

Week 2 – Logic

Student number: 585732

Assignment 2.1: Parking lot

Which gates do you need?

= AND GATE

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	0	1	0
1	1	0	0
1	1	1	1

Assignment 2.2: Android or iPhone

Which gates do you need?

= OR GATE

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

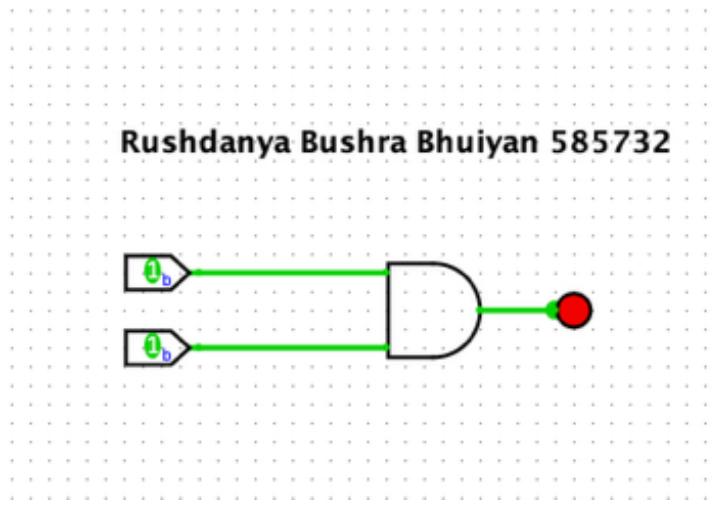
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

How can the design be simplified?

= As XOR gate also works similar, we can replace the 4 NAND gates into one XOR gate and we will still get the same result.

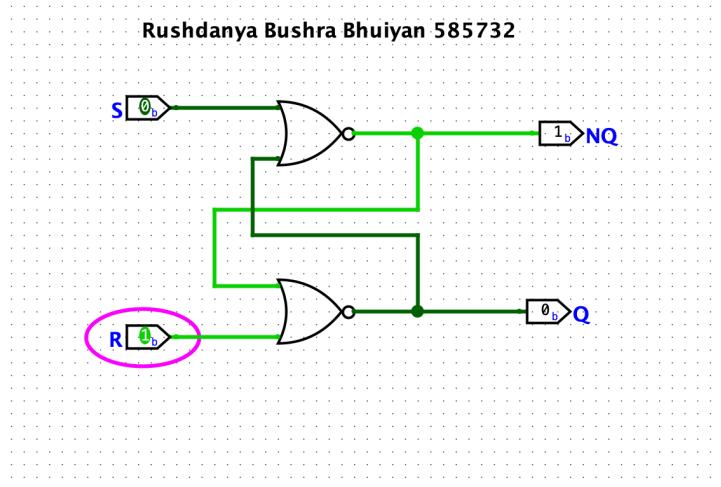
Assignment 2.4: Getting to know Logisim evolution

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



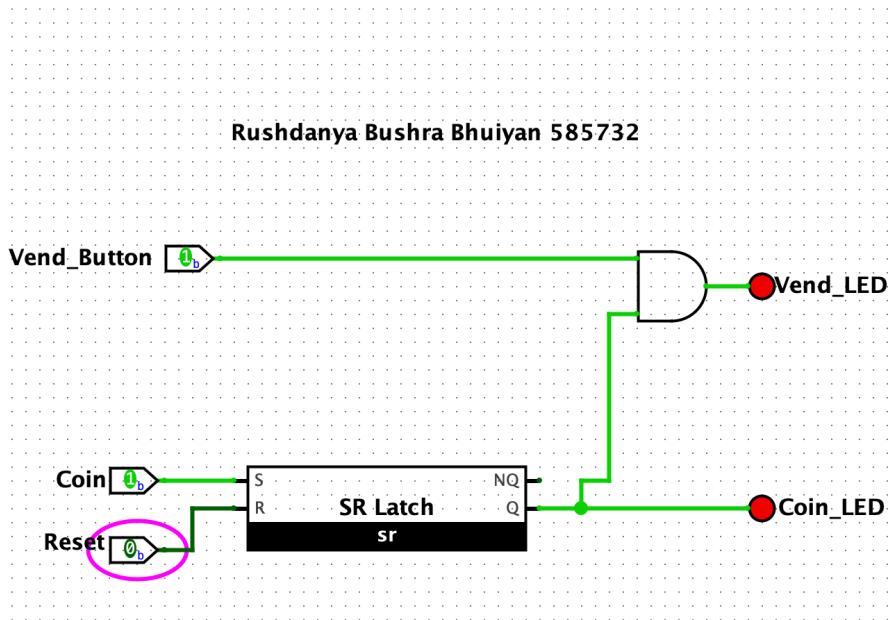
Assignment 2.5: SR Latch

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 public class Main {  
    public static void main(String[] args) {  
        int number = 2;  
  
        if ((number & 1) == 1) {  
            System.out.println("number is odd");  
        } else {  
            System.out.println("number is even");  
        }  
    }  
}
```

```
#2 public class Main {  
    public static void main(String[] args) {  
        int number = 2;  
  
        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 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 ((READ & userPermissions) != 0) {
            System.out.println("User has read permissions");
        } else {
            System.out.println("User can't read. No permissions.");
        }
    }
}
```

```
#4 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);
    }
}
```

```
#5 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);
}
}

```

```

#6 public class Main {

public static void main(String[] args) {
    int number = 5;
    number = ~number + 1;

    System.out.println("Number: " + number);
}
}

```

```

#7 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);
}
}

```

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(), 800, 800);
    }

    public void run() {
        int counter = 0;

        while (counter < 1) {
            drawMenu();

            int input = SaxionApp.readInt();

            if (input == 0) {
                break;
            } else if (input == 1) {
                isNumberOdd();
            } else if (input == 2) {
                isNumberPowerTwo();
            } else if (input == 3) {
                convertIntoNegative();
            }
        }
    }

    public void isNumberOdd() {
        SaxionApp.printLine("Pick a number to check if its odd");
        int input = SaxionApp.readInt();

        if ((input & 1) == 1) {
            SaxionApp.printLine("number is odd");
        } else {
            SaxionApp.printLine("number is even");
        }

        SaxionApp.pause();
    }
}
```

```
SaxionApp.clear();
}

public void isNumberPowerTwo() {
    SaxionApp.printLine("Pick a number to check if its power of 2");
    int input = SaxionApp.readInt();

    if (input > 0 && (input & (input - 1)) == 0) {
        SaxionApp.printLine("number is a power of 2");
    } else {
        SaxionApp.printLine("number isn't a power of 2");

    }
    SaxionApp.pause();
    SaxionApp.clear();
}

public void convertIntoNegative() {
    SaxionApp.printLine("Pick a number to convert into negative");
    int input = SaxionApp.readInt();

    input = ~input + 1;
    SaxionApp.printLine("Negative number: " + input);

    SaxionApp.pause();
    SaxionApp.clear();
}

public void drawMenu() {
    SaxionApp.printLine("MENU");
    SaxionApp.printLine("1. Check if a number is odd");
    SaxionApp.printLine("2. Check if a number is a power of 2");
    SaxionApp.printLine("3. Turn a number into negative");
    SaxionApp.printLine("0. Close application");
}
}
```

