Template Week 2 – Logic

Student number: 547201

Assignment 2.1: Parking lot

Which gates do you need?

AND logic 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
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

Assignment 2.2: Android or iPhone

Which gates do you need?

XOR logic 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

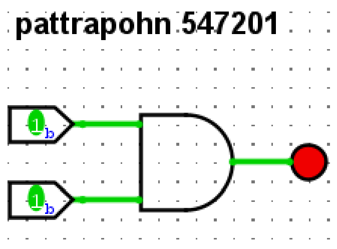
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

How can the design be simplified?

The whole 4-NAND circuit simplifies to A XOR B Which can be implemented as one XOR gate, or four NAND gates (as shown), or 2 NOT + AND + OR gates.

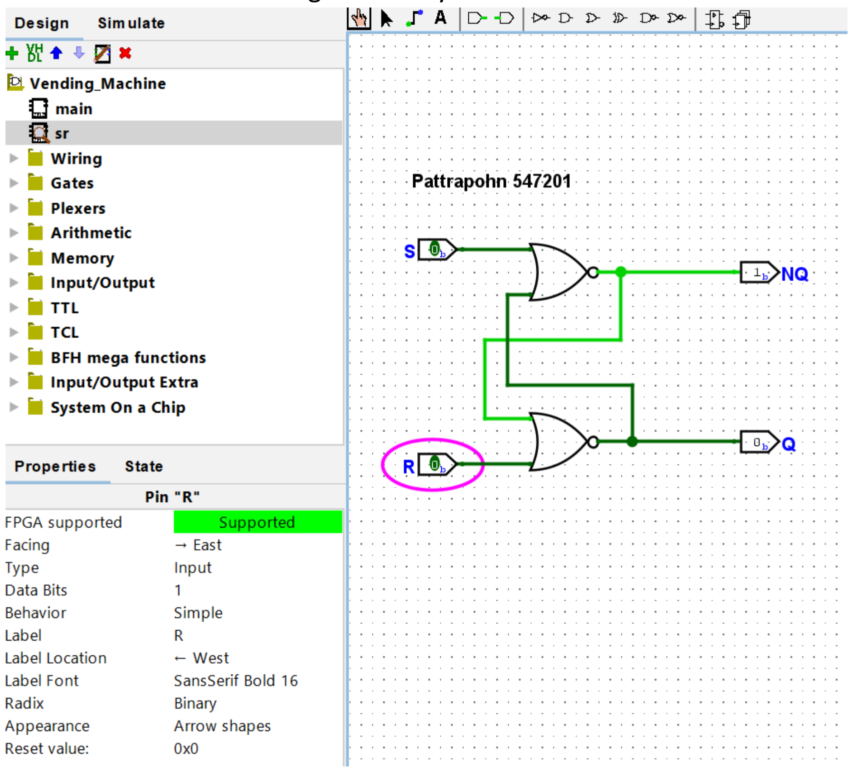
Assignment 2.4: Getting to know Logisim evolution

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:

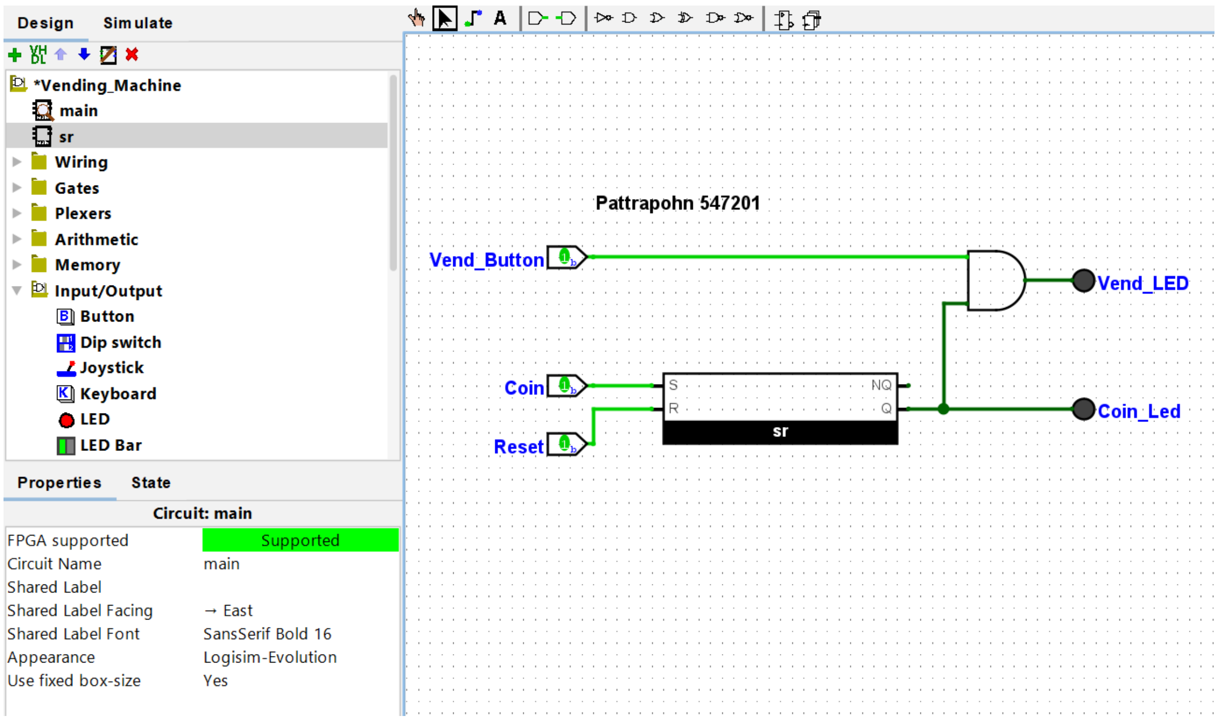


Pin "R"

FPGA supported	Supported
Facing	→ East
Type	Input
Data Bits	1
Behavior	Simple
Label	R
Label Location	← West
Label Font	SansSerif Bold 16
Radix	Binary
Appearance	Arrow shapes
Reset value:	0x0

Assignment 2.6: Vending Machine

Screenshot Vending Machine in Logisim with your name and student number:



Circuit: main

FPGA supported	Supported
Circuit Name	main
Shared Label	
Shared Label Facing	→ East
Shared Label Font	SansSerif Bold 16
Appearance	Logisim-Evolution
Use fixed box-size	Yes

Assignment 2.7: Bitwise operators

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

```
public void run() {
  // 1: Even or Odd
  SaxionApp.printLine("=== 1: Even or Odd ===");
  int number = 5;
  if((number & 1) == 1) SaxionApp.printLine("number is odd");
  else SaxionApp.printLine("number is even");
```

even numbers always end with 0 and odd always end with 1
& is the bitwise AND operator that compare each bit of number
5 = 101, 1 = 001, so 101 & 001 = 001 = 1 decimal

```
  // 2: Power of 2
  SaxionApp.printLine("=== 2: Power of 2 ===");
  number = 4;
  if((number > 0) && ((number & (number - 1)) == 0))
    SaxionApp.printLine("number is a power of 2");
  else
    SaxionApp.printLine("number isn't a power of 2");
```

Powers of 2 number has one 1 bit always 1 on the left
number-1 has all 1s to the right and when use & bitwise it's always give 0

```
    // 3: Check permissions
    SaxionApp.printLine("=== 3: Check Permissions ===");
    final int READ = 4;
    final int WRITE = 2;
    final int EXECUTE = 1;

    int userPermissions = 7;

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

    // 4: Assign permissions
    SaxionApp.printLine("=== 4: Assign Permissions ===");
    userPermissions = 0;
    userPermissions = READ | EXECUTE;
    SaxionApp.printLine("User permissions: "+userPermissions);
```

| is bitwise OR
100(4) | 001(1) = 101 = 5

```
    // 5: Update permissions
    SaxionApp.printLine("=== 5: Update Permissions ===");
    userPermissions = 6;
    userPermissions = userPermissions ^ WRITE;
    SaxionApp.printLine("User permissions: "+userPermissions);
```

^ is bitwise XOR, if there's only one 1 bit
WRITE = 010 XOR 100 = 100 = 4 then write remove

```
    // 6: Two's complement
    SaxionApp.printLine("=== 6: Two's Complement ===");
    number = 5;
    number = ~number + 1;
    SaxionApp.printLine("Number: "+number);
```

~ is bitwise NOT, it flip all bit and + 1

```
    // 7: Display number systems
    SaxionApp.printLine("=== 7: Number System Conversion ===");
    number = 10;
    SaxionApp.printLine("Decimal integer: "+number);
    SaxionApp.printLine("Binary representation: " + Integer.toBinaryString(number));
    SaxionApp.printLine("Octal representation: " + Integer.toOctalString(number));
    SaxionApp.printLine("Hexadecimal representation: " + Integer.toHexString(number));

}
```

Assignment 2.8: Java Application Bit Calculations

Create a java program that accepts user input and presents a menu with options.

