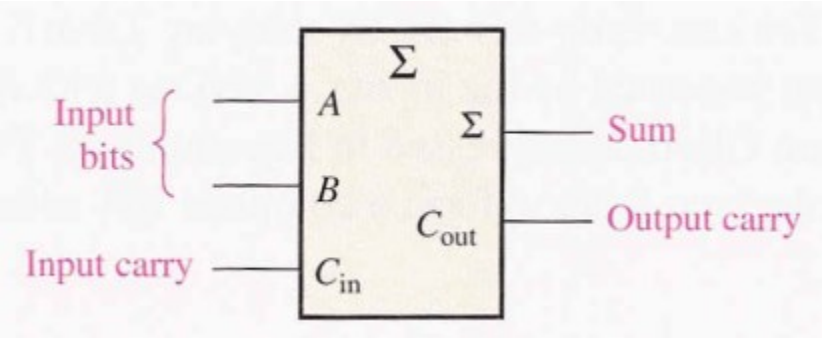
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{in} = input carry, sometimes designated as *CI*

C_{out} = output carry, sometimes designated as *CO*

Σ = sum

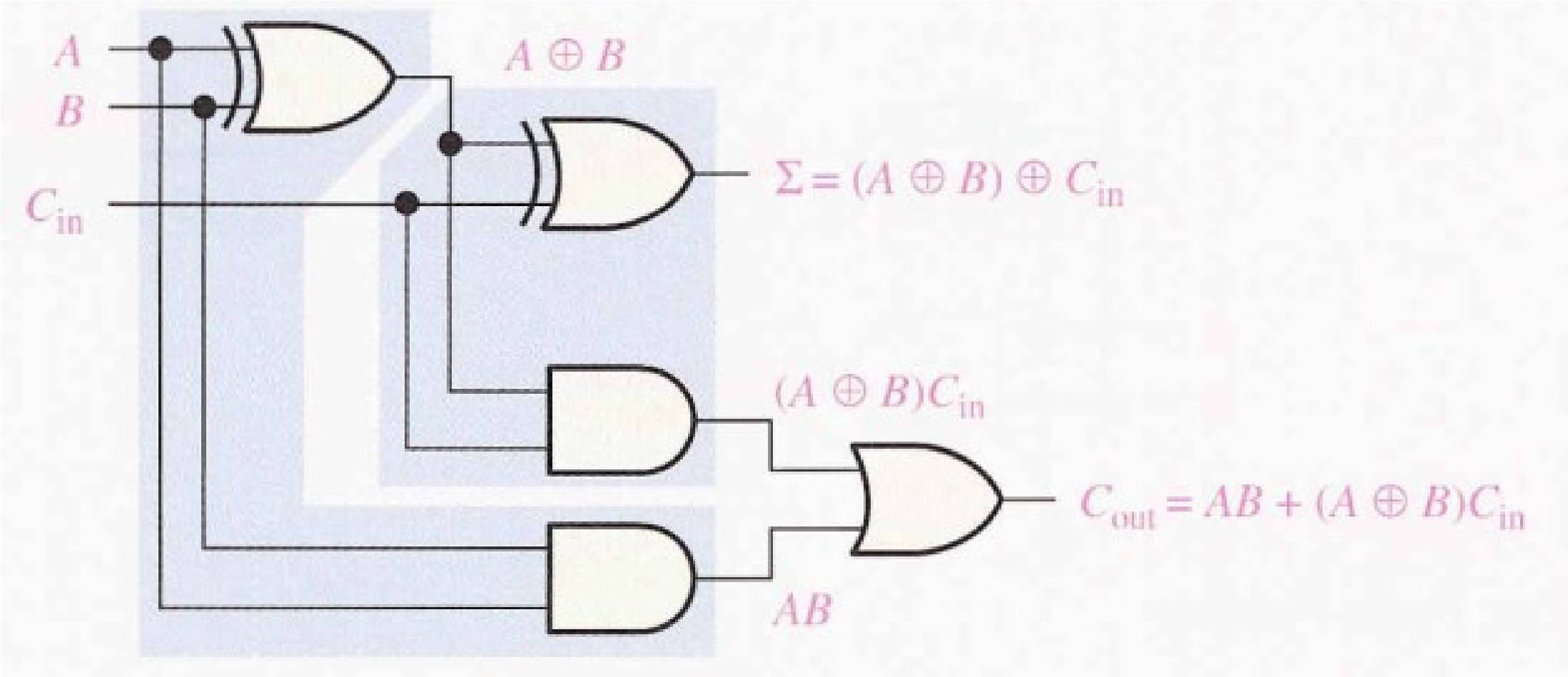
A and B = input variables (operands)



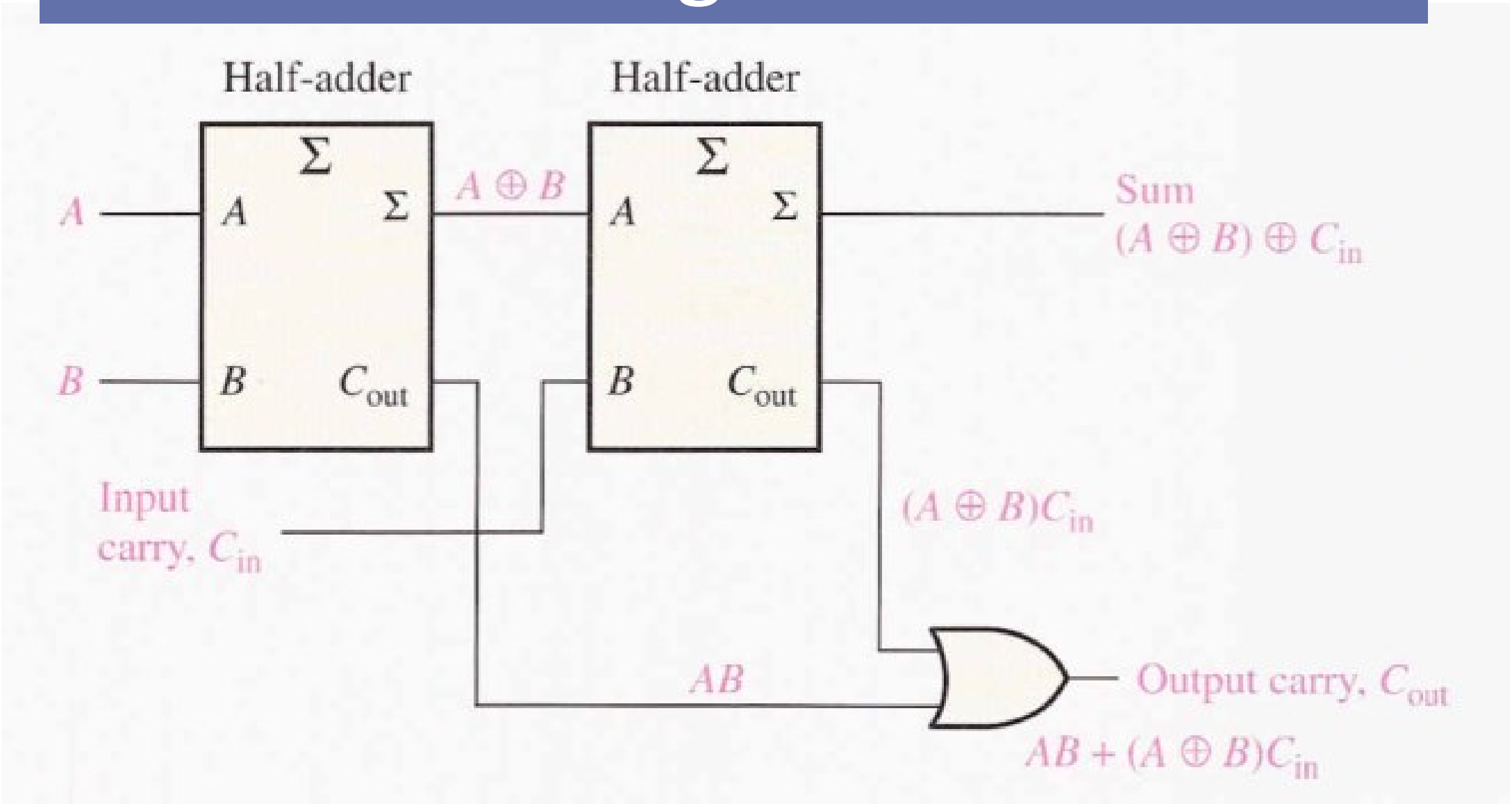
$$C_{out} = AB + (A \oplus B)C_{in}$$

$$\Sigma = (A \oplus B) \oplus C_{in}$$

Full Adder Equivalent Circuit

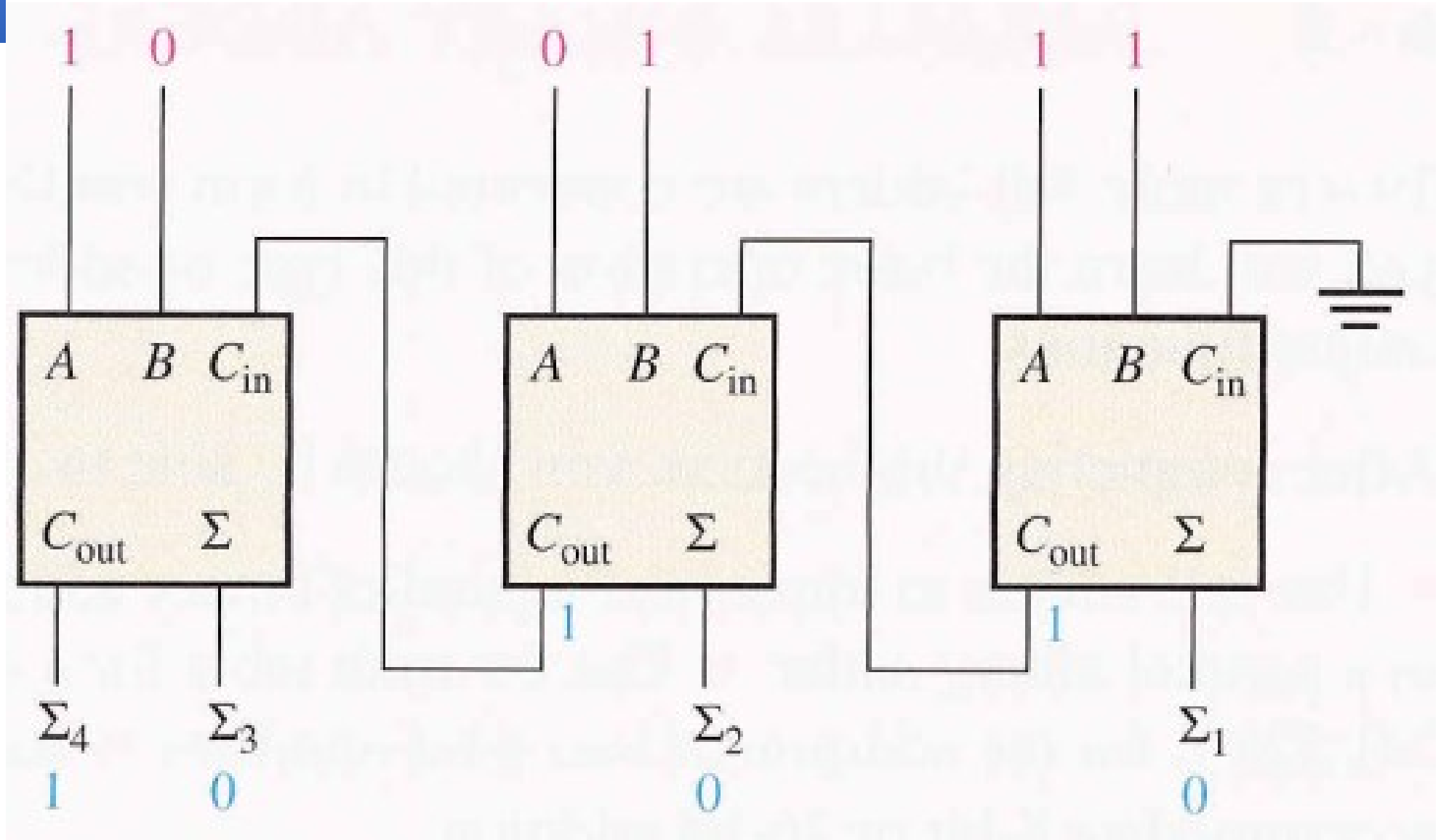


Full Adder Using Two Half Adders

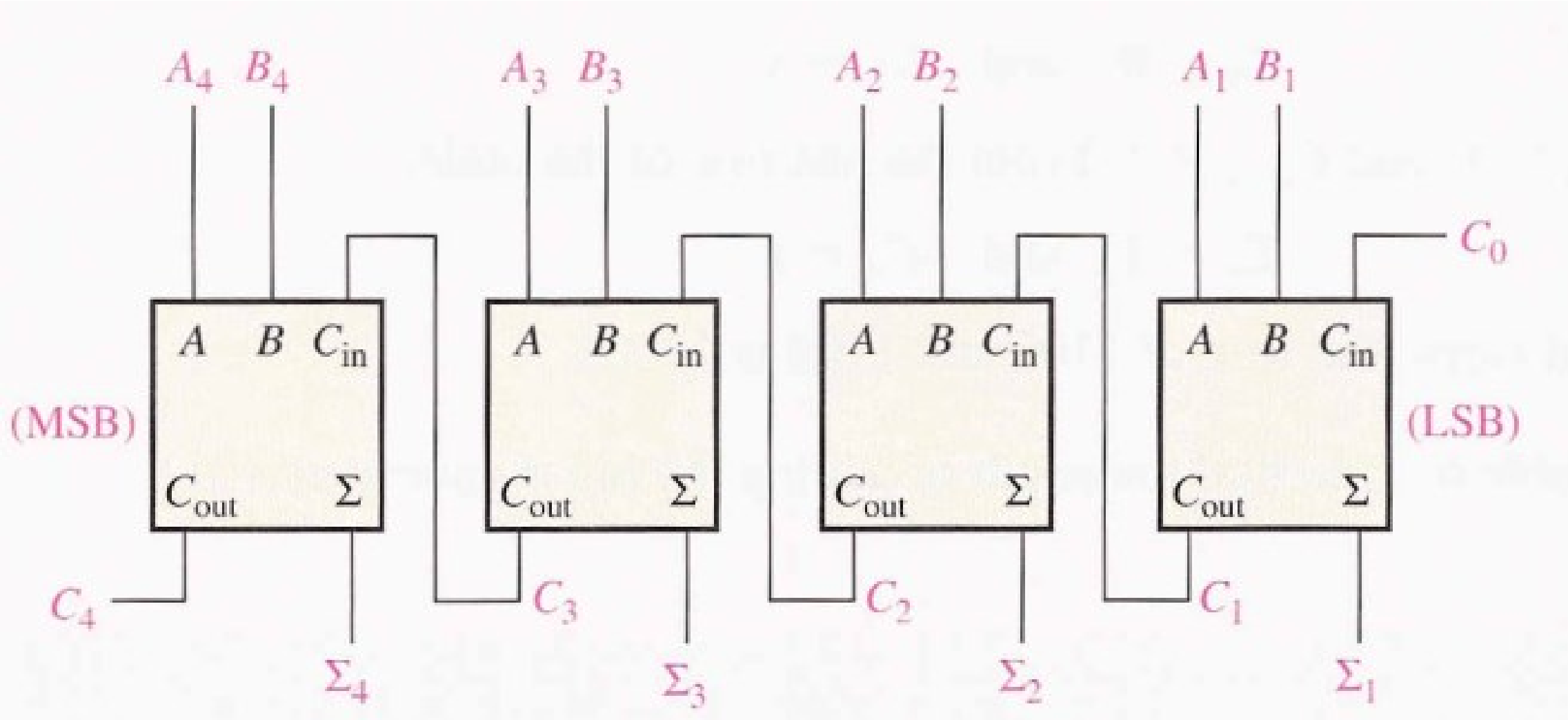


Ex-1

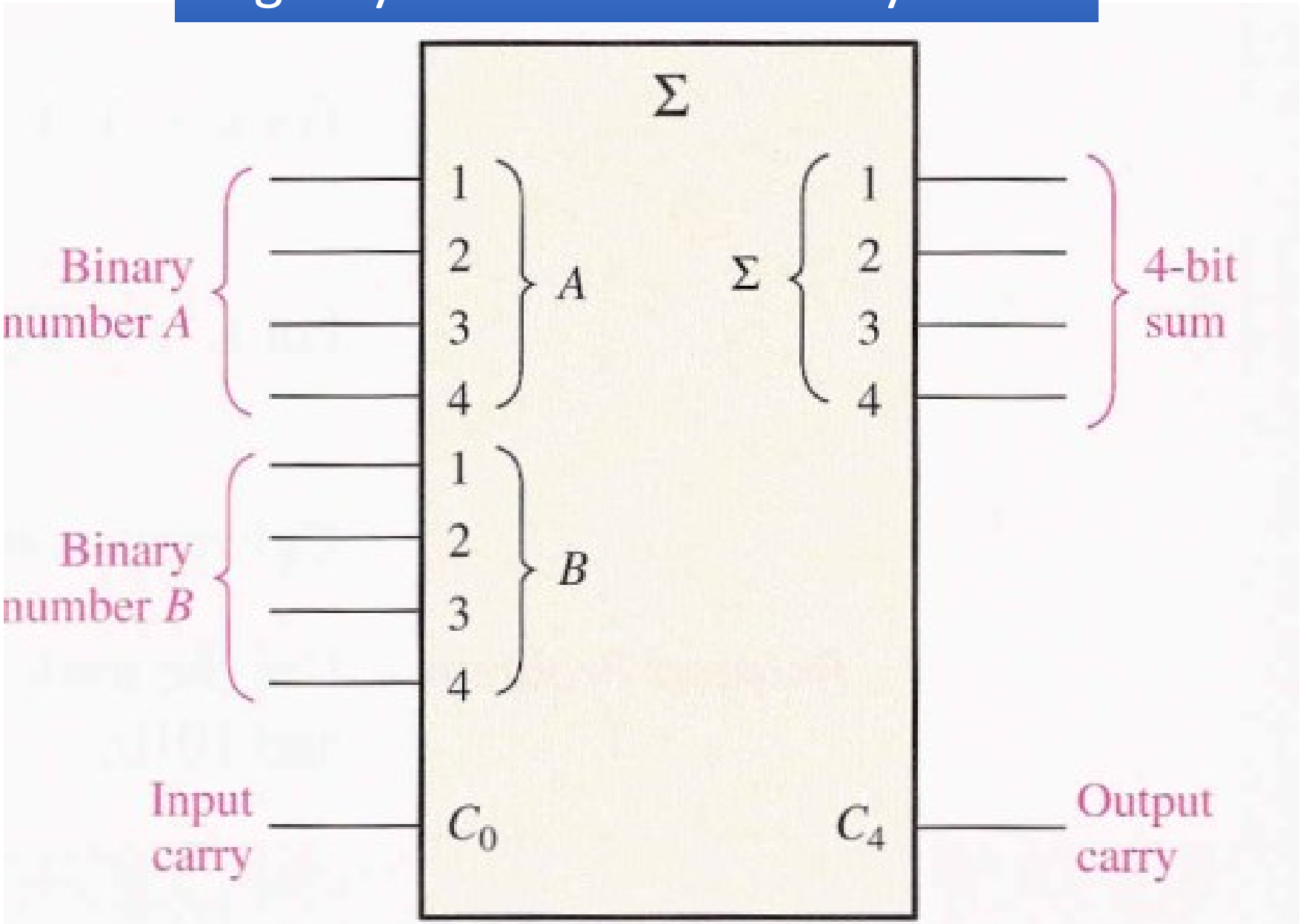
101+011=1000



4-bit Binary Adder



Logic Symbol of 4-Bit Binary Adder



T-T for 4-Bit Binary Adder

C_{n-1}	A_n	B_n	Σ_n	C_n
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

Ex-2

Use the 4-bit parallel adder truth table to find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

$$A_4A_3A_2A_1 = 1100 \quad \text{and} \quad B_4B_3B_2B_1 = 1100$$

For $n = 1$: $A_1 = 0$, $B_1 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_1 = \mathbf{0} \quad \text{and} \quad C_1 = 0$$

For $n = 2$: $A_2 = 0$, $B_2 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_2 = \mathbf{0} \quad \text{and} \quad C_2 = 0$$

For $n = 3$: $A_3 = 1$, $B_3 = 1$, and $C_{n-1} = 0$. From the 4th row of the table,

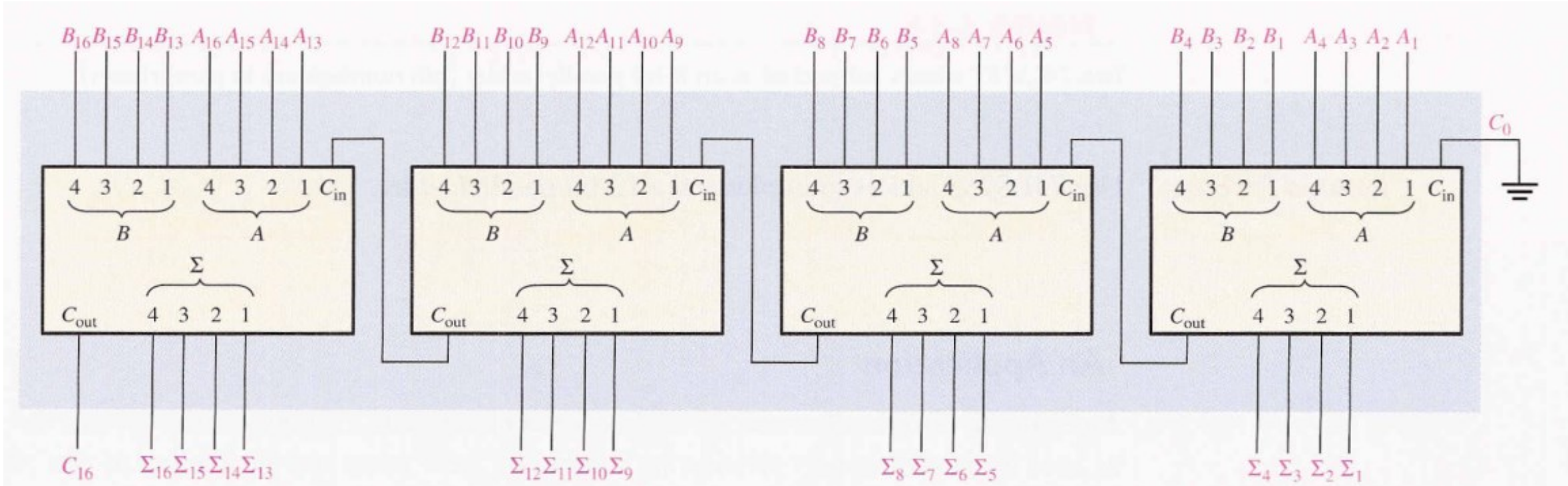
$$\Sigma_3 = \mathbf{0} \quad \text{and} \quad C_3 = 1$$

For $n = 4$: $A_4 = 1$, $B_4 = 1$, and $C_{n-1} = 1$. From the last row of the table,

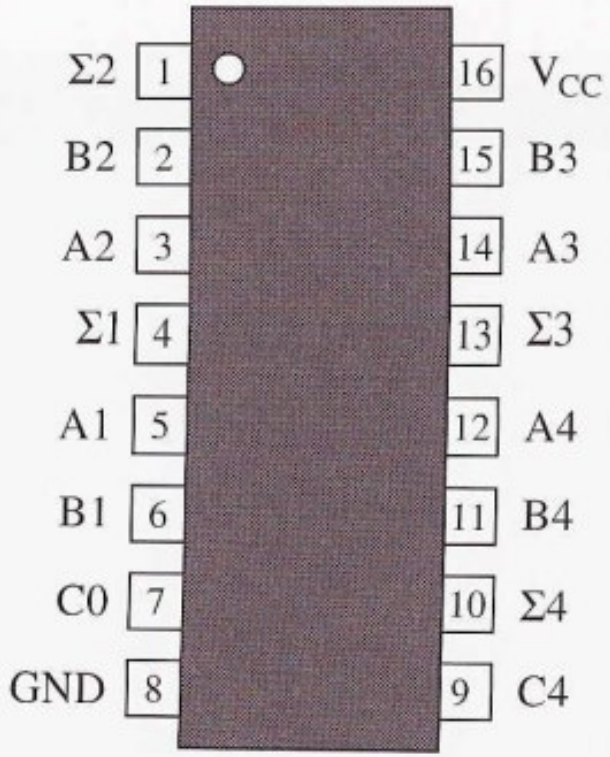
$$\Sigma_4 = \mathbf{1} \quad \text{and} \quad C_4 = \mathbf{1}$$

C_4 becomes the output carry; the sum of 1100 and 1100 is 11000.

