

## **Week 2 – Logic**

Student number: 589932

### **Assignment 2.1: Parking lot**

Which gates do you need?

To detect whether the parking lot is full, we need a 3-input AND gate. The FULL sign should turn on only when all three parking spaces are occupied (all sensors = 1).

If only 2-input AND gates are available, we can combine two AND gates.

Complete this table

<b>Parking lot 1</b>	<b>Parking lot 2</b>	<b>Parking lot 3</b>	<b>Result (full)</b>
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 or iPhone**

Which gates do you need?

To allow only one choice between an Android phone or an iPhone, we need an XOR gate.

An XOR gate outputs 1 only when exactly one input is 1, which matches the rule:

We cannot choose both phones and I cannot choose none.

Complete this table

<b>Android phone</b>	<b>iPhone</b>	<b>Result (Phone in possession)</b>

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	1
0	1	0
1	0	0
1	1	1

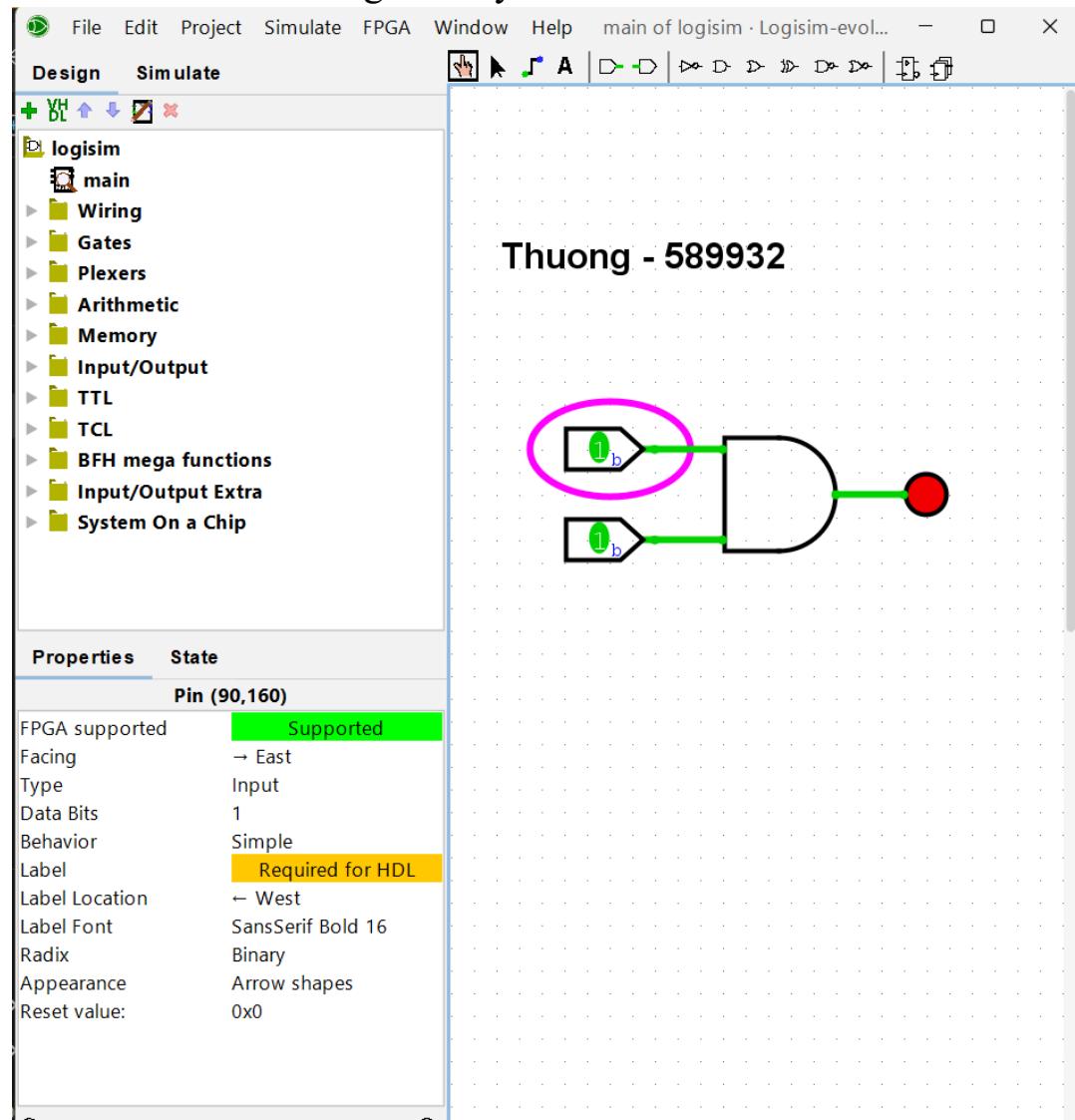
How can the design be simplified?

