FPGA Implementation of MIL-STD-1553B Protocol: A Review

Amrutha V.
Dept. of ECE
GCE, Kannur
Kerala, India
Email: amruthadotv@gmail.com.

Anu James
Dept. of ECE
GCE, Kannur
Kerala, India
Email:anujamespaul1992@gmail.com

Sreejith S.
Professor, Dept. of ECE
GCE, Kannur
Kerala, India
Email:arr@gcek.ac.in

***Abstract*—MIL-STD-1553B is a military standard that describes the electrical and protocol characteristics for a data bus. In this paper, a review of the MIL-STD-1553B bus protocol is presented. This paper also proposes the design and implementation of MILSTD-1553B bus controller on an FPGA board.**

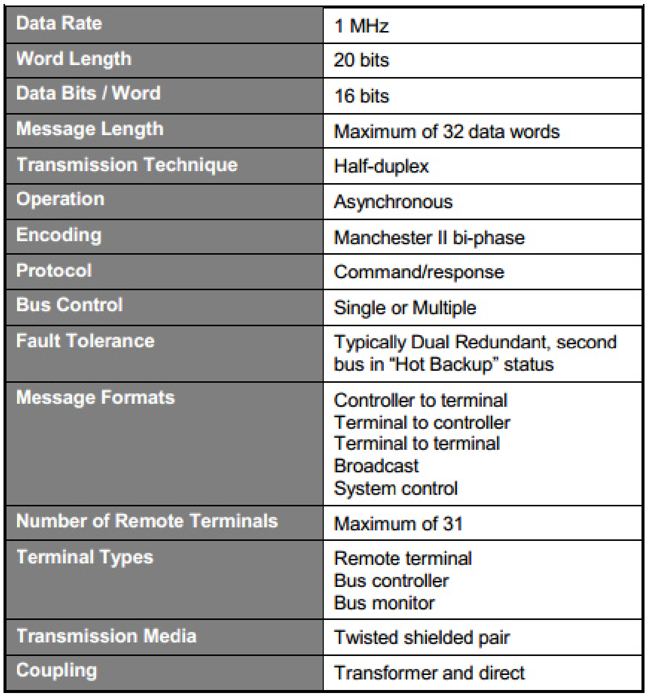
***Keywords*-MIL-STD-1553B, Bus-controller, FPGA**

I. Introduction

Modern military avionics system all devices need to communicate as efficiently as possible with a minimum amount of hardware.MIL-STD-1553 is a popular multiplex data bus standard which is originally developed by the US Air force for data integration in their military aircrafts. Since its inception in 1973 it has undergone a number of revisions, first a triservice version (means for Army, Navy and Airforce)is released and after further changes and improvements it has been opened to commercial applications as well. MIL-STD-1553B, which superseded the earlier 1975 specification MIL-STD-1553A, was published in 1978. The basic difference between the 1553A and 1553B revisions is that in the latter, the options are defined rather than being left for the user to define as required. It was found that when the standard did not define an item, there was no coordination in its use. Hardware and software had to be redesigned for each new application. The primary goal of the 1553B was to provide flexibility without creating new designs for each new user.

Now it has developed as one of the internationally accepted networking standart and an inevitable component for ship,satellites, missiles, and the international space station program, as well as advanced commercial avionicc application.

The rest of the paper is organized as follows. In Section II, we present the current state of the art related to the MIL-STD-1553B protocol . In Section III, we describe a method of implementing the protocol which would be followed for implementation of the bus controller. In Section IV we propose the work, and in Section V we discuss its applications. Finally Section V summarizes this paper with some concluding remarks.