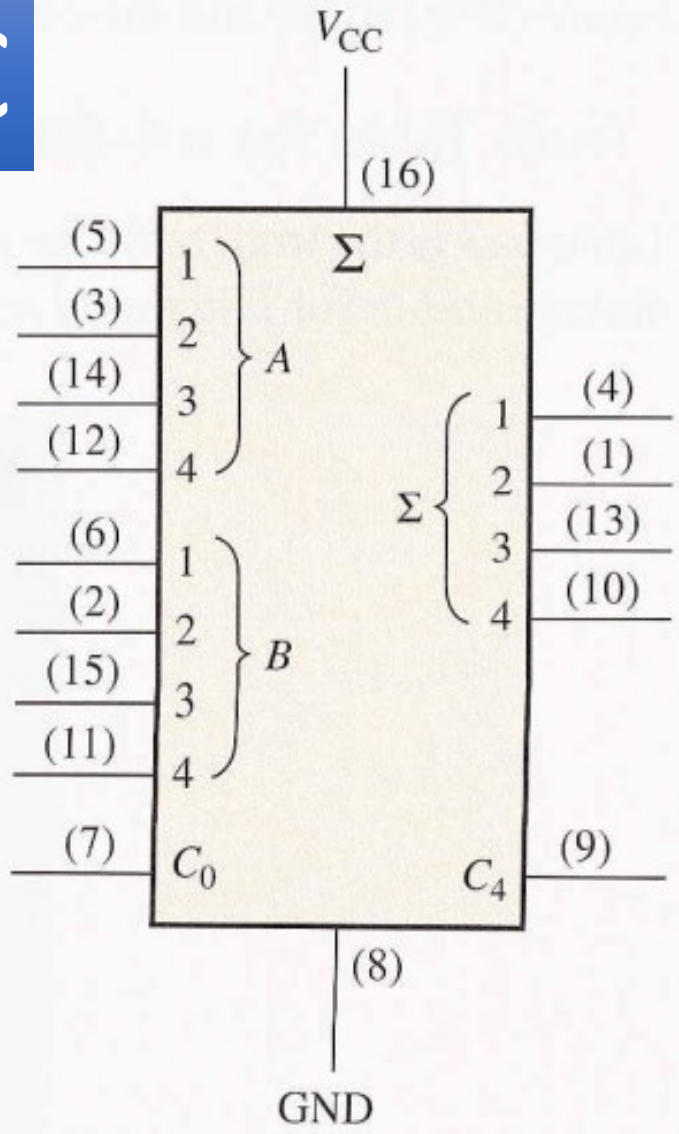
Cascading of four 4-bit adders to form a 16-bit adder



74LS283 IC



(a) Pin diagram of 74LS283



(b) 74LS283 logic symbol

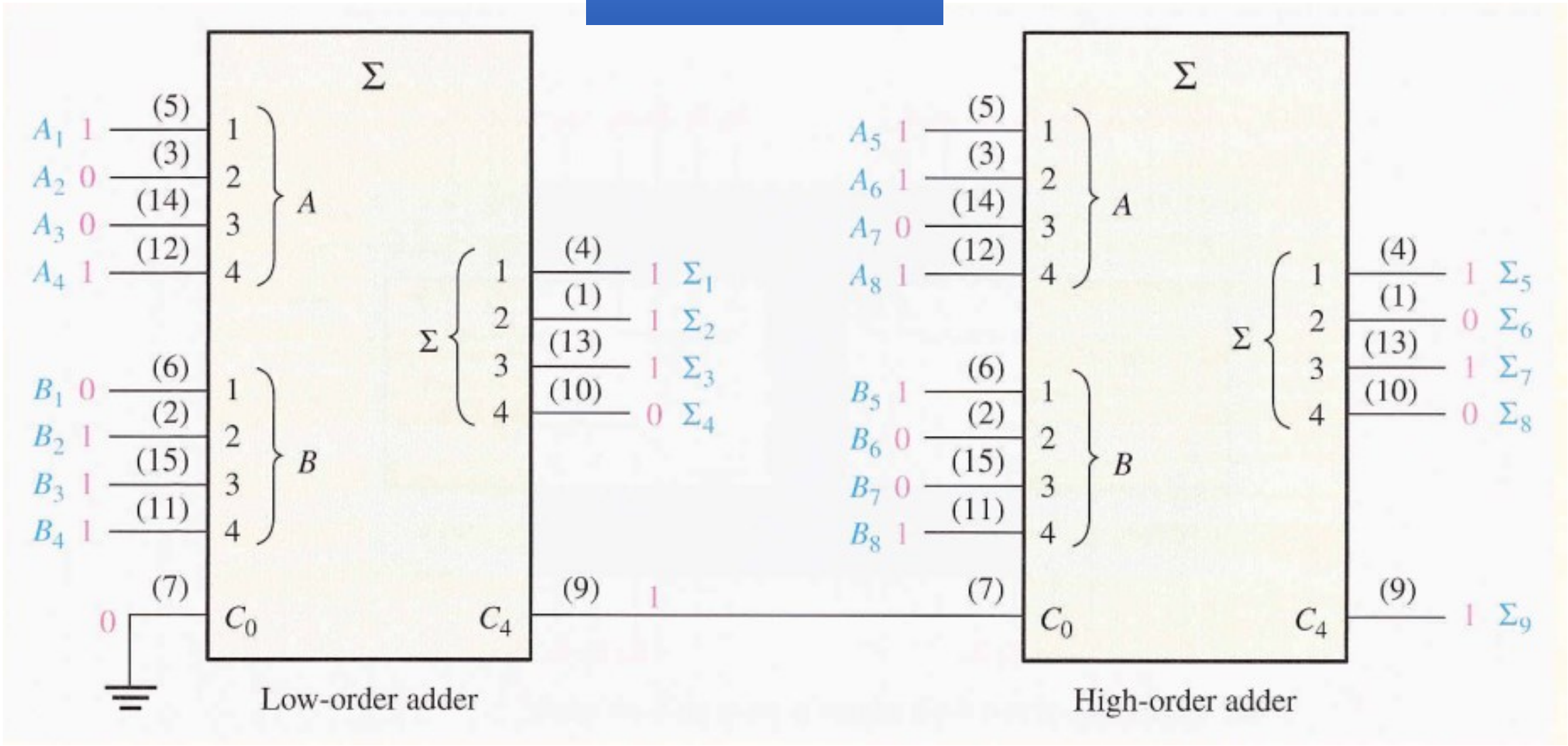
Ex-3

Show how two 74LS283 adders can be connected to form an 8-bit parallel adder. Show output bits for the following 8-bit input numbers:

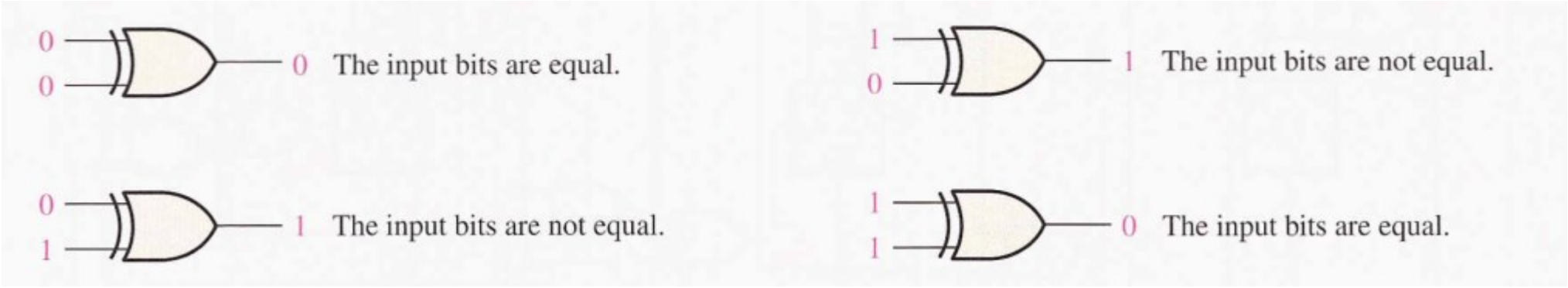
$$A_8A_7A_6A_5A_4A_3A_2A_1 = 10111001 \quad \text{and} \quad B_8B_7B_6B_5B_4B_3B_2B_1 = 10011110$$

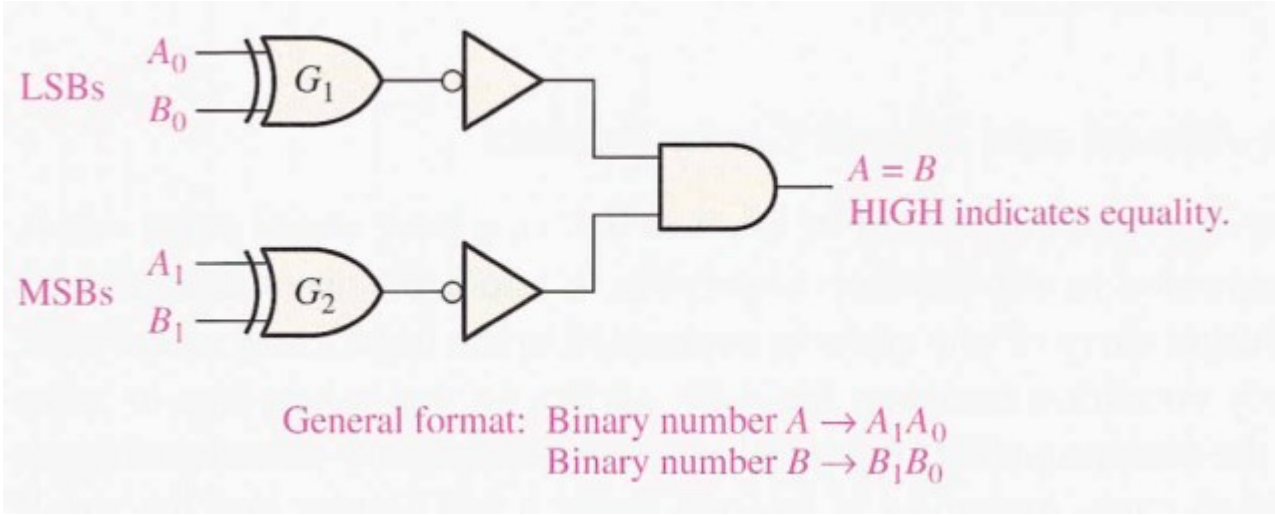
$$\Sigma_9\Sigma_8\Sigma_7\Sigma_6\Sigma_5\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 101010111$$

Solution


Σ
A1 1 (5) 1
A2 0 (3) 2
A3 0 (14) 3
A4 1 (12) 4
A
B1 0 (6) 1
B2 1 (2) 2
B3 1 (15) 3
B4 1 (11) 4
B
0 (7) C0
Σ 1 (4) 1 Σ1
2 (1) 1 Σ2
3 (13) 1 Σ3
4 (10) 0 Σ4
C4 (9) 1
Low-order adder
Σ
A5 1 (5) 1
A6 1 (3) 2
A7 0 (14) 3
A8 1 (12) 4
A
B5 1 (6) 1
B6 0 (2) 2
B7 0 (15) 3
B8 1 (11) 4
B
(7) C0
Σ 1 (4) 1 Σ5
2 (1) 0 Σ6
3 (13) 1 Σ7
4 (10) 0 Σ8
C4 (9) 1 Σ9
High-order adder

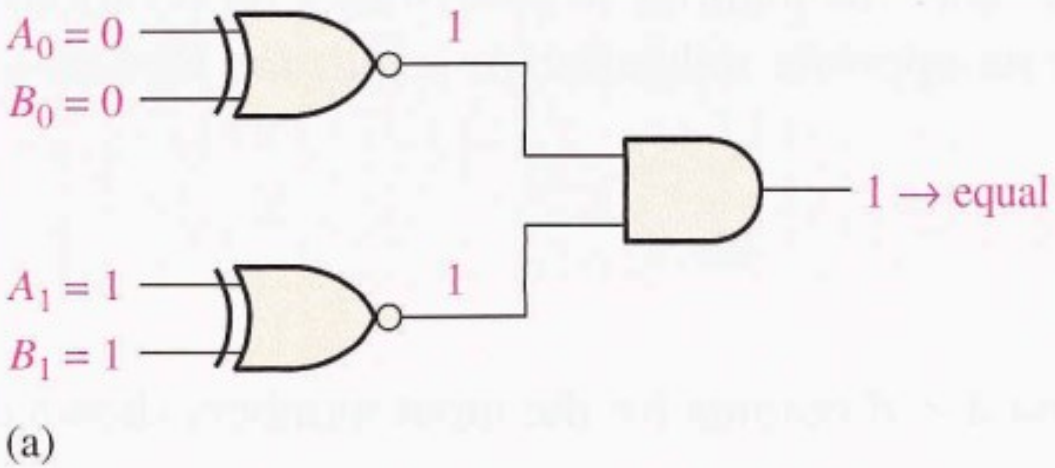
Comparators



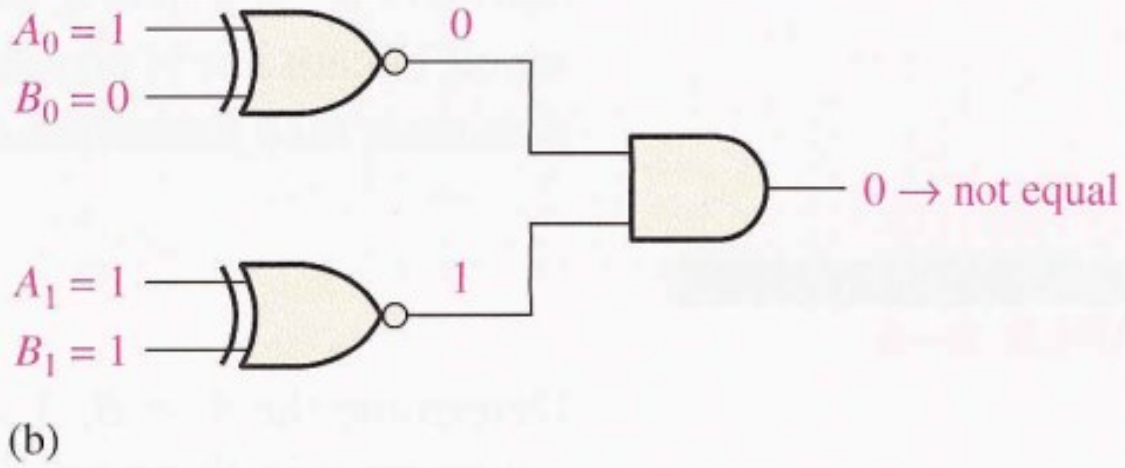


Ex-4: Compare the following two 2-bit Binary numbers

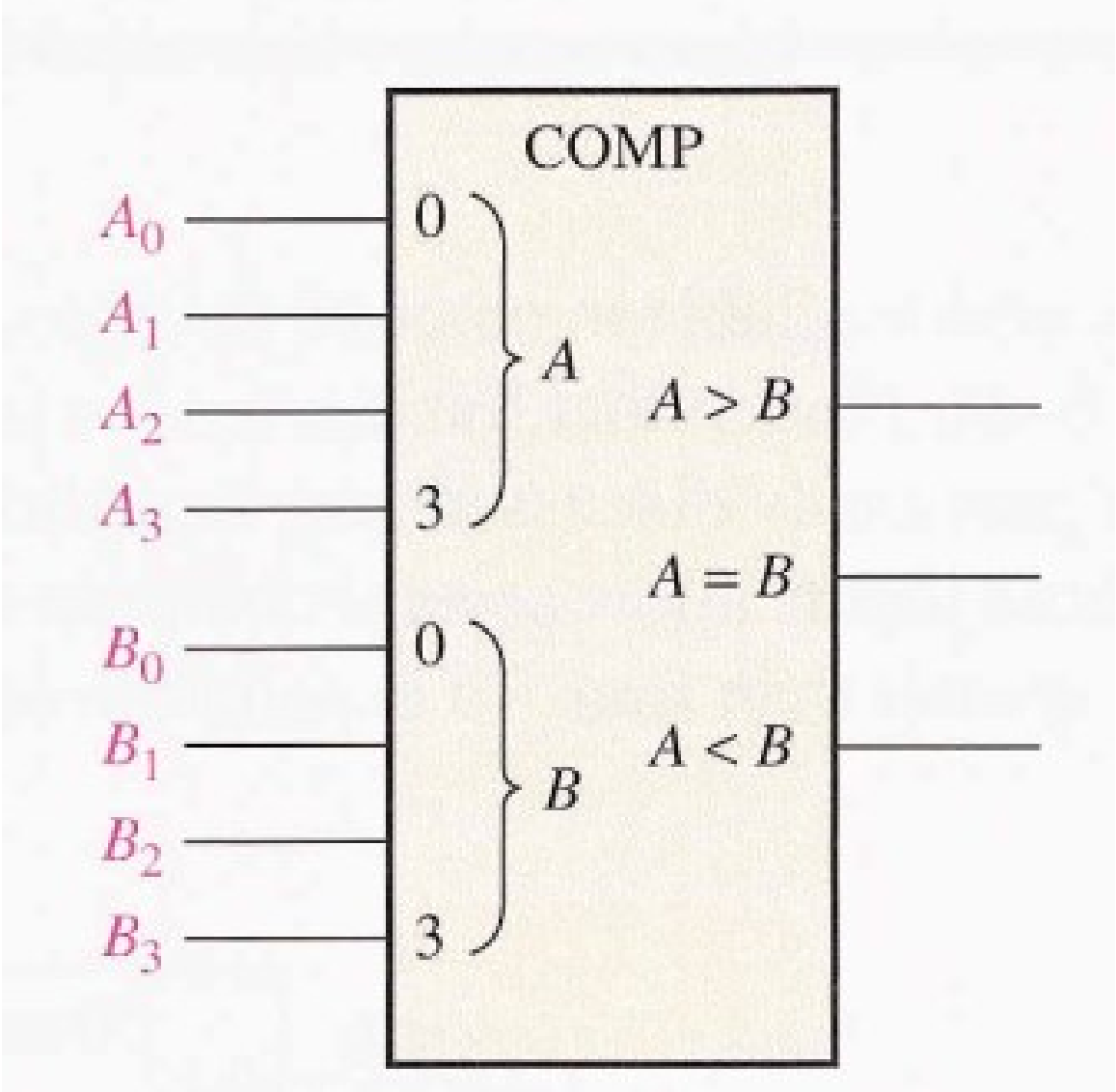
(a) 10 and 10



(b) 11 and 10

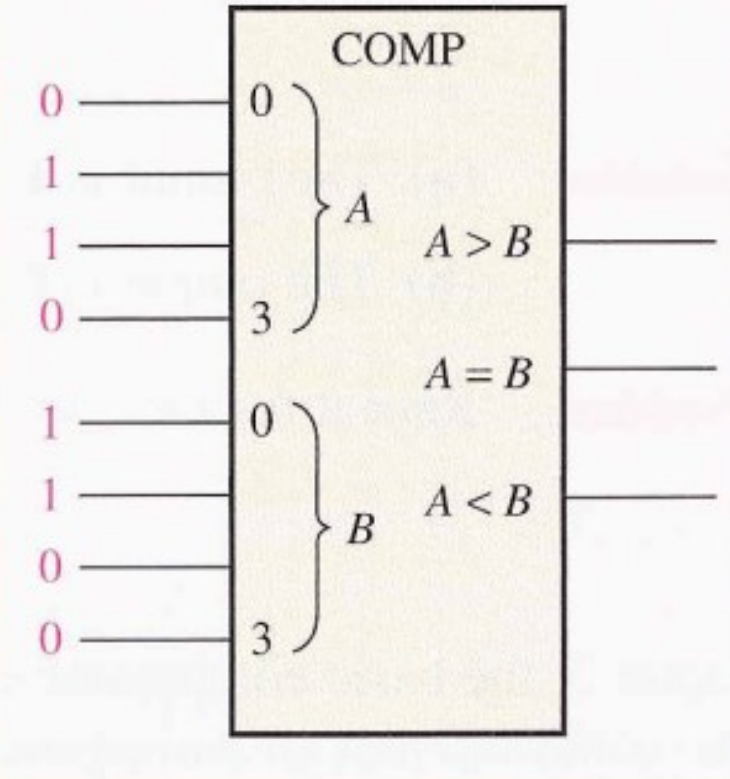


4-bit Comparator Logic Symbol



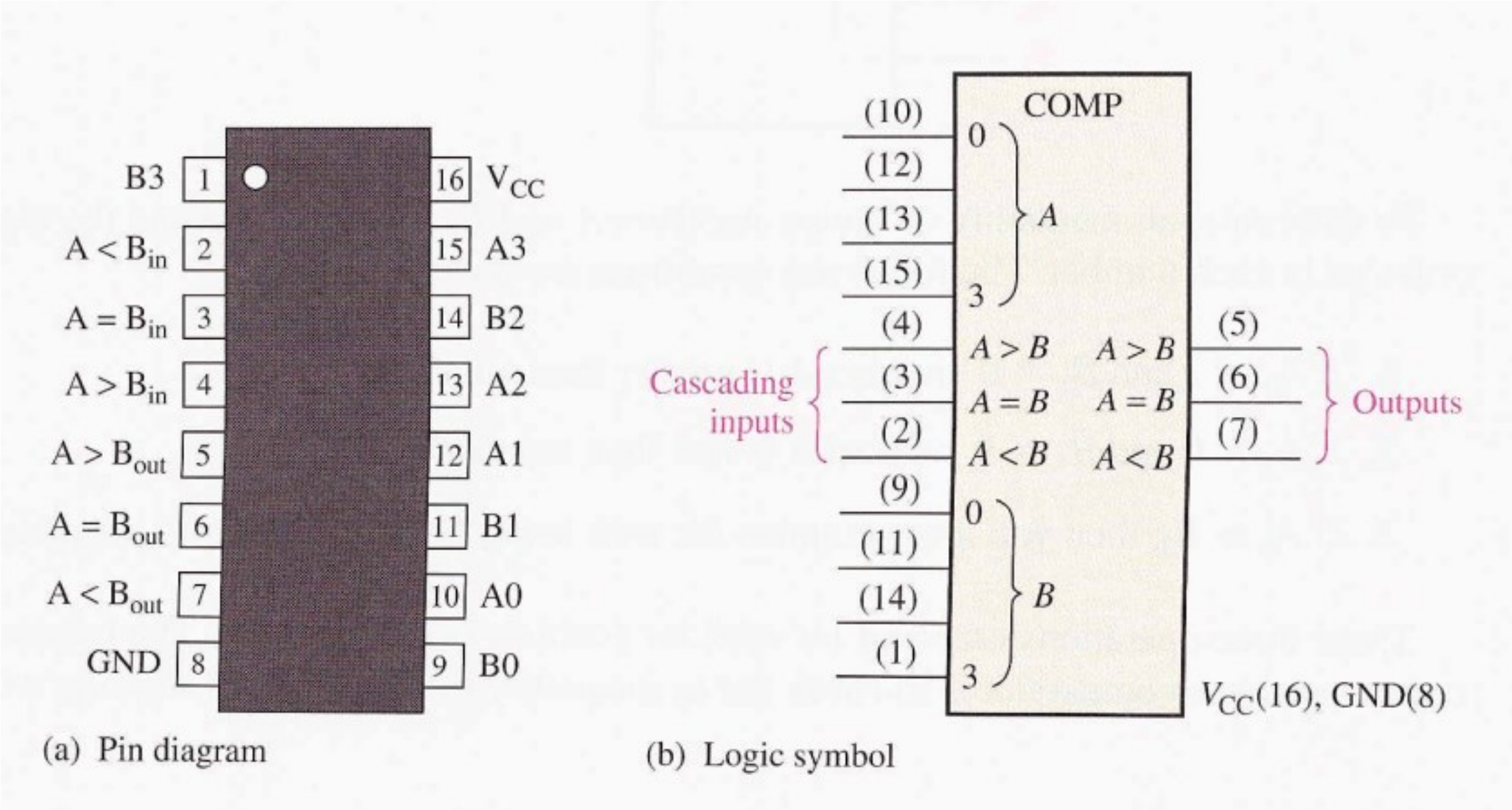
EX-5

Determine the $A = B$, $A > B$, and $A < B$ outputs for the input numbers shown on the comparator in Figure



The number on the A inputs is 0110 and the number on the B inputs is 0011. The **$A > B$ output is HIGH and the other outputs are LOW.**

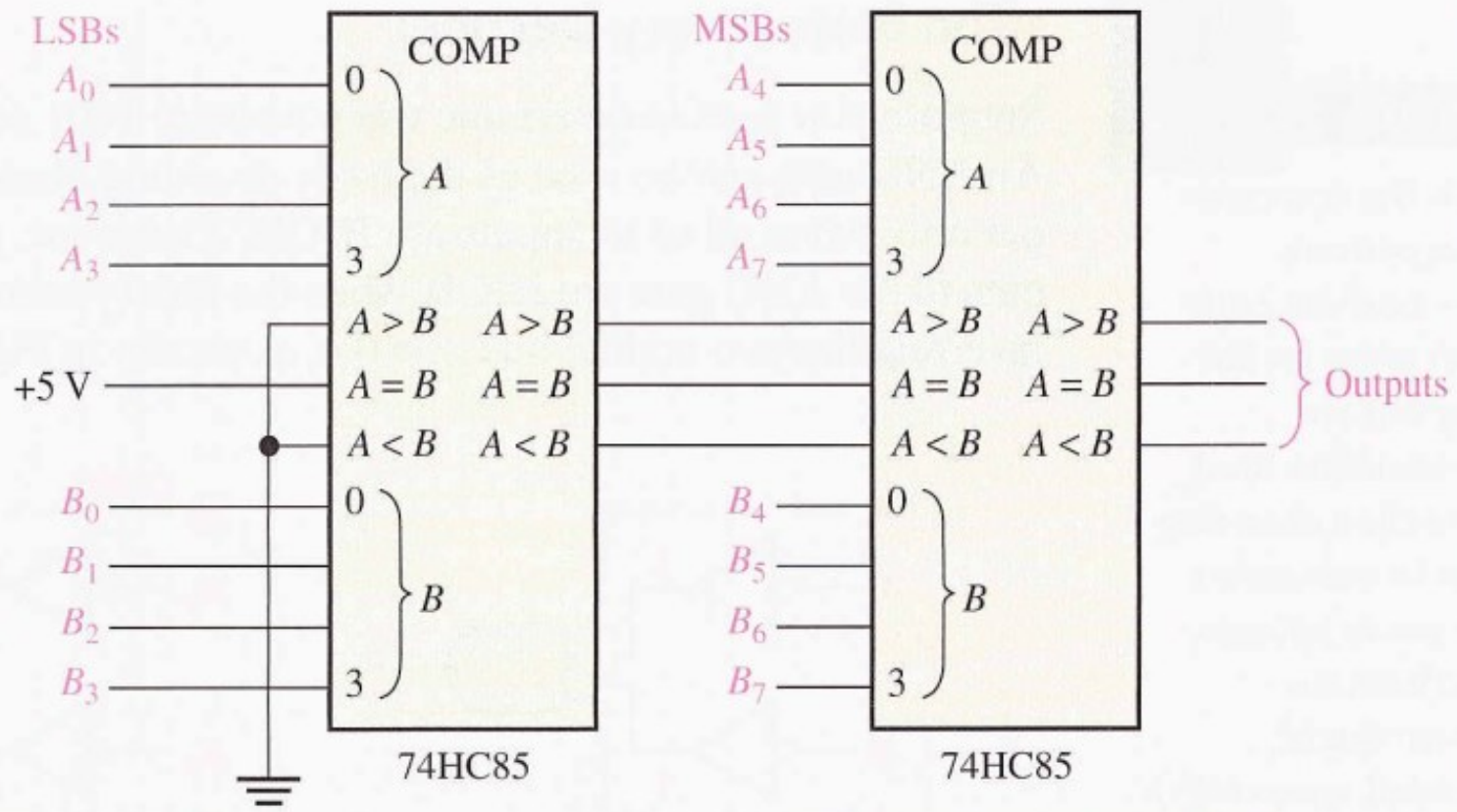
74HC85 IC


B3
1
A < B in
2
A = B in
3
A > B in
4
A > B out
5
A = B out
6
A < B out
7
GND
8
16
V CC
15
A3
14
B2
13
A2
12
A1
11
B1
10
A0
9
B0
(a) Pin diagram
COMP
(10)
(12)
(13)
(15)
0
A
3
(4)
(3)
(2)
Cascading inputs
A > B
A = B
A < B
A > B
A = B
A < B
(5)
(6)
(7)
Outputs
(9)
(11)
(14)
(1)
0
B
3
V CC (16), GND(8)
(b) Logic symbol

EX-6

Use 74HC85 comparators to compare the magnitudes of two 8-bit numbers. Show the comparators with proper interconnections.

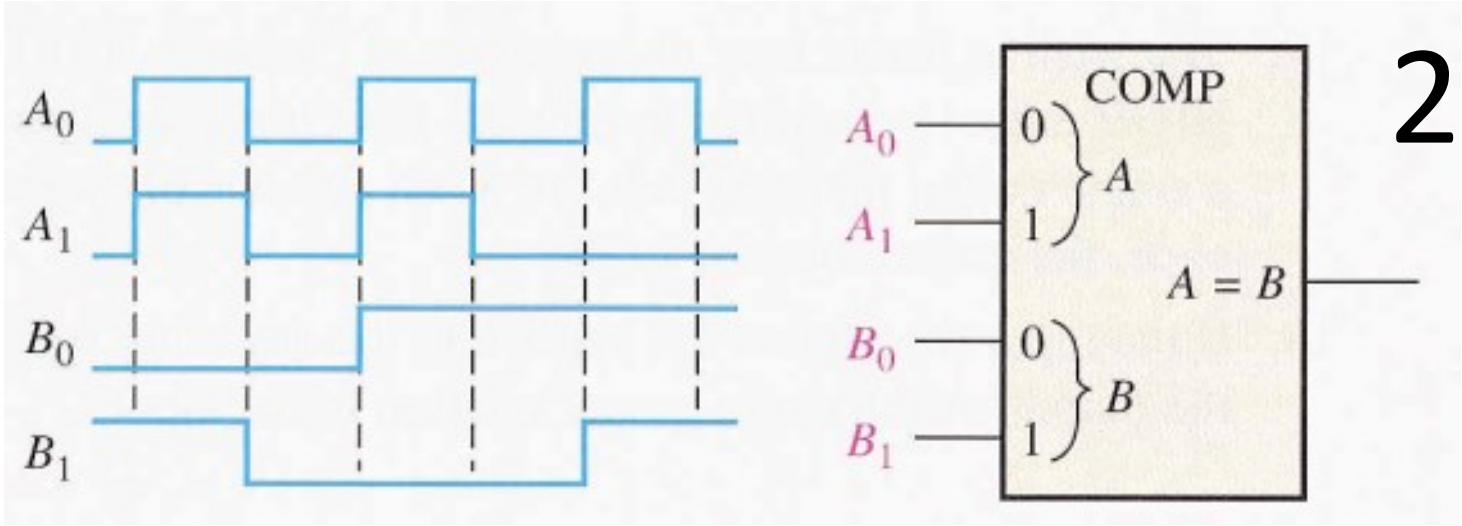
Two 74HC85s are required to compare two 8-bit numbers.



