



DINCON 2017

CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES

30 de outubro a 01 de novembro de 2017 – São José do Rio Preto/SP

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

Victor Afonso dos Reis¹

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

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

Lucas Arruda Ramalho²

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

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

Ailton Akira Shinoda³

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

Departamento de Engenharia Elétrica, Unesp, Ilha Solteira, SP

Resumo. Em sistemas com alta taxa de eventos, onde a informação provém de um grande número de sensores e os eventos de interesse são raros, usualmente uma codificação 8b/10b é recomendada para implementação no canal de transmissão. Para uma sequência digital gerada e transmitida em alta velocidade, pode ocorrer uma série de problemas na transmissão. Estes problemas são caracterizados por ruídos devido a radiações, interferências eletromagnéticas, ionizações indesejáveis, atenuações e uma dessincronização entre o transmissor e receptor dada por uma longa sequência de níveis lógicos, zeros (0's) ou um (1's), no canal de transmissão. Esta longa sequência interfere nos circuitos adicionais realizarem a sincronização, sendo necessário um balanceamento nos bits transmitidos. Neste projeto é realizado um estudo da codificação 8b/10b através de um sistema implementado em software Matlab TM dentro do ambiente do Simulink. Além disso, o sistema foi também descrito na linguagem de descrição de hardware VHDL usando o software Vivado TM. Pelo estudo da codificação obteve-se uma alta taxa de detecção de erros no canal de transmissão, possibilitando reduzir drasticamente a porcentagem de erros transmitidos por meio de outros circuitos.

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

1 Introdução

Os sistemas de comunicação tem por objetivo enviar, receber e processar dados por algum meio. Estes meios podem variar para cada sistema (fios metálicos, *wireless*, fibra

¹victor.afonsoreis35@gmail.com

²lucasarrudaramalho@gmail.com

³shinoda@dee.feis.unesp.br

óptica, cabo coaxial), sendo determinados pelo requisitos de cada aplicação. A complexidade e a velocidade deste sistemas evoluíram ao longo do tempo, porém um modelo matemático elementar de um sistema de comunicação é ilustrado na Figura 1. Em qualquer meio de transmissão, os dados transmitidos podem ser adicionados à ruídos gerando problema na transmissão [1].

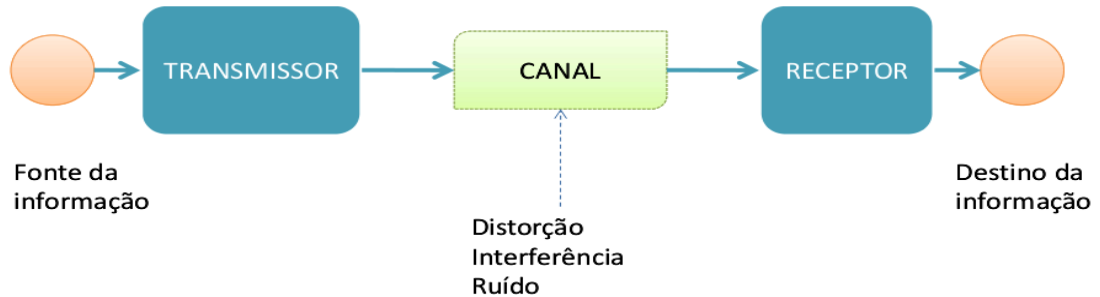


Figura 1: Modelo de um Sistema de Comunicação Elementar.

Diversos recursos tecnológicos trocam dados a pequenas ou longas distâncias nos dias atuais. Uma alta confiabilidade dos dados na transmissão, é garantido obtendo no lado do 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 e o receptor (*jitter*) 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*. A codificação de linha consiste na representação do dado digital a ser transmitido, pela uma forma de onda apropriada para as propriedades físicas específicas do canal de transmissão. A técnica de *scrambling* consiste na substituição de sequências que podem produzir uma tensão constante, gerando um balanceamento DC e capacidade para detectar de erros. Já a codificação em blocos 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 é feita de forma lógica, dessa maneira pode-se detectar erros no dado a ser transmitido e reduzir as taxas 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 MATLABTM, 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 MATLABTM (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 MATLABTM (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.

