International Journal of Cybernetics & Systems
Volume 1, Issue 1, 2011, pp-04-06
Available online at: http://www.bioinfo.in/contents.php?id=113

Simulation of Bluetooth using VHDL

S.V. Awale and V.P. Pawar
Department of Physics/Electronics/Computer Science Maharashtra Udyagiri Mahavidyalaya, Udgir, India
e-mail: awalesv@rediffmail.com

Abstract—Each network device or portable device with HOST interface (radio or cable) has several hierarchical structures which distribute functionality between different layers of that device. The hierarchy defined radio interface, base band layer. During development of the Bluetooth module, a new complex architecture is integrated; it supports low cost upgrades related to new generations of Bluetooth devices. This paper presents the Bluetooth technology and its architecture. The base band layer performs the functions related to interface interaction between the Bluetooth chipset and an external or integrated radio chip and a host system. In base band, different modules for data processing are integrated, like modules for generating scrambling base band module of blue tooth. The development of the module has been done with hardware description language VHDL. This permitted us to perform top level functional verification and debugging, as well as detailed subsystem simulations throughout the design process.

Keywords: Bluetooth, base band module, scrambler.

I. INTRODUCTION

The wireless technology has been applied to telephony services and other applications. Bluetooth is a specification for short range wireless communication using 2.4 GHz ISM band. It emphasizes low complexity, low power and low cost. Bluetooth operating in the ISM band, is a specification for short-range wireless communication. It was developed to substitute cables connecting portable or desktop devices and to build low-cost wireless networks for mobile and portable devices. It emphasizes low complexity, low power consumption, and low cost. It is crucial to implement digital base band processing in a cheap, small module and desirable to integrate the whole system on a chip to achieve the power and cost targets. Base band modules as reusable intellectual property (IP) cores enable those higher levels of integration through system on a chip (SoC) design and reduce time-to-market. The Bluetooth base band module is in general responsible for carrying out link control and link management tasks. The detailed tasks of the module vary significantly depending on applications. For the simplest applications, such as wireless headsets and cellular phone add-on dangles, the entire application as well as a basic part of the base band layer protocols may be implemented in software on the base band processor[1]. For more complex applications expecting high- speed full base band operation and host controller interface (HCI), most of the base band protocols would be implemented in hardware while more complex upper layer protocols and application software are processed on a host processor. Allocating more functions to hardware from software can reduce the load/interrupt frequency, but the trade-off could produce a significant increase in gate count and loss of connection flexibility with a resultant poor interoperability performance. The base band module should therefore be very flexible so as not to waste the processing power and hardware resource. This paper describes an area-efficient digital base band module that is suitable for use as an IP core on SoC, ASICs. To gain more flexibility, we used a programmable embedded microcontroller optimized to our Bluetooth core. The programmable embedded microcontroller performs as many tasks for channel control and interface as possible. In addition, it supports firmware programming capability. It gives the overall architecture of our Bluetooth base band module and implementation of scrambling the Bluetooth base band module[2].

II. BLUETOOTH TECHNOLOGY

Bluetooth operates in the unlicensed 2.4 gigahertz (GHz) to 2.4835 GHz Industrial, Scientific, and Medical (ISM) frequency band. Numerous technologies operate in this band, including the IEEE 802.11b/g WLAN standard, making it somewhat crowded from the standpoint of the volume of wireless transmissions. Bluetooth employs frequency hopping spread spectrum (FHSS) technology for all transmissions. FHSS reduces interference and transmission errors and provides a limited level of transmission security. With FHSS technology, communications between Bluetooth devices use 79 different radio channels by hopping (i.e., changing) frequencies about 1600 times per second for data/voice links and 3200 times per second during page and inquiry scanning. A channel is used for a very short period (e.g. 625 microseconds for data/voice links), followed by a hop designated by a pre-determined pseudo-random sequence to another channel; this process is repeated continuously in the frequency-hopping sequence. The range of Bluetooth devices is characterized by three classes that define power management.

