MIL-STD-1553 BUS PROTOCOL ALGORITHMS IMPLEMENTATION ON FPGA TO REALISE SYSTEM-ON-CHIP(SoC)

K. Padmanabham[1], Prabhakar Kanugo[2], M. Chandrashekar[3], K. Nagabhushan Raju[4]

[1],Analogic Controls India Ltd – Hyderabad-.[1], E-mail: padmanabham@analogiccontrols.com
[2],Analogic Controls India Ltd – Hyderabad, . [2]E-mail:kpkar@analogiccontrols.com
[3]Bharat Dynamics Limited, Hyderabad, India, [3]E-mail: cmatham@gmail.cpm
[4]Dept of Electronics, SK Univ., Ananthpur, India, [4]E-mail:knrbhushan@yahoo.com,

Absrtact : This paper provides design approach to device "System-On-Chip (SoC)" using FPGA with MIL-STD-1553B protocols and controller. This is possible by using FPGA having MIL-STD-1553 protocols and microcontroller/processor IP cores within the SoC. The major advantage in having SoC is that the application firmware can be embedded within the SoC for further increase of processing power. The major advantages are cost reduction, physical size/ foot print reduction, increase in the processing power to handle complex algorithms and availability at a shorter notice.

Keywords: System-On-Chip(SoC), FPGA, MIL-STD-1553B, Protocols, Micro Controller, IP-Core and Algorithms

1 Introduction

MIL-STD-1553 network/bus is a heterogeneous architecture where the various On-Board-Computers (OBC) and terminals have a master/slave relationship. Message communication is controlled by one master terminal called the Bus Controller (BC) the other slaves called Remote Terminals (RTs). The BC initiates all communications between the slaves.

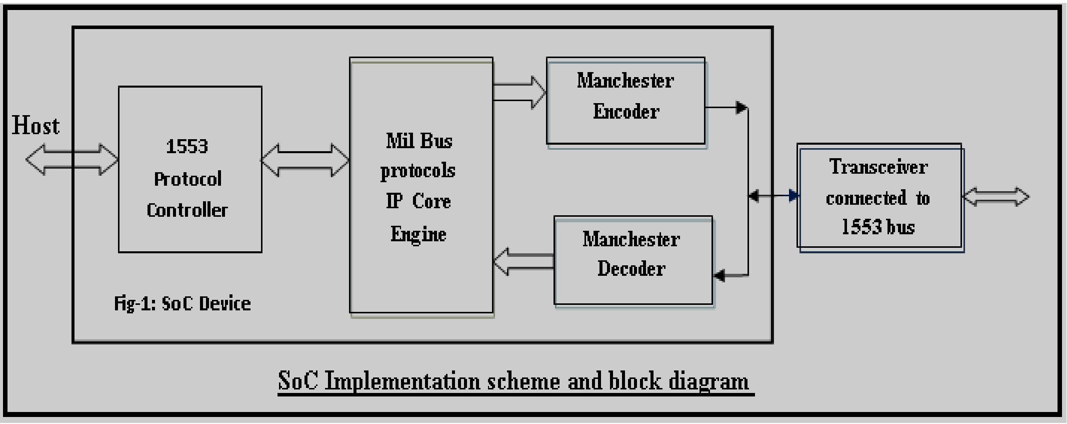
MIL-STD-1553 is a military standard that defines the electrical and functional characteristics of a serial data bus. It is designed as an avionic data bus for use with military avionics sub-systems commonly used in spacecraft on-board data handling subsystems, both military and civil. This bus can handle up to 30 Remote Terminals (devices)[8].

The MIL-1553-STD will stay for another 30 years to cater military, aircraft, other vehicles, Space etc. At this juncture the bus provides enough horsepower to handle routine aircraft communications with ease. Also the communication to happen in a deterministic, fault tolerant and keeping the redundant feature of the Mil Bus.

2 System-on-Chip Implementation scheme

2.1 Block diagram with description

The design is focused on generation of MIL 1553 Bus protocol algorithms suitable to implement in a FPGA using VHDL code to realize System-On-Chip (SoC) device. The SoC also requires controller core for interfacing with HOST and connect to MIL 1553 bus.


