ABHYASMITRA.IN

COMPREHENSIVE UNIVERSITY STUDY NOTES

DIGITAL ELECTRONICS AND LOGIC DESIGN

Unit I — Introduction to Digital Systems & Boolean Minimization

Topics Covered in this Unit

Positional Number Systems: Binary, Decimal, Octal & Hexadecimal Radix Representations

Inter-Base Number System Conversions: Base-r to Decimal, Decimal to Base-r & Direct Octal/Hex/Binary Conversions

Signed Binary Number Representations: Sign-Magnitude, 1's Complement & 2's Complement Forms

Binary Arithmetic: Signed Addition & Subtraction using 2's Complement and Hardware Overflow Detection

Fixed-Point Binary Number Representation (Integer and Fractional Bit Layouts)

Standard & Canonical Logic Expressions: Sum of Products (SOP / Minterms) & Product of Sums (POS / Maxterms)

Boolean Minimization via Karnaugh Maps (K-Maps up to 4 Variables): Gray Code Adjacency & Grouping Rules

Prime Implicants, Essential Prime Implicants & Incompletely Specified Functions (Don't Care Conditions)

Minimization of SOP and POS Expressions using K-Maps with Full Step-by-Step Circuit Implementations

High-Yield University Exam Question Bank with Detailed Model Answers & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Electronics, Electrical, CS & AI Students

COURSE SYLLABUS & MAPPING

[UNIT I SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit I. Verify with your university curriculum mapping.

Unit I Syllabus Breakdown:

Unit I - Introduction to Digital Systems: Number systems: Binary, Decimal, Octal and Hexadecimal Number system, conversion of one number system to another, representation of signed number- sign magnitude representation, 1’s complement and 2’s complement form, addition and subtraction of two binary numbers, Fixed point representation of a numbers, standard representation for logic functions, Minimization of Boolean function using K-map (up to 4 variables), Minimization of SOP and POS using K-map.

• Course Code: ___________________ • Semester / Year: ___________________

TABLE OF CONTENTS

1.1 Number Systems & Radix Conversions

Positional Number System Formulation

In a positional number system with base (radix) r, any real number N is evaluated as a power series of positional weights:

$$N = \sum_{i=0}^{n-1} d_i \times r^i + \sum_{j=1}^{m} d_{-j} \times r^{-j}$$

where d_i represents allowable digits in the set {0, 1, ..., r-1}, n is the number of integer digits, and m is the number of fractional digits.

Primary Number Systems in Digital Logic

Number System	Base (Radix r)	Permissible Digits / Symbols	Positional Weights Example
Binary	2	0, 1	$(1101.1)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} = 13.5_{10}$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	$(157.2)_8 = 1\times8^2 + 5\times8^1 + 7\times8^0 + 2\times8^{-1} = 111.25_{10}$
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$(742.5)_{10} = 7\times10^2 + 4\times10^1 + 2\times10^0 + 5\times10^{-1}$
Hexadecimal	16	0-9, A(10), B(11), C(12), D(13), E(14), F(15)	$(2A.8)_{16} = 2\times16^1 + 10\times16^0 + 8\times16^{-1} = 42.5_{10}$

Systematic Inter-Base Conversion Procedures

- **1. Decimal to Any Base-r:** Integer part: Successive division by r (remainders collected in bottom-up order / LSB to MSB). Fractional part: Successive multiplication by r (integers collected in top-down order / MSB to LSB).
- **2. Any Base-r to Decimal:** Evaluate the sum of products of each digit with its corresponding positional weight r^k.
- **3. Binary ↔ Octal & Hexadecimal Grouping:**
  - • **Binary to Octal:** Group binary bits in triplets (3 bits) from radix point outward (e.g., $101110_2 \rightarrow (101)(110)_2 = 56_8$).
  - • **Binary to Hexadecimal:** Group binary bits in nibbles (4 bits) from radix point outward (e.g., $11011010_2 \rightarrow (1101)(1010)_2 = DA_{16}$).

1.2 Signed Number Representations & 2's Complement Arithmetic

Signed Binary Formats for n-Bit Numbers

1. Sign-Magnitude: MSB represents Sign (0 = +, 1 = -); remaining (n-1) bits represent magnitude. Range: [-(2^(n-1) - 1) to +(2^(n-1) - 1)]. Has Dual Zero (+0 and -0).

