



## Análise de Robustez e Implementação da Codificação 8b/10b

Victor Afonso dos Reis<sup>1</sup>

Faculdade de Engenharia de Ilha Solteira, UNESP, Ilha Solteira, SP,

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

Lucas Arruda Ramalho<sup>2</sup>

Faculdade de Engenharia de Ilha Solteira, UNESP, Ilha Solteira, SP,

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

Ailton Akira Shinoda<sup>3</sup>

Faculdade de Engenharia de Ilha Solteira, UNESP, Ilha Solteira, SP,

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

**Resumo.** Em comunicações de altas frequências, onde a confiabilidade é relevante, recomenda-se a codificação 8b/10b. Neste tipo de transmissão, pode ocorrer uma série de problemas. Um problema de extrema relevância em comunicações digitais é o sincronismo entre transmissor e receptor, que pode ser prejudicado por uma longa sequência de níveis lógicos, zeros (0's) ou um (1's), no canal de transmissão. Neste projeto modela-se e implementa-se um sistema da codificação 8b/10b, que soluciona o desbalanceamento de bits, no kit de desenvolvimento em FPGA da Xilinx(Kintex 7), através da linguagem de descrição de hardware VHDL pelo software Vivado<sup>TM</sup>. Pela implementação obteve-se um sistema genérico para o teste de codificações descritas em VHDL, além de parâmetros de frequência, taxa de dados e características do funcionamento da transmissão usando o sistema da codificação 8b/10b.

**Palavras-chave.** VHDL, FPGA, Codificação 8b/10b, Transmissões, Implementação.

## 1 Introdução

A complexidade dos sistemas de comunicação e a sua velocidade evoluíram ao longo do tempo, porém um modelo elementar é ilustrado na Figura 1. Em qualquer meio de transmissão, os dados transmitidos podem sofrer distorções ou interferências causadas por problemas no canal de transmissão [1].

Diversos recursos tecnológicos trocam dados em pequenas ou longas distâncias nos dias atuais. Uma alta confiabilidade dos dados na transmissão, é garantido obtendo no lado do

---

<sup>1</sup>victor.afonsoreis35@gmail.com

<sup>2</sup>lucasarrudaramalho@gmail.com

<sup>3</sup>shinoda@dee.feis.unesp.br

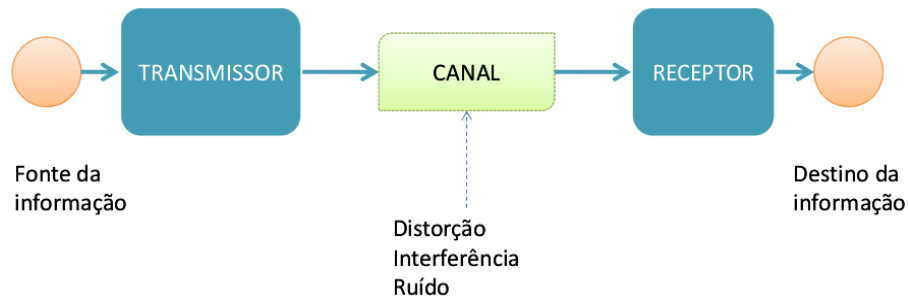


Figura 1: Modelo de um Sistema de Comunicação Elementar.

receptor os mesmos ou a maioria dos dados enviados pelo transmissor. Esta preocupação ocorre devido à problemas comuns em qualquer meio de transmissão, como por exemplo: a atenuação do sinal, dessincronização entre o transmissor e o receptor e ruídos apresentados no canal de transmissão [2]. Ao trabalhar com velocidades maiores os problemas tornam-se mais evidentes, principalmente a dessincronização entre o transmissor/receptor e a aparição de dados ruidosos. Esses dois tipos de problemas são um dos fatores que limitam velocidade de um canal de transmissão [3].

Desde a década de 40 são feitos estudos para obter comunicações mais confiáveis e com um menor custo, surgindo teorias e circuitos que realizam codificações para a transmissão de dados. Estas representam uma conversão de dados digital/digital e são separadas em três grandes técnicas: codificação de linha, codificação em blocos e *scrambling*. Neste trabalho será trabalhado com a codificação em blocos em que transforma um bloco com determinado número "m" de bits em outro bloco com "n" número de bits, sendo "n" maior que "m" [4].

