

Verification of System on Chip Integrated Communication Controllers

M.V. Petrochenkov <petroch_m@mcst.ru>

R.E. Mushtakov <mushtakov_r@mcst.ru>

D.I. Shpagilev <shpagilev_d@mcst.ru>

MCST, 1 Nagatinskaya st., Moscow, 117105, Russia

Abstract. This article presents an approach used to verify communication controllers developed for Systems on Chip (SOC) at MCST. We provide a list of communication controllers developed in MCST and present their characteristics. We describe principles of communication controller's operation on transaction, data link and physical layers and highlight their similarities. Then we describe a common method of device verification: principles of test system design, constrained random test stimuli generation and checking of device behavior. Based on common features of the controllers, we provide the general design of their test system. It includes components to work with transaction level interface (system agent of system on chip communication protocol) and physical interface (physical agent of protocol for SOC communication on a single board), configuration agent that determines device mode of operation and a scoreboard. Because controllers only execute transformation of transactions between different representation, scoreboard checks accordance of in and outgoing transactions. In addition, we describe specific features of devices that require the adjustments to the common approach. We describe how verification of those features affected the design of different test systems. We explain how a replacement of a physical agent with a second communication controller allows to speed up the development of test systems. We explain challenges of link training and status state machine (LTSSM) verification. We provide a way to work with devices with direct memory access (DMA) in a system agent. In conclusion, we present a list of found errors and directions of further research.

Keywords: Elbrus; system on chip; communication controller; Ethernet; DDR4; PCI Express; UVM; stand-alone verification

DOI: 10.15514/ISPRAS-2018-30(3)-14

For citation: Petrochenkov M.V., Mushtakov R.E., Shpagilev D.I. Verification of System on Chip Integrated Communication Controllers. *Trudy ISP RAN/Proc. ISP RAS*, vol. 30, issue 3, 2018, pp. 195-206. DOI: 10.15514/ISPRAS-2018-30(3)-14

1. Introduction

Modern systems on chip (SOC) may include multiple microprocessor cores, complex hierarchy of caches, peripheral controllers and other types of data processing modules. The task of interconnection between different systems on chip is solved by communication controller (CC) modules. Those modules solve the problem of

interprocessor communications, communication between CPU and random access memory (RAM), CPU and peripheral devices, network interfaces, etc. Performance and reliability of communication controllers is crucial for the quality of the whole system. To ensure that communication controllers satisfy all requirements, they must be thoroughly verified. Verification of complex communication controllers is a time-consuming task [1]. One of the widely used approaches to verification of SOC is system verification - execution of test programs (implemented in assembly language) on the model of microprocessor. Another approach is stand-alone verification of SOC components. In this approach, model of the device under verification (DUT) is included in a special program – a test system, which goal is to ensure that DUT satisfies all requirements. This article describes a problem of stand-alone verification of communication controllers with physical media access interfaces in the industrial setting.

The rest of the paper is organized as follows. Section 2 describes communication controllers for physical media access interfaces developed by MCST company. Section 3 presents a common approach to the design a test system and describes its components. In section 4 we provide a case study for suggested approach applied to specific devices, and adjustments to the approach that were implemented to verify specific features of those devices. In conclusion, we present of verification and provide a direction of further research.

2. Overview of communication controllers in «Elbrus-16C» microprocessor

“Elbrus-16C” System on Chip includes many communication controllers. In the following list we will describe ones that require the stand-alone verification: the most complex ones and the ones which reliability is crucial for the functionality of the system.

1. DDR4 Memory Controller is a digital circuit that manages the flow of data going to and from the computer's main memory. The controller contains the logical circuits necessary to perform read and write operations in DRAM, with all necessary delays (for example, between reading and writing). The flow of incoming requests is converted into sequences of DRAM commands, while monitoring various conflicts on banks, buses and channels. To increase the effective bandwidth of the memory channel, incoming requests can be buffered and reordered. The reordering mechanism is implemented on the basis of a sequential combination filter system.
2. PCI Express Root Complex (RC) Controller transforms packets from in-house protocol to standard PCI Express transaction level packets and implements RC configuration space for communication with peripheral devices. The controller is connected directly to on-chip network to improve throughput and reduce delays. The controller supports up to 16 lanes with speed up to 8 GT/s [2].

