

Desenvolvimento e validação do bloco lógico programável genérico para comunicação entre os ASIC GBT-SCA da eletrônica de Front-End para o upgrade do experimento LHCb

Development and validation of the generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade

Cairo Pimenta Cheble Caplan* e André Massafferri Rodrigues†

Centro Brasileiro de Pesquisas Físicas

Federico Alessio,‡ Clara Gaspar,§ Richard Jacobsson,¶ e Ken Wyllie**

Organização Europeia para a Pesquisa Nuclear

Submetido em 29/01/2016

Resumo: O Upgrade do experimento LHCb prevê uma atualização do sistema de aquisição de dados operar a 40 MHz, necessária para operar dez vezes a luminosidade a fim de seu projeto inicial. Para tanto, a eletrônica de Front-End de todos os subsistemas será modificada permitindo o gerenciamento de uma maior taxa de dados gerada, maior ocupância dos subdetectores e operação sem uso de Trigger.

Neste trabalho, sucessor experimental e mais aprofundado do trabalho *A generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade* [1], é descrito uma nova maneira de transmitir informações de controle para a eletrônica de Front-End, aproveitando as conexões ópticas bidirecionais e recursos dos ASICs GBT e GBT-SCA. São apresentados detalhes de sua implementação bem como testes de validação. O resultado deste trabalho será usado no próximo upgrade do LHCb, que ocorrerá entre 2018 e 2019, assim está sendo avaliado seu uso como um pacote de *firmware* padronizado para os outros experimentos do LHC - ATLAS, CMS e ALICE.

Palavras chave: LHCb; Processamento de sinais digitais (DSP); Controle e monitoramento de sistemas online; Sistemas de controle de detectores.

Abstract: The LHCb experiment has proposed an upgrade towards a full 40 MHz readout system to be able to run ten times its initial design luminosity. The entire Front-End electronics will be upgraded in order to cope with higher sub-detector occupancy, higher data rate and to work in a complete trigger-less fashion.

In this work, an updated and in-depth release of the *A generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade* [1], we describe a novel way to transmit slow control information to the Front-End electronics, by profiting from bidirectional optical connections and the GBT and GBT-SCA chipset capabilities. The implementation and preliminary validation tests are shown as well. The deliverable of this project will be used on the next upgrade of LHCb, which will occur between 2018 and 2019, and it is under evaluation to be set as a default package for the others LHC experiments - ATLAS, CMS and ALICE.

Keywords: LHCb; Digital signal processing (DSP); Control and monitor systems online; Detector control systems.

*Electronic address: cairo@cbpf.br

†Electronic address: massafferri@cbpf.br

1. O UPGRADE DO EXPERIMENTO LHCB

O experimento LHCb [2] é um experimento de precisão ligado ao acelerador LHC dedicado a busca por nova física ao medir seus efeitos em estudos relacionados a relação de simetria CP (Carga-Paridade) e decaimentos raros. Utilizando um método indireto, o LHCb é capaz de explorar efeitos que são fortemente suprimidos no Modelo Padrão de partículas elementares.

Por se tratar de um colisor próton-próton, o LHC é em grade medida uma fábrica de partículas de sabores pesados produzindo mais de 100 mil pares $b\bar{b}$ a cada segundo na luminosidade nominal do LHCb de $2.10^{33}\text{cm}^{-2}\text{s}^{-1}$. Considerando que pares $b\bar{b}$ são produzidos predominantemente nas direção frontal, o LHCb foi projetado como um espectrômetro de braço único, com os elementos do detector instalados ao longo da linha do feixe principal do LHC, cobrindo uma faixa de pseudorrapidez de $2 < \eta < 5$.

O LHCb vem apresentado uma boa performance durante o período de 2010-2012 [3] acumulando $\sim 3 \text{ fb}^{-1}$ dados e tem previsões de acumular outros $\sim 5 \text{ fb}^{-1}$ entre 2015-2018. No entanto, o LHCb é limitado em termos de capacidade de transferência de dados em 1 MHz, sendo especialmente ineficiente para os canais hadrônicos no hardware do L0 *Trigger*. Assim, os documentos de *Letter Of Intent* [4], *Framework TDR* [5] e de *Trigger and Online TDR* [6] descrevem os planos para um detector atualizado que permitirão ao LHCb aumentar sua produtividade de física em decaimentos com múons em uma escala de 10 vezes, a produtividade em canais hadrônicos por 20 e aumentar a coleta para 50 fb^{-1} a uma luminosidade constante de $2.10^{33}\text{cm}^{-2}\text{s}^{-1}$. Isto corresponde a 10 vezes a luminosidade atual e complexidade aumentada (*pileup*) de 5 vezes.

2. O UPGRADE DA ARQUITETURA DE READOUT DO LHCB

Com o objetivo de remover as principais limitações do detector LHCb atual, a estratégia para o upgrade do experimento consiste basicamente em remover o primeiro nível de *trigger* em hardware (L0 *trigger*), tornando assim o detector completamente sem *trigger*. Ao remover o L0 *trigger*, os

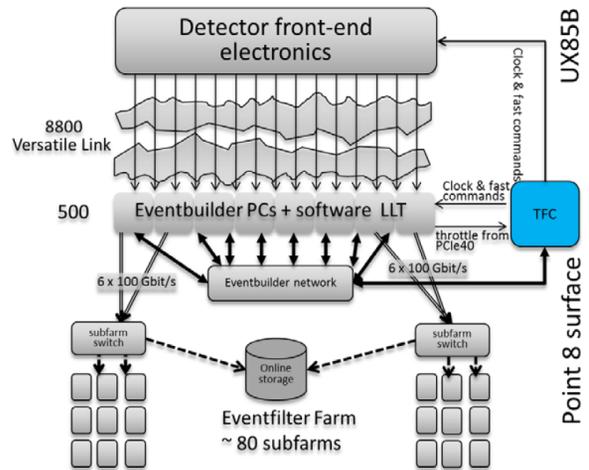


Figura 1: A arquitetura de aquisição de dados na atualização do LHCb

eventos do LHC são registrados e transmitidos a uma velocidade de 40 MHz, resultando num sistema de aquisição de dados operando a $\sim 40 \text{ Tb/s}$. Todos os eventos estarão então disponíveis na *farm* de processamento onde um *trigger* implementado em software, completamente flexível, fará a seleção dos eventos, onde sua saída de dados terá uma taxa aproximada de 20 KHz de eventos armazenados em discos. Isto irá maximizar a eficiência dos sinais a uma alta taxa de eventos.

As consequências diretas deste método são que alguns dos subdetectores do LHCb terão que ser completamente reprojatados para suportar a luminosidade média de $2.10^{33}\text{cm}^{-2}\text{s}^{-1}$ e todo o detector será equipado com uma eletrônica de FE (*Front-End*) completamente nova. Somado a isso, toda a arquitetura de *readout* terá de ser reconstruída para suportar a nova largura de banda, da ordem de Tb/s, e o fluxo de dados a uma frequência de 40 MHz. A figura 1 mostra a arquitetura de aquisição de dados [7] (chamada de *readout*) nova do programa de atualização do LHCb. Deve ser levado em conta no entanto que apesar do sistema final não possuir um *trigger*, o então *trigger* L0 feito em hardware será feito em software. Isto é normalmente referido como um *trigger* de baixo nível em software (Software LLT) e seu propósito principal é o de permitir uma instalação em estágios da rede de DAQ, gradualmente aumentando a taxa de leitura de dados dos atuais 1 MHz aos finais 40 MHz. Isto no entanto não irá alterar a taxa de eventos gravados na FE, que irá operar completamente sem um *trigger*, independente da taxa de saída de dados do DAQ.