Esta redundância na codificação em blocos é feita de forma lógica, dessa maneira pode-se aumentar a detecção de erros na transmissão. O uso da codificação de blocos aumenta a qualidade do sistema de transmissão, apesar de diminuir a taxa efetiva de dados no sistema [5]. A codificação em blocos gera dados com o máximo de transições possíveis, retirando muitas sequências de dados repetidos no canal. Logo, é possível facilitar a sincronização entre o emissor e receptor por meio de circuitos externos. Codificações como por exemplo a 4b/5b, a 6b/8b, a 8b/10b, a 64b/66b e a 128b/130b utilizam esse método [6].

Em sistemas de comunicação há a problemática da escolha da melhor codificação para a aplicação alvo. Algumas codificações fornecem mais confiabilidade na transmissão do que outras para o mesmo sistema. Dessa maneira ao implementar uma codificação testa-se a mesma para obter a melhora do sistema de comunicação. Em caso de erros de transmissão no canal, a decodificação da mensagem pode ou não detectar os dados como errôneos. Logo, a robustez da codificação é avaliada considerando a capacidade de distinguir com êxito os dados corretos e incorretos da transmissão. Para sistema de comunicação de altas velocidades é recomendada a utilização da codificação 8b/10b, independente do meio de transmissão [7]. Sendo assim, o objetivo deste trabalho é o estudo da robustez desta codificação 8b/10b obtendo a taxa de erro apresentada no lado do receptor. Esta análise

refere-se à um estudo realizado por meio do MATLAB<sup>TM</sup>, implementando a codificação em software e testando a taxa de erros por meio de recursos do SIMULINK.

Para o teste da codificação 8b/10b em ambiente real, deve-se implementá-la em hardwares e realizar transmissões entre um transmissor e um receptor. Por isso, a codificação 8b/10b foi descrita em linguagem de descrição de hardware VHDL e implementada no FPGA através do kit da Xilinx (kintex 7 KC705 Evaluation Kit).

Pela análise parcial do estudo feito no MATLAB<sup>TM</sup> (SIMULINK) obteve-se uma alta taxa de detecção de erros na transmissão quando comparada com a taxa de erros total. A seção 2 ressalta a teoria da codificação 8b/10b. A seção 3 apresenta a simulação no software MATLAB<sup>TM</sup> (SIMULINK) do estudo da robustez da codificação 8b/10b. A seção 4 descreve o sistema da codificação 8b/10b, implementada na linguagem de descrição de hardware VHDL. A seção 5 conclui os resultados do estudo e da implementação da codificação em VHDL.

## 2 Codificação 8b/10b

A codificação foi descrita para se adequar à locais com transmissões em alta velocidades, promovendo sistemas de baixo custo e com transmissões confiáveis. Pela descrição, a codificação promove um balanceamento DC no sinal, ou seja, o dado a ser transmitido não possui níveis lógicos altos ou baixos por muito tempo. Esse balanço torna-se importante para a recuperação do relógio e consequentemente sincronização entre o emissor e o receptor.

Os dados de 8 bits são codificados em 10 bits de forma que possua o maior número de transições possíveis. Neste mapeamento, nota-se que alguns dados de 10 bits não possuem equivalentes em dados de 8 bits. Dessa maneira, usa-se alguns dados de 10 bits com o mesmo número de bits 1's e 0's como dados de controle da transmissão [8].

Na codificação é descrito o conceito de disparidade, apresentando dois tipos: a disparidade do sistema, ou do inglês *Running Disparity* (RD), e a disparidade dos dados. A codificação é descrita de forma que os dados transformados em 10 bits só possuam disparidade de dados de +2, -2 ou 0 (nula). Desta maneira, a disparidade dos dados é obtida no momento em que se codifica o dado de entrada para 10 bits. A disparidade do sistema, ou RD, é obtida seguindo regras de acordo com a disparidade dos dados e também é responsável pela escolha dos dados no momento da codificação [7].

O RD alterna-se entre +1 e -1 ou permanece o mesmo, de acordo com as regras ilustradas na Figura 2. Se o RD é -1 e a disparidade dos dados de 10 bits for +2, o RD troca para +1. Caso o RD for +1 e a disparidade dos dados de 10 bits for -2, logo o RD troca para -1. Qualquer outra situação não pertencente a esses casos o RD permanece o mesmo.

