Coupling of a Powerline Communication Modem to an Industrial Fieldbus Network

Jose Puche Planells

Supervisor:
Dipl. –Ing. Joao Daniel Lopes

Universität Stuttgart
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen
Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

June 2014

Acknowledgment

I would like to give special thanks to the entire institute for supporting this project during this time and especially to my supervisor Joao Daniel Lopes who always trusted me and from who I have learnt a lot. I would want also to thank to my mother and my closer family, which always are supporting me and are always aware of me. To my girlfriend Patri, who always has been there supporting me and cheering me up in the bad moments. Finally I want to thank to all my friends. Especially all these friends that I have made during the last 18 months in Stuttgart and with I have spent such incredible moments.

Table of Contents

Table of Figures

1. Motivation

In a broad and heterogeneous industrial communications environment, the need of having flexibility as well as maintaining the reliability and low cost has pushed the researchers to look for new possibilities and market opportunities to cover all the needs. Actual industrial systems are needed of at least two ways of inputs: power input, for the system operation, and data input for controlling, testing and administrating the system. The main restrictions, that actual systems and networks have, are low flexibility and, sometimes, considerable expenses in terms on maintenance. But also they have advantages, e.g. low response time, high reliability, wide and highly known architecture etc. In this background, grows the thought of increasing the flexibility without impacting such important aspects as response time or reliability.

On the industrial communication protocols, Ethernet POWERLINK standard is one of the existing protocols in the market. Ethernet POWERLINK is a real-time industrial communication protocol based on Ethernet standard, which is used for controlling and commanding several sensors and actuators with high speed, time-synchronization and reliability, minimizing the global system latency.

The main aim of this project is to reach a balanced solution by joining both power and data inputs in just one cable that allows an increase in the flexibility, as well as decrease in the maintenance costs, restraining the response time and achieving compatibility with a wide spread standard as is Ethernet.

2. Industrial Communication Levels and Requirements

In the industrial environment, there are different communication levels to cover all the needs at different hierarchy levels.

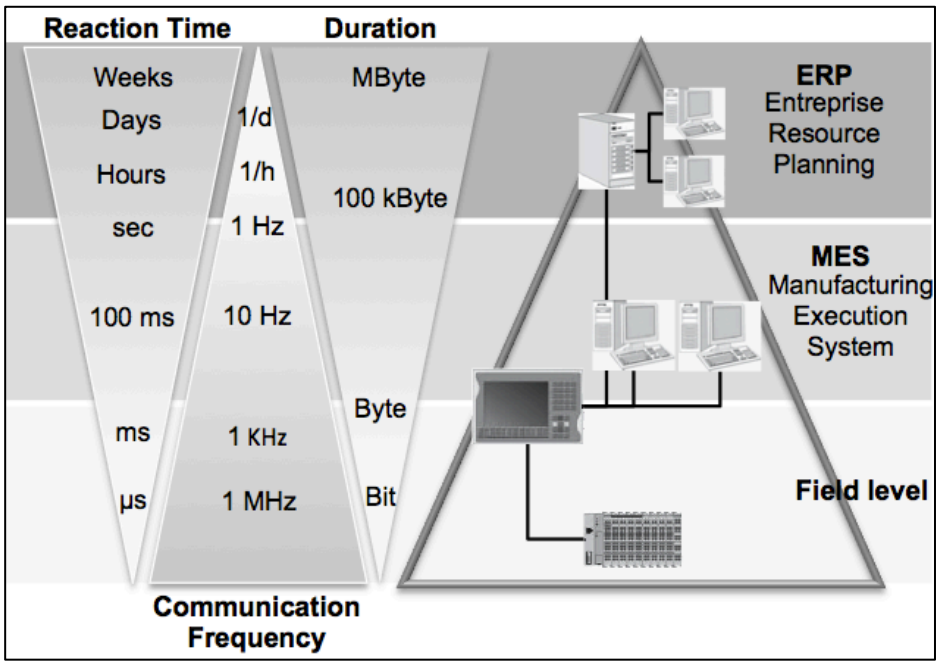

Figure 1: Industrial Communication levels

On top level, the Enterprise Resource Planning (ERP) concerns with the production planning, material management and quality management, having high reaction times and requiring high loads of data to be managed.

The next level is the Manufacturing Execution System (MES), which deals with the control and coordination of the industrial functions.

The low-end level is the Field Level, where industrial fieldbus communication takes place. Here is included the connection between sensors, actuators and Programmable Logic Controllers (PLC), which exchange lower amount of information but presenting a lower reaction time and thus, higher work frequencies.

The levels ate characterized with varying reaction time, jitter and data rates. From the general perspective, the reaction time, jitter and the size of the data decreases as the levels are descended. In the lowest level -the field level- the unwanted jitter leads to a varying cycle time. This is crucial in determining the real time capability parameter of the communication system.

The main parameters for industrial field level are defined following the next requirements:

- Data Rate > 10Mbit/s
- Lower Cycle Time: 10 ms
- Jitter: 10 μs

3. State of the art

3.1. Commercial Powerline Communication

There are commercially extended Powerline Communication systems, which are applicable to domestic uses (LAN topologies for non-critical data transmission) and achieve enough data transmission rates for common tasks (Fast-Ethernet). The most widely deployed Powerline networking standard is from the HomePlug Powerline Alliance, whose specifications were adopted by the IEEE 1901 group as baseline for their standard [1].

Figure 2: Commercial Powerline Systems

3.1.1. Main issues with Powerline

1. The power line is an analog channel, which means that the information must be modulated in order to achieve high data bit rates.

2. The power line is a noisy channel with high level of external disturbances coming from all the devices surrounding it. It is not prepared to protect the signal away from the electromagnetic interferences. Hence, the data cannot be transmitted easily with an acceptable Bit Error Rate. This problem is even more pronounced when the data is modulated.

These two issues force to protect the transmission channel by introducing data codification, interleaving and other error detection and correction techniques.

3.1.2. Commercial Ethernet Powerline in the industrial communication

As described before, in the field level communication, de reaction times are between the millisecond and the microsecond levels, keeping the jitter lower than 10 μs. Moreover, the needed data rate must be at least 10Mbit/s.

Although the commercial Powerline Communication systems achieve higher data transfer rates (higher than Fast-Ethernet in some cases), the achieved latency, jitter and cycle times are not enough small to be applicable for most industrial communications, where the real-time data transmission is essential.

3.1.3. Benefits of Powerline Communication?

The main benefit is the flexibility. With Powerline Communication the communication network can be established between two far away points (e.g. in large assembly lines) without need of extra cables, only the power supply cables. The robustness of the power line cables is also higher than conventional communication cables, which, on industrial environment, is a factor to take into account.

3.2. Developed Industrial Powerline Communication

The basis of this project was the implementation of a Powerline Communication system to command a drive through a dSpace Controller board, the DS1103 and two developed Powerline modems.

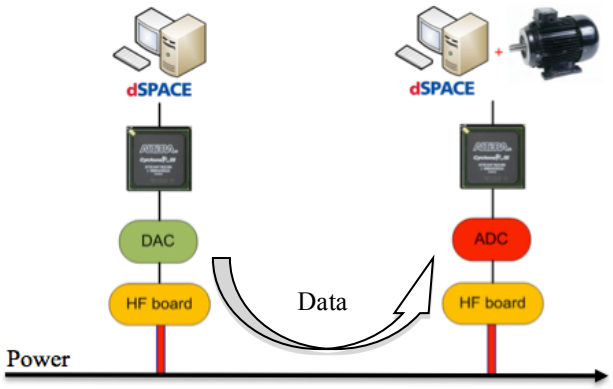

Figure 3: dSpace Communication over Industrial PLC

These Powerline modems were based on a DBF3C120 board with a FPGA Altera Cyclone III integrated, a transceiver board with analog-to-digital and digital-to-analog (DAC and ADC) converters and a high frequency board (HF board). This communication was unidirectional, i.e. the communication was only to command the drive but not to read any information from it. This implementation presents a limitation; it can only be used with a dSpace system, which is not an industrial communication standard [2].

4. Design concept

In this chapter the concept of the existing Industrial Powerline Communications modem as well as the target system will be explained. Here are included, the hardware, the software, the Ethernet POWERLINK protocol and the B&R System descriptions.

4.1. FPGA-based OFDM Modem

Transmitting digital data through a conventional DC cable, where different signals at different frequencies and voltages are being also transmitted, is more difficult than transmitting through an Ethernet cable. The data must be processed in order to avoid mixing it with unwanted signals.

To process the digital information on an analog carrier signal, an Orthogonal Frequency Division Multiplexing (OFDM) with modulation in amplitude modem is designed. The result signal can be further transmitted easily through different communication channels, e.g. cable, air, fiber, or in the project case, a DC cable.

4.1.1. Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing is a technique for multiplexing several signals into one channel, which means, sending several signals at the same time each one in a different frequency. Its main principle is grounded in the Frequency Division Multiplexing techniques (FDM) but with the objective of the efficient use of the available spectrum for transmitting multiple subcarrier frequencies.

With FDM several signals (or subcarriers) with a determined bandwidth and in a determined central frequency are transmitted. Both bandwidth and central frequency are chosen specifically for the transmission channel. FDM techniques are designed to divide the available spectrum in smaller frequency signals instead of transmitting a unique wide bandwidth signal. With this technique it is possible to adapt the signal to the transmission channel, compared with a single carrier method, by equalizing the different signals to the channel response. That means transmitting each subcarrier with slightly different characteristics to adapt it to the frequency that is going to be transmitted through. To this purpose, it is needed a previous channel estimation.

The OFDM is a concrete case of FDM, with better usage of the available bandwidth. This better usage will allow more subcarriers in the same bandwidth to be transmitted, and thus, higher data rate. The main difference with the FDM is that the frequencies that are used for multiplexing the different signals are closer, achieving a higher spectral efficiency of the available bandwidth. This is achieved thanks to the orthogonality of the transmitted subcarriers. As a result the crosstalk between subcarriers is theoretically eliminated and the inter-carrier guard band is in not required.

The orthogonality requires a minimum spacing between two consecutive subcarriers of

$$\Delta f = \frac{1}{T}\ Hz \tag{1}$$

Where T is the symbol duration. Therefore, with N subcarriers, the total bandwidth will be

$$BW = N \times \Delta f \tag{2}$$

In the case of FDM technique, is needed a spacing of

$$\Delta f = \frac{2}{T}\ Hz \tag{3}$$

Therefore the subcarriers are not orthogonal to each other and a guard band is needed to avoid channel crosstalk. The total bandwidth needed to transmit the same amount of subcarriers is then twice as with the OFDM. The total bandwidth for the project is set to 10 MHz and the number of OFDM subcarriers is set to 32. This means that each subcarrier must have a bandwidth of

$$\Delta f = \frac{B}{N} = \frac{10\ MHz}{32} = 312.5\ KHz \tag{4}$$

$$T_{symbol} = \frac{1}{\Delta f} = 3.2\ \mu s \tag{5}$$

In band base, the total bandwidth of 10MHz signal is generated in two 5 MHz channels. These are In-Phase channel and In-Quadrature channel. After generating these two signals, a signal modulator modulates both signals in Quadrature at the defined carrier frequency of 100 MHz.

In the next pictures both FDM and OFDM in band base are depicted. It is clear the difference in terms of subcarriers for a given bandwidth.

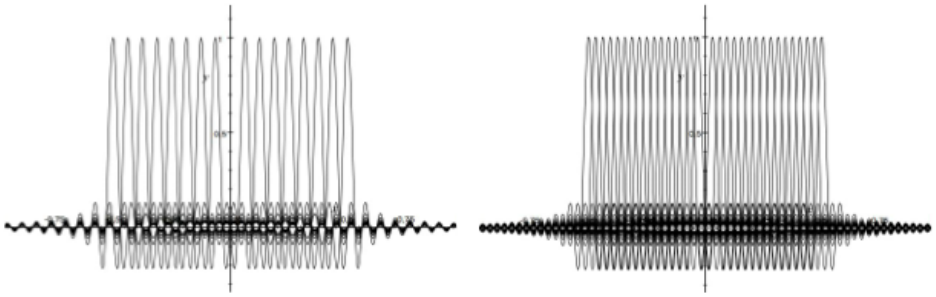

Figure 4: FDM vs. OFDM

In left picture, a FDM multiplexing scheme is used, and for a given bandwidth of 10 MHz (from -5 to +5 MHz), 16 subcarriers are transmitted. In the right picture, an OFDM multiplexing scheme is used and for the same bandwidth, 32 subcarriers are transmitted because the separation between them is reduced to the half.

In terms of digital signal codification, first the Binary Phase Shift Keying (BPSK) modulation was implemented. To achieve more efficient data transmission, and thus a higher data rate, a 16-Quadrature Amplitude Modulation (16-QAM) was finally implemented.

The following schematics offer an overview of the system:

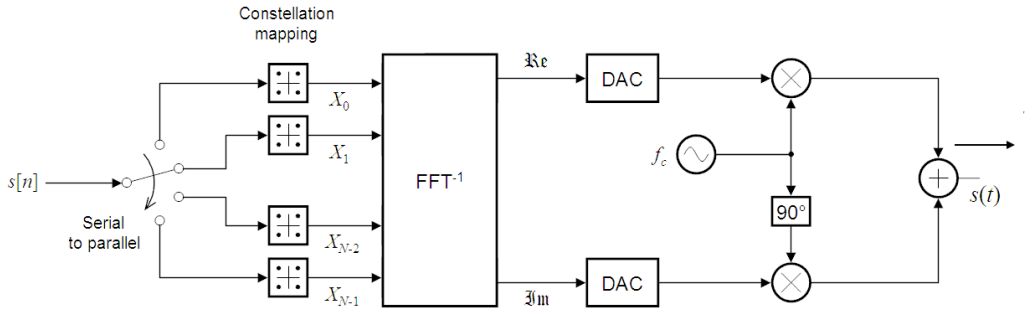

Figure 5: OFDM Transmitter

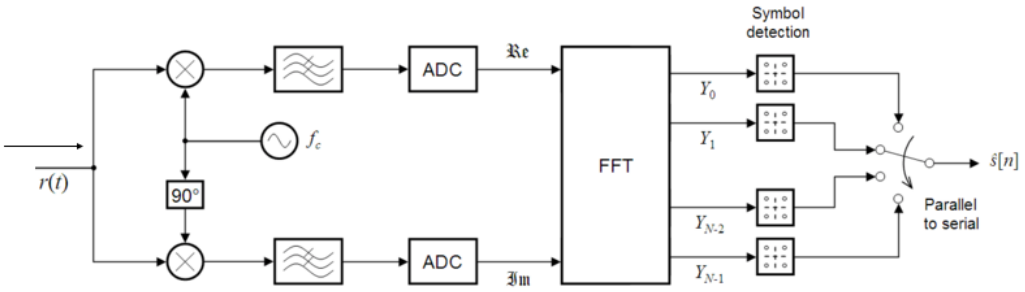

Figure 6: OFDM Receiver

Where the constellation mapping generates the BPSK or 16QAM symbols, the FFT^{-1} and FFT blocks indicate the Inverse Fast Fourier Transform and Fast Fourier Transform, essential blocks for the OFDM, f_c indicates the carrier center frequency of 100MHz, and $s(t)$ and $r\ (t)$ represent sent signal and received signal respectively [3].

4.1.2. FPGA-based Powerline Modem

To implement the described characteristics, a device that has the capabilities of digital signal processing is needed. For this purpose a Field Programmable Gate Arrays (FPGA) system was chosen.

An FPGA is an integrated circuit built with logic blocks that can be programmed by a designer for a desired functionality. This kind of devices is called Programmable Logic Devices (PLD), and unlike the Relay Logic Devices, which once that is implemented cannot be changed, PLDs allow that changes can be easily made. These devices are perfect for implementation of new features and make the tasks of developing, debugging and updating easier. Furthermore, the provided development boards included several ports which are needed for the implementation, e.g. Ethernet port, HSMC port as well as switches, LEDs,

LCD display etc. Everything together makes the FPGA development board the perfect choice for the purpose [4].

For the implementation of the hardware, a Hardware Description Language (HDL), and a development environment must be used. In the project the VHDL language and the Altera Quartus II environment are used. Altera offers a complete solution to design, program and optimize the desired system.

In the design process, an Altera's offered solution plays an important role, the Altera MegaCore IP Library. The IP (Intellectual Property) Cores are blocks previously designed to implement complex functions in an easier way, e.g. Cyclic Redundancy Check (generator and checker), Triple-Speed Ethernet with MAC support, Viterbi decoder etc. This device and its capabilities as well as the used software will be described during the two sections of this chapter, the hardware design and software design sections.

4.2. Hardware Design

The hardware is one of the two essential parts of the system. A proper implementation of the specification into hardware as well as the maximum optimization thereof, will play an important role in the global performance of the system, i.e. minimum Bit Error Rate, maximum Signal to Noise Ratio etc.

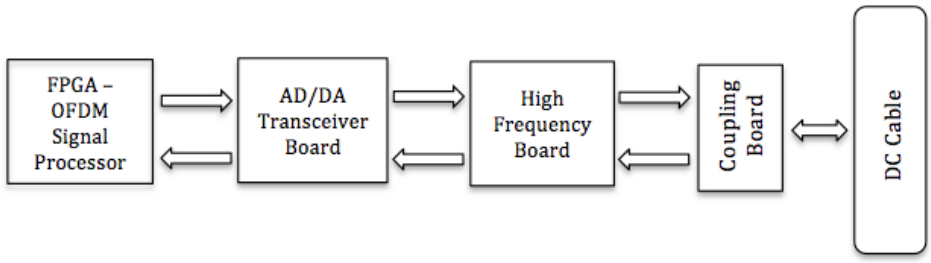

Figure 7: Transceiver Stages

On the hardware side the following parts are described:

- FPGA – OFDM Signal Processor.
- AD/DA Transceiver Board.
- High Frequency Modulation Board.
- Coupling/Decoupling Circuitry.

All these parts together make the whole Powerline system.

4.2.1. FPGA

The FPGA is the core of the OFDM modem. Its task is to generate the proper OFDM digital signals and send it further to the Digital-to-Analog converter, as well as receive the digital signals from the Analog-to-Digital converter and processing it.
The development board used is the Terasic DE2-115 with the Altera Cyclone IV chip. This board provides different solutions for multiple needs.

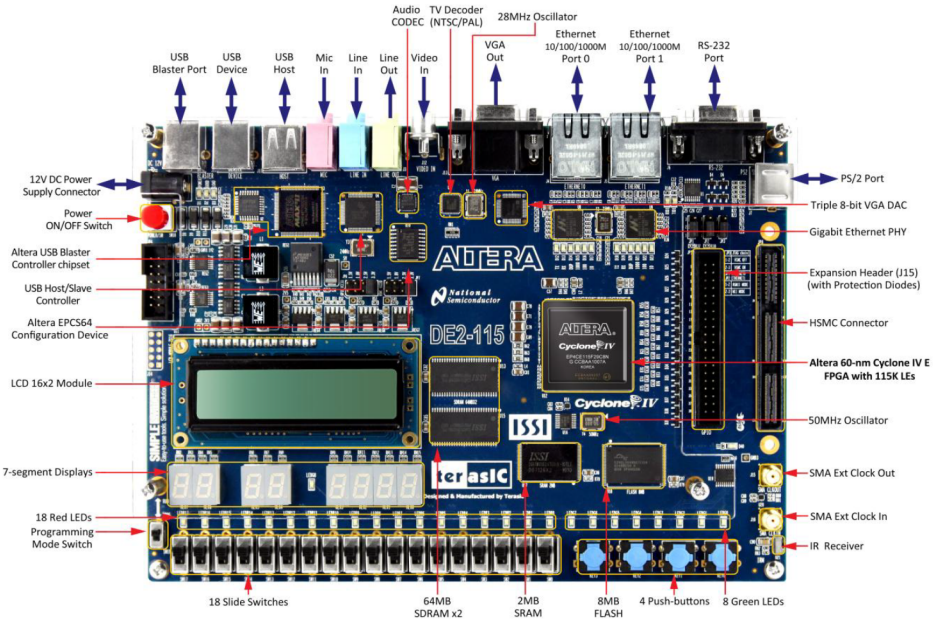

Figure 8: Terasic DE2-115 Development Board

Some of the features that are included are here illustrated:

- **Altera Cyclone IV 4CE115 FPGA device**
- **Altera Serial Configuration device**
- **USB Blaster (on board)**
- **2MB SRAM**
- **Two 64MB SDRAM**
- **8MB Flash memory**
- **SD Card socket**
- **4 Push-buttons**
- **18 Slide switches**
- **18 Red user LED and 9 Green user LEDs**
- **50 MHz external oscillator**
- **24-bit CD-quality audio CODEC**
- **VGA DAC**
- **2 Gigabit Ethernet PHY with RJ45 connectors**
- **USB Host/Slave**
- **RS-232 transceiver and 9-pin connector**
- **PS/2 mouse/keyboard connector**
- **IR Receiver**
- **2 SMA connectors for external clock input/output**
- **One 40-pin Expansion Header with diode protection (GPIO)**
- **One High Speed Mezzanine Card (HSMC) connector**
- **Eight 7 segments displays**
- **16x2 LCD module**
- **TV Decoder**

Figure 9: DE2-115 Features

For the project, the most important characteristics of this development board are the Cyclone IV 4CE115 FPGA device, the Gigabit Ethernet PHY Port, the High Speed Mezzanine Card (HSMC) connector, the switches and buttons and the LCD Display [5].

