

Ampliação de Ganho de Codificação em um Código EFEC BCH para Comunicações Ópticas

Diego Brito de Carvalho, Arley Henrique Salvador, Érico Nunes Ferreira Bastos, Luiz Juberto Rossi de Jesus e Max H. M. Costa

Resumo—Devido a fatores como o crescente número de dispositivos conectados à internet, o núcleo das redes de comunicação óptica vem sofrendo grande aumento de tráfego. Portanto, estudar alternativas que concedam mais capacidade e robustez a essas redes, incluindo códigos corretores de erro aprimorados, torna-se fundamental. Este artigo apresenta dados da implementação em FPGA (*Field-Programmable Gate Array*) de um código BCH derivado do proposto no Anexo I.9 da Recomendação do ITU-T G.975.1 [1]. Tal código se vale de um aumento de redundância (paridade) e da reorganização das estruturas de quadro para produzir um ganho de codificação superior ao do código original.

Palavras-Chave—Código Corretor de Erro, BCH, EFEC, Recomendação ITU G.975.1 – Anexo I.9, Comunicações Ópticas.

Abstract—Due to factors such as the growing number of internet-connected devices, the core of optical communication networks is suffering great increase in traffic. Therefore, studying alternatives that provide more capacity and robustness to these networks, including improved error correcting codes, is of fundamental importance. This paper presents data of an FPGA implementation of a BCH code derived from the channel code proposed in Annex I.9 of ITU-T Recommendation G.975.1 [1]. By increasing redundancy (parity) and by reorganizing frame structures, the new code produces a higher coding gain than the original code.

Keywords—Error Correcting Code, BCH, EFEC, Recommendation ITU G.975.1 – Annex I.9, Optical Communications.

I. INTRODUÇÃO

O crescente aumento do tráfego de dados através de redes de telecomunicações tem impulsionado a evolução dos sistemas de transmissão já existentes e a criação de novas soluções que atendam a essa demanda. A principal tecnologia responsável por esse crescimento é a tecnologia fotônica. Desde a introdução dos primeiros sistemas de transmissão óptica, a capacidade de transmissão tem aumentado e o custo por *bit* transmitido diminuído gradualmente. Ao longo do tempo surgiram (ou foram agregadas) novas tecnologias aos sistemas de transmissões ópticas que permitiram esta evolução, tais como a tecnologia de transmissão óptica WDM (*Wavelength Division Multiplexing*), a transmissão coerente e os algoritmos corretores de erro [2].

A primeira versão do padrão definido para redes ópticas (OTN – *Optical Transport Network*) foi publicada na década de 90 pelo ITU-T [3]. Trata-se da especificação de um envelope digital para encapsular diferentes tipos de protocolos como, por

exemplo, SDH (*Synchronous Digital Hierarchy*), ATM (*Asynchronous Transfer Mode*) e Ethernet, com taxas de dados que chegam até 10Gbit/s. Nesse mesmo período o Grupo de Estudos 15 da ITU-T também elaborou a recomendação para o uso do FEC (*Forward Error Correction*) em sistemas ópticos submarinos, de onde originou-se a Recomendação do ITU-T G.975 [4]. O documento especifica que o algoritmo Reed-Solomon RS(255,239) é um código adequado para aplicações em redes ópticas. Esse código corretor de erros é classificado como um código de bloco [5] e possui um ganho de correção de 6,3 dB (NCG - *Net Coding-Gain*), sob uma condição de taxa de erros (BER – *Bit Error Rate*) de 10^{-15} .

Posteriormente uma nova versão da Recomendação ITU-T G.975.1 [1] foi publicada com 8 novos esquemas de codificação FEC com ganhos superiores ao RS(255,239), os chamados Super-FEC ou *Enhanced FEC* (EFEC). O esquema apresentado no Anexo I.9 desse documento é um dos que possuem o melhor desempenho entre as 8 estruturas propostas e consiste de dois códigos BCH(1020,988) entrelaçados, com uma redundância de 6,69%, o que é compatível com o RS(255,239). O ganho de codificação deste tipo de estrutura de EFEC é alterado de acordo com o número de iterações entre os códigos. Conforme apresentado em [1], após 10 iterações e submetido a uma BER de entrada igual a 4×10^{-3} , o ganho de codificação do EFEC I.9 resulta maior que 8,5 dB, medido para uma BER de saída igual a 10^{-13} . Por esta razão, a classe de códigos corretores EFEC tornaram-se uma alternativa de melhoria do desempenho dos sistemas de transmissões ópticas em resposta ao crescimento da demanda de consumo.

