Tishk International University
Science Faculty
IT Department

Logic Design

Lecture 01: Number Systems

2nd Grade

Instructor: Alaa Ghazi

Course Name: LOGIC DESIGN
Code/Section: IT 231 - S

Instructor: Mr. Alaa Ghazi

Qualification: M.Sc. in Computer Engineering

Email: alaa.ghazi@tiu.edu.iq

Room No.: 313

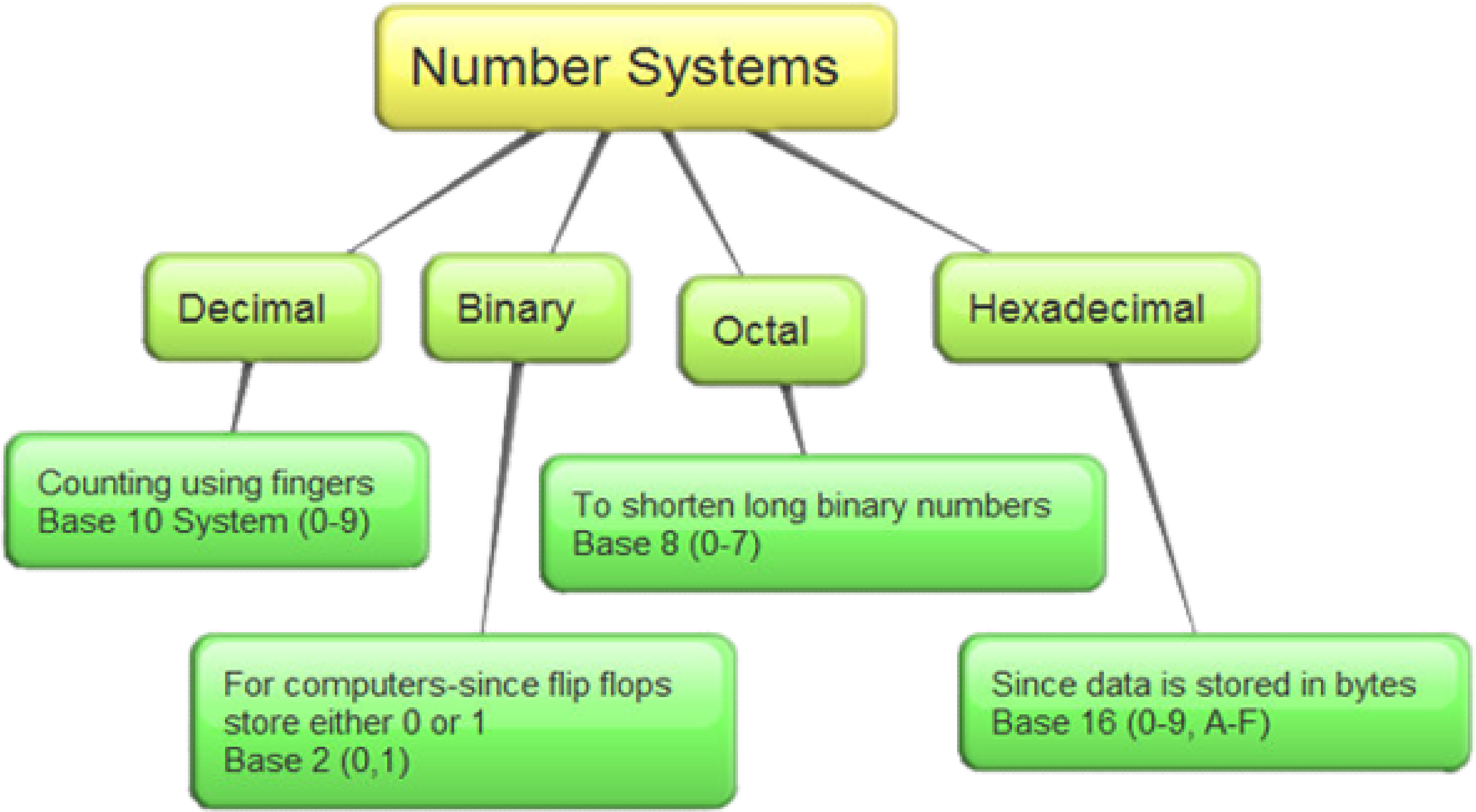
Reference:

- Thomas L. Floyd "Digital Fundamentals" 9th edition Prentice Hall

COURSE CONTENT

#	Lecture Name
1	Number Systems
2	Logic Gates and Boolean Algebra
3	Logic Simplification
4	Combinational and Sequential Logic Systems

Lecture 1
Number Systems



Topics

1. What are Numbers?
2. Decimal Numbers
3. Positional Notation
4. Binary Numbers
5. Bainary Arithmetic
6. Complements of Binary Numbers
7. Signed Numbers
8. Hexadecimal Numbers
9. Octal Numbers
10. Binary Coded Decimal (BCD)

Chapter Goals

- Distinguish among categories of numbers
- Describe positional notation
- Convert numbers in other bases to base 10
- Convert base-10 numbers to numbers in other bases
- Describe the relationship between bases 2, 8, and 16
- Explain the importance to computing of bases that are powers of 2

1. What are Numbers?

Natural Numbers

Zero and any number obtained by repeatedly adding one to it.

