VARDHAMAN
COLLEGE OF ENGINEERING

A Course End Project Report on

Design and Implementation of 8 to 3 Encoder amplifier

A8428- CMOS VLSI Design

Submitted by

B Praveen Kumar 22881A04D4

Course Facilitator

Dr. B.Srikanth
Assistant Professor

Dr. Sangeeta Singh Assistant
Professor

Department of Electronics and Communication Engineering Vardhaman College

of Engineering, Hyderabad

April 2025

1 Abstract

This report presents the design and simulation of an 8 to 3 priority encoder. Priority encoders are fundamental combinatorial logic circuits essential for applications requiring the identifica- tion of the highest-priority active signal among multiple inputs, such as interrupt controllers, keyboard interfaces, and data acquisition systems. An 8 to 3 priority encoder receives 8 in- put signals and outputs a 3-bit binary code corresponding to the highest-priority active input, along with a validity signal indicating whether any input is active. The design was implemented using the Cadence Virtuoso Design Platform, a leading Electronic Design Automation (EDA) tool suite widely used in the semiconductor industry. The implementation involved creating a schematic representation of the encoder logic using standard digital library cells (or transistor- level design, specify if applicable). Functional verification was performed through extensive simulation using the Cadence simulation environment (e.g., Spectre or Xcelium), applying var- ious input combinations, including simultaneous assertion of multiple inputs, to confirm the correct priority behavior. The report details the design methodology, including the truth table and logic expressions, the schematic capture process in Virtuoso, the testbench setup for simu- lation, and the analysis of simulation waveforms to validate the design's functionality according to the priority encoding rules.

Keywords: Priority Encoder, 8 to 3 Encoder, Cadence Virtuoso, EDA Tool, Schematic Capture, Circuit Simulation, Spectre, Xcelium, Digital Logic Design, Combinatorial Logic.

2 Introduction

Digital systems often require circuits capable of efficiently managing and compressing large amounts of input data into a simplified and compact format. Among the fundamental building blocks for such operations are encoders, which serve the critical function of converting informa- tion from multiple input lines into a smaller set of output lines, typically represented in binary form. This process of data reduction is not only essential for optimizing hardware usage but also for streamlining control logic in complex systems such as processors, communication interfaces, and memory devices.

Encoders work by detecting which input line is active and generating a binary output corresponding to that line's position. However, standard or binary encoders face limitations when multiple inputs are active simultaneously. In such cases, the output becomes undefined or erroneous because the encoder cannot determine which input to prioritize. This ambiguity severely restricts the application of basic encoders in real-world digital systems where multiple simultaneous signals are common.

To overcome this challenge, priority encoders are introduced. These enhanced circuits are designed to handle multiple active inputs by assigning a predefined hierarchy of importance or priority to each input. When more than one input is asserted, the encoder outputs the binary code corresponding to the input with the highest priority. For example, in an 8-to-3 priority encoder, eight input lines (D_0 through D_7) are encoded into a 3-bit binary output, where D_7 typically has the highest priority and D_0 the lowest. An additional signal, commonly referred to as Valid or Group Select (GS), indicates whether any input is currently active, enhancing system-level control and reliability.

Priority encoders are widely used in interrupt handling systems, keyboard encoders, communication protocols, and other areas where arbitration among multiple signals is necessary. Their ability to resolve input contention makes them indispensable in many digital applications. This project focuses on the design and simulation of an **8-to-3 priority encoder** using the **Cadence Virtuoso Design Platform**. Cadence Virtuoso is an industry-leading Electronic Design Automation (EDA) tool suite widely used for the design and verification of integrated cir- cuits (ICs). It supports a complete digital and analog/mixed-signal design flow, from schematic capture and behavioral simulation to layout and physical verification. Implementing a digital logic circuit such as a priority encoder in this professional environment bridges the gap between academic learning and industry practices. The process not only reinforces theoretical under- standing of digital logic but also provides practical experience in schematic design, simulation, and verification workflows used in real-world VLSI (Very-Large-Scale Integration) design.