2. 1's Complement: Positive numbers same as binary. Negative numbers obtained by inverting all bits (0 ↔ 1). Range: [-(2^(n-1) - 1) to +(2^(n-1) - 1)]. Has Dual Zero.

3. 2's Complement: Negative numbers obtained by taking 1's Complement and adding 1 (or leaving least significant 0s and first 1 unchanged, inverting rest). Range: [-2^(n-1) to +(2^(n-1) - 1)]. Unique Zero!

Representation (8-bit)	+45 in Binary	-45 Representation	Total Numeric Range (8-bit)
Sign-Magnitude	00101101	10101101	-127 to +127 (Has +0: 00000000 and -0: 10000000)
1's Complement	00101101	11010010 (Inverted bits)	-127 to +127 (Has +0: 00000000 and -0: 11111111)
2's Complement (Standard)	00101101	11010011 (1's comp + 1)	-128 to +127 (Single unique zero: 00000000)

Signed Addition, Subtraction & Hardware Overflow Detection

- • **2's Complement Subtraction Rule:** A - B is evaluated as A + (2's Complement of B). Any end carry generated out of the MSB position is simply discarded.
- • **Hardware Arithmetic Overflow Condition:** Overflow occurs when the result of adding two numbers of the same sign exceeds the representable range [-2^(n-1), 2^(n-1)-1], resulting in an incorrect sign bit:

Overflow = C_in ⊕ C_out   (where C_in is carry into MSB and C_out is carry out of MSB)

1.3 Fixed-Point Number Representation

In fixed-point binary representation, the binary radix point is held at a fixed, predefined position:

- • **Q-Format Representation (e.g., Q4.4):** 4 integer bits and 4 fractional bits. The number 1101.1010_2 evaluates to 13 + 0.625 = 13.625_10. Provides deterministic hardware computation in DSPs and low-power microcontrollers without floating-point unit (FPU) hardware overhead.

1.4 Standard & Canonical Logic Expressions: SOP and POS

Canonical Forms: Minterms vs. Maxterms

• Minterm (m_i): A product (AND) of all literals where each variable appears exactly once in true or complemented form (evaluated to 1 for that input combination).

• Maxterm (M_i): A sum (OR) of all literals where each variable appears exactly once (evaluated to 0 for that input combination).

• Sum of Products (SOP): F = ∑ m(m_1, m_2, ...) (ORing minterms where output is 1).

• Product of Sums (POS): F = ∏ M(M_1, M_2, ...) (ANDing maxterms where output is 0).

1.5 Karnaugh Map (K-Map) Minimization up to 4 Variables

Definition: Karnaugh Map (Maurice Karnaugh, 1953)

A K-Map is a graphical matrix of 2^n cells (for n variables) arranged in Gray Code sequence such that adjacent horizontal and vertical cells differ by exactly ONE binary bit, enabling the visual application of the Boolean adjacency rule: A B + A B' = A.

Structure of 2, 3, and 4-Variable K-Maps

THE 4-VARIABLE (4x4) KARNAUGH MAP GRID (VARIABLES AB \ CD)

```
          CD: 00 (C'D')   01 (C'D)    11 (CD)     10 (CD')
AB:
00 (A'B')   |   m0      |   m1      |   m3      |   m2      |
01 (A'B)    |   m4      |   m5      |   m7      |   m6      |
11 (AB)     |   m12     |   m13     |   m15     |   m14     |
10 (AB')    |   m8      |   m9      |   m11     |   m10     |
```

Figure: Notice Gray code order: 00, 01, 11, 10 ensuring adjacent cell Hamming distance = 1

Grouping Rules & Definitions

- **1. Power-of-2 Grouping:** Groups must contain 1, 2 (Pair), 4 (Quad), 8 (Octet), or 16 cells in rectangular shapes.

- **2. Toroidal Wrapping:** Cells on the extreme left column wrap around to the extreme right column; cells on top row wrap to bottom row. The 4 corner cells {m0, m2, m8, m10} form a valid Quad eliminating 2 variables!
- **3. Prime Implicant (PI):** A product term obtained by combining the maximum possible number of adjacent 1-cells.
- **4. Essential Prime Implicant (EPI):** A Prime Implicant that contains at least ONE minterm (1-cell) that is NOT covered by any other prime implicant. ALL EPIs must be included in the minimal Boolean equation.
- **5. Don't Care Conditions (d / ×):** Input combinations that can never physically occur or whose output is irrelevant. In K-maps, a Don't Care is treated as 1 if it helps form a larger group, or as 0 if not needed.

