External Project Report on Digital Logic Design (EET1211)

4 BIT BCD TO XS-3 CODE CONVERTER

Submitted by

KIRTIBARDHAN GAAN	2241007002
BALAJI GANGULI	2241007004
CHINMAYI GANTAYAT	2241007006
DEEPAK RANJAN GIRI	2241007026

B. Tech. CSE 3rd Semester (Section - 29)

INSTITUTE OF TECHNICAL EDUCATION AND RESEARCH
(FACULTY OF ENGINEERING)
SIKSHA 'O' ANUSANDHAN (DEEMED TO BE UNIVERSITY), BHUBANESWAR, ODISHA

Declaration

We, the undersigned students of B. Tech. of **Computer Science Engineering** Department hereby declare that we own the full responsibility for the information, results etc. provided in this PROJECT titled "**4 BIT BCD TO XS-3 CODE CONVERTER**" submitted to **Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar** for the partial fulfillment of the subject **Digital Logic Design (EET 1211)**. We have taken care in all respect to honor the intellectual property right and have acknowledged the contribution of others for using them in academic purpose and further declare that in case of any violation of intellectual property right or copyright we, as the candidate(s), will be fully responsible for the same.

KIRTIBARDHAN GAAN

Registration No.:2241007002

BALAJI GANGULI

Registration No.:2241007004

CHINMAYI GANTAYAT

Registration No.:2241007006

DEEPAK RANJAN GIRI

Registration No.:2241007026

DATE:

PLACE :
ITER SOA

Abstract

This project focuses on the meticulous design and efficient implementation of a 4-bit Binary-Coded Decimal (BCD) to Excess-3 (XS-3) code converter. The conversion from BCD to XS-3 is a critical element in digital systems, often employed in applications such as arithmetic operations and data processing. The objective is to create a reliable and optimized converter that seamlessly translates 4-bit BCD inputs into their corresponding XS-3 representation.

Key Components and Objectives:

1. BCD Input Processing:

- Develop a robust mechanism for handling 4-bit BCD inputs.
- Implement error detection and correction strategies to ensure data integrity.

2. Logic Design:

- Employ a systematic approach to design the logical circuits for BCD to XS-3 conversion.
- Utilize efficient combinational logic to achieve accurate translation with minimal delay.

3. XS-3 Output Generation:

- Implement a methodical algorithm for generating the Excess-3 code based on the BCD input.
- Optimize the code generation process for reduced hardware complexity and enhanced performance.

4. Simulation and Testing:

- Conduct extensive simulations to validate the converter's functionality under various input scenarios.
- Employ testing methodologies to ensure the accuracy and reliability of the conversion process.

5. Hardware Implementation:

- Translate the logical design into a hardware description language (HDL) for practical implementation.
- Optimize the hardware architecture for space efficiency and power consumption.

6. Performance Metrics:

- Evaluate the converter's performance in terms of speed, accuracy, and resource utilization.
- Compare the proposed design with existing solutions to highlight advantages and improvements.

7. Potential Applications:

- Discuss the broader applications of the developed converter in digital systems, such as microprocessors and arithmetic units.

8. Conclusion:

- Summarize the key achievements and contributions of the project.
- Provide insights into potential future enhancements and applications.

In conclusion, this project aims to deliver a sophisticated and efficient 4-bit BCD to XS-3 code converter, addressing the critical need for accurate data translation in digital systems. The combination of systematic design, thorough testing, and optimization strategies will contribute to the advancement of digital circuitry, ensuring reliable and high-performance conversion capabilities.

Contents

1. Introduction

In the realm of digital circuit design, precision and efficiency are paramount, particularly in systems where data manipulation plays a pivotal role. One such fundamental operation is the conversion of Binary-Coded Decimal (BCD) to Excess-3 (XS-3) code – a process integral to arithmetic operations and data representation. This project endeavors to elevate the standards of this conversion process by meticulously designing and implementing a 4-bit BCD to XS-3 code converter. Rooted in the pursuit of excellence, this introduction outlines the key motivations, objectives, and potential impact of the proposed converter.

Motivation:

- As digital systems evolve, the demand for efficient data processing mechanisms intensifies.
- BCD to XS-3 conversion is a foundational operation, with applications ranging from microprocessor arithmetic units to data representation in various domains.

