Chapter 2

Number System, Operations and Codes

1

Content

- Decimal Numbers
- Binary Numbers
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Binary Arithmetic
- 1'S AND 2'S Complement of Binary Numbers
- Signed Binary Number
- Arithmetic Operations with Signed Numbers
- Hexadecimal Number
- Octal Number
- Binary Codded Decimal(BCD)
- Digital Code

2

Decimal Numbers

- **Decimal Number System** is Composed of **10 numerals** or **symbols**.
- These 10 symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;
- Using these 10 symbols as digits of a number, we can express any **quantity**.
- The decimal system, also called the **base-10 system** because it has 10 digits.
- The position of each digit in a decimal number indicates the **magnitude of the quantity** represented and can be assigned a **weight**.

3

Cont.

- The weights for whole numbers are **positive powers of ten** that **increase** from **right to left**, beginning with $10^0 = 1$.
- For fractional numbers, the weights are **negative powers of ten** that **decrease** from **left to right** beginning with 10^{-1}.

$$10^2\ 10^1\ 10^0.10^{-1}\ 10^{-2}\ 10^{-3}\ldots$$

Decimal point

- **The value of a decimal number** is the sum of the digits after each digit has been multiplied by its weight.

4

Cont....

- Example1 Express the decimal number 47 as a sum of the values of each digit.

Solution

The digit 4 has a weight of 10, which is 10^1, as indicated by its position. The digit 7 has a weight of 1, which is 10^0, as indicated by its position.

$$47 = (4 \times 10^1) + (7 \times 10^0)$$
$$= (4 \times 10) + (7 \times 1) = \mathbf{40 + 7}$$

- Example 2:- Express the decimal number 568.23 as a sum of the values of each digit

Solution

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

5

Binary Numbers

- The binary number system has two digits (bits).
- The two binary digits (bits) are 1 and 0.
- The position of a 1 or 0 in a binary number indicates its weight.
- The weights in a binary number are based on **powers of two**.
- The right-most bit is the ***LSB*** (least significant bit) in a binary whole number and has a weight of $2^0 = 1$.

6

Cont.

- The weights increase from right to left by a power of two for each bit.
- The left-most bit is the ***MSB*** (most significant bit).
- The left-most bit is the ***MSB*** in a ***binary fractional*** number and has a weight of 2^{-1}= 0.5.
- The fractional weights decrease from left to right by a negative power of two for each bit.
- Generally, the weight structure of a binary number is

$$2^{n-1} \ldots 2^3\ 2^2\ 2^1\ 2^0 . 2^{-1}\ 2^{-2} \ldots 2^{-n}$$

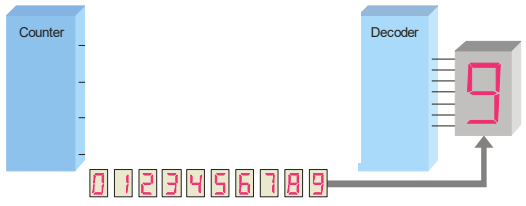
Binary point

where n is the number of bits from the binary point.

7

Binary Counting

- A binary counting sequence for numbers from zero to fifteen is shown.
- Notice the pattern of zeros and ones in each column.
- Digital counters frequently have this same pattern of digits:



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

8

Exercise

- Find the binary value of the decimal number 122

9

Binary Counting

- With n bits we can represent decimal numbers from 0 to $2^n - 1$.
- How many bits do we need to conunt(represent) N decimal numbers in a binary?

$$n \geq \log_2 N$$

  - ✓Where n is the numbers of bit needed (it is positive integre)
- How many decimal numbers can we represent with n bits binary number?

