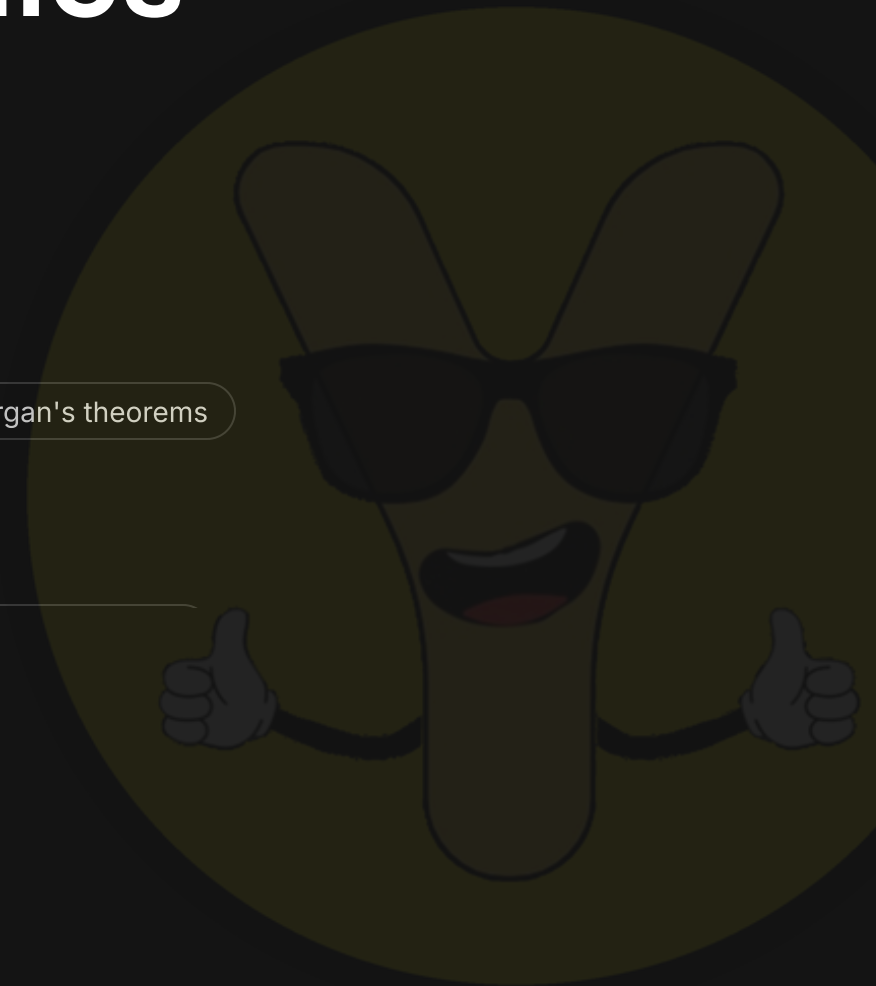
AKTU · 1ST YEAR · AEC101A / AEC201A

Fundamentals of Electronics Engineering

Unit 4 — Digital Electronics Basics

- Number systems: binary, decimal, hexadecimal (conversion only)
- logic levels (HIGH/LOW, 0/1)
- basic logic gates: AND, OR, NOT, NAND, NOR, XOR with truth tables
- Boolean algebra: basic laws and De Morgan's theorems
- k-map (upto 4 variables)
- half adder and full adder (truth table and logic diagram)
- multiplexer concept
- flip-flop: SR and D flip-flop (concept only)



