UNIVERSITY OF BOHOL
College of Engineering, Technology, Architecture, and Fine Arts
DR. CECILIO PUTONG ST., TAGBILARAN CITY

Second Semester

BINARY TO OCTAL CONVERSION USING DECODER: INTRODUCTION TO HDL

In Partial Fulfillment of the Requirements for CPEP 321 course

Submitted to:

Engr. Victor John L. Anunciado , PCpE

Instructor

Submitted by:

CAPILI, ROD VINCENT O.

BSCpE-3

May 2025

TABLE OF CONTENTS

FINALS

INTRODUCTION

In digital systems and hardware design, various number systems are used to represent and manipulate data efficiently. Two commonly used systems are the binary, which is primarily called base two, and the octal, as base eight. Binary numbers, consisting only of two digits, which are 1's and 0's, are fundamental in digital circuits and are directly understood by hardware. Octal numbers, using digits from 0 to 7, offer a more compact representation and are often used for simplifying binary data, especially in programming and hardware description language.

The purpose of converting Binary numbers to Octal is that it can reduce the length of binary strings, making them easier to read and interpret in the code and hardware documentation. Each octal representation corresponds to exactly three binary digits, which are called 'bits', allowing it to have a more straightforward in grouping and converting.

There are methods used in converting binary to octal to be able to how data is collected, solved, and manipulated to get the desired output. Let's say, for example, that you want to convert 0111_2 in Octal. It only needs two methods to do it, first, is to read it from the rightmost digit to the leftmost digit (Least Significant Bit to Most Significant Bit). Then putting indicators to be able to get the result of the conversion, and then summing it up.

Indicator	8	4	2	1
Binary Input	**0**	**1**	**1**	**1**
Octal Result	**4 + 2 + 1 = 7**		**7 (8)**	

Binary to octal conversion can also be achieved by first translating the binary number into its decimal form, followed by successive divisions by 8 to extract the octal digits. Although this approach requires more computational steps and is less optimal for direct hardware implementation, it holds significant value in Hardware Description Language (HDL) design due to its clarity and educational benefits. Writing this conversion in HDL allows designers and learners to gain a deeper comprehension of numerical systems, arithmetic processes, and data handling within digital hardware, which are essential for creating effective digital circuits. Additionally, this method aids in debugging, verification, and building modular components for complex hardware systems. Mastering such conversion techniques equips HDL developers with greater versatility and precision in designing digital systems that handle diverse data formats, making this approach a worthwhile skill despite its lower efficiency compared to direct binary grouping methods.

BINARY TO OCTAL CONVERSION

PROBLEM REQUIREMENTS

The following requirements are for this project to obtain a successful application and design of the project:

- To produce a Truth table for Binary to Octal conversion
- To solve and simplify the project with the use of Karnaugh maps and Boolean algebra laws.
- To apply a Logic Circuit diagram and design the circuit, and correct the logic output.
- To create the Icarus Verilog Design in Notepad++
- To design modules that are suitable for the design and its truth table.
- To produce the correct output of the code based on the truth table using the gtkwave command.

SCOPE AND LIMITATIONS

The binary to octal conversion via the intermediate decimal method provides a clear, step-by-step approach valuable for education, debugging, and modular design in HDLs. However, it is less efficient and slower than direct grouping, requires more hardware resources, and increases code complexity, making it unsuitable for high-speed or resource-limited designs and large-scale production where performance and simplicity are essential.

ANALYSIS

The input to the binary to octal conversion module is a fixed-width binary number represented as a bit vector in HDL, which is processed to produce the output—a compact octal number where each digit corresponds to three binary bits. This conversion simplifies data handling by providing a more readable and concise format useful for debugging, display, memory initialization, or communication interfaces. The module is typically used within larger digital systems or testbenches to improve data interpretation and system clarity, making it an essential component for efficient hardware design and verification.

