

Análise de Robustez e Modelagem em Máquina de Estado da Codificação 8b/10b

Reis, Victor Afonso

*Universidade Estadual Paulista “Júlio Mesquita Filho”; Faculdade de Engenharia de Ilha Solteira
victor.afonsoreis35@gmail.com*

Ramalho, Lucas Arruda

*Universidade Estadual Paulista “Júlio Mesquita Filho”; Faculdade de Engenharia de Ilha Solteira
lucasarrudaramalho@gmail.com*

Shinoda, Ailton Akira

*Universidade Estadual Paulista “Júlio Mesquita Filho”; Faculdade de Engenharia de Ilha Solteira
shinoda@dee.feis.unesp.br*

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 no sincronismo dos circuitos adicionais, sendo necessário um balanceamento nos *bits* transmitidos. Neste projeto é realizado um estudo da codificação 8b/10b através de um sistema implementado em software MatlabTM dentro do ambiente do *Simulink*. Além disso, o sistema foi também descrito na linguagem de descrição de hardware VHDL usando o software VivadoTM. 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.

1 - Introdução

Os sistemas de comunicação evoluíram durante décadas, chegando atualmente à taxas de transmissões na ordem de Gbps por segundo. Na figura 1 é ilustrado um esquema de um sistema de comunicação elementar. Dados que são transmitidos por qualquer meio (fibra óptica, fios metálicos, wireless, cabo coaxial) estão sujeitos à ruídos e problemas durante a transmissão pelo canal de transmissão como ilustrado.

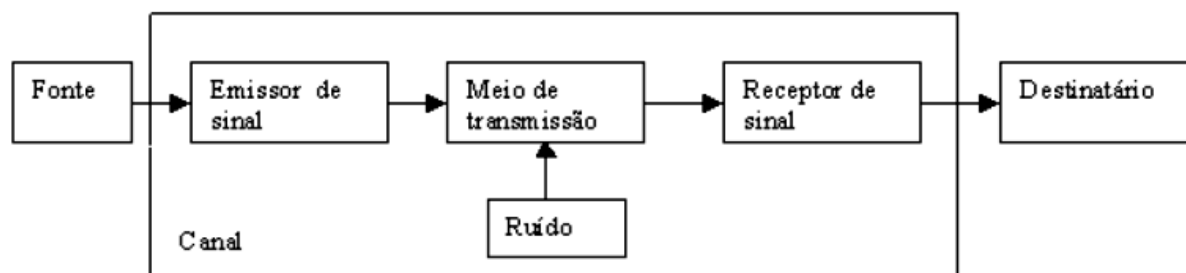


Figura 1 – Esquema de um sistema de comunicação elementar.

Em ambientes modernos, diversos recursos tecnológicos tendem a trocar dados a pequenas ou longas distâncias. Dessa forma, deve-se garantir uma alta confiabilidade dos dados na transmissão, de forma que no lado do receptor sejam recebidos os mesmos 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 [1][2].

Os hardwares que são responsáveis por transmitir os dados também são vulneráveis à ruídos e interferências, sendo possível enviar dados incorretos para o receptor. Com o aumento da velocidade de transmissão todos os problemas citados ficam mais evidentes, principalmente a dessincronização entre o transmissor e o receptor (jitter) e a aparição de vários dados ruidosos. Esses dois tipos de problemas são fatores limitante na velocidade de um canal de transmissão [3].

Uma forma de resolver os problemas citados é a introdução de codificações de canal na transmissão. Estas codificações pretendem introduzir redundâncias lógicas no dado a ser transmitido, como forma de reduzir as taxas de erro na transmissão. O uso destas codificações aumenta a qualidade do sistema de transmissão, apesar de diminuir a taxa efetiva de dados no sistema [4].

As codificações de canal possuem várias técnicas e características para codificar um dado. Dessa forma pode-se separá-las em 5 grandes categorias: unipolar, polar, bipolar, multinível e multitransição. Em métodos unipolares todos os níveis de tensão do sinal estão em um dos lados do gráfico tensão pelo tempo.

Um exemplo do método de codificação unipolar é a codificação em blocos. Esta codificação transforma um bloco com um determinado número “m” de *bits* em um outro bloco com “n” número de *bits*, adicionando uma redundância lógica facilitando a detecção de erros. Esta codificação 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 o 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 [5].

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 [6]. 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 e simulada em linguagem de descrição de *hardware* VHDL.

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 resalta 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 [10].

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 [10].

