

# IMPLEMENTAÇÃO DE PROTOCOLO DE SERIAL LINK EM FPGA

João Paulo Lawrence da Silva<sup>1</sup>, Renato Camargo Giacomini<sup>2</sup>

Departamento de Engenharia Elétrica, Centro Univeritário FEI, São Bernardo do Campo, Brasil  
joapaulo.lawrence@gmail.com, renato@fei.edu.br

O objetivo deste trabalho é estudar a implementação de Protocolo de Links Seriais (*Serial Link Protocols*), como o Aurora 8b/10b [1] em placa FPGA. Ademais, foi realizado um tutorial de implementação do protocolo Aurora 8b/10b em uma placa FPGA Xillinx KC705.

## 1. Introdução

Um dos mais frequentes tipos de *hardwares* utilizados para processamento de dados gerados pelos sensores do *The Hadron Colider* (LHC) é o baseado em FPGA, pela capacidade paralela, que permite maior velocidade. Contudo, somente a utilização de placas FPGA não garantem processamento rápido por si só, porque quando se trata de comunicação em altas frequências, existem desafios na camada física na transmissão e recepção destes sinais gerados pelos sensores do LHC.

Contudo, para vencer estes desafios, é implementado protocolos de comunicação nas placas FPGA's para controlar ou diminuir os efeitos de uma comunicação em altas frequências. Portanto, este trabalho irá explicar o funcionamento básico do protocolo de comunicação Aurora 8b/10b e mostrar resultados sobre sua implementação.

## 2. Placas FPGA e Linguagem VHDL

*Field Programmable Gate Array* (FPGA) é um circuito integrado composto por milhares de unidades lógicas, que podem ser manipuladas a partir de uma matriz de trilhas condutoras e *switches* programáveis. Para sua manipulação, é necessário sequências binárias para a configuração da FPGA, que pode ser gerado por ferramentas de *software*, através de uma linguagem de programação, como a VHDL, gerando um arquivo binário.

A Figura 1 mostra a placa FPGA utilizada para implementar o protocolo Aurora 8b/10b e realizar testes disponibilizadas por *São Paulo Research and Analysis Center* (SPRACE). Para realizar medidas sobre as portas de saída e entrada da placa FPGA, foi usado o Osciloscópio Tektronix AFG 3252.



Figura 1 – Placa FPGA utilizada para realizar testes. Modelo Xillinx KC705.

*VHSIC* (*Very High Speed Integrated Circuits* *Hardware Description Language* VHDL) é uma linguagem de programação para descrever sistemas eletrônicos e digitais de alta velocidade [2].

## 3. O protocolo Aurora 8b/10b

O protocolo Aurora 8b/10b tem como objetivo transmitir dados ponto-a-ponto por uma ou mais linhas seriais de comunicação de alta velocidade. As vantagens em destaque desse protocolo são portabilidade com protocolos ethernet e TCP IP e é um projeto *open source*. É implementado através de uma interface que se comunica através de um ou mais canais de linhas *full-duplex*. Nessa comunicação é transmitido dados já codificados e serializados pelo protocolo, e define-se que esta é uma camada de PDU's (*Protocol Data Unit*) e fluxo de PDU's. Esta interface é controlada por uma aplicação de usuário, por meio de mensagens de controle de fluxo e PDU's de usuários. Podemos observar esse comportamento pela figura 2:

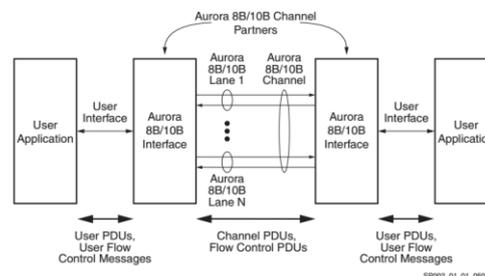


Figura 2 – Canais do Aurora 8B/10B

A transmissão e recepção de dados é em pares de pacotes de um octeto – 8 bits é equivalente a um octeto - e segue uma hierarquia de ordem de transmissão de pacote de dados (do maior para o menor): Compensação de *clock*, Sequenciais de inicialização, Controle de fluxo de PDU's nativo (mapeamento e dados codificados pela interface), Controle de Fluxo de PDU's de usuário (controle das mensagens das aplicações de usuário), Canais de PDU's (pacotes com dados codificados, serializados e preparados para envio) e Sequencias aleatórias (pacotes para preparar os canais para transmissão).

Na figura 3, é possível visualizar o processo de transmissão de PDU's de usuários, ou seja, de dados. Esse processo segue os seguintes passos: (1) Pareamento; (2) Encapsulamento com delimitadores de canal de PDU; (3) Codificação 8b/10b do carregamento vindo do canal de PDU; (4) Serialização e codificação do *clock*.

O processo (1), é responsável por garantir a transmissão de quantidades pares de octetos, pois a transmissão neste canal de comunicação é feita por

pares de octetos, ou seja, quando a transmissão é ímpar, adiciona-se um octeto de valor 0x9C para satisfazer a condição.