Home Work

1 Use 4-bit Binary adder to find 1011+1010 2M

2 Sketch the waveform A=B 2M

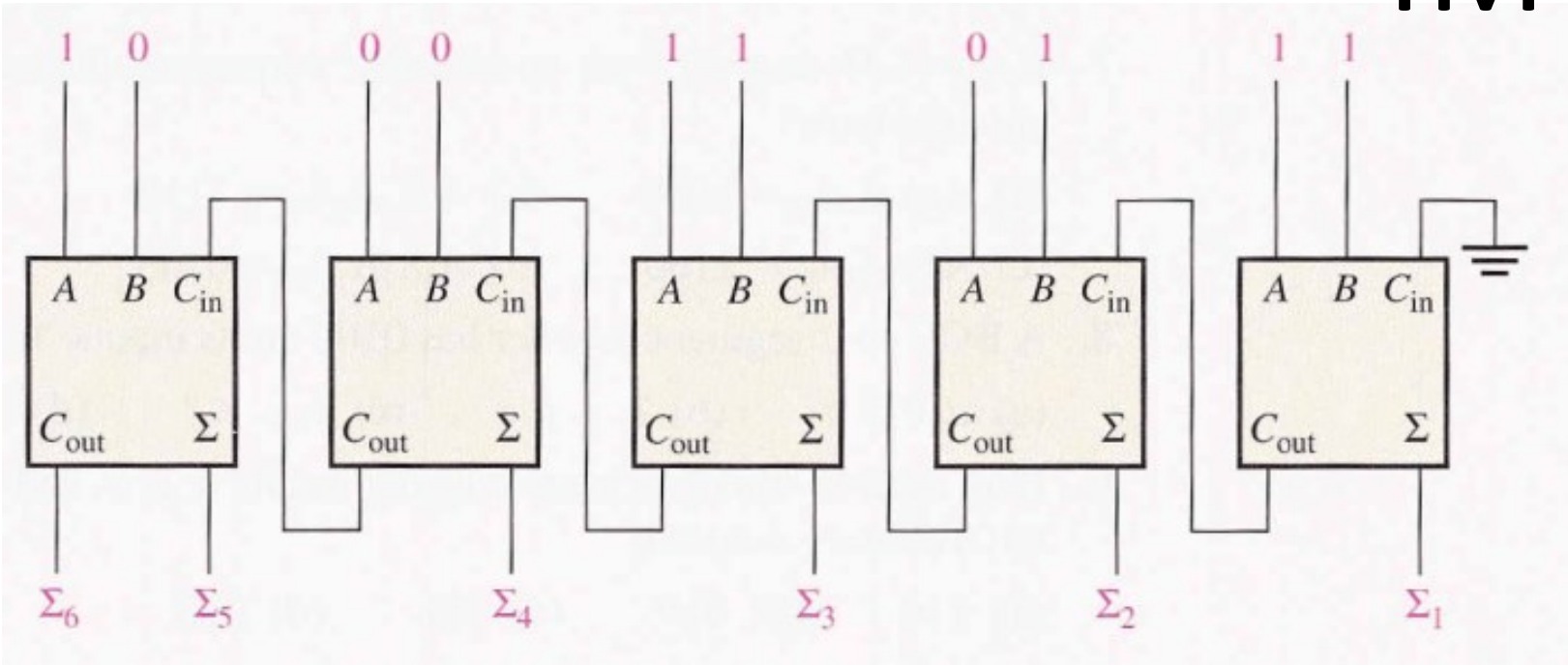


3 4M

In the process of checking a 74LS283 4-bit parallel adder, the following voltage levels are observed on its pins: 1-HIGH, 2-HIGH, 3-HIGH, 4-HIGH, 5-LOW, 6-LOW, 7-LOW, 9-HIGH, 10-LOW, 11-HIGH, 12-LOW, 13-HIGH, 14-HIGH, and 15-HIGH. Determine if the IC is functioning properly.

4M

4- Determine the complete Sum



University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

Lec. Dr. Taif Alawsi

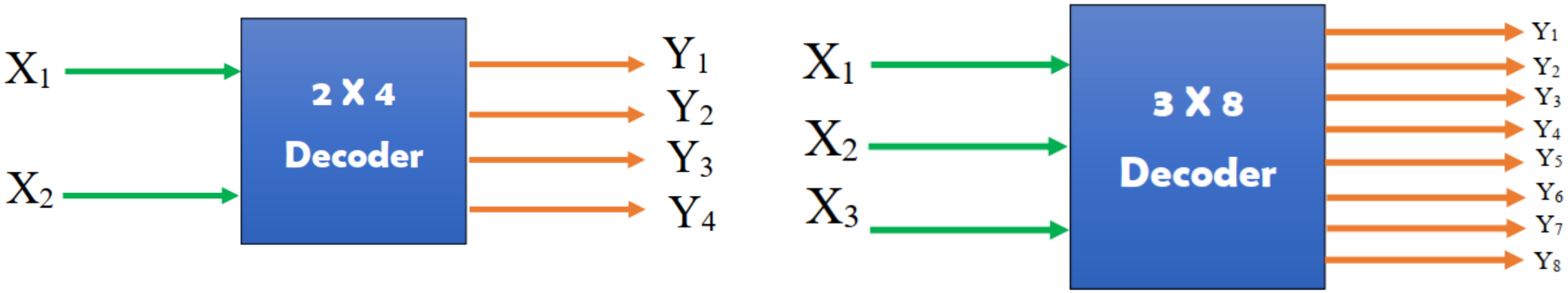
Lec. 7: Functions of Combinational Logic 2: 2024-Nov-10

Lecture Outline

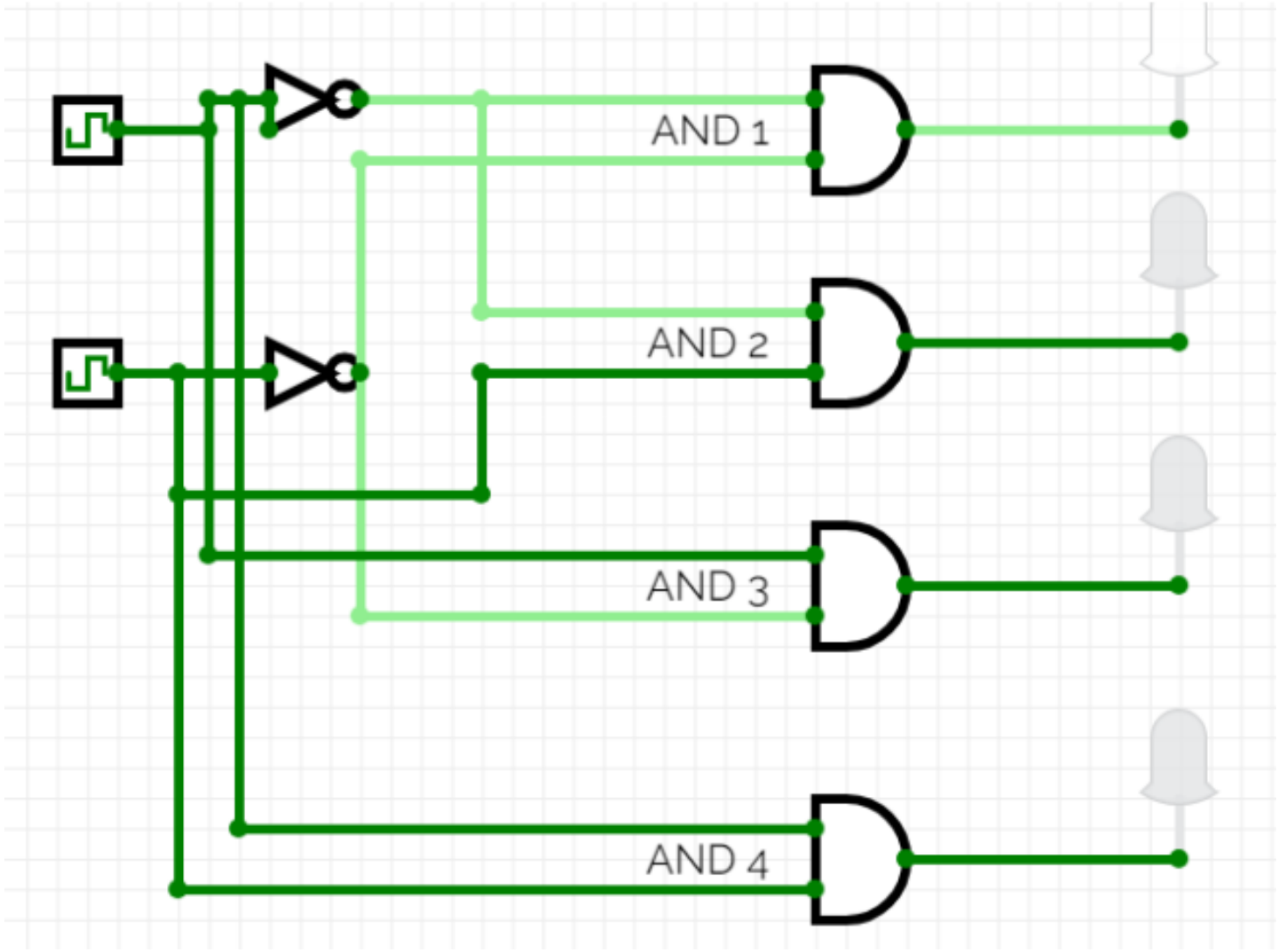
1. Decoders
2. Encoders
3. Multiplexers
4. Demultiplexers
5. 74HC47 IC (BCD/7-SEG)
6. HW

Decoders

A **decoder** is a digital circuit that detects the presence of a specified combination of bits (code) on its inputs and indicates the presence of that code by a specified output level. In its general form, a decoder has *n* input lines to handle *n* bits and from one to 2^n output lines to indicate the presence of one or more *n*-bit combinations



2X4 LINE Decoder



Truth Table for 2X4 Line Decoder

Inputs		Outputs			
X1	X2	Y1	Y2	Y3	Y4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

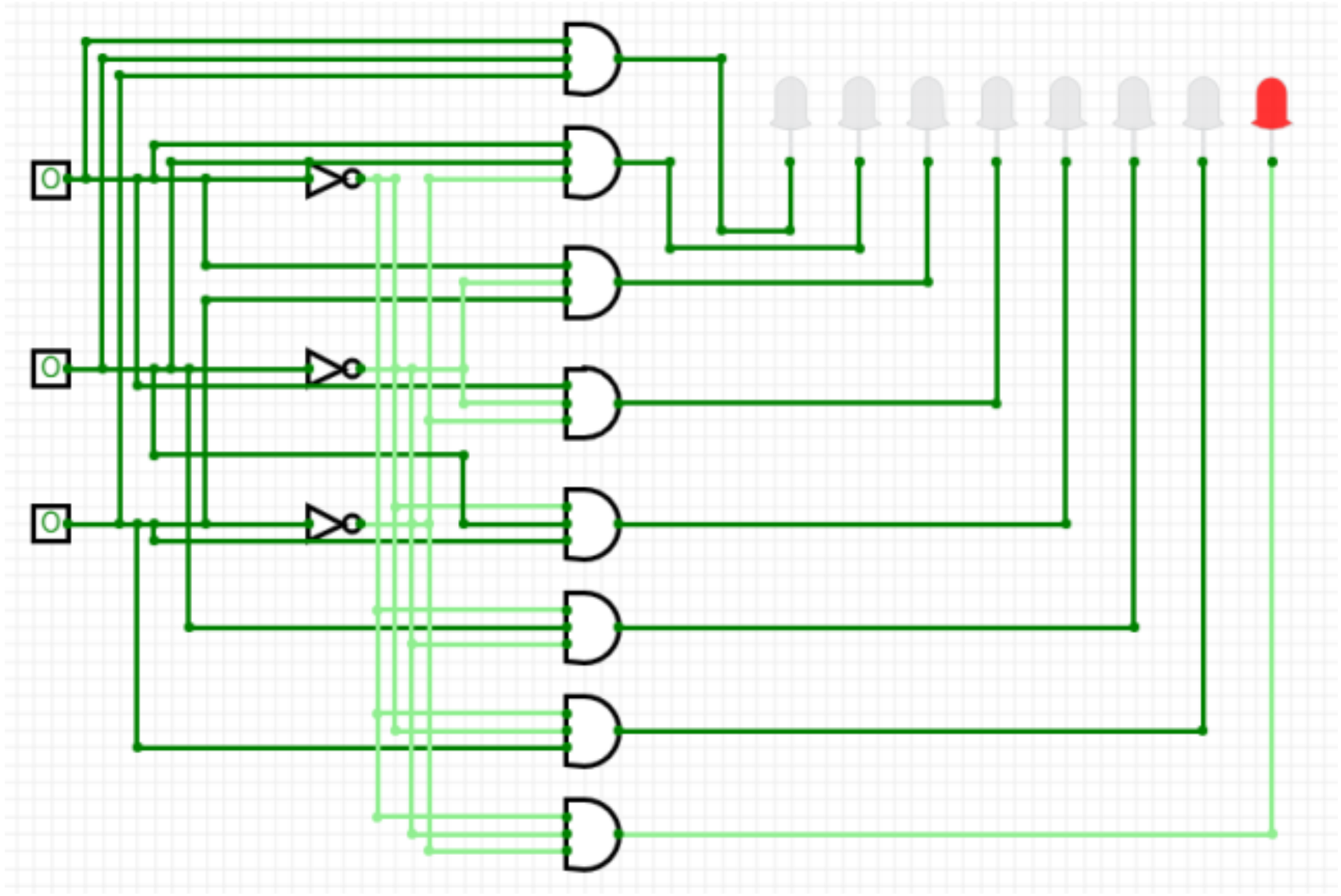
$$Y_1 = \overline{X_1 X_2}$$

$$Y_2 = \overline{X_1} X_2$$

$$Y_3 = X_1 \overline{X_2}$$

$$Y_4 = X_1 X_2$$

3X8 LINE Decoder



Truth Table for 3X8 Line Decoder

Inputs			Outputs							
X1	X2	X3	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
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

$$Y_1 = \overline{X_1 X_2 X_3}$$

$$Y_2 = \overline{X_1 X_2} X_3$$

$$Y_3 = \overline{X_1} X_2 \overline{X_3}$$

$$Y_4 = \overline{X_1} X_2 X_3$$

$$Y_5 = X_1 \overline{X_2 X_3}$$

$$Y_6 = X_1 \overline{X_2} X_3$$

$$Y_7 = X_1 X_2 \overline{X_3}$$

$$Y_8 = X_1 X_2 X_3$$

EX-1

Using the equations below, design a 4X16 line decoder

$$Y_1 = \overline{X_1X_2X_3X_4} \quad Y_2 = \overline{X_1X_2X_3}X_4 \quad Y_3 = \overline{X_1X_2}X_3\overline{X_4} \quad Y_4 = \overline{X_1X_2}X_3X_4$$

$$Y_5 = \overline{X_1}X_2\overline{X_3X_4} \quad Y_6 = \overline{X_1}X_2\overline{X_3}X_4 \quad Y_7 = \overline{X_1}X_2X_3\overline{X_4} \quad Y_8 = \overline{X_1}X_2X_3X_4$$

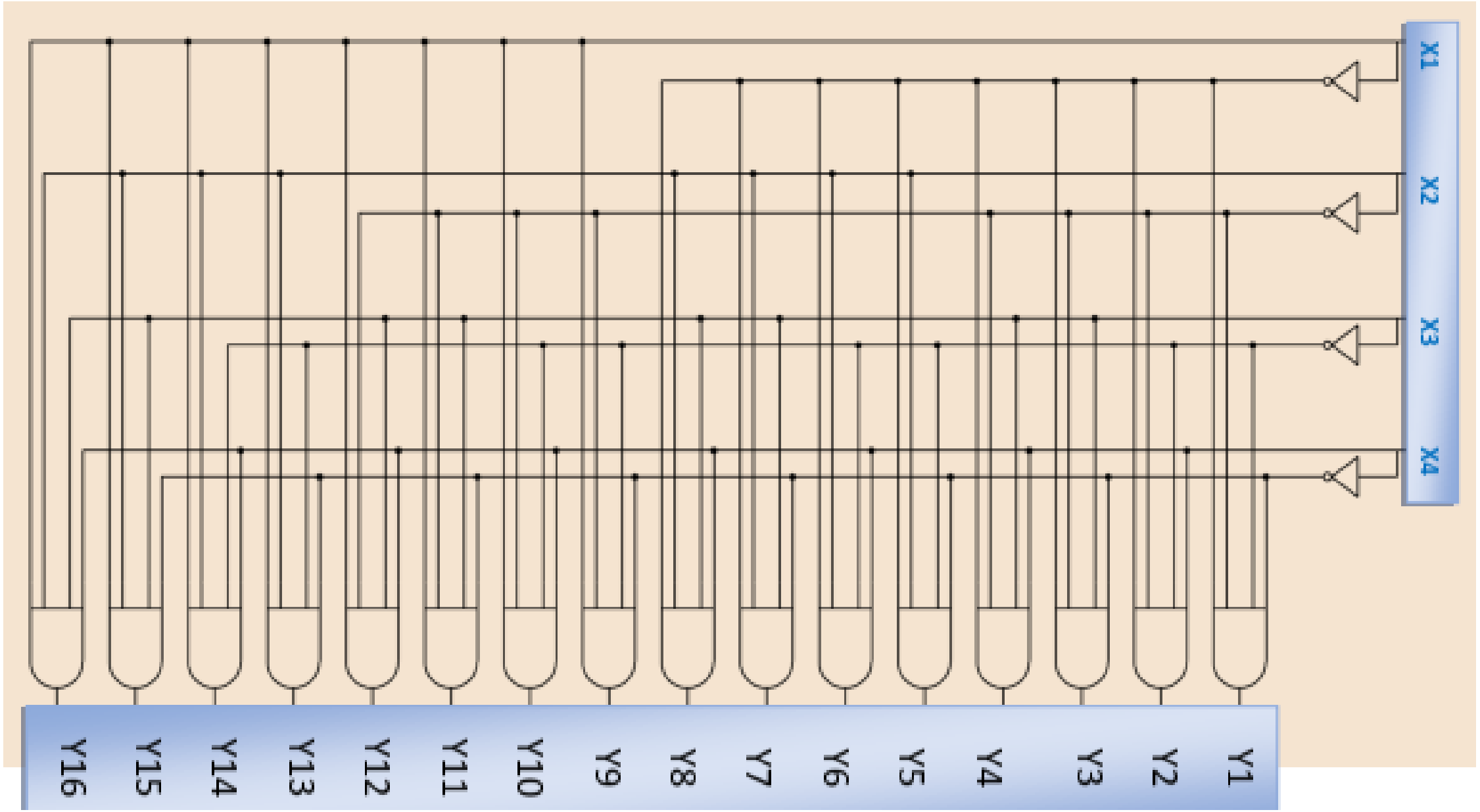
$$Y_9 = X_1\overline{X_2X_3X_4} \quad Y_{10} = X_1\overline{X_2X_3}X_4 \quad Y_{11} = X_1\overline{X_2}X_3\overline{X_4} \quad Y_{12} = X_1\overline{X_2}X_3X_4$$

$$Y_{13} = X_1X_2\overline{X_3X_4} \quad Y_{14} = X_1X_2\overline{X_3}X_4 \quad Y_{15} = X_1X_2X_3\overline{X_4} \quad Y_{16} = X_1X_2X_3X_4$$

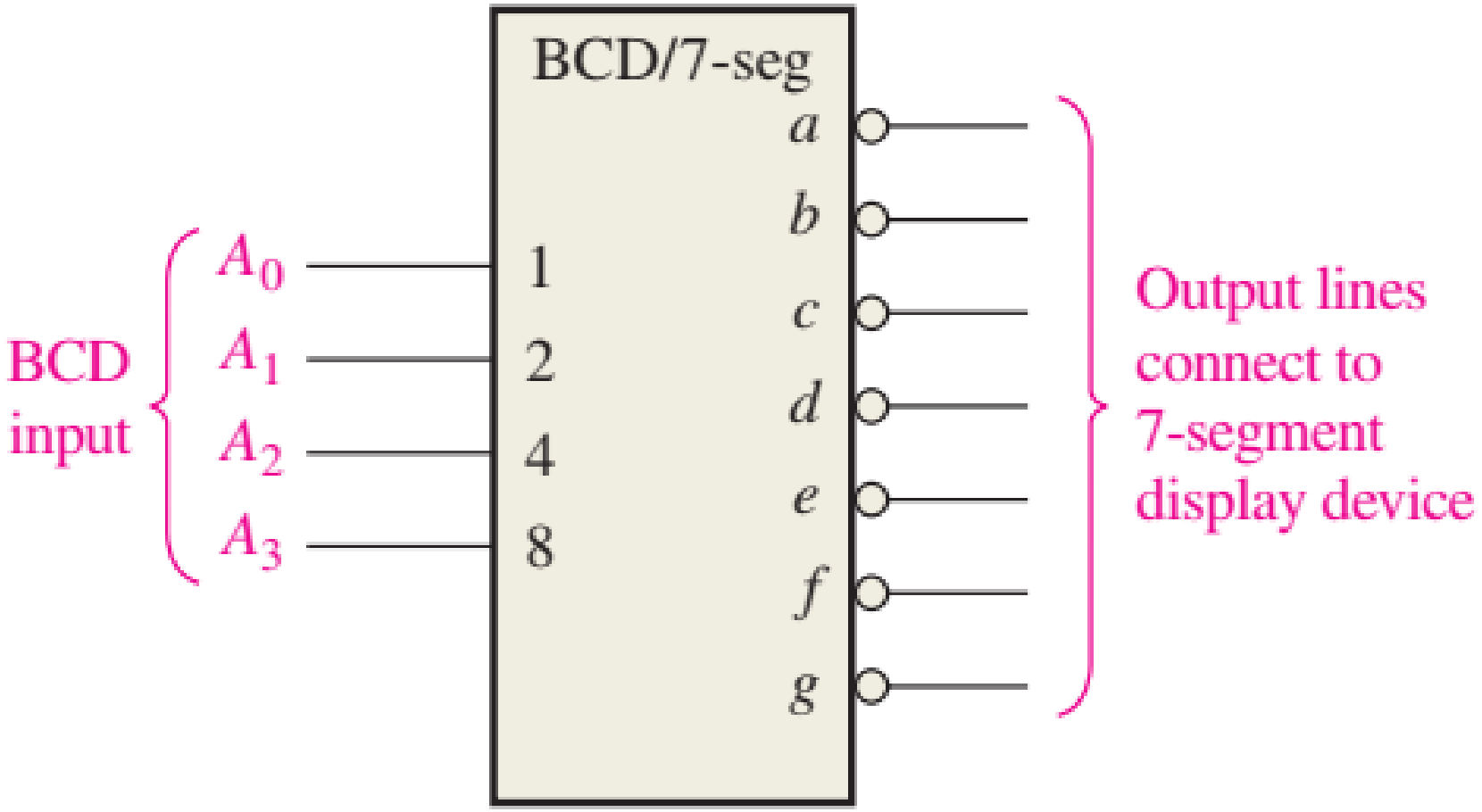
Sol

Inputs				Outputs															
X1	X2	X3	X4	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Sol



BCD/7-SEG

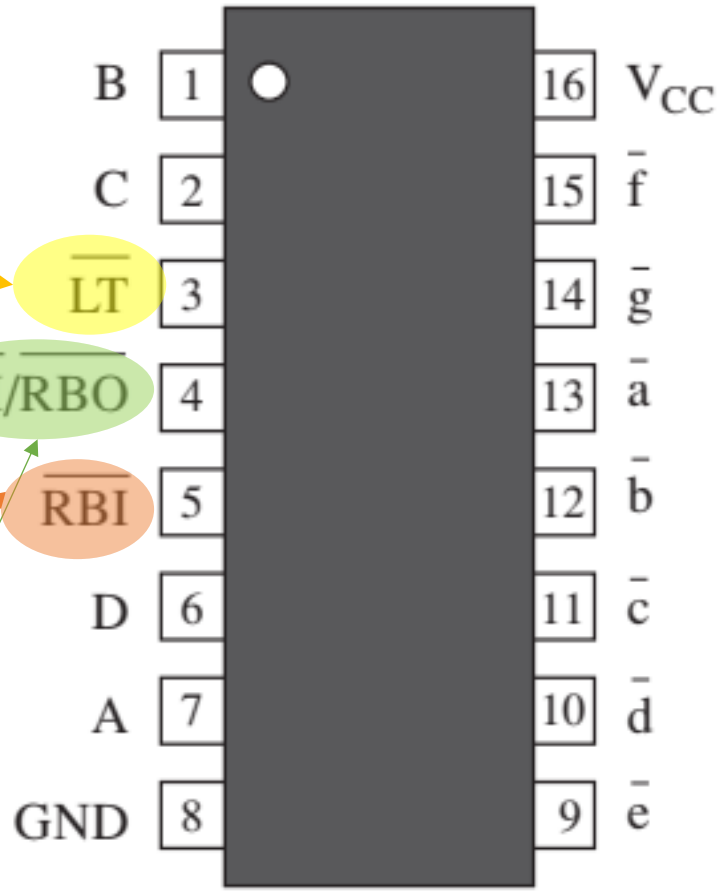

BCD/7-seg
BCD input
A_0
A_1
A_2
A_3
1
2
4
8
a
b
c
d
e
f
g
Output lines connect to 7-segment display device

