

Week 2 – Logic

Student number: 588406

Assignment 2.1: Parking lot

Which gates do you need?

You need a 3-input AND gate.

Because the parking is full only if:

$$\text{FULL} = A \wedge B \wedge C$$

Complete this table

Parking lot 1	Parking lot 2	Parking lot 3	Result (full) 1
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?

You need an XOR gate.

Reason:

- If the employee chooses Android = 1 and iPhone = 0, the output = 1
- If the employee chooses Android = 0 and iPhone = 1, the output = 1
- If the employee chooses both or none, the output = 0

This matches the XOR behavior.

Complete this table

? **A = Android chosen**

? **B = iPhone chosen**

? **Y = Valid choice** (output that is 1 only when exactly one is chosen)

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

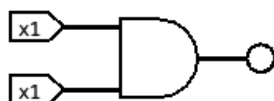
How can the design be simplified?

This circuit works like an XOR gate. So instead of using 4 NAND gates, you can use 1 XOR gate.

Assignment 2.4: Getting to know Logisim evolution

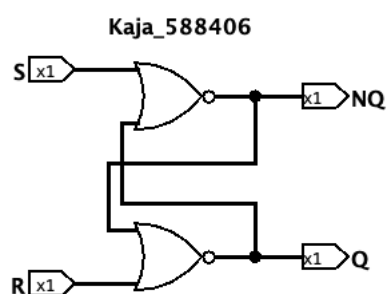
Screenshot of the design with your name and student number in it:

Kaja_588406



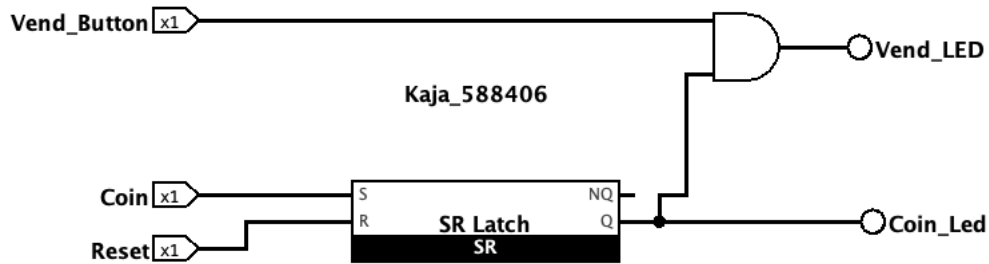
Assignment 2.5: SR Latch

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#

```

4
5 public class Application implements Runnable {
6
7     public static void main(String[] args) { SaxionApp.start(new Application(), width: 800,
8
9
10
11     public void run() {
12         System.out.println("Kaja-588406");
13         int number = 5;
14         if(((number & 1) != 0)) System.out.println("number is odd");
15         else System.out.println("number is even");
16     }
17
18 }
19

```

Run Application

```

/Downloads/Sandbox/resources/SaxionApp.jar Application
= SaxionApp version: 1.0.1 =
Kaja-588406
number is odd

```

2#

The screenshot shows an IDE with a Java file named `Application.java`. The code defines a `public class Application implements Runnable` with a `main` method and a `run` method. The `run` method prints "Kaja-588406", initializes `int number = 4`, and checks if `number` is a power of 2 using the condition `(number > 0 && (number & (number - 1)) == 0)`. The output console shows the program execution: "SaxionApp version: 1.0.1", "Kaja-588406", and "number is a power of 2".

```
5 public class Application implements Runnable {
7     public static void main(String[] args) { SaxionApp.start(new Application(), width: 800,
10
11     public void run() {
12         System.out.println("Kaja-588406");
13         int number = 4;
14         if (number > 0 && (number & (number - 1)) == 0)
15             System.out.println("number is a power of 2");
16         else
17             System.out.println("number isn't a power of 2");
18     }
19
20
21
```

Run Application

/Downloads/Sandbox/resources/SaxionApp.jar Application
= SaxionApp version: 1.0.1 =
Kaja-588406
number is a power of 2

Users > kajacingerla > Downloads > Sandbox > Sandbox8 > src > Application 19:1 LF UTF-8 4 spaces

