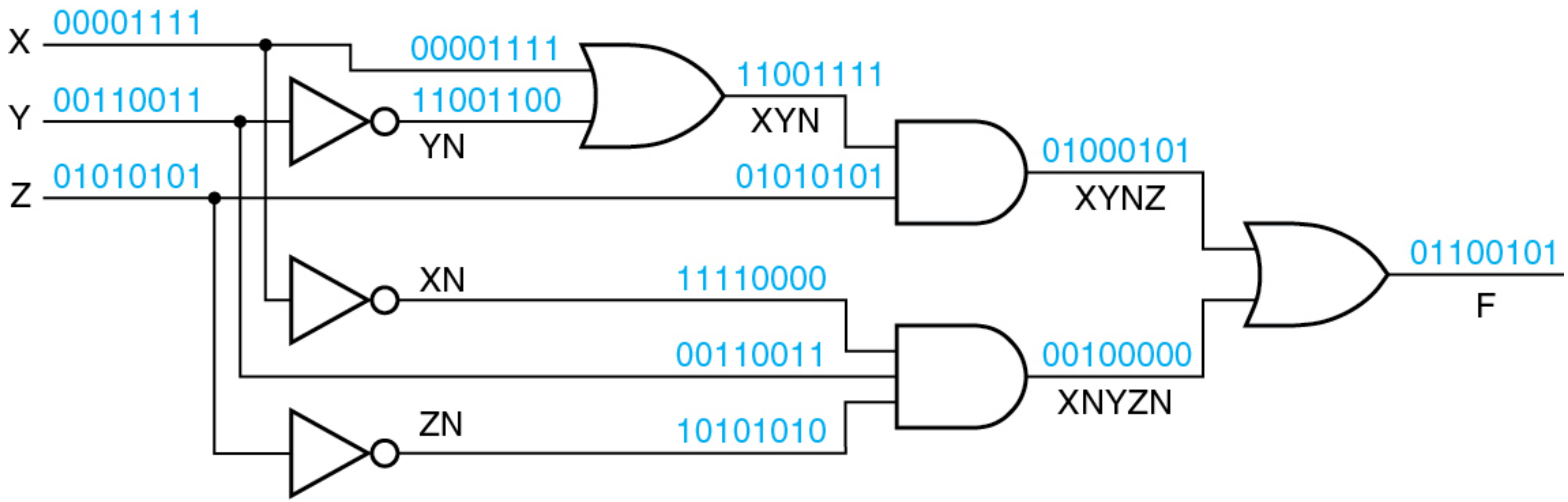
EECE 2322: Fundamentals of Digital Design and Computer Organization

Lecture 2_2: Gates and Numbers

Xiaolin Xu
Department of ECE
Northeastern University

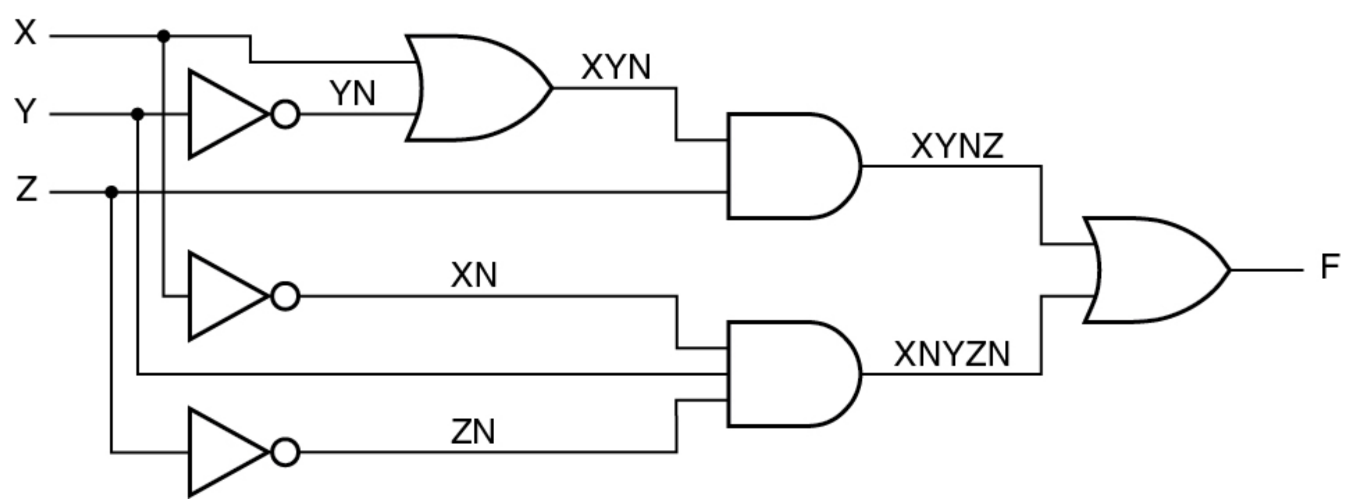
Gate Outputs Created by All Input Combinations