A característica da disparidade de dados na transmissão, possuindo um número definido da diferença de bits 1's e 0's, possibilita a detecção de erro pelo receptor e uma possível recuperação do dado por meio de circuitos adicionais. Como descrito, para codificar os dados usa-se o RD que sempre será -1 quando o sistema for inicializado. [8].

A codificação dos dados é feita separando os dados de 8 bits em duas partes. Um

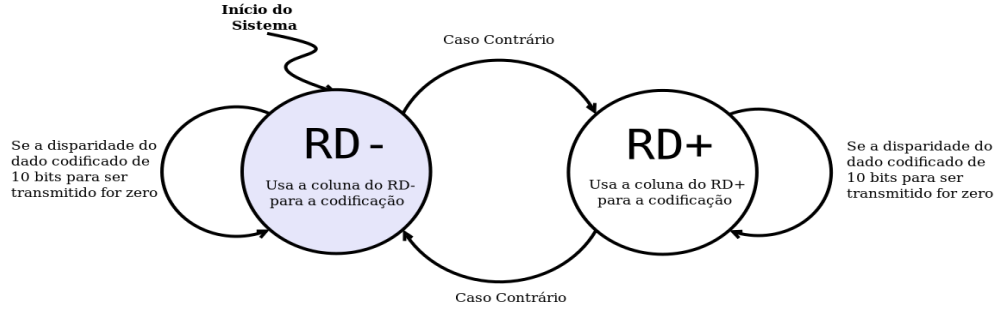


Figura 2: Máquina de estados para a transição do RD.

esquema dessa separação da codificação pode ser visualizado na Figura 3.

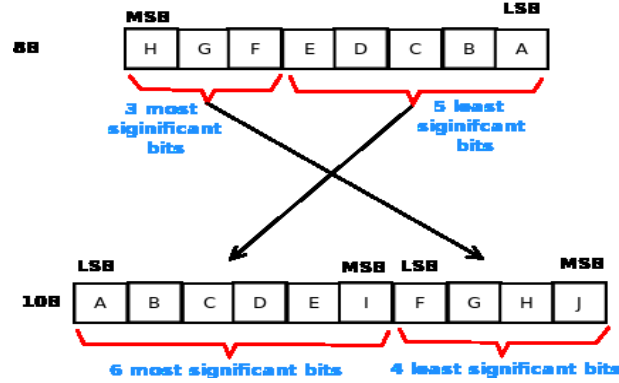


Figura 3: Esquema da separação para codificação 8b/10b..

Pela Figura 3 observa-se que o dado de entrada da codificação é dividido em dois blocos. O primeiro bloco, que possui 3 bits mais significativo dos dados de entrada, ao passar pelo *encoder* se torna um bloco de 4 bits menos significativos da saída. O segundo bloco de entrada, que possui 5 bits menos significativos, se torna um bloco de 6 bits mais significativos na saída do *encoder*. Totalizando um dado de saída de 10 bits, pela junção das partes mais significativa e menos significativa codificadas.

### 3 Descrição e Modelagem em Máquinas de Estado em VHDL do Sistema da Codificação 8b/10b

O estudo da robustez da codificação 8b/10b foi realizado em [9], o qual possui um sistema da codificação 8b10b descrito no MATLAB<sup>TM</sup> (SIMULINK) e um gráfico comparativo entre a taxa de erro total e a taxa de erro identificada. A modelagem da codificação 8b/10b pode ser feita por meio de máquinas de estados no VHDL. Pela máquina de estados é possível representar o comportamento de um circuito ou programa graficamente. Os diagramas de estados são métodos gráficos que ilustra um sistema em máquinas de estado,

estando presente apenas seus estados e a ação que provocou as transições.

