

# Modelagem e Implementação em FPGA da Codificação 8b/10b

V. A. dos Reis\*, L. A. Ramalho\*, A. A. Shinoda\*

\*UNESP, Ilha Solteira, Brasil

Departamento de Engenharia Elétrica

e-mail: victor.afonsoreis35@gmail.com

**Abstract** - *In high speed communications, where the reability is very important, the coding 8b10b is indicated. At digital communications, one of signal integrity issues is the synchronism between transmitter and receiver. A long sequence of logical levels, zeros (0's) or ones (1's), can cause a desynchronism in the transmission channel. At this work, modeling and implements a coding 8b/10b system, wich give a solution to the unbalanced bits, in the FPGA development kit of Xilinx (Kintex 7), through the hardware description language VHDL by the software Vivado™. By the implementation, was obtained a generic system for test of encodings described at VHDL, besides parameters of frequency, data rate and characteristics of transmission operation using the coding 8b/10b system.*

**Palavras-chave:** *High Speed Serial Link, Codificação 8b/10b, VHDL, FPGA.*

## 1 Introdução

A complexidade e performance dos sistemas de comunicação evoluíram ao longo do tempo. Em qualquer meio de transmissão, os dados transmitidos podem sofrer distorções ou interferências causadas por problemas no canal de transmissão [1].

Em altas frequências problemas como a dessincronização do sistema e ruídos no canal de transmissão, são relevantes na comunicação digital. A dessincronização entre o transmissor/receptor e a aparição de dados ruidosos são um dos fatores limitantes na velocidade dos sistemas [2].

A codificação 8b/10b é uma técnica de codificação em blocos, o qual adiciona-se redundâncias de forma lógica em um bloco de (m) bits gerando um bloco de (n) bits. Este mapeamento possui algumas características próprias da codificação, o qual possibilita a verificação de possíveis erros na transmissão. Algumas codificações em blocos realizam um balanço DC no dado, portanto permite 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 [3].

O uso da codificação em blocos aumenta a qualidade do sistema, apesar de diminuir a taxa efetiva

de dados transmitidos [4]. Dessa maneira, o desafio resume-se em torno da análise dos parâmetros de cada codificação, sendo capaz de fornecer uma solução de menor custo e que atenda os requisitos do sistema [5]. Sendo assim, o objetivo deste trabalho é modelar e implementar em *VHSIC Hardware Description Language* (VHDL) um sistema com a codificação 8b/10b, dentro de um kit de desenvolvimento em FPGA da Xilinx (Kintex 7 KC705). Através dessa implementação, é possível obter informações e parâmetros do sistema.

Pela análise parcial do sistema implementado em FPGA, foi possível obter características do comportamento da codificação quando esta é submetida a erros entre transmissor e receptor. A seção 2 ressalta a teoria, a modelagem em máquinas de estado e a ilustração de um sistema no FPGA da codificação 8b/10b. A seção 3 descreve a implementação no FPGA do sistema descrito em VHDL. A seção 4 conclui os resultados da implementação no FPGA do sistema da codificação 8b/10b.

## 2 Materiais e métodos

### 2.1 A Codificação 8b/10b

Os dados de 8 bits são mapeados em 10 bits, dessa maneira, alguns dados de 10 bits com o mesmo número de bits 1's e 0's são usados como dados de controle da transmissão [5]. 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 disparidade é definida como a diferença entre o números de bits 1's e 0's do dado codificado para 10 bits.

A codificação é descrita de forma que os dados transformados em 10 bits só possuam disparidade de dados +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 [5].

O RD alterna-se entre +1 e -1 ou permanece o mesmo, de acordo com as regras das disparidades dos dados, ilustradas na Figura 1. Quando o RD for -1 não é possível a disparidade dos dados ser -2, assim como se o RD for +1 é impossível a disparidade dos dados ser +2. Nestes casos, ao se obter esse tipo de disparidade para cada RD mencionado, no lado do receptor, representa-se um erro na transmissão.

