



**João Ricardo
Borges dos Santos**

**Leitor RFID para a gama UHF baseado em
Software-Defined Radio**

Aos meus Pais,



**João Ricardo
Borges dos Santos**

**Leitor RFID para a gama UHF baseado em
Software-Defined Radio**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Arnaldo Oliveira e do Professor Doutor Nuno Borges de Carvalho, Professores do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor José Carlos Esteves Duarte Pedro

Professor Catedrático da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Professor Doutor João Paulo de Castro Canas Ferreira

Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto (arguente)

Professor Doutor Arnaldo Silva Rodrigues de Oliveira

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

Agradeço aos meus pais por todas as oportunidades que me proporcionaram ao longo da minha vida, assim como, todo o apoio incondicional (nos momentos bons e menos bons) que me ofereceram nesta etapa académica. Agradeço também à minha irmã, cunhado e especialmente ao meu sobrinho, pelo apoio e momentos de distração e lazer.

Um especial agradecimento à Sofia por toda a sua paciência e amor infinito. Por me “aturar” nos momentos de maior *stress*, conseguindo, sempre com sucesso, manter-me com os pés assentes na Terra, assim como a sua preciosa ajuda durante a escrita deste documento.

Aos meus orientadores Professor Arnaldo Oliveira e Professor Nuno Borges pelas oportunidades, motivação e conhecimento transmitido. Ao André Prata, Diogo Ribeiro, Daniel Dinis, Jorge Santos, Diogo Riscado, Sérgio Alves e Alírio Boaventura por todo o apoio e disponibilidade, apresentada durante a realização deste trabalho de dissertação de mestrado.

Agradeço também a todos os meus amigos, pois todos eles fazem parte do quem eu sou.

Ao Instituto de Telecomunicações e à Universidade de Aveiro por todas as condições oferecidas para este trabalho se tornar numa realidade.

A todos, um Muito Obrigado!

Palavras-Chave

Software-Defined Radio, RFID, ISO 18000-6C, EPCgen2, ZedBoard, FM-COMMS1

Resumo

Esta dissertação apresenta a arquitetura, o desenvolvimento e a validação de um leitor de RFID para a banda de frequências de 860 MHz a 960 MHz (UHF). Na sua conceção foi usada uma abordagem baseada em sistemas de rádio definidos por *software* (SDR-*Software-Defined Radio*).

O RFID é uma tecnologia que permite a identificação de objetos através de ondas eletromagnéticas. Um leitor RFID, também designado por *reader*, emite um sinal eletromagnético para uma ou várias *tags*, e estas respondem (também na forma de um sinal eletromagnético) com um número de identificação único. Esta tecnologia é muito versátil e pode ser aplicada às mais diversas situações, tais como, identificação de animais, acesso a edifícios e identificação de objetos num armazém. Podem ser usadas diferentes bandas de frequência do espectro eletromagnético, tais como, as *low frequencies* (LF), as *high frequencies* (HF) e as *ultra high frequencies* (UHF).

O SDR é um paradigma de projeto de sistemas de rádio, onde os blocos analógicos responsáveis pelo processamento de sinal, são total ou parcialmente substituídos por blocos programáveis de processamento no domínio digital. Este conceito permite aos sistemas de rádio resultantes apresentarem uma maior flexibilidade e adaptabilidade face a alterações que possam surgir na arquitetura ou requisitos do sistema. A digitalização num sistema SDR pode ser efetuada em banda base, em frequência intermédia ou em RF.

O sistema de rádio presente nesta dissertação consiste num leitor RFID funcional para o protocolo de comunicação RFID ISO 18000-6C, implementado numa arquitetura SDR com digitalização em banda base, na cadeia de transmissão e digitalização em frequência intermédia, na cadeia de receção.

Keywords

Software-Defined Radio, RFID, ISO 18000-6C, EPCgen2, ZedBoard, FM-COMMS1

Abstract

This MSc thesis presents the architecture, development and validation of a RFID reader for the frequency band of 860 MHz to 960 MHz (UHF). In its design, a Software-Defined Radio (SDR) approach was used.

RFID is a technology that allows the identification of objects using electromagnetic waves. A RFID reader emits an electromagnetic signal to one or more tags, and these respond (also in the form of an electromagnetic signal) with a unique identification number. This technology is very versatile and can be applied to various situations, such as animal identification, building access and identification of objects in a warehouse. It can be used in different frequency bands of the electromagnetic spectrum, such as the low frequencies (LF), the high frequencies (HF) and ultra high frequencies (UHF).

The SDR is a design paradigm of radio systems, in which analog blocks responsible for signal processing, are partially or completely replaced by programmable processing blocks in the digital domain. This concept allows the resulting radio systems to offer greater flexibility and adaptability to changes that occur in the architecture or system requirements. The digitalization in a SDR system can be made at baseband, intermediate frequency (IF) or at RF.

The radio system, presented in this dissertation, is a functional RFID reader working on the protocol communication ISO 18000-6C and implemented with a baseband digitization SDR architecture at the transmission path, and a IF digitalization at the receiver path.

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	ix
Lista de Acrónimos	xi
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	3
1.3 Objetivos	6
1.4 Organização do Documento	7
2 RFID	9
2.1 Breve Perspetiva Histórica	9
2.2 Classificação dos Sistemas RFID	10
2.3 Protocolos de Comunicação para Sistemas RFID	11
2.4 EPC Generation-2 Class 1 UHF RFID	12
2.4.1 <i>Electronic Product Code</i> - EPC	12
2.4.2 Descrição e Funcionamento	13
2.4.3 Comunicação Reader para Tag (R→T)	17
Modulação do Sinal Transmitido	17
Codificação de Informação	18
Preamble e Frame-sync	18
Máscaras de Potência Transmitida	19
2.4.4 Comunicação Tag para Reader (T→R)	20
Codificação de Informação	20
Codificação FM0	21
Preamble FM0	22
Data-rates Suportados pelas Tags	23
2.4.5 Códigos de Redundância Cíclicos	23
2.4.6 Comandos do <i>Reader</i> e Respostas da <i>Tag</i>	24
• Query	24
• QueryAdjust	25
• QueryRep	26
• NAK	27

	• ACK	28
2.4.7	<i>Link Timing</i>	29
3	Arquitetura de um Sistema RFID em SDR	31
3.1	Introdução	31
3.2	Cadeia de Transmissão	32
3.2.1	Transmissores SDR	33
3.3	Cadeia de Recepção	34
3.3.1	Recetores SDR	34
3.4	Software-Defined Radio aplicado a RFID	36
3.4.1	Cadeia de Transmissão	36
3.4.2	Cadeia de Recepção	37
3.4.3	Cadeia de Recepção e Transmissão no contexto deste Trabalho	37
4	Implementação do Sistema RFID	39
4.1	Introdução	39
4.2	Camada Física Analógica - RF Front-end	40
4.2.1	Cadeia de Transmissão	42
	Bloco DAC - Reconstrução do Sinal R→T	42
	Bloco Modulador I/Q - Sinal <i>Single Side Band</i>	43
	Bloco Amplificador de Potência	44
	Bloco Antena	44
4.2.2	Cadeia de Recepção	45
	Bloco Antena	46
	Bloco Desmodulador I/Q	46
	Bloco Amplificador de Baixo Ruído	47
	Bloco ADC - Amostragem do Sinal T→R	48
4.2.3	Cadeia de Relógios	48
4.2.4	Programação dos Componentes	48
4.3	Camada Física Digital	49
4.3.1	Cadeia de Transmissão	50
	• Validação da Cadeia de Transmissão em Matlab	52
	• Implementação da Cadeia de Transmissão na FPGA	56
4.3.2	Cadeia de Recepção	56
	• Validação da Cadeia de Recepção em Matlab	57
	Quadrador (<i>signal power by 2</i>)	58
	Remoção das Componentes de Alta frequência e decimação do sinal	60
	Remoção da componente DC	63
	Transformada de Hilbert	66
	Descodificação FM0	68
	• Implementação da Cadeia de Recepção na FPGA	70
4.4	Camada MAC	71
4.4.1	Funções Desenvolvidas em C	72

5	Resultados	75
5.1	Introdução	75
5.2	Alteração da Frequência de Operação	76
5.3	Máscara de Transmissão de Potência	81
5.4	Alcance Máximo de Leitura por parte do <i>Reader</i>	82
5.4.1	Método: d constante e P_{reader} variável	82
5.4.2	Método: P_{reader} constante e d variável.	83
5.4.3	Resultados Obtidos	84
5.5	Desempenho do Sistema.	85
5.5.1	Ambiente <i>Single-tag</i>	85
5.5.2	Ambiente <i>Multi-tag</i>	87
5.5.3	Comparação Ambiente <i>Single-Tag</i> com <i>Multi-tag</i>	87
5.6	Sinais RFID Obtidos	89
5.7	Recursos de Implementação em FPGA	91
5.8	Estimativa do Consumo Energético	92
5.9	Estimativa do Custo do Sistema Implementado	93
6	Conclusão e Trabalho Futuro	95
6.1	Conclusão	95
6.2	Trabalho Futuro	96
A	Diagrama de Blocos do Sistema Final	99
B	Sistema Final Implementado	101
C	Teorema de Nyquist	104
D	Modulação ASK	106
E	Filtros FIR	110
	Bibliografia	111

Lista de Figuras

1.1	Esquema de funcionamento de um sistema RFID.	1
1.2	Arquitetura típica de um <i>reader</i>	2
1.3	Arquitetura de uma FPGA.	4
1.4	Distribuição mundial das frequências de operação UHF para o RFID.	4
1.5	Regulamentação imposta à tecnologia RFID pela FCC	5
1.6	Regulamentação imposta à tecnologia RFID pela ETSI	6
2.1	Evolução do RFID.	9
2.2	Bandas de operação do RFID	11
2.3	Exemplo de um EPC	12
2.4	Esquema simplificado do funcionamento do protocolo EPC Gen-2.	13
2.5	Esquema simplificado do funcionamento do protocolo EPC Gen-2 na obtenção do EPC de cada <i>tag</i>	14
2.6	Esquema simplificado do funcionamento do protocolo EPC Gen-2 na situação onde estão presentes, no mesmo ambiente, múltiplas <i>tags</i> e múltiplos <i>readers</i>	14
2.7	Esquema simplificado do funcionamento do protocolo EPC Gen-2 na situação onde o <i>reader</i> modifica o EPC de uma <i>tag</i> específica.	15
2.8	Início de uma <i>Inventory Round</i>	16
2.9	Esquema de funcionamento do protocolo ISO 18000-6C.	17
2.10	Símbolos PIE	18
2.11	<i>Preamble</i> e <i>Frame-sync</i>	19
2.12	Máscara de transmissão em ambientes <i>multiple-interrogator</i>	20
2.13	Máscara de transmissão em ambientes <i>dense-interrogator</i>	21
2.14	Funções base e diagrama de estados da codificação FM0	21
2.15	Preamble FM0	22
2.16	Tempos a respeito	30
3.1	Arquitetura ideal SDR. Tx-Cadeia de transmissão; Rx-Cadeia de recepção.	31
3.2	Arquitetura de transmissão homodina.	32
3.3	Arquitetura de transmissão heterodina.	32
3.4	Arquitetura de transmissão SDR sinal é gerado numa componente IF.	33
3.5	Arquitetura de transmissão SDR sinal é gerado numa componente RF.	34
3.6	Arquitetura de transmissão SDR completamente digital.	34
3.7	Arquitetura de recepção heterodina SDR com digitalização em banda base.	35
3.8	Arquitetura de recepção homodina SDR com digitalização em banda base.	35
3.9	Arquitetura de recepção SDR com digitalização em IF.	36

3.10	Arquitetura de recepção SDR com digitalização em RF.	36
3.11	Arquitetura de recepção e transmissão SDR no contexto desta dissertação de mestrado.	37
4.1	Blocos do sistema RFID implementado.	40
4.2	Placa de desenvolvimento <i>Front-End AD-FMCOMMS1-EBZ</i>	40
4.3	Diagrama de blocos da placa <i>Front-End AD-FMCOMMS1-EBZ</i>	41
4.4	Diagrama de blocos da cadeia de transmissão.	42
4.5	Arquitetura para gerar um sinal SSB	43
4.6	Sinal SSB no espectro de frequências.	44
4.7	Antena desenvolvida no IT, pólo de Aveiro.	45
4.8	Diagrama de blocos da cadeia de recepção.	45
4.9	Antena ALR-8610-AC <i>Alien</i>	46
4.10	Arquitetura de um recetor I/Q	47
4.11	Desmodulação no espectro de frequências.	47
4.12	Placa de desenvolvimento <i>Zedboard</i>	49
4.13	Diagrama de blocos da placa <i>Zedboard</i>	50
4.14	Diagrama de blocos cadeia de transmissão implementada na FPGA.	51
4.15	Sequência de de oito símbolos “Data-0” gerados em Matlab.	52
4.16	Diferenças espectrais entre um sinal ASK com transição abrupta e transição suave	53
4.17	(a) Resposta impulsional do filtro em função da frequência. (b) Amplitude da fase do filtro em função da frequência	54
4.18	Sinal original e sinal após a passagem do filtro passa-baixo.	55
4.19	(a) <i>Up-conversion</i> do sinal antes do filtro passa-baixo. (b) <i>Up-conversion</i> do sinal após o filtro passa-baixo.	55
4.20	Diagrama de blocos cadeia de recepção implementada na FPGA.	57
4.21	Sinal em IF amostrado pela ADC	58
4.22	Sinal em IF elevado ao quadrado	59
4.23	(a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência	61
4.24	(a) Cosseno original com 126 amostras (b) Mesmo cosseno decimado por 2 com 63 amostras.	62
4.25	Envolvente do sinal após filtragem e decimação	62
4.26	(a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência	64
4.27	Envolvente do sinal após filtragem e atenuação do DC	64
4.28	Resposta impulsional de um derivador	65
4.29	(a) <i>group delay</i> do filtro FIR passa-alto implementado. (b) Resposta impulsional do mesmo filtro.	65
4.30	(a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência.	66
4.31	Resposta impulsional da transformada de Hilbert.	67
4.32	Recuperação da envolvente do sinal após filtragem e atenuação do DC	67
4.33	Esquema de descodificação FM0 no sistema implementado.	69
4.34	Simulação Matlab da deteção de transições por zero.	69
4.35	Desvio de frequência no sinal <i>backscattering</i> enviado pela <i>tag</i>	70

4.36	Diagrama de blocos do decodificador implementado.	71
4.37	Máquina de estados implementada na camada MAC.	72
5.1	Esquema de medição utilizado.	75
5.2	Frequências de operação para RFID UHF estabelecido pela ANACOM. . . .	76
5.3	<i>Setup</i> de medidas utilizado.	76
5.4	Espectro do sinal a 865.7 MHz.	77
5.5	Espectro do sinal a 866.3 MHz.	77
5.6	Espectro do sinal a 866.9 MHz.	78
5.7	Espectro do sinal a 867.5 MHz.	78
5.8	Os quatro canais de transmissão implementados no sistema.	80
5.9	Sinal implementado comparado com a máscara de transmissão de potência . .	81
5.10	Esquema de medição utilizado	82
5.11	Esquema de medição utilizado para obter P_{min}	83
5.12	Esquema de medição utilizado para obter d_{max}	84
5.13	Esquema de medição utilizado no ambiente <i>multi-tag</i>	85
5.14	<i>Tags</i> utilizadas nas medições	86
5.15	Comparação do desempenho do <i>reader</i> entre ambientes <i>multi-tag</i> e <i>single-tag</i> . .	88
5.16	Comunicação entre <i>reader</i> e <i>tag</i>	89
5.17	Sinal RN16 após transformada de Hilbert implementada na cadeia de recepção. .	90
5.18	Sinal EPC após transformada de Hilbert implementada na cadeia de recepção. .	90
5.19	Porcentagem de recursos da FPGA utilizados pelo sistema implementado. . . .	91
5.20	Consumo energético “ <i>on-chip</i> ” do sistema implementado na FPGA.	92
A.1	Diagrama de Blocos do Sistema Implementado.	100
B.1	Sistema Implementado.	101
B.2	Detalhe Zedboard mais Front-End do Sistema Implementado	102
B.3	Detalhe antenas de transmissão e recepção do Sistema Implementado	103
C.1	Processo de amostragem	104
C.2	Processo de amostragem no domínio do tempo.	105
D.1	Variantes da modulação ASK	106
D.2	Modulação ASK com três tipos de profundidade diferentes	107
D.3	Espectro da modulação DSB-ASK	108
D.4	Espectro da modulação SSB-ASK	108
D.5	Espectro da modulação PR-ASK	109
E.1	Diagrama de blocos de um filtro FIR. z^{-1} representa um atraso temporal . .	110

Lista de Tabelas

2.1	<i>Standards</i> de comunicação RFID	11
2.2	T→R BLF.	23
2.3	Parâmetros CRC-16.	24
2.4	Parâmetros CRC-5.	24
2.5	(a) Trama do comando <i>Query</i> . (b) Resposta da <i>tag</i> ao comando <i>Query</i>	25
2.6	(a) Trama do comando <i>QueryAdjust</i> . (b) Resposta da <i>tag</i> ao comando <i>QueryAdjust</i>	26
2.7	(a) Trama do comando <i>QueryRep</i> . (b) Resposta da <i>tag</i> ao comando <i>QueryRep</i>	27
2.8	Trama do comando <i>NAK</i>	27
2.9	(a) Trama do comando <i>ACK</i> . (b) Resposta da <i>tag</i> ao comando <i>ACK</i>	28
2.10	Tempos relevantes para o protocolo de comunicação EPC Gen2 Class 1.	29
4.1	Parâmetros temporais da cadeia de transmissão.	52
4.2	caraterísticas do filtro FIR passa-baixo.	53
4.3	caraterísticas do filtro FIR passa-baixo.	60
4.4	caraterísticas do filtro FIR passa-alto.	63
4.5	Parâmetros de programação da <i>tag</i>	68
5.1	caraterísticas do sinal transmitido	79
5.2	Resultados obtidos para a medição d constante e P_{reader} variável.	84
5.3	Resultados obtidos para a medição P_{reader} constante e d variável.	84
5.4	Performance do sistema num ambiente <i>single-tag</i>	86
5.5	Número de leituras obtidas para cada <i>tag</i> num ambiente <i>multi-tag</i>	87
5.6	Recursos da FPGA utilizados pelo sistema implementado.	92
5.7	Estimativa do custo monetário do sistema implementado.	93

Lista de Acrónimos

A/D	Analógico para Digital
ADC	Analog-to-Digital Converter
ANACOM	Autoridade Nacional de Comunicações
ASK	Amplitude Shift Keying
ARM	Advanced RISC Machine
BLF	Backscatter Link Frequency
BRAM	Block Random Access Memory
BUFG	Global Clock Buffer
CRC	Cyclic Redundancy Check
D/A	Digital para Analógico
DAC	Digital-to-Analog Converter
DDR	Double Data Rate
DDS	Direct Digital Synthesis
DR	Divide Ratio
DSB	Double-Sideband
DSB-ASK	Double-Sideband Amplitude Shift Keying
DSP	Digital Signal Processor
EIRP	Equivalent Isotropically Radiated Power
EPC	Electronic Product Code
EPP	Extensible Processing Platform
ERP	Effective Radiated Power
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FF	Flip-Flop
FHSS	Frequency Hopping Spread Spectrum
FIR	Finite Pulse Response
FM0	Bi-phase space

FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FrT	Frequency Tolerance
HF	High Frequency
ID	IDentification
IEEE	Institute of Electrical and Electronic Engineers
IF	Intermediate Frequency
IFF	Identify Friend or Foe
IIC	Inter-Integrated Circuit
ISM	Industrial, Scientific and Medical
ISO	International Standards Organization
I2C	Inter-Integrated Circuit
I/O	Input/Output
I/Q	In phase / Quadrature
LF	Low Frequency
LNA	Low Noise Amplifier
LO	Local Oscillator
LPC	Low Pin Count
LSB	Lower Side Band
LVDS	Low Voltage Differential Signaling
LUT	Look-up Table
MAC	Media Access Control
MIFARE	MIkron FARE collection system
MMCM	Mixed-Mode Clock Manager
OOK	On-Off Keying
PC	Protocol Control
PHY	Physical
PIE	Pulse Interval Encoding
PR-ASK	Phase-Reversal Amplitude Shift Keying
PS	Processing System
PW	Pulse Width
RADAR	RAdio Detection And Ranging
RAM	Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency IDentification

RN16	16-bit Random or pseudo-Random Number
RTcal	Reader to Tag Calibration
SDR	Software-Defined Radio
SPI	Serial Peripheral Interface
SSB	Single-Sideband
SSB-ASK	Single-Sideband Amplitude Shift Keying
SW	Switch
Tari	Type A Reference Interval
TRcal	Tag to Reader Calibration
UHF	Ultra High Frequency
USB	Upper Side Band
VHDL	Very high speed integrated circuits Hardware Description Language

Capítulo 1

Introdução

1.1 Enquadramento

Nos últimos anos tem-se assistido a um interesse crescente, tanto a nível académico como industrial, pelos sistemas de identificação por rádio frequência, ou em inglês, *Radio Frequency IDentification* (RFID), em aplicações de monitorização e localização. Este interesse deve-se, essencialmente, ao facto desta tecnologia apresentar uma solução barata e flexível na identificação de objetos. A título de exemplo, supondo uma empresa que possua um conjunto de armazéns, facilmente se conclui que, com uma rápida e eficiente leitura de etiquetas ou *tags*, associadas aos mais variados objetos, a produtividade dessa (empresa) vai aumentar consideravelmente, uma vez que a velocidade de operações, tais como, a localização e o controlo do *stock*, vai aumentar. Temos também exemplos de aplicações desta tecnologia aplicada no dia a dia do cidadão comum, tais como cartões de acesso a transportes públicos e edifícios, pagamentos eletrónicos, identificação de animais domésticos, alarmes antirroubo em lojas, localização de carros em parques de estacionamento, localização de livros em bibliotecas, entre outros.

Um sistema RFID é constituído por um ou múltiplos leitores (*readers*) que tentam identificar, através de ondas eletromagnéticas, as *tags* associadas a objetos. O leitor é a entidade do sistema com maior complexidade, sendo as *tags* dispositivos mais simples.

A figura 1.1 representa o esquema de funcionamento de um sistema RFID.

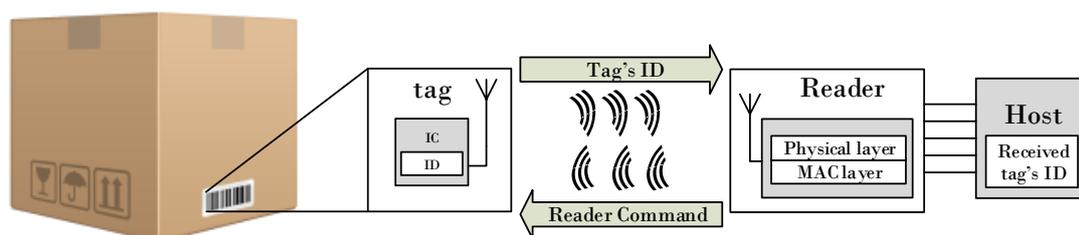


Figura 1.1: Esquema de funcionamento de um sistema RFID. O *reader* envia um comando à *tag* e esta responde com o seu identificador. A comunicação *reader*→*tag* designa-se por *downlink* e *tag*→*reader* por *uplink* [Dob07].

As *tags* são classificadas como passivas, semi-passivas/semi-ativas ou ativas. Esta classificação é atribuída de acordo com o facto destas, possuíram ou não, uma fonte de energia integrada [Ant12].

As *tags* RFID passivas não possuem transmissor de rádio nem fonte de energia independente. Estas dependem da energia enviada pelo *reader* através de ondas de rádio para alimentar os seus circuitos internos e limitam-se a emitir esta mesma energia em forma de reflexão (*backscattered communication*) [Vio05a].

Por outro lado, as *tags* RFID semi-passivas usam uma bateria local para alimentar os seus circuitos internos, contudo continuam a usar uma comunicação *backscattered* para o *reader* [Dob07].

Por fim, as *tags* RFID ativas possuem um transmissor de rádio e uma fonte de energia independente, transmitindo assim a informação armazenada no *microchip* sem necessitar de energia emitida por um *reader* [Vio05a].

Os *readers* são tipicamente compostos por uma unidade de processamento, seguido de uma cadeia analógica de receção e transmissão de ondas rádio, também designado por *Front-End* analógico. Este *Front-End* é tipicamente baseado numa arquitetura de um recetor/transmissor I/Q (*In phase/Quadrature*) [AI08].

A figura 1.2 apresenta, esquematicamente, a arquitetura típica de um *reader*.

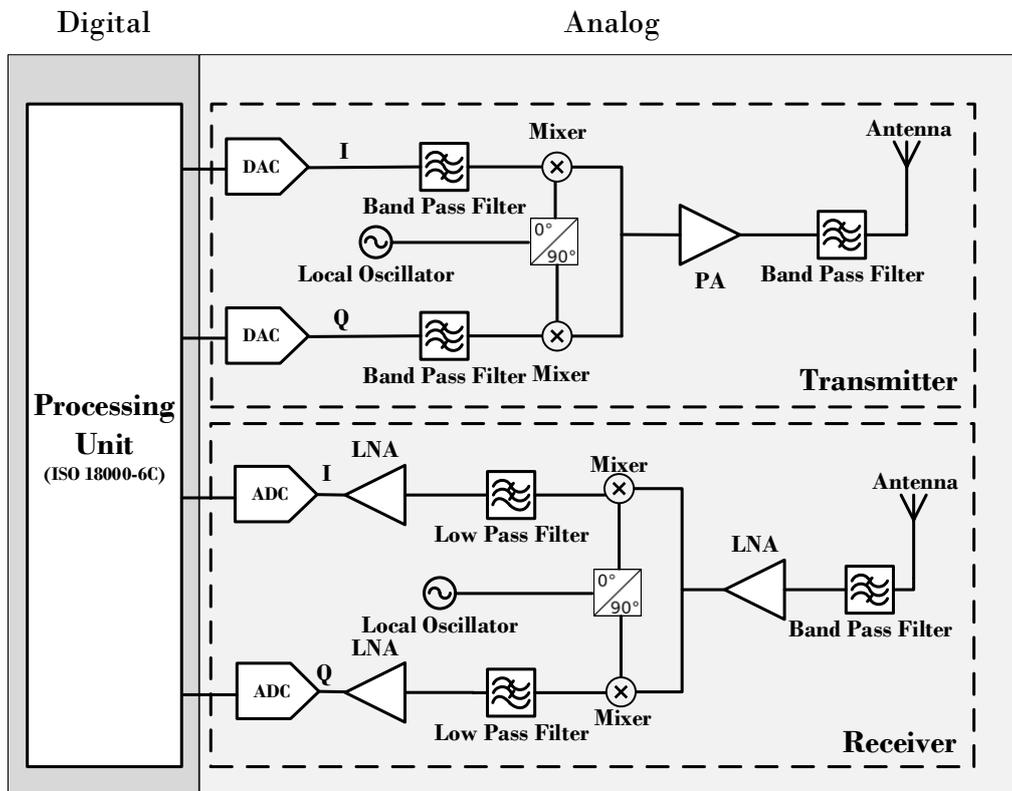


Figura 1.2: Arquitetura típica de um *reader*. *Transmitter* e *Receiver* correspondem à cadeia de transmissão e receção, respetivamente (figura adaptada de [ZDC10]).

Por sua vez, a unidade de processamento pode ser baseada em diversas arquiteturas, onde as mais comuns são baseadas em microcontroladores, em *Digital Signal Processors* (DSPs) ou em *Field Programmable Gate Arrays* (FPGAs).

Na secção seguinte, seguem as motivações que levaram à implementação, do sistema RFID, numa arquitetura baseada em FPGA.

1.2 Motivação

Através das arquiteturas, DSP e FPGA, surge um novo conceito aplicado aos sistemas de rádio - O *Software-Defined Radio* (SDR).

Existem várias definições para *Software-Defined Radio*, contudo o SDR Forum em colaboração com o *Institute of Electrical and Electronic Engineers* (IEEE) estabeleceram um definição simples e clara desta tecnologia:

“*Radio in which some or all of the physical layer functions are software defined*” [For07].

Ou seja, é um sistema onde as funções de processamento de sinal, tais como moduladores, desmoduladores, filtros, entre outros, são implementados em *software* ou *hardware* digital reconfigurável.

Esta tecnologia trás consigo os mais variados benefícios, tais como [Inn11]:

- Para as empresas que produzem equipamentos de rádio, os produtos são implementados numa plataforma/arquitetura comum, o que possibilita uma introdução mais rápida de novos produtos no mercado. Por sua vez, os custos de desenvolvimento são distribuídos por um maior número de produtos. Por último, é possível uma manutenção/reprogramação remota enquanto o produto está em serviço o que torna mais rápida e barata a correção de erros.

- Para as empresas que disponibilizam os serviços, o SDR torna possível que as suas infraestruturas sejam dinâmicas e facilmente atualizáveis, logo não é necessário investir muito capital para estender a infraestrutura sempre que haja novas funcionalidades a ser integradas.

- Para o consumidor final, torna-se um produto mais duradouro e passível de maior número de atualizações, tornando-o ao mesmo tempo, dinâmico e adaptável às suas necessidades.

Por sua vez, e como o nome indica, uma FPGA é composta por um *array* de portas lógicas. A programação deste dispositivo consiste em conectar estas portas nas mais diversas combinações, de maneira a obter os blocos de processamento, tais como μ -processadores, multiplicadores, filtros, entre outros.

A figura 1.3 apresenta um esquema simplificado da organização interna de uma FPGA.

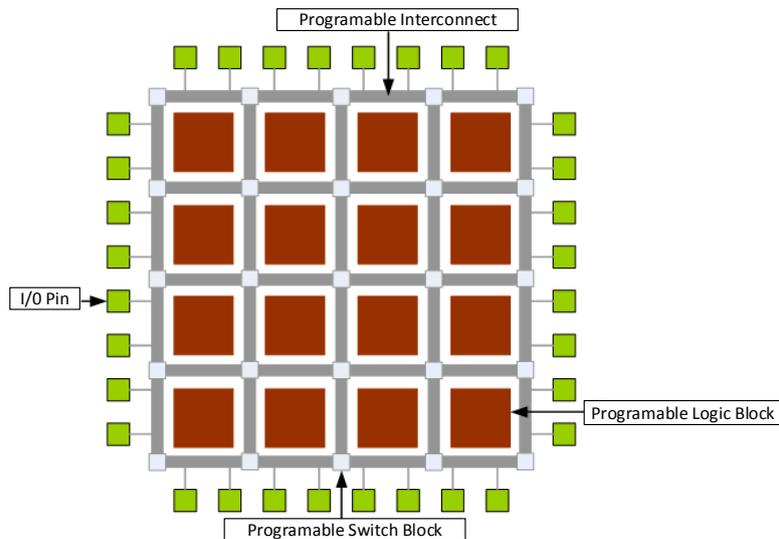


Figura 1.3: Arquitetura de uma FPGA (figura adaptada de [Kul14]).

O RFID, na gama UHF, consiste numa tecnologia dinâmica, ou seja, mantendo o mesmo protocolo de comunicação, a tecnologia é sujeita a normas e restrições implementadas pelo regulador das telecomunicações presente em cada país. Estas restrições diferem na frequência de operação, na largura de banda do sinal transmitido, nas bandas de guarda entre canais e potência máxima transmitida por canal [GS113b].

A figura 1.4 apresenta a distribuição mundial das frequências de operação UHF para o RFID.

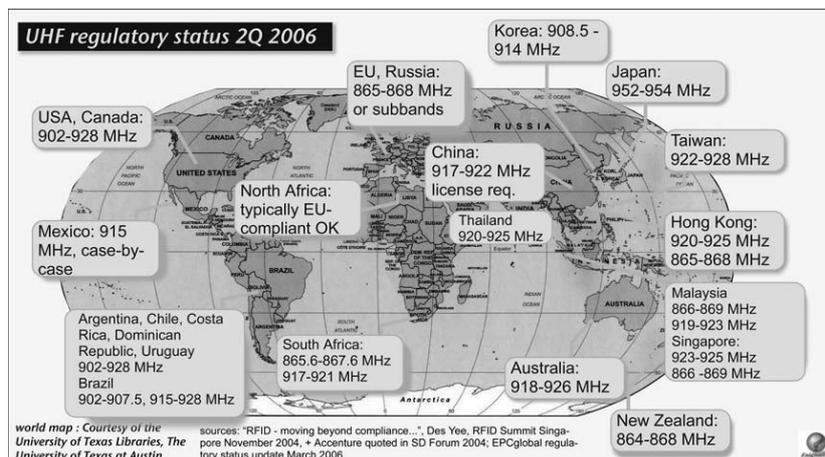


Figura 1.4: Distribuição mundial das frequências de operação UHF para o RFID (figura retirada de [Dob07]).

Estas diferenças provêm das técnicas de comunicação implementadas pelas duas maiores reguladoras de *standards* de telecomunicações: a *Federal Communications Commission* (FCC) dos Estados Unidos da América e a *European Telecommunications Standards Institute* (ETSI) da União Europeia [Imp14]. De seguida, são apresentadas as diferentes técnicas im-

plementadas por estas duas entidades [GS113a] [GS113b].

A entidade FCC estabelece a técnica FHSS – *Frequency Hopping Spread Spectrum* (ver figura 1.5):

- Frequência de operação do *reader* entre 902-928 MHz;
- Largura de banda dos canais de transmissão é de 500 kHz;
- Canais de transmissão contíguos;
- Potência máxima transmitida é de 4W ERP (*Effective Radiated Power*).

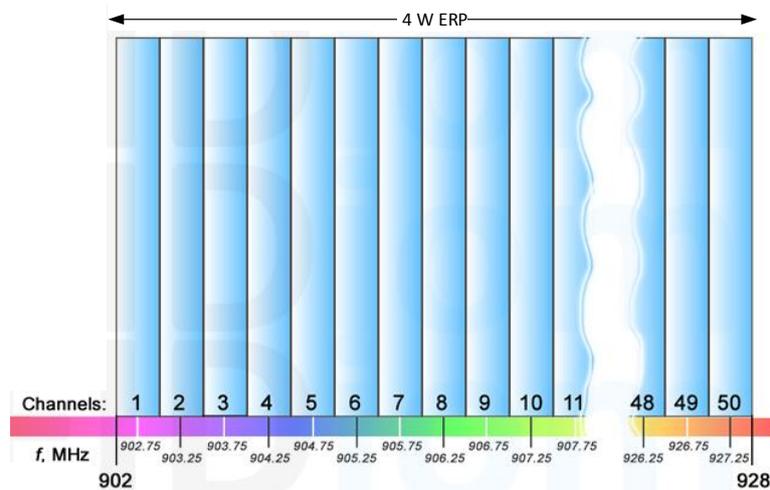


Figura 1.5: Regulamentação imposta à tecnologia RFID pela FCC (figura retirada de [RFI10]).

A entidade ETSI estabelece a técnica *4 Channel Plan* (ver figura 1.6):

- Frequência de operação do *reader* entre 865-868 MHz;
- Largura de banda dos canais de transmissão é de 200 kHz;
- Distância entre canais de transmissão é de 600 kHz;
- Potência máxima transmitida é de 2W ERP.

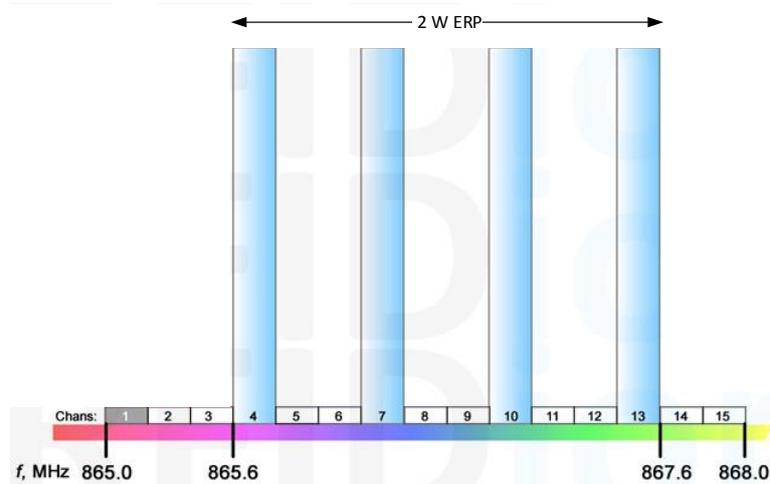


Figura 1.6: Regulamentação imposta à tecnologia RFID pela ETSI (figura retirada de [RFI10]).

Por último, com o dinamismo entre larguras de banda e frequências de operação presentes no RFID, torna-se vantajoso aplicar a versatilidade do SDR à arquitetura de um *reader*. Por sua vez, devido à facilidade da implementação de blocos de processamento, a FPGA torna-se a candidata ideal para desenvolver o leitor RFID. Por exemplo, num único *chip* é possível implementar um processador capaz de processar toda a camada protocolar, assim como os filtros e *mixers* utilizados nas cadeias de transmissão e recepção dos sinal RFID.

1.3 Objetivos

Esta dissertação de mestrado tem como objetivo principal a implementação em FPGA, baseada em abordagens SDR, de um leitor RFID na gama *Ultra High Frequency* (UHF).

É usado o ISO (*International Standards Organization*) 18000-6C como protocolo de comunicação RFID que visa operar na gama 860-960 MHz, banda livre ISM (*Industrial, Scientific and Medical*). Em Portugal, a Autoridade Nacional de Comunicações (ANACOM) aplica a técnica imposta pela ETSI, ou seja, estabelece que a gama UHF para RFID é 865.6-867.6 MHz, a largura de banda dos canais é de 200 kHz, o espaçamento entre canais adjacentes é de 600 kHz e, por fim, uma potência máxima radiada de 2W ERP, ou seja, aproximadamente 33.01 dBm [GS113b].

