Vol 05, Article 11489, November 2014
http://ijves.com

International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

MIL-STD 1553 REMOTE TERMINAL PROTOCOL PROCESSOR IMPLEMENTATION USING FPGA

VISHNU PRASAD S[1], SUNIL JACOB[2], VIVEKANAND V[3]
[1]M.Tech, VLSI and Embedded Systems, [2] Associate Professor, Electronics and Communication
SCMS School of Engineering & Technology, Karukutty, Ernakulam[1,2]
Scientist se, csdd/csog/mvit, VSSC, ISRO, Trivandrum[3]
[1]vishnuprasads.ece@gmail.com, [2]suniljacob01@gmail.com

ABSTRACT

Communication between various devices or components in aircrafts is inevitable and important, delays and errors cause in such internal communication can lead to various problems such as system failure and functional errors. Wire to wire connections between such internal devices increase hardware complexity and the probability of communication errors. Use of bus wires improves the integration complexity. MIL-STD 1553 bus provides an intelligent solution for serial data communication. MIL-STD 1553 is a Manchester encoded dual redundant, bidirectional serial data bus with shielded twisted pair wire for data transfer. A maximum of 31 terminal devices can be controlled by a single Bus controller. This protocol has been used in aircrafts and military avionic systems since 1973, flexibility and reliability is the reason behind the wide acceptance of this protocol. The protocol can be configured in three modes such as remote terminal, bus controller and bus monitor. In this paper, I describe the MIL-STD 1553 Remote terminal protocol processor implementation using FPGA.

Keywords: RT, BC, MT

[1] INTRODUCTION

The MIL-STD 1553 has become the predominant internationally accepted standard for networking in military and avionics platforms. It has under gone several revisions from the time of its inception in 1973, and now expanded for international space station program and commercial avionics applications. The protocol has been used where in high precession, inherent reliability, robustness is required. So the characteristics of the MIL-STD 1553 are designed in such a way that the amount of errors should be minimum. The environments where the 1553 data bus is used are tightly integrated platforms that may cause signal interference, in order to avoid this signal interference we use twisted shielded pair wires instead of point to point wires for information transfer. The shielded twisted pair wires are characterized to cancel the internal noise and limit the external interference.
The 1553 data bus has a redundant path for information transfer. The redundant path is used when the first path fails to transfer the information or any damage occurs to it. This improves the reliability of 1553 data bus. The 1553 data bus improves the speed of data transfer in aircrafts or in any applications it is being used. The MIL-STD 1553 has three defined functions, and are Remote Terminal, Bus Controller, and Monitor Terminal. And it works in a master slave manner more precisely command/response manner. Bus controller places the commands and remote terminals act according to the commands send by the bus controller. This paper presents the design and implementation of MIL-STD 1553 Remote Terminal protocol processor using hardware description language.

2. SYSTEM FEATURES

The bus system has following features.

2.1 Bus system features

The 1553 data bus standard is based on a command /Response protocol. The components of MIL-STD 1553 bus system can be configures into three modes; Bus controller, remote terminal and monitor terminal. And the system includes shielded twisted pair wire data bus. Among the devices that are connected in the bus system one will be configured as a bus controller, which will be controlling all the other devices and the data transfer through the shielded twisted pair data bus [1]. All the other devices except bus controller are configured into remote terminal or a monitor terminal, if we want to monitor the data transfer along the data bus. A maximum of 31 remote terminals can be controlled using a single bus controller.

2.2 Remote terminal features

The remote terminal act like a slave in the bus system, it response to the command send from the bus controller. The remote terminal sends or receives a specific number of data or it does a specific mode code function respective to the command it receives. For every command from BC except a broadcast and some specific mode code command it sends a status word back to the Bus Ccontroller. The status word shows the status of current operation executed by the remote terminal [2]. Every remote terminal has a 5 bit unique address and each of them has 32 sub addresses. RT has 32 internal registers and they are used to control its Message processing.

RT has index feature with which it can buffer up to 256 messages in a single sub address. Ping Pong buffering is another feature to support transfer of periodic data with the use of two data pointers in the sub address. For every message transfer a message information word is saved in the memory location where the data pointer points[3].