3. Inter-Processor Communication Controller (IPCC) is designed to solve problems of organization of multiprocessor architectures with shared memory [3]. IPCC functions are logically divided into two levels: the link layer (DLL - Data Link Layer) and the physical layer (PHL - Physical Layer). Exchange by link is carried out by transport packages (containers) of fixed size. Packages contain information about the type of the channel, data, as well as the CRC checksum. Packages are formed into containers according to special rules in order to ensure the priority and maximize the bandwidth of the link. The protocol packets are distributed among several virtual channels (VC) or streams with different priorities. To ensure the integrity of the data during the transmission over the link, the mechanism of sequential container numbering and CRC encoding are used.
4. Wide Link Communication Controller (WLCC) is used to connect south bridge controller to SOC using a protocol similar to PCI Express 2.0 but with reduced overhead. Controller supports memory and configuration space access operations. Supported link width is up to 16 lanes with speed 2.5 or 5 GT/s for each lane. To ensure channel reliability transmitted packets are protected by 16 bit CRC. After transmission, packets are stored in replay buffer waiting for receive confirmation. If negative packet acknowledge is received or time-out is reached, packets are retransmitted. Controller supports up to 8 virtual channels.
5. 10 Gigabit Ethernet Controller uses 10GBASE-KR interface [4]. It sends and receives Ethernet frames over backplane electrical interface. On a physical layer, it supports procedures of Clause 73 Auto-negotiation and Clause 72 Auto-adaptation. This device supports hardware calculation and checking of Ethernet CRC, IPv4, TCP and UDP checksums, various filtering mechanisms based on MAC addresses and VLAN tags and automatic handling of pause frames.
6. Gigabit Ethernet Controller uses 1000BASE-KX interface [4]. Ethernet frames are sent using backplane electrical interface. It supports calculation and checking of Ethernet frame CRC, calculation and checking of IPv4, TCP and UDP checksums, filtering based on mac and IP addresses and automatic handling of pause frames.

Despite the fact that those devices implement sufficiently different protocols, they nonetheless solve a lot of similar problems and implement similar features. Common features of controllers are:

- Register transfer level (RTL) models of this devices are implemented using Verilog and SystemVerilog [5] hardware description languages.
- Controllers communicate with other components on chip using the system interface that implements on-chip communication protocol, and represents transaction layer of the device.

- Controllers don't possess complex internal state and don't implement complex data processing or caching mechanisms. They transform packets between different representations: system level communication protocol packets (used for on-chip communications) and physical interface signals (used for communication on distances beyond the single chip).
- Controllers implement data link layer (DLL) that performs error detection and/or correction using such mechanisms as Cyclic Redundancy Checks (CRC) or forward error correction (FEC).
- Controllers expose the physical interface and implement logical and electrical parts of physical layer for communication with other components on a board. All aforementioned controllers communicate using low voltage differential signaling (LVDS). To ensure clock recovery and dc balancing devices use physical encoding schemes (for example 8b/10b, 64b/66b, 128b/130b) and signal scrambling.

3. Test system structure

Test systems are usually implemented using either general purpose programming languages (C++), hardware description languages (VHDL, Verilog) or dedicated verification languages (SystemVerilog, e, OpenVera). In our company we use SystemVerilog [5] with Universal Verification Methodology [6] (UVM). Use of this language allows for an easy interface with Verilog and SystemVerilog devices, and UVM describes a general test system structure and provides a library of basic verification components.

