Faculdade de Engenharia da Universidade do Porto

UWB Using Programmable Logic

Tiago dos Santos Maia Costa

For Jury Evaluation

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor Sérgio Reis Cunha

Second Supervisor: Professor José Carlos dos Santos Alves

January 25, 2016

Resumo

A tecnologia de Banda Ultra-Larga (Ultra-Wideband, ou UWB em inglês) pode servir um grande número de aplicações tais como comunicação rádio bidirecional, Radar, medições de tempos de propagação e distâncias de linhas de transmissão, aplicações biomédicas e captura de imagens através de objetos ou ainda sistemas de radar e anti colisão para automóveis. As aplicações UWB beneficiam de transmissões de baixa potência e são praticamente indetetáveis para os recetores tradicionais de banda estreita, uma vez que os impulsos têm uma Densidade Espectral que cobre uma gama bastante larga de frequências, que consequentemente é confundível com o típico ruído térmico nos recetores comuns. Este trabalho vai demonstrar uma arquitetura para um transmissor e recetor UWB baseado em impulsos, ou IR-UWB (Impulse-Radio Ultra-Wideband), que dispõe da capacidade de realizar medições de tempos de propagação em linhas de transmissão ou oferecer comunicação bidirecional. Também será demonstrado como a arquitetura IR-UWB proposta foi implementada e testada em FPGAs, com o intuito de apresentar duas provas-de-conceito distintas: medir o tempo de propagação de uma linha de transmissão e estimar o seu comprimento, e ainda oferecer a capacidade de comunicação entre dois sistemas com esta arquitetura.

Palavras-chave: IR-UWB, Banda Ultra-Larga, Lógica Programável, FPGA, Comunicação UWB, Medição Velocidade Propagação Electromagnética

Abstract

UWB technology potential applications include high data-rate Radio Transceivers, Radar and Ranging measurements devices, Biomedical and through-wall Imaging or Automotive Collision Avoidance Radar systems. UWB applications benefit from a low-power and virtually undetectable communication channel to Narrowband receivers as UWB pulses have a Power Spectral Density which covers a very large range of frequencies and consequently falls below typical thermal noise levels. This work will demonstrate a simple receiver and transmitter architecture to implement an Impulse-Radio Ultra-Wideband (IR-UWB) communication system on a Programmable Logic Device (FPGA), that can perform ranging measurements and also operate as a transceiver. It will also de demonstrated how this IR-UWB architecture was implemented and tested in FPGA hardware for two proof-of-concept scenarios: measurement of the propagation delay of a transmission line and line distance estimation, and also board-to-board transceiver operation.

Keywords: IR-UWB, Ultra Wide-Band, Programmable Logic, FPGA, UWB Transceiver, UWB Ranging measurements

Acknowledgements

I would like to personally thank everyone hereby mentioned for their continued support over the development of this work:

Supervisor Prof. Sérgio Reis Cunha, for suggesting this dissertation work and always sharing ingenious ideas, solutions and guidance for surpassing many design challenges as the work progressed.

Second supervisor Prof. José Carlos Alves, from who I have learned Verilog HDL coding and FPGA development constraints in his comprehensive programmable logic course.

My parents, for always being loving and supportive of my personal academic endeavours and continuously showing deep comprehension and compassion despite all circumstances and mishaps and never, ever questioning if all would be worth it in the end. For providing me with all the resources and opportunities from an early age, and many times depriving themselves of having a more comfortable life solely for the purpose of always putting myself first and investing in my education. You have always inspired me to pursue my passion for electronics that latter evolved into my most important life goal, an engineering degree in that field.

Diana, my soul-mate, for all the patience, care, affection, support and for continuously making me become a better person.

My entire family, and my auntie Irene in special, who always believed in my ability to pursue an higher education degree and continuously supported the pursuit of my academic goals.

To all, I would like to express my profound gratitude and consideration for supporting the development of my academic pathway, by making this dissertation possible and by inspiring me to become the person that I am today. Against unfavourable odds and a few setbacks, together we have overcome them all and proved that not only it was all worth it, but also that adversity only becomes defeat when one stops trying. This achievement is as much as yours as it is mine. Thank You.

Tiago dos Santos Maia Costa

“Don’t turn it on, take it apart!”

David L. Jones - EEVBLOG.COM

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analogue-to-Digital Converter
AM	Amplitude Modulation
ASIC	Application-Specific Integrated Circuit
BER	Bit Error Rate
BoM	Bill of Materials
BPF	Band-Pass Filter
BPSK	Binary Phase-Shift Keying
CDMA	Code Division Multiple Access
CMOS	Complementary Metal-Oxide-Semiconductor
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
CTBV	Continuous-Time Binary-Value
DDR	Double Data Rate
DPPM	Differential Pulse-Position Modulation
DS-CDMA	Direct Sequence Code Division Multiple Access
DS-UWB	Direct Sequence Ultra-WideBand
DSO	Digital Storage Oscilloscope
DUT	Device Under Test
EIRP	Equivalent Isotropically Radiated Power
EMI	Electromagnetic Interference
ETSI	European Telecommunications Standards Institute
FCC	(United States of America) Federal Communications Commission
FM	Frequency Modulation
FPGA	Field-Programmable Gate Array
FSM	Finite-State Machine
HDL	Hardware Description Language
HDMI	High-Definition Multimedia Interface
I/O	Input/Output
IC	Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Intellectual Property
IR-UWB	Impulse-Radio Ultra-WideBand
ISI	Inter-symbol Interference
ITU-R	International Telecommunication Union Radiocommunication Sector
LDO	Low-Dropout Regulator
LED	Light Emitting Diode
LFSR	Linear-Feedback Shift-Register

LNA	Low-Noise Amplifier
LSB	Least Significant Bit
LUT	Look Up Table
LVDS	Low-voltage Differential Signalling
MB-OFDM	Multi-Band Orthogonal Frequency-Division Multiplexing
MEMS	MicroElectroMechanical Systems
MSB	Most Significant Bit
M-UWB	Multi-carrier Ultra-WideBand
OOK	On-Off Keying
OPM	Orthogonal Pulse Modulation
OSI model	Open Systems Interconnection model
PAM	Pulse-Amplitude Modulation
PAN	Personal Area Network
PCB	Printed Circuit Board
PLL	Phase-Locked Loop
PM	Phase Modulation
PN	Pseudorandom Noise
PSM	Pulse-Shape Modulation
PPM	Pulse-Position Modulation
PWM	Pulse-Width Modulation
RF	Radio Frequency
RFID	Radio-Frequency IDentification
RTL	Register-Transfer Level
RTS/CTS	Request to Send / Clear to Send
RTT	Round Trip Time
RX	Receiver
SAR	Synthetic-Aperture Radar
SNR	Signal-to-Noise Ratio
SoC	System-on-a-Chip
ToA	Time of Arrival
ToF	Time of Flight
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UWB	Ultra-WideBand
VF	Velocity Factor
VSWR	Voltage Standing Wave Ratio

Chapter 1

Introduction

In the beginnings of wireless communication, during the 19th century, radio was primarily pulse-based and inherently Wideband. Everything changed with the ban on spark-gap based transmitters that caused interference on medium-wave radios and other receivers. Nevertheless, no practical functionality was really lost with the new rules as the channel capacity of those large RF bandwidths was not being harnessed by the spark-based radio devices of that age. In fact, Wideband utilization was more a result of the technological limitations of the Marconi transmitter era than a desired outcome [1]. Later, with the 20th century came the democratisation of radio when Narrowband and tuned radio communications allowed an efficient and well defined utilization of the ever more crowded RF spectrum, as mandated by the governmental regulatory agencies in most regions of the globe.

Only as of 2002 did the U.S. Federal Communications Commission (FCC) approved a rather conservative usage of some parts of the spectrum for Ultra-WideBand (UWB) communications, such as the 3.1 GHz to 10.6 GHz range, with limited Power Spectral Density emission (maximum of $-41.3\,\mathrm{dBm/MHz}$ *EIRP* [2], other restrictions apply), believed to not cause interference on conventional Narrowband transmissions in the same frequency band [3], but which limit UWB usage on other than short-range applications. The European Telecommunications Standards Institute (ETSI) approved similar rules for the European region [4].

The advantages of UWB radio have been long known: the ability to propagate signals through obstacles, the centimetre level ranging capability as a result of the nanosecond resolution and the high data-rates and large channel capacities [2]. Yet, due to regulatory limitations, the development and usage of UWB applications were simply out of reach for the masses. Nowadays, after the FCC liberation and following a large period in time of mostly Narrowband radio communications, a significant interest in UWB has emerged and UWB technology is currently being used and researched for a vast number of civilian, industrial and military applications [3]. Such application possibilities include biomedical and non-medical Imaging Devices [6, 7, 8] such as through-wall and ground mapping, short-range high bitrate Radio Transceivers [9, 10], centimetre-precision

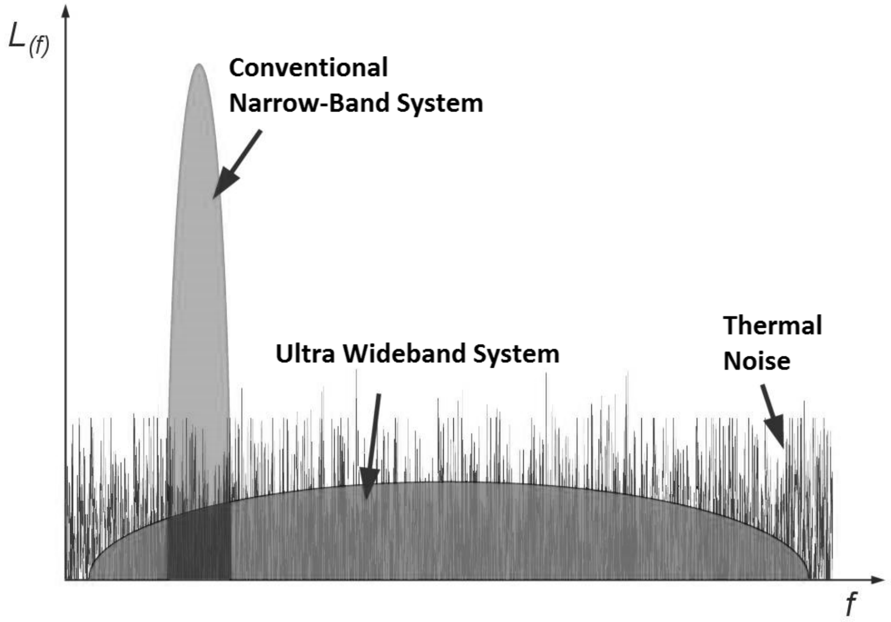


Figure 1.1: UWB vs. Narrowband relative signal levels over a frequency spectrum (Image source: [5]).

Radar Detection & Measurement Systems [11, 12] and Automotive Radar Collision Avoidance implementations [13]. UWB-based communications tend to not interfere with Narrowband or even other UWB communications due to the narrow-pulse high-bandwidth transmissions and its averaging effect on the Power Spectral Density of UWB signals over the RF spectrum. The transmission of symbols by making use of pseudo-random codes further helps to obfuscate these type of signals from traditional Narrowband receivers. Examples include Maximum Length sequences, Gold sequences, Kasami sequences or Barker sequences. Therefore, UWB systems are suitable for military purposes, as the radiated energy spreads over a large bandwidth and the power of the transmissions falls bellow the typical thermal noise floor observed in the environment, making UWB-based radars harder to detect in the field when compared with Narrowband radars. In medical applications, UWB might help eliminate patient exposure to potentially dangerous side-effects associated with diagnostic equipment still relying on ionizing radiation for producing human body imagery[6].

For these and other reasons, UWB is an exciting and promising technology for the 21st century with a large range of applications already under its scope. On the other hand, modern programmable logic has knowingly become ever more powerful and inexpensive as a side effect of the popular Moore's Law. As such, Field-Programmable Gate Arrays (FPGAs) are an attractive development platform for UWB systems due to their reconfigurable and reprogrammable nature. Yet, a purely digital and FPGA-based UWB implementation is still a challenge as of today because of the associated narrow-pulse timing constraints. But performing signal processing in the digital domain is advantageous in order to lower the complexity of RF front-end circuits. Using appropriate techniques, the processing may take place almost entirely on the digital domain and this work will propose and demonstrate some examples on how to design Impulse-Radio Ultra-WideBand (IR-UWB) systems for FPGAs, which can operate on baseband and do not require complex analogue IF filtering as the signals are sine-wave free as predicted by [2]. Also, more and more academic research is focussing on UWB solutions relying at some point on programmable logic so this appears to be an appropriate direction to take when developing UWB technology. Even though the solutions to be presented in this work might not have immediate commercial outcome in the foreseeable future, in part due to limitations of the low-budget hardware that was used, the design concepts could be further deployed into more capable hardware by the researchers of tomorrow as FPGA hardware gets increasingly better performance and economic viability.

1.1 Objectives, Motivation & Approach

The goal of this dissertation work is to design, develop and test an Ultra-WideBand (UWB) system based on programmable logic, in particular, an FPGA. The result will be a short range IR-UWB communications device that can perform ranging measurements and also operate as a transceiver.

From the beginning it was accepted that it would be difficult to offer the same level of high bandwidth and data-rates usually associated with UWB on a newly developed architecture to be applied in low-cost FPGAs instead the typical CMOS Application Specific Integrated Circuits (ASICs). Nevertheless, the main purpose of the work would be satisfied with the delivery of a proof-of-concept design that could demonstrate UWB concepts on FPGA hardware for both intended applications.

Motivation

The main motivation for the work was the challenge of developing a lower-cost and non-ASIC alternative to currently existing custom CMOS designs of UWB applications, ideally offering tens of Mbit/s performance for short-range communications, and also ranging measurements in the range of tens of centimetres. This would be made possible by designing an UWB architecture in Hardware Description Language (HDL), both for the receiver and transmitter chains, that would be compatible with common FPGA chips, as opposed to ASIC-based systems witch are typically more expensive to develop and deploy.

Approach

The solution came to be based on IR-UWB and Continuous-Time Binary-Value (CTBV) signal processing, as described per [14, 15], in which short-duration impulses in the *nanosecond* or *picosecond* range translate into the utilisation of a large portion of radio spectrum. This is a lower data-rate solution but also lower-power and less complex, therefore consuming fewer logic resources, as opposed to Multi-carrier Ultra-WideBand (M-UWB) options such as Multi-Band Orthogonal Frequency-Division Multiplexing (MB-OFDM) implementations.

Because the work also focused on obtaining data from testing the design on real hardware, one of the main challenges of the project was managing the variability of propagation delays between logic regions and working with the available speed ratings of the programmable gates, limitations even more relevant on low-budget FPGAs which have operational clock speeds in the range of a few hundreds of MHz. In order to meet the required functionalities of the proof-of-concept applications and accomplishing the goals of signal generation, detection and analysis, while pursuing the speeds and bandwidths as per the UWB definition, no further work was done to implement the system in other than wired propagation mediums. The pursue of a wireless UWB application was dropped as it would bring no additional validation to the design concept and could potentially undermine the development of the transceiver and wave reflection detection functionalities, two relevant proof-of-concept demonstrations later described in the following chapters.

1.2 Document Structure

Following this introductory 1st chapter, the report contains five additional chapters.

- **Chapter 1: Introduction**;
- **Chapter 2: Theory & Literature Review**, presents a Theoretical Background and a Literature Review on current UWB research and its applications;
- **Chapter 3: System Architecture**, introduces the architecture of the developed system and its possible applications;
- **Chapter 4: System Implementation on Programmable Logic Hardware**, demonstrates an FPGA implementation for the proof-of-concept demonstrations;
- **Chapter 5: Tests & Results**, contains relevant specifications of the hardware implementation;
- **Chapter 6: Final Remarks & Future Work**, summarizes the work done and suggests some potential Future Works to be developed from this dissertation.

1.3 Website & Code Repository

All related documentation, including this dissertation report, slide presentation of the project, software code and final results are available at the website `http://www.uwb-radio.com` for reference.

Chapter 2

Theory & Literature Review

Introduction

In this chapter it will be established a theoretical framework for understanding the work done and a brief description regarding the current state-of-the-art of different types of UWB applications. The literature review will offer a brief perspective about the current possibilities associated with this technology. Extended details will be given whenever the research subjects can relate significantly with the goals of this implementation, but other works considered less relevant will be only concisely referenced.

Although the goal of the dissertation work is to develop an IR-UWB system based on Continuous-Time Binary-Value (CTBV) signal processing, as described in [14, 15], it is still important to understand what other approaches are currently available that may provide similar functionality. For this literature evaluation, a couple hundred articles and papers mentioning UWB were accessed from the IEEE Xplore® Digital Library and other sources. However, only a limited few, which were considered the most relevant to the work, will be referenced in the following sections of this chapter. It's important to also note that the available studies on UWB technology mostly fall under short-range application scenarios. This is usually imposed by regulatory agencies such as the FCC [3] and ETSI [4] whose rules on UWB limit the power of UWB transmissions.

Following this introduction, the information in this chapter is divided into six major sections:

- **Section 2.1: UWB definition**, defines UWB technology and related concepts, including a comparison between UWB and traditional Narrowband radio;
- **Section 2.2: Single-Band or Impulse-based (IR-UWB)**,introduces IR-UWB and associated modulation schemes, medium access control techniques, signal detection and sequence despreading approaches;
- **Section 2.3: Multi-Band UWB**, explains the different aspects of Multi-Band UWB, when compared to Single-Band or IR-UWB;
- **Section 2.4: IR-UWB Applications**, focuses on literature mentioning general IR-UWB designs and its applications;

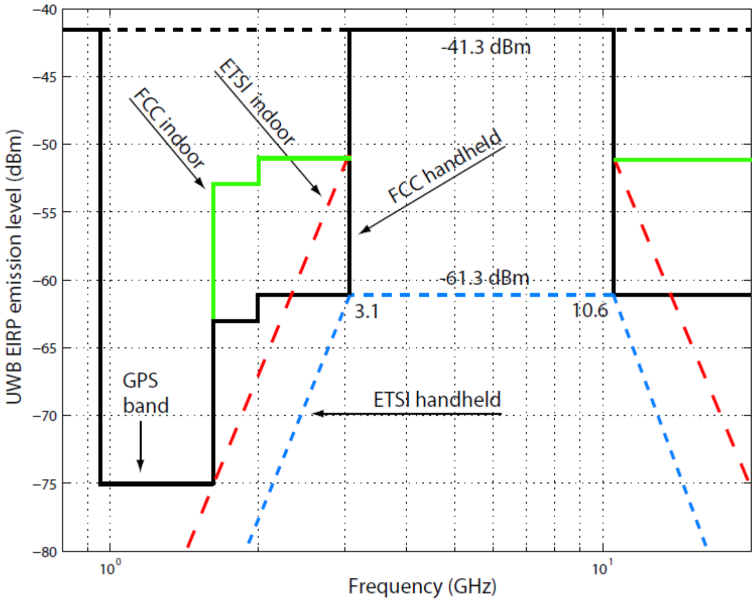


Figure 2.1: UWB power limitations regarding transmission regulations over the authorized frequency bands, as defined by the FCC and ETSI (Image source: [16]).

- **Section 2.5: FPGA implementations of UWB systems**, introduces examples of UWB implementations with FPGAs;

- **Section 2.6: Conclusions**, draws some conclusions about the state-of-the-art analysis and the potential of future IR-UWB applications in programmable logic.

2.1 UWB definition

UWB was initially known as pulse-radio. Currently, the FCC and ITU-R define UWB signals in terms of its bandwidth when transmitted from an UWB antenna. Signals with an absolute bandwidth greater than 500 MHz or which fractional bandwidth ratio is at least 0.20 are considered Ultra-WideBand.[3] The fractional bandwidth of a baseband UWB signal is then defined as the ratio:

$$Fractional\ BandWidth: \qquad f_{BW} = \frac{2 \cdot (f_H - f_L)}{f_H + f_L} \tag{2.1}$$

But it can also be defined as the ratio between the signal's bandwidth and its central frequency, $f_c = (f_H + f_L)/2$, leading to:

$$Fractional\ BandWidth: \qquad f_{BW} = \frac{(f_H - f_L)}{f_c} \tag{2.2}$$

In both equations, f_H and f_L are defined as the upper and lower boundaries of the spectrum where the transmitted signal crosses the -10 dB mark in relation to the power of the highest radiated peak (0 dB).

UWB vs Narrowband

Conventional and more common Narrowband broadcasts typically transmit information by varying the amplitude (AM), frequency (FM) and/or phase (PM) of a higher frequency carrier-wave, which is modulated based on a lower frequency baseband signal that holds the information to be transmitted. This ensures the signal is mainly concentrated within a narrow bandwidth around a central frequency that is the frequency of the carrier-wave. The modulated carrier-wave is them amplified by a Power Amplifier circuit and continuously fed to an antenna.

UWB differs from Narrowband radio because information is transmitted through narrow pulses within *nanosecond*-range time instants, resulting in a large bandwidth usage that typically exceeds 500 MHz. UWB can be carrier-free or carrier-based and there are different ways of generating UWB signals as per the definition of regulatory agencies: Impulse-Radio Ultra-WideBand (IR-UWB) was the first recognised method but there are others such as Muti-Band Orthogonal Frequency Division Multiplexing (MB-OFDM) and Frequency Modulation Ultra-Wideband (FM-UWB). This work explores a variant of IR-UWB that relies on Continuous-Time Binary-Value (CTBV) signal processing, later described in additional detail.

Due to the nature of how UWB works and considering *EIRP* limits, the pulse generator circuit can be directly coupled to antennas. This eliminates the need of using mixers or Power Amplifiers, therefore limiting the power consumption of the UWB transmitter. Likewise, the UWB receiver circuit can be as simple as an analogue correlator or a 1-bit ADC detector. As FPGA technology advances, FPGAs become an ever more attracting development platform for building UWB embedded pulse generator circuits and detectors without requiring complex external RF circuitry.

As a direct result of Information Theory and the Shannon–Hartley theorem, in which an increase in bandwidth enables higher channel capacity under the same SNR, UWB allows the deployment of high capacity and high bitrate communication systems particularly at relative short distances. This advantage could perhaps meet specific consumer or industrial level requisites regarding Personal Area Networks (PAN) with high bitrate requirements, as well as low-power specifications in distributed networks of sensors for the Internet of Things (IoT) or industrial applications [2].

$$Channel\ Capacity: \qquad C = B \cdot log_2(1+SNR) \qquad \text{(bit/s)} \tag{2.3}$$

2.2 Single-Band or Impulse-based (IR-UWB)

IR-UWB signals can be carrier-free (baseband) or single-carrier and consist of a sequence of short pulses typically under 2 ns. These narrow-width impulses in the time domain take a large frequency spectrum and therefore their equivalent Spectral Power signatures resemble the Power

Spectral Density of thermal noise to more commonly used Narrowband receivers. [3, 17]. Depending on the application and modulation scheme used, the rate of pulse repetition can vary between several orders of magnitude: in UWB data communication streams this rate is higher than what's usually found on UWB radar systems.[17]

Meeting UWB specifications

In order for a system to meet UWB specifications, referred in 2.1, and be accepted as Ultra-Wide Band, IR-UWB signals can be up-converted from their baseband frequencies up to the fractional bandwidth limit specification, even if the baseband pulses do not comply with the minimum 500 MHz bandwidth specification. Two viable alternatives to generate complying pulses: the *Switch-based* or *Up-conversion methods*, are described in [18].

- **Switch-based** In the Switch-based solution a UWB transmitter directly switches the pulse generator oscillator on and off. Because UWB pulses are very narrow in time, this is the most power efficient approach as the oscillator is only turned on for very short periods in time.

- **Up-conversion** The Up-conversion method is the most versatile of the two as the baseband signal from the transmitter is up-converted through a mixer to an upper central frequency that meets the fractional bandwidth requirement in cite equation 2.2. The central frequency can be as high as this fractional bandwidth ratio allows as long as the pulses fall within the *EIRP* mask of applicable regulatory limits.

2.2.1 Modulation Schemes

According to [17, 19], predominant modulation schemes for IR-UWB are Pulse-Amplitude Modulation (PAM), On-Off Keying (OOK), Pulse-Shape Modulation (PSM), Pulse-Position Modulation (PPM), Pulse-Width Modulation (PWM), Binary Phase-Shift Keying (BPSK) and others. Although this work is based on OOK modulation, a brief description of other types is given below in order to illustrate the differences between other IR-UWB systems. Each variant has its own advantages in transceiver complexity, multipath interference, crosstalk immunity and spectral efficiency, but its outside the scope of this work to discuss all the associated pros and cons. It shall also be noted that typical IR-UWB transmissions are made of n-derivative Gaussian pulses which equations are 2.4 and 2.5 as per [20]. These pulses are modulated accordingly to the chosen scheme as illustrated in Figure 2.2.

- **Pulse-Amplitude Modulation**

  In binary PAM, the binary signal is transmitted with two antipodal pulses. For instance, if a given Gaussian n-derivative pulse represents the binary *1*, its corresponding antipodal representation is used to transmit the binary *0*.

- **On-Off Keying**

  OOK is the simplest modulation scheme available for modulating UWB signals. One of the binary symbols is represented by transmitting a pulse, where the complementary binary symbol is represented by the absence of a pulse. This work makes use of OOK modulation for transmitting IR-UWB signals by toggling the state of FPGA output pins. External pulse-shaping circuitry could still be used to re-shape this signal to further resemble Gaussian pulses, improving transmission characteristics and minimizing group delay.

- **Pulse-Shape Modulation**

  PSM works by assigning a unique pulse for each transmitted binary symbol, hence preventing signal autocorrelation between symbols. Ideally, the pulses are orthogonal as in Orthogonal Pulse Modulation (OPM). In multiple access conditions, this modulation can also help eliminate cross-correlation between concurrent transmissions as each symbol of different channels can be assigned its unique pulse-shape.

- **Pulse-Position Modulation**

  In PPM each symbol is represented by a pulse in a precise time slot of a UWB transmission. The information is therefore contained in the precise time of arrival of the pulses at the receiver's end. As the transmitter's clock information is needed to correctly decode the transmissions, a synchronization reference can be obtained with a shared or common clock between transmitter and receiver. Frequently, the synchronization problem can be solved in a differential mode, with Differential Pulse-Position Modulation (DPPM), by encoding the information relatively to the last transmitted symbol.

- **Pulse-Width Modulation**

  PWM symbols carry information in the width of UWB pulses - unique pulse-widths in the time domain are reserved for each symbol. Clock information can be also be encoded in the PWM signal, which period can match the width of the shortest symbol's pulse-width.

- **Binary Phase-Shift Keying**

  BPSK modulation can be generated by continuously transmitting a pulse which phase shifts when sending the opposite binary symbol.

$$1^{\text{st}} - order\ Gaussian\ derivative: \qquad x^1(t) = \frac{A \cdot t}{\sqrt{2\pi} \cdot \sigma^3} \cdot e^{-\frac{t^2}{2\sigma^2}} \tag{2.4}$$

$$n^{\text{th}} - order\ Gaussian\ derivative: \qquad x^n(t) = \frac{d^n}{dt^n}\left(\frac{A \cdot t}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{t^2}{2\sigma^2}}\right) \tag{2.5}$$

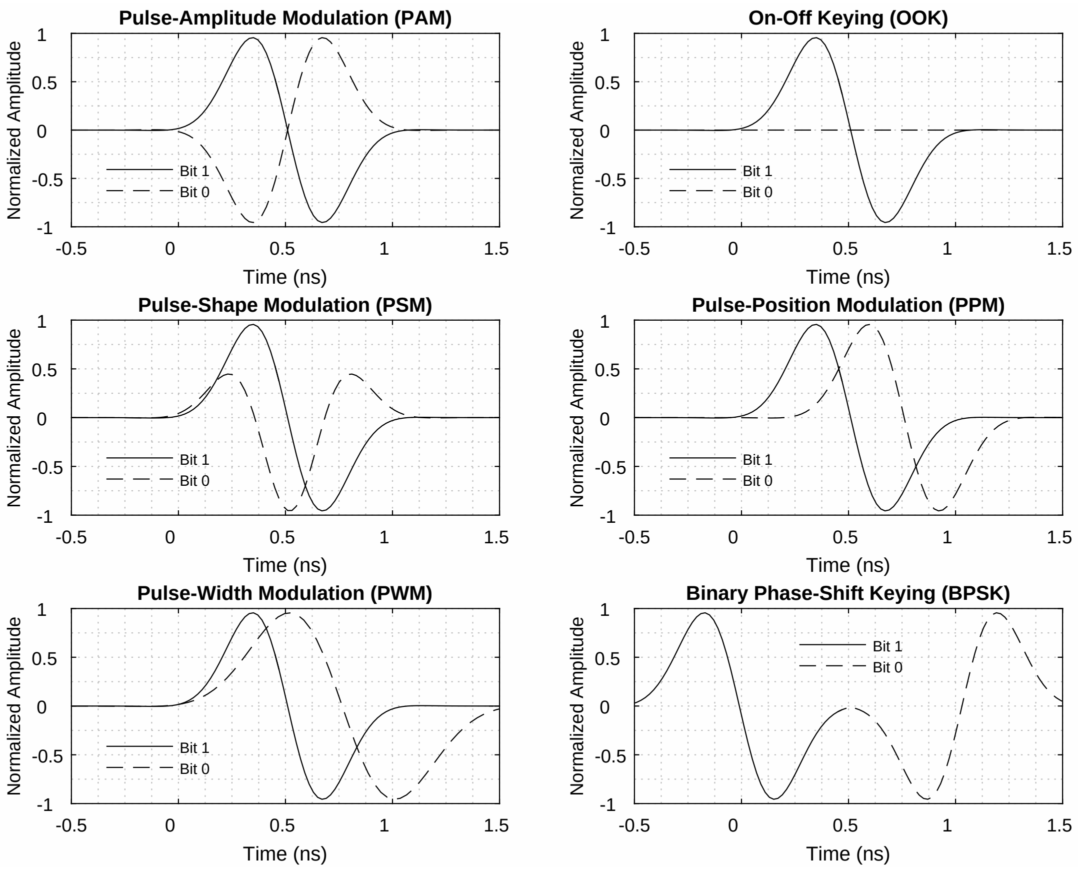


Figure 2.2: Common IR-UWB Modulation schemes according to [17, 19]. These IR-UWB pulses were generated with 1st order derivative Gaussian pulses, except bit 0 of PSM which is a 2nd order derivative pulse.

$$where: \qquad \sigma = \frac{\tau_{pulse}}{2\pi} \qquad and\ \tau_{pulse}\ the\ pulse\ length. \quad (\mathrm{s}) \tag{2.6}$$

2.2.2 Medium Access Control

Impulse-based UWB occupies a single-band, wide enough to comply with the UWB bandwidth definitions in 2.1. There is no band isolation between different IR-UWB transmitters using the same portion of the UWB spectrum. Whether impulses are transmitted in their original base-band or through an up-conversion carrier wave, medium access control techniques can be used to provide multiple access capability and prevent RF interference with other systems nearby.

Such techniques, described in [19], typically consist in randomizing IR-UWB signals in a way that causes the transmissions to appear like RF noise to third-party receivers. This can be achieved in at least two different ways: using a *Time-Hopping* (TH) or *Direct-Sequence* (DS) approach, although not all modulations work with both techniques. For Medium Access Control in Single-band UWB applications, Time-Hopping or Direct-Sequences can enable band sharing without enforcing RF channel isolation as it is usually the case with Multi-band UWB. Multi-band applications can also concentrate these techniques into their sub-bands for additional medium sharing capabilities.

- **Time-Hopping (TH)**

  TH varies the position of transmitted pulses in relation to a known PN code. In this technique, multiple transmissions are allowed if each user has their unique PN code. Every code specifies the position in time of the multiple pulses that spread a binary bit of data payload. PAM, PPM, PWM and PSM are suitable modulations for TH scenarios.

- **Direct-Sequence (DS)**

  DS uses different PN codes per modulated binary bit and per channel. The PN codes are converted to several modulated pulses, therefore forming a unique *chip* of suitable length. Simultaneous transmissions can be achieved if the receiver is capable of despreading the channel by matching the received signal with a range of known codes. Accepted modulations for DS are PAM, OOK and PSM, but not PPM as this modulation is the most simple form of TH. This work will focus on using a DS similar approach through making use of different PN codes that translate into *chips* (sequences of OOK pulses) to transmit a binary data payload of *ones* and *zeros* in an IR-UWB channel.

PN Sequences for DS IR-UWB

Pseudo-random Noise Sequences can be used to spread the binary data payload over the entire available bandwidth of the transmission channel. In an IR-UWB system, the transmitter generates pulses that match a predetermined PN sequence. Although the signals resemble white noise to common Narrowband transmitters, they are deterministic and can be despread and decoded by DS IR-UWB receivers. Examples of PN sequences are: Maximum Length Sequences (m-sequences), Gold codes, Kasami codes, Barker codes or Walsh–Hadamard codes.