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.

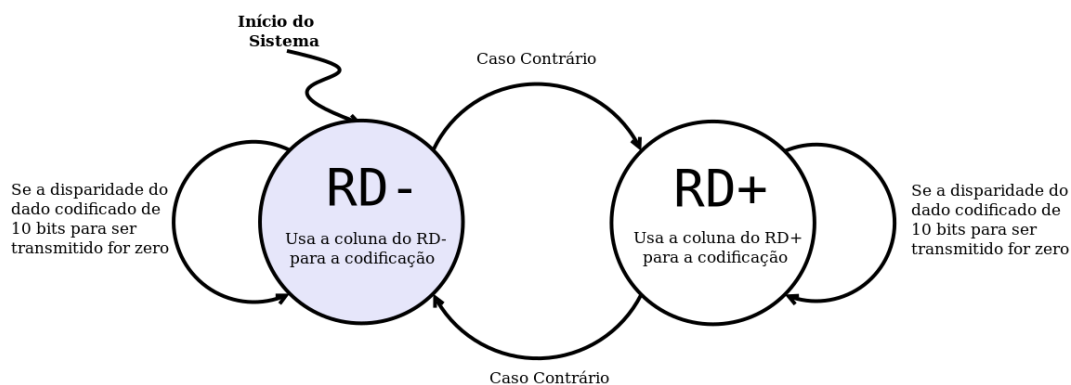


Figura 2 – Máquina de estados para a transição do RD.

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. [10].

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.

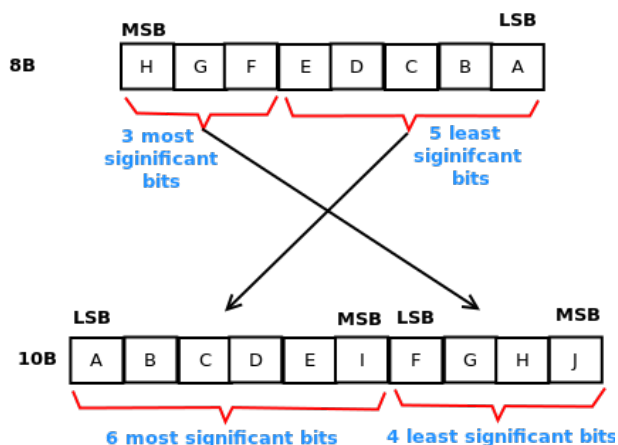


Figura 3 – Esquema da separação para codificação 8b/10b.

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.

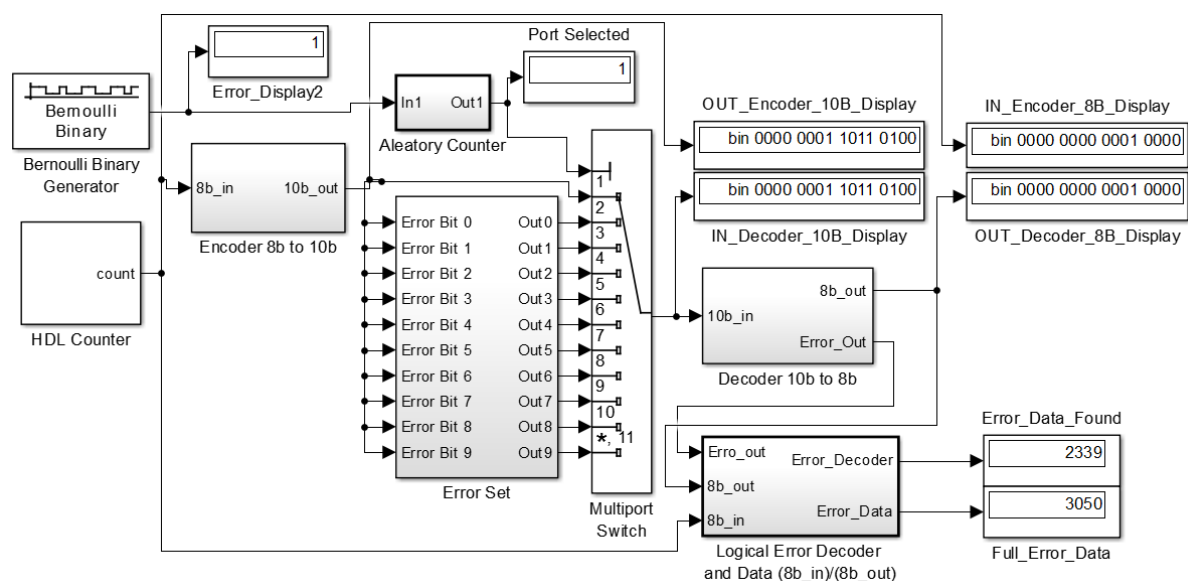


Figura 4 - Sistema desenvolvido no MatlabTM usando a EMFB do *Simulink*.