Para terminar, é igualmente importante referir que é usada a codificação PIE (*Pulse Interval Encoding*) juntamente com a modulação em amplitude DSB-ASK (*Double-Sideband Amplitude Shift Keying*) no sentido *downlink* e codificação FM0 (*bi-phase space*) no sentido *uplink*.

1.4 Organização do Documento

Além deste capítulo introdutório, este documento encontra-se dividido em cinco capítulos e cinco anexos.

Capítulo 2 - RFID - Este capítulo contém uma explicação mais detalhada sobre a tecnologia RFID, as classificações dos vários sistemas RFID e o protocolo de comunicação ISO 18000-6C.

Capítulo 3 - Arquitetura de um Sistema RFID em SDR - É apresentado neste capítulo a explicação do conceito *Software-Defined Radio* e as arquiteturas SDR mais comuns de transmissão e recepção. Este capítulo também aborda como a tecnologia SDR é aplicada a RFID;

Capítulo 4 - Implementação do Sistema RFID - Capítulo onde é apresentada a implementação do sistema proposto.

Capítulo 5 - Resultados - Resultados obtidos que comprovam o correto funcionamento do sistema implementado.

Capítulo 6 - Conclusão e Trabalho Futuro - Capítulo que encerra a dissertação de mestrado. Contém um resumo do documento e limitações do sistema, assim como, um possível trabalho futuro a ser implementado.

Anexo A - Anexo onde é apresentado o diagrama de blocos do sistema final implementado.

Anexo B - Anexo onde é apresentado o sistema final implementado.

Anexo C - Anexo onde é apresentado o Teorema de Nyquist.

Anexo D - Anexo onde é apresentado a modulação ASK.

Anexo E - Anexo onde é apresentado o conceito de filtro FIR.

Capítulo 2

RFID

Este capítulo tem como objetivo apresentar de forma resumida a tecnologia RFID. Começa com uma breve referência histórica sobre a sua origem, terminando no protocolo de comunicação ISO 18000-6C. Ao longo deste capítulo são referenciadas as várias classificações dos sistemas RFID, assim como os vários protocolos de comunicação suportados por esta tecnologia.

2.1 Breve Perspetiva Histórica

A identificação por rádio frequência está diretamente ligada às descobertas feitas por Michael Faraday (conceito de indução mútua) e Guglielmo Marconi (comunicação por ondas eletromagnéticas), em meados do século XIX, uma vez que estas descobertas constituem os pilares fundamentais da tecnologia RFID. Contudo só em meados do século XX é que esta tecnologia começou a emergir. A Figura 2.1 apresenta a evolução do RFID.

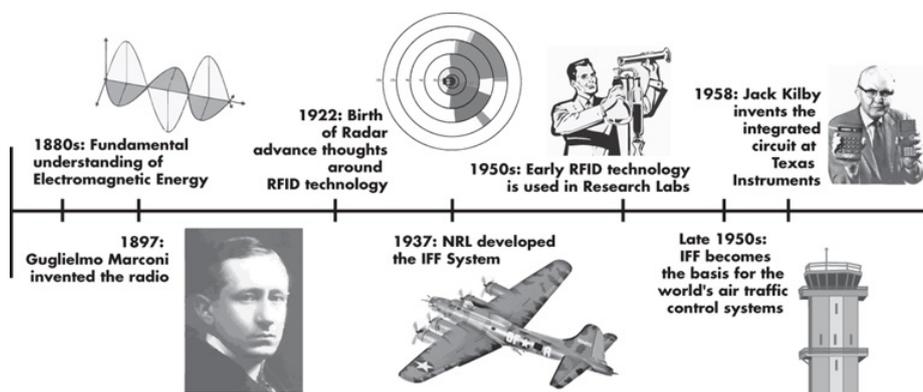


Figura 2.1: Evolução do RFID (figura retirada de [BM05]).

Em 1935 o físico Escocês Sir Robert Alexander Watson-Watt desenvolveu o RADAR (*RA*dio *D*etection *A*nd *R*anging), tecnologia emergente na II Guerra Mundial. Contudo, esta tecnologia apresentava um grande problema: não era possível identificar se as aeronaves que se aproximavam eram inimigas ou aliadas. Este problema foi resolvido pelos Alemães da seguinte maneira: se os seus pilotos “rodassem” os aviões (pilotos ficavam com o *cockpit*

virada para baixo) ao chegar à base, estes iriam refletir um sinal de rádio diferente. Podemos considerar este método como o primeiro sistema RFID passivo. Os Britânicos, liderados por Watson-Watt, desenvolveram um sistema ativo designado de *Identify Friend or Foe* (IFF). Este sistema consistia em colocar um transmissor em cada avião. Quando este recebia um sinal da torre de RADAR emitia um sinal de volta, que o identificava como um avião aliado. O RFID opera exatamente segundo o mesmo princípio [Dob07][AI08][Vio05b].

2.2 Classificação dos Sistemas RFID

Os sistemas RFID são classificados pela frequência das ondas rádio transmitidas e pelo método de energização das *tags*: sistemas ativos, passivos ou semi-passivos [Dob07].

As bandas de frequência mais comuns neste tipo de sistemas são 125/134 kHz, 13.56 MHz, 860-960 MHz e 2.4-2.45 GHz.

Conforme a gama de operação, os sistemas RFID são classificados do seguinte modo [Dob07]:

- Os sistemas que operam na gama de frequências [125/134 kHz] são designados por sistemas LF (*Low Frequency*), visto trabalharem na zona das baixas frequências do espectro eletromagnético;
- Os sistemas que trabalham na zona dos [13.56 MHz] são designados de sistemas HF (*High Frequency*);
- Os sistemas, cuja gama de frequências, correspondem a [860-960 MHz / 2.4-2.45 GHz] são designados por sistemas UHF.

Conforme as gamas de operação apresentadas em cima, os sistemas RFID são igualmente classificados como indutivos ou radiativos. Um sistema indutivo utiliza espiras como antenas. A tensão induzida nas espiras é proporcional ao número e tamanho destas, assim como à frequência de operação. Por exemplo, um sistema tipicamente usado na identificação de animais ou seres humanos, tem um número de espiras na ordem das centenas. Por outro lado, um sistema HF, também indutivo, possui tipicamente uma dezena de espiras. Esta variação (número de espiras) deve-se fundamentalmente ao facto dos sistemas HF operarem numa frequência duas ordens de grandeza acima da frequência dos sistemas LF. O sistema RFID HF é a solução usada em cartões de acesso, alarmes, cartões de multibanco, entre outros. Por fim, os sistemas UHF são radiativos. Nestes sistemas as antenas das *tags* são variantes de um dipolo genérico, ou seja, consistem num fio dividido ao meio com tamanho de metade do comprimento de onda. Estes são sistemas RFID usados para comunicações de longo alcance, tais como a identificação de objetos num armazém ou a identificação de um utilizador do serviço Via-Verde num pórtico de autoestrada [Dob07].

A figura 2.2 ilustra a classificação dos Sistemas RFID.

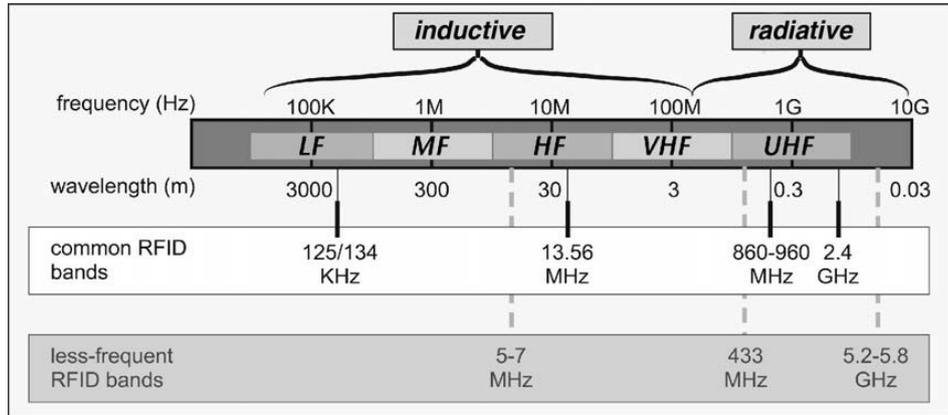


Figura 2.2: Bandas de operação do RFID (figura retirada de [Dob07]).

2.3 Protocolos de Comunicação para Sistemas RFID

As *tags* e os leitores de RFID, designados por *readers* daqui em diante, necessitam de comunicar entre si e, como qualquer outra forma de comunicação, é necessário um protocolo para esta (comunicação) se realizar.

A tabela 2.1 apresenta os vários protocolos de comunicação para os sistemas de RFID descritos no subcapítulo 2.2. De notar que o protocolo de comunicação ISO 18000-5, para 5.8 GHz, presente na tabela 2.1, foi descartado pela *International Standards Organization*, contudo, continua a ser utilizado em Portugal nos sistemas de Via-Verde.

Frequency Tag type	125 kHz	5-7 MHz	13.56 MHz	303/433 MHz	860-960 MHz	2.45 GHz	5.8 GHz
Passive	ISO 11784/5 14223 ISO 18000-2	ISO 10536 iPico DF/iPx	MIFARE (ISO 14443) Tag-IT (ISO 15693) ISO 18000-3		ISO 18000-6 EPC class 0 EPC class 1 EPC GEN II Intellitag tolls (Title 21) Rail (AAR S918)	ISO 18000-4 Intellitag μ-chip ISO 10374	
Semi-passive					Title 21 Rail (AAR S918)	ISO 18000-4 WereNet (ANSI 371.1)	ISO 18000-5 (Withdrawn)
Active				Savi (ANSI 371.2) ISO 18000-7 RFCode		ISO 18000-4 WhereNet (Ansi 371.1)	

Tabela 2.1: *Standards* de comunicação RFID (tabela adaptada de [BM05]).

De seguida é apresentada uma breve abordagem destes protocolos [BM05]:

- Na banda 125 kHz está presente **ISO 11784, 11785, e 14223**. Estes *standards* contêm a estrutura do código RFID para identificação de animais. O ISO 14223 estabelece a comunicação entre o *reader* e a *tag* baseado nas condições de compatibilidade nos ISO 11784-5.

- Nas bandas 5-7 MHz e 13.56 MHz: **ISO 10536, 14443, e 15693**. Estes *standards* são os responsáveis pela comunicação de cartões de proximidade, tais como os cartões de acesso a edifícios ou cartões de pagamento. O ISO 14443 é um dos protocolos RFID mais conhecidos, igualmente denominado por MIFARE (*MIkron FARE collection system*).

- Na banda 2.45 GHz **ISO 10374**. Este *standard* é o responsável pela identificação automática, codificação e segurança de informação de contentores de carga em comboios ou navios.

- **ISO 18000**. Este *standard* e os seus derivados focam-se em RFID para gestão de itens. Protocolos globalmente aceites para as frequências 125 kHz, 13.56 MHz, banda UHF, 2.45 GHz e 5.8 GHz.

Este trabalho de dissertação de mestrado é focado no protocolo **ISO 18000-6C**, também designado por *EPC Generation-2 Class 1 UHF (EPC Gen-2)* [O’C06].

Este *standard*, desenvolvido pela organização EPCglobal, foi aprovado em 2005 pela *International Standards Organization* como uma reformulação do protocolo de comunicação 18000-6, criando assim a emenda ISO 18000-6C [O’C06].

Seguidamente é apresentado o protocolo de comunicação EPC Generation-2 Class 1 UHF RFID.

2.4 EPC Generation-2 Class 1 UHF RFID

A presente secção apresenta o protocolo de comunicação implementado nesta dissertação de mestrado. Começa por apresentar o conceito de EPC (*Electronic Product Code*), seguido de uma breve descrição e funcionamento deste *standard*, e termina, com maior detalhe, nas especificações protocolares. Para tal, este subcapítulo, foi baseado no documento [GS113a].

2.4.1 *Electronic Product Code* - EPC

A EPCglobal é uma organização sem fins lucrativos, que trabalha na comercialização de um identificador único para produtos eletrónicos. Este identificador é designado por EPC (*Electronic Product Code*). Estes códigos são utilizados para rastrear produtos numa cadeia de distribuição [O’C06].

A figura 2.3 apresenta os vários campos que compõem um EPC.

01	0000A89	00016F	000169DC
Header 8-bit	EPC Manager 28-bit	Object Class 24-bit	Serial Number 36-bit

Figura 2.3: Exemplo de um EPC.

Cada campo apresentado na figura 2.3 tem uma função única [ER13]:

- **Header:** Identifica o tamanho, tipo, estrutura, versão e geração do EPC.
- **EPC Manager Number:** Entidade ou empresa responsável que usa a tecnologia.
- **Object Class:** Identifica a classe do produto que contém a *tag*.
- **Serial Number:** Identificador único de cada produto.

2.4.2 Descrição e Funcionamento

O EPC Gen-2 utiliza um protocolo do tipo *Aloha*, mais conhecido por *Q-protocol*. Resumidamente, um protocolo do tipo *Aloha*, consiste na premissa de que o *reader* “fala primeiro”, ou seja, ele começa a comunicação transmitindo uma trama, onde é definido o número de *slots* disponíveis. Caso a *tag* receba uma trama válida, esta irá enviar a sua informação num destes *slots* [ZKP09]. A figura 2.4 apresenta o funcionamento descrito anteriormente.

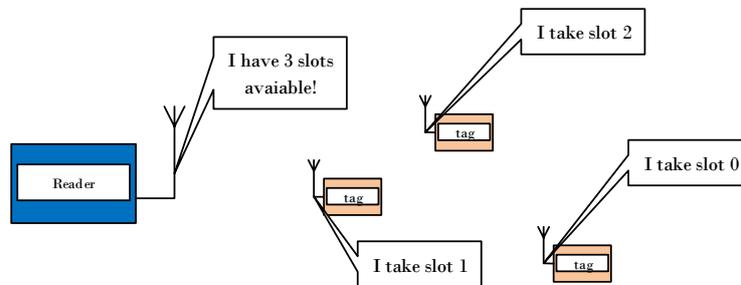


Figura 2.4: Esquema simplificado do funcionamento do protocolo EPC Gen-2.

Este protocolo é dividido em três partes designadas por *Inventory Round*, *Select Round* e *Access Round*. Cada uma destas partes são utilizadas, em conjunto ou individualmente, conforme o objetivo pretendido. Por exemplo:

- Quando o objetivo é a identificação individual das *tags*, ou seja, obtenção do EPC, o *reader* só necessita de implementar a *Inventory Round*, uma vez que, esta é a parte do protocolo responsável pela obtenção do EPC. Esta situação é apresentada na figura 2.5.

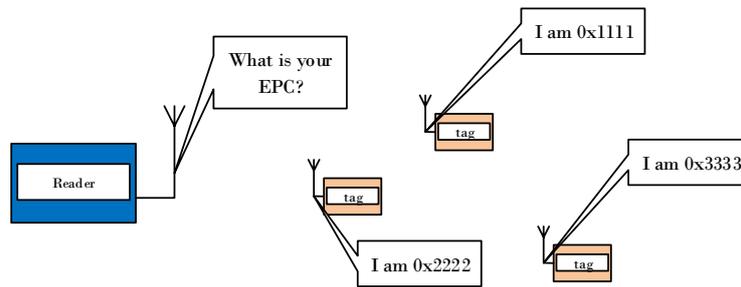


Figura 2.5: Esquema simplificado do funcionamento do protocolo EPC Gen-2 na obtenção do EPC de cada *tag*.

- Num ambiente onde estão presentes múltiplas *tags* e múltiplos *readers*, é necessário implementar uma solução que evite leituras redundantes, ou seja, dois *readers* não obtêm o EPC da mesma *tag*. Neste caso, os *readers* implementam a *Select Round* seguida da *Inventory Round*. A *Select Round* é a parte do protocolo que seleciona a população de *tags* que responde a um *reader* específico. A figura 2.6 apresenta esta situação.

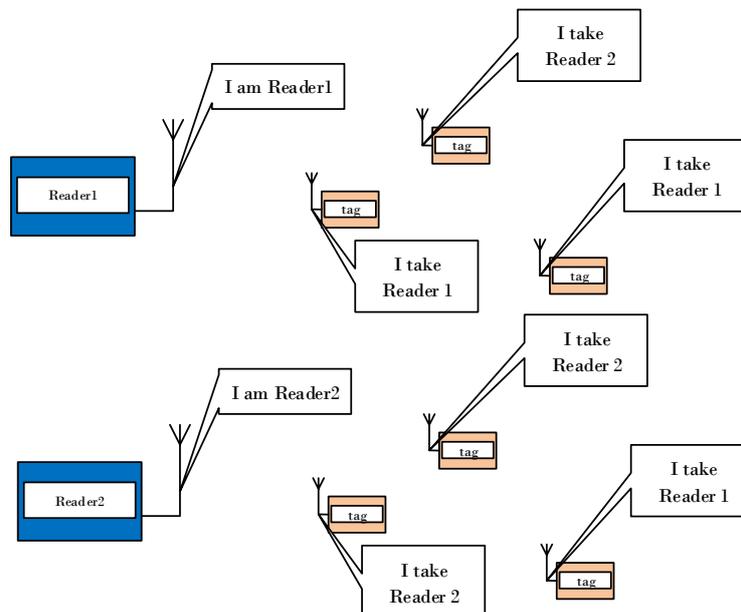


Figura 2.6: Esquema simplificado do funcionamento do protocolo EPC Gen-2 na situação onde estão presentes, no mesmo ambiente, múltiplas *tags* e múltiplos *readers*.

- O protocolo também implementa a opção de modificar o EPC de uma *tag* específica. Para esse efeito, o *reader* necessita de implementar a *Inventory Round* seguida da *Access Round*. A *Access Round* é a parte do protocolo responsável por escrever nos bancos de memória da *tag*. A figura 2.7 apresenta esta situação.

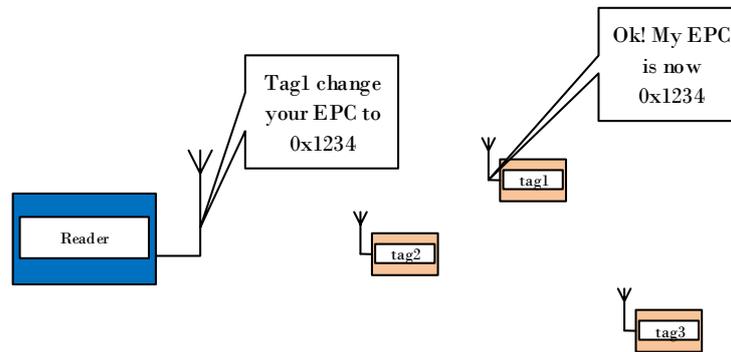


Figura 2.7: Esquema simplificado do funcionamento do protocolo EPC Gen-2 na situação onde o *reader* modifica o EPC de uma *tag* específica.

Esta dissertação de mestrado foca-se na identificação de uma *tag*, ou seja, na parte *Inventory Round* do protocolo, o qual é resumido de seguida:

O *reader* começa por enviar um comando designado por *Query*. Este comando contém os parâmetros onde se define quais as *tags* autorizadas a participar numa sessão. Para além destes, o comando *Query* contém outro parâmetro designado por 'Q', que especifica o número de *slots* (2^Q) disponíveis por parte do *reader* para inventariar as *tags* distribuídas aleatoriamente no espaço. Passo 1) da figura 2.9.

Uma *tag* participante vai gerar um número aleatório entre 0 e $(2^Q - 1)$. Este número aleatório representa a ordem pela qual as *tags* enviam informação para o *reader*, ou seja, o *slot* no qual a *tag* vai responder. O número de *slots* disponível necessita ser da mesma ordem de grandeza do número de *tags* presente na zona de leitura.

É de referir que 'Q' toma valores de 0 a 15. Sendo assim, numa *Inventory Round*, é possível ler desde uma *tag* até 32768 (2^{15}) *tags*.

Se o número aleatório gerado pela *tag* for zero, então esta vai responder no primeiro *slot*. Por outro lado, se o número for diferente de zero, a *tag* vai registar o número no seu *slot counter* e vai aguardar pela sua vez (ver figura 2.8). A *tag* que gerou o número zero vai agora gerar um número aleatório de 16-bit denominado por RN16. Este RN16 entende-se como o ID da transmissão, ou seja, após um *Query* enviado pelo *reader*, os comandos seguintes necessitam de incluir o mesmo RN16 anteriormente gerado pela *tag*. Passo 2) da figura 2.9.

Se o *reader* receber, em tempo útil, o RN16 depois de enviar o *Query*, este (o *reader*) vai sinalizar a *tag* com um comando de *acknowledgement*, *ACK*, que contém o mesmo RN16. Passo 3) da figura 2.9.

Se o RN16 enviado pelo *reader* corresponder ao RN16 enviado pela *tag*, então ela responde com o seu PC+EPC+CRC16, ou seja, com o seu *Protocol Control*, *Electronic Product Code* e *Cyclic Redundancy Check* 16-bit, respetivamente.

Caso o RN16 não corresponder, o *acknowledgement* simplesmente falha e a *Inventory Round* continua. Passo 4) da figura 2.9.

Após receber o EPC, o *reader* pode optar por ter acesso a esta *tag* específica, isto é, ler ou escrever em várias partes do banco de memória desta (*Access Round*). Passo 5), 6), 7) e 8) da figura 2.9.

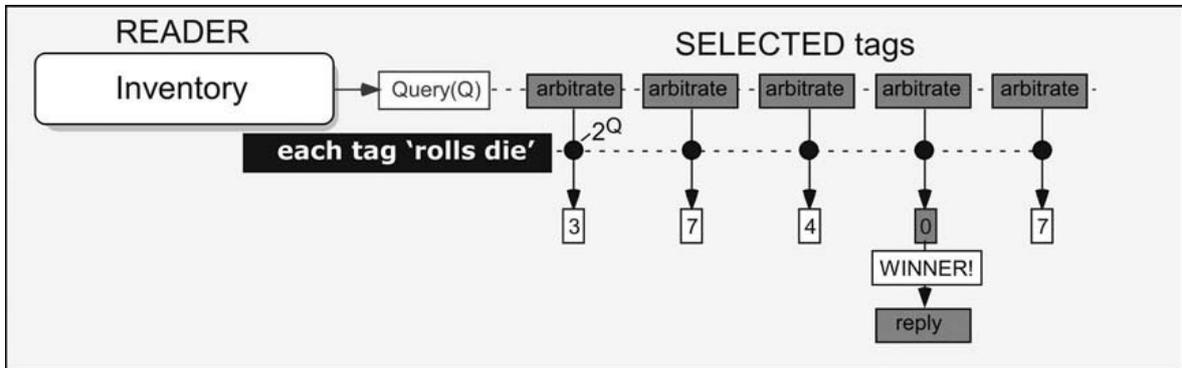


Figura 2.8: Início de uma *Inventory Round*. $Q=3$ (imagem retirada de [Dob07]).

Se o *reader* só necessitar do EPC, então pode enviar um novo comando *Query*, mais especificamente um *QueryRep*. Um comando *QueryRep* é usado para sinalizar o fim de um *slot*. As restantes *tags* vão decrementar uma unidade no seu *slot counter* e o processo começa todo de novo.

No caso onde o EPC recebido não é válido, o *reader* envia para a *tag* o comando *NAK*, *not acknowledgent*. Este comando sinaliza a *tag* para se manter no seu estado atual, até que ocorra uma leitura válida do seu EPC.

A figura 2.9 apresenta esquematicamente o funcionamento descrito em cima.

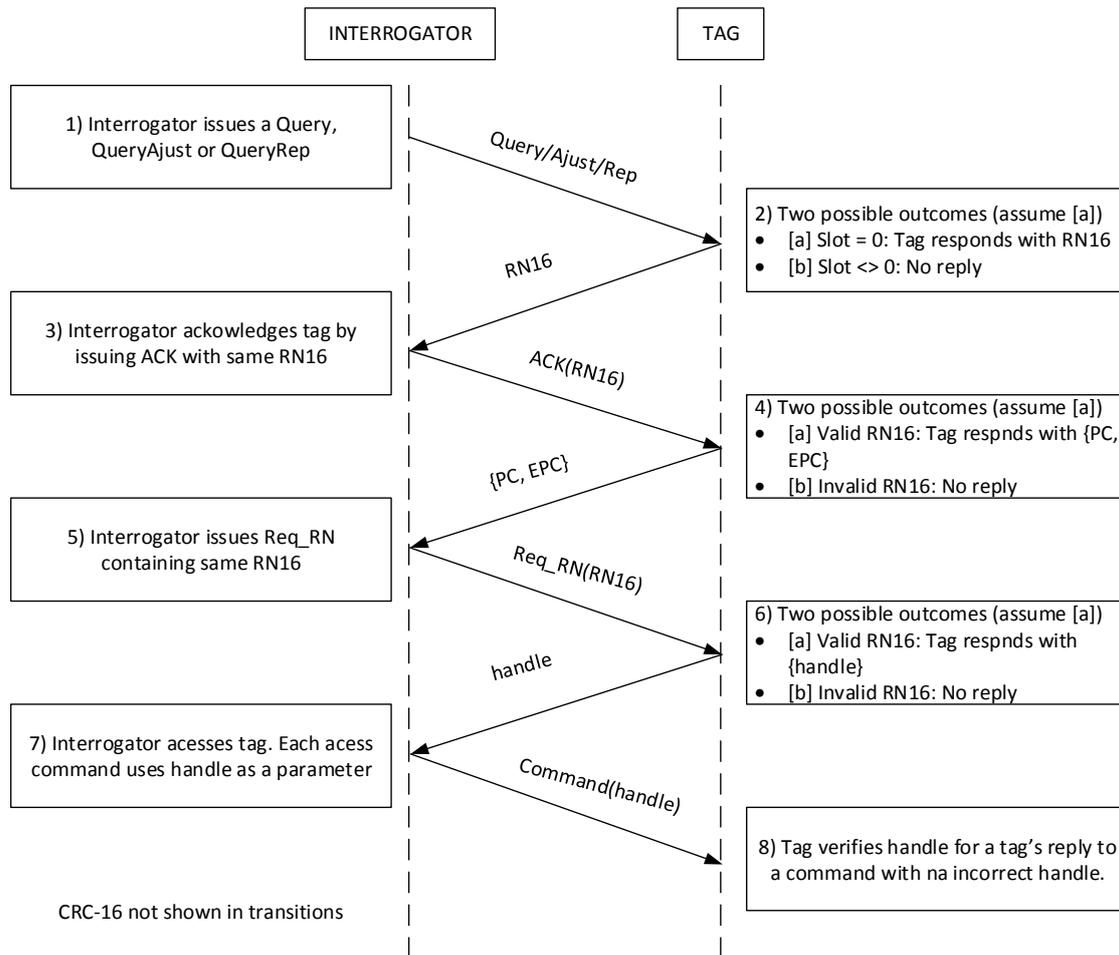


Figura 2.9: Esquema de funcionamento do protocolo ISO 18000-6C.

Apresentado um *overview* do protocolo ISO 18000-6C, é agora abordado com maior detalhe as especificações do *reader*.

2.4.3 Comunicação Reader para Tag (R→T)

Este subcapítulo foca-se na comunicação no sentido *downlink*, ou seja, na comunicação no sentido do *reader* para a *tag*.

Modulação do Sinal Transmitido

Os *readers* podem utilizar três tipos de modulação:

- DSB-ASK (*double-sideband amplitude shift keying*).
- SSB-ASK (*single-sideband amplitude shift keying*).
- PR-ASK (*phase-reversal amplitude shift keying*).

A modulação implementada é abordada mais tarde no capítulo 4.

Codificação de Informação

A informação proveniente do *reader* é composta por dois símbolos de informação: *data-0* e *data-1*. Estes símbolos são obtidos com recurso à codificação PIE.

- O símbolo *data-0* é composto por dois níveis lógicos, nível lógico '1' e '0', de igual duração. A duração do nível lógico '0' é designado por PW (*Pulsewidth*) e a duração total do símbolo designa-se por Tari. O PW toma duração mínima de $\text{MAX}(0.265 \times \text{Tari}, 2)\mu\text{s}$ e duração máxima de $0.525 \times \text{Tari} \mu\text{s}$ (ver figura 2.10).

- O símbolo *data-1* é similar ao *data-0*, contudo, o nível lógico '1' possui uma maior duração. Assim sendo, o símbolo *data-1* é de maior duração e toma valores entre $1.5 \times \text{Tari} \mu\text{s}$ e $2 \times \text{Tari} \mu\text{s}$ (ver figura 2.10).

Importante referir que os valores de Tari podem variar entre $6.25 \mu\text{s}$ e $25 \mu\text{s}$.

A figura 2.10 expõe a codificação do símbolo '0' e do símbolo '1'.

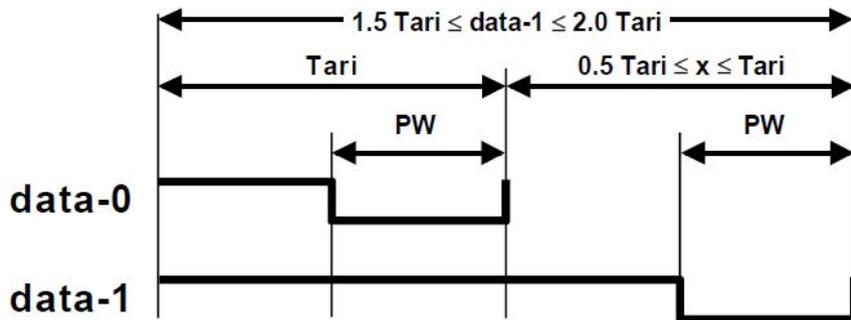


Figura 2.10: Símbolos PIE (imagem retirada de [GS113a]).

Preamble e Frame-sync

A comunicação R→T é precedida por um *Preamble* ou um *Frame-sync* consoante o comando a enviar para a *tag*. Por exemplo, um comando *Query* é precedido por um *Preamble* e, por sua vez, um comando *ACK* é precedido por um *Frame-sync*.

O *Preamble* é formado da seguinte forma (ver figura 2.11): um delimitador, um símbolo '0', um símbolo de calibração R→T (RTcal) e um símbolo de calibração T→R (TRcal). A duração do símbolo RTcal é igual à soma da duração *data-0* com a duração *data-1* (ver figura 2.11 R=>T *Preamble*).

O símbolo TRcal é usado pela *tag* para definir o seu *Backscatter Link Frequency* (BLF), (ver equação 2.1). Este símbolo (TRcal) toma valores mínimos de $1.1 \times \text{RTcal}$ e máximos de $3 \times \text{RTcal}$.

$$\text{BLF} = \frac{\text{DR}}{\text{TRcal}} \quad (2.1)$$

O BLF corresponde à frequência de símbolo enviado pela *tag*. No caso particular onde a informação proveniente da *tag* é codificada em FM0, o BLF representa o *data-rate* deste mesmo sinal. Na equação 2.1, DR representa um *Divide Ratio*, e toma um de dois valores: $\frac{64}{3}$ ou 8.

O *Frame-sync* é similar ao *Preamble*, contudo, o símbolo TRcal não se encontra presente. Ver figura 2.11 R=>T *Frame-sync*.

A figura 2.11 ilustra as tramas *Preamble* e *Frame-sync* mais detalhadamente.

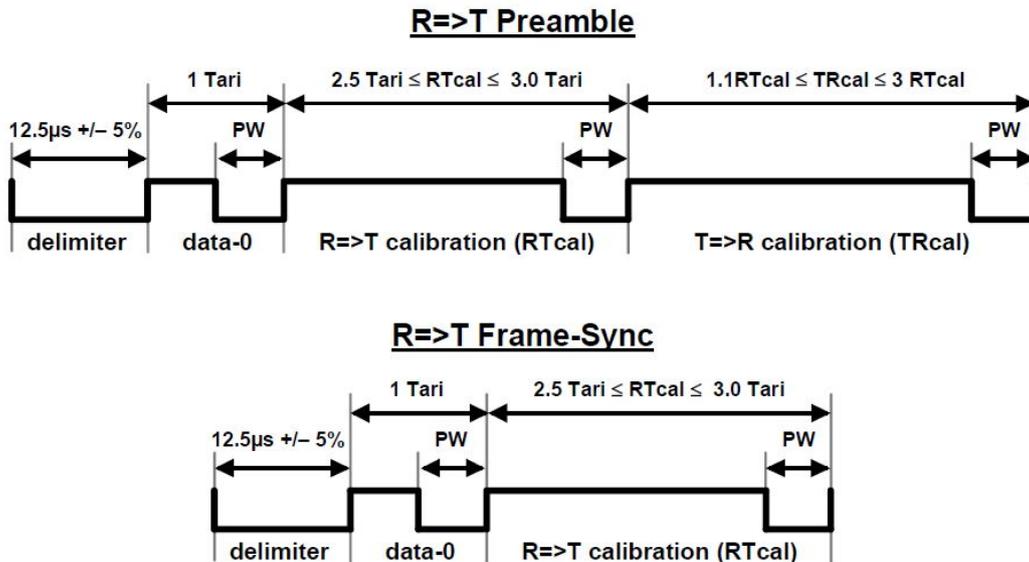


Figura 2.11: *Preamble* e *Frame-sync* (imagem retirada de [GS113a]).

Máscaras de Potência Transmitida

Existem três máscaras de potência transmitida aplicáveis. E estas variam consoante o número de *readers* presentes no meio. Num ambiente com um único *reader*, *single-interrogator environment*, só é necessário respeitar as normas estipuladas pelas entidades reguladoras. Como dito anteriormente, em Portugal, a ANACOM estabelece que para a gama RFID UHF é atribuída uma potência máxima radiada de 33.01 dBm e 200 kHz de largura de banda. Em ambientes com múltiplos *readers* (*multiple-interrogator environment*) o rácio entre a potência de um canal R e de um outro canal S diferente de R tem que respeitar os seguintes valores:

$$|R - S| = 1 : 10 \log(P(S)/P(R)) < -20 \text{ dB}$$

$$|R - S| = 2 : 10 \log(P(S)/P(R)) < -50 \text{ dB}$$

$$|R - S| = 3 : 10 \log(P(S)/P(R)) < -60 \text{ dB}$$

P() representa a potência transmitida num determinado canal.

A figura 2.12 apresenta graficamente a máscara de transmissão para o ambiente descrito anteriormente.

Por fim, num *dense-interrogator environment*, ou seja, um ambiente onde todos os canais/frequências disponíveis para transmissão estão ocupados por outros *readers*, um *reader* com frequência central f_c , largura de banda (Rbw) e frequência de *offset* (f_o) $2.5/Tari$, o rácio entre um canal S com largura de banda (Sbw) $2.5/Tari$ centrado em $(n \times f_o) + f_c$, ($n \in \mathbb{Z}$), tem que respeitar os valores:

$$|n|=1 : 10 \log(P(Sbw)/P(Rbw)) < -30 \text{ dB}$$

$$|n|=2 : 10 \log(P(Sbw)/P(Rbw)) < -60 \text{ dB}$$

$$|n|>2 : 10 \log(P(Sbw)/P(Rbw)) < -65 \text{ dB}$$

$P()$ representa a potência transmitida num determinado canal.

A figura 2.13 apresenta graficamente esta máscara de transmissão.

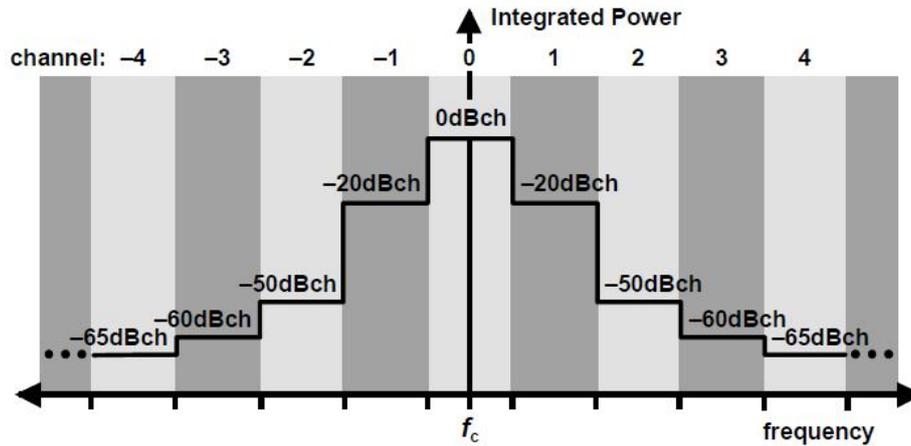


Figura 2.12: Máscara de transmissão em ambientes *multiple-interrogator* (imagem retirada de [GS113a]).

2.4.4 Comunicação Tag para Reader (T→R)

Este subcapítulo foca-se na comunicação no sentido *uplink*, ou seja, na comunicação no sentido da *tag* para o *reader*.

Codificação de Informação

A *tag* pode codificar a informação em quatro formatos diferentes: FM0 ou três modulações de Miller. O formato usado é indicado pelo *reader* no comando *Query*.

Nesta dissertação de mestrado é abordada a codificação no formato FM0, uma vez que foi a implementada no *reader* projetado.

