CS 221
Logic Design

Fall 2021

By Wessam El-Behaidy & Salwa Osama

INTRODUCTION

Lecture 1

Important Rules

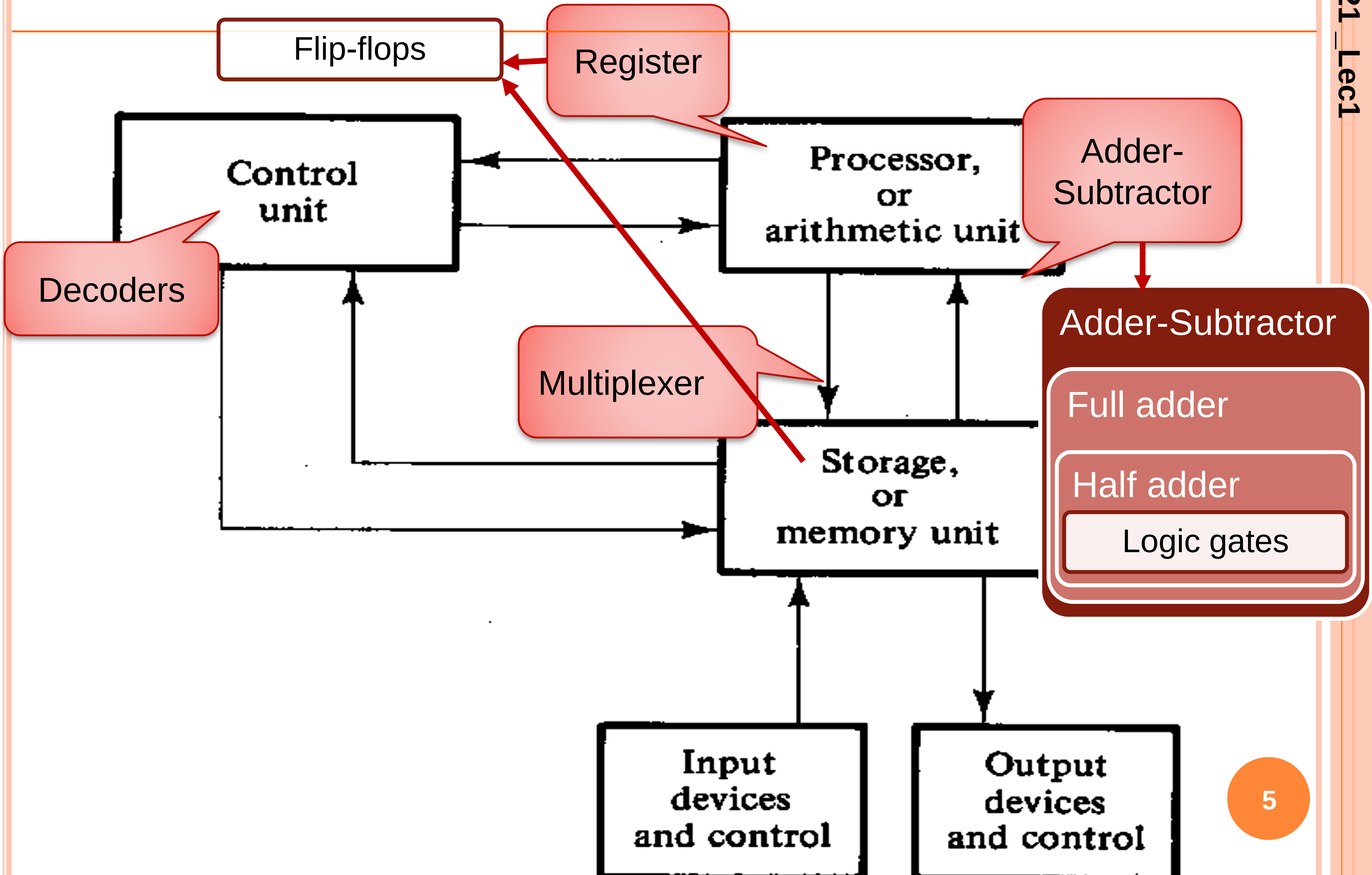
- Cellular phones 'OFF'

- No more than 10 min. delay

WHY STUDY LOGIC DESIGN?