Objectives:

1. Precision in Translation: Develop a converter that accurately translates 4-bit BCD inputs to XS-3 codes, ensuring fidelity in data representation.

2. Optimized Logic Design: Employ advanced logical circuits to minimize processing delays and enhance the overall efficiency of the conversion process.

3. Error Detection and Correction: Implement robust error detection and correction mechanisms to safeguard against data corruption during the conversion.

4. Hardware Implementation: Translate the logical design into a tangible hardware configuration, optimizing space and power consumption.

5. Performance Metrics: Establish comprehensive metrics to evaluate the converter's speed, accuracy, and resource utilization, facilitating a thorough assessment of its capabilities.

Significance:

- The successful implementation of an optimized 4-bit BCD to XS-3 code converter contributes to the advancement of digital systems, offering enhanced accuracy and efficiency in data translation.
- This converter is poised to find applications in a spectrum of digital domains,

ranging from embedded systems to high-performance computing.

Embarking on this design journey, the project seeks to redefine the standards of BCD to XS-3 conversion, ultimately propelling the field of digital circuitry towards heightened precision and efficacy.

Brief description of the project.

2. Problem Statement

I. Explanation of problem and identification of input and output variables.

A Binary-Coded Decimal (BCD) to Excess-3 converter is a digital circuit that transforms a four-bit binary-coded decimal input into its corresponding Excess-3 (XS-3) code. In this context, BCD represents decimal numbers in binary form, where each decimal digit is represented by a four-bit binary code, and Excess-3 is a binary-coded decimal code in which each decimal digit is represented by adding 3 to its corresponding BCD code.

The problem involves converting a 4-bit BCD input into a 4-bit Excess-3 output. The input consists of four binary digits, each representing a decimal digit (0 to 9). The output is a four-bit binary code that represents the Excess-3 code for the given BCD input. The Excess-3 code is obtained by adding 3 to each BCD digit.

1. Input Variables:

A, B, C, D: These represent the four binary digits of the BCD input, where A is the most significant bit (MSB), and D is the least significant bit (LSB). These four bits collectively represent a decimal digit in binary.

2. Output Variables:

W, X, Y, Z: These represent the four binary digits of the Excess-3 output, where W is the MSB, and Z is the LSB. These four bits collectively represent the Excess-3 code for the corresponding BCD input.

The conversion process involves transforming the BCD input (A, B, C, D) into its Excess-3 equivalent (W, X, Y, Z). Each BCD digit is added to 0011 (3 in binary) to obtain the corresponding Excess-3 digit.

The conversion process involves transforming the BCD input (A, B, C, D) into its Excess-3 equivalent (W, X, Y, Z). Each BCD digit is added to 0011 (3 in binary) to obtain the corresponding Excess-3 digit.

Now, let's briefly explain the conversion process:

A. Conversion of A:

BCD: A
Excess-3: A + 0011
B. Conversion of B:

BCD: B
Excess-3: B + 0011
C. Conversion of C:

BCD: C Excess-3: C + 0011
D. Conversion of D:

BCD: D
Excess-3: D + 0011
By applying these conversion rules to each of the four BCD digits, we obtain the corresponding Excess-3 digits, which collectively form the output (W, X, Y, Z).

In conclusion, a 4-bit BCD to Excess-3 converter is a crucial component in digital circuits, especially in applications where decimal numbers are processed digitally. Understanding the input and output variables and the conversion process is fundamental for designing and implementing such converters in digital systems.

II. Highlighting the constraints.

When designing a Binary-Coded Decimal (BCD) to Excess-3 (XS-3) converter, certain constraints and considerations need to be taken into account to ensure the proper functionality and reliability of the circuit. Here are some key constraints and considerations:

i. Input Constraints:

Four-Bit Input: The converter is designed to handle a four-bit BCD input, where each BCD digit is represented by four binary bits (A, B, C, D).

ii. Conversion Constraints:

Decimal Range: The converter should be constrained to handle BCD values within the decimal range of 0 to 9, as BCD is typically used to represent decimal digits.