The characteristics of UWB pulses limit the effects of multipath propagation like fading and interference, particularly if adequate PN codes and Rake Receiver techniques are used - both were included in the design of the IR-UWB architecture for this dissertation. Yet, UWB signals are still vulnerable to inter-symbol interference, autocorrelation and cross-correlation phenomena - the choice of ideal PN codes is key to obtaining the best performance out of a DS system.

Gold Codes

Gold codes are a set of PN sequences that are very effective in enabling multiple transmitter operation over the same channel. Sets of Gold codes with $2+2^n-1$ sequences of odd length $N = 2^n - 1$ can be generated by time-shifting *Preferred-Pairs* of m-sequences and using XOR operations between them. The resulting codes have predictable, bounded and lower peak cross-correlations than m-sequences within a particular set, which is an advantage in a multi-user scenario. About half of the codes in a set are balanced, meaning that the number of *zeros* and *ones* differ by one, enforcing the white noise spectral signature of the transmissions. Even though Gold sequences show slightly worse autocorrelation properties than m-sequences, their corresponding autocorrelations are not entirely problematic.

The usage of Gold Codes in favour of other variants of PN sequences was decided in agreement with Supervisor Professor Sérgio Reis Cunha, as these codes are commonly used in CDMA systems and GPS navigation and are well documented in literature. Many resources are available on-line to facilitate the understanding of the algorithms that can generate such codes.

Kasami Codes

Kasami codes are another kind of PN sequences with good cross-correlation properties. Kasami sequences are divided into two sets: the small set and the large set, but only the former offers the best correlation properties as they approach Welch's lower bound. However, despite the small set performing better than Gold codes, there are only $2^{n/2}$ sequences in the small set. As for the larger set, it not only includes the small set but also a set of Gold codes, for $mod(n,4) = 2$, and additional Kasami sequences. Kasami codes also have a length of $N = 2^n - 1$, but contrary to Gold codes, n must be even. Nevertheless, Kasami sets can be obtained using shift operations similar to those used for generating Gold codes.

- **Steps for generating *Preferred-Pairs* of m-sequences ([21, 22, 23])**

  1. Obtain a m-sequence, u, of length $N = 2^n - 1$ with a maximum-length Linear-Feedback Shift-Register (LFSR) of n shift-registers. A LFSR is a shift-register in which its input bit is a linear function of its previous state and can be implemented with cascading exclusive-or (XOR) operations, that can be represented by a feedback polynomial. A maximum-length LFSR has a primitive feedback polynomial and can go through all the possible states, or $2^n - 1$ states, except one (all bits equal to zero).
  2. Create a 2^{nd} m-sequence, v, by applying decimation by a factor of q to the original m-sequence. Decimation effectively reduces high-frequency components of a signal and performs down-sampling, keeping only every q^{th} sample while preventing aliasing. These m-sequences are considered *Preferred-Pairs* if the following conditions are simultaneously met:

     (a) n is odd, or $mod(n,4) = 2$, meaning n can't be divisible by 4

     (b) q is odd and, for a given integer k, $q = 2^k + 1$ or $q = 2^{2k} - 2k + 1$

     (c) n is odd and $gcd(n,k) = 1$, or $mod(n,4) = 2$ and $gcd(n,k) = 2$

     where:

     $mod()$: *is the modulo operation;*

     $gcd()$: *is the greatest common divider.*

- **Steps for generating Gold Codes ([21, 22])**

1. Obtain a *Preferred-Pair* of m-sequences,(u,v) of period $N = 2^n - 1$.
2. A set of $N+2$ Gold codes is defined by:

$$G(u,v) \quad = \quad \left\{u,\ v,\ u \oplus v,\ u \oplus Tv,\ u \oplus T^2 v,\ \ldots,\ u \oplus T^{N-1} v\right\} \tag{2.7}$$

where:

$\oplus$: *is the modulo 2 sum operator (XOR operator);*
T^{N-1} : *is a cyclic left shift operator by $N-1$ shifts.*

- **Steps for generating Kasami Codes ([24])**

1. Given a binary sequence u of period $N = 2^n - 1$, obtain w by decimating u by $2^{n/2}+1$.
2. For an even n, the small set of Kasami sequences is given by:

$$K_s(u,n,m) \quad = \quad \begin{cases} u, & m = -1 \\ u \oplus T^m w, & m = [0,\ \ 2^{n/2}-2] \end{cases} \tag{2.8}$$

where:

$\oplus$: *is the modulo 2 sum operator (XOR operator);*
m : *is a shift parameter for w;*
T^m : *is a cyclic left shift operator by m shifts.*

3. Considering u and w, obtain v by decimating u by $2^{(n/2)+1}+1$. For $mod(n,4) = 2$, the large set of Kasami sequences is given by:

$$K_L(u,n,k,m) \quad = \quad \begin{cases} u, & k=-2; \quad & m=-1 \\ v, & k=-1; \quad & m=-1 \\ u \oplus T^k v, & k=[0,\ \ 2^n-2]; \quad & m=-1 \\ u \oplus T^m w, & k=-2; \quad & m=[0,\ \ 2^{n/2}-2] \\ v \oplus T^m w, & k=-1; \quad & m=[0,\ \ 2^{n/2}-2] \\ u \oplus T^k v \oplus T^m w, & k=[0,\ \ 2^n-2]; \quad & m=[0,\ \ 2^{n/2}-2] \end{cases} \tag{2.9}$$

where:

$\oplus$: *is the modulo 2 sum operator (XOR operator);*
k : *is a shift parameter for v;*
m : *is a shift parameter for w;*
T^k : *is a cyclic left shift operator by k shifts.*
T^m : *is a cyclic left shift operator by m shifts.*

Cross-correlation & Autocorrelation

- **Cross-correlation** — can be defined as a measure of similarity between two signals when one lags relative to the other. Reducing this phenomenon in different transmitting channels or even different symbol's codes is critical to allow correct detection of a PN coded signal and limit the effect of multiple transmissions in a shared medium. In other words, if two systems communicate with orthogonal codes, both can use the medium simultaneously without interfering much with each other's transmissions as the receiver will still be able to correctly despread the symbols of each channel.

  The cross-correlation between sequences u and v of length N is given by:

$$R_{u,v}(i) \quad = \quad \sum_{j=-N+1}^{N-1} u(j)\, v(j+i) \tag{2.10}$$

  The cross-correlation function between any shifted or non-shifted Gold sequences takes only three possible values:

$$\theta_{u,v}(i) \quad = \quad \{-t(i),\ -1,\ t(i)-2\} \tag{2.11}$$

  where:

$$t(i) \quad = \quad \begin{cases} 2^{(i+1)/2}+1, & \textit{i is odd} \\ 2^{(i+2)/2}+1, & \textit{i is even} \end{cases} \tag{2.12}$$

  The corresponding cross-correlation function for Kasami sequences takes values from:

$$\theta_{u,w,v}(i) \quad = \quad \{-t(i),\ -s(i),\ -1,\ s(i)-2,\ t(i)-2\} \tag{2.13}$$

  where:

$$t(i) \quad = \quad 2^{(i+2)/2}+1, \qquad \textit{i is even} \tag{2.14}$$

$$s(i) \quad = \quad \frac{1}{2}(t(i)+1) \tag{2.15}$$

- **Autocorrelation** — or cross-autocorrelation, can be defined as the cross-correlation of a signal with itself at different instants in time. For instance, Inter-symbol Interference (ISI) caused by reflections on the receiver's end can attenuate/amplify parts of the signal and lead to false positives or misdetection of PN codes if the autocorrelation peak is not well delimited. This peak influences the capability of the receiver to self-synchronize, and good code synchronization capability is particularly crucial in wave propagation mediums affected by noise and multipath propagation.

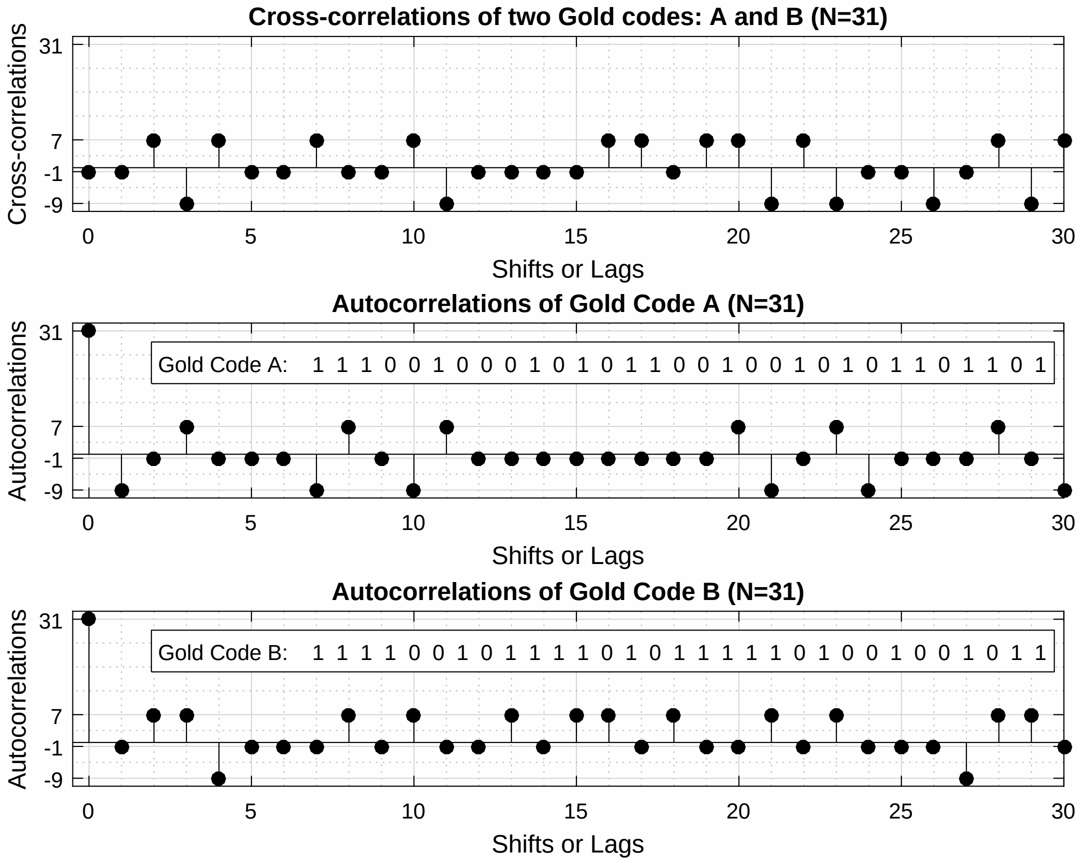


Figure 2.3: Plots of cross-correlations and autocorrelations of Gold codes with period $N = 31$.

The autocorrelation at lag i for a sequence u of length N is given by:

$$R_{u,u}(i) \quad = \quad \sum_{j=0}^{N-1} u(j)\, u(j-i) \tag{2.16}$$

The autocorrelation function of a m-sequence has period $N = 2^n - 1$ and has ideal autocorrelation properties, taking only two possible values:

$$R(i) \quad = \quad \begin{cases} N, & i = 0 \\ -1, & 1 \leq i \leq 2^n - 2 \end{cases} \tag{2.17}$$

2.2.3 CTBV Signal Processing & DS Detection

Although not unique, a popular method for processing IR-UWB signals and detecting UWB *chips* consists in feeding a Continuous-Time Binary Value (CTBV) signal to a digital circuit designed in a Rake Receiver topology, an approach described by [14, 15]. Following a threshold sampling stage, the CTBV input is fed upon a Rake Receiver CMOS circuit that performs correlation of the input signal with a range of expected DS codes. Taking into consideration the objectives of this dissertation work, which makes use of DS, the current section will briefly describe how CTBV signal processing works and how it can be used in a Rake Receiver topology.

- **CTBV Signal Processing**

  Continuous-Time Binary Value (CTBV) consists on implementing a receiver that performs continuous 1-bit threshold sampling and serial propagation of the binary value at the input of the receiver to several detection fingers, or correlators, of a Rake Receiver logic circuit. No sampling clock is used, the propagation takes place at the intrinsic speed of the CMOS logic gates, usually around a few tens of *picoseconds*. The length of the CTBV chain should be long enough to accommodate the pulses of at least one entire *chip*. Likewise, the propagation delay of the individual logic elements that form the chain should be significantly lower than the inverse of the bandwidth of the IR-UWB pulses. By combining sets of several logic elements, delay buffers can be implemented along the chain to provide pulse isolation for the Rake correlators.

$$CTBV\ chain\ length \geq chip_{size} \tag{2.18}$$

$$CTBV\ logic\ propagation\ delay \ll \tau_{pulse} \qquad \mathrm{(s)} \tag{2.19}$$

$$chip_{size} = N \tag{2.20}$$

$$\tau_{pulse} = \frac{1}{BW_{pulse}} \qquad \mathrm{(s)} \tag{2.21}$$

$$\tau_{chip} = \tau_{pulse} \times chip_{size} \qquad \mathrm{(s)} \tag{2.22}$$

  where:

  N : *is the length of the Gold code;*

  τ_{pulse} : *is the IR-UWB pulse length* (s)*;*

  BW_{pulse} : *is the pulse bandwidth (*Hz*);*

  τ_{chip} : *is the chip length* (s)*.*

- **Rake Receiver Topology & DS Detection**

  A Rake Receiver can be described as a logic circuit that performs correlations between its input signal and a sufficient number of delayed versions of the expected signals, which pulses match known PN sequences. When used in conjunction with a CTBV chain, either the chain or the correlators' output must be sampled at a frequency that should be at least equal to the inverse of the time of *chip* (equation 2.23). Each correlator, or finger, contains a slightly time-shifted version of the pulses derived from the CTBV chain and can potentially contain a recognizable PN sequence. The minimum number of correlators that assures detection of a single PN code is given by the ceiling of the ratio between the IR-UWB pulse bandwidth and the CTBV chain sampling frequency as shown in equation 2.24.

The Rake topology is resilient to the effects of multipath propagation and fading, as each finger attempts to detect a match for a known *chip* on its own, yielding more chances for a sequence to be detected in noisy or multipath environments. Detection thresholds can also be set in ways to enable partial *chip* recovery, therefore increasing the chances of detecting a sequence: for instance, by allowing an arbitrary number of misaligned bits in a received *chip* or by combining the partial detections of different correlator fingers.

$$CTBV\ chain\ sampling\ frequency: \qquad f_{s\ _{CTBV}} \geq \frac{1}{\tau_{chip}} \qquad (\mathrm{Hz}) \tag{2.23}$$

where:

$f_{s\ _{CTBV}}$: *is the sampling frequency of the CTBV chain (*Hz*);*

τ_{chip} : *is the chip length (*s*);*

$$Number\ of\ Rake\ Correlators\ (Fingers): \qquad R \geq \left\lceil \frac{BW_{pulse}}{f_{s\ _{CTBV}}} \right\rceil \tag{2.24}$$

where:

BW_{pulse} : *is the IR-UWB pulse bandwidth (*Hz*);*

$f_{s\ _{CTBV}}$: *is the sampling frequency of the CTBV chain (*Hz*);*

2.3 Multi-Band UWB

UWB transceivers can either operate in Single-Band (Impulse-Radio) or Multi-Band schemes, the latter option gaining ever more momentum while IR-UWB is already widely used in several UWB applications. Each of these schemes use corresponding modulation techniques, have appropriate medium access control options available, and each offer unique advantages over the other.

Unlike Single-Band Impulse-based UWB, Multi-Band separates the UWB available spectrum into sub-bands of at least 500 MHz as per the regulatory definition of UWB. This provides isolation between different channels, the bandwidth of each receiver is limited to the sub-band bandwidth. Higher data-rates can still be achieved if several sub-bands are used simultaneously. Multi-Band UWB can be achieved, essentially, in two different ways: Multi-Band Impulse-Radio (MB-IR) and

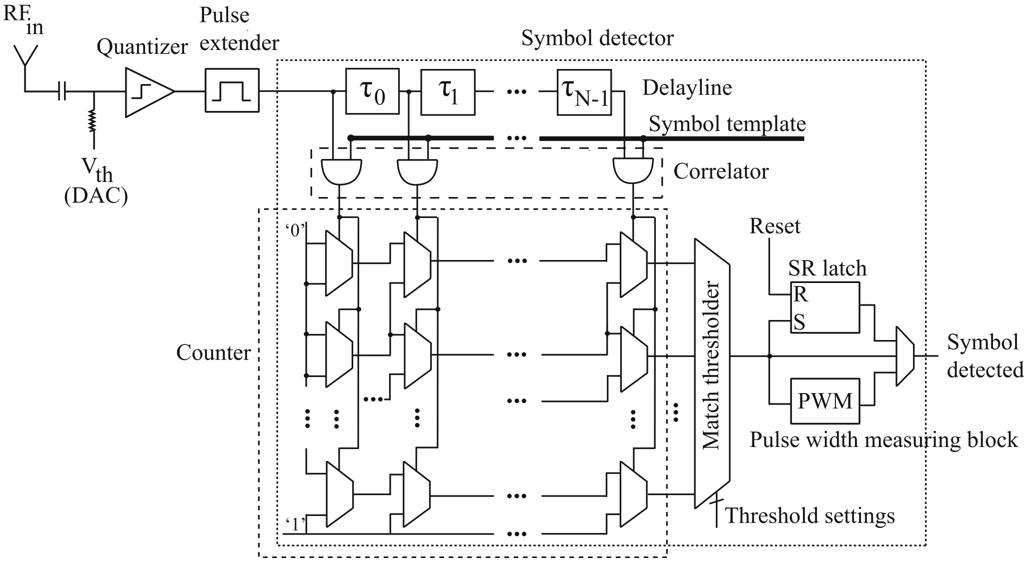


Figure 2.4: Block diagram of the classic CTBV Rake Receiver topology with multiple parallel correlator fingers (Image source: [25]).

Multi-Band Orthogonal Frequency-Division Multiplexing (MB-OFDM) according to [19]. While MB-IR works similarly to IR-UWB in each sub-band, MB-OFDM uses a Time-Frequency Code mechanism where each time-slot uses a different carrier frequency, useful both for diversity and multiple user access.

This work focus exclusively on Single-Band Impulse-based UWB (IR-UWB) for its simplicity and less complex implementation of RF front-ends over Multi-Band UWB. The aim of using as few resources as possible and low-cost components also determined the use of IR-UWB even if that implies a degradation of noise immunity or reduced number of available channels (user capacity) of the system when compared to a more complex Multi-Band system. Even though Multi-Band UWB receivers can make use of components with softer bandwidth requirements due to band isolation, these systems generally imply the use of multiple transceiver circuits and efficient synchronization schemes to make the all available channels usable, therefore increasing the costs and complexity of the implementation at no additional gain for the purpose of the proof-of-concept demonstrations that this dissertation aims to offer.

As such, IR-UWB architectures, modulation schemes and medium access control techniques have been described in this chapter, leaving out Multi-Band UWB to be later pursued by the reader's interest through other publications such as [19], where further information on Multi-Band UWB can be found. Following these definitions, some IR-UWB current applications will also be referenced in the following section.

2.4 IR-UWB Applications

A relevant author in IR-UWB research and prototype development is Professor Tor Sverre Lande from the Department of Informatics at University of Oslo. Extensive literature can be found from this author describing an equally vast range of UWB solutions, applications, techniques and hardware development. These include radar [26, 27, 28] and ranging measurement devices [26, 27, 28] based on IR-UWB and CTBV processing using Direct Sequence CDMA (DS-CDMA) Pseudo-random Noise (PN) codes on dedicated CMOS logic [15]. Other of his publicly known research works with UWB are biomedical sensing [29, 30, 31, 32, 33], radio communication [34, 35], radar imaging [36], RFID [37] and wireless sensor networks [38].

Nevertheless, several other authors have conducted experiments and research on UWB and IR-UWB, and some of those works will be presented in the next sections according to their category of application.

2.4.1 Short-Range Communications

- *CMOS Impulse Radio Receiver Front-end* [34] describes a CMOS implementation of an UWB receiver front-end architecture based on a combination of a LNA, integrator and threshold pulse shaper. The sequence of received impulses is passed through a delay line of CMOS inverters and sampled at a much lower rate than the input signal frequency, being later decoded by a parallel network of correlators in a Rake Receiver topology.

- *A* 3 GHz *to* 5 GHz *IR-UWB Receiver Front-End for Wireless Sensor Networks* [38] presents a working prototype of a CMOS IR-UWB receiver for sensor networks in the 3 GHz to 5 GHz frequency range, with a low power operation of about 26.6 mW. The system relies on CTBV and its targeted for low-power and low data-rate communication over short distances, offering 1 Mbit/s at 10^{-3} BER in a chip area of 0.9 mm^2 under 90 nm CMOS technology.

2.4.2 Short-Range Radar, Ranging Measurements and Indoor Positioning

- *Impulse-based ultra-wide-band (UWB) radio systems and applications* [15] describes in detail a CMOS implementation of a IR-UWB radar system using CTBV signal processing that makes use of PN codes. The time-domain pulse detector consists of a sweeping threshold sampler and a 1-bit quantizer on the incoming RF signal. This system is similar to the Radio Receiver front-end described in [34] and likewise, the main system and sampling clock can work at several orders of magnitude lower frequency than the UWB pulse frequency. This enables low-power operation in both the receiver and transmitter. Expression 2.25 indicates the relationship between the required system clock/sampling frequency (f_{clk}), temporal resolution (minimal pulse duration τ) and size of *chip* ($chip_{size}$) as per [15]. The prototype device can also perform ranging measurements based on Time of Arrival or Time of Flight

(ToA/ToF) estimations.

$$Clock/Sampling\ Frequency: \quad f_{clk} \quad = \quad \frac{1}{\tau \cdot chip_{size}} \qquad \text{(Hz)} \tag{2.25}$$

- *A High-Resolution Short-Range CMOS Impulse Radar for Human Walk Tracking* [28] refers an impulse-based UWB radar with a 4-channel sampling receiver, a LNA and an integrator. It can track the movement of a person indoor up to a distance of 12 m with a range resolution of about 1.5 cm while consuming 80 mW of power.

- *A Time to Digital Converter for use in Ultra Wide Band Radar Sensor Nodes* [39] describes a Time to Digital Converter architecture for UWB radar implementations that allows RF front-end interfacing with microcontrollers. Contrary to other solutions, the design does not rely on state-of-the-art ADCs, ultra fast memories or high-end logic elements; instead it uses a Time to Voltage converter made of simple latch memory elements, variable threshold comparators and discrete capacitors. The 16 comparator-capacitor pairs works as a form of analog memory chain that is then sampled by the built-in ADC of a microcontroller. The ranging computations are based on ToF/ToA estimation and the detection distance goes up to 10 m with a 1.95 cm range resolution.

$$Range\ Resolution: \quad range_{res} \quad \geq \quad \frac{c_0}{BW_{pulse}} \qquad \text{(m)} \tag{2.26}$$

2.4.3 Imaging and SAR

- *Ultrawideband (UWB) Radar Imaging of Building Interior: Measurements and Predictions* [7] explores through-the-wall imaging by implementing an UWB Synthetic-Aperture Radar (SAR).

- *Ultra Wide Band Synthetic Aperture Radar Real Time Processing with a Subaperture Nonlinear Chirp Scaling Algorithm* [40] describes an airborne UWB SAR system that can produce aerial imagery with up to 1 m × 1 m resolution (dependant on aircraft flight level).

- *Imaging of Oil-Well Perforations Using UWB Synthetic Aperture Radar* [41] applies UWB technology to monitor oil well perforations with a SAR algorithm.

2.4.4 Biomedical

- *Direct RF Subsampling Receivers Enabling Impulse-Based UWB Signals for Breast Cancer Detection* [6] promises a less risk-inducing diagnostic alternative based on UWB when compared to ionizing radiation or potentially dangerous high-energy medical diagnostic techniques.

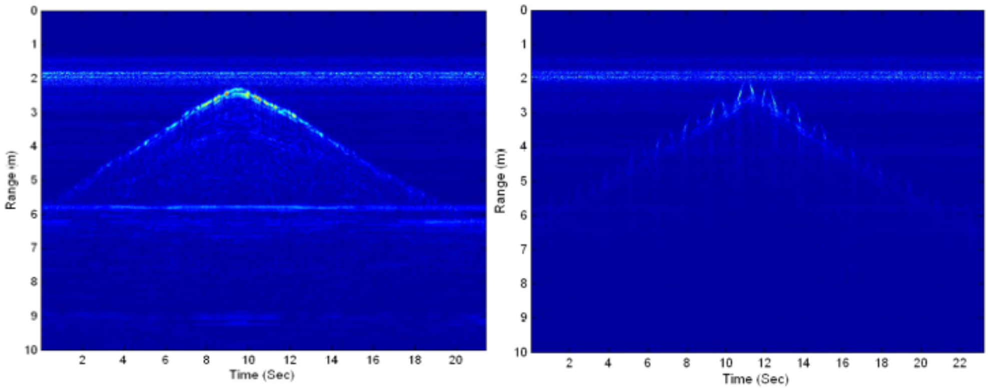


Figure 2.5: Micro-Doppler signature of a person walking in front of an UWB radar system. Left plot: signature of a person walking in front of an UWB radar system. Right plot: person walking and performing an arm swing motion. (Image source: [42]).

- *Vital Signs Monitoring with a UWB Radar Based on a Correlation Receiver* [43] demonstrates a Vital Signs Monitoring application of respiratory functions and heart-beat rates at a distance of tens of centimetres.

2.4.5 Automotive Radar and Collision Avoidance Systems

- *22–29 GHz Ultra-Wideband CMOS Pulse Generator for Short-Range Radar Applications* [13] describes a UWB automotive radar for short-range collision avoidance systems.

2.4.6 Commercially available solutions

Companies such as TimeDomain®, Camero® and Cambridge Consultants Ltd. have turn-key UWB solutions such as UWB radars on the market, including some for civilian use. Nevertheless, due to the closed development nature of commercial solutions those will not be considered in this state-of-the-art analysis.

2.5 FPGA implementations of UWB systems

The following applications include an FPGA in the design to at least partially execute the computations associated with digital signal processing. In the receiver element, the signal is usually obtained from an RF front-end that might include threshold detectors, ADCs or other sampling circuits. Extremely complex RF front-end circuits are out of the scope of the dissertation work but some are hereby referenced to demonstrate current and higher-complexity UWB implementations.

- *Towards Low Cost, High Speed Data Sampling Module for Multifunctional Real-Time UWB Radar* [42] achieves through-wall radar imaging up to 100 GSamples/s with an FPGA and external high-speed ADCs.

- *Crystal-less Duty-Cycled-When-Active IR-UWB Transceivers* [44] mentions a working prototype of an IR-UWB communication system that interfaces with a CMOS transceiver architecture.
- *Peak Detection Ultra-Wideband Transceiver Using Tunnel Backward Diode* [45] shows the implementation of non-coherent UWB receivers in both single-band and multi-band scenarios. The envelope detector is implemented with a diode detector circuit. The authors claim a data-rate of 190 Mbit/s in single-band mode and 400 Mbit/s in multi-band operation.
- *A Low-Cost COTS UWB Transceiver for Biological Applications Achieves* 50 Mbit/s *with* < 10^{-6} *Raw BER* [46] documents a working system built upon an RF front-end using off-the-shelf parts consisting on a LNA, a BPF, a Gain Amplifier, an RF detector and a Comparator. In the receiver, the output of the comparator is fed onto an FPGA in order to detect the PN sequences and recover the payload data. Although the achieved data-rate and communication range are somewhat limited, this is an interesting work that is closer to the work that is to be developed.

2.6 Conclusions

Taking into consideration the publicly available research on UWB technology, its applications and current capabilities, which only a few comparatively relevant ones were referenced above, its safe to say that UWB development is gaining a lot of momentum since the regulatory liberation. Nevertheless, the technology is still relatively expensive to be made available at the consumer or even industrial level and most current solutions demand the fabrication of custom CMOS chips. Therefore, there is still a gap for cost-effective UWB solutions at the present day. It shall be noted then that the aim of this dissertation work is to help bring UWB technology to a broader range of usages while bringing the deployment cost down by evaluating different methodologies with smaller and less expensive Bill of Materials (BoM) that mostly make use of inexpensive off-the-shelf parts.

Programmable logic could potentially become an attractive development and prototyping platform for IR-UWB technology as its easily reconfigurable and upgradable, unlike custom ASIC development and other more expensive solutions that offer improved performance but yet require higher development and deployment costs.

Contrary to ASIC solutions, the work to be presented in the following chapters will demonstrate how to use FPGAs in ways they are not usually supposed to be exploited such as using logic cells as constant delay buffers. Yet, even considering the inherent limitations and variability of the Place & Route process and the fact that no manual chip floor-planning was done, the design still yield relevant results. As such, the work provides a foundation for future IR-UWB lines of development with lower-cost programmable logic.

Chapter 3

System Architecture

Introduction

As mentioned in previous chapters, the proposed approach for implementing an IR-UWB system using programmable logic is based on the strategy introduced by Professor Tor Sverre Lande in [25, 15]. The principles of implementation will be prove to be very similar, but the methodology differs in the physical implementation of the system as no custom CMOS chip will be used for the RF front-ends or for PN sequence correlation. This chapter describes how the prototype was designed in order to accomplish the desired goals for the proof-of-concept demonstrations.

One of the challenges introduced by this different approach is the utilisation of off-the-shelf programmable logic chips, like CPLDs and FPGAs, instead of an ASIC solution. Because the CTBV topology relies on fixed and ideally constant delays through the input signal chain, application-specific chips are designed to offer this desired functionality at the maximum performance offered by their CMOS lithography technology. Thus, ASIC designs can meet precise and predictable timing requirements. Yet this CMOS implementation is not usually re-configurable. This means that a performance upgrade typically requires a re-design and fabrication of a new chip.

On the other hand, programmable logic designs are more easily upgradeable as long as the design can be ported to newer FPGA or CPLD cores. Notwithstanding, all upgrades will still need functional validation through formal testing. Despite the challenges, these programmable platforms are very attractive from a system's designer point of view if they could offer similar functionality to ASIC solutions, at a fraction of the development and implementation cost. Even if the performance is still limited, mostly because the design makes use of FPGA's logic elements in ways they were not really designed for, this work will propose a portable architecture for generating and detecting IR-UWB pulses using programmable logic devices with minimal external components.

Following this introduction, the information in this chapter is divided into four major sections:

- **Section 3.1: Design Challenges & Decisions**, describes some of the challenges encountered during the design phase and some decisions on how to overcome them;

- **Section 3.2: System Components**, defines the architecture of the components built into the IR-UWB system;
- **Section 3.3: System Configurations**, demonstrates two possible applications for the system: performing ranging measurements and operating as a transceiver;
- **Section 3.4: Conclusions**, closes this chapter by showing a summary of what been developed.

3.1 Design Challenges & Decisions

This section will summarize some of the foreseen or encountered implementation challenges and the corresponding design decisions.

- **FPGA vs CPLD** — Programmable logic is available in two different variants: FPGAs or CPLDs. FPGAs traditionally offer faster speeds, greater number of logic resources and more periphels like RAM. A balance on performance vs cost was reached and is described on Chapter 4, System Implementation on Programmable Logic Hardware, further justifying this choice, so the pulse detector (RX) and pulse generator (TX) blocks were implemented in this type of programmable logic device.
- **CTBV controlled jitter requirements** — FPGAs' combinational logic blocks, which do not rely on clock synchronization signals, do not offer constant or entirely predictable delays as immutable CMOS circuits do. This is particularly verifiable when the volatile Place & Route process has to accommodate design compilation changes or design extensions. Even though FPGA floor-planning is a possible solution for mitigating Place & Route constraints, this is not an ideally solution as it is time consuming and not universal to all FPGA chips. This option was not even available in the academic license of the IDE used, meaning that an alternative methodology had to be found.

  On the other hand, the CTBV topology requires a chain with predictable and equally distributed delays. Since guaranteeing timing consistency in this chain is so crucial, a possible solution is to implement delay chains of elementary logic elements with a common HDL primitive. These chains attempt to average out jitter in the CTBV chains on both the Receiver and Transmitter modules and seem to be sufficiently effective against most Place & Route variations.

  There are, however, some limitations: after final compilation, some fine-tuning might be necessary when porting the HDL code to other FPGA families, or even to different batches of identical chips, as manufacturing variations can induce performance differences between chips of the same logic family. On the upper side, Rake Receiver circuits are self-synchronizing if the delay chains successfully self-average out to almost identically delays throughout the CTBV chain – in other words, if some batches of boards perform more poorly, they can still be used for Ranging purposes, exhibiting worse measurement resolution, or in transceiver applications with other equally affected boards.

- **RF front-ends' complexity** — due to development and implementation costs, BoM restrictions and the objective to pursue a more inexpensive IR-UWB implementation, no external RF front-ends were developed to provide the system with Wireless connectivity. The concept will be tested using a coaxial cable, serving as an IR-UWB transmission line. This decision reduced the complexity and final cost of the implementation without sacrificing the core concept, which was to provide a solution for generating and detecting IR-UWB pulses with an FPGA. Furthermore, the design could be extended at any time to include Wireless functionality.