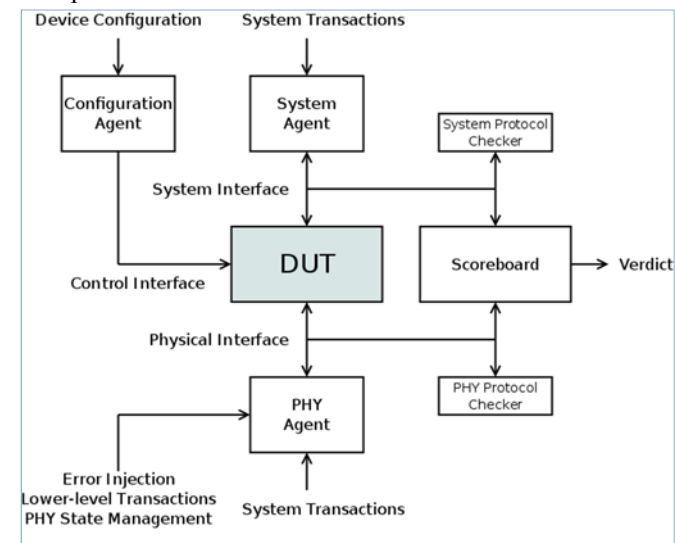


Fig 1. Structure of test system of communication controllers

Common principles of controller behaviour determine the general structure of the test system. All test systems include a set of basic components. Test system structure is presented in fig.1.

- A. Test stimuli generators are based on constrained randomization. In our case, stimuli generators communicate with system and physical interfaces of DUT. Transactions are described in terms of their attributes and constraints. To specify some test scenario, one must define specific constraints for transactions that will be issued by request generators. SystemVerilog offers a native support for constrained randomization constructs. In addition to transaction transmission and reception, physical agent is able to model some “non-standard” types of behavior: injection of corrupted or non-standard compliant transactions, or handling of received transactions in user-specified way (for example, send negative acknowledge for non-corrupted packet, drop the response to request from DUT, etc..).
- B. Test system scoreboard implements a correctness checks. Devices under verification do not possess complex data processing logic and simply perform transformation of transactions between different representations. Scoreboard receives transactions from system and physical interface monitors and performs comparison between ingress and egress transaction. If discrepancy between expected (transmitted) and received packets is detected, module reports an error in the test system.
- C. In addition to global test system scoreboard, test system contains local (system and physical) interface protocol checkers. Their goal is to check that interface rules and invariants are not violated and otherwise report an error.
- D. Configuration agent is used to access a set of memory-mapped configuration registers in the controllers. Those registers are accessed using separate configuration interface. Initial phase of a test is writing desired values to this registers.

4. Case study

This chapter describes the adjustment and highlights specific implementation details of different test systems.

4.1 Verification of Link Training and Status State Machine

One of the features of PCI Express, WLCC and IPCC links is a complex procedure of link initialization and training. During the initialization procedure device sends data patterns containing device capabilities and its current state across the link. Those data patterns are called a training sequence (TS). At the same time, using information from received training sequences, the controller detects the presence of a link partner, determines its active lanes and abilities. Based on this information, pair of devices establishes common mode of operation for transaction transfer. In addition, training

sequences are used to change the state of the link (for example, from active to low power mode or to the disabled state).

Presence of the LTSSM provides several additional challenges for the device verification.

- To send the transactions across the link, the active link must be established first. Thus, first action that the controller and its physical link agent partner performs is a link training sequence.
- One must test ability of the device to change its state and check that it reacts correctly to the state change of the link partner.
- In addition to “main” device states there are several “transient” states that the device passes when switching from one main state to another. Depending on training sequences received from link partner in transient states, link training procedure either continues successfully or terminates while reporting the error status.

It should be said that, despite the internal complexity of LTSSM protocols, they are almost invisible to the transaction layer. Only information available to transaction layer is whenever link is currently active or not.

4.2 Test systems based on a pair of controllers

To verify implementations of in-house communication protocols (IPCC and WLCC) additional type of test system was used [7]. It is based on the pair of RTL-models of communication controllers. In these test systems two controllers are connected using their corresponding physical interfaces. Errors are injected by manipulating the signals of physical interface. The structure of the test system is presented in fig.2.