3#

The screenshot shows an IDE with a Java file named `Application.java`. The code defines a `public class Application implements Runnable` with a `run` method. The `run` method prints "Kaja-588406", initializes `final int READ = 4`, `final int WRITE = 2`, and `final int EXECUTE = 1`. It then checks if `userPermissions` has the `READ` permission using the condition `((userPermissions & READ) != 0)`. The output console shows the program execution: "SaxionApp version: 1.0.1", "Kaja-588406", and "User has read permissions".

```
5 public class Application implements Runnable {
10
11     public void run() {
12         System.out.println("Kaja-588406");
13         final int READ = 4;
14         final int WRITE = 2;
15         final int EXECUTE = 1;
16         int userPermissions = 7;
17         if ((userPermissions & READ) != 0)
18             System.out.println("User has read permissions");
19         else
20             System.out.println("User can't read. No permissions.");
21     }
22
```

Run Application

/Downloads/Sandbox/resources/SaxionApp.jar Application
= SaxionApp version: 1.0.1 =
Kaja-588406
User has read permissions

Process finished with exit code 0

sers > kajacingerla > Downloads > Sandbox > Sandbox8 > src > Application > ru 16:33 LF UTF-8 4 spaces

4#

```
5 public class Application implements Runnable {
11     public void run() {
12         System.out.println("Kaja-588406");
13         final int READ = 4;
14         final int WRITE = 2;
15         final int EXECUTE = 1;
16         int userPermissions = READ | EXECUTE;
17         System.out.println("User permissions: " + userPermissions);
18     }
19 }
20 }
21
```

Run Application

/Users/kajacingerla/Downloads/Sandbox/out/production/Sandbox8:/Users/kajacingerla/Downloads/Sandbox/resources/SaxionApp.jar Application
= SaxionApp version: 1.0.1 =
Kaja-588406
User permissions: 5

5#

```
5 public class Application implements Runnable {
11     public void run() {
12         System.out.println("Kaja-588406");
13         final int READ = 4;
14         final int WRITE = 2;
15         final int EXECUTE = 1;
16         int userPermissions = 6;
17         userPermissions = userPermissions ^ WRITE;
18         System.out.println("User permissions: " + userPermissions);
19     }
20 }
21 }
22
```

Run Application

SaxionApp version: 1.0.1 =
Kaja-588406
User permissions: 4
Process finished with exit code 0

6#

```
5 public class Application implements Runnable {
11     public void run() {
12         System.out.println("Kaja-588406");
13         int number = 5;
14         number = ~number + 1;
15         System.out.println("Number (negative): " + number);
16         number = ~number + 1;
17         System.out.println("Number (positive again): " + number);
18     }
19 }
20
21
```

Run Application

/Downloads/Sandbox/resources/SaxionApp.jar Application
= SaxionApp version: 1.0.1 =
Kaja-588406
Number (negative): -5
Number (positive again): 5

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 java.awt.*;

public class Application implements Runnable {

    public static void main(String[] args) { SaxionApp.start(new Application(), width: 800, height: 600); }

    public void run() {
        System.out.println("Kaja-588406");

        int number = 5;
        int choice = 1;

        System.out.println("Number: " + number);
        System.out.println("1. Is number odd?");
        System.out.println("2. Is number a power of 2?");
        System.out.println("3. Two's complement of number?");
        System.out.println("Chosen option: " + choice);
    }
}
```

Run Application

= SaxionApp version: 1.0.1 =
Kaja-588406
Number: 5
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
Chosen option: 1
Number is odd

```
public class Application implements Runnable {

    public void run() {
        System.out.println("3. Two's complement of number?");
        System.out.println("Chosen option: " + choice);

        if (choice == 1) {
            if ((number & 1) != 0) {
                System.out.println("Number is odd");
            } else {
                System.out.println("Number is even");
            }
        }

        if (choice == 2) {
            if (number > 0 && (number & (number - 1)) == 0) {
                System.out.println("Number is a power of 2");
            } else {
                System.out.println("Number is NOT a power of 2");
            }
        }

        if (choice == 3) {
            int twosComplement = ~number + 1;
            System.out.println("Two's complement: " + twosComplement);
        }
    }
}
```

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