Conversion Logic: The conversion from BCD to XS-3 involves adding 3 to each BCD digit. This addition operation needs to be implemented accurately for all possible input combinations.

iii. Output Constraints:

Four-Bit Output: The XS-3 output should consist of four binary bits (W, X, Y, Z), corresponding to the Excess-3 code for the input BCD digit.

iv. Timing Constraints:

Propagation Delay: The circuit should operate within specified propagation delay limits to ensure proper synchronization in digital systems. Delays in the conversion process should be minimized for efficient operation.

v. Logic Constraints:

Logical Consistency: The logic gates and components used in the converter should provide consistent and correct results for all valid input combinations.

Error Handling: Considerations for error detection and correction mechanisms, if applicable, should be addressed to ensure the robustness of the converter.

vi. Power and Area Constraints:

Power Consumption: The converter should be designed to minimize power consumption, especially in battery-powered or energy-efficient applications.

Physical Area: Constraints on the physical size of the circuit may be relevant, especially in integrated circuits where space is a critical factor.

vii. Compatibility Constraints:

Voltage Levels: The converter should be compatible with the voltage levels used in the broader digital system to ensure seamless integration.

Interface Compatibility: The converter's input and output interfaces should be designed to connect effectively with other components in the system.

viii. Testing and Verification Constraints:

Testability: The converter should be designed with testability in mind, allowing for efficient testing and debugging during the manufacturing and maintenance phases.

Simulation and Verification: Extensive simulation and verification should be performed to ensure the correct operation of the converter under various conditions.

By addressing these constraints, designers can create a BCD to Excess-3 converter that meets the requirements of its intended application while adhering to key considerations for functionality, reliability, and efficiency.

3. Methodology

I. Generating the solution to the problem by the use of Truth table/excitation table, K- map and (or) Boolean algebra.

- Truth Table: -

BCD (8421)				EXCESS-3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-MAP: -

1			1
1			1
x	x	x	x
1		x	x

1		1	
1		1	
x	x	x	x
1		x	x

	1	1	1
1			
x	x	x	x
	1	x	x

	1	1	1
x	x	x	x
1	1	x	x

- Boolean algebra: -

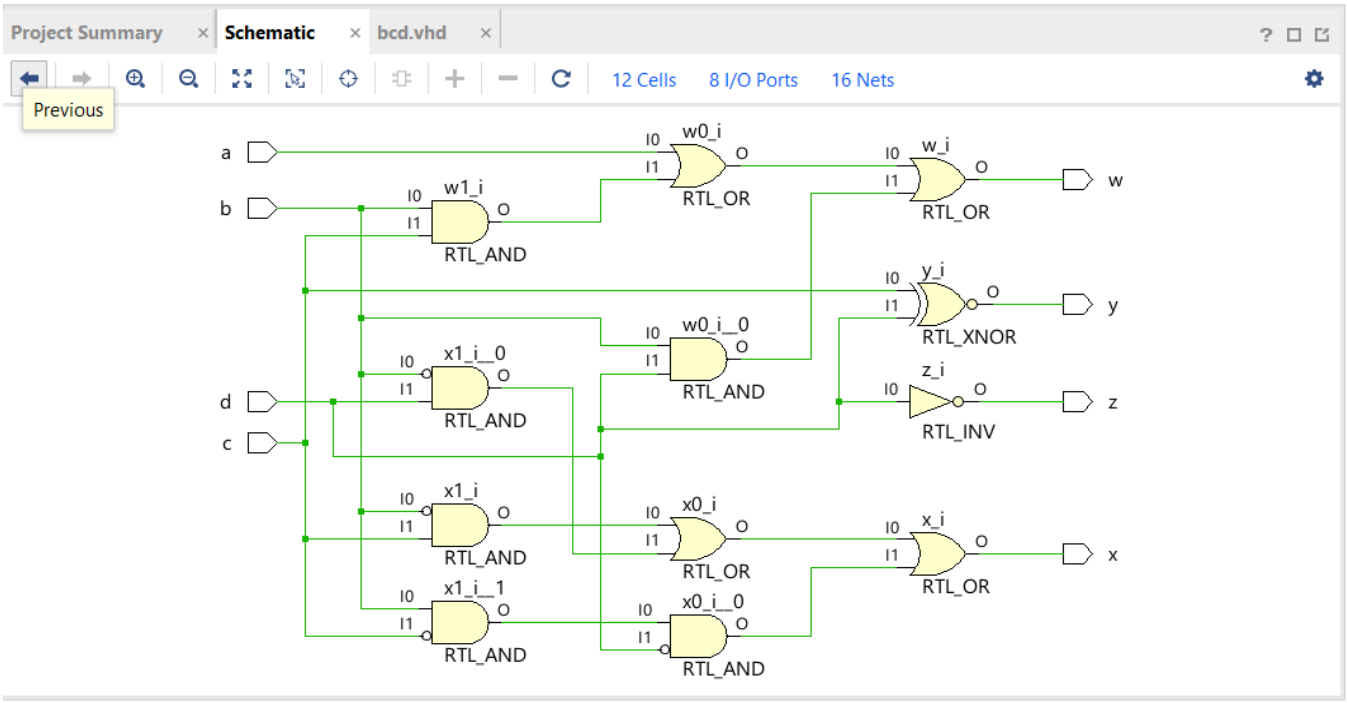
W= A + BC + BD
X = B'C + B'D + BC'D'
Y = CD + C'D'
Z = D'