- **Design Portability and Upgradeability** — in order for the design to be easily portable to different and newer FPGA families and to facilitate future upgrades, no proprietary FPGA technologies were adopted. Instead, only universally synthesizable logic was used like the Verilog primitive *wire* and similar universal primitives. Some proprietary IP cores were used for implementing FIFOs on the on-chip RAM, but similar alternatives can be found for different FPGA manufacturers. The HDL of choice was System Verilog and the IR-UWB control modules are designed in Finite State Machines (FSM).

3.2 System Components

The core components of the proposed IR-UWB system architecture based on programmable logic is illustrated by the Functional Block Diagram of Figure 3.1. This system is intended to be implemented using a Field-Programmable Gate Array (FPGA), as Chapter 4, System Implementation on Programmable Logic Hardware, further justifies. A description of the core components' functionality is given below:

- **IR-UWB TX Signal Flow**:

  UART RX input

  The input for a UART RX module which serves as an interface between this system and an external control system that provides the transmitted payload data.

  UART RX module

  Receives binary data from an external control system and populates the TX FIFO.

  TX control logic & TX FIFO

  Controls the TX chain and holds the binary data payload while the TX module is busy transmitting data. The TX FIFO is optional when the UART baud-rate is slower than the baud-rate of the TX module.

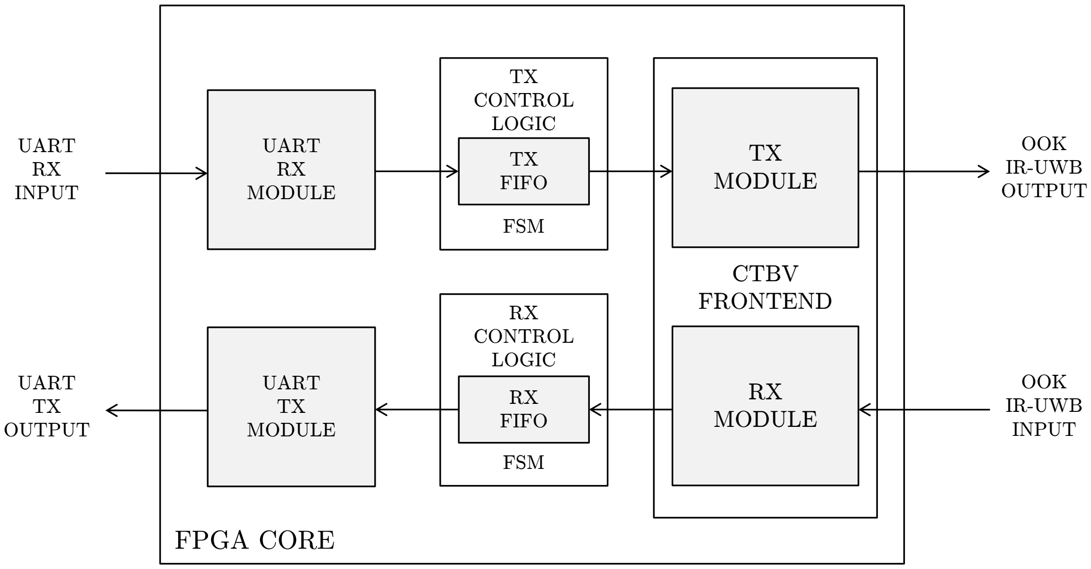


Figure 3.1: Functional Block Diagram of the FPGA's toplevel module. The arrows indicate the TX/RX signal's flow between different system modules.

TX module

Reads the TX FIFO and converts the binary data payload into CTBV *chips* according to the configured Gold or Kasami codes.

CTBV output

Represents an FPGA output pin from witch the CTBV is converted into OOK-modulated IR-UWB pulses and sent through a coaxial cable .

- **IR-UWB RX Signal Flow**:

CTBV input

Rrepresents a coaxial cable input that feeds an FPGA input pin where OOK-modulated IR-UWB pulses are sampled to form a CTBV signal in the RX module.

RX module

Processes the CTBV input signal in a Rake Receiver logic circuit, performing signal correlation and Gold or Kasami sequence detection. Is the first step in retrieving the binary payload data from received *chips*.

RX control logic & RX FIFO

Controls de RX chain and holds the correlators' data or binary data payloads, depending on the current configuration, while the UART TX module is busy transmitting serial data. The RX FIFO is needed when the UART baud-rate is slower than the TX module's baud-rate.

UART TX module

Reads the RX FIFO and forwards the binary data payload or correlator data, depending on the current configuration, through the UART TX Output.

UART TX output

The output for the UART TX module which serves as an interface between this system and an external control system to which the received data is sent for further processing.

3.2.1 Transmitter Module Architecture

In order to generate a CTBV output, the TX module can be described by a chain of $XORs$ logic gates with delay buffers in between them. After a initialization of the chain to a known state (all *zeros* or all *ones*), if a TX signal is received by the module the circuit simultaneously loads the polynomial coefficients of the *chip* to be transmitted and the desired pulses will begin to propagate through the chain to the CTBV output.

Because this circuit is strictly combinational, no internal clock signal controls the output from within the TX module, so a CTBV signal is permanently connected to the CTBV output until the TX control logic module overrides the TX module and takes over the CTBV output pin, putting it in High-Impedance Mode ($Hi-Z$). This only occurs approximately after $N \cdot \tau$ units of time, where N represents the number of polynomial coefficients and τ the delay of each delay buffer, giving the TX module enough time to complete its transmission. After sending a *chip*, the module needs to be re-initialized to a known state before being able to execute another transmission.

The TX module Architecture is illustrated in Figure 3.2.

$$Ideally, \qquad \tau = \tau_{pulse} = \frac{1}{BW_{pulse}} \qquad (\mathrm{s}) \tag{3.1}$$

where:

τ : *is the delay of a TX module delay buffer (*s*);*

τ_{pulse} : *is the IR-UWB pulse length (*s*);*

BW_{pulse} : *is the pulse bandwidth (*Hz*).*

$$N = \frac{\tau_{chip}}{\tau_{pulse}} \tag{3.2}$$

where:

N : *is the number of delay buffers of delay τ in the chain, the length of the Gold or Kasami code or the chip$_{size}$;*

τ_{chip} : *is the chip length (*s*);*

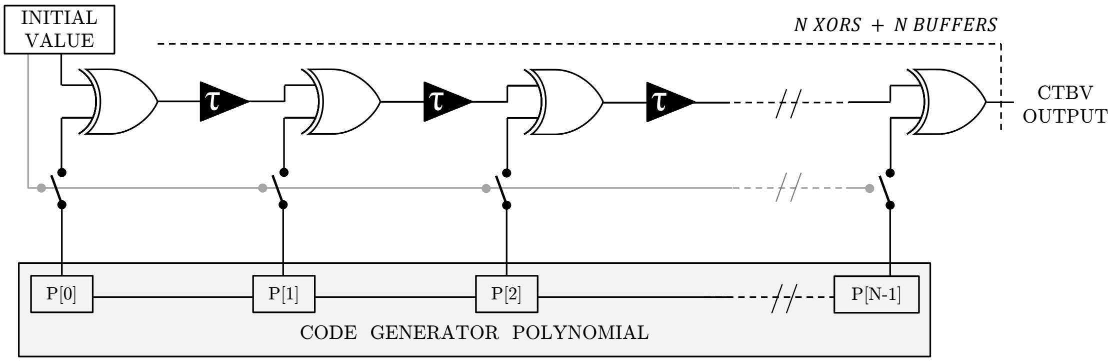


Figure 3.2: Functional Block Diagram of the TX Module Architecture. A τ buffer represents a delay chain. The CTBV output is generated by loading the coefficients of a code generator polynomial to a XOR chain of N elements, after the chain is initialized with *zeros* or *ones*. The length of the chain matches the length of the Gold or Kasami code used.

Code generator polynomial

Although the TX module chain does not immediately accept Gold or Kasami code coefficients, it does accept a generator polynomial that can be easily obtained after a XOR operation between a left-shifted version and the original code itself. The resulting generator polynomial has the same length, N, of the associated code and can be obtained by discarding the most significant bit (MSB) of this operation. Equation 3.3 demonstrates the case for a generic 9 bit code **010001101** and its corresponding generator polynomial.

$$\textit{Generator Polynomial} = \begin{array}{r} \mathbf{0\,1\,0\,0\,0\,1\,1\,0\,1}\,0 \\ \oplus\ 0\,\mathbf{0\,1\,0\,0\,0\,1\,1\,0\,1} \\ \hline 0\,\mathbf{1\,1\,0\,0\,1\,0\,1\,1\,1} \end{array} \tag{3.3}$$

Table 3.1: XOR truth table

A	B	(A $\oplus$ B)
0	0	0
0	1	1
1	0	1
1	1	0

Gold or Kasami code generation with a generator polynomial

In order to generate a Gold or Kasami code from the corresponding generator polynomial, a series of specific operations are applied to the polynomial itself, namely sequential XOR and right-shift binary operations. The output can be arranged in matrix form, where the code will eventually appear LSB-first on the last furthest right column of that matrix. This binary operation is mimicked by the TX module's XOR logical elements and delay buffers. The Mathworks® Matlab code on Listing 3.1 presents an algorithm for obtaining a Gold or Kasami code from its corresponding generator polynomial. The resulting matrix of the algorithm is presented in Equation 3.4.

Listing 3.1: Matlab code of an algorithm to obtain a Gold or Kasami code from the corresponding polynomial

```
% Define the polynomial
poly = [1 1 0 0 1 0 1 1 1];
code_length = length(poly);
code_matrix = (zeros(1,code_length));

% Calculate the code matrix
for i = 2:code_length+1
        code_matrix(i,:) = xor([0 code_matrix(i-1,1:end-1)],poly(1,1:
            length(poly)));
end

% Extract the resulting Gold or Kasami code by flipping
% the furthest right column of the code matrix
code = fliplr(code_matrix(2:end,code_length)');
```

$$code\_matrix = \begin{bmatrix} \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \mathbf{1} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \mathbf{0} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \mathbf{1} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \mathbf{1} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \mathbf{0} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \mathbf{0} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \mathbf{0} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \mathbf{1} \\ \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{1} & \enclose{horizontalstrike}{0} & \enclose{horizontalstrike}{1} & \mathbf{0} \end{bmatrix} \tag{3.4}$$

3.2.2 Receiver Module Architecture

The RX module follows a Rake Receiver topology with R correlators or fingers. The correlators, made of XOR primitives, tap into the CTBV delay line at twice as often as the pulse length, or $\tau/2$, increasing the changes of synchronization and code detection. It's also beneficial when performing ranging applications, as it increases the time resolution of the measurement. Equation 3.7 provides a formula for calculating the minimum number of Rake stages needed.

Each detection stage, or correlator, is made of N XOR elements. On one of the inputs, the $XORs$ tap the CTBV chain. The corresponding bit of the Gold or Kasami sequence to be detected is loaded on their second input. The resulting output of every XOR in a given correlator stage will be logic *zero* if all the input bits are aligned with the code sequence in that particular finger (Equation 3.5). The number of misaligned bits in the chain can be obtained by adding all the corresponding XOR outputs.

In other words, if the sum of all $XORs$ of a correlator is *zero*, a valid detection has occurred on that stage. The opposite scenario can be used to detect a complementary code: if the sum is equal to N, the code length, all bits are misaligned which indicates a detection of a complementary binary value (Equation 3.6). This property is very desirable, it enables the detection of two different *chips* with the same receiver module, saving resources on programmable logic implementations, and it has been successfully used in this work for a Transceiver configuration.

Its also possible to set tolerance thresholds when a full detection is not possible, minimizing the effects of interfering signals in noisy environments. In that case, the control logic of the receiver can allow an arbitrary maximum number of misalignments, depending on the size of the codes used. Although this implicates a trade-off between code detection and detection integrity, this option is a way of fine-tuning the system to better fit its working environment or to circumvent propagation delay differences between internal logic elements.

The RX module Architecture is illustrated in Figure 3.3.

$$Original\ code\ detection: \quad \begin{array}{r} 1\,0\,1\,1\,\mathbf{0\,1\,0\,0\,0\,1\,1\,0\,1}\,0\,1 \\ \underline{\oplus\ \mathbf{0\,1\,0\,0\,0\,1\,1\,0\,1}} \\ \mathbf{0\,0\,0\,0\,0\,0\,0\,0\,0} \end{array} \tag{3.5}$$

$$Complementary\ code\ detection: \quad \begin{array}{r} 0\,1\,0\,0\,\mathbf{1\,0\,1\,1\,1\,0\,0\,1\,0}\,1\,0 \\ \underline{\oplus\ \mathbf{0\,1\,0\,0\,0\,1\,1\,0\,1}} \\ \mathbf{1\,1\,1\,1\,1\,1\,1\,1\,1} \end{array} \tag{3.6}$$

$$Rake\ Correlators\ (Fingers): \qquad R \ \geq \ \left\lceil \frac{2 \cdot BW_{pulse}}{f_{s\ CTBV}} \right\rceil \tag{3.7}$$

where:

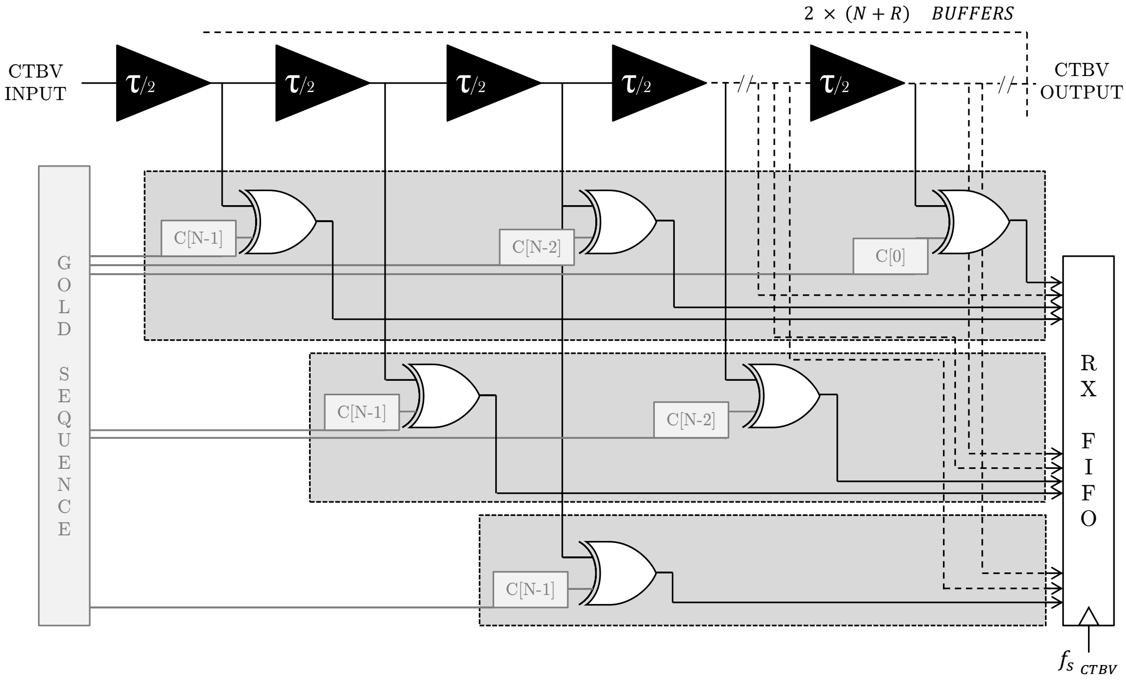


Figure 3.3: Functional Block Diagram of the RX Module Architecture. A $\tau/2$ buffer represents a delay chain. The CTBV signal is propagated from the input through the chain and through R Rake correlators or fingers. The output of the correlators is stored in the RX FIFO module in a synchronous configuration, at frequency $f_{s\ CTBV}$ that also matches the internal clock rate of the RX control module. The data is later analysed for Gold or Kasami code identification.

BW_{pulse} : *is the IR-UWB pulse bandwidth (*Hz*);*

$f_{s\ CTBV}$: *is the sampling frequency of the CTBV chain (*Hz*).*

As an example, the FPGA board on which this architecture was deployed has a maximum I/O bandwidth of at least 500 MHz. Due to the complexity of the design and internal routing, and the configuration used for I/O pins, the actual maximum generated pulse bandwidth was about 166 MHz for $K = 12$ delay buffer elements while still offering reliable Ranging measurements. The external oscillator which feeds the main FPGA clock runs at 50 MHz. Because the RX and TX modules work at the same clock of 50 MHz, based on Equation 3.7, an absolute minimum of 7 Rake fingers were needed, so the final design was built to use 7 correlators. As like any other FPGA design process, there is a constant trade-off here between system complexity and system performance. Using a larger R value would increase the quantity of logic resources used, adding more stress to the Place & Route process, which could further affect pulse performance. It would also increase the size requirements for the RX FIFO, without providing any practical benefit as the samples from the last correlators would become ever more redundant as design complexity increasingly reduces the IR-UWB pulse bandwidth.

3.2.3 Delay Buffer Architecture

The Delay Buffer architecture is illustrated in Figure 3.4. As previously mentioned in this chapter, the delay chain is composed of simple logic elements that are directly synthesizable from the Verilog primitive *wire*. The number of delays to include in each delay buffer varies with the target FPGA family on which the system is to be deployed and on the desired bandwidth of the IR-UWB pulses. There are some practical limits in FPGAs, like the maximum internal propagation speed of logic signals and the maximum switching rate, or allowable bandwidth, of I/O pins.

The maximum I/O switching rate will become the most limiting factor and absolute maximum allowable value for the IR-UWB system bandwidth. This derives from the fact that the usage of multiple I/O channels is not currently being proposed for simplicity and because low-cost FPGAs typically do not offer sufficient logic resources for implementing multiple TX/RX channels. Equation 3.9 defines a minimum value for K according to these parameters, although is absolutely not recommended to try to exceed or even design the system to operate too close to the limit of the I/O speed for reliability reasons.

However, the Place & Route process might also influence the maximum allowable bandwidth. Performance issues like different buffers having different propagation speeds in a chain might also imply a trade-off between stability and pulse bandwidth, leading to an increase of logic elements in the delay buffers in order to attempt averaging out theses effects and reach an equilibrium point.

It's also important to note that the K limit holds valid for the transmitter chain, but faster delay chains can and should be used in the receiver chain to increase code detection probability and the resolution of ranging measurements, as illustrated in the next sections of this chapter.

For reference, the hardware used for testing this design was roughly measured regarding internal propagation delay, which averaged to perform around 275 ps, or an average internal propagation rate of about 3.63 GHz per Verilog *wire* primitive. Based on the manufacturer's maximum I/O speed for the used I/O configuration - the pins can output a clock signal of up to 250 MHz [47]. This yields a ceiling for the K ratio of about 8, when considering that the pins should be able to provide a pulse bandwidth of at least double that figure. When solving the equation now for Maximum I/O Bandwidth, the maximum theoretical bandwidth of the IR-UWB pulses turns to be about 455 MHz. In practice, the pulse bandwidth proved to be less than that as other factors hereby already mentioned were also in place and affecting the overall performance. Chapter 4, System Implementation on Programmable Logic Hardware, gives additional information regarding hardware implementation.

$$Ideally, \qquad BW_{pulse} = BW_{IO} \qquad (\mathrm{Hz}) \tag{3.8}$$

where:

BW_{pulse} : *is the IR-UWB pulse bandwidth (*Hz*);*

BW_{IO} : *is the maximum achievable bandwidth on I/O pins (*Hz*).*

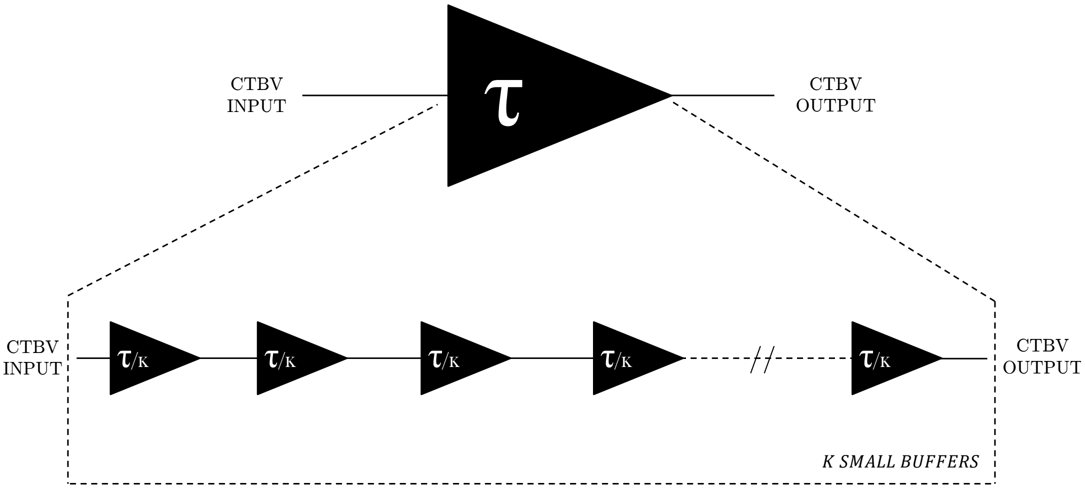


Figure 3.4: Functional Block Diagram of the Delay Buffer Architecture. A delay buffer τ is made of K logic elements, each introducing an ideal delay in the chain of τ/K.

$$Logic\ Elements\ per\ buffer: \qquad K \geq \left\lceil \frac{BW_{LE}}{BW_{I/O}} \right\rceil \tag{3.9}$$

where:

BW_{LE} : *is the typical bandwidth of internal logic elements (*Hz*);*

BW_{IO} : *is the maximum achievable bandwidth on I/O pins (*Hz*).*

3.2.4 Control Logic: Finite State Machines

The operation of the entire system is handled by dedicated logic blocks that implement Finite State Machines (FSMs). In the hardware implementation, these blocks are synchronous to the external clock oscillator running at 50 MHz. Each state machine handles either the RX or the TX module, separately:

- **TX control logic & TX FIFO**

  This block receives requests from the UART RX module, which is stored in the TX FIFO. When the system is working as a transceiver, upon detecting that there is a new data payload available in the TX FIFO, the TX control logic activates the TX module and the corresponding *chip* is transmitted.

  In ranging measurements mode, this FSM idles until a ranging command is sent through the UART – when a command is received, the TX module is instructed by the control logic to load a PN sequence for performing ranging measurements. The RX control logic also begins to sample the correlator data and storing it in the RX FIFO.

- **RX control logic & RX FIFO**

  The receiver control logic idles until the RX FIFO is populated with incoming correlator data from the RX module (Rake Receiver). For instance, in the transceiver configuration, the RX FIFO is continuously populated by the RX module and read by the RX control logic, which attempts to detect incoming *chips* by performing sums of the correlator outputs stored in the FIFO.

  For ranging operations, the RX control block keeps the RX FIFO clock disabled until the TX control logic sends a synchronization signal to enable RX FIFO operation. When the RX FIFO is finally full, no additional processing is done and the correlator data is transferred through the UART TX module for an external system to process it.

3.2.5 UART Interface

The UART modules provide a serial interface for the IR-UWB system to exchange data with an external control system. Depending on the configuration in place, the UART can either receive ranging requests or binary data payloads.

- **Ranging Configuration** — the UART receives a request from the external control system, setting the system to initiate a ranging measurement. After the measurement is completed the timing data is sent to the external control system for analysis.

- **Transceiver Configuration** — in this configuration, the UART functions as a "real-time" interface between the external control system and the IR-UWB system. Upon detection of a *chip* on the RX module, the UART TX module outputs the corresponding symbol. If the UART RX module receives a symbol from the control system, the symbol is converted into a *chip* and transmitted through the CTBV channel.

As the development of custom Verilog UART modules was out of the scope of this dissertation work, the modules used were obtained from an open-source third-party ([48]).

3.3 System Configurations

In this section, some of the possible configurations and applications of this IR-UWB architecture will be described in additional detail. In the Ranging configuration, the system can be used for performing ranging measurements of transmission lines or even detect faults in the lines. On the other hand, in Transceiver configuration the IR-UWB works as a bidirectional communication channel.

3.3.1 Ranging Configuration

The ranging configuration was inspired by Professor Tor Sverre Lande. His works in IR-UWB ranging applications, such as [14, 15], describe CMOS implementations for the concepts mentioned below. In wireless mediums, line-of-sight electromagnetic waves are reflected on a target depending on their frequency and on the properties of the target like geometry and the material's capability of absorbing the radiated energy at a given frequency. In wired mediums, ranging applications are made possible by the standing wave phenomenon observed in transmission lines, which results from impedance mismatching between a wave-guide and its load. Therefore, the Voltage Standing Wave Ratio (VSWR) can be used to estimate the potential of the reflecting waves according to known impedances of both the load and the transmission line, enabling predictions regarding the sensitivity of the receiver to a given load.

$$VSWR = \frac{1+|\Gamma|}{1-|\Gamma|} \tag{3.10}$$

$$\textit{Reflection Coefficient}: \qquad \Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{3.11}$$

where:

Γ : *is the reflection coefficient resulting from an impedance mismatch;*

Z_L : *is the impedance of the load (Ω);*

Z_0 : *is the impedance of the transmission line (Ω).*

In order for the signal to be reflected only at the load side, and not by the transmission line itself at the TX module's output, this output pin's impedance should match as much as possible the transmission line impedance. In situations where these power losses are not minimized, the reflected signal from the load end of the line might not be detectable by the RX module input pin. This mismatch loss at the transmitter side can be obtained from Equation 3.12, which can also be used to calculate the load side reflection's attenuation, if Z_L is finite and greater than zero.

$$\textit{Mismatch Loss} \qquad ML = -10\log_{10}\left(1-\Gamma^2\right) \qquad \text{(dB)} \tag{3.12}$$

where:

Γ : *is the reflection coefficient resulting from an impedance mismatch.*

ToF/ToA Estimation

The basic working principle behind this configuration is the Time of Flight / Time of Arrival (ToF/-ToA) concept. By measuring the time difference between the instant when a *chip* was transmitted

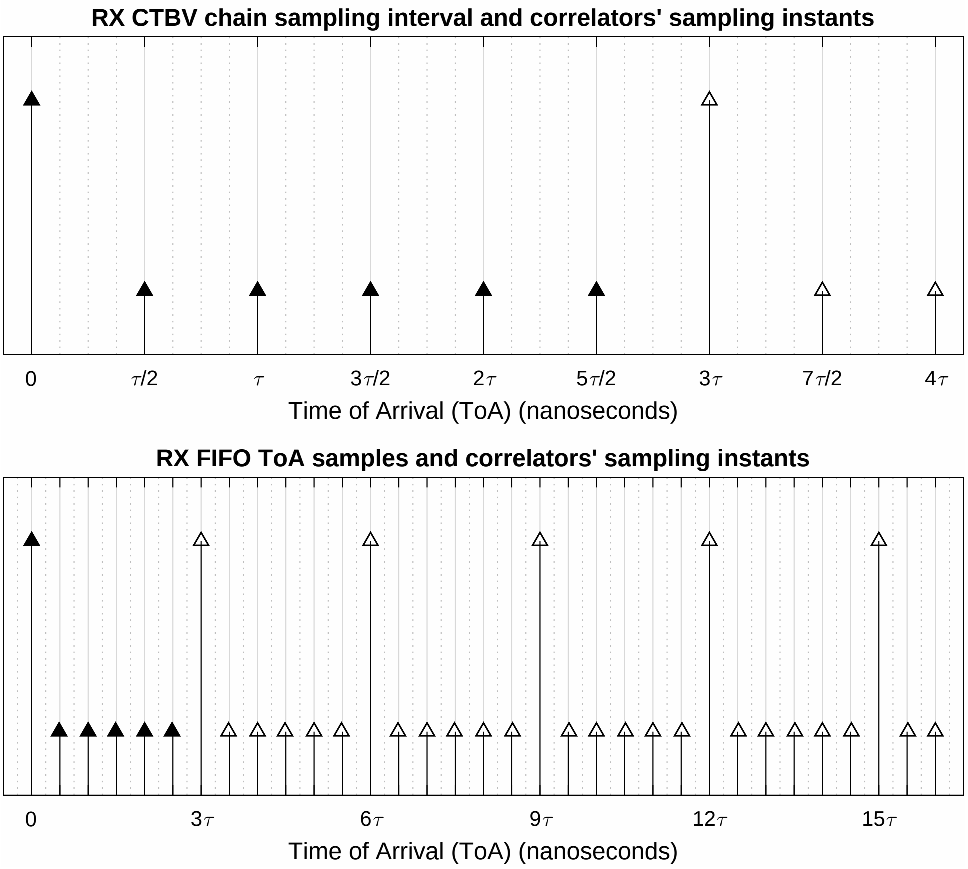


Figure 3.5: Time of Arrival estimation: on the top is represented the RX CTBV chain sampling interval, for $R = 6$ correlators; on the bottom the multiple RX FIFO ToA samples. In this example, the CTBV chain sampling period $\frac{1}{f_{s\ CTBV}} = 6 \cdot \tau/2 = 3 \cdot \tau$.

and the point of arrival after being reflected through the transmission medium, the Time of Flight of the *chip* through that medium can be estimated. Based on that time interval, the ToF is equal to half the interval between the two instants of time due to the Round Trip Time (RTT) nature of the measurement. After the ToF is known, the length of an open transmission line, or the distance between an antenna array and its target in free-space operation, can be estimated if the wave propagation velocity of the medium is known. Considering c_0, the constant that represents the speed of light in vacuum, wave propagation velocity can be roughly approximated to $0.66c_0$ in common coaxial cable transmission lines or to c_0 in free-space propagation. To improve the accuracy of the measurements, the actual Velocity Factor (VF) of the medium must be considered instead. In the CTBV chain of the RX module architecture described in this chapter, the resolution of this measurement is $\tau/2$, where τ is a delay desirably equal to the pulse length. Figure 3.5 illustrates how ToA is extracted based on the CTBV correlators' output stored in the RX FIFO; Equation 3.19 shows how to calculate ToA.

Considering now the present IR-UWB architecture, for this configuration to work the system had to be able to accurately synchronise the TX and RX modules together. This is accomplished by directly connecting the RX module to the TX module output, as illustrated by 3.6. Timing synchronization is achieved by listening in the receiver for the original *chip* sent by the TX module and continuing to listen until a reflection of that *chip* arrives. Then, the first detected *chip* serves as a reference time-frame for estimating the time it takes for the signal to propagate through the transmission line and be reflected on the load, Z_L.

The ToF/ToA technique can allow the system to estimate at what length a transmission line is open, or $Z_L = \infty$, up to the limit of maximum attenuation allowed by the receiver's input threshold voltage level in the FPGA's input pin. A potential divider circuit can be used to further counteract signal attenuation, in an attempt to extend the maximum operational range, or to accept loads with lower reflection coefficients. Notwithstanding, by bringing the transmission line to a potential that approaches the logic voltage threshold of the RX module input pin, the system is also more vulnerable to external interference.

The maximum depth of the RX FIFO also plays a role in limiting the maximum ranging distance of the measurements. The RX FIFO depth (Equation 3.21) is limited by the FPGA's peripherals, like RAM size, or the available general purpose logic resources depending on the FPGA family used. If the FPGA has available a sufficient number of fast adders but has limited RAM bits to be allocated to the RX FIFO, it's possible to extend the RX FIFO depth by compressing the correlators' data: if their outputs are added together before being stored in the FIFO, the size of the RX FIFO words is considerably smaller (Equation 3.23). Because the FPGA being used in this work has a large enough RAM but is not capable of performing fast-enough additions of the Rake fingers' outputs, each $XOR's$ output is stored in the RX FIFO, significantly increasing the number of bits needed per sample as Equation 3.22 shows. Equation 3.15 provides a way to estimate the maximum ranging distance when ignoring signal attenuation and the input sensitivity of the RX module input pin.