Examples: 100, 0, 45645, 32

Negative Numbers

A value less than 0, with a – sign

Examples: -24, -1, -45645, -32

Integers

A natural number, a negative number

Examples: 249, 0, - 45645, - 32

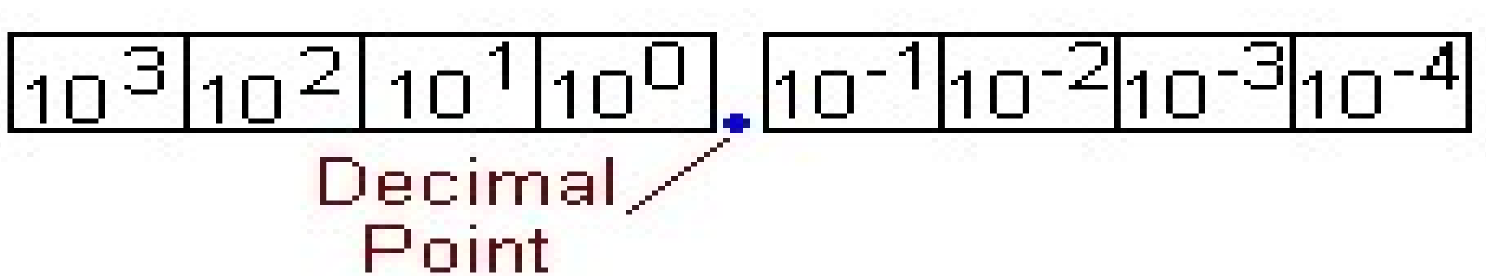
Rational Numbers

An integer or the quotient of two integers

Examples: -249, -1, 0, 3/7, -2/5

2. 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:



3. Positional Notation

642 is 600 + 40 + 2 in BASE 10

The base of a number determines the number of different digit symbols (numerals) and the values of digit positions

642 in base 10 *positional notation* is:

$$
\begin{aligned}
& 6 \times 10^2 = 6 \times 100 = 600 \\
+ & 4 \times 10^1 = 4 \times 10 = 40 \\
+ & 2 \times 10^0 = 2 \times 1 = 2 \quad = 642 \text{ in base 10}
\end{aligned}
$$

This number is in base 10

The power indicates the position of the number

Positional Notation

As a formula:

$$d_n * R^{n-1} + d_{n-1} * R^{n-2} + \ldots + d_2 * R^1 + d_1 * R^0$$

R is the base of the number

n is the number of digits in the number

d is the digit in the i^{th} position in the number

642 is 6 * 10^2 + 4 * 10 + 2 * 1

Decimal Number with Fraction

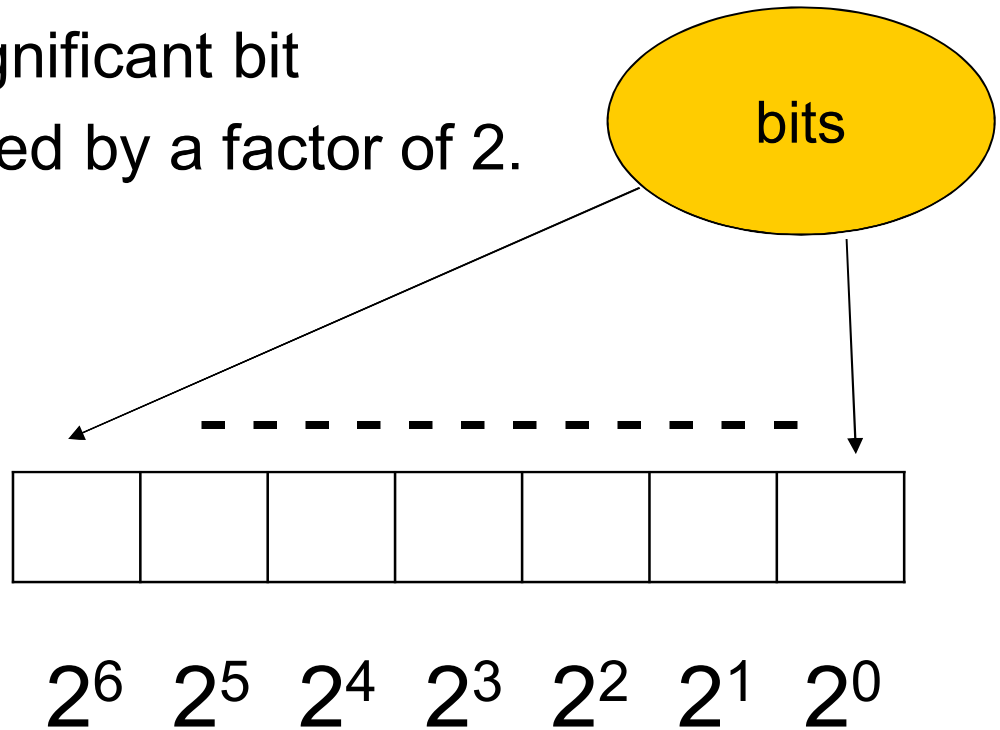
Express the decimal number 568.23 as a sum of the values of each digit.

