

System Level Modeling of a Digital Communication Channel Based on SystemC

Ibrahim Aref, *Member, IACSIT* and Khaled Elgaid

Abstract—In order to evaluate and investigate performance of a digital communication system, this paper presents a new approach that can be used to design, model and develop a Stochastic digital channel simulator to plug into any digital communication system under different modulation schemes. The data packet structure that has been used with this system is based on a High-Level Data Link Control (HDLC) standard. The simulation tool is a SystemC that is provide various features to perform system level modeling and simulation. The challenge was how to efficiently simulate this channel by using the hardware modeling features available in SystemC, and then how we have developed this channel. Also discussed how the modeling features can be used to simulate different noise sources that can be used to introduce noise events depends on specific distribution.

Index Terms—System C, modeling, simulation, channel.

I. INTRODUCTION

The design density of digital communication systems have raised in recent years, so designing and modelling a digital channel under different conditions becomes an important topic in order to recognize the overall system performance. In the last 20 years, hardware description languages (HDLs) have been widely used to model and simulate the systems in different engineering fields, but recently, the system designers are faced with three main issues :1- due to the economical reasons, systems designers need to reduce the cost and design time, 2- the shortage of a unified design environment that can be used efficiently to model and simulate modern systems, 3-growing in the complexity and size of the modern systems.

There are several approaches that have been published in [1] to design and simulate behavioral modeling of communications systems. Matlab and Simulink have been used for behavioral modelling of ADCs. These tools are flexible and easy to learn, but commercial products which are more expensive. In the other side, there are Verilog, VHDL and a lot of other HDLs to model and simulate digital and electronics systems, but all of them have can't deal very well with the increasing design complexity of those systems. To handle complexity, large designs should be broken down hierarchically. Hardware designs traditionally use blocks

interconnected with wires or signals for this purpose, but to model hardware hierarchy, SystemC uses the module entity interconnected to other modules using channels [2]. The author in [3] shows a comparison between the traditional design-flow with the SystemC design-flow which indicated that the SystemC-flow starts with an executable specification in SystemC as base for hardware and software design. This has an advantage that the hardware and software models exist in a homogeneous environment. It permits a gradual refinement of the hardware design without any language barrier. Also, the results presented in [4] emphasize the facility to describe hardware using SystemC, which provides a faster simulation time, when compared to VHDL. Moreover, SystemC is efficient to validate hardware when comparing to VHDL, and potentially enables a much wider community to design hardware.

This paper present a project based on SystemC to model, design and develop a digital channel of a wireless communication system which is represent the heart of any communication system, such the one shown in Fig. 1, which represents two communication nodes that exchange information through a noisy communication channel. . This work addresses the channel estimation for Point-to-Point (P2P) channel and Point-to-Multipoint (P2M) channel platforms with very high speed data rate. The model developed allows the investigators to vary the Bit-Error-Rate (BER) parameter of system, because as future work, we need to plug-in this channel into any digital communication system with a specific BER value that will be depended on the modulation scheme. Performance measures are estimated using Monte Carlo Simulation method (MCS) [5] as will illustrate in the next section. Although, much work found in literature on modeling in SystemC as in [6], but to our knowledge, this work is a new technique that has been done in digital channel modeling using SystemC.

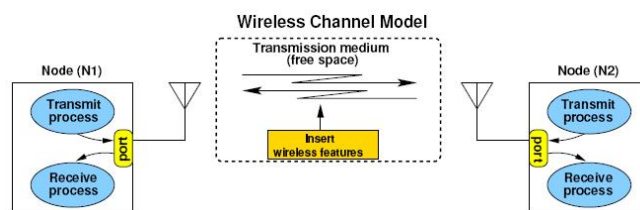


Fig. 1. Block diagram of a wireless communication system consisting of two nodes.

Manuscript received September 20, 2012; revised November 18, 2012.

I. Aref is with Computer Engineering Department., University of Tripoli, Tripoli – Libya (email: iaref@tripoliuniv.edu.ly).

K. Elgaid is with Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow- United Kingdom (email: khaled.elgaid@glasgow.ac.uk).