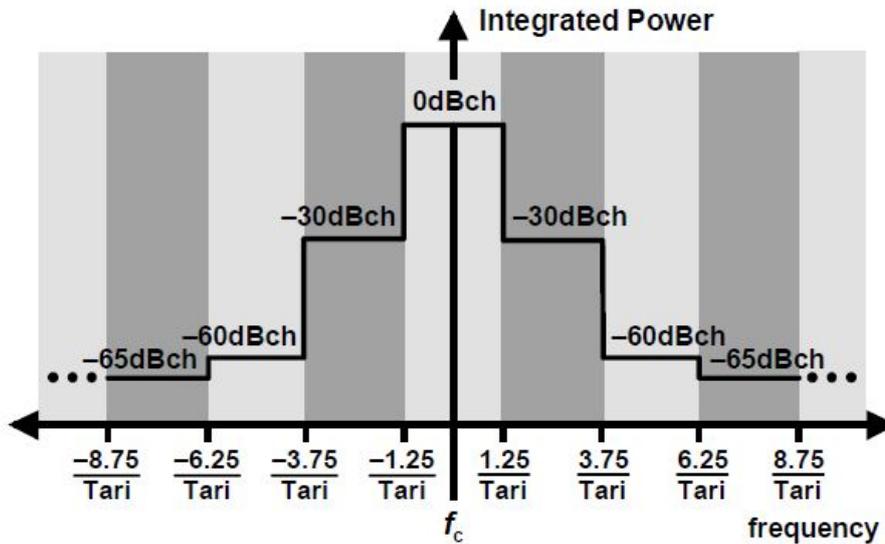


Figura 2.13: Máscara de transmissão em ambientes *dense-interrogator* (imagem retirada de [GS113a]).

Codificação FM0

FM0 é uma codificação onde a fase do sinal é invertida no final de cada símbolo. Por sua vez, também tem a particularidade do símbolo *data-0* sofrer uma inversão de fase adicional a meio da sua duração. A figura 2.14 apresenta os quatro estados possíveis da codificação FM0.

Existem quatro estados possíveis, S1, S2, S3 e S4. Os estados S2 e S3 representam o símbolo *data-0* e são o inverso um do outro, ou seja, $S3(t) = -S2(t)$. Por sua vez, os estados S1 e S4 representam o símbolo *data-1* e são igualmente inversos, $S4(t) = -S1(t)$.

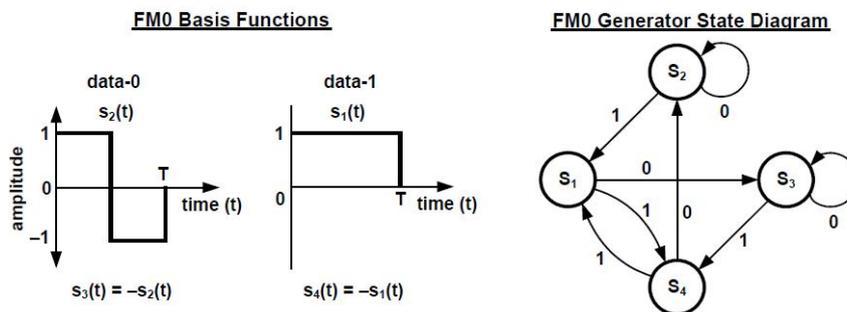


Figura 2.14: Funções base e diagrama de estados da codificação FM0 (imagem retirada de [GS113a]).

Uma trama de informação codificada em FM0 termina sempre com um símbolo *data-1*, ou símbolo *dummy*.

Preamble FM0

Uma trama de informação começa com o envio de um *Preamble* (não confundi com o *Preamble* enviado pelo *reader*) e este pode ou não conter um tom piloto constituído por 12 símbolos *data-0* consecutivos. O envio do tom piloto por parte da *tag* é definido pelo *reader*.

O *Preamble*, sem tom piloto, consiste numa trama formada por 6 símbolos, “1 0 1 0 v 1”, onde um deles é uma violação (v) da codificação FM0. Esta violação resume-se numa não inversão da fase do sinal no final de um símbolo transmitido (ver figura 2.15).

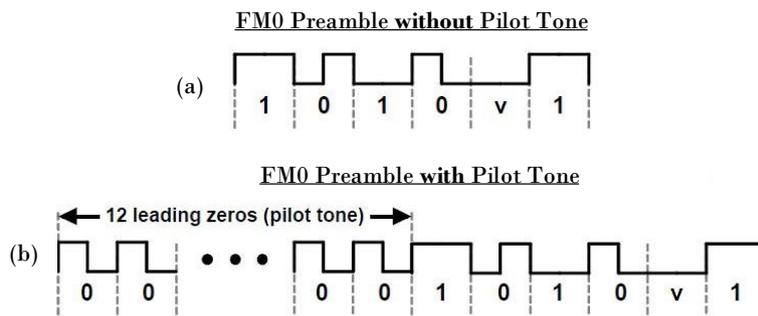


Figura 2.15: (a) Preamble FM0 sem tom piloto. (b) Preamble FM0 com tom piloto (imagem retirada de [GS113a]).

Data-rates Suportados pelas Tags

Como referido anteriormente, o BLF depende das variáveis DR e TRcal. A tabela 2.2 pretende representar a gama total BLF suportada pelas *tags*.

Divide Ratio (DR)	TReal ($\mu\text{s} \pm 1\%$)	BLF (kHz)	Frequency Tolerance FrT	Frequency variation during backscatter
64/3	33.3	640	$\pm 15\%$	$\pm 2.5\%$
	$33.3 < \text{TReal} < 66.7$	$320 < \text{BLF} < 640$	$\pm 22\%$	$\pm 2.5\%$
	66.7	320	$\pm 10\%$	$\pm 2.5\%$
	$66.7 < \text{TReal} < 83.3$	$256 < \text{BLF} < 320$	$\pm 12\%$	$\pm 2.5\%$
	83.3	256	$\pm 10\%$	$\pm 2.5\%$
	$83.3 < \text{TReal} \leq 133.3$	$160 \leq \text{BLF} < 256$	$\pm 10\%$	$\pm 2.5\%$
	$133.3 < \text{TReal} \leq 200$	$107 \leq \text{BLF} < 160$	$\pm 7\%$	$\pm 2.5\%$
	$200 < \text{TReal} \leq 225$	$95 \leq \text{BLF} < 107$	$\pm 5\%$	$\pm 2.5\%$
8	$17.2 \leq \text{TReal} < 25$	$320 < \text{BLF} \leq 465$	$\pm 19\%$	$\pm 2.5\%$
	25	320	$\pm 10\%$	$\pm 2.5\%$
	$25 < \text{TReal} < 31.25$	$256 < \text{BLF} < 320$	$\pm 12\%$	$\pm 2.5\%$
	31.25	256	$\pm 10\%$	$\pm 2.5\%$
	$31.25 < \text{TReal} < 50$	$160 < \text{BLF} < 256$	$\pm 10\%$	$\pm 2.5\%$
	50	160	$\pm 7\%$	$\pm 2.5\%$
	$50 < \text{TReal} \leq 75$	$107 \leq \text{BLF} < 160$	$\pm 7\%$	$\pm 2.5\%$
	$75 < \text{TReal} \leq 200$	$40 \leq \text{BLF} < 107$	$\pm 7\%$	$\pm 2.5\%$

Tabela 2.2: T→R BLF.

Após análise da tabela 2.2, conclui-se que uma *tag* pode enviar informação codificada em FM0 desde um *data-rate* mínimo de 40 kbps e máximo de 640 kbps.

2.4.5 Códigos de Redundância Cíclicos

Os Códigos de Redundância Cíclicos, *Cyclic-Redundancy Check* (CRC), são códigos que as *tags* utilizam para validar certos comandos provenientes dos *readers*. Por sua vez os *readers* também utilizam estes códigos para validar informação proveniente das *tags*. Este protocolo usa dois tipos de CRC: CRC-16 e CRC-5.

Este códigos, embora tenham sido implementados no sistema, não são abordados em maior detalhe neste documento. Contudo, são apresentados os parâmetros necessários para a implementação dos mesmos.

O CRC-16 e o CRC-5 são caracterizado com parâmetros da tabela 2.3 e da tabela 2.4, respetivamente.

Length	Polynomial	Preset	Residue
16 bit	$x^{16} + x^{12} + x^5 + 1$	FFFF _h	1D0F _h

Tabela 2.3: Parâmetros CRC-16.

Length	Polynomial	Preset	Residue
5 bit	$x^5 + x^3 + 1$	01001 ₂	00000 ₂

Tabela 2.4: Parâmetros CRC-5.

2.4.6 Comandos do *Reader* e Respostas da *Tag*

Nesta secção, são abordados com maior detalhe as tramas de informação enviadas por parte do *reader*, e as respetivas respostas provenientes da *tag*.

Como referido anteriormente, neste documento só serão abordados com detalhe os comandos da *Inventory Round*.

- **Query**

O comando *Query* é responsável pelo início da *Inventory Round*. É neste comando que é definido o DR, o tipo de codificação da *tag*, e o uso do tom piloto.

Os parâmetros desta trama são (ver tabela 2.5):

DR: Constante usada para definir o BLF.

M: Número de sub-portadoras por símbolo. Define se a codificação usada pela *tag* é FM0 (M=1) ou Miller (M=2,4,8).

TRext: Define se o *Preamble* proveniente da *tag* contem tom piloto.

Sel: Escolhe que *tags* poderão responder a este comando.

Session: Escolhe o número da sessão para a *Inventory Round*.

Target: Seleciona o tipo de *tags* que respondem.

Q: Define o número de *slots* disponíveis.

CRC: CRC-5.

Os parâmetros, **Sel** e **Target**, são utilizado em ambientes onde existem múltiplos *readers*. Como este cenário não foi considerado neste trabalho, estes parâmetros não são detalhados nesta secção.

O comando *Query* é precedido por um R→T *Preamble*, apresentado na subsecção 2.4.3.

A tabela 2.5 (a) apresenta a trama do comando *Query* e a tabela 2.5 (b) apresenta a resposta correspondente por parte da *tag*.

	Command	DR	M	TRExt	Sel	Session	Target	Q	CRC
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilote tone 1: Use pilot tone	00: All 01: All 10: ~SL 11: SL	00: S0 01: S1 10: S2 11: S3	0: A 1: B	0-15	CRC-5

(a)

	Reply
# of bits	16
description	RN16

(b)

Tabela 2.5: (a) Trama do comando *Query*. (b) Resposta da *tag* ao comando *Query*.

- **QueryAdjust**

O comando *QueryAdjust* tem como função ajustar o parâmetro Q. Os parâmetros desta trama são:

Session: Escolhe o número da sessão *Inventory Round*.

UpDn: Determina se mantém o valor de Q, se aumenta uma unidade ou se diminui uma unidade.

A tabela 2.6 (a) apresenta a trama do comando *QueryAdjust* e a tabela 2.6 (b) apresenta a resposta correspondente por parte da *tag*.

	Command	Session	UpDnt
# of bits	4	2	3
description	1001	00: S0 01: S1 10: S2 11: S3	110: $Q = Q + 1$ 000: No change to Q 011: $Q = Q - 1$

(a)

	Reply
# of bits	16
description	RN16

(b)

Tabela 2.6: (a) Trama do comando *QueryAdjust*. (b) Resposta da *tag* ao comando *QueryAdjust*.

O comando *QueryAdjust* é precedido por um *Frame-sync*, apresentado, na subsecção 2.4.3.

• QueryRep

A função do *QueryRep* passa pelo envio de um comando à *tag*, de forma a que esta decemente uma unidade ao seu *slot counter*, até obter o número zero e transmitir o RN16.

Os parâmetros desta trama são:

Session: Escolhe o número da sessão da *Inventory Round*.

A tabela 2.7 (a) apresenta a trama do comando *QueryRep* e a tabela 2.7 (b) apresenta a resposta correspondente por parte da *tag*.

	Command	Session
# of bits	2	2
description	00	00: S0 01: S1 10: S2 11: S3

(a)

	Reply
# of bits	16
description	RN16

(b)

Tabela 2.7: (a) Trama do comando *QueryRep*. (b) Resposta da *tag* ao comando *QueryRep*.

O comando *QueryRep* é precedido por um *Frame-sync*, apresentado subsecção 2.4.3.

- **NAK**

O comando *NAK* tem como função sinalizar a *tag* que o *reader* não recebeu um EPC válido. A *tag* não responde a este comando.

O comando *NAK* é precedido por um *Frame-sync*, apresentado na subsecção 2.4.2.3.

A tabela 2.8 mostra a trama do comando *NAK*.

	Command
# of bits	8
description	11000000

Tabela 2.8: Trama do comando *NAK*.

- **ACK**

Por fim, o comando *ACK* tem como função sinalizar a *tag* que o *reader* recebeu, seja o RN16 ou um sinal de controlo (*handle* no caso da *Access Round*).

Os parâmetros desta trama são:

RN: RN16 ou *handle*, enviados previamente pela *tag*.

A tabela 2.9 (a) apresenta a trama do comando *ACK* e a tabela 2.9 (b) apresenta a resposta correspondente por parte da *tag*.

	Command	RN
# of bits	2	16
description	01	Echoed RN16 or handle

(a)

	Reply
# of bits	96
description	PC, EPC and PacketCRC

(b)

Tabela 2.9: (a) Trama do comando *ACK*. (b) Resposta da *tag* ao comando *ACK*.

O comando *ACK* é precedido por um *Frame-sync*, apresentado na subsecção 2.4.3.

Após referido a composição das tramas protocolares de envio e receção, é necessário expor as especificações temporais entre elas.

2.4.7 Link Timing

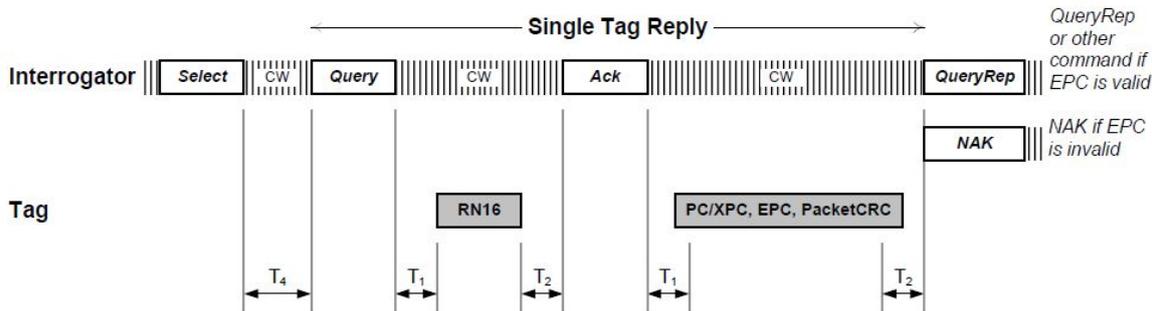
Consoante os comandos enviados pelo *reader* e a resposta da *tag*, existem tempos que têm que ser cumpridos. A tabela 2.10 apresenta os valores *standard* para estes tempos.

- T_1 é o tempo de resposta imediata da *tag* após receber um comando na *Inventory Round*.
- T_2 é o tempo que o *reader* tem para desmodular um sinal proveniente da *tag* e enviar um novo comando.
- T_3 é o tempo que o *reader* espera se não receber uma resposta por parte da *tag*.
- T_4 é o tempo mínimo entre o modo de operação *Select* e o modo de operação *Inventory Round*.
- FrT é a frequência de tolerância apresentada na tabela 2.2.

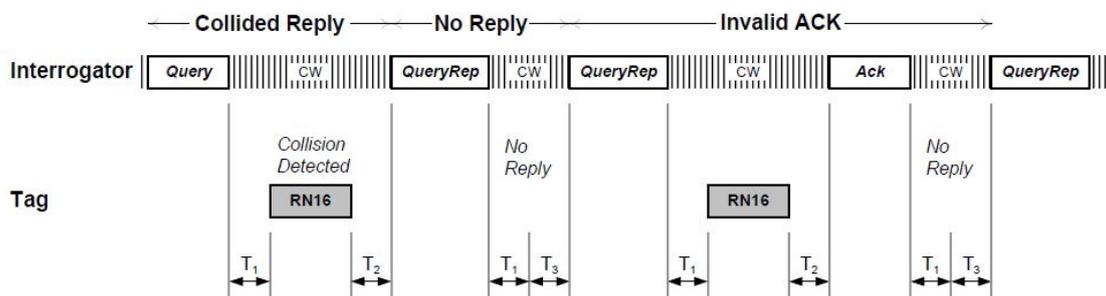
Parameter	Minimum	Nominal	Maximum
T_1	$\text{MAX}(\text{Rtcal}, 10 T_{\text{pri}}) (1 - \text{FrT}) - 2 \mu\text{s}$	$\text{MAX}(\text{Rtcal}, 10 T_{\text{pri}})$	$\text{MAX}(\text{Rtcal}, 10 T_{\text{pri}}) (1 + \text{FrT}) + 2 \mu\text{s}$
T_2	$3.0 T_{\text{pri}}$	-	$20.0 T_{\text{pri}}$
T_3	$0.0 T_{\text{pri}}$	-	-
T_4	2.0Rtcal	-	-

Tabela 2.10: Tempos relevantes para o protocolo de comunicação EPC Gen2 Class 1. $T_{\text{pri}} = 1/\text{BLF}$.

A figura 2.16 apresenta esquematicamente os tempos descritos na tabela 2.10.



(a)



(b)

Figura 2.16: (a) Tempos a respeitar pelo *reader* para leitura de uma única *tag*. (b) Tempos a respeitar pelo *reader* no caso de ocorrerem erros na leitura. Os erros que poderão ocorrer passam pela colisão de respostas em simultâneo, enviadas por múltiplas *tags*, a *tag* não enviar resposta ou a *tag* receber um *ACK* inválido (imagem retirada de [GS113a]).

Após a descrição da tecnologia RFID, em particular o protocolo ISO 18000-6C, é necessário implementar uma camada física que suporte todos os conceitos apresentados no presente capítulo.

Capítulo 3

Arquitetura de um Sistema RFID em SDR

Este capítulo aborda a temática *Software-Defined Radio* aplicada ao RFID. Tem ênfase nas arquiteturas de transmissão e recepção mais comuns nos sistemas de rádio SDR.

3.1 Introdução

Como referido no subcapítulo 1.2, *Software-Defined Radio* permite um sistema de rádio flexível, multi-banda reconfigurável e reprogramável por *software*. A figura 3.1 representa a implementação ideal da arquitetura SDR, onde todos os estágios de processamento de sinal na cadeia de transmissão (banda base→modulação→*up-conversion*→RF) e na cadeia de recepção (RF→*down-conversion*→desmodulação→banda base) são processados digitalmente [Mit95].

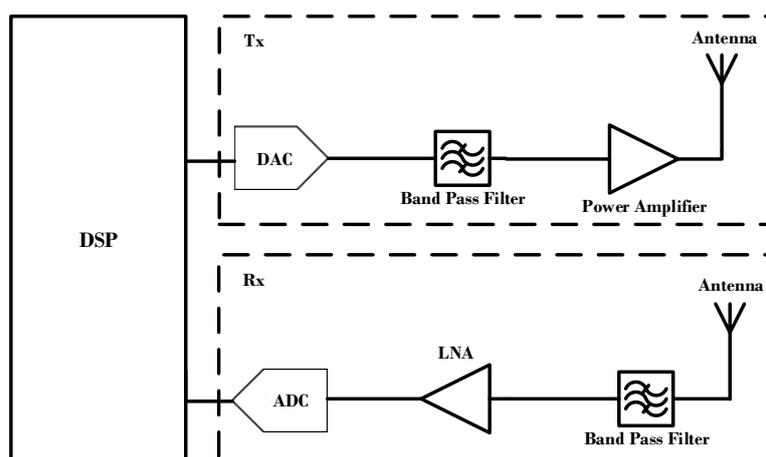


Figura 3.1: Arquitetura ideal SDR. Tx-Cadeia de transmissão; Rx-Cadeia de recepção.

No entanto, devido ao atual estado da arte dos componentes tais como ADCs (*Analog-to-Digital Converters*), DACs (*Digital-to-Analog Converters*), amplificadores e antenas, não é possível a implementação de um sistema SDR ideal [Nes10].

São apresentadas, de seguida, técnicas que permitem implementações reais próximas do sistema ideal.

3.2 Cadeia de Transmissão

Existem dois grandes tipos de arquiteturas tradicionais de transmissão de sinal rádio: [CCR10]

- **Conversão direta (*direct-conversion*) ou homodina.** Apresenta uma arquitetura onde o sinal de banda base é diretamente convertido para RF (ver figura 3.2).
- **Conversão de frequência intermédia (*dual-conversion*) ou heterodina.** É uma arquitetura mais complexa, onde o sinal em banda base é convertido primeiramente para uma frequência intermédia antes de sofrer a *up-conversion* para RF (ver figura 3.3).

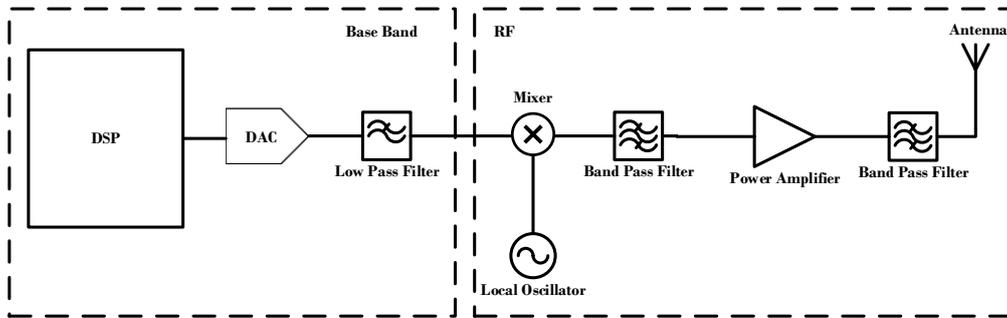


Figura 3.2: Arquitetura de transmissão homodina.

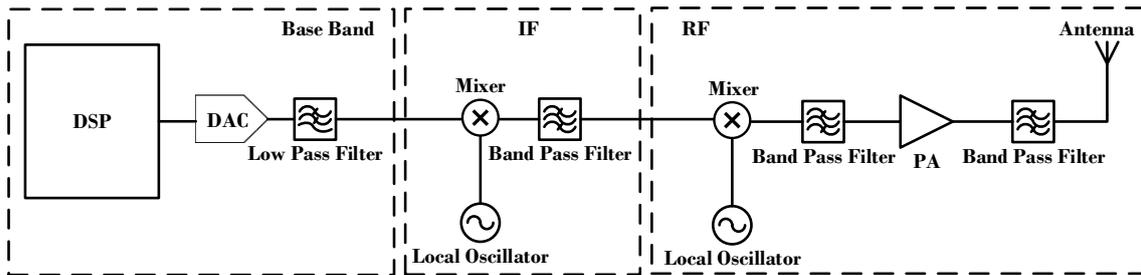


Figura 3.3: Arquitetura de transmissão heterodina.

A arquitetura homodina apresenta menor número de componentes. Geralmente esta arquitetura necessita unicamente de um oscilador local responsável pelo *up-conversion* do sinal em banda base diretamente para RF. Por sua vez, faz com que o sinal de saída esteja à frequência da onda portadora, gerada por este oscilador. Embora seja uma arquitetura mais simples de implementar, apresenta algumas desvantagens, tais como, possuir menor imunidade às fugas do oscilador, criando por sua vez ruído e componentes espúrias mais difíceis de filtrar em RF. Outra desvantagem está presente quando é pretendido uma modulação

em fase, pois o *mixer* tem que possuir precisão suficiente para controlar os atrasos de fase necessários [Dob07].

Por outro lado a arquitetura heterodina necessita de maior número de componentes. As arquiteturas deste tipo apresentam dois osciladores locais distintos. O primeiro, é responsável pelo *up-conversion* do sinal em banda base para uma frequência intermédia (IF), e o segundo, é responsável pelo *up-conversion* do sinal em IF para RF. Esta arquitetura apresenta grande flexibilidade, ou seja, com recurso a múltiplos osciladores, torna-se mais flexível quando é necessário realizar múltiplas conversões. Outra vantagem é a maior imunidade às fugas do oscilador local. Contudo, apresenta como desvantagens, tais como o facto de apresentar uma arquitetura mais complexa e a frequência do sinal de saída depende da diferença entre a frequência IF e a frequência do último oscilador na cadeia [Dob07].

3.2.1 Transmissores SDR

Existem várias abordagens para transmissores SDR. As mais comuns são [CCR10]:

- Utilizar DACs de elevada velocidade, onde o sinal é gerado numa componente IF, seguido de um transmissor homodino que se limita a fazer o *up-conversion* (ver figura 3.4);
- Utilizar DACs de elevada velocidade, onde o sinal é gerado já na componente RF (ver figura 3.5);
- Utilizar um transmissor completamente digital, onde o caminho do sinal a transmitir, desde a banda base até ao andar de RF, é integralmente digital (ver figura 3.6);

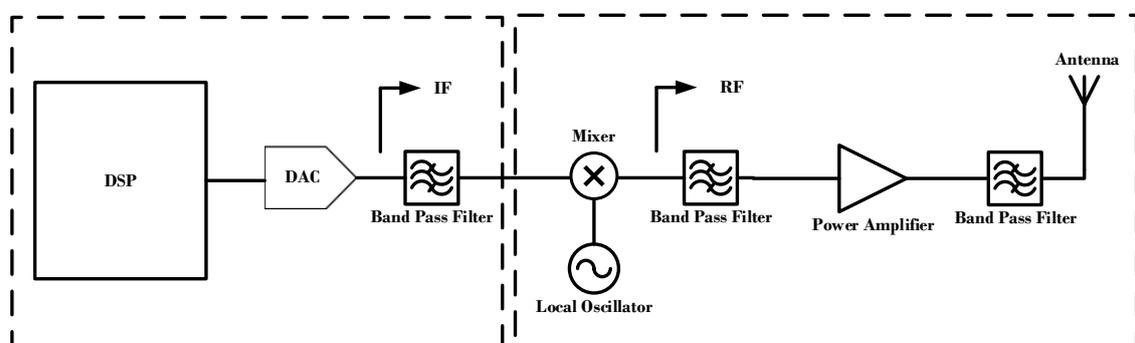


Figura 3.4: Arquitetura de transmissão SDR sinal é gerado numa componente IF.

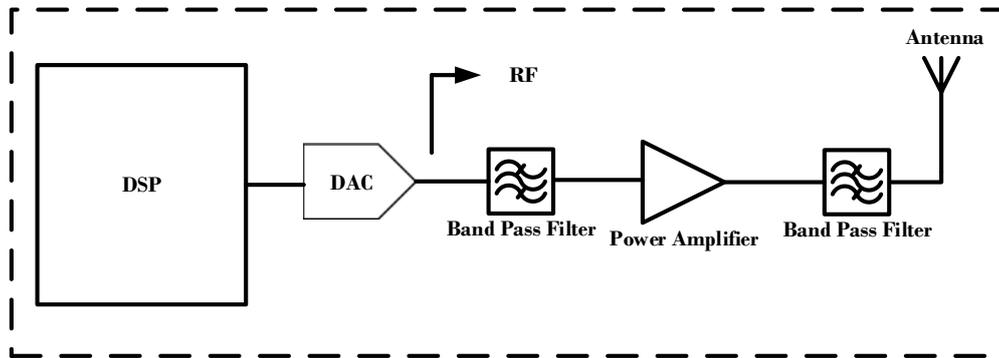


Figura 3.5: Arquitetura de transmissão SDR sinal é gerado numa componente RF.

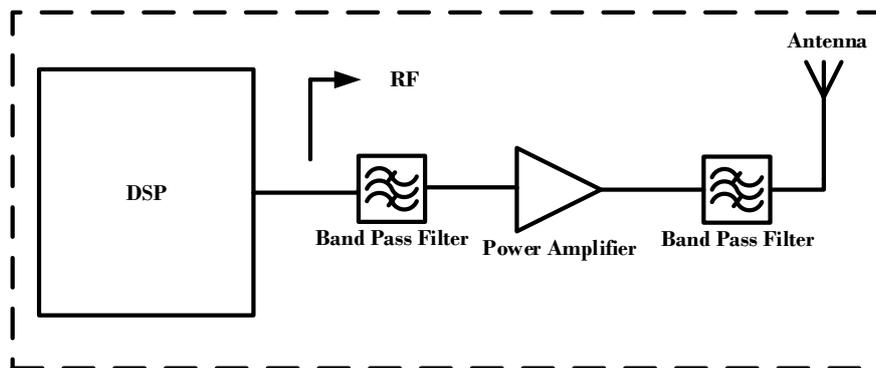


Figura 3.6: Arquitetura de transmissão SDR completamente digital.

3.3 Cadeia de Recepção

Homologamente às cadeias de transmissão, as cadeias de recepção para sinais de rádio também implementam arquiteturas homodinas ou heterodinas:

- **Conversão direta (*direct-conversion*) ou homodina.** Apresenta uma arquitetura onde o sinal de RF é diretamente convertido para banda base.
- **Conversão de frequência intermédia (*dual-conversion*) ou heterodina.** É uma arquitetura onde o sinal em RF é convertido primeiramente para uma frequência intermédia antes de sofrer a *down-conversion* para banda base.

3.3.1 Recetores SDR

Nos recetores SDR existem igualmente várias abordagens. As mais comuns são [CCR10]:

Digitalização em banda base: O sinal de rádio é desmodulado e convertido para banda base no domínio analógico. O sinal em banda base é processado logo após a ADC por um

DSP. É a arquitetura mais comum de receptores de rádio presente nos dias de hoje. Para além disso, é uma arquitetura de baixo custo de implementação e baixo consumo de potência da ADC. Contudo, encontra-se bastante longe da abordagem do SDR ideal [Pra13].

A figura 3.7 e 3.8 apresentam esquematicamente esta arquitetura.

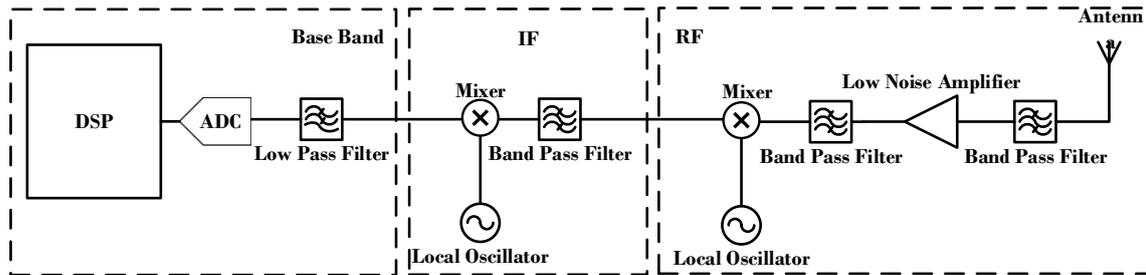


Figura 3.7: Arquitetura de recepção heterodina SDR com digitalização em banda base.

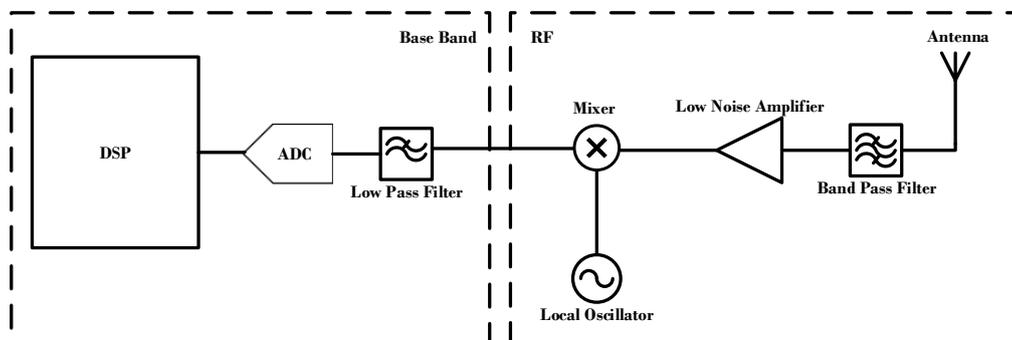


Figura 3.8: Arquitetura de recepção homodina SDR com digitalização em banda base.

Digitalização em IF (*Intermediate Frequency*): Este tipo de arquitetura corresponde à transposição da ADC para a zona, após o primeira *down-conversion*. O segundo *down-conversion*, conversão para a banda base, é feito no domínio digital através do uso de um DSP. Esta arquitetura permite receber uma maior largura de banda devido à possibilidade de realização de filtros no domínio digital. É de referir que esta encontra-se mais próxima abordagem do SDR ideal [Pra13].

A figura 3.9 apresenta esquematicamente esta arquitetura.

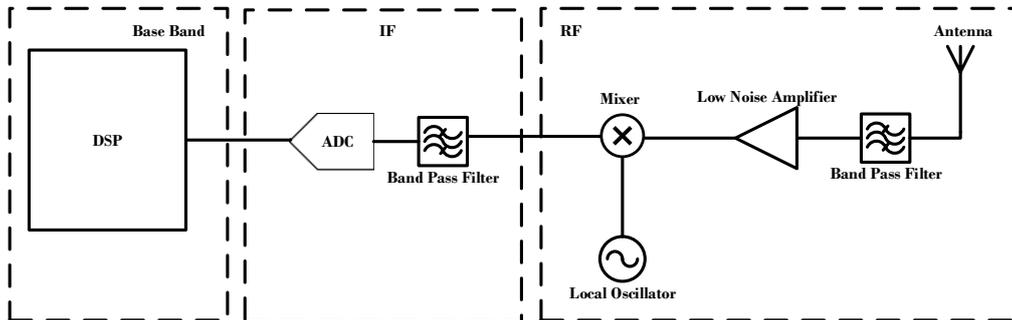


Figura 3.9: Arquitetura de recepção SDR com digitalização em IF.

Digitalização em RF: Nesta arquitetura a amostragem é feita diretamente ao sinal na componente RF. Esta arquitetura é difícil de implementar devido à ADC necessitar de uma elevada largura de banda, bem como uma maior taxa de amostragem de maneira a amostrar um sinal diretamente em RF. Esta é considerada a arquitetura SDR ideal [Pra13]

A figura 3.10 apresenta esquematicamente esta arquitetura.

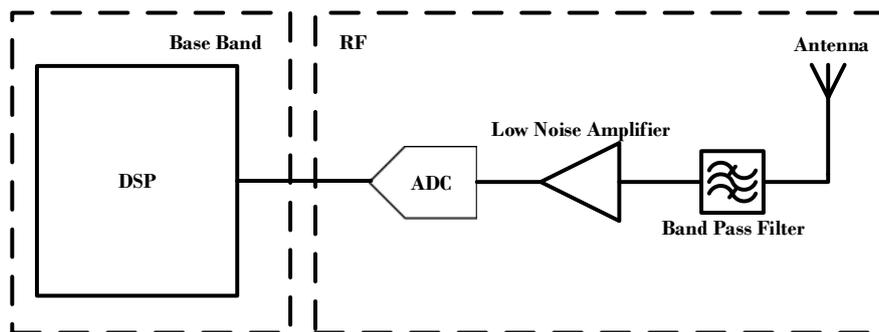


Figura 3.10: Arquitetura de recepção SDR com digitalização em RF.

3.4 Software-Defined Radio aplicado a RFID

Nesta secção são abordados arquiteturas de transmissão e recepção, expostos anteriormente, aplicados à tecnologia RFID.

3.4.1 Cadeia de Transmissão

Relativamente à cadeia de transmissão, a arquitetura de um sistema RFID vai depender do tipo de modulação implementado.

No caso em que a modulação implementada é DSB-ASK, não há necessidade de um sistema de elevada complexidade, uma vez que, o sinal em banda base é diretamente modulado para RF. Contudo, se o oscilador local não apresentar a frequência de operação necessária ao *up-conversion* é necessária uma abordagem heterodina.

No caso da modulação SSB-ASK e PR-ASK, é necessária uma abordagem heterodina, uma vez que, em ambas a modulação, é necessário gerar sinais em IF de maneira a controlar a fase do sinal a emitir.

3.4.2 Cadeia de Recepção

Relativamente à cadeia de recepção, a arquitetura a implementar vai depender dos recursos disponíveis.

Na situação onde o oscilador local atinge a frequência de operação suficiente para realizar o *down-conversion* do sinal diretamente para banda-base (não necessitando de passar pela frequência intermédia) e uma ADC de baixa largura de banda, uma abordagem com digitalização em banda base é tipicamente utilizada. Contudo, como referido anteriormente, esta abordagem está longe de uma arquitetura SDR ideal.

A digitalização em IF é a arquitetura utilizada num sistema RFID, quando o oscilador local não atinge a frequência de operação suficiente, e bate o sinal para uma frequência intermédia, que posteriormente, é amostrado por uma ADC e processada num DSP.

Por fim, a digitalização em RF seria o pretendido implementar, mas devido ao estado da arte dos componentes (ADCs, amplificadores, FPGAs), ainda é difícil uma implementação deste tipo.

3.4.3 Cadeia de Recepção e Transmissão no contexto deste Trabalho

No contexto deste trabalho de dissertação de mestrado, a cadeia de transmissão é baseada numa arquitetura homodina com digitalização em banda base, uma vez que foi implementada a modulação DSB-ASK. Por sua vez, a cadeia de recepção também é baseada numa arquitetura homodina, contudo, a digitalização é em IF (ver figura 3.11).

