FPGA - Based Braille Code Reader for Visually Impaired Individuals

Harshini J,
Student, Department of Electronics and Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, India.
harshinij2005@gmail.com

Larisa Maria V,
Student, Department of Electronics and Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, India.
larisamariav2506@gmail.com

Daslin Princy A,
Student, Department of Electronics and Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, India.
daslinprincya@gmail.com

Guruprriya RS,
Student, Department of Electronics Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, India.
guruprriyars2006@gmail.com

Venkatesh T,
Assistant Professor III,
Dept. of Electronics and Communication Engineering,
KPR Institute of Engineering and Technology
Coimbatore, India
venkatt87@gmail.com

***Abstract* —** People who have visual impairments can access better resources through assistive technologies which serve as essential tools that fulfill their specific requirements. The research presents a Braille text conversion system which operates through hardware components that use an Artix-7 FPGA. The system establishes a connection to the computer through its use of UART protocol for serial text transmission. The system decodes incoming ASCII characters through synthesizable Verilog logic which creates a mapping to standard six-dot Braille encoding. The system uses servo motor-driven actuators to create Braille output which generates tactile pins. The system clock operates at 50 MHz which enables synchronized processing while the system transmits data at a communication rate of 9600 baud. The system architecture maintains modular design which allows for scalability and enables fast processing times that meet requirements of portable real-time reading assistance devices.

Keywords: Artix-7 FPGA-, UART communication, Baud rate, Servo motors.

I. INTRODUCTION

Information is at the heart of human life, and written communication plays a central role in accessing it. For people who are visually impaired, supportive technologies make this possible, and among them, Braille remains one of the most effective and trusted methods because it delivers information directly through touch.

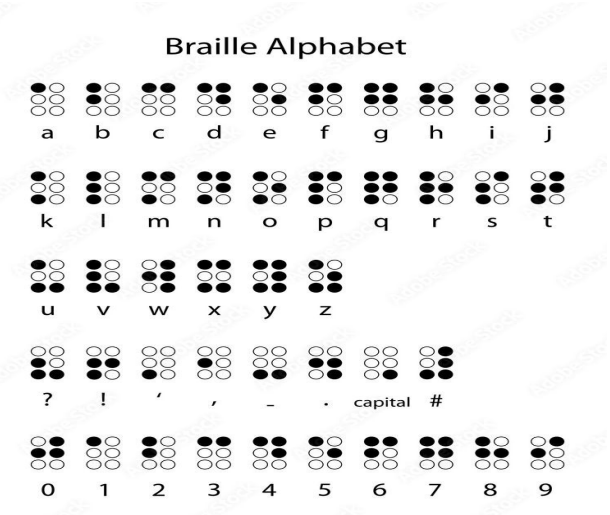
Still, many current Braille learning aids and electronic Braille display devices are either expensive, bulky or both, which seriously limits their everyday use and availability. In addition, most of these systems are difficult to adapt for different purposes or applications and strict in design.

New improvements in embedded systems and digital electronics have made it possible to build smaller, more adaptable Braille displays. In some research prototypes, scholars have used movers to raise and lower the Braille dots because they offer good correctness of position.

This paper introduces an FPGA-based Braille display system that uses servo motors to actuate the Braille dots. The end goal is to provide a practical, affordable system that allows for a more comfortable and easier reading experience

Field-Programmable Gate Arrays (FPGAs) are particularly well suited for tasks that involve specific, real-time coordination of multiple outputs in parallel. Their capacity to be reprogrammed also means the control logic for creating Braille patterns can be fully changed without transforming any hardware or bringing it up to date.

Through the blend of FPGA-based parallel control and servo motors, this work aims to provide a practical step forward in accessible technology, making Braille-based reading more accessible to visually impaired people across a wider range of situations.