Para manter o sincronismo por todo o sistema de leitura, controlar a eletrônica de FE e distribuir o clock e as informações de sincronismo por todo o sistema de leitura, um sistema centralizado de Controle Rápido e de Temporização (TFC, apresentado na figura 1) foi previsto, como um upgrade do sistema de TFC atual [8]. O upgrade do sistema de TFC será então interfaceado com todos os elementos na arquitetura de *readout* ao se aproveitar das capacidades de comunicação bidirecionais dos links ópticos e transceivers do FPGA e um alto nível de interconectividade. Em particular, o sistema TFC irá lucrar bastante dos recursos dos cir-

‡Electronic address: Federico.Alessio@cern.ch

§Electronic address: Clara.Gaspar@cern.ch

¶Electronic address: Richard.Jacobsson@cern.ch

**Electronic address: Ken.Wyllie@cern.ch

cuitos integrados do projeto GigaBit Traseiver (GBT) [9], atualmente desenvolvidos no CERN para a comunicação da eletrônica de suas FEs. O sistema TFC também será responsável em transmitir informações de controle de baixa velocidade (ECS), onde suas placas eletrônicas baseadas em FPGA será ligada ao ECS global do LHCb.

3. O CONTROLE DE ALTA E BAIXA VELOCIDADE PARA A FE POR VIA DO SISTEMA DE TFC

A figura 2 ilustra em detalhes a arquitetura lógica do sistema de TFC atualizado. Um grupo dos supervisores do sistema de *readout*, chamados de S-ODIN, gerenciam centralizadamente a leitura síncrona e assíncrona de comandos, distribuindo o clock do LHC e gerenciando a expedição de eventos. Cada S-ODIN é associada com uma partição de sub-detecores que na realidade são *clusters* de placas de *readout* (TELL40) e placas de interfaceamento (SOL40). Enquanto as TELL40s são dedicadas a leitura de fragmentos de eventos das FE e enviá-los para o DAQ para processamento em software, as SOL40 são dedicadas à distribuir dados de controle de alta e baixa velocidade para as FE, o que é feito ao retransmitir informações de temporização e clock no link óptico para a FE, e por anexar informações do ECS no mesmo datagrama de dados. Por aproveitar recursos do chipset GBT [9], comandos de alta velocidade, clock e controle de baixa velocidade são então transmitidos no mesmo link óptico bidirecional. Isto é uma novidade importante em relação ao experimento LHCb atual onde os controles de alta e baixa velocidade são transmitidos por redes de dados diferentes. Na FE, as informações síncronas do controle de alta velocidade são decodificadas e distribuídas por um chip GBT mestre em cada placa de FE, também responsáveis por recuperar e distribuir o clock de maneira determinística. As informações de controle de baixa velocidade são transmitidas aos circuitos integrados GBT-SCA, ou simplesmente SCA, por meio do GBT mestre. O GBT-SCA é capaz de eficientemente distribuir dados de configuração do ECS para os chips da FE através de uma gama completa de barramentos e interfaces eletrônicas, de maneira genérica [10]. Os dados de monitoramento são enviados de volta no *uplink* do mesmo link óptico seguindo o mesmo caminho de dados em sentido contrário, do GBT-SCA para o GBT Mestre e para a SOL40 correspondentemente em seguida.

A espinha dorsal em hardware de toda a arquitetura de aquisição é uma placa eletrônica do tipo PCIe Gen3 hospedada num PC comum. O mesmo hardware é usado para as placas TELL40, SOL40 e S-ODIN, o que as diferencia é o firmware usado em cada uma. A placa será equipada com até 48 links ópticos bidirecionais, um FPGA Arria 10 da Altera e uma conexão do tipo PCIe Gen3 de 16x com um PC com vários núcleos.

A figura 3 mostra o esquemático da implementação da fusão das fluxos de informações de controle em alta e baixa velocidade no mesmo link óptico para a eletrônica de FE [11] no firmware da placa SOL40.

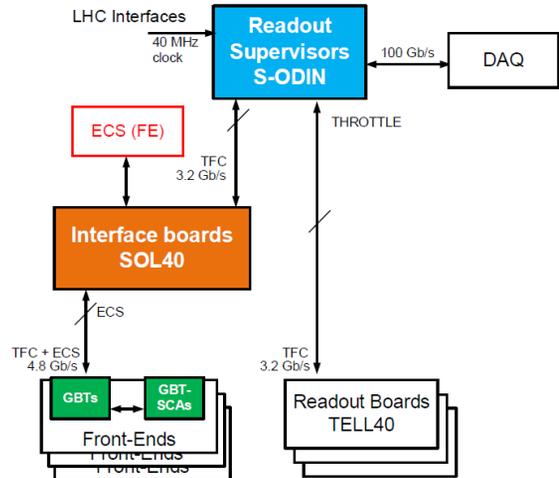


Figura 2: Arquitetura lógica do sistema de TFC atualizado

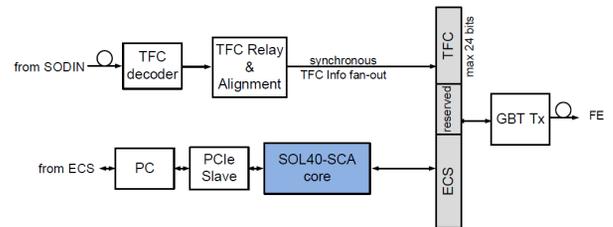


Figura 3: Visão esquemática do algoritmo de união de informações de TFC e ECS no link GBT em direção a eletrônica de FE no firmware da placa SOL40

Um bloco de retransmissão e alinhamento do TFC extrai um máximo de 24 bits da palavra de dados completa do TFC transmitido pelo S-ODIN codificando comandos de alta velocidade, informações de temporização e vários sinais de reset. Estes 24 bits são então retransmitidos para o link GBT para ser enviado a FE. A palavra de dados é gerada a 40 MHz e transmitida com uma latência constante. A palavra de dados de TFC originada do S-ODIN é usada para reconstruir o clock localmente no FPGA para então ser utilizada para alimentar a lógica dentro do firmware.

A respeito da parte de controle de baixa velocidade, o LHCb desenvolveu um núcleo de firmware, chamado SOL40-SCA, para controlar cada chip GBT-SCA localizado nas FEs, suportando todas as suas funcionalidades e protocolos. Sua localização dentro do firmware da SOL40 é destacado na figura 3. Isto foi obtido porque o projeto foi desenvolvido para ser totalmente reconfigurável, ou seja, o protocolo do SCA escolhido pode ser selecionado em tempo real via comandos executados pelo sistema de ECS do LHCb [12] junto com dados de configuração. O destino de tais dados podem ser selecionados por uma máscara de configuração. O bloco foi desenvolvido e configurado para suportar um link GBT com até 16 GBT-SCAs conectados a ele. Este pode ser replicado tantas vezes forem necessárias para compreender todos os links GBT conectados às placas SOL40. Ao final, o mesmo firmware permitirá controlar toda eletrônica de FE do upgrade do experimento LHCb, sendo um total de aproximadamente 2500 links ópticos bidirecionais e 90 pla-

cas SOL40. Este bloco lógico é independente da tecnologia, desenvolvido em linguagem HDL, além de não fazer uso de qualquer elemento específico de alguma tecnologia e ser completamente transparente quanto ao conteúdo do campo de dados. De forma simplificada ele é um gerenciador de SCAs genérico via links ópticos para a eletrônica de BE para os experimentos do LHC.

