IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 3, June-July, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Design and Implementation of Mil-Std-1553B Bus Protocol Controller with FPGA and ASIC

Sharon Theresa George[1], J. Mangaiyarkarasi[2]

**[1,2]Electronics and Communication, Anna University Regional Office,
Madurai, 625002, India**

Abstract

MIL-STD-1553B bus is widely used in military and space missions nowadays for its high reliability and efficiency. 1553B is a dual- redundant, bi-directional, Manchester coded, and digital time division command/response data bus .The trade-off between size and power in implementing 1553B bus systems limit its applications in small satellites. The design of 1553B bus protocol controller and its information transfer is completed in VHDL language on ModelSim. It is then realized using FPGA and ASIC approaches. This paper proposes the MIL-STD-1553 bus controller implementation onto a Xilinx based FPGA platform and development of physical design of protocol controller is done using cadence SoC encounter RTL compiler**.** Performance evaluation of the physical design with respect to area and power has been done and compared with FPGA based approach. The core consumes 1.7 mW of power for the core area of 14503 µm2.

Keywords: Mil-std, FPGA, BC, ASIC, SoC, RTL, Manchester coding, Physical Design

1. Introduction

MIL-STD-1553B data bus originated from the need of an infrastructure to develop systems that have components distributed over the various parts of the aircrafts. Military aircraft before 1970s were developed by integrating various avionics subsystems by direct point-to-point wires. Inserting new subsystems or removing existing ones posed great problems because of this direct wiring. As more and more subsystems were added the aircrafts became more complex and the overall weight increased. The need for a networking standard emerged from these problems. To overcome these difficulties, The US Department of Defense published the 1553 serial data bus standard and chose multiplexing because of the advantages like weight reduction, simplicity of system design, standardisation, and flexibility. The primary purpose of this data bus is to move data between black boxes. How these boxes are connected and the methodology with which the communication is accomplished is central to the operation of the data bus. The MIL-STD-1553B bus features a time division multiplexing, half duplex command/response protocol which runs on a single twisted shielded pair of wires. Only a single computer is allowed to transmit on the bus at a given time. On the other hand, full duplex systems like Ethernet and RS-232/422 have multiple cables to provide simultaneous transmit and receive. The data rate on a 1553 bus is 1 Mhz. Fault tolerance is supported by dual redundancy.

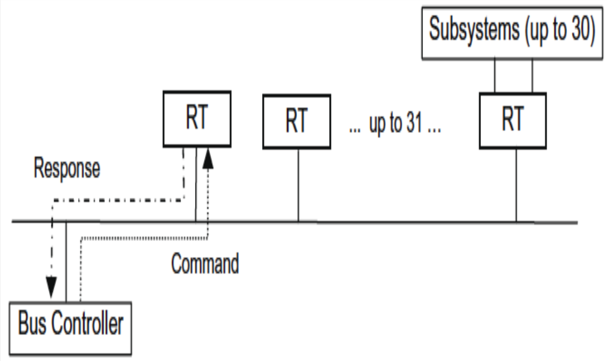


Fig. 1 Structure of Mil-Std 1553B Data Bus

1553 bus system architecture consists of multiple computers having a master/slave relationship. The computer that acts as the master and controls all the communication on the bus is called Bus Controller (BC). The BC can control multiple slave computers which are called Remote Terminals (RT) by sending commands to them. There may also be one or more passive Bus Monitors (BM) deployed on the bus which are only used to monitor or record the messages on the bus but can’t transmit any messages. There can be thirty-one remote terminals connected to the bus as in figure 1.3 in addition to the bus controller. Remote terminals receive commands from the bus controller and respond according those the commands. Only the bus controller can initiate a transmission on the bus.