II.LITERATURE REVIEW

R. Karthikeyan and S. Priya (2015) hand over a basic Braille display using a microcontroller. The system preserved only a finite set of characters and managed at low speed. Our project uses an Artix-7 FPGA, which allow faster processing and parallel execution. Different the fixed nature of the earlier system, our design supports real-time serial input through UART using PuTTY. Also, tactile output in our project is strongly developed using servo motors, making it more familiar and useful.

A. Joseph and M. Thomas (2016) planned a text-to-Braille conversion technique. The system set on mainly on software-based planning. The acceptance lacked direct tactile hardware output and real-time contact. On the other hand, our project combines both software and hardware, different ASCII characters into physical Braille patterns. The use of FPGA-based command in our system help system consistent and strong.

Thus, our work improvements are behind simple changing logic into a complete tactile reading solution.

S. Ravi and. The clarified Braille patterns are instantly converted into physical movement using servo motors. This combination improves real-time user-friendliness.

R. Singh and P. Arora (2019) created a Braille learning tool using a basic embedded platform.

The system is limited adaptability and slower reaction. Our FPGA-based design delivers better control correctness and higher speed action. The UART-based services also allow easy connecting with terminal software like PuTTY. These features make our system more flexible and advanced.

N. Rao and V. Sharma (2019) implemented a Braille display using a microcontroller. Although it is beginner-friendly, the system showed noticeable delays in updating the characters. Our project rectifies the delay as the ASCII values respond immediately to the serial input which is given by the user.

H. Park and J. Lee (2020) proposed a serial communication-based display system without tactile output, as the work focused mainly on data transmission. When it comes to our project, both serial communication and the corresponding tactile output are implemented for better accessibility for blind people.

S. Banerjee and P. Das (2020) presented Braille encoding techniques for embedded systems.

This depicts the theoretical explanation only, but it lags when it comes to hardware implementation. Our project came up with a hardware implementation by building a bridge between theoretical explanation and real-time implementation.

K. Mohan and R. Iyer (2021) developed a basic tactile display for the blind reader's educational purposes. The system is mainly concerns for its demonstration, while it failed to assess the tactile readability. Our project overcame this obstacle by incorporating servo motors for sensitivity.

J. Martin and A. Lopez (2021) proposed a simple hardware implementation with assistive technology. Although it is easy to implement, it does not provide better time efficiency. Our project overcame this with a better approach by implementing PWM signals.

V. Ramesh and S. Kannan (2022) introduced a low-speed Braille conversion system that can be operated in offline mode. But the contents available in online mode cannot be implemented easily. Our project allows live character entry with an even faster approach, making our system more effective and easy to implement.

T. Wilson and R. Brown (2022) proposed a user-friendly Braille pattern without the proper prototype; our project aims to overcome it by implementing a proper prototype.

II. METHODOLOGY

1. System Overview

The project aims to develop a Braille reader that converts text to Braille dots using servo motors. The system gets a character input from puTTy software through UART communication with a baud rate of 9600 bps.The serial communication is processed through FPGA board to obtain the desired braille pattern response. The servo motor reprents a braille pattern only when the serial input is sent ,if not it remains in default position.

The system demonstrates that assistive technology development for visually impaired users who require Braille output can be achieved through the use of FPGA reconfigurable hardware.

2. Hardware Architecture

2.1 FPGA Platform

The design operates on an Artix-7 FPGA development board which functions at a system clock frequency of 100 MHz. The system selected FPGA technology because it provides fast data processing capabilities combined with the ability to handle multiple tasks simultaneously. Users can design their digital circuits through this system. The system uses FPGA technology to control UART operations while decoding Braille signals and transmitting control signals to the servo motors.

2.2 UART Communication Interface

The Verilog HDL language was used to create a UART receiver module which receives data from PC systems. The PuTTY terminal has been set up with these configuration options:

Baud rate: 9600 bps

Data bits: 8

Stop bit: 1

Parity: None

The UART module starts to detect the start bit while it uses a baud rate counter to sample incoming bits and then proceeds to decode the 8-bit ASCII character which creates a valid signal after receiving the data.