4. O BLOCO SOL40-SCA

O bloco fornece uma maneira de controlar com alto paralelismo e flexibilidade vários dispositivos na eletrônica de FE interfaceados aos GBT-SCAs através de links GBT. Suas principais funcionalidades são listadas a seguir:

- Provê uma interface em hardware genérica (FPGA) entre o sistema ECS e a eletrônica de FE.
- Construir, codificar e decodificar pacotes compatíveis com o GBT-SCA.
- Serializar e desserializar pacotes de comando na palavra de comando enviada à eletrônica de FE de acordo as especificações do GBT-SCA.
- Suporte a todos os protocolos do GBT-SCA (SPI, I²C, JTAG, GPIO e ADC+DAC).
- Suporte para todos os comandos e canais do GBT-SCA
- Suporte para vários GBT-SCAs por link GBT e vários links GBT por FPGA.
- Possibilidade de retransmissão de pacotes e monitoramento da transmissão
- Modularidade, i.e., componentes podem ser removidos se não forem necessários
- Robustez, confiabilidade, programabilidade e flexibilidade.

O bloco é essencialmente composto de uma série de camadas como destacado na figura 5. Suas principais funções são:

- Armazenar os pacotes de configuração do ECS e decodificá-los como comandos e vice-versa nas camadas Interface Layer e Buffer Layer.
- Construir os pacotes do tipo GBT-SCA equivalentes com o protocolo escolhido na Protocol Layer.
- Codificar este pacote seguindo o protocolo de comunicação necessário (HDLC [13]) na MAC Layer
- Serializar e rotear os pacotes para o GBT-SCA selecionado conectado ao link GBT localizado na FE através da Link Layer

Na prática, o ECS gera um comando que é transmitido ao FPGA pelo barramento PCIe. Este comando contém um esquema de endereçamento estendido (chamado de *extended addressing scheme*) que informa o SOL40-SCA onde e como rotear o pacote de configuração e um esquema de código de comando (chamado de *command code scheme*) que informa o bloco quais ações realizar (i.e. leitura, escrita, espera por resposta ou não espera). Como recurso adicional, ele pode conter os dados de configuração a serem enviados para a FE em caso de uma operação de escrita. No FPGA, o comando é armazenado num *buffer* para poder ser carregado pela *Protocol Layer* quando esta não estiver ocupada. O comando ECS é então decodificado e os pacotes específicos de protocolos SCA são construídos de acordo. A informação sobre qual protocolo deve ser utilizado está contida no comando ECS e é completamente genérica, isto é o bloco é capaz de construir qualquer pacote SCA tempo de execução simplesmente baseado no conteúdo do comando. Finalmente, o pacote é encapsulado pelo protocolo HDLC para então ser roteado aos bits correspondentes na palavra de dados do link GBT para o chip GBT Mestre na FE. Este campo de bits escolhido é selecionado baseado nas conexões do GBT na FE. Para ser tão genérico quanto possível, este campo também é um parâmetro configurável tal que o bloco possa ser usado com qualquer configuração da FE. O bloco também possui a possibilidade de retransmissão de pacotes caso alguma transferência de dados tenha falhado.

4.1. Interface Layer

De modo a acessar o PC através barramento PCIe, a placa SOL40 utiliza internamente um barramento do tipo *Altera Avalon MM bus*, que é mapeado num dos Registradores Bases de Endereços (BARs) [14], no caso BAR 0. Logo, uma interface *Avalon MM Slave* é usada na ECS Interface Layer para realizar operações de leitura e escrita originadas e destinadas ao PC de controle.

A Figura 6 mostra a estrutura de um comando ECS genérico que é construído pelo sistema de controle via uma interface gráfica dedicada e scripts. Este comando é transmitido ao bloco no firmware e contém todas as informações relevantes a este, assim o mesmo pode construir genericamente e de forma flexível pacotes compatíveis com o GBT-SCA. No primeiro campo, um esquema de endereçamento estendido é implementado: Os endereços do link GBT, do GBT-SCA e de seu canal são incluídos. Somando-se a isto, há um campo chamado ECS Command dedicado a especificar comandos do ECS (por exemplo, *Read* ou *Write*). No segundo campo, a largura do *ECS Command* em número de *bytes* para informações de tamanho do pacote e um campo específico ao protocolo SCA utilizado são utilizados. Este é seguido por pacotes de Dados caso uma operação de escrita seja requisitada. Todos os campos tem alinhamento final de 32 bits permitindo que um comando completo seja uma tabela de palavras de 32 bits.

O mesmo comando ECS é gerado pelo firmware em resposta a cada operação de *polling* do ECS.

