International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Design and Performance Verification of High-Speed Mobile Data Transmission Application based Switching System using VHDL

Shlok Kumar, Uma Yadav
Department of VLSI Engineering,
Shri Venkateshwara University, School of Engineering and Technology, U.P.
shlokiankur@gmail.com, Er.yadavuma11@gmail.com

Abstract: Switching system is used for data transmission between two communicating Channels. This transmission can be either via trunks (Landline) or it can be completely wireless, technically termed as Telephone Switching and Mobile switching, respectively. We report the implementation of a switching system using very high-speed integrated circuit hardware description language (VHDL) which is more reliable and efficient than the present switching system. It converts entire bulky switching unit which consists of routers, multiplexers, decoders, counters in to a single integrated circuit (IC). Simulation results are presented for transfer of data from input subscriber to the output subscriber using sequential write /random read mode with the timing diagram. To verify the transfer of data from input subscriber to the output subscriber a 32-bit op-code is assumed in which each bit represents a specific function. In this paper we work on Landline Switching for two different channels using VHDL each channel has 16 users.

Keywords: Opcode, Switching System, VLSI, VHDL.

1. Introduction:

The transmission of telegraphic signals over wires marked the first major advancement in modern communications technology. Telegraphy was introduced in Great Britain in 1837 and in France in 1845. In March 1876, Alexander Graham Bell demonstrated his telephone, showcasing the potential for long-distance voice transmission. Bell's demonstration included a point-to-point telephone connection. Figure 1 below illustrates a network utilizing such a point-to-point connection.

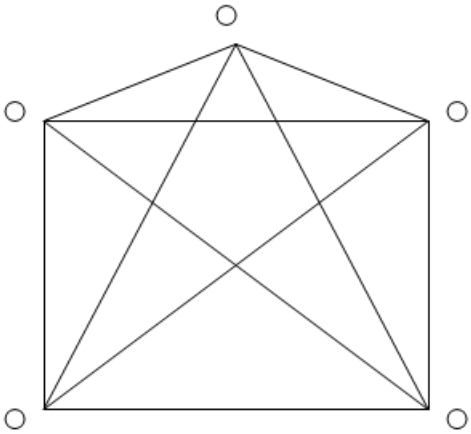

Fig 1: A network with point-to-point connection

In a general case with n entities, there are $n(n-1)/2$ links. Therefore, as nn increases, the number of links required for a fully connected system becomes very large. Consequently, the practical implementation of Bell's invention on a large or even moderate scale required not only the telephone sets and pairs of wires but also a switching system. With the introduction of the switching system, subscribers are connected to the system, which increases response speed and provides a more efficient way to establish connections. When a subscriber wants to communicate with another, a connection is established between the two at the switching system.

In this switching system, only one link per subscriber is needed between the subscriber and the switching system, and the total number of such links equals the number of subscribers connected to the system. Early switching systems were manual and operator-oriented. The limitations of these operator-oriented systems were quickly recognized, leading to the development of automatic switching systems, which can be classified into electromechanical and electronic types.

2. Related Work:

Rajeev Sivaram et al. proposed an appealing mechanism for efficiently executing multicast and other collective operations on direct networks. However, applying this mechanism to switch-based parallel systems presents significant challenges. In their work, they suggest alternative switch architectures with varying buffer organizations to implement multidestination worms on switch-based parallel systems. Initially, they address issues related to such implementation, such as deadlock-freedom, replication mechanisms, header encoding, and routing [10].

Subsequently, they demonstrate how an existing central-buffer-based switch architecture, originally supporting unicast message passing, can be augmented to facilitate multidestination message passing. Likewise, they discuss implementing multidestination worms on an input-buffer-based switch architecture, presenting two architectural alternatives that reduce wiring complexity in practical switch implementations. The central-buffer-based and input-buffer-based implementations are compared, along with corresponding software-based schemes, through simulation experiments under various traffic scenarios (multiple multicast, bimodal, varying degrees of multicast, and message length) and system sizes[9].

Their study reveals the superiority of the central-buffer-based switch architecture. Particularly, under bimodal traffic, the central buffer-based hardware multicast implementation demonstrates less adverse effects on background unicast traffic compared to a software-based multicast implementation. These findings illustrate that multidestination message passing can be readily and effectively applied to switch-based parallel systems, yielding commendable multicast and collective communication performance [8].

Furthermore, they present two switch architectures with differing buffer/queue organizations for implementing multidestination worms in switch-based parallel systems. They elaborate on how a central-buffer-based switch architecture, originally supporting only unicast message passing, can be adapted to support multidestination message passing with minimal additional cost. Similarly, they illustrate how an input-buffer-based switch architecture can be extended to support multidestination message passing and present two alternatives for implementing such an architecture, aimed at reducing wiring complexity [7].