- Constructing large systems from small components
  - It’ s the fundamental prerequisite to understand computer design and architecture.
  - Digital computer is an interconnection of digital modules.

BLOCK DIAGRAM OF A DIGITAL COMPUTER



CS 221 Objectives

Students, by the end of the course, should be able to:

- Analyze, design and implement combinational and synchronous digital circuits.

READINGS

- **Text Book**
  - Mano, M. M. and Ciletti, M. D. (2007), Digital design, Upper Saddle River, NJ: Prentice-Hall, 5th ed.
- **Course Handouts**

Grading Policy

Mid-term Exam	20 %
Semester Work	20 %
Final-Exam (Written)	60 %

HOW TO SUCCEED

- Attend the lecture
- Do your works
- We will learn together how to think.
  - Capture the essence about the topic
  - So you can solve similar problems based on what you learned!
  - [Thinking – vs. Memorizing].
- Participate!
- Ask for help! First from your TA. then from me.

Course Content

No.		Content
1	**Basic logic concepts**	Number Systems & Complements
		Binary Codes & logic gates
		Boolean Algebra & Boolean Functions
		Canonical & Standard forms
		K- Map
2	**Combina-tional Logical Design**	Analysis & design Combinational circuits
		Binary adder-Subtractor
		Midterm Exam (7th week)
		Decoder, Encoder, MUX
3	**Sequential Circuits**	Latches & Flip-flop
		Analysis& design of Seq. circuits
4	**Reg.&Count**	Registers & counters
5	**Memory**	RAM & ROM
		Final Exam

Join Our Team Class

- To communicate with us and get course materials (slides l sheets l textbook)

Team class code: xfemyc0

Chapter 1

Digital Systems and Binary Numbers

BINARY LOGIC

- **<u>Binary variables</u>** take on one of two values.
  - We use 1 and 0 to denote the two values.
  - **Examples: A, B, y, z, or X_1**

- **The three basic <u>logical operations</u> are:**
  - **AND**
    - Ex: $z = x \cdot y$ or $z=xy$ "z is equal to ***x AND y***"
  - **OR**
    - Ex: $z = x + y$ "z is equal to ***x OR y***"
  - **NOT**
    - Ex: $z = \overline{x}$ or $z=x'$ "z is equal to ***NOT x*** "