O funcionamento da codificação 8b/10b, descrita em VHDL através de máquinas de estados, é ilustrado na figura 4. Como pode-se analisar há a presença de sete estados no sistema do *encoder* e no *decoder*. No *encoder* o estado "ENC\_A" refere-se ao estado de espera do dado de 8 bits de entrada. O estado "ENC\_B" diz a respeito da codificação dos 3 bits mais significativos. Já o estado "ENC\_C" trata a codificação da parte menos significativa do dado de entrada (5 bits). No estado "ENC\_D" é realizada a junção das partes codificadas, formando a saída do dado de 10 bits. Além disso nesse estado é realizado o cálculo do novo RD do sistema. No estágio "ENC\_E" o RD da palavra é contabilizado e no próximo estágio "ENC\_F" o valor da saída é atualizado. No estágio "ENC\_G" é atualizado o RD do sistema e retorna ao estado de espera "ENC\_A" para receber o novo dado de entrada. Caso em algum momento, ou qualquer estágio, for inserido um sinal de *reset* no sistema, passa-se para o estágio "ENC\_F" que reinicializa as saídas e zera as entradas passando para o estágio "ENC\_A" novamente.

A máquina de estado do *decoder* apresenta comportamento similar, apesar de inverso, com o objetivo de decodificar os blocos do dado codificado. Assim a mesma é capaz de fornecer o dado de 8 bits decodificado e caso no estado "DEC\_D" identifique que houve erro, vai para os estados "DEC\_H", "DEC\_I" e "DEC\_J". Nestes estados a saída vai para alta impedância e um sinal de erro vai para nível lógico alto. Após o estado "DEC\_J" o sistema volta para o estado de espera "DEC\_A" para receber um novo dado de 10 bits.

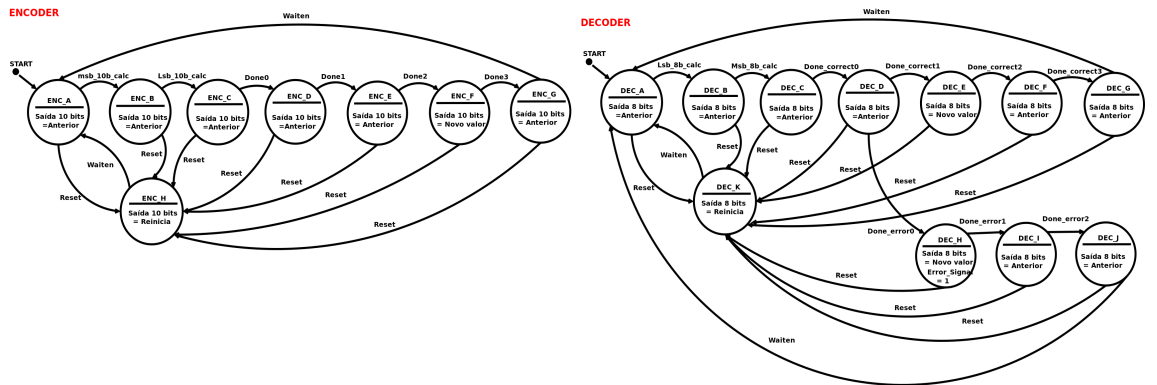


Figura 4: Máquina de estados do sistema implementado a codificação 8b/10b em VHDL.

A modelagem da máquina de estados foi descrita no software Vivado<sup>TM</sup> na linguagem VHDL, assim como o sistema para sua implementação no FPGA. Desta forma, além da descrição do *encoder* e *decoder*, há a necessidade de descrever todos os sistemas periféricos necessários para o funcionamento da codificação. O *encoder* junto com o *decoder* e os demais periféricos são interligados, descritos e referenciados dentro de um arquivo denominado *Top File*. A descrição do sistema para implementação no FPGA é ilustrado na figura 5.

Pela figura 5 observa-se para o funcionamento do sistema deve-se descrever um *Phase-locked Loop* (PLL), um *Virtual Input Output* (VIO), um *Integrated Logical Analiser* (ILA), um contador para inserir dado no *encoder*, um sistema para inserir o erro no sistema

