Number Systems & Boolean Algebra – Cheat Sheet (Theory Only)

---

1. Number Systems & Conversion

Common Number Systems:

- **Decimal (Base 10):** Digits {0-9} (e.g., 25_{10}).
- **Binary (Base 2):** Digits {0,1} (e.g., 1101_2).
- **Octal (Base 8):** Digits {0-7} (e.g., 37_8).
- **Hexadecimal (Base 16):** Digits {0-9, A-F} (e.g., $2F_{16}$).

Conversions Between Number Systems:

- **Decimal to Binary:** Repeated division by 2.
- **Binary to Decimal:** Multiply each bit by 2^position and sum up.
- **Binary to Octal:** Group binary digits into sets of 3 from right.
- **Binary to Hexadecimal:** Group binary digits into sets of 4 from right.

Example: Convert 25_{10} to Binary

```
25 ÷ 2 = 12  → remainder 1
12 ÷ 2 = 6   → remainder 0
6 ÷ 2  = 3   → remainder 0
3 ÷ 2  = 1   → remainder 1
1 ÷ 2  = 0   → remainder 1
Binary: 11001₂
```

Example: Convert 10110_2 to Decimal

```
(1 × 2⁴) + (0 × 2³) + (1 × 2²) + (1 × 2¹) + (0 × 2⁰)
= 16 + 0 + 4 + 2 + 0
= 22₁₀
```

---

2. Binary Arithmetic

Binary Addition Rules:

```
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (carry 1)
```

Example of Binary Addition:

```
  1011
+ 1101
---------
 11000
```

Binary Subtraction Rules:

```
0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1 (borrow 1 from the next higher bit)
```

Example of Binary Subtraction:

```
  1101
- 1010
---------
   0011
```

---

3. Binary Representation of Numbers

Signed Magnitude Representation:

- Leftmost bit = **Sign bit** (0 for positive, 1 for negative).
- Example: +5 = 00000101, -5 = 10000101.

Complement Representation:

- **1's Complement:** Flip all bits (0 → 1, 1 → 0).
- **2's Complement:** 1's complement + 1.

Example: Represent -5 in 2's Complement (8-bit)

```
Step 1: 5 in Binary → 00000101
Step 2: 1's Complement → 11111010
Step 3: Add 1 → 11111011 (Final 2's Complement)
```

---

4. Binary Codes

BCD (Binary-Coded Decimal): Each decimal digit is represented in 4-bit binary.
Gray Code: Used in error correction (only one bit changes at a time).
Excess-3 Code: Used in digital systems (BCD + 3).
Hamming Code: Used for error detection and correction.

---

5. Boolean Algebra & Boolean Functions

Boolean Variables: 0 (False) and 1 (True).

Basic Boolean Operations:

- **AND (A·B or A B)** → Output is 1 if both inputs are 1.
- **OR (A + B or A B)** → Output is 1 if at least one input is 1.
- **NOT (A' or ¬A)** → Inverts the input (0 → 1, 1 → 0).

Boolean Laws & Theorems:

- **Identity Law:** A + 0 = A, A · 1 = A.
- **Null Law:** A + 1 = 1, A · 0 = 0.
- **Idempotent Law:** A + A = A, A · A = A.
- **Complement Law:** A + A' = 1, A · A' = 0.
- **Commutative Law:** A + B = B + A, A · B = B · A.
- **Associative Law:** (A + B) + C = A + (B + C), (A · B) · C = A · (B · C).
- **Distributive Law:** A · (B + C) = (A · B) + (A · C).
- **De Morgan's Theorem:**
- (A·B)' = A' + B'
- (A+B)' = A'·B'

---

6. Canonical Forms of Boolean Functions

Sum of Products (SOP):

- OR of multiple AND terms.
- Example: A·B + A'·C.

Product of Sums (POS):

- AND of multiple OR terms.
- Example: (A + B) · (A' + C).

---

7. Boolean Function Simplification

(A) Karnaugh Map (K-Map) Method

Graphical method to minimize Boolean expressions.
Groups 1's in a 2D matrix (4-variable → 16 cells, 3-variable → 8 cells).
Steps for K-Map Simplification:

1st **Plot the 1's** in the K-Map.
2nd **Group adjacent 1's** in **powers of 2** (1, 2, 4, 8, ...).
3rd **Find the simplified expression** by eliminating variables.

Example of K-Map for 3-variable function:

```
AB\C | 0  | 1
----|----|----
0   | 1  | 1
1   | 0  | 1
```

Simplified Expression: A'C + BC.

(B) Quine-McCluskey Method

Tabular method to minimize Boolean functions systematically.
Steps:

1st List **minterms** and group by the number of 1's.
2nd Compare adjacent groups and **merge terms** by eliminating one variable.
3rd Continue merging until **prime implicants** are found.
4th Use **Prime Implicant Chart** to find the **essential terms**.

Example: Minimize F(A,B,C) = Σ(1, 2, 5, 6)

- Convert to Binary: 001, 010, 101, 110
- Group by 1's and merge:
- `001 (1 one)`
- `010 (1 one)`
- `101 (2 ones)`
- `110 (2 ones)`
-
- Find common terms → **Final Expression**

Comparison of Simplification Methods

Method	**Best for**	**Drawback**
K-Map	Up to 4-6 variables	Becomes complex for >6 variables
Quine-McCluskey	More than 6 variables	Time-consuming for large expressions

---

Key Takeaways

Number Systems: Binary, Octal, Decimal, Hexadecimal conversions.
Binary Arithmetic: Addition, subtraction, complements.
Boolean Algebra: Laws, De Morgan's Theorem, Canonical Forms.
Boolean Function Simplification: K-Map for small functions, Quine-McCluskey for larger ones.

This **Cheat Sheet on Number Systems & Boolean Algebra** covers **conversions, binary operations, Boolean functions, simplifications, and K-Map techniques**. Let me know if you need further explanations!