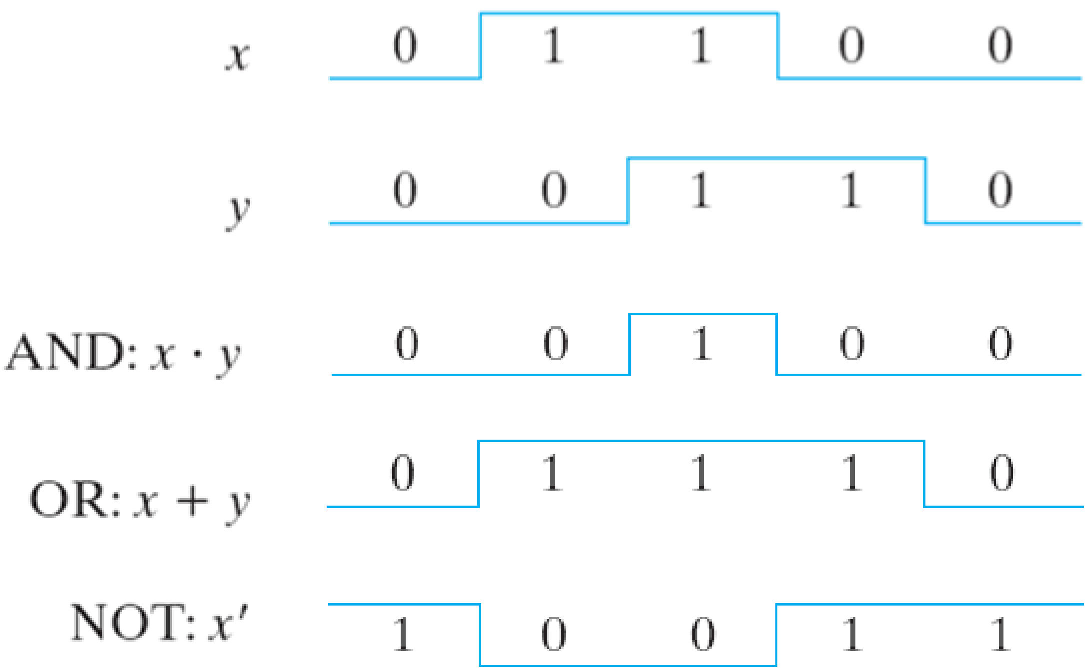
Truth Table

- It is a table of all possible combinations of the variables
- It shows the relation between
  - The values that the variables may take and
  - The result of the operation

Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Timing Diagram



Timing diagram

Figure 1.5

LOGIC GATES SYMBOLS

Denote True *and* False *by 1 and 0 that represent V_{cc} and 0 voltages.*

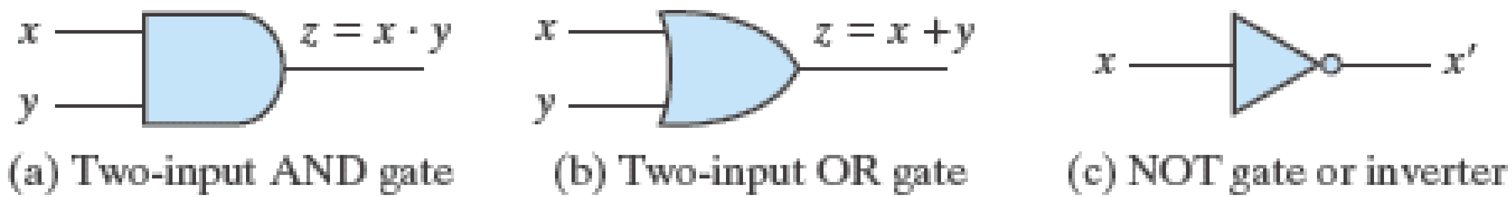


Figure 1.4

Gates with multiple inputs:

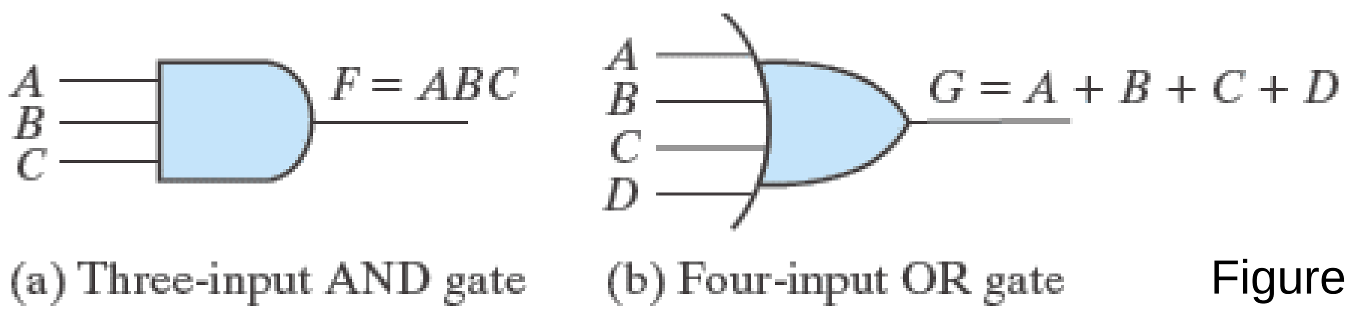
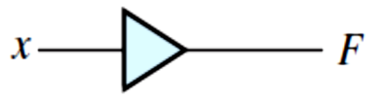


Figure 1.6

When F=1? and when G=1?