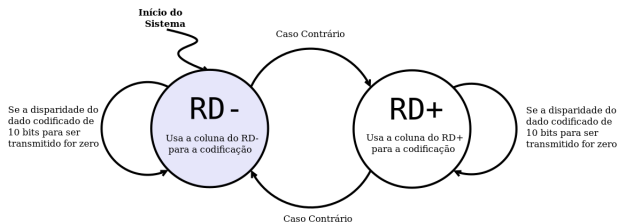


Figura 1 – Máquina de estados para a transição do RD

Uma característica da disparidade de dados codificados possuir sempre valores definidos (+2, -2 ou 0), possibilita a detecção de erro pelo receptor. Como descrito, para codificar os dados usa-se o RD que sempre será -1 quando o sistema for inicializado [5]. Na figura 2 é ilustrado o modo como a codificação dos dados é realizada. O dado de 8 bits é separado em duas partes, codificando-os e após a sua junção é gerado um dado de 10 bits como ilustrado.

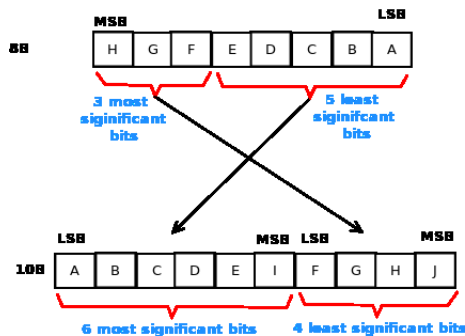


Figura 2 – Esquema da separação para codificação 8b/10b

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 DC torna-se importante para a recuperação do relógio e consequentemente sincronização entre emissor e receptor.

## 2.2 Modelagem em Máquinas de Estado da Codificação 8b/10b

O funcionamento da codificação 8b/10b, descrita em VHDL através de máquinas de estados, é ilustrado na figura 3. Como pode-se analisar há a presença de sete estados no sistema do *encoder*.

No *encoder*, o estado "ENC\_A" refere-se ao estado de espera do dado de 8 bits na entrada. Já o estado "ENC\_B" e "ENC\_C" codificam a parte menos e mais significativas do dado de entrada (8 bits), como na figura 2. Nos estados "ENC\_D", "ENC\_E", "ENC\_F" e "ENC\_G" realiza-se a junção das partes codificadas (saída de 10 bits), cálculo da disparidade de dados, atualização da saída e atualização do RD, respectivamente.

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. Caso identifique um erro, a máquina é direcionada para estados que tratam deste caso. Nestes estados, a saída vai para alta impedância, gerando um sinal de erro, e posteriormente espera-se um novo dado de entrada.

## 2.3 Descrição do Sistema da Codificação 8b/10b

A máquina de estados foi descrita em VHDL, no software Vivado™, 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, tais como geração de dados, geração aleatória de erros, verificação de dados recebidos e sistemas de depuração.

O design FPGA proposto inclui o sistema da codificação com os demais periféricos interligados dentro de um mesmo dispositivo. O canal de transmissão entre *encoder* e *decoder* é emulado internamente no FPGA e erros podem ser inseridos arbitrariamente. A descrição do sistema para implementação no FPGA é ilustrado na figura 4.

Pela figura 4 observa-se para o funcionamento do sistema deve-se descrever um *Phase-locked Loop* (PLL), um *Virtual Input Output* (VIO), um *Integrated Logical Analyser* (ILA), um contador para inserir dado no *encoder*, um sistema (VIO, Pipeline) para inserir erro no sistema e um arranjo para modificar o dado de saída do *encoder* para ser entrada no *decoder*.