Altera Cyclone IV 4CE115 FPGA device

The core of the board is the Altera Cyclone IV EP4CE115 C7. This device contains the programmable logic blocks inside. It has been chosen to meet the required specifications for the purpose [6].

EP4CE115	
Logic elements (LEs)	114,48
Embedded memory (Kbits)	3,888
Embedded 18 × 18 multipliers	266
General-purpose PLLs	4
User I/O Banks	8
Maximum user I/O	528

Figure 10: Altera Cyclone IV Features

Gigabit Ethernet PHY Port

The Terasic DE2_115 board provides Ethernet support via two Marvell 88E1111 Ethernet PHY controllers. These controllers are connected each one to a different Ethernet port installed on the board. Both controllers include 10/100/1000 Mbps Gigabit Ethernet transceiver and support GMII/MII/RGMII/TBI MAC interfaces. Nevertheless for the given board only MII and RGMII modes are supported. The selection of the operation mode is easily accessible through two jumpers provided on the board, JP1 and JP2, which allow the selection the mode for both ports independently.

In the next figure, the connections between the FPGA and one Ethernet port, through the Marvell controller, are depicted [5].

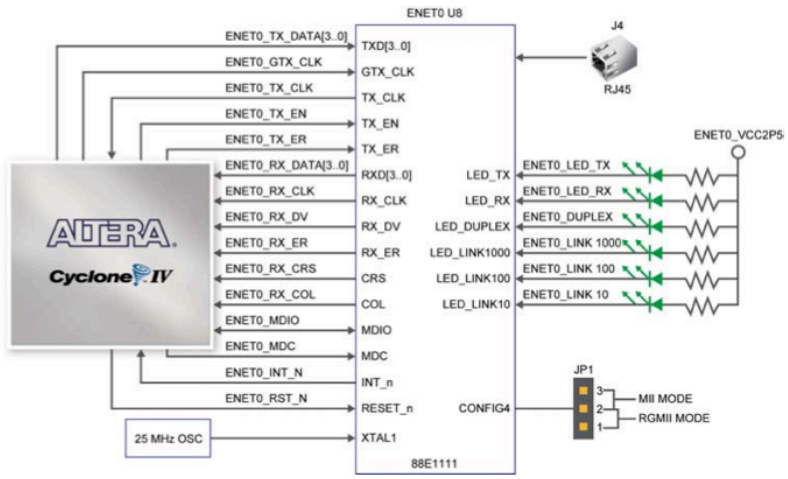

Figure 11: Marvell 88E1111 Ethernet Controller

High Speed Mezzanine Card (HSMC) connector

The HSMC interface provides the Terasic DE2-115 board a mechanism for extending the peripheral-set by means of add-on cards. This can couple high speed signalling requirement as well as with low-speed device interface support. The HSMC connector connects directly to the Cyclone IV FPGA with 172 pins. The voltage level of the I/O pins on the HSMC

connector can be adjusted to 3.3V, 2.5V, 1.8V or 1.5V for matching the required digital voltages level using a provided jumper on the FPGA. Since the designed boards work with 3.3V logical levels, this option must be selected.

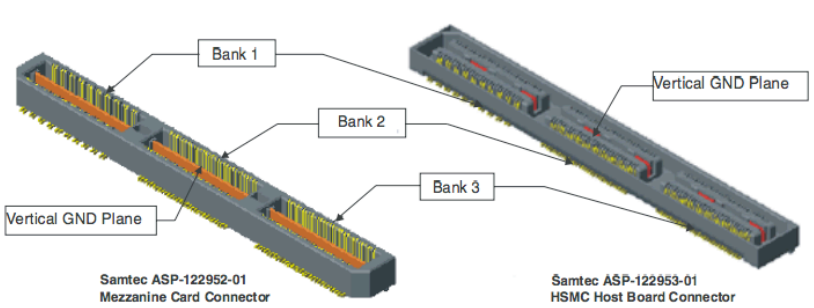

Figure 12: HSMC

The connector provides 3 independent signal banks. Each bank provides clock signals in both directions. Bank 1 has a dedicated JTAG and a system management bus (SMBus) and 8 channels CDR in both directions. Banks 2 and 3 have 12V and 3.3V power supply pins and provide 60 differential channels or 120 single-ended I/O pins.

For the project case, this connector is essential for the connection between the FPGA board and the transceiver board. This connection is possible thanks to a developed adapter board, which provides the connections between the HSMC port and the provided DBF ports on the transceiver boards [7].

4.2.2. Analog-to-Digital and Digital-to-Analog Transceiver Boards

Each FPGA board has to be able to send and receive digital signals. After generating the OFDM signals, in order to be modulated and transmitted to the channel, the signals have to be converted first into analog signals. Also, the received demodulated signals have to be converted to digital signals in order to be interpreted by the FPGA board.

To achieve these goals, a transceiver board was developed. This board provides two independent channels of communication, one in each direction.

The main features of the designed board are:

- Digital-to-Analog converter for transmitter side, symmetrical analogic output AD9765, up to 40 MSPS.
- Analog-to-Digital converter for receiver side, symmetrical analogic input AD9248, up to 40 MSPS.
- Connectivity with dSpace system.
- Testing jumpers and measure points.

The layout of the board is showed and described in the next figure:

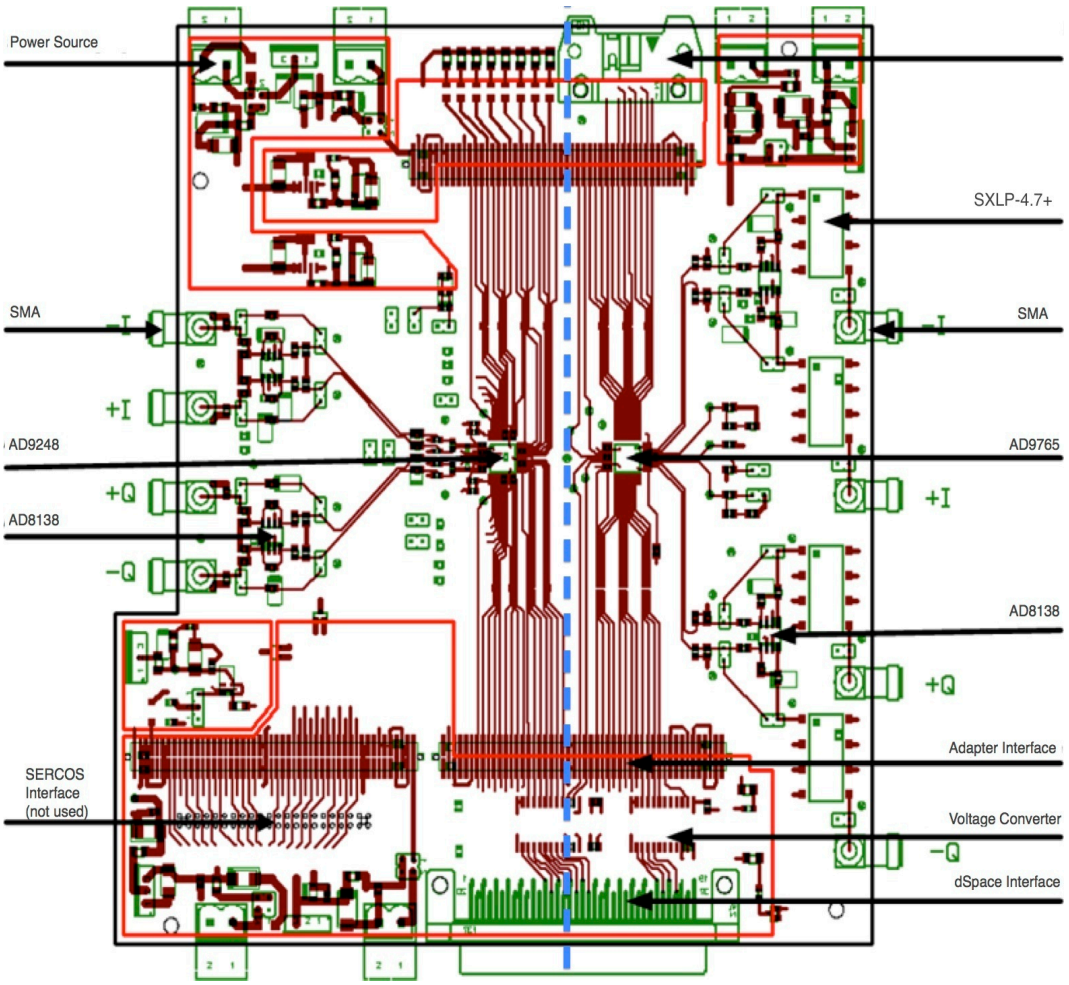

Figure 13: Transceiver Board

In the layout, the transmitter and receiver parts are distinguished and delimited by the vertical blue line, transmitter on the right side, receiver on the left side. The board needs positive and negative external power sources between 6.5 and 30V for positive and between -6.5 and -30V for negative (the needed voltage supply for the components is 3.3V and 5V and the regulators' datasheet specify a max dropout voltage of 1.5V and a absolute maximum of 30V). Connections to the FPGA board are possible thanks to the Adapter Interface and the adapter board mentioned before. The pins of the Analog-to-Digital converter and Digital-to-Analog converter are connected to these two interfaces. Connections to the High Frequency boards are possible thanks to the 8 SMA connectors, 4 on the transmitter side and 4 on the receiver side [8].

SMA Connectors

Analog connection for transmitting and receiving signals to/from the High Frequency Boards. These connectors offer 50Ω characteristic impedance and a maximum frequency of 18 GHz for the connection of a coaxial cable.

Analog Devices AD9765

12-Bit 125 MSPS Dual TxDAC+ Digital-to-Analog Converter with 50Ω characteristic impedance. This dual Digital-to-Analog converter is chosen due to the need of transmitting In-Phase and In-Quadrature signals of the OFDM. 3.3V logic voltage is supported. The main applications of this component are in Communications, Quadrature Modulation, Digital Synthesis and others [9].

Analog Devices AD9248

14-Bit Dual Analog-to-Digital Converter with 50Ω characteristic impedance. Available in 20,40 and 65 MSPS. Currently used the 40 MSPS version. This dual Analog-to-Digital converter receives the In-Phase and In-Quadrature single-ended analog signals, provided by the AD8138, and converts it into digital signals [10].

Analog Devices AD8138

Low distortion differential ADC Driver with 50Ω characteristic impedance.
The modulator needs a differential signal input centered on a 500mV DC, thus a driver and proper resistors are needed. Therefore, on the transmitter side, transmitted signal must be converted from single-ended to differential in order to be modulated. On the receiver side, received signal must be converted from differential to single-ended in order to do the sampling [11].

Mini-Circuits SXLP-4.7+

The SXLP-4.7+ is a surface mount Low-Pass Filter with stop frequency of 4.7 MHz with 50Ω characteristic impedance [12].

Adapter Interface

For the transmission of the digital signals from the FPGA to the Digital-to-Analog converters and from the Analog-to-Digital converters to the FPGA an interface is needed. For this purpose a DBF connector is used, due to the former system, which used a DBF3C120 board.

DSpace Interface and Voltage Converter

This interface provides communication between the FPGA board and the dSpace system.

4.2.3. High Frequency Modulation Boards

After generating the base band In-Phase and In-Quadrature signals, the signals have to be modulated at the desired working frequency. To the project case, this frequency is 100 MHz. A modulation is the process of changing the properties of a periodic signal, which is called carrier signal, with a modulating signal, which contains the information to be transmitted. The main aims of modulating the signal to upper frequencies are adapting the signal to the channel

characteristics, avoiding possible disturbances at low frequencies, and having more available bandwidth, which is equal to having more data rate.
The main features of this board are the following:

- 4 Differential inputs, 2 for In-Phase signal and 2 for In-Quadrature signal (+I, -I, +Q, -Q) with SMA connectors for modulating signal.
- 4 Differential outputs, 2 for In-Phase signal and 2 for In-Quadrature signal (+I, -I, +Q, -Q) with SMA connectors for demodulated signal.
- 100 MHz carrier central frequency.
- Quadrature modulator ADL5385 and demodulator ADL5387.
- Testing jumpers and measure points.
- Input for an external oscillator (optional)

The layout of the board is shown and described in the next figure:

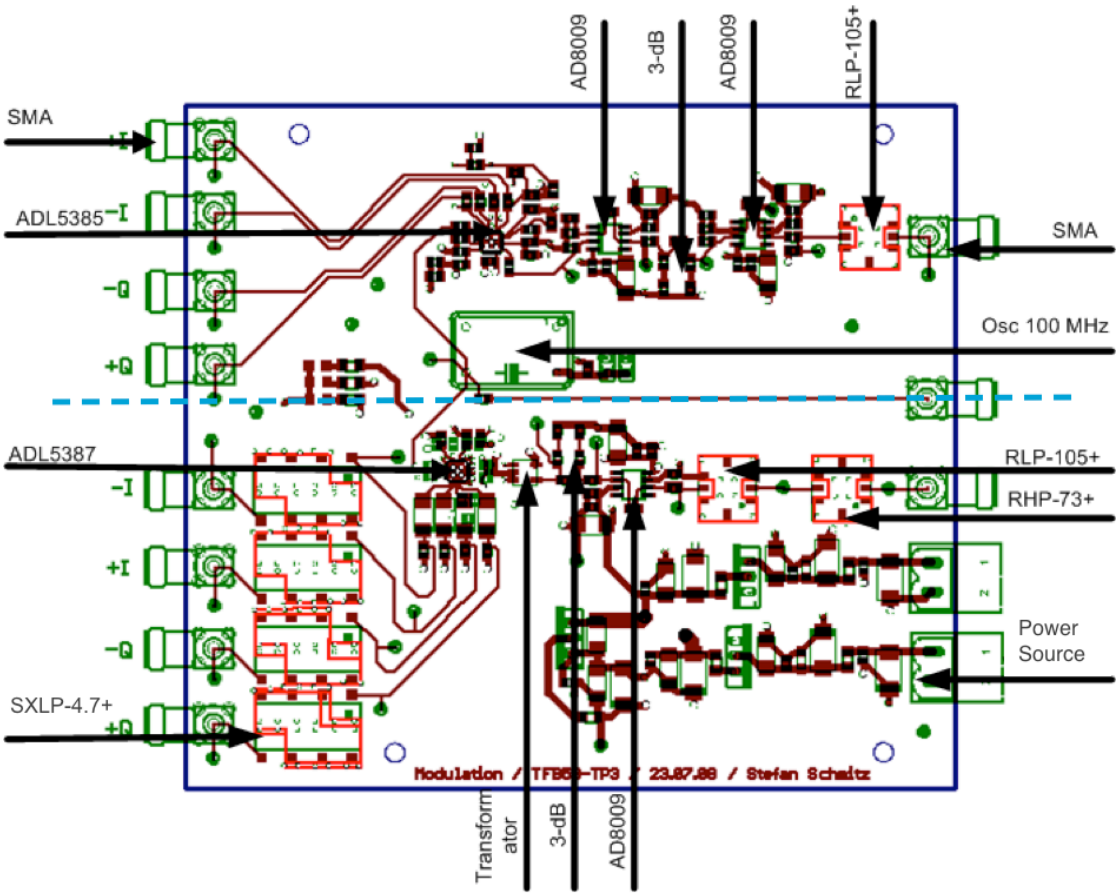

Figure 14: High Frequency Board

In this case, the transmitter and the receiver side are clearly defined and split by the blue striped line. It is important to notice, that both directions (transmitter and receiver) share the same transmission channel and are modulated at the same frequency, thus the communication is only possible in one direction, i.e. half duplex. The transmitter lays on the upper part and the receiver on the lower part. The board needs, as for the transceiver board, positive and negative power supplies between 6.5V and 30V for positive and between -6.5V and -30V on Power Source input [8].

Analog Devices ADL5385

The ADL5385 is a 30 MHz to 2200 MHz Quadrature Modulator with 50Ω characteristic impedance. The modulator requires differential signals in quadrature centred on a 500mV dc bias. The input carrier frequency (Local Oscillator) must be two times the desired carrier frequency i.e. 200 MHz This modulator is normally used for communication applications like Radio-link infrastructure, Cable modem termination system, UHF/VHF radio, Wireless infrastructure systems etc. hence is adequate for the purpose [13].

Analog Devices ADL5387

The ADL5387 is a 30 MHz to 2 GHz Quadrature Demodulator with 50Ω input impedance. The demodulator needs a differential RF input and offers a baseband I/Q driving a $2V_{p-p}$ signal into 200Ω. The voltage conversion gain of the demodulator is >4dB. The input carrier frequency (Local Oscillator) must be two times the desired carrier frequency i.e. 200 MHz The main applications of this demodulator are QAM/QPSK RF/IF demodulators, W_CDMA/CDMA/CDMA200/GSM, Broadband wireless and WiMAX etc [14].

Mini-Circuits RHP-73+

The RHP-73+ is a surface mount High-Pass filter with start frequency at 73 MHz with 50Ω characteristic impedance [15].

Mini-Circuits RLP-105+
The RLP-105+ is a surface mount Low-Pass filter with stop frequency at 105 MHz with 50Ω characteristic impedance.

In receiver side, RLP-105+ in serial with RHP-73+, a band pass filter with lower cut-off frequency of 73 MHz and upper cut-off frequency of 105 MHz is built [16].

Analog Devices AD8009

1GHz Low Distortion Amplifier. Presented in different configurations to achieve desired gains for each stage. The main applications of this AD8009 are in IF/RF Gain Stage [17].

3dB Attenuator

3dB attenuator made with a resistor network in PI design, designed to adequate the input signal to the next stage.

1:1 BALUN

Converts the signal from single-ended to differential to adequate it to the input of the demodulator. It is 1:1 so it will not affect to the impedance that remains 50Ω.

Crystek CC0-085

The CC0-85 is a true sine wave clock oscillator to generate the carrier signal in a 200 MHz frequency [18].

Board schematic:

A simplified scheme of the configuration is presented in the next picture:

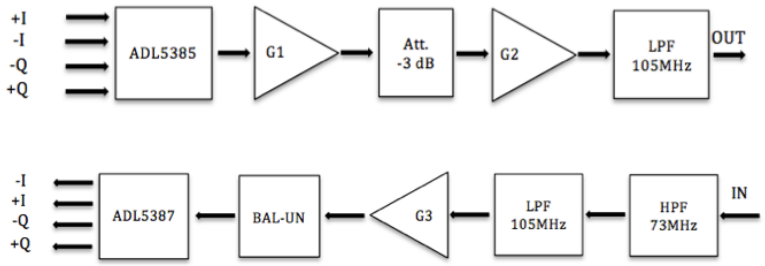

Figure 15: Transceiver Stages

To calculate gain values G1, G2 and G3, it is important to know the characteristics of the channel, losses in passive elements, conversion gain of the demodulator and minimum amplitude needed for demodulation in the reception part of the High Frequency Board. Also is important to specify the configuration of the amplifier used. In the case of stages G1 and G2, non-inverter amplifiers are configured. But in G3 stage an inverter amplifier is used. It is important to notice, that the equations that govern the behaviour and the gain of each configuration are different. The difference is relevant to obtain the gain. Just changing the relation between two resistors can easily change the gain of the stages as follows:

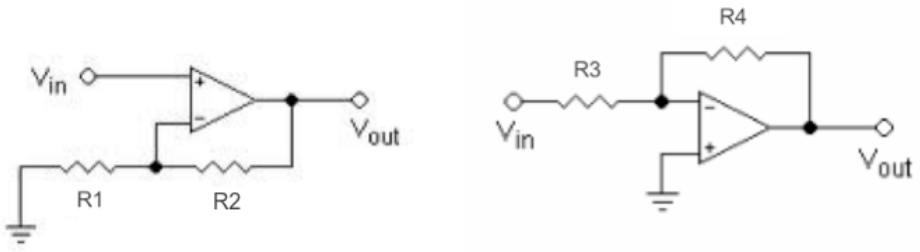

Figure 16: Non-Inverter and Inverter Configurations

Thus, for non-inverter configuration:

$$V_{out} = V_{in}(1 + \frac{R_2}{R_1}) \quad (6)$$

For inverter configuration:

$$V_{out} = -V_{in}(\frac{R_2}{R_1}) \quad (7)$$

The voltage gain is given by the equations:

$$G = \frac{V_{out}}{V_{in}} \tag{8}$$

$$G(dB) = 20\log\left(\frac{V_{out}}{V_{in}}\right) \tag{9}$$

4.2.4. Coupling/Decoupling Circuitry

The main characteristic of a Powerline Communication system is transmitting information through a Powerline cable. That means, in industrial fields, that the cable where the signal is coupled carries also a voltage difference of about 700V DC. To allow the transmitted signal to travel through the DC channel and protect the High Frequency Modulation Board from the DC signal, a Coupling/Decoupling circuitry is needed. Hence the main characteristics of this circuitry are the following:

- Main protection against high voltage DC input signal.
- Protection against high frequency and high power peaks.
- Coupling and decoupling the transmitted and received signal into/from the same channel.
- 75Ω to 50Ω Impedance conversion.
- Converting the transmitted single-ended signal to a differential signal and the differential received signal to single-ended signal.