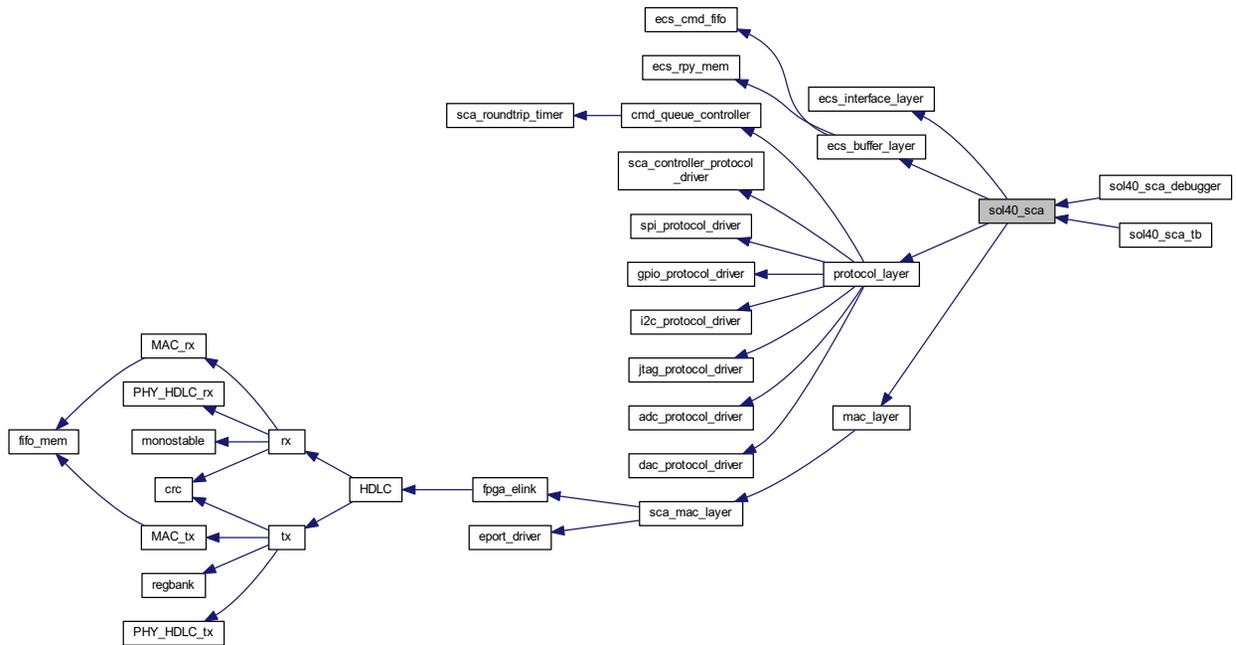


Figura 4: Formas de onda dos sinais do firmware

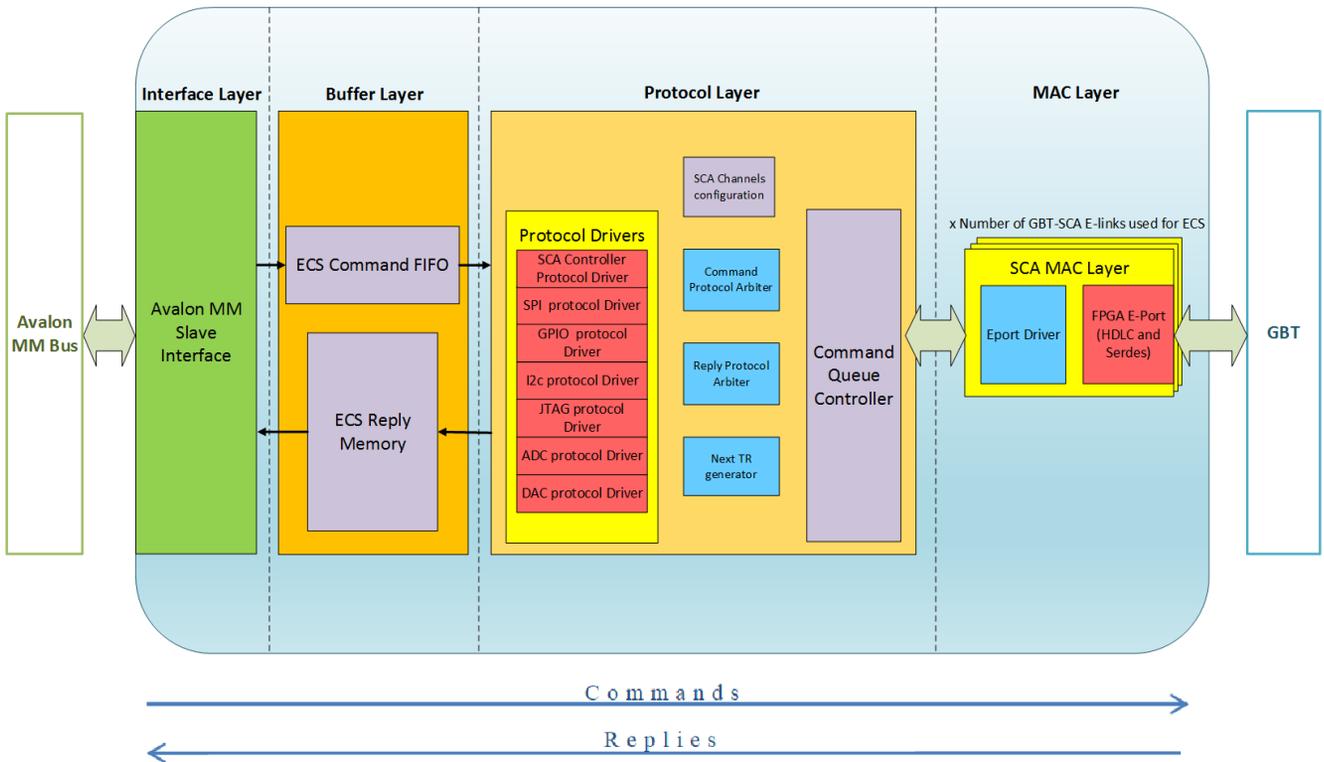


Figura 5: Visão de topo do diagrama de blocos da arquitetura do projeto SOL40-SCA

4.2. Buffer Layer

Os comandos ECS são armazenados numa *FIFO*. Isto é necessário porque a frequência de clock usadas pela interface

Avalon MM Slave é de 40 MHz e o tamanho da palavra de dados é fixa de 32 bits. Considerando que um comando ECS pode se estender por várias palavras de 32 bits, o comando ECS precisa ser armazenado para permitir a construção dos

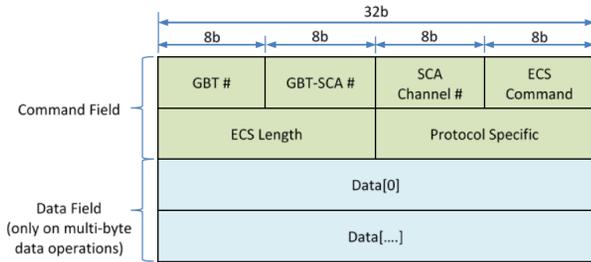


Figura 6: Formas de onda dos sinais do firmware

pacotes do tipo SCA correspondentes e transmiti-los pelos par de bits associado ao link GBT.

Uma FIFO, chamada de ECS Commands FIFO, é dedicada a armazenar pacotes de comando ECS. Foi escolhido que cada link GBT terá uma ECS Command FIFO associada porque cada fluxo de dados do ECS corresponde a uma única sequência de vários comandos. No entanto, eles são enviados de modo assíncrono para os GBT-SCAs e seus canais associados. Esta se torna uma maneira simples de criar um mecanismo de *back-pressure* e evitar congestionamento durante a construção e transmissão de pacotes. Isto também significa que o ECS pode enviar uma tabela de comandos em operações de escrita contínuas enquanto que o firmware lidará com a leitura e interpretação de comandos por cada canal.

Uma memória, chamada ECS Reply Memory, é dedicada a armazenar réplicas, chamadas de *replies*, a um comando ECS específico. Está foi concebida como uma estrutura RAM ao contrário de uma FIFO para permitir que o software possa acessar a memória seguindo um mapeamento a partir do esquema de endereçamento estendido. O ECS pode então realizar *polling* esperando por uma *reply* específica a partir de um comando gerado previamente.

4.3. Protocol Layer

O chipset GBT-SCA suporta uma grande variedade de barramentos eletrônicos que podem ser interfaceados com chips da FE. Na chamada Protocol Layer, cada comando ECS é transformado ou traduzido num ou mais comandos SCA, onde o protocolo específico para um dado canal do SCA é construído. Isto permite ao usuário a escolher de forma flexível quaisquer barramentos este quiser acessar simplesmente formatando-o no esquema mostrado na Figura 6. Desta maneira o mesmo firmware pode ser utilizado para todas as combinações possíveis de barramentos na FE sem ser dependente das escolhas dos subdetectores nas FE.

Além disto, uma característica importante é que a Protocol Layer mantém informações a respeito de informações dos comandos SCA gerados para fins de retransmissão de pacotes, gerenciamento da leitura de comandos ECS vindos da ECS Commands FIFO baseados num estado de ocupado ou não funcional (quando um SCA incorreto ou não disponível foi selecionado) e quando um pacote está pronto, este é transmitido para a MAC Layer. Isto é gerenciado por dois módulos, um dedicado a arbitragem de leitura ou escritas de co-

mandos ECS e outro à transmissão e recepção de comandos SCA.