DECODED BY

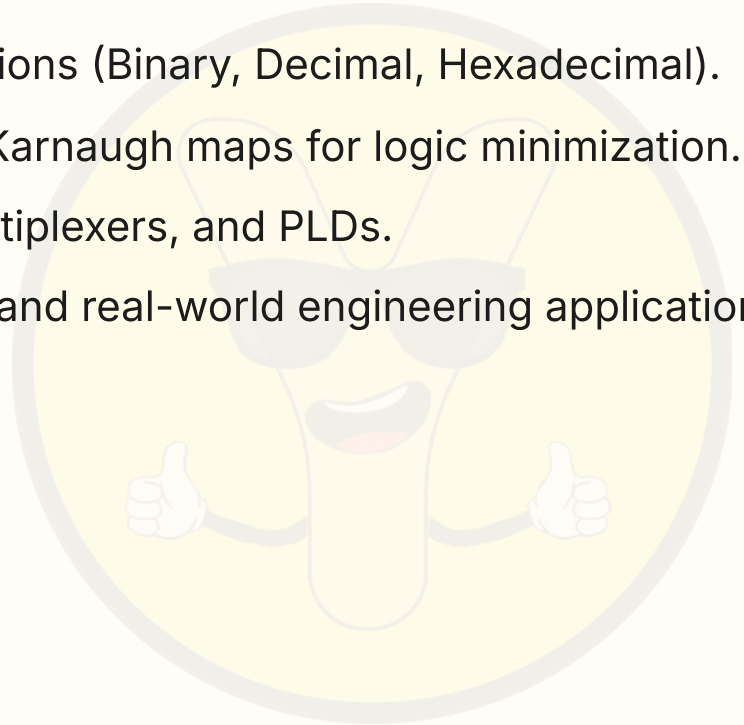
Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Unit 4 Overview: Digital Electronics Basics

This unit covers foundational digital concepts required for computing and microprocessor systems.

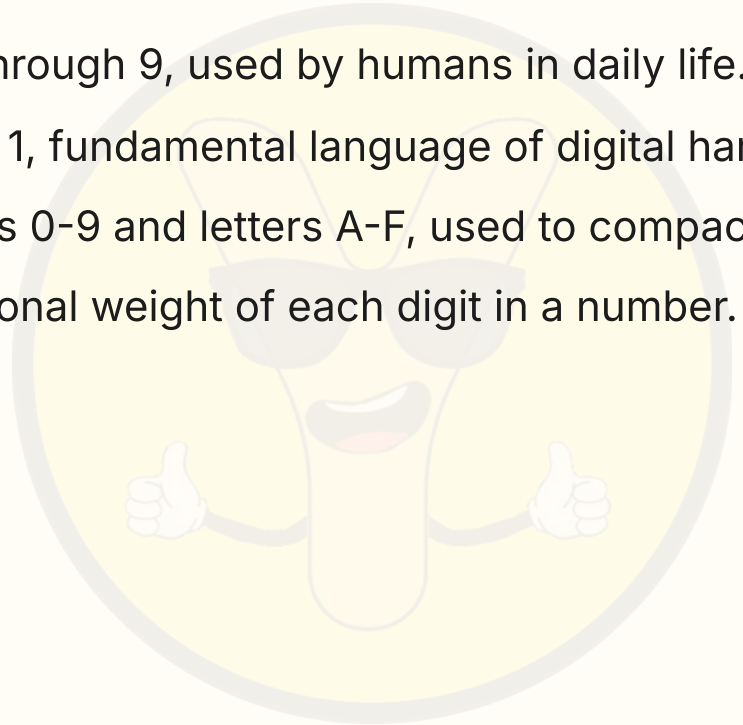
- Number systems and base conversions (Binary, Decimal, Hexadecimal).
- Logic gates, Boolean algebra, and Karnaugh maps for logic minimization.
- Combinational circuits: Adders, Multiplexers, and PLDs.
- Sequential circuits: Basic flip-flops and real-world engineering applications.



Number Systems: Introduction

Digital systems process information using discrete numeric values.

- Decimal System: Base 10, digits 0 through 9, used by humans in daily life.
- Binary System: Base 2, digits 0 and 1, fundamental language of digital hardware.
- Hexadecimal System: Base 16, digits 0-9 and letters A-F, used to compactly represent binary streams.
- Radix or base determines the positional weight of each digit in a number.



