Week 2 – Logic

Student number: 569091

Contents

Assignment 2.1: Parking lot

Gates

To decide when the parking lot is full, we need an AND gate. The AND gate outputs 1 (true) only when all inputs are 1.

Table

Complete 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

Gates

A XOR gate, because: XOR outputs 1 (true) when exactly one input is 1 (true).

XOR outputs 0 (false) when:

1. Both inputs are 0 (neither phone is chosen).
2. Both inputs are 1 (both phones are chosen, which is invalid in this scenario).

Table

Android phone	**iPhone**	**Result (Phone in possession)**	**Explanation**
0	0	0	**0, 0 -> 0:** Neither phone is chosen, so the result is 0 (no phone in possession).
0	1	1	**0, 1 -> 1:** Only the iPhone is chosen, so the result is 1 (phone in possession).
1	0	1	**1, 0 -> 1:** Only the Android phone is chosen, so the result is 1 (phone in possession).
1	1	0	**1, 1 -> 0:** Both phones are chosen, which is invalid, so the result is 0 (no valid choice).

Assignment 2.3: Four NAND gates

Table

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

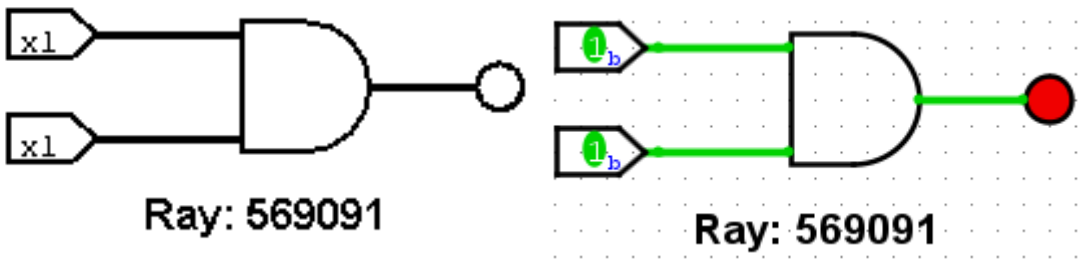
Simplified design

Instead of using 4 NAND gates to achieve the XOR function:

Direct XOR Gate: Use a single XOR gate to replace all 4 NAND gates.

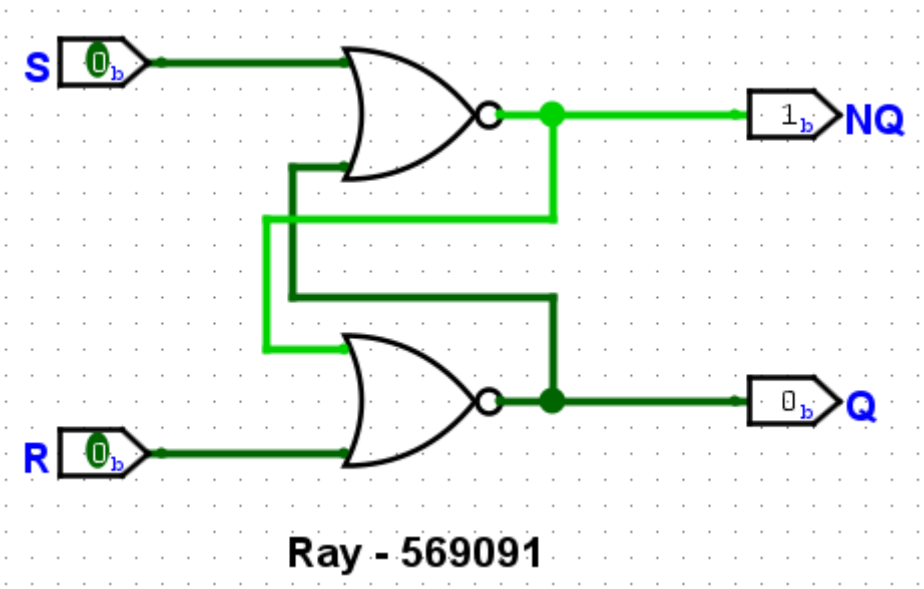
Assignment 2.4: Getting to know Logisim evolution

Screenshot of the design with student name and number:



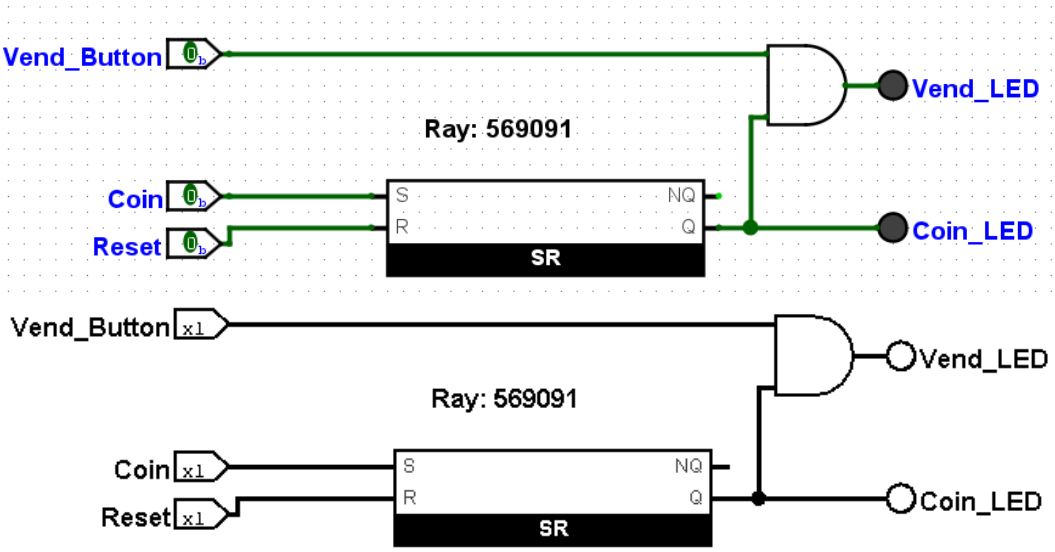
Assignment 2.5: SR Latch

Logisim SR Latch screenshot with student name and number:



Assignment 2.6: Vending Machine

Logisim Vending Machine screenshot with student name and number:



Assignment 2.7: Bitwise operators

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

#1 even or odd

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

Condition: (number & 1) == 1

Checks LSB: odd ends with 1, even ends with 0. & 1 masks all but LSB.

#2 Power of 2

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

Condition: (number & (number - 1)) == 0

Powers of 2 have exactly one 1 bit. n-1 flips all trailing 0s to 1s, so & gives 0.

#3 Check permissions (verse file)

```
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.");
	}
}
```

Condition: (userPermissions & READ) == READ
Masks permission bits. If result equals READ (4), user has read access.

#4 Assign permissions

```
public class Main {
	public static void main(String[] args) {
		final int READ = 4;
		final int WRITE = 2;
		final int EXECUTE = 1;
		int userPermissions = READ | EXECUTE;
		System.out.println("User permissions: "+userPermissions);
	}
}
```

Value: READ | EXECUTE // 4 | 1 = 5
OR combines bits: 100 | 001 = 101 (decimal 5 = read+execute).

#5 Update permissions

```
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);
	}
}
```

Value: userPermissions ^ WRITE; // 6 ^ 2 = 4 (READ only)

XOR toggles bits: 110 ^ 010 = 100 (removes WRITE, keeps READ).

#6 Two's complement

```
public class Main {
	public static void main(String[] args) {
		int number = 5;
		number = ~number + 1;
		System.out.println("Number: "+number);
	}
}
```

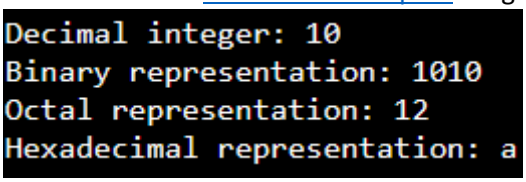
Value: ~number + 1; // 5 → -5, -5 → 5

Flips all bits (~), adds 1. Converts positive ↔ negative in 2's complement.

#7 Display binary, octal and hexadecimal values

```
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);
	}
}
```

When ran in the Online Java Compiler we get:



Assignment 2.8: Java Application Bit Calculations

Assignment

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.

Menu options

```
// 1. Is number odd?

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

// 2. Is number a power of 2?

public class Main {
    public static void main(String[] args) {
        int number = 4;

        // check if the number is a power of 2 using bitwise & operator
        // compare number with number -1 (always get false)
        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.  Two's complement of number?

public class Main {
  public static void main(String[] args) {
    int number = 5;
    number = ~number + 1; // Two's complement
    System.out.println("Number: " + number);
  }
}
```

Source Code

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

import java.awt.*;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 500, 800);
    }

    @Override
    public void run() {
        boolean running = true;

        while (running) {
            SaxionApp.print("What number would you like to check? ");
            int number = SaxionApp.readInt();

            SaxionApp.printLine("Select an option:", Color.blue);
            SaxionApp.printLine("1. Check if the number is odd", Color.green);
            SaxionApp.printLine("2. Check if the number is a power of two", Color.green);
            SaxionApp.printLine("3. Find the two's complement of the number", Color.green);
            SaxionApp.printLine("4. Exit", Color.red);

            int keyInput = SaxionApp.readInt();

            if (keyInput == 1) {
                checkOddNumber(number);
            } else if (keyInput == 2) {
                checkPowerOfTwo(number);
            } else if (keyInput == 3) {
                checkTwosComplement(number);
            } else if (keyInput == 4) {
                SaxionApp.printLine("Exiting program", Color.red);
                running = false;
            } else {
                SaxionApp.printLine("Invalid option. Please try again.", Color.red);
            }
        }
    }

    // Method to check if a number is odd
    public static void checkOddNumber(int number) {
        if ((number & 1) == 1) { // Check the LSB
            SaxionApp.printLine("Number is odd",Color.red);
        } else {
            SaxionApp.printLine("Number is even",Color.green);
        }
    }

    // Method to check if a number is a power of 2
    public static void checkPowerOfTwo(int number) {
        if (number > 0 && (number & (number - 1)) == 0) {
            SaxionApp.printLine("Number is a power of 2", Color.green);
        } else {
            SaxionApp.printLine("Number isn't a power of 2",Color.red);
        }
    }

    // Method to calculate the two's complement of a number
    public static void checkTwosComplement(int number) {
        int twosComplement = ~number + 1; // Two's complement operation
        SaxionApp.printLine("Two's complement of " + number + " is: " + twosComplement);
    }
}
```

Screenshot

Added a simple while loop to demonstrate all methods work with one screenshot.