Additionally, there is also a minimum distance at which below that value the measurements can not be performed accurately, as load reflections over-impose the transmitted *chip*. This means the line length should be greater than the *chip* propagation distance on that medium. Equation 3.14 provides a way to estimate that minimum distance.

$$Load\ Distance: \qquad d_{Z_L} \approx v_P \cdot c_0 \times \tau_{ToF} \qquad (\mathrm{m}) \tag{3.13}$$

$$Minimum\ Load\ Distance: \qquad d_{Z_L} > v_P \cdot c_0 \times \tau_{chip} \qquad (\mathrm{m}) \tag{3.14}$$

$$Maximum\ Load\ Distance: \qquad d_{Z_L} < v_P \cdot c_0 \times \frac{\tau_{Ranging}}{2} \qquad (\mathrm{m}) \tag{3.15}$$

$$Ranging\ Resolution: \qquad S_r \approx v_P \cdot c_0 \times \frac{\tau}{2} \qquad (\mathrm{m}) \qquad (3.16)$$

$$Time\ of\ Flight: \qquad \tau_{\ ToF} = \frac{t_{ToA} - t_0}{2} \qquad (\mathrm{s}) \qquad (3.17)$$

$$Minimum\ Time\ of\ Flight: \qquad \tau_{\ ToF} > \tau_{chip} \qquad (\mathrm{s}) \qquad (3.18)$$

$$Time\ of\ Arrival: \qquad t_{ToA} = \left(RX_n \times \frac{1}{f_{s\ CTBV}}\right) + \left(R_n \times \frac{\tau}{2}\right) \qquad (\mathrm{s}) \qquad (3.19)$$

$$Maximum\ Ranging\ Interval: \qquad \tau_{\ Ranging} = \frac{1}{f_{s\ CTBV}} \times RXFIFO_{depth} \qquad (\mathrm{s}) \qquad (3.20)$$

$$RX\ FIFO\ Depth: \qquad RXFIFO_{depth} = \frac{RXFIFO_{size}}{RXFIFO_{word_{size}}} \qquad (3.21)$$

$$RX\ FIFO\ Word\ Size: \qquad RXFIFO_{word_{size}} = N \times R \qquad (\mathrm{bit}) \qquad (3.22)$$

or

$$RXFIFO_{word_{size}} = \lceil \log_2(N) \times R \rceil \qquad (\mathrm{bit}) \qquad (3.23)$$

where:

$d_{\ Z_L}$: *is the load distance (*m*), considering $Z_L = \infty$;*

v_P : *is the Velocity Factor or wave propagation speed in the medium;*

c_0 : *is the speed of light in vacuum (*m/s*);*

$\tau_{\ ToF}$: *is the chip propagation delay from the TX module to the load, or Time of Flight of the chip in the medium (*s*);*

τ_{chip} : *is the chip length (*s*);*

$\tau_{\ Ranging}$: *is the maximum Ranging interval, limited by available FPGA resources (*s*);*

S_r : *is the resolution of the ranging measurement (*m*);*

τ : *is the delay of a TX module delay buffer (*s*);*

t_{ToA} : *is the instant in time at which a chip reflection is received on the RX module (*s*);*

t_0 : *is the instant in time at which a chip is sent by the TX module (*s*);*

RX_n : *is the n^{th} sample stored in the RX FIFO, varies from $[0,\ (RXFIFO_{length} - 1)]$;*

$f_{s\ CTBV}$: *is the sampling frequency of the CTBV chain (*Hz*);*

R_n : *is the n^{th} correlator that detects a valid chip, varies from $[0,\ (R - 1)]$;*

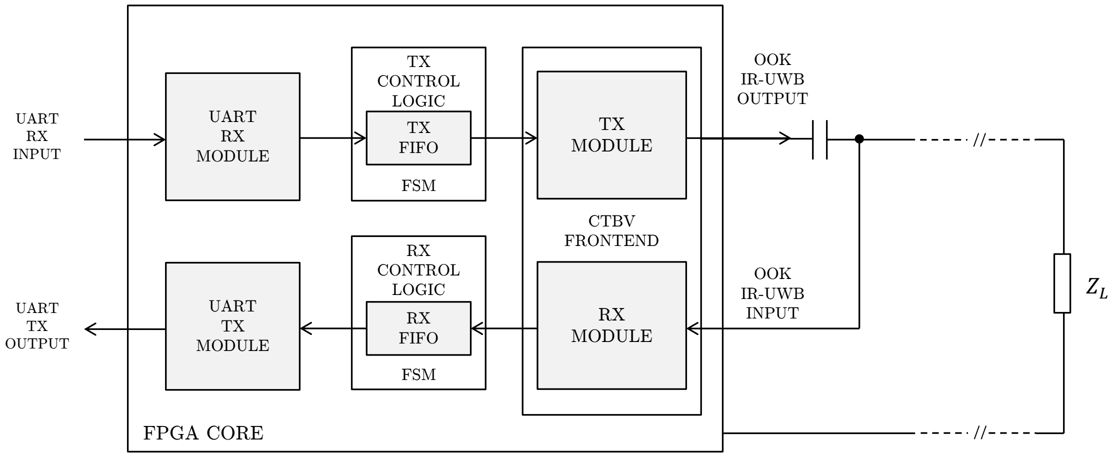


Figure 3.6: Functional Block Diagram of the FPGA in the Ranging configuration: the TX module output is connected to a load, Z_L, in parallel with the RX module input. The series output coupling capacitor can be removed if the pin supports high-impedance mode after transmission. Depending on load impedance, consecutive reflections from the load and from the transmitter side will be captured in the RX module. Impedance matching circuitry on the TX side can prevent reflections on the TX pin if desired.

$RXFIFO_{depth}$: *is the maximum number of FIFO words, or samples, to be stored in the RX FIFO;*

$RXFIFO_{size}$: *is the maximum number of bits allocated to the RX FIFO (*bit*);*

$RXFIFO_{word_{size}}$: *is the size of the RX FIFO's words (*bit*);*

N : *is the length of the Gold or Kasami code;*

R : *is the number of Rake correlators, or fingers, in the RX module.*

3.3.2 Transceiver Configuration

The Transceiver configuration is a way of establishing bidirectional or multi-directional communication with identical systems by means of an IR-UWB transmission channel (Figure 3.7). For this configuration, based on a set of $N+2$ Gold codes, this many number of transceivers could be operating in a single channel. The UART serial interface provides a way to exchange payload data with an external control system. When the UART RX module receives a UART frame, it signals the TX control logic to transmit the corresponding number of *chips* which represent that binary payload. In the opposite flow, the RX module listens for incoming *chips*. When the RX control logic detects that at least 8 payload bits have been received, a UART frame is sent to the external system through the UART TX module.

In this scenario, the usable bitrate of the transceiver is limited by the maximum bitrate of the UART interface between the UART TX/RX modules and the UART adapter of the external control system. Otherwise, in ideal conditions and not considering channel utilisation ratio, this would be limited by the inverse of the IR-UWB *chip* length.

$$\textit{Payload bit-rate}: \qquad Payload_{\textit{bit-rate}} = chip_{rate} = \frac{1}{\tau_{chip}} \qquad \text{(bit/s)} \tag{3.24}$$

where:

$Payload_{\textit{bit-rate}}$: *is the theoretical maximum bitrate available to the payload stream (*bit/s*);*

$chip_{rate}$: *is the maximum frequency at which chips can be transmitted (*Hz*);*

τ_{chip} : *is the chip length (*s*).*

As there is no data-link layer (OSI model) protocol in place to offer reliable frame transmission in the UART or in the IR-UWB channels, no overhead is purposely introduced in the channel. Nevertheless, reliable communication could be implemented in upper layers by the external control systems at both sides of a transmission line. However, the RX module control logic does contain a frame synchronization mechanism, in the event a *chip* is not detected within the predicted time-frame, to avoid de-synchronizing with the transmitter and provide misaligned frames to the external control system. When a *chip* timeout occurs, the RX control logic waits $W \times \tau_{chip}$, where W is a configurable integer parameter, and discards all *chips* in that timeout interval before attempting resynchronization. Although this mechanism can increase the number of payload timeouts when consecutive *chips* are not detected, it is a very simple way of preventing payload data misalignment in the absence of other built-in techniques such as Cyclic Redundancy Check (CRC), which can still be implemented in upper protocol layers.

As the TX module uses OOK modulation, Equation 3.25 as per [49] defines a way to estimate Bit Error Rate (BER) on an OOK channel.

$$BER_{OOK} = \frac{1}{2} erfc\left(\frac{1}{2\sqrt{2}}\sqrt{\frac{E}{N_0}}\right) \tag{3.25}$$

$$\textit{Error Function}: \qquad erfc(x) = (2\pi)^{-1/2} \int_0^{\infty} e^{-t^2} dt \tag{3.26}$$

where:

$\frac{E}{N_0}$: *is the energy per pulse to noise power spectral density ratio of the OOK pulses in the transmission channel.*

3.4 Conclusions

The description of the system architecture is hereby finalized. This chapter aimed to introduce the main components of the system, their functions and how they interact as a whole. Along the

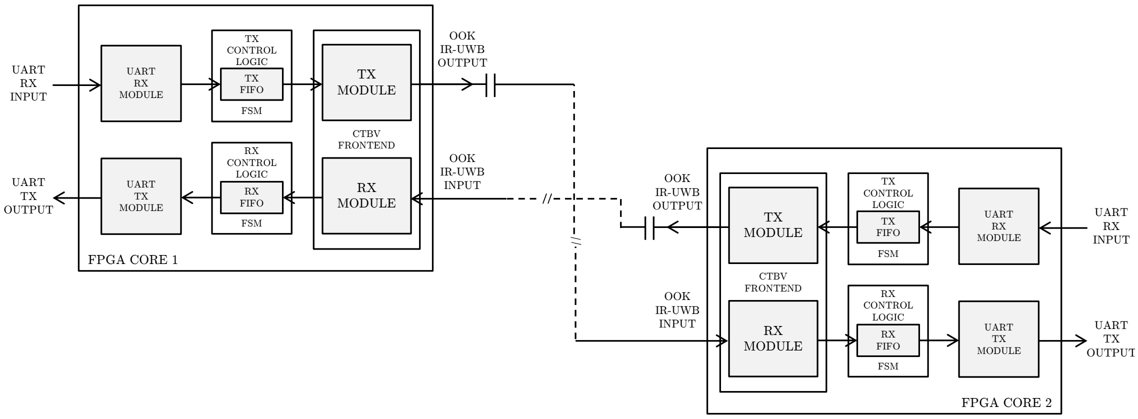


Figure 3.7: Functional Block Diagram of two FPGAs in the Transceiver configuration: in each board, the TX module output is connected to the RX module of the opposite system for bidirectional communication. The series output coupling capacitors can be removed if the pins support high-impedance mode after transmission.

way, the IR-UWB concepts behind the design decisions were explained when justifiable, enabling the reader to not only understand the principles of operation but also to be able to replicate this architecture in a work of its own implementation. Its also significant to note that the TX and RX modules not only work for Gold or Kasami sequences, but for any type of PN sequences, leaving some margin of design personalization without requiring additional architecture changes. Likewise, the generator polynomial algorithm can work for any desired code.

Additionally, two different configurations were introduced and described, Ranging measurements mode and Transceiver mode, making these the two most promising use cases for the proposed design approach.

In the following Chapter, System Implementation on Programmable Logic Hardware, it will be presented an hardware implementation of the system architecture hereby described.

Chapter 4

System Implementation on Programmable Logic Hardware

Introduction

This chapter documents an hardware implementation of the IR-UWB system architecture described in the previous chapter. The first task of the hardware implementation consisted on selecting an appropriate programmable logic platform that could be sufficiently low-cost yet effective in the proof-of-concept demonstrations. Complex Programmable Logic Devices (CPLDs) were considered but most currently available options fall short in number of available resources, available peripherals like RAM and overall logic performance. Field-Programmable Gate Arrays (FPGAs), on the other hand, turned to be a more appropriate choice, as they offer greater performance while maintaining a lot of versatility when compared to Application-Specific Integrated Circuits (ASICs).

In order to achieve a low Bill of Materials (BoM), only relatively less expensive FPGAs could be considered for the prototype. However, low-cost FPGAs typically have lower maximum operation frequencies of up to only a few hundreds of MHz. Yet, UWB systems do require fast sampling rates and, even if clock frequency reduction schemes can be achieved with parallel computing, I/O bandwidth and internal logic timing constraints can possibly still undermine system performance. Taking these limitations into consideration, an equilibrium between performance, cost and prototyping freedom had to be reached. The present chapter describes all the steps of this selection process but also includes how the system was implemented in HDL, tested and validated for correctness, finishing with expected system specifications with the current hardware.

Following this introduction, the information in this chapter is divided into seven major sections:

- **Section 4.1: Hardware Selection**, describes important aspects regarding the selection of the FPGA chip for the proof-of-concept demonstration;

- **Section 4.2: Hardware Components**, lists the components used in the final hardware implementation;
- **Section 4.3: HDL Development**, summarizes the development of Verilog modules which represent the building blocks of the IR-UWB system;
- **Section 4.4: RTL Simulation Results**, introduces how the modules were tested and shows how test-benches were used to visually validate the correctness of waveforms generated by the HDL models;
- **Section 4.5: Hardware Configuration**, describes how the boards were tested for alternative impedance matching scenarios in order to understand if a simpler hardware configuration would affect system performance;
- **Section 4.6: Hardware Performance Specifications**, shows the theoretical performance specifications that are expected from this hardware implementation;
- **Section 4.7: Conclusions**, presents some conclusions regarding the BoM of the implementation, noting that the hardware in use can serve as a baseline benchmark for the expected cost-performance compromises of IR-UWB implementations using the proposed architecture on FPGAs.

4.1 Hardware Selection

When selecting the hardware for the physical implementation phase, several aspects have been considered. One of the most determining factors would be the I/O bandwidth of the FPGA pins. However, when experimenting with a very capable but expensive system the degree of experimentation freedom is somewhat limited. This is usually not a concern when a lower cost equipment is used, specially if redundant units are available for continuing the work in the event hardware damage occurs as a result of the developing phase. From a prototyping point of view, it would be more important then to have several boards available for experimenting than a single and expensive board, as testing different impedance matching circuitry and various hardware configurations would require regular board modifications. This meant that ease of prototyping should not be ignored. As for the most relevant performance specifications, desired features and other important aspects, those under consideration included:

- I/O bandwidth constraints;
- Internal logic timing constraints;
- Embedded fast and constant-delay shift-register logic [1];
- Price;

[1] Including peripherals like DDR memory, HDMI pipelines, PCI Express pipelines or other high-speed serializers, as proposed by Professor José Carlos dos Santos Alves.

- Maximum clock frequency;
- DSP capabilities and built-in Radio Transceivers;
- Number of Logical Elements Resources;
- Power consumption;
- Applicable international export restrictions.

Considering that a trade-off would be necessary between performance, price and freedom of experimentation, the latter aspects have been given more weight in the final decision. It was also considered that a proof-of-concept only in expensive hardware could mean the IR-UWB architecture depicted in this work would miss an opportunity to be put to the test in less favourable conditions and to appeal to a more widespread use. Based on this requisites, FPGA chips from three different manufacturers have been considered and discussed, namely the families:

- **Microsemi Proasic3** family, model A3PE1500 [2];
- **Altera Cyclone II to Cyclone V** family;
- **Xilinx Spartan-6** family, like the model XC6SLX25T for its integrated DDR3 memory controllers and built-in high-speed transceivers.

Altera Cyclone II EP2C5T144C8N board

The choice eventually focused on the re-utilization of a custom-built PCB for a previous coursework that included an entry-level Altera Cyclone II FPGA (90 nm CMOS). As part of this process of selecting an appropriate yet inexpensive programmable logic platform, some initial tests were conducted in order to obtain real-world data regarding timing constraints of I/O pins and delay of internal logic elements on this board. The FPGA in question, an Altera Cyclone II EP2C5T144C8N, was subjected to experimentation with a Synthesis tool from Altera, the Quartus II Web Edition.

The official datasheet suggests that this model's I/O pins are capable of outputting a clock signal at a maximum frequency of 250 MHz when the pins are configured for 3.3 V LVTTL mode. On a select number of pins, up to 350 MHz can be achieved in 2.5 V mode ([47]). This figures are valid for the speed grade (-8) of the available hardware and for optimal load conditions. The 250 MHz rate was verified in laboratory by configuring the board to output an internally generated clock signal from a built-in PLL which used the main 50 MHz clock as reference. If internal routing and resource utilization would not affect device performance, the system should be capable of generating narrow pulses of at least 2 ns in length, therefore also meeting the 500 MHz UWB specification. Nevertheless, the fractional bandwidth specification could also classify the system as UWB, if this figures could not be met, by using up-conversion to displace the baseband signal up to a maximum central frequency as per Equation 2.2.

[2] As recommended by Professor Sérgio Reis Cunha.

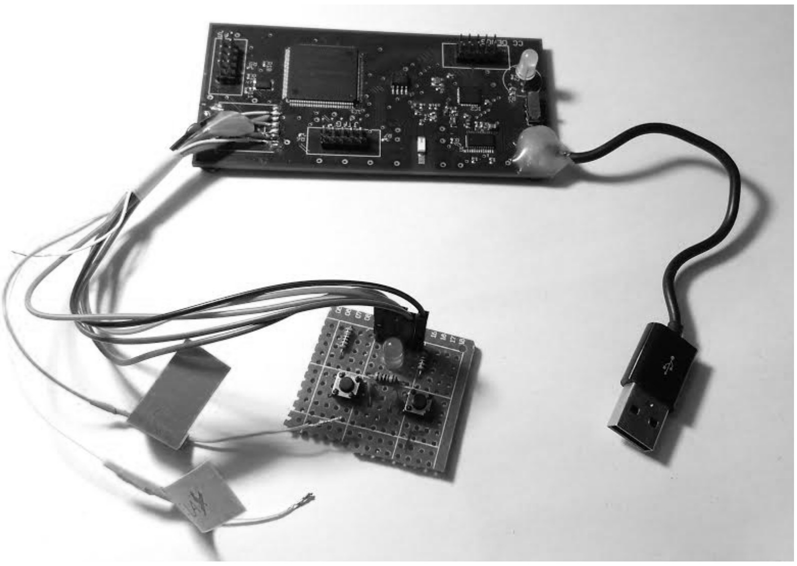


Figure 4.1: Altera Cyclone II based FPGA board: very early prototype which included an additional control board. One push-button could be used to switch between different reference signals to be fed to an output pin: either an external signal, internally bypassed, or a signal from one of the built-in PLLs which used the main 50 MHz clock as reference. The other push-button allowed cycling through various PLL clock speeds. This prototype was used for evaluating if the board could achieve the manufacturer's claims for the maximum clock frequency supported on it's I/O pins. The external control board was later discarded for the proof-of-concept demonstrations as the system could later be controlled via UART.

The board used in these tests can be seen in Figure 4.1.

EP2C5T144C8N Limitations

Although this FPGA chip could potentially reach I/O switching rates of 350 MHz, due to PCB layout limitations it was very impractical to change the I/O banks from a 3.3 V configuration to the required 2.5 V level or to LVDS modes, as it involved re-routing PCB traces leading to several I/O banks. This posed a risk of destroying the traces or possibly damaging the ICs. On top of that, the boards depended on an Active Serial EPCS4 memory to configure the FPGA on power up, and the manufacturer recommends an operating power supply of 3.3 V for this device ([50]). For all these reasons, it was considered best for the boards to continue to operate at the 3.3 V LVTTL voltage level.

Special I/O banks, which are expected be able to reach up to 640 Mbit/s (TX) and 805 Mbit/s (RX) (according to [51]), were also considered. Still, in the provided boards most pins are not accessible as they were not needed when the PCBs were designed. No impedance matching circuitry was in place near the chip's pins either, which was mandatory to ensure proper line termination and enable such high data-rates in LVDS setups.

Taking into account these limitations, this implementation should help establishing a baseline of what can be done even with low-budget FPGA hardware.

Table 4.1: EP2C5T144C8N Device Features ([47])

LEs [(1)]	4,608
M4K RAM blocks [(2)]	26
User-available RAM bits	106,496
Total RAM bits	119,808
Embedded Multipliers	13
PLLs	2
Maximum User I/O Pins	89

[(1)] A LE, or Logic Element, consists of a four-input LUT, a programmable register, and a carry chain, as per [52];
[(2)] 4Kbits plus 512 parity bits.

Early measurements performed on this FPGA indicated that the internal logic could be used to implement buffers as fast as approximately 275 ps, a value averaged from an estimated delay of 35 ns per 128 Verilog *wire* primitives in a linear daisy-chain configuration. This meant that the minimum 2 ns limit imposed by the I/O pins could be achieved by chaining up to 8 buffers of 275 ps (see Equation 3.9). However, when the design complexity and the number of used resources increased, an aggravation of performance was noticeable. Subsequent tests indicated a more realistically achievable pulse length, in the interval of $[6\text{ns},\ \ 8\text{ns}]$ for $K = 12$ or $K = 16$ delay elements, respectively. Although not impressive from the point-of-view of traditional IR-UWB CMOS systems, it was considered sufficient for demonstrating the concept. Notwithstanding, shorter pulse lengths can be used but offer less reliability regarding full *chip* detection.

On top of that, there were several boards available with the same FPGA chip, boards now obsolete after that previous course being completed. This would make the process of developing the prototype a lot faster as several physical configurations could be simultaneously tested. Likewise, there was enough redundant hardware to allow some eventual mishaps along the way. Another advantage found in this chip was the size of embedded RAM: with a total of $106,496$ usable bits that could be accessed at rates up to 250 MHz, common logic resources could be spared from implementing large memory blocks. This memory turned to be very useful when implementing the RX FIFO, giving more than enough Ranging memory depth to allocate the measurements.

Table 4.1 lists the most relevant specifications of the FPGA chip used for this implementation.

4.2 Hardware Components

The implementation uses only bare minimum number of components to support the FPGA chip. Besides the Altera Cyclone II EP2C5T144C8N FPGA, the system requires the already mentioned EPCS4 serial configuration device, a voltage regulator and an external oscillator. The serial configuration device is an Altera proprietary chip that automatically configures the FPGA upon power up, avoiding the need of manual configuration every time the system is turned on, as the FPGA

Table 4.2: Hardware parts list for the proof-of-concept boards. [(1)]

Type of Device	Manufacturer	Model number
FPGA chip	Altera Corporation	Cyclone II EP2C5T144C8N
Active Serial Configuration Device	Altera Corporation	EPCS4N
Dual-Output LDO Voltage Regulator	Texas Instruments	TPS70445
50 MHz MEMS Oscillator	Silicon Labs	501AAA50M0000BAF

[(1)] Excluding passive components, PCBs and connectors. Excluding external components: ALTERA USB Blaster for programming the Active Serial device and USB UART adapters for interfacing with an external control system.

configuration memory is volatile. The PCBs used in this implementation also have a number of peripherals that were not needed, and therefore were not used. A few board modifications were required, however, including a component that had to be discarded from the PCB: a radio module (RFM22B) was mounted on the PCB in a piggyback board configuration and had to be removed in order to free the pads that provided access to FPGA I/O pins. These were the only pins that were easily available and were used to implement TX and RX outputs, as well as other debugging signals, for instance, with LEDs.

Table 4.2 lists the parts list in use which were already populated in the boards.

4.3 HDL Development

The building blocks of the IR-UWB system, including the pulse shaper (TX module) and pulse detector (RX module) illustrated in Figure 3.1, were developed in SystemVerilog, an Hardware Description Language (HDL) commonly used in FPGA development environments. The IDE used was Altera's Quartus II Web Edition v13.0sp1 which proved to be sufficient for the development of this system, although it did lack some high-end features such as manual control over the floorplanning. Those additional features, however useful, did require the purchase of a commercial license. This is the reason why manual Place & Routing was not pursued in order to attempt to obtain additional performance from the system. If this kind of design tools were available, manual optimization of the on-chip location for timing-critical components would yield more control over the system behaviour, but this solution would not be as universal as standalone HDL code.

From the beginning, it was decided that code portability would be important to provide the option to test the design on different FPGA families and on more capable hardware. Lack of modularity also affects portability, and so the whole IR-UWB system was also designed in a very modular approach. Considering that the Delay Buffer, TX and RX modules only use common Verilog primitives, these can be instantiated by any other Verilog project. However, it is not always possible to assure fully portable designs due to the need of using peripherals that require proprietary IP cores, like RAM space. When that is the case, those modules will need to be replaced by IP cores with equivalent functionality in the FPGA development environments of the corresponding manufacturers. A characterization of the modules and a description of the modularity and portability of the design is given in the following items list.

- **TX/RX modules**

  The TX and RX modules can work separately and independently, although both the TX and RX modules require the Delay Buffer module for implementing their CTBV chains. These modules are also independent from the architecture of their control logic modules and can be easily ported and instantiated in other HDL designs. In section 4.4, test-bench results for Ranging and Transceiver scenarios demonstrate how these modules behave when integrated into a IR-UWB toplevel module that is subject to test-bench stimuli.

  The RX module suffered a few redesigns in order to increase performance and save logic resources by simplifying the receiver implementation. In the very beginning of the work, this module did not output the $XORs'$ logic states but instead attempted to do sequence identification by performing combinational sums of the correlators' outputs. This solution was valid for smaller codes, of $N < 15$, but proved problematic when the length of the codes was increased. Besides the performance issue, the FPGA chip could not entirely allocate the logic needed for larger codes like $N = 31$, due to lack of logic elements.

  In the final design, the TX module was configured for $K = 16$ delay buffers, generating pulses of about 8 ns in length and achieving a maximum of approximately 4 ns of timing resolution in the RX module. Tests with $K = 12$, or 6 ns pulse length also work but code detection sensibility is affected. These figures are valid for both the Ranging and Transceiver configurations.

- **TX/RX FIFOs**

  The RX and TX FIFOs were implemented by instantiating an Altera's IP core module, `dcfifo_mixed_widths`[53]. This IP core was the best solution to provide reliable data storage in the system as it implements a FIFO in the on-chip RAM, guaranteeing speed and integrity with the parity bits. More over, it allowed to save resources for other modules that can only be implemented by common logic elements. Because this particular FPGA model has generous amounts of RAM for the intended applications, this meant no trade-off had to be made regarding ranging data depth for maximum ranging distance. Instead, about half the RAM on the device was used to perform RX module buffering before transmitting the correlators' data through the UART TX module: precisely 256 samples at 20 ns intervals, or about 5 µs worth of ranging data for a theoretical maximum ranging distance of around 500 m and for a VF of 0.66, as per Equation 3.15. It shall be noted, however, that if porting the control logic of this IR-UWB system to FPGA families of other manufactures, these FIFOs must be implemented using their own proprietary IP cores or other custom solutions as the system was not designed to work without the RX FIFO module.

- **UART modules**

Table 4.3: EP2C5T144C8N resource utilization per configuration

Logic Resource	Ranging (N=15)	Ranging (N=31)	Transceiver (N=15)
Total LEs [(1)]	1,867 (41%)	2,563 (56%)	2,897 (63%)
> Combinational functions	1,806 (39%)	2,512 (55%)	2,812 (61%)
> Dedicated logic registers	229 (5%)	230 (5%)	317 (7%)
Total RAM bits [(2)]	65,536 (55%)	65,536 (55%)	55,552 (46%)

Valid for $K = 16$ delay elements. Percentages relate to the total resources available, according to the type of resource used.

[(1)] A LE, or Logic Element, consists of a four-input LUT, a programmable register, and a carry chain, as per [52];

[(2)] The value can be estimated by solving Equation 3.21 for the size of RX FIFO. In this implementation, all configurations are set to use the same amount of RAM, exactly 32 bits per correlator independently from code length used. The additional bits used in the Ranging configurations are reserved for a FIFO serial number from 0 to 255.

The UART functionality was obtained from an open-source third-party, [48], and its components were instantiated from the toplevel system module, as it was out of the scope of this dissertation to focus on developing such modules.

- **Toplevel module**

  The toplevel model is the core of the system and instantiates all the other components. It provides an interface for the RX,TX and FIFOs' modules to interact and is also responsible for implementing their corresponding control logic. This is achieved by using common Verilog code which results in the implementation of finite state machines in common Verilog compilers. Therefore, this module is fully portable as long as any non-supported sub-modules can be replaced by similar ones with identical functionality, namely the RX and TX FIFO modules.

The HDL source code for the TX Module, RX module and Delay Buffer are included in Appendix C: Source-Code. Table 4.3 summarizes logic resource utilization for both configurations.

4.4 RTL Simulation Results

The development of any HDL modules requires formal verification and simulation to test if the HDL code properly describes the desired functionality of the system. Before porting the design to FPGA hardware, in which it's a lot harder to identify HDL implementation faults, the RTL simulations were an important part of the design debugging phase. Initially, the simulations' code coverage focused in evaluating if the TX and RX modules could generate and detect CTBV pulses separately. Then, the system as a whole was also tested for both Ranging and Transceiver configurations. Although the TX and RX modules are purely combinational and rely on internal logic's propagation delays to work as expected, their functionality could still be tested on simulations in which those delays were also introduced by the simulator software. The HDL simulation environment used was ModelSim Altera Starter Edition v10.1d from Mentor Graphics.

The simulation results presented below can demonstrate how the design was debugged and validated before being deployed to the actual hardware. Test-benches for both configurations produce a visual representation of the internal signals in a waveform logic plot when the system is stimulated by control signals. This virtual logic analyser allows to visually inspect the waveforms and verify if their patterns match the expected behaviour of the system.

4.4.1 Ranging Configuration

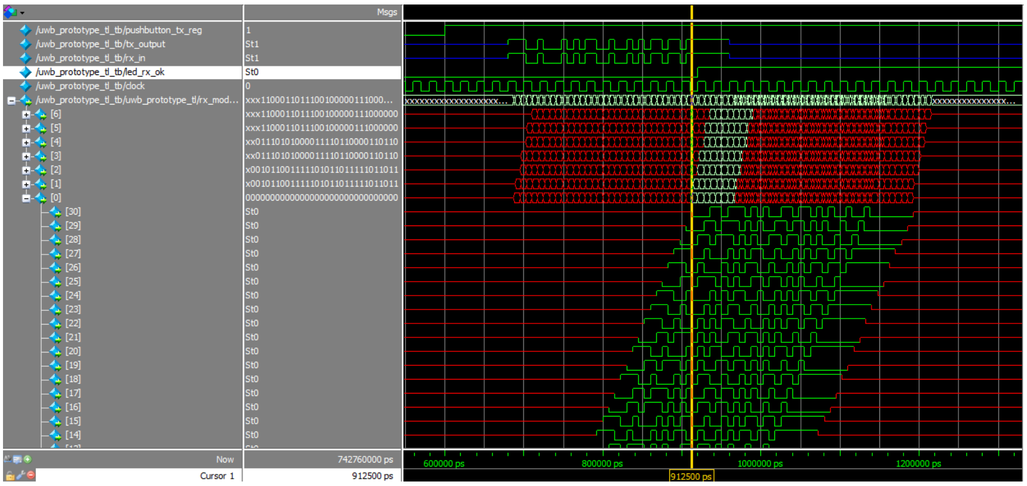
Under normal operation, the Ranging measurement begins when a command is received by the UART RX module instead. This command, $8'hAC$, is sent by an external control system. In other to prevent cluttering the waveform plot with UART signals of a module that is known to work from previous tests, the Device Under Test (DUT) has a special input solely for testing scenarios that overrides the UART modules.

In this configuration, the test-bench simulates a send command, *pushbutton_tx_reg*, which triggers the TX module control logic into operation as if a command had been received from the UART RX module. The TX module begins to generate a CTBV signal that propagates to its output and RX module' input, as both inputs are externally shorted together for measurement synchronization purposes. The correlators in the RX module begin to toggle rapidly as the received signal goes through the RX CTBV chain, and a valid code is detected in a given correlator when all its $XORs'$ outputs are equal to *zero*. This state only lasts for around half the pulse length, or 4 ns in the example of Figure 4.2. Nevertheless, by design, the next rising edge of the main 50 MHz clock should coincide with a period of time when at least one of the correlators is in this state as all cycle through the same logic values in a time-shifted arrangement. When sampling all correlators in the clock rising edge, an independent RX module comparator circuit identifies at least one correlator with a valid code detection and signals this to its toplevel control logic, which later enables the *led_rx_ok* output. This signal stays at logic level high until it is reset by its own counter, allowing the detection to be perceived by inspecting an LED when the system is working on FPGA hardware.

4.4.2 Transceiver Configuration

The Transceiver configuration works in a slightly different way when compared to the Ranging configuration. Instead listening for a UART command and sending one *chip* at a time, the UART RX module feeds 8 bit of incoming UART data to the TX module control logic. Next, the control logic instructs the TX module to transmit 8 *chips*, one for every UART payload bit.

In order to simulate this behaviour, besides the *pushbutton_tx_reg* special input, the DUT also contains a 8 bit input bus to receive a UART test pattern and simulate an incoming payload data.



	Msgs
/uwb_prototype_tl_tb/pushbutton_tx_reg	1
/uwb_prototype_tl_tb/tx_output	St1
/uwb_prototype_tl_tb/rx_in	St1
/uwb_prototype_tl_tb/led_rx_ok	St0
/uwb_prototype_tl_tb/clock	0
/uwb_prototype_tl_tb/uwb_prototype_tl/rx_mod...	xxx11000110111001000001110000...
[6]	xxx1100011011100100000111000000
[5]	xxx1100011011100100000111000000
[4]	xx01110101000011110110000110110
[3]	xx01110101000011110110000110110
[2]	x0010110011111010110111110110 11
[1]	x0010110011111010110111110110 11
[0]	0000000000000000000000000000000
[30]	St0
[29]	St0
[28]	St0
[27]	St0
[26]	St0
[25]	St0
[24]	St0
[23]	St0
[22]	St0
[21]	St0
[20]	St0
[19]	St0
[18]	St0
[17]	St0
[16]	St0
[15]	St0
[14]	St0
Now	742760000 ps
Cursor 1	912500 ps

Figure 4.2: RTL simulation waveforms of the Ranging configuration. Upon receiving a send command (*pushbutton_tx_reg* in the test-bench) the TX module begins to generate a CTBV signal that propagates to its output and to the RX module's input. The correlators in the RX module start to toggle rapidly and a valid code is detected in the first correlator. The RX module control logic identifies that a valid code detection has occurred and signals this through the *led_rx_ok* output. The Gold code used was *31'b1110 0100 0101 0110 0100 1010 1101 101* and the transmission is MSB first. A full-page version of this waveform plot is available in Appendix D: RTL Simulation Waveforms.