Number Conversion: Decimal to Binary and Hex

Convert the decimal number 45 to binary and hexadecimal.

1. Binary conversion by successive division by 2: 45/2 = 22 R 1, 22/2 = 11 R 0, 11/2 = 5 R 1, 5/2 = 2 R 1, 2/2 = 1 R 0, 1/2 = 0 R 1.
2. Read remainders bottom-up to get binary: $45_{10} = 101101_2$.
3. Hexadecimal conversion by dividing binary into 4-bit blocks from right: 0010 (2) and 1101 (D).
4. Combine blocks to get hexadecimal: $45_{10} = 2D_{16}$.

Binary: 101101_2, Hexadecimal: 2D_16

Logic Levels in Digital Circuits

LOGIC LEVEL

The discrete voltage ranges used in digital electronics to represent binary states 0 and 1.

- HIGH (Logic 1): Represented by a higher voltage level (e.g., $+3.3\text{V}$ to $+5\text{V}$ in TTL/CMOS).
- LOW (Logic 0): Represented by a lower voltage level (e.g., 0V to 0.8V).
- Noise margin ensures reliable operation by defining buffer zones between valid HIGH and LOW thresholds.

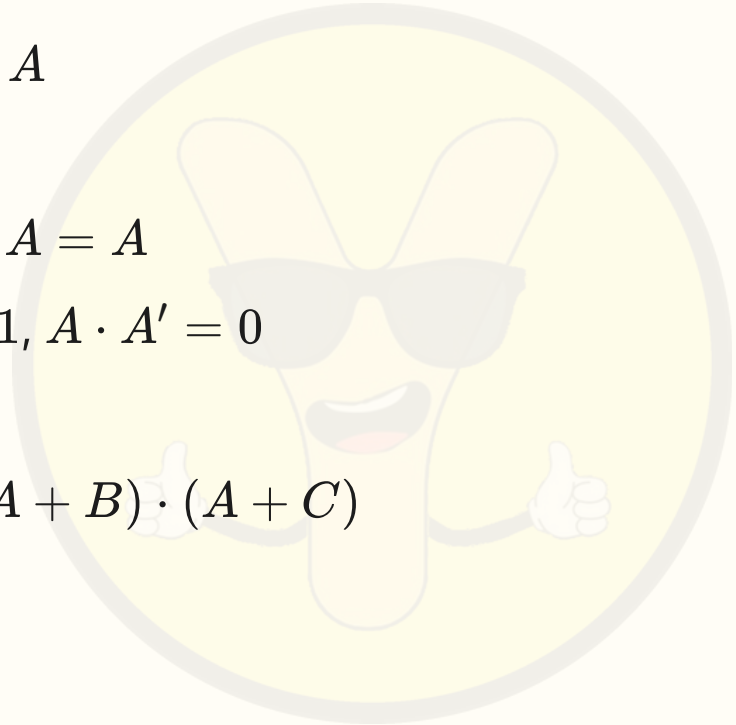
Basic Logic Gates and Truth Tables

Gate Type	Boolean Expression	Operation Summary
AND	Y = A . B	Output is HIGH only if all inputs are HIGH.
OR	Y = A + B	Output is HIGH if at least one input is HIGH.
NOT	Y = A'	Inverts the input state.
NAND	Y = (A . B)'	AND gate followed by NOT; universal gate.
NOR	Y = (A + B)'	OR gate followed by NOT; universal gate.
XOR	Y = A \oplus B	HIGH if inputs are different (odd number of 1s).

Boolean Algebra: Laws

Mathematical rules to simplify digital logic expressions and reduce hardware gate counts.

- Identity Laws: $A + 0 = A$, $A \cdot 1 = A$
- Null Laws: $A + 1 = 1$, $A \cdot 0 = 0$
- Idempotent Laws: $A + A = A$, $A \cdot A = A$
- Complementarity Laws: $A + A' = 1$, $A \cdot A' = 0$
- Involution Law: $(A')' = A$
- Distributive Law: $A + (B \cdot C) = (A + B) \cdot (A + C)$



De Morgan’s Theorems

State and verify De Morgan's theorems for two variables A and B.