$$N = 2^n$$

10

1 =	1 =	1
10 =	2+0 =	2
11 =	2+1 =	3
100 =	4+0+0 =	4
101 =	4+0+1 =	5
110 =	4+2+0 =	6
111 =	4+2+1 =	7
1000 =	8+0+0+0 =	8
1001 =	8+0+0+1 =	9
1010 =	8+0+2+0 =	10
1011 =	8+0+2+1 =	11
1100 =	8+4+0+0 =	12
1101 =	8+4+0+1 =	13
1110 =	8+4+2+0 =	14
1111 =	8+4+2+1 =	15
10000 =	16+0+0+0+0 =	16
10001 =	16+0+0+0+1 =	17
10010 =	16+0+0+2+0 =	18
10011 =	16+0+0+2+1 =	19
10100 =	16+0+4+0+0 =	20
10101 =	16+0+4+0+1 =	21
10110 =	16+0+4+2+0 =	22
10111 =	16+0+4+2+1 =	23
11000 =	16+8+0+0+0 =	24
11001 =	16+8+0+0+1 =	25
11010 =	16+8+0+2+0 =	26
11011 =	16+8+0+2+1 =	27
11100 =	16+8+4+0+0 =	28
11101 =	16+8+4+0+1 =	29
11110 =	16+8+4+2+0 =	30
11111 =	16+8+4+2+1 =	31

11

Binary to Decimal Conversion

- The decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0.
- Example:-Convert the binary whole number 1101101 to decimal

Solution

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

Binary number: 1 1 0 1 1 0 1

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

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

12

Cont.

- Example 2:- Convert the fractional binary number 0.1011 to decimal.

Solution

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

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

- Ex :- Convert the binary number 10111101.011 to decimal.

13

Decimal to Binary Conversion

a. Sum-of-Weights Method

- One way to find the binary number that is equivalent to a given decimal number is to determine the set of binary weights whose sum is equal to the decimal number.

Example

Solution

Convert the decimal number 49 to binary.

The column weights double in each position to the right. Write down column weights until the last number is larger than the one you want to convert.

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

64 32 16 8 4 2 1.

0 1 1 0 0 0 1.

Example:- Convert the following decimal numbers to binary:

a) 12 b)25 c)58 d)72

14

Decimal to Binary Conversion

b. Repeated Division-by-2 Method

- To get the binary number for a given decimal number, divide the decimal number by 2 until the quotient is 0.
- Note the remainders form the binary number.
- The first remainder to be produced is the LSB in the binary number, and the last remainder to be produced is the MSB.
- Example:- Convert the following decimal numbers to binary:

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

15

Repeated Division-by-2 Method

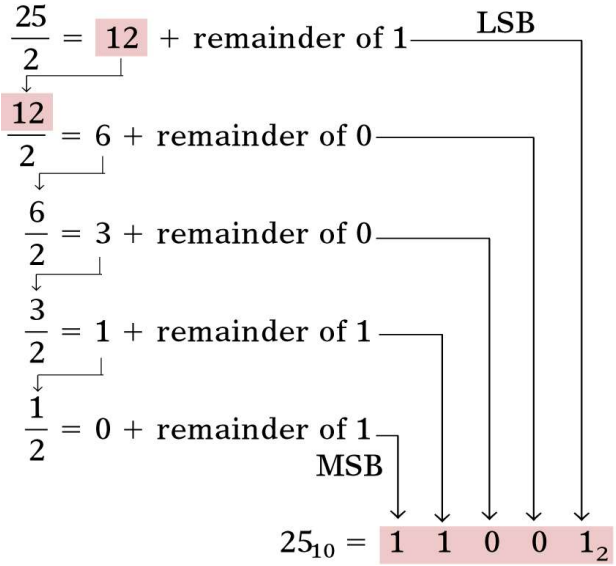
Repeated Division

Divide the decimal number by 2.

Write the remainder after each division until a quotient of zero is obtained.

The first remainder is the LSB.

The last is the MSB.



16

Converting Decimal Fractions to Binary

- ***Sum-of-Weights:***-The sum-of-weights method can be applied to fractional decimal numbers, as shown in the following example:

$$0.625 = 0.5 + 0.125 = 2^{-1} + 2^{-3} = 0.101$$

- ***Repeated Multiplication by 2 :***-Decimal fractions can be converted to binary by repeated multiplication by 2.
- 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.**

17

Converting Decimal Fractions to Binary

- 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.

Example Convert the decimal fraction 0.188 to binary by repeatedly multiplying the fractional results by 2.