- XN stands for X' or NOT(X)



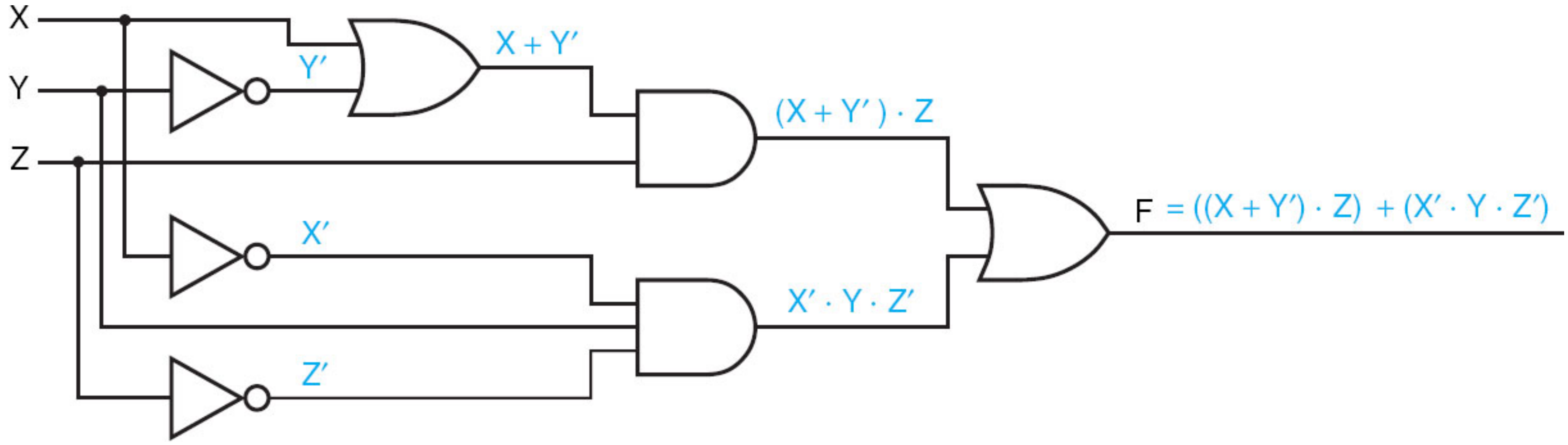
From Circuit to Truth Table

- Think about why

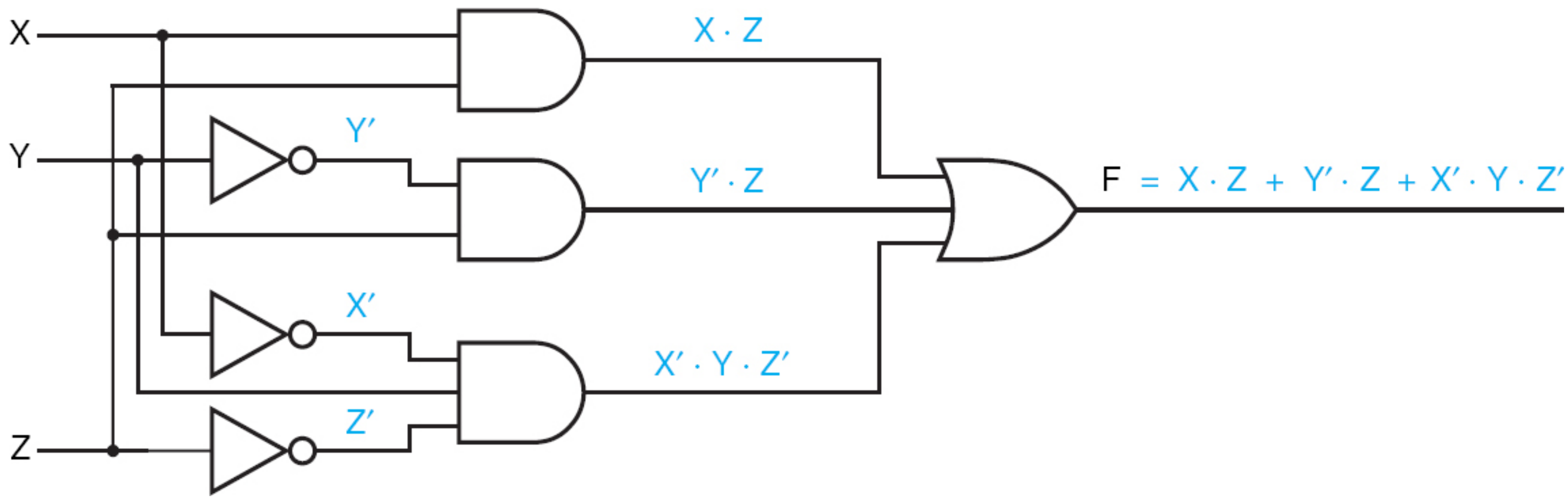


Row	X	Y	Z	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

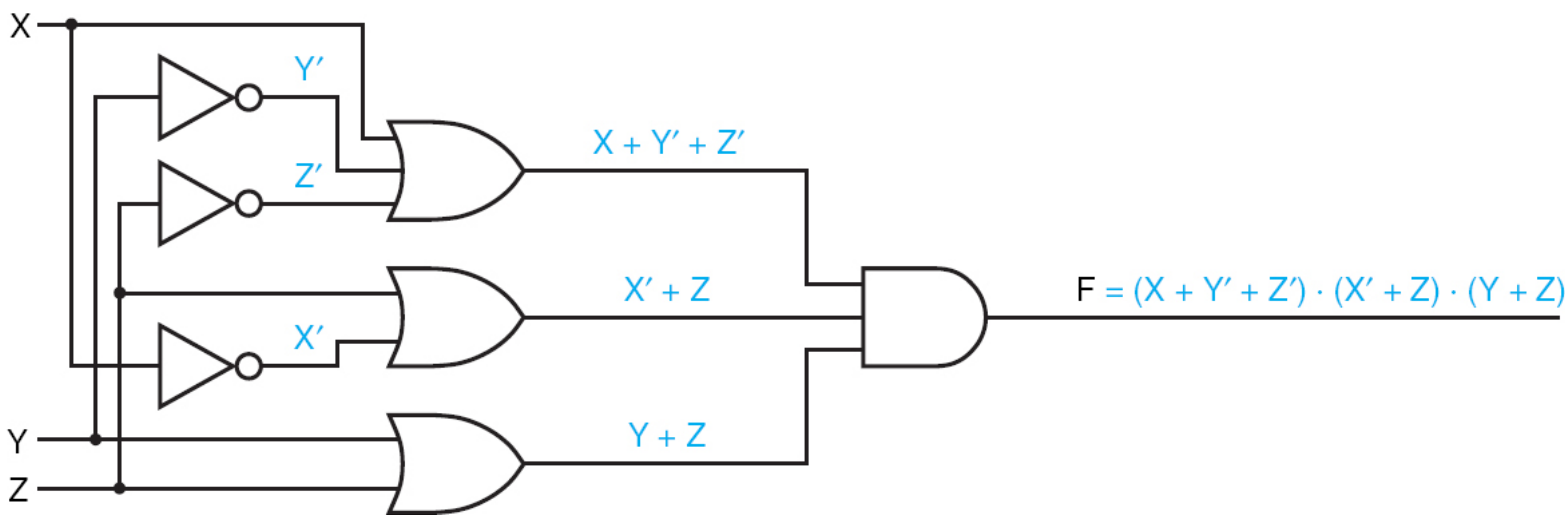
Logic Expressions for Signal Lines



