

JOSÉ JÚLIO GUIMARÃES FERNANDES

**IMPLEMENTAÇÃO DE ESPALHAMENTO ESPECTRAL POR
SEQUÊNCIA DIRETA**

Porto Alegre

2002

JOSÉ JÚLIO GUIMARÃES FERNANDES

IMPLEMENTAÇÃO DE ESPALHAMENTO ESPECTRAL POR SEQUÊNCIA DIRETA

ORIENTADOR: Prof. Dr. Álvaro Augusto A. de Salles

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), da Universidade Federal do Rio Grande do Sul (UFRGS), como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Automação e Instrumentação Eletro-Eletrônica.

Porto Alegre

2002

JOSÉ JÚLIO GUIMARÃES FERNANDES

IMPLEMENTAÇÃO DE ESPALHAMENTO ESPECTRAL POR SEQUÊNCIA DIRETA

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Álvaro Augusto A. de Salles, UFRGS

Ph.D. pela University College London, Londres, Inglaterra

Banca Examinadora:

Prof. Jorge Amoretti Lisboa, Doutor, UFRGS

Dr. pela Universidade Federal do Rio Grande do Sul – UFRGS – 1986

Prof. Luigi Carro, Doutor, UFRGS

Dr. pela Universidade Federal do Rio Grande do Sul – UFRGS – 1996

Prof. Paulo Roberto Girardello Franco Ph.D., PUCRS

Dr. pela University of New Mexico – 1993

Prof. Paulo Roberto Rosa Lopes Nunes Ph.D., IME/RJ

Dr. pela Stanford University – 1980

Coordenador do PPGEE: _____

Prof. Dr. Carlos Eduardo Pereira

Porto Alegre, Dezembro de 2002.

DEDICATÓRIA

À minha esposa e a meu filho, pela paciência demonstrada durante a elaboração desse trabalho.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, pelos recursos postos à disposição.

Ao Professor Dr. Jorge Amoretti Lisboa e ao Professor Dr. Álvaro Augusto de Salles, pelas valiosas contribuições a esse trabalho.

Ao Professor Claudio Fernández e ao Engenheiro Paulo Serafini, pela colaboração e contribuições a esse trabalho.

RESUMO

O presente trabalho tem como objetivo a descrição e apresentação dos resultados obtidos através de simulações e implementação da técnica de modulação por espalhamento de espectro (Spread Spectrum). Maior ênfase será dada à técnica de espalhamento por seqüência direta (DSSS), dada a larga utilização nos sistemas de comunicação correntemente em uso, assim como serão apresentados os resultados de simulações e implementação de um sistema de comunicação Spread Spectrum utilizando o microcontrolador PIC 16F876 para a interligação de computadores padrão PC via interface serial RS232.

Palavras-chaves: Engenharia Elétrica, Comunicações Sem Fio, Espalhamento Espectral.

ABSTRACT

The aim of this work is to characterize the Spread Spectrum modulation technique and show the obtained results from computer simulations and physical implementation. Emphasis is given to the Direct Sequence Spread Spectrum technique because of its large employment in current communication systems. Obtained results from computer simulations and physical implementation of a Spread Spectrum communication system using PIC16F876 microcontroller for PC computer connection via RS232 serial interface are also presented.

Keywords: Electrical Engineering, Wireless Communication, Spread Spectrum.

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	HISTORICO E APLICAÇÕES DOS SISTEMAS DE BANDA ESPALHADA.....	13
1.2	OBJETIVOS E DESCRIÇÃO DO TRABALHO.....	16
2	REVISÃO DE LITERATURA	19
3	SISTEMAS DE ESPALHAMENTO ESPECTRAL	20
3.1	DESCRIÇÃO DO SISTEMA DSSS	20
3.2	ANÁLISE MATEMÁTICA	23
3.2.1	<i>Sequência Direta com Modulação BPSK</i>	23
3.2.2	<i>Sequência Direta com Modulação QPSK</i>	27
3.3	ANÁLISE DOS BLOCOS DE UM SISTEMA SPREAD SPECTRUM.....	28
3.3.1	<i>Transmissor Spread Spectrum BPSK</i>	28
3.3.2	<i>Receptor Spread Spectrum BPSK</i>	33
4	SIMULAÇÃO.....	40
4.1	SISTEMA PROPOSTO PARA SIMULAÇÃO	40
4.1.1	<i>Transmissor</i>	41
4.1.2	<i>Receptor</i>	42
4.2	SIMULAÇÃO DO SISTEMA DSSS	48
5	IMPLEMENTAÇÃO FÍSICA	56
5.1	CONSIDERAÇÕES INICIAIS	56
5.2	MICROCONTROLADOR PIC16F876.....	57
5.3	DESCRIÇÃO DO HARDWARE DO SISTEMA DSSS/PIC16F876.....	57
5.4	DESCRIÇÃO DO SOFTWARE DO SISTEMA DSSS/PIC16F876	62
5.5	TESTES REALIZADOS	71
5.6	CIRCUITO RECUPERADOR DE RELÓGIO.....	82
6	CONCLUSÃO.....	84
7	APÊNDICE A: LISTAGEM DO PROGRAMA FONTE UTILIZADO PARA A IMPLEMENTAÇÃO FÍSICA DO SISTEMA PIC16F876.....	90
8	APÊNDICE B: LISTAGEM DO PROGRAMA FONTE PARA GERAÇÃO DA TABELA PN DE 8 BYTES A PARTIR DA SIMULAÇÃO DE UMA CADEIA DE 6 SHIFT REGISTERS COM REALIMENTAÇÕES NAS SAÍDAS 1 E 6.....	97
9	APÊNDICE C: TRABALHO SOBRE MODULAÇÕES EM FASE E SPREAD SPECTRUM APRESENTADO DURANTE O CURSO	99
10	APÊNDICE D: DQPSK $\Pi/4$ IMPLEMENTAÇÃO E SIMULAÇÃO.....	112
11	APÊNDICE E: ARTIGO PUBLICADO NO SEMINÁRIO BRASILEIRO DE MICROONDAS E ÓPTOELETRÔNICA (SBMO 2002).....	129
12	ANEXO A: NORMA ANATEL PARA O USO DE BANDAS POR EQUIPAMENTOS BASEADOS EM TECNOLOGIA DE ESPALHAMENTO ESPECTRAL	138

LISTA DE ILUSTRAÇÕES

FIGURA 1: Diagrama de blocos para: a) transmissor DSSS b) receptor DSSS.....	22
FIGURA 2: Sinais BPSK a) sinal digital b) sinal modulado em BPSK c) seqüência pseudoaleatória d) sinal spread spectrum DSSS.....	25
FIGURA 3: Espectro de potência para sinais BPSK e DSSS (Pot. em dBm e freq. em MHz) ..	27
FIGURA 4: Diagrama de blocos do gerador de seqüências pseudo aleatórias.....	30
FIGURA 5: Exemplo de enlace com efeitos de multi percurso (reflexão).....	32
FIGURA 6: Diagrama de blocos do misturador Spread Spectrum.....	32
FIGURA 7: Diagrama de blocos da etapa de FI e RF.....	33
FIGURA 8: Diagrama de blocos da etapa de FI e RF do receptor.....	34
FIGURA 9: Diagrama de blocos das etapas misturadora e de temporização/sincronismo.....	34
FIGURA 10: Diagrama de blocos de um filtro casado digital.....	36
FIGURA 11: Diagrama de blocos do circuito DLL.....	38
FIGURA 12: Diagrama de blocos proposto para simulação MatLab.....	41
FIGURA 13: Diagrama de blocos do transmissor DSSS/DBPSK.....	42
FIGURA 14: Diagrama de blocos do receptor DSSS/DBPSK (RASE).....	43
FIGURA 15: Diagrama de estados da Máquina de Estados RASE.....	44
FIGURA 16: Diagrama de estados do detetor de correlação.....	46
FIGURA 17: Circuito lógico da máquina de estados com Flip Flops tipo D.....	47
FIGURA 18: Circuito lógico do detetor de correlação.....	47
FIGURA 19: Curva BER x S/N para um sinal com modulação DBPSK em ambiente AWGN49	
FIGURA 20: Espectro simulado dos sinais em banda base Y e espalhado Tx_DSSS.....	51
FIGURA 21: Sinais DSSS/DBPSK sem ruído a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) saída do filtro PB e) saída do decodificador DBPSK f) sinal digital correlacionado (Y) g) Trigger h) Load i) Clear.....	52
FIGURA 22: Sinais DSSS/DBPSK com multipercurso e ruído a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) sinal DSSS/DBPSK recebido c/ multipath e ruído e) saída do filtro PB f) saída do decodificador DBPSK g) sinal digital correlacionado (Y).....	53
FIGURA 23: Sinais DSSS/DBPSK com interferência a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) sinal DSSS/DBPSK recebido c/ interferência e) saída do filtro PB f) saída do decodificador DBPSK g) sinal digital correlacionado (Y).....	54
FIGURA 24: Sinais DSSS/DBPSK com ruído maior que 10dB acima a) sinal digital b) sinal DSSS/DBPSK recebido c/ ruído c) Trigger d) sinal digital correlacionado (Y).....	55
FIGURA 25: Sinais DSSS/DBPSK com interferência a) sinal digital b) sinal DSSS/DBPSK recebido c/ interferência c) sinal digital correlacionado (Y).....	55
FIGURA 26: Hardware do sistema PIC16F876 / DSSS.....	58
FIGURA 27: Tela do SW Hyperterminal com as respostas dos testes com o PIC16F876 / DSSS.....	59
FIGURA 28: Diagrama de blocos do software PIC16F876 / DSSS.....	61
FIGURA 29: Diagrama de temporização para transmissão de um caracter.....	64
FIGURA 30: Diagrama de estados para o software Dsss.c.....	67

FIGURA 31: Estrutura de montagem e transmissão de chips para um caracter (“A” no exemplo)	68
FIGURA 32: Estrutura de recepção e correlação com a seqüência PN para obter o caracter (“A” no exemplo)	69
FIGURA 33: Montagem com a interligação de duas placas PIC16F876	72
FIGURA 34: Rx medido para o caracter “A”	73
FIGURA 35: TX_DSSS medido para o caracter “A”	74
FIGURA 36: TX_DSSS e sinal em RB2 medidos para a transmissão contínua do caracter “A”	75
FIGURA 37: Espectro de TX e TX_DSSS medidos para a transmissão contínua do caracter “A”	76
FIGURA 38: TX_DSSS e RX medidos após a entrada do caracter “A” via emulador de terminal	77
FIGURA 39: TX_DSSS e RX_DSSS medidos após a entrada do caracter “A” via emulador de terminal	78
FIGURA 40: Montagem com os módulos transmissor e receptor de RF/FM	80
FIGURA 41: Medida do espectro da portadora do transmissor FM (417,94MHZ)	80
FIGURA 42: Medida do espectro do sinal modulado e sem espalhamento	81
FIGURA 43: Medida do espectro do sinal modulado e com espalhamento	81
FIGURA 44: Recuperador de Relógio com o CI 74HC4046	83
FIGURA 45: Geração interna de relógio pelo timer0 (RTCC) a partir da primeira transição de Rx_dsss	83

LISTA DE ABREVIATURAS

ϕ	- Fase aleatória
ω_0	- Frequência angular (rad/s)
$\theta_d(t)$	- Fase variável no tempo
AWGN	- Additive White Gaussian Noise
BER	- Bit Error Rate
BPSK	- Binary Shift Keying
$c(t)$	- Seqüência pseudo aleatória
CDMA	- Code Division Multiple Access
DBPSK	- Differential Binary Phase Shift Keying
DPSK	- Differential Phase Shift Keying
DQPSK	- Differential Quad Phase Shift Keying
DSSS	- Direct Sequence Spread Spectrum
EEPROM	- Electrical Erase Programmable Read Only Memory
ERB	- Estação Rádio Base
FHSS	- Frequency Hopping Spread Spectrum
FM	- Frequency Modulation
FPGA	- Field Programmable Gate Array
G_p	- Ganho de Processamento
PN	- Pseudo Random Noise
QPSK	- Quad Phase Shift Keying
RASE	- Rapid Acquisition by Sequential Estimation
RF	- Rádio Frequência
RISC	- Reduced Instructions Set Computer
Rx	- Recepção RS232
$S_d(t)$	- Sinal modulado em fase
$S_t(t)$	- Sinal com espalhamento espectral

T	- Duração de um bit
T _c	- Duração de um chip
T _d	- Atraso de propagação
T _{de}	- Atraso de propagação estimado
T _x	- Transmissão RS232
USART	- Universal Synchronous Asynchronous Receiver Transmitter
W-LAN	- Wireless Local Area Network

1 INTRODUÇÃO

Nos últimos anos tem se verificado a ampla disseminação dos sistemas de comunicação sem fio (*wireless*) e, em particular, de sistemas baseados na tecnologia de espectro espalhado (*Spread Spectrum*). A técnica Spread Spectrum, como conceito de modulação, já existe há várias décadas, tendo sido inicialmente de uso restrito às forças armadas americanas, foi liberada para uso civil a partir dos anos 80 pelo governo dos Estados Unidos. No Brasil foram liberadas as primeiras faixas para uso de transceptores spread spectrum somente em meados dos anos 90. Basicamente, consiste em aumentar a largura do espectro de um sinal modulado digitalmente de forma que seja dificultada a detecção ou interferência deste sinal. Além de razões técnicas, a tecnologia de espalhamento do sinal vem se popularizando muito rapidamente por não necessitar de um licenciamento formal para sua utilização, desde que sejam utilizadas as faixas disponíveis para tais aplicações. As aplicações dessa técnica aparecem na telefonia celular, com o sistema CDMA (Code Division Multiple Access) na banda de 850MHz e na Internet sem fio, interconexão de redes locais (W-LANs), ponto-multiponto e ponto-a-ponto, e em enlaces de acesso de dados e telefonia (Nx64kbps a 4x2048kbps) nas bandas de 2.4GHz e 5.8GHz (sem necessidade de licenciamento).

Existem várias formas de se realizar o espalhamento espectral. Entre as mais utilizadas estão aquelas em que a frequência da portadora é variada de acordo com o um código pseudo aleatório (FHSS - Frequency Hopping Spread Spectrum), e a técnica baseada na multiplicação do sinal modulado por uma seqüência pseudo aleatória (DSSS - Direct Sequence Spread Spectrum). Em alguns casos é também utilizada uma combinação dos dois métodos (Hybrid FH/DS SS). Para fins de análise dos sinais nesse trabalho, será considerada a técnica de espalhamento DSSS nas modulações digitais BPSK (Binary Shift Keying), pela simplicidade de análise, e DBPSK (Diferencial Binary Shift Keying), que são das mais utilizadas nos modelos comerciais de rádios spread spectrum.

1.1 HISTORICO E APLICAÇÕES DOS SISTEMAS DE BANDA ESPALHADA

De uma maneira geral os sistemas de comunicações considerados atualmente como sendo de banda espalhada (Spread Spectrum – SS) são aqueles em que a banda do espectro utilizada na comunicação é bem maior do que a necessária para transmitir a informação desejada. As primeiras idéias de utilização de sinais de banda espalhadas apareceram por volta de 1920 [1], propostas para minimizar os efeitos de desvanecimento devido à interferência de multipercurso (*fading*) em sistemas de comunicações. A idéia era espalhar a energia do sinal em uma larga banda de frequências e reconstruir o sinal no receptor, pela recuperação de todas as componentes de frequência, o sinal original. Com isto, se em uma parte da banda emitida ocorresse interferência destrutiva, em outras regiões do espectro a interferência seria construtiva, permitindo assim a recuperação da informação emitida com um mínimo de distorção. Entretanto, a tecnologia existente na época não permitiu a implementação prática destas propostas.

A ocorrência da segunda guerra mundial trouxe consigo uma necessidade urgente de sistemas de comunicações seguros, capazes de manter comunicações sigilosas em presença de interferências deliberadamente geradas pelo inimigo. A guerra trouxe também um forte desenvolvimento tecnológico em diversas áreas, e os dispositivos e técnicas de comunicações não foram exceção. O resultado é que, ao final do conflito, ambos os lados já utilizavam sistemas de comunicações sofisticados, muitos dos quais utilizavam técnicas de espalhamento espectral. A aplicação destas técnicas em sistemas de comunicações militares tornou-se proveitosa porque, além da imunidade ao desvanecimento, o espalhamento espectral traz uma redução proporcional da intensidade do sinal por intervalo de frequência, dificultando assim a detecção do mesmo. Como ver-se-á adiante, a privacidade na comunicação em sistemas de banda espalhada é maximizada pela utilização de códigos pseudo aleatórios durante o processo de espalhamento. A recuperação do sinal só é possível se o receptor conhecer o código utilizado no espalhamento.

Devido ao fato de o desenvolvimento dos sistemas de banda espalhada ter se dado no âmbito militar, o seu conhecimento e aplicação ficaram restrito à aplicações de interesse

bélico por um longo tempo. Deve se mencionar, entretanto, a existência de um pedido de patente feito pela atriz de Hollywood Hedy Lamarr e de George Antheil para um sistema de comunicação secreto em 1942 [1]. A idéia surgiu para ambos quando estavam comentando das possibilidades de evitar a interceptação das comunicações norte-americanas pelos nazistas durante a Segunda Guerra. Basicamente pensaram na transmissão de mensagens com troca de frequência da portadora numa seqüência apenas conhecida pelo transmissor e pelo receptor. A idéia evoluiu dentro das organizações militares americanas de forma que, quando as mensagens eram captadas por antenas inimigas, parecia que existia apenas ruído. A primeira abordagem era portanto de um sistema SS baseado em salto de frequência (Frequency Hopping Spread Spectrum - FHSS).

As aplicações civis dos sistemas SS começam a ocorrer a partir de 1985, quando o órgão regulador das comunicações americanas, Federal Communication Commission (FCC), liberou o uso dessa tecnologia para aplicações comerciais, com as bandas de 2,4 GHz e 5,8 GHz reservadas para aplicações SS. A banda de 2,4 GHz é conhecida por ser também reservada para aplicações ISM (Industrial, Scientific and Medical) , usada por uma grande variedade de equipamentos, sendo o mais conhecido o forno de microondas. Equipamentos de comunicação de dados, telefonia ponto a ponto, ponto multiponto, Wireless LAN, Internet sem fio, tecnologia Bluetooth (pequenas distâncias), padrão IEEE 802.11, etc disputam a mesma faixa, que compreende o espectro de 2400-2483.5 MHz.

No Brasil a regulamentação dos sistemas de banda espalhada data de 1996, de acordo com a Norma No 12/96 (CONDIÇÕES DE USO DE FREQUÊNCIAS NAS FAIXAS DE 902 - 928 MHz, 2400 - 2483,5 MHz E 5725-5850 MHz, POR EQUIPAMENTOS DE RADIOCOMUNICAÇÃO EMPREGANDO TÉCNICA DE ESPALHAMENTO ESPECTRAL) [11]. A Agência Nacional de Comunicações (ANATEL) liberou essas faixas, não sendo necessário o licenciamento dos enlaces de comunicação em microondas instalados, apenas os equipamentos devem ser certificados.

Em vista da explosiva demanda por sistemas spread spectrum, existe uma grande variedade de fabricantes de equipamentos [2] os quais são baseados em uma pequena quantidade de *chipsets* disponíveis. Um dos *chipsets* pesquisados é o DSSS PRISM HFA3824A da Intersil Corporation [3], que se constitui em uma família de chips com os quais

podem-se construir sistemas para diversas aplicações, variando-se a interface de dados com o usuário.

A tecnologia Spread Spectrum está evoluindo também na interface aérea com o usuário de telefonia móvel [4]. Os sistemas celulares CDMA (Code Division Multiple Access) também são baseados nessa tecnologia, sendo usados na banda de 850 MHz, cuja licença é comprada pela operadora de telefonia celular para cada região determinada pelo órgão regulador, a qual pode fazer a escolha entre os sistemas FDMA, TDMA ou CDMA [4]. CDMA é uma tecnologia de múltiplo acesso por divisão de código, ou seja, vários usuários podem ocupar a mesma frequência ao mesmo tempo, porém o código pseudo aleatório (PN) usado para fazer o espalhamento do canal de voz de cada um é diferente. A estação rádio base fará a discriminação das chamadas telefônicas celulares através da atribuição desses códigos para cada usuário em conversação. Embora os equipamentos CDMA sejam mais caros que os concorrentes, existe um ganho de eficiência de 7 para 1 em relação ao sistema analógico FDMA/AMPS, maior que o de 3 para 1 do TDMA [4]. Consequentemente, em áreas congestionadas são necessárias menos Estações Rádio Bases (ERBs). A capacidade de transmissão de dados pelo celular também é maior como visto nos aparelhos celulares CDMA 1xRTT, que constitui na chamada geração 2,5 do padrão CDMA [10]. Aqui abre-se um parêntese, os sistemas celulares evoluíram a partir da primeira geração (FDMA por exemplo) até a segunda geração (TDMA, CDMA). As técnicas intermediárias, entre a segunda geração e a terceira, ainda por vir, com maior velocidade de transmissão de dados, são chamadas de geração 2,5. O padrão 1xRTT é uma evolução do sistema CDMA IS-95 para que possam ser atingidas velocidades de cerca de 144 kbps de pico [10]. O sistema TDMA, com a aplicação CSD (Circuit Switched Data), atinge no máximo 9600 bps. As aplicações de Internet Móvel, tais como WAP (Wireless Application Protocol), fazem a comunicação de dados sobre essas técnicas nos sistemas TDMA e CDMA.

O sistema CDMA é considerado de segunda geração (2G) entre outros, tais como TDMA e GSM (Global System for Mobile Communication), mas a evolução para a terceira geração (3G) de celulares irá passar necessariamente para soluções de múltiplo acesso por divisão de código, como as tecnologias IS-2000 (evolução do CDMA IS-95) e W-CDMA (migração de GSM para 3G). A evolução para 3G permitirá o aumento do número de usuários

por canal RF (melhor eficiência espectral) e velocidades de até 2Mbps de acesso para os terminais móveis [10]. Basicamente será a entrada na era da banda larga para os celulares. As aplicações seriam em acesso de celulares a Internet, Internet móvel para PDAs e laptops, comunicação de dispositivos hand held com servidores empresariais, comunicação multimídia, etc...

A tecnologia de espalhamento espectral surge também como solução para multi acesso sem fio dos mais diversos dispositivos em uma pequena área, com dimensões não maiores do algumas dezenas de metros. Diversas propostas tem sido apresentadas para viabilização prática desta aplicação, mas só muito recentemente se iniciou a comercialização dos primeiros dispositivos, com o predomínio do sistema Bluetooth, resultado de uma parceria da IBM com a Ericsson. Entretanto, embora promissor, o sistema vem ainda enfrentado dificuldades em alcançar uma maior disseminação, devido ao alto custo de sua implementação [5].

E por fim, a tecnologia Spread Spectrum é utilizada para a comunicação entre aparelhos GPS (Global Positioning System) e os satélites para localização [4]. A modulação usada é a BPSK/DSSS, escolhida devido à grande robustez quanto a ruídos, lembrando que o GPS é um receptor de sinais emitidos pelos satélites, que chegam com baixíssima potência ao aparelho.

1.2 OBJETIVOS E DESCRIÇÃO DO TRABALHO

Conforme mencionado acima, já existem diversos sistemas comerciais para operação de sistemas SS. Porém, estes sistemas são voltados para aplicações específicas, ainda a um custo relativamente alto. Por outro lado, embora a bibliografia existente sobre o assunto seja apreciável, a grande maioria dos trabalhos estão voltados para estudos teóricos da técnica. A literatura associada aos sistemas comerciais também é apresentada na forma convencionalmente chamada “receita de bolo”, isto é, os detalhes de sua implementação prática não são descritos. O que é fornecido é apenas um diagrama de bloco de como interligar os diversos componentes. Provavelmente isto se deve ao fato de que, no início de seu desenvolvimento, o caráter sigiloso das aplicações militares e, atualmente, ao interesse de

quem detém a tecnologia de manter o domínio sobre a mesma, através do uso dos chamados sistemas proprietários.

O objetivo inicial deste trabalho foi o de estudar os principais métodos utilizados para espalhar e recuperar o sinal e, num segundo momento, o de aprender como fazer, implementando na prática um sistema comunicações de banda espalhada de baixo custo.

A partir do estudo teórico, foi feita a escolha do sistema a ser implementado, tendo em vista a disponibilidade de componentes e a infra-estrutura existente no Laboratório de Comunicações. Neste ponto foi entendido que o método de sequenciamento direto (DS) seria o mais adequado devido à maior simplicidade de construção do hardware. O passo seguinte foi o de simular em computador todos os blocos de geração do sinal de banda espalhada e recuperação do sinal original. Após a análise dos resultados desta simulação partiu-se para a especificação dos sistema prático a ser desenvolvido. Duas alternativas foram avaliadas: implementação em dispositivos digitais programáveis tipo FPGA ou o uso de microcontroladores. Embora apresentasse limitações em frequência, por razões de disponibilidade e também devido à simplicidade, rapidez e reprogramabilidade em circuito, optou-se pelo uso de um microcontrolador Microchip PIC 16F876. Como os objetivos básicos do trabalho são os de dominar o processo de geração e decodificação do sinal de banda espalhada, optou-se na parte de RF pela utilização de módulos comerciais de baixo custo utilizados em sistemas de controle remoto.

Devido também a razões de disponibilidade e simplicidade de realização, foi estabelecido como objetivo prático a ser alcançado realizar a comunicação entre dois microcomputadores, utilizando para isto a interface RS232.

No capítulo 3 é apresentada análise teórica de um sistema de banda espalhada, com ênfase no processo de sequenciamento direto (DSSS). Também são descritos os principais tipos de modulação utilizadas neste método.

Os resultados das simulações realizadas no software MatLab, com módulo Simulink, são apresentados no capítulo 4, incluindo as diversas situações de interferência e ruído no meio de transmissão

A realização prática do sistema está no capítulo 5. É apresentado o hardware e uma descrição do software utilizado, assim como os resultados experimentais utilizados para caracterizar a performance do sistema.

O capítulo 7 apresenta as conclusões e recomendações.

As listagens do programa fonte da implementação no microcontrolador PIC16F876 e do programa gerador de tabelas de códigos PN estão nos apêndices A e B, respectivamente.

Os trabalhos desenvolvidos durante o curso sobre modulação por espalhamento espectral e DQPSK $\pi/4$ estão nos apêndices C e D.

O apêndice E apresenta o artigo publicado no Seminário Brasileiro de Microondas e Optoeletrônica (SBMO) em 2002.

A Norma Anatel regulamentando o uso das faixas de frequência para uso de sistemas de espectro espalhado está no anexo A.

2 REVISÃO DE LITERATURA

Para a elaboração desta dissertação foram estudados os principais livros disponíveis sobre espalhamento espectral bem como manuais de fabricantes de circuitos integrados dedicados e de equipamentos de comunicações. A descrição resumida de cada publicação é apresentada a seguir.

O livro “Introduction to Spread Spectrum Communications” (Prentice Hall, 1995) descreve a teoria de comunicações com espectro espalhado e a análise dos principais blocos necessários em circuitos de transmissão e recepção.

O livro “Spread Spectrum Update” (Tucson Amateur Packet Radio Corporation, 1998) apresenta uma série de artigos com aplicações práticas em circuitos transmissores e receptores com espectro espalhado.

O artigo “Aurora 2400 Spread Spectrum Digital Radio for Unlicensed T1/E1 Transport in the 2.4GHz ISM Band Application” (Harris Corporation, 1998) apresenta uma revisão da história e teoria de sistemas de comunicação de espectro espalhado e a introdução de um equipamento de transmissão de tributários E1/T1 na banda de 2,4GHz.

Por fim, o artigo “Tecnologia 3G, Visão Geral da Tecnologia 2.5/3G” (Agilent Technologies, 2000) descreve as tendências mais atuais até o momento da publicação para aplicações em comunicações móveis.

3 SISTEMAS DE ESPALHAMENTO ESPECTRAL

Existem várias formas de se obter o espalhamento espectral. Entre as mais utilizadas estão a em que a frequência da portadora é variada de acordo com um código pseudo aleatório (FHSS - Frequency Hopping Spread Spectrum), e a técnica baseada na multiplicação por uma seqüência pseudo aleatória (DSSS - Direct Sequence Spread Spectrum) do sinal modulado. Em alguns casos, principalmente em aplicações militares, é também utilizada uma combinação dos dois métodos (Hybrid FH/DS SS). Como o fator custo é muito importante nessa abordagem, a opção DSSS é a de menor custo de ser implementada em hardware pois o circuito de espalhamento é muito mais simples. Mais simples inclusive para a realização de testes em banda básica (antes da modulação).

3.1 DESCRIÇÃO DO SISTEMA DSSS

Embora a primeira abordagem de espalhamento espectral (principalmente em aplicações militares) tenha sido a de salto em frequência (FHSS), a tecnologia mais usada em sistemas comerciais é a de espalhamento espectral por seqüência direta (DSSS). O espalhamento espectral é obtido pela multiplicação do sinal digital (já modulado ou ainda em banda básica) por uma seqüência pseudo-aleatória (Pseudo Noise Sequence), dita seqüência PN de maior frequência.

No sinal digital em banda básica, cada unidade mínima de informação é chamada de bit. Nos sistemas DSSS, para diferenciar, cada unidade mínima de informação é chamada de *chip*. Assim tem-se a conversão de um sinal com uma taxa de bits por segundo (bps) para um sinal DSSS com taxa de chips por segundo (cps). A razão entre a taxa de chips e a taxa de bits é chamada de ganho de processamento (G_p), que também define o quanto pode ser diminuído da relação sinal ruído mínima para demodulação em uma determinada taxa de erros admissível. O ganho de processamento é expresso em decibéis, e serve de comparação entre os sinais com a mesma modulação.

As modulações mais utilizadas são as que codificam a informação na fase do sinal a ser transmitido [4]. Devido à natureza da técnica DSSS, que é baseada na multiplicação de

uma seqüência PN pelo sinal não espalhado espectralmente, as modulações em fase se prestam melhor para simplificação do projeto dos circuitos transmissor e receptor. Também pode-se pensar que possuem uma relação sinal ruído mais baixa para uma probabilidade de taxa de erros padronizada em relação às demais modulações. Como nos sistemas spread spectrum, a limitação de largura de banda não é o fator mais importante e pressupõe-se que o sinal deva ser extraído de meio extremamente ruidoso e com interferência na mesma banda, as modulações em fase são mais apropriadas do que as modulações em amplitude, que são mais afetadas pelo ruído. As modulações mais usadas são BPSK, QPSK, DBPSK e DQPSK, cujas características de eficiência espectral e relação sinal ruído mínima para taxa de erros de bit de 10^{-6} em canal AWGN (ruído aditivo gaussiano branco) sem desvanecimento (*fading*), estão listadas na tabela 1.

No lado receptor deve ser feita a operação inversa, ou seja, a multiplicação do sinal recebido pela réplica da seqüência PN gerada localmente com sincronismo de relógio e código obtido a partir desse sinal de entrada. Estando os sinais em fase, haverá a correlação e será obtido o sinal de dados modulado original na taxa de bits desejada. Para os sinais captados na mesma banda, mas com código PN diferente, haverá um maior espalhamento e diminuição do nível detectado de recepção devido ao efeito do ganho de processamento.

Considerando um enlace de rádio que transmite um feixe E1 (2048Mbps) em 2,4GHz com modulação DQPSK. Se o ganho de processamento é de 10dB, pode-se concluir:

- a) A taxa de transmissão de símbolos será de 1024 kbauds, pois em DQPSK cada variação de fase corresponde a 2 bits.
- b) Para o ganho de processamento de 10dB, deve-se ter uma taxa dez vezes maior em kchips/seg ($G_p = 10 \log f_{cps} / f_{bps}$), ou seja, de 10 240 kcps.
- c) Como a eficiência espectral de DQPSK é de 1 bps / Hz, a largura do canal ocupada pelo sinal espalhado espectralmente é de 10, 24 MHz.
- d) A relação sinal ruído mínima para uma probabilidade de taxa de erros de bit de 10^{-6} em DQPSK para o sinal E1 em um meio sem ocorrências de desvanecimento é de cerca de 13dB. Com o ganho de processamento de 10dB, essa relação sinal ruído mínima pode cair para 3dB, ou seja, o sinal recebido pode estar apenas com o dobro da potência de ruído e haverá BER de 10^{-6} para o sinal E1. Naturalmente, se forem levadas em conta as

possibilidades de degradação do sinal recebido devido a ocorrência de desvanecimentos, a relação sinal ruído mínima deve ser maior.

TABELA 1: Modulações digitais comparadas em eficiência espectral e relação sinal ruído mínima

Modulação	Eficiência Espectral (bps / Hz)	Relação sinal ruído p/ BER = 10^{-6} (sem fading)
BPSK	0.5	11 dB
DBPSK	0.5	11 dB
QPSK	1.0	11 dB
DQPSK	1.0	13 dB
4FSK	0.25	11 dB

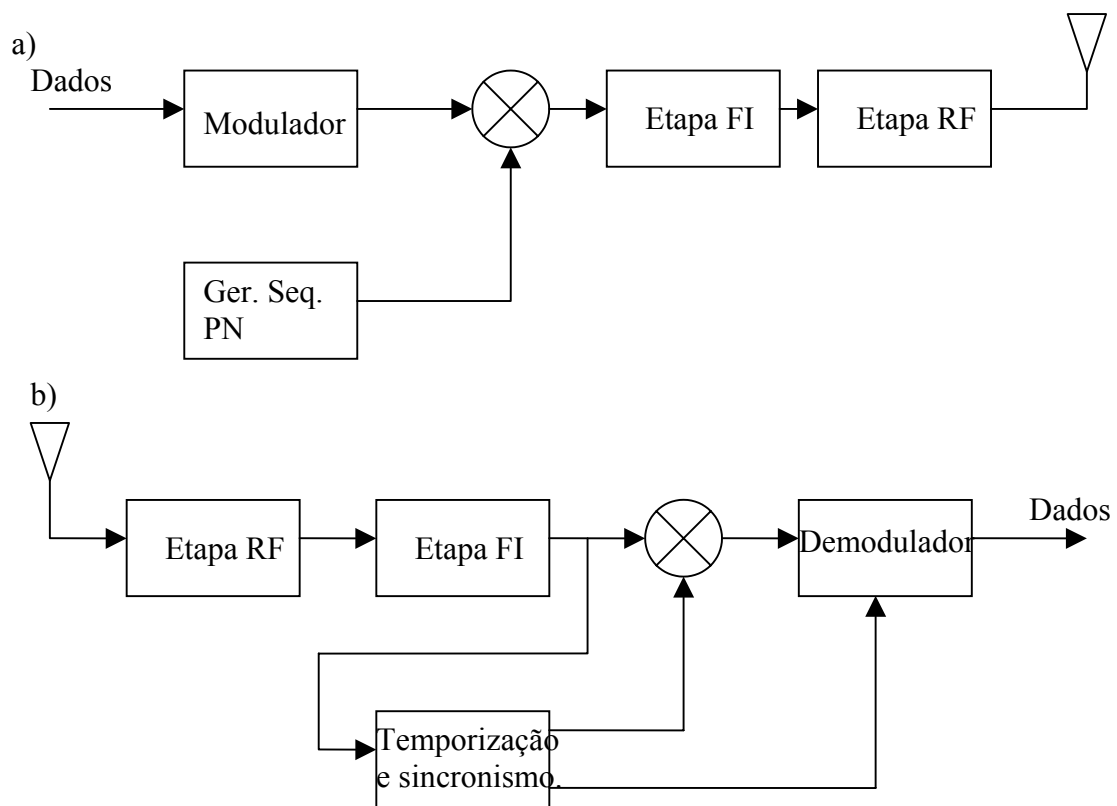


FIGURA 1: Diagrama de blocos para: a) transmissor DSSS b) receptor DSSS

3.2 ANÁLISE MATEMÁTICA

3.2.1 SEQÜÊNCIA DIRETA COM MODULAÇÃO BPSK

A forma mais simples de obter-se o espalhamento espectral de um sinal modulado digitalmente é o uso de seqüência direta com modulação BPSK. Para tanto é feita a multiplicação do sinal modulado em BPSK por um trem de pulsos (amplitudes de -1 ou +1) variando de acordo com uma seqüência pseudo aleatória (PN, do inglês Pseudo Random Noise). O resultado é um sinal com inversões de fase não apenas em função do sinal digital modulante, mas também da seqüência pseudo aleatória de espalhamento. A demodulação desse sinal será feita apenas com a correta seqüência pseudo aleatória sendo reproduzida no receptor, inclusive estimando a fase em que o sinal recebido se encontra (variável de acordo com o caminho de propagação). Para a análise matemática, o sinal modulado em BPSK a ser transmitido fica da forma [4]:

$$S_d(t) = A \cos(\omega_0 t + \theta_d(t)) \quad (3.1)$$

Onde S_d é o sinal modulado em fase, A e ω_0 são a amplitude e a freqüência angular da portadora e $\theta_d(t)$ é a modulação em fase de acordo com o dado a ser transmitido.

Sendo $c(t)$ a seqüência pseudo aleatória chaveando entre -1 e $+1$, obtém-se o sinal spread spectrum $S_t(t)$ [4]:

$$S_t(t) = A c(t) \cos(\omega_0 t + \theta_d(t)) \quad (3.2)$$

Esse é o sinal a ser transmitido por um caminho (na análise, sem distorção) que apresentará um atraso de propagação T_d . No lado receptor, aparecerá esse sinal junto com alguma forma de interferência e ruído do tipo AWGN que, como será visto mais adiante, serão minimizados na etapa de demodulação spread spectrum (desespalhamento).

Para que possa ser recuperado o sinal modulado em fase deve ser feita a multiplicação do sinal recebido por uma réplica da seqüência pseudo aleatória que inclua a estimativa do atraso de propagação T_d . O sinal na saída do bloco de “desespalhamento” fica da forma [4]:

$$S_{de}(t) = A c(t - T_d) c(t - T_{de}) \cos(\omega_0 t + \theta_d(t - T_d) + \phi) \quad (3.3)$$

Onde ϕ corresponde a uma fase aleatória do sinal S_{de} . Se a estimativa do atraso de propagação T_{de} estiver correta, o produto $c(t - T_d) c(t - T_{de})$ resultará sempre igual a 1, uma vez que esses sinais assumem apenas os valores +1 e -1. Portanto $S_{de}(t)$ fica praticamente igual a $S_d(t)$ no lado do transmissor, apresentando apenas o atraso de propagação T_d e uma fase aleatória ϕ . Notar que $S_d(t)$ pode apresentar qualquer modulação em fase, não apenas BPSK. O sinal digital é obtido pela demodulação desse sinal de acordo com as técnicas anteriormente mencionadas.