Solution

0.188 x 2 = 0.376 carry = 0
0.376 x 2 = 0.752 carry = 0
0.752 x 2 = 1.504 carry = 1
0.504 x 2 = 1.008 carry = 1
0.008 x 2 = 0.016 carry = 0

MSB ↓

Answer = .00110 (for five significant digits)

18

Converting Decimal Fractions to Binary

- Example:-Convert each decimal number to binary by using the repeated multiplication-by-2 for fractions.
  a) 0.625 b) 0.375

19

Binary Arithmetic

Binary Addition

The rules for binary addition are

0 + 0 = 0	Sum = 0, carry = 0
0 + 1 = 1	Sum = 1, carry = 0
1 + 0 = 1	Sum = 1, carry = 0
1 + 1 = 10	Sum = 0, carry = 1

When an input carry = 1 due to a previous result, the rules are

1 + 0 + 0 = 01	Sum = 1, carry = 0
1 + 0 + 1 = 10	Sum = 0, carry = 1
1 + 1 + 0 = 10	Sum = 0, carry = 1
1 + 1 + 1 = 11	Sum = 1, carry = 1

20

Binary Arithmetic

Binary Addition

Example Add the binary numbers 00111 and 10101 and show the equivalent decimal addition.

Solution

$$\begin{array}{r} {\scriptstyle 0\,1\,1\,1} \\ 00111 \\ 10101 \\ \hline 11100 \end{array} \quad \begin{array}{r} 7 \\ 21 \\ \hline = 28 \end{array}$$

Exercises Add the following binary numbers

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

21

Binary Arithmetic

Binary Subtraction

The four rules for binary subtraction of bits are

$0 - 0 = 0$
$1 - 1 = 0$
$1 - 0 = 1$
$10 - 1 = 1$ 0-1,with a borrow of 1

Example Subtract the binary number 00111 from 10101 and show the equivalent decimal subtraction.

Solution

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 10101 \\ 00111 \\ \hline 01110 \end{array} \quad \begin{array}{r} 21 \\ 7 \\ \hline = 14 \end{array}$$

22

Example

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

(a)

$X =$	1010100
2's complement of $Y =$	+ 0111101
Sum =	10010001
Discard end carry $2^7 =$	− 10000000
Answer: $X - Y =$	0010001

(b)

$Y =$	1000011
2's complement of $X =$	+ 0101100
Sum =	1101111

23

Binary Arithmetic

Binary Subtraction

Exercises Perform the following binary subtraction:

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

24

Binary Arithmetic

Binary Multiplication

The four basic rules for multiplying bits are as follows:

$0 * 0 = 0$
$0 * 1 = 0$
$1 * 0 = 0$
$1 * 1 = 1$

Exercises Perform the following binary multiplications:

(a) 11 * 11 (b) 101 * 111

25

Binary Arithmetic

Binary Division

Division in binary follows the same procedure as division in decimal.

Exercises Perform the following binary divisions:

(a) 110 ÷ 11 (b) 110 ÷ 10

26

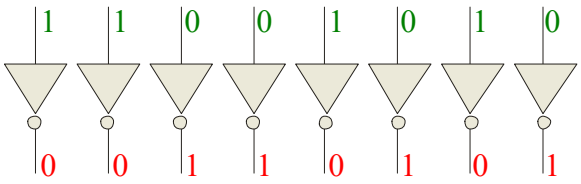
1'S AND 2'S Complement of Binary Numbers

- The 1's complement and the 2's complement of a binary number are important because they permit the representation of **negative numbers.**
- The method of 2's complement arithmetic is commonly used in computers to handle **negative numbers.**

27

1'S Complement

- The 1's complement of a binary number is just the inverse of the digits. To form the 1's complement, change all 0's to 1's and all 1's to 0's.
- For example, the 1's complement of 11001010 is 00110101
- In digital circuits, the 1's complement is formed by using inverters:



28

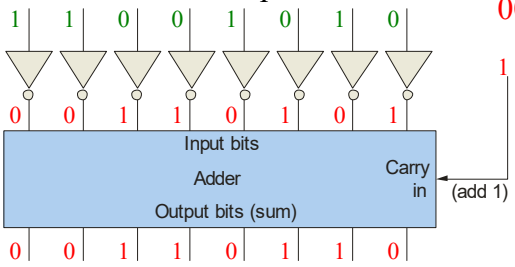
2'S Complement

The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

Recall that the 1's complement of 11001010 is 00110101 (1's complement)

To form the 2's complement, add 1:

$$\begin{array}{r} 00110101 \\ +1 \\ \hline 00110110 \end{array}$$ (2's complement)



29

2'S Complement

2's complement = (1's complement) + 1

EXAMPLE Find the 2's complement of 10110010.