Em projetos de sistemas de comunicação óptica, há casos em que se deseja ampliar ao máximo a distância de um enlace sem que haja elementos regeneradores de sinal entre o transmissor e o receptor. Nesses casos é comum o uso de códigos corretores de erro com alto ganho de codificação, como os propostos na Seção Anexo I.9 da Recomendação G.975.1 [1]. Também pode ser interessante o uso de uma taxa de transmissão um pouco mais elevada para permitir o aumento da redundância e, assim, viabilizar um código corretor de erros com ganho de codificação ainda maior, que se reflete em um enlace mais longo [6].

O presente trabalho dedica-se exclusivamente ao tema de códigos corretores de erro e tem como objetivo apresentar dados de duas implementações em FPGA de códigos BCH(1020,988) entrelaçados: uma referindo-se ao EFEC I.9 tal qual descrito em [1] e outra derivada da primeira, mas que busca melhorar o ganho obtido por ela através do aumento da redundância de 7% para 20%, aproximadamente. Portanto, os EFECs deste trabalho serão tratados como EFEC I.9 7% e EFEC I.9 20%.

O artigo foi dividido em 6 seções. Na Seção II são apresentadas as características da implementação do EFEC I.9 com 7% de redundância e também da arquitetura do circuito implementado. Na Seção III são apresentadas as modificações introduzidas para possibilitar a operação com 20% de redundância. A Seção IV descreve a etapa experimental de avaliação do uso de FEC em sistemas de comunicações ópticas. A Seção V expõe os resultados obtidos. Na Seção VI são apresentadas as conclusões deste trabalho e as contribuições oferecidas pela análise experimental realizada.

II. IMPLEMENTAÇÃO DO CÓDIGO EFEC I.9

O anexo I.9 da Recomendação G.975.1 utiliza dois códigos de bloco BCH(1020,988) entrelaçados para codificação e decodificação de dados. Esse código é uma versão derivada do código BCH(1023,993), que tem parâmetros $t=3$ e $m=10$. Isso quer dizer que 3 erros podem ser corrigidos por palavra-código e 30 bits de paridade são requeridos para tal. No BCH(1020,988) a área de dados é 5 bits menor do que no código BCH(1023,993), e à sua área de paridade são acrescidos 2 bits para proteção adicional.

O polinômio de campo utilizado é mostrado na Eq. 1:

$$p(x) = x^{10} + x^3 + 1 \quad (1)$$

Os polinômios mínimos são mostrados nas Eq. 2, 3 e 4:

$$m_1(x) = x^{10} + x^3 + 1 \quad (2)$$

$$m_3(x) = x^{10} + x^3 + x^2 + x + 1 \quad (3)$$

$$m_5(x) = x^{10} + x^8 + x^3 + x^2 + 1 \quad (4)$$

Os polinômios geradores são dados pelas Eq. 5 e 6:

$$g_h(x) = m_1(x)m_3(x)m_5(x)(x^2 + 1) \quad (5)$$

$$g_s(x) = x^{30}m_1(x^{-1})m_3(x^{-1})m_5(x^{-1})(x^2 + x + 1) \quad (6)$$

Os polinômios $m_1(x)$, $m_3(x)$ e $m_5(x)$ são polinômios mínimos padrões usados para o código BCH(1023,993) e os termos (x^2+1) e (x^2+x+1) são extensões para prover proteção adicional, reduzindo a probabilidade de detecção falsa de erro.

A implementação dos codificadores e dos decodificadores EFEC I.9 7% e 20% foram realizadas em linguagem VHDL (*VHSIC Hardware Description Language*), seguindo as arquiteturas mostradas nas Fig. 1 e 2.

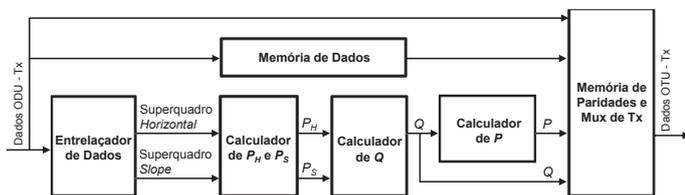


Fig. 1. Diagrama de Blocos dos Codificadores EFEC Implementados.