DESIGN AND IMPLEMENTATION

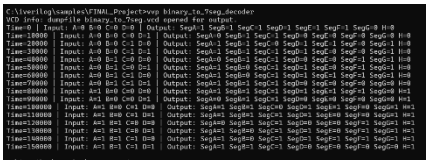

Figure 1.1: T-table Display

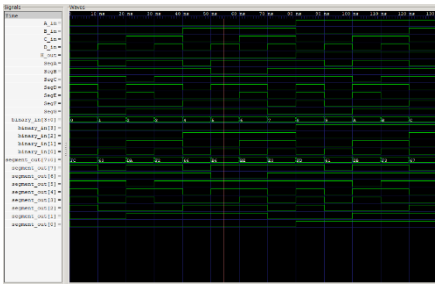

Figure 1.2: gtkwave Signal

TESTING AND DEBUGGING

```
module binary_to_7seg_decoder ( // Start of the module definition for the 7-segment decoder
    input [3:0] binary_in,       // Declare a 4-bit input 'binary_in' for the binary number
                                 // binary_in[3] corresponds to 'A' (Most Significant Bit) fro
                                 // binary_in[2] corresponds to 'B'
                                 // binary_in[1] corresponds to 'C'
                                 // binary_in[0] corresponds to 'D' (Least Significant Bit)
                                 // The order is important to match your truth table's A, B, C

    output reg [7:1] segment_out // Declare an 8-bit output 'segment_out' as a 'reg' type (for
                                 // This output controls the segments of the 7-segment display
                                 // The bit assignments are:
                                 // segment_out[7] = A_segment (controls segment 'a' of the di
                                 // segment_out[6] = B_segment (controls segment 'b')
                                 // segment_out[5] = C_segment (controls segment 'c')
                                 // segment_out[4] = D_segment (controls segment 'd')
                                 // segment_out[3] = E_segment (controls segment 'e')
                                 // segment_out[2] = F_segment (controls segment 'f')
                                 // segment_out[1] = G_segment (controls segment 'g')
                                 // segment_out[0] = H_LED (controls the decimal point or an e
);

    // Explicit wire declarations for individual input bits for better GTKWave visibility
    wire A_in = binary_in[3]; // Create a named wire 'A_in' for the MSB of binary_in
    wire B_in = binary_in[2]; // Create a named wire 'B_in' for the next bit
    wire C_in = binary_in[1]; // Create a named wire 'C_in'
    wire D_in = binary_in[0]; // Create a named wire 'D_in' for the LSB of binary_in
```

Figure 1.3: Icarus iVerilog Code

FUTURE DEVELOPMENT

Future developments could include support for fractional binary numbers and bidirectional conversion between binary and octal to increase flexibility. Optimizing the design for faster performance and lower hardware resource usage would enhance suitability for real-time applications. Additionally, expanding the module to handle other number systems, like hexadecimal, would improve its versatility in various hardware interfaces.

REFERENCES

https://byjus.com/maths/binary-to-octal-conversion/

STUDENT INFORMATION

CURRICULUM VITAE

ROD VINCENT O. CAPILI

PUROK 6 - BAYANIHAN, TAGUIHON,

BACLAYON, BOHOL, PHILIPPINES 6301

https://bit.ly/4kq6Efq

"If it is to be, it is up to me."

PROFILE

Date of Birth	:	May 15, 2002
Place of Birth	:	Tagbilaran City, Bohol
Nationality	:	Filipino
Status	:	Single
Gender	:	Male
Contact Number	:	09106724856
Email	:	rvocapili@universityofbohol.edu.ph

EDUCATIONAL BACKGROUND:

Primary	:	Immaculata High School Baclayon, Bohol 2017 - 2018
Tertiary	:	University of Bohol Tagbilaran City, Bohol 2019 - 2020
Tertiary	:	University of Bohol Tagbilaran City, Bohol Bachelor of Science in Computer Engineering 2024 – present

PROJECTS:

Final Project	:	BINARY TO OCTAL CONVERSION
Project Link	:	**http://bit.ly/3H97K0T**