The whole number digit 5 has a weight of 100, which is 10^2, the digit 6 has a weight of 10, which is 10^1, the digit 8 has a weight of 1, which is 10^0, the fractional digit 2 has a weight of 0.1, which is 10^{-1}, and the fractional digit 3 has a weight of 0.01, which is 10^{-2}.

$$
\begin{aligned}
568.23 &= (5 \times 10^2) + (6 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2}) \\
&= (5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0.1) + (3 \times 0.01) \\
&= \mathbf{500} + \mathbf{60} + \mathbf{8} + \mathbf{0.2} + \mathbf{0.03}
\end{aligned}
$$

4. Binary Numbers

- Two digits called bits
- 0 or 1
- Right most bit is least significant bit
- Left most bit is most significant bit
- Each position is weighted by a factor of 2.



Counting in Binary

Decimal	Binary			
	2^3	2^2	2^1	2^0
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

Binary to Decimal

- Add the weights of all the bits that are 1.
- Ignore the bits that are 0.
- Example 1: Convert 1101101 to decimal.

  1st determine the weights

2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	1	1	0	1

  $(1\ 1\ 0\ 1\ 1\ 0\ 1)_2 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0$

  $= 64 + 32 + 8 + 4 + 1$

  $= (109)_{10}$

- Example 2: Convert 1011110 to decimal.

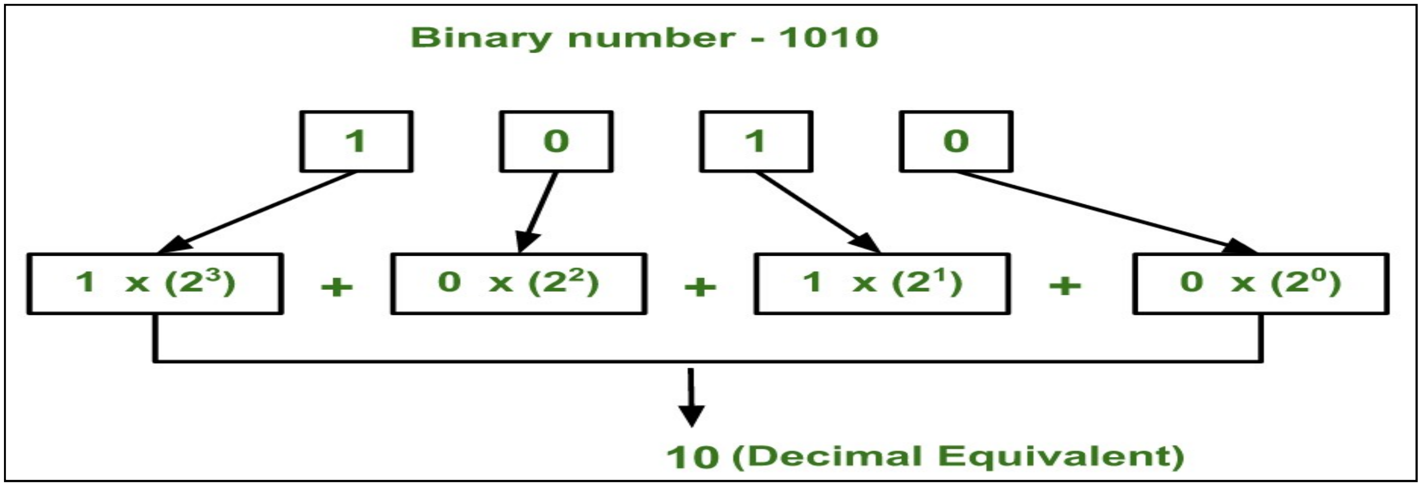
2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	1	1	1	0

  $(1\ 0\ 1\ 1\ 1\ 1\ 0)_2 = 2^6 + 2^4 + 2^3 + 2^2 + 2^1$

  $= 64 + 16 + 8 + 4 + 2$

  $= (94)_{10}$

Examples of Converting Binary to Decimal



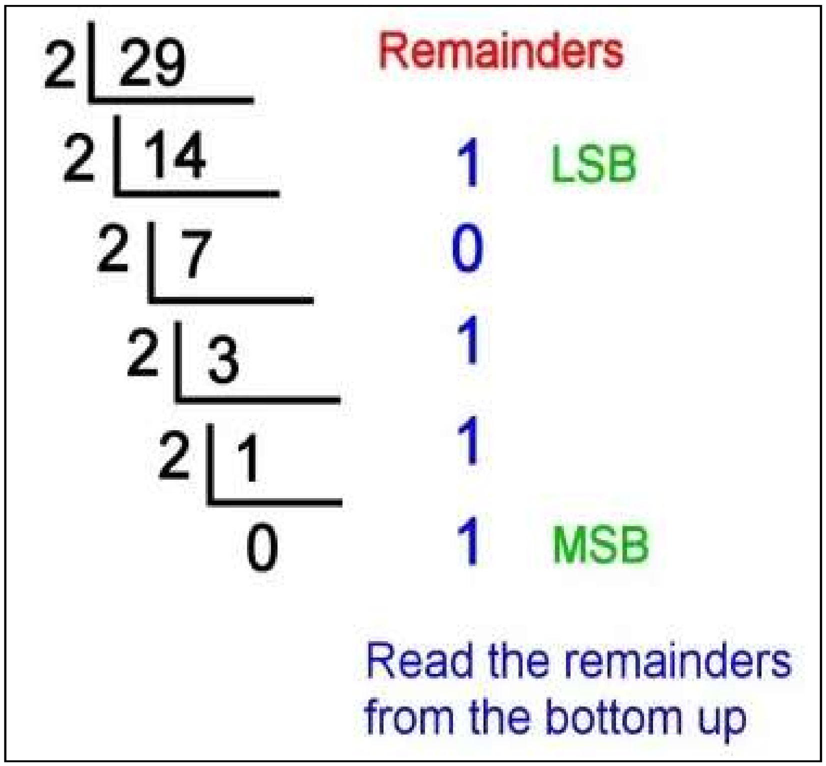
What is the decimal equivalent of the binary number 1101110?

