



Trabalho de Conclusão de Curso

**DESEMPENHO E POTÊNCIA DE DECODIFICADORES
LDPC APLICÁVEIS A SISTEMAS ÓPTICOS**

Tiago Carneiro de Lima

Brasília, Novembro de 2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

Trabalho de Conclusão de Curso

**DESEMPENHO E POTÊNCIA DE DECODIFICADORES
LDPC APLICÁVEIS A SISTEMAS ÓPTICOS**

Tiago Carneiro de Lima

*Relatório submetido
como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Dr. André Noll Barreto (ENE/UnB) _____
Orientador

Prof. Dr. Darli Augusto de Arruda Mello _____
(ENE/UnB)
Examinador Interno

Prof. Dr. Anderson Clayton Alves Nascimento _____
(ENE/UnB)
Examinador Interno

FICHA CATALOGRÁFICA

LIMA, TIAGO CARNEIRO DE
DESEMPENHO E POTÊNCIA DE DECODIFICADORES LDPC APLICÁVEIS A SISTEMAS
ÓPTICOS

[Distrito Federal] 2012.

xi, 55p., 210 x 297 mm (FT/UnB, Engenheiro, Eletricista, 2012).

Trabalho de Conclusão de Curso - Universidade de Brasília, Faculdade de Tecnologia.

- | | |
|---------------|-------------------------|
| 1. LDPC | 2. Comunicações Ópticas |
| 3. FEC | 4. FPGA |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

LIMA, T. C. (2012). DESEMPENHO E POTÊNCIA DE DECODIFICADORES LDPC APLICÁVEIS A SISTEMAS ÓPTICOS, Trabalho de Conclusão de Curso em Engenharia Elétrica, Publicação FT.TG-nº -, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 55p.

CESSÃO DE DIREITOS

AUTOR: Tiago Carneiro de Lima

TÍTULO: DESEMPENHO E POTÊNCIA DE DECODIFICADORES LDPC APLICÁVEIS A SISTEMAS ÓPTICOS.

GRAU: Engenheiro Eletricista ANO: 2012

É concedida à Universidade de Brasília permissão para reproduzir cópias deste trabalho de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse trabalho de conclusão de curso pode ser reproduzida sem autorização por escrito do autor.

Tiago Carneiro de Lima

Quadra 5, Conjunto A, Casa 8

CEP 73030-050, Sobradinho - DF - Brasil

Dedicatória

À meus pais Cláudio e Cinara, meus mestres da Engenharia da Vida.

À madrinha Helena, tio Gildinho e família.

À meus padrinhos Baltazar e Francisca.

Tiago Carneiro de Lima

Agradecimentos

Agradeço a Deus pela minha vida e por Ele me conceder a graça de eu chegar ao fim desse curso.

Muito obrigado a meus pais pelos valores e ensinamentos a mim transmitidos com muito amor, e por serem os grandes responsáveis pela minha formação acadêmica. A meus familiares, Jorge, Vera, Zuleide, Rebeca, Talitha e Dora, agradeço o carinho e paciência que tiveram comigo, especialmente durante a graduação. Obrigado a minha querida Larissa por ter me apoiado durante todo o curso e por ter compreendido minhas ausências em semanas de provas.

Agradeço ao meu orientador, Prof. André, pela oportunidade de realizar este trabalho e por compartilhar de seu conhecimento e de seu tempo com tanta dedicação. Ao Prof. Darli, pela oportunidade de entrada no projeto e pelo empenho de suas orientações. Aos colegas do OCNLab, Celio, Diego, Fernando, Thiago, Valery, Victor e Vinicius, obrigado pelos diversos momentos de “brainstorm” e de ajuda que me proporcionaram. Agradeço também ao João Paulo do MWSL por disponibilizar a versão inicial do algoritmo de decodificação e pela prestação em me receber no LEMOM diversas vezes.

Muito obrigado aos amigos Dayana Itai, Elias Rachid, Heyder Araújo, Jonaylton Moura, José Oniram, Júlio Seype e Luan Calaça pelas incontáveis horas de estudo e por terem tornado nosso convívio fora e dentro da faculdade tão prazeroso. Estendo meus agradecimentos aos amigos de engenharia: Antônio, Fernandes, Gabi, João Victor, Miguel, Renan, Sabino, Tiozão e Tonton.

Agradeço ao Prof. Celiuss pelas experiências de monitoria, pelos conselhos de amigo e pelas aulas de Latex que foram essenciais para a elaboração desta monografia.

Agradeço a experiência e o aprendizado recebidos de meus professores da Universidade de Coimbra, em especial do Prof. Marco Cravo e do Prof. Mário Silveirinha. Aproveito para registrar minha gratidão aos amigos de intercâmbio, Gabriel Ferraz, Jana Žáčková, João Marcos, Leonardo Pieroni e Tarcísio Neves.

Não poderia esquecer dos Professores da UnB, em especial os do Departamento de Engenharia Elétrica (ENE), que são responsáveis pela transmissão do conhecimento e por minha evolução de calouro para Engenheiro. Não menos importantes são os funcionários de todos os setores da Faculdade de Tecnologia: limpeza, segurança, administração e assistência técnica. Mando o meu agradecimento especial ao pessoal da secretaria: Carol, Evangelista, Lúcio e Vera. Obrigado pelo trabalho de todos vocês.

Agradeço o apoio e afeto da madrinha Helena, tio Gildinho, Moema, Norma e família. Nos momentos de maiores dificuldades desta reta final, a presença em espírito da tia Helena foi determinante para me dar coragem e perseverança para a conclusão deste trabalho.

Também agradeço o apoio incondicional de meus padrinhos Baltazar e Francisca, que desde o berço estão presentes em todas as etapas de minha caminhada.

Minha gratidão a todos os meus parentes, dentre eles os tios Cenildo, Cerize, César, Ciro, Cirene, Irene, Noeme, Reinaldo e vovó Judite. Agradecimento especial aos tios Givaldo e Cida pelo acolhimento e apoio constantes que me deram durante o curso.

Gostaria de reconhecer o apoio de meus queridos amigos Giselle, Guilherme Oliveira, Guilherme Ornellas e Nayara, João Paulo, Luciana, Márcio, Mário Dornellas, Núbia, Philipe, Rafael e Rodrigo Lima.

Muito obrigado aos irmãos da Imaculada Conceição pelo companheirismo e pelas orações direcionadas à concretização deste sonho, em especial agradeço aos casais Chaguinha e Deusimar, Ederlan e Ivone, João e Regina, Laurecy e Japi, Manuel e Fernanda, e Mário Jorge e Vera.

Enfim, a todos que contribuíram direta e indiretamente com a chegada do dia de hoje, muito obrigado!

Tiago Carneiro de Lima

RESUMO

O crescente volume de tráfego de dados da Internet está intimamente ligado com a evolução das comunicações ópticas. Os sistemas ópticos atuais já possuem enlaces que cobrem longas distâncias com taxas de transmissão da ordem das centenas de Gb/s. Para se construir sistemas eficientes com baixo custo, é necessária a utilização de técnicas de modulação e de processamento digital de sinal que aumentem a qualidade das transmissões. Uma dessas técnicas é a codificação de canal, que pode utilizar códigos corretores de erro (FEC - *forward error correction*) para reduzir a ocorrência de bits errados no receptor. Um dos códigos corretores de erro mais promissores para sistemas ópticos é o LDPC (*Low-Density Parity-Check*).

Por um lado, a utilização de FEC reduz a taxa de erro de bit (*bit error rate* - BER) . Por outro, ela aumenta a complexidade e o consumo global de potência do sistema. Assim, deve-se relacionar o desempenho de BER com as potências de codificação e de decodificação adicionadas pelo FEC.

É com base nesse contexto que este trabalho tem o foco na análise de desempenho de BER e de potência de decodificadores LDPC aplicáveis a sistemas ópticos. Para isso, foram realizadas simulações de taxa de erro de bit no Matlab, além de estimações de potência de decodificadores para tecnologia CMOS 28 nm.

Palavras-Chave: LDPC, Comunicações Ópticas, FEC, FPGA.

ABSTRACT

The increasing volume of Internet data traffic is deeply related with the optical communications evolution. Current optical systems already have links which cover long-haul distances with data rates of hundreds of Gb/s. In order to set up efficient systems with optimal cost, is required the utilization of modulation and digital signal processing (DSP) techniques that increase transmissions quality. One of these techniques is the channel coding, which can uses forward error correction (FEC) to decrease the occurrence of error bits at the receptor. One of the most promising FEC codes for optical systems is the Low-Density Parity-Check (LDPC) code.

On the one hand the FEC utilization decreases the error bit rate (BER). On the other hand it increases the complexity and the overall power consumption of the system. Therefore, the BER performance should be related with the coding and decoding powers added by FEC.

It's based on this context that this work has the focus on the analysis of BER performance and decoding power of LDPC decoders applicables for optical systems. For this, error bit rate simulations were performed in Matlab besides decoding power estimates for CMOS 28 nm technology.

Keywords: LDPC, Optical Communications, FEC, FPGA.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTO E MOTIVAÇÃO	1
1.1.1	BREVE HISTÓRICO - FEC EM COMUNICAÇÕES ÓPTICAS	2
1.2	OBJETIVO DO TRABALHO	4
2	SISTEMA ÓPTICO DE COMUNICAÇÃO	5
2.1	VISÃO GERAL	5
2.2	MODELO DE CANAL ÓPTICO	7
2.3	MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO	7
3	CÓDIGOS DE BLOCO LINEARES	11
3.1	DEFINIÇÕES BÁSICAS	11
3.1.1	TAXA DE CÓDIGO E REDUNDÂNCIA	11
3.1.2	DISTÂNCIA E PESO DE HAMMING	12
3.1.3	DISTÂNCIA MÍNIMA	12
3.1.4	DISTÂNCIA EUCLIDIANA	12
3.2	REPRESENTAÇÃO MATRICIAL	13
3.2.1	CÓDIGOS SISTEMÁTICOS	15
3.2.2	MATRIZ DE TESTE DE PARIDADE E EQUAÇÕES DE PARIDADE	16
3.3	DISTRIBUIÇÃO DE PESOS	18
3.4	ESFERAS DE HAMMING, CAPACIDADES DE DETECÇÃO E DE CORREÇÃO DE CÓDIGOS DE BLOCO LINEARES	19
3.5	DECISÃO <i>HARD</i> E <i>SOFT</i>	20
3.6	PROBABILIDADE DE ERRO - DECISÃO <i>SOFT</i>	21
4	CÓDIGOS LDPC	23
4.1	INTRODUÇÃO AO LDPC	23
4.2	MATRIZ ESPARSA DE TESTE DE PARIDADE	24
4.3	CODIFICAÇÃO	24
4.4	GRAFO DE TANNER E <i>GIRTH</i>	25
4.5	DECODIFICAÇÃO: ALGORITMO SOMA-PRODUTO	26
4.5.1	FUNÇÕES DE VEROSSIMILHANÇA	26
4.5.2	ALGORITMO DE DECODIFICAÇÃO: SOMA-PRODUTO ORIGINAL	27

4.5.3	ALGORITMO DE DECODIFICAÇÃO: SOMA-PRODUTO SIMPLIFICADO.....	30
4.6	PISO DE ERRO.....	31
5	CONSUMO DE POTÊNCIA EM FPGAS.....	33
5.1	INTRODUÇÃO FPGA.....	33
5.2	ETAPAS DE PROJETO.....	34
5.3	MODELO DE CONSUMO DE FPGAS.....	34
6	DESENVOLVIMENTO E RESULTADOS.....	37
6.1	DESCRIÇÃO GERAL.....	37
6.2	ESTIMAÇÃO DA BER.....	37
6.2.1	RESULTADOS - DESEMPENHO DE BER.....	39
6.3	ESTIMAÇÃO DA POTÊNCIA DE DECODIFICAÇÃO PARA CÓDIGOS COM TAMANHO DE BLOCO PEQUENO.....	40
6.3.1	ENTRADAS DO DECODIFICADOR.....	40
6.3.2	FLUXO DE ETAPAS.....	42
6.4	ESTIMAÇÃO DA POTÊNCIA DE DECODIFICAÇÃO PARA CÓDIGOS COM TAMANHO DE BLOCO GRANDE.....	44
7	CONCLUSÃO.....	48
7.1	TRABALHOS FUTUROS.....	48
	REFERÊNCIAS BIBLIOGRÁFICAS.....	50
	Anexos.....	54
I	DECISÃO MAP E ML.....	55

LISTA DE FIGURAS

1.1	Esquema de concatenação de códigos (Super FEC) da 2ª geração. Adaptado de [1].	3
2.1	Diagrama de blocos geral de um sistema óptico ponto-a-ponto.....	6
2.2	Ganho de código entre codificação LDPC(64800, 58320) e sistema BPSK sem codificação.	10
3.1	Espaço de sinais bidimensional. Distância Euclidiana d entre os sinais $s_1(t)$ - coordenadas (x_1, y_1) - e $s_2(t)$ - coordenadas (x_2, y_2)	13
3.2	Formato sistemático da palavra código.	15
3.3	Ilustração geométrica das esferas de Hamming. Adaptado de [2]	20
4.1	Matriz de teste de paridade no formato triangular inferior aproximado. Adaptado de [3].	25
4.2	Grafo de Tanner correspondente à matriz \mathbf{H} da Equação 4.2.	26
4.3	Grafo de Tanner de um código de Hamming(7,4). <i>Girth</i> igual a 4 em destaque.	26
4.4	Grafo de Tanner no qual os nós de verificação nh_1 e nh_2 são nós filhotes de nc_1	28
4.5	Grafo de Tanner no qual os nós de variável nc_2 , nc_4 e nc_5 são nós progenitores de nh_4	28
4.6	Troca de informações na decodificação LDPC.	29
4.7	Cálculo dos coeficientes R_{11}^0 e R_{11}^1	29
4.8	Cálculo dos coeficientes Q_{36}^0 e Q_{36}^1	29
4.9	Comportamento típico de códigos LDPC com piso de erro. Adaptado de [2].	32
5.1	Correntes de fuga num FET. Adaptado de [4].	35
6.1	Sistema de comunicação simulado.	38
6.2	Desempenho de códigos LDPC de altas taxas para 50 iterações.	39
6.3	Desempenho de taxa de erro de bit para diferentes valores de iterações. Código LDPC(64800,51840).	40
6.4	Relação entre BER e número de iterações do decodificador para o código LDPC(64800, 51840).	41
6.5	Fluxo de tarefas para a criação de descrição do algoritmo de decodificação em linguagem de <i>hardware</i> (VHDL).	43
6.6	Relatório de consumo de potência de decodificação (1 iteração), LDPC(10,5). Vazão de dados de 28 Gbps, frequência de relógio de 815 MHz.	43

6.7	Potência de decodificação pelo número de iterações para o código LDPC(10,5). Decodificador CMOS 28 nm, $T = 28$ Gbps e $f_c = 815$ MHz.	44
6.8	Potência de decodificação para n igual a 10, 20 e 30 bits, 1 iteração. Decodificador CMOS 28 nm, $T = 28$ Gbps e $f_c = 815$ MHz.	45
6.9	Relação entre BER e estimativa de potência de decodificação para código LDPC(64800, 51840). Decodificador CMOS 28 nm, $f_c = 815$ MHz e $T = 28$ Gbps.	46

LISTA DE TABELAS

3.1	Lista das 2^4 palavras codificadas do código de Hamming(7,4).....	18
3.2	Distribuição de pesos do código de Hamming(7,4).....	18
6.1	Comparação de desempenho de códigos LDPC	39

LISTA DE SÍMBOLOS

SÍMBOLOS

\oplus	Soma módulo-2
$ \cdot $	Módulo de (\cdot)
$\lfloor \cdot \rfloor$	Maior inteiro menor ou igual a (\cdot)
$\langle \cdot, \cdot \rangle$	Produto interno entre vetores
$(\cdot)^T$	Transposto de matriz ou vetor
$\tilde{(\cdot)}$	Valor estimado de (\cdot)
$\arg \max(\cdot)$	Argumento máximo de (\cdot)
$d_E(\cdot, \cdot)$	Distância Euclidiana entre duas palavras ou dois sinais
$d_H(\cdot, \cdot)$	Distância de Hamming entre duas palavras
$\operatorname{erfc}(\cdot)$	Erro complementar de (\cdot)
$\min(\cdot)$	Valor Mínimo de (\cdot)
$p(\cdot)$	Função densidade de probabilidade de (\cdot)
$p(a b)$	Função densidade de probabilidade condicional de a dado b
$P(\cdot)$	Probabilidade de (\cdot)
$P(a b)$	Probabilidade condicional de a dado b
$w_H(\cdot)$	Peso de Hamming de (\cdot)
α_{ij}	Constante de normalização
δQ_{ij}	Diferença entre os coeficientes Q_{ij}^x
δR_{ij}	Diferença entre os coeficientes R_{ij}^x
$\mu(\cdot)$	Valor médio de (\cdot)
ϕ_1 e ϕ_2	Funções ortonormais do espaço bidimensional
σ	Desvio padrão
$\sigma(\cdot)$	Desvio padrão do bit (\cdot)
σ^2	Variância
\mathbf{c}	Vetor palavra código
\mathbf{c}_k	k -ésimo vetor palavra código
c_i	i -ésimo bit da palavra código \mathbf{c}
$c_{k,i}$	i -ésimo bit da k -ésima palavra código \mathbf{c}_k
d_{\min}	Distância mínima do código
\mathbf{e}	Vetor de erros inseridos pelo canal binário simétrico

f	Frequência
f_c	Frequência do relógio
f_j^x	Probabilidade <i>a priori</i> do bit recebido
g	Ciclo mínimo (<i>girth</i>)
gap	Parâmetro da matriz de teste de paridade do algoritmo de Richardson
\mathbf{g}_i	i -ésimo vetor linha da matriz geradora \mathbf{G}
$g_{i,j}$	Elemento de posição (i, j) da matriz geradora \mathbf{G}
\mathbf{h}_i	i -ésimo vetor linha da matriz de teste de paridade \mathbf{H}
i e j	Índices de contagem
k	Número de bits da palavra mensagem \mathbf{m}
\mathbf{m}	Vetor palavra mensagem
\mathbf{m}_k	k -ésimo vetor palavra mensagem
m_i	i -ésimo bit da palavra mensagem \mathbf{m}
$m_{k,i}$	i -ésimo bit da k -ésima palavra mensagem \mathbf{m}_k
n	Número de bits da palavra código \mathbf{c}
$n(t)$	Sinal de ruído no domínio do tempo
nc_j	i -ésimo nó de variável
nh_i	j -ésimo nó de verificação
$p_{i,j}$	Elemento binário de posição (i, j) da submatriz de teste de paridade \mathbf{P}
q	Ordem do campo de Galois
\mathbf{r}	Vetor de sinal recebido
r_i	i -ésimo elemento do vetor de sinal recebido \mathbf{r}
$r(t)$	Sinal recebido no domínio do tempo
\mathbf{s}	Vetor de sinal transmitido
s_i	i -ésimo elemento do vetor sinal transmitido \mathbf{s}
$s(t)$	Sinal enviado no domínio do tempo
t	Capacidade garantida de correção do código em bits
x	Bit 0 ou 1
x_1 e x_2	Abcissas dos pontos 1 e 2 no espaço de sinais bidimensional
y_1 e y_2	Ordenadas dos pontos 1 e 2 no espaço de sinais bidimensional
w_c	Peso das colunas da matriz de teste de paridade
w_r	Peso das linhas da matriz de teste de paridade
A_i	Número de palavras código com peso i
$A(z)$	Polinômio enumerador de pesos do código C
B	Banda elétrica
$B(z)$	Polinômio enumerador de pesos do código C^\perp
B_{ref}	Banda óptica de referência
C	Código de bloco
C_{Total}	Capacitância Total
E_b	Energia média de bit
E_{bdec}	Energia por bit decodificado

$E_{b_{dec/iter}}$	Energia por bit decodificado por iteração
E_c	Energia média de bit codificado
G	Matriz geradora
\mathbf{G}_{sis}	Matriz geradora na forma sistemática
H	Matriz de teste de paridade
\mathbf{H}_{sis}	Matriz de teste de paridade na forma sistemática
\mathbf{I}_k	Matriz identidade de dimensões k por k
I_{CCAUXQ}	Corrente quiescente correspondente à tensão de alimentação V_{CCAUX} do FPGA
I_{CCINTQ}	Corrente quiescente correspondente à tensão de alimentação V_{CCINT} do FPGA
I_{CCOQ}	Corrente quiescente correspondente à tensão de alimentação V_{CCO} do FPGA
L	Número de ciclos por iteração efetuados pelo decodificador
M	Conjunto de palavras mensagem
M_c	Conjunto de palavras código
MA	Submatriz da matriz de teste de paridade de Richardson
MB	Submatriz da matriz de teste de paridade de Richardson
MC	Submatriz da matriz de teste de paridade de Richardson
MD	Submatriz da matriz de teste de paridade de Richardson
ME	Submatriz da matriz de teste de paridade de Richardson
MT	Submatriz da matriz de teste de paridade de Richardson
$M(j)$	Lista de índices de todos os nós de verificação conectados ao nó de variável nc_j
$M(j)\setminus i$	Lista de índices $M(j)$ com exceção do nó de verificação nh_i
$N(f)$	Densidade espectral de potência do sinal $n(t)$
$N(i)$	Lista de índices de todos os nós de variável conectados ao nó de verificação nh_i
$N(i)\setminus j$	Lista de índices $N(i)$ com exceção do nó de variável nc_j
$N_0/2$	Densidade espectral de ruído AWGN
$N_{ASE}/2$	Densidade espectral de potência de ruído ASE
N_{blocos}	Número de blocos de palavras mensagem ou palavras código
N_{iter}	Número de iterações efetuadas pelo decodificador
OH	Redundância
P	Submatriz de teste de paridade
P_b	Probabilidade de erro de bit
P_{Din}	Potência dinâmica da FPGA
$P_{Din/iter}$	Potência dinâmica da FPGA por iteração