OTHER LOGIC GATES

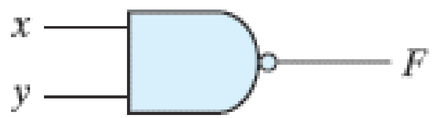
Buffer



$F = x$

x	F
0	0
1	1

NAND

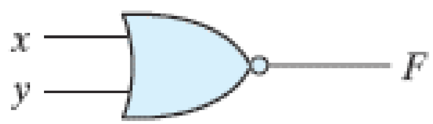


$F = (xy)'$

$= x \uparrow y$

0	0	1
0	1	1
1	0	1
1	1	0

NOR

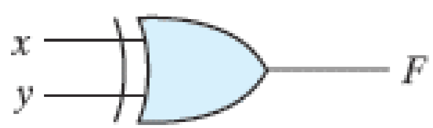


$F = (x + y)'$

$= x \downarrow y$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR (XOR)

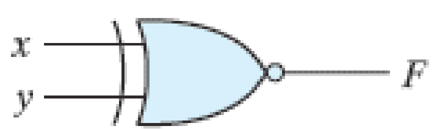


$F = xy' + x'y$

$= x \oplus y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR or equivalence



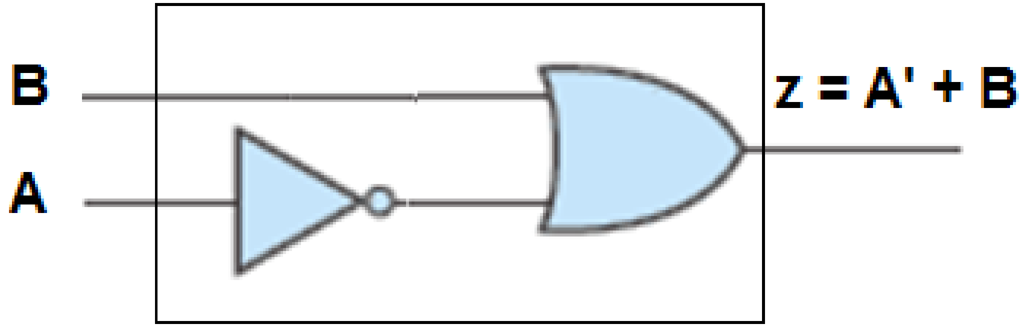
$F = xy + x'y'$

$= (x \oplus y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

EXAMPLE

- Draw a logic gate circuit of **A' +B** and get their truth table



A	**B**	**A'**	**Z= A' +B**
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