1.1 Bus Controller

The main function of the bus controller (BC) is to provide data flow control for all transmissions on the bus. In addition to initiating all data transfers, the BC must transmit, receive and coordinate the transfer of information on the data bus. All information is communicated in command/response mode - the BC sends a command to the RTs, which reply with a response. The bus controller, according to MIL-STD-1553B, is the “key part of the data bus system” and “the sole control of information transmission on the bus shall reside with the bus controller, which shall initiate all transmission”. The bus can support multiple BCs, but only one can be active at a time. Normal BC data flow control includes transmitting commands to RTs at predetermined time intervals. The commands may include data or requests for data (including status) from RTs. The BC has control to modify the flow of bus data based on changes in the operating environment. These changes could be a result of an air-to-ground attack mode changing to air-to-air, or the failure mode of a hydraulic system. The BC is responsible for detecting these changes and initiating action to counter them. Error detection may require the BC to attempt communications to the RT on the redundant, backup bus.

1.2 Message Formats

The MIL-STD-1553B standard strictly defines six information transfer formats and four broadcast information transfer formats. No other message formats are allowed to be used on the data bus. Figure 1.11 shows the six message formats and Figure 1.12 shows the remaining four broadcast message formats defined by the standard. The six message formats are allowed between the bus controller and a remote terminal. Bus controller to remote terminal transfers are triggered by a receive command which is immediately followed by some data words without any gaps in between. The number of data words is specified in the command word. The remote terminal then transmits a status word back to the controller. Remote terminal to bus controller transfers are triggered by a transmit command from the bus controller. The remote terminal then responds with a status word which is immediately followed by some data words without any gaps between them. Remote terminal to remote terminal transfers are triggered by a receive and transmit commands which are send from the bus controller without any gaps in between. The receive command is addressed to the remote terminal which will listen for the data, and the transmit command is addressed to the remote terminal that will send the data. The remote terminal that receives the transmit command sends a status word which is immediately followed by some data words without any gaps between them. Finally, the listening remote terminal which received the data sends a status word. Mode command without data word transfers are triggered by a mode command and finalised by a status word from the remote terminal. Mode command with data word (transmit) transfers are triggered by a transmit command that contains a mode code. The remote terminal responds with a status word and a single data word without any gaps in between. Mode command with data word (receive) transfers are triggered by a receive command that contains a mode code immediately followed by a single data word without any gaps in between. The remote terminal responds with a status word.

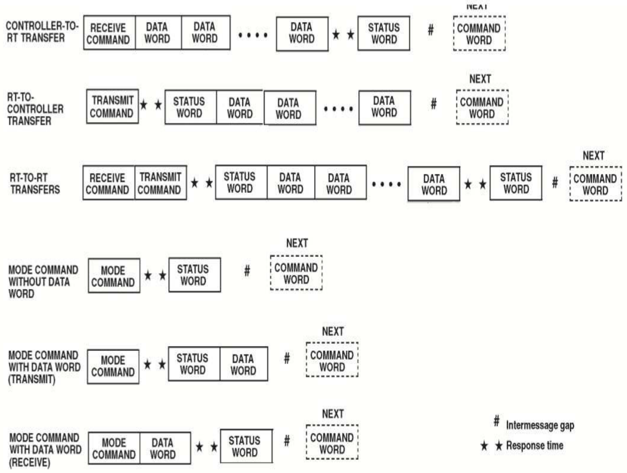


Fig. 2 MIL-STD-1553B Information Transfer Formats

1.3 Word Formats

There are three different words that form the messages that are transmitted on the bus; command word, data word and status word. Each word is formed by three-bit time sync, sixteen bits for the information field itself, and a parity bit at the end, which makes a total of twenty bits.

