Template Week 2 – Logic

Student number: 560830

Assignment 2.1: Parking lot

Which gates do you need?

Two AND gates, one checking if lot 1 AND 2 are full, and then one to check whether AND[1 & 2] and lot 3 are full. Hook up the result of the second AND gate to a display to show whether it's full or not

Complete this table

Parking lot 1	Parking lot 2	Parking lot 3	Result (full)
0	0	0	No
0	0	1	No
0	1	0	No
0	1	1	No
1	1	1	Yes
1	1	0	No
1	0	0	No
1	0	1	No

Assignment 2.2: Android or iPhone

Which gates do you need?

XOR

Complete this table

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

Assignment 2.3: Four NAND gates

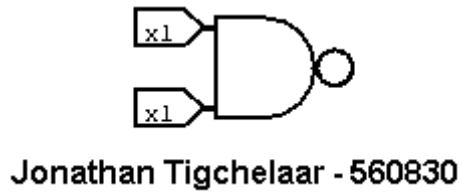
Complete this table

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

How can the design be simplified?

Assignment 2.4: Getting to know Logisim evolution

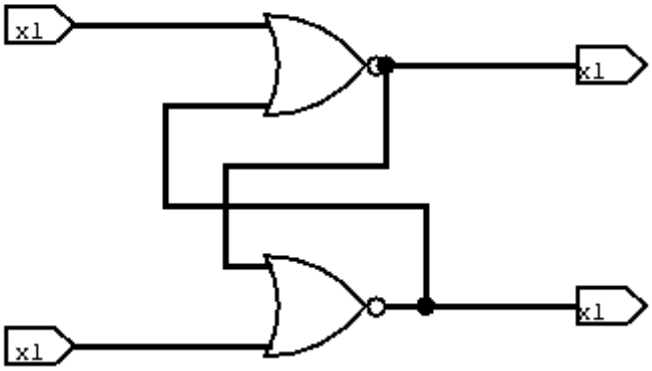
Screenshot of the design with your name and student number in it:



Assignment 2.5: SR Latch

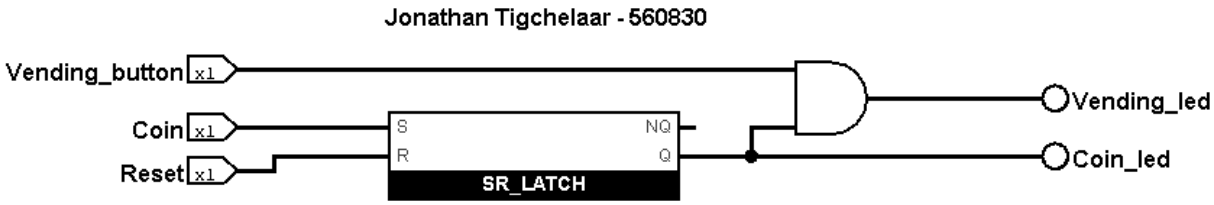
Screenshot SR Latch in Logisim with your name and student number:

Jonathan Tigchelaar - 560830



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.

I've made exercise #1, #2 and #6 into the final script in Assignment 2.8.

#3

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

```
public class Application2 implements Runnable {
   public void run() {
       final int READ = 4;
       final int WRITE = 2;
       final int EXECUTE = 1;

       int userPermissions = 7;

       if((userPermissions & READ) != 0){ SaxionApp.printLine("User has read permissions");}
       else {SaxionApp.printLine("User can't read. No permissions.");}
   }

   public static void main(String[] args) {
       SaxionApp.start(new Application2(), 1000, 300);
   }
}
```

#4

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

public class Application2 implements Runnable {
   public void run() {
       final int READ = 4;
       final int WRITE = 2;
       final int EXECUTE = 1;

       int userPermissions = READ | EXECUTE;