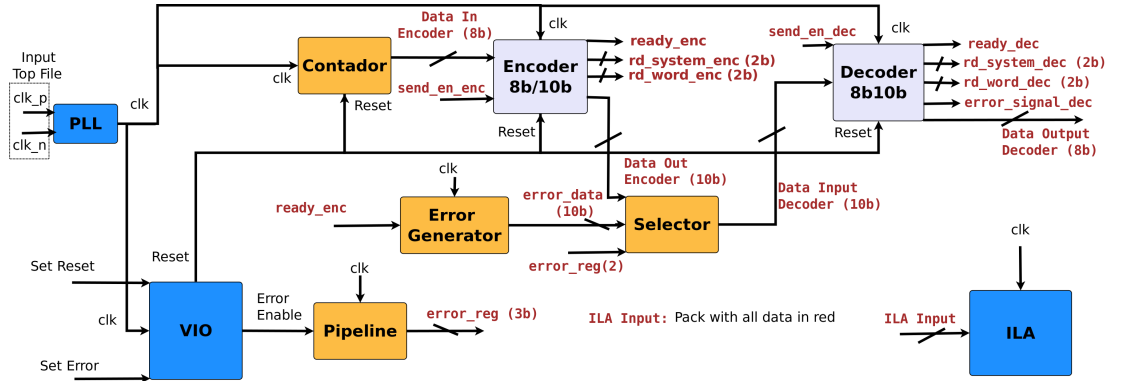


Figura 5: Descrição do Top File da codificação 8b/10b para implementação no FPGA.

descrito pelos componentes VIO/Pipeline e um arranjo para modificar o dado de saída do *encoder* o qual irá entrar no *decoder*.

Através do PLL pode-se inserir qualquer valor de clock que o FPGA suporte implementar. O VIO permite que o usuário set sinais dentro do sistema, como por exemplo o *reset* e o sinal de erro. Como a ação do usuário é muito mais lenta do que o *clock* do sistema, introduz-se um sistema de pipeline para setar o sinal de erro em um ciclo de *clock*. O sistema *Error Generator* gera um vetor de 10 bits, com apenas um bit em nível lógico alto, rotacionando-os quando o sinal de dados disponíveis na saída do *encoder* (*ready\_enc*) for alto. O bloco seletor é responsável por realizar uma operação XOR com o dado de 10 bits do *Error Generator* e o dado de saída do *encoder*, quando o sinal "error\_reg(2)" for alto. Caso contrário o bloco seletor deixa passar o dado de saída do *encoder* puro, ou seja, sem passar pela operação XOR. Pelo ILA é possível visualizar os sinais dentro do sistema, dessa forma todos os dados marcados em vermelho na figura 5 são inseridos no ILA para obter suas formas de onda.

## 4 Implementação e Simulação no Kit FPGA da Xilinx (Kintex 7)

A implementação do sistema foi realizada em um kit de desenvolvimento da Xilinx (Kintex 7), através do software Vivado<sup>TM</sup> o qual gerou toda síntese, implementação (roteamento, análise de *timing*, etc) e programação do FPGA. A frequência de *clock* definida no teste do sistema após a implementação foi de 300 MHz, não sendo a frequência de operação máxima do sistema. Portanto, pode-se enfatizar que o sistema nessa faixa de frequência opera a uma taxa de dados em torno de 2,4 Gb/s. O ILA e VIO dificultam na determinação da frequência máxima, pois ao se elevar a taxa de *clock* estes sistemas não suportaram. É impossível retirar o VIO pois este é necessário para iniciar a codificação. Dessa forma, foi retirado o ILA e a máxima frequência obtida foi de 400MHz, ou seja, uma taxa de dados transmitidos de 3,2 Gb/s.

Simulando o sistema no FPGA observou-se que a latência de codificação, ou seja,

os ciclos de *clock* até o dado de entrada estar devidamente disponível na saída é de 5 ciclos. Para a decodificação, esta latência é de 4 ciclos de *clock* o qual totaliza 9 ciclos para codificação e decodificação. Porém, o sistema para inserir erro na transmissão utiliza 2 ciclos de *clock* gerando um total de 11 ciclos de *clock* para o dado codificado estar devidamente decodificado na saída.

Na figura 6 é ilustrado a simulação do sistema no kit FPGA da Xilinx. O sinal *Error Input* é gerado e permanece em nível lógico alto até que o dado entre no *decoder*. Portanto gerou-se um erro no dado e 4 ciclos de *clock* depois é gerado um sinal de erro. O RD entre *encoder* e o *decoder* entra em descompasso e só se recupera após 33 ciclos de *clock*, ou seja, perdeu-se 3 palavras na transmissão. Este descompasso do RD entre o *encoder* e *decoder* é característico da codificação. Após alguns testes obteve-se um descompasso máximo de 88 ciclos, ou seja, 8 palavras perdidas na transmissão.

