DESIGN AND DEVELOPMENT OF ARINC 429 PROTOCOL

1SMITHA M R, 2APARNA

1M.Tech, VLSI design and embedded systems, Dept of ECE, 2Engineer, MCSRDC
1Sathagiri College of engineering Bangalore, INDIA.
2Hindustan Aeronautics Limited, Bangalore, INDIA
E-mail: 1smithamr01@gmail.com; 2acinthalkar@gmail.com

Abstract – ARINC is the major company that develops and operates systems and services to ensure the efficiency, operation and performance of aviation and travel industries. ARINC 429 specification defines how the avionics equipment and systems should communicate with each other. It employs a unidirectional data bus standard known as Mark 33 Digital Information Transfer Systems. ARINC 429 bus operates at 12.5 or 100 kilobits per second. The top level architecture of the ARINC 429 protocol has 4 modules ARINC 429 transmitter, ARINC 429 receiver, Transmit and Receive 512*32-bit FIFO to store and fetch the data and ARINC 429 clock generation. In this proposed work ARINC 429 protocol is designed and developed using Hardware Descriptive Language (HDL). The continuous data stream is stored in a 512*32-bit FIFO. The encoder module at ARINC 429 transmitter converts a continuous data stream into Bipolar Return to Zero format. This encoded data is decoded at the receiver. The decoded data is stored in the ARINC 429 Receive FIFO. The design and development is done using Xilinx software.

Keywords— ARINC 429, DITS, BPRZ encoding, FIFO, VHDL

I. INTRODUCTION

Aeronautical Radio, Incorporated is a major company that develops and operates systems and services to ensure the efficiency, operation and performance of the aviation and travel industries. It was organized in 1929 by four major airlines to provide a single license and initiator of radio communications outside the government. Only airlines and aviation related companies can be shareholders, although all airlines and aircraft can use ARINC’s services. It is now a $280 million company with headquarters in Annapolis, Maryland and over 50 operating locations worldwide. ARINC has provided leadership in developing specifications and standards for avionics equipment and these specifications are used to define physical packaging and mounting of avionics equipment, data communications standards and High level computer languages.

A. ARINC STANDARD SERIES

ARINC has number of series like 400, 500, 600, 700 and each series has its own advantages and used for different avionics equipment’s. For example ARINC 400 Series describes guidelines for installation, wiring, data buses and databases. ARINC 500 Series describes older analog avionics equipment used on early jet aircraft such as the Boeing 727, Douglas DC-9, DC-10, Boeing 737, 747 and Airbus A300. ARINC 600 is the predominant avionics packaging standard introducing the avionics Modular Concept Unit. ARINC 700 series describes the form, fit and function of avionics equipment installed predominately on transport category aircraft.

B. ARINC 429

ARINC 429 specification defines the standard requirements for the transfer of digital data between avionics systems on commercial aircraft. ARINC 429 employs unidirectional data bus standard known as the MARK 33 Digital Information Transfer System (DITS) specification. This is a very simple, point-to-point protocol. There can be only one transmitter on a twisted wire pair. The transmitter is always transmitting either 32-bit data words or the NULL state. The most messages consist of a single data word. There is at least one receiver on a wire pair, there may be up to 20.

C. FEATURES OF ARINC 429

i. It has selectable data rates on each channel as 12.5kbps or 100 kbps
ii. It provides direct cpu access to memory.
iii. It has configurable label memory up to 256 words.
iv. It has selectable clock speed of 1, 10, 16 or 20 Mhz.
v. It has programmable FIFO depth up to 512 words.
vi. It uses Bi-polar return to zero encoding and decoding methods.
vii. Transmission and reception is on separate ports.

D. ARINC 429 WORD FORMAT

ARINC 429 data words are always 32 bits and typically use the format shown in Figure 1 which includes five primary fields namely Parity, SSM, Data, SDI, and Label. ARINC 429 convention numbers the bits from 1 (LSB) to 32 (MSB).

Figure 1: Generalised ARINC 429 word format
- Transmission order
  The order of the bits transmitted on the ARINC 429 bus is as follows: 8, 7, 6, 5, 4, 3, 2, 1, 9, 10, 11, 12, 13 … 32. Here the least significant bit of each byte, except the label is transmitted first and the label is transmitted ahead of the data in each case.

- Parity
  The MSB 32 is always the parity bit for ARINC 429. Parity is normally set to odd except for certain tests. Odd parity means that there must be an odd number of 1 bits in the 32-bit word that is insured by either setting or clearing the parity bit.

- SSM
  Bits 31 and 30 contain the Sign/Status Matrix. This field contains hardware equipment condition, operational mode or validity of data content.

- Data
  Bits 29 through 11 contain data which may be in a number of different formats. There are also many non-standard formats that have been implemented by various manufacturers. In some cases, the data field overlaps down into the SDI bits. In this case, the SDI field is not used.

