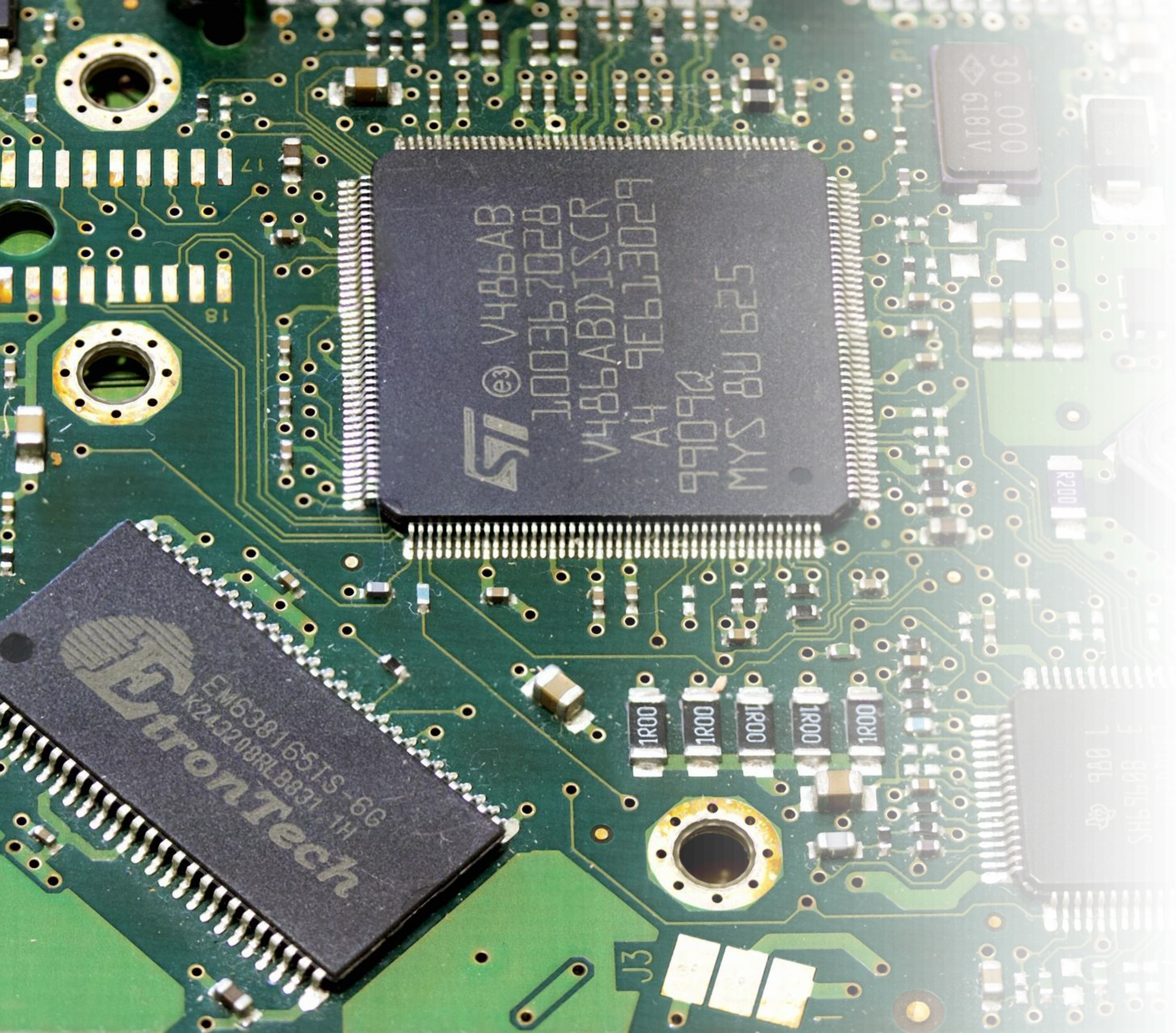


Digital Logic and System Design

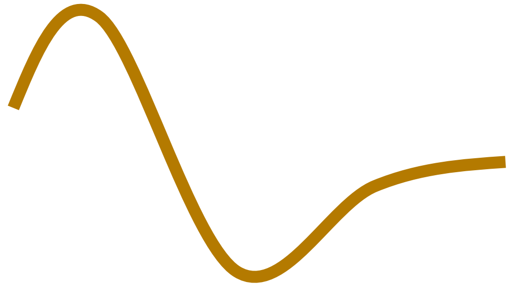
Lecture 2: Representation

COL215, I Semester 2022-2023
Venue: LHC 308
'E' Slot: Tue, Wed, Fri 10:00-11:00

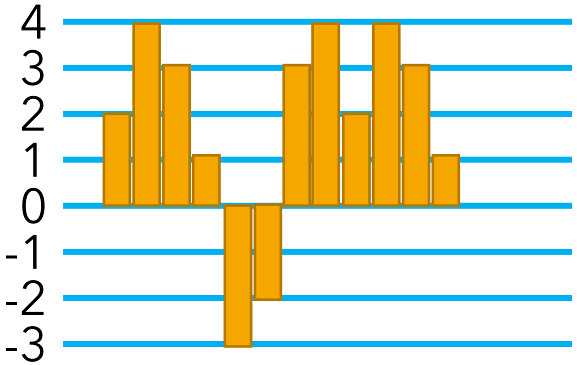
Instructor: Preeti Ranjan Panda
panda@cse.iitd.ac.in
www.cse.iitd.ac.in/~panda/
Dept. of Computer Science & Engg., IIT Delhi

A Digital System

- Represent and Manipulate **DISCRETE** Values
  - Instead of **CONTINUOUS** Values (Analog System)
- **FINITE** set of elements



Analog

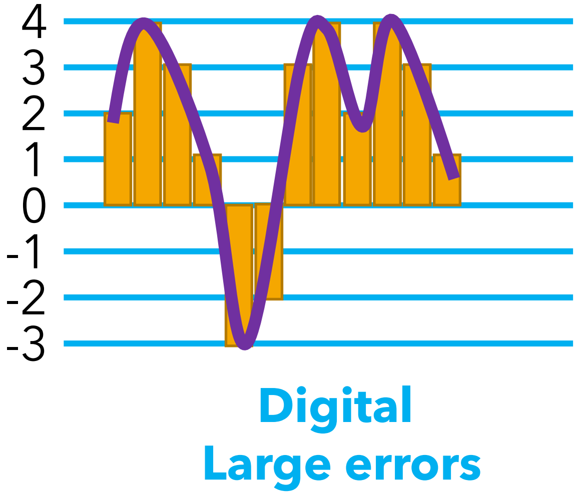


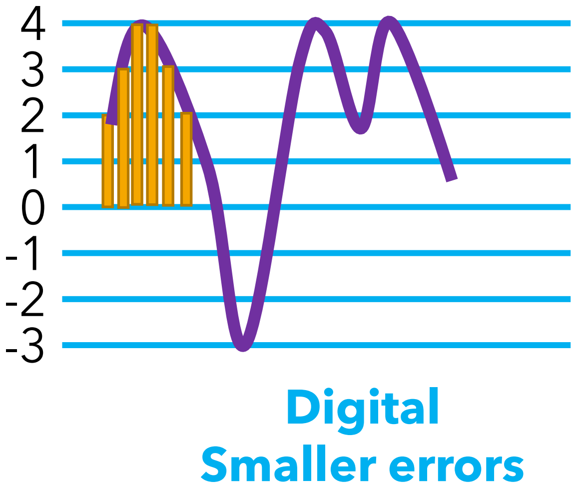
Digital

Why Digital?

- Information is lost! Why bother?
- Exact representation
- Reproducibility of results
