Journal of ELECTRICAL ENGINEERING, VOL. 60, NO. 1, 2009, 43–47

THE DESIGN OF A PLC MODEM AND ITS IMPLEMENTATION USING FPGA CIRCUITS

Rastislav Róka *

For the exploitation of PLC modems, it is necessary to have a detailed knowledge about the PLC transmitter/receiver circuits and their activities. In the first part of this contribution, various alternatives for the design of complex circuits are introduced together with the verification and testing of the VHDL code in the Matlab programming environment. Next, a simulation of the designed PLC modem activity by means of the Simulink is presented. The main part of the contribution is focused on the specification of processes that are required for time synchronization of separated blocks in the OFDM receiver. Finally, the main blocks of the OFDM receiver that are necessary in the case of its hardware realization are briefly discussed.

K e y w o r d s: circuit design, PLC modem, time synchronization of the OFDM receiver

1 INTRODUCTION

For the design of the PLC modem, flexible radio LAN (Local Area Network) standards HiperLAN/2 [1], [2] will be used. These standards are created for various types of broadband radio access networks BRAN (Broadband Radio Access Network), *eg* a 3G mobile network, ATM and IP networks, a wireless LAN network.

For digital signal processing in the PLC modem and for emulating of the ASIC circuit, we will use the programmable logical XILINX FPGA (Field Programmable Gate Array) circuits. The physical layer of the PLC modem is implemented by means of the ISE programming interface and the VHDL description language on the XILINX FPGA Spartan-3 Starter Kit set. An implementation of the PLC transmitter/receiver will be tested through the VHDL ModelSim simulation. General simulation of the PLC modem functionality is examined using the MATLAB/Simulink and the "Link for ModelSim in MATLAB" tools.

A family of new PLD (Programmable Logic Device) equipment with the FPGA design presents a structure of controlled logical cells or modules and their interconnections. There are two basic types of FPGA circuits, which differ in logical cell implementations and in the mechanisms of equipment interconnections:

- *SRAM* (*Static Random Access Memory*) circuits are re-programmable. A disadvantage of these FPGA circuits is a loss of circuit designs at the power supply switch- off. Therefore, they need a PROM (Programmable Read Only Memory) memory for configuration savings. Instead of logical cells, a table of possible values LUT (Look Up Table) is created. SRAM bits are applied for creating interconnections between equipment in the FPGA circuit.
- *OTP* (*One-Time Programmable*) *circuits* use permanent interconnections in the circuit. Therefore, these FPGA circuits do not require utilization of the PROM memory for configuration savings. However, a disadvantage of this solution is that we must interchange the whole FPGA circuit in the case of configuration changes.

2 THEORETICAL PART

2.1 Alternatives for a design of complex circuits

Utilization of a set of components is a traditional method for specifying the gate arrays and PLD devices. It represents a graphical tool that allows specifying logical gates and their interconnections. The last step of the circuit design is the so-called "Netlist file translation". However, one of large disadvantages of this schematic circuit design is the impossibility to realize a design of complex circuits. Moreover, this design method is strongly dependent on component manufacturers.

On present days, description languages of the HDL (High Description Language) type are used for description of complex circuits. HDL programs emulate the activity of digital systems. Several levels of abstraction can represent digital systems. One of large advantages of these description languages is the ability to integrate time specifications for system components. This ability allows to maintain the design of complex circuits in the transparent state. Since these description languages represent an activity of circuits, they are independent of logical gates implemented by manufacturers. Today, two main description languages — VHDL and Verilog are especially used for description of complex circuits. Both languages are very similar and effective. The VHDL (VHSIC (Very High Speed Integrated Circuits) Hardware Description Language) is specified as the IEEE standard 1076-1993 [3] and today it is one of the most used description languages for designing of highly integrated digital circuits.

* Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia, rastislav.roka@stuba.sk

ISSN 1335-3632

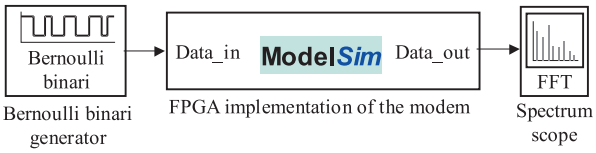

Fig. 1. The simple block model for the verification and testing of the VHDL circuit design by means of the "Link for ModelSim" tool

The VHDL language allows to describe digital systems on two levels of abstraction — structural and functional. Functional description can be further divided into types — data flow and algorithmic (sequential). Using this description language, every entity (an autonomous integrated function) is established. Particular entities can be mutually interconnected through the use of syntactic scripts and by means of "port map" relations. After mapping all inputs and outputs, the circuit design can be simulated. Thereby, its function is verified and proven.

