Design and Implementation of Morse Code Communication for Visual and Speech Impaired People

Dr. S L Mukthi[1], Sheethal C K[2]
[1]*Associate Professor, Department of ECE, Institute of Technology Bengaluru-04 India*
[2]*PG Student, Department of ECE, Bangalore Bangalore Institute of Technology Bengaluru-04, India*

***Abstract*—This paper presents the design and implementation of Morse Code Communication System using VHDL and behavior simulation with Vivado is described for application in assistive communications. Developed communication system consists of Morse code detector, Morse code decoder, baud rate generator, UART transmitter, simulation outputs logs and python-based speech synthesis of output text using Vivado Environment. Morse codes are entered by simulating timing-based input values using VHDL test bench and converting it to respective ASCII characters using Morse code decoding methodology. UART communication is done using Finite State Machine based Serial data transmission at 9600 baud rate. Simulation outputs are logged for storage purposes and then converted into audible speech through python text-to-speech synthesis program. Behavior simulation outputs confirmed successful Morse detection, ASCII conversion, UART communication and speech synthesis for various inputs including "I NEED WATER" and "PLEASE HELPME". Developed architecture proves to be an efficient and cost-effective approach for assistive communication system using FPGA simulation technology.**

***Index Terms*—Morse Code, FPGA, VHDL, UART, Vivado, Assistive Communication, Text To Speech Synthesis, ASCII Conversion.**

I. INTRODUCTION

In today's communication systems, assistive communication technologies play an important role in helping speech-impaired and visually impaired individuals communicate effectively with caretakers and society [1], [2]. Many differently abled individuals face difficulties in expressing their needs during emergency situations and daily communication activities. Existing communication systems often require expensive hardware devices, complex interfaces, and continuous human assistance. Therefore, there is a growing need for low-cost, reliable, and efficient assistive communication systems using digital technologies [5], [7].

Morse code is one of the earliest and most reliable digital communication techniques used for transmitting information through combinations of DOTs and DASHes[8], [12]. Due to its simplicity, Morse code has been widely used in military communication, aviation systems, maritime communication, emergency signaling applications, and wireless communication systems. Morse code communication requires only simple signal generation and timing analysis, making it suitable for assistive communication applications [6], [9].

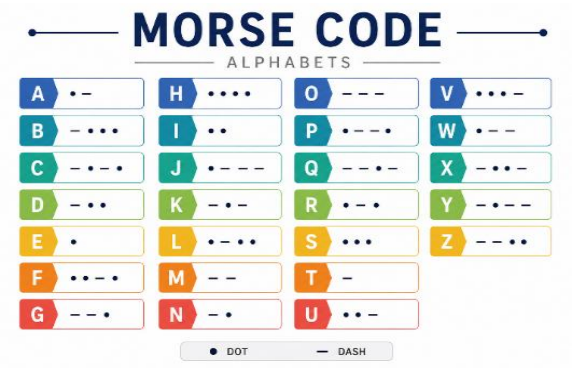

Fig. 1 Morse Code Alphabet Representation

With the advancement of FPGA and VLSI technologies, communication systems can be efficiently implemented using hardware description languages such as VHDL [4], [14]. FPGA-based systems provide flexibility, reconfigurability, parallel processing capability, faster execution speed, and reliable timing performance compared to conventional

microcontroller-based systems. These advantages make FPGA-based systems suitable for communication applications, embedded systems, signal processing, and assistive technologies [8], [15]. This project focuses on designing and implementing a Morse Code Communication System using VHDL and Vivado behavioral simulation. The complete system is developed using multiple modules such as Morse detector, Morse decoder, baud rate generator, UART transmitter, simulation output logging, and Python-based speech synthesis. The proposed system accepts Morse code inputs in the form of DOTs and DASHes generated through timing-based push-button simulation inside the VHDL testbench environment [10], [11].

The Morse detector module continuously monitors the input signal duration and identifies short-duration signals as DOT symbols and long-duration signals as DASH symbols. The detected Morse patterns are transferred to the Morse decoder module, where Morse sequences are converted into corresponding ASCII characters [12], [13]. UART communication is implemented for serial transmission of decoded ASCII data at 9600 baud rate using FSM-based UART transmission logic [14].

The generated communication outputs stored through simulation output logging using VHDL operations. Python-based text-to-speech conversion is integrated with the simulation outputs to generate audible speech for communication support [5], [10]. The entire communication setup is verified using the behavioral simulation of the Vivado design suite without using any FPGA hardware implementation [15], [16].

