University of Rome "La Sapienza"

School of Engineering
Graduate Course in Telecommunications
INFOCOM Department

Degree Thesis of Second Level about Ultra Wide Band and Optical Communication Systems
Academic Year 2004/2005

CONVERGENCE BETWEEN ULTRA WIDE BAND AND OPTICAL COMMUNICATION SYSTEMS

Tutors:
Prof. Rafael Pérez Jiménez
Prof. Maria-Gabriella Di Benedetto
Cotutor:
Eng. Santiago T. Pérez Suárez

Author:
Eugenio Manlio Ducci
Reg. N. 794700

La palabra progreso no tiene ningún sentido mientras haya niños infelices

Albert Einstein

A mi padre

Por su inestimable apoyo, su generosa sabiduría y su persistente espíritu de amistad, que serán por mí unas huellas permanentes que indican la justa dirección

A mi madre

Por su fuerza de ánimo, su riqueza espiritual y su bondad, que se quedarán para siempre en mi mundo interior

A mis hermanos

Que el camino que elegí hasta ahora pueda ser para ellos un buen ejemplo para su vida futura

Agradecimientos

Antes que nada mi gratitud se dirige hacia mi tutora Maria-Grabriella que me ha permitido comenzar y seguir serenamente esta maravillosa experiencia en un lugar en el cual la naturaleza ha sido tan generosa.

En segundo lugar, nunca podría haber ni planeado ni desarrollado el siguiente trabajo sin el amable soporte de Santiago acerca de mi inglés, mi español y mi proyecto; sin la sabiduría y cultura ecléctica de Rafael, que me han consentido en cada paso dirigirme hacia las soluciones más eficientes y originales; sin la preparación técnica y la cordialidad de José; sin el apoyo de Arthemy y Rufo, siempre buenos compañeros; y más que nada sin Ayaya, cuyas ayudas, consejos y charlas me han acompañado día tras día a través del camino de este proyecto.

En tercer lugar, quisiera dar las gracias a aquellos que formaron mi familia durante estos meses, compartiendo conmigo felicidades y tristezas: Davinia "Mariposa", Cristina "Chiuri", Espe "Ranza", Juan Luis "el Campesino", Miguel "el Diablo", Andrea, Simone, Reinhard, el "Compare" Fulvio, Sandra "Saxilo", Manu "Chao" y Daiana la cual, con cada gesto de amor desinteresado, me recuerda sin quererlo que nunca conocí una persona capaz de dar tanto amor. Ellos fueron mi familia, la familia Arehucas, y gracias a ellos pude añadir, a mi ya rica experencia profesional, aquella interior, sin duda la que más me formó como hombre y que yacerá indeleblemente en mi corazón de cara al futuro.

Por último no podría jamás despedirme sin agradecer, por su constante apoyo, a quienes son mis amigos de siempre: Ignazio y Ettore.

Eugenio Manlio Ducci *9 de Abril de 2006*

Index

CHAPTER 1 INTRODUCTION 1

1.1. Motivations underlying the work 2

1.2. Purposes 2

1.3. Description of the work 3

1.4. Material means and tools used 3

CHAPTER 2 THE OPTICAL COMMUNICATION SYSTEM 5

2.1. Introduction to the optical communications 6

2.2. The electrical driver 6

2.3. The optical emitter 6

2.4. The lenses 10

2.5. The channel 10

2.5.1. The optical fiber 10

2.5.2. The wireless channel 14

2.6. Noise and interference 19

2.7. Lenses and interferometer optical filter 23

2.8. The photodiode 24

2.9. The electrical processing 27

2.10. Appendix: the solitons 28

CHAPTER 3 MODULATION AND CODING SCHEMES 29

3.1. Introduction 30

3.2. Spread spectrum techniques 31

3.2.1. Direct Sequence Spread Spectrum 31

3.2.2. Time Hopping Spread Spectrum 35

3.3. On Off Keying modulation 37

3.4. Pulse Position Modulation 38

CHAPTER 4 THE ULTRA WIDE BAND COMMUNICATION SYSTEM 45

4.1. Introduction 46

4.2. Definition of UWB signal and its generation 48

4.2.1. Generation of TH-PPM-UWB signals and their PSD 49

4.2.2. Generation of TH-PAM-UWB signals and their PSD 53

4.2.3. Generation of MB-UWB signals and their PSD 55

4.3. Pulse shaping in UWB systems 56

CHAPTER 5 WIRELESS ACCESS CONTROL PROTOCOLS 59

5.1. Introduction 60

5.2. Random access protocols 62

5.2.1. Aloha 63

5.2.2. CSMA 66

5.2.3. In-Band CA: MACA 69

5.2.4. Out-Of-Band CA: BTMA 70

5.3. Scheduled protocols and hybrid protocols 73

5.4. Examples of standardized MAC protocols 74

5.4.1. MAC protocol for 802.11 standard: DFWMAC 74

5.4.2. MAC protocol for IrDA standard: IrLAP 79

5.4.3. UWB systems: MAC protocols for the standards 802.15.3 and 802.15.4 83

5.4.4. The $(UWB)^2$ MAC protocol 85

CHAPTER 6 IMPLEMENTATION OF $(UWB)^2$ OVER AN OPTICAL SYSTEM: HARDWARE'S DESIGN 93

6.1. Introduction 94

6.2. Optical transmitter 95

6.3. Optical receiver 102

6.4. Optical terminal 105

6.5. Optical system 106

6.6. Masters, slaves and relative EPLDs 108

CHAPTER 7 IMPLEMENTATION OF (UWB)^2 OVER AN OPTICAL SYSTEM: RESULTS 117

7.1. Introduction 118

7.2. (UWB)^2: software implementation 118

7.3. Optical System: hardware implementation 134

CHAPTER 8 BIBLIOGRAPHY AND WEB PAGES 141

8.1. Bibliography 142

8.2. Web Pages 145

ANNEX I. SCHEMATICS AND PCBS 147

AI.1. Schematic of the optical transmitter 149

AI.2. Schematic of the optical receiver 151

AI.3. Schematic of the optical terminal 153

AI.4. Schematic of the EPLD relative to the master number one 155

AI.5. Schematic of the EPLD relative to the master number two 157

AI.6. Schematic of the EPLD relative to the slave number one 159

AI.7. Schematic of the EPLD relative to the slave number two 161

AI.8. Schematic of the EPLDs of the optical system 163

AI.9. PCB of the optical transmitter 165

AI.10. PCB of the optical receiver 169

AI.11. PCB of the optical terminal 173

AI.12. PCB of the optical system 177

AI.13. PCB of the EPLDs of the optical system 183

AI.14. Schematic of the drive logic of the master number 1 and number 2 189

AI.15. Schematic of the drive logic of the slave number 1 and number 2 191

AI.16. Schematic of the 4 taps shift register 193

AI.17. Schematic of the 16 taps shift register 195

AI.18. Schematic of the Pseudo Random Number Generator 197

AI.19. Schematic of the optical system used in the software simulation 199

ANNEX II. COMPONENTS'S SPECIFICATIONS..... 201

AII.1. Technical specifications of the quad 2-input AND gate CD74AC08..... 202

AII.2. Technical specifications of the Infrared LED L7558 series..... 203

AII.3. Technical specifications of the Si PIN photodiode S6468 series..... 205

AII.4. Technical specifications of the Amplifier MAX410..... 207

AII.5. Technical specifications of the Differential Comparator LM360..... 222

AII.6. Technical specification of the 5.0V TTL Clock Oscillator F1100E..... 230

AII.7. Altera MAX 9000 programmable logic device family data sheet..... 231

AII.8. Altera device package information for EPM9320ALC84-10..... 239

ANNEX III. VHDL CODE..... 243

AIII.1. Masters: VHDL code of the block "Recognize Header"..... 244

AIII.2. Masters: VHDL code of the block "Buffer"..... 247

AIII.3. Masters: VHDL code of the block "Parallel Serial Converter"..... 250

AIII.4. Masters: VHDL code of the block "Multiplexer"..... 253

AIII.5. Masters: VHDL code of the block "Compare Header"..... 254

AIII.6. Masters: VHDL code of the block "Vector to Integer Converter"..... 255

AIII.7. Slaves: VHDL code of the block "Recognize Header "..... 256

AIII.8. Slaves: VHDL code of the block "Buffer"..... 259

AIII.9. Slaves: VHDL code of the block "Parallel Serial Converter"..... 261

AIII.10. Slaves: VHDL code of the block "Multiplexer"..... 263

AIII.11. Slaves: VHDL code of the block "Compare Header"..... 264

AIII.12. Slaves: VHDL code of the block "Compare Payload"..... 265

ANNEX IV. ACRONYMS..... 267

AIV.1. Acronyms..... 268

Figures

Figure 2.1: Block Diagram of an Optical Communication System 6
Figure 2.2: Effect of the temperature changing on the LED linearity 7
Figure 2.3: Effect of the temperature changing on the optical spectra 7
Figure 2.4: Spontaneous emission 8
Figure 2.5: Stimulated emission 8
Figure 2.6: Effect of the current threshold 9
Figure 2.7: Effect of the temperature on the input-output curve 9
Figure 2.8: The lens are used to focus the emitted beam on a reduced area, just like a fiber's core 10
Figure 2.9: Typical aspect of an optical fiber 11
Figure 2.10: Multiple reflections at core-cladding separation interface 11
Figure 2.11: Attenuation as function of the frequency 12
Figure 2.12: Effect of the intermodal dispersion 12
Figure 2.13: Curves of several chromatic dispersion parameters 13
Figure 2.14: Two rays model 15
Figure 2.15: Blade of knife model 16
Figure 2.16: Indoor application scenarios 17
Figure 2.17: Outdoor application scenarios (MAN UOWC) 18
Figure 2.18: Spectrum of the noise optical sources 19
Figure 2.19: Solar Spectral Irradiance through the atmosphere 20
Figure 2.20: Typical electrical spectrum of a 60 W, 50 Hz tungsten-filament 21
Figure 2.21: Power spectral density of shot noise of a fluorescent lamp 22
Figure 2.22: Example of performance obtained with several modulation techniques 23
Figure 2.23: Example of multipath effect inside the lenses 23
Figure 2.24: WDM access's scheme 24
Figure 2.25: Absorption of a photon 25
Figure 2.26: Effect of the temperature increasing on the dark current 25

Figure 3.1: Modulation and coding schemes 30
Figure 3.2: Representation of the baseband transmission scheme of a DSSS system 31
Figure 3.3: Representation of the baseband DSSS system 32
Figure 3.4: Power spectral densities in the points A and B of the DSSS system 32
Figure 3.5: Effect in the time domain of the bit-chips multiplication 33
Figure 3.6: Block diagram of a THSS system 35
Figure 3.7: Example of coding of two bits "1" using the THSS technique. The pulse waveform used, is a second order gaussian derivative 36
Figure 3.8: Examples of two signals modulated OOK 37
Figure 3.9: Effect on the bandwidth of reducing the duty cycle γ 38
Figure 3.10: Examples of two signals L-PPM modulated 39
Figure 3.11: Symbol Error Probability for L-ary PPM 40
Figure 3.12: Power requirements obtained varying the number of levels L 41
Figure 3.13: Comparison between OOK and L-PPM performance 41
Figure 3.14: Comparison between OOK and L-PPM bandwidth requirements 42
Figure 3.15: Comparison of OOK and L-PPM system when is present ambient noise 43

Figure 4.1: FCC indoor and outdoor emission masks 46
Figure 4.2: Energy spectrum of a modulated signal 48
Figure 4.3: Rectangular modulated signals and their spectrum 48
Figure 4.4: Block scheme for a TH-PPM-UWB transmission system 49
Figure 4.5: Example of a 2PPM-TH-UWB emitted signal 50
Figure 4.6: Graphical representation of a 2PPM-TH-UWB signal with $N_s = N_p = 3$ 51
Figure 4.7: Effect of increasing $N_s = N_p$ *on a 2PPM-TH-UWB signal* 52
Figure 4.8: Block scheme for a DS-PAM-UWB transmission system 53
Figure 4.9: Example of a 2PAM-DS-UWB emitted signal 54
Figure 4.10: Effect of increasing N_p*, maintaining constant* N_s*, on a 2PAM-DS-UWB signal* 55
Figure 4.11: PSD of an MB-OFDM-UWB signal 56
Figure 4.12: Effect of changing the shaper factor of the gaussian pulse 57
Figure 4.13: Effect on the peak frequency of modifying the differentiation order 57
Figure 4.14: Effect on the PSD of modifying the differentiation order 58

Figure 5.1: ISO/OSI stack model for a data network 60
Figure 5.2: Wireless MAC protocols 61
Figure 5.3: Examples of possible scenarios for transmitting of packets 63
Figure 5.4: Unslotted Aloha and ideal MAC throughputs 63
Figure 5.5: Example of scenario in the Aloha slotted protocol 64
Figure 5.6: Slotted Aloha, unslotted Aloha and ideal MAC throughputs 64
Figure 5.7: Unslotted Aloha, Unslotted Aloha with capture effect and ideal MAC 65
Figure 5.8: Typical cause of collision for CSMA protocol 66
Figure 5.9: Comparison between the throughputs seen until now 67
Figure 5.10: Example of scenario relative to the hidden terminal phenomena 68
Figure 5.11: An other hidden terminal scenario potentially subjected to collision 68
Figure 5.12: Example of scenario for the exposed terminal issue 69
Figure 5.13: Steps in the scenario considered for the handshaking 69
Figure 5.14: Example of centralized architecture using BTMA protocol 71
Figure 5.15: Example of the amplification of the exposed terminal problem 72
Figure 5.16: Main characteristics of 802.11 wireless LAN standard 74
Figure 5.17: Example of a 2-way handshaking in the DFWMAC 76
Figure 5.18: Example of a 4-way handshaking in the DFWMAC 77
Figure 5.19: Rates available in the IrDA physical layer evolution 79
Figure 5.20: The IrDA architecture 80
Figure 5.21: IrLAP logical flow between nonoperational and operational mode 80
Figure 5.22: The discovery procedure in IrDA 81
Figure 5.23: Connection establishment 82
Figure 5.24: Superframe used by PNC for the piconet global timing in 802.15.3 MAC 84
Figure 5.25: Superframe used by a PAN Coordinator in 802.15.4 MAC (beacon mode) 85
Figure 5.26: Structure of the LE PDU in (UWB)^2 87
Figure 5.27: Structure of the DATA PDU in (UWB)^2 87
Figure 5.28: Flow chart of the transmission procedure in (UWB)^2 89
Figure 5.29: Structure of the LC PDU in (UWB)^2 90
Figure 5.30: Structure of the ACK PDU in (UWB)^2 91
Figure 5.31: Flow chart of the reception procedure in (UWB)^2 92

Figure 6.1: Circuital scheme of the optical transmitter 95
Figure 6.2: Some input-output characteristics of the LED L7558 useful to design the physical layer of the optical system 96
Figure 6.3: Physical appearance of the optical transmitter 97
Figure 6.4: Outputs of the AND Port CD74AC08 used for the LED array 98
Figure 6.5: Circuital scheme of the optical transmitter using the array of two LEDs 98
Figure 6.6: Physical aspect of the optical transmitter using the array of LEDs 99
Figure 6.7: Output of the AND gate CD74AC08 99
Figure 6.8: Output of the circuit of pre-emphasis 100
Figure 6.9: Block scheme of the physical optical system 101
Figure 6.10: Hardware implementation of the block scheme represented in the Figure 6.9 101
Figure 6.11: Circuital scheme of the optical receiver 102
Figure 6.12: Appearance of the optical receiver 103
Figure 6.13: Input to the amplifier 103
Figure 6.14: Output of the amplifier 104
Figure 6.15: Output of the comparator 104
Figure 6.16: Circuital scheme of the optical terminal 105
Figure 6.17: Physical aspect of the optical terminal 106
Figure 6.18: Physical aspect of the optical system 107
Figure 6.19: Graphical representation of the connections of the pins for the masters 111
Figure 6.20: Graphical representation of the connections of the pins for the slaves 112
Figure 6.21: Schematic of the EPLD related to the master #1 114
Figure 6.22: Schematic of the EPLD related to the slave #1 115

Figure 7.1: Throughputs Obtained Simulating the UWB^2 Protocol 121
Figure 7.2: LE and LC packets relative to both the pairs of terminals and obtained through the Max+Plus simulator 122
Figure 7.3: DATA and ACK packets relative to both the pairs of terminals 123
Figure 7.4: Example of exchange of the packets defined by the (UWB)^2 protocol for the couple of terminal master #2 - slave #2 125
Figure 7.5: Example of sequential transmissions without errors 126
Figure 7.6: A first example of packet collision between a DATA packet and a LE packet 128

Figure 7.7: Another example of packet collision between a DATA packet and a LE packet 130
Figure 7.8: An interesting example of packet collision..... 132
Figure 7.9: An example of a set of sequential transmissions and receptions 133
Figure 7.10: Operative context of work in the laboratory 134
Figure 7.11: A visual particularity of the Figure 7.10..... 135
Figure 7.12: An other visual particularity of the Figure 7.10 135
Figure 7.13: Signals obtained at the output of the terminals in absence of interference . 136
Figure 7.14: A first situation in which the terminals begin to be affected by interference137
Figure 7.15: A second situation in which the interference begins to be noticeable..... 138
Figure 7.16: A third situation in which the interference affects strongly each terminal .. 138
Figure 7.17: A fourth situation in which the interference affects strongly each terminal 139

Tables

Table 2.1: Coefficient of attenuation at several and typical wavelength *12*
Table 2.2: General characteristics of IR and RF channel *14*
Table 2.3: Parameter γ as function of the building structure *16*
Table 2.4: Effect of optical filtering in the d.c. photocurrent *21*

Table 3.1: Truth table for the $\overline{XOR}$ multiplication *35*

Table 5.1: Main characteristics for the IrDA LAN standard *79*

Table 6.1: Roles of the terminals involved in the simulations *106*
Table 6.2: Values chosen for some fixed fields defined by the $(UWB)^2$ protocol *110*
Table 6.3: Main features of the EPLDs belonging to the family MAX 9320 ..... *111*
Table 6.4: User pins employed in to the EPLDs of the masters *113*
Table 6.5: User pins employed in to the EPLDs of the slaves *113*

Table 7.1: Throughputs obtained by the software simulation of the $(UWB)^2$ protocol *120*

Chapter 1

Introduction

Contents

1.1 Motivations underlying the work..... 2
1.2 Purposes 2
1.3 Description of the work..... 3
1.4 Material means and tools used 3

1.1. Motivations underlying the work

Nowadays, as never, it is becoming necessary to develop technological systems that satisfy the requirements of the more various operative contests. This is evident giving a look to the deep changes that technological devices like notebook, PCs, PDAs, mobiles, cameras, storage devices, etc., have brought and are bringing in the common life. At first coming out, they do not represent a necessity, but with the daily use one can experience the utility, the comfort, of those commodities. In this case it can be created a collective demand regarding improvements, modifies and even new features of those devices. We could affirm that sometimes the technology creates itself and, obviously, it will have to conform its characteristics according to the heterogeneous requests of the market. Among those we can cite the necessity to transmit using high or low bit rate, to get directional or diffused transmission systems, to cope efficiently to the interference generated either by other pre-existing systems or by other devices using the same technology, to manage an access more or less regulated, to allow scalability and interoperability among different systems, etc. In order to get the previous "system specifications" often, and quite obviously, a join between several existent technological solutions are employed. We could call them "hybrid solutions". Just for giving an example, we can mention that the IrDA standard, even if having its MAC protocol, can employ optionally the RTS/CTS mechanism, used as main access strategy by the 802.11 standard.

1.2. Purposes

Following the "hybrid solutions" philosophy presented at the end of the previous paragraph, the present investigation work wants to analyze the behaviour, the efficiency and the characteristics of adaptability of the $(UWB)^2$ MAC protocol when it is implemented over an optical system. The $(UWB)^2$ MAC protocol, based on the Aloha access strategy and described in (Di Benedetto et al., 2004), has been developed for working over UWB systems, those relating to the 802.15.4 standard, characterized by low bit rate over medium-long range.

What we want to develop is an Optical Communication System (OCS) formed by four terminals. As to the physical layer, each terminal can both transmit, and receive; hence, at least as regards this layer, differences between them do not exist. At the MAC layer, two terminals are masters of the communication, and the remaining two are slaves.

This choice was a hardware constraint related to the restricted resources of the EPLDs. The terminals transmit to a given data rates, using an OOK modulation. The presence of the interference between the slaves is ensured through the use in the masters of an array of two LEDs.

1.3. Description of the work

First of all, it has been undertaken the research of the documentation that could provide the necessary theoretical groundwork over which to develop the work. This study concerned both the Ultra Wide Band technology and the Optical Communication Systems. As regards the UWB systems, the interest of the writer has been directed both to the high bit rate over short range case and to the low bit rate over medium to long range version, whose respective standards are 802.15.3 and 802.15.4. Secondly, a study and analysis of the MAC protocol $(UWB)^2$ has been undertaken. Finally, as to the optical systems, the attention has been focused on the IrDA standard and on its successive versions.

As second step, it has been developed, using the software *Protel* for electronic design, the physical part of the optical transmission system, taking into account that $(UWB)^2$ is a MAC protocol developed for the low bit rate over medium to long range case and that therefore the data rate of the system could be set to 32 Kbps.

Subsequently, using the software application *Max+Plus II*, $(UWB)^2$ has been implemented and simulated in the calculator, ready to be transferred into the EPLDs.

Finally, the optical system has been built up.

1.4. Material means and tools used

As regards the software, the following applications were used:

- *Microsoft Office Word 2003, Microsoft Office PowerPoint 2003 and Microsoft Office Excel 2003*
- *Adobe Photoshop CS*
- *Adobe Acrobat 7.0 Professional*
- *Matlab 7.0*
- *Protel 99 SE*
- *Max+Plus II*

As to the hardware employed for the project, it has been made use of:

- Notebook
- Power supplies
- Signal generators
- Analogic oscilloscopes
- Digital oscilloscopes
- Spectrum analyzers
- Electronic components and tools as: integrated circuits, LEDs, photodiodes, EPLDs, resistances, capacitors, welders, cables, screwdrivers, etc.
- Digital Camera

Chapter 2

The Optical Communication System

Contents

2.1	**Introduction to the optical communications**	**6**
2.2	**The electrical driver**	**6**
2.3	**The optical emitter**	**6**
2.4	**The lenses**	**10**
2.5	**The channel**	**10**
	2.5.1 The optical fiber	10
	2.5.2 The wireless channel	14
2.6	**Noise and interference**	**19**
2.7	**Lenses and interferometer optical filter**	**23**
2.8	**The photodiode**	**24**
2.9	**The electrical processing**	**27**
2.10	**Appendix: the solitons**	**28**

2.1. Introduction to the optical communications

The physical layer of the system in analysis is represented by an *OCS* (*Optical Communication System*) whose block diagram is represented in the following figure:

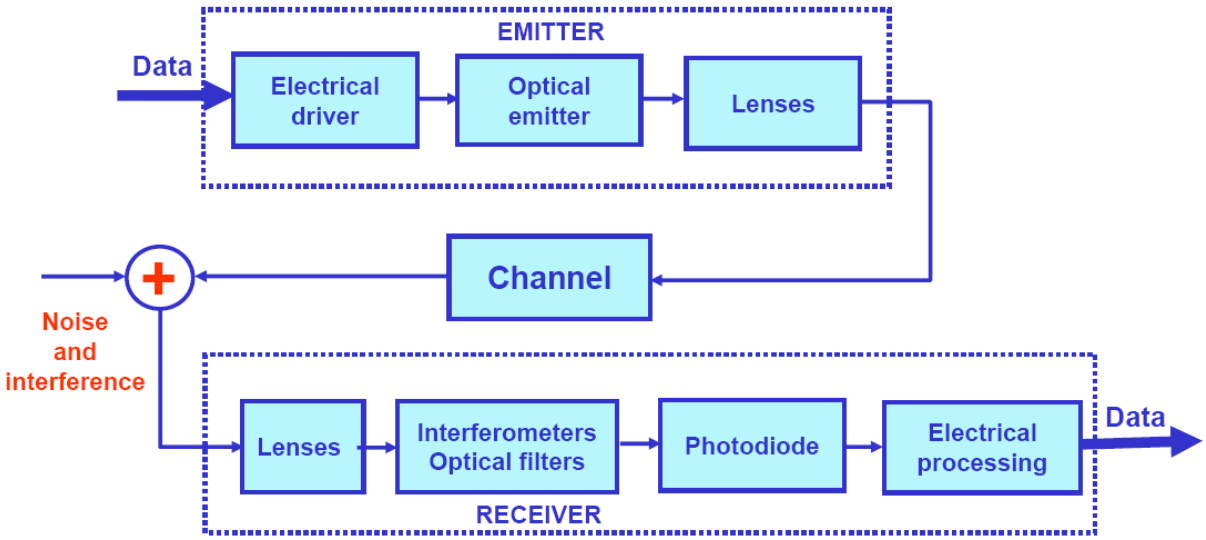


Figure 2.1: *Block Diagram of an Optical Communication System*

In the continue of this chapter, we will give a description, with some detail, of each component that composes the above diagram.

2.2. The electrical driver

The Electrical driver is necessary to provide the following function:

- To feed the optical emitter by the needed current
- To adapts the input signal as to power, modulation, etc
- Since the optical emitters are susceptible to temperature changes, it is needed a thermal control in order to keep constant the emitted optical power

2.3. The optical emitter

Nowadays two important classes of emitter are used: *LEDs* and *Laser Diodes*.

The main characteristics of a LED are:

\

- Typically, it is characterized by low cost
- The input-output curve is temperature dependent being anyhow more robust to the external conditions than laser diodes

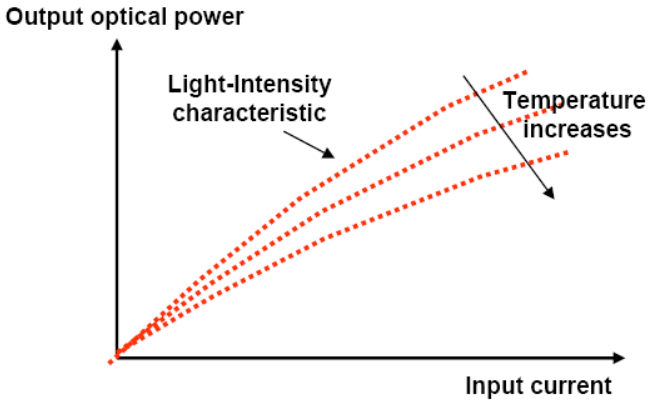


Figure 2.2: *Effect of the temperature changing on the LED linearity*

- With reference to the electrical driver, it is electrically simple to use and control
- Depending on the emitting window, the available transmission rates reach several hundred of MHz
- It produces scattered incoherent light
- It needs subsequent lens in order to shore up directive the Led beam
- Its use is restricted to large core fibers
- Broad spectral width, which is temperature dependent, generates material dispersion

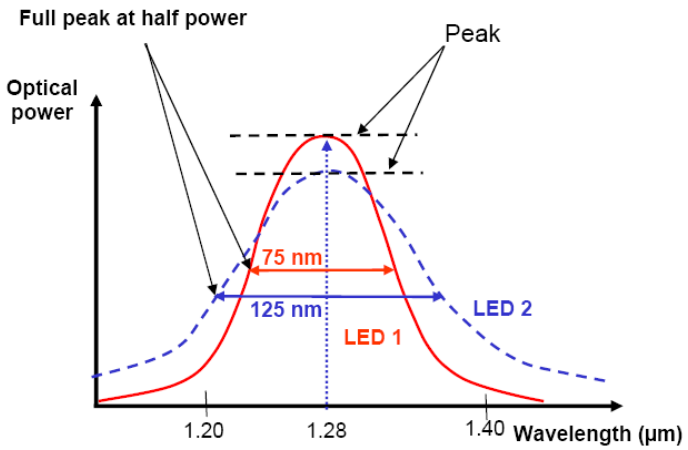


Figure 2.3: *Effect of the temperature changing on the optical spectra*

As regards the laser diode, its major characteristics are:

- It is based on spontaneous and stimulated emission

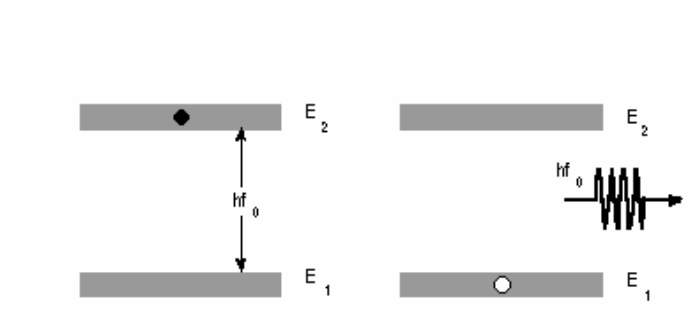


Figure 2.4: *Spontaneous emission*

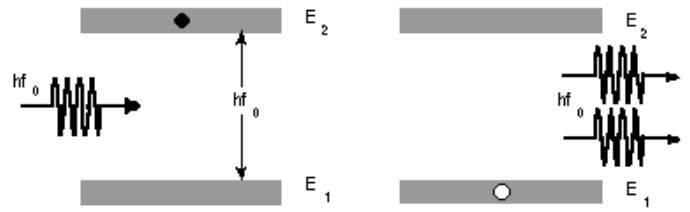


Figure 2.5: *Stimulated emission*

- The laser presents a current threshold below of which dominates the spontaneous emission

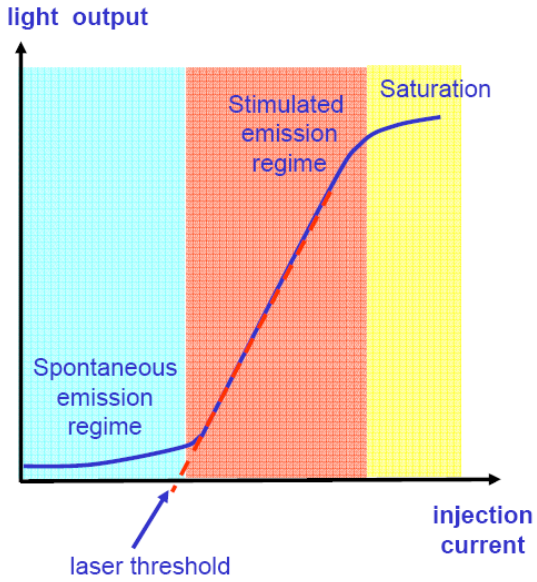
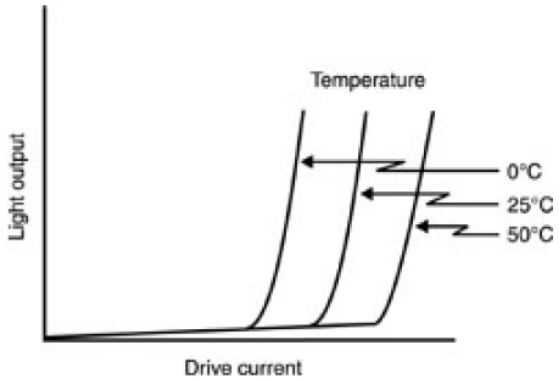


Figure 2.6: *Effect of the current threshold*

- Typically, it is characterized by high cost
- The available transmission rates arrive up to tens of GHz
- It produces coherent light (in phase, frequency and direction)
- It allows obtaining high optical power output
- Because of its dependence from external changes (as temperature), it requires more complex control than a LED



$$I_t(T_2) = I_t(T_1)\exp[(T_2 - T_1)/T_0]$$

where $T_0 = 150\ ^oK$

Figure 2.7: *Effect of the temperature on the input-output curve*

2.4. The lenses

In order to match as better as possible the emitter with the channel (typically, as we will show, fibers or wireless channel), lenses are interposed between the optical emitter and the channel.

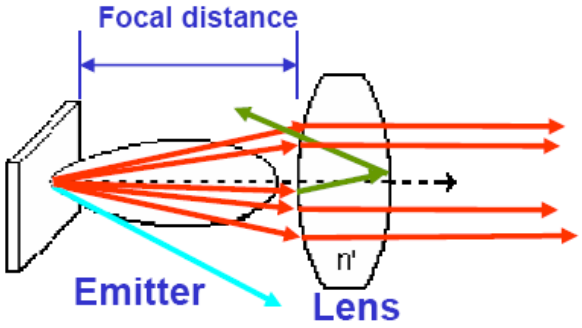


Figure 2.8: *The lens are used to focus the emitted beam on a reduced area, just like a fiber's core*

Though using lenses, we experiment three sources of loss:

- Since the emitter must be positioned at the focal distance, if some imprecision on its spatial position occurs, some rays are not concentrated and may be lost
- Because of the imperfections of the lenses, some ray may be sent backwards
- The material composing the lenses, constitutes itself a first attenuation in the rays path

2.5. The channel

As mentioned before, the channels that we are going to consider in the prosecution will be wireless channels and fibers.

2.5.1. The optical fiber

A fiber is composed by an internal core, an external cladding and a covering that preserves the integrity of the device from the external environment.

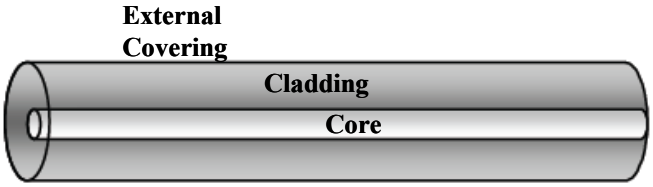


Figure 2.9: *Typical aspect of an optical fiber*

As known, the basic functioning principle of a fiber is to guide the luminous rays in the core by multiple reflections on the cladding internal surface that works as a mirror. In order to achieve total reflection, as application of the Snell law

$$n_1 \sin\theta_1 = n_2 \sin\theta_2 ,$$

we have to use a core with a reflection index $n_1 > n_2$. In this way we may define the *critical angle* as

$$\theta_c = \arcsin\left(n_2 / n_1\right)$$

over which occurs total reflection between core and cladding.

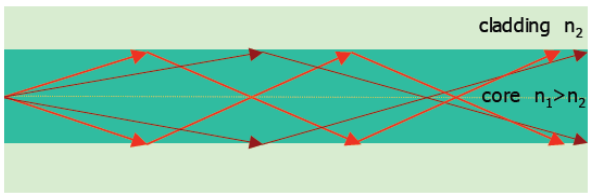


Figure 2.10: *Multiple reflections at core-cladding separation interface*

The most used fibers are the *step index fibers* and the *graded index fibers* and they can be either single mode or multi mode, having respectively a core of diameter within the range of 3-20 μm and 50-400 μm.

As shown in the following figure, depending on the window used for the transmission, we can experience different values of attenuation.

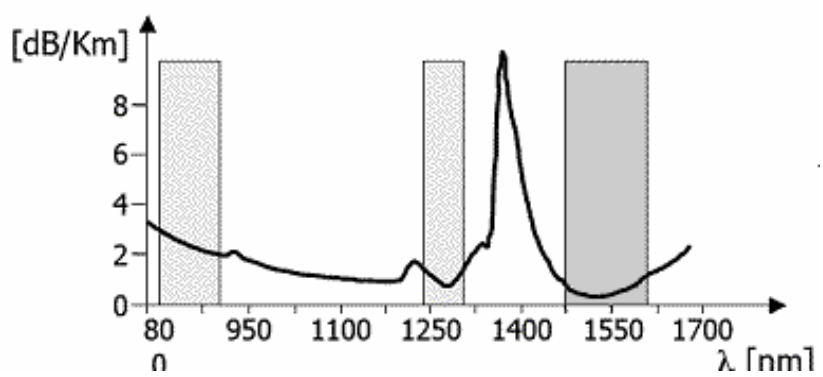


Figure 2.11: *Attenuation as function of the frequency*

Typical values are those shown in the following table:

λ	α_{dB}
0.85 μm	2.5 dB/Km
1.31 μm	0.4 dB/Km
1.55 μm	0.25 dB/Km

Table 2.1: *Coefficient of attenuation at several and typical wavelength*

In addition to the attenuation, two aspects that a fiber designer has to valuate closely are the intermodal and the chromatic dispersion. The first one is generated by the presence of multiple modes propagating in the core. Since each mode follows a specific path the original waveform of the signal transmitted is modified as shown in the following figure:

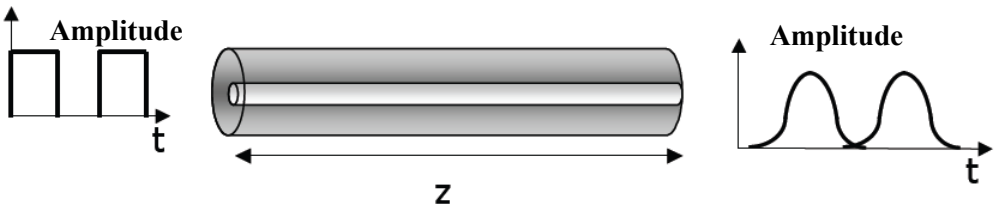


Figure 2.12: *Effect of the intermodal dispersion*

Nowadays, this issue is overcome using mono mode fibers that allow a unique mode of propagation.

The chromatic dispersion is generated because of using of light emitters not perfectly monochromatic and thus having several propagation velocities on dependence of the wavelength.

In order to mitigate this issue, we can use either an emitter characterized by a reduced spectrum, like a good laser, or we can take into account that fibers lead to different chromatic dispersions depending on the used windows. In particular, it is introduced a chromatic dispersion parameter $D(\lambda) = D_M(\lambda) + D_W(\lambda)$, that measures the intensity of the chromatic dispersion. $D_M(\lambda)$ and $D_W(\lambda)$ are respectively the material dispersion and the wave guide dispersion. Typically, $D(\lambda) = 0$ for $\lambda = 1.310\,\mu m$ (second window) but we have also special fibers, as *DS fibers* (Dispersion Shifted) and *DF fibers* (Dispersion Flattened), that allow modifying the dependence of the chromatic dispersion parameter from the wavelength λ. As shown in the following figure, DS fibers get the zero dispersion in the third window ($\lambda = 1.550\,\mu m$) whereas DF fibers have the chromatic dispersion parameter almost constant.

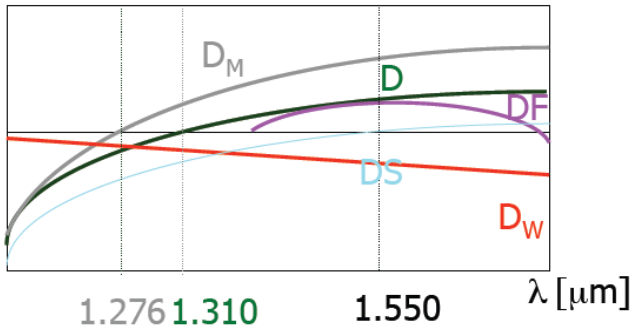


Figure 2.13: *Curves of several chromatic dispersion parameters*

For example, we may desire to get simultaneously a minimum both as to chromatic dispersion and for the attenuation, getting this purpose by means of a DS fiber.

Finally, it is important to notice that there are some other, important, effects limiting the performance of the transmission over optical fibers as, for example:

- *SPM* (Self Phase Modulation): in the propagation, the impulse suffers a non stationary phase displacement
- *XPM* (Cross Phase Modulation): if two or more channels are transmitted using different wavelengths simultaneously within the same fiber, an impulse can suffer a non linear phase displacement that depends on the intensity of the signals transmitted on the other channels

- *FWM* (Four Wave Mixing): it occurs when are presented in the fiber three simultaneous channels, with ω_1, ω_2 and ω_3 as respective pulsations, generating a new channel with pulsation $\omega_{FWM} = \omega_1 + \omega_2 - \omega_3$

These and other non linear effects are currently important matter of analysis and studies of physical researchers.

2.5.2. The wireless channel

Because of its intrinsic nature, the wireless channel presents specific properties and characteristics that are mostly dependent by two variables: the frequency of the radiation and the orography of the region. As to the frequency (or, equivalently, the wavelength), we must remember that, changing it, mutates the way to react of the transmission medium. As examples, the frequency can set variations in transmittance, absorptance, reflectance, attenuation and, since, in our channel modelling. For a fixed orography and taking into account IR and RF radiation (which will be mostly analysed in this work), we can summarize the effect of changing the frequency of radiation in the following table.

Properties	**IR**	**RF**
Paraxial gaussian approximation	Yes	No
Multi-path fading	No	Yes
Multi-path dispersion	Yes	Yes
Transmission through walls and obstacles	No	Yes
Dominant noise source	Illumination / Other users	Other users
Propagation losses	Very high	High

Table 2.2: *General characteristics of IR and RF channel*

For fixed wavelength, the orography heavily determines our channel model. Numerous models have been introduced and, in the following of this subsection, we will provide a sufficiently deep description of the mostly used.

In literature typically we find two distinct kinds of models: deterministic (or statistic) models and empiric (or site specific) models.

Deterministic models

They require a large amount of data with reference to the geometry, ground profile, position and characteristics of building, obstacles, etc. The use of these models leads to a great computational complexity and calculus resource but they provide to better and more accurate results.

Nowadays, the main and more used deterministic models are:

- Finite Differences Time Domain (FDTD)
- Method of Moment (MoM)

used principally for characterizing situations limited in space, as propagation in reduced environments or in proximity of the antennas (near field), and

- Ray tracing techniques

that is made use for

- link in sight (LOS)
- point to point links whose major contributes are the LOS ray and a unique reflected ray (ray tracing two rays model)

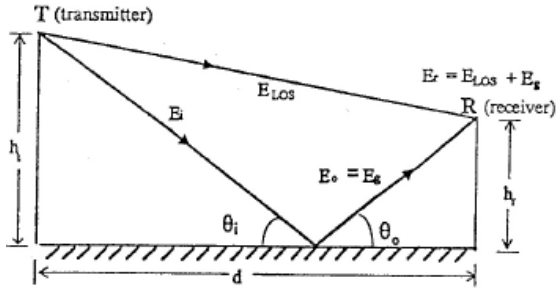


Figure 2.14: *Two rays model*

- point to point links in which the link can be represented by means of a single and simple diffraction mechanism generated by not much hills or buildings (blade of knife model)

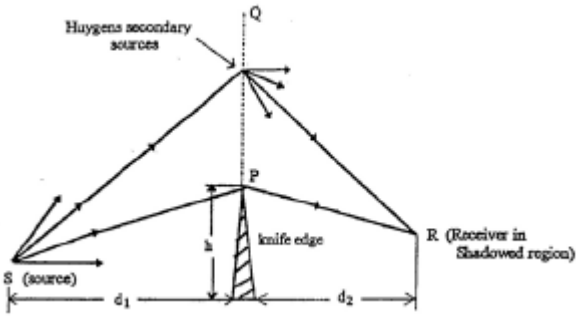


Figure 2.15: *Blade of knife model*

Empiric models

They are subdivided in the two big categories of models for indoor and outdoor links and they are based on simple analytic laws for the expected field that are derived from field measures performed with appropriate instruments.

As to the indoor models, they are based on an application of the *path loss* formula that, as known, is:

$$PL(dB) = 10\log_{10}\frac{P_T}{P_R} = 10\log_{10}\frac{(4\pi)^2 R^2}{G_T G_R \lambda^2}$$

and that, with some passage, can be rewritten as

$$PL(dB) = PL(R_0) + 10\log_{10}\frac{R^2}{R_0^2}$$

In this model, the key is to add a parameter γ whose role is to specify the particular indoor scenario. The parameter γ depends from the used frequency and the physical material used for the walls of the building but, typically, it has the following values:

γ	Typical situation
1.5-1.8	Transmitter and receiver are in light of sight
2	Free space
3-4	Transmitter and receiver are collocated in different rooms

Table 2.3: *Parameter γ as function of the building structure*

Introducing this parameter in the above equation, finally we obtain

$$\boldsymbol{PL(dB) = PL(R_0) + 10\gamma \log_{10} \frac{R}{R_0}}.$$

An example of an indoor scenario is shown in the following figure.

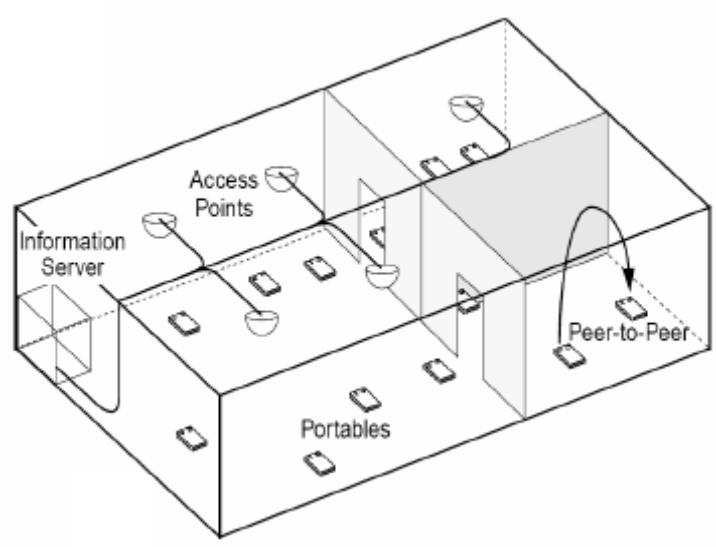


Figure 2.16: *Indoor application scenarios*

We can remember that, for our purposes, a typical indoor application scenario provides to:

- home networking and sensor interconnection
- cable replacement
- system interconnection

As regards the outdoor models, first we must notice the numerousity of them and then that they differ from each other for the approach, accuracy and complexity.

Some examples are:

- Okumura model (1968). It is one of the most used for planning mobile system (specially in Japan) and it is considered one of the better as to accuracy
- Hata model (1980). It introduced analytic explanation of the empiric results of the Okumura model
- Walfish - Bertoni model (1988). It is oriented to estimate the signal strength as three contributes of loss: free space path loss, the reduction determined by the roofs, the reduction caused by the final path from the last roof to the receiver
- Ikegami model. It is a specific model used to estimate the final path of the Walfish Bertoni model
- COST-Walfish-Ikegami model (COST-WI). This model allows estimating the path loss introducing parameters of the environments in which the radiation is propagating

An example of an outdoor optical link may be that represented in the following figure.

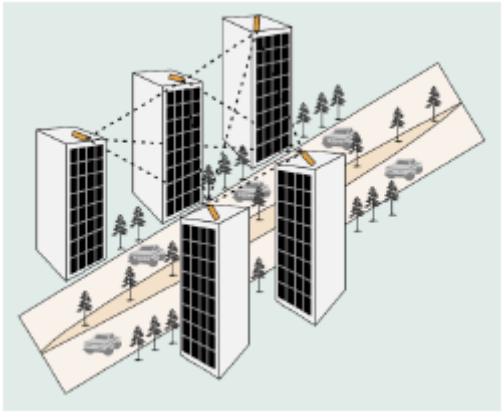


Figure 2.17: *Outdoor application scenarios (MAN UOWC)*

For our scopes, an outdoor application scenario allows:

- last-mile access
- building interconnection

2.6. Noise and interference

As regards the interference, we have already exposed that the main interference sources are caused by effects generated in the fiber because of the presence in the channel both of the interested channel and of multiple propagating channels.

Furthermore, the particular transmission system that we are analyzing possesses both classic noise sources, as the thermal noise, of which we have robust and consolidate theoretic models[1], and specific (of an OCS) noise sources, as natural or artificial illuminations, whose effects in the transmission chain are nowadays under theoretical and experimental study and of which we are going to investigate with some detail the main characteristics.

The main noise sources that we must consider in an OCS are:

- Sunlight illumination. It generates into the receiver, the photodiode, shot noise that can be considered as white gaussian noise
- Artificial lamps irradiance, as incandescent, fluorescent, halogen lamps. It produces shot noise that can be considered as a narrowband interference

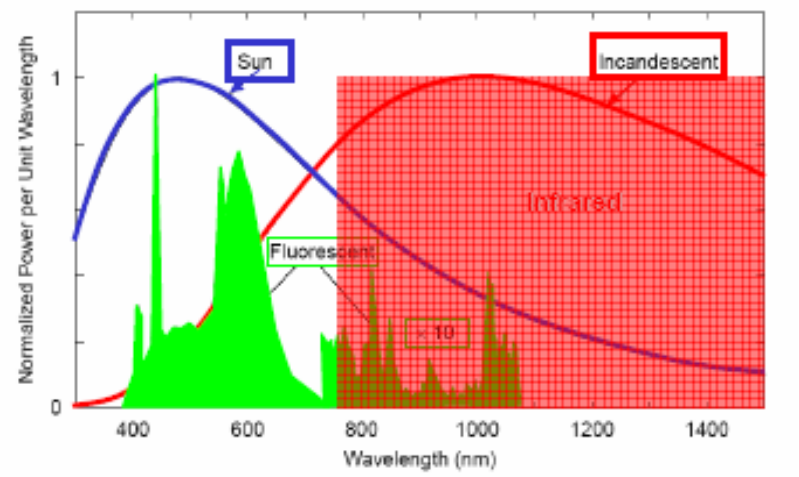


Figure 2.18: *Spectrum of the noise optical sources*

[1] we will show some of them in the section dedicated to the receiver

Even employing optical filtering, the photodiode, that transform the radiation in d.c. photocurrent, can receive an average power that overcomes the signal of interest causing, in last analysis, bit error.

As one can verify using the Wien shift law

$$\lambda_{\max} = \frac{C}{T} \quad \text{where } C = 0.2897 \cdot 10^{-2}\ Km,$$

the solar irradiation comes to the Earth at wavelengths that are determined by the photospheric temperature of the sun, which peak is around 5600 °C. As shown in Figure 2.18, the main contributes are in the wavelength range of 0.2 and 3.4 μm with the maximum around at 0.48 μm which is in the visible green region. Moreover, as shown in the following figure, the spectral irradiance is reduced by the atmospherical absorption.

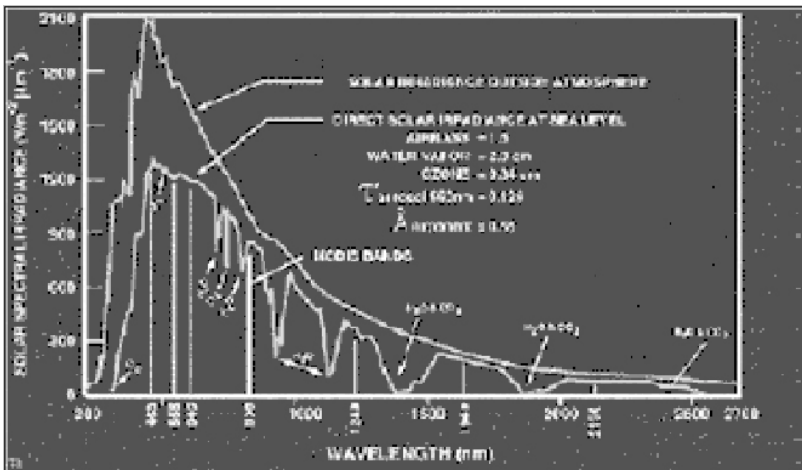


Figure 2.19: *Solar Spectral Irradiance through the atmosphere*

In order to limit the incident radiation of the ambient light noise, we can interpose lenses between the output of the channel and the input of the photodiode. This leads to a smaller d.c. photocurrent and thus to a reduced shot noise as it is shown in the following table.

	Without optical filter	**With optical filter**	**Optical filter reduction**
Direct sun light	5100 μA	1000 μA	5.1
Indirect sun light	740 μA	190 μA	3.9
Incandescent light	84 μA	56 μA	1.5
Fluorescent Light	40 μA	2 μA	20

Table 2.4: *Effect of optical filtering on the d.c. photocurrent*

By mean of experimentations, it has been shown that halogen and incandescent lamps can be modelled as sum of cosinusoids whose harmonics are multiple of 100 Hz. Over 800 Hz all the amplitude components are 60 dB below the fundamental and, therefore, negligible.

The following figure shows the typical electrical spectrum (namely, after transduction) of a 60 W, 50 Hz tungsten-filament incandescent lamp.

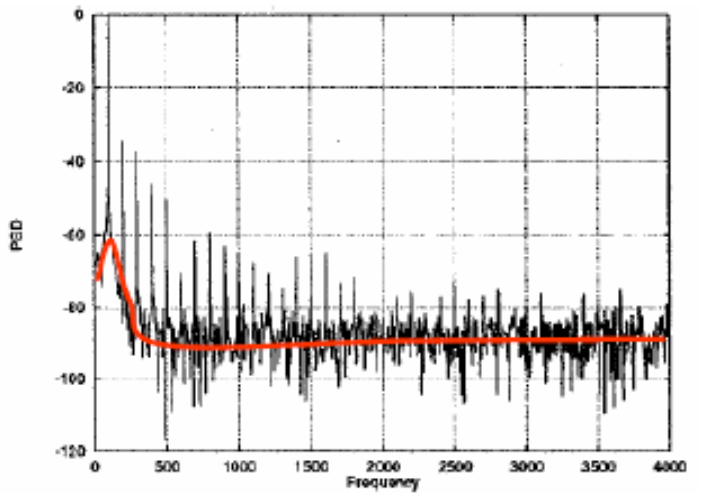


Figure 2.20: *Typical electrical spectrum of a 60 W, 50 Hz tungsten-filament incandescent lamp*

Because of its limited spectral power, it is possible to reduce sufficiently the effect of incandescent lamps on IR signals.

Vice versa, the typical electrical spectrum of a fluorescent lamp holds components at harmonics up to tens of kilohertz.

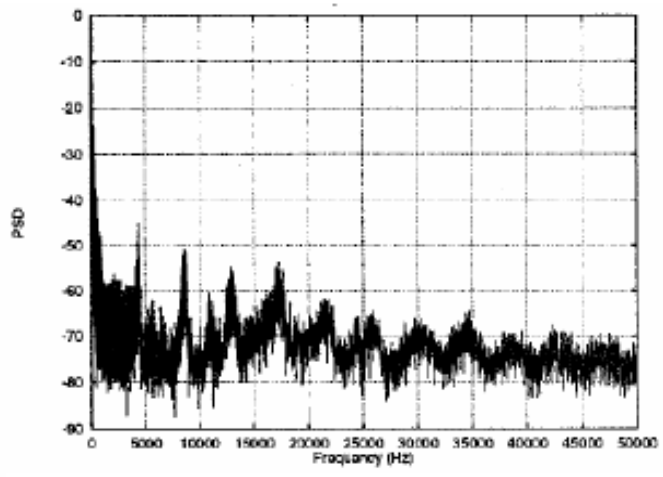


Figure 2.21: *Power spectral density of shot noise of a fluorescent lamp*

Much more problems are generated if electronic ballast is used. In fact, in this case the electrical d.c. interference current generated after transduction holds components with elevated amplitude up to hundreds of kilohertz.

In order to mitigate the interference correlated to fluorescence lamps when, in particular, driven by an electronic ballast, it is possible to use either lenses or to exploit the code gain obtained using, as we will see in the next chapter, *spread spectrum* techniques.

In order to give an idea of what said, it is shown in the following figure (which displays the SNR requirement for several modulation techniques, fixed the BER at10^{-9}) the bit rate at 10 Mb/s, for two electronic ballasts.

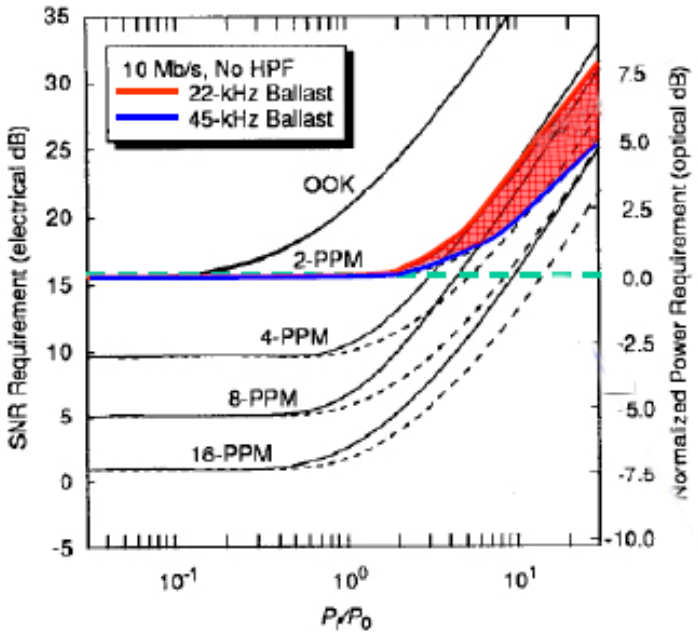


Figure 2.22: *Example of performance obtained with several modulation techniques*

2.7. Lenses and interferometer optical filter

As seen above, the channel introduces noise and interference that we can mitigate first of all using tinted lenses. The use of lenses not only allows reducing sufficiently sun light but also to increase the *FOV* (Field Of View), as the CPCs, implying a large amount of received optical power and, as counterpart, a more intensive multipath dispersion caused by different paths inside the lenses. This effect is schematically shown in the following figure.

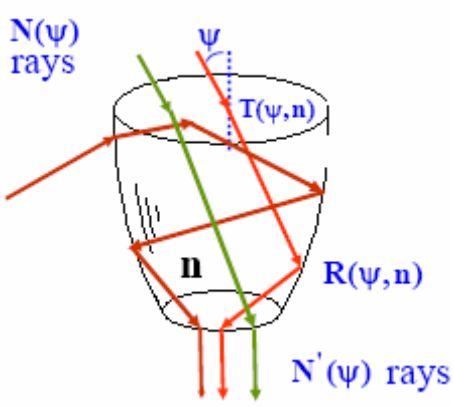


Figure 2.23: *Example of multipath effect inside the lenses*

The optical filters can be or less tunable. The most used are: Fabry Perot interferometer, Mach Zehnder interferometer, Bragg's grids.

The interferometer optical filters, over that to subsequently reduce ambient noise and interference, are used to isolate different channels in a *WDM* (Wavelength Division Multiplexing) scheme access. Some examples are: Mach Zehnder interferometer, waveguide grating, add/drop filter.

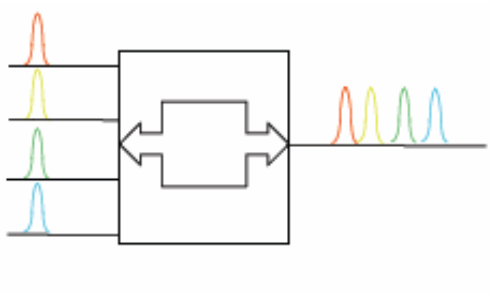


Figure 2.24: *WDM access's scheme*

2.8. The photodiode

Nowadays, there are in commerce two kinds of photodiodes: *PIN* and *APD* photodiodes. In an OCS, both are used for the optical-electrical conversion. In the following of this section we are going to present with some details the physical effects and the main characteristics on which they are based.

When light of appropriate wavelength impress on a junction, because of the absorption of photons, electrons overcame the bandgap energy going from the valence band to the conduction band, therefore leading to generate photocurrent. In particular, the photodiode can work either in zero bias, in which way the device has the same behaviour as that shown above, or in reverse bias, in which it holds an extremely high resistance up to that light falls on the junction causing its reduction and thus leading to a high sensitivity to the light exposure.

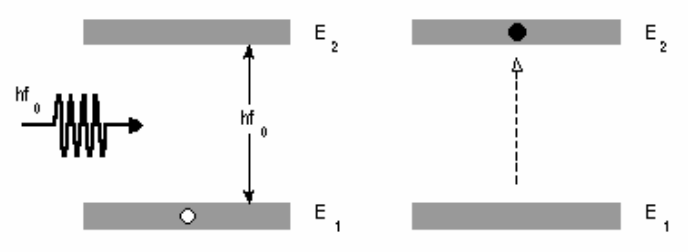


Figure 2.25: *Absorption of a photon*

Working with reverse bias junction presents also a drawback. It consists in the presence of a small current, called *dark current*, even though there is no light falling on the junction. The presence of dark current leads to a reduced value of the total SNR and, since the dark current is temperature dependent, the amount of this leakage goes up increasing the temperature.

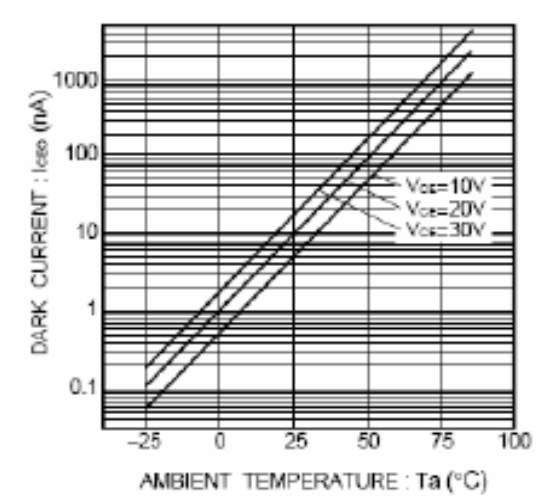


Figure 2.26: *Effect of the temperature increasing on the dark current*

When presents, the dark current is added to the useful current, the photocurrent, which is related to the optical power by the following formula:

$$i(t) = \rho \cdot p_{opt}(t) \quad [A]$$

where the quantity

$$\rho \overset{def}{=} \frac{I_{out}}{P_{in}^{(opt)}} = \frac{\eta q}{h\nu} = \frac{\eta}{1.24}\lambda_{\mu m} \quad \left[\frac{A}{W}\right]$$

is the *responsivity* and η the *quantum efficiency*. The dependence from the temperature of the quantum efficiency makes also the responsivity depending from the temperature and, thus, the photocurrent as well.

In order to obtain a sufficiently high value for the SNR we must limit both the thermal noise and the shot noise.

The first one is generated by the resistance of the device *R* and it is expressed by the well known Johnson formula

$$I_J = \sqrt{\frac{2RkFT_0 \cdot 2B}{R^2}} = \sqrt{\frac{4kFT_0B}{R}} \quad [A].$$

The contemporary presence of the useful current signal and the dark current leads to shot noise current that can be express by the following formula

$$I_S = \sqrt{(I_d + I_p + I_B)q \cdot 2BM^2F_e} \quad [A]$$

where I_d is the dark current, I_p the photocurrent generated, I_B the ambient noise, B the system bandwidth and $F_e = M^a$ a parameter related to the gain of the APD diode ($M = 1$ for the PIN diode).

The total SNR at the receiver is thus

$$SNR_{TOT} = \frac{I_p^2M^2R}{I_S^2R + I_J^2R} = \left((SNR_S)^{-1} + (SNR_J)^{-1}\right)^{-1}.$$

Generally, we have the two following situations:

- in an APD diode the shot noise is dominant and we have:

$$SNR_S \cong \frac{I_p^2M^2}{I_S^2} \cong \frac{I_p}{q2BF_e}(I_d \cong 0)$$

- in a PIN diode the thermal noise is dominant and then we obtain:

$$SNR_J \cong \frac{I_p^2 M^2}{I_J^2} = \frac{\left(\frac{\eta}{1.24}\lambda_{\mu m} P_{in}^{(opt)}\right)^2 RM^2}{4kFT_0 B}.$$

2.9. The electrical processing

The above physical SNR can be improve first of all by means of an electrical filtering and subsequently exploiting modulation techniques as we will show in the next chapter.

2.10. Appendix: the solitons

The possibility to obtain solitons inside optical fibers has been theoretically foreseen in the 1973 by the derivation of the Schrödinger equation for the propagation of the wave packet inside fibers, while the experimental observation of the formation of a soliton has been obtained in the 1980.

The solitons possess a particular property of orthogonality: when two propagating solitons interact among themselves, as the case of two impulse impacting that are propagating in opposite direction, they don't influence reciprocally but maintain the own "individuality" after the interaction, that is, they preserve the same wave shape (property generally not valid in nonlinear propagation). The above property is valid when the solitons have different wavelength and allows controlling the nonlinear problem in the optical transmission.

As counterpart, the above characteristic limits the use of solitons as impulses that represent one and zero bit in a numeric transmission. In fact, solitons at the same frequency, as those relative to two adjacent bit intervals, are subjected to attraction or repulsion strength in dependence of the relative phase of the wave packet. In a numerical transmission using solitons, the impulses must be maintained to a minimum distance, not generating hence appreciable interaction between them. In the opposite case, a variation in the position of the pulses, respect to the transmission clock, is generated. If t_0 is the time duration of the pulse, in order to avoid this problem, we must use at least a time separation $T \geq 20 \cdot t_0$ that, evidently, is a limit to the maximum bit rate of a solitonic system. Despite this, it has been shown that these systems lead to a bandwidth ten times better than the common infrared systems characterized by the use of the second windows or those that employ the third windows with DS fibers.

Finally, one of the most recent discoveries (INFM, 2004) has seen Italian researchers to control the direction of emission of a soliton beam, leading to the technical capability of deviate the direction of propagation of the solitons.

Chapter 3

Modulation and Coding Schemes

Contents

3.1 Introduction 30
3.2 Spread spectrum techniques 31
3.2.1 Direct Sequence Spread Spectrum 31
3.2.2 Time Hopping Spread Spectrum 35
3.3 On Off Keying modulation 37
3.4 Pulse Position Modulation 38

3.1. Introduction

As first marking, the intent of this chapter is not to exhaust completely the modulation and coding schemes in each of them articulate parts, but to provide a sufficiently detailed description of the most important[2] modulation techniques and coding schemes used in optical communication systems.

In order to realize our scope and taking in to account that nowadays the most used schemes in the optical systems are those carrierless, we will show the basic principles about: *Spread Spectrum* (*SS*) techniques (*DSSS*, *THSS*) and *Intensity Modulation* (*IM*) techniques (*OOK* and *PPM*).

In the following figure are shown the modulation and coding schemes mostly used in an optical communication system.

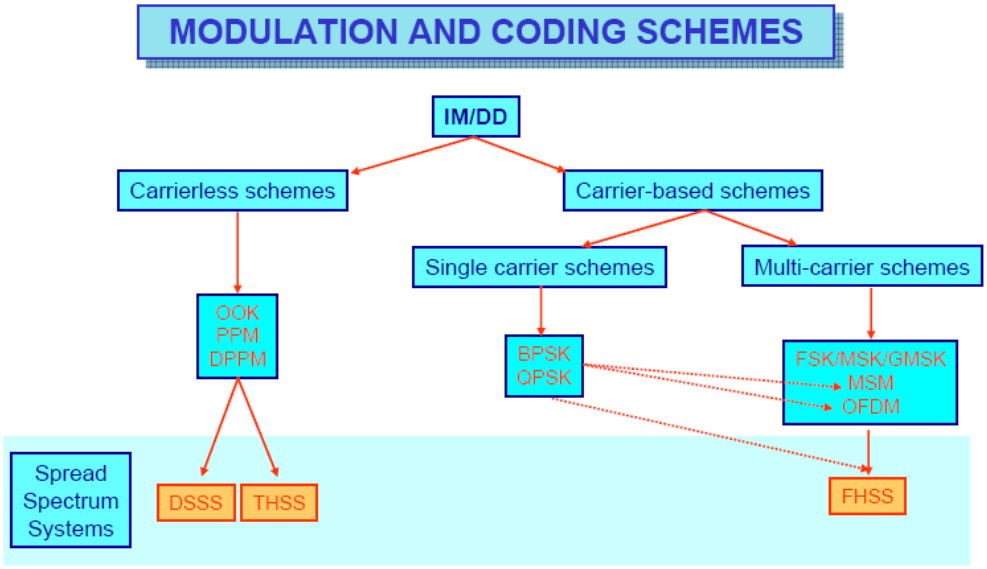


Figure 3.1: *Modulation and coding schemes*

[2] relatively both to the current use and to research for future implementations

3.2. Spread spectrum techniques

As it is shown in Figure 2.1, spread spectrum techniques are divided in carrierless schemes, DSSS (Direct Sequence Spread Spectrum) and THSS (Time Hopping Spread Spectrum), and carrier based schemes, FHSS (Frequency Hopping Spread Spectrum).

In the following two subsections we will give, with some detail, the main and fundamental principles about them.

3.2.1. Direct Sequence Spread Spectrum

The transmission scheme of a DSSS system is sketched in Figure 3.2.

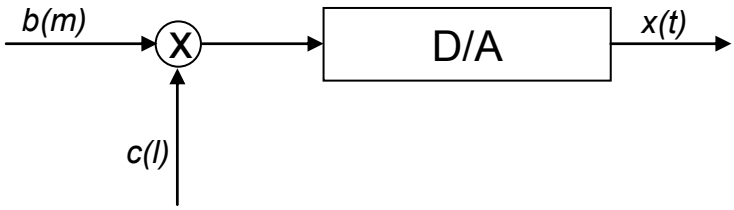


Figure 3.2: *Representation of the baseband transmission scheme of a DSSS system*

Using the representation of Figure 3.2, one has to pay attention to the fact that the data rate before and after the multiplication by the code $c(l)\,(l=0,\ldots,N-1)$ is different. In particular, being $R_B = 1/T_B$ the bit rate, T_B the bit period and multiplying each bit to be transmitted by a code composed of N elements, denoted as *chips* whose duration is $T_C = T_B/N$, after multiplication one gets a chip rate equal to $R_C = 1/T_C = N/T_B = N \cdot R_B$ leading (at least) to a bandwidth of $B_C = 1/2T_C = N \cdot B_B$. This feature justifies the name of the method, as the system spreads the transmit power over a spectrum that is larger by a factor N than the spectrum strictly required to send the original sequence of symbols.

For advantageous reasons that it will be shown in the following, the sequences $c(l)$ used in DSSS systems are typically *pseudo-noise* and, optionally, *orthogonal* (for CDMA) sequences having unitary modulus, that is $c^2(l) = 1$. When both properties are used, the system takes the name of DS-CDMA system.

The pseudo-noise nature of the spreading sequence $c(l)$ provides to a fundamental feature of the spread spectrum system called *Low Probability of Intercept* (*LPI*), namely spread spreading signals are typically hard to detect, to intercept and to demodulate.

The orthogonal nature of the spreading sequence allows using CDMA techniques that are exploited in order to grant to many users the same portion of spectrum at the same time.

