

Template Week 2 – Logic

Student number: 587910

Assignment 2.1: Parking lot

Which gates do you need?

Je hebt een and gate nodig, want de rode lamp met full gaat alleen aan als alle parkeerplekken bezet zijn. Er komt dus alleen een 1 (full) als alle plekken een 1 hebben staan (bezett zijn).

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 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

Assignment 2.2: Android or iPhone

Which gates do you need?

Je hebt een xor gate nodig, want de output is alleen 1, als precies 1 van de inputs waar is.

Complete this table

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

Assignment 2.3: Four NAND gates

Complete this table

| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

How can the design be simplified?

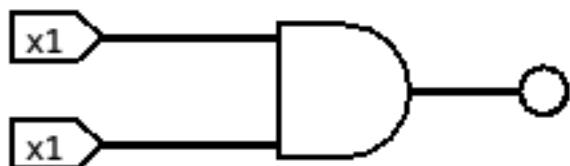
De output kan alleen 1 zijn als de inputs van A en B verschillend zijn. Dit is dus eigenlijk een xor poort, omdat maar 1 input waar mag zijn.

Assignment 2.4: Getting to know Logisim evolution

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

Micha Pool

587910

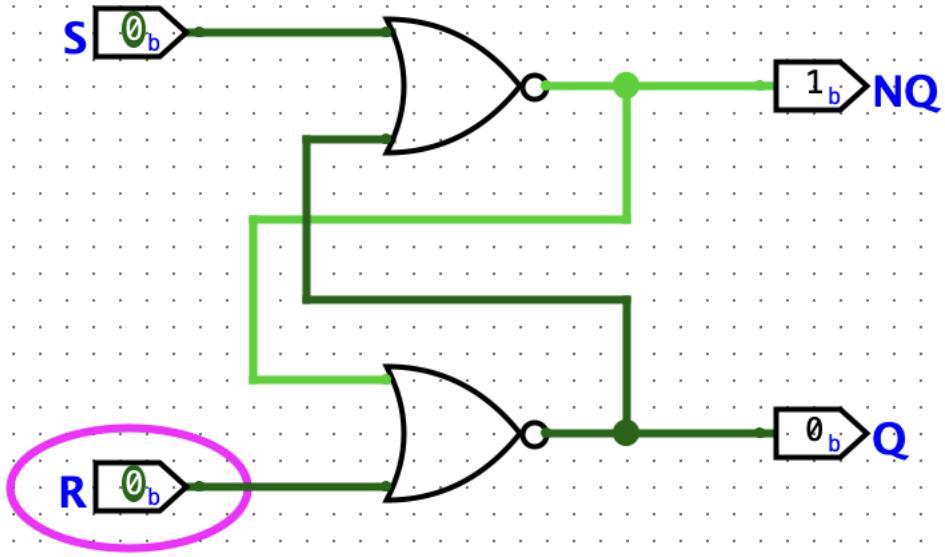


Assignment 2.5: SR Latch

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

Micha Pool

587910

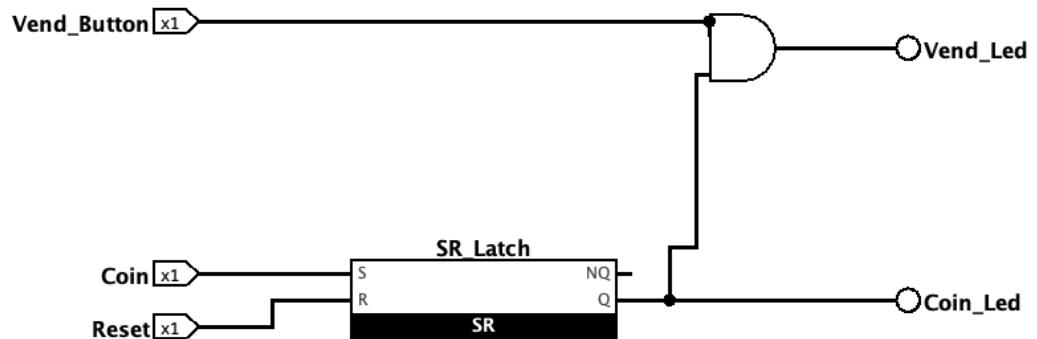


Assignment 2.6: Vending Machine

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

Micha Pool

587910



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 = 6;  
        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 = 0;  
        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 = READ | EXECUTE;
```

```
        System.out.println("User permissions (Octal/Decimal): " + 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; // Current: READ (4) and WRITE (2)
```

```
        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);  
        number = ~number + 1;  
        System.out.println("Number back again: " + 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.

