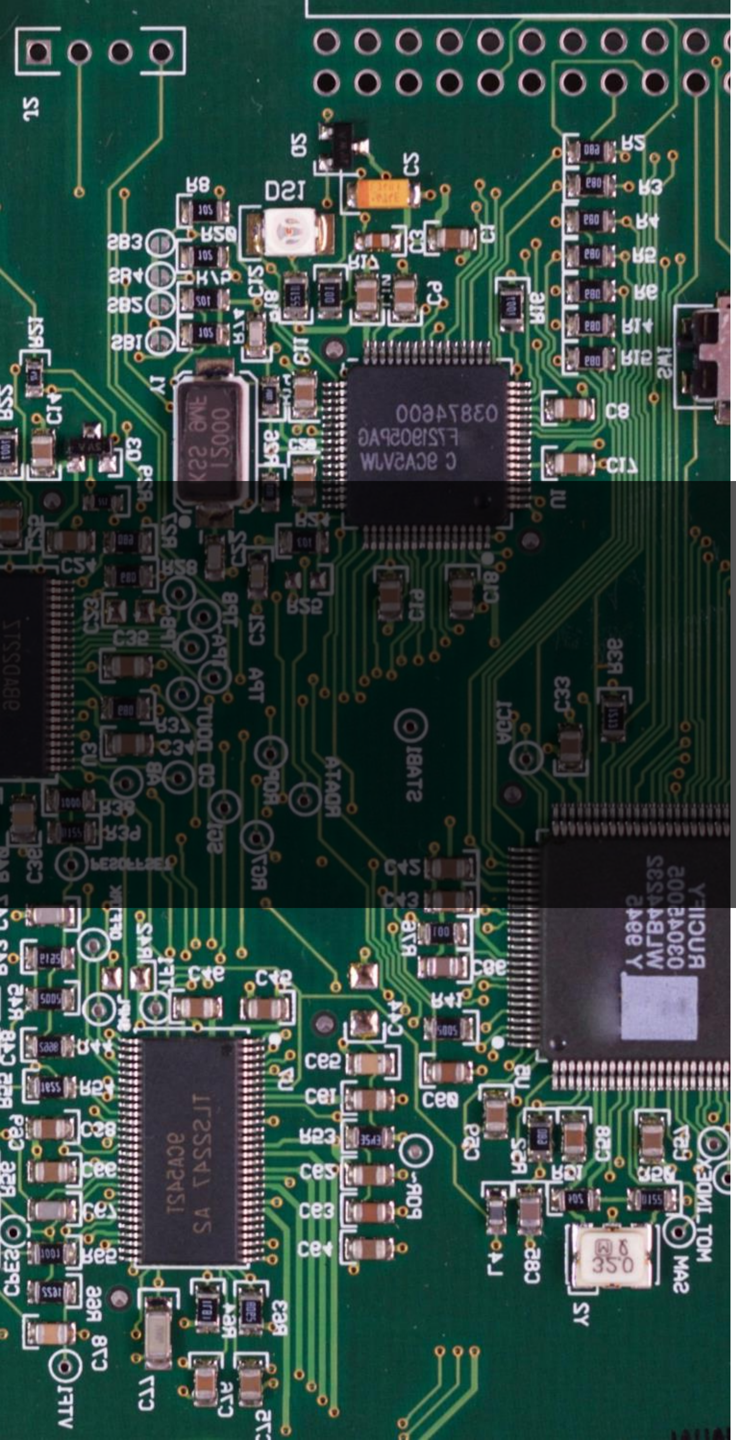


University of Al Maarif
Department of Medical Instruments Techniques Engineering
Class: 2nd