2.3 Types of words controlling the operation

The 1553 standard is based on a command/response protocol, which includes different types of words for communication. These words are used to control the flow of messages through the bus and to establish the communication between the devices. These words include; Command word, Control word, Status Word and Data word.

Command word is used to establish a communication between the remote terminals and Bus controller or between two remote terminals. It includes the address of remote terminal to identify it and instruct RT to transmits or receive the data words, fig 2 shows command word structure. The data words contain the information or messages to be transmitted or received as shown in fig 3. The number data words to be transmitted are mentioned in the command word, and a maximum of 32 data words can be sent or receive in single message. Status word is the response to the bus controller from the remote terminal. It shows the status of the communication including error status, the fig 4 shows the fields of status word.

These three words are being transferred in the flow of communication but the control word in the remote terminal memory is not being transmitted but influence the data manipulation. The data in the bus are encoded in Manchester II bi phase format and are decoded at the terminals, the first three bit times shows the synchronous wave form and last bit shows the parity bit, each words are 16 bit, bit time is shown in the fig 1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Fig 1. Bit time

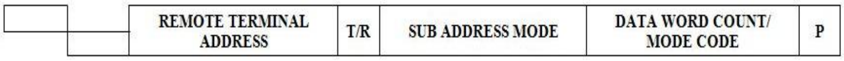


Fig 2. Command Word

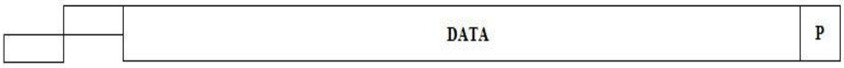


Fig 3. Data Word

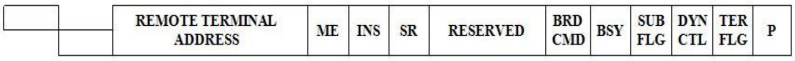


Fig 4. Status Word

Note:

ME	: Message error
INS	: Instrumentation
SR	: Service Request
BRD CMD	: Broadcast Command
BSY	: Busy
SUB FLG	: Subsystem Flag
DYN CTL	: Dynamic Bus Control
TER FLG	: Terminal Flag
P	: Parity Bit

2.4 Message format

The message transfer along the 1553 bus can be in three different ways. Data can be sent from the BC to RT, from RT to BC and it can be from RT to RT. Arrangements of command words data words and status words are different in all three kind of message transfer. The fig 6 shows bus controller to remote terminal message transfer, BC places a command word with TR bit 0 and data words are transmitted from the BC immediate after the command word.

RT responds with status word after the message processing [4]. For a RT to bus controller data transfer, BC again initiates the process by sending a command word with TR bit 1 after receiving the command word RT sends back the status word the time taken by the RT to respond with the status word is called response time, data words specified in the command word are send back to the bus controller as shown in the fig 6. RT can send data to another RT in the bus system with help of BC, for a RT-RT transfer bus controller places two command word, one with a transmit command and other with receive as shown in the fig 7. The remote terminal address in both command words must be different [5]. For every command from bus controller the Remote terminal respond with a delay called response time, and time delay between messages called inter message gap.

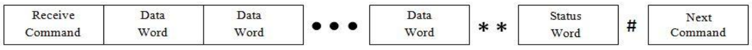


Fig 5. BC to RT transfer

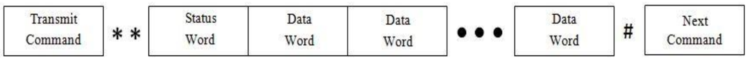


Fig 6. RT to BC transfer

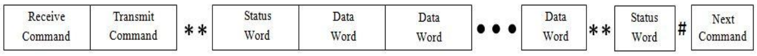


Fig 7. RT to RT transfer

Note: # : Intermessage gap
** : Response time

3. DESIGN AND IMPLEMENTATION

As discussed above the MIL-STD- 1553 has three defined terminals called BC, RT and MT. These three terminals are functionally different. Bus Controller controls the entire message transfer in the bus system but RT only send and receive data according to the commands received by it. In this paper, design of Remote terminal is being discussed. Remote terminal has certain features for its message processing and storage. Index, ping pong, and illegalization are such Remote terminal features. While designing a remote terminal these features must be given importance. RT responds to the BC command in the following way, the command word from the BC reaches the addressed RT through the BUS in a Manchester encoded form. Decoder in the RT converts it to binary bits [6].