- SDI
  Bits 10 and 9 provide a Source/Destination Identifier. This is used for multiple receivers to identify the receiver for which the data is destined. It can also be used in the case of multiple systems to identify the source of the transmission. In some cases, these bits are used for data. ARINC 429 can have only one transmitter on a pair of wires, but up to 20 receivers.

- Label
  Bits 8 through 1 contain a label identifying the data type and the parameters associated with it. The label is an important part of the message. It is used to determine the data type of the remainder of the word and therefore the method of data translation to use. Labels are typically represented as octal numbers.

II. OPERATIONS OF ARINC 429
An ARINC 429 data bus uses two signal wires to transmit 32 bit words. Transmission of sequential words is separated by at least 4 bit times of NULL (zero voltage). This eliminates the need for a separate clock signal wire. That’s why this signal is known as a self-clocking signal. The nominal transmission voltage is 10 ±1 volts between wires (differential), with either a positive or negative polarity. Therefore, each signal leg ranges between +5V and -5V. If one leg is +5V the other is -5V and vice-versa. One wire is called the “A” (or “+” or “HI”) side and the other is the “B” (or “-“ or “LO”) side. This is known as bipolar return-to zero (BPRZ) modulation as shown in figure 2. The composite signal state may be one of three levels:
- HI this should measure between 7.25 and 11 volts between the two wires (A to B).
- NULL this should be between 0.5 to -0.5 (A to B).
- LO this should be between -7.25 to -11 volts (A to B).

The received voltage depends on line length and the number of receivers connected to the bus. No more than 20 receivers should be connected to a single bus and no less than one receiver. Since each bus is unidirectional, a system needs to have its own transmit bus if it is required to respond or to send messages. The transmitting and receiving circuits must be designed for reliably sending and detecting the null transition between high and low states.

Figure 2: ARINC 429 Bipolar Return to zero format

A. FIRST PHASE OF ARINC 429
The core consists of three main blocks: Transmit, Receive and CPU Interface. Core 429 requires connection to an external CPU. The CPU interface configures transmit and receive control registers and initializes the label memory. The core interfaces to the ARINC 429 bus through an external ARINC 429 line driver and line receiver as shown in figure 3.

Figure 3: Typical core system

Core 429 provides a complete and flexible interface to a microprocessor and an ARINC 429 data bus. Connection to an ARINC 429 data bus requires additional line drivers and line receivers. Core 429 interfaces to a processor through the internal memory of the receiver. Core 429 can be easily interfaced to an 8, 16 or 32-bit data bus. Look-up tables loaded into memory enable the Core 429 receive circuitry to
filter and sort incoming data by label and destination bits. Core 429 supports multiple ARINC 429 receiver channels and each receives data independently. The receiver data rates (high or low speed) can be programmed independently. Core429 can decode and sort data based on the ARINC 429 Label and SDI bits and stores it in FIFO. Each receiver uses programmable FIFO to buffer received data.

**B. SECOND PHASE OF ARINC 429**

The Transmitter module converts the 32-bit parallel data from the TX FIFO to serial data. It also inserts the parity bit into the ARINC 429 data when parity is enabled. The CPU interface is used to fill the FIFO with ARINC 429 data. The TX FIFO can hold up to 512 ARINC 429 words of data. The transmission starts as soon as one complete ARINC 429 word has been stored in the transmit FIFO. The TX module contains two 8-bit registers. One is used for a control function and the other is used for status. The CPU interface allows the system CPU to access the control and status registers within the core. Depending on the TX FIFO status signals, the CPU will either read the TX FIFO before it overflows or not attempt to read the TX FIFO if it is empty.

**C. FINAL PHASE OF ARINC 429**

The Receiver block is responsible for recovering the clock from the input serial data and performs serial-to-parallel conversion and gap/parity check on the incoming data. It also interfaces with the CPU. The Rx module contains two 8-bit registers. One is used for control function and the other is used for status. The CPU interface configures the internal RAM with the labels, which are used to compare against the incoming labels from the received ARINC 429 data. If the label-compare bit in the receive control register is enabled, then the data which matches its labels with the stored labels will be stored in the RX FIFO. If the label compare bit in the receive control register is disabled, then the incoming data will be stored in the RX FIFO without comparing against the labels in RAM. The core supports reloading label memory using bit 7 of the Rx control register. Depending on the RX FIFO status signals, the CPU will either read the RX FIFO before it overflows or not attempt to read the RX FIFO if it is empty.

**III. TESTING METHODOLOGY OF ARINC 429**

There are two types of testing ARINC 429 protocol such as loopback test and individual test. This 32-bit data is encoded in Bi-polar Return to zero format. The encoded data is transmitted by ARINC 429 encoder and transmit it to the receiver. The ARINC 429 decoder decodes the data and stores the data in 512*32 receive FIFO. The data from the receive FIFO is looped back to the interface between input data and transmit FIFO to check if the data received is same as transmitted data.

