Template Week 2 – Logic

Student number: 593201

Assignment 2.1: Parking lot

Which gates do you need?

- A 3-input AND gate will output 1 (FULL) only when Parking lot 1, 2, and 3 are all 1.
- This matches the requirement that the sign lights up only when all spaces are occupied.

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?

You need an XOR (Exclusive OR) gate.

- An XOR gate outputs 1 only when one input is 1 and the other is 0.
- If both are 0 (no phone chosen) or both are 1 (both phones chosen), the output is 0.

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?

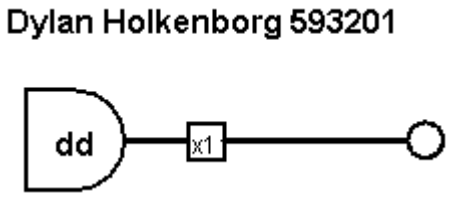
- The current design uses four NAND gates to implement the logic.
- Since the resulting behavior is XOR, the circuit can be simplified by:
    - Replacing the four NAND gates with one XOR gate, or
    - Using fewer gates if XOR is available as a standard logic component.

Conclusion:

- The four-NAND-gate circuit is a NAND-only implementation of an XOR gate.
- Using a single XOR gate would make the circuit simpler, smaller, and more efficient.

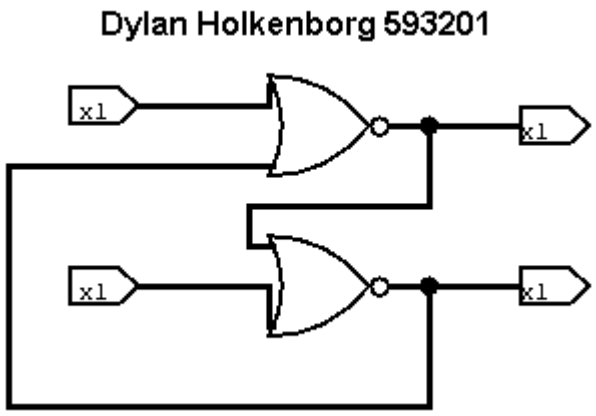
Assignment 2.4: Getting to know Logisim evolution

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



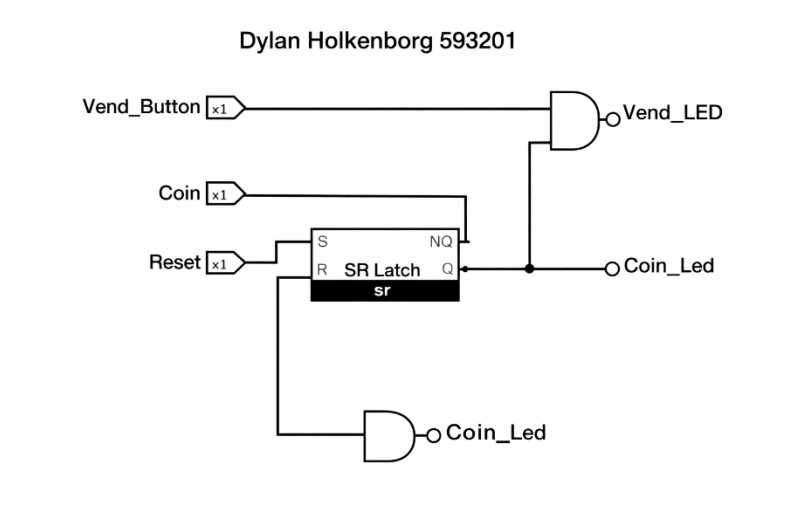
Assignment 2.5: SR Latch

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.

```
public class Main {
  public static void main(String[] args) {
    int number = 6;

    if ((number & 1) == 1) System.out.println("number is odd");
    else System.out.println("number is even");
  }
}
```

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.


```
public class Main {
  public static void main(String[] args) throws Exception {
    int number = readInt("Enter an integer: ");
    while (true) {
      printMenu();
      int option = readInt("Choose option: ");
      if (option == 0) {
        System.out.println("Bye!");
        break;
      }
      switch (option) {
        case 1:
          System.out.println("Result: " + number + (isOdd(number) ? " is odd" : " is even"));
          break;
        case 2:
          System.out.println("Result: " + number + (isPowerOfTwo(number) ? " IS" : " is NOT") + " a power
of 2");
          break;
        case 3:
          System.out.println("Result: two's complement (bitwise) = " + twosComplement(number));
          break;
        default:
          System.out.println("Invalid option. Choose 0-3.");
          break;
      }
    }
  }

  private static void printMenu() {
    System.out.println();
    System.out.println("Menu:");
    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("0) Exit");
  }
  // Reads an integer from stdin (no Scanner).
  private static int readInt(String prompt) throws Exception {
    while (true) {
      System.out.print(prompt);
      String line = readLine();
```

```
    line = trim(line);
    if (line.length() == 0) continue;
    boolean negative = false;
    int i = 0;
    if (line.charAt(0) == '-') {
      negative = true;
      i = 1;
      if (line.length() == 1) continue;
    }
    int value = 0;
    boolean ok = true;
    while (i < line.length()) {
      char c = line.charAt(i);
      if (c < '0' || c > '9') {
        ok = false;
        break;
      }
      int digit = c - '0';
      // basic overflow-safe build (still fine for normal inputs)
      value = value * 10 + digit;
      i++;
    }
    if (!ok) {
      System.out.println("Please enter a valid integer.");
      continue;
    }
    return negative ? -value : value;
  }
 }
 // Reads a line from System.in using System.in.read().
 private static String readLine() throws Exception {
  StringBuilder sb = new StringBuilder();
  int ch;
  while ((ch = System.in.read()) != -1) {
    if (ch == '\r') continue;
    if (ch == '\n') break;
    sb.append((char) ch);
  }
  return sb.toString();

 // Trim spaces/tabs without using imports
 private static String trim(String s) {
  int start = 0;
  int end = s.length() - 1;
  while (start <= end && isWhitespace(s.charAt(start))) start++;
  while (end >= start && isWhitespace(s.charAt(end))) end--;
  if (start > end) return "";
  return s.substring(start, end + 1);
```

```
  }
  private static boolean isWhitespace(char c) {
    return c == ' ' || c == '\t';
  }
  // 1) Odd if LSB is 1
  private static boolean isOdd(int n) {
    return (n & 1) != 0;
  }
  // 2) Power of 2: n > 0 and only one bit set
  private static boolean isPowerOfTwo(int n) {
    return n > 0 && (n & (n - 1)) == 0;
  }
  // 3) Two's complement using bitwise ops: ~n + 1
  private static int twosComplement(int n) {
    return (~n) + 1;
  }
}
```

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