The layout of the circuitry is showed in the next figure:

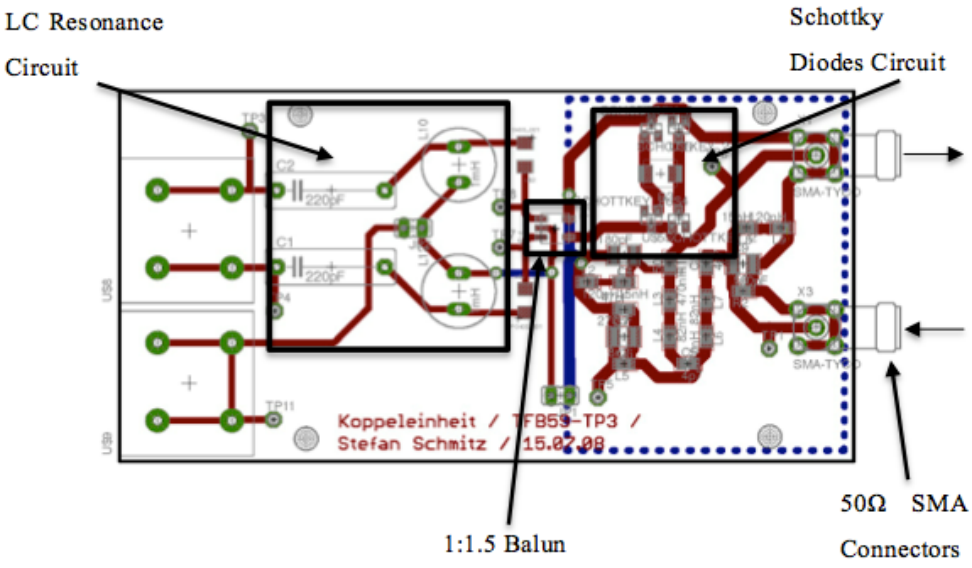

Figure 17: Coupling Board Design

In the design several stages are defined:

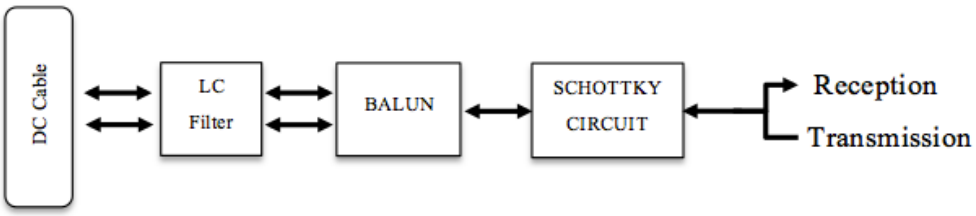

Figure 18: Coupling Stages

LC Resonance Circuit

To block the unwanted frequencies (mainly high-voltage DC) signals, a resonance circuit designed to be resonant at 100 MHz is designed. At this frequency the inductive reactance is equal to the capacitive reactance, thus the impedance is at its minimum. That means, the transmitted signal at 100 MHz pass through the filter with almost no attenuation, while other frequencies are blocked. Because the signal is differential, a LC filter circuit must be implemented on each line.

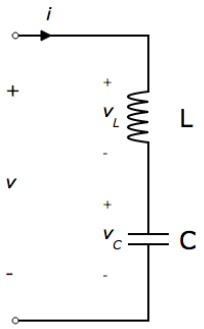

Figure 19: LC Resonator

The values of L and C were calculated following the formula:

$$f_{res} = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}} = 100\ MHz \tag{10}$$

Taking a L value of 10nH, then:

$$C = \frac{1}{4 \cdot \pi^2 \cdot 10nH \cdot (100MHz)^2} = 253.3\ pF \tag{11}$$

Thus a capacitor of 220 pF is chosen. Also important is the maximum voltage that the capacitor can support. In this case, a WIMA FKP1 capacitor, which supports up to 2000VDC [19].

1:1.5 BALUN

After the filter, a 1:1.5 balun is used. This component will convert the received differential signal into a single-ended signal. Also, a relation 1:1.5 is chosen in order to convert the input 50Ω impedance of the connectors into 75Ω impedance to maximize the transmitted power.

Schottky Diodes Bridge Circuit

To protect the High Frequency Board against possible high frequency and voltage peaks that could be produced, a Schottky diode bridge circuit is implemented.

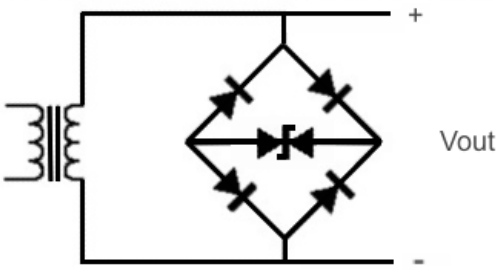

Figure 20: Schottky Diodes Circuit

Usually, the Schottky diode bridge is used for rectifying AC into DC signals, but in this case a SMBJ Transient Voltage Suppressor is placed in the middle of the diode bridge.

4.3. Software Design

The second main part of the system is the software part. Since the core of the system is a Programmable Logic Device, this can be programmed and optimized easily to achieve the desired behaviour.
To test the proper functioning of the system, as well as to do quality measurements of the channel, initial software that transmits a known sequence was developed. This software will be the basis of the further implementation of the Ethernet coupling as will be described in the Ethernet Coupling Principle section of this chapter.

4.3.1. FPGA-Based Transceiver Program Structure

Initially, the existing programs implemented transmitter and receiver in separated and independent boards. This implementation was needed for concept and validation. Nevertheless, for a transceiver program, both receiver and transmitter have to be located in the same board. To achieve this goal, a merging of the two working programs of transmitter and receiver parts was performed.

After merging both projects, a single project with two blocks at the top level of the hierarchy results. These two blocks are transmitter and receiver blocks, containing several stages that modify the data to be transmitted, in the case of the transmitter program, and properly recognized, in the case of the receiver program.

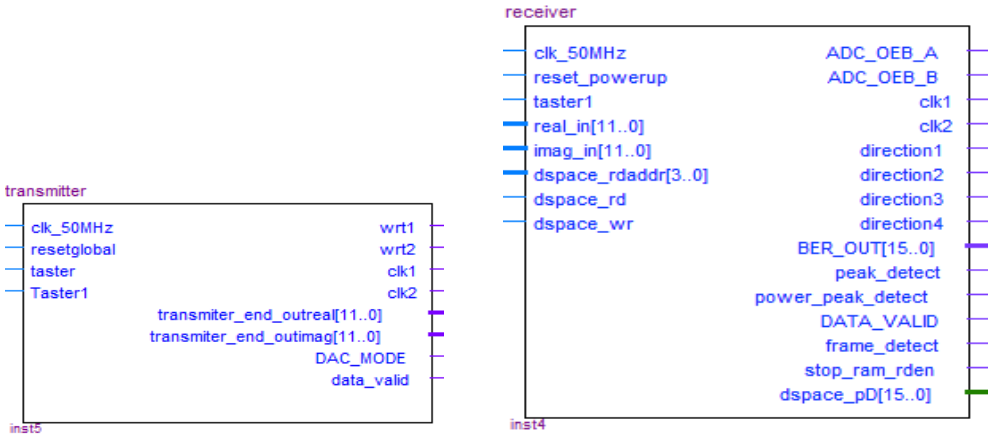

Figure 21: Transmitter and Receiver Blocks

The blocks showed before work with the 50 MHz signal clock provided by the FPGA circuitry. Furthermore, taster (to power up and to start the transmission) and reset signals are assigned to switches and buttons placed in the Terasic development board.

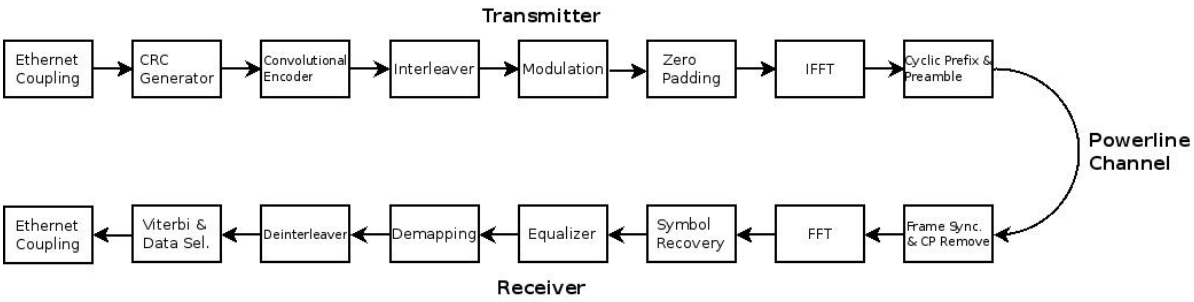

Figure 22: Transceiver Software Stages

During this section, the different blocks of the system will be explained through the ground theory of the transmission. There are two signal treatments along the system.
First, a treatment of the raw data, including data packaging, error prevention and modulation, where data still has a digital meaning.
Secondly, after the modulation block, modifications of the signal with the aim of constructing the analog subcarriers and the OFDM symbols to be transmitted.

4.3.1.1. Theoretical Introduction

The initial communication protocol of this Powerline Communication system was developed for dSPACE. DSPACE is a company that offers different hardware and software solutions to accelerate the development of complex control systems with ease in a vast amount of industries.

In the former system design, a dSPACE DS1103 Controller board was available in one computer. This Controller board was used to command a drive with a second dSPACE system. Both dSpace systems were connected to each FPGA through the dSPACE Interface of the Transceiver board. The process consisted in writing the desired information in real-time to

the FPGA to command the drive that is placed at the other side of the Powerline channel. The dSPACE also had the function of controlling the quality of the reception. Hence the communication between the dSPACE system and the FPGA was bidirectional. This system had a working frequency of 32 kHz, and the OFDM system should be designed according to this working frequency. That means that the cycle time of the communications should be according,

$$T_{cycle} = \frac{1}{f} = \frac{1}{32\ kHz} = 31.25\ \mu s \tag{12}$$

This cycle time value will define the number of data OFDM symbols and also the maximum data rate for a given modulation. In this case, as described at the OFDM description section in equation 5, the symbol duration is 3.2μs.

For the proper symbol detection and to avoid Intersymbol Interference (ISI), a guard time between symbols is inserted. This guard time is established as

$$T_{guard} = \frac{1}{4} \times T_{symbol} = 0.8\ \mu s. \tag{13}$$

Then the total OFDM symbol duration is

$$T_{ofdm} = T_{symbol} + T_{guard} = 4\ \mu s. \tag{14}$$

Then, the maximum OFDM symbols are

$$N_{symbols} = \frac{31.25\ \mu s}{4\ \mu s} = 7.8125 \rightarrow 7\ OFDM\ symbols. \tag{15}$$

Apart from data symbols, training symbols for the channel estimation and equalization are needed. These symbols have the same duration as the data symbols. In this case, 5 symbols are transmitted. Data symbols will transmit data, training symbols have two purposes, one symbol is for frame synchronization, and the other is for the channel estimation. Data symbols are generated by gathering and packing codified streams of transmitted bits. The need of transmitting training symbols has as side effect a reduced net data rate [2].

To codify the transmitted bits into analog signals in order to increase the data rate, a digital modulation scheme is used. A digital modulation allows the transmission of several bits into one analog signal by modifying an analog carrier signal. Several types of digital modulations are available:

- Phase Shift Keying (PSK): Symbols are represented by changes in the phase of the signal.
- Frequency Shift Keying (FSK): Symbols are represented by changes in the frequency of the signal.
- Amplitude Shift Keying (ASK): Symbols are represented by changes in the amplitude of the signal.

- Quadrature Amplitude Modulation (QAM): Symbols are represented by changes in the amplitude of two quadrature signals.

In the design, two codification schemes are available: Binary Phase Shift Keying (BPSK) and 16 Quadrature Amplitude Modulation (16-QAM) [20]:

- Binary Phase Shift Keying (BPSK):
  BPSK is the simplest form of Phase Shift Keying modulation. In this modulation, only two different codified symbols can be transmitted, "0" and "1", represented with a phase of 0° and 180° respectively. BPSK is the digital modulation with highest Signal to Noise Ratio (SNR) and lowest Bit Error Rate (BER) because the symbols are far away from each other. The BPSK is also the less efficient because only transmits 1 bit per symbol.

- 16 Quadrature Amplitude Modulation (16QAM):
  QAM is a form of Amplitude Shift Keying, but instead of transmitting one signal, two signals are transmitted at the same time: In-Phase and In-Quadrature signals. In 16-QAM, two amplitude signals with 4 different amplitude levels each one are transmitted. Each level of each signal represents two bits "00", "01", "10" and "11". Combining both signals, a maximum number of 16 possible codified symbols are achieved. Thus the efficiency increases compared to BPSK up to 4 bits per symbol.

The modulation used is essential for the system design, and must be adapted to meet the required data rate for the application. In this case, as specified in the OFDM modem description, 32 subcarriers along the 10 MHz bandwidth are available and 5 OFDM data symbols are transmitted. The maximum number of transmitted codified symbols on each cycle time is defined by these two parameters:

$$32\ subcarriers \times 5\ OFDM\ symbols = 160\ codified\ symbols \tag{16}$$

The advantage of the 16QAM modulation over the BPSK modulation is clear, because 16QAM transmits 4 bits each symbol whereas BPSK transmits only 1 bit.

Therefore, in BPSK the amount of bits transmitted each cycle time is:

$$N_{bits-BPSK} = Symbols \times Bits/Symbol = 160\ bits \tag{17}$$

Meanwhile in 16QAM is:

$$N_{bits-16QAM} = Symbols \times Bits/Symbol = 640\ bits \tag{18}$$

In each cycle time more data, apart from the desired data, is transmitted. A 16 Cyclic Redundancy Check (16CRC) block for frame checking as well as a Convolutional Encoder block for adding redundancy are added. That means that the effective number of bits is lower than the calculated above. The 16CRC block adds a 16-bit word to the data, and the Convolutional Encoder has a coding rate of $\frac{1}{2}$, meaning that for one bit of raw data, two bits

are transmitted. Both blocks are described in later lines. Therefore, for each modulation the following number of brute and effective bits are transmitted each cycle time:

Modulation	Number of Brute Bits	Effective Bits	Data Rate
BPSK	160	80	2.56 Mbits/s
16QAM	640	320	10.24 Mbits/s

Figure 23: BPSK vs. 16QAM Bits

The data is processed in words of 16 bits, then,

$$N_{words-BPSK} = \frac{80}{16} = 5\ words \quad , \quad N_{words-16QAM} = \frac{320}{16} = 20\ words \tag{19}$$

In both cases, one word is reserved to transmit the 16CRC, so the effective parameters are:

Modulation	Effective Bits	Total Words	Data Words	Effective Data Rate
BPSK	80	5	4	2.048 Mbits/s
16QAM	320	20	19	9.728 Mbits/s

Figure 24: BPSK vs. 16QAM Effective Data

After modulating the signal, the modem must build the different subcarriers that will conform the final OFDM signal.

The core of the OFDM is the Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) performed by the high efficient algorithms Fast Fourier Transform (FFT) and its inverse (IFFT). These are two computational algorithms used in signal theory to change between time-domain and frequency-domain signals. The signal from modulation block is time-domain, but to construct the OFDM signal, different subcarriers have to be differentiated by the IFFT. Therefore, the modulated signals are mapped in frequency domain and later converted to time-domain signals with the IFFT algorithm.

Since the FFT algorithms work on streams with lengths of power of two, some zeros should be added to each data symbol. The length of the FFT (N_{FFT}) is a key value to the system. The following relationship must be always kept:

$$\frac{Sample\ Rate\ (Hz)}{N_{FFT}} = \frac{Bandwidth\ (Hz)}{Subcarriers} = \frac{1}{T_{symbol}} \tag{20}$$

Since 32 subcarriers and a bandwidth of 10 MHz are specified, both sample rate and FFT points are variables to be chosen. For the purpose, as described in hardware design part of this

chapter, a 40 MSPS Analog-to-Digital converter is used, which limits the maximum sample rate to use. For each modulation, the chosen sampling rate and the points of the FFT are in the next table specified:

Modulation	N_{FFT}	Sampling Rate
BPSK	64	20 MSPS
16QAM	128	40 MSPS

Figure 25: BPSK vs. 16QAM Sampling Rate

Hence, the working frequency of each BPSK and 16QAM program will be 20 MHz and 40 MHz respectively.

To generate the required frequency, an Altera's PLL MegaCore block is used. This block can be configured to take the 50 MHz signal from the circuitry and modify it to obtain a clock signal with a different frequency. This PLL block also allows implementing more than one output independent clocks for a given input.

As defined before, the main characteristic of an OFDM modem is the transmission of several orthogonal subcarriers to increase the spectral efficiency. Sometimes can occur, that the orthogonality between both signals is lost, due to the non-ideal channel characteristics. Losing the orthogonality will cause Inter Symbol Interference (ISI). To avoid this effect, a cyclic prefix is added at the beginning of each OFDM symbol as guard interval.

Furthermore, as described at the OFDM symbols definition, there are the two training symbols introduced after adding the cyclic prefix. The first training symbol is a preamble symbol, and it is necessary to the proper synchronisation of the receiver. This symbol just transmits a complete OFDM symbol with zeros in order to detect the peak of power when the data is received.

To avoid Inter Symbol Interference (ISI) caused by time-shift between symbols, cyclic prefixes before each data symbol are added. These prefixes are added in the reserved guard interval time. If the time shift between symbols is shorter than the guard time, the ISI is completely removed.

4.3.1.2. Data Source

The first designed block is the data source of the program. In this case, a known sequence stored in a local file is transmitted. This sequence contains data with values from 0x0000 to 0xFFFF. This block will send 16-bit words from the local file to the next block at a fixed frequency. This sending frequency is determined by the modulation scheme and the cycle time used.

$$T_{send} = \frac{T_{cycle}}{Data\ Words}\ s \quad , \quad f_{send} = \frac{1}{T_{send}} = \frac{Data\ Words}{T_{cycle}}\ Hz \tag{21}$$

Modulation	Data Words	Sending Frequency
BPSK	4	0.128 MHz
16QAM	19	0.608 MHz

Figure 26: BPSK vs. 16QAM Data Frequency

To generate this frequency, another output clock of the implemented PLL block with the convenient ratio is activated.

After generating the symbols at the sending frequency, these are stored in a RAM module to be read afterwards at working frequency of the system, defined before in FFT description.

At the receiver side, the received data is compared with the data stored in a RAM module, which is identical to the transmitted data. A counter will be incremented by one every time a received bit does not match with the stored bit.

4.3.1.3. Cyclic Redundancy Check

Next to the Data Source block, a Cyclic Redundancy Check (CRC) generator block is implemented. The CRC is a commonly used method to help the receiver to check if there is some data corrupted. The block generates a code, based on the previous Data Words of the same cycle and then adds this code as the last 16-bit word of the transmission. The receiver, once it has received all the data words in one cycle, calculates the CRC and compares the result with the given CRC value from the transmitter in the last word. The block has inside an error counter, which is incremented by one every time the CRC calculated is not the same as the value received in the last block and CRC valid and CRC bad output signals, which are asserted to indicate if the CRC is valid or not.
To generate the value, successive modulo 2 divisions on the data stream are performed, storing the remainder. In these divisions, the received data is divided by a 17-bit word "11000000000000101" defined by the polynomial $x^{16} + x^{15} + x^2 + 1$.

To perform this calculation in both directions the Altera CRC Compiler MegaCore is available. The IP Core permits different configurations, e.g. configure it as CRC generator or CRC checker, the number of channels, the data width or the CRC code. In this block, a parallel to serial conversion is also performed. This is done because the next block needs serial data in order to do the convolutional encoding [21].

4.3.1.4. Convolutional Encoding

Since the transmission channel is not ideal, distortions over the signal can be produced. For bit error detection and correction, a convolutional encoding method is used.

In the transmitter part, the incoming bit stream of length L is transformed into another of length M, where M is greater than L. The output stream depends on the input at the current cycle and also on the input of the previous cycle.

In the receiver part, for error correction, a Viterbi error correction algorithm is used. It is based on maximum likelihood principle, which is a statistical model that can estimate the most probable message given a received sequence. For implementing this feature, the Altera Viterbi IP MegaCore is available.

4.3.1.5. Interleaver

As a complement of the convolutional encoding, an interleaver is also implemented. Its main goal is to prevent burst errors in detection due to long duration noise. Interleaving a message is a way of organizing the transmitted digital information, changing the order, and spreading several parts of the message along a defined stream of bits, making the detection easier. The difference between a non-interleaved transmission and an interleaved transmission that is affected by noise is shown in the next tables:

Without interleaving	
Message:	aaaabbbbccccddddeeeeffffgggg
Message sent:	aaaabbbbccccddddeeeeffffgggg
Message received:	aaaabbbbccc____deeeeffffgggg

With interleaving	
Message:	aaaabbbbccccddddeeeeffffgggg
Message sent:	abcdefgabcdefgabcdefgabcdefg
Message received:	aaaabbbbccc____deeeeffffgggg
Message reordered:	aa_abbbbccccdddde_eef_ffg_gg

Figure 27: Without Interleaving vs. with Interleaving

In the first case, the most part of the message “dddd” is not properly detected due to the noise. However in the interleaved case, only one bit is missing in each part of the message, so it can be rebuilt with the error correction techniques.