  - E.g., fewer errors due to atmospheric conditions
- Ease of design
  - We'll see in this course!
- Sophisticated automation techniques
- High speed
- Low cost

Can we reduce the information loss?





Errors can be reduced by taking more data points

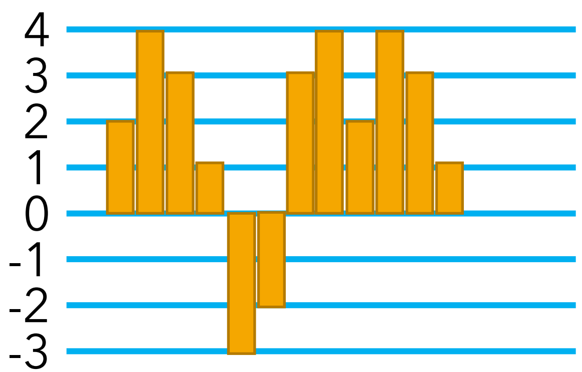
[Recall Fundamental Theorem of Integral Calculus]

Example Digital Systems

- Camera
  - Where is the digital element?
- Phone (over data connection)
  - What is digital about it?
- Computer
  - Was always digital

Representation

- Need ways to represent data
  - Store and Retrieve
  - Manipulate
- How do we represent the data on right?
- Sequence of **NUMBERS**



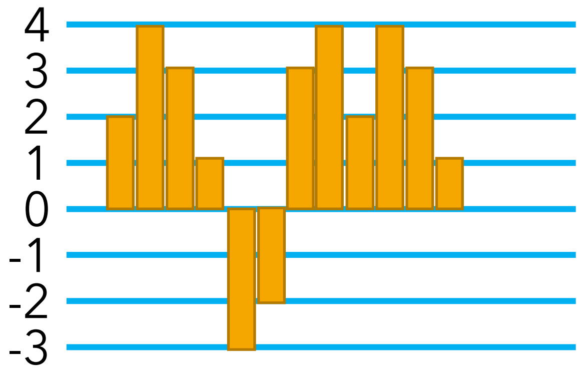
Digital

Representation:

[2, 4, 3, 1, -3, -2, 3, 4, 2, 4, 3, 1]

Representing a Number

- Can a number be represented **exactly**?
  - Integer?
  - Rational number?
  - Real number?
  - Complex number?
- Needs to be element of a **FINITE set**



Digital

Representation:

[2, 4, 3, 1, -3, -2, 3, 4, 2, 4, 3, 1]

Need to impose some restrictions

- Limited range
  - e.g., [-100, 200]
- Simple way:
  - **FIXED** number of digits
  - Each digit can take a **FIXED** number of values

Decimal Representation

- Number **3465** is a DECIMAL number
  - **base** is **10**
  - each **digit** of number $\in$ **{0,1,2,3,4,5,6,7,8,9}**
- Interpretation:

  $$\mathbf{3465} = \mathbf{3} \times 10^3 + \mathbf{4} \times 10^2 + \mathbf{6} \times 10^1 + \mathbf{5} \times 10^0$$

Other Bases

- We could represent the same number in a different **BASE** (also called **RADIX**)
  - E.g., Base **12**
  - in Base 12, each digit of number $\in$ **{0,1,2,3,4,5,6,7,8,9,10,11}**
  - $\mathbf{3465_{10}} = 2009_{12} = \mathbf{2} \times 12^3 + \mathbf{0} \times 12^2 + \mathbf{0} \times 12^1 + \mathbf{9} \times 12^0$
- ...or base **5**
  - in this base, each digit of number $\in$ **{0,1,2,3,4}**
  - $\mathbf{3465_{10}} = 102330_5 = \mathbf{1} \times 5^5 + \mathbf{0} \times 5^4 + \mathbf{2} \times 5^3 + \mathbf{3} \times 5^2 + \mathbf{3} \times 5^1 + \mathbf{0} \times 5^0$

Representing Integers in Arbitrary Bases

- Base **r**
- **n-digit** number
  - Digits $\mathbf{a_{n-1},...,a_2,a_1,a_0} \in \mathbf{\{0,1,2,...,r\text{-}1\}}$
- Interpretation of number in base r:

$$\mathbf{a_{n-1}} \times r^{n-1} + \mathbf{a_{n-2}} \times r^{n-2} + \ldots + \mathbf{a_2} \times r^2 + \mathbf{a_1} \times r^1 + \mathbf{a_0} \times r^0$$

Binary Numbers

- Binary number: **Base 2**
- n-digit number
  - Digits $a_{n-1,\dots,}a_{2,}a_{1,}a_0 \in \{0,1\}$
- Interpretation of number in base 2:

  $a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0$
- $1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

  $= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

  $= 8 + 4 + 1 = 13_{10}$
- Thus, 1101_2 is another way to represent thirteen

Which base should we use?

- Need **reliable** way to:
  - **Store** numbers
  - **Manipulate** numbers
- Decimal system:
  - need to find a way to represent 10 different entities for each digit
- Binary system:
  - find a way to represent 2 different things
- Modern digital systems: 2 voltage levels
  - 1 V (or 2V, etc.) represents '1'
  - 0 V represents '0'

13 or 15 or 1101?

13	15	1101
Decimal System	Octal System	Binary System

Choice based on engineering efficiency

- Should be easy/efficient to:
  - **Store/Retrieve** number
  - **Manipulate** numbers
- **Charge** stored on a capacitor
  - if capacitor is **charged**, a ‘**1**’ is stored
  - if capacitor is **discharged**, a ‘**0**’ is stored
  - Other physical phenomena could be used (e.g., magnetization direction)
- Since ANY number can be represented as a binary number, we have a way to store anything we want
- Binary is popular: **easier to distinguish between 2 values**
  - Exceptions: some memory types
  - Manipulation/computation usually in binary

Other Popular Bases

- Base **8** (Octal)
  - digits are: 0,1,2,3,4,5,6,7
- Base **16** (Hexadecimal)
  - digits are: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
  - 10=A, 11=B, 12=C, 13=D, 14=E, 15=F

Representing Real Numbers

- $\mathbf{3465.28} = \mathbf{3} \times 10^3 + \mathbf{4} \times 10^2 + \mathbf{6} \times 10^1 + \mathbf{5} \times 10^0 + \mathbf{2} \times 10^{-1} + \mathbf{8} \times 10^{-2}$
- $1101.11_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

$$= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25$$

$$= 8 + 4 + 1 + .5 + .25 = 13.75_{10}$$

Integral Part **Fractional Part**

Conversion between Number Systems

- Conversion from some base to decimal system easy
  - use equation
- Conversion from decimal system to other base
  - repeated division by base
  - keep track of remainders