Extensive simulations are conducted to evaluate the relative performance of the proposed switch architectures and compare the achieved hardware multicast performance with the best software multicast algorithm. To ensure fairness in comparisons, the high start-up overhead associated with software schemes is factored out. Their performance studies indicate that central-buffer-based switch architectures significantly enhance hardware multicast performance, delivering commendable performance across various applied loads, message lengths, multicast degrees, and system sizes [6].

Although the alternative implementation using an input FIFO buffer yields similar multicast performance to the central-buffer-based implementation for single multicasts and multiple multicasts at low loads, performance degradation is observed at relatively light loads. Ongoing investigations focus on enhancing the switch architecture to support reliable multicast, as well as efficient and reliable barrier synchronization. Additionally, the impact of these enhancements on benchmark application performance in the DM and DSM domains is being evaluated [5].

3. Methodology:

Phase 1: In the initial phase, the input subscribers within both exchanges undergo sequential scanning. This process requires 16 clock cycles to scan 32 subscribers, determining their transmission status. This sequential scanning enables the storage of data to be transmitted in the data memory sequentially. Additionally, information pertaining to the called subscriber is sequentially stored in the control memory, while the caller ID number is stored likewise in the caller ID memory. Thus, the system operates in a sequential write mode.

Phase 2: Upon completion of the scanning process, the data memory location is read based on the corresponding location in the control memory. For instance, if the first location of the data memory contains data 'd' and the corresponding location in the control memory is 2, 'd' will be communicated to the second user of the exchange, indicating random read functionality.

To determine the destination of the data, an 'I' bit in the opcode is utilized. If 'I' equals 1, indicating interexchange, the data will be routed to a user in the other exchange, facilitating communication between subscribers of different exchanges. Conversely, if 'I' equals 0, indicating intraexchange, the data will be transmitted to a user within the same exchange, enabling communication among subscribers of the same exchange.

Exchange between caller ID memories only occurs if the respective user is enabled. This condition also applies to the data memory. A user is deemed enabled if the 16th bit of their opcode is set to 1, and disabled if set to 0. Therefore, for a successful call, the caller must be enabled while the called party must be disabled. Communication entails the transfer of data (bits 0-15 of the opcode) between entities, depicted by overwriting the data bits if the recipient is the called subscriber. The caller ID feature allows the called user to identify the caller by examining the relevant bits of the opcode (bits 25 to 22). Figure 2 illustrates the structure of the switching system.

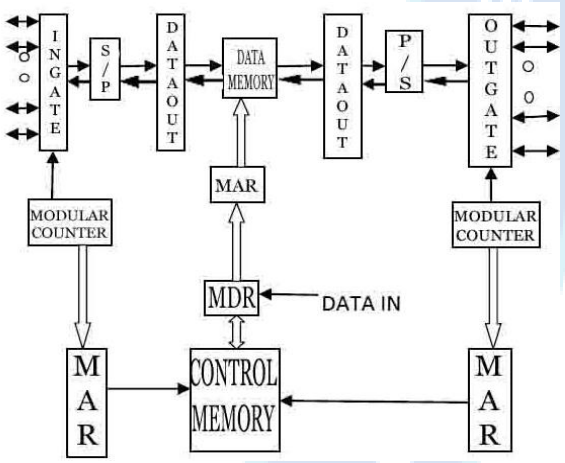

Fig 2: Dual Way Structure of Switching System

4. Result and Discussion:

There are two exchange centers denoted as 'din' and 'dout'. Our implementation covers both inter-exchange and intra-exchange scenarios. Setting 'I' to 1 activates the inter-exchange mode, while setting 'I' to 0 activates the intra-exchange mode. Our design relies on additional inputs: clock, reset, and enable. When the reset signal is high, the system remains inactive; all functions are executed during a low reset state and on the positive edge of the clock signal. If the enable signal is low, data exchange does not occur. Our exchanges operate when the enable bit is set to high.

In this instance, we demonstrate inter-exchange functionality for the first exchange. Here, with the enable bit set to 1 and 'I' set to 1, the source address is 0101, the destination address is 0110, and the data for transmission is "AD01". We observe

the reception of this data at the second exchange, 'dout', located at the 6th address as depicted in Figure 3.