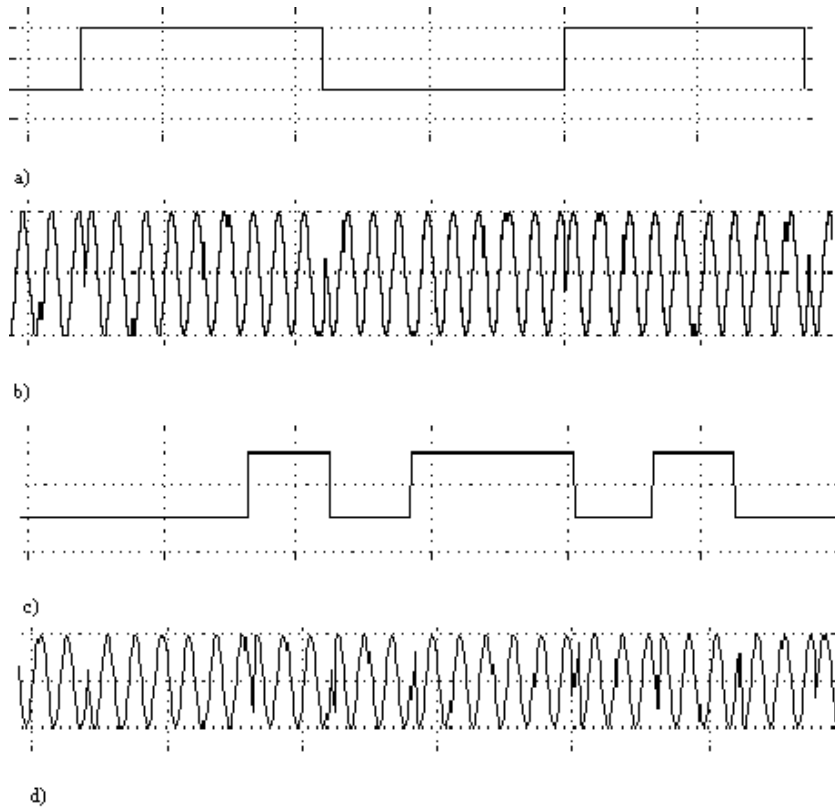


FIGURA 2: Sinais BPSK a) sinal digital b) sinal modulado em BPSK c) seqüência pseudoaleatória d) sinal spread spectrum DSSS

Antes de ser feita a análise do receptor quanto a presença de ruído e/ou interferência, será visto o espectro de potência do sinal spread spectrum.

O espectro do sinal spread spectrum é semelhante ao do sinal BPSK, apenas trocando o valor de T (duração de um bit) pelo tempo T_c (duração de um chip) e a equação está descrita abaixo [4]:

$$S_t(f) = \frac{1}{2} P T_c \{(\text{sinc}^2((f - f_0) T_c) + \text{sinc}^2((f + f_0) T_c))\} \quad (3.4)$$

Onde $\text{sinc}(x) = \text{sen}(x) / (x)$ e P é a potência do sinal S_t .

Como T_c é muito menor que T , a amplitude de potência é reduzida pelo fator T_c/T e a largura de banda $B = 2/T$ é ampliada pelo fator T/T_c (figura 3). No lado receptor, quando houver correlação entre $c(t - T_d)$ (embutida no sinal recebido) e $c(t - T_{de})$ (estimada) haverá o efeito contrário, a amplitude de potência será aumentada por T/T_c e o espectro será estreitado pelo fator T_c/T .

Quanto ao ruído e aos sinais interferentes, por não possuírem embutido neles a sequência pseudo aleatória a ser usada na etapa de desespalhamento, o fator $c(t - T_{de})$ não será cancelado, e esses sinais sofrerão a ação de espalhamento, reduzindo a potência por T_c/T e alargando o espectro por T/T_c . Notar que esse fator representa uma margem adicional ao receptor quanto ao ruído e interferência. Define-se Ganho de Processamento [4], essa relação:

$$G_p = 10 \log (T/T_c) \quad (3.5)$$

Outra forma de análise do efeito de espalhamento do espectro é através do teorema de Shannon [2]:

$$C = W \log_2 [1 + S/N] \quad (3.6)$$

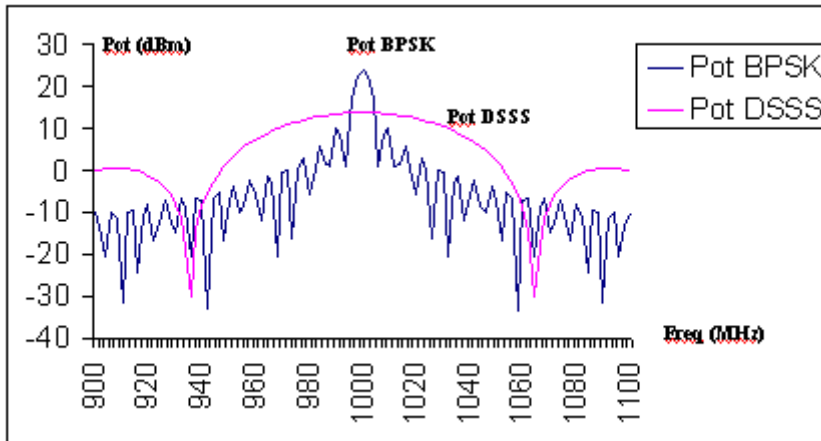


FIGURA 3: Espectro de potência para sinais BPSK e DSSS (Pot. em dBm e freq. em MHz)

Onde C é a taxa de bits máxima comportada pelo canal, W a largura espectral do sinal modulado e S/N a relação sinal ruído.

Como pode ser visto, para que se possa transmitir uma determinada taxa de bits em um meio suscetível a ruído e/ou interferência, pode ser feito o aumento da potência do sinal S ou o aumento da largura espectral W . Porém, o aumento da potência não produzirá grandes efeitos na taxa de bits máxima por causa do termo \log_2 , sobrando, portanto, o alargamento do espectro do canal. Se for suficientemente alargado o espectro pode-se transmitir uma determinada taxa de bits até mesmo em um meio com relação sinal/ruído menor que a unidade (potência de ruído maior que a de sinal).

3.2.2 SEQUÊNCIA DIRETA COM MODULAÇÃO QPSK

Embora o melhor aproveitamento do espectro disponível não seja a principal preocupação em sistemas spread spectrum, às vezes é interessante utilizar uma modulação mais eficiente para minimizar a probabilidade de detecção de sinais por outros receptores (aplicações militares). O sinal com espalhamento espectral em QPSK será equacionado como a seguir [4]:

$$S_t(t) = A c_1(t) \cos(\omega_0 t + \theta_d(t)) - A c_2(t) \sin(\omega_0 t + \theta_d(t)) \quad (3.7)$$

Onde $c_1(t)$ e $c_2(t)$ podem ser seqüências totalmente diferentes entre si, porém devem estar em quadratura de fase.

Essa técnica é utilizada também no sistema celular CDMA para os canais *forward* (estação rádio base para móvel) e *reverse* (móvel para estação rádio base).

3.3 ANÁLISE DOS BLOCOS DE UM SISTEMA SPREAD SPECTRUM

3.3.1 TRANSMISSOR SPREAD SPECTRUM BPSK

Pelo que foi visto até aqui, para implementar um circuito transmissor spread spectrum precisa-se dos seguintes blocos:

- a) Gerador de Seqüência Pseudo Aleatória:** é um circuito digital composto de vários estágios registradores de deslocamento (*shift registers*) com realimentação de determinadas saídas para a entrada através de uma lógica OR – exclusiva. O número de estágios (n) e as saídas a serem realimentadas são determinadas de acordo com o código da seqüência pseudo aleatória desejado. O número de estados possível é $2^n - 1$, uma vez que o estado com todas as saídas em zero não pode ocorrer para não travar o gerador. Os estados de saída variam entre -1 e $+1$ após a codificação de linha. Para que esse circuito gere todas as possíveis combinações de estados de saída, as realimentações devem ser tais que o menor valor de N para que exista a divisão de $x^N + 1$ por um polinômio primitivo $g(x)$ (formado pela soma das saídas realimentadas mais a unidade x^0) é $N=2^n - 1$ (número de estados possíveis). Significa que existe uma condição inicial que resultará em um ciclo de período N e que existe apenas um possível ciclo. Por exemplo, na figura 4 tem-se uma cadeia de 4 registradores de

deslocamento realimentada pelas saídas 1 e 4 ($g(x) = x^4 + x + 1$), onde 4 é o grau n . O número de estados possíveis é $2^4 - 1 = 15$, voltando em seguida à condição inicial. Esses estados são os seguintes, considerando a condição inicial 1111 e a forma g_4, g_3, g_2, g_1 onde g_1 é a primeira saída depois da realimentação:

TABELA 2: Estados possíveis para o gerador de sequência pseudoaleatória da figura 4

N	g_4	g_3	g_2	g_1
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	0
5	0	1	0	1
6	1	0	1	1
7	0	1	1	0
8	1	1	0	0
9	1	0	0	1
10	0	0	1	0
11	0	1	0	0
12	1	0	0	0
13	0	0	0	1
14	0	0	1	1
15	0	1	1	1
1	1	1	1	1

Conforme visto na tabela 2, após o décimo quinto estado as saídas voltam à condição inicial 1111. Essas seqüências com o máximo período dado por N são chamadas seqüências de máximo comprimento [4]. Para cada condição inicial resultará em uma diferente fase da mesma seqüência de máximo comprimento.

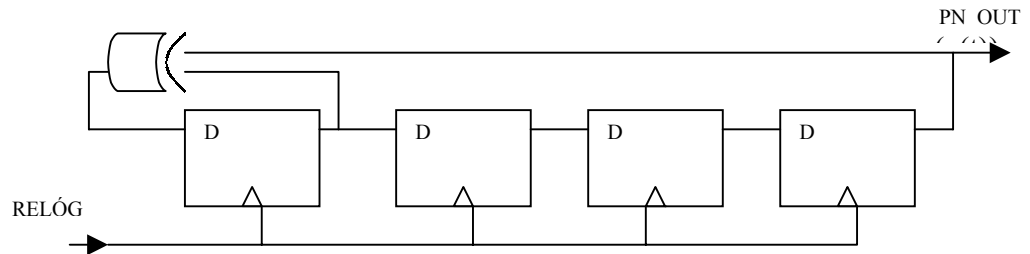


FIGURA 4: Diagrama de blocos do gerador de seqüências pseudo aleatórias.

A maior dificuldade ocorre quando deseja-se achar o polinômio (e conseqüentemente as realimentações necessárias) para produzir seqüências de comprimento máximo para um determinado número de estágios desejado. Felizmente existem tabelas, como as listadas na referência [4], com esses polinômios listados para cada grau, conforme parte dessas tabelas reproduzida na tabela 3:

TABELA 3: Polinômios de sequencia de máximo comprimento por grau desejado

Grau	Representação Octal do Polinômio com g_0 à direita e g_n mais à esquerda
4	[23]
5	[45] , [75], [67]
6	[103], [147], [155]
7	[211], [217], 235], [367], [277], [325], [203]. [313], [345]

Conforme visto no exemplo, para quatro estágios tem-se $n=4$, e o polinômio gerador é obtido a partir de [23], de acordo com a tabela acima. Decompondo em octal tem-se 010 011, como g_0 é o primeiro 1 a partir da direita, as saídas g_1 e g_4 serão usadas para a realimentação, como visto na figura 4.

Quando estão sendo usados múltiplos sistemas de acesso na mesma frequência e tempo, os códigos escolhidos para cada unidade de transmissão/recepção devem possuir a mais baixa correlação entre si, para que os receptores não façam o despalhamento da fonte de transmissão errada causando interferência. Existe um conjunto de tais códigos de baixa correlação entre si chamados códigos Gold, também tabelados[4]. A seguir está apresentada a definição de correlação entre duas seqüências pseudo aleatórias [4].

A correlação (θ) entre duas seqüências pseudo aleatórias b e b' pode ser definida por:

$$\theta_{bb'}(k) = 1/N \sum_{n=0}^{N-1} a_n a'_{n+k} \quad (3.8)$$

onde N é o número de estados possíveis, a_n e a'_{n+k} são os valores de saída para cada relógio nas duas seqüências (+1 ou -1) e k corresponde ao defasamento entre as duas seqüências.

Quanto mais próximo de +1,0 estiver o valor calculado da correlação para determinada defasagem, mais os sinais estarão correlacionados, e haverá risco de interferência de uma seqüência em outra na etapa de despalhamento do receptor.

Para o cálculo da auto correlação de uma seqüência PN basta fazer a_n igual a a'_n na equação 2.8. Por exemplo, tomando-se a seqüência gerada na tabela 2, substituindo 1 por +1 e 0 por -1 será obtido para $k=0$ (sem defasagem) correlação igual a +1,0. Para os demais valores de k será obtida a mesma auto correlação de -1/15. Esses resultados confirmam uma das propriedades das seqüências de máximo comprimento que estão listadas parcialmente na tabela 3, ou seja, haverá apenas correlação máxima para as seqüências PN em fase. Qualquer defasagem resultará em uma correlação mais baixa e constante para todas as possíveis defasagens. Essa propriedade é importante para aumentar a imunidade do receptor à chegada de cópias do sinal principal com atraso de fase provocado por efeitos de multi percurso (multipath) da onda eletromagnética transmitida.

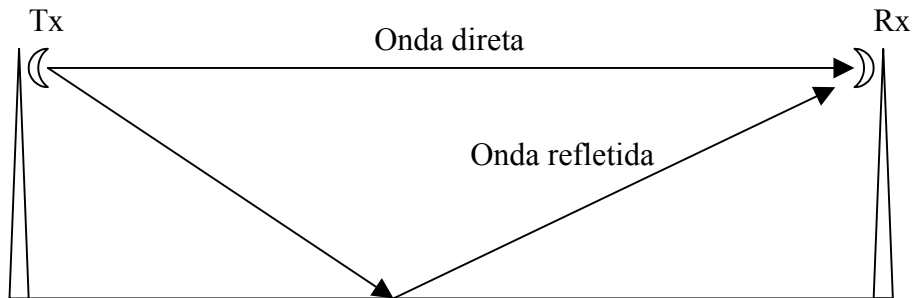


FIGURA 5: Exemplo de enlace com efeitos de multi percurso (reflexão).

- b) **Misturador Spread Spectrum:** é um circuito que faz a multiplicação do sinal modulado $S_d(t)$ pela seqüência pseudo aleatória $c(t)$. Pode ser implementado por uma ponte de diodos cuja polarização varia de acordo com a seqüência pseudo aleatória. Esses componentes são encontrados no mercado já encapsulados como dispositivos DBM (Double Balanced Mixers) e sintonizados para determinados valores de FI e freqüência de RF.

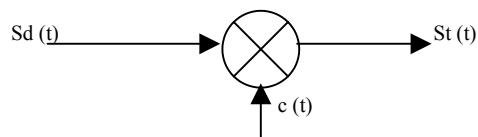


FIGURA 6: Diagrama de blocos do misturador Spread Spectrum.

- c) **Modulador em Fase:** é o circuito que modulará o sinal digital a ser transmitido em fase, podendo ser BPSK, QPSK, etc. Se for usada a modulação BPSK, por facilidade pode-se fazer a soma OR exclusiva da seqüência pseudo aleatória com o sinal digital a ser transmitido antes do estágio misturador Spread Spectrum. A entrada desse estágio deve ser compatibilizada com a interface de dados do usuário, como por exemplo: RS232, V.35, G.703, Ethernet 10BaseT, etc.

- d) **Circuitos de FI e RF:** uma vez que o processamento de espalhamento espectral é feito em FI (Frequência Intermediária), o sinal $S_t(t)$ deve passar por estágios de filtragem passa banda em FI, com a largura suficiente para o espalhamento, e então ser feita a conversão de frequência para cima pelo misturador de RF. A saída do misturador já estará na frequência apropriada para transmissão pelo canal espacial, após as etapas de filtragem passa banda, amplificação de RF, linha de transmissão (cabo coaxial ou guia de onda) e antena (tipicamente uma parabólica para enlaces de microondas Spread Spectrum, ou Omnidirecional/Setorizada para sistemas ponto-multiponto de acesso Spread Spectrum, tipo dados ou celular CDMA).

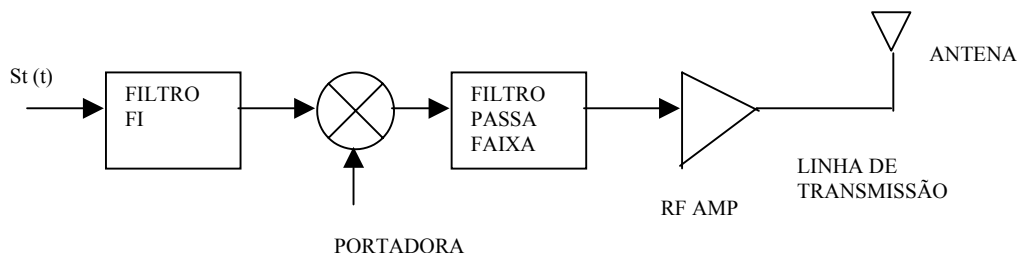


FIGURA 7: Diagrama de blocos da etapa de FI e RF.

3.3.2 RECEPTOR SPREAD SPECTRUM BPSK

Pelo que foi visto até aqui, para implementar um circuito receptor Spread Spectrum precisa-se dos seguintes blocos:

- a) **Circuitos de RF e FI:** o sinal é captado por uma antena (parabólica ou omni/setorizada), passa pela linha de transmissão (guia ou coaxial) até o estágio de entrada de amplificação de baixo ruído (LNA), cujo ganho é controlado pelo bloco de controle automático de ganho (CAG). Após essa etapa, o sinal recebido é convertido para baixa frequência por um misturador de RF, gerando um sinal em FI. Esse sinal FI passa por uma filtragem passa banda antes de entrar no bloco de desespalhamento.

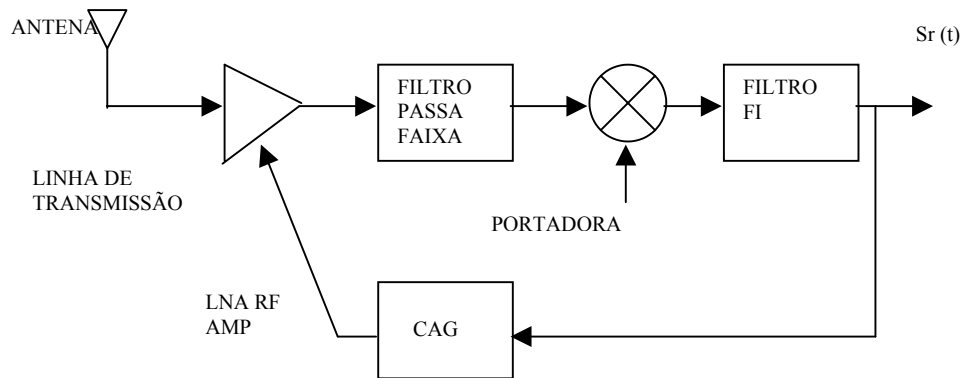


FIGURA 8: Diagrama de blocos da etapa de FI e RF do receptor.

- b) **Misturador Spread Spectrum:** esse estágio fará a multiplicação do sinal recebido em FI pela seqüência pseudo aleatória. Essa seqüência deve ser gerada pelo mesmo tipo de circuito do transmissor com as mesmas realimentações programadas. Também deve ser estimada a fase da seqüência pseudo aleatória embutida no sinal recebido para a recuperação total do sinal modulado $S_{de}(t)$, conforme ítem c).
- c) **Bloco de Temporização e Sincronismo:** a entrada desse bloco é o sinal recebido em FI, estando em uma malha de detecção de atraso de fase (Delay Locked Loop, por exemplo). Dessa forma a seqüência pseudo aleatória (PN) gerada por esse bloco já conterá o atraso de fase estimado e a freqüência correta. Também é feita a recuperação da portadora, para posterior demodulação BPSK ou QPSK.

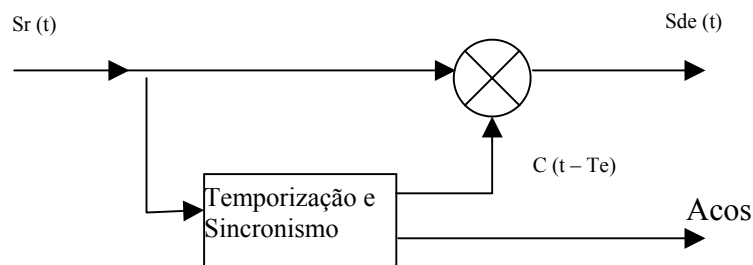


FIGURA 9: Diagrama de blocos das etapas misturadora e de temporização/sincronismo.

No receptor, essa é a parte mais complexa de um sistema de comunicação Spread Spectrum. A complexidade aparece devido à necessidade de ser feita a correlação do sinal recebido (somado a ruído e/ou interferência mais efeitos de multipath) com uma réplica da seqüência pseudoaleatória embutida nesse sinal com a mesma freqüência e fase. Existem duas etapas nesse processo: a primeira consiste na aquisição da fase correta da seqüência PN a partir do sinal recebido, e a segunda na manutenção do sincronismo de fase da seqüência PN após a aquisição inicial. Os circuitos que realizam essas tarefas (bloco de temporização e sincronismo) são chamados respectivamente de inicialização/aquisição de sincronismo e malhas de rastreamento de código (code tracking loops).

Inicialmente, pode-se analisar a etapa de aquisição de sincronismo de fase da seqüência PN no receptor. Existem várias formas de se obter o sincronismo a partir da seqüência PN embutida no sinal recebido, sendo que o tempo necessário para aquisição é o fator limitante. Por exemplo, em sistemas de múltiplo acesso, em que o usuário não irá usar o canal permanentemente, é importante que esse tempo de aquisição seja o mais curto possível.

A primeira maneira seria fazer uma busca serial da fase correta, variando-se a saída do gerador PN rapidamente até que a seqüência reproduzida localmente provocasse a correlação com o sinal recebido. Essa correlação é percebida pelo aumento da potência de sinal na saída do bloco correlacionador. Porém, o tempo para se atingir a condição de sincronismo pode ser demasiado, nos casos em que se usa códigos PN mais longos.

Para acelerar esse tempo de aquisição, pode-se usar um filtro casado (*matched filter*) com a envoltória do sinal recebido espalhado por uma seqüência PN conhecida. Quando chegar na entrada do filtro o sinal com a seqüência correta, será gerado um sinal de gatilho (trigger) para o gerador PN local do receptor. O filtro casado pode ser implementado de forma analógica ou digital, e até mesmo para inicialização em banda básica, como mostrado na figura 10. Nesse circuito é feita a comparação bit a bit da seqüência recebida com a seqüência PN gravada no receptor, gerando o valor mais baixo possível no somador digital de saída quando há correlação, pois os blocos OR exclusivo produzem saída zero para entradas iguais.

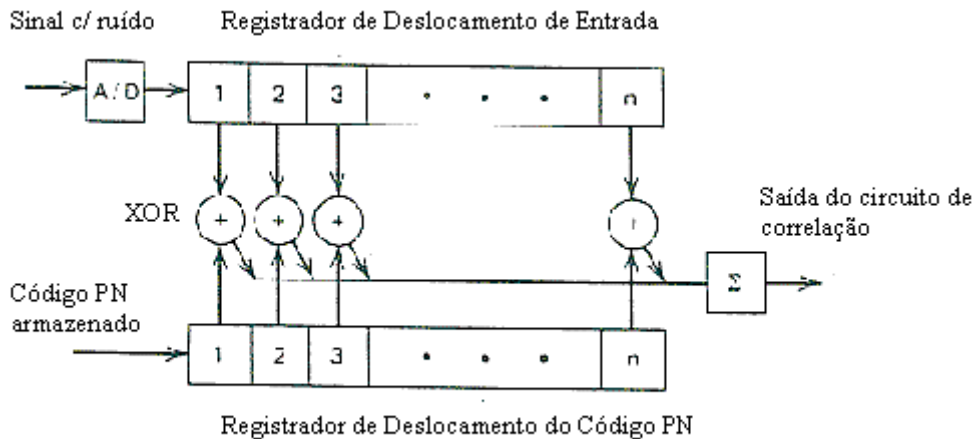


FIGURA 10: Diagrama de blocos de um filtro casado digital

Outra forma é a técnica chamada **RASE (Rapid Acquisition by Sequential Estimation)**. Essa técnica é implementada por um circuito mais simplificado, operando na banda base (digital). Consiste na carga dos registradores de deslocamento utilizados para gerar a seqüência PN com uma amostra do sinal recebido já demodulado. Uma vez que o sinal espalhado espectralmente em seqüência direta contém muitos bits da seqüência PN desejada, há uma grande possibilidade de o gerador de seqüência PN local inicializar na fase correta. Enquanto não houver a correlação, é repetido o processo. A vantagem dessa técnica é a velocidade de aquisição, porém exige uma relação sinal/ruído mais alta que as anteriores. Essa é a técnica utilizada para simulação do sistema Spread Spectrum no MatLab (ver capítulo 4).

Mesmo após a aquisição da fase correta para o gerador de seqüência PN local, existe a necessidade da manutenção do sincronismo com o transmissor. Isso ocorre devido ao atraso de propagação T_d poder ser variável, como por exemplo nos dispositivos móveis. As modulações em fase são as mais atingidas por esse efeito, assim como pelos efeitos de multipercurso, devido à informação ser obtida pela variação da fase. Quando ocorre o atraso ou adianto da fase do sinal recebido, o receptor já sincronizado, tende a manter a seqüência

PN na mesma fase do momento da aquisição. Se houver o escorregamento de pelo menos um chip, não haverá a correlação máxima e os dados desejados serão corrompidos. Para que o gerador de seqüência PN acompanhe a variação de fase do sinal recebido existem os circuitos de malhas de rastreamento de código (code tracking loop). O funcionamento desse circuito é muito semelhante ao de um PLL (Phase Locked Loop) [4]. Porém, o sinal recebido na entrada é comparado por dois blocos de correlação com versões anteriores (early) e posteriores (late) da seqüência PN. Se ocorrer tendência de atraso, por exemplo, a saída do comparador early irá apresentar uma média menor que a saída do comparador late, havendo o desequilíbrio da saída do amplificador diferencial para tensão menor. Uma diminuição de tensão fará com que a frequência do VCO diminua e a fase da seqüência PN irá se atrasar, alcançando a fase do sinal recebido. Pode-se ter essas malhas operando em banda básica ou então com o sinal recebido ainda não demodulado, como visto na figura 11. A vantagem da última alternativa é que existe uma maior tolerância à interferência e/ou ruído que na solução por banda básica. Todo esse processo é feito para atrasos ou adiantamentos de $\frac{1}{2}$ chip, pois se permitisse a variação de pelo menos 1 chip, haveria perda de sincronismo, e o receptor poderia entrar novamente no processo de aquisição. Deve ser lembrado que o código PN a ser usado deve ter baixa autocorrelação, portanto, a média do sinal na saída do correlacionador principal, se houvesse a variação de pelo menos um chip, iria ter um nível de potência muito baixo para a posterior demodulação dos dados do usuário.

Na referência [4] é mostrado que o sistema apresentado acima pode ser modelado da mesma forma que um sistema PLL, ficando a função de transferência no domínio s da forma:

$$H(s) = Tde(s) / Td(s) = K_d G_c F(s) / (s + K_d G_c F(s)) \quad (3.9)$$

Onde: $Tde(s)$: delay estimado

$Td(s)$: delay entre transmissor e receptor

K_d : ganho combinado dos blocos de correlação

G_c : ganho do VCO

$F(s)$: função de transferência do filtro Passa Baixa

Investigando a função $H(s)$ conclui-se que no regime permanente os delays estimado Tde e real Td são iguais. Porém, no projeto dos ganhos do VCO e da função do filtro deve-se ter cuidado quanto aos transientes (overshoot e fator de amortecimento) como em qualquer

projeto de sistemas realimentados. O ganho K_d depende do tipo de modulação digital usada para espalhamento do sinal.

A seguir está um exemplo de circuito de Code Tracking Loop:

- **DLL (Delay Locked Loop):** uma das técnicas mais utilizadas para manutenção de sincronismo PN após a aquisição inicial, consiste em uma malha formada por um VCO gerando o relógio do gerador de seqüência pseudoaleatória controlado por uma tensão obtida pela diferença entre sinais correlacionados com a seqüência anterior (early) e a posterior (late). Com a malha travada (locked), a freqüência de relógio corresponderá à freqüência usada no transmissor para espalhamento e a saída do circuito de correlação com a seqüência atual (on time) corresponderá ao trem de dados desejado. Na figura 11 está o diagrama de blocos.

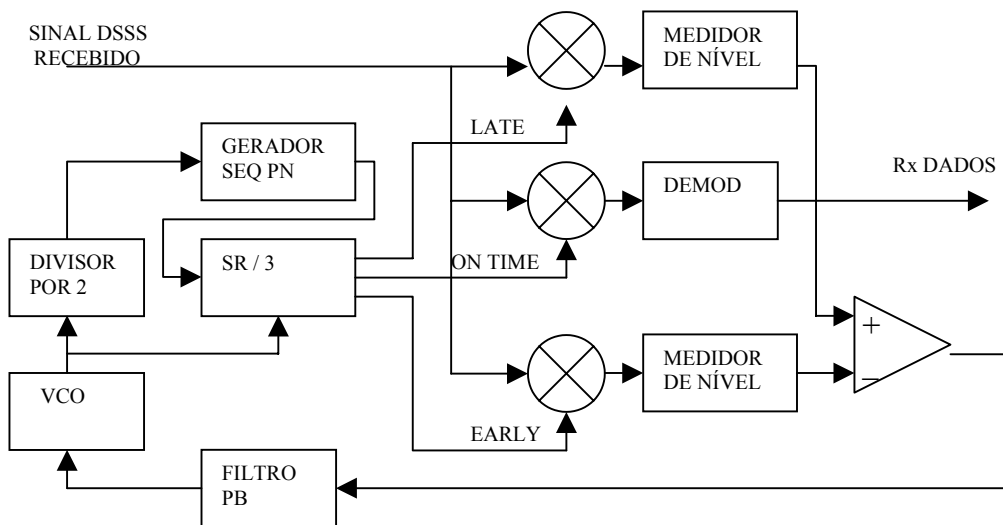


FIGURA 11: Diagrama de blocos do circuito DLL

- d) **Demodulador:** a demodulação em fase pode ser feita pela multiplicação do sinal modulado $S_d(t)$ pela portadora recuperada pelo bloco anterior ($A \cos \omega_c t$). Após a filtragem dos elementos de alta freqüência, tem-se o sinal em banda básica. Para a recuperação do sinal digital, esse sinal deve passar por um circuito decisor (comparador de tensão), sendo feita então a compatibilização com o tipo de interface

do usuário. Se for usada a modulação DBPSK ou DQPSK não há a necessidade de recuperação da fase da portadora.

4 SIMULAÇÃO

4.1 SISTEMA PROPOSTO PARA SIMULAÇÃO

Como já visto na introdução, o objetivo desse trabalho é a implementação de um sistema Spread Spectrum dedicado de baixo custo para comunicações de dados de pequenas distâncias. Porém, antes da implementação física, e também para obter mais conhecimento dessa técnica de modulação é necessário fazer uma simulação de um sistema DSSS a ser proposto. Duas perguntas aparecem: quais os blocos e parâmetros seriam mais interessantes e qual a estratégia de simulação?

A resposta à primeira pergunta: deve ser um sistema transmissor - receptor com interface do tipo RS232 (comum em microcomputadores PC e em uma infinidade de equipamentos) e permitir uma forma de múltiplo acesso por divisão de código. Também deve funcionar por rajadas de dados (bursts), portanto com tempo de aquisição de sincronismo de código o mais curto possível. A FI deve utilizar filtros padronizados, do tipo dos utilizados em TV (45MHz) de fácil aquisição e baixo custo.

Quanto à segunda pergunta, a simulação deve levar em conta a presença de ruído, sinais de mesma frequência interferindo com outro código PN e efeitos de multipercurso. As potências de ruído e interferência devem ser levadas ao limite de forma a aparecer dentro do intervalo de simulação e com os dados e demais sinais com facilidade de visualização nos gráficos gerados.

Todos os circuitos propostos serão desenhados e simulados dentro do módulo Simulink do MatLab. Nesse módulo pode-se escolher blocos já prontos tipo portas lógicas, flip flops, fontes senoidais, fontes de ruído AWGN, relógios, osciloscópios e analisadores de espectro. Na figura 12 está o diagrama de blocos para simulação.

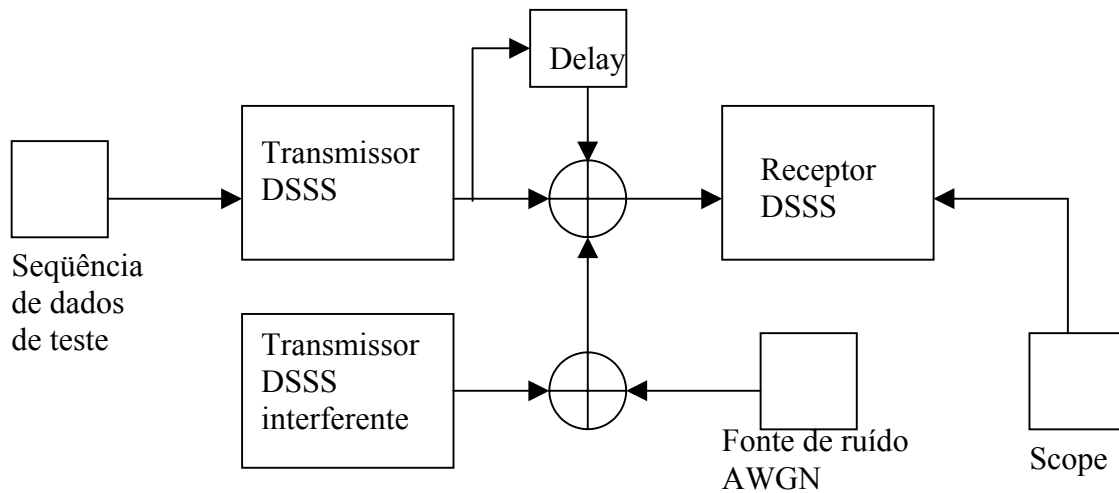


FIGURA 12: Diagrama de blocos proposto para simulação MatLab

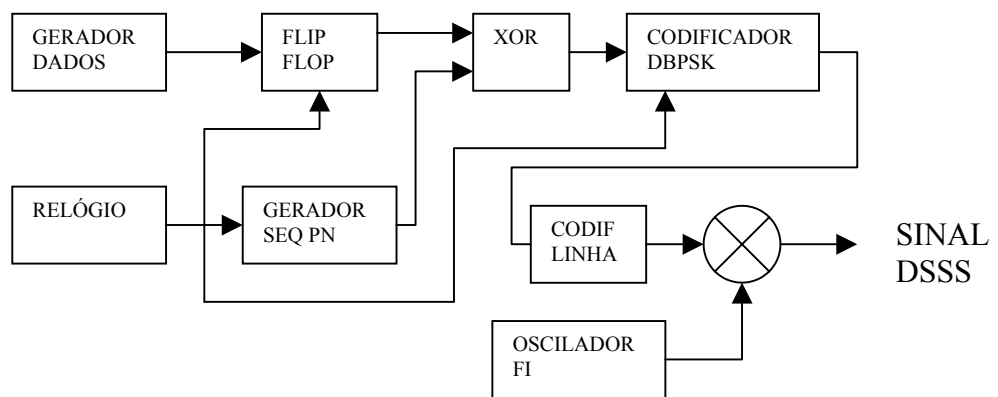
4.1.1 TRANSMISSOR

O transmissor foi implementado de acordo com o diagrama de blocos da figura 13, sendo utilizado um bloco de 4 estágios registradores de deslocamento para gerar a sequência de dados de teste da forma $x^4 + x + 1$. O gerador de sequência pseudo aleatória é constituído por 7 estágios registradores de deslocamento com a forma $x^7 + x + 1$ (a partir de [203] da tabela 3). Para a simulação foram normalizadas as taxas de transmissão de dados e frequência de portadora por um fator de divisão por 10^5 . A taxa de dados a transmitir é de 115,2 kbps (máxima velocidade RS232 em um microcomputador PC), a taxa de chips (taxa de espalhamento) é de 1.152 kbps (para obter $G_p=10\text{dB}$) e a frequência de portadora é de 45MHz. Como o objetivo da simulação é o teste da parte de espalhamento espectral, a portadora foi mantida na frequência intermediária (FI). Uma vez que a técnica de modulação empregada para a primeira etapa de modulação e para o bloco de espalhamento espectral é a mesma, DBPSK, pode-se fazer o espalhamento do sinal de dados (sincronizado com o gerador de sequência pseudo aleatória através de um flip flop com o mesmo sinal de relógio desse) através da lógica XOR com a sequência PN. A saída da XOR é aplicada ao codificador DBPSK, com saída de acordo com a tabela 4.

TABELA 4: Codificação DBPSK

Dado	Delta Fase
0	0°
1	180°

Dado	Fase atual	Fase seguinte
0	0	0
0	1	1
1	0	1
1	1	0

**FIGURA 13:** Diagrama de blocos do transmissor DSSS/DBPSK

4.1.2 RECEPTOR

Na figura 14 está o diagrama de blocos do receptor com os principais sinais utilizados. A técnica de aquisição é o método RASE já visto no ítem 3.3.2. O circuito recebe o sinal em FI e faz o batimento com o oscilador local, passando por um filtro passa baixa e aplicado a um comparador que gera bit 1 para entrada positiva e bit 0 para negativa. No bloco decodificador DBPSK é feita a decodificação DBPSK (de acordo com a tabela 5). Esse sinal corresponde ao sinal DSSS em banda básica, sendo usado para a regeneração do relógio de chip pelo circuito PLL (na simulação, para obter maior velocidade foi usado um filtro passa banda de alto fator de qualidade (Q) sintonizado na frequência de relógio de chip, (1,152MHz)).

Para a aquisição de sincronismo de código pelo método RASE foi implementada uma máquina de estados para controle da carga dos registradores de deslocamento e detecção da correlação do sinal de saída. Quando o circuito de correlação receber a entrada Trigger em zero (condição de não correlação entre a seqüência gerada localmente e a recebida), a máquina de estados dispara a contagem com duração de 8 relógios de chips. Neste período a saída Load permite a carga dos registradores de deslocamento (SRs) do gerador de seqüência PN como sinal de entrada (que contém a seqüência PN desejada embutida). Ao mesmo tempo o sinal recebido é atrasado em 8 relógios de chip por uma cadeia de 8 SRs. Depois a saída Clear é acionada para limpar o circuito detetor de correlação bem sucedida. O circuito detetor mantém a saída Trigger em nível alto por 8 relógios de chip para que seja testada a configuração carregada anteriormente no gerador PN; se não estiver correta, o circuito detetor fará a contagem de pulsos na saída da XOR de correlação. Se estiver correlacionado, não deve haver mais que uma transição nesse período, lembrando que a duração de um bit deve corresponder a 10 chips. Pulsando mais de 2 vezes, a saída Trigger é levada ao nível zero para que recomece tudo de novo até haver a correlação.