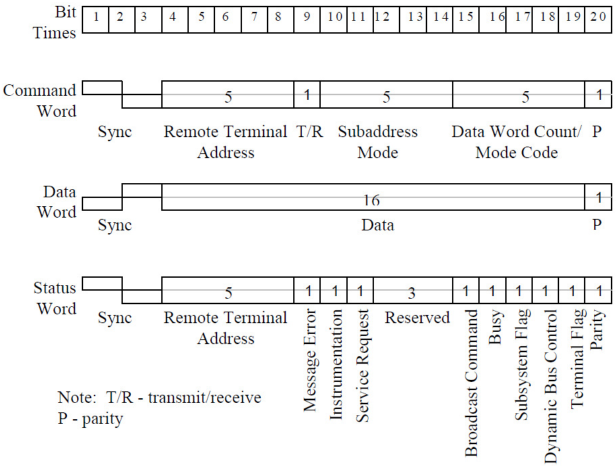


Fig. 3. Word Formats of 1553B

Each bit is timed as one microsecond, resulting in one megabit per second transmission rate for the bus. Figure 3 shows an illustration of the three word formats. The sixteen bit information fields of words are encoded using a bi-phase Manchester II format. Logic "1" is represented by a 0.5 µs high to a 0.5 µs low transition, and logic "0" is represented by the opposite low to high transition. As shown on Figure 3 each word is preceded by a three-bit sync which is encoded as 1.5 µs low and 1.5 µs high for data words and opposite 1.5 µs high and 1.5 µs low for command and status words. The command word can be transmitted only by the bus controller and as its name implies it contains a command to the remote terminals to perform. The sixteen bits of command payload contains five bits for the remote terminal address field, single bit for the command type field, five bits for the subaddress/mode field and five bits for the word count/mode code field. Each remote terminal has a unique address so the first five bits can uniquely adress them. The address of 31 (11111) is reserved as the broadcast address, so a maximum of thirty-one remote terminals are supported. The one-bit command type represents the action that the remote terminal should perform which is either a logic "1" for transmit or a logic "0" for receive. The five bit subaddress is used to direct the command to different functions within the subsystem. Binary values 00000 and 11111 are reserved and they indicate that the command is a Mode Code. The last five bits represent the number of words 10 to be transmitted or received. Binary value 00000 is interpreted as thirty-two words so a maximum of thirty-two words can be transmitted and received with a single message in a 1553 data bus.

2. Design Methodology

The Bus Controller is responsible for directing the flow of data on the data bus.The software in the PC generates packets or words of the command or data. encode the word into Manchester II Biphase, and recognize the command from BC whether it is for RT1 or RT2 and so on. The system works in the following manner. The BC PC generates a command word. With this command word is the address of the RT, which is to execute the command. The command is in asynchronous serial form. The 16 bit word is then given out through parallel ports. These sixteen parallel bits are then converted into a serial stream through parallel to serial shift register. This synchronous message is then given to the HI-15530 Manchester Encoder /Decoder. This is a high performance CMOS integrated circuit designed to meet the requirements of MIL-STD-1553B and similar Manchester II encoded, time division multiplexed serial data protocols. The 1553 data bus protocol has a synchronous command/response nature when the bus controller sends a command and waits the status messages back from the remote terminals. The device generates MIL-STD-1553B sync pulses, parity bits as well as the Manchester II encoding of the data bits. The decoder recognizes and identifies sync pulses, decodes data bits and performs parity checking. The HI-15530 supports the 1Mbit/s data rate of MIL-STD1553B over the full temperature and voltage range. The encoder requires a single clock with a frequency of twice the desired rate. The decoder requires a single clock with a frequency of 12 times the desired data rate.