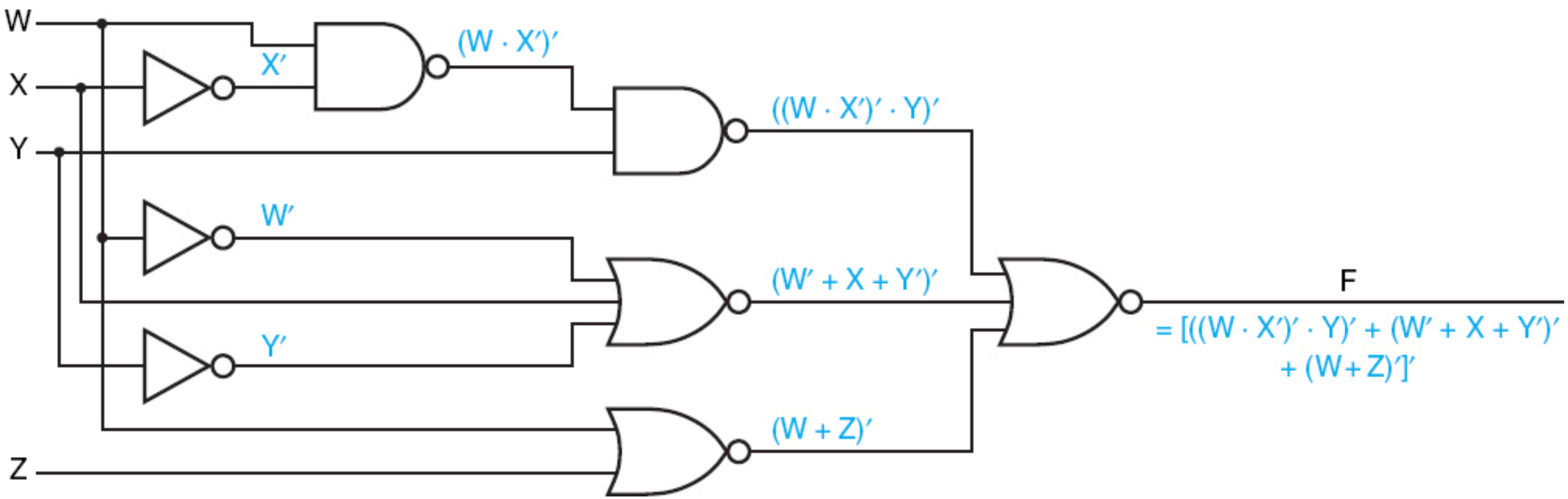
Two-Level AND-OR Circuit



Two-Level OR-AND Circuit



Algebraic Analysis of a Logic Circuit with NAND and NOR Gates



Logic Synthesis

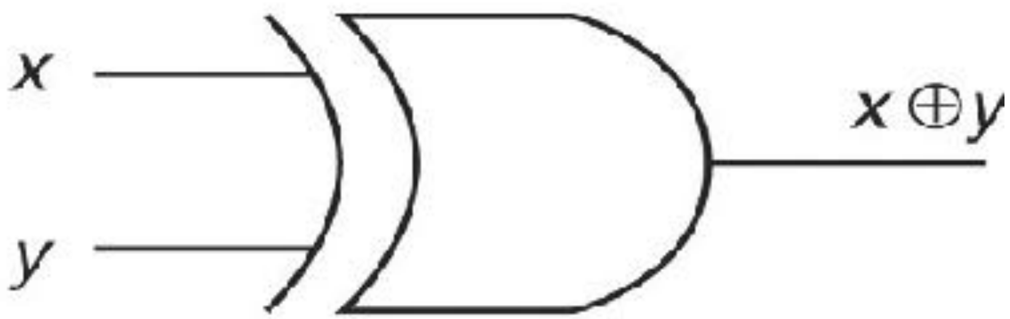
- What is logic synthesis? and Why?

Logic Synthesis

- What is logic synthesis? and Why?
- Only some standard logic cells are used in practical design.