To implement the technique at the transmitter side, a RAM stores the desired data and afterwards a state machine will interleave it following a pseudo-random sequence stored in a file. The opposite process is made at reception to undo the process.

Interleaving introduces some latency to the system because the transmitter has to read the defined stream length in order to reorganize the data, and the receiver must wait until the entire message is received to properly deinterleave the message.

4.3.1.6. Modulation

The modulation is the process of modifying a specific characteristic of an analog signal to transmit the information. In the design, as mentioned before, two modulations are implemented; Binary Phase Shift Keying (BPSK) and 16 Quadrature Amplitude Modulation (16QAM). The block that implements the modulation is the mapping block. This block generates a two-dimensional signal in the complex plane $a + jb$, to map it later in a constellation map. Both signals will be transmitted further at the same frequency but with a phase shift of 90°, representing real plane (cosine), and imaginary plane (sine).

In next figure, both real (I) and imaginary (Q) planes are represented:

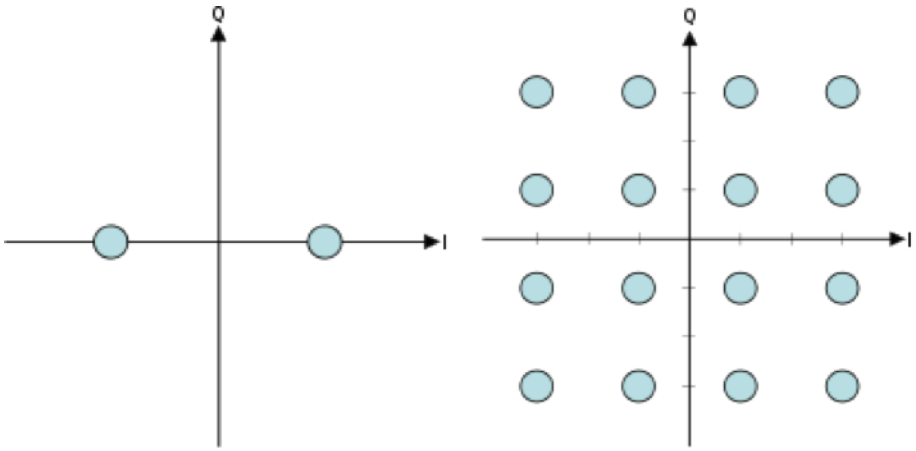

Figure 28: BPSK vs. 16QAM constellation

The constellations are mapped with a Gray codification. That means that between two consecutive values, only one bit is different for an easier detection. Following an example of a 2-bit Binary/Gray codification:

Decimal	Binary	Gray
0	00	00
1	01	01
2	10	11
3	11	10

Figure 29: Binary vs. Gray

As the output signal of the block contains amplitude information, represents an analog signal and thus must have the width of the analog-to-digital converter, which in this case is 12-bit wide.

- In case of BPSK modulation, only two values are available, “000010100000” (160) to represent “1”, and “111101100000” (-160) to represent “0”. These values are in real plane only; the imaginary plane is always zero. Therefore, every input bit will have a 12-bit output.

- In case of 16QAM modulation, four values on each plane are available, “000000110010” (50) to represent “01”, “000010010110” (150) to represent “00”, “111111001110” (-50) to represent “11” and “111101101010” (-150) to represent “10”. Therefore, every 4-bit input will correspond to two 12-bit outputs.

At the receiver part, a symbol recovery block will detect the symbols and will convert them back to digital data.

4.3.1.7. Zero Padding

The zero padding is a process of filling a sampled signal before performing any other process over it. The FFT and IFFT algorithms are more efficient with input data streams of a power of two lengths. To adapt the digital signal to this length, a zero padding block is used. In addition, zero padding is a good solution to prevent aliasing after generating the OFDM signal due to the oversampling.

By adding zeros to the signal the desired signal will be oversampled with a factor of:

$$v = \frac{N_{FFT}}{N}, N_{FFT} > N \tag{22}$$

Where N_{FFT} is the length of the FFT and N is the length of the data symbol.

The aliasing appears for a sampling frequency (fs) lower than the Nyquist frequency

$$F_N = 2 \times B \tag{23}$$

Where F_N is the Nyquist frequency and B is the bandwidth of the sampled signal.

When sampling, the spectrum of the sampled signal is 2π periodic, i.e. is repeated each fs, appearing alias of the original signal .In an ideal situation, with a perfect filter response and meeting the previous condition the alias are outside of the filter and aliasing does not appears.

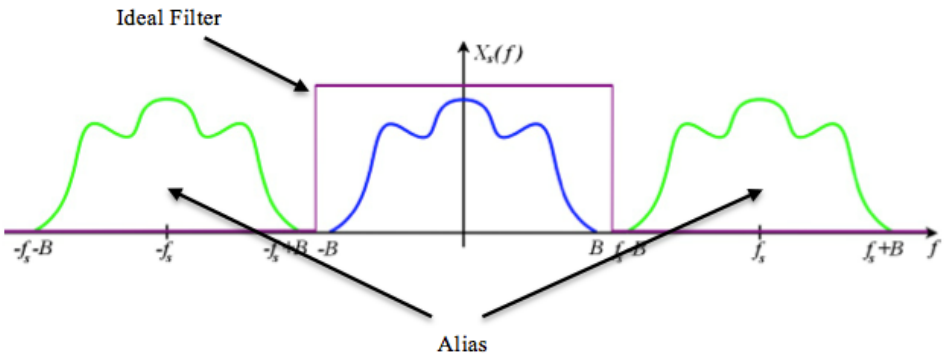

Figure 30: Ideal filtering

In a real situation, where the filters are not ideal, it might occur that part of the alias goes inside of the pass band of the filter, mixing with the desired signal and interfering with it.

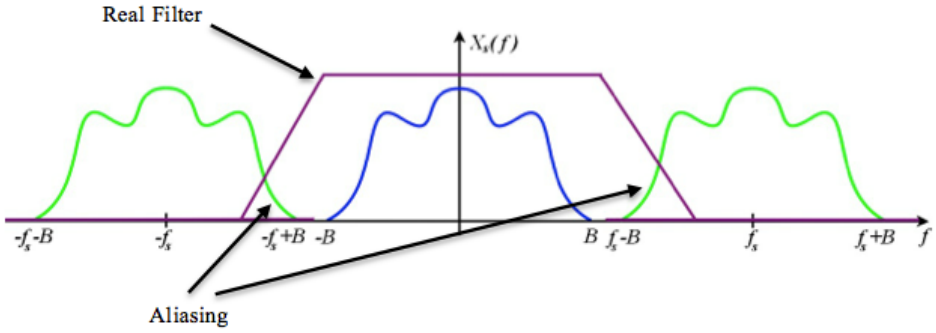

Figure 31: Real filtering

The oversampling of the signal prevents the aliasing by moving away the alias in frequency domain due to the new higher sample frequency.

Because the spectrum will be 2π periodic and it is impossible to transmit negative frequencies in base-band, the IFFT vector represents the frequencies between 0 and 2π, i.e. between 0 and fs. The start of the vector represents the lower frequencies meanwhile the middle of the vector (π) represents the high frequencies. The end of the vector represents the negative frequency components of the spectrum, which contains the half of the transmitted subcarriers. After the generation, the transmitted signal is 2π periodic and disposed as shown before.

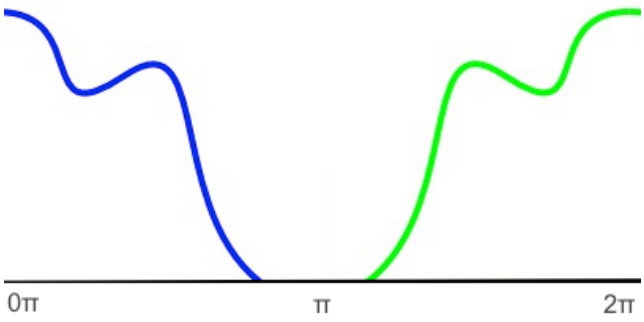

Figure 32: Periodic signal representation

To preserve the negative frequency components of the signal, which contain modulated information, and to preserve the conjugate symmetry of the vector, the zero padding must be done by adding the zeros in the middle of the stream, instead of just adding at the end. If this is not done, the negative spectrum part will be in higher frequencies that the positive part and then, will be filtered afterwards.

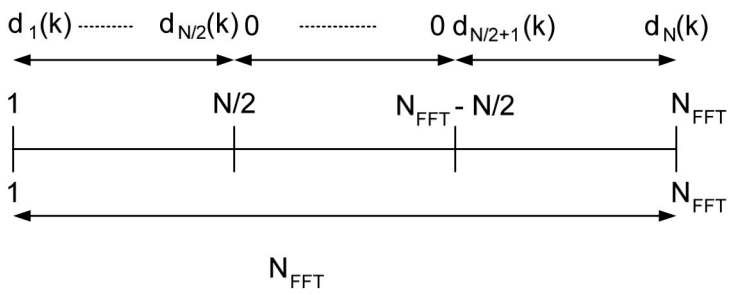

Figure 33: Zero Padding "in the Middle"

4.3.1.8. Fast Fourier Transform

As commented before, the Fast Fourier Transform (FFT) algorithm is the core of the OFDM modem and it is based on the Discrete Fourier Transform (DFT). Thanks to this algorithm the transition between time-domain and frequency-domain signals is possible. The FFT transforms a time-domain signal into a frequency-domain signal and the IFFT transforms a frequency-domain signal into a time-domain signal. An Altera's FFT MegaCore included in each transceiver performs both processes. It allows different configurations, including number of points and data precision [21].

The DFT defines the frequency-domain signals (X_k) as transformation of the time-domain signals (x_n) and vice-versa with N points.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-i2\pi kn}{N}}, k \in \mathbb{Z} \tag{24}$$

$$x_n = \frac{1}{N} \sum_{k=o}^{N-1} X_k \cdot e^{\frac{i2\pi kn}{N}}, n \in \mathbb{Z} \tag{25}$$

Each X_k is a complex number that encodes amplitude and phase of a sinusoidal component of function x_n.

The Altera's FFT MegaCore function performs a radix-n decimation-in-frequency FFT. This FFT algorithm redisposes the DFT calculation in two parts: computation of even-numbered discrete-frequency indices (X_k for k=[0,2,4,…,N-2]) and computation of the odd-numbered indices (X_k for k=[0,1,3,…,N-1]). Each DFT calculation part can be independently performed for an N/2-point DFT, multiplying the odd-indexed frequencies by the twiddle factor term $W_N^k = e^{-(\frac{i2\pi k}{N})}$

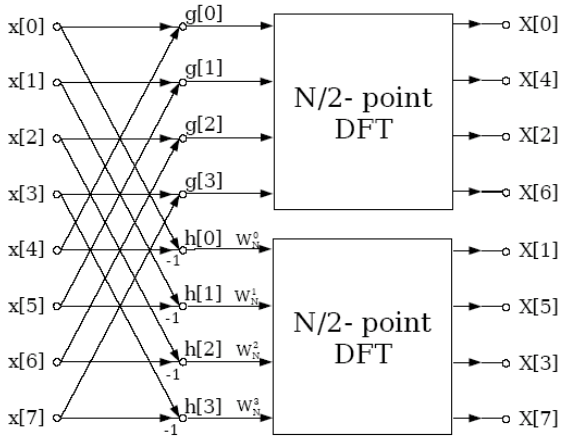

Figure 34: 8-Point FFT

After the computation, only a reordering of the resulted X_k components is needed [23].

Since DFT and IDFT are equals except for the 1/N factor, the IFFT computation is performed the same way.

To transmit the different subcarriers, generated data amplitude signals are disposed in the X_k vector. This vector will contain the signal with the data represented in the frequency domain as defined previously, i.e. 32 subcarriers with 312.5 KHz bandwidth each one. Then the IFFT is applied, transforming the signal to time domain generating the OFDM symbols and sending them forward.

At reception, the opposite sequence is applied. The OFDM symbols are split as they are received, and then the FFT is applied to each OFDM symbol to recover the original signal that contains the modulated data symbols.

4.3.1.9. Preamble and Prefix

After the IFFT block, the previously described functionalities of Cyclic Prefix and Preamble are implemented.

4.3.1.10. Receiver Synchronisation

For the synchronisation at the receiver part, the algorithm of Timothy M. Schmidl and Donald C. Cox is implemented, with a modification in the method; the autocorrelation of both real and imaginary received signals is continuously calculated.

The original method describes the use of a two-symbol training sequence for achieving the detection in two steps. The first step includes the frequency-offset correction by searching for a symbol, whose first and second halves are identical. Afterwards the second step performs a correlation with the second symbol to find the carrier frequency offset.

For the modified method, the detection of the data frame is only performed with the correlation of the guard interval instead of the whole training symbol.

Real and imaginary signals have the same preamble, so the value should be the same at the start of the reception. After this, the integration of both values will provide the power of the signal. To detect the start of the signal a predefined value is provided as threshold, avoiding the detection due to noise. This value should be adapted to the characteristics of the channel, in order to detect properly the start of the signal.

4.4. Ethernet POWERLINK and B&R System

Instead of using a dSpace system to command the drive, now an industrial communication protocol, Ethernet POWERLINK is available. It is a complete software solution to be implemented over IEEE 802.3 Ethernet standard. That means that Ethernet POWERLINK can be implemented on every standard 802.3 Ethernet device, just by programming it. The absence of proprietary hardware makes easy the implementation and takes the benefits of Ethernet flexibility and extra hardware such as Hubs, Switches etc [24].

Some of the advantages of implementing the Ethernet POWERLINK Protocol are:

- Support IP based protocols as TCP or UDP.
- Synchronize networked nodes with high precision.
- Possibility of implementing the protocol over a standard 802.3 Ethernet device.
- Avoid of collisions by the time division access. Only one node transmits at once.

4.4.1. Ethernet

POWERLINK uses standard Ethernet frames for data transfer. An Ethernet frame contains several sections to transmit and address the data. It begins with a 7-byte preamble (Pre) and 1 byte Start-of-Frame delimiter (SFD) followed by a 6-byte destination address (DA) and a 6-byte source address (SA). The next 2 bytes represent the EtherType (T). Then come the payload (minimum 46 bytes) and a 4-byte CRC-32 checksum to check the integrity of the data.

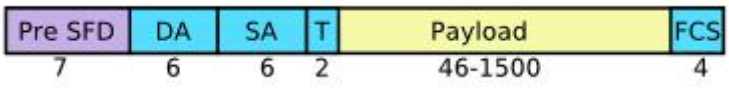

Figure 35: Ethernet frame

Then, the maximum frame length is defined as 1518 bytes of which maximum 1500 can be of payload. Both source address and destination address are unique MAC addresses.

4.4.2. Protocol Description

In order to achieve real-time capabilities, Ethernet POWERLINK mixes polling with time-slot methods allowing only one node to transmit data, in contrast to basic Ethernet protocols, where an arbitration system is needed in order to avoid collisions.

In Ethernet POWERLINK two different nodes can be described:

- Managing Node (MN): Moderates and decides which node can transmit (usually an Industrial PC).
- Controlled Node (CN): All other devices controlled by the Managing Node (sensors, servo drives and others).

To communicate with the Controlled Nodes, the Managing Node establishes a cycle time and defines the clock pulse to synchronize all the nodes. During this cycle time two different phases are defined, Isochronous Phase and Asynchronous Phase.

First, before Isochronous Phase, the Managing Node sends a "Start of Cycle" (SoC) frame to tell all the nodes the start of the cycle and synchronize them. Afterwards, the Managing Node starts polling each Controlled Node one by one with the "PollRequest" (PReq) frame. After each poll, only the asked node has a defined maximum response time to answer with a "PollResponse" (Pres) frame. In this phase, time-critical information is exchanged between nodes.
After Isochronous Phase, Asynchronous Phase starts with the transmission of the "End of Cycle" (EoC) and "Asynchronous Invitation" (AsynInv) frames, allowing one Controlled Node to transmit non-critical data. The Managing Node can poll each cycle a different Controlled Node. During this phase, information such as TCP/IP data or parameter configuration is exchanged. This phase is not mandatory and, although it must be considered in the cycle time, asynchronous data is not transmitted on every cycle.

Ethernet POWERLINK allows different network topologies, star, tree, line, ring or any combination of the structures are available. Furthermore, no extra configuration is needed [24].

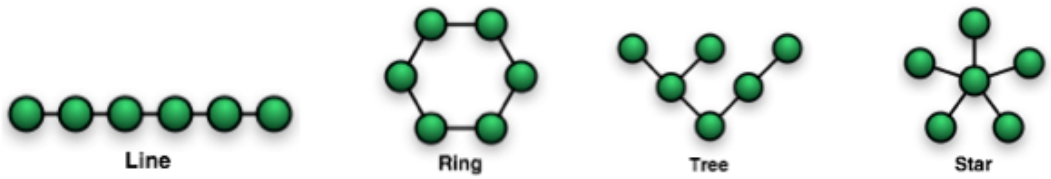

Figure 36: Network topologies

4.4.2.1. POWERLINK v1 vs. POWERLINK v2

There are two versions of POWERLINK available POWERLINK v1 and POWERLINK V2. The POWERLINK v2 meets the previous characteristics and adds additional features.

The main characteristic of POWERLINK v2 is the standardized application layer to the open IEC standard as an extension of POWERLINK v1, which is B&R proprietary. Furthermore POWERLINK v2 introduces other features:

- Introduces a unique "Start of Asynchronous" (SoA) frame instead of "End of Cycle" (SoC) and "Asynchronous Invitation" (AsynInv) frames.

- Support for safety products.
- Integration of OEM devices by introducing XML files.
- Intelligent Controlled Node (iCN).
- Configurable size of the asynchronous channel.
- PollResponse MN frame possibility.
- PollResponse chaining mode.
- DNA support.
- Continual performance optimization.

There is a slight change between the POWERLINK v1 and POWERLINK v2 cycles.

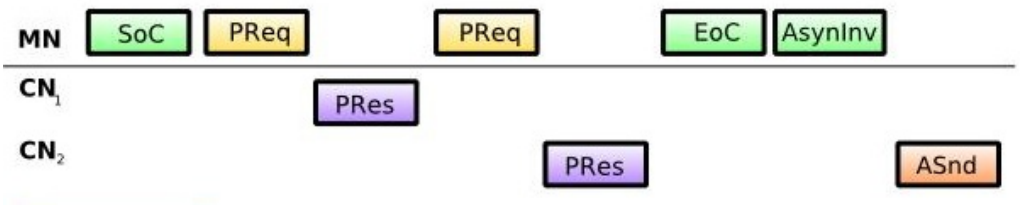

Figure 37: POWERLINK v1 cycle

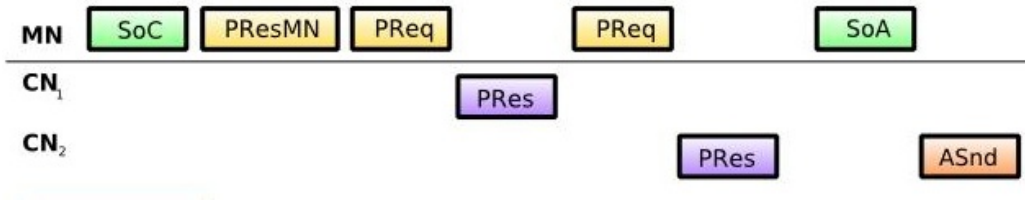

Figure 38: POWERLINK v2 cycle

In POWERLINK v2 a PresMN frame after the SoC frame can be transmitted. This is included for the "poll response chaining" feature, implemented on POWERLINK v2. With this feature, the controlled nodes are queried all at once by the PresMN frame, which is sent as multicast. All the nodes receive all the information and they answer sequentially.

4.4.3. B&R System Description

In the industrial automation environment, B&R Company develops different solutions for automation systems such as industrial PCs, control systems, networks and fieldbus modules, having presence in several industries.

For this purpose, B&R handles the development of hardware as well as software solutions.

4.4.3.1. Hardware topology

The available target system is a B&R Automation system that consists of an Industrial PC (IPC), which controls two Servo Drives; each one drives an electric motor. Furthermore, an HMI touch display and a keyboard are also integrated in a single solution.

The specific devices available are:

- A B&R 5PC810.SX01 Industrial PC.
- Two ACOPOS 1016 Servo Drives.
- Two Servo Motors.
- Two POWERLINK Bus Couplers.

To program and command the Industrial PC, an external computer is connected with it through an Ethernet cable.
The whole system works with either 24V DC power supply or with 230V AC supply. The way to supply each servo drive depends on the connectors placed on each servo drive (one supplies 24V DC and the other supplies 230V AC) and on a specific parameter on the software present in an external computer.

The servo drives include three expansion ports that allow expanding the capabilities. In the supplied system, one Ethernet POWERLINK module (with two Ethernet connectors) and one RS232 interfaces are included on each servo drive.

The POWERLINK bus couplers are connected to the drives in order to manage the information from/to the sensors and actuators of the drives.