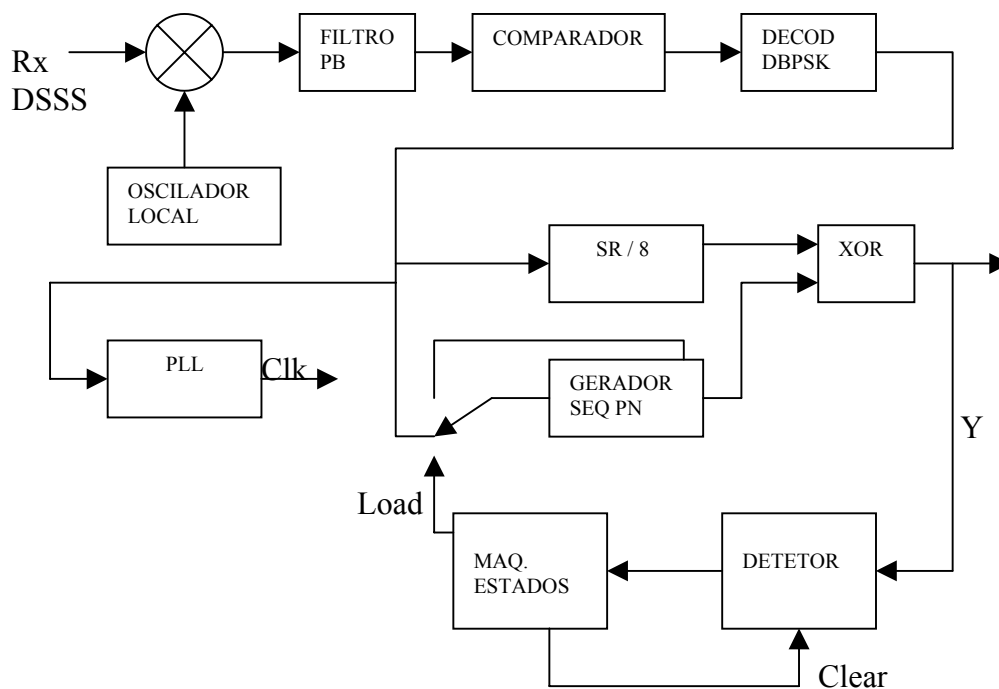


FIGURA 14: Diagrama de blocos do receptor DSSS/DBPSK (RASE)

TABELA 5: Decodificação DBPSK

Fase atual	Fase ant	Delta	Dado
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Agora deve ser feita a descrição do projeto da máquina de estados e do detetor de correlação. O diagrama a seguir mostra os estados possíveis da máquina de estados RASE de acordo com o já exposto acima.

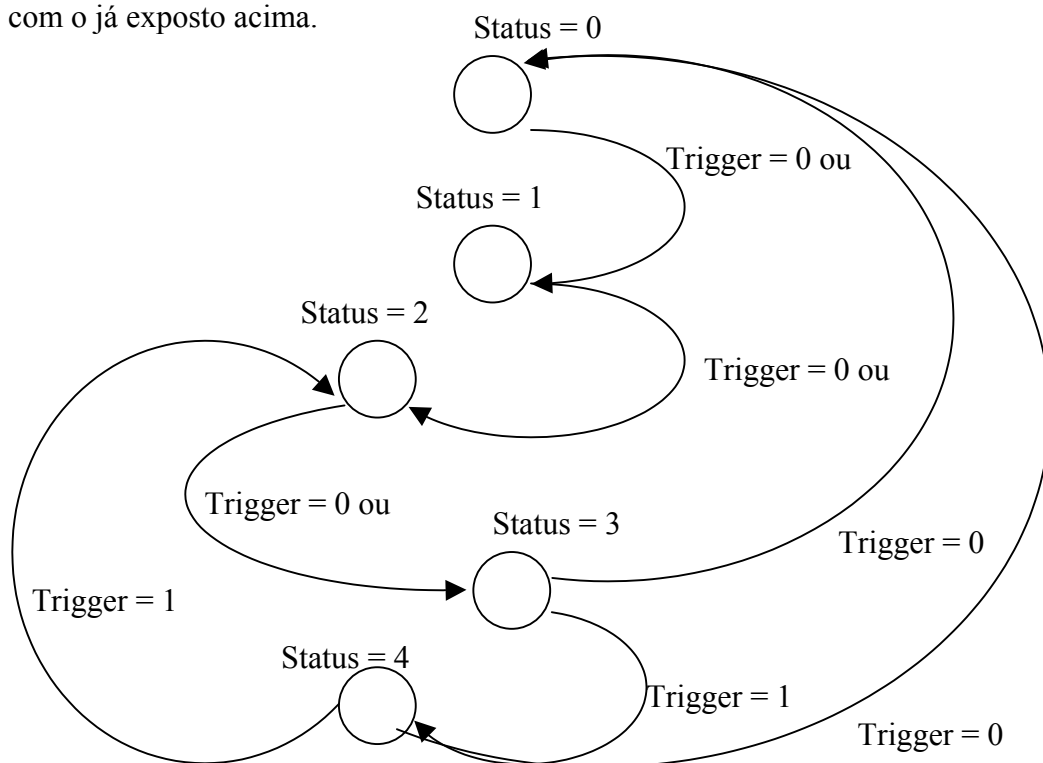


FIGURA 15: Diagrama de estados da Máquina de Estados RASE

TABELA 6: Sinais Load e Clear codificados por estado

Status	Load	Clear
0	1	0
1	1	0
2	0	1
3	0	0
4	0	0

Para simplificação da máquina de estados foi admitido que o relógio dos flip flops tipo D seria o resultado da divisão por 4 do relógio de chip. Dessa forma não é necessária a contagem de 8 pulsos de relógio para que a saída Load fique em nível alto para carga dos 8 estágios de registradores de deslocamento. Como visto acima, há a necessidade de apenas 2 estados para a carga dos registradores de deslocamento. Após essa carga, no estado 2 a saída Clear é ativada para reset dos contadores do detetor de correlação. Nos estados 3 e 4, a máquina de estados fica monitorando a entrada Trigger (saída do detetor de correlação), se zero volta ao estado inicial 0, se um, continua mudando de estado até voltar ao estado 2 (Clear ativo), repetindo o ciclo. Ou seja, quando há a condição de correlação, Trigger permanece igual a 1(um) fazendo com que a máquina de estados varie entre os estados 2, 3 e 4 (Clear mais dois estados para contagem de correlação por 8 chips).

A máquina de estados é implementada com 3 flip flops tipo D cujas entradas são geradas de acordo com as expressões lógicas abaixo:

$$D2 = \text{Trigger } Q1 \ Q0; \ D1 = \bar{Q}1Q0 + Q1\bar{Q}0 + \text{Trigger } Q2; \ D0 = \bar{Q}2 \ \bar{Q}1 \ \bar{Q}0 + Q1 \ \bar{Q}0$$

Onde Q2, Q1 e Q0 são a codificação binária para cada estado de 0 a 4.

As saídas Clear e Load são implementadas de acordo com as expressões lógicas abaixo:

$$\text{Clear} = \overline{Q1}Q0; \quad \text{Load} = \overline{Q2}\overline{Q1}$$

O detetor de correlação é outra máquina de estados que tem como entrada o próprio sinal correlacionado Y (dados do usuário) e, como saída, o sinal Trigger. O diagrama de estados pode ser visto na figura 16:

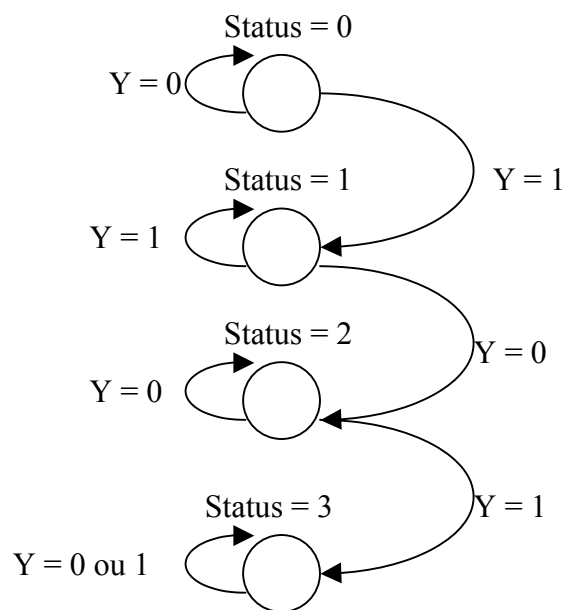


FIGURA 16: Diagrama de estados do detetor de correlação

O detetor de correlação é implementado com 2 flip flops tipo D com relógio a partir do relógio de chip recuperado. Basicamente, dentro da janela de 8 chips após os flip flops serem zerados por Clear gerado pela máquina anterior, o detetor faz a contagem de transições do sinal Y. Se superior a 2 transições, a saída Trigger é levada ao valor zero (status 3) fazendo com que a máquina de estados RASE faça a recarga dos registradores de deslocamento do gerador de seqüência PN. As expressões lógicas para as entradas D1 e D0 e também para a saída Trigger estão abaixo:

$$D1 = Q1 + \overline{Y} \overline{Q1} Q0; \quad D0 = Y + \overline{Y} Q1 Q0; \quad \text{Trigger} = \overline{Q1} Q0$$

Onde Q1 e Q0 são as saídas dos dois flip flops.

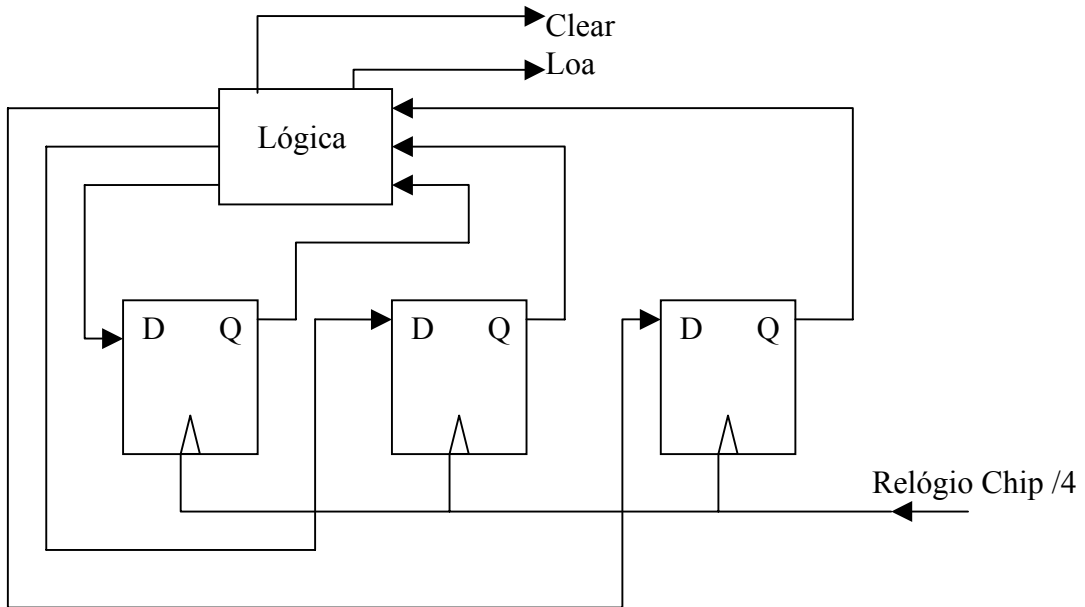


FIGURA 17: Circuito lógico da máquina de estados com Flip Flops tipo D

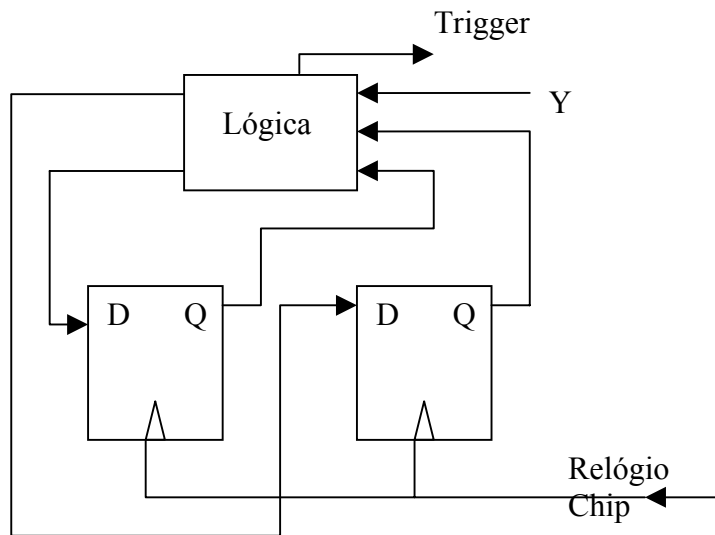


FIGURA 18: Circuito lógico do detetor de correlação

4.2 SIMULAÇÃO DO SISTEMA DSSS

Os circuitos transmissor e receptor DSSS foram implementados usando as funções do módulo Simulink do software MatLab. Foram considerados os casos em que o meio é o ar com a presença de ruído branco gaussiano (AWGN) e os efeitos de multipercurso, ou seja, a chegada no receptor de ondas refletidas que, por percorrerem outro percurso que não a linha reta entre transmissor e receptor, apresentam defasagem em relação ao sinal principal. Para modulações digitais, esse efeito é nocivo, podendo até anular o sinal principal se chegar uma onda refletida com mesmo nível e defasagem de 180° . O objetivo da simulação desse sistema DSSS é a comunicação entre dois dispositivos em uma pequena área, portanto é de se esperar que a diferença de percurso da onda direta e da onda refletida de caminho mais longo seja da ordem de dezenas de metros, 50 m por exemplo. Considera-se a velocidade de propagação próxima à velocidade da luz, é razoável trabalhar com um valor máximo de atraso entre a onda refletida e a direta de 0.17useg ($50\text{m} / 3 \times 10^8$). Na simulação foi feita a soma do sinal transmitido principal com uma réplica do mesmo defasada por um tempo que é um quarto da duração de um chip ($1/4 * 1,152\text{MHz} = 0.217\text{useg}$, excedendo o tempo máximo anterior) mais o ruído AWGN com quase 10dB acima da potência do sinal DSSS.

Através da curva BER x relação S/N para a modulação DBPSK da figura 19, espera-se que quando o nível recebido for igual ao nível de ruído AWGN ($S/N = 0\text{dB}$), tenha-se uma taxa de erros de cerca de 10^{-1} , ou seja, 1 bit errado a cada 10. Com o ganho de processamento de 10dB, pode-se simular o sistema com o nível de ruído com quase 10dB acima do sinal, que haverá demodulação na saída com probabilidade de erros próxima a 10^{-1} . Esses valores foram escolhidos para simular o sistema DSSS no limite para que os erros de bit aparecessem logo.

Através das simulações foi notado o aumento dos erros de bit quando a potência de ruído é aumentada para acima de 10dB acima do sinal DSSS, comprovando o ganho de processamento de 10dB. Na figura 22 tem-se os sinais de dados Tx e Rx para potência de ruído menor que 10dB acima e na figura 24, maior que 10dB.

Quanto à interferência, na figura 23 tem-se os resultados para um transmissor interferente com código PN da forma x^7+x^3+1 e na figura 25, com código PN da forma

$x^7+x^6+x^5+x^2+1$. Nas duas simulações, a potência do interferente estava no limite de 2dB abaixo do sinal DSSS principal (mínimo valor de interferência), que foi determinado empiricamente. Pode-se notar que, diferentemente da simulação com ruído AWGN, as potências dos sinais DSSS principal e interferente devem estar bem mais próximas, lembrando que os valores de relação S/N obtidas pela curva são o resultado do estudo do comportamento da modulação DBPSK em meio com ruído AWGN apenas. Outro aspecto é o comprimento do código usado que é relativamente pequeno (127 chips de período), sendo que quanto maior o comprimento da seqüência PN, mais parecido com ruído será o sinal interferente em relação ao principal.

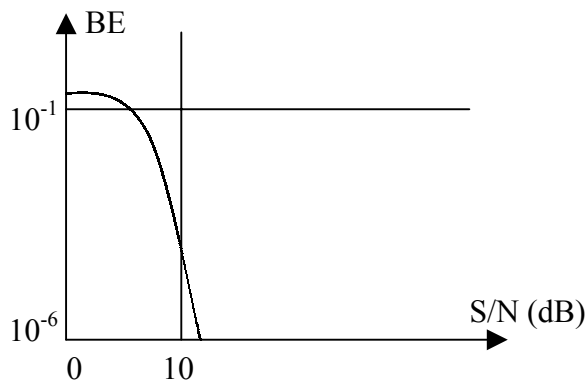


FIGURA 19: Curva BER x S/N para um sinal com modulação DBPSK em ambiente AWGN

O descrito no parágrafo acima é conhecido como problema near-far (próximo - distante) para o receptor de um sistema DSSS de múltiplo acesso. Em sistemas desse tipo (CDMA por exemplo) deve haver um controle rígido das potências de transmissão de forma que os sinais que chegam à antena receptora sejam da mesma ordem de grandeza independente da distância dos dispositivos transmissores. Ou seja, para transmissores distantes deve ser fixada potência de Tx mais alta do que para os próximos. Obedecida essa condição, os códigos PN de cada transmissor é que irão fazer a discriminação do sinal digital desejado. Como pode ser notado, com o uso de um código com menor correlação ($x^7+x^6+x^5+x^2+1$) com o usado no transmissor principal, obteve-se menor taxa de erros nos bits recebidos (ver figura 25).

A escolha dos códigos é uma questão importante em sistemas Spread Spectrum, uma vez que a possibilidade de múltiplos transmissores ocuparem a mesma banda depende do quanto os códigos PN de cada um sejam menos correlacionados entre si, isto é, o mais ortogonais possíveis, conforme visto no ítem 3.3.1 sobre o conjunto de códigos Gold.

A seguir está o resumo das conclusões obtidas da simulação:

- a) o uso da técnica RASE de aquisição de sincronismo de código apresentou realmente uma grande velocidade, atingindo a correlação em cerca de 80 chips (menos que os 127 chips de período da seqüência PN). Para o sistema proposto esse tempo seria de 69,4 useg ($80 \times (1/1.115.200)$ chips/seg) mesmo com ruído cerca de 10 dB acima do sinal recebido.
- b) Confirmado o ganho de processamento de 10dB com a simulação de ruído no limite máximo.
- c) Comprovação do problema near-far para receptores de múltiplo acesso. Os níveis dos sinais principal e interferente devem ser da mesma ordem de grandeza. O código usado nos dois sinais devem possuir a mais baixa correlação possível.
- d) O sistema apresentou imunidade à interferência de sinais refletidos com atraso de propagação compatível com as dimensões de ambientes esperados para a aplicação proposta.

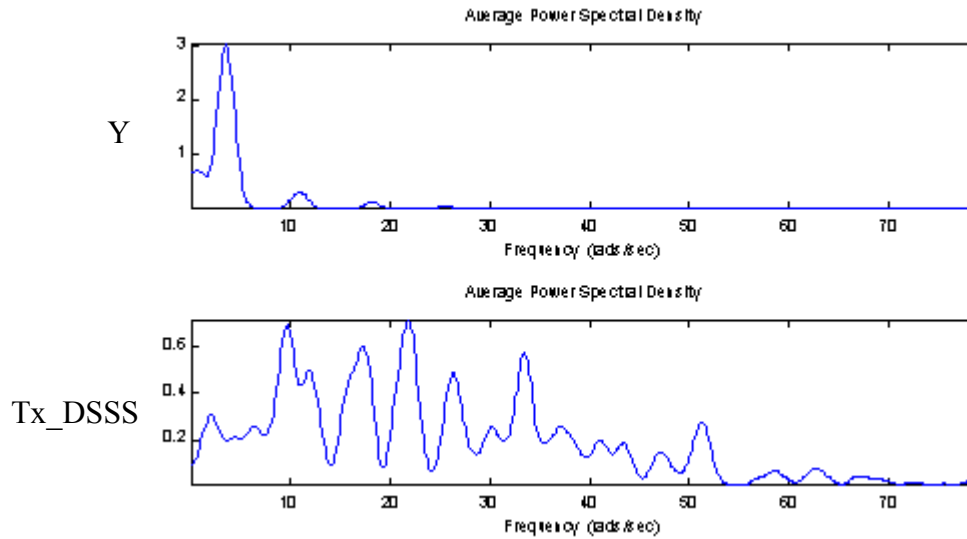


FIGURA 20: Espectro simulado dos sinais em banda base Y e espalhado Tx_DSSS

Nas figuras a seguir são apresentados os resultados para o sistema Tx-Rx: sem ruído (figura 21); com multipath e ruído (figura 22); com interferência na mesma frequência (45MHz) e código PN diferente (figura 23); com ruído acima de 10dB (figura 24) e com interferência de código PN com menor correlação com o principal (figura 25).

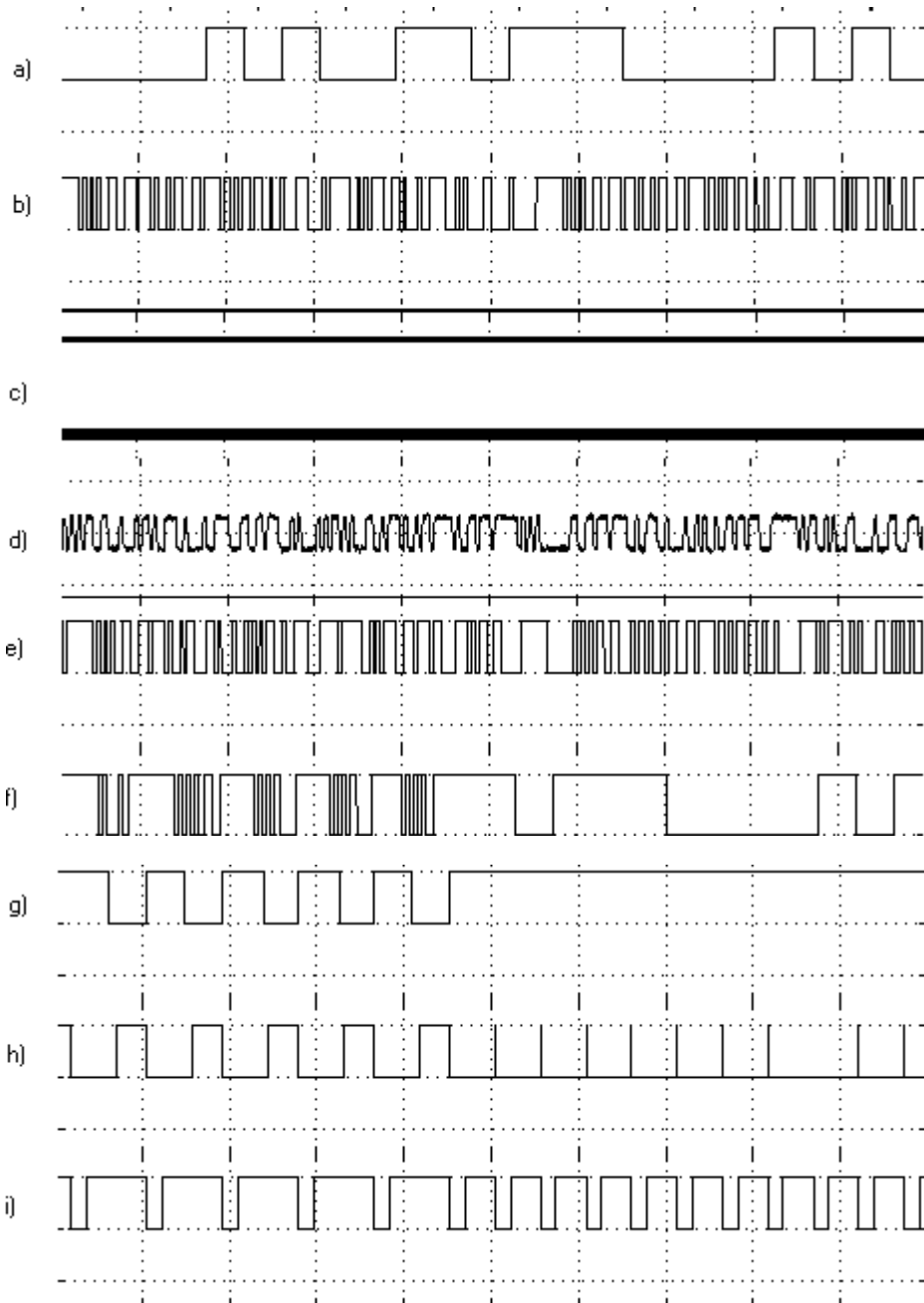


FIGURA 21: Sinais DSSS/DBPSK sem ruído a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) saída do filtro PB e) saída do decodificador DBPSK f) sinal digital correlacionado (Y) g) Trigger h) Load i) Clear

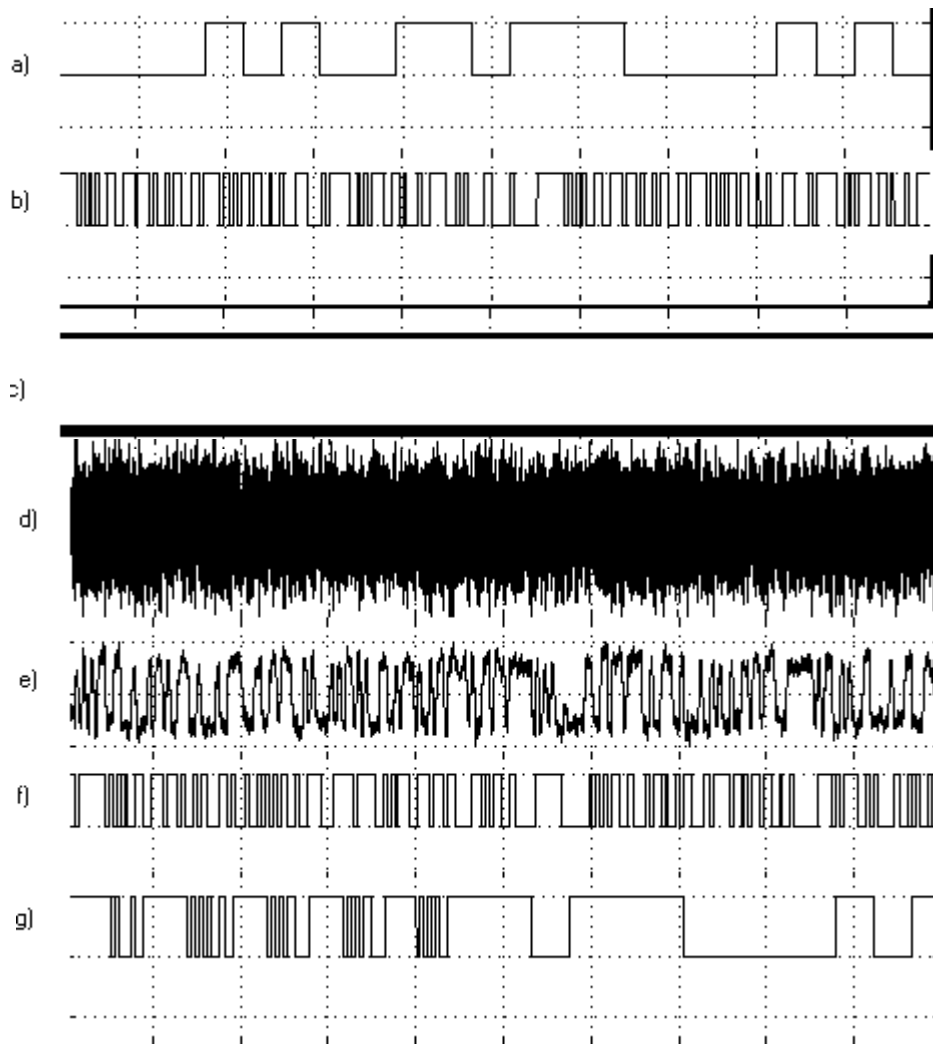


FIGURA 22: Sinais DSSS/DBPSK com multipercurso e ruído a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) sinal DSSS/DBPSK recebido c/ multipath e ruído e) saída do filtro PB f) saída do decodificador DBPSK g) sinal digital correlacionado (Y)

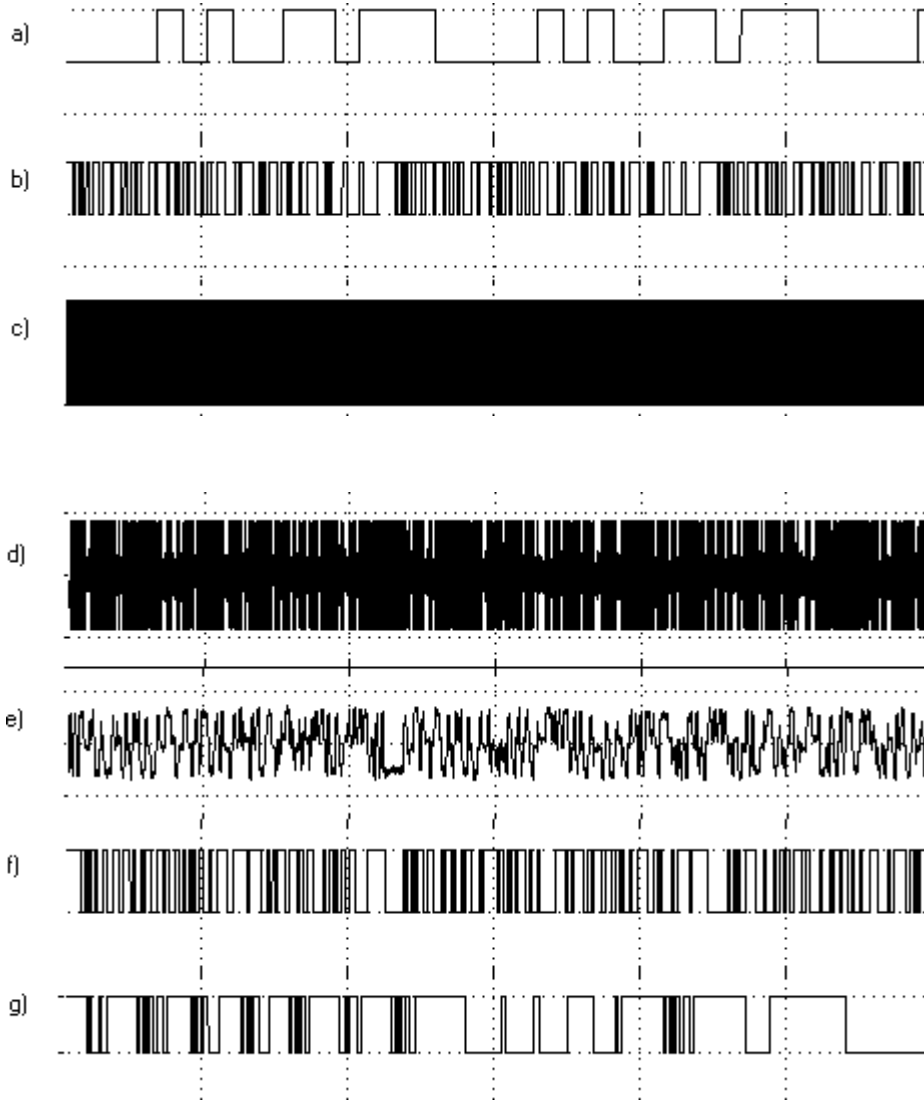


FIGURA 23: Sinais DSSS/DBPSK com interferência a) sinal digital b) seqüência pseudoaleatória c) sinal spread spectrum DSSS/DBPSK d) sinal DSSS/DBPSK recebido c/ interferência e) saída do filtro PB f) saída do decodificador DBPSK g) sinal digital correlacionado (Y)

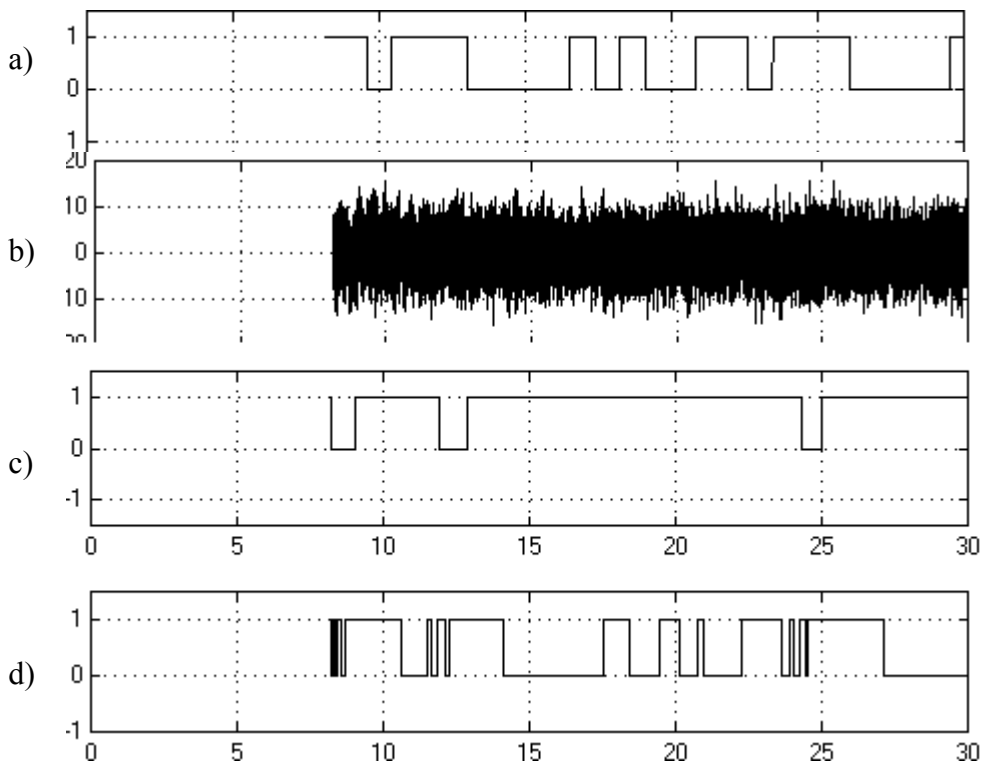


FIGURA 24: Sinais DSSS/DBPSK com ruído maior que 10dB acima a) sinal digital b) sinal DSSS/DBPSK recebido c/ ruído c) Trigger d) sinal digital correlacionado (Y)

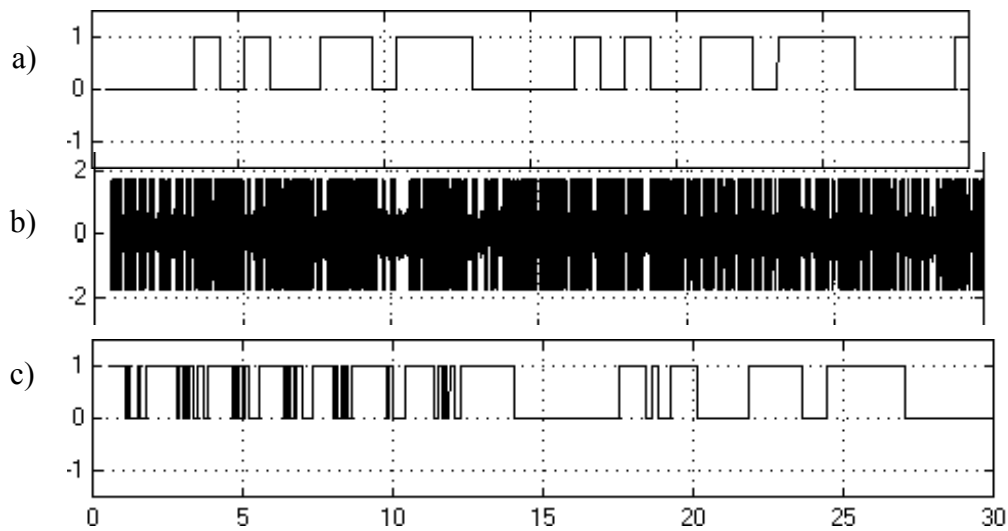


FIGURA 25: Sinais DSSS/DBPSK com interferência a) sinal digital b) sinal DSSS/DBPSK recebido c/ interferência c) sinal digital correlacionado (Y)

5 IMPLEMENTAÇÃO FÍSICA

5.1 CONSIDERAÇÕES INICIAIS

Para a implementação física de um sistema transmissor/receptor DSSS de baixo custo foram considerados os seguintes itens:

- O sistema deve permitir a comunicação entre dois computadores padrão PC via interface RS232 (permite velocidades de 600bps a 115200bps);
- O ganho de processamento deve ser da ordem de 10dB;
- Modulação DBPSK para facilitar a construção da parte moduladora de RF;
- O processamento digital deve ser feito por circuitos FPGA (Field Programmable Gate Array) ou microcontrolador para que o número de componentes fique reduzido;

Inicialmente foi considerada a hipótese de uso dos circuitos já simulados no MatLab para implementação do hardware, porém haveria a necessidade do uso de FPGAs (Field Programmable Gate Arrays). Devido à dificuldade de se obter uma amostra de FPGA e o programador para uso full time nesse projeto, foi reconsiderada a possibilidade de uso de um microcontrolador do tipo PIC da Microchip Technology Inc, mais barato e com maior facilidade de construção do circuito programador. Outra vantagem é o interfaceamento RS232, praticamente pronto nesses microcontroladores, assim como a possibilidade de programação em uma linguagem de nível mais alto, tipo C ou Basic, para posterior compilação para o código assembly. Como grande desvantagem, para se conseguir um ganho de processamento da ordem de 10dB, a velocidade de transferência de dados efetiva na RS232 deve ser reduzida, além da arquitetura de hardware ser modificada para implementação em software.

5.2 MICROCONTROLADOR PIC16F876

O microcontrolador PIC16F876 da Microchip Technology Inc. foi o componente usado para a implementação física devido a disponibilidade desse componente e por atender aos requisitos expostos. Antes de detalhar a construção do circuito e do programador, será dada uma visão resumida do funcionamento do microcontrolador PIC16F876 com maior ênfase às funções exploradas nessa implementação.

As principais características do PIC16F876 são [8]:

- CPU do tipo RISC;
- 35 instruções;
- Apenas um ciclo de instrução, exceto os desvios/saltos que usam 2 ciclos;
- Frequência de operação de DC até 20MHz;
- Ciclo de execução das instruções de 4 períodos de relógio;
- Até 13 fontes de interrupção (internas e externas);
- 8K words (14 bits) de memória FLASH (programa);
- 368 bytes de memória de dados;
- 256 bytes de memória EEPROM;
- Portas A, B e C para interfaceamento;
- 3 timers internos;
- Canais de comunicação MSSP e USART;
- 5 entradas Analógico-Digital na porta A;

5.3 DESCRIÇÃO DO HARDWARE DO SISTEMA DSSS/PIC16F876

Conforme a figura 26, um sistema mínimo para testes deve ter a alimentação de +5V/GND nos pinos 20 / 8 e 19, um cristal de 20MHz entre os pinos 9 e 10 com os capacitores de desacoplamento para terra. Para a interface do programador (vide circuito na figura 26-) devem ser reservados os pinos 28, 27 e 24 (RB7/PGD, RB6/PGC e RB3/PGM) assim como o pino 1 (!MCLR/Vpp) conectado a +5V via resistor de 10k.