1. First Theorem: The complement of a sum is equal to the product of the complements. Formula: $(A + B)' = A' \cdot B'$.
2. Second Theorem: The complement of a product is equal to the sum of the complements. Formula: $(A \cdot B)' = A' + B'$.
3. Verification First Theorem: For A=0, B=1, LHS = $(0 + 1)' = 1' = 0$, RHS = $0' \cdot 1' = 1 \cdot 0 = 0$. LHS equals RHS.
4. Verification Second Theorem: For A=0, B=1, LHS = $(0 \cdot 1)' = 0' = 1$, RHS = $0' + 1' = 1 + 0 = 1$. LHS equals RHS.

Formulas verified using exhaustive truth table evaluation.

Karnaugh Map (K-Map) Fundamentals

A pictorial method used to minimize Boolean expressions without extensive algebraic steps.

- Based on Gray code ordering where adjacent cells differ by only one bit variable.
- Grouping rules: Groups must contain 2^n cells (1, 2, 4, 8, 16).
- Larger groups result in fewer literals in the simplified product term.
- Corner wrapping and edge wrapping are allowed to form valid adjacencies.
- Unspecified conditions (Don't care states, 'X') can be used as 0 or 1 to maximize group size.

4-Variable K-Map Simplification

Minimize the SOP expression: $F(A, B, C, D) = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$

1. Plot 1s in cells 0, 2, 5, 7, 8, 10, 13, and 15 on a 4×4 K-map grid.
2. Form a quad using corners (cells 0, 2, 8, 10): yields $B'D'$.
3. Form a second quad using cells (5, 7, 13, 15): yields BD.
4. Combine simplified product terms to write the minimal Boolean expression.

F = B'D' + BD

Combinational Circuits: Half Adder

A basic arithmetic circuit that adds two single-bit binary numbers.

- Inputs: Two single-bit variables A and B.
- Outputs: Sum (S) and Carry (C).
- Boolean Expressions: $S = A \oplus B$, $C = A \cdot B$.
- Logic Implementation: Built using one XOR gate and one AND gate.
- Limitation: Cannot handle a carry-in from a previous lower-significant addition stage.

Combinational Circuits: Full Adder

An arithmetic circuit that adds three single-bit numbers to generate sum and carry.

- Inputs: A, B, and C_{in} (Carry-in from previous stage).
- Outputs: Sum (S) and C_{out} (Carry-out).
- Boolean Expressions: $S = A \oplus B \oplus C_{in}$, $C_{out} = AB + BC_{in} + AC_{in}$.
- Can be constructed by cascading two Half Adders and an OR gate.
- Forms the building block for multi-bit ripple carry adders in processor ALUs.

Multiplexer (MUX) Concept

MULTIPLEXER

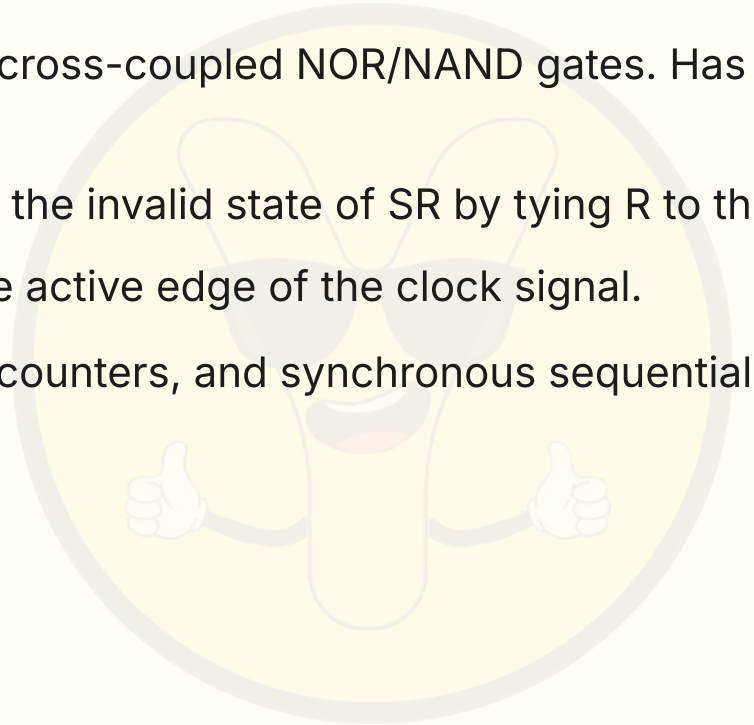
A combinational circuit that selects one of many input data sources and forwards it to a single output line.

