UVM Based Verification of MVB Encoder for Rolling Stock Application

[1]Nisha R N, [2]Ajeesh A, [3]Nevin Samuel, [4]Divya D S
[1,4]Dept. of Electronics & Communication, ER & DCI Institute of Tech., CDAC, Trivandrum, Kerala, India
[2,3]Power Electronics Group, CDAC, Trivandrum, Kerala, India

Abstract
This paper describes the implementation of Verification Intellectual Property (VIP) for MVB (Multifunction Vehicle Bus) Encoder, using UVM (Universal Verification Methodology). The SOPC (System on a Programmable Chip) based communication controller of rolling stock vehicles have an Encoder IP for MVB data, which is designed as per IEC 61375 standards. Before using the encoder as an IP, it must be verified. VIP for MVB Encoder is based on UVM. VIP is implemented with inbuilt features like random stimulus generation, error checking mechanisms, test bench generation etc. Coverage Driven Verification technology is adopted here to measure the efficiency of VIP.

Keywords
VIP, MVB, TCN, System Verilog, Manchester Coding, UVM, SOPC

I. Introduction

For rolling stock systems the digital communication between various subsystems is the central part of the design. IEC 61375, the Train Communication Network (TCN), defines communication architecture and the necessary protocols for non-vital communication for rolling stock [4]. It consists of a two-layered, hierarchical architecture to suit the needs of inter- and intra-vehicle communication. The proprietary hardware based on ASIC/microcontrollers suffer comparatively increased cost and obsolescence risk. Hence a cost effective and reusable communication controller based on SOPC methodology has been proposed [1].

According to IEC 61375 standard, a rolling stock vehicle contains two types of buses. The MVB is a highly robust real-time field bus specifically designed for control systems built into railroad vehicles. The WTB (Wire Train Bus) connects the vehicles within a train. The TCN network with WTB and MVB defines how information is exchanged [3]. Fig 1 shows the TCN architecture for the train communication.

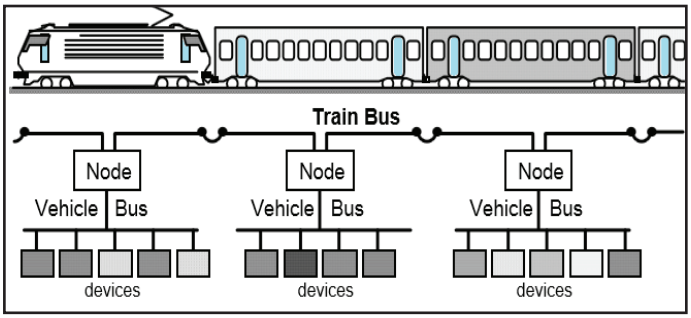

Fig. 1: TCN Architecture

The SOPC methodology explores the idea of using control blocks as hardware IPs (designed in HDL as custom IPs), which is faster and reusable than its equivalent software implementation. For the communication controller for TCN, a verified encoder IP is required for transmitting data onto the MVB as per IEC 61375. This is integrated into an FPGA along with soft CPU. VIP implementation of Encoder is based on UVM. The VIP is a verification model which helps the designers and verification engineers for the validation of their design's functionality. It can be used in most of the simulation based verifications. VIP composed of functional coverage blocks, test case generators and protocol monitors [2].

II. SOPC MVB Controller Operation

SOPC MVB controller is an interface between device applications and MVB bus. Fig. 2 shows the block diagram of SOPC MVB controller for rolling stock communication. Each device either sources or sinks data variables depending on their application. The data sourced from or sink to the device for a particular application is stored in a memory called traffic store. The Encoder IP in the controller has two main functions. It serializes data from traffic store and performs Manchester encoding on it before transferring to the MVB. At receiver, the decoder receives encoded data from MVB bus and converted it into parallel data for storage in traffic store. The Soft CPU provides necessary control signals for the encoder, decoder and traffic store.

