ISSN [Online]: 2583-2654

Implementation of ARINC 429 Controller using VIVADO

[1]*Prof. Tamil Vani R*
Assistant Professor
Department of Electronics and Communication Engineering
Dr. T Thimmaiah Institute of Technology KGF, India

[2]*Jincy P Savyar*
Department of Electronics and Communication Engineering
Dr. T Thimmaiah Institute of Technology KGF, India

[3]*Vishal V*
Department of Electronics and Communication Engineering
Dr. T Thimmaiah Institute of Technology KGF, India

[4]*Dameta Jeesha R*
Department of Electronics and Communication Engineering
Dr. T. Thimmaiah Institute of Technology KGF, India

Abstract: ARINC 429 is a widely used data communication standard in avionics systems, providing a robust and reliable means of exchanging information between aircraft subsystems. The signals obtained from the other electronic equipment in the avionics system should be converted to the ARINC 429 protocol standard. To convert the signals, an controller is designed to handle both transmission and reception of data over ARINC 429 buses, offering capabilities such as data encoding/decoding, error checking, and scheduling of messages. The proposed ARINC 429 controller is implemented using Verilog language and the functionality is verified using tool Vivado 2023.1v. This controller provides a versatile solution for avionics applications, enhancing the interoperability of avionics equipment and contributing to the overall safety and reliability of aircraft systems.

Keywords: ARINC 429,Vivado ,VLSI.

I. INTRODUCTION

ARINC 429 is a communication standard commonly employed in avionics to facilitate data exchange between different electronic systems within an aircraft[1]. It was developed by Aeronautical Radio, Inc. (ARINC) to provide a reliable, straightforward, and standardized method of transferring information within avionics systems. ARINC 429 uses a one-way communication protocol. Data is sent in a unidirectional manner from a single transmitter to multiple receivers. ARINC 429 employs twisted-pair cables and differential signaling for noise immunity. It uses a bipolar Return-to-Zero (RZ) modulation. Two data rates are supported : Low-speed: 12.5 kbps, High-speed: 100 kbps. A parity bit is utilized to detect errors.. ARINC 429 supports continuous or periodic data transmission, with a self-clocking scheme. Signals are transmitted with a voltage differential of ±10V, ensuring robust communication even in noisy environments.

ISSN [Online]: 2583-2654

II DATA FORMAT

ARINC 429 is a standard data communication protocol used in avionics systems to ensure reliable data exchange between aircraft equipment[2][3]. The protocol uses a 32-bit data word format for communication. The first bit is reserved for parity, implementing odd parity for error detection. Bits 2 to 10 represent the Source/Destination Identifier, which identifies the transmitting or intended receiving system. The core data is contained in bits 11 to 29, providing 18 bits of actual information. Bits 30 and 31 form the Sign/Status Matrix (SSM), which offers supplementary status information about the data's validity or condition. Finally, the Label field, encoded in bits 32, is a unique identifier in octal format, categorizing the type of data being transmitted. This structured format ensures clarity, reliability, and efficient communication in avionics systems[4].

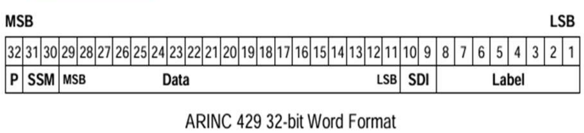

Fig 1 : ARINC 429 32 – bit Data Format

III METHODOLOGY

The above Fig 2 represents the block diagram of ARINC 420 controller where the inputs are of 32 – bits, the controller block consists of