Lab 0: XOR Gate

- XOR: Exclusive-OR gate



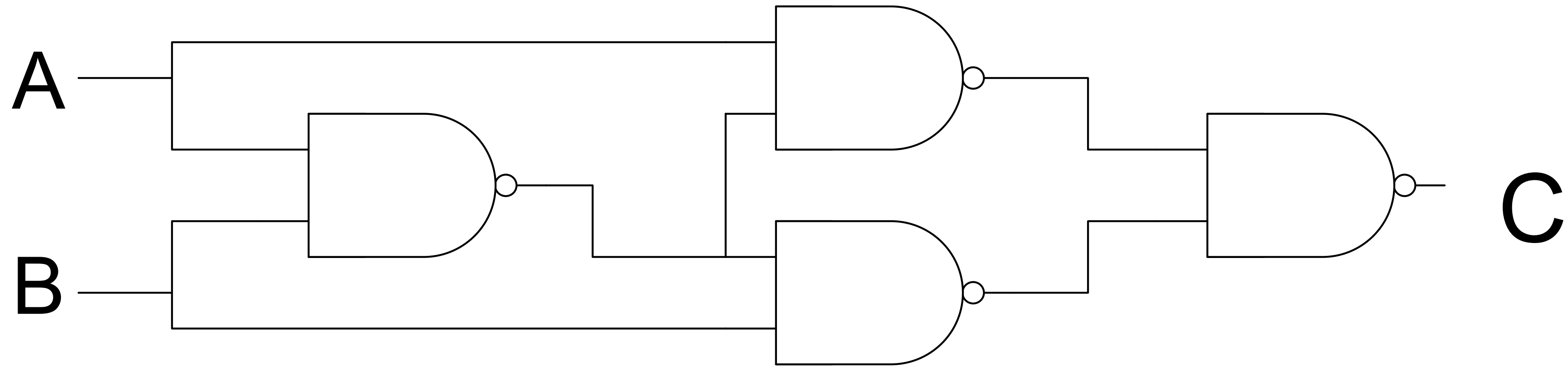
Truth Table:

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

Symbol ⊕ to designate its operation.

Similar to the OR gate but excludes (has the value 0 for) the combination with both X and Y equal to 1.