A Figura 7 mostra um exemplo de uma operação na Protocol Layer, onde um PC do ECS envia um comando de escrita I2C para um determinado dispositivo I2C na FE.

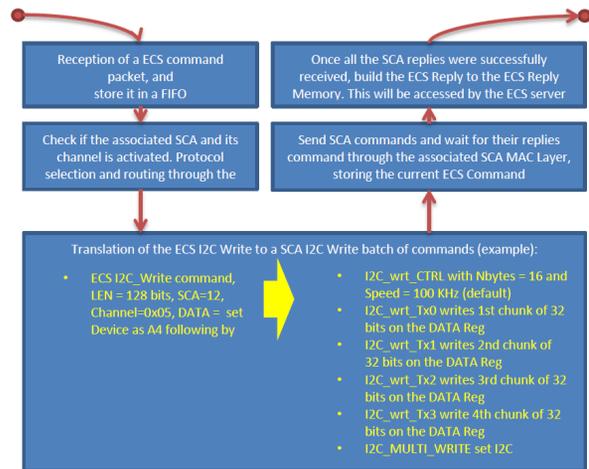


Figura 7: Formas de onda dos sinais do firmware

4.4. MAC Layer

A camada chamada MAC Layer, derivada do conceito usado no modelo OSI para protocolos de comunicação de dados "Controle de Acesso ao Meio", é principalmente responsável por encapsular o pacote de *payload* do SCA, por uma implementação da interface de comunicação Altera Atlantic [15], dentro do protocolo HDLC [13] e por serializá-lo em pares de bits, para então serem alocados em cada palavra GBT do link óptico através de interfaces do tipo e-link [16], padrão de interface elétrica criado no CERN baseado na sinalização SLVS. Além disso, a MAC Layer desserializa o fluxo de dados e extrai o *payload* quando uma réplica é recebida. Ela também suporta os comandos básicos do protocolo HDLC de reinicializar, estabelecer e testar conexão além de detecção de erros de transmissão [17].

O núcleo da MAC Layer é um bloco chamado FPGA E-Port. Este bloco é baseado no original E-Port IP Core [10, 16] mas com algumas diferenças fundamentais. O FPGA E-Port foi feito de maneira independente de dispositivos, não possui uma conexão auxiliar e foi projetado sem Redundância Modular Tripla [18] (chamada TMR ou *Triple Modular Redundancy*) e o código de Hamming [19] para integridade de dados das FIFOs, pois este bloco foi feito para ser usado em ambientes salvos contra radiação, e sem uma porta auxiliar de comunicação, pois o bloco será usado na eletrônica de Backend e a necessidade de redundância no caso de perda de comunicação ocorre na eletrônica de Frontend.

Um recurso adicional é a possibilidade de retransmitir um pacote se a transmissão de um comando anterior tiver sido falho. Isto é feito na MAC Layer pois o protocolo com-

Fitter Summary	
Fitter Status	Successful - Fri Jun 19 23:19:11 2015
Quartus II 64-Bit Version	14.1.1 Build 190 01/19/2015 SJ Full Version
Revision Name	mini_daq_firmware
Top-level Entity Name	Q_Top_Level_comp
Family	Stratix V
Device	5SGXEA7N2F45C3
Timing Models	Final
Logic utilization (in ALMs)	53,243 / 234,720 (23 %)
Total registers	60401
Total pins	127 / 1,064 (12 %)
Total virtual pins	0
Total block memory bits	17,625,925 / 52,428,800 (34 %)
Total DSP Blocks	2 / 256 (< 1 %)
Total HSSI STD RX PCSs	7 / 48 (15 %)
Total HSSI 10G RX PCSs	1 / 48 (2 %)
Total HSSI GEN3 RX PCSs	0 / 48 (0 %)
Total HSSI PMA RX Deserializers	8 / 48 (17 %)
Total HSSI STD TX PCSs	7 / 48 (15 %)
Total HSSI 10G TX PCSs	1 / 48 (2 %)
Total HSSI GEN3 TX PCSs	0 / 48 (0 %)
Total HSSI PMA TX Serializers	8 / 48 (17 %)
Total HSSI PIPE GEN1_2s	1 / 48 (2 %)
Total HSSI GEN3s	1 / 48 (2 %)
Total PLLs	15 / 92 (16 %)
Total DLLs	0 / 4 (0 %)

Figura 8: Utilização de recursos do firmware, incluindo S-ODIN, SOL40 e TELL40, no FPGA Stratix V da placa MiniDAQ

pleto incluindo o de comunicação já está prontos neste estágio. Um tempo de expiração programável é utilizado para aguardar a resposta dos pacotes dos SCA correspondentes e um bit programável transmitido dentro do comando ECS é usado para informar o bloco para retransmitir um pacote ou simplesmente enviar um aviso ao ECS sem retransmiti-lo. Isto pode ser feito num critério por comando em tempo de execução. Outro recurso adicional é a capacidade de esperar pela resposta do GBT-SCA correspondente.

5. VALIDAÇÃO DO PROJETO

A validação do projeto se deu em três fases diferentes. A primeira etapa foi a simulação do firmware para FPGA do projeto através do software ModelSim 10.3 da companhia Mentor Graphics, um ambiente de simulação para código em HDL. A segunda fase de testes em hardware da comunicação do bloco inserido no firmware usado no MiniDAQ e na placa GBTx SAT Board com o GBT-SCA contido no protótipo da placa VLDB. A terceira e última fase de testes foi constituída de todo um setup de dispositivos usados no fluxo de dados do ECS, partindo de um computador e terminando num dispositivo ou sensor acoplado a um dos barramentos do GBT-SCA, neste último caso destacaram-se a leitura e gravação de registradores de um chip GBTx pela sua interface de programação I²C [20] e a interface SPI [20] do ASIC Claro, que será usado na frontend dos subdetectores RICH do LHCb. Todas etapas de validação realizadas até a data da realização deste texto foram de caráter estritamente funcional.

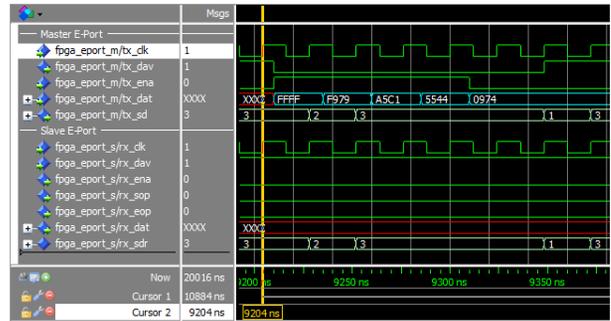


Figura 9: Simulação da transmissão de uma cadeia de 64 bits de dados, divididos em seções de 16 bits, de uma interface E-Port mestre para uma escrava.

5.1. Simulação

A etapa de simulação foi a primeira forma de verificação do trabalho. Sendo uma etapa recorrente quando aplicadas mudanças significativas à estrutura do projeto e também pelo seu baixo custo de execução. O primeiro bloco a ser simulado foi a adaptação do bloco E-Port original [10, 16], parte fundamental para a implementação do protocolo de comunicação com o GBT-SCA, onde este foi traduzido da linguagem de HDL Verilog para VHDL como exigência do LHCb, e eliminadas alguns recursos desnecessários para a implementação em FPGA na BE como iterados na seção sobre a MAC Layer 4.4.