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.

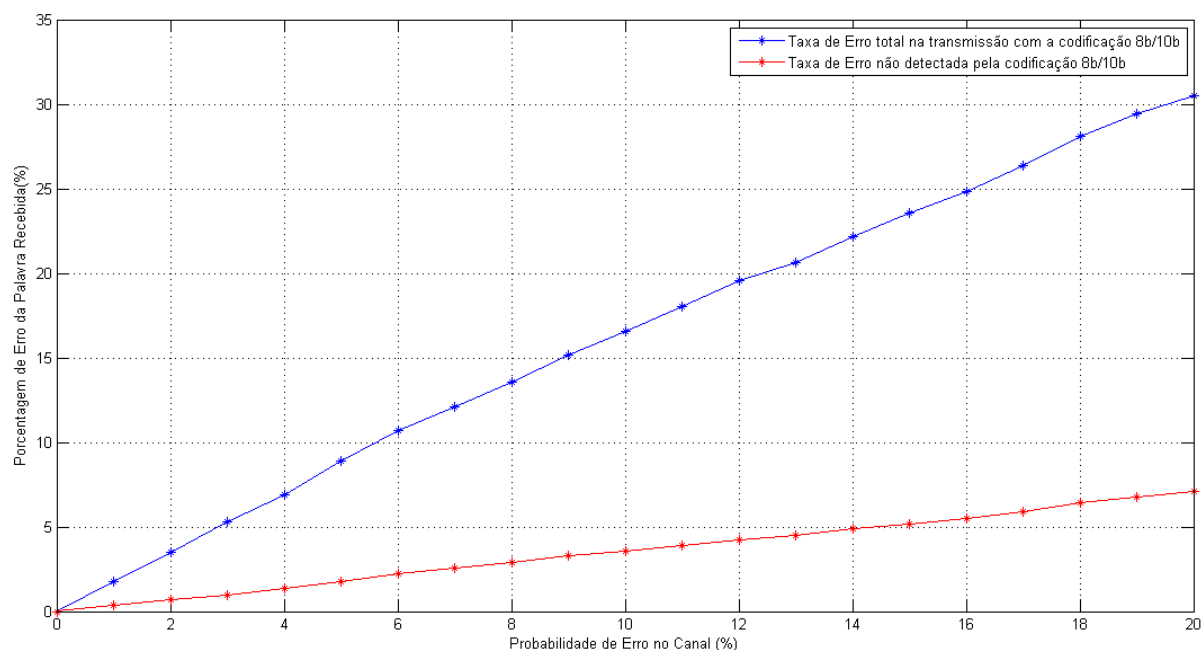


Figura 5 – Análise de robustez da codificação 8b/10b quando exposta a diferentes probabilidades de erro no canal de transmissão.

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.

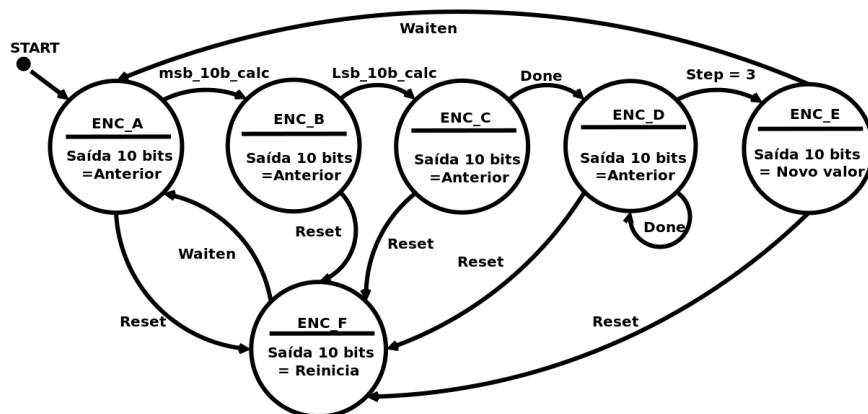
4 - Modelagem em Máquinas de Estado VHDL

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, é necessário o uso de máquinas de estado finito, para melhor controle dos procedimentos. 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 a representação por diagrama de estados. Esta representação é de representar 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.

