COMPACT IMPLEMENTATION OF DSSS WAVEFORM USING XILINX ZYNQ SOC AND AD9361 TRANSCEIVER

Sharvani Gadgil[1], Atul Pawar[2], C D Naidu[3], M Haritha[4]
[1,3,4]VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad,
[2]DRDO,RCI,Hyderabad

Abstract

Direct-Sequence Spread-Spectrum (DS-SS) is a widely used technique for secured wireless data communication today. The DS-SS technique is used in 2G (IS-95) and 3G (CDMA-2000, WCDMA) cellular mobile communication systems. Further, the IEEE802.11 also use DS-SS signal for wireless LAN. The DS-SS is widely used because of its features like providing secured communication by low probability of interference and detection (LPI/LPD) capability, multiple access and selective availability provided by (CDMA), better anti-jam strength achieved by using higher length pseudo-random noise (PN) sequence. The main objective of project activity is implementation of DS-SS signal with latest ARM enabled embedded FPGA (Xilinx-Zynq) and state of art Analog device's wideband transceiver AD9361. The implementation work includes generation of Gold code using ARM based 'C' program and FPGA based Verilog HDL. This is realized in Xilinx-Zynq based system on chip (SoC) available on Zedboard using latest version of Xilinx EDK software that includes Xilinx Platform Studio (XPS) and Xilinx SDK. The embedded resources such as timers, UARTs, GPIOs and Gigabit Ethernet are utilized for DS-SS waveform generation. Various signals required for spreading and code generation are generated and analyzed on spectrum analyzer and an oscilloscope. The modulation is carried out in 2.4 GHz ISM frequency band. The various features of AD9361 transceiver IC are explored. The signal properties are tested on spectrum analyzer.

Index Terms: DS-SS, CDMA, Gold code, SOC.

I. INTRODUCTION

Direct sequence spread spectrum (DSSS) is a communication technique in which information bandwidth spreaded by direct modulation of signals by a wideband PN codes. The DSSS signal is then modulated by a carrier before final transmission. In DSSS, the base band signals are usually called bits, and the code bits are called chips. Typically, the baseband signal bandwidth is multiplied several times by the spreading signals. In other words, the chip rate is much higher than the bit rate. The spreading signal sequence is unique for a transmitter, and the same chip sequence is used at the receiver to re-construct the signals (data bits).This paper aims at generation of a DSSS waveform using a standard SDR platform. The basic block diagram of the transmitter is as shown in Fig.1. As shown in Fig1 the basic steps that are present in DSSS generation are multiplication of message data sequence by PN code (Gold code in our case), code generation and BPSK modulation by a carrier. This is implemented as shown in Fig.2 using Xilinx Zynq SoC that is available on a Zedboard.

The existing system used in military communications for the design of DSSS transmitter uses both a processing system (ARM) and an FPGA for waveform generation. Deriving motivation from this we propose a new

system using Xilinx Zynq SoC which consists of the processing system and the programming logic on the same chip. The basic block diagram shown in Fig2 that uses the Zynq is implemented in this paper. The next section describes in detail how every part of this system is implemented for DSSS waveform generation.