Following the load of the test pattern on the 8 bit bus, the test-bench issues a send command, *pushbutton_tx_reg*, which triggers the TX module control logic to instruct the TX module to transmit the first *chip*. The TX module begins to generate a CTBV signal that propagates to its output. In Figure 4.3, the TX module output is connected to the RX module input in order to demonstrate the receiver side. The correlators in the RX module begin to toggle rapidly as the received signal goes through the RX CTBV chain, and a valid payload bit *zero* is detected in a given correlator when all its $XORs'$ outputs are equal to *zero*. When all the outputs equal *one*, a valid payload bit *one* is also detected. This process repeats until all 8 payload bits have been sent by the TX module. The RX module control logic signals a valid payload bit *zero* through the *led_rx_ok*_0 output and a valid payload bit *one* on *led_rx_ok*_1. Although not used for simulations, detection thresholds can be set in the RX module's control logic to allow an arbitrary number of misaligned bits in received *chips*.

4.5 Hardware Configuration

In the final hardware line-up, two boards were still in use of a total of five that were experimented with. Both could be used for the Ranging and Transceiver configurations, as the two had their TX module's output shorted to their own RX module's input. This allowed the boards to be tested in both applications, which was useful in an attempt to identify the best impedance matching alternative for each. Because the FPGA chip already contains an on-chip series termination resistor,

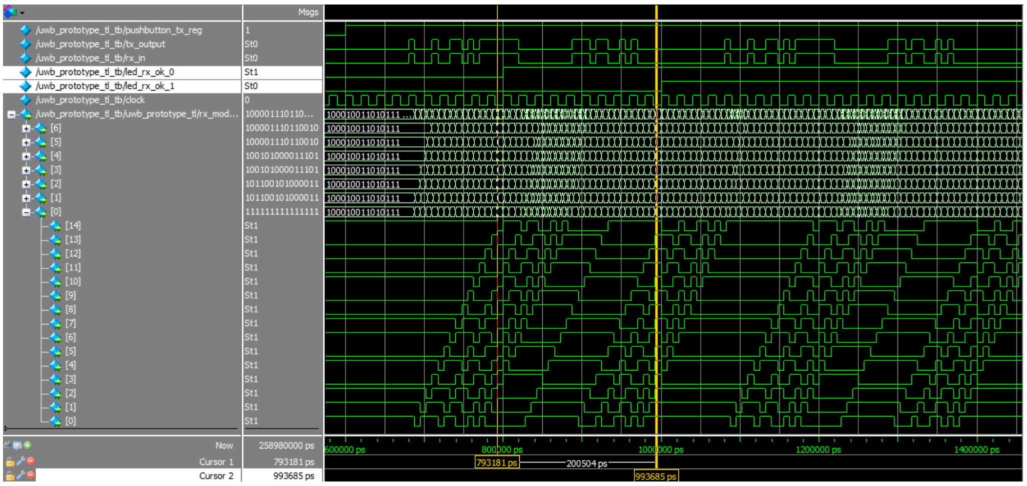


Figure 4.3: RTL simulation waveforms of the Transceiver configuration. Upon receiving a send command (*pushbutton_tx_reg* in the test-bench), the TX module control logic instructs the TX module to begin transmission of a *chip* according to the first of eight payload bits, as defined by an UART test pattern loaded by the simulator into the Devide Under Test (DUT). The correlators in the RX module start to toggle rapidly and the test pattern payload bits are sequentially detected, as indicated by *led_rx_ok_*0 and *led_rx_ok_*1 signals. The Kasami codes used, of length $N = 15$, were: *15'b1000 1001 1010 111* for payload bit *zero* and *15'b0111 0110 0101 000* for payload bit *one*. The UART sequence was *8'hAA*, or *8'b10101010*, and the transmission of the *chips* is LSB first. A full page version of this waveform plot is available in Appendix D: RTL Simulation Waveforms.

enabled by software, impedance matching could be done with a simple Series Termination scheme. Impedance matching with Series Terminations have low power losses and can help damping overshoot and undershoot, reducing EMI and line noise, but the signal's slew-rate is affected. As such, one of the boards was populated with a series impedance resistor of 50 Ω, between the TX output pin and the line, and the on-chip series termination resistor of 25 Ω was enabled. The other board did not include this adaptation and drove the transmission line directly. Figure 4.4 illustrates the TX output configuration of both boards. The hardware was assembled in two different ways to test the hypothesis that improved performance could be obtained from improving impedance matching between the TX module output and the transmission line itself.

In order to test the two boards for the Ranging configuration, a Matlab script was developed to collect ranging data while the generated waveforms were sampled from the load end of the transmission line. It was later concluded that both boards behaved very similar in this scenario, indicating that the unmatched board could actually be already closely matching the line's impedance. The source-code for this script is available in Appendix C: Source-Code and a demonstration of its functionality is explored further in Chapter 5. For the Transceiver tests, a UART terminal was open and a some UART frames were sent by the external control system to the boards under test. The corresponding generated waveforms, sampled from the load side, also proved to be very similar for both the series terminated board and the mismatched board.

Taking into account the obtained experimental data, a possible explanation for these results is extracted from an Altera's Application Note [54] on output impedance of pins configured for nominal current strength instead impedance matching. The document states that, for their high-end Stratix devices, an output driver's equivalent resistance is given by Equation 4.1. If the same holds true for the Cyclone II series, this means the actual impedance of the unmatched board is very close to the desired target of 75 Ω, as it is configured for maximum drive strength ($24mA$) and operates at 3.3 V LVTTL. These figures yield a value of 68.75 Ω equivalent impedance. If the approximation is valid, the impedance of the TX output pin in the mismatched board is closer to the line's impedance than what was previously thought.

$$R_{driver} = \frac{V_{CCIO}/2}{I_x} \qquad (\Omega) \qquad (4.1)$$

where:

R_{driver} : *is the resistance of the output pin's driver (*Ω*);*

V_{CCIO} : *is the voltage level at which the pin operates. (*V*);*

I_x : *is the current provided by the driver to the load. (*A*) This current is equal or larger than the configured drive strength.*

Considering this reduced mismatch possibility, in the non-terminated board the reflection coefficient between the output pin and the transmission line is limited to 4.3 % and the mismatch loss to less than 0.01 dB (see Equations 3.11 and 3.12). In the event of an higher mismatch loss during transmission, the power transfer from the TX module output to the transmission line could be affected by reflections. This would limit the reflected power from the open load at the other end of the transmission line to the RX module input and compromise code detection. However, measurements presented in the next chapter suggest that losses are negligible for the first load reflection, meaning that the Series Termination scheme is optional for the current FPGA chip and transmission line impedance.

When performing Ranging measurements, after the *chip* is transmitted, all ends of the line are put in an high-impedance situation, enforcing multiple reflections from each side as experimental measurements shows in Chapter 5: Tests & Results. High-impedance mode also prevents the TX pin driver from being damaged with forced voltage swings caused by reflections from the load. Even though reflections are good for Ranging, the Transceiver mode works best if the signals are not reflected back by the load or by the TX output, as 8 sequential *chips* are transmitted at a time whenever an UART frame is received. In the event of load reflections, the last *chips* could be attenuated by reflections at smaller line distances, and as such, in Transceiver mode the TX output pin is never kept in an high-impedance state.

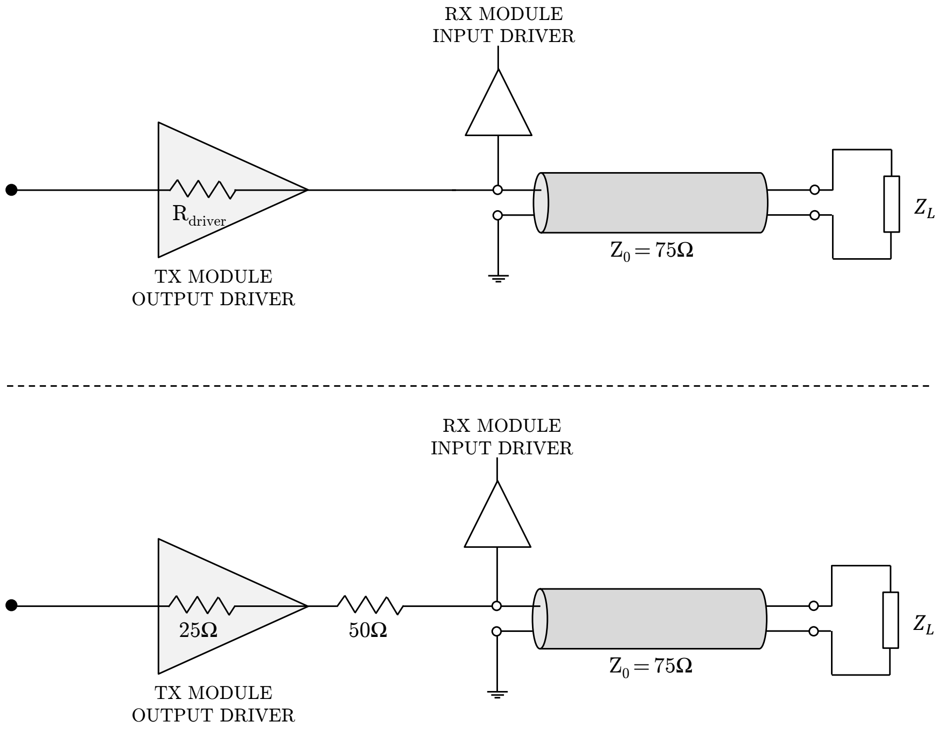


Figure 4.4: Functional Block Diagram of the TX module driver's impedance: the diagram on top of the middle dashed line represents the board with a TX driver configured for a maximum current strength of 24 mA. The diagram on bottom illustrates the board with the on-chip series termination resistor enabled in software plus an additionally 50 Ω external resistor close to the driver's output pin, in an attempt to match the impedance of the transmission line. The RX module driver's input is an high impedance load.

4.6 Hardware Performance Specifications

The expected, or theoretical hardware performance specifications of the present hardware implementation are presented in this section as a baseline of comparison for the experimental results presented in the following chapter. Table 4.4 provides a characterization of the general specifications when operating the aforementioned boards.

4.6.1 Ranging Configuration

In the Ranging configuration, a *chip* is transmitted and the TX output pin is put in high-impedance mode. Simultaneously, the system immediately begins sampling the correlators' states in the RX module in order to capture this *chip* for establishing a timing reference. The correlators continue to be sampled until reaching the RX FIFO is completely full. When a transmission line with no load termination is connected to the TX output, the *chip* is reflected on the opposite line of the transmitted line and is detected by the correlators which were previously sampled. Because each sample is stored in the RX FIFO and tagged by a sample serial number, which is a counter from

Table 4.4: FPGA board performance specifications on available PCBs

I/O mode	3.3 V LVTTL
I/O output clock toggling (max.)	250 MHz
I/O current strength (max.) (3.3 V LVTTL)	24 mA
Output pin's impedance [1] (3.3 V LVTTL)	25 Ω
Available I/O pin pads [2]	6
FPGA main clock speed	50 MHz

(1) Series on-chip termination can be enabled in Altera's Quartus II Pin Planner to prevent reflections and help maintain signal integrity[47]. When the series on-chip termination is enabled the current strength setting is overridden. For the proof-of-concept tests, one board has been assembled without series termination and other board was configured with this setting enabled and assembled with an additional 50 Ω resistor, between the TX output pin and the type F connector, in an attempt to match the impedance of the coaxial cable used of 75 Ω.
(2) Not all PCBs had this number of accessible pins as some pads were damaged during the hardware prototyping phase. Excluding UART pins.

0 to 255, its possible to properly identify the timing instant at which each sample was taken based on delta timing estimations.

In Ranging mode, the $XORs'$ outputs are sampled but no correlator sums take place, limiting the number of resources and allowing the design to be synthesizable for larger codes. In order to process the data stored in the RX FIFO, a Matlab script is used in an external control system that receives the ranging data through an UART interface and plots it in the screen. The plot allows for visual identification of line propagation delay and an estimation of load distance. This is typically achieved with an open transmission line, an infinite load.

In optimal conditions, given a pulse length of 8 ns, and for a velocity factor of 0.66, the minimum load distance would be around 24 m for $N = 15$, or 49 m for $N = 31$ (see Equation 3.14). Calculations in Chapter 5: Tests & Results suggest a greater velocity factor than 0.66, causing a slight increase of these values. Therefore, only codes with length up to $N = 31$ can be used with the current board on the same line. Under the same conditions, the corresponding maximum load distance is just over 500 m for both code lengths (see Equation 3.15).

4.6.2 Transceiver Configuration

In Transceiver operation, the system continuously operates as an UART to IR-UWB bridge. The RX CTBV chain is permanently sampled and the RX control logic performs sums of the correlators outputs. When up to 8 valid payload bits are detected, this data is sent through the UART TX module.

The maximum speed of the transceiver is limited by, and equal to, the UART baud-rate. Typical USB to UART adapters offer rates up to 1 or 3 Mbaud/s. Despite the 1 Mbaud/s rate being supported by the USB adapter used and by the UART modules, the UART interfaces were configured for 250 kbaud/s (2 Mbit/s payload equivalent) to minimize the chance of UART errors affecting UART payloads and bit error rate estimation. In optimal conditions, and not considering misdetection events or errors, the system could be able to theoretically achieve about 8 Mbit/s (1 Mbaud/s), given the 8 ns pulse length, a code size of $N = 15$ and a minimal guard interval between consecutive *chips*, as per Equation 3.25. The actual performance, as measured in Chapter 5: Tests & Results, indicates a TX module baud-rate of at least 687 kbaud/s (5.5 Mbit/s), which is still sufficiently fast for discarding the use of the UART TX FIFO at the current UART data-rate.

4.7 Conclusions

Since the beginning of the design and later on the hardware implementation phase, it was desired to limit the number of external components by choosing to do as much of signal processing as possible in programmable logic. When compared to more expensive and high-end design approaches which make use of complex RF front-ends with state-of-the-art ADCs and high-end FPGAs, some performance loss was expected. But those systems can impose a BoM of several hundred *euro*, sometimes thousands of *euro* for the FPGA boards alone such as the Xilinx Virtex-II XC2VP70 in [9].

That's not the case with this approach: only four main components are needed, as per Table 4.2. On top of that, the FPGA chip alone only costs about € 15 in single quantities, at major component retailers and as of January 2016. The Cyclone II family does work best with an Active Serial configuration device, EPCS4, with increases the cost by another € 15. Yet, the main clock oscillator, the passive components and even the voltage regulator can be obtained for only a fraction of the cost of the main components. Still, the main contribute of this hardware implementation is that the proposed IR-UWB system architecture is not exclusive to expensive hardware, although better timing performance and shorter pulse lengths could be obtained from more capable FPGA chips.

In the next Chapter, Tests & Results, this hardware implementation is put to the test under both configurations and the system specifications are measured in actual usage scenarios.

Chapter 5

Tests & Results

Introduction

Following the hardware implementation described in the previous chapter, the IR-UWB system architecture was evaluated in two proof-of-concept demonstrations, and its performance was measured so that relevant system specifications could be obtained. The data generated from these tests is presented in the current chapter.

Before introducing the test results, the procedures used for device characterization are also described so that all test scenarios are known in advance.

For reference, is shall be noted that the experiments conducted in this work did not use dedicated RF amplification neither the signals were purposely radiated from adapted antennas. Instead, coaxial cables were used as a transmission medium. As such, no measurements were taken to evaluate radiated energy from the system, which is believed to fall below the regulatory thresholds, as FPGA chip manufacturers are complied to certify their designs with FCC, ETSI and other agencies.

Following this introduction, the information in this chapter is divided into five major sections:

- **Section 5.1 : Testing Procedures**, describes how the boards were tested for both configurations;

- **Section 5.2 : Proof-Of-Concept Demonstrations**, presents real data and performance specifications obtained from both the Ranging and Transceiver configurations;

- **Section 5.3 : Summary of System Specifications**, summarizes the most relevant system specifications for both configurations;

- **Section 5.4 : Conclusions**, provides the final remarks regarding the hardware implementation of the IR-UWB system;

- **Appendix A : Complement to Tests & Results' Chapter**, includes Ranging data, frame and bit error information and waveforms of the generated pulses.

5.1 Testing Procedures

In this section, the testing procedures used for evaluating system functionality and performance are described for both the Ranging and Transceiver configurations. Two identical boards yet with different output driver's impedance (see Figure 4.4), were used in these configurations:

> **Board 1**: TX output driver configured for maximum drive strength (24 mA).
>
> **Board 2**: TX output driver configured for matching the impedance of the transmission line (75 Ω). Includes an on-chip series terminated resistor and an external resistor.

As for the measurement equipment used, the generated signals were sampled with an HP 16500A Logic Analysis System Mainframe populated with a 1 GSa/s DSO module. The probes used were UNI-T's UT-P05 model, a set of passive 200 MHz probes that were set to 10× attenuation mode.

Regarding signal integrity, it should be noted that generated signals are longer than the expected *chip* length as a result of the implementation of a safe-guard period to prevent the signal being affected by unexpected transient events at the end of transmission. Such transients can appear simply from switching the output mode to high-impedance. This safe-guard period can be observed in the "brick-wall" effect at the end of *chips* terminating in logic level *high*. The logic level is not forced on the output, it is the state of the last CTBV chain's *XOR* before the chain is reset for the next transmission.

5.1.1 Ranging Configuration

The tests for the Ranging configuration focused on measuring the propagation delay of a transmission line feeding an open load. This transmission line, visible in Figure 5.1, was known to have a length of approximately 60 m. By visual inspecting the cable coil, an estimation of the cable length returned 65 m, which is close to the expected value. This estimation was done using a very rough approximation, described in Equation 5.1, in which cable diameter and coil width were not considered.

More relevant than line length, the actual propagation delay of the cable was measured by Professor Sérgio Reis Cunha, using a Rohde & Schwarz ZVL Network Analyzer, to a value of approximately 254 ns. This delay was the most relevant metric for validating the Ranging tests, as only an estimation of cable length could be obtained and the exact Velocity Factor (VF) ratio of the medium was also unknown. Still, based on a 60 m length, the VF of the line would be 0.788 (see Equation 5.2), a figure close to the VF of some coaxial cables with nominal impedance of 75 Ω. This estimation only provided an order of magnitude for comparison when later estimating the distance of the transmission line with the IR-UWB system, and should not be taken as accurate as the measured propagation delay.

Ranging Test Procedure

After establishing a comparable and appropriate metric for evaluating how the system performed, and after connecting the board's UART pins to a laptop's USB-UART adapter, the testing scenario for the Ranging configuration consisted in the following steps:

1. Configure the FPGA chip for Ranging operation with a known PN sequence polynomial;

2. Attach the TX output connector to the transmission line, while leaving the other end of the line in high-impedance in order to encourage signal reflections ($Z_L = \infty$ in Figure 3.6);

3. Execute the Matlab script that performs ranging measurements and analyse the detected *chips*; the measurements are also automatically saved on a file by the script for later analysis (in an ideal CTBV chain of constant delays between elements, the best delta measurements take place between two different instants in time of different correlators with full *chip* detection);

4. Attach a DSO probe to the high-impedance (*Hi-Z*) side of the transmission line, set the DSO trigger and re-execute the Matlab script; analyse the waveforms on the DSO screen and document what is being observed;

5. Repeat this process for different code lengths and for both boards in order to verify if the impedance matching circuitry in one of the boards could improve performance.

$$Cable\ Length: \qquad L = N_{winds} \times 2\pi \cdot r \qquad (\mathrm{m}) \tag{5.1}$$

$$Velocity\ Factor: \qquad v_P = \frac{L}{\tau_L \times c_0} \tag{5.2}$$

where:

N_{winds} : *is the number of winds of cable in the coil;*

r : *is an approximation of the coil radius (*m*);*

v_P : *is the Velocity Factor or wave propagation speed in the medium;*

τ_L : *is the propagation delay of the transmission line (*s*).*

c_0 : *is the speed of light in vacuum (*m/s*).*

5.1.2 Transceiver Configuration

The Transceiver configuration was tested for misdetection faults that could lead to payload errors. Because the payload data is actually formed from 8 bit long UART payloads, the transmitting system generates 8 sequential *chips* and the receiving system holds until 8 *chips* are collected, before forwarding the corresponding payload data to its UART TX module. Although timeout counters are in place to let the system recover from misdetection events, there is no frame control protocol and, therefore, synchronization faults might occur. Nevertheless, it is always possible to add further transmission reliability by using an higher level transmission protocol on the external control system as with any traditional UART interface.

Due to design complexity, the code length used in Transceiver tests were limited to $N = 15$. Although the hardware had enough logic elements to fit Gold codes of lengths $N = 31$, device utilization soared to 93 %. When tested for payload errors, the number of misdetection events was to high, making the use of lengthier codes impractical with the current hardware in this configuration.

Regarding BER estimation, as no UART CTS/RTS pins were available for flow control, these synchronization signals were not used. UART baud-rate has been set to 250 kbaud/s, a lower speed than the maximum rate supported by the adapter, to prevent UART errors from affecting this estimation.

Transceiver Test Procedure

After connecting the board's UART pins to a laptop's USB-UART adapter, the testing scenario for the Transceiver configuration consisted in the following steps:

1. Configure the FPGA chip for Transceiver operation with a known PN sequence polynomial of length $N = 15$ and the corresponding polynomial for the same but inverted PN sequence; this set of code and inverted code can be used to detect payload *zero chips* and payload *one chips* with a single RX module, by adding the correlators binary outputs together as explained in 3.2.2;

2. Attach a DSO probe to the high-impedance side of the transmission line; open an UART terminal on the external control system and transmit a binary payload; analyse the waveforms on the DSO screen and document what is being observed;

3. Execute the Matlab script which performs transmission of UART payloads with binary test patterns from 0 to 255 and estimates the BER after a configurable number of test events is completed;

4. Repeat this process for both boards and end with a board-to-board test, by connecting the USB adapter's UART TX line to one board and the UART RX line to the other board. Alternatively, the second board can be configured to reply any received *chip* by simply shorting its UART TX and RX pins.

5.2 Proof-Of-Concept Demonstrations

In order to evaluate how the IR-UWB system performs in a real usage scenario, on the current hardware implementation, this section introduces the results obtained from applying the aforementioned testing procedures. For the purpose of these demonstrations, appropriate software had

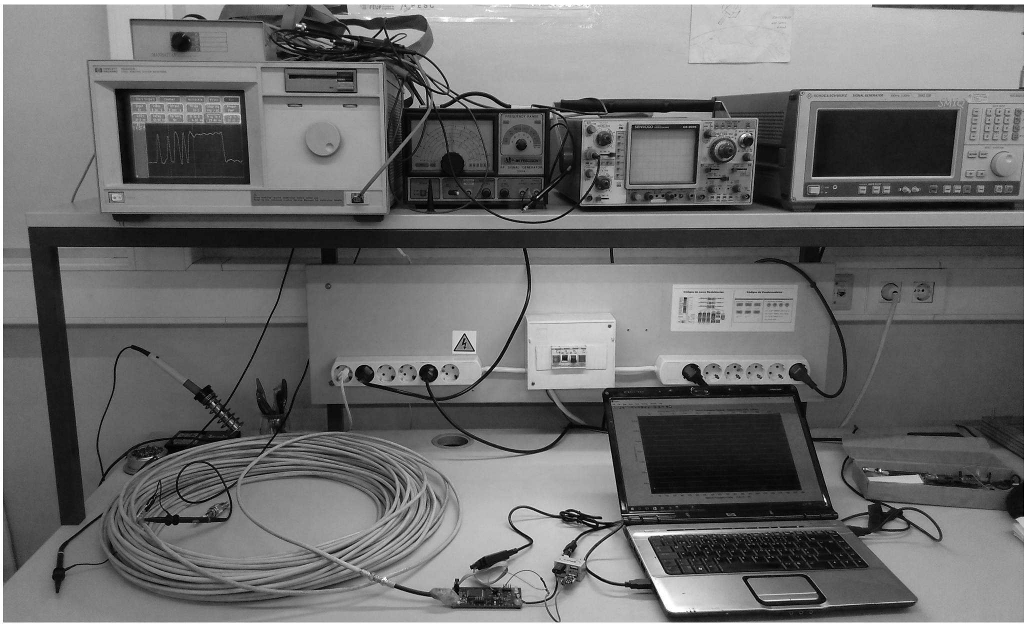


Figure 5.1: CycloneII-based FPGA board: proof-of-concept testing scenario. On the top left corner, the HP 16500A 1 GSa/s scope shows a waveform from the latest capture, taken from the load end of the transmission line right below it. In the middle, the FPGA board is connected both to the transmission line and to a laptop that serves as the external control system and receives ranging data. The data is then processed by a Matlab script, available in Appendix C, and shown on the laptop screen for analysis.

to be developed for the external control system. As any modern personal computer can interface with UART devices through USB to UART adapters, a laptop was used as the external control system. Matlab was selected as the programming environment of choice for its useful graphing capabilities, while still being universally available to common operating systems, and also for supporting UART communication.

5.2.1 Ranging Configuration

The Ranging configuration was subject to a range of tests regarding variable conditions of code lengths ($N = 15$ or $N = 31$), TX output pin's impedances (*Hi-Z* or forced logic level *low*, *0-Z*) and pulse lengths (K delay buffers with chains of 16, 12 and 8 elements with approximate delays of 8 ns, 6 ns or 4 ns). Chains of less than $K = 8$ elements did not yield usable results in Ranging operation. These tests were applied in the two boards with different output driver's impedance settings.

Ranging measurements made by processing correlator data with Matlab and waveforms obtained from the DSO are presented here for comparison. DSO measurements were taken at the load end of the transmission line. The plots are adjusted for the calculated VF of the transmission line, as estimated at 5.1.1, and compensated for the measured pulse lengths. For instance, a *chip* generated with delay buffers of 16 elements, or about 8 ns delay, was measured to last 110 ns/15

Table 5.1: Board Performance in Ranging Measurements [1]

K buffers τ_{pulse}	4 2 ns	8 4 ns	12 6 ns	16 8 ns
Board 1	—	Poor	OK	OK
Board 2	—	—	Unusable	Poor

[1] Qualitative classification based on full *chip* detection and valid for $N = 15$.

as per Figure A.6. All samples stored on the RX FIFOs were taken at 20 ns intervals and are represented by the dark dots in the Matlab graphs. Each graph shows the content of 7 correlators, each correlator baseline where all its $XORs'$ outputs equal *zero* and a threshold line per correlator for 1 misaligned bit in incoming *chips*.

Regarding the observed performances of the two boards, no measurement advantage was observed with Board 2, when compared to Board 1, in Ranging operations. In fact, Board 1 offered increased performance over Board 2 and could generate and detect pulses up to 6 ns, almost achieving complete *chip* detection around the 4 ns mark. For these reasons, Board 1 holds the recommended hardware configuration for Ranging measurements. See Table 5.1: Board Performance in Ranging Measurements, for a qualitative comparison between the performance of the two boards for different values of K delay elements.

Ranging with a Kasami sequence of $N = 15$

The Kasami sequence of length $N = 15$, *1000 1001 1010 111*, about half as long as a Gold code of $N = 31$, generated the best Ranging results. The length of the Kasami sequence associated with pulse lengths of up to 8 ns proved to be sufficiently short for the length of the transmission line and both boards could identify the *chips* when reflected by an open load. Yet, Board 1 outperformed Board 2 in the Ranging Configuration.

Plots of Ranging measurements:

- Figure A.1: shows a Matlab plot with Ranging measurement's data from Board 1, at $K = 16$, in which the first of subsequent full *chip* detection occurs after 260 ns, which is close to the known propagation delay of the transmission line (page 78);

- Figure A.2: includes failed Ranging measurements caused by load end attenuation or TX output pin in low impedance mode, forcing logic level low after transmission (page 79);

- Figure A.3: demonstrates a loss of measurement precision with $K = 12$ or $K = 8$, contrary to what would be expected from using shorter pulse lengths; possible causes include pulse width variations within the *chip*, caused by different delays between the CTBV chains' elements (page 80);

- Figure A.4: shows Ranging measurements with Board 2, which has worse Ranging performance than Board 1 for the same value of K delay elements (page 81).

Waveforms of *Pulses* and *Chips*:

- Figure A.6: shows the corresponding waveforms of $K = 16$ and $K = 12$ *chips* from Board 1 (page 83);
- Figure A.7: visually demonstrates the wave reflection phenomenon in the transmission line that enables Ranging measurements after a *chip* is transmitted; also includes an estimation for the propagation delay of the transmission line, at around 258.6 ns (page 84);
- Figure A.8: documents a situation of load attenuation caused by a 50 Ω load at the end of the transmission line, which effectively prevents wave reflections and Ranging measurements; the figure also includes a waveform caused by a negative reflection coefficient effect when the TX module's output is not put into high-impedance mode after a *chip* is transmitted (page 85);
- Figure A.9: shows a lower amplitude *chip* of $K = 16$ from Board 2, the board which includes a Series Termination Scheme at the TX module's output pin and that performs worse in Ranging measurements (page 86).

Ranging with a Gold sequence of $N = 31$

Tests with a Gold sequence of length $N = 31$, *1110 0100 0101 0110 0100 1010 1101 101* resulted in lack of clear code detections by the correlators. It is believed that the additional logic element utilization for handling a larger code has degraded the FPGA performance due to increased Place & Routing complexity. Nevertheless, the time of *chip* at this code length is also closer to the delay of the transmission line from one end to the other. Consequently, wave superposition might also be a conditioning factor.

Figure A.10 shows a waveform of the *chips* transmitted by Board 1, Figure A.5 contains the corresponding Ranging measurement in Matlab.

5.2.2 Transceiver Configuration

In the Transceiver configuration all transmissions use a Kasami sequence and corresponding inverted sequence, of length $N = 15$. Both boards use the same sequence, for convenience, as only one board transmitted at a given time (see Appendix B) and solely for frame and bit error testing purposes.

Transmissions were attempted with several pulse lengths: K delay buffers with chains of $K = 16$, $K = 12$, $K = 8$ and $K = 4$ elements and expected delays of 8 ns, 6 ns, 4 ns and 2 ns. Chains of less than $K = 4$ elements did not yield usable results in Transceiver operation. Even though there were not enough correlators for detecting all bits in the whole 20 ns sampling period for

$K < 12$, the Board 2 continued to work regardless. Regarding *chip* detection, the best results in all boards were obtained when the detection threshold was set to up to 3 misaligned bits, allowing the payload to still be recovered.

All tests were applied initially in the two boards working standalone. Later, it was attempted to set each board as a transmitter and as a receiver. Due to Ranging operation compatibility, the TX and RX pins continued shorted together in both boards and the communication is half-duplex, although full-duplex can be achieved if configuring the hardware for Transceiver use alone. TX output pins were set to high-impedance when a board was working as a receiver.

In Transceiver configuration, the FPGAs process the correlator data and do the respective *chip* detections. In order to achieve viable Transceiver operation, all tests were set to allow 3 bit misalignments and still validate the received *chip*. A Matlab script (see Appendix C) was used to send UART data to the boards and wait for the reply, indicating a detection of a valid frame (8 *chips*) in the CTBV RX module. The script collected payload timeouts, payload errors and bit errors for 100 repetitions of all 256 possible 8 bit UART payload sequences, per number of K delay buffers. The total payload frames transmitted in each test was, therefore, 25.600 byte or 204.800 bit. Payload timeouts were the result of *chip* misdetection on the transceiver board, leading to a total frame loss. Actual *chip* error rate cannot be detected by Matlab as the system either transmits a complete frame or discards incomplete frames in order to avoid desynchronizing. When a frame timeout occurs, caused by *chip* not being detected, the receiver logic control module waits $W \times \tau_{chip}$; discarding the *chips* in that interval before attempting re-synchronization. This timeout interval should, ideally, be much inferior than the UART's baud period ($1/baudrate$). The boards were set for $W = 5$. Nevertheless, bit errors can still occur in events when no *chip* misdetection occurs but instead false *chip* detection: the number of bit errors presented in this section were calculated by adding the XOR result between the transmitted payload and the received payload on the Matlab script.

Payload Timeouts, Frame Errors and Bit Errors:

- Table A.1: Transceiver Configuration: Payload Timeouts (Lost Frames);
- Table A.2: Transceiver Configuration: Payload Errors (Bad Frames);
- Table A.3: Transceiver Configuration: Bit Errors (XOR sums) presents a detailed overview over the best pulse length and board for the Transceiver configuration on the available hardware.