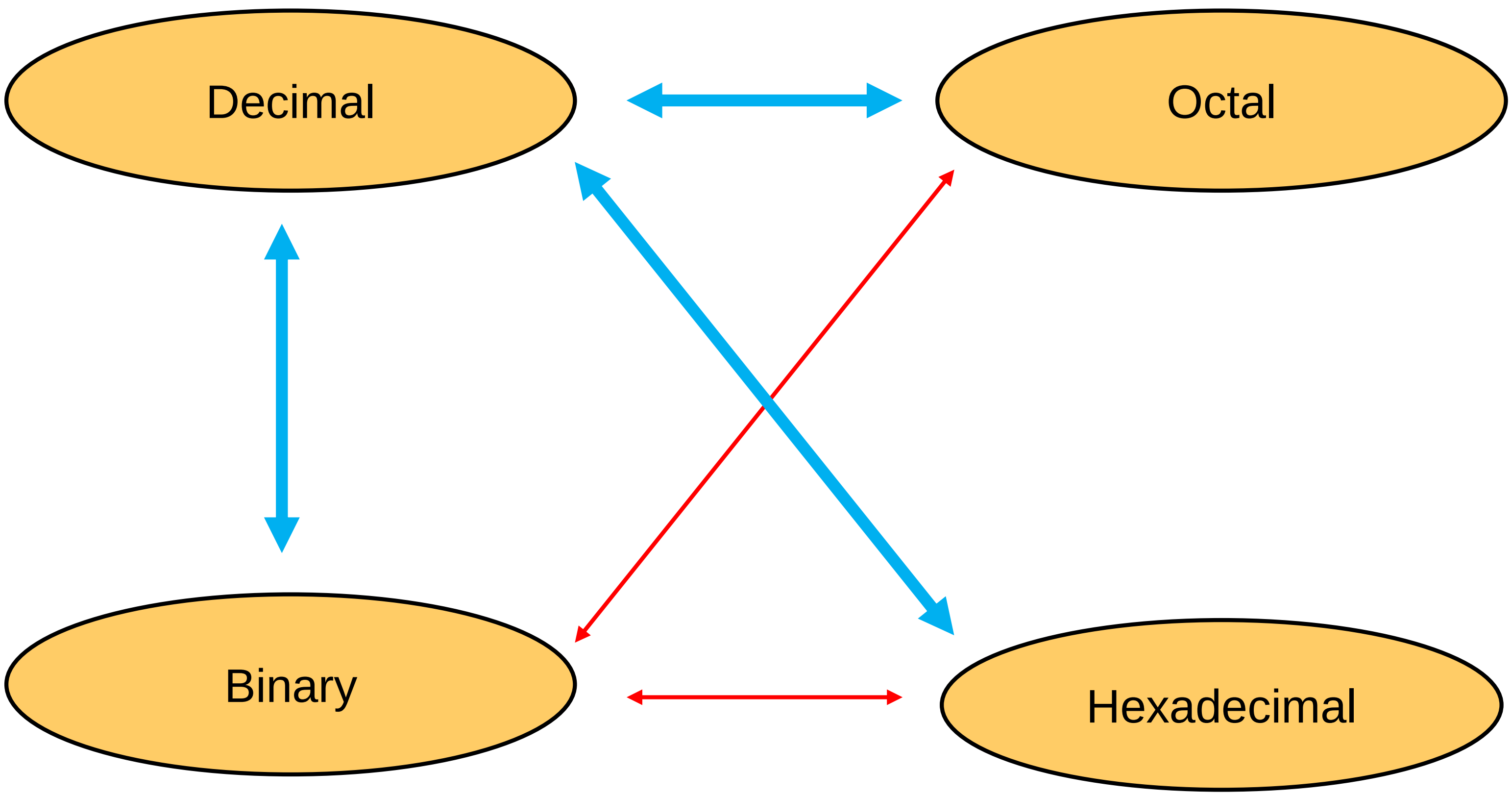
TRY TO SOLVE

- **Draw logic diagrams to implement the following Boolean expression**

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

Break

Number Systems and Conversions



Review on sec. 1.2 → sec. 1.4

COMPLEMENT

- It is used in digital computers to simplify
  - The subtraction operation
  - Logical manipulation

- Simplifying operations leads to:
  - Simpler, less expensive circuits to implement

COMPLEMENT

Diminished Radix Complement	Radix Complement

(r-1)'s complement=$(r^n - 1) - N$

Ex:

- r=2 ➔ r-1=1

➔ 1's complement

1's comp.=$(2^n - 1) - N$

$2^2= 4=100_2$ $-1=11_2$

$2^3= 8=1000_2$ $-1=111_2$

$2^4=16=10000_2$ $-1=1111_2$

$2^5=32=100000_2$ $-1=11111_2$

(r)'s complement = $r^n - N$

Ex:

- r=2

➔ 2's complement=

$2^n - N$

2^n is a binary number represented by **1** followed by ***n* 0's.**

2^n -1 is a binary number represented by ***n* 1's.**

COMPLEMENT

Given a number ***N*** in base ***r*** having ***n*** digits

Diminished Radix Complement

(r-1)'s complement=$(r^n - 1) - N$

Ex:

1's complement (1011000)

$=(2^7-1)- 1011000$

=1111111-1011000

=0100111

or

= toggle 1's to 0's and 0's to 1's

1-0= 1
1-1= 0

Radix Complement

(r)'s complement = $r^n - N$

Ex:

2's Complement (101100)

$=(2^6)- 101100$

= 1000000 –101100 = 010100

or

= **1's complement (101100)+1**

= 010100

or $\underbrace{010}_{\text{toggled}}\boxed{100}$

Unchanged until the first 1

COMPLEMENT

Diminished Radix Complement	Radix Complement

(r-1)'s complement=(r^n - 1) – N

Ex:

- r= 10 ➔ r-1=9

9's complement (2380)

=9999-2380

=7619

(r)'s complement = r^n - N

Ex:

- r= 10

10's Complement (2380)

= 10000 –2380 =7620

or

= 9's complement (2380)+1

=7620

The **(r-1)'s complement** of **octal** and **hexadecimal** numbers is obtained by subtracting each digit from **7** or **F** (decimal 15), respectively.