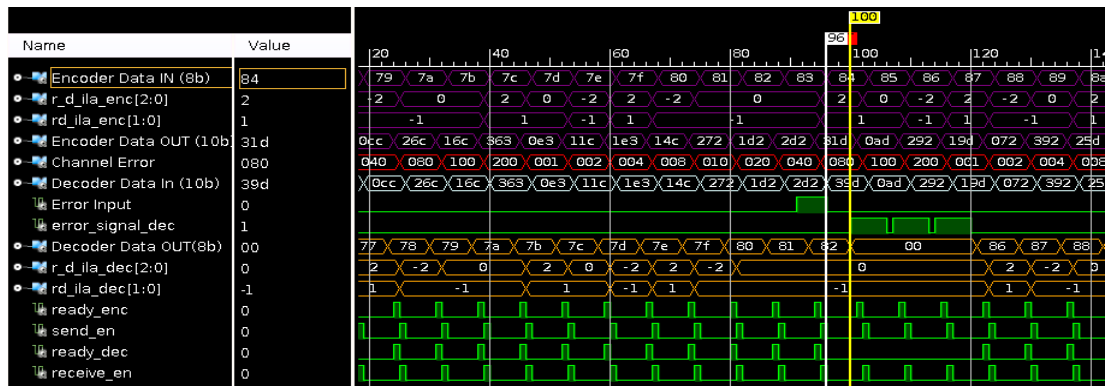


Figura 6: Simulação da codificação 8b/10b implementada no FPGA com erros inseridos no canal.

A latência mínima obtida entre gerar o erro no dado e a identificação de erros pelo *decoder* é de 4 ciclos de *clock*. Porém, pelos testes realizados foi possível inserir um erro em que o dado era decodificado erroneamente e só depois de algumas palavras o erro era identificado. Isto acontece pelo fato de embora o *encoder* e o *decoder* desbalancear o RD, alguns dados são possuem a mesma sequência de bits para qualquer um dos RD's. Dessa forma, pelos testes a latência máxima para encontrar o erro na transmissão foi de 53 ciclos de *clock*. Na figura 7 é ilustrado um caso em que o erro não foi detectado e só posteriormente, pelo desbalanço do RD entre os dois sistemas, foi gerado um sinal de erro.

Pela figura 7 observa-se que o dado em hexadecimal "46" foi inserido e posteriormente saiu o dado também em hexa "56". No ciclo 75 é inserido o dado em hexadecimal "45" e posteriormente sai o mesmo dado, dessa forma mesmo com o sistema com o RD em desbalanço o dado foi decodificado certo. Só alguns ciclos de *clock* posteriores foi gerado um erro por conta do desbalanço dos RD's.

## 5 Conclusão

A utilização de sistema codificadores para comunicação de alta velocidade é vital para sincronismos entre transmissão e recepção, bem como verificação de erros no canal. Exis-

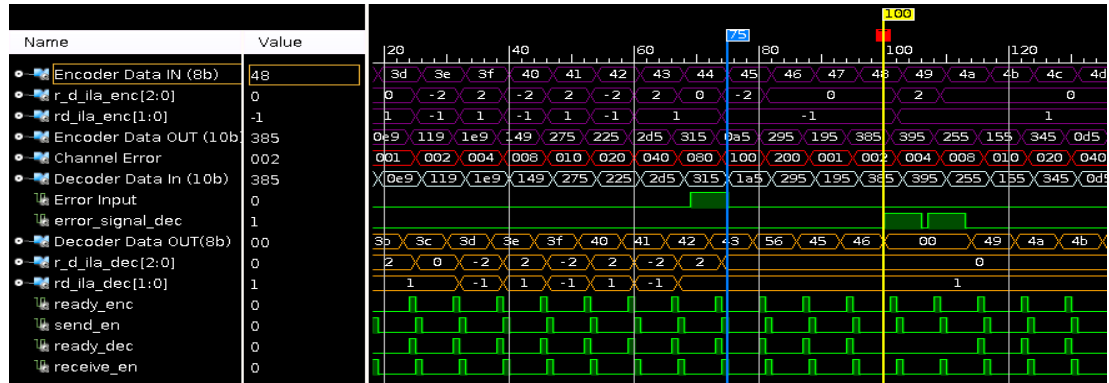


Figura 7: Simulação do sistema da codificação 8b/10b com erro não identificado.

tem diversas técnicas que efetuam essa tarefa de codificação. A escolha entre elas é realizada analisando o custo benefício entre robustez, confiabilidade, ocupação de taxa de dados e largura de banda, e qualidade do canal utilizado.