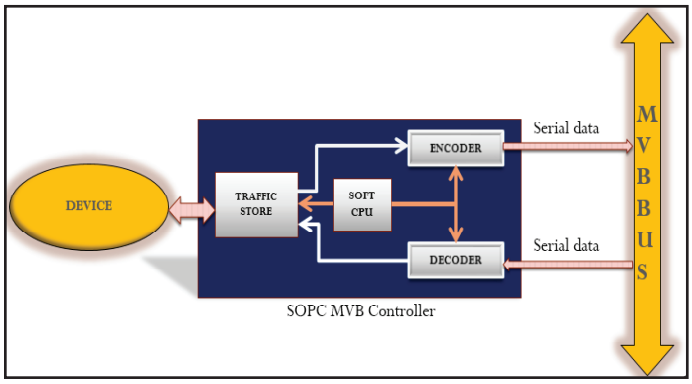

Fig. 2: SOPC MVB Controller

III. MVB Encoder

MVB Encoder is basically a Manchester Encoder. Manchester code is widely used in communication systems because of its simplicity; a single signal conveys data and clock information without the need for high-level protocol Additional benefits include self-synchronization, zero DC components, and independence from transmission media. A Manchester link consists of a transmitter (Manchester encoder) and a receiver (Manchester decoder). Manchester code embeds clock information with data in a very simple way: each bit is transmitted with a transition in the middle of the bit time. Fig 3 shows the Manchester bit encoding.
MVB Encoder transmits data in the form of frames. Different frames in MVB bus is shown in Fig 4. Depending upon the device there are two types of frames, master frame, and slave frame. The size of the slave frame can be 16bit, 32bit, 64bit, 128bit and 256bit, whereas master frame is 16bit in size. Here MVB Encoder engages in Manchester encoding of Master Frame and Slave Frame, and

generation and sending CRC check sequence. The specific tasks of the module are as follows,

- The parallel data are extracted from the buffer
- The parallel input data transform into serial data suitable for MVB,
- Add to delimiter for ready to send data
- According to the length of ready to send data, add one or more 8-bit CRC check sequence which is used for error detection purpose.
- Manchester encodes for data to send in order to improve communication quality.

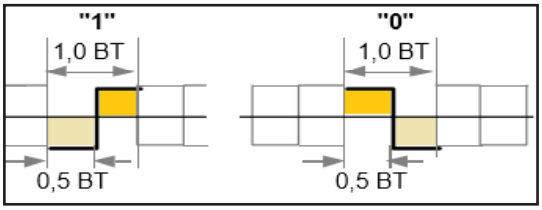

Fig. 3: Bit Encoding

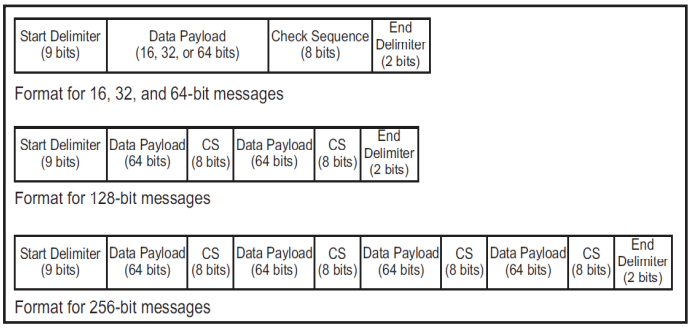

Fig. 4: Master and Slave frames

IV. MVB Encoder DUV

As shown in the Fig 5 Manchester encoder DUV (Design under verification) has 5 inputs along with a 12MHZ clock signal. START_ENC bit enables the encoder. DELIMIT_CTRL is a single bit input which decides whether the frame is master or slave. WR_IN input defines the size of the frames. FDEC is a 5-bit function code which denotes the type of expected slave frame. D_OUT is a single bit output which transmits data at a rate of 1.5Mb/s.