**A. LOOPBACK TEST**

In loopback test the input data is taken from HOST and the input data is stored in 512*32 transmit FIFO as 32-bit data.

**B. INDIVIDUAL TEST**

The individual test of ARINC 429 consists of Transmitter test and Receiver test. The ARINC 429 transmitter consists of four blocks: Transmit FIFO with the depth of 512 words, Parallel to serial register to convert the 32-bit data to a single bit for encoding, BPRZ encoder which encodes the serial data into waveform, clock generation which generates a 200 KHz. Transmit FIFO takes the input from a host as shown in figure 5.

**III. TESTING METHODOLOGY OF ARINC 429**

There are two types of testing ARINC 429 protocol such as loopback test and individual test. This 32-bit data is encoded in Bi-polar Return to zero format. The encoded data is transmitted by ARINC 429 encoder and transmit it to the receiver. The ARINC 429 decoder decodes the data and stores the data in 512*32 receive FIFO. The data from the receive FIFO is looped back to the interface between input data and transmit FIFO to check if the data received is same as transmitted data.

**A. LOOPBACK TEST**

In loopback test the input data is taken from HOST and the input data is stored in 512*32 transmit FIFO as 32-bit data.

**B. INDIVIDUAL TEST**

The individual test of ARINC 429 consists of Transmitter test and Receiver test. The ARINC 429 transmitter consists of four blocks: Transmit FIFO with the depth of 512 words, Parallel to serial register to convert the 32-bit data to a single bit for encoding, BPRZ encoder which encodes the serial data into waveform, clock generation which generates a 200 KHz. Transmit FIFO takes the input from a host as shown in figure 5.

The ARINC 429 receiver consists of BPRZ decoder, 32-bit shift register and 512*32 bit RX FIFO. The DATA_APLUS and DATA_AMINUS are the ARINC 429 transmitted data which is the input to the decoder. The clock is recovered and the BPRZ decoder recovers the original data. The recovered data is sent as a 32-bit parallel data to receive FIFO. The data from the receive FIFO is read by the host as shown in figure 6.
IV. IMPLEMENTATION OF ARINC 429

A. Top level block diagram of ARINC 429

The top level block diagram of ARINC 429 is as shown in Figure 7. The ARINC test bench provides the input data to the ARINC 429 module. It acts as a microcontroller. The test bench inputs are the addresses and data. The ARINC clock generation takes the clock provided by the microcontroller 100/50 MHz in this case and generates a ARINC 429 clock of 200 KHz which is used by the transmitter and the receiver.

The TX FIFO takes the input from the test bench and ARINC clock generation. The input from the test bench is buffered and stored in 512*32 FIFO as 32-bit data. This 32-bit data is taken serially and encoded into BPRZ format. The encoded data is transmitted by the ARINC 429 transmitter.

The ARINC 429 receiver decodes the data and stores the data in 512*32 RXFIFO. The data from the RXFIFO is looped back to the host to check if the data is received correctly. Host interface interfaces the ARINC 429 module to the microcontroller.

B. Top level Finite State Machine for ARINC 429 IP core

The Finite State Machine for ARINC 429 is as shown in the Figure 8. There are five states: Reset state, idle state (s1), configuration state, transmit state, and receive state. The events for the FSM are reset, config, START_TX, STOP_TX, START_RX, STOP_RX. The working of the FSM is as stated below:

When reset event occurs, reset state transitions to idle state or to reset state itself depending upon the event. If reset is 1, the state transitions to itself or if reset is 0, the state transitions to idle state. When configuration event is set to 1 the state transitions from idle state to configuration state. The transition from the configuration state to transmit state or receive state occurs depending on which of the event occurred. If START_TX event had occurred, the configuration state transitions to transmit state, else if START_RX had occurred, the transition will be to receive state. In transmit or receive state the config will be set to 0. During no transmission or reception of data the STOP_TX or STOP_RX event goes to 1 and the current state transitions to the idle state. The process repeats when the config event is set to 1.

V. SIMULATION RESULTS

Figure 9 Simulated output for the data given from the test bench to transmitter FIFO (waveform in red color)

Figure 10: Simulated output for the data stored in Transmit FIFO (waveform in white color)
CONCLUSION

This significant ARINC 429 protocol is designed, developed and simulated in VHSIC Hardware Description Language which is used to provide standard communication between avionics equipment. Bipolar Return to zero encoding method eliminates the need of extra bits for synchronization by self-clocking. This protocol is developed using Xilinx 14.2 version. This can be further implement on smart fusion evaluation kit.

REFERENCES

[3] https://www.actel.com

★★★