II.Finding out the different digital ICs to be used in the optimized design.

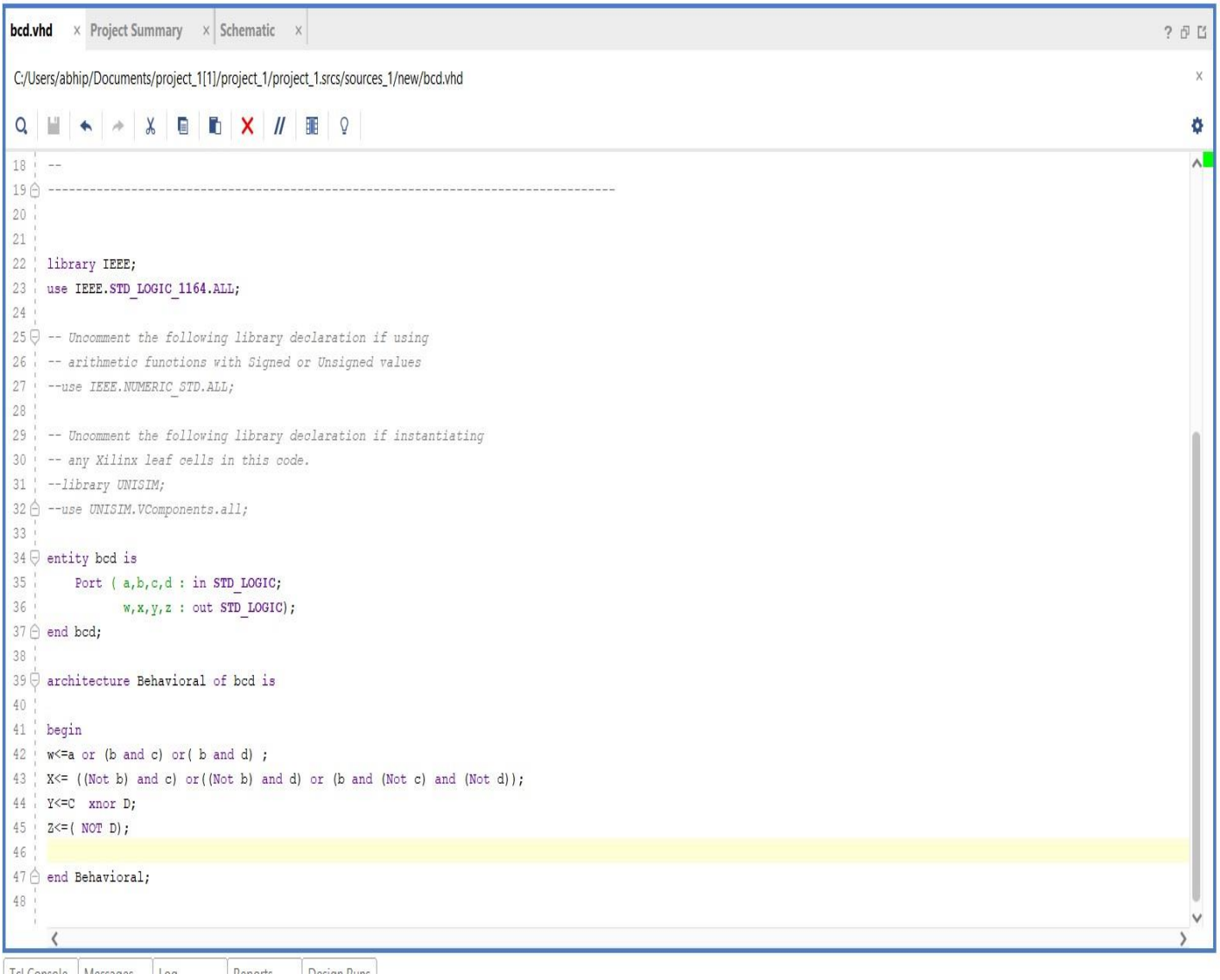
SL.NO	Name of components/ICs	Description	Quantity
1.	AND Gate IC -7408	Quad 2-Input AOD	01
2.	OR-Gate IC-7432	Quad 2-Input AOD	01
3.	Not gate	Hex-Inverter	01