P_{Est}	Potência estática da FPGA
P_N	Potência do ruído AWGN
P_R	Potência do sinal recebido
P_{Total}	Potência total do FPGA
$P_{Total/iter}$	Potência total do FPGA por iteração
Q	Fator Q
$Q(x)$	Valor complementar da função de distribuição acumulativa da variável aleatória normal x
Q_j^x	Probabilidade <i>a posteriori</i> do bit recebido
Q_{ij}^x	Estimativa enviada pelo nó de variável nc_j para o nó filhote nh_i
R	Taxa de código
R_{ij}^x	Estimativa enviada pelo nó de verificação nh_i para cada nó progenitor nc_j
R_S	Taxa de símbolos
S	Conjunto da constelação de sinais
T	Vazão de dados em bits por segundo
T_S	Taxa de transmissão em símbolos por segundo
V_{CCAUX}	Tensão auxiliar de alimentação do FPGA
V_{CCINT}	Tensão interna de alimentação do FPGA
V_{CCO}	Tensão de alimentação das portas de saída do FPGA

SIGLAS

ARQ	Repetição Automática de Retransmissão (<i>Automatic Repeat- reQuest</i>)
ASE	Emissão Espontânea Amplificada (<i>Amplified Spontaneous Emission</i>)
ASIC	Circuito Integrado de Aplicação Específica (<i>Application-Specific In- tegrated Circuit</i>)
AWGN	Ruído Branco Aditivo Gaussiano (<i>Additive White Gaussian Noise</i>)
BCH	Código Bose-Chaudhuri-Hocquenghem
BER	Taxa de Erro de Bit (<i>Bit Error Rate</i>)
BPSK	Chaveamento de Fase Binário (<i>Binary Phase Shift Keying</i>)
BSC	Canal Binário Simétrico (<i>Binary Symmetric Channel</i>)
CC	Corrente Contínua
CD	Dispersão Cromática (<i>Chromatic Dispersion</i>)
CG	Ganho de Código (<i>Code Gain</i>)
CMOS	Semicondutor Metal-Óxido-Complementar (<i>Complementary Metal- Oxide-Semiconductor</i>)
DP-QPSK	Multiplexação em Polarizações Ortogonais com Modulação QPSK (<i>Dual Polarization - QPSK</i>)
DSP	Processamento Digital de Sinal (<i>Digital Signal Processing</i>)
DVB-S2	Padrão de Transmissão Digital de Vídeo - Segunda Geração (<i>Digital Video Broadcasting - Second Generation</i>)
FEC	Código Corretor de Erro (<i>Forward Error Correction</i>)
FET	Transistor de Efeito de Campo (<i>Field-Effect Transistor</i>)
FPGA	Arranjo de Portas Programáveis em Campo (<i>Field-Programmable Gate Array</i>)
GF(q)	Campo de Galois de Ordem q (<i>Order q Galois Field</i>)
IPTV	Televisão via Protocolo de Internet (<i>Internet Protocol Television</i>)
ITU-T	União Internacional de Telecomunicações - Setor de Normatização das Telecomunicações (<i>International Telecommunications Union - Standardization Sector</i>)
LDPC	Código de Teste de Paridade de Baixa Densidade (<i>Low-Density Parity-Check</i>)
LED	Diodo emissor de luz (<i>light emitting diode</i>)
MAP	Máximo <i>A Posteriori</i> (<i>Maximum A Posteriori</i>)
ML	Máxima Verossimilhança (<i>Maximum Likelihood</i>)
OH	Redundância (<i>Overhead</i>)

OOK	Chaveamento por Liga-Desliga (<i>On-Off Keying</i>)
OSNR	Razão Sinal-Ruído Óptica (<i>Optical Signal-to-Noise Ratio</i>)
QAM	Modulação de Amplitude em Quadratura (<i>Quadrature Amplitude Modulation</i>)
QPSK	Modulação por Chaveamento de Fase em Quadratura (<i>Quadrature Phase Shift Keying</i>)
PDF	Função de Densidade de Probabilidade (<i>Probability Density Function</i>)
PMD	Dispersão dos Modos de Polarização (<i>Polarization Mode Dispersion</i>)
RS	Código Reed-Solomon
SMF	Fibra Monomodo (<i>Single Mode Fiber</i>)
SNR	Razão sinal-ruído (<i>Signal-to-Noise Ratio</i>)
VHDL	Linguagem de Descrição de <i>Hardware</i> VHSIC (<i>VHSIC Hardware Description Language</i>)
VHSIC	Circuito Integrado de Velocidade Muito Alta (<i>Very-High-Speed Integrated Circuit</i>)
VLSI	Integração em Grande Escala (<i>Very-Large-Scale Integration</i>)
VoIP	Voz sobre Protocolo de Internet (<i>Voice Over Internet Protocol</i>)
WDM	Multiplexação por Divisão de Comprimento de Onda (<i>Wavelength Division Multiplexing</i>)
WiMAX	Padrão de Interoperatividade Mundial para Acesso de Microondas (<i>Worldwide Interoperability for Microwave Access</i>)

Capítulo 1

INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

Nos últimos anos o tráfego de dados da Internet verificou crescimentos vertiginosos da ordem de 75% a 125% por ano [5]. Dentre as tecnologias que demandam o aumento crescente de banda da Internet se destacam IPTV, VoIP e transmissão de vídeo em alta definição. É evidente que esse crescente volume de troca de dados está intrinsecamente relacionado com a evolução dos sistemas ópticos, que já suportam capacidades de até 100 Gb/s por canal e cobrem enlaces de distâncias intercontinentais [6].

Um dos objetivos principais das evoluções tecnológicas relacionadas com as comunicações ópticas é o aumento do produto da banda disponível pela distância máxima de propagação alcançável [6]. Outro objetivo é a diminuição dos custos envolvidos na implementação de sistemas ópticos de comunicação, que está relacionada com o aumento da eficiência espectral do sistema [7].

No entanto, existem alguns fatores que dificultam a construção de enlaces ópticos eficientes a altas taxas de transmissão e a longas distâncias. Além da atenuação causada pela fibra, fatores degradantes do sinal, como ruído, dispersão cromática (*chromatic dispersion* - CD), dispersão dos modos de polarização (*polarization mode dispersion* - PMD) e efeitos não-lineares (interferências inter e intracanaís) completam a lista de obstáculos impostos para a transmissão óptica [6].

Desse maneira, é necessário um estudo constante de novas técnicas de modulação, de processamento de sinais (*digital signal processing* - DSP) e, em especial, de codificação de canal, que é o foco deste trabalho, representando uma das técnicas que permitem o aumento da robustez do sistema quanto à ocorrência de bits errados na recepção.

A codificação de canal é definida como a prática de se mapear uma sequência de k símbolos de informação em uma palavra código de n símbolos, formalmente representada por

$$C : M \rightarrow M_c, \quad (1.1)$$

em que C é o código de canal, M é o conjunto de palavras de informação (ou palavras mensagem) de comprimento k e M_c é o conjunto de palavras código de tamanho n . Os $n - k$ símbolos agregados

a cada palavra mensagem são utilizados como informação adicional para se determinar qual foi a palavra transmitida. Há três maneiras de se aplicar a codificação de canal [5]:

1. Detecção de Erro - O objetivo é somente a identificação da ocorrência de erros na estimação da mensagem transmitida. Um exemplo desse tipo de aplicação é o protocolo ARQ (*automatic repeat request*) [8], em que, caso seja identificada no receptor a existência de erro nos dados enviados, solicita-se uma nova transmissão da mensagem em questão;
2. FEC (*forward error correction*) - Os bits de redundância inseridos na transmissão permitem que, posteriormente, o receptor consiga detectar e corrigir um número máximo de bits com garantia de sucesso;
3. Códigos Híbridos - Aplicação que combina FEC com detecção de erro.

Dado que os sistemas ópticos atuais operam a altas taxas de transmissão, a utilização de codificação de canal baseada em requisição de retransmissão está fora de cogitação devido a dois fatores básicos: primeiro, por causa da latência adicional envolvida em cada retransmissão; segundo, devido à necessidade de memória para armazenamento dos dados enviados entre retransmissões. Desse modo, toda a análise deste trabalho será baseada em códigos FEC.

Uma das métricas de avaliação da qualidade do sinal recebido é a taxa de erro de bit (*bit error rate* - BER). Em comunicações ópticas, a BER requerida para um sistema confiável é menor ou igual a 10^{-15} [9]. Para uma relação sinal-ruído (*signal-to-noise ratio* - SNR) fixa, uma das maneiras de se melhorar a qualidade do sinal recebido é com a utilização de codificação de canal [5]. Logo, requer-se a utilização de códigos corretores eficientes para se atingir taxas de erro tão baixas em níveis de razão sinal-ruído aceitáveis para cada sistema.

1.1.1 Breve histórico - FEC em comunicações ópticas

- Primeiras Propostas

Até o final da década de 1980, a aplicação de códigos corretores de erro restringia-se a sistemas de comunicação sem fio, como comunicações via rádio e via satélite. Enquanto isso, os sistemas ópticos possuíam as altas taxas de transmissão como obstáculo para a inclusão de redundância (*overhead*) nas suas transmissões. Nesse contexto, em 1988, surgiu a primeira proposta de FEC para redes ópticas [10], que sugeriu a utilização de um código de Hamming encurtado (224,216), com taxa $R = 0,964$ e taxa de transmissão igual a 565 Mb/s. O ganho de código obtido era de apenas 2,5 dB para uma $BER = 10^{-13}$. O primeiro sistema óptico implementado com FEC [11] utilizava código BCH(167,151), permitindo que os *spans* submarinos chegassem a 200 km de alcance (sem amplificação) com uma taxa de 565 Mb/s.

- 1ª Geração

O primeiro código padronizado (1993) para comunicações ópticas foi o Reed-Solomon (RS) com $n = 255$ e $k = 239$, originando a denominada 1ª geração FEC, caracterizando as

padronizações ITU-T G.975 [12] e ITU-T G.709 [13]. Por não se tratar de uma codificação binária, o Reed-Solomon especificado proporciona uma boa correção dos erros em rajada. Além disso, o RS(255,239) tem capacidade de correção de até 8 *bytes* e possui ganho de código de aproximadamente 6,1 dB ($BER = 10^{-13}$) [14, 15]. A aplicação dessa codificação foi voltada especialmente para sistemas ópticos submarinos de longa distância.

- 2ª Geração

Nos anos 2000, com a introdução dos sistemas com multiplexação em comprimento de onda (*wavelength-division multiplexing* - WDM) [15], tornou-se necessária a busca de codificações mais robustas. Nesse contexto surgiu, entre 2000 e 2004, a proposta de variados tipos de códigos concatenados, originando assim a 2ª geração. A ideia básica era a concatenação em série de códigos BCH, RS ou Hamming nas etapas de codificação e decodificação, de modo que a codificação equivalente se tornasse mais robusta quanto à correção de erros. A Figura 1.1 descreve o esquema de concatenação de códigos. À luz dessa ideia, foi criada a padronização ITU-T G.975.1 (2004) [1]. O melhor desempenho encontrado nessa geração proporciona ganho de código de 10,3 dB para $R = 0,8$ [5].

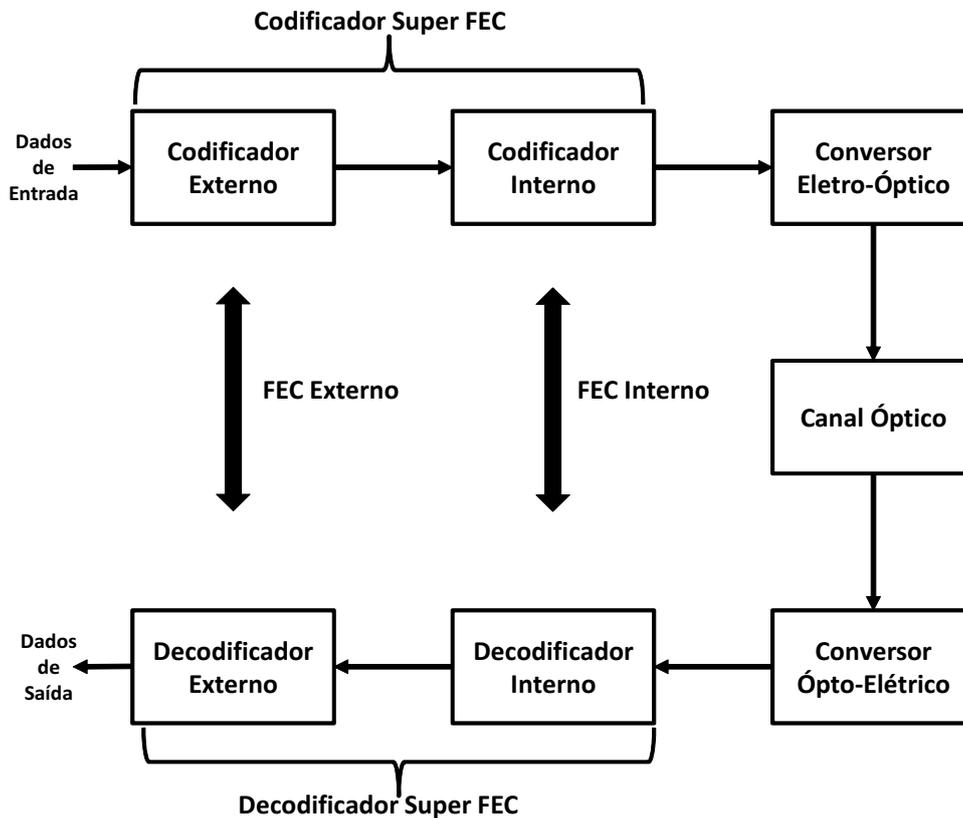


Figura 1.1: Esquema de concatenação de códigos (Super FEC) da 2ª geração. Adaptado de [1].

- 3ª Geração

Apesar de a segunda geração de códigos ter representado um significativo avanço na capacidade de transmissão, ela ainda não proporcionava a margem de ganho de código requerida

para a implementação em enlaces de distâncias intercontinentais [15]. Os códigos propostos na 3ª geração são baseados em códigos com decodificação *soft* iterativa, como os códigos Turbo e o LDPC (*Low-Density Parity-Check*). Em [15–17] mostra-se que códigos LDPC possuem desempenho de taxa de erro de bit superior tanto com relação aos códigos das gerações FEC anteriores quanto aos códigos Turbo-Produto. Outra vantagem do LDPC é a menor complexidade do decodificador com relação ao decodificador dos códigos Turbo-Produto [17]. Com base nesses fatos, os códigos LDPC são considerados fortes candidatos para serem implementados em sistemas a 100 Gb/s.

1.2 OBJETIVO DO TRABALHO

Dado que o atual estado da arte das comunicações ópticas indica a codificação LDPC como a mais promissora para a implementação de FEC em sistemas práticos, o presente trabalho focará no estudo de códigos LDPC aplicáveis a enlaces ópticos. Além do desempenho de taxa de erro de bit e de ganho de codificação, outro fator importante a ser estudado é o consumo de potência requerido por decodificadores LDPC e a sua variação com relação à BER atingida.

A primeira parte do projeto analisa o desempenho de BER para códigos LDPC de altas taxas de código ($R \geq 0,8$). A razão de se limitar o valor mínimo da taxa de código deve-se ao acréscimo de banda óptica requerida pela utilização de codificação de canal.

Em seguida, realiza-se uma implementação em VHDL do algoritmo de decodificação. Essa implementação é baseada numa metodologia automatizada de conversão de um código descrito em Matlab para uma descrição em linguagem de *hardware*. Por fim, estima-se a potência de consumo do decodificador, relacionando-a com a BER obtida.

Este trabalho obedece à seguinte divisão: o Capítulo 2 faz uma descrição geral do sistema óptico de comunicação e de como as etapas de codificação e decodificação estão inseridas nesse sistema. O Capítulo 3 apresenta os códigos de bloco lineares, classe de códigos que representa a grande maioria dos códigos utilizados em sistemas práticos, incluindo o LDPC. O Capítulo 4 apresenta os códigos LDPC e o algoritmo de decodificação a ser estudado, o Soma-Produto. O Capítulo 5 descreve brevemente o modelo de potência de consumo de arranjos de portas programáveis em campo (*field programmable gate arrays* - FPGAs) e as etapas de implementação de um código em FPGA. O Capítulo 6 apresenta o desenvolvimento e os resultados da estimação de desempenho de BER de códigos LDPC e calcula a potência requerida para decodificação desses códigos. Por fim, o Capítulo 7 tece as considerações finais e discorre sobre trabalhos futuros.

Capítulo 2

SISTEMA ÓPTICO DE COMUNICAÇÃO

Neste capítulo apresenta-se o modelo básico da camada física de um sistema óptico de comunicação. Modela-se matematicamente o ruído inserido pelo canal e descrevem-se algumas das principais métricas de avaliação de desempenho dos sistemas ópticos.

2.1 VISÃO GERAL

De modo geral, um sistema de comunicação óptico monocanal pode ser representado por um transmissor, onde é realizada a transdução elétrico-óptica, por um canal óptico e por um receptor, onde é feita a conversão ópto-elétrica do sinal. O canal óptico considerado neste trabalho é uma fibra monomodo (*single mode fiber* - SMF). Como o sinal óptico é atenuado ao longo da fibra, são necessários amplificadores ópticos ao longo do canal para que o sinal chegue ao final do enlace com uma potência igual ou superior à sensibilidade do receptor.

O sistema óptico pode ser classificado de acordo com vários critérios, como distância de enlace e taxa de transmissão. Quanto ao primeiro critério, pode-se classificar os sistemas ópticos em duas categorias: os sistemas de curta distância (*short-haul*) e os de longa distância (*long-haul*). Os de curta distância são caracterizados por enlaces de até 100 km e geralmente se aplicam a comunicações ópticas dentro de cidades ou em centros de processamento de dados. Os de longa distância cobrem desde enlaces interidades até comunicações intercontinentais.

Podem-se estudar com mais detalhes os elementos que compõem o transmissor e o receptor, como ilustra a Figura 2.1. Inicia-se a transmissão no gerador de bits, o qual, para fins de análise, pode ser considerado como um gerador de uma sequência aleatória de bits. Em seguida, insere-se redundância na informação gerada com o intuito de tornar os dados enviados mais robustos a erros. O próximo passo é a modulação digital dos bits, tipicamente para os formatos de chaveamento liga-desliga (*on-off keying* - OOK), chaveamento de fase binário (*binary phase-shifting* - BPSK), modulação por chaveamento de fase em quadratura (*quadrature phase-shifting* - QPSK)

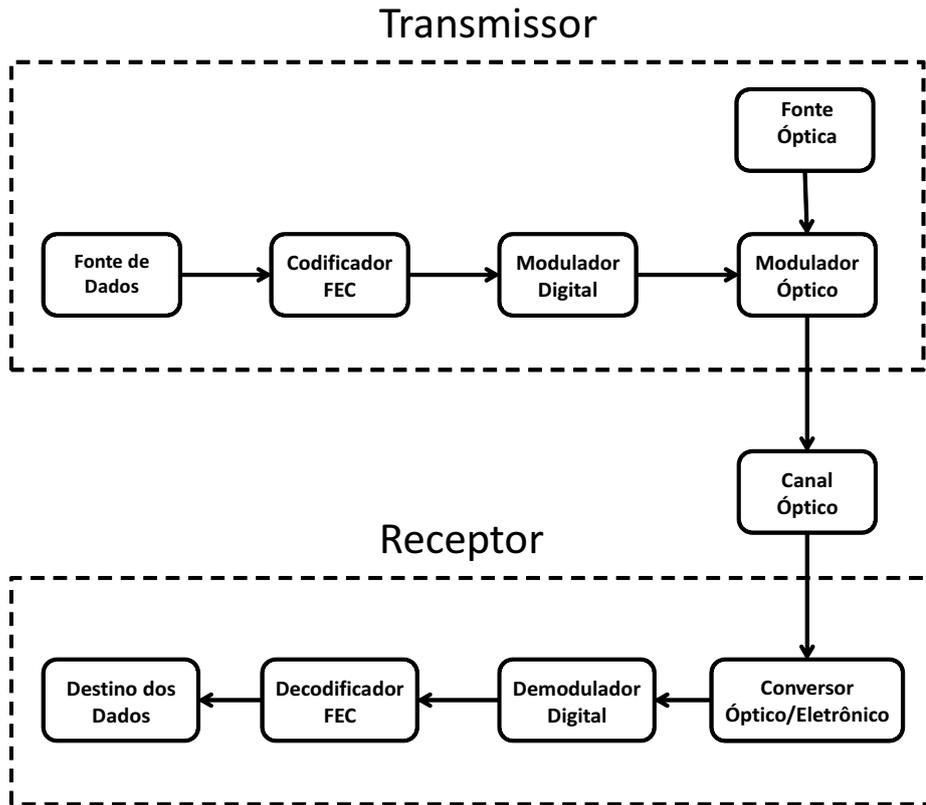


Figura 2.1: Diagrama de blocos geral de um sistema óptico ponto-a-ponto.

e modulação de amplitude em quadratura (*quadrature amplitude modulation* - QAM) [7]. Após os símbolos estarem eletronicamente modulados, utiliza-se uma fonte óptica - laser ou LED (*light emitting diode*) - para modular opticamente o sinal a ser lançado na fibra.