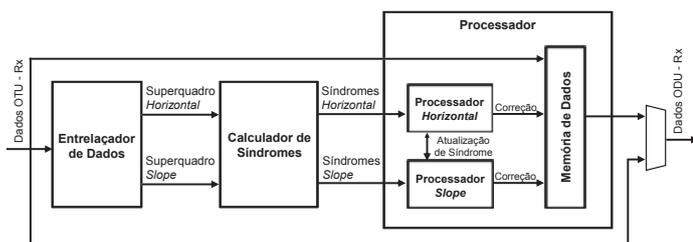


Fig. 2. Diagrama de Blocos dos Decodificadores EFEC Implementados.

A. O Codificador

No Entrelaçador de Dados, primeiro sub-bloco do Codificador, o fluxo de dados é organizado para formar duas estruturas de dados chamadas de superquadros *Horizontal* e *Slope*, ambas com dimensões iguais a 512 linhas por 1024 colunas, nas quais cada linha constitui uma palavra-código. Nas 4 colunas mais à esquerda, bits de enchimento iguais a zero são acomodados. Nas demais colunas, os mesmos dados de 4 ODU's (*Optical Data Unit*) serão acomodados coluna por coluna para formar um superquadro *Horizontal* e um *Slope* (segundo as regras de formação mostradas na Seção I.9.2.3 da G.975.1), de forma que cada *bit* seja protegido duas vezes. O superquadro *Horizontal* é ilustrado pela Fig. 3.

No segundo sub-bloco do codificador, o polinômio $g_h(x)$ é usado para codificar as palavras-código horizontais, enquanto o polinômio $g_s(x)$ é usado para codificar as palavras-código de *Slope*. Essa operação é feita obtendo-se o resto da divisão do polinômio mensagem pelo polinômio gerador. Como resultado, obtém-se as chamadas Paridades Intermediárias P_H para o código *Horizontal* e P_S para o código *Slope*.

Dois palavras-código (uma de *Horizontal* e outra de *Slope*) compartilham 64 bits do campo de paridade: 32 bits correspondentes à Paridade Extra (Q) e 32 bits correspondentes à Paridade Final (P). Isso somado ao fato de que os dados de ambos os superquadros são os mesmos, permite que apenas o superquadro *Horizontal* seja transmitido. No terceiro sub-bloco do codificador, a Eq. 7 é usada para o cálculo das Paridades Extras de tal forma que a Paridade Final, também de 32 bits, torna-se válida para as palavras-código de *Horizontal* e *Slope*. Os termos M_H e M_S são operadores lineares.

$$Q = (T \cdot M_H - M_S \cdot T)^{-1} (P_S - T \cdot P_H) \quad (7)$$

No quarto bloco a Paridade Final de cada palavra-código é calculada através da Eq. 8, enquanto o quinto bloco armazena as Paridades Extra e Final para que elas possam ser mapeadas no campo de FEC quadro OTU (*Optical Transport Unit*) e transmitidas alternadamente com os dados que foram armazenados no sub-bloco Memória de Dados.

$$P = M_H \cdot Q + P_H \quad (8)$$

B. O Decodificador

O primeiro sub-bloco do Decodificador é o Entrelaçador de Dados, cuja função é organizar os dados recebidos no formato dos superquadros *Horizontal* e *Slope*. Após o entrelaçamento, os dados seguem para o segundo sub-bloco denominado Calculador de Síndromes, onde são calculados os valores de 10 bits para as síndromes $S1$, $S3$ e $S5$ (para cada palavra-código) e também valores de 2 bits para a proteção contra detecção falsa de erro mencionada na Seção I. As síndromes correspondentes às palavras-código de *Horizontal* e *Slope* são calculadas usando multiplicadores de Horner [7] no campo de Galois, que é um algoritmo para avaliação eficiente de polinômios como as palavras-código BCH.

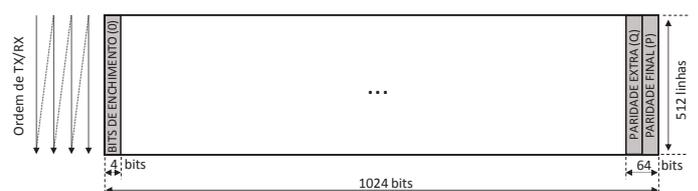


Fig. 3. Superquadro *Horizontal* EFEC I.9.