Data Rate	1 MHz
Word Length	20 bits
Data Bits / Word	16 bits
Message Length	Maximum of 32 data words
Transmission Technique	Half-duplex
Operation	Asynchronous
Encoding	Manchester II bi-phase
Protocol	Command/response
Bus Control	Single or Multiple
Fault Tolerance	Typically Dual Redundant, second bus in “Hot Backup” status
Message Formats	Controller to terminal Terminal to controller Terminal to terminal Broadcast System control
Number of Remote Terminals	Maximum of 31
Terminal Types	Remote terminal Bus controller Bus monitor
Transmission Media	Twisted shielded pair
Coupling	Transformer and direct

Figure 1. MIL-STD-1553 Characteristics

II. Current State of the Art

A. *MIL-STD-1553B*

MIL-STD-1553B is a Digital Internal Time Division Command/Response Multiplex Data Bus. And also it is Dual redundant, bidirectional and manchested encoded, which eliminates the use of point-to-point wiring [2]. The standard describes the method of communication and the electrical interface requirements for subsystems connected to the data bus. The 1 Mbps serial communication bus is used to achieve aircraft avionic and stores management integration. The main characteristics of the standard are listed in Fig. 1.

The key elements of the standard are shown in Fig. 2.

1) Bus Controller: The bus controller is responsible for directing the flow of data on the data bus. While several terminals may be capable of performing as the bus controller, only one bus controller may be active at a time [1]. The bus

ISSN 2348 – 7968

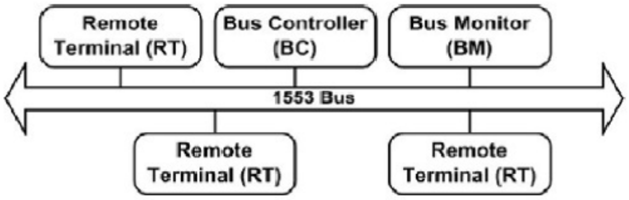


Figure 2. Key elements of 1553B Protocol

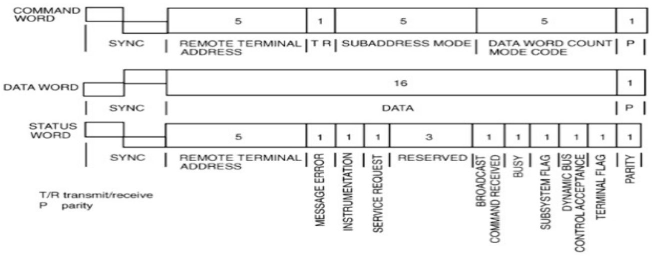


Figure 3. Word formats OF 1553B Protocol

controller is the only one allowed to issue commands onto the data bus. The commands may be for the transfer of data or the control and management of the bus (referred to as mode commands).

Typically, the bus controller is a function that is contained within some other computer, such as a mission computer, a display processor, or a fire control computer. The complexity of the electronics associated with the bus controller is a function of the subsystem interface (the interface to the computer), the amount of error management and processing to be performed, and the architecture of the bus controller.

2) Remote Terminal: Remote terminals are defined within the standard as All terminals not operating as the bus controller or as a bus monitor. The remote terminal comprises the electronics necessary to transfer data between the data bus and the subsystem. For 1553 applications, the subsystem is the sender or user of the data being transferred.

3) Bus Monitor : A bus monitor is a terminal that listens (monitors) to the exchange of information on the data bus. A monitor may collect all the data from the bus or may collect selected data [4]. The reason for restricting its use is that while a monitor may collect data, it deviates from the command-response protocol of the standard, in that a monitor is a passive device that doesn't transmit a status word and therefore cannot report on the status of the information transferred [10]. Bus monitors fall into two categories:

- A recorder for testing.
- A terminal functioning as a back-up bus controller.

B. Word formats

Communication between the subsystems in the bus is achieved by messages. A message is constituted by words. The protocol has three word formats, namely command,data and status words each 20 bits wide. These are used for the communication on the bus. Fig. 3 shows the three formats.

1) Command word: A command word has 6 fields. The first field is sync it is used to check the validity of the message it can either be 1 or zero. Sync field uses 3 bits for 1 the first one and half bits are high and the next one and half bits are low and for 0 the first section low and next section high [8]. Remote Terminal (RT) Address the next five bits following the sync shall be the RT address. Each RT is assigned a unique address. The next bit following the address shall be the transmit/receive (TIR) bit, which indicate the action required by the RT. A logic zero indicate the RT to receive, and logic one indicate the RT to transmit. The next five bits following the transmit/receive bit is used for either an RT subaddress or mode control, as is dictated by the individual terminal requirements.Following five bit is the quantity of data words to be either sent out or received by the RT. A maximum of 32 data words may be transmitted or received in any one message block. The last bit in the word is used for parity over the preceding sixteen bits. Odd parity is commonly used.