No receptor de sistemas não coerentes, um fotodetector realiza a transdução, convertendo intensidade de luz em pulsos elétricos. Em seguida o demodulador mapeia cada símbolo na sequência equivalente de bits. Por fim, o decodificador FEC recebe as sequências de palavras código corrompidas por ruído e, com base em um certo algoritmo de processamento, realiza a estimação das palavras mensagem enviadas pelo gerador de bits.

O foco deste trabalho está na estimação de potência na etapa de decodificação FEC. Alguns trabalhos de comunicações sem fio como [18, 19] consideram a potência de codificação desprezível relativamente ao consumo de decodificação. Essa aproximação é razoável pelo fato de que a codificação consiste simplesmente em se multiplicar as palavras mensagem pela matriz geradora, como será detalhado na Equação 3.15.

Como os códigos da 3ª geração de FEC para redes ópticas utilizam decodificação iterativa, a complexidade matemática e o número de operações realizadas na decodificação implicam consumo significativo de potência do decodificador, em especial, no caso deste do trabalho, do LDPC.

2.2 MODELO DE CANAL ÓPTICO

A Seção 2.1 descreveu um sistema óptico de um único canal. Essa simplificação é necessária e suficiente para a análise de decodificação a ser feita neste trabalho. No entanto, em sistemas práticos emprega-se a multiplexação de canais em comprimento de onda, de modo que uma única fibra tem capacidade de suportar dezenas de canais espaçados no domínio da frequência. Em geral, o espaçamento é de 50 GHz por canal [20] e, tipicamente, multiplexam-se até 80 canais por fibra [6].

Para a análise da decodificação *soft* é necessário o conhecimento das distribuições de probabilidades posteriores do canal utilizado. Desse modo, deve-se adotar um modelo estatístico que descreva o comportamento do canal óptico.

Em [20], observa-se que, para um sistema a 100 Gb/s, sem compensação de dispersão, com multiplexação em polarizações ortogonais com modulação QPSK (*dual polarization* QPSK - DP-QPSK), as quatro componentes do sinal recebido - em fase/polarização horizontal, em quadratura/polarização horizontal, em fase/polarização vertical e em quadratura/polarização vertical - apresentam histogramas aproximadamente gaussianos. Desse modo, para fins de análise de desempenho de taxa de erro de bit, considera-se o ruído inserido pelo canal óptico como ruído aproximadamente gaussiano.

Logo, o modelo de ruído inserido no canal óptico é modelado por $n(t)$, em que o ruído é branco aditivo gaussiano (*additive white gaussian noise* - AWGN). O sinal recebido $r(t)$, na saída do canal, é descrito por

$$r(t) = s(t) + n(t). \quad (2.1)$$

A densidade espectral de potência de $n(t)$ é dada por

$$N(f) = N_0/2, \text{ para } -\infty < f < \infty, \quad (2.2)$$

e a função densidade de probabilidade (*probability density function* - PDF) é uma distribuição gaussiana com variância $\sigma^2 = N_0/2$ (para modulação BPSK).

2.3 MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO

- BER

Um dos critérios de avaliação da qualidade do sinal recebido é a taxa de erro de bit (*bit error rate* - BER). Na prática ela é estimada pela razão do número de bits de informação erroneamente estimados pelo número total de bits de informação recebidos.

Após a estimação do vetor codificado \tilde{c} , deve-se, a princípio, definir quais são os bits de informação e descartar os de paridade. Tratando-se de desempenho de taxa de erro de bit de códigos lineares, assim como Mackay em [21], calcula-se a média da BER considerando todos os bits codificados e estimados envolvidos. Para códigos lineares é esperado que a BER

calculada para bits de informação e de paridade seja aproximadamente a mesma calculada para os bits de informação isoladamente, considerando uma quantidade razoável de blocos (quantidade de palavras código) simulados. Desse modo, o cálculo da BER é realizado matematicamente da seguinte forma:

$$BER = \frac{\text{número total de bits errados}}{\text{número total de bits codificados}} = \frac{\sum_{k=1}^{N_{\text{blocos}}} [d_H(\mathbf{c}_k, \tilde{\mathbf{c}}_k)]}{N_{\text{blocos}} * n}, \quad (2.3)$$

em que N_{blocos} representa o total de palavras código enviadas na simulação e a distância de Hamming $d_H(\cdot, \cdot)$ entre cada palavra código enviada (\mathbf{c}_k) e decidida ($\tilde{\mathbf{c}}_k$) é definida por

$$d_H(\mathbf{c}_k, \tilde{\mathbf{c}}_k) = \sum_{i=1}^n (c_{k,i} \oplus \tilde{c}_{k,i}), \quad (2.4)$$

em que o operador “ \oplus ” realiza a operação soma módulo-2 (ou-exclusivo), e $c_{k,i}$ e $\tilde{c}_{k,i}$ são os i -ésimos bits do k -ésimo bloco enviado e estimado, respectivamente.

- SNR, E_b/N_0 e OSNR

A razão sinal-ruído (*signal-to-noise ratio* - SNR) medida na entrada do receptor é definida por [22]

$$SNR = \frac{P_R}{P_N}, \quad (2.5)$$

em que P_R é a potência útil do sinal elétrico no receptor e P_N é a potência total do ruído na banda do sinal transmitido. Sabendo a densidade espectral do ruído AWGN, a taxa de símbolos R_S do sistema [23] e que a potência recebida é igual ao produto da energia de bit E_b pela taxa de símbolos, encontra-se que outra forma de se calcular a SNR (para modulação BPSK e banda elétrica B igual a R_S) é dada por

$$SNR = \frac{P_R}{N_0 R_S} = \frac{E_b}{N_0}. \quad (2.6)$$

A relação sinal-ruído óptica (*optical signal-to-noise ratio* - OSNR) é a razão entre a potência média P_R do sinal elétrico no receptor nos dois estados de polarização e a potência P_{ASE} do ruído de emissão espontânea amplificada (*amplified spontaneous emission* - ASE). Desse modo, pode-se expressar a OSNR por [23]

$$OSNR = \frac{P_R}{P_{ASE}} = \frac{P_R}{2N_{ASE}B_{ref}}, \quad (2.7)$$

em que $N_{ASE}/2 = N_0/2$ é a densidade espectral de ruído ASE em uma única polarização e B_{ref} é a banda óptica de referência (tipicamente igual a 12,5 GHz [6]) para a qual P_{ASE} está definida.

A relação entre SNR e OSNR é dada por [23]

$$OSNR = \frac{pR_S}{2B_{ref}} SNR, \quad (2.8)$$

em que $p = 1$ para sinal de única polarização e $p = 2$ para sinal com multiplexação de polarização.

- Fator Q

Considerando um sistema OOK com detecção direta, define-se [6] o fator Q por

$$Q = \frac{|\mu_1 - \mu_0|}{\sigma_1 + \sigma_0}, \quad (2.9)$$

em que μ_x e σ_x são, respectivamente, os valores médios e os desvios padrão dos sinais elétricos dos bits para $x \in \{0, 1\}$.

- Ganho de Código

A avaliação da eficiência da correção de erros de um certo código é avaliada principalmente com o auxílio da curva de BER com relação à razão sinal-ruído elétrica, ou com respeito à razão E_b/N_0 ou ainda com relação ao fator Q . Em geral, compara-se o desempenho do sistema com e sem o uso de FEC. Quanto mais próximo do limite de Shannon [24], melhor é a eficiência do código proposto.

Para quantificar o grau de eficiência de um certo código integrado num sistema de comunicação, utiliza-se a métrica do ganho de código (*code gain* - CG) [14, 15], que mostra de quantos dBs é possível reduzir a razão sinal-ruído de um sistema sem codificação para que ele proporcione uma probabilidade de erro de bit caso existisse uma certa codificação. Como exemplo, a Figura 2.2 compara a taxa de erro de bit de um sistema BPSK sem codificação com a curva de desempenho de um código LDPC(64800,58320). Para $BER = 10^{-4}$, $CG = 4,7 \text{ dB}$.

As figuras de mérito utilizadas neste trabalho são a taxa de erro de bit, o ganho de código e a razão E_b/N_0 . Como o fator Q e a OSNR são muito recorrentes na literatura de comunicações ópticas [5, 6, 16], é importante explicitar suas definições. Não se fez necessário o uso desses dois últimos parâmetros como referências de análise neste trabalho, pois toda a simulação foi feita na camada elétrica. De qualquer modo, a ideia de que o fator Q e a OSNR são proporcionais a E_b/N_0 já é suficiente para que a análise de desempenho possa ser feita com relação a essas métricas.

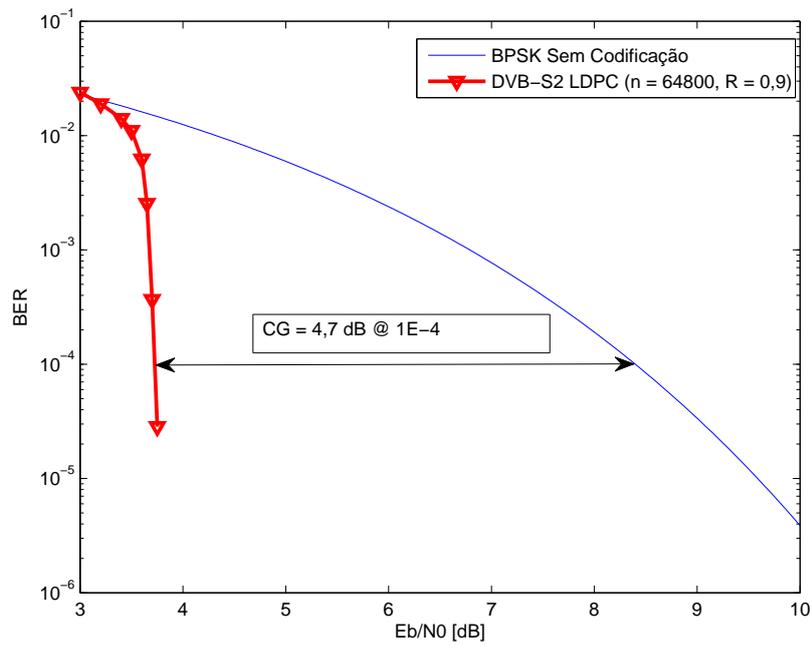


Figura 2.2: Ganho de código entre codificação LDPC(64800, 58320) e sistema BPSK sem codificação.

Capítulo 3

CÓDIGOS DE BLOCO LINEARES

Neste capítulo apresentam-se os principais aspectos presentes na teoria de códigos corretores de erro. Abordam-se definições importantes que são necessárias para o entendimento da construção e da escolha das características dos códigos de bloco lineares.

3.1 DEFINIÇÕES BÁSICAS

Considera-se um código de bloco linear $C(n, k)$, $n > k$. Na entrada do codificador, os vetores mensagem possuem comprimento fixo de k símbolos e, após a codificação, as palavras código têm comprimento também constante de n símbolos, justificando assim a alcunha de “código de bloco”. A classificação de “linear” é justificada por duas razões [24]. A primeira, pelo fato de que a combinação linear de palavras código resulta em outra palavra pertencente ao mesmo código. A segunda razão define que a palavra código nula (formada somente por zeros) obrigatoriamente faça parte de qualquer código de bloco linear.

O estudo deste trabalho se concentra em códigos pertencentes ao campo de Galois de ordem $q = 2$ (*order q galois field - GF(q)*) [2], ou seja, os símbolos são representado pelos bits $\{0,1\}$. As 2^k possíveis combinações de mensagens corresponderão cada uma a uma dada palavra código específica.

3.1.1 Taxa de código e redundância

A taxa R de um código de bloco é definida por $R = \frac{k}{n}$. Os k bits de entrada são designados como bits de informação, e os $n - k$ bits adicionais são conhecidos como redundância ou bits de paridade do código. A redundância (*overhead - OH*) é quantificada pelo percentual relativo de bits de paridade com relação ao total de bits de informação [23]. Assim,

$$OH = \left(\frac{n - k}{k}\right) * 100\% = \frac{1 - R}{R} * 100\%. \quad (3.1)$$

3.1.2 Distância e peso de Hamming

O peso de Hamming ($w_H(\mathbf{c})$) quantifica o número de elementos não-nulos de uma palavra código \mathbf{c} . Para $\mathbf{c} = (c_1, c_2, \dots, c_n)$, o peso de Hamming é dado por

$$w_H(\mathbf{c}) = \sum_{i=1}^n c_i = c_1 + c_2 + \dots + c_n. \quad (3.2)$$

Já a distância de Hamming define o número de posições nas quais os bits de duas palavras código diferem entre si. Sua definição matemática se encontra na Equação 2.4.

3.1.3 Distância mínima

A distância mínima é igual à menor distância de Hamming de um código. Para um código linear C , a sua distância mínima d_{min} é igual ao peso mínimo de Hamming [2], pois a distância mínima é igual a distância de Hamming entre a palavra de menor peso e a palavra nula. Enunciando a afirmação anterior matematicamente, a partir do princípio da linearidade dos códigos de bloco, tem-se que

$$d_{min} = \min_{\mathbf{c}_a, \mathbf{c}_b \in C, \mathbf{c}_a \neq \mathbf{c}_b} d_H(\mathbf{c}_a, \mathbf{c}_b) = \min_{\mathbf{c}_a, \mathbf{c}_b \in C, \mathbf{c}_a \neq \mathbf{c}_b} d_H(\mathbf{c}_a - \mathbf{c}_b, \mathbf{0}) = \min_{\mathbf{c} \in C, \mathbf{c} \neq \mathbf{0}} w_H(\mathbf{c}). \quad (3.3)$$

3.1.4 Distância Euclidiana

Sejam $s_1(t)$ e $s_2(t)$ os seguintes sinais do espaço bidimensional

$$\begin{aligned} s_1(t) &= x_1\phi_1(t) + y_1\phi_2(t) \text{ e} \\ s_2(t) &= x_2\phi_1(t) + y_2\phi_2(t), \end{aligned} \quad (3.4)$$

em que ϕ_1 e ϕ_2 são funções ortonormais que definem o espaço bidimensional no qual $s_1(t)$ e $s_2(t)$ estão contidos.

A distância $d(s_1(t), s_2(t))$ entre $s_1(t)$ e $s_2(t)$ é definida por [2]

$$d(s_1(t), s_2(t)) = \sqrt{\int_{-\infty}^{\infty} (s_1(t) - s_2(t))^2 dt}. \quad (3.5)$$

Como os sinais $s_1(t)$ e $s_2(t)$ são descritos no espaço de sinais, respectivamente, pelos pontos (x_1, y_1) e (x_2, y_2) da Figura 3.1, pode-se calcular a distância $d(s_1(t), s_2(t))$ pela distância geométrica ou Euclidiana entre os sinais, definida por

$$d(s_1(t), s_2(t)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (3.6)$$

Deseja-se agora estender a definição de distância Euclidiana da Equação 3.6 para distância Euclidiana entre palavras código, para depois relacioná-la com distância de Hamming.

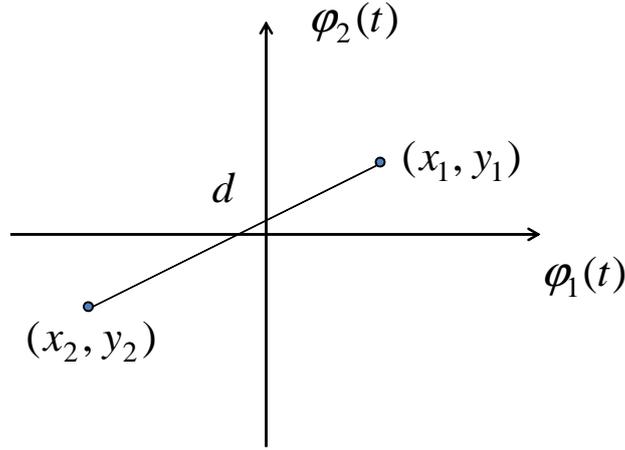


Figura 3.1: Espaço de sinais bidimensional. Distância Euclidiana d entre os sinais $s_1(t)$ - coordenadas (x_1, y_1) - e $s_2(t)$ - coordenadas (x_2, y_2) .

Supõe-se dois vetores \mathbf{c}_0 e \mathbf{c}_1 , dados por

$$\begin{aligned} \mathbf{c}_0 &= \{c_{0,1}, c_{0,2}, \dots, c_{0,n}\} \text{ e} \\ \mathbf{c}_1 &= \{c_{1,1}, c_{1,2}, \dots, c_{1,n}\}. \end{aligned} \quad (3.7)$$

Os vetores de sinais resultantes da modulação BPSK, \mathbf{s}_0 e \mathbf{s}_1 , são representados por

$$\begin{aligned} \mathbf{s}_0 &= (2\mathbf{c}_0 - 1)\sqrt{E_c} \text{ e} \\ \mathbf{s}_1 &= (2\mathbf{c}_1 - 1)\sqrt{E_c}. \end{aligned} \quad (3.8)$$

A distância Euclidiana entre as duas palavras código \mathbf{c}_1 e \mathbf{c}_2 é calculada pela distância Euclidiana dos seus respectivos vetores de sinais \mathbf{s}_1 e \mathbf{s}_2 , contidos no espaço de sinais n -dimensional. Assim,

$$d_E(\mathbf{s}_0, \mathbf{s}_1) = \sqrt{\sum_{i=1}^n (s_{0,i} - s_{1,i})^2} = \sqrt{\sum_{i=1}^n [(2c_{0,i} - 1)\sqrt{E_c} - (2c_{1,i} - 1)\sqrt{E_c}]^2}. \quad (3.9)$$

Os bits 0 e 1 são modulados em $-\sqrt{E_c}$, $+\sqrt{E_c}$, respectivamente. Simplificando a Equação 3.9, tem-se que

$$d_E(\mathbf{s}_0, \mathbf{s}_1) = \sqrt{4E_c d_H(\mathbf{c}_0, \mathbf{c}_1)} = 2\sqrt{E_c d_H(\mathbf{c}_0, \mathbf{c}_1)}. \quad (3.10)$$

3.2 REPRESENTAÇÃO MATRICIAL

Como o código C representa um espaço vetorial k -dimensional, existem k vetores linearmente independentes $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k$ tais que

$$\mathbf{c} = m_1\mathbf{g}_1 \oplus m_2\mathbf{g}_2 \oplus \dots \oplus m_k\mathbf{g}_k, \quad (3.11)$$

em que

$$\mathbf{m} = [m_1 \quad m_2 \quad \dots \quad m_k] \text{ e} \quad (3.12)$$

$$\mathbf{g}_i = [g_{i,1} \quad g_{i,2} \quad \dots \quad g_{i,n}]. \quad (3.13)$$

O conjunto dos k vetores linha \mathbf{g}_i é representado pela matriz geradora \mathbf{G} , dada por

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_k \end{bmatrix} = \begin{bmatrix} g_{1,1} & g_{1,2} & \dots & g_{1,n} \\ g_{2,1} & g_{2,2} & \dots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1} & g_{k,2} & \dots & g_{k,n} \end{bmatrix}. \quad (3.14)$$

Logo, o vetor \mathbf{c} da Equação 3.11 pode ser escrito como

$$\mathbf{c} = \mathbf{mG}. \quad (3.15)$$

Exemplo 3.1 Dada a matriz geradora de um código de Hamming (7,4) [2] representada por

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad (3.16)$$

deseja-se codificar a mensagem $\mathbf{m} = [0 \ 1 \ 0 \ 1]$.

Pela Equação 3.15, a codificação de \mathbf{m} se resume a realização da soma módulo-2 da segunda e quarta linha da matriz \mathbf{G} da Equação 3.16. Assim, calcula-se o vetor codificado da seguinte maneira:

$$\mathbf{c} = \mathbf{mG} = [0 \ 1 \ 0 \ 1] \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad (3.17)$$

$$\mathbf{c} = [0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]. \quad (3.18)$$

Observa-se que o vetor mensagem $\mathbf{m} = [0 \ 1 \ 0 \ 1]$ não pode ser identificado isoladamente no vetor codificado $\mathbf{c} = [0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$. No entanto, em alguns casos é desejável que a palavra codificada possua os bits de informação explícitos, precedendo ou sucedendo os bits de paridade, como mostrado na Figura 3.2. Para tanto, torna-se necessária a definição de códigos sistemáticos da Subseção 3.2.1.

3.2.1 Códigos sistemáticos

Seja C' um código de bloco não necessariamente linear. Um codificador é dito sistemático se os bits de mensagem m_1, m_2, \dots, m_k puderem ser encontrados explicitamente, na mesma sequência, na palavra código. Para que isso ocorra, é necessário que a matriz geradora esteja na forma sistemática descrita por

$$\mathbf{G}_{sis} = [\mathbf{P} \quad \mathbf{I}_k] = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} & 1 & 0 & 0 & \cdots & 0 \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad (3.19)$$

em que \mathbf{I}_k é a matriz identidade de dimensão $k \times k$, \mathbf{P} é a chamada submatriz de teste de paridade de dimensões $k \times (n-k)$ e $p_{i,j}$ é o elemento binário da posição (i, j) da submatriz \mathbf{P} . A codificação sistemática pode ser descrita da seguinte forma:

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{sis} = \mathbf{m} [\mathbf{P} \quad \mathbf{I}_k] = [\mathbf{m}\mathbf{P} \quad \mathbf{m}]. \quad (3.20)$$

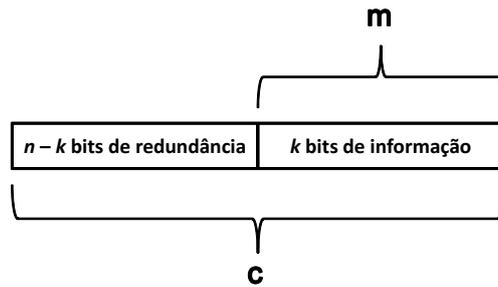


Figura 3.2: Formato sistemático da palavra código.