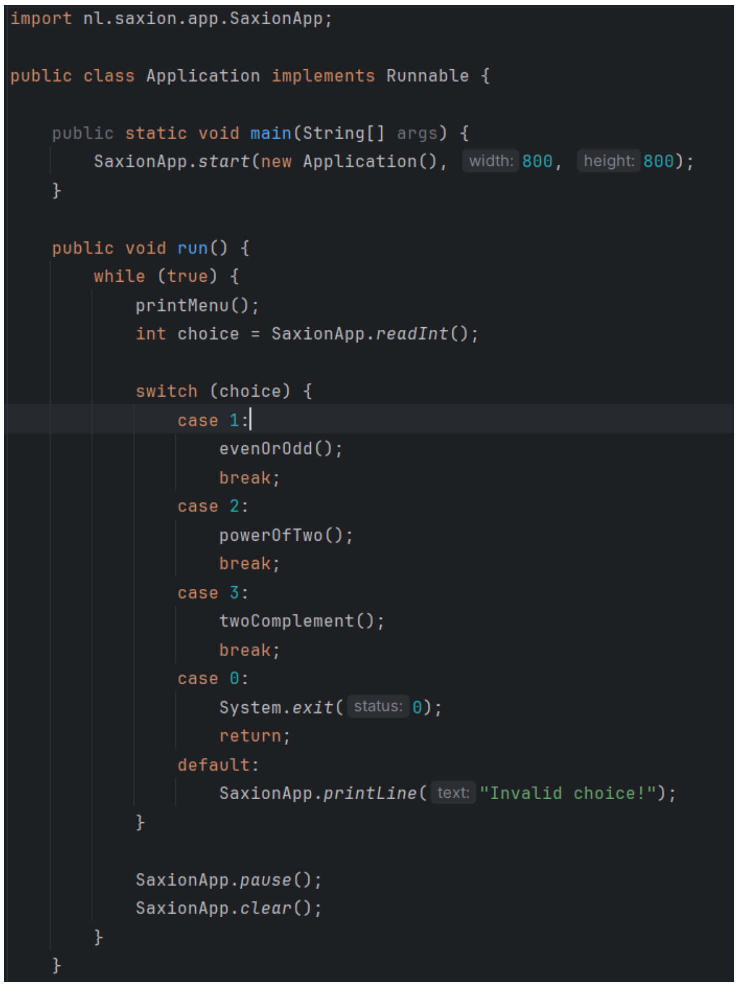
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?

Implement the methods by using the bitwise operators you have just learned.

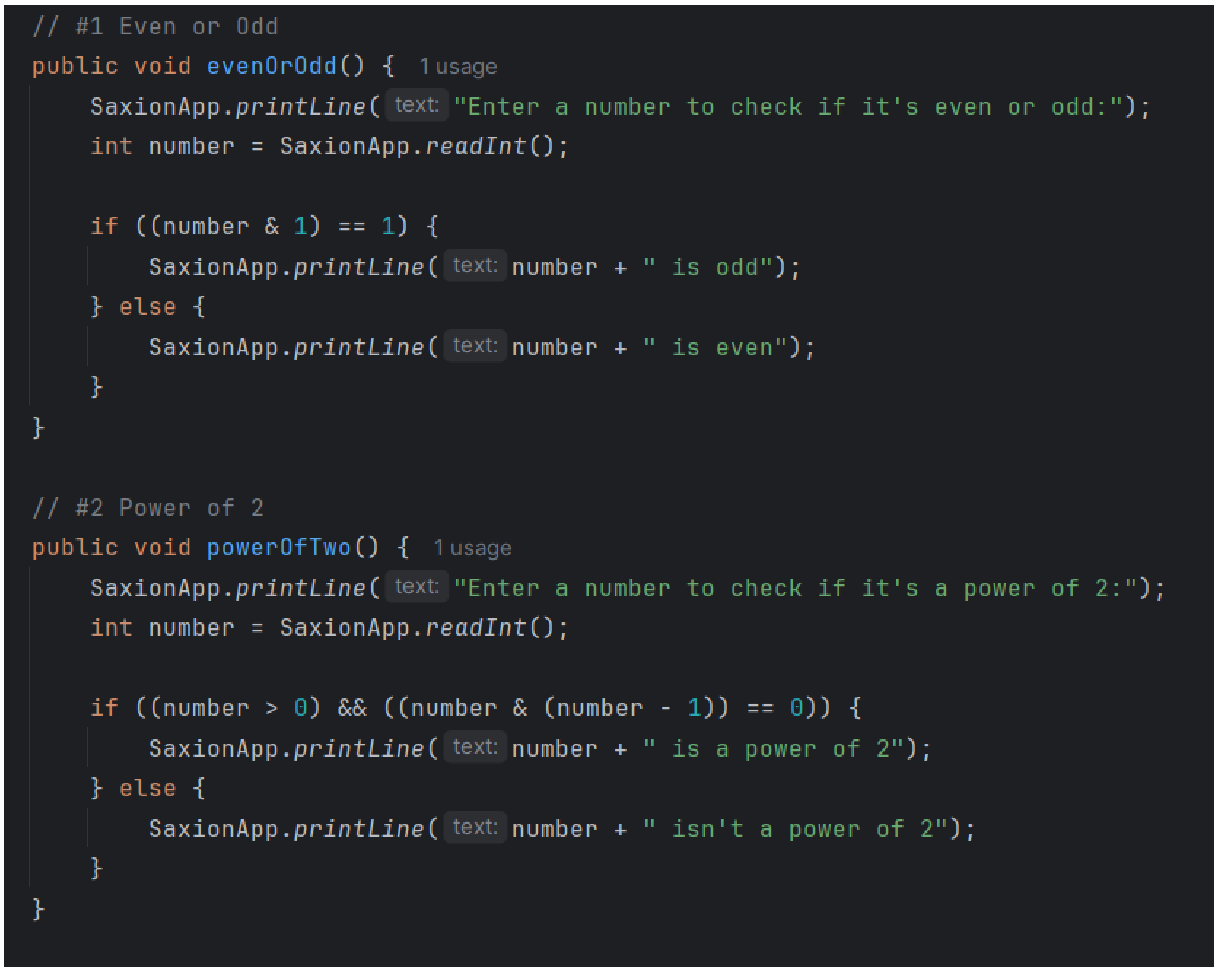
Organize your source code in a readable manner with the use of control flow and methods.