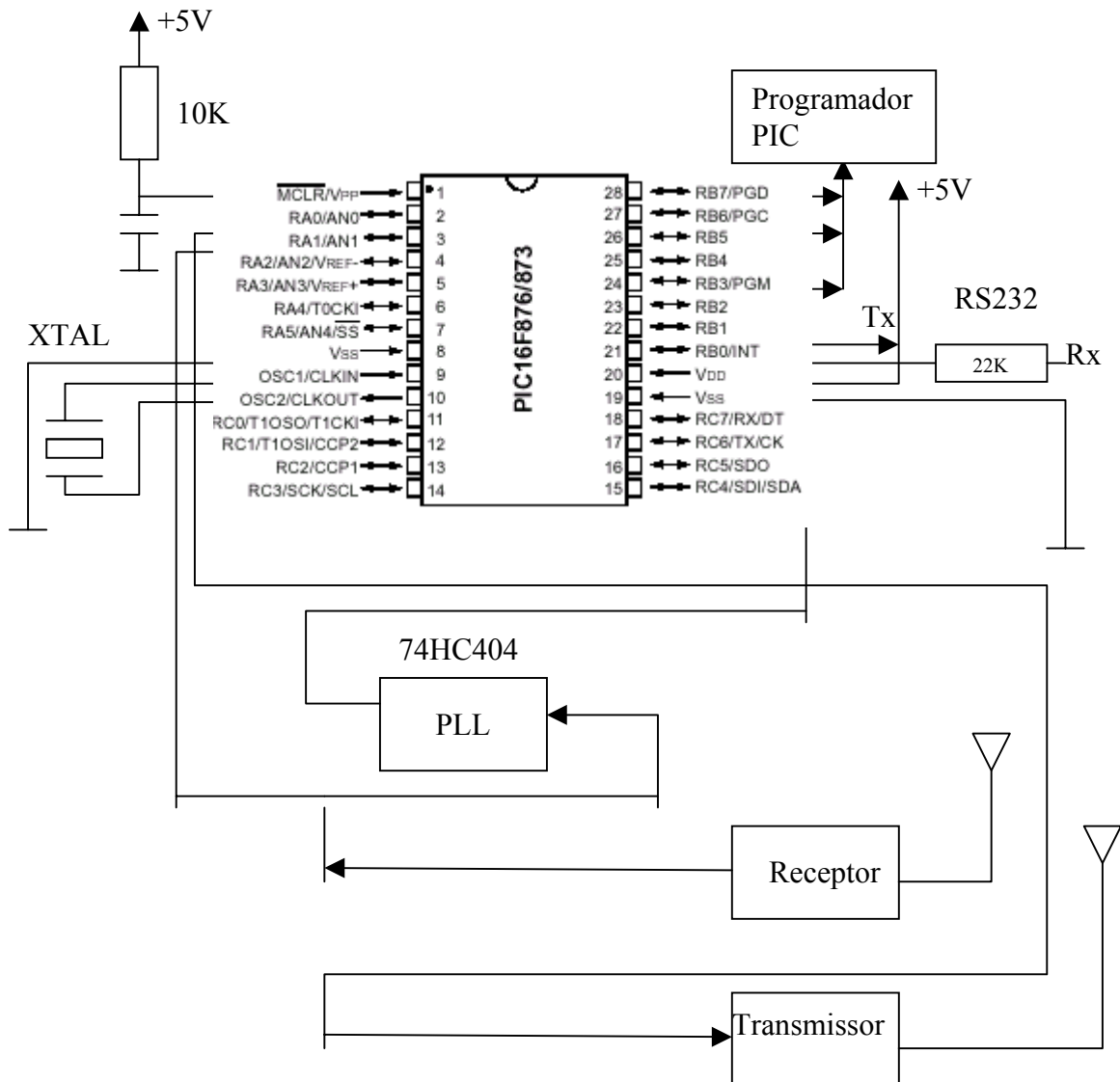


FIGURA 26: Hardware do sistema PIC16F876 / DSSS

Para a interface RS232 com o PC são usados os pinos 22 e 21 (Tx e Rx). Notar que, ao se usar cabo mais curto (menos de 1m nesse projeto) entre o PC e o sistema PIC/DSSS, pode-se dispensar os conversores de nível RS232 (+12/-12V). Para isso é colocado um resistor de 22K entre o conector DB9 da interface RS232 e o pino 21 (Rx). Como os conversores de nível realizam uma inversão (5V p/ -12v e 0 p/ +12V), na programação deve ser feita a opção de inversão dos níveis de saída e recepção do PIC (INVERT). Para o teste da interface RS232 foi usado o software Hyperterminal que acompanha o Windows (figura 27).

```

pic - HyperTerminal
Arquivo Editar Exibir Chamar Transferir Ajuda
t, rc: 192 192
v
Caracter a transmitir:

rx_buffer[ 0]: A8
rx_buffer[ 1]: 78
rx_buffer[ 2]: DB
rx_buffer[ 3]: 62
rx_buffer[ 4]: 6F
rx_buffer[ 5]: F5
rx_buffer[ 6]: BE
rx_buffer[ 7]: 56
pointer: 1
t, rc: 192 192
m
Caracter a transmitir:

rx_buffer[ 0]: 52
rx_buffer[ 1]: 0E
rx_buffer[ 2]: 49
rx_buffer[ 3]: C5
rx_buffer[ 4]: DE
rx_buffer[ 5]: EA
rx_buffer[ 6]: 7D
rx_buffer[ 7]: 2D
pointer: 0
t, rc: 192 192
k
Conectado 00:00:19 VT100 1200 8-N-1 SCROLL CAPS NUM Capturar Eco de impressão
Iniciar Microsoft Word - Implemen... pic - HyperTerminal 19:03

```

FIGURA 27: Tela do SW Hyperterminal com as respostas dos testes com o PIC16F876 / DSSS

O pino 2 (RA0) e o pino 23 (RB2) foram escolhidos para a saída TX_DSSS e relógio para testes de loopback. Os pinos 3 (RA1) e 25 (RB4) foram escolhidos para entrada dos

sinais RX_DSSS e relógio recuperado, aproveitando a capacidade de interrupção externa do 16F876 via mudança de estado nas portas RB4 a RB7.

O diagrama de blocos da figura 28 apresenta a relação entre as partes do software para fazer o espalhamento/desespalhamento da informação que chega via porta RS232.

A idéia inicial era fazer a inserção dos caracteres/bytes que chegam via RS232 em um fluxo contínuo enviado para a saída TX_DSSS, com enchimento de bytes (*stuffing*) colocando o valor 0x00h na variável Rx. Porém, durante os testes de eco, constatou-se que seria necessário desativar a interrupção interna timer0 em algumas partes do programa, interrompendo a continuidade do feixe de dados. Considerando-se também que a natureza da informação a ser transmitida (dados) é por rajadas (*bursts*), seria mais adequado o sistema funcionar também de forma não contínua. Dessa forma, toda vez que deve ser transmitido um feixe de dados é acionada a rotina de sincronização inicial da seqüência pseudoaleatória. A taxa de transferência de bits é determinada por uma interrupção interna do 16F876 gerada pelo timer0. Esse timer gera uma interrupção interna quando há o overflow de 0xffh para 0x00h no registrador de 8 bits TMR0. Também existe um bloco prescaler que faz com que o sinal de relógio de entrada (relógio do cristal dividido por 4) seja dividido por 2, 4, 8, 16, 32, 64, 128 ou 256. Portanto, a máxima taxa possível é de $20\text{MHz}/4/256/2 = 9765.6 \text{ Hz/bps}$ para interrupção interna de transmissão DSSS. Se for escolhida a taxa de 1200 bps na RS232 obtem-se um ganho de processamento de $10 \log (9765.6 / 1200) = 9.1 \text{ dB}$ próximo aos 10dB desejados.

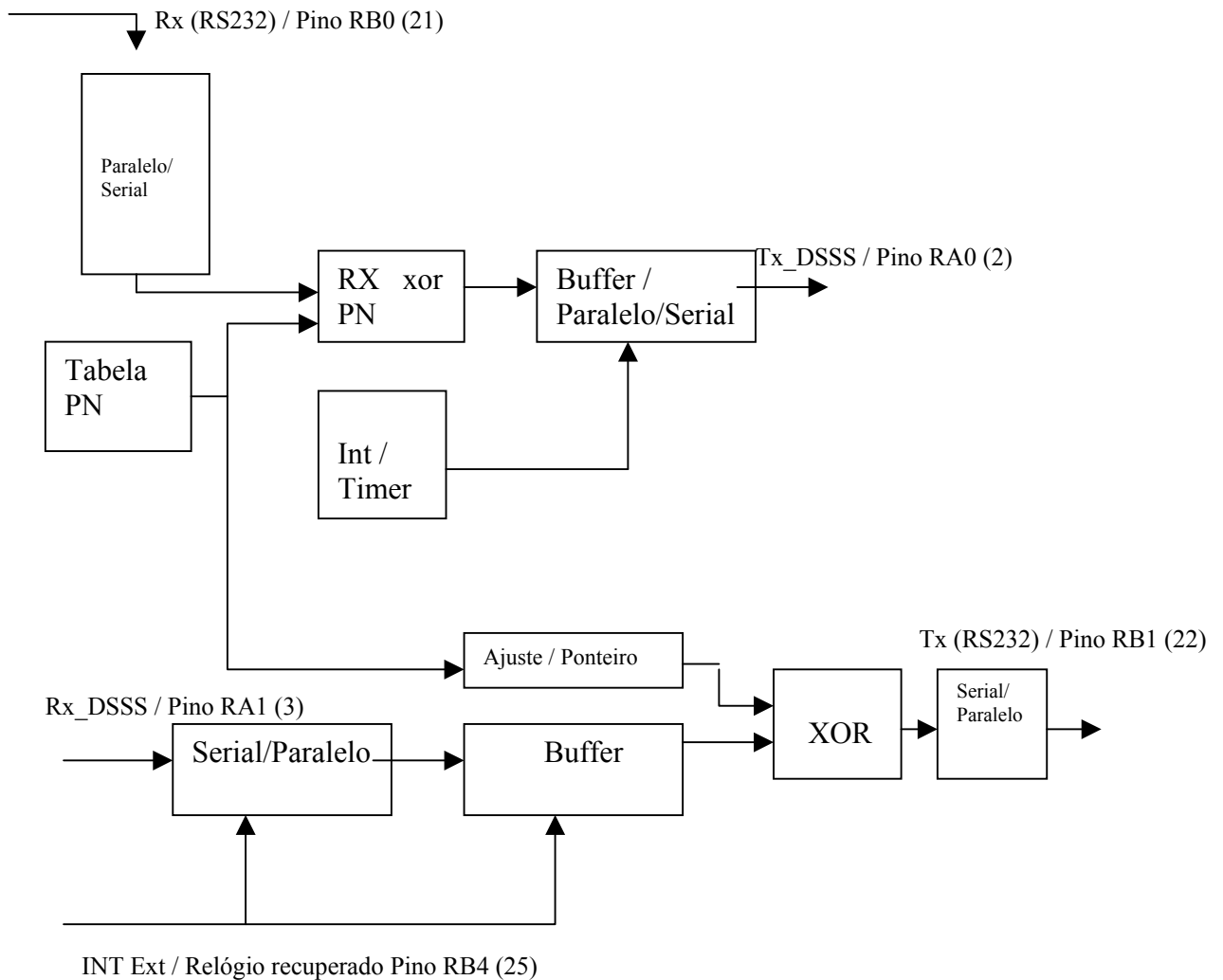


FIGURA 28: Diagrama de blocos do software PIC16F876 / DSSS

Outra discussão a ser feita é o número de instruções que podem ser executadas por ciclo de interrupção de transmissão. Levando-se em conta que devem ser reservados tempos para a interrupção de recepção e para o programa principal (que lida com a interface RS232 e com o processamento dos caracteres para/de os sinais DSSS), tem-se um tempo aproximado de 1/3 do período da interrupção ($1/9765/3 = 34.14\mu\text{seg}$). Cada instrução é executada em 4 ciclos de relógio ($4/20\text{MHz} = 0.2\ \mu\text{seg}$), portanto o número máximo de instruções fica limitado a: $34.14/0.2 = 170$. Após a compilação do programa fonte proposto deve ser feita

novamente essa avaliação para compatibilização com os outros blocos do software, de forma que não seja perdido nenhum byte. Para cada interrupção é transmitido um bit do sinal TX_DSSS.

Para a recepção pode ser usado o sinal de relógio recuperado a partir do sinal RX_DSSS como interrupção externa. Nos testes com apenas uma placa 16F876 é usado um sinal de relógio gerado pela própria rotina de tratamento da interrupção de transmissão do sinal TX_DSSS. Dessa forma é feito um loopback entre os sinais RX_DSSS e TX_DSSS com a saída de relógio ligada a entrada de interrupção externa. Também foram avaliadas duas formas para recuperação de relógio de recepção que são o uso do PLL 74HCT4046 e por software, usando a variação na entrada de interrupção como gatilho para o timer interno (timer0 com a mesma frequência de transmissão). Esse timer interno estará em fase com os dados de entrada permitindo a leitura bit a bit. A interrupção externa escolhida é a gerada por uma mudança de estado nas portas RB4 a RB7. A porta RB4 é a escolhida, já que as demais portas são usadas pelo programador PIC. Na rotina de interrupção é feita uma leitura de RB4 para efeito de filtro e clear do flag de interrupção. Como nas discussões anteriores, a rotina de interrupção deve possuir seu tempo de execução limitada para dar tempo às demais rotinas do software.

5.4 DESCRIÇÃO DO SOFTWARE DO SISTEMA DSSS/PIC16F876

O software é feito em linguagem C para o compilador CCS (Custom Computer Services) rodando sobre o ambiente de programação MPLAB for Windows/16 5.00.00 da Microchip Technology, Inc . O código gerado é transferido para o PIC 16F876 via software EBTP Internet Appliance Development Board v. 1.00 próprio para uso com a placa do programador PIC construída. A listagem do programa fonte em C pode ser vista no anexo 1.

O programa DSSS.C é dividido em três rotinas: interrupção para transmissão (RTCC_interrupt), interrupção para recepção (RB_interrupt) e principal (main). Para dar maior velocidade, a lógica XOR do sinal a transmitir com a sequência PN é feita byte a byte,

sendo armazenada em um buffer definido como variável global para que possa ser usado por todas as rotinas do programa. A idéia é fazer a decomposição do byte/caracter (da entrada Rx do PIC, vindo do PC) a transmitir de forma espalhada, bit a bit e, de acordo com o valor desse, enviar um byte da tabela de seqüências PN invertido ou não (um ou zero de acordo com a lógica XOR).

No lado da recepção o sinal recebido gera uma interrupção externa ao microcontrolador que trata de montar um byte a cada oito pulsadas na entrada correspondente (relógio recuperado a partir do sinal recebido em banda básica). Esse byte é comparado via lógica XOR com a tabela em memória com o ponteiro já ajustado para a diferença de atraso entre transmissor e receptor. Esse ajuste é feito na rotina de inicialização do programa em que o PIC transmite seqüencialmente o valor 0x00 (que pela lógica XOR como os códigos PN em tabela, transmite os próprios códigos bit a bit). Pela lógica XOR, se a maioria dos bits for igual a zero, é considerado bit 0, se a maioria é igual a 1, é considerado de valor 1 o bit correspondente do byte/caracter sendo montado, que posteriormente é enviado ao PC via Tx. O processo de montagem da seqüência de bytes/caracteres embutidos nas seqüências DSSS recebidas deve ser realizado por rotinas (protocolos) em camada de nível mais alto.

A tabela de seqüências PN é feita por uma seqüência de 8 bytes (64 estados possíveis para cada chip) obtida por um programa simulador de uma cadeia de registradores de deslocamento de 6 estágios com realimentações a partir das saídas 1 e 6, portanto $N=6$, $2^6 = 64$ (a partir de [103] da tabela 2 para o grau 6).

Outra observação é quanto ao interfaceamento com a porta serial do micro, pois no diagrama de blocos do software é apresentada a rotina que faz a conversão paralela/serial no transmissor e serial/paralela no receptor. Na verdade é realizada novamente a conversão serial do caracter enviado/recebido, mas no software em linguagem C, esses sinais são tratados byte a byte pelas funções `printf()` e `getch()` que, para o compilador, significam a escrita e leitura da porta serial do PIC.

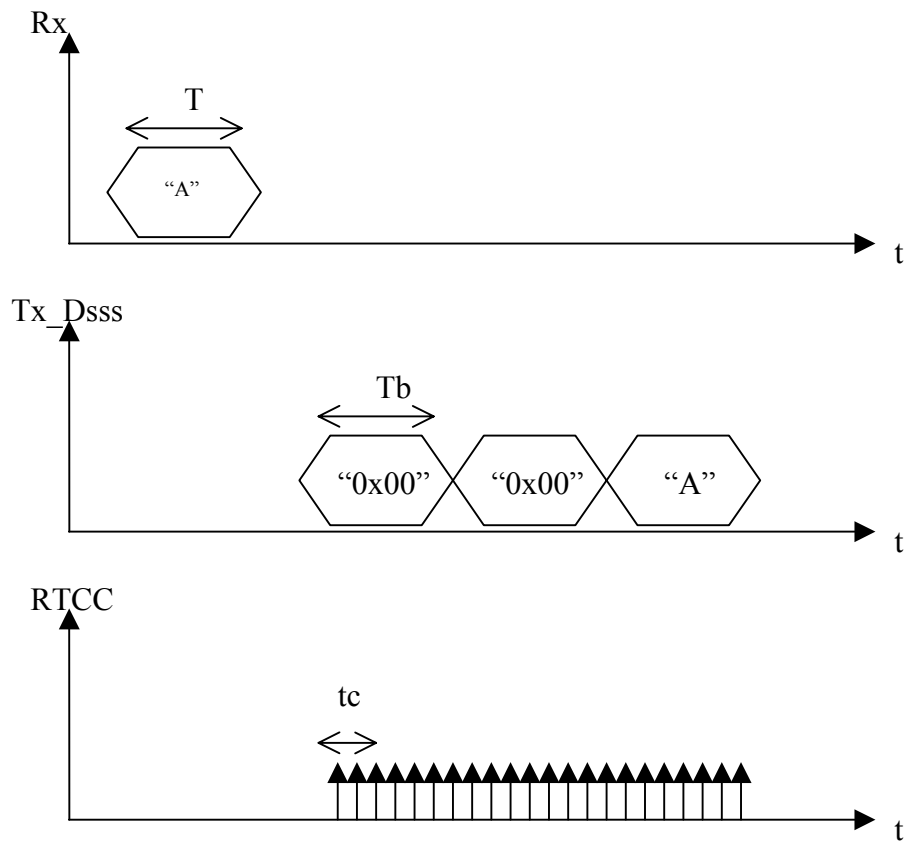


FIGURA 29: Diagrama de temporização para transmissão de um caractere

Na figura 29 podem ser vistas as formas de onda para transmissão de um caractere. Rx corresponde a entrada RS232 a partir da interface com o microcomputador e T é o tempo necessário para recepção de 8 bits do caractere (no exemplo, “A”) mais um bit de partida (start bit) e um de parada (stop bit) totalizando 10 bits. Para a velocidade de 1200bps, $T = 10 \cdot 1/1200 = 8.33\text{ms}$. Tx_Dsss é a saída com o espectro espalhado por seqüência direta em que cada ciclo de transmissão de caractere ou caractere de inicialização (0x00) corresponde a um tempo $T_b = 64 \cdot t_c$, onde t_c , período de chip é dada pelo temporizador interno ao PIC timer0 (RTCC) igual a $1/9765 = 12\mu\text{s}$ e $T_b = 6.55\text{ms}$. Como o sistema funciona por bursts (rajadas de dados) deve haver transmissão de caracteres de inicialização por tempo suficiente para o PLL do receptor fazer a captura e estabilização do relógio de interrupção de entrada, sendo necessários pelo menos 2 caracteres de inicialização. Para a transmissão dos dois

caracteres de inicialização e do caracter “A” são necessários 192 períodos do timer interno RTCC.

Segue abaixo a descrição das principais rotinas do programa dsss.c compilado para o sistema PIC16F876:

- 1) **Rotina `rtcc_interrupt()`**: esta é a rotina executada toda vez que é ativada a interrupção interna gerada pelo timer0. Essa rotina tem como principal função analisar bit a bit os bytes contidos no buffer de transmissão `tx_buffer[]` e colocá-los na porta RA0 (TX-DSSS). Após 64 interrupções (tabela `tx_buffer[]` com 8 bytes) é ativada a variável **`fim_tx`** que indicará à rotina principal que naquele instante houve a transmissão de todo o buffer de transmissão (que corresponde a um caracter/byte espalhado pela seqüência PN).
- 2) **Rotina `rb_interrupt()`**: ativada por mudanças de estado na entrada RB4 (relógio recuperado), quando então é feita a leitura da entrada RA1 (`rx_dsss`). A partir dos bits recebidos, essa rotina monta os bytes que entrarão na tabela `rx_buffer[]`. Após 64 interrupções é ativada a variável **`novo`** que indicará à rotina principal que acaba de ser completada a tabela `rx_buffer[]` e já pode ser feita a leitura e desespalhamento da informação recebida (no caso um byte/caracter). O descrito anteriormente é realizado caso se use PLL para recuperação do relógio; se for implementada a geração de relógio internamente ao PIC via timer0 ativado pela interrupção em RB4, é habilitada novamente a interrupção do timer interno. As funções de leitura serão feitas na rotina de atendimento **`rtcc_interrupt`** com filtragem pela identificação do estado atual da rotina principal.
- 3) **Rotina `main()`**: a rotina principal é dividida em cinco estados:

Estado 0: nesse estado inicial é feita a requisição do caracter a ser transmitido para ser salvo na variável **`s`** (após a rotina de inicialização o conteúdo dessa variável será passada para a variável **`Rx`**). O buffer de transmissão `tx_buffer[]` é preenchido com a lógica XOR entre o conteúdo de cada bit da variável **`Rx`** (valor 0x00 para inicialização) e cada um dos 8 bytes da seqüência PN tabelada.

Estado 1: é habilitada a interrupção `int_rtcc`. O buffer de transmissão é enviado duas vezes para inicialização e sincronização do receptor.

Estado 2: nesse estado é montado o buffer de transmissão com a lógica XOR entre a variável `s` (caracter a transmitir) e a seqüência PN tabelada bit a bit. É feita a transmissão do buffer de transmissão até ser sinalizada a variável `fim_tx` pela rotina de interrupção de transmissão.

Estado 3: nesse estado a rotina entra em um loop aguardando a confirmação de nova tabela `rx_buffer[]` recebida pela variável `novo`. Se não está ainda em processo de sincronização inicial, é carregada essa tabela em `buf_mem[]` para posterior análise e liberar a tabela `rx_buffer[]` para a rotina `rb_interrupt()` ainda habilitada. É feita a lógica XOR entre o conteúdo de `buf_mem[]` e `pn[]` byte a byte. Se a maioria dos bits estiver correlacionada existe o alinhamento entre o sinal recebido e a seqüência PN tabelada e é feita a saída desse loop. Se não houver alinhamento, é feito o deslocamento bit a bit do conteúdo de `buf_mem[]` até se obter esse alinhamento. O número de vezes necessário para deslocamento bit a bit é armazenado na variável `pointer` que indicará o início da seqüência PN embutida nas tabelas `buf_mem[]` seguintes. Esse processo é realizado novamente para a seqüência seguinte, pois são enviados duas vezes os caracteres de sincronização. Notar que essa rotina implementa a técnica de busca serial de sincronização inicial do código de espalhamento.

Estado 4: esse estado aguarda a confirmação de novo caracter recebido para proceder o desespalhamento da informação contida em `buf_mem`. O ponto exato para início da análise é indicado pela variável `pointer` obtida no estado de sincronização inicial e é feita a comparação bit a bit com a seqüência PN para decisão de acordo com a soma dos 8 bits. Se for superior a 4, é decidido bit 1, caso contrário, bit 0. O caracter é enviado via saída Tx do PIC para impressão no Hyperterminal (comunicação serial RS232). A rotina volta para o estado 0 para aguardar novo caracter a transmitir.

Segue na figura 30 o diagrama da máquina de estados descrita acima. As principais variáveis do programa dsss.c envolvidas no processo são **rx**, **tx**, **nov**, **fim_tx**, **corr**, **tx_buffer[8]** e **rx_buffer[8]**. Nas figuras 31 e 32 serão mostradas as relações entre os conteúdos dessas variáveis em cada etapa do programa.

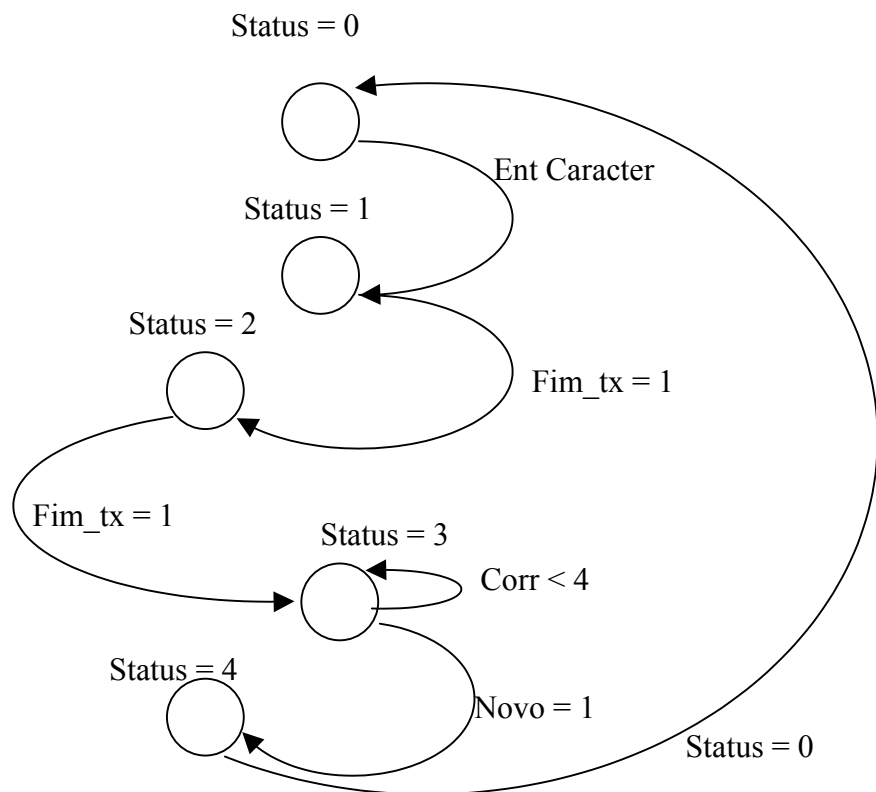


FIGURA 30: Diagrama de estados para o software Dsss.c

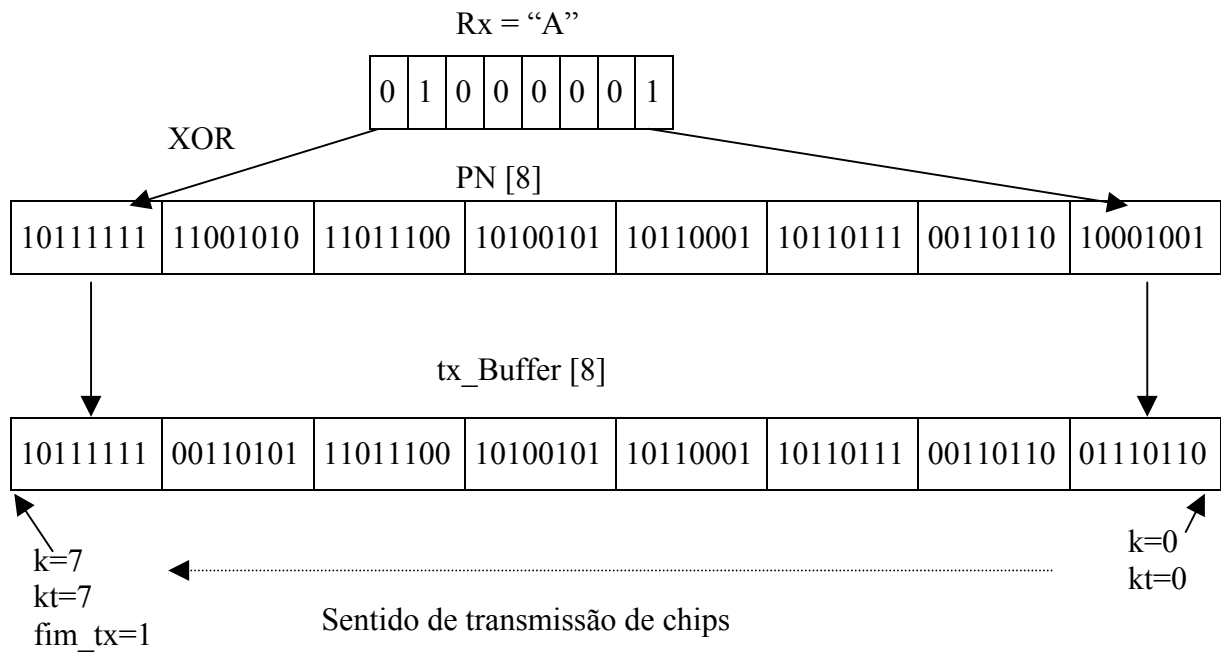


FIGURA 31: Estrutura de montagem e transmissão de chips para um caracter ("A" no exemplo)

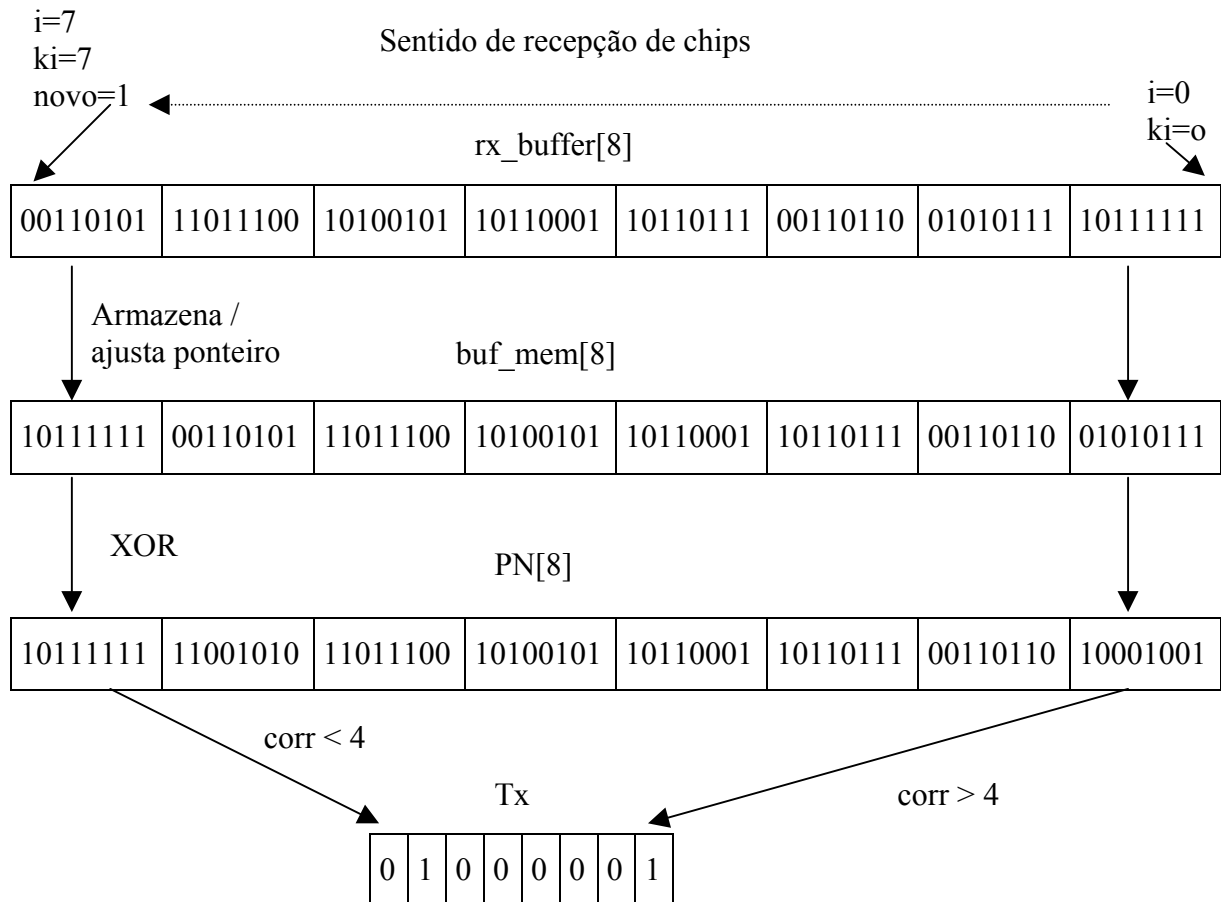


FIGURA 32: Estrutura de recepção e correlação com a seqüência PN para obter o caracter (“A” no exemplo)

No exemplo da figura 31, para cada bit do caracter “A” codificado em ASCII é feita a operação lógica XOR com o byte da tabela PN[8] correspondente. Dessa forma, quando o bit é zero, resulta na armazenagem em tx_buffer do próprio byte PN, se for um, é colocado na tabela um byte cujos bits são o inverso. A análise do caracter em Rx começa no bit menos significativo assim como a tabela PN[8] é lida a partir do primeiro byte. A transmissão do conteúdo de toda a tabela tx_buf[8] é feita do chip menos significativo para o mais. Notar que o espalhamento de rx é feito nos conteúdos das variáveis e não de forma online, sendo que a transmissão dos chips será dada pelo timer0 que irá gerar as interrupções de transmissão.

Quando se trata dos bytes de `tx_buf[8]` (após a operação XOR de espalhamento), cada unidade de informação é tratada por chip, de acordo com a terminologia de Spread spectrum. A variável `fim_tx` é levada a um quando as variáveis `k` e `kt` atingem o valor 7 dentro da rotina de interrupção de transmissão.

Na figura 32 é mostrada a estrutura de recepção e correlação com a seqüência PN. Notar que a recepção pode apresentar um atraso em relação à ordem dos bytes armazenados na tabela `PN[8]` assim como erros de chip recebidos. Na rotina de inicialização é calculado o valor de ponteiro para que seja feito o ajuste da tabela `rx_buf[8]` antes de copiar para a tabela `buf_mem[8]` de trabalho. Dessa forma, enquanto o PIC está recebendo uma nova tabela `rx_buf[8]` via interrupção de recepção, pode fazer a correlação da tabela anterior para obter o caracter recebido nos intervalos entre solicitações de interrupção. A lógica XOR entre o conteúdo de `buf_mem[8]` e `PN[8]` é feita byte a byte a partir do menos significativo, porém na rotina de correlação é feita a contagem dos bits 1 do resultado da XOR na variável `corr`. Se maior que 4, é decidido bit 1 para o caracter Tx, se menos que 4, bit 0. Isso ocorre para ganhar imunidade a chips errados na recepção, pois haverá apenas todos os bits 1 ou 0 no resultado da XOR se todos os bits conferirem com a ordem no byte correspondente da tabela `PN[8]`. No exemplo, o primeiro byte de `buf_mem[8]` contém 2 bits com erros e o resultado da XOR é o seguinte:

$$0101\ 0111\ \text{XOR}\ 1000\ 1001 = 1101\ 1110$$

Como o resultado apresenta número de bits 1 maior que 4, o bit correspondente é o “1” no caracter Tx que será enviado pela porta serial RS232. Essa rotina faz com que o sistema possa operar em meios com maior presença de ruído e interferência e manter a mesma taxa de erros na porta RS232. Após a montagem do caracter Tx é feita a transmissão pela interface RS232.

Na rotina de inicialização é feito um processo semelhante ao apresentado acima, porém o conteúdo de `buf_mem[8]` é deslocado até que o conteúdo de `tx` apresente o maior número possível de bits zero, lembrando que o transmissor envia primeiro o caracter 0x00 para que a seqüência transmitida seja exatamente a seqüência PN. O número de deslocamentos até atingir correlação é armazenado na variável pointer que será usada adiante para ajuste da seqüência recebida em `rx_buf[8]`. Assim que o receptor detecta a correlação

(corr>4) , a próxima seqüência é recebida e armazenada em buf_mem[8] para correlação. Essa rotina implementa por software a técnica de correlação com filtro casado com a seqüência PN.

Pode-se agora fazer uma descrição do processo de comunicação entre dois sistemas A e B para a transmissão de um caracter. O sistema usa o método half-duplex, ou seja, quando uma ponta está transmitindo, não recebe e vice versa. Inicialmente o sistema A requer o caracter a ser transmitido ao usuário (estado 0). Enquanto isso o sistema B também está no estado 0, aguardando entrada de caracter. Assim que é dada a entrada do caracter no sistema A, este passa para o estado 1 no qual é liberada a interrupção de Tx, fazendo com que sejam transmitidas as seqüências PN (Rx=0x00) para inicialização da ponta B. Porém, como B está no estado 0, não há sincronização por parte deste. Para tanto deve ser dada a entrada de um caracter em B (pressionando uma tecla no Hyperterminal) fazendo com que passe para os estados 0, 1, 2 e 3. Como A já está transmitindo sua seqüência PN, o sistema B entra na rotina de inicialização obtendo o sincronismo. Então B passa ao estado 4 assim como A passa para o estado 3. O sistema B acusa novo caracter recebido e imprime na tela do Hyperterminal o caracter enviado por A. B volta ao estado 0 e A fica no estado 3, aguardando caracter de B.

5.5 TESTES REALIZADOS

Para o teste do programa foram construídas duas placas PIC, com o software descrito anteriormente carregado, e foram conectadas por fios entre os pinos Tx_DSSS e RX_DSSS. Após a compilação para o código dsss.hex pelo software MPLAB, esse código é enviado para cada uma das placas PIC via programador através do software EDTP.

O software Hyperterminal é chamado para comunicação serial com a primeira placa PIC I. Após ser ativado pelo botão **run** do EDTP, a placa PIC I fica aguardando a entrada do caracter a transmitir. Com a entrada do caracter, o PIC faz a transmissão e a placa PIC II (em estado de recepção) com loopback na interface RS232 e posterior retransmissão de volta para a placa I. A informação é decodificada na placa I e é feita a impressão para o Hyperterminal

do conteúdo do buffer de recepção e do carácter recebido. Se não houver erros, o carácter recebido deve ser igual ao teclado anteriormente. Nesse teste foram teclados todos os caracteres possíveis sem erro na resposta. A montagem para esse teste pode ser vista na figura 33.

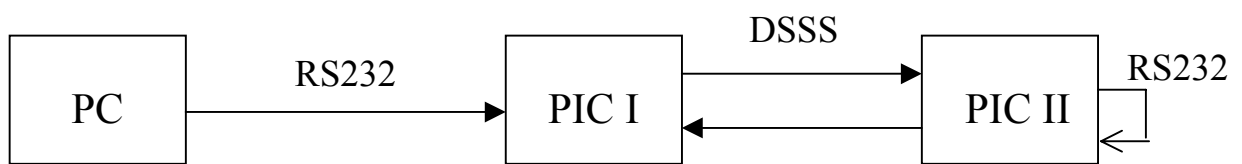


FIGURA 33: Montagem com a interligação de duas placas PIC16F876

Nas figuras a seguir estão as medidas feitas no osciloscópio para os sinais RX, TX_DSSS e RX_DSSS na placa PIC I. No padrão RS232 um byte é transmitido de forma serial utilizando normalmente um bit de partida, seguido dos oito bits de dados e mais um bit de parada. Um bit 1 é representado eletricamente como uma tensão entre -5 e -15 volts e um bit 0 é representado por uma tensão entre $+5$ e $+15$ volts. No estado de espera, a linha TX é mantida sempre em 1, sendo o bit de partida um bit 0 e o de parada um bit 1. Na figura 34 está o sinal elétrico RX, recebido pelo PIC da porta serial do PC, medido no pino RB0, para o carácter “A”, o qual corresponde ao código ASCII decimal 65 ou binário 01000001. O comprimento total é de ± 8 ms confirmando o valor calculado anteriormente de $T=8.33$ ms.