The circuit can be simplified by replacing all four NAND gates with one XNOR gate

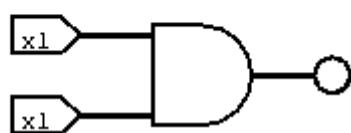
If only NAND gates are allowed, this is already the minimal NAND-only implementation of an XNOR

## **Assignment 2.4: Getting to know Logisim evolution**

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

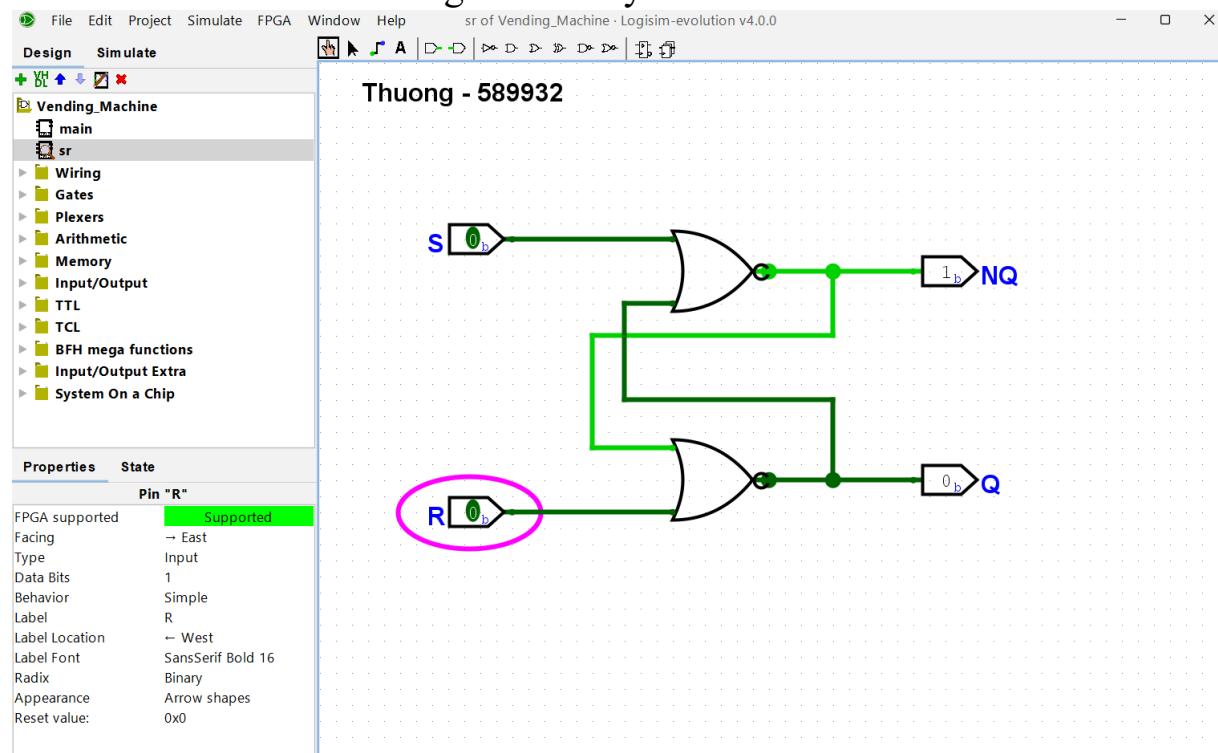


Thuong - 589932

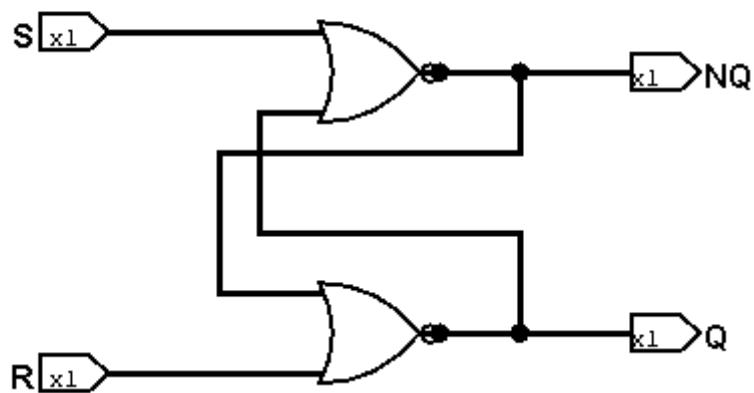


## Assignment 2.5: SR Latch

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

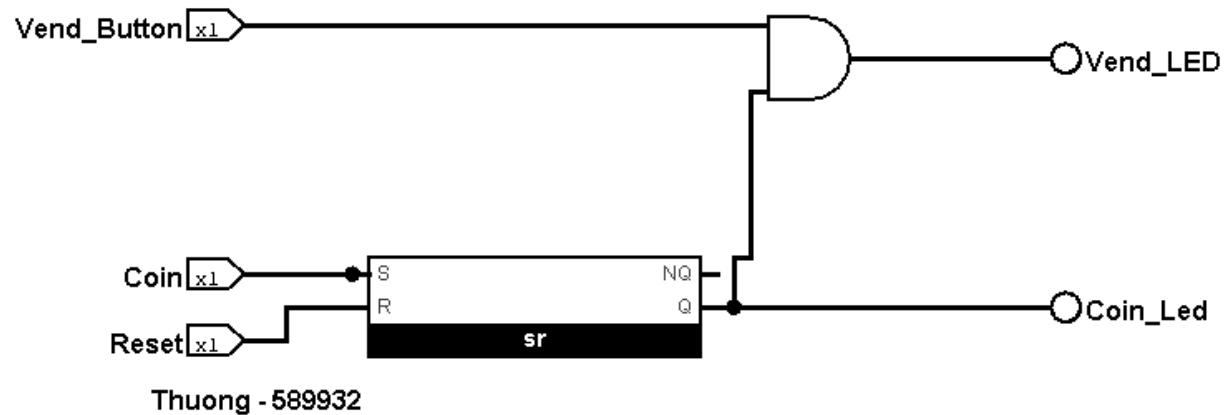
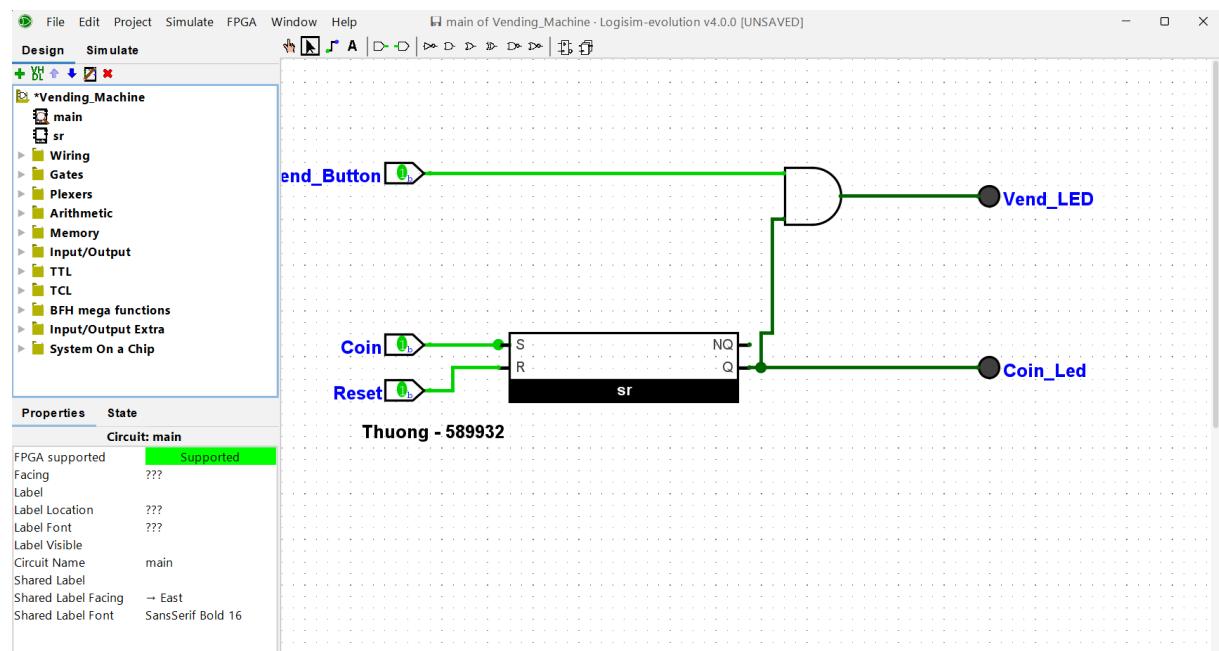