In traditional communication systems, microcontrollers were commonly used for signal processing and communication operations. However modern communication applications require higher speed, better timing synchronization, hardware flexibility, and parallel processing capabilities. FPGA-based communication systems satisfy these requirements efficiently because programmable logic resources, communication interfaces, and digital control modules can be integrated into a single platform [4], [8].

UART (Universal Asynchronous Receiver Transmitter) communication protocol is widely used in embedded systems because of its simplicity, low hardware complexity, and reliable serial communication support[14]. In the proposed system, UART communication is employed for transmitting decoded ASCII characters serially with proper baud-rate synchronization and timing control.

The developed VHDL-based Morse Code Communication System provides a low-cost, flexible, and efficient assistive communication solution for differently abled individuals. The system also demonstrates the integration of FPGA communication concepts, UART serial communication, Morse code decoding, TEXTIO file handling, and Python-based speech synthesis within a single simulation-based communication platform [5], [8], [10].

II. METHODOLOGY

This paper presents the design and implementation of a Morse Code Communication System using VHDL and Vivado behavioral simulation for assistive communication applications. The developed communication model integrates Morse detector, Morse decoder, baud rate generator, UART transmitter, simulation output logging, and Python-based speech synthesis using Vivado Design Suite 2022. The methodology mainly focuses on Morse code detection, ASCII conversion, UART communication, waveform verification, and speech generation using simulation-based communication techniques. The proposed system attempts to provide a low-cost and flexible assistive communication solution compared to traditional hardware-dependent communication systems.

The complete architecture integrates multiple modules such as Morse detector, Morse decoder, baud rate generator, UART transmitter, simulation output file generation, and Python-based speech synthesis into a single communication system.

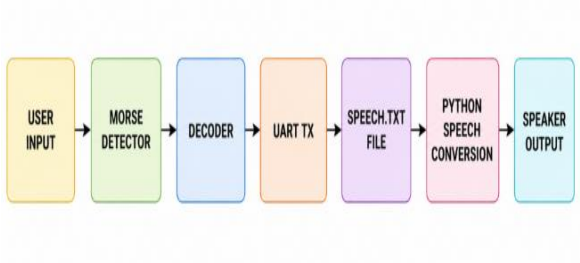

Fig. 2 Proposed Morse Code Communication System Architecture

The Morse code inputs are created by means of timing based simulated input signal within VHDL testbench

environment. There are following block components present within the overall system architecture design:

- Morse detector
- Morse decoder
- Baud rate generator
- UART transmitter
- VHDL testbench
- Simulation output file generation
- Python speech synthesis

The Morse detector component continuously checks on the duration of simulated input signals within the VHDL testbench environment. Simulated input signals having smaller timing duration are categorized as DOT symbols whereas those that have larger timing durations are considered as DASH symbols. The Morse detector makes use of simulation clock counting techniques to make timing based Morse code symbols identification without having any need for physical implementation of hardware. It produces symbols, valid, and done signals that facilitate communication with decoder module.

The Morse decoder module translates Morse code into ASCII codes. Morse symbols are stored in sequence and matched with preset Morse code to generate ASCII codes of the characters. The Morse decoder module works effectively and translates Morse codes to ASCII codes needed for communication.

The baud rate generator module generates timing signals for UART communication. The designed communication system employs a 100 MHz system clock frequency for communication at 9600 baud rate. The UART transmitter module sends out decoded characters serially using the UART data transmission protocol with start bit, data bit, and stop bit logic.

The proposed work also integrates simulation output file generation operations within the VHDL testbench environment. The generated ASCII communication outputs are stored into a speech.txt file during simulation. Python-based text-to-speech synthesis is integrated with the generated text outputs for audible speech generation. All modules in the communication system have been simulated with the Vivado software for behavioral analysis without needing hardware implementation. Waveform simulation was performed to monitor the operation of Morse code detection, ASCII code generation, UART transmission, and baud timing for all communication components.

III. PROPOSED WORK

The proposed work is related to the design and implementation of a Morse Code Communication System with the help of VHDL and behavioral simulation in the Vivado environment for assistive communication purposes. The designed communication system takes the timing-based Morse input signals, which have been provided using the VHDL testbench environment, and converts them into ASCII and audible messages.

The entire communication architecture contains the Morse detector, Morse decoder, baud rate generator, UART transmitter, simulation output logging, and speech synthesis by means of Python language for the generation of ASCII character and audible messages. Morse detector detects DOT and DASH characters using the timing analysis process, whereas Morse decoder translates the Morse code into appropriate ASCII characters required for communication message generation.

Serial communication by means of UART is done using FSM-based serial transmission technique along with the synchronization with baud rate. Generated simulation output messages are logged into an output file and then translated into ASCII messages using text-to-speech synthesis technique by the use of Python programming.