Lab 0: XOR with NAND Gates



Properties of XOR Gate

$X \oplus 0 = X$

$X \oplus X = 0$

$X \oplus \overline{Y} = \overline{X \oplus Y}$

$X \oplus 1 = \overline{X}$

$X \oplus \overline{X} = 1$

$\overline{X} \oplus Y = \overline{X \oplus Y}$

Commutative:

$A \oplus B = B \oplus A$

Associative:

$(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$

Binary, Decimal, Octal, and Hexadecimal Numbers

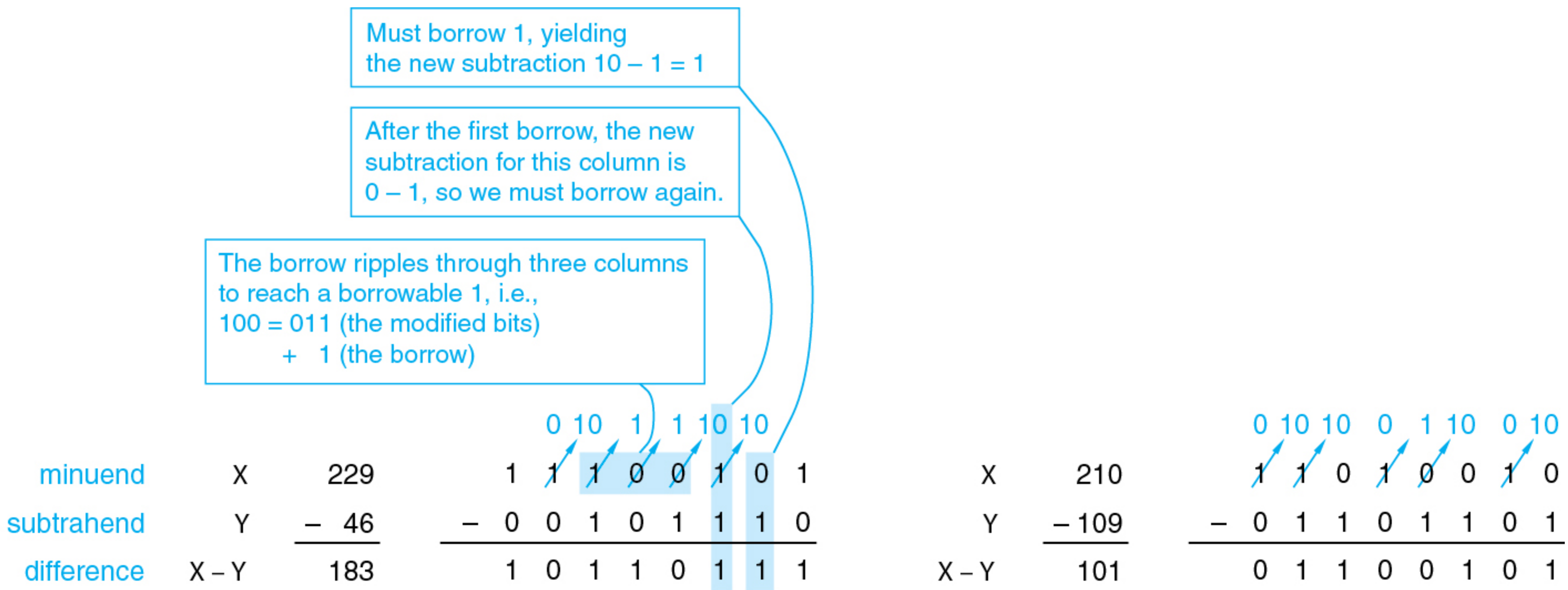
Binary	Decimal	Octal	3-Bit String	Hexadecimal	4-Bit String
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	—	8	1000
1001	9	11	—	9	1001
1010	10	12	—	A	1010
1011	11	13	—	B	1011
1100	12	14	—	C	1100
1101	13	15	—	D	1101
1110	14	16	—	E	1110
1111	15	17	—	F	1111

Binary Additions

				1	1	1	1			
X	190		1	0	1	1	1	1	1	0
Y	+ 141	+	1	0	0	0	1	1	0	1
X + Y	331	1	0	1	0	0	1	0	1	1

				1		1	1			
X	173		1	0	1	0	1	1	0	1
Y	+ 44	+	0	0	1	0	1	1	0	0
X + Y	217		1	1	0	1	1	0	0	1

Binary Subtractions



Representation of Negative Numbers

- Computer needs to know negative numbers as well!
- Signed-Magnitude Representation
- Complement Number Systems
  - Two's complement representation
  - One's complement representation

Signed-Magnitude Representation

- In the sign-magnitude system, a number is interpreted in the usual way, and the sign is denoted with:
  - 1 for -
  - 0 for +

