OSUN STATE UNIVERSITY, OSOGBO

ASSIGNMENT: Number Systems and Codes

IFT 211/CSC211

Instructions to Students

- Answer **ALL questions**.
- Show **clear steps** for every conversion/arithmetic question.
- Where applicable, give final answers in **both** the requested base and **decimal** for verification.

SECTION A: Concepts and Definitions (Short Answers)

1. Define a **number system** and explain what **radix/base** means.
2. State the place values (weights) for the first **8 positions** in:
    - binary
    - octal
    - hexadecimal
3. Differentiate between:
    - **positional** and **non-positional** number systems
4. Define **bit, nibble, byte, word** (give typical sizes).
5. List **three practical uses** of hexadecimal in digital systems.
6. Explain why **octal and hexadecimal** are convenient for representing binary numbers.
7. Define **BCD**, **Excess-3**, **ASCII**, and **Gray code** (one sentence each).
8. State two advantages of **Gray code** in digital systems.

SECTION B: Base Conversions (Show Steps)

Convert the following:

(i) Binary ↔ Decimal

1. $(101101)_2$to decimal
2. $(1100111)_2$to decimal
3. $(10010.101)_2$to decimal

(ii) Decimal → Binary
4. $(45)_{10}$to binary
5. $(156)_{10}$to binary
6. $(23.625)_{10}$to binary

(iii) Binary ↔ Octal
7. $(110101011)_2$to octal
8. $(725)_8$to binary
9. $(563.4)_8$to binary

(iv) Binary ↔ Hexadecimal
10. $(111011001101)_2$to hexadecimal
11. $(3AF)_{16}$to binary
12. $(B9.6)_{16}$to binary

(v) Decimal ↔ Hexadecimal
13. $(255)_{10}$to hexadecimal
14. $(1024)_{10}$to hexadecimal
15. $(7D)_{16}$to decimal

SECTION C: Binary Arithmetic (Show Work)

Perform the following operations:

1. $(101101)_2 + (11011)_2$
2. $(1110101)_2 + (1010110)_2$
3. $(100000)_2 - (1101)_2$
4. $(1010001)_2 - (111011)_2$
5. $(1101)_2 \times (101)_2$
6. $(11100)_2 \div (100)_2$(give quotient and remainder)

SECTION D: Signed Binary Numbers

1. Represent **+25** and **−25** in:
   - sign-magnitude (8-bit)
   - 1's complement (8-bit)
   - 2's complement (8-bit)
2. Find the 2's complement of each (8-bit):
   - 00110110

- 10000000
- 01111111

3. Perform using 2’s complement arithmetic (8-bit):
   - $25 + (-13)$
   - $(-18) + (-7)$
   - $14 - 27$
4. State whether **overflow occurs** in each operation and why.

SECTION E: BCD and Excess-3 Codes

(i) BCD

1. Encode these decimal numbers in **8421 BCD**:
   - 59
   - 307
   - 948
2. Decode these BCD numbers into decimal:
   - 0101 1001
   - 0011 0001 0110
3. Perform **BCD addition** and show correction where necessary:
   - 39 + 27
   - 58 + 76

(ii) Excess-3
4. Convert these decimal digits to Excess-3:

- 0, 4, 7, 9

5. Convert these Excess-3 codes to decimal:
   - 0100
   - 1001
   - 1100
6. Perform XS-3 addition (show correction):
   - 25 + 18

- 46 + 39

SECTION F: ASCII and Character Codes

1. Write the **7-bit ASCII** for:
   - A, Z, a, z, 0, 9
2. Convert the following ASCII codes to characters:
   - 1000001
   - 0110001
   - 0100000
3. A system stores the word **"CAT"** in ASCII. Write the binary for each letter and the full stream.

SECTION G: Gray Code

1. Convert the following binary numbers to Gray code:
   - 0000
   - 0101
   - 1001
   - 1110
2. Convert the following Gray codes to binary:
   - 0110
   - 1101
   - 1011
3. Explain (briefly) why Gray code reduces errors in mechanical/rotary encoders.

SECTION H: Mixed Test-Style Challenge (Very Similar to CBT)

1. Convert $(11011101)_2$to hexadecimal and decimal.
2. Convert $(3B7)_{16}$to binary and octal.
3. Add in binary: $(10101111)_2 + (01110101)_2$. State if overflow occurs (8-bit signed).
4. Represent -45in 8-bit 2's complement.

5. Convert decimal 92 to BCD and Excess-3.
6. Convert binary 101011to Gray code.