With the warning in mind about the rate change, the overall DSSS system can be represented as in Figure 3.3:

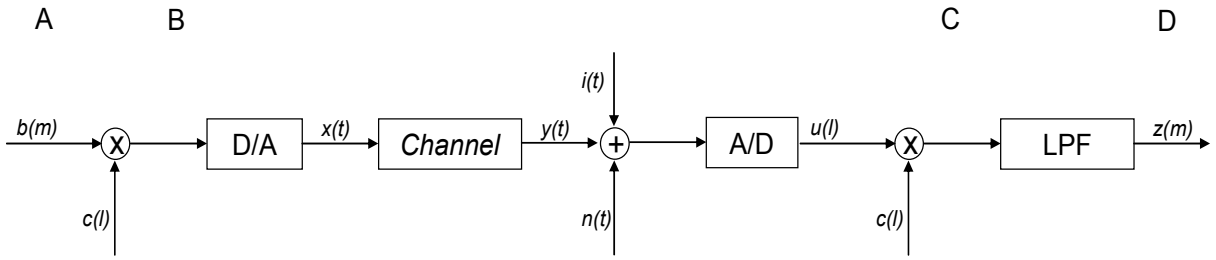


Figure 3.3: *Representation of the baseband DSSS system*

where $i(t)$ represents the interference, whereas $n(t)$ is noise. The final block in the receiving chain is a low-pass filter (LPF), with bandwidth $B_B = 1/2T_B$.

To have a physical insight into the properties of a DSSS system, it is useful to analyze how the spectrum of useful signals and interference change as they pass through the system. To this end, in Figure 3.3 we have labelled four points: *A*, *B*, *C*, and *D*. The power spectral densities in the points *A* and *B*, denoted by $S_A(f)$ and $S_B(f)$, are shown in Figure 3.4.

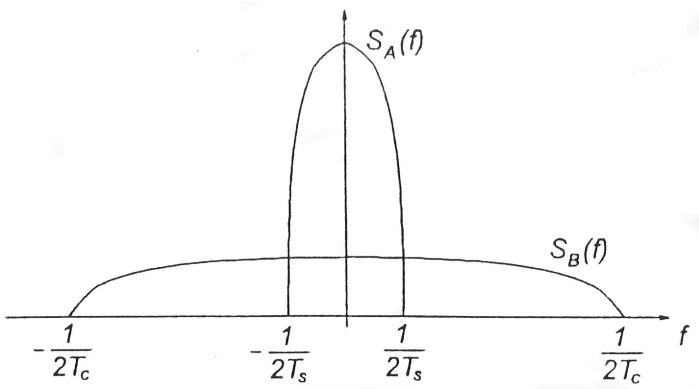


Figure 3.4: *Power spectral densities in the points A and B of the DSSS system*

As it is depicted in Figure 3.5, the increased spectrum can be justified by the augmented rate that, in the point *B*, is N times that of the point *A*.

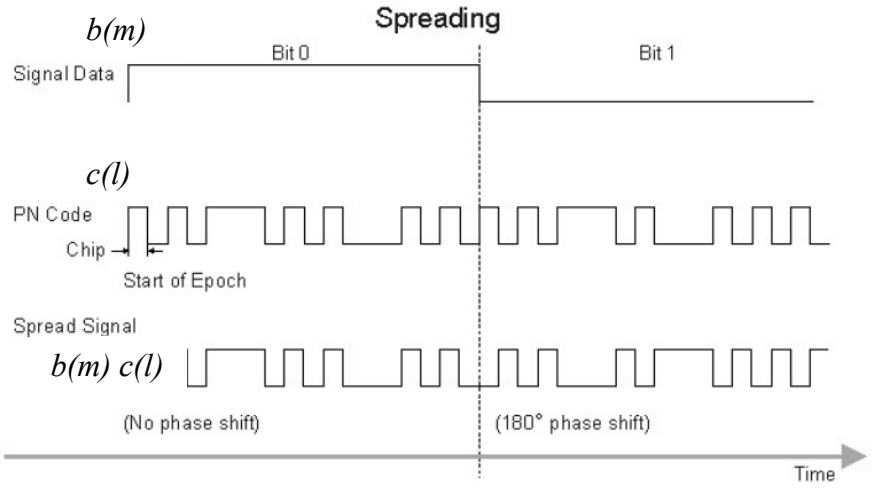


Figure 3.5: *Effect in the time domain of the bit-chips multiplication*

Comparing $S_A(f)$ with $S_B(f)$, we can observe how multiplying by the code leads to spread the original spectrum. Furthermore, since the spreading sequence $c(l)$ has unitary modulus, the multiplication does not alter the signal power. For this reason, the areas subtended by the two power spectral densities in Figure 3.4 are the same.

At the receiver, after multiplication for the spreading code $c(l)$, exploiting the property that $c^2(l)=1$, the power spectral density of useful signal shrinks back into the bandwidth between $-1/2T_B$ and $1/2T_B$. Conversely, all the other signals, namely interference and noise, have a spectral density whose shape is like $S_B(f)$.

At the input of the final low-pass filter, the useful signal has then a spectrum like $S_A(f)$, whereas interference and noise, because of the multiplication by the code sequence, have a spectrum as $S_B(f)$. Hence, at the output of the final low-pass filter, we obtain that all the power of the useful signal passes, whereas only a fraction of the interference goes through the receiver. More specifically, we will show that the overall system holds a gain N in the total SNR, namely in the signal-to-noise-plus-interference ratio SNIR.

Let us consider the m-th transmitted bits:

$$y_m(n) = b(m)c(n) + i(n) + v(n),$$

where $i(n)$ and $v(n)$ are respectively the interference and the noise sequences. Let us denote with P_B^{IN}, P_I^{IN} and P_N^{IN} respectively the power of signal, interference and noise at the input of the DSSS receiver. Let us compute the power of these three contributions at the output of the receiver. Multiplying $y_m(n)$ by $c(n)$ and summing over *n*, we obtain

$$z(m) = \sum_{n=0}^{N-1} c(n) y_m(n) = N\, s(m) + \sum_{n=0}^{N-1} c(n) i(n) + \sum_{n=0}^{N-1} c(n) v(n),$$

in which we have used the property $c^2(n) = 1$. To compute the power of the interference and noise contribution we exploit now the property that the spreading sequence is pseudo-noise. In particular, considering the elements of $c(n)$ as i.i.d. random variables with zero mean and unit variance, the power of the interference and noise at the output of the DSSS receiver are $P_I^{out} = N\, P_I^{in}$ and $P_N^{out} = N\, P_N^{in}$, respectively. As a consequence, the SNIR at the output of the DSSS receiver is

$$SNIR^{out} = \frac{N^2\, P_B^{in}}{N\, P_I^{in} + N\, P_N^{in}} = N \cdot SNIR^{in}.$$

This property explains one of the most fundamental reasons for using DSSS system: the relative immunity against interference, property denominated *Anti Jam* (*AJ*). To this regard, we must introduce a fundamental parameter, the process gain G_p, expressed by the following formula:

$$G_p = \frac{W}{R_b},$$

namely by the ratio of the bandwidth occupation and the data bit rate. The processing gain G_p leads to a better capacity in the interference rejection (directly proportional to G_p), and to an increased system capacity (directly proportional to G_p ,as well).

Considering that spread spectrum techniques are jointed with Code Division Multiple Access (DS-CDMA systems), we can state by the above formula that the main limitation of a DS-CDMA system is the *Multiple User Interference* (*MUI*).

Finally, we remind briefly that there can be different kinds of sequences and, depending of them, changes the particular multiplication used. For example, if the sequences were antipodal, we should use the typical multiplication among real numbers whereas, if they were binary sequences, as in the Optical Orthogonal Codes (*OOC*), the complemented XOR would be proper.

b_1	b_2	$b_1 \overline{XOR} b_2$
1	0	0
0	1	0
0	0	1
1	1	1

Table 3.1: *Truth table for the* $\overline{XOR}$ *multiplication*

3.2.2. Time Hopping Spread Spectrum

As shown above, the DSSS technique, multiplying the useful signal by a pseudo-noise code, enlarges its spectrum. Another way to spread the spectrum of a signal is to divide the bit time T_b in N_S intervals of duration T_S, to subdivide them in N_h subintervals of duration T_c in which to transmit, depending of the code, a chip whose value is the same of the bit coded.

What said, can be shown considering the following scheme:

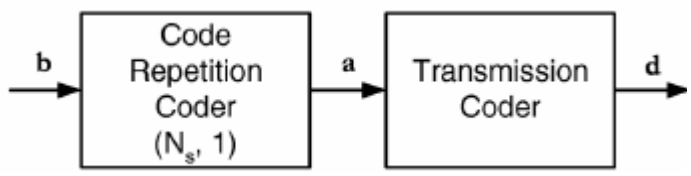


Figure 3.6: *Block diagram of a THSS system*

in which the Code Repetition Coder, introducing redundancy, has an input data rate of $R_b = 1/T_b$, and an output average data rate of $R_S = 1/T_S = N_S/T_b = N_S \cdot R_b$, being T_b the

bit period, T_S the average pulse repetition time and N_s the repetition parameter. The reader can check directly the analogy with what we have shown at the start of the section 3.2.1 for DSSS system.

The transmission coder, using a code ***c*** of length N_p, arranges the elements of the sequence ***a*** inside subintervals whose progressive index is given by

$$d_j = jT_S + c_j T_c ,$$

where T_c is the chip time. Because of this, using a pulse waveform indicated with $p(t)$, we can express analytically a THSS signal as follows

$$s(t) = \sum_{j=-\infty}^{\infty} a_j p(t - d_j) = \sum_{j=-\infty}^{\infty} a_j p(t - jT_S - c_j T_c).$$

The following figure displays the waveform of two bits "1" coded using a THSS modulation technique.

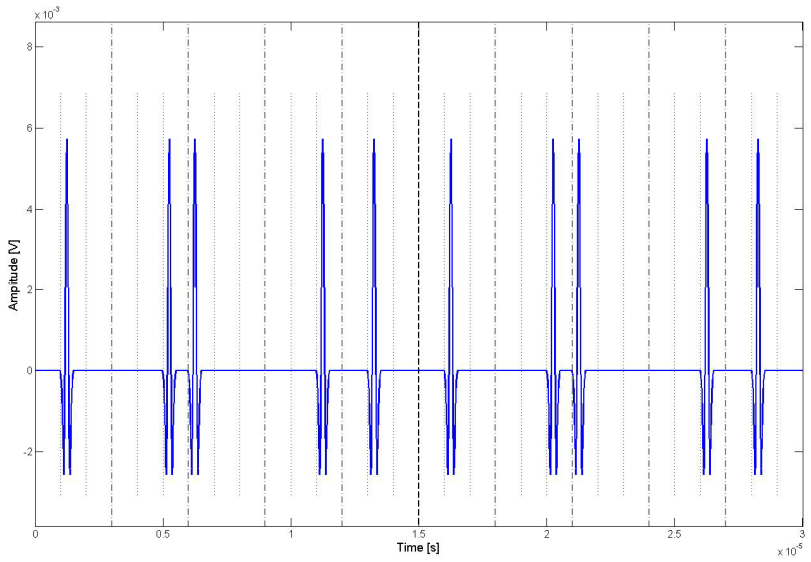


Figure 3.7: *Example of coding of two bits "1" using the THSS technique. The pulse waveform used, is a second order gaussian derivative*

3.3. On Off Keying modulation

As we will see, for practical reasons of implementation, the most used detection technique for infrared links is *Direct Detection* (*DD*). It can be used with several *Intensity Modulation* (*IM*) techniques of which the simplest is the *OOK*.

The OOK modulation allows transmitting using either NRZ pulses or ZR. Transmitting NZR pulses, the rectangular waveform has a time length of T whereas using a ZR it has duration of γT, where the parameter γ is the *duty cycle* defined as:

$$\gamma \overset{def}{=} \frac{T_S}{T_S^{(P)}}$$

where T_S is the effective duration of the waveform and $T_S^{(P)}$ is the symbol period.

For a given available medium power per bit $\overline{P} = P_t$, the OOK modulation foresees either to transmit a rectangular pulse of amplitude $2P_t/\gamma$ if the bit transmitted is "1", or to not transmit if the bit is set to the logical value "0".

The following figure shows two examples of OOK signals using respectively $\gamma = 1$ and $\gamma = 0.5$.

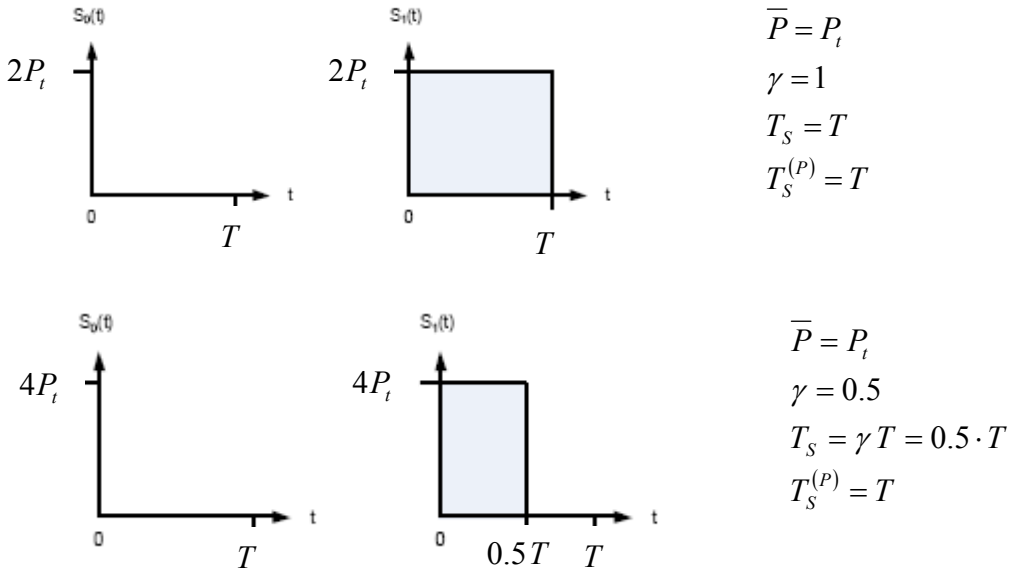


Figure 3.8: *Examples of two signals modulated OOK*

For an OOK modulation, under AWGN hypothesis and supposing that the channel is distortion-free, the ideal maximum-likelihood (ML) receiver can be realized by the following chain: continuous-time filter matched to the pulse waveform, sampler, detector whose threshold is set to $\frac{1}{2}\cdot\left(2P_t/\gamma\right)=\frac{P_t}{\gamma}$.

Eventually, as it is displays in Figure 3.9, one must notice that using shorter pulses implies a larger bandwidth requirement of a factor $1/\gamma$.

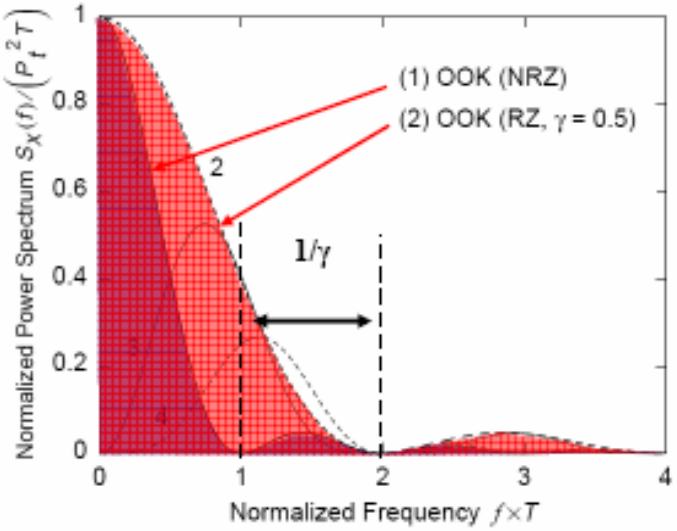


Figure 3.9: *Effect on the bandwidth of reducing the duty cycle* γ

3.4. Pulse Position Modulation

The pulse position modulation consists in transmitting a pulse waveform during a subinterval obtained dividing in several parts the symbol period $T_S^{(P)}$. In particular, if we are interested to transmit $\log_2 L$ bits coded in a symbol (L-PPM) with the limit of an available medium power per bit equal to $\overline{P}=P_t$, we will have to use a constant power of $L\cdot P_t$ assigned to every symbol. This can be shown recalling that the medium power per symbol is given by

$$\overline{P}=\sum_{i=0}^{L-1}p_{S_i}\left(T_{S_i}A\right)/T_{S_i}^{(P)}$$

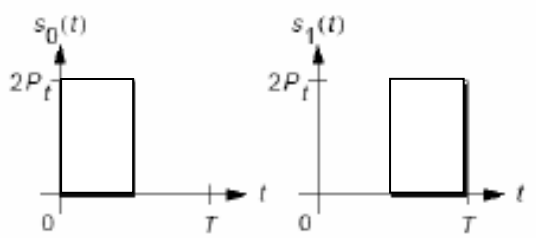
where s_i is the i-th symbol, A the relative power assigned and p_{S_i} the correspondent probability. Obviously, if one is interested to calculate the medium power per bit, the above formula becomes

$$\overline{P_{bit}} = \frac{1}{\log_2 L}\sum_{i=0}^{L-1} p_{S_i}\left(T_{S_i} A\right)/T_{S_i}^{(P)} .$$

For a given symbol, this power $\left(L \cdot P_t\right)$ is allocated to the correspondent pulse transmitted in one of the L subintervals in which the symbol period $T_S^{(P)}$ is divided (remaining all the others with zero power transmitted).

The following figure displays two examples of signals L-PPM modulated.

(a)



(a)	(b)
$\overline{P} = P_t$	$\overline{P} = P_t$
$L = 2$	$L = 4$
$T_S = \frac{1}{L}T = 0.5T$	$T_S = \frac{1}{L}T = 0.25T$
$T_S^{(P)} = T$	$T_S^{(P)} = T$

(b)

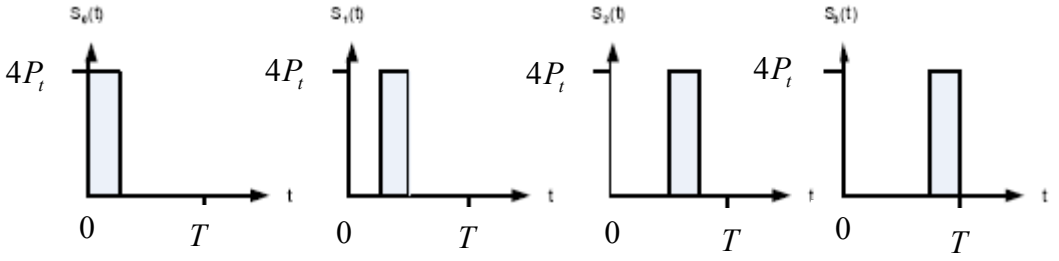


Figure 3.10: *Examples of two signals L-PPM modulated*

One of the main advantage of using L-PPM consists in the fact that, increasing L, the peak power rises linearly with L whereas noise with $L/\log_2 L$. This implies a SNR that goes as $\log_2 L$.

It can be shown that, under AWGN hypothesis and supposing that the receiver is optimum (ML detector), if the SNR is sufficiently elevated, in particular if $E_b/N_0 > 4.43\ dB$, the symbol error probability is constrained by the following upper limit

$$\Pr_e < e^{-\log_2 L\,(E_b/N_0 - 2\log_e 2)/2}.$$

The following figure shows the SER obtained in the cases of $L = 2$, $L = 4$ and $L = 8$.

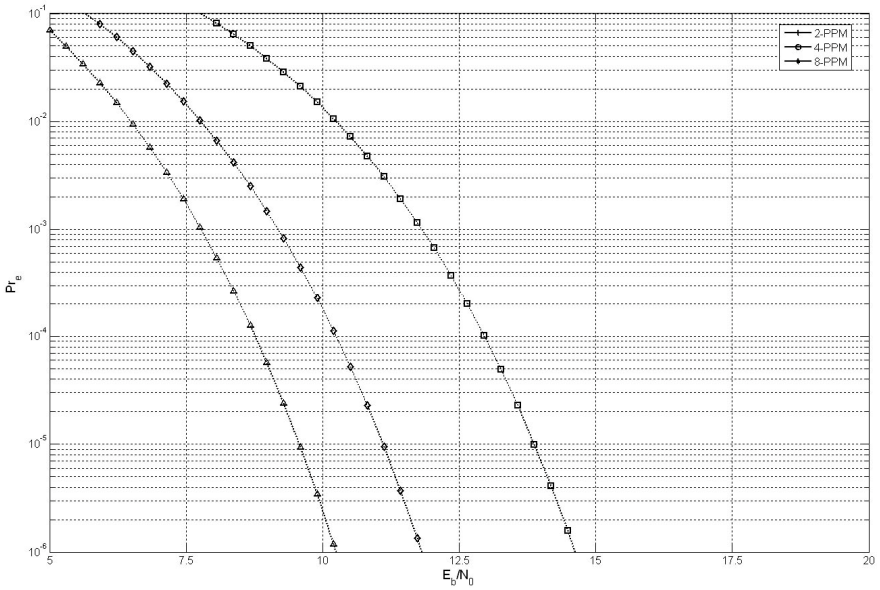


Figure 3.11: *Symbol Error Probability for L-ary PPM*

Vice versa, we can fix the Bit Error Rate and to obtain the power requirements to achieve that BER. The Figure 3.12 gives an example in which we can notice how the performance can be improved by means of coding.

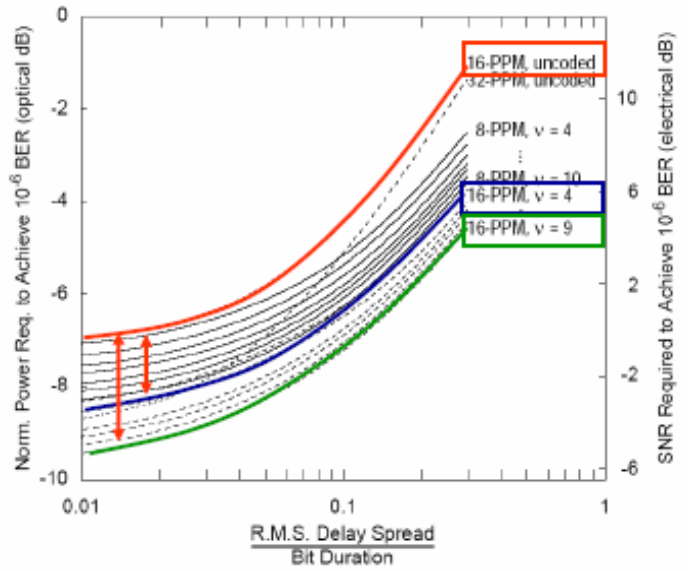


Figure 3.12: *Power requirements obtained varying the number of levels L*

Under AWGN hypothesis, comparing OOK with L-PPM results in the same performance if $L = 2$ whereas, for $L > 2$, L-PPM presents a decreasing energy per bit requirement and, as counterpart, an increasing bandwidth requirement by a factor $L/\log_2 L$.

A useful graphic that represents clearly what said, is the following.

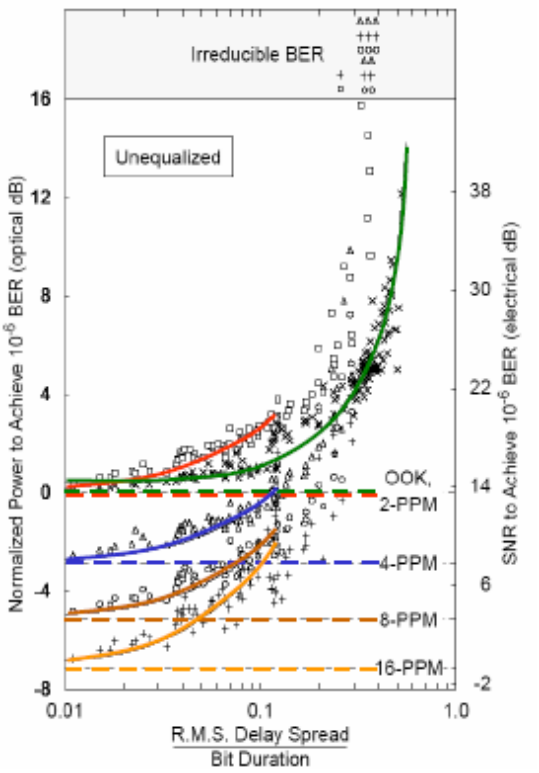


Figure 3.13: *Comparison between OOK and L-PPM performance*

As already said, OOK and 2-PPM get the same power requirement in the obtaining of equal BER. On the other hand, increasing the value of L, leads the L-PPM to a better performance, at least regarding the power requirement. In fact, as evident in the Figure 3.14, if the main limit of the system is the bandwidth, we must do attention about the value of L.

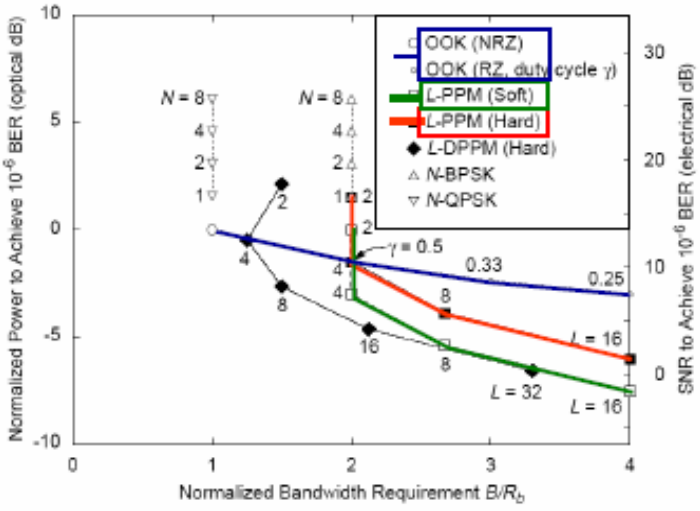


Figure 3.14: *Comparison between OOK and L-PPM bandwidth requirements*

Further, to show the differences in power and bandwidth requirements for OOK and L-PPM modulations, the above figure evidences that, as it is known, there are 1.5 dB of difference in performance between L-PPM decoded with hard and soft decision. Of course, this loss is paid by a relative simplicity to implement hard decoding.

Finally, we should must notice that in an optical system affected by ambient noise, even starting from $L = 2$, the L-PPM systems get better performances than those using OOK modulation.

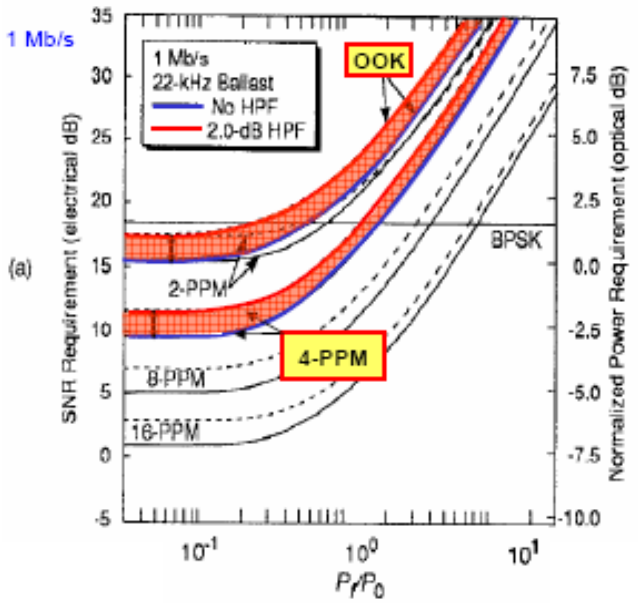


Figure 3.15: *Comparison of OOK and L-PPM system when is present ambient noise*

Chapter 4

The Ultra Wide Band Communication System

Contents

4.1 Introduction 46
4.2 Definition of UWB signal and its generation 48
4.2.1 Generation of TH-PPM-UWB signals and their PSD 49
4.2.2 Generation of TH-PAM-UWB signals and their PSD 53
4.2.3 Generation of MB-UWB signals and their PSD 55
4.3 Pulse shaping in UWB systems 56

4.1. Introduction

UWB is a new and appealing technology that, because of its characteristics, in the last years interested an increasing numbers of manufacturing companies, academic researchers and standardization groups.

As we will see in the following, exploiting advanced transmission techniques, it offers potentialities and features that can lead to a revolution, rather than to an evolution, in the telecommunication systems.

Because of the spectral characteristics of the transmitted signals, UWB is a technology that must coexist with the other pre-existent systems. Therefore, it is fundamental to define and to analyse in detail the interferences caused by UWB systems. In the United States the *Federal Communication Commission* (*FCC*) commissioned measurements campaigns to the *National Telecommunications and Information Administration* (*NTIA*). The report provided the information necessaries to the FCC to issue a first document in which it is allowed the intentional emission of UWB signals and are specified the emission masks.

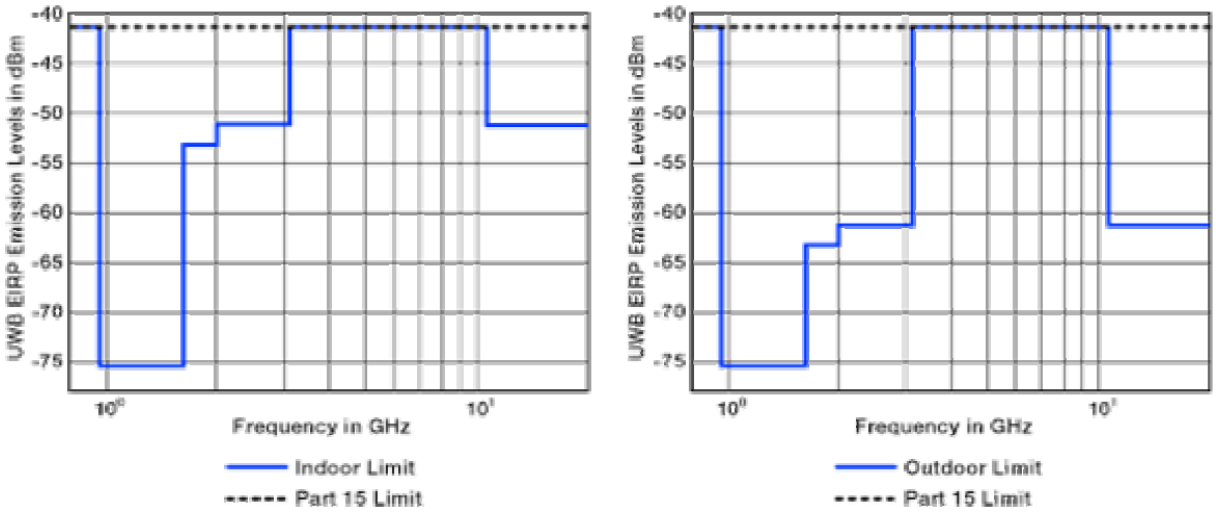


Figure 4.1: *FCC indoor and outdoor emission masks*

Evidently, it is foreseeable that analogous campaigns will be performed in other countries willing to use UWB communication systems.

To date, two main standardization groups, both IEEE, have worked in order to define two kinds of UWB systems, each correspondent to specific fields of applications:

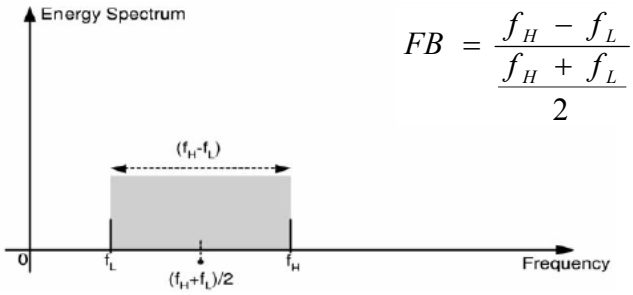
- IEEE 802.15.3 (3a) Task Group is developing an UWB communication system oriented to transmit high bit rate over short range. This technology is suitable for building Wireless Personal Area Networks (WPANs), for transmitting traffic requiring high bit rate such as multimedia traffic (video, audio, images, etc.) both for substituting cable transmissions (USB, IEEE 1394), and to link wearable devices such mobiles, Hi-Fi headphones, stereos, cameras, etc. As to this, meaningful is the choice, taken just in these months by the Bluetooth Special Interest Group, to exploit the advantages of the Ultra Wide Band technology for future versions of the Bluetooth technology.
- IEEE 802.15.4 (4a) Task Group is blooming instead the complementary technology, namely an UWB system oriented to low transmission rates over medium to long ranges. Transmitting typically around 1 Mbps over tens of meters, 802.15.4 is a suitable technology for non real time data applications (e-mailing, messaging, etc.). More over, because of its characteristics, it is particularly appropriate for ad-hoc sensor networks requiring a no stationary infrastructure as it is needful in case of earthquakes, fires, avalanches, intercommunications in fleets of vehicles (taxicabs, trucks, buses) and so on. For these specific applications, the network must provide special features, representing the main challenges for this technology, such as high-precision ranging and localization capabilities, power constraints and low cost.

In the following sections, the main characteristics and challenges of an UWB communication system will be provided, such as: definition of an UWB signal, its generation and spectral characteristics, propagation and interference models, receiver structures, ranging – positioning techniques.

4.2. Definition of UWB signal and its generation

Before to enter in the details of the generation of UWB signals, an exact definition of what is an UWB signal and when a signal can be considered UWB must be provided.

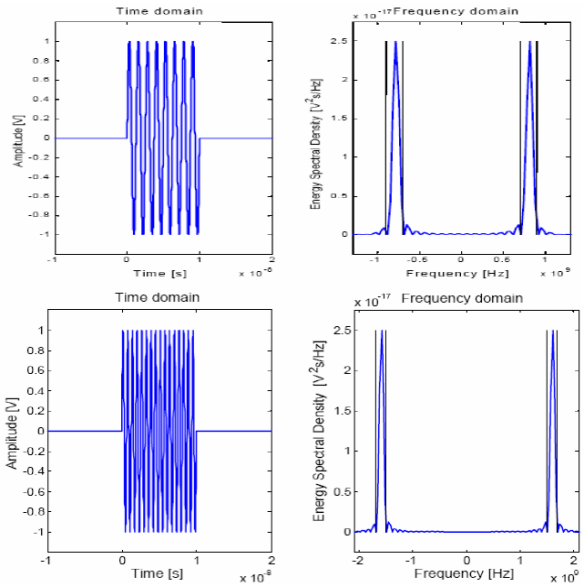
As it is known, the fractional bandwidth is so defined:



$$FB = \frac{f_H - f_L}{\frac{f_H + f_L}{2}}$$

Figure 4.2: *Energy spectrum of a modulated signal*

namely as the ratio between the energy bandwidth of the signal and the central frequency of its spectrum. Whereas following the statement of the radio community, an UWB signal is characterized to have an instantaneous fractional bandwidth larger than about 0.2, in the report of 2002, FCC gives also the possibility to a signal to be UWB when its bandwidth exceeds 500 MHz. This limit leads to a carrier threshold of 2.5 GHz that allows defining the follow definition of UWB signal: given the threshold of $f_t = 2.5\,GHz$, below f_t a signal is UWB if its fractional bandwidth is greater than 0.2, whereas above f_t it is UWB if its bandwidth exceeds 500 MHz.



The signals have carriers respectively of 0.8 GHz and 1.6 GHz (both under f_t). The first one (up), having a fractional bandwidth of 0.25, is an UWB signal, whereas the second (bottom), having a fractional bandwidth of 0.125, is not a UWB signal.

Figure 4.3: *Rectangular modulated signals and their spectrum*

Nowadays, three main transmission techniques have been developed to be used in UWB systems: TH-PPM-UWB, DS-PAM-UWB and MB-UWB. The first two get impulsive transmission techniques, namely discontinuous transmissions. In the MB-UWB case, a continuous time signal is transmitted using an OFDM modulation. In November 2004, for now unofficially, the DS-PAM-UWB case has been chosen over MB-OFDM by the 802.15.3a Task Group for the physic layer of the correspondent standard, that is to say 802.15.3a. It is foreseeable that the TH-PPM-UWB transmission techniques will be chosen for the 802.15.4a physic layer. In the following subsections we will show the three above quoted cases, remarking our attention on the TH-PPM-UWB and DS-PAM-UWB cases.

4.2.1. Generation of TH-PPM-UWB signals and their PSD

In the TH-PPM-UWB case, a PPM modulation is applied to the signal to transmit. A TH code is used, leading to a medium access by means of the time domain. The following figure shows the transmission block scheme of a TH-PPM-UWB system.

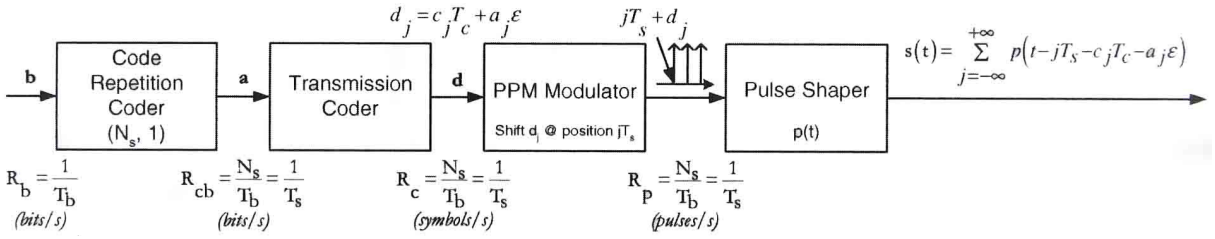


Figure 4.4: *Block scheme for a TH-PPM-UWB transmission system*

The Code Repetition Coder gets the binary sequence **b,** whose bit rate is $R_b = 1/T_b$, as input and introduces in it redundancy. The coder has a coding rate of N_S , leading to an output bit rate of $N_S/T_b = N_S \cdot R_b$.

The transmission coder uses the TH code **c**, whose periodicity is N_p , for arranging the elements of the sequence **a** in the time axis. More over, a time shift of ε is applied if the bit currently coded is "1". Overall, the *j*-th element of the sequence **a** is shifted in time of the quantity $d_j = c_j T_c + a_j \varepsilon$, where T_c is the chip rate and ε the *PPM shift*. Since this shift must be within a time interval of length T_s , called average pulse repetition period, the condition $c_j T_c + \varepsilon < T_s$ is required.

The PPM modulator allows both arranging the *j*-th chip in the temporal instant $jT_s + d_j$, and generating dirac impulses in correspondence of each element.

The final block, the pulse shaper, allows shaping the spectrum of the UWB emitted signal. It has an impulse response $p(t)$, whose analysis will be given in the following of this chapter. Because of the discontinuous nature of the generated signal, it must produce a sequence of strictly not overlapping pulses.

For what said, it is clear that the analytical form of a TH-PPM-UWB signal is given by

$$s_{PPM}(t) = \sum_{j=-\infty}^{\infty} p\left(t - jT_s - d_j\right) = \sum_{j=-\infty}^{\infty} p\left(t - jT_s - c_j T_c - a_j \varepsilon\right).$$

The following figure displays a TH-PPM-UWB signal in the case of two bits transmitted, average transmitted power of -30 dBm, average pulse repetition period $T_s = 3ns$, number of pulses per bit $N_s = 5$, chip time $T_c = 1ns$, code **c** = [2, 0, 1, 2, 2], with cardinality of $N_h = 3$ and periodicity $N_p = 5$, PPM shift $\varepsilon = 0.5ns$ and using a pulse shaper whose impulse response $p(t)$ is a second derivative gaussian pulse.

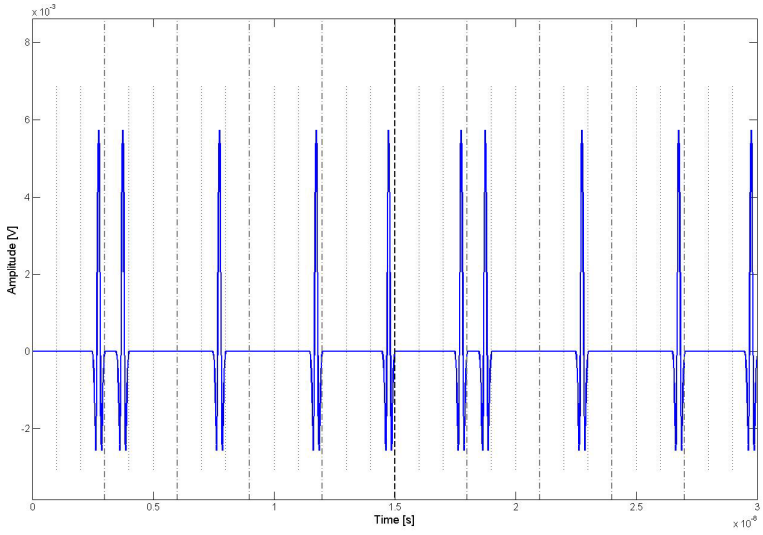


Figure 4.5: *Example of a 2PPM-TH-UWB emitted signal*

The spectrum of a such signal can be found under the hypothesis that the periodicity N_p of the code **c** is equal to the number of pulses per bit N_s leading the signal to be periodic with period $T_b = N_s T_s = N_p T_s$.

In such signal, a temporal structure is repeated every T_b sec; its name is multi-pulse and its analytical expression is given by

$$v(t) = \sum_{j=0}^{N_s-1} p\left(t - jT_s - c_j T_c\right).$$

Depending on the bit transmitted, the PPM shift is applied just to this temporal frame. Therefore, we can express the above formula of the modulated signal as

$$s_{PPM}(t) = \sum_{j=-\infty}^{\infty} v\left(t - jT_b - b_j \varepsilon\right).$$

In the Figure 4.6 is presented an example in which are sequentially transmitted first a "0" bit, after a "1" bit in case of $N_s = N_p = 3$.

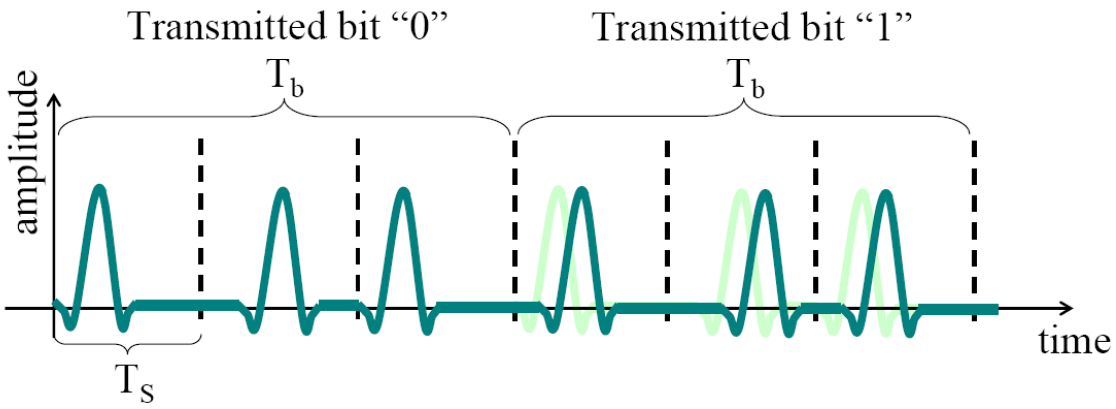


Figure 4.6: *Graphical representation of a 2PPM-TH-UWB signal with* $N_s = N_p = 3$

It can be shown that, under the above hypothesis, the PSD of such signal is given by

$$P_{S_{PPM}} = \frac{|P_v(f)|}{T_b}\left[1-|W(f)|^2+\frac{|W(f)|^2}{T_b}\sum_{n=-\infty}^{\infty}\delta\left(f-\frac{n}{T_b}\right)\right],$$

where $W(f)$ is the Fourier transform of the probability density function of the random bits b_j and $P_v(f)$ the Fourier transform of $v(t)$.

Over that a continuous part given by $W(f)$ and $P_v(f)$, the previous spectrum presents also a discrete part, whose spectral lines have a periodicity of $1/T_b = 1/N_s T_s = 1/N_p T_s$. Obviously, increasing N_s (namely N_p) the periodicity of signal tenses to disappear, leading to an approach of the lines. This effect is displayed in the following figure.

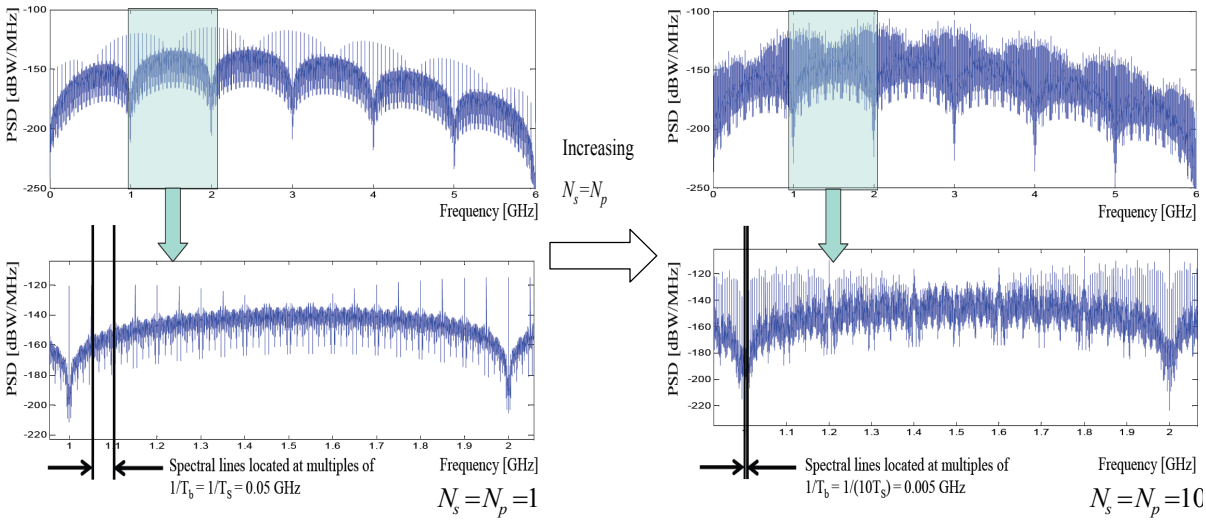


Figure 4.7: *Effect of increasing* $N_s = N_p$ on a 2PPM-TH-UWB signal

However, no longer supposing the equality between N_s and N_p, it is possible to reduce the discrete part of the spectrum increasing the value of N_p, maintaining constant the value of N_s. This technique goes under the name of *PSD whitening*.

4.2.2. Generation of TH-PAM-UWB signals and their PSD

In the DS-PAM-UWB case, a PAM modulation is applied to the signal to modulate and, this time, a DS code is used. The following figure shows the transmission block scheme of a DS-PAM-UWB system.

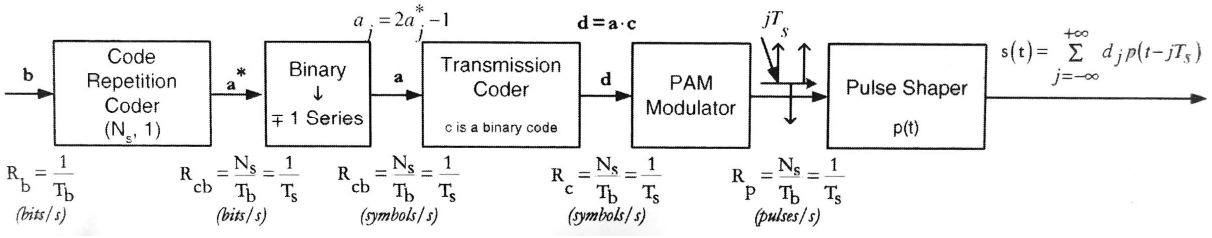


Figure 4.8: *Block scheme for a DS-PAM-UWB transmission system*

Once time again the Code Repetition Coder has in input the binary sequence **b,** whose bit rate is $R_b = 1/T_b$, and introduces redundancy in it. The coder has still a coding rate of N_S , and an output bit rate of $N_S/T_b = N_S \cdot R_b$.

Before to enter into the transmission coder, the binary sequence **a** is first of all changed in a binary antipodal sequence, associating to a "1" bit a value equal to "+1" and to a "0" bit a "-1". The new sequence so generated is indicated with $\mathbf{a}^*$.

The transmission coder uses a DS code **c**, whose periodicity N_p is typically a multiple of N_s, for generating a new sequence **d** obtained multiplying directly each element of **a** for the correspondent in **c**, namely $d_j = a_j c_j$. Evidently, the output bit rate is the same of that in input, that is to say $N_s/T_b = 1/T_s$.

With period T_s , the PAM modulator inserts into the time axis a sequence of dirac pulses whose amplitude is that of the elements of the coded binary antipodal sequence **d**. Once again, this system block does not change the rate.

Likewise to the TH-PPM-UWB case, the pulse shaper shapes the spectrum of the signal emitted using an impulse response $p(t)$. Given the discontinuous nature of the signal emitted, the sequence of pulses must be again strictly not overlapped.

Finally, the emitted DS-PAM-UWB signal can be analytically expressed as

$$s_{PAM}(t) = \sum_{j=-\infty}^{\infty} d_j \, p(t - jT_s)$$

whose PSD, in the case of 2PAM modulation, is given by

$$P_{S_{PAM}} = \frac{1}{T_s}|P(f)|^2 P_c(f)$$

where $P_c(f)$ is the DFT of the autocorrelation of the sequence $\{d_j\}$.

A graphical example of a signal 2PAM-DS-UWB is given in the Figure 4.9. The transmission parameters are: three bits transmitted, average transmitted power of -30 dBm, average pulse repetition period $T_s = 1\,ns$, number of pulses per bit $N_s = 10$, code **c** = [1, 1, 1, 1, 1, -1, 1, 1, 1, -1], with periodicity $N_p = 10$ and using a pulse shaper whose impulse response $p(t)$ is a second derivative gaussian pulse.

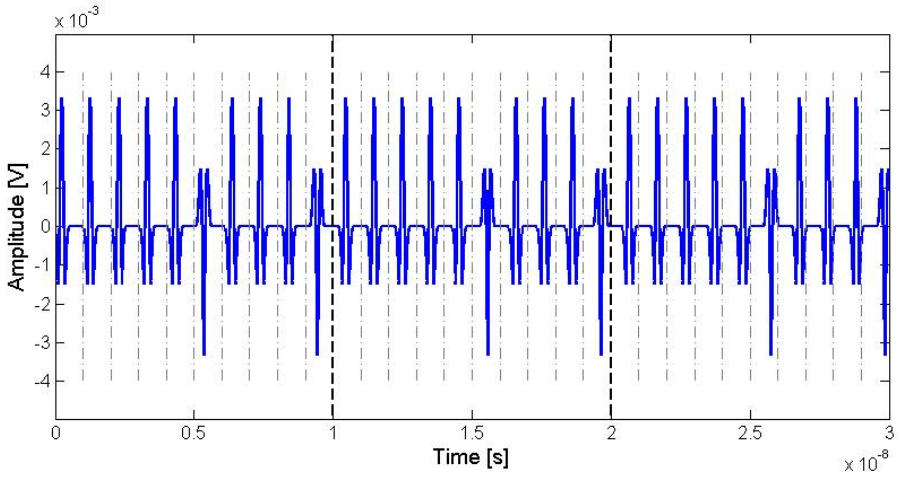


Figure 4.9: *Example of a 2PAM-DS-UWB emitted signal*

The spectrum of a 2PAM-DS-UWB is composed both by a continuous and by a discrete part. Greater is the independence of the elements of **d** (increasing N_p and maintaining constant N_s), more similar to $P(f)$ is the spectrum of the emitted 2PAM-DS-UWB signal.

The following figure displays what said above.

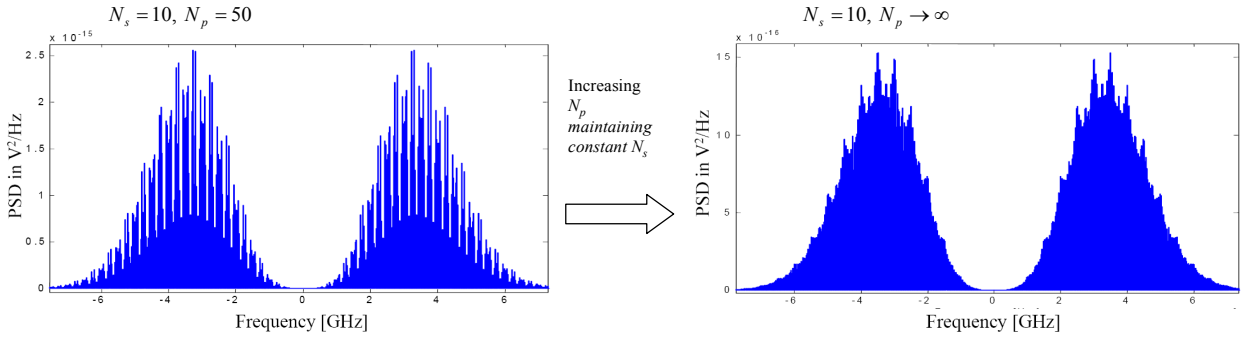


Figure 4.10: *Effect of increasing N_p, maintaining constant N_s, on a 2PAM-DS-UWB signal*

4.2.3. Generation of MB-UWB signals and their PSD

Following what stabilized by the FCC, UWB systems can exploit for their transmissions 7.5 GHz of bandwidth, namely the frequency range within 3.1 GHz and 10.6 GHz. Remembering the definition of UWB signal given previously, in that range a signal is UWB if its bandwidth exceeds 500 MHz.

The MB-OFDM case is based on the OFDM modulation. It has been proposed by the IEEE 802.15.TG3; it foresees 128 sub-carriers, equally spaced by 4.1254 MHz and located at the carrier f_c=3.432 GHz.

The complex envelope of an OFDM symbol is equal to

$$\underline{x}(t) = rect\left(\frac{t}{T}\right)\sum_{m=0}^{N-1} c_m e^{-j2\pi f_m t},$$

to which correspond the following PSD

$$P(f) = \sum_{m=0}^{N-1}\sigma_{c_m}^2 \sum_{m=0}^{N-1} P_{f_m}(f),$$

where $P_{f_m}(f) = \sin c\left(\frac{\pi(f - f_m)}{\Delta f}\right)$ is the spectrum centred on the m-th sub-carrier.

A graphical representation is given in the next figure.

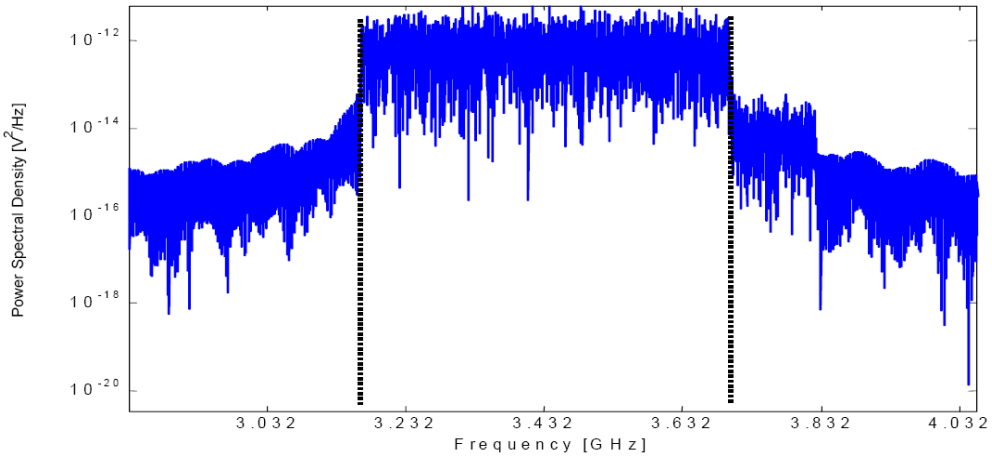


Figure 4.11: *PSD of an MB-OFDM-UWB signal*

4.3. Pulse shaping in UWB systems

As told previously, in the final report of the 2002 the FCC stabilized the emission masks to must respect, at least in the United States (Figure 4.1). In particular, for a system designer this implies a quite heavy constrain in the emitted power.

In order to exploit as much as possible the emission mask, several studies have been taken regarding the system that in an UWB transmission chain mostly conditions the shape of the spectrum: the pulse shaper and its impulse response.

The gaussian pulse, because of its easiness to be generated, is the most used waveform. However, despite of this advantage, alone it does not optimize the use of the available transmission power. It is possible to do much better varying its generation parameters such as the time duration of the pulse, its order of differentiation or, even, a combination of them.

For fixed differentiation order, the duration of a gaussian pulse is related to a fundamental parameter called *shape factor*. In fact, for any differentiation order of the gaussian pulse, one can write:

$$p_d(t) = f(t,\alpha)e^{-\frac{2\pi^2}{\alpha^2}}$$

where $f(t,\alpha)$ is a rational function of t and α. This means that the shorter is the pulse duration (small values of α), the larger is the bandwidth occupation of the transmitted signal.

This effect is shown in the next figure.

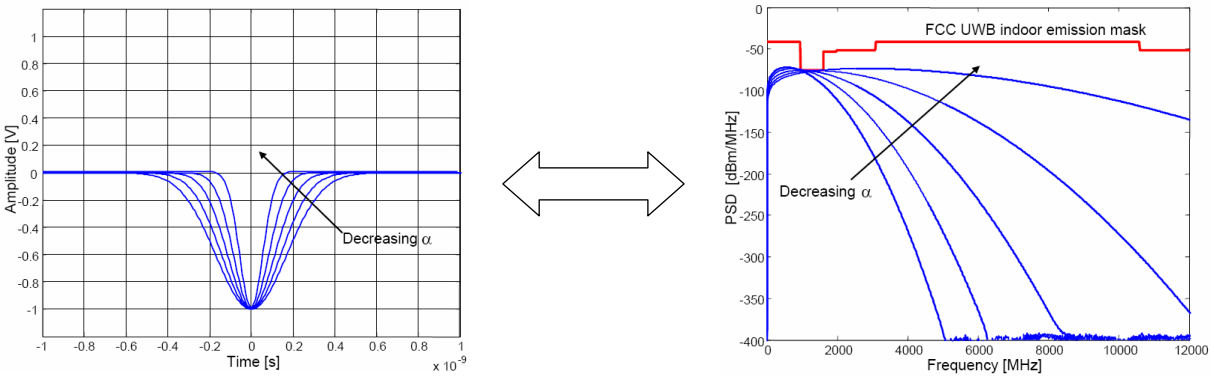


Figure 4.12: *Effect of changing the shaper factor of the gaussian pulse*

On the contrary, if one maintains fixed the duration of the pulse, namely its shape factor, it is possible to obtain higher peak frequency of the spectrum increasing the differentiation order. The peak frequency is in fact related to the differentiation order according with the formula

$$f_{k,\alpha}^{peak} = \frac{\sqrt{k}}{\alpha\sqrt{\pi}}.$$

The following figure provides a graphical representation of this effect for the first fifteen derivatives of a basic gaussian pulse.

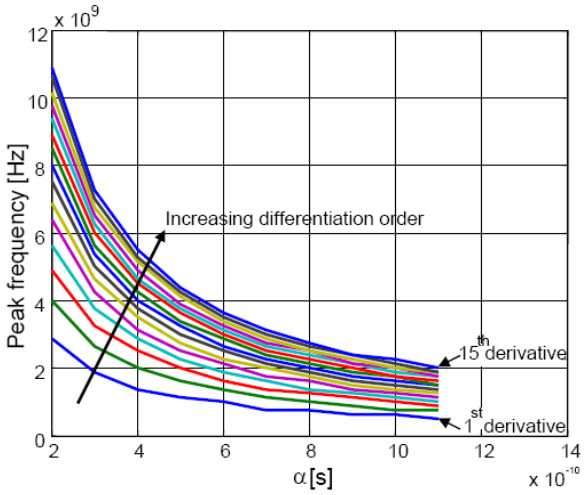


Figure 4.13: *Effect on the peak frequency of modifying the differentiation order*

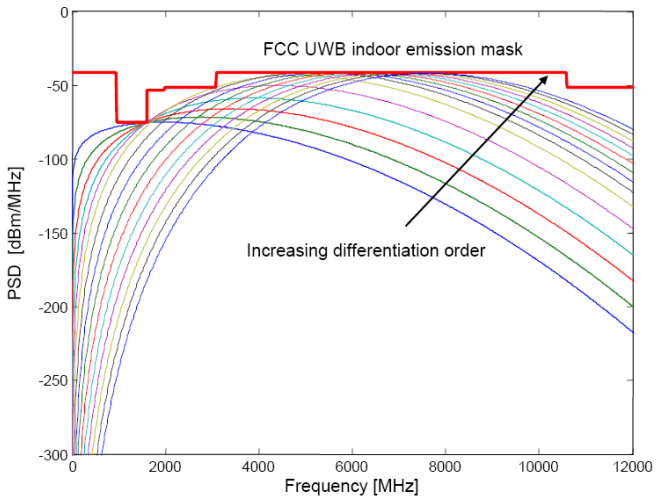


Figure 4.14: *Effect on the PSD of modifying the differentiation order*

Given the previous considerations, an immediate generalization can be effectuated considering the advantage coming from changing the shaper factor joined with that deriving by a variation of the differentiation order. Analytically this means to use a linear combination whose generic term is a gaussian pulses with appropriate shape factor and differentiation order. Therefore, the expression of this new impulse response is given by

$$p_c(t) = \sum_{i=1}^{N} c_i p_i(t)$$

where the coefficients c_i are chosen with the condition of respecting the FCC mask. Examples of algorithms used for obtaining those coefficients are the *Random Selection* algorithm and the *Least Mean Square Minimization* algorithm.

Chapter 5

Wireless Access Control Protocols

Contents

5.1	**Introduction**	**60**
5.2	**Random access protocols**	**62**
5.2.1	Aloha	63
5.2.2	CSMA	66
5.2.3	In-Band CA: MACA	69
5.2.4	Out-Of-Band CA: BTMA	70
5.3	**Scheduled protocols and hybrid protocols**	**73**
5.4	**Examples of standardized MAC protocols**	**74**
5.4.1	MAC protocol for 802.11 standard: DFWMAC	74
5.4.2	MAC protocol for IrDA standard: IrLAP	79
5.4.3	UWB systems: MAC protocols for the standards 802.15.3 and 802.15.4	83
5.4.4	The $(UWB)^2$ MAC protocol	85

5.1. Introduction

In a layered architecture for data network respecting the ISO/OSI stack model, the MAC layer is situated just upper the *Physical layer* (*PHY*) and constitutes the bottom part of the *Data Link Control* (*DLC*) layer whose upper sublayer is the *Logical Link Control* (*LLC*).

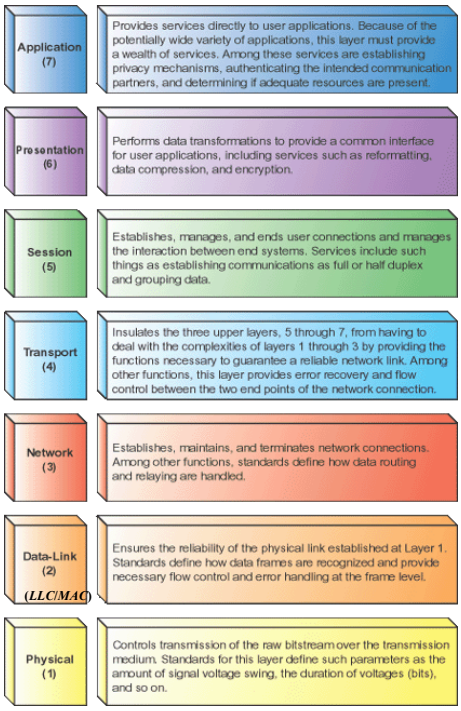


Figure 5.1: *ISO/OSI stack model for a data network*

Although the theoretical definition of each level of the OSI stack does not require to specify the other levels often, for getting an efficient MAC, it is necessary to know with some detail the characteristics of the physical layer. In fact, the MAC layer holds the task of a fair resource sharing and a suitable definition of resource is related to how the physical layer manages transmission and multiple access techniques. Without entering in technical details, we mention as main multiple access techniques TDMA, FDMA and CDMA.

In this *cross-layer design* conception, the service that the MAC offers to the DLC layer consists to prevent and/or to resolve contentions that can be produced in the multiple access to the physical medium.

It has been shown that a set of parameters that can be used for characterizing a MAC layer independently by the characteristics of the physical layer are:

- Throughput, that is the percentage of the channel capacity used during data transmission
- Delay, namely the average time spent by a packet in a MAC queue
- Degree of fairness, that measures the equality of obtaining the medium access by the nodes of the network

A classification of wireless MAC protocols can be provided relatively to the modality of transmitting the data traffic. In fact, whereas random access protocols bring with themselves the possibility of data packet collisions (just because, for how they are defined, the random access phase interests data packets), scheduled protocols foresee that, after a random access phase in which are transferred control packets that can be subjected to collisions, there is a phase whose characteristic is to assure immunity from collisions on data packets.

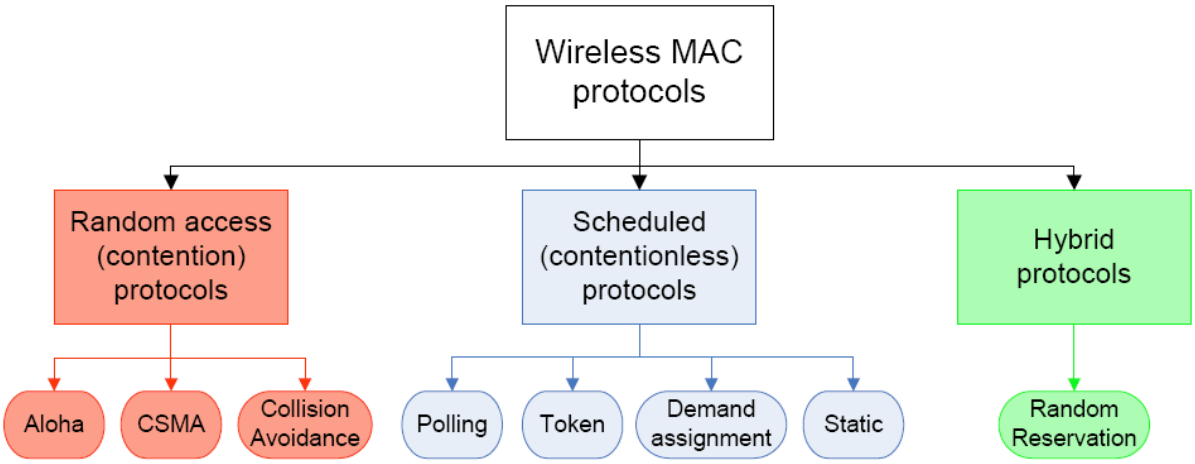


Figure 5.2: *Wireless MAC protocols*

In the following sections we will show in detail the characteristics of wireless protocols, concentrating our attention on the random access protocols.

5.2. Random access protocols

As said in the previous section, random access protocols does not foresee a phase in which is guaranteed resource reservation. For this reason data packet collisions can occur but, as counterpart, these families of protocols are characterized by *simplicity*, in the sense that:

- For transmitting, the terminals do not require specific and detailed information about other terminals (as it is necessary, for example, in a scheduled protocol as *token protocol*)
- The fact that a terminal does not spend (or spends little) time to transmit a data packet assures low delays, in the sense of what said at the end of the previous section

The main drawback derives really from the relative, above quoted, rapidity of transmitting. In fact, when the offered traffic increases, the probability of collision arises as well, leading both to reduce the throughput, and to enlarge the delay, since a terminal must spend time in the retransmission protocol (*ARQ*). Therefore, in networks with high traffic load, *Collision Avoidance* (*CA*) mechanisms are often adopted in order to mitigate this loss of performance.

Collision Avoidance techniques can be classified in the two big families of *In-Band Collision Avoidance* and *Out-of-Band Collision Avoidance*.

Examples belonging to the first one family are *MACA* (*Medium Access with Collision Avoidance*) and *802.11 DFWMAC* (*Distributed Foundation Wireless MAC*) protocols in which CA mechanisms (exchange of control packets between transmitter and receiver) are used in the data channel.

On the contrary, Out-of-Band Collision Avoidance mechanisms require a dedicated control channel in which to transmit control information which, in these protocols, instead to be organized into control packets, often is given as sinusoidal tones. Examples of Out-of-Band CA mechanisms are *BTMA* (*Busy Tone Multiple Access*) and *DBTMA* (*Dual Busy Tone Multiple Access*). We will enter in detail in them characteristics.

5.2.1. Aloha

The Aloha protocol was introduced in 1970 at the University of Hawaii and, as it is known, is one of the simplest random access protocols never developed. A terminal willing send a packet transmits immediately, without taking into account if its transmission can potentially generate a collision. In order to allow to the receiver getting information about the correctness of its reception, a checksum is added at the end of a packet. In case of collision, the checksum will give negative result, the receiver will discard the corrupted received packet and then it will start the retransmission protocol (*ARQ*).

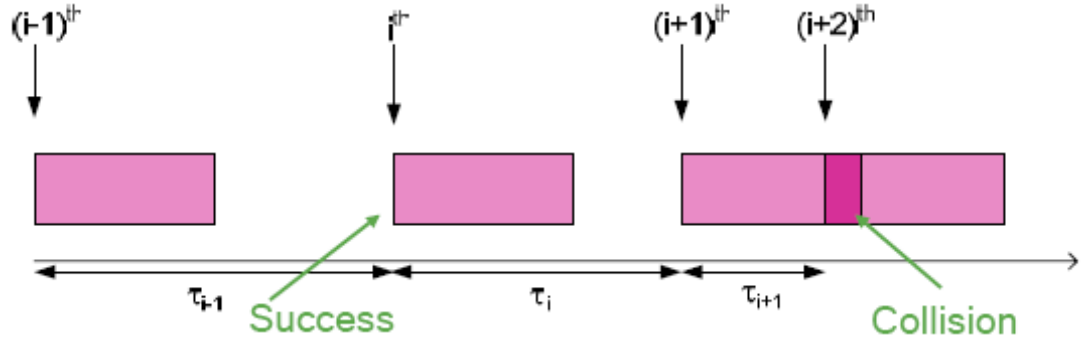
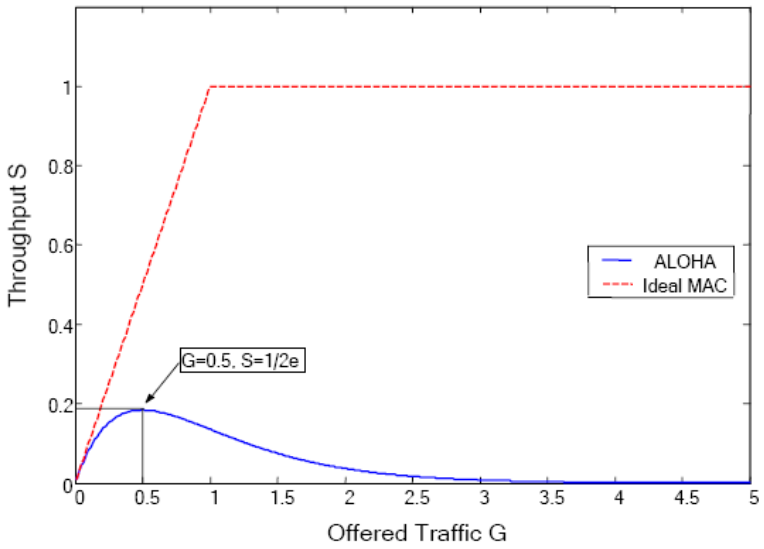


Figure 5.3: *Examples of possible scenarios for transmitting of packets*

By mean the above overview, one can realize that this protocol works well in network characterized by low traffic load but, conversely, its performance is going to reduce when traffic load will increase. The analytic results (found under opportune hypothesis) about the aloha throughout emphasize what said.



$$S = G(n)e^{-2G(n)}$$

where $G(n)$ is the cumulative arrival rate and n the number of terminals waiting for a transmission.