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

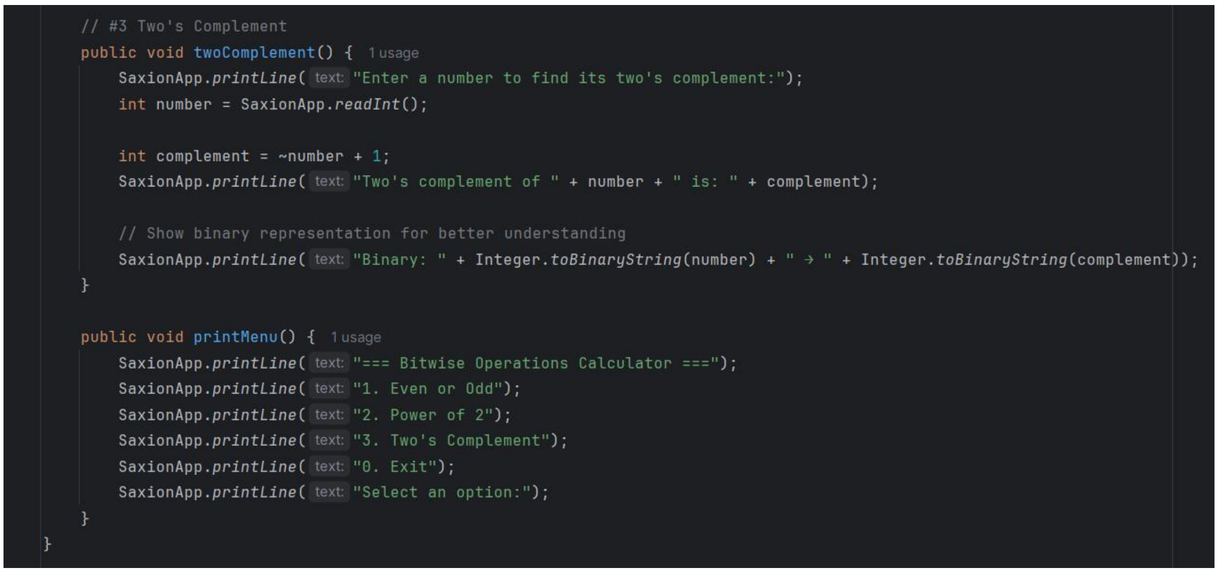
Paste source code here, with a screenshot of a working application.