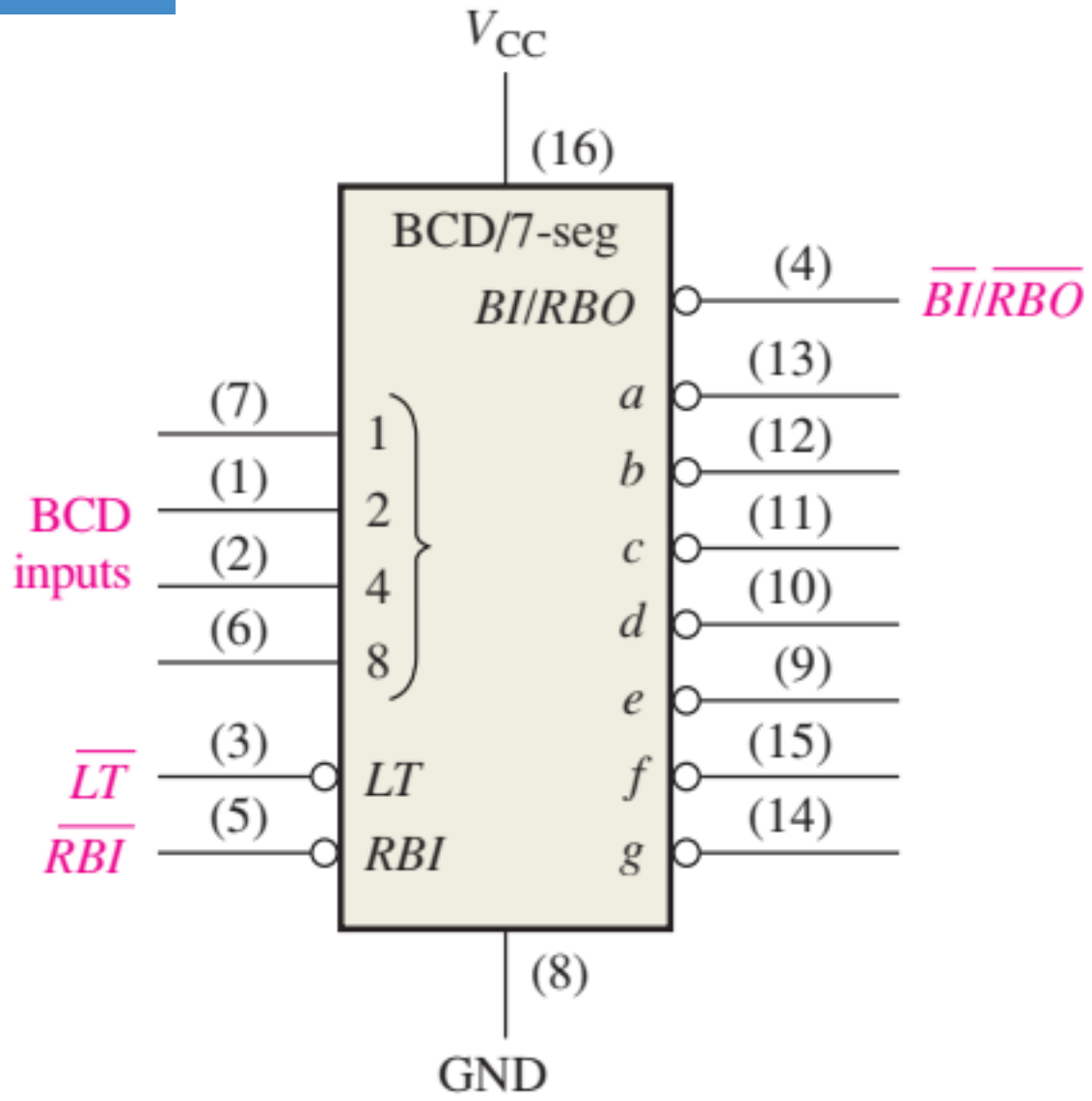
74HC47 IC

Ripple Blanking Input

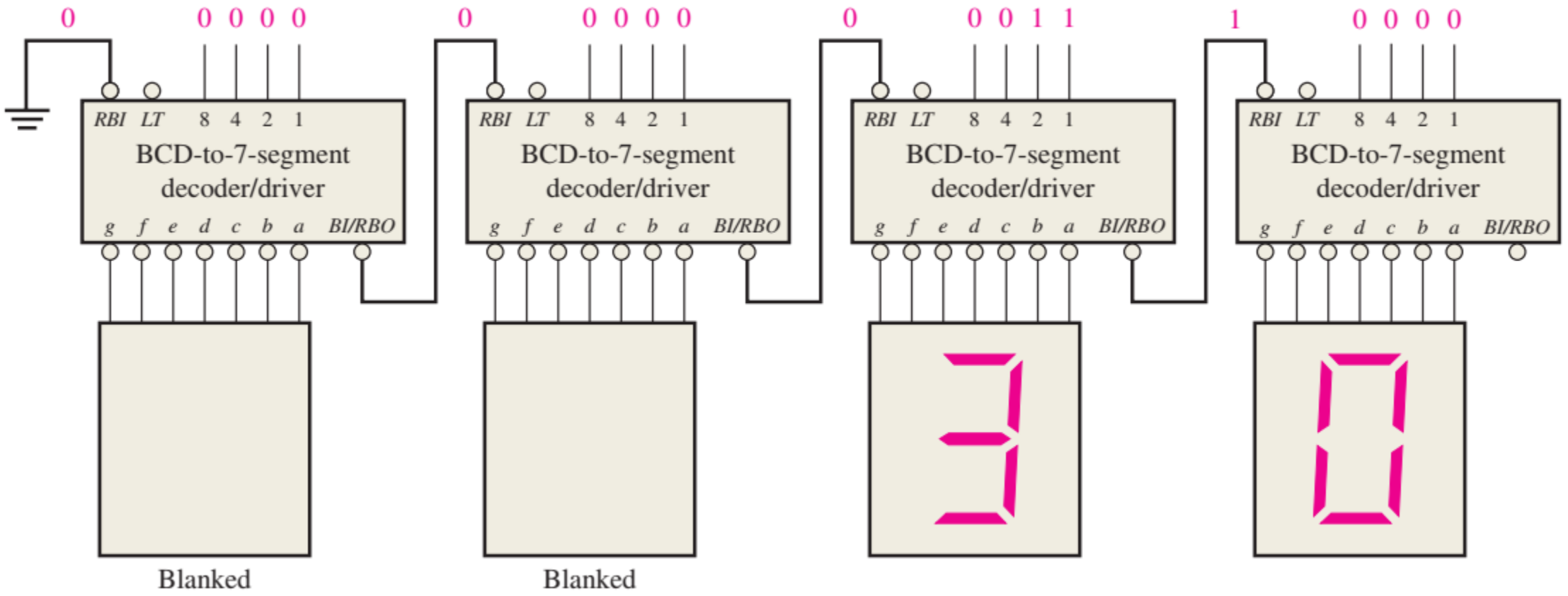
Lamp Test

Blanking Input / Ripple Blanking Output

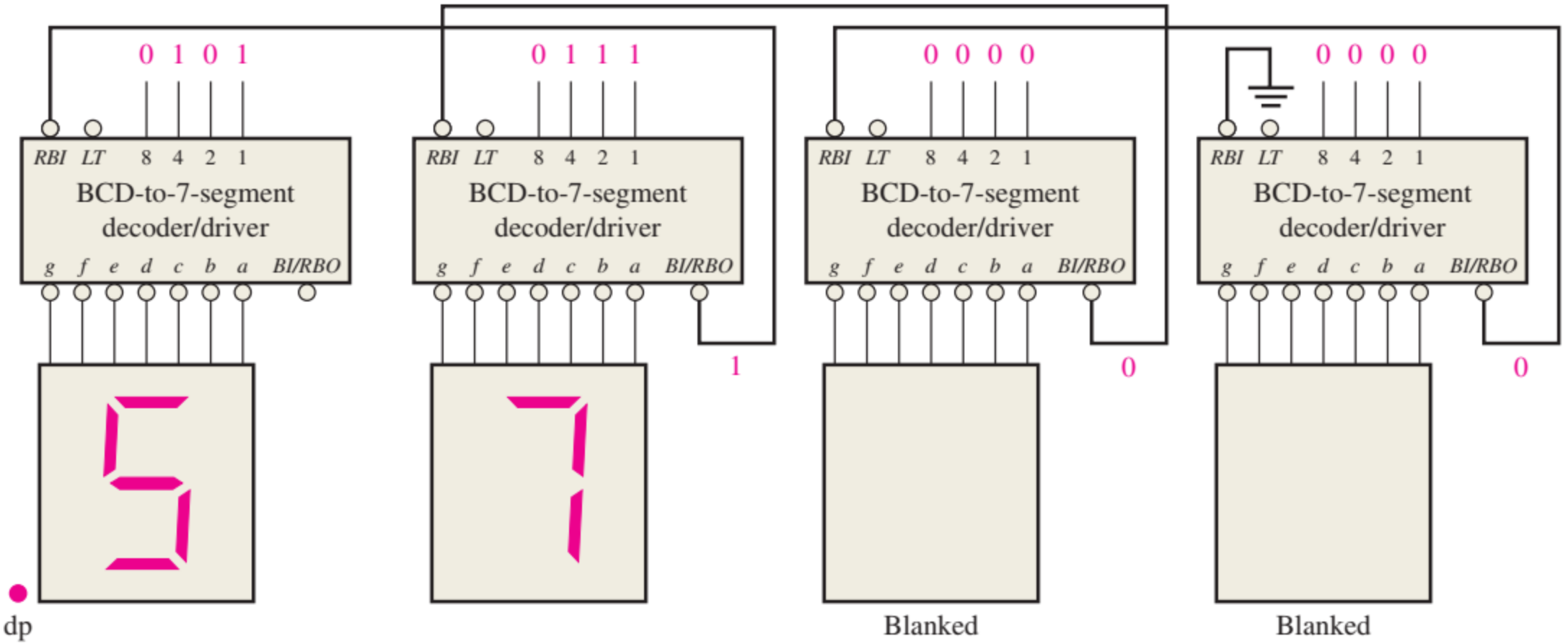




Leading zero suppression

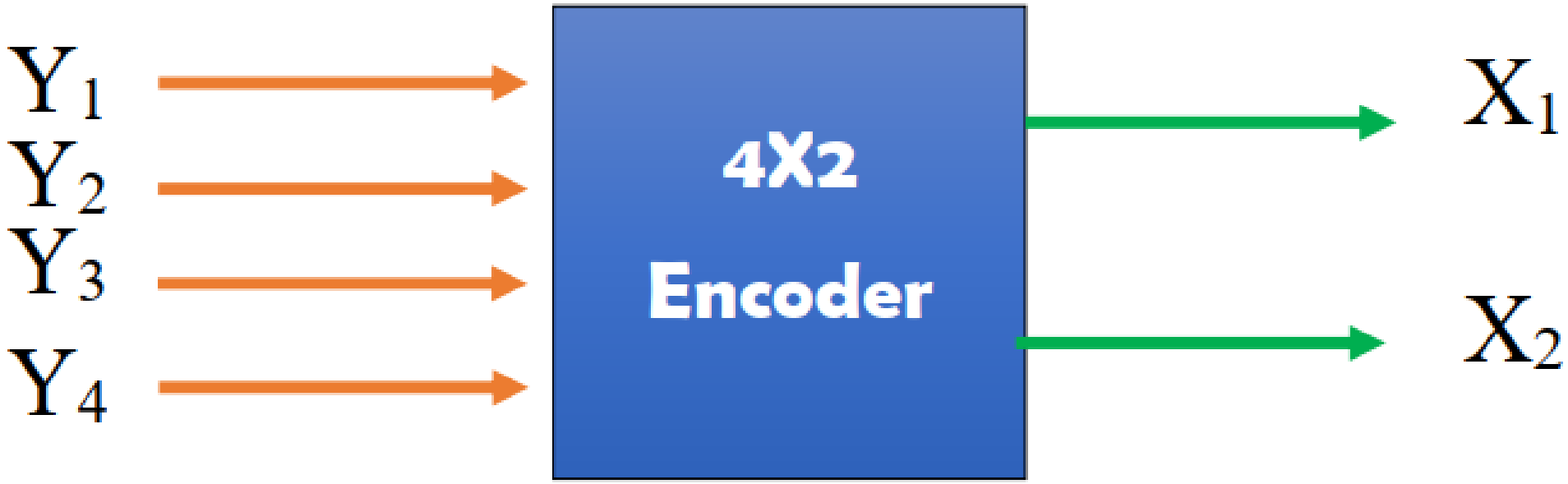


Trailing zero suppression

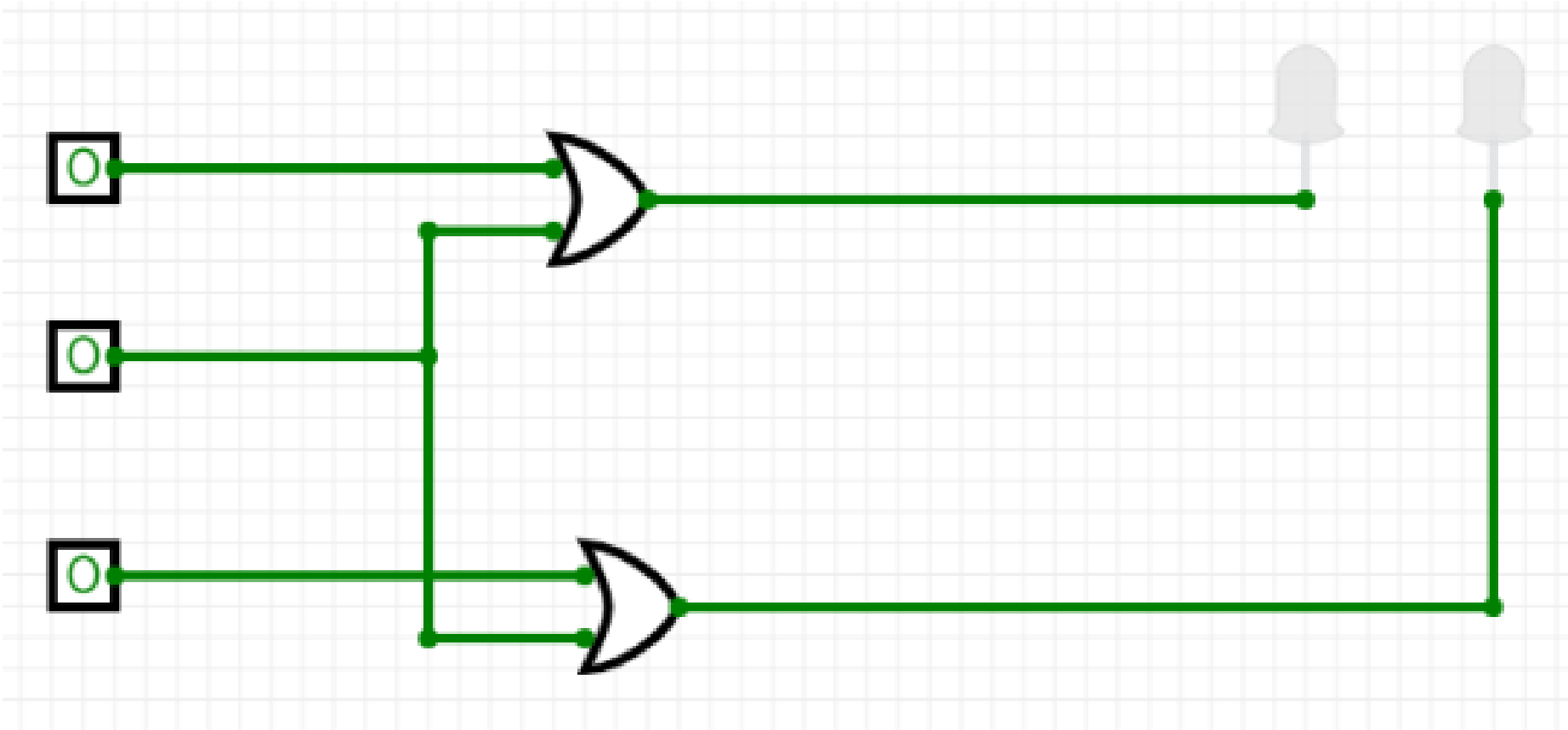


Encoder

An **encoder** is a combinational logic circuit that essentially performs a "reverse" decoder function. An encoder accepts an active level on one of its inputs representing a digit, such as a decimal or octal digit, and converts it to a coded output, such as BCD or binary. Encoders can also be devised to encode various symbols and alphabetic characters. The process of converting from familiar symbols or numbers to a coded format is called *encoding.*



4 x 2 Line Encoder



Truth Table for 4X2 Line Encoder

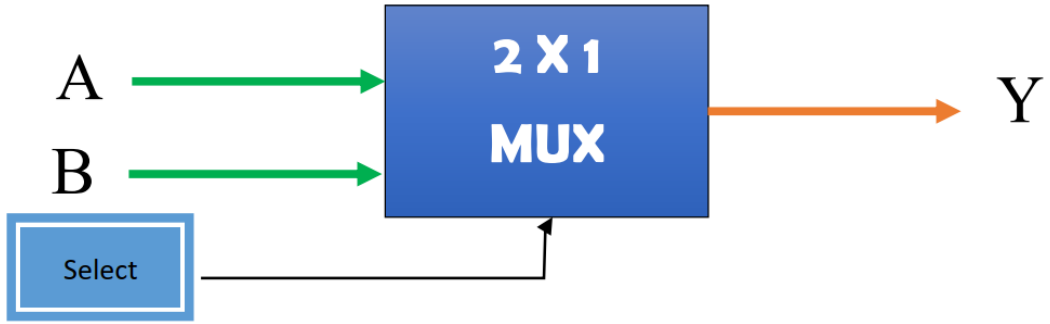
Inputs				Outputs	
Y1	Y2	Y3	Y4	X1	X2
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

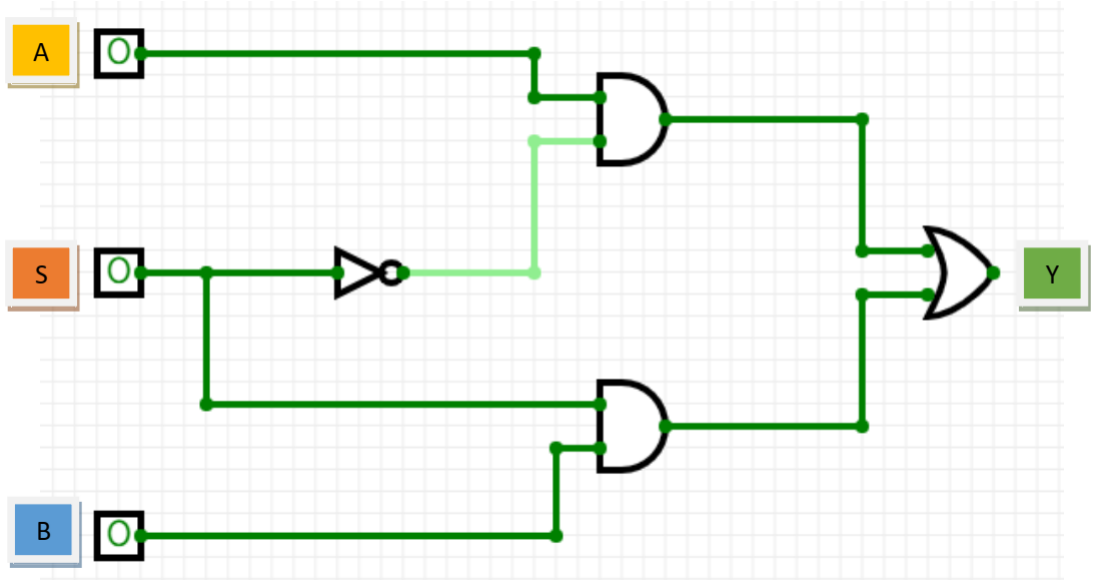
$$X_1 = Y_2 + Y_1 \qquad X_2 = Y_3 + Y_1$$

Multiplexer

A **multiplexer (MUX)** is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination. The basic multiplexer has several data-input lines and a single output line. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line. Multiplexers are also known as **data selectors**

$$N_S = LOG_2(N_V)$$





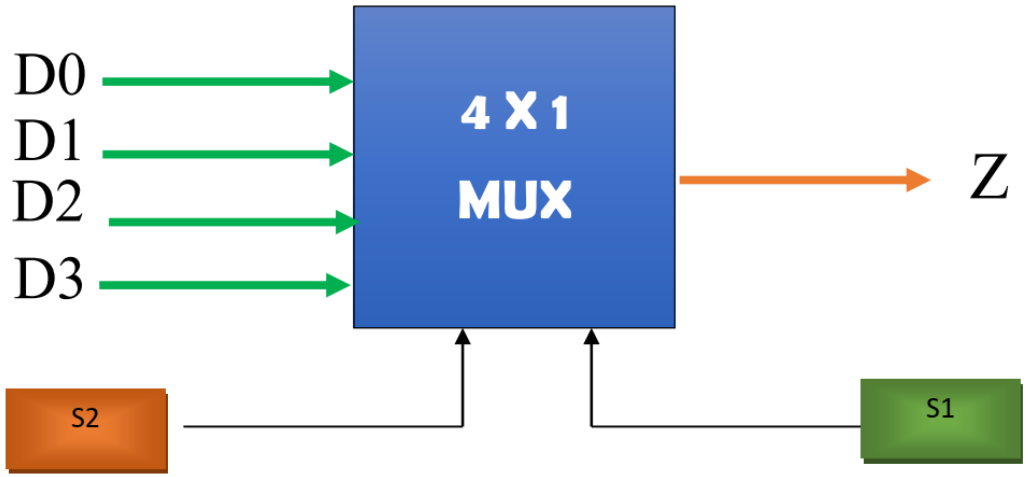
Truth Table for 2X1 Line Multiplexer

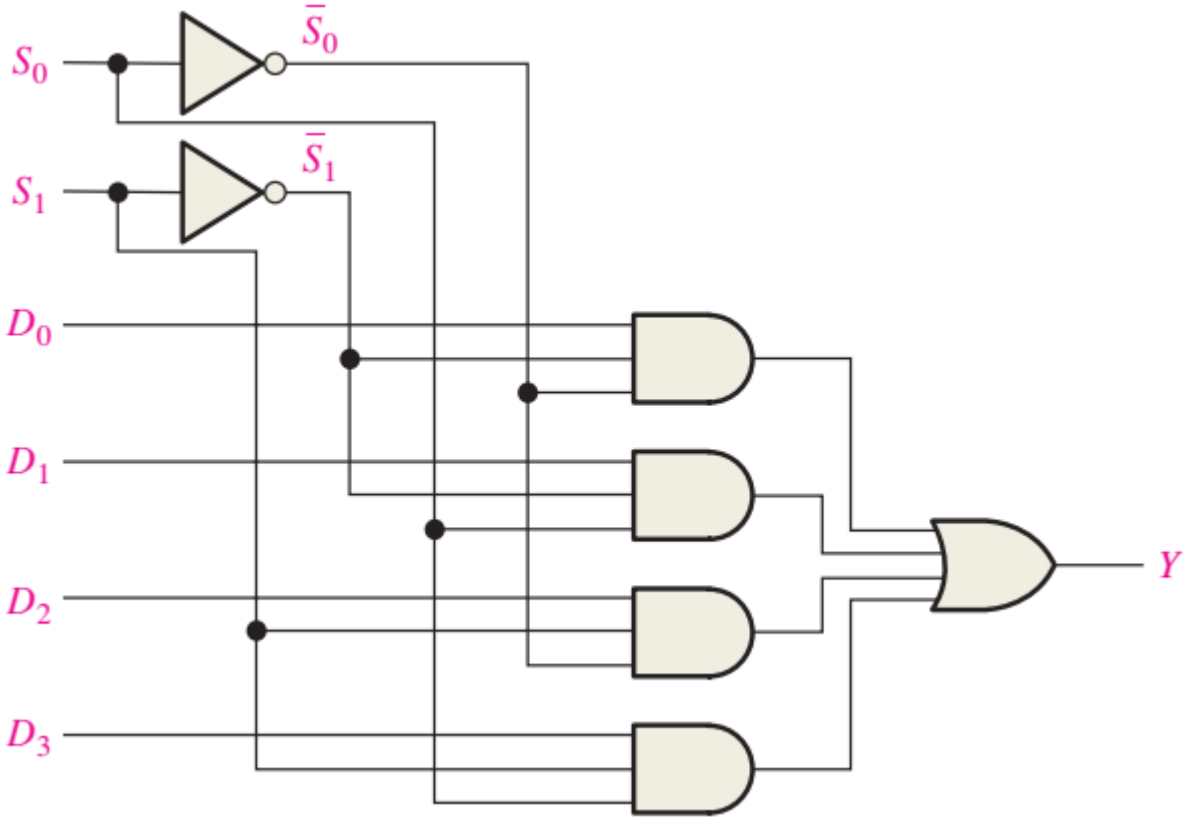
Inputs		Selector	Output
A	B	S	Y
1	X	0	A
X	1	1	B

$$Y_1 = A$$

$$Y_2 = B$$

4x1 Multiplexer





Truth Table for 4X1 Line Multiplexer

$Y_1 = \bar{A}$ $Y_2 = A$

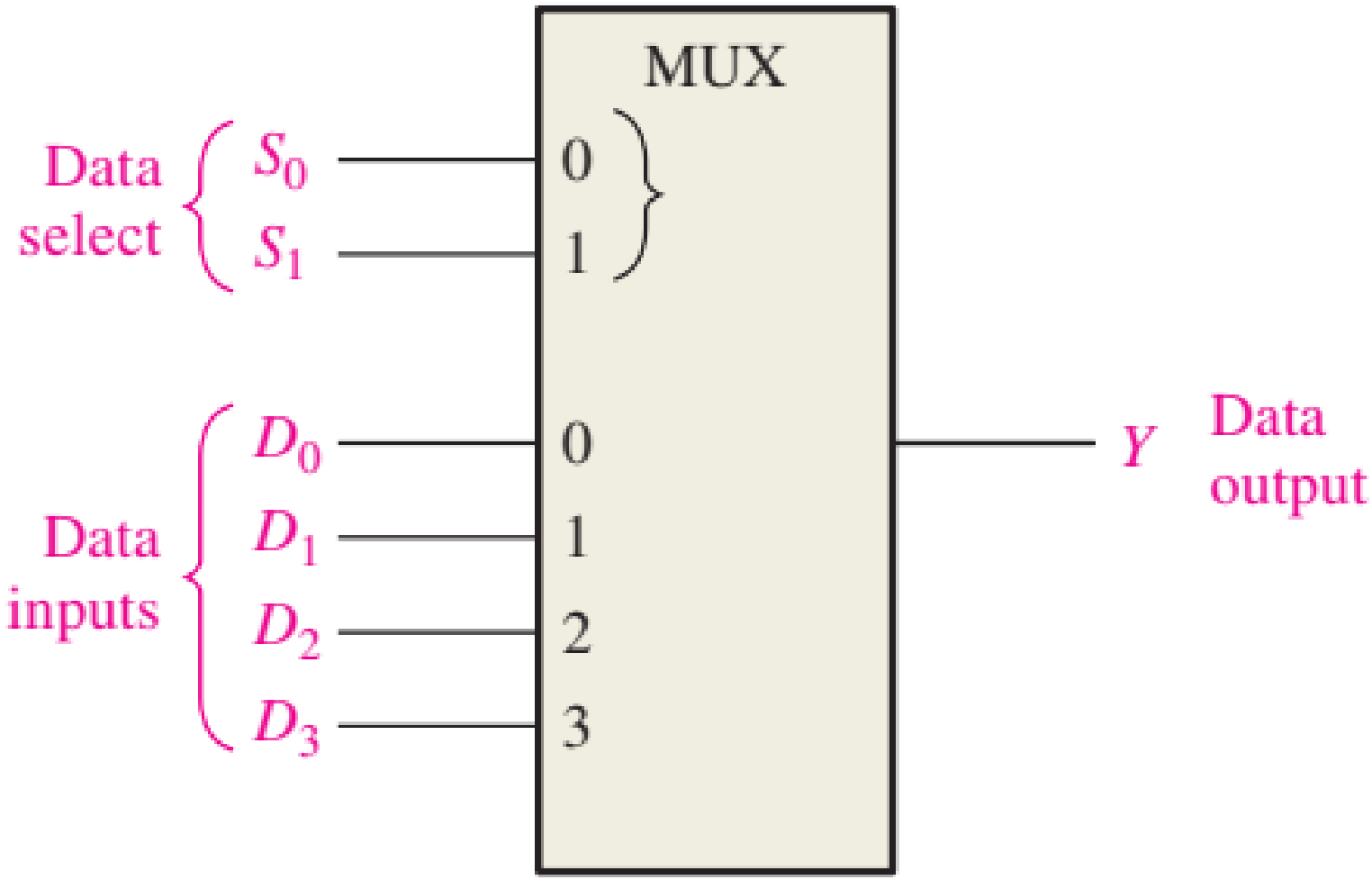
$Y_3 = \bar{B}$ $Y_4 = B$

$Y_5 = \bar{C}$ $Y_6 = C$

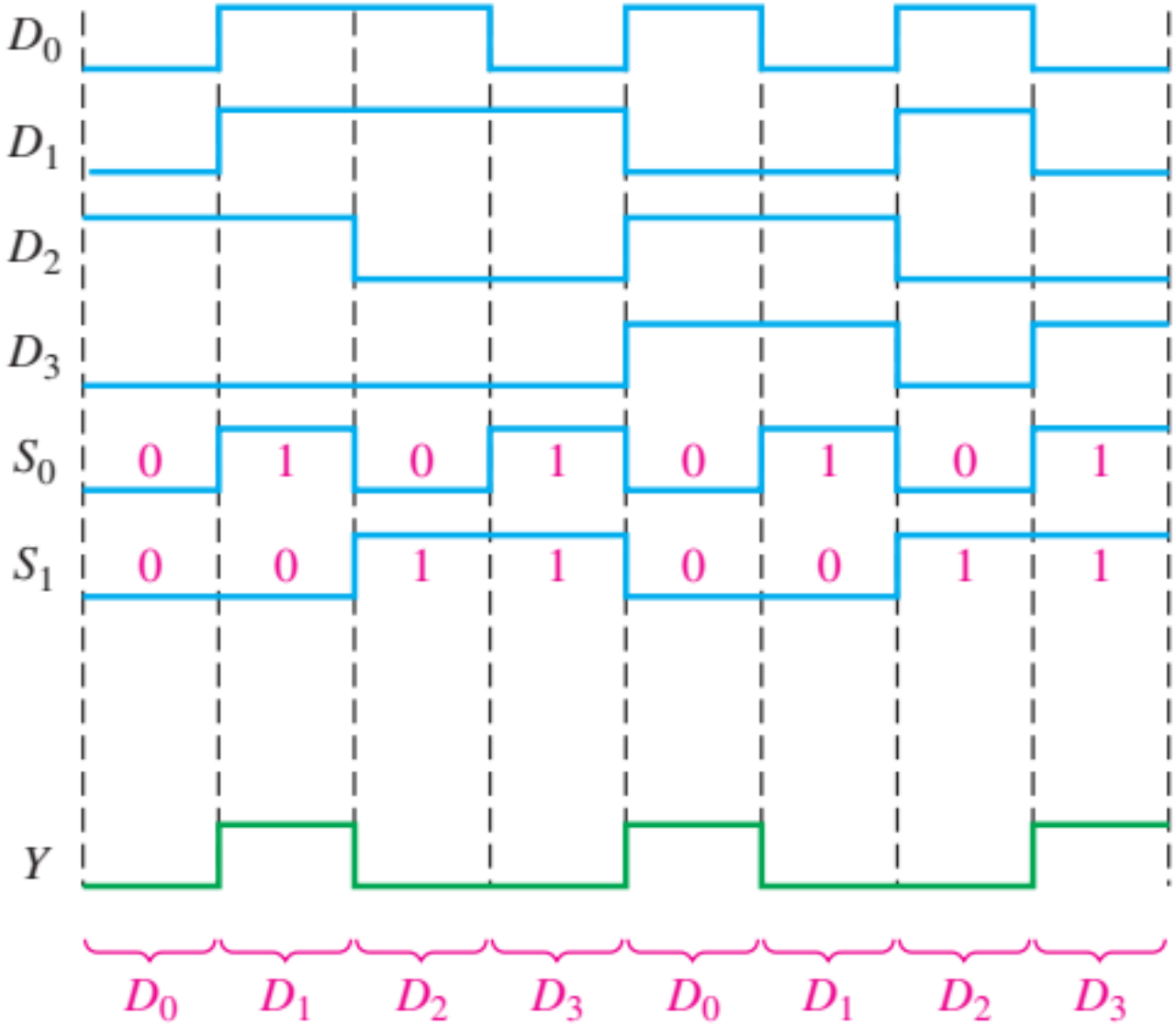
$Y_7 = \bar{D}$ $Y_8 = D$

Inputs				Selectors		Output
D0	D1	D2	D3	S0	S1	Z
0	X	X	X	0	0	$\bar{A}$
1	X	X	X	0	0	A
X	0	X	X	0	1	$\bar{B}$
X	1	X	X	0	1	B
X	X	0	X	1	0	$\bar{C}$
X	X	1	X	1	0	C
X	X	X	0	1	1	$\bar{D}$
X	X	X	1	1	1	D