3 Literature Review

Digital logic design is a foundational aspect of computer engineering, providing the building blocks for digital systems. Numerous authoritative sources have explored this field, each con- tributing to a comprehensive understanding of logic gates, combinational and sequential circuits, and implementation using hardware description languages like Verilog.Mano and Ciletti [1] de- liver a classical approach to digital logic design, beginning with the basics of Boolean algebra and progressing to complex topics such as flip-flops and memory units. Their work is widely rec- ognized for its clarity and pedagogical approach, making it a standard text in the field.Similarly, Brown and Vranesic [2] focus on integrating Verilog into the study of logic design. Their book provides a balanced mix of theory and practical examples, allowing students to simulate and implement circuits effectively. This dual approach is valuable in today's digital circuit design landscape.

Hayes [3] offers a thorough theoretical foundation, with special emphasis on number systems and logic functions. His detailed mathematical treatment aids in building a strong conceptual

understanding, essential for advanced design and analysis.Tocci et al. [4] expand on this foun- dation by including numerous real-world applications. Their discussion of programmable logic devices and digital system integration supports practical design needs, especially for engineer- ing projects.Katz and Borriello [5] approach the subject with a modern perspective, integrating CAD tools and contemporary design challenges. Their text is particularly useful for under- standing how logic design interacts with current hardware technologies.

In addition to textbooks, online tutorials and educational websites offer practical insights into specific topics such as encoders. For instance, GeeksforGeeks [6] provides a concise expla- nation of encoders with truth tables and implementation details, making it a useful supplement for students.TutorialsPoint [7] offers another accessible guide, presenting the classification of encoders and their logic diagrams, beneficial for quick reference and revision.Electronics Tu- torials [8] delves into the differences between normal and priority encoders, with illustrative circuit diagrams. This source bridges theoretical understanding and practical design consider- ations.ElProCus [9] explores the applications of priority encoders in communication systems, adding a real-world context to the theoretical principles.Lastly, GitHub repositories such as the one by Abhinandan [10] provide working Verilog code for encoder circuits, facilitating hands-on experimentation and validation of concepts discussed in academic resources.

4 Objectives

The primary objectives of this project are:

- To understand the functional requirements and priority logic of an 8 to 3 priority encoder.
- To define the truth table for an 8 to 3 priority encoder, including a validity output.
- To translate the priority logic into a circuit design using standard digital library cells (or transistor-level logic, specify if applicable) within the Cadence Virtuoso schematic editor.
- To set up a testbench environment in Cadence Virtuoso for simulating the designed priority encoder.
- To perform functional simulation (e.g., transient analysis) using a Cadence simulator (Spectre or Xcelium) to verify the circuit's behavior.
- To analyze the simulation waveforms obtained from Cadence to confirm correct operation under various input conditions, including multiple simultaneous inputs, according to the defined priority.

5 Methodology

The design and simulation of the 8-to-3 priority encoder were executed in accordance with a standard integrated circuit (IC) design flow, using the Cadence Virtuoso design environment. The process began with the specification of the logical functionality of the encoder, identifying the input-output behavior and priority conditions to be met.

Following the logic definition, the design was captured using the schematic editor in Vir- tuoso, where logic gates and interconnections were drawn to represent the encoder's behavior accurately. This schematic served as the structural basis for subsequent analysis and verifica- tion.

Next, a simulation testbench was constructed to validate the encoder's functionality under various input conditions. This testbench included stimulus generators to apply all possible input combinations and monitors to observe the encoder's output and ensure correct prioritization.

5.1 Block Diagram

The 8 to 3 priority encoder has 8 input lines (D_0 to D_7), 3 output lines for the binary code (Y_2, Y_1, Y_0), and a validity output (GS). The inputs are prioritized from D_7 (highest) to D_0 (lowest). The outputs Y_2, Y_1, Y_0 represent the 3-bit binary code of the highest-priority active input, and GS is active high if any input is active.