No terceiro sub-bloco do Decodificador, o Processador, é realizado o armazenamento e o processamento das síndromes geradas no sub-bloco anterior. Para possibilitar um melhor desempenho nessa etapa da decodificação, o processamento passa a ser feito não mais no fluxo de dados *bit a bit* e sim em bancos de síndromes dedicados, localizados dentro dos Processadores *Horizontal* e *Slope*.

No EFEC I.9 a eficiência de correção de erros é potencializada ao executar-se o processo de decodificação *Horizontal* e *Slope* de forma iterativa por várias vezes. Durante o processo *Horizontal*, ao se processar uma palavra-código na qual foram localizados mais erros do que o limite de correção do BCH(1020,988) (3 bits), nenhuma correção é executada e o processamento passa ao Banco de Síndromes *Slope*. Com isso, outros bits errados serão corrigidos e, caso um ou mais desses bits pertença também à palavra-código de *Horizontal* citada, ela poderá não ter mais erros acima do limite e, portanto, será possível corrigir todos os erros contidos na mesma. Sempre quando o processador atualizar a correção de um erro no banco de síndromes, seja ele *Horizontal* ou *Slope*, os bancos de síndromes local e oposto serão atualizados para que as síndromes não mais reflitam a existência de um *bit* que estava errado mas foi corrigido.

Dentro do bloco Processador existe o sub-bloco Memória de Dados, cuja função é armazenar o quadro OTU e receber acessos do processador toda vez que for necessário realizar uma correção nos dados. É desse sub-bloco que partirão os dados que serão transmitidos quando o EFEC estiver em operação.

III. CÓDIGO EFEC I.9 MODIFICADO PARA OPERAR COM 20% DE REDUNDÂNCIA

O EFEC I.9 com 20% de redundância é uma versão alterada do EFEC I.9 proposto em [1], cujo objetivo é aumentar o ganho de codificação. A principal alteração consiste no fato de que a área de informação do superquadro BCH *Horizontal* original é dividida em três, formando-se três palavras-código a partir da informação contida em cada palavra-código do superquadro original. A Fig. 4 ilustra essa divisão, que ocorre também para o superquadro BCH de *Slope*.

Para o cálculo da paridade, os bits restantes das palavras-código são completados com zeros, o que não afeta o resultado final da paridade calculada e permite que o mesmo *hardware* e arquitetura usados na implementação do EFEC I.9 7% sejam aproveitados. Como consequência, o quadro OTU passa a conter o triplo de redundância sendo, portanto, estendido para quatro linhas de 4592 bytes, conforme mostrado na Fig. 5.

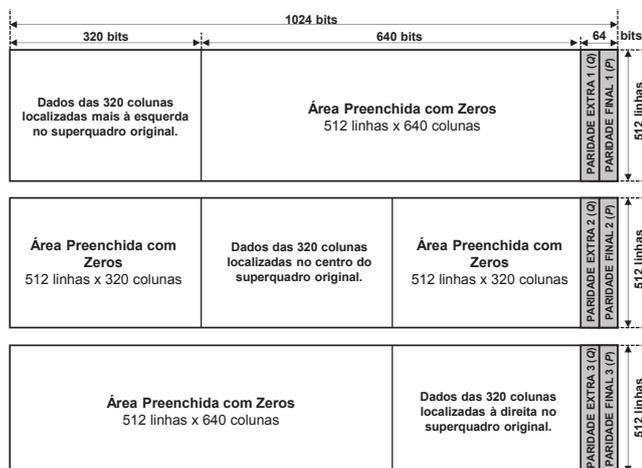


Fig. 4. Superquadros *Horizontais* do EFEC I.9 20%.

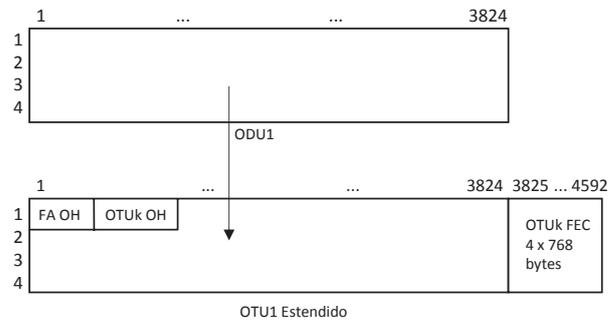


Fig. 5. Frames ODU1 e OTU1 Estendido.

IV. ARRANJO EXPERIMENTAL

A. O Ambiente de Testes

Para que fosse possível testar e colher resultados das implementações dos EFECs explanadas nas Seções II e III deste artigo, um ambiente de testes conforme ilustra o diagrama da Fig. 6 foi montado.