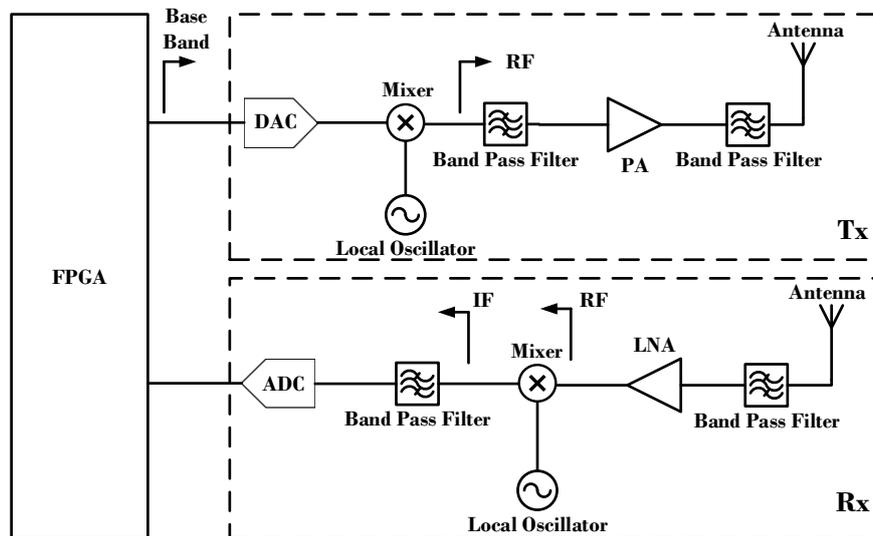


Figura 3.11: Arquitetura de recepção e transmissão SDR no contexto desta dissertação de mestrado.

Capítulo 4

Implementação do Sistema RFID

O capítulo é dividido em três secções. A primeira secção retrata a camada física analógica - *Analog PHY Layer* - do sistema, a segunda parte retrata a camada física digital - *Digital PHY Layer* - e por fim, a terceira parte retrata camada protocolar dos sistema - *MAC Layer*.

4.1 Introdução

Nesta dissertação de mestrado, devido à elevada largura de banda do RF *Front-End* utilizado, foi implementado um *reader* onde a cadeia de transmissão é baseada numa arquitetura homodina (o sinal em banda base sofre uma *up-conversion* diretamente para RF), e a cadeia de receção é baseada numa arquitetura homodina com digitalização IF, ou seja, o *Front-End* realiza um *down-conversion* do sinal em RF para uma frequência intermédia de 13 MHz.

Na camada física analógica, são abordadas as características associadas ao RF *Front-End AD-FMCOMMS1-EBZ* e antenas utilizadas, que permitiram a transmissão de um sinal *single side band* para a *tag*.

Na camada física digital são abordados os blocos, implementados em *hardware*, necessários à modulação DSB-ASK (cadeia de transmissão) e os blocos necessários à receção e desmodulação dos sinais provenientes de uma *tag*. Para este efeito foi utilizada na placa de desenvolvimento *Zedboard*.

Por fim, na camada MAC, são abordadas todas as funções desenvolvidas em C. Estas funções foram desenvolvidas de modo a implementar as tramas de transmissão, tempos de relevância, codificações, etc, necessários à camada protocolar ISO 18000-6C. Para este efeito, foi utilizado o ARM presente placa de desenvolvimento *Zedboard*.

A figura 4.1 apresenta os blocos implementados.

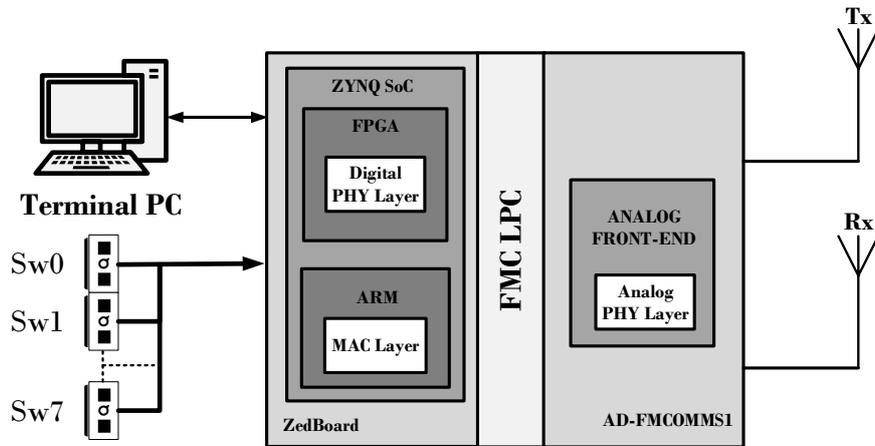


Figura 4.1: Blocos do sistema RFID implementado.

4.2 Camada Física Analógica - RF Front-end

Nesta secção é apresentada a camada física analógica utilizada neste trabalho de dissertação de mestrado. Entende-se por camada física analógica, os componentes responsáveis pelas conversões A/D e D/A, os componentes que permitem a *up-conversion* e *down-conversion*, os componentes que permitem filtragem e antenas para transmitir e receber sinais de rádio.

O RF *Front-End AD-FMCOMMS1-EBZ* desenvolvido pela *Analog Devices*, é um placa de desenvolvimento, desenhada para trabalhar em aplicações rádio, cujo processamento digital é baseado em FPGA. A figura 4.2 apresenta a placa de desenvolvimento utilizada.

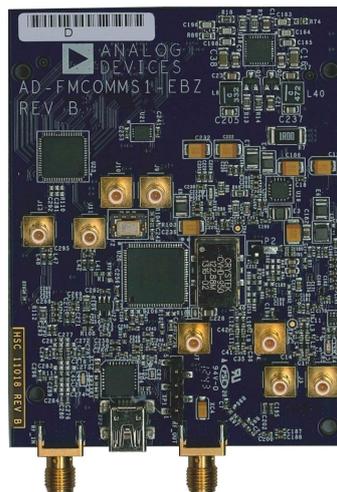


Figura 4.2: Placa de desenvolvimento *Front-End AD-FMCOMMS1-EBZ* (imagem retirada de [Dev13]).

Esta placa apresenta um conjunto de características, mecânicas, elétricas e de utilização,

úteis ao tipo de projeto que se pretende desenvolver, tais como [Dev14]:

- Possuir uma arquitetura modulador/desmodulador I/Q;
- Ser composto por duas ADCs 14-Bit @ [20; 250] MSPS;
- Ser composto por duas DACs 16-Bit @ [20; 1250] MSPS;
- Permitir uma portadora com uma frequência entre 400 MHz e 4 GHz.
- Possuir um interface FMC (*FPGA Mezzanine Card*) compatível com diversas placas de desenvolvimento baseadas em FPGA e SoC programáveis, por exemplo a *Zedboard*;
- Ter potência máxima radiada de 20 dBm;
- Ser disponibilizado um *reference design* com os IPcores para *interface* entre FPGA e DACs/ADCs.

Uma vez que não foi objetivo, nesta dissertação de mestrado, a implementação de um *front-end* analógico, esta placa de desenvolvimento foi escolhida com o intuito de haver flexibilidade entre esta camada física e as outras camadas (*digital PHY* e *MAC*) implementadas.

A figura 4.3 apresenta o diagrama de blocos do RF *front-end* utilizado no sistema.

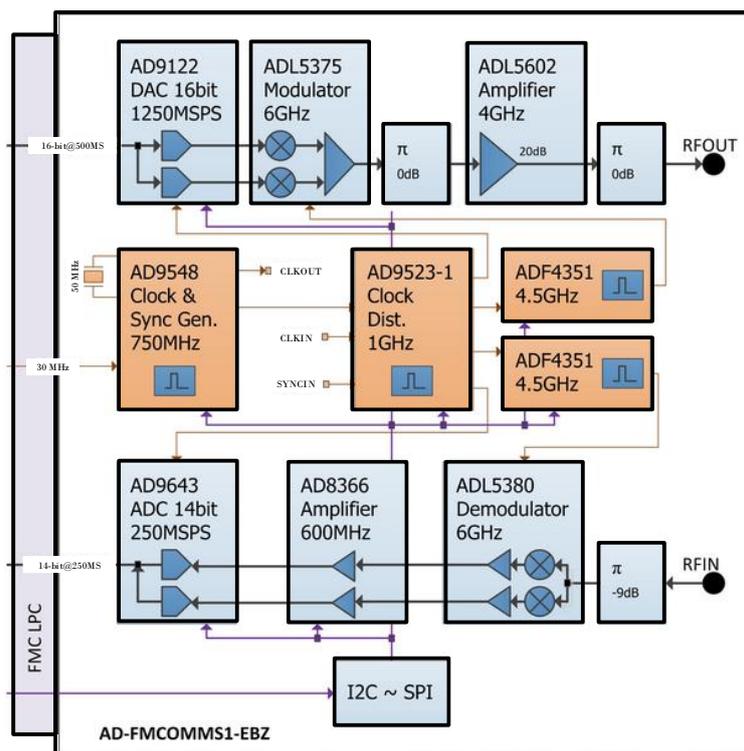


Figura 4.3: Diagrama de blocos da placa *Front-End AD-FMCOMMS1-EBZ* (imagem retirada de [Dev13]).

A descrição do diagrama de blocos da figura 4.3 segue de seguida.

4.2.1 Cadeia de Transmissão

Como já referido no subcapítulo 4.2, a *front-end* apresenta uma arquitetura I/Q na cadeia analógica de transmissão. Esta arquitetura foi escolhida de maneira a implementar um sinal SSB.

A cadeia de transmissão é apresentada em maior detalhe na figura 4.4.

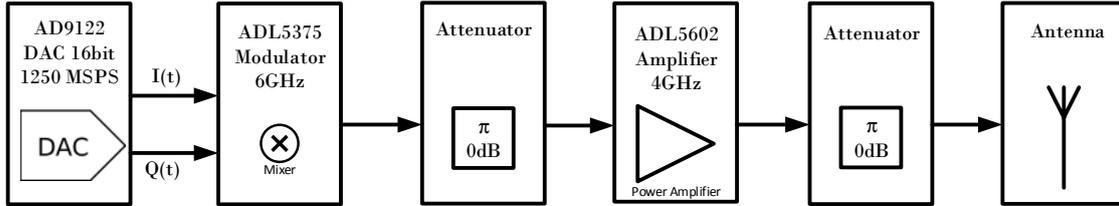


Figura 4.4: Diagrama de blocos da cadeia de transmissão.

Segue uma breve descrição dos blocos apresentados na figura 4.4. A descrição é feita segundo o fluxo do sinal, ou seja, do Bloco DAC para o Bloco Antena.

Bloco DAC - Reconstrução do Sinal R→T

Neste bloco, os sinais digitais I/Q, proveniente da FPGA, são reconstruídos em sinais analógicos por duas DACs independentes (DAC_I e DAC_Q) presentes no *chip* AD9122. Este *chip* também implementa uma cadeia de filtros interpoladores [Dev13] de maneira a mitigar réplicas/imagens criadas por este processo de reconstrução do sinal (ver Anexo C).

Aplicando os conceitos abordados pelo Teorema de Nyquist (ver Anexo C), modulação DSB-ASK (ver Anexo D) e pela informação da tabela 4.1 foi possível calcular a frequência de operação mínima das DACs, necessária para reconstruir um sinal em banda base, proveniente da FPGA, num sinal analógico.

Considerando o caso limite, ou seja, T_{ari} de $6.25 \mu s \Rightarrow PW = 3.125 \mu s$, foi possível calcular pela equação 4.1, a largura de banda máxima necessária para transmitir o sinal:

$$LB_{max} = 2 \times \frac{1}{3.125 \mu s} = 2 \times 320 kHz = 640 kHz \quad (4.1)$$

Obtida a largura de banda máxima, foi calculada a frequência de operação mínima:

$$f_s = 2 \times f_{max} \Leftrightarrow f_s = 2 \times 640 kHz = 1.280 MHz \quad (4.2)$$

Pela equação 4.2 concluiu-se, que para reconstruir um sinal deste tipo, seria necessário uma DAC com $f_{smin} = 1.280 MHz$. Contudo, a DAC do RF *Front-End*, tem como frequência de operação mínima 20 MSPS. A DAC foi então configurada com a frequência de operação 61.440 MSPS. Esta frequência foi escolhida devido a ser um valor de referência do *clock* AD9523 do RF *Front-End* [Dev10] e por se encontrar acima do limite mínimo de 20 MSPS.

Bloco Modulador I/Q - Sinal *Single Side Band*

Após a reconstituição, os sinais em fase e em quadratura sofrem um *up-conversion* no modulador ADL5375, e são somados entre si.

Esta arquitetura tem o intuito de gerar um sinal SSB (*Single Side Band*), ou seja, um sinal onde tanto a frequência da portadora (f_c) como uma das bandas laterais (*Low Side Band* (LSB) ou *Upper Side Band* (USB)) são suprimidas. Foi escolhido um sinal SSB devido às vantagens que possui [Tod12]:

- Conservação de potência: como só uma banda lateral é transmitida, não é necessário energia para transmitir a portadora e a segunda banda de frequência;
- Utilização mais eficiente do espectro: mais uma vez, como só é transmitida uma banda lateral, o transmissor só necessita de metade da largura de banda;
- O ruído térmico é reduzido para metade dos sistemas que usam *double side band*;

Contudo apresenta a desvantagem de possuir uma maior complexidade ao nível dos seus componentes.

A figura 4.5 apresenta um esquema básico da geração de um sinal SSB no caso onde é pretendido obter a banda lateral inferior LSB.

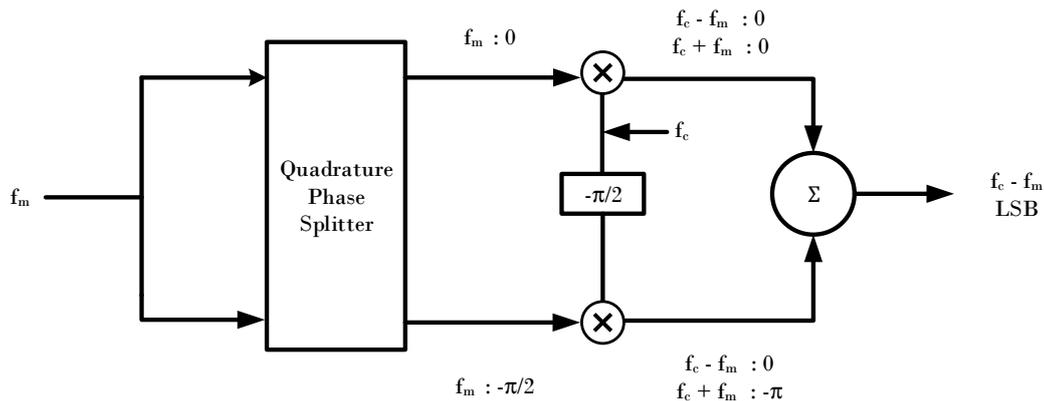


Figura 4.5: Arquitetura para gerar um sinal SSB. O sinal f_m é desfasado $\pi/2$ pelo bloco *quadrature phase splitter*, criando assim um sinal em fase ($f_m : 0$) e em quadratura ($f_m : \pi/2$). Estes sinais são multiplicados respetivamente por uma portadora em fase e em quadratura. Somando todas as componentes criadas, obtém-se a banda lateral LSB centrada em $f_c - f_m$ [Uni12].

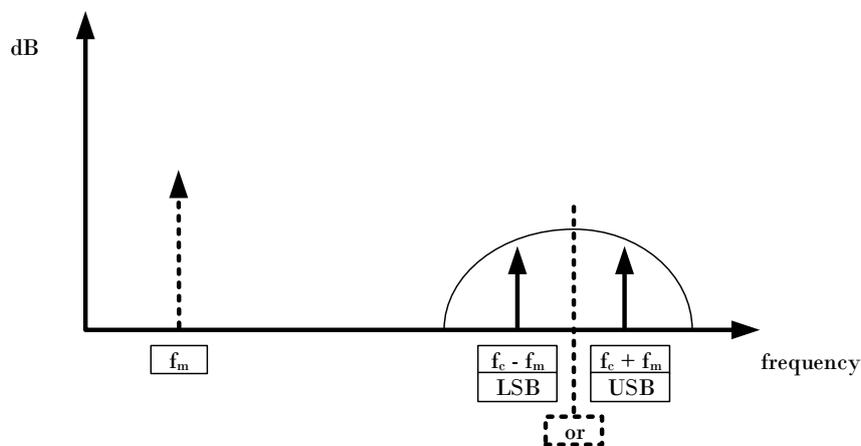


Figura 4.6: Sinal SSB no espectro de frequências.

A figura 4.6 apresenta graficamente um sinal SSB em termos de espectro de frequência.

Bloco Amplificador de Potência

Como o nome indica, neste bloco, o sinal modulado é amplificado pelo amplificador presente no *chip* ADL5602. Este amplificador aplica um ganho de 20 dB [Dev14].

Bloco Antena

Na cadeia de transmissão foi utilizada uma antena desenvolvida no Instituto de Telecomunicações (IT), pólo de Aveiro. Tem como principais características¹

- Frequência de operação: 860 - 915 MHz;
- Polarização: Circular;
- Ganho: 5.3 dBi

A figura 4.7 apresenta a antena utilizada.

¹medições realizadas na câmara Anecoica com a antena padrão "Kathrein Indoor Directional Antenna 870-960 MHz".

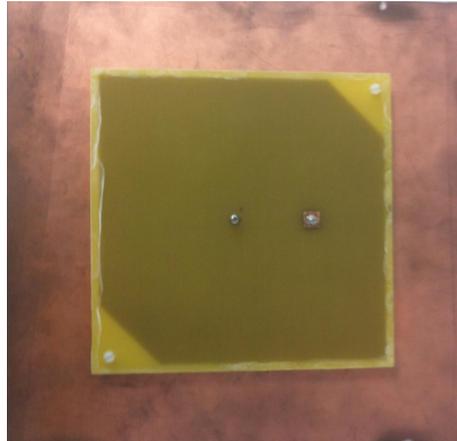


Figura 4.7: Antena desenvolvida no IT, pólo de Aveiro.

Terminada a cadeia de transmissão analógica, segue a cadeia de receção.

4.2.2 Cadeia de Receção

Similarmente à cadeia de transmissão, a cadeia de receção do *Front-End AD-FMCOMMS1 EBZ* também possui uma arquitetura I/Q, neste caso, desmodulador I/Q. A figura 4.8 apresenta o diagrama de blocos deste sistema.

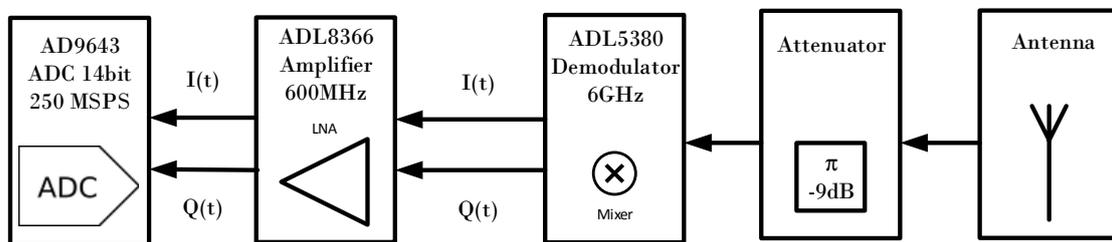


Figura 4.8: Diagrama de blocos da cadeia de receção.

Segue agora uma breve descrição dos blocos apresentados na figura 4.8. A descrição é feita segundo o fluxo do sinal, ou seja, do Bloco Antena para o Bloco ADC.

Bloco Antena

A antena utilizada na cadeia de recepção foi a antena comercial ALR-8610-AC do fabricante *Alien*. Tem como principais características: [AIS04]

- Frequência de operação: 865 - 965 MHz;
- Polarização: Circular;
- Ganho: 5.5 dBi;

A figura 4.9 apresenta a antena utilizada.



Figura 4.9: Antena ALR-8610-AC *Alien*.

Bloco Desmodulador I/Q

O método de funcionamento do desmodulador I/Q é similar ao modulador mas com diferença no sentido do fluxo do sinal. Um sinal é captado pela antena e o desmodulador, a partir do mesmo sinal, gera uma componente em fase e em quadratura que, seguidamente, é amostrado pela ADC. A figura 4.10 apresenta um esquema básico de um desmodulador I/Q.

Em termos de frequência, e visto que foi objetivo obter digitalização em IF, a cadeia de desmodulação foi programada de maneira ao sinal proveniente da *tag* sofre uma *down-conversion* para uma frequência intermédia de 13 MHz. A figura 4.11 retrata o conceito.

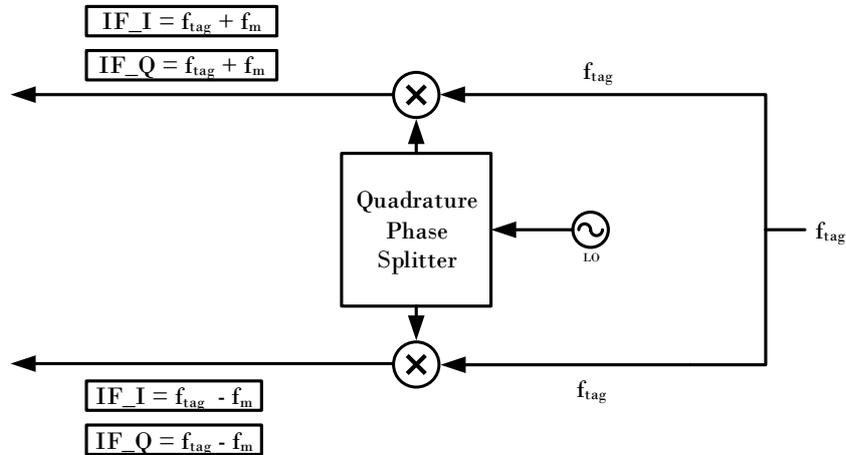


Figura 4.10: Arquitetura de um recetor I/Q. O sinal proveniente do oscilador local(LO) é desfasado $\pi/2$ pelo bloco *quadrature phase splitter*, criando assim um sinal em fase e em quadratura. Estes sinais são multiplicados respetivamente pelo sinal proveniente da *tag*, resultado assim em quatro componentes. Duas frequências, em fase e quadratura, centradas na frequência intermédia $f_{tag}-f_m$ e outras duas, também em fase e quadratura, centradas na frequência intermédia $f_{tag}+f_m$ [Uni12].

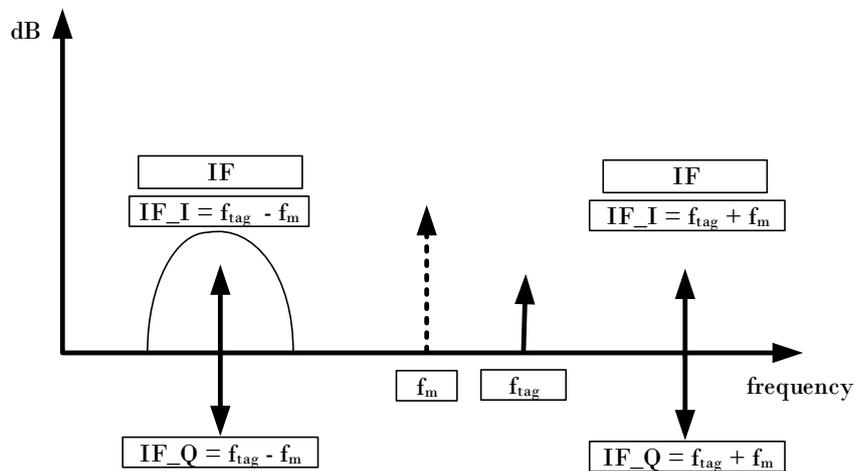


Figura 4.11: Desmodulação no espectro de frequências.

Bloco Amplificador de Baixo Ruído

Como o nome indica, neste bloco, o sinal I/Q proveniente do desmodulador é amplificado pelo LNA presente no *chip* ADL8366. Este amplificador é programável aplica um ganho entre 4.5dB e 20 dB [Dev14].

Bloco ADC - Amostragem do Sinal T→R

Similarmente à reconstituição do sinal, aplicando os conceitos abordados pelo Teorema de Nyquist (Anexo C) e pela informação da tabela 2.2 foi possível calcular a frequência de amostragem mínima necessária para amostrar um sinal proveniente de uma *tag*.

Considerando IF de 13.01 MHz e o caso extremo, ou seja, BLF = 640 kHz, o que implica, na codificação FM0, um *data-rate* de 640 kbps.

$$f_s = 2 \times f_{max} \Leftrightarrow f_s = 2 \times (13.01 + 0.640) \text{ MHz} = 27.30 \text{ MHz} \quad (4.3)$$

Pela equação 4.3 concluiu-se, que para amostrar um sinal deste tipo, seria necessário uma ADC com $f_{smin} = 27.30$ MHz.

Concluiu-se que a frequência de amostragem a 61.440 MSPS é adequada, à amostragem do sinal proveniente da *tag*.

4.2.3 Cadeia de Relógios

A cadeia de relógios da placa de desenvolvimento pode ser alimentada por um cristal *on board* de 50 MHz ou então por um relógio de referência enviado pela FPGA através da porta FMC [Dev14].

Esta cadeia é composta por quatro *chips* [Dev14]:

- O AD9548 é um *network clock generator/synchronizer* que recebe o relógio de referência, proveniente da FPGA, e alimenta o *chip* AD9523-1;
- O AD9523-1 é *low jitter clock generator* que fornece um relógio de referência à DAC, ADC e aos dois ADF4351;
- Os dois ADF4351 correspondem aos LOs da cadeia de transmissão e recepção.

4.2.4 Programação dos Componentes

A programação dos componentes é realizada com recurso aos protocolos I2C e SPI, ou seja, as configurações, provenientes da FPGA, são enviadas em I2C, e posteriormente convertidas em SPI na placa *AD-FMCOMMS1-EBZ*.

O I2C é utilizado pela FPGA uma vez que, face ao SPI, para implementar são necessários unicamente dois fios, um para dados e outro para relógio. O SPI necessita de pelo menos quatro fios.

Na placa *AD-FMCOMMS1-EBZ* é utilizado o SPI uma vez que, este transmite dados a uma velocidade muito maior que o I2C, ou seja o SPI permite transferências *full-duplex* com velocidades até 10 Mbps e o I2C não possui *full-duplex* e atinge velocidades entre 1 Mbps e 3.4 Mbps [Byt13].

Neste trabalho de dissertação, os componentes desta placa de desenvolvimento foram programados, no código em C desenvolvido, com os seguintes valores:

- DAC: 61.440 MSPS;

- ADC: 61.440 MSPS;
- Modulador: [865.7; 866.3; 866.9; 867.5] MHz;
- Desmodulador: [852.7; 853.3; 853.9; 854.5] MHz;
- LNA: 20 dB.

4.3 Camada Física Digital

Nesta secção é apresentada a camada física digital desenvolvida neste trabalho de dissertação de mestrado. Entende-se por camada física digital, todos os componentes responsáveis pelo processamento digital do sinal.

A *ZedBoard* é uma placa de desenvolvimento baseada na *Extensible Processing Platform* (EPP) *Zynq-7000* da *Xilinx*. Contém um *dual Cortex-A9 Processing System* (PS). A *Zedboard* é uma placa com elevado número de periféricos e com capacidade de expansão. Esta placa foi escolhida, por apresentar um conjunto de características específicas úteis à implementação do sistema desenvolvido, tais como [Zed12]:

- Conter uma FPGA com capacidade lógica reprogramável adequada à implementação da camada física digital e um processador ARM útil no processamento do protocolo de comunicação;
- Possuir um conjunto de periféricos que possibilitam a *interface* com o utilizador, tais como *interface USB-UART* para troca de informação entre a FPGA e um terminal (computador) e *switchs* para mudar a frequência de operação do *reader*. Também possui componentes como um memória 512 MB DDR3 para armazenar informação;
- Possuir *Interface FMC LPC (Low Pin Count)* útil para a ligação entre a FPGA e o *front-end* de RF;

A figura 4.12 apresenta a placa *Zedboard* usada no sistema.

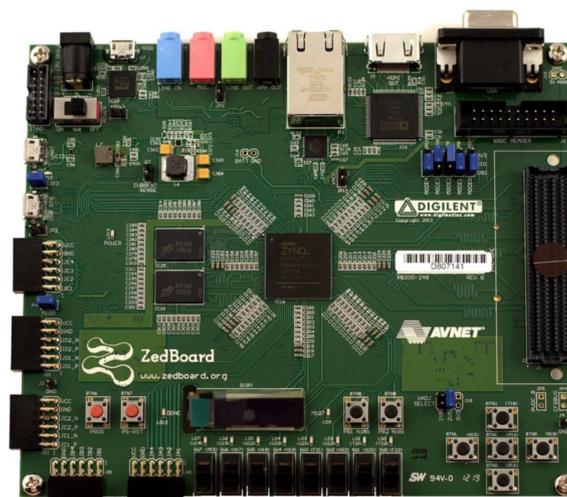


Figura 4.12: Placa de desenvolvimento *Zedboard* (imagem retirada de [Zed12]).

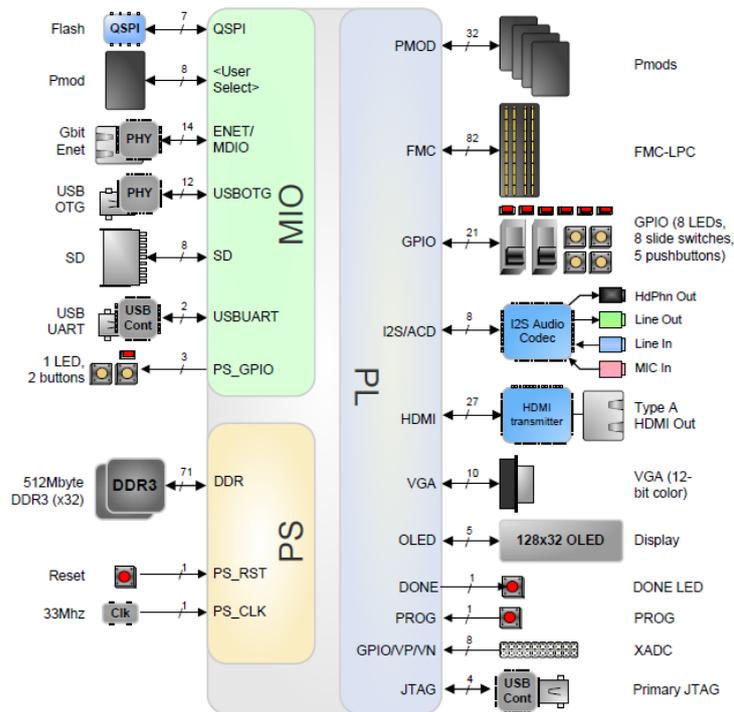


Figura 4.13: Diagrama de blocos da placa *Zedboard* (imagem retirada de [Zed12]).

4.3.1 Cadeia de Transmissão

A cadeia de transmissão referente à parte digital pode ser dividida em 6 blocos principais (ver figura 4.14):

- **Bloco ZYNQ IPcore (1)** [Xil13]: Bloco de processamento que oferece programabilidade de *software*, *hardware* e I/O. Este bloco incorpora o processador *dual core* ARM *Cortex-A9*. É nesta unidade de processamento que toda a camada protocolar (camada MAC) do ISO 18000-6C foi implementada. As funções de modulação DSB-ASK e codificação PIE também são aqui processadas, ou seja, a função de modulação DSB-ASK envia um sinal de seleção ao *multiplexer*, e por sua vez, a função de codificação PIE estabelece a duração desse sinal;

- **Bloco Multiplexer (2)**: Bloco que realiza a multiplexagem entre duas palavras de 16 bit: 0x7FFF e 0x0B85. Estas palavras são usadas para obter uma modulação DSB-ASK com profundidade 91%. A multiplexagem é conseguida através de um sinal de seleção proveniente da função *DSB-ASK MODULATOR* do bloco *ZYNQ IPcore*, ver figura 4.14;

- **LP Filter (3)** [Xil11]: Bloco que implementa um filtro FIR passa-baixo com frequência de corte 200 kHz;

- **Bloco IPcore AD9122 (4)** [Dev09]: Bloco que realiza a *interface* (DAC LVDS *inter-*

face) entre a FPGA e a DAC. Este IPcore foi fornecido pelo *reference design* desenvolvido pela Analog Devices, contudo, foi modificado de acordo com as características da cadeia de transmissão implementada. Estas alterações consistiram em remover uma cadeia de paralelização de informação, implementada com recurso a DDS's, com o objetivo de poupar recursos lógicos da FPGA;

- **Bloco *reference clock* IPcore (5)** [Xil14b]: Bloco que gera um relógio de referência de 30 MHz usado pelo RF *Front-End* na sua cadeia de osciladores locais;

- **Bloco *IIC* IPcore (6)** [Xil14a]: Bloco que implementa o protocolo de comunicação IIC, utilizado pelo RF *Front-End AD-FMCOMMS1 EBZ* para efeitos de configuração dos seus componentes, tais como, frequência de amostragem da DAC e ADC, frequência do Modulador e Desmodulador e ganho do LNA. Foi implementado em C uma *driver* que permite a comunicação entre a unidade de processamento e este bloco;

Os *switches* SW1, SW2 e SW3, presentes na figura 4.14, estão destinados à alteração em tempo real da frequência de operação do *reader*, de maneira a este (o *reader*) operar nos quatro canais de frequência impostos pela ANACOM.

O SW1 ativa esta funcionalidade e as combinações '00', '01', '10' e '11' de SW2 e SW3, correspondem a 863.7 MHz, 866.3 MHz, 866.9 MHz e 867.5 MHz respetivamente.

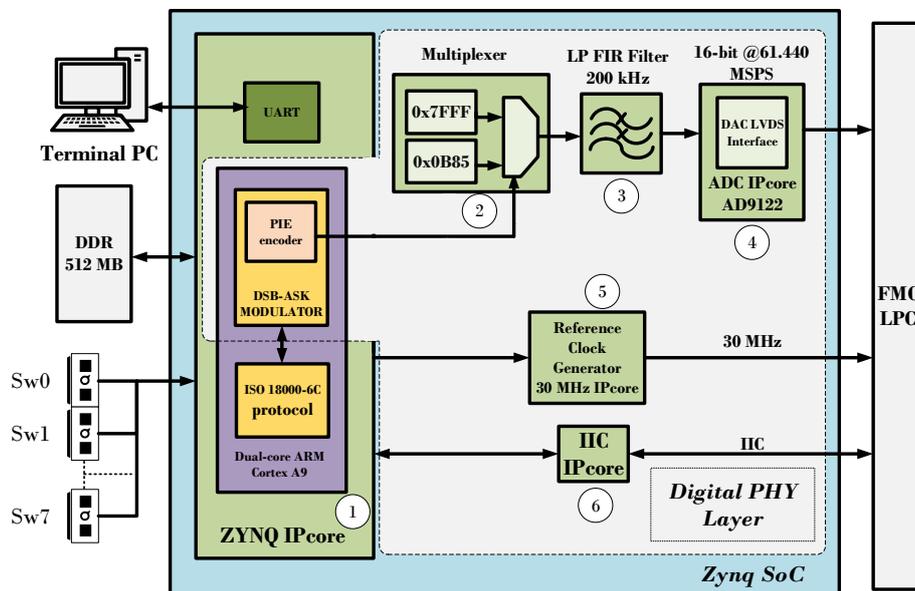


Figura 4.14: Diagrama de blocos cadeia de transmissão implementada na FPGA.

De seguida, na tabela 4.1, são expostos os parâmetros da camada física digital programados para a cadeia de transmissão.

Modulação	PW (μ s)	Tari (μ s)	Data-0 (μ s)	Data-1 (μ s)	RTcal (μ s)	TRcal (μ s)
DSB-ASK (91%)	10.4	20.8	20.8	36.2	58.7	96.2

Tabela 4.1: Parâmetros temporais da cadeia de transmissão.

Finalizada a descrição da cadeia de transmissão, é agora explicado com maior detalhe, os estágios necessários para a criação de um sinal DSB-ASK.

• Validação da Cadeia de Transmissão em Matlab

Antes de implementar a cadeia de transmissão na FPGA, esta teve que ser validada. Para tal, foi desenvolvida uma rotina em Matlab. Esta rotina simula um sinal em banda base, constituído por uma sequência de de oito símbolos “Data-0”, a ser transmitido para o *mixer* após passagem por um filtro FIR passa baixo.

Foram escolhidas duas palavras, 0x0B85 e 0x7FFF, para gerar o sinal modulante em banda base. Estas palavras foram escolhidas com o propósito de criar uma modulação ASK com profundidade 91%, ou seja, uma vez que a DAC está programada para operar com palavras de 16-bit em complemento para 2, a palavra 0x7FFF é o valor positivo de amplitude máxima e, por sua vez, a palavra 0x0B85 corresponde a 9% da amplitude do mesmo. O sinal em banda base toma o aspeto da figura 4.15.

É agora apresentada esta simulação:

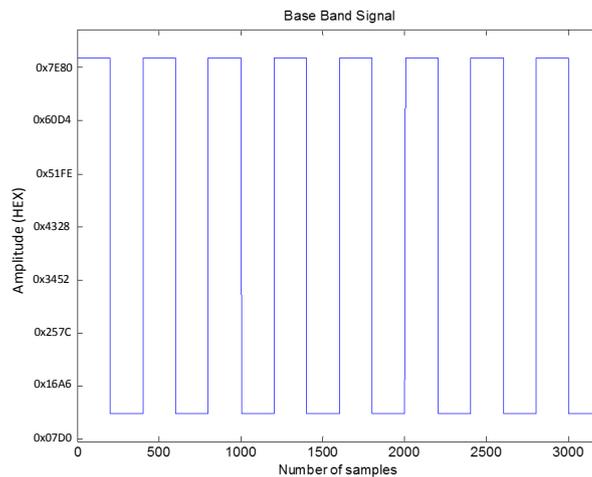


Figura 4.15: Sequência de de oito símbolos “Data-0” gerados em Matlab.