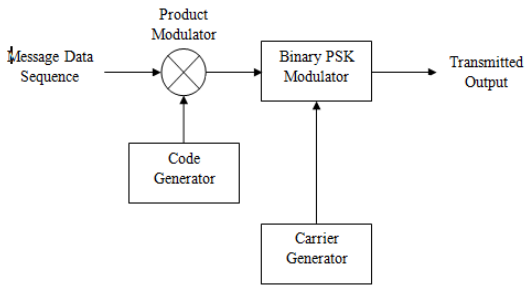

Fig.1 Basic Block diagram of DSSS transmitter

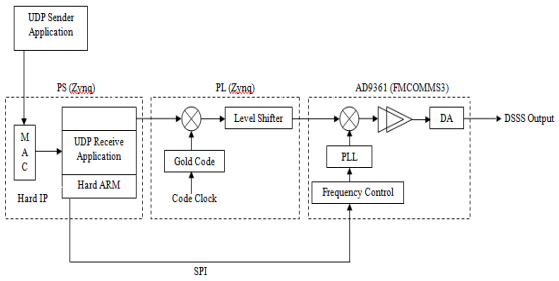

Fig.2 DSSS transmitter Implementation using Xilinx Zynq

A standard message data sequence that is to be transmitted, aka Interface control document, is received via the Ethernet, by a UDP sender application that runs on a PC which sends a packet every 100ms to the Zynq. This message is received by the board from the MAC (the board has a default IP address) by the Zynq processor with the help of a UDP receive application written for the ARM. Also the code that supplies the carrier frequency to the RF section (AD9361) is written in the Zynq processing system. The message data is 32 bytes long is framed in the required format by doing the bit stuffing which is important in order to achieve synchronization, and then sent to the Programming logic (PL) with a help of a custom IP peripheral. Once the data is sent to the PL, it is multiplied with the 1024 bit Gold code sequence that is generated with the help of a code clock, and hence the data spreading is done. This spreaded data is further transmitted to the AD9361 block where the BPSK modulation takes place in the 2.4GHz frequency band (the carrier frequency comes from the PS (Zynq)) and the final DSSS waveform hence obtained is analysed on a spectrum analyzer.

II. IMPLEMENTATION

A. Receiving Data on Ethernet and Data framing

Various embedded resources like UART, GPIOs, Ethernet and SPI etc are supported by the Zynq processor. Among these we use the Gigabit Ethernet in our work. Ethernet is the most widely used computer networking technologies. It is a link layer protocol that defines how data is transmitted in a network.

In our system, data comes to the Zedboard via Ethernet.UDP protocol is used, packets (one packet of 32 bytes) of data are sent with the help of a UDP sender application (basically a C program for sockets) written on the PC to the board (to the default IP address of the board).For the UDP receive application that needs to be written for the Zynq processor, we make use of a Lightweight IP applications article [1], which contains an echo server application provided by Xilinx. This application receives the data packet that is sent at consecutive intervals of time. Next this data packet is framed to be forwarded to the next section .For this data is bit stuffed and HDLC is done. Bit stuffing maintains transparency in a communication channel. It is a process in which non-information bits are added to affect synchronous transmission of information. Whenever in a data sequence, a sequence of 6 consecutive 1s is found, a 0 bit is inserted. Along with the bit stuffing, HDLC (High-level data link control) protocol is applied, according to which 0x7E sequence or 01111110 sequence is appended to the end of each packet.

Once the data framing, which includes bit stuffing and HDLC, is done for the data packet it is ready to be sent from the PS(Zynq processor) to the PL(programming logic of Zynq).

B. Dataflow from PS to PL

The data packet that is received on the Ethernet port by the board is framed in the processor and next is to be sent to the PL where the spreading is done. For this a custom IP peripheral is created

in the Programmable logic part of the Zynq. We can talk to this peripheral via one of the ARM cores. The custom peripheral acts as a bridge between the processor and the PL of the Zynq. A custom peripheral named encoder is created using the XPS (Xilinx Platform studio) software. While creating the peripheral we choose the number of registers that we might need for our job. Once the peripheral is created it maps to the ARM processor with a base address which can be seen in the XPS. We can start reading or writing data into these registers at this base address.

The tasks to be performed by the PL are to receive the data packet, generate code, obtain spreaded data and send it to the RF block. The PL receives the data from the PS with the help of the custom peripheral in the form of an array of some considerable length. In our case since we are sending a packet of 32 bytes after the data framing we need an array of length of about 320 bits. Each register is 32bit long so we need 10 registers to contain the entire data packet. The code for receiving the data, Gold code generation and data spreading is written in Verilog in the HDL files that are generated in the created custom peripheral and the HDL files related to the RF block.

Spreading of data is an integral part of the DSSS transmitter, which is done by way of multiplying the incoming data packet with a code. In our system we use Gold codes. A gold code is a binary sequence that is formed by EX-ORing two m-sequences. The number of bits in a Gold code is 2n - 1, where n is the length of the LFSR (Linear Feedback Shift Register) that is used to generate each m-sequence. For generating Gold codes with good cross correlation properties a preferred pair of m-sequences need to be taken. The preferred pair of polynomials also known as the generator polynomials that are selected for our system with n=10(we use 10 bit LFSRs to generate gold code) are

1+x^2 +x^9 and 1+ x + x^2 + x^5 +x^7 +x^8 + x^9 as illustrated in the Fig.3 of the Gold code generator.