Os códigos foram compilados, sintetizados e gravados em um componente de lógica programável FPGA do fabricante Altera, modelo Arria II GX 2AGX260N, que equipa o *kit* de desenvolvimento da mesma marca, modelo DK-DEV-2AGX260N. Para tais atividades, o *software* Altera Quartus II 64-*Bit*, versão 13.0.1 SP1 foi utilizado. Uma interface de linha de comando ou CLI (*Command-Line Interface*) foi desenvolvida em linguagem C++ e gravada em um processador Freescale Coldfire 52259, que equipa o *kit* da Axiom Manufacturing, modelo CMM-52259 AXM-0515. Através do referido *kit* e de um microcomputador, foi possível monitorar e configurar os EFECs por intermédio de registros localizados nos sub-blocos de controle. Foi utilizado também um sintetizador de relógio Silicon Labs Si5338 que equipa o *kit* do mesmo fabricante, modelo Si5338-EVB, para gerar relógios de referência para dois PLLs (*Phase Locked Loop*) também da Silicon Labs, modelo Si5326, que equipam dois *kits* Si5325/26-EVB.

O sinal cliente foi gerado em meio óptico (fibra óptica) pelo equipamento da marca Anritsu, modelo MP1570A, que gerou uma sequência pseudoaleatória ou PRBS (*Pseudorandom Binary Sequence*). Para o teste do EFEC I.9 7%, os dados foram gerados a uma taxa de 2,66 Gbps, enquanto que para os testes do EFEC I.9 20%, a taxa era de 3,01 Gbps. A taxa de sinal cliente, portanto, era de 2,488 Gbps para ambos os casos. O sinal gerado era convertido para o meio elétrico por meio de um módulo óptico aplicado no *kit* de desenvolvimento Altera.

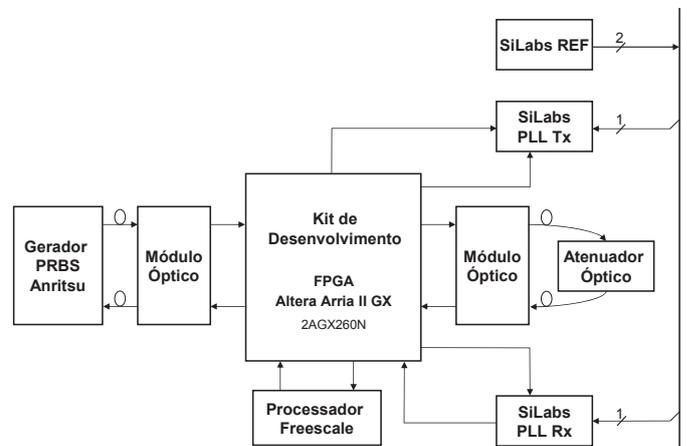


Fig. 6. Diagrama de Blocos do Ambiente de Testes.

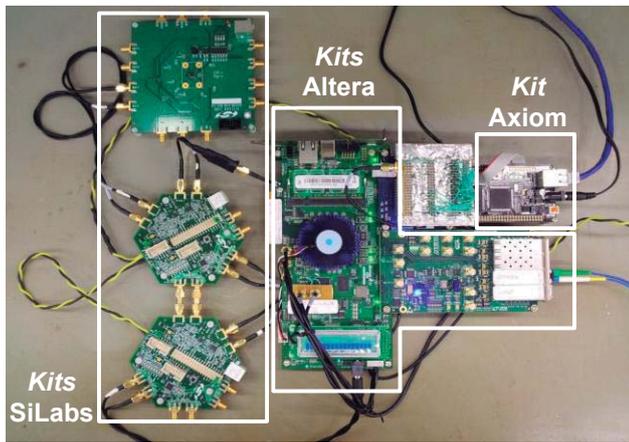


Fig. 7. Montagem dos Kits para Teste dos EFECs.

A Fig. 8 é um diagrama de blocos que mostra o que foi implementado dentro do FPGA. Como se pode notar, outros blocos foram desenvolvidos em VHDL para tratar o fluxo de dados e dar suporte aos testes. Entretanto, tais blocos não são o foco deste trabalho e não serão abordados com detalhes.