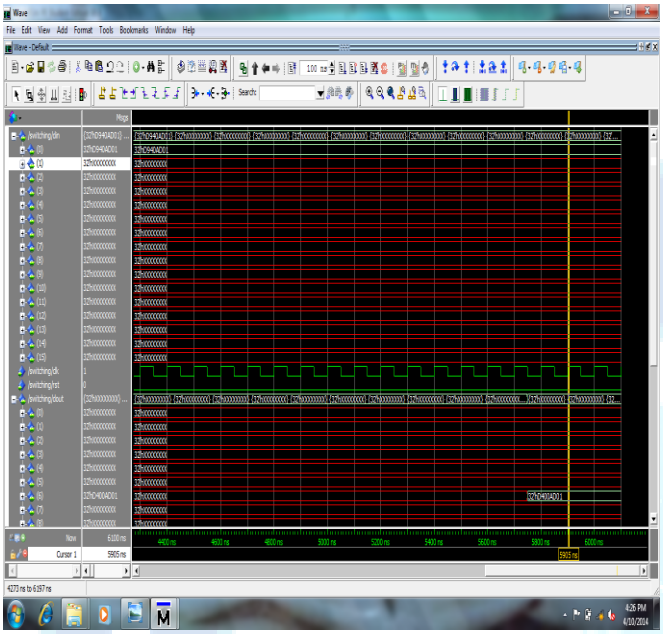

Fig 3: Simulation result for First Exchange (Inter Communication)

In this scenario, we demonstrate intra-exchange functionality for the first exchange. Here, with the enable bit set to 1 and 'I' set to 0, the source address is 0110, and the destination address is 0101. The data for transmission is "AD01". We observe the reception of this data at the first exchange, 'din', located at the 6th address as depicted in Figure 4.

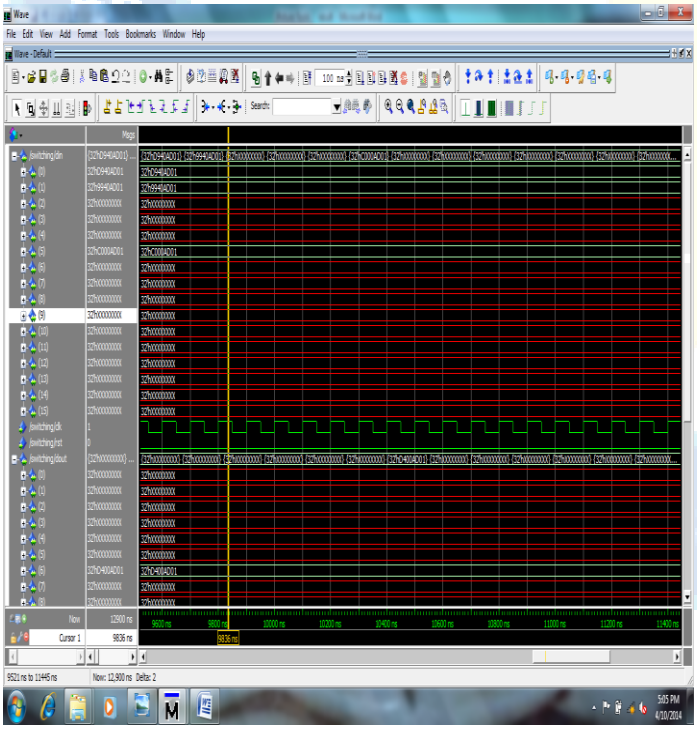

Fig 4: Simulation result for First Exchange (Intra Communication)