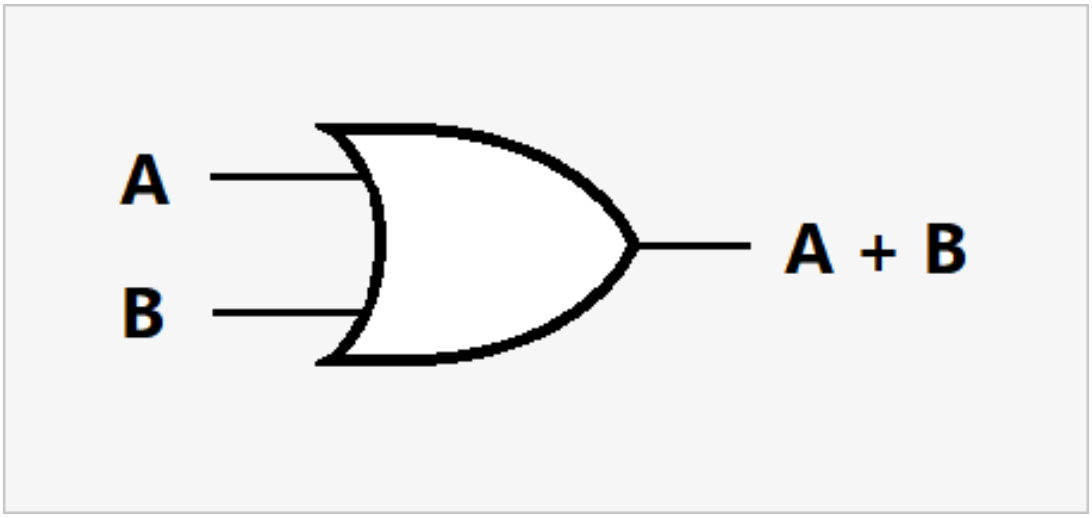

Figure 1: or-gate schematic diagram

5.2 Logic Design

The logic function of the priority encoder is defined by its truth table and corresponding Boolean expressions based on the priority rule.

5.2.1 Truth Table

The truth table for an 8 to 3 priority encoder with inputs D_0 to D_7 (D_7 highest priority) and outputs Y_2, Y_1, Y_0, and GS (active high if any input is high) is presented in Table 1. 'X' denotes a don't care condition.

Table 1: Truth Table for 8 to 3 Priority Encoder

Inputs								**Outputs**			**Output**
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_2	Y_1	Y_0	GS
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	X	0	0	1	1
0	0	0	0	0	1	X	X	0	1	0	1
0	0	0	0	1	X	X	X	0	1	1	1
0	0	0	1	X	X	X	X	1	0	0	1
0	0	1	X	X	X	X	X	1	0	1	1
0	1	X	X	X	X	X	X	1	1	0	1
1	X	X	X	X	X	X	X	1	1	1	1

5.2.2 Logic Expressions

The Boolean expressions for the outputs can be derived from the truth table. For instance, using simplified logic reflecting the priority:

$$Y_2 = D_4D_5D_6D_7 + \overline{D_5D_6D_7} + \overline{D_6D_7} + D_7 = D_4D_5\overline{-_7} + \overline{D_5D_6}\overline{-_7} + \overline{D_6D_7} + D_7$$

$$Y_1 = \overline{D_2D_3}\overline{-_7} + D_3\overline{D_4}\overline{-_7} + D_6\overline{D_7} + \overline{D_7}$$

$$Y_0 = D_0D_1\overline{-_7} + \overline{D_2D_3}\overline{-_7} + \overline{D_4D_5}\overline{-_7} + D_6\overline{D_7} + D_1D_2\overline{-_7} + \overline{D_3D_4}\overline{-_7} + \overline{D_5D_6}\overline{-_7} + D_7$$

$$GS = D_0 + D_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7$$

Where $\overline{D_{i-j}} = D_i\overline{D_{i+1} \ldots D_j}$. These expressions were translated into a circuit schematic.