       SaxionApp.printLine("User permissions: " + userPermissions);
   }

   public static void main(String[] args) {
       SaxionApp.start(new Application2(), 1000, 300);
   }
}
```

#5

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


public class Application2 implements Runnable {
   public void run() {
        final int READ = 4;
        final int WRITE = 2;
        final int EXECUTE = 1;


        int userPermissions = 6;
        userPermissions = userPermissions ^ WRITE;


        SaxionApp.printLine("User permissions: " + userPermissions);
   }


   public static void main(String[] args) {
        SaxionApp.start(new Application2(), 1000, 300);
   }
}
```

#7

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


public class Application2 implements Runnable {
   public void run() {
        int numberToConvert = 10;


        SaxionApp.printLine("Decimal integer: "+numberToConvert);


        String binary = Integer.toBinaryString(numberToConvert);
        String octal = Integer.toOctalString(numberToConvert);
        String hexadecimal = Integer.toHexString(numberToConvert);


        SaxionApp.printLine("Binary representation: " + binary);
        SaxionApp.printLine("Octal representation: " + octal);
        SaxionApp.printLine("Hexadecimal representation: " + hexadecimal);
```

```
    }

    public static void main(String[] args) {
        SaxionApp.start(new Application2(), 1000, 300);
    }
}
```

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.

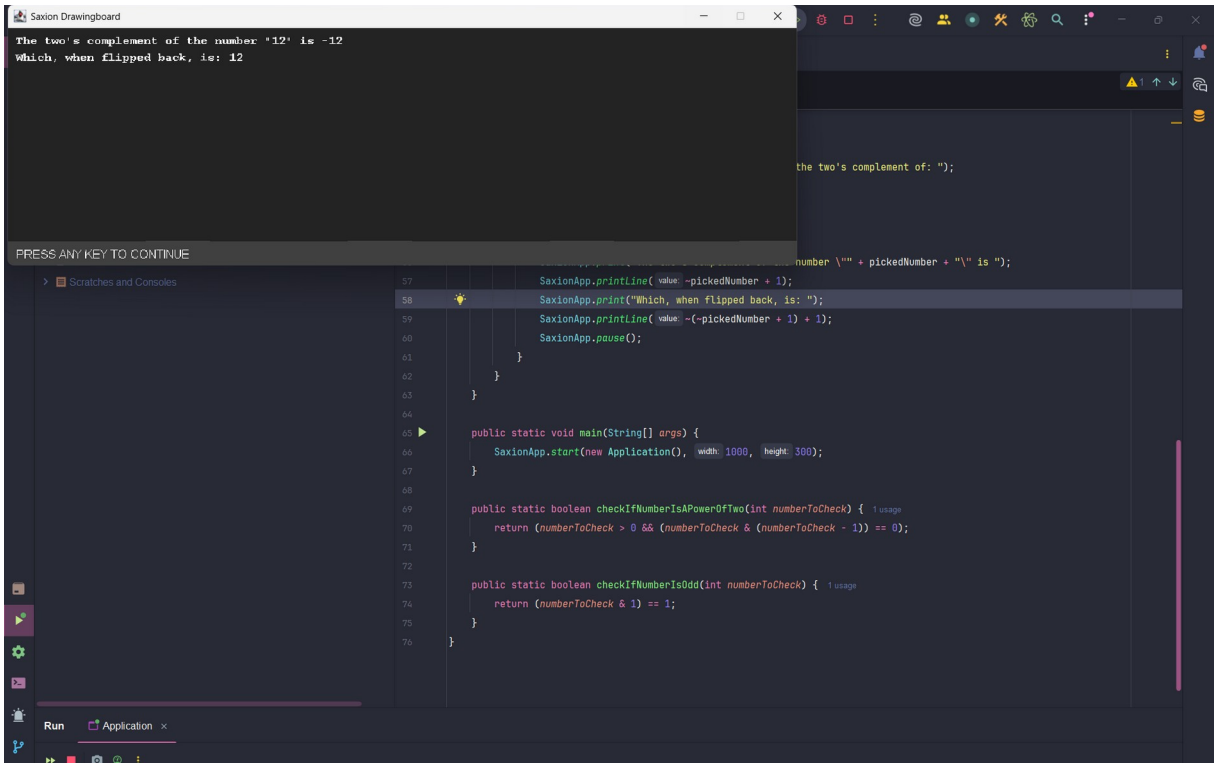
Keep this application because you need to expand it in week 6 for calculating network segments.

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

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

public class Application implements Runnable {
    public void run() {
        while(true) {
            SaxionApp.clear();
            SaxionApp.printLine("This program has three modes:");
            SaxionApp.printLine("1. Check if a number is odd");
            SaxionApp.printLine("2. Check if a number is a power of two");
            SaxionApp.printLine("3. Check the two's complement of a number");
            SaxionApp.print("\nWhich option do you pick: ");

            int pickedMode = SaxionApp.readInt();

            SaxionApp.clear();

            if (pickedMode == 1) {
                SaxionApp.print("Input a number to check: ");
                int pickedNumber = SaxionApp.readInt();
```

```
                SaxionApp.clear();

                SaxionApp.print("The number \"" + pickedNumber + "\" is ");

