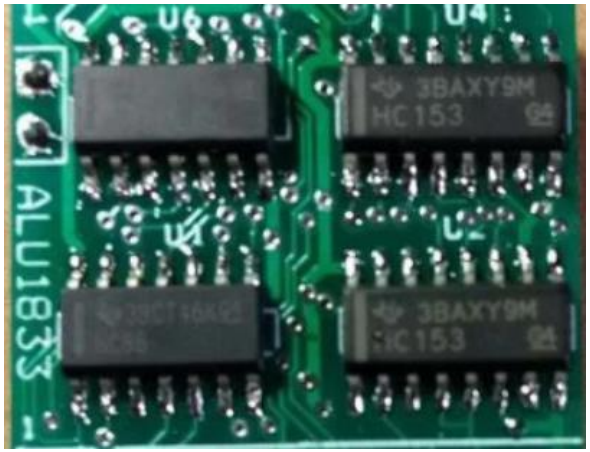
COMP S266F

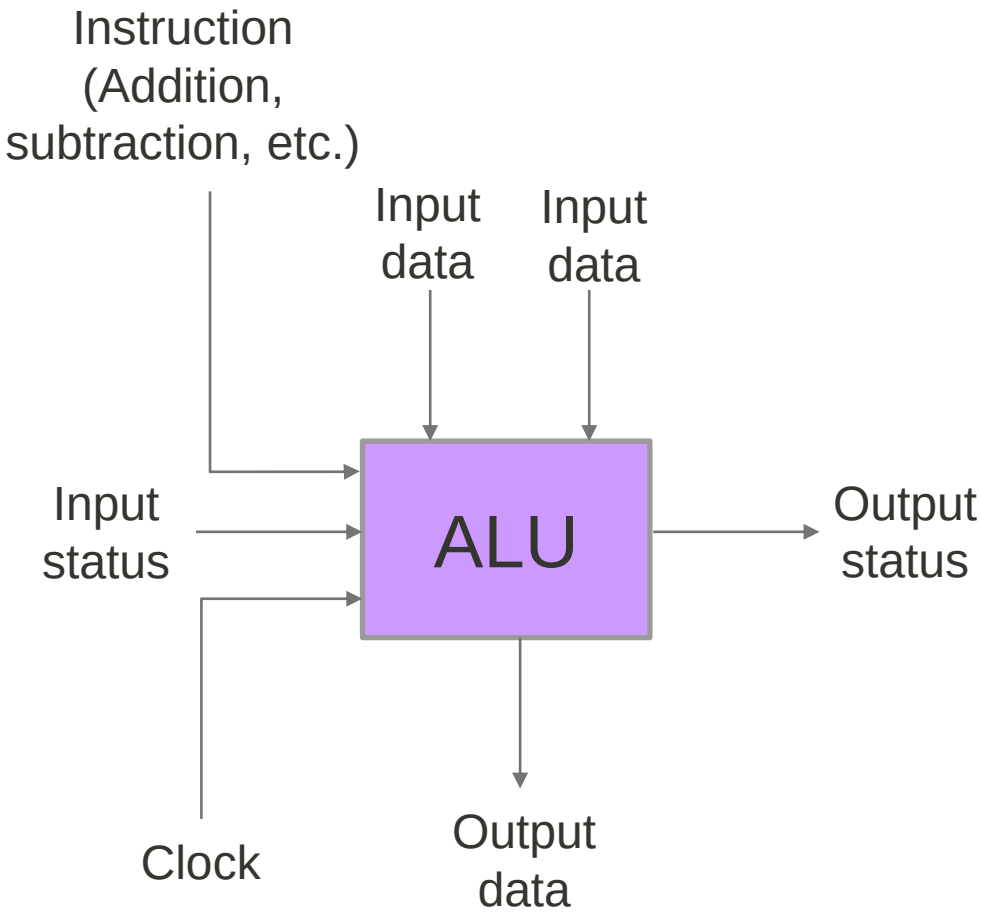
Computer Architecture

Chapter 2.1 Arithmetic and Logic Unit Design: Processing Integers



the arithmetic and logic unit

Arithmetic and Logic Unit (ALU)



Start from a basic ALU

- The requirements and trade-off of ALU for programmable computer
- Typical approaches of ALU designs
- Instruction: E.g. Addition and Subtraction
  - The details of the operation

ALU Requirements

- High Reliability
- Error Free
- Simple Design
- Efficient Operation
- Large Data Range
- Economical Cost

