

# Template Week 2 – Logic

Student number: 529471

## Assignment 2.1: Parking lot

Which gates do you need?

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/iPhone

Which gates do you need?

OR gate

Complete this table

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

### Assignment 2.3: Four NAND gates

Complete this table

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

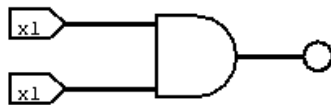
How can the design be simplified?

It can be simplified by using one XOR gate instead of four NAND gates, as the truth table represents a XOR gate's truth table in the end. Q is only true (1) when exactly one of its inputs is true (1).

### Assignment 2.4: Getting to know Logisim evolution

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

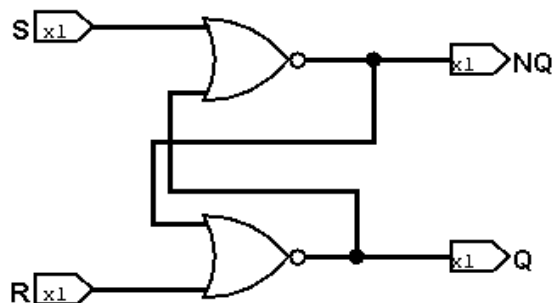
Kayla Gencer  
529471



### Assignment 2.5: SR Latch

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

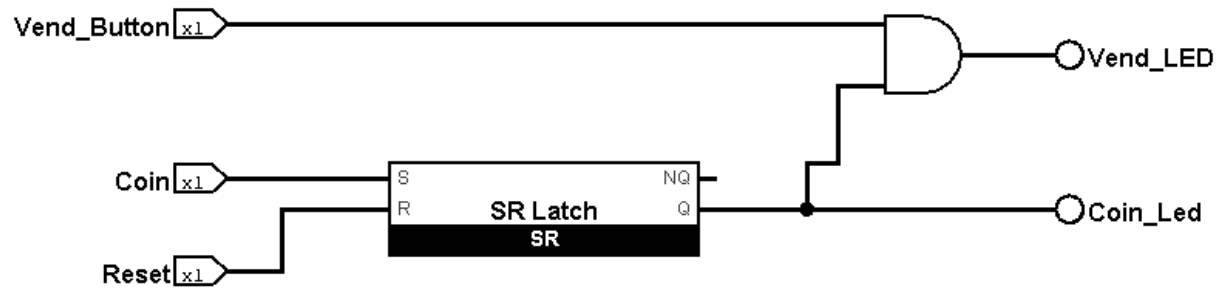
Kayla Gencer  
529471



## Assignment 2.6: Vending Machine

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

Kayla Gencer  
529471



## Bonus point assignment – week 2

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

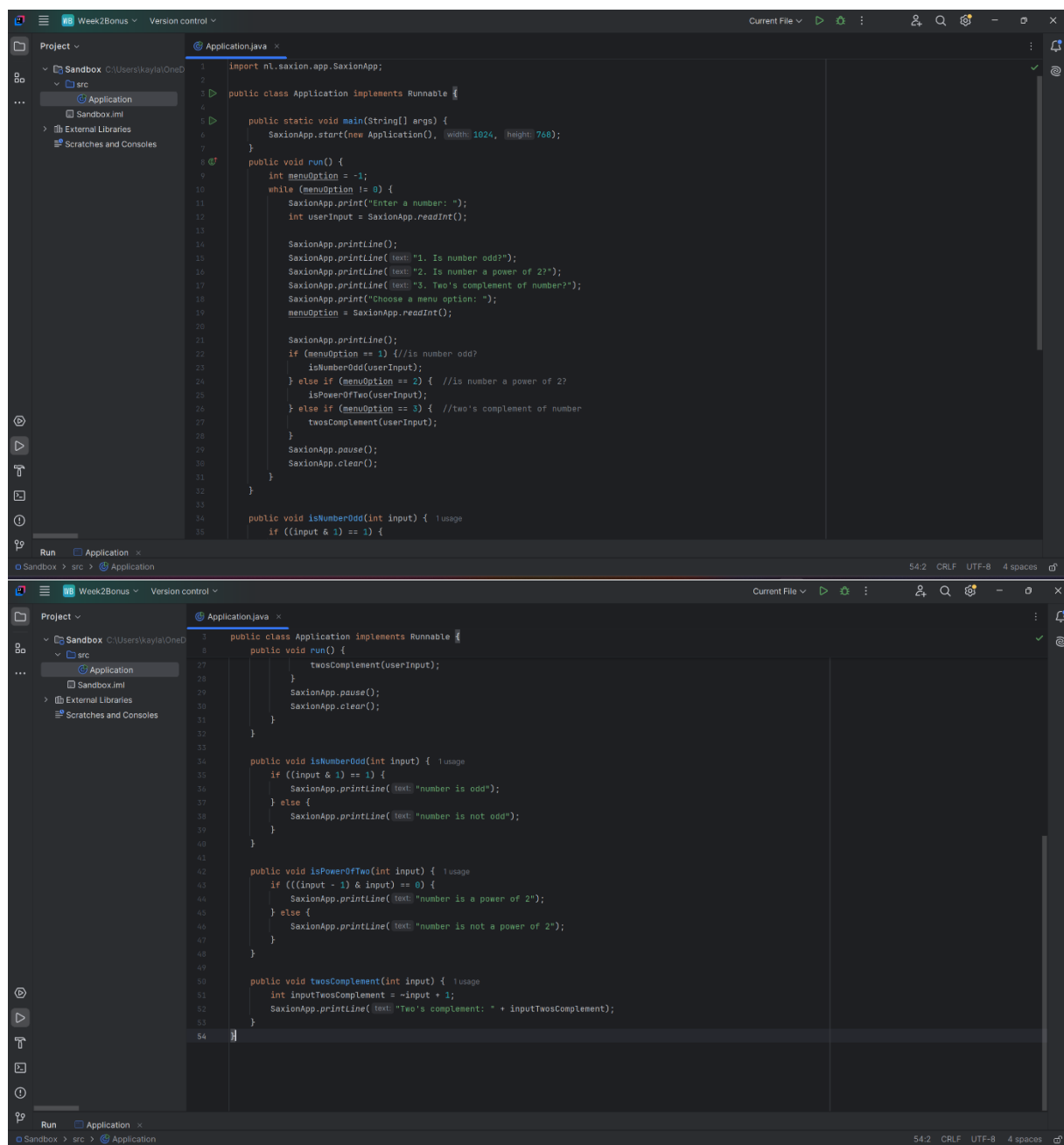
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?