Os dados são inseridos no FPGA serialmente, sendo o *Transceiver* de Cliente responsável por paralelizar esse fluxo, formando um barramento de 16 bits de largura. Em seguida, um sub-módulo executa o tratamento do cabeçalho (OH – *Overhead*) do quadro OTU1 e o mapeamento em quadro ODU1. Nesse processo, o sinal de cliente não é desencapsulado, sendo somente amostrado em taxa de *bit* constante ou CBR (*Constant Bitrate*) e mapeado diretamente no quadro, segundo as recomendações do ITU-T G.709 [8] e G.798 [9].

O próximo bloco no fluxo de transmissão é o Codificador EFEC, que processa os dados de cliente, calcula as paridades e gera o quadro OTU1, preenchendo a área destinada a FEC. O bloco Embaralhador (*Scrambler*) embaralha os dados segundo a Seção 11.2 do documento G.709 [8]. Após passar pelo *Transceiver* de Rede, os dados estão em formato serial e prontos para serem tratados pelo Módulo Óptico, onde o sinal é adaptado para trafegar pelo *loop* de fibra óptica que simula a transmissão pela rede. Um Atenuador Óptico, cuja função será comentada posteriormente, também foi inserido no referido *loop*.

Depois de passar pelo *loop* óptico, os dados entram novamente no FPGA e são paralelizados pelo *Transceiver* de Rede, iniciando o fluxo de recepção. Em seguida, é a vez do Alinhador de Quadro tratar os dados, encontrando a sequência de alinhamento e alinhando o quadro segundo descrito na Seção 8.2 da Recomendação G.798 [9]. Após esse procedimento, o Desembaralhador desfaz o embaralhamento realizado pelo Embaralhador antes da transmissão e os dados estão prontos para serem processados pelo Decodificador EFEC.

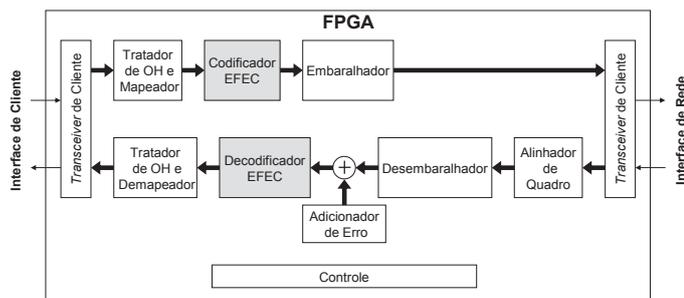


Fig. 8. Estrutura do Projeto Dentro do FPGA.

Quando o Decodificador EFEC está configurado como habilitado, os processos de detecção e correção de erros são realizados e os dados corrigidos encaminhados ao Demapeador que, por sua vez, retira a área de dados destinada a FEC do quadro OTU1 e encaminha ao *Transceiver* de Cliente para que os dados sejam serializados novamente.

B. A Estratégia de Testes

Para análise de desempenho dos EFECs, um bloco Adicionador de Erro foi implementado em paralelo com o fluxo de dados de recepção, entre os blocos Desembaralhador e Decodificador EFEC. Por meio de acessos de escrita do processador em registros do bloco Controle, o Adicionador de Erro é capaz de gerar taxas de erro de entrada programáveis desde 10^{-3} até 10×10^{-3} , variando em passos de 10^{-3} . Os erros são produzidos invertendo-se bits recebidos, simulando a ocorrência de problemas como ruídos, distorções, atenuações e outros fenômenos que degradam a informação durante os processos de transmissão, propagação pelo canal óptico e recepção. Um Atenuador Óptico do fabricante Hewlett-Packard, modelo 8156A, foi inserido no *loop* óptico e usado para aferir o Adicionador de Erro implementado, levando-o a refletir o comportamento do Atenuador Óptico real.

O processo de medição é realizado mediante o levantamento da curva de taxa de erro de entrada (BER_{in} ou B_{in}) versus taxa de erro de saída (BER_{out} ou B_{ref}), onde BER_{in} corresponde ao valor com o EFEC desligado e BER_{out} corresponde ao valor com o EFEC ligado. Não foi possível fazer o levantamento por completo da curva BER_{in} versus BER_{out} por meio de degradação do sinal óptico pelo fato de que o equipamento Anritsu não mede taxas de erro de entrada acima de 4×10^{-3} . Dado o elevado desempenho deste EFEC é necessário injetar, de forma controlada e conhecida, taxas de erro de até 10×10^{-3} na entrada do Decodificador. Como o bloco Adicionador de Erro permite inserir erros da forma mencionada anteriormente, o mesmo foi utilizado para degradar o sinal recebido.