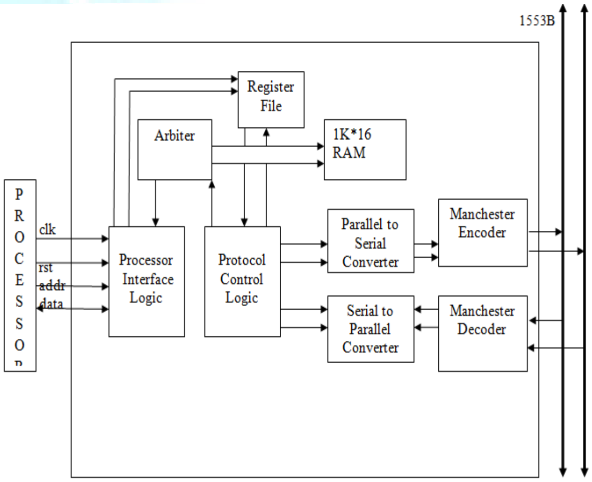


Fig. 4 Design of 1553B BC

The data from BC is received by all the Remote Terminals. However, only that RT processes the command which has the same address as contained in the command word. The data sent by the RT contains the address of BC. Therefore only BC receives this data and processes it as required. The data on the bus is in synchronous form, so for the BC or RT to read the data it has to be converted into asynchronous one. The decoder decodes the Manchester encoded data and gives it to Serial to Parallel shift register. This data is in parallel form and then converts the data into asynchronous serial format and gives it to the BC or RT.

2.1 State Machine Design

The protocol controller blocks are designed using state machine modelling in VHDL. Blocks are designed related to the requirements. State diagram of BC controller is given in figure 5. First BC sends CW, which is common for all the three types of message transfers. The next states of the diagram are different for each type of the transfer. And this state machine strictly follows the protocol rules of 1553. As we can send a maximum of 32 DWs only in a single message, there is a counter for counting data words which checks whether the count exceeds 32. Similarly for messages, there will be a message counter which checks the number of messages, which should not be more than 32 in a single transfer. Bus controller is associated with three types of transfers which as, BC-RT transfer, RT-BC transfer and RT-RT. Each of these message transfers is initiated by a CW from BC. Each transfer may contain a maximum of 32 Data Words and at least one Status Word indicating the status of that message transfer. And a maximum of 32 such message transfers are allowed in each communication.

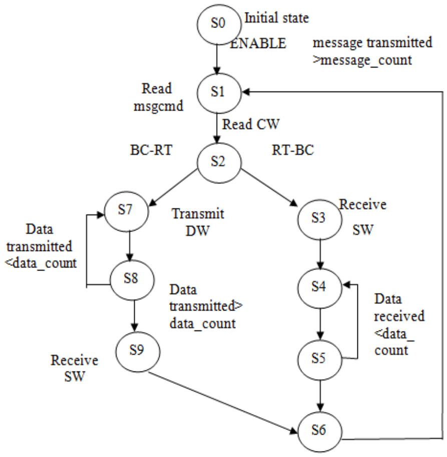


Fig. 5 State Diagram of 1553B Protocol Controller

There are pre-defined formats for each of the above said transfers. BC controller's state diagram follows the same formats. For all message transfers, first the BC sends a CW; which initiates the message transfer in 1553 data bus. It may be either transmit CW or receive CW according to the type of transfer.

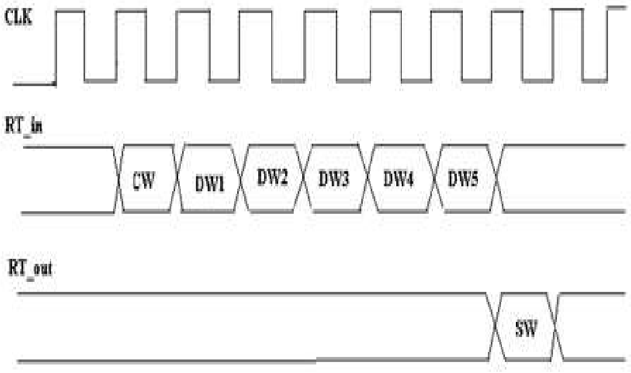


Fig. 6 Timing diagram of Bus controller (BC-RT transfer)

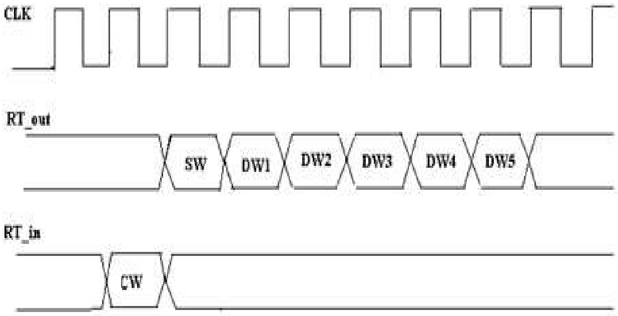


Fig. 7 Timing diagram of Bus controller (RT-BC transfer)

