

# Week 2 – Logic

Student number: 576255

## Assignment 2.1: Parking lot

Which gates do you need?

- A 3-input AND gate

Complete this table

Parking lot 1	Parking lot 2	Parking lot 3	Result (full)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

## Assignment 2.2: Android or iPhone

Which gates do you need?

- An OR gate

Complete this table

Android phone	iPhone	Result (Phone in possession)
0	0	0
0	1	1
1	0	1
1	1	1

### Assignment 2.3: Four NAND gates

Complete this table

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

How can the design be simplified?

The truth table shows that **Q = 1** only when **A** and **B** are **different**.

This is exactly the behaviour of an **XOR gate**:

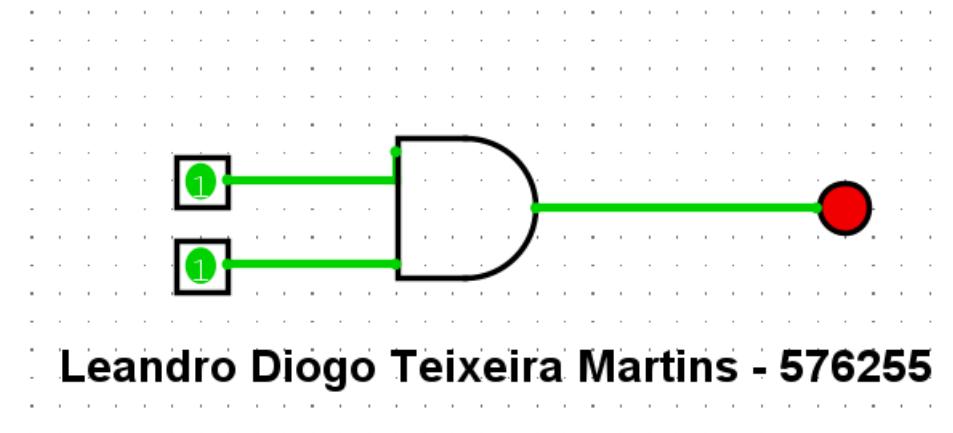
- $Q = A'B + AB'$
- $Q = A \text{ XOR } B$

The current circuit uses **four NAND gates** to build an **XOR function**.

It can be simplified by replacing **all four NAND gates** with **one XOR gate**.

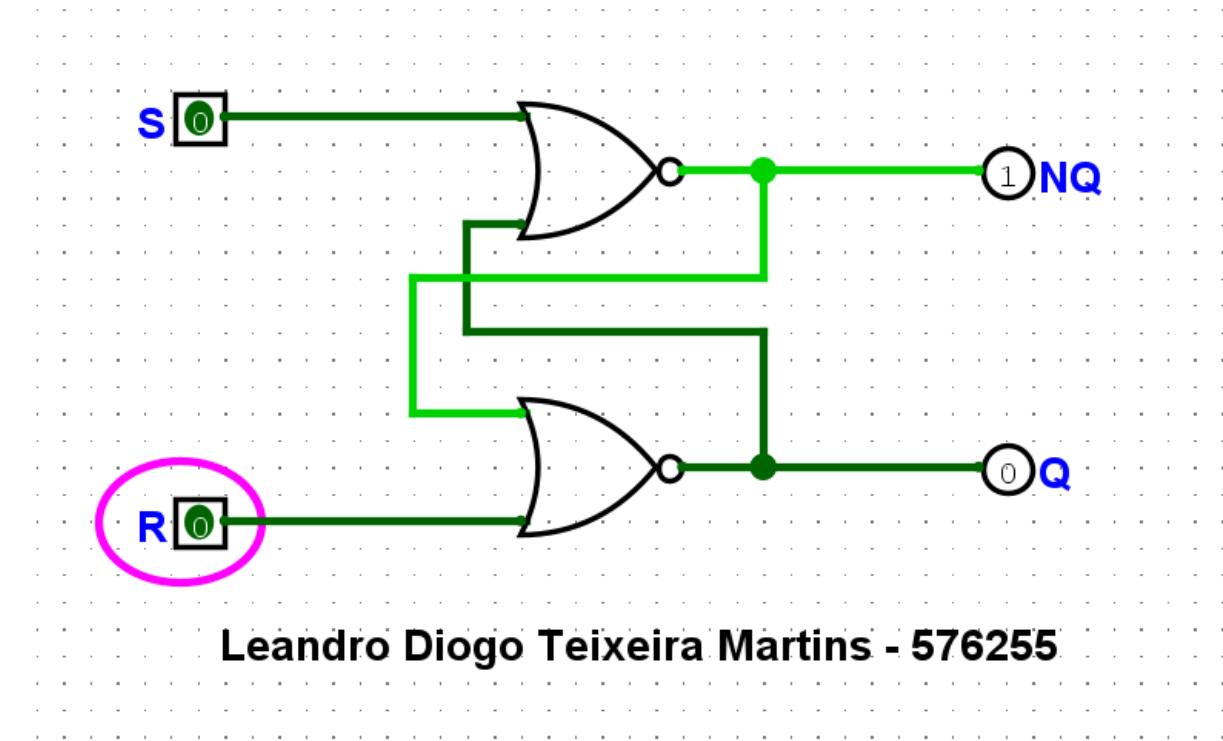
### Assignment 2.4: Getting to know Logisim evolution