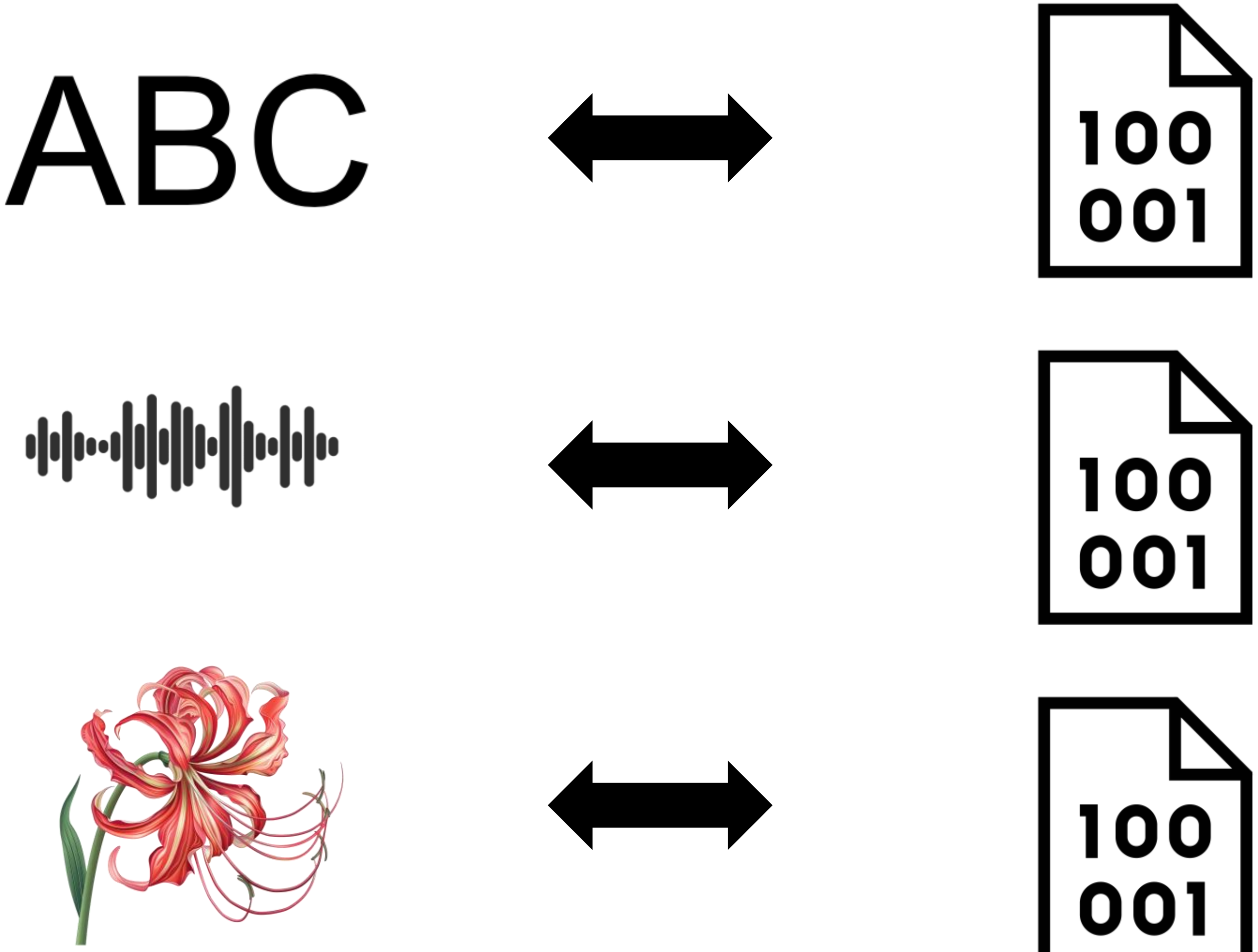
Digital Electronics

MIET2203

Lecture 2: Binary codes

Lecturer: Mohammed Jabal
2024 - 2025

Objective



A. Binary-coded-decimal code (BCD)

- It is a form of binary encoding where **each digit** in a decimal number is represented in the form of bits. This encoding can be done in either **4-bit** or **8-bit** (usually 4-bit is **preferred**).

