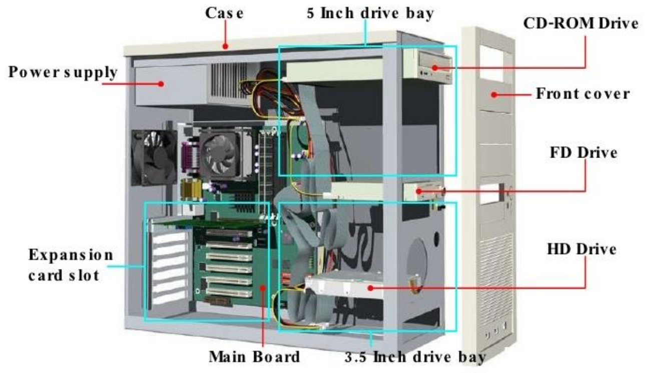
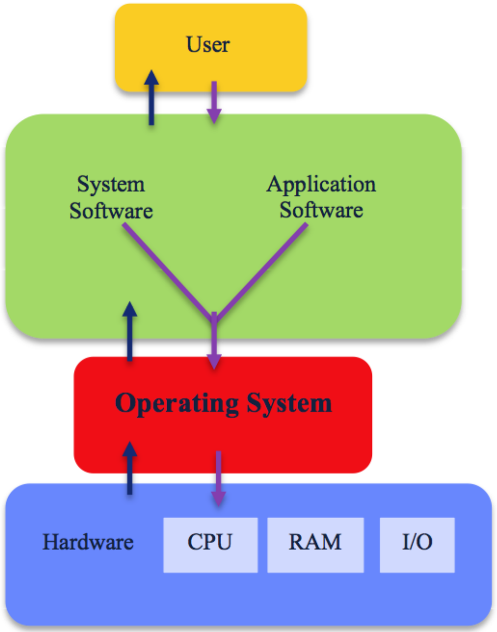
Warning notification!!!!!

- The PPTs are prepared for the offline interactive teaching in the class using the materials from different books and web. The author may not have the legal permission for online sharing those materials via social media/web/email or use in business etc in public domain.

- Therefore, students are requested not to share the PPTs outside the class/institute, which can violation the copy-write related issues.





Digital circuits are employed in the digital Systems e.g. digital computer, data communication and many other applications

⇩

To design the digital electronics circuits → requires knowledge about the methods/concept/procedure and tools

⇩

Digital logic design

Digital Logic Design (DLD)

LTPC 2-1-1-4

Objectives	The objective of this course is to understand the fundamentals of digital circuit design. This would begin with number representation and difference between analog and digital systems. Students will be able to analyze logical operations using combinational and sequential circuits together with Verilog implementation.
Unit 1	Number systems-Brief review of Digital systems, Binary numbers, Number base conversions, Representation of Negative Numbers, Complements, Binary arithmetic, Binary Codes for Decimal Numbers.
Unit 2	Boolean Algebra-Basic Definitions, Axiomatic Definition of Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Canonical and Standard Forms, Digital Logic Gates and timing concepts.
Unit 3	Gate level minimization-The Map Method - K-map 4 variable, Product of Sums Simplification, NAND and NOR Implementation, Other Two-Level Implementations. Review of, RTL, DTL, TTL, ECL, CMOS families
Unit 4	Design of Combinational Logic Circuits-Analysis Procedure, Design Procedure, Binary Adder-Subtractor, Parallel Adder, Carry look Ahead Adder, Binary Multiplier, Code Converters -Binary to Gray, Gray to Binary, BCD to Excess-3 Code Conversion and vice versa, BCD to 7-segment code converter, Magnitude Comparator-4 bit, Decoders, Encoders, Multiplexers, De-multiplexer, Parity generator and checker

