Template Week 2 – Logic

Student number: 569091

Assignment 2.1: Parking lot

Which gates do you need?

To decide when the parking lot is full, we need an AND gate. The AND gate outputs 1 (true) only when all inputs are 1.

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/iPhone

Which gates do you need?

A XOR gate, because:

XOR outputs 1 (true) when exactly one input is 1(true).

XOR outputs 0 (false) when:

1. Both inputs are 0 (neither phone is chosen).
2. Both inputs are 1 (both phones are chosen, which is invalid in this scenario).

Complete this table

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

- **Row 1 (0, 0 -> 0):** Neither phone is chosen, so the result is 0 (no phone in possession).
- **Row 2 (0, 1 -> 1):** Only the iPhone is chosen, so the result is 1 (phone in possession).
- **Row 3 (1, 0 -> 1):** Only the Android phone is chosen, so the result is 1 (phone in possession).
- **Row 4 (1, 1 -> 0):** Both phones are chosen, which is invalid, so the result is 0 (no valid choice).

Assignment 2.3: Four NAND gates

Complete this table

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

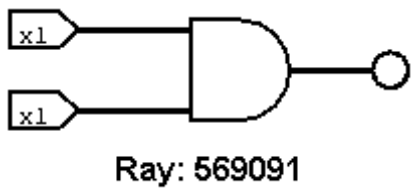
How can the design be simplified?

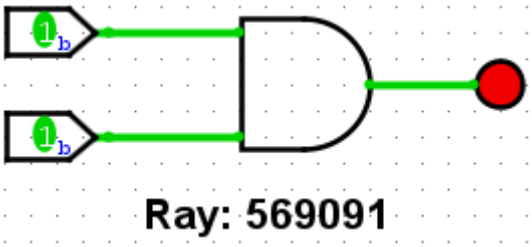
Instead of using 4 NAND gates to achieve the XOR function:

Direct XOR Gate: Use a single XOR gate to replace all 4 NAND gates.

Assignment 2.4: Getting to know Logisim evolution

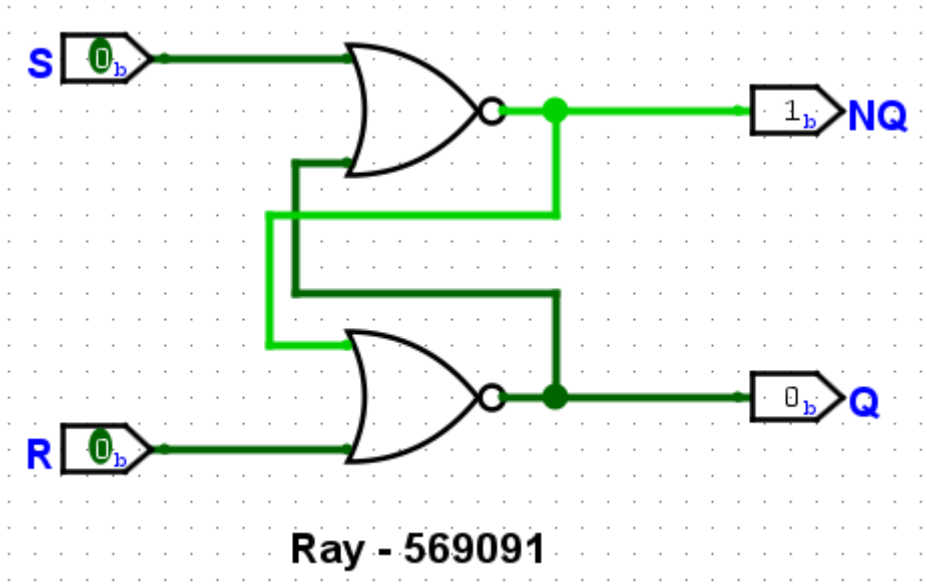
Screenshot of the design with your name and student number in it:





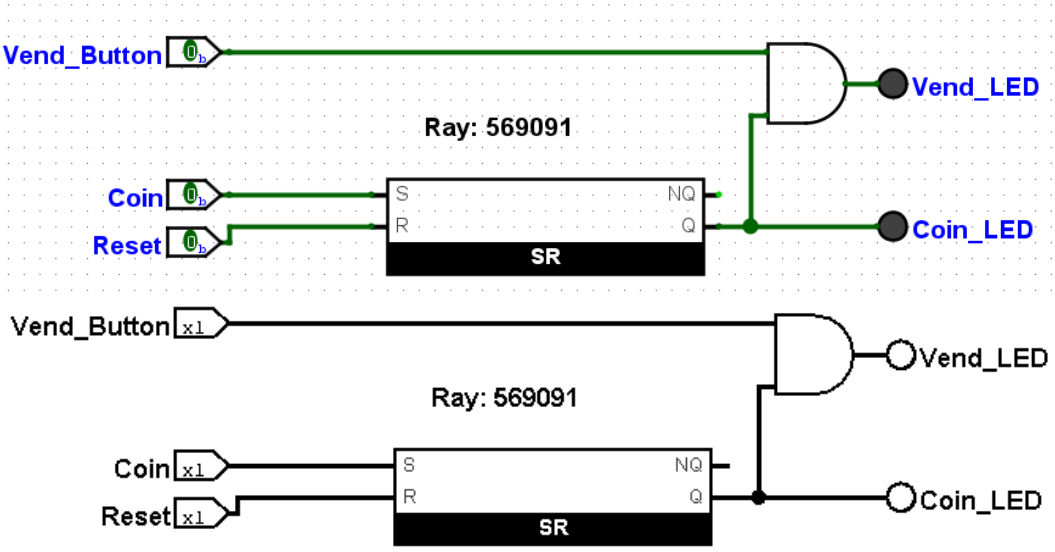
Assignment 2.5: SR Latch

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:



Bonus point assignment – week 2

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

1. Is number odd?

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

2. Is number a power of 2?

```
public class Main {
    public static void main(String[] args) {
        int number = 4;

        // check if the number is a power of 2 using bitwise & operator
        // compare number with number -1 (always get false)
        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. Two’s complement of number?

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

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

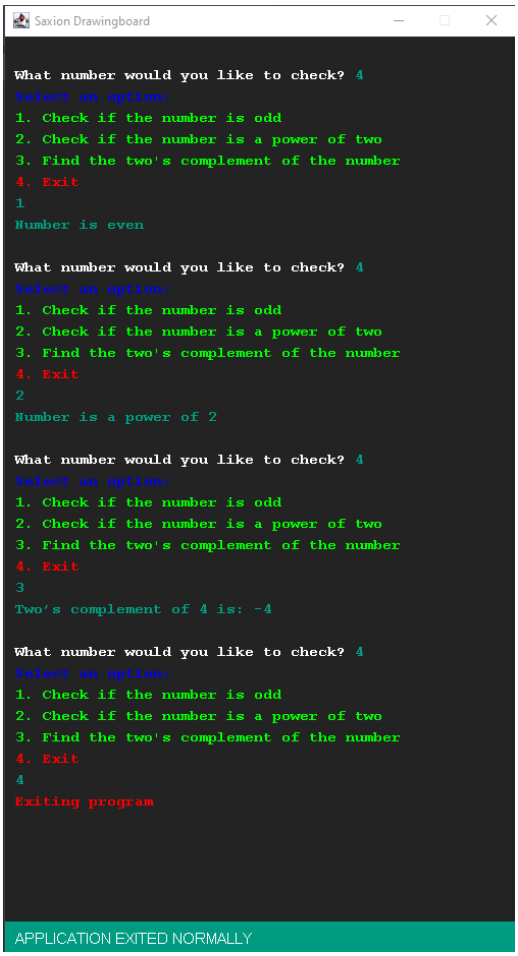
Organize your source code in a readable manner with the use of control flow and methods.

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

```
import nl.saxion.app.SaxionApp;

import java.awt.*;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 500, 900);
    }

    public void run() {
        boolean running = true;

        while (running) {
            SaxionApp.printLine();
            SaxionApp.print("What number would you like to check? ");
            int number = SaxionApp.readInt();

            SaxionApp.printLine("Select an option:", Color.blue);
            SaxionApp.printLine("1. Check if the number is odd", Color.green);
            SaxionApp.printLine("2. Check if the number is a power of two", Color.green);
            SaxionApp.printLine("3. Find the two's complement of the number", Color.green);
            SaxionApp.printLine("4. Exit", Color.red);

            int keyInput = SaxionApp.readInt();

            if (keyInput == 1) {
                checkOddNumber(number);
            } else if (keyInput == 2) {
                checkPowerOfTwo(number);
            } else if (keyInput == 3) {
                checkTwosComplement(number);
            } else if (keyInput == 4) {
                SaxionApp.printLine("Exiting program", Color.red);
                running = false;
            } else {
                SaxionApp.printLine("Invalid option. Please try again.", Color.red);
            }
        }
    }

    // Method to check if a number is odd
    public static void checkOddNumber(int number) {
        if ((number & 1) == 1) { // Check the LSB
            SaxionApp.printLine("Number is odd",SaxionApp.SAXION_GREEN);
        } else {
            SaxionApp.printLine("Number is even",SaxionApp.SAXION_GREEN);
        }
```

```
    }

    // Method to check if a number is a power of 2
    public static void checkPowerOfTwo(int number) {
        if (number > 0 && (number & (number - 1)) == 0) {
            SaxionApp.printLine("Number is a power of 2", SaxionApp.SAXION_GREEN);
        } else {
            SaxionApp.printLine("Number isn't a power of 2",SaxionApp.SAXION_PINK);
        }
    }

    // Method to calculate the two's complement of a number
    public static void checkTwosComplement(int number) {
        int twosComplement = ~number + 1; // Two's complement operation
        SaxionApp.printLine("Two's complement of " + number + " is: " + twosComplement,
SaxionApp.SAXION_GREEN);
    }
}
```

Added a simple while loop to demonstrate all methods work with one screenshot.



Ready? Then save this file and export it as a pdf file with the name: **week2.pdf**