- The BCD code is more precisely known as **8421 BCD** code , with 8,4,2 and 1 representing the weights of different bits in the **four-bit** groups, Starting from **MSB** and proceeding towards **LSB**. This feature makes it a **weighted code**, which means that each bit in the four-bit group representing a given decimal digit has **an assigned weight**.

$$(2^3, 2^2, 2^1, 2^0)$$

A. Binary-coded-decimal code (BCD)

All you have to remember are the ten binary combinations that represent the ten decimal digits as shown in Table:

Decimal/BCD conversion.

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

A. Binary-coded-decimal code (BCD)

- Many decimal values, have an infinite place-value representation in binary but have a finite place-value in binary coded decimal. For example, 0.2 in binary is (0.001100…) and in BCD is (0.0010). It avoids fractional errors and is also used in huge financial calculations.

- For example $(15)_{10}$ to Binary is $(1111)_2$

But the BCD is

1 Is 0001

5 Is 0101

15 Is 0001 0101

A. Binary-coded-decimal code (BCD)

EXAMPLE 2–33

Convert each of the following decimal numbers to BCD:

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

Solution

(a) 3 → 0011, 5 → 0101: **00110101**

(b) 9 → 1001, 8 → 1000: **10011000**

(c) 1 → 0001, 7 → 0111, 0 → 0000: **000101110000**

(d) 2 → 0010, 4 → 0100, 6 → 0110, 9 → 1001: **0010010001101001**

Related Problem

Convert the decimal number 9673 to BCD.

A. Binary-coded-decimal code (BCD)

EXAMPLE 2–34

Convert each of the following BCD codes to decimal:

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

Solution

(a) 1000 0110 → **8** **6**

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

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

Related Problem

Convert the BCD code 10000010001001110110 to decimal.

B. Excess - 3 - code

- It is performed in the same manner as BCD except that 3 is added to each decimal digit before encoding it in binary. The following table shows this code.

- Excess-3 code is also called as XS-3 code. It is a type of non-weighted code used to express decimal numbers. Excess-3 code words are derived from the 8421 BCD code words adding $(0011)_2$ or $(3)_{10}$ to each code word in 8421.

B. Excess - 3 - code

Decimal	Binay (BCD) 8 4 2 1	Excess-3 Code
0	0 0 0 0	0011
1	0 0 0 1	0100
2	0 0 1 0	0101
3	0 0 1 1	0110
4	0 1 0 0	0111
5	0 1 0 1	1000
6	0 1 1 0	1001
7	0 1 1 1	1010
8	1 0 0 0	1011
9	1 0 0 1	1100

C. Gray code

- The Gray code belongs to a class of codes called **minimum change** codes, in which only from one step to the next. The following table shows this code.

- Gray codes are a type of **non-weighted** code. They are not arithmetic codes, which means there are no specific **weights assigned** to the bit position.

- Gray codes are popularly used in Shaft Position encoders. A shaft position encoder produces a code word that represents the angular position of the shaft.

- Gray codes are **cyclic** codes and they cannot be used in arithmetic operation.

C. Gray code

- Gray codes have a very special feature that, only **one bit** will change each time the **decimal number** is incremented (see the figure below). As only one-bit changes at a time, gray codes are also known **unit distance** code.

Four-bit Gray code.

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

C. Gray code

- **Binary-to-Gray Code Conversion**

Conversion between binary code and Gray code is sometimes useful. The following rules explain how to convert from a binary number to a Gray code word:

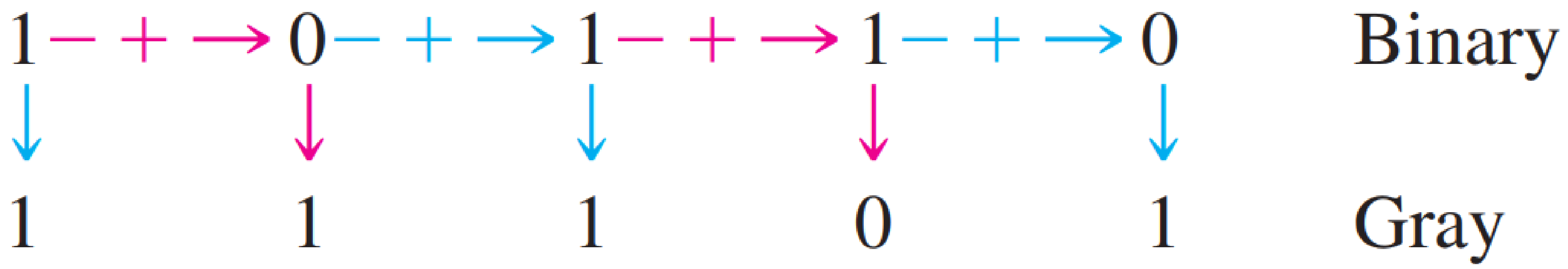
1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.