Figure 5.4: *Unslotted Aloha and ideal MAC throughputs*

A way for obtaining a better throughput is to divide the time axis into time slots leading to the so called *Slotted Aloha* in which packets are transmitted only at the start of each time slot.

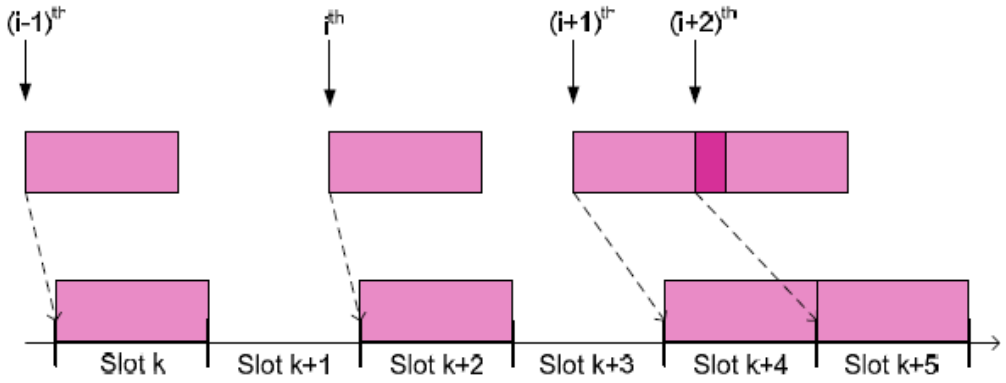


Figure 5.5: *Example of scenario in the Aloha slotted protocol*

It can be shown that in this case the Aloha throughput can be expressed by the following formula

$$S = G(n)e^{-G(n)},$$

that leads to a double maximum throughput and whose graphical representation is provided in the Figure 5.6.

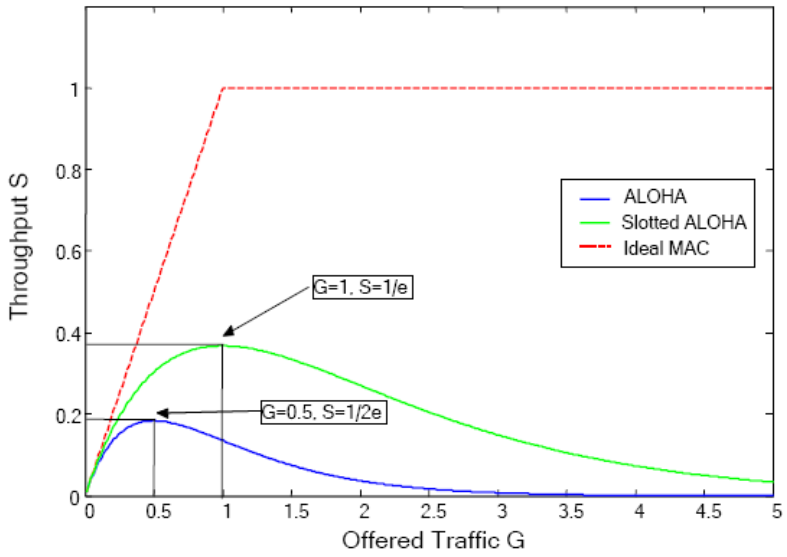


Figure 5.6: *Slotted Aloha, unslotted Aloha and ideal MAC throughputs*

A fundamental remark that one must do regards the potentiality of increasing the throughput in particular scenarios. In fact, the previous formula about the Aloha throughput has been found under the simplistic hypothesis that when a collision occurred the corrupted packet has to be discarded. Actually, even though corrupted, a packet can be decoded correctly if the useful power is sufficiently greater than the interference, namely if the *Signal to Interference Ratio* (*SIR*) is large enough. This effect, called *Capture Effect*, leads to increase the actual throughput, as displayed is the following figure.

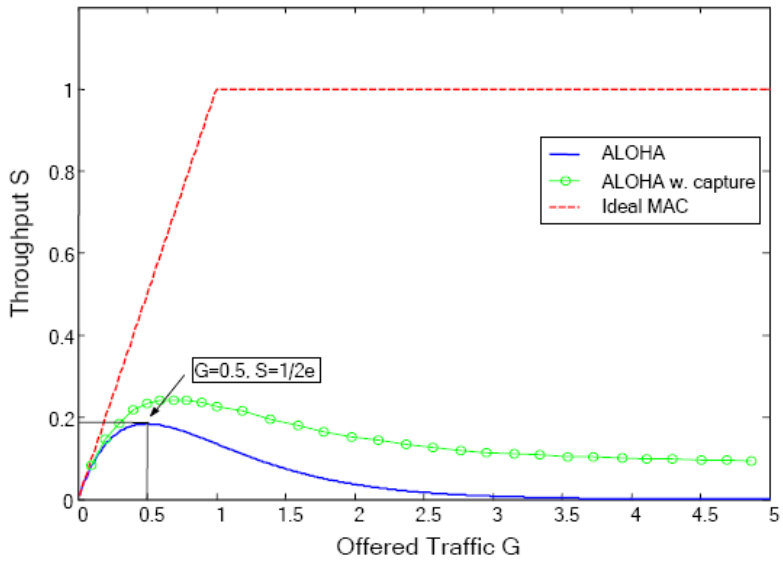


Figure 5.7: *Unslotted Aloha, Unslotted Aloha with capture effect and ideal MAC throughputs*

Albeit the capture effect can lead to an increased throughput, nevertheless it is not always desirable. In fact, it can bring to an unfairness access, favoring those terminals that are nearer to the receiver, as in centralized network architectures.

The Aloha protocol is suitable in environment whose traffic load is low. In this framework, it can be implemented with facility, leading to a low delay and to a good throughput, specially using its slotted version. In network characterized by an increasing traffic load the performance of the protocol goes down and it is necessary to address our attention to other, more sophisticated, random access protocols.

5.2.2. CSMA

Contrariwise of Aloha, the *CSMA* (*Carrier Sense Multiple Access*) protocol allows maintaining high the throughput even if the traffic offered by the network goes up.

The functioning of the protocol is quite simple: the terminal willing to transmit senses the channel for a time given by w and, if the channel is detected as idle for this time interval, it transmits. Vice versa, if the channel is busy, the terminal can use the following way:

- 0-persistent. The terminal will try to sense again the channel after a random delay
- 1-persistent. The terminal finishes to sense the channel for the time w and then it transmits
- *p*-persistent. The terminal adopts the 1-persistent strategy with probability given by *p* and the 0-persistent strategy with probability 1-*p*

The main limit of CSMA lies on the natural finite propagation velocity of the radiation. A terminal, in fact, could start to sense the channel without becoming aware of a transmission that currently it is occurring. This effect is well represented in the following figure.

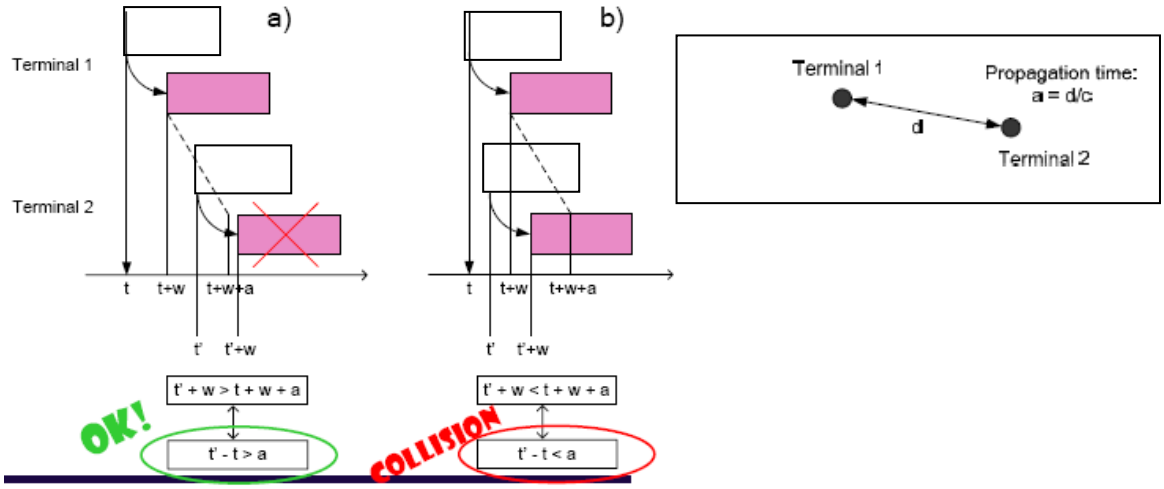


Figure 5.8: *Typical cause of collision for CSMA protocol*

Under the same hypothesis through which the Aloha throughput was derived and conjecturing that in case of collision it will be employed the 0-persistent strategy, it can be shown that the CSMA throughput is given by

$$S = \frac{Ge^{-aG}}{G(1+2a)+e^{-aG}},$$

where the propagation time a is the largest possible in the network, namely S is a lower bound for the system performance.

The following figure displays a graphical comparison of the protocols seen so far.

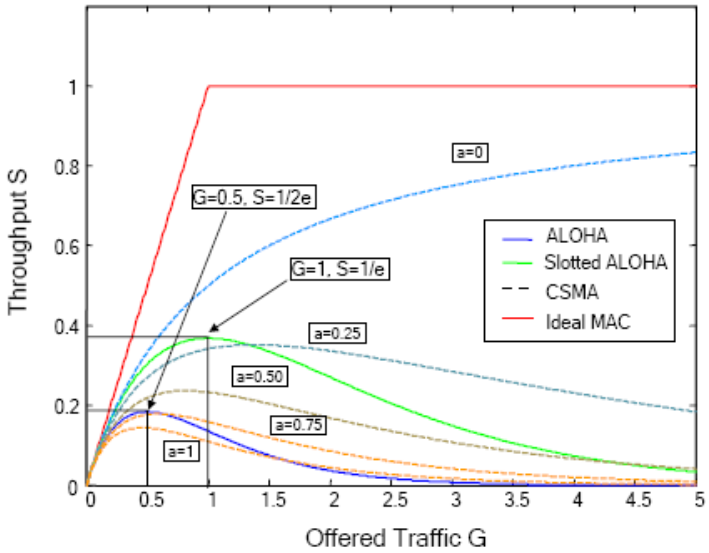


Figure 5.9: *Comparison between the throughputs seen until now*

Although better than Aloha, performance of CSMA can fall if the network scenario presents, as unfortunately often happens in wireless environments, two phenomena that considerably decrease the efficiency of the protocol: the *hidden terminal* and the *exposed terminal* issues. Before to enter in the details of protocols that relieves these problems (CA mechanisms), it will be provided a general description of the two, above quoted, phenomena.

Let us consider a transmission between a terminal S, the source of the transmission, and a terminal D, the destination. A terminal H is said hidden terminal when it is into the radio coverage of the destination D but not into that of the source S.

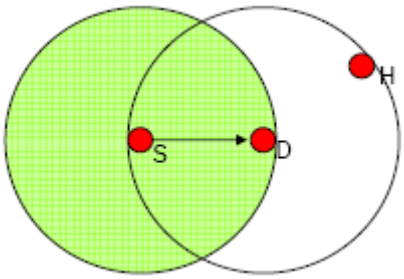


Figure 5.10: *Example of scenario relative to the hidden terminal phenomena*

Applying the CSMA protocol, if the terminal H wanted to transmit to D as well, it should sense the channel for the time interval *w* and, not able to detect the transmission of S, would interfere with the transmission of S, causing collision. Even though the terminal H had wanted to transmit to another terminal D2, it would have generated a collision in D. This last situation is pictured in the Figure 5.11.

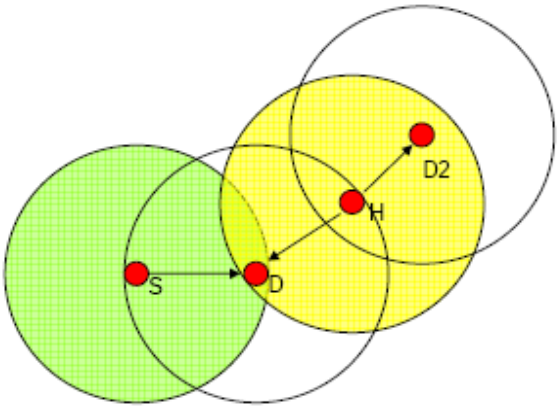


Figure 5.11: *An other hidden terminal scenario potentially subjected to collision*

An efficient protocol should mitigate this problem making the terminal H aware of being in the radio coverage of the destination D but not into that of S.

As regards the exposed terminal issue, a terminal is called exposed terminal if it is into the radio coverage of the transmitter but it is out of that of the receiver.

Let us consider a similar scenario in which a terminal S1 is the source and D1 is the receiver of their communication as is depicted in Figure 5.12.

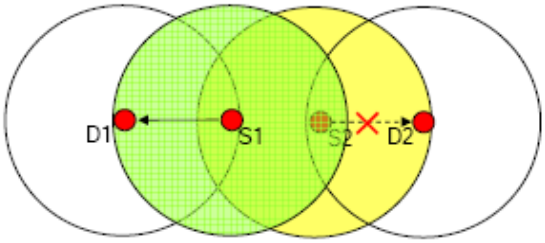


Figure 5.12: *Example of scenario for the exposed terminal issue*

In the above scenario the terminal S2, willing to transmit to D2, senses the channel and finds it busy because of S1->D1 transmission. For this reason S2 postpones its transmission. This phenomena does not lead to collisions but is a source of efficiency loss.

Conversely of the hidden terminal, this issue could be relieve if terminal S2 was aware of being in the radio coverage of S1 (transmitter) but out of the radio coverage of D1 (receiver). The goal of collision avoidance mechanisms is really to make a terminal more aware of its position and role relatively to a transmission.

5.2.3. In-Band CA: MACA

MACA is acronym for Medium Access with Collision Avoidance. It is not based on the carrier sense technique but it employs, before the transmission of a data packet, a change of control packet among the source and the receiver of the communication. This preparatory phase, called *handshaking*, allows preventing collisions between data packets, as soon as possible. The handshaking consists of three phases:

1. With reference to Figure 5.13, the source of transmission S sends to D, the receiver, a control packet called *RTS* (Request To Send) piggybacking the ID of S, that of D and the expected duration of the data that S is going to transmit

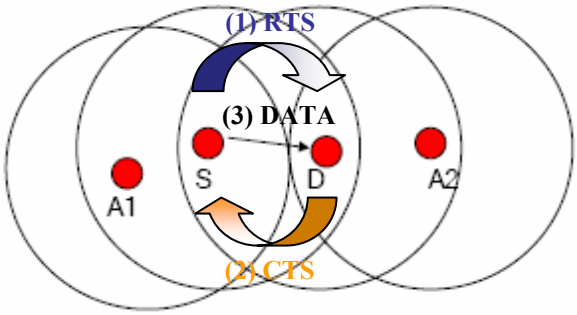


Figure 5.13: *Steps in the scenario considered for the handshaking*

The RTS packet allows to all the terminals in the radio coverage of S, as the terminal A1 (exposed terminal), making themselves aware about their inclusion within the radio coverage of S. As A1, each terminal that hears the RTS, will postpone its access to the medium, therefore avoiding a potential collision

2. The recipient of the RTS message, the terminal D, will send a *CTS* (Clear To Send) packet to S in order to trigger to the data packet transmission of S. As the RTS, the CTS piggybacks the ID of S, of D and the expected duration of the transmission. Each terminal in the radio coverage of D that will hear the CTS, as A2 (hidden terminal), will postpone its eventual transmission, independently from the listening of the previous RTS. At this point, it is important to notice that, if some terminals as A1 (an exposed terminal) do not hear the CTS, they will can transmit without causing collision in D, just because of its absence in the radio coverage of D. Evidently, we can assert the previous state since we are supposing of working with a symmetrical channel, namely, if a terminal T1 can hear another terminal T2 then T2 can hear T1 and, vice versa, if T1 can not hear T1 then T2 can not as well
3. After that S has heard the CTS, it can legitimately transmit its data packet. The collision on data packet is avoided by the fact that RTS and CTS, containing the expected duration of the data packet transmission, lead to a virtual carrier sense

Obviously, collisions can still occur but among RTS packets. This means that, in case of collision, no data packets will be corrupted and the channel will be uselessly busy for a short time, given that RTS packets are usually very short (20 Bytes).

5.2.4. Out-Of-Band CA: BTMA

As said above, Out-Of-Band collision avoidance procedures do not use the data channel but employ a dedicated control channel in which to transmit information for managing the access to the medium. Typically, as we are going to show, these kinds of information are simply sinusoidal tones, therefore not implying an organization of the control information into packets.

BTMA employs two channels: a message channel in which to transfer the data packets; a control channel, using a different carrier, employed for transferring the control information and whose denomination is *busy-tone channel*.

The concept on which it is based is quite simple: if a terminal T wants to transmit, it senses the busy-tone channel for a time t_d and, if it is detected as idle, is allowed transmitting. On the other hand, if the control channel is detected as busy, this means that in the data channel is occurring currently a data transmission implying to postpone the transmission of T.

Obviously, it is necessary a terminal that manages the busy-tone channel. In a network with centralized architecture it can be realized quite simply employing a Base Station. In fact, as soon as the base station senses the message channel as busy, it produces the tone on the busy-tone channel whose semantic is to hold back each other terminal willing to transmit.

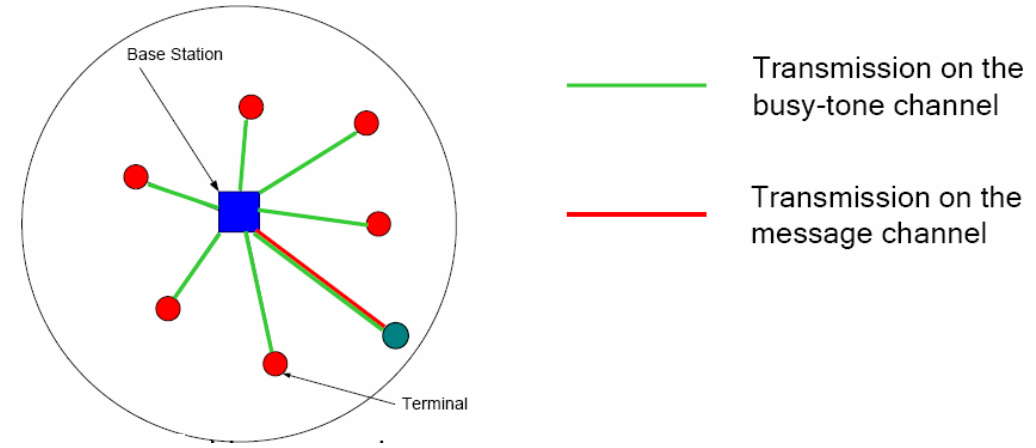


Figure 5.14: *Example of centralized architecture using BTMA protocol*

Analogous address has validity for a network with distributed architecture but characterized by a centralized organization.

Conversely, more complex is the implementation of the protocol if both the architecture and the organization are distributed. In that case, the terminal willing to transmit hears the busy-tone channel for t_d and if it is detected busy, the terminal puts off its transmission. On the contrary, if the control channel is idle, the terminal turns on its busy-tone channel and transmits data on the message channel. Any other terminal that senses the message channel as busy switches on its busy-tone signal, leading to inhibit the transmissions of those terminals that are within a two-hop range from the transmitter.

Obviously in this way we are relieving the network from the hidden terminal issue but the terminals inhibited are more than those enough for a no collision transmission. In other words, in this way we are preventing collision with the drawback of amplifying the exposed terminal phenomena. This effect is displayed in the following figure.

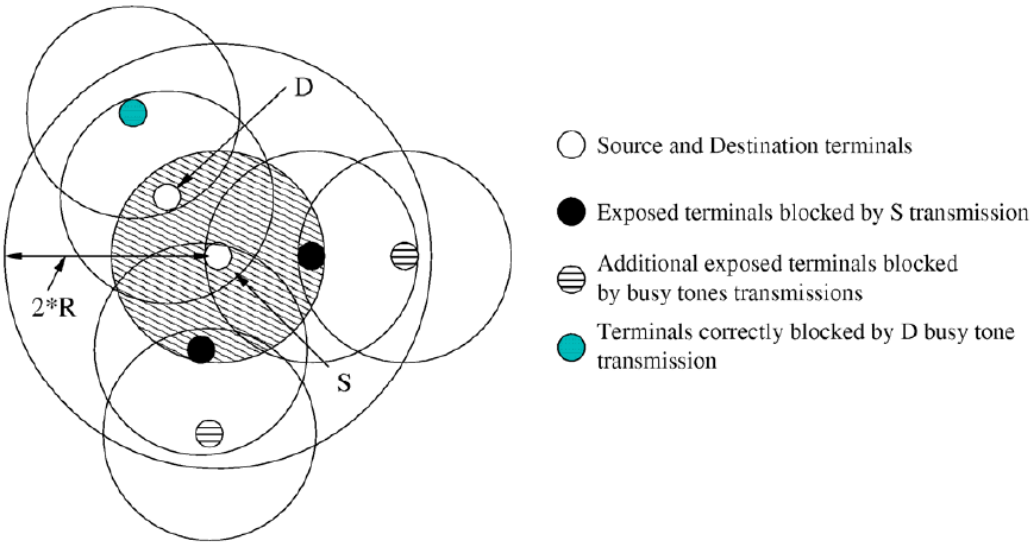


Figure 5.15: *Example of the amplification of the exposed terminal problem*

We cite only the fact that the evolution of BTMA, Dual BTMA (DBTMA), employs two control channels for knowing of the state of a terminal, transmitter or receiver, leading to mitigate the exposed terminal effect.

5.3. Scheduled protocols and hybrid protocols

Without to enter in their particularity of implementation, we can say that, as common goal, scheduled access protocols allow to a terminal in the network knowing which terminals are currently transmitting, leading to avoid collisions between data packets in a contention less fashion. For a centralized architecture and for a distributed architecture with a centralized organization where it can be defined a controller (Base Station, Access Point), suitable protocols are:

- Polling. The controller "calls" one terminal at the time
- Demand Assignment. The controller assigns a message channel after a request
- Static. A resource (time, frequency, code) is statically assigned to a terminal when it joins to the network

In distributed architectures with distributed organization it can be possible adopt the token protocol.

The hybrid protocols merge the contention less approach of scheduled protocols with that contention based characterizing by random access protocols. A terminal which wants to transmit emits randomly a first data packet that piggybacks a reservation request. If it is received correctly by the destination, then the protocol guaranteed that, thanks to the previous reservation, the remaining packets will be received without collisions.

Examples of hybrid protocols are PRMA (reserves slots in a TDMA frames) and RRA-ISA (reserves slots in a polling sequence).

5.4. Examples of standardized MAC protocols

Often, in order to increase the efficiency of a protocol, many of the random access protocols above quoted are combined jointly. This can lead to fill the drawbacks that a protocol could present in its "solitary" implementation. In the following it will be shown with some example that, effectively, the standards used for MAC protocol provide this characteristic.

First of all, we will start showing the MAC of IEEE 802.11 standard (WiFi), namely DFWMAC. Then it will be described the IrDA MAC, that is IrLAP and, finally, a MAC protocol proposed for the UWB systems.

5.4.1. MAC protocol for 802.11 standard: DFWMAC

The original definition of the IEEE 802.11 standard included the following characteristics.

	IEEE 802.11 Wireless LAN Standard					
Frequency	2.4GHz		2.4GHz		Infra-red (850mm~950mm)	
Spread Spectrum	DSSS		FHSS			
Baseband Modulation	DBPSK	DQPSK	2GFSK	1GFSK	16-PPM	4-PPM
Max bandwidth	1Mbps	2Mbps	1Mbps	2Mbps	1Mbps	2Mbps
Access Control	CSMA/CA, RTS/CTS					
Area(open air/office)	100~300m/20~100m				20~30m/about 5m	

Figure 5.16: *Main characteristics of the 802.11 wireless LAN standard*

Nowadays many successive versions has been developed in order to improve its characteristics:

- IEEE 802.11a. It reaches 54 Mbit/s using an OFDM modulation technique in the 5 GHz band. It holds the same MAC of 802.11
- IEEE 802.11b. It is the most diffused system and reaches 11 Mbit/s using the 2.4 GHz band with DSSS and FHSS modulation

- IEEE 802.11e. In the 5 GHz band as well, it employs an OFDM modulation. In order to manage opportunely real-time applications as audio and video streams, it introduces QoS mechanisms
- IEEE 802.11f. It is introduced an Interaccess Point Protocol by means of that to provide the possibility for the interconnection of access point of different brands
- IEEE 802.11g. It offers the same reachable bit rate of the 802.11a standard, namely 54 Mbit/s, in the same band and with the same modulation of the 802.11b standard (2.4 GHz, DSSS and FHSS). More over, it is assured backward compatibility with 802.11b
- IEEE 802.11h. It uses the 5 GHz band and, exploiting OFDM modulation, manages up to 54 Mbps. It allows a more suitable management of the spectral resources such as power control and dynamic frequency selection
- IEEE 802.11i. In the 5 GHz band, with OFDM modulation and managing at least 54 Mbps, it introduces security mechanisms as encryption and authentication
- IEEE / ETSI 802.11j. The spectral resource are allocated at 5 GHz, employing OFDM and GMSK modulation and leading to, at least, 54 Mbps. It represents the convergence of 802.11a and HiperLAN standards

The 802.11 standard foresees two distinct ways of multiple access:

- *Distributed Coordination Function* (*DCF*). It is a distributed modality of access based on the *CSMA/CA* MAC protocol. We will concentrate our attention on it
- *Point Coordination Function* (*PCF*). It is an optional centralized modality of access based on a mechanism of reservation and polling. We will not enter in its details

The DCF modality is based on a combination of CSMA and CA protocols (CSMA/CA). A mobile terminal using the DFWMAC can transfer a data packet in two different ways: 2-way and 4-way handshaking, both starting with the carrier sense procedure of CSMA on a slotted time axes.

In the 2-way handshaking a terminal T, that wants to transmit a packet, senses the channel for a time interval denominated *DIFS*. If T detects the channel as idle for a DIFS, then it transmits. Conversely, it waits until the channel become idle for a DIFS. At this time, for minimizing potential collisions with other terminals, any station waiting as T for its transmission, does not transmit immediately but begins the *backoff procedure*. It sets a random time interval and it is activated a countdown timer. The timer is:

- diminished until the channel remains idle
- frozen when the channel is detected as busy, because of another terminal having a lower backoff time
- restarted from the value to which it arrived when it was stopped, when the channel is afresh detected as idle for a DIFS

Finally, the terminal transmits when its backoff timer gets to zero.

Once the destination receives correctly the data packet, it waits for a *SIFS*, whose duration is lower than the DIFS length and sends to the transmitter an *ACK* whose semantic is to make it aware about the result of the communication. If the transmitter does not receive the ACK within a given time, it repeats its transmission following the same procedure described above.

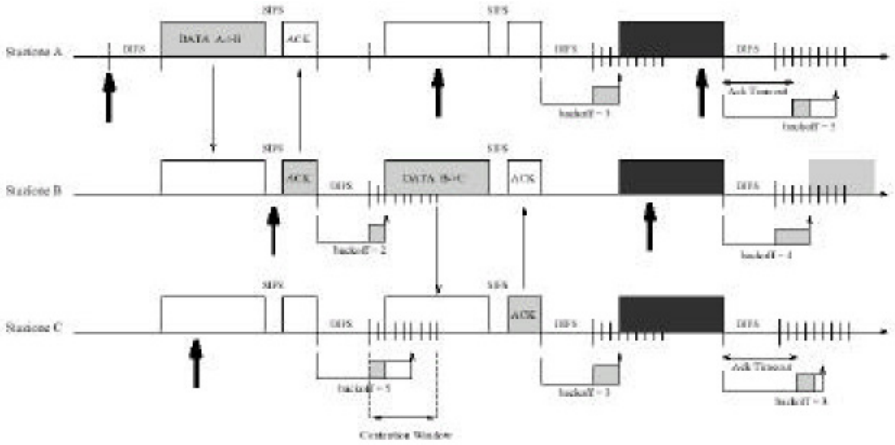


Figure 5.17: *Example of a 2-way handshaking in the DFWMAC*

As in the 2-way handshaking, in the 4-way handshaking as well, a terminal T willing to transmit to a terminal D, first of all follows the carrier sense procedure as above described. After, it begins the handshaking:

- T transmits the RTS to D
- When D receives the RTS, it replies with a CTS after a SIFS
- The source can transmits its packet, after a SIFS as well
- Eventually, after a correct delivery of the data packet, D sends to S the ACK

As noted in the previous section, the RTS and CTS packets piggyback the information relative to the duration of the communication. This leads to a virtual carrier sense which is exploited by the other stations sensing the channel. In particular these terminals, when hear the RTS/CTS, refresh a *Net Allocation Vector* (*NAV*) containing the information regarding the duration in which the channel will remain busy.

For what said above, the 4-way handshaking is employed when the network scenario is affected by the hidden and exposed terminals issue leading to a better efficiency of the protocol respect to what obtained using the 2-way technique.

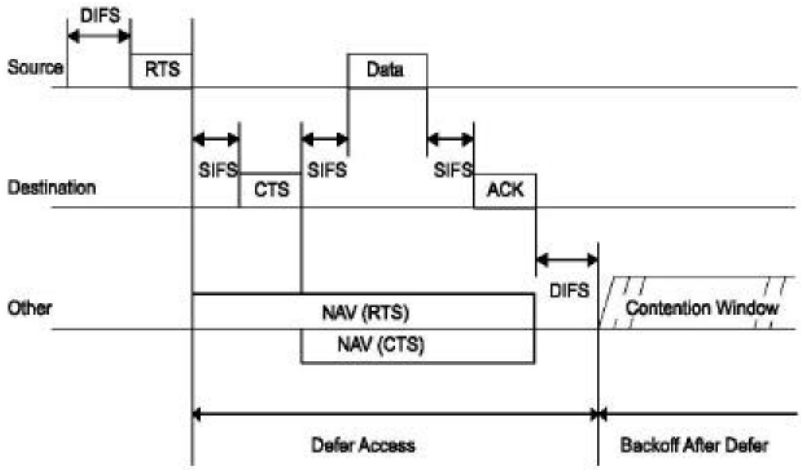


Figure 5.18: *Example of a 4-way handshaking in the DFWMAC*

When a data unit exceeds a given value, the protocol allows fragmenting it into multiple data units. In this case the protocol foresees a special transmission. The terminal that wants to transmit contends the channel by means of the backoff procedure. Obtained

the channel, it transmits sequentially the fragments in the way that, passed a SIFS from the reception of any ACK acknowledgmenting the previously packet, it emits the next packet.

One must notice that, because of the shorter duration of the SISF respect to the DIFS, in this way the remaining packets own a greater access priority. The packets sequentially transmitted piggyback:

- a number of sequence. It is equal for all the packets relative to a sequential transmission but is not the same between different sequential transmission
- a number of fragment. This numeration allows to the receiver reassembling the fragments, discarding possible duplicates

More over, data packets and ACK packets piggyback information relative to the expected duration of the next fragment.

When the transmitter does not receive an ACK of a fragment, it enters in the backoff procedure at the end of which, it restart its transmission beginning from the last packet unacknowledged.

The fragmentation technique can be combined with the 4-way handshaking. The RTS-CTS packets are transmitted only at the start of the sequence. The future fragments with the relative ACK will play the role of RTS and CTS. All the remaining procedure is identical to which above described.

5.4.2. MAC protocol for IrDA standard: IrLAP

The original definition of the *IrDA* (*Infrared Data Association*) standard included the following main characteristics.

	IrDA Standard
Wavelength	875 nm (with a tolerance around 30 nm)
Baseband Modulation	OOK (Zero Return with $T_P = 3/16 T_{bit}$)
Distance	Up to 1 mt
Viewing Angle	Between 15^o and 30^o
Rates Supported (bps)	9600, 19200, 38400, 57600, 115200
Access Control	IrLAP

Table 5.1: *Main characteristics for the IrDA LAN standard*

To date they have been introduced addictions and improvements to the standard as, for example, the capacity to transmit at available rates up to tens of Mbps. This can be obtained modifying the physical layer *IrPHY* (*SIR*, *FIR*, *VFIR*, etc.). For example, in place of OOK, FIR uses a 4PPM as modulation scheme.

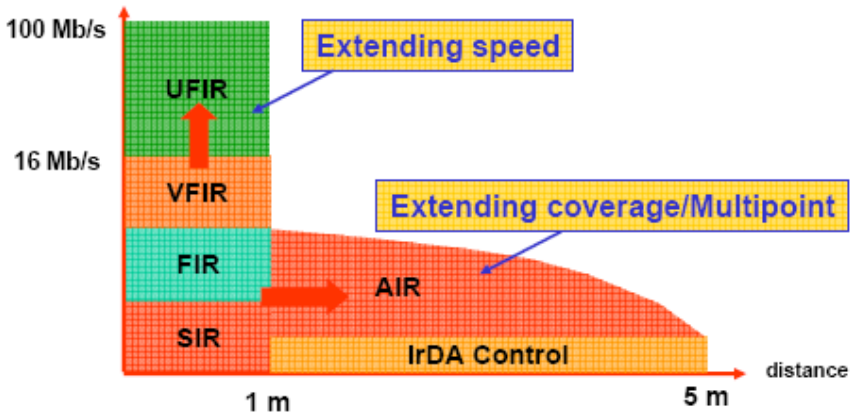


Figure 5.19: *Rates available in the IrDA physical layer evolution*

The IrDA data protocol stack is displayed in the following figure.

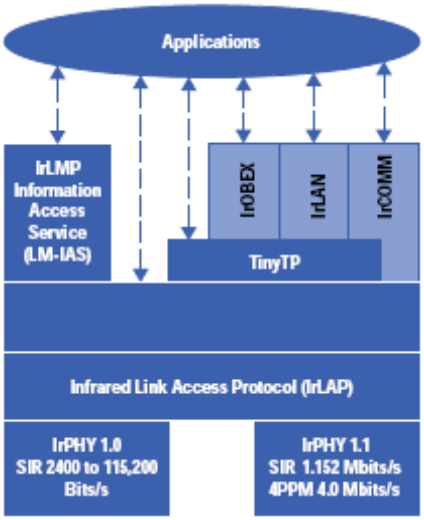


Figure 5.20: *The IrDA architecture*

The core of the IrDA architecture is constituted by *IrPHY*, *IrLAP* and *IrLMP*. *IrOBEX* is a protocol regarding the definitions for generic object exchange of an IrDA-compliant device. *IrLAN* is an IrDA specification for accessing to a LAN over an infrared medium. It allows a translation of protocol leading to a medium access that is a simplified version of the 802.11 MAC (DFWMAC) previously described. *IrCOMM* allows emulating serial and parallel port communications. Finally, *TinyTP* is the transport protocol used by the standard.

IrLAP is the MAC protocol of the stack. It has three distinct phases of operation: link initialization, nonoperational mode and operational mode.

The following figure displays the logical progression through the two modes above quoted.

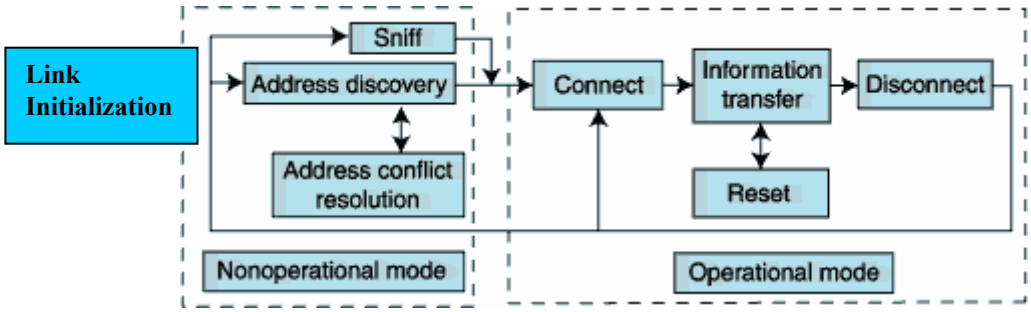


Figure 5.21: *IrLAP logical flow between nonoperational and operational mode*

Let us consider a terminal A willing to connect itself to the network. In the link initialization phase the terminal A chooses a random address of 32 bits.

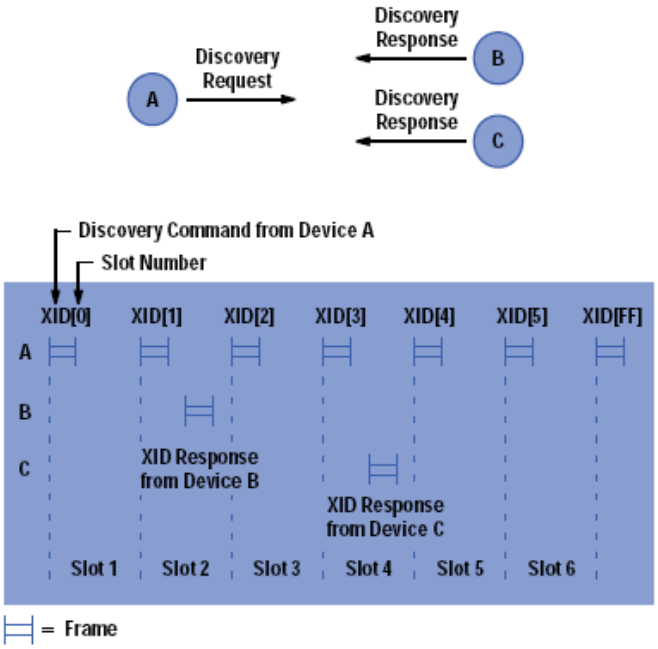


Figure 5.22: *The discovery procedure in IrDA*

Obtained its own address, the terminal A needs to know which terminal is within its field of view. Therefore, with reference to Figure 5.22, it broadcasts a discovery XID command frames marking the beginning of each slot. Every terminal that hears the discovery frame will choose randomly a slot in which to respond to A. In the case in Figure 5.22 the terminals labelled with B and C will respond respectively on the second and fourth slot. Finally, A acknowledges the discovery sending a discovery response frame. At the end of the discovery procedure A will know the addresses and other useful information about B and C and, vice versa, B and C will identify the terminal A. We should have to notice that, in the case under examination, for B and C no information is known about each other. In fact, because of the limited field of view, they are mutually hidden.

In the quite improbable case that two terminals chose identical addresses, an address conflict resolution procedure would be foreseen. The terminal that detects an address conflict starts the procedure sending, likewise the discovery procedure, an address resolution XID command to the terminal implicated in the conflict. This time, the communication interests only conflicting terminals. The station selects another address and

responds in a slot chosen randomly. It is foreseen the repetition of the procedure in the unlikely case of a relapse in the address conflict.

Only for completeness we cite the sniff procedure, which is a special low-power connection procedure intended to support devices with power-critical modes of operation.

Once the discovery and address resolution processes are completed, the terminal A, willing to connect for example with the terminal B, sends to it a *SNRM* (Set Normal Response Mode) command frame. If B accepts the connection, it replies with a *UA* (Unnumbered Acknowledge) frame. The terminals are ultimately in operational mode and the connection is so established: A is the primary devices and B the secondary.

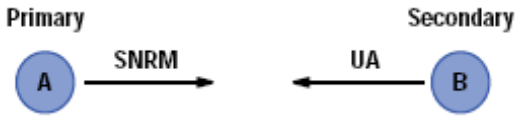


Figure 5.23: *Connection establishment*

Data exchange is performed in a master-slave mode, using a polling scheme in which the primary device sends command frames to the secondary device which responds with response frames.

Both terminals implicated in the communication can decided to disconnect. If it is the primary to wish it, it transmits a disconnect command to the secondary, which replies with an UA frame. Conversely, if the secondary desires to disconnect, it sends a request disconnect response to the primary, which responds with a disconnect command. At the end of this phase, both terminals are in disconnect mode and the medium is now available for any device willing to start the discovery, address resolution, or connection procedures.

Finally, reset is a command that, through a mutual agreement of the two stations, causes all counters and timers to be reset.

5.4.3. UWB systems: MAC protocols for the standards 802.15.3 and 802.15.4

Because of the strong power emission constraints imposed by regulatory bodies, UWB is rising as a technology particularly suited for transmissions characterized either by a high bit rate over a short range, or by a low bit rate over a long range, to which respectively correspond the 802.15.3 and 802.15.4 standards.

In 802.11.3 the MAC domain is denominated piconet and it is managed by a *PicoNet Coordinator* (*PNC*). When terminals wish to organize themselves into a network, they define a piconet to which can belong up to 256 terminals, each identified univocally by an address of 8 bits. One of those terminals is designed to be the PNC. The choice of the PNC depends on characteristics of the terminal such as maximum power level in transmission and maximum transmission rate. A terminal willing to join to the piconet listens the broadcast messages sent by the PNCs of several piconets and selects one of them. If it has PNC potentialities better than the pre-existent PNC, a special PNC handover procedure is foreseen for changing PNC.

All the operations of terminals belonging at the same piconet must follow a global timing given by a superframe defined by the PNC. The superframe contains the following three periods:

- *Beacon Period*. It is used by the PNC to broadcast information to all terminals within a piconet such as the *CTAs* in which to transmit in the next *CFP*
- *Contention Access Period* (*CAP*). In this period the terminals can demand to the PNC reserved Channel Time Allocations (CTAs) in a CSMA/CA fashion. CAP it also used by terminals for transmitting small amount of data to other terminals
- *Contention Free Period* (*CFP*). It is used for the transmissions of those terminals to which the PNC, using the beacon period, has granted CTAs in a TDMA fashion. It can be used also for controlling traffic during Managements CTAs (*MCTAs*)

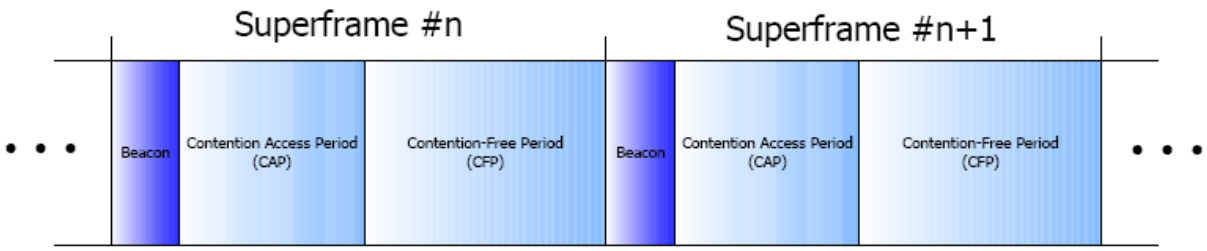


Figure 5.24: *Superframe used by PNC for the piconet global timing in 802.15.3 MAC*

We must notice that, even though the PNC schedules CTAs accepting *Channel Time Request* (*CTR*) during the CAP, it is not involved in data packet exchange.

Finally, whereas the standard completely defines procedure for intra-piconet operations, it does not do the same for inter-piconet operations.

As regards the low bit rate/medium-to-long range case, the standardization process is quite recently and dates back to the first meeting, in May 2004, of the IEEE 802.15.4a Task Group. The 802.15.4 MAC protocol remembers the 802.15.3 MAC, having anyhow different nomenclatures and functionalities.

Its architecture is distributed with a centralized organization as well. The MAC domain, formed by a group of terminals, is called Personal Area Network (PAN) and it is manages by a coordinator which must be a FFD (Full Function Device). Two different types of devices are defined:

- *Reduced Function Device* (*RFD*). This kind of device contains a subset of the IEEE 802.15.4 features. The RFD cans only acts as a device in a network. Normally the RFD would be used for battery powered devices, since a device with a pure device capability is very power efficient. Examples of RFDs could be light switches and temperature sensors
- *Full Function Device* (*FFD*). This device contains all of the IEEE 802.15.4 features. The FFD can acts in a network as device or coordinator. The FFD is targeted to be used for backbone powered devices. A FFD that takes on the role of coordinator uses significantly more power than a RFD in the device role and is more complex. Examples of FFDs could be light controllers with router capability (light bulbs) and PAN coordinators (main network controllers)

More over, the 802.15.4 MAC foresees two different topologies for a PAN: star topology and peer-to-peer topology. The terminals belonging to a PAN can be of three types: RFD, FFD and PAN Coordinator, which must be however a FFD.

A PAN (both star and peer-to-peer) can operate in either beacon mode or non-beacon mode. In beacon mode, following a TDMA fashion, the coordinator within the PAN transmits synchronization frames (superframes) to its associated devices. All data transmissions occur, in a CSMA/CA fashion, in the Content Access Period (CAP), subdivided into *Time Slots* (*TS*). As in 802.15.3, it is defined a Contention Free Period (CFP), used however for low-latency devices. Conversely by 802.15.3, it is an optional field.

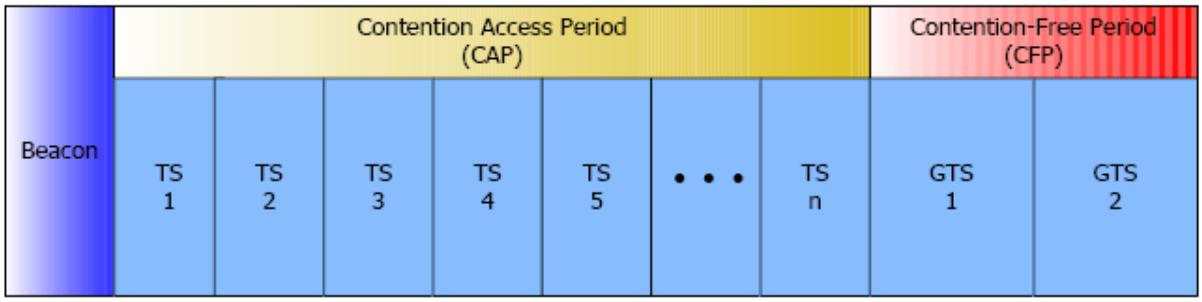


Figure 5.25: *Superframe used by a PAN Coordinator in 802.15.4 MAC (beacon mode)*

In non-beacon mode data transmissions can take place at any time using CSMA/CA. Since in this mode there is not the need to define a slotted time axis, the functionalities of the beacon period are reduced to broadcast the ID of the PAN and to transmit information available to those devices willing to join to the PAN.

5.4.4. The $(UWB)^2$ MAC protocol

The $(UWB)^2$ protocol is specifically designed for low data rate networks. In the code assignment it uses a hybrid scheme, that is to say are employed in the communication either a common control channel, or a dedicated data channel, respectively identified by a common TH code and a transmitter TH code, which can be obtained directly by means of the MAC ID of the transmitter using appropriated algorithms.

As first step, in the next we are going to analyze the transmission procedure of a terminal T towards a terminal R.

The protocol foresees the following steps:

1. The terminal T periodically checks the transmission queue and, in the case it finds MACPDUs to transmit, it extracts the ID of the receiver from the PDU
2. T determines the overall number N_{PDU} of MACPDUs directed to the receiver R
3. If other MACPDUs has been sent to R in the last T_{ACTIVE} seconds, T considers R as an active receiver, and moves to the step 5 of the transmission procedure
4. If R is not an Active receiver, T generates a *Link Establish PDU* (*LE PDU*), which goal is to produce in the receiver R the relative *Link Confirm PDU* (*LC PDU*)
5. The terminal T sends the LE PDU, using the Common TH code, and waits for the Link Confirm PDU of the receiver R
6. If the LC PDU is not received within the interval T_{LC} the terminal T tries again, coming back to the five point of the procedure. The retransmission of the LE PDU can occur maximum for N_{LC} times, after that the transmission is considered to be failed
7. Having received the LC PDU from R, T transmits the DATA PDU using the TH code previously communicated in the LE PDU.
8. If the DATA PDU requires it, T waits for the ACK relative to the packet sent
9. If the ACK is not received by the terminal T, then it retransmits the same packet using a backoff scheme
10. On the contrary, got the ACK, T checks again if other packets directed to R are into the queue, in which case it starts again the procedure in order to transmit the next packet

In the following figures are shown the structures of the LE PDU, the DATA PDU and the flow chart of the transmission procedure.

The structure of the LE PDU is the following:

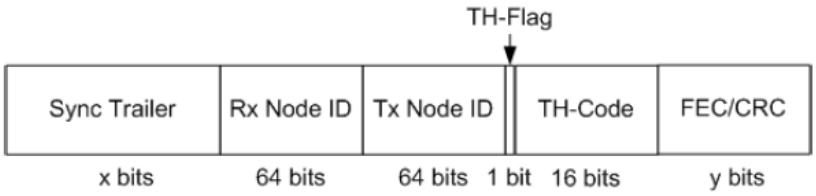


Figure 5.26: *Structure of the LE PDU in (UWB)*[2]

The fields have the following meaning:

- Sync Trailer is used to synchronize the terminals involved in the communication
- Rx Node ID is the MAC ID of receiver R
- Tx Node ID is the MAC ID of transmitter T. It can be associated univocally to the Transmission TH code
- TH-Flag is a flag set to true if the standard TH code associated to Tx Node ID is going to be adopted. The flag is instead set to false if a different TH code will be adopted
- TH-Code is an optional field used when the TH-Flag is set to false, in which case the information on the TH code to be adopted is provided in this field
- FEC/CRC is the field containing the bits for the error correction/revelation

The structure of the DATA PDU is the following:

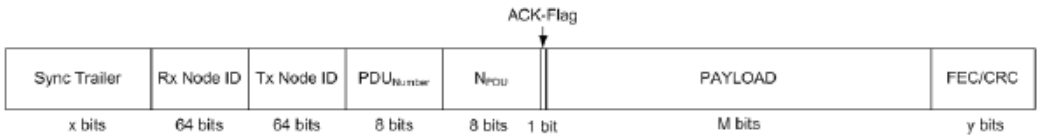


Figure 5.27: *Structure of the DATA PDU in (UWB)*[2]

The fields have the following meaning:

- Sync Trailer is used to synchronize the terminals involved in the communication

- Header, including the fields Tx Node ID, Rx Node ID, PDU_{Number} (identification number of the PDU) and N_{PDU} (number of PDU currently in the queue)
- ACK flag communicates to the receiver R to answer with an ACK PDU to the DATA PDU sent
- Payload is the field containing the data information
- FEC/CRC contains the bits necessary for the error correction/revelation.

Finally, the flow chart of the transmission procedure is shown in the Figure 5.28.

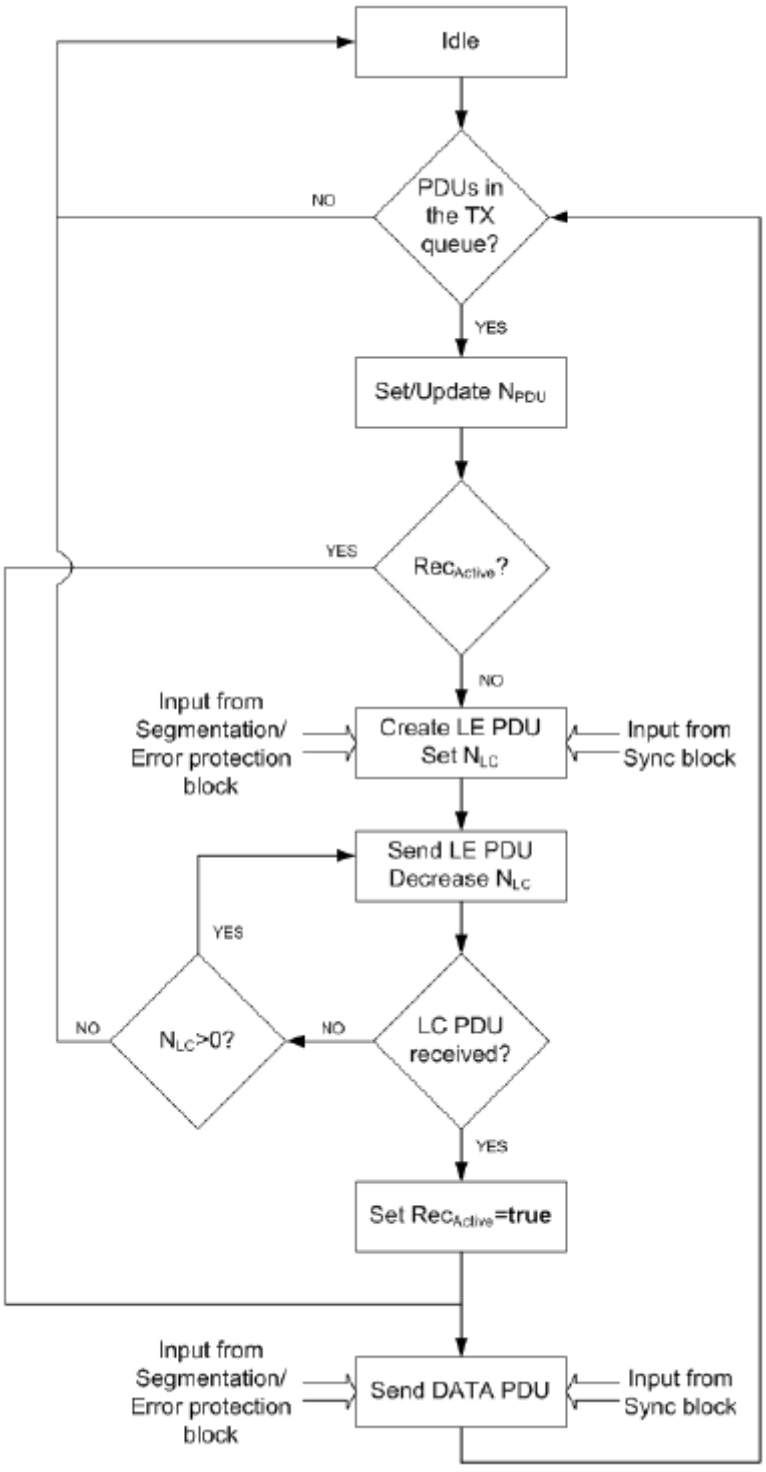


Figure 5.28: *Flow chart of the transmission procedure in (UWB)*[2]

Having seen the procedure to follow by the terminal T, now we are going to look at the reception procedure, namely the steps that the receiver has to follow when it receives a PDU from the terminal T. Considering the case in which no broadcast packets are going to send, the protocol foresees the following points:

1. The terminal R, which state is currently Idle, listens the common control channel, looking for its MAC ID. If it is not found, then the reception procedure ends. Vice versa, given that R is listening the common channel and no one broadcasting packet will be send, R has received a LE PDU from the terminal T
2. R generates and sends a LC PDU, which goal is to provide the availability to establish the link with T. R changes its state in active receiver and switches the code to the transmission TH code communicated by T in the LE PDU
3. If no data packets are received within the time T_{DATA} then R returns to the idle state and the reception procedure ends
4. Vice versa, receiving a DATA PDU, if the ACK flag is set to true, R generates the ACK PDU and sends it to T, reporting in the appropriate field the status of the reception. Moreover, if $N_{PDU} > 0$ then R remains in the active state waiting for the remaining PDU whereas, if $N_{PDU} = 0$, then the whole reception is completed and R turns to the Idle state

In the following the structures of the LC PDU, the ACK PDU and the flow chart of the reception procedure are provided.

The structure of the LE PDU is the following:

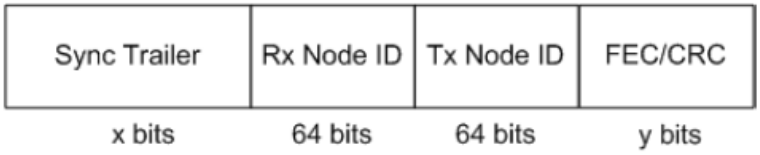


Figure 5.29: *Structure of the LC PDU in (UWB)*[2]

The fields have the following meaning:

- Sync Trailer is used to synchronize the terminals involved in the communication
- Rx Node ID is the MAC ID of receiver R
- Tx Node ID is the MAC ID of transmitter T. It can be associated univocally to the Transmission TH code
- FEC/CRC is the field containing the bits for the error correction/revelation.

The structure of the ACK PDU is the following:

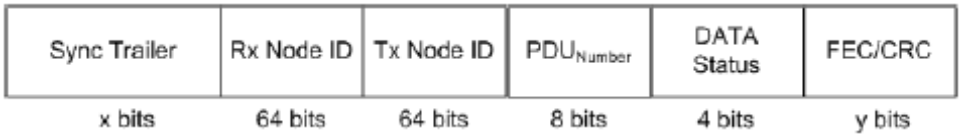


Figure 5.30: *Structure of the ACK PDU in (UWB)*[2]

The fields have the following meaning:

- Sync Trailer is used to synchronize the terminals involved in the communication
- Rx Node ID is the MAC ID of receiver R
- Tx Node ID is the MAC ID of transmitter T
- PDU_{Number} is the number of the DATA PDUs that R expects again from T
- FEC/CRC is the field containing the bits for the error correction/revelation.

Finally, the flow chart of the reception procedure is given in the Figure 5.31.

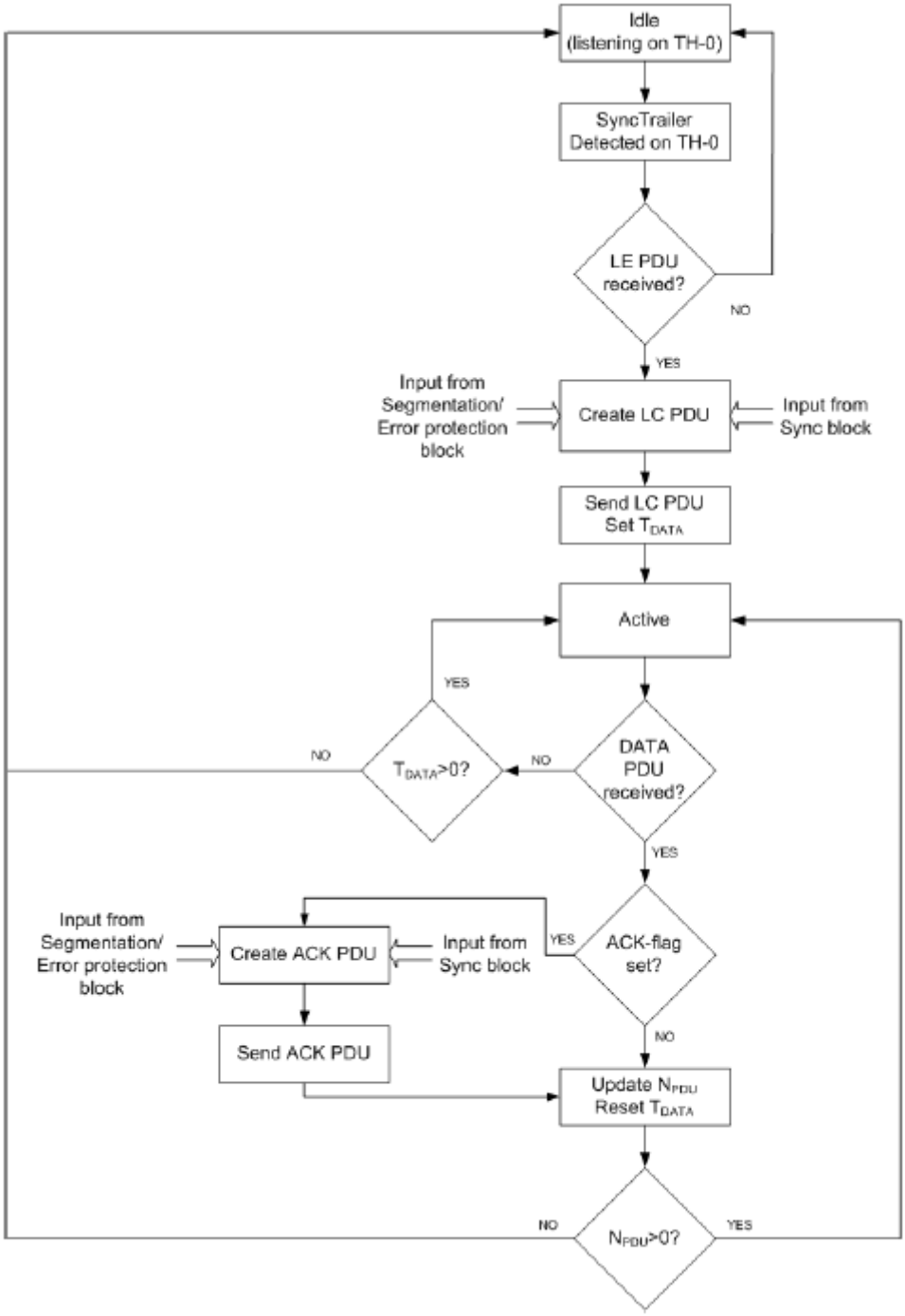


Figure 5.31: *Flow chart of the reception procedure in (UWB)*[2]

Chapter 6

Implementation of $(UWB)^2$ over an Optical System: Hardware's Design

Contents

6.1 Introduction 94
6.2 Optical transmitter 95
6.3 Optical receiver 102
6.4 Optical terminal 105
6.5 Optical system 106
6.6 Masters, slaves and relative EPLDs 108

6.1. Introduction

In this chapter we are going to expose the hardware architecture used for implementing the $(UWB)^2$ MAC protocol over optical links.

What we want to develop is an optical system formed by four terminals. At the physical layer each terminal can both transmit, and receive. Hence, at least as regards the physical layer, differences among them do not exist. At the MAC layer, two terminals are masters of the communication, and the remaining two are slaves. This choice was a hardware constraint related to the restricted resources of the EPLDs. The terminals transmit to a given data rates, using an OOK modulation and employing the $(UWB)^2$ MAC protocol. The presence of the interference between the slaves is ensured through the use in the masters of an array of two LEDs.

First of all, in order to allow in the future improvements and additions, in every step followed in the design of the system, it has been adopted a modular approach. To give an example, an OOK modulation has been chosen; anyhow any other kind of modulation technique can be added, without modifying the remaining architecture of the system.

Following this policy, the first step of the project consisted in the design of the optical part of the system. In particular this required to choose the LEDs to use, the photodiodes, the amplifiers, the comparators, and so on. More over, as we will see in the following of this chapter, its planning has been realized taking into account the specifications of interfacing of the EPLDs that "will contain" the protocol.

Once tested the good functioning of the physical layer, the $(UWB)^2$ MAC protocol has been implemented using an appropriate EPLD[3]. To this end, the protocol has been developed and simulated, by means of the software *Max+Plus II*. Once verified its functioning in the case of a communication between two terminals, it has been tested working in a scenario characterized by four terminals, still through the software simulation. In order to evaluate the different behaviours of the protocol at different data rates, a set of tests has been performed, each characterized by a distinct data rate. The rates used in the testing belong to the interval [4 Kbps, 256 Kbps]. The simulation effectively emphasizes that as greater is the data rates, as worse is the throughput obtained.

Finally, the physical optical system has been constructed.

[3] in terms of logical ports, number of inputs and outputs, needful to implement the protocol

6.2. Optical transmitter

The components chosen for the hardware design of the optical transmitter were an IRED L7558, from Hamamatsu and an AND gate, the CD74AC08 from Texas Instruments.

The IRED L7558 admits to manage, as input, signals until 50 MHz, allowing a radiant flux until 250 mW. The typical peak emission wavelength is 850 nm. More detailed characteristics can be found in its datasheet, attached in the annex AII.2.

The AND gate CD74AC08 is employed as circuit of excitation and gets as output the current to pass to the IRED, which finally will transmit the signal. Its datasheet is provided in the annex AII.1.

The circuital scheme of the optical transmitter is shown in the Figure 6.1 and has been designed through the software application Protel 99 SE.

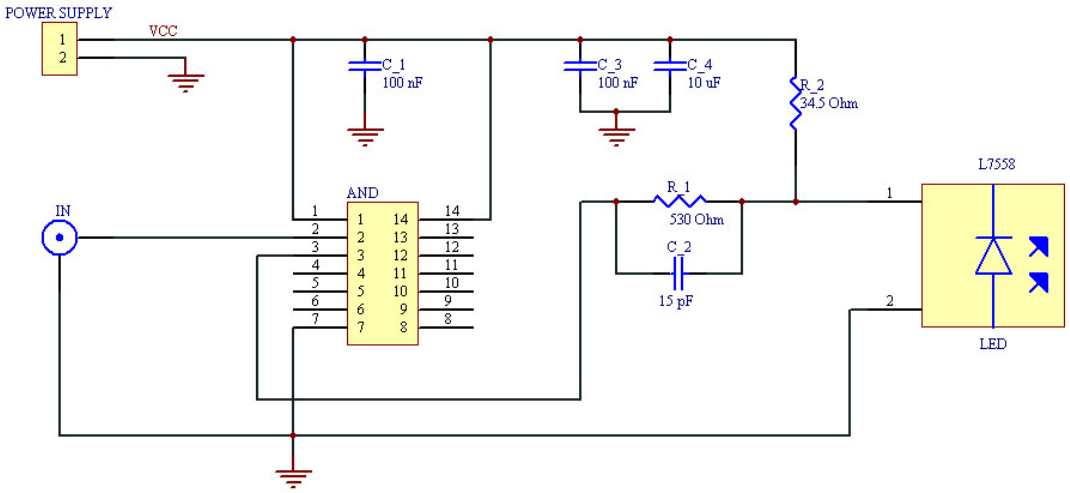


Figure 6.1: *Circuital scheme of the optical transmitter*

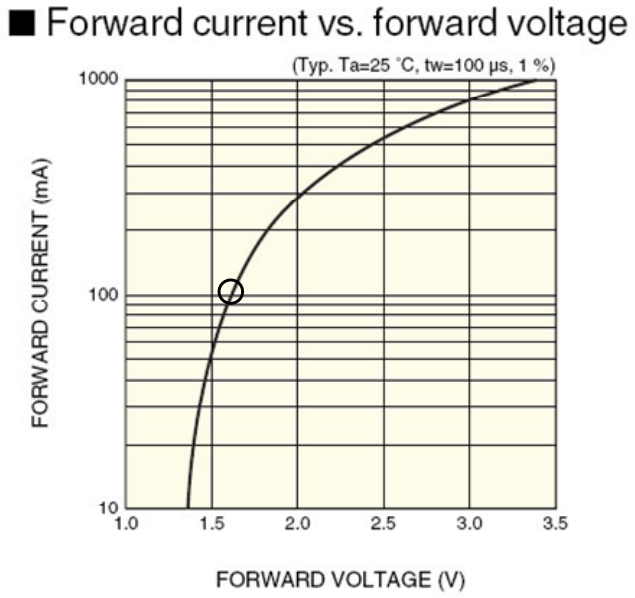
The power supply provides an alimentation of 5 V; the stability of its line is managed by the three capacitors C_1 = C_3 = 100 nF, C_4 = 10 μF.

As said before, the physical interface between the original signal to transmit and the LED is constituted by the AND gate CD74AC08. Its output is passed to a circuit of pre-emphasis, which function consists in the addition of a peak of current to that of polarization coming from the resistor R_2.

The polarization current, as said previously, proceeds from R_2. Its value has been calculated taking into account that, willing to use a forward voltage of 1.55 V, it is

necessary to use a forward current of 100 mA, which finally leads to a radiant flux of 30 mW. Hence, for what said, the value of the resistor is calculated as:

$$I_F = \frac{V_{CC} - V_F}{R\_2} = \frac{5 - 1.55}{R\_2} = 100\,mA \rightarrow R\_2 = 34.5\,\Omega\,.$$



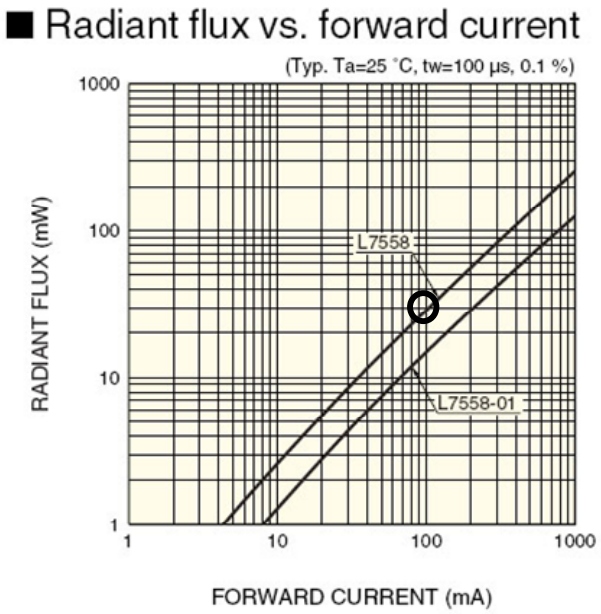


Figure 6.2: *Some input-output characteristics of the LED L7558 useful to design the physical layer of the optical system*

As regards the circuit of pre-emphasis, first of all it has been fixed the value of the capacitor, set equal to 15 pF. Secondly, taking into account that this circuit must not limit the rate of the system, it has been chosen a cut frequency equal to ten times that of the maximum working frequency, set to 512 KHz. Therefore, for what said, the value of the resistor R_1 is calculated as:

$$f_{cut} = 10 \cdot f_w = \frac{1}{2\pi \cdot R\_1 \cdot C\_2} \rightarrow \left(f_w = 512\,Mhz, C\_2 = 15\,pF\right) \rightarrow \; R\_1 = 530\,\Omega\,.$$

The appearance of a single optical transmitter is displayed in the following figure.