4x1 Multiplexer Symbol

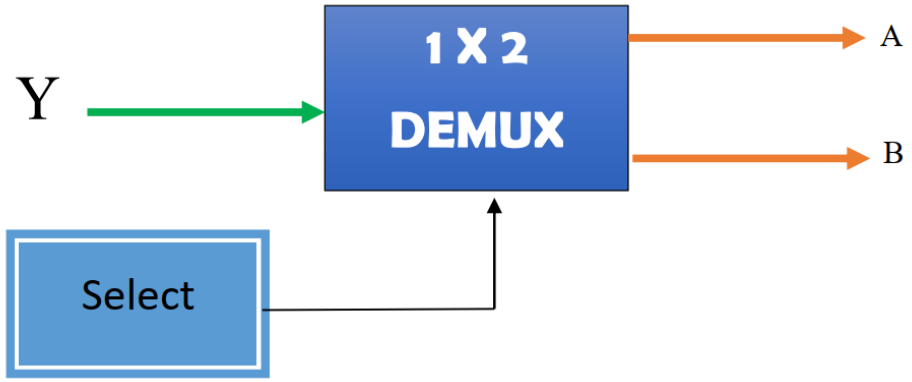


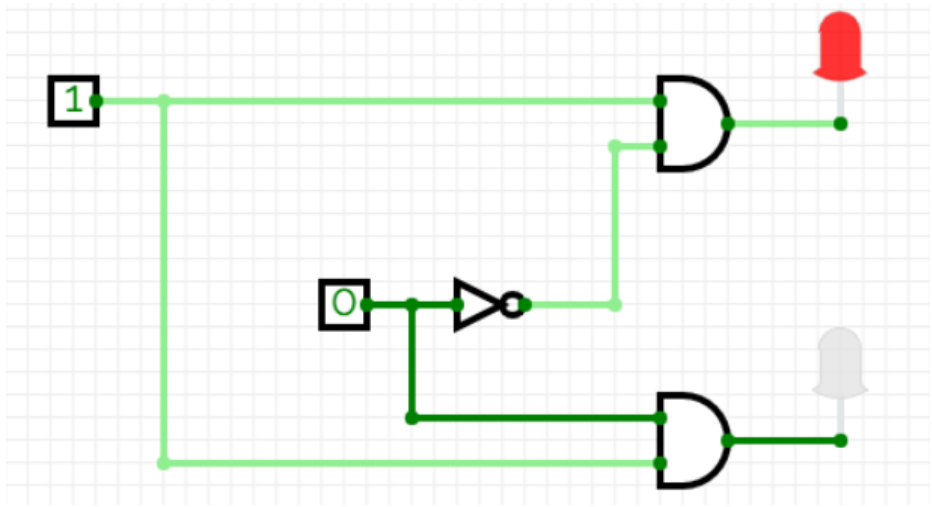
4x1 Multiplexer Timing Diagram

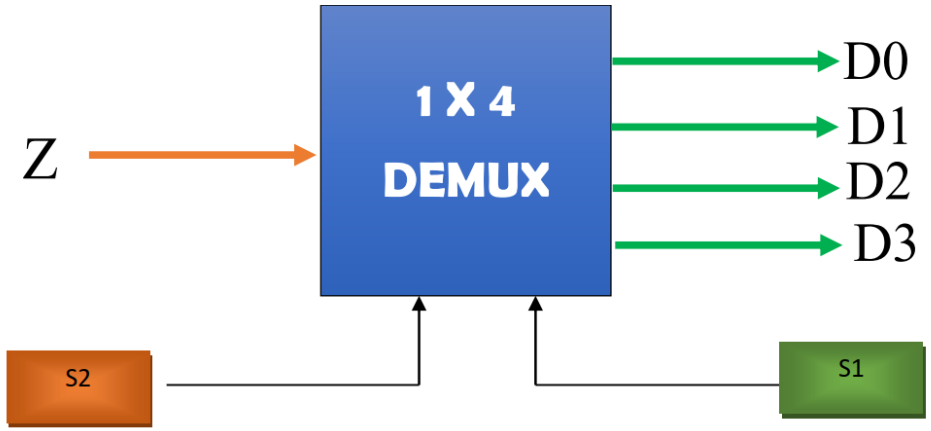


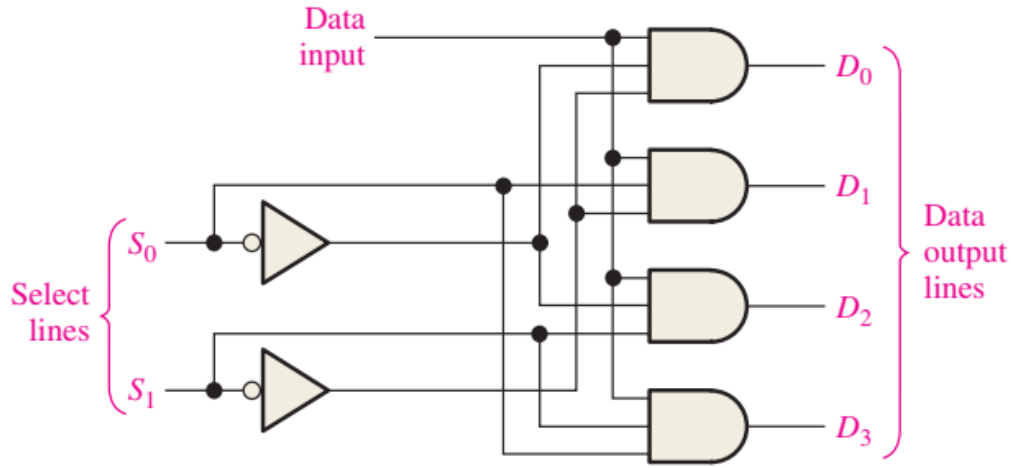
Demultiplexer

A **demultiplexer (DEMUX)** basically reverses the multiplexing function. It takes digital information from one line and distributes it to a given number of output lines (**data distributor).**









Truth Table for 1x4 Line Demultiplexer

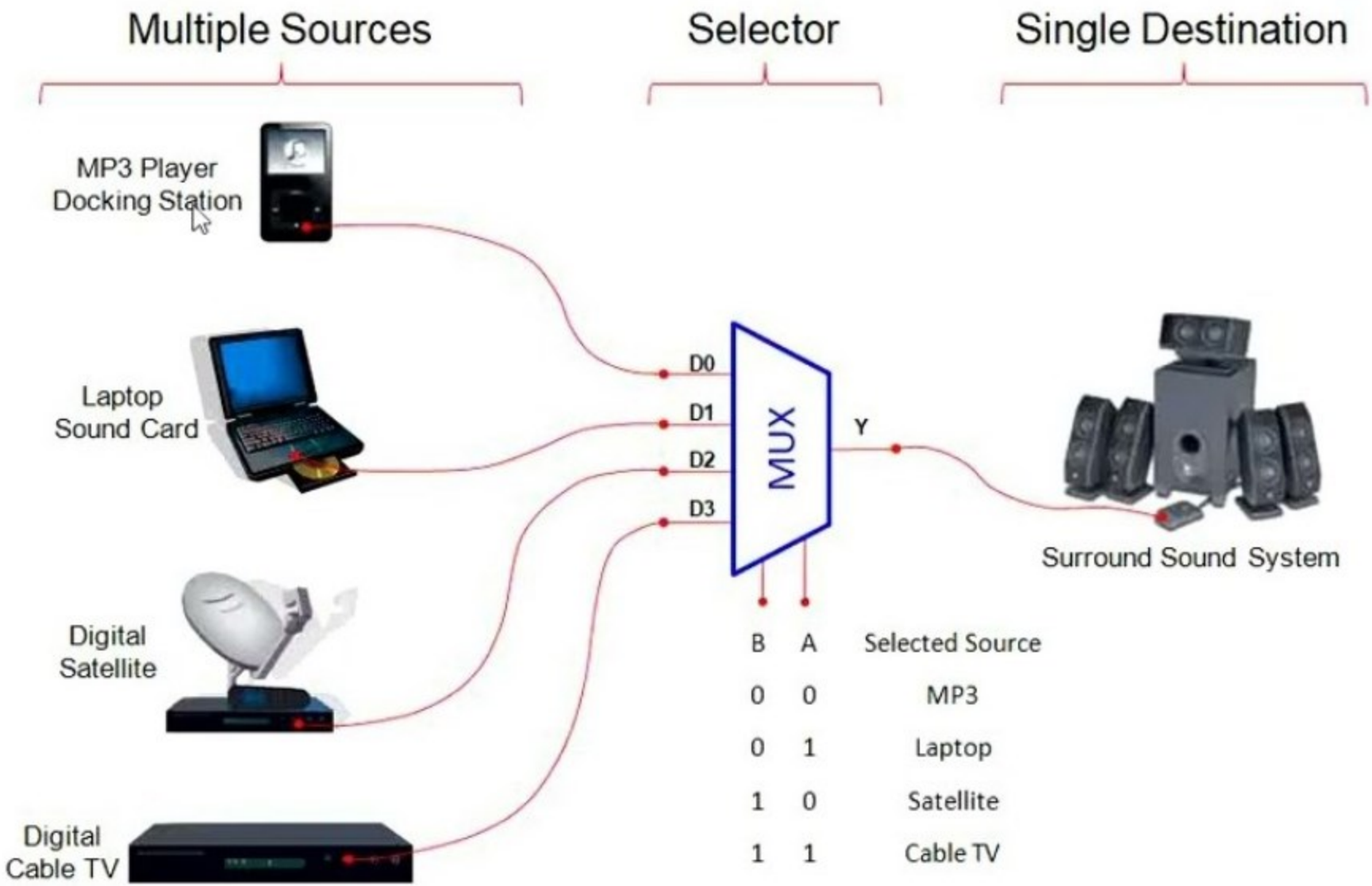
Inputs	Selector		Output			
ON	S0	S1	D0	D1	D2	D3
A	0	0	1	X	X	X
B	0	1	X	1	X	X
C	1	0	X	X	1	X
D	1	1	X	X	X	1

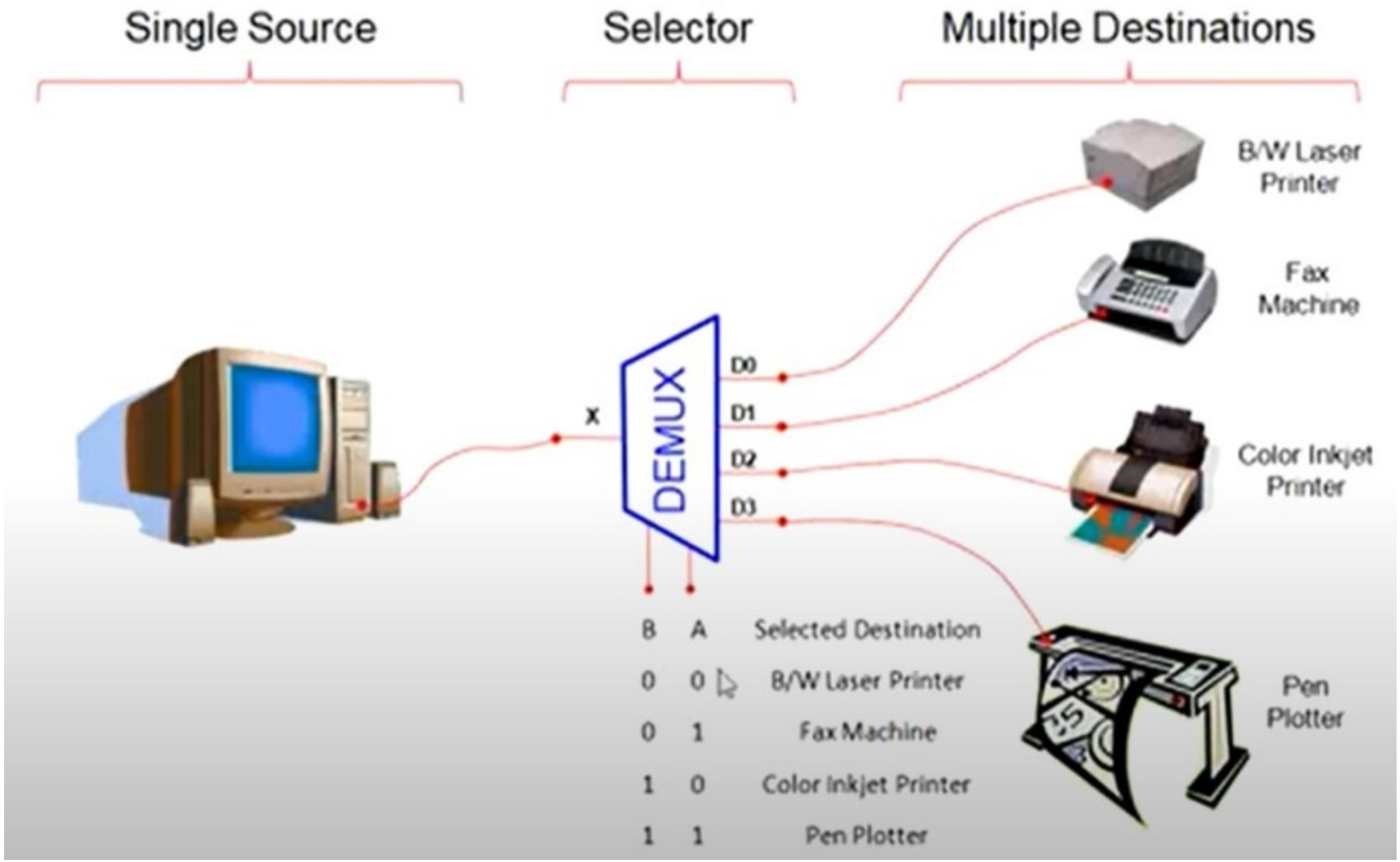
$$Y_1 = A$$

$$Y_2 = B$$

$$Y_3 = C$$

$$Y_4 = D$$


Multiple Sources
Selector
Single Destination
MP3 Player
Docking Station
Laptop
Sound Card
Digital
Satellite
Digital
Cable TV
D0
D1
D2
D3
MUX
Y
Surround Sound System
B A Selected Source
0 0 MP3
0 1 Laptop
1 0 Satellite
1 1 Cable TV


Single Source
Selector
Multiple Destinations
B/W Laser Printer
Fax Machine
Color Inkjet Printer
Pen Plotter
DEMUX
x
D0
D1
D2
D3
B A Selected Destination
0 0 B/W Laser Printer
0 1 Fax Machine
1 0 Color Inkjet Printer
1 1 Pen Plotter

Home Work

1	Design a 1-to-32 line demultiplexer Using Both 1-to-4 and 1-to-8 line demultiplexers	10 M
2	Explain the operation of Fig 1H, next slide	6 M

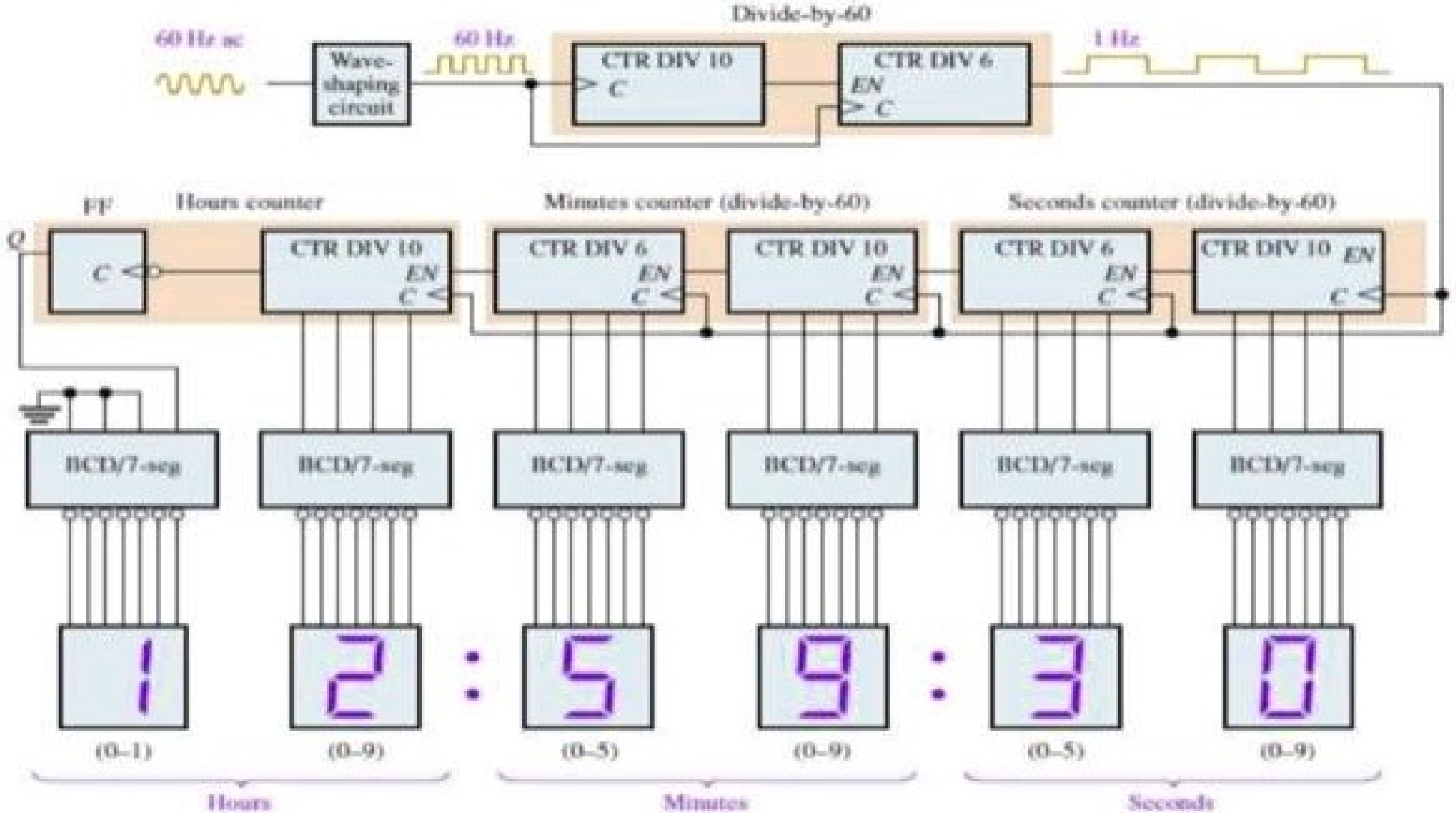

Divide-by-60
60 Hz ac
Wave-shaping circuit
60 Hz
CTR DIV 10
CTR DIV 6
1 Hz
FF
Hours counter
Minutes counter (divide-by-60)
Seconds counter (divide-by-60)
CTR DIV 10
CTR DIV 6
CTR DIV 10
CTR DIV 6
CTR DIV 10
BCD/7-seg
BCD/7-seg
BCD/7-seg
BCD/7-seg
BCD/7-seg
BCD/7-seg
1
2
5
9
3
0
(0–1)
(0–9)
(0–5)
(0–9)
(0–5)
(0–9)
Hours
Minutes
Seconds

Fig. 1-H

University of Technology
Laser & Optoelectronics Engineering Department

Digital Electronics

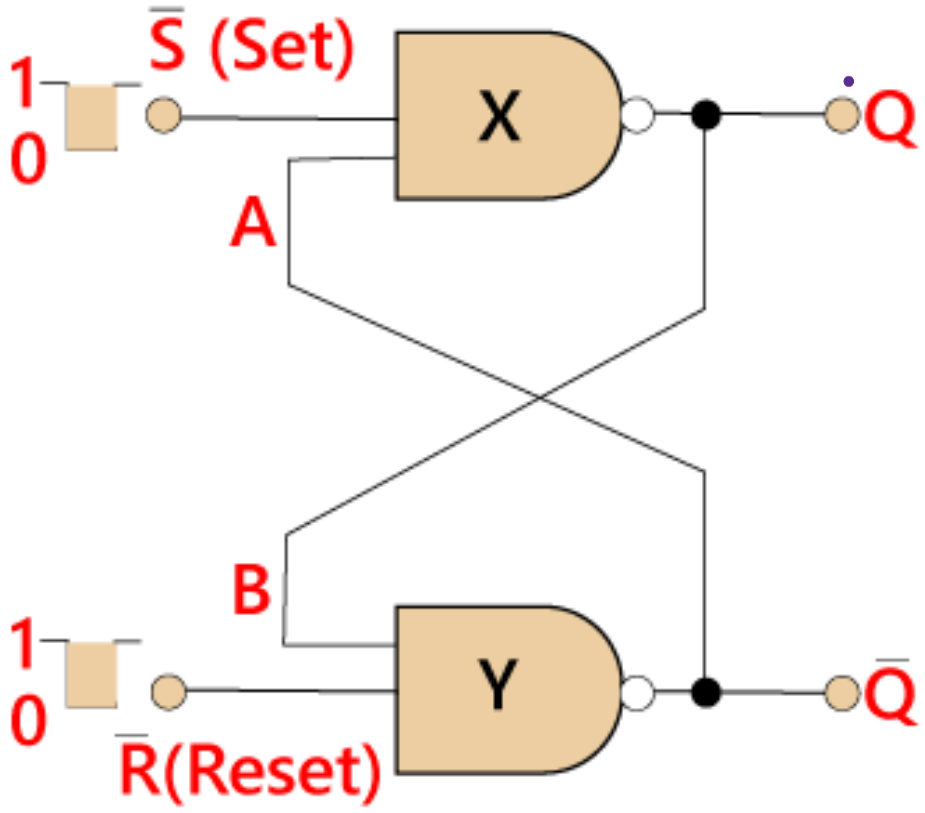
Lec. Dr. Taif Alawsi

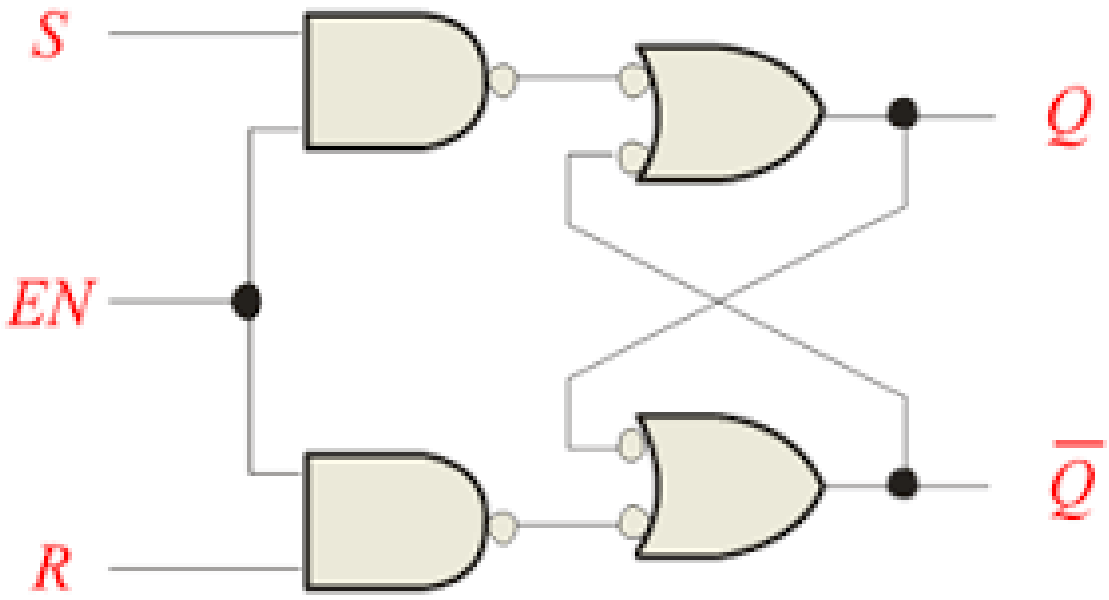
Lec. 8: Sequential Circuits: 2024-Nov-17