As mentioned above, SystemC has been used in this paper to design and model a digital channel due to its versatility, high modelling fidelity, early verification of the entire system, support homogeneous system models and ability to express system functionality at various levels of abstraction.

Managing abstraction, early verification and using the homogeneous model are the strongest weapon in combating complexity. Facing design complexity effectively with specific techniques is why SystemC is emerging as today's design language standard. Also in the industrial side, the primary motivation for using SystemC is to obtain the productivity increases needed to design modern digital systems with their ever increasing complexity, because without productivity grows, many new system concepts will be impractical.

The paper starts with briefly description of SystemC features and how we can model and simulate a digital system using SystemC, then structural description of the system as a hierarchical set of behavioral modules and finally presents simulation results with discussion and conclusion.

II. DIGITAL CHANNEL MODEL DISCRIPTION

A. SystemC Features

SystemC is a C++ class library and a methodology that can be used to model system-level designs and to effectively create a cycle-accurate model of software algorithms, hardware architecture, and interfaces of SoC (System On Chip) and system-level designs [2, 7]. In this work, structural designs for transmitting, receiving and channel models are implemented in SystemC using modules, ports, processes and signals which represent the fundamental constructs of SystemC libraries. The Modules can contain other modules, allowing the hierarchical construction of the system model. Processes communicate to each other via interfaces, channels and ports, and can synchronize with each other via events objects. Also there are a variety of data types are supported to include single bits, bit vectors and fixed-point integers [8,9]. Moreover, SystemC supports several techniques for addressing the complexity of modern designs. Today's system designer can use several approaches for attacking the complexity issues that come with complex system design, these approaches are abstraction, design reuse, team discipline, project reuse and automation [2]. In the other side, blocks communicate via ports/pins and signals or wires in traditional HDLs. In SystemC, modules are interconnected using either primitive channels or hierarchical channels. Both types of channels connect to modules via ports. The powerful ability to have interchangeable channels is implemented through a component called an interface that is implemented in this work [7].

B. Channel Implementation Description

As stated earlier, the authors have proposed the platforms P2P channel and P2M channel, the system performance is limited by noise and interference from other resources. From this point of view, our idea is introduce noise in the communication channel. Sophisticated communication systems depends on the accuracy of modeling noise as a random process, however this technique require more sophisticated calculations [5]. The main idea of introducing a noise is determining the time or position of occurrence of next error. This event is a positive random number counts the number of errors occurring in a given interval.

From this assumption, let T denote the waiting time until the first error occurs when observing this process in which the mean number of errors in the unit interval is λ . Then T is a continuous type random variable, and let proceed to find its distribution function. This has been done depending on purely stochastic process of the continues exponential distribution as shown in Fig. 2. In the first step, we have to choose S/N value that is suitable for the system that we are going to design it, then getting the corresponding BER from the Waterfall Curves Relate BER and S/N, so we can Calculate $\lambda = 1/\text{BER}$ and also calculate random value according to P.D.F. Finally, insert it in simulation program. Moreover a description of each platform will now be presented in the following sections.

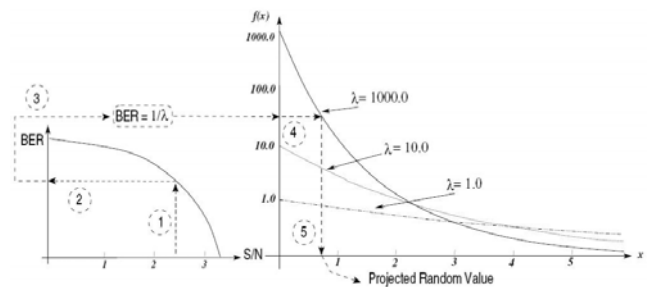


Fig. 2. Stochastic projection of Noise, Choosing S/N, Getting corresponding BER, Calculate λ , Calculate Random value, Insert it in the simulation program.

C. Point-to-Point Channel

P2P channel has been designed to implement the communication scheme just between the transmitter and receiver models. The basic design consists of a transmitter model, a receiver model, noise model, compare model and a model representing the data transfer medium which called digital channel and can be modeled wired and/or wireless. In this work, we have focused on wireless networks.

