# EEE283

## Digital Logic Design

**Course Instructor: A  K M Anindya Alam**

# Faculty Introduction


## A K M ANINDYA ALAM

**Lecturer**, Department of Electrical and Electronic Engineering
BSRM School of Engineering, BRAC University


**Previous Affiliation**

- Adjunct Lecturer, Department of EEE, BUET April 2025- July 2025

**Education**

- BSc. in EEE from BUET 2020 - 2025
- HSC in Notre Dame College 2019
- SSC in St. Joseph Higher Secondary School 2017


*E-mail : anindya.alam@bracu.ac.bd*


# Class Ground rules

1. Stop whispering loudly

**EVERYONE CAN HEAR YOU!**

2. No tolerance for plagiarism

**What you can't do**

**What you can do**

# Course Information

A. Course General Information:

| | |
|---|---|
| **Course Code:** | EEE 283<br>EEE 283L |
| **Course Title:** | Digital Logic Design<br>Digital Logic Design Laboratory |
| **Credit Hours** (Theory + Laboratory): | 3 + 1 |
| **Contact Hours** (Theory + Laboratory): | 3 + 3 |
| **Category:** | Program Core |
| **Type:** | Required, Engineering, Lecture + Laboratory |
| **Prerequisites:** | EEE 205 Electronic Circuit I<br>EEE 205L Electronic Circuit I Laboratory |
| **Co-requisites:** | None |
| **Equivalent Course** | ECE 283 Digital Logic Design<br>EEE 301 Digital Electronics – v1, v2<br>ECE 301 Digital Electronics – v1, v2<br><br>ECE 283L Digital Logic Design Laboratory<br>EEE 302 Digital Electronics Laboratory (1.5 credits) – v1, v2<br>ECE 302 Digital Electronics Laboratory (1.5 credits) – v1, v2 |

# Course Outcome

**D. Course Outcomes (COs):**

Upon successful completion of this course, students will be able to

| Sl. | CO Description |
|---|---|
| CO1 | **Apply** the concept of digital logic design to solve the problem using gates to replicate logic functions |
| CO2 | **Analyze** combinational and sequential logic circuits built with various logic gates, flip-flops, registers, counters etc. represented through schematic diagram or hardware description language. |
| CO3 | **Design** combinational and sequential logic circuits using various logic gates, flip-flops, registers, counters as building blocks |
| CO4 | **Perform** effectively as an individual or in a team to design and build combinational and sequential logic circuits in the laboratory or project development |
| CO5 | **Communicate** the findings of hardware and software experiments and projects through reports and presentations |

# General Outline

1. **Fundamental Logic Gates:** Introduction to Digital Logic, Concept of Logic Gates, NOT Logic, AND Logic, OR Logic, Universal Logic Gates, NAND Logic, NOR Logic, XOR Logic, XNOR Logic, Logic Gate Network, NAND/NOR implementation, Cost of Logic Circuit.

2. **Boolean Algebra**: Boolean Operators, Logic Functions, Truth Table, DeMorgan's Theorem, Maxterms & Minterms, SOP vs. POS Logic, Karnaugh Map, Cost Minimization.

3. **Combinational Logic:** Multiplexer, Look-Up Table, Encoder, Decoder, Demultiplexer, Shannon's Expansion Theorem, Binary Coded Decimal, Seven Segment, Comparator.

4. **Binary Arithmetic:** Binary Number Representation, Carry Flag, Half Adder, Full Adder, Ripple Carry Adder, Binary Multiplication, Binary Negative Numbers, Adder-Subtractor Unit, Overflow Flag, Binary Coded Decimal Adder, Hexadecimal Number System.

5. **Sequential Logic:** Memory Circuits, S-R Latch, D Latch, Master-Slave Flip-Flop, D FlipFlop, T Flip-Flop, J-K Flip-Flop, Hold & Setup Time, Counter, Asynchronous vs. Synchronous Counter, Ring Counter, Johnson Counter, BCD Counter, Shift Register.

5. **Sequential Logic:** Memory Circuits, S-R Latch, D Latch, Master-Slave Flip-Flop, D FlipFlop, T Flip-Flop, J-K Flip-Flop, Hold & Setup Time, Counter, Asynchronous vs Synchronous Counter, Ring Counter, Johnson Counter, BCD Counter, Shift Register.

6. **Finite State Machines:** Moore and Mealy Implementation, Synchronous Finite State Machines, Logic Gate implementation.

7. **CMOS Implementation:** Introduction to CMOS technology, CMOS vs NMOS technology, Implementation of NOT, NAND and NOR logic gates, Implementation of AND, OR, XOR & XNOR Logic, Implementation of complex logic via NAND/NOR Integrated Implementation of Complex Logic, Transmission Gates, Programmable Logic Array.