Behavioral analysis of waveform has been done to validate Morse detection, ASCII generation, UART transmission, and speech generation. There are several advantages that can be gained from the proposed communication architecture.

Letter	Morse	Length	Pattern (binary)	Case key	ASCII
A	.-	010	0001	010 & 0001	0x41
E	.	001	0000	001 & 0000	0x45
T	-	001	0001	001 & 0001	0x54
I	..	010	0000	010 & 0000	0x49
M	--	010	0011	010 & 0011	0x4D
S	...	011	0000	011 & 0000	0x53
O	---	011	0111	011 & 0111	0x4F
H	....	100	0000	100 & 0000	0x48
B	-...	100	1000	100 & 1000	0x42
C	-.-.	100	1010	100 & 1010	0x43

Fig. 3 Morse Code to ASCII Mapping Table

Morse decoder translates recognized Morse patterns into relevant ASCII characters using Morse to ASCII mapping tables. Morse codes recognized by the

detector are stored and compared with preset Morse codes. Once the match is completed, appropriate ASCII characters are created.

Fig. 3 shows the mapping tables that include Morse representation, storage of binary patterns, creation of case key, and ASCII equivalents utilized by the Morse decoder. The use of binary representation of Morse codes makes comparison much easier.

The baud rate generator block generates timing pulses needed for UART serial communication. For the generation of timing pulses at 9600 baud-rate UART, 100 MHz system clock has been considered with the help of clock divider using counters.

The baud divider value is calculated as:

Baud Divider=100000000 / 9600=10416.

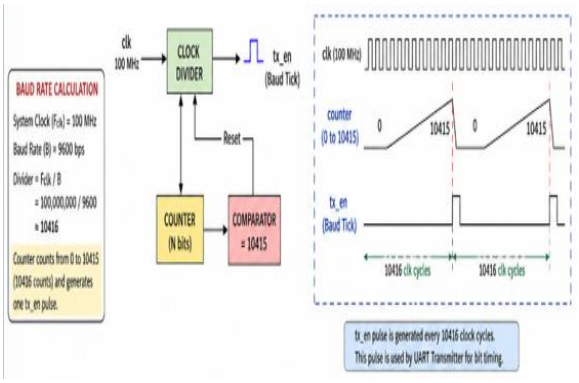

Fig.4 Baud Rate

The UART serial communication frame contains one start bit, eight data bits, and one stop bit. Thus, the size of UART Serial frame will be:

1+8+1=10 bits.

The UART bit transmission time is:

Bit Time = 1 / 9600=104.16 µs

Thus, the total UART serial frame transmission time will be:

10 × 104.16 µs = 1.0416 ms

The generated tx_en pulse will provide synchronization in the UART serial communication system.

The UART transmitter module performs serial transmission of decoded ASCII characters using FSM-based UART communication logic implemented in VHDL. The transmitter architecture mainly consists of FSM controller, bit counter, shift register and serial output logic blocks for synchronized serial communication.

The FSM controller manages different UART transmission states such as: IDLE state, START state, DATA state and STOP state.

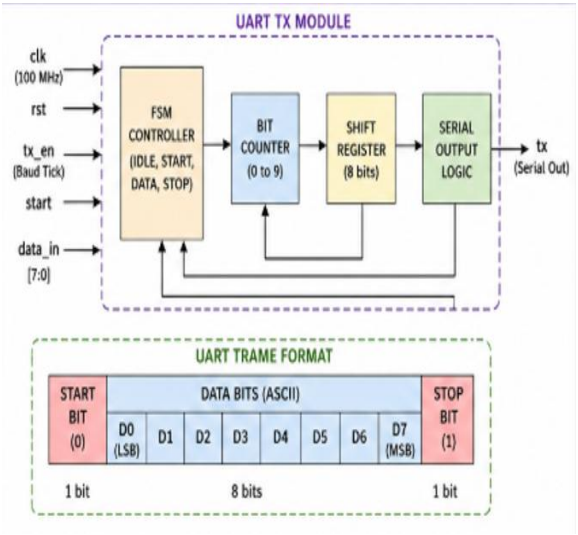

Fig. 5 UART Transmission Architecture and Frame Format

During the IDLE state, the tx signal remains HIGH until transmission begins. Once the start signal becomes active, the UART transmitter enters the START state and transmits the start bit. In the DATA state, ASCII characters are serially transmitted from Least Significant Bit (LSB) to Most Significant Bit (MSB). After completion of data transmission, the STOP state transmits the stop bit and returns the transmitter back to IDLE condition.