The transmitter model sends data packets that are designed and created based on HDLC format to the channel which is received those packets, and sends them on to the receiver. The digital channel can be introduced different types of errors from noise model to represent the actual error rate of the system. The noise model has introduced errors randomly by applying stochastic technique which is used to simulate the error event [5]. The occurrence of the error event is dependent on the selected distribution and then the error rate will be related to it. In the other side, the receiver model receives data packets with some error from the channel model and then sends them to the compare model.

Finally, the compare model is used to analysis the packets in order to detect the errors and then evaluate error rate by comparing the packet that is sent from the transmitting model with the packet received by receiving model. The process continues until all data packets are sent. As mentioned above, the module is the basic building block in SystemC. Within module processes and other modules are instantiated. Processes within a module are running concurrent. Processes within a module execute whenever one of their sensitive inputs change. In this work, the transmitting, receiving and channel models are implemented by using module and connected together through the ports which are created from

the base class `sc_port` and are bound to an interface type. The digital channel being designed here employs FIFOs as interface type and elasticity buffer as shown in Fig. 3. FIFO by simplicity, it is a first-in first-out buffer. Each FIFO has a number of slots for storing values. The number of slots is fixed when the object is constructed. In this system, FIFOs have been used to buffer information packets as they traverse from point to point in a network, because we have to guarantee that all the packets will remain in the right order during transmitting process. For the simulation purpose, data packets are represented by binary vector data type and are designed based on HDLC.

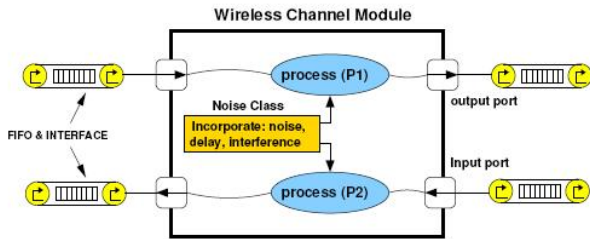


Fig. 3. Digital channel module and interfaces.

In the other side, the two functional blocks appear in the simulation model which is not part of the physical system being analyzed. These are the blocks labeled Compare and Delay. The Compare block has a clear function. The received packets are compared to the original data packets so that the error count, N_e , can be determined. A moment's thought shows the need for the block labeled Delay. A number of the functional blocks in the communication system have a nonzero phase response, and therefore a signal passing through these functional blocks incurs a time delay. As a result, the signal at the output of the data source must be delayed so that a given packet at the output of the receiver is compared to the corresponding packet at the output of the data source. The delay is introduced by caching the transmitted data packets in FIFO's and force it to wait till the packets from receive have arrived. The determination of this delay must be done with care, because we have to guarantee that both packets present in the module at the comparing moment.

D. Point-to-Multipoint Channel

By using the multi-port and/or port_array feature of SystemC and provide the second parameter of `sc_port<>`, the array size parameter can be set. At this point the idea is to provide of a number of like-defined ports. Here we assumed that our communications system has a number of interfaces all with the same connectivity and a number of N devices connected to it. Moreover by using dynamic instantiation feature of the modules and dynamic channel binding to the multipoint interface of the transmitter. We can generate as much as we wanted from these devices. Dynamic instantiation and binding must be done before elaboration of the simulation. The multipoint structure of the system is constructed as shown in Fig. 4.

E. Multipoint-to-Multipoint Construction

At this first stage of the implementation phase, our goal is to construct a channel model that can be used to support contention and noncontention based shared wireless channel

access, and includes P2M and Multipoint-to-Multipoint (M2M) communication scenarios. The wireless channel module is responsible for carrying the data packets to all stations, and represents our communication medium in this work. The wireless channel module, as shown in Fig. 5, models the communication medium. It behaves like a multi-tap bus, where the multiple nodes are connected through it. The behaviour of the link has been modelled as a wireless communication medium.