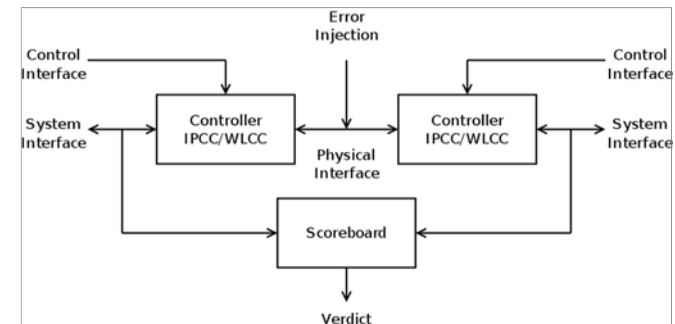


Fig 2. Structure of test system based on a pair of controllers

Advantages of the approach are as following.

- Simulation of device behaviour in realistic scenarios. Those devices (IPCC and WLCC) use our company’s proprietary protocols to connect identical devices, developed in-house. Thus, test system of this kind represents a realistic use-case of the device.

- Simplicity of implementation. The development of physical level agent is a labor-intensive and time-consuming, and its development cannot be avoided by purchasing a third party Verification IP (VIP). In this approach, the development of only a system agent is necessary, and verification can start earlier.

Disadvantages are as following.

- Lower simulation performance is caused by the need to simulate two identical controllers. This doubles the required computational resources.
- More difficult state and error injection control. To inject errors into sent and received transactions one must either directly manipulate external signals of the controller or use hierarchical access to modify the behaviour of the controllers.
- Inability to detect “self-correcting” bugs (for example, incorrect CRC polynomial). This disadvantage is mitigated by the fact those bugs will also self-correct in “real” device.
- Absence of checks on lower protocol levels. The main way to detect an error is to receive an unexpected packet on system interfaces. This may cause difficulties in bug detection and localization in many cases. For example, an error that causes an incorrect request to repeat a transaction can be detected only by performance degradation.

One can reduce the disadvantages while keeping most of some of the advantages of the approach by adding physical monitor on a link between devices.

4.3 Complex system agent in the Ethernet test systems

Distinctive feature of Ethernet test systems (both 10 Gigabit and Gigabit) is a complex system agent [8]. To reduce CPU usage and increase device efficiency controllers implement Direct Memory Access (DMA). Instead of sending Ethernet frames directly to device interfaces, frames are stored in system memory and the device reads the memory when it is ready for frame transmission. In a same way, the system must prepare a memory space for device to store received frames. The device will write the data to this location after the frame reception. Ethernet controllers are managed using a set of memory-mapped registers. The most important ones are descriptor pointer registers (head and tail). Descriptors contain an Ethernet frame metadata (size of frame, memory location address, higher-level protocol information, etc...). The head register points to the first descriptor available to the controller, and the tail points to the last processed by it. Using those registers the controller reads and writes transaction descriptors and a frame memory. The structure of Ethernet agents is presented in fig. 3.