$$
\begin{aligned}
& 1 \times 2^6 = 1 \times 64 = 64 \\
+ & 1 \times 2^5 = 1 \times 32 = 32 \\
+ & 0 \times 2^4 = 0 \times 16 = 0 \\
+ & 1 \times 2^3 = 1 \times 8 = 8 \\
+ & 1 \times 2^2 = 1 \times 4 = 4 \\
+ & 1 \times 2^1 = 1 \times 2 = 2 \\
+ & 0 \times 2^0 = 0 \times 1 = 0 \\
& = (110)_{10}
\end{aligned}
$$

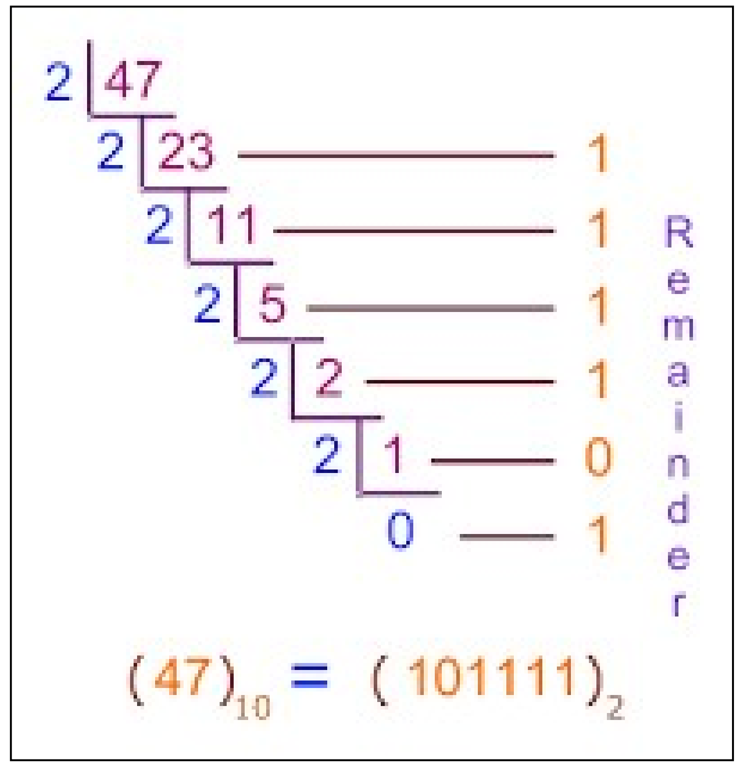
Decimal-to-Binary Conversion

- Divide the number by 2 and write down the remainder.
- Continue the process until the whole-number quotient is 0.
- Examples:

$(29)_{10} = (11101)_2$



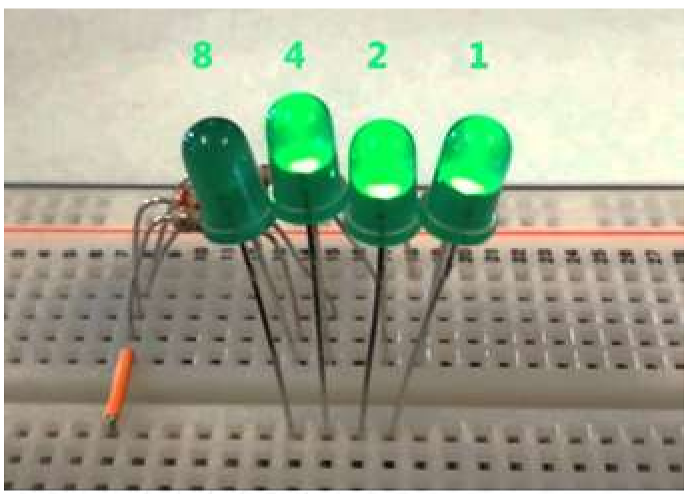
$(47)_{10} = (101111)_2$

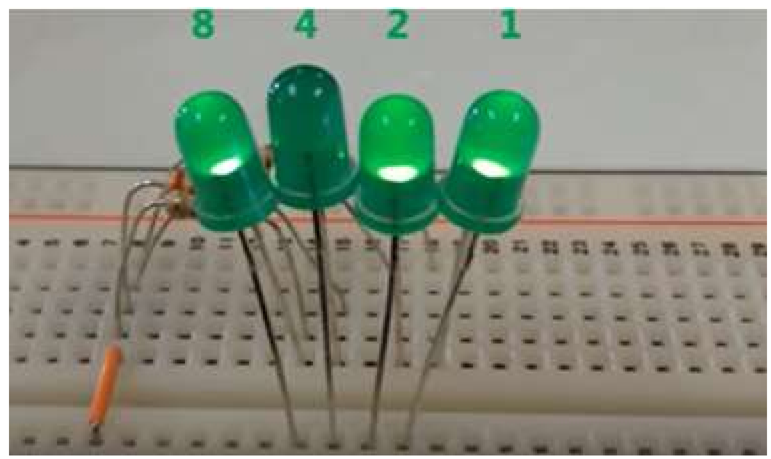


Binary Number in LAB

Low Voltage = 0
High Voltage = 1

- Using LED, each LED is representing a bit.
- LED is ON means 1, and if LED is OFF means 0
- $(0111)_2 = (7)_{10}$ $(1011)_2 = (11)_{10}$