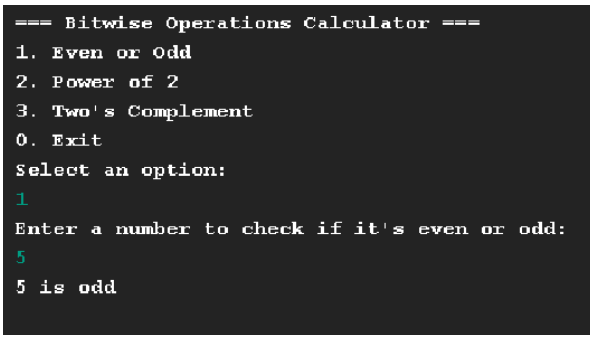
```
import nl.saxion.app.SaxionApp;

public class Application implements Runnable {

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

    public void run() {
        while (true) {
            printMenu();
            int choice = SaxionApp.readInt();

            switch (choice) {
                case 1:
                    evenOrOdd();
                    break;
                case 2:
                    powerOfTwo();
                    break;
                case 3:
                    twoComplement();
                    break;
                case 0:
                    System.exit( status: 0);
                    return;
                default:
                    SaxionApp.printLine( text: "Invalid choice!");
            }

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



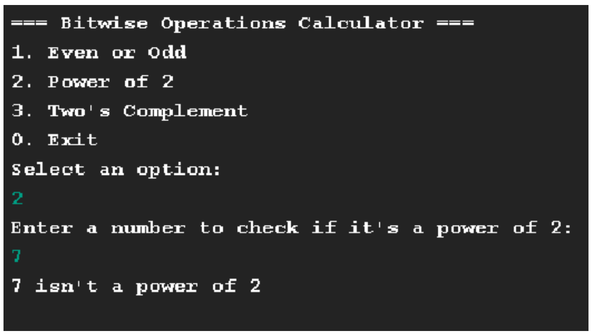
```java
// #1 Even or Odd
public void evenOrOdd() {  1 usage
    SaxionApp.printLine( text: "Enter a number to check if it's even or odd:");
    int number = SaxionApp.readInt();

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

// #2 Power of 2
public void powerOfTwo() {  1 usage
    SaxionApp.printLine( text: "Enter a number to check if it's a power of 2:");
    int number = SaxionApp.readInt();

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



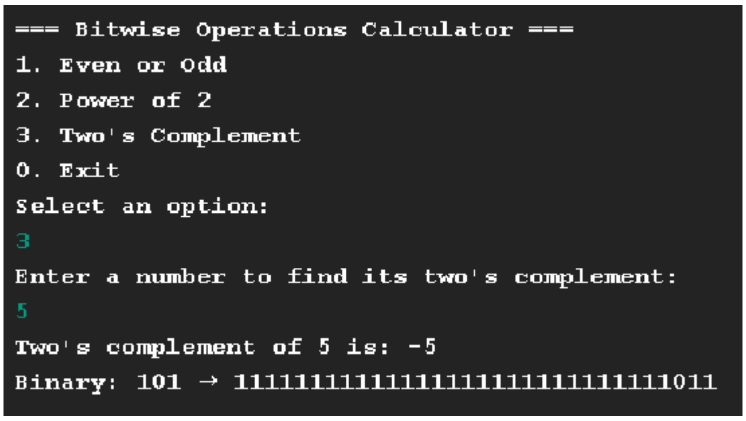
```java
    // #3 Two's Complement
    public void twoComplement() {  1 usage
        SaxionApp.printLine( text: "Enter a number to find its two's complement:");
        int number = SaxionApp.readInt();

        int complement = ~number + 1;
        SaxionApp.printLine( text: "Two's complement of " + number + " is: " + complement);

        // Show binary representation for better understanding
        SaxionApp.printLine( text: "Binary: " + Integer.toBinaryString(number) + " → " + Integer.toBinaryString(complement));
    }

    public void printMenu() {  1 usage
        SaxionApp.printLine( text: "=== Bitwise Operations Calculator ===");
        SaxionApp.printLine( text: "1. Even or Odd");
        SaxionApp.printLine( text: "2. Power of 2");
        SaxionApp.printLine( text: "3. Two's Complement");
        SaxionApp.printLine( text: "0. Exit");
        SaxionApp.printLine( text: "Select an option:");
    }
}
```



```
=== Bitwise Operations Calculator ===
1. Even or Odd
2. Power of 2
3. Two's Complement
0. Exit
Select an option:
1
Enter a number to check if it's even or odd:
5
5 is odd
```



```
=== Bitwise Operations Calculator ===
1. Even or Odd
2. Power of 2
3. Two's Complement
0. Exit
Select an option:
2
Enter a number to check if it's a power of 2:
7
7 isn't a power of 2
```



```
=== Bitwise Operations Calculator ===
1. Even or Odd
2. Power of 2
3. Two's Complement
0. Exit
Select an option:
3
Enter a number to find its two's complement:
5
Two's complement of 5 is: -5
Binary: 101 → 11111111111111111111111111111011
```