30

Signed Binary Numbers

- Digital systems, such as computers, 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.

31

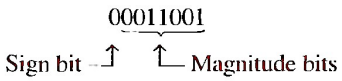
Signed Binary Numbers

- There are three forms in which signed integer (whole) numbers can be represented in binary:
    a. Sign-magnitude,
    b. 1's complement, and
    c. 2' complement.
- Of these, the 2's complement is the most important and the sign-magnitude is the least used.
- ***The Sign Bit:-*** the left-most bit in a signed binary number is the sign bit, which tells you whether the number is positive or negative.
- A 0 sign bit indicates a +ve number, and a 1 sign bit indicates a –ve number.

32

Sign-Magnitude Form

- When a signed binary number is represented in sign-magnitude, the left-most bit is the sign bit and the remaining bits are the magnitude bits.
- The magnitude bits are in true (uncomplemented) binary for both positive and negative numbers.
- For example, the decimal number + 25 is expressed as an 8-bit signed binary number using the sign-magnitude form as



The decimal number −25 is expressed as

10011001

- *Notice that the only difference between + 25 and - 25 is the sign bit.*

33

1' s Complement Form

- Positive numbers in 1's complement form are represented the same way as the positive sign-magnitude numbers.
- Negative numbers, however, are the 1's complements of the corresponding positive numbers.
- For example:- using eight bits, the decimal number -25 is expressed as the 1's complement of +25 (00011001) as 11100110

34

2's Complement Form

- Positive numbers in 2's complement form are represented the same way as in the sign-magnitude and 1's complement forms.
- Negative numbers are the 2's complements of the corresponding positive numbers.
- Again, using eight bits, let's take decimal number -25 and express it as the 2's complement of

  +25 (00011001) ⟶ 111O0111
- In the 2's complement form, a negative number is the 2's complement of the corresponding positive number.

35

Cont....

Example

Express the decimal number -39 as an 8-bit number in the

a) sign-magnitude
b) 1's complement, and
c) 2's complement forms.

36

Four-Bit Signed Binary Numbers in Sign–Magnitude, One's Complement, and Two's Complement

Binary Form	*Sign Magnitude*	*One's Complement*	*Two's Complement*
0000	+0	+0	0
0001	+1	+1	+1
0010	+2	+2	+2
0011	+3	+3	+3
0100	+4	+4	+4
0101	+5	+5	+5
0110	+6	+6	+6
0111	+7	+7	+7
1000	–0	–7	–8
1001	–1	–6	–7
1010	–2	–5	–6
1011	–3	–4	–5
1100	–4	–3	–4
1101	–5	–2	–3
1110	–6	–1	–2
1111	–7	–0	–1

37

Hexadecimal Numbers

Hexadecimal uses sixteen characters to represent numbers: the numbers 0 through 9 and the alphabetic characters A through F.

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

38

Binary-to-Hexadecimal Conversion

- Large binary number can easily be converted to hexadecimal by grouping bits 4 at a time and writing the equivalent hexadecimal character.
- Simply break the binary number into 4-bit groups, starting at the LSB and replace each 4-bit group with the equivalent hexadecimal symbol.

EXAMPLE

Solution

Convert the following binary numbers to hexadecimal:

(a) 1100101001010111 (b) 1111110001011010 01

(a) 1100 1010 0101 0111
C A 5 7 = $CA57_{16}$

(b) 0011 1111 0001 0110 1001
3 F 1 6 9 = $3F169_{16}$

39

Hexadecimal-to-Binary Conversion

- To convert from a hexadecimal number to a binary number, reverse the process and replace each hexadecimal symbol with the appropriate four bits.
- Hexadecimal is a convenient way to represent binary numbers.

