Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

NAME OF STUDENT: **Roll No:**

ROLL NUMBER:

Class & Division: S.Y.-ECE-XIII

EXPT. NO: **Date of Performance:** **Date of Submission:**

TITLE : Study of code converter circuits.

PRE-REQUISITE:

To design and implement code converter circuits such as Binary to Gray, Gray to Binary, BCD to Excess-3, and Excess-3 to BCD, students must have a foundational understanding of digital electronics. This includes knowledge of number systems (binary, decimal, BCD, Gray code, etc.), basic Boolean algebra, logic gates (AND, OR, NOT, XOR), and combinational logic design principles. Familiarity with truth tables, Karnaugh maps (K-maps) for logic simplification, and the use of adders and subtractors in circuit design is essential. Additionally, prior hands-on experience with digital circuit simulation tools (like Proteus or Multisim) and breadboard-level implementation will help in practically realizing these logic circuits.

OBJECTIVE :

The objective of this experiment is to understand the design and working principles of various digital code converter circuits using basic logic gates. It aims to enable students to analyze and implement 4-bit Binary to Gray code and Gray to Binary code converters, which are essential in minimizing errors in digital communication systems. Additionally, students will design and implement 4-bit BCD to Excess-3 and Excess-3 to BCD code converters to gain insight into arithmetic code manipulation and decimal data encoding techniques. This experiment strengthens the foundational knowledge of combinational logic design and its real-time applications in digital systems.

APPARATUS: Digital Trainer Kit, Logic gates (IC), Connecting wires.

THEORY:

1. Types of Codes

In digital electronics, *codes* are systematic arrangements of binary digits (bits) to represent numbers, characters, or symbols. Codes help in data representation, storage, and communication.

A. Weighted Codes

Each bit position has a fixed weight, and the value is obtained by summing the weights of bits that are 1. Examples:

1. **Binary (Pure Binary)** – Standard positional number system (weights: 8, 4, 2, 1 for 4-bit numbers).
2. **BCD (Binary Coded Decimal)** – Each decimal digit is represented by its 4-bit binary equivalent.

B. Non-Weighted Codes

Bits have no positional weights; codes are designed for specific applications like error detection or minimization of bit changes.

Examples:

1. **Gray Code** – Consecutive values differ in only one bit; minimizes switching errors.
2. **ASCII (American Standard Code for Information Interchange)** – Represents text characters using 7 or 8 bits.
3. **Excess-3 Code** – Non-weighted variant of BCD obtained by adding 3 to each decimal digit in BCD form.

C. Error Detecting and Correcting Codes

Designed to detect or correct errors during transmission/storage.

Examples:

1. **Parity Codes** – Add parity bits to check even/odd number of 1s.
2. **Hamming Code** – Detects and corrects single-bit errors.
3. **CRC (Cyclic Redundancy Check)** – Detects burst errors.

2. Code Conversions

Often, data needs to be converted from one code to another for processing, display, or transmission.

A. Binary ↔ Decimal

- **Binary to Decimal**: Multiply each bit by its positional weight and sum.
- **Decimal to Binary**: Repeated division by 2; remainders give binary bits (LSB to MSB).

B. Binary ↔ BCD

- **Binary to BCD**: Convert binary number to decimal, then represent each digit in 4-bit binary.
- **BCD to Binary**: Combine decimal digit values from each 4-bit group and convert to binary.

C. Binary ↔ Gray Code

- **Binary to Gray**: MSB remains same; each next Gray bit = XOR of current binary bit and the previous binary bit.
- **Gray to Binary**: MSB remains same; each next binary bit = XOR of previous binary bit and current Gray bit.

D. BCD ↔ Excess-3

- **BCD to Excess-3**: Add 3 (0011) to each 4-bit BCD digit.
- **Excess-3 to BCD**: Subtract 3 (0011) from each 4-bit digit.

E. ASCII ↔ Binary

- **ASCII to Binary**: Use ASCII table to find 7/8-bit binary representation of character.
- **Binary to ASCII**: Group bits into 7/8-bit values and look up corresponding characters.

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

I. Design and implement 4-bit Binary to Gray code converter using logic gates.

- Truth Table: -

B3	B2	B1	B0	G3	G2	G1	G0

- K-Map: -

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

- Logical Equation: -

- Logic Circuit Diagram: -

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

II. Design and implement 4-bit Gray to Binary code converter using logic gates.

- Truth Table: -

G3	G2	G1	G0	B3	B2	B1	B0

- K-Map: -

- Logical Equation: -

- Logic Circuit Diagram: -

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

III. Design and implement 4-bit BCD to EX-3 code converter using logic gates.

- Truth Table: -

B3	B2	B1	B0	E3	E2	E1	E0

- K-Map: -

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

- Logical Equation: -

- Logic Circuit Diagram: -

IV. Design and implement EX-3 to BCD code converter using logic gates.

- Truth Table: -

E3	E2	E1	E0	B3	B2	B1	B0

- K-Map: -

Department of Electronics & Computer Engineering

A.Y. 2025-26 (Semester: I)

Subject: Analog and Digital Electronics Lab (5303202)

- Logical Equation: -

- Logic Circuit Diagram: -

Conclusion: -

____

____

____

____

____

____

____

____

____

REFERENCES :

1. R.P. Jain, "Modern Digital Electronics", 3rd edition, 12th reprint Tata McGraw Hill Publication,2007.
2. Anand Kumar, "Fundamentals of Digital Circuits" 1st edition, Prentice Hall of India, 2001.

Attempt the Following Questions

1. Implement Binary to Gray code converter using NAND gates (universal gate) only.
2. Design a two-stage circuit to convert **Gray → Binary → Excess-3**.
3. Modify the BCD-to-Excess-3 converter so that it outputs all zeros (0000) if the input BCD code is invalid (1010–1111).

Signature of Course Teacher **Remark**