Host
1553 Protocol Controller
Mil Bus protocols IP Core Engine
Manchester Encoder
Manchester Decoder
Transceiver connected to 1553 bus
Fig-1: SoC Device
SoC Implementation scheme and block diagram

Fig(1) : System-on-Chip Implementation scheme Block diagram

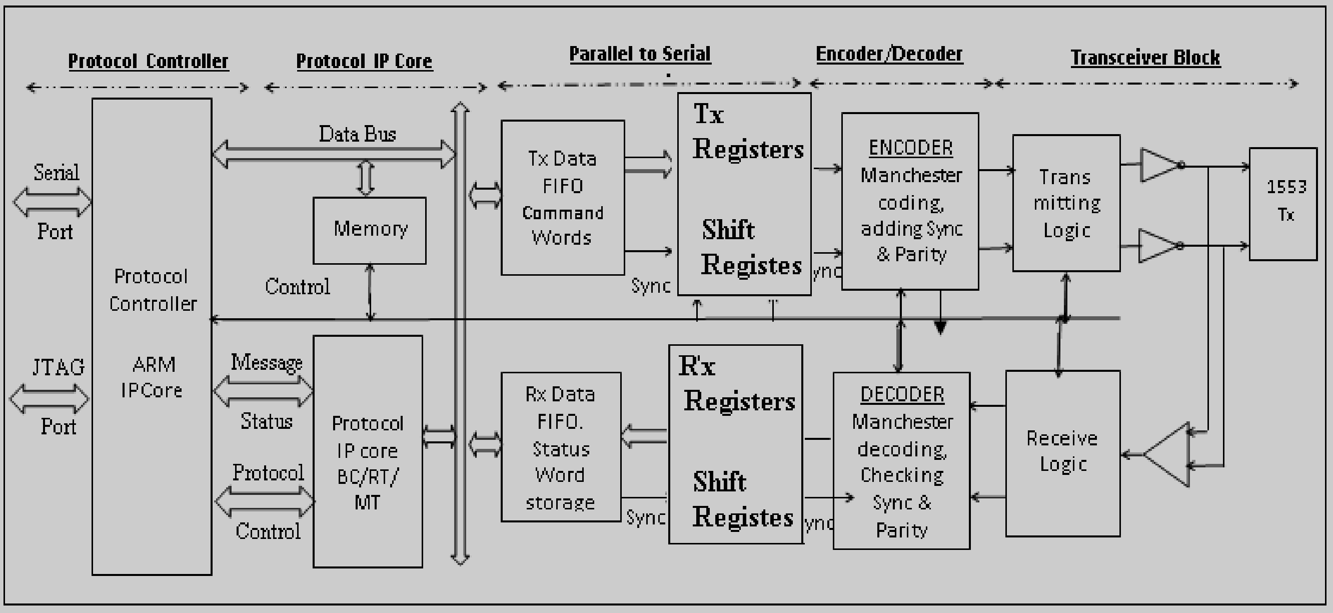
The methodology adapted to realize System On Chip (SoC) with Protocol controller using ARM IP-Core and implementing MIL-STD-1553 bus protocol algorithms, both in a single Chip FPGA as shown in Fig(1).

2.2 Major Functional Blocks in the Implementation.

The protocols are divided into the following major functional blocks for easy implementation, as shown in Fig(2).

Table (1) Major Functional Blocks

1. Protocol Controller based on ARM IP Core.
2. Memory
3. BC/RT/MT Protocol IP Corel.
4. Transmit(Tx) Data FIFO for command words.
5. Receive (Rx) Data FIFO for status words.
6. Transmit (Tx) and Shift Registers.
7. Receive (Rx) and Shift registers.
8. Manchester encoder
9. Manchester decoder.
10. Transmitting Logic.
11. Receiving Logic.
12. 1553 Isolation Transformer (Tx)



Fig(2) MIL-1553 System Architecture for System on Chip (SoC) design.