ALU Requirements Trade-off

- Cannot have everything desirable
- Some requirements are inter-related
  - E.g. high reliability means higher cost
- Some requirements are conflicting to each other
  - E.g. high reliability and simple design
- Importance of trade-off
  - Give away the lesser important desires and take the most important ones

Typical approaches of ALU designs

- Digital representation
  - to code data in electrical signals
- Analogue representation
  - To code data in a continuous range
- Binary numeral system
  - to code numbers for arithmetic and logic operation
- Two's complement binary representation
  - a variant of binary numeral system
  - for addition and subtraction of positive and negative numbers

Digital/Analogue Representation

- Physical attribute as a method to encode data
  - Usually represented in a physical attribute
  - The common physical attribute of electronic system is voltage
    - Voltage is continuous and scalar
- Two ways to represent using voltages:
  - Digital representation
    - greater reliability and error tolerance
  - Analogue representation
    - Better flexibility in representing a wide range of possible values
    - Noise sensitive

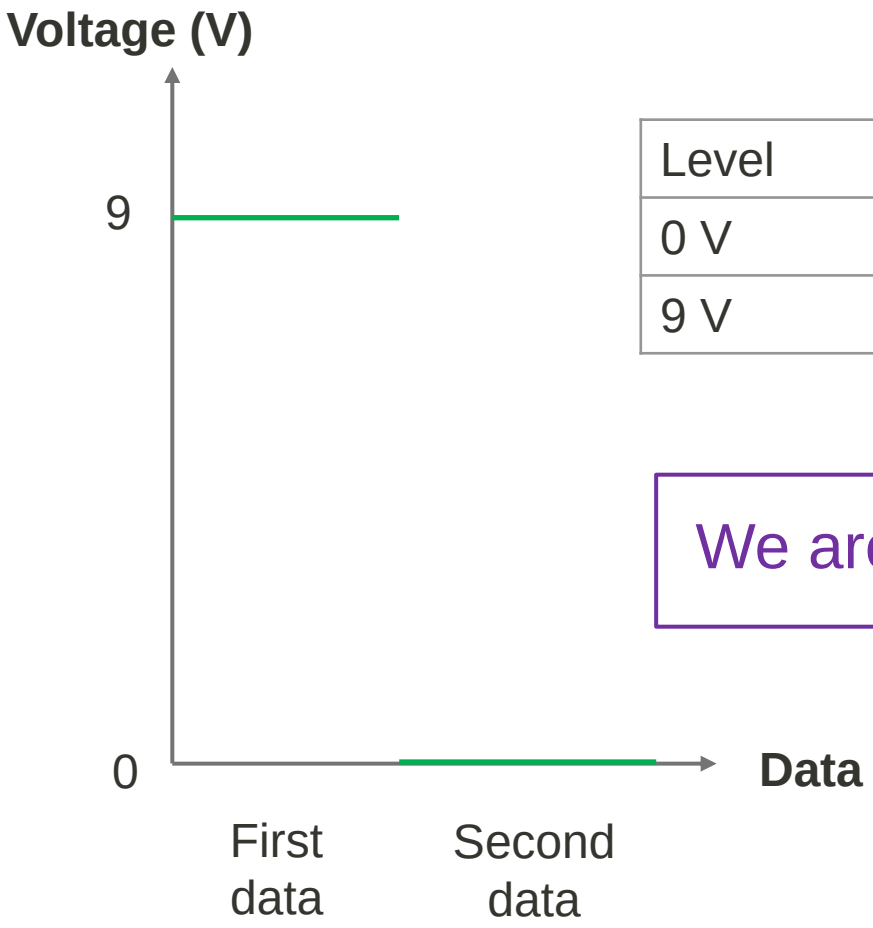
digital representation

Digital Representation

- Digital representation is discrete
- Voltage range is divided into levels
- Sufficiently small fluctuation is acceptable
  - The small change keeps the signal at the same level, e.g. round the value to the nearest level
- Less prone to errors