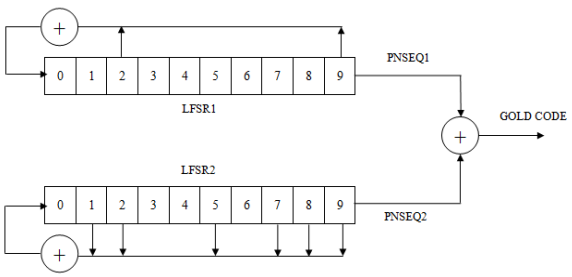

Fig.3 Gold Code Generator

As shown in the Fig.3, the coefficients in the preferred polynomials of each LFSR are EX-ORed and feedback to it. Each LFSR generates a PRN sequence. The two PRN sequences hence generated by both the LFSRs are again EX-ORed to output the desired code (Gold code). This code hence generated by the by the Gold code generator is multiplied with the incoming data packet in the created custom peripheral and the result is the spreaded data that is further sent to the RF block for modulation.

C. Dataflow from PL to AD9361

The spreaded data that is obtained by spreading the incoming data packet is sent to the AD9361 Transceiver IC. It goes as input to the DAC (Digital to Analog Converter) of the AD9361.We use the AD9361 as a transmitter. A data clock for the incoming data packet and a code clock for code generation is to be set. Since the code is 1023 bits long, the code clock is divided by 1023 times to obtain a data clock. The code for clock generation is also written in Verilog. The DAC takes 12 bit input and the spreaded data which comes in digital form is converted to analog. Now this signal is ready for modulation. It is BPSK (Binary Phase Shift Keying) modulated using a carrier that is generated using the Zynq processor. This carrier of frequency 2.4GHz that comes to the AD9361 is SPI (Serial Peripheral Interface) controlled. After the modulation is done the signal goes through the transmit chain of the RF block and finally is output from one of the TX channels which is connected to the Spectrum analyzer on which we can see out desired DSSS waveform. This completes the DSSS waveform generation implementation using the Xilinx Zynq and AD9361 Transceiver.

III. TEST RESULTS

The final result of system is a DSSS waveform that is seen on a Spectrum analyzer. Fig4 shows the test bench setup that includes a PC, the Zedboard, the AD9361 transceiver and the Spectrum analyzer.

Fig5 shows the Zedboard connected to the AD9361 board. The programming of the onboard Zynq is done using a Xilinx JTAG programmer. The modulating data comes through the Ethernet cable connected to the Ethernet port on the Zedboard. An onboard USB/UART used to see the messages and events on a terminal window. The UART is also connected as a USB to the board which contains a USB to UART converter.

The Fig6 shows the simulation result for the Gold code generation, data clock and code clock and for the transmission of data from PL to the RF block as seen on the Xilinx ISE Design suite simulator.

The final DSSS output waveform that is generated at 2.4GHz frequency for the incoming data packet is as shown in Fig7

Fig.4 The testbench setup for DSSS transmitter Implementation

Fig.5 The Zedboard and AD9361 Transceiver

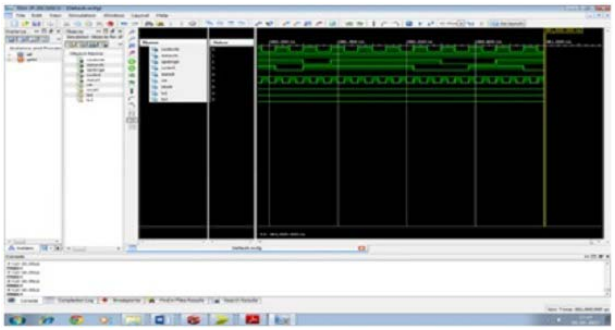

Fig.6 Simulation Result

Fig.7 The Final output DSSS waveform on Spectrum Analyser

IV. CONCLUSION

Implementation of a complete DSSS transmitter is done with the help of Xilinx Zynq and AD9361 transceiver that can be used in military communication for secure transmission of data. Typically a DSSS transmitter system uses UART as embedded resource, an ARM processor and a Spartan 3E FPGA. The proposed system implemented in this paper realizes the