At the beginning and the end of command processing RT access a four word descriptor block stored in the external memory. And these descriptor blocks are arranged in the memory in a sequential manner. The descriptor block address can be obtained from the sub address/ mode code and TR bit informations from the command word. The top of the descriptor table can reside at any memory location, and a descriptor block consists of a control word and three data pointers; last data pointer for broadcast commands [7]. For a transmit broadcast command RT does not send any data words. Control word allows RT to control the message processing, generating interrupts and buffering. The descriptor block arrangement is shown in the fig 8.

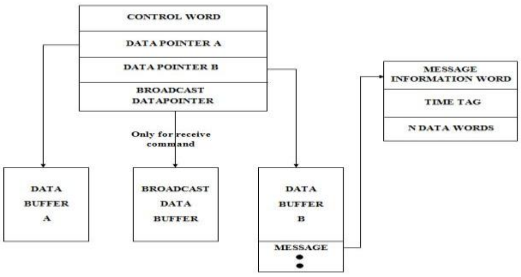


Fig 8. RT Descriptor block

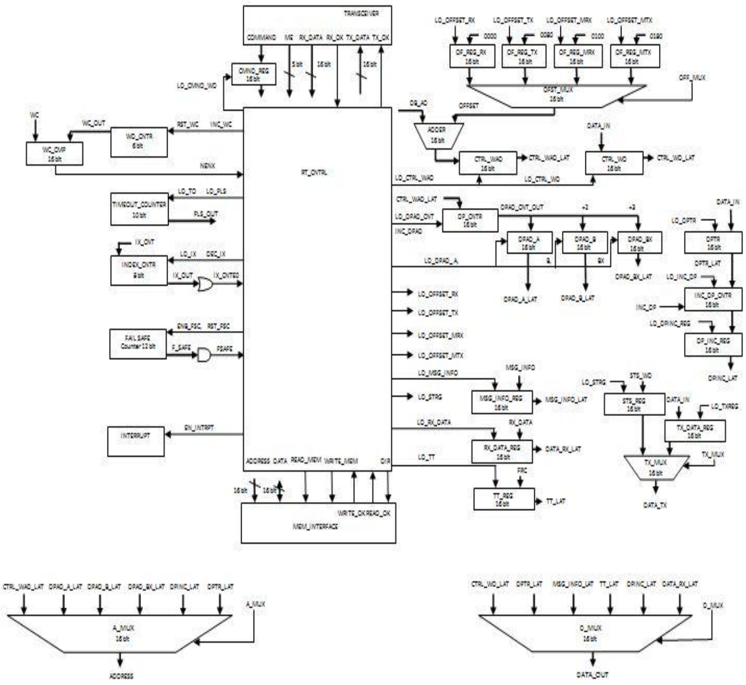


Fig 9. RT Module block diagram

For every command RT stores a message information word in the memory where the data pointer points. A free running counter value called Time Tag in the consecutive memory location.

The Fig 9 shows the block schematic of RT control. It shows components needed to calculate descriptor block and data pointer and how to store and retrieve data to and from the memory locations. While transmitting status word and data words it first given to encoder where it is converted to the Manchester encoded waves and places to the bus.

State machine for RT control can be drawn from the block schematic. It gives the flow of message processing and the entire operations done by the RT module. The state machine defines when and which components are to be enabled for the operation of the RT module. RT control state machine is shown in the fig 10. It has two branches for transmit and receive operations.

