## Problem 1: Implement Exercise 2.35 Using Logisim

A close-up of a text

AI-generated content may be incorrect.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | **A3 A2 A1 A0** | | | **Prime (P)** | | **Divisible by 3 (D)** | | |
| 0 | 0000 | | | 0 | | 1 | |
| 1 | 0001 | | | 0 | | 0 | |
| 2 | 0010 | | | 1 | | 0 | |
| 3 | 0011 | | | 1 | | 1 | |
| 4 | 0100 | | | 0 | | 0 | |
| 5 | 0101 | | | 1 | | 0 | |
| 6 | 0110 | | | 0 | | 1 | |
| 7 | 0111 | | | 1 | | 0 | |
| 8 | 1000 | | | 0 | | 0 | |
| 9 | 1001 | | | 0 | | 1 | |
| 10 | 1010 | | | 0 | | 0 | |
| 11 | 1011 | | | 1 | | 0 | |
| 12 | 1100 | | | 0 | | 1 | |
| 13 | 1101 | | | 1 | | 0 | |
| 14 | 1110 | | | 0 | | 0 | |
| 15 | 1111 | | | 0 | | 1 | |
| **Prime KMap** | | Y = A1A2 + A3A0 + A3’A2A1’A0 + A3A2A1A0’ | | | | |
| **A3A2\A1A0** | | 00 | 01 | | 11 | 10 |
| 00 | | 0 | 0 | | 1 | 1 |
| 01 | | 0 | 1 | | 1 | 1 |
| 11 | | 0 | 1 | | 0 | 1 |
| 10 | | 0 | 1 | | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Divisible by 3 KMap** | Y = A1’A0’ + A1A0 | | | |
| **A3A2\A1A0** | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | 0 |
| 01 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 |

A diagram of a circuit

AI-generated content may be incorrect.

#### **Problem 2: Implement a 1-Bit 2:1 Multiplexer Using AND, OR, and NOT Gates**

#### **Introduction to Circuits in Logisim**

Construct a **1-bit 2:1 multiplexer** using **only AND, OR, and NOT gates** in Logisim. Your multiplexer should take two data inputs (**A, B**) and a select line (**S**) to determine which input is passed to the output.

|  |  |  |  |
| --- | --- | --- | --- |
| **S (Select)** | **A (Input 0)** | **B (Input 1)** | **Y (Output)** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Multiplexer** | Y = AS’ + BS | | |
| **AB\S** | 0 | 1 |
| 00 | 0 | 0 |
| 01 | 0 | 1 |
| 11 | 1 | 1 |
| 10 | 1 | 0 |

### 1 Bit MUX Circuit

A diagram of a building

AI-generated content may be incorrect.

### 1 Bit MUX Circuit Test

A diagram of a computer program

AI-generated content may be incorrect.

**Problem 3: Implement a 4-Bit AND Module Using Input and Output Buses**

Create a **4-bit AND module** in Logisim that takes **two 4-bit input buses** and produces a **4-bit output bus**, applying bitwise AND to each pair of bits.

A diagram of a circuit

AI-generated content may be incorrect.

**Problem 4: Implement a BCD to 7-Segment Decoder with Error Handling**

A **7-segment decoder** converts **Binary-Coded Decimal (BCD)** inputs into control signals for a **7-segment display**. In this exercise, you will build a **decoder module** that:

1. Displays **digits 0-9** normally.
2. Shows **"Er"** when the input is **greater than 9**.

Hi, I am Tam, the president of the Hive collective, a new ECS club that prioritizing on actionable and safe environment for entrepreneurship. What are some resources besides the Carlson center, out there for us?

Instructional related activities – Venture Studio Model – Accelerators LLC– Equity(partner) vs Licensing

Operating Agreement - Milestone –

Warrant for Grant

Grant warrant for equity

Safe agreement

Entity will contractz