In BC-RT transfer, the controller sends a CW to the bus which is followed by continuous DWs (a maximum of 32 DWs are possible). The initial CW carries the address of the RT to which BC wish to communicate. All RTs decode the CW and only the particular RT whose address is present in the CW receives the following DWs. After receiving the CWs and DWs the RT sends the SW as acknowledgement. The RT-BC transfer is also initiated with a BC Command Word. The RT, whose address is specified in the CW, after receiving it, acknowledges by sending a SW. This is followed by continuous DWs to the BC, which is received serially by BC as shown in figure 7.

3. Results

The design of 1553B bus protocol controller and its information transfer is completed in VHDL language on ModelSim, including Manchester encoder and Manchester decoder. The proposed system has successfully worked as 1553 bus system, which provided information transfer between BC and RTs in the Manchester II bi-phase encoded format and the test result that is matching the expected timing diagrams is obtained is shown in figure 8.

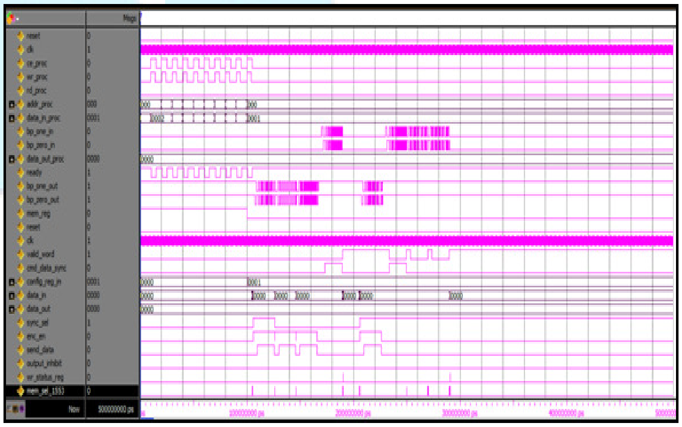


Fig. 8 Simulation Result of information Transfer of 1553B BC

3.1 FPGA Implementation of the design

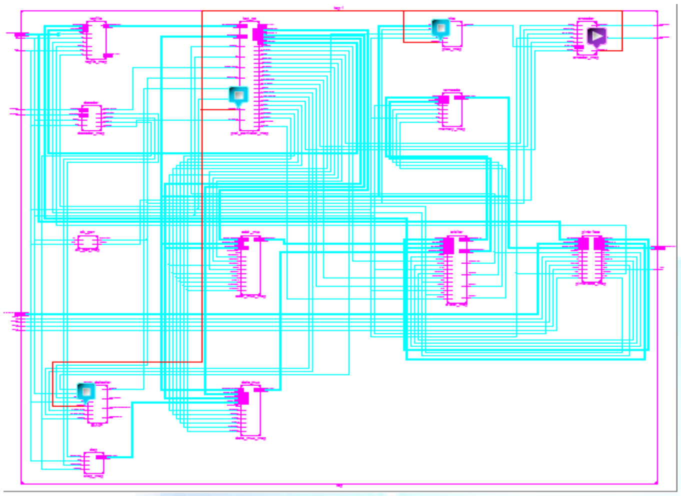
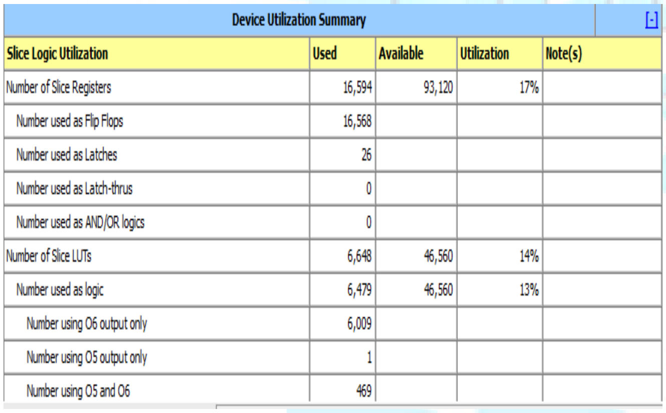