2.2.1 Protocol Controller:

The protocol controller is basically IP Core installed in the SoC to perform the supervisory role of interconnecting all the functional modules of the SoC with Host and MIL bus. Proposed to use the “ARM IP Core” for this supervisory function. The protocol controller will have Two interfaces i.e Serial Port and JTAG. The “Serial port” is meant for connecting to the Host for the Data to transmitting or to Receiving from the 1553 MIL Bus. The JTAG port is for programming the code into the SoC.

2.2.2 Memory:

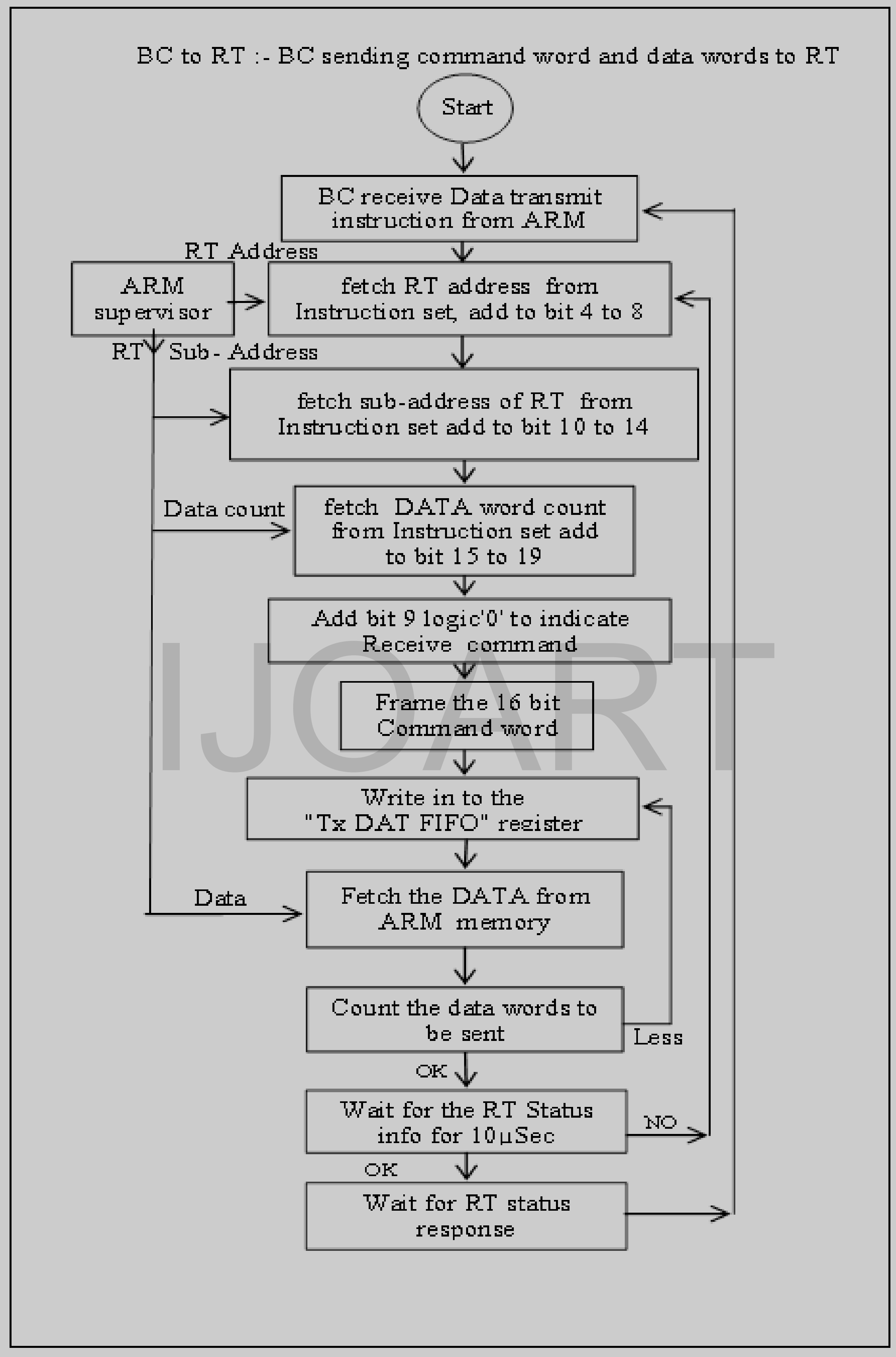
The memory part of SoC, acts like buffer and to store the incoming and outgoing data under the supervision of protocol controller.