Screenshot of the design with your name and student number in it:



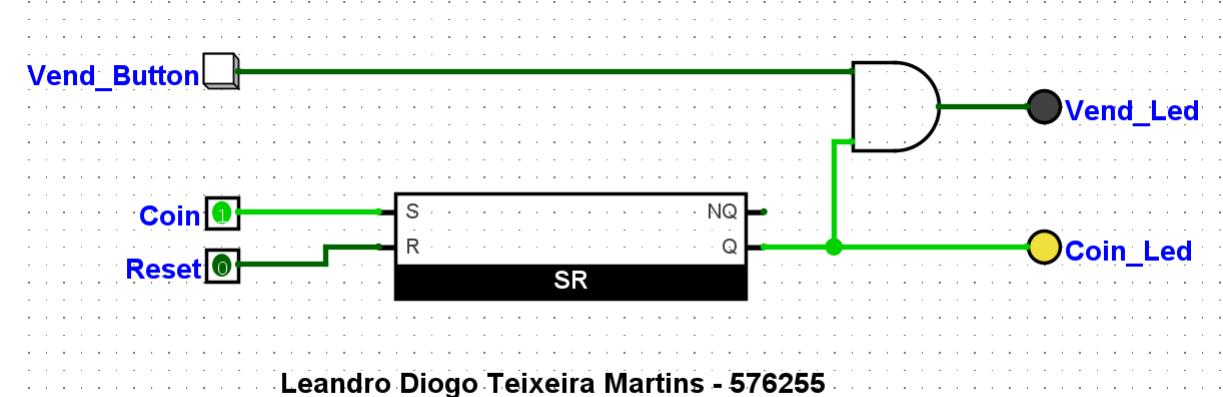
### Assignment 2.5: SR Latch

Screenshot SR Latch in Logisim with your name and student number:



### Assignment 2.6: Vending Machine

Screenshot Vending Machine in Logisim with your name and student number:



### Assignment 2.7: Bitwise operators

Complete the java source code for bitwise operators. Put the source code here.

#### #1 even or odd

```
● ● ●  
public class Main {  
    public static void main(String[] args) {  
        int number = 5;  
  
        if ((number & 1) == 1)  
            System.out.println("number is odd");  
        else  
            System.out.println("number is even");  
    }  
}
```

#### #2 Power of 2

```
● ● ●  
public class Main {  
    public static void main(String[] args) {  
        int number = 4;  
  
        if (number > 0 && (number & (number - 1)) == 0)  
            System.out.println("number is a power of 2");  
        else  
            System.out.println("number isn't a power of 2");  
    }  
}
```

### #3 Check permissions

```
● ● ●

public class Main {
    public static void main(String[] args) {
        final int READ = 4;
        final int WRITE = 2;
        final int EXECUTE = 1;

        int userPermissions = 9;

        if ((userPermissions & READ) != 0)
            System.out.println("User has read permissions");
        else
            System.out.println("User can't read. No permissions.");
    }
}
```