Verifica-se pela Equação 3.20 que, de fato, o vetor codificado é dividido em duas partes, mostradas na Figura 3.2: os bits mais à esquerda representam os bits de paridade ($\mathbf{m}\mathbf{P}$) e os mais à direita explicitam os bits de informação (\mathbf{m}).

Exemplo 3.2 Deseja-se transformar a matriz da Equação 3.16 na forma sistemática e, em seguida, codificar a mensagem $\mathbf{m} = [0 \ 1 \ 0 \ 1]$.

Uma das maneiras de se fazer isso se dá pela realização de operações lineares nos vetores linha da matriz geradora, com o objetivo de isolar a matriz identidade \mathbf{I}_4 dentro de \mathbf{G} . Seguindo a estrutura descrita na Equação 3.19 e fazendo as seguintes operações lineares propostas abaixo, obtém-se a matriz geradora sistemática (\mathbf{G}_{sis}):

$$\mathbf{g}_3 = \mathbf{g}_3 \oplus \mathbf{g}_1, \quad (3.21)$$

$$\mathbf{g}_4 = \mathbf{g}_4 \oplus \mathbf{g}_1 \oplus \mathbf{g}_2, \quad (3.22)$$

$$\mathbf{G}_{sis} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.23)$$

Codificando a mensagem $\mathbf{m} = [0 \ 1 \ 0 \ 1]$, encontra-se que

$$\mathbf{c} = [0 \ 1 \ 0 \ 1] * \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.24)$$

$$\mathbf{c} = [1 \ 1 \ 0 \ | \ 0 \ 1 \ 0 \ 1]. \quad (3.25)$$

No vetor codificado acima, observa-se a separação dos bits de paridade (à esquerda), dos bits de informação (à direita, em negrito).

3.2.2 Matriz de teste de paridade e equações de paridade

Dado um código $C(n, k)$, existe um código dual $C^\perp(n, n - k)$ tal que

$$\langle \mathbf{c}_i, \mathbf{c}_j^\perp \rangle = \mathbf{0}, \forall i / 0 \leq i < 2^k, \forall j / 0 \leq j < 2^k, \quad (3.26)$$

em que matriz geradora de C^\perp é a matriz de teste de paridade \mathbf{H} do código C e o operador “ $\langle \cdot, \cdot \rangle$ ” representa a operação de produto interno entre dois vetores. A matriz \mathbf{H} do código C é dada por

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{n-k} \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{bmatrix}, \quad (3.27)$$

e satisfaz a seguinte condição

$$\mathbf{GH}^T = \mathbf{0}, \mathbf{G} \in C. \quad (3.28)$$

Multiplicando por \mathbf{m} ambos os lados da Equação 3.28, tem-se que

$$\mathbf{mGH}^T = \mathbf{cH}^T = \mathbf{0} = [0 \ 0 \ 0 \ \cdots \ 0]_{1 \times n}. \quad (3.29)$$

Logo, \mathbf{c} é uma palavra código válida de C se a condição da Equação 3.29 for satisfeita. Substituindo os elementos do vetor \mathbf{c} e da matriz \mathbf{H} na Equação 3.29, tem-se que

$$\mathbf{c}\mathbf{H}^T = [c_1 \ c_2 \ \cdots \ c_n] * \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{bmatrix}^T. \quad (3.30)$$

Utilizando a matriz \mathbf{H} do código de Hamming(7,4) [2], descrita por

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \quad (3.31)$$

chega-se ao seguinte sistema de equações de paridade:

$$\begin{aligned} 1^{\text{a}} \text{ equação de paridade: } & c_1 \oplus c_4 \oplus c_6 \oplus c_7 = 0, \\ 2^{\text{a}} \text{ equação de paridade: } & c_2 \oplus c_4 \oplus c_5 \oplus c_6 = 0, \\ 3^{\text{a}} \text{ equação de paridade: } & c_3 \oplus c_5 \oplus c_6 \oplus c_7 = 0, \end{aligned} \quad (3.32)$$

em que a i -ésima equação de paridade da matriz \mathbf{H} é definida como a equação de paridade corresponde à i -ésima linha da matriz de teste de paridade. Logo, o sistema descrito na Equação 3.32 deve ser satisfeito para que $\mathbf{c} = [c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7]$ seja palavra código válida do código de Hamming descrito pela Equação 3.31.

De modo similar à matriz \mathbf{G}_{sis} , a matriz \mathbf{H} também possui uma representação particular para código sistemático, dada por

$$\mathbf{H}_{sis} = [\mathbf{I}_{n-k} \ \mathbf{P}^T] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{1,1} & p_{2,1} & \cdots & p_{k,1} \\ 0 & 1 & 0 & \cdots & 0 & p_{1,2} & p_{2,2} & \cdots & p_{k,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} \end{bmatrix}. \quad (3.33)$$

Conhecendo a matriz de teste de paridade, é possível calcular a distância mínima do código sem a necessidade de se listar todas as 2^k palavras código e de se calcular o peso de cada uma delas. Para isso é necessário encontrar o número mínimo de combinações de colunas que somadas em modulo-2 resultam no vetor nulo.

Por exemplo, conhecendo a matriz \mathbf{H} do código de Hamming(7,4) (Equação 3.31), verifica-se que $d_{min} = 3$, pois não há colunas repetidas em \mathbf{H} e a soma módulo-2 da 1ª, 5ª e 6ª colunas dá origem ao vetor nulo, como mostrado a seguir:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow d_{min} = 3. \quad (3.34)$$

3.3 DISTRIBUIÇÃO DE PESOS

Um fator muito importante no desempenho de um código de bloco é a sua distribuição de pesos das palavras codificadas, ou seja, a listagem de todas as 2^k palavras código, relacionando o número de palavras que possuem um certo peso de Hamming. Quanto menor o número de palavras código de peso pequeno, melhor é o desempenho do código quanto à sua capacidade garantida de correção de erros. Para exemplificar o comportamento da distribuição de pesos, utiliza-se a matriz geradora sistemática encontrada na Equação 3.23 para catalogar todas as suas palavras código (Tabela 3.1) e listar sua distribuição de pesos (Tabela 3.2).

Tabela 3.1: Lista das 2^4 palavras codificadas do código de Hamming(7,4)

mensagem (m)	código (c)	$w_H(\mathbf{c})$
0 0 0 0	0 0 0 0 0 0 0	0
0 0 0 1	1 0 1 0 0 0 1	3
0 0 1 0	1 1 1 0 0 1 0	4
0 0 1 1	0 1 0 0 0 1 1	3
0 1 0 0	0 1 1 0 1 0 0	3
0 1 0 1	1 1 0 0 1 0 1	4
0 1 1 0	1 0 0 0 1 1 0	3
0 1 1 1	0 0 1 0 1 1 1	4
1 0 0 0	1 1 0 1 0 0 0	3
1 0 0 1	0 1 1 1 0 0 1	4
1 0 1 0	0 0 1 1 0 1 0	3
1 0 1 1	1 0 0 1 0 1 1	4
1 1 0 0	1 0 1 1 1 0 0	4
1 1 0 1	0 0 0 1 1 0 1	3
1 1 1 0	0 1 0 1 1 1 0	4
1 1 1 1	1 1 1 1 1 1 1	7

Tabela 3.2: Distribuição de pesos do código de Hamming(7,4)

Peso de Hamming ($w_H(\mathbf{c})$)	Quantidade A_i de Palavras Código com Peso i
0	1
3	7
4	7
7	1

É conveniente representar-se a distribuição de pesos, exemplificada na Tabela 3.2, pelo polinômio enumerador

$$A(z) = A_0 + A_1z + A_2z^2 + \cdots + A_nz^n, \quad (3.35)$$

em que

$$A_i \geq 0, 0 \leq i \leq n, \quad (3.36)$$

representa o número de palavras código com peso i .

No exemplo das Tabelas 3.1 e 3.2, o polinômio enumerador do código de Hamming (7, 4) é dado por

$$A(z) = 1 + 7z^3 + 7z^4 + z^7. \quad (3.37)$$

A partir do polinômio $A(z) \in C$, pode-se encontrar o polinômio enumerador $B(z)$ do código dual C^\perp utilizando a Identidade de MacWilliams [2]:

$$B(z) = q^{-k} [1 + (q-1)z]^n A\left(\frac{1-z}{1+(q-1)z}\right), \quad (3.38)$$

em que q , como descrito na Seção 3.1, é a ordem do campo $GF(q)$.

3.4 ESFERAS DE HAMMING, CAPACIDADES DE DETECÇÃO E DE CORREÇÃO DE CÓDIGOS DE BLOCO LINEARES

As palavras de um código $C(n, k)$ podem ser representadas como determinadas posições em um espaço n -dimensional, em que a menor distância entre duas palavras é dada pela Equação 3.3. Considerando que as palavras codificadas são corrompidas por um canal binário simétrico (*binary symmetric channel* - BSC) com probabilidade cruzada de erro menor que 50%, os vetores recebidos são calculados pela relação

$$\mathbf{r} = \mathbf{c} \oplus \mathbf{e} \quad (3.39)$$

em que \mathbf{c} é a palavra código na entrada do canal e \mathbf{e} é o vetor de erro inserido pelo canal BSC.

Pode-se representar todas as possibilidades de palavras recebidas pelas chamadas esferas de Hamming, exemplificadas na Figura 3.3.

Cada círculo vermelho da Figura 3.3 representa uma palavra código e cada círculo azul caracteriza um vetor que não pertence a C . Considera-se que todas as esferas da Figura 3.3 possuem o mesmo raio t e a menor distância entre palavras código é d_{min} . Nesse caso, tem-se que

$$\text{Capacidade Garantida de Correção de Erros} = t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor, \quad (3.40)$$

em que a função $\lfloor (\cdot) \rfloor$ calcula o maior inteiro menor ou igual a (\cdot) . Pelo critério de detecção por máxima verossimilhança (*maximum likelihood* - ML) (Anexo I), cada vetor \mathbf{r} contido numa

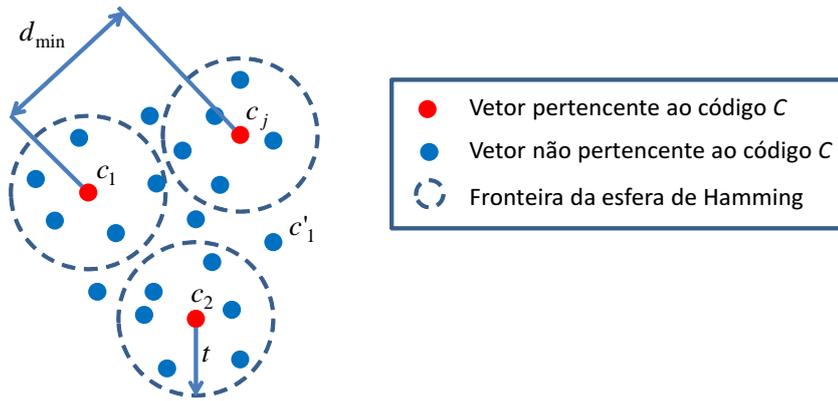


Figura 3.3: Ilustração geométrica das esferas de Hamming. Adaptado de [2]

certa esfera é estimado pela palavra código localizada na mesma esfera. Assim, caso a palavra recebida possua t ou mais erros, o decodificador não será capaz de garantir correção dos erros nem a estimação fiel da palavra código transmitida. Existem situações nas quais o vetor de erros e corrompe a palavra código \mathbf{c}_1 de tal modo que $\mathbf{r} = \mathbf{c}_2$. Nesse caso, o decodificador é incapaz de detectar o erro inserido pelo canal. Logo,

$$\text{Capacidade Garantida de Detecção de Erros} = d_{\min} - 1 \quad (3.41)$$

Outro caso importante a ser analisado ocorre quando a palavra estimada se encontra fora de todas as esferas de Hamming, como a $\mathbf{r} = \mathbf{c}'_1$, na Figura 3.3. Nesse caso, o decodificador declara uma falha de decodificação.

3.5 DECISÃO *HARD* e *SOFT*

Um detector de decisão *hard* amostra o sinal recebido e verifica se cada amostra se encontra abaixo ou acima do limiar de decisão. Ou seja, o detector recebe um sinal contínuo, realiza a amostragem e quantiza cada amostra em 0 ou 1, no caso binário, ou em q valores distintos, para um alfabeto não-binário. Após a decisão *hard*, o decodificador determina a palavra código estimada como sendo a palavra código que possui menor distância de Hamming do vetor decidido.

Já na detecção *soft*, a decisão é feita baseada nos critérios de grau de confiabilidade da informação ou com base em valores probabilísticos. Desse modo, a decisão *soft* utiliza a métrica da distância Euclidiana como base para determinação da informação decodificada. Esse tipo de decisão requer uma maior complexidade do decodificador, mas, como regra geral, a decodificação *soft* oferece um ganho de até 3 dB¹ [24] com relação à decodificação *hard*.

O Exemplo 3.3, adaptado de [8], ilustra uma situação em que é clara a superioridade em

¹Para valores elevados de E_b/N_0 . Considerando valores mais realistas, a decisão *soft* oferece ganhos da ordem de 2,5 dB frente à decodificação *hard*.

desempenho da decodificação *soft* frente à *hard*.

Exemplo 3.3 Considere um código de bloco com $n = 3$ e paridade par, ou seja, a soma módulo-2 dos bits das palavras código é igual a 0. O vetor $\mathbf{c} = [1\ 0\ 1]$ é modulado em $\mathbf{s} = [+1\ -1\ +1]$ e, em seguida, é transmitido por um canal gaussiano. O vetor \mathbf{r} amostrado na saída do canal é dado por $\mathbf{r} = [-0,1072\ -1,0504\ 1,6875]$. Sejam \mathbf{c}_1 , \mathbf{c}_2 , \mathbf{c}_3 e \mathbf{c}_4 as palavras código que são moduladas, respectivamente, em $\mathbf{s}_1 = [-1\ -1\ -1]$, $\mathbf{s}_2 = [-1\ +1\ +1]$, $\mathbf{s}_3 = [+1\ -1\ +1]$ e $\mathbf{s}_4 = [+1\ +1\ -1]$.

Considerando que o limiar de decisão é igual a 0 (fonte de dados equiprovável), a decisão *hard* estima a palavra recebida como $\tilde{\mathbf{c}} = [0\ 0\ 1]$ e, em seguida, calcula a distância de Hamming entre $\tilde{\mathbf{c}}$ e as palavras códigos, obtendo assim

$$\begin{aligned} d_H(\tilde{\mathbf{c}}, \mathbf{c}_1) &= d_H([0\ 0\ 1], [0\ 0\ 0]) = 1, \\ d_H(\tilde{\mathbf{c}}, \mathbf{c}_2) &= d_H([0\ 0\ 1], [0\ 1\ 1]) = 1, \\ d_H(\tilde{\mathbf{c}}, \mathbf{c}_3) &= d_H([0\ 0\ 1], [1\ 0\ 1]) = 1, \\ d_H(\tilde{\mathbf{c}}, \mathbf{c}_4) &= d_H([0\ 0\ 1], [1\ 1\ 0]) = 3. \end{aligned} \tag{3.42}$$

Assim, pelo critério de decisão *hard* e pela análise da distância de Hamming, o decodificador não consegue determinar com certeza qual das palavras código \mathbf{c}_1 , \mathbf{c}_2 ou \mathbf{c}_3 foi a palavra transmitida.

Já pelo critério de decisão *soft*, as seguintes distância Euclidianas são calculadas:

$$\begin{aligned} d(\mathbf{r}, \mathbf{s}_1) &= \sqrt{[(-0,1072) - (-1)]^2 + [(-1,0504) - (-1)]^2 + [(1,6875) - (+1)]^2} = 2,83, \\ d(\mathbf{r}, \mathbf{s}_2) &= \sqrt{[(-0,1072) - (-1)]^2 + [(-1,0504) - (+1)]^2 + [(1,6875) - (+1)]^2} = 2,34, \\ d(\mathbf{r}, \mathbf{s}_3) &= \sqrt{[(-0,1072) - (+1)]^2 + [(-1,0504) - (-1)]^2 + [(1,6875) - (+1)]^2} = 1,30 \text{ e} \\ d(\mathbf{r}, \mathbf{s}_4) &= \sqrt{[(-0,1072) - (+1)]^2 + [(-1,0504) - (+1)]^2 + [(1,6875) - (-1)]^2} = 3,56. \end{aligned} \tag{3.43}$$

A distância Euclidiana mínima encontrada determina o vetor modulado \mathbf{s}_3 como o mais provável de ter sido enviado e, conseqüentemente, determina com êxito que a palavra código transmitida foi $\tilde{\mathbf{c}}_{est} = \mathbf{c}_3 = [1\ 0\ 1]$.

O algoritmo do decodificador LDPC que será apresentado na Seção 4.5.3 é de decisão *soft*, pois a sua decisão é tomada com base nos argumentos das funções de probabilidade *a posteriori* dos bits 0 e 1. No entanto, este decodificador também pode ser entendido como decodificador de entrada *soft* - as entradas são as probabilidades posteriores do canal - e de saída *hard* - a saída do decodificador é uma sequência de 0s e 1s que não apresenta o grau de confiabilidade de cada bit.

3.6 PROBABILIDADE DE ERRO - DECISÃO *SOFT*

Para ilustrar o desempenho de um sistema com decodificação *soft*, faz-se necessário optar por um esquema de modulação e por um modelo de canal. Consideram-se as seguintes condições de contorno que são adotadas para todo este trabalho:

- Codificação
Código $C(n, k)$, taxa $R = k/n$, energia de bit E_b e energia de bit codificado $E_c = RE_b$;
- Modulação BPSK
Os bits $\{0, 1\}$ são modulados, respectivamente, pelos símbolos $\{-\sqrt{E_c}, \sqrt{E_c}\}$;
- Canal
AWGN com variância σ^2 .

Supondo fonte de dados equiprovável, decisão *soft* e o cenário descrito acima, limita-se superiormente a probabilidade de erro de bit por [2]

$$P_b \leq \sum_{j=d_{min}}^n \left[A_j Q \left(\frac{d_E(\mathbf{c}_j, \mathbf{0})}{2\sigma} \right) \right], \quad (3.44)$$

em que

- $Q(\cdot)$ é a função definida por

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt; \quad (3.45)$$

- A_j é o coeficiente do polinômio enumerador de peso, descrito na Seção 3.3.

Substituindo a Equação 3.10 na Equação 3.44 e recordando que $E_c = RE_b$ e que $\sigma^2 = N_0/2$, tem-se que

$$\begin{aligned} P_b &\leq \sum_{j=d_{min}}^n \left[A_j Q \left(\frac{2\sqrt{E_c}j}{2\sqrt{N_0/2}} \right) \right], \\ P_b &\leq \sum_{j=d_{min}}^n \left[A_j Q \left(\sqrt{\frac{2jRE_b}{N_0}} \right) \right]. \end{aligned} \quad (3.46)$$

Para altos valores de E_b/N_0 , aproxima-se a Equação 3.46 por

$$P_b \approx A_{d_{min}} Q \left(\sqrt{\frac{2d_{min}RE_b}{N_0}} \right). \quad (3.47)$$

Na prática é muito trabalhoso o cálculo analítico da probabilidade de erro de bit pelas Equações 3.46 e 3.47. O primeiro motivo é o fato de que nem sempre se sabe com exatidão a distância mínima do código utilizado. A segunda dificuldade é imposta pelo coeficiente A_j , já que o seu valor exige o conhecimento dos pesos das 2^k palavras código, o que é inviável para códigos de tamanho de bloco maior que 1000 bits.

Dessa forma, será mostrado no Capítulo 4 que, em geral, a probabilidade de erro *soft* é calculada por simulação. No entanto, a Equação 3.47 será utilizada para justificar uma característica típica das curvas de taxa de erros de bit de códigos LDPC, que é o piso de erro, a ser descrito também no Capítulo 4.

Capítulo 4

CÓDIGOS LDPC

Neste capítulo introduz-se o conceito dos códigos LDPC, descrevem-se as principais características de sua matriz de teste de paridade e da sua representação em grafos. Citam-se brevemente as técnicas mais comuns de codificação e apresenta-se o principal algoritmo de decodificação soft utilizado em códigos LDPC. Por fim, abordam-se o problema do piso de erro e as suas formas de mitigação.

4.1 INTRODUÇÃO AO LDPC

Os códigos de Gallager, mais conhecidos como LDPC, foram propostos inicialmente em 1962 por Robert Gallager [25]. Esses códigos fazem parte dos códigos lineares de bloco e as suas matrizes de teste de paridade possuem pequena quantidade de valores não nulos, justificando a alcunha de código de matriz de teste de paridade esparsa (*Low-Density Parity-Check*).

Desde a sua criação até os anos 1990, essa classe de códigos ficou no anonimato devido a duas razões: as características de *hardware* da época não eram suficientes para a implementação do decodificador LDPC e a grande fama dos códigos concatenados utilizando Reed-Solomon e BCH camuflou a descoberta do LDPC.

Recentemente, com o trabalho de Mackay [21], os códigos de Gallager começaram a se popularizar, de maneira que hoje eles são utilizados em sistemas como Wi-Fi (802.11n), WiMAX (802.16e) e DVB-S2 [26]. Uma das razões da grande divulgação do LDPC consiste na popularização dos códigos iterativos, como código Turbo e o próprio LDPC. Outra vantagem é o seu potencial para atingir maiores ganhos de código do que os alcançados pelos próprios códigos Turbo, que até a década de 90 despontavam como os de maior eficiência quanto à razão sinal-ruído requerida para uma dada taxa de erro de bit.