Lecture Outline

1. Flip-Flops
2. Counters
3. 555 Timer
4. Oscillator
5. HW

SR & GATED SR

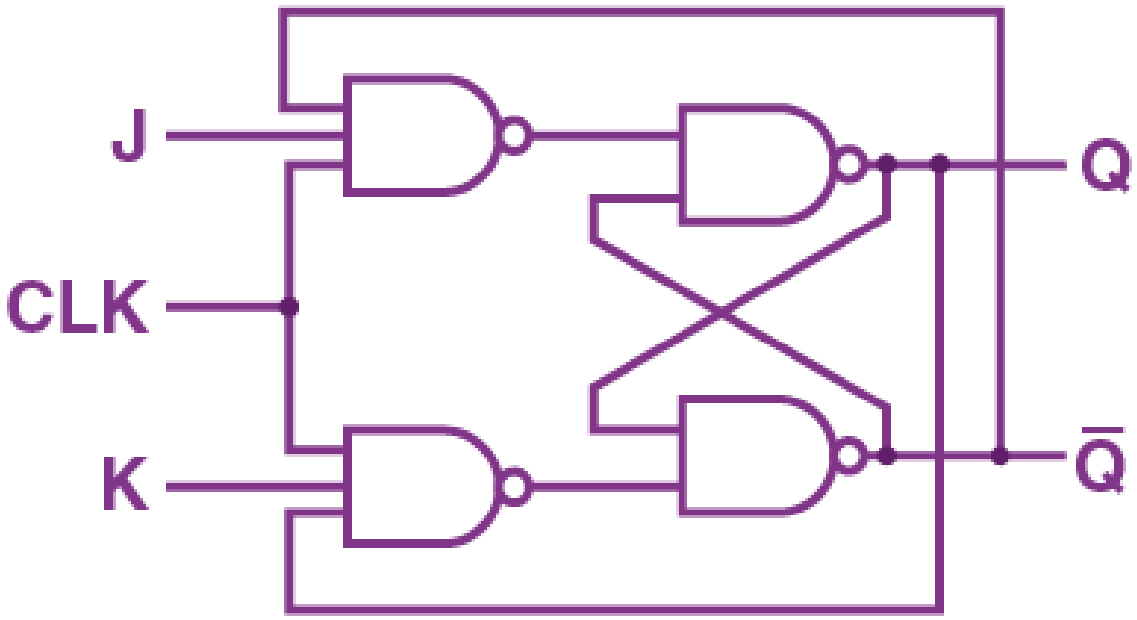




State Table

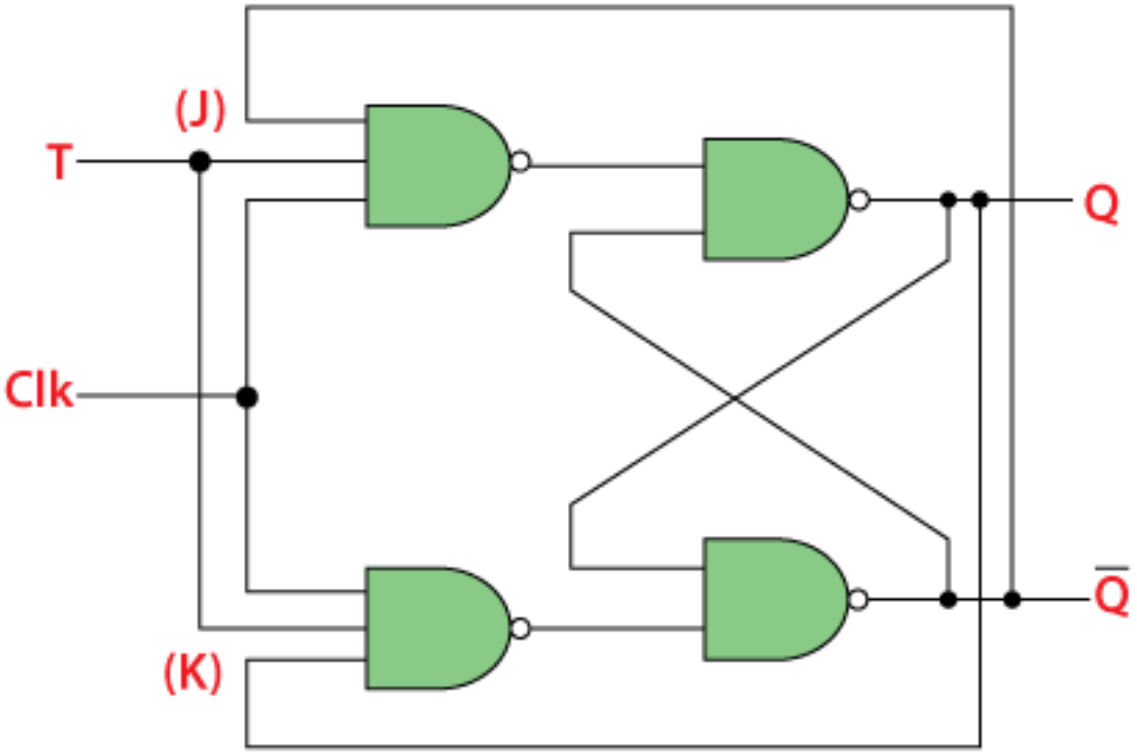
State	S	R	Q	Q'	Description
Set	1	0	0	1	Set Q'>>1
	1	1	0	1	No change
Reset	0	1	1	0	Reset Q'>>0
	1	1	1	0	No change
Invalid	0	0	1	1	Invalid Condition

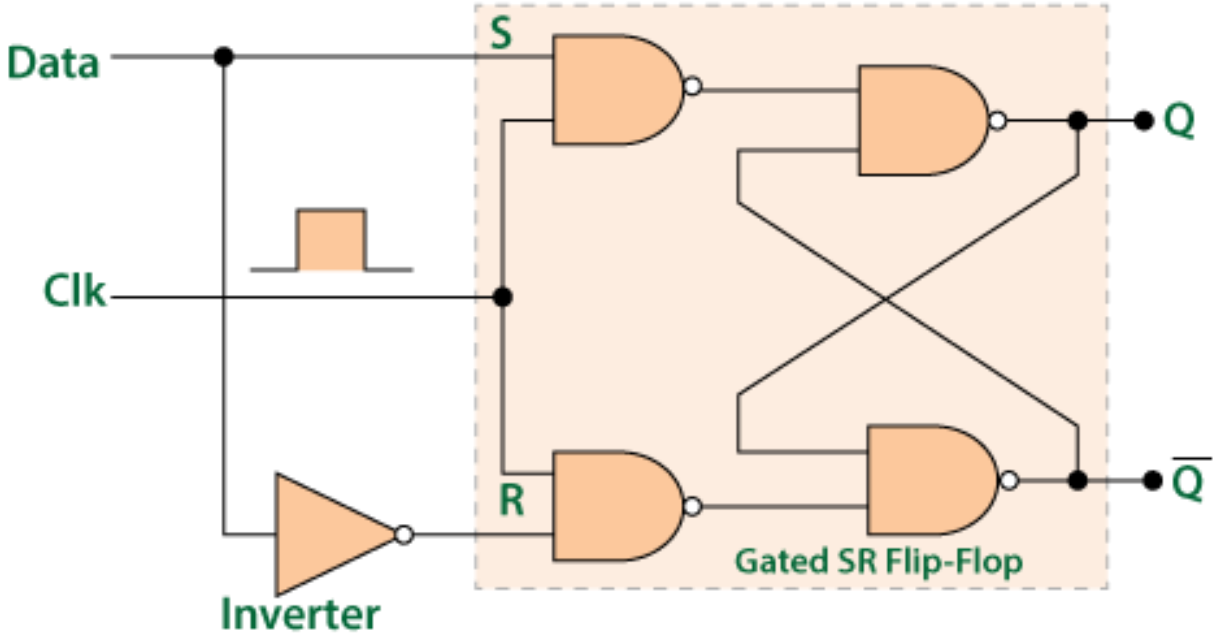
JK FF



Inputs			Outputs		
J	K	CLK	Q	$\bar{Q}$	Comments
0	0	↑	Q_0	$\bar{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	$\bar{Q}_0$	Q_0	Toggle

T & D FF


(J)
T
Clk
(K)
Q
$\overline{Q}$


Data
Clk
Inverter
S
R
Q
$\overline{Q}$
Gated SR Flip-Flop

T & D FF

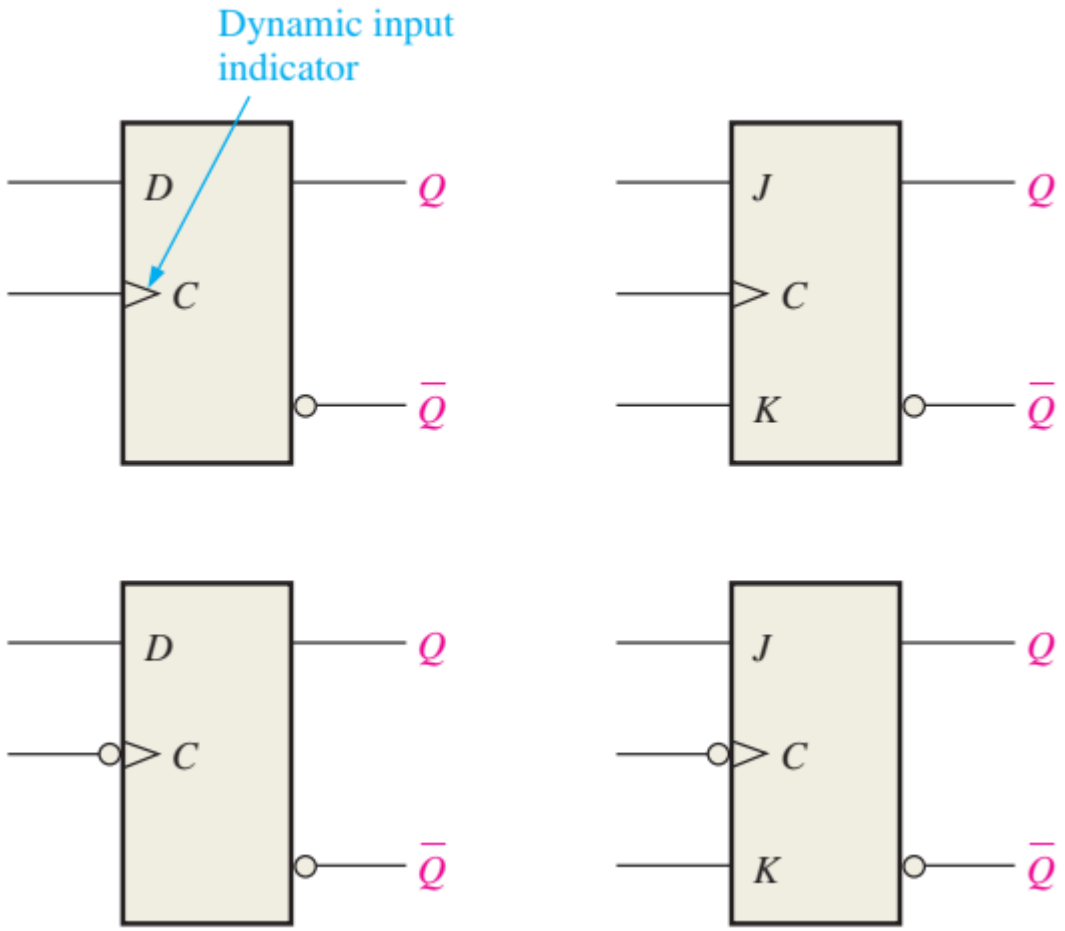
Truth table			
CLK	T	Q_{next}	Comment
Rising edge	0	Q	Hold state
Falling edge	0	Q	Hold state
Rising edge	1	$\overline{Q}$	Toggle
Falling edge	1	Q	No change

Q_{next} - "after the clock transition" output

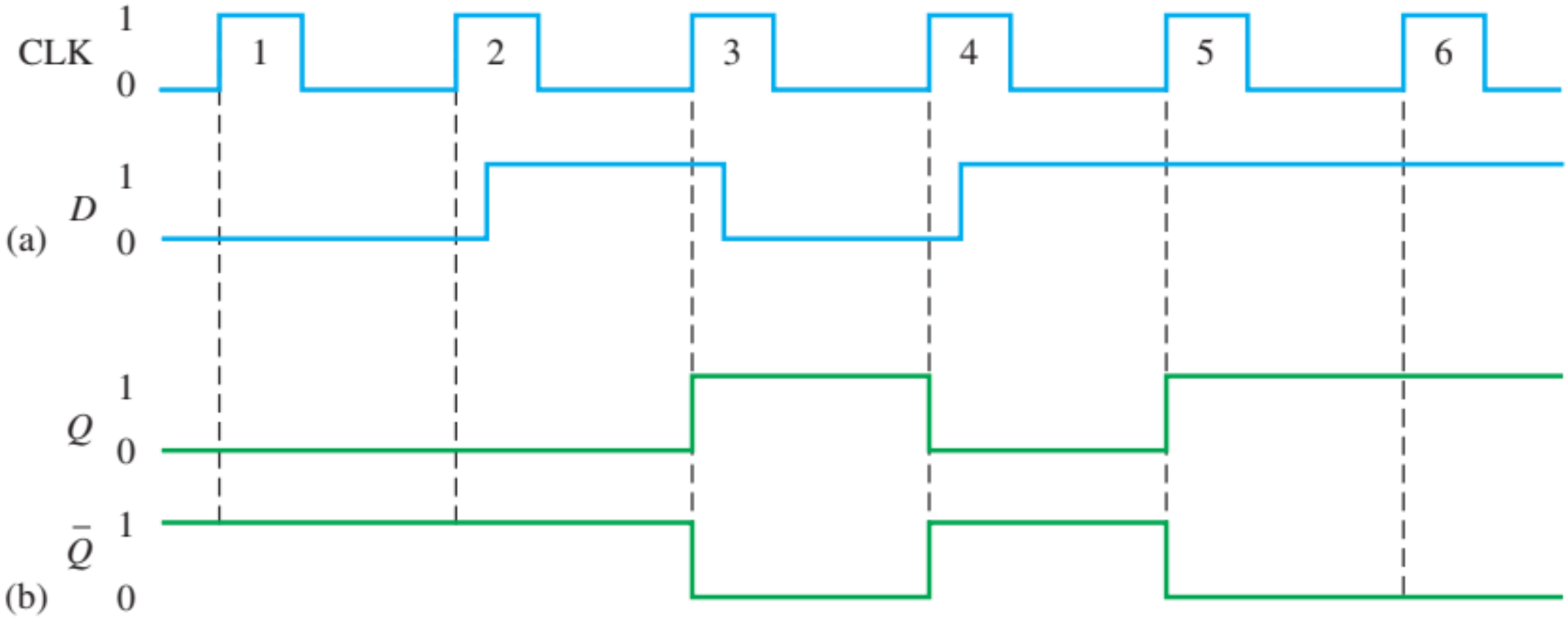
Q - the current output

Inputs		Outputs		
D	CLK	*Q*	$\overline{Q}$	Comments
1	↑	1	0	SET
0	↑	0	1	RESET

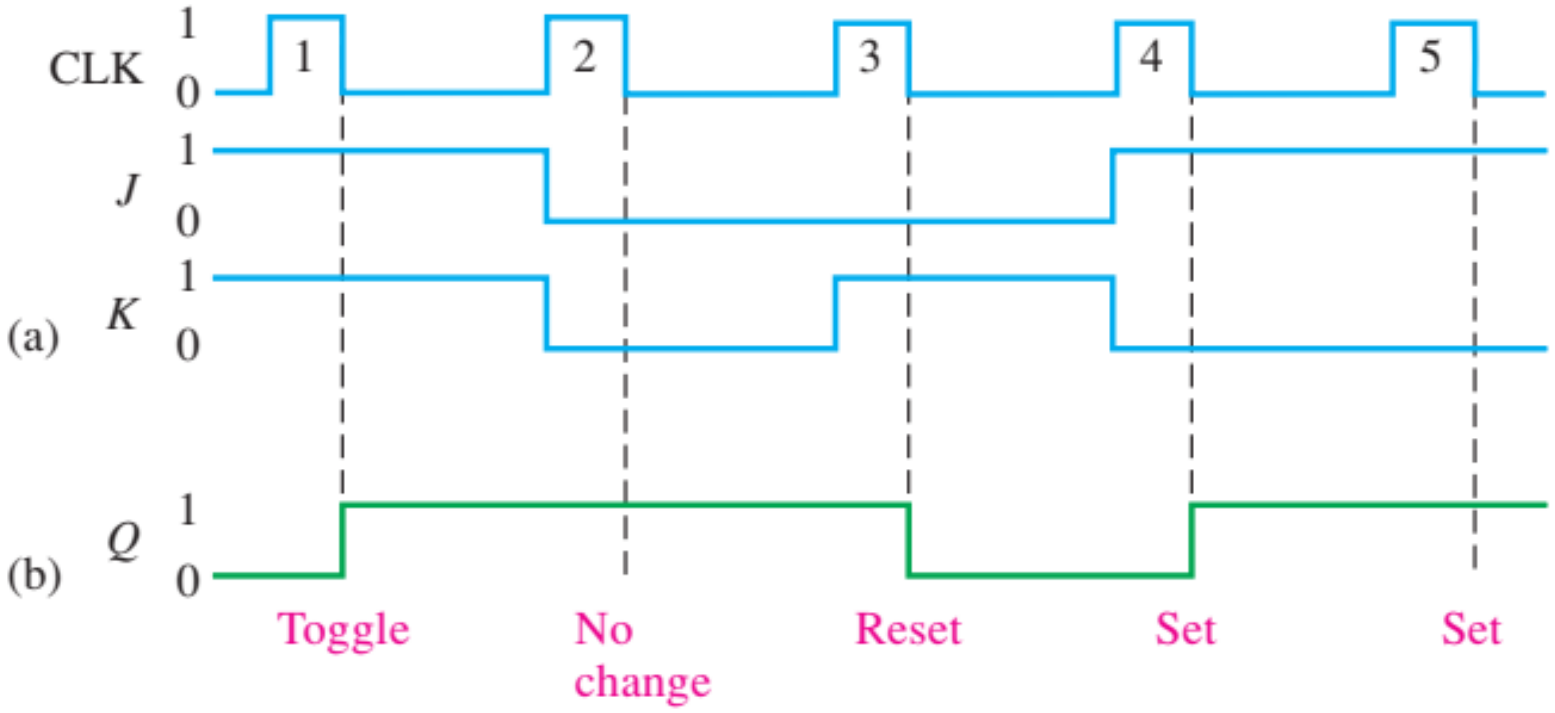
FF (+VE & -VE Edge)

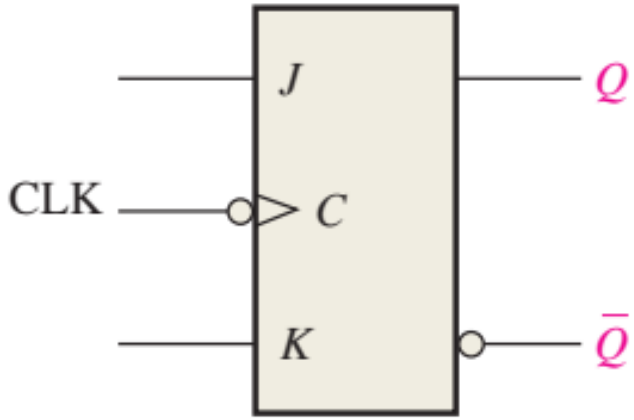

Dynamic input indicator
D
C
Q
$\overline{Q}$
J
C
K
Q
$\overline{Q}$
D
C
Q
$\overline{Q}$
J
C
K
Q
$\overline{Q}$