5.3 Implementation and Simulation in Cadence Virtuoso

The priority encoder was designed and verified using the Cadence Virtuoso Design Platform. The key steps are outlined below:

1. **Project Setup**: A new library was created in the Cadence Library Manager, referencing a specific technology library [Specify the technology library name, e.g., ‘gpdk045‘].
2. **Schematic Capture**:
    - A new cellview (type ‘schematic‘) was created for the 8 to 3 Priority Encoder.
    - The logic derived in Section 4.2.2 was implemented using standard digital gates (e.g., AND, OR, NOT, NAND, NOR) available from the technology library [Specify if using gate-level cells or transistor-level design].
    - Inputs (D_0 to D_7) and outputs (Y_2, Y_1, Y_0, GS) were added as pins to the schematic.
    - The gates were instantiated and wired together according to the logic expressions.
    - The completed schematic is shown conceptually in Figure 2.
3. **Testbench Creation**:
    - A separate cellview (type ‘schematic‘) was created to serve as the testbench for the encoder schematic.
    - The encoder schematic was instantiated as a symbol within this testbench schematic.
    - Input stimuli were generated using voltage sources configured to produce desired digital waveforms (e.g., ‘vsource‘ with piece-wise linear (PWL) characteristics, or ‘vpulse‘) to cover various input combinations as defined by the truth table, including single active inputs and multiple active inputs simultaneously.
    - Output signals were connected to net labels or output pins for observation.
    - Ground and power supply (‘vddı, ‘gndı) were connected as required by the technology library.
4. **Simulation Setup and Execution**:
    - The Virtuoso Analog Design Environment (ADE) [Specify if using ADE L, XL, or Assembler] was launched from the testbench schematic.
    - A simulation analysis was configured. For functional verification, a ‘transient‘ anal- ysis was chosen to observe the output behavior over time in response to the input stimuli. The simulation duration was set sufficient to cover all test cases.
    - The desired input and output nets/pins were selected for plotting.

- The simulation was run using the chosen simulator (e.g., Cadence Spectre or Xcelium).

5. **Waveform Analysis**:

   - The simulation results were viewed in the Virtuoso Waveform Viewer.
   - The output waveforms (Y_2, Y_1, Y_0, GS) were inspected visually and analyzed against the input waveforms (D_0 to D_7) to confirm that the outputs changed correctly ac- cording to the priority logic and truth table for all simulated input combinations.

5.4 Circuit Diagram (Schematic View)

The actual circuit diagram implemented in Cadence Virtuoso is the schematic capturing the logic derived from the truth table. This schematic consists of interconnected logic gates (or transistors) instantiated from the technology library.

Figure 2 shows a conceptual representation of the schematic captured in Cadence Virtuoso. *(Note: The actual schematic can be complex; show a representative part or a simplified top-level view if the full schematic is too large/dense).*