4.2 MATRIZ ESPARSA DE TESTE DE PARIDADE

A matriz de teste de paridade pode ser regular ou irregular. No primeiro caso, todas as linhas de \mathbf{H} possuem pesos de Hamming constantes e iguais a w_r e todas as colunas têm pesos de Hamming fixos iguais a w_c , dando origem a seguinte notação de código: LDPC(w_c, w_r, n) [2]. Por outro lado, códigos irregulares possuem os pesos w_r e w_c variáveis ao longo da matriz de teste de paridade. Para não se fazer distinção entre códigos de Gallager regulares e irregulares, este trabalho utiliza a notação de códigos empregada no Capítulo 3: LDPC(n, k).

Como regra geral, os códigos LDPC irregulares possuem desempenho de taxa de erro de bit ligeiramente superior aos códigos regulares [8]. Em [27], os códigos irregulares atingem ganhos de até 0,5 dB frente aos códigos regulares.

A maneira mais comum de se construir a matriz regular de teste de paridade é feita definindo-se w_c , w_r , k e n , em que o peso das colunas/linhas é muito menor do que o tamanho do código. Caso todas as $n - k$ linhas sejam linearmente independentes, a taxa de código é dada por $R = 1 - w_c/w_r$. Caso contrário, eliminam-se as linhas linearmente dependentes e a taxa de código efetiva torna-se $R' = 1 - w'_c/w_r$, em que w'_c representa o peso das colunas da nova matriz \mathbf{H}' .

4.3 CODIFICAÇÃO

Uma vez que o código LDPC foi definido, tem-se conhecimento de sua matriz de teste de paridade (Equação 3.27). Para se encontrar a matriz geradora (Equação 3.14) a partir de \mathbf{H} , existem algumas maneiras como as citadas a seguir.

Uma das técnicas de codificação (válida para todos os códigos de bloco) mais comum é a sistematização de \mathbf{H} por eliminação de Gauss-Jordan [5]. Transforma-se a matriz de teste de paridade na forma descrita pela Equação 3.33 e, posteriormente, encontra-se a matriz \mathbf{G}_{sis} (Equação 3.19), como ilustrado no Exemplo 3.2. Feito isso, a codificação, em si, é simplesmente feita pelo processo de codificação sistemática (Equação 3.20).

Outro modo bem difundido de codificação é o algoritmo proposto por Richardson [3], desenvolvido especialmente para codificação eficiente LDPC. Antes de se iniciar a codificação, realiza-se um pré-processamento da matriz de teste de paridade original, de modo a transformá-la no formato indicado pela Figura 4.1. Esse pré-processamento é feito por meio de várias permutações de linhas e colunas da matriz \mathbf{H} , até que em sua região direita superior seja identificada a matriz triangular inferior \mathbf{MT} , como mostrado na Figura 4.1.

Em seguida obtém-se o vetor codificado \mathbf{c} por

$$\mathbf{c} = [\mathbf{m} \ \mathbf{p}_1 \ \mathbf{p}_2], \quad (4.1)$$

em que \mathbf{p}_1 e \mathbf{p}_2 são encontrados resolvendo-se um sistema linear de duas equações envolvendo as matrizes \mathbf{MA} , \mathbf{MB} , \mathbf{MC} , \mathbf{MD} , \mathbf{ME} e \mathbf{MT} , descritas na Figura 4.1.

O primeiro método apresenta complexidade $O(n^2)$ e o segundo $O(n + gap^2)$ [2], em que n é o

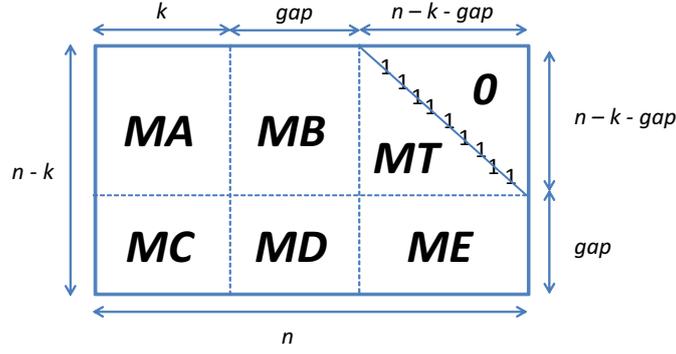


Figura 4.1: Matriz de teste de paridade no formato triangular inferior aproximado. Adaptado de [3].

tamanho da palavra código e gap é um parâmetro do algoritmo de [3], em que $gap \leq n \times R$, sendo que R é a taxa do código.

Como o LDPC também é um código de bloco linear, o desempenho de taxa de erro de bit é o mesmo para todas palavras código. Sabendo que a palavra nula é palavra código de todos códigos de bloco lineares e que as distribuições de probabilidades $p(c_j = 0|r_j)$ e $p(c_j = 1|r_j)$ são ambas gaussianas com variâncias iguais (modulação BPSK), pode-se verificar o desempenho de BER dos códigos LDPC considerando a transmissão de palavras mensagem nulas de k bits e de palavras código nulas de n bits, sem perda de generalização [2, 19].

4.4 GRAFO DE TANNER E *GIRTH*

O grafo de Tanner é um grafo bipartite que descreve a matriz de paridade do código [2]. Dado um código (n, k) , as n colunas da matriz \mathbf{H} representam os nós de variáveis nc_j , para $1 \leq j \leq n$, e as $n - k$ linhas representam os nós de verificação nh_i , para $1 \leq i \leq n - k$. Assim, cada “1” da matriz \mathbf{H} numa certa posição (i, j) representa uma aresta do grafo de Tanner ligando o nó de variável nc_j ao nó de verificação nh_i . Como exemplo, conhecendo-se a matriz de teste de paridade dada por

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad (4.2)$$

representa-se o seu grafo de Tanner correspondente na Figura 4.2. No grafo em questão, os bits de paridade (nós de verificação) são representados pelos nós nh_1 a nh_4 e os bits de palavra código (nós de variável) pelos nós nc_1 a nc_6 .

Uma propriedade muito importante do grafo é o *girth* (g), que representa o número de arestas do ciclo mínimo do grafo [2]. O *girth* geralmente está diretamente relacionado com o tamanho

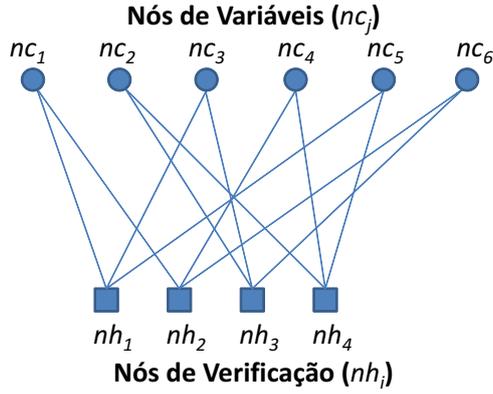


Figura 4.2: Grafo de Tanner correspondente à matriz \mathbf{H} da Equação 4.2.

da palavra código [26]. Como uma estimativa grosseira [28], códigos com g iguais a 6, 8 e 10 requerem, respectivamente, n da ordem de 460, 4000 e 20000 bits. A Figura 4.3 exemplifica um grafo de Tanner com $girth$ igual a 4.

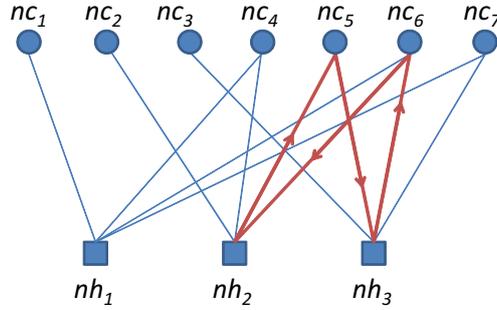


Figura 4.3: Grafo de Tanner de um código de Hamming(7,4). $Girth$ igual a 4 em destaque.

4.5 DECODIFICAÇÃO: ALGORITMO SOMA-PRODUTO

4.5.1 Funções de verossimilhança

Como descrito na Seção 3.6, para modulação BPSK com codificação, o conjunto de bits $\{0, 1\}$ corresponde à constelação $S = \{-\sqrt{E_c}, +\sqrt{E_c}\}$. Considerando a transmissão por canal AWGN, as funções de verossimilhança dos bits 0 e 1 são, respectivamente, dadas pelas funções de densidade de probabilidade (*probability density function* - PDF) [2]

$$p(r|s = -\sqrt{E_c}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(r+\sqrt{E_c})^2} e \quad (4.3)$$

$$p(r|s = \sqrt{E_c}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(r-\sqrt{E_c})^2}. \quad (4.4)$$

No caso do algoritmo Soma-Produto Simplificado, descrito na Subseção 4.5.3, a informação do canal é fornecida para o decodificador LDPC na forma das probabilidades posteriores do canal $P(c_j = 0|r_j)$ e $P(c_j = 1|r_j)$. A partir do Teorema de Bayes [2], verifica-se que

$$P(c_j = 1|r_j) = \frac{p(r_j|s_j = \sqrt{E_c})P(s_j = \sqrt{E_c})}{p(r_j)}. \quad (4.5)$$

Sabendo que

$$p(r_j) = p(r_j|s_j = \sqrt{E_c})P(s_j = \sqrt{E_c}) + p(r_j|s_j = -\sqrt{E_c})P(s_j = -\sqrt{E_c}), \quad (4.6)$$

e supondo fonte de dados equiprovável ($P(s_j = \sqrt{E_c}) = P(s_j = -\sqrt{E_c}) = 0,5$), a Equação 4.5 pode ser reescrita como

$$P(c_j = 1|r_j) = \frac{p(r_j|s_j = \sqrt{E_c})P(s_j = \sqrt{E_c})}{p(r_j|s_j = \sqrt{E_c})P(s_j = \sqrt{E_c}) + p(r_j|s_j = -\sqrt{E_c})P(s_j = -\sqrt{E_c})}, \quad (4.7)$$

$$P(c_j = 1|r_j) = \frac{p(r_j|s_j = \sqrt{E_c})}{p(r_j|s_j = \sqrt{E_c}) + p(r_j|s_j = -\sqrt{E_c})}. \quad (4.8)$$

Substituindo as Equações 4.3 e 4.4 em 4.8, tem-se que

$$\begin{aligned} P(c_j = 1|r_j) &= \frac{1}{1 + e^{-2E_c r_j / \sigma^2}}, \\ P(c_j = 0|r_j) &= 1 - P(c_j = 1|r_j), \end{aligned} \quad (4.9)$$

definindo assim (para modulação BPSK) as probabilidades posteriores do canal para cada amostra recebida r_j . A partir da premissa de que as amostras recebidas são independentes, podem-se definir as probabilidades posteriores do vetor de amostras \mathbf{r}_k (k -ésimo bloco de n amostras recebido) por

$$\begin{aligned} P(c_j = 1|\mathbf{r}_k) &= \frac{1}{1 + e^{-2E_c \mathbf{r}_k / \sigma^2}}, \\ P(c_j = 0|\mathbf{r}_k) &= 1 - P(c_j = 1|\mathbf{r}_k). \end{aligned} \quad (4.10)$$

4.5.2 Algoritmo de decodificação: Soma-Produto Original

De conhecimento da matriz \mathbf{H} e das probabilidades posteriores dadas pela Equação 4.9, realiza-se a etapa de decodificação com base na estrutura do grafo de Tanner. O objetivo do algoritmo é trocar informações entre bits da palavra código e os bits de paridade de forma a acumular evidências que apontem qual é a palavra código estimada $\tilde{\mathbf{c}}_k$ mais provável de ter sido transmitida. Ou seja, estima-se $\tilde{\mathbf{c}}_k$ que maximiza a função $P(\mathbf{c}_k|\mathbf{r}_k)$.

Para facilitar o entendimento do algoritmo, necessitam-se definir alguns termos relacionados com o grafo de Tanner. Todos os nós de verificação conectados a um determinado nó de variável c_j são chamados de nós filhotes de c_j . Todos os nós de variável conectados a um determinado nó de verificação h_i são chamados de nós progenitores de h_i . As Figuras 4.4 e 4.5 exemplificam a ocorrência de nós filhotes e progenitores num certo grafo.

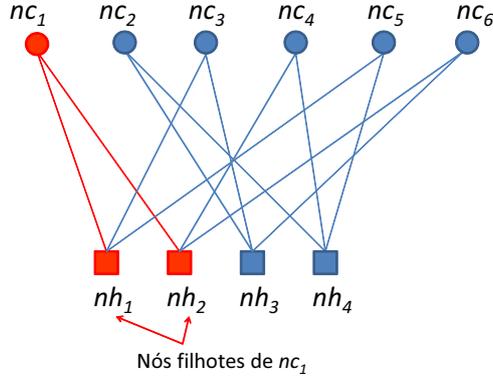


Figura 4.4: Grafo de Tanner no qual os nós de verificação nh_1 e nh_2 são nós filhotes de nc_1 .

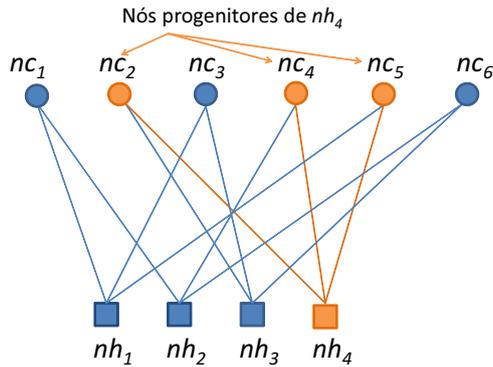


Figura 4.5: Grafo de Tanner no qual os nós de variável nc_2 , nc_4 e nc_5 são nós progenitores de nh_4 .

A inicialização do algoritmo consiste na atribuição das probabilidades posteriores $P(c_j = 0|\mathbf{r}_k)$ e $P(c_j = 1|\mathbf{r}_k)$ como informação inicial para as probabilidades de que cada nó de variável nc_j seja igual a 0 ou 1, respectivamente.

Após a inicialização, começa a troca de informações entre os nós de verificação e os nós de variável. Cada nó de verificação nh_i envia para cada um de seus nós progenitores nc_j a estimativa R_{ij}^x de que a i -ésima equação de paridade da matriz \mathbf{H} seja satisfeita para $c_j = x$, em que $x \in \{0, 1\}$.

Em seguida, cada nó de variável nc_j envia para cada um de seus nós filhotes nh_i a probabilidade Q_{ij}^x de que $h_i = x$ baseado nas informações $(R_{qj}^x, q \neq i)$ fornecidas pelos demais nós filhotes conectados a nc_j .

O processo de troca de informações entre nós do grafo de Tanner é ilustrado nas Figuras 4.6, 4.7, e 4.8.

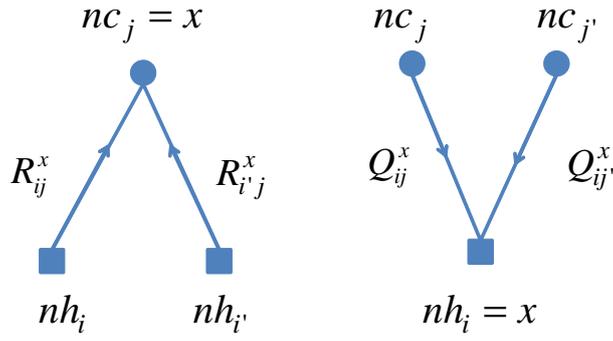


Figura 4.6: Troca de informações na decodificação LDPC.

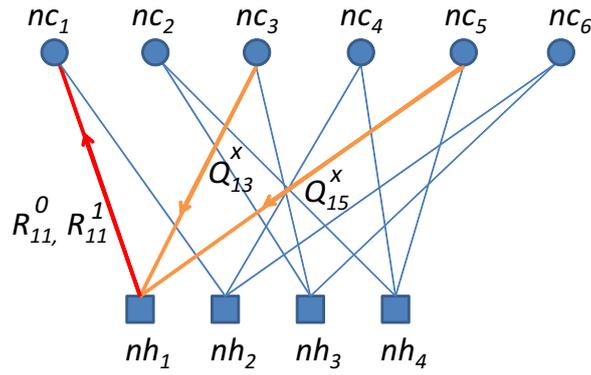


Figura 4.7: Cálculo dos coeficientes R_{11}^0 e R_{11}^1 .

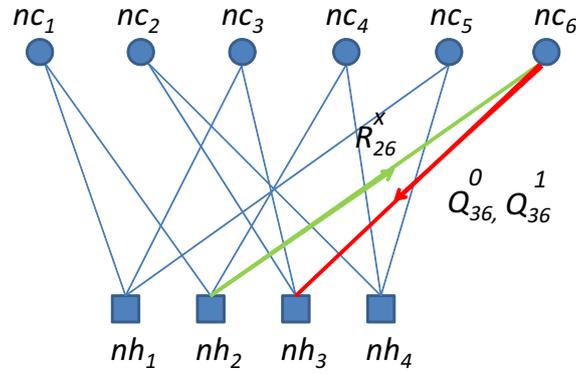


Figura 4.8: Cálculo dos coeficientes Q_{36}^0 e Q_{36}^1 .

Após realizadas as trocas de informações envolvendo todos os nós do grafo, cada bit \tilde{c}_j estimado é dado por

$$\tilde{c}_j = \arg \max_x \left(l_j^x \prod_{q \in M(j)} R_{qj}^x \right), \quad (4.11)$$

em que $M(j)$ representa a lista de índices de todos os nós de verificação conectados ao nó de variável nc_j .

Caso a condição $\tilde{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$ seja satisfeita, o algoritmo chega ao fim. Caso contrário, novas trocas de mensagem entre os nós de verificação e de variável são realizadas até que a condição citada seja validada ou até que o algoritmo atinja o número máximo de iterações estipuladas pelo projetista.

4.5.3 Algoritmo de decodificação: Soma-Produto Simplificado

De acordo com o algoritmo da Subseção 4.5.2, supondo que a matriz \mathbf{H} de um código LDPC regular possui peso de linha w_r , o cálculo de cada informação R_{ij}^x requer a análise de 2^{w_r-2} [8] combinações de palavras código que satisfazem a i -ésima equação de paridade para $c_j = x$. Já os coeficientes Q_{ij}^x dependem, por sua vez, dos valores de R_{ij}^x .

No entanto, Mackay e Neal introduziram em [29] o algoritmo Soma-Produto Simplificado. A vantagem desta proposta é que esse algoritmo não leva em consideração todas as combinações de palavras código que validam as equações de paridade no cálculo de R_{ij}^x . Desse modo, essa simplificação resulta em diminuição da complexidade do decodificador com uma baixíssima penalidade em termos de desempenho de taxa de erro de bit obtida.

As probabilidades *a priori* f_j^0 e f_j^1 ² são definidas por

$$\begin{aligned} f_j^1 &= P(c_j = 1|\mathbf{r}), \\ f_j^0 &= P(c_j = 0|\mathbf{r}), \end{aligned} \quad (4.12)$$

em que $P(c_j = 1|\mathbf{r})$ e $P(c_j = 0|\mathbf{r})$ são definidas na Equação 4.10.

A primeira etapa do algoritmo é a *Inicialização*, em que se calcula Q_{ij}^0 e Q_{ij}^1 por

$$\begin{aligned} Q_{ij}^0 &= f_j^0, \\ Q_{ij}^1 &= f_j^1. \end{aligned} \quad (4.13)$$

A seguir inicia-se o *Passo Horizontal*. Primeiro calcula-se δQ_{ij} da seguinte maneira:

$$\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1. \quad (4.14)$$

Depois, encontra-se δR_{ij} por

$$\delta R_{ij} = \prod_{j' \in N(i) \setminus j} \delta Q_{ij'}, \quad (4.15)$$

em que $N(i) \setminus j$ representa a lista de todos os nós progenitores de nh_i , a menos do nó progenitor nc_j . Seguindo com o algoritmo, calcula-se agora R_{ij}^0 e R_{ij}^1 , em que

$$\begin{aligned} R_{ij}^0 &= (1/2)(1 + \delta R_{ij}), \\ R_{ij}^1 &= (1/2)(1 - \delta R_{ij}). \end{aligned} \quad (4.16)$$

²Sob a ótica da teoria da informação, a probabilidade $P(c_j = x|r_j)$ é definida como probabilidade *a posteriori*. No entanto, seguindo a notação de [8], a probabilidade f_j^x é chamada de *a priori* no sentido de que a informação do código LDPC - contida no grafo de Tanner - ainda não foi utilizada para se encontrar a estimativa *a posteriori* dos bits recebidos.

Em seguida os coeficientes Q_{ij}^0 e Q_{ij}^1 são atualizados no *Passo Vertical*. Logo, para os nós (i, j) calcula-se

$$\begin{aligned} Q_{ij}^0 &= \alpha_{ij} f_j^0 \prod_{i' \in M(j) \setminus i} R_{i'j}^0 \text{ e} \\ Q_{ij}^1 &= \alpha_{ij} f_j^1 \prod_{i' \in M(j) \setminus i} R_{i'j}^1 \end{aligned} \quad (4.17)$$

em que:

- $M(j) \setminus i$ representa a lista $M(j)$ com exceção do nó de verificação nh_i ;
- α_{ij} é tal que $Q_{ij}^0 + Q_{ij}^1 = 1$.

Para realizar a decisão do vetor decodificado para a iteração atual é necessária a análise das probabilidades *a posteriori* [8] Q_j^0 e Q_j^1 dadas por

$$\begin{aligned} Q_j^0 &= \alpha_j f_j^0 \prod_{i \in M(j)} R_{ij}^0 \text{ e} \\ Q_j^1 &= \alpha_j f_j^1 \prod_{i \in M(j)} R_{ij}^1, \end{aligned} \quad (4.18)$$

sendo que α_j é escolhido de forma que $Q_j^0 + Q_j^1 = 1$.