ENCODER



DECODER

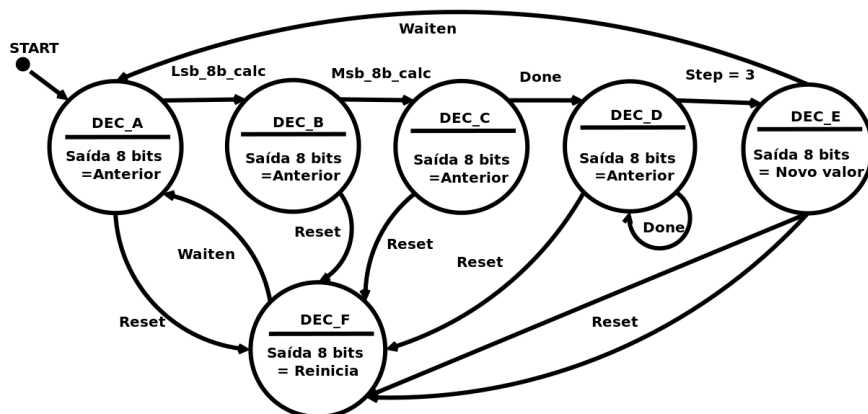


Figura 6 – Máquina de estados do sistema implementado a codificação 8b/10b em VHDL.

O sistema implementado em máquina de estado apresenta facilidade de programação e depuração do comportamento dos dispositivos. Entretanto o mesmo adiciona ciclos de operação para executar tarefas. As máquinas de estado propostas neste trabalho necessitam de 7 ciclos de operação

para codificar ou decodificar os dados. Como forma de mitigar o tempo de espera entre entrada e saída do sistema, recomenda-se que o dispositivo apresente dois *clock's* de funcionamento. O primeiro *clock* sincroniza a entrada de dados na codificação. O segundo deve ser 7 vezes mais rápido, de forma a tornar o processamento interno da codificação transparente à entrada.

Na figura 7 é ilustrado a simulação do sistema da codificação 8b/10b, descrito em VHDL no software Vivado™ da Xilinx.

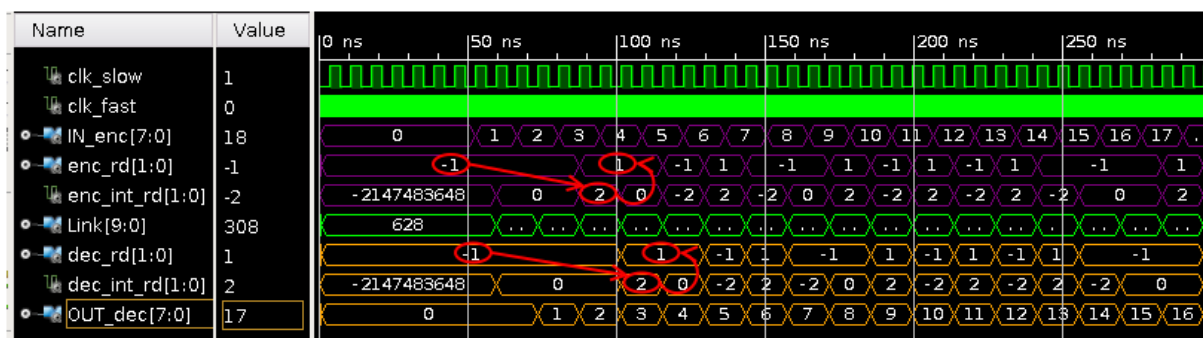


Figura 7 – Simulação do sistema da codificação 8b/10b descrita em VHDL.

Os sinais representados em roxo referem-se ao sistema do *encoder*, enquanto em laranja os do sistema do *decoder*. O *clock* do sistema pode ser observado pelo “clk_fast” (1 GHz). Adicionalmente o *clock* de entrada de dados pode ser observado pelo “clock_slow” (142,8 MHz). Os *clocks* inseridos no sistema tem o propósito de mostrar a relação entre suas frequências. Em um sistema real, o valor das frequências será relativo às capacidades do oscilador e dos *Phase-Locked Loop* (PLL) disponíveis.