Caso o sinal, apresentado na figura 4.15, não sofresse mais nenhuma alteração² desde a sua criação até ao *up-conversion*, o sinal ASK resultante apresentaria uma queda abrupta

² não está a ser considerado uma alteração a passagem do sinal do domínio digital para o domínio analógico.

entre o nível de maior amplitude e de menor amplitude. Este facto resulta num espalhamento da potência por todo o espectro, poluindo assim todos os canais adjacentes (ver figura 4.16). De maneira a evitar este grave problema, é necessário criar uma transição suave entre estes dois níveis de amplitude. Esta transição é conseguida através da filtragem do sinal por um filtro passa-baixo [Dob07].

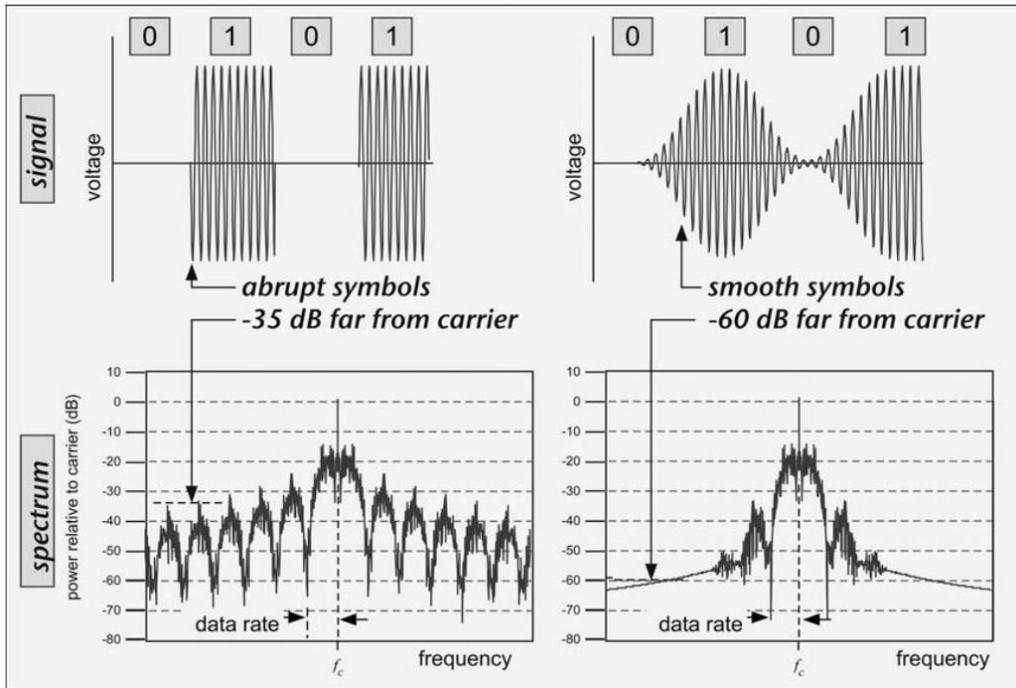


Figura 4.16: Diferenças espectrais entre um sinal ASK com transição abrupta e transição suave (imagem retirada de [Dob07]).

Foi implementado um filtro FIR passa-baixo com os parâmetros apresentados na tabela 4.2. (O conceito de filtro e as suas propriedades são abordadas no Anexo E deste documento).

Low-Pass Filter				
Design Method	β	Order	Fs (MHz)	Fc (MHz)
Kaiser Window	2.5	80	15.36	0.20

Tabela 4.2: características do filtro FIR passa-baixo.

Este filtro foi projetado em Matlab com recurso à ferramenta *Filter Design & Analysis Tool* e os parâmetros de configuração, apresentados na tabela 4.2, foram obtidos da seguinte forma:

- Os parâmetros “*Design Method*”, “ β ” e “*Order*” foram obtidos com recurso a várias iterações de maneira ao filtro apresentar um comportamento desejado.

- O parâmetro F_s (frequência de amostragem) foi 15.36 MHz para simular um relógio proveniente da DAC quatro vezes menor que a frequência de operação 61.440 MHz.

- O parâmetro F_c (frequência de corte) foi 200 kHz uma vez que esta frequência corresponde à largura de banda do sinal em banda base (192.3 kHz) (ver equação 4.5).

$$PW = 10.4 \mu s \Leftrightarrow \frac{1}{PW} = \frac{1}{10.4} kHz = 96.15 kHz \quad (4.4)$$

$$LB = 2 \times \frac{1}{PW} \Leftrightarrow LB = 2 \times 96.1 kHz = 192.3 kHz \quad (4.5)$$

A figura 4.17 apresenta o comportamento do filtro FIR passa-baixo desenvolvido.

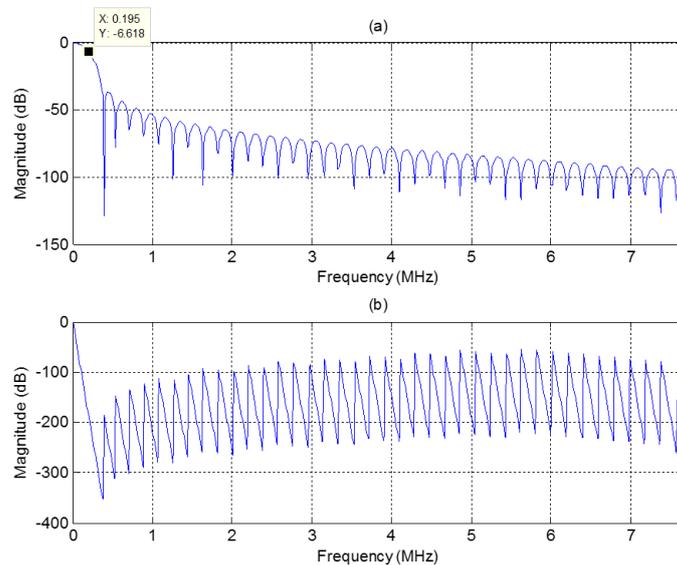


Figura 4.17: (a) Resposta impulsional do filtro em função da frequência. (b) Amplitude da fase do filtro em função da frequência

A figura 4.18 apresenta a simulação em Matlab do sinal da figura 4.17, após a passagem pelo filtro passa-baixo.

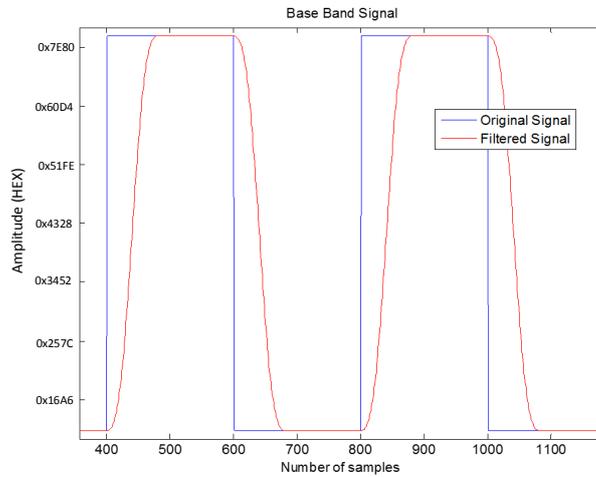


Figura 4.18: Sinal original e sinal após a passagem do filtro passa-baixo.

Pela figura 4.18 é possível notar a transição suave entre os diferentes níveis de amplitude do sinal.

A figura 4.19 apresenta a simulação, em Matlab, dos sinais da figura 4.18, após sofrer uma *up-conversion*.

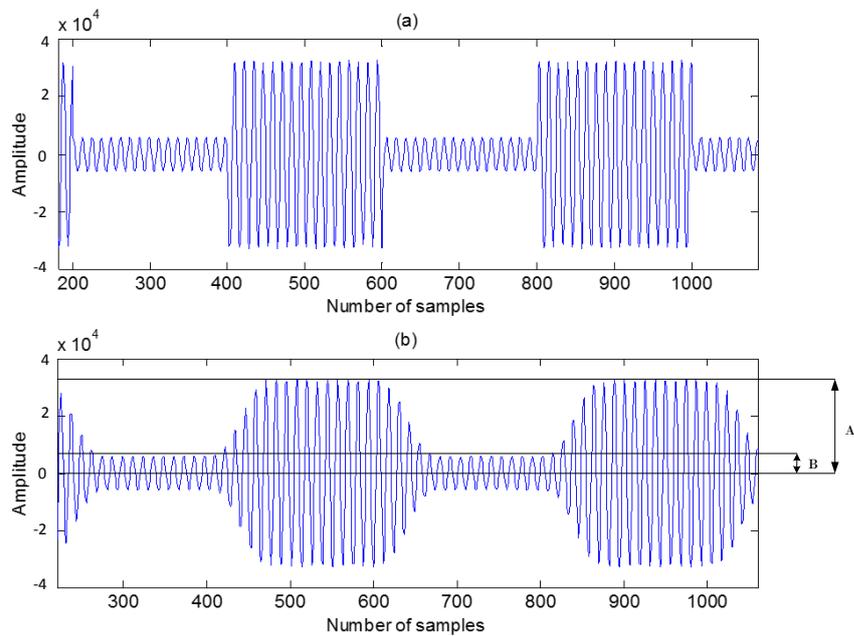


Figura 4.19: (a) *Up-conversion* do sinal antes do filtro passa-baixo. (b) *Up-conversion* do sinal após o filtro passa-baixo.

Pela figura 4.19 e equação 4.6, comprovou-se uma modulação com profundidade de 91%.

$$h = \frac{A - B}{A} = \frac{32767 - 2949}{32767} = 0.91 \quad (4.6)$$

Validada a cadeia de transmissão em Matlab, esta foi implementada na FPGA.

• Implementação da Cadeia de Transmissão na FPGA

O sinal modulante em banda base foi gerado com recurso aos blocos (1), (2) e (3) da figura 4.14.

Para gerar as palavras 0x0B85 e 0x7FFF, foi implementado o bloco (2) com recurso a um GPIO com uma saída de 16-bit. Este é controlado pelo bloco de processamento ZYNQ IPcore(1), ou seja, no bloco DSB-ASK Modulator é indicado ao GPIO se deve transferir a palavra 0x0B85 ou a palavra 0x7FFF ao filtro passa-baixo.

O filtro passa-baixo, bloco (3), foi implementado com recurso ao FIR *Compiler* disponibilizado pela Xilinx. Este filtro foi projetado com os coeficientes obtidos pela simulação em Matlab e configurado para funcionar no modo single-rate, com uma frequência de operação de 15.36 MHz e precisão de 16-bit em complemento para dois.

O sinal proveniente do filtro passa-baixo foi convertido para o domínio analógico com recurso ao bloco (4). Como referenciado anteriormente, este bloco implementa a *interface* entre a FPGA e a DAC. Este foi programado para operar a uma frequência de amostragem de 61.440 MSPS.

O bloco (5) foi implementado com recurso ao *Clocking Wizard* da Xilinx. Foi programada para trabalhar com a primitiva MMCM, possui como *input* um relógio de 200 MHz proveniente do bloco ZYNQ IPcore(1) e gera um *output* de 30 MHz.

Por fim, o bloco IIC IPcore(6) foi implementado através do *reference design*.

Terminada a cadeia de transmissão, segue em detalhe, a cadeia de receção.

4.3.2 Cadeia de Receção

A cadeia de receção apresenta maior complexidade face à cadeia de transmissão. Além dos blocos ZYNQ IPcore e IIC IPcore, referenciados na cadeia de transmissão, pode ser dividida em 6 blocos principais (ver figura 4.20):

- **Bloco IPcore AD9643** (1) [Dev11]: Bloco que realiza a interface (ADC LVDS *interface*) entre a FPGA e a ADC. Este bloco disponibiliza, entre outras, uma saída de 32-bit de data (16-bit menos significativos é o sinal em fase; 16-bit mais significativos é o sinal em quadratura) e uma saída de relógio, *ADC_clk*, que permite a sincronização da cadeia de receção. Este IPcore foi fornecido pelo *reference design* desenvolvido pela Analog Devices;

- **Power by 2** (2): Bloco que eleva o sinal ao quadrado. Primeira fase para detetar a envolvente de um sinal modulado em ASK;

- **LP Filter** (3) [Xil11]: Bloco que implementa um filtro FIR passa-baixo com frequência de corte 2 MHz seguido de um decimador de rácio 20. O filtro passa-baixo teve o objetivo de remover as componentes de alta frequência do sinal proveniente do quadrador. Este filtro foi projetado com a frequência de corte 1 MHz, uma vez que, a largura de banda máxima do sinal proveniente da *tag* é 640 kHz. Por sua vez, o decimador de rácio 20 foi implementado para diminuir a frequência de amostragem do sinal, ou seja, diminuir de 61.440 MSPS para 3.072 MSPS, tornando assim, o sistema mais rápido a decodificar a informação da *tag*;

- **HP Filter** (4) [Xil11]: Bloco que implementa a remoção de DC através de um filtro FIR passa-alto com frequência de corte 50 kHz. Este filtro foi projetado com a frequência de corte 50 kHz, uma vez que, o sinal de interesse, no sistema desenvolvido, vem centrado na frequência 222 kHz;

- **Hilbert Transform** (5) [Xil11]: Bloco que implementa uma transformada de Hilbert que inverte 90° a fase do sinal. Este bloco foi implementado para corrigir a inversão de 90° de fase criada pelo filtro passa-alto;

- **Decoder IPcore** (6): Bloco onde é realizada a descodificação FM0 dos sinais provenientes das *tags*;

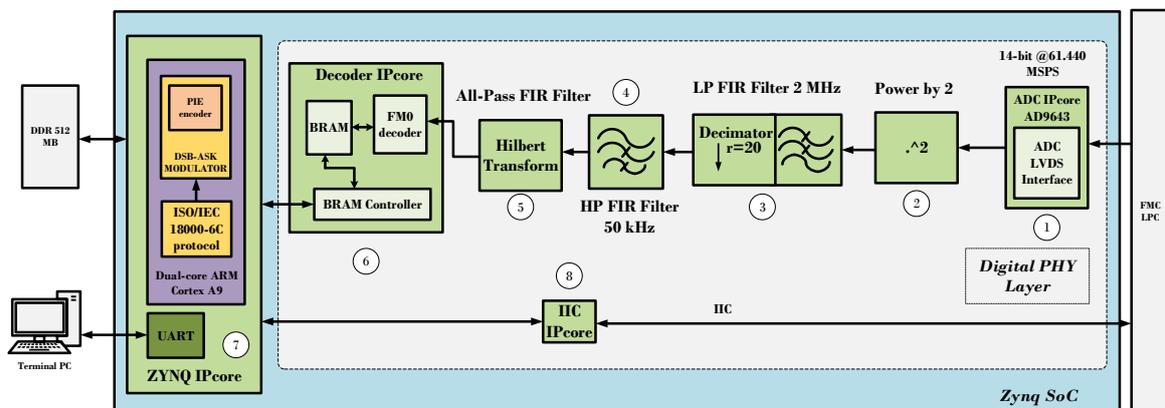


Figura 4.20: Diagrama de blocos cadeia de recepção implementada na FPGA.

Finalizada a descrição da cadeia de recepção, é agora explicado com maior detalhe, os estágios necessários para a descodificar um sinal proveniente da *tag*.

• Validação da Cadeia de Recepção em Matlab

Similarmente à cadeia de transmissão, antes de implementar a cadeia de recepção na FPGA, esta teve que ser validada. Para tal, foi desenvolvida uma rotina em Matlab. Esta rotina teve como parâmetro de entrada, um sinal real proveniente da *tag*, amostrado pela ADC a 61.440 MSPS. Este sinal vem contido numa portadora IF de 13 MHz.

A figura 4.21 apresenta o sinal em IF amostrado pela ADC.

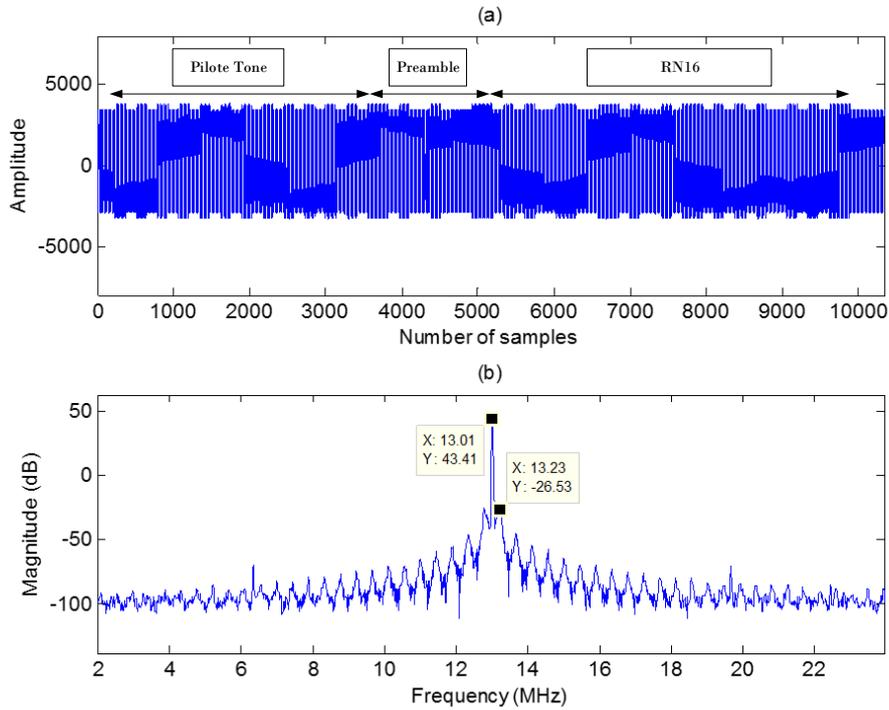


Figura 4.21: (a) Sinal em IF amostrado pela ADC. (b) Sinal em IF amostrado pela ADC no espectro. BLF = 222 kHz e IF = 13.01 MHz

Na figura 4.21 (a) é possível de observar dois níveis de amplitude diferentes. Estes níveis de amplitude, correspondem ao sinal proveniente da *tag*, modulado em ASK. Nesta figura está presente o Pilot tone + Preamble + RN16.

Na figura 4.21 (b) é possível de observar a portadora IF a 13.01 MHz e o sinal da *tag* a 13.23 MHz, ou seja, BLF = 222 kHz.

De seguida, foi necessário isolar a frequência do sinal de interesse. Para tal, foi implementado um detetor da envolvente com recurso a um quadrador (*signal power by 2*) e uma cadeia de filtros FIR a seguir apresentada.

Quadrador (*signal power by 2*)

A primeira fase, para detetar a envolvente de um sinal modulado em ASK, consistiu em elevar o sinal ao quadrado.

Um sinal OOK (caso da modulação ASK com profundidade 100%) pode ser descrito matematicamente pela equação 4.7 [Pra13].

$$S(t) = \begin{cases} A \cos(\omega_p t), & \text{data} = 1 \\ 0, & \text{data} = 0 \end{cases} \quad (4.7)$$

O sinal à saída do quadrador é dado por [Pra13]:

$$S^2(t) = S(t) \times S(t) = \frac{A^2}{2} \times \frac{A^2}{2} \cos(2\omega_p t) \quad (4.8)$$

A figura 4.22 apresenta a simulação em Matlab do sinal após o quadrador.

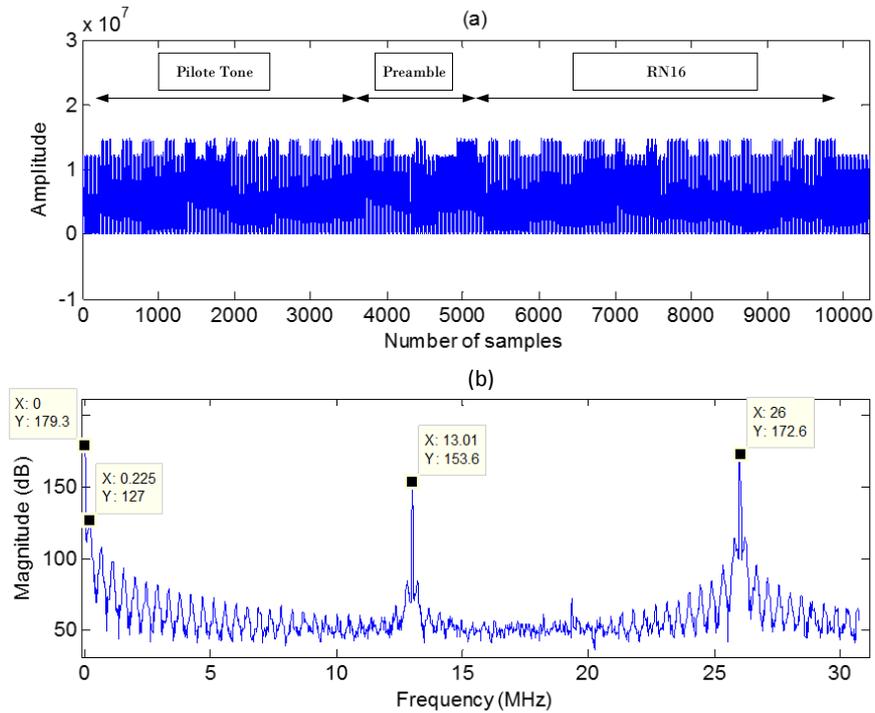


Figura 4.22: (a) Sinal em IF elevado ao quadrado. (b) Sinal em IF elevado ao quadrado no espectro. BLF = 222 kHz e IF = 13.01 MHz.

Pela figura 4.22, comprova-se a criação de uma componente DC (o sinal de interesse encontra-se a 222 kHz desta componente) e a criação de uma frequência ao dobro da frequência IF. Uma vez que a modulação não possui uma profundidade 100%, a quadratura também gera uma componente em IF.

Após o quadrador, foi necessário agora, remover as componentes de alta frequência. Para esse efeito, foi implementado um filtro passa-baixo em que a frequência de corte igual ou superior à largura de banda do sinal de interesse [Pra13].

Remoção das Componentes de Alta frequência e decimação do sinal

Para remover as componentes de alta frequência foi simulado um filtro FIR (*Finite Impulse Response*) passa-baixo (ver Anexo E).

Foi decidido implementar filtros do tipo FIR por serem muito estáveis e de fácil implementação. Estes filtros são facilmente projetados para apresentarem uma fase linear e são adequados para implementar em FPGA. Por fim, os filtros FIR de fase linear, são requeridos quando são apresentadas especificações temporais.

Neste sistema foi implementado um filtro FIR passa-baixo com recurso a uma janela do tipo Retangular. A tabela 4.3 apresenta os parâmetros utilizados para a obtenção dos coeficientes deste filtro.

Low-Pass Filter			
Design Method	Order	Fs (MHz)	Fc (MHz)
Rectangular Window	31	64.440	2

Tabela 4.3: características do filtro FIR passa-baixo.

Este filtro foi projetado em Matlab com recurso à ferramenta *Filter Design & Analysis Tool* e os parâmetros de configuração, apresentados na tabela 4.3, foram obtidos da seguinte forma:

- Os parâmetros “*Design Method*” e “*Order*” foram obtidos com recurso a várias iterações de maneira ao filtro apresentar um comportamento desejado.
- O parâmetro Fs (frequência de amostragem) foi 61.440 MHz devido ao sinal proveniente da *tag* estar amostrado a 61.440 MSPS.
- O parâmetro Fc (frequência de corte) foi 2 MHz. Foi utilizada esta frequência de corte de maneira ao filtro cobrir a largura de banda do sinal de interesse sem o corromper.

A figura 4.23 apresenta o comportamento do filtro FIR passa-baixo desenvolvido.

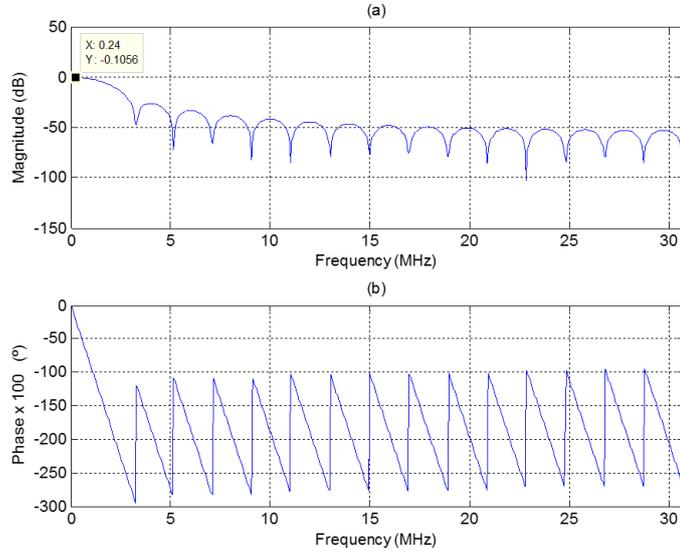


Figura 4.23: (a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência

Depois de aplicado o filtro passa-baixo, o sinal sofreu uma decimação por fator de 20. Foi necessário implementar esta decimação devido ao elevado número de amostras gerado pelo *oversampling* do sinal.

A decimação consiste na redução da frequência de amostragem de um sinal discreto no tempo, $x(m)$, por um fator de L , ou seja, a cada L amostras, é retirado $L-1$ amostras [Vas12].

O processo de decimação é composto por duas etapas:

- A primeira etapa é o processo de anti-*aliasing*, ou seja, passagem por um filtro passa-baixo com frequência de corte $F_s/2L$, onde F_s é a frequência de amostragem original. Este processo foi implementado no filtro FIR passa-baixo;
- A segunda etapa consiste na diminuição do número de amostras.

Matematicamente é possível descrever o processo de decimação como o produto de um sinal original $x(m)$ por um trem de pulsos $p(m)$, como mostra a equação 4.9 [Vas12].

$$x_r(m) = x(m).p(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(k - Lm) = \begin{cases} x(Lm), & m = 0, \pm 1, \pm 2, \dots \\ 0, & \text{outros} \end{cases} \quad (4.9)$$

onde, o trem de pulsos $p(m)$, consiste em:

$$p(m) = \sum_{k=-\infty}^{\infty} \delta(k - Lm) \quad (4.10)$$

A figura 4.24 apresenta um exemplo de um cosseno decimado por 2.

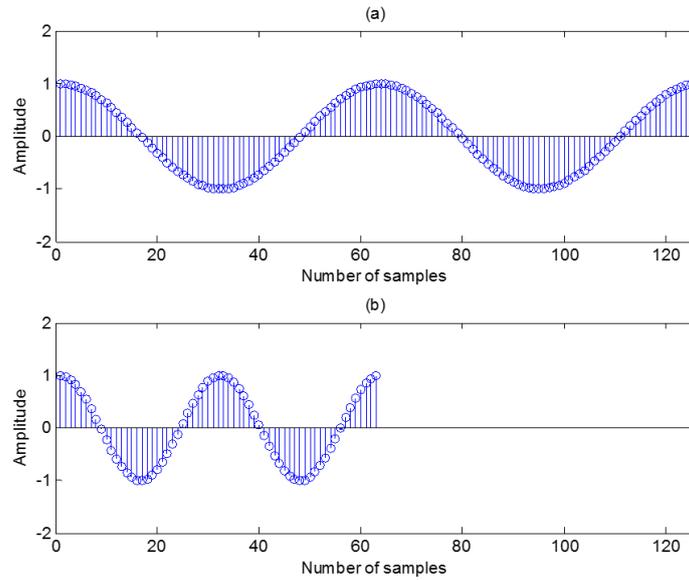


Figura 4.24: (a) Cosseno original com 126 amostras (b) Mesmo cosseno decimado por 2 com 63 amostras.

A figura 4.25 apresenta a simulação em Matlab do filtro do sinal da figura 4.21, após a passagem pelo filtro passa-baixo com decimador de fator 20. É possível observar a atenuação do *phase-noise* assim como a diminuição do número de amostras, contudo continua presente uma componente DC elevada.

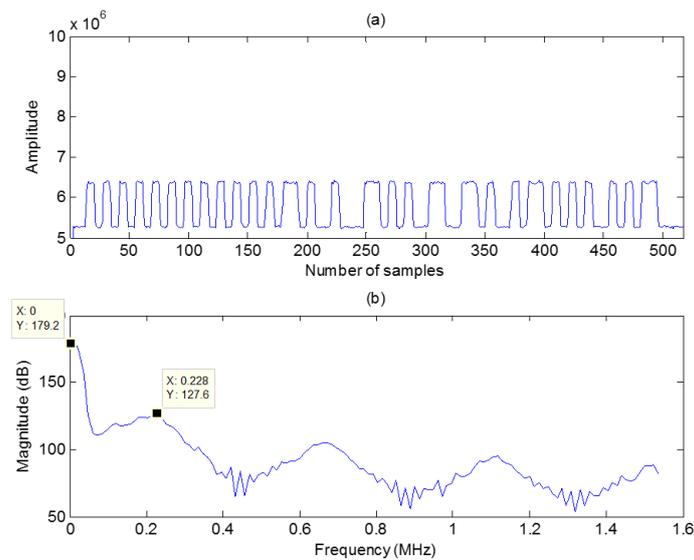


Figura 4.25: (a) Envoltiva do sinal após filtragem e decimação. (b) Envoltiva do sinal após filtragem e decimação no espectro. BLF = 222 kHz.

Concluído este estágio, seguiu-se à remoção da componente DC do sinal.

Remoção da componente DC

Similarmente ao estágio anterior, foi também aplicado um filtro FIR, contudo neste caso, foi implementado um filtro passa-alto. A tabela 4.4 apresenta os parâmetros utilizados para a obtenção dos coeficientes deste filtro

High-Pass Filter				
Design Method	β	Order	Fs (MHz)	Fc (MHz)
Kaiser Window	2.5	21	3.072	0.05

Tabela 4.4: características do filtro FIR passa-alto.

Este filtro foi projetado em Matlab com recurso à ferramenta *Filter Design & Analysis Tool* e os parâmetros de configuração, apresentados na tabela 4.3, foram obtidos da seguinte forma:

- Os parâmetros “*Design Method*”, “ β ” e “*Order*” foram obtidos com recurso a várias iterações de maneira ao filtro apresentar um comportamento desejado.
- O parâmetro Fs (frequência de amostragem) foi 3.072 MHz devido ao sinal proveniente da *tag*, após decimação, apresentar uma nova frequência de amostragem 3.072 MSPS.
- O parâmetro Fc (frequência de corte) foi 50 kHz. Foi utilizada esta frequência uma vez que, o sinal de interesse se encontra centrado a 222 kHz.

Assim sendo, a figura 4.26 apresenta o comportamento do filtro FIR passa-alto desenvolvido.

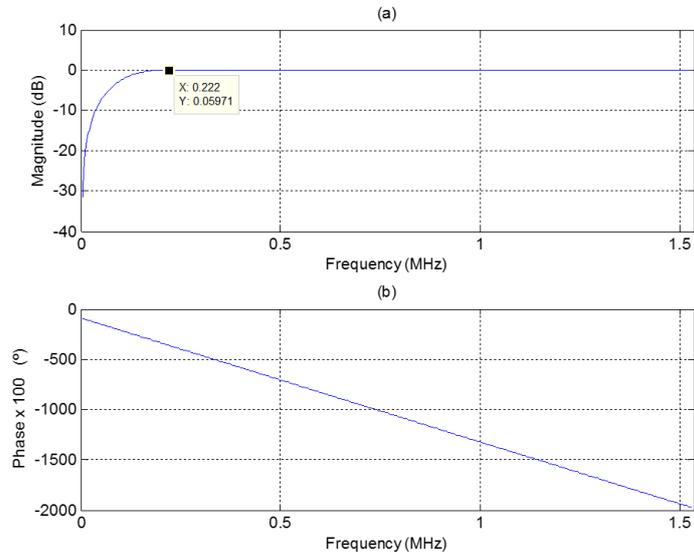


Figura 4.26: (a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência

A figura 4.27 apresenta a simulação em Matlab do sinal RN16 da figura 4.25 após passagem do filtro passa-alto . Pela figura é possível observar a remoção da componente DC do sinal.

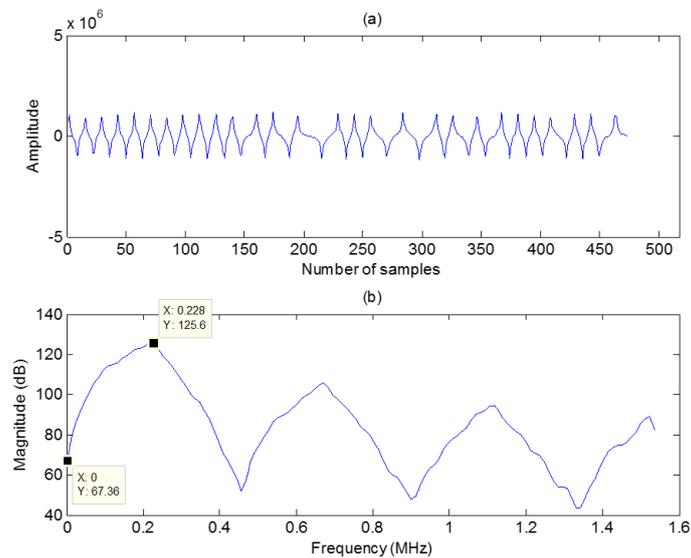


Figura 4.27: (a) Envoltiva do sinal após filtragem e atenuação do DC. (b) Envoltiva do sinal após filtragem e atenuação do DC no espectro. BLF = 222 kHz.

Pela figura 4.27, é possível notar uma deformação da forma do sinal face à forma do sinal de entrada. Este facto é justificado pelo seguinte motivo: a função transferência de

um filtro FIR passa-alto com um número par de coeficientes tem a mesma forma da função transferência de um derivador. Por sua vez, um derivador, onde a resposta em frequência aumenta linearmente com o aumento da frequência e mantém uma rotação de fase constante de 90° [Ste00], ver figura 4.28 (a).

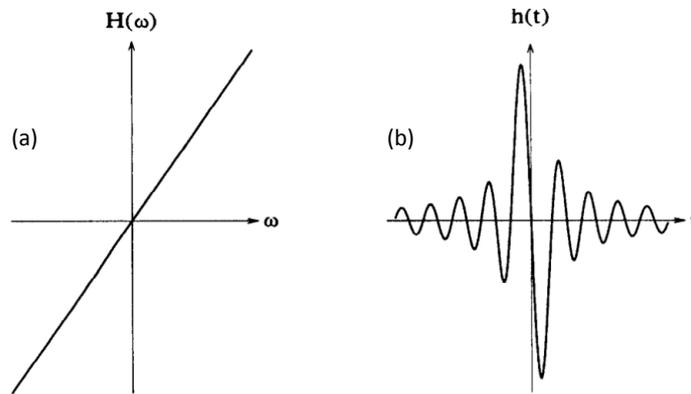


Figura 4.28: (a) Resposta em ordem à frequência de um derivador. (b) Resposta impulsional de um derivador (imagem retirada de [Ste00]).

Contudo um número par de coeficientes não consegue derivar o sinal no instante $t = 0$, derivando só meia amostra antes ou depois, ou seja, apresenta um *group delay* de $\frac{1}{2}$ [Ste00]. Este *group delay* provoca uma resposta impulsional apresentado na figura 4.29 (b).

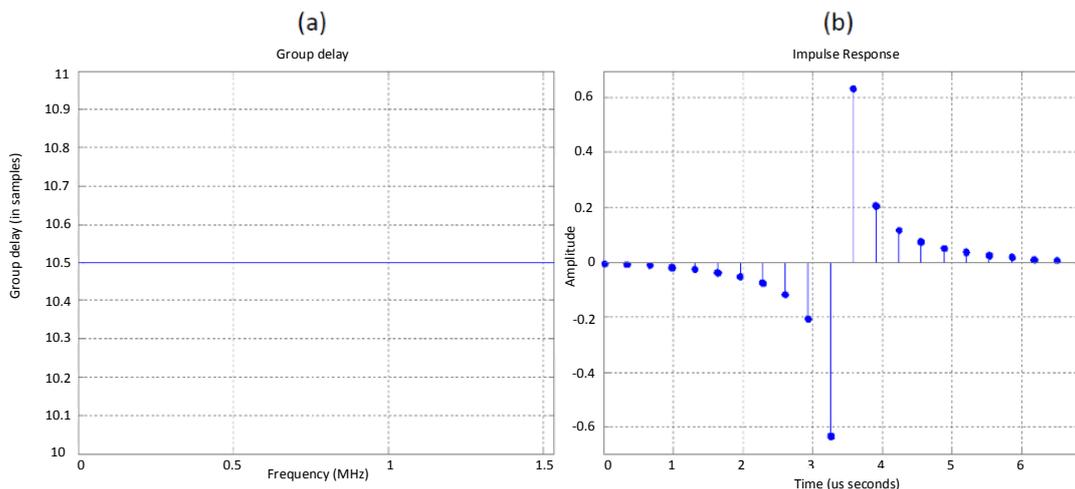


Figura 4.29: (a) *group delay* do filtro FIR passa-alto implementado. (b) Resposta impulsional do mesmo filtro.

Pela figura 4.29 (a) comprova-se o *group delay* de $\frac{1}{2}$ e pela figura 4.29 (b) comprova-se a similaridade da resposta impulsional do filtro FIR passa-alto implementado com resposta

impulsional de um derivador apresentada na figura 4.28 (b).

Por fim, e de maneira a recuperar a envolvente original, foi necessário repor a fase original do sinal. Para tal, foi implementado uma transformada de Hilbert com recurso a um último filtro FIR.

Transformada de Hilbert

A transformada de Hilbert é uma operação matemática que, dado um sinal de entrada, desloca $+90^\circ$ a fase das componentes negativas e -90° a fase das componentes positivas. Esta transformada é facilmente implementada com recurso a um filtro FIR *all-pass* [Ste00].