Thuong - 589932



## 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.

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

number is odd

```
public class Main {  
    public static void main(String[] args) {  
        int number = 4;  
  
        if ((number & (number - 1)) == 0 && number > 0)  
            System.out.println("number is a power of 2");  
        else  
            System.out.println("number isn't a power of 2");  
    }  
}
```

number is a power of 2

```
public class Main {  
    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.");  
    }  
}
```

User has read 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: " + userPermissions);  
    }  
}
```

User permissions: 5

<pre> public class Main {     public static void main(String[] args) {         final int READ = 4;         final int WRITE = 2;         final int EXECUTE = 1;          int userPermissions = 6;          userPermissions = userPermissions ^ WRITE;          System.out.println("User permissions: " + userPermissions);     } } </pre>	User permissions: 4
<pre> public class Main {     public static void main(String[] args) {         int number = 10;         System.out.println("Decimal integer: " + number);          String binary = Integer.toBinaryString(number);         String octal = Integer.toOctalString(number);         String hexadecimal = Integer.toHexString(number);          System.out.println("Binary representation: " + binary);         System.out.println("Octal representation: " + octal);         System.out.println("Hexadecimal representation: " + hexadecimal);     } } </pre>	Decimal integer: 10 Binary representation: 1010 Octal representation: 12 Hexadecimal representation: a

## 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 static void main(String[] args) {
        SaxionApp.start(new Application(), 300,

```

```
300);
}

public void run() {
    SaxionApp.printLine("Enter a number: ");
    int number = SaxionApp.readInt();

    SaxionApp.printLine("Choose an option:");
    SaxionApp.printLine("1. Is number odd?");
    SaxionApp.printLine("2. Is number a power
of 2?");
    SaxionApp.printLine("3. Two's complement");
    int choice = SaxionApp.readInt("Your
choice: ");

    switch (choice) {
        case 1:
            if (isOdd(number))
SaxionApp.printLine("The number is odd.");
            else SaxionApp.printLine("The
number is even.");
            break;

        case 2:
            if (isPowerOfTwo(number))
SaxionApp.printLine("The number IS a power of 2.");
            else SaxionApp.printLine("The
number is NOT a power of 2.");
            break;

        case 3:
            int result =
twosComplement(number);
            SaxionApp.printLine("Two's
complement: " + result);
            break;

        default:
            SaxionApp.printLine("Invalid
option.");
    }
}
```

```
    }

    boolean isOdd(int number) {
        return (number & 1) == 1;
    }

    boolean isPowerOfTwo(int number) {
        if (number <= 0) return false;
        return (number & (number - 1)) == 0;
    }

    int twosComplement(int number) {
        return (~number) + 1;
    }
}
```

```
Saxion Drawingboard
Enter a number:
10
Choose an option:
1. Is number odd?
2. Is number a power of 2?
3. Two's complement
3
Two's complement: -10
APPLICATION EXITED NORMALLY
```

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