1.6 High-Yield University Exam Question Bank & Model Answers

Part A: Short Answer Questions (2–3 Marks)

- **Q1. What are the advantages of 2's Complement representation over 1's Complement?**

Answer: (1) 2's complement has a single, unique representation for zero (00000000), whereas 1's complement has dual zeros (+0 and -0), (2) Subtraction is performed directly using standard binary adder hardware by inverting bits and adding 1, and (3) It provides an extra negative number range [-2^(n-1)].

- **Q2. Define Essential Prime Implicant (EPI) in K-Map simplification.**

Answer: An Essential Prime Implicant is a Prime Implicant (maximum-sized group of adjacent 1-cells) that covers at least one minterm that is not covered by any other prime implicant. Every EPI is mandatory in the minimal SOP expression.

- **Q3. Convert (53.625)_10 into Binary and Octal.**

Answer: Integer 53 = 110101_2; Fraction 0.625 × 2 = 1.25 → 1, 0.25 × 2 = 0.5 → 0, 0.5 × 2 = 1.0 → 1 = .101_2. Binary = 110101.101_2. Grouping in triplets: (110)(101).(101)_2 = (65.5)_8.

- **Q4. State the condition for detecting Arithmetic Overflow in 2's complement addition.**

Answer: Overflow occurs when the carry into the sign bit (MSB) does NOT equal the carry out of the sign bit: Overflow = C_in ⊕ C_out = 1.

Part B: Long Answer Questions (8–10 Marks)

- **Q5. Minimize the Boolean function F(A, B, C, D) = ∑ m(0, 1, 2, 5, 8, 9, 10) + d(7, 14, 15) using a 4-variable K-Map. Identify all Prime Implicants and Essential Prime Implicants, and implement the minimal circuit using NAND gates.**

Model Answer Outline:

- 1. Plotting the 4×4 K-map with Gray code row/col coordinates (00, 01, 11, 10).
- 2. Inserting 1s at minterms 0, 1, 2, 5, 8, 9, 10 and × at don't cares 7, 14, 15.
- 3. Identifying Groups:
- - Quad 1 (Corner cells m0, m2, m8, m10) → Term = B' D' (EPI).
- - Quad 2 (m0, m1, m8, m9) → Term = B' C' (EPI).
- - Quad 3 (m1, m5, m3[if used] / m5 with d7 → m5, m7 with m1, m3 or Pair m1, m5) → Term = A' C' D.
- 4. Final Minimal SOP Expression: F = B' D' + B' C' + A' C' D.
- 5. Conversion to NAND-NAND 2-level logic circuit diagram using De Morgan's Law.

- **Q6. Perform the following arithmetic operations using 8-bit 2's Complement representation: (1) (+43) + (-27), (2) (-35) - (+18). Verify the results in decimal and state if overflow occurs.**

Model Answer Outline:

- 1. 8-bit binary representation of +43 = 00101011; +27 = 00011011 → 2's complement (-27) = 11100101.
- 2. Operation (1): 00101011 + 11100101 = (1) 00010000_2 = +16_10. Discard carry out. Result is correct (+16); Overflow = 0.
- 3. Operation (2): -35 = 11011101; - (+18) = + (-18) = 11101110.
- 4. Addition: 11011101 + 11101110 = (1) 11001011_2. 2's complement of result = -(00110101_2) = -53_10. Result verified (-35 - 18 = -53); Overflow = 0.

1.7 Unit I Summary & Quick Revision Sheet

Digital Systems Topic	Key Theoretical & Mathematical Takeaways for Exams
Number Systems	Binary (2), Octal (8), Decimal (10), Hex (16). Octal=3-bit groups, Hex=4-bit groups.
Signed Arithmetic	2's Comp = 1's Comp + 1. Range: [-2^(n-1), +2^(n-1)-1]. Overflow = C_in ⊕ C_out = 1.
Standard Forms	Minterms (m_i, AND terms) → SOP. Maxterms (M_i, OR terms) → POS.
K-Map Minimization	Gray code order (00,01,11,10). Groups of 2^k (Pairs, Quads, Octets). Corners wrap.
Implicants & Don't Cares	PI = Max group. EPI = Covers unique 1-cell. Don't cares (×) included only to enlarge groups.