Unit 5	Sequential logic circuits-Latches, Flip -Flops-SR, D, JK & T, realization of FFs, synchronous and asynchronous sequential circuits-State table and state diagrams, State reduction, Shift Registers-SISO, SIPO, PISO,PIPO, Design of counters-Modulo-n, Johnson, Ring, Up/Down, Design of Serial Adder, Serial Multiplier, FSM, Mealy and Moore state machines - State minimization – Sequence detection. Programmable devices-PAL and PLA
Unit 6	Introduction to Verilog-Verilog Implementation of combinational and sequential circuits

List of proposed experiments:

1. Introduction to Logisim
2. Study of Basic Gates and Universal gates
3. Half Adder, Half Subtractor, Full Adder and Full Subtractor
4. BCD to Seven Segment Display (simulation)
5. Decoder (2-to -4)
6. Encoder
7. SR and D Flip flop using NAND gates
8. Counter1 (Asynchronous)
9. Counter-2 (Synchronous)
10. BCD to Seven Segment Display (1 digit & 2 digit) using 7447 IC
11. Full Adder and Full Subtractor using 4*1 Multiplexer (74153)
12. SR and D Flip flop using NAND gates, 7474 IC and 7475 IC Verification
13. Programmable 1-bit ALU

Text books:	1. M. Morris Mano , Michael D. Ciletti, Digital Design With an Introduction to the Verilog HDL, VHDL, and System Verilog, Pearson Education; Sixth edition (18 May 2018).
References:	1. Samir Palnitkar, Verilog HDL, Pearson Education; Second edition (2003) 2. John F wakerly, Digital Design: Principles and Practices, Pearson, Pearson India; 4th edition (1 January 2008)

Course Outcomes	At the end of the course, students should have the ability: • To perform the conversion among different number systems; Familiar with basic logic gates -- AND, OR & NOT, XOR, XNOR to build simple logic circuits • To understand the basic properties of Boolean algebra and simplify Boolean functions • To optimize simple logic circuits using Karnaugh maps, understand "don't care“ • To design simple and complex combinational logics using basic gates • To understand sequential logic components: SR Latch, D Flip-Flop and their usage in practical applications. Able to experimentally implement with typical data path designs: Register, Adders, Shifters, Comparators; Counters, Multiplier, Arithmetic-Logic Units (ALUs) • To understand hardware description language (HDL) concepts and implement sequential circuit design based HDL and state table using D-FFs.

Mode of Evaluation	(%)
1. Mid-sem + Viva	(15+5)
2. End-sem + Viva	(25+5)
3. Surprise Quizzes	15
4. Schedule Quiz	15
5. Assignments	5
6. Lab	15

Simulation software	
Java	https://java.com/en/download/
Logisim	https://sourceforge.net/projects/circuit/

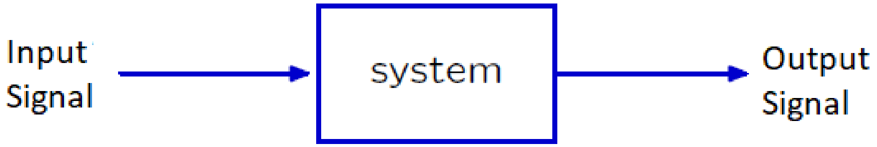
Warning notification!!!!!

- The PPTs are prepared for the offline interactive teaching in the class using the materials from different books and web. The author may not have the legal permission for online sharing those materials via social media/web/email or use in business etc in public domain.

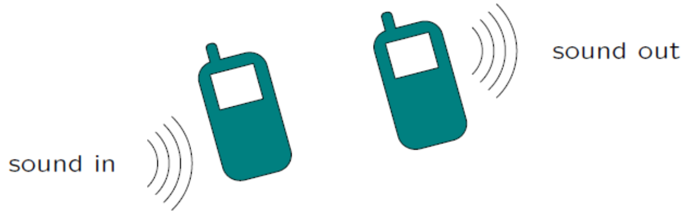
- Therefore, students are requested not to share the PPTs outside the class/institute, which can violation the copy-write related issues.