Este artigo efetuou uma implementação da codificação 8b/10b em um kit de desenvolvimento em FPGA da Xilinx (Kintex 7). Os resultados mostram que a codificação pode apresentar falha ao detectar erros, por conta do desbalanço do RD entre o *encoder* e *decoder*. Entretanto, o sistema da codificação 8b/10b implementada no kit possui uma latência máxima de codificação e decodificação de 11 ciclos de *clock*. A máxima frequência de *clock* obtida foi de 400 MHz, com uma taxa de transferência de 3,2 Gb/s.

O sistema descrito e implementado no FPGA para a codificação 8b/10b, pode ser utilizado como base para a implementação e simulação no FPGA de outras codificações. Logo este trabalho também descreve a modelagem realizada em máquinas de estado para a descrição do sistema com encoder e decoder da codificação 8b/10b em linguagem de descrição de hardware VHDL. Pretende-se em trabalhos futuros, descrever um sistema que realize a análise estatística dos dados, obtendo a robustez do sistema implementado no FPGA.

## Agradecimentos

Os autores agradecem à FUNDUNESP e ao laboratório SPRACE pelo apoio técnico e financeiro obtido ao longo deste trabalho.

## Referências

- [1] F. Borges, Transmissão de Dados, Carnaxide: Schneider Electric, 2008. Disponível em: [http://www.schneiderelectric.pt/documents/product-services/training/transmissao\\_dados.pdf](http://www.schneiderelectric.pt/documents/product-services/training/transmissao_dados.pdf). Acesso em: 1 ago. 2017.
- [2] C. E. Shannon, "Communication In The Presence Of Noise," in Proceedings of the IEEE, vol. 86, no. 2, pp. 447-457, Feb. (1998). DOI: 10.1109/JPROC.1998.659497



- [3] T. D. PRA, Medidor de Taxa de Erro de Bit para Fibra Óptica. 2012. 76 f. TCC (Graduação) - Curso de Engenharia Elétrica, Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, (2012). Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/65425/000858235.pdf?sequence=1>. Acesso em: 1 ago. 2017.
- [4] B. A. FOROUZAN, "Digital Transmission", In: B. A. FOUROUZAN, "Data Communications and Networking", 4. ed, New York: McGraw-hill Education, (2007). Cap. 4. p. 101-118.
- [5] E. Berlekamp, R. Peile and S. Pope, "The application of error control to communications", in IEEE Communications Magazine, vol. 25, no. 4, pp. 44-57, April (1987). DOI: 10.1109/MCOM.1987.1093590.
- [6] B. A. FOROUZAN, "Digital Transmission", In: B. A. FOUROUZAN, "Data Communications and Networking", 4. ed, New York: McGraw-hill Education, (2007). Cap. 10. p. 267-306.
- [7] A. X. Widmer and P. A. Franaszek, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," in IBM Journal of Research and Development, vol. 27, no. 5, pp. 440-451, Sept. 1983. DOI: 10.1147/rd.275.0440
- [8] L. SEMICONDUCTOR. 8B/10B ENCODER/DECODER. Hillsboro: Lattice Semiconductor, (2015).
- [9] V. A. REIS, L. A. RAMALHO, A. A. SHINODA, "Análise de Robustez e Modelagem em Máquina de Estado da Codificação 8b/10b" In: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, "Caderno de Trabalhos Completos e Resumos", Bauru, (2017). p. 48 - 55. Disponível em: [http://www.fc.unesp.br/Home/Departamentos/Matematica/ermac/caderno-ermac\\_2017.pdf](http://www.fc.unesp.br/Home/Departamentos/Matematica/ermac/caderno-ermac_2017.pdf). Acesso em: 9 ago. 2017.