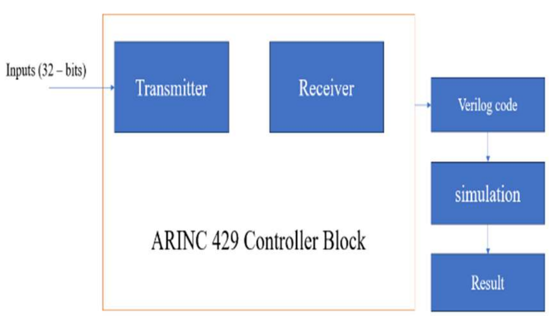

Fig 2 : General Block diagram of Implementation of ARINC 429 Controller

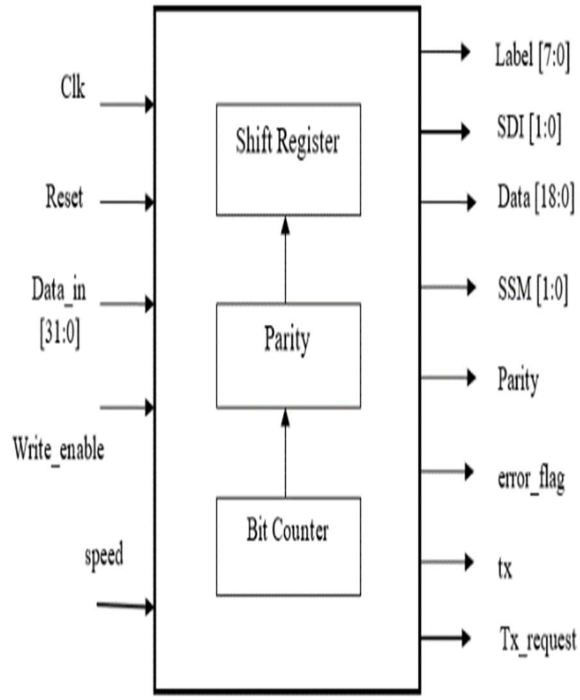

Fig 3 : Transmitter Block of ARINC 429 Controller

Transmitter and Receiver block. The functionality is coded using Verilog language and simulation is verified and results are obtained[5].

The transmitter is initialized using reset input. The transmitter module's shift register transforms the test bench's 32-bit parallel data into serial data. When parity is enabled, the parity bit is also included to the ARINC data. In order to ensure proper data receipt, ARINC uses odd parity as an error check. The Label is transmitted first, that is MSB , followed by the remainder of the bit field, that is LSB, when transmitting data words on the ARINC bus. The Label is always communicated first, followed by the remainder of the ARINC word, in reverse order. This ensures compatibility with legacy systems. The receiving unit is in charge of translating data and rearranging bits into the correct sequence. This is illustrated in Fig 3.

ISSN [Online]: 2583-2654

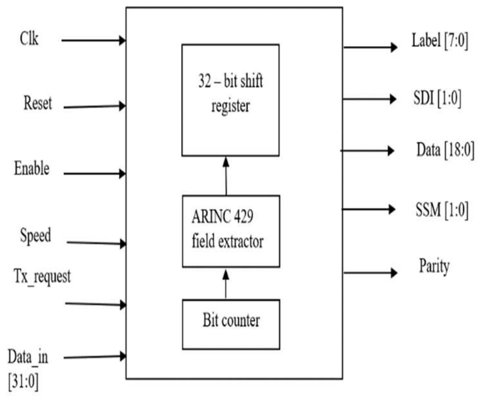

Fig 4 : Receiver Block of ARINC 429 Controller

The receiver listens for messages transmitted over the bus but does not send any information back. This simplifies the communication process and reduces system complexity. The primary function of the ARINC 429 receiver is to receive, decode, and process the 32-bit data words transmitted over the bus. The receiver constantly listens to the differential signals transmitted on the ARINC 429 bus. It is designed to handle signal reception at 100 Hz or 400 Hz baud rates (standard transmission speeds). The receiver processes the 32-bit word by extracting its individual components, with the Label indicating the type of data (e.g., altitude, airspeed, etc.).

The Data part contains the actual value being transmitted (e.g., numerical data like the airspeed).The Parity bit is checked to ensure that no errors have occurred during transmission. The ARINC 429 receiver includes error-detection capabilities to ensure data integrity. The parity bit is used to check whether an error has occurred during transmission. If a received word does not match the expected label or format, the receiver can discard or flag the word as invalid. Errors such as bit errors, framing errors, and sync errors can be detected by the receiver. If an error is detected, it may flag the data or request a retransmission. The receiver typically only processes data that matches its expected label. Each system on the ARINC 429 bus can have a different label associated with the data it sends. For instance, one receiver might only accept data with a label of "22" (airspeed), while another might accept a label of "19" (altitude). As shown in Fig 4.

