Template Week 2 – Logic

Student number:

Assignment 2.1: Parking lot

Which gates do you need?

> AND: alles moet waar zijn voordat de rode LED moet gaan branden.

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?

> XOR; precies 1 moet waar zijn, dus de medewerker moet Android of een iPhone kiezen, je kan niet beide en er moet een keuze gemaakt worden.

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
1	0	1
0	1	1
1	1	0

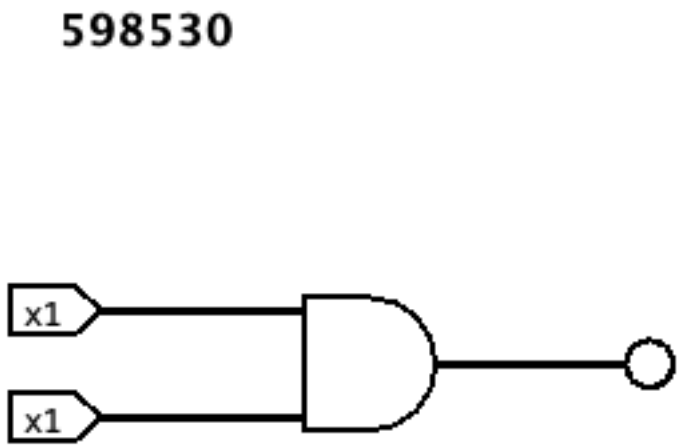
How can the design be simplified?

Het is makkelijker om simpelweg 1 XOR te hebben. Er mag maximaal 1 waar zijn. Hier is de waarheid tabel voor een XOR en is hetzelfde als die van hierboven.

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

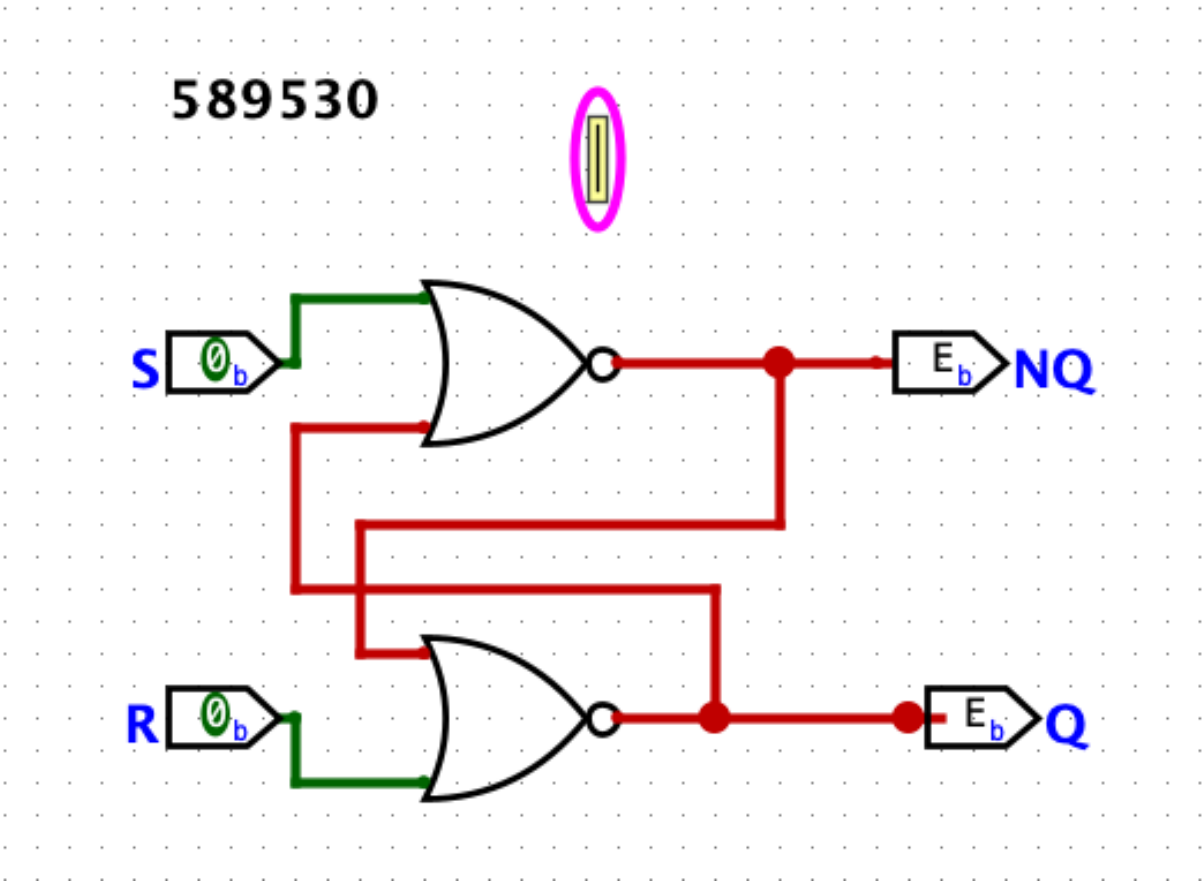
Assignment 2.4: Getting to know Logisim evolution