The Byte, Nibble, and Word

- 1 byte = 8 bits
- 1 nibble = 4 bits
- 1 word = size depends on data pathway size.
  - Word size in a simple system may be one byte (8 bits)
  - Word size in a PC is eight bytes (64 bits)

Binary Fractions
(not required in the exam)

- Similar to decimal fractions, binary numbers can also be represented as fractional numbers by placing the binary digits to the right of the binary point.
- Thus all the fractional digits to the right of the binary point have respective weightings which are negative powers of two, creating a binary fraction.

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

4. Binary Arithmetic

- Binary Addition

Remember that there are only 2 digit symbols in binary, 0 and 1 , the bit by bit addition results are as below:

0 + 0 is 0 with a carry 0
1 + 0 is 1 with a carry 0
1 + 1 is 0 with a carry 1
1 + 1 + 1 is 1 with a carry 1

Example: Find the result of adding 1010111 to 1001011

```
1 0 1 1 1 1 1      <-- Carry Values
  1 0 1 0 1 1 1
 +1 0 0 1 0 1 1
 ---------------
1 0 1 0 0 0 1 0
```

• Binary subtraction

The bit by bit addition results are as below:

<u>Subtraction Table</u>

```
                  10
   0              0̸            1          1
-  0           -  1         -  0       -  1
   0              1            1        0
```

Need to "Borrow" from a more significant bit

Example: Find the result of 1010 - 0011

```
     1  10
  Ø 10  Ø 10     <-- Borrow Values
  1  0  1  0
  0  0  1  1
-
  0  1  1  1
```

• Examples of Binary Subtraction

Example: Find the result of adding 11000001 to 100010

verification

			0	10	10	10	10	10	
193		1	1	0	0	0	0	0	1
- 34	-	0	0	1	0	0	0	1	0
159_{10}		1	0	0	1	1	1	1	1_2

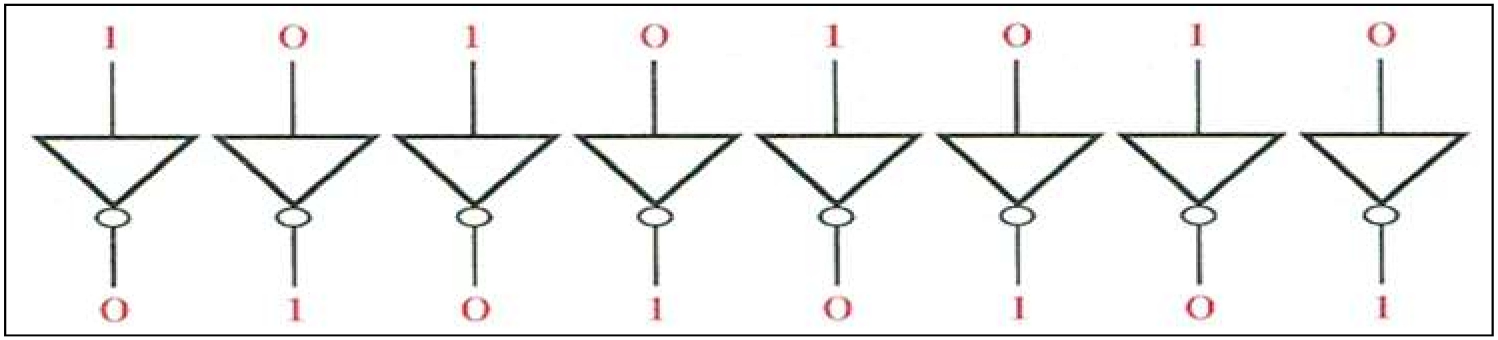
Example: Find the result of adding 11010011 to 1101110

