**FORMAN CHRISTIAN COLLEGE (A CHARTERED UNIVERSITY)**

****

**CSCS306 - A**

**FA24**

**Lab 2**

**Hafsah Shahbaz – 251684784**

**Daim Bin Khalid - 251686775**

**Introduction**

In this lab we write an Arduino program that takes a binary number as input and performs several operations, including converting the binary number to its decimal equivalent, finding its complement, and retrieving the binary digit at a specified bit position. This program enhances understanding of binary representations and basic data manipulation in embedded systems.

**Functions and Libraries**

**Functions**

* **getSize()**: Retrieves the size of the binary array from the user input.
* **displayArray(int arr[], int size)**: Displays the contents of the binary array on the Serial Monitor.
* **populateArray(int arr[], int size)**: Populates the binary array with user-entered values (0s and 1s).
* **getDecimalValue(int arr[], int size)**: Computes the decimal value of the binary number stored in the array.
* **getComplement(int arr[], int complNumber[], int size)**: Computes the binary complement of the input binary number.
* **getBitPos()**: Gets the bit position from the user.
* **readBit(int arr[], int size, int p)**: Retrieves the binary value at a specified position.

**Algorithm and Logic**

1. **Setup Phase**:
   * Initialize serial communication.
   * Display the lab title.
2. **Input Phase**:
   * Prompt the user to enter the size of the binary array and read the input using getSize().
   * Create two arrays: one for the binary number (binNumber) and one for its complement (complNumber).
3. **Population Phase**:
   * Prompt the user to enter binary values and populate binNumber using populateArray().
4. **Conversion Phase**:
   * Call getDecimalValue() to convert the binary array to decimal and display the result.
   * Compute the complement of the binary number with getComplement() and display it.
5. **Bit Retrieval Phase**:
   * Ask the user for a bit position, read the input with getBitPos(), and retrieve the corresponding bit using readBit().
6. **Output**:
   * Display all relevant results, including the decimal value, complement, and specified binary digit.

**Code Breakdown**

int getNumber();

int getSize();

void displayArray(int arr[], int size);

void populateArray(int arr[], int size);

float getDecimalValue(int arr[], int size);

void getComplement(int arr[], int complNumber[], int size);

int getBitPos();

int readBit(int arr[], int size, int p);

int size;

char n;

char arr[5];

int num;

**Function Declarations**: Declare all functions, variables for array size, input characters, and numbers.

void setup()

{

Serial.begin(9600);

Serial.println("\t=====LAB 01=====");

...

}

**Setup Function**: Initializes the Serial Monitor and prompts for user inputs, manages the main logic of the program.

int getSize()

{

while (Serial.available() == 0) {}

...

}

**getSize()**: Waits for user input and converts it to an integer.

void populateArray(int arr[], int size)

{

int i = 0;

while (i < size) { ... }

}

**populateArray()**: Fills the array with binary digits, checking for valid input (0 or 1).

void displayArray(int arr[], int size)

{

for (int i = 0; i < size; i++) { ... }

}

**displayArray()**: Loops through the array and prints each value to the Serial Monitor.

float getDecimalValue(int arr[], int size)

{

float decimalValue = 0;

...

}

**getDecimalValue()**: Converts the binary array to its decimal equivalent using binary-to-decimal conversion logic.

void getComplement(int arr[], int complNumber[], int size)

{

for (int i = 0; i < size; i++) { ... }

}

**getComplement()**: Calculates the complement by flipping 0s to 1s and vice versa.

int readBit(int arr[], int size, int p)

{

return arr[size - 1 - p];

}

**readBit()**: Accesses the specific bit in the binary array based on the provided position.

**CODE**

int getNumber();

int getSize();

void displayArray(int arr[], int size);

void populateArray(int arr[], int size);

float getDecimalValue(int arr[], int size);

void getComplement(int arr[], int complNumber[], int size);

int getBitPos();

int readBit(int arr[], int size, int p);

int size;

char n;

char arr[5];

int num;

void setup()

{

*Serial*.begin(9600);

*Serial*.println("\t=====LAB 01=====");

*Serial*.println();

    // get array size

*Serial*.print("Enter size of array: ");

    size = getSize();

*Serial*.println(size);

    int binNumber[size];

    int complNumber[size];

*Serial*.println("Enter binary values in the array: ");

    populateArray(binNumber, size);

*Serial*.println("Binary number entered is: ");

    displayArray(binNumber, size);

*Serial*.print("Decimal value of this binary number is: ");

    float decValue = getDecimalValue(binNumber, size);

*Serial*.println(decValue);

*Serial*.print("Complement of given binary number is: ");

    getComplement(binNumber, complNumber, size);

    displayArray(complNumber, size);

*Serial*.print("Enter a bit position (Position of LS Bit being 0): ");

    int pos = getBitPos();

*Serial*.println(pos);

*Serial*.print("The binary value of digit at position ");

*Serial*.print(pos);

*Serial*.print(" is ");

    int binVal = readBit(binNumber, size, pos);

*Serial*.println(binVal);

}

void loop(){}

int getNumber()

int getSize()

{

    while (*Serial*.available() == 0){}

    String input = *Serial*.readStringUntil('\n');

    int size = input.toInt();

    return size;

}

void populateArray(int arr[], int size)

{

    int i = 0;

    while (i < size)

    {

        while (*Serial*.available() == 0)

        {

        }

        int num = *Serial*.read();

        if (num == '0' || num == '1')

        {

            // convert from ASCII to binary (0 or 1)

            arr[i] = num - '0';

            i++;

        }

    }

}

void displayArray(int arr[], int size)

{

    for (int i = 0; i < size; i++)

    {

*Serial*.print(arr[i]);

*Serial*.print(" ");

    }

*Serial*.println();

float getDecimalValue(int arr[], int size)

{

    float decimalValue = 0;

    for (int i = 0; i < size; i++)

    {

       // mult each binary digit by the corresponding power of 2

        decimalValue += arr[i] \* pow(2, size - 1 - i);

    }

    return decimalValue;

}

void getComplement(int arr[], int complNumber[], int size)

{

    for (int i = 0; i < size; i++)

    {

        if (arr[i] == 0)

        {

            complNumber[i] = 1;

        }

        else complNumber[i] = 0;

    }

}

int getBitPos()

{

    while (*Serial*.available() == 0){}

    String input = *Serial*.readStringUntil('\n');

    int size = input.toInt();

    return size;

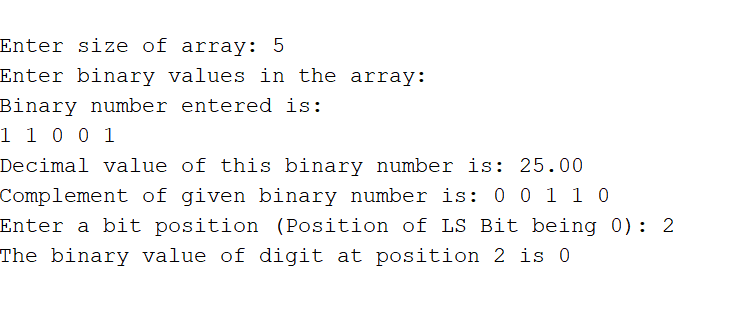
}

int readBit(int arr[], int size, int p)

{

    return arr[size - 1 - p];

**Output**

****

**Comment**: The output demonstrates that the code successfully converts a binary input to decimal, calculates the complement, and retrieves the specified binary digit based on user input.