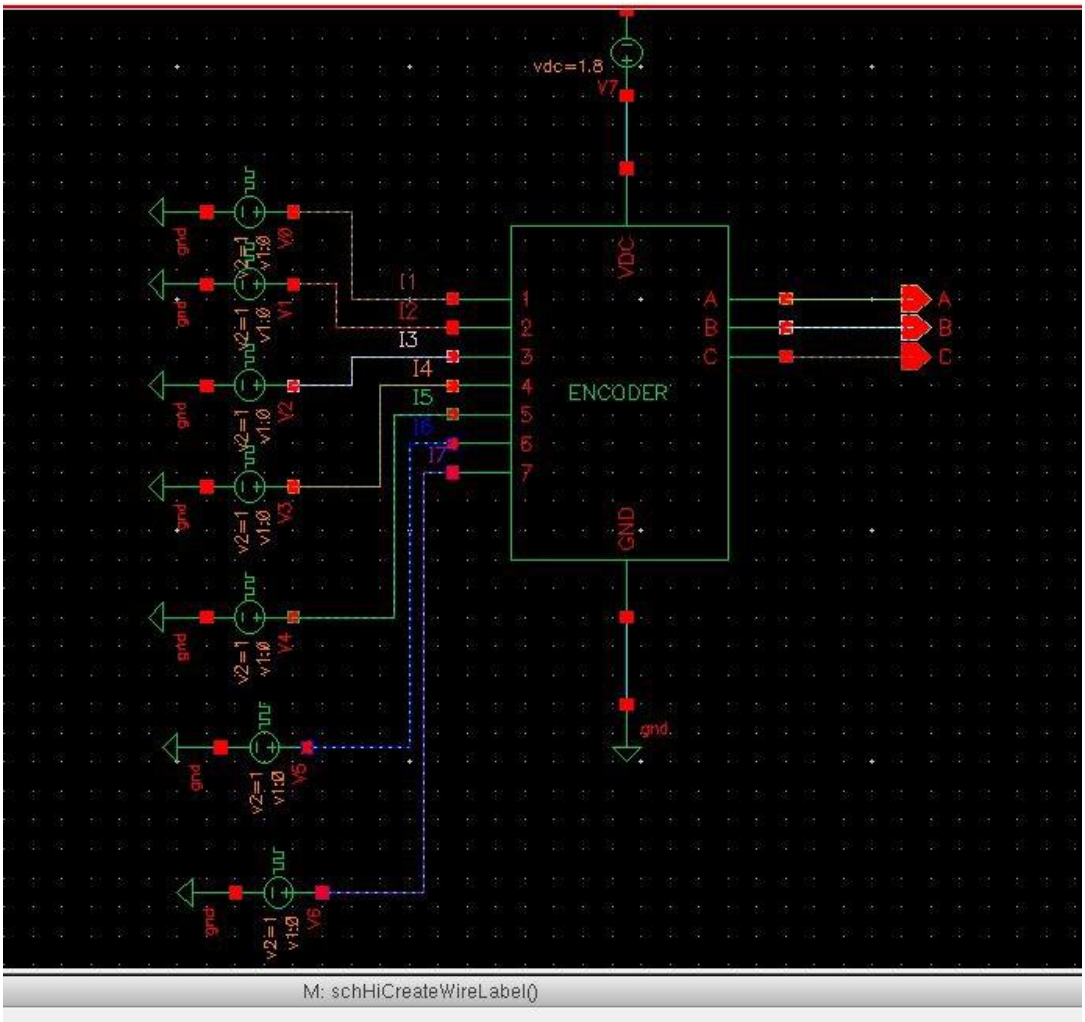

Figure 2: Conceptual Schematic of 8 to 3 Priority Encoder in Cadence Virtuoso

Figure 2 depicts the schematic implementation of the 8 to 3 Priority Encoder using stan- dard digital logic gates from the [Technology Library Name] library. The gates are inter-

connected to realize the Boolean expressions for outputs Y_2, Y_1, Y_0, and GS, incorporating the required priority logic.

6 Results and Discussion

The 8 to 3 priority encoder designed in Cadence Virtuoso was successfully simulated, and its functionality was verified using transient analysis. The simulation results are presented as waveforms, demonstrating the circuit's response to various input stimuli.