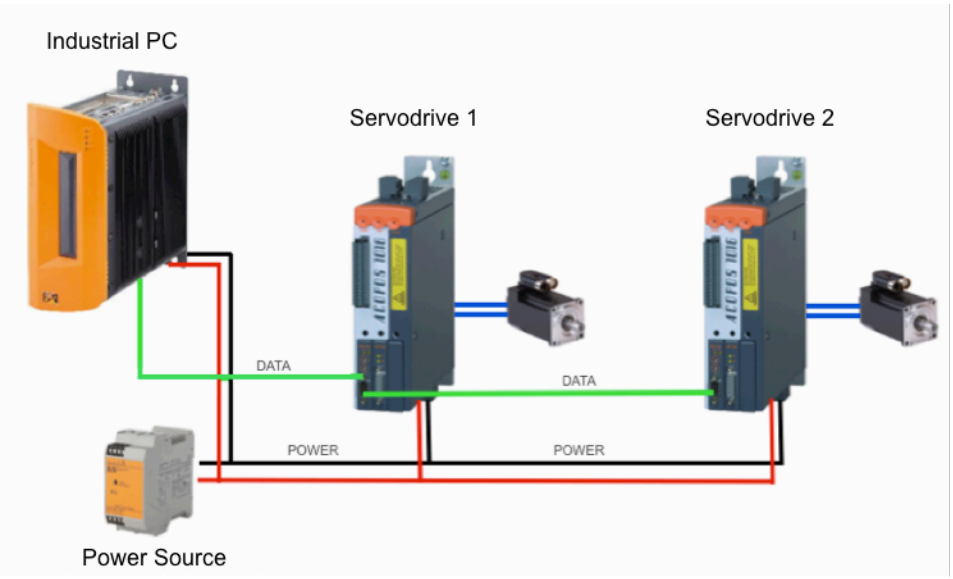

Figure 39: B&R system

The connection between the IPC and the nodes is possible thanks to the Ethernet POWERLINK ports. Each port has a half-duplex connection with either previous node or next node. One connector receives and sends from/to the previous node; the other receives and sends from/to the next node. The network topology is hence a line topology.

When a frame is transmitted from the IPC, the first node receives the frame through one connector. If the node is the destination of the frame, it will respond back through the same connector. If it is not, it will forward the frame through the second Ethernet port to the next node with a minimum delay.

4.4.3.2. Software

B&R Automation Studio

B&R Automation Studio is an integrated software development environment, which includes tools for all parts of an automation project, making in the foundation for applications of any size and scope.

For every project, several stages are defined: planning, implementation, testing, production, commissioning and service. For every stage, Automation Studio always provides an interface to the machine. Therefore the Automation Studio offers solutions for:

- Creating programs
- Creating visualizations
- Creating motion applications
- Running diagnostics
- Configure communication parameters

For the described system there is a project already implemented, which operates the two described drives. For each drive a specific is created with variables and operation commands.

B&R Automation Runtime

The target system runs a Windows XP embedded version to manage the devices. This Windows XP version offers Real-Time capabilities. Installed on the IPC the B&R Automation Runtime (ARwin) software is fully embedded in the target system. It allows application programs to access I/O systems and other devices. There are different types of target systems defined by the processor type and architecture used in the system. B&R uses several platforms defined by the generation of the system.

- System Generation 3 (SG3)
- System Generation Compact (SGC)
- System Generation 4 (SG4)

The used system is a SG4 and thus, the applied configurations will be determined by this platform.

4.4.4. Real-Time System Configuration

Programming and configuring the real-time system is basic to drive the servomotors. For the communication, in order to transfer the program and the configuration, a direct connection between the external pc and the target system is needed. To establish the communication, an Ethernet cable is available to connect the external PC with the target system. Then, an IP configuration must be performed to allow both computers to connect.

This section introduces also the POWERLINK parameters for real-time communication such as cycle time or response time that are essential for the proper working of the system. This

configuration will be performed in the Automation Studio and afterwards transmitted to the target system.

4.4.4.1. Network Configuration

As described, a network configuration in both IPC and the Engineering computer must be done, to allow the communication. For this purpose ARwin offers a configurator program. Furthermore a guide in the Automation Studio is provided.

The configuration consists on reserving three IP addresses of the same subnet:

- Two IP addresses are for the target system. Of these two, one IP address is for an external access to the system, and the other for a virtual port, which communicates the RTOS with the drives.
- The third IP address is for the external PC, where the Automation Studio is installed. The external IP of the target system will be configured in the Automation Studio to access to it.

4.4.4.2. Ethernet POWERLINK Parameters

In this section, the most important parameters for the project will be introduced and described. The adaptation of these parameters to the topology of the system is essential to the proper functioning and the quality of the real-time communications. Two groups of parameters are described, parameters of the managing node side and parameters of the controlled node side [25].

Managing Node parameters

1. Module Type
   The module type parameter determines the functions that are available for a POWERLINK interface module. This is a read-only value and cannot be changed.
2. Operating Mode
   Specifies the POWERLINK protocol that the network should work with. The available modes are: Ethernet, POWERLINK V1 and POWERLINK V2 (default value)
3. MTU Size
   Determines the size of the data range of an asynchronous POWERLINK frame. It can take different values depending on which operating mode is the system in: 1500 for Ethernet, 262 for POWERLINK V1 or a value between 300 and 1500 of POWERLINK V2 (default 300).
4. Baud Rate
   Determines the baud rate on 1-port Ethernet modules. For Ethernet mode 100 Mbit auto and 100 Mbit half duplex is available. For POWERLINK V1 and POWERLINK V2 only 100 Mbit half duplex is available.
5. Cycle time
   Establishes the POWERLINK cycle time in microseconds. It can takes values from 400 to 30000 for POWERLINK V1 and from 200 to 100000 for POWERLINK V2.

6. Mode
   Specifies the mode used to operate the interface between Managing Node (default), Controlled Node or Secondary Managing Node.
7. Asynchronous timeout
   This parameter can be used to set the timeout for asynchronous frames in microseconds. It is defined between 5 and 30000 for POWERLINK V1 and between 5 and 100000 for POWERLINK V2. The default value for POWERLINK V2 is 25 microseconds.

Controlled Node parameters

1. Mode
   Specifies the mode used to operate the interface between Managing Node (default), Controlled Node or Secondary Managing Node.
2. Response timeout
   Specifies the response timeout for the node in microseconds. Can take values from 1 to 30000 in both versions of POWERLINK, default value is 25 microseconds.

3. Synchronization Mode
   This parameter is available for POWERLINK bus controllers and determines the way of synchronization of the Controlled Node. The different options are Start of POWERLINK cycle as sync (default), Poll request as sync, Poll response MN as sync or no synchronization.

5. Ethernet POWERLINK over Powerline

After describing the basic system, where this project started from, and the system, where the modems are implemented, this chapter introduces and describes the updates performed in software as well as in hardware designs to transmit Ethernet POWERLINK frames.

One of the key concepts of this task is the transparency of the modems for the B&R system; i.e. the transmitted Ethernet frames must not be modified. Therefore, the modems must transmit the complete received frames without encapsulate or add extra information such origin/destination (MAC layer) or CRC.

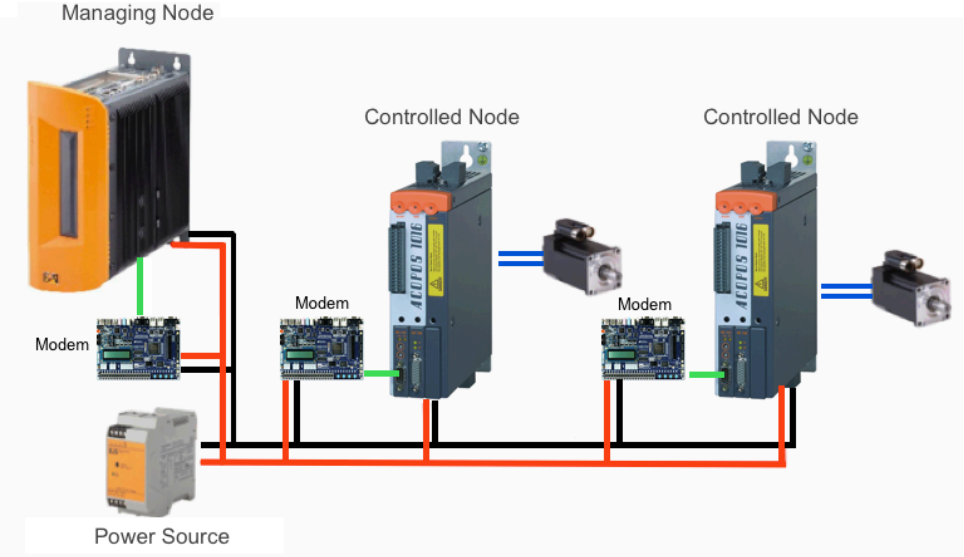

Figure 40: POWERLINK over Powerline implementation

As seen on the previous figure, the network topology is no longer a line topology, but a bus topology. The information travels through the power line cable and each node reads and writes the information from/to the bus.

5.1. Ethernet and Media Independent Interface

Since the goal of the project is to couple an Ethernet POWERLINK signal to the described OFDM modem, first a theoretical introduction of the Ethernet transmission characteristics is needed.

The first step is defining the way of data transmission along the Ethernet cable. The target system specifies a 100BASE-TX Ethernet standard for data transmissions in a half-duplex mode. That means, only one direction of communication at a given time is possible. The 100BASE-TX is the most used Fast Ethernet specification, and uses Cat 5 Ethernet cables for data transmission. These cables have four pairs of wires inside, but only uses two of them,

one pair for transmission and other for reception. In addition, the transmitted signals are differential and the pairs are twisted for noise cancelling.

Before transmitting the data through the Ethernet cable, the data must be adapted. To this purpose the described Marvell 88E1111 chip is available. This chip allows two different media independent interfaces modes as described in hardware design part. To match with the POWERLINK system, the MII mode is selected by shorting connectors 2 and 3 of jumper JP1 and JP2 on the Terasic development board as indicated in the next figure:

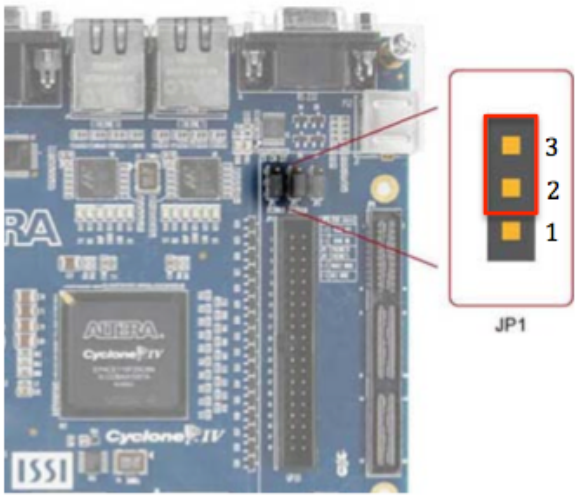

Figure 41: MII Board configuration

The MII transfers 4-bit words in each direction clocked at 25 MHz to achieve 100Mbit/s speed. Selecting the proper mode in the FPGA is essential to make the transmission properly matching the configuration with the connected devices. After taking the 4-bit words, the MII controller encodes the data through a 4B5B encoding clocked at 125 MHz to send it further to the Ethernet cable [26].

To manage the Marvel controller chip, as well as to provide MAC and CRC support, an Altera's MegaCore IP block is provided, the Triple Speed Ethernet IP Core. Since the system needs to be transparent and no MAC support is needed, this block will not be used, and the signals must be managed directly from the designed blocks.

At the software part, the provided signals by the MII are the following:

Transmitter	
TX_CLK	Transmit Clock
TXD(3..0)	Transmit Data
TX_ER	Forces transmission of illegal code
TX_EN	Enable Transmit Data

Figure 42: Transmitter Ethernet signals

Receiver	
RX_CLK	Receive Clock
RXD(3..0)	Received Data
RX_DV	Receive Data Valid
RX_ER	Receive Error Indication
COL	Collision Indication
CRS	Non-Idle Medium Indication

Figure 43: Receiver Ethernet signals

The most relevant signals for the Ethernet coupling are TX_CLK, TXD(3..0), TX_EN, RX_CLK, RXD(3..0) and RX_DV.

For transmission, the wanted data is written at 25 MHz clock in the TXD (3..0) bus. At the same time, TX_EN line must be asserted to indicate the validity of the transmitted data. TX_EN must be kept during all the time valid data is transmitted. TX_CLK is a clock source of 25 MHz from the Marvell controller for data clocking.

At reception part, RXD(3..0) data signal is provided by the MII at 25 MHz The 25 MHz reference clock is on RX_CLK signal and the validity of the received signal is assured while RX_DV is asserted [5].

5.2. Coupling Principle

As described before, words of 4 bits clocked at 25 MHz are transmitted to the MII to be sent to the Ethernet cable. Since the OFDM modem works with a 16-bit width bus, an adaptation block is necessary.

The main function of this block is taking 4-bit words from the MII Ethernet at 25 MHz and making 16-bit OFDM words to be transmitted by the modem (receiver_ethernet). At reception side, another block (transmitter_ethernet) will undo this operation in order to transmit data further to the Ethernet cable. Both blocks operate at a frequency of 100 MHz in order to process incoming data faster than received.

The special characteristics of the modem i.e. the maximum data rate of 9.728 Mbits/s, which is far lower from the Ethernet standard data rate of 100Mbit/s or the fixed number of 16-bit OFDM words transmitted each cycle time, will determine the design of these blocks.

5.3. Software Designs

5.3.1. Ethernet Coupling Design

In this section, the first implementation of the desired feature, the Ethernet Coupling, is described. The implementation is designed in two blocks, receiver_ethernet and transmitter_ethernet that are placed inside the transmitter and receiver modem blocks respectively. These blocks have the main task of receive/send data from/to the Ethernet

connectors to make the transmission possible. Both blocks work with a frequency of 100 MHz, which is fast enough to process the signal, as well as with a 25 MHz frequency, which is the data coming frequency, and the sending frequency of the OFDM modem, 0.608 MHz.

5.3.1.1. Receiver_ethernet block

A described in the introduction of the chapter, the main target of the receiver_ethernet block is receiving 4-bit words from MII, clocked at 25 MHz, making words of 16 bits, and sending further to the CRC generator block of the modem at a frequency of 0.608 MHz (clk_process).

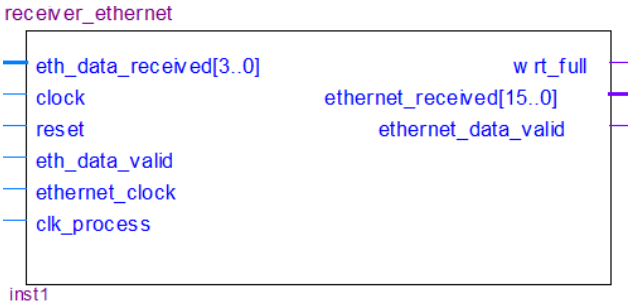

Figure 44: Receiver_ethernet block

This block is placed inside of the data generation block and substitutes the BER generator block, which contained the known sequence to be transmitted in the original program.

For the implementation, three stages are implemented inside of this block.

1. Storage of 4-bit received words:

In this stage, after eth_valid (RX_DV) rising edge, four independent 4-bit registers store the received 4-bit word at a certain clock time. A finite state machine commands the load signal of every register, asserting the registers sequentially from the rising edge of eth_valid signal until eth_valid is denied.

2. Packing received words:

After receiving the fourth 4-bit word, a register packs the four words from the registers. Then, a finite state machine, zero_send, reads the register and sends every word to a FIFO register, asserting the write enable of the FIFO. The FIFO is needed because the reading frequency of the modem (0.608 MHz) is lower than the frequency of the incoming data. The size of the FIFO must be chosen to be able to store enough data without overflow. Since the protocol specifies a poll-request communication (i.e. only one node is asked at once), the largest data frame transmitted from the Node will determine the minimum capacity of the FIFO. In Powerlink, the Start of Cycle (SoC) and the Poll Request (PReq) to the first node frames are transmitted sequentially, so they can be considered together. The sizes of these frames are specified later on chapter 6.4 and will give an approximation of the needed size of the FIFO.

3. Sending to OFDM modem:

The last stage of the receiver_ethernet block is a finite state machine, data_send, which plays an important role in the coupling with the OFDM modem. This block will command the read request signal of the FIFO at 0.608 MHz at the right time in order to send the information to the next block.

5.3.1.2. Transmitter_ethernet block

The main purpose of this block is receiving the packets received by the Powerline modem at a frequency of 40 MHz, and packing them. Then, transmit the result data further through the Ethernet connection at a frequency of 25 MHz

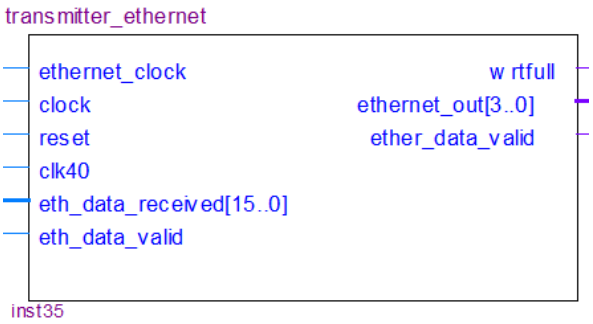

Figure 45: Transmitter_ethernet block

The transmitted_ethernet block is placed after the data symbol detection of the OFDM receiver.

To implement the functionality, two stages were designed inside of the Transmitter_ethernet block.

1. Storage of received data: First stage is a FIFO that stores the received data clocked at 40 MHz in order to read it after at a 25 MHz clock rate.

2. Processing and transmission to Ethernet: Then, a finite state machine (piso_fsm) takes the output and splits it in four packets of four bits for sending forward to a transmission FIFO. This FIFO has a writing frequency of 100 MHz and a reading frequency of 25 MHz.Once this is done, the same state machine controls the reading from the FIFO directly to the Ethernet port as well as the assertion of the Ethernet data valid signal.

5.3.2. Additional Designs

Apart from the available main program blocks, some blocks were designed to make measurements of latency and maximum data rate. These blocks will be only used in specific programs for testing and demo purposes.

5.3.2.1. Latency Measurement

To measure the latency of the system in the transmission of a concrete transmitted 16-bit value, a specific block was designed. This block counts clock cycles from the time a value is generated at the transmitter part, until the same value is detected at the receiver part. Then, the value is converted to seconds by multiplying by the clock frequency and displayed on the available LCD display.
To measure the latency with this block, only a single board must be used, connecting the transmitter part with the receiver part.

5.3.2.2. Data Rate Measurement

To measure the data rate of the system another specific block was designed. This block counts the number of valid blocks received in the interval of one second. For this, every time the CRC check block asserts the data valid to "1" (that means that all the packets of the last cycle are properly received), the counter will increase one value. After each second, the total value of this counter will be multiplied either by 4 (in the case of BPSK) or by 19 (in the case of 16QAM), in order to get the amount of packets received. The resulting value is multiplied by 16 to obtain the total amount of bits received within one second. The final result is the amount of bits per second, e.g. the data rate. This calculation is done every second.

As in the latency measurement block, the data rate is showed on the LCD display.

5.4. System Implementation

5.4.1. Software

5.4.1.1. Migrating and Merging

The old FPGA worked with some libraries that were not compatible with the most recent version of the Altera's development environment available, and thus an older version of the program was needed. But this old version didn't support the Altera's MegaCore IP libraries of the new FPGA board, used for this project. Therefore in order to adapt the existing software, an update of the MegaCore IP blocks of the programs was needed to make possible the migration to the new board.

Moreover, as described in previous lines, a transceiver program for the transmission and reception at the same board was needed. This task presented the issue that, due to the independency of the program design on each part (receiver and transmitter), some files had the same name in both designs, but with different functionality. Hence when everything was placed in one folder, the compiler was not able to identify which was the correct file for each block, either transmitter or receiver. Therefore, in order to be properly recognized and compiled, these files had to be identified and changed.

5.4.1.2. Ethernet Coupling

The Ethernet coupling presents some special subjects in the software part. These were:

1. The transmitter_ethernet block must transmit the whole Ethernet frame as the receiver_ethernet block has received it. Since an Ethernet frame has usually more data than the data transmitted in one OFDM cycle time, this will infer in the transmission of several OFDM packets to transmit one Ethernet frame.
2. The OFDM modem only sends data to the channel after the reception of the 19th word, which implies that for a given cycle that it will wait until the reception of this last word, although there is no more data to transmit.
3. The fact of sending more than two consecutive words of zeros will affect to frame synchronization at reception.

The solution achieved is described in the next points.

1. Storing and delimitation of the Ethernet frame

The amount of data transmitted each OFDM cycle time determines the design of the Ethernet coupling because, unlike the Ethernet, which transmits always data at a certain speed and there is not a cycle time, the OFDM cycle sends on every cycle a determined number of packets. This has implications in terms of data rate, response time and also represents an issue to face with.

First, the OFDM transmitter has to receive the data from the MII clocked at 25 MHz and has to adapt it to the modem characteristics and speed, i.e. packets of 19 words of 16 bits, and a sending frequency of 0.608 MHz. Then, the OFDM receiver must store the received data converting again to the 4-bit Ethernet words, and transmit at 25 MHz frequency. However, the receiver cannot know the limits of the received Ethernet frames, because the received words only contain data, and the modem speed is several times slower than the Ethernet speed. Furthermore, the possibility that a 16-bit word is not filled with 4-bit words, due to the end of the reception, must be covered. To deal with this issue, the receiver_ethernet block was updated.