Este filtro foi construído também com recurso à ferramenta *Filter Design & Analysis Tool*.

Assim sendo, a figura 4.30 apresenta o comportamento do filtro FIR *all-pass* de ordem 30 implementado. A figura 4.31 apresenta a resposta impulsional da transformada de Hilbert.

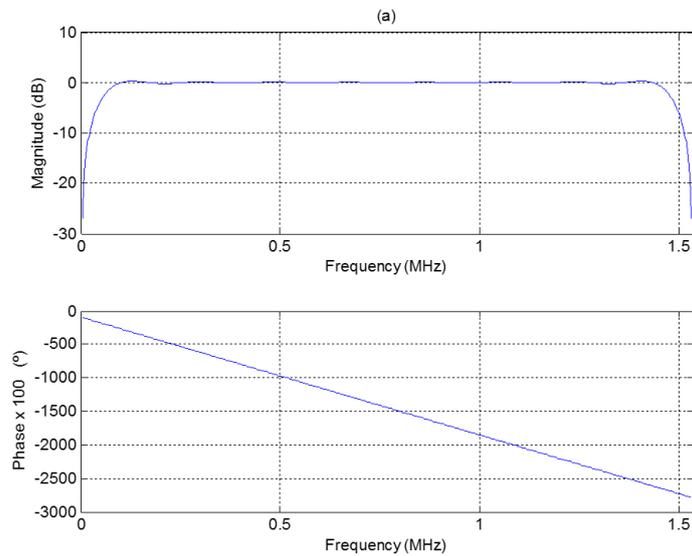


Figura 4.30: (a) Comportamento da magnitude do filtro em função da frequência. (b) Comportamento da fase do filtro em função da frequência.

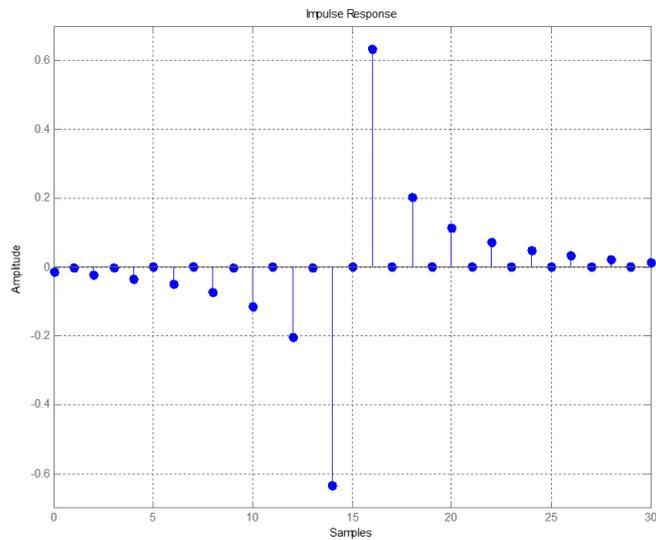


Figura 4.31: Resposta impulsional da transformada de Hilbert.

A figura 4.32 apresenta a simulação em Matlab do sinal RN16 da figura 4.27 após passagem pela transformada de Hilbert. Pela figura é possível observar a recuperação da envolvente original do sinal.

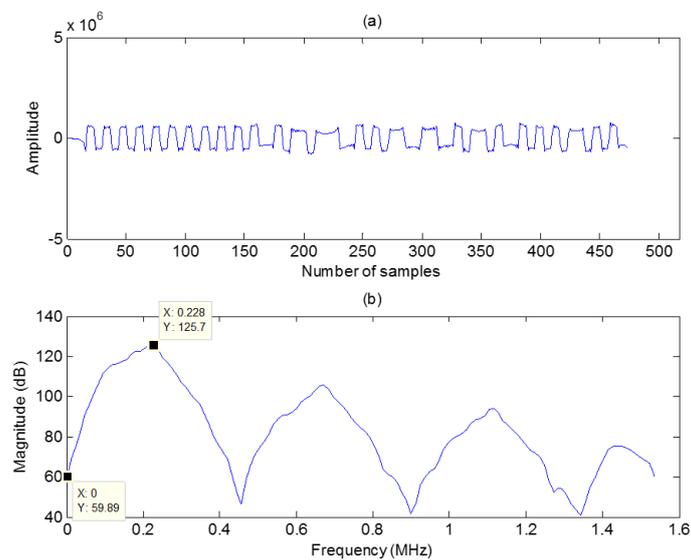


Figura 4.32: (a) Recuperação da envolvente do sinal após filtragem e atenuação do DC. (b) Recuperação da envolvente do sinal após filtragem e atenuação do DC no espectro. BLF = 222 kHz.

Após a passagem pela cadeia de filtros FIR, obteve-se a envolvente do sinal apresentada

na figura 4.27. Por fim, foi necessário decodificar o sinal codificado em FM0. Para tal, foi desenvolvido e validado, também em Matlab, um decodificador para este efeito.

Descodificação FM0

Na cadeia de transmissão, a *tag* foi programada com os parâmetros apresentados na tabela 4.5.

DR	TRcal (μ s)	Data-rate (kHz)	Codificação	Tom Piloto
64/3	96.1	222	FM0	Sim

Tabela 4.5: Parâmetros de programação da *tag*.

Como referido no subcapítulo 2.4.4, a codificação FM0 contém três símbolos de informação: *data-0*, *data-1* e *violation*.

Com a informação presente na tabela 4.5, foi possível calcular por quantas amostras é composto um símbolo de informação (ver equação 4.11).

$$N_{samples} = \frac{F_s}{Data_rate} = \frac{3.072}{0.222} \cong 13.83 \text{ samples} \quad (4.11)$$

Pela variação da frequência durante o *backscattering* (tabela 2.2), foi possível estabelecer um intervalo de amostras, entre o qual a composição de um símbolo pode variar (ver equação 4.12).

$$N_{samples} = \frac{F_s}{Data_rate \pm 2.5\%} = \frac{3.072}{0.222 \pm 0.0056} \cong [13.50; 14.20] \text{ samples} \quad (4.12)$$

Sabendo o número de amostras, o bloco de decodificação FM0 consistiu em dois contadores: um com a função de contar o número de amostras processadas, e outro com a função de contar o número de passagens por zero ocorridas após 13 amostras processadas.

De uma forma simplificada:

- o símbolo *data-0* foi atribuído quando processadas entre 13 a 16 amostras com duas passagens por zero ocorridas;
- o símbolo *data-1* foi atribuído quando processadas entre 13 a 16 amostras com uma passagem por zero ocorrida;
- o símbolo *violation* foi atribuído quando processadas mais de 16 amostras com duas passagens por zero ocorridas;

A figura 4.33 representa a lógica de decodificação FM0 implementada.

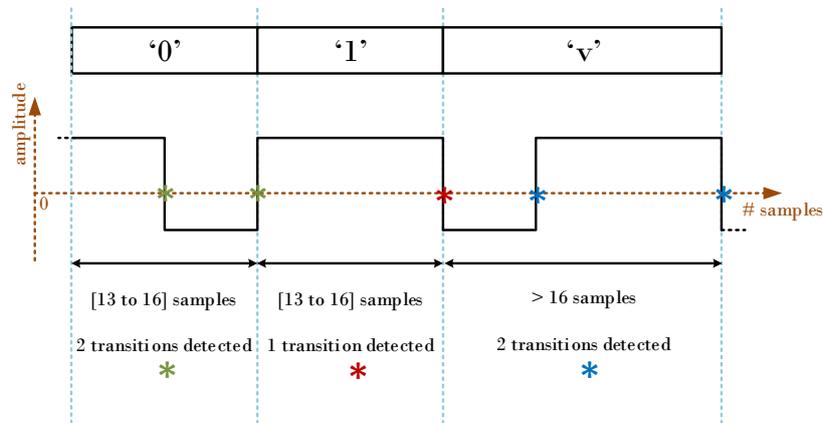


Figura 4.33: Esquema de descodificação FM0 no sistema implementado.

A descodificação foi simulada e validada em Matlab (ver figura 4.34).

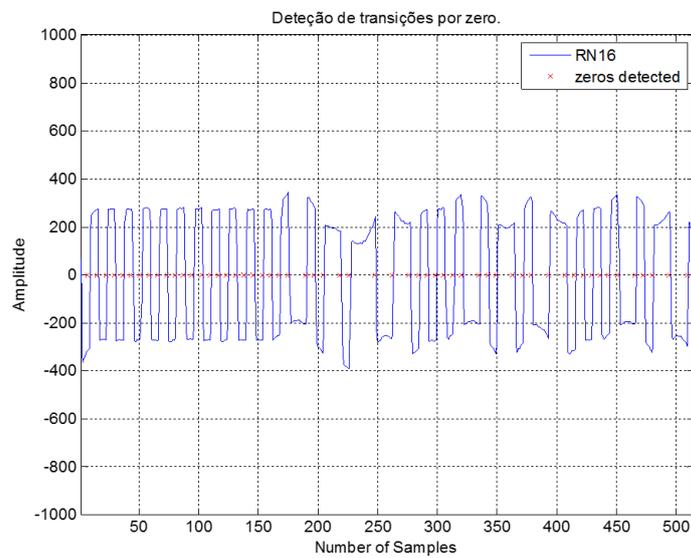


Figura 4.34: Simulação Matlab da detecção de transições por zero.

Este método de descodificação, ou seja, detetar as passagens por zero, foi implementado devido à existência de um desvio de frequência no BLF [Jin12]. Entende-se por desvio de frequência como a diferença entre a frequência real enviada pela *tag* e a frequência nominal (BLF). A figura 4.35 apresenta este fenómeno.

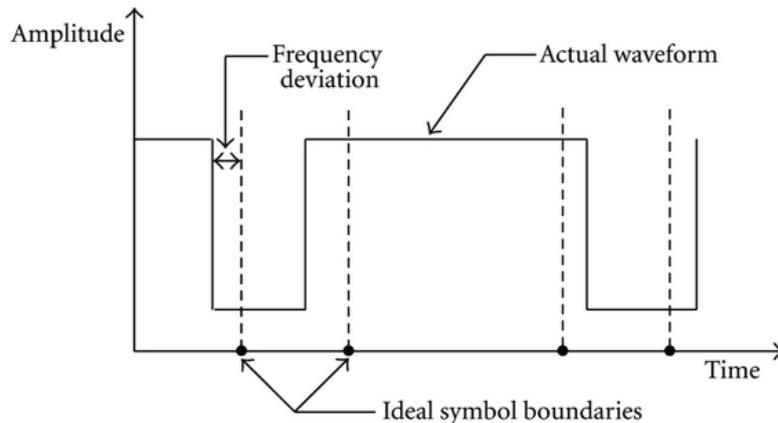


Figura 4.35: Desvio de frequência no sinal *backscattering* enviado pela *tag* (imagem retirada de [Jin12]).

Validada a cadeia de recepção em Matlab, esta foi implementada na FPGA.

• Implementação da Cadeia de Recepção na FPGA

A recepção e decodificação do sinal proveniente da *tag* teve recurso aos blocos (1), (2), (3), (4), (5) e (6) da figura 4.20.

O sinal proveniente da *tag* foi convertido para o domínio digital com recurso ao bloco (1). Como referenciado anteriormente, este bloco implementa a *interface* LVDS entre a FPGA e a DAC. Este foi programado para operar a uma frequência de amostragem de 61.440 MSPS.

O quadrador, bloco (2), foi implementado com recurso a um multiplicador programado para funcionar em complemento para dois. O multiplicador teve as entradas A e B ligadas aos 16-bit menos significativos (sinal *in phase*) da saída do bloco (1). Desta maneira, multiplicando o sinal por ele próprio, foi conseguido implementar um quadrador.

O filtro passa-baixo, bloco (3), foi implementado com recurso ao FIR *Compiler* disponibilizado pela Xilinx. Este filtro foi projetado com os coeficientes obtidos pela simulação em Matlab e configurado para funcionar no modo *decimation*, *decimate rate value* de 20, com uma frequência de amostragem de 61.440 MHz e precisão de 32-bit em complemento para dois.

O filtro passa-alto, bloco (4), foi implementado com recurso ao FIR *Compiler* disponibilizado pela Xilinx. Este filtro foi projetado com os coeficientes obtidos pela simulação em Matlab e configurado para funcionar no modo *single-rate*, com uma frequência de amostragem de 3.072 MHz e precisão de 32-bit em complemento para dois.

A transformada de Hilbert, bloco (5), foi implementado com recurso ao FIR *Compiler* disponibilizado pela Xilinx. Este filtro foi projetado com os coeficientes obtidos pela simulação em Matlab e configurado para funcionar no modo *hilbert*, com uma frequência de amostragem de 3.072 MHz e precisão de 32-bit em complemento para dois.

Por fim, o descodificador foi constituído por três blocos fundamentais (ver figura 4.36):

- O bloco FM0 decoder com RAM *interface* desenvolvido em VHDL;
- Um bloco RAM implementado com recurso ao *Block Memory Generator* disponibilizado pela Xilinx. Foi configurado para funcionar como uma memória do tipo *True Dual Port RAM* no modo *BRAM Controller*;
- Um bloco *AXI BRAM Controller* disponibilizado pela Xilinx.

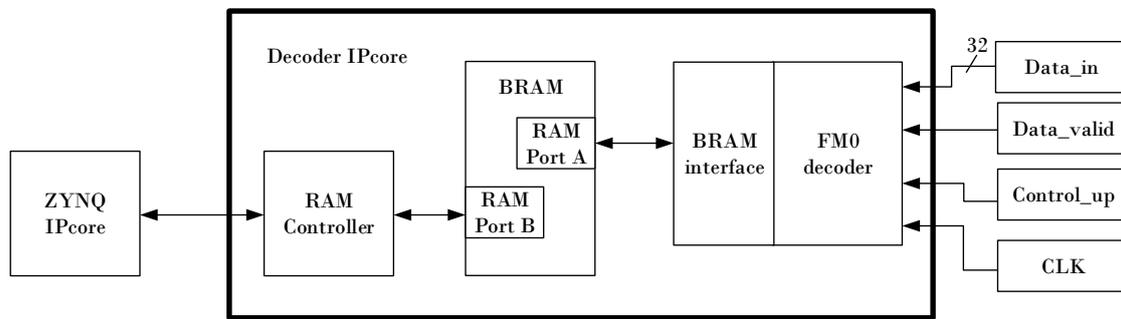


Figura 4.36: Diagrama de blocos do descodificador implementado.

O bloco *FM0 decoder* é controlado por um sinal, *control_up*, proveniente do bloco *ZYNQ IPcore*. Este sinal é assíncrono e indica o tempo que o descodificador se encontra no estado ativo. O descodificador só aceita informação do sinal *Data_in*, a cada transição ascendente do sinal *CLK* e quando os sinais *control_up* e *data_valid* se encontrarem no nível lógico '1'.

Por fim, é gerado um endereço no bloco *RAM interface* e a informação descodificada, é armazenada no bloco *RAM*. Por sua vez, esta informação é acedida pelo processador com recurso ao bloco *RAM Controller*.

4.4 Camada MAC

Nesta secção é apresentada a camada MAC desenvolvida neste trabalho de dissertação de mestrado. Entende-se por camada MAC, todas as funções desenvolvidas em C, responsáveis pela camada protocolar do ISO 18000-6C. Esta camada foi implementada no processador ARM presente no SoC ZYNQ da FPGA utilizada.

O *software* desenvolvido é dividido em duas secções:

- Inicialização dos componentes de *hardware* da camada física. Esta função, de nome *ADFMCOMMS1_Start*, inicializa e configura a frequência de amostragem da DAC e ADC, o ganho do LNA, as frequências dos LOs de transmissão/receção, e por fim, inicializa o bloco

descodificador e o bloco de memória BRAM. As funções *driver* para a comunicação IIC entre o RF *front-end* e a FPGA foram implementadas com recurso ao *reference design* disponibilizado;

- *Main Loop* com uma máquina de estados implementada de maneira a processar o protocolo de comunicação. Esta função, de nome *inventory_round_multi_tag*, implementa a máquina de estados a seguir apresentada na figura 4.37.

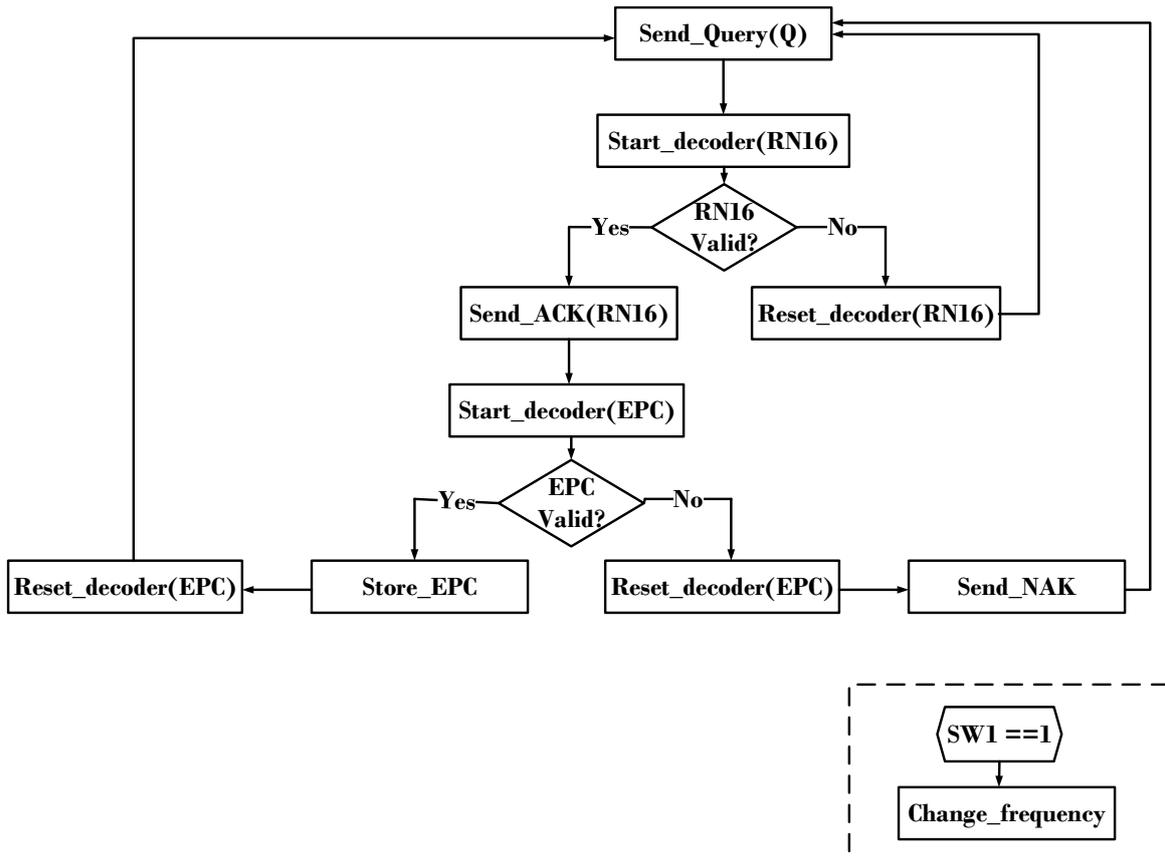


Figura 4.37: Máquina de estados implementada na camada MAC.

No diagrama apresentado na figura 4.37 estão apresentadas as funções principais. Caso o SW1 da placa Zedboard comute para o estado lógico '1', o processador muda para a função *Change_freq*, permitindo assim, alterar a frequência de transmissão em tempo real.

As funções implementadas em C são agora explicadas em maior detalhe:

4.4.1 Funções Desenvolvidas em C

- A função *Send_Query* constrói a trama *query* (ver tabela 2.5) conforme o número de *tags* presentes (2^Q) e envia esta trama para a função *PIE_encoder*. A função predefine os parâmetros $DR = \frac{64}{3}$ e o envio do *pilot tone* por parte da *tag*. Esta função utiliza outras

duas funções: *wait_us* e *Send_preamble*.

- A função *wait_us* tem como objetivo programar os tempos de espera apresentados na tabela 2.10. A função implementa um ciclo *for* com um contador. Este ciclo *for* coloca o processador em “pausa” durante o número de μs estabelecido como parâmetro de entrada;

- As funções *Send_preamble* e *Send_frame_sync*, são as funções responsáveis pelo envio das tramas Preamble e Frame_Sync, apresentadas na figura 2.11;

- As funções *Start_decoder* e *Stop_decoder* são sub-funções da função *decoder_FM0_data*. Estas funções controlam o tempo que o decodificador se encontra ativo, ou seja, controlam o sinal *Control_up* apresentado na figura 4.36. Estas funções têm como parâmetro de entrada uma *flag* que sinaliza se a informação a analisar é uma trama RN16 ou uma trama EPC;

- A função *Reset_decoder* é uma sub-função da função *decoder_FM0_data*, que inicializa as posições de memória do bloco RAM com o valor zero, ou seja, indica ao controlador de memória (*BRAM Controller*) para escrever zeros nas posições de memória deste bloco;

- A função *decoder_FM0_data* tem a finalidade de confirmar se as tramas RN16 e EPC são válidas, ou seja, no caso de uma trama RN16, confirma se o *pilot tone* e o *preamble* correspondem ao predefinido pelo protocolo. No caso do EPC, para além do *pilot tone* e o *preamble*, confirma também o PC. Esta função tem como parâmetro de entrada uma *flag* que sinaliza se a informação a analisar é uma trama RN16 ou uma trama EPC;

- A função *Send_ACK* constrói a trama *ACK* apresentada na tabela 2.9, ou seja, indica ao bloco RAM *Controller* para recolher a informação referente ao RN16 presente no bloco RAM e envia esta informação para a função *PIE_encoder*. Esta função utiliza outras duas funções: *wait_us* e *Send_frame_sync*;

- A função *Send_NAK* constrói a trama *NAK* apresentada na tabela 2.8 e envia esta informação para a função *PIE_encoder*;

- A função *DSB_Modulation* é a função responsável por gerar o sinal modulante em banda base. Esta função contém as sub-funções *data_mux* e *PIE_encoder*;

- A função *data_mux* é uma sub-função da função *DSB_Modulation* responsável pela multiplexagem das palavras 0x7FFF e 0x0B85 (ver figura 4.15), ou seja, esta função envia as palavras para o GPIO implementado na cadeia de transmissão;

- A função *PIE_encoder* é uma sub-função da função *DSB_Modulation* que codifica em PIE o sinal a transmitir. Para tal, esta função recorre às funções *data_mux* e *wait_us*;

- A função *Change_freq* permite a alteração da frequência de transmissão, ou seja, através da combinação dos *switches* SW2 e SW3, presentes na *Zedboard*, altera a frequência dos LOs do *front-end*;

Este capítulo apresentou os aspetos mais relevantes da implementação do sistema RFID

desenvolvido. Deste modo, segue-se a apresentação dos resultados que visam demonstrar o correto funcionamento do sistema.

Capítulo 5

Resultados

Este capítulo tem como objetivo apresentar os resultados obtidos, através de uma série de medições, de modo a comprovar o correto funcionamento do sistema implementado. A maioria das medições¹ efetuadas, foram realizadas num ambiente controlado, mais concretamente, dentro de uma câmara Anecoica. A figura 5.1 apresenta a configuração do *setup* de medições utilizado.

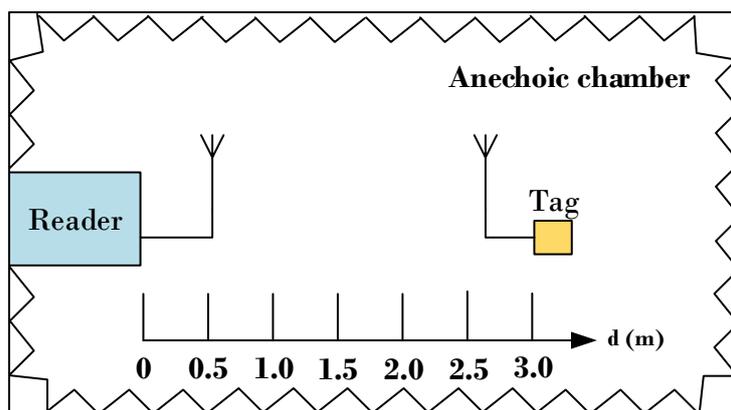


Figura 5.1: Esquema de medição utilizado.

5.1 Introdução

As medidas, seguidamente apresentadas, consistiram em:

- alterar a frequência de operação do leitor de RFID implementado, de maneira a comprovar o bom funcionamento do mesmo, para os quatro canais de leitura;
- medir a máscara de potência transmitida, de modo a comprovar que cumpre as especificações do protocolo de comunicação;
- medir o alcance máximo de leitura do *reader*;

¹a câmara Anecoica foi utilizada para a medição da distância de leitura e a desempenho do sistema.

- medir o desempenho do sistema;
- comprovar o correto envio e recepção dos sinais RFID;
- apresentar os recursos utilizados para a implementação do sistema.

5.2 Alteração da Frequência de Operação

Nesta secção são apresentados os resultados obtidos quando alterada a frequência de operação.

Como referido no sub-capítulo 1.3, a gama de operação para o RFID UHF, estabelecida pela ANACOM, é 865.6-867.6 MHz com espaçamento entre canais adjacentes de 600 kHz e largura de banda 200 kHz (ver figura 5.2).

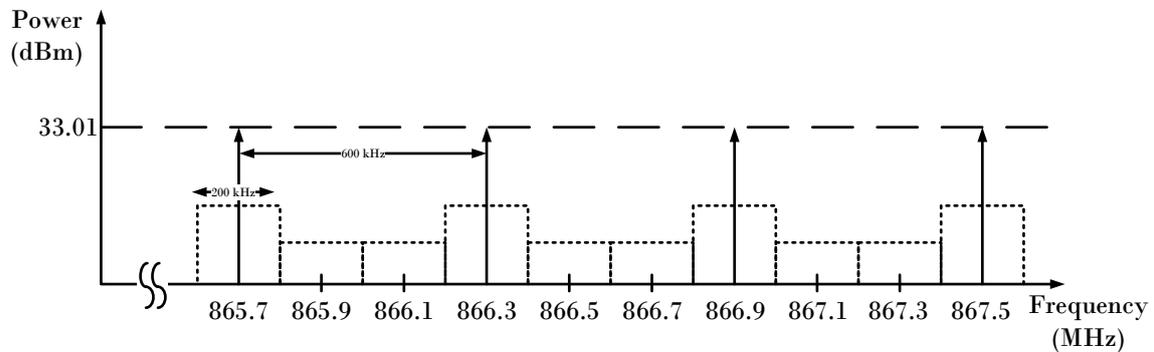


Figura 5.2: Frequências de operação para RFID UHF estabelecido pela ANACOM.

O *reader* implementado disponibiliza quatro frequências possíveis para transmissão, 865.7 MHz, 866.5 MHz, 867.1 MHz e 867.5 MHz. Foram retiradas capturas para cada uma das portadoras, utilizando um analisador de espectros (Rohde & Schwarz FSH13). A figura 5.3 apresenta o esquema utilizado para medições.

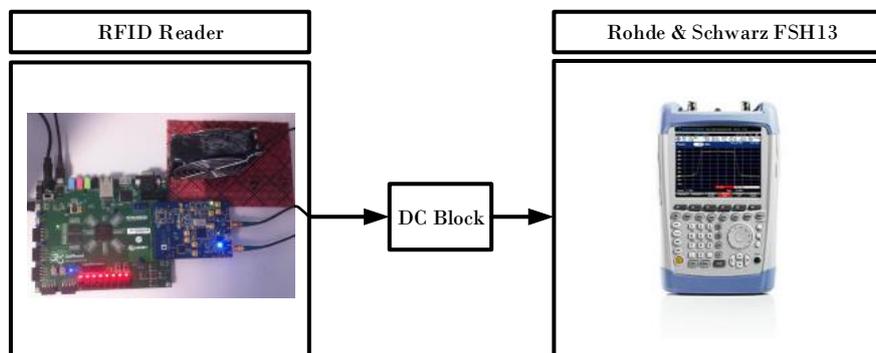


Figura 5.3: *Setup* de medidas utilizado.

Nas figuras 5.4, 5.5, 5.6 e 5.7 encontram-se, representadas no domínio da frequência, os

quatro sinais modulados implementados para transmissão. Os dados retirados destas figuras encontram-se na tabela 5.1. Comprova-se assim a flexibilidade que o sistema apresenta relativamente à seleção da portadora, podendo escolher-se entre 865.7; 866.3; 866.9 ou 867.5 MHz. De salientar que estas medidas foram efetuadas fixando o PW de transmissão do sistema a $10.4 \mu s$, ou seja, $T = 10.4 \mu s \Leftrightarrow f = \frac{1}{T} = 96.15 kHz \Rightarrow LB = 2 \times f = 192.30 kHz$.

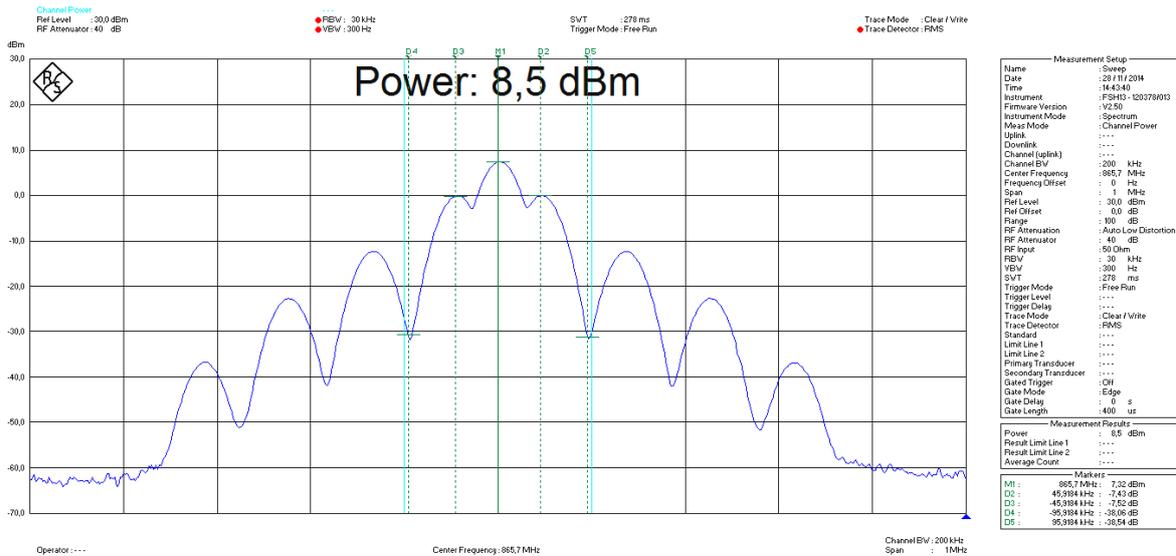


Figura 5.4: Espectro do sinal a 865.7 MHz.



Figura 5.5: Espectro do sinal a 866.3 MHz.

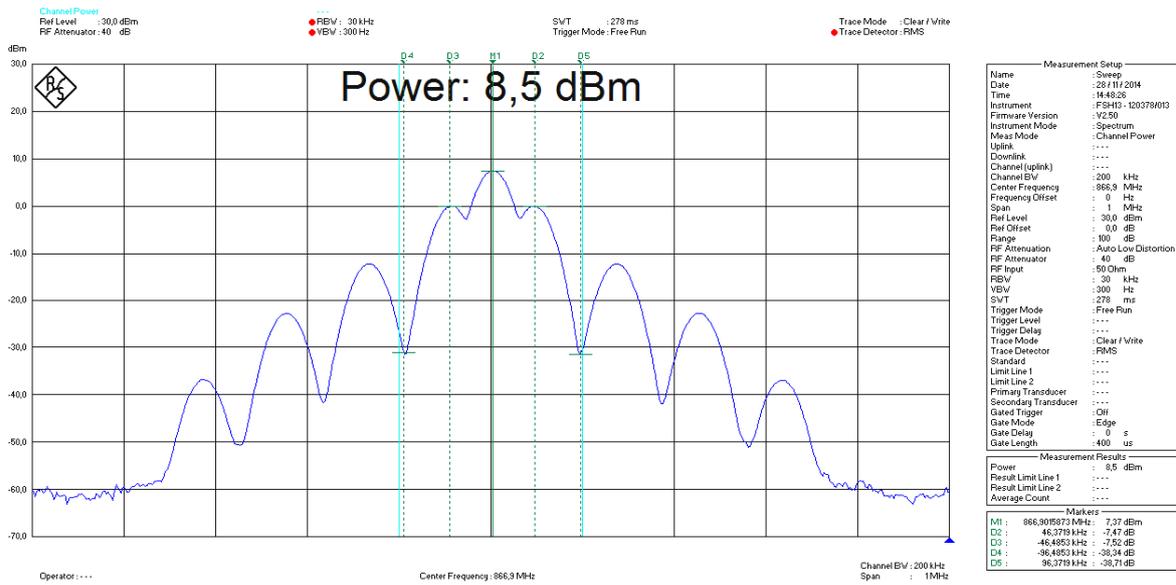


Figura 5.6: Espectro do sinal a 866.9 MHz.



Figura 5.7: Espectro do sinal a 867.5 MHz.

A tabela 5.1 apresentam as características dos vários sinais apresentados anteriormente. Nestas tabelas são apresentados os valores de largura de banda ocupada, valor da frequência central, potência transmitida no canal e os valores de relevância para a modulação DSB-ASK, ou seja, os valores de frequência central $\pm \frac{1}{2T}$ e frequência central $\pm \frac{1}{T}$.

Bandwidth (kHz)	Channel Power (dBm)
191.84	8.5
Central Freq (MHz)	Mgnitude (dBm)
865.7	7.32
Frequency	Δ Frequency (MHz)
865.7+1/2T	45.92
865.7-1/2T	-45.92
865.7+1/T	95.92
865.7-1/T	-95.92

(a)

Bandwidth (kHz)	Channel Power (dBm)
192.86	8.5
Central Freq (MHz)	Mgnitude (dBm)
866.3	7.35
Frequency	Δ Frequency (MHz)
866.3+1/2T	43.31
866.3-1/2T	-43.42
866.3+1/T	96.37
866.3-1/T	-96.49

(b)

Bandwidth (kHz)	Channel Power (dBm)
192.87	8.5
Central Freq (MHz)	Mgnitude (dBm)
866.9	7.37
Frequency	Δ Frequency (MHz)
866.9+1/2T	46.37
866.9-1/2T	-46.49
866.9+1/T	96.37
866.9-1/T	-96.49

(c)

Bandwidth (kHz)	Channel Power (dBm)
192.86	8.5
Central Freq (MHz)	Mgnitude (dBm)
867.5	7.40
Frequency	Δ Frequency (MHz)
867.5+1/2T	46.37
867.5-1/2T	-46.49
867.5+1/T	96.37
867.5-1/T	-96.49

(d)

Tabela 5.1: (a) caraterísticas do sinal a 865.7 MHz. (b) caraterísticas do sinal a 866.3 MHz. (c) caraterísticas do sinal a 866.9 MHz. (d) caraterísticas do sinal a 867.5 MHz.

Comparando o espectro de um sinal DSB-ASK, apresentado na figura D.3, com o espectro apresentado na figura 5.4 e tabela 5.1, concluiu-se que o espectro da modulação SSB-ASK, do sinal implementado, encontra-se de acordo com o esperado.

Para obter uma melhor percepção das várias portadoras implementados, as figuras 5.4, 5.5, 5.6 e 5.7 foram sobrepostas e apresentadas no gráfico da figura 5.8.

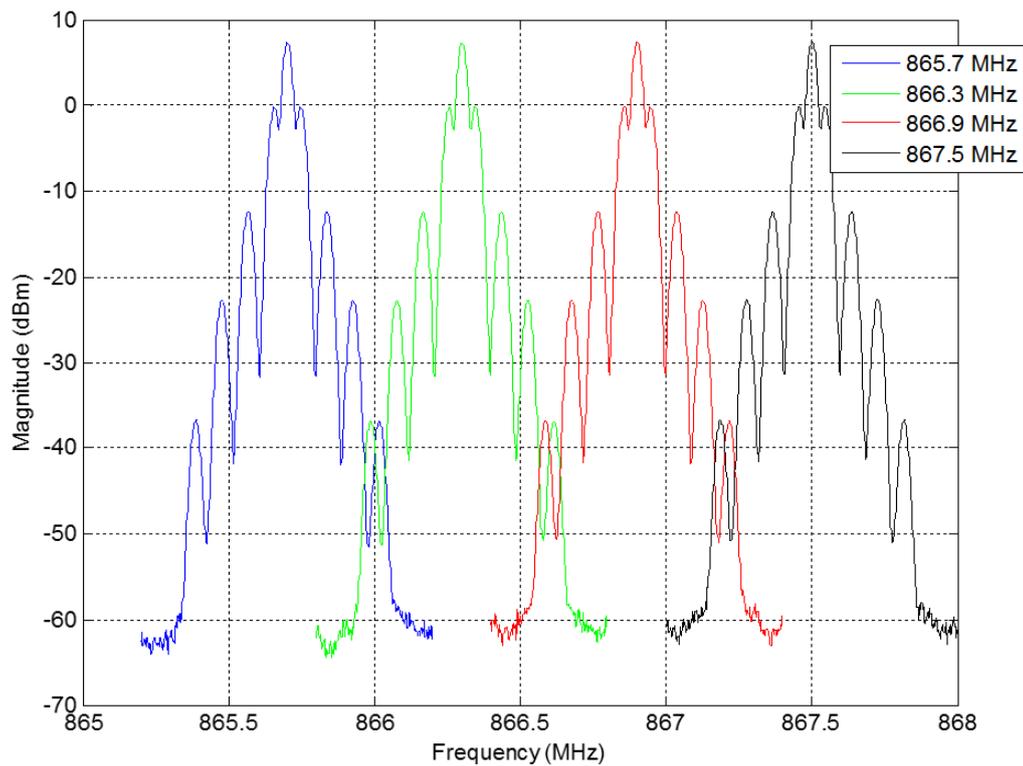


Figura 5.8: Os quatro canais de transmissão implementados no sistema.