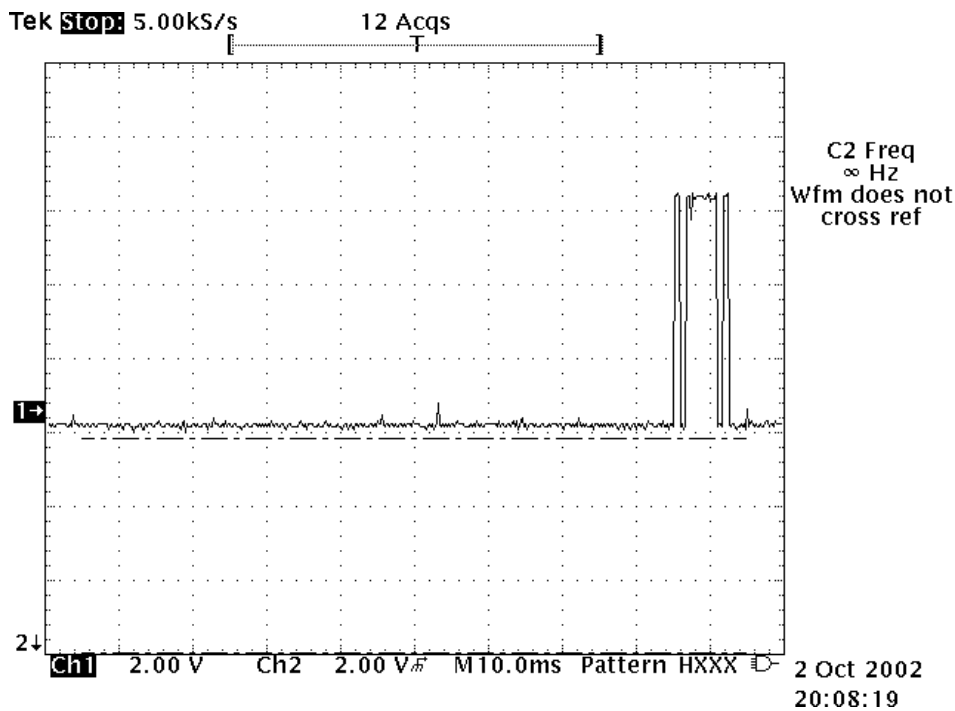


FIGURA 34: Rx medido para o caracter “A”

Na figura 35 está a medida do sinal TX_DSSS correspondente à transmissão do caracter recebido em RX. O comprimento é de +/- 19.5 ms de acordo com o tempo já calculado de transmissão de cada caracter espalhado ($T_b = 6.55\text{ms}$), já que existe a transmissão de dois caracteres 0x00 mais o caracter de fato, ou seja $3 \times T_b = 19.65\text{ms}$. A mesma seqüência é usada para o espalhamento de cada caracter. Pode-se notar que os dois primeiros caracteres (largura de 64 chips) apresentam os primeiros chips transmitidos em 11111101 sendo que no último esses chips são invertidos para 00000010 devido ao primeiro bit a ser espalhado do caracter “A” ser 1 (lógica XOR com 1 resulta na inversão dos bits). Lendo os chips da direita para a esquerda tem-se BFh que é o primeiro byte armazenado na tabela pn[8] no programa dsss.c.

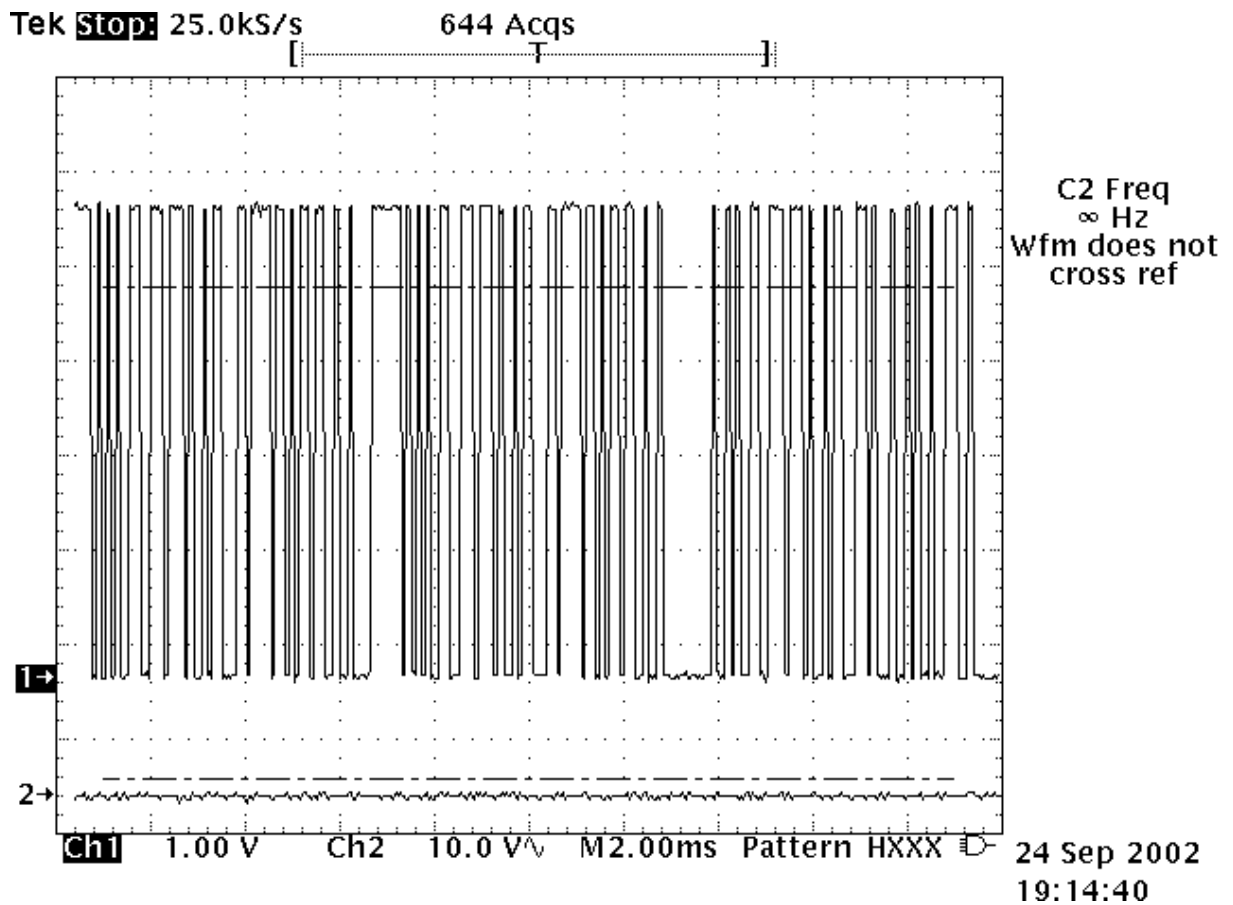


FIGURA 35: TX_DSSS medido para o caracter “A”

Para a medida da figura 36 foi forçada a transmissão de forma contínua do caracter “A” nas formas **sem espalhamento** através do pino RB2 (sem os bits de início e parada e sinal elétrico invertido com relação ao padrão RS232) e **espalhada** para comparação nos canais 1 e 2 do osciloscópio. Na figura 36 pode-se notar que existem transições de bit mesmo no longo período de 5 bits em zero do sinal sem espalhamento. Isso é uma vantagem para manutenção do sincronismo de relógio recuperado no receptor.

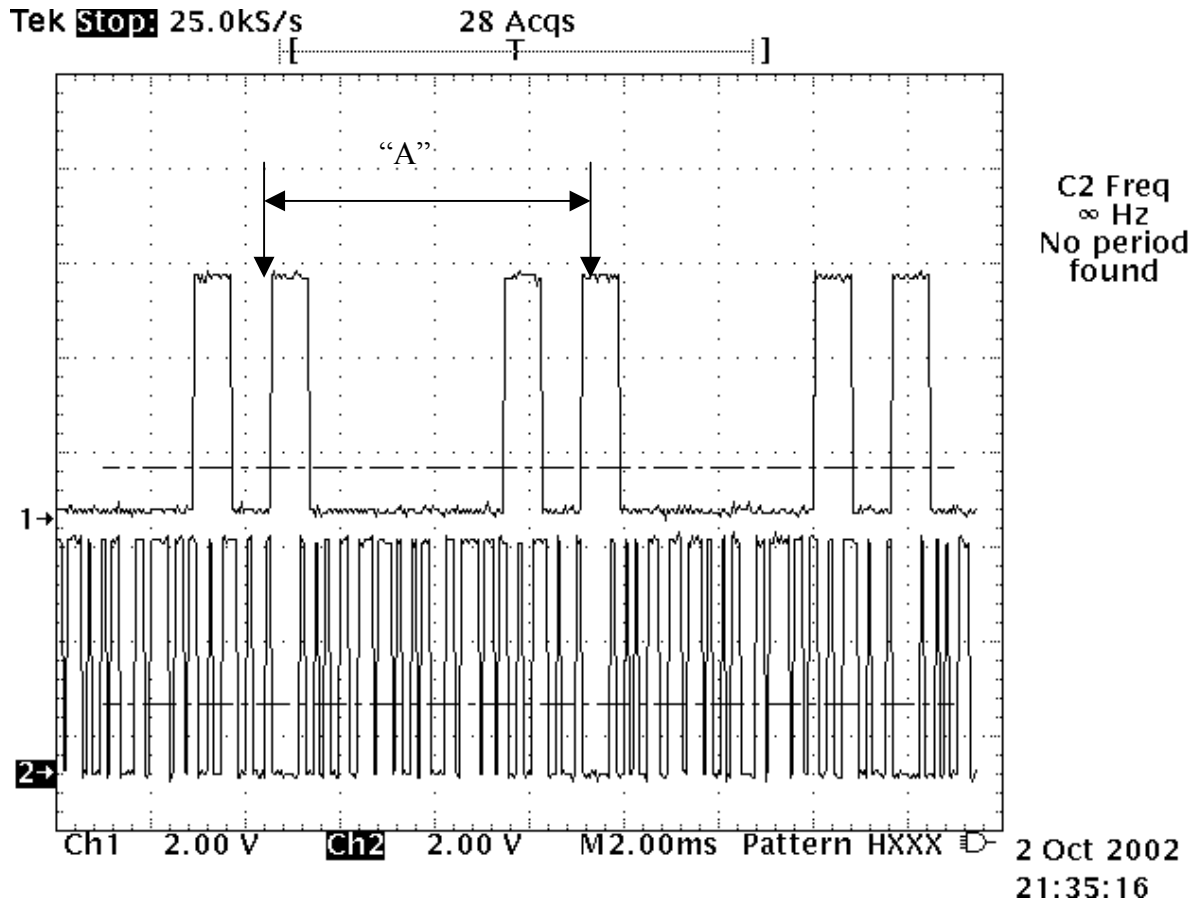
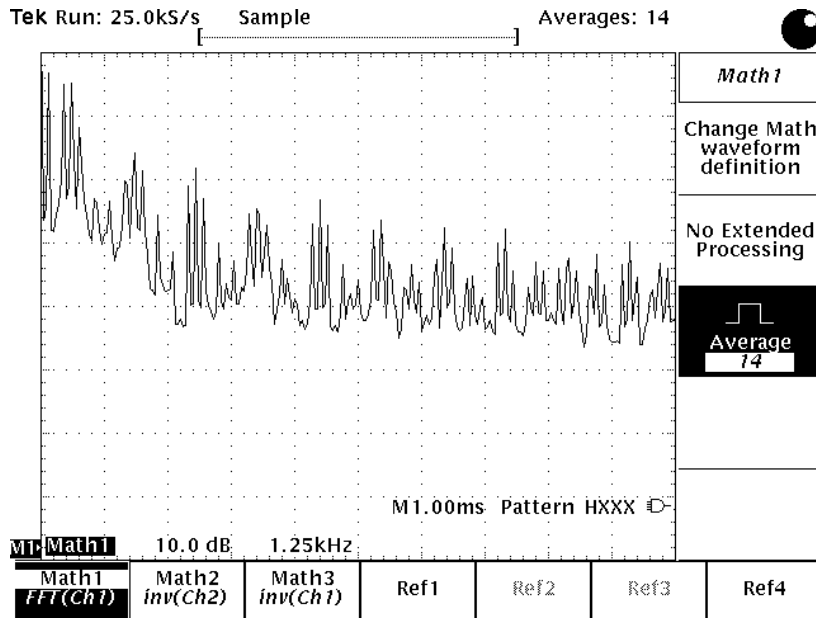
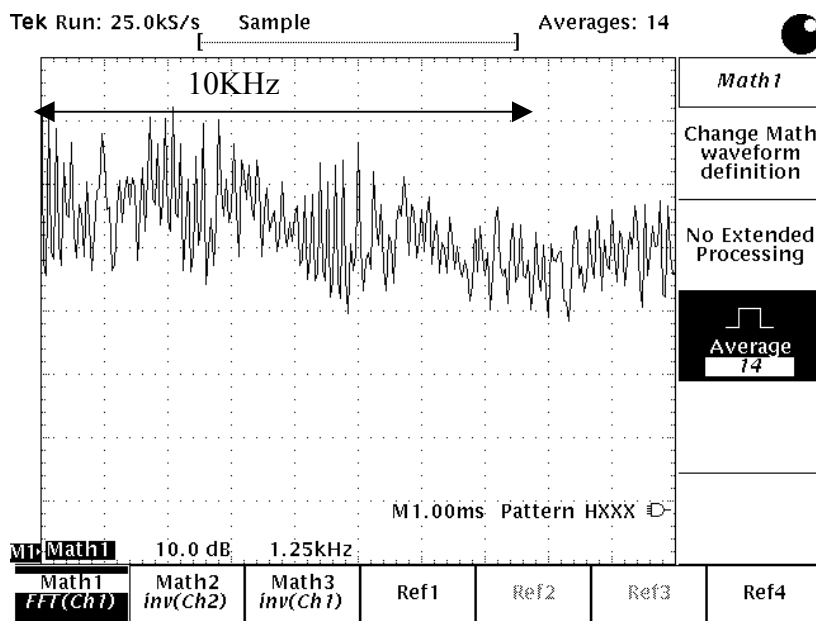


FIGURA 36: TX_DSSS e sinal em RB2 medidos para a transmissão contínua do caracter “A”

Nas figuras 37a e 37b, tem-se as medidas do espectro dos dois sinais analisados anteriormente. Nelas pode-se notar que no espectro do sinal sem espalhamento, há uma concentração de energia no primeiro 1KHz (lembrando que a taxa de transmissão é de 1/8 da velocidade de 9765 chips/s, 1220 bits/s). Na outra figura existe o espalhamento do espectro, com a energia se concentrando de forma mais uniforme ao longo dos primeiros 10KHz e caindo no final.



a)



b)

FIGURA 37: Espectro de TX e TX_DSSS medidos para a transmissão contínua do caracter "A"

Na figura 38 pode-se notar a temporização dos sinais Rx e TX_DSSS, confirmando os tempos T_b e T calculados anteriormente.

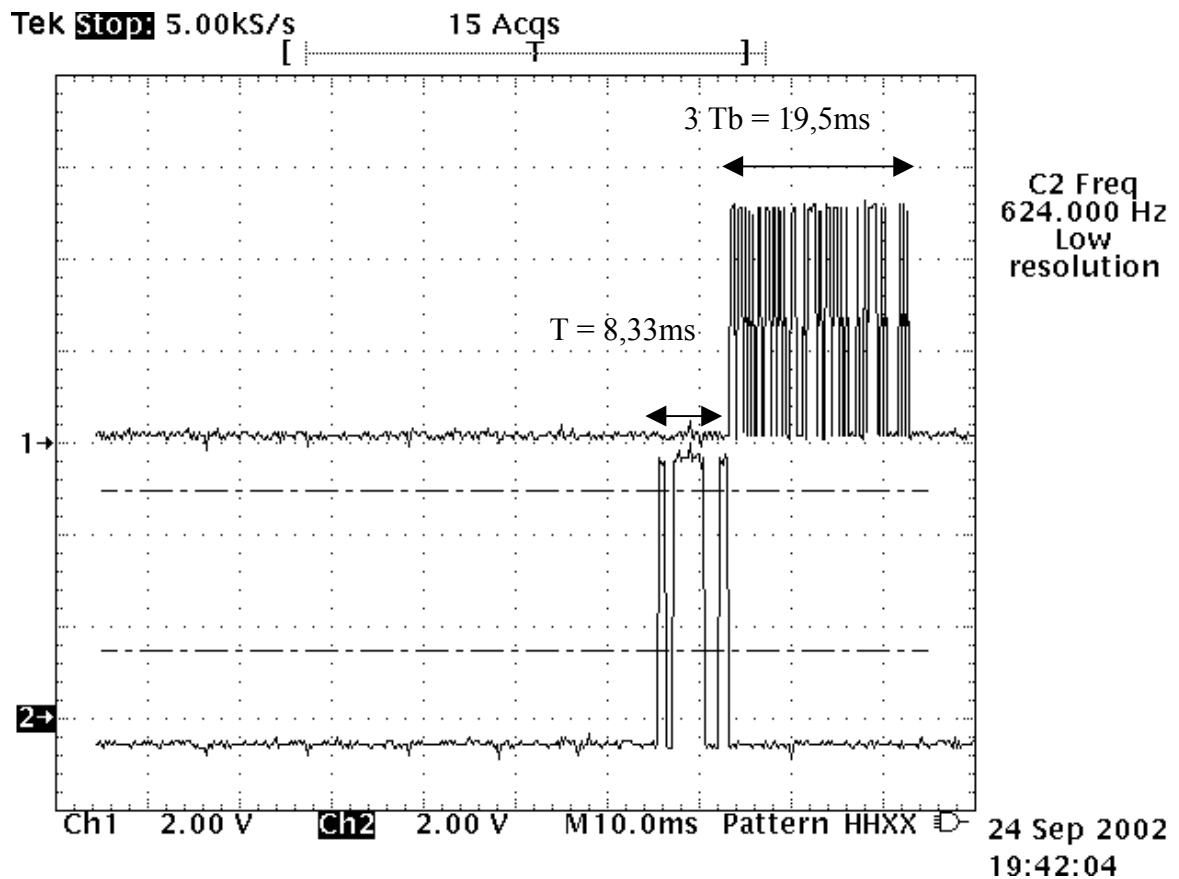


FIGURA 38: TX_DSSS e RX medidos após a entrada do caracter “A” via emulador de terminal

Na figura 39 tem-se os sinais TX_DSSS e RX_DSSS medidos nos canais 1 e 2, mostrando uma diferença de +/- 100ms entre o término da transmissão e o início da recepção. Isso ocorre devido à inserção de um delay de 100ms na rotina do programa em PIC II para separação dos trens de dados de recepção e transmissão na visualização da tela do osciloscópio.

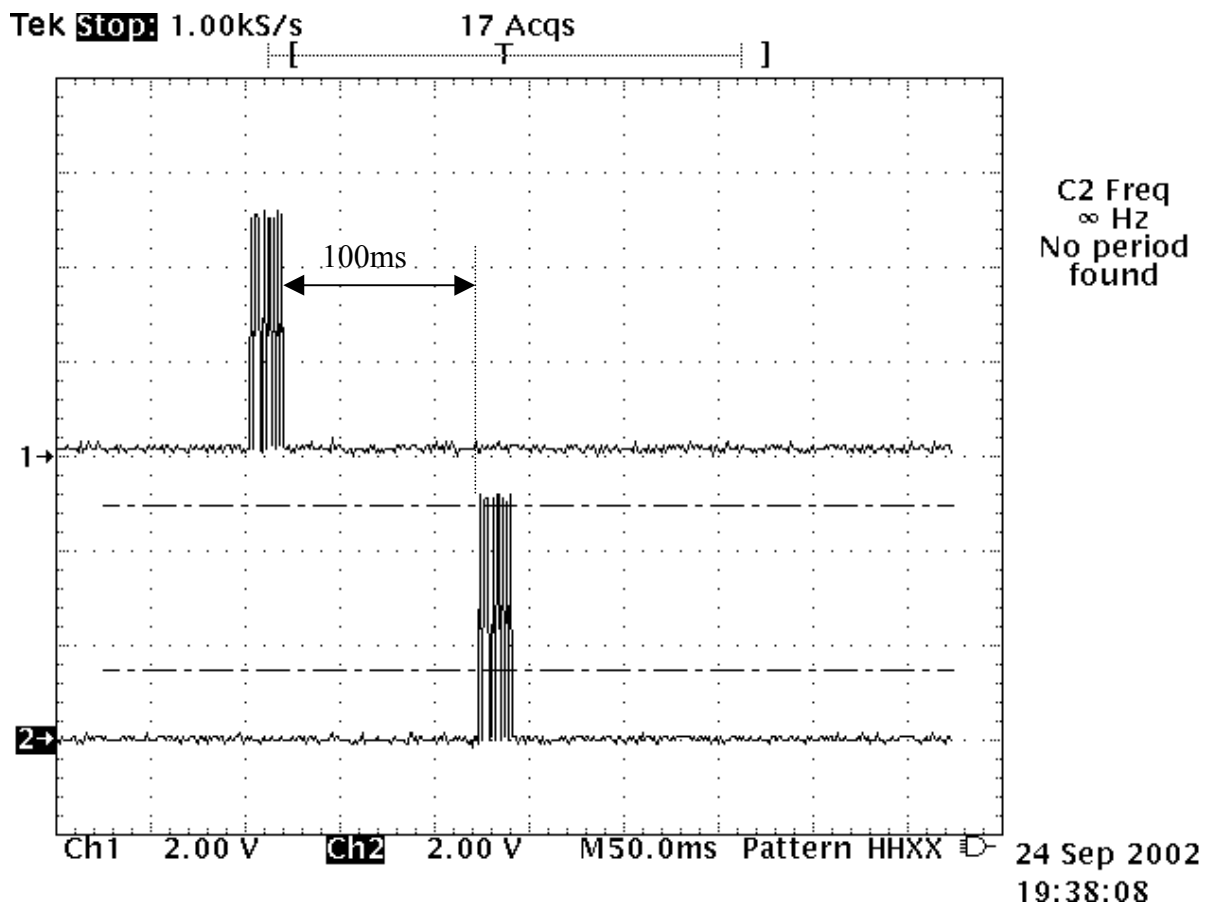


FIGURA 39: TX_DSSS e RX_DSSSS medidos após a entrada do caracter “A” via emulador de terminal

Também foi feita a montagem da figura 40, em que um módulo transmissor de FM foi conectado a uma placa PIC e um módulo receptor na mesma faixa na outra placa PIC. Uma placa foi programada para transmitir continuamente o caracter “A” enquanto que a outra foi conectada a um computador PC via interface RS232. As medidas foram feitas no analisador de espectro portátil Anritsu MS2711 (resolução espectral mínima de 10 KHz) e a frequência central da portadora é de 417,94 MHz.

Deve ser notado que, devido à precisão do instrumento, existe um alargamento adicional verificado durante a medida da portadora sem modulação. Esses módulos denominados TXM-4xx-F para transmissão e RXM-4xx-A para recepção, fabricados pela Abacom Technologies, aceitam tanto entrada analógica quanto digital com taxa de dados até

20kbps. A frequência é modulada entre 418 MHz e 433,92MHz. A alimentação típica é de +5V.

Para a realização das medidas foi usado um potenciômetro entre a saída Tx_Dsss (ou Tx para medidas sem espalhamento) do PIC e a entrada do módulo transmissor para colocar a modulação FM em faixa estreita.

O alargamento total observado, incluindo a contribuição do analisador é dado por:

$$(\text{largura total})^2 = (\text{largura do sinal})^2 + (\text{resolução analisador})^2 \quad (5.1)$$

Portanto, conhecendo a resolução do analisador de espectro, do valor medido pode ser avaliada a largura real do sinal. Da figura 42 (espectro do sinal sem espalhamento), tem-se em torno de 15KHz de largura medida. De 4.5.1 calcula-se largura do sinal igual a 11,2KHz. Para a figura 43 (espectro do sinal com espalhamento) obtem-se 60KHz de largura medida e 59,2KHz de largura do sinal. A relação entre as medidas de espectro espalhado e sem espalhamento é 5,28 e o ganho de processamento medido é de 7,23dB, próximo ao calculado de 9,1dB.

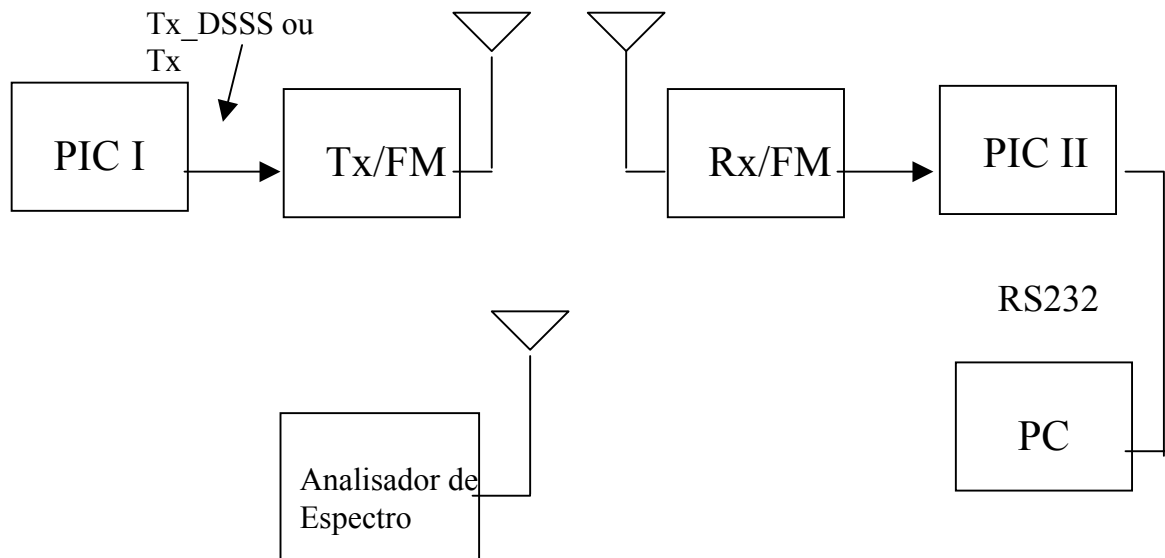


FIGURA 40: Montagem com os módulos transmissor e receptor de RF/FM

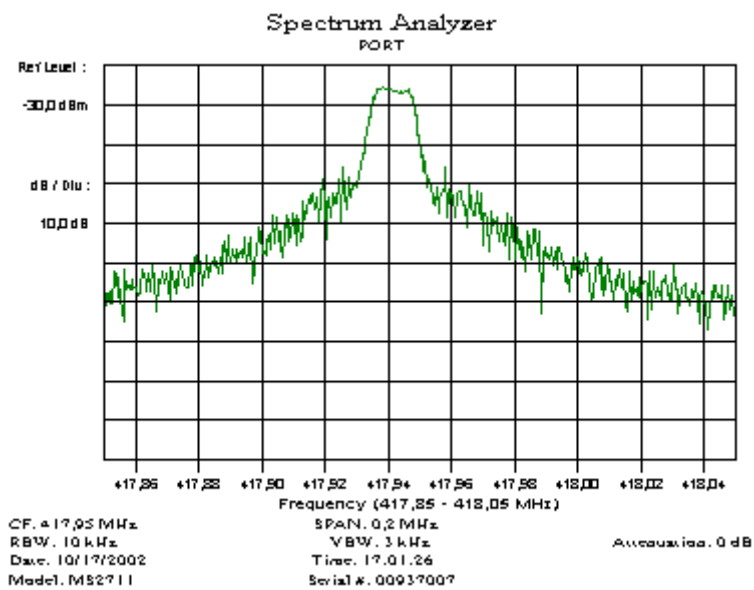


FIGURA 41: Medida do espectro da portadora do transmissor FM (417,94MHz)

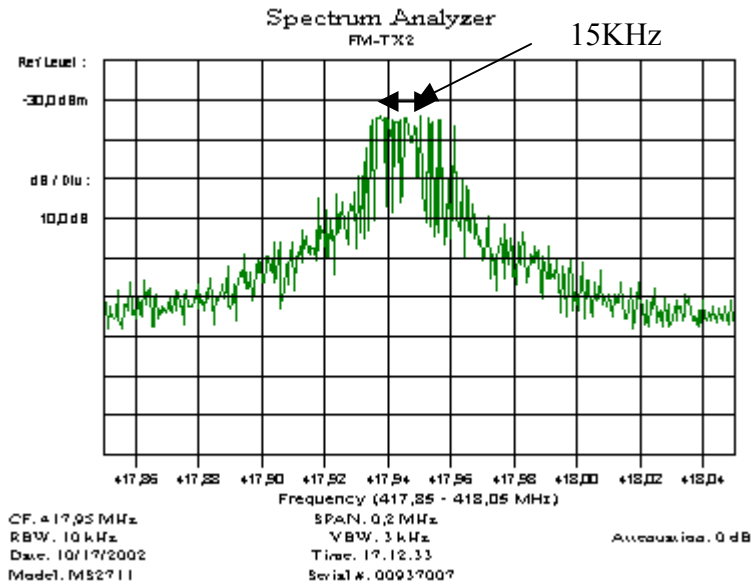


FIGURA 42: Medida do espectro do sinal modulado e sem espalhamento

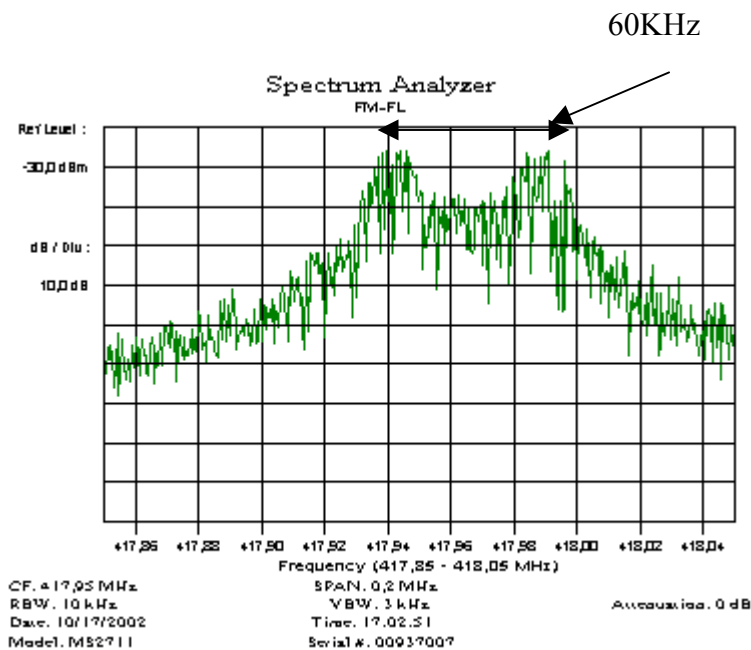


FIGURA 43: Medida do espectro do sinal modulado e com espalhamento

5.6 CIRCUITO RECUPERADOR DE RELÓGIO

Em uma das placas foi montado o circuito recuperador de relógio com o CI 74HC4046 (PLL) da figura 44. Notar que da forma proposta não é implementado o circuito de Code Tracking Loop para manter o sincronismo com a seqüência PN após a inicialização. Os valores calculados de acordo com as curvas do manual do fabricante do CI [11] são os seguintes, considerando a frequência central f_0 (relógio de recepção igual à frequência de interrupção timer0(), 9,765 KHz), frequência natural ω_n igual a 5 vezes $1/t_{set}$ (t_{set} , tempo de estabilização, deve ser menor que $T_b=6.55ms$) e fator de amortecimento ξ igual a 0,707:

$$R1 = 180K$$

$$R2 = 18K$$

$$C1 = 22nF$$

$$R3 = 24K$$

$$R4 = 12K$$

$$C2 = 100nF$$

O comparador de fase escolhido é o Phase Comparator 1. Conforme já comentado, também foram feitos testes sem o uso de PLL, utilizando o timer interno como gerador de interrupção de leitura. As formas de onda para essa configuração estão na figura 45.

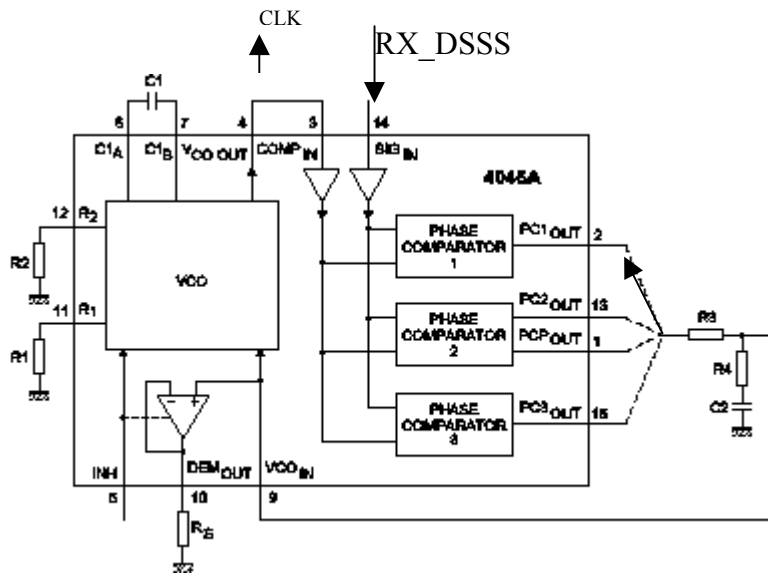


FIGURA 44: Recuperador de Relógio com o CI 74HC4046

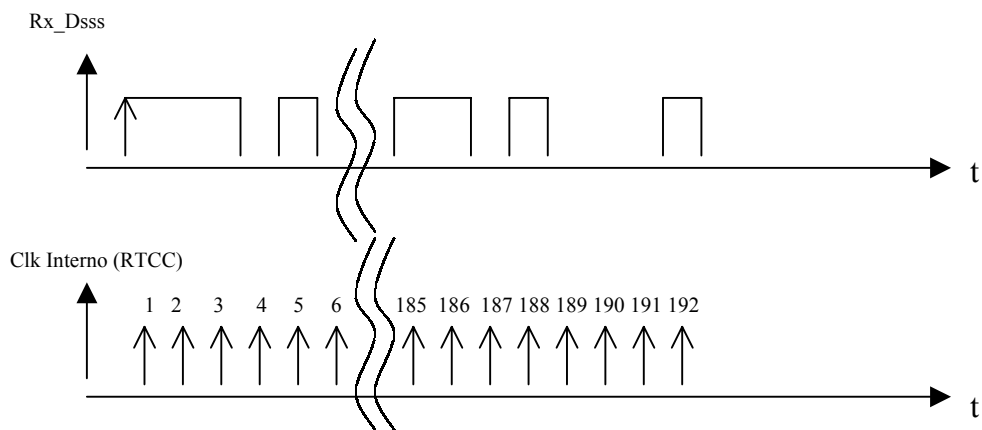


FIGURA 45: Geração interna de relógio pelo timer0 (RTCC) a partir da primeira transição de Rx_dsss

6 CONCLUSÃO

Neste trabalho foram feitas as análises matemáticas dos sinais envolvidos no processo de espalhamento espectral em seqüência direta assim como a descrição dos blocos necessários para implementação do transmissor e receptor DSSS.

Todo o processo de geração de banda espalhada por sequenciamento direto, bem como de recuperação do sinal original foi simulado no MatLab. Testes de introdução de ruído branco gaussiano e de interferência na mesma freqüência também foram simulados e os resultados obtidos, conforme visto no ítem 4.2, estão resumidos abaixo:

- (a) O uso da técnica RASE de aquisição de sincronismo de código apresentou realmente uma grande rapidez, atingindo a correlação em cerca de 80 chips (menos que os 127 chips de período da seqüência PN). Para o sistema proposto esse tempo seria de 69,4 useg ($80 \times (1/1.115.200)$ chips/seg), mesmo com ruído cerca de 10 dB acima do sinal recebido.
- (b) Confirmado o ganho de processamento de 10dB com a simulação de ruído no limite máximo.
- (c) Comprovação do problema near-far para receptores de múltiplo acesso. Os níveis dos sinais principal e interferente devem ser da mesma ordem de grandeza. O código usado nos dois sinais devem possuir a mais baixa correlação possível.
- (d) O sistema apresentou imunidade à interferência de sinais refletidos com atraso de propagação compatível com as dimensões de ambientes esperados para a aplicação proposta.

Foi implementado na prática um elo de comunicação por espalhamento DSSS em banda básica utilizando microcontroladores PIC16F876, visando a comunicação entre dois PCs através de suas portas seriais. Os testes foram feitos com uma das placas com Tx e Rx em loopback, sendo que todo caracter digitado via software emulador de terminal no PC era espalhado em banda básica, transmitido e desespalhado pela outra placa. Essa outra placa fazia a retransmissão de volta com o caracter obtido espalhado novamente. A primeira placa devolveu com êxito todos os caracteres digitados. Outro teste realizado foi fazer uma placa transmitir seqüencialmente o mesmo caracter para medidas de espectro dos sinais espalhados

(TX_DSSS) e sem espalhamento (TX) com o osciloscópio no modo FFT (Fast Fourier Transform). Essas medidas comprovaram o espalhamento espectral do sinal recebido. Medidas no domínio tempo foram realizadas para comprovar as temporizações calculadas dos blocos de transmissão e recepção de bits (sem espalhamento) ou chips (com espalhamento) sendo todos os tempos comprovados. Conforme visto, devido a limitações de velocidade de execução do software nos microcontroladores PICs, a velocidade de comunicação com o PC ficou no máximo de 1200bps. O programa, cuja listagem se encontra no anexo 1, foi testado com a interligação via fios entre as duas placas PIC, portanto em banda básica.

Com o uso de um par de módulos Tx/RX FM (417.9KHz) para comunicação de dados até 20kbps foram feitas medidas com o analisador de espectro portátil Anritsu MS2711. Como haviam disponíveis apenas um par, o teste teve de ser feito com uma das placas transmitindo o mesmo caracter seqüencialmente. Devido à resolução mínima de 10KHz desse analisador de espectro, as avaliações de alargamento ficaram comprometidas, uma vez que o próprio instrumento introduziu alargamentos.

Algumas alternativas a serem exploradas no prosseguimento deste trabalho são discutidas a seguir.

A velocidade de transmissão de dados do usuário pode ser aumentada através da implementação das soluções propostas para simulação ou da arquitetura de software testada com o microcontrolador PIC16F876 em microcontroladores mais rápidos (por exemplo Scenix-SX28) ou em circuitos integrados do tipo FPGA. Nesta segunda opção, os algoritmos serão implementados em hardware e não em software, podendo se ganhar muito em termos de velocidade.