- $01010101_2 = +85_{10}$ $11010101_2 = -85_{10}$
- $01111111_2 = +127_{10}$ $11111111_2 = -127_{10}$

Two's Complement Representation

- In the two's complement system, a n-bit number is obtained by subtracting it from 2^n
- Suppose we want to get -17
- $17_{10} = 00010001_2$
- $2^8 - 17_{10} = ?$
  - $2^n = (2^n-1)+1$
  - $2^n - A = (2^n - 1 - A) + 1$
  - $2^n - 1 = 1111\ldots111$
- $-17_{10} = 11111111_2 - 00010001_2 + 1 = ?$

Two's Complement Representation

- $17_{10} = 11111111_2 - 00010001_2 + 1$
- $= 11101110_2 + 1$
- $= 11101111_2$
- $= -17_{10}$
- The MSB has a negative weight -2^(n-1)

Two's Complement Representation

- Why?

Two's Complement Representation

- Computer is stupid
- Add two numbers 55 and 11 together

- $\quad 0011\ 0111_2(55)$
- $+\ 0000\ 1011_2(11)$
- $\quad 0100\ 0010_2(66)$

Two's Complement Representation

- Computer is stupid
- Subtract 11 from 55
  - $-11 = 1111\ 1111_2 - 0000\ 1011_2 + 1 = 1111\ 0100_2 + 1 = 1111\ 0101_2$
- $\quad 0011\ 0111_2 (55)$
- $+\ 1111\ 0101_2 (-11)$
- $\quad 0010\ 1100_2 (44)$

Two's Complement Representation

- Computer is stupid
- Subtract 55 from 11
  - $-55 = 1111\ 1111_2 - 0011\ 0111_2 + 1 = 1100\ 1000_2 + 1 = 1100\ 1001_2$
- $\quad 0000\ 1011_2 (11)$
- $+\ 1100\ 1001_2 (-55)$
- $\quad 1101\ 0100_2 (-44)$

What is the “E-reason”?

- Any reason to design/use two’s complement systems?

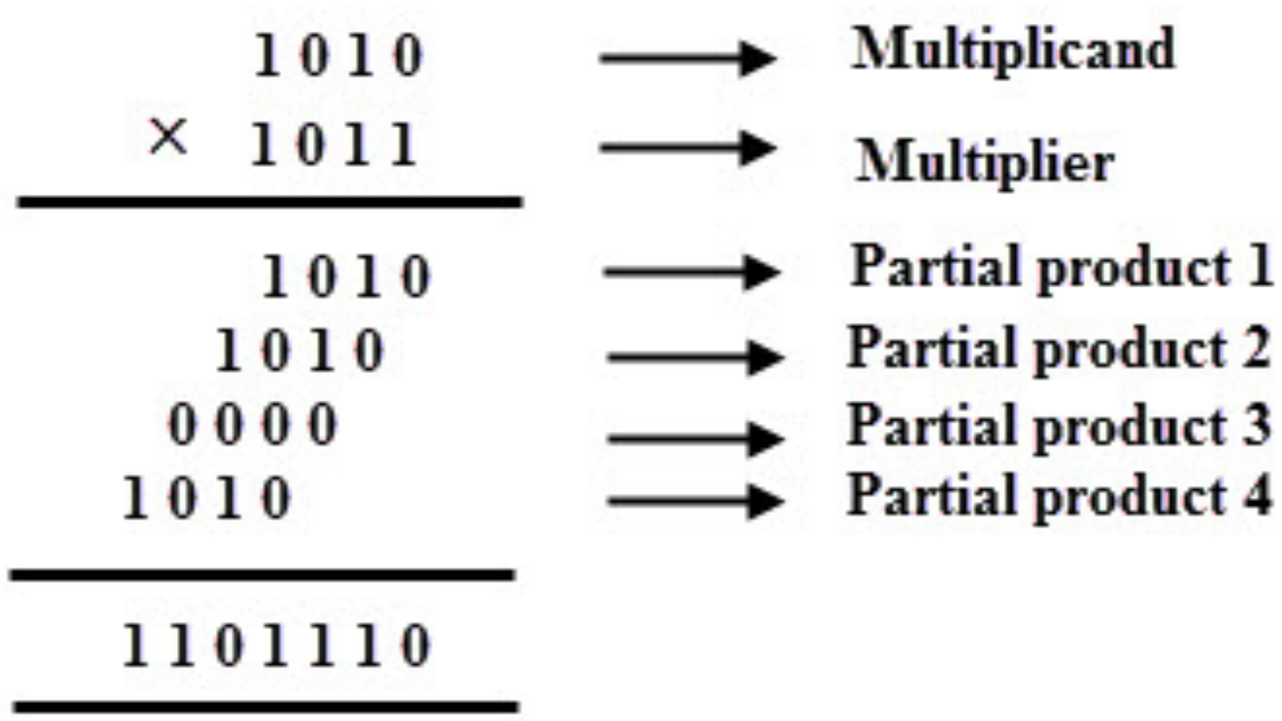
One's Complement Representation

- In the one's complement system, a n-bit number is obtained by subtracting it from 2^n-1
- Suppose we want to get -17
- $17_{10} = 00010001_2$
- $-17_{10} = 11111111_2 - 00010001_2 = ?$