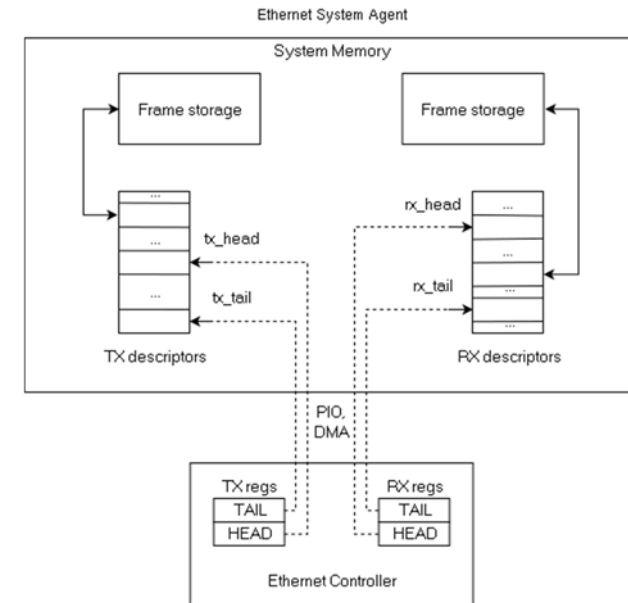


Fig 3. Structure of Ethernet controller test system

4.4 DDR4 Memory Controller protocol checks

A system agent in the memory controller test system consists of a set of two modules: the management agent of the information written into the memory and the agent for transferring requests from the system to the controller. The test system requires more sophisticated physical protocol checkers. For this purpose, two modules are used: the DFI protocol verification module and the DDR protocol verification module.

Before active work with the memory is started, the controller performs programming of the operating modes of the DRAM memory modules, conducts its initialization and training. To verify these processes, the DDR Protocol Checker is used. In addition to the fact that the module monitors the initialization and training of the memory, it also controls the execution of all the time constraints imposed to the controller when it issues commands to the memory.

Another important function of the memory controller is to periodically update the data stored in the DRAM using a refresh command. Without periodic updates, DRAM memory chips would gradually lose information, as capacitors storing bits are discharged by leakage currents. DDR protocol checker is used to analyze transactions on physical interface and to check if Refresh commands are issued within specified timing constraints. In addition, the memory state is checked before executing the Refresh command. The memory must be in the IDLE state. The controller has built-in noise immunity mechanisms that allow to check the integrity of the data, and to correct it if necessary. Such mechanisms include: rectification of parity errors of the

DDR bus, calculation of checksums, correction of CRC errors on the data bus of the DFI interface while writing, and correction of ECC errors on the DFI data bus during reading. Verification of noise immunity of transmitted data is provided by the DFI Protocol Checker. In addition, checker provides a way to verify the process of switching to and from power saving modes of memory chips by checking their timing parameters.

5. Conclusion

Methods described in the paper were used to verify components of “Elbrus-16C” microprocessor. Errors found in the controllers as a result of stand-alone verification are presented in table 1.

Table 1. Results of stand-alone verification

Device	Number of bugs
DDR4 MC	32
PCI Express RC	48
IPCC	13
WLCC	2
10 Gigabit Ethernet	51
Gigabit Ethernet	22

Verification of those devices is still ongoing. Our future work is aimed at improving those test systems, developing additional test scenarios and using the approach to verify other devices.

References

- [1] Stotland I., Shpagilev D., Starikovskaya N. UVM based approaches to functional verification of communication controllers of microprocessor systems. In Proc. of the 2016 IEEE East-West Design & Test Symposium (EWDTS).
- [2] PCI Express Base Specification Revision 3.0, <http://pcisig.com/specifications>
- [3] Belyanin I., Petrakov P., Feldman V. Functional organization and hardware means of network interconnection of modules in computer cluster on «Elbrus» microprocessors. Voprosy radioelektroniki [Issues of radio electronics], ser. EVT, no. 3, 2015, pp. 7–20 (in Russian).
- [4] IEEE Standard for Ethernet. IEEE Std 802.3-2012. 1983 p.
- [5] IEEE Standard for SystemVerilog — Unified Hardware Design, Specification, and Verification Language. IEEE Std 1800-2012
- [6] 1800.2-2017 - IEEE Standard for Universal Verification Methodology Language Reference Manual
- [7] Stotland I., Shpagilev D., Petrochenkov M. Features of High Speed Communication Controllers Standalone Verification of «Elbrus» Microprocessor Systems. Voprosy radioelektroniki [Issues of radio electronics], ser. EVT, 3, 2017, pp. 69-75 (in Russian).
- [8] S. Chitti, P. Chandrasekhar, M. Asha Rani. Gigabit Ethernet Verification using Efficient Verification Methodology. In Proc. of the International Conference on Industrial Instruments and Control (ICIC), 2015, pp.1231-1235.