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

```

public class Main implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Main());
    }

    public void run() {

        SaxionApp.print("Enter an integer to analyze: ");
        int number = SaxionApp.readInt();

        SaxionApp.printLine("\n--- Bitwise Menu ---");
        SaxionApp.printLine("1. Is number odd?");
        SaxionApp.printLine("2. Is number a power of 2?");
        SaxionApp.printLine("3. Two's complement of number?");
        SaxionApp.print("\nSelect an option (1-3): ");

        int choice = SaxionApp.readInt();
        SaxionApp.printLine("-----");

        if (choice == 1) {
            // Bitwise AND (&) to check the LSB (Least Significant Bit)
            if ((number & 1) != 0) {
                SaxionApp.printLine(number + " is ODD.");
            } else {
                SaxionApp.printLine(number + " is EVEN.");
            }
        } else if (choice == 2) {
    }
}

```

```

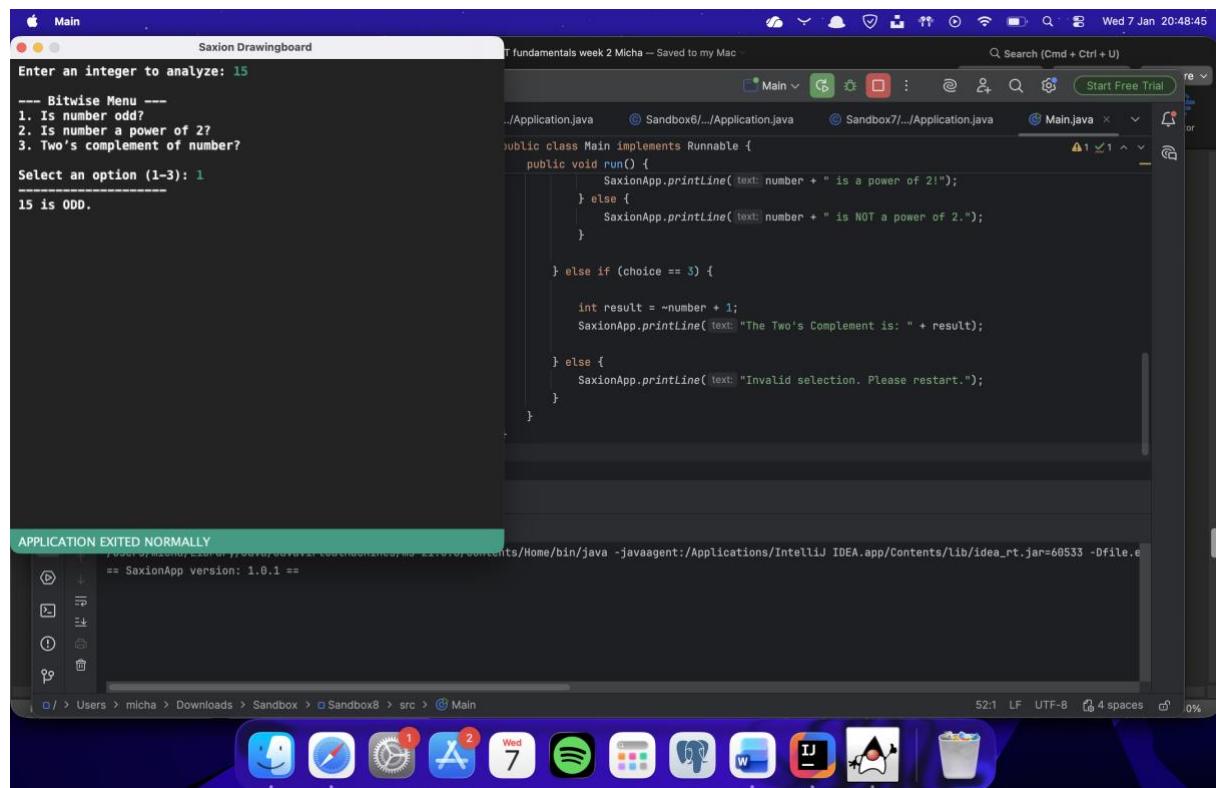
        if (number > 0 && (number & (number - 1)) == 0) {
            SaxonApp.printLine(number + " is a power of 2!");
        } else {
            SaxonApp.printLine(number + " is NOT a power of 2.");
        }

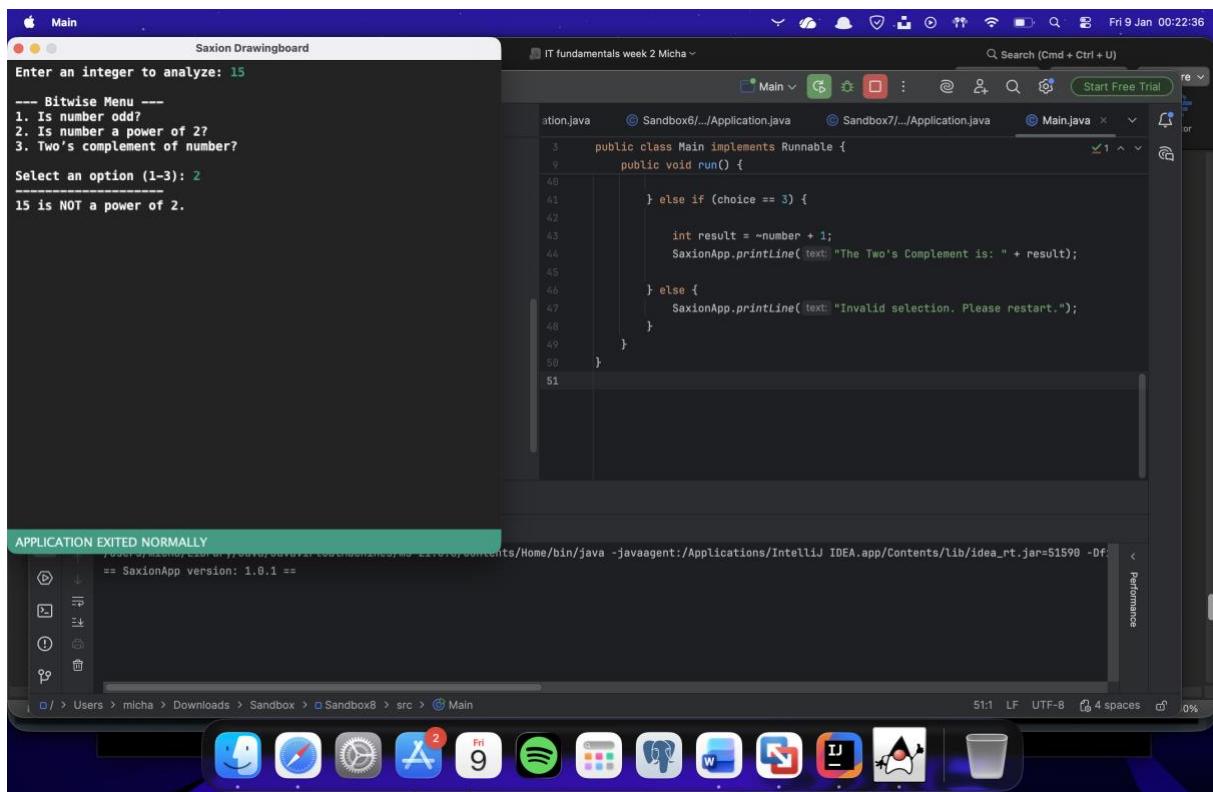
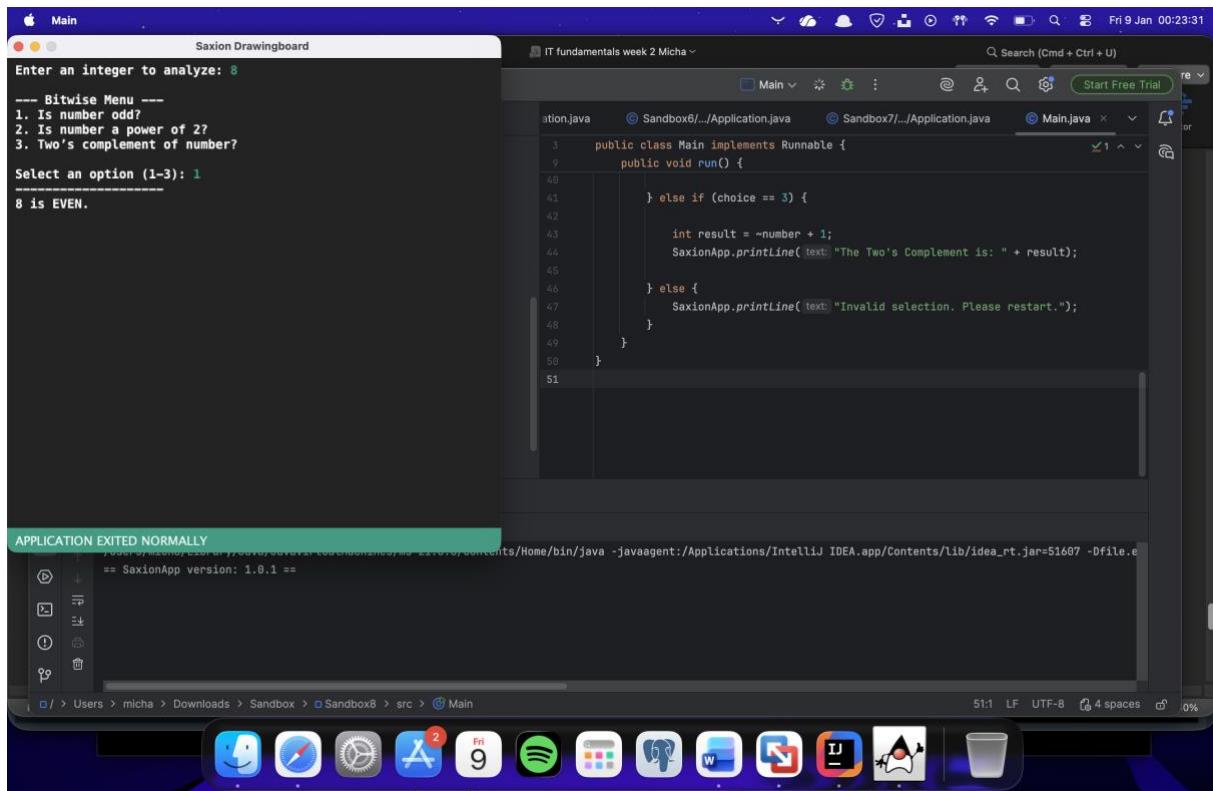
    } else if (choice == 3) {

        int result = ~number + 1;
        SaxonApp.printLine("The Two's Complement is: " + result);

    } else {
        SaxonApp.printLine("Invalid selection. Please restart.");
    }
}