EXAMPLE

Solution

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

40

Hexadecimal-to-Decimal Conversion

- One way to find the decimal equivalent of a hexadecimal number is to first convert the hexadecimal number to binary and then convert from binary to decimal.

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

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

- Another is to multiply the decimal value of each hexadecimal digit by its weight and then take the sum of these products.
- ***Example:***-Convert the following hexadecimal numbers to decimal: (a) $E5_{16}$ (b) $B2F8_{16}$

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

41

Hexadecimal Addition

- *Addition can be done directly with hexadecimal numbers as the following example*
- ***Example:-*** *Add the following hexadecimal numbers:*

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

Solution

(a)
$$\begin{array}{r} 23_{16} \\ +16_{16} \\ \hline \mathbf{39}_{16} \end{array}$$

(b)
$$\begin{array}{r} 58_{16} \\ +\ 22_{16} \\ \hline \mathbf{7A}_{16} \end{array}$$

(c)
$$\begin{array}{r} 2B_{16} \\ +\ 84_{16} \\ \hline \mathbf{AF}_{16} \end{array}$$

(d)
$$\begin{array}{r} DF_{16} \\ +\ AC_{16} \\ \hline \mathbf{18B}_{16} \end{array}$$

42

Hexadecimal Subtraction

- *Convert the hexadecimal number to binary. Take the 2's complement of the binary number. Convert the result to hexadecimal.*
- ***Example:-*** Subtract the following hexadecimal numbers:

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

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

2's complement of $2A_{16} = 11010110 = D6_{16}$

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

43

OCTAL NUMBERS

- *The octal number system is composed of eight digits.*
- *These are 0, 1, 2, 3, 4, 5, 6, 7.*
- *The octal number system has a base of 8.*

Octal-to-Decimal Conversion

- The evaluation of an octal number in terms of its decimal equivalent is accomplished by multiplying each digit by its weight and summing the products.
- *Example: Convert $(127.4)_8$ to Decimal*

Solution:

$$(127.4)_8 = 1x8^2 + 2x8^1 + 7x8^0 + 4x8^{-1} = (87.5)_{10}$$

44

Convert $(0.513)_{10}$ to octal.

$$0.513 \times 8 = 4.104$$
$$0.104 \times 8 = 0.832$$
$$0.832 \times 8 = 6.656$$
$$0.656 \times 8 = 5.248$$
$$0.248 \times 8 = 1.984$$
$$0.984 \times 8 = 7.872$$

The answer, to seven significant figures, is obtained from the integer part of the products:

$$(0.513)_{10} = (0.406517\ldots)_8$$

The conversion of decimal numbers with both integer and fraction parts is done by converting the integer and the fraction separately and then combining the two answers.

$$(41.6875)_{10} = (101001.1011)_2$$

we have

$$(153.513)_{10} = (231.406517)_8$$

45

Octal-to-Binary Conversion

- *Because each octal digit can be represented by a 3-bit binary number, it is very easy to convert from octal to binary.*

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

46

OCTAL TO BINARY

- *To convert an octal number to a binary number, simply replace each octal digit with the appropriate three bits.*
- *Example:-*
- *Convert each of the following octal numbers to binary:*

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

Solution

(a) 1 3 → 001011

(b) 2 5 → 010101

(c) 1 4 0 → 001100000

(d) 7 5 2 6 → 111101010110

47

Binary-to-Octal Conversion

- Conversion of a binary number to an octal number is the reverse of the octal-to-binary conversion.
- Start with the right-most group of three bits and, moving from right to left, convert each 3-bit group to the equivalent octal digit.
- Example:-Convert each of the following binary numbers to octal:

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

Solution

(a) 110101 → 6 5 = 65_8

(b) 101111001 → 5 7 1 = 571_8

(c) 100110011010 → 4 6 3 2 = 4632_8

(d) 011010000100 → 3 2 0 4 = 3204_8

48

Exercise

. Perform each of the following conversions.

