Binary Data

Assis. Prof. Dr. MOHAMMED ELAMASSIE.
IEEE Senior Member | Optica Senior Member | Optica Travelling Lecturer.
Executive Co-Director of OKATEM.
Executive Co-Director of CT&T Research Group.

Spring 2024-25

Outlines

- Boolean Algebra.
- Binary numbers
- Numbers with different bases.

Boolean Algebra

Boolean Algebra

- **Boolean algebra** is a branch of mathematics and mathematical logic that deals with variables that have two possible values: **true (1)** or **false (0)**.
- It's named after George Boole, who first formulated it.
- Boolean algebra is extensively used in computer science, particularly in digital electronics and computer programming, for designing and analyzing digital circuits, logical operations, and conditional expressions.
- In MATLAB, logical operations and Boolean algebra are fundamental concepts. MATLAB has built-in support for logical operations, and you can perform Boolean algebraic operations on arrays of logical values.

We have mentioned them before in the previous lectures. Here, we will add more details including Truth tables.

Basic Definitions

- Logical AND (&): Returns true only if both operands are true.

  Example:
  ```
  A = [1, 0, 1];
  B = [1, 1, 0];
  result = A & B; % [1, 0, 0]
  ```

- Logical OR (|): Returns true if at least one of the operands is true.

  Example:
  ```
  A = [1, 0, 1];
  B = [1, 1, 0];
  result = A | B; % [1, 1, 1]
  ```

- Logical NOT (~):Negates the input value; true becomes false, and false becomes true.

  Example:
  ```
  A = [1, 0, 1];
  result = ~A; % [0, 1, 0]
  ```

Boolean Algebra Expressions

You can create more complex Boolean expressions by combining basic logical operations.

Example 1:

```
% Expression: (A AND B) OR (~A AND ~B)
A = [1, 0, 1];
B = [1, 1, 0];
result = (A & B) | (~A & ~B); % [1, 0, 1]
```

Example 2:

```
% Expression: A AND (B OR ~C)
A = [1, 0, 1];
B = [1, 1, 0];
C = [0, 1, 1];
result = A & (B | ~C); % [1, 0, 0]
```

Truth Table Generation

You can generate **Truth Tables** for Boolean expressions using MATLAB to list all possible combinations of input values along with the corresponding output.

Example

```
% Truth Table for (A AND B) OR (~A AND ~B)
A = [0, 0, 1, 1];   B = [0, 1, 0, 1];
result = (A & B) | (~A & ~B);
truth_table = [A', B', result'];      disp(truth_table);
```

- **For better demonstration, you can use the function table**

```
% Define the inputs
A = [0, 0, 1, 1];      B = [0, 1, 0, 1];

% Define the output based on the expression: (A AND B) OR (~A AND ~B)
output = (A & B) | (~A & ~B);

% Create a table with inputs and output
truth_table = table(A', B', output', 'VariableNames', {'A', 'B', 'Output'});

% Display the truth table
disp(truth_table);
```

* Replace (A & B) | (~A & ~B) by not(xor(A,B)) and check the results

EXAMPLES (1/4)

Write a MATLAB code to simulate a simple digital logic circuit that implements a 2-input AND gate. Your code should prompt the user to enter two binary inputs (0 or 1) and then display the output of the AND gate based on these inputs.

Instructions:

1. Prompt the user to enter two binary inputs (0 or 1).
2. Perform the logical AND operation on the inputs.
3. Display the result of the AND gate operation as the output.