2) Data word: The Data Word (DW) contains the actual information that is being transferred within a message. The first three-bit time contains a data sync. This sync pattern is the opposite of that used for command and status words and therefore is unique to the word type. Data words can be transmitted by either a remote terminal (transit command) or a bus controller (receive command). Transmit and Receive, by convention, references the remote terminal. The next sixteen bits of information are left to the designer to define [9]. The only standard requirement is that the most significant bit (MSB) of the data be transmitted first.

The last bit (bit time 20) is the word parity bit. Only odd parity is used.

3) Status word: A remote terminal in response to a valid message transmits only the status word (SW). The status word is used to convey to the bus controller whether a message was properly received or to convey the state of the remote terminal (i.e., service request, busy, etc.).

A status word consist of a sync waveform, RT address, message error bit, instrumentation bit, service request bit, three reserved bits, broadcast command received bit, busy bit, subsystem flag bit, dynamic bus control acceptance bit, terminal flag bit, and a parity bit. For optional broadcast operation, transmission of the status word shall be suppressed.

C. Message formats

The exchange of data is based on message transmissions. The standard defines ten types of message transmission formats. All of these formats are based on the three word types just defined.

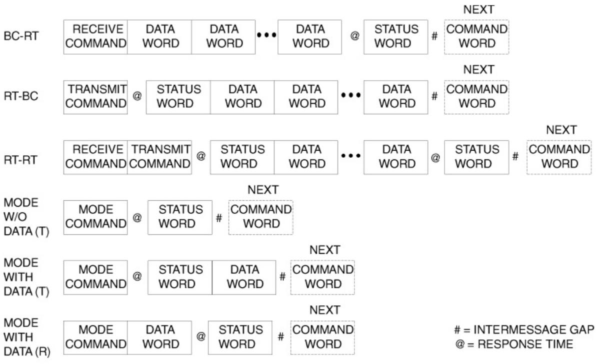


Figure 4. 1553B Information Transfer Formats

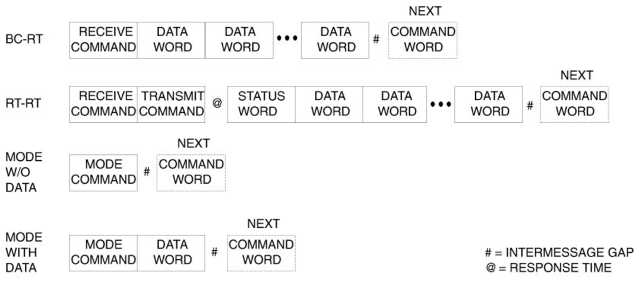


Figure 5. 1553B Information Transfer Formats (Broadcast)

The message formats have been divided into two groups. These are referred to within the standard as the information transfer formats (Fig. 4) and the broadcast information transfer formats (Fig. 5).

1) Information transfer formats: The information transfer formats are based on the command/response philosophy in that all error free transmissions received by a remote terminal are followed by the transmission of a status word from the terminal to the bus controller [8]. This handshaking principle validates the receipt of the message by the remote terminal.

1.Controller to RT Transfer. The Bus Controller sends one 16-bit receive command word, immediately followed by 1 to 32 16-bit data words. The selected Remote Terminal then sends a single 16-bit Status word.

2.RT to Controller Transfer. The Bus Controller sends one transmit command word to a Remote Terminal. The Remote Terminal then sends a single Status word, immediately followed by 1 to 32 words.

3.RT to RT Transfers. The Bus Controller sends out one receive command word immediately followed by one transmit command word. The transmitting Remote Terminal sends a Status word immediately followed by 1 to 32 data words. The receiving Terminal then sends its Status word.