Digital Representation – Example 1

- Two-level (Binary) Representation



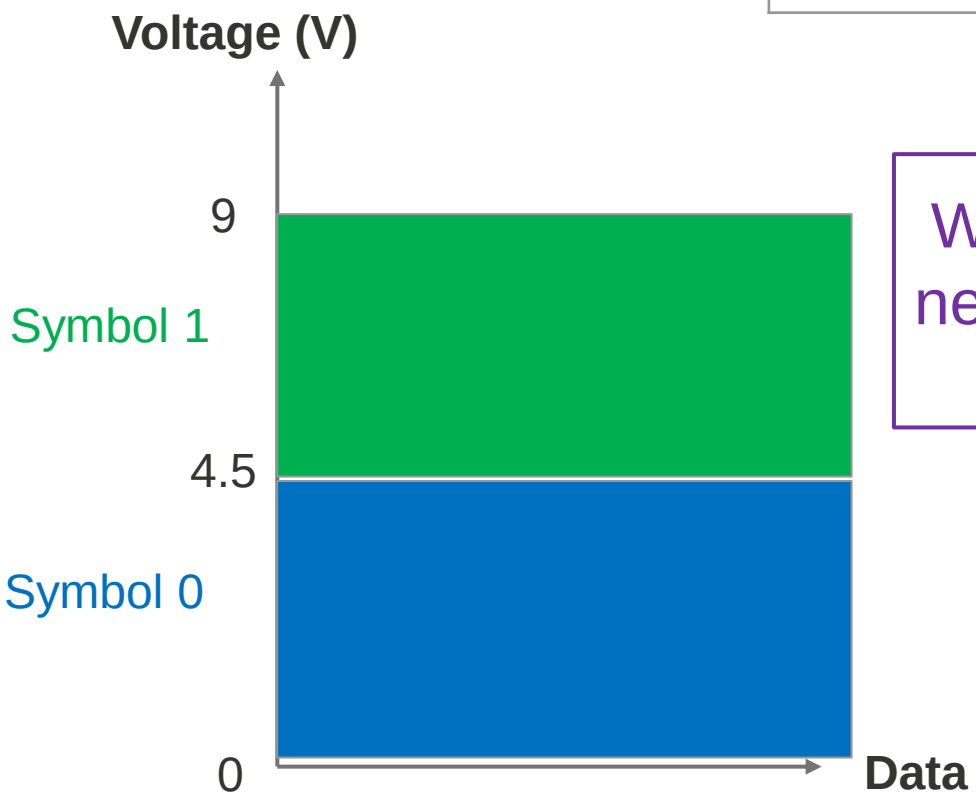
Level	Symbol
0 V	0
9 V	1

We are receiving data stream 10

Digital Representation – Example 1

- Error tolerance

Voltage (V)	Symbol
[0,4.5)	0
>=4.5	1

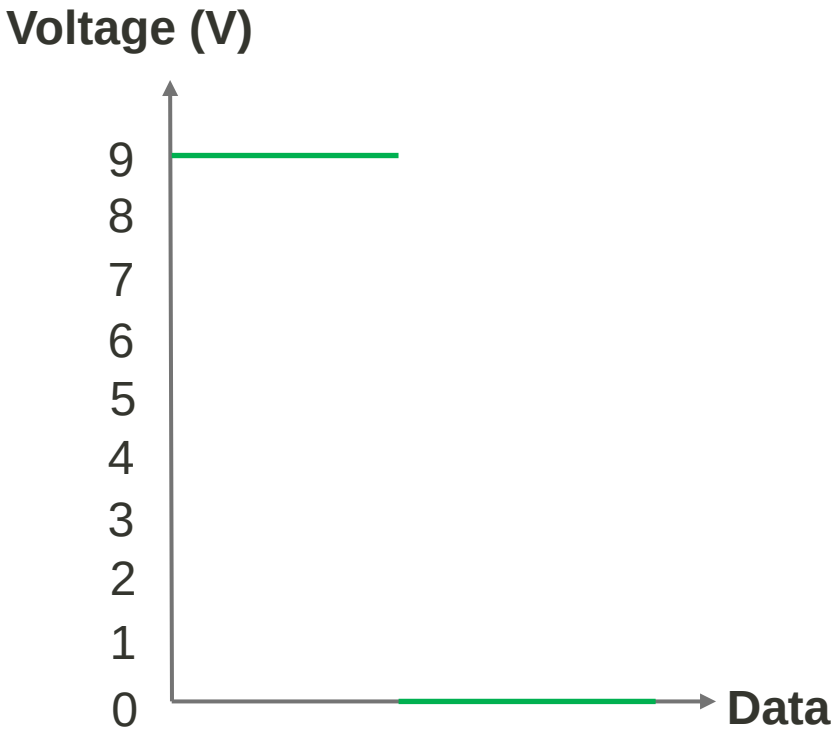


We could round the values to nearest levels to withstand the fluctuation of signal