Your code should handle invalid input.

```
clear all;   close all; clc
% Prompt the user to enter two binary inputs
input1 = input('Enter first binary input (0 or 1): ');
input2 = input('Enter second binary input (0 or 1): ');

% Check if inputs are valid (0 or 1)
if ~(input1 == 0 || input1 == 1) || ~(input2 == 0 || input2 == 1)
        disp('Invalid input. Please enter 0 or 1 for binary inputs.');
else
        % Perform logical AND operation
        output = input1 & input2;
        % Display the result
        fprintf('Output of the AND gate: %d\n', output);
end
```

EXAMPLES (2/4)

I will use here while loop to ask the user to re-enter the numbers if they are not valid

```
clear all; close all; clc

while true
    % Prompt the user to enter two binary inputs
    input1 = input('Enter first binary input (0 or 1): ');
    input2 = input('Enter second binary input (0 or 1): ');


    % Check if inputs are valid (0 or 1)
    if ~(input1 == 0 || input1 == 1) || ~(input2 == 0 || input2 == 1)
        clc
        disp('Invalid input. Please enter 0 or 1 for binary inputs.');
    else
        % Perform logical AND operation
        output = input1 & input2;
        % Display the result
        fprintf('Output of the AND gate: %d\n', output);
        break
    end
end
```

EXAMPLES (3/4)

Write a MATLAB code to simulate a digital logic circuit that implements a 3-input OR gate. Your code should prompt the user to enter three binary inputs (0 or 1) and then display the output of the OR gate based on these inputs.

Instructions:

1. Prompt the user to enter three binary inputs (0 or 1 for each input).
2. Perform the logical OR operation on the inputs.
3. Display the result of the OR gate operation as the output.

To enhance the challenge, you should implement the logical OR operation using nested logical operations (i.e., combining multiple OR operations).
Your code should handle invalid input (i.e., inputs other than 0 or 1).

EXAMPLES (4/4)

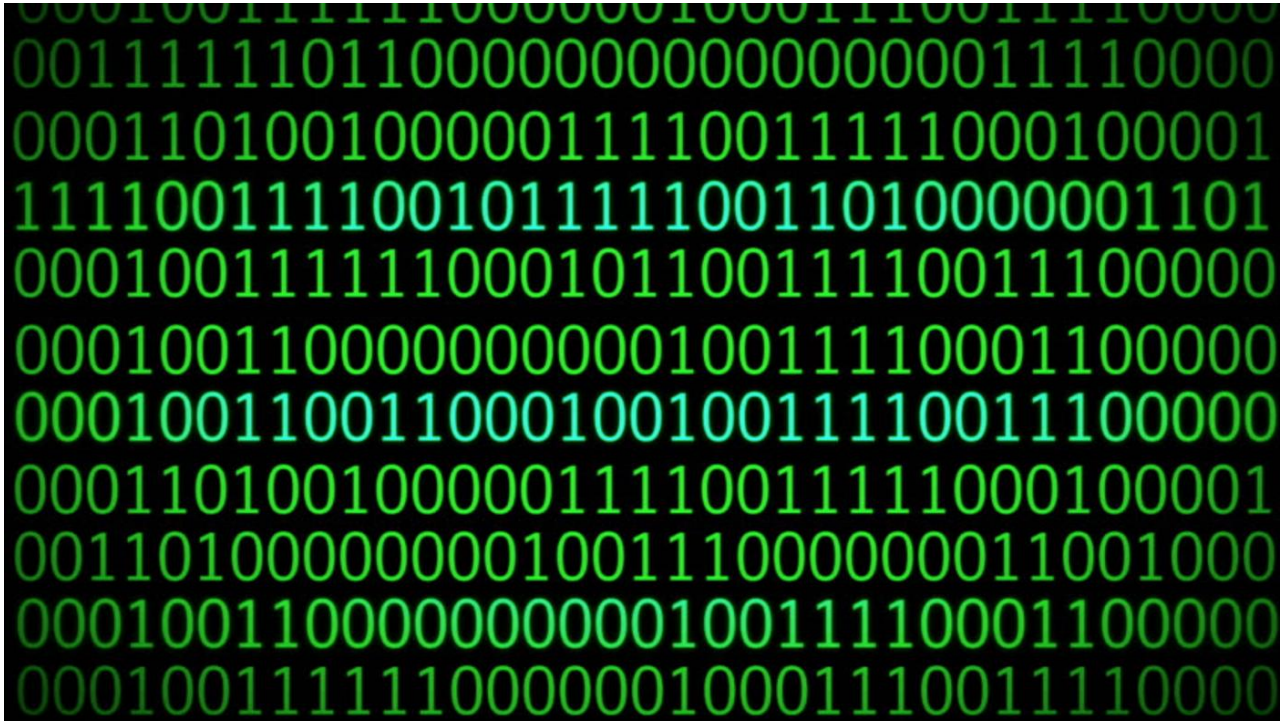
```
clear all; close all; clc;

% Prompt the user to enter three binary inputs
while true
    input1 = input('Enter first binary input (0 or 1): ');
    input2 = input('Enter second binary input (0 or 1): ');
    input3 = input('Enter third binary input (0 or 1): ');
    % Check if inputs are valid (0 or 1)
    if ~(input1 == 0 || input1 == 1) || ~(input2 == 0 || input2 == 1) || ~(input3 ==
    0 || input3 == 1)
        clc;
        disp('Invalid input. Please enter 0 or 1 for binary inputs.');
    else
        % Perform logical OR operation using nested logical operations
        output = (input1 | input2) | input3;
        % Display the result
        fprintf('Output of the OR gate: %d\n', output);
        break;
    end
end
```