4.Mode Command Without Data Word. The Bus Controller sends one command word with a Sub-address of 0 or 31 signifying a Mode Code type command. The Remote Terminal responds with a Status word.

5.Mode Command With Data Word (Transmit). The Bus Controller sends one command word with a Sub-address of 0 or 31 signifying a Mode Code type command. The Remote Terminal responds with a Status word immediately followed by a single Data word.

6.Mode Command With Data Word (Receive). The Bus Controller sends one command word with a Sub-address of 0 or 31 signifying a Mode Code type command immediately followed by a single data word. The Remote Terminal responds with a Status word [7].

2) Broadcast transfer formats: Broadcast messages are transmitted to multiple remote terminals at the same time. The terminals suppress the transmission of their status words (not doing so would have multiple boxes trying to talk at the same time and thereby jam the bus). In order for the bus controller to determine if a terminal received the message, a polling sequence to each terminal must be initiated to collect the status words.

The broadcast information transfer formats, as shown in figure 5, is identical to the non-broadcast formats described above with the following two exceptions.

- The bus controller issues commands to terminal address 31 (11111B) that is reserved for this function.
- The remote terminals receiving the messages (those that implement the broadcast option) suppress the transmission of their status word.

The broadcast option can be used with the message formats where the remote terminal receives data. Obviously, multiple terminals cannot transmit data at the same time, so the RT-BC transfer format and the transmit mode code with data format cannot be used. The broadcast RT-RT allows the bus controller to instruct all remote terminals to receive and then instructs one terminal to transmit, thereby allowing a single subsystem to transfer its data directly to multiple users.

III. Methods of Implementation

Implementation of bus protocol include the design and implementation of different modules of bus controller [3]. The bus controller (BC) is an interface device linking a MIL-STD-1553 serial data bus to a host microprocessor and/or subsystem. The architecture can be developed based on a Command Block structure and internal, programmable registers. Thus we can make the FPGA work as a bus controller.

As defined in MIL-STD-1553, the bus controller initiates all communications on the bus [5]. To meet MIL-STD-1553 bus controller requirements, a Command Block architecture that takes advantage of both internal registers and external memory can be used. Each command word transmitted over the bus must be associated with a Command Block. The Command Block requires eight contiguous memory

Control Word
Command Word 1
Command Word 2
Data Pointer
Status Word 1
Status Word 2
Branch Address
Timer Value

Figure 6. 1553B Command Block architecture

locations for each message. These eight locations include a control word, two command word locations, a data pointer, two status word locations, a branch address location, and a timer value.The host, or ROM for autonomous operation, must initialize each of the locations associated with each Command Block (the exception is for the two status locations which will be updated as command words are transmitted and corresponding status words are received). Fig. 6 shows the Command Block architecture.

To process messages, the BC uses data supplied in the internal registers along with data stored in external memory [1]. BC accesses eight words stored in external memory called a command block and executes it for each message. Command block is accessed at the beginning and end of command processing.

A. Hardware Implementation

The total design function can be described using VHDL, i.e, the bus controller may be modelled as a state machine in VHDL. The modules can then be implemented on FPGA hardware after verification of the simulation result on ModelSim.

IV. Proposed System

The proposed M.Tech project of the first author is the FPGA implementation of the bus controller for MIL-STD-1553B protocol. It contains three phases : first is the design of the architecture of the bus controller, second is to develop and simulate the VHDL code and the final phase is implementing the above code on an FPGA board and interface it with remote terminals via an MIL-STD-1553B bus [4].

For the processor to work as a bus controller it requires an external memory, in which the command words are stored and the data to be received or transmitted resides. An architecture is to be developed for the message processing of 1553B with the use of command block structure. The processor's interconnection with the memory also needs to be developed. A proposed architecture for this is shown in Fig. 7.