Através do PLL insere-se um clock de forma que o FPGA consiga implementar. O VIO permite

## ENCODER

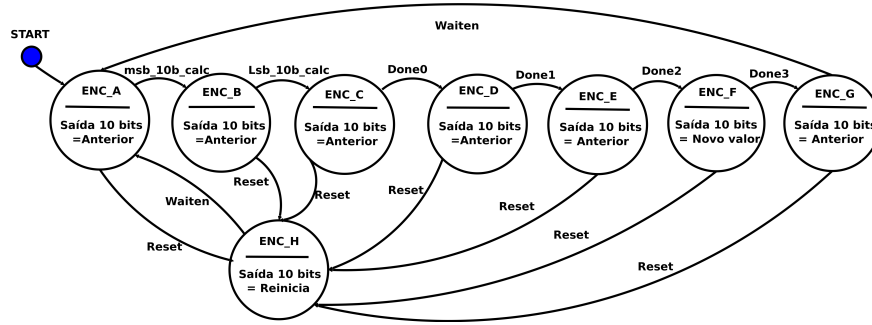


Figura 3 – Máquina de estados do sistema implementado a codificação 8b/10b em VHDL

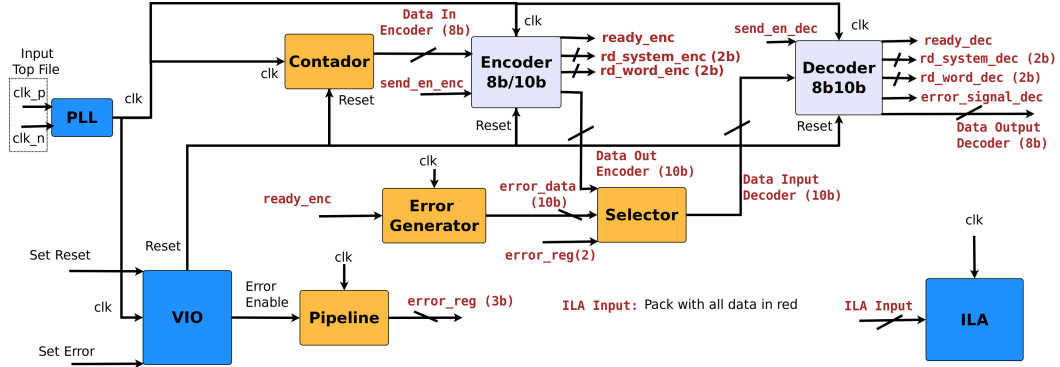


Figura 4 – Descrição do Top File da codificação 8b/10b para implementação no FPGA

inserir sinais dentro do sistema, já o sistema de pipeline seta um sinal de erro em somente um ciclo de *clock*. O sistema *Error Generator* gera um vetor de 10 bits, com apenas um bit em nível lógico alto, rotacionando-o quando o sinal *ready\_enc* for alto. O bloco *Selector* realiza uma operação XOR com o dado de 10 bits do *Error Generator* e o dado de saída do *encoder*, somente quando o sinal "error\_reg(2)" for alto. Caso contrário, o bloco seletor deixa passar somente o dado de saída do *encoder*, ou seja, sem passar pela operação XOR. Pelo ILA pode-se visualizar todos os sinais dentro do sistema, dessa forma todos os dados marcados em vermelho na figura 4 são inseridos no ILA para obter suas formas de onda.

## 3 Resultados e Discussão

Após a implementação através do software Vivado<sup>TM</sup>, a frequência de *clock* definida no teste do sistema foi de 300 MHz.

Pelo teste da implementação obteve-se uma latência de codificação, em ciclos de *clock*, de 5 ciclos até o dado de entrada estar devidamente disponível na saída. Para a decodificação, esta latência é de 4 ciclos totalizando 9 ciclos para codificação e decodificação. Porém, o sistema que in-

sere erro na transmissão utiliza 2 ciclos de totalizando 11 ciclos para o dado codificado estar devidamente decodificado na saída.

Dessa forma, considerando uma frequência de 300 MHz a transmissão tem a capacidade de saída de dados da ordem de 480 Mbps. O ILA e VIO impedem a determinação da frequência máxima, uma vez que estes sistemas não suportaram o aumento da frequência do *clock*. Dessa forma, testou-se retirar o ILA e a máxima frequência obtida foi de 400MHz, ou seja, uma taxa de dados transmitidos de 640 Mb/s. A única opção para remoção de algum componente do sistema era o ILA, pois o sistema depende do VIO para o funcionamento.