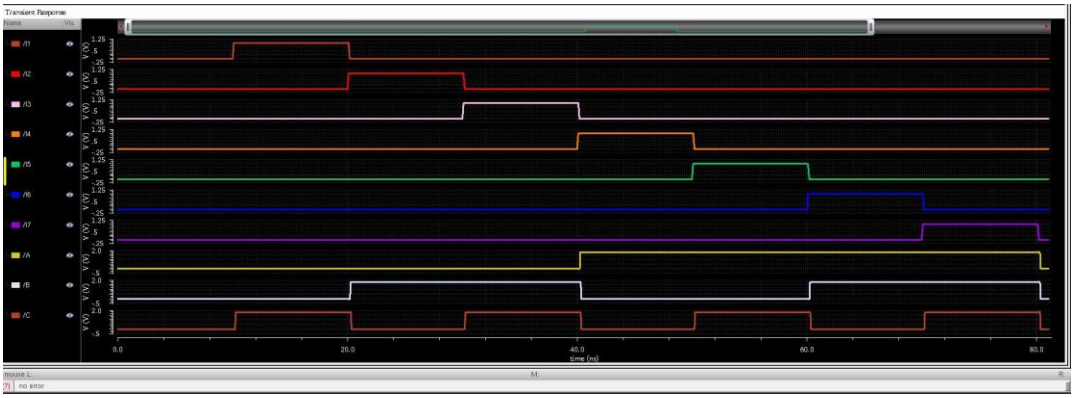

Figure 3: 8 to 3 outputwaveform

Figure 3 shows a snapshot of the simulation waveforms from the Cadence Virtuoso Wave- form Viewer. The input signals (D_0 through D_7) are driven by the configured voltage sources in the testbench. The output signals (Y_2, Y_1, Y_0, GS) respond according to the priority logic.

Key observations from the waveform analysis include:

- When all input signals $D_0 - D_7$ are low, the outputs Y_2, Y_1, Y_0 are low, and the GS signal is low, indicating no active input.
- When a single input D_i is high, the outputs Y_2, Y_1, Y_0 show the binary representation of i, and GS is high.
- When multiple inputs are high simultaneously, the outputs Y_2, Y_1, Y_0 correctly show the binary representation of the highest index i for which D_i is high. For example, as demonstrated in the waveform, when both D_3 (binary 011) and D_5 (binary 101) are high, the outputs correspond to D_5, resulting in $Y_2Y_1Y_0$ = 101 and GS high. Similarly, if D_1, D_4, D_7 are high, the outputs are 111 (for D_7), and GS is high.
- The GS signal transitions high whenever at least one input $D_0 - D_7$ is high, and low only when all inputs are low.
- The simulation also allows observing the propagation delay, i.e., the time delay be- tween an input transition and the corresponding output transition. While detailed timing analysis requires specific characterization corners, the transient simulation visually shows this delay.

The simulation results confirm that the schematic design implemented in Cadence Vir- tuoso correctly realizes the 8 to 3 priority encoding function according to the specified priority levels and truth table. The use of Cadence tools allowed for a detailed functional verification at the circuit level.

7 Mapping POs and SDGs

7.1 Program Outcomes (POs) Mapping

Table 2: Mapping of Program Outcomes (POs)

PO No.	Program Outcome	Relevance
PO1	Engineering Knowledge	Applied principles of Digi- tal Logic Design, Boolean al- gebra, combinatorial circuits, and fundamental IC design concepts.
PO2	Problem Analysis	Analyzed the priority encod- ing requirement and trans- lated it into a circuit design using library components.
PO3	Design/Development of Solutions	Designed the logic circuit schematic for the 8 to 3 pri- ority encoder in an EDA envi- ronment.
PO4	Conduct Investigations of Complex Problems	Verified the design's function- ality and observed behavior (including potential timing) through circuit-level simula- tion using Cadence tools.

7.2 Sustainable Development Goals (SDGs) Mapping

The project on the Design and Simulation of 8 to 3 Priority Encoder using Cadence Virtuoso contributes indirectly to several United Nations Sustainable Development Goals (SDGs) by fostering technical skills and contributing to the knowledge base for developing modern electronic systems.