8. **Hardware Description Language:** Introduction to Verilog, Boolean Function in Verilog, Coding via Wire Command, Coding via Assign Command, Coding via Always Command, Blocking vs. Non-Blocking Procedure, Coding for Combinational vs Sequential Logic.

# Assessment tool for Grading

**Theory**

| **Assessment Tools** | **Weightage** |
|---|---|
| Class Attendance and Performance | 10% |
| Assignment | 10% |
| Quiz | 10% |
| Midterm Exam | 25% |
| Final Exam | 25% |
| Project (Design) | 20% |

# EEE283

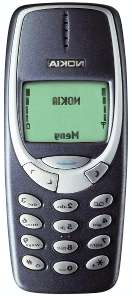
## Digital Logic Design

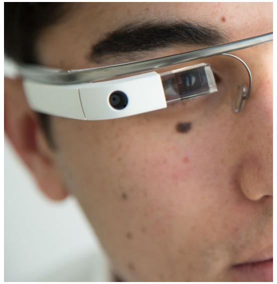
**Course Instructor: A K M Anindya Alam**

# Topic 1:

## Introduction to Digital Electronics

# Digital Electronics



# Analog vs. Digital

Analog signal- one whose output varies continuously in step with the input.

Example:


Digital signal- one whose output varies at discrete voltage levels commonly called HIGH or LOW (1 or 0).

Example:

# Why Digital Electronics?

- Data can be stored (memory characteristics)
- Data can be used in calculations
- Compatible with display technologies
- Compatible with Computer technologies
- Systems can be programmed
- Digital IC families make design easier

# Why Analog Electronics?

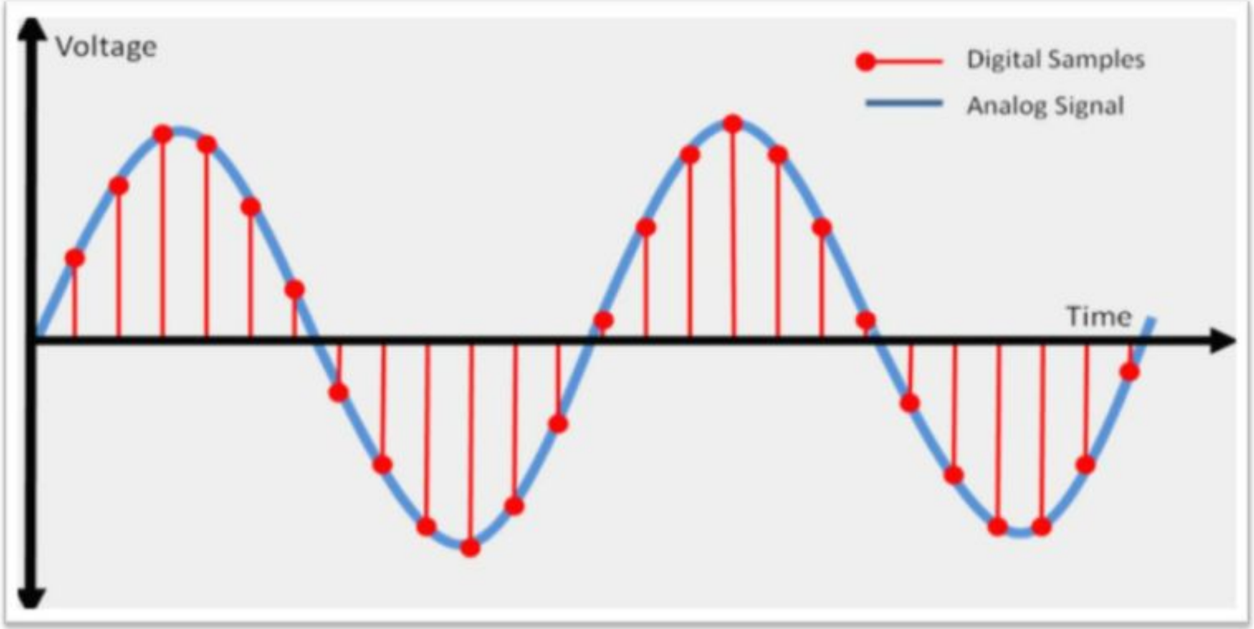
- Most “real-world events are analog in nature
- Analog processing is usually simpler
- Analog processing is usually faster
- Traditional electronics systems were mostly analog

# Benefits of Digital over Analog

- Reproducibility
- Not affected by noise (better quality)
- Ease of design
- Data protection
- Programmable
- Speed
- Economy