The servo motor output interface uses servo motors to create six distinct dots which make up a Braille cell. The system uses one servo motor to control each individual Braille dot. The servo motor turns 90 degrees to show a raised dot whenever the Braille dot becomes active. The servo motor maintains its 0 degree position when the dot remains inactive.

2.3 Servo Motor Output Interface

The system requires an external power supply because servo motors consume higher current while operating at 5V. The FPGA generates PWM control signals which operate the servo motors. The system establishes a common ground connection between the FPGA and the external power supply to ensure proper system operation.

3. UART Receiver Design The UART receiver is implemented in Verilog HDL. A clock divider is used to generate the baud rate timing from the 100 MHz clock. The receiver starts data receipt by detecting the start bit which it uses to sample data bits that get saved into an 8-bit register until the receiver checks the stop bit. A valid signal gets generated after the system successfully receives the data.

4. ASCII to Braille Conversion The received ASCII character is converted into Braille pattern using a lookup table inside the FPGA. Each character is mapped to a 6-bit Braille code, where each bit represents one Braille dot.

The letter A gets transformed into its Braille dot representation. The system uses combinational logic to create this conversion which allows for character expansion.

5. Servo Control Using PWM

The system uses FPGA-generated PWM signals for controlling the operation of servo motors. The PWM module generates a 50 Hz signal which produces various pulse width outputs.

A pulse width of one millisecond shows a zero degree angle position. A pulse width of 1.5 milliseconds shows the 90 degree position.

The FPGA transmits PWM signals to the servos which operate according to the Braille code. Active dots rotate to 90 degree, and inactive dots remain at 0 degree.

6. System Integration

The system consists of different Verilog modules such as clock divider, UART receiver, ASCII to Braille decoder, PWM generator and top-level integration module. The top module connects all modules and controls the data flow from UART to servo motors. The design uses Xilinx Vivado for implementation and the resulting bitstream gets loaded onto the FPGA board.

7. Experimental Procedure

The system tests itself by sending characters from the PuTTY terminal. The FPGA receives the ASCII value when a character is typed and it converts the value into a Braille pattern which activates the corresponding servo motors. The servo motors move to 90 degree position for active dots while they remain at 0 degree position for inactive dots.

Quite a few characters were tested to confirm that the Braille output aim correctly. The system give an idea of real-time performance because engaged dots caused the servo motors to respond at once.

8. Performance Evaluation

The system performance analysis requires five different measurement components which measure UART reception exactness and servo response time and Braille conversion accuracy and FPGA measure usage. The system successfully transforms text characters into tactile Braille through its usage of servo motors to create Braille output.

9. Advantages of the Proposed System

The suggested system delivers Braille output in real-time because its hardware design supports character add on functions. The processing speed of systems based on FPGA technology transcend that of microcontroller-based systems because FPGA technology enables at the same time execution of various tasks. The system has potential to evolve into a complete Braille reading solution which will support visually impaired users.

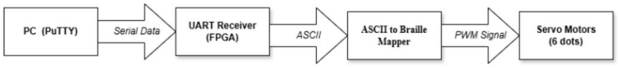


Fig 1: Systematic flow

Figure 1 demonstrates the overall system architecture, including the key components and processes described in this section

IV.ADVANTAGES

The FPGA helps the system work fast, so text is converted into Braille without delay. All functions like communication and Braille conversion are done in one FPGA, making the design compact and simple. The system is low cost because it uses basic hardware rather than costly Braille display units.

Direct control from the FPGA gives exact Braille output. The design is flexible and Braille patterns can be updated by changing the code. This makes the system suitable for future improvements and educational use.

V. CHALLENGES

Even though the system gives exceedingly promising outcomes, there are some challenges:

During the implementation of this idea several challenges were encountered, particularly when the integration of both servo motors and UART communication comes into force.