As Figuras 9 e 10 mostram a transmissão e recepção de uma cadeia de 64 bits de dados entre dois blocos FPGA E-Port, um com o papel de mestre e o outro como escravo. Esta simulação representa o tamanho máximo do *payload* do pacote de transmissão de dados usados em comunicações com o GBT-SCA, onde este *payload* é encapsulado no protocolo HDLC e serializado em 2 bits com *bit-stuffing* antes de serem transmitidos de fato.

Todos os recursos do bloco FPGA Eport, como comandos de reset, teste e conexão, verificação de erros e transmissão de dados bidirecional foram simulados e verificados. Foram feitas simulações funcionais de integridade dos pacotes de todos os outros blocos do firmware SOL40-SCA, porém não foi realizar uma simulação dos pacotes com o firmware do GBT-SCA pois este não foi disponibilizado, sendo possível seu uso apenas com o hardware real na etapa de testes.

5.2. Testes em Hardware

Para a realização dos testes em hardware foram preparados dois arranjos diferentes. Cada um deles têm em comum versões dos primeiros protótipos da placa VLDB (Versatile Link Demonstrator Board) [21], parte do programa Versatile Link [22], que contém basicamente um chip GBTx, a primeira versão do GBT-SCA, módulos de comunicação óp-

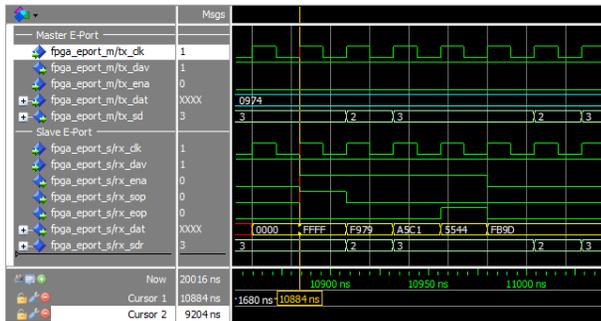


Figura 10: Recepção dos 64 bits da Figura 9 na interface E-Port escrava.

tica bidirecional VTRx [23, 24] e portas E-Link na forma de conectores HDMI, todos estes elementos desenvolvidos de forma customizada pelo projeto Radiation Hard Optical Link pelo CERN, que também inclui todo o sistema GBT, para a emulação de elementos comuns na FE dos futuros upgrades dos experimentos do LHC.

Foram utilizadas duas placas VLDB para os testes, uma principal e outra secundária. Na VLDB principal foi utilizado o GBTx como interface de rede para os dados de Slow Control e um GBT-SCA para a criação de barramentos eletrônicos genéricos de controle e monitoração a partir dos sinais da interface E-Link gerados no GBTx. A VLDB secundária teve o propósito único de servir como dispositivo I²C escravo para testes do GBT-SCA na VLDB principal, no caso o próprio GBTx desta foi utilizado como chip I²C. A figura 11 mostra todos os elementos utilizados nos dois diferentes arranjos do setup de testes do projeto, montado num laboratório do edifício 13 do CERN.

O primeiro setup é baseado na placa eletrônica desenvolvida para avaliação e testes do chip GBTx, chamada GBTx SAT Board [25]. Os principais elementos desta placa são o chip GBTx, um FPGA Cyclone V da fabricante Altera e conectores ópticos do tipo SFP. Utilizando esta placa como uma emulação da eletrônica Backend de um experimento, foi possível modificar o firmware do FPGA de modo a incluir o bloco SOL40-SCA ligando-o à lógica existente de alimentação de dados do chip GBTx. Deste modo o SOL40-SCA se comunicava com o GBT-SCA da VLDB principal através de uma ponte de dados óptica, transparente do ponto de vista operacional, através dos chips GBTx contidos nas duas diferentes placas.

Para a realização rápida e robusta dos testes funcionais deste primeiro setup foi criado, como parte deste trabalho, um software em Tcl/Tk para programação e comando do GBT-SCA, através da SOL40-SCA, a partir de um computador operando o software Quartus da Altera e conectando-o com o FPGA através do cabo de conexão JTAG [26] da USB-Blaster. Este software foi baseado no trabalho Altera JTAG-to-Avalon-MM Tutorial de D. W. Hawkins [27].

O segundo setup dos testes em hardware é baseado utiliza o MiniDAQ [28] como portadora do bloco SOL40-SCA.

Esta placa, específica do upgrade do experimento LHCb, serve como protótipo das placas de interface de Backend do LHCb [8, 29]. Ela inclui um FPGA Stratix V com o firmware da eletrônica de aquisição (Readout), controle rápido (TFC) e controle lento (ECS), onde o firmware deste último é chamado de SOL40 8. o MiniDAQ também possui um módulo de *Credit-Card PC* que funciona como um computador linux operando o software do supervisor do sistema de controle e depuração da eletrônica, seja através de scripts ou através de módulos no sistema SCADA WinCC, da Siemens. Neste último foram realizados testes utilizando scripts e módulos WinCC, estes últimos de autoria de Maurício Féo do grupo de pesquisa no CBPF.

As interfaces testadas e verificadas do GBT-SCA quando utilizado nos diferentes setups incluem os barramentos ADC, DAC, GPIO e I²C, SPI. Para os três primeiros foram realizados testes simples, dados o estado frágil do protótipo da VLDB, onde o GBT-SCA disponível precisou ser ressoldado por *wire-bonding* pois estava com um *footprint* invertido.

Como dispositivos I²C foram usados os chips GBTx, onde alguns dos resultados são representados através da ferramenta SignalTap da Altera na figura 12 e de medição por osciloscópio na figura 13 operando em até 1MHz, e uma placa de microcontrolador Arduino por meio de um I²C *level shifter* baseado no chip PCA9306 da Texas Instruments. A interface SPI foi verificada em conjunto com o chip CLARO8v2 [30], que será usado no upgrade do subdetector RICH do LHCb [31] com diferentes velocidades, assim como em modo loopback. A figura 14 mostra o gráfico de medição de uma operação SPI com o chip claro operando na velocidade máxima de 20 MHz.

6. CONCLUSÃO

Dentro do seu programa de atualização, o experimento LHCb desenvolveu um bloco de firmware genérico para gerenciar qualquer GBT-SCA dentro da atualização do experimento. Isto foi obtido pela implementação de um código do tipo HDL, capaz de gerenciar qualquer protocolo de qualquer GBT-SCA sobre qualquer link GBT, programável em tempo de execução. Este bloco é tão genérico que pode ser utilizado em qualquer ambiente de FE que tem por característica a utilização do chipset GBT.

O bloco de firmware está em estágio funcional e estável, sendo usado nos sistemas de testes nos subdetectores do LHCb. Uma campanha de testes em conjunto com primeiros chips GBT-SCA foi realizada para fins de testar sua robustez, confiabilidade e compatibilidade, onde está sendo avaliado o uso deste bloco sob pequenas modificações para uso nos outros grandes experimentos do LHC.