Timing Diagram

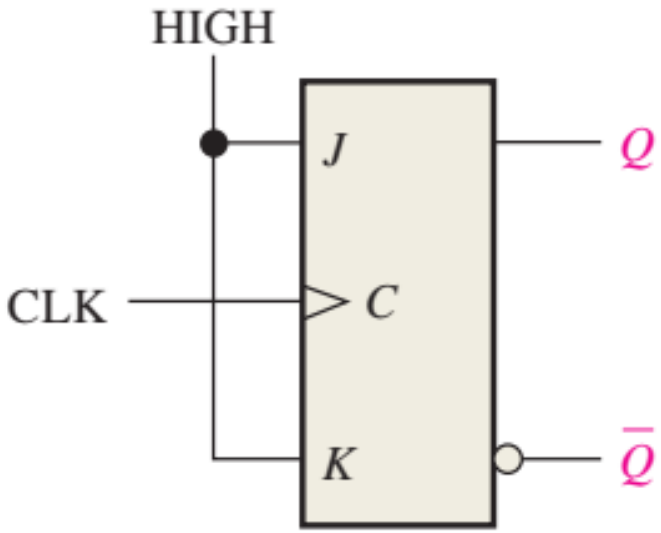


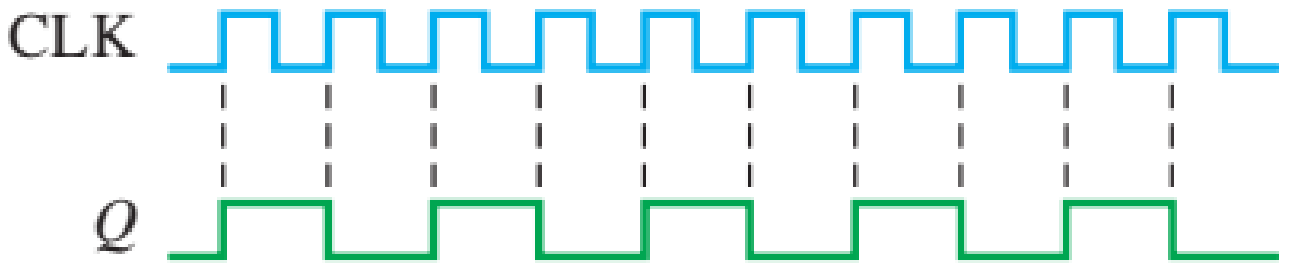
-ve edge JK FF Operation



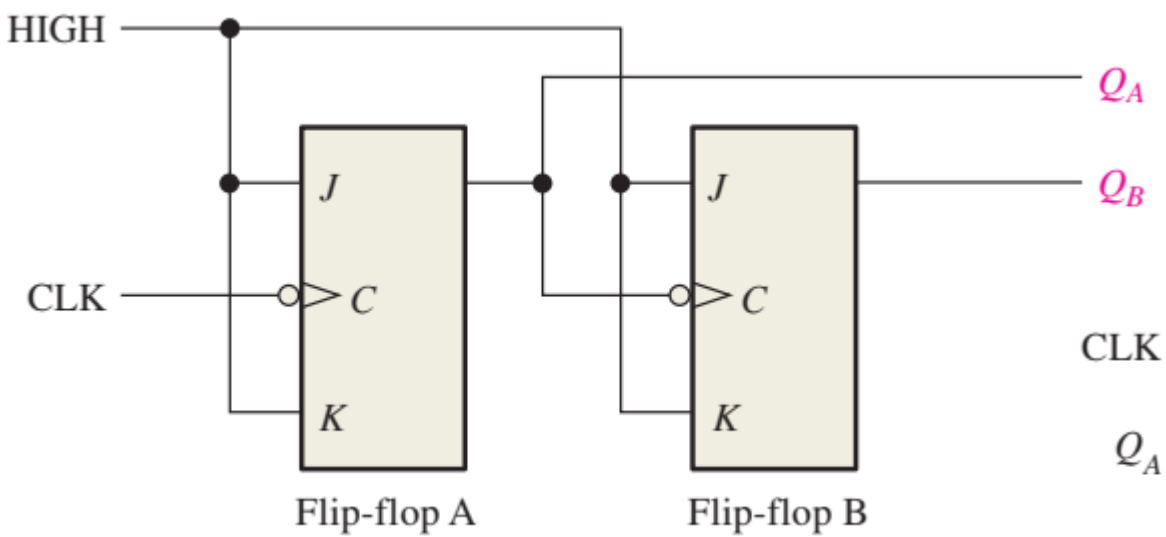


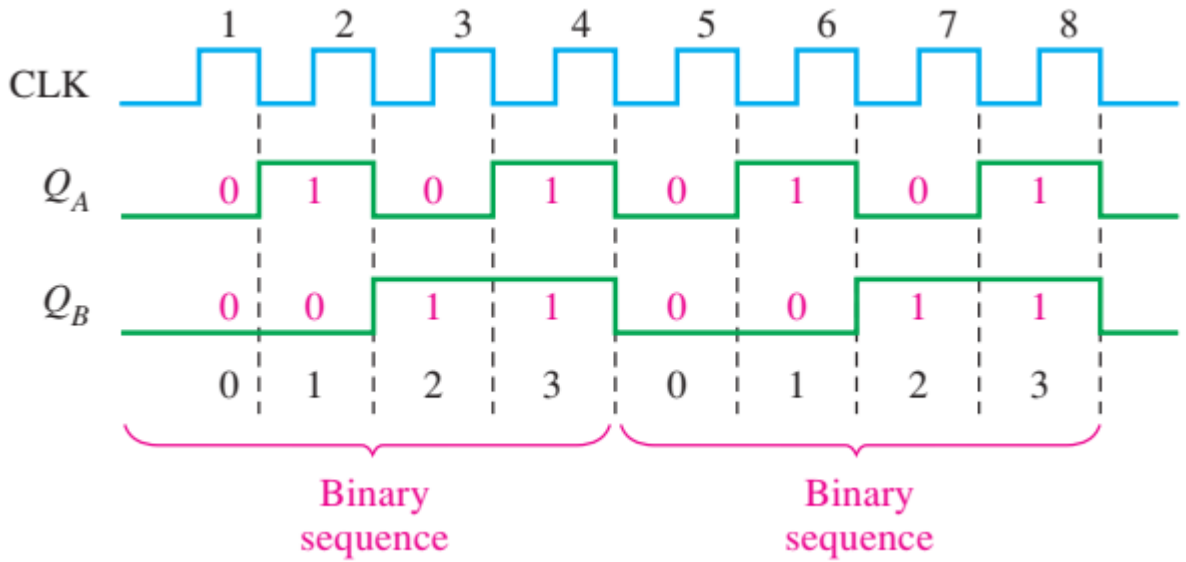
Frequency Doubler



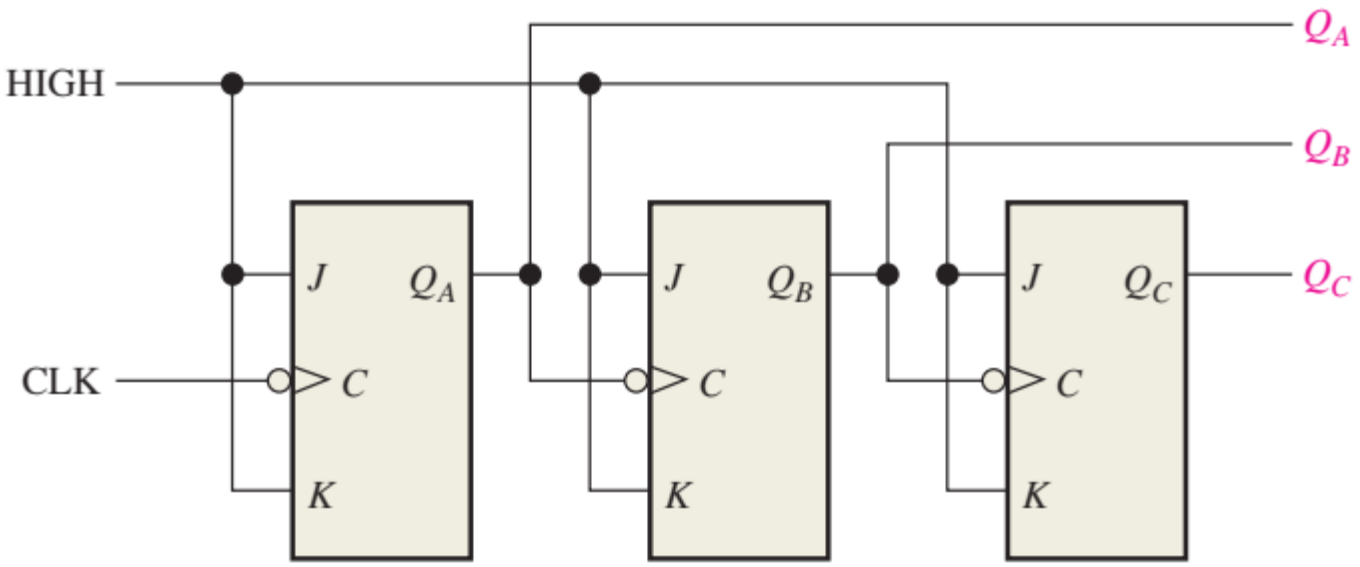


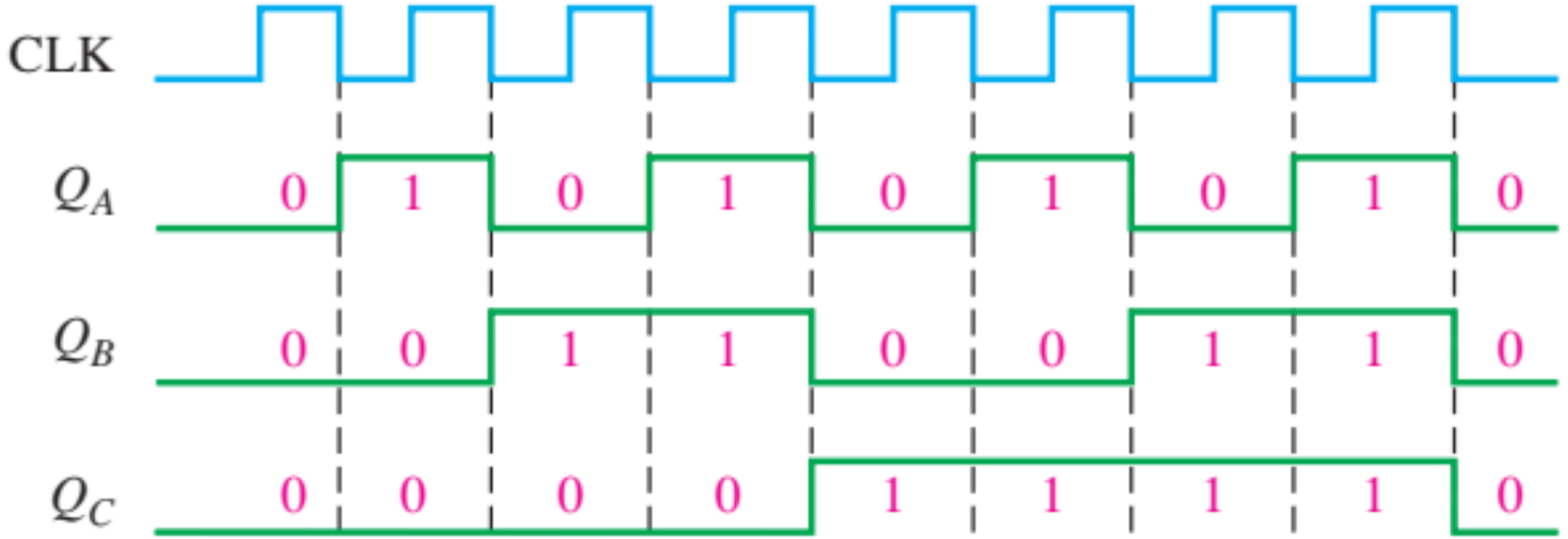
2-bit Binary Counter



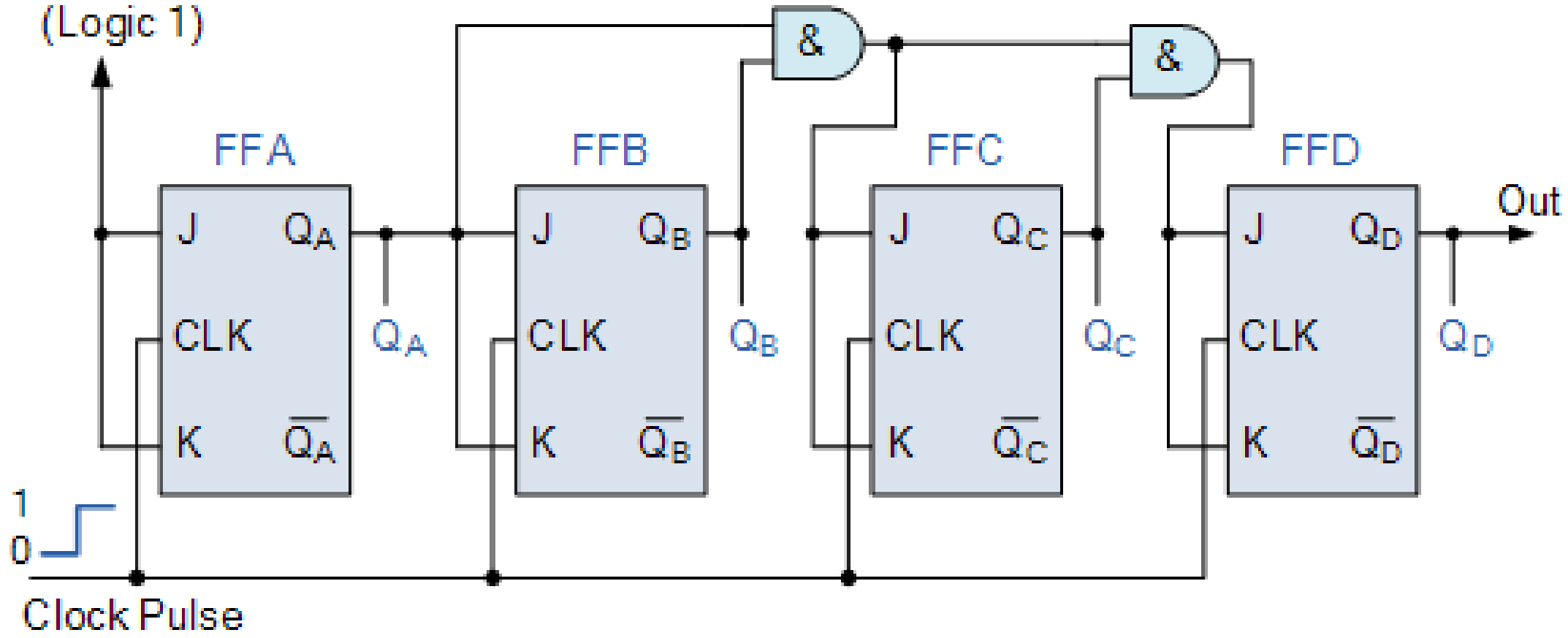


3-bit Binary Counter

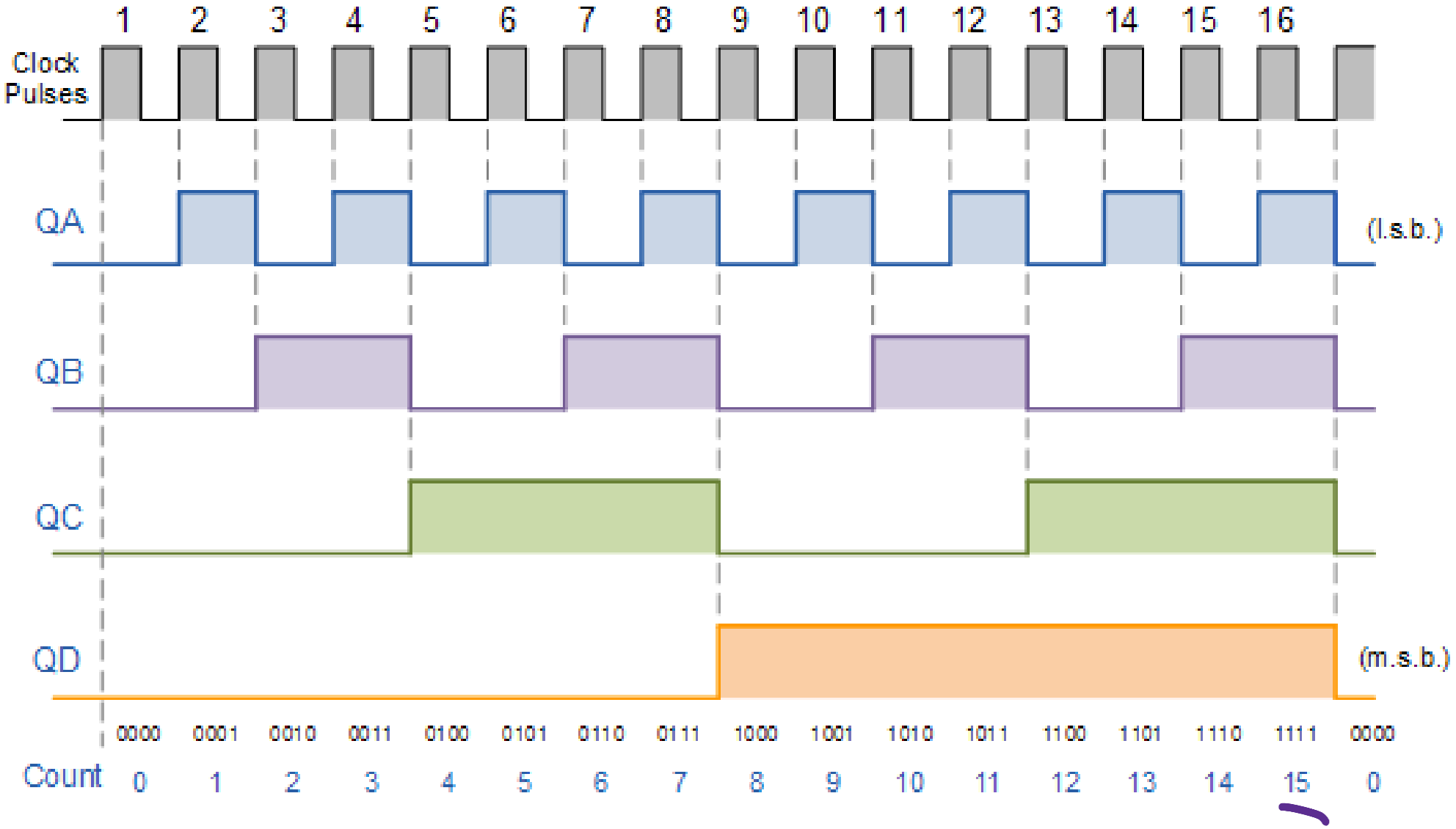




Four Stage Synchronous Binary Counter



Four Stage Synchronous Binary Counter



States of a BCD decade counter.

Clock Pulse	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (recycles)	0	0	0	0

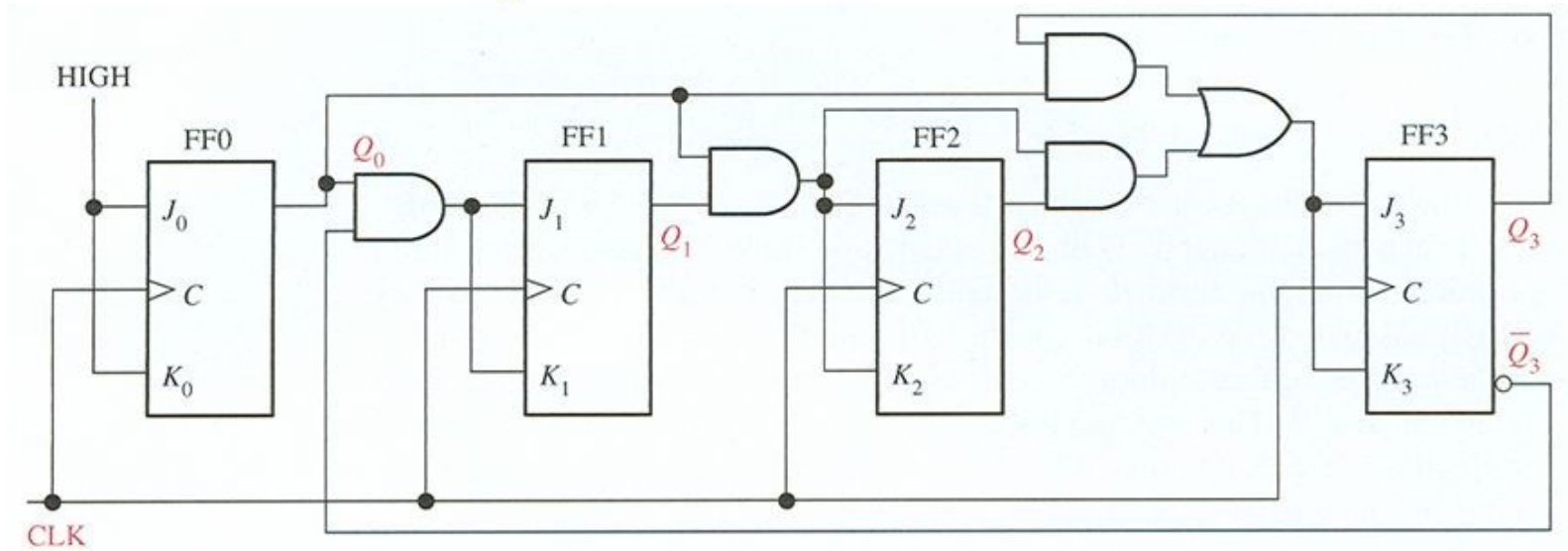
$$J_0 = K_0 = 1$$

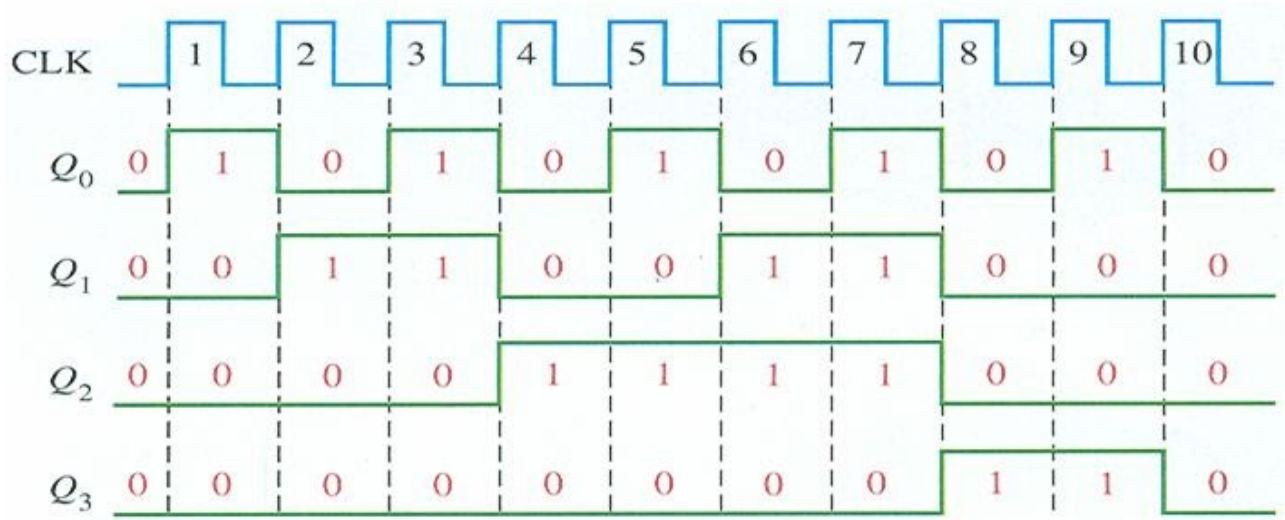
$$J_1 = K_1 = Q_0\overline{Q}_3$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$

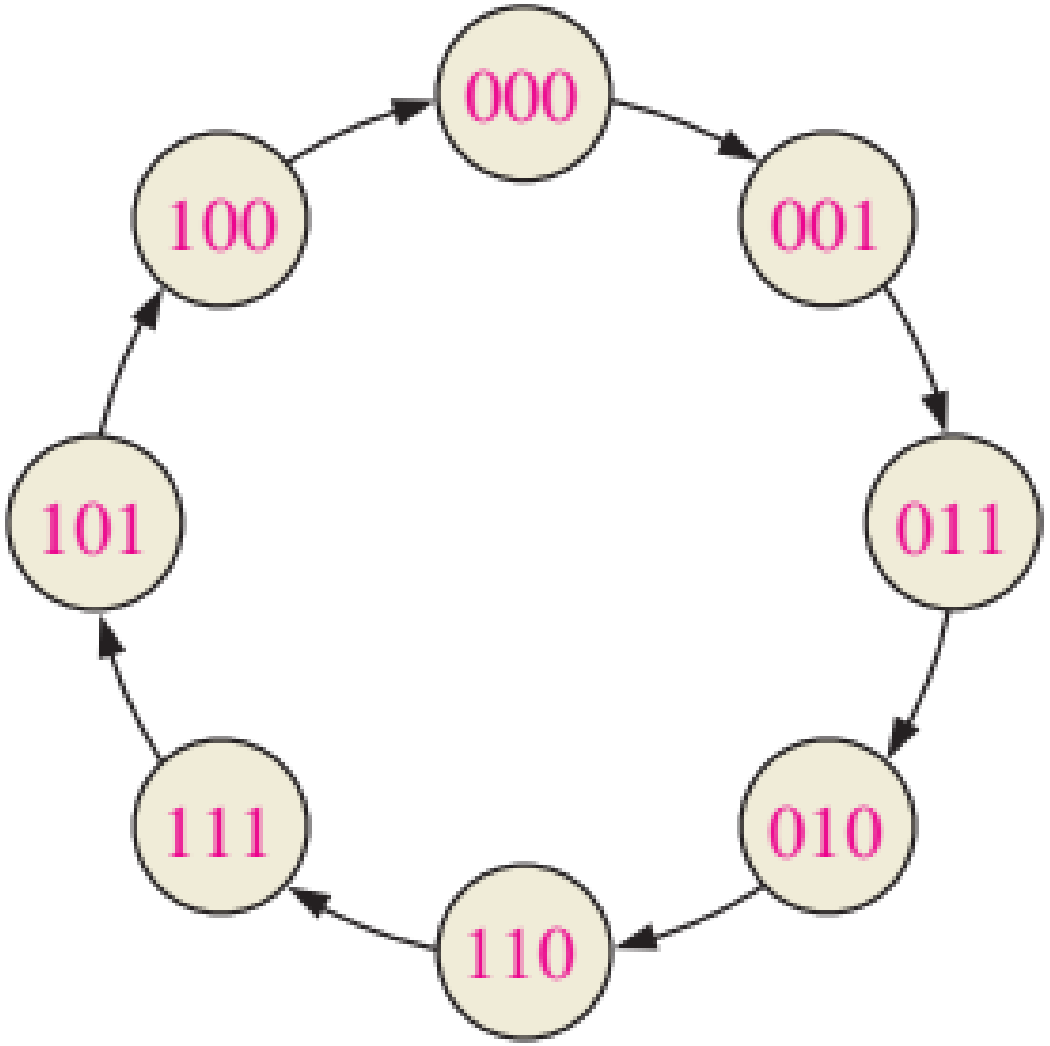
A 4 bit Synchronous Decade Counter





Counters Design Steps

STEP 1
Given in the Question



STEP 2: Create the State Table

Present State			Next State		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

STEP 3: Each FF Has its own Transition Table

Transition table for a J-K flip-flop.

Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

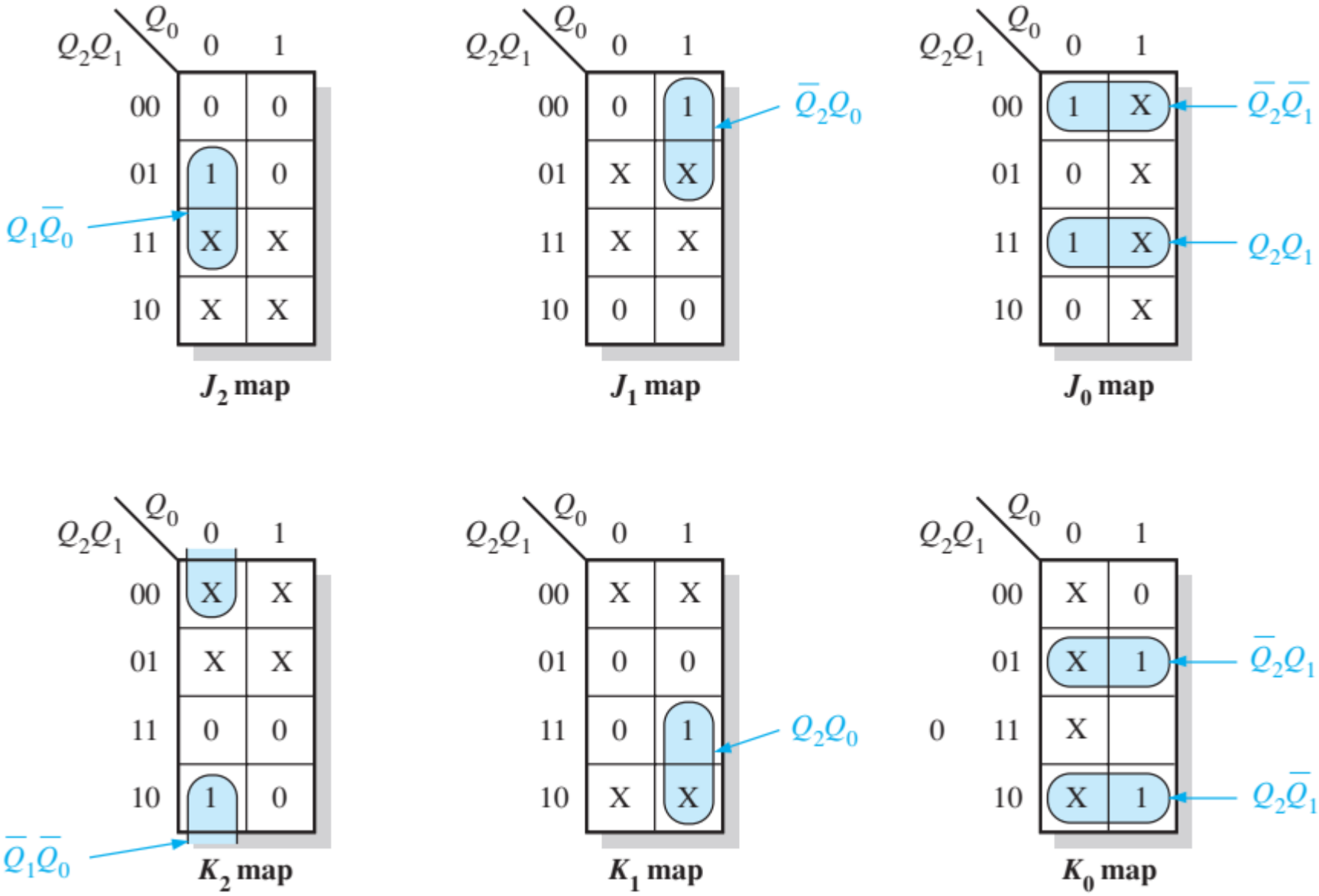
Q_N: present state
Q_{N+1}: next state
X: "don't care"

We will use only JK FF
Memorize this table

STEP 4: Use the Transition Table to find the output

Outputs					
J2	**K2**	**J1**	**K1**	**J0**	**K0**
0	X	0	X	1	X
0	X	1	X	X	0
0	X	X	0	X	1
1	X	X	0	0	X
X	0	X	0	1	X
X	0	X	1	X	0
X	0	0	X	X	1
X	1	0	X	0	X