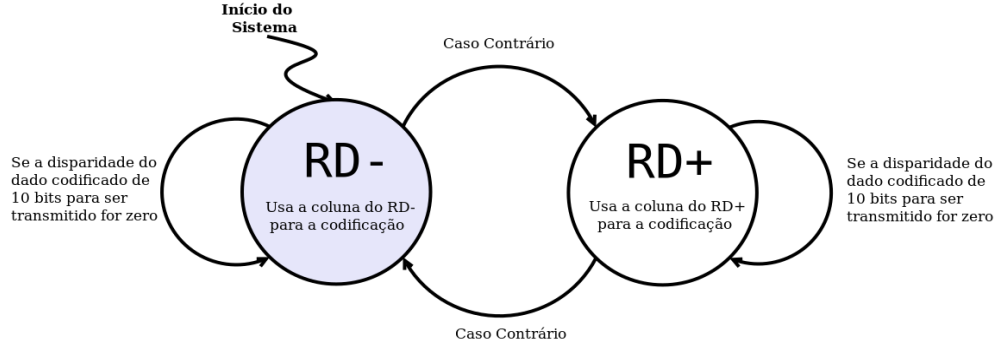


Figura 2: Máquina de estados para a transição do RD.

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

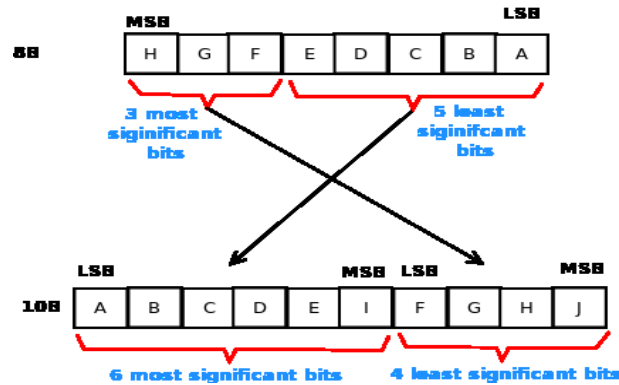


Figura 3: Esquema da separação para codificação 8b/10b..

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 Simulação e Análise da Robustez da codificação 8b/10b

O sistema foi implementado no MatlabTM usando os recursos *Embedded MatlabTM Function Block* (EMFB) do SIMULINK. Nesta ferramenta pode-se descrever o *encoder* e o *decoder* por meio da grande variedade de recursos, além de criar mecanismos para obter as características da codificação. A obtenção destas características provém de inserção de erros na transmissão de dados do *encoder* para o *decoder* e análise do número de erros obtidos pelo número de dados transmitidos. Estes erros inseridos devem ser aleatórios para a obtenção de uma característica que se aproxime da realidade.

Na figura 4 é ilustrado o sistema desenvolvido usando a ferramenta EMFB do SIMULINK. Neste sistema observa-se que o bloco HDL *counter* insere dados binários de 8 bits no sistema. Os subsistemas *Bernoulli Binary Generator* (BBG), *Aleatory Counter* (ALC) e *Multiport Switch* (MPS) são responsáveis por gerarem o erro na transmissão, depois que os dados são codificados para 10 bits no subsistema *Encoder 8b to 10b*. Ou seja, estes subsistemas emulam a probabilidade de erro no canal.

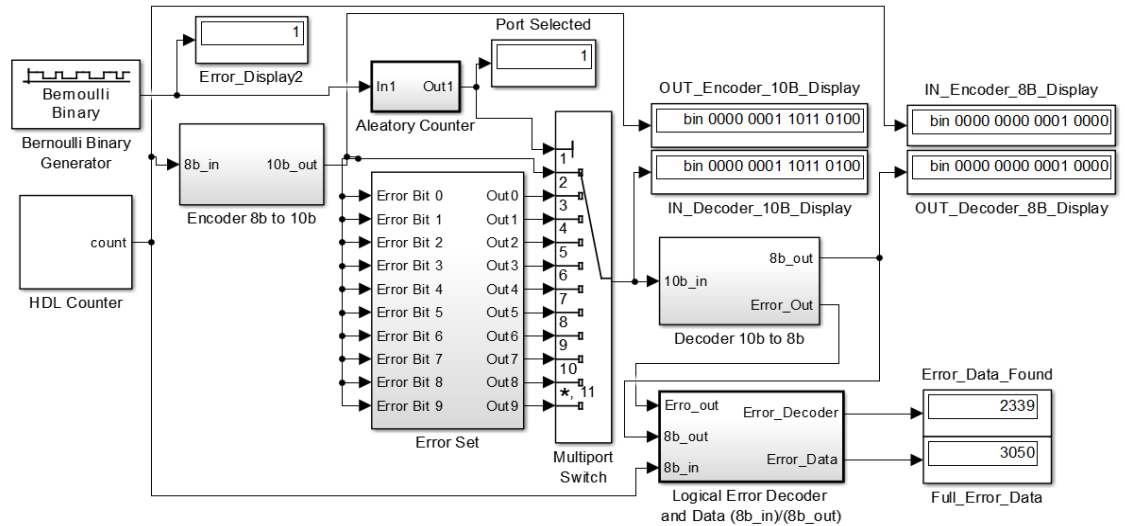


Figura 4: Sistema desenvolvido no MatlabTM usando a EMFB do Simulink.