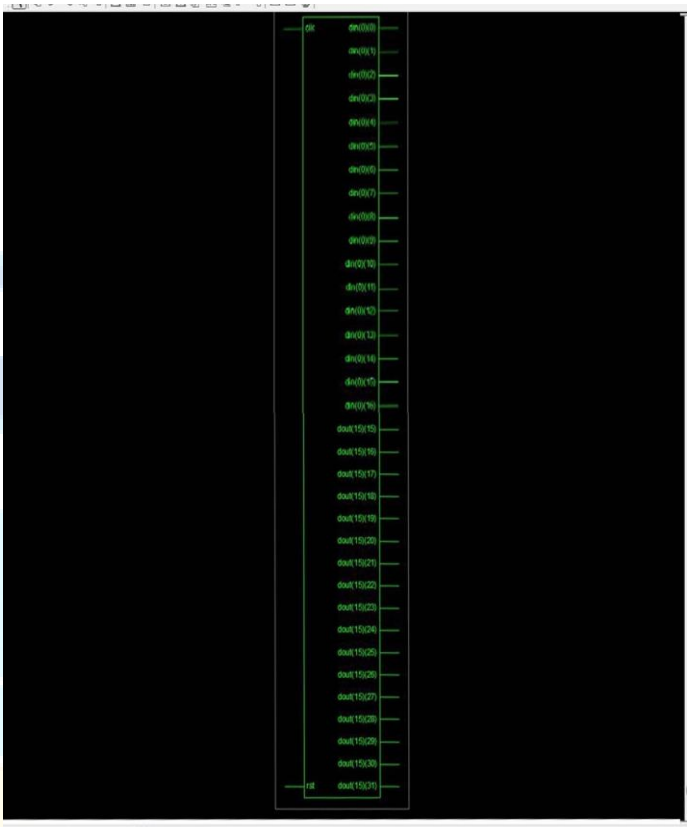

Fig 5: Main RTL

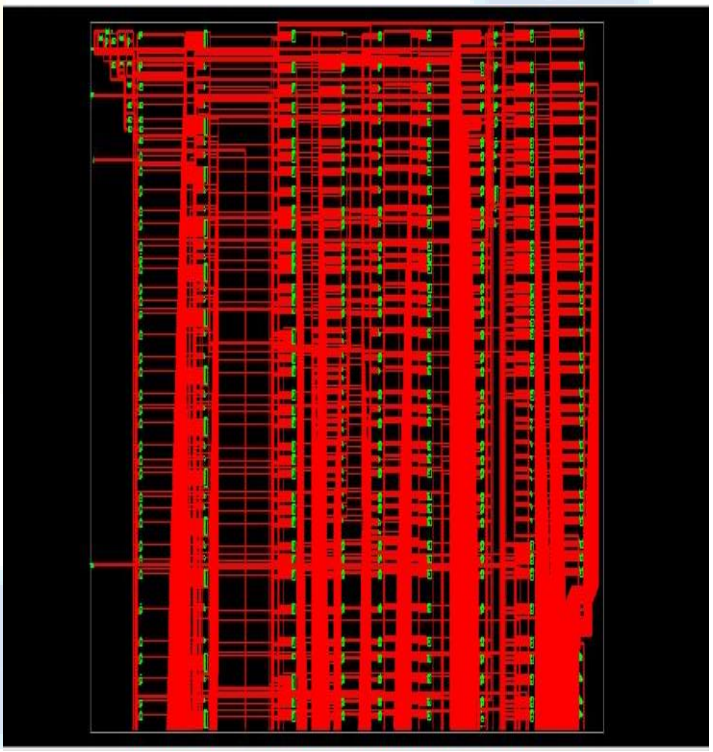

Fig 6: Internal RTL

5. Conclusion:

The aim of this work was to design ICs for a Switching System. Current switching systems utilize multiplexers, routers, and switches, which are inefficient due to their analog nature and high-power requirements. In contrast, we have aimed to digitize the entire system to enhance efficiency, reduce power consumption, and minimize delay. VHDL was

chosen for programming the ICs due to its user-friendly nature, making future modifications straightforward.

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, Institue 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] T. Groetker, S. Liao, G. Martin, and S. Swan, "System design with SystemC," 2002.
[8] HP, HDLC Frame Protocol Users Guide, HP invent, 1 ed., February, 2004.
[9] Maryam Keyvani et al, "VHDL IMPLEMENTATION OF A HIGH-SPEED SYMMETRIC CROSSBAR SWITCH" School of Engineering Science of SIMON FRASER UNIVERSITY August 2001.
[10] Rajeev Sivaram et al, "Implementing Multidestination Worms in Switch-Based Parallel Systems: Architectural Alternatives and Their Impact" , IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 11, NO. 8, AUGUST 2000.
[11] Krishna V. et al, "Energy Aware Computing through Probabilistic Switching: A Study of Limits" IEEE TRANSACTIONS ON COMPUTERS, VOL. 54, NO. 9, SEPTEMBER 2005.
[12] Ahmed F. Aref et al , "A Fully Integrated Adaptive Multiband Multimode Switching-Mode CMOS Power Amplifier" IEEE TRANSACTIONS ON MICROWAVE THE ORY AND TECHNIQUES, VOL. 60, NO. 8, AUGUST 2012.
[13] Dapeng Tian et al , "Wireless Haptic Communication Under Varying Delay by Switching-Channel Bilateral Control With Energy Monitor" IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL. 17, NO. 3, JUNE 2012.
[14] Prashant Rai et al, "VHDL Implementation of a prototype switching system" Department of ECE, Dehradun Institute of Technology, Mussoorie Diversion Road Dehradun, Uttarakhand-248009, India (CAC2S 2013).
[15] Gopinath Chakrabortty et al , "HUMAN NETWORK COMMUNICATION" International Conference on Information Systems and Computing (ICISC-2013), INDIA.