Fig. 9 RTL view of design



Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	16,594	93,120	17%	
Number used as Flip Flops	16,568			
Number used as Latches	26			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	6,648	46,560	14%	
Number used as logic	6,479	46,560	13%	
Number using O6 output only	6,009			
Number using O5 output only	1			
Number using O5 and O6	469			

Fig. 10 Target Device Utilization

Tool used for testing and simulation is ModelSim and the design has been downloaded onto a Xilinx Spartan FPGA kit using Xilinx ISE. The target system selected was a Xilinx Spartan3 FPGA (xc3s1800a). The target device utilization by the design is given in figure 10. Finally the resulting bit streams have been downloaded onto the FPGA platform in order to verify the design and the expected results were obtained.

3.2 ASIC Implementation of Design

The design is then synthesized using Cadence RTL compiler and the steps involved in the synthesis of the design are similar to the logic synthesis in Xilinx. First step in physical design is importing the design files and libraries, then floor plan, where the area allocated for the design on the chip will be decided and the power planning for the design will be done. The next step is special routing to provide power to the standard cells and next step is the placement.

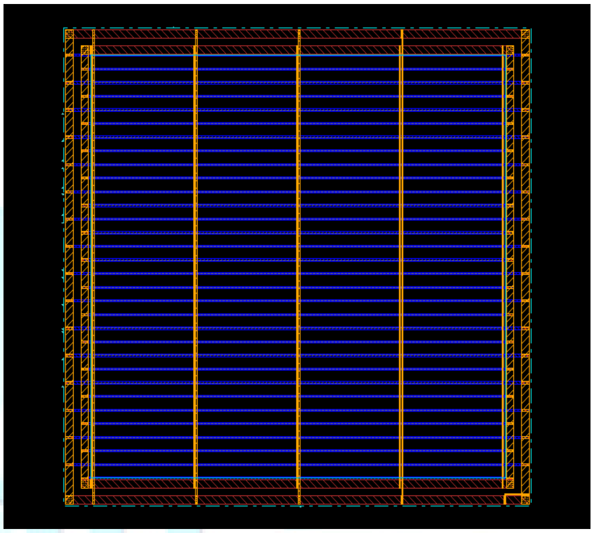


Fig. 11 Floor Plan

The design is further checked for DRC, ARC and ERC and the GDSII file has been extracted. Figure 12 shows the complete physical layout of system after Timing Analysis and Routing. It is then compared against the FPGA implementation, and it is observed that the core consumes 1.7 mW of power for the core area of 14503 μm2.It is clear that an FPGA consumes large dynamic power than an equivalent ASIC on average.

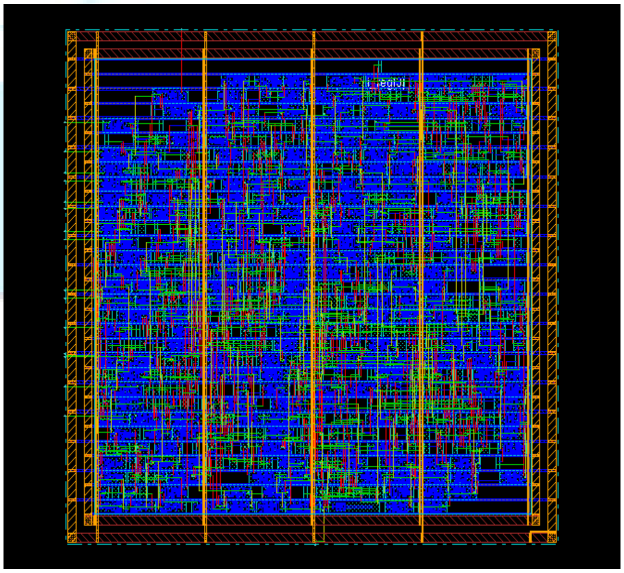


Fig. 12 Layout of the system

.