Uma possibilidade interessante para a implementação física da parte de RF é o uso de componentes utilizados nos canais de FI de TV (CIs e filtros passa banda), na frequência de 45MHz. Esta opção é atraente porque além do baixo custo e disponibilidade, pode se implementar amplificadores com banda de alguns MHz, portanto com banda suficientemente larga para implementação de sistemas SS. Neste caso, a transmissão de bytes/caracteres via espalhamento DSSS poderá ser feita com modulação DBPSK, que foi a utilizada na simulação e que apresenta características mais adequadas.

Uma outra alternativa interessante a ser testada será a transmissão através de infravermelho, com LEDs e fotodetectores de baixo custo usados em sistemas de controle remoto. Visando avaliar a possibilidade de operação de vários sistemas de comunicação utilizando infravermelho emitindo no mesmo comprimento de onda e na mesma área, poderão ser realizados testes quanto à interferência entre estes vários elos de comunicação, usando códigos de espalhamento diferente.

Também o software deve ser melhorado através do estudo e implementação de um protocolo adequado para transmissões desse tipo. Esse protocolo poderia, por exemplo, emular, para os computadores PCs nas pontas, um cabo físico RS232 direto através de controle de fluxo em hardware (RTS/CTS) ou software (XON/XOFF) tal como feito em modems. Nos testes feitos nesse trabalho, a transmissão foi feita caracter a caracter.

Quanto à tecnologia, a tendência atual e futura é de um maior crescimento das aplicações sem fio com espalhamento espectral, tanto em pequenas distâncias (p. ex: tecnologia Bluetooth) como em grandes sistemas (p. ex: W-CDMA na terceira geração dos sistemas celulares). Esse crescimento deve-se à maior eficiência de uso do escasso espectro eletromagnético disponível, que é o maior objetivo para ampliação dos serviços de comunicações.

REFERÊNCIAS

- [1] LAINE, R. U. et al. Aurora 2400 Spread Spectrum Digital Radio for Unlicensed T1/E1 Transport in the 2.4GHz ISM Band Application. Digital Microwave Link Engineering Technical Paper Collection, Redwood Shores-CA EUA, Harris Corporation, v. 1, p. 1-9, Set. 1998.
- [2] JONES, G. et al. Spread Spectrum Update. Tucson-AZ EUA, Tucson Amateur Packet Radio Corporation, 1998.
- [3] PRISM HFA3824A Direct Sequence Spread Spectrum Baseband Processor. Milpitas-CA EUA, Intersil Corporation, 1998.
- [4] PETERSON, R. L. et al. Introduction to Spread Spectrum Communications. Englewood Cliffs-NJ EUA, Prentice Hall, 1995.
- [5] ALDERTON, M. The Bluetooth Question. Disponível em: <http://rfdesign.com/ar/radio_blueetooth_question/index.htm>. Acesso em: 2 de novembro de 2002.
- [6] AURORA 2400 Reference Manual. Redwood Shores-CA EUA, Harris Corporation, 1998.
- [7] VIGODA, B. A Nonlinear Dynamic System for Spread Spectrum Code Acquisition. Cambridge-MA EUA, 1999, Dissertação de Mestrado, MIT.
- [8] PIC 16F87X 28/40-Pin 8-bit CMOS FLASH Microcontrollers Data Sheet. Chandler-AZ EUA, Microchip Technology Inc, 2001.
- [9] LISBOA, J. A.; FERNANDES, J. J. G. Simulação MatLab e Implementação de um Sistema Spread Spectrum no Microcontrolador PIC. In: SIMPÓSIO BRASILEIRO DE MICROONDAS E OPTOELETRÔNICA, 10., 2002, Recife – PE. Anais... Recife-PE, E. Fontana e AJB de Oliveira, 2002. p. 499-503.

- [10] Tecnologia 3G Visão Geral da Tecnologia 2.5/3G. In: SEMINÁRIO DE INSTRUMENTAÇÃO E MEDIDAS, 1., 2000, Curitiba-PR. Agilent Technologies, 2000. 1 CD-ROM.
- [11] ANATEL. Norma N° 12/96 Condições de Uso de Frequências nas Faixas de 902-928 MHz, 2400-2483.5 MHz e 5725-5850 MHz, por Equipamentos de Radiocomunicação empregando a Técnica de Espalhamento Espectral. Brasília-DF, 1996.
- [12] 74HC/HCT4046A Phase-Locked-Loop with VCO. Eindhoven Holanda, Philips Semiconductors, 1997.

APÊNDICE A:

LISTAGEM DO PROGRAMA FONTE UTILIZADO PARA A IMPLEMENTAÇÃO
FÍSICA DO SISTEMA PIC16F876.

7 APÊNDICE A: LISTAGEM DO PROGRAMA FONTE UTILIZADO PARA A IMPLEMENTAÇÃO FÍSICA DO SISTEMA PIC16F876

```

Listagem do programa fonte dsss.c:
//Autor: Julio Fernandes
//
// Programa fonte para o transmissor/receptor DSSS (PIC 16F876)
//
//

#include <16f876.H>
#include delay(clock=20000000)
#include rs232(baud=1200, xmit=PIN_B1, rcv=PIN_B0, INVERT)
#include fuses HS,NOWDT, NOPROTECT, NOBROWNOUT, LVP
char tx_dsss;
char rx_dsss;
char tx_buffer[8];          // tabela de armazenamento do caracter espalhado a
transmitir //
char rx_buffer[8];        // tabela de armazenamento do caracter espalhado recebido //
const char pn[8]={ 0xBF, 0xCA, 0xDC, 0xA5, 0xB1, 0xB7, 0x36, 0x89
};          // tabela para geração da seqüência PN //

char rx_buf;
char t_test;
char r_test;
char d_test;
char rx;
BOOLEAN fim_tx;          // variável que indica fim de transmissão de um bloco de 64
chips //
BOOLEAN novo;          // variável que indica fim da recepção de um bloco de 64 chips
//
BOOLEAN n=0;
int ki;
int kt;
int t;
int r;
int rc;
int status;
char pq;
// interrupção interna / timer 0 p/ recepção no pino RA1 (3) e transmissão bit a bit via pino
RA0 (2) //
#include int_rtcc
void rtcc_interrupt() {
    int k;
    char w;
    char l;
    BOOLEAN q;
    if ((status==3)||((status==4)) {q=input(PIN_A1);
        // se está nos estados de recepção (3 e 4), executa
        leitura//
        rc++;
        if(q==1)          {rx_dsss=rx_dsss|r_test;}          else
        {l=r_test^0xff;rx_dsss=rx_dsss&l;}

        r_test = (r_test*2);
        if(r_test==0x100) {
            for(k=0;k<7;k++) {
                rx_buffer[k]=rx_buffer[k+1];
            }
            rx_buffer[7]=rx_dsss;
    }
}

```

```

        r_test=0x01;
        ki++;
        if (ki>7) {novo=1;ki=0;rx_buf=rx_buffer[0];}
                // indica fim da recepção de 64 chips //
                }
    }
escrita //
    if ((status==1)||((status==2)) { // se está nos estados de transmissão (1 e 2) faz
        t++;
        w = tx_buffer[0] & t_test;

        if (w==0x00) {
            output_low(PIN_A0);
            } else (output_high(PIN_A0));

            t_test = (t_test*2);
            if (t_test==0x100) {

                t_test=0x01;
                tx_dsss=tx_buffer[0];
                for(k=0;k<7;k++) {
                    tx_buffer[k]=tx_buffer[k+1];
                }
                tx_buffer[7]=tx_dsss;
                kt++;
                if(kt>7) {fim_tx=1; // fim de transmissão de 64
chips //
                    kt=0;}
            }
        }
    }

//interrupção externa no pino RB4 (25) ligada junto da entrada rx_dsss no pino RA1(3)//
#include <avr/interrupt.h>
void rb_interrupt(){
    BOOLEAN m;
    m=input(PIN_B4);
    if(m==1&&n==0){
set_rtcc(0);enable_interrupts(INT_RTCC);disable_interrupts(INT_RB);}
    // essa parte detecta início de recepção de um bloco de chips para disparar timer0() //
    n=m;
    }

// Rotina principal //

void main(){
    char tx;
    int k;
    char test;
    char n_test;
    char w;
    int p;
    long int i;
    int pointer;

```

```

char buf_mem[8];
int init;
char cy;
char carry;
int corr;
int l;
char bit;
char x;
char w_test;
char s;
int novo2;
int fim_tx2;
test=0x01;
rx=0x00;
status=0;
fim_tx=1;

SETUP_ADC_PORTS(NO_ANALOGS);
setup_counters(RTCC_INTERNAL, RTCC_DIV_2);
// contador interno dividido por 2, valor máximo de taxa de interrupção em
9,765 KHz //

delay_ms(250);
printf("Inicio do programa \n\r");

while (TRUE) {
if(status==0) {
disable_interrupts(GLOBAL);
delay_ms(1);printf("Caracter a transmitir:\n\r");
output_low(PIN_A0);delay_ms(1000); // limpa saída tx_dsss //
rx=getch(); s=rx; // obtém caracter do teclado via serial//

t=0;r=0;n=0;

kt=0;

t_test=0x01;

fim_tx=0;
rx=0x00;
// a rotina abaixo faz o espalhamento do caracter 0x00 de acordo com a tabela pn //
for (k=0; k < 8; k++) {
w = rx & test;
if (w==0x00) {

tx_dsss = pn[k];
} else (tx_dsss = pn[k]^0xff);

test = (test*2);
if (test==0x100) {test=0x01;}
tx_buffer[k]=tx_dsss;

}
status++;
}

if(status==1){

```

```

        p=1;
        ki=0;
        t_test=0x01;
        r_test=0x01;
        d_test=0x01;
        novo=0;
        t=0; r=0;n=0;
pq=0x01;
init=0;
i=0;

enable_interrupts(GLOBAL);
enable_interrupts(INT_RTCC); // habilita interrupção para transmissão //
while(fim_tx==0) { }
        status++;fim_tx=0;
// transmissão do primeiro caracter (0x00) para sincronização do receptor//
}

if(fim_tx==1&&status==2){ t_test=0x01;
// entra nessa rotina após transmissão do segundo caracter 0x00 //
        fim_tx=0;
        test=0x01;
//monta a tabela tx_buffer com o caracter obtido espalhado pela seqüência pn //
        for (k=0; k < 8; k++) {
                w = s & test;
                if (w==0x00) {
                        tx_buffer[k] = pn[k];
                } else (tx_buffer[k] = pn[k]^0xff);

                test = (test*2);
        }

        kt=0;

        while(fim_tx==0) { } // transmite terceiro caracter //
        disable_interrupts(INT_RTCC);
        delay_us(50);output_low(PIN_A0); // fim de transmissão //
        fim_tx=0;
        status=3;
}

if(status==3){
        enable_interrupts(INT_RB); // aguarda seqüência de 3 x 64 chips na
entrada//

        while(p){

if(novo) { novo=0;r++;
        if(fim_tx) {fim_tx=0;output_low(PIN_B2);}
        if (init==0) {
                for(k=0;k<8;k++) {
                        buf_mem[k]=rx_buffer[k];
                }
                init=1;
        }
        } // monta buffer de recepção //

if(init) {
        corr=0;

```

```

        for(k=0;k<8;k++) {
            if(buf_mem[k]==pn[k]) corr++;
        }
segunda vez//
if(corr>=6&&r==2) {p=0;r=0;} // detecta recepção da seq. pn correta pela

achar a fase//
if(corr<6) {carry=0x00;cy=0x00;i++; // se não, desloca bits da seq pn para

        for(k=0;k<8;k++) {
            w=buf_mem[k]&0x80;
            if(w==0x80) cy=0x01;
            buf_mem[k]=2*buf_mem[k];
            buf_mem[k]=(buf_mem[k]|carry)&0xff;
            carry=cy;
            cy=0x00;}
            buf_mem[0]=(buf_mem[0]|carry)&0xff;
        }

        if(i>64) {init=0;i=0;
        }
    }
}

pointer=i; // ponteiro indica a fase da seq pn recebida //
status=4;
}

if (novo==1&&status==4) { // obtém terceiro caracter espalhado (o desejado) //
    disable_interrupts(INT_RTCC);

if(r==0) {for(k=0;k<8;k++) {buf_mem[k]=rx_buffer[k];}

i=0; // a rotina abaixo ajusta a fase da seqüência pn de acordo com o ponteiro
ajustado antes //
while(i<pointer) {carry=0x00;cy=0x00;i++;
    for(k=0;k<8;k++) {
        w=buf_mem[k]&0x80;
        if(w==0x80) cy=0x01;
        buf_mem[k]=2*buf_mem[k];
        buf_mem[k]=(buf_mem[k]|carry)&0xff;
        carry=cy;
        cy=0x00;}
        buf_mem[0]=(buf_mem[0]|carry)&0xff;
    }

caracter //
    test=0x01; // a rotina abaixo faz o desespalhamento para encontrar o

    for(k=0;k<8;k++) {

        w=buf_mem[k]^pn[k];
        w_test=0x01;corr=0;
        for(l=0;l<8;l++) {x=w&w_test;
            if(x!=0) corr++;
            w_test=w_test*2;
        }
        if(corr>4) {bit=1;} else {bit=0;}

        if(bit==0) {n_test=test^0xff;tx=tx&n_test;}

        else {tx=tx|test;}
    }
}

```

```
        test = (test*2);
        }
    }

    delay_ms(100);printf(" \n\r");
    for (k=0;k<8;k++) {delay_ms(100);printf("rx_buffer[%d]: %X \n\r", k, rx_buffer[k]);

        delay_ms(100);printf("%c \n\r",tx);

        // as rotinas acima imprimem o caracter espalhado na forma de tabela (rx_buffer) e
        // desespalhado (tx) na saída TX RS232 //
        t=0; rc=0;

        ki=0;

        r_test=0x01;

        novo=0;
        status=0;
        i=0;
        fim_tx=1;
        test=0x01;

    }

}

// zera variáveis e volta ao estado zero para aguardar próximo caracter //
```


APÊNDICE B:

LISTAGEM DO PROGRAMA FONTE PARA GERAÇÃO DA TABELA PN DE 8 BYTES
A PARTIR DA SIMULAÇÃO DE UMA CADEIA DE 6 SHIFT REGISTERS COM
REALIMENTAÇÕES NAS SAÍDAS 1 E 6.

8 APÊNDICE B: LISTAGEM DO PROGRAMA FONTE PARA GERAÇÃO DA TABELA PN DE 8 BYTES A PARTIR DA SIMULAÇÃO DE UMA CADEIA DE 6 SHIFT REGISTERS COM REALIMENTAÇÕES NAS SAÍDAS 1 E 6

```
//Autor: Julio Fernandes
// Programa fonte para o gerador de tabela de codigos PN (PIC 16F876)

#include <16f876.H>
#use delay(clock=2000000)
#use rs232(baud=1200, xmit=PIN_B1, rcv=PIN_B0, INVERT)
#fuses HS,NOWDT, NOPROTECT, NOBROWNOUT, LVP
// Rotina principal //
void main(){

    char sr;
    char rs;
    int i;
    int k;
    char bit0;
    char bit;
    char x;
    char y;
    char z;
    char pn[8];
    char rpn[8];
    char test_x=0x01;
    char test_y=0x40;
    char test;
    char ntest;
    char itest;

    SETUP_ADC_PORTS(NO_ANALOGS);

    delay_ms(250);
    printf("Inicio do programa \n\r");
    sr=0x01;i=0;k=0;
    do{
        x=sr&test_x;if(x==0) {x=0;} else {x=1;}
        y=sr&test_y;if(y==0) {y=0;} else {y=1;}
        sr=sr*2;
        bit0=x^y;
        if (bit0==0) {sr=sr&0xfe;} else {sr=sr|0x01;}
        i++;
        if(i>7){ test=0x01;itest=0x80;rs=0x00;
            for(i=0;i<8;i++) {bit=sr&test;
                if (bit==test) {rs=rs|itest;} else {ntest=itest^0xff;rs=rs&ntest;}
                test=test*2;
                itest=itest/2;}

            rpn[k]=sr;
            pn[k]=rs;i=0;k++;}
        } while (k<8);

    for (k=0;k<8;k++) {delay_ms(100);printf("pn[%d]: %X \n\r", k, pn[k]);}
    for (k=0;k<8;k++) {delay_ms(100);printf("rpn[%d]: %X \n\r", k, rpn[k]);}
}
```

APÊNDICE C:

TRABALHO SOBRE MODULAÇÕES EM FASE E SPREAD SPECTRUM
APRESENTADO DURANTE O CURSO

9 APÊNDICE C: TRABALHO SOBRE MODULAÇÕES EM FASE E SPREAD SPECTRUM APRESENTADO DURANTE O CURSO

Spread Spectrum

Técnicas de Modulação

O objetivo desse trabalho é demonstrar a forma de modulação por espalhamento espectral na técnica DSSS (Direct Sequence Spread Spectrum). Para tanto serão feitas análises matemáticas dos sinais envolvidos assim como a apresentação dos resultados das simulações realizadas no software Matlab (módulo Simulink).

-José Júlio Fernandes

Introdução

Para fins de análise dos sinais serão consideradas as modulações digitais BPSK (BiPhase Shift Keying), pela simplicidade de análise, e DPSK (Diferencial Phase Shift Keying) que é a mais utilizada nos modelos comerciais de rádios spread spectrum. Após a discussão dessas técnicas de modulação será visto a forma de espalhamento do espectro do sinal modulado e, por fim, serão analisados os diversos blocos necessários para a implementação de um sistema Spread Spectrum.

1) Técnicas de Modulação Digital

1.1 Modulação BPSK (Biphase Shift Keying)

A modulação BPSK é obtida pela inversão ou não de fase de acordo com o estado do sinal digital de entrada do modulador. Ou seja:

$$S_i(t) = A \cos(2\pi f_c t + (i-1)\pi + \lambda) \quad (1.1)$$

Onde λ é a fase inicial da portadora e t deve estar compreendido entre 0 e T (duração do símbolo) e i pode ser 0 ou 1.

O modulador pode ser entendido como uma chave controlada pelo sinal digital, selecionando entre uma entrada em fase e outra defasada em 180 graus ou a multiplicação do sinal digital codificado em variações de +1 ou -1 pela portadora.

Para a demodulação deve ser feita a recuperação da portadora na frequência e fase corretas para multiplicação do sinal recebido, como visto na equação a seguir:

$$R = A \cos(2\pi f_c t + (i-1)\pi + \phi) \times \cos(2\pi f_c t + \phi) \quad (1.2)$$

Onde ϕ é o atraso do sinal até o receptor e t deve estar compreendido entre 0 e T (duração do símbolo) e i pode ser 0 ou 1.

Após a filtragem do sinal R através de um filtro passa baixa, obtém-se o feixe de dados desejado, variando entre +1 e -1.

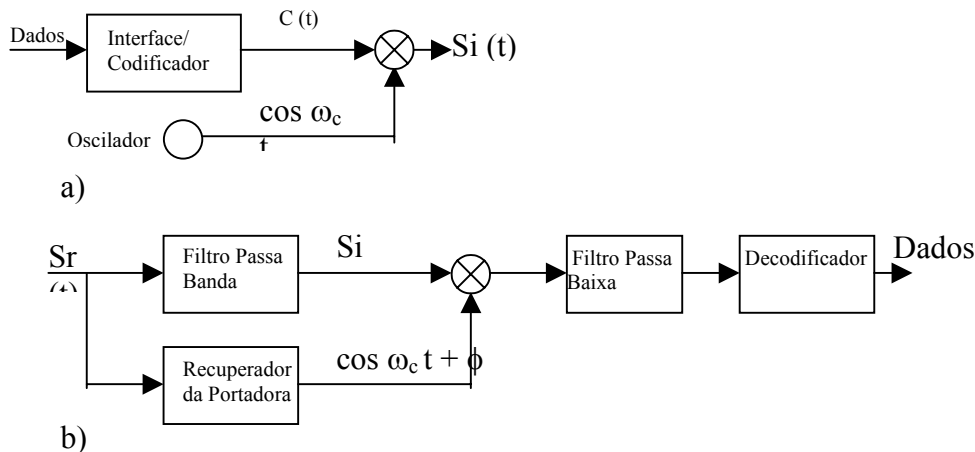


FIGURA 1: Diagrama de blocos BPSK: a) modulador b) demodulador.

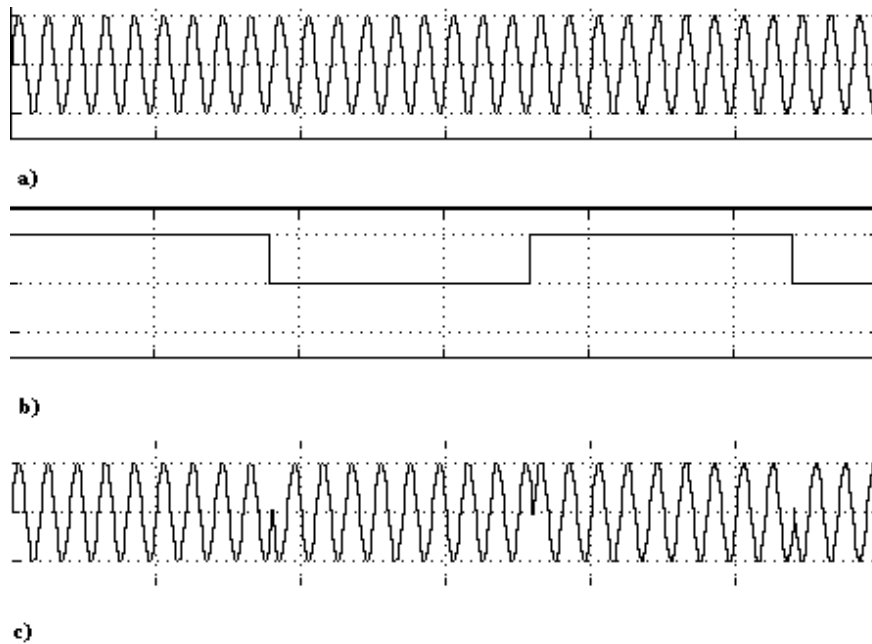


FIGURA 2: Sinais BPSK a) portadora b) sinal digital c) sinal modulado em BPSK.

Outros aspectos a serem considerados em uma modulação digital são a eficiência espectral e a probabilidade de erro de bit mínima em meios sob a ação de ruído branco gaussiano médio (AWGN) e fading (Raileigh , log normal, etc) tal como o ar em sistemas de rádio.

- Eficiência espectral: mede o quanto de taxa de dados em bps é transmitido para uma determinada largura espectral em Hz para um esquema de modulação.

Para tanto é necessário saber o espectro ocupado por um sinal modulado digitalmente. Sabe-se que o espectro de um pulso retangular de largura T é:

$$S(f) = AT \operatorname{sinc}(fT) \quad (1.3)$$

Onde: A = amplitude do pulso

T = largura do pulso

$\operatorname{Sinc}(x) = \sin(x) / x$

Se o pulso estiver modulando uma portadora cossenoidal de frequência f_0 o espectro será:

$$S(f) = AT/2 \{(\operatorname{sinc}((f - f_0)T) + \operatorname{sinc}((f + f_0)T))\} \quad (1.4)$$

A largura de banda do lóbulo principal desse espectro será:

$$B = 2/T \text{ em Hz}$$

Portanto a eficiência espectral será de 0.5 bps / Hz. uma vez que cada bit ocupa uma largura de tempo igual a T.

Probabilidade de Erro de Bit mínima: mede a probabilidade de ocorrência de bits errados de acordo com a relação sinal ruído para um esquema de modulação em ambientes sob o efeito de ruído do tipo AWGN e/ou fading.

Para a modulação BPSK essa probabilidade é dada por:

$$P_E = e^{-z} / 2 (\pi z)^{1/2} \text{ para } z \gg 1 \quad (1.5)$$

Onde $z = E_b / N_0$ (relação sinal-ruído)

1.2 Modulação QPSK e DQPSK

A escolha do tipo de modulação depende de dois fatores: eficiência de espectro (transmitir mais bits por hertz) e de potência (possuir menor probabilidade de erro em meios com baixa relação sinal - ruído). Além disso deve ser levado em conta a simplificação do lado receptor. Será analisado a seguir a modulação QPSK e DQPSK.

A) Modulação QPSK (Quaternary Phase Shift Keying)

A modulação QPSK pode ser considerada um caso especial da PSK, onde a informação é codificada na fase da portadora, como visto na equação abaixo:

$$S_i(t) = A \cos(2 \pi f_c t + (i-1) \pi/2 + \lambda) \quad (1.6)$$

Onde λ é a fase inicial da portadora e t deve estar compreendido entre 0 e T (duração do símbolo) e i varia de 0 a 3.

Cada uma das quatro fases possíveis corresponde a um par de bits de informação.

Definindo ϕ_i a fase instantânea do sinal modulado, temos:

$$\phi_i = (i-1) \pi/2 + \lambda \quad (1.7)$$

Então obtemos de (1.7),

$$S_i(t) = A \cos(2 \pi f_c t + \phi_i) \quad (1.8)$$

Expandindo a função cosseno na equação (1.8), temos:

$$S_i(t) = I_i A \cos(2 \pi f_c t) - Q_i A \sin(2 \pi f_c t) \quad (1.9)$$

Onde

$$I_i = \cos \phi_i \quad \text{e} \quad Q_i = \sin \phi_i \quad (1.10)$$

O sinal descrito em (1.9) pode ser visto como duas portadoras em quadratura com amplitudes $A \cos \phi_i$ e $A \sin \phi_i$, variando de acordo com as fases transmitidas em cada intervalo de sinalização.

Para que o sinal seja demodulado, deve ser feita a multiplicação do sinal QPSK recebido pela portadora recuperada do próprio sinal.

A eficiência espectral em QPSK é obtida por:

$$R_b / B = 0.5 \log_2 M$$

onde $M = 4$ p/ QPSK (número de estados possíveis para uma dada transição no sinal).

Logo, a eficiência espectral é de 1 bps / Hz.

A probabilidade de erro de símbolo é dada por:

$$P_s = 2 Q \left\{ \left(2 E_b / N_0 \right)^{1/2} \right\}$$

(1.11)

Onde $Q(x)$ é a função de distribuição Q (tabelada).

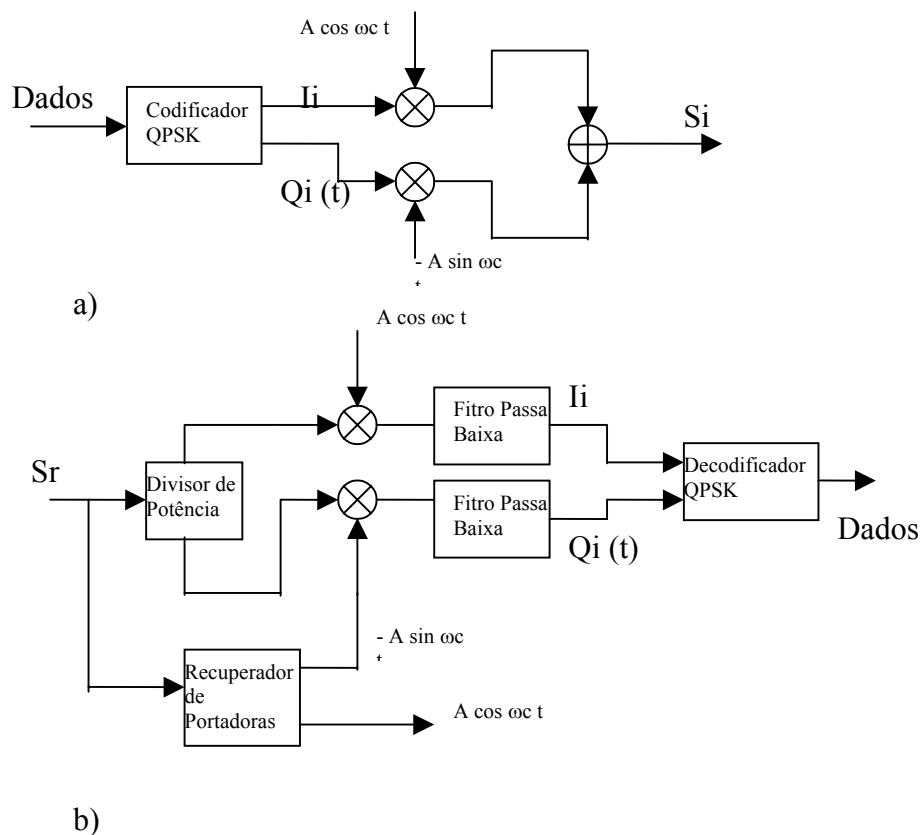


FIGURA 3: Diagrama de blocos QPSK: a) modulador b) demodulador.

B) Modulação DQPSK (Differential Quaternary Phase Shift Keying)

A modulação DQPSK pode ser considerada um caso especial da PSK, onde a informação é codificada em mudanças na fase corrente da portadora, como visto nas equações abaixo:

$$\phi_i = \phi_{i-1} - \Delta\phi_i \quad (1.12)$$

onde $\Delta\phi_i$ corresponde à mudança na fase anterior ϕ_{i-1} de acordo com a informação a ser transmitida. Usando a equação (1.5) aplicada à (1.6), temos:

$$\begin{aligned} I_i &= I_{i-1} \cos \Delta\phi_i - Q_{i-1} \sin \Delta\phi_i \\ Q_i &= I_{i-1} \sin \Delta\phi_i - Q_{i-1} \cos \Delta\phi_i \end{aligned} \quad (1.13)$$

Onde $I_{i-1} = \cos \phi_{i-1}$ e $Q_{i-1} = \sin \phi_{i-1}$ são as amplitudes do intervalo de símbolo prévio.

$$S_i(t) = I_i A \cos(2\pi f_c t) - Q_i A \sin(2\pi f_c t) \quad (1.14)$$

A equação (1.9) também se aplica à modulação DQPSK como demonstrado na equação 1.14. Se λ for escolhido $\pi/4$, as amplitudes das portadoras em quadratura podem assumir os seguintes valores: 0, +/- A, +/- A/ $\sqrt{2}$.

Tabela de codificação $\pi/4$ DQPSK:

MSB	LSB	$\Delta\phi_i$
0	0	$\pi/4$
0	1	$3\pi/4$
1	1	$-3\pi/4$
1	0	$-\pi/4$

O modulador a ser utilizado para QPSK e DQPSK é basicamente o mesmo, (implementa integralmente a equação 1.9) diferenciando-se em que os sinais I e Q são proporcionais à fase absoluta codificada em QPSK e à diferença de fase entre a atual e a anterior no DQPSK.

Já no demodulador, na técnica QPSK é necessário um estágio de recuperação de portadora para obter os sinais I e Q. Na técnica DQPSK, não é necessária a recuperação da portadora, bastando apenas usar no lugar dessa o próprio sinal atrasado na ordem de grandeza da duração de cada símbolo recebido.

A eficiência espectral é de 1 bps / Hz como em QPSK.

Apesar de QPSK possuir melhor performance quanto à taxa de bits errados (BER), a técnica DQPSK é bastante utilizada, uma vez que possui o demodulador simplificado pela não necessidade de recuperação do sinal da portadora.

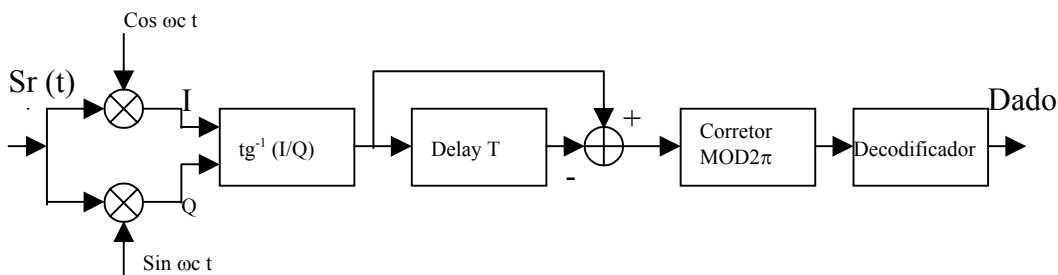


FIGURA 4: Diagrama de blocos do demodulador DQPSK.

2) Técnicas de Modulação Spread Spectrum

2.1 Sequência Direta com Modulação BPSK

A forma mais simples de obter-se o espalhamento espectral de um sinal modulado digitalmente é o uso de seqüência direta com modulação BPSK. Para tanto é feita a multiplicação do sinal modulado em BPSK por um trem de pulsos (-1 ou +1) variando de acordo com uma seqüência pseudo aleatória. O resultado é um sinal com inversões de fase não apenas em função do sinal digital modulante, mas também da seqüência pseudo aleatória de espalhamento. A demodulação desse sinal será feita apenas com a correta seqüência pseudo aleatória sendo reproduzida no receptor, inclusive estimando a fase em que o sinal recebido se encontra (variável de acordo com o caminho de propagação). Para a análise matemática o sinal modulado em BPSK a ser transmitido fica da forma:

$$S_d(t) = A \cos(\omega_0 t + \theta_d(t)) \quad (1.15)$$

Onde S_d é o sinal modulado em fase, A e ω_0 são a amplitude e a freqüência angular da portadora e $\theta_d(t)$ é a modulação em fase de acordo com o dado transmitido.

Seja $c(t)$ a seqüência pseudo aleatória chaveando entre -1 e +1, obtemos o sinal spread spectrum $S_t(t)$:

$$S_t(t) = A c(t) \cos(\omega_0 t + \theta_d(t)) \quad (1.16)$$

Esse é o sinal a ser transmitido por um caminho (na análise, sem distorção) que apresentará um atraso de propagação T_d . No lado receptor, aparecerá esse sinal junto com alguma forma de interferência e ruído do tipo AWGN que, como será visto mais adiante, serão minimizados na etapa de demodulação spread spectrum (despreading).

Para que possa ser recuperado o sinal modulado em fase deve ser feita a multiplicação do sinal recebido por uma réplica da seqüência pseudo aleatória que inclua a estimativa do atraso de propagação T_d . O sinal na saída do bloco de “despreading” fica da forma:

$$S_{de}(t) = A c(t - T_d) c(t - T_{de}) \cos(\omega_0 t + \theta_d(t - T_d) + \phi) \quad (1.17)$$

Onde ϕ corresponde a uma fase aleatória do sinal. S_{de} . Se a estimativa do atraso de propagação T_{de} estiver correta, o produto $c(t - T_d) c(t - T_{de})$ fica igual a 1 sempre, uma vez que esses sinais assumem apenas os valores +1 e -1. Portanto $S_{de}(t)$ fica praticamente igual a $S_d(t)$ no lado do transmissor, apresentando apenas o atraso de propagação T_d e uma fase aleatória ϕ . Notar que $S_d(t)$ pode apresentar qualquer modulação em fase, não apenas BPSK. O sinal digital é obtido pela demodulação desse sinal de acordo com as técnicas anteriormente vistas.

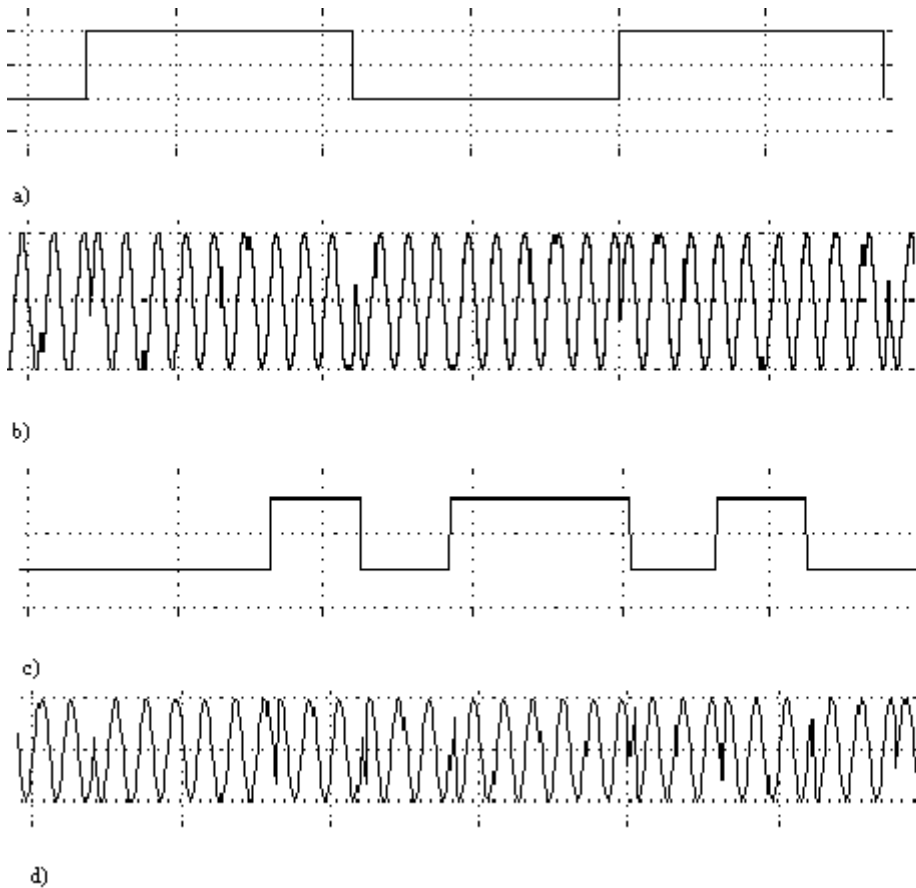


FIGURA 5: Sinais BPSK a) sinal digital b) sinal modulado em BPSK c) seqüência pseudoaleatória d) sinal spread spectrum DSSS

Antes de ser feita a análise do receptor quanto a presença de ruído e/ou interferência, será visto o espectro de potência do sinal spread spectrum.

O espectro do sinal spread spectrum é semelhante ao visto para o sinal BPSK, apenas trocando o valor de T (duração de um bit) pelo tempo T_c (duração de um chip). Chip corresponde à duração de cada estado possível na seqüência pseudo aleatória.

$$S_t(f) = \frac{1}{2} P T_c \{(\text{sinc}^2((f - f_0) T_c) + \text{sinc}^2((f + f_0) T_c))\} \quad (1.18)$$

Como T_c é muito menor que T , a amplitude de potência é reduzida pelo fator T_c/T e a largura de banda $B = 2/T$ é ampliada pelo fator T/T_c . No lado receptor, quando houver correlação entre $c(t - T_d)$ (embutida no sinal recebido) e $c(t - T_{de})$ (estimada) haverá o efeito contrário, a amplitude de potência será aumentada por T/T_c e o espectro será estreitado pelo fator T_c/T .

Quanto ao ruído e aos sinais interferentes, por não possuírem embutido neles a seqüência pseudo aleatória a ser usada na etapa de “despreading”, o fator $c(t - T_{de})$ não será cancelado e esses sinais sofrerão a ação de espalhamento, reduzindo a potência por T_c/T e alargando o espectro por T/T_c . Notar que esse fator representa uma margem adicional ao receptor quanto ao ruído e interferência. Define-se Ganho de Processamento essa relação:

$$G_p = 10 \log(T/T_c) \quad (1.19)$$

Outra forma de análise do efeito de espalhamento do espectro é através do teorema de Shannon:

$$C = W \log_2 [1 + S/N] \quad (1.20)$$

Onde C é a taxa de bits máxima comportada pelo canal, W a largura espectral do sinal modulado e S/N a relação sinal ruído.