Another way for description of complex circuits is the use of the Intellectual Property (IP) kernels. These represent certified complex functions that are utilized at the implementation of circuit designs. IP kernels are in reality macros specified by manufacturers for several description languages. These IP kernels are attached to a circuit design at the implementation of some components, *eg* the PCI bus or DSP filters *etc.*

2.2 The verification and testing of circuits designed in the VHDL code

After the phase of circuit design, the phase of its implementation is coming. Now, the designed circuit is verified and tested by means of an appropriate circuit simulator, *eg*, ModelSim.

The Mathsoft company implemented into the Matlab v.7.1 programming environment a tool that is called "Link for ModelSim" (Fig. 1). The technique of verification or simulation of the VHDL circuit design by means of the ModelSim tool is called the VHDL cosimulation. Between Matlab and ModelSim tools, communication is generated that serves as data transmission. This communication can be realized by the TCP/IP connection that is useful if the ModelSim tool is installed on a networked computer. For this communication, a sharing memory can also be used. Alternatively, we use the Simulink as part of the Matlab program that provides a more interactive form of simulation. In this case, a nearly realistic behaviour of the PLC modem can be proven. However, very substantial adjustment is required.

At the functional simulation, a correct output of the designed circuit is tested, *ie*, if its behaviour is in compliance with the circuit specification. Since there is a possibility for testing the designed circuit before saving its configuration into the PROM memory, it is possible to return to the circuit design and to revise it. Another possibility is also the use of limitations that later assist at the implementation. If the simulation is in progress successfully, phase "Netlist file translation" is executed and optimizations of equipment according to manufacturer's specifications are realized. After this phase, the phase "Place and Route" is in progress, *ie*, location and identification of interconnections are realized. Process "Place" is dedicated to selecting the specific modules and logical blocks for equipment that will be used and process "Route" is intended for finding physical interconnections between the above-mentioned modules on the equipment. This phase is one of the most complicated and is very time-consuming. Finally, the created configuration (a bit string) is placed and saved into the PROM memory, where from the FPGA circuit is programmed.

3 EXPERIMENTAL PART

3.1 Simulation of the designed PLC modem by means of the Simulink

In the Simulink environment, we created a model for PLC modem simulation (Fig. 2). The simple model allows to present utilized modulations together with channel interferences (reference channels) according to [4]. This Simulink model includes a transmitting part as well as a receiving part. In the transmitting part, the Bernoulli block represents a binary data generator with the Bernoulli probability. Data strings are then modulated by the orthogonal QAM modulator for baseband frequencies with mapping of one bit per symbol. After the QAM modulation, the modulated symbols are coupled with zero symbols to create the pre-OFDM symbol. At the start and at the end of the OFDM symbol, zero symbols are added primarily for start-up edges of the OFDM symbol transmitting that are defined in [2]. Thereafter, the IFFT transformation is executed for translating data into the OFDM symbol. For a given OFDM symbol, a duplicated copy of this symbol called the CP (cyclic prefix) is superadded. Then, particular samples pass through the multipath Rayleigh fading channel that simulates the PLC channel with coefficients proposed in a previous work [5]. In the receiving part, removing of the CP, the FFT transformation, removing of zero symbols and finally displaying of constellations for particular symbols are integrated. For presentation of the OFDM symbol after channel transmission, the "Spectrum Scope" block that shows the frequency spectrum of the received signal is utilized.

The transmitter of OFDM symbols in the PLC modem is specified in standards [1], [2] and its modified sections represent basic components of the receiver. At the OFDM receiver side, a very important part for a design of the time synchronization is added. It is a more complicated part than in the case of the transmitter because it is not specified in the above-mentioned standards. Time synchronization in equipment utilizing the OFDM modulation is one of the most complex and complicated parts of the FPGA circuit design. The reason is that OFDM