SUBTRACTION WITH COMPLEMENT

- Example:

If x=21 and y=5, calculate x-y=? And y-x=? using 10’s complement

$$\begin{array}{r} 21 \\ -\ \boxed{05} \\ \hline ? \end{array} \xrightarrow[\text{Comp.}]{\text{10's}} \begin{array}{r} 21 \\ +\ \boxed{95} \\ \hline \boxed{1}16 \end{array}$$

It means x>y
Discard it

Answer = 16

$$\begin{array}{r} 5 \\ -\ \boxed{21} \\ \hline ? \end{array} \xrightarrow[\text{Comp.}]{\text{10's}} \begin{array}{r} 5 \\ +\ \boxed{79} \\ \hline \boxed{}84 \end{array}$$

No end carry
It means x<y

Answer **= - (10’s complement of 84)**
= -16

SUBTRACTION WITH COMPLEMENT

- Example 1.7 (pp.28-29)

Given the two binary numbers $X = 1010100$ and $Y = 1000011$, perform the subtraction (a) $X - Y$ and (b) $Y - X$ using 2's complements.

(a)

$$
\begin{aligned}
X &= \quad 1010100 \\
\text{2's complement of } Y &= +\ \underline{0111101} \\
\text{Sum} &= \boxed{1}0010001 \\
\text{Answer: } X - Y &= \quad 0010001
\end{aligned}
$$

It means x>y
Discard it

(b)

$$
\begin{aligned}
Y &= \quad 1000011 \\
\text{2's complement of } X &= +\ \underline{0101100} \\
\text{Sum} &= \quad 1101111
\end{aligned}
$$

There is no end carry.

Answer: $Y - X = -(\text{2's complement of } 1101111) = -0010001$

SUBTRACTION WITH COMPLEMENT (CONT.)

- Example 1.8: Previous problem using 1's complement.

(a) $X - Y = 1010100 - 1000011$ using 1's complement.

$$
\begin{array}{rr}
X = & 1010100 \\
\text{1's complement of } Y = & +\ \underline{0111100} \\
\text{Sum} = & \boxed{1}0010000 \\
\text{End-around carry} & \underline{+\quad 1} \\
\text{Answer: } X - Y = & 0010001
\end{array}
$$

(b) $Y - X = 1000011 - 1010100$ using 1's complement.

$$
\begin{array}{rr}
Y = & 1000011 \\
\text{1's complement of } X = & +\ \underline{0101011} \\
\text{Sum} = & 1101110
\end{array}
$$

There is no end carry.

Answer: $Y - X = \boxed{-(\text{1's complement of } 1101110)} = -0010001$

SIGNED BINARY NUMBERS

- Signed number last bit (one MSB) is **sign bit**

  0 = + **1= -**

  Assume: 8 bit number
  - Unsigned 9 : 0000 1001
- Positive number
  - Signed +9 : 0000 1001
- Negative number
  - Signed magnitude -9 : 1000 1001
  - Signed Complement:
    - 1's Complement of 9 = 1111 0110
    - 2's Complement of 9 = 1111 0111

Most used in signed binary arithmetic

SIGNED BINARY ARITHMETIC

Pay Attention:

1) Any carry out of the sign bit position → Discard it
2) Overflow

Addition

$$\begin{array}{rr} +6 & 00000110 \\ +13 & 00001101 \\ \hline +19 & 00010011 \end{array} \qquad \begin{array}{rr} -6 & 11111010 \\ +13 & 00001101 \\ \hline +7 & 00000111 \end{array}$$

2's complement of +6

$$\begin{array}{rr} +6 & 00000110 \\ -13 & 11110011 \\ \hline -7 & 11111001 \end{array} \qquad \begin{array}{rr} -6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & \boxed{1}1101101 \end{array}$$

2's complement of +13

Means negative result

Subtraction

$$(\pm A) - (+B) = (\pm A) \boxed{+} (-B)$$

$$(\pm A) - (-B) = (\pm A) \boxed{+} (+B)$$

Subtraction becomes addition

THANKS

We covered:

Ch.1 (sec. 1.2 → sec. 1.6, sec.1.9)

Next Week : Boolean Algebra & binary codes