O cálculo do Ganho Líquido de Codificação (NCG – *Net Coding Gain*) é baseado na BER_{in} versus BER_{out} , seguindo a equação que está presente na Seção 7.1.3 da Recomendação G.975.1 [1], representada neste documento pela Eq. 9. Usando tal equação, para valores de BER_{in} (B_{in}) e BER_{out} (B_{ref}) iguais a $6,5 \times 10^{-3}$ e 10^{-15} respectivamente, chega-se a um valor de NCG igual a 9,3 dB para o EFEC I.9 20%.

$$NCG = 20 \log_{10} [erfc^{-1}(2B_{ref})] - 20 \log_{10} [erfc^{-1}(2B_{in})] + 10 \log_{10} R \quad (9)$$

V. RESULTADOS

A Fig. 9 mostra um gráfico com as curvas BER_{in} versus BER_{out} para os códigos: GFEC RS(255,239) (curva identificada pelo número 1), EFEC I.9 7% (curva identificada pelo número 2) e EFEC I.9 20% (curva identificada pelo número 3). As duas primeiras foram retiradas da Recomendação G.975.1 [1] Fig. I.38 e são de referência. A curva do EFEC I.9 20% foi levantada segundo o procedimento de medição mencionado na subseção anterior. Configurando-se o bloco Adicionador de Erro para gerar valores de BER_{in} entre 10^{-3} e 6×10^{-3} , a BER_{out} é menor que 10^{-16} . Para uma BER_{in} igual a 7×10^{-3} a BER_{out} medida é 2×10^{-13} . Com uma BER_{in} de 8×10^{-3} o valor medido para a BER_{out} é de 2×10^{-12} , enquanto que com uma BER_{in} de 9×10^{-3} a BER_{out} medida é de $3,5 \times 10^{-12}$. Finalmente, configurando-se o valor mais alto de BER_{in} que o Adicionador de Erro implementado é capaz de gerar, ou seja, 10×10^{-3} ou 10^{-2} , a BER_{out} medida é igual a $4,7 \times 10^{-12}$, completando os quatro pontos medidos no teste. Tais pontos foram usados para traçar a curva 3 da Fig. 9 e são mostrados com um “x” sobre ela.

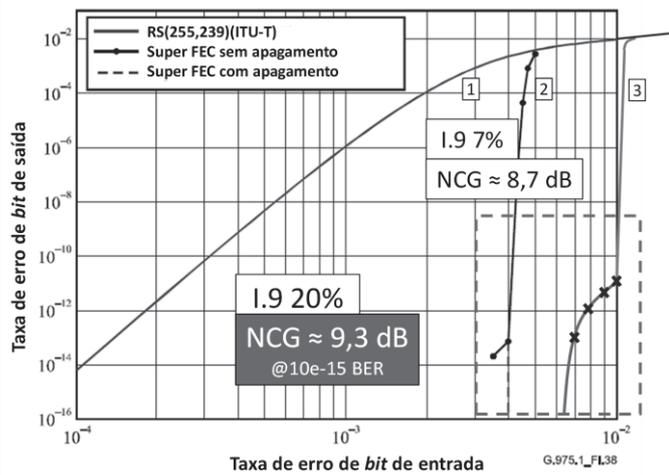


Fig. 9. Curva de Ganho BER_{in} versus BER_{out} dos EFECs.

Para efeito de obtenção dos dados de utilização de recursos de hardware no FPGA, a síntese dos códigos que correspondem ao Codificador e o Decodificador das duas implementações dos EFECs foi realizada individualmente com o software Quartus e os resultados estão expostos nas Tabelas I e II. O termo “ALUTs Comb” corresponde à quantidade do recurso *Adaptative Look-Up Tables* combinacionais utilizado, enquanto a sigla BMB corresponde à quantidade do recurso *Block Memory Bits* utilizado.

TABELA I. DADOS DE DESEMPENHO E OCUPAÇÃO DE RECURSOS PARA O EFEC 1.9 7%.

Parâmetro		Codificador	Decodificador
Utilização de Recursos	Registros	10513	6386
	BMB	1251328	1391104
	ALUTs Comb.	6,224	8042
Desempenho	Freq.Máx.	275,63 MHz	226,96 MHz

TABELA II. DADOS DE DESEMPENHO E OCUPAÇÃO DE RECURSOS PARA O EFEC 1.9 20%.