Por fim, realiza-se a tentativa de decisão: se $Q_j^0 > Q_j^1$, então $\tilde{c}_j = 0$. Caso contrário, $\tilde{c}_j = 1$. Após se estimar os n bits da palavra código, verifica-se se $\tilde{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$. Em caso afirmativo, termina-se aqui a decodificação. Caso contrário, analisa-se o número de iterações feitas até o momento. Se o número atual de iterações for menor do que o número máximo configurado para o decodificador (N_{iter}), o decodificador dá prosseguimento para a iteração seguinte. Caso contrário, declara-se falha de decodificação e considera-se a última palavra decodificada como a estimada pela última iteração. Ou seja, a regra de decisão resume-se a

$$\tilde{c}_j = \arg \max_x (Q_j^x). \quad (4.19)$$

4.6 PISO DE ERRO

Em códigos LDPC, assim como em códigos Turbo, observa-se que a curva de taxa de erro de bit por E_b/N_0 possui duas regiões bem caracterizadas. Inicialmente, a curva de taxa de erro de bit apresenta declive muito acentuado, caracterizando assim a região de queda brusca, mais conhecida como *waterfall region*. Depois, a partir de um certo valor de E_b/N_0 , a BER diminui a uma taxa de decaimento menor, originando assim a região de piso de erro (*error floor region*). Ambas as regiões são visualizadas na Figura 4.9.

A existência do piso de erro é justificada com base na Equação 3.46. Conforme E_b/N_0 aumenta, o primeiro termo do somatório da Equação 3.46 torna-se o termo predominante. Assim, o desempenho assintótico da probabilidade de erro é aproximado pela Equação 3.47, que é significativamente influenciada pela distância mínima do código. Dessa maneira, o piso de erro é consequência do projeto de códigos com distância mínima pequena.

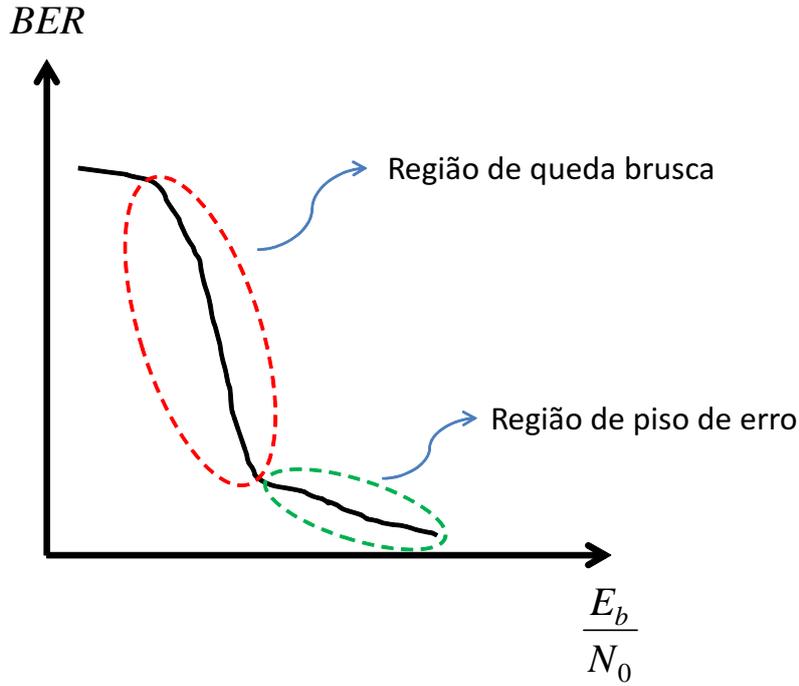


Figura 4.9: Comportamento típico de códigos LDPC com piso de erro. Adaptado de [2].

Para códigos LDPC regulares, caracterizados pela matriz \mathbf{H} com peso das colunas w_c e *girth* g , a distância mínima é limitada inferiormente por [5]

$$d_{min} \geq 1 + \frac{w_c}{w_c - 2} \left[(w_c - 1)^{\lfloor (g-2)/4 \rfloor} - 1 \right], \text{ para } g/2 \text{ ímpar,}$$

$$d_{min} \geq 1 + \frac{w_c}{w_c - 2} \left[(w_c - 1)^{\lfloor (g-2)/4 \rfloor} - 1 \right] + (w_c - 1)^{\lfloor (g-2)/4 \rfloor}, \text{ para } g/2 \text{ par.} \quad (4.20)$$

Logo, uma das formas de se mitigar o piso de erro em códigos LDPC se dá pela construção de códigos com *girth* elevado [5]. Conseqüentemente, o aumento do *girth* requer utilização de códigos com tamanhos de bloco elevados, o que aumenta a latência do decodificador.

Outra forma de se lidar com esse problema é a utilização de outro código concatenado com o LDPC, como, por exemplo, o Reed-Solomon [30]. Com essa configuração, o decodificador Reed-Solomon (externo) consegue corrigir os erros remanescentes do decodificador LDPC (interno). O custo dessa solução está no aumento da complexidade do sistema devido a inserção de um codificador e um decodificador adicionais.

Capítulo 5

CONSUMO DE POTÊNCIA EM FPGAS

Neste capítulo descrevem-se, de maneira breve, as etapas envolvidas na implementação de um projeto em FPGA e na estimação por software da potência consumida pela implementação. Apresenta-se o modelo matemático da potência total gasta pelo FPGA e insere-se o conceito de eficiência energética de decodificadores

5.1 INTRODUÇÃO FPGA

Uma das maneiras de se estimar a potência de um circuito digital consiste na implementação do projeto em *hardware* reconfigurável, de modo especial em um arranjo de portas programáveis em campo (*field programmable gate arrays* - FPGA) ou em um circuito integrado de aplicação específica (*application-specific integrated circuit* - ASIC). A vantagem de se utilizar FPGA está na maior flexibilidade de programação e de reconfiguração do circuito, o que requer dos FPGAs maior número de transistores de efeito de campo (*field-effect transistor* - FET) e maior consumo de potência [31] quando comparadas com ASICs.

Realiza-se a estimação do consumo de potência de projetos voltados para FPGA em uma das três fases distintas do projeto [4]:

- Fase de Concepção ou Pré-Implementação

Calcula-se uma estimativa imprecisa do consumo de potência baseada em informações fornecidas para planilhas de dados interativas, como a ferramenta Power Estimator [32]. Utilizam-se como dados de entrada informações de recursos de *hardware* utilizados, taxa de variação dos sinais, condições térmicas do ambiente, dentre outros.

- Fase de Implementação

Calcula-se a potência com base nas informações fornecidas pelo projeto baseado em descrição em linguagem de *hardware*, pronto para ser instalado em FPGA. Com auxílio de um arquivo

que fornece os sinais de entrada do projeto (arquivo de *test bench*), utiliza-se um *software* capaz de interpretar esses dados e fazer a estimaco do consumo de potncia. A ferramenta utilizada neste trabalho   a XPower Analyzer [33].

- Fase de Integrao do Sistema

Ap s a configurao do FPGA, mede-se em laborat rio o consumo de potncia dispendido pelo projeto.

5.2 ETAPAS DE PROJETO

Neste trabalho a potncia ser  estimada na fase de implementao do projeto. Para se concretizar tal meta, so necess rios os seguintes passos [34, 35]:

- Projeto em VHDL

Inicialmente   necess ria a descrio da funo ou do algoritmo que se deseja implementar em linguagem de *hardware*. No caso deste trabalho, a descrio   convertida da linguagem do Matlab para um c digo VHDL. Em seguida, determina-se a fam lia do FPGA e o modelo de dispositivo que se deseja implementar o projeto.

- S ntese L gica

Esta etapa   respons vel por converter a descrio VHDL em uma implementao de circuito digital que utilize portas l gicas. Nessa etapa definem-se a quantidade e o modelo de elementos l gicos requeridos pelo circuito, como multiplexadores, somadores, registradores, comparadores, multiplicadores, entre outros elementos.

- *Place and Route*

Na etapa de *place* decide-se onde devem se localizar todos os componentes eletr nicos em uma certa  rea limitada. J  na etapa de *route* decide-se a arquitetura exata do dispositivo e realiza-se o roteamento das conex es e dos sinais do circuito.

5.3 MODELO DE CONSUMO DE FPGAS

De modo geral, a estimaco de potncia de circuitos digitais depende de uma s rie de fatores, como tecnologia CMOS, arquitetura do projeto, frequ ncia do rel gio, transio de sinais, temperatura ambiente, dentre outros. No entanto, um modelo simplificado bastante utilizado para se estimar a potncia total consumida   descrito por [4, 36]

$$P_{Total} = P_{Est} + P_{Din}, \quad (5.1)$$

em que as potncias est tica (P_{est}) e din mica (P_{din}) so descritas abaixo:

- Potência Estática - Também conhecida como potência quiescente, representa a potência consumida pela polarização CC (corrente contínua) e pelas correntes de fuga. Considerando as fontes de alimentação do FPGA que fornecem maior parte da potência quiescente do dispositivo, pode-se modelar a potência estática por [4]

$$P_{Est} = V_{CCINT}I_{CCINTQ} + V_{CCAUX}I_{CCAUXQ} + V_{CCO}I_{CCOQ}, \quad (5.2)$$

em que [37]

- V_{CCINT} é a tensão de alimentação CC interna;
- V_{CCAUX} é a tensão de alimentação CC auxiliar;
- V_{CCO} é a tensão de alimentação CC das portas de saída;
- I_{CCINTQ} é a corrente quiescente da fonte de alimentação V_{CCINT} ;
- I_{CCAUXQ} é a corrente quiescente da fonte de alimentação V_{CCAUX} ;
- I_{CCOQ} é a corrente quiescente da fonte de alimentação V_{CCO} .

Cada uma das correntes I_{CCINTQ} , I_{CCAUXQ} e I_{CCOQ} possui grande influência das duas fontes de correntes de fuga do transistor: da corrente entre fonte (*source*) e dreno (*drain*) - I_{SD} - e da corrente entre porta (*Gate*) e substrato - I_G . A Figura 5.1 mostra os sentidos das correntes de fuga num transistor.

Quanto menor a tecnologia de fabricação do transistor (*e.g.*, 45 nm), menores serão o comprimento de canal transistor e a espessura da porta de óxido, acentuando assim o efeito das correntes de fuga e do consumo de potência estática.

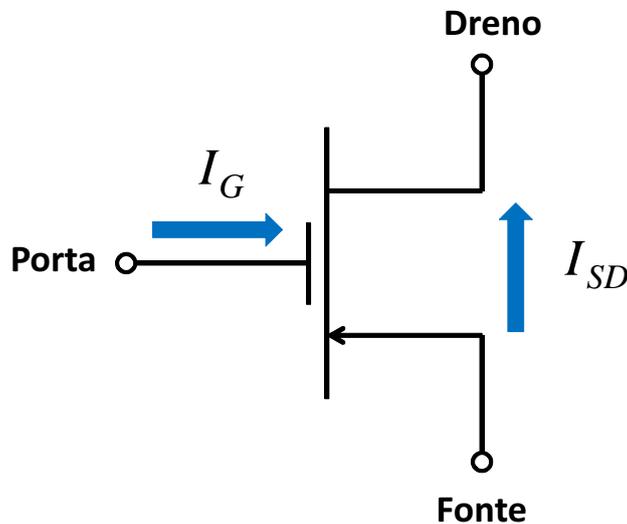


Figura 5.1: Correntes de fuga num FET. Adaptado de [4].

- Potência Dinâmica - Está relacionada com o consumo adicional de potência ligado ao processamento lógico e chaveamento de sinais. Pode ser modelada por [38,39]

$$P_{Din} = C_{total}f_c (V_{CCINT} + V_{CCAUX} + V_{CCO})^2, \quad (5.3)$$

em que C_{total} é a capacitância total de chaveamento e f_c é a frequência de operação do relógio.

Desse modo, pelas Equações 5.1, 5.2 e 5.3, o consumo total de potência em FPGA pode ser modelado por

$$P_{total} = V_{CCINT}I_{CCINTQ} + V_{CCAUX}I_{CCAUXQ} + V_{CCO}I_{CCOQ} + C_{total}f_c(V_{CCINT} + V_{CCAUX} + V_{CCO})^2. \quad (5.4)$$

Um parâmetro importante na avaliação da eficiência energética de um decodificador é a energia por bit decodificado Eb_{dec} , definida por [40]

$$Eb_{dec} = \frac{P_{total}}{T}, \quad (5.5)$$

em que P_{total} é a potência total consumida pelo decodificador e T é a vazão de dados de saída. A unidade típica de avaliação de Eb_{dec} é nJ/bit. No caso de decodificadores iterativos, como o LDPC, uma adaptação da Equação 5.5 é feita, de modo de que se quantifique a energia consumida por bit decodificado por iteração ($Eb_{dec/iter}$), ou seja

$$Eb_{dec/iter} = \frac{P_{total/iter}}{T}, \quad (5.6)$$

em que a potência de decodificação por iteração é igual a $P_{Total/iter} = P_{Total}/N_{iter}$ e N_{iter} é o número de iterações realizadas pelo decodificador

Capítulo 6

DESENVOLVIMENTO E RESULTADOS

6.1 DESCRIÇÃO GERAL

Como definido anteriormente, o objetivo deste trabalho é o estudo da relação entre potência de decodificação e taxa de erro de bit, em especial para códigos LDPC aplicáveis a sistemas ópticos, ou seja, códigos LDPC de altas taxas ($R \geq 0,8$) e com tamanhos de bloco elevados (palavras código maiores que 1000 bits).

Considerando canal AWGN e fixando-se o esquema de modulação, analisam-se as curvas de BER por E_b/N_0 e verifica-se qual dos códigos simulados possui maior CG . Para o código de melhor desempenho de taxa de erro de bit, analisa-se em seguida seu comportamento de desempenho para diferentes quantidades de iterações feitas pelo decodificador.

A ideia inicial desta seção era a estimação de potência de decodificação para códigos LDPC de quaisquer tamanhos. No entanto, devido a algumas razões descritas posteriormente, a estimação de potência do projeto completamente elaborado (neste trabalho) em VHDL restringiu-se a códigos de tamanho de bloco pequeno. Para o código que apresenta melhor desempenho de BER, realizam-se algumas aproximações considerando os modelos matemáticos de potência de FPGAs, os valores de potência de um decodificador LDPC implementado para $n > 1000$ em [41] e o comportamento da curva de BER pelo número de iterações.

6.2 ESTIMAÇÃO DA BER

A Figura 6.1 mostra o diagrama de blocos do sistema de comunicação utilizado no ambiente de simulação do Matlab. Seguem abaixo algumas considerações importantes sobre cada etapa da Figura 6.1:

- Codificação

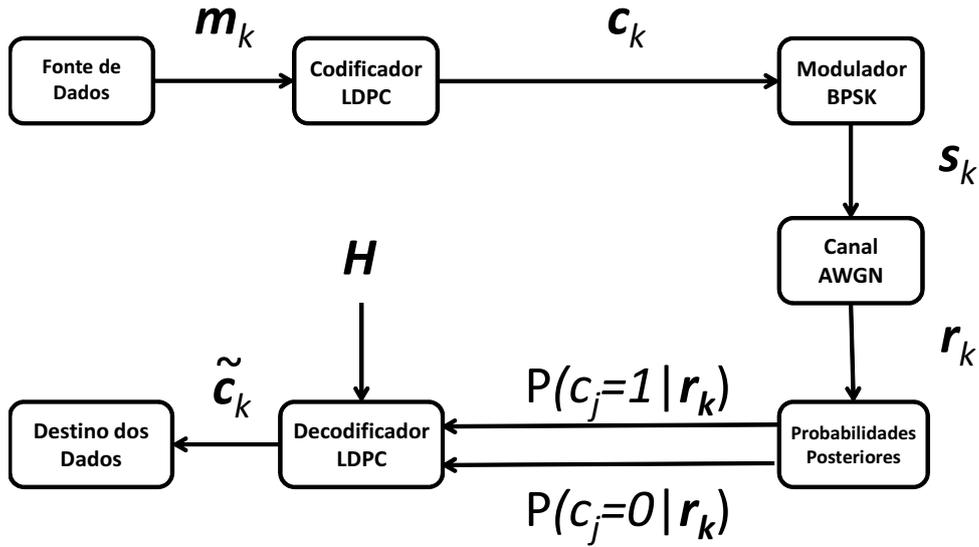


Figura 6.1: Sistema de comunicação simulado.

Como justificado na Seção 4.3, a etapa de codificação resume-se a adição de $n - k$ bits iguais a zero a cada palavra mensagem de zeros enviada.

- Modulação BPSK

Os bits 0 e 1 são modulados em $S = \{-\sqrt{E_c}, \sqrt{E_c}\}$, respectivamente, e E_c representa a energia de bit codificado. Dessa forma, descreve-se a relação entre o símbolo codificado c_k e o símbolo modulado s_k no instante de tempo k por

$$s_k = (2c_k - 1)\sqrt{E_c}. \quad (6.1)$$

- Canal AWGN

Descrito na Seção 2.2.

- Cálculo das Probabilidades Posteriores do Canal

Utilizam-se as probabilidades $P(c_j = 0|\mathbf{r}_k)$ e $P(c_j = 1|\mathbf{r}_k)$ definidas na Equação 4.10.

- Decodificação LDPC

De conhecimento da matriz \mathbf{H} e das probabilidades posteriores do canal, realiza-se então o algoritmo Soma-Produto Simplificado descrito na Subseção 4.5.3. Para manter constante a vazão de dados de saída do decodificador, o número de iterações do algoritmo é fixo.

Logo, nas simulações de decodificação deste trabalho, não se realiza o teste de verificação da palavra código estimada (Equação 3.29), diferentemente do que é feito na etapa final do algoritmo Soma-Produto Simplificado.

- Cálculo da BER

O cálculo é feito conforme descrito na Seção 2.3.

6.2.1 Resultados - desempenho de BER

A Figura 6.2 compara o desempenho de cinco códigos LDPC irregulares, listados na Tabela 6.1. A escolha dos códigos LDPC foi pautada na seleção de códigos de altas taxas e com tamanhos de bloco elevados. A primeira restrição está relacionada com o fato de que códigos com taxas mais baixas aumentariam consideravelmente a banda óptica requerida, e a segunda com o fato de que códigos maiores tendem a possuir *girth* elevados [28], o que melhora o desempenho de BER e mitiga o problema do piso de erro [5].

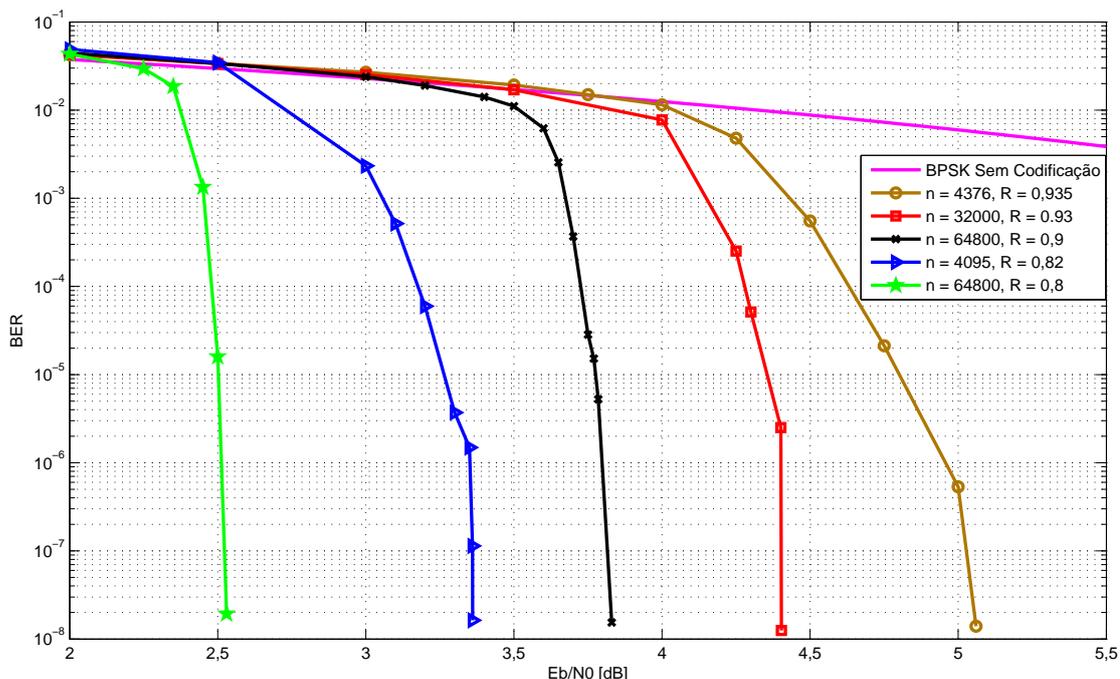


Figura 6.2: Desempenho de códigos LDPC de altas taxas para 50 iterações.

Tabela 6.1: Comparação de desempenho de códigos LDPC

Código	R	CG [dB] @ 10^{-8}	Limite de Shannon [24] [dB]
LDPC (4376, 4094) [21]	0,935	6,9	3,8
LDPC (32000, 29760) [21]	0,93	7,6	3,7
LDPC (64800, 58320) [42]	0,9	8,1	3,2
LDPC (4095, 3357) [21]	0,82	8,6	2,2
LDPC (64800, 51840) [42]	0,8	9,4	2,0

Dentre os códigos da Tabela 6.1, o que apresenta maior ganho de código (9,6 dB de ganho) e o que mais se aproxima do limite de Shannon (0,53 dB de distância) é o LDPC(64800, 51840). Desse modo, analisa-se a seguir, na Figura 6.3, como o aumento de E_b/N_0 reduz a taxa de erro de bit obtida desse código para variadas quantidades de iterações. Observa-se que a partir das 50 iterações o ganho de codificação tende a se estabilizar.