Simulation output logging is integrated within the VHDL testbench environment to store decoded communication outputs generated during simulation. The generated text outputs are later accessed using Python-based speech synthesis for audible voice generation.

Python Speech Synthesis module decodes the generated communication data and converts textual information to speech signals. The combination of VHDL simulation with Python Speech Synthesis enhances communication capabilities for persons with disabilities, especially those with speech disabilities and visual impairments.

The proposed communication system shows successful implementation of Morse Code decoding, UART communication, timing synchronization, and

speech synthesis in one FPGA simulation platform.

IV. RESULTS AND DISCUSSION

The proposed Morse Code Communication System was successfully verified using Vivado Design Suite 2022 behavioral simulation. The complete communication flow including Morse detection, ASCII conversion, UART transmission, simulation output file generation, and Python-based speech synthesis was tested successfully.

The simulation results confirm accurate Morse code detection, reliable ASCII generation, synchronized UART communication, and proper speech output generation for assistive communication applications.

A. Morse Detection Result

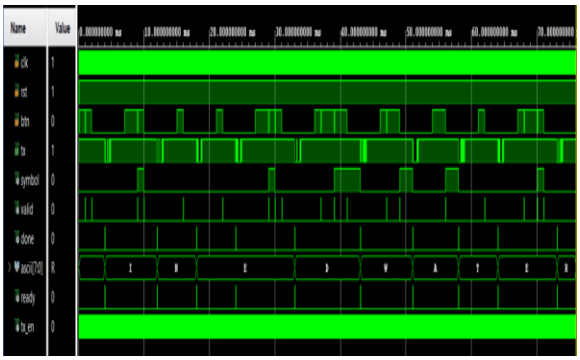

Fig. 3 Morse Detection and ASCII Conversion Waveform for “I NEED WATER”

The waveform shows timing-based Morse detection and ASCII conversion performed inside the VHDL simulation environment. Morse input signals generated through the testbench are identified as DOT and DASH symbols using signal duration analysis.

The detector generates symbol, valid, and done signals for synchronization with the decoder module. After pattern matching, the decoder converts Morse sequences into corresponding ASCII characters. The ascii[7:0] signal changes according to the decoded character outputs and generates the message “I NEED WATER”.

The observed timing behavior confirms proper Morse detection, decoder operation, and synchronized character generation within the developed communication model.

B. UART Transmission Result

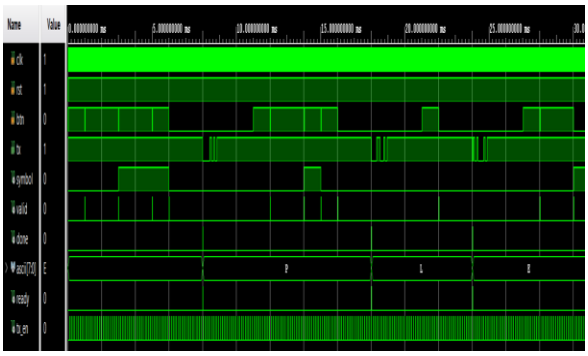

Fig. 4 UART Serial Transmission Waveform

UART Waveform shows serial transmission of ASCII data characters decoded at 9600 Baud Rate. During the idle state, the tx data is HIGH, and when the start-bit is transmitted, it goes to LOW. Communication between UART happens using a single start-bit, eight data-bits, and a single stop-bit.

The tx signal generated is a valid UART data stream following proper timing according to baud-rate control.

C. Complete Morse Communication Result

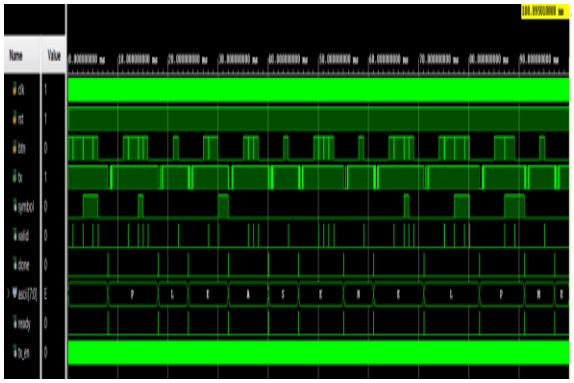

Fig. 5 Complete Morse Communication Waveform

The waveform shows that the whole process of Morse processing to obtain ASCII and then transmission by UART has been achieved successfully. The timing pattern Morse codes have been decoded successfully in corresponding ASCII symbols, and the ascii[7:0] gives output “PLEASE HELPME”.

A valid signal implies that Morse has been detected, whereas done signifies completion of the processing of individual character. UART transmission has been done through the use of the tx signal in serial communication.