Como pode ser visto, para que possamos transmitir uma determinada taxa de bits em um meio suscetível a ruído e/ou interferência, pode ser feito o aumento da potência do sinal S ou o aumento da largura espectral W . Porém o aumento da potência não produzirá grandes efeitos na taxa de bits máxima por causa do termo \log_2 , sobrando, portanto, o alargamento do espectro do canal. Se for suficientemente alargado o espectro podemos transmitir uma determinada taxa de bits até mesmo em um meio com relação sinal/ruído menor que a unidade (potência de ruído maior que a de sinal).

2.2 Sequência Direta com Modulação QPSK

Embora o melhor aproveitamento do espectro disponível não seja a principal preocupação em sistemas spread spectrum, às vezes é interessante utilizar uma modulação mais eficiente para minimizar a probabilidade de detecção de sinais por outros receptores (aplicações militares). O sinal em QPSK ficará da forma:

$$S_t(t) = A c_1(t) \cos(\omega_0 t + \theta_d(t)) - A c_2(t) \sin(\omega_0 t + \theta_d(t)) \quad (1.21)$$

Onde $c_1(t)$ e $c_2(t)$ podem ser seqüências totalmente diferentes entre si, porém devem estar em quadratura de fase.

3) Principais Blocos Necessários para Modulação e Demodulação Spread Spectrum

3.1 Transmissor Spread Spectrum BPSK

Pelo que foi visto até aqui, para implementar um circuito transmissor spread spectrum precisamos dos seguintes blocos:

e) Gerador de Sequência Pseudo Aleatória: é um circuito digital composto de vários estágios registradores de deslocamento (shift registers) com realimentação de determinadas saídas para a entrada através de uma lógica OR – exclusiva. O número de estágios (n) e as saídas a serem realimentadas são determinadas de acordo com o código da seqüência pseudo aleatória desejado. O número de estados possível é $2^n - 1$, uma vez que o estado com todas as saídas em zero não pode ocorrer para não travar o gerador. Os estados de saída variam entre -1 e $+1$ após a codificação de linha.

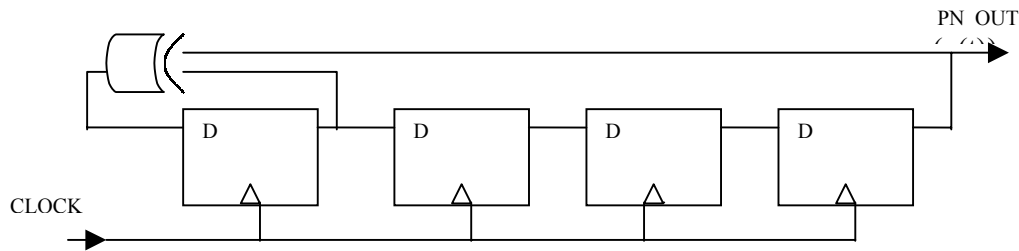


FIGURA 6: Diagrama de blocos do gerador de seqüências pseudo aleatórias.

f) Misturador Spread Spectrum: é um circuito que faz a multiplicação do sinal modulado $S_d(t)$ pela seqüência pseudo aleatória $c(t)$. Pode ser implementado por uma ponte de diodos cuja polarização varia de acordo com a seqüência pseudo aleatória.

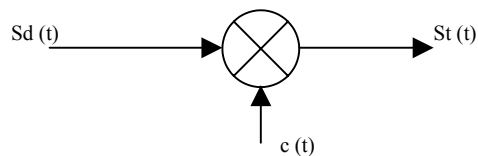


FIGURA 7: Diagrama de blocos do misturador Spread Spectrum.

g) Modulador em Fase: é o circuito que modulará o sinal digital a ser transmitido em fase, podendo ser BPSK, QPSK, etc. Se for usada a modulação BPSK, por facilidade pode-se fazer a soma OR exclusiva da seqüência pseudo aleatória com o sinal digital a ser transmitido antes do estágio misturador Spread Spectrum. A entrada desse estágio deve ser compatibilizada com a interface de dados do usuário, V.35, G.703, Ethernet IEEE 801.1, etc.

h) Circuitos de FI e RF: uma vez que o processamento de espalhamento espectral é feito em FI (Frequência Intermediária), o sinal $S_t(t)$ deve passar por estágios de filtragem passa banda em FI, com a largura suficiente para o espalhamento, e então ser feita a conversão de frequência para cima pelo misturador de RF. A saída do misturador já estará na frequência apropriada para transmissão pelo meio ar, após as etapas de filtragem passa banda, amplificação de RF, linha de transmissão (cabo coaxial ou guia de onda) e antena (tipicamente uma parabólica para enlaces de microondas Spread Spectrum, ou Omnidirecional para sistemas ponto-multiponto de acesso Spread Spectrum, tipo dados ou celular CDMA).

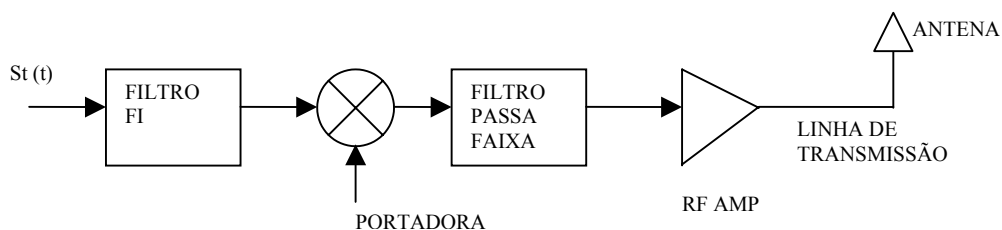


FIGURA 8: Diagrama de blocos da etapa de FI e RF.

3.2 Receptor Spread Spectrum BPSK

Pelo que foi visto até aqui, para implementar um circuito receptor Spread Spectrum precisamos dos seguintes blocos:

e) Circuitos de RF e FI: o sinal é captado por uma antena (parabólica ou omni), passa pela linha de transmissão (guia ou coaxial) até o estágio “front end” de amplificação de baixo ruído. Após essa etapa, o sinal recebido é convertido em frequência para baixo por um misturador de RF, gerando um sinal em FI. Esse sinal FI passa por uma filtragem passa banda antes de entrar no bloco de “despreading”.

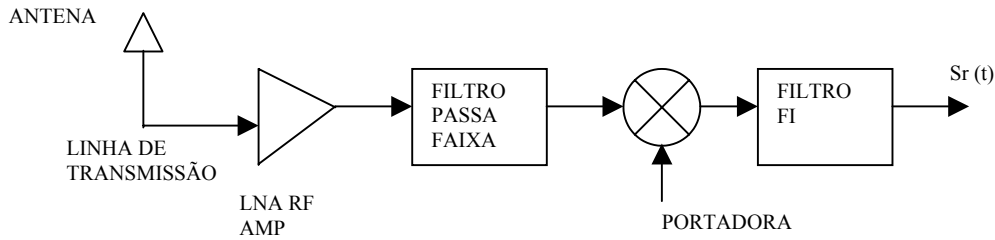


FIGURA 9: Diagrama de blocos da etapa de FI e RF do receptor.

f) Misturador Spread Spectrum: esse estágio fará a multiplicação do sinal recebido em FI pela seqüência pseudo aleatória. Essa seqüência deve ser gerada pelo mesmo tipo de circuito do transmissor com as mesmas realimentações programadas. Também deve ser estimada a fase da seqüência pseudo aleatória embutida no sinal recebido para a recuperação total do sinal modulado $S_{de}(t)$.

g) Bloco de Temporização e Sincronismo: a entrada desse bloco é o sinal recebido em FI, estando em uma malha de detecção de atraso de fase (Delay Locked Loop). Dessa forma a seqüência pseudo aleatória gerada por esse bloco já conterá o atraso de fase estimado e a frequência correta. Também é feita a recuperação da portadora, para posterior demodulação BPSK ou QPSK, por essa etapa.

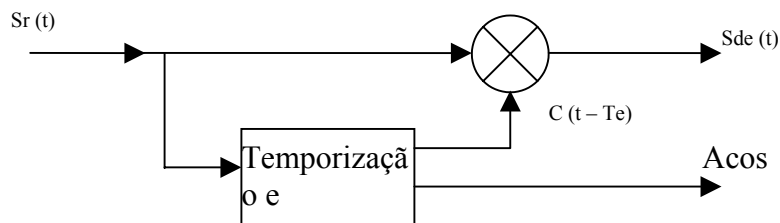


FIGURA 10: Diagrama de blocos das etapas misturadora e de temporização/sincronismo.

h) Demodulador: a demodulação em fase pode ser feita pela multiplicação do sinal modulado $S_{de}(t)$ pela portadora recuperada pelo bloco anterior ($A \cos \omega_c t$). Após a filtragem dos elementos de alta frequência, teremos o sinal em banda básica. Para a recuperação do sinal digital, esse sinal deve passar por um circuito decisor (comparador de tensão), sendo feita então a compatibilização com o tipo de interface do usuário.

Conclusão

Neste trabalho foram feitas as análises matemáticas dos sinais envolvidos no processo de espalhamento espectral em seqüência direta assim como a descrição dos blocos necessários para implementação do transmissor e receptor DSSS. Falta ainda um maior aprofundamento na análise dos circuitos componentes do sistema Spread Spectrum, ficando para um próximo trabalho juntamente com a análise do sistema Spread Spectrum tipo Frequency Hopping..

Bibliografia

- [1] Peterson, R. L. et al, "*Introduction to Spread Spectrum Communications*", Prentice Hall, 1995 .
- [2] Jones, G. et al, "*Spread Spectrum Update*", Tucson Amateur Packet Radio Corporation, 1998 .
- [3] Laine, R. U. et al, "*Aurora 2400 Spread Spectrum Digital Radio for Unlicensed T1/E1 Transport in the 2.4GHz ISM Band Application*", Harris Corporation, 1998 .
- [4] "*Aurora 2400 Reference Manual*", Harris Corporation , 1998 .

APÊNDICE D:

DQPSK $\pi/4$
IMPLEMENTAÇÃO E SIMULAÇÃO

10 APÊNDICE D: DQPSK $\pi/4$ IMPLEMENTAÇÃO E SIMULAÇÃO

DQPSK $\pi/4$

Implementação e Simulação

O objetivo desse trabalho é expor uma forma de implementação da técnica de modulação digital DQPSK $\pi/4$. Para tanto serão feitas as análises dos circuitos modulador e demodulador e dos sinais envolvidos, assim como a apresentação dos resultados das simulações realizadas no software Matlab (módulo Simulink).

-José Júlio Fernandes

Introdução

Para fins de análise dos sinais será considerada inicialmente a definição de modulação digital DQPSK (Diferencial Phase Shift Keying). Esse tipo de modulação (DQPSK $\pi/4$; 8 níveis) é usada para acesso de sistemas celulares DAMPS (TDMA-IS136) devido à maior robustez às variações de fase do sinal recebido/emitido pela antena do terminal móvel, uma vez que a propagação se dá por multi caminhos através de reflexões do sinal. A variação DQPSK (4 níveis) também é muito utilizada em rádios Spread Spectrum ponto-a-ponto e ponto-multiponto do tipo DSSS (seqüência direta), como pode ser visto em catálogos de fabricantes de chip sets DSSS. Será mostrado como podem ser implementados esses conceitos através de circuitos montados no módulo Simulink do software MatLab. E, por fim, serão analisadas as formas de onda obtidas por simulação dos circuitos modulador e demodulador.

1) Análise da Modulação Digital DQPSK

1.1 Modulação QPSK e DQPSK

A) Modulação QPSK (Quaternary Phase Shift Keying)

Inicialmente será analisada a modulação QPSK, onde a informação é codificada na fase da portadora, como visto na equação abaixo:

$$S_i(t) = A \cos(2\pi f_c t + (i-1)\pi/2 + \lambda) \quad (1.1)$$

Onde λ é a fase inicial da portadora e t deve estar compreendido entre 0 e T (duração do símbolo) e i varia de 0 a 3.

Cada uma das quatro fases possíveis corresponde a um par de bits de informação.

Definindo ϕ_i a fase instantânea do sinal modulado, temos:

$$\phi_i = (i-1)\pi/2 + \lambda \quad (1.2)$$

Então obtemos de (1.2),

$$S_i(t) = A \cos(2\pi f_c t + \phi_i) \quad (1.3)$$

Expandindo a função cosseno na equação (1.3), temos:

$$S_i(t) = I_i A \cos(2\pi f_c t) - Q_i A \sin(2\pi f_c t) \quad (1.4)$$

Onde

$$I_i = \cos \phi_i \quad \text{e} \quad Q_i = \sin \phi_i \quad (1.5)$$

O sinal descrito em (1.4) pode ser visto como duas portadoras em quadratura com amplitudes $A \cos \phi_i$ e $A \sin \phi_i$, variando de acordo com as fases transmitidas em cada intervalo de sinalização.

Para que o sinal seja demodulado, deve ser feita a multiplicação do sinal QPSK recebido pela portadora recuperada do próprio sinal.

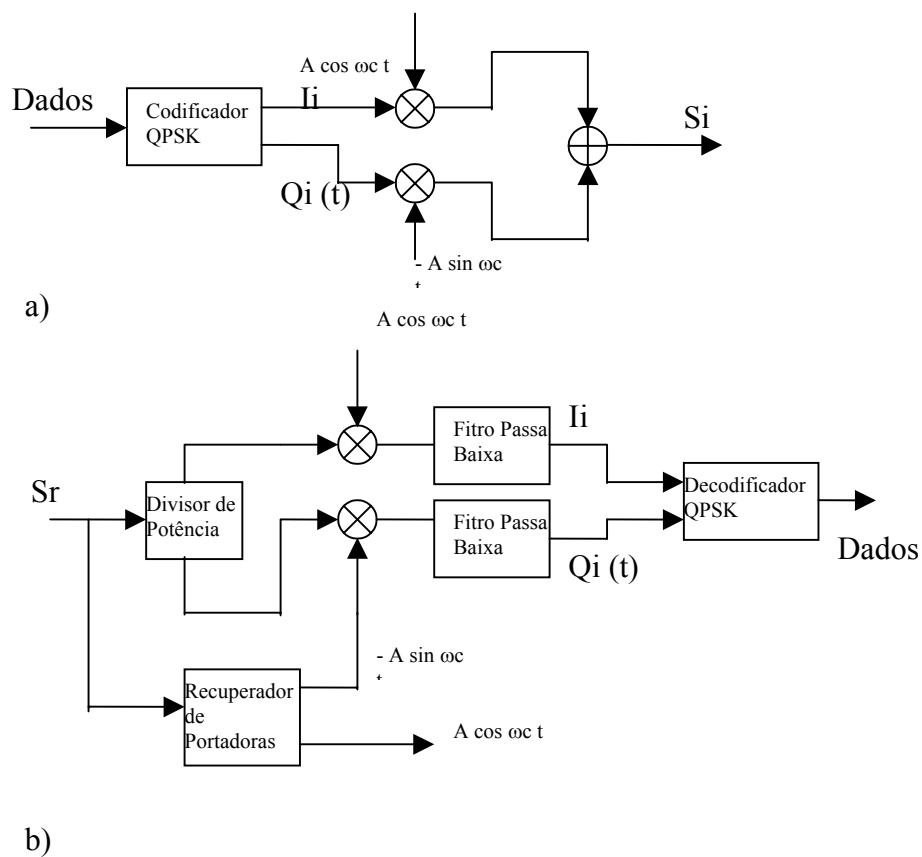


FIGURA 1: Diagrama de blocos QPSK: a) modulador b) demodulador.

C) Modulação DQPSK (Differential Quaternary Phase Shift Keying)

A modulação DQPSK pode ser considerada um caso especial da QPSK, onde a informação é codificada em mudanças na fase corrente da portadora, como visto nas equações abaixo:

$$\phi_i = \phi_{i-1} - \Delta\phi_i \quad (1.6)$$

onde $\Delta\phi_i$ corresponde à mudança na fase anterior ϕ_{i-1} de acordo com a informação a ser transmitida. Usando a equação (1.5) aplicada à (1.6), temos:

$$\begin{aligned} I_i &= I_{i-1} \cos \Delta\phi_i - Q_{i-1} \sin \Delta\phi_i \\ Q_i &= I_{i-1} \sin \Delta\phi_i - Q_{i-1} \cos \Delta\phi_i \end{aligned} \quad (1.7)$$

Onde $I_{i-1} = \cos \phi_{i-1}$ e $Q_{i-1} = \sin \phi_{i-1}$ são as amplitudes do intervalo de símbolo prévio.

$$S_i(t) = I_i A \cos(2\pi f_c t) - Q_i A \sin(2\pi f_c t) \quad (1.8)$$

A equação (1.4) também se aplica à modulação DQPSK como demonstrado na equação 1.8. Se λ for escolhido $\pi/4$, as amplitudes das portadoras em quadratura podem assumir os seguintes valores: 0, +/- A, +/- A/ $\sqrt{2}$.

TABELA 1: Tabela de codificação $\pi/4$ DQPSK

MSB	LSB	$\Delta\phi_i$
0	0	$\pi/4$
0	1	$3\pi/4$
1	1	$-3\pi/4$
1	0	$-\pi/4$

O modulador a ser utilizado para QPSK e DQPSK é basicamente o mesmo, (implementa integralmente a equação 1.8) diferenciando-se em que os sinais I e Q são proporcionais à fase absoluta codificada em QPSK e à diferença de fase entre a atual e a anterior no DQPSK.

Já no demodulador, na técnica QPSK é necessário um estágio de recuperação de portadora para obter os sinais I e Q. Na técnica DQPSK, não é necessária a recuperação da portadora, bastando apenas usar no lugar dessa o próprio sinal atrasado na ordem de grandeza da duração de cada símbolo recebido.

A eficiência espectral é de 1 bps / Hz como em QPSK.

Apesar de QPSK possuir melhor performance quanto à taxa de bits errados (BER), a técnica DQPSK é bastante utilizada, uma vez que possui o demodulador simplificado pela não necessidade de recuperação do sinal da portadora.

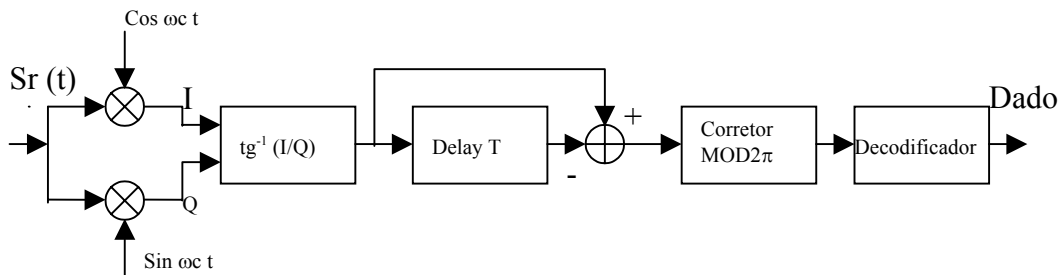


FIGURA 2: Diagrama de blocos do demodulador DQPSK.

2) Implementação do Modulador DQPSK

Como pode ser observado na equação 1.8, o sinal modulado em DQPSK é obtido pela soma de dois sinais senoidais em quadratura cujas variações de amplitude definem a fase do sinal resultante. Pode-se obter os valores possíveis para as amplitudes dos sinais I (em fase) e Q (em quadratura) pela análise do diagrama abaixo.

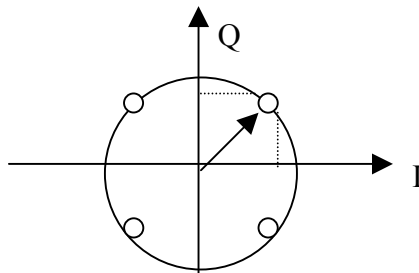


FIGURA 3: Diagrama de fasores para sinais modulados em fase

No caso da modulação DQPSK $\pi/4$ são possíveis oito fases com espaçamento de $\pi/4$ (45°) que são: 0° , 45° , 90° , 135° , 180° , 225° , 270° e 315° . Para compor essas fases, o modulador deve fazer a multiplicação das duas portadoras em quadratura ($\cos \omega t$ e $\sin \omega t$) por constantes de acordo com a tabela abaixo, já considerando o sinal negativo para a componente em quadratura.

TABELA 2: Fases possíveis, codificação em binário e os sinais I e Q necessários

Fase ($^\circ$)	F2	F1	F0	I	Q
0	0	0	0	1	0
45	0	0	1	0,707	-0,707
90	0	1	0	0	-1
135	0	1	1	-0,707	-0,707
180	1	0	0	-1	0
225	1	0	1	-0,707	0,707
270	1	1	0	0	1
315	1	1	1	0,707	0,707

Essa codificação é válida tanto para moduladores 8PSK (oito níveis absolutos) quanto para DQPSK $\pi/4$ (4 níveis diferenciais). Nesse caso deve ser feita a variação da fase de acordo com o par de bits a ser transmitido como mostrado na tabela 1. Esse par de bits é obtido por um conversor série paralelo de 2 bits cuja entrada é o trem de dados a transmitir. A codificação binária é mostrada na tabela 3, com o par de bits já codificado em gray.

TABELA 3: Tabela de codificação DQPSK para a fase do sinal modulado

Fase (°)	Anterior			Atual				
	g1	g0	F2	F1	F0	A2	A1	A0
0	0	0	0	0	0	0	0	1
45	0	0	0	0	1	0	1	0
90	0	0	0	1	0	0	1	1
135	0	0	0	1	1	1	0	0
180	0	0	1	0	0	1	0	1
225	0	0	1	0	1	1	1	0
270	0	0	1	1	0	1	1	1
315	0	0	1	1	1	0	0	0
0	0	1	0	0	0	0	1	1
45	0	1	0	0	1	1	0	0
90	0	1	0	1	0	1	0	1
135	0	1	0	1	1	1	1	0
180	0	1	1	0	0	1	1	1
225	0	1	1	0	1	0	0	0
270	0	1	1	1	0	0	0	1
315	0	1	1	1	1	0	1	0
0	1	0	0	0	0	1	1	1
45	1	0	0	0	1	0	0	0
90	1	0	0	1	0	0	0	1
135	1	0	0	1	1	0	1	0
180	1	0	1	0	0	0	1	1
225	1	0	1	0	1	1	0	0
270	1	0	1	1	0	1	0	1
315	1	0	1	1	1	1	1	0
0	1	1	0	0	0	1	0	1
45	1	1	0	0	1	1	1	0
90	1	1	0	1	0	1	1	1
135	1	1	0	1	1	0	0	0
180	1	1	1	0	0	0	0	1
225	1	1	1	0	1	0	1	0
270	1	1	1	1	0	0	1	1
315	1	1	1	1	1	1	0	0

O circuito capaz de fazer a codificação proposta na tabela acima é uma máquina de estados com a realimentação da fase anterior em binário e o par de dados a ser transmitido, codificado em gray, como entradas; e como saída a fase atualizada.

A fase codificada em binário deve ser convertida para os sinais I e Q por um conversor digital - analógico de acordo com a tabela 2. Esse conversor pode ser implementado por um

multiplexador analógico do tipo CMOS, cujas entradas são proporcionais às tensões 0V, +/- 0.707V e +/- 1V.

A multiplicação pelas portadoras em quadratura é feita por um circuito misturador que pode ser um circuito integrado específico para essa função ou um circuito composto de um estágio a transistor com a saída sintonizada na frequência da FI (frequência intermediária) e largura de faixa adequada ao esquema de modulação cujas entradas seriam a base e o emissor.

Após esse estágio as duas portadoras em quadratura moduladas são somadas produzindo o sinal DQPSK final a ser aplicado aos circuitos seguintes (etapas de RF).

Outros circuitos necessários são o recuperador de clock a partir do feixe de dados de entrada, o gerador de portadoras em quadratura e o filtro passa baixa de Nyquist para a saída do conversor digital – analógico.

O recuperador de clock deve ser implementado preferencialmente através de um circuito PLL devido à estabilidade de fase do clock regenerado, porém, para fins de simulação, foi utilizado um filtro passa banda de elevado fator de qualidade, sintonizado na frequência fundamental do sinal de clock para a simulação do modulador DQPSK.

O gerador de portadoras é constituído por um oscilador sintonizado na frequência intermediária com uma saída direta ($\cos \omega t$) e outra defasada de 90° ($\sin \omega t$), obtida por processos de integração ou derivação. Na simulação foi feita a derivação do sinal da fonte senoidal com o ajuste de amplitude $-1/\omega$ para compensar o aumento de amplitude e ajustar o sinal antes da soma das duas portadoras.

O filtro passa baixa de Nyquist tem a função de eliminar as variações bruscas do sinal de saída do conversor digital analógico (componentes de alta frequência) antes do estágio misturador a fim de limitar a largura de banda do sinal modulado resultante. Para fins de simulação, esse filtro não foi utilizado no circuito montado no Simulink.

No Anexo 1 encontram-se as figuras com os esquemas dos circuitos usados no modulador DQPSK para simulação no MatLab.

3) Implementação do Demodulador DQPSK

No primeiro estágio do demodulador deve ser feito o batimento (multiplicação) do sinal em FI de entrada pelas portadoras em quadratura. Pode ser demonstrado que na saída do bloco misturador surgirão sinais proporcionais aos sinais I e Q mais componentes senoidais com o

dobro da frequência intermediária. Após a filtragem dessas componentes de alta frequência por um estágio de filtro passa baixa, os sinais I e Q (que são proporcionais a fase corrente do sinal em FI) devem ser convertidos para a forma binária, uma vez que o processamento digital desses sinais é mais facilmente implementado (por exemplo, circuitos integrados do tipo FPGA).

O conversor analógico digital deve ser suficientemente rápido para a obtenção do sinal em FI digitalizado. Por exemplo, para um rádio DQPSK de 2Mbps, os sinais I e Q irão variar a uma taxa de 1×10^6 símbolos por segundo (1 símbolo corresponde a 2 bits). Esse conversor pode ser implementado por um array de comparadores de tensão de alta velocidade, referindo-se aos níveis possíveis para os sinais I e Q (tensões proporcionais a 0V, +/- 0.707V e +/- 1 V), desde que o sinal em FI seja mantido a uma amplitude constante por um circuito controlador de ganho na entrada do demodulador, cuja tensão de amostra pode ser um dos sinais I ou Q, por exemplo. Para a simulação no MatLab, foi implementado um array de 4 comparadores com um decodificador ligado às saídas, gerando os códigos binários correspondentes a cada nível de tensão de acordo com a tabela 4 abaixo.

TABELA 4: Tabela de codificação para os níveis dos sinais I e Q

I / Q	I2/Q2	I1/Q1	I0/Q0
-1	0	0	0
-0,707	0	0	1
0	0	1	0
0,707	0	1	1
1	1	0	0

Com os sinais I e Q codificados em binário fica mais fácil o processamento para a obtenção da fase corrente. Para tanto é seguido o fluxo demonstrado na figura 2, entrando com os valores binários de I e Q em uma tabela cuja saída é a fase correspondente ao arco tangente da divisão Q/I. Essa tabela, mostrada abaixo, pode ser implementada por um circuito combinacional.

TABELA 5: Tabela de obtenção da fase atual a partir dos sinais I e Q digitalizados

Q2	Q1	Q0	I2	I1	I0	F2	F1	F0	Fase
0	0	0	0	0	0	x	x	x	90
0	0	0	0	0	1	x	x	x	
0	0	0	0	1	0	0	1	0	
0	0	0	0	1	1	x	x	x	
0	0	0	1	0	0	x	x	x	
0	0	1	0	0	0	x	x	x	135
0	0	1	0	0	1	0	1	1	
0	0	1	0	1	0	x	x	x	
0	0	1	0	1	1	0	0	1	45
0	0	1	1	0	0	x	x	x	180
0	1	0	0	0	0	1	0	0	
0	1	0	0	0	1	x	x	x	
0	1	0	0	1	0	x	x	x	
0	1	0	0	1	1	x	x	x	
0	1	0	1	0	0	0	0	0	0
0	1	1	0	0	0	x	x	x	225
0	1	1	0	0	1	1	0	1	
0	1	1	0	1	0	x	x	x	
0	1	1	0	1	1	1	1	1	315
0	1	1	1	0	0	x	x	x	270
1	0	0	0	0	0	x	x	x	
1	0	0	0	0	1	x	x	x	
1	0	0	0	1	0	1	1	0	
1	0	0	0	1	1	x	x	x	
1	0	0	1	0	0	x	x	x	

Na tabela acima, os sinais de entrada estão codificados de acordo com a tabela 4 e os de saída com a tabela 2. Por exemplo, se tomarmos os valores binários da terceira linha, temos o sinal Q codificado em 000 (-1V na tabela 4) e o sinal I como 010 (0V na tabela 4). Pela tabela 2, obtemos a fase de 90° (010 nessa tabela), pois arco tangente de $1/0$ (∞) é 90° , não esquecendo o sinal trocado em Q devido ao fato de a portadora em quadratura ser subtraída na função descritiva do sinal DQPSK (equação 1.8).

A fase atual, obtida na saída do circuito combinacional $\arctg(Q/I)$ deve ser armazenada em uma memória (latch) pelo tempo de um símbolo a fim de ser subtraída da fase seguinte, obtendo-se a diferença de fase corrente. Nessa diferença de fase está codificado o par de dados desejado. Essa etapa é implementada por um circuito seqüencial com temporização a

partir do sinal de clock de símbolo (obtido na etapa de recuperação de relógio) e por um circuito subtrator de 3 bits.

O decodificador DQPSK é obtido pela tabela abaixo:

TABELA 6: Tabela de obtenção dos dados D1,D0 a partir da diferença de fase

Dif Fase	F2	F1	F0	D1	D0
45	0	0	1	0	0
135	0	1	1	0	1
315	1	1	1	1	1
225	1	0	1	1	0

Esses dados estão na forma paralela de 2 bits, sendo necessário serializar na taxa do clock de bit obtido no estágio de recuperação de relógio. O circuito que executa essa função é implementado por um conversor paralelo - série com um elemento flip flop na saída sincronizado no clock de bit.

Finalmente, para efeitos de simulação, o recuperador de relógio foi implementado por um filtro passa faixa de alto fator de qualidade (Q) sintonizado na frequência fundamental do clock de bit com entrada de um dos bits correspondentes ao sinal I digitalizado (na saída do conversor A/D). A saída do filtro é aplicada em um comparador de sinal, obtendo-se o sinal ClkB (bit) e, após a passagem por um divisor de frequência digital, obtém-se o sinal ClkS (símbolo).

No Anexo 1 encontram-se os diagramas dos circuitos utilizados para simulação do demodulador DQPSK. Os blocos de lógica combinacional e o subtrator não são apresentados pois apenas implementam com portas lógicas as tabelas acima descritas.

4) Análise dos Sinais Obtidos na Simulação

4.1 Simulação do Modulador DQPSK

Para a simulação do circuito modulador DQPSK foi usado como fonte de dados um circuito gerador de seqüência pseudo aleatória de 4 estágios. Para essa simulação foi normalizada a amplitude de 1V e frequência de 300 Hz para as portadoras em quadratura (FI)

e a taxa de 100bps para o feixe de dados a transmitir. Nas figuras a seguir podem ser visualizadas as formas de onda obtidas nos diversos estágios do circuito.

Tomando os dois primeiros bits a serem transmitidos do sinal Din, a partir da seta, obtemos a seqüência 00, que pela tabela 1 corresponde a um avanço de 45° na fase corrente. A fase atual é obtida pelo sinal F2,F1,F0 (111) que é de 315° , sendo a próxima de 000, isto é 0° . Portanto existe o avanço de 45° .

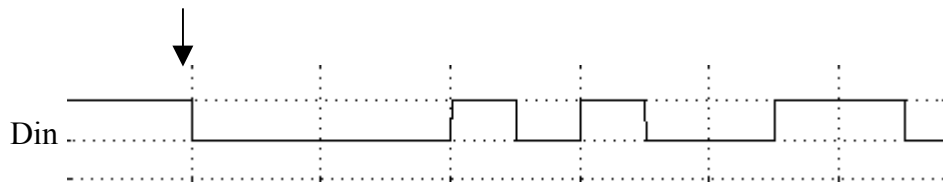


FIGURA 4.1.1: Sinal de saída do gerador de seqüência pseudo aleatória

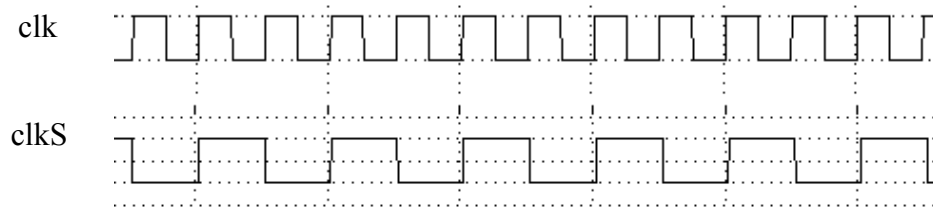


FIGURA 4.1.2: Sinais de saída do recuperador de relógio (clkB e clkS)

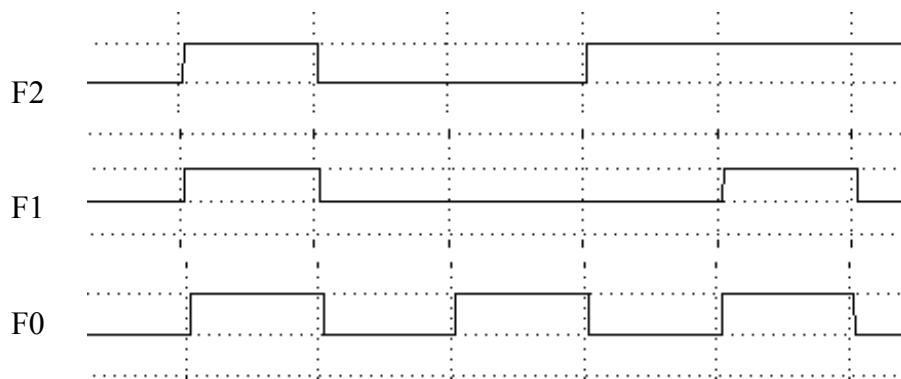


FIGURA 4.1.3: Sinais de saída da máquina de estados (F2, F1, F0)

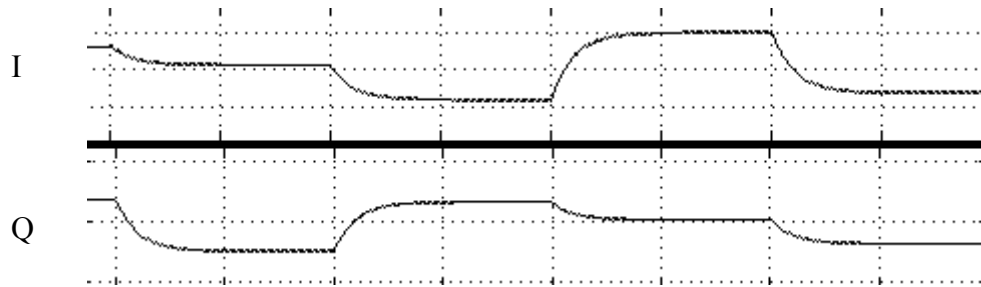


FIGURA 4.1.4: Sinais I e Q de saída do conversor digital analógico

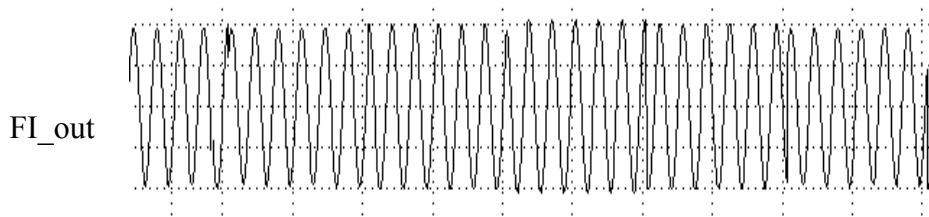


FIGURA 4.1.5: Sinal de saída do somador de portadoras em quadratura

4.2 Simulação do Demodulador DQPSK

Para a simulação do circuito demodulador DQPSK foi usado como fonte de sinal em FI modulado um bloco de função matemática cosseno com o fator ωt gerado a partir do passo de simulação do software Simulink somado a uma fase variável em passos de 45° . Por um fator de multiplicação pode-se programar esse passo para $+45^\circ$, $+135^\circ$, -45° e -135° .

Nessa simulação foi usado como fase variável um avanço de 135° . Também foram padronizadas as amplitudes máximas em 1V, tanto para o sinal em FI quanto para os sinais I e Q.

Como pode ser observado nos sinais de saída do conversor A/D, existem oscilações no sinal digital de saída devido a presença de flutuações remanescentes nos sinais I ou Q após a filtragem passa baixa. Essas oscilações podem ser eliminadas nos estágios seguintes pelo uso de flip flops sincronizados no sinal de clock de símbolo.