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.

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

### Source Code:



The image displays two screenshots of an IDE (IntelliJ IDEA) showing the source code for a Java application named 'Application.java'. The code implements a menu-driven program that accepts user input and presents a menu with options.

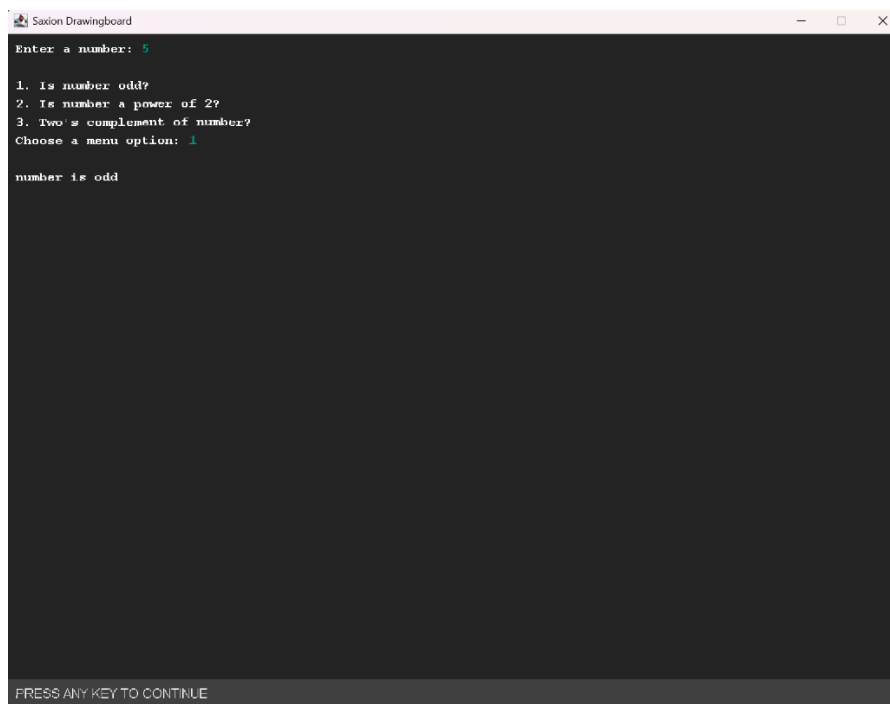
**Top Screenshot:** Shows the initial code structure. The class 'Application' implements 'Runnable'. The 'main' method starts the application with a width of 1024 and height of 768. The 'run' method contains a loop that prompts the user to enter a number and displays a menu with three options: '1. Is number odd?', '2. Is number a power of 2?', and '3. Two's complement of number?'. The user is prompted to choose a menu option, and the code calls 'isNumberOdd', 'isPowerOfTwo', or 'twoComplement' based on the selected option. The code also includes a 'pause' and 'clear' call to SaxonApp.

```
1 import nl.saxon.app.SaxonApp;
2
3 public class Application implements Runnable {
4
5     public static void main(String[] args) {
6         SaxonApp.start(new Application(), width: 1024, height: 768);
7     }
8
9     public void run() {
10         int menuOption = -1;
11         while (menuOption != 0) {
12             SaxonApp.print("Enter a number: ");
13             int userInput = SaxonApp.readInt();
14
15             SaxonApp.println();
16             SaxonApp.println(text: "1. Is number odd?");
17             SaxonApp.println(text: "2. Is number a power of 2?");
18             SaxonApp.println(text: "3. Two's complement of number?");
19             SaxonApp.print("Choose a menu option: ");
20             menuOption = SaxonApp.readInt();
21
22             SaxonApp.println();
23             if (menuOption == 1) { //is number odd?
24                 isNumberOdd(userInput);
25             } else if (menuOption == 2) { //is number a power of 2?
26                 isPowerOfTwo(userInput);
27             } else if (menuOption == 3) { //two's complement of number
28                 twoComplement(userInput);
29             }
30             SaxonApp.pause();
31             SaxonApp.clear();
32         }
33     }
34
35     public void isNumberOdd(int input) { //usage
36         if ((input & 1) == 1) {
```