Initially the individual integration of the servo motors and UART separately appears to be uncomplicated when compared to the combined integration of these components as the problems arouse due to the relation of clock frequency synchronization. As both of these requires precise timing with the clock pulse of 50MHz, integration without affecting the signal stability becomes the most difficult part.

Another important challenge is that when the generation of the PWM signals with the help of conventional duty cycle does not cooperate with the angular movement of servo motors. Therefore after continuous research it is known that generating the clock pulse based on the PWM signal is more efficient than varying the duty cycle. AS a result of this the rotation of servo is finally implemented as expected.

Next challenge occurred while the direct integration of servo to the FPGA board. Even a minor mistake resulted in incorrect output. To resolve this repeated testing and adjustment of the timing pulse until the desired output is acquired. However rectifying all these errors enhanced the experience and practical knowledge.

VI. RESULTS AND DISCUSSION

This project highlights using an FPGA with servo motors can create a simple and effective Braille display system. One of the most , meaningful outcomes is how the design becomes more easier and `compact to manage servo-motors setups. In addition, the FPGA makes controlling multiple Braille dots good and smooth, which improves the general people experience. Such a combination reduces complexity without risky performance. Generally, the present system stands as a realistic and people determined solution that can support open and low-cost helpful technology for visually impaired individuals.

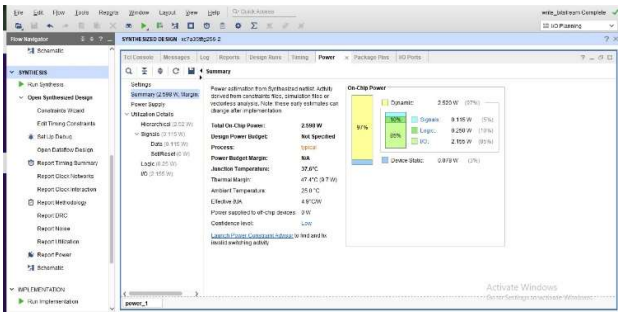


Fig 2 : On-chip power consumption breakdown of the synthesized design in Vivado Tool.

Figure 2 represent that the on-chip power calculation for the synthesized design targeting the Xilinx Artix-7 FPGA (xc7a35ttg256-2) is shown in Fig.2. The power report, bring forth post-synthesis in structural mode, publish a total on-chip power of 2.598 W, with the huge majority (97%) assigned to dynamic power utilization.

Dynamic power dominates at 2.520 W (97% of total), while static (quiescent) power remains very low at 0.078 W (3%).

Within dynamic power, I/O power is the largest writer by far, calculating for 2.155 W (85%). This denote that the design is heavily interface-driven, likely involving a large number of active output pins, high toggle rates on I/Os, or important external interfacing activity.

Logic power supplies 0.250 W (10%), and signal (interconnect) power is minimal at 0.115 W (5%), suggesting logical internal logic utilization and fairly low switching activity in the core fabric at this stage.

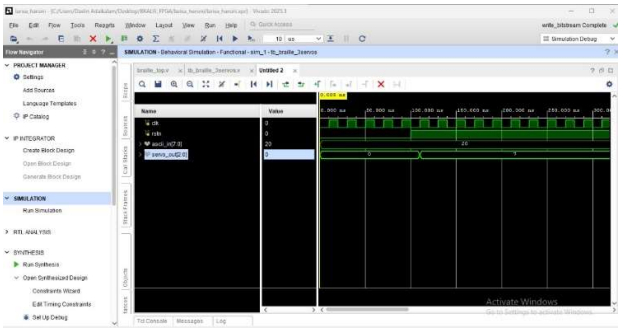


Fig 3 : Simulation result in Xilinx Vivado (2025.1)