Верификация контроллеров связи в системах на кристалле

М.В. Петроченков <petroch_m@mcst.ru>

Р.Е. Муштаков <mushtakov_r@mcst.ru>

Д.И. Шпагилев <shpagilev_d@mcst.ru>

АО «МЦСТ», 117105, Россия, г. Москва, ул. Нагатинская, д.1, стр.1

Аннотация. В статье описаны подходы, которые использовались для верификации контроллеров связи в системах на кристалле, разрабатываемых в МЦСТ. Представлен список контроллеров связи, а также их характеристики. Приведены принципы работы контроллеров на уровне транзакций, канальном и физическом, и отмечен их общий функционал. Затем описан общий подход к верификации устройств: принцип проектирования тестовой системы, генерации случайных тестовых воздействий и проверки поведения устройства. Представлена общая структура тестовой системы, основанная на общих свойствах устройств. Она включает компоненты для работы с интерфейсом уровня транзакций (системный агент, реализующий коммуникационный протокол системы на кристалле), интерфейсом физического уровня (физический агент, реализующий коммуникационный протокол между различными системами на кристалле на одной плате), модуль конфигурационного интерфейса, определяющего режим работы устройства, а также модуль проверки. Отмечено, что поскольку устройства исполняют только преобразования транзакций между различными представлениями, заключение о корректности поведения осуществляется на основании простой проверки совпадения входящих и исходящих транзакций. Кроме того, приведены особенности функционала устройств, которые требуют адаптации общего подхода. Объяснено, как верификация данных особенностей работы устройств определила детали структуры тестовых систем. Описано, как замена физического агента на второй контроллер связи позволяет ускорить разработку тестовой системы. Представлены методы и сложности верификации конечного автомата тренировки и состояния линка (LTSSM). Описана структура и принцип работы системных агентов, поддерживающих прямой доступ к памяти (DMA). В заключение приведен список найденных ошибок и направления дальнейшей работы.

Ключевые слова: Эльбрус; система на кристалле; контроллер связи; Ethernet; DDR4; PCI Express; UVM; автономная верификация

DOI: 10.15514/ISPRAS-2018-30(3)-14

Для цитирования: Петроченков М.В., Муштаков Р.Е., Шпагилев Д.И. Верификация контроллеров связи в системах на кристалле. Труды ИСП РАН, том 30, вып. 3, 2018 г., стр. 195-206 (на английском языке). DOI: 10.15514/ISPRAS-2018-30(3)-14

Список литературы

- [1] Stotland I., Shpagilev D., Starikovskaya N. UVM based approaches to functional verification of communication controllers of microprocessor systems. In Proc. of the 2016 IEEE East-West Design & Test Symposium (EWDTS).
- [2] PCI Express Base Specification Revision 3.0, <http://pcisig.com/specifications>

- [3] Белянин И., Петраков П., Фельдман В. Функциональная организация и аппаратура сетевого взаимодействия модулей в вычислительном кластере на базе микропроцессоров с архитектурой «Эльбрус». Вопросы радиоэлектроники, серия ЭВТ, вып. 3, 2015 г., стр. 7-20.
- [4] IEEE Standard for Ethernet. IEEE Std 802.3-2012. 1983 p.
- [5] IEEE Standard for SystemVerilog — Unified Hardware Design, Specification, and Verification Language. IEEE Std 1800-2012
- [6] 1800.2-2017 - IEEE Standard for Universal Verification Methodology Language Reference Manual
- [7] Стотланд И., Шпагилев Д., Петроченков М. Особенности функциональной верификации контроллеров высокоскоростных каналов обмена микропроцессорных систем семейства «Эльбрус». Вопросы радиоэлектроники, серия ЭВТ, вып. 3, 2017, стр. 69-75.
- [8] S. Chitti, P. Chandrasekhar, M. Asha Rani. Gigabit Ethernet Verification using Efficient Verification Methodology. In Proc. of the International Conference on Industrial Instruments and Control (ICIC), 2015, pp.1231-1235.