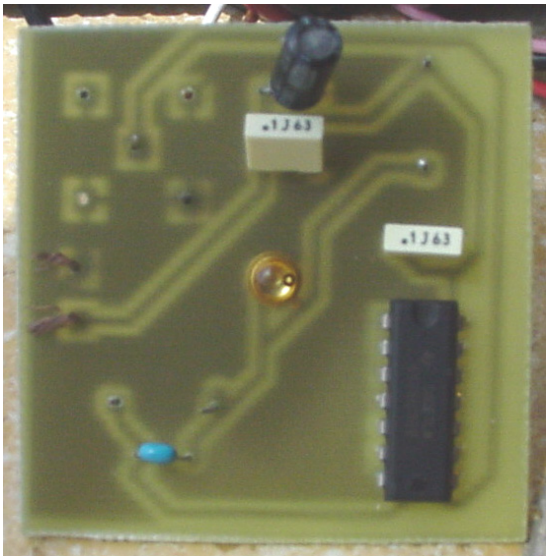


Figure 6.3: *Physical appearance of the optical transmitter*

Subsequently, both in order to increase the radiant flux at the receiver of the data, and to generate a perceptible and sufficient interference in the other receivers, an array of two LEDs has been designed and employed. To this end, it has been exploited the fact that the AND port has four outputs, two of which have been used.

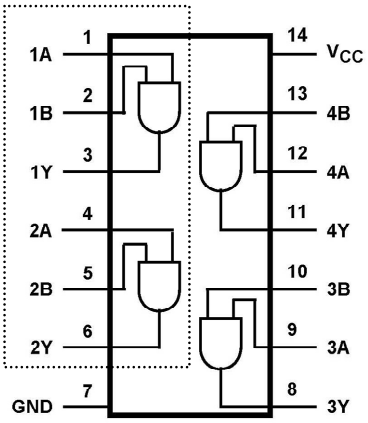


Figure 6.4: *Outputs of the AND Port CD74AC08 used for the LED array*

The circuital scheme of the optical transmitter using the array of LEDs is shown in the Figure 6.5.

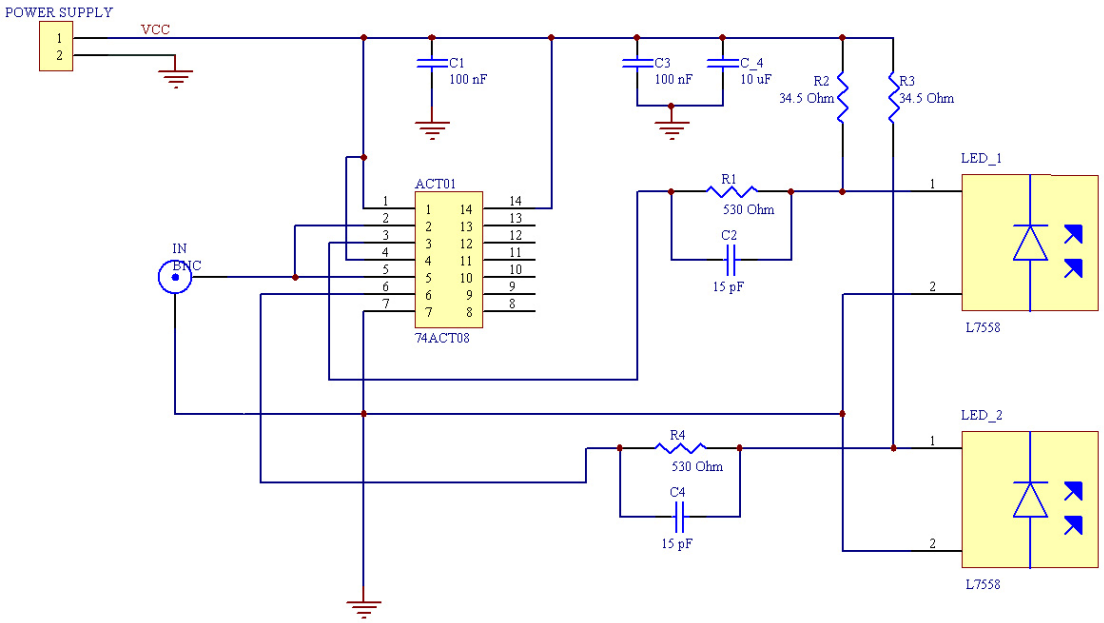


Figure 6.5: *Circuital scheme of the optical transmitter using the array of two LEDs*

The look of a single optical transmitter using the above described array is displayed in the following figure.

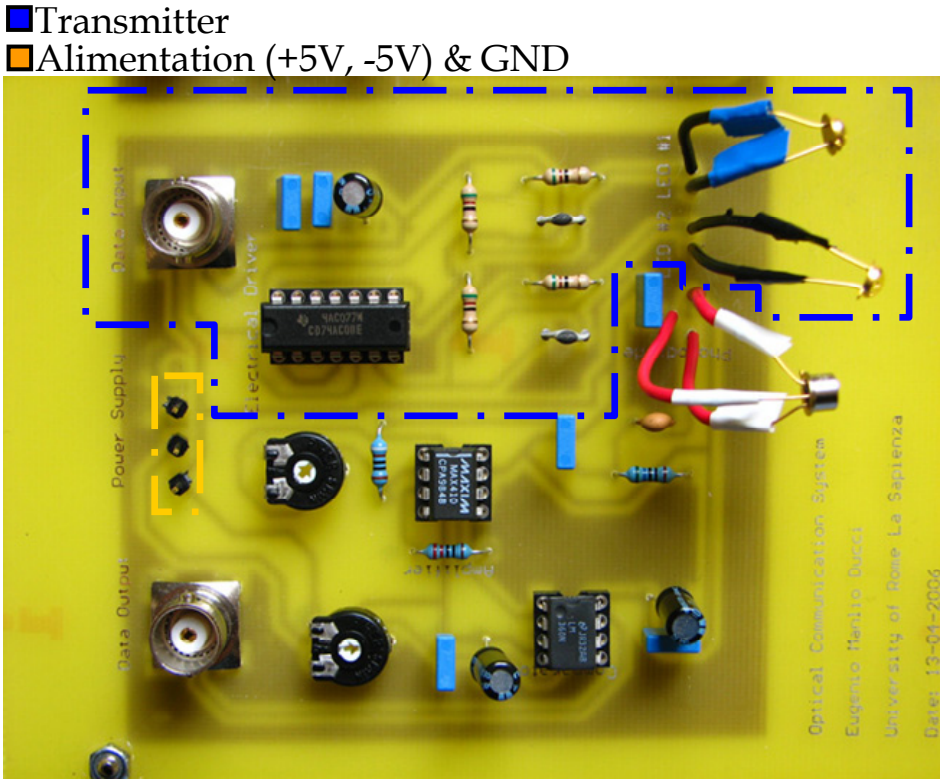


Figure 6.6: *Physical aspect of the optical transmitter using the array of LEDs*

As regards the transmitter, is interesting to point out some outputs of some circuits constituting it.

First of all, how seen above, the output of the AND gate CD74AC08 has to be TTL. Effectively, the following figure points out this nature of the signal.

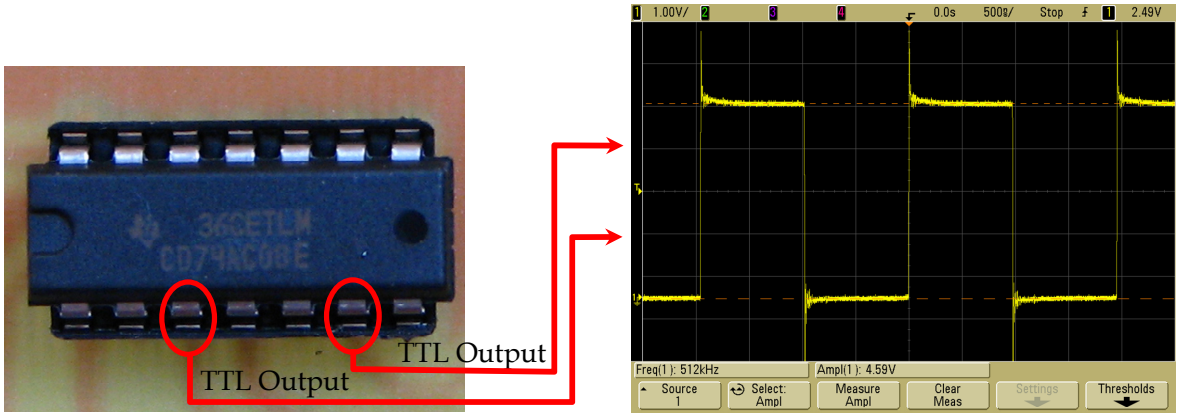


Figure 6.7: *Output of the AND gate CD74AC08*

Secondly, in the design of the optical transmitter we employed a circuit of pre-emphasis. The following figure represents the signal obtained at the output of this circuit. How evident, its function is to filter the low frequency of the signal, so that the LED can respond better to it.

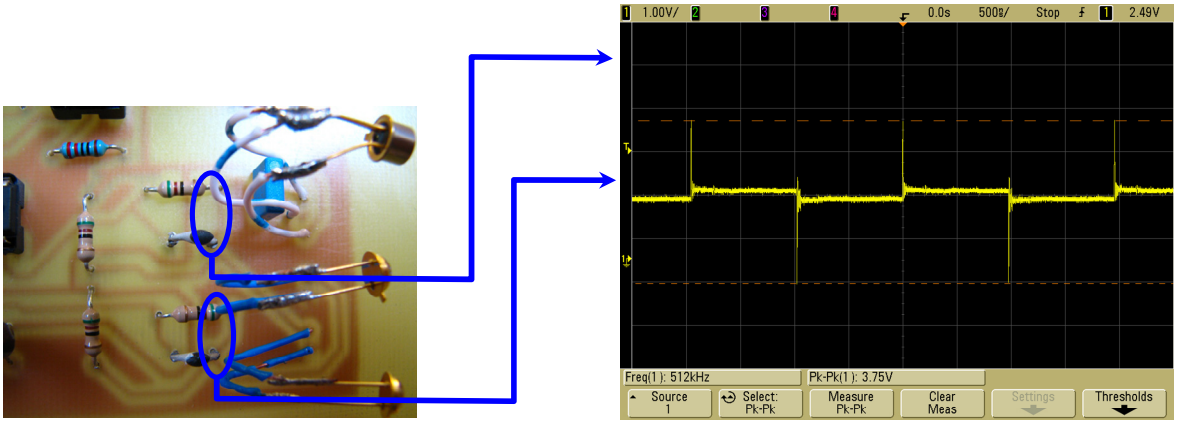


Figure 6.8: *Output of the circuit of pre-emphasis*

Finally, given that the final idea is to build an optical system formed by four terminals, two masters and two slaves of the communication, one LED of each master will be directed toward the photodiode of the relative slave and the other LED, whose output is exactly the same of the first LED (in particular, without phase displacement), will be focused on the photodiode of the slave not interested in the communication, generating in this way the wanted interference.

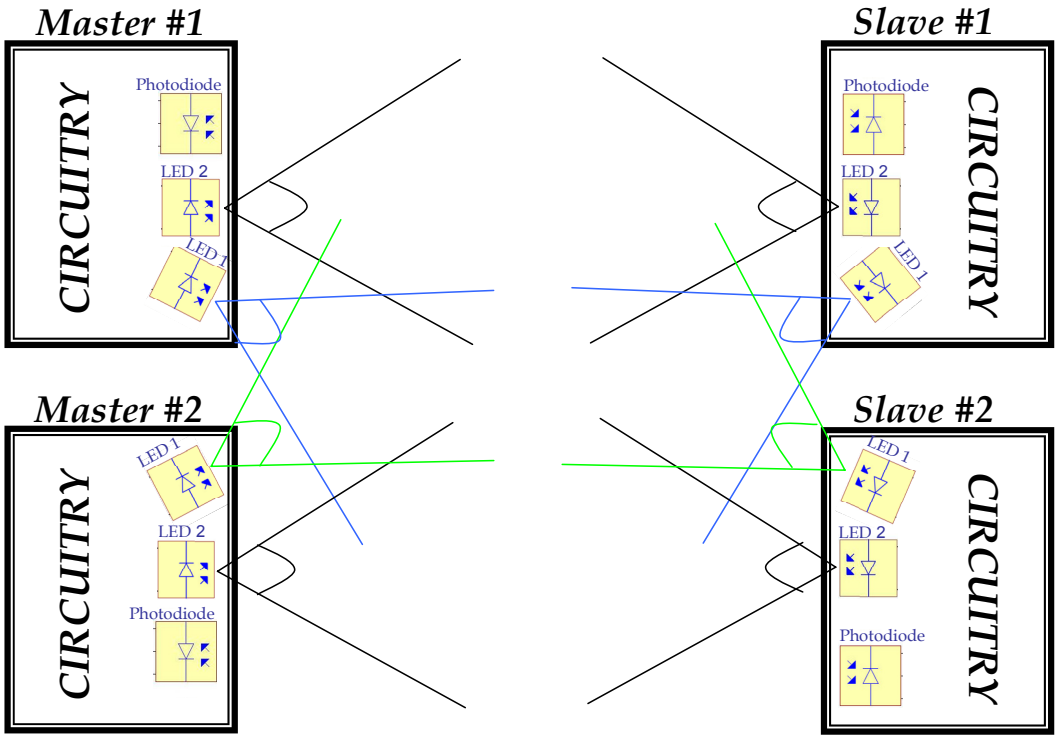


Figure 6.9: *Block scheme of the physical optical system*

The previous scheme, once realized via hardware, assumes the following aspect.

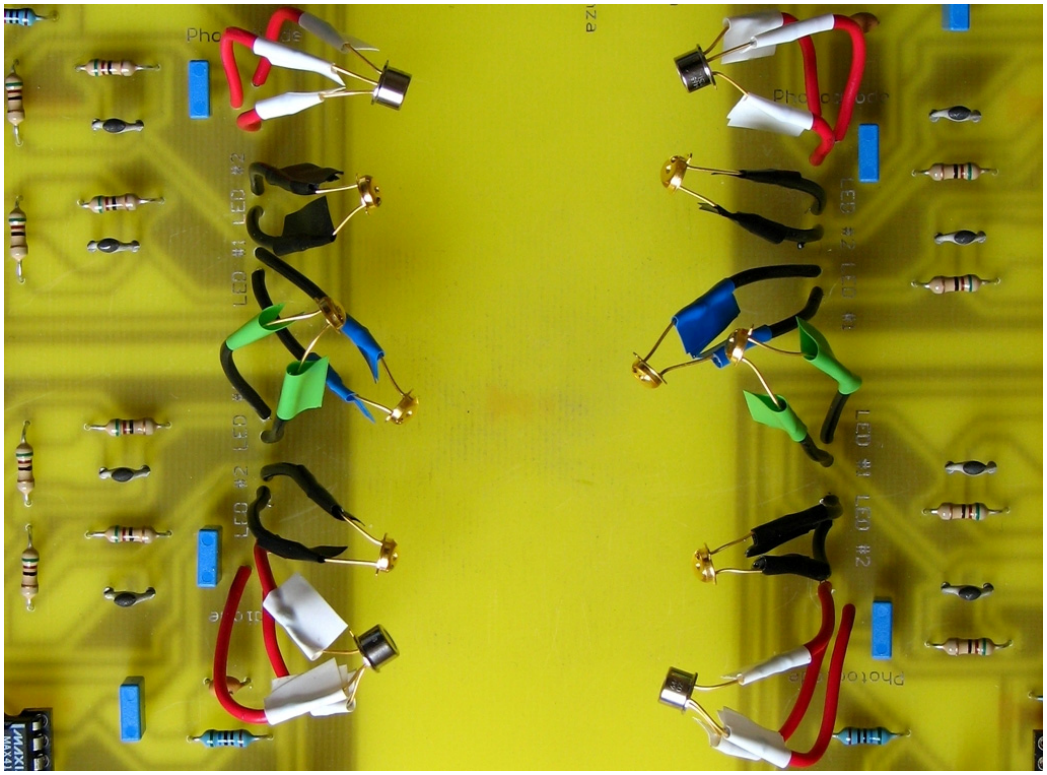


Figure 6.10: *Hardware implementation of the block scheme represented in the Figure 6.9*

6.3. Optical receiver

The components employed for the hardware design of the optical receiver were a photodiode S6468-02, from Hamamatsu, an amplifier, the MAX410 from Maxim, and a comparator, the LM360 from National semiconductor.

The S6468-02 is a Si PIN photodiode in which is already present a preamplifier and that can receive signals in input until 35 MHz. More detailed characteristics can be found in its datasheet, attached in the annex AII.3.

The amplifier MAX410, low noise and of 28 MHz of bandwidth, allow getting as output a signal whose amplitude, of 2-3 V, is such that the comparator can work properly. Its datasheet is given in the annex AII.4.

The high speed differential comparator LM360, whose datasheet is presented in the annex AII.5, is necessary to get a TTL signal, namely a signal which value is either 0 V or 5 V. This signal will be passed as input to the relative EPLD.

The circuital scheme of the optical receiver is shown in the Figure 6.11.

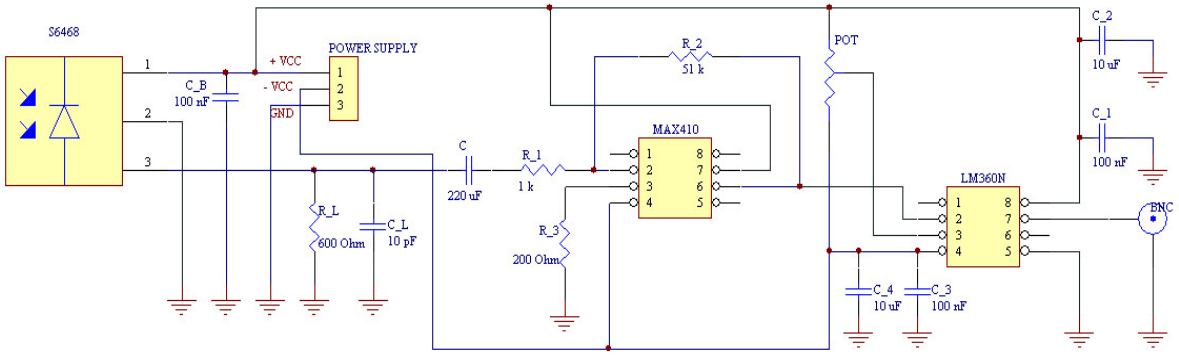


Figure 6.11: *Circuital scheme of the optical receiver*

The values of the resistors (R_L, R_1, R_2, and R_3) and of the capacitors (C_B, C_L, C, C_1, C_2, C_3, and C_4) can be directly got from the datasheets of the photodiode, of the amplifier and of the comparator.

The aspect of the optical receiver, without EPLD, is displayed in the following figure.

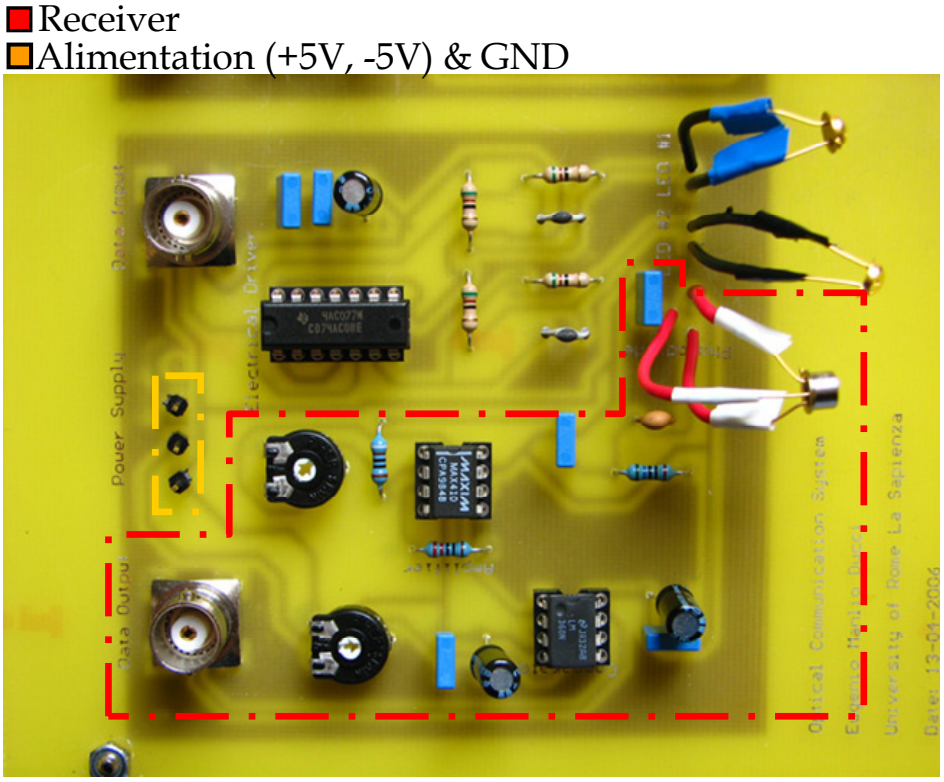


Figure 6.12: *Appearance of the optical receiver*

How done for the transmitter, now we are going to show some important output in the reception chain of the receiver.

First of all, after the acquisition of the optical signal by the photodiode and after to have removed the DC component through the capacitor C, the input to the amplifier is the following.

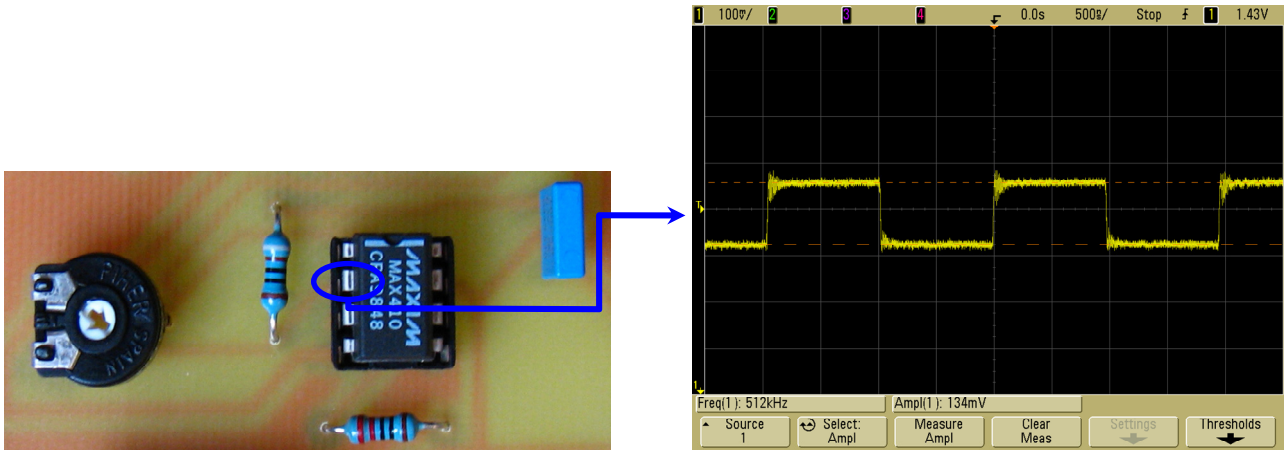


Figure 6.13: *Input to the amplifier*

The output of the amplifier, once set properly the trimmer[4], is the following.

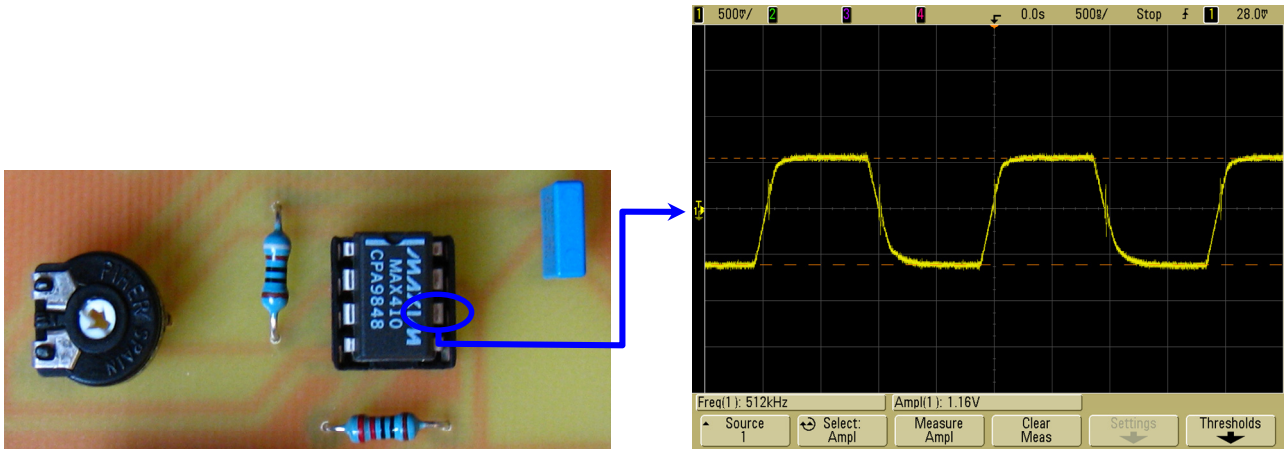


Figure 6.14: *Output of the amplifier*

How wanted, its output is greater than 1 V that is the minimum threshold at which the comparator can work.

The previous signal represents the input to the comparator. Anew, once adjusted the value of the trimmer, the output of the comparator is that displayed in the following figure. How manifest, its output is a TTL signal.

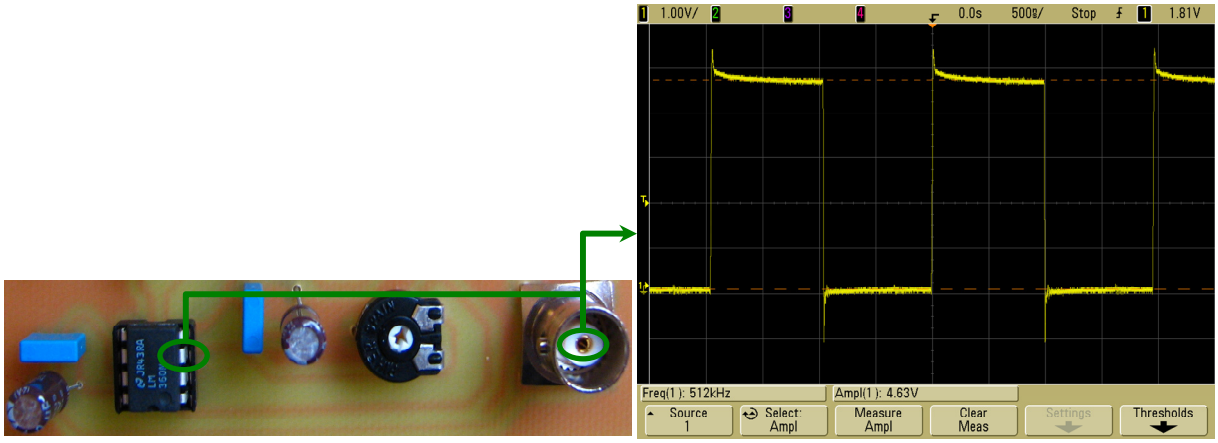


Figure 6.15: *Output of the comparator*

[4] in particular, without the amplifier goes in the non linear region

6.4. Optical terminal

In $(UWB)^2$ the preliminary handshake between the master and the slave presupposes implicitly the dual capacity of transmitting and of listening the channel. This means in particular that, at the physical layer, the generic terminal has to possess jointly the hardware necessary for receiving and emitting signals.

For what said, without taking into account for now the hardware relative to the EPLD, each optical terminal used in the simulation will be constituted by the cohesion of the hardware characteristics presented in the section 6.2 and 6.3. The relative schematic obtained is shown in the Figure 6.16. The relative PCBs are presented in the Annex I.

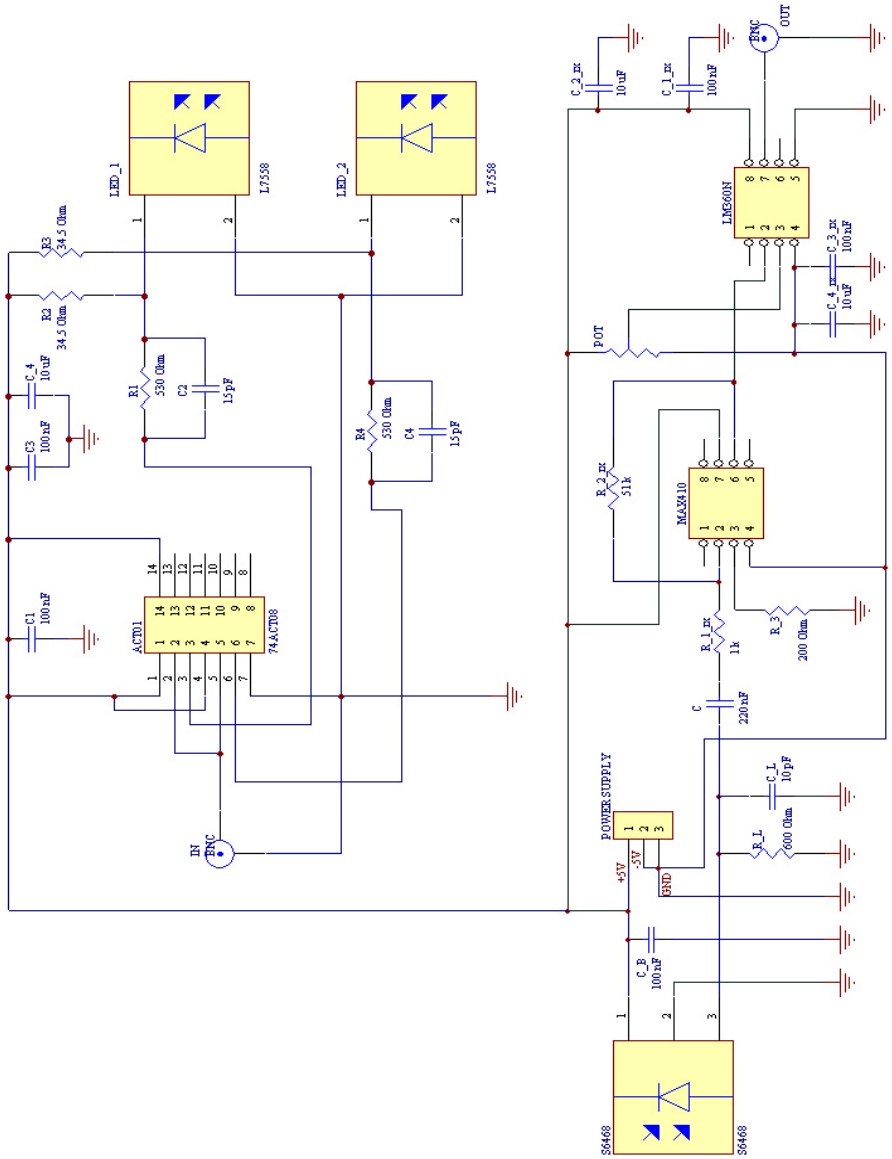


Figure 6.16: *Circuital scheme of the optical terminal*

The physical aspect of the terminal is provided in the Figure 6.17.

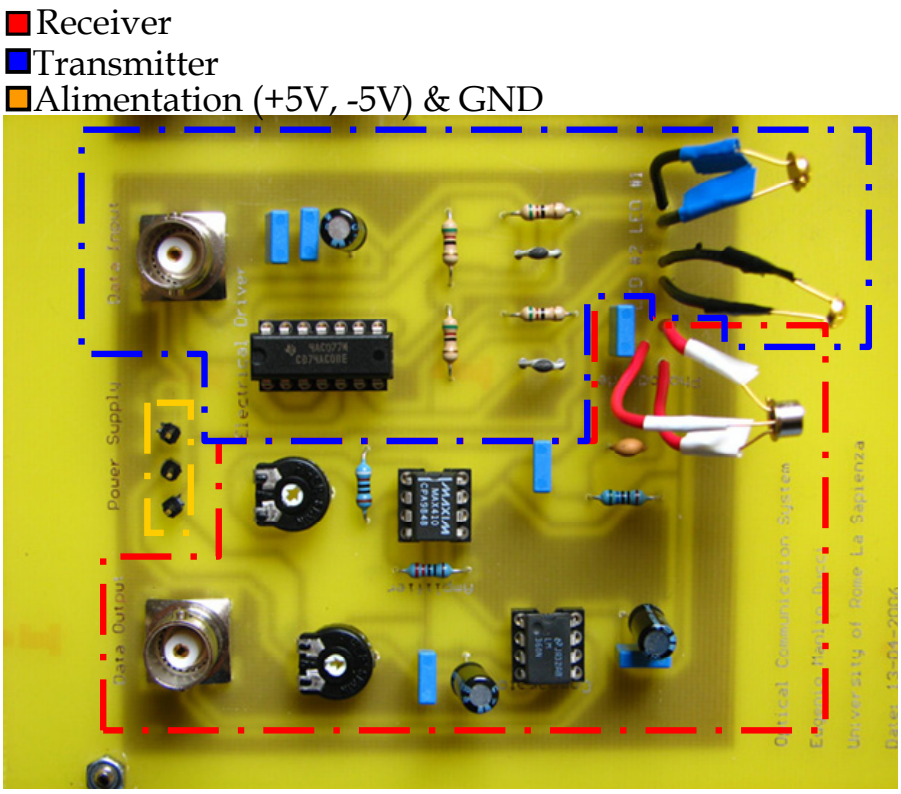


Figure 6.17: *Physical aspect of the optical terminal*

6.5. Optical system

As said above, at the physical layer each terminal is formed by the same hardware characteristics. In particular, the physical transmitters are constituted by LEDs that are intrinsically directives. Hence, as shown above in the Figure 6.9, in order to get the reciprocal interference necessary to test the protocol, an opportune positioning of the emitters is needed. Let us denote the role of each terminal as follows.

TERMINAL	ROLE
T1	Master #1
T2	Master #2
T3	Slave #1
T4	Slave #2

Table 6.1: *Roles of the terminals involved in the simulations*

The mutual dislocation employed is such that T3 and T4 are in visibility respectively with T1 and with T2; symmetrically, the terminals T1 and T2 are in visibility both with T3 and with T4. Effectively this is what represented in the Figure 6.9.

Finally, the following figure depicts the physical aspect of the optical system. Furthermore, the complete PCB's specifications are provided in the Annex I.

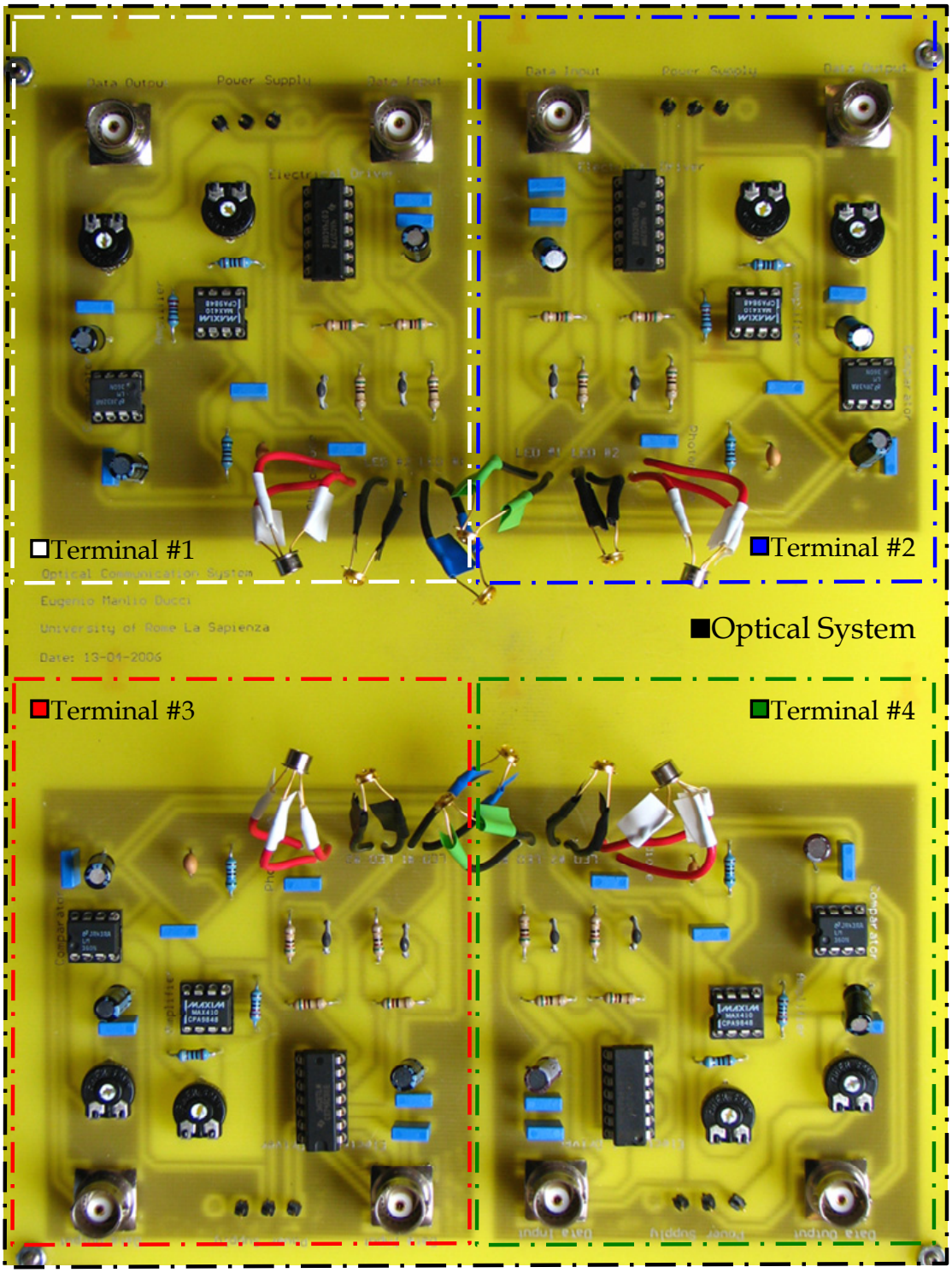


Figure 6.18: *Physical aspect of the optical system*

6.6. Masters, slaves and relative EPLDs

As stated in the section 6.3, the differential comparator LM360 is necessary to get a TTL signal that after will be passed to the EPLD. In fact, this is the nature of the signals that must be passed to the inputs of the EPLDs.

Even though the EPLD employed for the implementation of $(UWB)^2$, the Altera EPM9320ALC84-10, presents advanced features and a large amount of available resources, we had to simplify the logic of the protocol. To this end, we followed the next guidelines:

- to save and to economize the device's resources so that the protocol can fit to the EPLD
- to reduce the size of some field in order to: both to respect what above affirmed, and to manage in a simpler way the results of the simulations
- to remove the logical functionalities of some field that represents an option of implementation. In particular our choice was not to remove completely these fields from the packets, but rather to keep a single unused bit so that the VHDL code can be modified simply.

As remark, we can state that effectively, the changes introduced in the protocol implementation do not affect its original nature and its basic logic.

The changes and choices common to every packet defined by the protocol are the following:

- the fields "Tx Node ID" and "Rx Node ID" are reduced from 64 to 8 bits
- the "FEC/CRC" fields are not used. In any case, as said before, the fields have been kept and they get the length of $y = 1$ bit

The only modification interesting the LE packet is:

- the length of the "TH-Code" field has been chosen equal to 1 bit, rather than 16 bits. In fact, the PPM modulation is not used, and then no PPM code is needed

The changes and choices associated to the DATA packet are:

- as length of the DATA field, it has been chosen the value M = 256 bits. As we are going to show, this value determines directly the data rate chosen for the simulations
- the fields "PDU_{Number}" and "N_{PDU}" get both a length of 1 bit, instead of 8 bits. This choice is motivated by the fact that we realized a pseudo-casual generator whose function is to generate pseudo-aleatorily a request to send a single data packet, no more, and only when the previous transmission is finished. This means to keep a queue containing maximum one packet. The Pseudo Random Number Generator (PRNG) adopted is shown in the annex AI.18 and it has been implemented through a shift register having 13 taps. This allows obtaining a number generated following a uniform distribution and whose value belongs to the interval [0, 8191]. This number indicated the clock cycles that the master will have to wait before to send its DATA packet. Given the distribution employed, chosen the clock frequency ($f_{clk} = 512\,KHz$) and the numbers of data bits ($M = 256$) and known the expected value of the uniform distribution ($m_n^{(1)} = 4096$), the data rate is obtained as:

$$R_{data} = \frac{M}{m_n^{(1)} \cdot T_{clk}} = \frac{256}{4096 \cdot \frac{1}{512000}} = 32\,Kbps\,.$$

As regards the ACK packet, we did the following modifications:

- the "PDU_{Number}" field is constituted by 1 bit, rather than 8 bits
- the "DATA Status" field gets 1 bit, not 4 bits. In fact, in our implementation, this field keeps the only semantic of acknowledge positively or negatively the DATA packet

For the remaining fields, it has been chosen the original length and no further changes were made.

Moreover, and this time only because of "fitting reasons" into the EPLD, it was necessary to implement separately the logic of the master of the communication from that of the slave. This means that, whereas at the physical layer each terminal can transmit and receive packets, at the MAC layer a terminal is considered either a master or a slave of the communication. The corresponding roles were presented already in the Table 6.1.

Among the different fields defined by the protocol, some of those assumes a fixed value. How it is possible to verify directly through the codes presented in the Annex III, in our implementation we chose the following:

Node ID	*Master #1*: 35	*Master #2*: 2B	*Slave #1*: CA	*Slave #2*: C1
LE packet "Sync Trail"	A2			
LC packet "Sync Trail"	8E			
DATA packet "Sync Trail"	96			
ACK packet "Sync Trail"	ED			
Payload	96D496D496D496D4CACACACACACACACA 96D496D496D496D4CACACACACACACACA			
* all the values are expressed in hexadecimal				

Table 6.2: *Values chosen for some fixed fields defined by the (UWB)^2 protocol*

As said before, the EPLD chosen for the implementation of (UWB)^2 was the EPM9320ALC84-10. It is an EEPROM belonging to the Altera MAX 9000 family, the third generation of the Altera MAX architecture. It is fabricated with an advanced CMOS technology and between 84 pins, allows using totally 60 user pins. Actually, in our implementation, does not matter the total number of user pins, but the total resources of which the EPLD disposes. Regarding this aspect, the following table provides the main features of the EPLDs belonging to the family EPM9320.

Feature	EPM9320 EPM9320A
Usable gates	6,000
Flipflops	484
Macrocells	320
Logic array blocks (LABs)	20
Maximum user I/O pins	168

Table 6.3: *Main features of the EPLDs belonging to the family MAX 9320*

A graphical representation of how the physical pins, both of the masters and of the slaves, will be connected electronically is provided in the following figures.

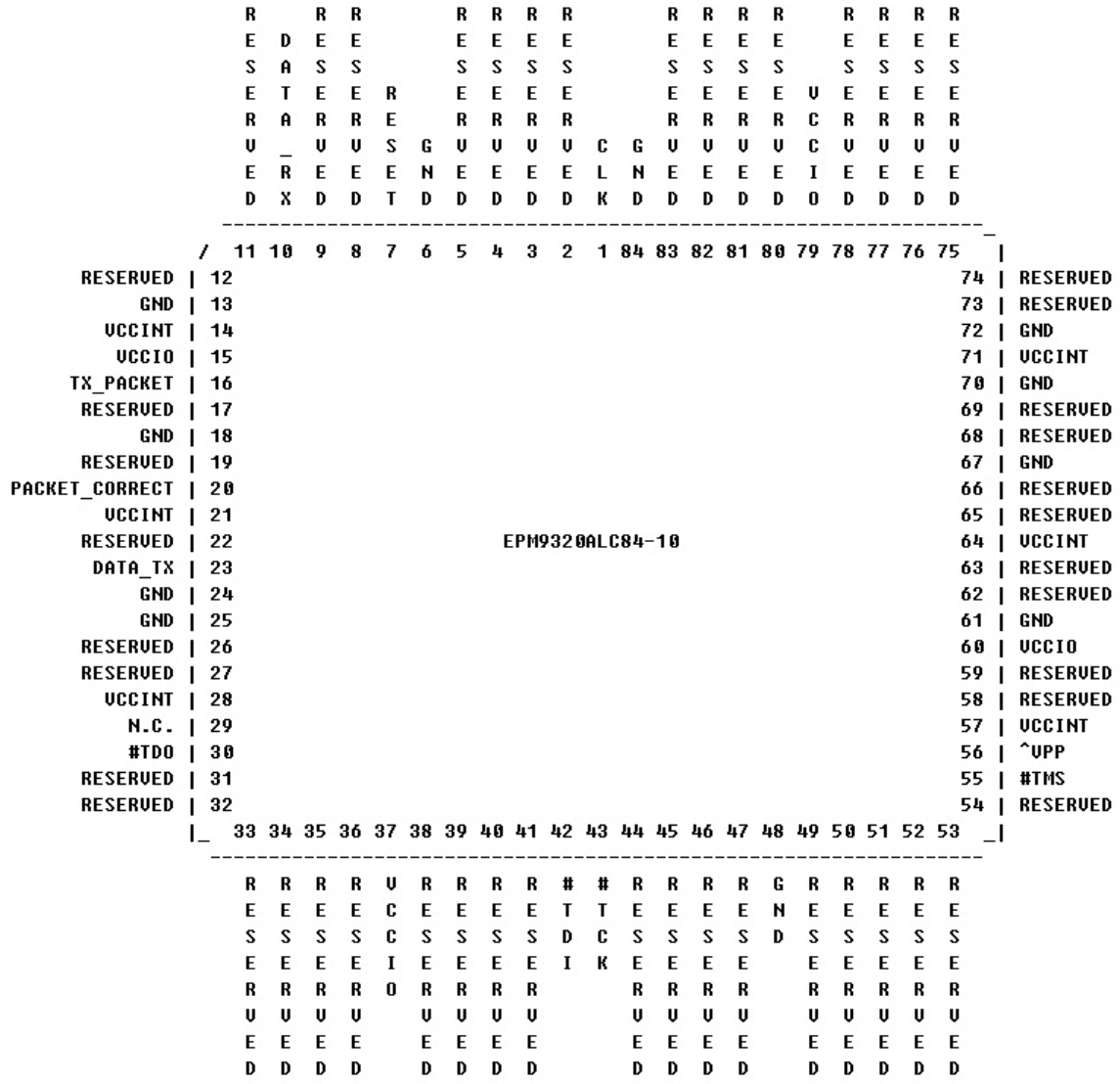


Figure 6.19: *Graphical representation of the connections of the pins for the masters*

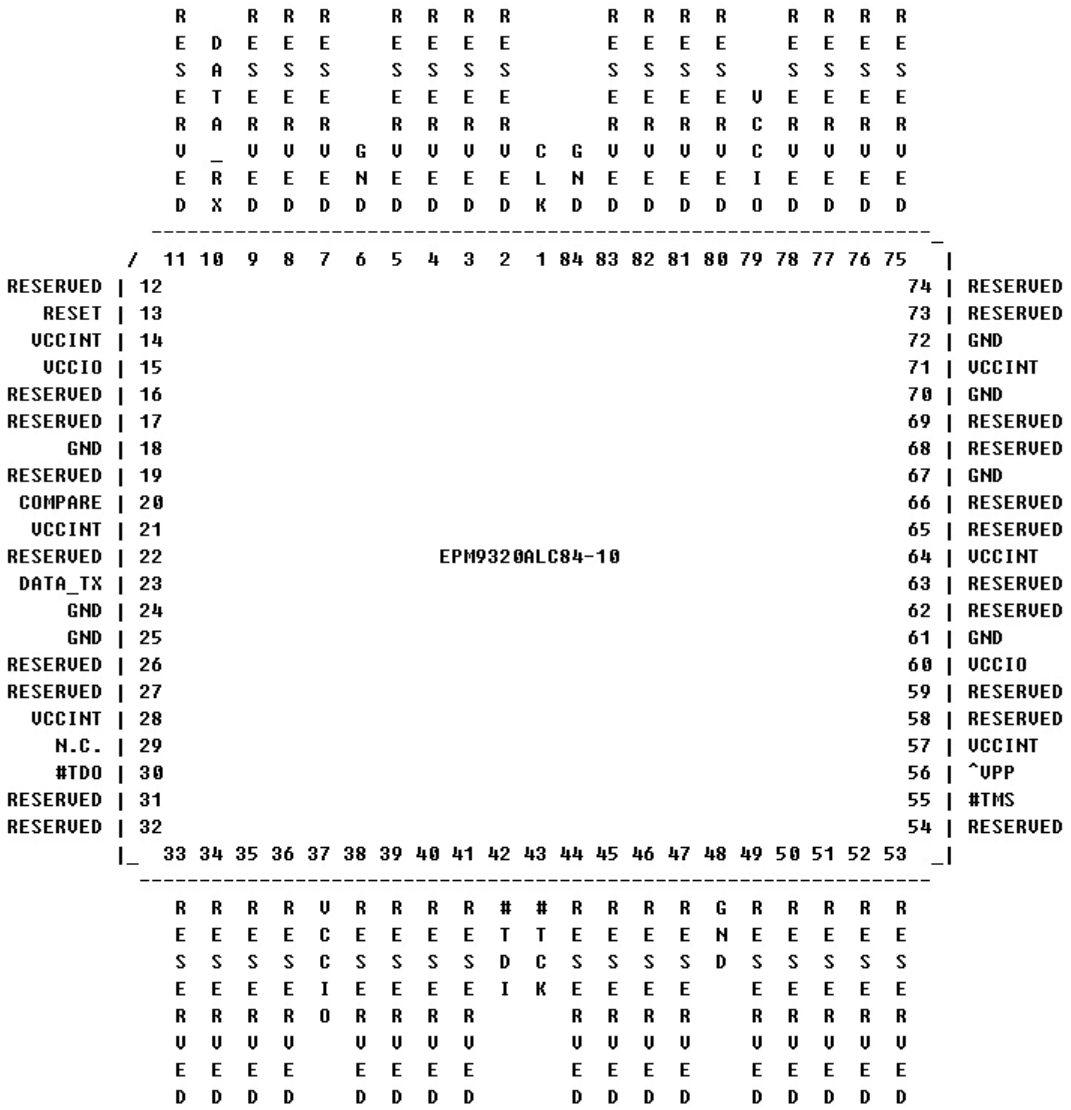


Figure 6.20: *Graphical representation of the connections of the pins for the slaves*

How it is possible to see directly in the two above figures, the user pins utilized for the masters and for the slaves are respectively those presented in the following tables.

PIN NUMBER	LABEL I/O PORT
1	CLK
7	RESET
10	DATA_RX
16	TX_PACKET
20	PACKET_CORRECT
23	DATA_TX

Table 6.4: *User pins employed in to the EPLDs of the masters*

PIN NUMBER	LABEL I/O PORT
1	CLK
10	DATA_RX
13	RESET
23	DATA_TX

Table 6.5: *User pins employed in to the EPLDs of the slaves*

Both the EPLD of the masters and that of the slaves gets the CLK, common to the four terminals and whose frequency is $f_{clk} = 512\,KHz$, in the pin number 1. For the clock it was employed the 5.0V TTL Clock Oscillator F1100E from FASTFOX, whose technical specifications are provided in the annex AII.6.

Similarly, both the masters and the slaves receive the pin number 10 and that number 23 respectively as input and output of the data. Thanks to the differential comparator LM360, the input DATA_RX is a TTL signal.

The masters and the slaves get them reset respectively in the pin number 7 and in that number 13. The optical system presents two reset buttons: the first one is assigned to the pair master #1 – slave #1, the second to that master #2 – slave #2.

Moreover the masters present two outputs: the first one is TX_PACKET, located at the pin number 16; the second, PACKET_CORRECT, located at the pin 20. The output TX_PACKET emits a rectangular waveform every time that a packet is transmitted. It will be used to measure the number of packets generated. The output PACKET_CORRECT emits a rectangular waveform when a packet is received correctly by the slave, that is to

say when the ACK field contains a bit equal to "1". It will be used to count the number of packets received correctly.

The remaining pins can be divided in two families: those listed in the annex AII.7, namely the dedicated device pins, and those not used because not required by our implementation. As regards the first family, we can remember that the dedicated device pins are necessary to connect the GND, the alimentation and the pins that allow the in-system programmability (ISP) by means of the JTAG – interface (Joint Test Action Group).

The next figures represent the schematics of the EPLDs related to the two terminals belonging to the pair of terminals number 1, namely the master #1 and the slave #1. They were realized using PROTEL 99 SE and clarify what above exposed.

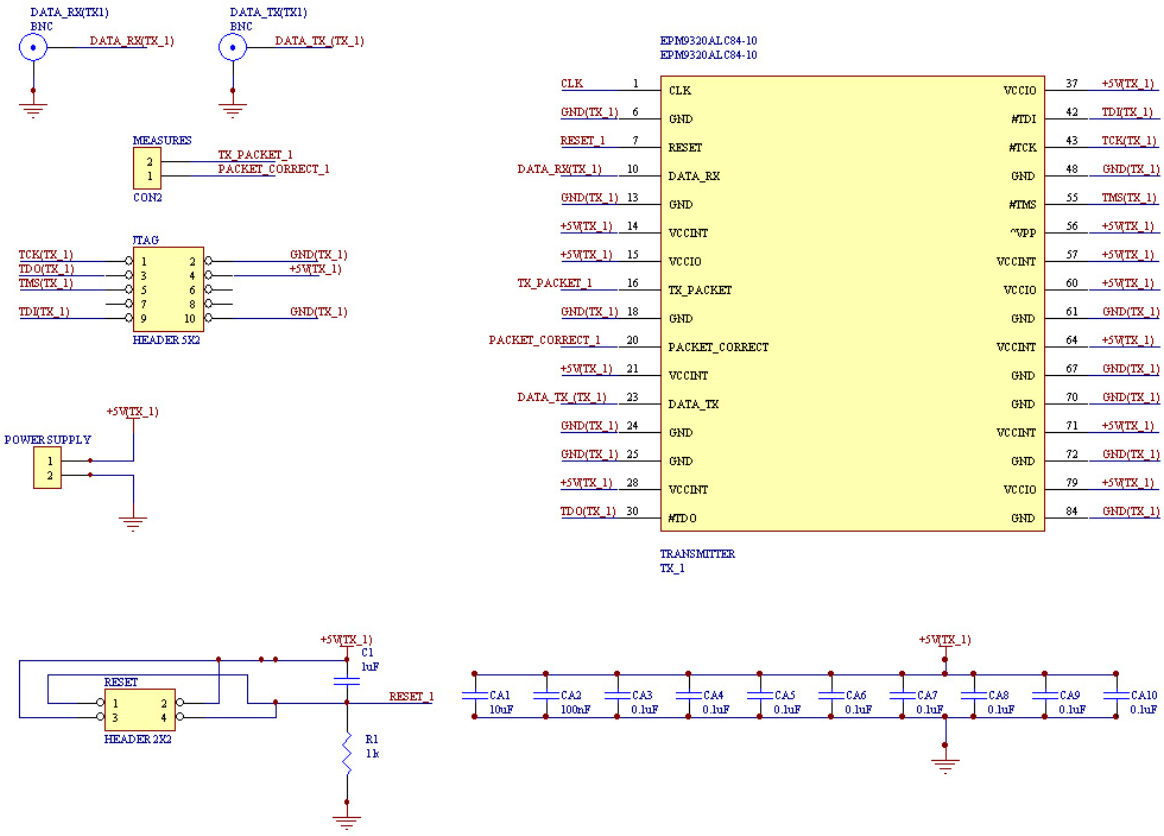


Figure 6.21: *Schematic of the EPLD related to the master #1*

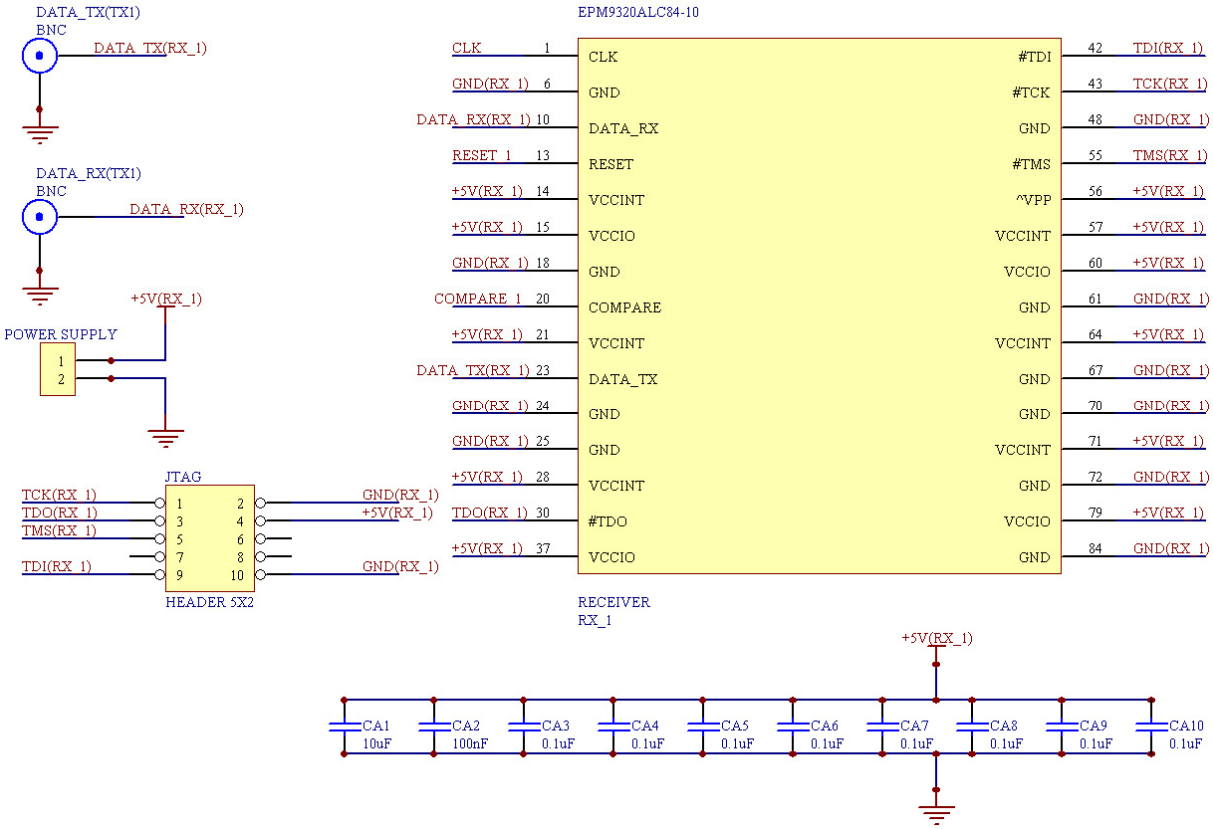


Figure 6.22: *Schematic of the EPLD related to the slave #1*

Finally, all the schematics and all the PCBs designed are shown in the Annex I, the component's specifications are provided in the Annex II and the VHDL codes are presented in the Annex III.

Lastly, we wish to point out that the main board with the EPLDs has been only designed because of the very high cost of its physically realization (more than 1600 €) resulting mainly by the absence in the stock of the EPLD chose for implementing the protocol, the EPM9320ALC84-10, whose price, a bit higher than 100 €, would have still allowed building the whole system. This lack leaded to chose, at least as potential change, the EPLD EPM9320ALC84-15 that, even if worse, has a price exceeding 300 €.

Chapter 7

Implementation of $(UWB)^2$ over an Optical System: Results

Contents

7.1 Introduction 118
7.2 (UWB)2: software implementation 118
7.3 Optical System: hardware implementation 134

7.1. Introduction

In this chapter we are going to expose the results obtained through the simulations of the $(UWB)^2$ protocol.

First of all, we will present the results achieved by means of the Max+Plus simulator. Regarding this point, several simulations have been realized. The aim of each simulation was to test the protocol working at different data rates, obtaining consequently both the total number of data packets transmitted by the masters, and the total number of packets correctly received by the slaves. In this way we could estimate the throughput of the net as:

$$Throughput = \frac{N_{CR}}{N_T} = \frac{\sum_{i=1}^{2} N_{CR}^{i}}{\sum_{i=1}^{2} N_T^{i}},$$

where N_{CR} is the total number of data packets correctly received by the slaves (namely $N_{CR}^1 + N_{CR}^2$) and N_T the total number of data packets transmitted by the masters (that is to say $N_T^1 + N_T^2$). Evidently, the above throughput is dependent from the data rate employed.

7.2. $(UWB)^2$: software implementation

As said previously, the $(UWB)^2$ protocol has been tested working at different low data rates. Those chosen for the simulations are: [256 Kbps, 128 Kbps, 64 Kbps, 32 Kbps, 16 Kbps, 8 Kbps, 4 Kbps]. At the physical layer the changing of the data rate has been achieved modifying in the masters architecture the length of the shift register used for implementing the PRNG. In particular, given that the PRNG generates pseudo-casually a number following a uniform distribution, we can state that for each PRNG employed, whose length is *N*, the data rate can be calculated as:

$$R_{data} = \frac{M}{m_n^{(1)} \cdot T_{clk}} = \frac{256}{\frac{2^N}{2} \cdot \frac{1}{512000}} bps = \frac{2^{18}}{2^N} \cdot 10^3 \, bps = 2^{18-N} \, Kbps$$

where *N* assumes the values [10, 11, 12, 13, 14, 15, 16]. To give a graphical example, the shift register with 13 taps is represented in the annex AI.18.

A remark to do now is that in our nomenclature we are distinguishing between data rate and bit rate. The data rate is what we have calculated using the above formula, that is to say the amount of information that a terminal wants to transmit in the time. With bit rate we are meaning the rate at which the above information is transferred into the channel, namely the clock rate that has been fixed to 512 KHz.

The throughputs obtained executing the simulations are presented in the Table 7.1.

As foreseeable, the throughput increases when the data rate decreases. This result can be explained thinking that when the data rate reduces, less packets are transmitted in a given interval time, and therefore the probability of collision between two packets diminishes as well, leading to an upper value of the throughput.

The table contains the Simulation Time Interval (STI) as well, namely the total duration time in which the system has been simulated. Given that a lower data rate means a smaller amount of DATA packets transmitted, it was necessary to increase the STI in order to obtain a sufficient number of exchanges of packets.

To end, further it has been calculated what we have called "partial throughput", namely the ratio:

$$Throughput = \frac{N_{CR}^{(i)}}{N_{T}^{(i)}},$$

where $N_{CR}^{(i)}$ is the number of data packets correctly received by the slave number *i* and $N_{T}^{(i)}$ is the total number of data packets transmitted by the master number *i*. This kind of measure has been performed only to ascertain that really the protocol assures a good degree of fairness in the medium access. Effectively examining the values in the table, it provides equality of obtaining the medium access.

The Figure 7.1 shows in a graphic the values of throughputs obtained vs. the data rates employed.

DATA RATE (Kbps)	*NET THROUGHPUT*	*PARTIAL THROUGHPUT MASTER #1- SLAVE #1*	*PARTIAL THROUGHPUT MASTER #2- SLAVE #2*	*SIMULATION TIME INTERVAL (STI)*	*DATA PACKETS TRANSMITTED BY THE MASTER #1 DURING THE STI*	*DATA PACKETS TRANSMITTED BY THE MASTER #2 DURING THE STI*	*DATA PACKETS TRANSMITTED IN THE NET DURING THE STI*	*DATA PACKETS CORRECTLY RECEIVED BY THE SLAVE #1 DURING THE STI*	*DATA PACKETS CORRECTLY RECEIVED BY THE SLAVE #2 DURING THE STI*	*DATA PACKETS CORRECTLY RECEIVED IN THE NET DURING THE STI*
256	0.24	0.273	0.2	1 sec	912	770	1682	249	155	404
128	0.52	0.52	0.52	1 sec	374	381	755	194	199	393
64	0.686	0.685	0.688	2 sec	416	423	839	285	291	576
32	0.86	0.85	0.86	4sec	430	458	888	367	395	762
16	0.92	0.92	0.92	6 sec	326	335	661	300	309	609
8	0.97	0.97	0.97	9 sec	277	259	536	268	250	518
4	0.997	0.997	0.997	20 sec	395	397	792	394	396	790

Table 7.1: *Throughputs obtained by the software simulation of the $(UWB)^2$ protocol*

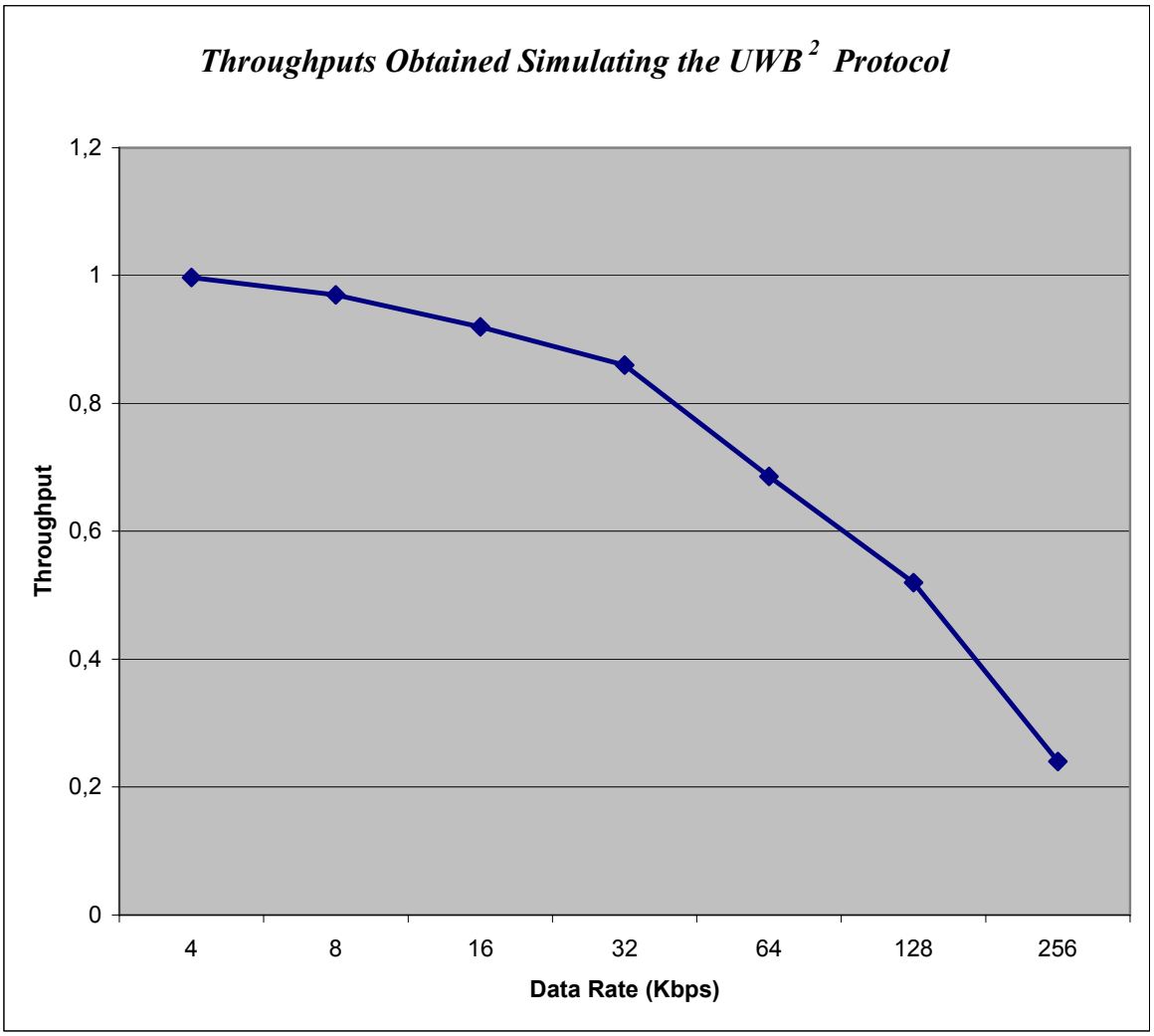


Figure 7.1: *Throughputs Obtained Simulating the UWB^2 Protocol*

Additionally, thanks to the graphical representation of the simulations provided by the Max+Plus simulator, it is really interesting to analyze the different situations that can occur during the development of the communication. Referring to the case in which the masters are transmitting with a data rate of 32 Kbps, in the following of this paragraph we are going to show some of those.

First of all, we are going to provide the graphical representation of the packets defined by the (UWB)^2 protocol: the LE packet, the LC packet, the DATA packet and the ACK packet.

Following this order, the next figures represent these packets, both for the first and for the second pair of terminals. These pictures allow getting a clearer visualization of each field contained in the above quoted packets. More over, we remember that their structures were provided respectively in the Figure 5.26, Figure 5.29, Figure 5.27 and Figure 5.30.