2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

C. Gray code

- **Binary-to-Gray Code Conversion**

For example, the conversion of the binary number (10110) to Gray code is as follows:



The Gray code is (11101).

C. Gray code

- **Gray-to-Binary Code Conversion**

To convert from Gray code to binary, use a similar method; however, there are some differences. The following rules apply:

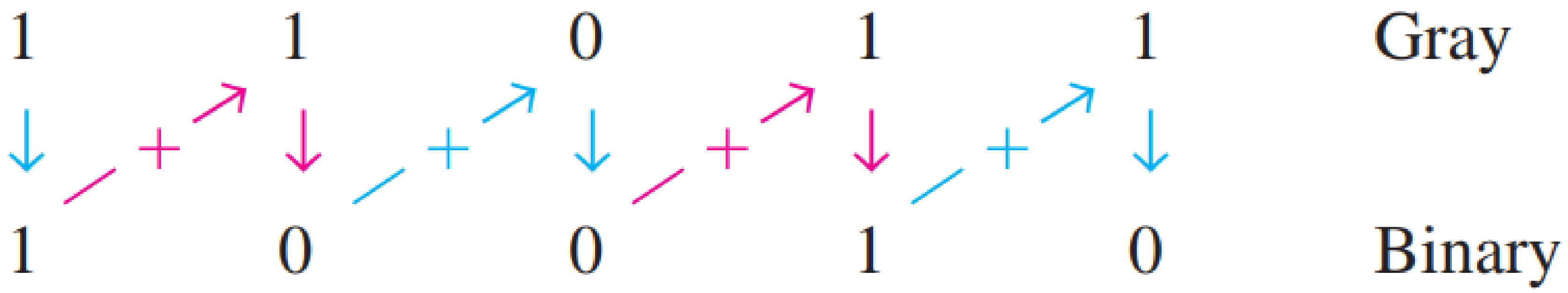
1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.

2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

C. Gray code

- **Gray-to-Binary Code Conversion**

For example, the conversion of the Gray code word (11011) to binary is as follows:



The binary number is (10010).

Boolean Algebra

- **Boolean algebra differs in a major way from ordinary algebra in that Boolean constants and variables are allowed to have only two possible values, 0 or 1. Therefore, Boolean algebra is relatively easy to work with as compared to ordinary algebra.**

Sum-of-Products Form

- **What does the sum-of-products form mean?**

First let us review products in Boolean algebra. A product of two or more variable or their complements is simply the AND function of these variables. The product of two variables can be expressed as AB, the product of three variables as ABC, the product of four variables as ABCD. Recall that a sum in Boolean algebra is the same as OR function, so a sum-of-products expression is two or more AND functions ORed together. For instance , AB+CD is a sum-of-products expression.

AB+BCD

ABC+DEF

- **A sum-of products form can also contain a term with a single variable, such as A+BCD+EFG**

Product-of-sums forms

- The product-of-sums form can be thought of as the dual of the sum-of-products. It is , in terms of logic functions, the **AND** of two or more **OR** functions. For instance,(A+B)(B+C) is a product-of-sum expression
(A+B)(B+C+D)
(A+B+C)(D+E+F)

- A product-of-sums expression can also contain a **single variable** term such as **A**(B+C+D)(E+F+G).

References

[1] Digital fundamentals / Thomas L. Floyd. —Eleventh edition.

[2]