Each 4-bit register of the first stage is substituted for a 5-bit register to store the received four bits and also the data valid bit. Moreover, the finite state machine, which controls the load signals of the registers, is updated to fill always the 4 registers. Hence, if there is no more data but any register is not filled, it will be filled with the available data (“0000”) and the corresponding data valid, a “0” (i.e. not valid). This will help to discard later the invalid 4-bit words. Then, a word of 20 bits is built and registered.

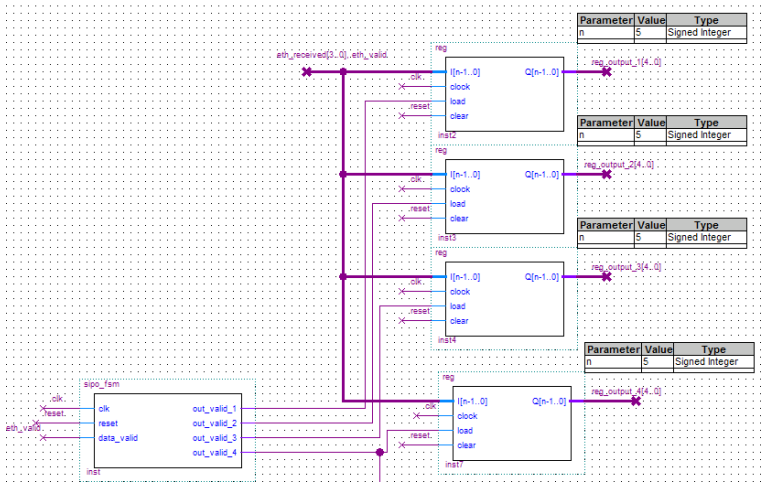

Figure 46: Ethernet Data storage stage

After this stage, the zero_send block is also modified to check if there is new data incoming after the current built word. This is performed checking directly eth_valid signal, which will be denied before sending the current word (this signal is some cycles delayed with respect to the current eth_valid signal). Zero_send machine includes now an output bit, last_packet, which indicates whether the built word is the last one. This signal is added to the sent data to the FIFO, which has a 21-bit input (16 data bit plus 4 data_valid plus 1 last_pkt). For the case, if the word is not the last one, last_packet is asserted to "1". If there is no new data incoming, the zero_send block asserts last_packet output to "0". This bit will be only read by the next stage and discarded afterwards.

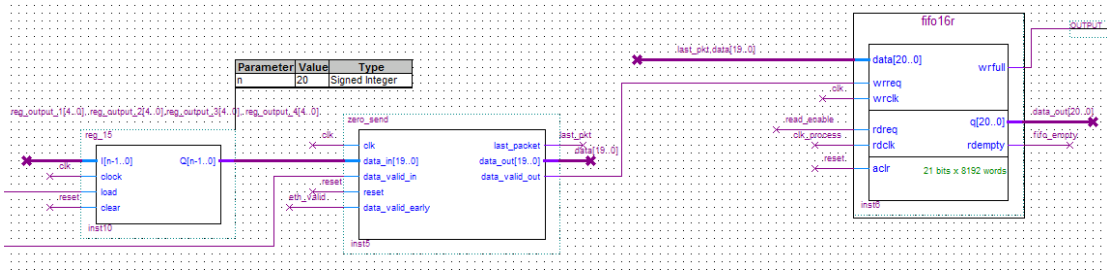

Figure 47: Ethernet Data Treatment Stage

Due to the lower speed of the modem compared with the Ethernet, several different Ethernet frames can be stored sequentially inside of the FIFO making impossible to differentiate them (e.g. when a second stream arrives before the first has been completely transmitted). To solve this issue, after sending the last value, a 0x0000 word is also stored in the FIFO to split different frames. Therefore, the 21-bit words will have the next structure:

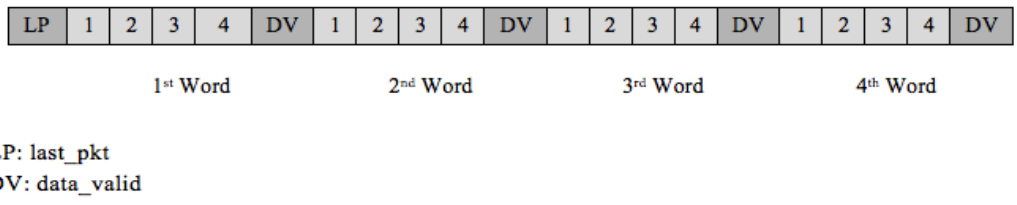

Figure 48: Ethernet Data

After storing the built 21-bit words in the FIFO, the data_send block performs the reading of the FIFO, as well as the 21-bit to 16-bit width transformation.

The proposed solution includes sending 18 words of data and one word of block information, which contains the number of valid received 4-bit words or a last packet value. Hence the transmitted OFDM packets are defined as follows:

1st Data Word	2nd Data Word	3rd Data Word	…	18th Data Word	Number of valid words

Figure 49: OFDM Transmission packet

To perform the proposed solution the data_send block performs the next actions:

- First, checks if the FIFO is empty or not, if it is not empty, asserts the read signal of the FIFO, which reads at a frequency of 0.608 MHz (16QAM-OFDM sending frequency).
- Secondly, two counters are set. One counts the number of data valid bits asserted to "1" on each 21-bit word. This counter gives the number of valid 4-bit words. The other counts the number of 21-bit words on each cycle to make packets of 19 words.
- Then, the last_pkt and the data valid bits are taken out and only the 16 data bits are transmitted.
- When the 18th data word is transmitted to the modem, the data_send block stops reading from the FIFO and writes the number of transmitted 4-bit words as the 19th word of the packet. For this "Number of valid words" field, the maximum value that can achieve, (that is the maximum number of 4-bit words) shall not exceed:

$$Max_{Words} = \frac{16_{bits} \times 18_{packets}}{4_{bits/word}} = 144 \rightarrow 0x0072\ words \qquad (26)$$

- When the last 16-bit available OFDM word of the last packet is transmitted (i.e. when last_pkt bit is to "0") we must cover two possible cases:

    - If this last 16-bit OFDM word is not the 18th word of the transmitted packet, (i.e. the packet is not complete) the [24] number of valid words will be the value of the counter (a value lower than 0x0072).
    - If this last 16-bit OFDM word is the 18th word of the transmitted packet, (i.e. the packet is complete) a defined value greater than 0x0072 is transmitted to let the receiver know, that the frame is finished. The transmitted value will be 0x0080.

After this procedure, the transmitter_ethernet block at receiver part will know exactly when the end of the frame is reached and can build the Ethernet frame again in order to send it further to the destination. Therefore, when an OFDM packet is received different cases are differentiated checking the last word. If the last word of the packet is 0x0072, the transmission continues, if it is less than 0x0072 means the transmission has ended but some 4-bit words have to be discarded because there was not enough incoming data to fill up the

whole packet, and if the last word is 0x0080 the transmission has ended and the whole packet is valid data.

2. Packet Filling

The previously described implementation also has to take care of the cyclic characteristics of the OFDM modem. The modem only transmits 19 words on each packet, and if there are not enough words, the modem will wait until new incoming data in order to fill the packet. This would cause the data delaying or in the worst case, the no transmission of some information if no new incoming data is available. To solve this, data_send block adds a filling (0x0840) when last_pkt is denied. This is just a random number that is chosen to this purpose but has no meaning. The data_valid counter of this block must not count when this filling is introduced use the previous described feature to make the transmitter_ethernet discard the filling words.

3. Frame Synchronization

The frame synchronization is one of the most important parts of the data reception. Without synchronization, the receiver cannot detect the start of a cycle and cannot detect the different symbols properly to demodulate the received data. This synchronization is performed thanks to a transmitted training symbol, which has the same duration as one data symbol. However, with the implemented synchronization method, no power is transmitted in order to detect the power peak when the first data symbol starts. During each OFDM symbol, about 3.8 16-bit words (i.e. 19 words divided by 5 OFDM symbols) are transmitted.

During the test of the described system in the implementation with the Ethernet POWERLINK protocol, after a certain point of a transmitted cycle, the received data was not the same as the transmitted data, but unknown and senseless data. After checking and comparing the Ethernet received data, the sent data through the Powerline and the received data from the modem at the other side, it was proved that at the time a large sequence of zeros is sent (several 16-bit words sequentially transmitted being all zeros), the received data after Powerline was no longer the same as transmitted.

During the transmission of a known sequence, as it was with the Bit Error Rate testing program that sends sequentially words from 0x0000 until 0xFFFF, only a burst of 16 zeros as much is transmitted (when data 0x0000 is transmitted, then comes 0x0001, 0x0002 etc.) and hence, there was no synchronization problems. But with the final implementation with real Ethernet POWERLINK data, sometimes, large bursts of zero bits should be transmitted. These long bursts of zero bits can reach a length of about 300 bits. To compare, the total amount of transmitted bits in an OFDM cycle is

$$19\ words \times 16\ bits/word = 304\ bits \tag{27}$$

Hence, comparing the duration of the training symbol with the transmitted amount of zeros it can be seen that due to the similarity, in terms of time duration, of the transmission three or more (3.8 as calculated before) complete words of zeros with a transmitted training symbol for receiver synchronization, the synchronization of current frame is lost. When received,

these sequential zeros are detected as a training symbol and a synchronization of a supposedly new frame (what in reality it is not) is done causing the loss of information.

To solve this problem, an adaptation of the bit-stuffing concept is implemented. The bit stuffing is a method to avoid the receiver losing the bit synchronization by limiting the sequences equal zeros or ones. For example in a standard as the Universal Serial Bus (USB), the data is Non-Return-to-Zero-Inverted (NRZI) encoded. In USB, the digital value "0" is represented with a transition from a positive value to a negative value whereas the value "1" is represented with no transition. A successive transmission of "1" will cause the losing of the synchronization in reception due to the impossibility of determine the limit between different consecutive bits. To solve this, a "0" is inserted after 6 consecutive "1" bits. Therefore, the addition of one word different to 0x0000 after two 0x0000 sent words is implemented. It is important to mention, that this implementation has to fit with the previous described solution for frame delimitation and filling. Therefore, although this additional word it is not a data word, to make the design slightly simpler, the counter of 4-bit words must increase. The OFDM word counter also must count because is an extra packet, and must fit with the OFDM modem characteristics.

To perform this addition of extra data, after two consecutive 0x0000 words, the data_send block stops reading from the FIFO, by denying read request signal, and sends the "1001001001001001" (0x9249) word, then continues reading. An example of the principle is shown:

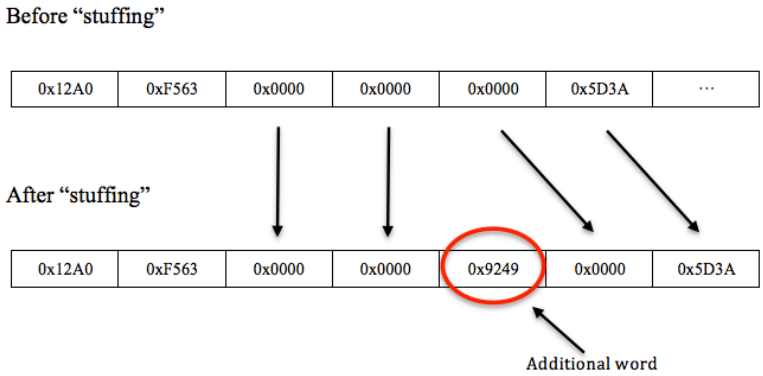

Figure 50: Word Stuffing

On transmitter_ethernet part, the finite state machine will always proceed the same way. After two consecutive zero words the next word is discarded unless the second word of zeros is the 18th word of the packet. Then, the extra packet is not introduced, but the "Number of valid words" packet, as described before.

It is important to notice, that this method introduces extra useless data that will decrease the net data rate, but is an essential task in order to achieve a proper synchronization of the receiver.

To sum up, the Ethernet implementation in the existing system introduces a big delay on the signal when the Ethernet frame exceeds 288 bits $(16\ bits \times 18\ words)$, which is almost sure that will occur. And what is also important, this delay can oscillate, depending always on the

number of required OFDM cycles, if the number of received bits involves a different number of cycles to complete the transmission.

5.4.2. Hardware

During the hardware implementation, it is necessary to adjust and optimize the hardware in order to achieve the best possible signal quality.

5.4.2.1. Channel Response

The first step to implement the system is to know the channel behavior in order to adapt the transmission to the channel characteristics.

To obtain a good representation of the channel response, a Vector Network Analyzer is used.

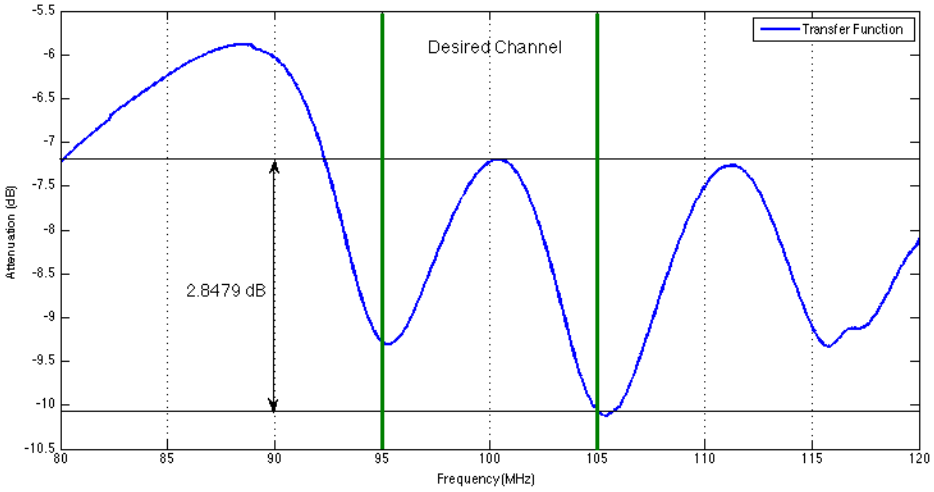

Figure 51: Channel Response

It is noticeable that the channel response is not plain. The difference between the minimum attenuation and the maximum attenuation in transmission bandwidth is 2.8479 dB

5.4.2.2. Baseband Analysis

During the hardware development of the system, the proper and deep understanding of the used hardware is essential to achieve the best optimization.

To detect, and discard possible hardware issues in transmission as well as in order to have a reliable reference of a full functioning demodulator, an evaluation board of the demodulator provided by the manufacturer was used. This evaluation board is developed by Analog Devices and includes everything needed to make the demodulation properly, although it does not include filters or amplifying stages, necessary to work in a noisy environment.

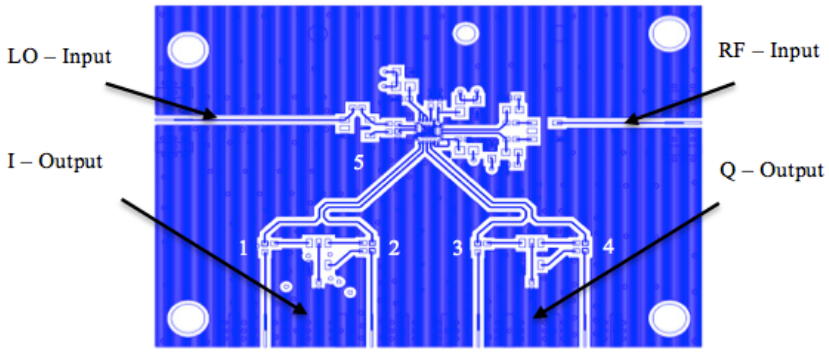

Figure 52: Evaluation board

The evaluation board can be used for both single-ended and differential baseband analysis. As the default configuration of the board is single-ended and the system works with differential signals, it is needed a change in the configuration, bypassing the provided 9:1 TCM9-1 transformers in both outputs by changing the direction of 0 Ω resistors in 1, 2, 3 and 4. Therefore, the signals will go directly to the outputs. To provide a differential local oscillator signal to the demodulator, another 1:1 transformer had to be placed in 5 [14].

Using the described development board and the already implemented transceiver program, it was proved the functioning of the transmitter part; the demodulation was made properly with no Bit Error Rate.

To achieve the same grade of quality signal with the high frequency boards, several measurements with the spectrum analyser as well as with the oscilloscope showed a lack of power at the input of the demodulator. The power at the demodulator input was lower than the minimum necessary and thus the demodulation could not be performed properly.

In the amplification stage on the receiver part of the board, there was lower voltage at the output that at the input as it is shown on the picture of the schematic:

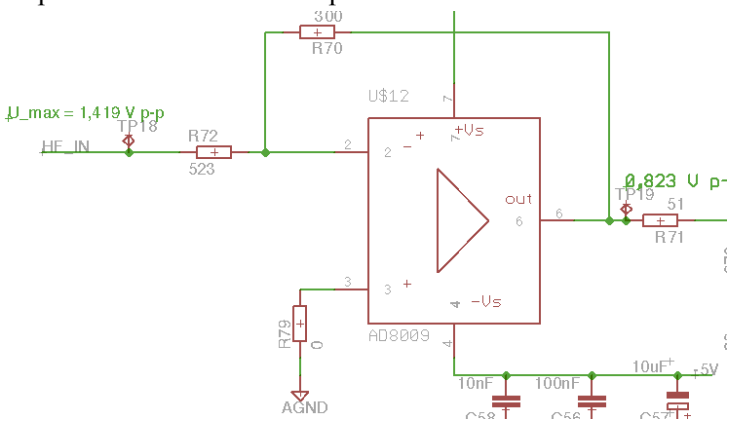

Figure 53: Receiver amplification stage

Looking at design's documentation, where a deeper description of the boards was made, it was expected a gain of:

$$Gain_{exp} = \left|1 + \frac{R_{70}}{R_{72}}\right| = 1.5736 \quad (28)$$

This would be the gain in a non-inverter configuration of the amplifier. But the real configuration is an inverter amplifier, which has a different formula:

$$Gain_{inv} = \left|\frac{R_{70}}{R_{72}}\right| = 0.5736 \quad (29)$$

The issue was solved by changing the relation of the resistors to obtain a gain higher than 1. With the resistors R_{70} = 220Ω and R_{72} = 120Ω a gain of about 1.83 (~2.63dB) is obtained.

5.4.2.3. Channel Access Management

The transmission channel is the same for transmitting and for receiving the signal, and because, the access to the medium is time-divided (i.e. the frequency used for transmission and reception is the same and the medium is only accessed by one transceiver at the same time) each transceiver is connected to one coupling board. This board defines an input and an output port to/from the channel. But these two ports are essentially connected to the same path in the board; there is no possibility to define the direction of the transmission.

Therefore, when a transmission is performed by one transceiver, the half of the transmitted power (~3dB) returns to itself through the receiver port. Due to the high losses provoked in the transmission channel to the signal (about 10 dB), it results that more power returns to the sender than reaches the receiver. The sender will also receive information at the same time the same information that is sending.

For example, for a -10dBm transmitted signal, the returning signal and the received signal power are:

$$Returning\ Power\ (dBm) = -10dBm - 3dB = -13dBm \quad (30)$$

$$Received\ Power\ (dBm) = -10dBm - 3dB - 3dB - 15dB = -31dBm \quad (31)$$

The fact that the returned signal has higher power than the received signal makes impossible the detection of the real received signal. Therefore, the next situation is presented:

1. The circuit is designed to send enough power to overcome the losses of the channel. Then if there is too much power at reception of transmitter because of the returning signal, the components could be damaged. This must been taken in account when looking for a solution.
2. Connecting only one direction on each transceiver, the transmission is possible.
3. It is not possible to solve this just by increasing the transmission power because of the high losses provoked by the channel. The returning power will always increase at the same time as the received power, but due to these high losses, the returning power on transmitter part will always be higher than the received power on reception, making the discrimination impossible.

Therefore, a way to isolate both transmission and reception channels is needed. To achieve this goal, two possible solutions are offered.

1. **Splitter/Combiner and Software**

In order to isolate both channels, the Wilkinson splitter concept is available. The Wilkinson splitter is a transmission line design to split a signal from one port into two ports keeping these output ports isolated. This design works also in the other direction. It can combine two different signals from the output ports into the common port. The isolation between output ports is theoretically infinite but in real conditions it reaches about 30-40 dB of isolation. This design is mainly for microwave applications (above 300 MHz) because transmission line theory is applied. However, the working frequency is only 100 MHz. Even so, a solution is available. A component (ADP-2-1), which performs this functionality with lumped elements, can be used at lower frequencies. The ADP-2-1 is a surface mount Power Splitter/Combiner, which works in frequencies from 0.5 to 400 MHz [27].

Figure 54: ADP-2-1 electrical schematic

This component splits the received signal in PORT S to PORTS 1 and 2 with an attenuation of 3dB in each one (50% of the signal power to each port). It allows also combining signals from PORT 1 and PORT 2 into PORT S, each with an equal weighing of 50% of the incoming power.

The important feature of this component, which makes it interesting for the application, is the shown isolation between PORT 1 and PORT 2. Using this component, theoretically extra losses of about 25 dB are added to the returned signal to overcome the losses caused by the channel (about 15dB) and allowing the detection. Furthermore, introducing these losses, a possible damage of the receiver part in next stages due to high power is avoided.
In theory, in the final implementation, with a transmitter and a receiver, the signal would decrease about 6dB, 3dB in transmission from PORT 2 to PORT S, and 3dB more in reception from PORT S to PORT 1. Then the expected new power values of the received and returning signals are:

$$Returning\ Power\ (dBm) = -10dBm - 3dB - 25dB = -38dBm \quad (32)$$

$$Received\ Power\ (dBm) = -10dBm - 3dB - 3dB - 15dB = -31dBm \quad (33)$$

With these theoretical results, the received power would be always about 7dB higher than the returning power and thus the detection should be possible just by increasing the transmitted power about 6-7dB.