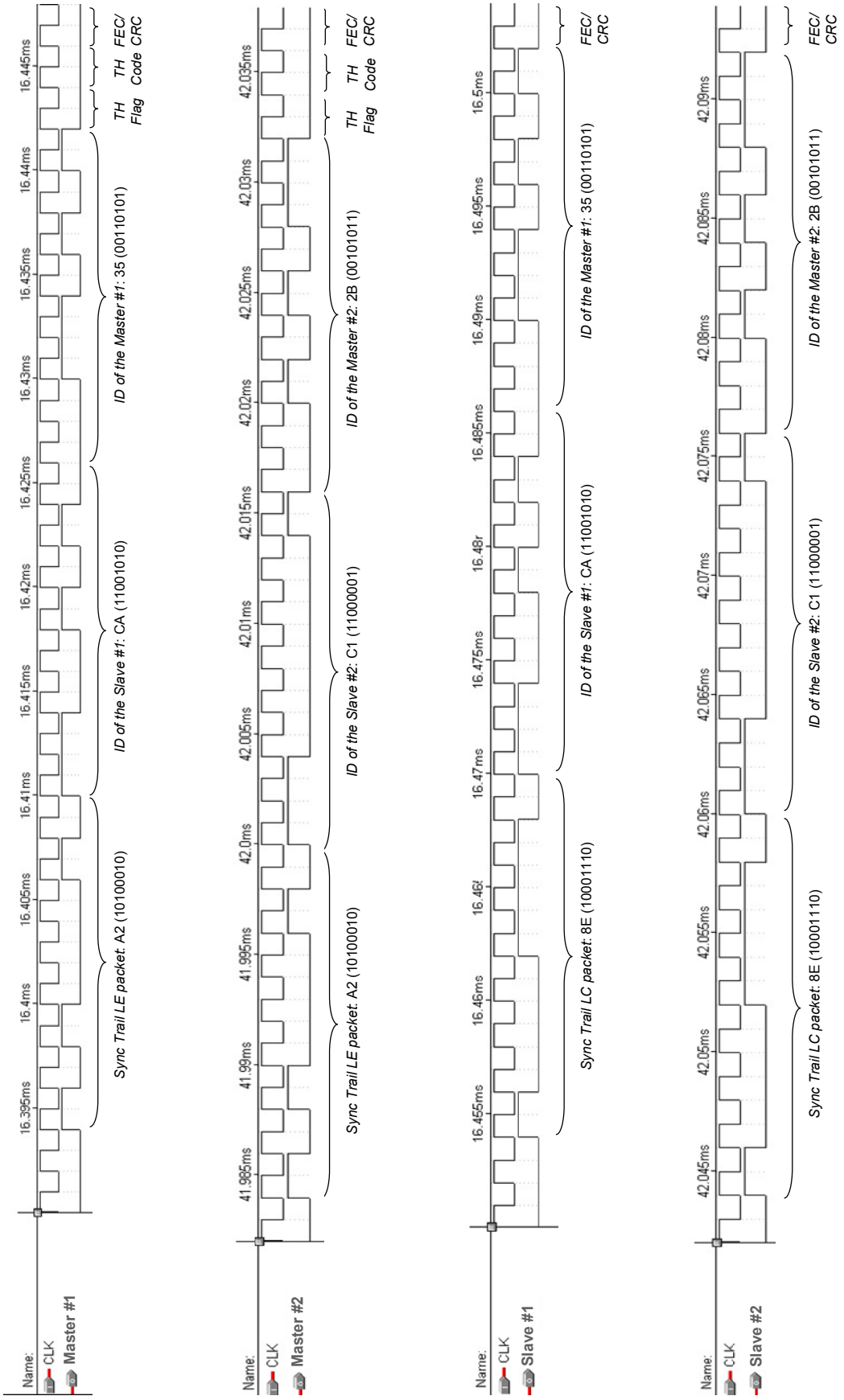


Figure 7.2: *LE and LC packets relative to both the pairs of terminals and obtained through the Max+Plus simulator*

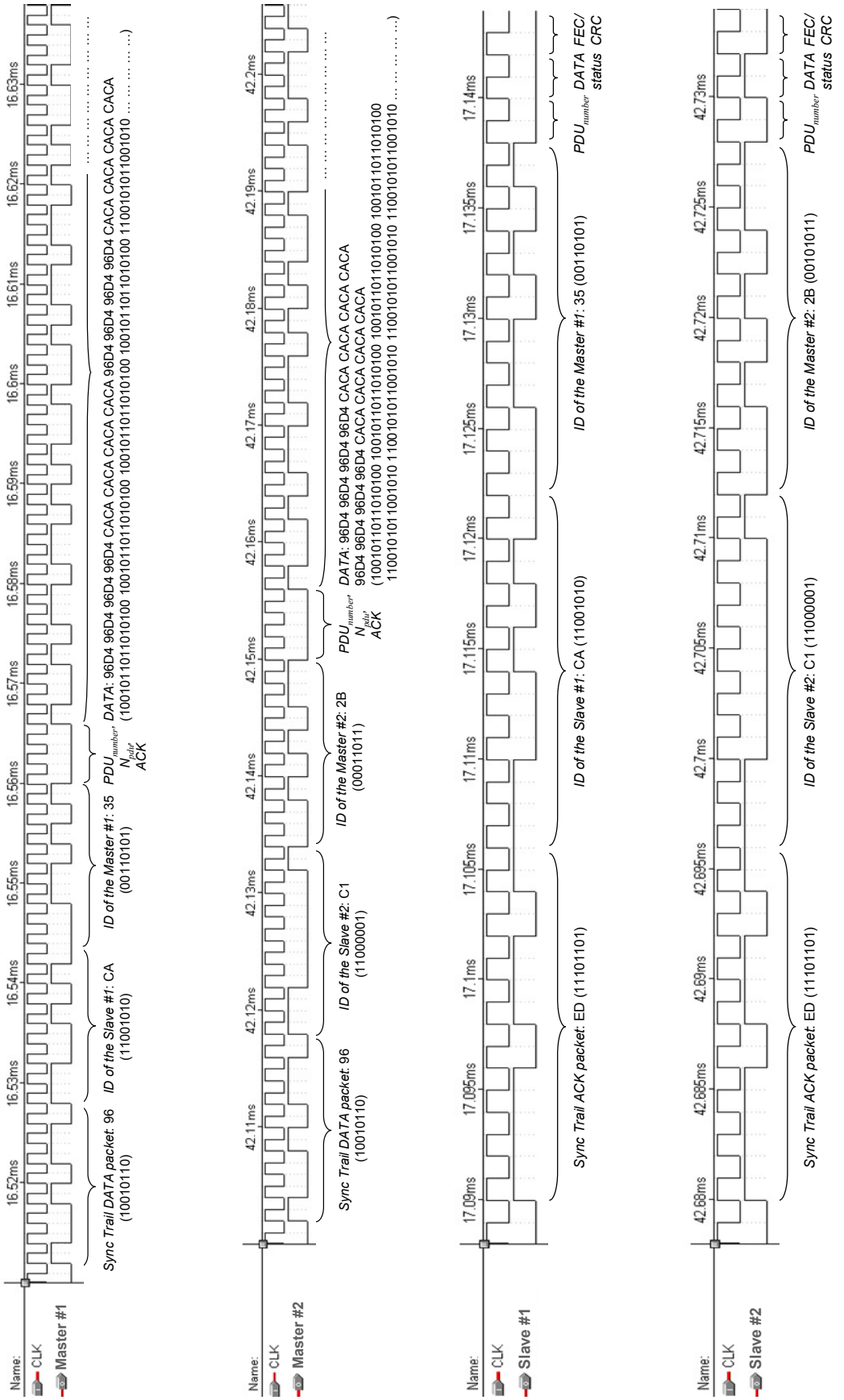


Figure 7.3: *DATA and ACK packets relative to both the pairs of terminals*

Now, let us consider the whole communication of a data packet between, for example, the second pair of terminals: the master #2 and the slave #2. As known, in absence of errors the whole communication chain is constituted by the following packets: LE packet, LC packet, DATA packet and ACK packet[5], in the chronological order of writing. The Figure 7.4 provides a graphical example of the above quoted exchange of packets. The master #1 does not transmit any packet during this time, whereas the master #2 sends a LE packet to the slave #2 in order to establish a connection. Subsequently, the LE packet is received both by the slave #1 and by the slave #2. Decoding the packet, the slave #1 becomes aware, through the Rx Node ID field, that the communication is not directed to him and then it follows to hear the channel. Vice versa, the slave #2 recognizes its ID and replies to it with a LC packet. Received and decoded this packet, the master #2 sends the DATA packet. Once more, the slave #1 receives the DATA packet too, passing over it as before. On the contrary, the slave #2 decodes the packet and, by means of the comparison of the DATA field received with that stored in its memory, becomes aware if errors are occurred. If so, it increases of one unity the variable "ERRORS_2", otherwise the variable increased is that measuring the total number of packets correctly received and decoded, namely "PACKET_CORRECT2[0..9]". In the case that we are analyzing no error is occurred. Hence, the slave #2 replies with an ACK packet, acknowledging positively the DATA packet.

A further comment to do regards the presence of the counter "PACKET_TX2". This variable, necessary for calculating the final throughput, is increased every time that the master #2 wants to send a DATA packet to the slave #2, that is to say when it transmits a LE packet.

As final observation, we point out that effectively what said until now is applicable symmetrically to the couple of terminals master #1 - slave#1.

The Figure 7.5 depicts again another situation in which no error occurs. The first couple of terminals realizes its communication with success and immediately after, the second pair performs its communication, without this last transmission interferes with the previous. The masters increase their "PACKET_TX" variables just before to transmit the LE packet and they augment them "PACKET_CORRECT" variables of one unity when the DATA packets are received correctly by the correspondent slaves (ACK bit set to "1").

[5] The reader should remember that we chose to use the ACK packet by default

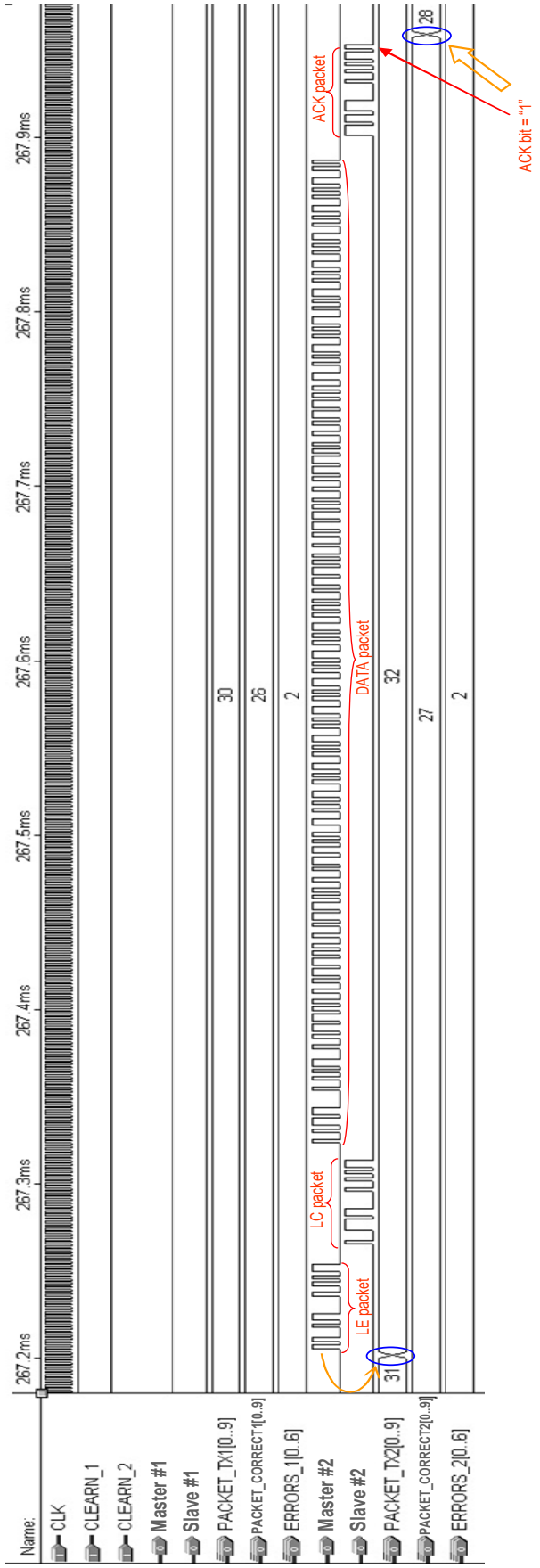


Figure 7.4: *Example of exchange of the packets defined by the (UWB)^2 protocol for the couple of terminal master #2 - slave #2*

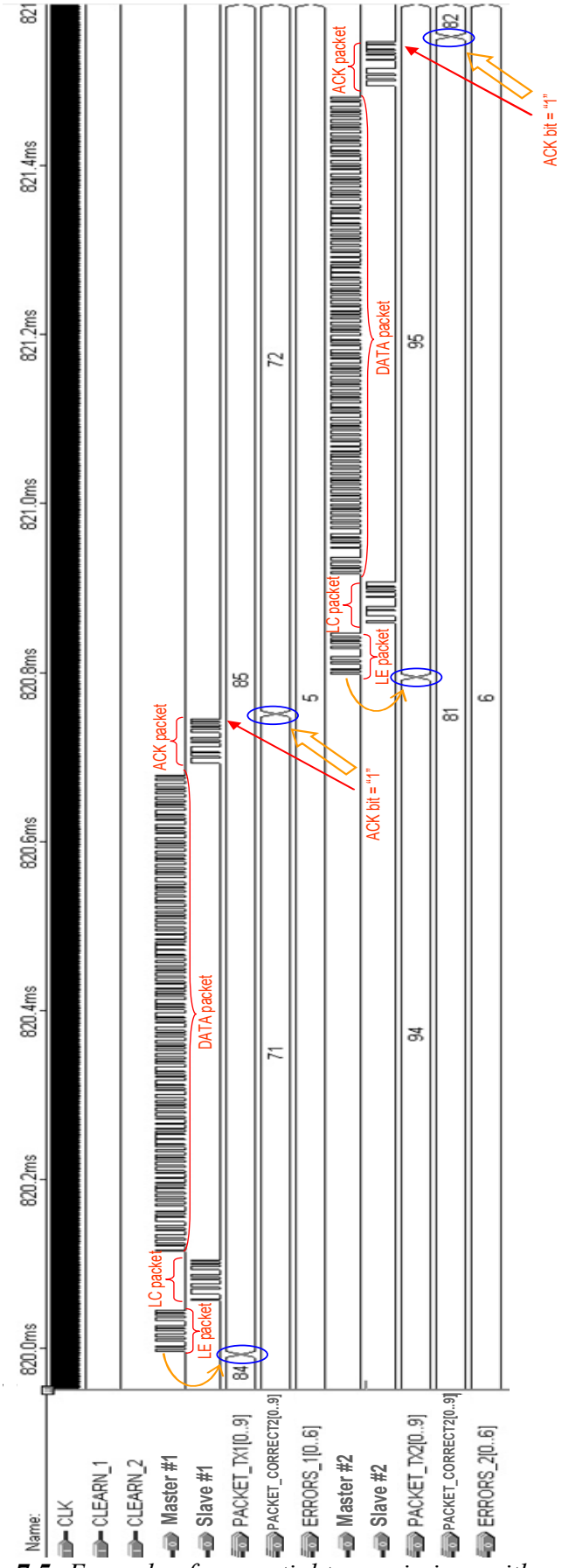


Figure 7.5: *Example of sequential transmissions without errors*

During the communication, many kinds of interferences can be detected, in the sense that the interfering packets can fall in distinct parts of the useful packets. The Figure 7.6 represents a situation in which the LE packet emitted by the master #2 interferes with the DATA field of the DATA packet sent by the master #1. After the LE – LC exchange between the first pair of terminals, the master #1 sends the DATA packet. While occurs this transmission, the master #2 sends its LE packet to the slave #2 and this generates a collision in the DATA packet, in particular just in the DATA field. After few bits, the slave #1 detects the errors and increases the errors counter, namely the variable "ERRORS_1". Obviously, the relative ACK packet, sent by the slave #1, will contain an ACK bit equal to "0". Furthermore, the whole LE packet transmitted by the master #2 has been affected by the collision as well, meaning with this that the slave #2 will not be able to decode correctly the LE packet and therefore it will not send the LC packet to the master #2.

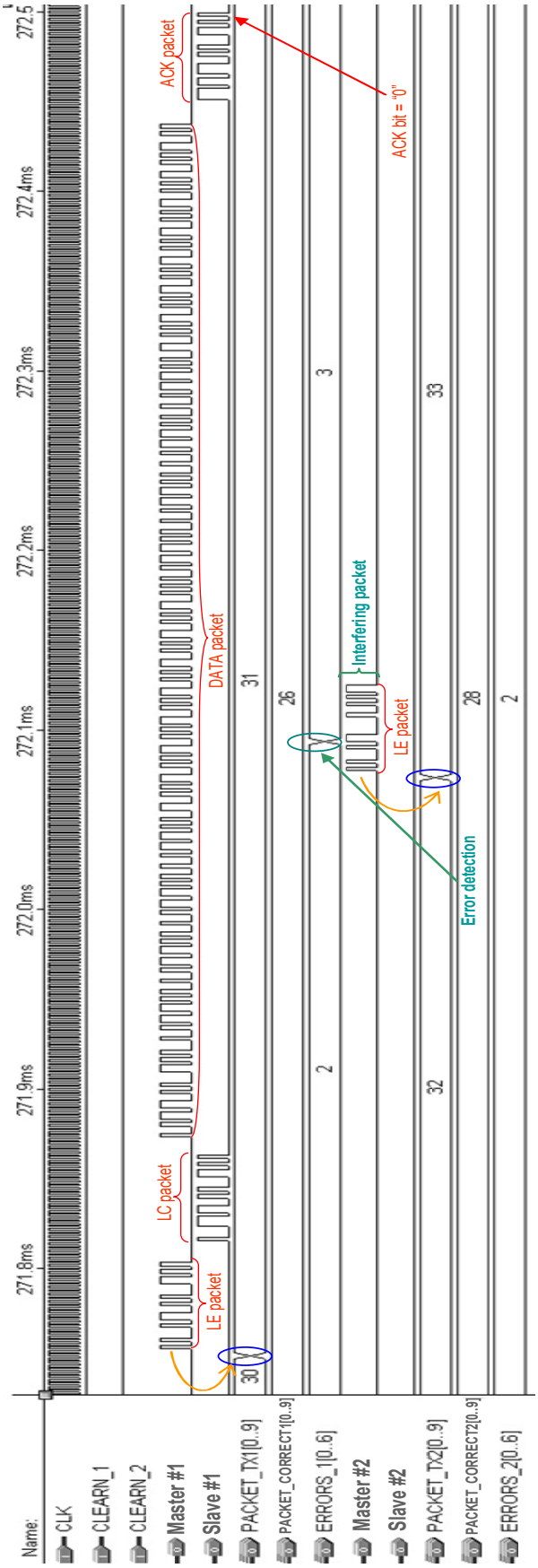


Figure 7.6: *A first example of packet collision between a DATA packet and a LE packet*

In the previous example the collision occurred in way that the slave #2 still could able to decode the header of the DATA packet. This means that it knew that a DATA packet was received, but affected by errors. Vice versa, the Figure 7.7 shows a situation in which the collision interests jointly the header of the DATA packet sent by the master #1 and that of the LE packet transmitted by the master #2. This means that not only the slave #2 will not be able to decode correctly the LE packet (and thus to establish the link), but additionally that the slave #1 will not be able to realize to have received a DATA packet and then it will not acknowledge the DATA packet, neither positively nor negatively.

Immediately after the end of the DATA packet the master #2 tries to establish the link for sending a DATA packet to the slave #2 and, because of the absence of interference, the whole communication gets a positive outcome.

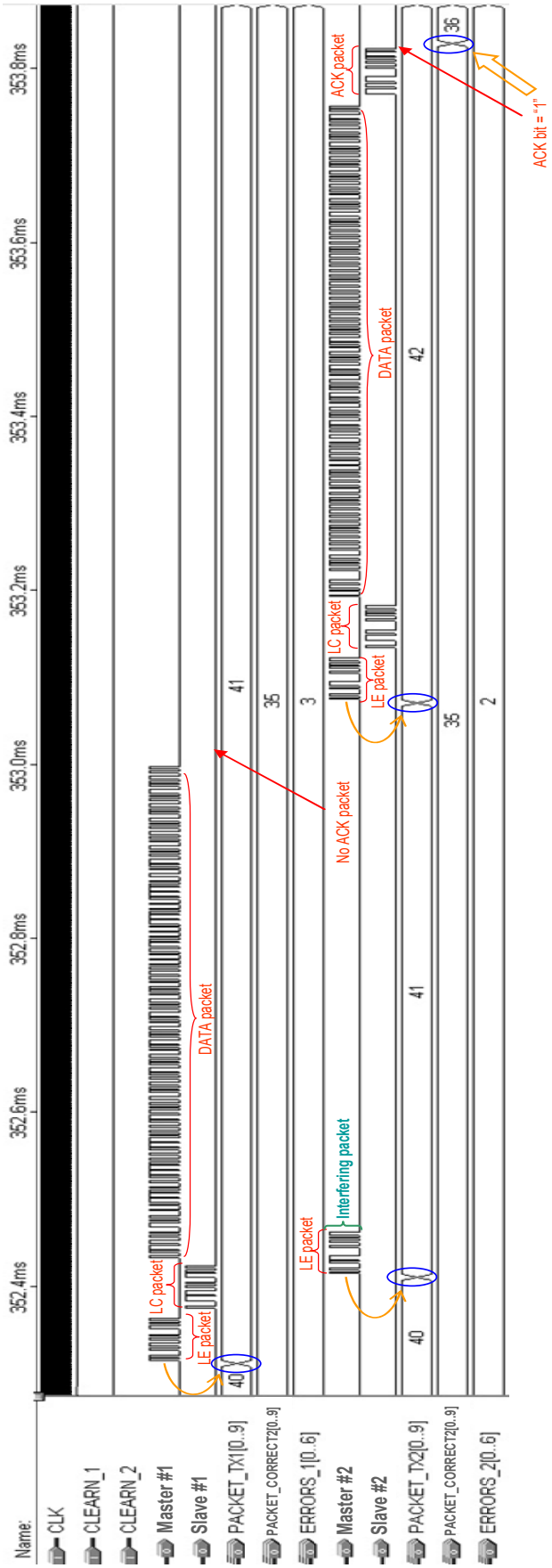


Figure 7.7: *Another example of packet collision between a DATA packet and a LE packet*

A very interesting situation is that pictured in figure 7.8. Both the masters want to transmit a DATA packet but the first to send a LE packet is that number 2. Obviously, this packet is not affected by interference. Subsequently and almost simultaneously, the master #1 sends its LE packet to the slave #1 and the slave #2 replies with a LC packet to the master #2. Once again, these packets do not interfere between themselves. In fact, with reference to the Figure 6.9, in the case in analysis any collision can occur neither at the receiver of the slave #1 (that can hear packets of the type LE and DATA coming from any master), nor at the receiver of the master #2 (that can hear packets of the type LC and ACK coming from any slave). This means that both the slave #1 and the master #2 will succeed in decoding the packets to them addressed.

After, and still almost simultaneously, the slave #1 replies to the master #1 with a LC packet and the master #2 starts to transmit its DATA packet. Anew and for the same reasons above exposed, no collision occurs between this LC packet and the first part of the DATA packet sent by the master #2.

Hence, for what said until now and with reference to the figure 7.8, the part do not affected by collision is that contained in the blue box.

Approximately after the instant 706.3 ms the collision between the DATA packets takes place, but with different effects for the slaves. While the initial part of the DATA packet transmitted by master #2 and decoded by the slave #2 is sufficient to understand the nature of the current packet (namely that it is a DATA packet coming from the master #2 and directed to the slave #2), the whole header of the DATA packet sent by the master #1 is affected by the collision, not allowing to recognize therefore the DATA packet. What said leads to two consequences: the slave #2 detects with errors the DATA packet addressed to itself and then acknowledges it with an ACK packet with the ACK bit set to "0"; the slave #1 does not recognize the DATA packet directed to itself and therefore does not send the ACK packet.

Lastly, the Figure 7.9 depicts a set of sequential exchanges of packets, some affected by collisions, others not.

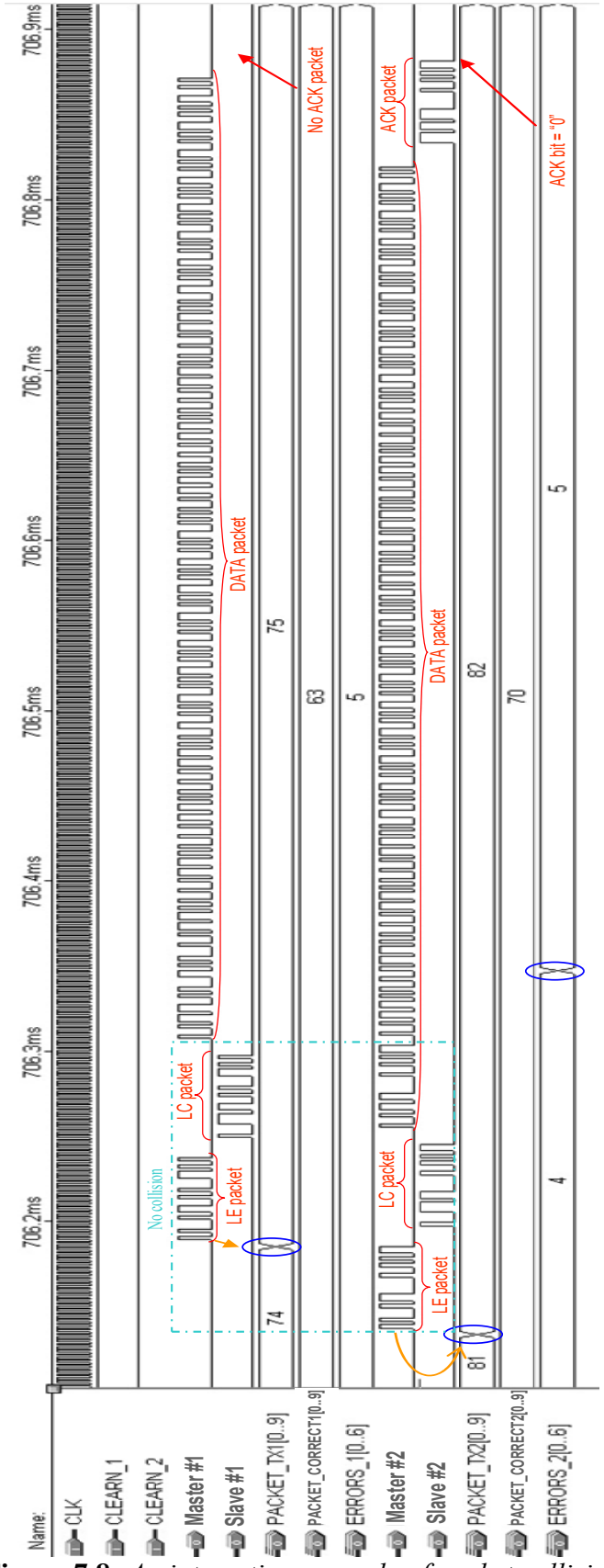


Figure 7.8: *An interesting example of packet collision*

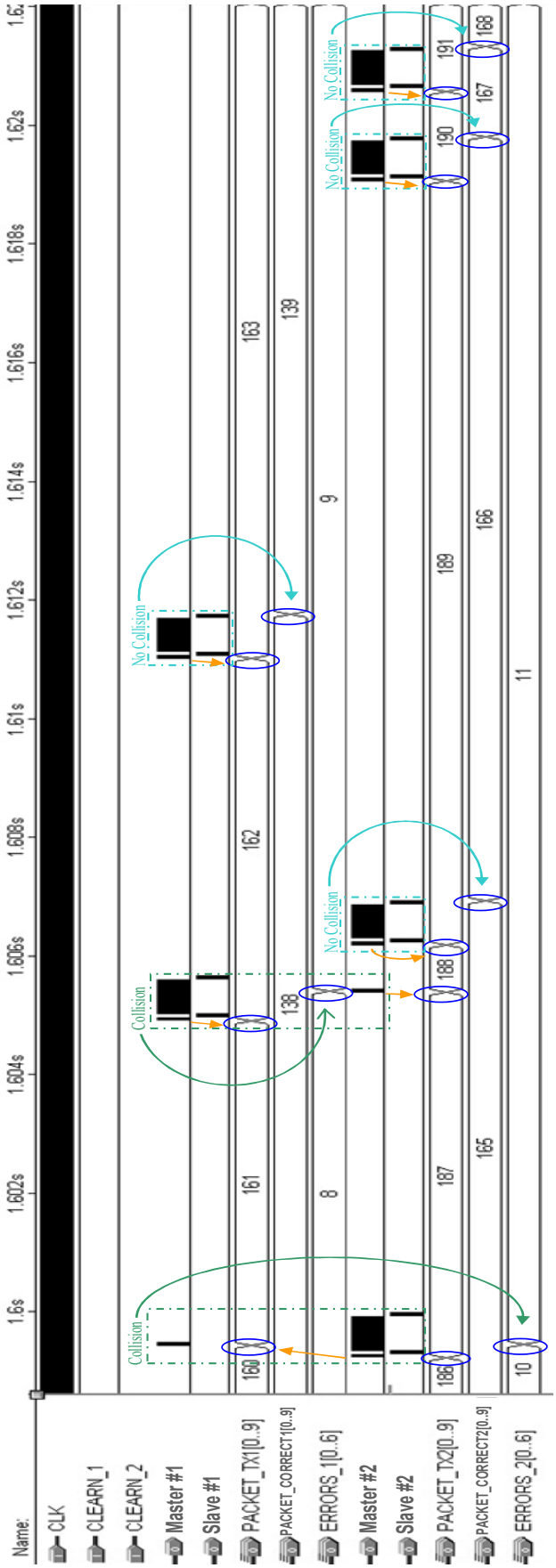


Figure 7.9: *An example of a set of sequential transmissions and receptions*

7.3. Optical System: hardware implementation

As seen in the Chapter 6, the optical system is constituted by four terminals, divided in two groups: master #1 – slave #1 and master #2 – slave #2 and each communication of a couple of terminals interferes with the other. To check the functioning of the system we chose to use a function generator for each pair of terminals. The following figures show the operative context in which the system was checked.

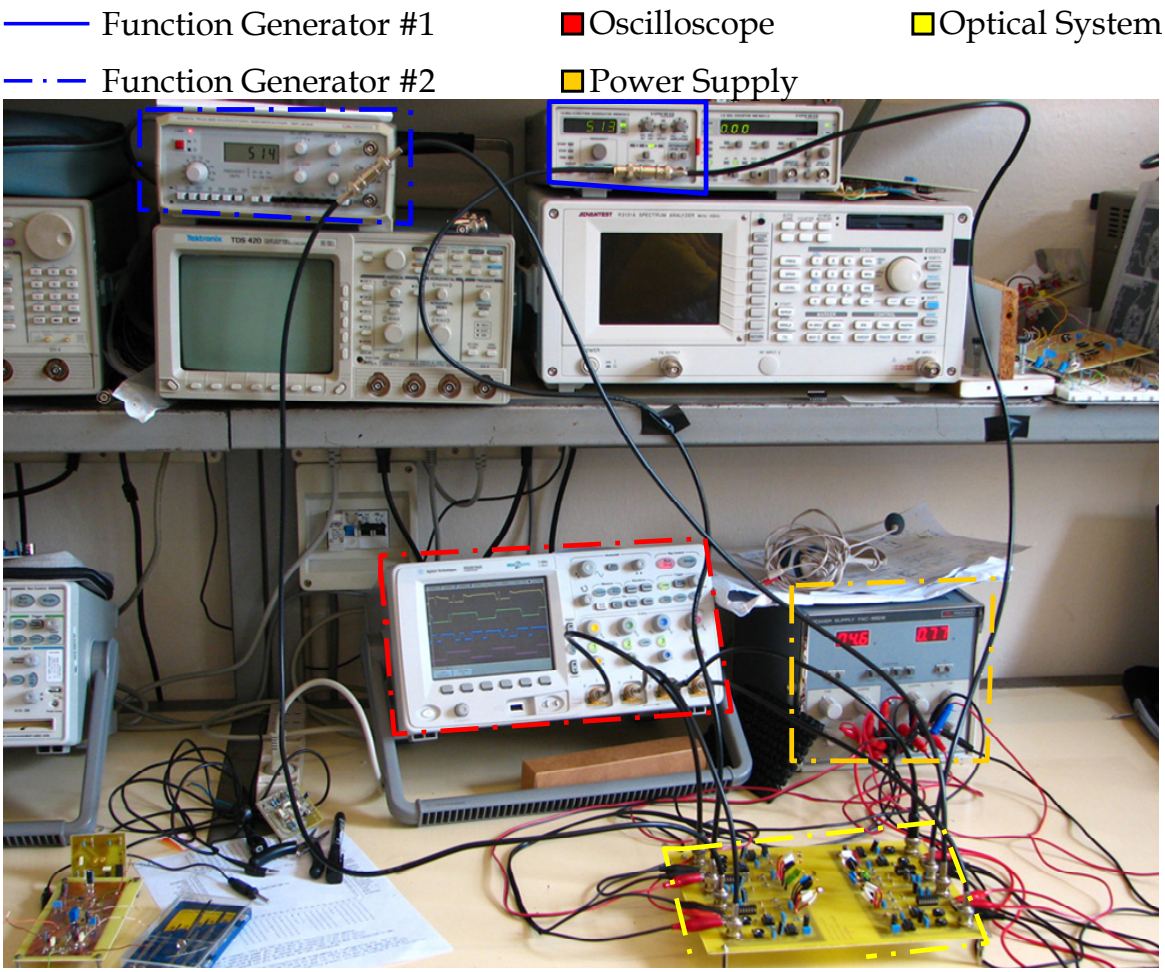


Figure 7.10: *Operative context of work in the laboratory*

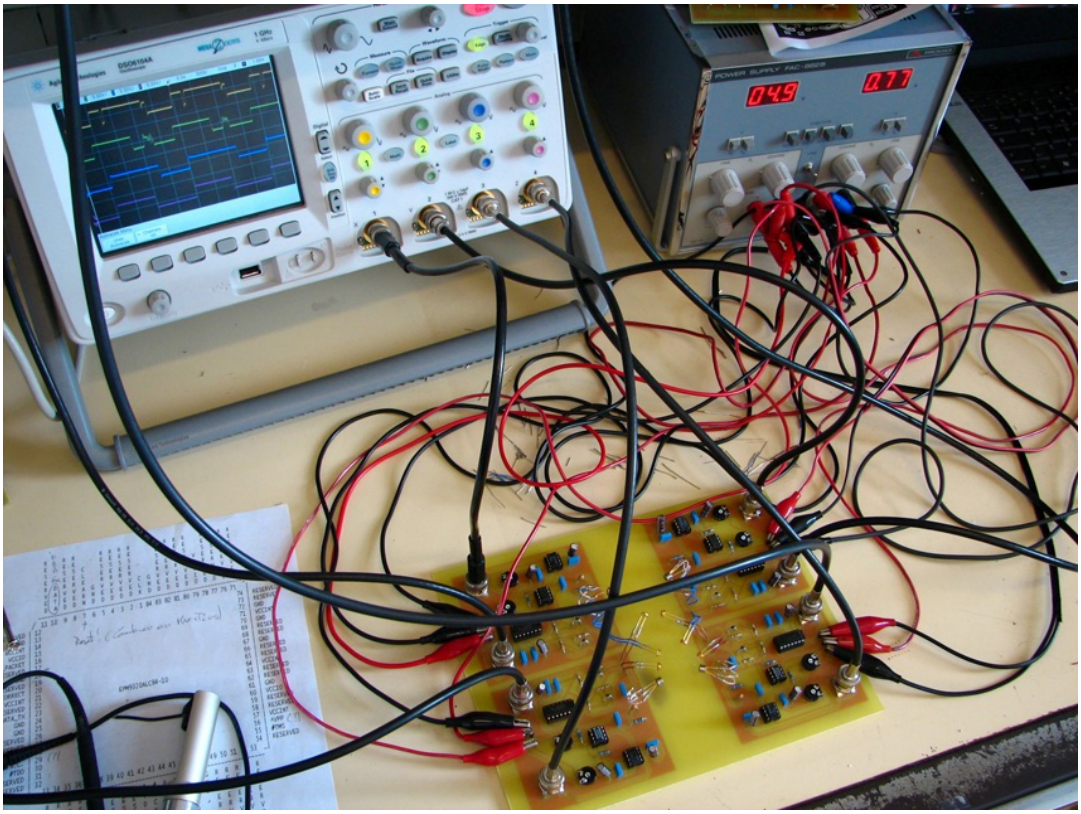


Figure 7.11: *A visual particularity of the Figure 7.10*

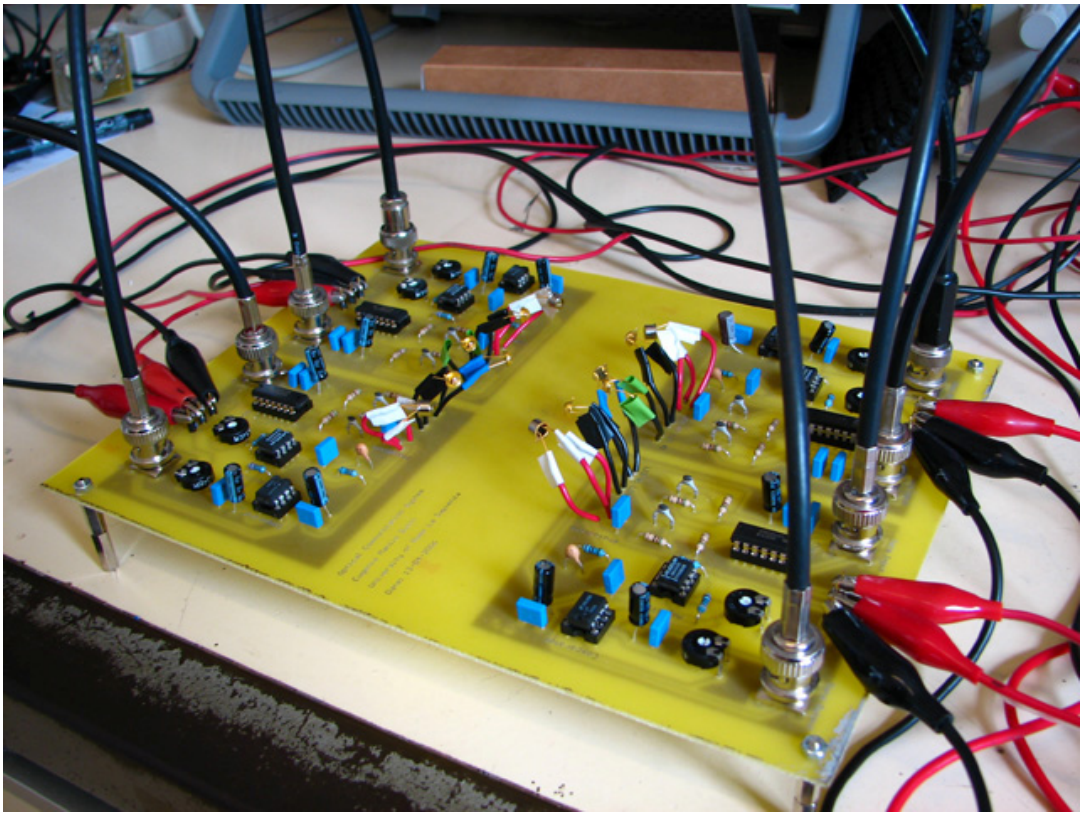


Figure 7.12: *An other visual particularity of the Figure 7.10*

In the following of this paragraph we are going to display some pictures representing the four signals implicated in the communication between the terminals. The pictures was taken out through the digital oscilloscope DS06104A of Agilent Technologies. Additionally, we decided to switch off the second LED of the terminal #1 with the purpose to not generate interference in the terminal #4, in way to take this terminal as reference for the others terminals[6].

To start, the following figure represents a situation of absence of interference. How it is clear, every terminal gets an output of kind TTL, that is to say a signal whose value is 0V or +5V. Moreover, the working frequency is, as wanted, of 512 KHz.

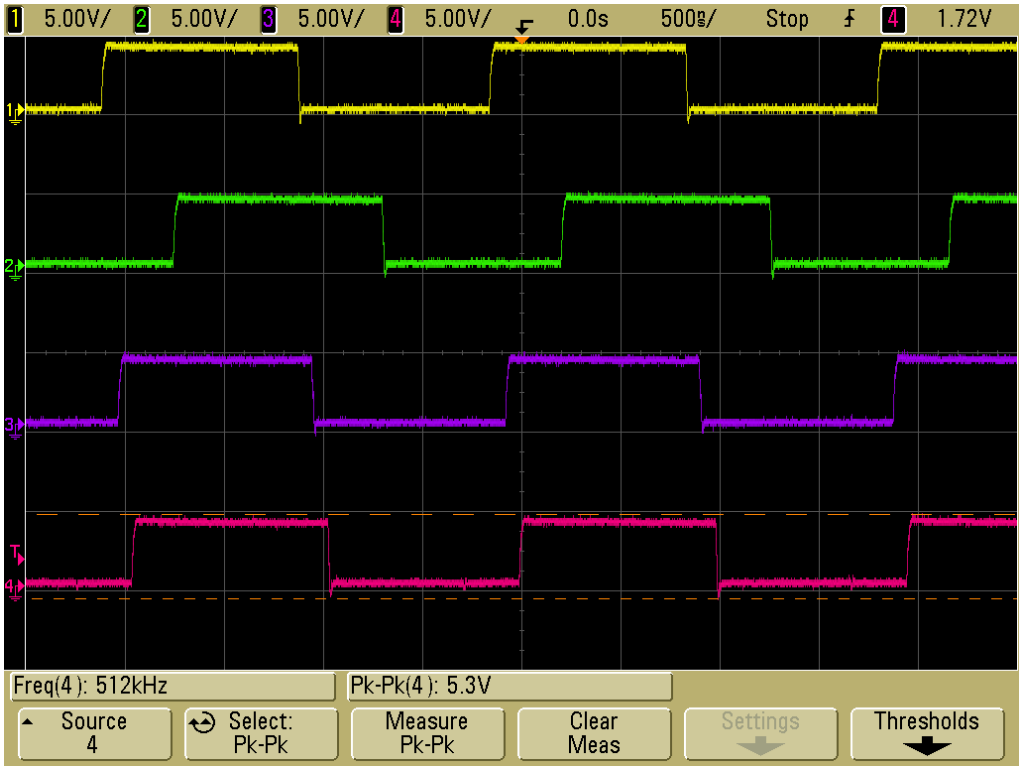


Figure 7.13: *Signals obtained at the output of the terminals in absence of interference*

A second situation is that pictured in the Figure 7.14. This time the terminals #1 and that number 3 are affected by interference. As evident comparing the waveform in green (terminal 2) with that purple (terminal 4, of reference), the transmission of the terminal #3 does not affect the reception of the terminal #2.

[6] for the numerical reference of the terminals, see the Figure 6.18

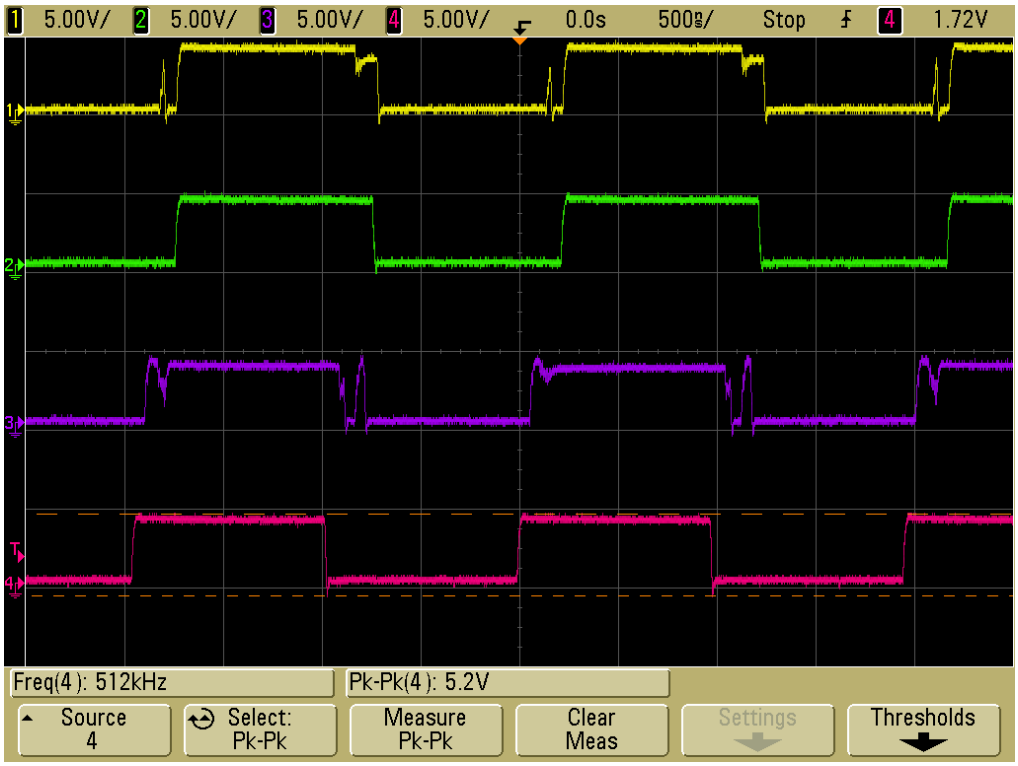


Figure 7.14: *A first situation in which the terminals begin to be affected by interference*

The Figure 7.15 shows an other situation in which the interference begins to be quite strong, at least for the terminals #1 and 3. Once again the terminal #2 is not affected by interference. Moreover we want to point out that this time the effects of the interference is strongest; for example the time duration of the high level (+5V) of the yellow signal is extended and the waveform of the terminal #3 is almost completely damage.

To end, the Figure 7.16 and Figure 7.17 depict two situations in which the interference is begun very strong, affecting all the communications of the terminal number 1, 2, 3. The terminal #4 continues to be of reference, maintaining as output a TTL signal with a repetition period of 512 KHz.

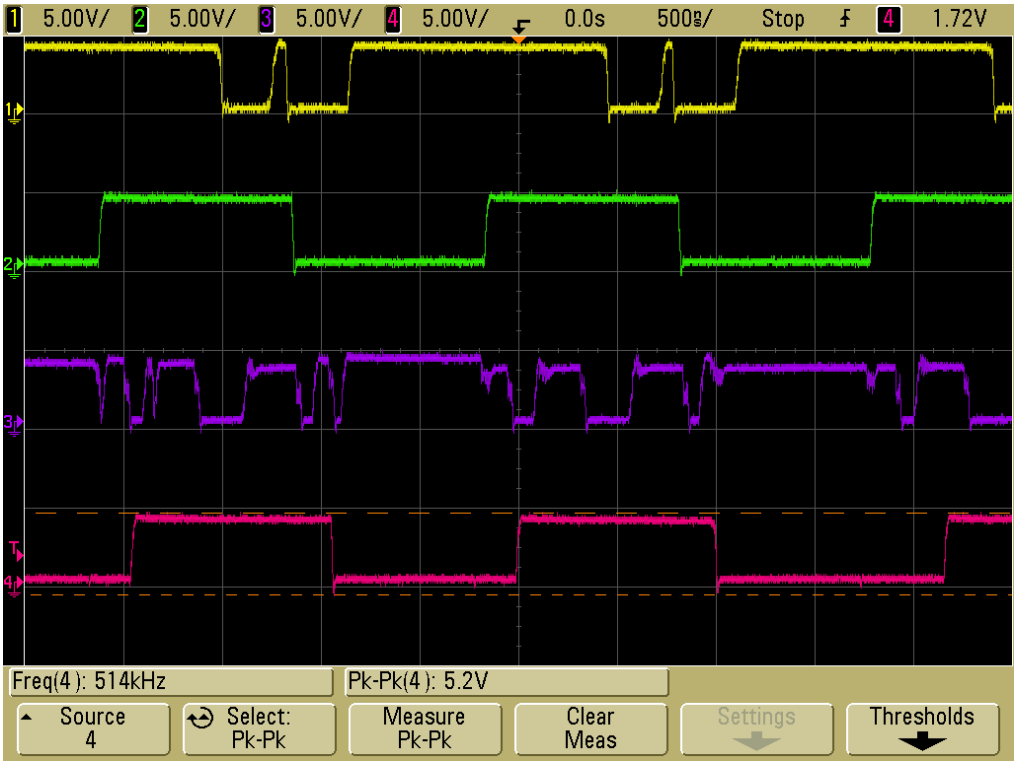


Figure 7.15: *A second situation in which the interference begins to be noticeable*

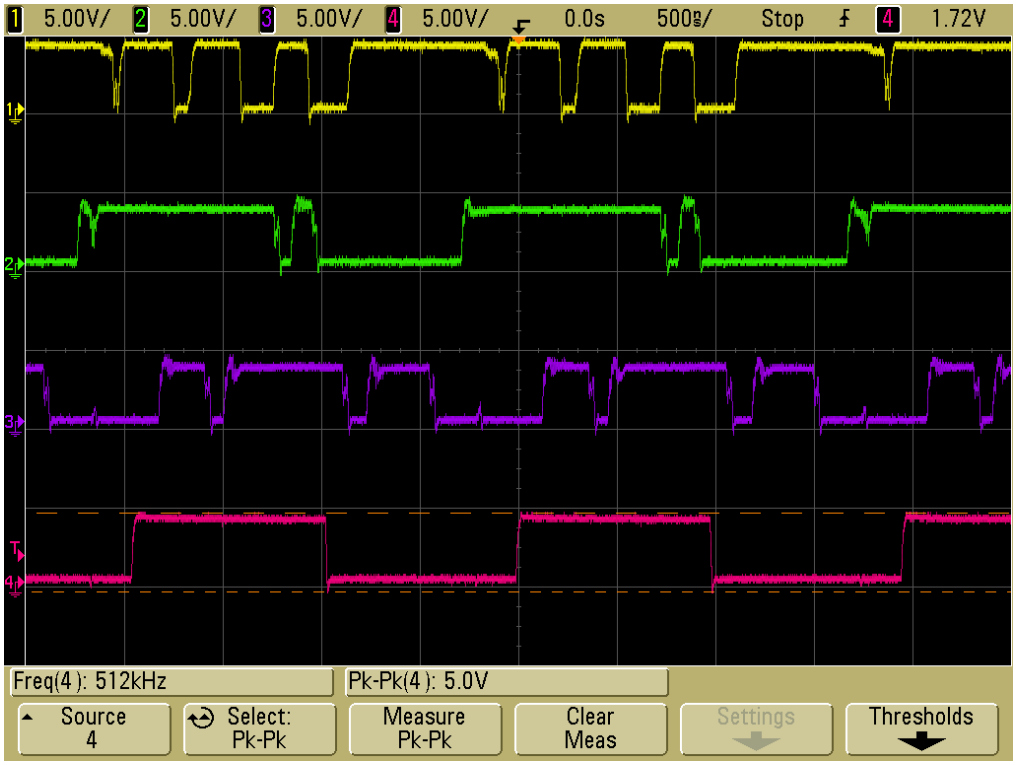


Figure 7.16: *A third situation in which the interference affects strongly each terminal*

Figure 7.17: *A fourth situation in which the interference affects strongly each terminal*

Chapter 8

Bibliography and Web Pages

Contents

8.1 Bibliography..... 142
8.2 Web Pages 145

8.1. Bibliography

[1] S. Barbarossa, "Multi-antenna systems", 2004.

[2] M. Cavagnaro, Lectures of "Environmental Impact of Electromagnetic Fields", University of Rome "La Sapienza", April – June 2004.

[3] M.G. Di Benedetto, Lectures of "Wireless Access", University of Rome "La Sapienza", October – December 2004.

[4] M.G. Di Benedetto, Lectures of "*Optical Transmission*", University of Rome "La Sapienza", April – June 2004.

[5] C. Mencuccini, V. Silvestrini, "*Physics II*", Liguori, 1998.

[6] A.V. Oppenheim and R. W. Schafer, "*Discrete-Time Signal Processing*", Prentice-Hall, 1989.

[7] G. Columpsi, M. Leonardi, A. Ricci, "*UMTS, Techniques and Architecture for Multimedia Mobile Communication Network*", Hoepli, 2003.

[8] Thomas L. Floyd, "*Fundamentos de Sistemas Digitales*", 7ª edición, Prentice-Hall, 2000.

[9] M.G. Di Benedetto and G. Giancola, "*Understanding Ultra Wide Band Radio Fundamentals*", Prentice-Hall, 2004.

[10] Altera, "*Max+Plus II getting Started*", 1997.

[11] National Semiconductors, "*National Operational Amplifiers Databook*", 1995.

[12] "Introductory Tutorial, Exploring Protel 99 SE".

[13] Manuel Torres, "*Diseño e ingenieria electrónica asistida sobre PROTEL*", Ra-Ma, 2000.

[14] C-MAC Frequency Products, "*Crystal Product Data Book 2000*".

[15] Freescale Semiconductor, "802.15.4 MAC PHY Software, User's Guide", 2005.

[16] Santiago T. Pérez Suárez, "*Sistema de comunicación fhss síncrono para ccoo no guiadas*", Universidad de Las Palmas de Gran Canaria, 2002.

[17] Luca De Nardis, "*Location-Aware, Power-Efficient MAC and Routing Strategies for UWB Wireless Communications*", University of Rome "La Sapienza", 2004.

[18] A. E. García Braun, J. A. Rabadán Borges, M. Á. Bacallado Marrero y R. Pérez Jiménez, "Sistema de comunicación óptica no guiada basado en tecnología IrDA y técnicas DSSS", July 2005.

[19] I. Millar, M. Beale, B. J. Donoghue, K. W. Lindstrom, S. Williams, "The IrDA Standards for High-Speed Infrared Communications", The Hewlett Packard Journal, 1998.

[20] Andrea J. Goldsmith, Stephen B. Wicker, "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks", IEEE Wireless Communication, August 2002.

[21] V. Bharghavan et Al., "MACAW: A Media Access Protocol for Wireless LANs" *Proc. ACM SIGCOMM '94*, pp. 212-225, Aug. 1994.

[22] K. K. Wong, Tim O'Farrell, "Spread Spectrum Techniques for Indoor Wireless IR Communications", IEEE Wireless Communication, pp. 54-63, April 2003.

[23] Maria-Gabriella Di Benedetto, Luca De Nardis, Matthias Junk and Guerino Giancola, : "$(UWB)^2$: Uncoordinated, Wireless, Baseborn Medium Access for UWB Communication Networks", 2005.

8.2. Web Pages

General information

[URL I]	http://www.irda.org	IrDA
[URL II]	http://www.sss-mag.com	Spread Spectrum
[URL III]	http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html	
[URL IV]	http://datasheetarchive.com/	Datasheets
[URL V]	http://www.uwbinsider.com/	UWB

Official Organisms

[URL VI]	http://www.acit-canarias.org	IEEE
[URL VII]	http://www.fcc.gov/	FCC
[URL VIII]	http://www.ntia.doc.gov/	NTIA
[URL IX]	http://www.etsit.ulpgc.es	Escuela Técnica Superior de Ingenieros de Telecomunicación (ULPGC)

Component's Searchers

[URL X]	www.chipcenter.com	ChipCenter
[URL XI]	www.techonline.com	TechOnline
[URL XII]	www.icmaster.com	IC Master

Manufacturer

[URL XIII]	http://www.altera.com	Altera
[URL XIV]	http://www.analog.com	Analog Devices
[URL XV]	http://www.hamamatsu.es	Hamamatsu
[URL XVI]	https://maxim-ic.com	Maxim

Component's Supplier

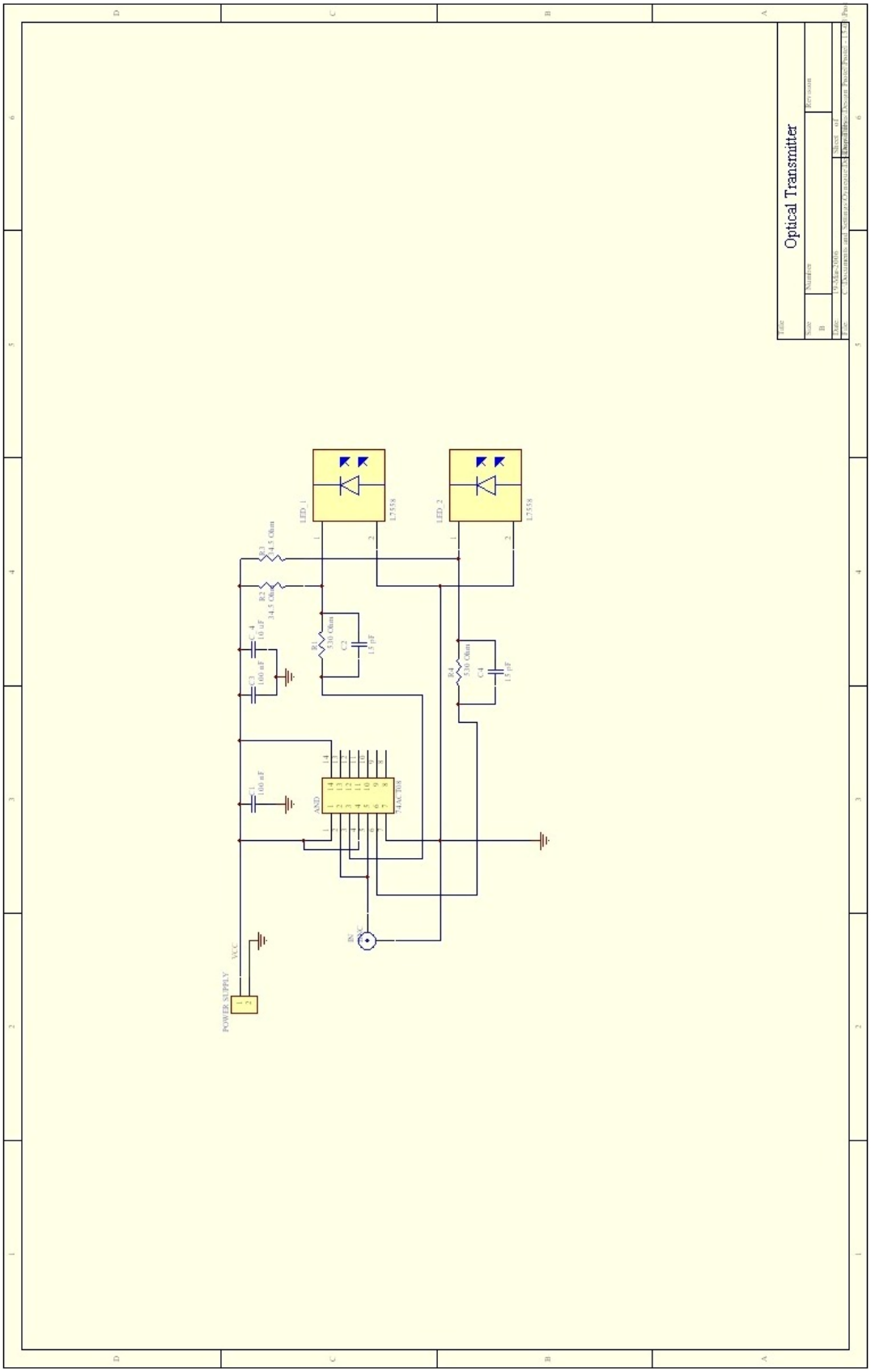
[URL XVII]	http://www.lopacan.com	Lopacan
[URL XVIII]	http://www.amidata.es	RS
[URL XIX]	http://www.farnell.com	Farnell

Annex I. Schematics and PCBs

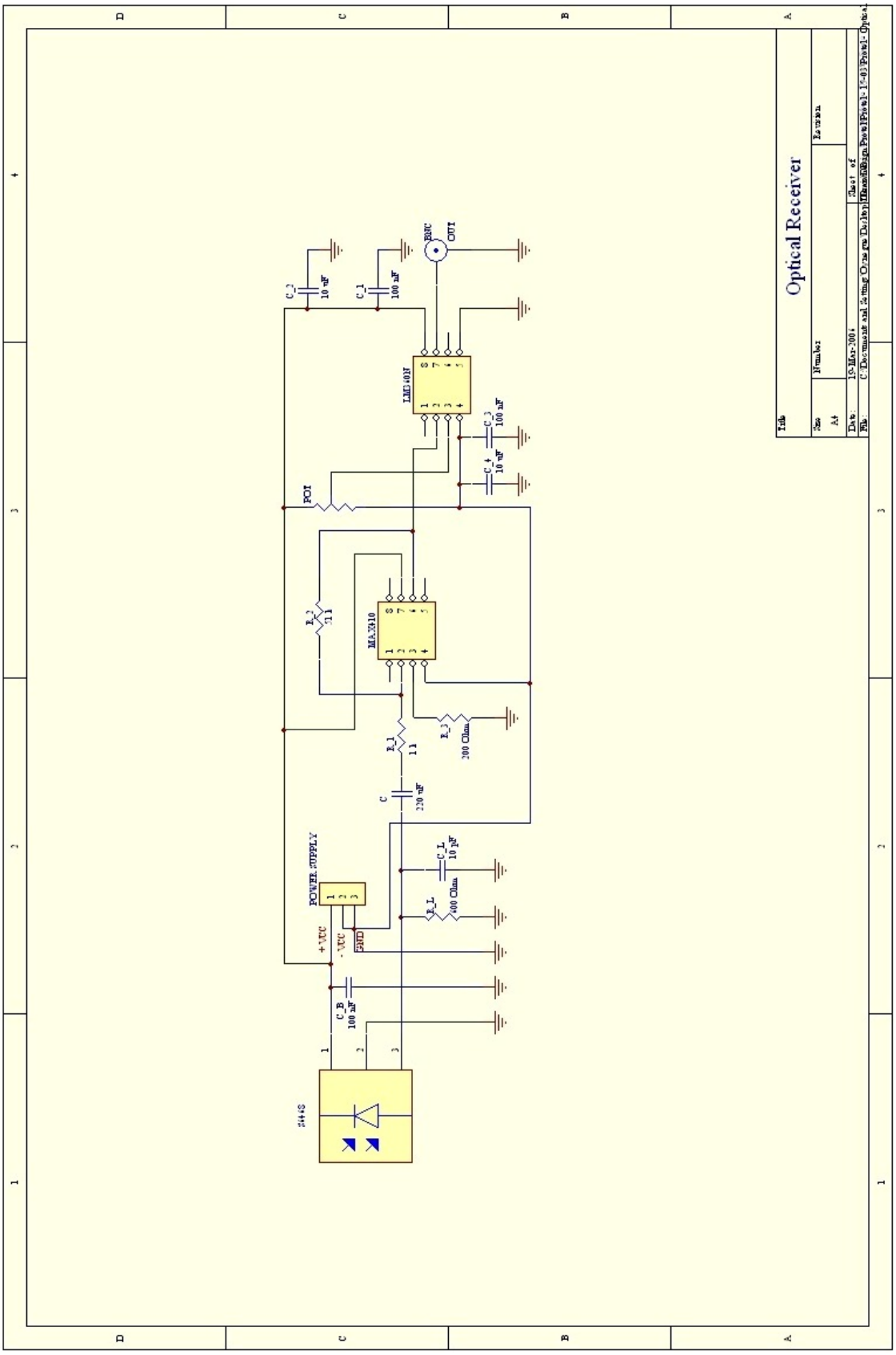
Contents

AI.1	**Schematic of the optical transmitter**	**149**
AI.2	**Schematic of the optical receiver**	**151**
AI.3	**Schematic of the optical terminal**	**153**
AI.4	**Schematic of the EPLD relative to the master number one**	**155**
AI.5	**Schematic of the EPLD relative to the master number two**	**157**
AI.6	**Schematic of the EPLD relative to the to the slave number one**	**159**
AI.7	**Schematic of the EPLD relative to the to the slave number two**	**161**
AI.8	**Schematic of the EPLDs of the optical system**	**163**
AI.9	**PCB of the optical transmitter**	**165**
AI.10	**PCB of the optical receiver**	**169**
AI.11	**PCB of the optical terminal**	**173**
AI.12	**PCB of the optical system**	**177**
AI.13	**PCBs of the EPLDs of the optical system**	**183**
AI.14	**Schematic of the drive logic of the master number 1 and number 2**	**189**
AI.15	**Schematic of the drive logic of the slave number 1 and number 2**	**191**
AI.16	**Schematic of the 4 taps shift register**	**193**
AI.17	**Schematic of the 16 taps shift register**	**195**
AI.18	**Schematic of the Pseudo Random Number Generator**	**197**
AI.19	**Schematic of the optical system used in the software simulation**	**199**

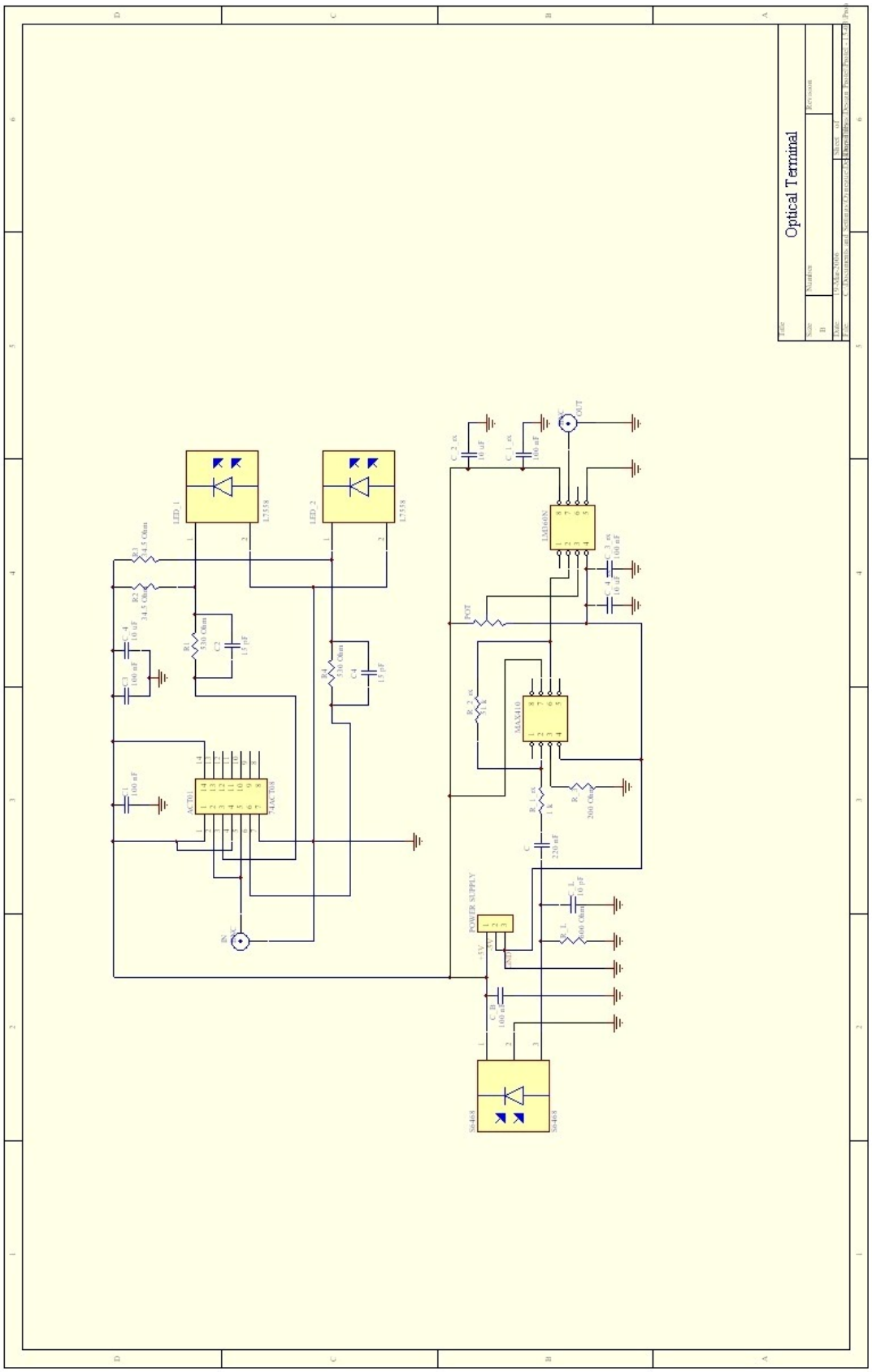
AI.1. Schematic of the optical transmitter