Binary Numbers

Binary Numbers

- In MATLAB, binary numbers are represented using the binary numeral system, which consists of only two digits: 0 and 1.
- MATLAB provides various functions and operations to work with binary numbers.



Representing Binary Numbers

1. Binary Literal Representation: You can directly represent binary numbers in MATLAB using the prefix '0b' followed by the binary digits.
   Example
   binary_num = 0b1010; % Represents the binary number 1010

- See the output, it is 10 since 1010_2 =10

2. Using Binary Strings: Binary numbers can also be represented as strings of characters '0' and '1'.
   Example
   binary_str = '1010'; % Represents the binary number 1010

Converting Binary Numbers

1. Binary to Decimal Conversion: MATLAB provides the 'bin2dec' function to convert binary numbers to decimal.
   Example
   ```
   binary_str = '1010';
   decimal_num = bin2dec(binary_str); % Convert binary to decimal: 10
   ```

2. Decimal to Binary Conversion: The 'dec2bin' function converts decimal numbers to binary.
   Example
   ```
   decimal_num = 10;
   binary_str = dec2bin(decimal_num); % Convert decimal to binary: '1010'
   ```

Binary Arithmetic

1. Binary Addition: MATLAB can perform binary addition using regular addition operations.

   Example

```
clear all; close all; clc;
binary_num1 = 0b1010;
binary_num2 = 0b1100;
result_decimal = binary_num1 + binary_num2 % Decimal addition: 22
result_binary = dec2bin(result_decimal)
```

1. Binary Subtraction: MATLAB can perform binary addition using regular addition operations.

   Example

```
clear all; close all; clc;
binary_num1 = (0b1100);
binary_num2 = (0b1010);
result_decimal = binary_num1 - binary_num2 % Decimal subtraction: 2
result_binary = dec2bin(result_decimal) % Convert decimal to binary: '10'
```

Bitwise Operations

- Bitwise operations are fundamental operations that manipulate individual bits within binary representations of numbers.
- In MATLAB, bitwise operations can be performed using the bitwise logical functions ('bitand', 'bitor', 'bitxor', 'bitcmp') and the bit shift functions ('bitshift', 'bitset', 'bitget').

Examples

```
clear all; close all; clc;
% Bitwise AND
a = 5; % binary: 101
b = 3; % binary: 011
result_and = bitand(a, b) % result: 1 (binary: 001)

% Bitwise OR
result_or = bitor(a, b) % result: 7 (binary: 111)

% Bitwise XOR
result_xor = bitxor(a, b) % result: 6 (binary: 110)
```

Numbers with Different Bases

Numbers with Different Bases (1/2)

- In MATLAB, you can work with numbers in different bases, such as **binary**, **decimal**, **hexadecimal**, and **octal**.
- MATLAB provides functions to convert numbers between different bases and perform arithmetic operations on them.