The layer design to fit the new component shows as follows:

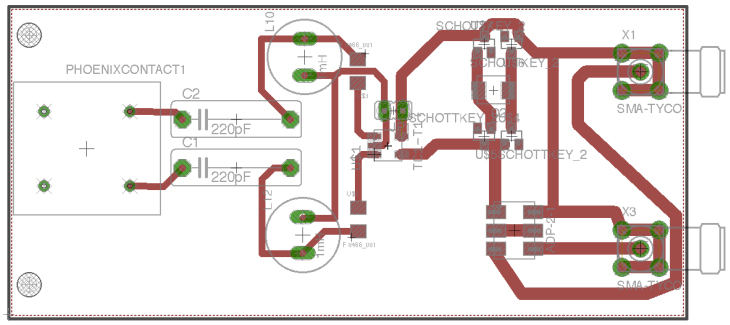

Figure 55: Coupling board with ADP-2-1

Besides of introducing this component and in order to avoid unexpected detections, an additional software block is also designed. This block has the main goal of switching the received signal off when something is transmitted just before the synchronization takes place. Therefore, when something is going to be transmitted, the receiver part of the transmitter will not detect anything.

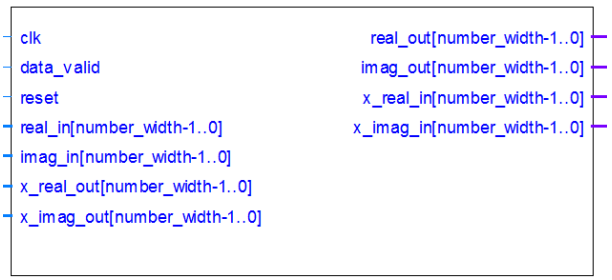

Figure 56: Direction switching block

On the depicted block, the inputs x_real_out and x_imag_out are the connected to the transmitter block, where the transmitted signal arrives from, and the inputs real_in and imag_in are connected to the Analog-to-Digital Converter, where the received signal comes from.

The outputs real_out and imag_out are connected to the Digital-to-Analog Converter, to send the signal, and the outputs x_real_in and x_imag_in are connected to the receiver block, to carry the received signal.

The functions description of the block are:

1. Detecting if there is new data to be transmitted (if data_valid is asserted).
2. If there is no new data, data flow between real_in/imag_in and x_real_in/x_imag_in is established and each clock cycle, data_valid signal is checked.
3. If there is new data, data flow between x_real_out/x_imag_out and real_out/imag_out is established. Meanwhile, any data from real_in/imag_in is discarded and no longer transmitted to the receiver block. Once the data flow is established, data_valid is checked every cycle time (every 31.25 μs).

4. When no more data is incoming, the state machine keeps the data flow to the output until the current cycle is done to be sure that all the information is transmitted.

Although this presented solution seemed a good possible solution, in practice the maximum isolation achieved between ports was only about 14-15 dB, which was not enough and therefore the returned power through the receiver channel was greater than the received power. This was caused by an impedance mismatch at the output port of the ADP-2-1. The ADP-2-1 applications note specifies a drop of the maximum isolation if the output is not properly matched with the next stage. Is specified an isolation of about 6dB in case of a full mismatch and a theoretically unlimited isolation in case of full match.

2. Software Controlled Switch

The second possible solution includes implementing a switch between transmitter and receiver introducing a new component, the MSWA-2-20, and implementing some changes in the software block and in the hardware designed for the previous solution.

The MSWA-2-20 component is a surface mount switch that has one common RF input port and two independent RF outputs and is designed to conduct an RF signal from the common port to one of the two outputs. The output port is selected with two control ports. The aim is to connect two pins of the FPGA in order to control the switch. The switch will substitute the ADP-2-1 to achieve the desired isolation between both ports.

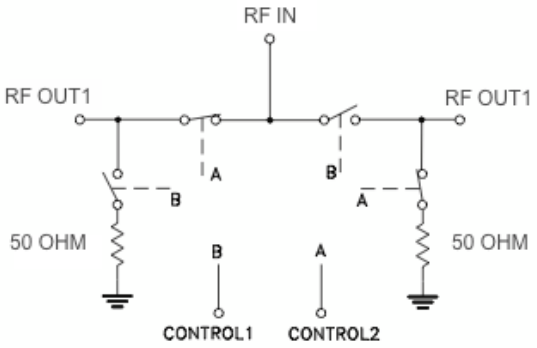

Figure 57: MSWA-2-20 electrical schematic

This component offers a typical isolation between outputs of 60dB @ 100 MHz with typical insertion losses of 0.65dB @ 100 MHz. The MSWA-2-20 is also absorptive, which means that the non-switched port is terminated with 50Ω impedance. This will help to avoid reflections in case of full impedance matching. To select the output, two control signals are needed. The output is selected by introducing a 0V or a –V voltage in the control ports, following the next table:

Control Ports		RF Outputs	
1	2	1	2
0	-V	OFF	ON
-V	0	ON	OFF

Figure 58: Output selection logic

Where –V is a negative voltage between -5 and -8 volts [28].

These voltages are generated on the FPGA with the control block designed in the previous solution in the transceiver program but with a new feature, activate or deactivate the signals that switch both control ports. The signals are mapped to two pins of the GPIO connector available on the Terasic development board. Then, two cables connect these two pins with the designed board. The supplied FPGA voltages are logic voltages, which can be selected between 1.5V, 1.8V, 2.5V or 3.3V. To fit the supplied voltages with the needed voltages of the control ports, an operational amplifier is used on each channel [5].

To choose the proper operational amplifier, the frequency response is a parameter that has a big importance. The control signal must change between logic states fast enough to activate the switch before the modulated signal arrives to it. If an operational amplifier with bad frequency response is chosen, the output signal to the control port will not reach the desired voltage level at the right time and the switch will not be at the right position to transmit or receive.

For example for a LM258 general-purpose amplifier, with a maximum frequency response without high signal degradation of about 10 KHz the signal frequency response to a pulse is [29]

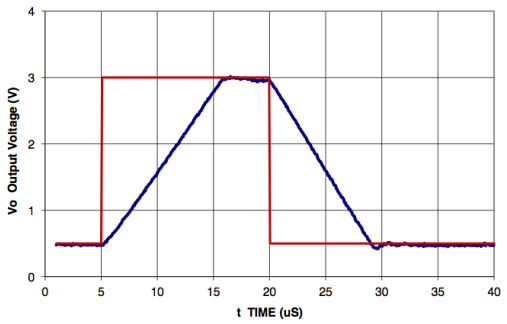

Figure 59: LM258 Pulse Response

It is perceived that the maximum value at the output (blue line) for a given pulse at the input (red line) is reached after 15μs, which is not enough fast for this application (as mentioned before, the cycle time is 31.25μs). Therefore, an operational amplifier with higher frequency response has to be chosen in order to insure that the switch is activated before the signal arrives.

The chosen operational amplifier is the TLC072A. This is a general-purpose operational amplifier, with a better frequency response. The pulse response of this operational amplifier is

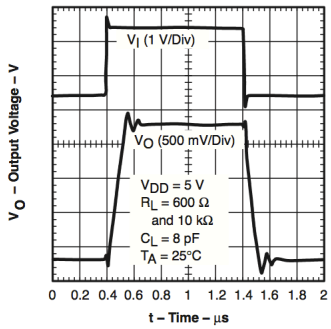

Figure 60: TLC072A Pulse Response

This pulse response is considerably faster and the output will reach the desired values (about -5V) at the right time in order to command the switch before the signal arrives [30]. The time taken by the MSWA-2-20 to switch between channels is about 5 ns, which is negligible.

A new small board is designed to place the new components. This board contains only these two components, connecting the outputs of the operational amplifiers to the control inputs, and the inverted inputs to a cable to connect easily the GPIO port of the Terasic board as described before. The input and output ports of the switch are connected to the high frequency board to receive and send the signal through a common connector.

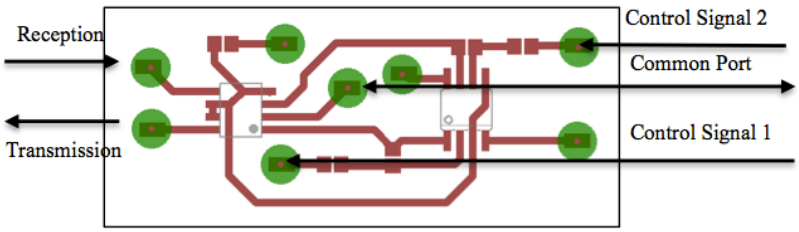

Figure 61: Switching board

The amplifier is connected in inverter configuration. To achieve the required level of -5V, an amplification of about 2 is needed. Following the equation of the amplification for an inverter amplifier configuration (7), the used resistors are:

$$R_1 = 200\ \Omega\,, R_2 = 523\ \Omega \tag{34}$$

The obtained gain will be about 2.615 (~4.1 dB).

The amplifier must be supplied with a symmetrical DC voltage of at least the desired output voltage plus 1.5V. In this case, the regulated supply voltages of the high frequency board, which have approximated values of +8V and -8V, are taken. Then the maximum output can reach maximums between +6.5V and -6.5V. The second case, -6.5V, is enough to control the switch.

For this implementation, the coupling boards are redesigned and now only one common port for receiving and transmitting is needed. Also the ADP-2-1 of the previous design is no longer

needed. Therefore, the board contains a DC connector, the DC blocking capacitance a 1:1.5 balun, the diode bridge, and one SMA connector.

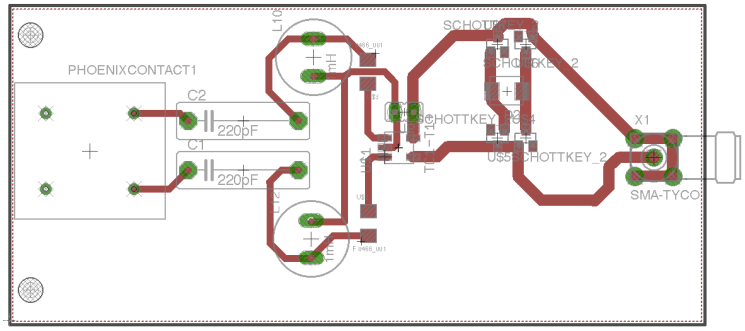

Figure 62: Coupling Board one port

The schematic of the boards' disposition is the following

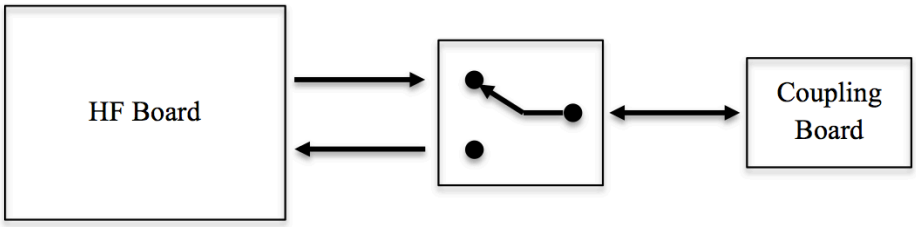

Figure 63: New coupling schematic

The effectiveness of this component is measured on an isolation of about 40 dB between the two ports. The time-domain results of the performance achieved with the use of the described amplifier in addition with the switch are depicted in the next figure:

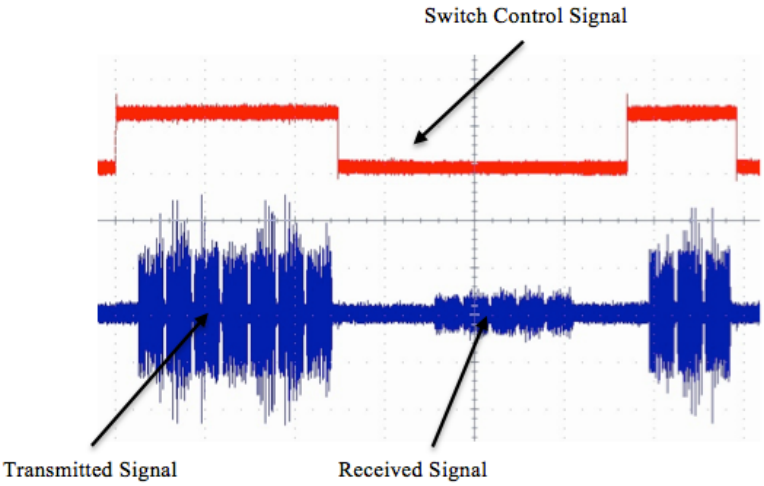

Figure 64: Switch Control signal vs. Data signal

In red color one of the signals that activates the switch. In blue color, the signal of the common port, where are differentiated the incoming and the outgoing signals.

5.4.2.4. Fine Tuning

To achieve the best transmission quality, with the lower bit error rate possible, the transmission power has to be adapted.

For an easier calculation of the power in the different nodes, the Managing Node Modem is placed in between the two Controlled Nodes so it can be assumed that the losses caused on the transmission channel are the same for both Controlled Nodes since the distance are the same. Therefore, similar power transmitted from the Managing Node must arrive into both Controller Nodes and the same amplification is needed on both Controlled Nodes in order to transmit the signal to the Managing Node.

On the design, it is possible to adapt the gain either on the transmitter part or the receiver part of the High Frequency Boards. To this purpose, the designs of the boards are checked in order to achieve the best adaptation possible.

1. Transmitter:

The transmitted power of each board must be adapted for a proper transmission of the signal. The signal transmitted from the Managing Node must arrive properly to both Controlled Nodes. The gain adaptations, in addition to the used topology and the software detection threshold (chapter 3.3.1.10), will help to detect properly the signals transmitted between the Managing Node and the Controlled Nodes. It is possible to raise this threshold to avoid detecting low power signals that can arrive improperly. Since the Managing Node is placed between the Controlled Nodes in the channel, the transmitted signal from one Controlled Node will be detected only on the Managing Node.

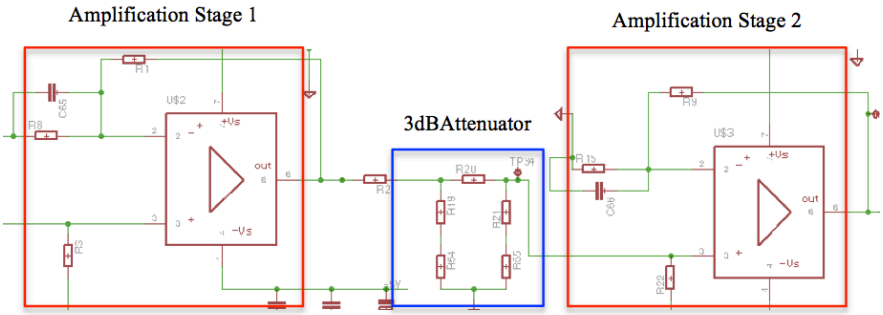

Figure 65: Transmitter stages

After performing some tests and measurements, the final resistors configuration is

R (Ω)	Managing Node	Controlled Node 1	Controlled Node 2
R_1	680 Ω	750 Ω	680 Ω
R_8	330 Ω	220 Ω	300 Ω
R_9	-	750 Ω	-
R_{15}	-	300 Ω	-

Figure 66: Transmitter Amplification stages configuration

Unlike the Managing Node or the Controlled Node 2, whose power after first amplification stage was enough for the transmission, the Controlled Node 1 modulator presented a malfunction, which caused a low output power. Therefore the two amplification stages were needed on this node.

With the given resistors and the formulas 6 (non-inverter configuration) the obtained gains are:

Node	Stage 1 Gain	Stage 2 Gain	Total Gain
Managing Node	4.85 dB	-	4.85 dB
Controlled Node 1	6.44 dB	5.44 dB	8.88 dB
Controlled Node 2	5.14 dB	-	5.14 dB

Figure 67: Transmission gains

2. Receiver:

At reception, the gain adaptation is made to assure the proper power level at the input of the demodulator.

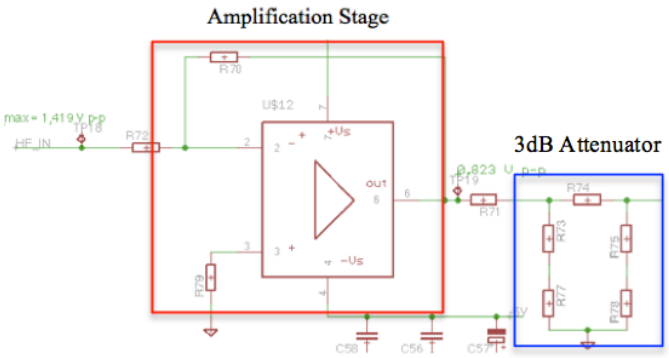

Figure 68: Receiver stages

R (Ω)	Managing Node	Controlled Node 1	Controlled Node 2
R_{70}	680 Ω	680 Ω	680 Ω
R_{72}	221 Ω	330 Ω	330 Ω

Figure 69: Receiver Amplification stage configuration

With the given resistors and 7 (inverter configuration) the obtained gains are

Node	Amp. Stage Gain	Total Gain
Managing Node	4.88 dB	1.88 dB
Controlled Node 1	3.14 dB	0.14 dB
Controlled Node 2	3.14 dB	0.14 dB

Figure 70: Reception gains

With these configurations a perfect communication between the Managing Node and the Controlled Nodes is achieved, with no Bit Error Rate. The power measurements are described on next chapter.

6. Results and Discussions

6.1. Signal-to-Noise Ratio Measurements

The Signal-to-Noise Ratio (SNR) is an essential measurement to test the quality of the transmission channel and the components. In this measurement the ratio between signal power and noise power is given.

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{35}$$

In decibel scale:

$$SNR\ (dB) = 10log_{10}(P_{signal}) - 10log_{10}(P_{noise}) \tag{36}$$

The measured Signal-to-Noise Ratio on output port in transmission can be calculated from the captured spectrum

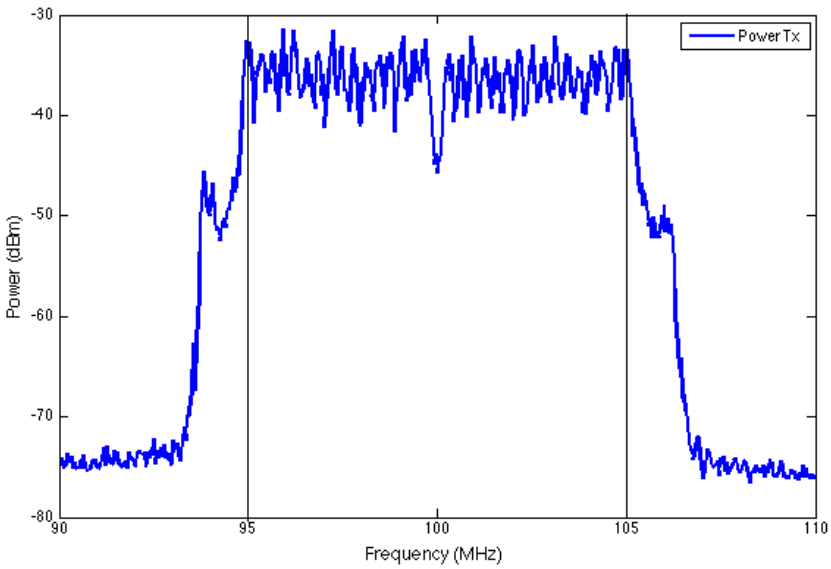

Figure 71: Signal-To-Noise Ratio at transmission

The average signal-power between 95 and 105 MHz is -36.5 dBm, which is the transmitted signal channel. The noise floor can be considered at -74.1 dBm, hence the Signal-to-Noise Ratio is about

$$SNR_{TX}\ (dB) = (-36.5\ dBm) - (-74.1\ dBm) = 37.6\ dB \tag{37}$$

The same measure is made at reception part with the spectrum analyzer with the drive off (red) and with the drive on (green)

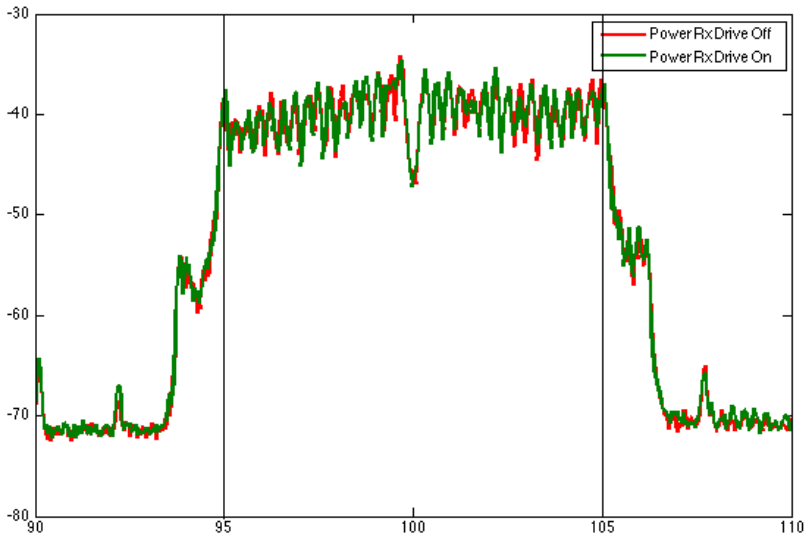

Figure 72: Signal-to-Noise Ratio at reception

The average power in signal channel bandwidth (between 95 MHz and 105 MHz) at reception is -40 dBm and the noise floor in this case is about -72 dBm. Therefore, the SNR at reception is

$$SNR_{RX}\ (dB) = (-40\ dBm) - (-72\ dBm) = 32\ dB \tag{38}$$

Apart from the channel loses there is a modification of the signal shape at frequencies close to 95 MHz, because of the characteristics of the channel. The channel estimation and equalization implemented on the system will help to correct this.