- Referred to as a 'data selector' controlled by select lines.
- If there are n select lines, there can be up to 2^n input lines.
- Example: A 4-to-1 MUX has 4 data inputs, 2 select lines, and 1 output.
- Eliminates complex wiring by time-sharing a single transmission channel.

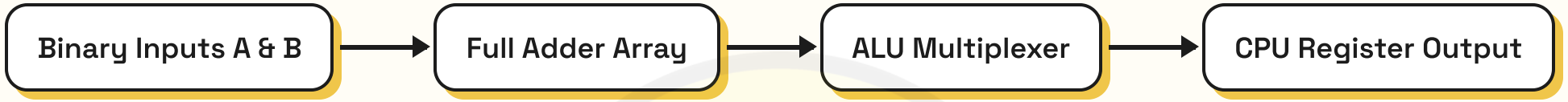
Sequential Circuits: SR and D Flip-Flop

Flip-flops are 1-bit memory elements capable of storing binary states.

- SR Flip-Flop: Set-Reset latch using cross-coupled NOR/NAND gates. Has an invalid state when both S and R are HIGH simultaneously.
- D Flip-Flop (Data/Delay): Eliminates the invalid state of SR by tying R to the inverted S input.
- Output Q follows the input D on the active edge of the clock signal.
- Forms the foundation for registers, counters, and synchronous sequential systems.



Digital Applications in Processor ALU

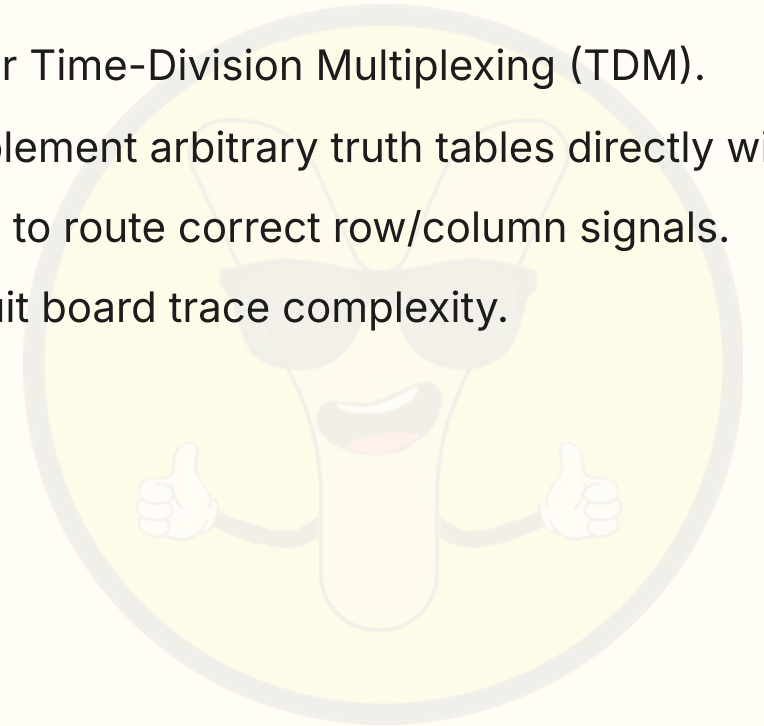


Data flow during binary addition inside a central processing unit Arithmetic Logic Unit.