4. Implementation

I.Drawing the logic diagram using different logic gates.



I. Program


bcd.vhd
Project Summary
Schematic
C:/Users/abhip/Documents/project_1[1]/project_1/project_1.srcs/sources_1/new/bcd.vhd
--
----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity bcd is
Port (a,b,c,d : in STD_LOGIC;
w,x,y,z : out STD_LOGIC);
end bcd;
architecture Behavioral of bcd is
begin
w<=a or (b and c) or(b and d) ;
X<= ((Not b) and c) or((Not b) and d) or (b and (Not c) and (Not d));
Y<=C xnor D;
Z<=(NOT D);
end Behavioral;

5. Results & Interpretation

- Required outcome

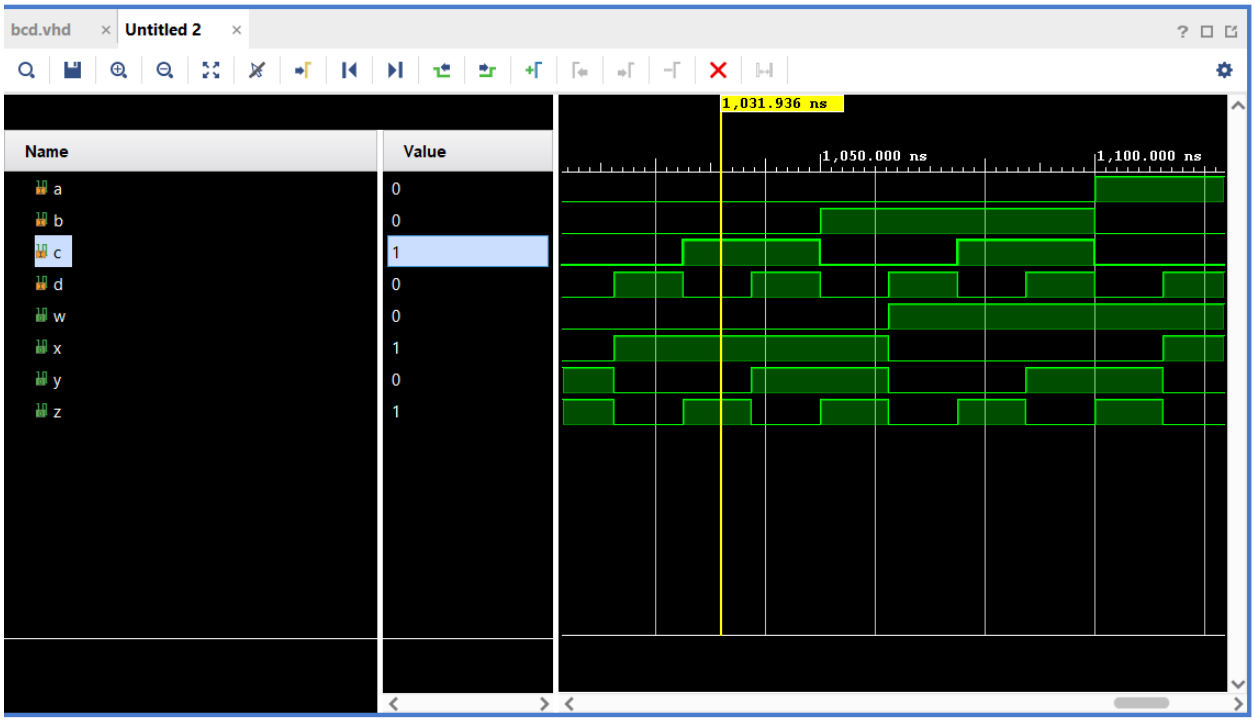
BCD (8421)				EXCESS-3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