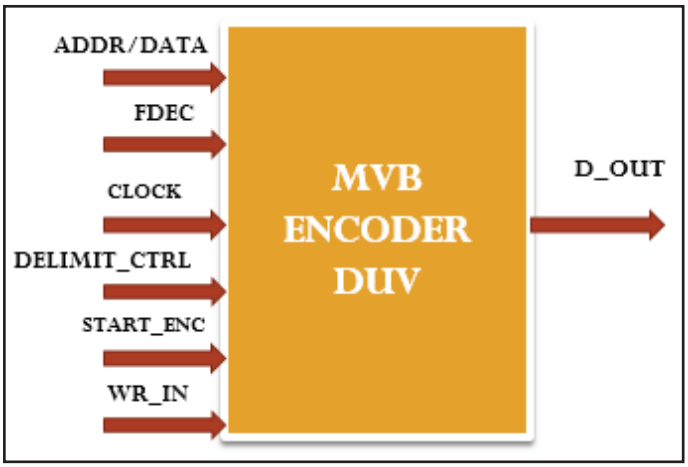

Fig. 5: MVB Encoder DUV

V. UVM Methodology

The Universal Verification Methodology is a collection of API and proven verification guidelines written for System Verilog that help an engineer to create an efficient verification environment. UVM has structured test bench architecture.

Role of each test bench element is,

- Agent-The purpose of the agent is to provide a verification component which allows users to generate and monitor pin level transactions. It includes base data objects called as transactions. The sequence is the collection of transactions for a particular test case. The driver will drive the data to the DUV. Finally, monitor observes data going to and from the DUV.
- The Environment is an entity that assembles the test bench structure. It contains scoreboard and agent. The scoreboard is used to verify the test scenario results.
- Test- It is the top level of the test bench component hierarchy. It controls the generation of the environment and initiates the test sequences
- Top- The main function of a top block includes the generation of clock signals, creating instances of the test bench and DUV. A virtual interface will connect them together.

VI. MVB Encoder Verification Environment

MVB Encoder verification environment having various modular components is shown in Fig 6. The MVB Encoder top block will connect the instance of DUV with the Test-bench using the interface. Signals from the Test-bench are sent through the interface. MVB Encoder test will configure the environment for MVB Encoder. It connects various test sequences to the MVB Encoder sequencer and applies stimulus by invoking the sequences through the environment to the DUV.

MVB Encoder Drivers translate the operations produced by the Sequencer into the actual inputs for the DUV. MVB Encoder Monitor collects the output response from DUV and converts back to transactions and sends the scoreboard.

Scoreboard designed for MVB Encoder receives output transactions from the monitor. It has a predictor (reference model) which receives input transactions which are sent to DUV to produce expected transactions. The expected output is compared with actual output.

MVB Encoder Agent connects the Encoder sequencer, driver, and monitor. MVB Encoder Transactions are the inputs which are sent to the MVB Encoder DUV. They are of high abstraction level. For example, transactions generated for the Encoder verification will include the Delimiter control bit and 16bit data. The transaction would randomize these two variables and the verification environment would make sure that the variables would assume all possible and valid values to cover all combinations.

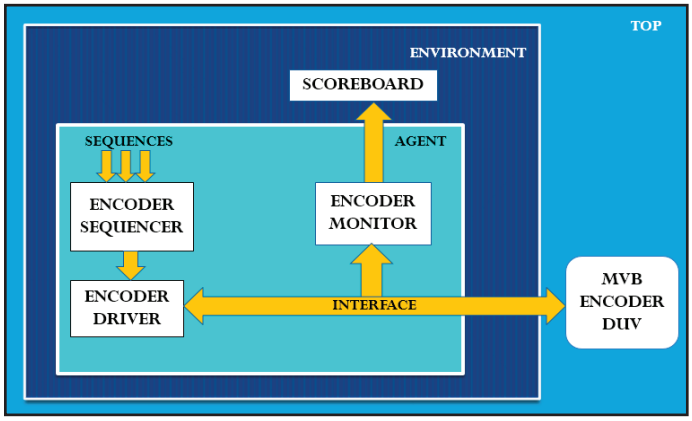

Fig 6.MVB Encoder Verification Environment

Verification environment for various frames is shown in Fig 7. Transaction level model for slave frame and master frame will contain various transactions required for slave frame and master frame data transmission. DELIMIT_CTRL input selects master or slave frame. Depending on the value of FDEC and WR_IN pin, Slave frame can be 16bit, 32bit, 64bit, 128bit or 256bit, whereas the size of the master frame should be 16 bit.