Application: Multiplexers in Data Routing

MUX circuits dynamically route data streams in complex digital systems.

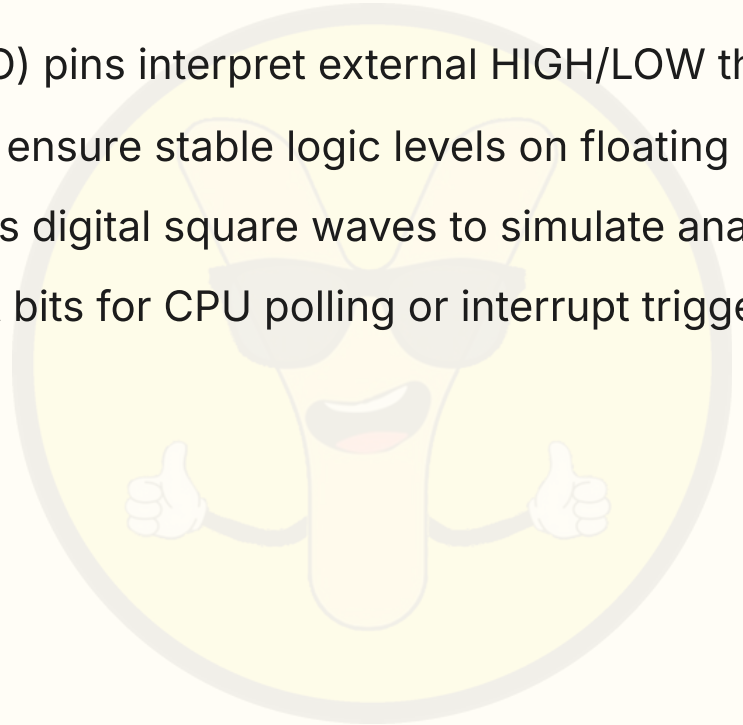
- Used in communication systems for Time-Division Multiplexing (TDM).
- Acts as a function generator to implement arbitrary truth tables directly without gates.
- Used in memory address decoding to route correct row/column signals.
- Reduces IC count and printed circuit board trace complexity.



Application: Digital Signals in Microcontroller I/O

Microcontrollers interface with physical world sensors using digital input/output pins.

- General Purpose Input Output (GPIO) pins interpret external HIGH/LOW thresholds.
- Internal pull-up/pull-down resistors ensure stable logic levels on floating input pins.
- PWM (Pulse Width Modulation) uses digital square waves to simulate analog control.
- Microcontroller registers latch input bits for CPU polling or interrupt triggering.



Basics of PLA and PLD Devices

PROGRAMMABLE LOGIC DEVICE (PLD)

An integrated circuit with configurable internal logic gates and interconnection pathways.

- PLA (Programmable Logic Array): Features programmable AND array followed by a programmable OR array.
- PAL (Programmable Array Logic): Features programmable AND array and fixed OR array.
- Allows implementation of complex combinational logic without designing custom IC masks.
- Precursor to modern FPGAs (Field Programmable Gate Arrays).

Exam corner

Q1 Convert BAD_{16} to binary and decimal equivalents.

Q2 State and prove De Morgan's theorems using truth tables and algebraic manipulation.

Q3 Minimize the Boolean expression $F(A, B, C, D) = \sum m(1, 3, 5, 7, 9, 11, 13, 15)$ using a K-Map.

Q4 Draw the logic diagram of a Full Adder using two Half Adders and an OR gate.

Q5 Explain the working principle and truth table of a 4-to-1 Multiplexer.

Q6 Differentiate between combinational and sequential circuits with examples.

Q7 Write short notes on Programmable Logic Arrays (PLA) and D Flip-Flops.

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