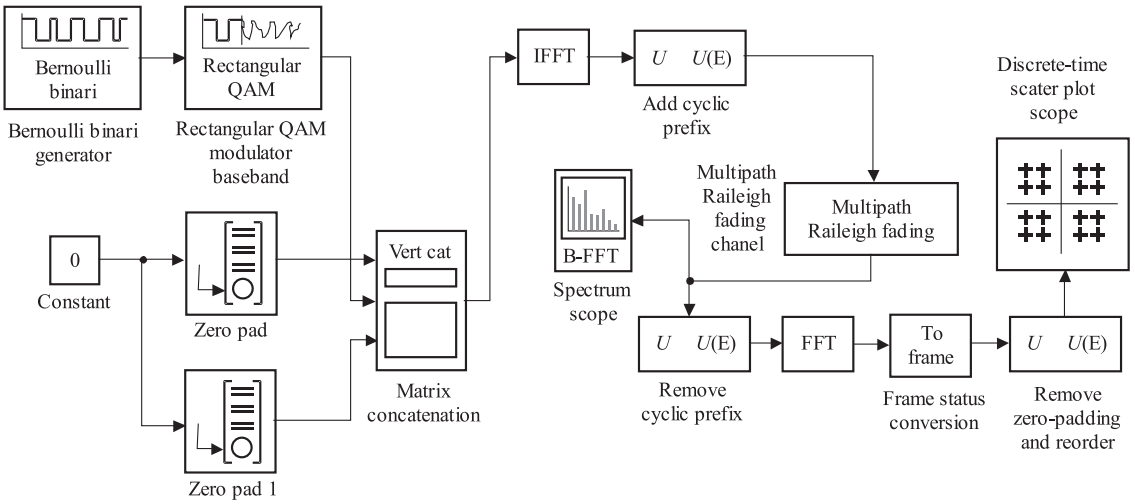

Fig. 2. The simulation model of the PLC modem in the Simulink environment

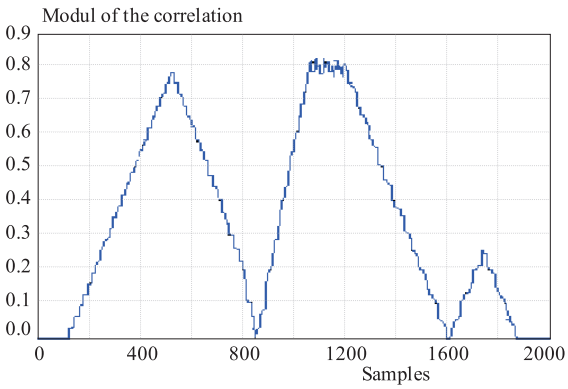

Fig. 3. The autocorrelation of the broadcast preamble for a distortionless signal

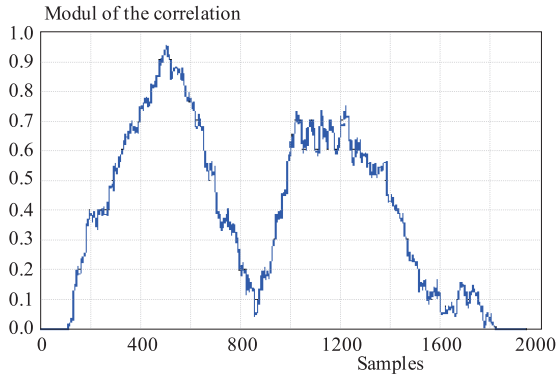

Fig. 4. The autocorrelation of the OFDM symbol at the SNR=5dB

modulation is very sensitive to frequency shifts that can cause waste of the OFDM symbol orthogonality. Time synchronization alone is composed of these sections:

1. Frame detection
2. Determination of the carrier frequency shift and its correction
3. Channel estimation

3.2 Frame detection

Frame detection is one of the most important sections of the PLC receiver because two PLC modems can communicate with each other only after correct frame detection. For frame detecting, the algorithm introduced in [6] was used. The received signal is correlated by the delayed part of its copy (autocorrelation). The length of the window used for performing of the correlation is equal to the length of the short symbol. The result of this transaction at the maximum value of the correlation represents a determination of the maximum module size and also the AGC (Automatic Gain Control) adaptation. If the autocorrelation module size is larger than the threshold value, the system is justified to receive the second part from the preamble and to recognize time synchronization of the frame. In Fig. 3, autocorrelation of the broadcast preamble is displayed.

In Fig. 4, autocorrelation of the OFDM symbol in the presence of the AWGN (Additive White Gaussian Noise) noise with SNR=5dB is presented. As we can see, signal degradations at the channel transmission occur.

In time synchronization, the maximum value of correlation is found. In the presence of noise it could happen that the accurate time instant cannot be obtained. However, the task is to estimate the time synchronization of OFDM symbols as good as possible. In Fig. 5, synchronization of the received data with transmitted data at various SNR values is shown.

For SNR values below 15 dB, the system does not synchronize at all. The reason is that a pseudorandom sequence with good correlation properties is taken over from a standard for radio environment [2] and so it was not matched for utilizing in the power distribution lines environment of the PLC modem.