```





The screenshot shows a Mac desktop with two windows open. The terminal window at the bottom has the following text:

```
APPLICATION EXITED NORMALLY
== SaxonApp version: 1.0.1 ==
```

The IntelliJ IDEA window above it displays Java code in a file named Main.java:

```
public class Main implements Runnable {
    public void run() {
        if (choice == 1) {
            int result = ~number + 1;
            SaxonApp.printLine( text: "The Two's Complement is: " + result);
        } else if (choice == 2) {
            SaxonApp.printLine( text: "The number is a power of 2." );
        } else {
            SaxonApp.printLine( text: "Invalid selection. Please restart." );
        }
    }
}
```

The status bar at the bottom of the screen shows the date and time as Fri Jan 00:23:55.

The screenshot shows a Mac desktop with two windows open. The terminal window at the bottom has the following text:

```
APPLICATION EXITED NORMALLY
== SaxonApp version: 1.0.1 ==
```

The IntelliJ IDEA window above it displays Java code in a file named Main.java:

```
public class Main implements Runnable {
    public void run() {
        if (choice == 1) {
            int result = ~number + 1;
            SaxonApp.printLine( text: "The Two's Complement is: " + result);
        } else if (choice == 2) {
            SaxonApp.printLine( text: "The number is a power of 2." );
        } else {
            SaxonApp.printLine( text: "Invalid selection. Please restart." );
        }
    }
}
```

The status bar at the bottom of the screen shows the date and time as Fri Jan 00:22:36.

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