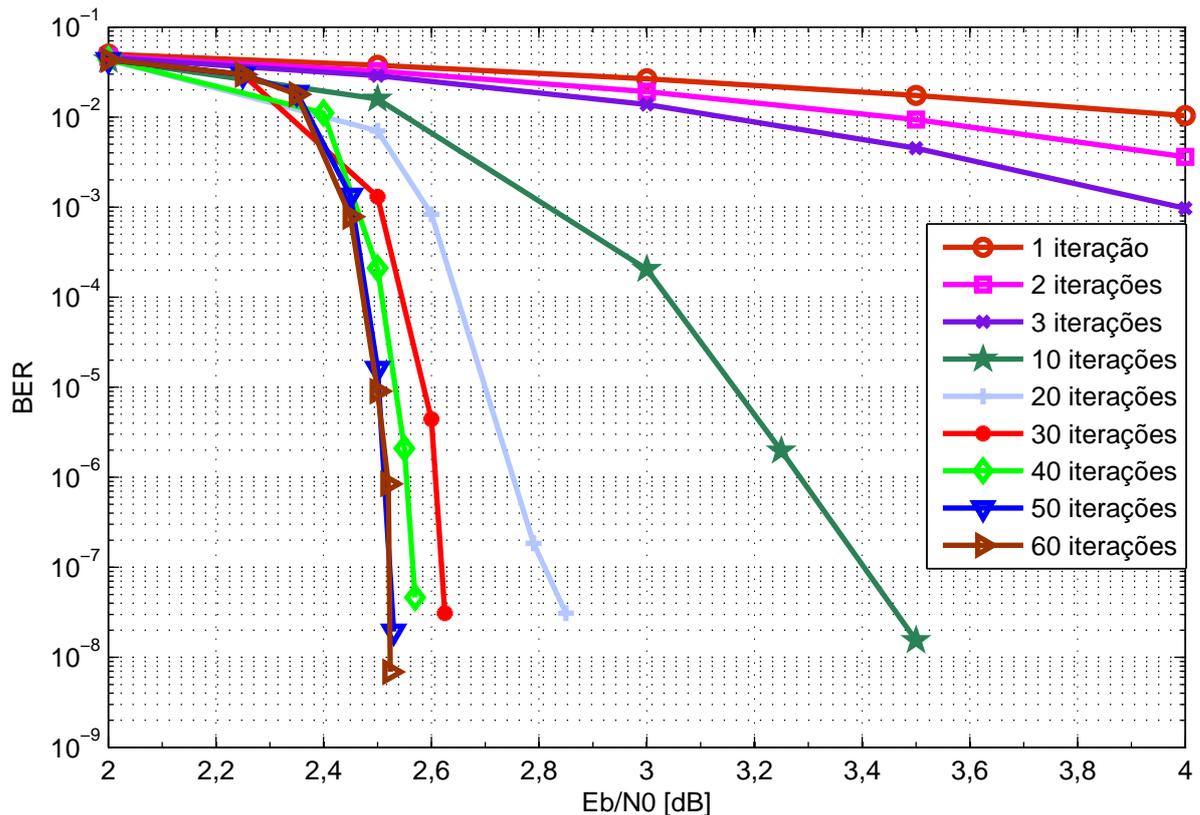


Figura 6.3: Desempenho de taxa de erro de bit para diferentes valores de iterações. Código LDPC(64800,51840).

De conhecimento do código que possui maior CG , analisa-se, para valores fixos de E_b/N_0 , a redução da BER com a quantidade de iterações realizadas, conforme mostrado na Figura 6.4. Observa-se que, para $E_b/N_0 \leq 2,45$ dB, o decodificador é pouco sensível ao aumento do número de iterações. A partir de $E_b/N_0 = 2,50$ dB, a taxa de erro de bit apresenta queda mais acentuada com o aumento de iterações, de modo que a partir de 50 iterações a BER não apresenta reduções expressivas.

Na Seção 6.4, a Figura 6.4 é utilizada como base para a estimação de potência de decodificação do código LDPC(64800, 51840) .

6.3 ESTIMAÇÃO DA POTÊNCIA DE DECODIFICAÇÃO PARA CÓDIGOS COM TAMANHO DE BLOCO PEQUENO

6.3.1 Entradas do decodificador

Na análise do consumo de potência do decodificador, algumas modificações no fluxo de sinais da Figura 6.1 foram feitas. Analisando o diagrama de blocos do sistema simulado, verifica-se

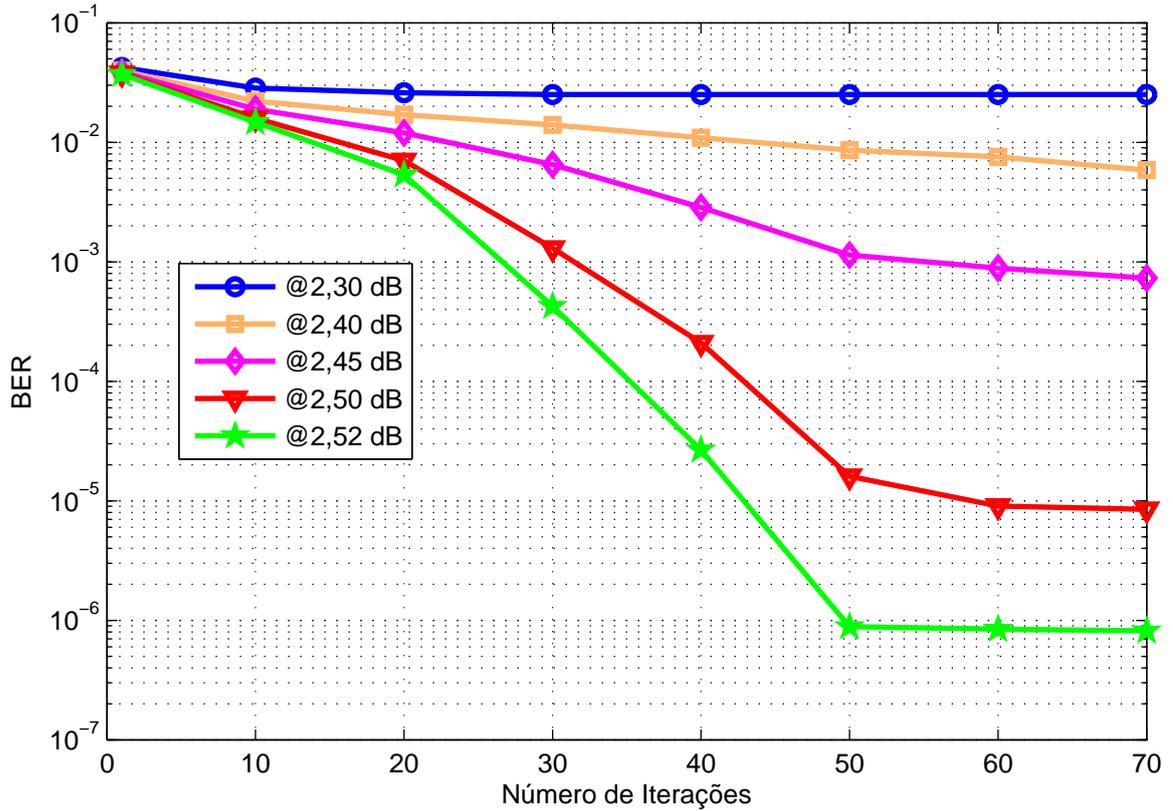


Figura 6.4: Relação entre BER e número de iterações do decodificador para o código LDPC(64800, 51840).

que as informações de entrada requeridas pelo decodificador são as probabilidades posteriores do canal e a matriz de teste de paridade do código. Entretanto, como a probabilidade $P(c_j = 0|\mathbf{r}_k)$ é função da probabilidade $P(c_j = 1|\mathbf{r}_k)$ (Equação 4.10), o envio do vetor de dados $P(c_j = 0|\mathbf{r}_k)$ não é necessário, já que essa informação pode ser calculada pelo decodificador. Com essa simplificação, o volume de dados de entrada do decodificador é reduzido, sem acréscimo significativo de complexidade no processamento do algoritmo de decodificação.

Outra simplificação realizada foi o armazenamento da matriz \mathbf{H} no próprio decodificador, descartando a necessidade de envio da matriz de teste de paridade a cada decodificação realizada.

Logo, as entradas do decodificador utilizado para o estudo de potência resumem-se aos valores quantizados do vetor $P(c_j = 1|\mathbf{r}_k)$ e ao sinal de referência do relógio.

Considerando que o nível de quantização utilizado para as entradas é de 8 bits, o vetor de probabilidades posteriores $P(c_j = 1|\mathbf{r}_k)$ de tamanho n , possui, após a quantização, $8n$ bits. Como esses $8n$ bits são processados simultaneamente cada vez que um vetor de dados é recebido, a análise de potência limitou-se a códigos muito pequenos ($n \leq 30$), dado que o número total de portas de entrada/saída do FPGA é limitado a 1200 bits [43] e que não foi possível multiplexar a entrada de dados no domínio do tempo.

6.3.2 Fluxo de etapas

A descrição de etapas desta Subseção é realizada para a estimação de potência de decodificação dos códigos LDPC(10,5) ($w_c = 3$, $w_r = 6$), LDPC(20,15) ($w_c = 3$, $w_r = 12$) e LDPC(30,25) ($w_c = 3$, $w_r = 18$).

A partir da implementação do decodificador LDPC descrita na linguagem do Matlab, realizou-se o fluxo de tarefas da Figura 6.5 para a criação de uma descrição em VHDL do algoritmo de decodificação. Para tanto, utilizou-se a ferramenta HDL Coder [44] disponível no ambiente de simulação Matlab R2012a.

Seguindo as etapas da Figura 6.5, o primeiro passo consiste em determinar a função a ser traduzida (decodificador LDPC Soma-Produto Simplificado) e as suas condições de contorno (entradas da função) definidas num arquivo de teste (*test bench*). Depois, valida-se a simulação dessa função original e armazenam-se os valores de saída do decodificador (1º teste de verificação), sendo que todas as variáveis do algoritmo estão em ponto flutuante.

A seguir, baseado nas faixas de valores encontradas para cada variável da função de decodificação, escolhe-se a quantidade de bits de quantização requeridos para cada variável. Para que o decodificador em VHDL apresente desempenho similar ao descrito pela linguagem do Matlab, utilizaram-se 8 bits de quantização para os sinais de entrada e até 10 bits para os sinais internos do decodificador.

Após a quantização, gera-se a função de decodificação em ponto fixo, que posteriormente é simulada e tem seu resultado (palavra código estimada) comparado com a saída do decodificador original, caracterizando o 2º teste de verificação. Caso as saídas do decodificador em ponto fixo e em ponto flutuante não coincidam, retoma-se a reavaliação das etapas anteriores. Caso contrário, prossegue-se com a geração da descrição em VHDL.

De modo similar à verificação do projeto em ponto fixo, valida-se a implementação da função em VHDL comparando o vetor estimado obtido com o vetor esperado calculado pelo algoritmo original (3º teste de verificação). Em caso de sucesso, prossegue-se para a criação do projeto em FPGA. Caso contrário, é necessário verificar a ocorrência de erro causada por alguma das etapas anteriores. Para a criação do projeto define-se a ferramenta de síntese (ISE Design Tool 13.1 [45]), a família do *chip* (Virtex 7) e o modelo do dispositivo (xc5vfx2000t) [37, 43].

Uma vez que o projeto foi criado na ferramenta de síntese de FPGA, realizou-se no *software* ISE Design 13.1 as etapas de síntese e *place and route*.

Na última etapa da Figura 6.5, o programa XPower Analyzer realiza a estimação de potência do projeto finalizado. Para se aproximar do cenário óptico, adotou-se neste projeto uma vazão de dados de 28 Gbps³ com relógio operando a 815 MHz, frequência máxima permitida pelo relatório de síntese do projeto. O relatório de consumo potência (1 iteração) é mostrado na Figura 6.6.

Para avaliar aumento da potência consumida devido ao aumento do número de iterações,

³Valor típico adotado em receptores ópticos DP-QPSK [46].

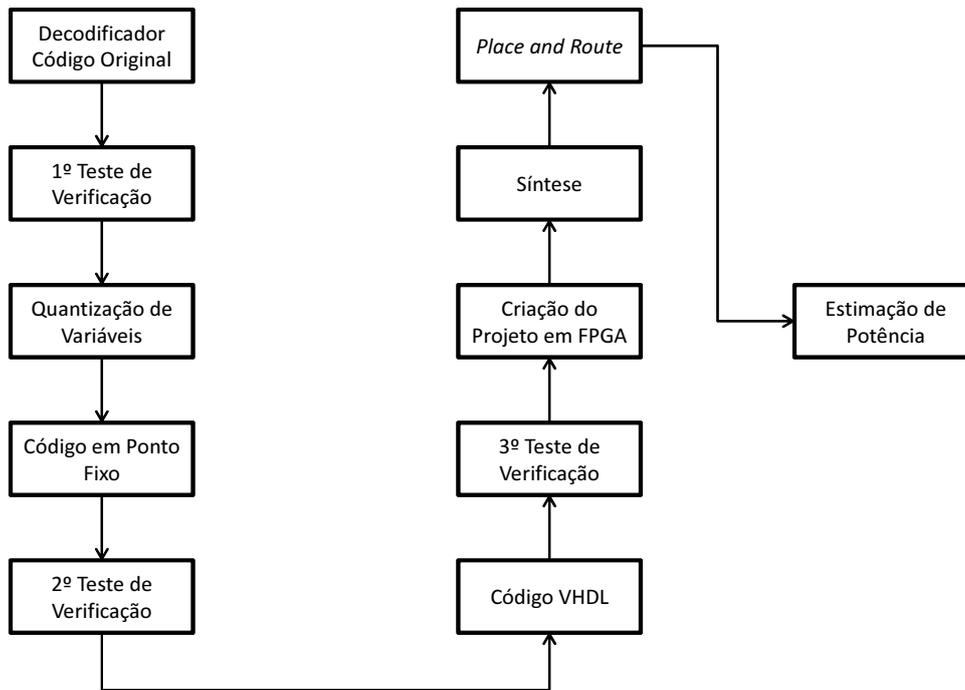


Figura 6.5: Fluxo de tarefas para a criação de descrição do algoritmo de decodificação em linguagem de *hardware* (VHDL).

Device	On-Chip	Power (W)	Used	Available	Utilization (%)	Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)		
Family	Virtex7					Clocks	0.013	1	---	---		
Part	xc7v2000t					Logic	0.028	692	1221600	0		
Package	ffg1925					Signals	0.247	2195	---	---		
Grade	Commercial					DSPs	0.647	150	2160	7		
Process	Typical					I/Os	1.397	94	1200	8		
Speed Grade	-1					Leakage	2.360					
Environment						Total	4.692					
Ambient Temp (C)	50.0					Thermal Properties		Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)		
Use custom TJA?	No							2.1	75.1	59.9		
Custom TJA (C/W)	NA					Supply Power (W)						
Airflow (LFM)	250							Total	Dynamic	Quiescent		
Heat Sink	None							4.692	2.332	2.360		
Custom TSA (C/W)	NA					Supply Summary						
Board Selection	Medium (10"x10")							Vccint	1.000	2.661	0.979	1.681
# of Board Layers	12 to 15							Vccaux	1.800	0.421	0.044	0.377
Custom TJB (C/W)	NA							Vcco18	1.800	0.708	0.708	0.000
Board Temperature (C)	NA											

Figura 6.6: Relatório de consumo de potência de decodificação (1 iteração), LDPC(10,5). Vazão de dados de 28 Gbps, frequência de relógio de 815 MHz.

considerou-se a seguinte relação para o cálculo da vazão de dados (T) [47]:

$$T = \frac{f_c * n}{N_{iter} * L}, \quad (6.2)$$

em que f_c é a frequência do relógio, n é o tamanho da palavra código, N_{iter} é o número de iterações realizadas e L é o número de ciclos do relógio requeridos para o processamento de cada iteração. Assim, pela Equação 6.2 verifica-se que, para manter a vazão de dados constante, o aumento do número de iterações requer que o decodificador opere numa frequência proporcionalmente maior. Como a potência dinâmica depende linearmente da frequência do relógio (Equação 5.3), a potência total consumida pelo decodificador pode ser modelada da seguinte forma:

$$P_{Total} = P_{Est} + P_{Din/iter}N_{iter}, \quad (6.3)$$

em que $P_{Din/iter}$ é a potência dinâmica consumida por iteração. Desse modo, pelos valores de potência descritos no relatório de consumo da FPGA (Figura 6.6), a potência total de decodificação do código LDPC(10,5) é dada por

$$P_{total} = 2,36 + 2,33N_{iter}, \quad (6.4)$$

em que 2,36 W correspondem à potência estática e 2,33 W são de potência dinâmica requerida por iteração. O gráfico da Figura 6.7 mostra a relação entre potência total de decodificação e número de iterações descrita pela Equação 6.4 para até 4 iterações.

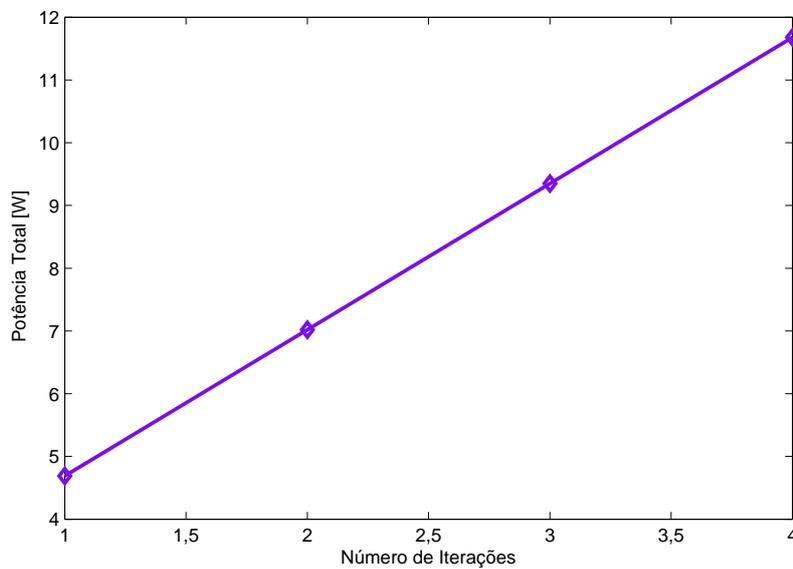


Figura 6.7: Potência de decodificação pelo número de iterações para o código LDPC(10,5). Decodificador CMOS 28 nm, $T = 28$ Gbps e $f_c = 815$ MHz.

Em seguida, deseja-se observar o comportamento da potência de decodificação com o aumento de n . Simulou-se, então, a potência consumida para os códigos regulares (20,15) e (30,25) (ambos com $w_c = 3$). A Figura 6.8 ilustra o reflexo em potência causado pelo aumento do tamanho da palavra código.

6.4 ESTIMAÇÃO DA POTÊNCIA DE DECODIFICAÇÃO PARA CÓDIGOS COM TAMANHO DE BLOCO GRANDE

Relacionando a Equação 6.3 com o gráfico de desempenho de BER da Figura 6.4, conclui-se que, independente dos valores de P_{Est} e $P_{Din/iter}$, o formato da curva de BER por potência de

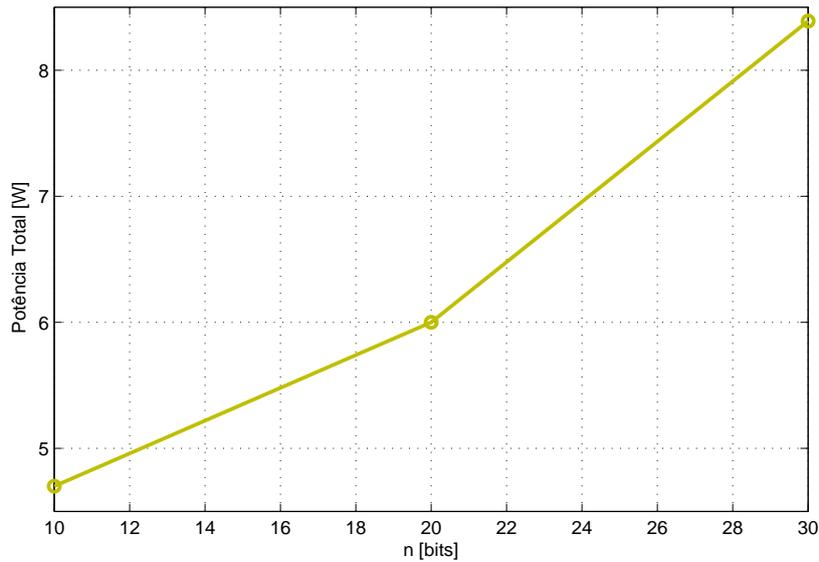


Figura 6.8: Potência de decodificação para n igual a 10, 20 e 30 bits, 1 iteração. Decodificador CMOS 28 nm, $T = 28$ Gbps e $f_c = 815$ MHz.

decodificação (P_{Total}) apresentará o mesmo formato da curva de BER pelo número de iterações. A razão disso acontecer deve-se à dependência linear entre P_{Total} e N_{iter} existente na Equação 6.3.

A partir das Figuras 6.8 e 6.7, e da Equação 6.3, constata-se que a estimativa de potência consumida pelo decodificador em arquitetura serial resultaria em valores de potência muito acima dos toleráveis caso n fosse da ordem dos milhares de bits e caso o decodificador operasse com dezenas de iterações.

