

CS303 Digital Design

**Experiment 1**

**Number Systems**

**Objectives:**

After completing this experiment, you will be able to

* Learn about the LOGISIM simulator.
* Convert binary or binary coded decimal (BCD) numbers to decimal.
* Construct a portion of a digital system that decodes a BCD number and displays it on a seven-segment display.

**Theory:**

The number of symbols in a number system is called the *base,* or *radix*, of that system. The decimal number system uses ten counting symbols, the digits 0 through 9, to represent quantities. Thus, it is a base ten system. In this system, we represent quantities larger than 9 by using positional weighting of the digits. The position, or column, that a digit occupies indicates the weight of that digit in determining the value of the number. The base 10 number system is a weighted system because each column has a value associated with it.

Digital systems use two states to represent quantities and thus are *binary* in nature. The binary counting system has a radix of two and uses only the digits 0 and 1. (These are often called *bits,* which is a contraction of Binary digit). It too is a weighted counting system, with each column value worth twice the value of the column to the immediate right. Because binary numbers have only two digits, large numbers expressed in binary require a long string of 0s and 1s. Other systems, which are related to binary in a simple way, are often used to simplify these numbers. These systems include octal, hexadecimal, and BCD.

The BCD system uses four binary bits to represent each decimal digit. It is a convenient code because it allows ready conversion from base ten to a code that a machine can understand; however, it is wasteful of bits. A 4-bit binary number could represent the numbers 0 to 15, but in BCD it represents only the quantities 0 through 9. The binary representations of the numbers 10 through 15 are not used in BCD and are invalid.

The conversion of BCD to a form that can be read by humans is a common problem in digital systems. A familiar display is called the seven-segment display, which is used in many digital applications such as clocks. A basic seven-segment display is described in this experiment with the instructions for wiring it. Later, in the class, the inner workings of the decoder and seven-segment display are explored.

You will construct a simplified version of the display for this experiment.

**Procedure:**

Step 1: Download the library from Teams/Files/Lab/EXP1.

Step 2: Then create a new circuit pressing "CTRL + N".

A screenshot of a computer

Description automatically generated

Step 3: Right click at your circuit folder, go to "Load Library" > "Logisim Library" then find the "7-segment-display-drive.circ" file

A screenshot of a computer

Description automatically generated

Step 4: After loading the library, you need to add the component to your circuit. It's easy to see down in the list, the component:

A screenshot of a computer

Description automatically generated

Step 5: The display pins of the seven-segment are shown below:

A number with letters and numbers

Description automatically generated

Connect the driver’s inputs to the switches and the driver’s outputs to the display as follows:

A diagram of a circuit board

Description automatically generated

Step 6: Try different values of the inputs, and check what is the output in the seven-segment.

**Report**:

1. Fill up the following table:

|  |  |  |
| --- | --- | --- |
| Inputs | Outputs | |
| Binary Number | BCD Number | Seven-segment Display (try it your circuit) |
| 0000 | 0000 | 0 |
| 0001 | 0001 | 1 |
| 0010 | 0010 | 2 |
| 0011 | 0011 | 3 |
| 0100 | 0100 | 4 |
| 0101 | 0101 | 5 |
| 0110 | 0110 | 6 |
| 0111 | 0111 | 7 |
| 1000 | 1000 | 8 |
| 1001 | 1001 | 9 |
| 1010 | 0001 0000 | Not a number! |
| 1011 | 0001 0001 | 9 |
| 1100 | 0001 0010 | 9 |
| 1101 | 0001 0011 | 5 |
| 1110 | 0001 0100 | 6 |
| 1111 | 0001 0101 | 9 |

1. Briefly explain the difference between binary and BCD.

* Binary and BCD (Binary-Coded Decimal) are both number systems used in digital computing. The key difference lies in their representation of decimal digits.

Binary operates with a base of two, utilizing only 0 and 1. It is a straightforward system for representing numbers. When we reach the highest value (1 in a single digit), we increment the next digit, similar to how we increment in any base number system.

BCD, on the other hand, uses groups of 4 binary digits to represent a decimal digit. In BCD, each decimal digit (0 to 9) is represented by its 4-bit binary equivalent. For example, in BCD, the decimal digit 9 is represented as 1001. When a BCD value reaches 1001, we reset it to 0000 and add another group of 4 binary digits. BCD is designed to make it easier to perform arithmetic operations and conversions between binary and decimal, particularly in applications where precise decimal representation is necessary.

1. Convert each number shown into the other bases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Binary | Octal | Hexadecimal | Decimal | BCD |
| 1001100 | 114 | 4C | 76 | 0111 0110 |
| 100010111 | 213 | 117 | 279 | 0010 0111 1001 |
| 1011111 | 137 | 5F | 95 | 1001 0101 |
| 100000 | 40 | 20 | 32 | 0011 0010 |
| 110000 | 60 | 30 | 48 | 0100 1000 |

1. Using Logisim and the previous experiment, design a circuit that shows numbers from 0 to 99. You need to use two seven-segments.

A screenshot of a computer

Description automatically generated