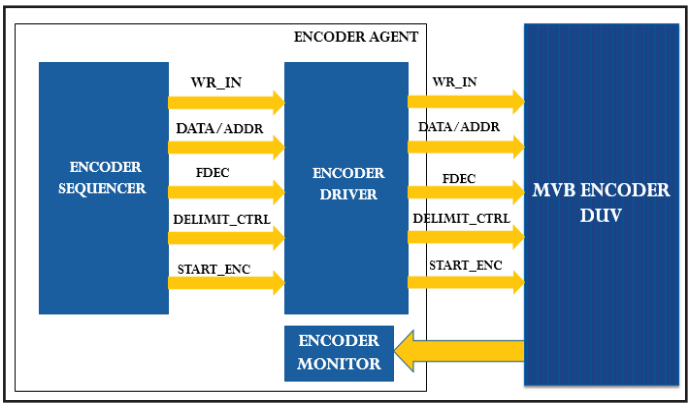

Fig. 7: Verification Environment for Different Frames

VII. Results

Different test cases like the master frame, slave 16 bit, slave 32 bit, slave 64 bit, slave 128 bit, slave 256 bit and combination different frames have been implemented. Verification engineer can select the test cases depending upon the requirement. Various master frames and slave frames of MVB Encoder is tested using test cases. Waveforms for various frames are shown in Fig 8, Fig 9, Fig 10.

UVM testbench will generate random input for the selected test case. The output from the DUV as well as the predictor is collected and it has been verified that MVB Encoder DUV and the predictor generate the same output i.e. MVB Encoder correctly transmit random encoded data with the corresponding delimiter, CRC and stop bits. Master and Slave frame data transmission starts with the delimiter sequences, which is specified in the IEC 61375 standard.