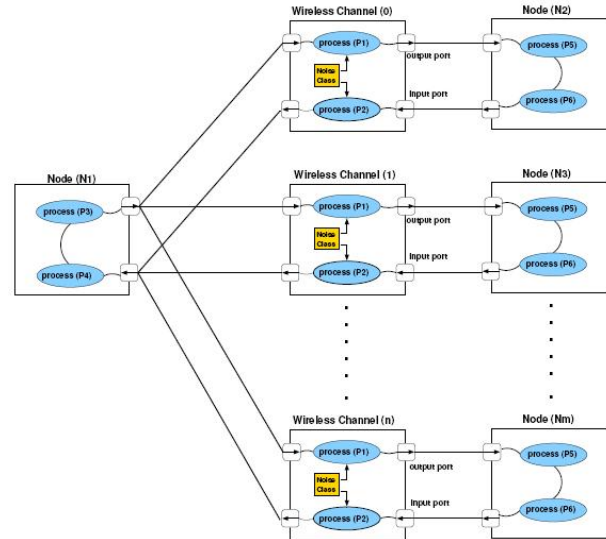


Fig. 4. Point to multipoint platform.

In this scenario an external object called a 'mutual exclusion lock', or 'mutex', has been used to control access to the shared medium. The behaviour of a mutual exclusion lock is used to control access to a resource shared by concurrent processes. A mutex will be in one of two exclusive states: unlocked or locked. Only one process can lock a given mutex at one time. Whenever a node has a packet to send, it tries to lock the mutex. When it locks the mutex it has access to the channel; the rest of the nodes have to wait until the node has released the channel. The other nodes may be subsequently locked by the mutex [10, 2].

F. Data Packetization

In data transmission process that achieved in this system, there is no mechanism to mark the beginning or end of a packet, so by using HDLC technique, the beginning and end of each packet has to be identified. This is done by using a packet delimiter, or flag, which is a unique sequence of bits that is guaranteed not to be seen inside a packet [11]. In this system, we have used Asynchronous Balanced Mode (ABM) which either combined stations may initiate a transfer.

III. RESULTS AND ANALYSIS

Since a simulation can process only a finite number of bits or packets, the BER needed for evaluation can only be determined depending on the number of packets [12]. In this work, the BER is determined by passing a large number of packets through the system and counting the errors at the destination, i.e. packets with errors observed at the monitor module divided by the total number of packets sent from the source. We have set our simulation to run until the BER converges. In the simulation of P2P, we set the BER to 0.02;

the result of replicating the random experiment of passing a large number of packets or bits through the random channel is shown in Fig. 6. A BER based on any number of transmissions gives a spread of results, which are evaluated when an error occurs. This spread is related to the variance of the estimated value in general. In order for simulation results to be useful, the spread should be small and converge to the given value of BER. Note that, for the results shown, the variance grows smaller as the number of packets injected grows larger. This is typical behavior for a correctly developed estimator. Fig. 7 compares the total number of packets received over a noisy wireless channel against packets successfully received and packets with an error.

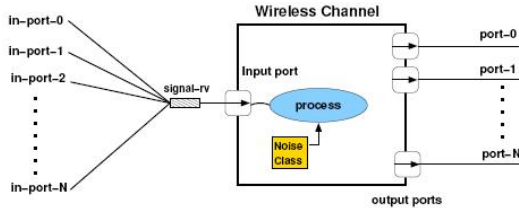


Fig. 5. Multipoint-to-Multipoint communication channel.

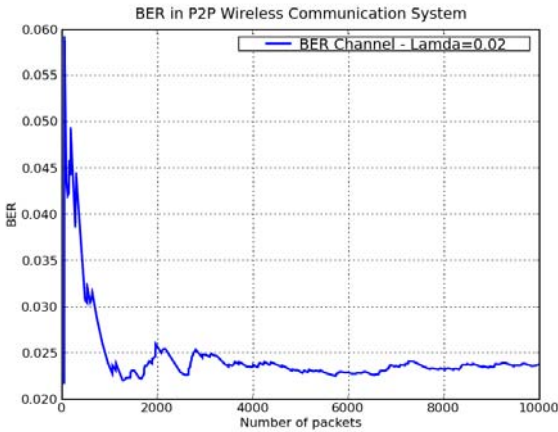


Fig. 6. Changing in BER versus number of packets in P2P communication Channel.