verification

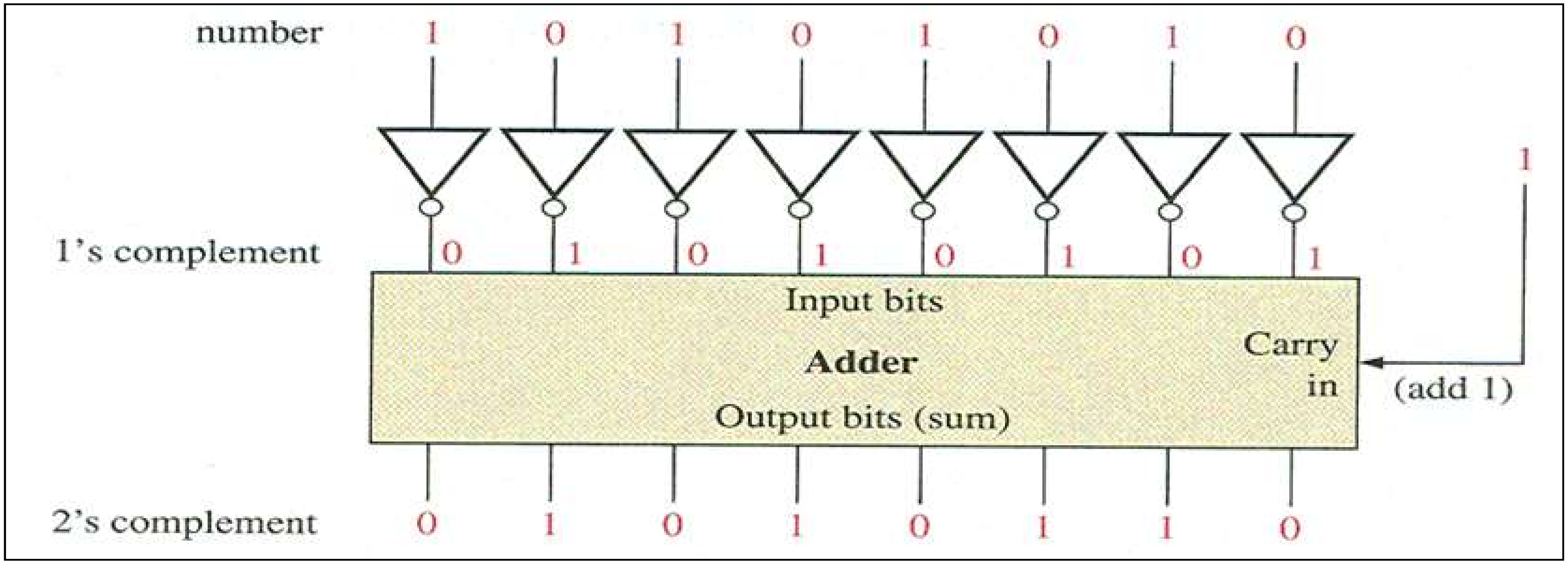
			10			1			
		0	0	10	0	10	10		
211		1	1	0	1	0	0	1	1
- 110	-	0	1	1	0	1	1	1	0
101_{10}		0	1	1	0	0	1	0	1_2

6. Complements of Binary Numbers

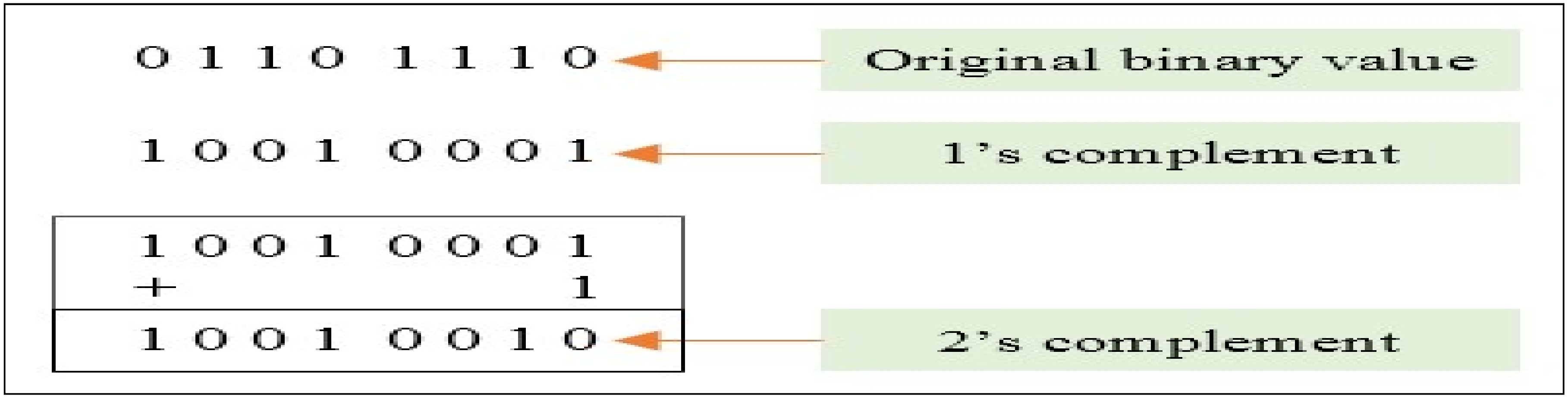
- 1's complement: Change all 1s to 0s and all 0s to 1s