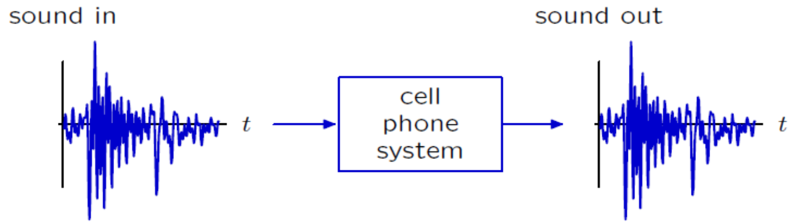
What is system?

- System is a device or algorithm which process or transforms an input signal into an desired output signal



$x(t)$ $h(.)$ $y(t) = h(x(t))$

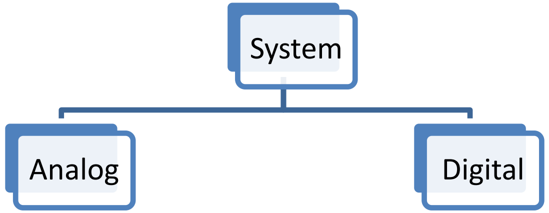




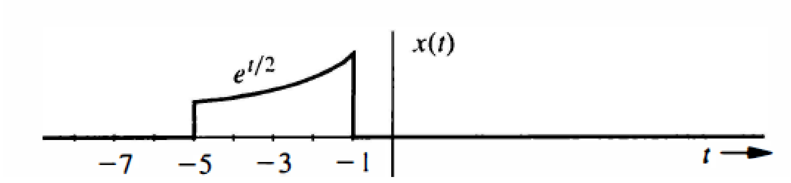
Examples:

$$y(t) = -4x(t),\ \frac{dy(t)}{dt} + 3y(t) = -\frac{dx(t)}{dt} + 6x(t),$$
$$y(n) - \frac{1}{2}y(n-2) = 3x(n) + x(n-2)$$

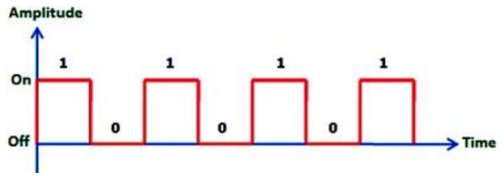
Analog versus Digital systems



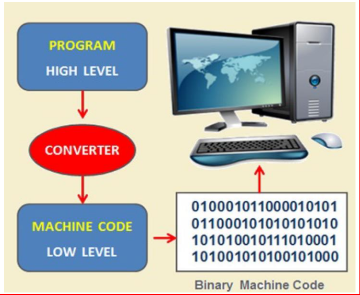
Analog systems: process time-varying signals that can take on any value across a continuous interval (a, b), *where* a *can be* $-\infty$ *and* b can be $+\infty$