2.2.3 BC/RT/MT Protocol IP Core :

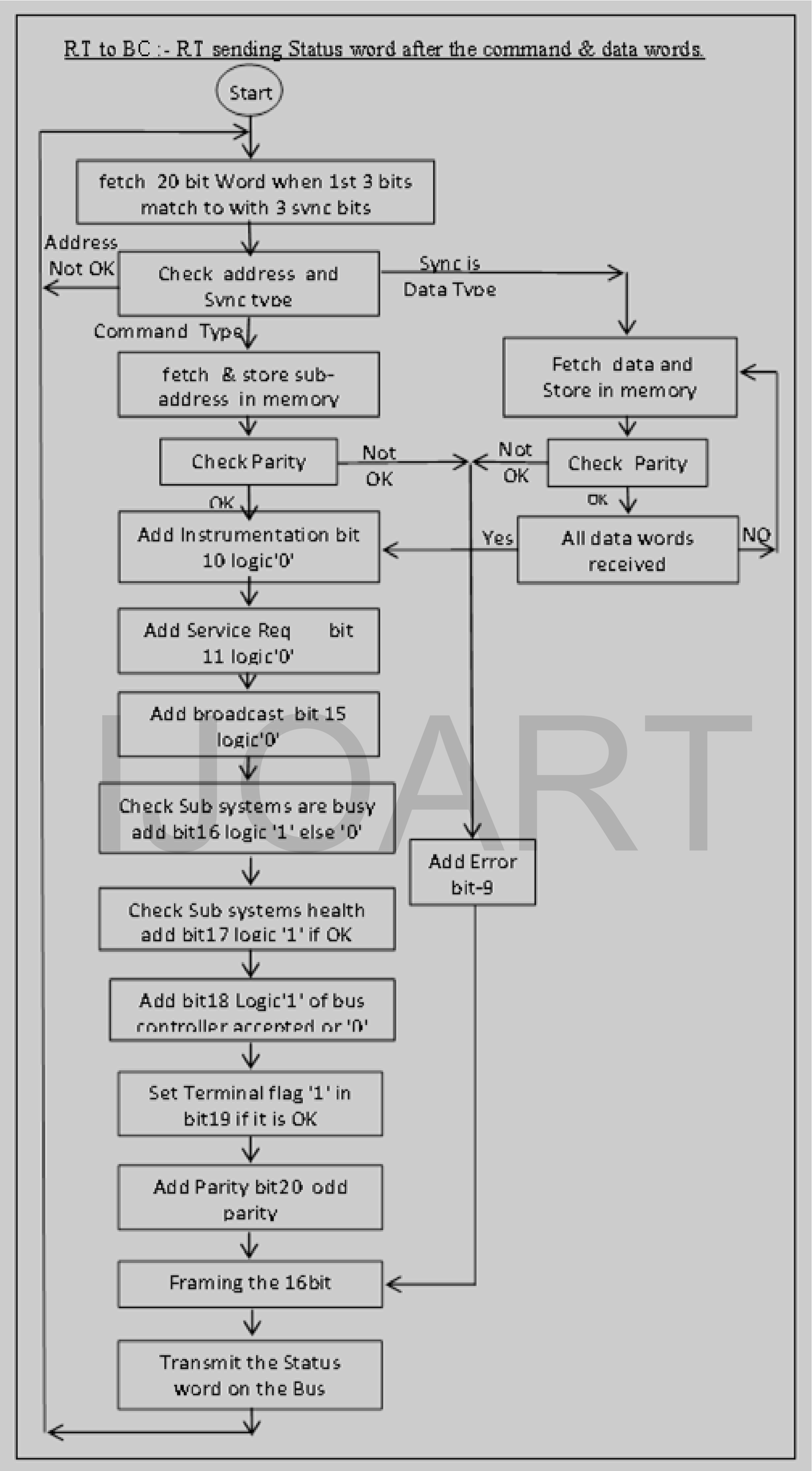
The main functions to be implemented in the IP Core, are the communication between BC,RTs. The communication between BC, RT is nothing but the 1553Bus protocols, these protocols are set of messages exchanged between the BC & RTs.

