Digital Logic Concepts — Comprehensive Notes

Introduction

1. **Clear hierarchical organization** using sections and subsections.
2. **Mathematical formalism** using LaTeX notation for:
   - Truth tables,
   - Binary conversions,
   - Boolean expressions.
3. **Systematic presentation** of:
   - Two's complement conversion methodology,
   - SOP derivation,
   - Thermometer code conversion,
   - K-map optimization principles.

For additional work, we could:

1. Add more detailed K-map examples,
2. Include step-by-step conversion examples,
3. Expand the don't care conditions analysis,
4. Add practice problems with solutions.

What aspect would be most helpful to explore further?

1 Two's Complement Conversion Methodology

1.1 Core Principles

For an 8-bit system, two's complement conversion follows these recursive steps:

For negative numbers (n):

1. Convert $|n|$ to binary (8 bits).
2. Invert all bits (i.e., perform the NOT operation).
3. Add 1 to the result.

For positive numbers:

- Use the direct binary representation with a leading 0.

1.2 Representation Framework

Each number requires four distinct representations:

Base-10	Positive Binary	Hexadecimal	2's Complement
n	$b_7b_6b_5b_4b_3b_2b_1b_0$	0x??	$c_7c_6c_5c_4c_3c_2c_1c_0$

1.3 Example Transformations

For $n = -12$:

$$\begin{aligned}|n| &= 12_{10} = 00001100_2,\\ \text{Invert} &= 11110011_2,\\ \text{Add 1} &= 11110100_2,\\ \text{Hex} &= \text{0x0C}.\end{aligned}$$

2 Sum-of-Products (SOP) Derivation

2.1 Boolean Function Construction

For a truth table with inputs A and B:

A	B	Out
0	0	0
0	1	1
1	0	0
1	1	1

The SOP expression is constructed as:

$$\text{Out} = \overline{A}B + AB.$$

2.2 Three-Input Systems

For inputs A, B, and C:

A	B	C	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

3 Thermometer Code Conversion

3.1 State Mapping

Thermometer code follows the sequence:

$$000_2 \to 001_2 \to 011_2 \to 111_2.$$

Converting to binary output $[X_1 X_0]$:

$$\begin{aligned} 000_2 &\to 00_2, \\ 001_2 &\to 01_2, \\ 011_2 &\to 10_2, \\ 111_2 &\to 11_2. \end{aligned}$$

3.2 Don't Care Conditions

For input bits $[T_2 T_1 T_0]$, invalid states are marked as don't care (X):

T_2	T_1	T_0	$[X_1 X_0]$
0	0	0	00
0	0	1	01
0	1	0	X
$\vdots$	$\vdots$	$\vdots$	$\vdots$

4 Karnaugh Map Optimization

4.1 Prime Implicant Identification

For a K-map with variables A and B:

AB'	AB
$A'B'$	$A'B$

4.2 Minimization Rules

1. Group adjacent 1's in powers of 2.
2. Include don't cares (X) when beneficial.
3. Minimize the number of terms in the final expression.

The minimal SOP expression is derived from the largest possible groupings of 1's and the strategic use of don't care conditions.