Regarding the observed performances, Board 2 proved to be more fitted to transceiver operation than Board 1. Board 2 could generate and detect pulses up to 2 ns while still offering error-free communication. Although Board 2 performed better, the Matlab script is not fast enough to full stress the UART interface, allowing for intervals of several *milliseconds* between simultaneous transmissions. Still, the objective of this test was to determine the data integrity of the CTBV

Table 5.2: Board Performance in Transceiver Operation [1]

K buffers τ_{pulse}	4 2 ns	8 4 ns	12 6 ns	16 8 ns
Board 1	—	OK	OK	Poor
Board 2	OK	OK	OK	OK
B1-B2 [2]	—	Unusable	Very Poor	Poor
B1-B2 [3]	Unusable	Very Poor	OK	Unusable

(1) Qualitative classification based on payload timeouts (lost frames), see Table A.1.
(2) Board 1 transmitting and Board 2 receiving. TX module output in high-impedance on Board 2.
(3) Board 2 transmitting and Board 1 receiving. TX module output in high-impedance on Board 1.

transmissions rather than guaranteeing a given payload bitrate. See Table 5.2: Board Performance in Transceiver Operation, for a qualitative comparison between the performance of the two boards for different values of K delay elements. Figure A.11 shows waveforms sampled from Board 2 where the 8 *chips* of a complete payload frame are depicted, and Figure A.12 shows the shortest pulse width measured on the same board for $K = 4$.

5.3 Summary of System Specifications

A summary of expected and observed pulse widths is included in Table 5.3, listing expected pulse delays vs. number of K delay elements and an average pulse length based on the Kasami code used of $N = 15$. In spite of those values, for stable operation the actual reserved safe-guard intervals for *chip* transmission are longer than the true *chip* length. These reserved lengths are based on a $20ns$ counter based on the main FPGA clock.

Regarding the maximum bandwidth of the system, on the current hardware the shortest and still detectable pulse length is still over the minimum 2 ns limit imposed by the official 500 MHz minimum bandwidth definition for UWB signals (see Chapter 2, Section 2.1: UWB definition). However, as per the fractional bandwidth definition, the current hardware can serve as a pulse generator and detector block in conjunction with an up-converter block up to a maximum frequency allowed by the fractional bandwidth definition as described in Chapter 2, Section 2.2: Meeting UWB specifications. Nevertheless, the IR-UWB architecture for programmable logic proposed in this work can be used in more capable hardware and meet both UWB definitions.

Appendix A: Complement to Tests & Results' Chapter, includes additional measurements and raw data regarding Ranging and Transceiver performance for both boards.

Table 5.3: Number of K Delay Elements vs. Pulse Length

	τ_{pulse}			τ_{chip}	
Delay Elements	Expected [1]	Shortest [2]	Average [3]	Ranging [4]	Transceiver [5]
K=4	2 ns	2.7 ns	3.3 ns	—	100 ns
K=8	4 ns	3.9 ns	4.7 ns	80 ns	100 ns
K=12	6 ns	6.1 ns	6.0 ns	100 ns	120 ns
K=16	8 ns	7.3 ns	7.3 ns	140 ns	120 ns

(1) Considering a CTBV 100 % duty-cycle pulse. Average values expected from preliminary measurements of multiple chains of delay elements.
(2) Shortest τ_{pulse} width measured at 50 % amplitude level. Not full duty-cyle binary pulses, therefore extending the actual pulse repetition interval (τ_{chip}/N). Yet, as the absence of pulses (*zero* in an OOK CTBV signal) is of shorter length, for balanced codes the duration of *chips* remains close to what was expected ($\tau_{pulse} \times N$).
(3) Considering $N = 15$ and measured τ_{chips} of approximately 50 ns, 70 ns, 90 ns and 110 ns, respectively.
(4) Minimum reserved interval for reliable *chip* detection. Can be extended to improve detection performance by damping transient effects.
(5) Minimum reserved interval for reliable *chip* detection and frame synchronization.

5.3.1 Ranging configuration

The best Ranging measurements were obtained with Board 1, for $K = 16$ delay elements and with a $N = 15$ Kasami code. Regarding the observed performances, the closest propagation delay measured for the transmission line was 260 ns. At 7.3 ns average delay per pulse with $K = 16$, the measurement is within the half pulse length expected precision if compared with the 258.6 ns delay measured in Figure A.7.

It is, however, not as close to the 254 ns figure measured with the Rohde & Schwarz ZVL Network Analyzer. Besides possible asymmetric delays in the CTBV chains, causing *chip* misdetection or delayed sampling, other causes can possible be the use of an external coaxial cable of different material between the TX output pin and the transmission line, in which group delay could be affected. The possibility of a constant offset delay affecting the measurements could not be verified, as it was not possible to measure the propagation delay of cables of the same material and different lengths, however it is believed that constant internal offsets would not affect delta time measurements. Given that the system could not be calibrated against different references, it is still only about 6 ns off the most reliable line delay measurement, or about 1.42 m in distance equivalent for the calculated VF of 0.788.

Additionally, the lack of additional measurement precision can be explained by the fact that when the CTBV chains were projected, a full duty-cycle pulse was considered for estimating pulse propagation within the chain. Actual pulses have shorter duty-cycles and pulse duty-cycle might not be constant between different pulses of the same *chip*. Pulses might pass undetected in the chain if to short and desynchronized with sampling instants, even if the correlators are designed for an half pulse delay resolution as in the current architecture. There is however a limitation with

Table 5.4: Ranging: Specifications Overview [1]

	Expected			Measured		
Delay Elements	$K=8$	$K=12$	$K=16$	$K=8$	$K=12$	$K=16$
τ_{pulse}	4.0 ns	6.0 ns	8.0 ns	4.7 ns	6.0 ns	7.3 ns
τ_{chip}	60 ns	90 ns	120 ns	70 ns	90 ns	110 ns
Minimum Load Distance	14.2 m	21.3 m	28.3 m	16.5 m	21.3 m	26.0 m
Ranging Resolution [2]	0.47 m	0.71 m	0.94 m	—	—	—
Maximum Load Distance	605 m					
Maximum Ranging Interval	5.12 µs					

[1] Considering $N=15$ and the calculated Velocity Factor of 0.788. Equations from Chapter 3, page 37;
[2] Actual Ranging resolution could not be measured due to lack of identical cables of different lengths for comparison. Since pulse widths vary within the *chip*, the measurement resolution will be worse than half pulse width.

programmable logic: increasing the resolution of the RX CTBV chain might actually produce worse detection performance as more CTBV sampling taps cause more internal routing congestion and its associated propagation delays. The same routing congestion problem also affects longer TX CTBV chains, as observed for $N=31$, so a compromise should be reached for the hardware in use.

Regarding other Ranging specifications, Table 5.4: Ranging: Specifications Overview summarizes approximations of relevant Ranging characteristics in function of the number of K delay elements used.

5.3.2 Transceiver configuration

As previously stated, Board 2 offered best performance than Board 1 in the Transceiver configuration. Regarding the maximum achievable data-rates, although the shorter the pulses are, the higher is the bandwidth utilization, data-rates are still dependent on maximum pulse repetition times. These are obtained from τ_{chip}/N, and the time of *chip* must include a safe-guard interval for the Transceiver modules to work as intended. On top of that, a safe-guard interval between consecutive *chips* allows for the RX CTBV chain to more accurately detect incoming *chips* and its control logic to synchronize properly with the transmitter. These guard intervals reduce the maximum achievable data-rate but increase detection reliability, particularly important when thresholds are used for accepting partial *chip* detection.

See Table 5.5 for the maximum achievable data-rates with different values of K delay elements, and Figure A.11 for the corresponding waveform that illustrates the frame periods of Transceiver transmissions. Figure A.12 shows the shortest pulse measured from Board 2, in Transceiver configuration, for $K=4$.

Table 5.5: Transceiver: Payload Bitrate Overview [1]

	Expected				Measured			
Delay Elements	$K = 4$	$K = 8$	$K = 12$	$K = 16$	$K = 4$	$K = 8$	$K = 12$	$K = 16$
τ_{pulse}	2.0 ns	4.0 ns	6.0 ns	8.0 ns	3.3 ns	4.7 ns	6.0 ns	7.3 ns
τ_{chip}	30 ns	60 ns	90 ns	120 ns	50 ns	70 ns	90 ns	110 ns
Payload bitrate	33.3 Mbit/s	16.7 Mbit/s	11.1 Mbit/s	8.3 Mbit/s	6.2 Mbit/s		5.5 Mbit/s	

[1] Valid for $N = 15$. Expected values based on ideal scenarios without *chip* or payload guard intervals. Measured rates were not tested, but instead obtained based on frame delay measurements (see Figure A.11) and do not consider errors or payload misdetection.

5.4 Conclusions

This section provided not only a validation of the proposed IR-UWB architecture for programmable logic, with the proof-of-concept demonstrations, but also a set of performance specifications regarding the hardware implementation described in the previous chapter.

As for relevant conclusions gained from the current implementation, it was observed that for different numbers of K delay elements, the delays don't scale as linearly as expected. This is particularly true for lower numbers of delay elements as the design approaches the maximum operational limits of the internal logic or I/O ports. Propagation delay differences between logic elements also prevent increased Ranging resolution or increased payload bitrate, as a larger number of K elements has an averaging effect on propagation delay deltas between the chain's elements. There is, however, a compromise between design complexity and detection resolution, and each FPGA family should be experimented with a range of K values in an attempt to reach its operational sweet spot.

Regarding the performance metrics listed in this chapter, some limitations of the current hardware implementation could be perceived. In particular, the minimum pulse lengths that offered reliable operation were still longer than typical UWB implementations. Yet, the architecture can be scaled to more capable hardware regarding internal logic's propagation delays and I/O switching rates. Notwithstanding, the FPGA in the current hardware implementation can still immediately serve as a CTBV pulse generator and pulse detector block in a up-conversion IR-UWB scheme up to the limit of the UWB fractional bandwidth definition described in Chapter 2, Sections 2.1: UWB definition and 2.2: Meeting UWB specifications.

It was also concluded than different operational configurations, Ranging or Transceiver, require different impedance matching circuitry to work at full potential. In this implementation, Board 1 lacked a proper line impedance termination and performed better in the Ranging configuration. On the other hand, Board 2 performed better in the Transceiver configuration because it included a Series Termination scheme on the TX output pin.

The following Chapter, Final Remarks & Future Work, presents relevant conclusions over the work done and possible future works with the proposed IR-UWB architecture demonstrated in the aforementioned proof-of-concept Ranging and Transceiver applications.

Chapter 6

Final Remarks & Future Work

This dissertation work was motivated by the desire to bring UWB technology to a wider range of uses, by lowering its hardware implementation costs while still offering a useful solution for fast IR-UWB system deployment in programmable logic, in particularly FPGAs.

Since the liberation of the UWB spectrum and the authorization of UWB communications by regulatory agencies, this technology has been thoroughly researched in the academic world but only few and expensive commercial solutions are available up to this day, generally only accessible to military, government or academic institutions.

In order to further contribute to a wider utilization of UWB communications, and reaping the benefits of centimetre level ranging capability and higher communication data-rates, UWB implementations on programmable logic can help to expand the use of UWB radios to hardware already commonly available. This is relevant as it allows IR-UWB prototyping in various types of applications scenarios, otherwise not possible with less practical solutions. The use of FPGAs also makes the architecture more versatile and portable when a system upgrade is demanded.

One of the main challenges for accomplishing the proposed goals was to deal with uncontrollable delays within an FPGA's internal logic elements and ensuring that CTBV chains would offer more or less constant propagation delays for precise pulse generation and pulse detection. Ensuring this, in spite of having no control over the FPGA's Place & Routing process with the IDE used, actually allowed the development of a more universal approach that does not require manual floor-planning with the current hardware. By using common Verilog primitives, in ways not usually recommended by HDL development guidelines, it was actually possible to average out inherent combinational logic delays when no sequential logic and clock controlled buffers are used.

There are, however, associated limitations with the current hardware implementation. For instance, performance is still somewhat limited when compared with ASIC UWB solutions. Despite these limitations, the CTBV IR-UWB architecture inspired in Professor Tor Sverre Lande's works can demonstrate how even the less capable FPGAs can be used in various UWB use cases. The solution is also portable and can be further developed, integrated and deployed on more capable hardware to find its use in many applications of the modern day. It is a simple, low-cost and easily re-configurable IR-UWB baseband generator and detector architecture for a less complex next-generation IR-UWB system for the masses.

6.1 Summary of the Work Done

The main accomplishments and performed tasks for this dissertation work are briefly summarized in this section:

- Development an description of an adapted CTBV IR-UWB architecture suitable for being implemented in FPGAs and associated techniques for code generation and detection;
- Development of two IR-UWB applications for proof-of-concept scenarios: Ranging measurements and Transceiver operation, in order to demonstrate the validity of the proposed architecture;
- Implementation of the mentioned architecture in Verilog HDL and Matlab scripts for supporting the proof-of-concept demonstrations;
- Deployment of the proof-of-concept scenarios to an actual hardware implementation;
- Testing and performance characterization of the hardware implementation for the Ranging and Transceiver configurations under multiple conditions.

6.2 Future Work

UWB technology has an enormous potential number of use cases in the modern day, in applications that can benefit from high-resolution ranging and fast communications at high data-rates. Undetectable communication by common Narrowband receivers is another possibility for stealth or military agencies or even in privacy-sensitive communications, as shorter pulses spread the energy over a very large bandwidth.

Besides the Ranging and Transceiver applications demonstrated in this work, which can be used for transmission line fault mitigation and for digital communication purposes, there are many other possibilities. For instance, the deployment of automotive radar and collision avoidance systems can be based on UWB, leading to an ever more safe driving. Others include low-power sensor networks for the IoT and industrial applications that require precise distance measurements.

Additional applications and future works include:

- A true random number generators based on actual hardware, by letting the input pin of the RX CTBV chain floating close to is threshold potential,VCC_{IO}, and continuously sampling the chain for random binary sequences of a desired length;
- The inclusion of fast communication interfaces, such as USB soft-cores, to explore the possibility of faster data-rates between the system and the external control system;

- The utilization of video or audio signals as Transceiver payloads to evaluate the viability of using the system for meeting tight timing requirements;
- The deployment of the architecture to a more capable hardware and reaching *sub-nanosecond* measurement resolution and faster data-rates;
- A dissertation work focused on continuing the work done and porting the current hardware implementation to the wireless domain, enabling radar measurements and IR-UWB wireless communications.

Appendix A

Complement to Tests & Results' Chapter

This Appendix Chapter includes Ranging measurements with Matlab, Transceiver performance data regarding payload integrity and relevant waveform captures to demonstrate generated pulses in different scenarios, for the Ranging and Transceiver configurations. It supports the information contained in Chapter 5: Tests & Results, whenever the content could not be included for being to extensive to fit in the chapter.

Two identical boards, yet with different output driver's impedance (see Figure 4.4), were used in these configurations:

Board 1: TX output driver configured for maximum drive strength (24 mA).

Board 2: TX output driver configured for matching the impedance of the transmission line (75 Ω). Includes an on-chip series terminated resistor and an external resistor.

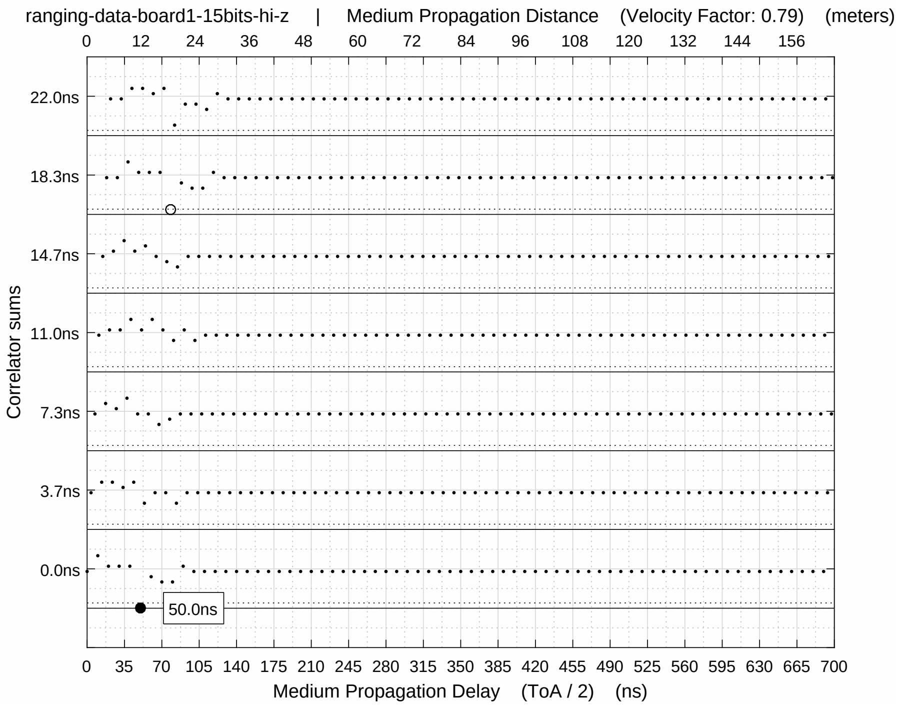
A.1 Ranging Proof-of-Concept

A.1.1 Ranging Measurements with Matlab

A Matlab script was created to send a Ranging measurement request to the boards and to collect the received data, later displaying it on the screen and keeping a file copy of the data for later analysis (see Appendix C: Source-Code). The measurements are plotted in a graph that is adjusted for average pulse delay according to the number of delay elements *K* in use, as listed in 5.3.

A.1.2 Waveforms of Ranging Measurements

This section includes screen captures of waveforms generated by both boards. The signals were sampled at the load end of the transmission line with an HP 16500A Logic Analysis System Mainframe populated with a 1 GSa/s DSO module. The probes used were UNI-T's UT-P05 model, a set of passive 200 MHz probes that were set to 10× attenuation mode.



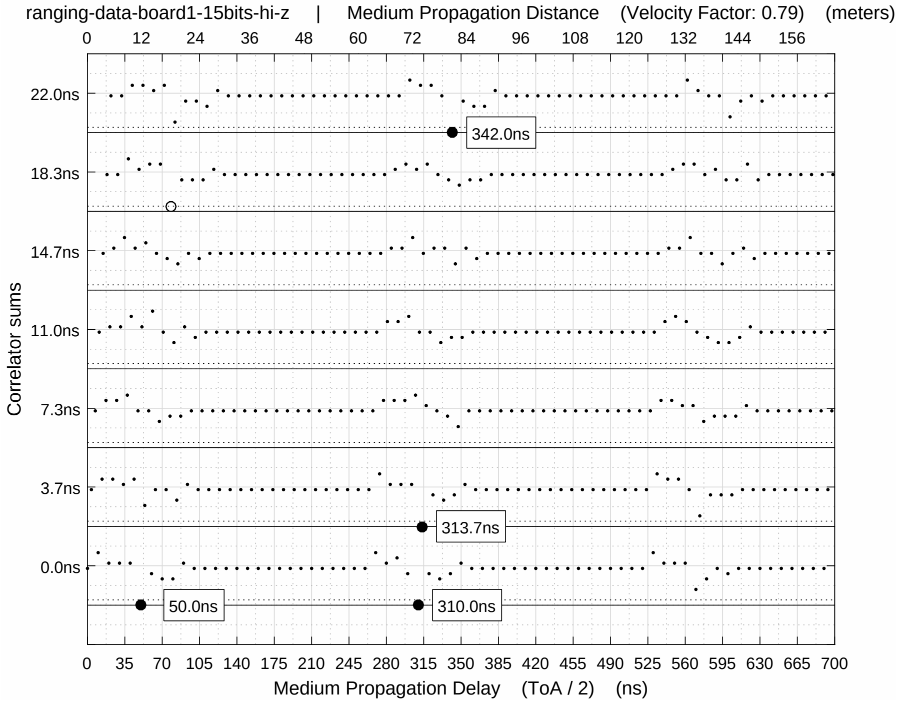
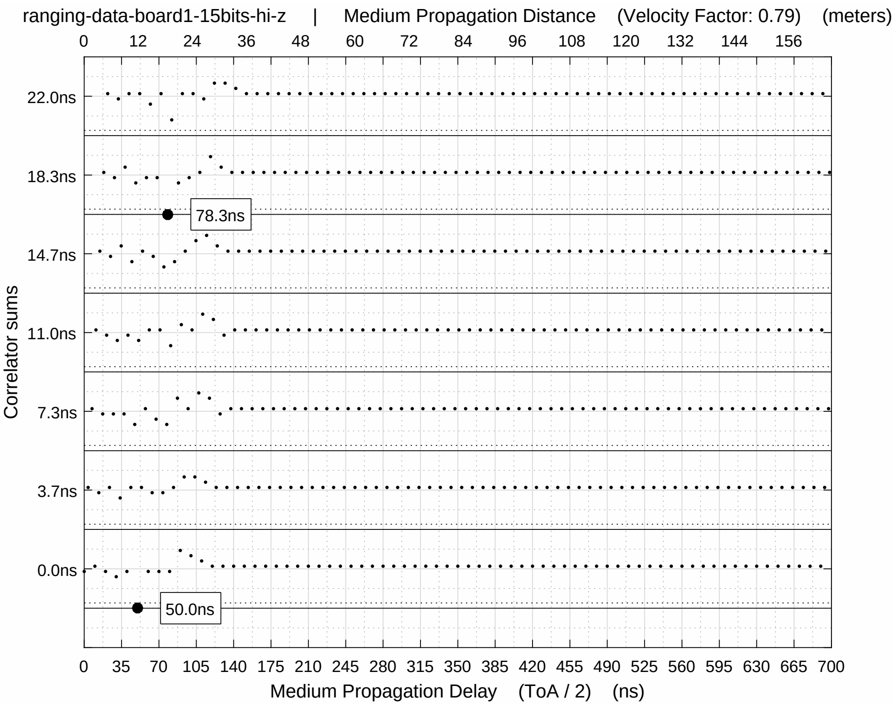


Figure A.1: Ranging measurements: Board 1, plot of correlator data in Matlab for $K = 16$, $N = 15$, TX output in *Hi-Z* after transmission.

- **On top**: the measurement was taken before attaching the transmission line to the TX module's output pin. Points marked with labels represent a sample in which all the outputs of the correlator's *XORs* are equal to *zero*, indicating a valid *chip* detection. The first oscillation sets the time reference for measurements: it is the result of the TX and RX pins being shorted and is demonstrated by the detection at 50 ns.
- **On bottom:** measurement was taken after connecting the transmission line to the TX output. The secondary detections derive from the first load reflection after 260 ns, based on the delta time between 50 ns and 310 ns. The first reflection is followed by subsequent reflections with the same period, as the signal is also reflected back from the TX and RX end of the line because both are set to high-impedance mode. The data also suggests a cable length of approximately 61.4 m (see Equation 3.13).



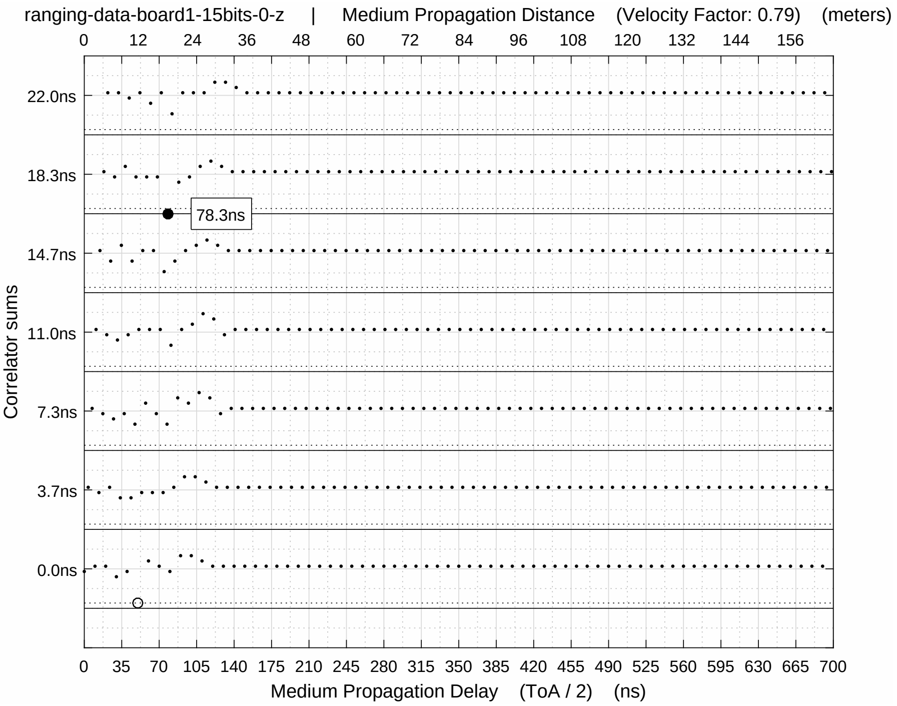
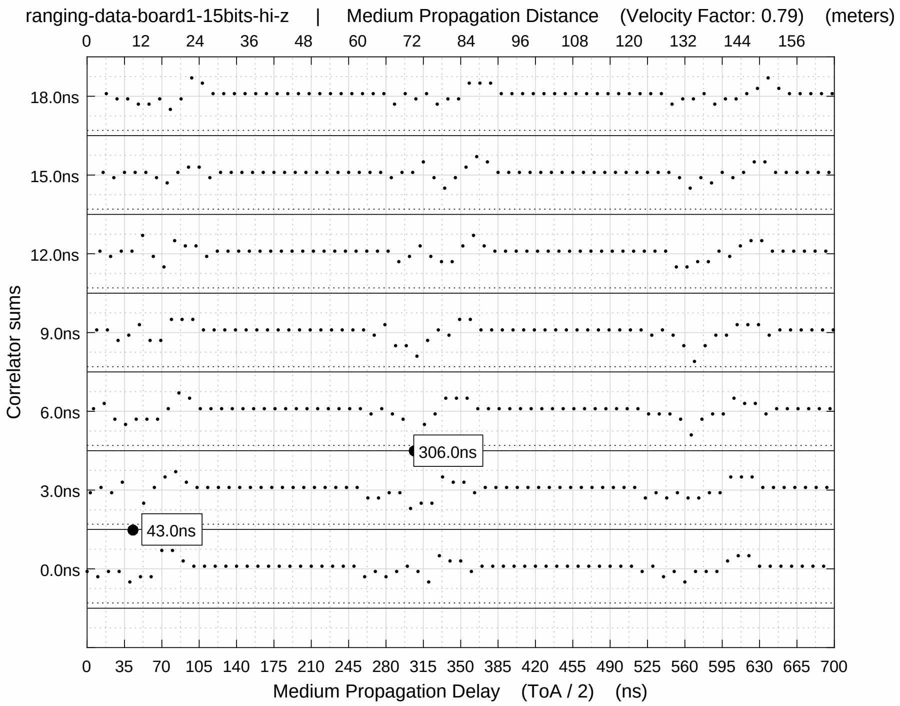


Figure A.2: Ranging measurements: Board 1, plot of correlator data in Matlab for $K = 16, N = 15$, TX output in *Hi-Z* (top) and *0-Z* (bottom) after transmission.

- **On top:** measurement taken after connecting the transmission line to the TX module output pin and inserting a Midwest Microwave Model 290 $-10\,\text{dB}$ $50\,\Omega$ load. No discernible reflections take place, as the load absorbs most of the energy in the line (see Figure A.8 top).
- **On bottom:** the result is similar to what is shown in the top plot. No load is connected but the TX module's output pin is set to logic level *low* after transmission, and not high-impedance, forcing the logic level on the RX input in spite of the reflected signal. This latter scenario also causes a situation of short-circuit (*0-Z*) on the TX and RX end of the line, with a corresponding negative reflection coefficient as seen in Figure A.8 (bottom).



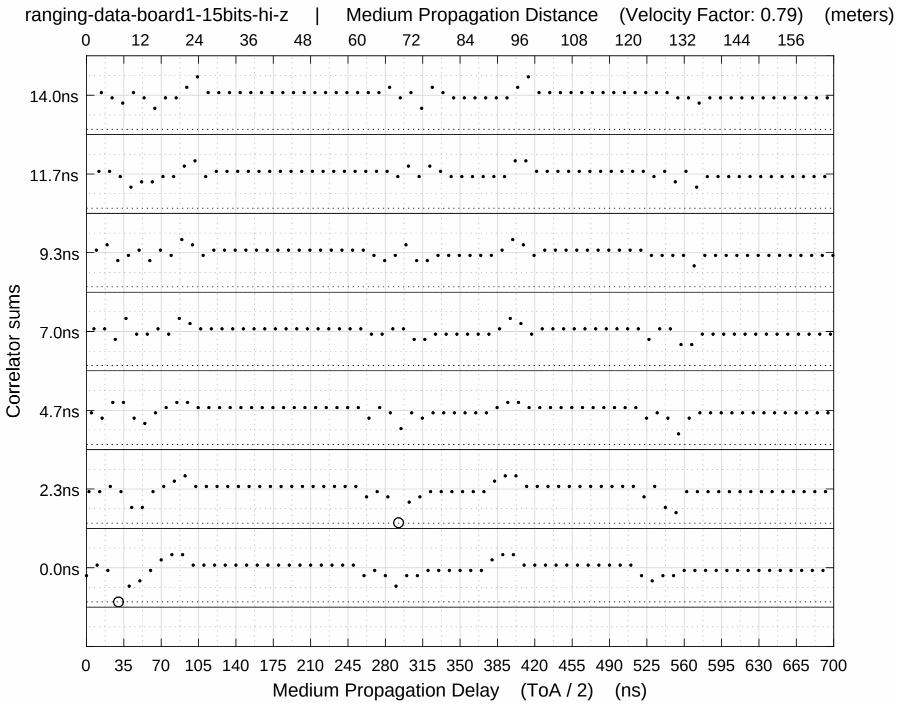
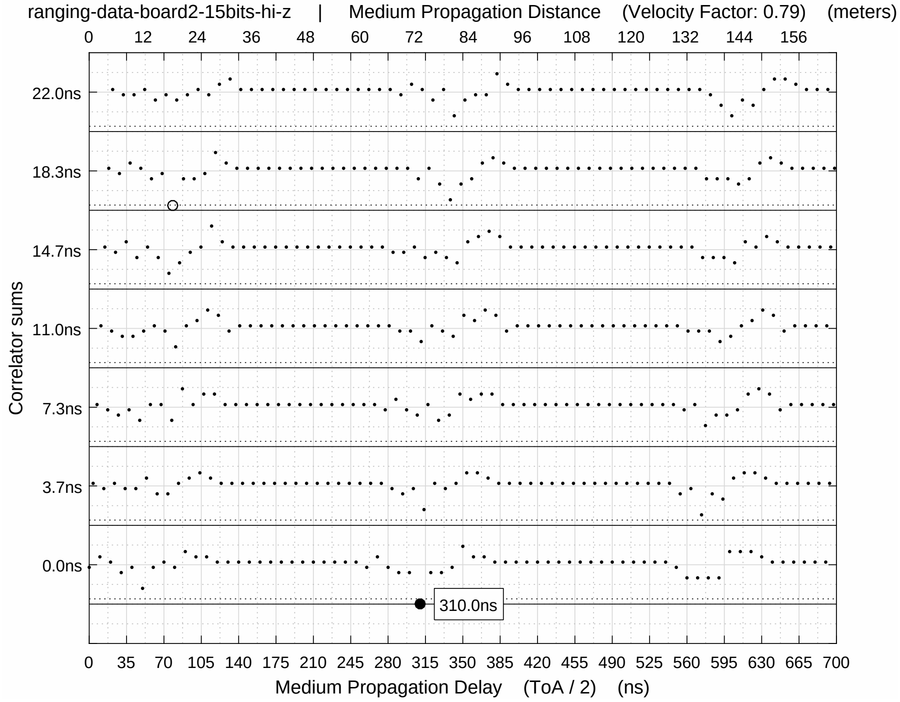


Figure A.3: Ranging measurements: Board 1, plot of correlator data in Matlab for $K = [12,\ 8]$, $N = 15$, TX output in *Hi-Z* after transmission.
• **On top**: for $K = 12$, pulses are shorter and the receiver is more sensitive to variations between different pulse lengths generated on the same *chip*. As a result, few *chips* are detected and some measurement accuracy is lost, contrary to what should happen in with a shorter pulse length. Since the number of delay elements was reduced, the averaging effect on possible pulse width variations is diminished.
• **On bottom**: for $K = 8$, full *chip* detection does not occur and the measurement almost looses significance. As in the example above, the first detected partial *chip* appear earlier, indicating shorter pulses in spite of all examples using the same reserve time for *chip* transmission before the output is switched to high-impedance. At this pulse length, the 7 built-in correlators are also not enough for covering the full Ranging interval between samples.