(l) $38{,}187_{10}$ = _____ 16
(m) 865_{10} = _____ (BCD)
(n) 100101000111 (BCD) = _____ 10
(o) 465_8 = _____ 16
(p) $B34_{16}$ = _____ 8
(q) 01110100 (BCD) = _____ 2
(r) 111010_2 = _____ (BCD)

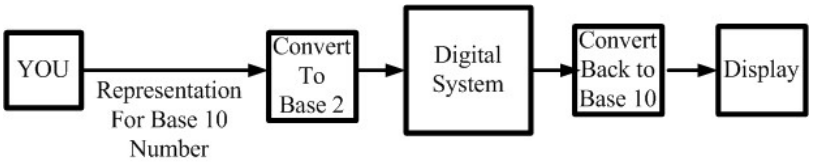
Decode the following ASCII coded message:

1001000 1100101 1101100 1101100 1101111 0101110
0100000 1001000 1101111 1110111 0100000 1100001
1110010 1100101 0100000 1111001 1101111 1110101
0111111

49

Human Perception

- We naturally live in a base 10 environment(decimal number system)
- Computer exist in a base 2 environment(binary number system)
- So give the computer/digital system the task of doing the conversions for us.



50

Binary Codes

- "An *n*-bit *binary code* is a group of *n* bits that assume up to 2^n distinct combinations of 1s and 0s, with each combination representing one element of the set being coded"
- For the 10 digits need a 4 bit code.
- The four bits code representation of decimal number is called **Binary Coded Decimal (BCD)**

51

Binary Coded Decimal (BCD)

- **Binary coded decimal** means that each decimal digit, **0 through 9**, is represented by a **binary code of four bits**.
- Binary coded decimal (BCD) is a **weighted code** that is commonly used in digital systems when it is necessary to show decimal numbers such as in **clock displays**.
- The **8421** code is a type of BCD (binary coded decimal) code.
- The designation 8421 indicates the binary weights of the four bits (2^3 , 2^2 , 2^1 ,2^0)

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

- **Invalid Codes**:-The six code combinations that are not used in BCD
  - 1010, 1011, 1100, 1101, 1110, and 1111:-are invalid in the 8421 BCD code..

52

BCD

- The table illustrates the difference between straight binary and BCD.
- BCD represents each decimal digit with a 4-bit code.
- Notice that the codes 1010 through 1111 are not used in BCD.

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

53

Decimal to BCD

- To express any decimal number in BCD, simply replace each decimal digit with the appropriate 4-bit code, as shown by Example:
- Example:-Convert each of the following decimal numbers to BCD: (a) 35 (b)98 (c) 170 (d)2469

Solution (a) 3 5 → 0011 0101 (b) 9 8 → 1001 1000

(c) 1 7 0 → 0001 0111 0000 (d) 2 4 6 9 → 0010 0100 0110 1001

54

BCD to Decimal

- It is equally easy to determine a decimal number from a BCD number.
- Start at the right most bit and break the code into groups of four bits.
- Then write the decimal digit represented by each 4-bit group.
- **Example:-**Convert each of the following BCD codes to decimal:

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

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

8 6 3 5 1 9 4 7 0

55

Digital Codes

- Many specialized codes are used in digital systems.
- You have just learned about the BCD code; now let's look at a few others.
- Some codes are strictly **numeric**, like BCD, and others are **alphanumeric**; that is, they are used to **represent numbers, letters, symbols, and instructions.**
- The codes introduced in this section are
  - The Gray code
  - The ASCII code and

56

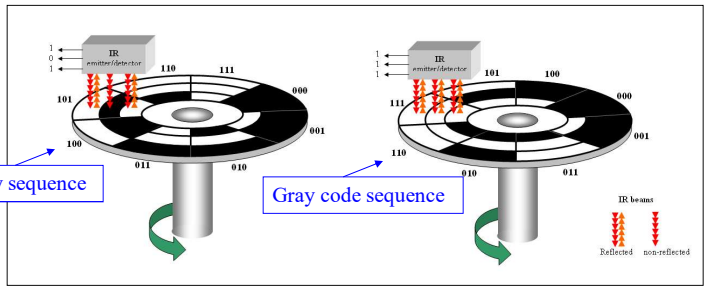
Gray Code

- ***The Gray code*** is unweighted and is not an arithmetic code; that is, there are no specific weights assigned to the bit positions.
- The important feature of the Gray code is that it exhibits ***only a single bit change*** from one code word to the next in sequence.
- Like binary numbers, the Gray code can have ***any number*** of bits. Notice the single-bit change between successive Gray code words.
- Gray code is used to **avoid problems** in systems where an error can occur **if more than one bit changes at a time**.
- This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