Example 1

```
dec_num = 42;
bin_num = dec2bin(dec_num);
disp(bin_num) % Output: 101010
```

Example 2

```
bin_num = '101010';
dec_num = bin2dec(bin_num);
disp(dec_num) % Output: 42
```

Numbers with Different Bases (2/2)

Example 3

```
dec_num = 255;
hex_num = dec2hex(dec_num);
disp(hex_num) % Output: FF
```

Example 4

```
hex_num = 'FF';
dec_num = hex2dec(hex_num);
disp(dec_num) % Output: 255
```

Arithmetic Operations with Different Bases (1/3)

- MATLAB supports arithmetic operations on numbers in different bases. When performing arithmetic, MATLAB implicitly converts inputs to decimal, so you may need to convert results back to the desired base.

Example 1: Addition in binary

```
bin_num1 = '1010';
bin_num2 = '1100';
sum_bin = bin2dec(bin_num1) + bin2dec(bin_num2);
disp(dec2bin(sum_bin)) % Output: 10110
```

Example 2: Multiplication in hexadecimal

```
hex_num1 = 'FF';
hex_num2 = '2';
prod_hex = hex2dec(hex_num1) * hex2dec(hex_num2);
disp(dec2hex(prod_hex)) % Output: 1FE
```

Arithmetic Operations with Different Bases (2/3)

- Defining that the base is 8 when you convert from octal or to octal

Example 1

```
octal_num = '52';
dec_num = base2dec(octal_num, 8); % Convert octal to decimal
disp(dec_num) % Output: 42
```

Example 2

```
dec_num = 42;
octal_num = dec2base(dec_num, 8); % Convert decimal to octal
disp(octal_num); % Output: 52
```

Homework
Questions on
Logic gates and
Boolean functions

Homework for both EE 532 and EE 432

- **Logic Gates and Boolean Functions**

  Write MATLAB codes to implement the Boolean functions in the form of

  F(A, B, C) = A AND (B OR C)

  F(A, B, C) = (A AND B) XOR (NOT C)

using logic gates (AND, OR).

- **Number with Different Basis**

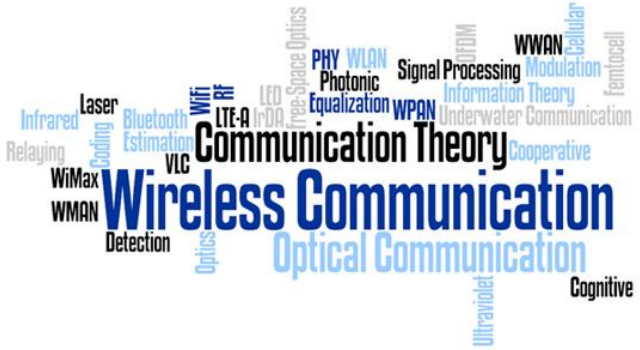
  1- Write code to convert a decimal number into its binary representation.

  2- Write code to convert a binary number into its hexadecimal representation.

  3- Write code to convert a hexadecimal number into its binary representation

Note: You need to include the following:

1. Prompt the user to enter the required input.
2. Ensure that the input provided is valid.
3. If the input is invalid, prompt the user to enter again until valid input is received, using a while loop.
4. Valid input criteria should be specified according to the requirements of each task (e.g., valid input ranges for binary numbers, decimal numbers, etc.).



Thanks for Your Attention…

Assis. Prof. Dr. Mohammed Elamassie
mohammed.elamassie@ozyegin.edu.tr

KATEM
CENTER OF EXCELLENCE IN OPTICAL WIRELESS COMMUNICATION TECHNOLOGIES

ÖZYEĞİN UNIVERSITY