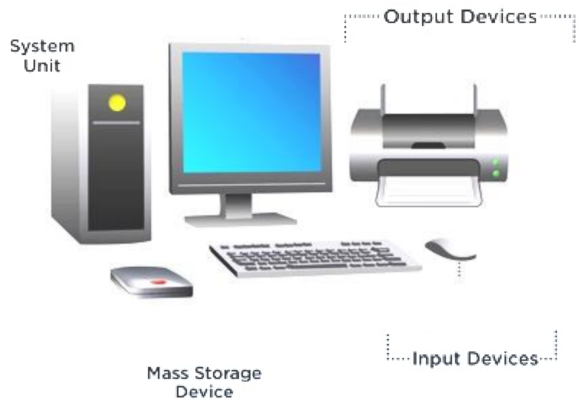


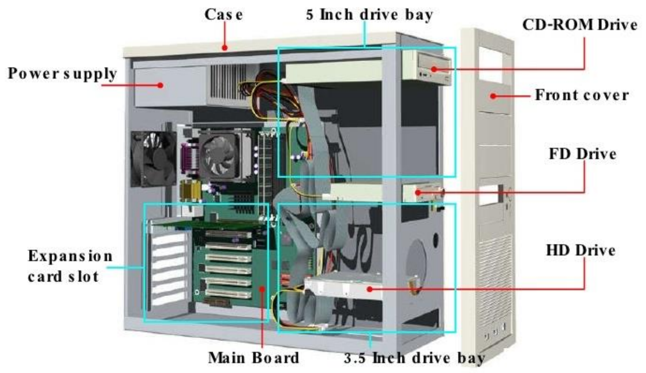
Digital systems: Digital systems are designed to store, process, and communicate information in digital form (in two states: 0 "or" 1 → also called binary)

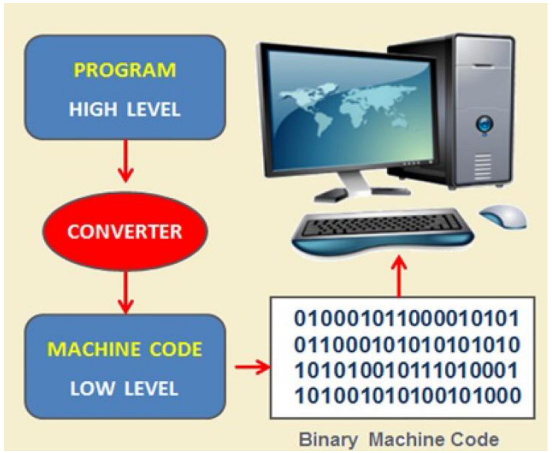


Example: Digital computer -- manipulates information in digital, or more precisely, binary form









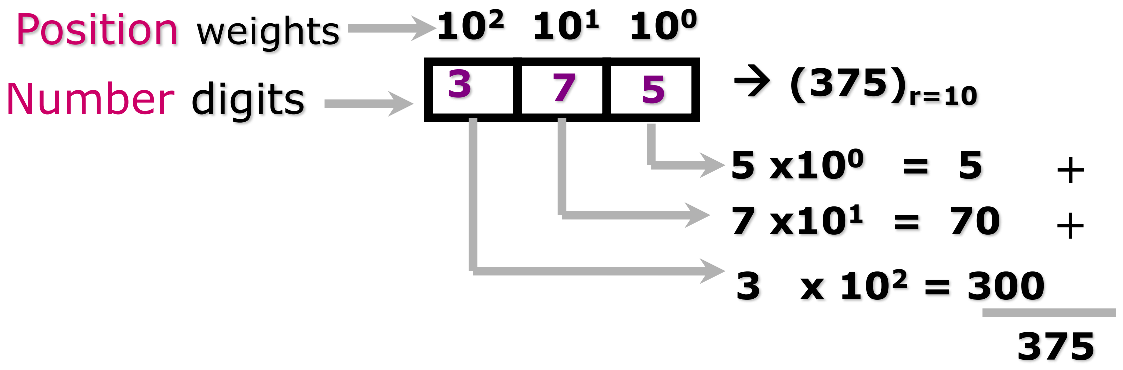
Computer programs contain instructions and data for performing a specific task. These programs, written using any programming language such as C++, must be translated into binary prior to execution by the computer. This is because the computer only understands binary

Number systems

- **Decimal** (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- **Binary** (0, 1)
- **Hexadecimal** (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
- **Octal** (0, 1, 2, 3, 4, 5, 6, 7)

Decimal number

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 $\longrightarrow$ Base/Radix (r) $\rightarrow$ 10



$(375.34)_{10} = 3 \times 10^2 + 7 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2}$

In general, $a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3}$ =

$$10^5a_5 + 10^4a_4 + 10^3a_3 + 10^2a_2 + 10^1a_1 + 10^0a_0 + 10^{-1}a_{-1} + 10^{-2}a_{-2} + 10^{-3}a_{-3}$$

Thank you