57

Gray Code

A shaft encoder is a typical application. Three IR emitter/detectors are used to encode the position of the shaft. The encoder on the left uses binary and can have three bits change together, creating a potential error. The encoder on the right uses gray code and only 1-bit changes, eliminating potential errors.



58

Gray Code

The following table is listing of the 4-bit Gray code for decimal numbers 0,1,2,…15.

Decimal	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

59

One-bit Gray code	Two-bit Gray code		Three-bit Gray code		Four-bit Gray code
0	0	00	00	000	000	0000
1	1	01	01	001	001	0001
	1	11	11	011	011	0011
	0	10	10	010	010	0010
			10	110	110	0110
			11	111	111	0111
			01	101	101	0101
			00	100	100	0100
					100	1100
					101	1101
					111	1111
					110	1110
					010	1010
					011	1011
					001	1001
					000	1000

60

Binary-to-Gray Code Conversion

- The following rules explain how to convert from a binary number to a Gray code word:
  1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.
  2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

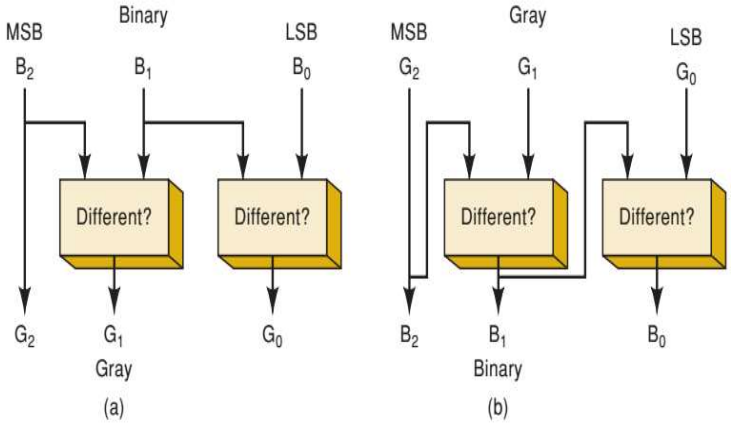
61

Gray-to-Code Binary Conversion

- Gray-to-Binary Conversion to convert from Gray code to binary, use a similar method; however, there are some differences.
- The following rules apply:
  1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.
  2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

62

a) Binary to Gray b) Gray to Binary



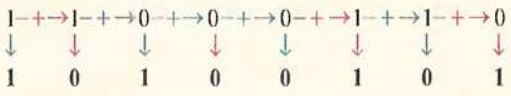
63

Cont....

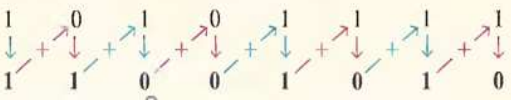
EXAMPLE (a) Convert the binary number 11000110 to Gray code.

(b) Convert the Gray code 10101111 to binary.

Solution (a) Binary to Gray code:



(b) Gray code to binary:



64

Decimal	Binary	Hexadecimal	BCD	GRAY
0	0	0	0000	0000
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

65

Alphanumeric Codes

- Alphanumeric codes are codes that represent **numbers** and **alphabetic characters** (letters).
- Most such codes, however, also represent other characters such as **symbols** and various **instructions** necessary for conveying information that are found on a computer keyboard.
- The **ASCII** is the most common alphanumeric code.
- ***ASCII:-***is the abbreviation for **American Standard Code for Information Interchange.**

66

ASCII

- ASCII is a universally accepted alphanumeric code used in most computers and other electronic equipment.
- Most computer keyboards are standardized with the ASCII.
- When you enter a letter, a number, or control command, the corresponding ASCII code goes into the computer.
- ASCII has **128 characters and symbols** represented by a **7-bit binary code**.
- Actually, ASCII can be considered an 8-bit code with the MSB always 0. This 8-bit code is 00 through 7F in hexadecimal.
- The following table is a listing of partial ASCII code showing the decimal, hexadecimal, and binary representations for each character and symbol.