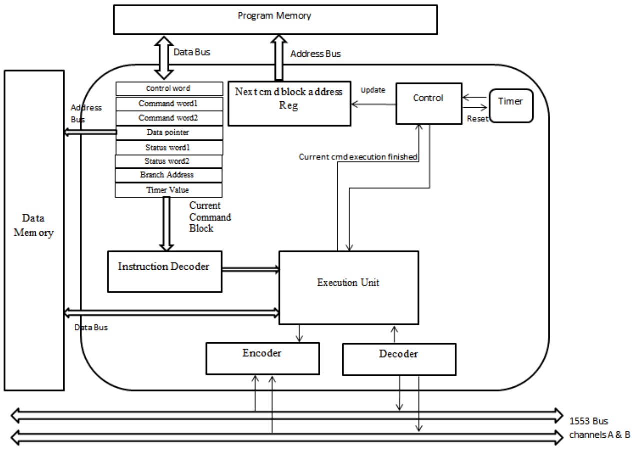


Figure 7. 1553B Bus Controller architecture

V. Applications

Since its inception, MIL-STD-1553 has found numerous applications [10]. The standard has been applied to satellites as well as payloads within the space shuttle (it is even being used on the International Space Station), its military applications are the most numerous and far ranging. It has been employed on large transports, aerial refuelers, and bombers, tactical fighters, and helicopters [11]. It is even contained within missiles and serves, in some instances, as the primary interface between the aircraft and a missile. The Navy has applied the data bus to both surface and subsurface ships. The Army, in addition to its helicopters, has put 1553 into tanks and howitzers. Commercial applications have applied the standard to systems including subways, for example the Bay Area Rapid Transit (BART), and manufacturing production lines. MIL-STD-1553B has also been accepted and implemented by NATO and many foreign governments. The UK has issued Def Stan 00-18 (Part 2)and NATO has published STANAG 3838 AVS, both of which are versions of MIL-STD-1553B. The broad acceptance and application of MIL-STD-1553B has also fostered the development of other standardization efforts. MIL-STD-1773 is a fiber optic version of 1553B. And MIL-STD-1760A, the Aircraft/Store Interconnect, has 1553B embedded within it.

VI. Conclusion

We have discussed in detail about the word and message formats of MIL-STD-1553B protocol. The significance and various applications of the protocol were also reviewed. Also the paper proposes a simplified system for the implementation of the 1553B protocol.

Acknowledgment

The authors gratefully acknowledge the following individuals for their support: Reena P., C. S. M. Kumar, Sreejesh K. V. for their support and valuable instrucions.

REFERENCES

[1] Glass and Mike, "MIL-STD-1553 Physical Layer for Time-Triggered Networks," *Digital Avionics Systems Conference (DASC)*, 2011, pp.7B1-1–7B1-8, June 2006.

[2] Yunfeng Bai and Zucheng Zhou and Junbi Chen "The implementation of MIL STD 1553B processor," *Emerging Technologies (ICET), 2012 International Conference on*, pp.1–6, Oct 2012.

[3] J. Yousaf, M. Irshad, and I, Mehmood,"Implementation of 1553B bus protocol on FPGA board using digital phase lock loop," *Proc. of fourth IEEE workshop on Multimedia signal processing*, 2001.

[4] Jifeng Li and Minggang Chai, "Design of 1553B avionics bus interface chip based on FPGA," *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pp. 3642–3645, Sept 2011.

[5] J. Jose, and S. Varghese., "Design of 1553 protocol controller for reliable data transfer in aircrafts," *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, pp.686–691, Nov 2012.

[6] D. S. D'Avino, and J. Rivera, and C. DeLong,"MIL-STD-1553 validation and test facility," *Digital Avionics Systems Conference, 1990. Proceedings., IEEE/AIAA/NASA 9th*, pp. 436–440, October 1990.

[7] Yuan Ye and Hu Xiaoguang and Zhang Guofeng, "Design of MIL-STD 1553B data bus simulation platform based on TCP/IP," *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, pp. 2119–2121, June 2014.

[8] [Online].Available: http://www.altadt.com/support/tutorials/mil-std-1553-tutorial-and-reference/

[9] [Online].Available: http://www.altadt.com/support/tutorials/mil-std-1553-tutorial-and-reference/

[10] [Online].Available: http://www.altadt.com/support/tutorials/mil-std-1553-tutorial-and-reference/

[11] [Online].Available: http://www.milstd1553.com/