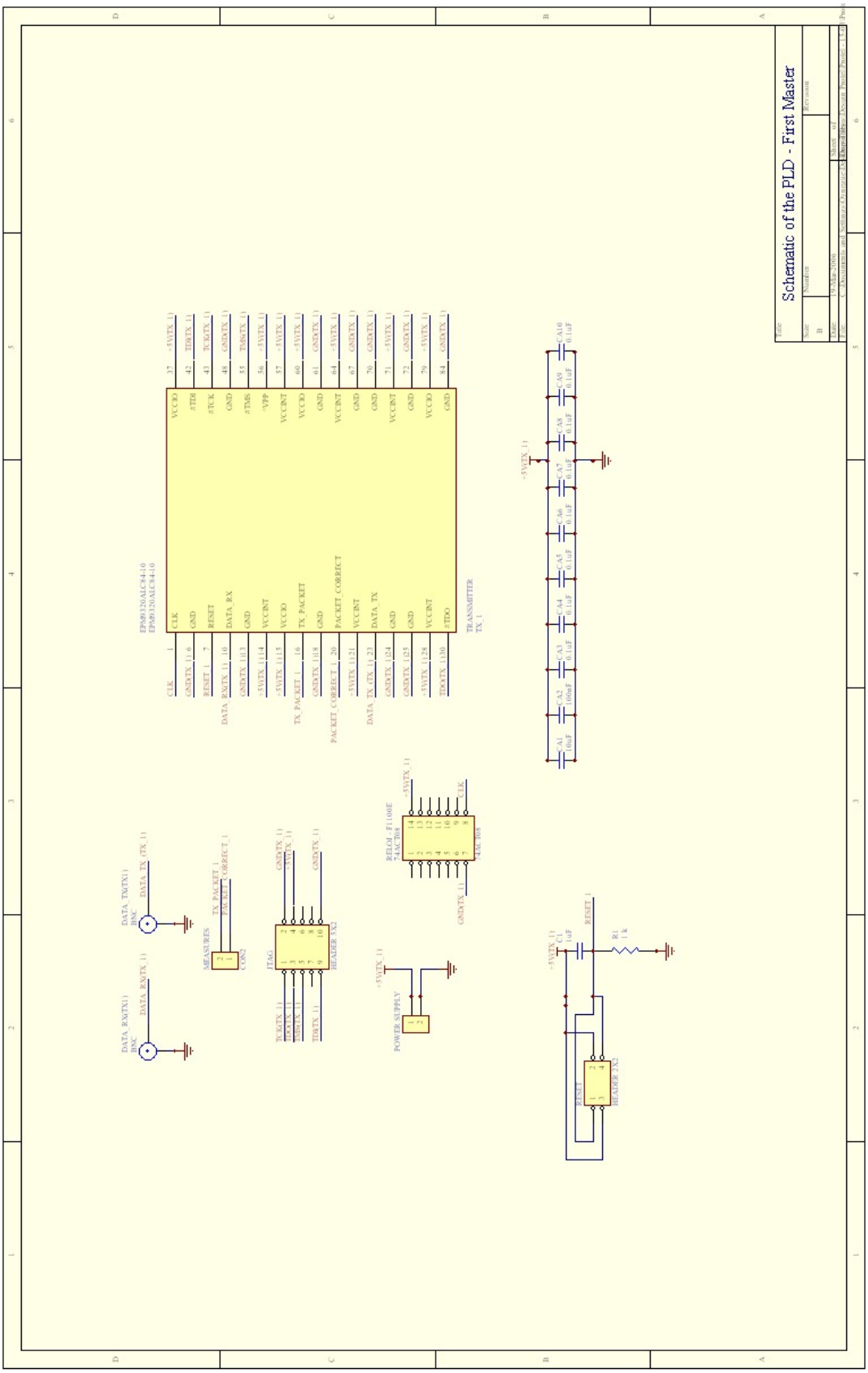
AI.2. Schematic of the optical receiver



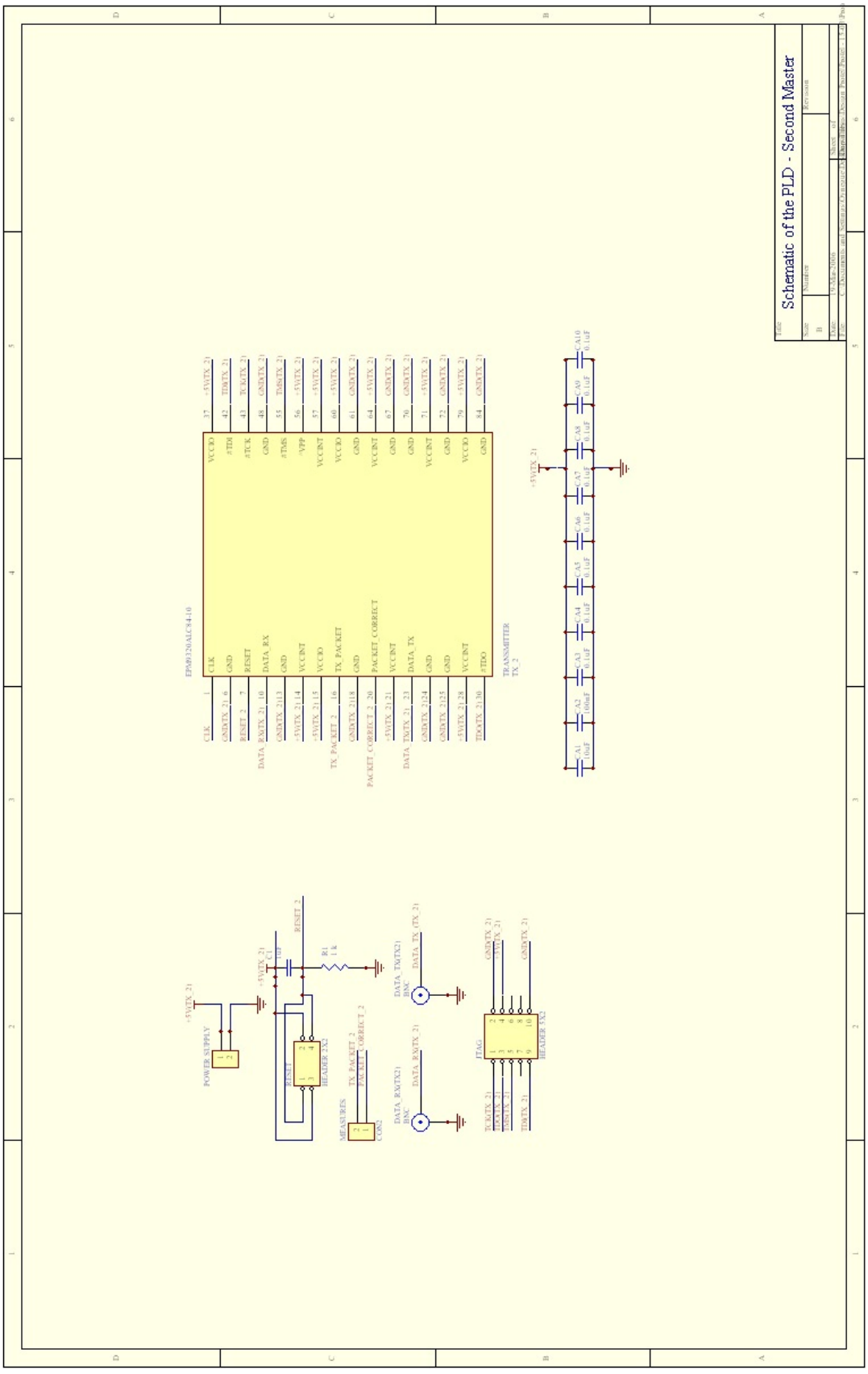
AI.3. Schematic of the optical terminal



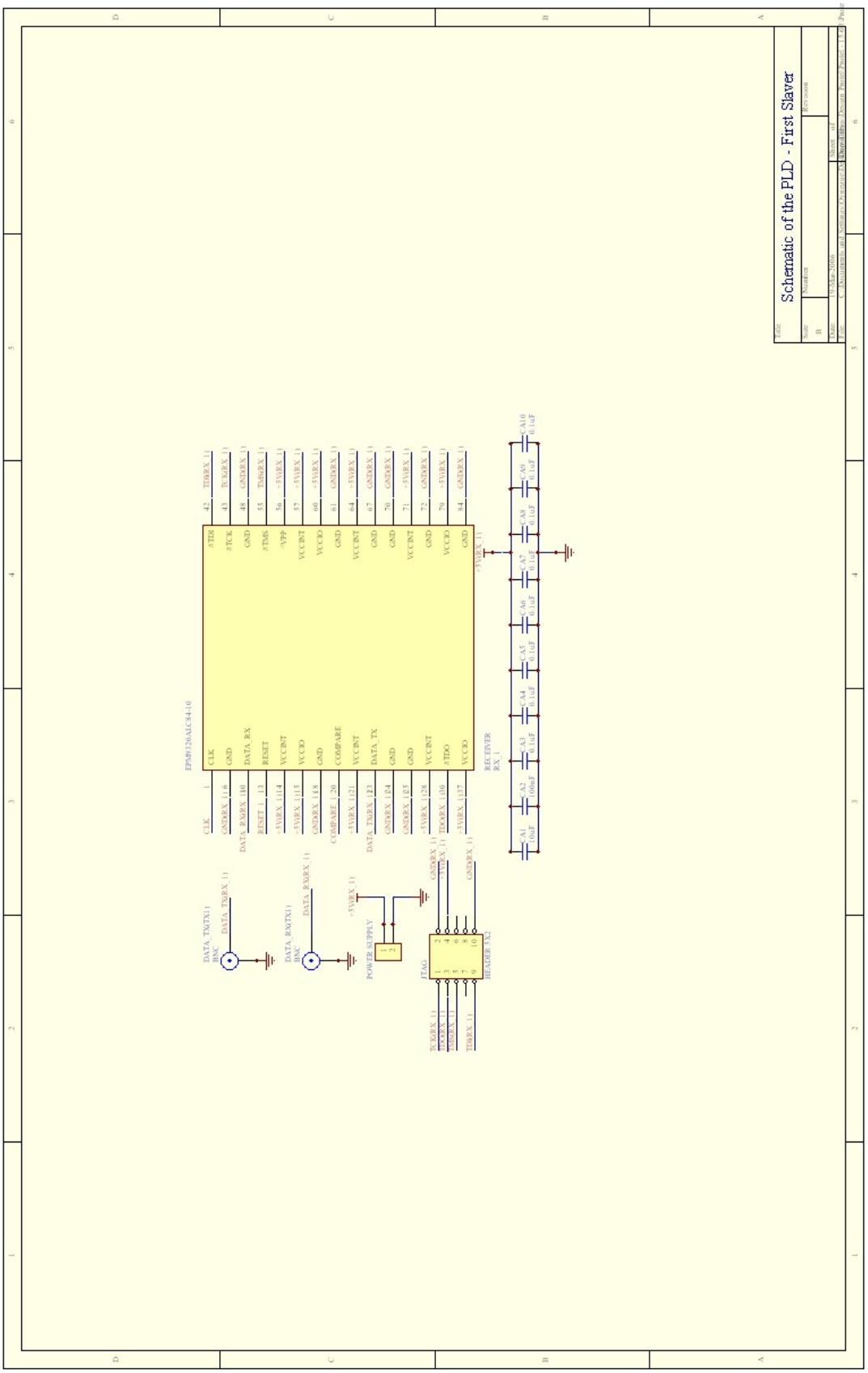
AI.4. Schematic of the EPLD relative to the master number one



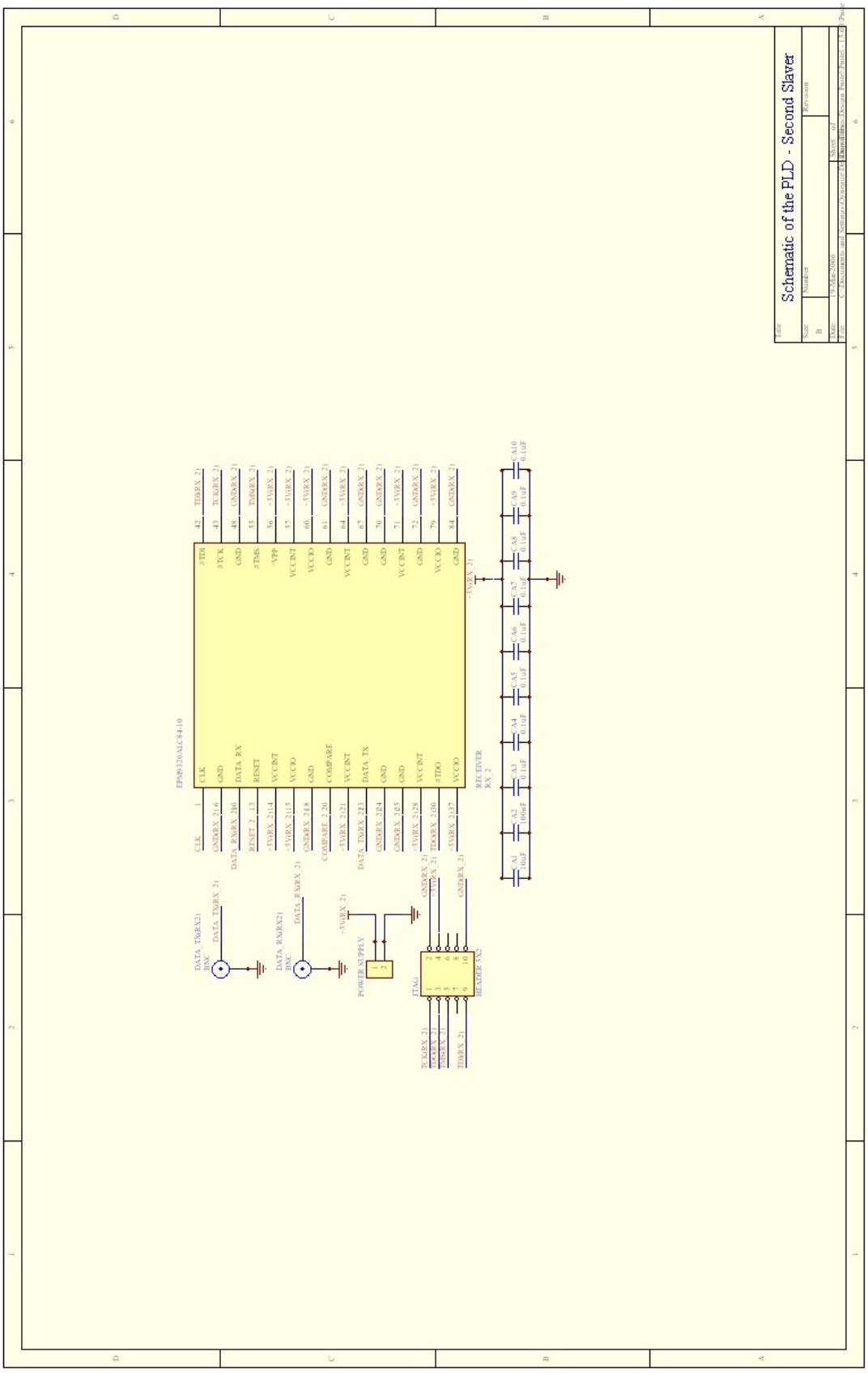
AI.5. Schematic of the EPLD relative to the master number two



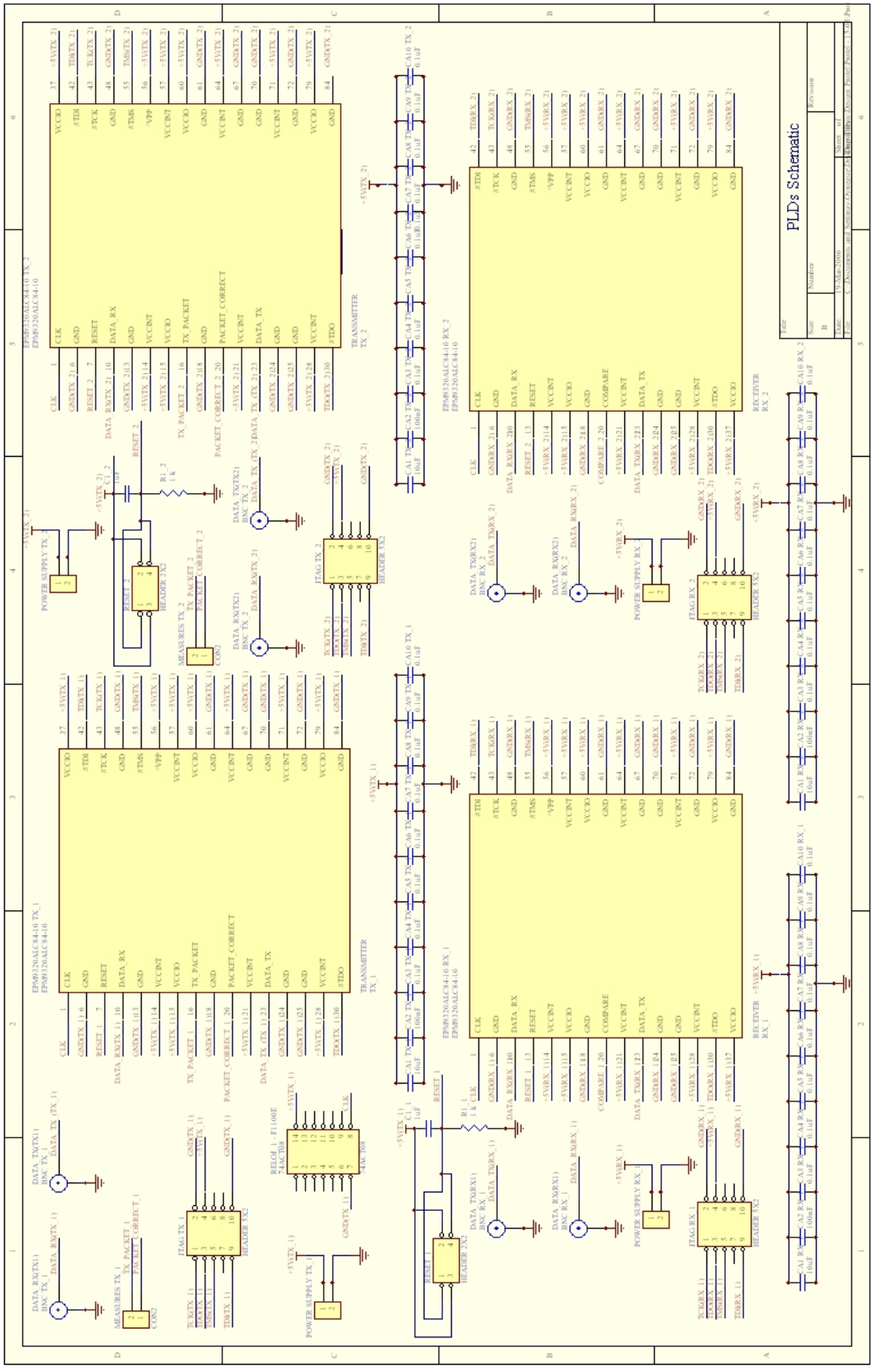
AI.6. Schematic of the EPLD relative to the slave number one



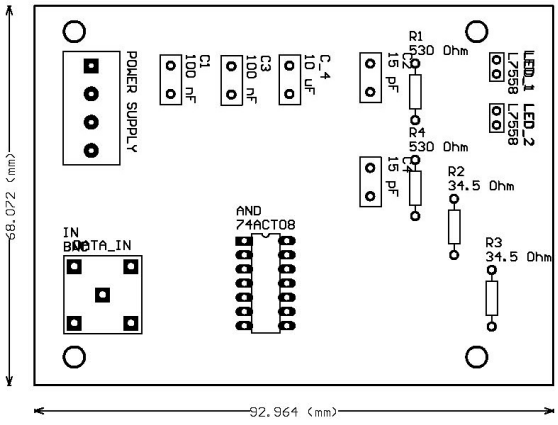
AI.7. Schematic of the EPLD relative to the slave number two

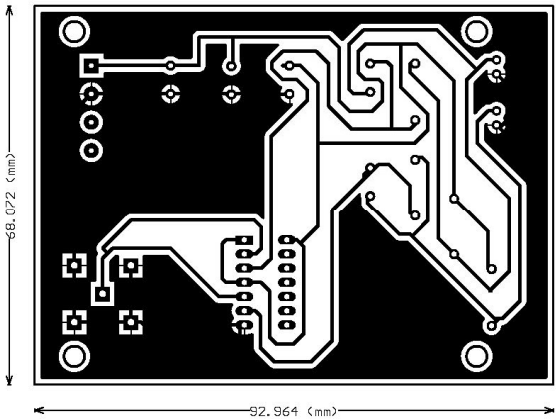


AI.8. Schematic of the EPLDs of the optical system

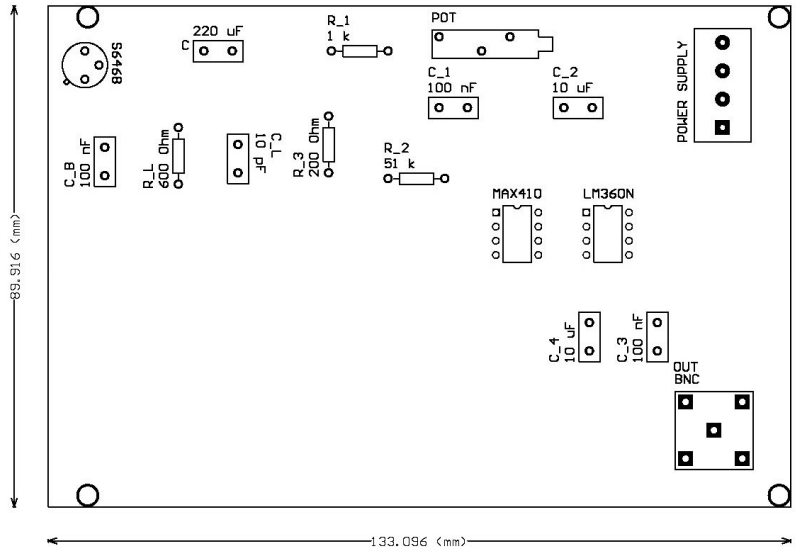


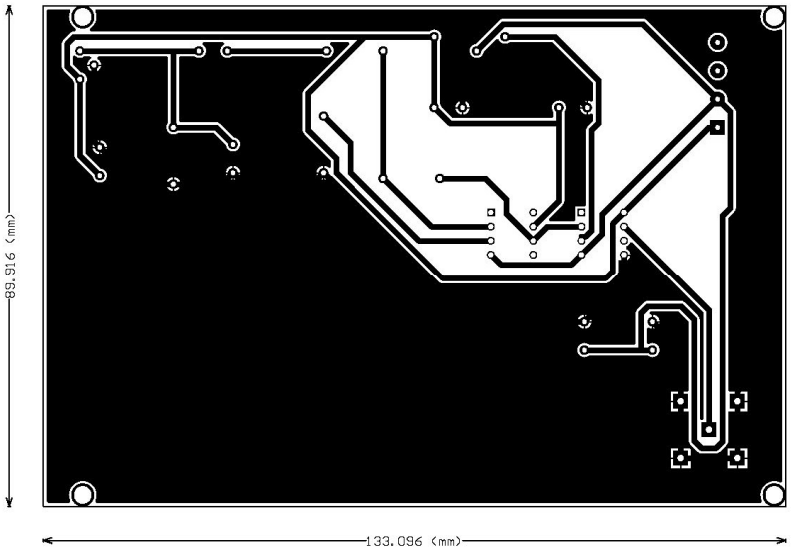
AI.9. PCB of the optical transmitter



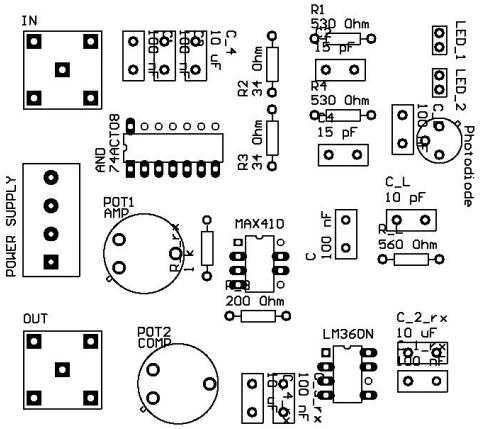

68.072 (mm)
92.964 (mm)

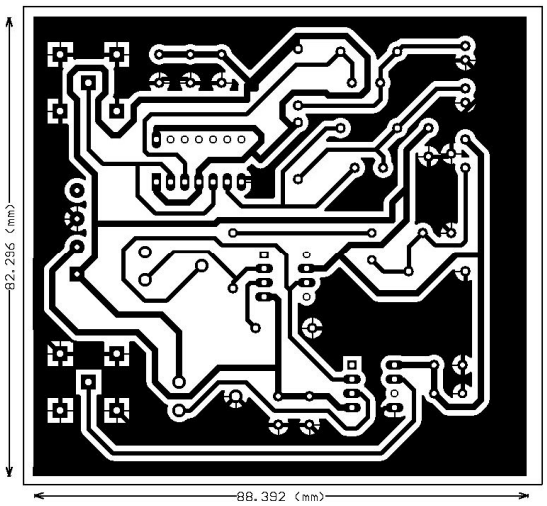
AI.10. PCB of the optical receiver



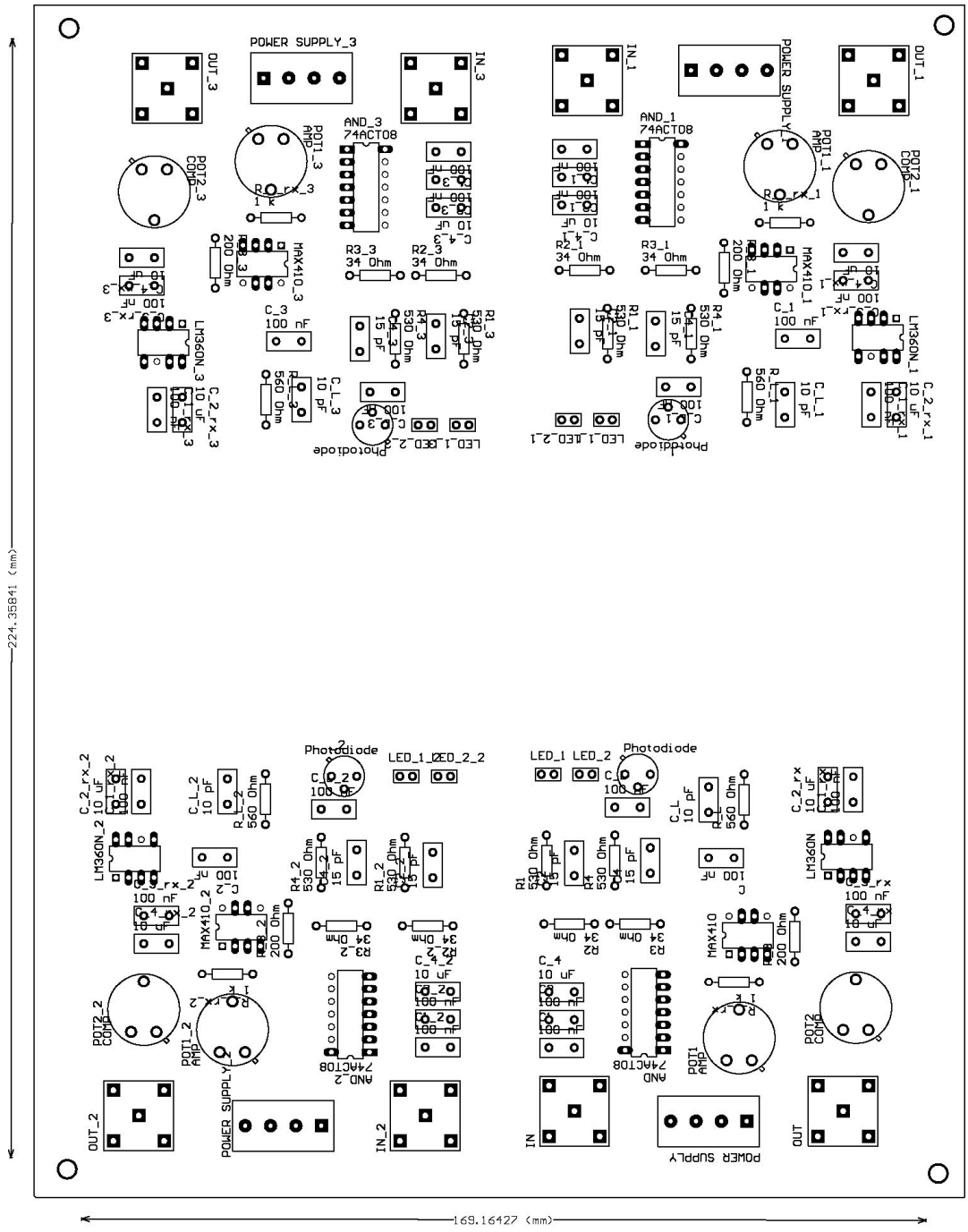

89.916 (mm)
133.096 (mm)

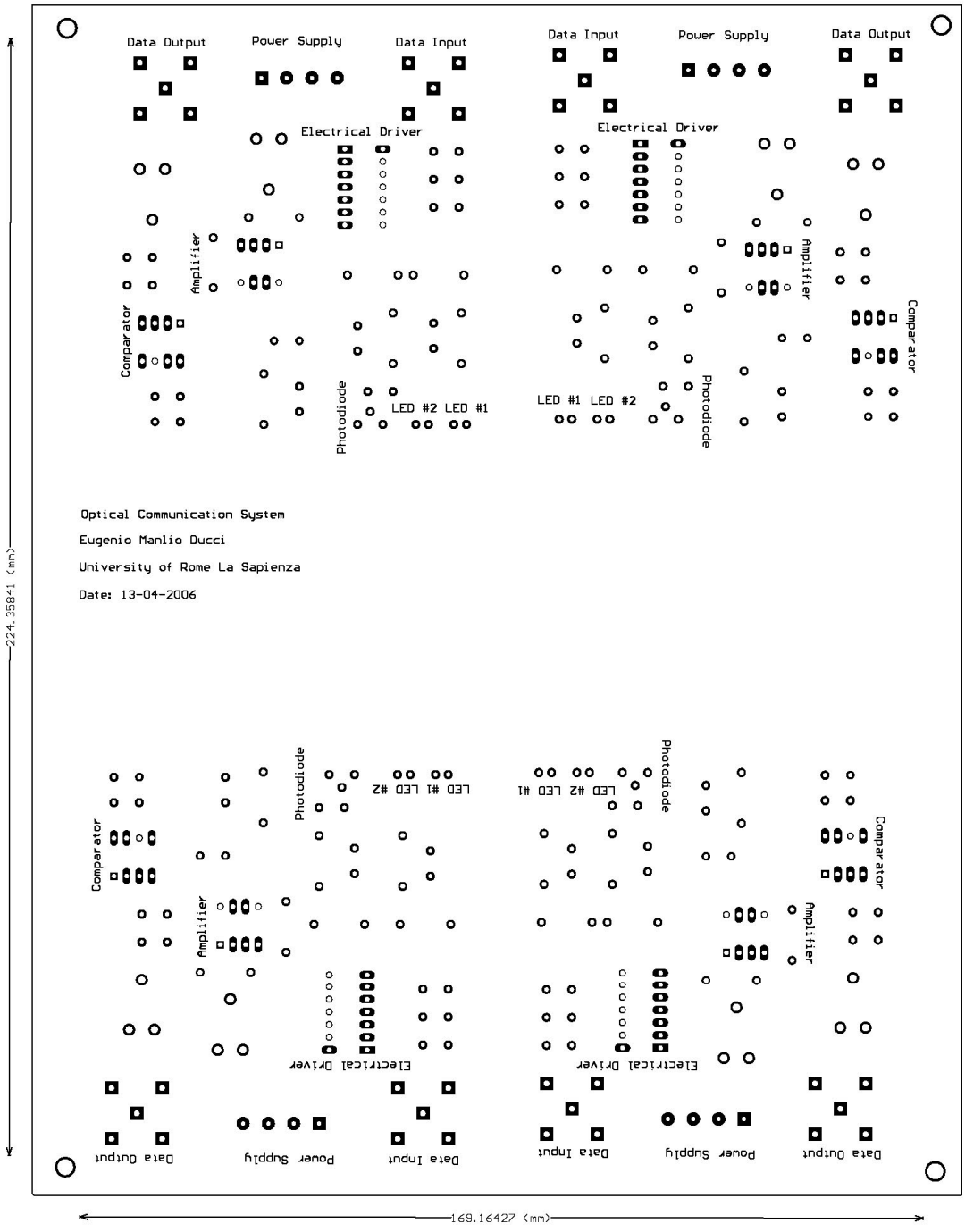
AI.11. PCB of the optical terminal

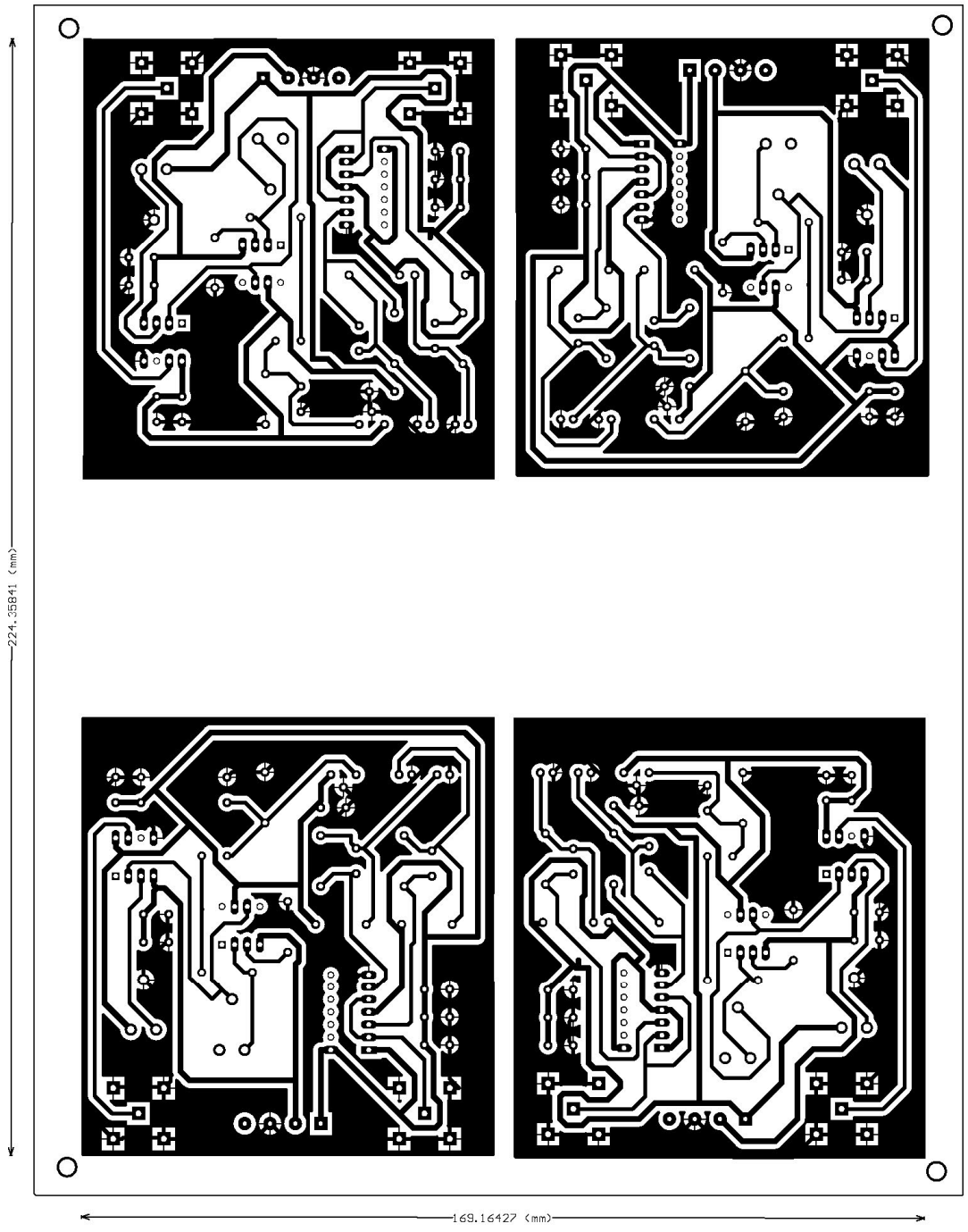



82.296 (mm)
88.392 (mm)

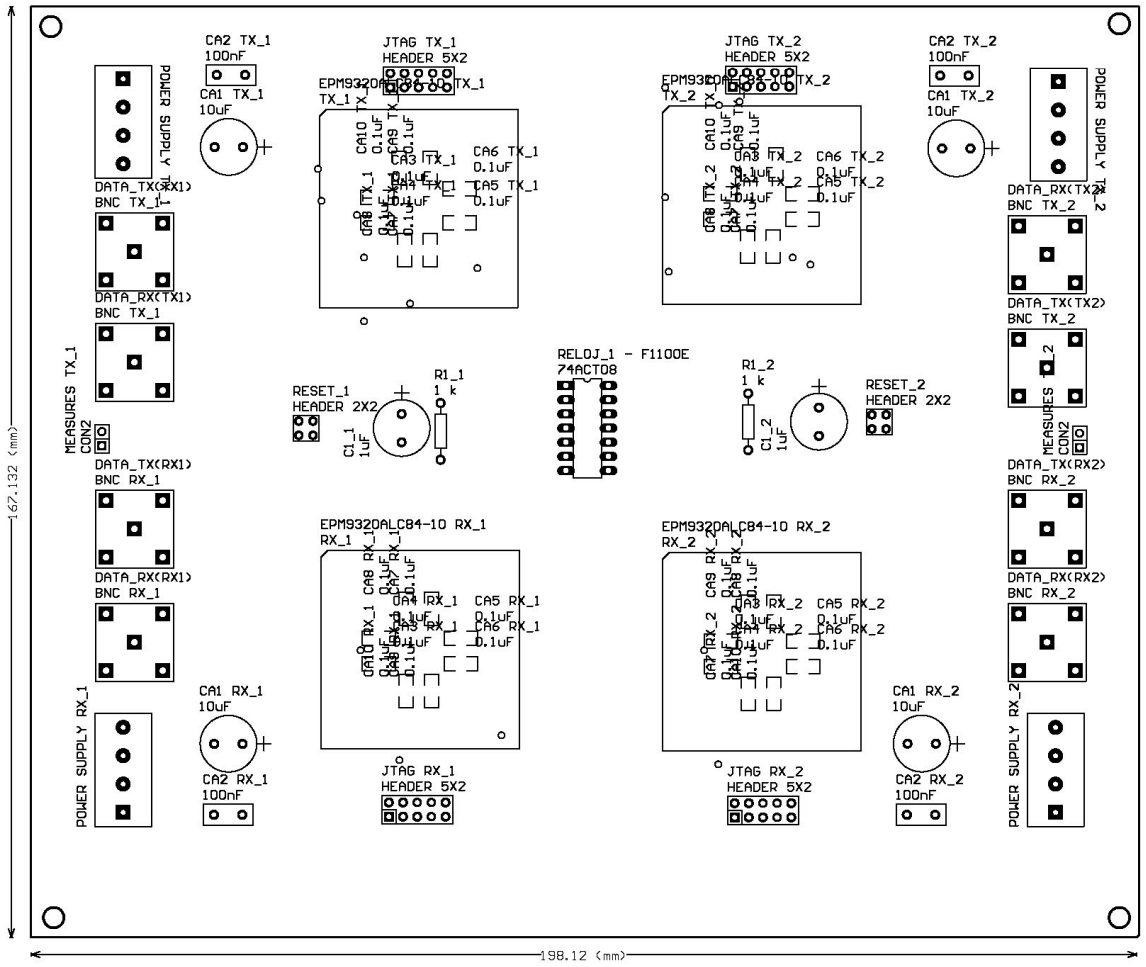
AI.12. PCB of the optical system

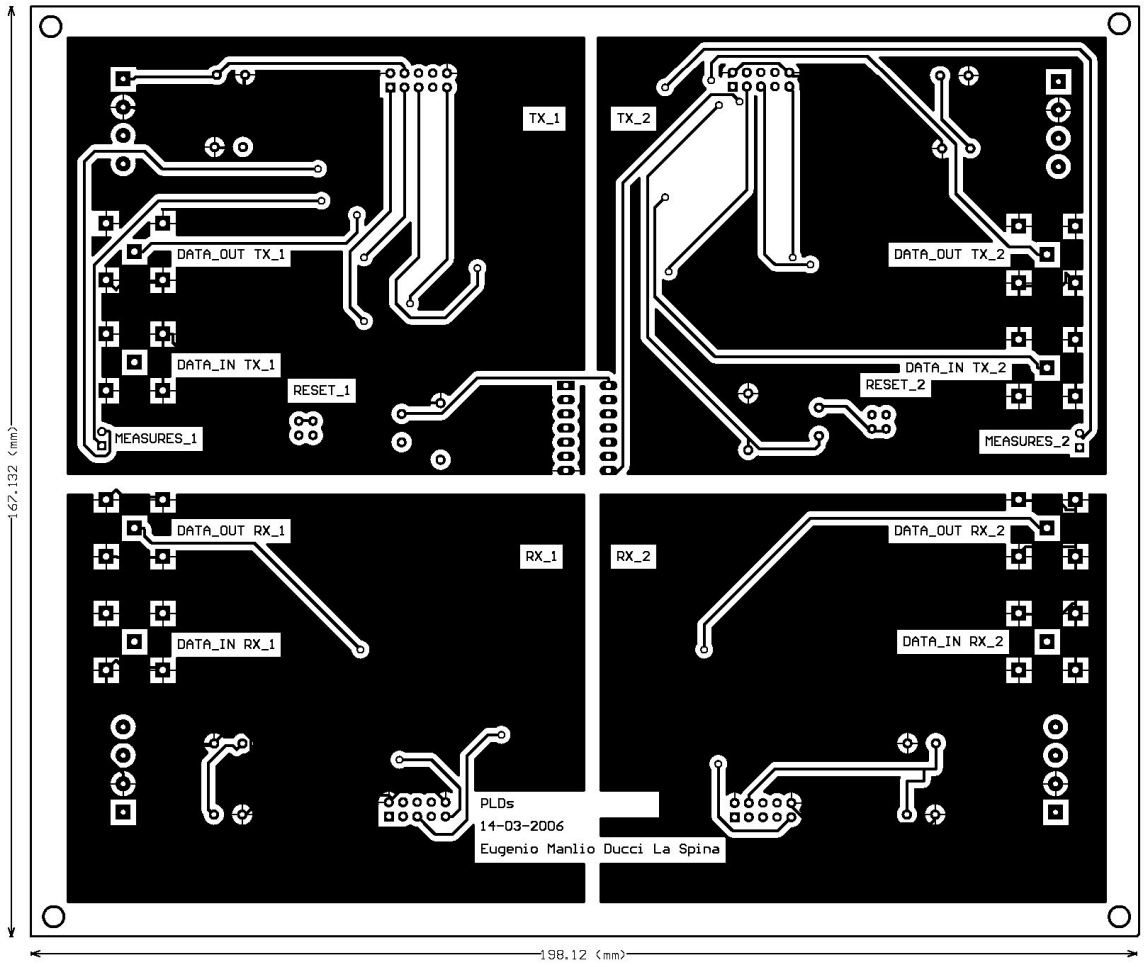


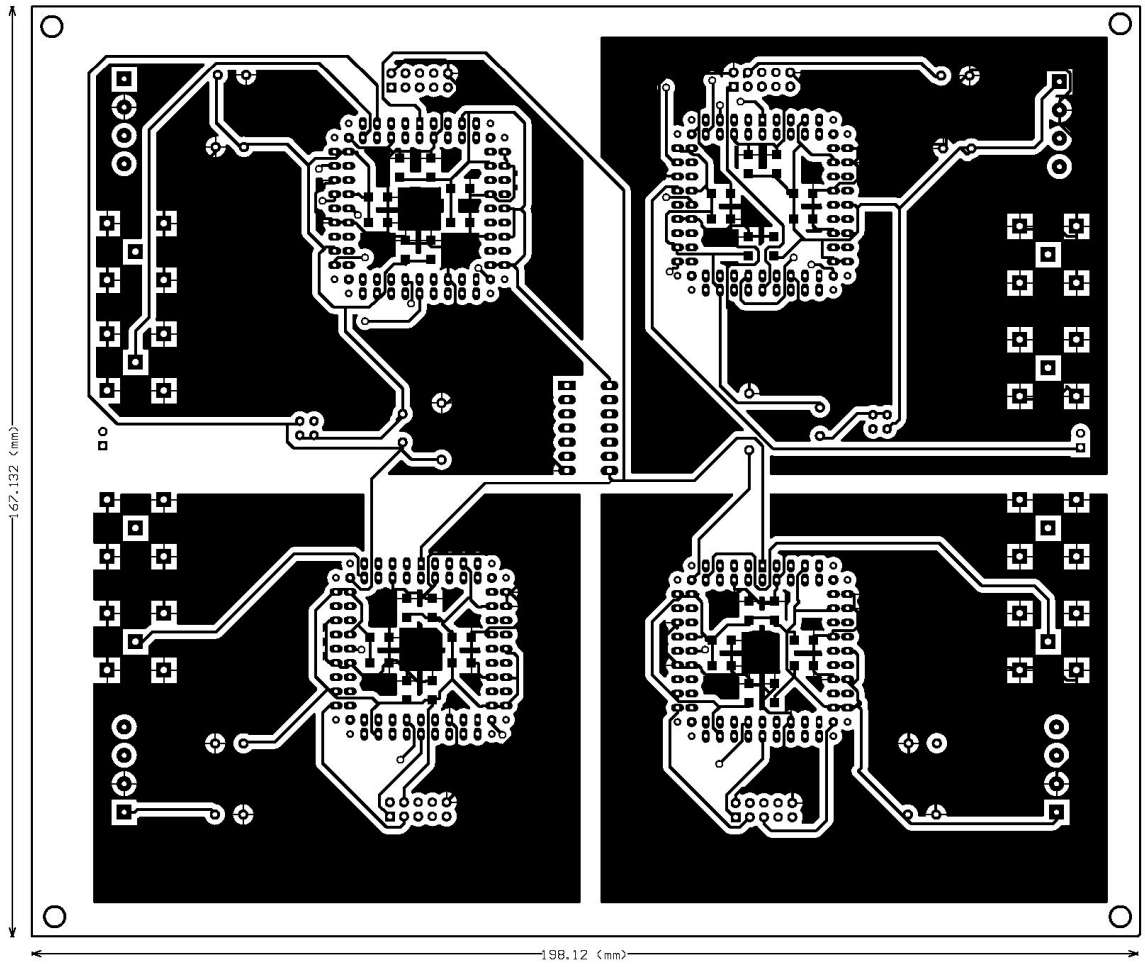

Data Output
Power Supply
Data Input
Electrical Driver
Amplifier
Comparator
Photodiode
LED #2 LED #1
Optical Communication System
Eugenio Manlio Ducci
University of Rome La Sapienza
Date: 13-04-2006
224.35841 (mm)
169.16427 (mm)


224.35841 (mm)
169.16427 (mm)

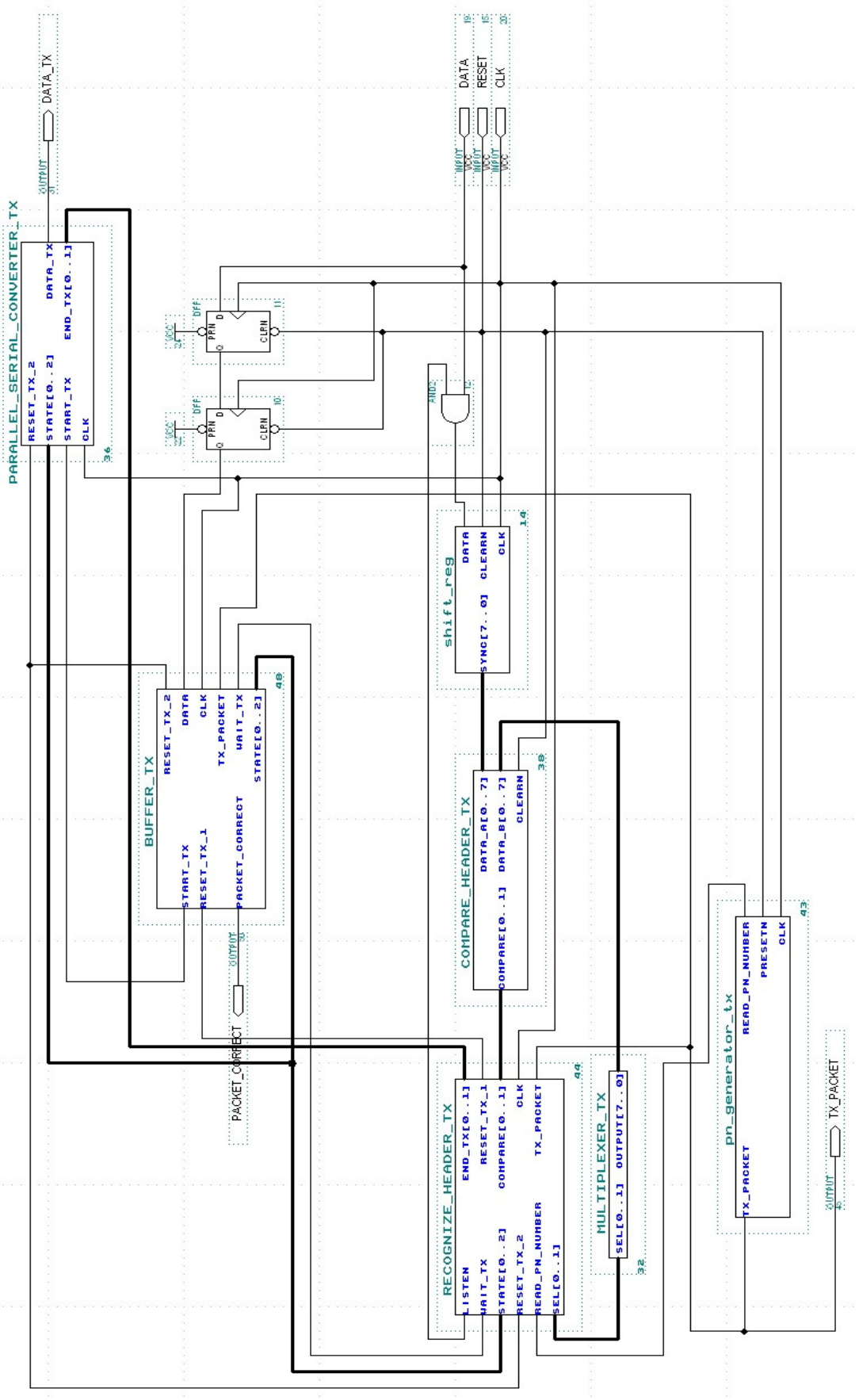
AI.13. PCBs of the EPLDs of the optical system



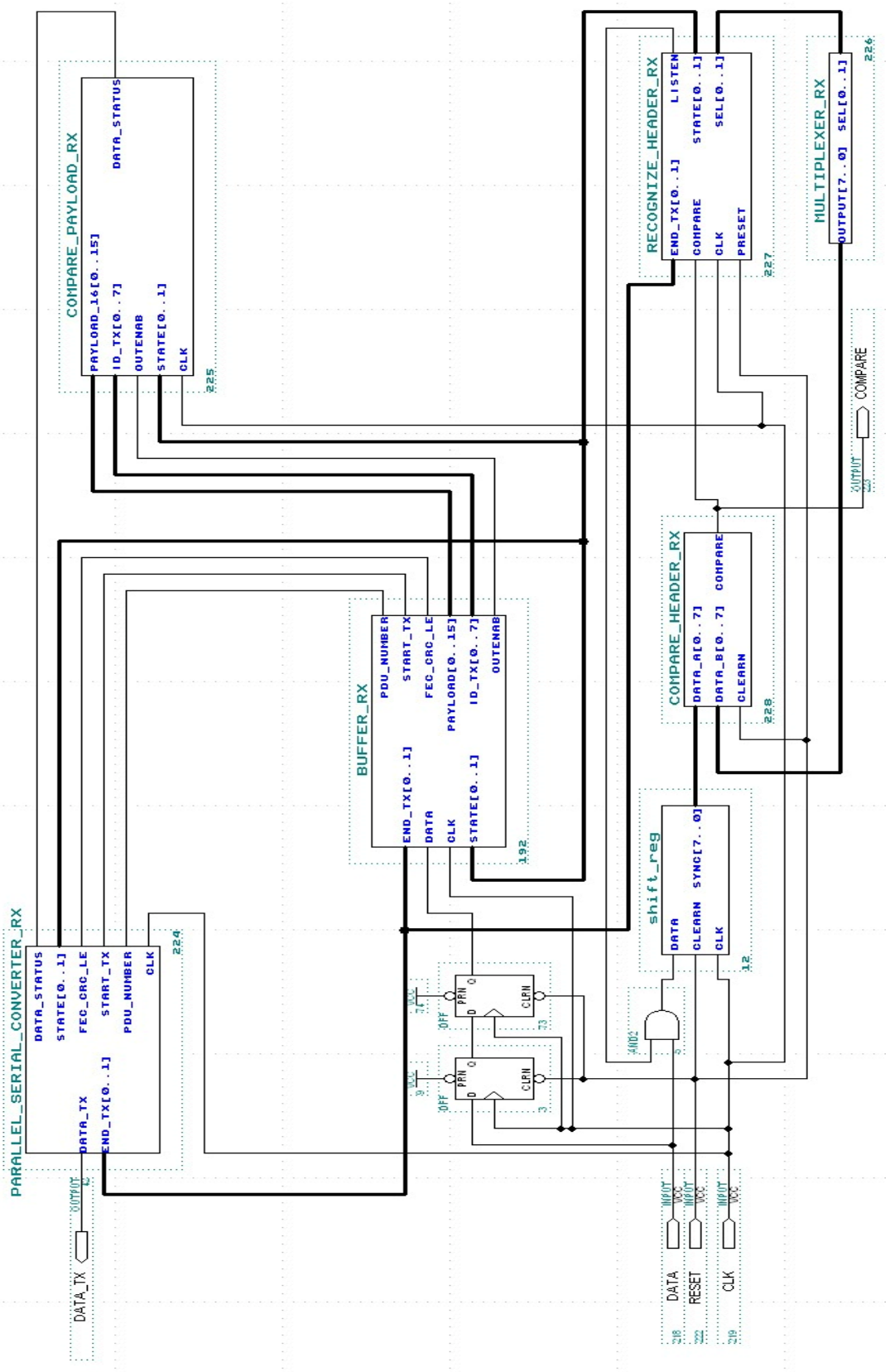

TX_1
TX_2
DATA_OUT TX_1
DATA_OUT TX_2
DATA_IN TX_1
DATA_IN TX_2
RESET_1
RESET_2
MEASURES_1
MEASURES_2
DATA_OUT RX_1
DATA_OUT RX_2
RX_1
RX_2
DATA_IN RX_1
DATA_IN RX_2
PLDs
14-03-2006
Eugenio Manlio Ducci La Spina
167.132 (mm)
198.12 (mm)


167.132 (mm)
198.12 (mm)

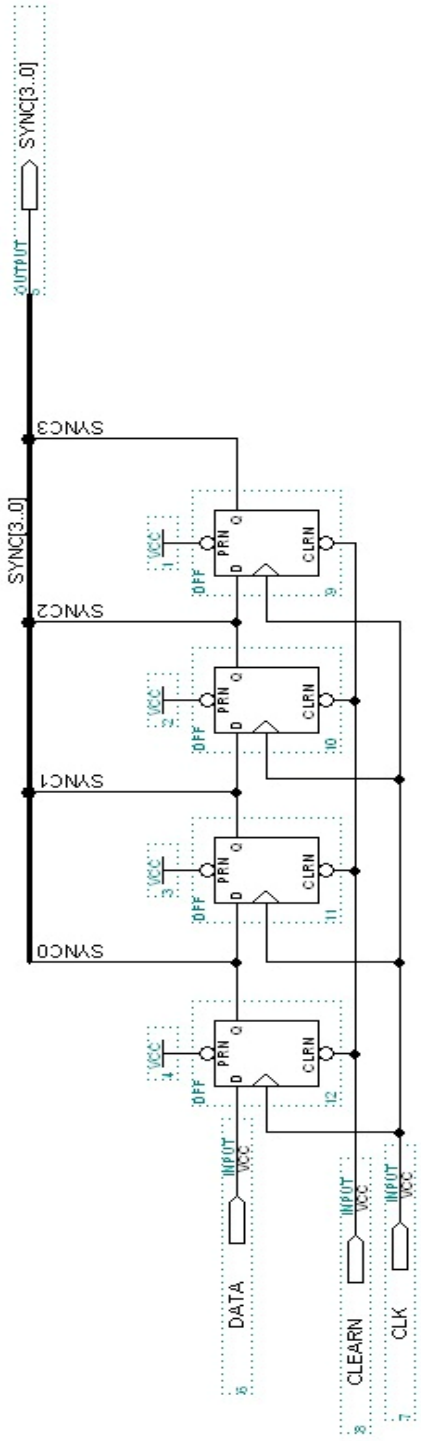
AI.14. Schematic of the drive logic of the master number 1 and number 2



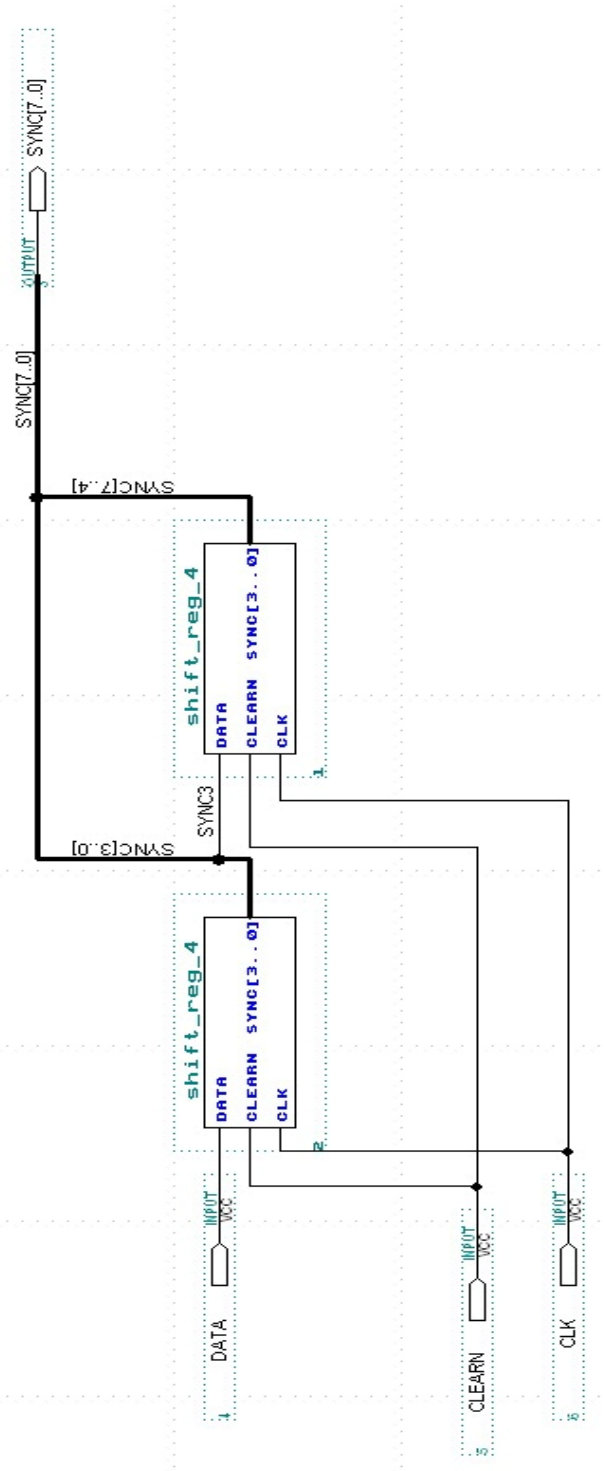
AI.15. Schematic of the drive logic of the slave number 1 and number 2



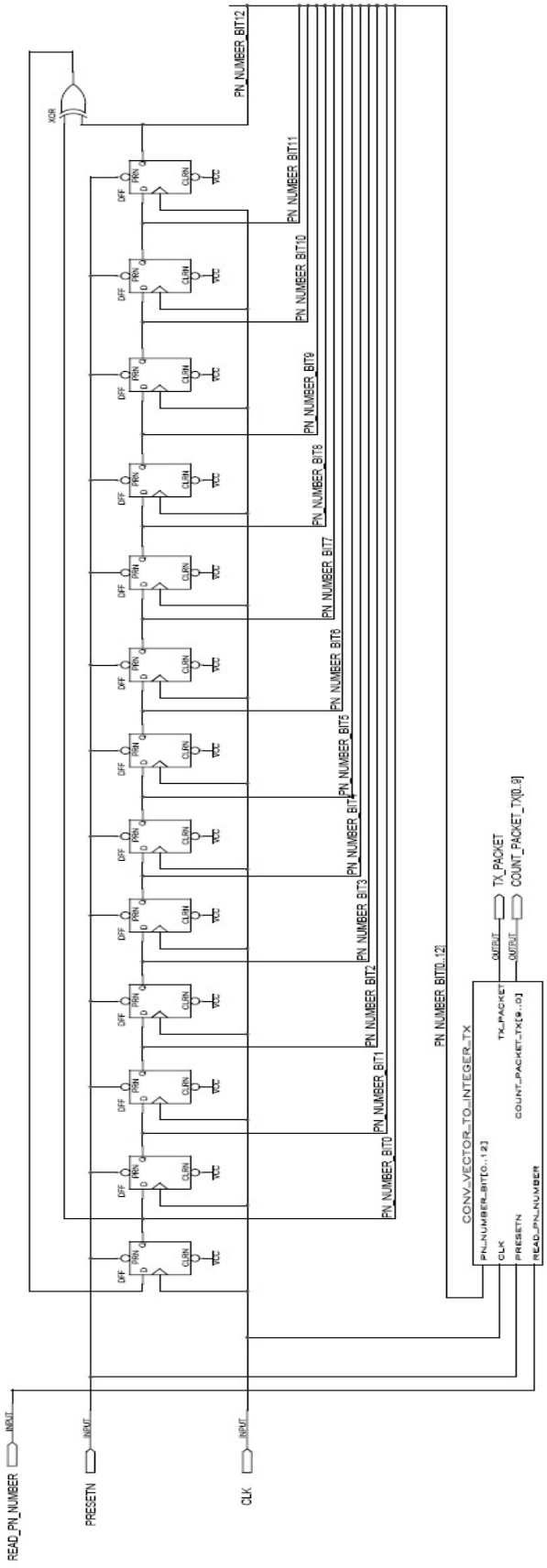
AI.16. Schematic of the 4 taps shift register



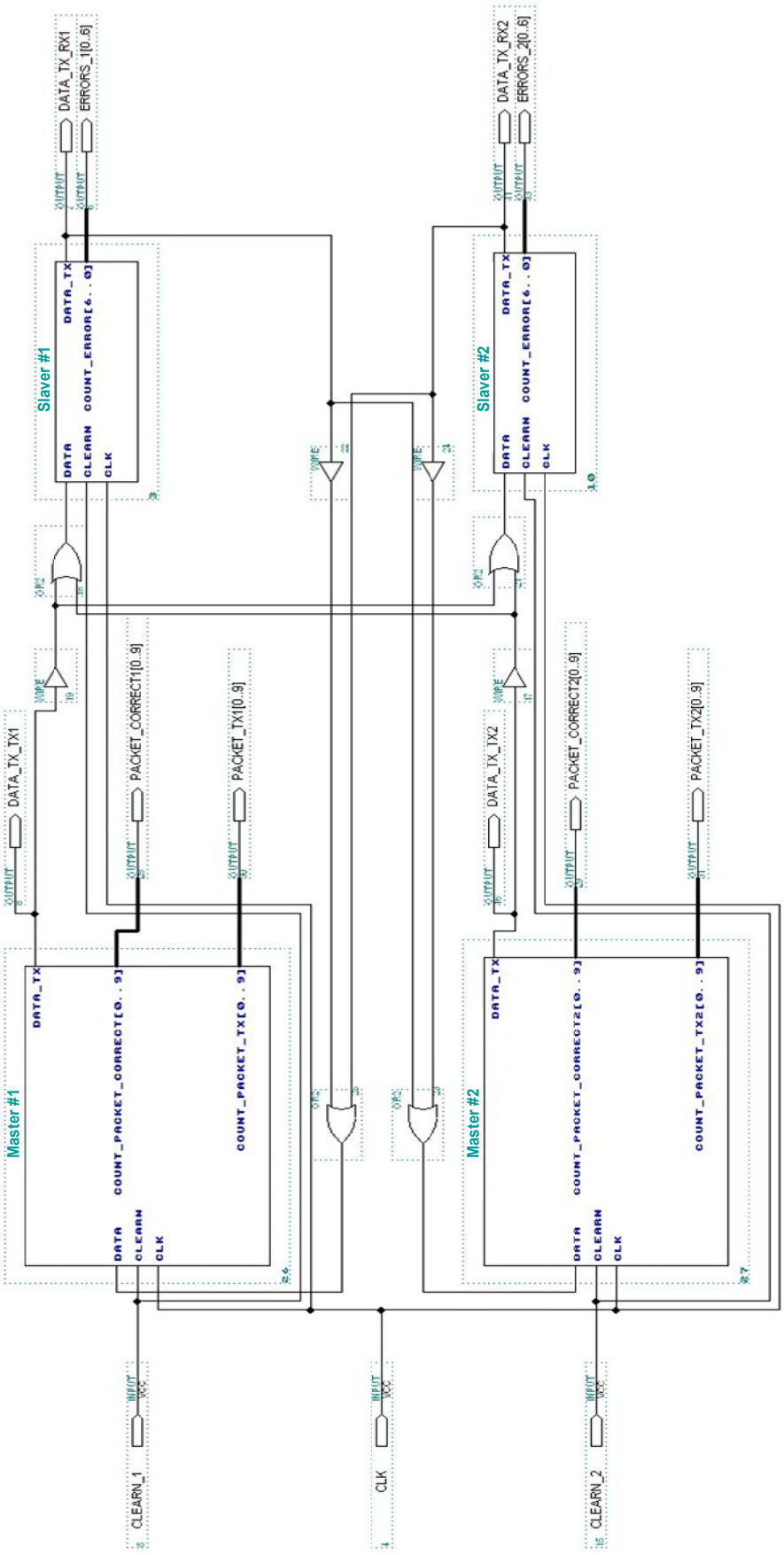
AI.17. Schematic of the 16 taps shift register



AI.18. Schematic of the Pseudo Random Number Generator



AI.19. Schematic of the optical system used in the software simulation



Annex II. Components's Specifications

Contents

AII.1 Technical specifications of the quad 2-input AND gate CD74AC08 202
AII.2 Technical specifications of the Infrared LED L7558 series 203
AII.3 Technical specifications of the Si PIN photodiode S6468 series..... 205
AII.4 Technical specifications of the Amplifier MAX410..... 207
AII.5 Technical specifications of the Differential Comparator LM360 222
AII.6 Technical specification of the 5.0V TTL Clock Oscillator F1100E 230
AII.7 Altera MAX 9000 programmable logic device family data sheet 231
AII.8 Altera device package information for EPM9320ALC84-10 239

AII.1. Technical specifications of the quad 2-input AND gate CD74AC08

:

TEXAS INSTRUMENTS

Data sheet acquired from Harris Semiconductor
SCHS226

September 1998

CD74AC08, CD74ACT08

Quad 2-Input AND Gate

Features

- Buffered Inputs
- Typical Propagation Delay
  - 4.3ns at V_{CC} = 5V, T_A = 25°C, C_L = 50pF
- Exceeds 2kV ESD Protection MIL-STD-883, Method 3015
- SCR-Latchup-Resistant CMOS Process and Circuit Design
- Speed of Bipolar FAST™/AS/S with Significantly Reduced Power Consumption
- Balanced Propagation Delays
- AC Types Feature 1.5V to 5.5V Operation and Balanced Noise Immunity at 30% of the Supply
- ±24mA Output Drive Current
  - Fanout to 15 FAST™ ICs
  - Drives 50Ω Transmission Lines

Description

The CD74AC08 and CD74ACT08 are quad 2-input AND gates that utilize the Harris Advanced CMOS Logic technology.

Ordering Information

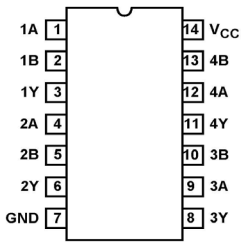
PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
CD74AC08E	-55 to 125	14 Ld PDIP	E14.3
CD74ACT08E	-55 to 125	14 Ld PDIP	E14.3
CD74AC08M	-55 to 125	14 Ld SOIC	M14.15
CD74ACT08M	-55 to 125	14 Ld SOIC	M14.15

NOTES:

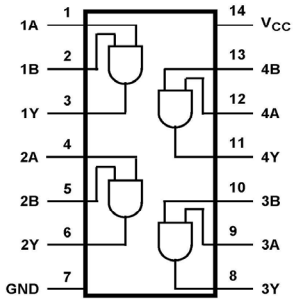
1. When ordering, use the entire part number. Add the suffix 96 to obtain the variant in the tape and reel.
2. Wafer and die for this part number is available which meets all electrical specifications. Please contact your local sales office or Harris customer service for ordering information.

Pinout

CD74AC08, CD74ACT08
(PDIP, SOIC)
TOP VIEW



Functional Diagram



TRUTH TABLE

INPUTS		OUTPUTS
nA	nB	nY
L	L	L
H	L	L
L	H	L
H	H	H

CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

FAST™ is a Trademark of Fairchild Semiconductor.

File Number **1950.1**

AII.2. Technical specifications of the Infrared LED L7558 series

LED

Infrared LED
L7558 series

High-speed, high-power infrared LED for spatial light transmission

L7558 series infrared LEDs were developed for spatial light transmission of high-density information such as image data signals, and operate at high speeds of 50 MHz.
L7558 delivers high output of 14 mW and is used in combination with a light projection lens that matches the application. L7558-01 is sealed in a metal package capped with a glass lens that ensures narrow directivity of ±7° (full angle at half maximum). Metal stem package gives L7558 and L7558-01 higher reliability than plastic package devices.

Features

- High-speed response: 50 MHz Typ. (I_F=50 mA)
- High radiant output power
  L7558 : 14 mW Typ. (I_F=50 mA)
  L7558-01: 7 mW Typ. (I_F=50 mA)
- High reliability

Applications

- Spatial light transmission

■ Absolute maximum ratings (Ta=25 °C)

Parameter	Symbol	Condition	Value	Unit
Forward current	I_F		100	mA
Reverse voltage	V_R		5	V
Pulse forward current	I_{FP}	Pulse width =10 µs Duty ratio =1 %	1.0	A
Operating temperature	Topr		-30 to +85	°C
Storage temperature	Tstg		-40 to +100 *	°C

* Guaranteed to resist temperature cycle test of up to 5 cycles.