This simulation results obtained clearly depict proper

timing synchronization and Morse decoding as well as UART transmission.

D. Python Speech Generation Result

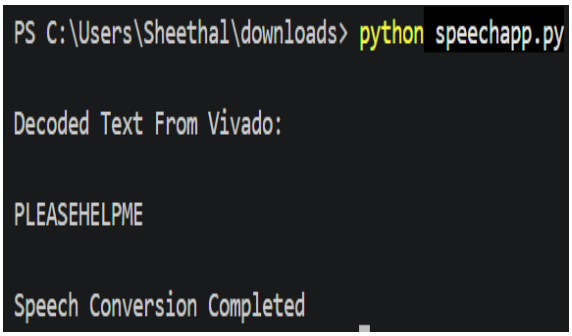

Fig. 6 Python Speech Synthesis Output

The simulation result text log is synthesized with the use of text -to-speech synthesis in Python. Confirmation from the terminal output on the successful extraction of text "PLEASEHELPME" and voice synthesis is received. Integration of the VHDL simulation output results with Python allows for synthesizing the Morse message into voice messages.

V. CONCLUSION

Based on the results, the designed Morse Code Communication model was successfully implemented through VHDL and behavioral simulation using Vivado. By integrating the Morse detector, Morse decoder, UART communication, simulation output logging, and Python for speech synthesis, effective assistive communication implementation was established. Through simulation waveforms, it was verified that the Morse codes were successfully detected, converted into ASCII, transferred via UART serial communication, and finally, generated as speech outputs for communication inputs. It is demonstrated that FPGA technology can be used to establish effective assistive communication through Morse code detection, UART communication, and speech synthesis.

REFERENCES

[1] Anuj Singh, Ishan Tandel, Sanket Salvi, and Bhavana Tiple, "Eye Blink-based Morse Code Communication Tool for ALS Patient," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), IEEE, 2024.

[2] Pravin Kumar S K et al., "Morse Code-Based Assistive Communication System for Individuals with Partial Paralysis," International Conference on Smart Assistive Technologies, 2024.

[3] Ritabrata Roy Choudhury, "Morse Code-Enabled Speech Recognition for Individuals with Visual and Hearing Impairments," Journal of Intelligent Communication Systems, vol. 12, no. 2, 2024.

[4] A. Reddy et al., "FPGA Based Real-Time Communication System," IEEE International Conference on Communication Systems, 2023.

[5] S. Kumar et al., "FPGA Based Assistive Communication using Morse Keypad," International Journal of Embedded FPGA Systems, 2023.

[6] P. Nair et al., "Assistive Communication System using Text-to-Speech Conversion," International Journal of Smart Assistive Systems, 2023.

[7] Boppana Jaswanth Kranthi et al., "Two-Way Communication System using Morse Code for Disabled People," International Journal of Assistive Communication Technologies, 2023.

[8] R. Seetharaman, M. Vignesh, S.S. Sreeja Mole, M. Ramajayam, R.L. Saran, K. Kamalakannan, and K. Anandan, "FPGA Based Morse Code Communicator for Visual and Speech Impaired People using Basys-3," Proceedings of the International Conference on Electronics and Renewable Systems (ICEARS), IEEE, 2022.

[9] N. Tarek, M. A. Mandour, N. El-Madah, R. Ali, S. Yahia, B. Mohamed, D. Mostafa, and S. El-Metwally, "Morse Glasses: An IoT Communication System Based on Morse Code for Users with Speech Impairments," Computing Journal, Springer, vol. 104, no. 4, 2022.

[10] P. Sharma et al., "Python Based Speech Synthesis for Communication Systems," International Journal of Software and Communication Technologies, 2022.

[11] D. Harish et al., "Finite State Machine Based Morse Detection System," International Journal of Advanced FPGA Communication Systems, 2022.

[12] K. Mukherjee et al., "Eye Blink Detection and Conversion to Morse Code," International Journal of Innovative Research in Technology, vol. 8, no. 4, pp. 210–215, 2021.

[13] Ming-Che Hsieh et al., “Adaptive Morse Code Recognition for Physically Disabled Persons,” IEEE Access, vol. 9, pp. 112345–112356, 2021.
[14] M. Reddy et al., “Real-Time UART Communication using FPGA,” International Journal of Scientific Research in Engineering and Management, vol. 5, issue 7, pp. 1–6, 2021.
[15] Xilinx Inc., Vivado Design Suite User Guide, UG910, 2023.
[16] Xilinx Inc., Vivado Design Suite 2022 User Guide, UG973, 2022.