Digital Representation – Example 2

- Ten-level Representation

Level	Symbol
0 V	0
1 V	1
2 V	2
3 V	3
4 V	4
5 V	5
6 V	6
7 V	7
8 V	8
9 V	9



We are receiving data stream 90

Digital Representation – Example 2

- Ten-level Representation

More levels
smaller error tolerance

Level	Symbol
[0,0.5) V	0
[0.5,1.5) V	1
[1.5, 2.5) V	2
[2.5, 3.5) V	3
[3.5, 4.5) V	4
[4.5, 5.5) V	5
[5.5, 6.5) V	6
[6.5, 7.5) V	7
[7.5, 8.5) V	8
>= 8.5 V	9

- Much smaller margins
- 0.5 for Symbols 0
- 1 for Symbols 1-8

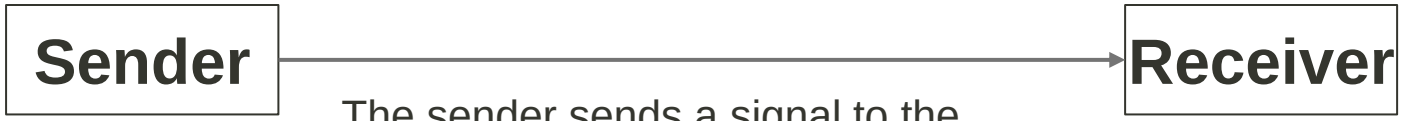
Digital Representation

- Question: An electronic system can support voltage from 0V to 5V. If we design a system that requires the representation of integers from 0 to 10, how should we assign the voltage levels to the values?

analogue representation

Analogue Representation

Data value is represented by the voltage level in a continuous manner



The sender sends a signal to the receiver by setting a voltage to the line

For example, we use 0.01 V to represent 1. Thus, 1 V represents 100.

A data value 325 is sent by the sender, using a voltage of 3.25 V

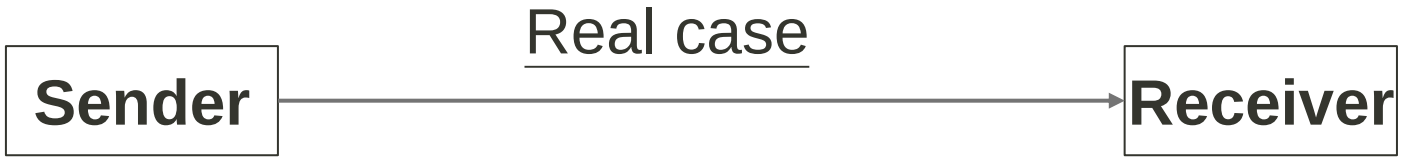
Ideal case

The receiver reads from the line and got 3.25 V. Therefore, receiver interprets the data is 325.

Limitations of Analogue Representation

In transmission, the voltage level may be changed by **noise interference** or simply the **signal degradation (loss)** due to a long signal line

A data value 325 is sent by the sender, using a voltage of 3.25 V



The receiver reads from the line and got 3.24 V. Therefore, receiver interprets the data is 324

Wrong

The major limitations are:

- **Not tolerant to noise**
- **Not tolerant to other forms of signal degradation (loss)**

binary numeral system

Scenario 1

- How would you represent the following message in a computer?

GOOD GOOD BAD BAD

- Hint: how many digital levels (symbols) do we need?

- X to represent ‘GOOD’
- O to represent ‘BAD’
- The above message can then be represented as: XXOO

Scenario 2 (using three-level digital)

- How would you represent the following message in a computer:
  - GOOD GOOD BAD BAD NEUTRAL
- How many symbols (digital levels) do we need?
  - X to represent GOOD
  - O to represent BAD
  - # to represent NEUTRAL
  - The above message can then be represented as: XXOO#
- Are there other alternatives?
  - Can we use two digital levels (i.e. 2 symbols)?

Scenario 2 (using two-level digital)

- XX to represent GOOD
- XO to represent BAD
- OO to represent NEUTRAL
- The message GOOD GOOD BAD BAD NEUTRAL can now be represented as:
  - XXXXXOXOOO

Scenario 2 (three-level v.s. two-level)

- Comparing the two solutions:
  - XXOO# (P. 21)
  - XXXXXOXOOO (P. 22)
- What's the main tradeoff of these two solutions?