                if (checkIfNumberIsOdd(pickedNumber)) {
                    SaxionApp.printLine("odd.");
                } else {
                    SaxionApp.printLine("even.");
                }

                SaxionApp.pause();
            }

            else if (pickedMode == 2) {
                SaxionApp.print("Input a number to check: ");

                int pickedNumber = SaxionApp.readInt();

                SaxionApp.clear();
                SaxionApp.print("The number \"" + pickedNumber + "\" is ");

                if (checkIfNumberIsAPowerOfTwo(pickedNumber)) {
                    SaxionApp.printLine("a power of two.");
                } else {
                    SaxionApp.printLine("not a power of two.");
                }
                SaxionApp.pause();
            }

            else if (pickedMode == 3) {
                SaxionApp.print("Input a number to calculate the two's complement
of: ");

                int pickedNumber = SaxionApp.readInt();

                SaxionApp.clear();
```

```
                SaxionApp.print("The two's complement of the number \"" +
pickedNumber + "\" is ");

                SaxionApp.printLine(~pickedNumber + 1);

                SaxionApp.print("Which, when flipped back, is: ");

                SaxionApp.printLine(~(~pickedNumber + 1) + 1);

                SaxionApp.pause();

            }

        }

    }


    public static void main(String[] args) {

        SaxionApp.start(new Application(), 1000, 300);

    }


    public static boolean checkIfNumberIsAPowerOfTwo(int numberToCheck) {

        return (numberToCheck > 0 && (numberToCheck & (numberToCheck - 1)) == 0);

    }


    public static boolean checkIfNumberIsOdd(int numberToCheck) {

        return (numberToCheck & 1) == 1;

    }
}
```


Saxion Drawingboard
The two's complement of the number "12" is -12
Which, when flipped back, is: 12
PRESS ANY KEY TO CONTINUE
Scratches and Consoles
the two's complement of: ");
number \"" + pickedNumber + "\" is ");
SaxionApp.printLine(value: ~pickedNumber + 1);
SaxionApp.print("Which, when flipped back, is: ");
SaxionApp.printLine(value: ~(~pickedNumber + 1) + 1);
SaxionApp.pause();
}
}
}
public static void main(String[] args) {
SaxionApp.start(new Application(), width: 1000, height: 300);
}
public static boolean checkIfNumberIsAPowerOfTwo(int numberToCheck) { 1 usage
return (numberToCheck > 0 && (numberToCheck & (numberToCheck - 1)) == 0);
}
public static boolean checkIfNumberIsOdd(int numberToCheck) { 1 usage
return (numberToCheck & 1) == 1;
}
}
Run
Application