Ainda no primeiro semestre de 2016 serão realizados testes e validação do protocolo JTAG junto ao grupo de ECS do LHCb, testes quantitativos de tolerância a falhas e desempenho estão sob estudo.

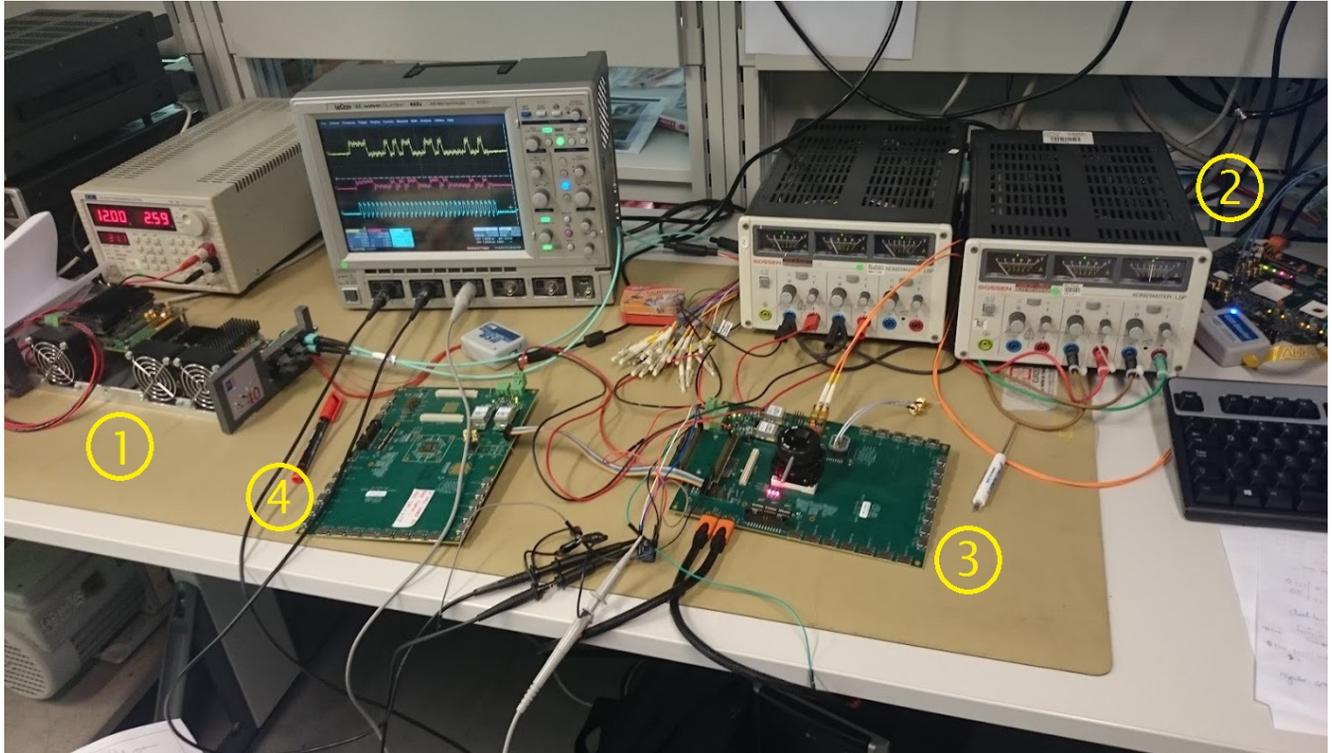


Figura 11: Setup de Testes no laboratório do edifício 13, no CERN. Estão representados 1 - MiniDAQ, 2 - GBTx SAT Board, 3 - VLDB principal com o GBT-SCA e 4 - VLDB secundária com GBTx como dispositivo I²C.

log: 2015/05/27 18:58:49 #0		click to insert time bar									
Type	Alias	Name	-2048	0	2048	4096	6144	8192	10240	12288	14336
		...gger_inst sol40_sca:u_sol40_sca SRES_i									
		⊕..._0_sca:u_sol40_sca ECS_COMMAND_i	0000000h					0000000h			
		...ca:u_sol40_sca WRITE_ECS_COMMAND_i									
		⊕..._sca:u_sol40_sca ECS_RESPONSE_o	0000000h						0000000h		
		...a:u_sol40_sca ECS_RESPONSE_VALID_o									
		...ca:u_sol40_sca READ_ECS_RESPONSE_i									
		⊕...cs_interface_layer_inst block_counter	0h							0h	
		...link:fpga_elink_inst HDLC:HDLC_inst rx_clk									
		⊕..._pga_elink_inst HDLC:HDLC_inst tx_dat	0000h					0000h			
		...nk:fpga_elink_inst HDLC:HDLC_inst tx_ena									
		⊕..._pga_elink_inst HDLC:HDLC_inst rx_dat	EB07h		32CEh					BB83h	
		...nk:fpga_elink_inst HDLC:HDLC_inst rx_ena									
		⊕..._HDLC_rx:PHY_HDLC_rx_pri rx_data	EBh		32h					BBh	
		⊕..._HDLC_tx:PHY_HDLC_tx_inst tx_data	00h					01h			
		⊕...roller_inst command_queue.counter[0]	0h		1h					0h	
		⊕..._inst command_queue.prot_cmd[0].ch					05h				
		⊕...mand_queue.prot_cmd[0].cmd_count	1h					2h			
		⊕..._inst command_queue.prot_cmd[0].err					00h				

Figura 12: Exemplo de operação de leitura no barramento I²C vistos sob os registradores da ferramenta SignalTap da Altera. O pulso do sinal ECS_COMMAND_i representa o recebimento de um comando ECS originado de um PC com um software supervisor. Os dois pulsos de tx_dat e tx_ena representam dois pacotes de dados enviados pelo bloco para o GBT-SCA correspondente, seguidos de dois pulsos de respostas correspondentes, sendo o segundo bastante tardio devido ao tempo de resposta necessário de uma operação de leitura no barramento I²C de 100 KHz em comparação com o clock de 40 MHz operação do bloco. Os vários pulsos dos sinais de READ_ECS_RESPONSE_i e ECS_RESPONSE_VALID_o correspondem ao procedimento de *polling* do supervisor esperando o retorno do comando de leitura no barramento I²C do SCA.

7. AGRADECIMENTOS

Nós reconhecemos a ajuda de Sandro Bonacini, Alessandro Caratelli e Kostas Kloukinas (todos do CERN) para o

desenvolvimento do bloco de encapsulamento usando o protocolo HDLC em FPGA e suporte ao chip GBT-SCA, a ajuda da colaboração da eletrônica do subdetector de SciFi



Figura 13: Exemplo de operação de leitura no barramento I²C medidos, operando na frequência de 1 MHz, no osciloscópio LeCroy 44MXs-B.