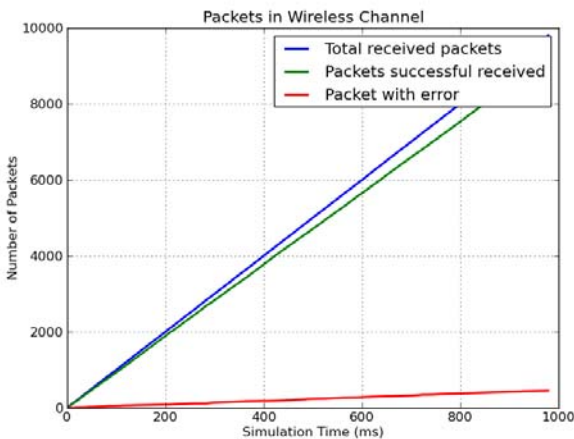


Fig. 7. Packets in the noisy communication wireless channel.

For the P2M platform we have set $10 \leq \lambda \leq 10000$ randomly to ensure different values of λ ; also, each channel is set to process each data packet with different noise. In this simulation, we obtained $\lambda_1 = 8400$, $\lambda_2 = 7800$ and $\lambda_3 = 9100$ with number of packets $N = 10,000$. Fig. 8 shows the P2M

BER results with three channels and three receivers.

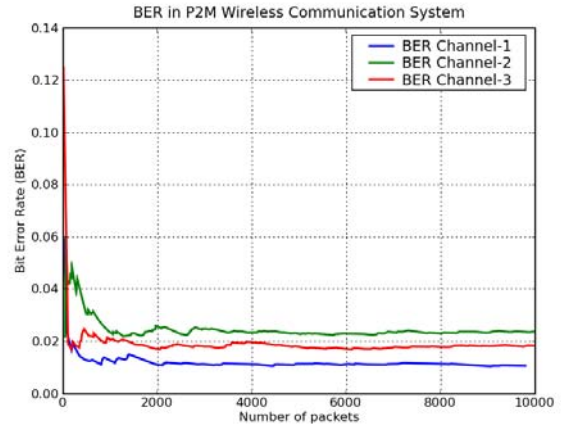


Fig. 8. Changing in BER versus number of packets in P2M communication channel.

We have seen that the noisy channel has been simulated correctly and our platform has verified the correct simulation of the system and the correct injection of errors in the bit stream. A sample of the expected throughput for P2P (shown in Fig. 9) and for P2M (shown in Fig. 10) has been determined by the monitor with an ARQ window size set to 7.

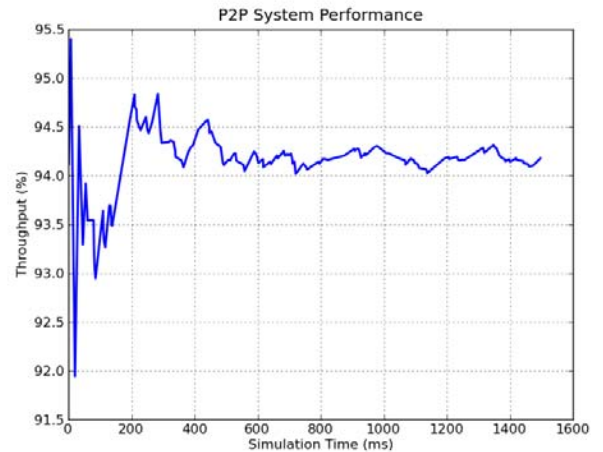


Fig. 9. System throughput for P2P with ARQ window size 7.

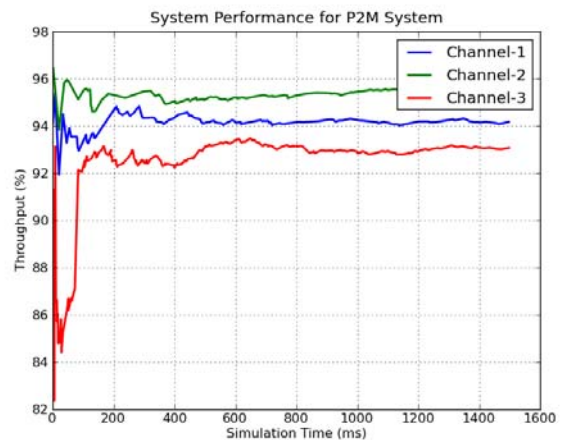


Fig. 10. System throughput for P2M with ARQ window size 7.