same system on a Xilinx Zynq All programmable SoC that can replace the ARM processor and FPGA present in the typical system hence providing for being able to do the same job of DSSS waveform generation in a lesser space making it compact. Further the proposed system uses Ethernet for data transfer which has comparatively more benefits than that of the conventional UART serial communication medium. The Zynq board receives data packets on the Ethernet every 100ms which are framed in the processing system and then passed to the Programming logic of the Zynq where the PN code generation and data spreading is done. The spreaded data goes to the RF transceiver AD9361 where BPSK modulation is done at 2.4GHz frequency. The final DSSS waveform generation is completed and it is analyzed on a Spectrum analyzer. A DSSS transmitter using the Zynq SoC, therefore, is designed in the work presented in this paper.

V. ACKNOWLEDGEMENT

We express our sincere thanks to the management of VNR Vignyana Jyothi college of Engineering and Technology for their esteemed guidance and support towards this research.
We express our heartfelt gratitude towards RCI lab, DRDO, Hyderabad for the expert technical guidance and for the logistical support for the completion of this work.

REFERENCES

[1] The Xilinx website ,Online.Available :https://www.xilinx.com/support/documentation /application_notes/xapp1026.pdf/.
[2] Mahalinga V.Mandi, Ramesh S, Santhosh Kulal, Santosh Kumar S, Dileep D and Yajnesh Padiyar *An FPGA Implementation of a Pseudo-Chaotic Direct Sequence Spread Spectrum (DS-SS) Communication System* , ISSN 1749-3889 (print). 1749-3897,International Journal of Nonlinear Science,Vol.8(2009) No.4,pp.387-401.
[3] Sweta Malviya,Poonam Kumari *Implementation of Pseudo-Noise Sequence Generator on FPGA Using Verilog*,ISSN 0974-2174 . International Journal of Electronic and Electrical Engineering,Volume 7, Number 8 (2014), pg.887-892.
[4] Amsavalli A, Kashwan K. R *FPGA Implementation of Low Complexity VLSI Architecture for DS-CDMA Communication System* International Journal of Computer Applications, 0975 â˜A ¸S 8887,Volume 42,No 20, March 2012.
[5] W. A. Mahmoud, Hikmat N. Abdullah, J. M. Al-Samarei*Simulation of a DS/SS System Using MATLAB-Simulink* www.researchgate.net, Publication No 234774095.
[6] Md. Sohrab Ansari, Mathew Oommen, Velmurugan.S*Review on Design and Implementation of DSSS-CDMA Transmitter using HDLwith Raised Cosine Filter to Minimize ISI* ,IJCSMC, Vol. 3, Issue. 3, March 2014, pg.837-844.
[7] Adina Matache, Esteban L. Valles*Performance of decoder-based algorithms for signal synchronization for DSSS waveforms* ,Aerospace Conference, 2013 IEEE,2-9 March 2013.
[8] A. Maghawry,E. Eldiwany*FPGA-based MSK DS-SS modulator for digital satellite communications* Radio Science Conference, 2009. NRSC 2009.17-19 March 2009.
[9] A. Maghawry,E. Eldiwany*FPGA-based MSK DS-SS modulator for digital satellite communications* Radio Science Conference, 2009. NRSC 2009.17-19 March 2009.
[10] Fei-fei Chen, Li-xin Wang*SCR-Based DSSS Waveform Application Development*, International Conference on Computational Intelligence and Software Engineering, 2009.11-13 Dec 2009.
[11] Gao Yu *A Synchronization Design of Double-PN QPSK-DSSS System* , International Conference on Internet Technology and Applications (iTAP), 2011.16-18 Aug 2011.
[12] M. Hasan, J. M. Thakur, and P. Podder *Design and Implementation of FHSS and DSSS for Secure Data Transmission*, International Journal of Signal Processing systems Vol. 4,No.2,April 2016.
[13] John Proakis, *Digital Communications*, 5th Edition, McGrawHill Publications.