■ Electrical and optical characteristics (Ta=25 °C)

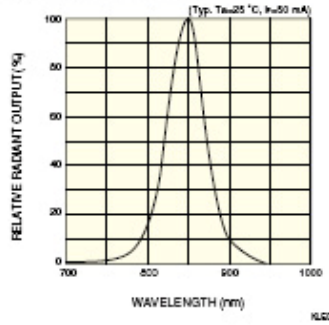
Parameter	Symbol	Condition	L7558			L7558-01			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Peak emission wavelength	λp	I_F=50 mA	820	850	880	820	850	880	nm
Spectral half width	Δλ	I_F=50 mA	-	50	-	-	50	-	nm
Forward voltage	V_F	I_F=50 mA	-	1.45	1.60	-	1.45	1.60	V
Pulse forward voltage	V_{FP}	I_F=1 A	-	3.4	4.3	-	3.4	4.3	V
Reverse current	I_R	V_R=5 V	-	-	10	-	-	10	µA
Radiant flux	φe	I_F=50 mA	11	14	-	5.5	7.0	-	mW
Radiant illuminance	P_E	I_F=50 mA	-	1.5	-	-	4.0	-	mW/cm²
Cut-off frequency	fc	I_F=50 mA ± 1mAp-p	35	50	-	35	50	-	MHz

SOLID STATE DIVISION

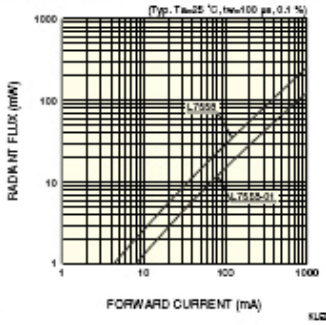
HAMAMATSU

Infrared LED L7558 series

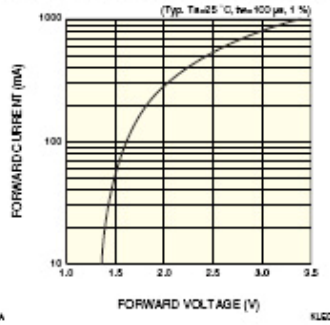
■ Emission spectrum



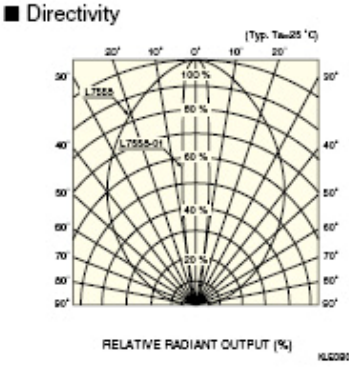
■ Radiant flux vs. forward current



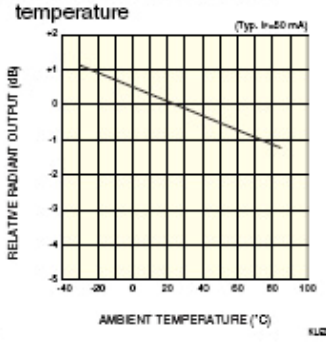
■ Forward current vs. forward voltage



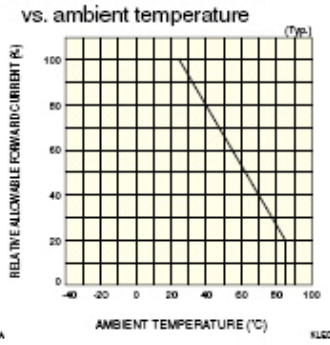
■ Directivity



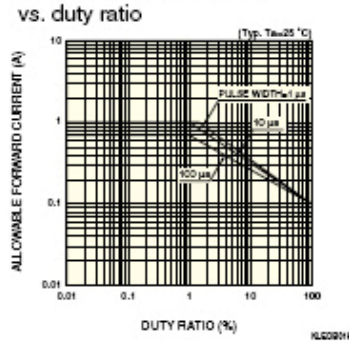
■ Radiant output vs. ambient temperature



■ Allowable forward current vs. ambient temperature

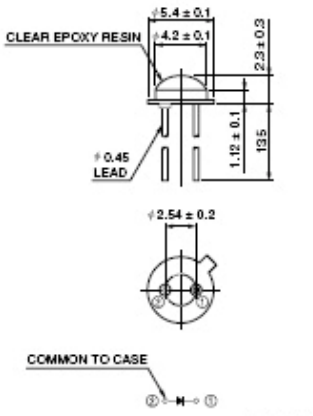


■ Allowable forward current vs. duty ratio

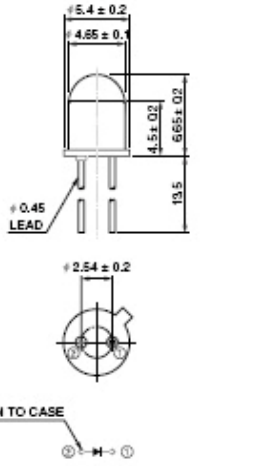


■ Dimensional outlines (unit: mm)

L7558



L7558-01



HAMAMATSU PHOTONICS K.K., Solid State Division
1126-1 Ichino-cho, Hamamatsu City, 435-8558 Japan, Telephone: (81) 053-434-3311, Fax: (81) 053-434-5184, http://www.hamamatsu.com
U.S.A.: Hamamatsu Corporation: 360 Foothill Road, P.O.Box 6910, Bridgewater, N.J. 08807-0910, U.S.A., Telephone: (1) 908-231-0960, Fax: (1) 908-231-1218
Germany: Hamamatsu Photonics Deutschland GmbH: Arzbergerstr. 10, D-82211 Herrsching am Ammersee, Germany, Telephone: (49) 08152-3750, Fax: (49) 08152-2658
France: Hamamatsu Photonics France S.A.R.L.: 8, Rue du Saule Trapu, Parc du Moulin de Massy, 91882 Massy Cedex, France, Telephone: 33-(1) 69 53 71 00, Fax: 33-(1) 69 53 71 10
United Kingdom: Hamamatsu Photonics UK Limited: 2 Howard Court, 10 Tewin Road, Welwyn Garden City, Hertfordshire AL7 1BW, United Kingdom, Telephone: (44) 1707-294888, Fax: (44) 1707-325777
North Europe: Hamamatsu Photonics Norden AB: Smidesvägen 12, SE-171 41 Solna, Sweden, Telephone: (46) 8-509-031-00, Fax: (46) 8-509-031-01
Italy: Hamamatsu Photonics Italia S.R.L.: Strada della Moia, 1/E, 20020 Arese, (Milano), Italy, Telephone: (39) 02-935-81-733, Fax: (39) 02-935-81-741

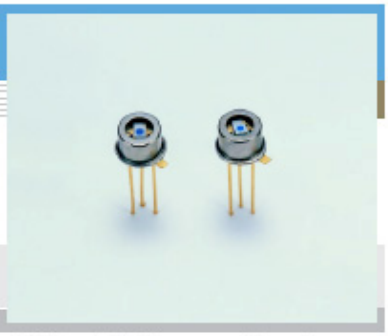
Cat. No. KLED1009E02
Apr. 2001 DN

AII.3. Technical specifications of the Si PIN photodiode S6468 series

PHOTODIODE

Si PIN photodiode with preamp
S6468 series

High-speed sensor with preamp



S6468 series is a high-speed photodetector consisting of a Si PIN photodiode and a preamplifier chip integrated in the same package. They feature high-speed response and high sensitivity over a wide spectral range from visible to near infrared light. The small package (TO-18) allows compact optical design. The amplifier input is at a virtual ground, so external noise which may appear when detecting high-speed signals can be suppressed.

Features

- Cut-off frequency (Vcc=5 V)
  S6468 : 15 MHz
  S6468-02: 35 MHz
- Low noise (f=1 MHz)
  S6468 : 25 nVrms/$Hz^{1/2}$
  S6468-02: 28 nVrms/$Hz^{1/2}$
- 3 pin TO-18 package
- Active area: ϕ0.8 mm

Applications

- Optical fiber communication
- Video signal transmission
- Optical disk pick-up

■ Electrical and optical characteristics [Ta=25 °C, Vcc=5 V, RL=500 Ω, CL=13 pF] *1

Parameter	Symbol	Condition	S6468			S6468-02			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Spectral response range	λ			320 to 1060			320 to 1000		nm
Peak sensitivity wavelength	λp		-	900	-	-	800	-	nm
Photo sensitivity	S	λ=660 nm	-	13.5	-	-	8.5	-	mV/μW
		λ=780 nm	-	15.5	-	-	11	-	
		λ=830 nm	-	16.5	-	-	11	-	
Trans-impedance	RT		-	30	-	-	20	-	kΩ
Power supply current	Icc	RL=∞	-	-	3	-	-	3	mA
Output bias voltage *2	Vo	RL=∞ Pin=0 μW	0.55	0.65	0.8	0.65	0.8	0.9	V
Temperature coefficient of output bias voltage	-		-	-2	-	-	-2	-	mV/°C
Cut-off frequency	fc	Pin=10 μW *3	12	15	-	28	35	-	MHz
Maximum output voltage amplitude	-	Nonlinear distortion: 10 % Max.	0.5	-	-	0.5	-	-	Vp-p
Output impedance	Zo	f=5 Hz	-	30	-	-	30	-	Ω
Output noise voltage	Vn	Pin=0 μW f=1 MHz	-	25	-	-	28	-	nV/$Hz^{1/2}$
Overshoot	-	Pin=10 μW *3	-	-	10	-	-	10	%

■ Absolute maximum ratings

Parameter	Symbol	Min.	Max.	Unit
Power supply voltage *4	Vcc	-0.5	7	V
Power dissipation	P	-	300	mW
Operating temperature	Topr	-20	70	°C
Storage temperature	Tstg	-40	100	°C

■ Recommended operating conditions

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	Vcc	4.75	5	5.25	V
Load resistance	RL	500	-	-	Ω
Load capacitance	CL	-	-	13	pF
Operating temperature	Topr	0	-	60	°C

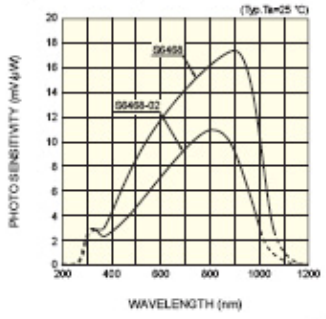
*1: For definitions of RL and CL, refer to the basic connection.
*2: Output voltage Vout =Vo-(Pin × S) Pin: incident radiant flux (μW)
*3: Peak value
*4: A bypass capacitor (0.01 μF to 0.1 μF ceramic) is connected between the Vcc lead and the GND lead. The lead length should be less than 20 mm.

SOLID STATE DIVISION

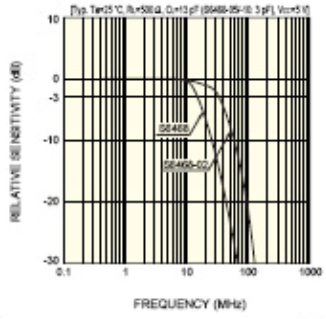
HAMAMATSU

Si PIN photodiode with preamp S6468 series

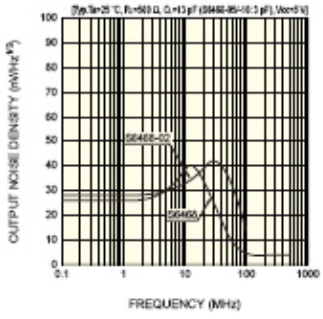
■ Spectral response



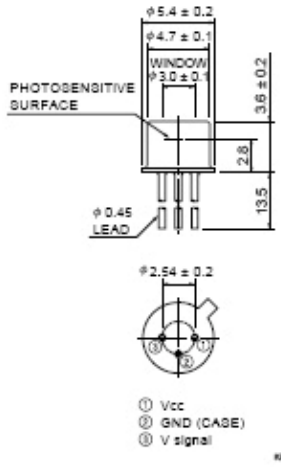
■ Frequency characteristics



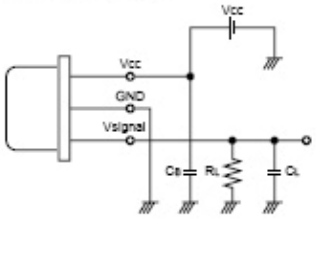
■ Output noise spectrum



■ Dimensional outline (unit: mm)



■ Basic connection



Precautions for use

● ESD

S6468 series may be damaged or their performance may deteriorate by such factors as electro static discharge from the human body, surge voltages from measurement equipment, leakage voltages from soldering irons and packing materials, etc. As a countermeasure against electro static discharge, the device, operator, work place and measuring jigs must all be set at the same potential. The following precautions must be observed during use:

- To protect the device from electro static discharge which accumulate on the operator or the operator's clothes, use a wrist strap or similar tools to ground the operator's body via a high impedance resistor (1 MΩ).
- A semiconductive sheet (1 MΩ to 100 MΩ) should be laid on both the work table and the floor in the work area.
- When soldering, use an electrically grounded soldering iron with an isolation resistance of more than 10 MΩ.
- For containers and packing, use of a conductive material or aluminum foil is effective. When using an antistatic material, use one with a resistance of 0.1 MΩ/cm² to 1 GΩ/cm².

● Wiring

- R_L and C_L are total resistive load and capacitive load viewed from the V signal terminal. When connecting a cable or circuit to the latter stage of the basic connection diagram, the cable or circuit resistance and capacitance should also be taken into account. They should be used in accordance with the recommended operating conditions: $R_L \geq 500\ \Omega$ and $C_L \leq 13$ pF.
- A bypass capacitor (C_B=0.01 μF to 0.1 μF ceramic) is connected between the Vcc lead and the GND lead.
- The lead length should be less than 20 mm.
- If electric current or voltage is applied in reverse polarity to an electronic device such as a preamplifier, this can degrade device performance or destroy the device. Always check the wiring and dimensional outline to avoid misconnection.

HAMAMATSU PHOTONICS K.K., Solid State Division
1126-1 Ichino-cho, Hamamatsu City, 435-8558 Japan, Telephone: (81) 053-434-3311, Fax: (81) 053-434-5184, http://www.hamamatsu.com
U.S.A.: Hamamatsu Corporation: 360 Foothill Road, P.O.Box 6910, Bridgewater, N.J. 08807-0910, U.S.A., Telephone: (1) 908-231-0960, Fax: (1) 908-231-1218
Germany: Hamamatsu Photonics Deutschland GmbH: Arzbergerstr. 10, D-82211 Herrsching am Ammersee, Germany, Telephone: (49) 08152-3750, Fax: (49) 08152-2658
France: Hamamatsu Photonics France S.A.R.L.: 8, Rue du Saule Trapu, Parc du Moulin de Massy, 91882 Massy Cedex, France, Telephone: 33-(1) 69 53 71 00, Fax: 33-(1) 69 53 71 10
United Kingdom: Hamamatsu Photonics UK Limited: 2 Howard Court, 10 Tewin Road, Welwyn Garden City, Hertfordshire AL7 1BW, United Kingdom, Telephone: (44) 1707-294888, Fax: (44) 1707-325777
North Europe: Hamamatsu Photonics Norden AB: Smidesvägen 12, SE-171 41 Solna, Sweden, Telephone: (46) 8-509-031-00, Fax: (46) 8-509-031-01
Italy: Hamamatsu Photonics Italia S.R.L.: Strada della Moia, 1/E, 20020 Arese, (Milano), Italy, Telephone: (39) 02-935-81-733, Fax: (39) 02-935-81-741

Cat. No. KPIN1026E05
Aug. 2003 DN

AII.4. Technical specifications of the Amplifier MAX410

19-4194; Rev 4; 6/03

MAXIM

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

General Description

The MAX410/MAX412/MAX414 single/dual/quad op amps set a new standard for noise performance in high-speed, low-voltage systems. Input voltage-noise density is guaranteed to be less than 2.4nV/√Hz at 1kHz. A unique design not only combines low noise with ±5V operation, but also consumes 2.5mA supply current per amplifier. Low-voltage operation is guaranteed with an output voltage swing of 7.3VP-P into 2kΩ from ±5V supplies. The MAX410/MAX412/MAX414 also operate from supply voltages between ±2.4V and ±5V for greater supply flexibility.

Unity-gain stability, 28MHz bandwidth, and 4.5V/µs slew rate ensure low-noise performance in a wide variety of wideband and measurement applications. The MAX410/MAX412/MAX414 are available in DIP and SO packages in the industry-standard single/dual/quad op amp pin configurations. The single comes in an ultra-small TDFN package (3mm x 3mm).

Applications

Low-Noise Frequency Synthesizers
Infrared Detectors
High-Quality Audio Amplifiers
Ultra Low-Noise Instrumentation Amplifiers
Bridge Signal Conditioning

Features

- ♦ Voltage Noise: 2.4nV/√Hz (max) at 1kHz
- ♦ 2.5mA Supply Current Per Amplifier
- ♦ Low Supply Voltage Operation: ±2.4V to ±5V
- ♦ 28MHz Unity-Gain Bandwidth
- ♦ 4.5V/µs Slew Rate
- ♦ 250µV (max) Offset Voltage (MAX410/MAX412)
- ♦ 115dB (min) Voltage Gain
- ♦ Available in an Ultra-Small TDFN Package

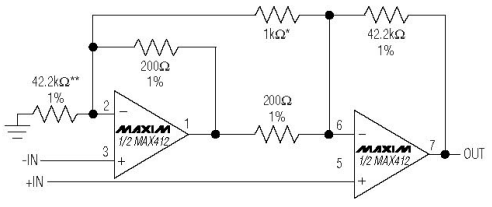
Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX410CPA	0°C to +70°C	8 Plastic DIP
MAX410BCPA	0°C to +70°C	8 Plastic DIP
MAX410CSA	0°C to +70°C	8 SO
MAX410BCSA	0°C to +70°C	8 SO
MAX410EPA	-40°C to +85°C	8 Plastic DIP
MAX410BEPA	-40°C to +85°C	8 Plastic DIP
MAX410ESA	-40°C to +85°C	8 SO
MAX410BESA	-40°C to +85°C	8 SO
MAX410ETA	-40°C to +85°C	8 TDFN-EP*

Ordering Information continued at end of data sheet.

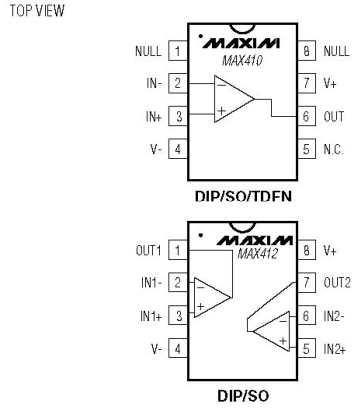
**EP—Exposed paddle. Top Mark—AGQ.*

Typical Operating Circuit



LOW-NOISE INSTRUMENTATION AMPLIFIER

Pin Configurations



Pin Configurations continued at end of data sheet.

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

ABSOLUTE MAXIMUM RATINGS

Supply Voltage 12V
Differential Input Current (Note 1) .....±20mA
Input Voltage Range.....V+ to V-
Common-Mode Input Voltage .....(V+ + 0.3V) to (V- - 0.3V)
Short-Circuit Current Duration.....Continuous
Continuous Power Dissipation (T_A = +70°C)
MAX410/MAX412
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C) ...727mW
8-Pin SO (derate 5.88mW/°C above +70°C).....471mW
8-Pin TDFN (derate 24.4mW/°C above +70°C) 1951mW
MAX414
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)800mW
14-Pin SO (derate 8.33mW/°C above +70°C).....667mW
Operating Temperature Ranges:
MAX41_C_ _ .....0°C to +70°C
MAX41_E_ _.....-40°C to +85°C
Storage Temperature Range .....-65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C

Note 1: The amplifier inputs are connected by internal back-to-back clamp diodes. In order to minimize noise in the input stage, current-limiting resistors are not used. If differential input voltages exceeding ±1.0V are applied, limit input current to 20mA.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V+ = 5V, V- = -5V, **T_A = +25°C**, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Input Offset Voltage	V_{OS}	MAX410, MAX410B, MAX412, MAX412B			±120	±250	µV
		MAX414, MAX414B			±150	±320	
Input Bias Current	I_B				±80	±150	nA
Input Offset Current	I_{OS}				±40	±80	nA
Differential Input Resistance	$R_{IN(Diff)}$				20		kΩ
Common-Mode Input Resistance	$R_{IN(CM)}$				40		MΩ
Input Capacitance	C_{IN}				4		pF
Input Noise-Voltage Density	e_n	MAX410, MAX412, MAX414	10Hz		7		nV/√Hz
			1000Hz (Note 2)		1.5	2.4	
		MAX410B, MAX412B, MAX414B	1000Hz (Note 2)		2.4	4.0	
Input Noise-Current Density	i_n	f_O = 10Hz			2.6		pA/√Hz
		f_O = 1000Hz			1.2		
Common-Mode Input Voltage	V_{CM}			±3.5	+3.7/ -3.8		V
Common-Mode Rejection Ratio	CMRR	V_{CM} = ±3.5V		115	130		dB
Power-Supply Rejection Ratio	PSRR	V_S = ±2.4V to ±5.25V		96	103		dB
Large-Signal Gain	A_{VOL}	R_L = 2kΩ, V_O = ±3.6V		115	122		dB
		R_L = 600Ω, V_O = ±3.5V		110	120		
Output Voltage Swing	V_{OUT}	R_L = 2kΩ		+3.6 -3.7	+3.7/ -3.8		V
Short-Circuit Output Current	I_{SC}				35		mA
Slew Rate	SR	10kΩ ll 20pF load			4.5		V/µs
Unity-Gain Bandwidth	GBW	10kΩ ll 20pF load			28		MHz
Settling Time	t_S	To 0.1%			1.3		µs
Channel Separation	C_S	f_O = 1kHz			135		dB

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

ELECTRICAL CHARACTERISTICS (continued)

(V+ = 5V, V- = -5V, **T_A = +25°C**, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Supply-Voltage Range	V_S		±2.4		±5.25	V
Supply Current	I_S	Per amplifier		2.5	2.7	mA

ELECTRICAL CHARACTERISTICS

(V+ = 5V, V- = -5V, **T_A = 0°C to +70°C**, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Offset Voltage	V_{OS}			±150	±350	µV
Offset Voltage Tempco	$\Delta V_{OS}/\Delta T$	Over operating temperature range		±1		µV/°C
Input Bias Current	I_B			±100	±200	nA
Input Offset Current	I_{OS}			±80	±150	nA
Common-Mode Input Voltage	V_{CM}		±3.5	+3.7/ -3.8		V
Common-Mode Rejection Ratio	CMRR	V_{CM} = ±3.5V	105	121		dB
Power-Supply Rejection Ratio	PSRR	V_S = ±2.4V to ±5.25V	90	97		dB
Large-Signal Gain	A_{VOL}	R_L = 2kΩ, V_O = ±3.6V	110	120		dB
		R_L = 600Ω, V_O = ±3.5V	90	119		
Output Voltage Swing	V_{OUT}	R_L = 2kΩ	±3.5	+3.7/ -3.6		V
Supply Current	I_S	Per amplifier			3.3	mA

ELECTRICAL CHARACTERISTICS

(V+ = 5V, V- = -5V, **T_A = -40°C to +85°C**, unless otherwise noted.) (Note 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Offset Voltage	V_{OS}	MAX410, MAX410B, MAX412, MAX412B		±200	±400	µV
		MAX414, MAX414B		±200	±450	
Offset Voltage Tempco	$\Delta V_{OS}/\Delta T$	Over operating temperature range		±1		µV/°C
Input Bias Current	I_B			±130	±350	nA
Input Offset Current	I_{OS}			±100	±200	nA
Common-Mode Input Voltage	V_{CM}		±3.5	+3.7/ -3.6		V
Common-Mode Rejection Ratio	CMRR	V_{CM} = ±3.5V	105	120		dB
Power-Supply Rejection Ratio	PSRR	V_S = ±2.4V to ±5.25V	90	94		dB
Large-Signal Gain	A_{VOL}	R_L = 2kΩ, V_O = ±3.6V	110	118		dB
		R_L = 600Ω, V_O = +3.4V to -3.5V	90	114		
Output Voltage Swing	V_{OUT}	R_L = 2kΩ	±3.5	+3.7/ -3.6		V
Supply Current	I_S	Per amplifier			3.3	mA

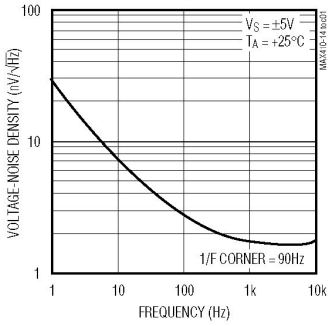
Note 2: Guaranteed by design.
Note 3: All TDFN devices are 100% tested at T_A = +25°C. Limits over temperature for thin TDFNs are guaranteed by design.

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

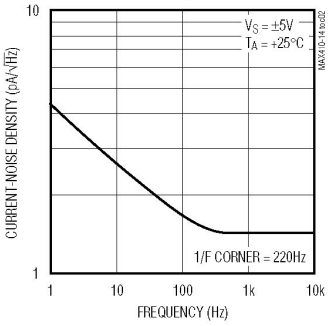
Typical Operating Characteristics

(V+ = 5V, V- = -5V, T_A = +25°C, unless otherwise noted.)

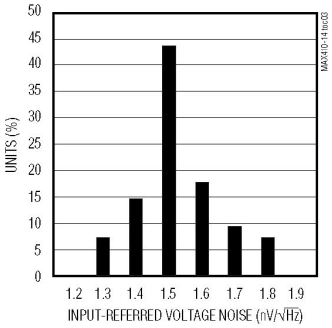
VOLTAGE-NOISE DENSITY vs. FREQUENCY



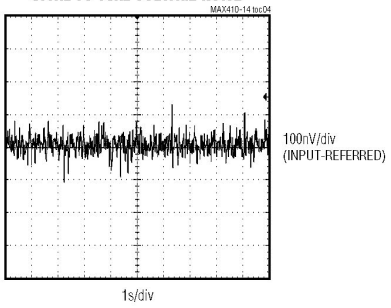
CURRENT-NOISE DENSITY vs. FREQUENCY



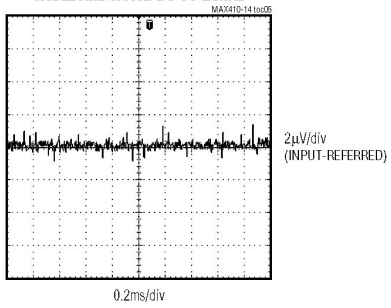
1kHz VOLTAGE NOISE DISTRIBUTION



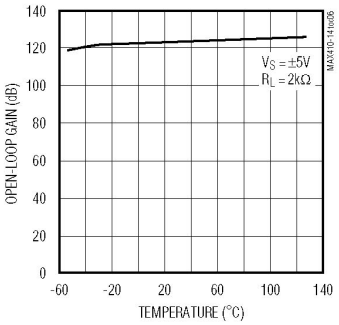
0.1Hz TO 10Hz VOLTAGE NOISE



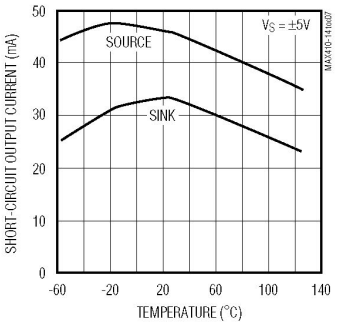
WIDEBAND NOISE DC TO 20kHz



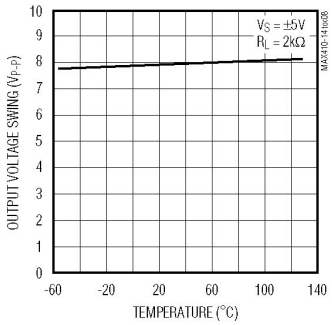
OPEN-LOOP GAIN vs. TEMPERATURE



SHORT-CIRCUIT OUTPUT CURRENT vs. TEMPERATURE



OUTPUT VOLTAGE SWING vs. TEMPERATURE

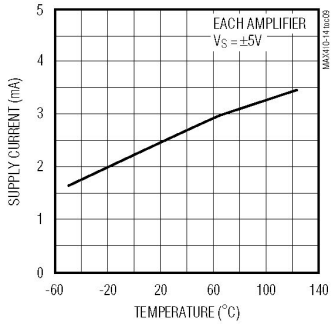


Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

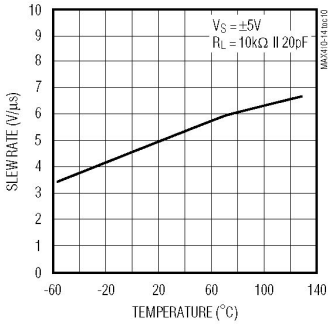
Typical Operating Characteristics (continued)

(V+ = 5V, V- = -5V, T_A = +25°C, unless otherwise noted.)

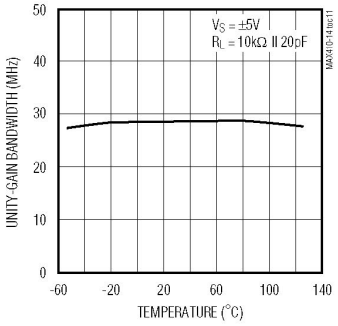
SUPPLY CURRENT vs. TEMPERATURE



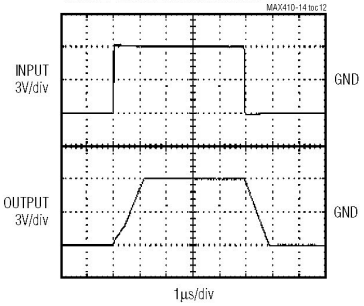
SLEW RATE vs. TEMPERATURE



UNITY-GAIN BANDWIDTH vs. TEMPERATURE

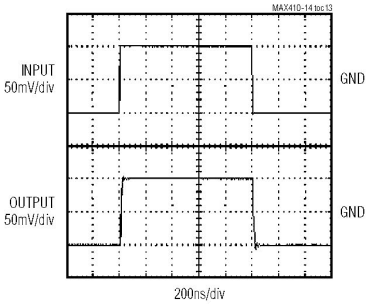


LARGE-SIGNAL TRANSIENT RESPONSE



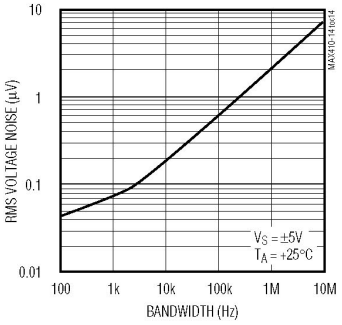
AV = +1, RF = 499Ω, RL = 2kΩ II 20pF, VS = ±5V, TA = +25°C

SMALL-SIGNAL TRANSIENT RESPONSE

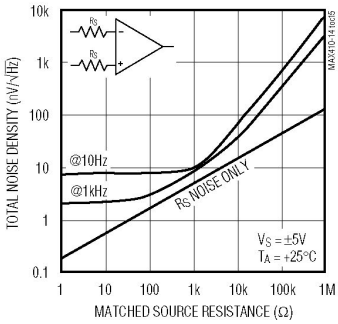


AV = +1, RF = 499Ω, RL = 2kΩ II 20pF, VS = ±5V, TA = +25°C

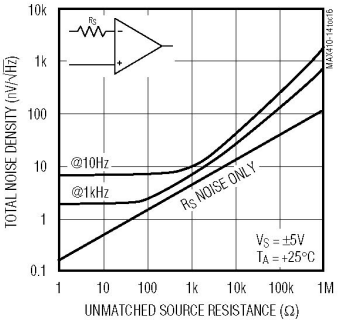
WIDEBAND VOLTAGE NOISE (0.1Hz TO FREQUENCY INDICATED)



TOTAL NOISE DENSITY vs. MATCHED SOURCE RESISTANCE



TOTAL NOISE DENSITY vs. UNMATCHED SOURCE RESISTANCE

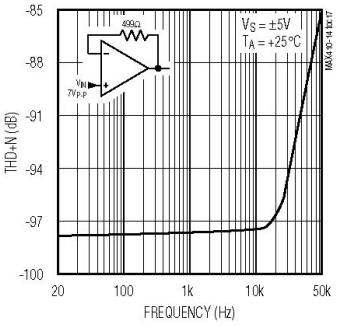


Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

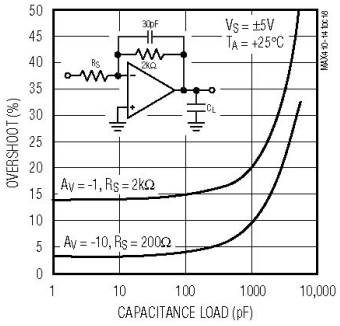
Typical Operating Characteristics (continued)

(V+ = 5V, V- = -5V, T_A = +25°C, unless otherwise noted.)

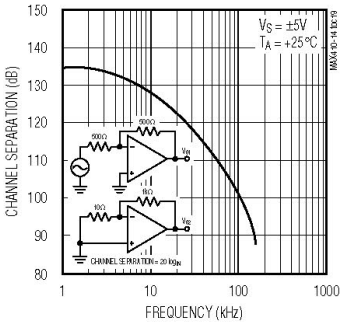
TOTAL HARMONIC DISTORTION PLUS NOISE vs. FREQUENCY



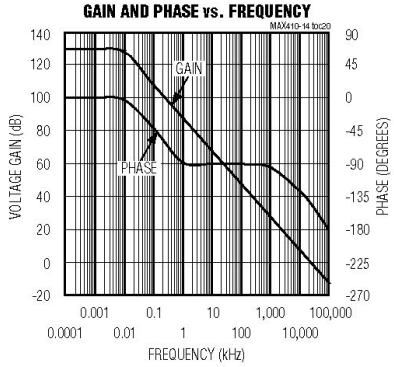
PERCENTAGE OVERSHOOT vs. CAPACITIVE LOAD



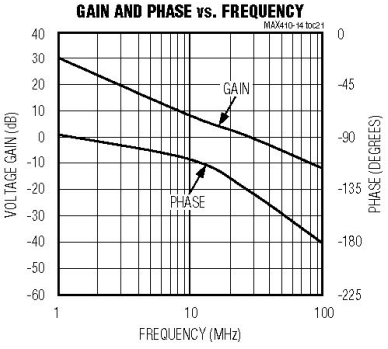
MAX412/MAX414 CHANNEL SEPARATION vs. FREQUENCY



GAIN AND PHASE vs. FREQUENCY



GAIN AND PHASE vs. FREQUENCY



Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Applications Information

The MAX410/MAX412/MAX414 provide low voltage-noise performance. Obtaining low voltage noise from a bipolar op amp requires high collector currents in the input stage, since voltage noise is inversely proportional to the square root of the input stage collector current. However, op amp current noise is proportional to the square root of the input stage collector current, and the input bias current is proportional to the input stage collector current. Therefore, to obtain optimum low-noise performance, DC accuracy, and AC stability, minimize the value of the feedback and source resistance.

Total Noise Density vs. Source Resistance

The standard expression for the total input-referred noise of an op amp at a given frequency is:

$$e_t = \sqrt{e_n^2 + (R_p + R_n)^2\, i_n^2 + 4kT\,(R_p + R_n)}$$

where:

R_n = Inverting input effective series resistance

R_p = Noninverting input effective series resistance

e_n = Input voltage-noise density at the frequency of interest

i_n = Input current-noise density at the frequency of interest

T = Ambient temperature in Kelvin (K)

k = 1.28 x 10^{-23} J/K (Boltzman's constant)

In Figure 1, R_p = R3 and R_n = R1 ll R2. In a real application, the output resistance of the source driving the input must be included with R_p and R_n. The following example demonstrates how to calculate the total output-noise density at a frequency of 1kHz for the MAX412 circuit in Figure 1.

Gain = 1000

4kT at +25°C = 1.64 x 10^{-20}

R_p = 100Ω

R_n = 100Ω ll 100kΩ = 99.9 W

e_n = 1.5nV/√Hz at 1kHz

i_n = 1.2pA/√Hz at 1kHz

e_t = [(1.5 x 10^{-9})^2 + (100 + 99.9)^2 (1.2 x 10^{-12})^2 + (1.64 x 10^{-20}) (100 + 99.9)]$^{1/2}$ = 2.36nV/√Hz at 1kHz

Output noise density = (100)e_t = 2.36µV/√Hz at 1kHz.

In general, the amplifier's voltage noise dominates with equivalent source resistances less than 200Ω. As the equivalent source resistance increases, resistor noise becomes the dominant term, eventually making the voltage noise contribution from the MAX410/MAX412/MAX414 negligible. As the source resistance is further increased, current noise becomes dominant. For example, when the equivalent source resistance is greater than 3kΩ at 1kHz, the current noise component is larger than the resistor noise. The graph of Total Noise Density vs. Matched Source Resistance in the *Typical Operating Characteristics* shows this phenomenon. Optimal MAX410/MAX412/MAX414 noise performance and minimal total noise achieved with an equivalent source resistance of less than 10kΩ.

Voltage Noise Testing

RMS voltage-noise density is measured with the circuit shown in Figure 2, using the Quan Tech model 5173 noise analyzer, or equivalent. The voltage-noise density at 1kHz is sample tested on production units. When measuring op-amp voltage noise, only low-value, metal film resistors are used in the test fixture.

The 0.1Hz to 10Hz peak-to-peak noise of the MAX410/MAX412/MAX414 is measured using the test

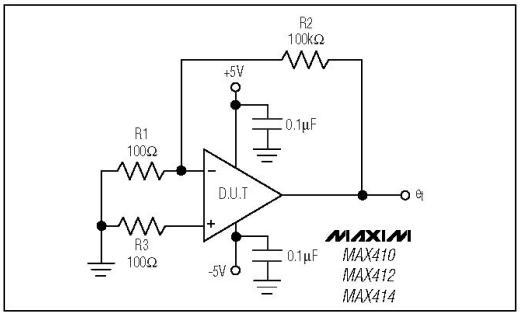


Figure 1. Total Noise vs. Source Resistance Example

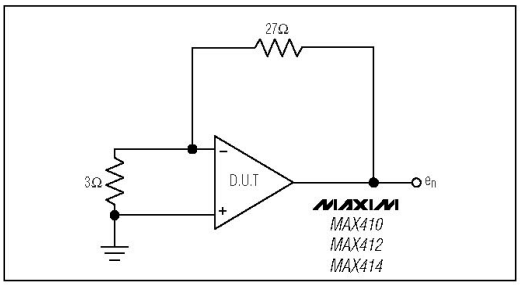


Figure 2. Voltage-Noise Density Test Circuit

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

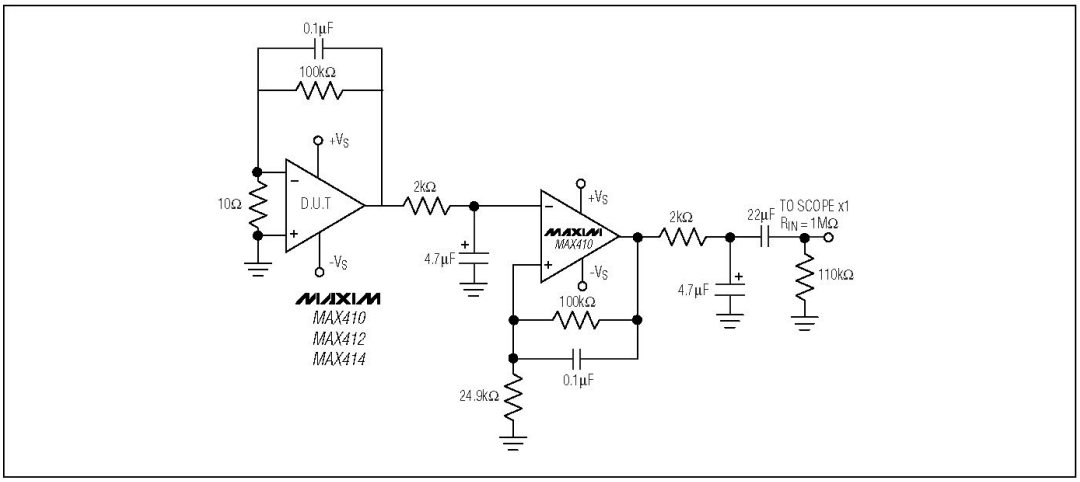


Figure 3. 0.1Hz to 10Hz Voltage Noise Test Circuit

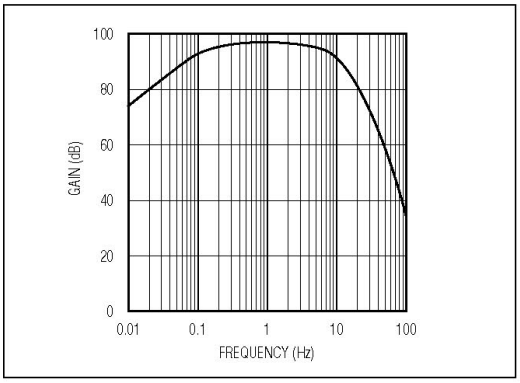


Figure 4. 0.1Hz to 10Hz Voltage Noise Test Circuit, Frequency Response

circuit shown in Figure 3. Figure 4 shows the frequency response of the circuit. The test time for the 0.1Hz to 10Hz noise measurement should be limited to 10 seconds, which has the effect of adding a second zero to the test circuit, providing increased attenuation for frequencies below 0.1Hz.

Current Noise Testing

The current-noise density can be calculated, once the value of the input-referred noise is determined, by using the standard expression given below:

$$i_n = \frac{\sqrt{e_{no}^2 - \left[(A_{VCL})^2(4kT)(R_n + R_p)\right]}}{(R_n + R_p)(A_{VCL})} \; A/\sqrt{Hz}$$

where:

R_n = Inverting input effective series resistance

R_p= Noninverting input effective series resistance

e_{no} = Output voltage-noise density at the frequency of interest ($V/\sqrt{Hz}$)

i_n = Input current-noise density at the frequency of interest ($A/\sqrt{Hz}$)

A_{VCL} = Closed-loop gain

T = Ambient temperature in Kelvin (K)

k = 1.38 x 10^{-23} J/K (Boltzman's constant)

R_p and R_n include the resistances of the input driving source(s), if any.

If the Quan Tech model 5173 is used, then the A_{VCL} terms in the numerator and denominator of the equation given above should be eliminated because the Quan

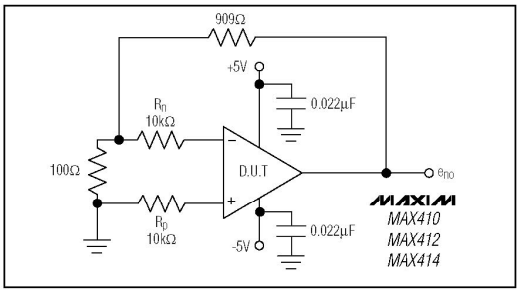


Figure 5. Current-Noise Test Circuit

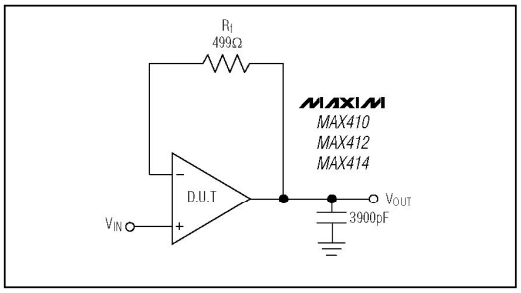


Figure 6a. Voltage Follower Circuit with 3900pF Load

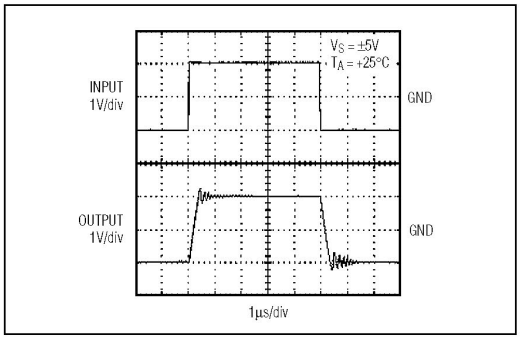


Figure 6b. Driving 3900pF Load as Shown in Figure 6a

Tech measures input-referred noise. For the circuit in Figure 5, assuming R_p is approximately equal to R_n and the measurement is taken with the Quan Tech model 5173, the equation simplifies to:

$$i_n = \frac{\sqrt{e_{no}^2 - \left[(1.64 \times 10^{-20})(20 \times 10^3)\right]}}{(20 \times 10^3)} \; A/\sqrt{Hz}$$

Input Protection

To protect amplifier inputs from excessive differential input voltages, most modern op amps contain input protection diodes and current-limiting resistors. These resistors increase the amplifier's input-referred noise. They have not been included in the MAX410/MAX412/MAX414, to optimize noise performance. The MAX410/MAX412/MAX414 do contain back-to-back input protection diodes which will protect the amplifier for differential input voltages of ±0.1V. If the amplifier must be protected from higher differential input voltages, add external current-limiting resistors in series with the op amp inputs to limit the potential input current to less than 20mA.

Capacitive-Load Driving

Driving large capacitive loads increases the likelihood of oscillation in amplifier circuits. This is especially true for circuits with high loop gains, like voltage followers. The output impedance of the amplifier and a capacitive load form an RC network that adds a pole to the loop response. If the pole frequency is low enough, as when driving a large capacitive load, the circuit phase margin is degraded.

In voltage follower circuits, the MAX410/MAX412/MAX414 remain stable while driving capacitive loads as great as 3900pF (see Figures 6a and 6b).

When driving capacitive loads greater than 3900pF, add an output isolation resistor to the voltage follower circuit, as shown in Figure 7a. This resistor isolates the load capacitance from the amplifier output and restores the phase margin. Figure 7b is a photograph of the response of a MAX410/MAX412/MAX414 driving a 0.015µF load with a 10Ω isolation resistor

The capacitive-load driving performance of the MAX410/MAX412/MAX414 is plotted for closed-loop gains of -1V/V and -10V/V in the % Overshoot vs. Capacitive Load graph in the *Typical Operating Characteristics*.

Feedback around the isolation resistor RI increases the accuracy at the capacitively loaded output (see Figure 8). The MAX410/MAX412/MAX414 are stable with a 0.01µF load for the values of R_I and C_F shown. In general, for decreased closed-loop gain, increase R_I or C_F. To drive larger capacitive loads, increase the value of C_F.

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

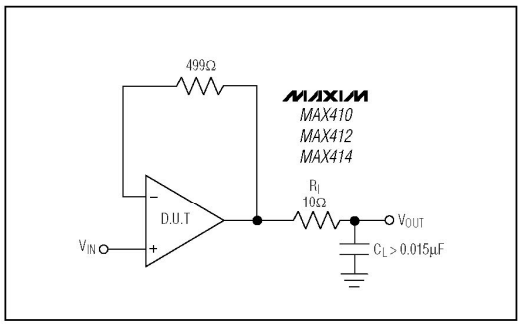


Figure 7a. Capacitive-Load Driving Circuit

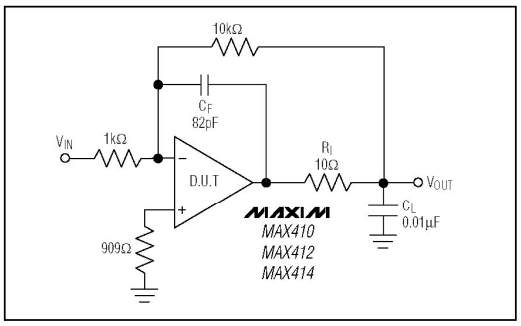


Figure 8. Capacitive-Load Driving Circuit with Loop-Enclosed Isolation Resistor

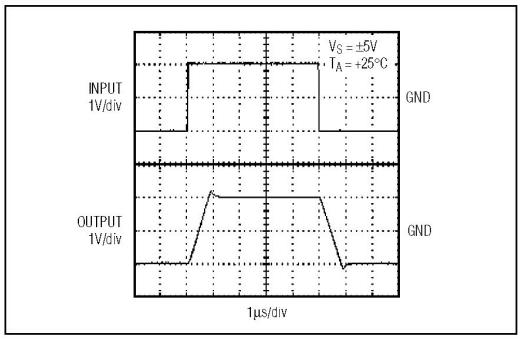


Figure 7b. Driving a 0.015µF Load with a 10Ω Isolation Resistor

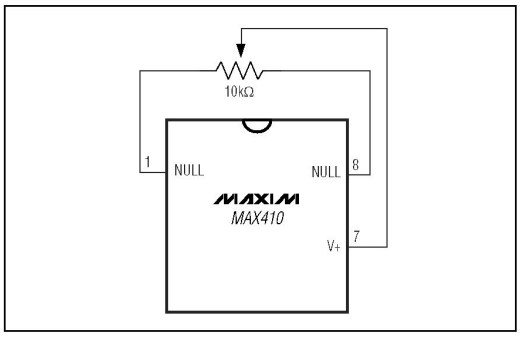


Figure 9. MAX410 Offset Null Circuit

TDFN Exposed Paddle Connection

On TDFN packages, there is an exposed paddle that does not carry any current but should be connected to V- (not the GND plane) for rated power dissipation.

Total Supply Voltage Considerations

Although the MAX410/MAX412/MAX414 are specified with ±5V power supplies, they are also capable of single-supply operation with voltages as low as 4.8V. The minimum input voltage range for normal amplifier operation is between V- + 1.5V and V+ - 1.5V. The minimum room-temperature output voltage range (with 2kΩ load) is between V+ - 1.4V and V- + 1.3V for total supply voltages between 4.8V and 10V. The output voltage range, referenced to the supply voltages, decreases slightly over temperature, as indicated in the ±5V *Electrical Characteristics* tables. Operating characteristics at total supply, voltages of less than 10V are guaranteed by design and PSRR tests.

MAX410 Offset Voltage Null

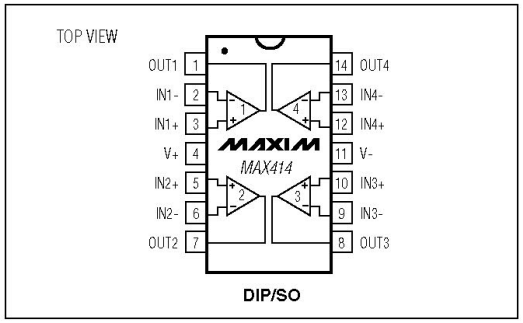
The offset null circuit of Figure 9 provides approximately ±450µV of offset adjustment range, sufficient for zeroing offset over the full operating temperature range,

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Ordering Information (continued)

PART	TEMP RANGE	PIN-PACKAGE
MAX412CPA	0°C to +70°C	8 Plastic DIP
MAX412BCPA	0°C to +70°C	8 Plastic DIP
MAX412CSA	0°C to +70°C	8 SO
MAX412BCSA	0°C to +70°C	8 SO
MAX412EPA	-40°C to +85°C	8 Plastic DIP
MAX412BEPA	-40°C to +85°C	8 Plastic DIP
MAX412ESA	-40°C to +85°C	8 SO
MAX412BESA	-40°C to +85°C	8 SO
MAX414CPD	0°C to +70°C	14 Plastic DIP
MAX414BCPD	0°C to +70°C	14 Plastic DIP
MAX414CSD	0°C to +70°C	14 SO
MAX414BCSD	0°C to +70°C	14 SO
MAX414EPD	-40°C to +85°C	14 Plastic DIP
MAX414BEPD	-40°C to +85°C	14 Plastic DIP
MAX414ESD	-40°C to +85°C	14 SO
MAX414BESD	-40°C to +85°C	14 SO

Pin Configurations (continued)



Chip Information

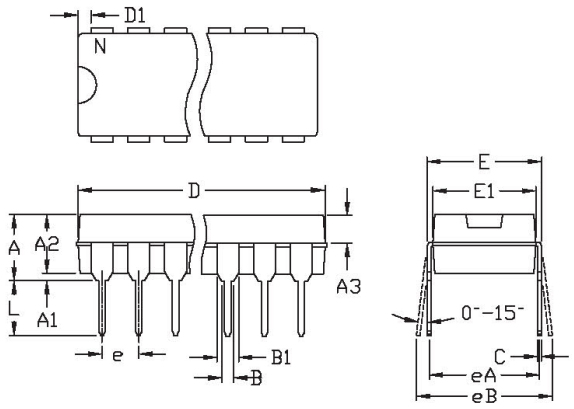
MAX410 TRANSISTOR COUNT: 132
MAX412 TRANSISTOR COUNT: 262
MAX414 TRANSISTOR COUNT: 2 x 262 (hybrid)
PROCESS: Bipolar

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Package Information

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)

PDIPN.EPS



	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	---	0.180	---	4.572
A1	0.015	---	0.38	---
A2	0.125	0.175	3.18	4.45
A3	0.055	0.080	1.40	2.03
B	0.015	0.022	0.381	0.56
B1	0.045	0.065	1.14	1.65
C	0.008	0.014	0.2	0.355
D1	0.005	0.080	0.13	2.03
E	0.300	0.325	7.62	8.26
E1	0.240	0.310	6.10	7.87
e	0.100 BSC.		2.54 BSC.	
eA	0.300 BSC.		7.62 BSC.	
eB	0.400 BSC.		10.16 BSC.	
L	0.115	0.150	2.921	3.81

	INCHES		MILLIMETERS			
	MIN	MAX	MIN	MAX	N	MS001
D	0.348	0.390	8.84	9.91	8	AB
D	0.735	0.765	18.67	19.43	14	AC
D	0.745	0.765	18.92	19.43	16	AA
D	0.885	0.915	22.48	23.24	18	AD
D	1.015	1.045	25.78	26.54	20	AE
D	1.14	1.265	28.96	32.13	24	AF
D	1.360	1.380	34.54	35.05	28	*5

NOTES:
1. D&E DO NOT INCLUDE MOLD FLASH
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED .15mm (.006")
3. CONTROLLING DIMENSION: MILLIMETER
4. MEETS JEDEC MS001-XX AS SHOWN IN ABOVE TABLE
5. SIMILIAR TO JEDEC MO-058AB
6. N = NUMBER OF PINS

MAXIM — PROPRIETARY INFORMATION

TITLE: PACKAGE FAMILY OUTLINE: PDIP .300" — 1/1

DOCUMENT CONTROL NUMBER: 21-0043 D — REV.

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Package Information (continued)

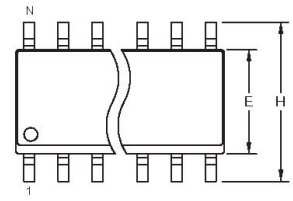
(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)

SOICN .EPS

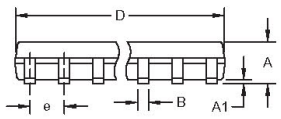
	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.053	0.069	1.35	1.75
A1	0.004	0.010	0.10	0.25
B	0.014	0.019	0.35	0.49
C	0.007	0.010	0.19	0.25
e	0.050 BSC		1.27 BSC	
E	0.150	0.157	3.80	4.00
H	0.228	0.244	5.80	6.20
L	0.016	0.050	0.40	1.27

VARIATIONS:

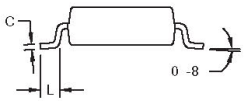
	INCHES		MILLIMETERS			
DIM	MIN	MAX	MIN	MAX	N	MS012
D	0.189	0.197	4.80	5.00	8	AA
D	0.337	0.344	8.55	8.75	14	AB
D	0.386	0.394	9.80	10.00	16	AC



TOP VIEW



FRONT VIEW



SIDE VIEW

NOTES:
1. D&E DO NOT INCLUDE MOLD FLASH.
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED 0.15mm (.006").
3. LEADS TO BE COPLANAR WITHIN 0.10mm (.004").
4. CONTROLLING DIMENSION: MILLIMETERS.
5. MEETS JEDEC MS012.
6. N = NUMBER OF PINS.

DALLAS SEMICONDUCTOR MAXIM

PROPRIETARY INFORMATION

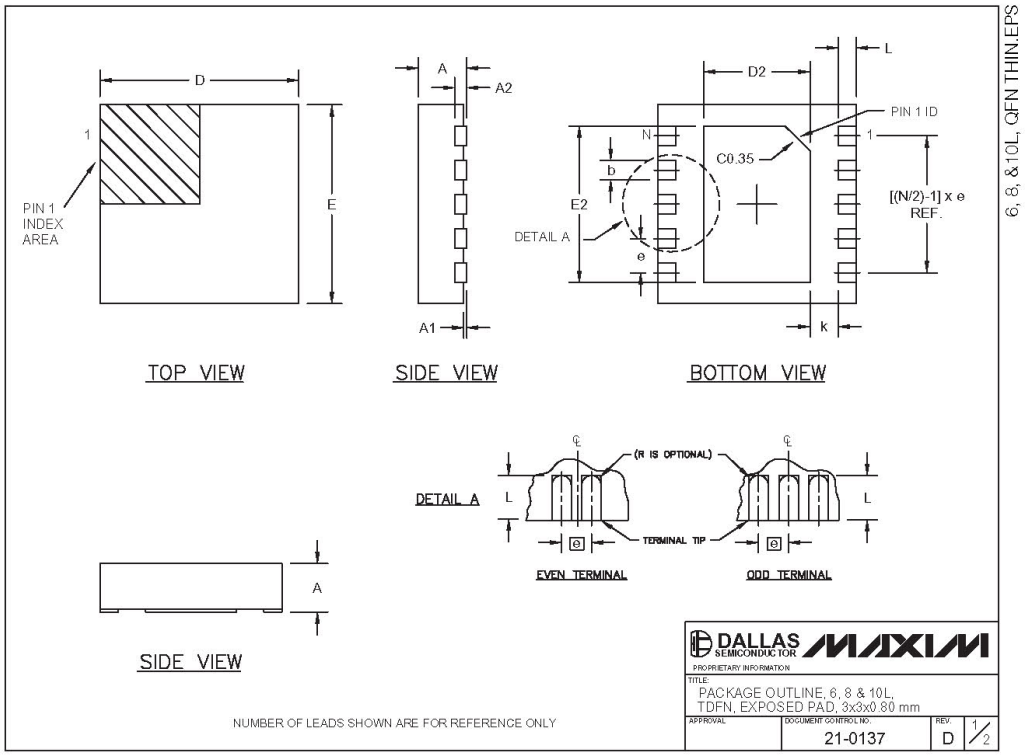
TITLE: PACKAGE OUTLINE, .150" SOIC

APPROVAL	DOCUMENT CONTROL NO.	REV.	
	21-0041	B	1/1

Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)



Single/Dual/Quad, 28MHz, Low-Noise, Low-Voltage, Precision Op Amps

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)

COMMON DIMENSIONS		
SYMBOL	MIN.	MAX.
A	0.70	0.80
D	2.90	3.10
E	2.90	3.10
A1	0.00	0.05
L	0.20	0.40
k	0.25 MIN.	
A2	0.20 REF.	

PACKAGE VARIATIONS							
PKG. CODE	N	D2	E2	e	JEDEC SPEC	b	[(N/2)-1] x e
T633-1	6	1.50–0.10	2.30–0.10	0.95 BSC	MO229 / WEEA	0.40–0.05	1.90 REF
T833-1	8	1.50–0.10	2.30–0.10	0.65 BSC	MO229 / WEEC	0.30–0.05	1.95 REF
T1033-1	10	1.50–0.10	2.30–0.10	0.50 BSC	MO229 / WEED-3	0.25–0.05	2.00 REF

NOTES:
1. ALL DIMENSIONS ARE IN mm. ANGLES IN DEGREES.
2. COPLANARITY SHALL NOT EXCEED 0.08 mm.
3. WARPAGE SHALL NOT EXCEED 0.10 mm.
4. PACKAGE LENGTH/PACKAGE WIDTH ARE CONSIDERED AS SPECIAL CHARACTERISTIC(S).
5. DRAWING CONFORMS TO JEDEC MO229, EXCEPT DIMENSIONS "D2" AND "E2".
6. "N" IS THE TOTAL NUMBER OF LEADS.

DALLAS SEMICONDUCTOR MAXIM
PROPRIETARY INFORMATION

TITLE: PACKAGE OUTLINE, 6, 8 & 10L, TDFN, EXPOSED PAD, 3x3x0.80 mm

APPROVAL	DOCUMENT CONTROL NO.	REV.	
	21-0137	D	2/2

AII.5. Technical specifications of the Differential Comparator LM360

National Semiconductor

August 2000

LM160/LM360
High Speed Differential Comparator

General Description

The LM160/LM360 is a very high speed differential input, complementary TTL output voltage comparator with improved characteristics over the µA760/µA760C, for which it is a pin-for-pin replacement. The device has been optimized for greater speed, input impedance and fan-out, and lower input offset voltage. Typically delay varies only 3 ns for overdrive variations of 5 mV to 400 mV.

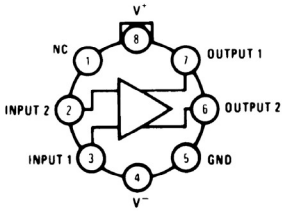
Complementary outputs having minimum skew are provided. Applications involve high speed analog to digital convertors and zero-crossing detectors in disk file systems.

Features

- Guaranteed high speed: 20 ns max
- Tight delay matching on both outputs
- Complementary TTL outputs
- High input impedance
- Low speed variation with overdrive variation
- Fan-out of 4
- Low input offset voltage
- Series 74 TTL compatible

Connection Diagrams

Metal Can Package

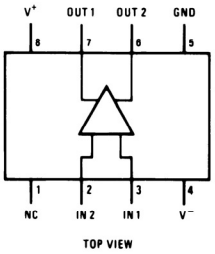


TOP VIEW

00570704

Order Number LM160H/883 (Note 1)
See NS Package Number H08C

Dual-In-Line Package



TOP VIEW

00570705

Order Number LM360M, LM360MX or LM360N
See NS Package Number M08A or N08E

Note 1: Also available in SMD# 5962-8767401

 DS005707

Absolute Maximum Ratings (Notes 6, 8)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Positive Supply Voltage	+8V
Negative Supply Voltage	−8V
Peak Output Current	20 mA
Differential Input Voltage	±5V
Input Voltage	$V^+ \geq V_{IN} \geq V^-$
ESD Tolerance (Note 9)	1600V
Operating Temperature Range	
LM160	−55˚C to +125˚C
LM360	0˚C to +70˚C
Storage Temperature Range	−65˚C to +150˚C
Lead Temperature	
(Soldering, 10 sec.)	260˚C
Soldering Information	
Dual-In-Line Package	
Soldering (10 seconds)	260˚C
Small Outline Package	
Vapor Phase (60 seconds)	215˚C
Infrared (15 seconds)	220˚C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics

($T_{MIN} \leq T_A \leq T_{MAX}$)

Parameter	Conditions	Min	Typ	Max	Units
Operating Conditions					
Supply Voltage V_{CC}^+		4.5	5	6.5	V
Supply Voltage V_{CC}^-		−4.5	−5	−6.5	V
Input Offset Voltage	$R_S \leq 200\Omega$		2	5	mV
Input Offset Current			0.5	3	µA
Input Bias Current			5	20	µA
Output Resistance (Either Output)	$V_{OUT} = V_{OH}$		100		Ω
Response Time	$T_A = 25^\circ C$, $V_S = \pm 5V$ (Notes 2, 7)		13	25	ns
	$T_A = 25^\circ C$, $V_S = \pm 5V$ (Notes 3, 7)		12	20	ns
	$T_A = 25^\circ C$, $V_S = \pm 5V$ (Notes 4, 7)		14		ns
Response Time Difference between Outputs					
(t_{pd} of $+V_{IN1}$) − (t_{pd} of $-V_{IN2}$)	$T_A = 25^\circ C$ (Notes 2, 7)		2		ns
(t_{pd} of $+V_{IN2}$) − (t_{pd} of $-V_{IN1}$)	$T_A = 25^\circ C$ (Notes 2, 7)		2		ns
(t_{pd} of $+V_{IN1}$) − (t_{pd} of $+V_{IN2}$)	$T_A = 25^\circ C$ (Notes 2, 7)		2		ns
(t_{pd} of $-V_{IN1}$) − (t_{pd} of $-V_{IN2}$)	$T_A = 25^\circ C$ (Notes 2, 7)		2		ns
Input Resistance	f = 1 MHz		17		kΩ
Input Capacitance	f = 1 MHz		3		pF
Average Temperature Coefficient of Input Offset Voltage	$R_S = 50\Omega$		8		µV/˚C
Average Temperature Coefficient of Input Offset Current			7		nA/˚C
Common Mode Input Voltage Range	$V_S = \pm 6.5V$	±4	±4.5		V
Differential Input Voltage Range		±5			V
Output High Voltage (Either Output)	$I_{OUT} = -320$ µA, $V_S = \pm 4.5V$	2.4	3		V
Output Low Voltage (Either Output)	$I_{SINK} = 6.4$ mA		0.25	0.4	V
Positive Supply Current	$V_S = \pm 6.5V$		18	32	mA
Negative Supply Current	$V_S = \pm 6.5V$		−9	−16	mA

Note 2: Response time measured from the 50% point of a 30 mVp-p 10 MHz sinusoidal input to the 50% point of the output.

Note 3: Response time measured from the 50% point of a 2 Vp-p 10 MHz sinusoidal input to the 50% point of the output.

Electrical Characteristics (Continued)

Note 4: Response time measured from the start of a 100 mV input step with 5 mV overdrive to the time when the output crosses the logic threshold.

Note 5: Typical thermal impedances are as follows:

Cavity DIP (J):	θ_{jA}	135°C/W	Header (H)	θ_{jA}	165°C/W	(Still Air)
Molded DIP (N):	θ_{jA}	130°C/W			67°C/W	(400 LF/min Air Flow)
				θ_{jC}	25°C/W	

Note 6: The device may be damaged if used beyond the maximum ratings.

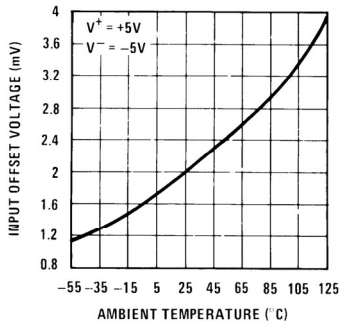
Note 7: Measurements are made in AC Test Circuit, Fanout = 1

Note 8: Refer to RETS 160X for LM160H, LM160J-14 and LM160J military specifications.

Note 9: Human body model, 1.5 kΩ in series with 100 pF.