IV. CONCLUSIONS

This paper has explored a new approach to model and

develop a digital channel using SystemC at a high level of abstraction and then as future work we can refine the design down to a level that allows hardware implementation (RTL level). Although, much works have found in literature on modeling with SystemC, but to our knowledge, this is a new technique that has been done in digital channel modeling using SystemC, and much work with it, is still as future potential. Also, we can say that Systemic is provided various features to perform system level modeling and simulation, which are missing in the generic HDL's such as VHDL and Verilog and it is intended to enable system level design and IP exchange. Moreover, we need to say that SystemC is not the perfect solution for all tasks, but it combines many design characteristics that are missing in other language.

REFERENCES

[1] J. Bjornsen and T. Ytterdal, "Behavioral modeling and simulation of high-speed analog-to-digital converters using systemc," *IEEE Circuits and Systems, ISCAS '03*, vol. 3, pp. III-906-III-909, May 2003.

[2] D. Black and J. Donovan, "SystemC: From the Ground-up," Kluwer Academic Publishers, first ed., 2004.

[3] L. Franzens, "Systemc for embedded system design," University of Innsbruck, Institute for Informatics, 2006.

[4] N. Calazans, E. Moreno, F. Hessel, V. Rosa, F. Moraes, and E. Carara, "From vhdl register transfer level to systemc transaction level modeling: A comparative case study," *IEEE International Symposium on Circuits and Systems*, 2003.

[5] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, "Principles of Communication Systems Simulation with Wireless Applications," first ed., 2004.

[6] M. Petrov, T. Murgan, P. Zipf, and M. Glesner, "Functional modeling techniques for a wireless lan ofdm transceiver," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 3973-3973, May 2005.

[7] O. S. I. (OSCI). Systemc. [Online]. Available: <http://www.systemc.org>.

[8] J. Booth, "Systemc modeling of a parallel processor broadcast interconnection system," *Southeast-Con, Proc IEEE*, pp. 76-81, 2002.

[9] A. Ghosh, S. Tjiang, and R. Chandra, "System modeling with systemc," in *ASIC*, 2001.

[10] T. Groetker, S. Liao, G. Martin, and S. Swan, "System design with SystemC," 2002.

[11] HP, HDLC Frame Protocol Users Guide, HP invent, 1 ed., February, 2004.

[12] M. Jeruchim, P. Balaban, and K. Shanmugan, "Simulation of communication systems - modeling, methodology and techniques," Kluwer Academic Publishers, 2002.



Ibrahim Aref received the B.Sc. degree from University of Tripoli, Tripoli, Libya, in 1993, The M.Sc. degree from University Putra Malaysia (UPM), Serdang, Selangor Darul Ehsan, Malaysia and the Ph. D. degree from the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, U.K., in 2011. He has a more than 18 years experience in research, academic, IT and development,

also highly proficient IT professional with proven analytical abilities and organizational skills, as well as 8 years of industry project experience involving leadership and consulting on object-oriented distributed information systems, enterprise application integration, hardware and software development processes, System-on-Chip design methodology, customer interaction, and end-user organization. Dr. Aref currently is a Lecturer with the Computer Engineering Department, University of Tripoli and has been a member of International Association of Computer Science and Information Technology (IACSIT).



Khaled Elgaid received the B.Sc. degree from the University of Evansville, IN, in 1984, the M.Sc. degree from Marshall University, Huntington, WV, in 1986, the M.Sc. degree from the University of Cincinnati, OH, in 1989, and Ph.D. degree from the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, U.K., in 1998. He has authored or coauthored more than 70 journal and conference

papers. He has more than 12 years of experience in the design, fabrication, and on-wafer characterization of MMICs and millimeter-wave active and passive devices for imaging, radar, and communications applications. He has extensive experience in three-dimensional lithography using both e-beam and photolithography. Dr. Elgaid is a Lecturer with the Electronic Design Centre, Department of Electronics and Electrical Engineering Department, University of Glasgow, leading group in the area of millimeter-wave monolithic integrated circuits (MMICs).