Na figura 5 é ilustrado a simulação do sistema no kit FPGA da Xilinx. O sinal *Error Input*, gerado pelo usuário, permanecendo em nível lógico alto até que seja inserido dado no *decoder* gerando erro em um bit no dado. Depois de 4 ciclos de *clock*, após gerado o erro, o sistema fornece um sinal de erro. O RD entre *encoder* e *decoder* entra em descompasso, recuperando-se após 33 ciclos de *clock*, ou seja, após 3 palavras na transmissão.

Este descompasso do RD entre o *encoder* e *decoder* é característico da codificação, fazendo com que algumas palavras transmitidas sejam perdidas. Após alguns testes obteve-se um descompasso máximo de 54 ciclos, ou seja, 8 palavras perdidas

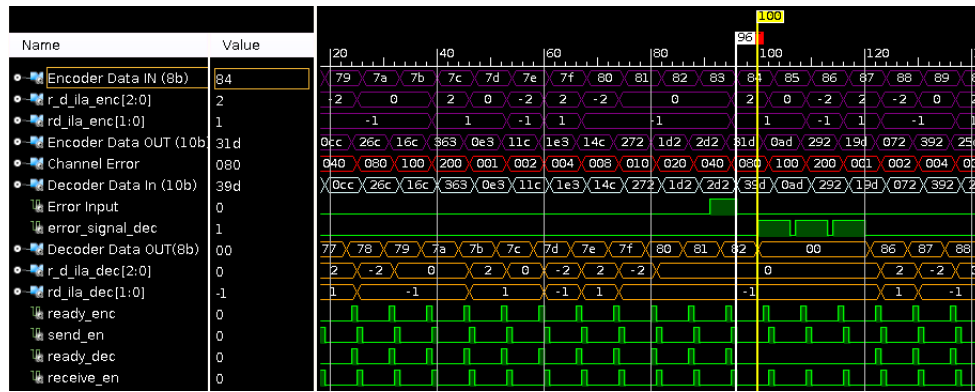


Figura 5 – Simulação da codificação 8b/10b implementada no FPGA com erros inseridos no canal

na transmissão.

Emulando mais erros no canal, observou-se que o sistema, em algumas situações, decodificava o dado recebido para um valor errôneo e somente após algumas palavras o sistema sinalizava um sinal de erro. Dessa forma, é possível obter um dado válido mesmo inserindo erro, porém o sistema desbalanceia o RD ao recebê-lo. Os dados seguintes são codificados corretamente por conta de possuem o mesmo formato para qualquer RD. Após alguns ciclos, o sistema detecta um erro por conta do desbalanço do RD. Pelos testes, a latência máxima para encontrar este erro foi de 53 ciclos.

## 4 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. A escolha entre as codificações 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 *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 pelo sistema foi de 400 MHz, com uma taxa de transferência de 480 Mb/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 tam-

bém descreve máquinas de estado para o sistema da codificação 8b/10b em 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 o apoio do SPRACE (Processo Fundunesp 2448/2015) e da CAPES.

## Referências

- [1] F. Borges, “Transmissão de dados,” Tech. Rep. 3, 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: 3 ago. 2017.
- [2] T. D. Pra, “Medidor de taxa de erro de bit para fibra Óptica,” Master’s thesis, Universidade Federal do Rio Grande do Sul, 2012. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/65425/000858235.pdf?sequence=1>. Acesso em: 1 ago. 2017.
- [3] B. A. Forouzan, *Data Communications and Networking*. McGraw-Hill, 4 ed., 2007.
- [4] E. Berlekamp, R. Peile, and S. Pope, “The application of error control to communications,” *IEEE Communications Magazine*, vol. 25, pp. 44–57, April 1987.
- [5] A. X. Widmer and P. A. Franaszek, “A dc-balanced, partitioned-block, 8b/10b transmission code,” *IBM Journal of Research and Development*, vol. 27, pp. 440–451, Sept 1983.