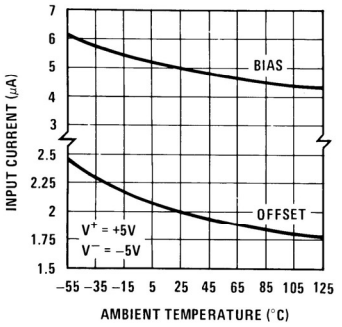
Typical Performance Characteristics

Offset Voltage



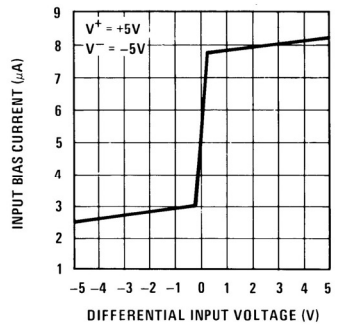
00570708

Input Current vs Ambient Temperature



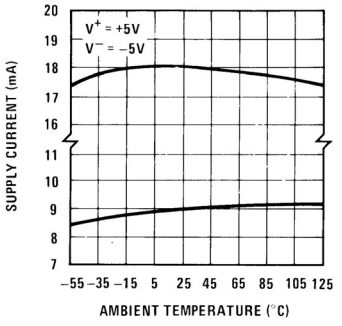
00570709

Input Characteristics



00570710

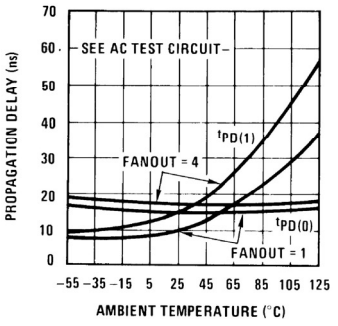
Supply Current vs Ambient Temperature



00570711

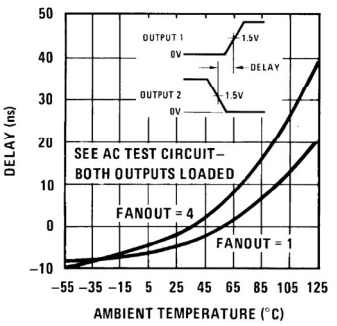
Typical Performance Characteristics (Continued)

Propagation Delay vs Ambient Temperature



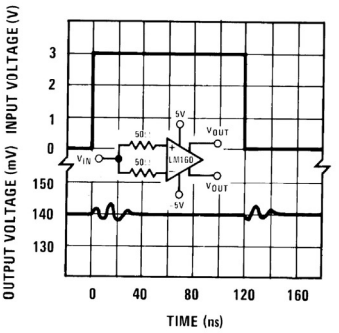
00570712

Delay of Output 1 With Respect to Output 2 vs Ambient Temperature



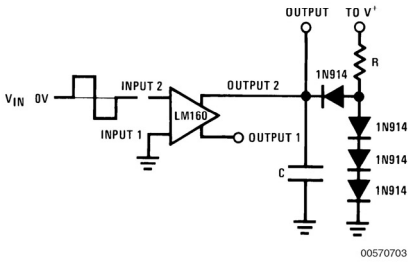
00570713

Common-Mode Pulse Response



00570714

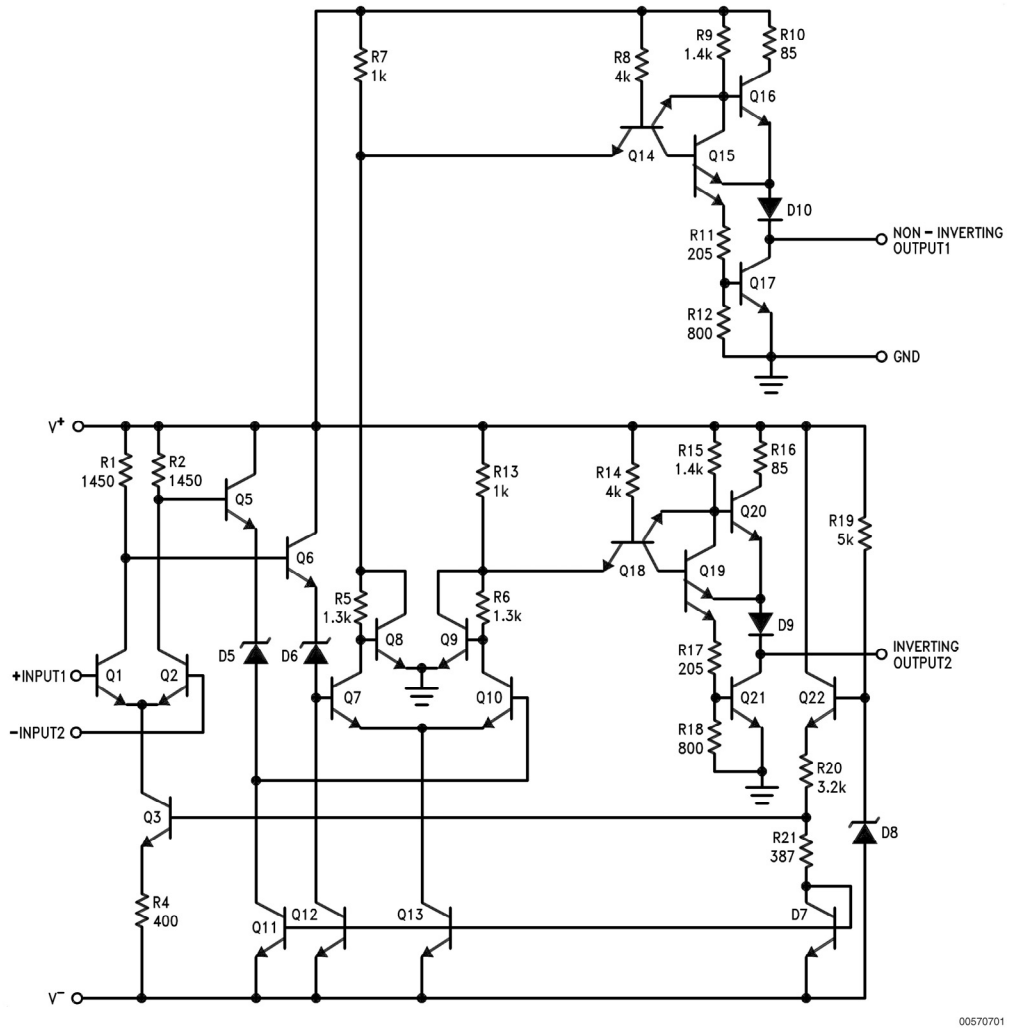
AC Test Circuit



00570703

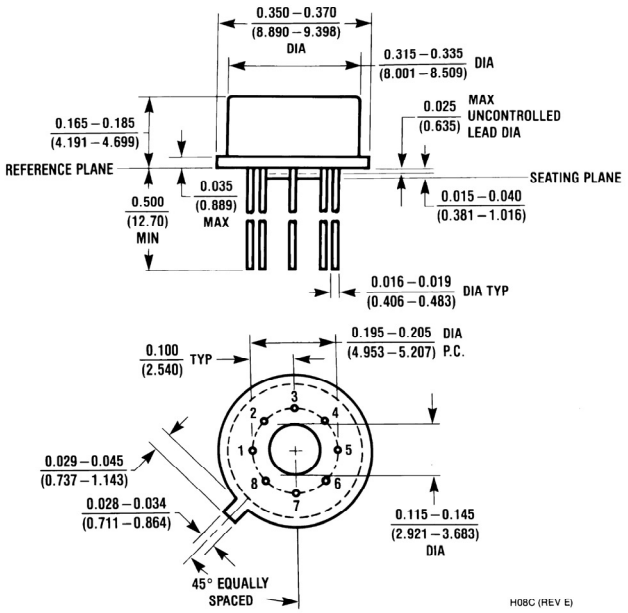
V_{IN}=±50 mV	FANOUT=1	FANOUT=4
V^+=+5V	R=2.4k	R=630Ω
V^-=−5V	C=15 pF	C=30 pF

Schematic Diagram



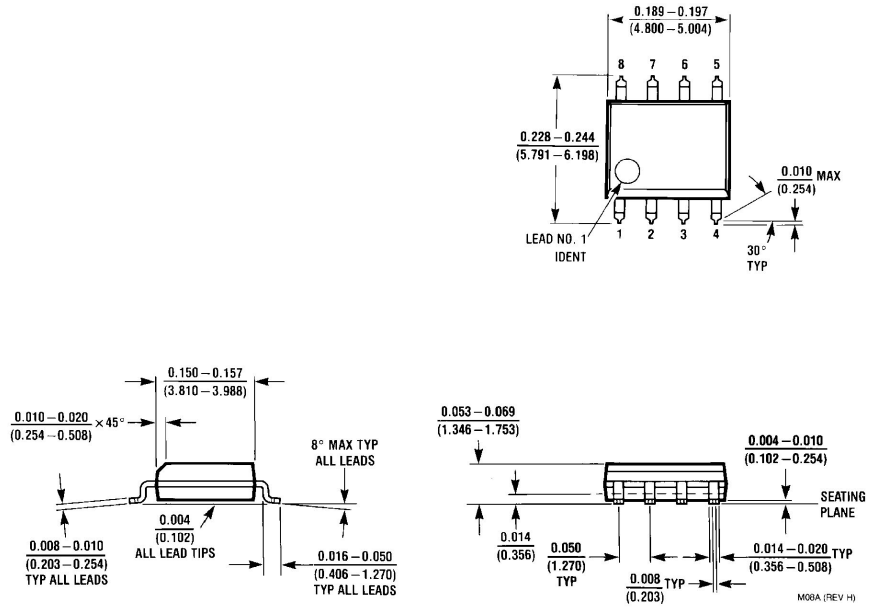
Physical Dimensions inches (millimeters)

unless otherwise noted



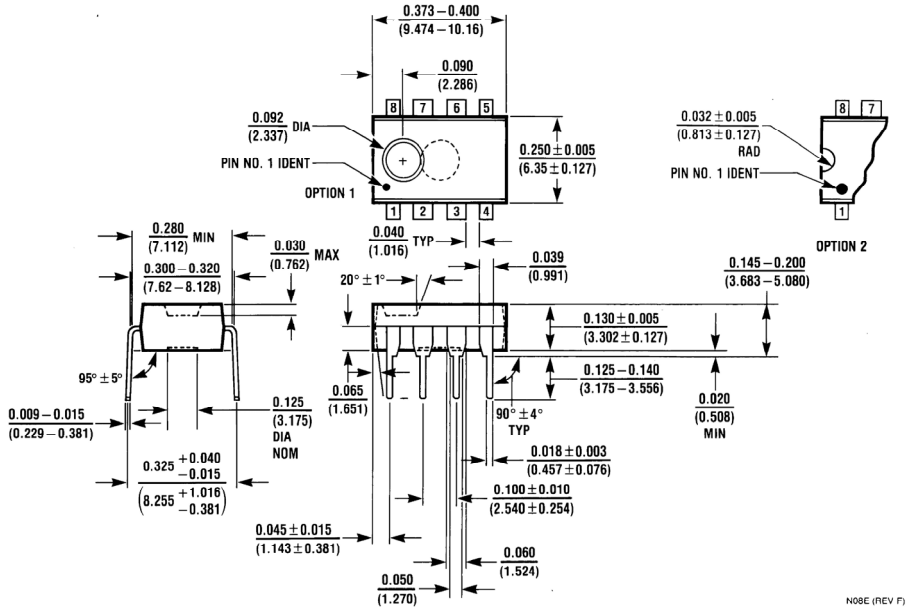
Metal Can Package (H)
Order Number LM160H/883
NS Package Number H08C

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Dual-In-Line Package (M)
Order Number LM360M or LM360MX
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Dual-In-Line Package (N)
Order Number LM360N
NS Package Number N08E

National Semiconductor
Americas Customer
Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

www.national.com

National Semiconductor
Europe Customer Support Center
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor
Asia Pacific Customer
Support Center
Email: ap.support@nsc.com

National Semiconductor
Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560

AII.6. Technical specification of the 5.0V TTL Clock Oscillator F1100E

5.0V TTL CLOCK OSCILLATOR

F1100E

FEATURES

- 5.0V Operation
- TTL Output
- 14-Pin DIP

FASTFOX

• MODEL NUMBER SELECTION

Model Number	Frequency Stability[1]	Operating Temperature (°C)	Frequency Range (MHz)
F1100E	±100PPM (STD)	0 ~ +70	1.000 ~ 100.000
F1100ER	±100PPM	-40 ~ +85	1.000 ~ 100.000
F1145E	±50PPM	0 ~ +70	1.000 ~ 100.000
F1145ER	±50PPM	-40 ~ +85	1.000 ~ 70.000
F1144E	±25PPM	0 ~ +70	1.000 ~ 100.000
F1144ER	±25PPM	-40 ~ +85	1.000 ~ 70.000

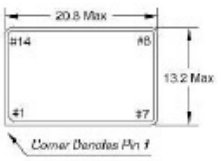
• ELECTRICAL CHARACTERISTICS

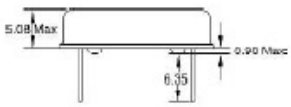
PARAMETERS	MAX (unless otherwise noted)
Frequency Range (Fo)	1.000 ~ 100.000 MHz
Storage Temperature Range (Tsto)	-55°C ~ +125°C
Supply Voltage (Vdd)	5.0V ± 10%
Input Current (Idd)	
1.000 ~ 8.000 MHz	15mA
8.000+ ~ 24.000 MHz	30mA
24.000+ ~ 70.000 MHz	70mA
70.000+ ~ 100.000 MHz	80mA
Output Symmetry (1.4V Level)	
1.000 ~ 8.000 MHz	45% ~ 55%
8.000+ ~ 100.000 MHz	40% ~ 60%
Rise Time (0.5V ~ 2.4V) (Tr)	
1.000 ~ 25.000 MHz	10 nS
25.000+ ~ 70.000 MHz	5 nS
70.000+ ~ 100.000 MHz	4 nS
Fall Time (2.4V ~ 0.5V) (Tf)	
1.000 ~ 25.000 MHz	10 nS
25.000+ ~ 70.000 MHz	5 nS
70.000+ ~ 100.000 MHz	4 nS
Output Voltage	
1.000 ~ 25.000 MHz (Vol)	0.4V
25.000+ ~ 100.000 MHz	0.5V
1.000 ~ 100.000 MHz (Voh)	2.4V Min
Output Current (Iol)	20mA Min
(Ioh)	-1.0mA Min
Output Load	10TTL
Start-up Time (Ts)	
1.000 ~ 3.500 MHz	20mS
3.500+ ~ 4.000 MHz	35mS
4.000+ ~ 6.000 MHz	30mS
6.000+ ~ 20.000 MHz	20mS
20.000+ ~ 100.000 MHz	15mS

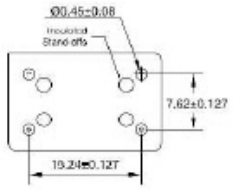
[1] Inclusive of 25°C tolerance, operating temperature range, input voltage change, load change, aging, shock, and vibration.

See page 30 for mechanical specifications, test circuits, and output waveform.

All specifications subject to change without notice. *Rev. 02/10/03*







Pin Connections

#1 N.C. #8 Output
#7 GND (Case) #14 +5Vdc

All dimensions are in millimeters.

AII.7. Altera MAX 9000 programmable logic device family data sheet

Includes MAX 9000A

MAX 9000

Programmable Logic Device Family

February 1998, ver. 5.01 — Data Sheet

Features...

- High-performance CMOS EEPROM-based programmable logic devices (PLDs) built on third-generation Multiple Array MatriX (MAX®) architecture
- 5.0-V in-system programmability (ISP) through built-in Joint Test Action Group (JTAG) interface
- Built-in JTAG boundary-scan test (BST) circuitry compliant with IEEE Std. 1149.1-1990
- High-density erasable programmable logic device (EPLD) family ranging from 6,000 to 12,000 usable gates (see Table 1)
- 7.5-ns pin-to-pin logic delays with counter frequencies of up to 178 MHz (the -7 speed grade is under development)
- Peripheral component interconnect (PCI)-compliant -7 and -10 speed grade devices
- Dual-output macrocell for independent use of combinatorial and registered logic
- FastTrack Interconnect™ continuous routing structure for fast, predictable interconnect delays
- Input/output registers with clear and clock enable on all I/O pins
- Programmable output slew-rate control to reduce switching noise
- MultiVolt™ I/O interface operation, allowing devices to interface with 3.3-V and 5.0-V devices
- Configurable expander product-term distribution allowing up to 32 product terms per macrocell
- Programmable power-saving mode for more than 50% power reduction in each macrocell

Table 1. MAX 9000 Device Features

Feature	EPM9320 EPM9320A	EPM9400	EPM9480 EPM9480A	EPM9560 EPM9560A
Usable gates	6,000	8,000	10,000	12,000
Flipflops	484	580	676	772
Macrocells	320	400	480	560
Logic array blocks (LABs)	20	25	30	35
Maximum user I/O pins	168	159	175	216
t_{PD1} (ns)	7.5/10	15	10	10
t_{FSU} (ns)	3.0	5	3.0	3.0
t_{FCO} (ns)	4.5	7	4.8	4.8
f_{CNT} (MHz)	178	118	154	154

...and More Features

- Programmable macrocell flipflops with individual clear, preset, clock, and clock enable controls
- Programmable security bit for protection of proprietary designs
- Software design support and automatic place-and-route provided by Altera's MAX+PLUS® II development system on 486- and Pentium-based PCs, and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations
- Additional design entry and simulation support provided by EDIF 2 0 0 and 3 0 0 netlist files, library of parameterized modules (LPM), Verilog HDL, VHDL, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, and VeriBest
- Programming support with Altera's Master Programming Unit (MPU), BitBlaster™ serial download cable, and ByteBlaster™ parallel port download cable, as well as programming hardware from third-party manufacturers
- Offered in a variety of package options with 84 to 356 pins (see Table 2)

Table 2. MAX 9000 Package Options & I/O Counts *Notes (1), (2)*

Device	84-Pin PLCC	208-Pin RQFP	240-Pin RQFP	280-Pin PGA	304-Pin RQFP	356-Pin BGA
EPM9320	60 *(3)*	132	–	168	–	168
EPM9320A	60 *(3)*	132	–	–	–	168
EPM9400	59 *(3)*	139	159	–	–	–
EPM9480	–	146	175	–	–	–
EPM9480A	–	146	175	–	–	–
EPM9560	–	153	191	216	216	216
EPM9560A	–	153	191	–	–	216

Notes:

(1) MAX 9000 device package types include plastic J-lead chip carrier (PLCC), power quad flat pack (RQFP), ceramic pin-grid array (PGA), and ball-grid array (BGA) packages.

(2) Contact Altera Customer Marketing at (408) 544-7104 for up-to-date information on package availability. All information on EPM9320A, EPM9480A, and EPM9560A devices is preliminary.

(3) Perform a complete thermal analysis before committing a design to this device package. See *Application Note 74 (Evaluating Power for Altera Devices)* in the ***1998 Data Book***.

General Description

The MAX 9000 family of in-system-programmable, high-density, high-performance EPLDs is based on Altera's third-generation MAX architecture. Fabricated on an advanced CMOS technology, the EEPROM-based MAX 9000 family provides 6,000 to 12,000 usable gates, pin-to-pin delays as fast as 7.5 ns, and counter speeds of up to 178 MHz. The -7 and -10 speed grades of the MAX 9000 family are compliant with the *PCI Local Bus Specification, Revision 2.1*. Table 3 shows the speed grades available for MAX 9000 devices.

Table 3. MAX 9000 Speed Grade Availability

Device	Speed Grade			
	-7 *Note (1)*	-10	-15	-20
EPM9320			✓	✓
EPM9320A	✓ *(2)*	✓ *(2)*	✓ *(2)*	
EPM9400			✓	✓
EPM9480			✓	✓
EPM9480A		✓ *(2)*	✓ *(2)*	
EPM9560			✓	✓
EPM9560A		✓ *(2)*	✓ *(2)*	

Notes:
(1) The -7 speed grade is under development.
(2) This information is preliminary. Contact Altera for up-to-date information on speed grade availability.

Table 4 shows the performance of MAX 9000 devices for typical functions.

Table 4. MAX 9000 Performance *Note (1)*

Application	Macrocells Used	Speed Grade				Units
		-7 *(2), (3)*	-10 *(2)*	-15	-20	
16-bit loadable counter	16	178	144	118	100	MHz
16-bit up/down counter	16	178	144	118	100	MHz
16-bit prescaled counter	16	178	144	118	100	MHz
16-bit address decode	1	4.4 (7.5)	5.6 (10)	7.9 (15)	10 (20)	ns
16-to-1 multiplexer	1	5.8 (9.3)	7.7 (12.1)	10.9 (18)	16 (26)	ns

Notes:
(1) Internal logic array block (LAB) performance is shown. Numbers in parentheses show external delays from row input pin to row I/O pin.
(2) This information is preliminary.
(3) The -7 speed grade is under development.

The MAX 9000 architecture supports high-density integration of system-level logic functions. It easily integrates multiple programmable logic devices ranging from PALs, GALs, and 22V10s to field-programmable gate array (FPGA) devices and EPLDs. With speed, density, and I/O resources comparable to commonly used masked gate arrays, MAX 9000 EPLDs are also ideal for gate-array prototyping.

All MAX 9000 device packages provide four dedicated inputs for global control signals with large fan-outs. Each I/O pin has an associated I/O cell register with a clock enable control on the periphery of the device. As outputs, these registers provide fast clock-to-output times; as inputs, they offer quick setup times.

MAX 9000 EPLDs provide 5.0-V in-system programmability (ISP). This feature allows the devices to be programmed and reprogrammed on the printed circuit board (PCB) for quick and efficient iterations during design development and debug cycles. MAX 9000 devices are guaranteed for 100 program and erase cycles.

MAX 9000 EPLDs contain from 320 to 560 macrocells that are combined into groups of 16 macrocells, called logic array blocks (LABs). Each macrocell has a programmable-AND/fixed-OR array and a configurable register with independently programmable clock, clock enable, clear, and preset functions. For increased flexibility, each macrocell offers a dual-output structure that allows the register and the product terms to be used independently. This feature allows register-rich and combinatorial-intensive designs to be implemented efficiently. The dual-output structure of the MAX 9000 macrocell also improves logic utilization, thus increasing the effective capacity of the devices. To build complex logic functions, each macrocell can be supplemented with both shareable expander product terms and high-speed parallel expander product terms to provide up to 32 product terms per macrocell.

The MAX 9000 family provides programmable speed/power optimization. Speed-critical portions of a design can run at high speed/full power, while the remaining portions run at reduced speed/low power. This speed/power optimization feature enables the user to configure one or more macrocells to operate at 50% or less power while adding only a nominal timing delay. MAX 9000 devices also provide an option that reduces the slew rate of the output buffers, minimizing noise transients when non-speed-critical signals are switching. MAX 9000 devices offer the MultiVolt feature, which allows output drivers to be set for either 3.3-V or 5.0-V operation in mixed-voltage systems.

The MAX 9000 family is supported by Altera's MAX+PLUS II development system, a single, integrated software package that offers schematic, text—including VHDL, Verilog HDL, and the Altera Hardware Description Language (AHDL)—and waveform design entry; compilation and logic synthesis; simulation and timing analysis; and device programming. The MAX+PLUS II software provides EDIF 2 0 0 and 3 0 0, LPM, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX-workstation-based EDA tools. The MAX+PLUS II software runs on 486- and Pentium-based PCs, and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations.

For more information on development tools, go to the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* in the ***1998 Data Book***.

Device Pin-Outs

Tables 10, 11, 12, and 13 show the pin names and numbers for the dedicated pins for each EPM9320, EPM9320A, EPM9400, EPM9480, EPM9480A, EPM9560, and EPM9560A device package.

Table 10. EPM9320 & EPM9320A Dedicated Pin-Outs (Part 1 of 2)

Pin Name	84-Pin PLCC, *Note (1)*	208-Pin RQFP	280-Pin PGA, *Note (2)*	356-Pin BGA
DIN1 (GCLK1)	1	182	V10	AD13
DIN2 (GCLK2)	84	183	U10	AF14
DIN3 (GCLR)	13	153	V17	AD1
DIN4 (GOE)	72	4	W2	AC24
TCK	43	78	A9	A18
TMS	55	49	D6	E23
TDI	42	79	C11	A13
TDO	30	108	A18	D3
GND	6, 18, 24, 25, 48, 61, 67, 70	14, 20, 24, 31, 35, 41, 42, 43, 44, 46, 47, 66, 85, 102, 110, 113, 114, 115, 116, 118, 121, 122, 132, 133, 143, 152, 170, 189, 206	D4, D5, D16, E4, E5, E6, E15, E16, F5, F15, G5, G15, H5, H15, J5, J15, K5, K15, L5, L15, M5, M15, N5, N15, P4, P5, P15, P16, R4, R5, R15, R16, T4, T5, T16	A9, A22, A25, A26, B25, B26, D2, E1, E26, F2, G1, G25, G26, H2, J1, J25, J26, K2, L26, M26, N1, N25, P26, R2, T1, U2, U26, V1, V25, W25, Y26, AA2, AB1, AB26, AC26, AE1, AF1, AF2, AF4, AF7, AF20
VCCINT (5.0 V only)	14, 21, 28, 57, 64, 71	10, 19, 30, 45, 112, 128, 139, 148	D15, E8, E10, E12, E14, R7, R9, R11, R13, R14, T14	D26, F1, H1, K26, N26, P1, U1, W26, AE26, AF25, AF26
VCCIO (3.3 or 5.0 V)	15, 37, 60, 79	5, 25, 36, 55, 72, 91, 111, 127, 138, 159, 176, 195	D14, E7, E9, E11, E13, R6, R8, R10, R12, T13, T15	A1, A2, A21, B1, B10, B24, D1, H26, K1, M25, R1, V26, AA1, AC25, AF5, AF8, AF19

Table 10. EPM9320 & EPM9320A Dedicated Pin-Outs (Part 2 of 2)

Pin Name	84-Pin PLCC, *Note (1)*	208-Pin RQFP	280-Pin PGA, *Note (2)*	356-Pin BGA
No Connect (N.C.)	29	6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 109, 140, 141, 142, 144, 145, 146, 147, 149, 150, 151	B6, K19, L2, L4, L18, L19, M1, M2, M3, M4, M16, M17, M18, M19, N1, N2, N3, N4, N16, N17, N18, N19, P1, P2, P3, P17, P18, P19, R1, R2, R3, R17, R18, R19, T1, T2, T3, T17, T18, T19, U1, U2, U3, U17, U18, U19, V1, V2, V19, W1	B4, B5, B6, B7, B8, B9, B11, B12, B13, B14, B15, B16, B18, B19, B20, B21, B22, B23, C4, C23, D4, D23, E4, E22, F4, F23, G4, H4, H23, J23, K4, L4, L23, N4, P4, P23, R3, R26, T2, T3, T4, T5, T22, T23, T24, T25, T26, U3, U4, U5, U22, U23, U24, U25, V2, V3, V4, V5, V22, V23, V24, W1, W2, W3, W4, W5, W22, W23, W24, Y1, Y2, Y3, Y4, Y5, Y22, Y23, Y24, Y25, AA3, AA4, AA5, AA22, AA23, AA24, AA25, AA26, AB2, AB3, AB4, AB5, AB23, AB24, AB25, AC1, AC2, AC23, AD4, AD23, AE4, AE5, AE6, AE7, AE9, AE11, AE12, AE14, AE15, AE16, AE18, AE19, AE20, AE21, AE22, AE23
VPP, *Note (3)*	56	48	C4	E25
Total User I/O Pins, *Note (4)*	56	128	164	164

Notes:

(1) Perform a complete thermal analysis before committing a design to this device package. See *Application Note 74 (Evaluating Power for Altera Devices)* in the ***1998 Data Book***.

(2) This package is not offered for EPM9320A devices.

(3) During in-system programming, each device's VPP pin must be connected to the 5.0-V power supply. During normal device operation, the VPP pin is pulled up internally and can be connected to the 5.0-V supply or left unconnected.

(4) Pins that are not listed are user I/O pins.

Revision History

The information contained in the *MAX 9000 Programmable Logic Device Family Data Sheet* version 5.01 supersedes information published in the *MAX 9000 Programmable Logic Device Family Data Sheet* version 5.0, which can be found in the ***1998 Data Book***.

The *MAX 9000 Programmable Logic Device Family Data Sheet* version 5.01 contains the following changes:

- Note (1) in Table 4 has been reworded for clarification.
- A sentence has been added to the dedicated inputs paragraph referring readers to Figure 2 for more information.
- An erroneous reference to UESCODE in the IEEE 1149.1 (JTAG) Boundary Scan Support section has been removed.
- The graph slope in Figure 15 has been corrected.
- Minor textual, illustration, and style changes were made to the data sheet.

AII.8. Altera device package information for EPM9320ALC84-10

84-Pin Plastic J-Lead Chip Carrier (PLCC)

- All dimensions and tolerances conform to ASME Y14.5M – 1994.
- Controlling dimension is in inches.
- Pin 1 is generally indicated by an indentation in the plastic body, in Pin 1's proximity, on package surface.

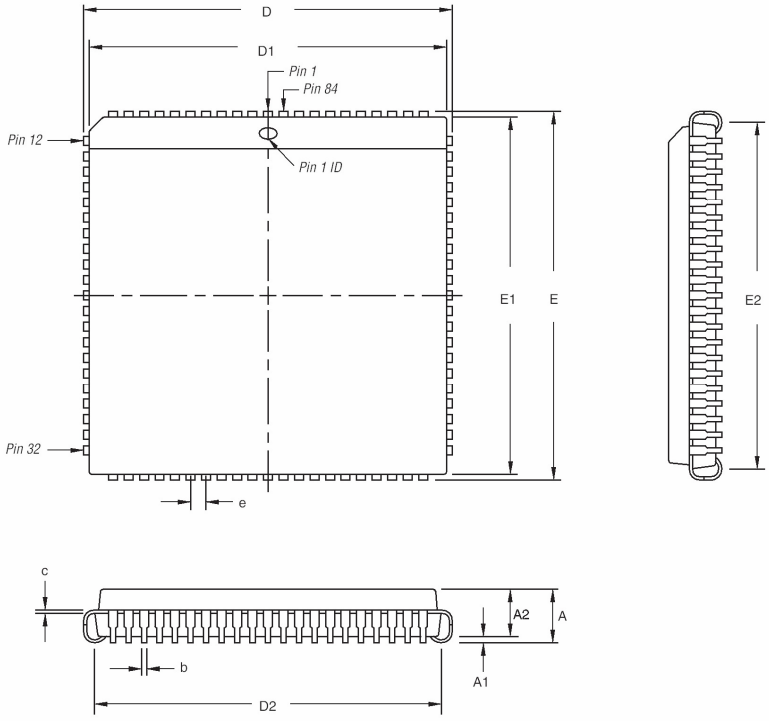
Package Information

Description	Specification
Ordering Code Reference	L
Package Acronym	PLCC
Leadframe Material	Copper
Lead Finish (Plating)	Regular: 85Sn:15Pb (Typ.) Pb-free: Matte Sn
JEDEC Outline Reference	MS-018 Variation: AF
Maximum Lead Coplanarity	0.004 inches (0.10mm)
Weight	6.8 g
Moisture Sensitivity Level	Printed on moisture barrier bag

Package Outline Dimension Table

Symbol	Inches		
	Min.	Nom.	Max.
A	0.165	0.172	0.180
A1	0.020	–	–
A2	0.150 TYP		
D	1.185	1.190	1.195
D1	1.150	1.154	1.158
D2	1.082	1.110	1.138
E	1.185	1.190	1.195
E1	1.150	1.154	1.158
E2	1.082	1.110	1.138
b	0.013	–	0.021
c	0.008 TYP		
e	0.050 TYP		

Package Outline



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Annex III. VHDL code

Contents

AIII.1	**Masters: VHDL code of the block "Recognize Header"**	**244**
AIII.2	**Masters: VHDL code of the block "Buffer"**	**247**
AIII.3	**Masters: VHDL code of the block "Parallel Serial Converter"**	**250**
AIII.4	**Masters: VHDL code of the block "Multiplexer"**	**253**
AIII.5	**Masters: VHDL code of the block "Compare Header"**	**254**
AIII.6	**Masters: VHDL code of the block "Vector to Integer Converter"**	**255**
AIII.7	**Slaves: VHDL code of the block "Recognize Header "**	**256**
AIII.8	**Slaves: VHDL code of the block "Buffer"**	**259**
AIII.9	**Slaves: VHDL code of the block "Parallel Serial Converter"**	**261**
AIII.10	**Slaves: VHDL code of the block "Multiplexer"**	**263**
AIII.11	**Slaves: VHDL code of the block "Compare Header"**	**264**
AIII.12	**Slaves: VHDL code of the block "Compare Payload"**	**265**

AIII.1. Masters: VHDL code of the block “Recognize Header”

```
-- RECOGNIZE  HEADER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY RECOGNIZE_HEADER_TX IS

        PORT
        (
                END_TX                  :IN     STD_LOGIC_VECTOR (0 TO 1);
                RESET_TX_1              :IN     STD_LOGIC;
                COMPARE                 :IN     STD_LOGIC_VECTOR (0 TO 1);
                CLK                     :IN     STD_LOGIC;
                TX_PACKET               :IN     STD_LOGIC;
                LISTEN                  :OUT    STD_LOGIC;
                WAIT_TX                 :OUT    STD_LOGIC;
                STATE                   :OUT    STD_LOGIC_VECTOR (0 TO 2);
                RESET_TX_2              :OUT    STD_LOGIC;
                READ_PN_NUMBER          :OUT    STD_LOGIC;
                SEL                     :OUT    STD_LOGIC_VECTOR (0 TO 1)
        );

END RECOGNIZE_HEADER_TX;

ARCHITECTURE A OF RECOGNIZE_HEADER_TX IS

                SIGNAL count                    :INTEGER RANGE 20 DOWNTO 0;
                SIGNAL recog_sync               :STD_LOGIC := '0';
                SIGNAL recog_id_rx              :STD_LOGIC := '0';
                SIGNAL sel_signal               :STD_LOGIC_VECTOR (0 TO 1);
                SIGNAL listen_signal            :STD_LOGIC;
                SIGNAL state_signal             :STD_LOGIC_VECTOR (0 TO 2);
                SIGNAL state_signal_aux         :STD_LOGIC_VECTOR (0 TO 2);
                SIGNAL reset_tx_2_signal        :STD_LOGIC;
                SIGNAL timeout_signal           :INTEGER RANGE 19 TO 0;
                SIGNAL waiting_flag             :STD_LOGIC;
                SIGNAL waiting_time             :INTEGER RANGE 0 TO 333;
                SIGNAL read_pn_number_signal    :STD_LOGIC;
                SIGNAL flag_tx                  :STD_LOGIC;

BEGIN

        PROCESS (END_TX, RESET_TX_1, COMPARE, CLK, TX_PACKET)
        BEGIN


        wait until CLK'event AND CLK = '1';

                IF (TX_PACKET = '1') THEN
                        flag_tx <= '1';
                        listen_signal <= '0';
                        sel_signal <= "00";
                        state_signal_aux <= "000";
                        state_signal <= "000";
                        recog_sync <= '0';
                        recog_id_rx <= '0';
                        count <= 0;
                END IF;

                IF (read_pn_number_signal = '1') THEN
                        read_pn_number_signal <= '0';
                END IF;
```

```
		IF (reset_tx_2_signal = '1') THEN
			reset_tx_2_signal <= '0';
			flag_tx <= '0';
			read_pn_number_signal <= '1';
		END IF;

		IF (waiting_flag = '1') THEN
			waiting_time <= waiting_time - 1;
			IF (waiting_time = 0) THEN
				waiting_flag <= '0';
				reset_tx_2_signal <= '1';
			END IF;
		END IF;

		IF (RESET_TX_1 = '1') THEN
			reset_tx_2_signal <= '1';
		END IF;

		IF state_signal_aux /= "000" THEN
			state_signal_aux <= "000";
		END IF;

		IF (flag_tx = '1' AND waiting_flag = '0') THEN

			CASE state_signal IS
				WHEN "000" =>
					IF END_TX = "01" THEN
						state_signal_aux <= "001";
						state_signal <= "001";
						listen_signal <= '1';
						timeout_signal <= 19;
					END IF;

				WHEN "001" =>
					IF (timeout_signal = 0) THEN
						reset_tx_2_signal <= '1';
					ELSIF (recog_sync = '0') THEN
						timeout_signal <= timeout_signal - 1;
					END IF;
					IF (COMPARE = "01" AND recog_sync = '0') THEN
						count <= 0;
						recog_sync <= '1';
						sel_signal <= "10";
					ELSIF (COMPARE = "01" AND recog_sync = '1') THEN
						count <= count + 1;
						recog_id_rx <= '1';
						listen_signal <= '0';
						state_signal_aux <= "010";
						state_signal <= "001";
					ELSIF recog_sync = '1' THEN
						count <= count + 1;
					END IF;
					IF (count = 7 AND recog_id_rx = '0' AND COMPARE /= "11" AND
COMPARE /= "01") THEN
						reset_tx_2_signal <= '1';
					END IF;
					IF (COMPARE = "11") THEN
						waiting_flag <= '1';
						waiting_time <= 333;
					END IF;
					IF END_TX = "10" THEN
						sel_signal <= "01";
						state_signal_aux <= "011";
						state_signal <= "011";
						listen_signal <= '1';
						recog_sync <= '0';
```

```
                                        recog_id_rx <= '0';
                                        count <= 0;
                                        timeout_signal <= 19;
                                END IF;

                        WHEN "011" =>
                                IF (timeout_signal = 0) THEN
                                        reset_tx_2_signal <= '1';
                                ELSIF (recog_sync = '0') THEN
                                        timeout_signal <= timeout_signal - 1;
                                END IF;
                                IF (COMPARE = "01" AND recog_sync = '0') THEN
                                        count <= 0;
                                        recog_sync <= '1';
                                        sel_signal <= "10";
                                ELSIF (COMPARE = "01" AND recog_sync = '1') THEN
                                        count <= count + 1;
                                        recog_id_rx <= '1';
                                        listen_signal <= '0';
                                        state_signal_aux <= "111";
                                        state_signal <= "011";
                                ELSIF recog_sync = '1' THEN
                                        count <= count + 1;
                                END IF;
                                IF (count = 8 AND recog_id_rx = '0') THEN
                                        reset_tx_2_signal <= '1';
                                ELSIF (count = 20) THEN
                                        recog_sync <= '0';
                                        recog_id_rx <= '0';
                                        count <= 0;
                                END IF;

                        WHEN OTHERS =>
                                NULL;
                END CASE;
        END IF;

    END PROCESS;

RESET_TX_2 <= reset_tx_2_signal;
SEL <= sel_signal;
LISTEN <= listen_signal;
STATE <= state_signal_aux;
WAIT_TX <= waiting_flag;
READ_PN_NUMBER <= read_pn_number_signal;

END A;
```

AIII.2. Masters: VHDL code of the block “Buffer”

```
-- BUFFER BLOCK


LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY BUFFER_TX IS

	PORT
	(
		RESET_TX_2		:IN	STD_LOGIC;
		DATA, CLK		:IN	STD_LOGIC;
		TX_PACKET		:IN	STD_LOGIC;
		WAIT_TX			:IN	STD_LOGIC;
		STATE			:IN	STD_LOGIC_VECTOR (0 TO 2);
		START_TX		:OUT	STD_LOGIC;
		RESET_TX_1		:OUT	STD_LOGIC;
		CORRECT			:OUT	INTEGER RANGE 0 TO 1023;
		PACKET_CORRECT		:OUT	STD_LOGIC
	);

END BUFFER_TX;

ARCHITECTURE A OF BUFFER_TX IS

		SIGNAL count_1			:INTEGER RANGE 8 DOWNTO 0;
		SIGNAL count_2			:INTEGER RANGE 10 DOWNTO 0;
		SIGNAL  flag_buffer		:BOOLEAN := false;
		SIGNAL mem_ind			:INTEGER RANGE 7 DOWNTO 0 := 0;
		SIGNAL start_tx_signal		:STD_LOGIC := '0';
		SIGNAL state_signal		:STD_LOGIC_VECTOR (0 TO 2);
		SIGNAL id_tx_signal		:STD_LOGIC_VECTOR (0 TO 7);
		SIGNAL  ack_signal		:STD_LOGIC;
		SIGNAL set
	:STD_LOGIC;
		SIGNAL reset_tx_signal		:STD_LOGIC;
		SIGNAL count_packet_correct_signal		:INTEGER RANGE 0 TO 1023;
		SIGNAL packet_correct_signal			:STD_LOGIC;

BEGIN

	PROCESS (RESET_TX_2, DATA, CLK, TX_PACKET, WAIT_TX, STATE)
	BEGIN

	wait until CLK'event AND CLK = '1';

		IF (reset_tx_signal = '1') THEN
			reset_tx_signal <= '0';
		END IF;

		IF (packet_correct_signal = '1') THEN
			packet_correct_signal <= '0';
		END IF;

		CASE STATE IS
			WHEN "000" =>
				IF (set = '0' AND TX_PACKET = '1' AND WAIT_TX = '0') THEN
					start_tx_signal <= '1';
					set <= '1';
				ELSE
					start_tx_signal <= '0';
				END IF;
			WHEN "010" | "111" =>
```

```
                                state_signal <= STATE;
                                count_1 <= 0;
                                count_2 <= 0;
                                flag_buffer <= true;
                                mem_ind <= 0;
                                reset_tx_signal <= '0';
                                id_tx_signal <= (OTHERS => '0');
                        WHEN OTHERS =>
                                state_signal <= STATE;
                END CASE;

                IF (RESET_TX_2 = '1') THEN
                        set <= '0';
                END IF;

                CASE state_signal IS
                        WHEN "010" =>
                                IF flag_buffer THEN
                                        CASE count_1 IS
                                                WHEN 0 to 7 =>
                                                        id_tx_signal(mem_ind) <= DATA;
                                                        mem_ind <= mem_ind + 1;
                                                WHEN 8 =>
                                                        flag_buffer <= false;
                                                        IF (id_tx_signal /= X"35") THEN
                                                                reset_tx_signal <= '1';
                                                        ELSE
                                                                start_tx_signal <= '1';

                                                        END IF;
                                                WHEN OTHERS =>
                                                        NULL;
                                        END CASE;
                                        count_1 <= count_1 + 1;
                                END IF;

                        WHEN "111" =>
                                IF flag_buffer THEN
                                        CASE count_2 IS
                                                WHEN 0 to 7 =>
                                                        id_tx_signal(mem_ind) <= DATA;
                                                        mem_ind <= mem_ind + 1;
                                                WHEN 8 =>
                                                        NULL;
                                                WHEN 9 =>
                                                        ack_signal <= DATA;
                                                WHEN 10 =>
                                                        flag_buffer <= false;
                                                        IF ( (id_tx_signal /= X"35") OR (ack_signal = '0')
)THEN
                                                                reset_tx_signal <= '1';
                                                        ELSIF ack_signal = '1' THEN
                                                                reset_tx_signal <= '1';
                                                                count_packet_correct_signal <=
count_packet_correct_signal + 1;
                                                                packet_correct_signal <= '1';
                                                        END IF;
                                                WHEN OTHERS =>
                                                        NULL;
                                        END CASE;
                                        count_2 <= count_2 + 1;
                                END IF;
                                WHEN OTHERS =>
                                        NULL;
                END CASE;
```

```
        END PROCESS;

START_TX <= start_tx_signal;
RESET_TX_1 <= reset_tx_signal;
CORRECT <= count_packet_correct_signal;
PACKET_CORRECT <= packet_correct_signal;

END A;
```

AIII.3. Masters: VHDL code of the block “Parallel Serial Converter”

```
-- PARALLEL SERIAL CONVERTER BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY PARALLEL_SERIAL_CONVERTER_TX IS

	PORT
	(
		RESET_TX_2			:IN	STD_LOGIC;
		STATE				:IN	STD_LOGIC_VECTOR (0 TO 2);
		START_TX			:IN	STD_LOGIC;
		CLK				:IN	STD_LOGIC;
		DATA_TX				:OUT	STD_LOGIC;
		END_TX				:OUT	STD_LOGIC_VECTOR (0 TO 1)
	);

END PARALLEL_SERIAL_CONVERTER_TX;

ARCHITECTURE A OF PARALLEL_SERIAL_CONVERTER_TX IS

		SIGNAL  count_1			:INTEGER RANGE 27 DOWNTO 0;
		SIGNAL  count_2			:INTEGER RANGE 284 DOWNTO 0;
		SIGNAL mem_ind_1		:INTEGER RANGE 0 TO 7;
		SIGNAL mem_ind_2		:INTEGER RANGE 0 TO 15;
		SIGNAL flag_tx			:BOOLEAN := false;
		SIGNAL end_tx_signal		:STD_LOGIC_VECTOR (0 TO 1);
		SIGNAL state_signal		:STD_LOGIC_VECTOR (0 TO 2);

		CONSTANT  id_tx			:STD_LOGIC_VECTOR (0 TO 7) := X"35";
		CONSTANT  id_rx			:STD_LOGIC_VECTOR (0 TO 7) := X"CA";
		CONSTANT  sync_trail_le		:STD_LOGIC_VECTOR (0 TO 7) := X"A2";
		CONSTANT  sync_trail_data	:STD_LOGIC_VECTOR (0 TO 7) := X"96";
		CONSTANT  payload_a		:STD_LOGIC_VECTOR (0 TO 15) := X"96D4";
		CONSTANT  payload_b		:STD_LOGIC_VECTOR (0 TO 7) := X"CA";
		CONSTANT  th_flag		:STD_LOGIC := '0';
		CONSTANT  th_code		:STD_LOGIC := '0';
		CONSTANT  fec_crc_le		:STD_LOGIC := '0';
		CONSTANT  PDU_number		:STD_LOGIC := '0';
		CONSTANT  N_pdu			:STD_LOGIC := '0';
		CONSTANT  ack			:STD_LOGIC := '0';
		CONSTANT  fec_crc_data		:STD_LOGIC := '0';

BEGIN

		PROCESS (RESET_TX_2, STATE, START_TX, CLK)
		BEGIN

		wait until CLK'event AND CLK = '1';

			IF START_TX = '1' THEN
				flag_tx <= true;
				count_1 <= 0;
				count_2 <= 0;
				mem_ind_1 <= 7;
				mem_ind_2 <= 15;
			END IF;

			IF STATE /= "000" THEN
				IF STATE = "111" THEN
					state_signal <= "000";
				ELSE
					state_signal <= STATE;
```

```
                END IF;
        END IF;

        IF (RESET_TX_2 = '1') THEN
                state_signal <= "000";
        END IF;

        IF (end_tx_signal = "01" OR end_tx_signal = "10") THEN
                end_tx_signal <= "00";
        END IF;

        CASE state_signal IS

                WHEN "000" =>
                        IF flag_tx THEN
                                CASE count_1 IS
                                        WHEN 0 to 7 =>
                                                DATA_TX <= sync_trail_le(mem_ind_1);
                                        WHEN 8 to 15 =>
                                                DATA_TX <= id_rx(mem_ind_1);
                                        WHEN 16 to 23 =>
                                                DATA_TX <= id_tx(mem_ind_1);
                                        WHEN 24 =>
                                                DATA_TX <= th_flag;
                                        WHEN 25 =>
                                                DATA_TX <= th_code;
                                        WHEN 26 =>
                                                DATA_TX <= fec_crc_le;
                                        WHEN 27 =>
                                                DATA_TX <= '0';
                                                end_tx_signal <= "01";
                                                flag_tx <= false;
                                        WHEN OTHERS =>
                                                NULL;
                                END CASE;
                                count_1 <= count_1 + 1;
                                mem_ind_1 <= mem_ind_1 - 1;
                        END IF;

                WHEN "010" =>
                        IF flag_tx THEN
                                CASE count_2 IS
                                        WHEN 0 to 7 =>
                                                DATA_TX <=
sync_trail_data(mem_ind_1);
                                                mem_ind_1 <= mem_ind_1 - 1;
                                        WHEN 8 to 15 =>
                                                DATA_TX <= id_rx(mem_ind_1);
                                                mem_ind_1 <= mem_ind_1 - 1;
                                        WHEN 16 to 23 =>
                                                DATA_TX <= id_tx(mem_ind_1);
                                                mem_ind_1 <= mem_ind_1 - 1;
                                        WHEN 24 =>
                                                DATA_TX <= PDU_number;
                                        WHEN 25 =>
                                                DATA_TX <= N_pdu;
                                        WHEN 26 =>
                                                DATA_TX <= ack;
                                        WHEN 27 TO 90 =>
                                                DATA_TX <= payload_a(mem_ind_2);
                                                mem_ind_2 <= mem_ind_2 - 1;
                                        WHEN 91 TO 154 =>
                                                DATA_TX <= payload_b(mem_ind_1);
                                                mem_ind_1 <= mem_ind_1 - 1;
                                        WHEN 155 TO 218 =>
                                                DATA_TX <= payload_a(mem_ind_2);
```

```
                                                    mem_ind_2 <= mem_ind_2 - 1;
                                                WHEN 219 TO 282 =>
                                                    DATA_TX <= payload_b(mem_ind_1);
                                                    mem_ind_1 <= mem_ind_1 - 1;
                                                WHEN 283 =>
                                                    DATA_TX <= fec_crc_data;
                                                WHEN 284 =>
                                                    end_tx_signal <= "10";
                                                    flag_tx <= false;
                                                WHEN OTHERS =>
                                                    NULL;
                                            END CASE;
                                            count_2 <= count_2 + 1;
                                        END IF;
                                    WHEN OTHERS =>
                                        NULL;
                                END CASE;

        END PROCESS;

END_TX <= end_tx_signal;

END A;
```

AIII.4. Masters: VHDL code of the block “Multiplexer”

```
-- MULTIPLEXER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY MULTIPLEXER_TX IS

	PORT
	(
		SEL			:IN	STD_LOGIC_VECTOR (0 TO 1);
		OUTPUT			:OUT	STD_LOGIC_VECTOR (7 DOWNTO 0)
	);

END MULTIPLEXER_TX;

ARCHITECTURE A OF MULTIPLEXER_TX IS

		CONSTANT  sync_trail_lc		:STD_LOGIC_VECTOR (0 TO 7) := X"8E";
		CONSTANT  sync_trail_ack	:STD_LOGIC_VECTOR (0 TO 7) := X"ED";
		CONSTANT  id_rx			:STD_LOGIC_VECTOR (0 TO 7) := X"CA";

BEGIN

	PROCESS (SEL)
	BEGIN

			CASE SEL IS
			WHEN "00" =>
				OUTPUT <= sync_trail_lc;
			WHEN "01" =>
				OUTPUT <= sync_trail_ack;
			WHEN "10" =>
				OUTPUT <= id_rx;
			WHEN OTHERS =>
				NULL;
			END CASE;

	END PROCESS;

END A;
```

AIII.5. Masters: VHDL code of the block “Compare Header”

```
-- COMPARE HEADER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY COMPARE_HEADER_TX IS

	PORT
	(
		DATA_A			:IN	STD_LOGIC_VECTOR (0 TO 7);
		DATA_B			:IN	STD_LOGIC_VECTOR (0 TO 7);
		CLEARN			:IN	STD_LOGIC;
		COMPARE			:OUT	STD_LOGIC_VECTOR (0 TO 1)
	);

END COMPARE_HEADER_TX;

ARCHITECTURE A OF COMPARE_HEADER_TX IS

BEGIN

	PROCESS (DATA_A, DATA_B, CLEARN)
	BEGIN

	IF(CLEARN = '1') THEN
		IF (DATA_A = DATA_B) THEN
			COMPARE <= "01";
		ELSIF (DATA_B = X"CA" AND DATA_A = X"C1") THEN
			COMPARE <= "11";
		ELSE
			COMPARE <= "00";
		END IF;
	ELSE
		COMPARE <= "00";
	END IF;

	END PROCESS;

END A;
```

AIII.6. Masters: VHDL code of the block “Vector to Integer Converter”

```
-- CONVERTER VECTOR BIT TO INTEGER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

ENTITY CONV_VECTOR_TO_INTEGER_TX IS

	PORT
	(
		PN_NUMBER_BIT		:IN	STD_LOGIC_VECTOR (0 TO 15);
		CLK			:IN	STD_LOGIC;
		PRESETN			:IN	STD_LOGIC;
		READ_PN_NUMBER		:IN	STD_LOGIC;
		TX_PACKET		:OUT	STD_LOGIC;
		COUNT_PACKET_TX		:OUT	INTEGER RANGE 0 TO 1023
	);

END CONV_VECTOR_TO_INTEGER_TX;

ARCHITECTURE A OF CONV_VECTOR_TO_INTEGER_TX IS

		SIGNAL pn_number_integer_signal		:INTEGER RANGE 0 TO 65535;
		SIGNAL tx_packet_signal			:STD_LOGIC;
		SIGNAL tx_signal			:STD_LOGIC;
		SIGNAL count_packet_tx_signal		:INTEGER RANGE 0 TO 1023;

BEGIN

	PROCESS (PN_NUMBER_BIT, READ_PN_NUMBER, PRESETN, CLK)
	BEGIN

	wait until CLK'event AND CLK = '1';
	IF (tx_packet_signal = '1') THEN
		tx_packet_signal <= '0';
		count_packet_tx_signal <= count_packet_tx_signal + 1;
	END IF;

	IF (READ_PN_NUMBER = '1' OR PRESETN = '0') THEN
		pn_number_integer_signal <= CONV_INTEGER(PN_NUMBER_BIT);
		tx_signal <= '1';
	END IF;

	IF (pn_number_integer_signal > 0 AND tx_signal = '1') THEN
		pn_number_integer_signal <= pn_number_integer_signal - 1;
	ELSIF (pn_number_integer_signal = 0 AND tx_signal = '1') THEN
		tx_packet_signal <= '1';
		tx_signal <= '0';
	END IF;

	END PROCESS;

TX_PACKET <= tx_packet_signal;
COUNT_PACKET_TX <= count_packet_tx_signal;

END A;
```

AIII.7. Slaves: VHDL code of the block “Recognize Header ”

```
-- RECOGNIZE_HEADER_BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY RECOGNIZE_HEADER_RX IS

        PORT
        (
                END_TX          :IN             STD_LOGIC_VECTOR (0 TO 1);
                COMPARE         :IN             STD_LOGIC;
                CLK             :IN             STD_LOGIC;
                PRESET          :IN             STD_LOGIC;
                LISTEN          :OUT            STD_LOGIC;
                STATE           :OUT            STD_LOGIC_VECTOR (0 TO 1);
                SEL             :OUT            STD_LOGIC_VECTOR (0 TO 1)
        );

END RECOGNIZE_HEADER_RX;

ARCHITECTURE A OF RECOGNIZE_HEADER_RX IS

                SIGNAL count                    :INTEGER RANGE 18 DOWNTO 0;
                SIGNAL recog_sync               :STD_LOGIC := '0';
                SIGNAL recog_id_rx              :STD_LOGIC := '0';
                SIGNAL sel_signal               :STD_LOGIC_VECTOR (0 TO 1);
                SIGNAL listen_signal            :STD_LOGIC;
                SIGNAL state_signal             :STD_LOGIC_VECTOR (0 TO 1);
                SIGNAL state_signal_aux         :STD_LOGIC_VECTOR (0 TO 1);
                SIGNAL timeout_signal           :INTEGER RANGE 19 TO 0;
                SIGNAL reset_rx_signal          :STD_LOGIC;

BEGIN

        PROCESS (END_TX, COMPARE, CLK, PRESET)
        BEGIN


                wait until CLK'event AND CLK = '1';

                        IF(PRESET = '0') THEN
                                listen_signal <= '1';
                                sel_signal <= "00";
                                state_signal_aux <= "00";
                                state_signal <= "00";
                        END IF;

                        IF (reset_rx_signal = '1') THEN
                                reset_rx_signal <= '0';
                                listen_signal <= '1';
                                sel_signal <= "00";
                                state_signal_aux <= "00";
                                state_signal <= "00";
                                recog_sync <= '0';
                                recog_id_rx <= '0';
                                count <= 0;
                        END IF;

                        IF state_signal_aux /= "00" THEN
                                state_signal_aux <= "00";
                        END IF;

                        CASE state_signal IS
```

```
		WHEN "00" | "01" =>
			IF (COMPARE = '1' AND recog_sync = '0') THEN
				count <= 0;
				recog_sync <= '1';
				sel_signal <= "10";
			ELSIF (COMPARE = '1' AND recog_sync = '1') THEN
				count <= count + 1;
				recog_id_rx <= '1';
				listen_signal <= '0';
				state_signal_aux <= "01";
				state_signal <= "01";
			ELSIF recog_sync = '1' THEN
				count <= count + 1;
			END IF;
			IF (count = 8 AND recog_id_rx = '0') THEN
				reset_rx_signal <= '1';
			ELSIF (count = 18) THEN
				recog_sync <= '0';
				recog_id_rx <= '0';
				count <= 0;
			END IF;
			IF END_TX = "01" THEN
				sel_signal <= "01";
				state_signal_aux <= "10";
				state_signal <= "10";
				listen_signal <= '1';
				timeout_signal <= 19;
			END IF;

		WHEN "10" =>
			IF (timeout_signal = 0) THEN
				reset_rx_signal <= '1';
			ELSIF (recog_sync = '0') THEN
				timeout_signal <= timeout_signal - 1;
			END IF;
			IF (COMPARE = '1' AND recog_sync = '0') THEN
				count <= 0;
				recog_sync <= '1';
				sel_signal <= "10";
			ELSIF (COMPARE = '1' AND recog_sync = '1') THEN
				count <= count + 1;
				recog_id_rx <= '1';
				listen_signal <= '0';
				state_signal_aux <= "11";
				state_signal <= "10";
			ELSIF recog_sync = '1' THEN
				count <= count + 1;
			END IF;
			IF (count = 8 AND recog_id_rx = '0') THEN
				reset_rx_signal <= '1';
			END IF;
			IF (END_TX = "10") THEN
				sel_signal <= "00";
				state_signal_aux <= "00";
				state_signal <= "00";
				listen_signal <= '1';
				recog_sync <= '0';
				recog_id_rx <= '0';
				count <= 0;
			END IF;

		WHEN OTHERS =>
			NULL;
END CASE;
```

```
        END PROCESS;

SEL <= sel_signal;
LISTEN <= listen_signal;
STATE <= state_signal_aux;

END A;
```

AIII.8. Slaves: VHDL code of the block “Buffer”

```
-- BUFFER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY BUFFER_RX IS

	PORT
	(
		END_TX			:IN		STD_LOGIC_VECTOR (0 TO 1);
		DATA, CLK		:IN		STD_LOGIC;
		STATE			:IN		STD_LOGIC_VECTOR (0 TO 1);
		PDU_number		:OUT		STD_LOGIC;
		START_TX		:OUT		STD_LOGIC;
		FEC_CRC_LE		:OUT		STD_LOGIC;
		PAYLOAD			:OUT		STD_LOGIC_VECTOR (0 TO 15);
		ID_TX			:OUT		STD_LOGIC_VECTOR (0 TO 7);
		OUTENAB			:OUT		STD_LOGIC
	);

END BUFFER_RX;

ARCHITECTURE A OF BUFFER_RX IS

		SIGNAL count_1			:INTEGER RANGE 9 DOWNTO 0;
		SIGNAL count_2			:INTEGER RANGE 268 DOWNTO 0;
		SIGNAL  flag_buffer		:BOOLEAN := false;
		SIGNAL mem_ind_1		:INTEGER RANGE 7 DOWNTO 0 := 0;
		SIGNAL mem_ind_payload		:INTEGER RANGE 15 DOWNTO 0 := 0;
		SIGNAL start_tx_signal		:STD_LOGIC := '0';
		SIGNAL state_signal		:STD_LOGIC_VECTOR (0 TO 1):= "00";
		SIGNAL outenab_signal		:STD_LOGIC := '0';

BEGIN

	PROCESS (END_TX, DATA, CLK, STATE)
	BEGIN

	wait until CLK'event AND CLK = '1';

		IF (STATE = "01" OR STATE = "11") THEN
			count_1 <= 0;
			count_2 <= 0;
			flag_buffer <= true;
			mem_ind_1 <= 0;
			mem_ind_payload <= 0;
		END IF;

		IF start_tx_signal <= '1' THEN
			start_tx_signal <= '0';
		END IF;

		IF STATE /= "00" THEN
			state_signal <= STATE;
		END IF;

		CASE state_signal IS
			WHEN "01" =>

				IF flag_buffer THEN
					CASE count_1 IS
						WHEN 0 to 7 =>
							ID_TX(mem_ind_1) <= DATA;
```

```
                                        mem_ind_1 <= mem_ind_1 + 1;
                                    WHEN 8 =>
                                        FEC_CRC_LE <= DATA;
                                    WHEN 9 =>
                                        flag_buffer <= false;
                                        start_tx_signal <= '1';
                                    WHEN OTHERS =>
                                        NULL;
                                END CASE;
                                count_1 <= count_1 + 1;
                            END IF;
                            IF (END_TX = "01") THEN
                                ID_TX <= (OTHERS => '0');
                            END IF;

                        WHEN "11" => NULL;

                            IF flag_buffer THEN
                                CASE count_2 IS
                                    WHEN 0 to 7 =>
                                        ID_TX(mem_ind_1) <= DATA;
                                        mem_ind_1 <= mem_ind_1 + 1;
                                    WHEN 8 =>
                                        PDU_number <= DATA;
                                    WHEN 9 | 10 =>
                                        NULL;
                                    WHEN 11 to 266 =>
                                        PAYLOAD(mem_ind_payload) <= DATA;
                                        IF outenab_signal = '1' THEN
                                            outenab_signal <= '0';
                                        END IF;
                                        IF mem_ind_payload = 15 THEN
                                            outenab_signal <= '1';
                                            mem_ind_payload <= 0;
                                        ELSE
                                            mem_ind_payload <= mem_ind_payload + 1;
                                        END IF;
                                    WHEN 267 =>
                                        outenab_signal <= '0';
                                    WHEN 268 =>
                                        flag_buffer <= false;
                                        start_tx_signal <= '1';
                                    WHEN OTHERS =>
                                        NULL;
                                END CASE;
                                count_2 <= count_2 + 1;
                            END IF;
                            IF (END_TX = "10") THEN
                                ID_TX <= (OTHERS => '0');
                                PDU_number <= '0';
                                PAYLOAD <= (OTHERS => '0');
                                FEC_CRC_LE <= '0';
                            END IF;

                        WHEN OTHERS =>
                            NULL;
                END CASE;

        END PROCESS;

START_TX <= start_tx_signal;
OUTENAB <= outenab_signal;

END A;
```

AIII.9. Slaves: VHDL code of the block "Parallel Serial Converter"

```
-- PARALLEL SERIAL CONVERTER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY PARALLEL_SERIAL_CONVERTER_RX IS

        PORT
        (
                DATA_status             :IN             STD_LOGIC;
                STATE                   :IN             STD_LOGIC_VECTOR (0 TO 1);
                FEC_CRC_LE              :IN             STD_LOGIC;
                START_TX                :IN             STD_LOGIC;
                PDU_number              :IN             STD_LOGIC;
                CLK                     :IN             STD_LOGIC;
                DATA_TX                 :OUT            STD_LOGIC;
                END_TX                  :OUT            STD_LOGIC_VECTOR (0 TO 1)
        );

END PARALLEL_SERIAL_CONVERTER_RX;

ARCHITECTURE A OF PARALLEL_SERIAL_CONVERTER_RX IS

                SIGNAL  count_1                 :INTEGER RANGE 25 DOWNTO 0 := 0;
                SIGNAL  count_2                 :INTEGER RANGE 27 DOWNTO 0 := 0;
                SIGNAL mem_ind_1                :INTEGER RANGE 0 TO 7 := 0;
                SIGNAL flag_tx                  :BOOLEAN := false;
                SIGNAL end_tx_signal            :STD_LOGIC_VECTOR (0 TO 1):= "00";
                SIGNAL state_signal             :STD_LOGIC_VECTOR (0 TO 1):= "00";

                CONSTANT  id_tx                 :STD_LOGIC_VECTOR (0 TO 7) := X"35";
                CONSTANT  id_rx                 :STD_LOGIC_VECTOR (0 TO 7) := X"CA";
                CONSTANT  sync_trail_lc         :STD_LOGIC_VECTOR (0 TO 7) := X"8E";
                CONSTANT  sync_trail_ack        :STD_LOGIC_VECTOR (0 TO 7) := X"ED";
BEGIN

        PROCESS (DATA_status, STATE, FEC_CRC_LE, START_TX, PDU_number, CLK)
        BEGIN

                wait until CLK'event AND CLK = '1';

                        IF START_TX = '1' THEN
                                flag_tx <= true;
                                count_1 <= 0;
                                count_2 <= 0;
                                mem_ind_1 <= 7;
                        END IF;

                        IF (state /= "00") THEN
                                state_signal <= STATE;
                        END IF;

                        IF (end_tx_signal = "01" OR end_tx_signal = "10") THEN
                                end_tx_signal <= "00";
                        END IF;

                        CASE state_signal IS
                                WHEN "01" =>

                                        IF flag_tx THEN
                                                CASE count_1 IS
```

```
                                                WHEN 0 to 7 =>
                                                        DATA_TX <= sync_trail_lc(mem_ind_1);
                                                WHEN 8 to 15 =>
                                                        DATA_TX <= id_rx(mem_ind_1);
                                                WHEN 16 to 23 =>
                                                        DATA_TX <= id_tx(mem_ind_1);
                                                WHEN 24 =>
                                                        DATA_TX <= FEC_CRC_LE;
                                                WHEN 25 =>
                                                        DATA_TX <= '0';
                                                        end_tx_signal <= "01";
                                                        flag_tx <= false;
                                                WHEN OTHERS =>
                                                        NULL;
                                        END CASE;
                                        count_1 <= count_1 + 1;
                                        mem_ind_1 <= mem_ind_1 - 1;
                                END IF;

                        WHEN "11" =>

                                IF flag_tx THEN
                                        CASE count_2 IS
                                                WHEN 0 to 7 =>
                                                        DATA_TX <=
sync_trail_ack(mem_ind_1);
                                                        mem_ind_1 <= mem_ind_1 - 1;
                                                WHEN 8 to 15 =>
                                                        DATA_TX <= id_rx(mem_ind_1);
                                                        mem_ind_1 <= mem_ind_1 - 1;
                                                WHEN 16 to 23 =>
                                                        DATA_TX <= id_tx(mem_ind_1);
                                                        mem_ind_1 <= mem_ind_1 - 1;
                                                WHEN 24 =>
                                                        DATA_TX <= PDU_number;
                                                WHEN 25 =>
                                                        DATA_TX <= DATA_status;
                                                WHEN 26 =>
                                                        DATA_TX <= FEC_CRC_LE;
                                                WHEN 27 =>
                                                        DATA_TX <= '0';
                                                        end_tx_signal <= "10";
                                                        flag_tx <= false;
                                                WHEN OTHERS =>
                                                        NULL;
                                        END CASE;
                                        count_2 <= count_2 + 1;
                                END IF;
                        WHEN OTHERS =>
                                NULL;
                END CASE;

        END PROCESS;

END_TX <= end_tx_signal;

END A;
```

AIII.10. Slaves: VHDL code of the block “Multiplexer”

```
-- MULTIPLEXER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY MULTIPLEXER_RX IS

	PORT
	(
		SEL			:IN		STD_LOGIC_VECTOR (0 TO 1);
		OUTPUT			:OUT		STD_LOGIC_VECTOR (7 DOWNTO 0)
	);

END MULTIPLEXER_RX;

ARCHITECTURE A OF MULTIPLEXER_RX IS

		CONSTANT	sync_trail_le		:STD_LOGIC_VECTOR (0 TO 7) := X"A2";
		CONSTANT	sync_trail_data		:STD_LOGIC_VECTOR (0 TO 7) := X"96";
		CONSTANT	id_rx			:STD_LOGIC_VECTOR (0 TO 7) := X"CA";

BEGIN

	PROCESS (SEL)
	BEGIN

			CASE SEL IS

			WHEN "00" =>
				OUTPUT <= sync_trail_le;
			WHEN "01" =>
				OUTPUT <= sync_trail_data;
			WHEN "10" =>
				OUTPUT <= id_rx;
			WHEN OTHERS =>
				NULL;

			END CASE;

	END PROCESS;

END A;
```

AIII.11. Slaves: VHDL code of the block “Compare Header”

```
-- COMPARE  HEADER  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY COMPARE_HEADER_RX IS

	PORT
	(
		DATA_A		:IN		STD_LOGIC_VECTOR (0 TO 7);
		DATA_B		:IN		STD_LOGIC_VECTOR (0 TO 7);
		CLEARN		:IN		STD_LOGIC;
		COMPARE		:OUT		STD_LOGIC
	);

END COMPARE_HEADER_RX;

ARCHITECTURE A OF COMPARE_HEADER_RX IS

BEGIN

	PROCESS (DATA_A, DATA_B, CLEARN)
	BEGIN

	IF(CLEARN = '1') THEN
		IF (DATA_A = DATA_B) THEN
			COMPARE <= '1';
		ELSE
			COMPARE <= '0';
		END IF;
	ELSE
		COMPARE <= '0';
	END IF;

	END PROCESS;

END A;
```

AIII.12. Slaves: VHDL code of the block “Compare Payload”

```
-- COMPARE  PAYLOAD  BLOCK

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY COMPARE_PAYLOAD_RX IS

	PORT
	(
		PAYLOAD_16		:IN		STD_LOGIC_VECTOR (0 TO 15);
		ID_TX			:IN		STD_LOGIC_VECTOR (0 TO 7);
		OUTENAB			:IN		 STD_LOGIC;
		STATE			:IN		STD_LOGIC_VECTOR (0 TO 1);
		CLK			:IN		STD_LOGIC;
		DATA_STATUS		:OUT		STD_LOGIC;
		COUNT_ERROR		:OUT		INTEGER RANGE 0 TO 100
	);

END COMPARE_PAYLOAD_RX;

ARCHITECTURE A OF COMPARE_PAYLOAD_RX IS

		SIGNAL  flag				:BOOLEAN;
		SIGNAL  count_error_signal		:INTEGER RANGE 0 TO 100;

		CONSTANT	payload_a	:STD_LOGIC_VECTOR (0 TO 15) := X"96D4";
		CONSTANT	payload_b	:STD_LOGIC_VECTOR (0 TO 7) := X"CA";


BEGIN
	PROCESS (PAYLOAD_16, ID_TX, OUTENAB, STATE, CLK)
	BEGIN

		wait until CLK'event AND CLK = '1';

		IF STATE = "11" THEN
			flag <= true;
			DATA_STATUS <= '0';
		END IF;


		IF OUTENAB = '1' THEN

			IF flag THEN

				IF ID_TX /= X"35" THEN
					flag <= false;
					DATA_STATUS <= '0';
					count_error_signal <= count_error_signal + 1;
				ELSIF (PAYLOAD_16 = payload_a) THEN
					DATA_STATUS <= '1';
				ELSIF (PAYLOAD_16 = payload_b & payload_b) THEN
					DATA_STATUS <= '1';
				ELSE
					flag <= false;
					DATA_STATUS <= '0';
					count_error_signal <= count_error_signal + 1;
				END IF;

			END IF;
```

```
                    END IF;
            END PROCESS;
COUNT_ERROR <= count_error_signal;
END A;
```

Annex IV. Acronyms

Contents

AIV.1 Acronyms 268

AIV.1. Acronyms

ACK	*Acknowledgement*
A/D	*Analogic/Digital*
AJ	*Anti Jam*
APD	*Avalanche Photodiode*
ARQ	*Automatic Repeat Request*
AWGN	*Additive White Gaussian Noise*
BER	*Bit Error Rate*
BPF	*Band Pass Filter*
BPS	*Bits Per Second*
BPSK	*Binary Phase Shift Keying*
BTMA	*Busy Tone Multiple Access*
BW	*Band Width*
CA (CSMA)	*Collision Avoidance*
CD (CSMA)	*Collision Detection*
CDMA	*Code Division Multiple Access*
CRC	*Cyclic Redundancy Check*
CSMA	*Carrier Sense Multiple Access*
CTS	*Clear To Send*
D/A	*Digital/Analogic*
DBTMA	*Dual BTMA*
DCF	*Distributed Coordination Function*
DIFS	*DCF Interframe Space*
DFWMAC	*Distributed Foundation Wireless MAC*
DLC	*Data Link Control*
DS	*Direct Sequence*
DSSS	*Direct Sequence Spread Spectrum*
EPLD	*Erasable Programmable Logic Device*
FCC	*Federal Communication Commission*
FDMA	*Frequency Division Multiple Access*
FEC	*Forward Error Correction*
FDTD	*Finite Differences Time Domain*
FHSS	*Frequency Hopping Spread Spectrum*
FIR (IrDA)	*Fast InfraRed*
FOV	*Field Of View*
FPGA	*Field Programmable Gate Array*
FWM	*Four Wave Mixing*
GMSK	*Gaussian Minimum Shift Keying*
GPS	*Global Positioning System*
GTFC	*Grupo de Tecnología Fotónica y Comunicaciones (ULPGC)*
HPF	*High Pass Filter*
ID	*Identity code*
IEEE	*Institute of Electrical and Electronics Engineers*

IM	*Intensity Modulation*
IrDA	*InfraRed Data Association*
IRED	*InfraRed Emitting Diode*
ISM	*Industrial Scientific and Medical*
ISO	*International Standardization Organization*
ISP	*In System Programmability*
JTAG	*Joint Test Action Group*
LAN	*Local Area Network*
LASER	*Light Amplification by Stimulated Emission of Radiation*
LED	*Light Emitting Diode*
LC PDU	*Link Confirm PDU*
LE PDU	*Link Establish PDU*
LLC	*Logical Link Control*
LOS	*Line Of Sight*
LPF	*Low Pass Filter*
LPI	*Low Probability of Intercept*
MAC	*Medium Access Control*
MACA	*Medium Access with Collision Avoidance*
MAN	*Metropolitan Area Network*
MB	*Multi Band*
MIR (IrDA)	*Medium InfraRed*
ML	*Maximum Likelihood*
MOM	*Method Of Moment*
MUI	*Multi User Interference*
NAV	*Net Allocation Vector*
NRZ	*Non Return to Zero*
NTIA	*National Telecommunications and Information Administration*
OCS	*Optical Communication System*
OFDM	*Orthogonal Frequency Division Multiplex*
OOC	*Optical Orthogonal Codes*
OOK	*On Off Keying*
OSI	*Open System Interconnection*
PAM	*Pulse Amplitude Modulation*
PCF	*Point Coordination Function*
PDU	*Protocol Data Unit*
PHY	*Physical layer*
PIFS	*PCF Interframe Space*
PIN	*Positive Intrinsic Negative*
PNC	*PicoNet Coordinator*
PPM	*Pulse Position Modulation*
PRNG	*Pseudo Random Number Generator*
PRMA	*Packet Reservation Multiple Access*
PSD	*Power Spectral Density*
QAM	*Quadrature Amplitude Modulation*

RF	*Radio Frequency*
RTS	*Request To Send*
SER	*Symbol Error Rate*
SIFS	*Short Interframe Space*
SIR (IrDA)	*Serial InfraRed*
SIR	*Signal to Interference Ratio*
SNR	*Signal to Noise Ratio*
SNIR	*Signal to Noise plus Interference Ratio*
SPM	*Self Phase Modulation*
SS	*Spread Spectrum*
STI	*Simulation Time Interval*
TDMA	*Time Division Multiple Access*
TH	*Time Hopping*
THSS	*Time Hopping Spread Spectrum*
USB	*Universal Serial Bus*
UFIR (IrDA)	*Ultra Fast InfraRed*
UWB	*Ultra Wide Band*
VFIR (IrDA)	*Very Fast InfraRed*
WDMA	*Wavelength Division Multiple Access*
WPAN	*Wireless Personal Area Networks*
XOR	*eXclusive OR*

Author's address:
Eugenio Manlio Ducci
Via Cagliari, 29 – 95127 Catania, Italy
Web telephone: 06452218954
E-mail: em.ducci@gmail.com