The BC operates as a bus controller. The BC initiates all information transfers on the data bus. The BC sends commands to the RTs for execution. The RT is the device that connects the data bus to the sub-system and transfers data in and out of the subsystem in response to the BC commands. The MT function is to examine the activities on the bus and record all data or selected data for off-line analysis and use for backup.

As a case study, Algorithms generated for commands from BC to RT as shown in Fig(3) and RT to BC as shown in Fig(4) i.e RT response to BC, are given below, can be implemented in FPGA.


BC to RT :- BC sending command word and data words to RT
Start
BC receive Data transmit instruction from ARM
RT Address
ARM supervisor
fetch RT address from Instruction set, add to bit 4 to 8
RT Sub- Address
fetch sub-address of RT from Instruction set add to bit 10 to 14
Data count
fetch DATA word count from Instruction set add to bit 15 to 19
Add bit 9 logic'0' to indicate Receive command
Frame the 16 bit Command word
Write in to the "Tx DAT FIFO" register
Data
Fetch the DATA from ARM memory
Count the data words to be sent
Less
OK
Wait for the RT Status info for 10µSec
NO
OK
Wait for RT status response

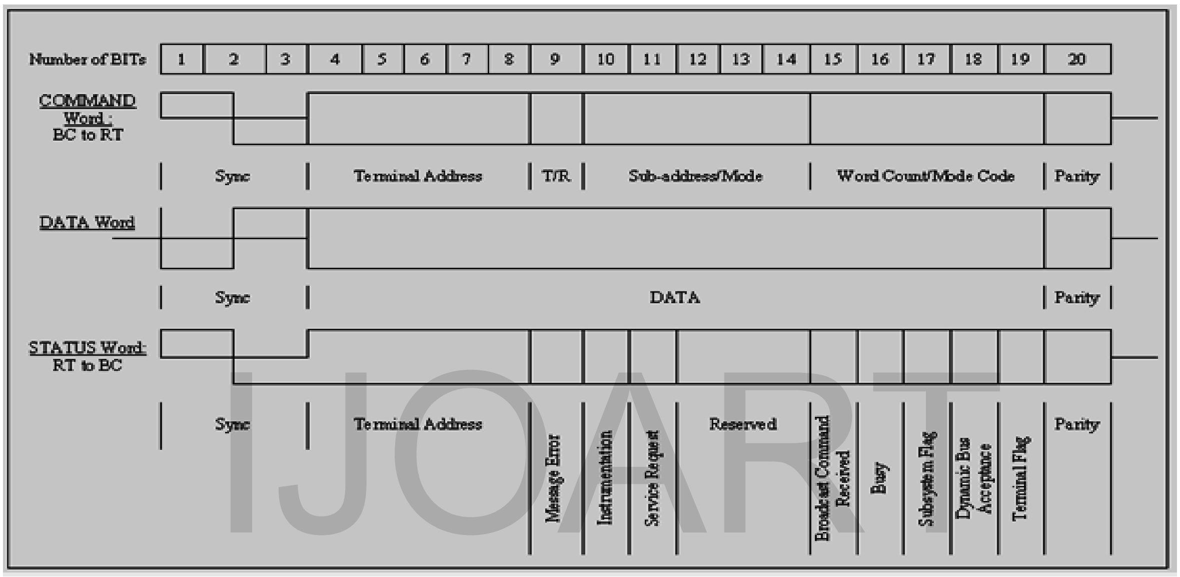
Fig(3) BC to RT command word passing Flowchart