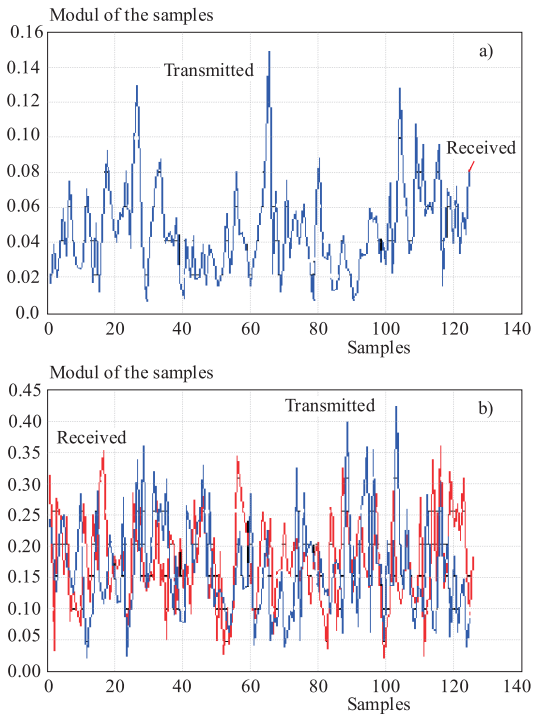

Fig. 5. The synchronization of received data with transmitted data at the SNR = 30 dB (a) and at the SNR = 15 dB (b)

3.3 Determination of the carrier frequency's shift and its correction

The frequency shift originates as a consequence of non-synchronization between the transmitter and receiver timing signals. The receiving signal is sampled with the same sampling frequency, but with a different phase. Or, the sampling frequency at the receiver is shifted up by a specific value Δf in comparison with the transmitter. For determination of the frequency shift, the HiperLAN/2 standard specifies the short OFDM symbol (element B) that follows the element A [2]. These elements are identical and determination of the frequency shift is executed after correlation of these received symbols.

Determination of the frequency shift can be divided into two parts — coarse determination and fine determination. Coarse determination is based on finding the frequency shift by means of autocorrelation of short symbols (A, B). Fine determination is calculated from the symbol (element C) following short symbols. After coarse determination, correction of symbols in the element C is performed and thereafter the shift on symbols in element C is found. At work with element C, autocorrelation with a one-symbol delay (*ie*, 512 samples) is used. Also, summation of particular correlation parts is done in the 1024 pattern length window.

The effect of the frequency shift offset affects the system and can be measured and expressed using the SNR_{LOSS} parameter. For relatively small offsets, the SNR_{LOSS} can be defined according to [4]

$$SNR_{LOSS} = \frac{10}{3\ln 10}(\pi T \Delta f)^2 \frac{E_S}{N_0} \quad \text{(dB)} \qquad (1)$$

where Δf is the frequency shift. Since time synchronization of the OFDM symbol can only be accomplished for very high SNR values, the coarse frequency shift is already in the order of units of Hz.

3.4 Channel estimation

In this part, the frequency response of the transmission channel is assigned. The frequency response of the transmission channel after executing the FFT transformation can be calculated using the following formula [4]

$$\widehat{H} = \frac{C_O}{C_R} \qquad (2)$$

where C_O is the sequence of the receiver and C_R is the sequence of the received data.

4 REALIZATION

On the basis of simulations of the PLC modem circuits designed by means of the VHDL description language, a PLC modem hardware prototype was realized. Whereas the structure of the OFDM modulation transmitter is based on the standard, the structure of the complete OFDM modulation receiver is shown in Fig. 6.

The block of time synchronization is shown in Fig. 7. The autocorrelation part is formed by the FIFO memory for signal propagation delay, by a function of the sample conjugation and multiplication of two samples. The result of this operation is inserted into the accumulator for averaging of given values. The resulting value is then utilized for frame detecting. Evidently, this functional logic is driven by a state diagram that after the transition through the threshold value travels to a state for determination of symbol bounds. Also, the CORDIC algorithm working in a vector mode performs coarse determination of the frequency shift. It means that the input is presented by a complex number with real and imaginary components and the output is the value of the angle. In this way, the value of the frequency shift is acquired.

Especially, functional blocks FFT, FIFO, and others IP which must be adjusted at data receiving, form other parts of the OFDM receiver. Data are read from a computer thanks to the USB interface that also includes a state diagram. The data are transmitted parallel by the EPP protocol. The data transmission rate depends on the implementation of the state diagram (engine, machine).