Decimal and 4-Bit Numbers

Decimal	Two's Complement	Ones' Complement	Signed Magnitude	Excess 2^{m-1}
–8	1000	—	—	0000
–7	1001	1000	1111	0001
–6	1010	1001	1110	0010
–5	1011	1010	1101	0011
–4	1100	1011	1100	0100
–3	1101	1100	1011	0101
–2	1110	1101	1010	0110
–1	1111	1110	1001	0111
0	0000	1111 or 0000	1000 or 0000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

Binary Multiplication

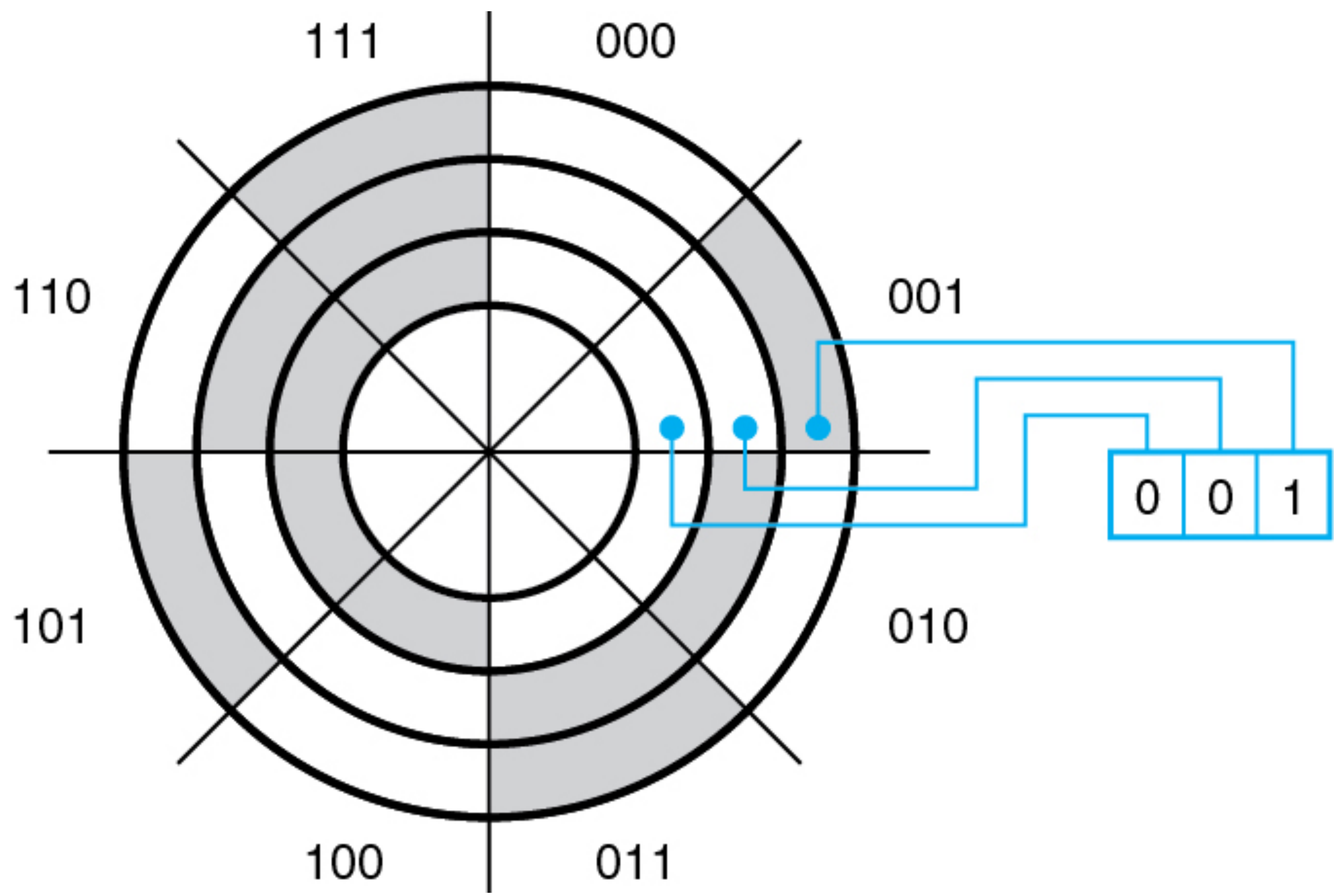


Binary Division

19		10011	quotient
11)217	1011	)11011001	dividend
11		1011	shifted divisor
107		0101	reduced dividend
99		0000	shifted divisor
8		1010	reduced dividend
		0000	shifted divisor
		10100	reduced dividend
		1011	shifted divisor
		10011	reduced dividend
		1011	shifted divisor
		1000	remainder