Pela figura 5.8, também se comprovou a flexibilidade do sistema implementado. A nível de leitura de *tags*, comprovou-se que a alteração da frequência de operação, desde que esta não esteja fora da gama RFID UHF, não influenciou o comportamento do sistema.

De seguida são apresentados os resultados obtidos face à potência emitida pelo *reader*.

5.3 Máscara de Transmissão de Potência

Nesta secção, foi comparado o espectro do sinal modulado 867.5 MHz transmitido, com a máscara de potência num ambiente *multiple-interrogator*, apresentada anteriormente na figura 2.12. A figura 5.9 apresenta a máscara de transmissão de potência sobreposta ao sinal modulado 867.5 MHz.

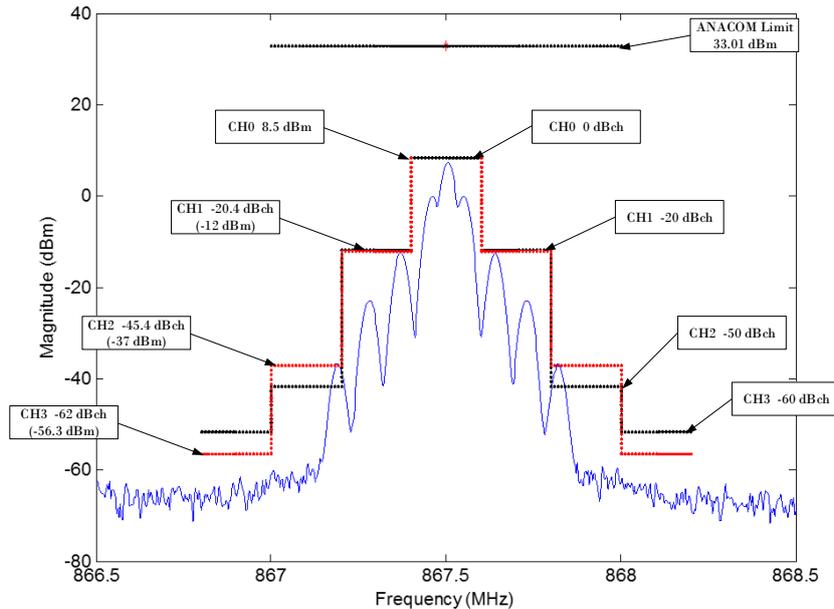


Figura 5.9: Sinal implementado comparado com a máscara de transmissão de potência. Vermelho: máscara de potência do sinal real, medido no analisador de espectros. Preto: máscara de potência de transmissão exigida num ambiente *multiple-interrogator*.

Pela figura 5.9 conclui-se:

- O sinal transmitido encontra-se abaixo do limite máximo de potência transmitida, imposto pela ANACOM (8.5 dBm face aos 33.01 dBm);
- O sinal transmitido encontra-se dentro da largura de banda máxima entre canais adjacentes, imposto pela ANACOM (192.86 kHz face aos 200 kHz);
- O sinal transmitido, em geral, respeita a atenuação imposta pelo protocolo, com a exceção, de que, no canal adjacente número 2, existir 4.6 dBm de excesso, em relação à máscara de transmissão.

5.4 Alcance Máximo de Leitura por parte do *Reader*

Esta secção foi baseada no método de instrumentação apresentada no documento [RV09]. Este método consiste em dois modos para medir o alcance máximo de leitura, ambos são baseadas no cenário, descrito na figura 5.10, onde o *reader* transmite uma determinada potência, P_{reader} , e a *tag* encontra-se a uma distância d , das antenas do mesmo (*reader*).

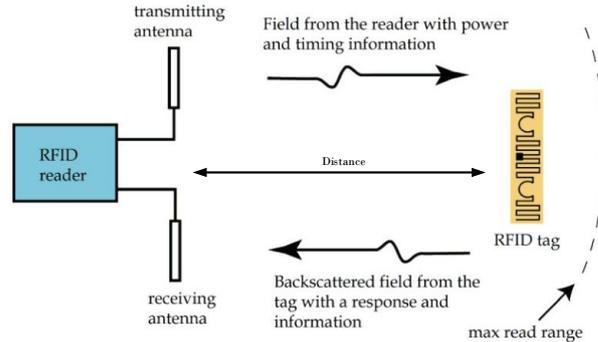


Figura 5.10: Esquema de medição utilizado (imagem adaptada de [Tur10]).

O primeiro método consistiu em fixar a potência enviada pelo *reader* e variar a distância d da *tag*. Por outras palavras, P_{reader} foi constante e d foi variada. Por fim, o segundo método, consistiu em fixar a distância da *tag* e variar a potência do *reader*, ou seja, d foi constante e P_{reader} foi variada.

Nos sistemas RFID UHF passivos, a distância de leitura máxima é limitada pela comunicação *reader-to-tag* do sistema. De modo a calcular esta distância, foram necessários dois passos: obter a sensibilidade e o alcance da *tag*, ou seja, obter a potência mínima e a distância máxima, em espaço livre, necessária para a *tag* responder a um comando proveniente do *reader*.

5.4.1 Método: d constante e P_{reader} variável

Neste método, a medida fundamental, é a potência mínima (P_{min}) necessária por parte da *tag* de maneira a calcular a sensibilidade, P_{tag} . Este cálculo foi efetuado com base na equação de Friis para o espaço livre

$$P_{tag} = P_{min} G_t \left(\frac{\lambda}{4\pi d} \right)^2 \quad (5.1)$$

onde λ corresponde ao comprimento de onda, G_t o ganho da antena de transmissão do *reader* e d a distância da *tag*.

Para variar a potência transmitida por parte do *reader* foi necessário utilizar o atenuador variável, modelo “8494A Attenuator /11 dB” do fabricante *Hewlett-Packard*, entre a saída Tx do *front-end* e a antena de transmissão.

Por fim, de maneira a evitar reflexões, tanto quanto possível, as medidas foram efetuadas na câmara Anecoica, como é apresentado no esquema da figura 5.11.

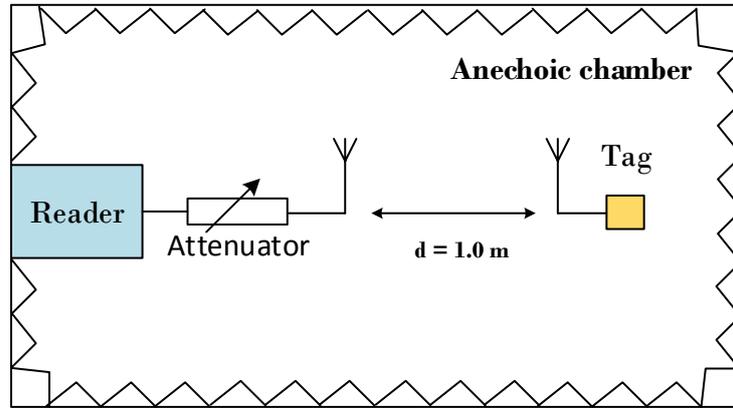


Figura 5.11: Esquema de medição utilizado para obter P_{min} .

Após a medição, foram obtidos os resultados apresentados na tabela 5.2.

5.4.2 Método: P_{reader} constante e d variável.

Neste método, a medida fundamental, é a distância máxima de comunicação por parte da *tag* (d_{max}), de modo a calcular o alcance máximo r_{tag} . Este cálculo foi igualmente efetuado com base na equação de Friis para o espaço livre

$$P_{tag} = \text{EIRP} \left(\frac{\lambda}{4\pi r_{tag}} \right)^2 \quad (5.2)$$

onde EIRP é a potência isotrópica equivalente radiada por parte do *reader* e P_{tag} é a sensibilidade apresentada na tabela 5.2.

A sensibilidade e o alcance máximo podem se relacionar através das equações 5.1 e 5.2

$$r_{tag} = d \sqrt{\frac{\text{EIRP}}{P_{min} G_t}} \quad (5.3)$$

Por fim, e igualmente, de maneira a evitar reflexões, tanto quanto possível, as medidas foram efetuadas na câmara Anecoica, como é apresentado no esquema da figura 5.12.

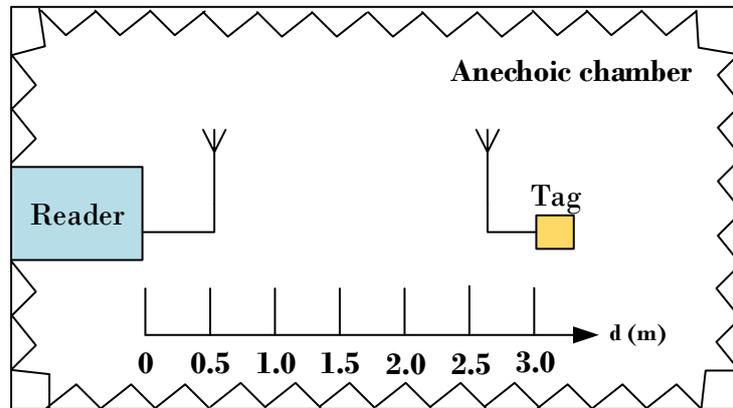


Figura 5.12: Esquema de medição utilizado para obter d_{max} .

Após a medição, foram obtidos os resultados apresentados na tabela 5.3.

5.4.3 Resultados Obtidos

Pelas tabelas 5.2 e 5.3 conclui-se que a potência mínima necessária transmitir à *tag* foi cerca de -15.07 dBm (sensibilidade da *tag*) e a distância máxima de leitura obtido pelo sistema foi de 2.0 metros. Pelos dados anteriores, conclui-se, simultaneamente, que o alcance máximo por parte da *tag* foi de aproximadamente de 4 metros. A diferença entre a distância máxima de leitura e o alcance máximo é justificado devido à perda de potência, presente nos cabos (L_c - *Cable Losses*), que não foram contabilizadas no cálculo da EIRP.

$$EIRP = P_{reader} - L_c + G_t \quad (5.4)$$

Frequency (MHz)	d (m)	Gt (dBi)	Preader (dBm)	Pmin (dBm)	Ptag (dBm)
866.3	1.0	5.3	8.5	2.4	-15.07

Tabela 5.2: Resultados obtidos para a medição d constante e P_{reader} variável.

Frequency (MHz)	d max (m)	Gt (dBi)	EIRP (dBm)	Pmin (dBm)	r tag (m)
866.3	2	5.3	13.7	2.4	3.99

Tabela 5.3: Resultados obtidos para a medição P_{reader} constante e d variável.

5.5 Desempenho do Sistema.

Esta secção tem como objetivo apresentar a taxa de erros ocorridos num ambiente *single-tag* (uma *tag* para leitura) e num ambiente *multi-tag* (três *tags* para leitura). Para ambos os ambientes foi contabilizado o número de EPCs, válidos, nulos e inválidos em relação ao número de *ACK*s enviados por parte do *reader*. Foi considerado como resposta nula ou inválida, respetivamente, um EPC não recebido (*missed*) ou um EPC recebido com erros/colisões.

De maneira a obter os resultados pretendidos, no ambiente *single-tag* foi utilizado o esquema de medições, anteriormente apresentado, da figura 5.12, e para o ambiente *multi-tag* foi utilizado o esquema de medições da figura 5.13.

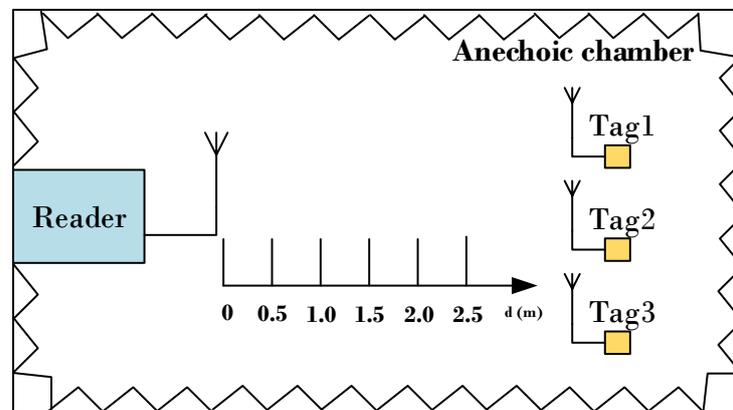


Figura 5.13: Esquema de medição utilizado no ambiente *multi-tag*.

O *reader* também teve que ser instrumentado, ou seja, na máquina de estados, implementada na camada MAC, foram acrescentados contadores. Estes contadores tiveram a finalidade de contar o número de *ACK*s enviados, o número de EPCs válidos e o número de EPCs não válidos. O SW6 também foi programado para, sempre que tomasse valor lógico '1', colocava em pause a máquina de estados e era apresentado, no terminal do computador, os dados referentes aos contadores implementados.

5.5.1 Ambiente *Single-tag*

No ambiente *single-tag*, o *reader* foi programado com um valor de 'Q'=0 e frequência de operação 866.3 MHz. Foram obtidos os resultados apresentados na tabela 5.4 e na figura 5.15.

Measurement time (s)	Frequency (MHz)	Distance (m)	# EPCs	# ACKs	# NAKs	# Missed	ERROR (%)
30 ± 1	866.3	0.50	6558	8745	2182	5	25.01
		1.00	6252	8699	2444	3	28.13
		1.50	5891	8783	2879	13	32.92
		2.00	4387	8605	4201	17	49.02

Tabela 5.4: Performance do sistema num ambiente *single-tag*.

Pela análise da tabela 5.4, conclui-se que a eficiência do sistema diminui com o aumento da distância de leitura, ou seja, com o aumento da distância, aumenta também o número de EPCs inválidos provenientes da *tag*. Este facto é justificado pela secção 5.4, onde com o aumento da distância, menor potência é recolhida pela *tag*.

Por fim, concluiu-se que, para a potência de saída do sistema implementado, a zona ideal para leitura por parte do *reader* situou-se entre os 0.5 e 1.5 metros.

De seguida, são expostos os resultados para o ambiente *multi-tag*. Nesta medição foram usadas as *tags* da figura 5.14.



Figura 5.14: *Tags* utilizadas nas medições. 1-Smartrac; 2-IDSolutions; 3-Miller Engineering.

5.5.2 Ambiente *Multi-tag*

No ambiente *multi-tag*, o *reader* foi programado com um valor de 'Q'=2 e frequência de operação 866.3 MHz. Foram obtidos os resultados apresentados na tabela 5.5 e na figura 5.15.

Measurement time (s)	Frequency (MHz)	Distance (m)	Tag	# EPCs	# Total EPCs	# ACKs	# NAKs	# Missed	ERROR (%)
30 ± 1	866.3	0.5	Miller Engineering	1334	3812	5444	1623	9	29.98
			IDSolutions	953					
			Smartrac	1525					
		1.0	Miller Engineering	1189	3396	5398	1991	11	37.08
			IDSolutions	848					
			Smartrac	1359					
		1.5	Miller Engineering	952	2721	5439	2697	21	49.97
			IDSolutions	681					
			Smartrac	1088					
		2.0	Miller Engineering	551	1575	5322	3689	58	70.40
			IDSolutions	394					
			Smartrac	630					

Tabela 5.5: Número de leituras obtidas para cada *tag* num ambiente *multi-tag*.

Similarmente ao ambiente *single-tag*, o aumento da distância diminuiu o desempenho do sistema. Contudo, devido à existência múltiplas colisões, o desempenho é menor, visto que, o algoritmo de anti-colisão descarta todos os EPCs que sofreram colisão.

Concluiu-se também, que zona ideal para leitura, por parte do *reader*, situou-se entre os 0.5 e 1.5 metros.

5.5.3 Comparação Ambiente *Single-Tag* com *Multi-tag*

A figura 5.15 compara o desempenho do *reader* para estes dois tipos de ambientes.

Observou-se que o sistema possuiu uma maior desempenho num ambiente *single-tag*. Este facto era espectável devido à existência de colisões num ambiente *multi-tag*.

Em termos percentuais, num ambiente *single-tag*, a percentagem de erro variou entre os 25% e 49%. No ambiente com múltiplas *tags*, percentagem de erro variou entre os 30% e 70%.

Por fim, é novamente salientado, que a percentagem de erros consistiu no número de EPCs inválidos em comparação com o número de *ACKs*, ou seja, noutra perspetiva, o *reader* a 2.0

metros de distância, num ambiente *multi-tag*, possuiu 70% de erros, o que correspondeu a 1575 EPCs em 30 segundos.

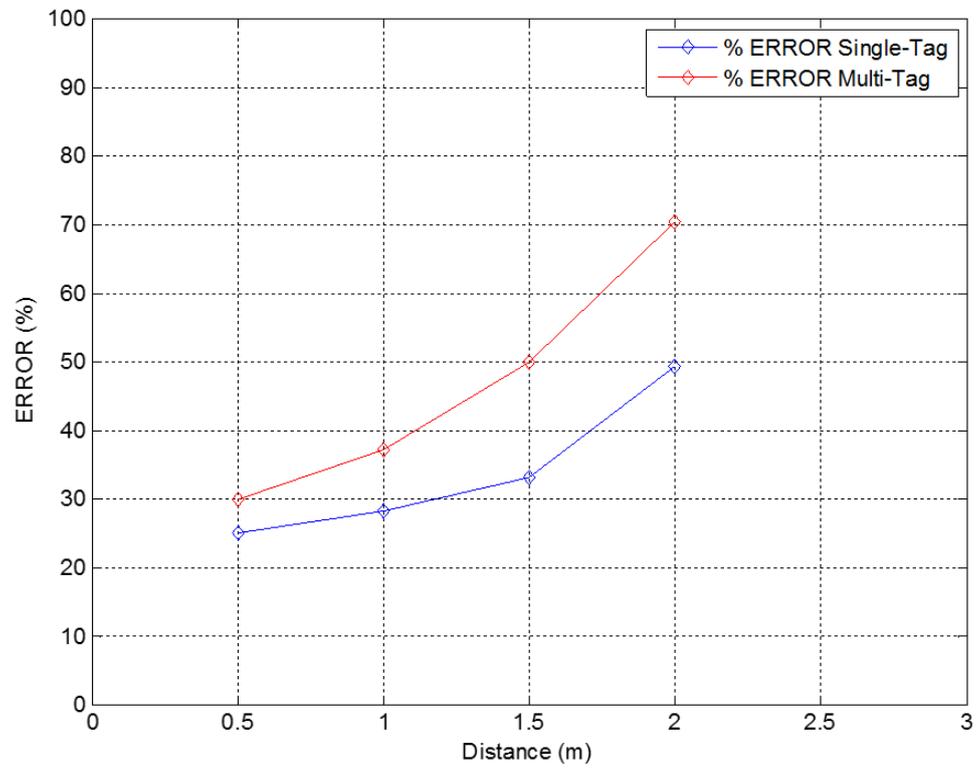


Figura 5.15: Comparação do desempenho do *reader* entre ambientes *multi-tag* e *single-tag*.

5.6 Sinais RFID Obtidos

É agora apresentado, nesta secção, as formas de onda dos sinais RFID. Os resultados foram obtidos com recurso à ferramenta *Hardware Manager*, presente no *software* VIVADO 2014.2, e posteriormente tratados em Matlab.

A figura 5.16 apresenta a comunicação entre o *reader* e a *tag*. Nesta figura, é possível observar o comando *Query* e a resposta RN16, assim como, o comando *ACK* e a resposta EPC da *tag*.

O sinais, apresentados na figura 5.16, foram obtidos na saída do decimador, após passagem pelo filtro FIR passa-baixo implementado na cadeia de receção (ver figura 4.20).

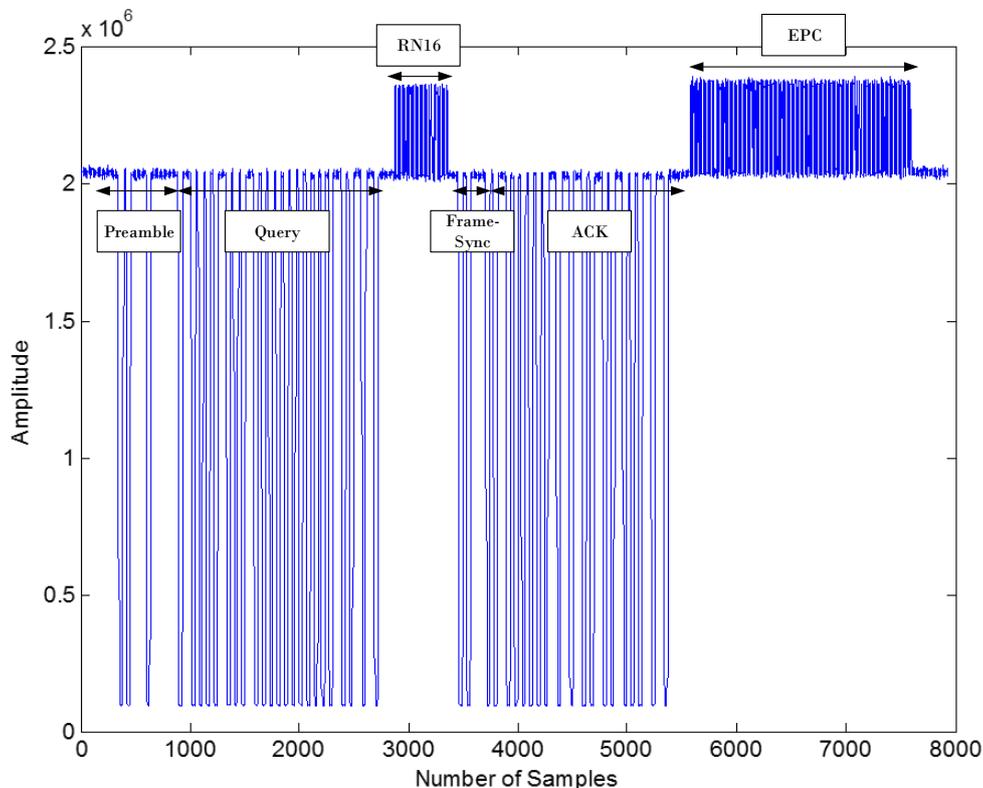


Figura 5.16: Comunicação entre *reader* e *tag*.

De seguida, nas figuras 5.17 e 5.18, são apresentados respetivamente, os sinais RN16 e EPC, recebidos pelo *reader*. Estes dois sinais, foram obtido após a transformada de Hilbert implementada na cadeia de receção (ver figura 4.20).

Por observação das imagens referidas em cima, concluiu-se que a cadeia de filtros FIR foi bem projetada e possuiu um correto funcionamento. O correto funcionamento do protocolo de comunicação, também foi comprovado, pelas imagens referidas.

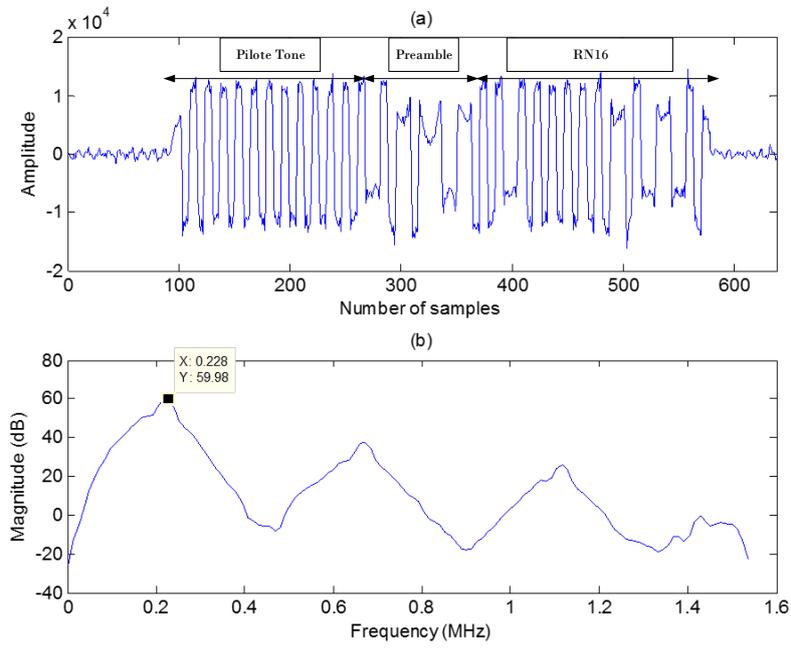


Figura 5.17: Sinal RN16 após transformada de Hilbert implementada na cadeia de recepção.

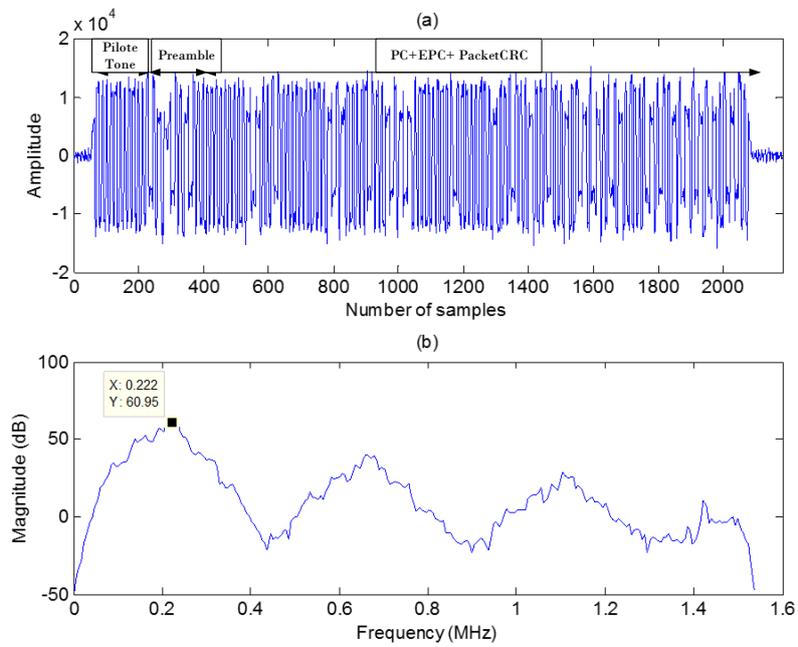


Figura 5.18: Sinal EPC após transformada de Hilbert implementada na cadeia de recepção.

5.7 Recursos de Implementação em FPGA

Nesta secção são apresentados os dados referentes à utilização de recursos da FPGA. Esta informação foi obtida através de um relatório gerado automaticamente pela ferramenta de desenvolvimento VIVADO 2014.2 da Xilinx após implementação.

O consumo de recursos da FPGA é apresentado graficamente na figura 5.19. e mais detalhadamente na tabela 5.6.

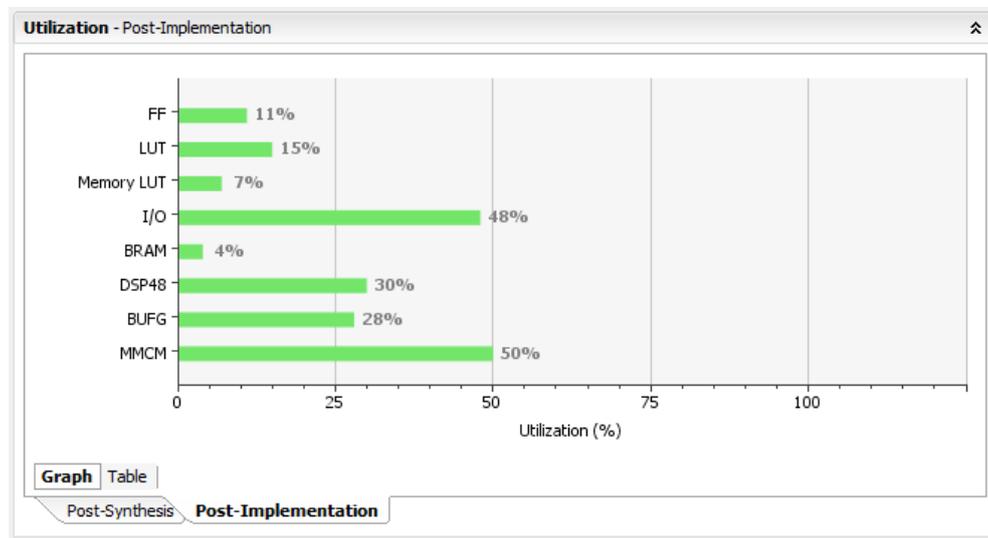


Figura 5.19: Percentagem de recursos da FPGA utilizados pelo sistema implementado.

Visto que a cadeia de receção apresenta uma maior complexidade, grande percentagem dos recursos foram utilizados nesta cadeia. Exemplos destes recursos são os *flip-flops* (FFs) e *global clock buffer* (BUFG) utilizados no bloco de descodificação, os *DSPSlices* (DSP48) utilizados na cadeia de filtros FIR e os BRAMs para guardar a informação proveniente do descodificador.

No sistema implementado, também foram utilizados outros recursos, tais como, *look-up tables* (LUT) para executar operações lógicas, portos físicos *Input/Output* (I/O) e *Mixed-Mode Clock Managers* (MMCM) para gerar relógios, como por exemplo o relógio de referência de 30 MHz e o relógio do IPcore da DAC.

Resource	Utilization	Available	Utilization %
FF	12041	106400	11.32
LUT	8182	53200	15.38
Memory LUT	1186	17400	6.82
I/O	95	200	47.50
BRAM	5	140	3.57
DSP48	65	220	29.55
BUFG	9	32	28.12
MMCM	2	4	50.00

Tabela 5.6: Recursos da FPGA utilizados pelo sistema implementado.

Pela figura 5.19 foi concluído, numa maneira geral, que menos de metade dos recursos da FPGA foram utilizados. Este facto torna evidente a elevada capacidade de recursos disponíveis por parte desta tecnologia, o que justifica o aumento da utilização de FPGAs na implementação de sistemas de rádio.

5.8 Estimativa do Consumo Energético

A ferramenta de desenvolvimento VIVADO 2014.2 da Xilinx, também permite gerar um relatório, onde são apresentadas estimativas do consumo energético da parte digital. Pela figura 5.20 foi estimado que o sistema implementado na FPGA tem um consumo estimado total de 2.638 W.

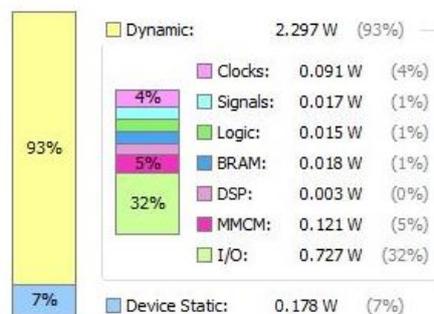


Figura 5.20: Consumo energético “on-chip” do sistema implementado na FPGA.

5.9 Estimativa do Custo do Sistema Implementado

Nesta secção é apresentada uma estimativa do custo monetário do sistema implementado. Para esta estimativa foram contabilizados os componentes principais ao sistema e o custo da PCB. Os preços unitários, apresentados na tabela 5.7, foram obtidos simulando a compra entre 100 a 500 unidades por componente.

A tabela 5.7 apresenta o custo unitário de cada componente, assim como, o custo total do sistema implementado.

Source	Component	Model	Price (USD)
Analog Devices	DAC	AD9122	40.58
Analog Devices	Modulator	ADL5375	5.56
Analog Devices	PA	ADL5602	1.75
Analog Devices	Clock & Sync Gen.	AD9548	26.57
Analog Devices	Clock Dist.	AD9523-1	10.20
Analog Devices	LO	ADF4351	8.25 x 2
Analog Devices	ADC	AD9643	85.28
Analog Devices	LNA	AD8366	6.57
Analog Devices	Demodulaotr	ADL5380	5.28
Digikey	FPGA	XC7Z020-1CLG484C	119.04
Future Electronics	USB Programming	FT232HQ-TRAY	2.95
Digikey	USB/UART Brige Controller	CY7C64225-28PVXC	1.73
TOTAL			322.01 (258.92 EUR)

PCB (EuroCircuits)	
Width (mm)	160.00
Height (mm)	70.00
Thickness (mm)	1.55
N. Layers	8
Price (EUR)	24.32

TOTAL COST	
Price (EUR)	283,24

Tabela 5.7: Estimativa do custo monetário do sistema implementado.

Pela tabela 5.7 observou-se que o sistema implementado tem um custo monetário estimado de 263,29 Euros.

Terminada a exposição dos resultados obtidos nesta dissertação de mestrado, segue o capítulo com as conclusões.

Capítulo 6

Conclusão e Trabalho Futuro

Este capítulo aborda dois tópicos: a conclusão e possíveis direções de trabalho futuro. No primeiro tópico é exposto um resumo geral do sistema implementado, assim como as suas limitações. Por fim, no trabalho futuro, são realçadas as possíveis melhorias ao sistema e novos caminhos de investigação associados ao mesmo.

6.1 Conclusão

Esta dissertação de mestrado teve como objetivo a implementação de um leitor RFID para a gama UHF, com abordagem *Software-Defined Radio*.

O trabalho iniciou-se com uma abordagem geral à tecnologia RFID, mais propriamente ao protocolo ISO 18000-6C para comunicações RFID UHF.

Seguidamente, foi abordada a temática do *Software-Defined Radio*. Neste capítulo foram expostas as arquiteturas mais comuns para implementar um sistema de rádio em SDR, assim como, um breve discussão sobre as melhores abordagens SDR aplicadas a um leitor RFID.

Após discutidos os conhecimentos sobre RFID e SDR, o sistema foi implementado. Nesta implementação, foi utilizada a placa de desenvolvimento RF Front-End AD-FMCOMMS1-EBZ e a placa de desenvolvimento *Zedboard*.

Antes de implementar a cadeia de transmissão e de receção no sistema, estas necessitaram de ser validadas, para tal, foram desenvolvidas rotinas em Matlab.

Validadas as cadeias de transmissão/receção, estas foram implementadas na FPGA.

De maneira a comprovar o correto funcionamento do sistema, foram efetuadas medidas. Estas medidas consistiram em alterar a frequência de transmissão, medir a potência transmitida à *tag*, assim como, calcular o alcance máximo de leitura e o desempenho do sistema.

Relativamente à alteração da frequência de transmissão, comprovou-se que o sistema implementado possui versatilidade, ou seja, com recurso a três *switches* o sistema altera a frequência entre 865.7, 866.3, 866.9 e 867.5 MHz.

Em relação à potencia transmitida, o *reader* cumpre todos os requisitos impostos pela ANACOM, ou seja, apresenta uma potência transmitida de 8.5 dBm, largura de banda de aproximadamente 190 kHz e um espaçamento entre canais de 600 kHz.

Através das medições na câmara Anecoica, obteve-se o alcance máximo de leitura por parte deste. O *reader* apresentou um alcance máximo de dois metros.

Também na câmara Anecoica, foi estimado o desempenho do sistema em ambientes *single-tag* e *multi-tag*. Comprovou-se assim que *reader* apresenta uma zona ótima de leitura com a

tag entre 0.5 e 1.5 metros.

Por fim, foi feita uma estimativa do custo monetário do sistema implementado. Contabilizando só os componentes principais ao sistema e a PCB, este possuiu um custo estimado de 263,29 Euros. Este custo de implementação, torna o leitor RFID muito competitivo ao nível do mercado, uma vez que, os leitores RFID comerciais, para a gama UHF, apresentam um custo entre 500 USD a 2000 USD (402,39 EUR a 1609,58 EUR) [Jou14].

6.2 Trabalho Futuro

De maneira a tornar este descodificador num produto comercial, as máscaras de transmissão de potência para um ambiente *dense-interrogator* teriam de ser cumpridas. Numa implementação futura, também seria necessário aumentar a distância de leitura, para tal, duas soluções são propostas: mudar o tipo de modulação, de modo a aproveitar com maior eficiência a potência transmitida no canal de comunicação; acrescentar um *power amplifier* à saída do *front-end* utilizado. Em relação à escrita no banco de memória é necessário implementar, na máquina de estados, os estágios correspondentes à escrita. Por fim, o leitor necessita ser mais dinâmico, ou seja, o descodificador implementado, deverá incluir um sinal de seleção que, mediante o sinal *Query* enviado, o sinaliza-se da codificação proveniente por parte da *tag*.

Em relação a novos caminhos de investigação, este leitor poderá ser o ponto de partida para uma arquitetura *all-digital* com digitalização em RF, recorrendo a uma FPGA e ADC/DACs de maior desempenho [Dob07].

Anexos

Anexo A

Diagrama de Blocos do Sistema Final

Neste Anexo é apresentado, na figura A.1, o diagrama de blocos completo do leitor de RFID implementado.