Parâmetro		Codificador	Decodificador
Utilização de Recursos	Registros	13506	15692
	BMB	1482752	1862627
	ALUTs Comb.	7021	19493
Desempenho	Freq.Máx.	279,33 MHz	230,36 MHz

VI. CONCLUSÃO

Baseado nos dados de duas implementações em FPGA de códigos corretores de erro BCH entrelaçados, o presente trabalho mostrou que é possível partir de uma implementação de código BCH com um determinado ganho para gerar uma

segunda implementação com capacidade ampliada, bastando para isso dividir os dados das palavras-código originais em múltiplas palavras-código complementares (três no caso deste trabalho) e também aumentando-se a quantidade de informação redundante. Essa estratégia permite que, mesmo que a palavra-código tenha sido dividida em três partes, cada uma dessas partes seja processada da mesma forma que uma palavra-código integral, simplificando o projeto pelo fato de ser possível utilizar o mesmo circuito base para implementar dois códigos BCH com diferentes capacidades de correção de erro.

Comparando-se a implementação do EFEC original com a do EFEC modificado, destaca-se como principal vantagem do segundo ante o primeiro os acréscimos nos valores de NCG de 0,04 dB para BER_{out} igual a 10⁻¹³ e de 0,6 dB para BER_{out} igual a 10⁻¹⁵, o que pode ser considerado expressivo para aplicações de transmissão de longa distância. Em contrapartida, as desvantagens que merecem destaque são o maior consumo de banda (causada pela maior taxa de transmissão) e um maior consumo de recursos do FPGA.

A evolução deste trabalho se dará através do reprojeto dos códigos aqui apresentados, aproveitando-se as arquiteturas e circuitos desenvolvidos, porém com modificações para aumentar a capacidade de processamento de dados em paralelo, de modo que seja possível operar em taxas de transmissão bem mais elevadas. O intuito é atingir taxas de transmissão para atender à unidade de transporte óptico OTU4 (em torno de 100 Gbps), que é o padrão de maior velocidade usado atualmente.

AGRADECIMENTOS

Os autores agradecem à Fundação CPqD, à Padtec S.A., à Zeuxion e também aos demais colegas que trabalharam neste projeto: Marcelo Marcos Polidoro, Giuliano Ferronato, Rodolfo Soares Caproni e Antônio Unias de Lucena.

REFERÊNCIAS

- [1] ITU-T, “Forward Error Correction for High Bit-Rate DWDM Submarine Systems”, Recomendação G.975.1/Y.1331, Fev 2004.
- [2] R. Bernardo, T. R. Tronco, C. L. A. Lessa e E. Mobilon, “OTN: A Evolução das Redes de Transporte”, Cad. CPqD Tecnologia, v. 7, n. 2, p. 75-86, Jul/Dez 2011.
- [3] Tronco, T. R., *Redes de Nova Geração*, 1ª edição, São Paulo: Editora Érica, 2006.
- [4] ITU-T, “Forward Error Correction for Submarine Systems”, Recomendação G.975, Out 2000.
- [5] Haykin, S., *Communication Systems*, 4ª edição, John Wiley and Sons, 2001.
- [6] C. Xie, Y. Zhao, Z. Xiao, D. Chang, F. Yu, “FEC for high-speed optical transmission”. China, Nov 2011.
- [7] Mayer-Lindenberg, F., *Dedicated Digital Processors: Methods in Hardware/Software Co-Design*, John Wiley and Sons, Fev 2004.
- [8] ITU-T, “Interfaces for the Optical Transport Network (OTN)”, Recomendação G.709/Y.1331, Dez 2009.
- [9] ITU-T, “Characteristics of Optical Transport Network Hierarchy Equipment Functional Blocks”, Recomendação G.798, Jan 2002.

Diego Brito de Carvalho, Arley Henrique Salvador, Érico Nunes Ferreira Bastos e Luiz Juberto Rossi de Jesus, Vice Presidência de Pesquisa e Desenvolvimento, Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações – Fundação CPqD, Campinas-SP, Brasil, E-mails: {dbrito, arleys, ebastos, ljuberto}@cpqd.com.br. Max H. M. Costa, Departamento de Comunicações, Universidade de Campinas – Unicamp, Campinas-SP, Brasil, E-mail: max@decom.fee.unicamp.br. Este projeto foi financiado pela Lei da Informática e pela Padtec S.A.