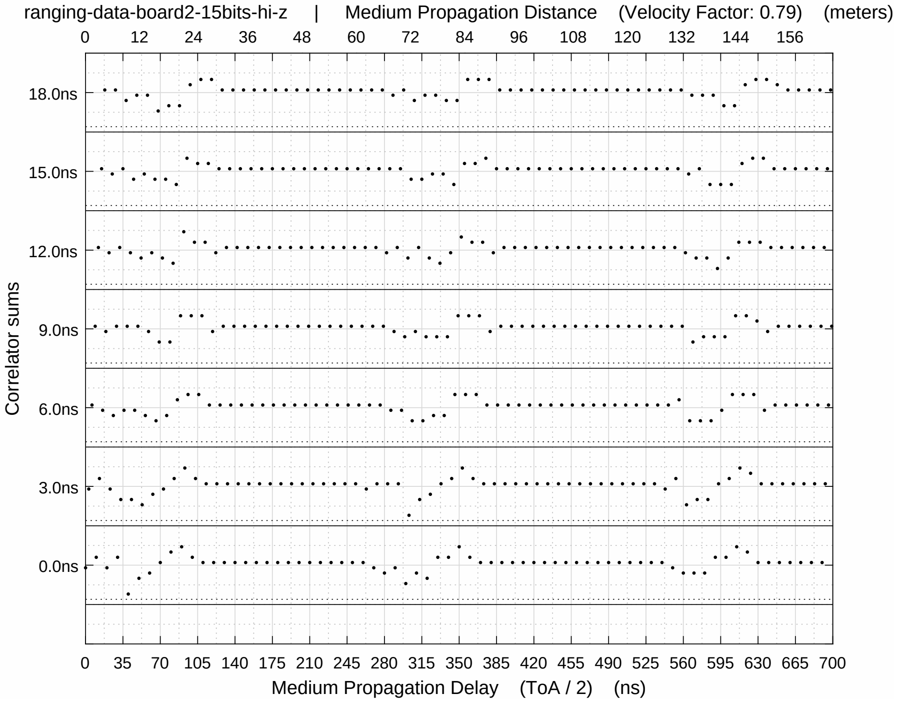


Figure A.4: Ranging measurements: Board 2, plot of correlator data in Matlab for $K = [16,\ 12]$, $N = 15$, TX output in *Hi-Z* after transmission.

- **On top**: for $K = 16$, Board 2 shows worse Ranging performance than Board 1.
- **On bottom**: for $K = 12$, Board 2 can no longer detect *chips*. Board 2 appears to be less sensitive for Ranging measurements, particularly for shorter pulses, possible due to the fact that its Series Termination scheme is attenuating the generated and reflected signals as illustrated by Figure A.9.

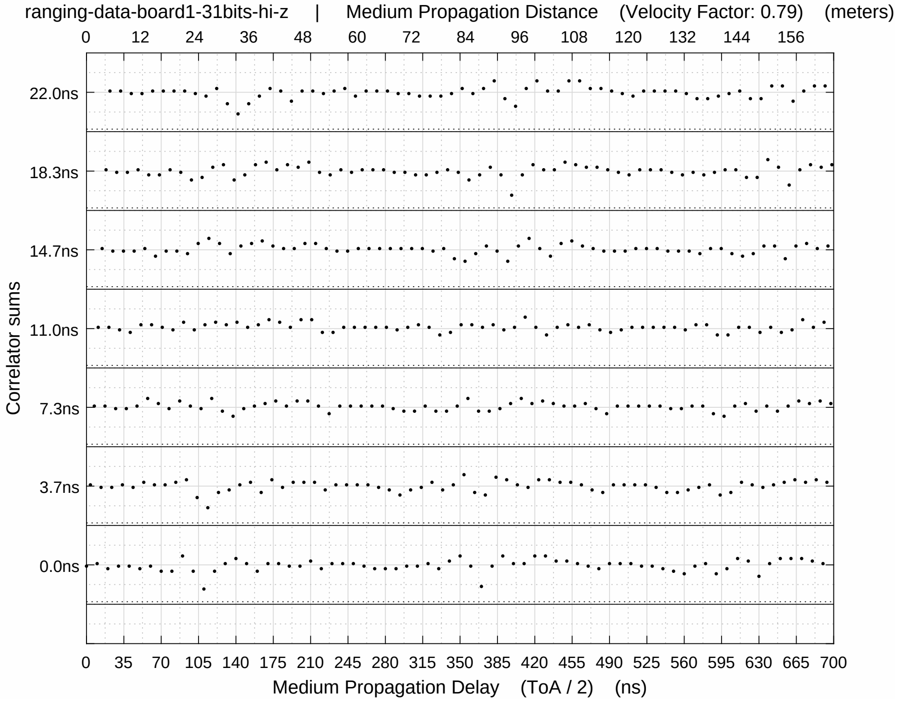
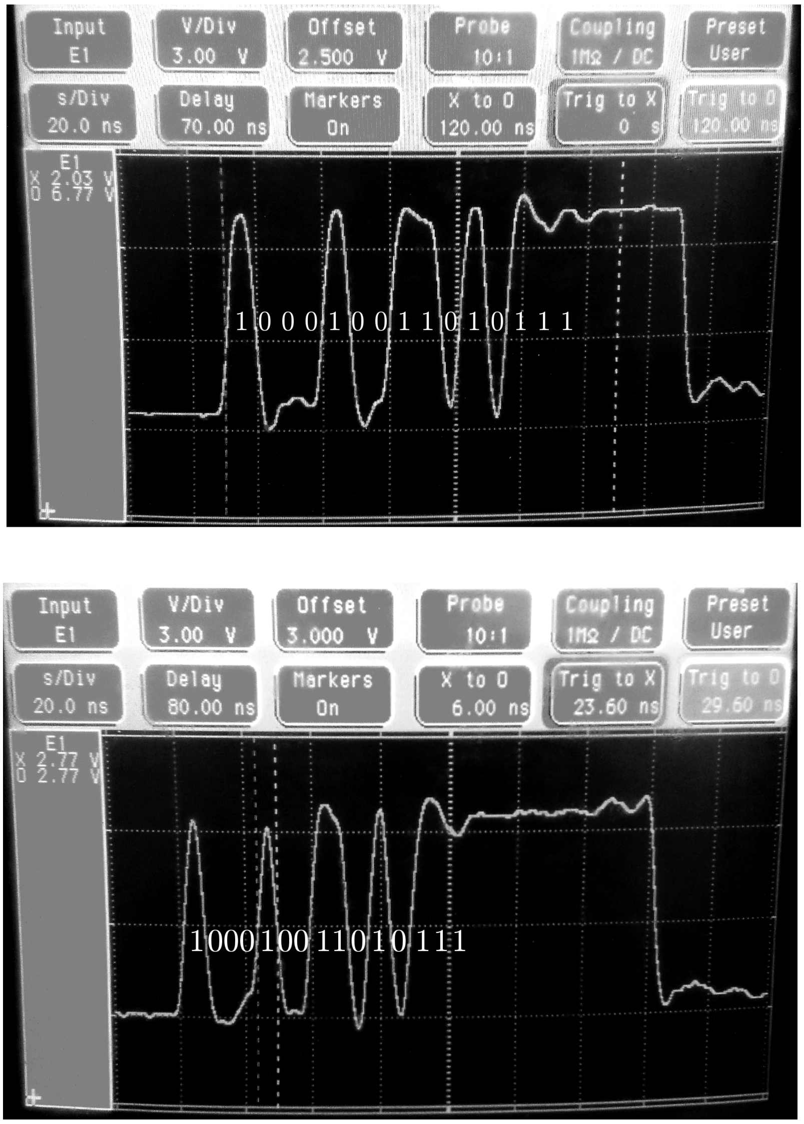


Figure A.5: Ranging measurements: Board 1, plot of correlator data in Matlab for $K = 16$, $N = 31$, TX output in *Hi-Z* after transmission. This Ranging measurement with an $N = 31$ Gold sequence did not produce valid detections, contrary to previous results with $N = 15$ Kasami sequences. Possible reasons for measurement failure include additional logic element utilization, causing lack of performance, or wave superposition in the medium as for this *chip* length the line approaches the limit defined in Equation 3.14. However, using a smaller number of K elements did not improve the measurements.



Input E1 | V/Div 3.00 V | Offset 2.500 V | Probe 10:1 | Coupling 1MΩ / DC | Preset User

s/Div 20.0 ns | Delay 70.00 ns | Markers On | X to O 120.00 ns | Trig to X 0 s | Trig to O 120.00 ns

E1 X 2.03 V O 6.77 V

Input E1 | V/Div 3.00 V | Offset 3.000 V | Probe 10:1 | Coupling 1MΩ / DC | Preset User

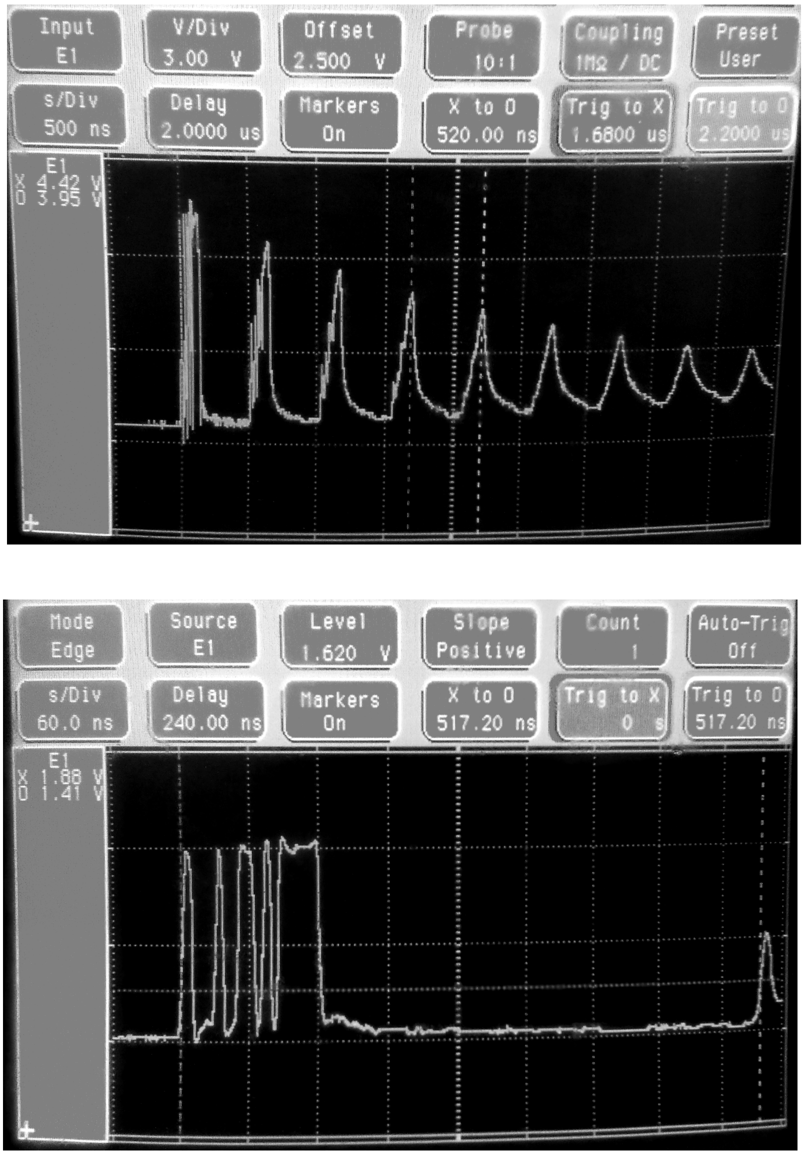
s/Div 20.0 ns | Delay 80.00 ns | Markers On | X to O 6.00 ns | Trig to X 23.60 ns | Trig to O 29.60 ns

E1 X 2.77 V O 2.77 V

Figure A.6: Ranging measurements: Board 1, waveforms of an $N = 15$ *chip*, made of pulses under 8 ns in length on top and about 6 ns on bottom, with the corresponding Kasami sequence overlaid on the figure. The reserved *chip* length on top is 120 ns plus an additional guard interval to prevent interference on the last elements of the sequence when the TX output state switches to the high-impedance mode.

• **On top**: for $K = 16$, the observed pulse length is just under 8 ns, as the 15 bit code is transmitted before the 120 ns mark.

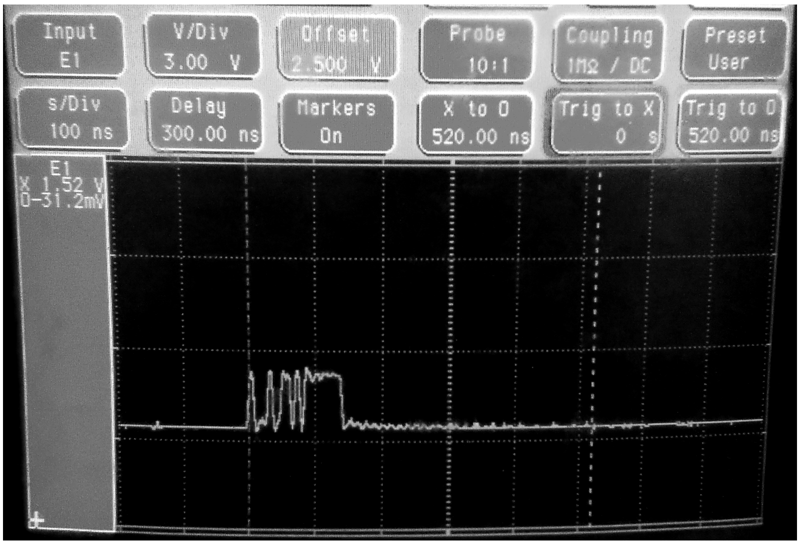
• **On bottom**: for $K = 12$, when given the same interval for transmission, the *chip* appears to be completed near the 90 ns mark.



Input E1
V/Div 3.00 V
Offset 2.500 V
Probe 10:1
Coupling 1MΩ / DC
Preset User
s/Div 500 ns
Delay 2.0000 us
Markers On
X to O 520.00 ns
Trig to X 1.6800 us
Trig to O 2.2000 us
E1
X 4.42 V
O 3.95 V

Mode Edge
Source E1
Level 1.620 V
Slope Positive
Count 1
Auto-Trig Off
s/Div 60.0 ns
Delay 240.00 ns
Markers On
X to O 517.20 ns
Trig to X 0 s
Trig to O 517.20 ns
E1
X 1.88 V
O 1.41 V

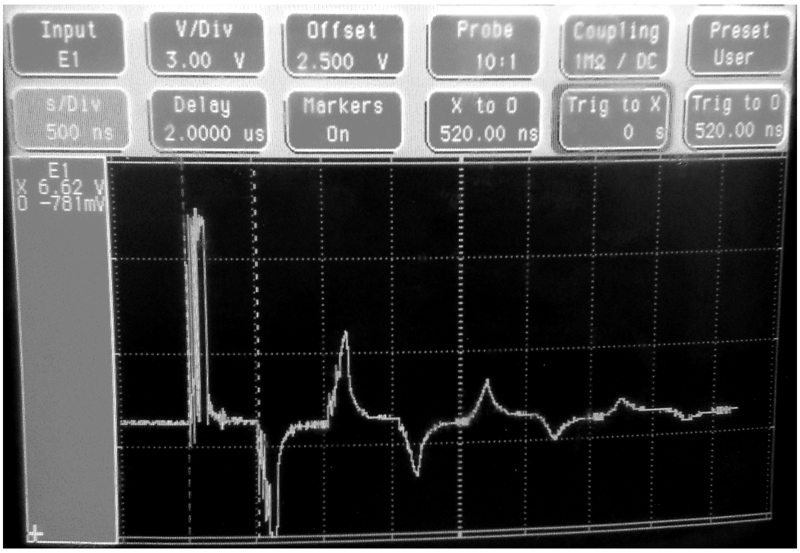
Figure A.7: Ranging measurements: Board 1, wave reflections of an $N = 15$ *chip*, made of pulses under 8 ns in length. This measurement, taken at the load's end of the transmission line, includes not only the transmitted *chip* but also the multiple reflections between both ends of the transmission line. The reflections' amplitudes are always positive, indicating a positive reflection coefficient at both sides (both line ends in *Hi-Z*), and their approximate periods, of 258.6 ns × 2, are close to the delay value estimated with Board 1 in Figure A.1 (bottom).



Input E1 | V/Div 3.00 V | Offset 2.500 V | Probe 10:1 | Coupling 1MΩ / DC | Preset User

s/Div 100 ns | Delay 300.00 ns | Markers On | X to O 520.00 ns | Trig to X 0 s | Trig to O 520.00 ns

E1
X 1.52 V
O-31.2mV



Input E1 | V/Div 3.00 V | Offset 2.500 V | Probe 10:1 | Coupling 1MΩ / DC | Preset User

s/Div 500 ns | Delay 2.0000 us | Markers On | X to O 520.00 ns | Trig to X 0 s | Trig to O 520.00 ns

E1
X 6.62 V
O -781mV

Figure A.8: Ranging measurements: Board 1, attenuation and inverted wave reflections of an $N = 15$ *chip*, made of pulses under 8 ns in length.

- **On top**: the waveform is the result of load attenuation caused by a Midwest Microwave Model 290 -10 dB 50 Ω load (see Figure A.2, top). Also noticeable is the lack of discernible reflections.
- **On bottom**: inverted wave reflections are originated by a condition of short-circuit (*0-Z*) at the TX and RX end of the transmission line, mentioned in Figure A.2, bottom). No load is connected at the load end, as in the waveform above, but the TX module's output pin is set to logic level *low* after transmission instead of high-impedance. This forces the logic level on the RX input in spite of the reflected signal and originates the inverted wave reflection caused by a negative reflection coefficient effect.

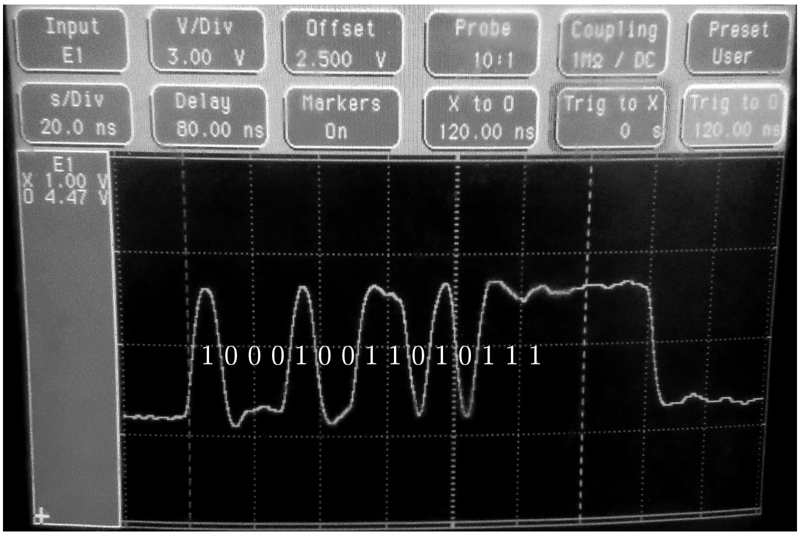


Figure A.9: Ranging measurements: Board 2, plot of a *chip* of $N = 15$, made of pulses under 8 ns in length, and the corresponding Kasami sequence overlaid on the figure. The board used did include a series termination resistor to match the impedance of the transmission line. Compared to Figure A.6, the signal's amplitude is reduced but otherwise the waveforms are similar.

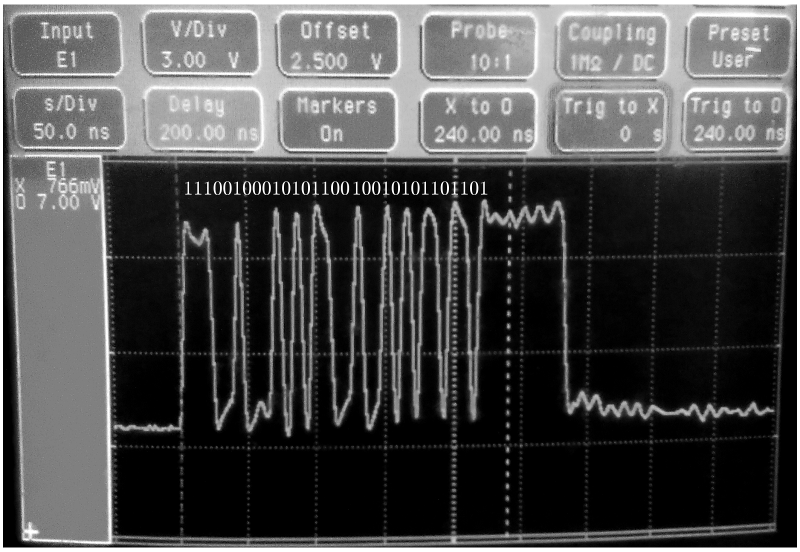


Figure A.10: Ranging measurements: Board 1, waveform of an $N = 31$ *chip*, made of pulses under 8 ns in length, and the corresponding Gold sequence overlaid on the figure. The reserved *chip* length is 240 ns plus an additional guard interval to prevent interfering with the last elements of the sequence when the TX output state switches to the high-impedance mode. The observed pulse length is just under 8 ns, as the 15 bit code is transmitted before the 240 ns mark.

A.2 Transceiver Proof-of-Concept

A.2.1 Transceiver: Data Integrity vs. Hardware Configuration

In this section, the following tables present a detailed overview over the best pulse length and best board for the Transceiver configuration on the current hardware:

- **Table A.1**: Transceiver Configuration: Payload Timeouts (Lost Frames) (page 87);
- **Table A.2**: Transceiver Configuration: Payload Errors (Bad Frames) (page 88);
- **Table A.3**: Transceiver Configuration: Bit Errors (XOR sums) (page 88).

A.2.2 Waveforms of Payload Transmissions

This section includes screen captures of waveforms generated by Board 2, which holds the hardware configuration that yielded the best performance in Transceiver operations. The signals were sampled at the board's TX output pin with an HP 16500A Logic Analysis System Mainframe populated with a 1 GSa/s DSO module. The probes used were UNI-T's UT-P05 model, a set of passive 200 MHz probes that were set to 10× attenuation mode.

Table A.1: Transceiver Configuration: Payload Timeouts (Lost Frames) [1]

	Payload Timeouts (Lost Frames)			
Delay Elements	$K = 4$	$K = 8$	$K = 12$	$K = 16$
τ_{pulse}	2 ns	4 ns	6 ns	8 ns
Board 1	—	37 (0.1%)	18 (0.1%)	5035 (19.7%)
Board 2	0	0	0	15 (0.1%)
B1-B2 [2]	—	10729 (41.9%)	3899 (15.2%)	4995 (19.5%)
B2-B1 [3]	22984 (89.8%)	8547 (33.4%)	0	16287 (63.6%)

[1] Valid for $N = 15$. Expected τ_{pulse} values, actual pulse lengths can differ.
[2] Board 1: TX, Board 2: RX with TX disabled (*Hi-Z*).
[3] Board 2: TX, Board 1: RX with TX disabled (*Hi-Z*).

Table A.2: Transceiver Configuration: Payload Errors (Bad Frames) [1]

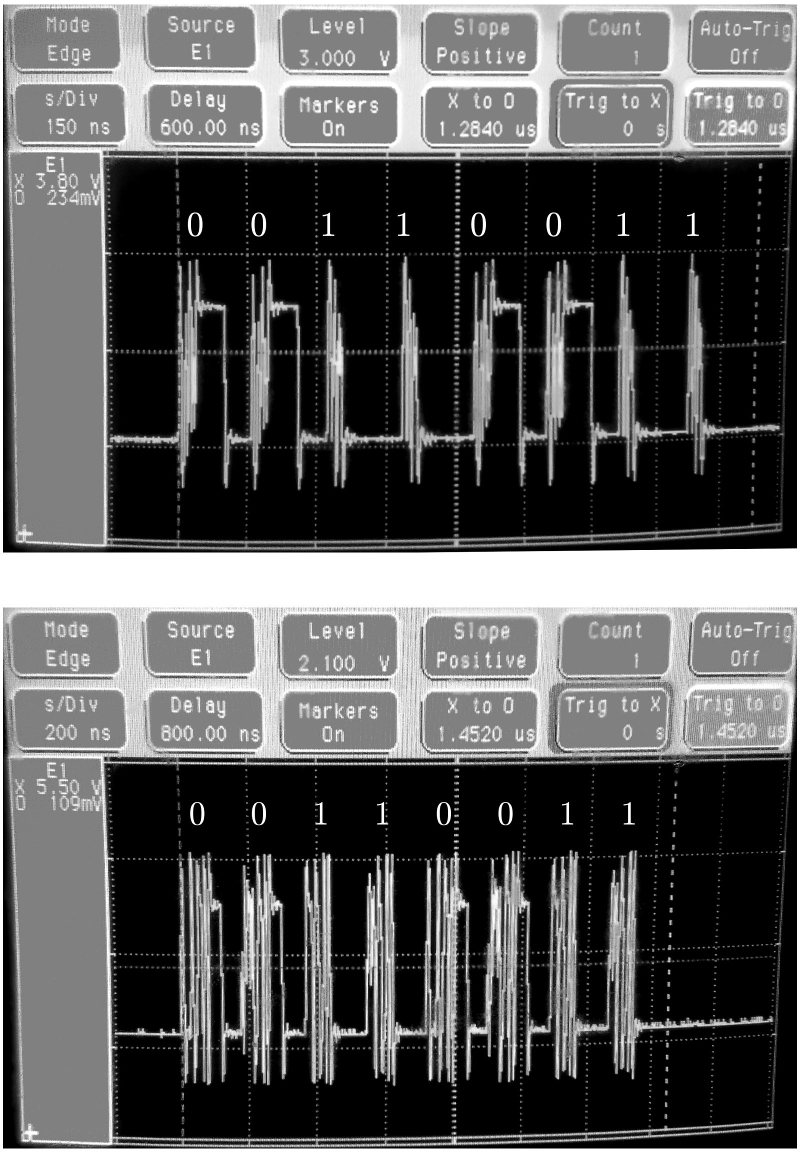
	Payload Errors (Bad Frames)			
Delay Elements	$K = 4$	$K = 8$	$K = 12$	$K = 16$
τ_{pulse}	2 ns	4 ns	6 ns	8 ns
Board 1	–	0	1 (0.0%)	6672 (26.1%)
Board 2	0	0	0	0
B1-B2 [2]	—	4907 (19.2%)	17522 (68.4%)	11661 (45.6%)
B2-B1 [3]	501 (2%)	3529 (13.8%)	0	2115 (8.3%)

(1) Valid for $N = 15$. Expected τ_{pulse} values, actual pulse lengths can differ.
(2) Board 1: TX, Board 2: RX with TX disabled (*Hi-Z*).
(3) Board 2: TX, Board 1: RX with TX disabled (*Hi-Z*).

Table A.3: Transceiver Configuration: Bit Errors (*XOR* sums) [1]

	Bit Errors (*XOR* sums)			
Delay Elements	$K = 4$	$K = 8$	$K = 12$	$K = 16$
τ_{pulse}	2 ns	4 ns	6 ns	8 ns
Board 1	—	0	4 (0.0%)	13288 (6.5%)
Board 2	0	0	0	0
B1-B2 [2]	—	10773 (5.3%)	43423 (21.2%)	33580 (16.5%)
B2-B1 [3]	1175 (0.6%)	6958 (3.4%)	0	4902 (2.4%)

(1) Valid for $N = 15$. Expected τ_{pulse} values, actual pulse lengths can differ.
(2) Board 1: TX, Board 2: RX with TX disabled (*Hi-Z*).
(3) Board 2: TX, Board 1: RX with TX disabled (*Hi-Z*).



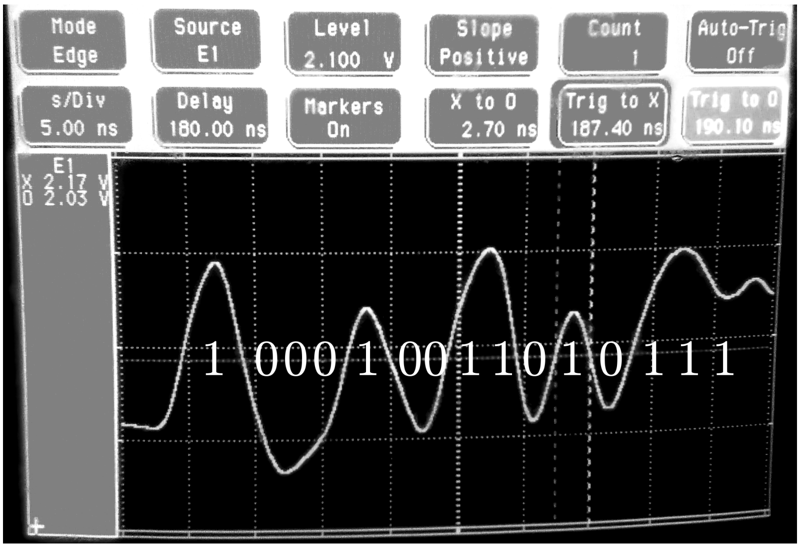
Mode Edge
Source E1
Level 3.000 V
Slope Positive
Count 1
Auto-Trig Off
s/Div 150 ns
Delay 600.00 ns
Markers On
X to O 1.2840 us
Trig to X 0 s
Trig to O 1.2840 us
E1 X 3.80 V O 234mV

Mode Edge
Source E1
Level 2.100 V
Slope Positive
Count 1
Auto-Trig Off
s/Div 200 ns
Delay 800.00 ns
Markers On
X to O 1.4520 us
Trig to X 0 s
Trig to O 1.4520 us
E1 X 5.50 V O 109mV

Figure A.11: Transceiver measurements: Board 2, waveforms of 8 *chips* of $N = 15$, made of pulses under 8 ns in length, and the corresponding payload sequence (LSB first: 11001100) overlaid on the figure.

- **On top**: for $K = 4$ and $K = 8$, the maximum payload data-rate is about (6.2 Mbit/s).
- **On bottom**: for $K = 12$ or $K = 16$, the limit is set to (5.5 Mbit/s) for reliable operation.

These rates are best-case scenarios given the current guard intervals between *chips*, excluding frame timeout situations and de-synchronization caused by undetected *chips*.



Mode Edge

Source E1

Level 2.100 V

Slope Positive

Count 1

Auto-Trig Off

s/Div 5.00 ns

Delay 180.00 ns

Markers On

X to 0 2.70 ns

Trig to X 187.40 ns

Trig to 0 190.10 ns

E1

X 2.17 V

0 2.03 V

Figure A.12: Transceiver measurements: Board 2, waveform of the shortest pulse ($N = 15$) with $K = 4$ delay elements per CTBV bit, measured at approximately 50 % potential level regarding the maximum observed peak. In this setting, Board 2 can still work without errors if thresholds are set to validate *chips* with up to 3 misaligned bits. Although not useful for Ranging measurements, payload frames can still be sent and received at this level of tolerance with good reliability in noise-free scenarios.

Appendix B

Gold & Kasami codes

B.1 Set of Gold codes

A set of Gold codes for the Ranging measurements, of $N = 31$, was generated with a modified Matlab script from [55] (see Appendix C). The code used was: *31'b1110 0100 0101 0110 0100 1010 1101 101*, and its corresponding autocorrelation plot is available in Chapter 2: Theory & Literature Review, in Figure 2.3 (Gold code A, page 15). The code generator polynomial for this Gold code is: *31'b0010 1100 1111 1010 1101 1111 0110 111*.

B.2 Set of Kasami codes

A small set of $N = 15$ Kasami codes was obtained from [56]. The codes in the set are:

- *15'b1000 1001 1010 111*: code used in the Ranging and Transceiver configurations
- *15'b0011 1111 0111 010*
- *15'b0101 0010 1100 001*
- *15'b1110 0100 0001 100*

The corresponding code generator polynomial for the used code is: *15'b1001 1010 1111 001*. The inverted code also used in the Transceiver configuration is: *15'b0111 0110 0101 000*, with the corresponding code generator polynomial: *15'b1001 1010 1111 000*. The cross-correlations and autocorrelations of the Kasami codes used are shown on Figure B.1.

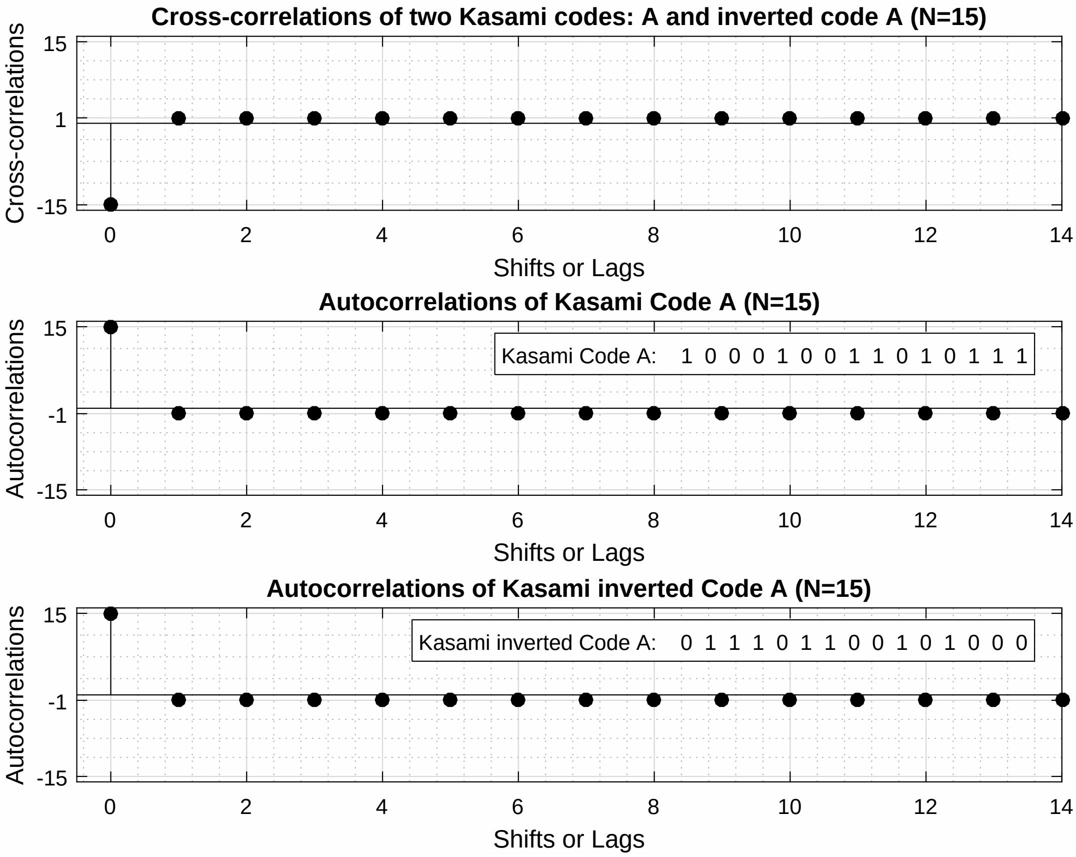


Figure B.1: Plots of cross-correlations and autocorrelations of Kasami codes with period $N = 15$.