### #4 Assign permissions

```
● ● ●

public class Main {
    public static void main(String[] args) {
        final int READ = 4;
        final int WRITE = 2;
        final int EXECUTE = 1;

        int userPermissions = 0;

        userPermissions = userPermissions | READ;
        userPermissions = userPermissions | EXECUTE;

        System.out.println("User permissions: " + userPermissions);
    }
}
```

## #5 Update permissions

```
public class Main {  
    public static void main(String[] args) {  
        final int READ = 4;  
        final int WRITE = 2;  
        final int EXECUTE = 1;  
  
        int userPermissions = 6;  
  
        userPermissions = userPermissions ^ WRITE;  
  
        System.out.println("User permissions: " + userPermissions);  
    }  
}
```

## #6 Two's complement

```
public class Main {  
    public static void main(String[] args) {  
        int number = 5;  
  
        number = ~number + 1;  
  
        System.out.println("Number: " +  
    number);  
    }
```

### Assignment 2.8: Java Application Bit Calculations

Create a java program that accepts user input and presents a menu with options.

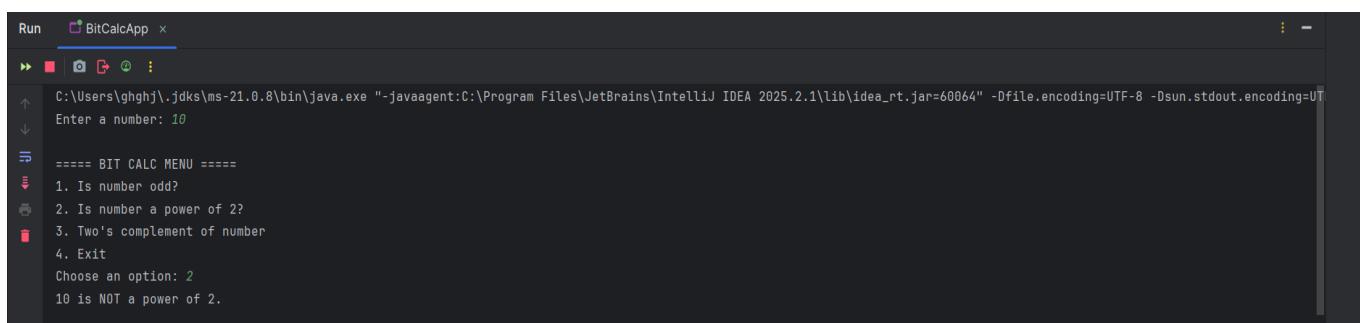
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?

Implement the methods by using the bitwise operators you have just learned.

Organize your source code in a readable manner with the use of control flow and methods.

Keep this application because you need to expand it in week 6 for calculating network segments.

Paste source code here, with a screenshot of a working application.



```
C:\Users\ghghj\.jdks\ms-21.0.8\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=60064" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
Enter a number: 10
===== BIT CALC MENU =====
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number
4. Exit
Choose an option: 2
10 is NOT a power of 2.
```

```
● ● ●

import java.util.Scanner;

public class BitCalcApp {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        int choice;
        do {
            System.out.println("\n===== BIT CALC MENU =====");
            System.out.println("1. Is number odd?");
            System.out.println("2. Is number a power of 2?");
            System.out.println("3. Two's complement of number");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    if (isOdd(number))
                        System.out.println(number + " is odd.");
                    else
                        System.out.println(number + " is even.");
                    break;

                case 2:
                    if (isPowerOfTwo(number))
                        System.out.println(number + " is a power of 2.");
                    else
                        System.out.println(number + " is NOT a power of 2.");
                    break;

                case 3:
                    int twosComp = twosComplement(number);
                    System.out.println("Two's complement of " + number + " is: " +
twosComp);
                    break;

                case 4:
                    System.out.println("Exiting application...");
                    break;

                default:
                    System.out.println("Invalid choice. Try again.");
            }
        } while (choice != 4);
    }

    // methods

    public static boolean isOdd(int n) {
        return (n & 1) == 1;
    }

    public static boolean isPowerOfTwo(int n) {
        return n > 0 && (n & (n - 1)) == 0;
    }

    public static int twosComplement(int n) {
        return ~n + 1;
    }
}
```