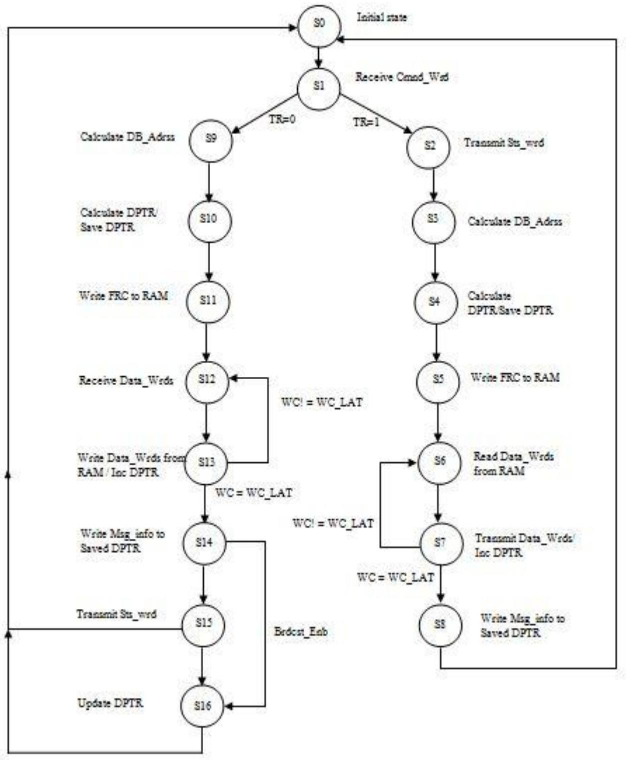
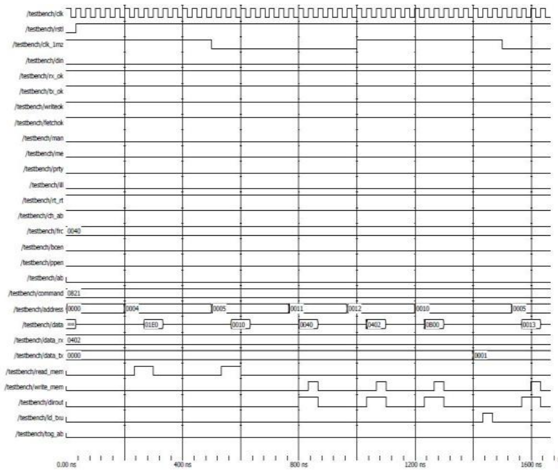


Fig10. RT control State Machine

From the state machine the HDL coding can be begun. Hdl codes are written in Actel Libero IDE. State machine modeling helps writing an effective code in Hardware Description Language.

4. SIMULATION AND RESULT

The codes written in HDL are tested using different test-bench values. Every mode of operation and message buffering were verified. Simulation wave forms obtained using Modelsim simulator. The following figures shows some of the simulation results



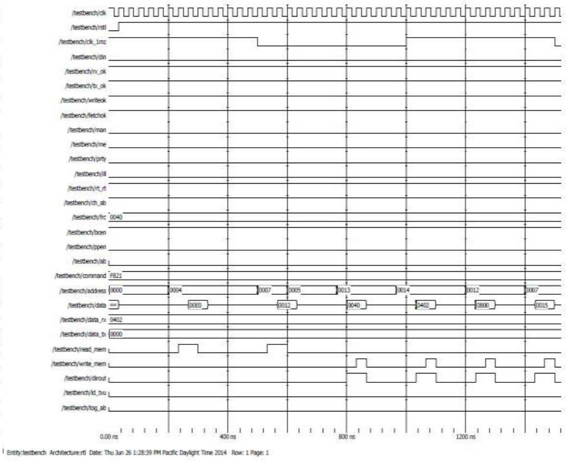
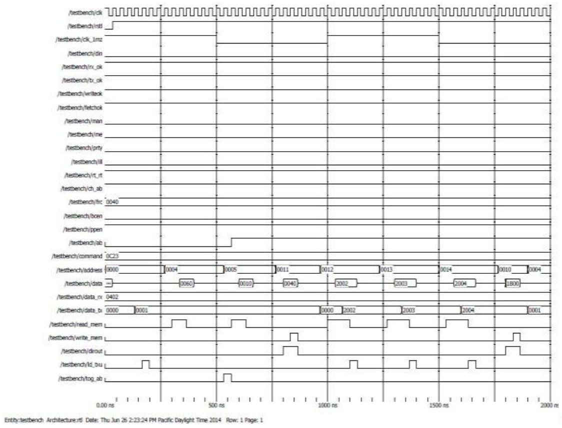