A entrada de dados é representada pelo sinal “IN_enc” funcionando com um contador de 8 bits. Observa-se que ao codificar, o RD do sistema é atualizado, revelando um correto funcionamento do *encoder* pelo sinal “enc_int_rd” permanecer entre “+2, -2 e 0”. Pelo sinal “enc_rd” quando o estado atual do sistema está negativo (-1) e o próximo RD dos dados é positivo (+2), o sinal RD do sistema (enc_rd) é atualizado para um valor positivo (+1) como descrito na codificação. O sinal “Link” representa os dados codificados em 10 bits transmitidos para o *decoder*, estando representados na simulação na forma decimal. Nos sinais de RD do *decoder* (dec_rd, dec_int_rd) observa-se a mesma transição obtida no *encoder*, revelando o correto funcionamento do *decoder* na decodificação e do *encoder* na codificação. O sinal “OUT_dec” representa a saída do sistema, em que se verifica a presença dos mesmos dados de entrada na saída indicando que o sistema é funcional.

5 - Conclusões

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. Existem 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 simulação de análise de robustez da codificação 8b/10b em diferentes probabilidades de erro no canal. Os resultados mostram que a codificação pode apresentar falha ao detectar erros, e a porcentagem depende da probabilidade de erro ou qualidade do canal.

A técnica utilizada no MATLAB pode ser aplicada para testes de robustez de outras codificações em nível de simulação. Entretanto, a escolha da codificação para determinada aplicação (tipo do canal) deve ser realizada com testes reais. Logo este trabalho também descreve o



funcionamento do sistema com *encoder* e *decoder* em linguagem de descrição de hardware VHDL. Pretende-se em trabalhos futuros, testar a modelagem de máquinas de estado, apresentadas e simuladas neste artigo, em dispositivos FPGA que possibilitem os testes reais da codificação.

Além disso, será feita a análise de outras codificações comparando o desempenho e o grau de melhora do canal de transmissão entre diferentes condições de probabilidade de erro.

6 - Agradecimentos

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

7 - Referências

- [1] MACHADO, Renato. PROBLEMAS DE TRANSMISSÃO. 2016. Disponível em: <<http://coral.ufsm.br/gpscom/professores/RenatoMachado/ComunicacaoDeDados/ComDados09Renato.pdf>>. Acesso em: 06 mar. 2017.
- [2] TRONCO, Tania Regina; AVILA, Luis Fernando de. FUNDAMENTOS DE COMUNICAÇÕES ÓPTICAS. 2012. Disponível em: <[http://www.ft.unicamp.br/~lfavila/TT503/fundamentos de comunicacoes opticas.pdf](http://www.ft.unicamp.br/~lfavila/TT503/fundamentos_de_comunicacoes_opticas.pdf)>. Acesso em: 06 mar. 2017.
- [3] PRA, Thiago Dai. 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: 06 mar. 2017.
- [4] ALTOÉ, Mariana Olivieri Caixeta. CODIFICAÇÃO CONJUNTA FONTE-CANAL UTILIZANDO CODIFICADORES UNIVERSAIS ADAPTATIVOS. 2007. 92 f. Dissertação (Mestrado) - Curso de Engenharia Eletrônica e Computação, Instituto Tecnológico de Aeronáutica, São José dos Campos, 2007. Disponível em: <<http://www.bd.bibl.ita.br/tesesdigitais/000543362.pdf>>. Acesso em: 06 mar. 2017.
- [5] FOROUZAN, Behrouz A.. COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES. 4. ed. São Paulo: Amgh, 2010. 1134 p.
- [6] WIDMER, Albert X.; FRANSZKE, Peter A.. A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code. IBM Journal Of Research And Development. New York, p. 440-451. set. 1983.
- [7] RUSCHEL, Orlando T.. PRINCÍPIOS DA COMUNICAÇÃO DIGITAL. Porto Alegre: Edipucrs, 1996. 227 p.
- [8] CHAKRABARTI, Partha Pratim. TRANSMISSION CHARACTERISTICS OF OPTICAL FIBERS. New Delhi: Mcgraw-hill Education, 2015. 750 p.
- [9] SILVÉRIO, Ricardo Luiz. FIBRAS ÓPTICAS. 2002. 37 f. Monografia (Especialização) - Curso de Física, Física, Universidade Federal de Uberlândia, Uberlândia, 2002. Disponível em: <http://www.infis.ufu.br/infis_sys/pdf/RICARDO_LUIZ_SILVERIO.pdf>. Acesso em: 10 mar. 2017.
- [10] LATTICE SEMICONDUCTOR. 8B/10B ENCODER/DECODER. Hillsboro: Lattice Semiconductor, 2015.