Development and Testing of ARINC429 IP using FPGA for Navigation System

[1]Sri J.Balakrishna, [2]K. Suryateja, [3]Dr. N. Alivelu Manga

[1]*Assistant Professor, Department of ECE, Chaitanya Bharathi Institute of Technology (A), Hyderabad.*

[2]*ME-ESVD, Department of ECE, Chaitanya Bharathi Institute of Technology (A), Hyderabad, India.*

[3]*Associate Professor, Department of ECE, Chaitanya Bharathi Institute of Technology (A), Hyderabad.*

***Abstract*—** ARINC 429 is an Aeronautical Radio INCorporated (ARINC) protocol used data bus for Defence Aircraft, commercial, helicopters, and Military Aircraft. It defines the process of communication among various modules of avionics. This paper presents the design and development of required individual blocks (Serial In Parallel Out, Parallel In Serial Out, FIFO Tx, FIFO Rx, and memory) of ARINC429IP using Verilog HDL further integrate all the blocks as a single module and verify the results by programming the module in the RCI developed Navigation board using ChiproScope Pro and CRO.

Keywords*— *ARINC429, DDC

I. Introduction

The ARINC specification defines the electrical and data characteristics and protocols, Data words are 32 bits in length and most messages consists of a single data word. Messages are transmitted at either 12.5 or 100 Kbit/s to other system elements that are monitoring the bus messages. ARINC 429 has been installed on most commercial transport aircraft including; Airbus A310/A320 and A330/A340; Bell Helicopters; Boeing 727, 737, 747, 757, and 767; and McDonnell Douglas MD-11.

Data can be transmitted in one direction only – simplex communication – with bi-directional transmission requiring two channels or buses.

II. BLOCK DIAGRAM

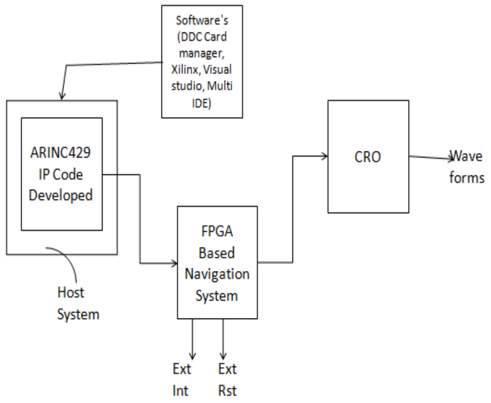
Software's (DDC Card manager, Xilinx, Visual studio, Multi IDE)
ARINC429 IP Code Developed
Host System
FPGA Based Navigation System
CRO
Wave forms
Ext Int
Ext Rst

Fig. 1.1: Generalized Block Diagram

First Required software is to get installed in the host system, and the code is developed using the corresponding EDA tool and after developing the code, the code is dumped in to the navigation system which it contains external interrupt pin and external reset pin and after that CRO is connected with CRO probes and place that probes on the test points of the board to get the results.

III. Data format

ARINC data words are 32 bits. This includes five primary fields, namely Parity, SSM, Data, SDI, and Label. ARINC convention numbers the bits from 1 (LSB) to 32 (MSB). The following table is showing the word format.

TABLE I. ARINC WORD FORMAT

32	31 30	29 11	10 9	8 1
P	SSM	MSB DATA LSB	SDI	LABEL

Parity (P)

The MSB is always the parity bit for ARINC 429. Odd parity means that there must be an odd number of “1” bits in the 32-bit word. For even parity 1-31 contain an even number of ‘1’ bits.

Sign Status Matrix (SSM)

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

TABLE II. SSM Code for BCD data

Bit		**Meaning**
0	0	Plus, North, East, Right, To, Above
0	1	No Computed Data
1	0	Functional Test
1	1	Minus, South, West, Left, From

Data

Bits 29 through 11 contain the data, which may be in a number of different formats.

Source Destination Identifier (SDI)

Bits 10 and 9 provide a Source/Destination Identifier or SDI. This is used for multiple receivers to identify the receiver for which the data is destined.

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.

TRANSMISSION FLOW

The least significant bit (LSB) of each byte except the label is transmitted first, and the label is transmitted ahead of the data in each case. When a 32-bit ARINC word is transmitted on the bus, in the case of the label, the most significant bit (MSB) is transmitted first and next LSB is transmitted. But for the remaining fields LSB is transmitted first and next MSB is transmitted. Transmission flow of all 32 bits as of order 8,7,6,5,4,3,2,1,9,10,11,12,13….32.

IV. DDC USB AVIONICS MODULE DESCRIPTION

DDC stands for Data Device Corporation.