STEP 5: Map the Outputs into K-Map



STEP 6: Find the Equation Using K-Map Minimization Techniques

$$J_0 = Q_2Q_1 + \overline{Q}_2\overline{Q}_1 = \overline{Q_2 \oplus Q_1}$$

$$K_0 = Q_2\overline{Q}_1 + \overline{Q}_2Q_1 = Q_2 \oplus Q_1$$

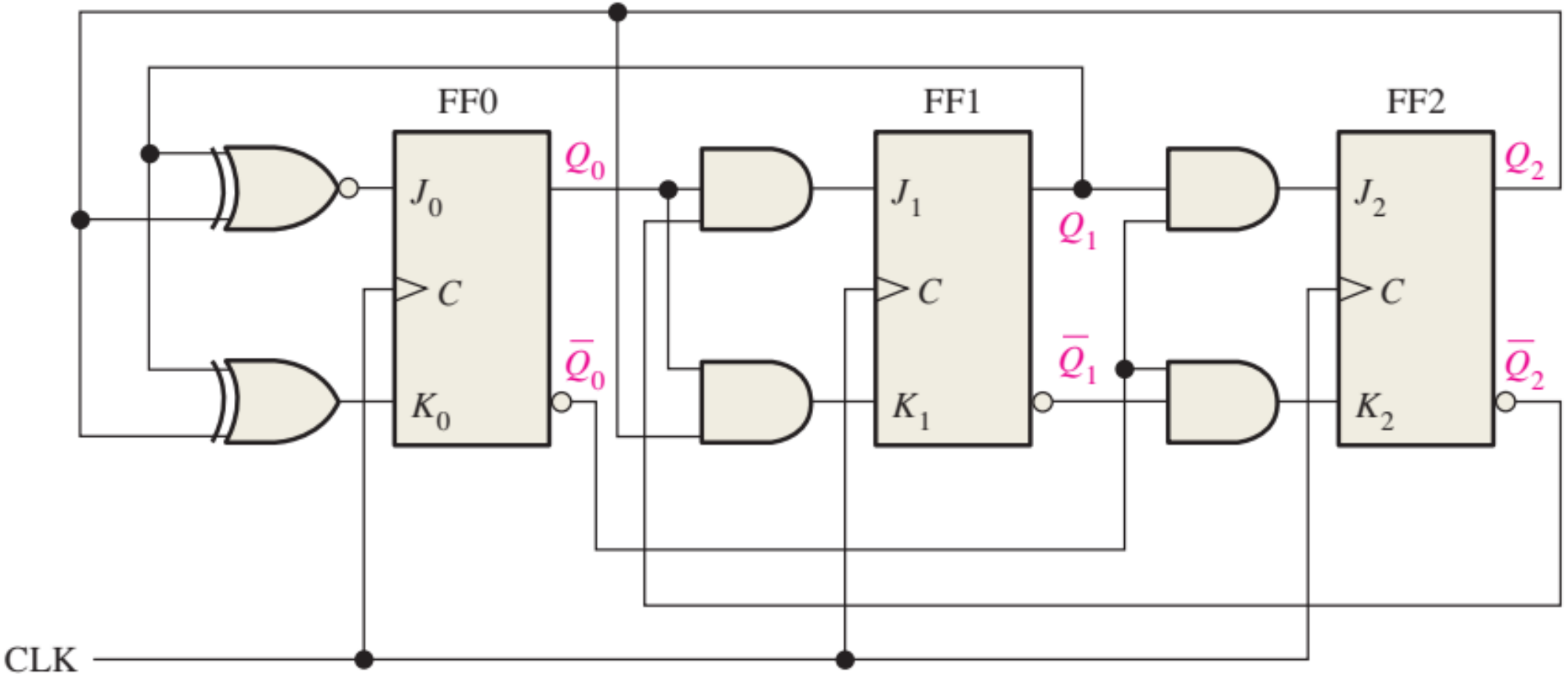
$$J_1 = \overline{Q}_2Q_0$$

$$K_1 = Q_2Q_0$$

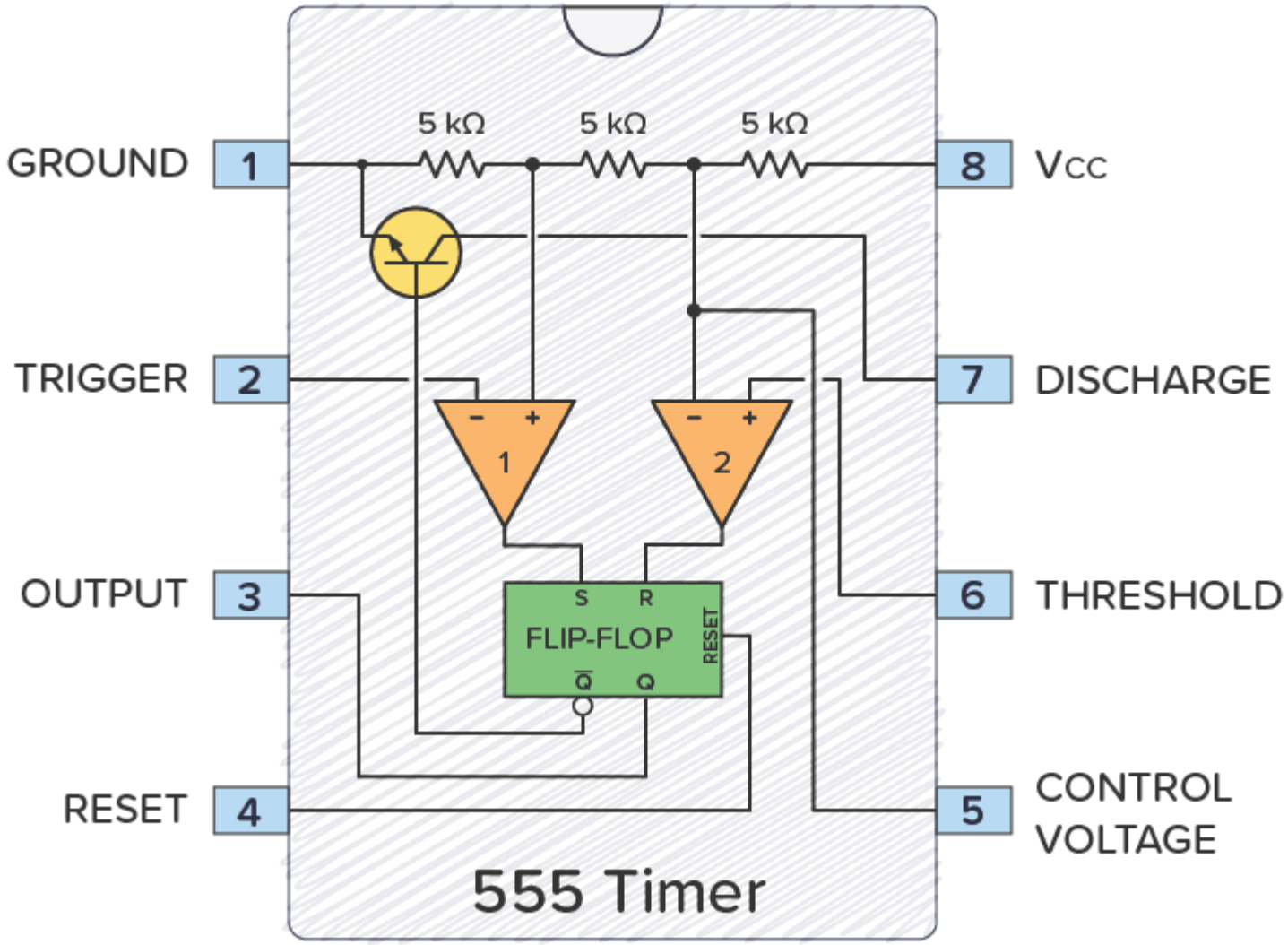
$$J_2 = Q_1\overline{Q}_0$$

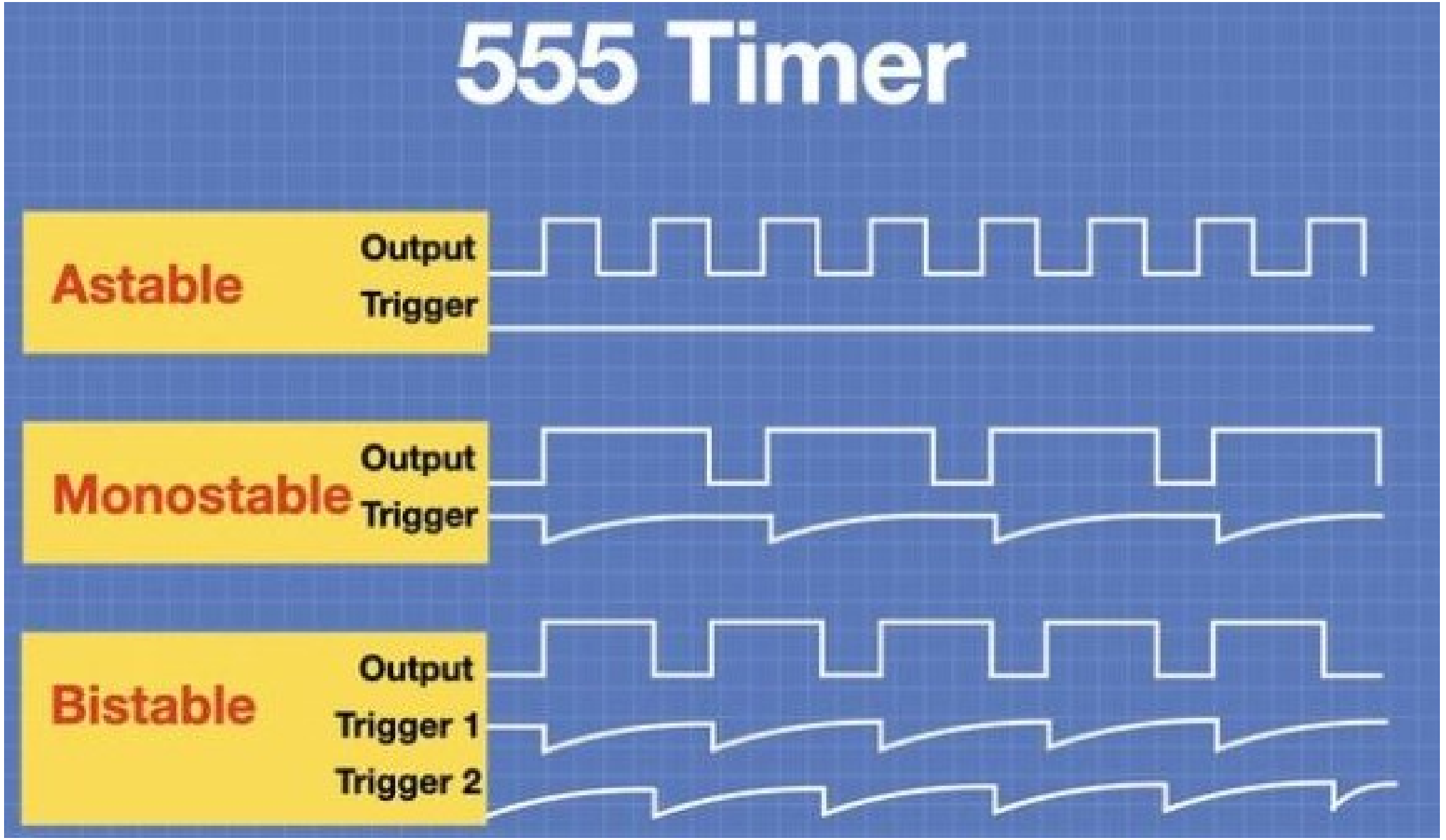
$$K_2 = \overline{Q}_1\overline{Q}_0$$

STEP 7: Build the Counter

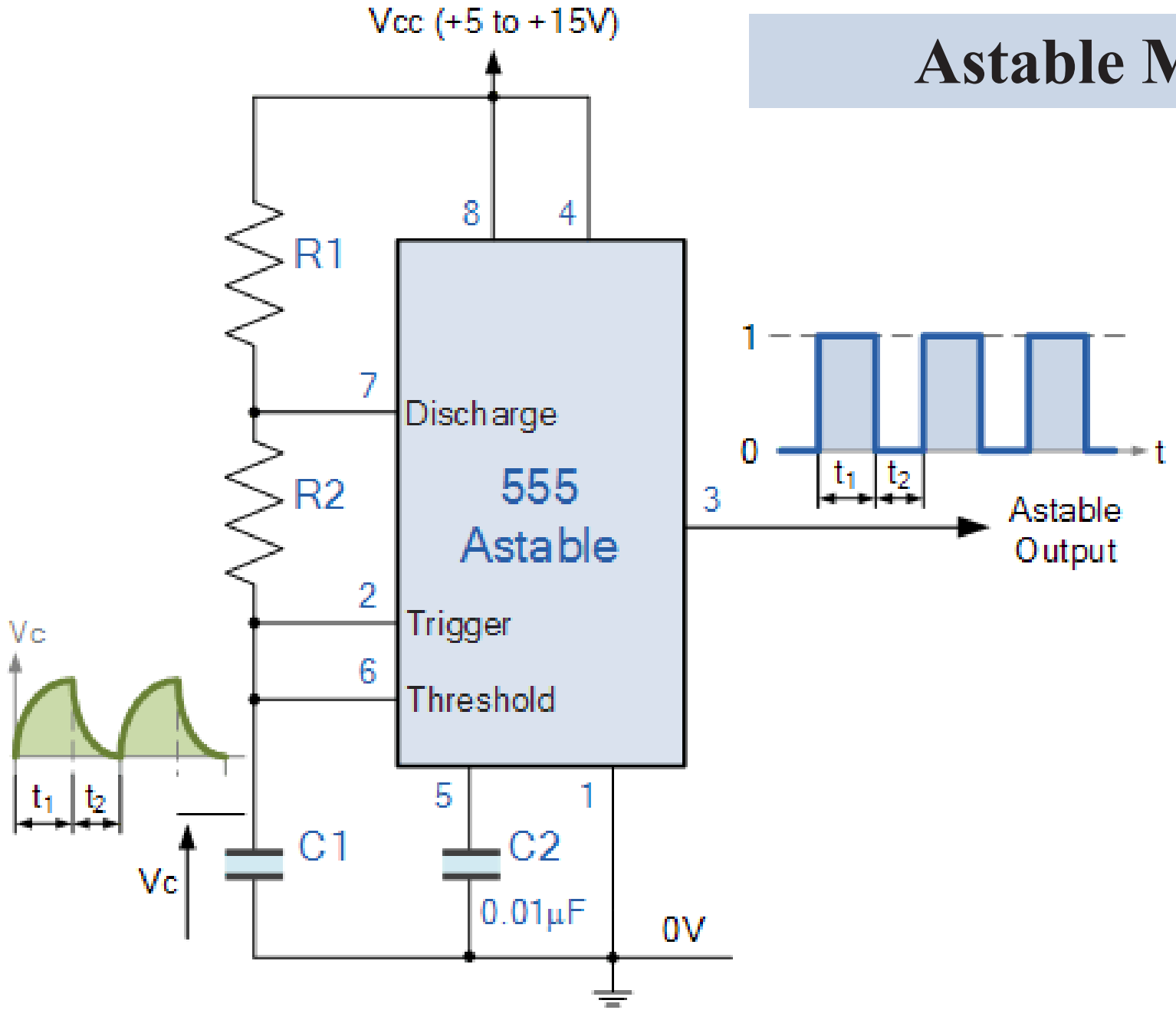


555 Timer IC

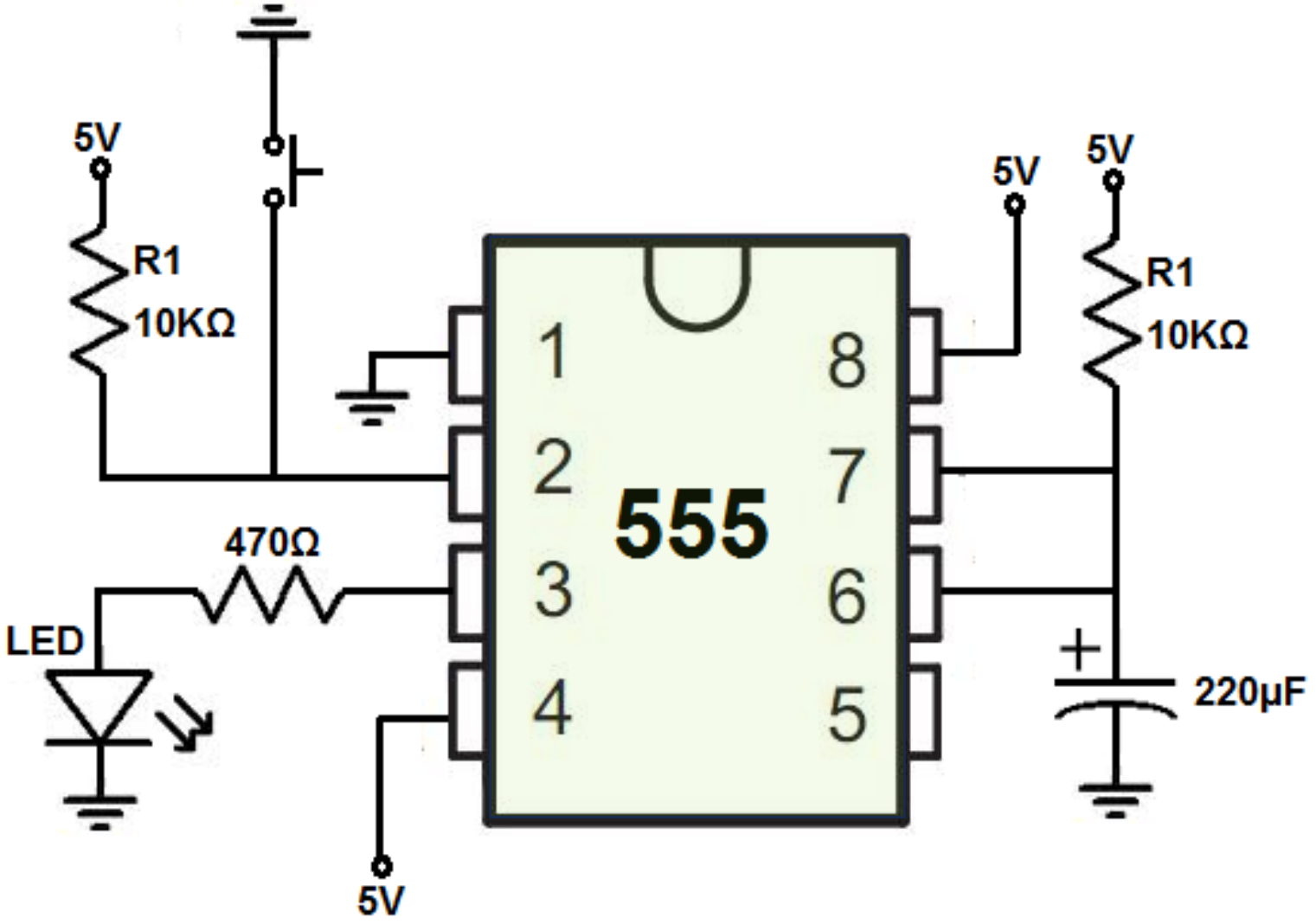

GROUND
1
TRIGGER
2
OUTPUT
3
RESET
4
8
Vcc
7
DISCHARGE
6
THRESHOLD
5
CONTROL VOLTAGE
5 kΩ
5 kΩ
5 kΩ
1
2
S
R
FLIP-FLOP
RESET
555 Timer


555 Timer
Astable
Output
Trigger
Monostable
Output
Trigger
Bistable
Output
Trigger 1
Trigger 2

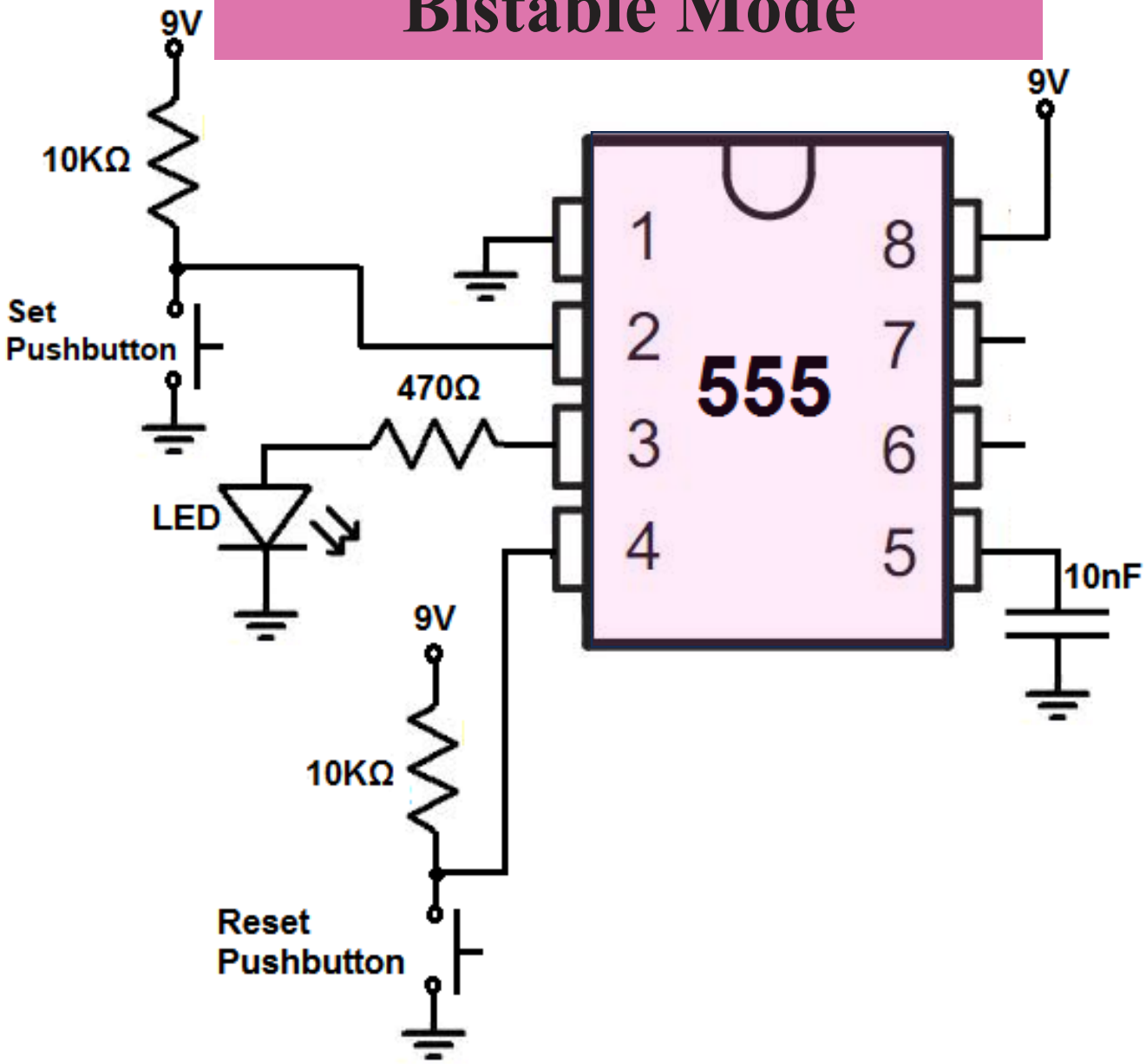
Astable Mode



Monostable Mode



Bistable Mode



One-Shot

Non-retriggerable

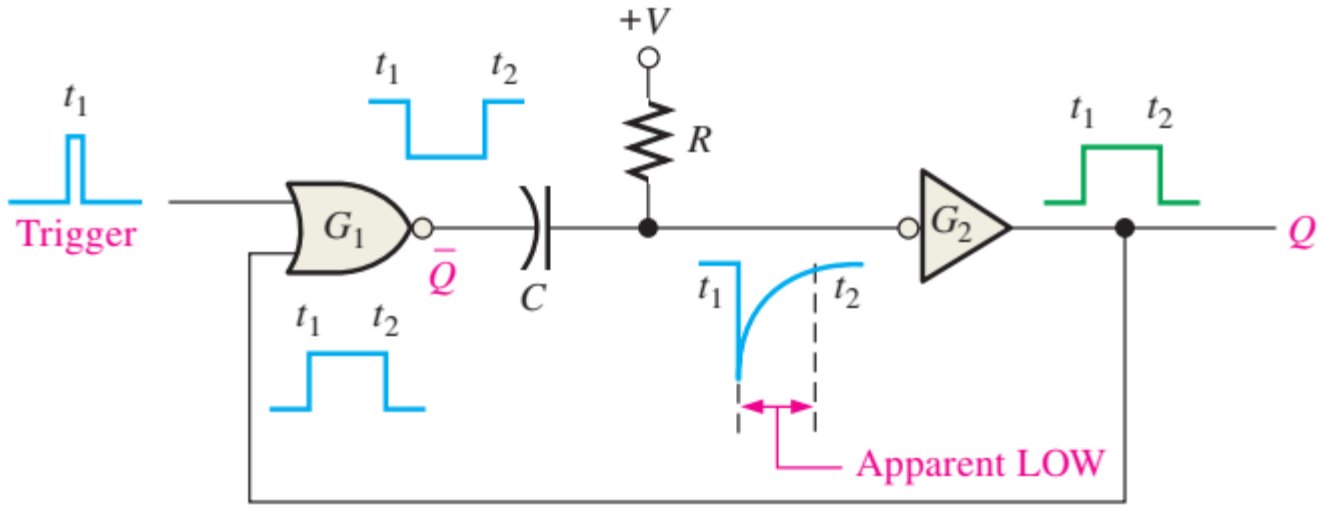
$$t_W = 0.7RC_{\text{EXT}}$$

The Schmitt-Trigger

$$t_W = 0.32RC_{\text{EXT}}\left(1 + \frac{0.7}{R}\right)$$

The 555 Timer

$$t_W = 1.1R_1C_1$$



Oscillators (Astable Multivibrators)

The frequency of oscillation

$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$

The time that the output is HIGH (t_H)

$$t_H = 0.7(R_1 + R_2)C_1$$

The time that the output is LOW (t_L)

$$t_L = 0.7R_2C_1$$

The period, T

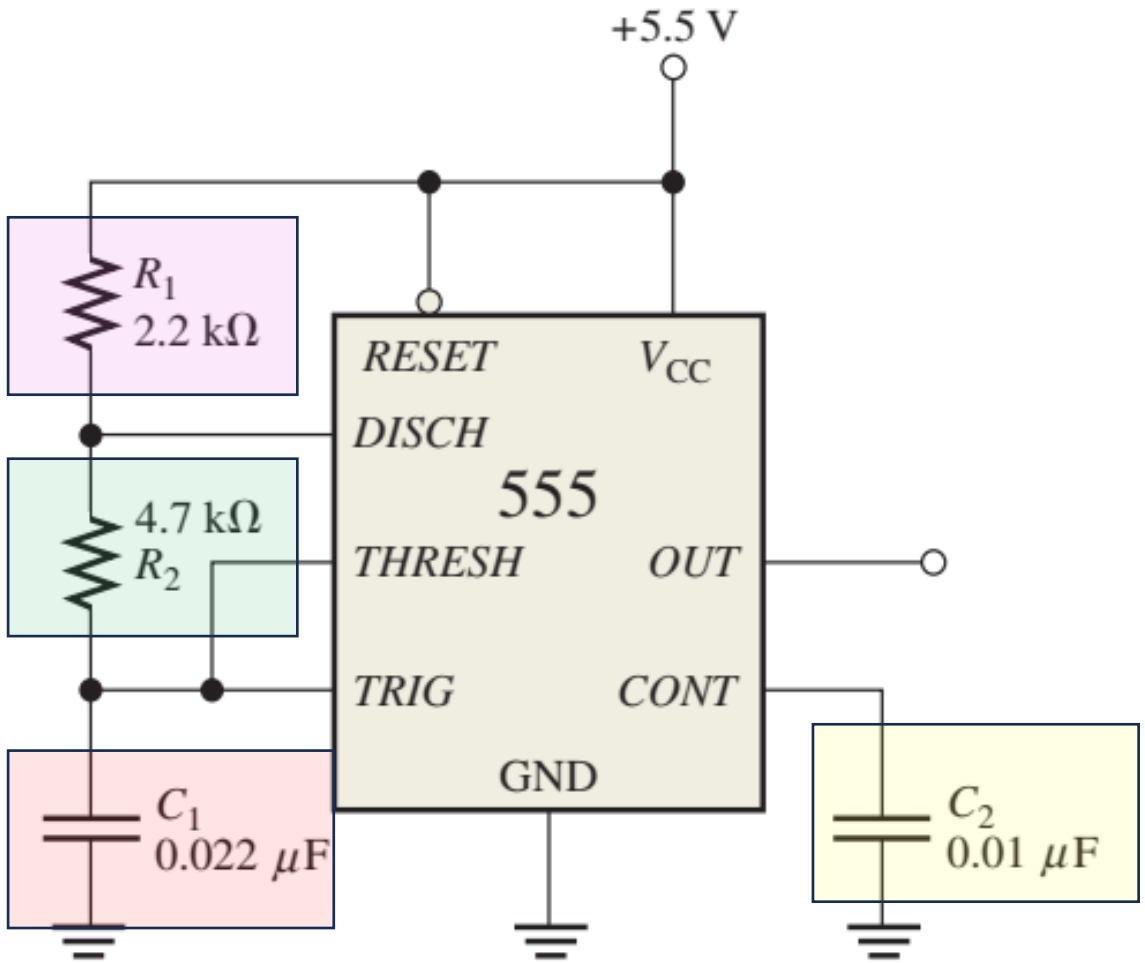
$$T = t_H + t_L = 0.7(R_1 + 2R_2)C_1$$

The duty cycle is

$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\%$$

EXAMPLE

A 555 timer configured to run in the astable mode (pulse oscillator)
Determine the frequency of the output and the duty cycle.



Solution

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} = \frac{1.44}{(2.2\,\text{k}\Omega + 9.4\,\text{k}\Omega)0.022\,\mu\text{F}} = \mathbf{5.64\,kHz}$$

$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\% = \left(\frac{2.2\,\text{k}\Omega + 4.7\,\text{k}\Omega}{2.2\,\text{k}\Omega + 9.4\,\text{k}\Omega}\right)100\% = \mathbf{59.5\%}$$

Home Work

1 Determine the values of the external resistors for a 555 timer used as an astable multivibrator with an output frequency of 20 kHz, if the external capacitor C is 0.002 μF and the duty cycle is to be approximately 75%. 6 M

2 Design A Synchronous counter with JK flip flop that gives the sequence shown in Fig 1-H 14 M

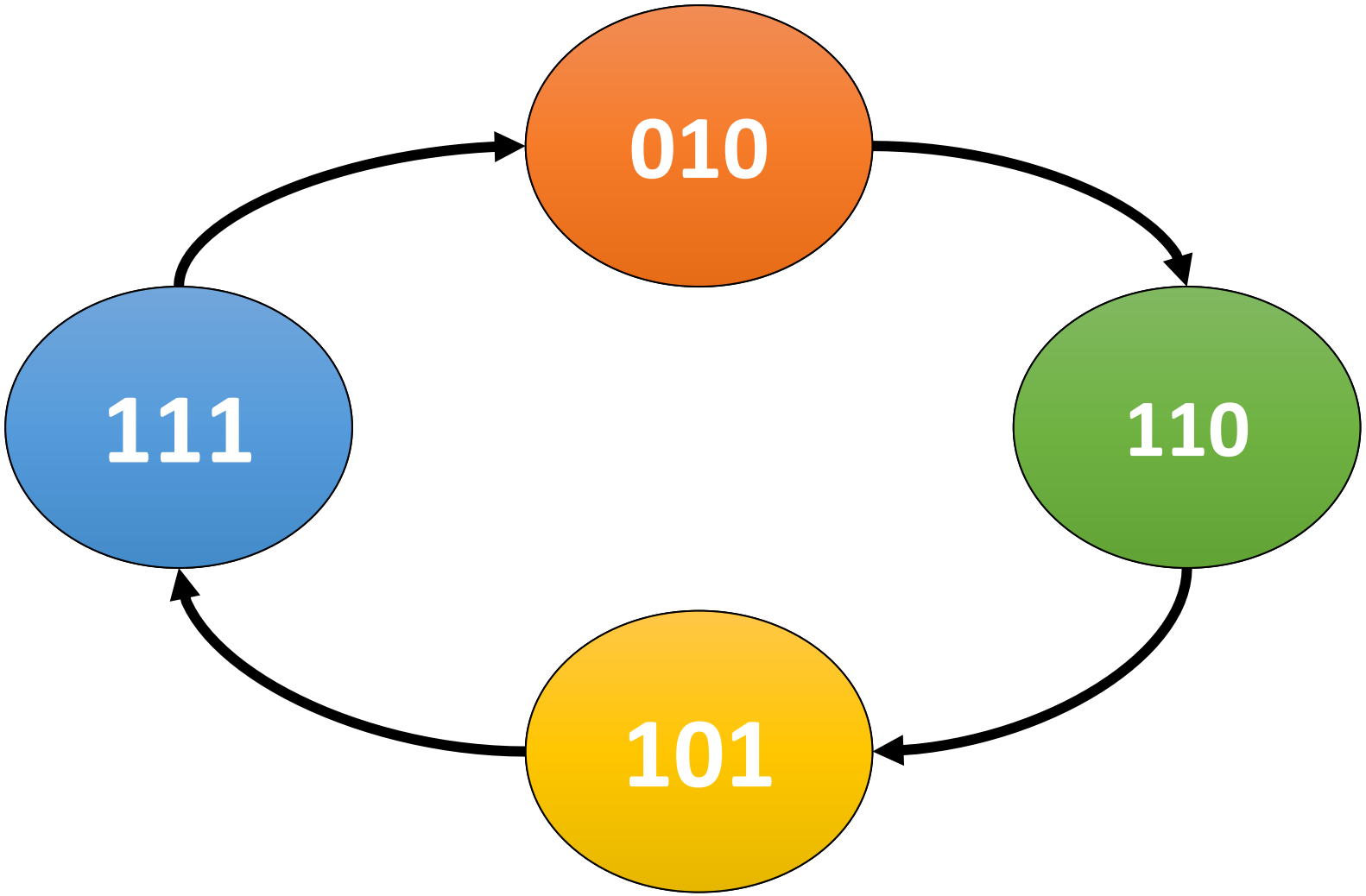


Fig 1-H