It is noticeable also that there are not significant changes on the signal when the drive is on. Therefore there is no disturbances caused by the drives, and the communication is possible regardless whether the drive is running or not.

6.2. Constellations

The constellation is the graphic representation of the detected digital signals into a map to display the received information.

In the next pictures the transmitted constellation as well as the received constellation are shown in order to have an overview of the effect of the different stages at transmission part (generation, modulation, amplification, filtering), at reception part (filtering, amplification demodulation, detection) and the channel into the signal. In both pictures several samples are taken and represented in the same graph in order to have a realistic plot of the information in time.

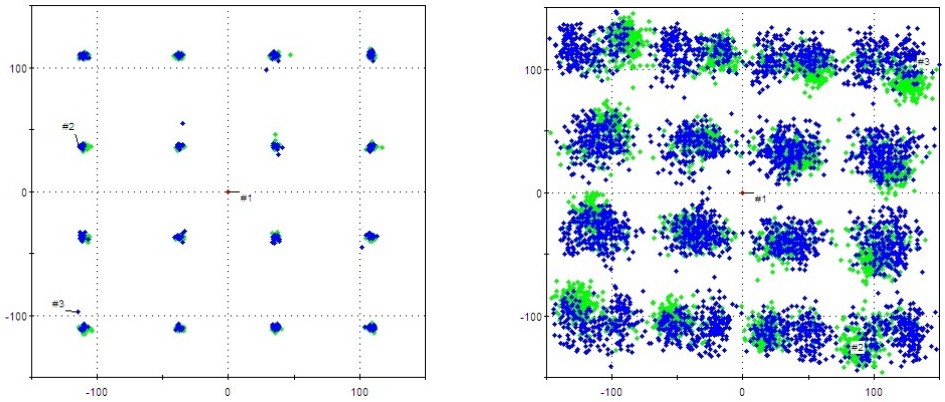

Figure 73: Transmitted Constellation vs. Received Constellation

In the left picture, the transmitted constellation is depicted. The dots are clearly defined as described in chapter 3.3.1.6. The dots are overlapped almost perfectly because no disturbances on the signal are present. In the right picture, the received signal is depicted and now the dots are more scattered, being distinguishable the different received information along the time. The more close are the points of one symbol to the points of another symbol, the more difficult are the symbols to be properly demodulated and recognized, and hence, the higher the probability of Bit Error.

The application of encoding and interleaving techniques are essential to prevent loss of information in noisy channels.

Although these plots do not give quantitative results to measure the goodness or badness of the transmission, they give a good and easy understandable representation of the digital transmission coding and the effects of the channel on the detection.

6.3. Latency Measurements

In real-time communication, time-domain characteristics like latency or jitter are essential to the proper working of the system. A proper estimation of the transmission performance is essential to adapt the system requirements. After the implementation of the Ethernet Coupling (chapter 5 - POWERLINK over Powerline) and due to the characteristics of the modems, several OFDM cycles are needed in order to transmit a complete Ethernet frame. Therefore, two kinds of latency are described:

1. Basic latency: Delay between a single OFDM packet is generated and is detected.
   This latency is due to the several software stages that process the incoming signal, at transmission part as well as at reception part. The stages that introduce more latency are the interleaver/deinterleaver and the IFFT/FFT stages. To measure this latency the described Latency Measurement Block in chapter 5.3.1.1 is used. The resulting latency of a single OFDM packet is

$$T_{delay} = 94.86\ \mu s \tag{39}$$

After the first packet, the following packets arrive with a difference of 31.25 µs (one OFDM cycle) respect to the previous one.

2. Ethernet data latency: Delay between the reception at the Ethernet port of the source modem and the complete packing and generation of the received Ethernet frame at the destination modem. It is important to notice, that the latency introduced in Ethernet frames depends on the size of the Ethernet frame received as described in chapter 5.4.1; the more bytes the Ethernet frame has, the more latency introduces the Modem. Moreover, the stuffing packets to prevent modem synchronization problems are a variable to include in the latency; the more consecutive zeros transmitted, the more stuffing words needed and hence the more latency introduced by the Modem.

To measure the introduced latency on Ethernet frames, an analysis of the Ethernet communication is performed as described in next section. To calculate the expected minimum latency for an Ethernet frame of about 120 bytes (except the Ethernet preamble or the extra stuffing data), first is needed to calculate the one-way transmission minimum delay:

$$N_{cycles} = \frac{120\ \frac{8\ \frac{bits}{byte}}{16\ \frac{bits}{word}}}{18\ words} = 3.333 \rightarrow 4\ cycles \tag{40}$$

$$T_{delay\_eth} = 94.86\ \mu s + (31.25\ \mu s \times 3\ cycles) = 188.61\ \mu s \tag{41}$$

For a two-way communication with 120 bytes sent from each side the minimum response time will be about 377.22 µs. In addition to this time, there is a time of processing and packing the Ethernet data as well as the basic response time of the node (estimated on values between 1 and 3 µs) that will make the final value higher.

6.4. Ethernet Connection Analysis

To analyze the Ethernet POWERLINK communication between the managing node and the different the Wireshark Protocol Analyzer and an Ethernet tap are used. This device allows analyzing the network introducing minimum disturbance or delay on the transmitted data.
Altogether with the Ethernet tap, the Wireshark Protocol Analyzer allows to analyze and show statistics of the communication. Furthermore the export into .csv files functionality is very useful to import the raw information into spreadsheets in order to make calculations and graphs.

77	0.041712000	Bernecke_15:55:26	Broadcast	EPL_V1	76	SoC	dest = 255	src = 0
78	0.041715000	Bernecke_15:55:26	Bernecke_00:49:54	EPL_V1	94	PReq	dest = 84	src = 0
79	0.042277000	Bernecke_00:49:54	Broadcast	EPL_V1	116	PRes	dest = 0	src = 84
80	0.042282000	Bernecke_15:55:26	Bernecke_00:49:55	EPL_V1	94	PReq	dest = 85	src = 0
81	0.042752000	Bernecke_00:49:55	Broadcast	EPL_V1	116	PRes	dest = 0	src = 85
82	0.042757000	Bernecke_15:55:26	Broadcast	EPL_V1	76	EoC	dest = 255	src = 0

Figure 74: Wireshark Capture - POWERLINK cycle

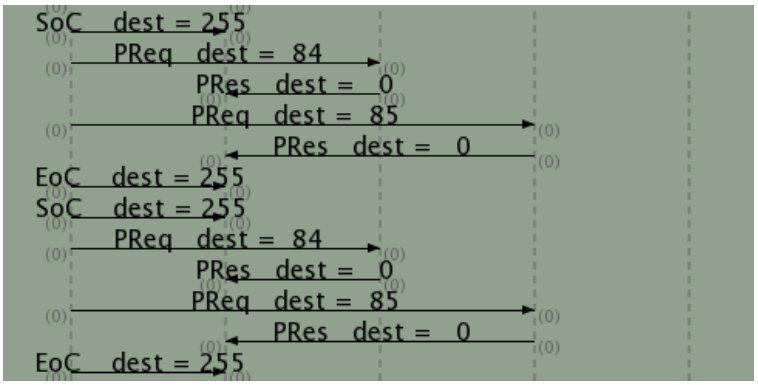

Figure 75: Wireshark Capture - POWERLINK flow graph

On the previous captures, the described Ethernet POWERLINK cycle is depicted. First the Managing Node (Bernecke_15:55:26) sends a Start of Cycle (SoC) frame to everyone on the net (dest=255) and immediately starts polling the different nodes defined on the program, in this case, the addresses 84 and 85 (Bernecke_00:49:54 and Bernecke_00:49:55). Each Controlled Node answers only when is asked, and only the responses received within the predefined response time are valid. After polling the last node, the Managing Node sends the End of Cycle (EoC) frame, which indicates the end of the isochronous cycle and the start of the asynchronous cycle.

The latency measurements are essential to adapt the previously defined Ethernet POWERLINK parameters.

77	0.041712000	Bernecke_15:55:26	Broadcast
78	0.041715000	Bernecke_15:55:26	Bernecke_00:49:54
79	0.042277000	Bernecke_00:49:54	Broadcast
80	0.042282000	Bernecke_15:55:26	Bernecke_00:49:55
81	0.042752000	Bernecke_00:49:55	Broadcast
82	0.042757000	Bernecke_15:55:26	Broadcast

Figure 76: Ethernet POWERLINK analysis

For a PReq to the nodes 84 and 85 is measured a response time of:

$$\Delta t_{84} = t(79) - t(78) = 0.042277 - 0.041715 = 555\ \mu s \quad (42)$$

$$\Delta t_{85} = t(81) - t(80) = 0.042752 - 0.042282 = 470\ \mu s \quad (43)$$

This is a response time for a 94 bytes request from the Managing Node and the 116 bytes response from the Controlled Node asked. The difference of delay between both nodes is due to the Powerline modems. When the Managing Node sends the SoC and the first PReq frames it sends almost sequentially and due to the lower speed of the modem, before finishing sending completely the first frame to the Powerline, the second frame (PReq) arrives and has to be stored until the SoC is completely transmitted.

From each Ethernet frame, some information about the exchanged information can be checked. For example in the next figures the Managing Node polls the Controlled Node 85,

sending the Poll Request (PReq) frame. The Controlled Node answers sending a Poll Response (PRes) back with the bit RD (Ready) set to 1, indicating to the Managing Node that the drive is ready to operate.

```
▽ ETHERNET Powerlink V1.0
    .000 0011 = Service: Poll Request (PReq) (3)
    Destination: 85
    Source: 0
    ..0. .... = MS (Multiplexed Slot): 0
    .... ...1 = RD (Ready): 1
    Poll Size OUT: 32
    OUT Data: a9000080bb450000000001ed0000000000000000000000000...
```

Figure 77: Managing Node Ethernet information

```
▽ ETHERNET Powerlink V1.0
    .000 0100 = Service: Poll Response (PRes) (4)
    Destination: 0
    Source: 85
    ..0. .... = MS (Multiplexed): 0
    ...0 .... = EX (Exception): 0
    .... 0... = RS (Request to Send): 0
    .... .0.. = WA (Warning): 0
    .... ..0. = ER (Error): 0
    .... ...1 = RD (Ready): 1
    Poll Size IN: 76
    IN Data: a90094130000000000001ed200b180023df1200fe7fbb45...
```

Figure 78: Controlled Node Ethernet information

Thanks to the acquired data in the Protocol Analyzer, it is possible to explain the difference of time between the estimated response time and the real Ethernet response time. For example:

For a 94 Bytes request sent by the Managing Node (PReq) to the node 85, the new calculated delay time is:

- Total Ethernet data: Data Bytes + Preamble Bytes = 94 Bytes + 7 Bytes = 101 Bytes.
- Estimation of necessary extra packets to keep synchronization: This estimation can be done looking at the raw data received at the Protocol Analyzer

```
0000 | 00 60 65 00 49 55 00 60  65 15 55 26 3e 3f 03 55
0010 | 00 00 36 00 00 00 00 00  00 00 00 00 00 00 00 00
0020 | 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030 | 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0040 | 00 00 00 00 00 00 00 00  00 00 00 00 00 00 01 01
0050 | 05 10 00 00 01 00 18 c2  de f3 76 04 00 00
```

Figure 79: PReq Frame Data

  It is noticed a big amount of sequential zeros, which will cause the adding of extra packets to avoid the loss of synchronization. For this case the amount of extra words caused by the sequentially transmitted zeros will be approximately 16 words (32 Bytes).
- Total Transmitted data: Total Ethernet data + Extra data for sync. = 101 Bytes + 32 Bytes = 133 Bytes = ~67 OFDM words ➔ 4 OFDM packets.

$$t_{PReq} = 94.86\ \mu s + (\ 3\ packets \times 31.25\ \mu s\ ) = 188.61\ \mu s \qquad (44)$$

For a 116 Bytes response sent by the Controlled Node (PRes) from the node 85, the new calculated delay time is:

- Total Ethernet data: Data Bytes + Preamble Bytes = 116 Bytes + 7 Bytes = 123 Bytes.
- Estimation of necessary extra packets to keep synchronization: This estimation can be depicted from the raw data received at the Protocol Analyzer

```
0000 ff ff ff ff ff ff 00 60  65 00 49 55 3e 3f 04 00
0010 55 00 4c 00 13 5c ff ef  86 01 02 00 00 00 00 00
0020 00 00 01 47 20 00 18 00  59 12 00 00 00 00 00 00
0030 59 12 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0060 00 00 00 00 01 01 05 10  00 00 02 00 c0 bb 0f f4
0070 76 04 00 00
```

Figure 80: PRes Frame Data

  As done with the PReq estimation, the amount of extra words caused by the sequentially transmitted zeros will be approximately again 16 words (32 Bytes).
- Total Transmitted data: Total Ethernet data + Extra data for sync. = 123 Bytes + 32 Bytes = 155 Bytes = ~78 OFDM words ➔ 5 OFDM packets.

$$t_{PRes} = 94.86\ \mu s + (\ 4\ packets \times 31.25\ \mu s\ ) = 219.86\ \mu s \quad (45)$$

The total response time estimated for the transmitted data (without calculating the Ethernet coupling data treatment times) will be:

$$t_{resp} = 219.86\ \mu s\ +\ 188.61\ \mu s\ = 408.47\ \mu s \quad (46)$$

6.4.1. Ethernet POWERLINK Runtime

Since the Powerline modems introduce a considerable delay into the transmitted channel, the Ethernet POWERLINK parameters need be changed in order to be adapted to the new conditions.

The response time calculated before should be introduced on the Automation Studio parameters to allow the communication. Although the minimum measured response time is 555 µs and 470 µs respectively, to give some flexibility and prevent losing Ethernet frames due to eventual higher delays, the introduced Response Timeout will be 1000 µs and 600 µs respectively. The Asynchronous Timeout is also defined at 1000 µs.

Therefore the new Cycle time must be also changed to adapt it to the new conditions. There are two Controlled Nodes with maximum response time of 1000 µs and 600 µs respectively, and eventually can occur an Asynchronous Frame, which has a maximum response time of 1000 µs. Hence the Cycle time is

$$t_{PL_{cycle}} = R_{time1} + R_{time2} + R_{tAsync} = 2600\ \mu s \quad (47)$$

Due to the requirements of the task time, the total Cycle time must be multiple of 400 µs, hence the calculated Cycle time is 2800 µs.

SL1.IF2		5LS189.6-1 (POWERLINK)
Module type	Type 3	Indicates module features
Operating mode	POWERLINK V1	
MTU size	262	
Baud rate	100 MBit half dupl...	
POWERLINK parameters		
Activate POWERLINK communication	on	
Device name	<InterfaceAddress>	
Cycle time [µs]	2800	
Multiplexing prescale	8	
Mode	managing node	
Advanced		
Node definition	get from hardware	
Asynchronous timeout [µs]	1000	
Data transfer restricted to active station	off	
Optimization	data throughput	
PRes frame loss limit	200	Number of lost PRes frames un

Figure 81: Managing Node configuration

SL1.IF2.ST84		8V1016.50-2
POWERLINK parameters		
Mode	controlled node	
Response timeout [µs]	1000	
Multiplexed station	off	
Advanced Optimization		
Optimization	data throughput	
Chained station	off	
Process Data Mapping		
Configuration	Single Axis	Modules operating mode
SL1.IF2.ST85		8V1016.50-2
POWERLINK parameters		
Mode	controlled node	
Response timeout [µs]	600	
Multiplexed station	off	
Advanced Optimization		
Optimization	data throughput	
Chained station	off	
Process Data Mapping		
Configuration	Single Axis	Modules operating mode

Figure 82: Controlled Node configuration

With the described configurations and measurements, the communication between the Managing Node and the Controlled Nodes is possible with low rate of losses. Therefore, it was possible to command of the servo drives, as well as reading the information from the sensors (speed, position, direction etc.).

7. Conclusions and Further Development

The described results and measurements show that the implementation of OFDM transmission technique for Powerline Communication is applicable in industrial communications systems. The data rate and the latency achieved show the viability of the system and open new possibilities with the further development. The interferences produced by the drives are far lower in frequency spectrum, so with the use of the coupling board, the modem and the signal are immune to them.

The basic points that have to be improved are:

- The latency must be lower.
- The latency must be constant to minimize the jitter.
- The achieved data rate must be higher to get closer to the Fast-Ethernet standard.

To achieve these goals, the implementation of Orthogonal Frequencies Division Multiple Access (OFDMA) to assign different subcarriers to different nodes in order to have several transmissions at the same time is the first step. With this implementation, the nodes are permanently connected to the managing node, reducing the latency.

In order to increase the data rate, the implementation of 64QAM digital modulation is also a good point to start. Moreover, the adaptation of the modem cycle time to the application cycle time (in this case Ethernet POWERLINK cycle time) is also an interesting modification to be implemented, transmitting a complete Ethernet frame each modem cycle and reducing therefore latency.

8. Bibliography and References

[1] "Homeplug Alliance - IEEE 1901," 2014. [Online]. Available: http://www.homeplug.org/tech/ieee_1901/.

[2] S. Schmitz, "Industrielle Powerline-Kommunikation für Antriebseinheiten in Werkzeugmaschinen," Stuttgart, 2010.

[3] H. Rohling, OFDM Concepts for Future Communication Systems, Hamburg: Springer, 2011.

[4] V. D. Eraso Garcia, "Commissioning and Enhancement of a Powerline Communication System," 2012.

[5] *DE2_115 User Manual,* Terasic Technologies Inc., 2010.

[6] *Cyclone IV FPGA Device Family Overview,* Altera Corporation, 2014.

[7] *High Speed Mezzanine Card (HSMC) Specification,* Altera Corporation, June 2009.

[8] S. Schmitz, *Dokumentation des OFDM-Modems,* 2009.

[9] *AD9765 12-Bit 125 MSPS Dual Digital-to-Analog Converter Datasheet (Rev. G),* Analog Devices, Inc, 1999-2011.

[10] *AD9248 14-Bit 40 MSPS Dual A/D Converter Datasheet (Rev. B),* Analog Devices, Inc, 2010.

[11] *AD8138 Low Distortion Differential ADC Driver Datasheet (Rev. F),* Analog Devices, Inc, 2006.

[12] *SXLP-4.7+ Surface Mount Low Pass Filter Datasheet (Rev. OR),* Mini-Circuits.

[13] *ADL5385 30 MHz to 2200 MHz Quadrature Modulator Datasheet (Rev. A),* Analog Devices, Inc, 2012.

[14] *ADL5387 30 MHz to 2 GHz Quadrature Demodulator Datasheet (Rev. B),* Analog Devices, Inc, 2013.

[15] *RHP-73+ Metal Shield High Pass Filter Datasheet (Rev. OR),* Mini-Circuits.

[16] *RLP-105+ Metal Shield Low Pass Filter Datasheet (Rev. OR),* Mini-Circuits.

[17] *AD8009 Low Distortion Amplifier Datasheet (Rev. F),* Analog Devices, Inc, 2004.

[18] *True Sinewave Clock Oscillator Datasheet,* Crystek Crystals, April 2008.

[19] *FKP 1 Polypropylene Capacitors for Pulse Applications,* WIMA, 2013.

[20] *Digital Modulation in Communications Systems — An Introduction,* Agilent Technologies, 2001.

[21] *CRC Compiler User Guide,* Altera Corporation, November 2009.

[22] *FFT MegaCore Function User Guide,* Altera Corporation, November 2013.

[23] D. Jones, "Decimation-in-Frequency (DIF) Radix-2 FFT," 17 Sep 2006. [Online]. Available: http://cnx.org/content/m12018/1.6/.

[24] "Ethernet-Powerlink Website," 2014. [Online]. Available: http://www.ethernet-powerlink.org.

[25] *Automation Studio Help,* B&R, 2014.

[26] *IEEE Standards for Local and Metropolitan Area Networks - 802.3u Fast Ethernet,* IEEE , 1995.

[27] *ADP-2-1 Surface Mount Power Splitter/Combiner Datasheet (Rev. D),* Mini-Circuits.

[28] *MSWA-2-20+ Surface Mount Switch Datasheet (Rev. C),* Mini-Circuits.

[29] *LM258 Dual Operational Amplifier Datasheet,* Texas Instruments, May 2013.

[30] *TC072 Low-Noise JFET-Input Operational Amplifier Datasheet,* Texas Instruments, February 2014.

[31] *F5D4079 Surf Powerline AV+ Networking Adapter Product Sheet,* Belkin, 2010.

[32] T. M. Schmidl and D. C. Cox, "Robust Frequency and Timing Synchronization. s.l.," in *IEEE Transactions on communications, VOL. 45, NO. 12*, December 1997.