Table 1: Comparison of Synthesis Results

Design	Area	Power
FPGA	17 % utilization	115 mW
ASIC	14503 µm²	1.7 mW

4. Conclusion and Future Works

In this work data bus protocol controller for the mil-std 1553B realized by Xilinx based FPGA and ASIC design approaches. The system design has been done using state machine modelling in VHDL and the simulation is done using ModelSim. The design is then downloaded onto a Xilinx Spartan FPGA kit. The target system selected was a Xilinx Spartan3A FPGA (xc3s1800a). After implementation it's obtained that the area power utilization is somewhat high. Minimising area and power is one of the prime objectives for space applications. The size and power trade-off in implementing 1553B bus systems limit its applications in small satellites. To solve the preceding problem and realize the satellites miniaturization ASIC based design of the controller is proposed in this project. The development of physical design is done using cadence SoC RTL encounter and performance of the physical design with respect to area and power has been done. The core consumes power of 1.7 mW for the core area of 14503 µm2. It is clear from results that an FPGA consumes large dynamic power than an equivalent ASIC on average.

Since this standard has only a limited data rate of 1 Mbps, the future work should be concentrated on the speed improvements of the standard and integration of new technologies into this. Even with the recent developments of newer and higher-speed technologies, 1553 is still used for data transfer between mission critical systems of an aircraft where the reliability is of more importance than speed. Some of modern aircrafts use a mix of high-performance data buses and 1553. So it is clear that 1553 will continue its journey in the new applications and integration platforms for years to come.

References

[1] Junling Tian, Kai Hu, Huiying Zhang, Jianwei Niu, Hong Jiang, "Design of MILSTD-1553B Protocol Simulation System", Int. Conf. on Advanced Computer Theory and Engineering,2010, pp 561-566

[2] Iang Zhijian, "Research and Design of 1553B Protocol Bus Control Unit" Int. Conf. on Educational and Network Technology, 2010, pp. 16-20.

[3] Actel Corporation, Core1553BBC- MIL-STD -1553B Bus Controller, CA, USA, 2005.

[4] A. A. Name, "Conference Paper Title", in Conference Name, Year, Vol. x, pp. xxx-xxx.

[5] DDC ILC Data Device Corporation "MIL-STD-1553 Designer's Guide" ", Third Edition, 2003

[6] United States Department of Defense "Interface Standard for Digital Time Division Command/Response Multiplex Data Bus", MIL-STD-1553B, Notices 1-4, 1996.

[7] J. Furgerson, "MIL-STD-1553 Tutorial",AIM-Avionics Databus Solutions, U.K, 2010.

[8] Ian Rose, "Measuring the gap between FPGA's and ASIC's", IEEE Trans. On Compuer-Aided Design of IC, Vol.26, pp 203-215.

[9] D. Dalnodar, K Sottile, "Unique Applications of Custom MIL STD 1553 ASIC"," AIAA/IEEE Conf. on Digital Avionics Systems Conference, 2011, pp 363-367.

[10] J. Li Zohu, Linsche, "Study on Low Power Technologies on 1553 Bus", IEEE Conf. Signal Processing and Computing, 2013, pp 1-5.

[11] D. J. Jose, "Design of Manchester II bi-phase Encoder for MIL-STD-1553 Protocol", In Proc. of IEEE Int. Multi Conference on Automation, Computing, Control, Communication and Compressed Sensing (iMac4s), India, in press,2013

[12] Kai Hu, Cheng Liu, "The design and implementation of IPCore to support large fike transfer", Int. Conf. on Computing and Convergence Technology (ICCT), 2012, pp 494-498.