Figure 3 shows a simulation result in Xilinx Vivado (2025.1) for a Braille-based FPGA project, displayed in a very practical and easy-to-understand way. The waveform window mean that how the system acts over time when different inputs are applied. The clock signal (clk) is continuously toggling, which verify that the design has a stable timing instance of. The reset signal (rstn) starts in a low state and is later set free, permit the circuit to starts normal operation

An 8-bit ASCII input (ascii_in[7:0]) is applied with a value of 20, act for the character being carry on by the Braille logic. After this input is given, the servo output signal (servo_out[2:0]) changes clearly from 0 to 7, reveal that the FPGA correctly explains the input and create the expected control pattern for the servos.Generally, this simulation confirms that the signal change are clean, the design logic is working correctly and the system answer properly to the given input before moving to hardware implementation

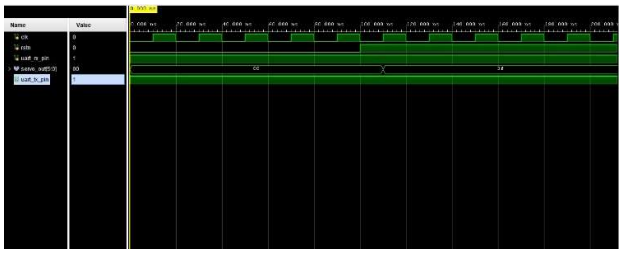


Fig 4 : Simulation waveform from Xilinx Vivado

Figure 4 shows a simulation waveform from Xilinx Vivado that explains how the FPGA design acts for a specific input in a action very clear way. The clock signal (clk) is running continuously with a steady pattern, which means the system timing is stable and the circuit is active. The reset signal (rstn) is kept at logic high, so the design is not in reset mode and is run normally.

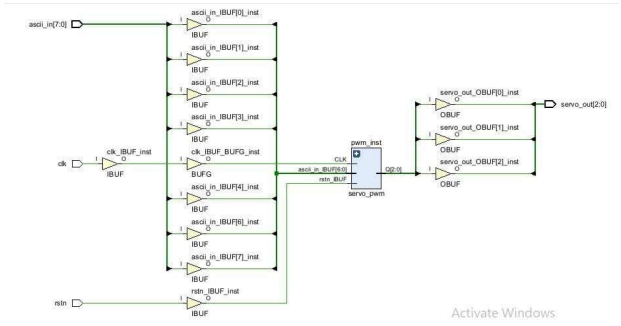


Fig 5: Schematic Layout from Vivado Tool.

The ASCII input (ascii_in[7:0]) is set to the value 42, representing the character currently given to the system. Even after this input is applied and retain constant, the servo output (servo_out[2:0]) remains at 0 throughout the simulation time. This shows that the given ASCII value does not match any specific Braille pattern or servo activation condition in the design. Overall, the waveform confirms that the system correctly maintains stable signals, reads the input and safely avoids unplanned servo movement when a baseless or unknown character is provided.

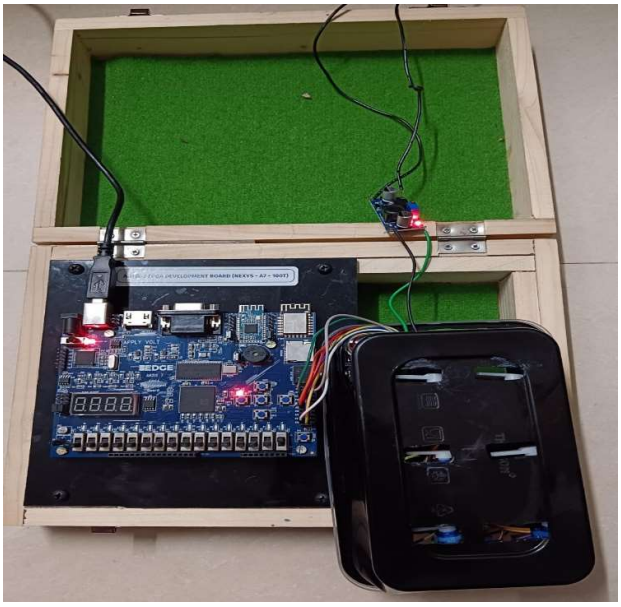


Fig 6: Hardware model of FPGA-Braille Code Reader.