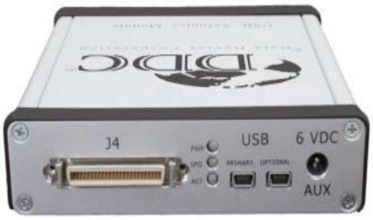

Fig. 4.1: USB port facing of the DDC module

4.1 Main features of the module

The following are the features of the BU-67103ux:

i. Portable USB 2.0 Device.

ii. 4 ARINC429 Receive channels.

iii. 2 ARINC429 transmit channels.

iv. 8 user programmable digital discrete I/O's.

v. 1 Pulse per Second Output.

vi. 1 MB RAM with Parity per 1553 Channel.

vii. 48-bit / 100ns Time Stamp.

viii. Extended Temperature Hardware Available.

ix. Multiple Configuration Options: ARINC429 only, MIL-STD-1553 only, ARINC429 & MIL-STD-1553.

Included Software: - High-Level C Software Development Kits (SDK) and Drivers-Common Test/Embedded API.

4.2 System Requirements

i. USB 2.0 Port supporting 5 Volts–Limited Performance on USB 1.1.

ii. Supported Operating System: Windows® 2000/XP/Vista/7 or Linux 2.6.x.

After opening the DDC card manager the opening screen looks as shown in below.

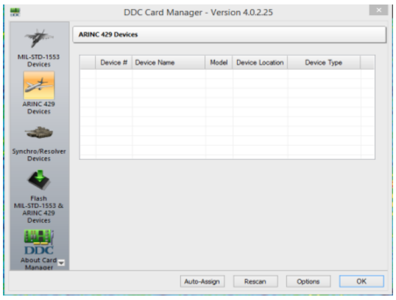

Fig 4.2: DDC Card manager software opening screen

V. MULTI INTEGRATED DEVELOPMENT ENVIRONMENT

The MULTI Integrated Development Environment (IDE) is a product launched by the GREEN HILLS company. MULTI is a complete Integrated Development Environment (IDE) designed especially for embedded systems engineers to assist them in analyzing, editing, compiling, optimizing, and debugging embedded applications.

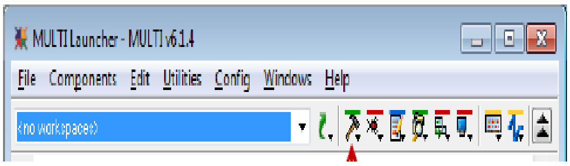

Fig. 5.1: Launcher screen

VI. RESULTS

Clock Division: Generating a 20 MHz clock signal from the 100MHz clock as operating is on 20 MHz clock frequency. Based on the clock frequency the system works.

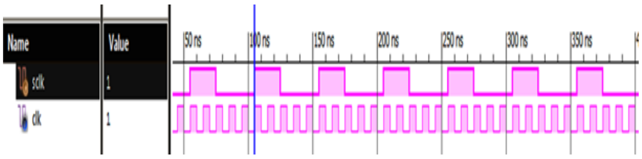

Fig. 6.1: Generation of clock signal of 20 MHz

Parallel In Serial Out:

Generating a parallel in serial out to transfer the data from parallel type data to serial type data. This helps in the conversion of the data in the interface stage.

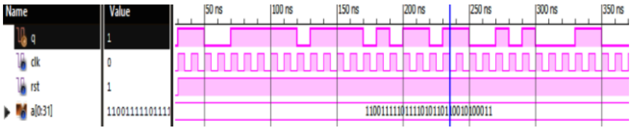

Fig. 6.2: Parallel In Serial Out Of 32bit

Serial In Parallel Out: Generates a serial in parallel out for conversion of serial data to parallel data, which helps in the conversion of data.

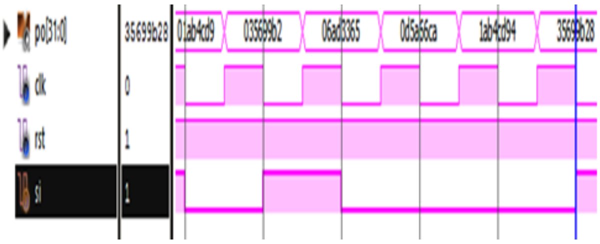

Fig. 6.3: Serial In Parallel Out of 32bit

FIFO Logic: Performing FIFO Logic for the transmission of the data in the FIFO Logic format. i.e. First in First out Logic.

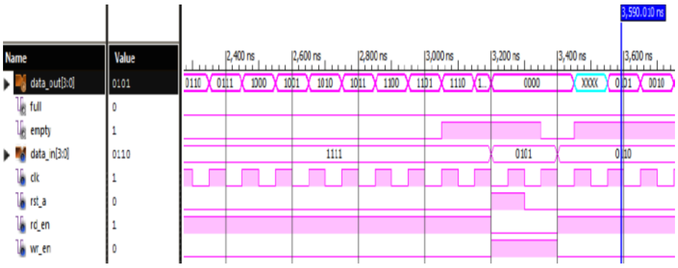

Fig. 6.4: FIFO logic

After developing the individual blocks and making it as a single code that work as ARINC429 logic. Later dump the code by generating mcs and bit file.