IV IMPLEMENTATION

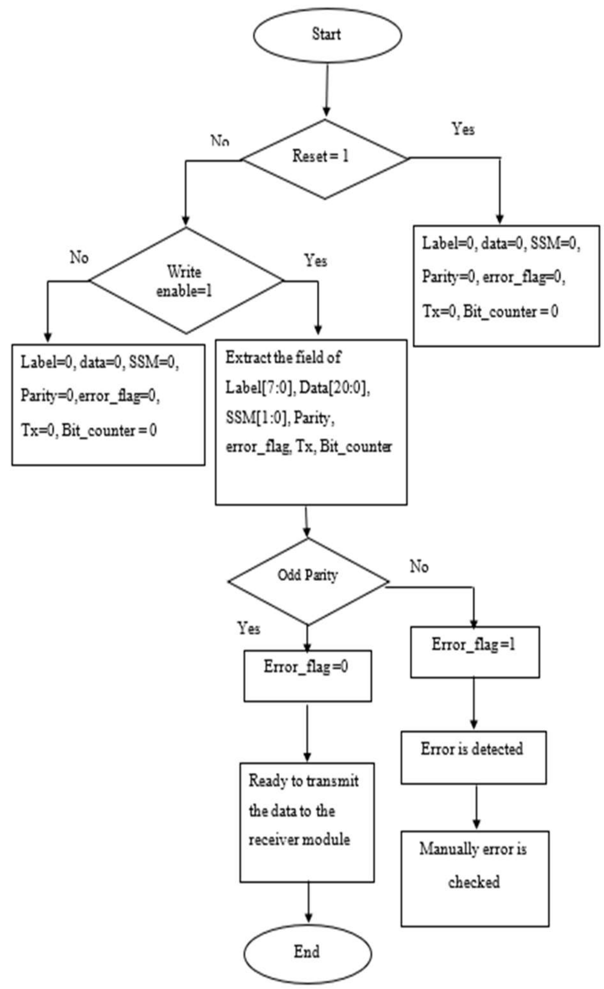

Fig 5 : Flow chart of ARINC 429 Transmitter

The provided flow chart as shown in Fig 5 outlines the operational sequence of a transmitter module. It commences by ascertaining whether a reset condition exists, and if so, it sets the reset flag to 1. Subsequently, it initializes several variables:

ISSN [Online]: 2583-2654

Label, data, SSM (Sub-System Mode), Parity, error flag, and bit counter, assigning them initial values of 0. The module then proceeds to extract the 8-bit Label field from the incoming data. Following this, it extracts the 21-bit Data field, the 2-bit SSM field, the Parity bit, and the Error Flag from the incoming data stream. A parity check is then conducted to assess the data's integrity. If the Parity bit is 0, it indicates no error, and the error flag is set to 0. Conversely, if the Parity bit is 1, it signifies an error, and the error flag is set to 1.