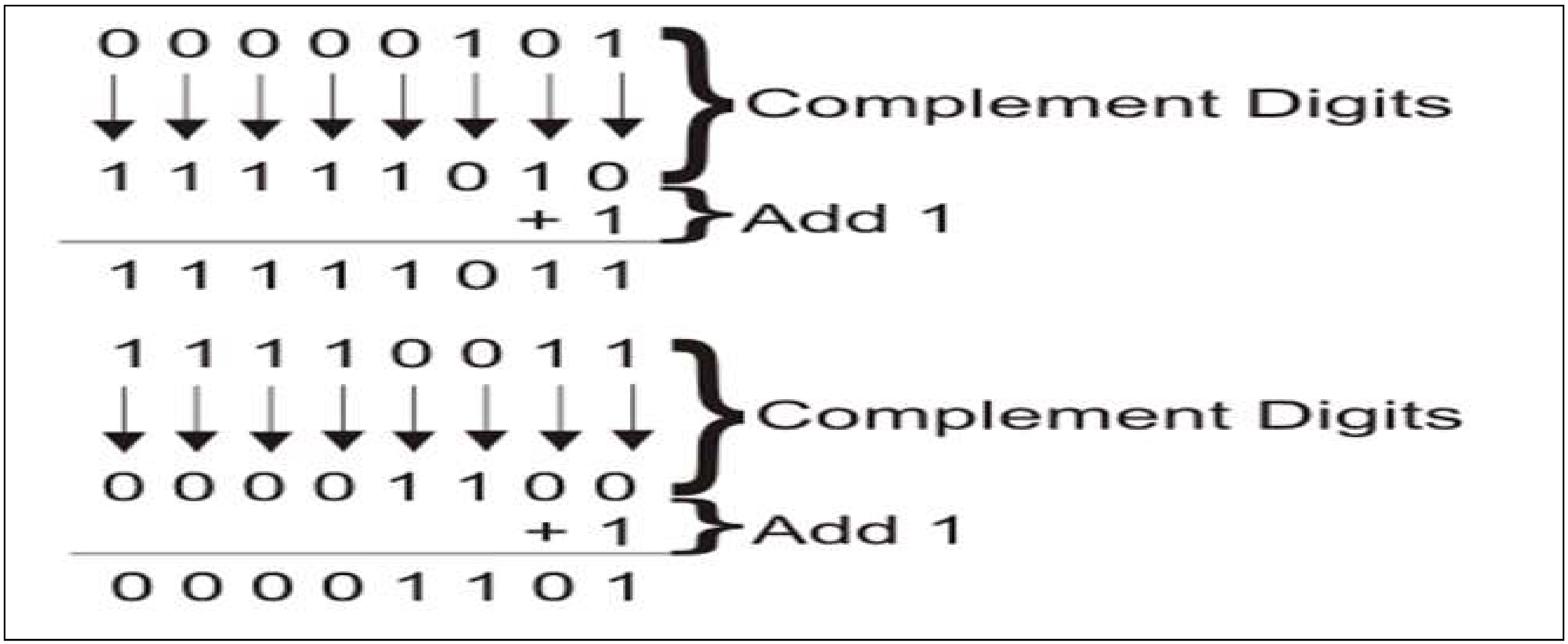


- 2's complement: Find 1's complement and then add 1



Examples of Twos-complement Evaluation





7. Signed Number

- How do we write negative binary numbers?
- Historically: Three types of Signed Numbers are there:
  - Sign-and-magnitude: Negative number has sign bit =1
  - Ones-complement: Negative number is 1’s complement
  - Twos-complement: Negative number is 2’s complement
- For all 3, the most-significant bit (MSB) is the sign digit
  - 0 ≡ positive
  - 1 ≡ negative
- twos-complement is the important one
  - Simplifies arithmetic
  - Used almost universally

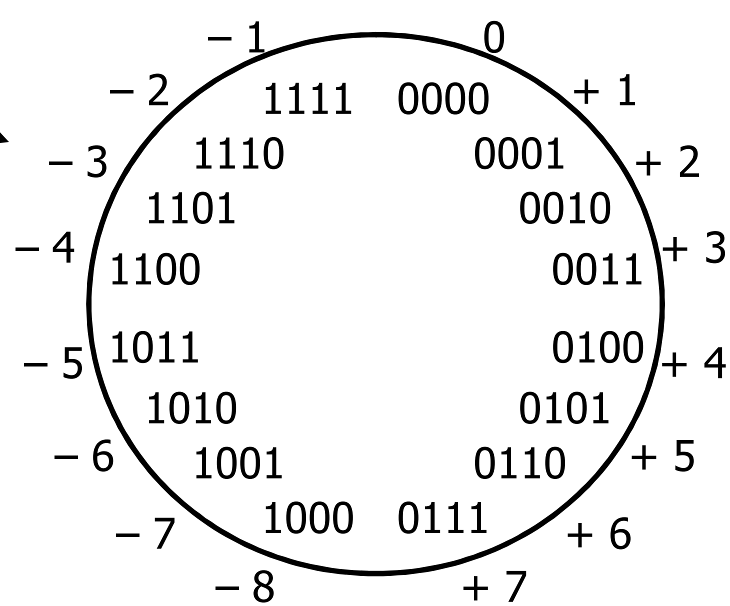
Twos-complement Method

- Negative number: Bitwise complement plus one

  $0011 \equiv 3_{10}$

  $1101 \equiv -3_{10}$

- Number wheel for 4-bit 2's complement signed binary number



- Only one zero!

- □ MSB is the sign digit
- □ 0 ≡ positive
- □ 1 ≡ negative

- Complementing a complement gives the original number

Twos-complement Method

- Arithmetic is easy
  – Subtraction = negation and addition
    - Easy to implement in hardware

Add		Invert and add		Invert and add	
4	0100	4	0100	– 4	1100
+ 3	+ 0011	– 3	+ 1101	+ 3	+ 0011
= 7	= 0111	= 1	1 0001	– 1	1111
		drop carry	= 0001		

Signed Numbers Ranges

- Range of Values

Total combinations = 2^n

2's complement form:

$$-(2^{n-1}) \text{ to } +(2^{n-1}-1)$$

Range for 8 bit number:

n = 8

$-(2^{8-1}) = -2^7 = -128$ minimum

$+(2^{8-1}) - 1 = +2^7 - 1 = +127$ maximum

Total combination of numbers is 2^8 = 256.