67

Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal
NUL (null)	0	0	Space	20	32	@	40	64	`	60	96
Start Heading	1	1	!	21	33	A	41	65	a	61	97
Start Text	2	2	"	22	34	B	42	66	b	62	98
End Text	3	3	#	23	35	C	43	67	c	63	99
End Transmit.	4	4	$	24	36	D	44	68	d	64	100
Enquiry	5	5	%	25	37	E	45	69	e	65	101
Acknowlege	6	6	&	26	38	F	46	70	f	66	102
Bell	7	7	`	27	39	G	47	71	g	67	103
Backspace	8	8	(	28	40	H	48	72	h	68	104
Horiz. Tab	9	9	)	29	41	I	49	73	i	69	105
Line Feed	A	10	*	2A	42	J	4A	74	j	6A	106
Vert. Tab	B	11	+	2B	43	K	4B	75	k	6B	107
Form Feed	C	12	,	2C	44	L	4C	76	l	6C	108
Carriage Return	D	13	-	2D	45	M	4D	77	m	6D	109
Shift Out	E	14	.	2E	46	N	4E	78	n	6E	110
Shift In	F	15	/	2F	47	O	4F	79	o	6F	111
Data Link Esc	10	16	0	30	48	P	50	80	p	70	112
Direct Control 1	11	17	1	31	49	Q	51	81	q	71	113
Direct Control 2	12	18	2	32	50	R	52	82	r	72	114
Direct Control 3	13	19	3	33	51	S	53	83	s	73	115
Direct Control 4	14	20	4	34	52	T	54	84	t	74	116
Negative ACK	15	21	5	35	53	U	55	85	u	75	117
Synch Idle	16	22	6	36	54	V	56	86	v	76	118
End Trans Block	17	23	7	37	55	W	57	87	w	77	119
Cancel	18	24	8	38	56	X	58	88	x	78	120
End of Medium	19	25	9	39	57	Y	59	89	y	79	121
Substitue	1A	26	:	3A	58	Z	5A	90	z	7A	122
Escape	1B	27	;	3B	59	[	5B	91	{	7B	123
Form separator	1C	28	<	3C	60	\	5C	92	|	7C	124
Group separator	1D	29	=	3D	61	]	5D	93	}	7D	125
Record Separator	1E	30	>	3E	62	^	5E	94	~	7E	126
Unit Separator	1F	31	?	3F	63	_	5F	95	Delete	7F	127

68

Cont....

- Example:- The following is a message encoded in ASCII code. What is the message? 1001000 1000101 1001100 1010000
- Solution:- Convert each seven-bit code to its hex equivalent.
- The results are 48 45 4C 50
- Now locate these hex values in Table above and determine the character represented by each. The results are **HELP**

- *Example2: write your full name using ASCII Code.*
- *Example3.* Decode the following ASCII coded message:

1001000 1100101 1101100 1101100 1101111 0101110
0100000 1001000 1101111 1110111 0100000 1100001
1110010 1100101 0100000 1111001 1101111 1110101
0111111

69

Extended ASCII

- In 1981, IBM introduced extended ASCII, which is an 8-bit code and increased the character set to 256.
- Other extended sets (such as Unicode) have been introduced to handle characters in languages other than English.

70

Parity Method

- The parity method is a method of **error detection** for simple transmission errors involving one bit (or an odd number of bits).
- A parity bit is an "extra" bit attached to a group of bits to force the number of 1's to be either even (even parity) or odd (odd parity).

Example The ASCII character for "a" is 1100001 and for "A" is 1000001. What is the correct bit to append to make both of these have odd parity?

Solution The ASCII "a" has an odd number of bits that are equal to 1; therefore the parity bit is 0. The ASCII "A" has an even number of bits that are equal to 1; therefore the parity bit is 1.

71