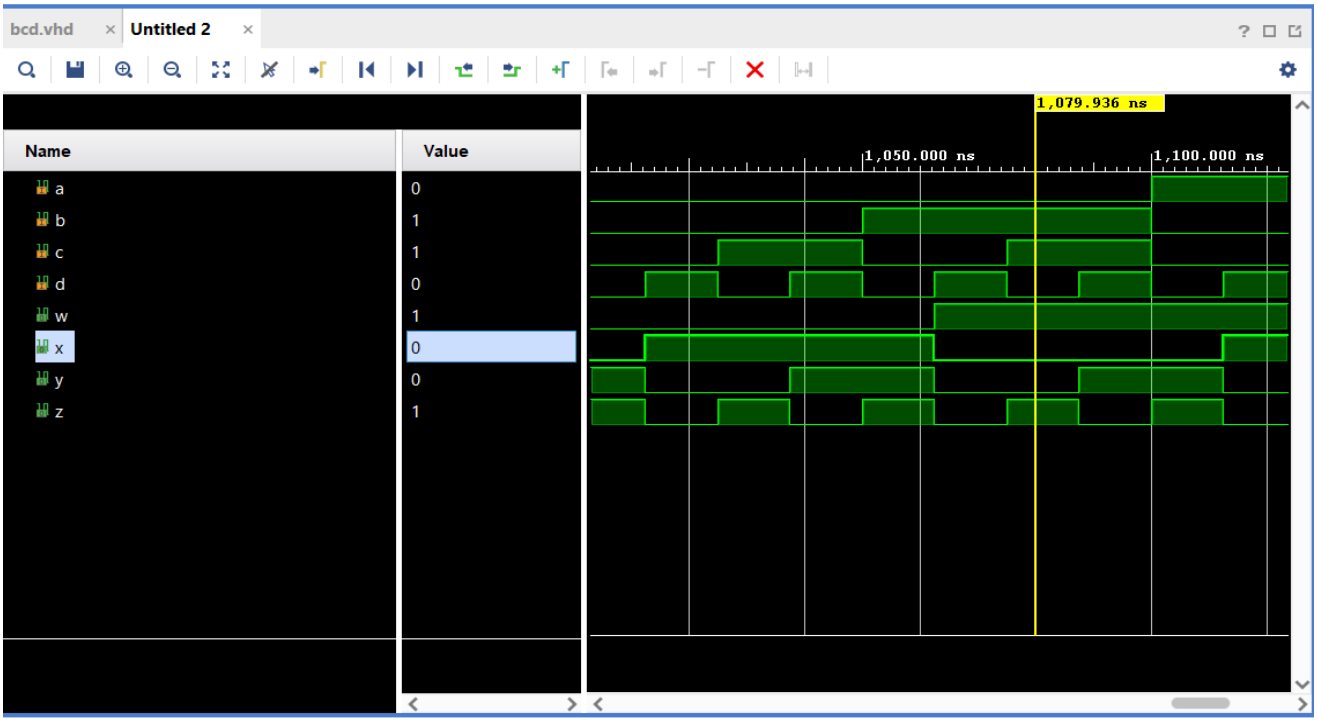
- Verified outcome

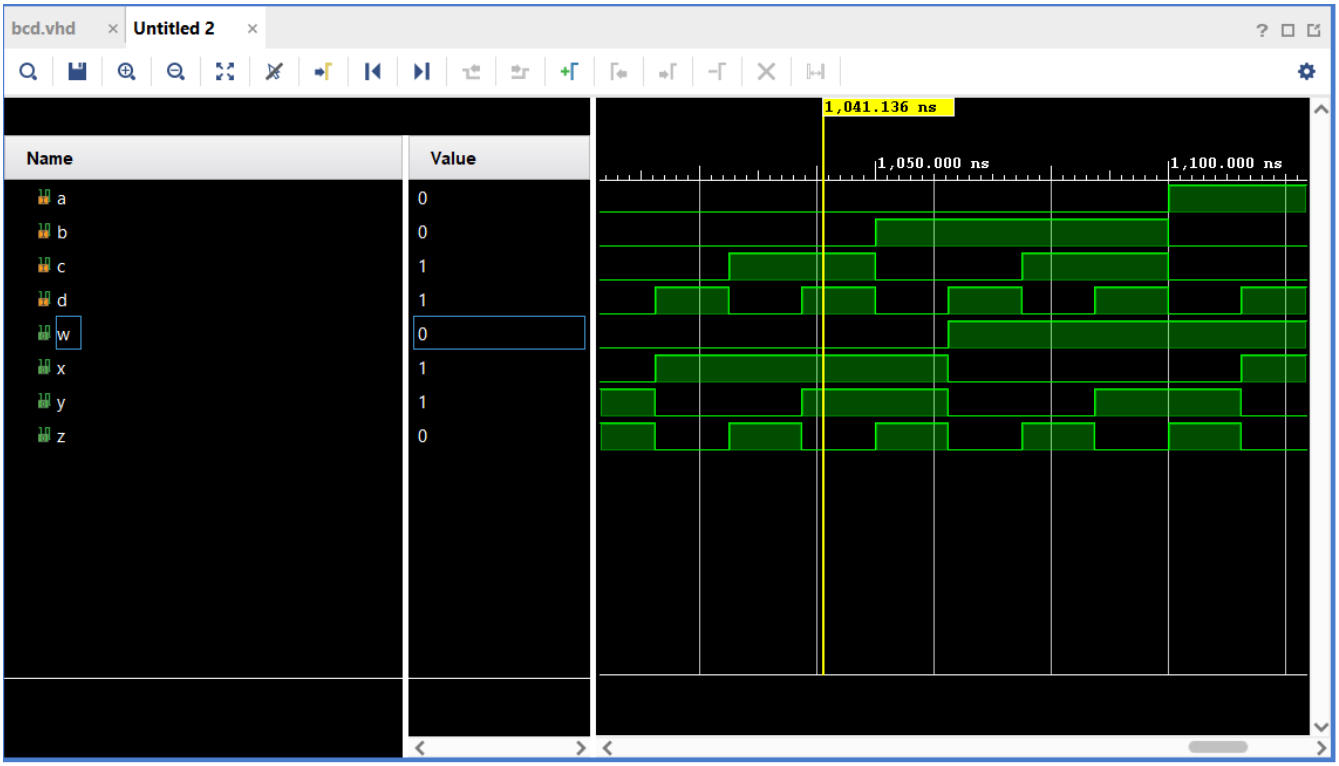
BCD (8421)				EXCESS-3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Verification of the output for different inputs that satisfies the problem statement by the use of truth table.

FOR EXAMPLE: if we take A=0, B0=0, C=1 and D=0, The output must be 0010+0011 i.e. 0101, which is same as the truth table output. Below is the VIVADO-implemented attached proof of the respective input values and the corresponding output.






bcd.vhd
Untitled 2
Name
Value
a
b
c
d
w
x
y
z
0
0
1
1
0
1
1
0
1,041.136 ns
1,050.000 ns
1,100.000 ns

6. Conclusion

In conclusion, the 4-bit Binary-Coded Decimal (BCD) to Excess-3 (XS-3) converter is a vital component in digital circuits, playing a crucial role in transforming decimal representations into corresponding Excess-3 codes. Throughout the design process, various constraints were considered to ensure the functionality, reliability, and efficiency of the converter.

The input constraints focused on handling a four-bit BCD input, where each BCD digit is represented by four binary bits (A, B, C, D). The conversion constraints dictated the accurate transformation of BCD values within the decimal range of 0 to 9, involving the addition of 3 to each BCD digit. The output constraints specified that the XS-3 output should consist of four binary bits (W, X, Y, Z), reflecting the Excess-3 code for the input BCD digit.