RT to BC :- RT sending Status word after the command & data words.
Start
fetch 20 bit Word when 1st 3 bits match to with 3 sync bits
Address Not OK
Check address and Sync type
Sync is Data Type
Command Type
fetch & store sub-address in memory
Fetch data and Store in memory
Check Parity
Not OK
Not OK
Check Parity
OK
OK
Add Instrumentation bit 10 logic'0'
Yes
All data words received
NO
Add Service Req bit 11 logic'0'
Add broadcast bit 15 logic'0'
Check Sub systems are busy add bit16 logic '1' else '0'
Add Error bit-9
Check Sub systems health add bit17 logic '1' if OK
Add bit18 Logic'1' of bus controller accepted or '0'
Set Terminal flag '1' in bit19 if it is OK
Add Parity bit20 odd parity
Framing the 16bit
Transmit the Status word on the Bus

Fig(4) RT to BC Data Transfer

2.2.4 Command word in 20 Bit pattern

The above Algorithms will generate a 20 Bit word in the pattern given in the Fig(5).These 20 bit messages to be converted into serial form and encode into machester encoding to make suitable for transmitting onto 1553 bus. These word messages are stored in Transmit & Receive Data FIFO's the series of commands from the protocol controller are loaded onto 16 bit bidirectional data bus and written in "Tx Data FIFO" 16 bit registers. The commands from the FIFO register are converted from parallel to serial by Tx register and shift registers. The serial command is encoded into Manchester coding and transmitted onto the bus by transmitting logic. Similarly the Status/data words from the bus are received by the receiver logic, decoded by Manchester decoding and converted into parallel by Shift & Receiver registers then written into the "Rx Data FIFO" 16 bit register.



Fig(5) 20 Bit 1553 Bus Command, Data and Status word formats

The output messages from the registers are converted into Manchester encoding/decoding
Manchester II Encoding & Decoding : The critical functional block in MIL-STD-1553 bus is Manchester encoding & decoding . The Mil Bus is defined to be bipolar, and encoded in a Manchester format, so no DC component appears on the bus. This allows transformer coupling and excellent isolation among systems and their environment.

3 The Manchester Code

The **Manchester code** is a line code in which the encoding of each data bit has at least one transition and occupies the same time and has no DC component. The DC component of the encoded signal is not dependent on the data and therefore carries no information, allowing the signal to be conveyed conveniently by media which usually do not convey a DC component. Signal is inductively coupled to the bus and the clock signal can be recovered from the encoded data.

Manchester code always has a transition at the middle of each bit period. The direction of the mid-bit transition indicates the data. They exist only to place the signal in the correct state to allow the mid-bit transition. The existence of guaranteed transitions allows the signal to be selfclocking, and also allows the receiver to align correctly, the receiver can identify if it is misaligned by half a bit period.

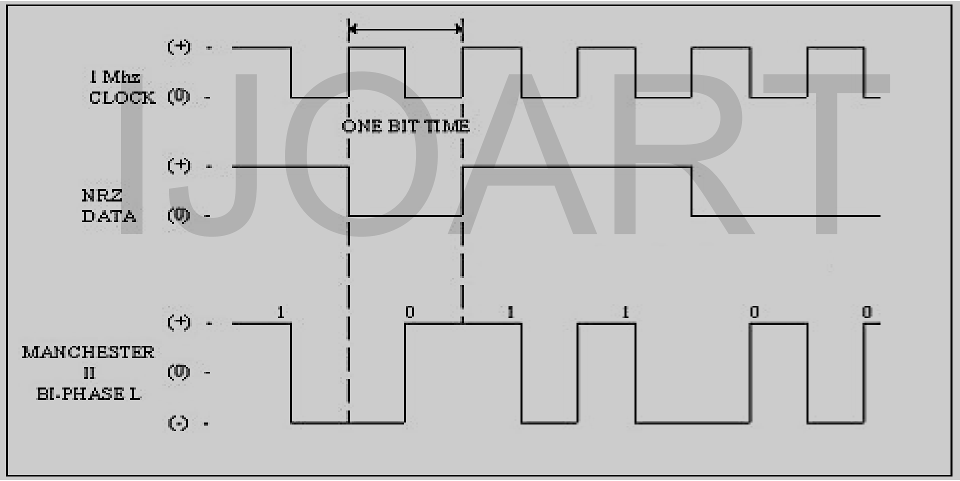
3.1 The Encoder block

The Encoder block converts the incoming 16bit serial data from the Tx Data FIFO into Serial output adding 3 bits Sync and parity bits and makes the 20 bit word. This block is divided into two sections, parallel to serial conversion and parity bit adding with Manchester coding.