Main Approaches of ALU Design

- Create a data representation system for the ALU
  - Digital representation to code data in electrical signals.
  - **Binary numeral system** to code numbers for arithmetic and logic operations
  - Two's complement binary representation for addition and subtraction of positive and negative numbers

Binary numeral system

- Binary numeral system uses two symbols to represent data '0' and '1'
  - 2-level digital representation
    - Most error tolerant
    - Less technically challenging

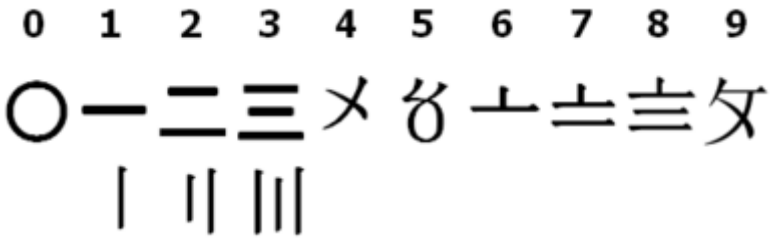
What is a numeral system?

- Numeral system supports a systematic and consistent set of rules to represent numbers
  - Examples: decimal, binary, and hexadecimal
- Key characteristics:
  - Define a set of numbers such as integers or positive numbers.
  - Provide each number in the number set a unique representation.
  - Contain of a set of unique symbols
    - Decimal: the ten symbols are 0, 1, 2, ..., 9
  - Provide rules for combining symbols to represent a larger range of numbers

The Base of a Numeral System

- Number of unique symbols used in the system

Base	Numeral System
2	Binary
3	Ternary
8	Octal
10	Decimal
12	Duodecimal
16	Hexadecimal
20	Vigesimal
60	Sexagesimal



The Base of a Numeral System

- Binary numeral system has 2 symbols 0 and 1
  - Single digit binary number can represent the values from 0 to 1 (decimal) only
- Hexadecimal numeral system has 16 symbols, from 0, 1, 2, to 9, A, B, C, D, E, and F
  - Single digit hexadecimal number can represent the values from 0 to 15 (decimal)
  - Need to support more symbols
  - Represent a greater range of values with same number of digits
- What's the tradeoff between base-2 and base-16?

Positional Notation

- Positional notation provides rules for combining symbols to represent a larger range of numbers
  - Each digit is related to the next by a multiplier
    - Multiplier is the base or the radix of the number system
  - Number is the sum of all digits multiplied by their multipliers
  - Larger numbers constructed by putting symbols together

Positional Notation (Example 1)

- Why is the number 3456 representing the value 3456 in the decimal numeral system?

Digits	**3**	**4**	**5**	**6**
Representation	10^3 or 1000	10^2 or 100	10^1 or 10	10^0 or 1
Value of digit	3000	400	50	6

- Total value is the summation of the contribution of each digit, which is 3000 + 400 + 50 + 6 = 3456

Positional Notation (Example 2)

- Octal system has 8 symbols from 0 to 7. In the same way, we can sue positional notation to represent larger values in octal. What is the value of the octal number 2476 in decimal?

Digits	**2**	**4**	**7**	**6**
Representation	8^3 or 512	8^2 or 64	8^1 or 8	8^0 or 1
Value of digit	1024	256	56	6

- Total value is the summation of the contribution of each digit, which is 1024 + 256 + 56 + 6 = 1342 (decimal)

Positional Notation

- Range of numbers representable depends on:
  - Number of symbols
  - Number of digits

Base	Numeral system	No. of symbols	Range represented by 8 digits	Range in decimal
2	Binary	2	00000000 to 11111111	0 to 255
3	Ternary	3	00000000 to 22222222	0 to 6560
8	Octal	8	00000000 to 77777777	0 to 16777215
10	Decimal	10	00000000 to 99999999	0 to 99999999
16	Hexadecimal	16	00000000 to FFFFFFFF	0 to 4294967295

Positional Notation

- More about binary and decimal numeral systems

Number of digits	Range	Maximum (Binary)	Maximum (Decimal)
1	0 to 2^1 -1	1	1
8	0 to 2^8 -1	11111111	255
16	0 to 2^{16} -1	1111111111111111	65535
32	0 to 2^{32} -1	11111111111111111111111111111111	4294967295

Representing Negative Numbers

- Adding a new symbol
  - For example: the '-' symbol
    - -10
    - 8
  - Cost: an additional symbol for the system
- Use a designated digit
  - For example: designate the left most position
    - 0111 (binary) represents -7 (decimal)
    - 1111 (binary) represents 7 (decimal)
  - Cost: an additional digit in every number

0 -> negative
1 -> positive

Brief Summary

- Use 2-level digital representation
  - Most error tolerant representation
- Binary numeral system uses two symbols 0 and 1 and the positional notation to represent positive values.
  - A digit is a bit
  - There are 8 bits in a byte
  - An 8-bit binary number can represent 256 different values
  - If the smallest value is 0, then the range is from 0 to 255