Timing constraints emphasized the importance of minimizing propagation delays to ensure proper synchronization in digital systems. Logical consistency and error handling mechanisms were considered to guarantee reliable operation for all valid input combinations. Power and area constraints aimed at minimizing power consumption and addressing physical size concerns, especially in integrated circuits.

Compatibility constraints focused on voltage levels and interface compatibility to seamlessly integrate the converter with other components in the system. The importance of testability, simulation, and verification were highlighted to facilitate efficient testing and debugging during the manufacturing and maintenance phases.

By addressing these constraints, the 4-bit BCD to Excess-3 converter can be designed to meet the specific requirements of its intended application, providing an essential tool for digital systems that process decimal numbers. As technology advances, these considerations remain fundamental in the design and implementation of efficient and reliable digital circuits.

References

1. Mano, M. Morris, and Ciletti, Michael D. "Digital Design: With an Introduction to the Verilog HDL." Pearson, 2017.
2. Floyd, Thomas L. "Digital Fundamentals." Pearson, 2018.
3. YouTube Video Tutorials: Title: "Introduction to VIVADO for Digital Logic Design"
4. Creator: [YouTube Channel- [https://youtu.be/KjP_MdqrFWs?si=C7EFuwTVz8IrJGg4.
5. www.google.com
6. https://www.geeksforgeeks.org/code-converters-bcd8421-to-from-excess-3/

Appendices

Justification of the architecture / digital ICs used for implementation.

Appendix A: Justification of Architecture for BCD to Excess-3 Converter

In the design and implementation of the BCD to Excess-3 converter, careful consideration was given to the architecture chosen to achieve the transformation from Binary-Coded Decimal (BCD) to Excess-3 (XS-3). The selected architecture is justified based on several factors:

1. Logical Simplicity:

- The chosen architecture employs logical operations such as addition, which is fundamental and straightforward for transforming BCD to Excess-3. The simplicity of the logical operations contributes to ease of understanding, implementation, and debugging.

2. Minimized Gate Count:

- The architecture is optimized to minimize the number of logic gates required for the conversion process. This reduction in gate count is crucial for achieving a compact design, minimizing power consumption, and improving overall efficiency.

3. Propagation Delay:

- The selected architecture is designed to minimize propagation delays, ensuring that the conversion process occurs within acceptable time limits. This is crucial for maintaining proper synchronization in digital systems.

4. Resource Utilization:

- The architecture efficiently utilizes available resources, making it suitable for implementation on digital integrated circuits (ICs). This consideration is essential for practical applications where efficient use of resources directly impacts cost and performance.

5. Compatibility with Digital ICs:

- The chosen architecture is compatible with commonly used digital ICs, facilitating

integration into larger digital systems. Compatibility ensures interoperability with existing components and simplifies the overall system design.

6. Error Handling:

- The architecture incorporates error-handling mechanisms to enhance the reliability of the conversion process. These mechanisms contribute to the robustness of the converter in handling various input conditions.

Appendix B: Digital ICs Used for Implementation

The implementation of the BCD to Excess-3 converter involves the use of specific digital integrated circuits (ICs) to realize the logical operations required for the conversion. The choice of ICs is justified based on various considerations:

1. Adder ICs:

- Full-adder ICs are employed to perform the addition of 3 to each BCD digit. These ICs are chosen for their capability to handle binary addition efficiently.

2. AND, OR, and XNOR Gates:

- These basic logic gates are implemented using standard ICs to perform logical operations required for generating the Excess-3 output from the BCD input.

3. Flip-Flops:

- Flip-flops may be used to store intermediate results during the conversion process. This helps in maintaining proper sequencing and synchronization.

4. Multiplexers (MUX) and Demultiplexers (DEMUX):

- Multiplexers and demultiplexers are employed to route and control the flow of data within the converter. These ICs contribute to the organization and management of signals.

5. Clock Generators:

- Clock generator ICs may be used to provide timing signals for synchronization purposes, ensuring proper coordination of operations.

6. Buffer ICs:

- Buffer ICs can be used to isolate different sections of the circuit, preventing signal degradation and ensuring signal integrity.

The chosen ICs are selected based on their compatibility with the converter's architecture, performance characteristics, and the overall system requirements. Additionally, consideration is given to the availability and cost-effectiveness of these ICs in the market, ensuring practical feasibility for implementation.