7.2.1 SDG 1: Quality Education

This project directly supports SDG 4 by providing a practical and in-depth learning experience in microelectronics and digital design:

- It applies theoretical knowledge of digital logic and circuit design within a professional EDA tool environment, bridging the gap between theory and practice.
- It develops crucial skills in using industry-standard software for schematic design, simulation, and verification, which are highly valued in the electronics industry.
- It enhances analytical skills through interpreting simulation results and verifying complex logic behavior.

7.2.2 SDG 2: Industry, Innovation, and Infrastructure

The work aligns with SDG 9 by contributing to the fundamental technological capabilities required for modern industries and infrastructure:

- Designing and simulating basic digital components like priority encoders in profes- sional tools is a foundational step in the design of complex integrated circuits used in computing, communication, automation, and control systems.
- Proficiency in EDA tools like Cadence Virtuoso is essential for innovation in hard- ware design and the development of advanced electronic systems that power smart infrastructure and industries.
- Understanding the IC design flow from schematic to simulation contributes to the ability to develop more efficient, reliable, and high-performance electronic hardware.
- While a basic component, successful implementation and verification using profes- sional tools signify the acquisition of skills necessary for contributing to the develop- ment of cutting-edge technology.

8 Conclusion

This project successfully designed and functionally verified an 8 to 3 priority encoder using the Cadence Virtuoso Design Platform. The design was implemented by capturing the circuit schematic based on the priority logic and truth table, utilizing standard digital gates from a specified technology library.

Functional simulation was performed using a transient analysis within the Cadence sim- ulation environment (Spectre/Xcelium). A dedicated testbench was created to generate input stimuli covering all critical operating conditions, including scenarios with multi- ple inputs active simultaneously. The analysis of the resulting waveforms confirmed that the designed circuit correctly implemented the priority encoding function, validating the design against the truth table specifications.

This project provided practical experience with schematic capture, testbench setup, and simulation using industry-standard EDA tools, demonstrating the complete front-end IC design flow from logic concept to verified schematic.

Future work could extend this project to include the physical layout design of the priority encoder in Cadence Virtuoso, followed by post-layout simulation to analyze the impact of parasitic effects on timing and performance. Further investigations could involve optimiz- ing the design for specific parameters like power consumption, area, or speed within the chosen technology library.

Bibliography

[1] M. Morris Mano and Michael D. Ciletti, *Digital Logic and Computer Design*, [Specify Edition if different]. Prentice Hall, [Specify Year].

[2] Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, [Specify Edition if different]. McGraw-Hill, [Specify Year].

[3] John P. Hayes, *Introduction to Digital Logic Design*. Addison-Wesley, [Specify Year].

[4] Ronald J. Tocci, Neal S. Widmer, and Gregory L. Moss, *Digital Systems: Principles and Applications*, [Specify Edition if different]. Pearson Education, [Specify Year].

[5] Randy H. Katz and Gaetano Borriello, *Contemporary Logic Design*, [Specify Edition if different]. Pearson Education, [Specify Year].

[6] GeeksforGeeks, ”Encoder in Digital Logic,” [Online]. Available: https://www.geeksforgeeks.org/encoder-in-digital-logic/ [Accessed May 1, 2025].

[7] TutorialsPoint, ”Digital Electronics Encoders,” [Online]. Available: https://www.tutorialspoint.com/digital$_e$*lectronics/digital*$_e$*ncoders.htm*[*AccessedMay*1*,* 2025].

[8] Electronics Tutorials, ”Priority Encoder and Digital En- coder Tutorial,” [Online]. Available: https://www.electronics-tutorials.ws/combination/comb$_6$*.html*[*AccessedMay*1*,* 2025].

[9] ElProCus, ”Priority Encoder: Truth Table, Verilog Code & Its Applications,” [Online]. Available: https://www.elprocus.com/priority-encoder/ [Accessed May 1, 2025].

[10] abhinandann, ”8-to-3-bit-Priority-Encoder” [Source Code], GitHub. Available: https://github.com/abhinandann/8-to-3-bit-Priority-Encoder [Accessed May 1, 2025].