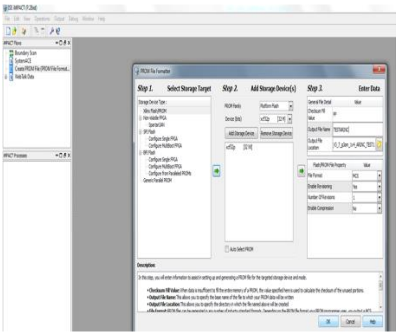

Fig. 6.5: Generating mcs file

After generation of mcs file, load the code in FPGA by giving required details. After dumping of the code, it shows as program success.

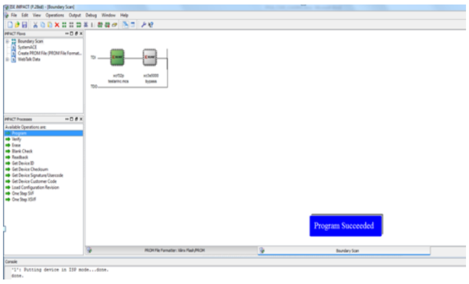

Fig. 6.6: FPGA dumping

chipscope resultant for data given (2-2-2-2-2-2-8-8)

data to come (1000-1000-0100-0100-0100-0100-0100-0100)

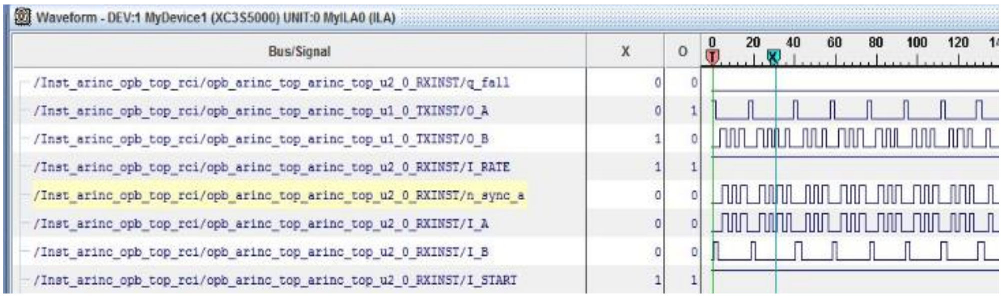
Waveform - DEV:1 MyDevice1 (XC3S5000) UNIT:0 MyILA0 (ILA)
Bus/Signal
X
O
0
20
40
60
80
100
120
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/q_fall
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u1_0_TXINST/O_A
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u1_0_TXINST/O_B
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/I_RATE
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/n_sync_a
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/I_A
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/I_B
/Inst_arinc_opb_top_rci/opb_arinc_top_arinc_top_u2_0_RXINST/I_START

Fig. 6.7: chip scope resultant wave

CRO resultant wave form for data (B-C-C-C-C-C-3-3) Here in Data4 the Label field data is 1-1(in binary 0001-0001), in this the data from MSB to LSB is 0001-0001 and for remaining the data is 9-1-1-1-1-1(1001-0001-00010001-0001-0001) in this the data from LSB to MSB is (1000-1000-1000-1000-1000-1001).

Finally the result to get is (0001-0001-1000-1000-1000-1000-1000-1001).

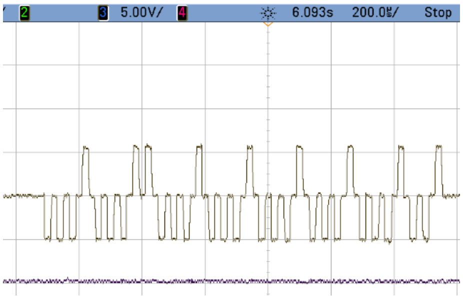

Fig. 6.8 CRO resultant wave form

VII. CONCLUSION

The ARINC429 is developed by developing the individual blocks using Xilinx software and tested the simulated results using ChipScope Pro for different input data by generating mcs file and bit file. To check another set of results of different data on CRO, interface is made between board and the system with help of DDC module and debugger and again dumping of the code is done by generation of mcs and bit files. After that the data is sent and the board is responding according to the ARINC429 signal word format. By placing the probes on the test points the test results are observed with different values and are according to the ARINC429 signal format.

REFERENCES

[1] YunlinLuo , XiaoqingZhao, "Design and Research of an ARINC 429 Bus Testing System", 26th Chinese Control and Decision Conference (CCDC), 2014.

[2] ARINC 429 specification tutorial, AIM GmbH-Avionics Data Bus Solutions, Nov 2010.

[3] An ARINC Inc., Document, and ARINC Specification 429 Part 1-17.

[4] Green Hills Software, Inc., Green Hills Multi Debugging.

[5] Zhang, J. G., Pervez, A., and Sharma, A. B. "Avionics data buses: An overview", IEEE Aerospace and Electronic Systems Magazine, pp.18-22, Mar 2003.

[6] ISE Design suite 14: Release notes, UG631 (v14.7) October 2, 2013.