FIGURA 4.2.1: Sinal de saída do gerador de sinal DQPSK

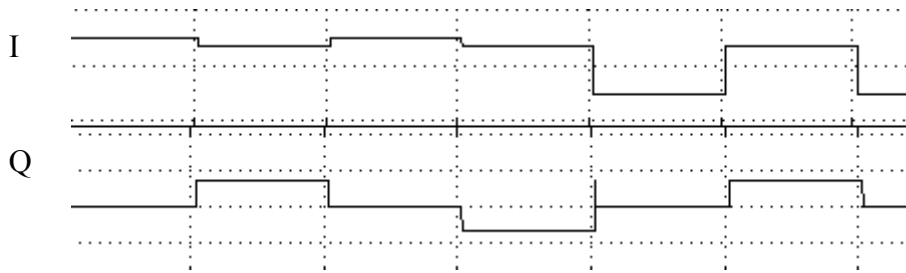


FIGURA 4.2.2: Sinais I e Q obtidos na saída do misturador

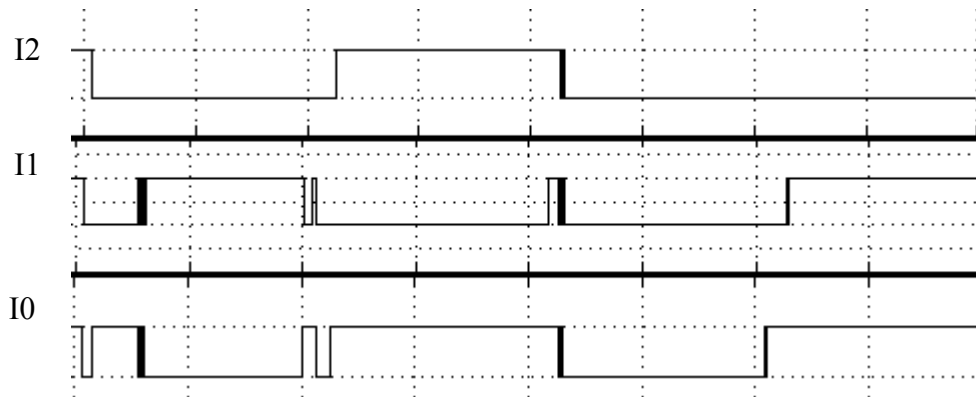


FIGURA 4.2.3: Sinais I2, I1 e I0 obtidos na saída do conversor A/D

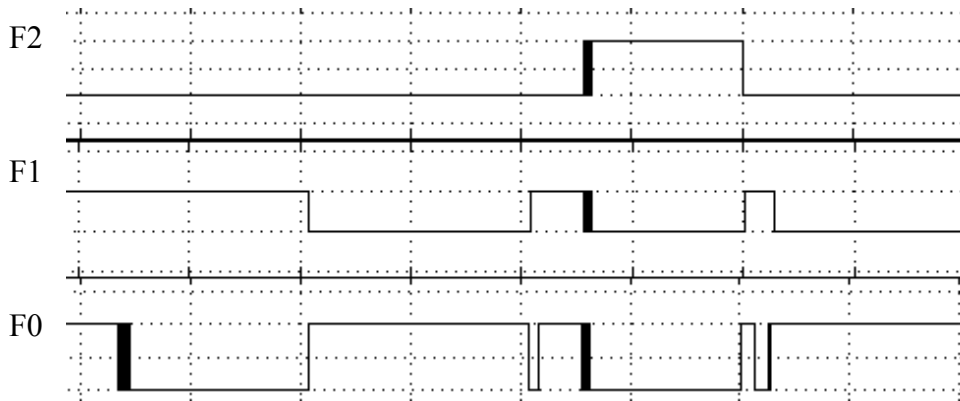


FIGURA 4.2.4: Sinais de saída do circuito arc tg (Q/I) F2, F1 e F0

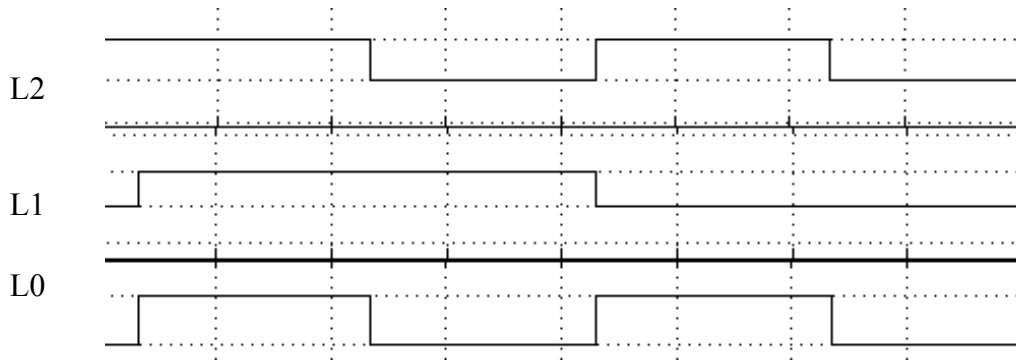


FIGURA 4.2.5: Sinais de saída do circuito latch L2, L1 e L0

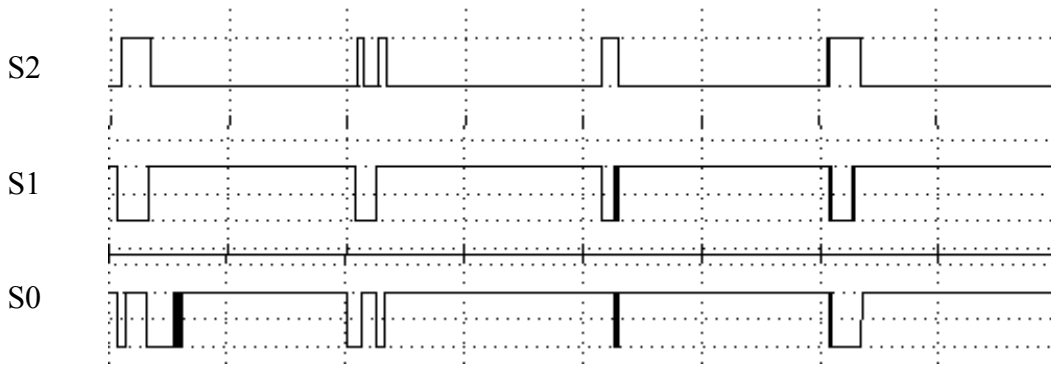


FIGURA 4.2.6: Sinais de saída do subtrator S2, S1 e S0

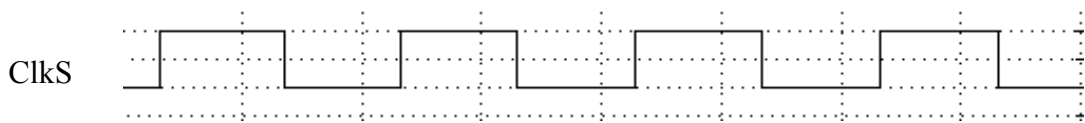


FIGURA 4.2.7: Sinal de clock de símbolo

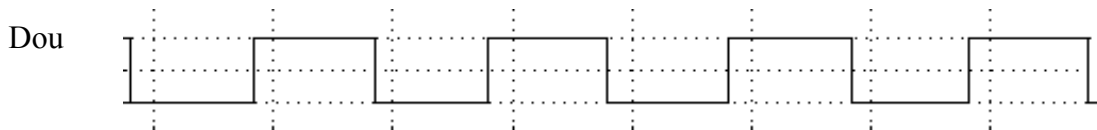


FIGURA 4.2.7: Sinal de saída do conversor paralelo série

Dos sinais mostrados anteriormente, podemos pegar como exemplo a seqüência após a seta. Os sinais I e Q passam de aproximadamente $-0.707V$, $+0.707V$ para $+1V$, $0V$, correspondendo aos sinais digitais I/Q de 001, 011 para 100, 010. Após o circuito $\arctg Q/I$ obtemos o sinal digital F2,F1, F0 passando de 101 para 000. A diferença entre as duas fases é de 011, que corresponde a $+135^\circ$ pela tabela 6. Também pode ser observado que o sinal L2,L1,L0 sempre corresponde à fase anterior do sinal F2,F1,F0. O sinal S2,S1,S0 apresenta a diferença de fase 011, porém com oscilações devido aos efeitos da digitalização dos sinais I e Q. Para evitar essas oscilações é usada a borda de descida do sinal clkB (que fica exatamente no meio da marca do sinal clkS) para obtenção do par de bits D1,D0, pois nesse tempo já ocorreu a estabilização dos bits S2,S1 e S0. Finalmente o par de bits D1,D0 é serializado de acordo com o sinal clkB, enviando o bit menos significativo primeiro. Pelas figuras anteriores, é enviado uma seqüência 01 após o circuito demodulador detectar a mudança de fase (ver seta). De acordo com a tabela 6, 01 corresponde ao avanço de fase de 135° .

Conclusão

Neste trabalho foram feitas as análises dos sinais envolvidos no processo de modulação DQPSK assim como a descrição dos blocos necessários para implementação do modulador e demodulador DQPSK. Como pode ser visto, a maior parte dos circuitos podem ser implementadas por soluções digitais (FPGA por exemplo), porém atenção especial deve ser dada à parte analógica devido ao compromisso de qualidade do sinal recebido pelo demodulador. Nesse trabalho, procurou-se apenas a simulação dos sinais em FI, sem a parte de RF (rádio frequência).

Bibliografia

- [1] Peterson, R. L. et al, "*Introduction to Spread Spectrum Communications*", Prentice Hall, 1995 .
- [2] "*Aurora 2400 Reference Manual*", Harris Corporation , 1998 .
- [3] Radiocommunication Bureau, "*Handbook Digital Radio Relay Systems*", International Telecommunication Union, 1996 .

APÊNDICE E:

ARTIGO PUBLICADO NO SEMINÁRIO BRASILEIRO DE MICROONDAS E
ÓPTOELETRÔNICA (SBMO 2002)

11 APÊNDICE E: ARTIGO PUBLICADO NO SEMINÁRIO BRASILEIRO DE MICROONDAS E ÓPTOELETRÔNICA (SBMO 2002)

Simulação MatLab e Implementação de um Sistema Spread Spectrum no Microcontrolador PIC16F876

JORGE AMORETTI LISBOA E JOSÉ JÚLIO GUIMARÃES FERNANDES

Depto. de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Av. Osvaldo Aranha 103, Porto Alegre, 90.035-190, RS

Resumo — O presente trabalho tem como objetivo a descrição e apresentação dos resultados obtidos através de simulações e implementação da técnica de modulação por espalhamento de espectro (Spread Spectrum). Maior ênfase será dada a técnica de espalhamento por seqüência direta (DSSS) dada a larga utilização nos sistemas de comunicação correntemente em uso, assim como serão apresentados os resultados de simulações e implementação de um sistema de comunicação Spread Spectrum utilizando o microcontrolador PIC 16F876 para a interligação de computadores padrão PC via interface serial RS232.

Palavras-chaves — Spread Spectrum, Simulação MatLab, microcontrolador PIC16F876.

I. INTRODUÇÃO

A substituição de cabos por sistemas sem fio é uma tendência que vem se intensificando nos últimos anos. No caso particular de conexões de curta distância (até algumas dezenas de metros) diversas tecnologias tem sido propostas e alguns “chip sets” para sua implementação estão começando a ser comercializadas (Bluetooth, Home RF, Wireless Access Protocol) (1). Entretanto estes ainda apresentam custo elevado e indisponibilidade no mercado nacional. Este trabalho tem como objetivo desenvolver um sistema “wireless” de baixo custo, utilizando componentes disponíveis no mercado nacional, para comunicações de dados com taxas típicas utilizadas no padrão RS-232. Para prover múltiplos acessos usaremos a técnica de DSSS (Direct Sequence Spread Spectrum) com códigos de espalhamento de baixa correlação.

Spread Spectrum, como conceito de técnica de modulação, já existe há várias décadas. Sendo inicialmente de uso militar, foi liberada para uso civil a partir dos anos 80 pelo governo americano. Basicamente, consiste em aumentar a largura do espectro de um sinal digital de forma que seja dificultada a detecção ou interferência deste sinal.

As aplicações dessa técnica aparecem na telefonia celular com o sistema CDMA (Code Division Multiple Access), na Internet sem fio, ou interconexão de redes locais (W-LANs), ponto-multiponto e ponto-a-ponto e em enlaces de acesso de dados e telefonia (Nx64kbps a 4x2048kbps) nas bandas de 2.4GHz e 5.8GHz (sem necessidade de licenciamento).

Existem várias formas de se obter o espalhamento espectral: o salto em frequência da portadora de acordo com o código pseudo aleatório (FHSS - Frequency Hopping Spread Spectrum), a multiplicação por uma seqüência pseudo aleatória (DSSS - Direct Sequence Spread Spectrum) do sinal modulado e até mesmo uma combinação dos dois métodos (Hybrid FH/DS SS). Para fins de análise dos sinais nesse trabalho, será considerada a técnica de espalhamento DSSS nas modulações digitais BPSK (BiPhase Shift Keying), pela simplicidade de análise, e DBPSK (Diferencial BiPhase Shift Keying), que são das mais utilizadas nos

¹ Jorge Amoretti Lisboa, jal@if.ufrgs.br, Tel +55-51-33163300, Fax+55-51-33163293, José Júlio Guimarães Fernandes, juliof@cpovo.net, Tel +55-51-33163300

modelos comerciais de rádios spread spectrum.

II. ANÁLISE DE SISTEMAS SPREAD SPECTRUM

A forma mais simples de obter-se o espalhamento espectral de um sinal modulado digitalmente é o uso de seqüência direta com modulação BPSK (2).

Para tanto é feita a multiplicação do sinal modulado em BPSK por um trem de pulsos (-1 ou +1) variando de acordo com uma seqüência pseudo aleatória (PN, do inglês Pseudo Random Noise). O resultado é um sinal com inversões de fase não apenas em função do sinal digital modulante, mas também da seqüência pseudo aleatória de espalhamento. A demodulação desse sinal será feita apenas com a correta seqüência pseudo aleatória sendo reproduzida no receptor, inclusive estimando a fase em que o sinal recebido se encontra (variável de acordo com o caminho de propagação). Para a análise matemática, o sinal modulado em BPSK a ser transmitido fica da forma:

$$S_d(t) = A \cos(\omega_0 t + \theta_d(t)) \quad (1)$$

onde S_d é o sinal modulado em fase, A e ω_0 são a amplitude e a freqüência angular da portadora e $\theta_d(t)$ é a modulação em fase de acordo com o dado a ser transmitido.

Seja $c(t)$ a seqüência pseudo aleatória chaveando entre -1 e +1, obtemos o sinal spread spectrum $S_t(t)$:

$$S_t(t) = A c(t) \cos(\omega_0 t + \theta_d(t)) \quad (2)$$

Esse é o sinal a ser transmitido por um caminho (na análise, sem distorção) que apresentará um atraso de propagação T_d . No lado receptor, aparecerá esse sinal

junto com alguma forma de interferência e ruído do tipo AWGN que, como será visto mais adiante, serão minimizados na etapa de demodulação spread spectrum (despreading).

Para que possa ser recuperado o sinal modulado em fase deve ser feita a multiplicação do sinal recebido por uma réplica da seqüência pseudo aleatória que inclua a estimativa do atraso de propagação T_d . O sinal na saída do bloco de “despreading” fica da forma:

$$S_{de}(t) = A c(t - T_d) c(t - T_{de}) \cos(\omega_0 t + \theta_d(t - T_d) + \Phi) \quad (3)$$

Onde Φ corresponde a uma fase aleatória do sinal. Se a estimativa do atraso de propagação T_d estiver correta, o produto $c(t - T_d) c(t - T_{de})$ resultará sempre igual a 1, uma vez que esses sinais assumem apenas os valores +1 e -1. Portanto $S_{de}(t)$ fica praticamente igual a $S_d(t)$ no lado do transmissor, apresentando apenas o atraso de propagação T_d e uma fase aleatória Φ . Notar que $S_d(t)$ pode apresentar qualquer modulação em fase, não apenas BPSK. O sinal digital é obtido pela demodulação desse sinal de acordo com as técnicas apropriadas.



Fig. 1: Espectro de potência para sinais BPSK e DSSS (Pot. em dBm e freq. em MHz)

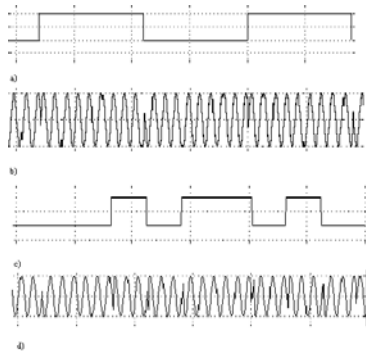


Fig. 2: Sinais BPSK a) sinal digital b) sinal modulado em BPSK c) seqüência pseudoaleatória d) sinal spread spectrum DSSS

Antes de ser feita a análise do receptor quanto a presença de ruído e/ou interferência, será visto o espectro de potência do sinal spread spectrum.

O espectro do sinal spread spectrum é semelhante ao do sinal BPSK, apenas trocando o valor de T (duração de um bit) pelo tempo T_c (duração de um chip). Chip corresponde à duração de cada estado possível na seqüência pseudo aleatória.

$$S_t(f) = \frac{1}{2} P T_c \{ (\text{sinc}^2((f - f_0) T_c) + \text{sinc}^2((f + f_0) T_c)) \} \quad (4)$$

Como T_c é muito menor que T , a amplitude de potência é reduzida pelo fator T_c/T e a largura de banda $B = 2/T$ é ampliada pelo fator T/T_c (figura 1). No lado receptor, quando houver correlação entre $c(t - T_d)$ (embutida no sinal recebido) e $c(t - T_{de})$ (estimada) haverá o efeito contrário, a amplitude de potência será aumentada por T/T_c e o espectro será estreitado pelo fator T_c/T .

Quanto ao ruído e aos sinais interferentes, por não possuírem embutido neles a seqüência pseudo aleatória a ser usada na etapa de “despreading”, o fator $c(t - T_{de})$ não será cancelado e esses sinais sofrerão a ação de espalhamento, reduzindo a potência por T_c/T e alargando o espectro por T/T_c (3) e (4). Notar que esse fator representa uma margem adicional ao receptor quanto ao ruído e

interferência. Define-se Ganho de Processamento essa relação:

$$G_p = 10 \log (T/T_c) \quad (5)$$

Outra forma de análise do efeito de espalhamento do espectro é através do teorema de Shannon:

$$C = W \log_2 [1 + S/N] \quad (6)$$

onde C é a taxa de bits máxima comportada pelo canal, W a largura espectral do sinal modulado e S/N a relação sinal ruído.

Como pode ser visto, para que possamos transmitir uma determinada taxa de bits em um meio suscetível a ruído e/ou interferência, pode ser feito o aumento da potência do sinal S ou o aumento da largura espectral W . Porém o aumento da potência não produzirá grandes efeitos na taxa de bits máxima por causa do termo \log_2 , sobrando, portanto, o alargamento do espectro do canal. Se for suficientemente alargado o espectro podemos transmitir uma determinada taxa de bits até mesmo em um meio com relação sinal/ruído menor que a unidade (potência de ruído maior que a de sinal).

III. SIMULAÇÃO MATLAB DE UM SISTEMA DSSS

Para a simulação foi implementado no módulo Simulink do software Matlab um sistema Spread Spectrum de acordo com os blocos descritos abaixo (3):

Transmissor: Gerador de Seqüência Pseudoaleatória, Misturador Spread Spectrum, Modulador em Fase, Circuitos de FI e RF;

Receptor: Amplificador de RF, Circuito de FI, Etapa de Inicialização de Sincronismo de Código e Rastreamento de Fase de Código PN, Demodulador de Fase.

O transmissor foi implementado utilizando um bloco de 4 estágios shift registers para gerar o trem de dados a transmitir da forma $x^3 + 1$. O gerador de

seqüência pseudo aleatória é constituído por 7 estágios shift registers com a forma $x^6 + 1$. Para a simulação foram normalizadas as taxas de transmissão de dados e frequência de portadora por um fator de divisão por 10^5 . A taxa de dados a transmitir é de 115,2 kbps, a taxa de chips (taxa de espalhamento) é de 1.152 kbps (para obter $G_p=10\text{dB}$) e a frequência de portadora é de 45MHz (FI). Como o objetivo da simulação é o teste da parte de espalhamento espectral, a portadora foi mantida na frequência intermediária (FI). Uma vez que a técnica de modulação empregada para a primeira etapa de modulação e para o bloco de espalhamento espectral é a mesma, DBPSK, podemos fazer o espalhamento do sinal de dados (sincronizado com o gerador de seqüência pseudo aleatória através de um flip flop com o mesmo sinal de clock desse) através da lógica XOR com a seqüência PN. A saída da XOR é aplicada ao codificador DBPSK, com saída de acordo com a tabela I abaixo.

TABELA I CODIFICAÇÃO DBPSK

Dado	Delta Fase	Dado	Fase atual	Fase seguinte
0	0°	0	0	0
1	180°	0	1	1
		1	0	1
		1	1	0

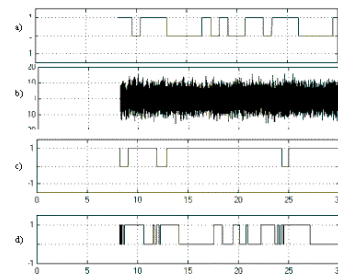
O circuitos transmissor e receptor DSSS foram implementados usando as funções do módulo Simulink do software MatLab. Foram considerados os casos em que o meio é o ar com a presença de ruído branco gaussiano (AWGN) e os efeitos de multipath, ou seja, a chegada no receptor de ondas refletidas que, por percorrerem outro percurso que não a linha reta entre transmissor e receptor, apresentam defasagem em relação ao sinal principal. Para modulações digitais, esse efeito é nocivo, podendo até anular o sinal principal se chegar uma onda refletida com

mesmo nível e defasagem de 180° . Na simulação foi feita a soma do sinal transmitido principal com uma réplica do mesmo defasada por um tempo que é um quarto da duração de um chip ($1/4 * 1,152\text{MHz} = 0.217\text{useg}$) mais o ruído AWGN com quase 10dB acima da potência do sinal DSSS.

Através das simulações foi notado o aumento dos erros de bit quando a potência de ruído é aumentada para acima de 10dB acima do sinal DSSS, comprovando o ganho de processamento de 10dB. Ver figura 3.

Quanto à interferência, foram simulados um transmissor interferente com código PN da forma x^6+x^2 e com código PN da forma $x^6+x^5+x^4+x^1$. Nas duas simulações, a potência do interferente estava no limite de 2dB abaixo do sinal DSSS principal. Como foi notado, com o uso de um código com menor correlação ($x^6+x^5+x^4+x^1$) com o usado no transmissor principal ($x^6 + 1$), obteve-se menor taxa de erros nos bits recebidos.

Fig. 3: Sinais DSSS/DBPSK a) sinal digital b) sinal



DSSS/DBPSK recebido c/ ruído c) Trigger d) sinal digital correlacionado

A escolha dos códigos é uma questão importante em sistemas Spread Spectrum, uma vez que a possibilidade de múltiplos transmissores ocuparem a mesma banda depende do quanto os códigos PN de cada um sejam menos correlacionados entre si, isto é, quase ortogonais. Outro aspecto importante é a baixa autocorrelação do código escolhido para melhorar a imunidade aos efeitos de multipath. As ondas que chegam ao receptor estarão com

fases diferentes, e portanto, com fases distintas do código PN embutido. Com o uso de um código de baixa autocorrelação, o circuito de correlação do receptor irá diminuir o nível desses sinais ao mesmo tempo que aumentando o nível do sinal principal (que chega pelo caminho da linha reta entre transmissor e receptor).

IV. IMPLEMENTAÇÃO NO MICROCONTROLADOR PIC16F876 DE UM SISTEMA SPREAD SPECTRUM

O microcontrolador PIC16F876 da Microchip Technology Inc. foi o componente usado para a implementação física, devido a disponibilidade e baixo custo desse componente (7).

O software é feito em linguagem C para o compilador CCS (Custom Computer Services) rodando sobre o ambiente de programação MPLAB for Windows/16 5.00.00 da Microchip Technology, Inc

O programa DSSS.C é dividido em três rotinas: interrupção para transmissão (RTCC_interrupt), interrupção para recepção (RB_interrupt) e principal (main). Para dar maior velocidade, a lógica XOR do sinal a transmitir com a sequência PN é feita byte a byte, sendo armazenada em um buffer definido como variável global para que possa ser usado por todas as rotinas do programa. A idéia é fazer a decomposição do byte/caracter (da entrada Rx do PIC, vindo do PC) a transmitir de forma espalhada, bit a bit e, de acordo com o valor desse, enviar um byte da tabela de seqüências PN invertido ou não (um ou zero de acordo com a lógica XOR).

No lado da recepção o sinal recebido gera uma interrupção externa ao microcontrolador que trata de montar um byte a cada oito pulsadas na entrada correspondente (clock recuperado a partir do sinal recebido em banda básica). Esse byte é comparado via lógica XOR com a tabela em memória com o ponteiro já ajustado para a diferença de atraso entre transmissor e receptor. Esse ajuste é feito

na rotina de inicialização do programa em que o PIC transmite seqüencialmente o valor 0x00 (que pela lógica XOR como os códigos PN em tabela, transmite os próprios códigos bit a bit). Pela lógica XOR, se a maioria dos bits for igual a zero, é considerado bit 0, se a maioria é igual a 1, é considerado de valor 1 o bit correspondente do byte/caracter sendo montado, que posteriormente é enviado ao PC via Tx. Lógico que o processo de montagem da seqüência de bytes/caracteres embutidos nas seqüências DSSS recebidas deve ser provida por rotinas (protocolos) em camada de nível mais alto.

Para fins de teste do programa, por enquanto, a tabela de seqüências PN é feita por uma seqüência de 8 bytes (64 estados possíveis para cada chip), portanto gerada por uma cadeia de shift registers de 6 estágios.

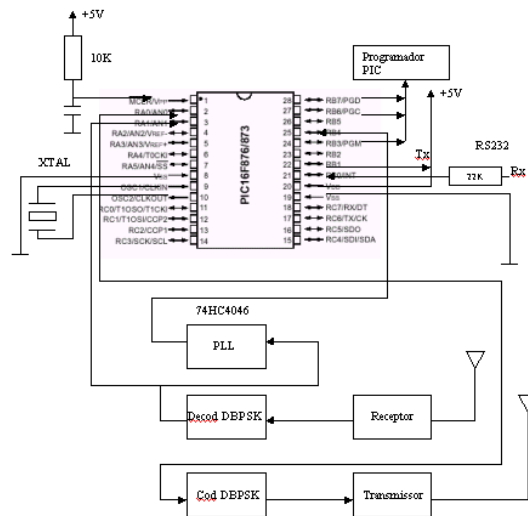


Fig. 4: Hardware do sistema PIC16F876 / DSSS

Outra observação é quanto ao interfaceamento com a porta serial do micro, pois no diagrama de blocos do software é apresentada a rotina que faz a conversão paralela/serial no transmissor e serial/paralela no receptor. Na verdade é realizada novamente a conversão serial do caractere enviado/recebido, mas no software em linguagem C, esses sinais são tratados byte a byte pelas funções `printf()` e `getch()` que, para o compilador, significam a escrita e leitura da porta serial do PIC.

Segue abaixo a descrição das principais rotinas do programa `dsss.c` compilado para o sistema PIC16F876:

- 1) **Rotina `rtcc_interrupt()`**: esta é a rotina executada toda vez que é ativada a interrupção interna gerada pelo `timer0`. Essa rotina tem como principal função analisar bit a bit os bytes contidos no buffer de transmissão `tx_buffer[]` e colocá-los na porta RA0 (TX-DSSS). Após 64 interrupções (tabela `tx_buffer[]` com 8 bytes) é ativada a variável `fim_tx` que indicará à rotina principal que naquele instante houve a transmissão de todo o buffer de transmissão (que corresponde a

um caractere/byte espalhado pela sequência PN.

Também é gerada uma referência de clock na saída RB2 para testes de loopback.

- 2) **Rotina `rb_interrupt()`**: ativada por mudanças

de estado na entrada RB4 (clock recuperado), quando então é feita a leitura da entrada RA1 (`rx_dsss`). A partir dos bits recebidos, essa rotina monta os bytes que entrarão na tabela `rx_buffer[]`. Após 64 interrupções é ativada a variável `rx_buffer[]` e já pode ser feita a leitura e desespalhamento da informação recebida (no caso um byte/caractere).

- 3) **Rotina `main()`**: a rotina principal é dividida em quatro estados:

- **Estado 0**: nesse estado inicial é feita a requisição do caractere a ser transmitido para ser salvo na variável `s`. O buffer de transmissão `tx_buffer[]` é preenchido com a lógica XOR entre o conteúdo de cada bit

da variável Rx (valor 0x00) e cada um dos 8 bytes da seqüência PN tabelada.

- **Estado 1:** nesse estado são liberadas as interrupções int_rtcc e int_rb . Então entra em um loop aguardando a confirmação de nova tabela rx_buffer[] recebida pela variável novo. Se não está ainda em processo de sincronização inicial é carregada essa tabela em buf_mem[] para posterior análise e

liberar a tabela rx_buffer[] para a rotina rb_interrupt() ainda habilitada. É feita a lógica XOR entre o conteúdo de buf_mem[] e pn[] byte a byte. Se a maioria dos bits estiver correlacionada existe o alinhamento entre o sinal recebido e a

seqüência PN tabelada e é feita a saída desse loop. Se não houver alinhamento, é feito o deslocamento bit a bit do conteúdo de buf_mem[] até se obter esse alinhamento. O número de vezes necessário para deslocamento bit a bit é armazenado na variável pointer que indicará o início da seqüência PN embutida nas tabelas buf_mem[] seguintes. Notar que essa rotina implementa a técnica de busca serial de sincronização inicial do código de espalhamento.

- **Estado 2:** nesse estado é montado o buffer de transmissão com a lógica XOR entre a variável s (caracter a transmitir) e a seqüência PN tabelada bit a bit.

- **Estado3:** esse estado aguarda a confirmação de novo caracter recebido para proceder o desespalhamento da informação contida em buf_mem. O ponto exato para início da análise é indicado pela variável pointer obtida no estado de sincronização inicial. O caracter é enviado via saída Tx do PIC para impressão no Hyperterminal (comunicação serial RS232). A rotina volta para o estado 0 para aguardar novo caracter a transmitir.

Pode-se agora fazer uma descrição do processo de comunicação entre dois sistemas A e B para a transmissão de um caracter. Inicialmente o sistema A requer o caracter a ser transmitido ao usuário (estado 0). Enquanto isso o sistema B

também está no estado 0, aguardando entrada de caracter. Assim que é dada a entrada do caracter no sistema A, este passa para o estado 1 no qual são liberadas as interrupções Tx e Rx , fazendo com que sejam transmitidas as seqüências PN para inicialização da ponta B. O sistema A fica aguardando recepção de B para que seja feita a sincronização inicial da seqüência PN de B. Porém, como B está no estado 0, não há transmissão por parte deste. Para tanto deve ser dada a entrada de um caracter em B (pressionando uma tecla no Hyperterminal) fazendo com que passe para o estado 1. Como A já está transmitindo sua seqüência PN, o sistema B entra na rotina de inicialização obtendo o sincronismo. Então B passa ao estado 2 assim como A, pois no estado 1, B também transmitiu sua sequencia PN de inicialização. No estado 2, A monta o buffer de transmissão do caracter e envia para B assim como B envia para A . O sistema B acusa novo caracter recebido e imprime na tela do Hyperterminal o caracter enviado por A. Os dois sistemas voltam ao estado 0, aguardando caracter a ser transmitido nas duas pontas.


```

pic - Hyperterminal
Início do programa
Caracter a transmitir:
rx: A: 65
rx_buffer[ 0]: -128
rx_buffer[ 1]: -22
rx_buffer[ 2]: 53
rx_buffer[ 3]: 18
rx_buffer[ 4]: 68
rx_buffer[ 5]: -122
rx_buffer[ 6]: 32
rx_buffer[ 7]: 85
tx_buffer[ 0]: -22
tx_buffer[ 1]: 53
tx_buffer[ 2]: 18
tx_buffer[ 3]: 68
tx_buffer[ 4]: -122
tx_buffer[ 5]: 32
tx_buffer[ 6]: 85
tx_buffer[ 7]: -128
rx: rx: A: 65
pointer: 56
Caracter a transmitir:

```

Fig. 5: Tela do SW Hyperterminal com as respostas dos testes com o PIC16F876 / DSSS

V. OBSERVAÇÕES FINAIS

Neste trabalho foram feitas as análises matemáticas dos sinais envolvidos no processo de espalhamento espectral em seqüência direta assim como a descrição dos blocos necessários para implementação do transmissor e receptor DSSS. A operação do sistema foi testada através de simulação em MatLab. Posteriormente foi feita a implementação do mesmo em banda base, utilizando um microcontrolador PIC16f876, o qual

realiza o processo de espalhamento espectral na transmissão e recuperação do sinal banda base na recepção. No momento estamos realizando a implementação física da geração do sinal de FI, na frequência de 45 MHz com modulação BPSK. A escolha desta frequência permite o uso de filtros e outros componentes utilizados em canais de FI de receptores de TV, o quais são disponíveis e de baixo custo. A etapa seguinte no desenvolvimento do projeto será a conversão do sinal de FI para 433 MHz, que será a frequência de trabalho do sistema.

AGRADECIMENTOS

Os autores agradecem aos profs. Álvaro A de Salles e Claudio Fernández e ao Eng Paulo Serafini pelas contribuições a esse trabalho.

REFERÊNCIAS

- [1] Alderton, M "The Bluetooth question" RF Design, January, 2001.

ANEXO A:
NORMA ANATEL PARA O USO DE BANDAS POR EQUIPAMENTOS BASEADOS EM
TECNOLOGIA DE ESPALHAMENTO ESPECTRAL ANEXO A:

12 ANEXO A: NORMA ANATEL PARA O USO DE BANDAS POR EQUIPAMENTOS BASEADOS EM TECNOLOGIA DE ESPALHAMENTO ESPECTRAL

NORMA Nº 12/96

CONDIÇÕES DE USO DE FREQUÊNCIAS NAS FAIXAS DE 902 - 928 MHz, 2400 - 2483,5 MHz E 5725-5850 MHz, POR EQUIPAMENTOS DE RADIOCOMUNICAÇÃO EMPREGANDO TÉCNICA DE ESPALHAMENTO ESPECTRAL

1-OBJETIVO

Estabelecer as condições de uso de frequências nas faixas de 902-928 MHz, 2400-2483,5 MHz e 5725-5850 MHz por equipamentos de radiocomunicação empregando técnica de espalhamento espectral, para aplicações fixas (ponto a ponto e ponto-multiponto) e móveis.

2-DEFINIÇÕES

2.1-Espalhamento espectral - sistemas nos quais a energia média do sinal transmitido é espalhada sobre uma largura de faixa que é muito mais larga que a largura de faixa que contém a informação. Esses sistemas compensam uma maior largura de faixa de transmissão, por uma menor densidade espectral de potência e uma melhora na rejeição aos sinais interferentes operando na mesma faixa de frequências.

2.2-Técnicas - o espalhamento espectral pode ser efetuado através de duas técnicas:

Seqüência direta - os sistemas em seqüência direta combinam a informação do sinal, que normalmente é digital, com uma seqüência binária de maior velocidade. Esta combinação é então usada para modular a portadora de radiofrequência. O código binário domina a função de modulação e é a causa direta do espalhamento largo do sinal transmitido. Este código é uma seqüência de bits pseudoaleatória de comprimento fixo. O sistema continuamente recicla o mesmo código binário.

Saltos em frequência - os sistemas com saltos em frequência espalham sua energia mudando a frequência central de transmissão várias vezes por segundo de acordo com uma seqüência de canais gerada de forma pseudoaleatória. Essa mesma seqüência de canais é usada repetidamente, de forma que o transmissor recicla continuamente a mesma série de mudança de canais.

2.3-Seqüência Pseudoaleatória - uma seqüência de dados binários que tem na sua formação algumas características de seqüência aleatória mas também tem algumas características que não são aleatórias.

2.4-Ganho de processamento - é a melhoria da relação sinal/ruído que um sistema que utiliza a técnica de espalhamento espectral é capaz de obter em relação a um sistema que não utiliza esta mesma técnica. Para sistemas que empregam espalhamento espectral em seqüência direta esse ganho está diretamente relacionado à taxa na qual o código de espalhamento é gerado. Para sistemas por saltos em frequência, o ganho

de processamento é uma função direta do número de canais de salto nos quais está sendo espalhada a informação transmitida.

3-FREQÜÊNCIAS

Os equipamentos de radiocomunicação utilizando técnica de espalhamento espectral estão autorizados a operar nas faixas de freqüências de 902-928 MHz, 2400-2483,5 MHz e 5725-5850 MHz.

4-OUTRAS CARACTERÍSTICAS

4.1-Potência

4.1.1-A potência de pico máxima de saída do transmissor não pode ser superior a 1 W, limitando-se ainda a potência EIRP máxima a 6 dBW.

4.1.2-Para sistemas em seqüência direta a densidade média de potência em qualquer faixa de 3 kHz, medida em um intervalo de 1 s, não deve ser maior que 8 dBm.

4.2-Limites de emissões não desejadas

As emissões não desejadas devem atender a um dos limites a seguir:

4.2.1-A potência de radiofreqüência produzida em qualquer intervalo de freqüência de 100 kHz fora das faixas autorizadas no item 3 deve estar, no mínimo, de 20 dB abaixo da potência máxima produzida num intervalo de 100 kHz dentro das faixas autorizadas; ou

4.2.2-Não podem exceder aos limites de intensidade de campo especificados na tabela abaixo:

FAIXA DE FREQÜÊNCIA (MHz)	INTENSIDADE DE CAMPO (microvolts/m)	DISTÂNCIA DE MEDIDA (m)
0,009 - 0,490	2400/f (kHz)	300
0,490 - 1,705	24000/f (kHz)	30
1,705 - 30,0	30	30
30 - 88	100	3
88 - 216	150	3
216 - 960	200	3
Acima de 960	500	3

4.3-Ganho de processamento

4.3.1-O ganho de processamento de um sistema em seqüência direta deve ser pelo menos de 10 dB, e deve ser determinado a partir da relação sinal/ruído em dB com o código do espalhamento desligado e essa mesma relação com o código de espalhamento ligado, medido na saída do demodulador do receptor.

4.3.2-Sistema híbridos que utilizam uma combinação das técnicas de modulação em seqüência direta e saltos em freqüência, devem alcançar um ganho de processamento de, no mínimo, 17 dB, na combinação dessas técnicas. A operação com saltos em freqüência do sistema híbrido, com a operação em seqüência direta desligada, deve ter um tempo médio de ocupação em qualquer freqüência que não exceda 0,4 s, em um período de tempo, em segundos, igual ao número de freqüências de salto utilizadas, multiplicado por 0,4. A operação em seqüência direta do sistema híbrido com a operação por saltos em freqüência desligada, deve obedecer ainda aos requisitos de densidade de potência do item 4.1.2.

4.4-Outros requisitos dos sistemas

4.4.1-Sistemas de salto em frequência

As frequências portadoras dos canais de salto devem estar separadas por um mínimo de 25 kHz ou pela largura de faixa do canal de salto, a 20 dB, devendo ser considerado o maior valor. O sistema deve saltar para as frequências selecionadas na taxa de salto a partir de uma lista de frequências de salto ordenadas de forma pseudoaleatoria. Cada frequência deve ser, em média, usada igualmente por cada transmissor. Os receptores do sistema devem ter largura de faixa de entrada compatível com a largura de faixa do canal de salto dos respectivos transmissores e devem mudar as frequências em sincronia com os mesmos.

Os sistemas operando na faixa de 902-928 MHz devem usar no mínimo 50 frequências de salto, devem limitar a máxima largura faixa ocupada, a 20 dB, em 500 kHz e o tempo médio de ocupação de qualquer canal não deve ser superior a 0,4s num intervalo de 20s.

Os sistemas operando nas faixas 2400-2483,5 MHz e 5725-5850 MHz devem usar no mínimo 75 frequências de salto, devem limitar a máxima largura de faixa ocupada, a 20 dB, em 1 MHz e o tempo médio de ocupação de qualquer canal não deve ser superior a 0,4s num intervalo de 30s.

4.4.2-Sistemas de seqüência direta

Para estes sistemas a largura de faixa a 6 dB deve ser, no mínimo, 500 kHz.

5-CONDIÇÕES DE USO

5.1-As estações são dispensadas de licenciamento, porém os equipamentos devem ser certificados pelo Ministério das Comunicações, de acordo com as normas vigentes.

5.2-O Ministério das Comunicações poderá determinar alteração das características técnicas e das condições de uso aqui estabelecidas, mesmo dos sistemas em operação, com a finalidade de otimizar o uso do espectro de radiofrequências.

5.3-Os sistemas que operam em conformidade com esta norma o fazem em caráter secundário isto é, não tem direito a proteção contra interferência prejudicial, mesmo de estações do mesmo tipo, e não podem causar interferência a sistemas operando em caráter primário.