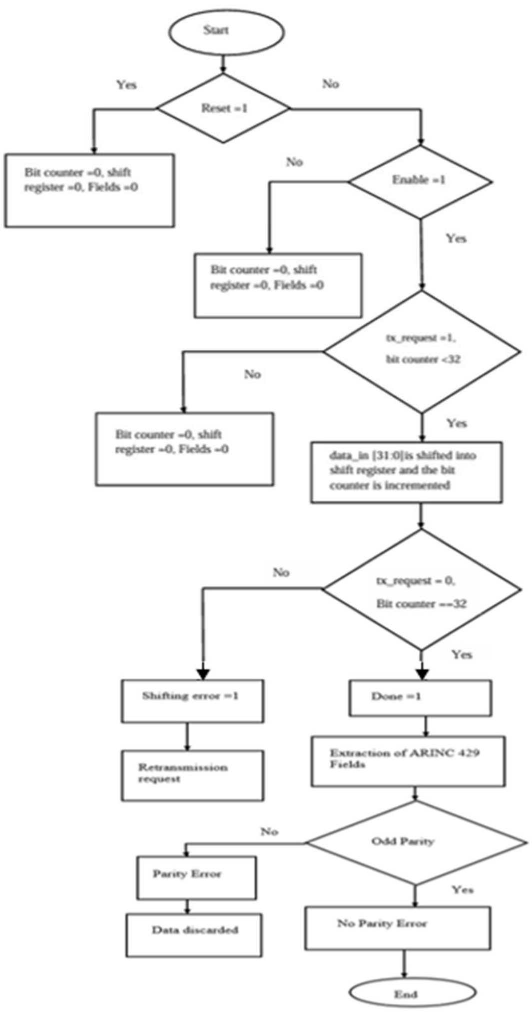

Fig 6 : Flow chart of ARINC 429 Transmitter

The ARINC 429 receiver circuit listens to the data bus (typically 32 bits) and acquires the incoming bits one at a time. The 32-bit shift register receives the bits in serial fashion. The process involves shifting each bit of the data in the register as new bits are received. For each complete 32-bit transmission After the 32nd bit is received, the full 32-bit word is stored in the shift register. Once the shift register has captured a full 32-bit word (after the shift process), As shown in Fig 6. It ensures that words are processed sequentially. The FIFO buffer may be designed to have an overflow mechanism, where new words are discarded or flagged as errors if the buffer is full. Once a 32-bit word is stored in the FIFO, the ARINC 429 field extractor will parse the word and ARINC 429 uses odd parity, meaning the total number of 1's in the word (including the parity bit) should be odd. The field extractor checks the parity of the extracted word by counting the number of 1's in the 32-bit word, including the parity bit. If the parity check fails (even number of 1's), the word is flagged as an error.