Fig 11. RT Receives Index mode

Fig 12. RT Receives Broadcast command



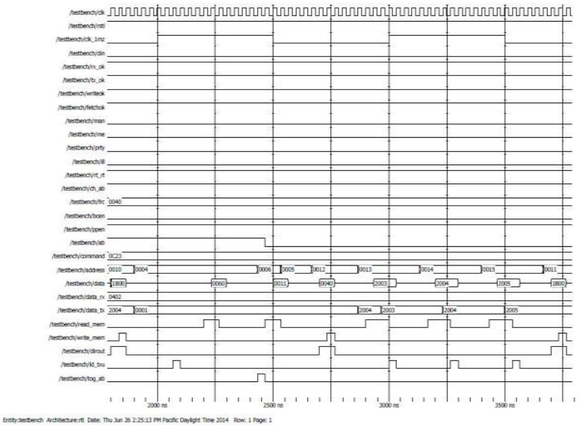


Fig 13. RT Transmits with Ping Pong buffering using data pointer A

Fig 14. RT Transmits with Ping Pong buffering using data pointer B

The design discussed above successfully simulated using Modelsim. And the information transfer tested using an existing BC module. Memory address locations and data were tabulated and verified for both receive and transmit operation. Different sub address and descriptor blocks are used in the test bench to ensure the error free operation. Timing issues were solved using simulation results. And synthesis results were checked to solve the glitches. Transfer of all possible data words were done using all possible data buffers. Message transfer observed and verified using simulation wave forms. And the proposed system worked as expected. The design is synthesized using Actel libero software.

CONCLUSIONS

The MIL-STD 1553 protocol is used in applications where high precession and reliability are required. The protocol processors are mainly used in avionics applications. The cost of this protocol processor is very high when it used in ground applications. This paper describes the implementation of MIL-STD 1553 protocol processor on to a FPGA platform, so that it can be reused for ground application in a less expensive way.

The bus system provides high speed and reliability in data transfer. To improve the speed, optical fibers can be used in the bus system. An encoder/decoder has to be designed which can convert the terminal outputs to optical signal. Optical fiber cables can be used to carry the optical signal and at the other end reverse conversion has to be done, so that maximum speed can be obtained. Storing the data of adjacent terminals in a virtual stack can also help make an error free data transfer. In the mirror stack arrangement every RT will include the informations about the adjacent RTs and are called virtual information. This virtual information can help improving the distance of data transfer using this bus system. Missed links can easily be retrieved using such virtual information. More studies has to be carried out in mirror RT virtual information system.

ACKNOWLEDGEMENTS

The authors would like to thank for the support of Department Electronics and Communication Engineering
SSET, Karukutty, Ernakulum, India.

REFERENCES

[1] Rizwan Hamid Randhawa, Muhammad Imran, "A Low Cost Design of MIL-STD-1553 devices, IEEE 2012.

[2] Aeroflex "UT69151 summit product handbook", january 2013, http://www.aeroflex.com/avionics.

[3] Carey B.R, The Avionics Magazine, December 2007, Available: http://www.avionicstoday.com.

[4] MIL-STD 1553 Application Handbook, Ballard Technology, available: http://www.ballardtech.com/T/MIL-HDBK-1553A-NII.PDF.

[5] "MIL-STD-1553 Designer's Guide", Data Device Corporation, Aug 2003, Available: http://www.ddcweb.com/Documents/dguidehg.pdf.

[6] Jemti Jose, "Design of Manchester II bi-phase Encoder for MIL-STD-1553 Protocol", IEEE 2013.

[7] Military aerospace magazine, May 2010, Available: http://www.militaryaerospace.com.