**Bottom Screenshot:** Shows the implementation of the methods 'isNumberOdd', 'isPowerOfTwo', and 'twoComplement'. The 'isNumberOdd' method checks if the input is odd using a bitwise AND operation. The 'isPowerOfTwo' method checks if the input is a power of 2 using a bitwise AND operation. The 'twoComplement' method calculates the two's complement of the input using a bitwise NOT operation and an increment.

```
37         SaxonApp.println(text: "number is odd");
38     } else {
39         SaxonApp.println(text: "number is not odd");
40     }
41 }
42
43 public void isPowerOfTwo(int input) { //usage
44     if (((input - 1) & input) == 0) {
45         SaxonApp.println(text: "number is a power of 2");
46     } else {
47         SaxonApp.println(text: "number is not a power of 2");
48     }
49 }
50
51 public void twoComplement(int input) { //usage
52     int inputTwoComplement = ~input + 1;
53     SaxonApp.println(text: "Two's complement: " + inputTwoComplement);
54 }
```

## Working Application

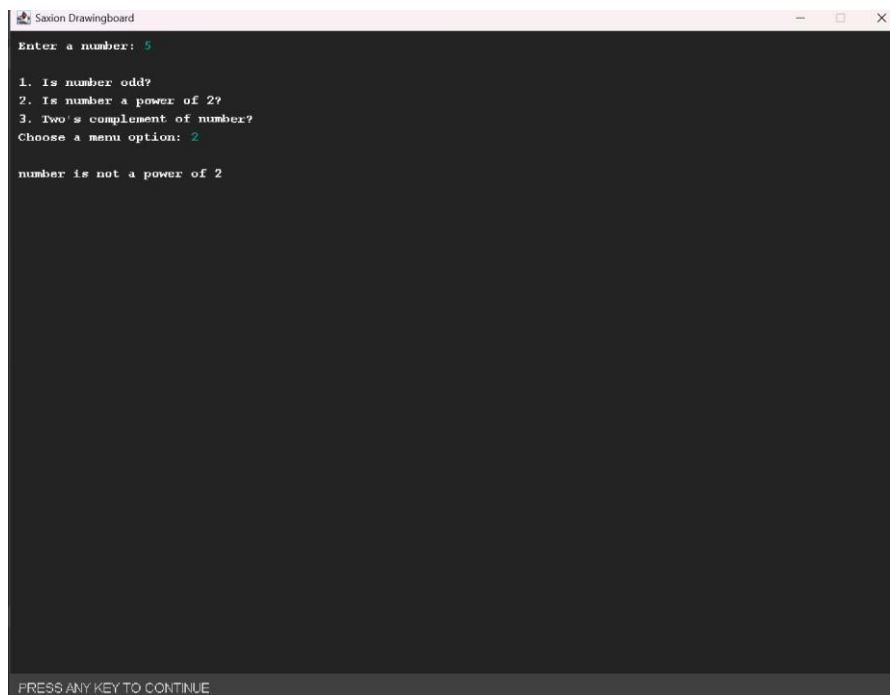


```
Enter a number: 5

1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
Choose a menu option: 1

number is odd

PRESS ANY KEY TO CONTINUE
```



```
Enter a number: 5

1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
Choose a menu option: 2

number is not a power of 2

PRESS ANY KEY TO CONTINUE
```

```
Enter a number: 5

1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
Choose a menu option: 3

Two's complement: -5

PRESS ANY KEY TO CONTINUE
```

Ready? Then save this file and export it as a pdf file with the name: [week2.pdf](#)