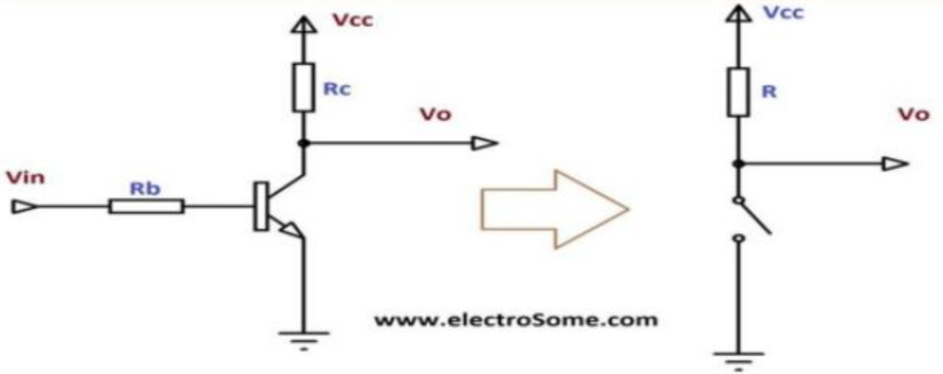
# Analog to Digital Conversion

- SAMPLING

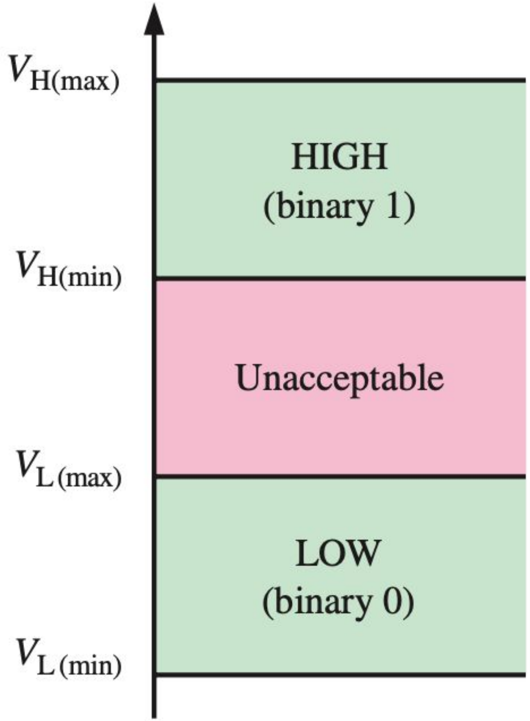


# Analog to Digital Conversion

- At any point in circuit, only two Voltage states are present – HIGH and LOW
- Also sometimes called TRUE or FALSE. In Boolean logic, 0 and 1
- Generally, +5V is considered high, 0V considered ground



# Binarization



# Information Representation

- Information variables are represented by physical quantities (voltage, current, charge, magnetic field orientation, temperature, phase)
- For digital systems, the variables take discrete values
- Two level or binary values are most common

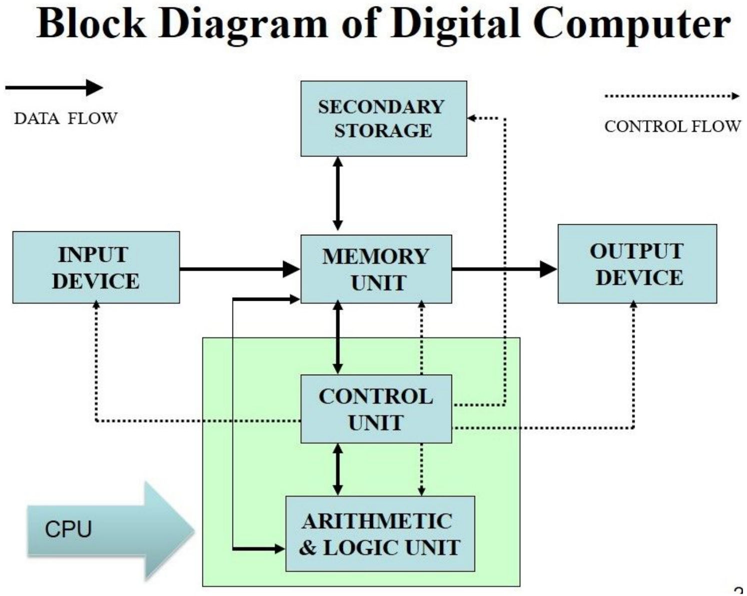
# Binary Representation of Physical Quantities

- CPU: Voltage
- HDD: Magnetic Field Direction
- CD: Physical shape of pits
- DRAM: Electric Charge
- Optical Fiber: Light intensity

# Types of Digital Systems

- No state Present, *Output = f(input)*
  - Combinational logic system
- State Present, *Output = f(input, state)*
  - Synchronous Sequential System (state updated at discrete times)
  - Asynchronous Sequential System (state updated at any time)

# Digital System Example


**Digital clock**
Input: Time set
Output: Current time
State: Current time

**Digital computer**