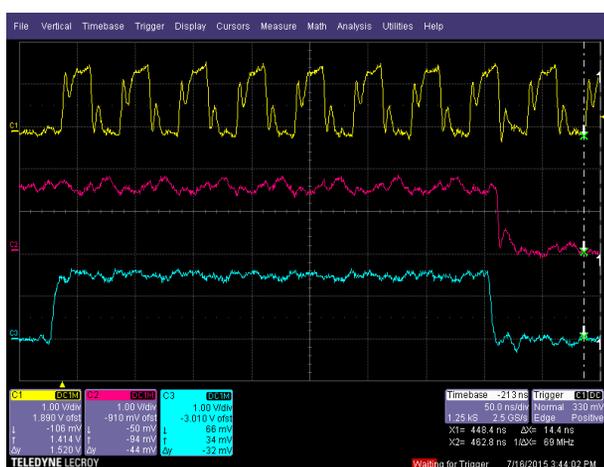


Figura 14: Exemplo de operação de leitura no barramento SPI medidos, operando na frequência de 20 MHz, no osciloscópio LeCroy 44MXs-Bs.

do LHCb Antonio Pellegrino e Wilco Vink (Nikhef) e Mauricio Féo (CBPF) na etapa de validação do projeto. Agradecemos ao CNPQ, CBPF, CERN e Nikhef pelo suporte a este projeto de pesquisa.

- [1] ALESSIO, F. et al. A generic firmware core to drive the front-end gbt-scas for the lhcb upgrade. *Journal of Instrumentation*, v. 10, n. 02, p. C02013, 2015. Disponível em: <<http://stacks.iop.org/1748-0221/10/i=02/a=C02013>>.
- [2] THE LHCb Detector at the LHC. *JINST*, v. 3, n. 08, p. S08005, 2008. Disponível em: <<http://stacks.iop.org/1748-0221/3/i=08/a=S08005>>.
- [3] PERFORMANCE of the LHCb Detector during the LHCb Proton Runs 2010-2012. *IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, v. 2012, p. 1479, 2012. Disponível em: <<http://dx.doi.org/10.1109/NSSMIC.2012.6551357>>.
- [4] *Letter of Intent for the LHCb Upgrade*, 2011.
- [5] *Framework TDR for the LHCb Upgrade*, 2012.
- [6] *LHCb Trigger and Online Upgrade Technical Design Report*,

- 2014.
- [7] TRIGGER-LESS readout architecture for the upgrade of the LHCb experiment at CERN. *JINST*, v. 8, n. 12, p. C12019, 2013. Disponível em: <<http://stacks.iop.org/1748-0221/8/i=12/a=C12019>>.
- [8] TIMING and Fast Control for the Upgraded Readout Architecture of the LHCb experiment at CERN. *IEEE Trans. Nucl. Sci.*, v. 60, p. 3438, 2013. Disponível em: <<http://dx.doi.org/10.1109/TNS.2013.2281210>>.
- [9] THE GBT Ser-Des ASIC prototype. *JINST*, v. 5, n. 11, p. C11022, 2010. Disponível em: <<http://stacks.iop.org/1748-0221/5/i=11/a=C11022>>.
- [10] THE GBT-SCA, a Radiation Tolerant ASIC for Detector Control and Monitoring Applications in HEP Experiments, in

- proceedings of Topical Workshop on Electronics for Particle Physics 2014.
- [11] A New Readout Control system for the LHCb upgrade at CERN. *JINST*, v. 7, n. 11, p. C11010, 2012. Disponível em: <<http://stacks.iop.org/1748-0221/7/i=11/a=C11010>>.
- [12] THE LHCb Experiment Control System: on the path to full automation, at 13th International Conference on Accelerator and Large Experimental Physics Control Systems. p. , pg. 20, 2011.
- [13] TELECOMMUNICATIONS and information exchange between systems — HDLC procedures. Disponível em: <<https://cds.cern.ch/record/442122>>.
- [14] BUDRUK, R.; ANDERSON, D.; SHANLEY, T. *PCI Express System Architecture*. [S.l.: s.n.].
- [15] CORPORATION, A. Atlantic interface. 2002. Disponível em: <http://www.altera.com/literature/fs/fs_atlantic.pdf>.
- [16] BONACINI, S.; KLOUKINAS, K.; MOREIRA, P. E-link: A Radiation-Hard Low-Power Electrical Link for Chip-to-Chip Communication. 2009. Disponível em: <<https://cds.cern.ch/record/1235849>>.
- [17] PETERSON, W. W.; BROWN, D. T. Cyclic codes for error detection. *Proceedings of the IRE*, v. 49, n. 1, p. 228–235, Jan 1961. ISSN 0096-8390.
- [18] JOHNSON, J. M.; WIRTHLIN, M. J. Voter insertion algorithms for fpga designs using triple modular redundancy. In: *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. New York, NY, USA: ACM, 2010. (FPGA '10), p. 249–258. ISBN 978-1-60558-911-4. Disponível em: <<http://doi.acm.org/10.1145/1723112.1723154>>.
- [19] HAMMING, R. W. Error detecting and error correcting codes. *Bell System technical journal*, Wiley Online Library, v. 29, n. 2, p. 147–160, 1950.
- [20] HOROWITZ, P.; HILL, W. *The Art of Electronics*. 3. ed. Cambridge: Cambridge University Press, 2015. 1224 p.
- [21] CERN. Disponível em: <<https://espace.cern.ch/GBT-Project/VLDB/default.aspx>>.
- [22] AMARAL, L. et al. The versatile link, a common project for super-lhc. *Journal of Instrumentation*, v. 4, n. 12, p. P12003, 2009. Disponível em: <<http://stacks.iop.org/1748-0221/4/i=12/a=P12003>>.
- [23] TROSKA, J. et al. Versatile transceiver developments. *Journal of Instrumentation*, v. 6, n. 01, p. C01089, 2011. Disponível em: <<http://stacks.iop.org/1748-0221/6/i=01/a=C01089>>.
- [24] SO6S, C. et al. The versatile transceiver: towards production readiness. *Journal of Instrumentation*, v. 8, n. 03, p. C03004, 2013. Disponível em: <<http://stacks.iop.org/1748-0221/8/i=03/a=C03004>>.
- [25] LEITAO, P. et al. Test bench development for the radiation hard gbtx asic. *Journal of Instrumentation*, v. 10, n. 01, p. C01038, 2015. Disponível em: <<http://stacks.iop.org/1748-0221/10/i=01/a=C01038>>.
- [26] IEEE Standard for Test Access Port and Boundary-Scan Architecture. *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, p. 1–444, May 2013.
- [27] HAWKINS, D. *Altera JTAG-to-Avalon MM Tutorial*. March, 2012. Disponível em: <http://www.ovro.caltech.edu/%7Edwh/correlator/pdf/altera_jtag_to_avalon_mm_tutorial.pdf>.
- [28] LHCb. Disponível em: <https://lbredmine.cern.ch/attachments/download/157/lhcb_upgrade_minidaq_handbook.pdf>.
- [29] WYLLIE, K. et al. *Electronics Architecture of the LHCb Upgrade*. Geneva, 2013. Previous version: 01/03/2013 Current version: 18/06/2013 (version 2.6) based on LHCb-INT-2011-006 (removed from CDS). Disponível em: <<http://cds.cern.ch/record/1340939>>.
- [30] CARNITI, P. et al. Claro-cmos, a very low power asic for fast photon counting with pixellated photodetectors. *Journal of Instrumentation*, v. 7, n. 11, p. P11026, 2012. Disponível em: <<http://stacks.iop.org/1748-0221/7/i=11/a=P11026>>.
- [31] COLLABORATION, L. *LHCb PID Upgrade Technical Design Report*. Geneva, 2013. Disponível em: <<http://cds.cern.ch/record/1624074>>.