No subsistema *Error Set* aplica-se o erro, em um único bit no dado transmitido. No subsistema *Decoder 10b to 8b* os dados são decodificados para 8 bits novamente. No subsistema *Logical Error Decoder and Data (8b_in)/(8b_out)* (LEDD) o número de erros do sistema é obtido de duas formas. Primeiramente analisa-se o sinal de saída do *decoder* *Error_out*, identificando que o *decoder* teve sucesso ao classificar o dado como errôneo. A segunda forma de obter o número de erros é a comparação entre os dados de entrada do subsistema *HDL counter* com os dados de saída do *decoder*, identificando que o *decoder* falhou ao detectar o dado como errôneo.

Para observar a proporção de erros em relação ao número de dados transmitidos no sistema, configurou-se no ambiente MATLABTM SIMULINK para simular 10000 pacotes de dados transmitidos ao receptor. Variou-se a probabilidade de erro no canal no bloco

Bernoulli Binary Generator de 0 até 20 por cento, coletando o número de erros pela comparação entre o dado de entrada e saída do sistema. O componente *Full_Error_Data* refere-se ao número de erros total da transmissão. E a diferença do número dos componentes *Error_Data_Found* e *Full_Error_Data*, refere-se ao número de erros que não foram detectados pela codificação. Como resultado, obtém-se uma análise da robustez da codificação 8b/10b quando exposta a diferentes probabilidades de erro no canal.

Na figura 5 é ilustrado o gráfico da simulação do sistema, possuindo a probabilidade de erro no canal no eixo horizontal e a porcentagem de erro obtida na palavra recebida no eixo vertical.

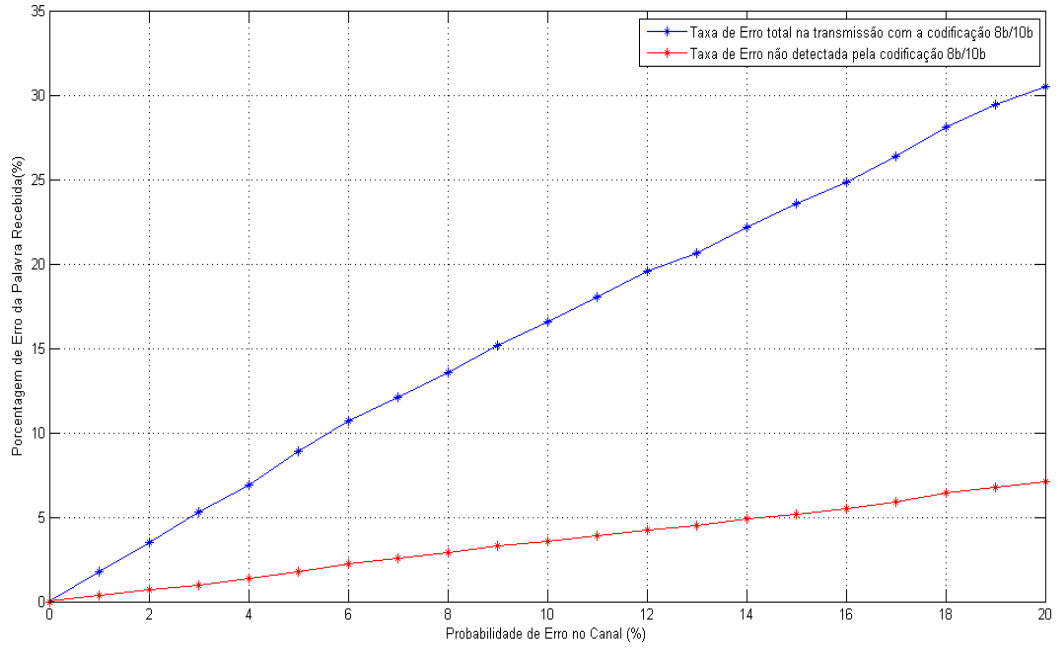


Figura 5: Análise de robustez da codificação 8b/10b quando exposta a diferentes probabilidades de erro no canal de transmissão.

O gráfico plotado em azul refere-se ao número total de erros na transmissão e o gráfico plotado em vermelho refere-se à porcentagem de erros que a codificação não detectou. Pelo gráfico ilustrado para uma probabilidade de erro no canal de 5%, a porcentagem de erro na palavra que a codificação não detectou foi de 1,77%.