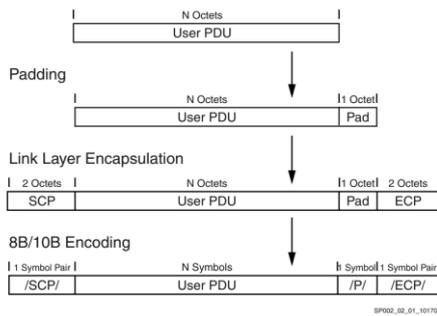


Figura 3 – Procedimentos de transmissão de dados

O processo (2) é responsável pelo controle de prioridade de transmissão de dados. Para isso, o PDU de usuário é encapsulado com sequências de símbolos de controle, chamados de *ordered sets*. Isso significa que os dados a serem transmitidos são categorizados para maior eficiência na identificação do tipo de dado. O processo (3), codifica os dados para o envio. Define-se *link layer payload* (carregamento de link de camada) o resultado da estrutura de dado formada após o pareamento.

Todos os caracteres, com exceção do octeto de paridade, são codificados utilizando a simbologia especificadas em tabelas na documentação do protocolo Aurora 8b/10b. Para codificação de 256 caracteres de dados (Dx.y), separamos os 8 bits em dois grupos, “x” e “y”. Cada bit é designado a um caractere do alfabeto de “A” ate “H”, como mostra na figura 4.

D25.3	HGF 011	EDCBA 11001
	y = 3	x = 25

sp002\_03\_01\_081302

Figura 4 – Exemplo de sequência de 8 bit a ser transmitido.

A codificação do protocolo aurora acontece quando rotacionamos a ordem dos dois grupos formados além de adicionar apenas um bit para cada grupo, e designando novamente cada bit a um caractere do alfabeto de “a” ate “j”. Neste exemplo, a ordem dos bits podem ser vistas na figura 5:

/D25.3/	abcdei 100110	fghj 1100
	from x term	from y term

sp002\_03\_02\_082102

Figura 5 – Exemplo de sequência de 8 bits codificados

O bit adicional nos grupos tem como objetivo a paridade das quantidades de “0” e “1”, ou seja, quantidades pares para cada nível lógico. Observe que a notação de grupo de código foi alterado para /D25.3/. Isso significa que este volume de bits, que agora somados são 10, já foi codificado pelo protocolo.

#### 4. Resultados

Os seguintes resultados foram obtidos a partir da execução do tutorial fornecido pela Xilinx, *Aurora 8B10B Protocol Specification, 2014* [3]: Estudo sobre o diagrama de blocos da implementação do protocolo e

observação do comportamento dos dados trafegados pelo canal de link de comunicação de transmissão.

O projeto elaborado pelo tutorial, consiste em módulos escritos e estruturados em VHDL. Os módulos em destaque são: (a) *traffic.frame\_gen\_i* e (b) *aurora\_module\_i*. O módulo (a) é responsável por gerar dados aleatórios a serem tratados pelo protocolo Aurora 8b/10b na placa transmissora (TX). O módulo (b) é responsável por fazer a codificação (em TX) ou decodificação (na placa receptora RX) e (des)serialização dos dados. Na figura 6, é possível ver a simulação feita no *software* Vivado.



Figura 6 – Simulação de transmissão de dados

A porta TXDATA\_IN foi forçada com valor a critério do aluno, que no caso foi a palavra “4abc”, para ser codificada pelo protocolo. Já na porta receptora, o valor do dado foi diferente, comprovando que o dado foi tratado pelos módulos transmissores de codificação.

#### 5. Conclusões

Protocolos como Aurora 8B10B são expoentes na transmissão de dados em alta velocidade, e por isso são utilizados para *High Energy Physics* em aceleradores de partículas como o CERN.

A partir do estudo teórico feito sobre o Protocolo Aurora 8b/10b e revisão da linguagem VHDL, o aluno teve maior entendimento sobre design elaborado pelo tutorial executado, por meio da análise do fluxo do projeto e pela revisão dos códigos em VHDL dos módulos e abrindo a esquematização dos módulos e seus respectivos sub módulos.

Neste estudo também foi investigado conceitos sobre um protocolo de comunicação mais recente e rápido, o protocolo GBT-FPGA [4].

#### 6. Referências

[1] Xilinx, *Aurora 8B/10B Protocol Specification*, v.2.3, 2014.  
 [2] IEEE Computer Society. *IEEE Standard VHDL Language Reference Manual. Revision of IEEE Std 1076-2002*, 2009.  
 [3] Kumar, D. et al. “*Designing a System Using the Aurora 8B/10B Core (Simplex) on the KC705 Evaluation Kit*”. 05 de Janeiro, 2015.  
 [4] S. Baron et al. “*Implementing the GBT data transmission protocol in FPGAs*”. CERN, 1211 Geneva 23, Suíça. 2009. DOI: 10.5170/CERN-2009-006.631. Conference: C09-09-21.10.

#### Agradecimentos

Os autores gostariam de agradecer o apoio financeiro da CAPES e do CNPq. 1 Aluno de IC do Centro Universitário FEI. Projeto com vigência de 10/18 a 09/19. Agradecimentos ao time de laboratório do SPRACE.