V RESULTS

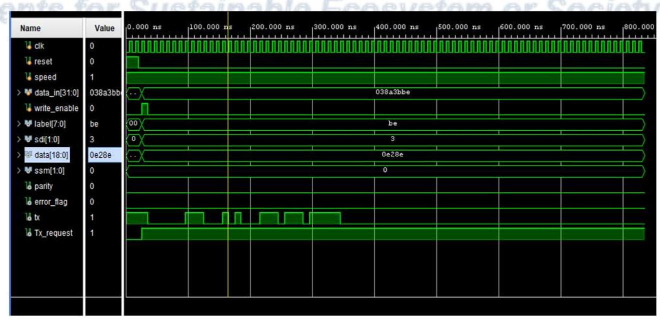

Fig 7 : Simulation of ARINC 429 Transmitter without parity and error_flag

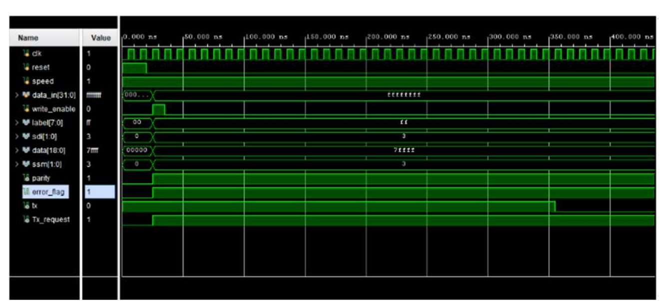

Fig 8 : Simulation of ARINC 429 Transmitter with parity and error_flag

ISSN [Online]: 2583-2654

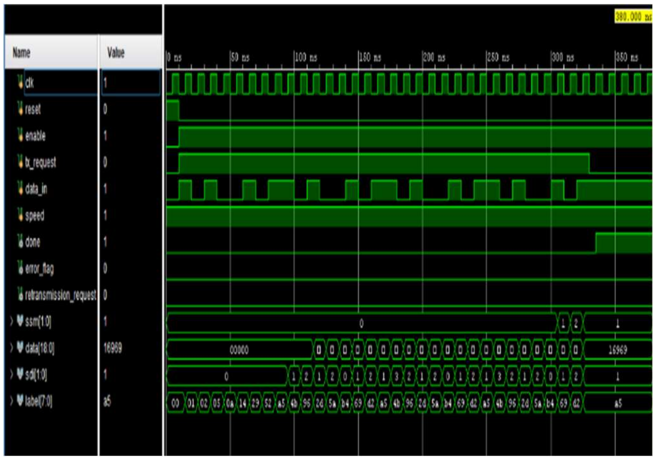

Fig 9 : Simulation of ARINC 429 Receiver with Parity error

Fig 10 : Simulation of ARINC 429 Receiver without parity error

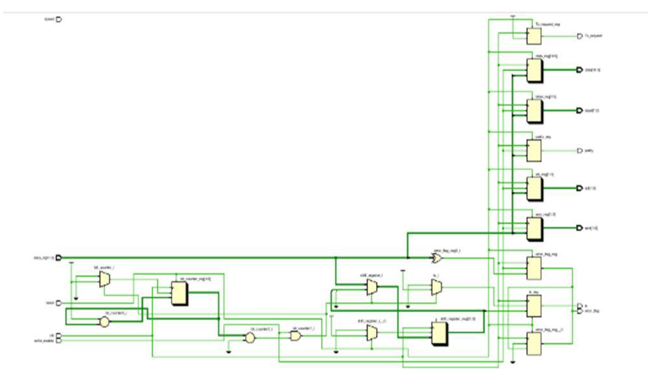

Fig 11 : RTL Schematic of ARINC 429 Transmitter

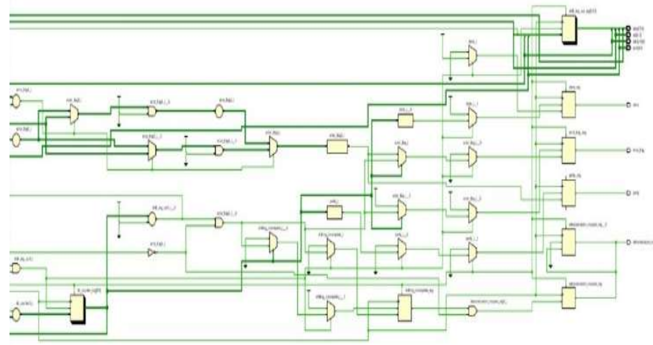

Fig 12 : RTL Schematic of ARINC 429 Receiver

VI. CONCLUSION

The ARINC 429 receiver is a vital part of the avionics systems used in modern aircraft, ensuring that critical flight data is received, decoded, and processed accurately. It plays a key role in maintaining the integrity and performance of aviation systems by ensuring reliable data transmission, error detection, and synchronization. Through its robust error-checking and data validation capabilities, it supports safe and efficient aircraft operations in the highly demanding aerospace environment.

REFERENCES

*[1] D. Mukherjee, N. Kumar, K. Singh, H. Mondal, B. V. R. Reddy, **"Implementation of ARINC 429 16 Channel Transmitter Controller on FPGA"**, Communications in Computer and Information Science, vol 147, Springer, Berlin, January, 2011.*

*[2] L. Hao G. Cao ,M. Shi-guo, L. Xin , H. Gang, **"ARINC - 429 airborne communications transceiver system based on FPGA implementation"**, International Conference on Information Sciences, Machinery, Materials and Energy, 2015.*

*[3] I. Tamrakar, C. M. Ananda, B. P. Kumar, P. S. Ramani, **"Design and Implementation of Ethernet based Analyzer for Avionics Serial Bus on FPGA"**, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, India, May 2016.*

*[4] R. Vyas, A. Bhattacharjee, A. Khandelwal, **"Development of SoC for Interfacing Avionics Units with ARINC 429 Bus",** International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, October 2016.*

*[5] Prof. M. Kamaraju, Dr A. V. N. Tilak, Dr K. Lalkishore, K. Bahurao **"VHDL Implementation and Verification of ARINC 429 Core",** Gudlavalleru Engineering College, Gudlavalleru, JNTU, Hyderabad.*