A taxa de erro na transmissão ser maior do que a probabilidade de erro é explicada pelo descompasso do RD do *encoder* e *decoder*, quando há um erro na transmissão. O funcionamento da codificação 8b/10b baseia-se no valor do RD. Dessa maneira, pode-se estar na situação de o *encoder* mude o RD para +1. Neste caso, existe a possibilidade que em caso de erro na transmissão, o *decoder* mude o RD para -1. Esse descompasso entre os RDs do *encoder* e o *decoder* causam erros na decodificação. Esse efeito permanece por alguns ciclos de *clock*, no máximo 5 ciclos, até que o RD do *encoder* e *decoder* voltem a serem o mesmo.

4 Modelagem em Máquinas de Estado VHDL

A modelagem da codificação 8b/10b para ser descrita na linguagem VHDL requer a utilização de algumas técnicas de circuitos digitais. O funcionamento da codificação, inicialmente, processa os dados de entrada, de forma a separar os 5 bits menos significativos e os 3 bits mais significativos do dado de entrada de 8 bits. Com os bits separados, cada parte é codificada de acordo com o RD e as tabelas da codificação para 6 e 4 bits, respectivamente. Posteriormente, as partes codificadas são unidas, seguindo um padrão, calculando o novo RD e transmitidas ao receptor que realiza o caminho inverso na decodificação.

Para esse processamento do dado no sistema descrito em VHDL, pode ser usado a técnica de máquina de estados finitos. As máquinas de estado finito é um modelo matemático, largamente utilizado para representação de programas de computador e circuitos lógicos. Com este recurso é possível representar o comportamento de um circuito ou programa graficamente, podendo ser utilizado várias maneiras de representação como por exemplo o diagrama de estados. Esta maneira representa o sistema em máquinas de estado, com apenas seus estados e a ação que provocou as transições.

Na figura 6 são ilustradas as máquinas de estado do sistema descrito em VHDL da codificação 8b/10b. Pela figura estão presentes seis 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. No estado ?ENC_D? é realizado 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 valor da saída é atualizado 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? 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 ou identificar possíveis erros.

Agradecimentos

Aqui os autores devem apresentar os agradecimentos as pessoas e instituições pertinentes. A Comissão Organizadora do DINCON 2017 agradece aos autores por seguirem as recomendações descritas neste template.

Referências

- [1] F. Borges, Transmissão de Dados, Carnaxide: Schneider Electric, 2008. Disponível em: http://www.schneiderelectric.pt/documents/product-services/training/transmissao_dados.pdf. Acesso em: 1 ago. 2017.

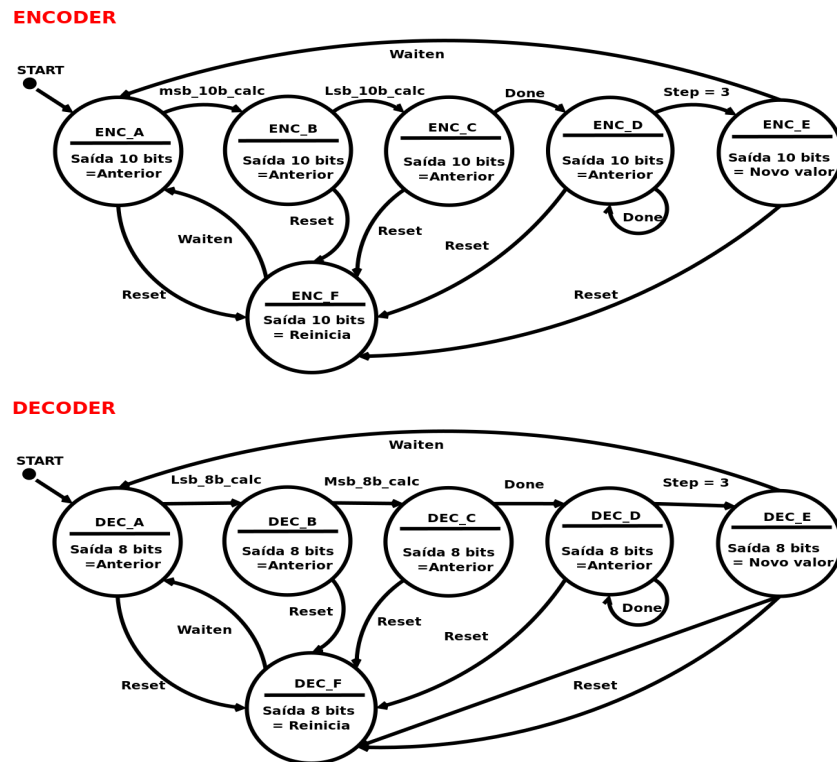


Figura 6: Máquina de estados do sistema implementado a codificação 8b/10b em VHDL.

- [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).