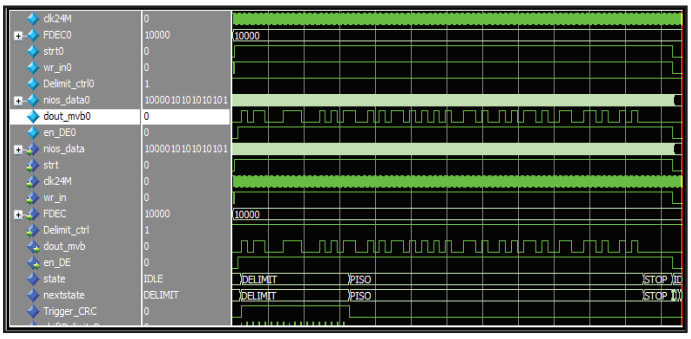

Fig. 8: Master Frame

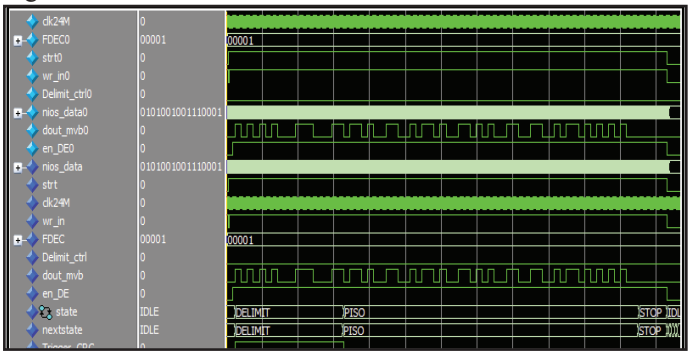

Fig. 9: 16-bit Slave Frame

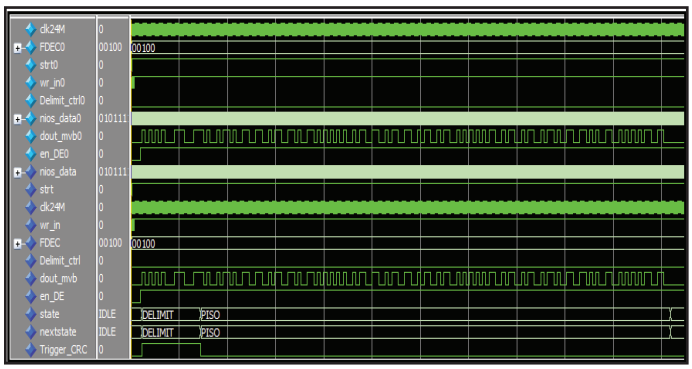

Fig. 10: 128-bit Slave Frame

VIII. Conclusion

In this paper, VIP for MVB Encoder is implemented by adopting UVM methodology. Implemented verification environment was designed to be the most flexible in terms of its utility. Testing of features of Verification Environment at transaction level runs faster and thus, it overall speeds up functional verification. VIP contains various test cases for master and slave frames. The result of each test case is also successfully compared with predictor outputs by the scoreboard. The entire logic of Manchester MVB Encoder is verified by doing coverage analyses.

References

[1] C. Schifers, G. Hans,"IEC 61375-1 and UIC 556-International Standards for Train Communication,"Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st, Tokyo, 2000, pp.1581-1585 vol.2

[2] B .Dalay, "Accelerating system performance using SOPC builder, " System-on- Chip, 2003, Proceedings, International Symposium on,2003,pp.3-5.

[3] G. A.zurBonsen, "The Multifunction Vehicle Bus(MVB), " Factory Communication Systems, 1995, WFCS '95, Proceedings., 1995 IEEE International Workshop on, Leysin, 1995, pp. 27-34.

[4] H. Kirrmann, P.A. Zuber,"The IEC/IEEE Train Communication Network", IEEE Micro, 21(2):81–9

[5] J. Bergeron, F. Delguste, S. Knoeck, S. McMaster, A. Pratt, A. Sharma,"Beyond UVM: Creating Truly Reusable Protocol Layering," Synopsys, Inc., 2013.

[6] Y Cai.,"Multifunctional compartment bus controller (MVBC) Research and Design," [D]:master's thesis.Chengdu: Southwest Jiaotong University, 2005.

[7] Yongxiang Wang, Lide Wang, Wenqing Liu, "Design and Implementation of MVB Controller using SOPC Technology", Industrial Electronics and Applications 2007. ICIEA 2007. 2nd IEEE Conference on, pp. 2666-2669, 2007.

Nisha R N received her graduation in Electronics and Communication Engineering from Cochin University of Science and Technology (CUSAT), in 2015. Currently, she is pursuing her MTech degree in VLSI & Embedded systems from ER&DCI Institute of Technology, CDAC, Trivandrum affiliated to APJ Abdul Kalam Technological University. Her areas of interest include UVM based Verification, IP Core development, Embedded system design.

Ajeesh A received B.Tech (Electrical and Electronics Engineering) Degree from Cochin University of Science and Technology (CUSAT), India in 2005 and MTech (VLSI and Embedded System) in 2008 from the same University. From 2008, he joined in CDAC as Staff Scientist. His area of interest includes FPGA- based real-time controller for Power electronics, Embedded system for Power electronics, SoC and SOPC solution for Industrial controllers, Real-time digital simulation techniques for Power electronics and Power Systems AC Drives. He is the co-author of Real-time Digital Simulation software for Full spectrum Simulator. He has published two International papers and a National paper.

Nevin Samuel received B.Tech Degree in Instrumentation from CUSAT , India in 2003. Started his career as a Project Assistant in IISc Bangalore in 2003. Joined CDAC as research fellow in 2004. Currently working as Principal Engineer in CDAC. His areas of interest include: Real-time simulation methods for Power Electronics and Power Systems, SOPC/SoC based controllers for communication and Power Electronics and Network solutions for rolling stock applications. He has published two International papers.

Divya D S received her ME Degree in VLSI Design from the Anna University of Technology Tirunelveli in 2011. She completed her graduation in Electronics and Communication Engineering from CUSAT in 2009. After that, she joined CDAC, Trivandrum as Project Engineer. Currently, she is part of Electronics and Communication Department in ER&DCI Institute of Technology which is the academic wing of CDAC, Trivandrum.