3.2 The Decoder

The Decoder is continuously monitors its data input lines for a valid sync character and two valid Manchester data bits to start the cycle. When a valid sync is recognized , the type of sync is indicated on COMMAND/DATA SYNC output
If the sync character is a command sync, next sixteen clock periods data will be sent to Shift register to convert serial into parallel date. After all sixteen decoded bits have been transmitted the data is checked for odd parity and shifted into an Rx Receive register on every low-to high transition of clock.

A typical Manchester encoding scheme is depicted below:-



Fig(6) Manchester coding. (Courtesy : www.interfacebus.com)

4 Conclusions:

The Various blocks have been simulated using Xilinx Vivado development environment. Language used is VHDL which is portable across any target FPGA device. Functionality was verified through timing simulation. Physical realization of the product can be planned with any available FPGA with a gate count 700K or more.

MIL-STD-1553 is a command/response, time-multiplexed, serial data bus with a 1 Mbit/sec data rate. The bus contains a bus controller and up to 31 remote terminals. Our cores meet all requirements for dual-redundant bus operation for use with its devices in high reliability applications.

References:

[1] Theresa George, J. Mangaiyarkarasi. Electronics and Communication, Anna University, India. “Design and Implementation of Mil-Std-1553B Bus Protocol Controller with FPGA and ASIC Sharon”, IJREAT, June-July-2015, Volume 3, Issue 3, ISSN: 2320 – 8791.

[2] “Hardware-Software Co-design for SoC” Part of the SoC Design Flow and Tools Course, Department of Computer Science & Information Engineering, National Chung Cheng University Chiayi, Taiwan.

[3] Amrutha V., Anu James, Sreejesh K. V. “MIL-STD-1553B Bus Controller Implementation: A Review” 7th June 2015, Proceedings of 26th IRF International Conference, Chennai, India, ISBN: 978-93-85465-30-7.

[4] Enumala Srikrishna, Asst.professor, Thandra Paparaya Institute of Science and Technology, Bobbili, India, L.MadanMohan Sr.Engineer, Adept Chips, A.Mallikarjuna Prasad, Associate Professor, JNTUK, Kakinada “Development of MIL-STD-1553B Synthesizable IP Core for Avionic Applications”, IJCSI, September 2011, Vol. 8, Issue 5, No 3, ISSN (Online): 1694-0814.

[5] Sandi Hubric, “Using VHDL Cores in System-On-Chips Development”, European Space Agency – Netherlands.

[6] Fabrizio Ferrandi, Giovanna Ferrara, Roberto Palazzo, Vincenzo Rana, Marco D. Santambrogio, “VHDL to FPGA automatic IPCore generation: A case study on Xilinx design flow”, DEI - Politecnico di Milano.

[7] Mr. Sameer Vaghela, “Mil-Std-1553B Interface & Test-Setup Design” E.C. Department, SardarVallbhbhai Institute of Technology, Vasad GTU- Gujarat, India.Prof. Y. B. Shukla, E.C. Department, SardarVallbhbhai Institute of Technology, Vasad GTU-Gujarat, India

[8] MIL-STD-1553 Tutorial – Condor Engineering

[9] MIL-STD-1553 DESIGNER’S GUIDE – Data Devices Corporation. 105 Wilbur Place Bohemia, New York 11716-2482.

[10] Jing Reng Huang, Madhu K Iyar, Kwang Ting Cheng University of California – CA
“A self-test methodology for IP Core in Bus Based Programmable SoC”

[11] Chandrashekar, et. al. “FPGA implementation of 1553 bus Interface Unit”, Bharat Dynamics Limited and et.al., E-mail: cmatham@gmail.cpm proc. of the National conference on emerging trades in electronics & communication 2007, 23rd June 2007.
.

[12] http://www.ddc-web.com.

[13] Designing a MIL-STD-1553 System Using Core1553….http://www.microsemi.com