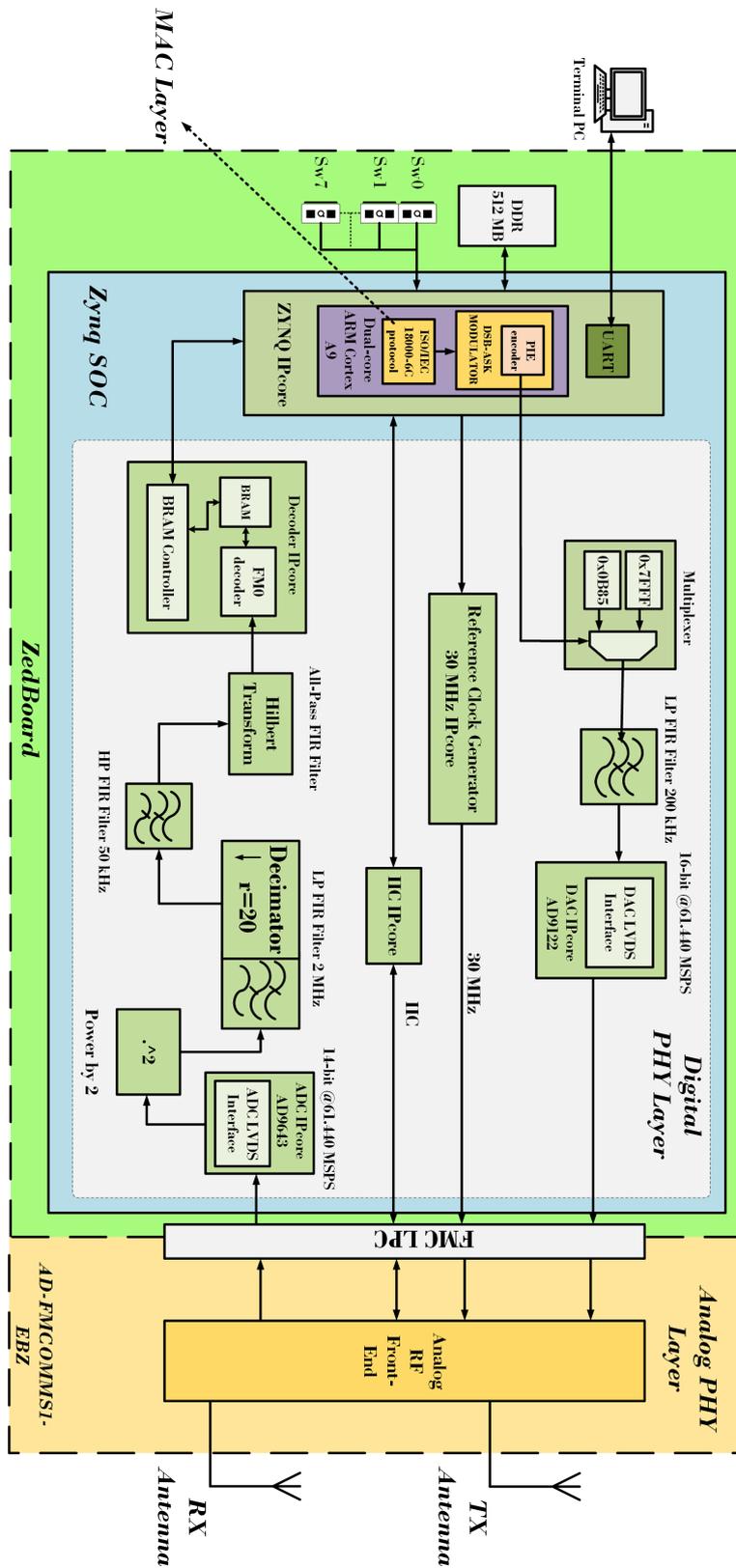


Figura A.1: Diagrama de Blocos do Sistema Implementado.

Anexo B

Sistema Final Implementado

Neste Anexo é apresentado três fotografias do sistema final implementado.

A figura B.1 apresenta uma imagem geral do sistema, enquanto que as figuras B.2 e B.3 apresentam num plano mais aproximado os elementos do sistema.

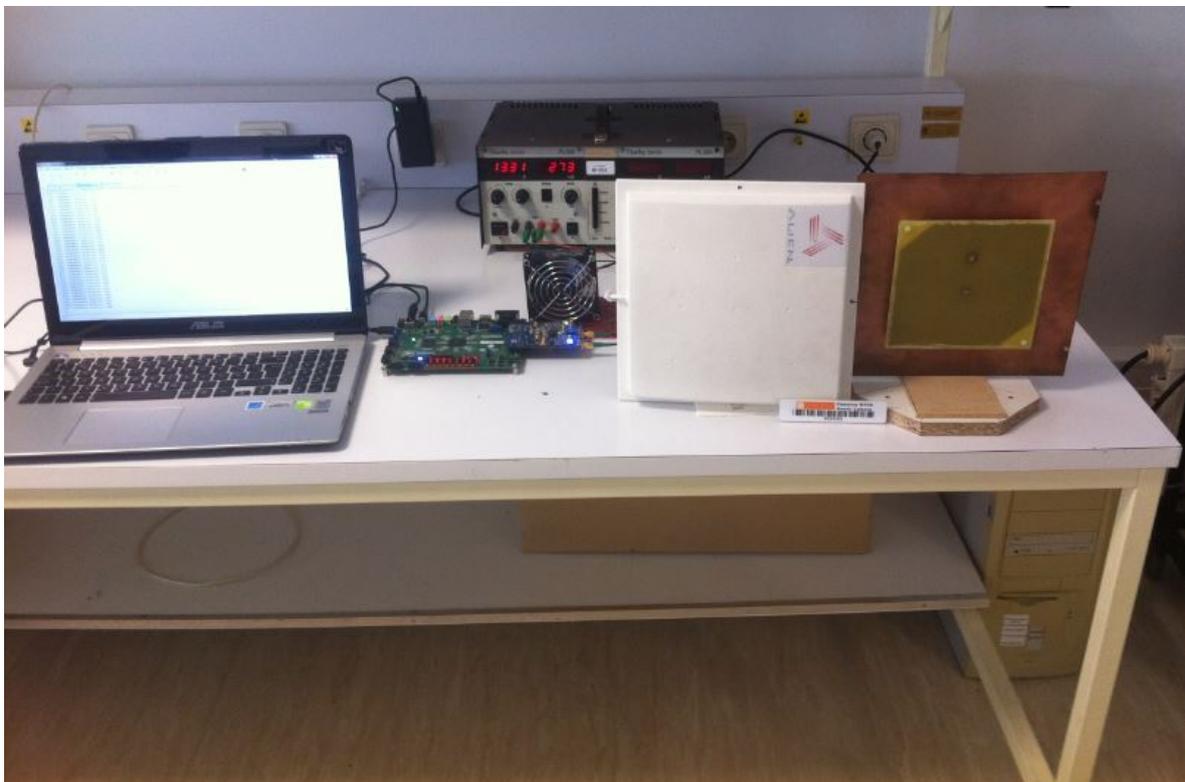


Figura B.1: Sistema Implementado.

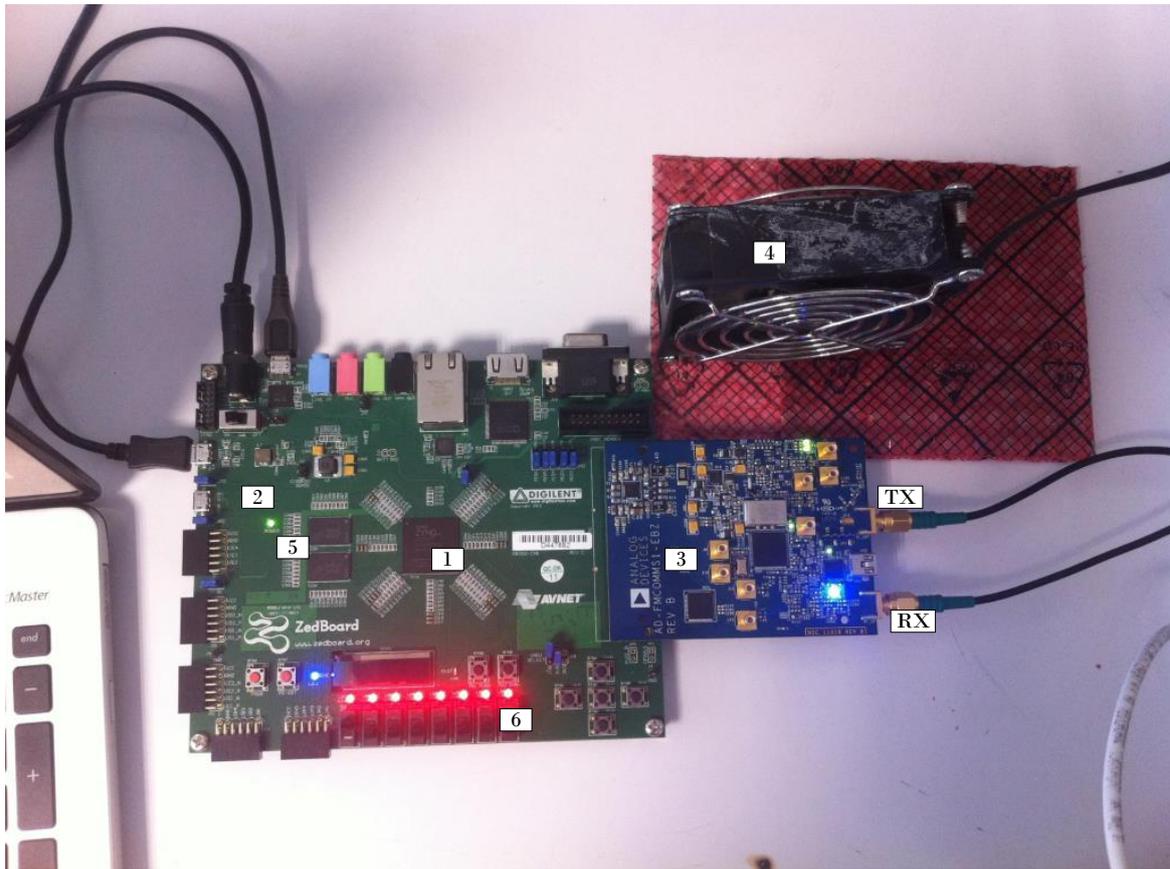


Figura B.2: Detalhe Zedboard mais Front-End do Sistema Implementado. 1)FPGA; 2)Zed-board; 3)AD-FMCOMMS1-EBZ; 4) *Fan*; 5)DDR *memory*; 6)Switches.



Figura B.3: Detalhe antenas de transmissão e recepção do Sistema Implementado. 1)Antena RX; 2)Antena TX; 3)Tag.

Anexo C

Teorema de Nyquist

Neste Anexo é apresentado o Teorema de Nyquist.

O processo de amostragem, como mostra a figura C.2, pode ser matematicamente descrito como a multiplicação de um sinal contínuo no domínio do tempo, $x(t)$, por um trem de impulsos periódicos $\delta(t)$, de período $T = \frac{1}{f_s}$ [Rib11].

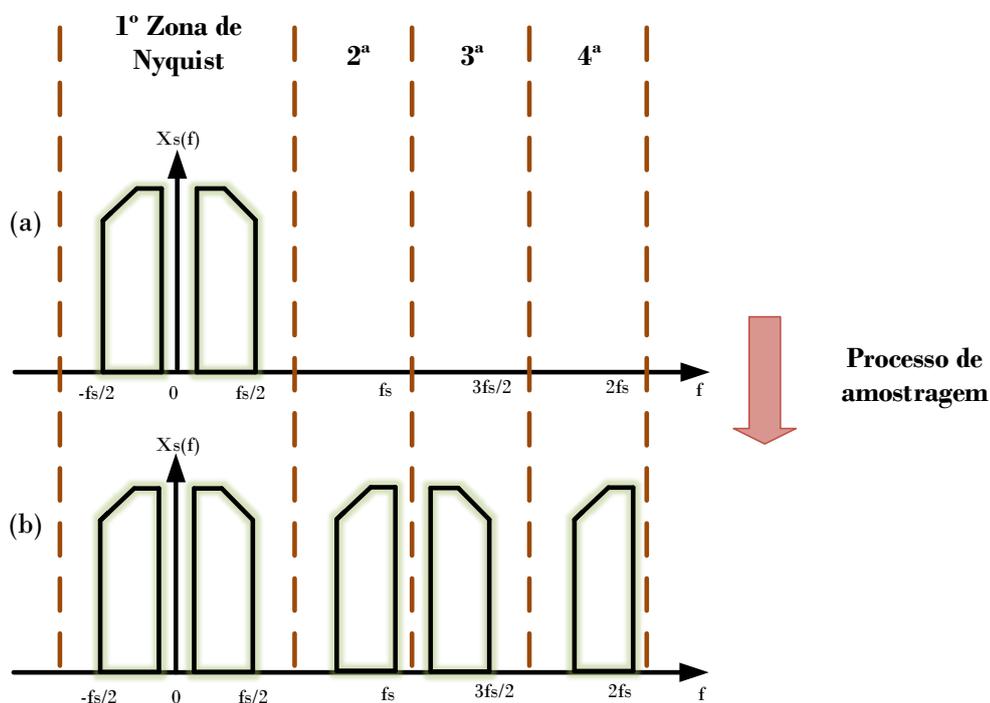


Figura C.1: Processo de amostragem. (a) Sinal cujo conteúdo espectral se encontra na primeira zona de Nyquist. (b) Sinal original e surgimento de réplicas espaçadas de f_s , provocadas pela amostragem.

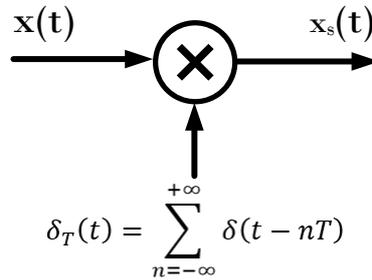


Figura C.2: Processo de amostragem no domínio do tempo.

Desta operação surge o sinal amostrado, designado por $x_s(t)$. Equação C.1.

$$\delta_t(t) = \sum_{n=-\infty}^{+\infty} x(t)\delta(t - nT) \quad (\text{C.1})$$

No domínio da frequência, $x_s(t)$ pode ser representado recorrendo à transformada de Fourier, equação C.2. O sinal resultante é uma sequência de réplicas espaçadas de f_s .

$$X_s(t) = f_s \sum_{k=-\infty}^{+\infty} X(f - kf_s) \quad (\text{C.2})$$

Os sinais ao serem amostrados, sofrem espalhamento pelas diferentes zonas do espectro. As estas zonas dá-se o nome de zonas de Nyquist e encontram-se separadas por $f_s/2$ (ver figura C.1). Em cada zona, o seu conteúdo espectral, será uma versão espelhada do conteúdo presente nas zonas contíguas.

O teorema diz que a frequência de amostragem mínima para efetuar a recuperação completa de um sinal deverá ser $f_s = 2f_{max}$, onde f_{max} é a frequência máxima contida no sinal a amostrar, ou seja, o sinal a amostrar deverá ser limitado em largura de banda por $f_s/2$. Nos casos em que estas condições não são respeitadas, o sinal sofrerá *aliasing*, isto é, haverá sobreposições e o sinal original ficará corrompido [Pra13].

Anexo D

Modulação ASK

Neste Anexo é apresentado a modulação ASK.

Como referido no capítulo 2.2, o protocolo de comunicação ISO 18000-6C, contém suporte para três modulações distintas:

- DSB-ASK.
- SSB-ASK.
- PR-ASK.

Dada uma portadora sinusoidal $A_p \cos(\omega_p t + \theta)$, a informação pode ser transmitida por variação da amplitude A_p , frequência ω_p ou fase θ .

Na modulação de amplitude designada por ASK, no caso particular do sinal modulante ser digital, a portadora modulada $x_p(t)$ é descrita matematicamente pela seguinte forma: [SC110]

$$x_p(t) = A_p m_t \cos(\omega_p t) \quad (\text{D.1})$$

onde m_t , é a envolvente do sinal modulante. Conforme o sinal envolvente, é possível criar variantes da modulação ASK.

A figura D.1 apresenta um exemplo das modulações DSB/SSB-ASK e PR-ASK.

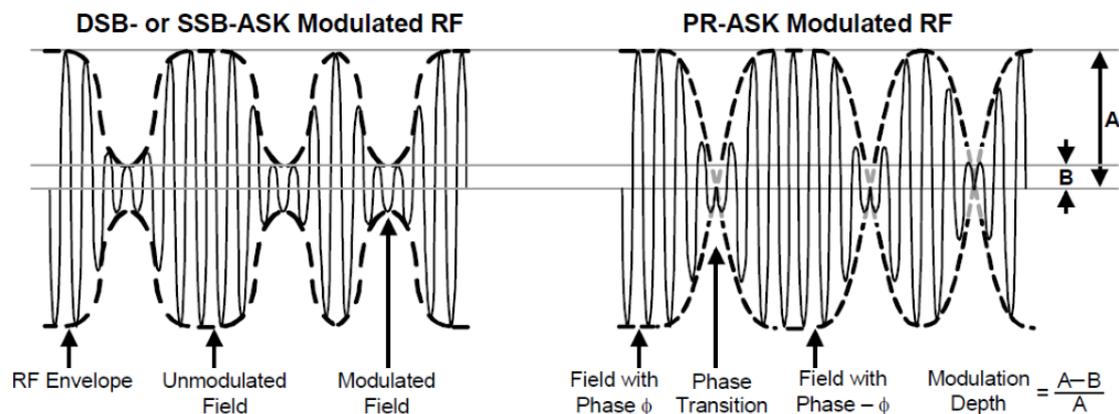


Figura D.1: Variantes da modulação ASK (imagem retirada de [GS113a]).

Pela figura D.1 é possível também retirar a profundidade da modulação (h). A profundidade é dada pela equação D.2.

$$h = \frac{A - B}{A} \quad (\text{D.2})$$

onde A e B são, respetivamente, os valores de máximos do sinal e maior amplitude e menor amplitude.

A figura D.2 apresenta a modulação ASK com três tipos de profundidade diferentes.

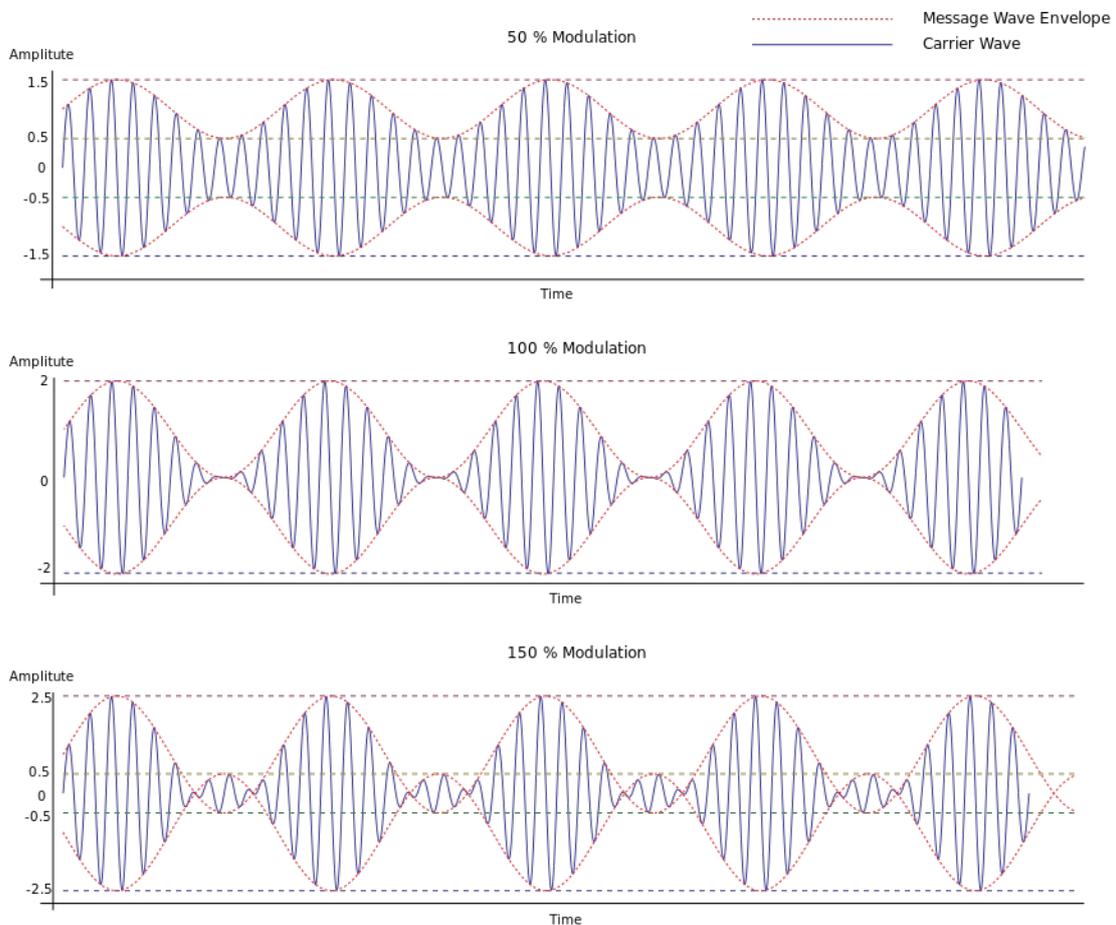


Figura D.2: Modulação ASK com três tipos de profundidade diferentes (imagem retirada de [Wik12]).

A modulação DSB-ASK ocupa maior largura de banda e dispende energia na transmissão da portadora, ou seja gera três componentes espectrais, uma à frequência da portadora e as outras duas a $f_p + 1/2T$ e $f_p - 1/2T$. T corresponde à duração do sinal de maior amplitude. Contudo, é a mais utilizada nos sistemas de rádio devido à sua simplicidade de implementação [SC110]. Em termos espectrais, a modulação DSB-ASK está apresentadas na figura D.3.

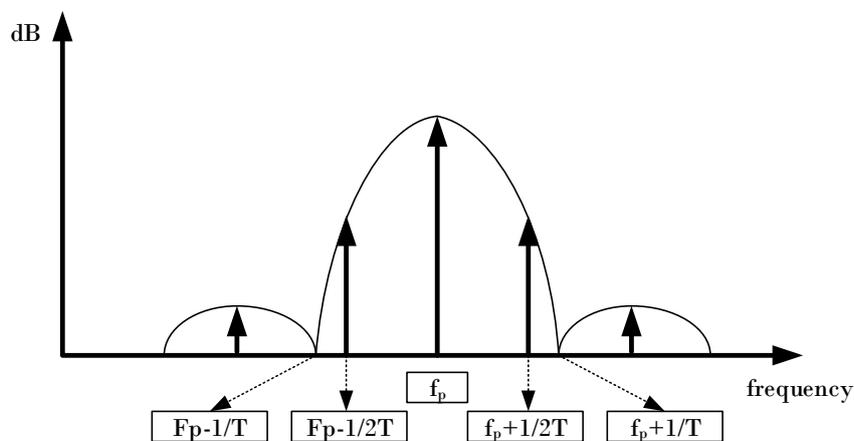


Figura D.3: Espectro da modulação DSB-ASK (imagem adaptada de [GS113a] e de [SC110]).

A modulação SSB-ASK ocupa metade da largura de banda e não dispende energia na transmissão da portadora, ou seja, gera uma só componente espectral centrada em $f_p + 1/2T$ ou $f_p - 1/2T$. Contudo, a sua implementação é mais complexa [SC110]. Em termos espectrais, a modulação SSB-ASK está apresentadas na figura D.4.

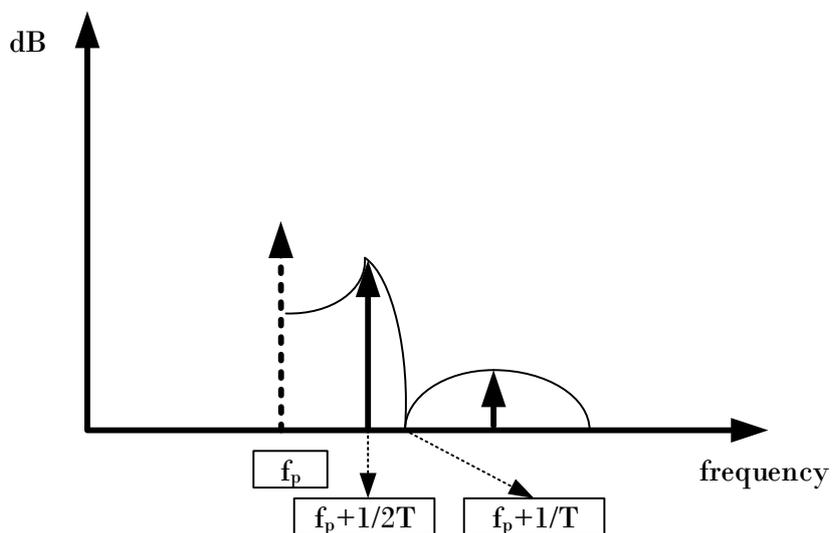


Figura D.4: Espectro da modulação SSB-ASK (imagem adaptada de [GS113a] e de [SC110]).

Por fim, a modulação PR-ASK é uma modulação onde a fase do sinal de maior amplitude para o sinal de menor amplitude (e vice-versa), sofre uma mudança de 180° . Esta modulação, em termos espectrais, é similar à modulação DSB-ASK, contudo suprime a portadora como

acontece na modulação SSB-ASK [SC110]. Em termos espectrais, a modulação PR-ASK está apresentadas na figura D.5.

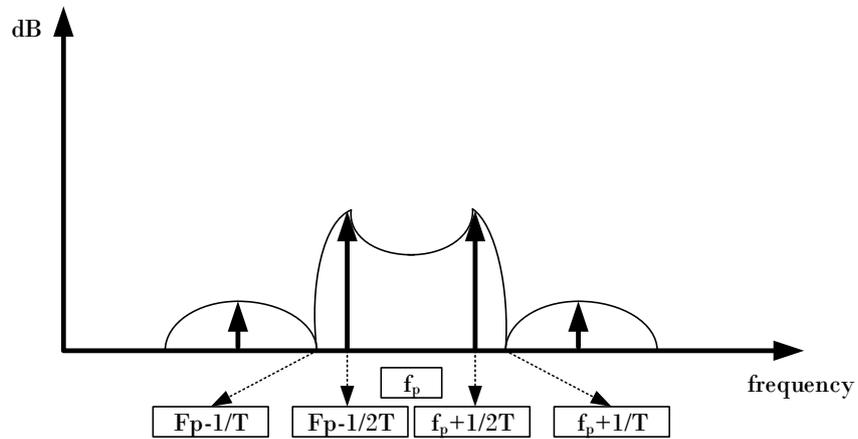


Figura D.5: Espectro da modulação PR-ASK (imagem adaptada de [GS113a] e de [SC110]).

Anexo E

Filtros FIR

Neste Anexo é apresentado o conceito de filtros FIR.

Um filtro FIR digital e discreto no tempo é matematicamente descrito pela equação E.1:

$$y[n] = b_0x[n] + b_1x[n - 1] + \dots + b_Nx[n - N] = \sum_{i=0}^N b_i \cdot x[n - i] \quad (\text{E.1})$$

onde, $y[n]$ e $x[n]$ são, respectivamente, o sinal de saída e de entrada no instante n , N é a ordem do filtro e $b_i (0 \leq i \leq N)$ são os coeficientes do filtro. A figura E.1 pretende mostrar esquematicamente o comportamento de um filtro FIR [Chi07].

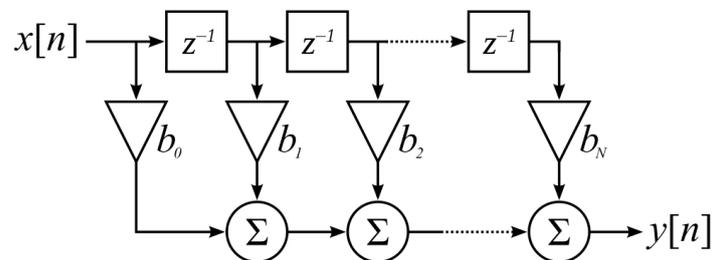


Figura E.1: Diagrama de blocos de um filtro FIR. z^{-1} representa um atraso temporal (imagem retirada de [Chi07]).

Existem uma grande diversidade de técnicas para projetar um filtro digital, entre todas destaca-se o *window design method*, onde uma resposta impulsional ideal é multiplicada por uma *window function*. Existe uma grande variedade de janelas, tais como, Butterworth, Chebyshev e Kaiser [MKM⁺11].

Bibliografia

- [AI08] Syed Ahson and Mohammad Ilyas. *RFID Handbook: Applications, Technology, Security, and Privacy*. ISBN-13: 978-1-4200-5499-6. ISBN-10: 1-4200-5499-6. CRC Press, 2008.
- [AIS04] AISCI. Alr-8610-accircularly polarized RFID antenna, 2004. Online em <http://www.aisci.de/upload/default/Datenblaetter/Alien/ALR-8610highresprinting.pdf>. Consultado em Agosto 2014.
- [Ant12] T. I. Tokic Antic. *RFID: Past, Present, Future*, volume 4. Scientific Publications of the state University of Novi Pazar Ser. A: Appl. Math. Inform. and Mech, 2012.
- [BM05] Manish Bhuptani and Shahram Moradpour. *RFID Field Guide: Deploying Radio Frequency Identification Systems*. ISBN: 0-13-185355-4. Prentice Hall PTR, 2005.
- [Byt13] ByteParadigm. Introduction to I2C and SPI protocols, 2013. Online em <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>. Consultado em Novembro 2014.
- [CCR10] Pedro Cruz, Nuno Borges Carvalho, and Kate Rmley. Designing and testing software defined radio. *IEEE microware magazine*, page 83, 2010.
- [Chi07] J.S Chitode. *Digital Processing System*. ISBN-10: 9350381958. ISBN-13: 978-9350381953. Technical Publications, 1 2007.
- [Dev09] Analog Devices. AD9122: Dual, 16-bit, 1230 MSPS, TxDAC+ digital-to-analog converter rev b, 2009. Online em http://www.analog.com/static/imported-files/data_sheets/AD9122.pdf. Consultado em Agosto 2014.
- [Dev10] Analog Devices. AD9523: Jitter cleaner and clock generator with 14 differential or 29 LVCMOS outputs. rev C, 2010. Online em http://www.analog.com/static/imported-files/data_sheets/AD9523.pdf. Consultado em Setembro 2014.
- [Dev11] Analog Devices. AD9643: 14-bit, 170 MSPS/210 MSPS/250 MSPS, 1.8 V dual analog-to-digital converter (ADC) rev E, 2011. Online em http://www.analog.com/static/imported-files/data_sheets/AD9643.pdf. Consultado em Agosto 2014.

- [Dev13] Analog Devices. AD-FMCOMMS1-EBZ functional overview, Janeiro 2013. Online em http://wiki.analog.com/resources/eval/user-guides/ad-fmcomms1-ebz/hardware/functional_overview. Consultado em Agosto 2014.
- [Dev14] Analog Devices. AD-FMCOMMS1-EBZ introduction, Dezembro 2014. Online em <http://wiki.analog.com/resources/eval/user-guides/ad-fmcomms1-ebz/introduction>. Consultado em Agosto 2014.
- [Dob07] Daniel Dobkin. *Communications Engineering series: The RF in RFID - Passive UHF RFID in Praticce*. ISBN 978-0-7506-8209-1. Elsevier Inc, 2007.
- [ER13] EPC-RFID. EPC information, 2013. Online em <http://www.epc-rfid.info/>. Consultado em Novembro 2014.
- [For07] SDR Forum. Cognitive radio definitions, Novembro 2007. Online em <http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V1.0-0.pdf>. Consultado em Agosto 2014.
- [GS113a] GS1. EPC radio-frequency identity protocols generation-2 UHF RFID: Specification for RFID air interface protocol for communications at 860 MHz and 960 MHz, Novembro 2013. Online em <http://www.gs1.org>. Consultado em Agosto 2014.
- [GS113b] GS1. Regulatory status for using RFID in the EPC gen 2 band (860 to 960 MHz) of the UHF spectrum, Maio 2013. Online em <http://www.gs1.org>. Consultado em Agosto 2014.
- [Imp14] Impinj. UHF RFID in ETSI region, Junho 2014. Online em <https://support.impinj.com/hc/en-us/articles/202756618-UHF-RFID-in-ETSI-Region>. Consultado em Novembro 2014.
- [Inn11] Wireless Innovation. What is software defined radio, 2011. Online em <http://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf>. Consultado em Agosto 2014.
- [Jin12] Sung Ho Jin, Cheng; Cho. A robust baseband demodulator for ISO 18000-6C RFID reader systems. *International Journal of Distributed Sensor Networks*, 2012. Online em <http://www.hindawi.com/journals/ijdsn/2012/406710/>. Consultado em Agosto 2014.
- [Jou14] RFID Journal. How much do RFID readers cost today? *RFID Journal*, 2014. Online em <http://www.rfidjournal.com/faq/show?86>. Consultado em Dezembro 2014.
- [Kul14] Ste Kulov. Intro to programmable logic & FPGAs, Abril 2014. Online em <http://pumpingstationone.org/2014/04/intro-to-programmable-logic-fpgas-april-27/>. Consultado em Novembro 2014.
- [Mit95] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, page 26 a 38, Maio 1995.

- [MKM⁺11] Saptarshi Mukherjee, Rajib Kar, Durbadal Mandal, Sangeeta Mondal, and S. P. Ghoshal. Linear phase low pass FIR filter design using improved particle swarm optimization. *IEEE Student Conference on Research and Development*, 2011. Department of Electronics and Communication Engineering National Institute of Technology, Durgapur, West Bengal, India.
- [Nes10] T. Nesimoglu. A review of software defined radio enabling technologies. *Mediterranean Microwave Symposium (MMS)*, 2010.
- [O’C06] Catherine O’Connor. Gen 2 EPC protocol approved as ISO 18000-6C. *RFID Journal*, Junho 2006. Online em <http://www.rfidjournal.com/articles/view?2481/>. Consultado em Agosto 2014.
- [Pra13] André Prata. Sistema de rádio digital para white spaces UHF. Master’s thesis, Departamento de Eletrónica, Telecomunicações e Informática. Universidade de Aveiro, 2013.
- [RFI10] RFIDiom. What is RFID?, 2010. Online em <http://www.rfidiom.com/rfid-technology-introduction/>. Consultado em Novembro 2014.
- [Rib11] Diogo Ribeiro. Sistema de medida analógico-digital para software-defined radios. Master’s thesis, Departamento de Eletrónica, Telecomunicações e Informática. Universidade de Aveiro, 2011.
- [RV09] Pavel V. Rao and Seshagiri K. V. LabVIEW-based UHF RFID tag test and measurement system. *IEEE Transactions in Industrial Electronics*, 56, 7 2009.
- [SC110] *Sistemas de Comunicação 1: III - Modulação de portadoras sinusoidais em sistemas digitais. Aulas técnicas, Universidade de Aveiro*, 2010.
- [Ste00] Jonathan Y. Stein. *Digital Signal Processing: A Computer Science Perspective*. ISBN 0-471-29546-9 Online ISBN 0-471-20059-X. John Wiley & Sons, Inc., 2000.
- [Tod12] ERT Electronics Radio Today. Single sideband modulation, SSB, 2012. Online em <http://www.electronics-radio.com/articles/radio/modulation/single-sideband/ssb-basics.php>. Consultado em Agosto 2014.
- [Tur10] Cristina Turcu. *Radio Frequency Identification Fundamentals and Applications, Design Methods and Solutions - 4 Design of Passive UHF RFID Tag Antennas Using Metamaterial-Based Structures and Techniques*. ISBN 978-953-7619-72-5. Braaten, Benjamin D. and Scheeler, Robert P. InTech, 2 2010.
- [Uni12] California State University. Single sideband (SSB), 2012. Online em <http://zimmer.csufresno.edu/pkinman/pdfs/Singleem> Agosto 2014.
- [Vas12] Saeed Vaseghi. Changing sampling rate: Decimation and interpolation. Brunel, University London, 2012. Online em http://dea.brunel.ac.uk/cmsp/Home_Saeed_Vaseghi/Decimation Consultado em Agosto 2014.
- [Vio05a] Bob Violino. The basics of RFID technology. *RFID Journal*, Janeiro 2005. Online em <http://www.rfidjournal.com/articles/view?1337/>. Consultado a Agosto 2014.

- [Vio05b] Bob Violino. The history of RFID technology. RFID Journal, Janeiro 2005. Online em <http://www.rfidjournal.com/articles/view?1338/>. Consultado em Agosto 2014.
- [Wik12] Wikipedia. Amplitude modulation, Novembro 2012. Online em http://en.wikipedia.org/wiki/Amplitude_modulation#mediaviewer/File:Amplitude_Modulated_Wave-hm-64.svg. Consultado em Novembro 2014.
- [Xil11] Xilinx. IP LogiCORE FIR compiler v5.0 ds534, Março 2011. Online em http://www.xilinx.com/support/documentation/ip_documentation/fir_compiler_ds534.pdf. Consultado em Agosto 2014.
- [Xil13] Xilinx. Zynq-7000 all programmable SoC overview DS190 (v1.6), Dezembro 2013. Online em http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf. Consultado em Agosto 2014.
- [Xil14a] Xilinx. LogiCORE IP AXI IIC bus interface v2.0 product guide. vivado design suite. pg090, Abril 2014. Online em http://japan.xilinx.com/support/documentation/ip_documentation/axi_iic/v2_0/pg090-axi-iic.pdf. Consultado em Agosto 2014.
- [Xil14b] Xilinx. LogiCORE IP clocking wizard v5.1 pg065, Abril 2014. Online em http://www.xilinx.com/support/documentation/ip_documentation/clk_wiz/v5_1/pg065-clk-wiz.pdf. Consultado em Agosto 2014.
- [ZDC10] Davide Zanetti, Boris Danev, and Srdjan Capkun. Physical-layer identification of UHF RFID tags. *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2010.
- [Zed12] Zedboard. Zynq evaluation and development hardware user's guide, Agosto 2012. Online em http://zedboard.org/sites/default/files/ZedBoard_HW_UG_v1_1.pdf. Consultado em Agosto 2014.
- [ZKP09] Wei Zhou, Gaurav Kapoor, and Selwyn Piramuthu. Modified slotted ALOHA protocol for RFID tag collisions. 2009. Information Systems and Operations Management ESCP Europe, France. University of Florida, USA.