The design takes ASCII input, operates it with a clock-controlled PWM module, and direct the servo motors properly. The clock safe guards keeps timing stable, and the reset protected smooth startup. In the end, output buffers send clean signals to control the servo movements correctly.

VII. REFERENCES

[1] S. Patil, R. Kulkarni, "FPGA Based Braille Display System for Visually Impaired," International Journal of Assistive Technologies, vol. 12, no. 3, pp. 45–52, 2019.

[2] A. K. Verma, P. Singh, "Text to Braille Conversion Using Embedded Hardware Platforms," IEEE International Conference on Smart Systems, pp. 210–215, 2020.

[3] M. R. Ahmed, S. Hossain, "Design of Refreshable Braille Cell Using Programmable Logic Devices," Microprocessors and Microsystems, vol. 79, 2020.

[4] K. R. Meena, A. Joseph, "Real Time Text to Braille Conversion System Using FPGA," IEEE Access, vol. 11, pp. 55678–55686, 2023.

[5] D. Kumar, R. Patel, "An FPGA Based Smart Braille Learning Device," International Journal of Embedded Systems, vol. 18, no. 1, pp. 12–20, 2024.

[6] C. Loconsole, D. Leonardis, M. Gabardi, and A. Frisoli, "BrailleCursor: An Innovative Refreshable Braille Display Based on a Single Sliding Actuator and Simple Passive Pins," *IEEE World Haptics Conference*, pp. 390–395, 2019.

[7] R. Gupta, P. K. Singh, and S. Bhanot, "Design and Implementation of Arduino-Based Refreshable Braille Display Controller," *Indian Journal of Science and Technology*, vol. 9, no. 33, pp. 1–6, 2016.

[8] B. Bhanushali, A. Dhoot, P. Gandhi, and K. Mehta, "Refreshable Braille Displays," *International Journal of Computer Applications*, vol. 180, no. 37, pp. 15–19, 2018.

[9] H. P. Savindu, R. Madushanka, T. Weerasinghe, and J. Wijesinghe, "BrailleBand: Blind Support Haptic Wearable Band for Communication Using Braille Language," *arXiv preprint arXiv:1901.03329*, 2019.

[10] M. Etezad, R. Joshi, and F. L. Cibrian, "Advancements in Refreshable Braille Display Technology: A Comprehensive Survey," *Displays*, vol. 90, pp. 103133, 2025.

[11] A. Walker and S. Sangelkar, "Design Exploration of Affordable Refreshable Braille Display Technology for Low-Income Visually Impaired Users," *Proceedings of the ASME Design Engineering Technical Conference*, pp. 1–8, 2018.

[12] M. N. Kamarudin, M. Y. A. Aboali, and N. S. Zainal, "Braille Display Systems for Blind People with Haptic Belt," *International Journal of Human and Technology Interaction*, vol. 4, no. 2, pp. 23–30, 2022.

[13] D. S. Mehta and R. N. Patel, "Electromechanical Design of a Refreshable Braille Display System," *Journal of Assistive Technologies*, vol. 14, no. 2, pp. 85–92, 2020.

[14] K. Mohan, R. Iyer, and S. Prakash, "Low-Cost Tactile Braille Display for Educational Applications," *International Journal of Engineering Research & Technology*, vol. 9, no. 6, pp. 742–747, 2020.

[15] L. Chen and Y. Wu, "Hardware Implementation of Braille Encoding and Display Systems," *International Journal of Embedded and Real-Time Communication Systems*, vol. 10, no. 1, pp. 45–54, 2019.