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



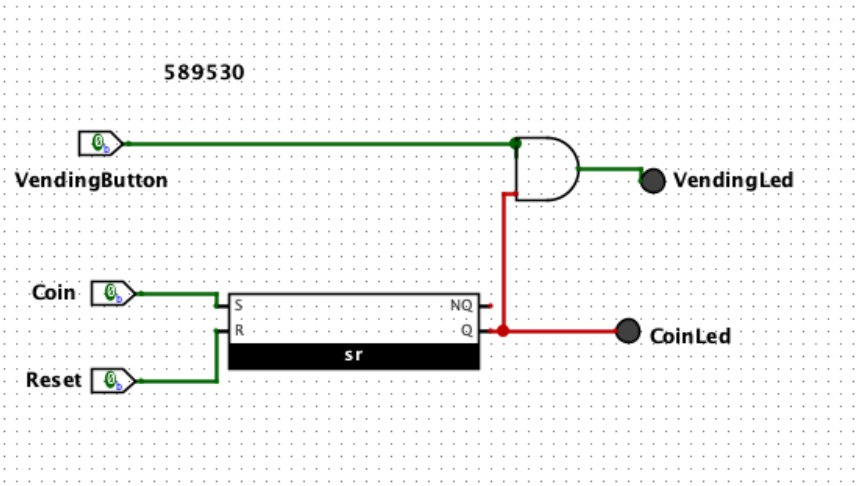
Assignment 2.5: SR Latch

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



Assignment 2.6: Vending Machine

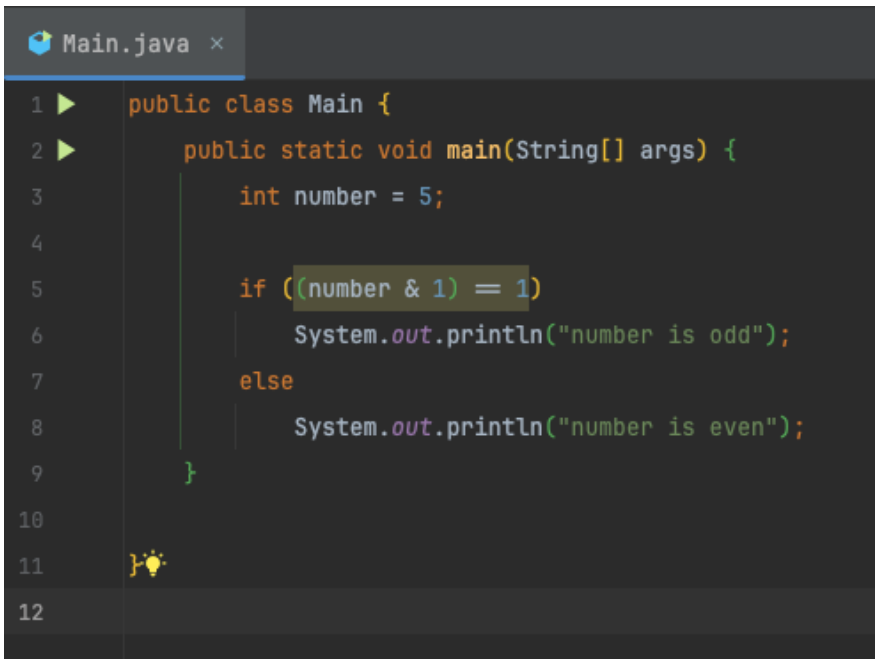
Screenshot Vending Machine in Logisim with your name and student number:



Assignment 2.7: Bitwise operators

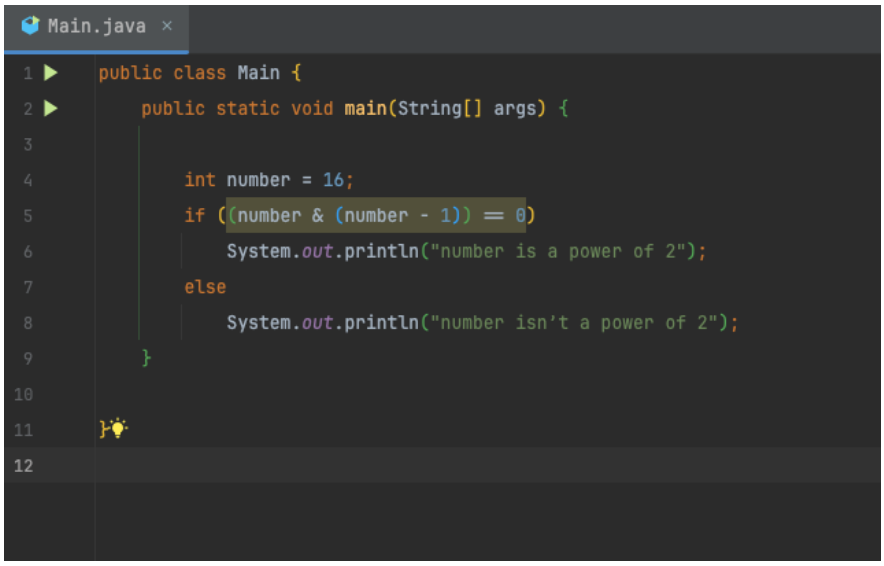
Complete the java source code for bitwise operators. Put the source code here.

#1



Dit werkt omdat een oneven nummer eindigt op een 1, en wanneer je binair 0001 erbij doet krijg je veranderd het laatste getal, bij een even getal niet.

#2



#3

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

    int userPermissions = 7;

    if ((userPermissions & READ) == READ) {

        System.out.println("User has read permissions");
    } else {
        System.out.println("User can't read. No permissions.");
    }
}
```

#4

```java
public static void main(String[] args) {
    final int READ = 4; //  0100
    final int WRITE = 2; // 0010
    final int EXECUTE = 1;//0001

    int userPermissions = 0;
    userPermissions = userPermissions | READ | EXECUTE;

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

}
```

Uitleg:

4 = 0100

1 = 0001

---------- +

5 = 0101

En dat is precies wat een | (OR) doet, 1 moet waar zijn en dan het het antwoord waar, daarom kom je op 5 uit.

#5

```java
public static void main(String[] args) {
    final int READ = 4; //  0100
    final int WRITE = 2; // 0010
    final int EXECUTE = 1;//0001

    int userPermissions = 6;
    userPermissions = userPermissions ^ WRITE;
    System.out.println("User permissions: "+userPermissions);

}
```

Uitleg: bij XOR mag er 1 maar waar zijn.

6 = 0110

2 = 0010

---------- ^

4 = 0100 dus dan heb je alleen READ.

#6

```java
public static void main(String[] args) {
    int number = 6; // 0101
    number = ~number + 1;
    System.out.println("Number: "+number);

}
```

Uitleg: ~ flipt alle bits, dus 0101 (6) wordt 1010 en omdat het 2 compliment is altijd + 1 omdat zonder de + 1 krijg je 1 compliment.

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.

```java
import java.util.Scanner;

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

        System.out.println("Please enter an number");
        int number = scn.nextInt();

        System.out.println("is the number odd?: " + isNumberOdd(number));
        System.out.println("is the power of 2?: " + isNumberPowerOfTwo(number));


    }

    public static boolean isNumberOdd(int number) {  1 usage
        return (number & 1) == 1;
    }

    public static boolean isNumberPowerOfTwo(int number) {  1 usage
        return (number & (number - 1)) == 0;
    }
}
```

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