8 bit examples:

10000000 =	**-128**	**10000001 =**	**-127**
11111111 =	**-1**	**01111111 =**	**+127**

Examples of Signed Numbers Conversions

```
  5 = 0 0 0 0 0 1 0 1  }
      ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓  } Complement Digits
      1 1 1 1 1 0 1 0  }
                  + 1  } Add 1
      ---------------
 -5 = 1 1 1 1 1 0 1 1

-13 = 1 1 1 1 0 0 1 1  }
      ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓  } Complement Digits
      0 0 0 0 1 1 0 0  }
                  + 1  } Add 1
      ---------------
 13 = 0 0 0 0 1 1 0 1
```

Examples of Singed Numbers Conversions

What is the binary representation of -105_{10} in 8 bits?

2	105		
2	52	1	LSB
2	26	0	
2	13	0	
2	6	1	
2	3	0	
2	1	1	
	0	1	MSB

$$(105)_{10} = (01101001)_2$$

To obtain the binary representation of a negative number we must flip all the bits of the positive representation and add 1:

$$\begin{array}{r} 10010110 \\ +\ 00000001 \\ \hline 10010111 \end{array}$$

Thus:

$-105_{10} = 10010111_2$

Examples of Singed Numbers Conversions

Determine the decimal values of the signed binary numbers expressed in 2's complement:

(a) 01010110 **(b)** 10101010

(a) The bits and their powers-of-two weights for the positive number are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	1	1	0

Summing the weights where there are 1s,

$$64 + 16 + 4 + 2 = \mathbf{+86}$$

(b) The bits and their powers-of-two weights for the negative number are as follows. Notice that the negative sign bit has a weight of $-2^7 = -128$.

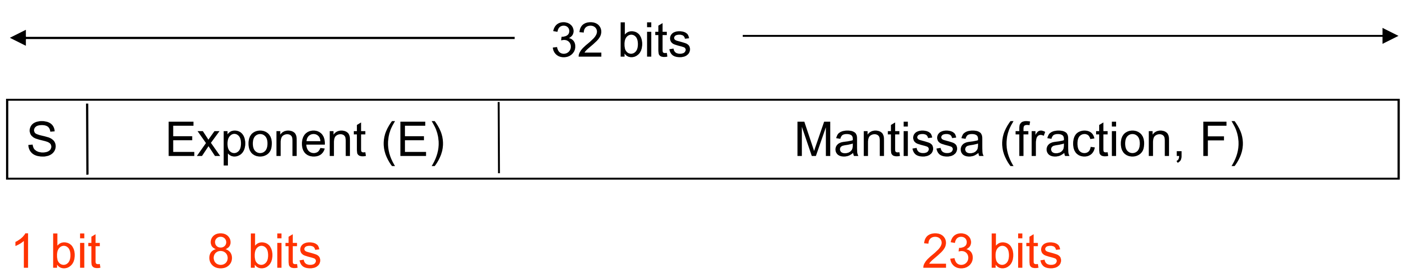
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

Summing the weights where there are 1s,

$$-128 + 32 + 8 + 2 = \mathbf{-86}$$

Floating-Point Numbers in Computer (not required in the exam)

- Floating-point numbers
  - Can represent very large or very small numbers based on scientific notation. Binary point “floats”.
- Two Parts
  - **Mantissa** represents magnitude of number
  - **Exponent** represents number of places that binary point is to be moved
- Example of Floating-point numbers forms
  - Single-precision (32 bits) float
  - Double-precision (64 bits) double



8. Hexadecimal Numbers

- Decimal, binary, and hexadecimal numbers
- 4 bits is a nibble
- $FF_{16} = 255_{10}$

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

Binary-to-Hexadecimal Conversion

1. Break the binary number into 4-bit groups
2. Replace each group with the hexadecimal equivalent digit

- Convert 1100101001010111 to Hex

C A 5 7 = $CA57_{16}$

- Convert $10A4_{16}$ to binary

0001 0000 1010 0100 = 0001000010100100

9. Octal Numbers

- Not used as frequently
- Convert binary to octal
  - Can group in 3 bits instead of 4 bits like Hex.
  - Symbols range from 0 to 7

- Example. Convert 001011010 to octal.

  1 3 2 = 132_8

- Example. Convert 105_8 to binary.

  001 000 101 = $(1000101)_2$

Binary and Octal Conversions

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) 110101 → 6 5 = $\mathbf{65_8}$

(b) 101111001 → 5 7 1 = $\mathbf{571_8}$

(c) 100110011010 → 4 6 3 2 = $\mathbf{4632_8}$

(d) 011010000100 → 3 2 0 4 = $\mathbf{3204_8}$

10. Binary Coded Decimal (BCD)

Decimal and BCD digits

DECIMAL DIGIT	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

- Applications of BCD
  - Digital clocks, digital thermometers, digital meters, and other devices with seven-segment displays typically use BCD code to simplify the displaying of decimal numbers.

Decimal to BCD

Convert each of the following decimal numbers to BCD:

(a) 35 **(b)** 98 **(c)** 170 **(d)** 2469

(a) 3 5

↓ ↓

00110101

(b) 9 8

↓ ↓

10011000

(c) 1 7 0

↓ ↓ ↓

000101110000

(d) 2 4 6 9

↓ ↓ ↓ ↓

0010010001101001

BCD to Decimal

Convert each of the following BCD codes to decimal:

(a) 10000110 (b) 001101010001 (c) 1001010001110000

Solution

(a) 10000110 → 1000 0110 → 8 6

(b) 001101010001 → 0011 0101 0001 → 3 5 1

(c) 1001010001110000 → 1001 0100 0111 0000 → 9 4 7 0