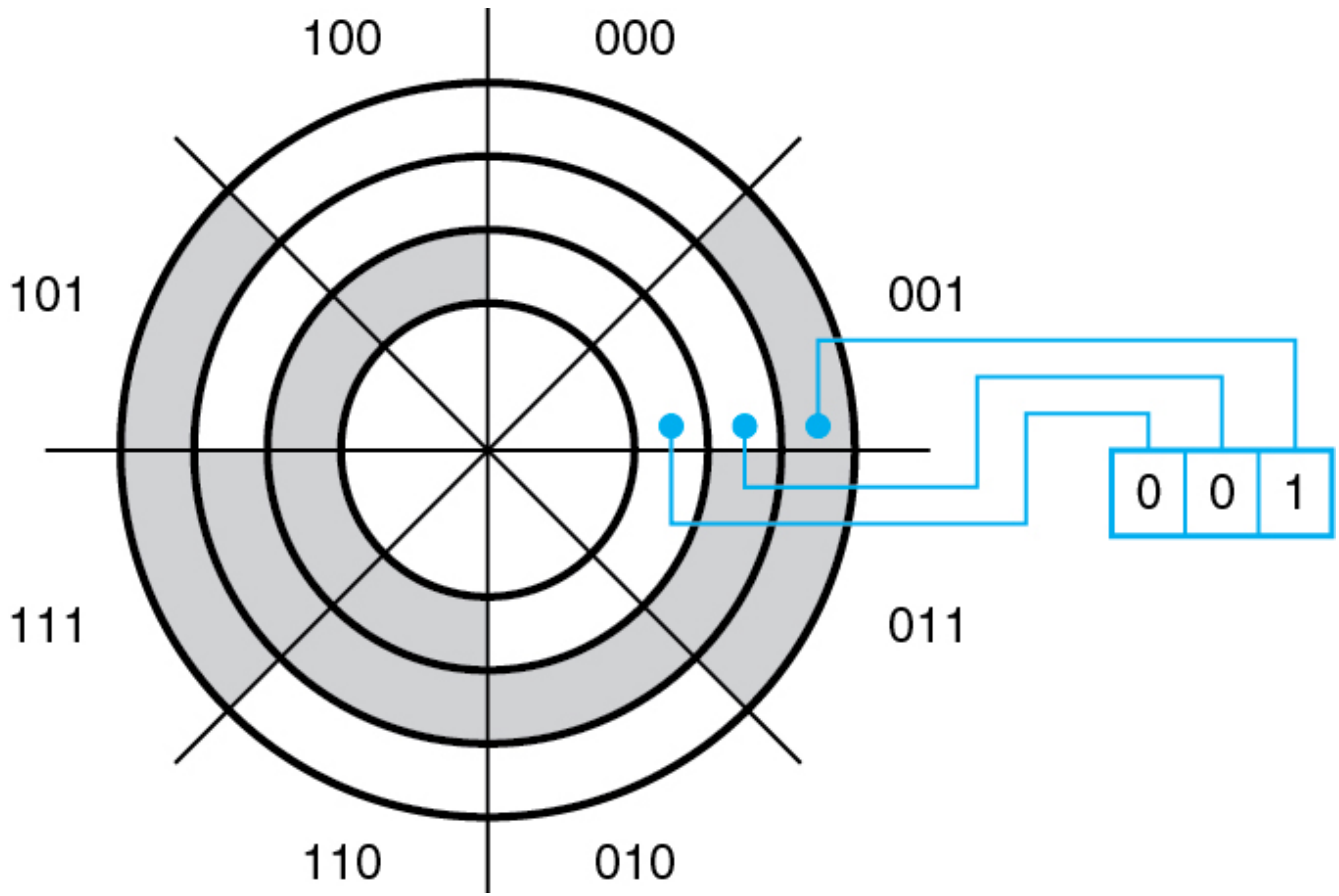
Binary Code and Real-world Applications

- A Mechanical Encoding Disk Using a 3-Bit Binary Code
- Any problem?



Binary Code and Real-world Applications

- A Mechanical Encoding Disk Using a 3-Bit Binary Code
  - Gray Code



Hamming Distance

- In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different