Dessa forma, com o intuito de representar a relação entre BER e P_{Total} com valores de potência mais realistas dos que os seriam encontrados com a arquitetura utilizada neste trabalho (serial), utilizaram-se como referência os dados de um decodificador LDPC implementado em arquitetura paralela [41]. Para tal análise, são necessárias algumas adaptações das condições de contorno do trabalho de [41].

Segundo [39], observa-se que a energia por bit decodificado por iteração ($E_{b_{dec/iter}}$) varia quadraticamente com a tensão de alimentação do dispositivo⁴ e linearmente com o tamanho da tecnologia de processo de fabricação.

O decodificador LDPC de [41] possui as seguintes características gerais a serem adaptadas para este trabalho: tecnologia de fabricação CMOS 160 nm, alimentação de 1,5 V, 1 Gbps de vazão de dados, potência de consumo⁵ igual a 690 mW (para 64 iterações) e eficiência energética de 10,78 pJ/bit/iteração. Adaptando esses valores para as condições requeridas neste projeto, a energia de bit decodificado por iteração será igual a

⁴O FPGA possui mais de uma fonte de alimentação, como descrito na Figura 6.6. A análise é feita tomando por base somente a fonte de alimentação interna, de tensão V_{CCINT} .

⁵Quanto maior o processo de fabricação, mais desprezível é a parcela estática da potência total [4]. Desse modo, para a tecnologia CMOS 160 nm, considera-se $P_{Total} \approx P_{Din}$.

$$Eb_{dec/iter} = 10,78 * \left(\frac{1}{1,5}\right)^2 * \left(\frac{28}{160}\right) = 0,838 \text{ [pJ/bit/iteração]}. \quad (6.5)$$

Como $Eb_{dec/iter} = P_{Total/iter}/T$ e $P_{Total} \approx P_{Din}$ em [41], para vazão de 28 Gbps, a potência dinâmica por iteração seria de 23,5 mW. Considerando que a potência estática torna-se mais significativa conforme a tecnologia de fabricação diminui de tamanho [4] e levando-se em conta que a potência estática para a tecnologia 160 nm empregada em [41] é desprezível, soma-se à potência dinâmica a potência estática dos projetos de 28 nm, aproximadamente igual a 2,3 W na estimacão descrita da Figura 6.6.

Assim, o modelo adaptado de potência para decodificador CMOS 28 nm ($T = 28$ Gbps e $f_c = 815$ MHz) é descrito pela relação

$$P_{total} = 2,36 + (0,0235)N_{iter} \text{ [W]}. \quad (6.6)$$

A partir da Equação 6.6 e do desempenho do código LDPC(64800, 51840) descrito na Figura 6.4, chega-se às relações entre taxa de erro de bit e potência total consumida pelo decodificador mostradas na Figura 6.9.

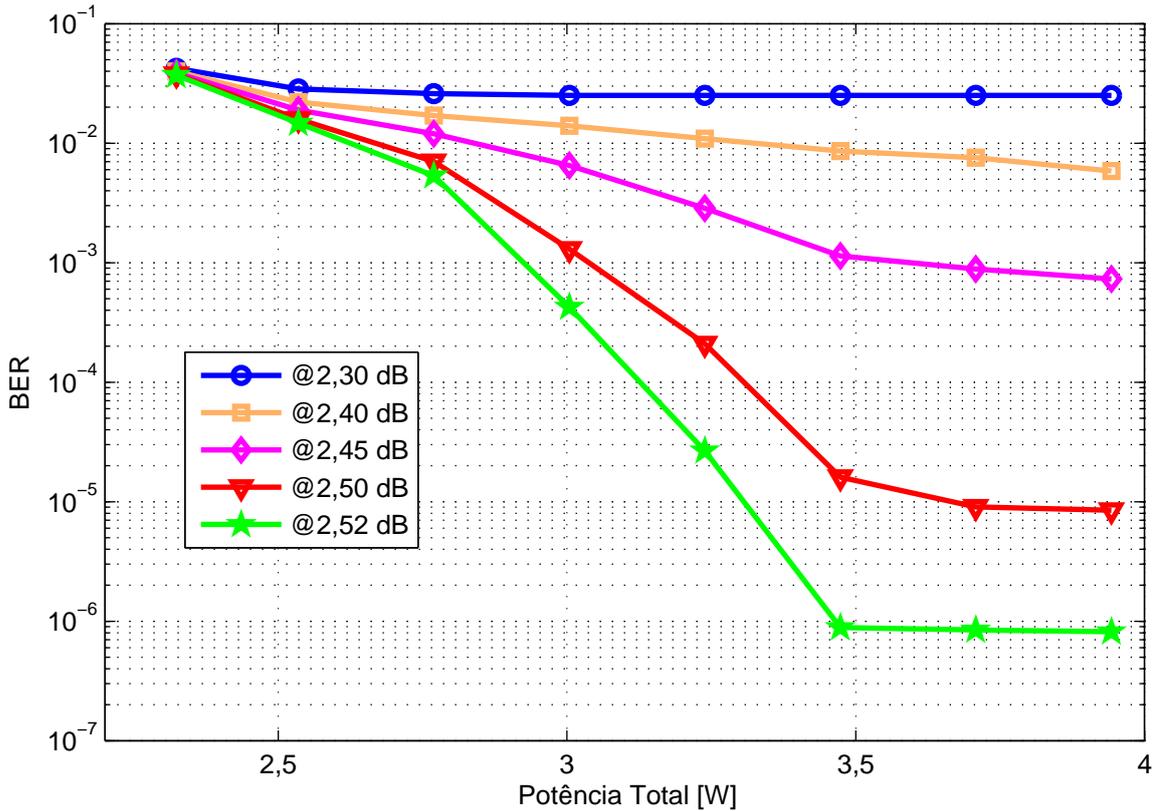


Figura 6.9: Relação entre BER e estimativa de potência de decodificação para código LDPC(64800, 51840). Decodificador CMOS 28 nm, $f_c = 815$ MHz e $T = 28$ Gbps.

Verifica-se, na Figura 6.9, que para as curvas nas quais $E_b/N_0 \geq 2,50$ dB existe uma região (acima de 3,47 W) na qual a capacidade de correção do decodificador satura e não há melhora de desempenho caso o decodificador consuma mais potência. Para as curvas nas quais $E_b/N_0 < 2,50$ dB, o aumento do número de iterações não promove melhoras significativas da taxa de erro de bit. Isso ocorre porque o sinal recebido pelo decodificador está tão corrompido por ruído que o código LDPC não consegue prover ganho de informação extrínseca suficiente para correção significativa dos erros.

Outra observação importante é que acima de um limiar de E_b/N_0 (aproximadamente 2,50 dB) o decodificador começa a atingir a sua região de melhor desempenho (região de queda brusca). Outra forma de se avaliar o E_b/N_0 de limiar de um código LDPC é dada pela técnica analítica de *density evolution* [2].

Dessa forma, para se obter o melhor desempenho do decodificador ao menor custo de potência, deve-se aplicar até 50 iterações para o código LDPC(64800, 51840) com E_b/N_0 maior que 2,50 dB.

Apesar de a Figura 6.3 não registrar a existência de piso de ruído para $BER > 10^{-8}$, existe a possibilidade de esse fenômeno ocorrer para $BER \leq 10^{-8}$. Para contornar esse problema, umas das soluções apresentadas na Seção 4.6 deve ser empregada.

Capítulo 7

CONCLUSÃO

Este trabalho apresentou um breve panorama do estado da arte dos códigos corretores de erro aplicados a comunicações ópticas. Seguindo a evolução das gerações de FEC, focou-se a análise em códigos LDPC.

As principais qualidades dos códigos de Gallager são seu desempenho muito próximo ao limite de Shannon e a baixa complexidade computacional de seu decodificador.

Sabendo que os principais objetivos das evoluções tecnológicas focam na redução dos custos de implementação do sistema e no aumento do produto banda disponível por alcance, deve-se conhecer os limites de desempenho e de potência dos decodificadores LDPC.

Baseado em ferramentas automatizadas de conversão de códigos em linguagem do Matlab para VHDL, este trabalho estimou o consumo de potência de códigos pequenos e descreveu como a potência de decodificação varia com o número de iterações e com o tamanho da palavra código. Como a implementação projetada para FPGA possui arquitetura serial, a potência de decodificação encontrada apresentou valores muito altos para tornar viável a utilização do projeto em sistemas práticos.

A partir da análise da eficiência energética e do estudo do comportamento da potência de decodificação com o número de iterações, realizou-se uma estimativa de potência de decodificação considerando o emprego de arquitetura paralela de [41] em tecnologia de fabricação 28 nm, com condições de contorno próximas do cenário óptico.

Por fim, verificou-se o comportamento da taxa de erro de bit com a potência demandada pelo decodificador. Constatou-se que, a partir de um certo consumo de potência, o decodificador não proporciona ganhos de desempenho relevantes. Além disso, é requerido um valor mínimo de E_b/N_0 para se atingir a região de melhor desempenho (quedra brusca) do código LDPC.

7.1 TRABALHOS FUTUROS

A partir dos resultados apresentados neste trabalho, conclui-se que para a aplicação de códigos LDPC longos requer-se o emprego da arquitetura paralela. Essa arquitetura reduz o consumo de

potência e dá mais agilidade ao processamento, e, por outro lado, aumenta bastante a complexidade em *hardware* da implementação [48]. Dessa maneira, necessita-se de um estudo mais detalhado do projeto em FPGA por meio de uma implementação direta em VHDL.

Outra questão que não foi avaliada no trabalho é a latência do decodificador. Códigos com n da ordem dos milhares de bits levam o decodificador a apresentar sérios problemas de latência [5]. Uma proposta para contornar esse obstáculo está na utilização de códigos LDPC não-binários [5].

Além disso, a simulação de um enlace óptico com modelo de canal que também considere os efeitos de dispersão cromática, dispersão dos modos de polarização e não-linearidades proporcionaria resultados mais próximos da realidade dos sistemas ópticos.

Por fim, deve-se considerar também que a taxa de erro de bit requerida por sistemas ópticos é da ordem de 10^{-15} , região na qual provavelmente o código possuirá piso de erro. As soluções descritas na Seção 4.6 devem ser cuidadosamente estudadas para se mitigar esse fenômeno.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] International Telecommunication Union. *Recommendation G.975.1: Forward error correction for high bit rate DWDM submarine systems*. Feb. 2004.
- [2] MOON, T. *Error Correction Coding: Mathematical Methods and Algorithms*. : Wiley, 2005. ISBN 978-0-471-73914-2.
- [3] RICHARDSON, T.; URBANKE, R. Efficient encoding of Low-Density Parity-Check codes. *Information Theory, IEEE Transactions on*, v. 47, n. 2, p. 638 –656, Feb. 2001. ISSN 0018-9448.
- [4] XILINX. *FPGA Design Training: FPGA Power*. Nov. 2012. Disponível em: <<http://www.xilinx.com/training/free-video-courses.htm>>.
- [5] DJORDJEVIC, I. B.; RYAN, W.; VASIC, B. *Coding for Optical Channels*. : Springer, 2010. ISBN 978-1-441-95568-5.
- [6] AGRAWAL, G. P. *Fiber-Optic Communication Systems*. Fourth. : Wiley, 2010. ISBN 978-0-470-50511-3.
- [7] WINZER, P. J.; ESSIAMBRE, R.-J. Advanced Modulation Formats for High-Capacity Optical Transport Networks. *Lightwave Technology, Journal of*, v. 24, n. 12, p. 4711 –4728, Dec. 2006. ISSN 0733-8724.
- [8] MOREIRA, P. G. F. J. *Essentials of Error-Control Coding*. : Wiley, 2006. ISBN 978-0-470-02920-6.
- [9] DESBRUSLAIS, S.; SAVORY, S. Optimizing the fec overhead in a 100 gbe pdm-qpsk digital coherent transmission system. *Lightwave Technology, Journal of*, v. 29, n. 8, p. 1118 –1126, april15, 2011. ISSN 0733-8724.
- [10] GROVER, W. Forward error correction in dispersion-limited lightwave systems. *Lightwave Technology, Journal of*, v. 6, n. 5, p. 643 –654, May 1988. ISSN 0733-8724.
- [11] MORO, P.; CANDIANI, D. 565 Mb/s optical transmission system for repeaterless sections up to 200 km. In: *Communications, 1991. ICC '91, Conference Record. IEEE International Conference on*. 1991. p. 1217 –1221 vol.3.

- [12] International Telecommunication Union. *Recommendation G.975: Forward error correction for high bit rate DWDM submarine systems*. Nov. 1996.
- [13] International Telecommunication Union. *Recommendation G.709: Interfaces for the Optical Transport Network (OTN)*. Apr. 2012.
- [14] MIZUOCHI, T. Next Generation FEC for Optical Communication. In: *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*. 2008. p. 1 –33.
- [15] TYCHOPOULOS, A.; KOUFOPAULOU, O.; TOMKOS, I. FEC in optical communications - A tutorial overview on the evolution of architectures and the future prospects of outband and inband FEC for optical communications. *Circuits and Devices Magazine, IEEE*, v. 22, n. 6, p. 79 –86, Nov. 2006. ISSN 8755-3996.
- [16] DJORDJEVIC, I.; ARABACI, M.; MINKOV, L. Next Generation FEC for High-Capacity Communication in Optical Transport Networks. *Lightwave Technology, Journal of*, v. 27, n. 16, p. 3518 –3530, Aug. 2009. ISSN 0733-8724.
- [17] DJORDJEVIC, I.; ARABACI, M.; MINKOV, L. Next Generation FEC for High-Capacity Communication in Optical Transport Networks. *Lightwave Technology, Journal of*, v. 27, n. 16, p. 3518 –3530, Aug. 2009. ISSN 0733-8724.
- [18] GROVER, P.; WOYACH, K. A.; SAHAI, A. Towards a communication-theoretic understanding of system-level power consumption. p. 1–24, 2012.
- [19] GANESAN, K.; GROVER, P.; RABAEY, J. The power cost of over-designing codes. In: *Signal Processing Systems (SiPS), 2011 IEEE Workshop on*. 2011. p. 128 –133. ISSN 2162-3562.
- [20] CARENA, A. et al. Statistical characterization of PM-QPSK signals after propagation in uncompensated fiber links. In: *Optical Communication (ECOC), 2010 36th European Conference and Exhibition on*. 2010. p. 1 –3.
- [21] MACKAY, D. J. C. *Encyclopedia of Sparse Graph Codes*. Sep. 2012. Disponível em: <<http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>>.
- [22] GOLDSMITH, A. *Wireless Communications*. : Cambridge University Press, 2005. ISBN 978-0-521-83716-3.
- [23] ESSIAMBRE, R.-J. et al. Capacity Limits of Optical Fiber Networks. *Lightwave Technology, Journal of*, v. 28, n. 4, p. 662 –701, Feb. 2010. ISSN 0733-8724.
- [24] LIN, S.; COSTELLO, D. *Error Control Coding*. : Prentice Hall, 2nd edition, 2004. ISBN 0-13-017973-6.
- [25] GALLAGER, R. In: *Low-Density Parity-Check Codes*. 1962. IT-8, p. 21–28.

- [26] BONELLO, N.; CHEN, S.; HANZO, L. Design of Low-Density Parity-Check Codes. *Vehicular Technology Magazine, IEEE*, v. 6, n. 4, p. 16–23, Dec. 2011. ISSN 1556-6072.
- [27] LUBY, M. et al. Improved Low-Density Parity-Check Codes Using Irregular Graphs. *Information Theory, IEEE Transactions on*, v. 47, n. 2, p. 585–598, feb 2001. ISSN 0018-9448.
- [28] MIYATA, Y. et al. Efficient FEC for Optical Communications using Concatenated Codes to Combat Error-floor. p. 11–13, 2008.
- [29] MACKAY, D.; NEAL, R. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, v. 32, n. 18, p. 1645, Aug. 1996. ISSN 0013-5194.
- [30] DIAS, C. F. et al. Análise de códigos corretores de erros aplicados a canais ópticos. *XXX Simpósio Brasileiro de Telecomunicações*, 2012.
- [31] POON, K. K. W.; WILTON, S. J. E.; YAN, A. A detailed power model for field-programmable gate arrays. *ACM Transactions on Design Automation of Electronic Systems*, v. 10, n. 2, p. 279–302, Apr. 2005. ISSN 10844309. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1059876.1059881>>.
- [32] XILINX. *Xilinx Power Estimator*. Nov. 2012. Disponível em: <<http://goo.gl/zvOkH>>.
- [33] XILINX. *Xilinx XPower Analyzer*. Nov. 2012. Disponível em: <<http://goo.gl/eLRuh>>.
- [34] CHU, P. *RTL Hardware Design using VHDL: Coding for Efficiency, Portability, and Scalability*. : John, Wiley and Sons, 2006. ISBN 978-0-471-72092-8.
- [35] HELP Documentation: HDL Code Generation from MATLAB Code. : Mathworks, 2012.
- [36] HOWARD, S. L.; SCHLEGEL, C.; INIEWSKI, K. Error Control Coding in Low-Power Wireless Sensor Networks: When Is ECC Energy-Efficient? *EURASIP Journal on Wireless Communications and Networking*, v. 2006, p. 1–14, 2006. ISSN 1687-1472. Disponível em: <<http://jwcn.erasipjournals.com/content/2006/1/074812>>.
- [37] XILINX. *Virtex-7 T and XT FPGAs Data Sheet: DC and Switching Characteristics*. Nov. 2012. Disponível em: <<http://goo.gl/FMZV1>>.
- [38] XIANG, B. et al. An Area-Efficient and Low-Power Multirate Decoder for Quasi-Cyclic Low-Density Parity-Check Codes. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, v. 18, n. 10, p. 1447–1460, Oct. 2010. ISSN 1063-8210.
- [39] DARABIHA, A.; CARUSONE, A. C.; KSCHISCHANG, F. Power Reduction Techniques for LDPC Decoders. *Solid-State Circuits, IEEE Journal of*, v. 43, n. 8, p. 1835–1845, Aug. 2008. ISSN 0018-9200.
- [40] KASNAVI, S. et al. Application of Error Control Codes (ECC) in Ultra-Low Power RF Transceivers. *2005 IEEE Dallas/CAS Workshop on Architecture, Circuits and Implementation of SOCs*, IEEE, n. d, p. 195–198, 2005. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1611169>>.

- [41] BLANKSBY, A. J.; HOWLAND, C. J. A 690-mW 1-Gb/s 1024-b, Rate-1/2 Low-Density Parity-Check Code Decoder. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, v. 37, n. 3, p. 404–412, 2002.
- [42] European Standart (Telecommunications series) ETSI EN 302 307 - v1.2.1. *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)*. Aug. 2009.
- [43] XILINX. *7 Series FPGAs Overview*. Nov. 2012. Disponível em: <<http://goo.gl/NTtAU>>.
- [44] MATHWORKS. *HDL Coder*. Nov. 2012. Disponível em: <<http://www.mathworks.com/help/hdlcoder/gs/example-generating-hdl-code-from-matlab-code.html>>.
- [45] XILINX. *ISE Design Suite*. Nov. 2012. Disponível em: <<http://www.xilinx.com/products/design-tools/ise-design-suite/>>.
- [46] SANKAR, B. et al. Chromatic Dispersion Compensation - An Energy Perspective. n. 1, p. 3–5, 2012.
- [47] FANUCCI, L.; ROSSI, F. A throughput/complexity analysis for the VLSI implementation of LDPC decoder. In: *Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International Symposium on*. 2004. p. 409 – 412.
- [48] YING, Y. et al. A cost efficient LDPC decoder for DVB-S2. In: *ASIC, 2009. ASICON '09. IEEE 8th International Conference on*. 2009. p. 1007 –1010.

ANEXOS

I. DECISÃO MAP E ML

A partir da função de verossimilhança (Equações 4.3 e 4.4) é necessário determinar regras de decisão para o vetor estimado $\tilde{\mathbf{s}} = [\tilde{x}_1 \ \tilde{x}_2]$. O critério de decisão para $\tilde{\mathbf{s}}$ que minimiza a probabilidade de erro é dada por

$$\tilde{\mathbf{s}} = \arg \max_{\mathbf{s} \in \mathcal{S}} P(\mathbf{s}|\mathbf{r}). \quad (\text{I.1})$$

Pela Teorema de Bayes, sabe-se que

$$P(\mathbf{s}|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{s})P(\mathbf{s})}{P(\mathbf{r})}. \quad (\text{I.2})$$

Substituindo a Equação I.2 em I.1, tem-se que

$$\tilde{\mathbf{s}} = \arg \max_{\mathbf{s} \in \mathcal{S}} \frac{P(\mathbf{r}|\mathbf{s})P(\mathbf{s})}{p_R(\mathbf{r})}. \quad (\text{I.3})$$

Como o denominador da Equação I.3 não depende de \mathbf{s} , verifica-se que

$$\tilde{\mathbf{s}} = \arg \max_{\mathbf{s} \in \mathcal{S}} P(\mathbf{r}|\mathbf{s})P(\mathbf{s}), \quad (\text{I.4})$$

definindo assim o critério de decisão de máxima probabilidade *a posteriori* (*maximum a posteriori probability* - MAP)

Caso todos os símbolos $\tilde{\mathbf{s}}$ sejam equiprováveis ($P(\mathbf{s}) = 1/M$), a Equação I.4 simplifica-se em

$$\tilde{\mathbf{s}} = \arg \max_{\mathbf{s} \in \mathcal{S}} P_{R|S}(\mathbf{r}|\mathbf{s}), \quad (\text{I.5})$$

que, por sua vez, define o critério de decisão de máxima verossimilhança (*maximum likelihood* - ML).