5 CONCLUSION

In this contribution, various alternatives for the design of complex circuits were introduced together with the verification and testing of the VHDL code in the Matlab programming environment. For this work, we choose the VHDL description language because of its very good availability for the design of complex circuits. Following,

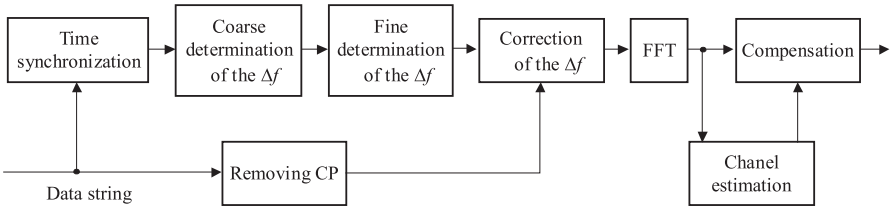

Fig. 6. The block scheme of the OFDM modulation receiver

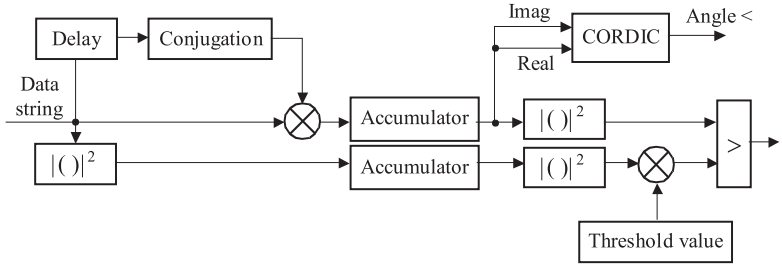

Fig. 7. The block of the time synchronization of the OFDM modulation receiver

simulation of the designed PLC modem activity by means of the Simulink was presented. Simulation of the designed circuits in the Simulink can bring the verification of their activity. However, hardware realization of the PLC modem or construction of both PLC transmitter and receiver parts means real problems that are not foreseeable in simulation. Therefore, this is the most important part of the PLC modem implementation by the FPGA circuits.

The main part of the contribution was focused on the specification of processes that are required for time synchronization of separated blocks in the OFDM receiver. Also, the main blocks of the OFDM receiver that are necessary in the case of its hardware realization were shortly discussed. It is necessary to remind that the structure of the configured OFDM receiver represents a base for other upgrading of the presented block scheme that will be focused on in future.

Appendix – Abbreviations

AGC	Automatic Gain Control
AWGN	Additive White Gaussian Noise
BRAN	Broadband Radio Access Networks
CP	Cyclic Prefix
EPP	Extensible Provisioning Protocol
FFT	Fast Fourier Transformation
FPGA	Field Programmable Gate Array
HDL	High Description Language
IFFT	Inverse Fast Fourier Transformation
IP	Intellectual Property
LAN	Local Area Network
LUT	Look Up Table
OFDM	Orthogonal Frequency Division Multiplexing
OTP	One-Time Programmable
PLC	Power Line Communication
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
QAM	Quadrature Amplitude Modulation
SNR	Signal-to-Noise Ratio
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

References

[1] ETSI Technical specification TS 101 475 v1.1.1: Broadband Radio Access Networks (BRAN), HIPERLAN Type 2, Physical (PHY) Layer, April 2000.

[2] ETSI Technical specification TS 101 761-1 v1.3.1: Broadband Radio Access Networks (BRAN), HIPERLAN Type 2, Data Link Control (DLC) Layer, Part 1: Basic Data Transport Functions, December 2001.

[3] IEEE Standard 1076-1993: VHDL Language Reference Manual, 1993.

[4] URMINSKÝ, J.: The Implementation of the PLC Modem Using the FPGA on the Base of the HiperLAN/2 Protocol, FEI STU Bratislava, December 2005, The MSc Project.

[5] RÓKA, R.–URMINSKÝ, J.: Experimental Measurements for Verification of the Parametric Model for Reference Channels in the Real PLC Environment, J. Electrical Engineering - Elektrotechnický časopis **59** (2008), 146-153.

[6] SCHMIDL, T.–COX, D.: Robust Frequency and Timing Synchronization for OFDM, IEEE Transactions on Communications (December 1997), 1613–1621.

Received 13 September 2007

Rastislav Róka (Ing, PhD) was born in Šaľa, Slovakia on January 27, 1972. He received his MSc and PhD degrees in telecommunication from the Slovak University of Technology, Bratislava, in 1995 and 2002. Since 1997, he has worked as a senior lecturer at the Department of Telecommunications, FEI STU, Bratislava. At present, his research activity is focused on the high-speed signal transport through wirelined access networks by means of xDSL and PLC technologies using various techniques of digital signal processing.