III. BLUETOOTH ARCHITECTURE

Bluetooth permits devices to establish either ad hoc or infrastructure networks. Infrastructure networks use fixed Bluetooth access points (AP), which facilitate communication between Bluetooth devices. This document focuses on ad hoc piconets, which are much

International Journal of Cybernetics & Systemics
Volume 1, Issue 1, 2011, pp-04-06
Available online at: http://www.bioinfo.in/contents.php?id=113

more common than infrastructure networks. Ad hoc networks provide easy connection establishment between mobile devices in the same physical area (e.g., the same room) without the use of any infrastructure devices. A Bluetooth client is simply a device with a Bluetooth radio and software incorporating the Bluetooth protocol stack and interfaces[3]. The Bluetooth specification provides separation of duties for performing stack functions between a host and a host controller. The host is responsible for the higher layer protocols, such as Logical Link Control and Adaptation Protocol (LCAP) and Service Discovery Protocol (SDP). The host functions are performed by a computing device like a laptop or desktop computer. The host controller is responsible for the lower layers, including the Radio, Base band, and Link Manager Protocol (LMP). The host controller functions are performed by an integrated or external (e.g., USB) Bluetooth dongle. The host and host controller send information to each other using the Host Controller Interface (HCI). Figure1 depicts the basic Bluetooth network topology. In a Pico net, one device serves as the master, with all other devices in the piconet acting as slaves. Piconets can scale to include up to seven active slave devices and up to 255 inactive slave devices[4].

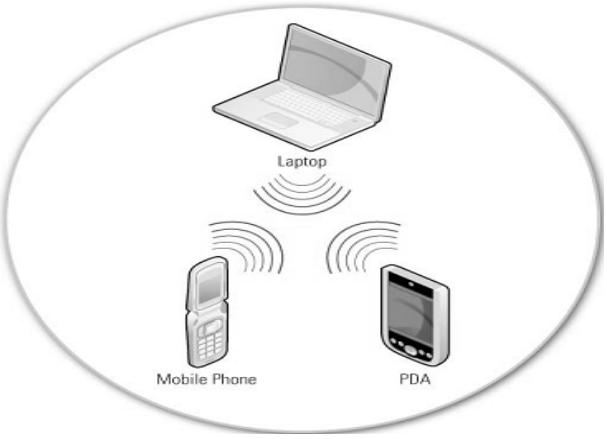

Fig.1 Bluetooth Ad Hoc Topology

The master device controls and establishes the network (including defining the network's frequency hopping scheme). Although only one device can serve as the master for each piconet, time division multiplexing (TDM) allows a slave in one piconet to act as the master for another piconet simultaneously, thus creating a chain of networks. This chain, called a scatternet, allows several devices to be networked over an extended distance in a dynamic topology that can change during any given session. As a device moves toward or away from the master device, the topology, and therefore the relationships of the devices in the immediate network, may change. Bluetooth uses a combination of packet-switching and circuit-switching technologies. The use of packet switching in Bluetooth allows devices to route multiple packets of information over the same data path. This method does not consume all the resources of a data path, thereby allowing Bluetooth devices to maintain data flow throughout a scatternet[5].

IV. SCRAMBLING BLOCK

This block performs the scrambling and descrambling of the header and payload for both the Rx and Tx directions so it has 3 Din inputs. As in the CRC block, the system controller must initialize it.

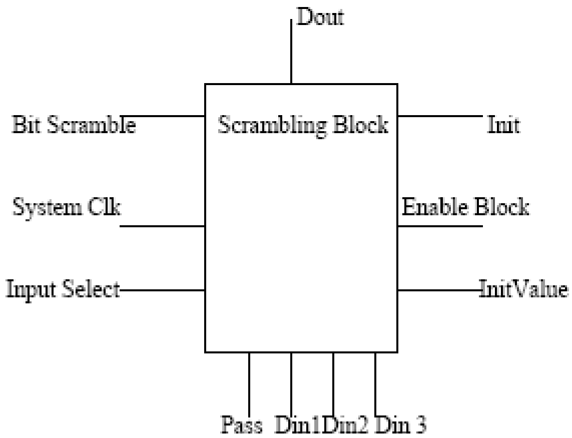

Fig. 2: Scramble Block

Scramblers are a class of substitution ciphers and have been found to be suitable for various security requirements such as those used by cable and satellite TV operators and mobile phone service providers. A Scrambler is a coding operation which basically randomizes the data streams. In addition to its use as a stream cipher, a scrambler is commonly used to avoid long strings of 0's and 1's which are responsible for DC wander and synchronization problems in communication circuits. Scramblers are very popular for their use to encrypt video and audio signals for broadcasting and many other applications. The low cost and complexity, high speed of operation and easy to use are the main features of scramblers.