Appendix C

Source-Code

C.1 Matlab scripts

The source-code of the Matlab scripts developed for the proof-of-concept demonstrations of Ranging and Transceiver applications are available on the project's website, at `http://www.uwb-radio.com`, for reference.

C.2 Verilog HDL Source-Code

This section includes the most relevant modules of the IR-UWB CTBV architecture. The complete source-code is available on the project website, `http://www.uwb-radio.com`, for reference.

C.2.1 TX module

Listing C.1: Verilog code of the TX module

```
// ### CTBV TX module ###
module tx #(
        parameter code_length = 15, //default code length
        parameter chain_length = 16 //default number of K delay elements
)(
    poly,
    codeword,
    tx_out,
    tx_signal
);

// ### Inputs - Wires - Outputs - Local Params ###
input [code_length-1:0] poly; //generator polynomial
input [code_length-1:0] codeword; //codeword to be generated
input tx_signal;

wire [code_length-1:0] poly;
wire [code_length-1:0] codeword;
wire initial_value = 1'b0;
wire xor_in [code_length-1:0];
wire delay_chain_in [code_length:0];
wire delay_chain_out [code_length:0];

output tx_out;

// ### Delay Chains ###
generate
  genvar i;
  for (i=0; i<code_length+1; i=i+1) begin : delay_chain_n
    delay_chain_tx #(
            .chain_length(chain_length)
       ) i_delay_chain(
         .in(delay_chain_in[i]),
         .out(delay_chain_out[i]),
         .mid_tap()
      )/* synthesis keep = 1 */;
  end
endgenerate

// ### XORs ###
assign xor_in[0] = (tx_signal) ? poly[0] : codeword[code_length-1];

generate
  for (i=1; i<code_length; i=i+1) begin : xor_assigns
    assign xor_in[i] = (tx_signal) ? poly[i] : initial_value;
  end
endgenerate

generate
  for (i=0; i<code_length; i=i+1) begin : xor_gates
    xor i_xor(delay_chain_in[i+1], delay_chain_out[i], xor_in[i]);
  end
endgenerate

// ### Inputs -- Outputs ###
assign delay_chain_in[0] = initial_value;
assign tx_out = delay_chain_out[code_length];

endmodule
```

C.2.2 RX module

Listing C.2: Verilog code of the RX module

```
// ### CTBV RX module ###
module rx #(
        parameter code_length = 15, //default code length
        parameter chain_length = 16, //default number of K delay elements
        parameter n_correlators = 7, //number of correlators used
        parameter half_correlators = 4,
        parameter rx_corr_out_length = code_length //defines the width of RX FIFO's words
)(
    codeword_0,
    codeword_1,
    rx_in,
    rx_out,
    rx_ok_0,
    rx_ok_1,
    rx_xor_out
);

// ### Inputs - Wires - Outputs - Local Params ###
input [code_length -1:0] codeword_0; //expected code for payload 0
input [code_length -1:0] codeword_1; //expected code for payload 1
input rx_in;

wire [code_length -1:0] codeword_0;
wire [code_length -1:0] codeword_1;
wire rx_in;
wire rx_out;
wire rx_ok_0;
wire rx_ok_1;
wire [rx_corr_out_length -1:0] rx_corr_in [n_correlators -1:0];
wire [rx_corr_out_length-1+n_correlators:0] delay_chain_in;
wire [rx_corr_out_length-1+n_correlators:0] delay_chain_out;
wire [rx_corr_out_length-1+n_correlators:0] delay_chain_mid_tap;

output rx_out;
output rx_ok_0;
output rx_ok_1;
output [code_length -1:0] rx_xor_out [n_correlators -1:0];

localparam n_xors = rx_corr_out_length;

//### Delay chains of K delays elements###
generate
  genvar i;
  for (i=0; i<rx_corr_out_length+n_correlators; i=i+1) begin : delay_chain_n
    delay_chain #(
            .chain_length(chain_length)
       ) i_delay_chain(
             .in(delay_chain_in[i]),
             .mid_tap(delay_chain_mid_tap[i]),
             .out(delay_chain_out[i])
      );
  end
endgenerate

assign delay_chain_in[0] = rx_in;
generate
  for (i=0; i<rx_corr_out_length+n_correlators-1; i=i+1) begin : delay_chain_wires
    assign delay_chain_in[i+1] = delay_chain_out[i];
  end
endgenerate

// ### Correlators' CTBV inputs ###
generate
genvar k;
for (k=0; k<half_correlators;k=k+1) begin : k_chain
  for (i=0; i<n_xors; i=i+1) begin : k_i_rx_xors
            assign rx_corr_in[2*k][i] = delay_chain_out[i+k];
    end
end
endgenerate

generate
```

```
for (k=0; k<half_correlators-1;k=k+1) begin : k_chain_2
  for (i=0; i<n_xors; i=i+1) begin : k_i_rx_xors
        assign rx_corr_in[(2*(k+1))-1][i] = delay_chain_mid_tap[i+k+1];
    end
end
endgenerate

// ### Correlators' XORs (outputs) ###
generate
genvar l;
for (k=0; k<half_correlators;k=k+1) begin : k_xor
    for (l=0; l<code_length;l=l+1) begin : l_xor
            xor l_xor0(rx_xor_out[k+k][l], rx_corr_in[k+k][l], codeword_0[l]);
    end
end
endgenerate

generate
for (k=0; k<half_correlators-1;k=k+1) begin : k_xor_2
    for (l=0; l<code_length;l=l+1) begin : l_xor
            xor l_xor1(rx_xor_out[k+k+1][l], rx_corr_in[k+k+1][l], codeword_0[l]);
    end
end
endgenerate

// ### Alternative Combinational Outputs for simulation and control output LED signals ###
// ### Signals full code detections for debugging purposes ###
assign rx_out = delay_chain_out[rx_corr_out_length-1];
assign rx_ok_0 = (codeword_0 == rx_corr_in[0][code_length-1:0] || codeword_0 == rx_corr_in[2][code_length-1:0] || codeword_0 == rx_corr_in[4][code_length-1:0] || codeword_0 == rx_corr_in[6][code_length-1:0] || codeword_0 == rx_corr_in[1][code_length-1:0] || codeword_0 == rx_corr_in[3][code_length-1:0] || codeword_0 == rx_corr_in[5][code_length-1:0])? 1'b1 : 1'b0;
assign rx_ok_1 = (codeword_1 == rx_corr_in[0][code_length-1:0] || codeword_1 == rx_corr_in[2][code_length-1:0] || codeword_1 == rx_corr_in[4][code_length-1:0] || codeword_1 == rx_corr_in[6][code_length-1:0] || codeword_1 == rx_corr_in[1][code_length-1:0] || codeword_1 == rx_corr_in[3][code_length-1:0] || codeword_1 == rx_corr_in[5][code_length-1:0])? 1'b1 : 1'b0;
endmodule
```

C.2.3 XOR module

Listing C.3: Verilog code of the XOR module used in the RX module

```
/// ### XOR module used in the TX and RX modules ###
module rx_xor (
    codeword_in,
    delay_chain_in,
    xor_out
);

// ### Inputs - Wires - Outputs - Local Params ###
input codeword_in;
input delay_chain_in;
output xor_out;

xor xor_rx(xor_out, delay_chain_in, codeword_in);

endmodule
```

C.2.4 Delay Buffer chain of *K* delay elements

Listing C.4: Verilog code of the delay chain with *K* delay elements

```verilog
// ### Delay Chain of K delay elements ###
// ### Includes a mid-chain tap for doubling the RX CTBV chain's resolution, ###
// ### the mid-chain tap is not used in the TX chain. ###
`timescale 1ps/1ps
module delay_chain #(
    parameter chain_length = 16 //default is K=16 delay elements
)(  in,
    mid_tap,
    out
);

// ### Inputs - Wires - Outputs - Local Params ###
input in;

wire in;
wire mid_tap;
wire out;
wire [chain_length-1:0] delay_chain;

output out;
output mid_tap;

// ### Delay Chains ###
assign delay_chain[0] = in;
assign out = delay_chain[chain_length-1];
assign mid_tap = delay_chain[{$clog2(2^(chain_length/2))}-1];

generate
      genvar i;
      for (i=0; i<chain_length-1; i=i+1) begin : delay_chain_wiring
         assign #500 delay_chain[i+1] = delay_chain[i];
         // ### Alternative instantiations ###
         //assign buf   #500 U_i(delay_chain[i+1],delay_chain[i]);
         //assign lcell #500 delay_i(.in(delay_chain[i]), .out(delay_chain[i+1]));
      end
endgenerate


endmodule
```

Appendix D

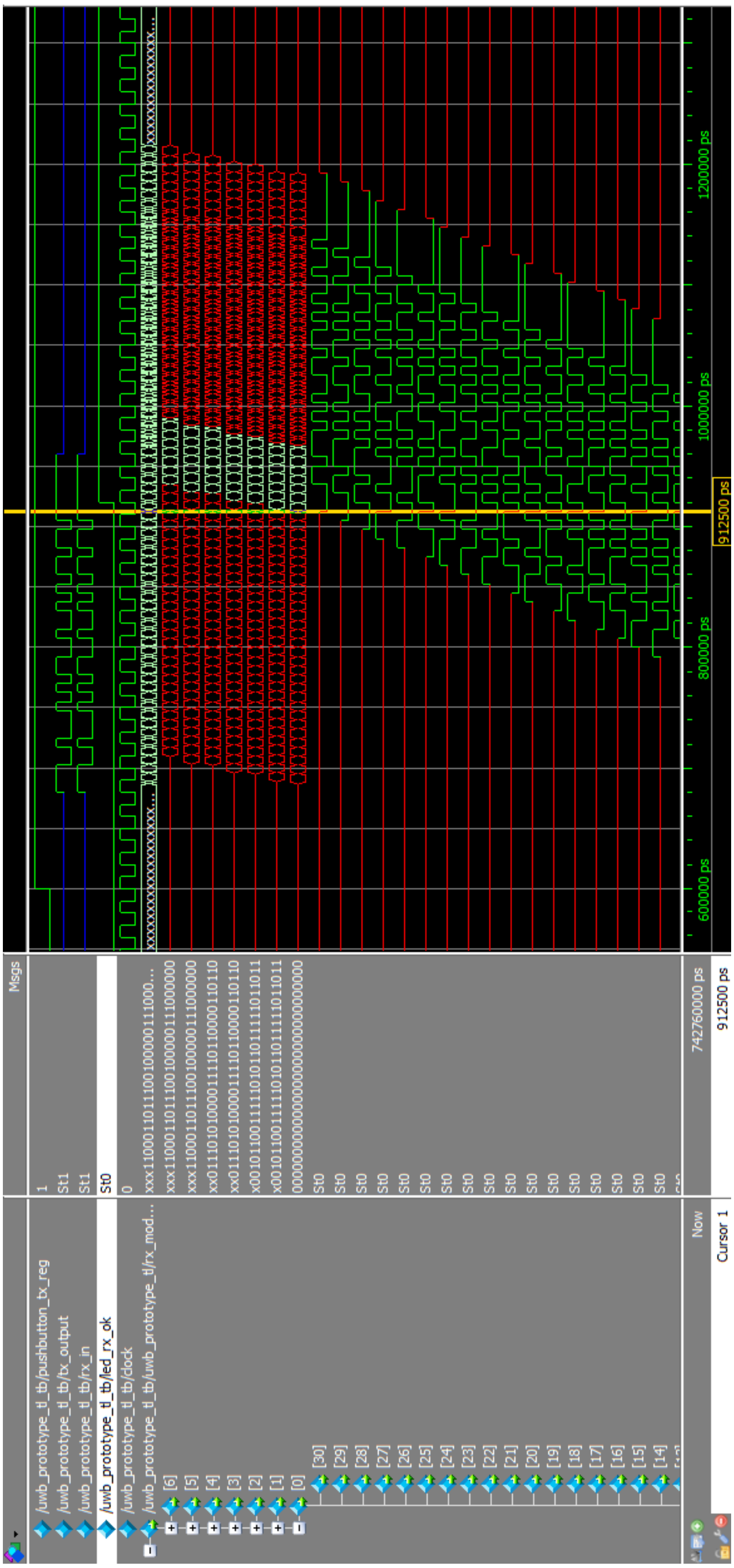
RTL Simulation Waveforms

Ranging Configuration

A larger version of Figure 4.2, *RTL simulation waveforms of the Ranging configuration*, is available on page 100.

Transceiver Configuration

A larger version of Figure 4.3, *RTL simulation waveforms of the Transceiver configuration*, is available on page 101.



Msgs

/uwb_prototype_tl_tb/pushbutton_tx_reg 1
/uwb_prototype_tl_tb/tx_output St1
/uwb_prototype_tl_tb/rx_in St1
/uwb_prototype_tl_tb/led_rx_ok St0
/uwb_prototype_tl_tb/clock 0
/uwb_prototype_tl_tb/uwb_prototype_tl/rx_mod... xxx11000110111001000000111000...
[6] xxx110001101110010000001110000000
[5] xxx110001101110010000001110000000
[4] xx0111010101000011110110000110110
[3] xx0111010101000011110110000110110
[2] x00101100111110101011011111011011
[1] x00101100111110101011011111011011
[0] 0000000000000000000000000000000
[30] St0
[29] St0
[28] St0
[27] St0
[26] St0
[25] St0
[24] St0
[23] St0
[22] St0
[21] St0
[20] St0
[19] St0
[18] St0
[17] St0
[16] St0
[15] St0
[14] St0

Now 742760000 ps
Cursor 1 912500 ps

Figure D.1: RTL simulation waveforms of the Ranging configuration. Upon receiving a send command (*pushbutton_tx_reg* in the test-bench) the TX module begins to generate a CTBV signal that propagates to its output and to the RX module's input. The correlators in the RX module start to toggle rapidly and a valid code is detected in the first correlator. The RX module control logic identifies that a valid code detection has occurred and signals this through the *led_rx_ok* output. The Gold code used was *31'b1110 0100 0101 0110 0100 1010 1101 101* and the transmission is MSB first.

.

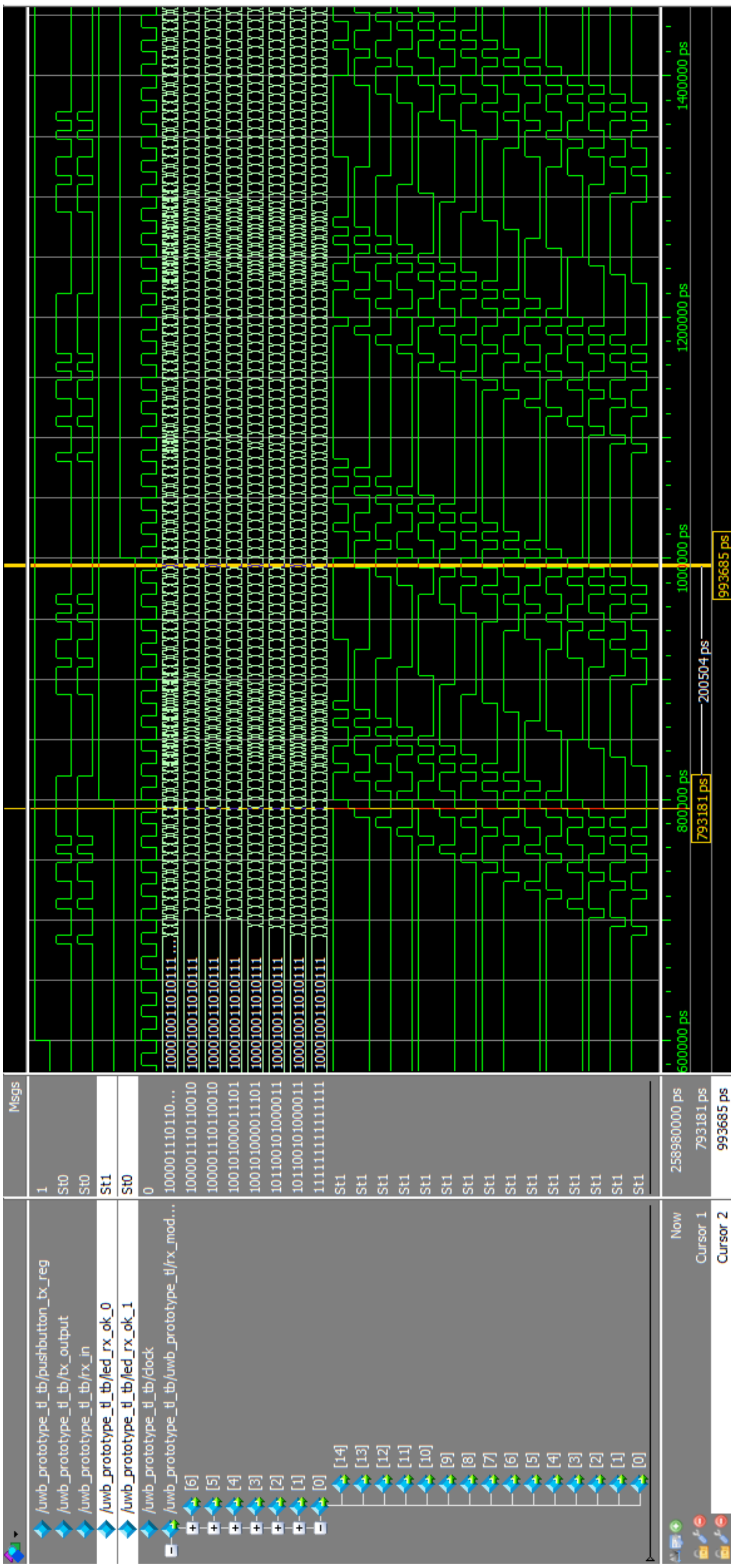


Figure D.2: RTL simulation waveforms of the Transceiver configuration. Upon receiving a send command (*pushbutton_tx_reg* in the test-bench), the TX module control logic instructs the TX module to begin transmission of a *chip* according to the first of eight payload bits, as defined by an UART test pattern loaded by the simulator into the Device Under Test (DUT). The correlators in the RX module start to toggle rapidly and the test pattern payload bits are sequentially detected, as indicated by *led_rx_ok_*0 and *led_rx_ok_*1 signals. The Kasami codes used, of length 15, were: *15'b1000 1001 1010 111* for payload bit *zero* and *15'b0111 0110 0101 000* for payload bit *one*. The UART sequence was *8'hAA*, or *8'b1010 1010*, and the transmission of the *chips* is LSB first.

References

[1] Hans G. Schantz. Three centuries of uwb antenna development. In *Three centuries of UWB antenna development*, jan 2012.

[2] G.B. Giannakis. Ultra-wideband communications: an idea whose time has come. In *Signal Processing Advances in Wireless Communications, 2003. SPAWC 2003. 4th IEEE Workshop on*, June 2003. doi:10.1109/SPAWC.2003.1318908.

[3] Federal Communications Commission. Revision of part 15 of the commission's rules regarding ultra-wideband transmission systems. FIRST REPORT AND ORDER ET Docket 98-153, FCC, Federal Communications Commission, Washington, D.C. 20554, mar 2002.

[4] European Telecommunications Standards Institute. Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd) using ultra wide band technology (uwb) for communications purposes. Harmonized European Standard ETSI EN 302 065-1, ETSI, jun 2013.

[5] Christian Wolff. Uwb-radar - figure 1: The uwb-radar transmits in a wide spectrum with very low power. Available at http://www.radartutorial.eu/02.basics/UWB%20Radar.en.html, jul 2015.

[6] M. Strackx, E. D'Agostino, P. Leroux, and P. Reynaert. Direct rf subsampling receivers enabling impulse-based uwb signals for breast cancer detection. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 62(2):144–148, Feb 2015. doi:10.1109/TCSII.2014.2387676.

[7] C. Le, T. Dogaru, Lam Nguyen, and M.A. Ressler. Ultrawideband (uwb) radar imaging of building interior: Measurements and predictions. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(5):1409–1420, May 2009. doi:10.1109/TGRS.2009.2016653.

[8] T.K. Kumar and P.K. Kumar. Uwb impulse radar for three dimensional through-the-wall radar imaging. In *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2011 IEEE International Conference on*, pages 1–5, Nov 2011. doi:10.1109/COMCAS.2011.6105810.

[9] D. Agarwal, C.R. Anderson, and P.M. Athanas. An 8 ghz ultra wideband transceiver prototyping testbed. In *Rapid System Prototyping, 2005. (RSP 2005). The 16th IEEE International Workshop on*, pages 121–127, June 2005. doi:10.1109/RSP.2005.12.

[10] P. Michalis, P. Ioannis, and M. Dimitrios. Design and implementation of an uwb digital transmitter based on the multiband ofdm physical layer proposal. In *Rapid System Prototyping, 2009. RSP '09. IEEE/IFIP International Symposium on*, pages 166–169, June 2009. doi:10.1109/RSP.2009.10.

[11] Talat Karim Minhas Md. Iqbal Hossain. Distance measurement using ultra wideband. Masters thesis, School of Technology and Health, Kungliga Tekniska Högskolan, feb 2012.

[12] Y. Salih-Alj, C. Despins, and S. Affes. Design considerations for an uwb computationally-efficient fast acquisition system for indoor line-of-sight ranging applications. *Wireless Communications, IEEE Transactions on*, 10(8):2776–2784, August 2011. doi:10.1109/TWC.2011.060811.102158.

[13] A. Oncu, B.B.M. Badalawa, and M. Fujishima. 22 ndash;29 ghz ultra-wideband cmos pulse generator for short-range radar applications. *Solid-State Circuits, IEEE Journal of*, 42(7):1464–1471, July 2007. doi:10.1109/JSSC.2007.899099.

[14] S. Sudalaiyandi, H.A. Hjortland, Tuan-Anh Vu, O. Naess, and T.S. Lande. Continuous-time high-precision ir-uwb ranging-system in 90 nm cmos. In *Solid State Circuits Conference (A-SSCC), 2012 IEEE Asian*, pages 349–352, Nov 2012. doi:10.1109/ASSCC.2012.6570786.

[15] Tor Sverre Lande. Impulse-based ultra-wide-band (uwb) radio systems and applications. Available at https://www.youtube.com/watch?v=mqq9HrLqYJY, jun 2008.

[16] Claus Limbodal. A spatial rake receiver for real-time uwb-ir applications. Master's thesis, UNIVERSITY OF OSLO, jul 2005.

[17] Matt Welborn and Kai Siwiak. Ultra-wideband tutorial. In Matthew Welborn and Kai Siwiak, editors, *Ultra-Wideband Tutorial*, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs). XtremeSpectrum and TimeDomain, mar 2002.

[18] Fu Delong, Huang Lu, Cai Li, and Lin Fujiang. *A 3–5 GHz BPSK transmitter for IR-UWB in 0.18 um CMOS*. University of Science and Technology of China, Hefei, China, sep 2010. URL: http://www.jos.ac.cn/bdtxben/ch/reader/create_pdf.aspx?file_no=10022401.

[19] Seyed Mohammad-Sajad SADOUGH. *A Tutorial on Ultra Wideband Modulation and Detection Schemes*. Shahid Beheshti University, apr 2009. URL: http://www.researchgate.net/file.PostFileLoader.html?id=544215eed5a3f2d9108b461d&assetKey=AS%3A272164186460162%401441900406696.

[20] Adrian POPA. *An Optimization of Gaussian UWB Pulses*. Gheorghe Asachi Technical University of Iasi, Suceava, Romania, may 2010. URL: http://www.dasconference.ro/cd2010/data/papers/B66.pdf.

[21] R. Gold. Optimal binary sequences for spread spectrum multiplexing (corresp.). *Information Theory, IEEE Transactions on*, 13(4):619–621, October 1967. doi:10.1109/TIT.1967.1054048.

[22] Inc The MathWorks. *Gold Sequence Generator*. The MathWorks, Inc, jan 2016. URL: http://www.mathworks.com/help/comm/ref/goldsequencegenerator.html.

[23] Mathuranathan Viswanathan. *Preferred Pairs m-sequences generation for Gold Codes*. Gaussian Waves, oct 2010. URL: http://www.gaussianwaves.com/2010/10/preferred-pairs-m-sequences-generation-for-gold-codes-2/.

[24] Inc The MathWorks. *Kasami Sequence Generator*. The MathWorks, Inc, jan 2016. URL: `http://www.mathworks.com/help/comm/ref/kasamisequencegenerator.html`.

[25] S. Sudalaiyandi, T.S. Lande, and H.A. Hjortland. A continuous-time ir-uwb rake receiver for coherent symbol detection. In *NORCHIP, 2012*, pages 1–4, Nov 2012. `doi:10.1109/NORCHP.2012.6403098`.

[26] H.A. Hjortland, D.T. Wisland, T.S. Lande, C. Limbodal, and K. Meisal. Cmos impulse radar. In *Norchip Conference, 2006. 24th*, pages 75–79, Nov 2006. `doi:10.1109/NORCHP.2006.329248`.

[27] H.A. Hjortland, D.T. Wisland, T.S. Lande, C. Limbodal, and K. Meisal. Thresholded samplers for uwb impulse radar. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 1210–1213, May 2007. `doi:10.1109/ISCAS.2007.378326`.

[28] Piljae Park, Sungdo Kim, Sungchul Woo, and Cheonsoo Kim. A high-resolution short-range cmos impulse radar for human walk tracking. In *Radio Frequency Integrated Circuits Symposium (RFIC), 2013 IEEE*, pages 9–12, June 2013. `doi:10.1109/RFIC.2013.6569508`.

[29] S. Brovoll, T. Berger, Y. Paichard, O. Aardal, T.S. Lande, and S.-E. Hamran. Time-lapse imaging of human heartbeats using uwb radar. In *Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE*, pages 142–145, Oct 2013. `doi:10.1109/BioCAS.2013.6679659`.

[30] S. Brovoll, T. Berger, Y. Paichard, O. Aardal, T.S. Lande, and S.-E. Hamran. Time-lapse imaging of human heart motion with switched array uwb radar. *Biomedical Circuits and Systems, IEEE Transactions on*, 8(5):704–715, Oct 2014. `doi:10.1109/TBCAS.2014.2359995`.

[31] H.A. Hjortland and T.S. Lande. Ctbv integrated impulse radio design for biomedical applications. *Biomedical Circuits and Systems, IEEE Transactions on*, 3(2):79–88, April 2009. `doi:10.1109/TBCAS.2009.2014962`.

[32] Ø. Aardal, S.-E. Hamran, T. Berger, J. Hammerstad, and T.S. Lande. Radar cross section of the human heartbeat and respiration in the 500mhz to 3ghz band. In *Radio and Wireless Symposium (RWS), 2011 IEEE*, pages 422–425, Jan 2011. `doi:10.1109/RWS.2011.5725466`.

[33] O. Aardal, Y. Paichard, S. Brovoll, T. Berger, T.S. Lande, and S.-E. Hamran. Physical working principles of medical radar. *Biomedical Engineering, IEEE Transactions on*, 60(4):1142–1149, April 2013. `doi:10.1109/TBME.2012.2228263`.

[34] K. Meisal, C. Limbodal, T.S. Lande, and D.T. Wisland. Cmos impulse radio receiver front-end. In *NORCHIP Conference, 2005. 23rd*, pages 133–136, Nov 2005. `doi:10.1109/NORCHP.2005.1597007`.

[35] O. Dahl, H.A. Hjortland, T.S. Lande, and D.T. Wisland. Close range impulse radio beamformers. In *Ultra-Wideband, 2009. ICUWB 2009. IEEE International Conference on*, pages 205–209, Sept 2009. `doi:10.1109/ICUWB.2009.5288755`.

[36] E. Bakken, T.S. Lande, and S. Holm. Real time uwb radar imaging using single chip transceivers. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 2461–2464, June 2014. `doi:10.1109/ISCAS.2014.6865671`.

[37] Kin Keung Lee, H.A. Hjortland, and T.S. Lande. Ir-uwb technology on next generation rfid systems. In *NORCHIP, 2011*, pages 1–4, Nov 2011. `doi:10.1109/NORCHP.2011.6126697`.

[38] Tuan Anh Vu, H.A. Hjortland, O. Nass, and T.S. Lande. A 3-5 ghz ir-uwb receiver front-end for wireless sensor networks. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2380–2383, May 2013. `doi:10.1109/ISCAS.2013.6572357`.

[39] D. Genschow. A time to digital converter for use in ultra wide band radar sensor nodes. In *Wireless Sensors and Sensor Networks (WiSNet), 2015 IEEE Topical Conference on*, pages 38–40, Jan 2015. `doi:10.1109/WISNET.2015.7127419`.

[40] Li Yueli, Yan Shaoshi, Zhu Guofu, Li Jiangyang, and Zhou Zhimin. Ultra wide band synthetic aperture radar real time processing with a subaperture nonlinear chirp scaling algorithm. In *Synthetic Aperture Radar (APSAR), 2011 3rd International Asia-Pacific Conference on*, pages 1–4, Sept 2011.

[41] D. Oloumi, M.I. Pettersson, P. Mousavi, and K. Rambabu. Imaging of oil-well perforations using uwb synthetic aperture radar. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(8):4510–4520, Aug 2015. `doi:10.1109/TGRS.2015.2400918`.

[42] Quanhua Liu, Yazhou Wang, and A.E. Fathy. Towards low cost, high speed data sampling module for multifunctional real-time uwb radar. *Aerospace and Electronic Systems, IEEE Transactions on*, 49(2):1301–1316, APRIL 2013. `doi:10.1109/TAES.2013.6494415`.

[43] M. Leib, W. Menzel, B. Schleicher, and H. Schumacher. Vital signs monitoring with a uwb radar based on a correlation receiver. In *Antennas and Propagation (EuCAP), 2010 Proceedings of the Fourth European Conference on*, pages 1–5, April 2010.

[44] X.Y. Wang, R.K. Dokania, and A.B. Apsel. Crystal-less duty-cycled-when-active ir-uwb transceivers. In *Electronics, Circuits and Systems (ICECS), 2014 21st IEEE International Conference on*, pages 850–853, Dec 2014. `doi:10.1109/ICECS.2014.7050119`.

[45] Jun Wang, Yinhua Rui, Zhizhen Huang, and Weidong Wang. Peak detection ultra-wideband transceiver using tunnel backward diode. In *Millimeter Waves, 2008. GSMM 2008. Global Symposium on*, pages 59–62, April 2008. `doi:10.1109/GSMM.2008.4534557`.

[46] K.R. Duncan and R. Etienne-Cummings. A low-cost cots uwb transceiver for biological applications achieves 50 mbps. In *Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE*, pages 169–172, Nov 2011. `doi:10.1109/BioCAS.2011.6107754`.

[47] Altera Corporation. *Cyclone II Device Handbook, Volume 1*, feb 2008.

[48] Alex Forencich. *GitHub Repository for verilog-uart*, 2015. URL: `https://github.com/alexforencich/verilog-uart`.

[49] Viorel MANEA, Radu DRAGOMIR, and Sorin PUSCOCI. *OOK and PPM modulations effects on bit error rate in terrestrial laser transmissions*. I.N.S.C.C., feb 2011.

[50] Altera Corporation. *Serial Configuration (EPCS) Devices Datasheet*, apr 2014.

[51] Altera Corporation. *High-Speed Differential Interfaces in Cyclone II Devices*, feb 2007.

[52] Altera Corporation. *Logic Element Definition*, 2015.

[53] Altera Corporation. *SCFIFO and DCFIFO IP Cores User Guide*, 2015.

[54] Altera Corporation. *Application Note 476: Impact of I/O Settings on Signal Integrity in Stratix III Devices*, 2007.

[55] JC. *Gold Code Generator*, may 2009. URL: `http://www.mathworks.com/matlabcentral/fileexchange/24018-gold-code-generator/content/goldcode_seq_gen.m`.

[56] Rodger E. Ziemer. *Fundamentals of Spread Spectrum Modulation.* Morgan & Claypool Publishers, 207.