V. VHDL IMPLEMENTATION

We moved the dedicated buffer blocks in the Bluetooth base band module into the FPGA. However, these operations may induce performance degradation in running an application and influence data transfer in the base band module. Therefore, the performance degradation of the base band module due to removing dedicated buffer blocks should be carefully analyzed to meet the performance specification. The figure 3 shows VHDL code performs the simulation of Bluetooth using VHDL.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;
entity scramble is
   Port ( clk : in std_logic;
          rst_n : in std_logic;
          din : in std_logic;
          init : in std_logic_vector ( 6 downto 0);
          load : in std_logic;
          dout : out std_logic);
end scramble;
architecture scrambler_beh of scramble is
            signal lfsr : std_logic_vector ( 6 downto 0);
begin
scrambler_Proc : process ( clk, rst_n)
begin
if rst_n = '0' then
            dout <= '0';
            lfsr <= (others => '0');
else if clk'event and clk = '1' then
            if load='1' then lfsr <= init;
            dout<= '0';
            else
            dout <= din xor lfsr(6);
lfsr( 6 downto 5 ) <= lfsr ( 5 downto 4);
lfsr(4) <= lfsr(3) xor lfsr(6);
lfsr( 3 downto 0) <= lfsr(2 downto 0) & lfsr(6);
end if;
end if;
end if;
end process scrambler_proc;
```

Fig. 3: VHDL Code for Scramble Bluetooth

VI. SIMULATION RESULTS AND VERIFICATION

The proposed scheme has been synthesized and simulated using Xilinx Software. The waveforms obtained at various check nodes have been investigated and found in conformity with the theoretical observations. Various waveforms obtained have been presented in figure 4.

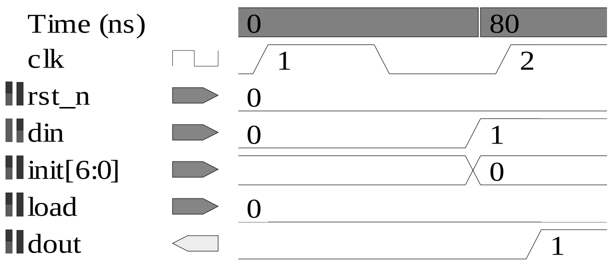

Fig. 4: Test Bench Waveforms

Clk and rst_n are input signals and represent clock (that drives the system) and the Plain text to be scrambled. Dout are output signals that denote scrambled signal and received text respectively. It is evident from the Fig. 3 that input data and received data is in complete agreement while as input data does not resemble the scrambled data.

VII. CONCLUSION

We have presented a small, flexible base band module for Bluetooth wireless communication. The complex control tasks of the Bluetooth base band layer protocols were implemented in software running on the embedded microcontroller. An efficient scheme for simulating the Bluetooth for base band data scrambling and descrambling for data transmission using VHDL has been proposed. The proposed scheme has been synthesized and simulated for target device of Spartan FPGA family. It has been found that the proposed scheme is capable of providing a range of applications in Spread Spectrum Modulation, Code Division Multiple Access and Global Positioning Systems. The proposed scheme can be synthesized and implemented on any of the existing CPLD and FPGA systems as per the degree of optimization required. The simulation and verification carried out at transmitting and receiving ends of the system has proved the efficacy of the proposed scheme. The results have been presented in the form of various waveforms.

REFERENCES

[1] I. V. Costov and K. Filipoya K., "Design of electronics using VHDL described with example of based band and LMP layers in Bluetooth", IEEE Electronics Technology, vol. 3, pp. 453-456, May 2004.
[2] R. Ahola, D. Wallner and M. Sida, "Bluetooth Transceiver Design with VHDL-AMS", in Proc. DAT, 2003, paper 1530-1591, p. 20268.
[3] http://www.icgst.com
[4] I. Chun, B. Kim and I. Park, "A Fully Synthesizable Bluetooth Base band module for a system-on-a-chip", ETRI Journal, vol. 25, pp. 328-336, Oct. 2003.
[5] G.M. Bhat, M. Mustafa, S. Ahmad and J. Ahmad, "VHDL modeling and simulation of data scrambler and descrambler for secure data communication", INDIAN Journal of Science and Technology, vol. 2, pp. 41-43, Oct. 2009.